International Journal of Engineering & Technology, 7 (3.29) (2018) 741-743

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

BCD Divider Architecture For High Speed VLSI Application Using Vedic Mathematics

Rishabh S [1*], Harshitha K M [2], Prajwal R Kashyap [3], Chaitanya C V S [4], Sundaresan C[5]

[1,2,3,4,5] School of Information Science, Manipal Academy of Higher Education, Manipal, Karnataka, India
**Corresponding Author E-mail : chaitanya.cvs@manipal.edu*

Abstract

Power, delay, area and speed are the parameters used to evaluate any processor performance in digital domain. These parameters should be less to get an effective results. In all VLSI architecture, Division operation is always considered to be more complex, time consuming and bulky. Vedic Mathematics gives a new perspective to mathematics. In this paper we have implemented a BCD division architecture using Nikhilam Sutra a formula (sutra) from the vedic mathematics. Here divider has been designed to improve the results of delay using simple algorithms. The proposed designed is modeled in Verilog, simulated and synthesized using Cadence EDA tools. The proposed Vedic BCD divider performed 29.9% faster than the existing method.

Keywords: *BCD divider, Vedic technique, Nikhilam Sutra.*

1. Introduction

The one of the most importance given in signal processing is processing of data using mathematical operations like Addition, multiplication and division. Division operation is more complex and has higher propagation delay compared to multiplication, addition and subtraction etc[1].
Many algorithms have been implemented and used, to make the operation more effective (to overcome complexities) such as digit repetition problems (recurring, non-recurring), other complex methods namely Newton-Raphson method (convergence techniques), and Goldschmidt algorithm (series expansion technique).These techniques involve large computations and has more number of iterations thereby chip area and hardware complexities reduces. In other words, the propagation delay of a system increases during these computations[2],[3].
However, the researchers work on a larger and more complex implementations which involve high radices. So by reducing the number of iterations improves the speed of the system with such algorithms with which we can achieve a better performance which is inter-dependent with the hardware complexities[4].
Newer division techniques have been developed in order to establish better outcome in terms of reduction in delay of the divider circuit; which reduces the iterations. One of the ancient techniques of computation in Indian mathematics is Vedic Mathematics and have a variety of Sutras which support easy and reliable computation compared to conventional methods[5],[6]. Here, by using the Nikhilam sutra we portray the results on division algorithms and their architectures based on Vedic Mathematics.

2. Division Algorithm Using Vedic Mathematics

Vedic Mathematics comprises of 16 Sutras. These Sutras are simple to understand and play a vital role in resolving tedious and time-consuming operations and to a major extent helps to execute mentally with ease. These Sutras covers all the major branches of mathematics like arithmetic, algebra and geometry. This paper gives a complete overview of BCD division algorithm using Nikhilam Sutra.

2.1. Nikhilam Sutra

The Nikhilam sutra works as below:
Nikhilam Navatascaramam Dasatah (NND), is a Sanskrit verse which means “all from 9 and the last from 10” is derived from the Vedas. Nikhilam division Sutra is a different technique which just involves only addition and multiplication.

2.2. Nikhilam Sutra for Decimal Numbers

We take an example to illustrate this method. Let us compute 20/9 using the Sutra.

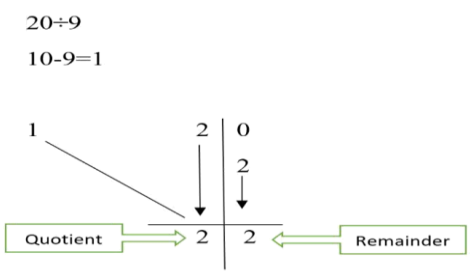

Fig.1: Nikhilam sutra for Decimal division.

The algorithm can be understood by the following steps:
Step 1: Consider Dividend as 20 and Divisor as 9, with 10^n being the base of operation. i.e Divisor should be nearer to 10^n. Here it is 10. Now, subtract the divisor from base of operation i.e. equals to 1(10-9=1).
Step 2: Consider MSB of the dividend and this is written down below as the MSB, which is equal to 2.
Step 3: MSB is multiplied by the subtraction results. Now the result is equal to 2(1x2=2).
Step 4: Sum and left-most digit of the 10's complement product will be placed under the next digit of the dividend(i.e 0), while sum and the digit to its right product will be placed under the next digit to the right.
Step 5: Add the LSB dividend digit with multiplication result.
Step 6: If the divisor is lesser than the remainder, follow the same steps from the beginning and quotient is the incremented and the current remainder is considered as the remainder.

2.3. Nikhilam Sutra for a BCD Number

We take same example for BCD numbers to compute division using sutra

$0010\ 0000_{BCD} \div 1001_{BCD}$

10's Complement 1001-1001 = 0000

$+\ 1$

0001

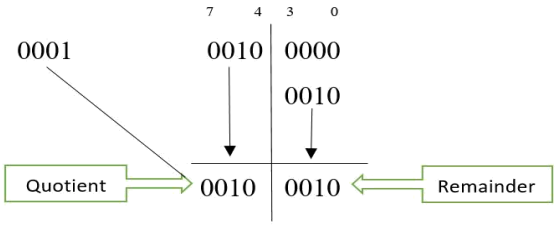

Fig.2: BCD division using Nikhilam sutra

Following are steps of algorithm:
Step 1: Assuming Dividend as 00100000_{BCD}(20) and Divisor as 1001_{BCD}(9). Here it is 1010_{BCD}(10). Subtract the divisor from base of operation which result in 0001_{BCD}(1).
Step 2: Take the 4 bits of MSB (i.e. 0010_{BCD}) of the dividend and place below as MSB, which is equal to 0010_{BCD}(2).
Step 3: MSB is multiplied with subtraction results, placed under the dividend of [3:0]. Result is equal to (0001x0010=0010).
Step 4: The product of sum and the left-most digit of the 10's complement is placed under the next digit of the numerator [3:0] LSB i.e. 0000. while sum and the digit to its right product will be placed under the next digit to the right.
Step 5: Add LSB [3:0] of dividend digits with the multiplication result. From the above Fig.2 the quotient is equal to 0010_{BCD} and the remainder is 0010_{BCD}.
Step 6: If the divisor is lesser than the remainder, follw the same steps from the beginning and quotient is the incremented and the current remainder is considered as the remainder.
By using Nikhilam Sutra technique, we can calculate quotient and remainder without performing subtraction unlike in conventional method. Hence it is easier to perform division operation which involves only multiplication and addition.

2.4. Architecture and Specification

The architecture for proposed Vedic divider is shown in Fig.3. The design has 8-bit divisor (A) and a 4-bit dividend (B). The proposed design consists of six major following blocks

(i) 10's Complement block,
(ii) 4x4 Multiplier,
(iii) Adder block,
(iv) 2:1 Multiplexer,
(v) Decision box and
(vi) Incremental block.

The A[3:0] from divisor is given to the 10's Complement block. Initially select line (carry) of the Multiplexer is zero, therefore Multiplexer's output will be B[7:0]. The output of the Complement block is given to the Multiplier MSB i.e. [7:4] of Multiplexer and same 4 MSB of dividend i.e. B[7:4] is fed to the Incremental block. The result of Multiplier is given to the Adder with LSB i.e. [3:0] of Multiplexer. The result of Adder is again given to Multiplexer as new dividend if divisor is less than result of Adder. Now the quotient is incremented by 1 and remainder is updated as the new remainder.

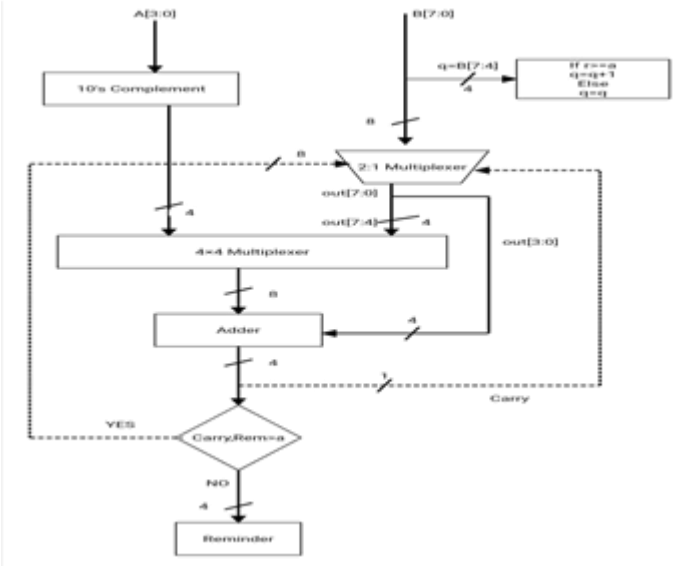

Fig 3: Architecture of BCD Vedic Divider

3. Results

The division algorithm is modeled using Verilog HDL for 8-bit dividend by 4-bit divisor. It is simulated and synthesized using Cadence EDA tools. All the tested parameters are shown below. The proposed Vedic BCD divider reports a total delay of 1.15ns and total power consumption of 10.94mW. It shows a significant reduction of 29.9% in total delay compared to the conventional method. The results are given as,

3.1. Simulation Results

The tested examples 20/9, 29/9, 22/8, 40/9, 52/9, 72/9 and 19/9 are shown in Fig4.The simulated results for Vedic BCD divider using Cadence simulator for the divisor width as 4 bits and the dividend width as 8 bits according to the design. We could observe that all the examples which are tested has given required quotient and remainder.

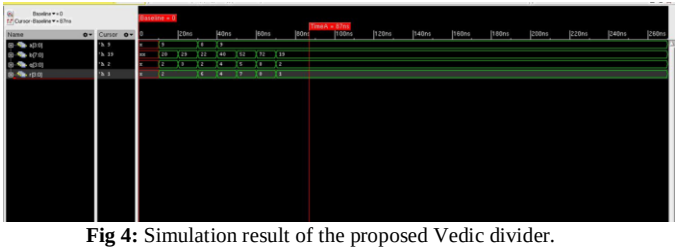

Fig 4: Simulation result of the proposed Vedic divider.

3.2. Synthesis Results

The proposed architecture consists of six major blocks namely complementary circuit, multiplexer, multiplier, adder, incremental block, and decision box. We took same examples and same data width (8bits dividend and 4bits divisor) for the comparison between Vedic Division method and Normal division method (which involves multiplication and subtraction).And also we calculated the total delay, power consumption, area and the number of gates using the Cadence design suit 13.1 is shown in the Fig.5.We observed that we got a significant results in all the parameters discussed above by using our proposed Vedic divider.

Description	Proposed Vedic Divider	Normal Repetitive Division Method
Tool	Cadence design suit 13.1	Cadence design suit 13.1
Total Delay	1.150 ns	1.494 ns
Power	10.94 mW	42.10 nW
Area	8326 nm^2	38563 nm^2
Gates	524	1080

Fig 5: Comparison Table

3.3. RTL Schematic

The logical block diagram of the proposed BCD divider (Fig 6) circuit is taken from the Cadence RTL compiler. The RTL compilation gave a total gate count of 524 whereas it was 1080 in the conventional divider for the same number of bit division. Hence we observed a prominent reduction in the area covered by the design.

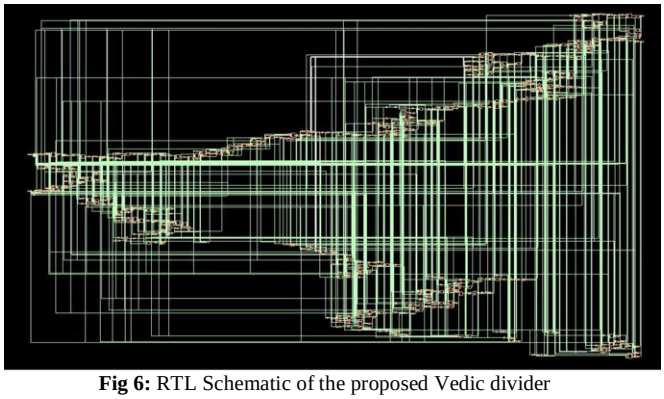

Fig 6: RTL Schematic of the proposed Vedic divider

4. Conclusion

The aim of the proposed design is to introduce Vedic methods for performing the division. Using the Nikhilam Sutra, we could successfully apply and test the implementation of BCD division in Verilog coding style. We have implemented the design which gave efficient results for both the quotient and the remainder after division. The Vedic divider algorithm is 29.9% faster than the conventional divider and showed a significant gate count reduction which in turn reduces the overall area of the circuit.

References

[1] Saha, Prabir, Arindam Banerjee, Partha Bhattacharyya, and Anup Dandapat. "Vedic divider: Novel architecture (ASIC) for high speed VLSI applications." In Electronic System Design (ISED), 2011 International Symposium on, pp. 67-71. IEEE, 2011.

[2] Obermann, Stuart F., and Michael J. Flynn. "Division algorithms and implementations." IEEE Transactions on computers 46, no. 8 (1997): 833-854.

[3] Ercegovac, MiloTs Dragutin, and Tomas Lang. Division and square root: digit-recurrence algorithms and implementations. Kluwer Academic Publishers, 1994.

[4] Senapati, Ratiranjan, Bandan Kumar Bhoi, and Manoranjan Pradhan. ”Novel binary divider architecture for high speed VLSI applications.” Information & Communication Technologies (ICT), 2013 IEEE Conference on. IEEE, 2013.

[5] Tirtha, Swami Bharati Krishna, and Vasudeva Sharana Agrawala. Vedic mathematics. Vol. 10. Motilal Banarsidass Publ., 1992.

[6] Jain, Surabhi, and Sandeep Saini. "High speed convolution and deconvolution algorithm (Based on Ancient Indian Vedic Mathematics)." In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference on, pp. 1-5. IEEE, 2014.

[7] Sengupta, Diganta, Mahamuda Sultana, and Atal Chaudhuri. "An algorithm facilitating fast BCD division on low end processors using Ancient Indian Vedic Mathematics Sutras." In Communications, Devices and Intelligent Systems (CODIS), 2012 International Conference on, pp. 373-376. IEEE, 2012.