Sādhanā (2021)46:83
https://doi.org/10.1007/s12046-021-01605-4

Performance analysis of Vedic mathematics algorithms on re-configurable hardware platform

RHEA BIJI and VIJAY SAVANI*

Department of Electronics and Communication Engineering, Institute of Technology, Nirma University, Ahmedabad, India
e-mail: 19mecv15@nirmauni.ac.in; vijay.savani@nirmauni.ac.in

MS received 26 November 2020; revised 18 January 2021; accepted 10 February 2021

Abstract. For the overall performance of systems like microprocessors and digital signal processors (DSPs) platforms, arithmetic units, all must be efficient in terms of speed, power, and area. Multipliers and dividers are inevitable hardware employed in such systems. This paper focuses on Vedic mathematics algorithms for multiplication and division for power-efficient, faster, and area-efficient design. For four- and eight-bit Vedic multiplication algorithms, Urdhva Tiryagbhyam and Nikhilam Sutras are employed in this paper. For eight-bit Vedic division algorithms, Nikhilam and Dhwajank Sutras are used. The Vedic mathematics algorithms are also compared to conventional methods of multiplication (like Array multiplier) and division (using Booth multiplication algorithm). As an application of DSP, the linear convolution operation is implemented using both conventional and Vedic algorithms. It has been observed that the Vedic algorithms operate faster, consume less power, and occupy less area on a targeted hardware platform. The implementations were carried out using the Verilog HDL language and Xilinx's Vivado EDA tool. To measure various performance parameters, Cadence simvision (using 180-nm GPDK CMOS Technology) and Xilinx's ISE tool were also used.

Keywords. Digital signal processing; Vedic mathematics algorithms; Urdhva Tiryagbhyam; Nikhilam Sutra; Verilog.

1. Introduction

Vedic mathematics has been the source of inspiration in the field of computation for many centuries [1]. It is an ancient system of mathematics based on 16 sutras [2]. Out of these, Urdhva Tiryagbhyam and Nikhilam Sutras for multiplication and Nikhilam Sutra and Dhwajank Sutra for division are used in this paper. Multipliers and dividers, both are an important computational unit in a processor and controllers. It affects the overall performance of the system also [3]. Hence, the optimization of area, speed, and power of these units is very important [4, 5]. Thus Vedic mathematics concepts and their algorithms are used in this paper to achieve a reduction in major performance parameters, i.e. area, power, and speed (delay), as compared with conventional multiplier and division architectures. Using Vedic multiplication, a convolution operation is implemented. This is a time-efficient way of implementing the operation [6].

In [1] the authors have presented and compared the performance of different division sutras, namely Nikhilam, Paravartya, and Dhwajank. Principles in developing a completely new division logic for the base 2 number system are discussed. The logic is much better on the count of power consumption, hardware requirements, execution time, and space management. Key advantages from all the sutras have been taken and a novel architecture is developed for the fast and efficient division [7].

In [8], the authors have proposed a multiplier architecture for signed multiplication. Signed multiplier architecture is based on two's complement and for unsigned data, Urdhva Triyakbhyam Vedic multiplier is utilized. A carry select adder is used in this method while calculating the partial product. The major advantage obtained is less combinational path delay compared with the existing methods [9].

In [6], the convolution operation is implemented. It includes multiplication and additions in it. To improve the overall speed of the convolution operation, Vedic (Urdhva Triyakbhyam) multiplier is used.

The paper is organized as follows. In section 2, a conventional Array multiplier and a conventional Binary divider are discussed. In section 3, Vedic algorithms for multiplication and division are discussed. Section 4 presents the performance analysis of the algorithms. Implementation and its comparison of conventional and Vedic algorithms are highlighted and presented in tabular form. Finally, the concluding remarks are drawn in section 5.

*For correspondence
Published online: 21 April 2021

2. Conventional multiplier and divider

In this section, the conventional methods of multiplication and division are explained. Four- and eight-bit Array multipliers and eight-bit Binary dividers are described.

2.1 *Four-bit Array multiplier*

The whole process requires multiplication and addition operations. In binary, two bits can be multiplied using the AND operation. The final result is obtained by adding the partial products and the carry generated from the previous additions [10, 11]. Here the full adders are being used for this purpose. Four full adders have the third input as a fixed value "0". Hence, they are equivalent to half adders. For a 4×4 Array multiplier, 16 AND gates, 4 half adders, and 8 full adders (totally 12 adders) are required.

2.2 *Eight-bit Array multiplier*

The structure of the eight-bit Array multiplier can be realized by extending the four-bit Array multiplier structure. In general $x * y$ multiplier needs $x * y$ AND gates, y half adders, and $(x-2) * y$ full adders (totally $(x-1) * y$ adders). Thus eight-bit Array multiplication needs 64 AND gates, 8 half adders, and 48 full adders (totally 56 adders).

2.3 *Binary divider*

In division operation, dividend and divisor are the inputs and the outputs are in the form of quotient and remainder. Mainly, a counter is present and some decisions need to be taken. Also, shifting left operation is used. Booth's division algorithm is implemented for an 8-bit division operation. According to the steps, a program is written using Verilog HDL and simulated to verify the functionality.

3. Vedic mathematics algorithms

In this section, Vedic mathematics algorithms for multiplication and division are presented and explained. Two algorithms for multiplication (i.e. Urdhva Tiryagbhyam Sutra and Nikhilam Sutra) and two algorithms for division, (i.e. Nikhilam Sutra and Dhwajank Sutra, are discussed).

3.1 *Urdhva Tiryagbhyam Sutra for multiplication*

This sutra is known as Vertical and Cross-wise multiplication. It is generic for any of the n-bit multiplication [12]. The multiplication and the addition operations are done in parallel [13].

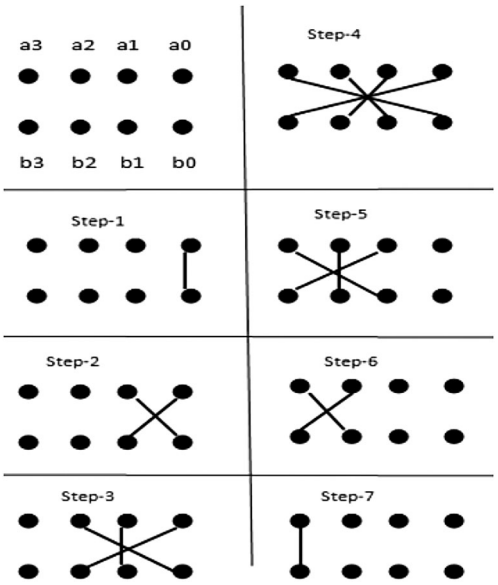

Figure 1. Line diagram for 4×4 multiplication [13, 14].

The line diagram in figure 1 shows the steps involved in 4×4 Vedic multiplication. The same procedure can be extended to build higher-order multipliers [13].

A four-bit Vedic multiplier is made using 4 two-bit Vedic multipliers and 3 adders; 1 four-bit adder and 2 eight-bit adders are required for the same. An eight-bit Vedic multiplier is made with the help of 4 four-bit Vedic multipliers and 3 adders. Totally, One 8-bit adder and two 12-bit adders are needed for this implementation.

3.2 *Nikhilam Sutra for multiplication*

In this method, the complement of the larger number from its nearby base is calculated (see figure 2) before multiplication [13].

3.3 *Nikhilam Sutra for division*

The basics of division using Nikhilam Sutra for the division is shown in figure 3. Complement of divisor, multiplication, addition, incrementing of the counter, and comparison are the steps involved in this sutra [1].

3.4 *Dhwajank Sutra for division*

Mainly addition and multiplication of cross-products are involved in the Dhwajank Sutra operation. MSBs of the divisor are kept aside. Later, the MSB of the dividend is divided with the MSB of the divisor. Cross-product of quotient and rest of the bits is taken and addition is done.

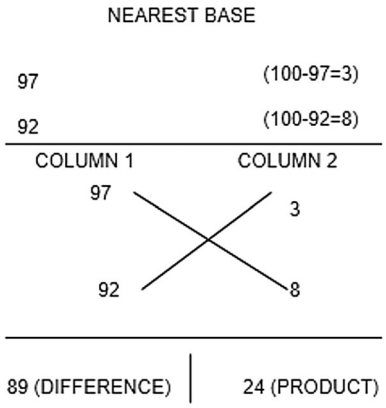

Figure 2. Example of Nikhilam (base) Sutra [13].

Denominator 1001	Numerator=1010001						
Deficit=10000-1001=0111	Bits allotted for quotient			Bits allotted for remainder			
	1	0	1	0	0	0	1
		0	1	1	1		
			0	0	0	0	
			0	01	01	01	
	1	0	10	11	0	1	1

Figure 3. Example of Nikhilam (base) Sutra [15].

Sum is subtracted from a combination of the previous remainder and the next digit of the dividend.

The final remainder is obtained by subtraction of the right part of dividend prefixed by the last remainder and cross-multiplication of quotient and divisor [16].

3.5 *An application of digital signal processor (DSP): linear convolution*

Linear convolution helps to estimate the output of the system when an arbitrary input and the impulse response is available [17]. Basically, in linear convolution, multipliers and adders are element components for this operation. The convolution operation is done using two types of multipliers: Array multiplier and Vedic multiplier. The delays obtained are compared in the Implementation and comparisons section. It has been observed that the operations performed using the Vedic multiplier are faster as compared with the conventional method.

4. Implementation and comparisons

In this section, the whole gist of the work is summarized in terms of implementation results and comparison. Results obtained are compared to see the significance of the Vedic algorithms in the arithmetic unit. The implementation and results for power are obtained using the Cadence EDA tool with 180-nm GPDK CMOS Technology. The Xilinx Vivado tool is used to figure out the area and delay (speed) of the respective algorithms.

A conventional 4×4 Array multiplier is compared with 2 4-bit Vedic multipliers for performance parameters area, delay, and power. The results of these performance parameters are enumerated in table 1. There is a significant decrease in the delay due to the efficient Vedic algorithm. The other two parameters, which are area and power, do not significantly decrease as compared to the conventional method/algorithms. Also, the performance parameters of the Vedic multiplier are compared to the results presented [2, 8].

Percentage reduction in area, delay, and power as compared with conventional method are listed in table 2. Percentage reduction is calculated as *[(value in the conventional algorithm – value in Vedic algorithm)/value in conventional algorithm]* $\times 100$. Significant improvement in the delay is observed. The other two parameters have comparable reduction as shown in table 2.

A conventional 8×8 Array multiplier is compared with two 8-bit Vedic multipliers for performance parameters and their results are presented in table 3. A significant decrease in delay, area, and power is observed using both the Vedic algorithms. Also, performance parameters of Vedic multiplier from literature are mentioned in it.

Percentage reduction in terms of area, delay, and power as compared with the conventional method are listed in table 4. Significant improvement in the area, power, and delay is observed. Thus Vedic multiplication is faster and the resources required to store the intermediate values are

Table 1. Performance comparison of 4-bit conventional and Vedic multipliers (NR: not reported in corresponding reference).

Parameters	Array multiplier (4×4)	Vedic_Crisscross multiplier (4×4)	Vedic_Nikhilam multiplier (4×4)	Multiplier [2]	Multiplier [8]
Cells (area)	39	37	58	24	39
Power (nW)	48475.89	46363.21	46005.71	NR	125.06
Delay (ns)	5.85	3.75	3.29	10.43	14.62

Table 2. Percentage reduction in area, power, and delay for 4-bit multipliers.

Percentage reduction as compared with conventional multiplier (%)	Vedic_Crisscross multiplier (4 × 4)	Vedic_Nikhilam multiplier (4 × 4)
Cells (area)	5	No reduction
Power (nW)	4.35	5.10
Delay (ns)	35.9	43.8

decreased. Finally, the reduction in delay and area helps in the reduction of the overall power of the system.

In table 5 and table 6, a conventional divider is compared with two Vedic dividers (i.e. Nikhilam and Dhwajank) for performance evaluation. Dividers have a higher area than the multipliers. Hence, improvement using Vedic dividers is very useful to reduce the overall area, power, and delay of the system.

As seen from table 7, Vedic algorithms have advantages in terms of delay when applied to a convolution operation in DSP-related applications. The Vedic multiplier used here is the Urdhva Tiryagbhyam multiplier. Percentage reduction in terms of delay obtained in this paper and a reference paper is also mentioned in the table.

As seen from table 8, 22.64% reduction in the delay is observed when Vedic multiplier is used in the convolution operation as compared with a conventional Array multiplier. Percentage reduction as per [17] is 28%. Hence, it can be concluded that convolution operation can be performed significantly faster when the Vedic multipliers are used.

5. Conclusion

Vedic algorithms have advantages in terms of power, area, and delay. Thus they are used in systems like DSPs and microprocessors so that the overall system becomes efficient. In this paper two arithmetic units, i.e. multipliers and dividers, are implemented using Vedic algorithms. For 4-bit Urdhva multiplier, 5%, 35.9%, and 4.35% reductions in area, delay, and power, respectively, are obtained as compared with the conventional 4-bit Array multiplier; for the 4-bit Vedic multiplier (Nikhilam), 5.1% and 43.8%

Table 4. Percentage reduction in area, power, and delay (8-bit multipliers).

Percentage reduction as compared with conventional multiplier (%)	Vedic_Crisscross multiplier (8 × 8)	Vedic_Nikhilam multiplier (8 × 8)
Cells (area)	26.3	20.53
Power (nW)	19.0	5.4
Delay (ns)	70.00	68.25

Table 5. Performance comparison of 8-bit conventional and Vedic dividers.

Parameters	Binary divider	Vedic_Nikhilam divider	Vedic_Dhwajank divider
Cells (area)	238	41	30
Power (nW)	3,71,982.66	3,01,062.29	3,51,893.61
Delay (ns)	22.59	2.27	17.08

Table 6. Percentage reduction in area, power, and delay for 8-bit dividers.

Percentage reduction as compared with conventional divider (%)	Vedic_Nikhilam divider	Vedic_Dhwajank divider
Cells (area)	82.77	87.39
Power (nW)	19.06	5.4
Delay (ns)	89.0	24.4

Table 7. Convolution using Array and Vedic multipliers.

Parameters	Convolution using Array multiplier	Convolution using Vedic multiplier
Delay (ns)	7.89	6.11

Table 3. Comparison of 8-bit conventional and Vedic multipliers (NR: not reported in corresponding reference).

Parameters	Array multiplier (8 × 8)	Vedic_Crisscross multiplier (8 × 8)	Vedic_Nikhilam multiplier (8 × 8)	Multiplier [2]	Multiplier [8]
Cells (area)	190	140	151	125	159
Power (nW)	1376457.98	31896.31	73683.43	NR	138.45
Delay (ns)	22.42	6.69	7.12	18.46	23.67

Table 8. Percentage reduction in delay (convolution using Vedic multiplier).

Parameters	Convolution using Vedic multiplier
Reduction in delay [this work]	22.64%
Reduction in delay [17]	28.00%

reduction in power and delay are observed. Implementation of 8-bit Urdhva multiplier results in 26.3%, 70%, and 19% reduction in terms of area, delay, and power, respectively, compared with the conventional 8-bit Array multiplier; for the 8-bit Nikhilam multiplier, 20.53%, 68.25%, and 5.4% reductions in area, delay, and power are obtained; nearly 80% reduction in terms of area and power is observed for 8-bit Nikhilam and Dhwajank dividers as compared with the conventional divider whereas reduction in the delay is 89% and 24.4%. Convolution using Vedic operations has a reduction in delay of 22.64% over conventional methods. Implementations can be extended to 16-, 32-, 64-bit multipliers and dividers.

References

[1] Tadas A and Rotake D 2015 64 bit divider using vedic mathematics. In: *Proceedings of the 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, IEEE, pp. 317–320

[2] Akhter S and Chaturvedi S 2019 Modified binary multiplier circuit based on vedic mathematics. In: *Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, pp. 234–237

[3] Kishor D R and Bhaaskaran V K 2014 Low power divider using vedic mathematics. In: *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, pp. 575–580

[4] Thakur K and Sharma T 2019 Area efficient high speed vedic multiplier. *International Journal of Innovative Technology and Exploring Engineering* 8(9S): 302–306

[5] Akhter S, Saini V and Saini J 2017 Analysis of vedic multiplier using various adder topologies. In: *Proceedings of the 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, pp. 173–176

[6] Batham N and Anjum S 2016 Algorithm for convolution operation in DFT using vedic multiplication. *International Journal of Engineering Innovations and Research* 5(5): 288–291

[7] Toro S, Patil A, Chavan Y V, Patil S C, Bormane D S, and Wadar S 2016 Division operation based on vedic mathematics. In: *Proceedings of the 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, IEEE, pp. 450–454

[8] Pichhode K, Patil M D, Shah D and Rohit B C 2015 FPGA implementation of efficient vedic multiplier. In: *Proceedings of the 2015 International Conference on Information Processing (ICIP)*, IEEE, pp. 565–570

[9] Sapkal K J and Shrawankar U 2017 Complexity analysis of vedic mathematics algorithms for multicore environment. *International Journal of Rough Sets and Data Analysis* 4: 31–47

[10] Kumar A 2017 Comparative analysis of vedic and array multiplier. *International Journal of Electronics and Communication Engineering and Technology* 8(3): 17–27

[11] Sriraman L and Prabakar T N 2012 Design and implementation of two variable multiplier using KCM and vedic mathematics. In: *Proceedings of the 2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, IEEE, pp. 782–787

[12] Sudeep M C, Bimba M S, and Vucha M 2014 Design and FPGA implementation of high speed vedic multiplier. *International Journal of Computer Applications* 90(16): 10.5120/15802-4641

[13] Prasada G S V, Seshikala G, and Sampathila N 2018 Performance analysis of 64×64 bit multiplier designed using urdhva tiryakbyham and nikhilam navatashcaramam dashatah sutras. In: *Proceedings of the 2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, IEEE, pp. 28–31

[14] Prasada G S V, Seshikala G, and Niranjana S 2019 Design of high speed 32-bit floating point multiplier using urdhva triyagbhyam sutra of vedic mathematics. *International Journal of Recent Technology and Engineering* 8(2 special issue 3): 1064–1067

[15] Rajani M and Sridevi N 2015 Survey on implementation of IEEE754 floating point number division using vedic techniques. *International Journal of Engineering Development and Research* 3(3)

[16] Ugra Mohan Kumar S K, Singh M P, and Yadav A K 2017 Fast and efficient division technique using vedic mathematics in Verilog code. *International Journal of Scientific and Engineering Research* 8(10): 99–103

[17] Punwantwar N R and Chatur P N 2015 Convolution and deconvolution using vedic mathematics. *International Journal of Advanced Research in Electrical, Electronics, and Instrumentation Engineering* 4(6): 5216–5223