HIGH-SPEED VEDIC MULTIPLIER BASED ON QUATERNARY SIGNED DIGIT NUMBER SYSTEM DESIGNED USING VLSI

[1]S.Anbukaruppu Sami, [2]N.Lakshmi, [3]K.Ramesh, [4]M.Srilatha
[1,2,3]Assistant Professor, [4]UG Student, [1,2,3,4]Department of Electronics and Communication Engineering,
Visvesvaraya College of Engineering & Technology, Hyderabad, India.

Abstract
The Urdhva Tiryagbhyam sutra of Vedic mathematics, which is the basis for the high-speed Vedic multiplier presented in this study, also includes a unique adder that is entirely based on the Quaternary Signed digit number system. Multiplication naturally includes the three processes of partial product production, partial product discount, and addition. Consequently, a rapid adder structure significantly increases the speed of the regular procedure. It is suggested to use an architecture for quaternary logic adders that is a combination of binary and quaternary variety systems. The carry propagation latency is reduced by first splitting a given binary string into quaternary digits of two bits each, proceeded by parallel addition. The plan doesn't require a radix conversion module as the sum is immediately generated in binary the usage ofthe novel concept of an adjusting bit. The proposed multiplier layout is in contrast with a Vedicmultiplier based on multi voltage or multi cost goodjudgment [MVL], Vedic Multiplier that comprises aQSD adder with a conversion module forquaternary to binary conversion, Vedic multiplier that uses Carry Select Adder and a oftentimes used quickly multiplication mechanism such as Boothmultiplier. All these designs have been developed using Verilog HDL and synthesized through Synopsys Design Compiler. They have been realized the usage of the open source NAN gate 15nm science library. The inspiration suggests amaximum of 88.75% velocity enchantment with recognize to Multi Value good judgment primarily based 128x128 Vedic multiplier while the minimumis 17.47%.
Keywords: Multiplier; Quaternary Signed Digit adder [QSD]; Urdhva Tiryagbhyam; Vedic Mathematics

INTRODUCTION

The speed of a processor's math unit is one of the key characteristics that helps us determine the degree of computation it can handle. Due to the fact that it accounts for the bulk of the execution time in most numerical calculations, an important capability of a number-crunching square is expansion. In this sense, the development of a rapid multiplier has long been a focus of research.

The Array, Booth, and Wallace multipliers [1]–[5] are some of the main computations suggested for speedy writing enhancement. Vedic Mathematics [6, 7] is a method of making numerical judgments that considers progressively efficient use with respect to speed. Three stages make up the increase in this process: the age of midway objects, the decline of unfinished items, and convey proliferate expansion. Multiplier configuration dependent on Vedic arithmetic hasnumerous points of interest as the fractional items and wholes are created in one stage, which diminishes the convey engendering from LSB toMSB.

The last advance, convey spread expansion, requires a quick snake plot since it shapes a piece ofthe basic way. An assortment of viper plans has been proposed in writing to streamline the exhibition of Vedic multiplier [13]. Viper dependent on QSD demonstrates an improvement inspeed over other cutting edge adders [14, 15]. Prior usage of QSD viper depended on Multi Voltage or Multi Value Logic (MVL) [16]. The trouble in use of quaternary expansion outside MVL (Multi Voltage rationale) is that, the snake is just a little unit of the plan whose yields will should have been changed over back to parallel for further preparing. Be that as it may, utilization of a change module undermines the focal points picked up in speed by utilizing QSD. In this paper, a novel execution of a snake dependent on QSD is proposed, whichdiminishes the convey proliferation delay in the plan by utilizing convey free number juggling. The proposed snake configuration chips away at a cross

breed of paired and quaternary number frameworks wherein the entirety is legitimately created in twofold utilizing the idea of an altering bit, wiping out the transformation module. The structure can bescaled to bigger piece usage, for example, 32, 64,
128 or more with negligible increment inengendering postpone inferable from the parallelismcommon in the plan. We have contrasted our structure and a Vedic multiplier dependent on MVLrationale that uses a swell convey snake [16], Vedic Multiplier that consolidates a QSD viper and a transformation module for quaternary to twofold change, Vedic multiplier that utilizations best in class quick snake plan, for example, Carry select viper [17] and a regularly utilized quick increase instrument, for example, Booth multiplier [18], to demonstrate the attainability of our plan crosswise over significant examination focuses.
This paper is sorted out as pursues. Segment II portrays the Proposed Multiplier engineeringdependent on Vedic multiplication and Quaternary expansion. Segment III includes Result in which gadget use synopsis and computational way deferralgot for the proposed Vedic multiplier (after blend) is talked about and Section IV comprises ofConclusion.

PROPOSED ARCHITECTURE

A. 4x4 Multiplier Block layout of a 4x4 multiplier is proven inFig. 1 In this multiplier, four 2x2 multipliers are organized systematically. Each multiplier accepts 4 input bits; two bits from multiplicand and different two bits from multiplier. Addition of partial products is performed the usage of two 4 bit Quaternary adders, a two-bit adder and a half of adder. The last end result is received by concatenating the least vast two bits of the first multiplier, four sum bits of the 2nd four-bitQuaternary adder and the sum bits of two-bit adder.

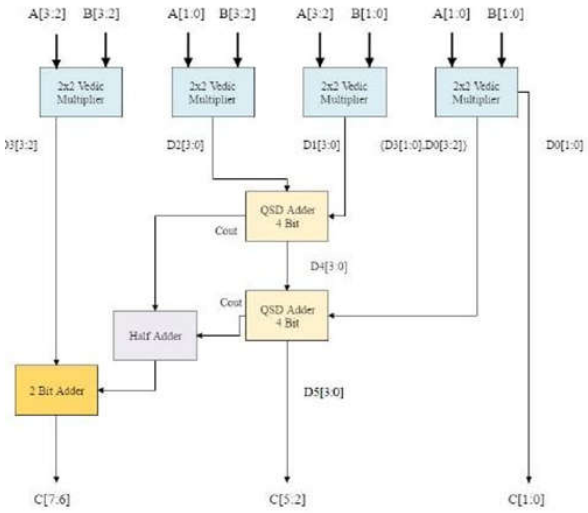


Fig. 1. Proposed 4x4 Multiplier

Table I shows all intermediate and final results involved in the multiplication process of two binary numbers, A = (1111)2 and B = (1001)2.
The data flow in the proposed 4x4 multiplier isgiven below:
1) A[1:0] and B[1:0], A[3:2] and B[1:0], A[1:0] and B[3:2], and A[3:2] and B[3:2] are multiplied by 2x2Vedic multipliers, giving output D0[3:0], D1[3:0], D2[3:0] and D3[3:0] respectively.
2) D1 [3:0] and D2[3:0] are added by the proposed 4 bit QSD adder, giving D4[3:0] and a carry out as the outputs.
3) D4[3:0] and {D3[1:0], D0[3:2]} are added by thesecond 4 bit QSD adder, giving D5[3:0] and a carry out as the outputs.
4) The half adder is used to add the carry outs of the QSD adders. The output obtained is fed to the 2 Bit Adder along with D3[3:2].

5) The result, C, in binary is obtained by concatenation of output of 2 Bit Adder, D5[3:0] andD0[1:0].

The proposed plan can be reached out toduplicate both negative and positive numbers by an expansion of a sign piece in the two information sources. A XOR rationale would then be able to be utilized to process the sign piece of the last yield. The increase of the sizes will continue at the same time along these lines to the model depicted previously.

Table I. Multiplication Result of Two 4 Bit Binary Numbers using TheProposed Design

	Binary equivalent	Decimal equivalent	Explanation
A	$(1111)_2$	15	Input 1
B	$(1001)_2$	9	Input 2
D0	$(0011)_2$	3	Output of 2x2 Vedic Multiplier 1
D1	$(0011)_2$	3	Output of 2x2 Vedic Multiplier 2
D2	$(0110)_2$	6	Output of 2x2 Vedic Multiplier 3
D3	$(0110)_2$	6	Output of 2x2 Vedic Multiplier 4
D4	$(01001)_2$	9	Output of 4 bit QSD adder 1 (D1+D2)
D5	$(10001)_2$	17	Output of 4 bit QSD adder 2 (D4 +{D3[1:0],D0[3:2]})
C[1:0]	$(11)_2$	3	D0[1:0]
C[5:2]	$(0001)_2$	1	D5[3:0]
C[7:6]	$(10)_2$	2	Output of 2 Bit Adder (D3[3:2]+D4[4]+D5[4])
C[7:0]	$(10000111)_2$	135	Final Result

A. 32x32 multiplier The 4x4 multiplier design can be scaled tomultiply larger numbers as shownin Fig.2, where the design is scaled up for a 32 bit multiplier

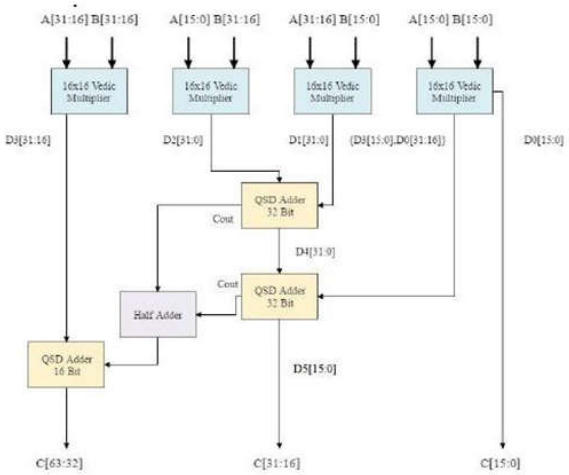


Fig. 2. Proposed 32x32 Multiplier

C. Proposed adder design based on QSD

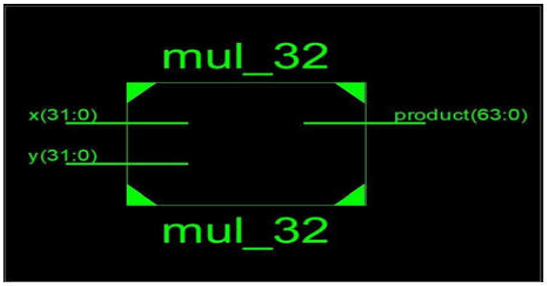
In this paper, a novel concept of an adder, basedtotally on QSD (Quaternary Signed Digit) is proposed. The algorithm for the proposed adder uses a hybrid of quaternary and binary range systems. The outputs from smaller multipliers are received as binary strings. Inside the addition module, this string is damaged into quaternary digits of two bits each. Addition the usage of QSD lets in us to minimize the lift propagation lengthen by making use of elevate free arithmetic i.e. the lift doesn't ripple previous the subsequent quaternary digit. Especially for greater bit enter strings thismethod is extremely efficient.

The concern in application of quaternary addition outside MVL (Multi Voltage logic) is that the least massive 2 bits of the binary representation of the quaternary digits can't be at once concatenated to structure an output binary string for each case as depicted in Table II. Each string would have to be study in my view and a conversion module that converts quaternary to binary would have to be employed. To overcome this limitation,

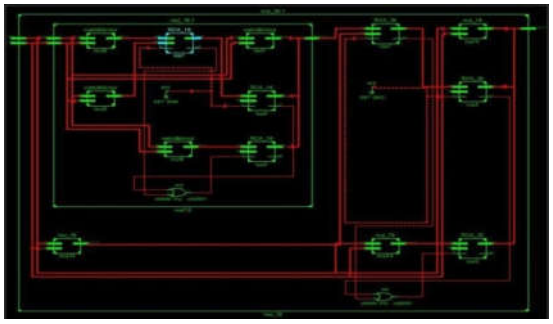
the thought of an adjusting bit has been introduced.

RESULTS

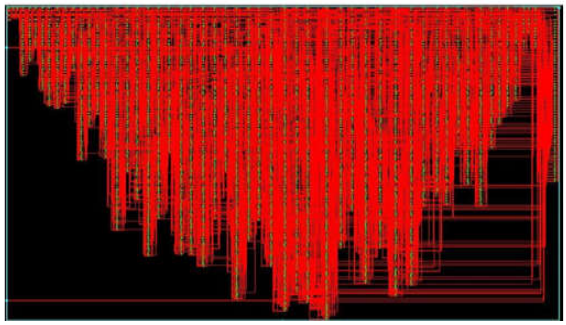
Simulation output



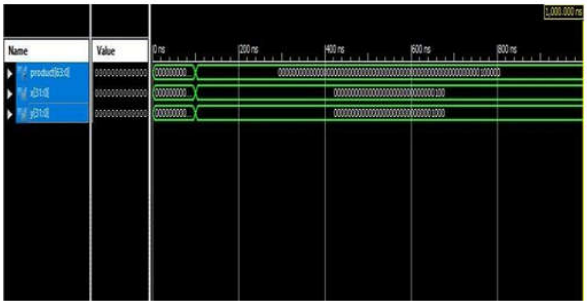
Top level Block



RTL Schematic



Technological schematic



CONCLUSION

Because of its central characteristics, it is quite likely that the plan only shows a modest increase in delay when scaled to larger bits. The parallelism connected to its incomplete item age is immediately apparent. Additionally, the fresh viper it fuses has a decrease in convey spread deferral. By introducing the notion of shifting bit reasoning into its engineering, the structure may combine transmit free math while eliminating radix change module speed overhead.

Due to increased parallelism and superior engineering nuances, the suggested structure showed an extension in execution territory over several plans. The suggested strategy prioritizes sophisticated frameworks with minimal idleness and high throughput at the price of zone overhead. For instance, in a DSP framework, activities, for example, Fast Fourier Transform, Convolution, Filtering and Discrete Wavelet changeand so forth. Multipliers assume a key job in deciding the speed of the framework. Essentially, this design would be a decent contender to be executed as a huge piece of frameworks like DCT, Central Processing Unit (CPU), MAC (Multiply and Accumulate) Unit, Image Processors where rapidaugmentations are basic to the presentation of the framework.
It can likewise be seen that regardless of the target of diminishing the deferral, the proposed plan performs superior to anything most structures thought about as far as power for lower info bit sizes [16 and 32 bit]. In spite of the fact that it expends more power than different structures higherinfo bit sizes [64 and 128 bit], it is legitimate when figured in with focal points picked up in speed for higher information bits.

REFERENCES

[1] M. Rabaey, A. Chandrakasan, and B. Nikolic, “Digital Integrated Circuits A Design Perspective,” PHI, 2003.
[2] B. Pahrami, “Computer Arithmetic and Hardware Design,” New York, Oxford University Press, 2000.
[3] M. Ercegovac, and T. Lang, “Digital Arithmetic, San Francisco, Morgan Kaufmann,” 2004.
[4] C S Wallace, “A Suggestion for a Fastmultiplier”, IEEE Transactions on Electronic Computers, Vol. EC-13, Issue 1, pp. 14-17,1964.
[5] K. Choi and M. Song, “Design of a high performance 32x32-bit multiplier with a novel sign select booth Encoder,” in IEEE International Symposium on Circuits and Systems, Volume 2, 2001, pp.701-704.
[6] J. Swami S. B. K.Tirthaji Maharaja, “Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda,” Delhi,1965.
[7] S. N. A and K. N, “Implementation of Power Efficient Vedic Multiplier,” International Journal of Computer Applications (0975 –8887), Vol. 43– No.16, 2012, pp.21-24.
[8] S. Vaidya and D. Dandekar, “Delay-Power Performance Comparison of Multipliers InVLSI Circuit Design,” InternationalJournal of Computer Networks & Communications (IJCNC), Vol.2, No.4,July2-010.
[9] H. Thapliyal and M. B. Srinivas, “High Speed Efficient N X N Bit Parallel Hierarchical Overlay Multiplier Architecture Based On Ancient Indian Vedic Mathematics”, Enformatika (Transactions on Engineering, Computing andTechnology),Vol. 2, Dec 2004, pp.225-228.