

Efficient Realization of Vinculum Vedic BCD Multipliers for High Speed Applications

G. Sreelakshmi¹, Kaleem Fatima², B. K. Madhavi³

¹Geethanjali College of Engineering and Technology, Hyderabad, India

²Muffakamjah College of Engineering and Technology, Hyderabad, India

³Sridevi Women's Engineering College, Hyderabad, India

Email: gantisiriphd@gmail.com, kaleemfatima@mjcet.ac.in, bkmadhavi2009@gmail.com

How to cite this paper: Sreelakshmi, G., Fatima, K. and Madhavi, B.K. (2018) Efficient Realization of Vinculum Vedic BCD Multipliers for High Speed Applications. *Circuits and Systems*, 9, 87-99. <https://doi.org/10.4236/cs.2018.96009>

Received: May 14, 2018

Accepted: June 26, 2018

Published: June 29, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Decimal multipliers play an important role in our day to day life for commercial, financial and tax applications. Every processor multiplier acts as the basic building block which decides the performance of processor. Time and again research is going on to design high-performance, low-latency BCD multiplier architectures. This paper proposes a new approach to BCD multiplication using vinculum number system. The key feature of the proposed architecture uses entirely a new one digit ROM based BCD multiplier that uses vinculum numbers as operands. Using this one digit BCD multiplier, an N digit BCD multiplier is built by using the vedic vertical cross wire method (Urdhav Triyagbhyam). We have also used our proposed multi operand VBCD Adder (Vinculum BCD Adder) [my paper 26] to add the partial products. In this paper, we show that this approach is a promising alternative to conventional BCD multiplication or other decimal multiplication methods that use alternative decimal representations like 5211, 4221, Xs3 etc.

Keywords

Signed Digit, Vedic Multiplier, Urdhav Triyagbhyam, Multi Operand Adder, VBCD Number System

1. Introduction

Designing of hardware units for decimal arithmetic is a growing interest among researchers to achieve better latency and throughput for highly complex, accurate fast computation required in business and commercial applications. The basic binary number system can be used for decimal arithmetic operations but it requires conversions at both ends. These conversions will take significant

amount of processing time which increases delay. Binary and Decimal number system supports integer and fractional parts in numbers and the system which uses fractional numbers may result in lack of accuracy which in turn has a greater –ve impact on commercial, financial and tax applications. To solve these problems, interest in hardware design of decimal arithmetic is growing. This has led to the incorporation of specifications of decimal arithmetic in the IEEE-754 2008 standard for floating-point arithmetic [1]. With high performance and low resource usage it is expected to facilitate the implementation of business applications [2] [3]. In DFP formats, multipliers play an important role in multiplication of mantissas. Among all arithmetic operations, multiplication is a complex operation. To speed up this operation, different methods were explained in the literature: Mixed Binary and BCD Approach [4], Multiplication via Carry Save Addition [5], Efficient Partial Product Generation [6], using Radix-10 multipliers [7] [8] [9], Parallel Decimal Multipliers [10] [11] [12] [13], Compressor Trees for partial product reduction [14], Multi operand Decimal Adders [15] [16], Redundant BCD and Signed Digit Adders [17] [18], High performance vedic decimal multiplier using binary to BCD converter [19], Vinculum BCD Multiplier (VBCD multiplier) [20] [21]. In our earlier approach in [20], we had proposed a vinculum BCD multiplier based on Ten's complement method. There also we used vertical cross wire method to generate partial products but the generated partial products were checked for +ve or –ve and if it is –ve, it was passed through Ten's complemented circuit and those products were given to the adder circuit.

In this paper, a vedic BCD multiplier based on vinculum number system is proposed. It uses the same vertical and cross wire method as used in [21] for generation of partial products. However, the number system used is different. The proposed system uses a unique number set $\{0, 1, 2, 3, 4, 5, \bar{4}, \bar{3}, \bar{2}, \bar{1}\}$, in which –ve numbers represented in two's complement forms [24]. The advantage of this code is same binary architectures can be used for designing. VBCD adders are used to add partial products. The correction logic is included in the adder block itself where the output of the VBCD adder is a valid vinculum output. This design is referred to as signed because it uses both +ve and –ve number digits.

The proposed VBCD multiplier is different from [20] in the following aspects.

1) Representation of vinculum number system in which each digit is represented using 4 bits. 2) Parallel VBCD adders and multi operand VBCD adders are used to add partial products. Our simulation results indicate that this approach is viable and efficient. The synthesis results show an improvement in speed.

The outline of the paper is organized as follows. In Section 2, reviews of decimal multiplication, vertical and cross wire method used in generation of partial products were discussed. The proposed methods of partial product addition using parallel adders and multi operand VBCD adders are discussed in section 3. In section 4 area and delay parameters are compared with other implementations found in the literature and finally conclusion and future scope are provided in Section 5.

2. Review of BCD Representations and Decimal Multiplication

Multiplication mainly consists of three stages: the generation of partial products, the fast addition (reduction) of partial products and the final carry propagate addition.

Vazquez and E. Antelo implemented a BCD multiplier using a recoding technique [7]. Signed-Digit (SD) Radix5 was employed to recode one of the input operands of the multiplier for the generation of the partial products. 6-Input LUTs and fast carry chains in Xilinx FPGAs were used to generate the building blocks and the decimal adders. Another SD-based decimal multiplier approach was proposed in [18]. The recoding was based on SD Radix10. BCD4221, 5211, and 5421 converters were used for the partial product generation. BCD4221-based compressors and adders were utilized in this approach. Although the BCD4221-based operations are similar to binary operation, the recoding and the different code conversions still lead to delay and resource cost.

1) SD Radix10 Recoding and Decimal multiplier based on BCD-4221/5221. In this method author assumed both multiplier and multiplicand to be unsigned BCD decimal integers of n digits each. The product $P = x * y$ is a non redundant BCD format with $2n$ digits. They used BCD4221/5211 in recoding [13].

2) Redundant BCD Representation and Decimal Partial Product Generation and Reduction:

In this method author used the same BCD 4221 and BCD5211 encodings to reduce partial product reduction [12] [13]. It is passed through a pre-computed correction, binary CSA tree structure, decimal sum correction blocks and 3:2 compressors to get final BCD 8421 corrected sum with $2d$ digits [14] [15].

3) Decimal Multiplier using Hybrid BCD codes: In this design author uses various types of BCD codes like 4221, 5211, ODDS, XS-3 and XS-6 codes [13] in which binary partial product reduction trees are non-fixed size.

The above method uses the weighted codes where conversions are required from one code to other code.

Vertical and Cross Wire Method (Urdhav Triyagbhyam)

This method is very simple and suitable for both binary and decimal number systems. It follows the principle of divide and conquer method where large module is divided into small modules of regular structures [20] [21]. This feature became an advantage in designing VLSI architectures. This method is very efficient for high speed applications [22]. **Figure 1** shows an example of two digit multiplication using Urdhav Triyagbhyam method.

3. Proposed VBCD Multipliers

3.1. Proposed Vinculum Number Representation

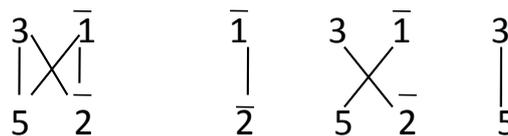
It's a Vedic mathematics of representing numbers. It allows only the digits

from 0 to 5 either in +ve form or in -ve form. The higher order numbers from 6 to 9 must be converted into its equivalent numbers. In our method we selected the two's complement representation to denote -ve numbers. Instead of 6,7,8,9 the equivalent less complex digits $\bar{4}, \bar{3}, \bar{2}, \bar{1}$ are included in the set of vinculum numbers. Therefore the new vinculum number system is $\{0, 1, 2, 3, 4, 5, \bar{4}, \bar{3}, \bar{2}, \bar{1}\}$. These digits are represented in binary using 4 bits each [24].

3.2. Generation of Partial Products

Single digit VBCD multiplier is developed using LUT where all partial products are saved in memory as shown in Figure 2. The maximum value of the partial product generated by single digit is +25 (5×5) in BCD the maximum value generated is 81 (9×9). Very less combinations are available in proposed number

$$3 \bar{1} * 5 \bar{2} = 1 4 1 \bar{2}$$



Step1: $\bar{1} \times \bar{2} = 0 2$

Step2 & 3: $3 \times \bar{2} + \bar{1} \times 5 + 0 = \bar{1} 1$

Step 4= $3 \times 5 = 1 5 + \bar{1} = 1 4$

Result: 1 4 1 2

Figure 1. Two Digit multiplication using vertical cross wire method.

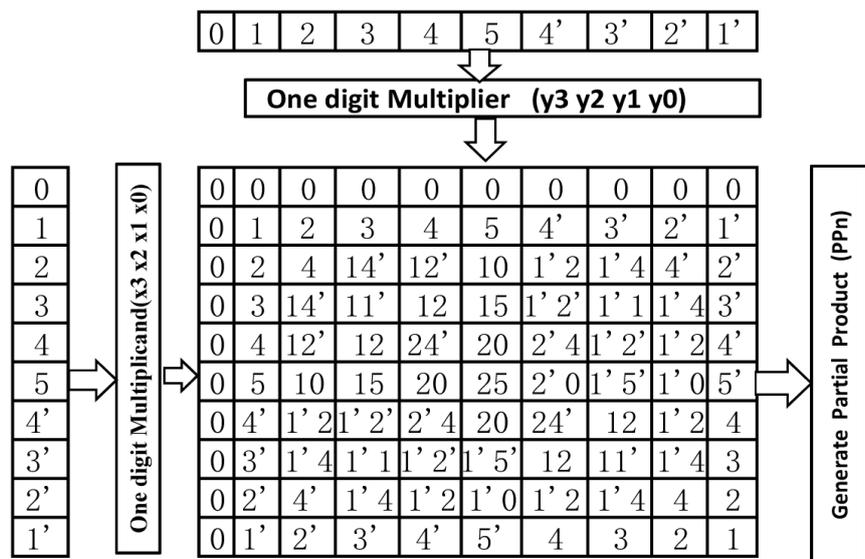


Figure 2. One digit VBCD multiplier.

system method which is simple and faster. This forms the basic multiplier for all other higher multipliers.

3.3. Two Digit VBCD Multiplier

Figure 3 shows an example of 2 digit Vedic multiplier using vertical cross wire method where it generates always only four partial products and these are added to get final product.

Figure 4 shows the pictorial representation of addition of partial products with their intermediate sum and carry bits of various levels to get their final product using VBCD parallel Adders [26].

Two Digit VBCD Multiplier Architecture

Figure 5 shows the basic 2×2 digit Vedic multiplier. Multiplier and Multiplier-cand are the two inputs to the system which produces four partial products. These four partial products are passed through parallel VBCD adder [26] for addition. The output of the adder is nothing but Final Result. The addition process is explained in next paragraph.

3.4. Example for 4 Digit Vedic VBCD Multiplier

The above figure shows an example of 4 digit BCD multiplication using vertical cross wire method only by divide and conquer method. In this each digit is subdivided into 2 digits and multiplication is performed as shown in above figure.

Figure 7 shows the pictorial representation of addition of partial products their intermediate sum and carry bits of various levels to get their final product.

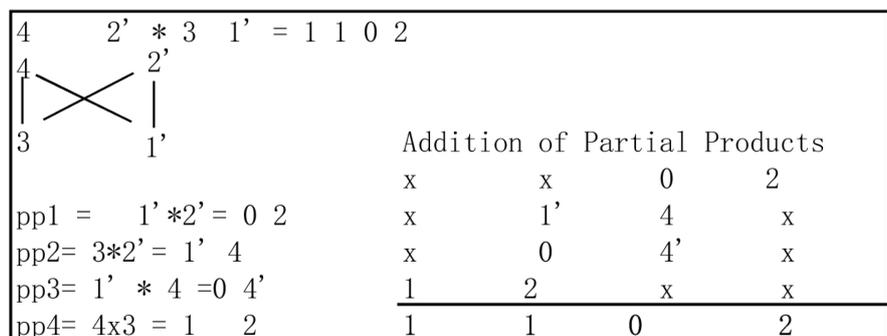


Figure 3. Example for 2 digit multiplication.

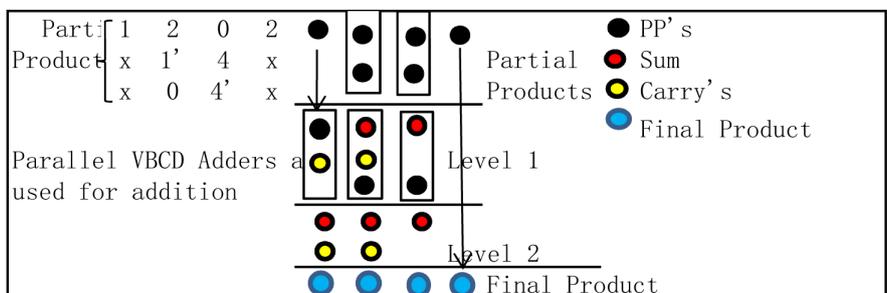


Figure 4. Addition of partial products using VBCD parallel adders.

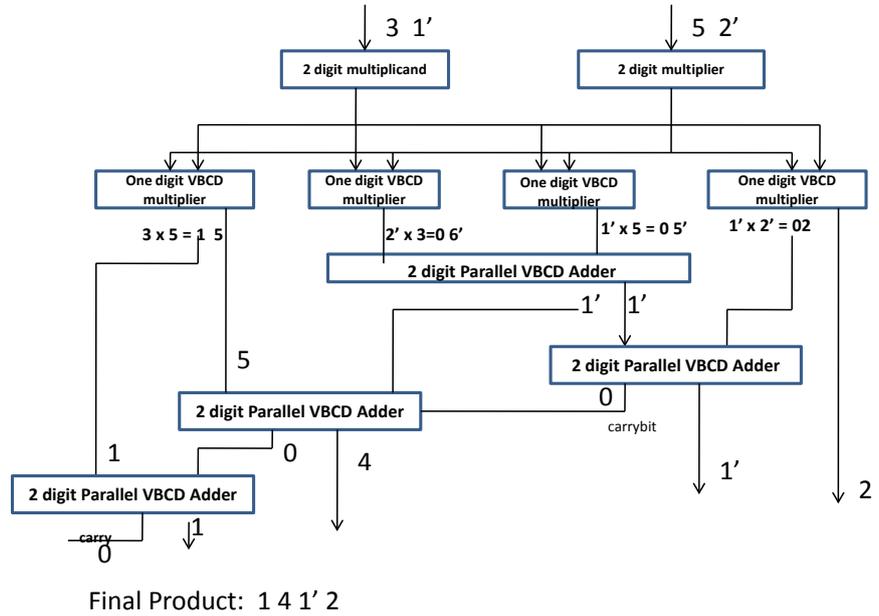


Figure 5. Two digit vedic VBCD multiplier.

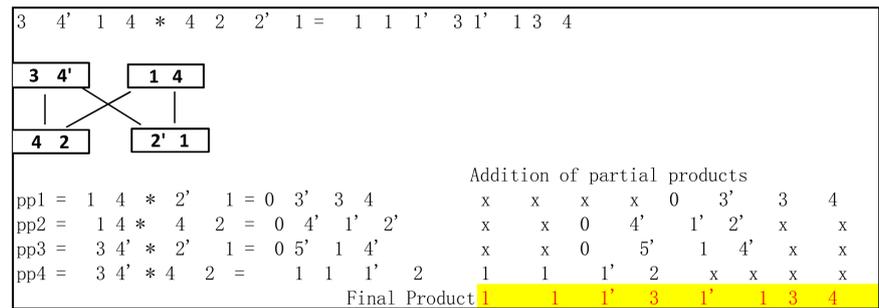


Figure 6. Example for 4 digit multiplication.

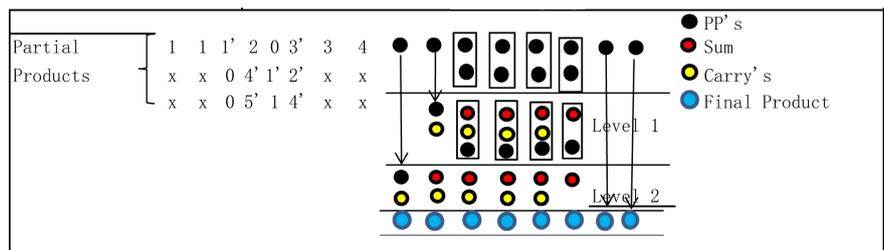


Figure 7. Addition of partial products using parallel adder.

It was observed that as the number of digits increases BCD adders increases but the number of levels or stages remains same because it generates only four partial products always.

Four Digit VBCD Multiplier Architecture

Figure 8 illustrates 4×4 multiplication using 2×2 digit Vedic multiplier (divide and conquer approach). Using this approach we will get only four partial product rows at any time there by addition becomes simple and faster. Four

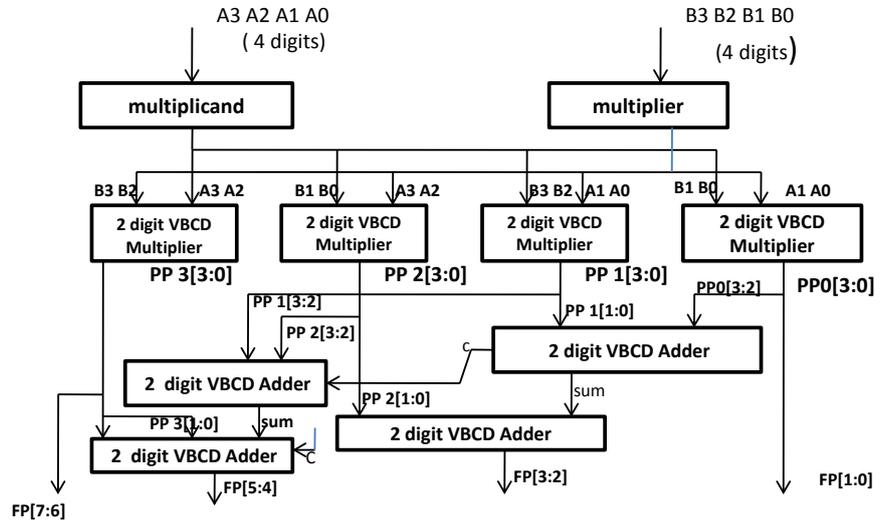


Figure 8. Four digit VBCD multiplier.

digit VBCD adders are used to add partial products and the output of the adder structure becomes Final product of 8 digits (32 bits).

3.5. Adder Structures for Adding Partial Products

Efficient adders were designed to add partial products for high performance and less delay. Literature gives various adder structures like simple simple Ripple carry adder, CLA, Carry Save Adders etc to complex prefix adder structures like Kogge stone, Brent kung adders etc, compressor logics (3:2 to 7:2 compressors), parallel Adders, Multi operand adders. Our proposed method uses 3 different methods to add partial products.

3.5.1. First Method (VBCD Parallel Adder)

In this method we used signed parallel adders to add partial products. The input to the adder may be +ve or -ve numbers which produces a valid vinculum sum. The advantage of signed digit adders are carry depends only on $i-1^{\text{th}}$ stage for i^{th} bit addition as shown in **Figure 9** which means only one bit delay exists. This concept was explained more clearly in refs [25] [26] and **Figure 10** shows an N-digit parallel adder in which $i + 1$ stage depends only on i^{th} stage output.

3.5.2. Second Method (Multi Operand VBCD Adders)

It uses Multi-operand signed digit adders to add partial products. Minimum depth of the adder is two which means we require two operands to add also known as parallel adder and maximum we went up to 8 operands as shown in **Figure 11** and **Figure 12** with 4, 8, 16, 32 bit operands. 4×4 digit multiplier uses 4 rows with 7 columns. The maximum depth for 4 digit multiplier is 5 along with previous carry bit. So we used 5:2 multi operand adder with 5 inputs and two outputs sum and carry. We observed that delay is reduced when compared to first method. **Figure 13** explains addition of partial products using multioper and adder concept for the example which is shown in **Figure 6**.

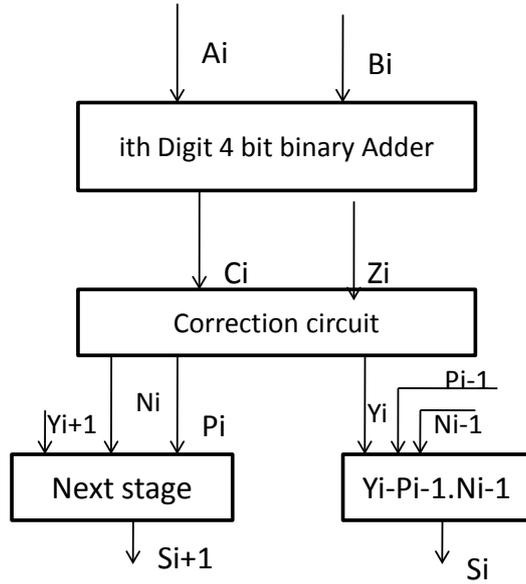


Figure 9. i^{th} digit decimal adder.

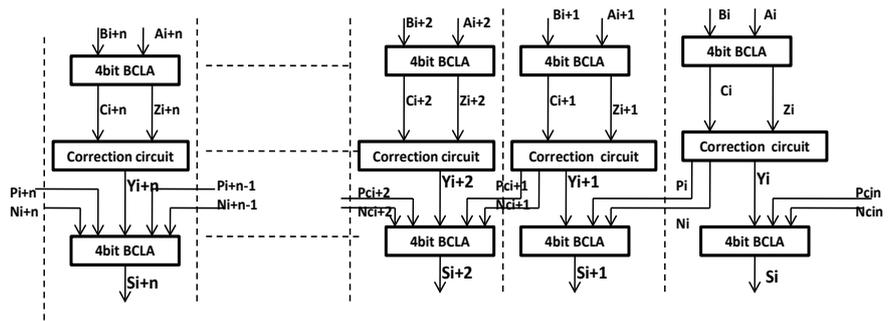


Figure 10. N-digit parallel adder.

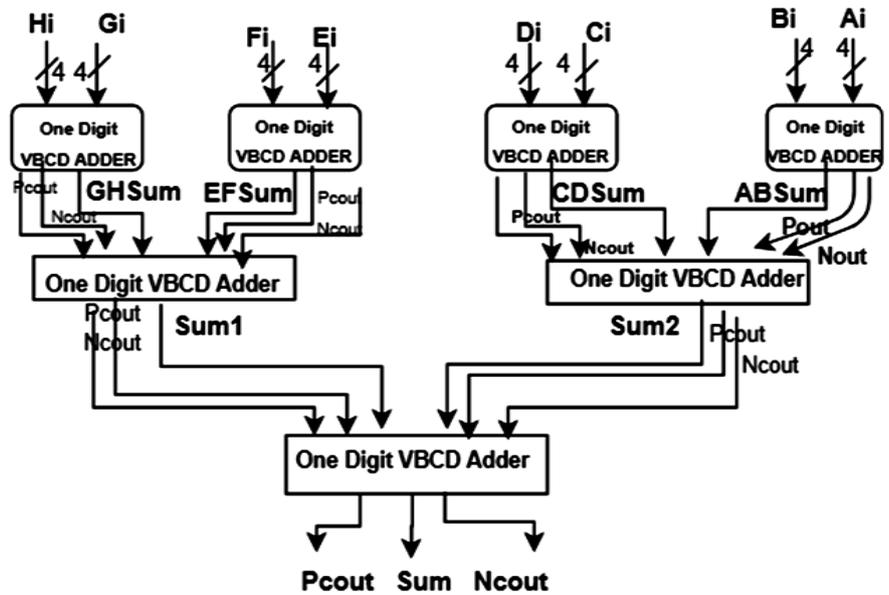


Figure 11. Multi operand adder.

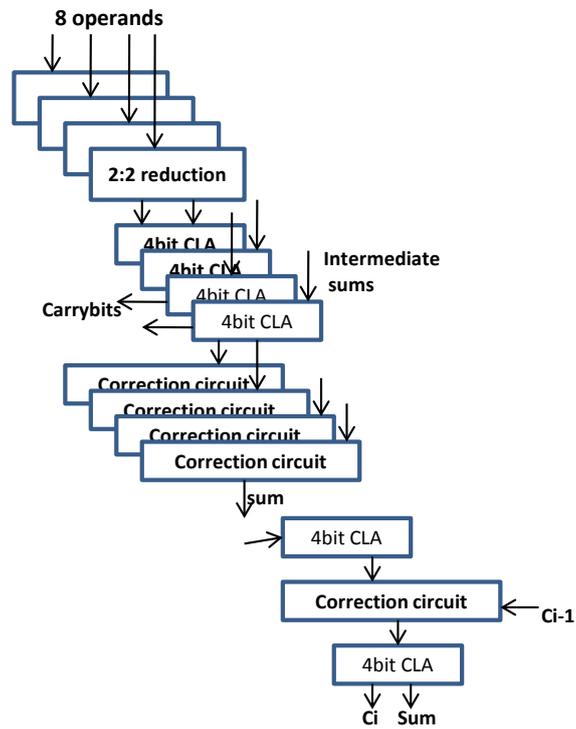


Figure 12. Multi operand adder with correction unit.

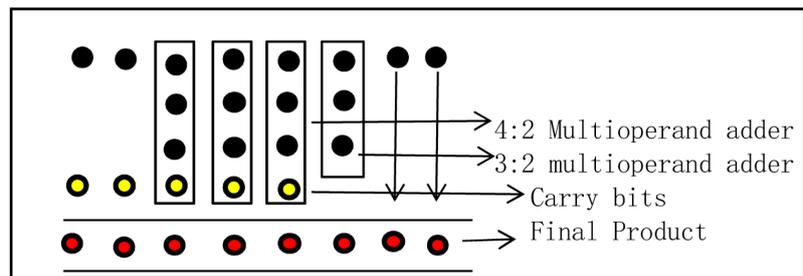


Figure 13. Addition of partial products using Multi Operand Adders.

a3 a2 a1 a0 * b3 b2 b1 b0								Multiplier & Multiplicand
				P03	P02	P01	P00	
			P13	P12	P11	P10		
		P23	P22	P21	P20			
	P33	P32	P31	P30				
	FP6	FP5	FP4	FP3	FP2	FP1	FP0 *	

•Indicates Final product bits
 •PP indicates Partial products

Figure 14. Partial product addition.

3.5.3. Third Method (Rearrangement of Partial Products)

In this method (refer Figure 15) instead of using partial products in conventional method (as shown in Figure 14) we rearranged partial products without changing its position value thereby we can use hardware efficiently. In this me-

thod we observed that in hardware the number of LUT's utilized are very less when compared to the above two methods.

In this paper we designed, implemented, simulated and synthesized 2×2 digit, 4×4 , 8×8 , 16×16 digit multipliers and these are compared with conventional multipliers. It is observed that in proposed method delay was significantly reduced with very little overhead in other parameters like area and power which can be used in high speed applications.

4. Results: Simulation Results for Multipliers

Using the proposed Adder structures in PPA block, the multipliers from 1 digit to 16 digit are evaluated and implemented in this section. The result of 2 digit multiplier is compared with few designs mentioned in the technical literature as shown in **Table 1**.

The decimal multiplier designs are described at gate level in verilog HDL,

a3 a2 a1 a0 * b3 b2 b1 b0								Multiplier & Multiplicand
	P33	P32	P31	P03	P02	P01	P00	} Partial products
		P23	P13	P12	P11	P10		
		-----	P22	P21	P20			
	-----	-----	-----	P30				
	FP6	FP5	FP4	FP3	FP2	FP1	FP0 *	

FP Indicates Final product bits
•PP indicates Partial products

Figure 15. Rearranged partial products for addition.

Table 1. Comparison with Different Multipliers of size 2 digit.

Sl. No.	Multiplier	LUT's	Slices	Delay (ns)
1	VBCD Proposed	210	108	11.42
2	[20] self Reference	430	242	12.74
3	[19] Mehta	922	585	77
4	[5] M. A. Erle	769	612	58
5	[23] M. P. Vestias	841	748	68
6	[24] G. Sutter	250	139	24.26

Table 2. Synthesis report for various size multipliers.

Sl. No.	No. of bits	LUT's	slices	Delay (ns)
1	4	27	--	1.811
2	8	210	108	11.417
3	16	951	248	16.690
4	32	3887	431	21.917
5	64	3180	319	24.440

Table 3. Synthesis report for 2 digit multiplier using different methods.

Sl. No.	Type	LUT's	Slices	Delay (ns)
1	Method 1	192	136	12.376
2	Method 2	210	108	11.417
3	Method 3	175	124	11.823

simulated and synthesized by Xilinx 14.2i simulator tool. **Table 2** shows the proposed Vedic VBCD Multipliers for various sizes and comparison table is shown for 8 bit multipliers.

Table 3 shows two digit multiplier using three different methods explained in above section and it as observed that Multi operand VBCD adders (method 2) is faster and method 3 occupies less area as shown in table.

5. Conclusion and Future Scope

Hence we conclude that in this paper we designed an efficient vinculum vedic BCD multiplier for faster operations. Performance of multipliers has been investigated and compared with other multipliers. These multipliers can be used in floating point multipliers for multiplication of mantissas. The proposed multiplier has very less delay and hence can be used for high speed applications.

References

- [1] Cowlshaw, M.F. (2003) Decimal Floating-Point: Algorithm for Computers. *Proceedings 2003 16th IEEE Symposium on Computer Arithmetic*, Santiago de Compostela, 15-18 June 2003, 104-111. <https://doi.org/10.1109/ARITH.2003.1207666>
- [2] IEEE Std 754(TM)-2008 (2008) IEEE Standard for Floating-Point Arithmetic. IEEE Computer Society, Washington, DC.
- [3] Aswal, M., Perumal, G. and Prasanna, G.N.S. (2012) On Basic Financial Decimal Operations on Binary Machines. *IEEE Transactions on Computers*, **61**, 1084-1096. <https://doi.org/10.1109/TC.2012.89>
- [4] Dadda, L. (2007) Multioperand Parallel Decimal Adder: A Mixed Binary and BCD Approach. *IEEE Transactions on Computers*, **56**, 1320-1328. <https://doi.org/10.1109/TC.2007.1067>
- [5] Erle, M.A. and Schulte, M.J. (2003) Decimal Multiplication via Carry-Save Addition. *Proceedings IEEE International Conference on Application-Specific Systems, Architectures, and Processors ASAP2003*, The Hague, 24-26 June 2003, 348-358. <https://doi.org/10.1109/ASAP.2003.1212858>
- [6] Erle, M.A., Schwarz, E.M. and Schulte, M.J. (2005) Decimal Multiplication with Efficient Partial Product Generation. *Proceedings of 17th IEEE Symposium on Computer Arithmetic*, Cape Cod, MA, 27-29 June 2005, 21-28. <https://doi.org/10.1109/ARITH.2005.15>
- [7] Vazquez, E.A. and Bruguera, J. (2014) Fast Radix-10 Multiplication Using Redundant BCD Codes. *IEEE Transactions on Computers*, **63**, 1902-1914. <https://doi.org/10.1109/TC.2014.2315626>
- [8] Lang, T. and Nannarelli, A. (2006) A Radix-10 Combinational Multiplier. 2006 *Forthieth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA,

- 29 October-1 November 2006, 313-317.
<https://doi.org/10.1109/ACSSC.2006.354758>
- [9] Dadda, L. and Nannarelli, A. (2008) A Variant of a Radix-10 Combinational Multiplier. 2008 *IEEE International Symposium on Circuits and Systems*, Seattle, WA, 18-21 May 2008, 3370-3373.
- [10] Gorgin, S. and Jaberipur, G. (2009) A Fully Redundant Decimal Adder and Its Application in Parallel Decimal Multipliers. *Microelectronics Journal*, **40**, 1471-1481.
<https://doi.org/10.1016/j.mejo.2009.07.002>
- [11] Jaberipur, G. and Kaivani, A. (2009) Improving the Speed of Parallel Decimal Multiplication. *IEEE Transactions on Computers*, **58**, 1539-1552.
<https://doi.org/10.1109/TC.2009.110>
- [12] Vazquez, E.A. and Montuschi, P. (2010) Improved Design of High-Performance Parallel Decimal Multipliers. *IEEE Transactions on Computers*, **59**, 679-693.
<https://doi.org/10.1109/TC.2009.167>
- [13] Cu, X., Lui, W., Dong, W. and Lombardi, F. (2016) A Parallel Decimal Multiplier Using Hybrid Binary Coded Decimal (BCD) Codes. 2016 *IEEE 23rd Symposium on Computer Arithmetic (ARITH)*, Santa Clara, CA, 10-13 July 2016, 150-155.
<https://doi.org/10.1109/ARITH.2016.8>
- [14] Castellanos, D. and Stine, J.E. (2008) Compressor Trees for Decimal Partial Product Reduction. *Proceedings of the 18th ACM Great Lakes Symposium on VLSI*, New York, NY, 4-6 May 2008, 107-110. <https://doi.org/10.1145/1366110.1366137>
- [15] Vazquez and Antelo, E. (2010) Multi-Operand Decimal Addition by Efficient Reuse of a Binary Carry-Save Adder Tree. 2010 *Conference Record of the 44th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 7-10 November 2010, 1685-1689.
- [16] Kenney, R.D. and Schulte, M.J. (2005) High-Speed Multi Operand Decimal Adders. *IEEE Transactions on Computers*, **54**, 953-963. <https://doi.org/10.1109/TC.2005.129>
- [17] Shirazi, D.Y., Yun, Y. and Zhang, C.N. (1989) RBCD: Redundant Binary Coded Decimal Adder. *IEE Proceedings—Computers and Digital Techniques*, **136**, 156-160.
<https://doi.org/10.1049/ip-e.1989.0021>
- [18] Svoboda (1969) Decimal Adder with Signed Digit Arithmetic. *IEEE Transactions on Computers*, **C-18**, 212-215.
- [19] Mehta, A.K., Gupta, M., Jain, V. and Kumar, S. (2013) High Performance Vedic BCD Multiplier and Modified Binary to BCD Converter. *Annual IEEE India Conference*, Mumbai, 13-15 December 2013, 1-6.
<https://doi.org/10.1109/INDCON.2013.6725995>
- [20] Sreelakshmi, G., Fatima, K. and Madhavi, B.K. (2016) Implementation of High Speed Vedic BCD Multiplier Using Vinculum Method.
- [21] Tirthaji, S.B.K. (1965) Vedic Mathematics. Motilal Banarsidass, Delhi.
- [22] Vestias, M.P. and Neto, H.C. (2010) Parallel Decimal Multipliers Using Binary Multipliers. *Proceedings IEEE VI Southern Programmable Logic Conference*, Ipojuca, 24-26 March 2010, 73-78. <https://doi.org/10.1109/SPL.2010.5483001>
- [23] Sutter, G., Todorovich, E., Bioul, G., Vazquez, M. and Deschamps, J.-P. (2009) FPGA Implementations of BCD Multipliers. *International Conference on Reconfigurable Computing and FPGAs*, Quintana Roo, 9-11 December 2009, 36-41.
- [24] Sreelakshmi, G., Fatima, K. and Madhavi, B.K. (2018) A Novel Approach to the Learning of Vinculum Numbers in Two's Complement Method for BCD Arithmetic Operations. *IEEE Conference ICCMC 2018*, IEEE Conference Record #42656.

- [25] Sreelakshmi, G., Ahmed, M.S., Fatima, K. and Madhavi, B.K. (2018) Efficient Signed Digit Decimal Adder. *IEEE Conference ICDCS 2018*, Coimbatore, 16-17 March 2018.
- [26] Sreelakshmi, G., Fatima, K. and Madhavi, B.K. (2018) Hybrid Signed Digit Parallel and Multi Operand BCD Adders.