DESIGNING OF DIGITAL CIRCUITS USING VEDIC MATHEMATICS FOR ENGINEERING APPLICATIONS

Ganti Sreelakshmi

Abstract

Digital architectures were proposed in this paper to improve the performance by mainly exploiting the properties of Vinculum number system and Vedic mathematics. These designs can be used in performing computational calculations like addition, subtraction and multiplication operations. In this paper we proposed a new approach in designing adders, subtractors and multiplier architectures. Analysis and comparison using Xilinx 14.2i was done and results shows that proposed digital architectures are faster when compared to conventional architectures found in the technical literature.

Key words: Vinculum number system, Vedic mathematics, Arithmetic operations

1. Introduction:

An Intelligent System is a combination of hardware and its software. Its performance is determined by the devices that are used in that system and the algorithms or software we are using. Same system gives better performance if we use latest software or by using efficient architectures. In digital circuits hardware plays an important role for performance parameters like Area, delay and power. Hence efficient digital architectures are required. As an example Arithmetic and Logical Unit of any processor requires hardware modules like arithmetic unit and logical unit. Arithmetic unit performs basic operations like addition, subtraction, multiplication, division operations and Logical unit consists of AND, OR, NOT, XOR operations. Therefore high performance system requires efficient Adders/Subtractors/Multipliers etc. Not only these hardware modules by using efficient algorithms we can improve the performance of system. Vedic Mathematics is one which permits to thinks in different ways for solving the problems easily and accurately. We took this and tried digital circuits for addition, subtraction and multiplication.

This paper deals with the hardware circuits built for addition, subtraction and multiplication for decimal number system.

The outline of the paper is arranged as follows. In section 2 representation of vinculum numbers in binary notation and in section 3 adders/ subtractors, parallel adders and multi operand adders along with examples were examined and in section 4 one digit, two digit, four digit, 8 digit multipliers using UrdhavTriyakbhyam method, explored along with examples and in section 5 synthesis results of digital architectures using Xilinx 14.2i EDA tools were discussed. Finally conclusions and future scope are provided in section 6.

2. Representation of decimal numbers in Vinculum form

World is turning around digital technology but there are few applications in our daily life which uses decimal number system like financial and commercial applications. This makes decimal number system very important for all of us. Smartness will come when we perform calculations and run applications without compromising accuracy within less time and with less hardware. A new approach to represent decimal number system was made by choosing vinculum numbers and writing its equivalent in binary form. This feature used to build digital architectures for various arithmetic operations.

Vinculum numbers are used to represent decimal number system between [+5 to -5]. In this set there won't be high complex numbers such as 6,7,8,9. Instead they are represented in to its equivalent form such as - 4,-3,-2,-1. Hence Vinculum number system consists of digits {+5, -4}. Vinculum number system consists of numerals 0,1,2,3,4,5 same as decimal number system and 6,7,8 and 9 are represented using negative numbers less than or equal to 5. Therefore Vinculum number system consists a set of numbers as {0, 1, 2, 3, 4, 5, −4, −3, −2, −1}. Hence complexity of higher order numerals like 6,7,8 and 9 are converted into less complex numbers. Vinculum is the Vedic method of representing decimal number system. Decimal numbers are represented in Binary Coded Decimal numbers for getting compatibility with Computer systems. Similarly we have used 2's complement number system for representing Vinculum numbers. This helped in representing signed Vinculum numbers. A unique set of tuples are represented in Vinculum number system which are suitable for any decimal arithmetic operation.

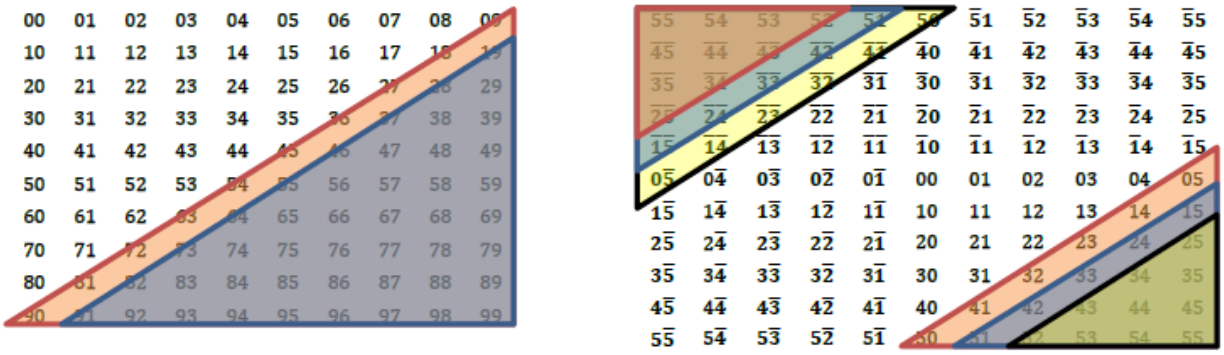

Figure 1: Comparision between BCD number system and VBCD number system

Table 1: Analysis of carry generations between BCD and VBCD number system

		BCD		VBCD	
Total number of combinations		100		121	
		Number	%	Number	%
with carry '0'	corrections required	45	45%	30	24.80%
with carry '1'		55	55%	31	25.60%
with carry '-1'		not applicable		31	25.60%
with carry '0'	No carry out	55	55%	91	75.20%
with carry '1'		45	45%	90	74.40%
with carry '-1'		not applicable		90	74.40%

From the table 1 we can observe that the percentage of corrections required and carry outs generated are very less in VBCD form when compared to BCD number system. This leads to the investigation of decimal architectures using vinculum number system.

Binary representation of Vinculum Numbers

Table 2. Vinculum numbers in binary form.

Vinculum Numbers	Binary number representation
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
$\bar{4}$	1100
$\bar{3}$	1101
$\bar{2}$	1110
$\bar{1}$	1111

From the table it was clear that for the digits 0 to 5 the binary form is equivalent to bcd 0 to 5 and –ve numbers are represented in 2's complement form.

3. Vinculum BCD Arithmetic Operations

3.1 VBCD Adders/Subtractors

Decimal Numbers	BCD Addition	Vinculum BCD Addition
1st no.→ 1 8 3 2nd no.→ 1 4 1 I sum → 3 2 4 3rd no.→ 1 2 9 F Sum → 4 5 3	0001 1000 0011 0001 0100 0001 0010 1100 0100 (Invalid) 0110 (Add 6 if each digit>9) 0011 0010 0100 (I sum) 0001 0010 1001 (129) 0100 0100 1101 (Invalid) 0110 (Add 6) 0100 0101 0011 (F sum)	2 $\bar{2}$ 3 0010 1110 0011 1 4 1 0001 0100 0001 3 2 4 0011 0010 0100 1 3 $\bar{1}$ 0001 0011 1111 4 5 3 0100 0101 0011 (453→ Final sum)

Figure 3: Example for BCD addition and VBCD addition

The above figure explains the addition between conventional and Vinculum addition. In BCD addition if digit is greater than 6 we must add 6 as shown in above example. In case of Vinculum representation it is a combination of +ve and –ve numbers so generation of carries and propagation are less when compared to the BCD form. In VBCD also correction factor is required if sum is greater than 5 or less than -5.

Decimal Numbers	BCD Subtraction	VBCD Subtraction
4 - 5→ -1	4+(2's complement of 5) =1 discard carry bit	4-(- 5)→4+5=9=1 $\bar{1}$
2 4 – 1 8→ 6	0010 0100 + 1110 1000(2's complement of 18) 0000 0110 = 06	2 4 – 1 8 = 2 4 – 2 $\bar{2}$→ 24+$\bar{2}$ 2 = 06

Figure 4: Example for Subtraction

From the above example we can conclude that subtraction can be performed by taking its 2's complement of –ve number and added to the first number as described in example. In VBCD subtraction if any input is -ve it will become +ve and +ve input will become –ve as shown

above.Hence same hardware can be used for subtraction with one extra input known as control input. If that control input is '0' it performs addition and if control signal is '1' it performs subtraction.

The proposed decimal adder utilises a signed 2's complement vinculum representation of the decimal numbers. The design although generates a dual carry, i.e. a positive and a negative carry, analysis of the adder has revealed a much lower probability of carry generation as compared to the conventional decimal adder allowing the possibility of parallel decimal addition.

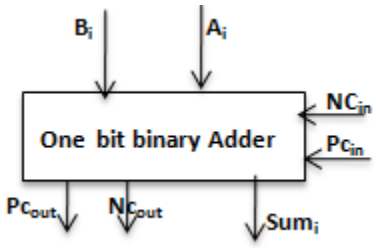

Figure 5: Digital circuit for one bit VBCD Adder

Table 3: Sum and Carry expressions for different cases of carry inputs

	PCin	NCin	Ai	Bi	Sumi	PCout	NCou	Expressions
Case 1	0	0	0	0	0	0	0	Sumi=Ai ⊕ Bi
			0	1	1	0	0	NCout=$\overline{Ai}$ $\overline{Bi}$ NCin
			1	0	1	0	0	PCout=(Ai.Bi)$\overline{Ncin}$+(Ai+Bi)Pcin
			1	1	0	1	0	
Case 2	0	1	0	0	1	0	1	Sumi=Ai ⊕ Bi⊕1=Ai Xnor Bi
			0	1	0	0	0	NCout=$\overline{Ai}$ $\overline{Bi}$ NCin
			1	0	0	0	0	PCout=(Ai.Bi)$\overline{Ncin}$+(Ai+Bi)Pcin
			1	1	1	0	0	Sumi=Ai⊕Bi ⊕1 = Ai Xnor Bi
Case 3	1	0	0	0	1	0	0	
			0	1	0	1	0	normal full adder
			1	0	0	1	0	
			1	1	1	1	0	
Case 4	1	1	x	x				Unused Condition

From table 3 it was observed that NC_{out} is logic high '1' when PC_{in}=0, NC_{in}=1 with inputs A_i& B_i=0 in case and in rest of all combinations it is '0'. Case 1 & 3 is similar to normal binary full adder. Hence the same hardware can be used for Vinculum architectures also. This is the main reason to build decimal architectures using Vedic mathematics.

Table 4 Carry outputs with PC_{in} & NC_{in}

PC_{in}	NC_{in}	Carry condition
0	0	No carry input
1	0	+ve carry input
0	1	-ve carry input
1	1	*Does not occur

The expressions for single digit VBCD adder for sum output, +ve carry output, PC_{out} and –ve carry output, Nc_{out} are given below.

Defining

$$P_i = A_i \oplus B_i \quad (1)$$

$$g_i = A_i\, B_i, \quad (2)$$

$$X_i = \overline{A_i}\,\overline{B_i} \quad (3)$$

and $$Y_i = A_i + B_i \quad (4)$$

then

$$Sum_i = (P_i\,\overline{Pc_{in}}\,\overline{Nc_{in}}) + \overline{P_i}\,(Pc_{in} + Nc_{in}) \quad (5)$$

$$Pc_{out} = (g_i.\overline{Nc_{in}}) + Y_i.\, Pc_{in} \quad (6)$$

$$Nc_{out} = X_i.Nc_{in} \quad (7)$$

Table 5 VBCD Adder outputs in terms of carry bits

PC_{in}	NC_{in}	Sum_i	PC_{out}	$*NC_{out}$
0	0	P_i	g_i	0
0	1	$\overline{P_i}$	g_i+Y_i	0
1	0	$\overline{P_i}$	0	X_i

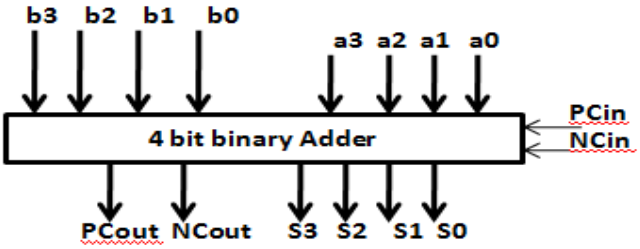

Figure 6 Proposed One digit adder

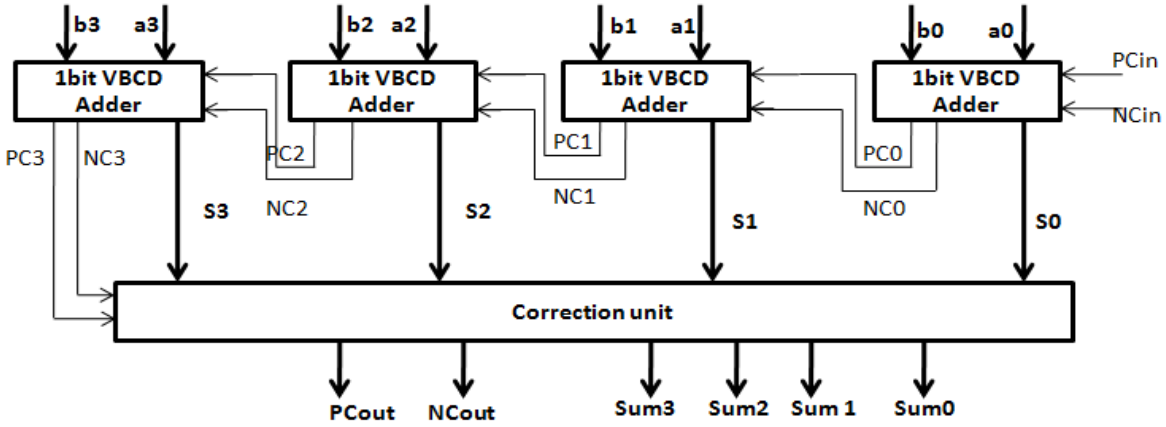

Figure 7 One digit VBCD adder using one bit elements

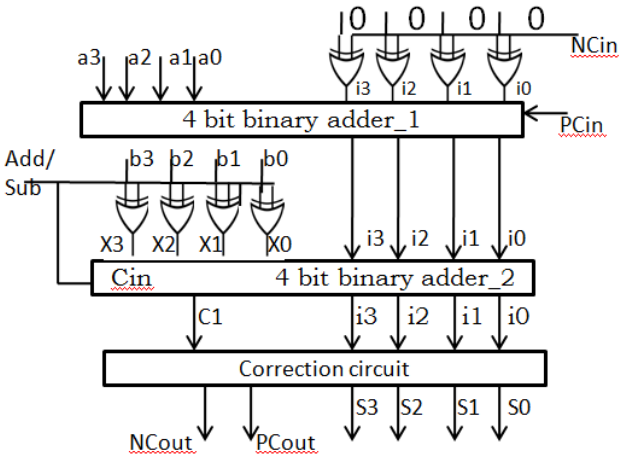

Figure 8 VBCD Adder/subtractor with Add/Sub as control signal

In the figure 8 when NCin=1 it passes through XOR gates and the output of gate is '1'. It gets added with input A and produces output as i. This output is added to input b and it is passed through correction circuit if sum>9.

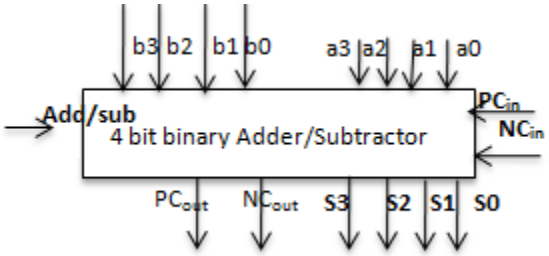

Figure 9 Block diagram of VBCD Subtractor

3.2 Parallel Adder and Multi-operand Adder:

Unlike in sequential adders in parallel adders we can add all digits at a time in parallel with out waiting for carry bits. In the example below Ai & Bi are inputs with intermediate sum Zi . It is passed through correction circuit and the output of correction circuit is Yi along with carry outs Pi & Ni. These are added to get valid final sum Si and Carry out PCout or NCout

Assume Augend: $2\ \bar{3}\ \ 4\ \overline{13}\ 1\ 3\ \bar{3}$ and Addend $2\ 3\ 3\ \bar{4}\bar{4}\ \ 3\ 2\ \ \bar{4}$

Augend (Ai)	$2\ \bar{3}\ \ 4\ \bar{1}\ \ \bar{3}\ 1\ 3\ \bar{3}$
Addend (Bi)	$2\ 3\ \ 3\ \bar{4}\ \bar{4}\ 3\ 2\ \bar{4}$
Intermediate Sum (Zi)	$4\ \ 0\ 7\ \ 5\ \bar{7}\ 4\ \ 5\ \bar{7}$
corrected Sum (Yi)	$4\ \ 0\ \ \bar{3}\ \ \ 5\ \ 3\ \ 4\ \ 5\ \ 3$
carry bits (Ni or Pi)	$0\ \ 1\ \ \bar{1}\ \ \bar{1}\ \ 0\ \ \ 0\ \ \bar{1}\ \ 0$
Final sum (Si)	$4\ \ 1\ \ \bar{4}\ \ 4\ \ 3\ \ 4\ \ 4\ \ \ 3$

Figure 10 Example for addition of two 8 digit numbers using Vinculum method

Special case: It was observed that in very few cases one more correction is required. Corrected sum (Yi) 5 and carry bit is +1 final sum bit is +6 which is not vinculum. So one more correction stage is required.

The general algorithm is as follows:

Algorithm:

Step1:(Z_i, C_i) = A_i+B_i // where Z_i=binary sum of the i^{th} digit and C_i is the carry out which may take the value C_i=0 or C_i=1. Where i is the i^{th} digit of the signed decimal digit.

Step2: Set correction.

$Y_i = Z_i + 6*C_i$

If Ci=0 and $6 \leq Zi \leq 10$, then (Y_i,P_i)= Z_i+6 where P_i= 1 termed as +ve carry C_i=1 and $-6 \leq Z_i \leq -11$, then (Y_i, N_i)=Z_i− 6 where N_i= 1 termed as − ve carry

Else

$Y_i = Z_i$

Step3: (S_i, P_i, N_i) =Yi+P_{i-1} − N_{i-1}. Where Si, is the final sum

1) For each digit, add the two operands to get the intermediate sum, Z_i. The range for Z_i is −9 to 11 inclusive of carry bit.
2) If Z_i> 5, set correction i =6 and C_i =1. If Z_i<−5, set correction i = -6 and C_i = −1.

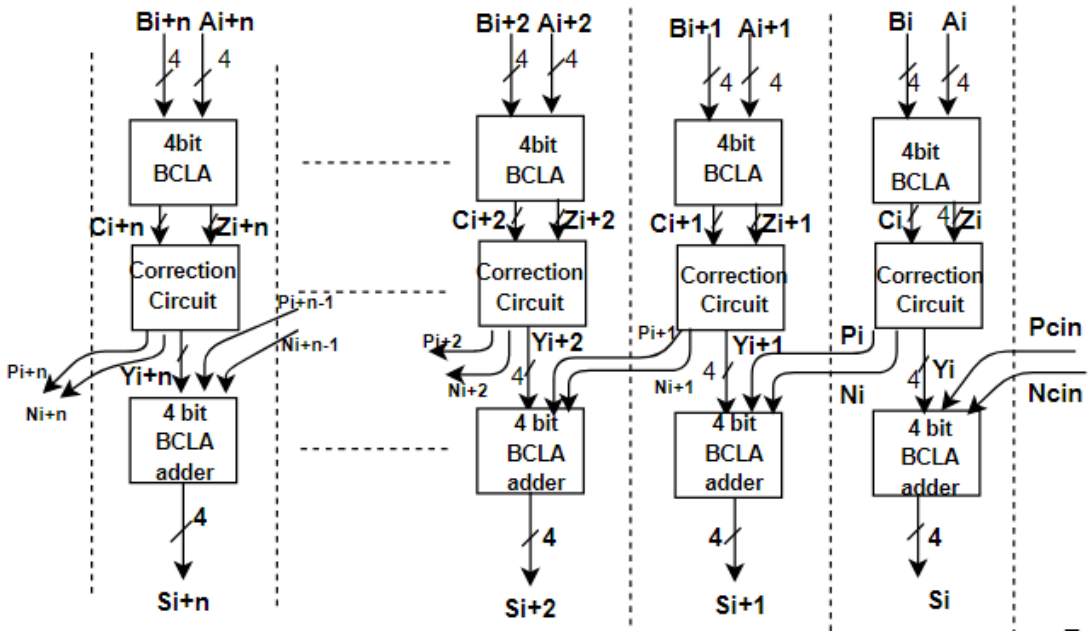

Figure 11 N digit parallel adder

3.3 Multi Operand Adders:

The operation for n-operand addition for a digit in the i^{th}position can be summarized in equations (8) - (10).

$Z_i = A_i + B_i$ and C_{out} (8)

$S_i = Z_i + 6 * C_i + C_{i-1}$ (9)

$$C_i = \begin{cases} +1 \text{ if } 6 \leq Z < 10 \text{ and } C_{out} = 0 \\ -1 \text{ if } -6 \leq Z < -11 \text{ and } C_{out} = 1 \\ 0 \text{ otherwise} \end{cases} \quad (10)$$

($a_{i,n}$ represents the n^{th} input for the i^{th} digit position). These shows that an intermediate sum can be calculated from multiple operands and a correction procedure must be used to get final sum.

For three operand addition, $-5 >= u_i - 6*ci >= 5$, for all possible u_i in {-15, 15} a C_i value can be found that makes $-5 <= U_i - 6*C_i <= 5$ true. Since we are representing multi operand using two operand adders intermediate sum at any stage will be in the range [-10, 10]. In our paper we proposed 8 operands with 8 digit each and went up to 32 bits. [publication 1]

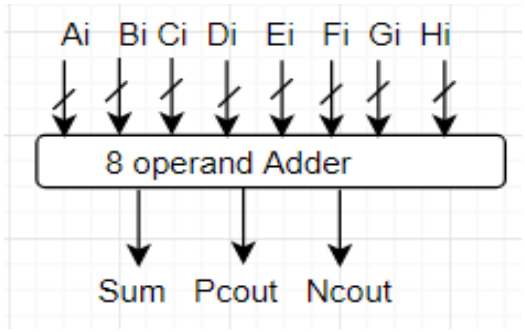

Figure 12: Multi-operand Adder

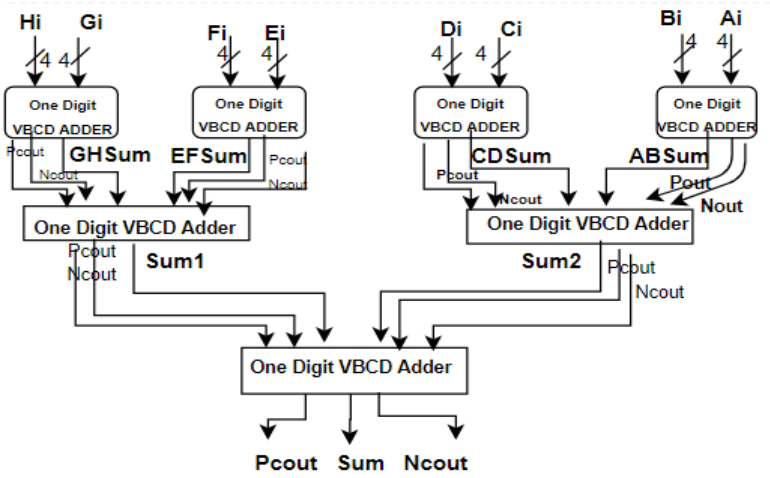

Figure 13: Multi _operand Adder using 2-operand Adders

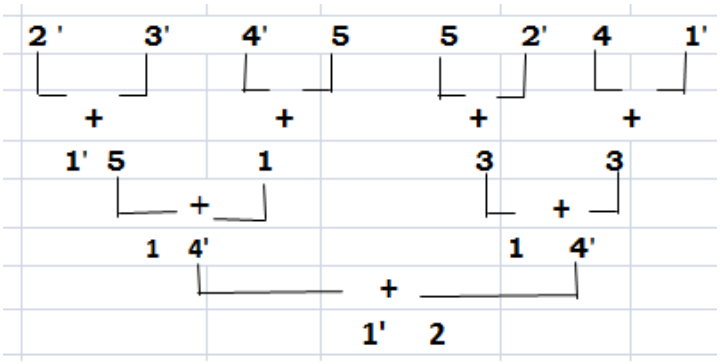

Figure14:Example for multioperand adder.

The above example performs addition on eight four bit operands and gives result in the form of sum and carry. In this 2 is sum and 1' as carry out in negative form. i,e –ve carry output.

4. BCD Multiplication:

Multiplier unit is very important module in an Arithmetic and Logic unit of a system and performance of any system can be measured by its hardware modules. So high speed units are required for better performance. Vedic Mathematics is one such approach where we can calculate in a simple and easiest method.

Out of 16 main sutras there are 3 sutras and 2 sub-sutras given for multiplication, as shown below

a) Urdhva-tiryagbhyam (Vertically and crosswise).
b) NikhilamNavatashcaramamDashatah (All from 9 and last from 10)
c) Anurupyena (Proportionality)
d) EkanyunenaPurvena (By one less than the previous one).
e) Antyayordasake'pi (Last totaling to ten)
f) Vinculum method (conversion of bigger numbers into smaller numbers)

Among all sutras of multiplication Urdhva-Tiryagbhyam is universally adopted method because it is suitable for both number systems binary and decimal.

The vinculum method is specifically used for BCD multiplication to reduce carry overs. This chapter deals with Vedic Multiplier using various adder structures like Ripple Carry Adder and Carry Look Ahead Adder with proposed adder structures like VBCD Adders, Parallel adders and Multi Operand Adders. All methods presented are implemented and compared against the proposed Multiplier.

4.1 Urdhva-Tiryagbhyam (Vertical and cross wire method)

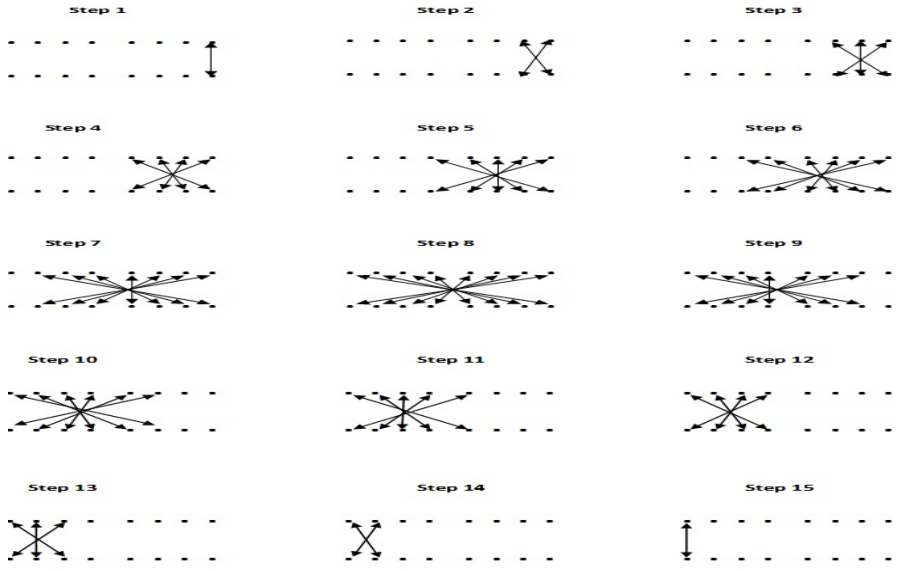

Figure 15 Line diagram of 8 bit Vedic multiplier

The above figure shows the line diagram of Vedic multiplier which holds good for both binary and decimal numbers. The main advantage of this Vedic multiplier is divide and conquers method. Any large digit 'N' can be divided into its basic building blocks of N/2 each. Example

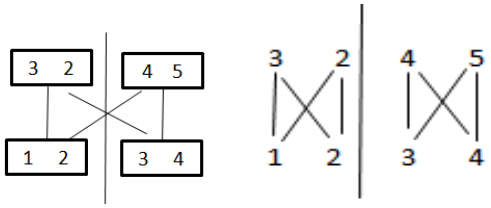

Figure 16 Four digit Vedic multiplier using divide and conquer method

Figure 16 shows an 4digit operand can be divided into two simple two digit operands for multiplication and internally two digits were divided into single digit.We can extend this technique to any number of digits with out affecting its other performance metrics.

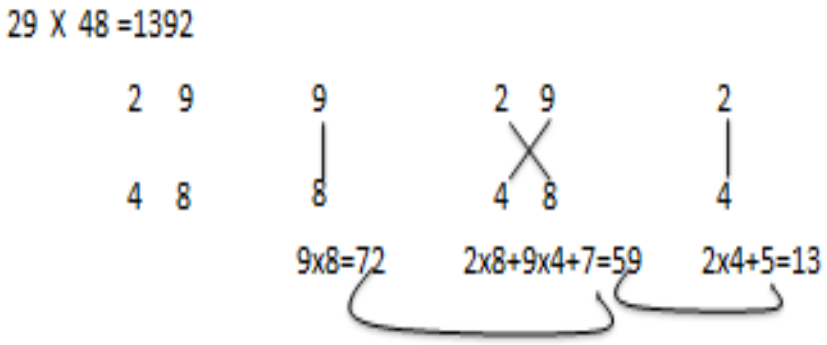

Figure17a) Example of Vedic multiplication using decimal numbers

Step1: 9x8=72

Step2: 2x8+9x4+7=59

Step3: 4x2+5=13 Result:1392

Red color indicates carry bit and it is forwarded to the next stage.

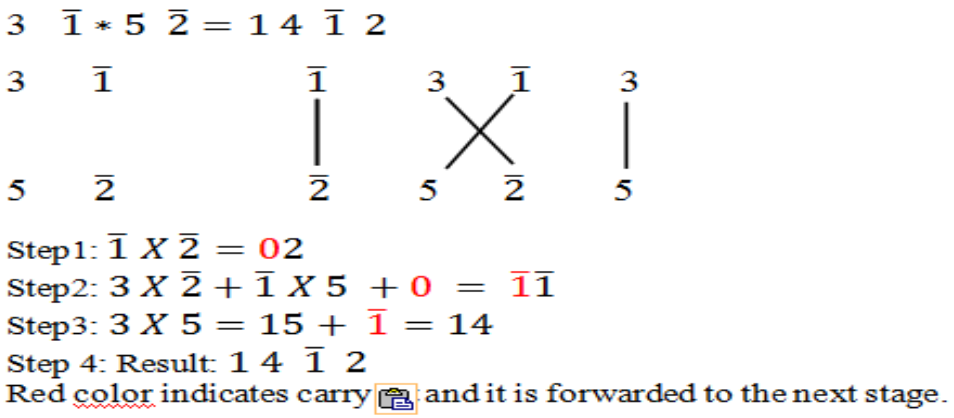

Figure 17 b) Example of Vedic multiplication using Vinculum numbers

4.1 One digit VBCD Multiplier:

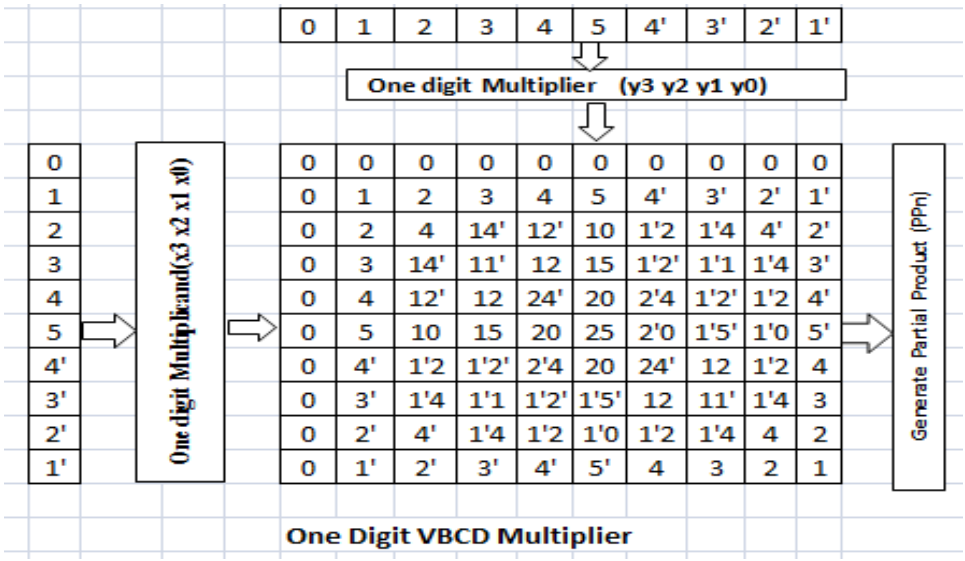

Figure 18 One digit VBCD Multiplier

The above figure shows one digit VBCD Multiplier using ROM based design. All partial products are stored in memory there by time taken to retrieve data is very less. Since it uses memory for all combinations a little overhead in area was observed.

4.2 Two digit VBCD Multiplier:

Figure below shows an example of two digit vinculum VBCD multiplication and its addition of partial products using VBCD parallel adders. The main feature of this multiplier is that it generates all partial products at a time and add those in parallel.

This feature improved delay of the device.

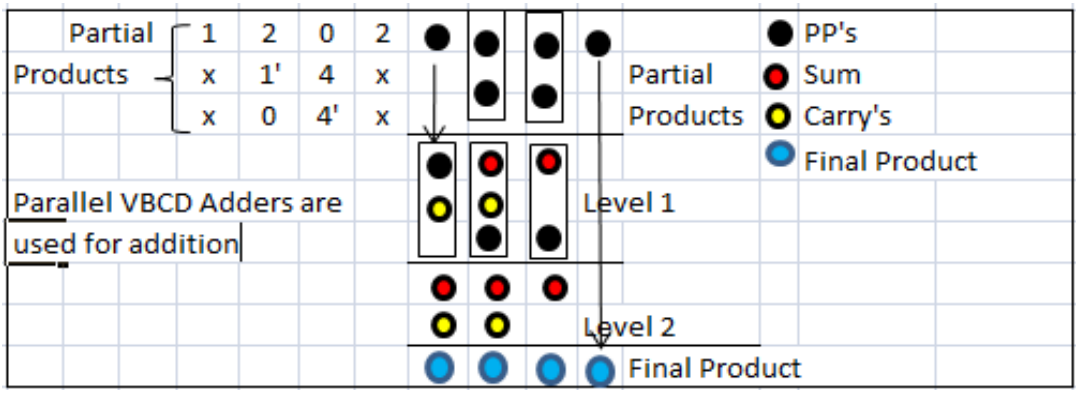

Figure 19 Addition of partial products using parallel VBCD adders

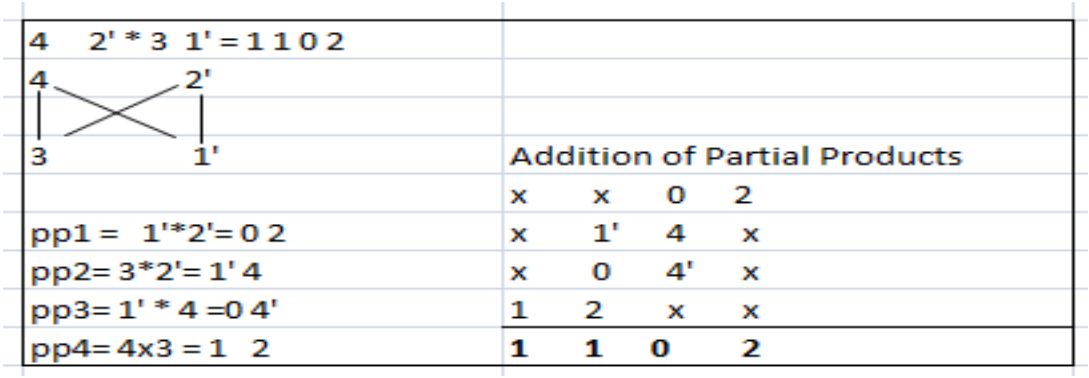

Figure 20 Arragement of partial products

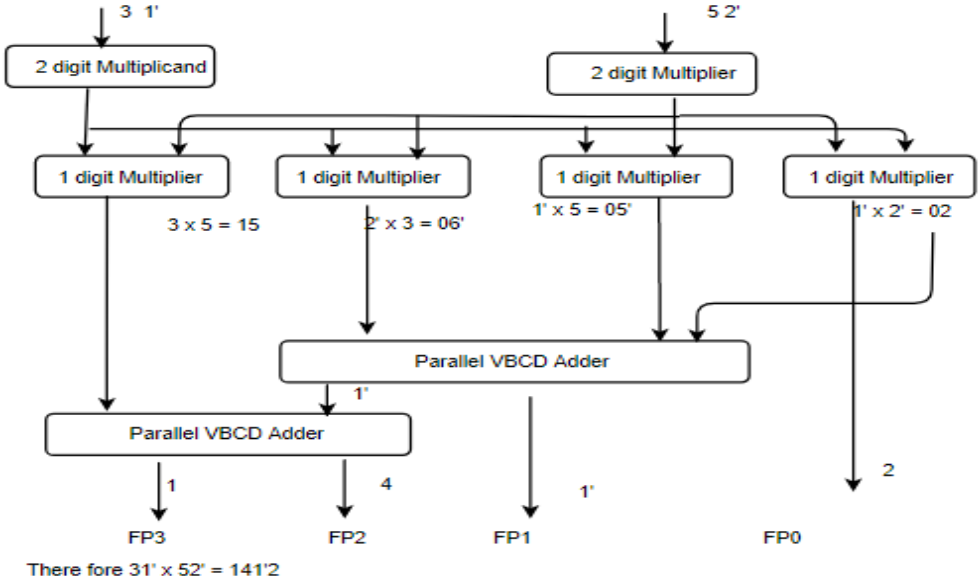

Figure 21 2x2 Digit Vedic VBCD Multiplier

In figure multiplicand and multiplier are 2 digit vinculum inputs which are passed through multiplier circuit it generates four partial products as shown in figure and these partial products are added through Parallel VBCD adder circuit. Parallel VBCD adder concept was explained in chapter[3] The output of the Adder circuit is final product which is a valid vinculum product.

4.3 Four digit VBCD Multiplier:

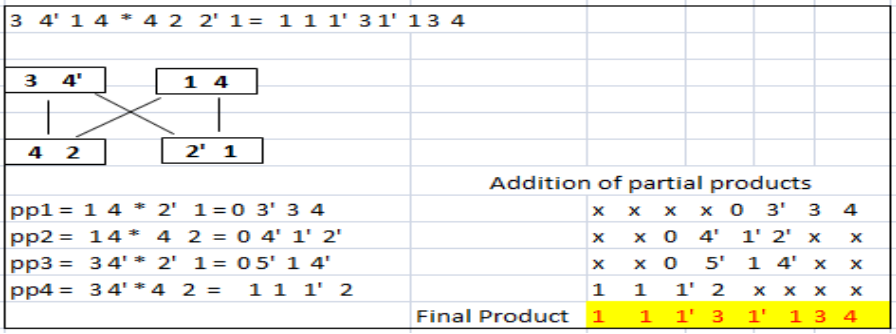

Figure 22 Example of 4 digit multiplication using vinculum numbers

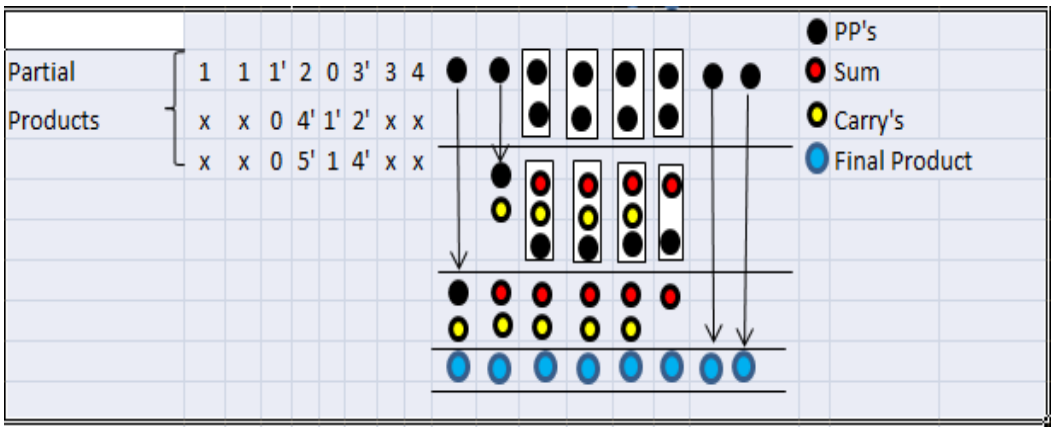

Figure 23 Structure for adding partial products using parallel adder

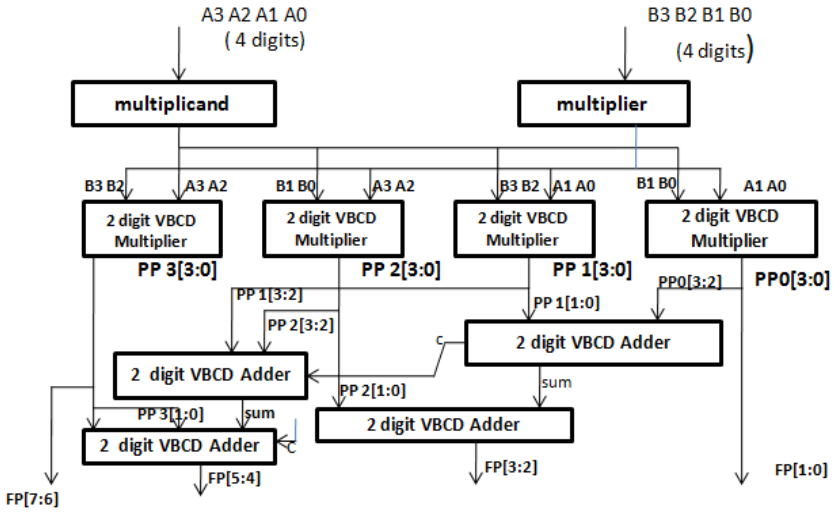

Figure 24 4x4 digit VBCD multiplier

Above figure shows architecture of 4x4 digit VBCD multiplier using 2digit multipliers. It generates four partial product rows each of size 16 bits each and these are added through four digit parallel VBCD Adders. The output of the adders is final product of size 8 digits as shown in figure.

4.4 Eight digit VBCD Multiplier:

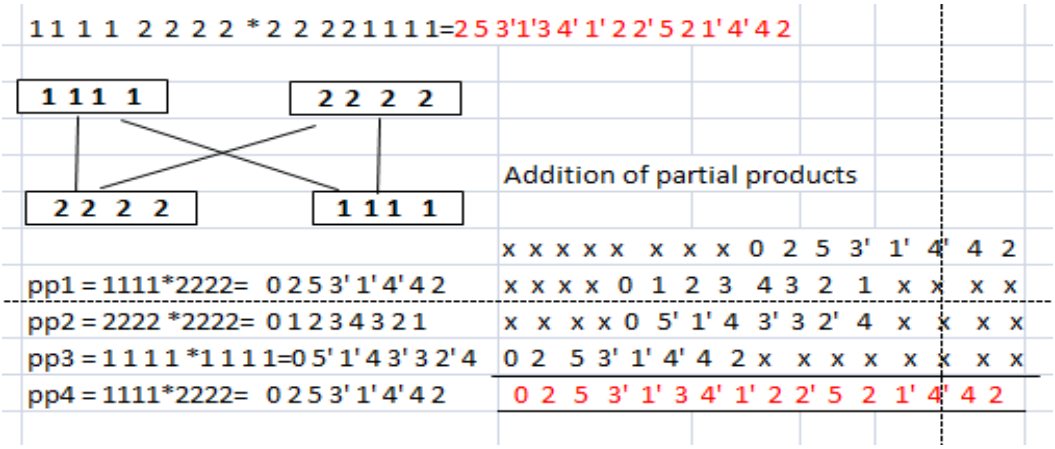

Figure 25 Example for 8 digit Vinculum VBCD multiplication

The above figure shows an example of 8 digit multiplication example using VBCD multiplication concept. Addition of partial products can be done using VBCD parallel adder or

Multi operand VBCD adders. Always in vertical and cross wire method number of partial products remain four as shown in above examples.

5. Simulation Results:

5.1 VBCD Adders:

Table 6 Comparison between conventional BCD and VinculumBCD Adders

No. of bits	Conventional BCD's		Vinculum Decimal Architectures	
	Delay(ns)	Cell Usage	Delay(ns)	Cell usage
4 bit BCD Adder	13.917	12	2.541	10
8 bit BCD Adder	15.618	28	4.372	35
16 bit BCD Adder	17.866	47	9.127	60
32 bit BCD Adder	34.981	116	14.618	138
64 bit BCD Adder	68.054	267	37.107	240

5.2 VBCD Adder/Subtractor

Table 7 Synthesis Results on BCD Subtractors

Type of Architecture	Conventional BCD's		Vinculum Decimal Architectures	
	Delay(ns)	Cell usage	Delay(ns)	Cell usage
4 bit BCD Subtractor	8.010	14	2.783	23
16 bit BCD Subtractor	19.425	55	9.127	64
16 bit BCD Subtractor CLA	9.390	88	8.528	77
32 bit BCD Subtractor(CLA)	14.625	270	13.697	176

5.3 Parallel VBCD Adders

Table 8 Synthesis Results for Parallel Adders

Sl.No.	No. of bits	LUT's	IOB's	Delay(ns)
1	4	15	14	3.432
2	8	32	28	4.037
3	16	67	52	5.352
4	32	132	112	6.432
5	64	268	224	8.595

5.4 Multi operand VBCD Adders

Table 9 Synthesis Results for Multi Operand Adder (4 bits)

Sl.No.	No. of operands	LUT's	IOB's	Delay(ns)
1	2	15	14	3.732
2	4	43	24	4.970
3	8	97	40	6.928

5.5 VBCD Multipliers

Table 10 Simulation results of proposed VBCD Multiplier

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

6 Conclusions and Future scope

Hence in this paper we proposed a new approach to decimal architectures using Vinculum number system which is a group of Vedic mathematics. Representing –ve numbers in its two's

complement form gave a major advantage in building hardware architectures and it has been observed that the performance parameters like Delay and Area improved a lot. The proposed architectures can be used to build various applications like MAC units, Floating point Architectures, Signal Processing applications.

References

[1] Sri BharatiKrisnaTirthaji, "Vedic Mathematics", published by MotilalBanarsidass, 1965. ISBN 81208-0163-6.

[2] Dr. Anil Kumar Teotia (2014) .Fundamentals and Applications of Vedic Mathematics.Published by: State Council of Educational Research & Training.

[3] HimanshuThapliyal, "Vedic Mathematics for Faster Mental Calculations and High Speed VLSI Arithmetic", Invited talk at IEEE Computer Society student Chapter, University of South Florida, Tampa, FL, Nov 14 2008.

[4] S. Hwang, "High Speed Binary and Decimal Arithmetic Unit," USA Patent No. 4,866,656.

[5] Svoboda, "Decimal Adder with Signed Digit Arithmetic," IEEE Transactions on Computers, Vol. C-18, No. 3, 1969, pp. 212-215. doi:10.1109/T-C.1969.222633

[6] Avizienis, "Signed Digit Number Representations for Fast Parallel Arithmetic," IRE Transactions on Electronic Computers, Vol. EC-10, No. 3, 1961, pp. 389-400. doi:10.1109/TEC.1961.5219227

[7] Vazquez, E. Antelo and P. Montuschi, "Improved Design of High-Performance Parallel Decimal Multipliers", IEEE Transactions on Computers, vol. 59, no. 5, pp. 679-693, May 2010.

[8] Vazquez and E. Antelo, "Multi-Operand Decimal Addition by Efficient Reuse of a Binary Carry-Save Adder Tree", in Proc. 44th ASILOMAR Conference on Signals, Systems and Computers, pp. 1685- 89, Nov. 2010.

[9] G.Sree Lakshmi, Dr.Kaleem Fatima, Dr.B.K.Madhavi "Efficient Realization of Vinculum Vedic BCD Multipliers for High Speed Applications" in Journal of Circuits and Systes,2018,vol 9, with ISSN 2153-1293 , DOI:10.4236/cs.2018.96009 June 2018.

[10] G.Sree Lakshmi, Dr.Kaleem Fatima, Dr.B.K.Madhavi "Hybrid Signed Digit Parallel and Multi Operand BCD Adders" ICIDEST 17th& 18th April 2018, KCG college of

Engineering and Technology, Chennai. (Conference is attached to SCI journals yet to be published)

[11] G.Sree Lakshmi, Mohammad Salman, Dr.Kaleem Fatima, Dr.B.K.Madhavi "Efficient Vedic Signed Digit Decimal Adder" Fourth International Conference on Devices, Circuits and Systems (ICDCS'18)ISBN: 978-1-5386-3476-9/18/$31.00 ©2018 IEEE 16th & 17th Mar 2018, Electronics and Communication Engineering Karunya Institute of Technology and Sciences

[12] G.Sree Lakshmi, Dr.Kaleem Fatima, Dr.B.K.Madhavi "A Novel Approach to The Learning Of Vinculum Numbers In Two's Compliment Method For BCD Arithmetic Operations"Proceedings of the Second International Conference on Computing Methodologies and Communication (ICCMC 2018)IEEE Conference Record # 42656; IEEE Xplore ISBN:978-1-5386-3452-3978-1-5386-3452-3/18/$31.00 ©2018 IEEE 475

[13] G.Sree Lakshmi, Dr.Kaleem Fatima, Dr.B.K.Madhavi"Implementation of High Speed Vedic BCD Multiplier using Vinculum Method" TENCON IEEE conference 22nd to 25th Nov 2016, Singapore, 978-1-5090-2597-8/16/$31.00 c 2016 IEEE Publisher: IEEE Digital Xplorer

Note: The figures that are used in paper are from my self-reference papers.