# High speed constant vector multiplication based on improved signed digit Approach

[*1]sunguluru Kiran Kumar

M.Tech Scholar, Department Of Ece, Siddharth Institute Of Engineering And Technology, Puttur

Email: kirankumar.yits@gmail.com

[*2]N. Vamsi Praveen

Associate Professor, Department of ECE, Siddharth Institute Of Engineering And Technology ,Puttur

Email-ID: vamsipraveen7@gmail.com

[*3]M.JanardhanaRaju

Professor, Department of ECE, Siddharth Institute Of Engineering And Technology ,Puttur

Email-ID: sietkece.f6@gmail.com

**ABSTRACT:** Real-time implementation of many digital signal processing (DSP) algorithms and multimedia applications are limited by the available speed, energy efficiency, and area requirement of multiplication. The multiplier-free implementation of the constant vector multiplication is reexamined. A novel improved signed digit representation technique is proposed to overcome the two main drawbacks of the current multiplier-free techniques: 1) computational redundancy and 2) circuit irregularity. The fundamental difference between the proposed technique and the existing multiplier-free techniques is a novel optimization framework based on vector decomposition. Experimental results demonstrate that the proposed technique outperforms the existing multiplier-free techniques in less operations and more regular circuit structure. In proposed system we use parallel prefix adder like Brent Kung adder (BKA) for optimizing the performance parameters like area and delay. The proposed multiplication scheme is coded in Verilog HDL and simulated using Xilinx ISE.

***Keywords*** **–** Canonical signed digit (CSD), Improved signed digit (CSD), Vector multiplication, Brent Kung Adder (BKA)

## I. INTRODUCTION

In digital computing systems, process controllers, and signal processors, Multiplication is the key operation. Each Multiplication involves 2 stages of operation, first one is generation of partial products and the second stage is accumulation of partial products. Therefore, the multiplication performance is enhanced by minimizing the partial products or enhancing their speed of accumulation by improved adders [1]. Besides, the performance of partial product accumulation depends on the enumeration system behavior to represent the operands. Although, in arithmetic units, a standard weighted binary number system of mathematical notation like two's complement representation is widely used. The obtained carry propagation chain within the standard number representation system will increase the delay of multiplication operation. Therefore the performance is reduced.

For typical binary mathematical representation, In n-bit fast adder CLI the input carry to position i, could be a perform of all inputs from zero to i, when ($i = 0, 1, \dots, n-1$). The dependency leads to increasing the multiplication delay that is dependent on width of the operands, within the worst case. Binary signed-digit range representation is AN acceptable numbering system to limit or eliminate the carry propagation chain in addition by eliminating the dependency between output and input carries. Numbers are represented as a string of bits in digital circuits, with the help of the logic symbols zeros and ones. Commonly the processed data is assumed to require values within the range of [–1, 1]. Approximate processing has been propelled by the extensive and developing class of applications that exhibit error flexibility, for example, DSP, interactive media (pictures /sound /video), illustrations, remote interchanges, and rising outstanding tasks at hand, for example, acknowledgment, mining, and synthesis[2],[3]. The requirements for a accurate calculation is leaved because of a few factors: (1) the restricted perceptual capacity of human (e.g., sound, video, designs because of the capacity of the human cerebrum to "fill in" lacking records and filters the

high-recurrence designs), (2) a remarkable end result is difficult to represent or does not exist (e.g., net are searching for, facts analytics), (3) users will acknowledge inexact but adequate outputs, and (4) boisterous inputs.

The class of techniques and techniques called surmised computing which commonly loosen up the necessity of correctness between the unique and execution of a processing framework. This rest permits exchanging the exactness of numerical outputs to reducing in location, postpone, and additionally power consumption of the structure [6].

Over the maximum recent many years, many designing techniques, circuits, and calculations have been supplied for inexact figuring. The person is alluded to, for itemized opinions on anticipated, stochastic, and probabilistic registering. The methodologies for the structure of hard DSP frameworks can for the most element be assembled in 3 lessons: (i) transistor level, (ii) gate level, and (iii) algorithmic measurement [5].

Multiplier fewer methodologies, as an instance, dispersed variety juggling or circulate/encompass utilizing CSD, can be utilized to execute advanced channels with consistent coefficients. despite the fact that, for channels with coefficients that are not understand from the earlier, for instance, versatile channels, fashionable multipliers are required, and they are often carried out utilizing motion and-include activities based double portrayal. canonical signed digit (CSD) illustration [11] and minimum signed digit (MSD) portrayal, the 2 of that have a base wide variety of nonzero digits in their portrayals, are a good deal of the time used to diminish device complexity in nature and increment throughput.

The improvement of quick CSD and MSD change calculations has been the focal point of much exertion. In 1951, Booth's recoding was exhibited to productively duplicate two numbers utilizing recoding multipliers. In 1960, Reit wiesner built up a calculation to change over two's supplement numbers to CSD [14].

Although all the current procedures can lessen the computational excess fundamentally, the consistency of the circuit might be sacrificed. For instance, in the insatiable based procedures, any vertices of the information stream can be utilized as a middle of the road result for the calculation of different vertices. The variations include ways for various sections influence the routability of the circuit and make the organized design style unmanageable.

In this paper, we recommend a novel progressed signed digit (ISD) approach for the consistent vector multiplication. the primary distinction among the proposed method and the traditional multiplier loose techniques is that the circuit normality is considered at the same time with the redundancy decrease process. we propose a singular advancement structure dependent on vector decomposition. to be specific, in the new shape, the regular vector is disintegrated into two terms: an open vector and a personal network. Alongside those parameters, the augmentation of the records flag and the constant vector is executed by a modern boom of the general population vector and personal framework. Humans in well known vector create quite a few ordinary center of the road vertices of the information move which may be reused for each passage. The normally talking computational excess may be dwindled via proscribing the length of people in popular vector and the quantity of responsibilities of the personal network. at that factor we suggest a voracious system to take care of the vector decomposition hassle.

Rest of the paper is organized as follows In chapter II We presented the work related to the proposed work. In chapter III presents the evaluation of the proposed approach and in chapter IV we portrayed results of the proposed work. Finally we conclude the paper in Chapter V.

## II. RELATED WORK

### *i) Number Representation*

The binary representation is the technique where we can decomposes a number in terms of set of additions by powers of two. The representation of numbers using a signed digit system makes the use of positive and negative digits. Accordingly, an integer number represented in the binary signed digit (BSD) [15], [16] framework utilizing n digits can be written as $\sum_{i=1}^{n-1} d_i 2^i$, where $d_i \in \{1,0,\bar{1}\}$and $\bar{1}$ denotes $-1$. The BSD framework is a repetitive number portrayal framework, e.g., both 0101 and $10\bar{1}\bar{1}$ compare to 5.

The canonical signed digit (CSD) representation, a subset of BSD, has a considered one of a type portrayal for every range and checks the accompanying essential residences: i) nonzero digits aren't close by; ii) the amount of nonzero digits is insignificant. Any n digit number in CSD has at maximum $(n + 1)/2$ nonzero digits, and via and massive, the quantity of nonzero digits is diminished via 33% while contrasted with paired [13]. This illustration is broadly applied inside the multiplier much less structure of regular multiplications because it decreases the hardware conditions because of the minimal range of nonzero digits.

For example, consider a constant number 25 which is defined in six bits. The representation of 25 in binary is, 011001 which include 3 nonzero digits. The same constant is represented as $100\bar{1}\bar{1}\bar{1}$ in CSD using 4 nonzero digits.

### *ii) Problem Definitions*

here, firstly we gift the issue of upgrading the amount of operations in a CMVM plan and later on, gift the issue of advancing the variety of sports beneath a postpone vital. Optimization of the Number of Operations Given an m × n constant matrix C with $cjk \in Z$ and an n × 1 variable vector X with $x_k \in Z$, the multiplication of C by X is a linear transformation from $Z^n$ to $Z^m$ and each linear transform can be computed as

$$y_j = \sum_{k=1}^{n} c_{jk}\, x_k$$

Where j and k ranges from 1 to m and 1 to n, respectively. Accordingly, the optimization problem of the range of operations in linear transforms may be calculated as follows:

**Lemma 1**: CMVM problem. Given $Y = \{y1, \ldots, ym\}$, a set of linear transforms, find the minimum number of addition and subtraction operations that generate the linear transforms.

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \begin{bmatrix} 3 & 11 \\ 5 & 13 \end{bmatrix} \cdot \begin{bmatrix} x1 \\ x2 \end{bmatrix}$$

$$y_1 = 3x_1 + 11x_2$$

$$y_2 = 5x_1 + 13x_2$$

(a)

$$y_1 = 3x_1 + 11x_2 = (11)_{bin}x_1 + (1011)_{bin}x_2$$

$$= x_1 + x_1 \ll 1 + x_2 + x_2 \ll 1 + x_2 \ll 3$$

$$= x_1 + 2x_1 + x_2 + 2x_2 + 8x_2$$

$$y_2 = 5x_1 + 13x_2 = (101)_{bin}x_1 + (1101)_{bin}x_2$$

$$= x_1 + x_1 \ll 2 + x_2 + x_2 \ll 2 + x_2 \ll 3$$

$$= x_1 + 4x_1 + x_2 + 4x_2 + 8x_2$$

(b)

Figure. 1. (a) A CMVM realizing $y_1 = 3x_1 + 11x_2$ and $y_2 = 5x_1 + 13x_2$; (b) Decomposition of the constants in the linear transforms in binary.

A straight route for the multiplier less acknowledgment of direct changes, for the most part known as the digit-based recoding strategy, is to characterize the constants in parallel, and for every 1 in the paired representation of the constant, is to move the variable and include the shift factors. As a basic precedent, consider the augmentation of a steady time by a variable vector given in Figure 1(a). The deteriorated types of direct changes are given in Figure 1(b).

## III. PERFORMANCE EVALUATION OF THE PROPOSED APPROACH

We use three experiments to assess the performance of the proposed ISD approach: a Gaussian clear out, stated filters and random filters with variant orders.

We pick a Gaussian filter with 11 orders for instance to show the circuit design system with the proposed ISD technique.

The filter coefficients are shown in below

[1 4 19 57 108 134 108 57 19 4 1]

Considering the symmetry of the filter coefficients, we implement as shown in Fig.2.

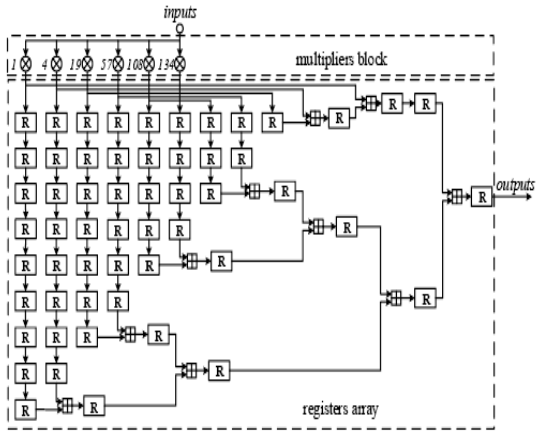



Figure .2 Gaussian filter implementation

In Figure.2, this usage incorporates two segments: the multipliers square and the registers cluster. The multipliers square fulfills and gives the duplications of each information and the gets or adjusts the out coefficients. The registers cluster gets the sifting impacts with the guide of including the duplication impacts after interesting postponements. It's miles basic that the greatest complex activity in this execution is to accomplish the duplications of the sources of info and 6 sift through coefficients

In order to get the CSD form multiplier-free implementation, the filter coefficients are represented [18],[19] as follows.

$$\begin{bmatrix} 1 \\ 4 \\ 19 \\ 57 \\ 108 \\ 134 \end{bmatrix} = \begin{bmatrix} 2^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2^0 & 0 & 0 & 0 & 0 & 0 \\ 2^0 & 2^0 & 0 & 0 & 2^0 & 0 & 0 & 0 \\ 2^0 & 0 & 0 & -2^0 & 0 & 0 & 2^0 & 0 \\ 0 & 0 & 2^0 & 2^0 & 0 & 2^0 & 2^0 & 0 \\ 0 & 2^0 & 2^0 & 0 & 0 & 0 & 0 & 2^0 \end{bmatrix} \times \begin{bmatrix} 2^0 \\ 2^1 \\ 2^2 \\ 2^3 \\ 2^4 \\ 2^5 \\ 2^6 \\ 2^7 \end{bmatrix}$$

According to above matrix, we can get the multipliers block hardware in CSD form as shown in Figure. 3.

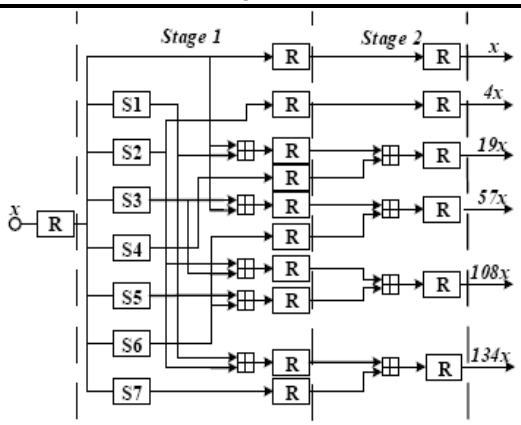



Figure.3 Multipliers block implementation in CSD form

In Fig.4, the implementation of the multipliers block needs 9 adders, and the maximum delay between every two registers is one adder.

With the proposed ISD approach, we can get below matrix and the implementation in Figure.4.

$$\begin{bmatrix}1\\4\\19\\57\\108\\134\end{bmatrix} = \begin{bmatrix}2^0 & 0 & 0\\ 2^2 & 0 & 0\\ 2^1 & 0 & 2^0\\ 0 & 2^3 & 2^0\\ 0 & 2^3 & 2^2\\ -2^1 & 0 & 2^3\end{bmatrix} \times \begin{bmatrix}2^0\\ 2^2+2^0\\ 2^4+2^0\end{bmatrix}$$

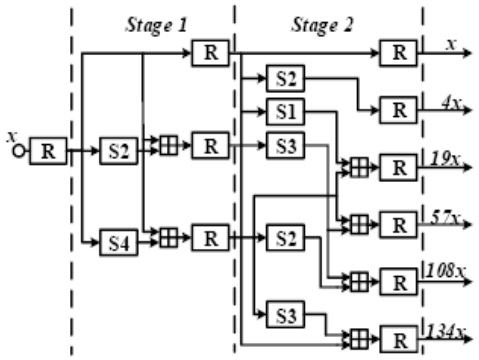



Figure.4 Multipliers block implementation with the ISD technique

In fig.4, this multipliers block implementation consists of levels. The first degree accomplishes the same sub-operators, $2^0x, (2^0 + 2^2)$x and $(2^0 + 2^4)$x. inside the 2nd level, the intermediate effects of stage 1 are moved and blanketed with some parallel adder systems. The forwarded bits inside the 2d level are selected by the coefficients of pm. actually the best put off between every registers in this execution is also one adder. But, the quantity of adders is dwindled to six.
The detailed numbers of operations in the two kinds of implementations are listed in Table 1.

Table 1: comparison of CSD and ISD operations

| Implementation | CSD form | Proposed ISD |
|---|---|---|
| Number of adders | 9 | 6 |
| Number of registers | 17 | 10 |
| Maximum delay between every two registers | One adder | One adder |

**Parallel prefix adder**
The PPA is kind of a carry look ahead adder. The creation of the suppliers the prefix adders can be structured in heaps of exceptional methodologies basically dependent on the uncommon necessities. We use tree structure shape to blast the speed of science activity. Parallel prefix adders are snappier adders and these are used for high overall performance arithmetic systems in industry.

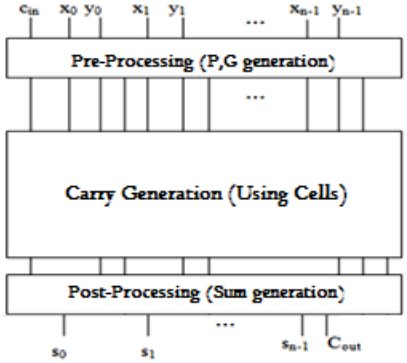



Figure.5 Structure of Parallel Prefix Adder

Fig. 5 demonstrates the based graph of a PPA. PPA might be partitioned into three essential segments, especially the Pre-handling, bring chart and set up-preparing. The Pre-handling segment will create the propagate (p) and generate (g) bits. Inside the second stage the procurement of the PPA conveys bit is separates PPA from various sorts of adders. it is a parallel type of procuring the convey bit that makes it plays expansion addition arithmetic faster and inside the specific last stage we can discover the whole of the operands through method for the utilization of convey from 2d level and proliferates shape first dimension.
There are various parallel prefix adders like Brent kung, Kogge stone, spreading over tree, etc. in this paper we use Brent Kung viper in the place of regular adders which makes the entire design it's most extreme quality in expressions of region and put off. The Brent kung adder comprises of 3 cells like black cell, buffer cell, grey cell inside the carry innovation stage.

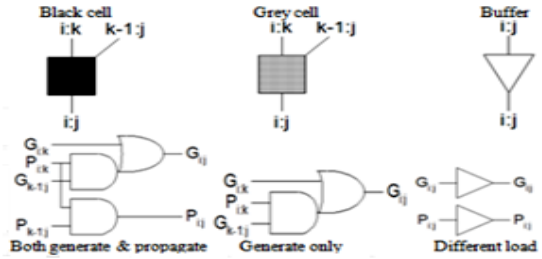



Figure .6 Black, Grey cells, Buffer and there Schematics.

**Brent Kung Adder structure**
Compared to previous adders the Brent Kung adder consists of less delay, so figure.7 Brent Kung adder is used in the proposed system.

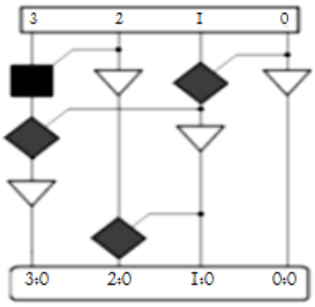
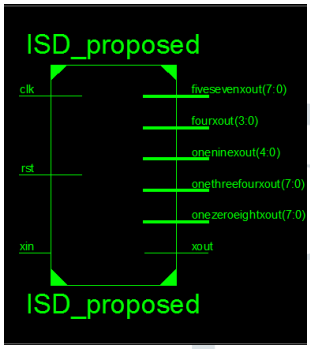


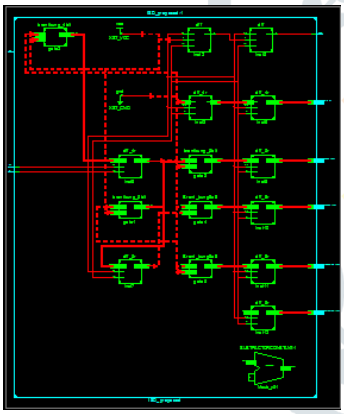

Figure.7 Brent Kung 4 bit adder

## IV. RESULTS

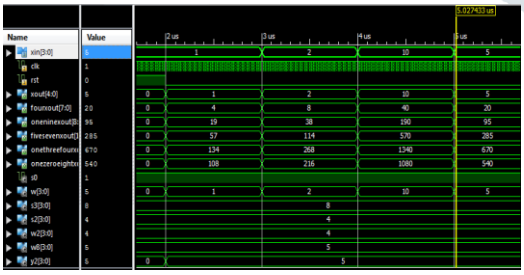
***Block diagram***




***RTL Schematic***



***Simulation result***



## V. CONCLUSION

We propose an improved signed digit (ISD) representation approach for constant vector multiplication. In the proposed ISD technique, the constant vector is decomposed into two terms: a private matrix and a public vector: the private matrix contains the private sub-operators for each entry and the public vector contains the same sub-operators for all the entries. Based on the decomposition results, the public vector generates a set of regular intermediate vertices of the data flow which can be reused for each entry. Therefore Brent Kung adder the area and delay will be reduced compared to existing systems. The propose ISD technique, which can improve the rout ability in the implementation and make it more suitable for the circuit design than the existing techniques.

## REFERENCES

[1] L. I. Bluestein, “A linear filtering approach to the computation of discrete Fourier transform”, *IEEE Transactions on Audio & Electroacoustics*, vol. 18, pp. 451-455, 1971.

[2] D. P. M. Netravali and A. N., "Reconstruction Filters In Computer-Graphics," *Siggraph Proceedings of Annual Conference on Computer Graphics & Interactive Techniqu*, vol. 22, pp. 221-228, 1988.

[3] I. Fogel and D. Sagi, “Gabor filters as texture discriminator”, *Biological Cybernetics*, vol. 61, pp. 103-113, 1989.

[4] W. A. Gardner, “Cyclic Wiener filtering: theory and method”, *Communications, IEEE Transactions on*, vol. 41, pp. 151-163, 1993.

[5] M. Unser, P. Thevenaz, and L. Yaroslavsky, "Convolution-based interpolation for fast, high-quality rotation of images," *IEEE Transactions on Image Processing*, vol. 4, pp. 1371-1381, 1995.

[6] G. Dimitrakopoulos and D. Nikolos, “High-speed parallel-prefix VLSI ling adders”, IEEE Trans. Comput., vol. 54, no. 2, Feb.2005, Pages. 225–231.

[7] Lagae, A. Lefebvre, S. Drettakis, G. Dutré, and P. &Dutré, “Procedural noise using sparse Gabor convolution”, *Acm Transactions on Graphics*, vol. 28, pp. 341-352, 2009.

[8] K. A. Syed, “The Discrete Cosine Transform (DCT): Theory and Application”, *Texts in Computer Science*, vol. 41, pp. 135-147, 2003.

[9] S. Chai Beng, R. Gani, and V. Jeoti, “Wavelet-DCT based image coder for video coding applications”, in *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*, 2007, pp. 748-752.

[10] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze, and M. Sadafale, "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, pp. 1029-1041, 2013.

[11] A. Avizienis, "Signed-Digit Number Representations For Fast Parallel Arithmetic", *Electronic Computers Ire Transactions on*, vol. ec-10, pp. 389-400, 1961.

[12] Hwang K, "Computer arithmetic-principles, architecture and design," New York: Wiley Press, 1979.

[13] I. S. E. Chen and T. N. Rajashekhara, "A fast multiplier design using signed-digit numbers and 3-valued logic", in *Circuits and Systems, 1990., Proceedings of the 33rd Midwest Symposium on*, 1990, pp. 881-884 vol.2.

[14] H. I. Jeon and M. A. G. Abushagur, "Digital optical modified signed-digit arithmetic based on symbolic substitution and its encoding scheme," in *System Theory, 1989. Proceedings., Twenty-First Southeastern Symposium on*, 1989, pp. 252-256.

[15] M. Saremi and S. Timarchi, "Efficient 1-out-of-3 Binary Signed-Digit multiplier for the moduli set {2n-1, 2n, 2n +1}", in *Computer Architecture and Digital Systems (CADS), 2013 17th CSI International Symposium on*, 2013, pp. 123-124.

[16] M. Joye and Y. Sung-Ming, "Optimal left-to-right binary signed-digit recoding," *Computers, IEEE Transactions on*, vol. 49, pp. 740-748, 2000.

[17] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients", *Circuits and Systems, IEEE Transactions on*, vol. 36, pp. 1044-1047, 1989.

[18] R. I. Hartley, "Optimization of canonic signed digit multipliers for filter design," in *Circuits and Systems, 1991., IEEE International Sympoisum on*, 1991, pp. 1992-1995.

[19] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers", *IEEE Trans.circuitsSyst.ii Analog Digit.signal Process*, vol. 43, pp. 677-688, 1996.