



Design of Truncated Multipliers for Precision Improvement and Its Application to a Filter Structure

V Bhatathi¹, Dr.P.Kalpna Devi², P.Vamsi Krishna³

¹Ph.D Research Scholar, Department of E.C.E, Veltech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology, Chennai

² Associate Professor, Department of E.C.E, Veltech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology, Chennai

³M. Tech PG Scholar, Department of E.C.E, VEMU Institute of Technology, P.Kothakota, AP
¹honey.bharathi76@gmail.com, ³pakalavamsi810@gmail.com,

Abstract: Truncated multipliers offers significant improvements in area, delay, and power. The proposed method finally reduces the number of full adders and half adders during the tree reduction. While using this proposed method experimentally, area can be saved. The output is in the form of LSB and MSB. Finally the LSB part is compressed by using operations such as deletion, reduction, truncation, rounding and final addition. In previous related papers, to reduce the truncation error by adding error compensation circuits. In this project truncation error is not more than 1 ulp (unit of least position). So there is no need of error compensation circuits, and the final output will be précised. To further extend the work the design is realized in a FIR filter.

Keywords: Computer arithmetic, faithful rounding, fixed- width multiplier, tree reduction, and truncated multiplier.

I. INTRODUCTION

Digital signal processing is one of most emerging and increasing popular research which frequently used in recent days. An efficient utilization of area, power consumption and speed of the multimedia applications are increased by the extensive use of FIR filters. To change the signal behaviors within a time interval and in a frequency domain, FIR filters are mainly used in signal processing. Hence, as a basic DSP element, it is recognized. In terms of commercial processors, DSP applications are getting more prominence. Other than conventional processors, DSP processors have more features and novel architectures. [1] To design the FIR filter-based processors, there are more algorithms are required due to the large demand for these unique features of DSP processors. DSP processing like filtering, convolution and inner products, multiplier and multiplier-accumulator (MAC) are the important elements. In MAC unit, the sum of products is calculated, whereas it is the heart of algorithms like FIR and FFT. To obtain high performance, the capability of MAC is playing a vital role in DSP.

To implement the Multiply-And-Accumulate (MAC) blocks that constitute the central piece in FIR filters and several functions, the design methods are mainly focused in multiplier-based architectures in DSP. For different DSP applications, FIR filters are very much important building blocks. It is essential to provide a high speed and higher order programmable FIR filters for



shaping, equalizing, adjusting and controlling signal frequencies in real time especially a big demand in video signal processing and transmission due to the emerging applications growth. FIR can able to do channel equalization and ghost cancellation. [2] Hence, an effective FIR filter design is required by an efficient VLSI architecture for emerging applications. By designing a direct form FIR filter, the efficiency can be increased. By Canonic Signed Digit (CSD) representation including Multiple Constant Multiplication (MSM), it can be obtained by reducing the number of adders and multipliers.[3] For improving the efficiency, FIR filter architectures can be reconfigured and it is suitable for any emerging applications. Reconfiguring FIR architecture is not time and cost effective and it suits only for certain kind applications. Hence, by modifying and extending the existing research works, this research work motivated to design a novel FIR architecture. CSD, MSM, Square Root Carry Select Adder (SQRT CSLA) and Improved Carry Save Adder (ICSA) are used after modification to do that.

a) Objectives

The key objective of this research work is to design and implement a novel architecture of a FIR filter in order to save the power, memory, delay, complexity and increase the throughput. To manage that, this chapter offers detailed information about FIR filter, applications, various designs and merits and demerits of FIR filter comparing with other filters. Also, this chapter provides the importance of the FIR filter in DSP applications. From this chapter it can be able to understand the concepts and functionalities of FIR filter.

b) Research Problem

It is noticed that FIR filter is the most important component in communication systems and digital signal processing including several portable applications from the above discussion. Also, by focusing on multipliers and adders configuration in FIR filter architectures, the efficiency can be increased. Using efficient multiplier and adder circuits for an optimized area, power, delay and increase in speed in digital signal processing (DSP), this problem is considered and this research work focused on designing a direct-form Finite Impulse Response (FIR) digital filter from the earlier research works, it is found that the performance of the FIR filter mainly depends on the multipliers used from the experimental results whereas by concentrating on the multipliers and adders the performance of the FIR filters can be improved. Hence, this research work is motivated for improving the efficiency which suits for any emerging DSP applications this research work is focused on the multipliers and adders involved in the FIR filter architecture.

II. LITERATURE SURVEY

A faithfully rounded truncated multiplier design is presented where the maximum absolute error is guaranteed to be not more than 1 unit of least position. In there proposed method, they jointly considers the delete non require bits, reduce the level, truncation, round up result using correction logic and final addition of partial product bits in order to minimize the number of full adders and



half adders during tree reduction. In this method efficiency of the proposed faithfully truncated multiplier with area saving rates of more than 30%. In addition, the truncated multiplier design also has smaller delay due to the smaller bit width in the final carry-propagate adder. The faithfully truncated multiplier has a total error of no more than 1 ulp and can be used in applications which need accurate result. By using this method we can be easily extended to signed or Booth multiplier design [1]. Low-cost finite impulse response (FIR) designs are presented using the concept of faithfully rounded truncated multipliers. They jointly consider the optimization of bit width and hardware resources without sacrificing the frequency response and output signal precision. Non uniform coefficient quantization with proper filter order is proposed to minimize total area cost. Multiple constant multiplications-accumulations in a direct FIR Structure is implemented using an improved version of truncated multipliers. Compare to other FIR design approaches show that the proposed designs achieve the best area, delay and power results [2].

III. SYSTEM DESIGN

PP (partial product) generation produces partial product bits from the multiplicand and multiplier. PP reduction is used to compress the partial product bits to two. Finally the partial products bits are added by using carry propagate addition. Two famous reduction methods are available,

1. Dadda tree
2. Wallace tree

a) Dadda tree

Dadda reduction performs the compression operation whenever it required. Wallace tree reduction always compresses the partial product bits. In the proposed method, uses RA reduction method. So that the final bit will be reduced. In the proposed truncated multiplier design, introduces column-by-column reduction. Here two reduction schemes are used, to minimize the half adders in each column because the full adder has high compression rate when compared to HA.

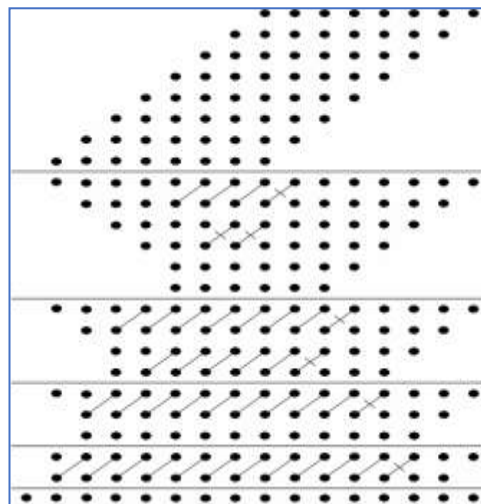


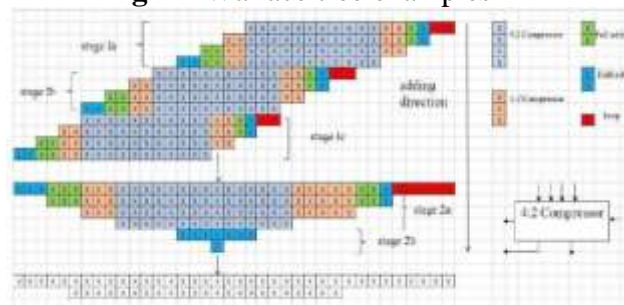


Fig – 1 Dot Diagram of an 8x8-bit Dadda Multiplier

b) Wallace tree

Wallace proposed an unusual way of parallel addition. Partial product addition of bits is performed using tree of carry save adder. This implies multiplication of two integers using Wallace method. Partial product matrix is reduced to a two-row matrix by using a carry save adder. The left two layers are added using a fast carry propagate adder to form the product. 3:2 compressors are used to propagate the conventional Wallace tree algorithm. The propagation of higher order compressor is minimized by using Wallace tree algorithm.

Fig – 2 Wallace tree examples



The conventional carry save adder is compared with kogge stone adder or KSA is implemented. Design I stipulate the multipliers with HDL – Hardware Description Language. This uses 8-bit unsigned data. It generates Power and speed output.

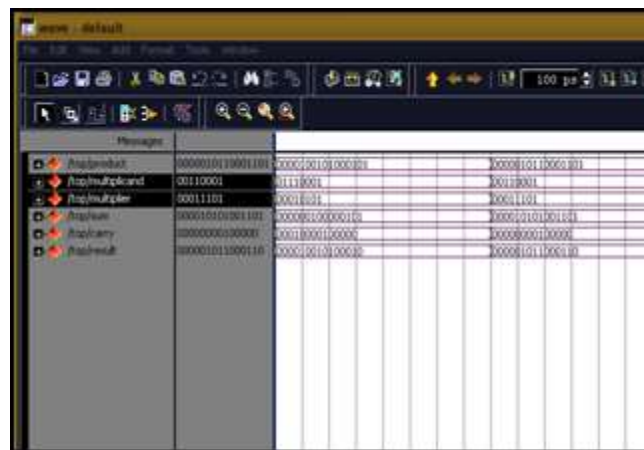


Fig-3 Reduced Wallace multiplier

a) Proposed Fractional Multiplier

In the Proposed precision truncated multiplier design the objective of a good multiplier is to provide a physically compact, good speed and low power consuming chip. To save significant power consumption of a VLSI design. In a truncated multiplier, several of the least significant



columns of bits in the partial product matrix are not formed. This reduces the area and power consumption of the multiplier. It also reduces the delay of the multiplier in many cases, because the carry propagate adder producing the product can be shorter. 3.1 Deletion, Reduction, and Truncation of partial product bits In the first step deletion operation is performed, that removes all the avoidable partial product bits which are shown by the light gray dots. In this deletion operation, delete as many partial product bits as possible. Deletion error E_D should be in the range $-1/2 \text{ ulp} \leq E_D \leq 0$. Hereafter, the injection correction bias constant of $1/4 \text{ ulp}$. Fractional Number Reduction Two fractional numbers with the size of $N \times N$ can be represented by [4].

$$Z = \sum_{k=1}^{2N} z_k 2^{-k} = X \cdot Y = \sum_{l=1}^N x_l 2^{-l} \cdot \sum_{m=1}^N y_m 2^{-m} \quad (1)$$

Where k is the weight of the bit

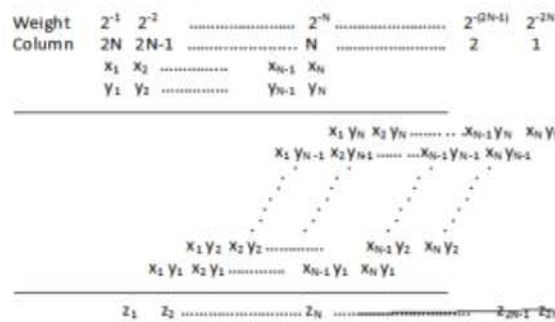


Fig-4 Multiplication of Fractional number

In the final result, the N most significant bits : z_1 to z_N are kept for further calculations. The rest of the bits are not accounted. The maximum truncation error for the above cancellation should be 1ulp , Where $1 \text{ulp} = 2^{-N}$. The total error of the rounded truncated multiplier is represented by equation where E_{total} is the total error of the multiplier truncation and rounding error respectively. Deletion, Reduction and Truncation Deletion is performed in the first stage, where the unnecessary PP bits are removed by deletion as depicted by grey dots in Figure for an example 8×8 fractional multiplication in which the last 8 bits are truncated. The first two rows are not considered for the deletion process but they participate in the final truncation and rounding steps. Deletion of partial products starts from column 3 and continues until the weight reaches 2^{-N-1} as shown figure The deletion error range can be represented by

$$-\frac{1}{4} \text{ulp} \leq E_D \leq \frac{1}{4} \text{ulp}$$

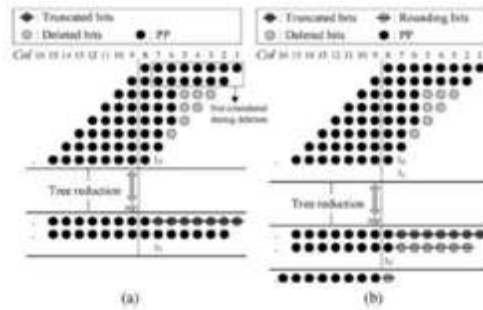


Fig-5 Deletion and Truncation Scheme for Fractional Multiplier

Rounding and Final addition the final product of N bits are generated by adding the PP bits using CPA after the deletion, reduction and truncation, which is shown in figure then, the bits in column 2 to N-1 in second row are removed during the rounding process which is shown as crossed spotted dots in [1].The total error of multiplier should not be more than 1 ulp, so as to achieve faithful deletion, truncation and rounding.

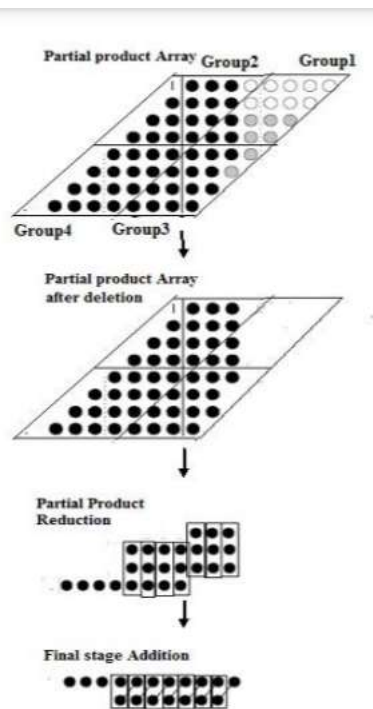


Fig-5 Fractional Reduction of 8bit

b) Checking the outputs Example:

For 8x8 fixed number multiplication, the inputs are 11111111 and 11111111, then the output is 1111111000000001. For 8x8 fractional multiplier, the partial products are deleted till the $E_{total-D}$ is less than 2^{-N-1} , where $N=8$ and $E_{total-D}$ is total deletion error [1]. Suppose the inputs of 8x8

multiplier are 1111.1111 and 1111.1111, then the output is 1111110.011 in which the last three bits are removed during rounding and thus the output becomes 1111110. For 16x16 fixed number multiplication, the inputs are 11111111111111 and 11111111111111, then the output is 1111111111110000000000000001. In 16x16 fractional multiplier, the partial products are deleted till the $E_{total}-D$ is less than $2-N-1$, where $N=16$. In 16x16 fractional multiplier, the inputs are 1111111.111111 and 1111111.111111 then the output is 11111111111110.11110 in which last five bits are removed during rounding and thus the output becomes 11111111111111.

IV. SIMULATIONS AND RESULTS

In this project three different Multiplier Algorithms are compared

1. Truncated Multiplier (Proposed Multiplier)
2. Wallace Multiplier (Fixed Number Multiplier)
3. Dadda Multiplier (Fixed Number Multiplier)



Fig-6 Truncated Multiplier Output Waveforms

In Truncated Multiplier the input is given as $a=15.9375$ $b=11.9375$ the output for truncated is the decimal Part =190 and its fractional part is truncated. According to the Truncated multiplier algorithm .

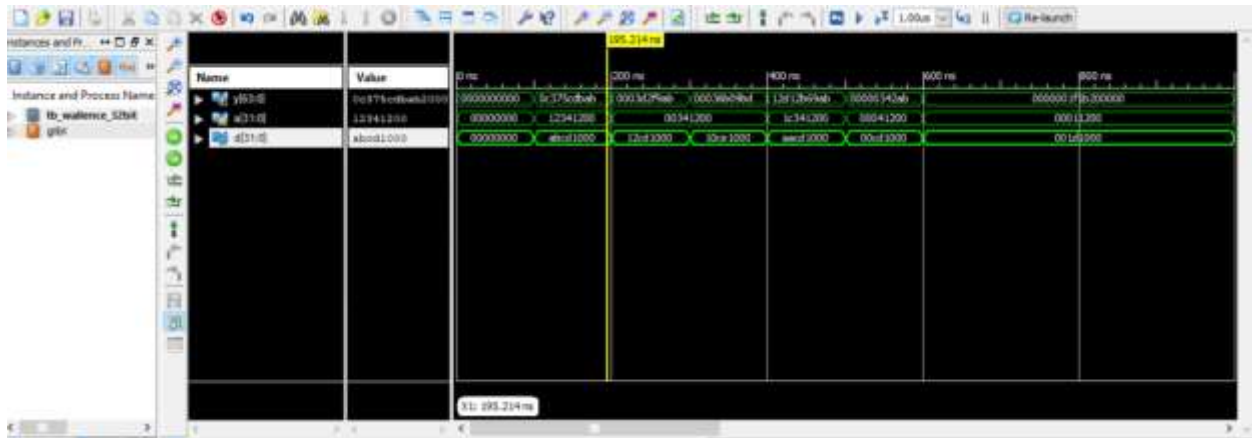


Fig-7 Wallace Tree Multiplier Output Waveforms



Fig-8 Dadda Tree Multiplier Output Waveforms

Wallence and Dadda Tree multiplier input is fixed number (not Fractional) Accordingly we get the 32bit multiplier output

If $a = 266752$ $b = 13438976$ output = 3,584,873,725,952

Table-1 Power and Delay Area Comparison Results

Algorithm	Power	Delay	Area (LUTs)
Wallence	2.42 mw	34.409 mw	1595
Dadda	3.57	30.41	2182
Truncated	4.10	27.103	2157

IV. CONCLUSION



There are many works proposed to reduce the truncation error by adding error compensation circuits so as to produce a précised output. This approach jointly considers the tree reduction, truncation, and rounding of the PP bits during the design of fast parallel truncated multipliers, so that the final truncated product satisfies the precision requirement. In this approach truncation error is not more than 1ulp, so there is no need of error compensation circuits, and the final output will be précised.

REFERENCES

- [1] J. E. Stine and O. M. Duverne, —Variations on truncated multiplication,|| in Proc. EuromicroSymp. Digit. Syst. Des., 2003, pp. 112–119.
- [2] J. M. Jou, S. R. Kuang, and R. D. Chen, —Design of low-error fixed- width multipliers for DSP applications,|| IEEE Trans. Circuits Syst. II, s Analog Digit. Signal Process., vol. 46, no. 6, pp. 836–842, Jun. 1999.
- [3] L.-D. Van and C.-C. Yang, —Generalized low-error area-efficient fixed width multipliers,|| IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 8, pp. 1608–1619, Aug. 2005.
- [4] M. J. Schulte and E. E. Swartzlander, Jr., —Truncated multiplication with correction constant,|| in VLSI Signal Processing VI. Piscataway, NJ:IEEE Press, 1993, pp. 388–396.
- [5] E. J. King and E. E. Swartzlander, Jr., —Data-dependent truncation scheme for parallel multipliers,|| in Proc. 31st Asilomar Conf. Signals, Syst. Comput., 1997, pp. 1178–1182.
- [6] M. J. Schulte, J. G. Hansen, and J. E. Stine, —Reduced power dissipation through truncated multiplication,|| in Proc. IEEE Alessandro Volta Memorial Int. Workshop Low Power Des., 1999, pp. 61–69.
- [7] T.-B. Juang and S.-F. Hsiao, —Low-error carry-free fixed-width multipliers with low-cost compensation circuits,|| IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 6, pp. 299–303, Jun. 2005.
- [8] A.G.M. Stollo, N. Petra, and D. De Caro, —Dual-tree error compensation for high-performance fixed-width multipliers,|| IEEE Trans. Circuits Syst. II, Exp. Briefs, vol.52, no. 8, pp. 501–507, Aug. 2005.
- [9] E. G. Walters and M. J. Schulte, —Efficient function approximation using truncated multipliers and squarers,|| in Proc. 17th IEEE Symp. ARITH, 2005, pp. 232–239.
- [10] C. S. Wallace, —A suggestion for a fast multiplier,|| IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.



- [11] L. Dadda, —Some schemes for parallel multipliers,|| *Alta Frequenza*, vol. 34, pp. 349–356, 1965.
- [12] N. Petra, D. De Caro, V. Garofalo, E. Napoli, and A. G.M. Strollo, —Truncated binary multipliers with variable correction and minimum mean square error,|| *IEEE Trans.Circuits Syst. I, Reg. Papers*, vol. 57, no. 6, pp. 1312–1325, Jun. 2010.
- [13] P Brundavani, DV Vardhan, P Mahesh, R Suresh - DESIGN AND ANALYSIS OF 16-BIT RISC PROCESSOR USING LOW POWER PIPELINE, National Conference on Emerging trends in Information, management and Engineering sciences “NC’ea-TIMES#1.0”-2018, *ijet journal*, ISSN: 2395-1303, PP. 1-8.
- [14] J.-A. Pineiro, S. F. Oberman, J. M. Muller, and J. D. Bruguera, —Highspeed function approximation using a minimax quadratic interpolator,|| *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 304–318, Mar. 2005.
- [15] K. C. Bickerstaff, M. Schulte, and E. E. Swartzlander, Jr., —Parallel reduced area multipliers,|| *J. VLSI Signal Process.*, vol. 9, no. 3, pp.181–191, 1995.
- [16] J.-P. Wang, S.-R. Kuang, and S.-C. Liang, —High-accuracy fixed-width modified booth multipliers for lossy applications,|| in *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, Jan. 2011, vol. 19, no.1, pp. 52–60.