

DESIGN OF MCMA/MCM TECHNIQUE BY ROUNDED TRUNCATED MULTIPLIER FOR FIR FILTER

NAM. SANDHYA⁽¹⁾, SANDEEP SUNKARI⁽²⁾

M.TECH, VLSI AND EMBEDDED SYSTEM DESIGN, PRASAD ENGINEERING COLLEGE⁽¹⁾

M.TECH, ASSOC.PROFESSOR, , PRASAD ENGINEERING COLLEGE⁽²⁾

ABSTRACT:-The high speed FIR Filter designs are presented using the concept of faithfully rounded truncated multipliers. Truncated multipliers offer significant improvements in area, delay, and power. We 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 multiplication / accumulation in a direct FIR structure is implemented using an improved version of truncated multipliers. Comparisons with previous FIR design approaches show that the proposed designs achieve the best area and power results. This design is simulated using Modelsim6.4b and synthesized using XILINX ISE10.1 software's.

INDEX TERMS:-Digital signal processing (DSP), faithful rounding, finite impulse response (FIR) filter, truncated multipliers, VLSI design.

I. INTRODUCTION

Finite impulse response (FIR) digital filter is one of the fundamental components in many digital signal processing (DSP) and communication systems. It is also widely used in many portable applications with limited area and power budget. A general FIR filter of order M can be expressed as

$$y[n] = \sum_{i=0}^{M-1} a_i x[n - i].$$

There are two basic FIR structures, direct form and transposed form, as shown in Fig. 1 for a linear-phase even-order FIR filter. In the direct form in Fig, the multiple constant multiplication (MCM) / accumulation (MCMA) module performs the concurrent multiplications of individual delayed signals and respective filter coefficients, followed by accumulation of all the products. Thus, the operands of the multipliers in MCMA are delayed input signals $x[n-i]$ and coefficients a_i .

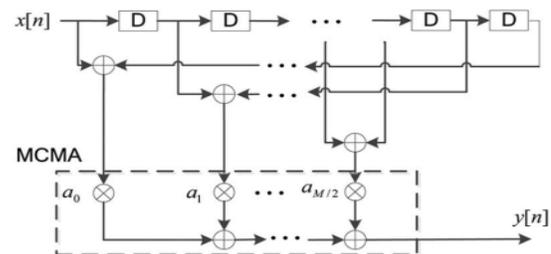


Fig: Structures of linear-phase even-order FIR filter direct form

In the transposed form in Fig, the operands of the multipliers in the MCM module are the current input signal $x[n]$ and coefficients. The results of individual constant multiplications go through structure adders (SAs) and delay elements. In the past decades, there are many papers on the designs and implementations of low-cost or high-speed FIR filters. In order to avoid costly multipliers, most prior hardware implementations of digital FIR filters can be divided into two categories: multiplier less based and memory based.

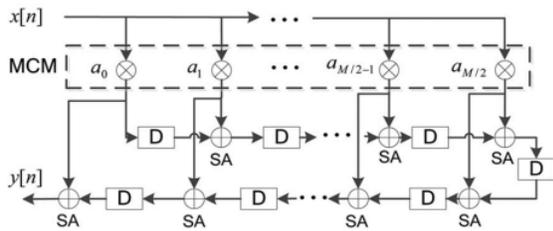


Fig: Structures of linear-phase even-order FIR filters transposed form

Multiplier less-based designs realize MCM with shift-and add operations and share the common sub operations using canonical signed digit (CSD) recoding and common sub expression elimination (CSE) to minimize the added cost of MCM. Mostly area savings are achieved by jointly considering the optimization of coefficient quantization and CSE. Most multiplier less MCM-based FIR filter designs use the transposed structure to allow for cross-coefficient sharing and tend to be faster, particularly when the filter order is large. However, the area of delay elements is larger compared with that of the direct form due to the range expansion of the constant multiplications and the subsequent additions in the SAs. In Blad and Gustafsson presented high-throughput (TP) FIR filter designs by pipelining the carry-save adder trees in the constant multiplications using integer linear programming to minimize the area cost of full adders (FAs), half adders (HAs), and registers (algorithmic and pipelined registers).

II.FIR FILTER DESIGN AND OPTIMIZATION

A digital FIR filter design consists of three steps and they are finding the filter order and coefficient, coefficient quantization, and hardware optimization. In the first step, the filter order and its corresponding coefficients are determined in order to satisfy the frequency response specifications. Then in the second step, coefficient quantization is performed to quantize the coefficients to finite bit width accuracy. Finally hardware optimization is done to reduce the area and power consumption of

hardware structure.

Specification of frequency response

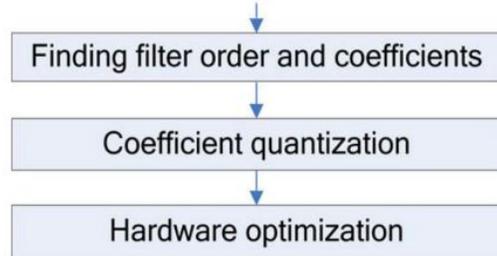


Fig: Three stages in digital FIR filter design and implementation

Hardware optimization is done by optimizing the design of multipliers used in FIR filter structure which occupies more area and consumes more power. Generally the design of parallel multipliers consists of following steps, i.e., partial product generation, partial product reduction, final carry propagate addition. In the first step, the partial products are generated from multiplicand and multiplier. In the partial product reduction step, the generated partial products are compressed and reduced to two partial products and then they are finally added using final carry propagate adder.

Already two famous reduction techniques available and they are Dadda tree and Wallace tree reduction. In Dadda reduction, whenever compression operation is required the reduction is performed whereas in Wallace tree reduction, the partial product bits are always compressed.

There are two types of reduction schemes adopted in this paper called scheme-1 and scheme-2 reductions in order to reduce the number of half adders used in each column and also to perform flexible column-by-column reduction. By combining scheme-1 and scheme-2 reduction in the design of truncated multiplication, the area cost can be minimized.

compression does not need to generate the

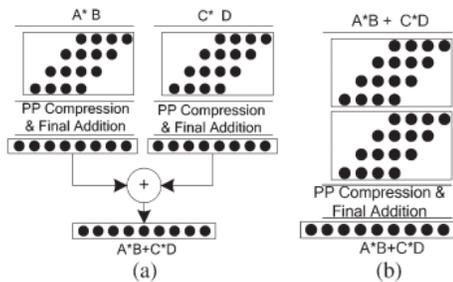


Fig: Multiplication/accumulation using individual PP compression and combined PP compression. (a) (b)

III. TRUNCATED MULTIPLIER DESIGN

Truncated multiplication reduces power consumption by computing only the most significant bits of product result and most common approach used for truncation is physical reduction of partial product bit matrix. The main objective of improved faithfully rounded truncated multiplier design is to compute P most significant bits of the product with a total truncation error of not more than $1ulp$, where $1ulp = 2^{-P}$. In the Existing design of truncated multiplication, there are three processes to remove the unnecessary Partial Product Bits (PPBs) and they are: deletion, truncation, and rounding. Two rows of partial product bits are set undeletable because they will be removed at the subsequent process of truncation and rounding. The two white dots at level 1 of reduction and the two white dots at level 2 of reduction are not generated during the process of compression with FAs or HAs.

Thus, two simplified versions of the full adder (FA) and half adder (HA) cells are introduced, i.e., full adder and half adders without the sum output bits. For column T , the carry bit only is needed to be generated (to column $T + 1$) for the last full adder compression because the sum output bit will be discarded during the rounding process. For example, the FA

white dot (the sum output bit) at level 4 of reduction.

For column $T + 1$ to $M + N$, although Scheme 1 is adopted to determine whether an HA is needed or not, we actually do not compress the column height to one because this compression will cause ripple carry. Indeed, at the last level of the reduction process, some column, for example column i , has a height of three, and the remaining columns beyond this specific column, i.e., columns $i + 1, i + 2, \dots$ etc have a column height of two, as shown in level 4 of Fig. Finally compressed partial product bits are added using final carry propagate adder.

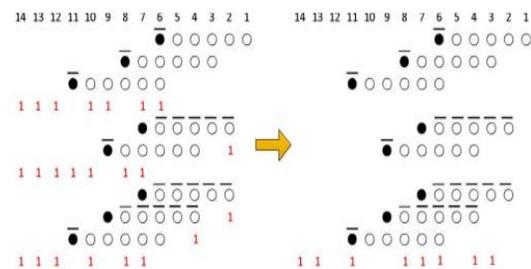


Fig: Generation of PPBs considering sign extension and negation.

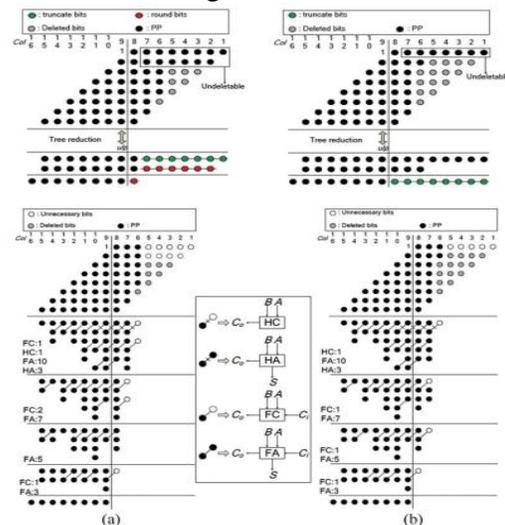


Fig: Truncated multiplier designs using (a) the Existing System (b) the improved version.

The error ranges of deletion, truncation, and rounding before and after adding the offset constants are given as follows:



- [2] C.-H. Chang, J. Chen, and A. P. Vinod, "Information theoretic approach to complexity reduction of FIR filter design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 8, pp. 2310–2321, Sep. 2008.
- [3] F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution—A new approach to versatile sub expressions sharing in multiple constant multiplications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 2, pp. 559–571, Mar. 2008.
- [4] F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution algorithms for common sub expression elimination in digital filter design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 10, pp. 695–700, Oct. 2005.
- [5] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on an algorithm to generate all minimal signed digit representations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 12, pp. 1525–1529, Dec. 2002.
- [6] C.-Y. Yao, H.-H. Chen, T.-F. Lin, C.-J. J. Chien, and X.-T. Hsu, "A novel common-sub expression-elimination method for synthesizing fixed-point FIR filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 11, pp. 2215–2221, Sep. 2004.
- [7] O. Gustafsson, "Lower bounds for constant multiplication problems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 11, pp. 974–978, Nov. 2007.
- [8] Y. Voronenko and M. Puschel, "Multiplierless multiple constant multiplication," *ACM Trans. Algorithms*, vol. 3, no. 2, pp. 1–38, May 2007.
- [9] D. Shi and Y. J. Yu, "Design of linear phase FIR filters with high probability of achieving minimum number of adders," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 1, pp. 126–136, Jan. 2011.
- [10] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficient," *IEEE Trans. Circuits Syst.*, vol. 36, no. 7, pp. 1044–1047, Jul. 1989.
- [11] Y. C. Lin and S. Parker, "Discrete coefficient FIR digital filter design based upon an LMS criteria," *IEEE Trans. Circuits Syst.*, vol. 30, no. 10, pp. 723–739, Oct. 1983.
- [12] K. C. Bickerstaff, M. Schulte, and E. E. Swartzlander, Jr., "Reduced area multipliers," in *Proc. Int. Conf. Appl.-Specific Array Processors*, 1993, pp. 478–489.