To Reduce The Complexity Of FIR Filters Using Different Algorithms – A Review

Sandeep Kaur
M. Tech Student, Department of ECE, Punjabi University, Patiala

Er. Sangeet Pal Kaur
Assistant Professor, Department of ECE, Punjabi University, Patiala

Abstract: This paper aims at studying the various algorithms available in literature for the design of FIR filters. It further aims at to reduce the complexity of FIR filters. This complexity can be dominated by the number of adders or subtractors. Thus, it is important to reduce the numbers of adders or subractors for hardware efficient FIR filters.

Keywords: Common subexpression elimination (CSE), Canonic signed digit (CSD), Iterated matched (ITM).

I. INTRODUCTION

Finite impulse response (FIR) filters are the most popular type of filters which are used in every field of digital signal processing. Filters are the devices that removes unwanted signals. Each function is accepting an input signal, blocking the pre-specified frequency components and passing through the original signal minus those components to the output. In digital filtering application, software running on a digital signal processor (DSP) reads input samples from an A/D converter, performs the mathematical manipulations required for the filter type, and outputs results via D/A converter. In DSP, some applications need FIR filter to operate at high frequencies such as video processing, whereas some at low-power circuit with high throughput such as multiple-input multiple-output (MIMO) systems used in cellular wireless communication. In this paper, discuss various algorithms to reduce the complexity of the FIR filter in the past [1]–[7]. Multipliers are the major portions where more hardware consumption for the parallel FIR filter implementation. The main aim is to reduce the number of multipliers at the expense of adders to reduce area, delay and power. Replacing multipliers with adders is advantageous because adders weigh less than multipliers in terms of area and they do not increase the length of the filter whereas the multipliers increases along with the length of the filter and also complexity increases.

II. FINITE IMPULSE RESPONSE

Filters can be classified in several different groups, depending on the criteria are used for classification. There are two major types of digital filters are: finite impulse response digital filters (FIR filters) and infinite impulse response digital filters (IIR). Both types have some advantages and disadvantages that can be considered when designing a filter. Besides, it is necessary to take into all fundamental characteristics of a signal to be filtered for deciding which filter is used.

III. REVIEW OF VARIOUS ALGORITHMS TO REDUCE COMPLEXITY OF FIR FILTERS

The complexity of Finite Impulse Response (FIR) filters is dominated by number of adders used to implement the coefficient multipliers. A greedy Common Subexpression Elimination (CSE) algorithm is used with a look-ahead method which is based on the Canonic Signed Digit (CSD) representation of filter coefficients for implementing low complexity FIR filters. Look ahead algorithm chooses the maximum number of frequently occurring common subexpressions and hence reduces the number of adders required to implement the filter. This adder reduction is achieved without any increase in critical path length. This method offers an average adder reduction of about 20% over the best known CSE method. In this paper, the look-ahead method maximizes the grouping of the subexpressions, thus leaving minimum number of unpaired nonzero bits. The logic depths of filters implemented using this method is almost identical to that of BSE[8].

A novel optimization technique is used to optimize filter coefficients of linear phase finite impulse response (FIR) filter to share common subexpressions within and among coefficients. Existing approaches of common subexpression elimination optimize digital filters in two stages: first, FIR filter is designed in a discrete space such as finite word length...
space or signed power-of-two (SPT) space to meet given specifications and in second stage, an optimization algorithm is applied on the discrete coefficients to find and eliminate the common subexpressions. Such a two-stage optimization technique suffers from the problem of search in the second stage is limited by the finite word length or SPT coefficients obtained in the first stage optimization. The algorithm used in [9] overcomes this problem by optimizing the filter coefficients directly in subexpression space for a given specification. The required number of adders obtained using the algorithm is much less than those obtained using two-stage optimization approaches. Mixed integer linear programming (MILP) is used to optimize filter coefficients in subexpression spaces. The number of adders used to synthesize the filter coefficients are significantly reduced when comparing with those obtained using the traditional two-stage optimization approach. Conventionally, MILP is notorious for its high computation load for the design of high-order filters. In this paper, a multi-step optimization approach is taken to overcome the difficulty of optimizing long filter in subexpression space. The algorithm used in [9] also showed its flexibility for designing filters with other constraints such as the maximum adder steps of coefficients when filters are implemented in transposed direct form [9].

A novel algorithm for designing low-power and hardware-efficient linear-phase finite impulse response (FIR) filters. The algorithm finds filter coefficients with reduced number of signed-power-of-two (SPT) terms given the filter frequency response characteristics. The algorithm is a branch-and-bound-based algorithm that fixes a coefficient to a certain value. The value is determined by finding the boundary values of the coefficient using linear programming. Although the run time of the algorithm is exponential, its capability to find appreciably good solutions in a reasonable amount of time makes it a desirable for designing low-power and hardware-efficient filters (at least 100 times faster than traditional optimum MILP based formulations). The superiority of the algorithm on existing methods in terms of SPT term count, design time, hardware complexity, and power performance. Since the algorithm employs a gradually expanding subexpression space, it does not need to specify a good basis set that is required by the subexpression space method given in [9]. Compared with earlier results that the number of adders produced by this method is less than or equal to that of the earlier results. For the best cases, the number of adders produced by this method is 81.8% and 84.2% of the best results [10]. A new cost-efficient algorithm to implement finite impulse response (FIR) filters based upon the common subexpression elimination (CSE) algorithm using canonical signed digit (CSD) representation of filter coefficients. The complexity of implementation of FIR filters is determined by the number of adders/subtractors needed to implement the multiplication operations. This algorithm enables an efficient architecture, especially for high order FIR filters, that requires fewer adders/subtractors than similar algorithms previously. This algorithm relies on significant modifications to the iterated matched (ITM) algorithm presented in [9] to find and eliminate more common subexpressions amongst filter coefficients which is translated into a significant power and area saving when implementing high-order FIR filters. Experimental results show that this algorithm can contribute up to a 20 percent reduction in the complexity of FIR filters over some other CSE-based algorithms. The experimental results show that implementation of FIR filters using the Modified Iteration matched algorithm (MITM) algorithm can save up to 20 percent of the additions/subtractions when compared to conventional CSE algorithms [11], especially when the comparison is made with the ITM algorithm [9]. The results also proves that implementing high order FIR filters using the MITM algorithm is more efficient than their implementation using other algorithms. Additionally, the runtime of the MITM algorithm is shorter than the runtime of the ITM algorithm in [9].

A common subexpression elimination tree algorithm to minimize the complexity of the multiple constant multiplication operation. The coefficients of multiple constant multiplication are represented using binary signed digit number system. The binary signed digit representation of each coefficient is enumerated using representation tree. The algorithm transverses to calculate the possible subexpressions at each node. Each expression is used to find the possible decomposition for the coefficient to be encoded. The algorithm is designed to prune the tree when it finds a decomposition with minimum complexity [12]. Further various improved algorithms were proposed namely expanding subexpression space set - membership normalized least mean square (SMNLM), greedy common subexpression elimination. These algorithms enjoy less computation complexity [13].

IV. CONCLUSION

In this paper, a study of the various algorithms to reduce the complexity of FIR filters available in literature are presented. Out of all the algorithms reviewed, the expanding common subexpression elimination algorithm performs better results than the others in terms of number of adders. Certain issues still remains and new and improved techniques are required in order to tackle the issues.

REFERENCES


