

General Article

Design of Digit-Serial FIR Filters using Graph Base Technique

Shailesh S. Nichat^{a*} and Shrikant J. Honade^a

^aG. H. Raisoni College of Engineering & Management, Amravati

Accepted 29 May 2013, Available online 1 June 2013, Vol.3, No.2 (June 2013)

Abstract

In last two decades, many efficient algorithms and architectures have been introduced for the design of low complexity bit-parallel Multiple Constant Multiplications (MCM) operation which dominates the complexity of many digital signal processing systems. On the other hand, little attention has been given to the digit-serial MCM design that offers alternative low complexity MCM operations at the cost of an increased delay. In this topic, we address the problem of optimizing the gate-level area in digit-serial MCM designs and introduce high level synthesis algorithms, design architectures, and a computer aided design tool. The proposed optimization algorithms for the digit-serial MCM architectures in the design of digit-serial MCM operations and finite impulse response filters yields better performance compared with multiple Constant multiplier using Common Sub-expression Elimination (CSE) algorithm with high efficiency.

Keywords: Gate-level area optimization, multiple constant multiplications, Common Sub-expression Elimination (CSE) algorithm, Graph Base (GB) algorithm

Introduction

Finite Impulse Response (FIR) filters are of great importance in Digital Signal Processing (DSP) systems since their characteristics in linear-phase and feed-forward implementations make them very useful for building stable high-performance filters. The direct and transposed-form FIR filter implementations are illustrated in Fig. 1(a) and (b), respectively. Although both architectures have similar complexity in hardware, the transposed form is generally preferred because of its higher performance and power efficiency. The multiplier block of the digital FIR filter in its transposed form [Fig. 1(b)], where the multiplication of filter coefficients with the filter input is realized, has significant impact on the complexity and performance of the design because a large number of constant multiplications are required. This is generally known as the Multiple Constant Multiplications (MCM) operation and is also a central operation that has performance bottleneck in many other DSP systems such as fast Fourier Transforms, Discrete Cosine Transforms (DCTs), and error-correcting codes.

Literature Review

In the last two decades, many efficient algorithms and architectures have been introduced for the design of low

complexity bit-parallel multiple constant multiplications (MCM). Bit-parallel realization of the multiplication of a

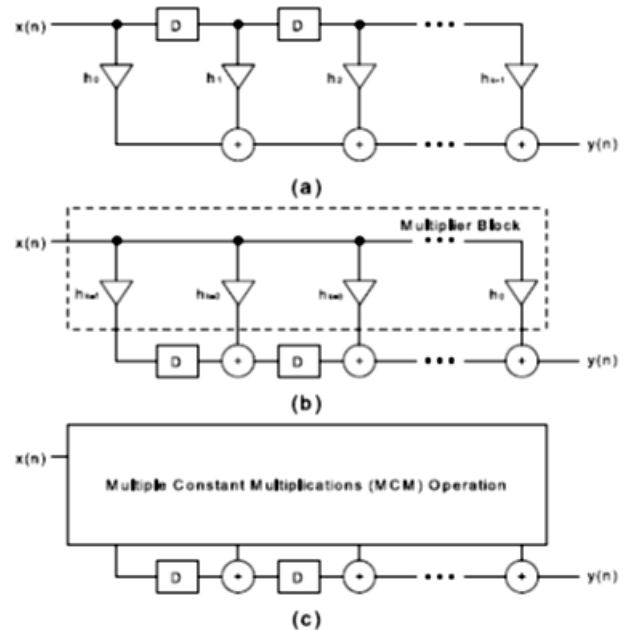


Fig.1: FIR filters implementations. (a) Direct form. (b) Transposed form with generic multipliers. (c) Transposed form with an MCM block.

variable by a set of constants using only addition, subtraction, and shift operations has been explored extensively over the years as large number of constant

*Corresponding author: Shailesh S. Nichat

multiplications dominates the complexity of many digital signal processing systems. On the other hand, digit-serial architectures offer alternative low-complexity designs since digit-serial operators occupy less area and are independent of the data word length. Also, it is mentioned that the full flexibility of a multiplier is not necessary for the constant multiplications, since filter coefficients are fixed and determined beforehand by the DSP algorithms. Hence, the multiplication of filter coefficients with the input data is generally implemented under a shift adds architecture, where each constant multiplication is realized using addition/subtraction and shift operations in an MCM operation.

One of the important implementation of the Multiplication of a variable with a set of constants also known as the MCM operation, is a central operation and achieves performance bottleneck in many DSP applications such as, error correcting codes, linear DSP transforms, and Finite Impulse Response (FIR) filters. In hardware, the multiplication operation is considered to be expensive, as it occupies significant area. Hence, constant multiplications are generally realized using only addition, subtraction, and shift operations.

Yuen-Hong Alvin Ho: In the context of multiple constant multiplications (MCM) design, they proposed a novel Common-Sub expression-Elimination (CSE) algorithm that models the synthesis of coefficients into an estimated cost function. Although the proposed algorithm generally is not capable of finding the minimum/minima of the function in practically sized problems and don't guarantee an optimum solution.

From the review of various research paper, it is found that an approximate Graph Base (GB) algorithm is consider to be the best solution to overcome this problem in order to achieve the optimal gate level area in digit serial MCM design compared to the CSE algorithm and also introduce the design architecture for the digit serial MCM operations and FIR filters.

Exact CSE Algorithm

The exact CSE algorithm consists of four main steps. First, all possible implementations of constants are extracted from the nonzero digits of the constants defined under a number representation: binary, CSD, or MSD. Then, the implementations of constants are represented in terms of a Boolean network. Third, the gate-level area optimization problem is formalized as a 0-1 ILP problem with a cost function to be minimized and a set of constraints to be satisfied. Finally, a set of operations that yields the minimum area solution is obtained using a generic 0-1 ILP solver. These four steps are described in detail next.

Finding the Partial Terms

In the preprocessing phase, the constants to be multiplied by a variable are converted to positive, and then made odd by successive divisions by 2. The resulting constants are stored without repetition in the target set T . Thus, T includes the minimum number of necessary constants to

be implemented. The part of the algorithm where the implementations of the target constants and partial terms are found is as follows.

- 1) Take an element from T , ti , find its representation(s) Under the given number representation, and store it (them) in a set called S . Form an empty set O_i , associated with ti , that will include the inputs and the amount of left shifts at the inputs of all addition/subtraction operations which generate ti .
- 2) For each representation of ti in the set S .
 - a) Compute all non symmetric partial term pairs that cover the representation of ti .
 - b) In each pair, make each partial term positive and odd, and determine its amount of left shift.
 - c) Add each pair to the set O_i with the amount of left shifts of partial terms.
 - d) Add each partial term to T , if it does not represent the input that the constants are multiplied with, i.e., denoted by 1, and is not in T .
- 3) Repeat Step 1 until all elements of T are considered.

Observe that the target set T only includes the target constants to be implemented in the beginning of the iterative loop, and in later iterations it is augmented with the partial terms that are required for the implementation of target constants. All possible implementations of an element in the target set ti are found by decomposing the nonzero digits in therepresentation of ti , into two partial terms. As an example, consider 25 as a target constant defined under MSD, which has two representations 011001 and 101001. All possible implementations of 25 are given in Fig. 6. Observe from Fig. 6 that the last implementations of 25 on both representations, i.e., $1 + 3 \leq 3$, are identical, therefore one of them can be eliminated. Also, the duplications of implementations, such as $1 \leq 4 + 9 = 9 + 1 \leq 4$, are not listed in Fig. 6. After the partial terms required for the implementation of 25 under MSD, i.e., 3, 7, 9, 17, and 33, are found, they are added to the target set T without repetition and their implementations are determined in a similar way.

Approximate GB Algorithm

Note that the solution of an exact CSE algorithm described in privies Section is not the global minimum since all possible implementations of a constant are found from its representation. Also, the optimization of gate-level area problem in digit-serial MCM design is an NP-complete problem due to the NP-completeness of the MCM problem. Thus, naturally, there will be always 0-1 ILP problems generated by the exact CSE algorithm that current 0-1 ILP solvers find difficult to handle. Hence, the GB heuristic algorithms, which obtain a good solution using less computational resources, are indispensable. In our approximate algorithm called MINAS-DS, as done in algorithms designed for the MCM problem given in Definition1, we find the fewest number of intermediate

constants such that all the target and intermediate constants are synthesized using a single operation. However, while selecting an intermediate constant for the implementation of the not-yet synthesized target constants in each iteration, we favor the one among the possible intermediate constants that can be synthesized using the least hardware and will enable us to implement the not-yet synthesized target constants in a smaller area with the available constants. After the set of target and intermediate constants that realizes the MCM operation is found, each constant is synthesized using an *A-operation* that yields the minimum area in the digit-serial MCM design. In MINAS-DS, the area of the digit-serial MCM operation is determined as the total gate-level implementation cost of each digit-serial addition, subtraction, and shift operation under the digit size parameter d .

area and power dissipation than the algorithms designed for the MCM problem and the optimization of area problem in a digit-serial MCM operation at gate-level. The realization of digit-serial FIR filters under the shift-adds architectures yields significant area and power reductions when compared to those whose multiplier blocks are implemented using digit-serial constant multipliers. In this paper, we proposed an approximate Graph Base (GB) algorithm that finds the best partial product in each iteration which yields the optimal gate level area in digit serial MCM design and also introduce the design architecture for the digit serial MCM operations and FIR filters.

References

R. Hartley and P. Corbett (1990), Digit-Serial Processing Techniques. IEEE TCAS II, 37(6):707-719,
 L. Aksoy, E. Costa, P. Flores, and J. Monteiro (2008), Exact and Approximate Algorithms for the Optimization of Area and Delay in Multiple Constant Multiplications. IEEE TCAD, 27(6):1013-1026.
 L. Aksoy, E. Gunes, and P. Flores (2010), Search Algorithms for the Multiple Constant Multiplications Problem: Exact and Approximate. Elsevier Journal on Microprocessors and Microsystems, 34:151-162.
 L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro (2011), Optimization of Area in Digit-Serial Multiple Constant Multiplications at Gate-Level. In ISCAS, to appear.
 I.C. Park and H.-J. Kang (2001), Digital filter synthesis based on minimal signed digit representation, in Proc. DAC, 2001, pp. 468-473.

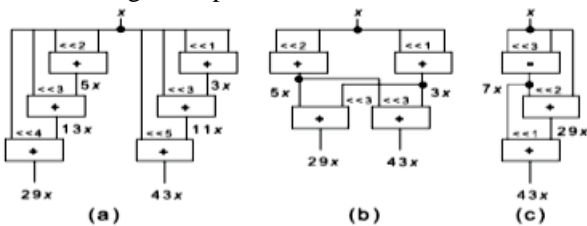


Fig. 2. Shift-adds implementations of 29x and 43x. (a) Without partial product sharing and with partial product sharing. Exact CSE algorithm. (c) Exact GB algorithm.

Proposed Work

The CSE algorithm obtains better solutions in terms of