International Journal of Engineering Research and General Science Volume 3, Issue 2, Part 2, March-April, 2015
ISSN 2091-2730

FPGA IMPLEMENTATION OF FIR FILTER IN SIGNAL PROCESSING

*Saranya.S, Anand Kumar.V

M.E(Applied Electronics),Sri Eshwar college of Engineering Anna University, Coimbatore

Email: saranyasoundher15@gmail.com

Abstract-Multiple constant multiplication scheme is the most effective common sub expression sharing technique which is used for implementing the transposed FIR filters. Ripple carry operation allows adder tree to minimize hardware cost, unfortunately it detriment timing and gives low speed operation. To outperform this high speed adder is proposed and analyzed for real time speech signal applications. The resource minimization problem in the scheduling of adder tree operation based mixed integer programming (MIP) algorithm for more efficient multiple constant multiplication (MCM)based implementation of FIR filters are identified. The proposed adder tree consist of carry select adder to improve the speed of the fir filter. By using Graph based algorithm to choose the particular partial product in the multiple constant multiplication block. Which reduces number of adders in the scheduling of adder tree for FIR filter

Keywords— Adder tree optimization, multiple constant multiplication, MCM, Transposed finite impulse response filter, FIR

INTRODUCTION

Digital signal processing has many advantages over analog signal processing. Digital signals are more robust than analog signals with respect to temperature and process variations. The accuracy in digital representations can be controlled better by changing the word length of the signal. Furthermore, DSP techniques can reduce noise and interference while amplifying the signal. In contrast, both signal and noise are amplified in analog signal received, processed and manipulated, all virtually without error. While analog signal processing is indispensable for systems that require extremely high frequencies such as the radio frequency transceiver in wireless communications, or extremely low area and low power such as micro machine sensors used to detect cracks and other stress-related material defects, many complex systems are realized digitally with high precision, high signal to noise ratio (SNR), repeatability, and flexibility. The DSP systems can be realized using programmable processors or custom designed hardware circuits fabricated using very large scale integrated (VLSI) technology. The goal of digital design is to maximize the performance while keeping the cost down. Two important features that distinguish DSP from other general purpose computations are the real time throughput requirement and the data driven property.

The finite-impulse response (FIR) filter has been and continues to be one of the fundamental processing elements in any digital signal processing (DSP) system. FIR filters are used in DSP applications that range from video and image processing to must be a low-power circuit, capable of operating at moderate frequencies. Parallel, or block, processing can be applied to digital FIR filters to either increase the effective throughput or reduce the power consumption of the original filter. FIR filters are digital filters with finite impulse response. They are also known as non-recursive digital filters as they do not have the feedback (a recursive part of a filter), even though recursive algorithms can be used for FIR filter realization. FIR filters can be designed using different methods, but most of them are based on ideal filter approximation. FIR filter transfer function can be expressed as

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{n=0}^{N-1} h[n].z^{-n}$$

The frequency response realized in the time domain is of more interest for FIR filter realization (both hardware and software). The transfer function can be found via the z-transform of a FIR filter frequency response. FIR filter output samples can be computed using the following expression:

$$y(n) = \sum_{n=0}^{N-1} h[k].x[n-k]$$

Where

x[k] are FIR filter input samples;
h[k] are the coefficients of FIR filter frequency response; and
y[n] are FIR filter output samples.

FIR realized using two forms
a)Direct Form
b)Transposed Form

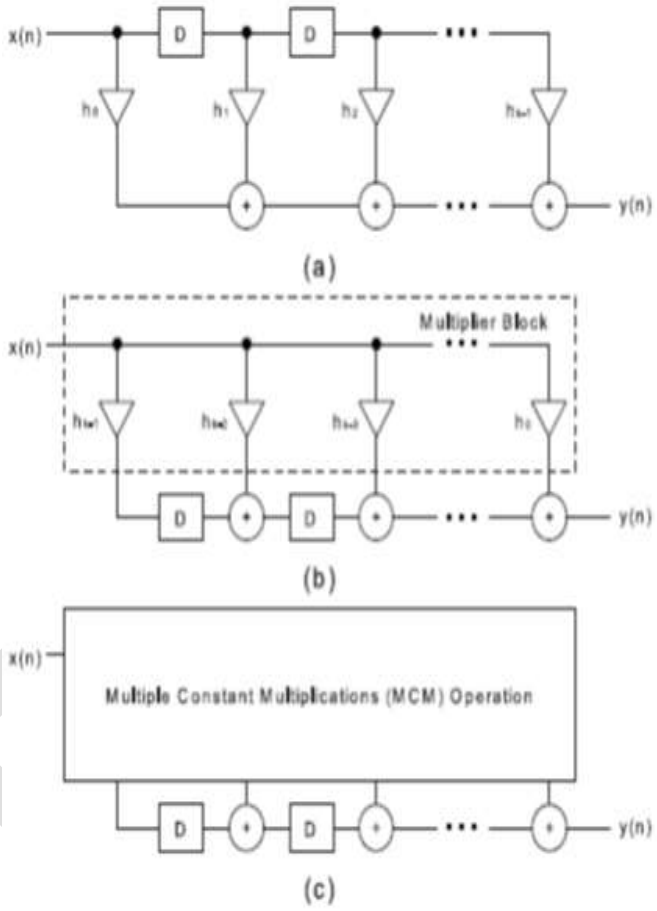

Figure 1.1 (a) Direct form (b) Transposed form (c) Transposed form with MCM block

In direct form delay units are present between the multipliers. But in transposed form delay units are between adders so that multipliers can fed directly. The main drawback of direct form is consumes more power. so transposed form is preferred due to its higher performance and power efficiency.

MCM Concept

Efficient FIR filters use more number of multipliers. Multipliers are used for the multiplication of filter coefficient. But multiplier occupies more area. Due to this FIR filter consumes large chip area. Multiplier block of FIR filter performs multiplication of filter input with the set of filter coefficients. This is known as Multiple Constant Multiplication (MCM) i.e, multiplication of a variable by set of constant. Example: 29x,43x. For the multiplier-less realization of constant multiplications, the fundamental operation is called A-operation. It is an operation with two integer inputs and one integer output that performs a single addition or subtraction, and an arbitrary number of shifts. It is defined as follows. For the multiplier-less realization of constant multiplications, the fundamental operation is called A-operation. It is an operation with two integer inputs and one integer output that performs a single addition or subtraction, and an arbitrary number of shifts. It is defined as follows

where

1) $s \in [0,1]$ is the sign, which determines if an addition or a subtraction operation is to be performed.

2) $l_1, l_2 \geq 0$ are the integers denoting left shifts of the operands.

3) $r \geq 0$ is an integer indicating a right shift of the result.

$$w = A(u, v) = \left|2^{l_1}u + (-1)^s 2^{l_2}v\right|2^{-r}$$

Shift –Add Architecture

Multiple constant multiplication (MCM) can be realized using "Shift-Add Architecture" which includes shifting and adding operation.

There exist several techniques to implement this shift-add architecture. While implementing occurs many problems.

PROPOSED SYSTEM

Approximate GB Algorithm

In the solution of Exact CSE algorithm all possible implementations are found from its representation, hence it is not the global minimum. 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. Exact CSE algorithm always generate 0-1 ILP problems which is difficult to handle by the current 0-1 ILP solvers. Hence GB heuristic algorithms, which provide a good solution using less computational resources are necessary.

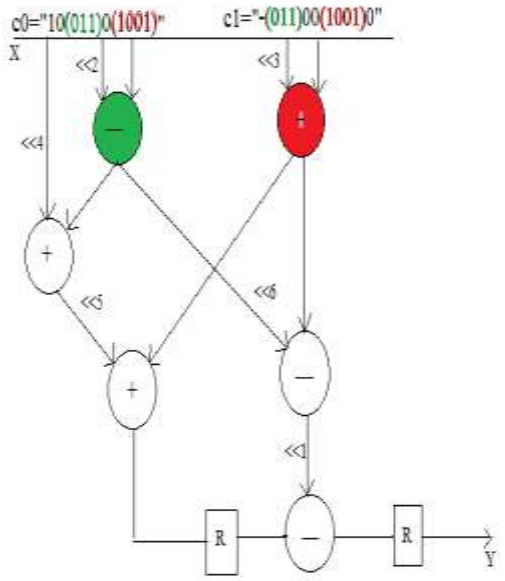

Figure 1.2 Adder tree of the FIR Filter

In MINAS-DS algorithm designed for the MCM problem, few number of intermediate constants are found. All the targets and intermediate constants are synthesized using a single operation. While selecting an intermediate constants for the implementation of not yet synthesized target constants we choose the one that can be synthesized using least hardware .This 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 are found, each constant is synthesized using a A-operation that yields the minimum area in the digit-serial MCM design. Usually GB is based on MINAS-DS algorithm.

In MINAS-DS, the ready set R ={1}is formed initially. The target constants, which can be implemented using a single operation are found and moved to the ready set using the Synthesize function. If unimplemented constants exist in the target set, then an intermediate constant is added to the ready set until there is no element in the target set. The MINAS-DS algorithm considers positive and odd constants that are not included in the current ready and target sets. These constants are implemented with the elements of current ready set using a single operation. The Compute Cost Function searches all A-operation that computes the constant with the elements of the current ready set. Then the cost of each operation under the digit-serial architecture is determined and returns minimum implementation cost among possible operations. Else it returns a 0 value, indicating that the constant cannot be synthesized using an operation with the elements of the current ready set.

After a possible intermediate constant is found, it is added into the working ready set A and its effect on the current target set are found using ComputeTCost function. Using this function the minimum digit-serial implementation costs of the target constants that can be synthesized with the elements of the working ready set are determined. The cost of intermediate constant is determined which is equal to its minimum implementation cost plus the cost of the not-yet synthesized target constants. After the cost value of each possible intermediate constant is found the one with the minimum cost is added to the current ready set and its effect on the current target set are found using the Synthesize function.

When no elements are left in the target set , the Synthesize MinArea function is applied on the final set to find the set of a A-operations that yields a solution with the optimal area. The size of 0-1 ILP problem is much smaller than the 0-1 ILP generated by the exact CSE algorithm. Finding the minimum solution of this 0-1 ILP problem is much simpler because the possible implementations of a constant are limited to the elements in the final ready set.

CONCLUSION

In this project, the resource minimization formalization for designing digit-serial MCM operation with optimal area at the gate level by considering the implementation costs of digit-serial addition, subtraction, and shift operations was introduced. Since there are still instances with which the exact greedy algorithm cannot cope, an approximate GB algorithm that finds the best partial products in each iteration which yield the optimal gate-level area in digit-serial MCM design was proposed. The experimental results indicate that the complexity of digit-serial MCM designs can be further reduced using the high-level optimization algorithms proposed in this project. It was shown that the realization of digit-serial FIR filters under the shift-adds architecture yields significant area reduction when compared to the filter designs whose multiplier blocks are implemented using digit serial constant multipliers.

REFERENCES:

[1] Yu Pan and Pramod Kumar Meher, Senior Member, IEEE, "Bit-Level Optimization of Adder-Trees for Multiple Constant Multiplications for Efficient FIR Filter Implementation,"ieee transaction on circuits and systems-1: Regular papers, vol. 61, no. 2, Feb 2014.

[2] P. K. Meher and Y. Pan, "Mcm-based implementation of block fir filters for high-speed and low-power applications," in Proc. VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th Int. Conf., Oct. 2011, pp. 118–121. [7]

[3] L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro, "Design of digit-serial FIR filters: Algorithms, architectures, and a CAD tool," IEEE Trans. Very Large Scale Integration (VLSI) Syst., vol. 21, no. 3, pp. 498–511, Mar. 2013. [8]

[4] P.Prashanti,Dr. B.RajendraNaik,"Design and Implementation of High Speed Carry Select Adder" International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 9- Sep 2013.

[5] Paulchamybalaiah and Dr.IlaVennila,"A Proficient Design of HybridSynchronous and Asynchronous Digital FIR Filter using FPGA", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012 ,ISSN (Online): 1694-0814

[6]M. B. Gately, M. B. Yeary, and C. Y. Tang, "Multiple real-constant multiplication with improved cost model and greedy and optimal searches," in Proc. IEEE ISCAS, May 2012, pp. 588–591. [7]M.Kumm,P.Zipf,Faust,andC.H.Chang"Pipelinedaddergraph for MCM"ieee, may2012.