

A methodology for implementing decimator FIR filters on FPGA



Saliha Harize*, Mohamed Benouaret, Nouredine Doghmane

Badji Mokhtar University, Annaba, Algeria

ARTICLE INFO

Article history:

Received 30 November 2012

Accepted 30 May 2013

Keywords:

Distributed arithmetic

FIR filter

FPGA

Look up table (LUT)

Polyphase structure

ABSTRACT

This paper presents a methodology which can be used to implement any decimator symmetric/antisymmetric (S/A) finite impulse response (FIR) filter. Two varieties are developed: a classic distributed arithmetic (CDA) based and a modified distributed arithmetic (MDA) based one. Both exploit the polyphase structure and the symmetry/antisymmetry of the filter and are evaluated in terms of area efficiency, speed and power consumption. The choice of the algorithm depends on the performance metrics targeted. The methodology has been applied to implement the filter bank CDF9/7 which constitutes a one dimensional (1D) and one level discrete wavelet transform (DWT). The filter bank also known as the bior4.4 biorthogonal wavelets is recommended by the JPEG2000 standard for lossy compression of images and video. The architecture has been implemented on an Altera field programmable gate array (FPGA) and the simulations run in Matlab, Modelsim and Altera Quartus II. The results prove the efficiency of the algorithms and show the tradeoff between the area occupied, the throughput and the power consumption.

© 2013 Elsevier GmbH. All rights reserved.

1. Introduction

There is a growing trend toward the implementation of digital signal processing algorithms such as discrete Fourier transform (DFT), discrete cosine transform (DCT) and DWT on FPGA. The FPGAs have an advantage over digital signal processors (DSPs). Indeed, in some cases, the latter cannot achieve the required throughput performance because of their architecture based on sequential processing while the former have the benefit of concurrent processing as well as the sequential one. The DWT is intensively used in signal, image and video for denoising and compression purposes. The DWT is based on filter banks, in particular FIR filters. This category of filters is also widely applied to a variety of digital signal processing areas for the virtues of providing linear phase and stability. Linear phase is a property where the phase response of the filter is a linear function of the frequency, excluding the possibility of wraps at $\pm\pi$. Since a linear phase filter has constant group delay, all frequency components have equal delay times. That is, there is no distortion due to the time delay of frequencies relative to one another whereas a filter with non-linear phase has a group delay that varies with frequency, resulting in phase distortion. The symmetry/antisymmetry

is a main characteristic of the biorthogonal wavelets known as the “biorx.y” family, heavily used in image and video compression. The operation of a FIR filter is based on a sum of products known as the convolution operation. The performance metrics depend largely on the architecture of the multiplier. Therefore, different techniques are used to implement FIR filters and they are evaluated in terms of area, speed and power consumption. In the literature, a great number of reports are based on distributed arithmetic, a technique which was first introduced by Croisier et al. [1], and further developed by Peled and Liu [2]. In [9] the authors presented a systolic decomposition scheme for DA-based FIR implementation and found that an address length of 4 yields the best area-delay-power-efficient realizations. Zhou and Shi [10] implemented a 31 taps FIR filter with an even symmetry and compared the implementation cost between traditional arithmetic and DA. They reported that DA saves 50 percent hardware resources. In [13], a constant coefficient multiplier was designed to implement a db2-based signal denoising scheme. The input data, used to address the LUT is subdivided into a higher and lower part and an offset is determined to solve the problem of negative addresses corresponding to the most significant half. The idea is very convenient for short filters but becomes complex when dealing with longer filters.

This paper presents a methodology for the implementation of any type of S/AFIR. Based on distributed arithmetic (DA), it gives the designer a variety of techniques depending on the targeted filter performance with an emphasis on the tradeoff area and power for a small latency which is of most importance in real time applications such as live broadcast audio, telephone calls, real time video, etc.

* Corresponding author at: Department of Electronics, Badji Mokhtar University, BP 12, Sidi Ammar, Annaba, Algeria. Tel.: +213 771401352.

E-mail addresses: shrzdj@gmail.com (S. Harize), mohamed.benouaret@gmail.com (M. Benouaret), ndoghmane@univ-annaba.org (N. Doghmane).

In Section 2, basic FIR filters concepts are covered as well as the distributed arithmetic technique. The CDA-based methodology is developed in Section 3, while Section 4 presents the MDA-based algorithm. The different simulations and the results are exposed in Section 5.

2. Background and concepts

2.1. Linear phase

The transfer function of a filter of length $(p + 1)$, i.e., of order p is:

$$H(z) = \sum_{n=0}^P h_n z^{-n} \quad (1)$$

Such a filter has a linear phase if its impulse response is either symmetric: $h_n = h_{(p-n)}$ for $0 \leq n \leq p$ or is antisymmetric: $h_n = -h_{(p-n)}$ for $0 \leq n \leq p$. Since the filter impulse response can be either even or odd, four types of linear phase FIR filters can be defined:

- Symmetric FIR filters of even/odd length.
- Antisymmetric FIR filters of even/odd length.

As the proposed methodology will be applied to the CDF9/7 filter bank, the group delays are calculated. For the Low pass 9 taps filter, using Eq. (1) with $P=8$ and exploiting the symmetry of the filter, we end up with $H(z)$ written as:

$$H(z) = z^{-4} [h_0(z^4 + z^{-4}) + h_1(z^3 + z^{-3}) + h_2(z^2 + z^{-2}) + h_3(z + z^{-1}) + h_4] \quad (2)$$

The corresponding frequency response, using normalized frequencies, is then given by:

$$H(e^{j\omega}) = e^{-j4\omega} [2h_0 \cos(4\omega) + 2h_1 \cos(3\omega) + 2h_2 \cos(2\omega) + 2h_3 \cos(\omega) + h_4] \quad (3)$$

The standard measure of the phase linearity of a system is the group delay defined by:

$$\tau(\omega) = \frac{-d\phi(\omega)}{d\omega} \quad (4)$$

ω being the normalized pulsation ($\omega = 2\pi f$) and $\phi(\omega)$ being the phase.

The phase function of the 9 taps filter is $\phi(\omega) = -4\omega + \theta$ (θ is either 0 or π) and it is a linear function of ω . The group delay is $\tau(\omega) = -4$ which indicates a constant group delay of 4 samples.

The same calculation for the high pass 7 taps filter gives a constant group delay of 3 samples.

2.2. Polyphase structure

The general structure of a symmetric FIR filter is described by Eq. (1), known as the convolution product:

$$y[n] = \begin{cases} \sum_{i=0}^n h_i x[n-i] & n \leq P \\ \sum_{i=0}^P h_i x[n-i] & n > P \end{cases} \quad (5)$$

where $h_0, h_1, \dots, h_P$ represent the filter's $(P + 1)$ coefficients and $x[n]$ the data sequence to be filtered. P characterizes the length of the impulse response of the digital filter. N is the total number of data samples to be processed.

An approach for implementing FIR decimator filters or the convolution operation can be realized with either a non-polyphase structure corresponding to Eq. (1) or a polyphase structure [6,7]. A polyphase implementation of an FIR decimator splits the FIR filter impulse response into M different subfilters, where M is the downsampling, or decimation factor. The key to the efficiency of polyphase filtering is that specific input values are only multiplied by selected values of the impulse response. For $M=2$, the input values $x[0], x[2], x[4], \dots$ are only combined with the filter coefficients $h_0, h_2, h_4, \dots$, and the input values $x[1], x[3], x[5], \dots$ are only combined with the filter coefficients $h_1, h_3, h_5, \dots$.

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + h_4 z^{-4} + h_5 z^{-5} + \dots + h_P z^{-P} \quad (6)$$

For an odd filter length (p even):

$$H(z) = (h_0 z^{-2} + h_2 z^{-2} + h_4 z^{-4} + \dots + h_P z^{-P}) + (h_1 z^{-1} + h_3 z^{-3} + h_5 z^{-5} + \dots + h_{(p-1)} z^{-(p-1)}) \quad (7)$$

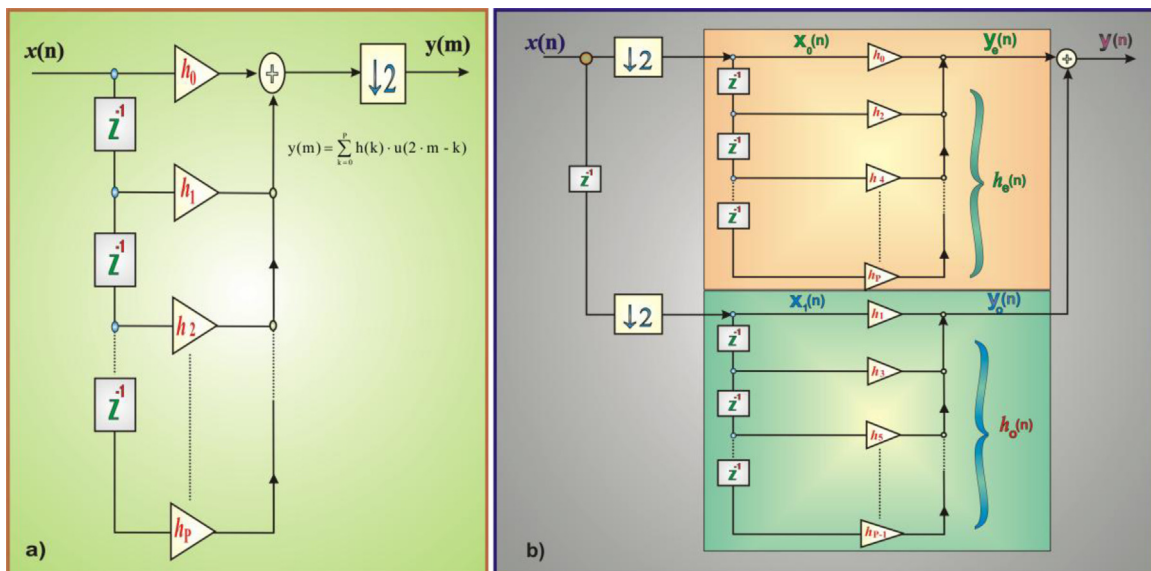


Fig. 1. Decimator: a transversal filter structure followed by a downsampler (a), and an efficient polyphase filter configuration (b).

Download English Version:

<https://daneshyari.com/en/article/446579>

Download Persian Version:

<https://daneshyari.com/article/446579>

[Daneshyari.com](https://daneshyari.com)