Microprocessors and Microsystems 61 (2018) 117–125

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Compact FPGA architectures for the two-band fast discrete Hartley transform

Lampros Pyrgas[a], Paris Kitsos[a,*], Athanassios Skodras[b]

[a] Computer and Informatics Engineering Department, Technological Educational Institute of Western Greece, National Road Antirrion - Ioannina GR-30020, Greece
[b] Electrical and Computer Engineering Department, University of Patras, Patras, Greece

ARTICLE INFO

Keywords:
Two-band fast discrete hartley transform
FPGA architecture
Digital signal processing
VHDL

ABSTRACT

The discrete Hartley transform is a real valued transform similar to the complex Fourier transform that finds numerous applications in a variety of fields including pattern recognition and signal and image processing. In this paper, we propose and study two compact and versatile hardware architectures for the computation of the 8-point, 16-point and 32-point Two-Band Fast Discrete Hartley Transform. These highly modular architectures have a symmetric and regular structure consisting of two blocks, a multiplication block and an addition/subtraction block. The first architecture utilizes 8 multipliers and 16 adders/subtractors, achieving a maximum clock frequency of 95 MHz. The second architecture utilizes only 4 multipliers and 8 adders/subtractors, achieving a maximum clock frequency of 100 MHz; however it requires additional multiplexers and more clock cycles (from 1 to 58 clock cycles depends on the points) for the computation. As a result, the proposed hardware architectures constitute an efficient choice for area-restricted applications such as embedded or pervasive computing systems.

1. Introduction

The discrete Hartley transform (DHT) was introduced in 1983 [1] and the first fast DHT in 1984 [2]. Since then, continuous attempts are made in reducing its computational complexity and thus increasing the speed of computation in software and hardware [3–18]. DHT's popularity is due to its real-valued and symmetric transform kernel that is identical to that of its inverse (self-inverse or involuntary), up to a scale factor of 1/N. The DHT, as a real-valued alternative to the discrete Fourier transform (DFT), leads to more efficient computation of the DFT, as well as other widely used unitary transforms like discrete cosine and sine transforms. The DHT finds a number of signal and image processing applications, as for example circular convolution and interpolation, error control coding, adaptive filtering, multi-carrier modulation, image classification, image encryption or image compression.

The computation of the DHT is intensive, requiring the repetitive calculation of products between sample values and cosine / sine coefficients. Numerous architectures have been proposed for its hardware computation, which fall in one of the following categories: (a) Memory-based, where look-up tables (LUT) or distributed arithmetic (DA) is used for product calculation; (b) Systolic-arrays, where special processing elements (PE) are developed for the computation of the products, given that the whole structure is highly regular and modular [13,16]; (d) DSP-based, where the multiply-accumulate (MAC), the Harvard architecture and the pipeline benefits of the digital signal processors are exploited [6]; (c) FPGA, where the flexibility and the rapidness of the gate-level design are made use of [5,7].

In the present work a compact FPGA architecture is proposed for the computation of the two-band fast DHT, which recently appeared in the open literature [14]. This two-band fast DHT algorithm is based on a simple decomposition of the input data in two bands (high and low), thus resulting in a very regular structure, avoiding the input or output data shuffling, and eventually requiring slightly less multiplications than the existing approaches, at the expense of an increase in the number of additions. This versatile design supports the computation of 8-, 16- and 32-point fast DHT's. Input samples and output (transform) coefficients are communicated in serial mode. Recently, the authors have proposed also a two-band fast DHT architecture for FPGA implementation that supports the computation of 64 points [19]. The architectures in the present work are more compact and support different computation lengths.

The paper is organized as follows. In Section 2 the two-band fast DHT algorithm is described in brief. The proposed architecture and hardware implementation are presented in detail in Section 3 and the synthesis and FPGA implementation results are given in Section 4.

* Corresponding author.
E-mail address: pkitsos@teimes.gr (P. Kitsos).

https://doi.org/10.1016/j.micpro.2018.06.002
Received 24 January 2018; Received in revised form 22 April 2018; Accepted 4 June 2018
Available online 05 June 2018

Section 5 concludes the paper.

2. Two-band fast DHT algorithm

The DHT is a linear real-to-real transform with identical the forward and the inverse DHT transforms. The DHT transform of a N real-valued input data *x(0), x(1), ..., x(N-1)* is defined as:

$$H_N(k) = \sum_{n=0}^{N-1} x(n) cas\left(\frac{2\pi}{\mathrm{N}} nk\right) \tag{1}$$

where $k = 0, 1, 2, \ldots, N-1$, and

$$cas(\cdot) = cos(\cdot) + sin(\cdot) \tag{2}$$

The Two-Band Fast DHT algorithm that is presented in [14] is based on the decomposition of the input data into low-band $x_L(n)$ and high-band $x_H(n)$ values:

$$x_L(n) = \frac{1}{2}[x(2n) + x(2n+1)] \tag{3}$$

$$x_H(n) = \frac{1}{2}[x(2n) - x(2n+1)] \tag{4}$$

or

$$x(2n) = x_L(n) + x_H(n) \tag{5}$$

$$x(2n+1) = x_L(n) - x_H(n) \tag{6}$$

After the two-band decomposition, the core DHT transform is applied on the summations and the differences on the low-band and high-band values of adjacent samples. This, based also on the properties of the trigonometric function cas(•), leads to [14]:

$$H_N(k) = H_L(k) + [\cos\theta\, H_H(k) + \sin\theta\, H_H(-k)] \tag{7}$$

where

$$\theta = 2\pi\kappa/\mathrm{N} \tag{8}$$

$$\mathrm{H}_L(k) = H^L_{N/2}(k) + H^H_{N/2}(k) \tag{9}$$

$$\mathrm{H}_H(k) = H^L_{N/2}(k) - H^H_{N/2}(k) \tag{10}$$

and

$$H^L_{N/2}(k) = \sum_{n=0}^{N/2-1} x_L(n) cas\left(\frac{2\pi\kappa}{\mathrm{N}/2} n\right) \tag{11}$$

$$H^H_{N/2}(k) = \sum_{n=0}^{N/2-1} x_H(n) cas\left(\frac{2\pi\kappa}{\mathrm{N}/2} n\right) \tag{12}$$

The DHT coefficients of (7) that are N/2 positions apart from k are given by:

$$H_N\left(k + \tfrac{N}{2}\right) = H_L(k) - [\cos\theta\, H_H(k) + \sin\theta\, H_H(-k)] \tag{13}$$

Finally, given that

$$H_H(-k) = H_H\left(\frac{\mathrm{N}}{2} - k\right)\Bigg] \tag{14}$$

the N-point DHT becomes:

$$H_N(k) = H_L(k) + \left[\cos\theta\, H_H(k) + \sin\theta\, H_H\left(\frac{\mathrm{N}}{2} - k\right)\right] \tag{15}$$

$$H_N\left(k + \frac{N}{2}\right) = H_L(k) - \left[\cos\theta\, H_H(k) + \sin\theta\, H_H\left(\frac{\mathrm{N}}{2} - k\right)\right] \tag{16}$$

where $k = 0,\ 1,\ \ldots,\ \frac{N}{2} - 1$.

The 32-point Two-Band Fast DHT flow graph is shown in Fig. 1, with the 16-point and 8-point Two-Band Fast DHTs highlighted [14]. It is composed of two 16-point Two-Band Fast DHTs, combined with highly symmetrical butterflies and 28 multiplications by twiddle factors. There are butterflies that need only addition or subtractions for their computations. Their output nodes are depicted by solid (filled) or empty cycles respectively. There exist butterflies that need multiplications by cosines before the addition and subtraction for their computations. Their output nodes are depicted by solid (filled) rectangles or empty rectangles, respectively, and the symbols c_i, c_j above and below of each rectangle. This means that the two inputs to the node should be multiplied by c_i and c_j respectively before they are added or subtracted.

3. Proposed hardware architectures

Due to the very regular and modular structure of the algorithm, a 32-point Two-Band Fast DHT architecture can also be used for 16- and 8-point Two-Band Fast DHT computations. Two versatile 32-point Two-Band Fast DHT architectures (that also support 16- and 8-point computations) are proposed in this paper. They constitute two alternatives of a compact implementation of the flow graph of Fig. 1. In both our designs we have chosen the representative case of 8-bit signed inputs (the input values ranging from -127 to 128), which lead to 13-bit signed outputs. Output values, for the 32-point Two-Band Fast DHT, range from -4064 (32 times the minimum input value) to 4096 (32 times the maximum input value).

3.1. Architecture 1: 16 adders and 8 multipliers

The proposed architecture 1, denoted as arch1 in Table 2 below, uses a total of 16 adders and subtractors in order to perform the computations in the butterfly stages and 8 signed multipliers, followed by 4 adders/subtractors, for the computations in the multiplication stages. This hardware architecture is shown in Fig. 2.

The two operations (16- and 8-point computations) are controlled by a 2-bit mode signal. Based on that, all the necessary signals for the correct flow of computation for the required N-point DHT are generated by a control unit that mainly consists of a Finite State Machine (FSM). By selecting 8- or 16-point DHT computation, only the operations (additions / subtractions / multiplications) that are included in the corresponding dashed boxes of Fig. 1 are performed, while all other operations before or after, are skipped. The computation mainly consists of two types of stages: the butterfly stages and the multiplication stages.

The flow of computation of the architecture is as follows: In the first stage, and for a number of clock cycles equal to the number of points, the input data points are stored one by one in the register file. During the main computation stage, the input data are first driven to the adders/subtractors block in groups, in order for the adders and subtractors to perform the required stages of butterflies as shown in Fig. 1. The outputs of the butterflies are 13 bits long in order to keep full accuracy. A shift block follows right after the adders/subtractors block. Based on the selection signal from the control unit, the results are either pass unaffected or they are right-shifted by one, in order for a division by two to be performed. The data are then driven to the appropriate registers depending on the signals from the control unit block.

This process continues, until all the necessary butterfly stages are executed. After the butterfly stages are completed, a stage of multiplications follows. The necessary data points are driven to the multipliers block in groups, and after the computations, the results are stored back in the same registers. This continues, until all the necessary multiplication stages are executed. As shown in Fig. 1, stages of butterflies and multiplications are following one another, based on the selected mode of computation, until all the Hartley coefficients are finally computed.

Two ADD_M_MUX multiplexers (muxes) are used for routing the appropriate data from either the adders/subtractors block or the multipliers block to the register file. Specifically, these muxes drive the