

AN EFFICIENT DESIGN APPROACH FOR A PRIME-LENGTH GENERALIZED HARTLEY TRANSFORM SYSTOLIC ARRAY

D.F. Chiper, M.N.S. Swamy, and M.O. Ahmad

Center for Signal Processing and Communications
Dept. of Electrical and Computer Engineering
Concordia University
Montreal, Quebec, Canada H3G 1M8
[chiper,swamy,omair]@ece.concordia.ca

ABSTRACT

An efficient approach to the design of a VLSI array for a prime-length type III generalized discrete Hartley transform (GDHT) is presented. The design makes use of an appropriate decomposition of the GDHT into two half-length circular correlation structures having the same length and form that can be concurrently computed and implemented on a single linear systolic array using some of the known hardware sharing techniques. As a consequence, a substantial increase in computational speed with a simplified control structure and low hardware complexity can be achieved, while preserving all the advantages, such as the low I/O and hardware costs, of the circular correlation based VLSI implementations.

1. INTRODUCTION

In digital signal processing, discrete Hartley transform (DHT) has gained an increased popularity over the recent years for the processing of real sequences. Its generalized versions, known as the generalized Hartley transforms (GDHT) [1], have proved to be useful in digital filter banks, signal representation, and in the computation of 2D-DHT and various kinds of convolutions. One of the most useful forms of the GDHT is the type-III GDHT.

Due to the fact that the DHT and GDHT are both computationally intensive, it is necessary to

design efficient, dedicated, VLSI-based hardware structures for real-time applications. The data flow plays a crucial role in determining the efficiency of a VLSI implementation of a DSP algorithm. The use of regular and modular computational structures, as those of cyclic convolution and circular correlation, have proved to be an efficient solution for the VLSI implementation of various discrete transforms providing certain advantages, for example, lower I/O and hardware costs [2], [3]. In such a case, the efficiency of the VLSI implementation can be further improved by employing modular and regular hardware structures of distributed arithmetic [5] and systolic arrays [6].

There are several software solutions to implement GDHT but until now no efficient hardware algorithm seems to have been proposed. In this paper, an efficient design approach for a VLSI implementation of a prime-length type-III GDHT using circular correlation structures is presented. It uses an efficient hardware algorithm based on an appropriate decomposition method of the GDHT into two half-length circular correlation structures having the same length and structure.

2. PROPOSED ALGORITHM FOR GDHT

The type-III Generalized Discrete Hartley Transform (GDHT) of the input sequence $\{x(i) : i = 0, \dots, N - 1\}$ is defined as [1]:

$$Y_{III}(k) = \sum_{i=0}^{N-1} x(i) \cdot \text{cas}[(2k+1)i\alpha] \quad (1)$$

where, $\alpha = \frac{\pi}{N}$, and $\text{cas}(\theta) = \cos(\theta) + \sin(\theta)$.

If $N > 2$ is a prime number, we can compute the type-III GDHT using the following equations:

$$Y_{III}(0) = x(0) + H_C(0) + H_S(0) \quad (2)$$

$$Y_{III}(k) = x(0) + H_C(k) + H_S(k) \quad (3)$$

$$Y_{III}(N-k) = x(0) + H_C(k-1) - H_S(k-1) \quad (4)$$

$$k = 1, \dots, (N-1)/2.$$

The auxiliary output sequences

$$\begin{aligned} \{H_C(k) : k = 1, \dots, (N-1)/2\} \\ \{H_S(k) : k = 1, \dots, (N-1)/2\} \end{aligned}$$

can be computed recursively as

$$H_C(k) = 2 \cdot T_C(k) - H_C(k-1) \quad (5)$$

$$H_S(k) = 2 \cdot T_S(k) + H_S(k-1) \quad (6)$$

$$k = 1, \dots, (N-1)/2.$$

with

$$H_C(0) = \sum_{i=1}^{(N-1)/2} [x_C(\psi(i)) - x_C(\phi(i))] \quad (7)$$

$$H_S(0) = \sum_{i=1}^{(N-1)/2} [x_S(\psi(i)) - x_S(\phi(i))] \quad (8)$$

where

$$x_C(i) = x(\zeta(i)) \cdot \cos[2i\alpha] \quad (9)$$

$$x_S(i) = x(\zeta(i)) \cdot \sin[2i\alpha] \quad (10)$$

$$k = 1, \dots, (N-1)/2$$

In order to obtain the circular correlation structures, we first re-order the input and output samples using the following index mappings

$$\psi(k) = \begin{cases} \varphi(k) & \text{if } \varphi(k) \leq (N-1)/2 \\ \varphi(k + (N-1)/2) & \text{otherwise} \end{cases} \quad (11)$$

$$\phi(k) = \begin{cases} \varphi(k) & \text{if } \varphi(k) > (N-1)/2 \\ \varphi(k + (N-1)/2) & \text{otherwise} \end{cases} \quad (12)$$

$$\varphi(k) = \langle g^k \rangle_N \quad (13)$$

$$\zeta(k) = \langle 2k \rangle_N \quad (14)$$

where g represents the primitive root of the Galois field of the indices, and $\langle x \rangle_N$ represents x modulo N .

Now, we can decompose the computation of the GDHT into two circular correlations defined by

$$T_C(\psi(k)) = \sum_{i=1}^{(N-1)/2} [x_C(\psi(i)) - x_C(\phi(i))] \cdot \cos[\psi(i+k) \cdot 4\alpha] \quad (15)$$

$$T_S(\psi(k)) = \sum_{i=1}^{(N-1)/2} [x_S(\psi(i)) - x_S(\phi(i))] \cdot \cos[\psi(i+k) \cdot 4\alpha] \quad (16)$$

$$k = 1, \dots, (N-1)/2$$

Equations (15) and (16) represent two half-length circular correlations having the same length and structure and can be concurrently computed. Due to the fact that they have the same form and length, we can obtain a significant hardware reduction by using hardware sharing techniques.

3. AN EXAMPLE

In order to illustrate the proposed approach, we use an example with length $N=7$ and the primitive root $g=3$. In this case, we can compute the two circular correlations of length 3 giving (15) and (16) in the form

$$\begin{bmatrix} T_C(3) \\ T_C(2) \\ T_C(1) \end{bmatrix} = \begin{bmatrix} \cos(4a) & \cos(2a) & \cos(6a) \\ \cos(2a) & \cos(6a) & \cos(4a) \\ \cos(6a) & \cos(4a) & \cos(2a) \end{bmatrix} \cdot \begin{bmatrix} x(6) \cdot \cos(3a) - x(1) \cdot \cos(4a) \\ x(4) \cdot \cos(2a) - x(3) \cdot \cos(5a) \\ x(2) \cdot \cos(a) - x(5) \cdot \cos(6a) \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} T_S(3) \\ T_S(2) \\ T_S(1) \end{bmatrix} = \begin{bmatrix} \cos(4a) & \cos(2a) & \cos(6a) \\ \cos(2a) & \cos(6a) & \cos(4a) \\ \cos(6a) & \cos(4a) & \cos(2a) \end{bmatrix} \cdot \begin{bmatrix} x(6) \cdot \sin(3a) - x(1) \cdot \sin(4a) \\ x(4) \cdot \sin(2a) - x(3) \cdot \sin(5a) \\ x(2) \cdot \sin(a) - x(5) \cdot \sin(6a) \end{bmatrix} \quad (18)$$

where $a = 2\alpha$. We can recursively compute the samples of the auxiliary output sequences

$$\begin{aligned} \{H_C(k) : k = 1, \dots, (N-1)/2\} \\ \{H_S(k) : k = 1, \dots, (N-1)/2\}, \end{aligned}$$

and finally, the output sequence using (2)-(4) as

$$\begin{bmatrix} Y_{III}(0) \\ Y_{III}(1) \\ Y_{III}(2) \\ Y_{III}(3) \end{bmatrix} = \begin{bmatrix} x(0) + H_C(0) + H_S(0) \\ x(0) + H_C(1) + H_S(1) \\ x(0) + H_C(2) + H_S(2) \\ x(0) + H_C(3) + H_S(3) \end{bmatrix} \quad (19)$$

$$\begin{bmatrix} Y_{III}(6) \\ Y_{III}(5) \\ Y_{III}(4) \end{bmatrix} = \begin{bmatrix} x(0) + H_C(0) - H_S(0) \\ x(0) + H_C(1) - H_S(1) \\ x(0) + H_C(2) - H_S(2) \end{bmatrix} \quad (20)$$

4. HARDWARE IMPLEMENTATION

Using the dependence-graph based design procedure [7] and the tag control scheme [8] to implement the control structure, we can obtain two linear systolic arrays corresponding to (15) and (16), having the same structure and length. As a consequence, a substantial reduction in the hardware complexity can be achieved using the hardware sharing techniques as employed for the systolic arrays in [9]. We can thus obtain, for the example considered in the previous section, a linear systolic array as shown in Fig. 1, where the following notations have been used:

$$\begin{aligned} xc61 &= x(6) \cdot \cos(3a) - x(1) \cdot \cos(4a); xc42 = x(4) \cdot \cos(2a) - \\ &x(3) \cdot \cos(5a); xc25 = x(2) \cdot \cos(a) - x(5) \cdot \cos(6a); xs61 = \\ &x(6) \cdot \sin(3a) - x(1) \cdot \sin(4a); xs43 = x(4) \cdot \sin(2a) - x(3) \cdot \\ &\sin(5a); xs25 = x(2) \cdot \sin(a) - x(5) \cdot \sin(6a); \end{aligned}$$

It can be easily seen from Fig. 1, that all the I/O channels are kept at the two ends of the array and the number of I/O channels and their bandwidths are independent of the transform length N . Also, the structure of the systolic array does not depend on N .

Apart from the hardware core presented in Fig. 1, the proposed VLSI array uses pre-processing and post-processing stages that are necessary to perform the required conversions. The pre-processing stage is used to reorder and compute the desired input sequence for the systolic array, as can be seen from (17) and (18). The post-processing stage is used to permute the output samples of the systolic array, to recursively compute the auxiliary sequences in parallel using an addition/subtraction module and a latch, and finally, to obtain the output sequence using some additional modules of addition/subtraction. These stages represent a hardware overhead, but the overhead is small and independent of the transform length.

It is noted that the VLSI architecture proposed in this paper has a higher throughput with a hardware complexity similar to that of systolic array based VLSI implementation for DCT given in [3]. Further, it has almost the same performance and hardware complexity as that of a systolic array based implementation of DCT and DST [4]. It is important to note that this hardware complexity and performance is almost the same as those of DCT and DST realizations, in spite of the number

of operations involved in the definition of type-III GDHT being twice that of DCT and DST.

5. CONCLUSION

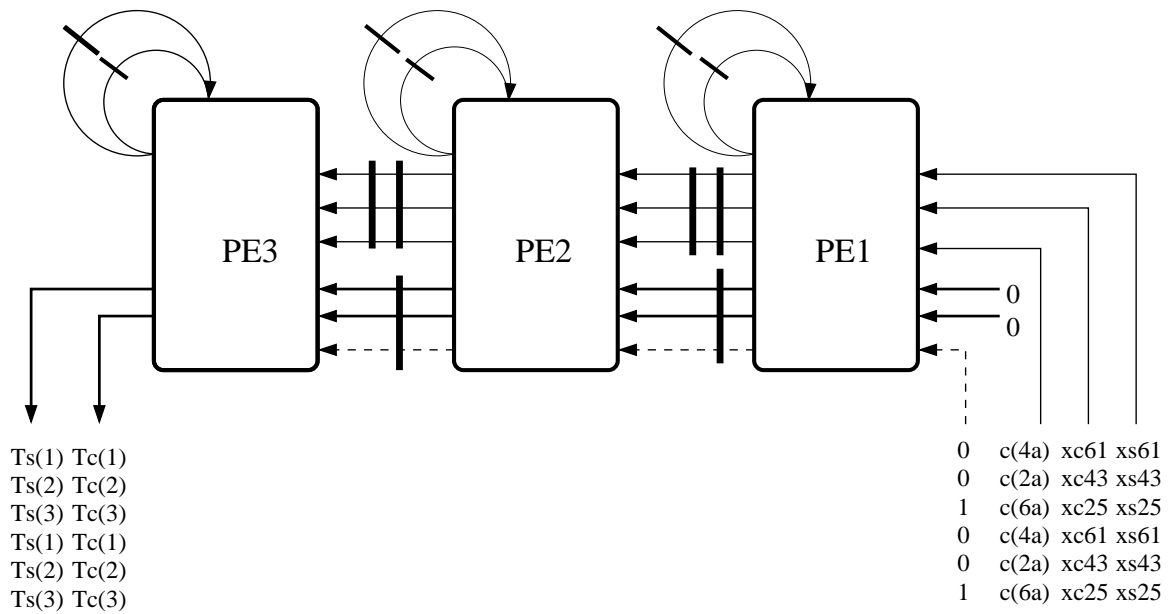
A new approach to derive a systolic algorithm and architecture for a prime length type-III GDHT has been presented. This approach has led to a VLSI architecture with an increased speed of operation, reduced hardware complexity and I/O costs, and a simplified control structure. Further, this architecture preserves all the advantages of the circular correlation and cyclic convolution-based VLSI implementations. The approach presented in this paper in conjunction with the work contained in [4] represents an important step towards a unified VLSI implementation of DCT, DST and DHT.

6. ACKNOWLEDGMENT

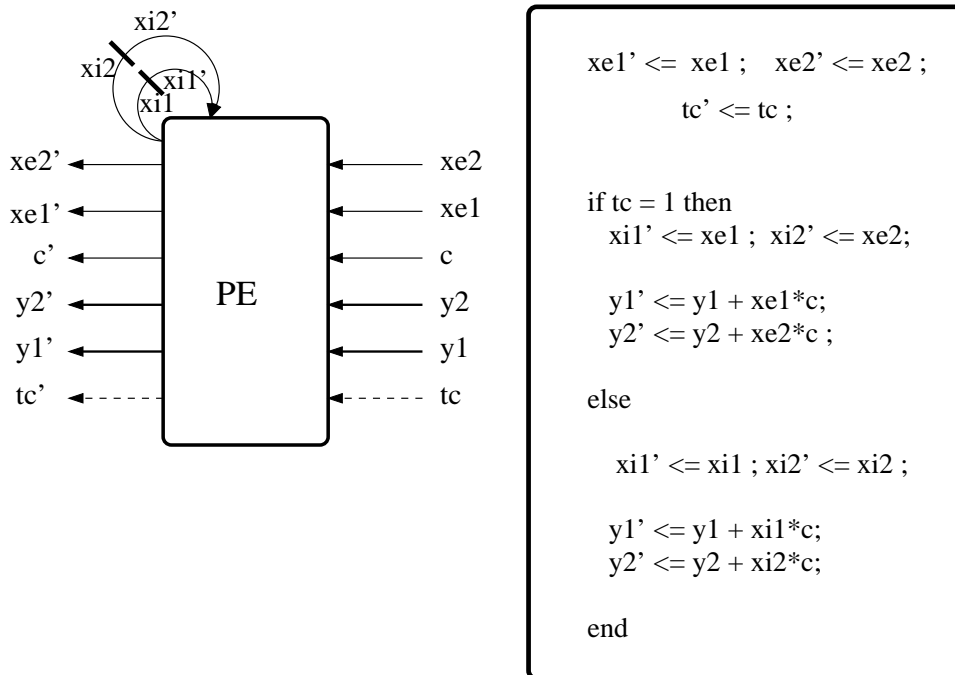
This work was partially supported by the Micronet Network of Centres of Excellence, the Natural Sciences and Engineering Council (NSERC) of Canada, and Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR) of Quebec.

7. REFERENCES

- [1] N.C.Hu, H.I. Chang, O.Ersoy, "Generalized Discrete hartley Transform," *IEEE Trans. Signal Processing*, pp. 2931-2940, vol. 40, no. 6, June 1992.
- [2] C.M.Rader, "Discrete Fourier Transform when the number of the data samples is prime," *Proc. IEEE*, pp. 1107-1108, 56, 1968.
- [3] J. Guo, C.M. Liu, and C.W. Jen, "A New Array Architecture for Prime-Length Discrete Cosine Transform," *IEEE Trans. on Signal Processing*, pp.436-442, vol. 41, no.1, Jan. 1993.
- [4] D.F. Chiper, "A new unified systolic array algorithm for Discrete cosine and sine transforms with improved performances," in *Proc. IEEE Int. Symp. Circ. Syst. ISCAS'97*, pp. 2589-2592, 1997.
- [5] S.A. White, "Applications of the distributed arithmetic to digital signal processing: A tutorial," *IEEE ASSP Magazine*, vol. 6, July 1989.
- [6] H. T. Kung, "Why Systolic Architectures," *Computer*, 15, Jan. 1982.
- [7] S. Y. Kung, "*VLSI Array Processors*," Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- [8] C.W. Jen and H. Y. Hsu, "The design of systolic arrays with tag input," in *Proc. IEEE Int. Symp. Circ. Syst. ISCAS'88*, pp. 2263-2266, 1988.
- [9] D.F. Chiper, "Parallel structures used in Digital Signal Processing," PhD Dissertation, TUI, 1996



(a)



(b)

Figure 1: The linear systolic array of the core hardware of the proposed VLSI array for GDHT.