4th WSEAS International Conference on ELECTRONICS, CONTROL and SIGNAL PROCESSING, Miami, Florida, USA, 17-19 November, 2005 (pp.267-270)

An efficient Radix-two Algorithm to Compute the 2D Fourier Transform

D. CHIKOUCHE, N. AMARDJIA, R. E. BEKKA
Electronics Department, Faculty of Engineering
University of Setif
19000 Setif
ALGERIA
http: //www.univ-setif.dz/cige04.htm

Abstract: - In this paper, we propose a new approach for computing 2D FFT's that are suitable for implementation on a systolic array architectures. Our algorithm is derived in this paper from a Cooley decimation-in-time algorithm by using an appropriate indexing process. It is proved that the number of multiplications necessary to compute our proposed algorithm is significantly reduced while the number of additions remains almost identical to that of conventional 2D FFT's. Comparison results shows the powerful performance of the new 2D FFT algorithm against the row-column FFT transform

Key-Words: - New Approach, Two-Dimensional, Fourier Transform, Decimation-in-Time, Algorithm, Performance

1 Introduction

In recent years there has been a growing interest regarding the development of efficient computational algorithms for the discrete Fourier transform [1-11]. These algorithms must be capable of matching the advantages offered by the high speed digital computer and the rapid advances in *VLSI* technology.

The computation of 2D Fourier transform (multi-dim.*FT*) is an interesting and challenging problem. At the present, the2D.*FFT* finds its practical significance in tomography, image processing, computer vision and in nuclear magnetic response imaging [3]. It is therefore a powerful tool for analyzing and providing a better means of understanding 2D signals in the "*frequency space*". However, the 2D *FT* requires a high amount of computations which motivates us to search for "*efficient*" algorithms [4].

The evaluation of the 2D *DFT* is based on three widely used classes of *FFTs*. There are the row-column, the vector radix and the polynomial transform *FFT* [4,5]. We have proposed a new fast algorithm for the 2D *DFT*, in a previous work [11], which is presented in a simple matrix form that allows straight forward *VLSI* implementation.

In this paper, we present a radix-2 fast algorithm for the computation of the 2D *DFT* that is based on the same ideas of [11]. We will analyze the computational complexity and relations of our new algorithm against well-known 2D *FFT* conventional algorithms.

2 Proposed Algorithm for the 2D Discrete Fourier Transform

In the following sections, we will present a fast algorithm that is developed for computing the discrete Fourier transform of a two-dimensional data set with N points along each array, where N is an arbitrary integer. The usual method of computing this *DFT* (*N,2*) is by performing *2N* distinct *1D DFT* (*N,1*) computations [1]. An algorithm based on new ideas of reference [11] has been constructed. We will show that the new algorithm will have a butterfly structure. We also give a count of the number of arithmetic operations which this algorithm uses and compare it with that of traditional methods.

The two-dimensional *DFT* transform (*2D DFT*) of $x(k_1,,k_2)$ is defined as:

$$X(n_1,n_2) = \sum_{k_1=0}^{N-1} \cdot \sum_{k_2=0}^{N-1} x(k_1,k_2) W_N^{\sum_{j=1}^{2} n_j k_j} \tag{1}$$

where $n_j \in [0,N-1]$ and $W_N = exp(-j2\pi/N)$
or in a matrix form as [4]:

$$X = W_N^2 \, x \tag{2}$$

The basic matrix W_N^2 $(N^2 \times N^2)$ is generated by a Kronecker product of the matrix W_N [6]

$$W_N^2 = W_N \otimes W_N \qquad (3)$$

The direct computation of N^2 points *2D DFT* of equation *(2)* requires: N^4 complex multiplications, $N^2(N^2-1)$ complex additions and $2N^2$ loads and stores. The usual methods used to reduce this amount of computations are row-column methods [4,7,8].

2.1 Traditional method for the computation of the 2D DFT

The usual way to compute this *2D DFT N* points is by performing the computation of *2N* distinct *1D DFT N* points [4,8]. By applying the separability principle to equation (1), we get the equation that define the traditional method of computing the 2D DFT.

$$\begin{aligned} X_1(n_1,k_2) &= \sum_{k_1=0}^{N-1} x(k_1,k_2) W_N^{n_1 k_1} \\ X_2(n_1,n_2) &= \sum_{k_2=0}^{N-1} X_1(n_1,k_2) W_N^{n_2 k_2} \end{aligned} \qquad (4)$$

This method calls for *2* equations, each of which can be done with *N 1D DFTs*. Thus, the total number of *1D DFTs* necessary to compute the entire *2D DFT* is *2N* and the total number of complex operations is: $O_2 = 2N\ O_u$, where O_u is the number of complex operations required to compute a *1D DFT*. If a radix-2 1D *FFT* is used, then the number of complex multiplications necessary for the entire *2D DFT* is $N^2(\log_2 N - 1)$ and the number of complex additions is $2N^2 \log_2 N$.

We will see later that our constructed algorithm can actually reduce significantly this considerable amount of computations.

2.2 The proposed radix-2 2D FFT

The same ideas of reference [11] are used in this paper to derive this radix-2 proposed algorithm. The proposed algorithm combines the advantages of the Cooley-Tukey method, the Kronecker product and an efficient indexing process to give an optimal 2D FFT algorithm expressed in a simple matrix form. The recursive equation for this radix-2 decimation-in-time two-dimensional algorithm is:

$$V_i = W_2^2 D_2^2 V_{i-1} \qquad (5)$$

where

- $V_i(m) = V_i(m_1, m_2) = X_i(p_1 + m_1 2^{r-i}, p_2 + m_2 2^{r-i})$
- $V_{i-1}(k) = V_{i-1}(k_1, k_2) = X_{i-1}(p_1 + k_1 2^{r-i}, p_2 + k_2 2^{r-i})$
- D_2^2 is an $(2^2 \times 2^2)$ diagonal matrix whose elements are given by:

$$D_{2^2}(k,k) = W_N^{\sum_{j=1}^{2} k_j C_{i-2}(P_j)} \qquad (6)$$

or,

$$D_{2^2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & W_N^{C_{i-2}(p_2)} & 0 & 0 \\ 0 & 0 & W_N^{C_{i-2}(p_1)} & 0 \\ 0 & 0 & 0 & W_N^{C_{i-2}(p_1)+C_{i-2}(p_2)} \end{bmatrix} \qquad (7)$$

- $N=2^r$; $i=1$ to r; $p_j=0$ to $N-1$ except p_j+2^{r-i}, $j=1$ to 2.
- $p_j=(p_j)_{r-1} \ldots\ldots (p_j)_{r-i} \ldots\ldots (p_j)_0$, $m=m_1m_2$, $k=k_1k_2$, $(m,k) \in [0,2^2-1]$
- The digits $(p_j)_i$, m_j and k_j take the values $0, 1$.
- $C_{i-2}(P_j) = \left[\sum_{h=0}^{i-2} 2^h (p_j)_{r-1-h}\right] 2^{r-i}, \quad C_{-1}(p_j) = 0$
- The matrix W_2^2 is obtained from equation (12) by replacing B with 2:

$$W_{2^2} = \prod_{j=1}^{2} \left(I_{2^{j-1}} \otimes W_2 \otimes I_{2^{2-j}} \right) \qquad (8)$$

This matrix is known as Hadamard matrix of dimension 2^2 and is denoted H_2^2, so we have:

$$H_{2^2} = \prod_{j=1}^{2} \left(I_{2^{j-1}} \otimes H_2 \otimes I_{2^{2-j}} \right) \qquad (9)$$

with:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Thus,

$$H_{2^2} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \qquad (10)$$

This matrix; subject of many investigations [9], finds also its importance as a basic matrix for our new radix-2 *2D FFT* algorithm. We can easily see that the matrix of equation (10) consists only of zeros and ± 1 elements. So, only complex additions are introduced by this matrix in the computation of equation (7).

Therefore, the total number of operations necessary to perform our radix-2 *2D FFT* is:

$$2N^2 \log_2 N \text{ complex additions}$$

and

$$\frac{(2^2-1)}{2^2} N^2(\log_2 N - 1) = \frac{3}{4} N^2 (\log_2 N - 1)$$

complex multiplications

3 Comparison Between the New 2D FFT Algorithm and Traditional Algorithms

The main critirium that can be used to compare between *2D FFT* algorithms is the computation speed which is strongly dependent on the number of operations involved in each algorithm. Table 1 presents a comparison between the *2D DFT*, the traditional *2D FFT* and the new *2D FFT* in the sens of number of operations involved and when we transform an two-dimensional data set with *N* points along each array.

The new *2D FFT* has properties such that the number of multiplications necessary to compute the *2D DFT* is significantly reduced while the number of additions, for most cases, remains at the same level as traditional methods.

Table 1 Comparison between the proposed *2D FFT* algorithm and traditional algorithms.

	2D DFT	Traditional method	Proposed algorithm
		1D radix-2 FFT	radix-2 2D FFT
Number of Complex additions	$N^2(N^2-1)$	$2N^2\log_2N$	$2N^2\log_2N$
Number of complex multiplications	N^4	$N^2 \log_2 \frac{N}{2}$	$\frac{3}{4} N^2(\log_2 N - 1)$

4 Conclusion

A new radix-2 algorithm for computing two-dimensional decimation-in-time DFT's has been proposed, and its advantages relative to the standard row-column FFT algorithms has been demonstrated. The proposed algorithm combines the advantages of the Cooley-Tukey method, the Kronecker product and an efficient indexing process to give an optimal 2D FFT algorithm expressed in a simple matrix form. This has resulted in a substantial computational savings compared to standard row-column FFT algorithms. Furthermore, this matrix form of the algorithm can lead to systolic array implementation in a forward manner.

References:

[1] W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. Comput.*, vol.19, no.90, pp.297-301,1965.

[2] H. V. Sorensen and C. S. Burrus, " Efficient computation of the DFT with only a subset of input/output points," *IEEE Trans. on Sig. Proc.*, vol.41, No.3, pp.1184-1200, Mar. 1993.

[3] W. S. Hinshaw *et al.*, "An Introduction to NMR Imaging: from the Block Equation to the Imaging Equation," *Proc. IEEE*, vol.71, no.3, Mar. 1983.

[4] I. Gertner and . Shamash, "VLSI Architectures for Multidimensional Fourier Transform Processing," *IEEE Trans. on Computers*, vol.C-36, No.11, pp.1265-1274, Nov.1987.

[5] G. Angelopoulos and I. Pitas, "Two-Dimensional FFT Algorithms on Hypercube Machines," *Proc. Transputer Applications 91*, Glasgow, 1991.

[6] J. Granata, M. Conner and R. Tolimieri, "The Tensor Product: a Mathematical Programming Language for FFTs and Other Fast DSP Operations," *IEEE SP Magazine*, pp.40-48, Jan.1992.

[7] E. L. Zapata, F. F. Rivera, I. Benavides, J. M. Carazo and R. Peskin, "Multidimensional fast Fourier Transform into SIMD Hypercubes," *IEEE proc.*, vol.137.Pt.E, No.4, pp.257-260, July 1990.

[8] L. P. W. Niemel and R. Prasad, "On the Reduction of Multidimensional DFT to Separable DFT by Smith Normal Form Theorem," *European Trans. on Telecomm. and Related Techn.*, vol.5, No.3, pp.377, May-June 1994.

[9] B. J. Falkawski, "Properties and Ways of Calculation of Multi-Polarity Generalized Walsh Transforms," *IEEE Trans. on Circ. and Sys.-II: Analog and digital signal processing*, vol.41, No.6, pp.380-391, June 1994.

[10] J. M. Rius and R. De Porrata-Doria, "New FFT Bit-Reversal Algorithm," *IEEE Trans. on Sig. Proc.*, vol.43, ISS.4, pp.991-994, Apr. 1995.

[11] D. Chikouche, A. Khellaf, S. Bouguezel, "A new proposed algorithm of arbitrary radix for the

computation of the 2D DFT", *Inter. Journal Numerical Methods in Engineering*, vol. 46, pp. 103-115, 1999.