Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information Sciences 176 (2006) 1–26

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Fast, prime factor, discrete Fourier transform algorithms over $GF(2^m)$ for $8 \leqslant m \leqslant 10$ ☆

T.K. Truong [a,*], P.D. Chen [a], L.J. Wang [b], Y. Chang [a], I.S. Reed [c]

[a] *Department of Information Engineering, I-Shou University, 1, Section 1, Hsueh-Cheng Rd, Ta-Hsu Hsiang, Koahsiung County 84008, Taiwan*

[b] *Department of Information Technology, National Pingtung Institute of Commerce, Pingtung 900, Taiwan*

[c] *Department of Electrical Engineering-Systems, EEB 500, University of Southern California, Los Angeles, CA 90089-2565, USA*

Received 21 February 2004; received in revised form 9 October 2004; accepted 12 October 2004

Abstract

In this paper it is shown that Winograd's algorithm for computing convolutions and a fast, prime factor, discrete Fourier transform (DFT) algorithm can be modified to compute Fourier-like transforms of long sequences of $2^m - 1$ points over $GF(2^m)$, for $8 \leqslant m \leqslant 10$. These new transform techniques can be used to decode Reed–Solomon (RS) codes of block length $2^m - 1$. The complexity of this new transform algorithm is reduced substantially from more conventional methods. A computer simulation verifies these new results.

☆ This work was supported by the National Science Council, ROC, under Grant NSC 89-2745-P-214-004, NSC 90-2213-E-214-012, and NSC 91-2213-E-251-006.

* Corresponding author. Tel.: +886 7 6577711 ext.6006; fax: +886 7 6577293.

E-mail addresses: truong@isu.edu.tw (T.K. Truong), cpd14642@ms2.hinet.net (P.D. Chen), ljwang@npic.edu.tw (L.J. Wang), milly@milly.usc.edu (I.S. Reed).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2004.10.001

Keywords: Winograd's algorithm; Prime factor DFT algorithm; Cyclic convolution; Reed–Solomon codes

1. Introduction

The discrete Fourier transform (DFT) [1] has the form

$$A_j = \sum_{i=0}^{d-1} a_i \omega^{ij} \quad \text{for } 0 \leqslant j \leqslant d-1, \tag{1}$$

where A_j and a_i are elements in the complex number field for $0 \leqslant i \leqslant d-1$ and ω is the dth root of unity.

In 1968, Rader [1] proposed a method which can be used to compute the DFT by the use of a cyclic convolution when the transform length, namely, d is a prime integer. Also, Winograd [2] showed how to compute this cyclic convolution with a minimum number of multiplications. By the use of Winograd's idea, Agarwal and Cooley [3] developed convolution algorithms that achieved this minimum number of multiplications. A prime factor DFT algorithm for computing Fourier transforms over the complex number field was developed by Kolba and Park [4]. It is based on Winograd's algorithm [2] together with the Chinese remainder theorem for polynomials [5]. More generally, in 1979, the first and fourth authors [6,7] showed in detail that the modified prime factor DFT algorithm could be used to compute d-point transforms over $GF(2^m)$, where $d = 2^m - 1$ for $4 \leqslant m \leqslant 8$, which is defined by

$$A_j = \sum_{i=0}^{d-1} a_i \beta^{ij} \quad \text{for } 0 \leqslant j \leqslant d-1, \tag{2}$$

where β is the dth root of unity in $GF(2^m)$ and $a_i \in GF(2^m)$ for $0 \leqslant i \leqslant d-1$. This latter method required only a small fraction of the number of multiplications and additions required for the previous DFT.

More recently, based on the general ideas of [8,9], Trifonov and Fedorenko (TF) [10] developed a fast computation of the Fourier transform of length d over $GF(2^m)$. Such a new type of fast Fourier transform (FFT) algorithm requires less multiplications and additions than those of Horner's method [11], the modification of Goertzel's algorithm for finite fields [12,13], Zakharova's method [14], and all of other algorithms described in [15,16]. However, the TF algorithm was shown to be efficient only for computing short transforms over finite fields. For the large values of transform length, it requires a lot of computer memory. As a result, the TF algorithm is not practical for hardware or firmware implementations.