FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 16, Dec. 2003, 415-422

# FFT and Decision Diagram Methods for Calculation of Discrete Spectral Transforms

**Radomir S. Stanković**

**Abstract:** In this paper, we discuss FFT and Decision diagrams (DD) methods for calculation of discrete spectral transforms of signals represented by large sequences. It has been pointed out that DD-methods are performing of FFT-like algorithms over decision diagrams instead of vectors. Differences between these methods are discussed and it is explained where from DD-methods take the advantages in processing of large functions.

**Keywords:** Digital signal processing, FFT, Haar transform, Walsh transform.

## 1 Introduction

Fourier analysis and various spectral methods derived from it are important mathematical tools in signal processing and many other related areas. Fast Fourier Transform (FFT) [1], an algorithm for calculation of the Discrete Fourier Transform (DFT) efficiently in terms of space and time, and FFT-like algorithms for various other discrete spectral transforms makes the application of these methods possible in engineering practice. Roughly speaking, FFT converts calculation of the Fourier transform of a large order into a series of Fourier transforms of smaller orders [1]. In particular, for a discrete signal represented by a sequence of $2^n$ elements, FFT converts calculation of the discrete Fourier transform of order $2^n$ into calculation of $n$ transforms of order 2 [2], [3]. In FFT and related algorithms, vectors

Manuscript received June 22, 2003. An earlier version of this paper was presented at the 10th Telecommunications Forum TELFOR 2002, November 26-28, 2002, Belgrade, Serbia.

The author is with University of Niš, Faculty of Electronic Engineering, Dept. of Computer Science, Beogradska 14, 18 000 Niš, Serbia and Montenegro (e-mail: rstankovic@bankerinter.net).

are usually assumed as data structures for representation of input signals, results of intermediate calculations, and output signals, i.e., the spectra of discrete transforms.

There are areas where processing of signals defined in a large number of points, thus, represented by large vectors, is required. Examples of such areas are switching theory and logic design, that can be viewed as subareas of signal processing and discrete systems theory devoted to the processing and realization of logic signals, i.e., signals described by logic functions. Examples of spectral transforms used in these areas are the discrete Walsh transform, which is the Fourier transform on finite dyadic groups, the Haar transform, the Reed-Muller transform, the Arithmetic transform, etc. In these areas, application of FFT-like algorithms appears inefficient, and in many cases even impossible, due to the request to manipulate and calculate with huge vectors.

Decision diagrams (DDs) [4] are data structures permitting compact representations of large discrete functions. Due to that, DDs can be used to efficiently calculate spectra of discrete transforms for large functions, where the FFT-like algorithms cannot be used.

In this paper, we discuss features of FFT and DD methods for calculation of spectral transforms, and point out the differences between these methods.

## 2 FFT-like Algorithms

In what follows, we briefly discuss basis features of FFT algorithms. For simplicity, the presentation will be given for functions in $C(C_2^n)$, where $C$ is the complex field, and $C_2$ is the basic cyclic group of order 2, i.e., $C_2 = (\{0,1\}, \oplus)$, where $\oplus$ denotes the addition modulo 2. However, all the main characteristics of FFT in $C(C_2^n)$ are true in the general case of finite decomposable not necessarily Abelian groups.

### 2.1 Spectral transforms

In this paper we consider discrete spectral transforms defined by Kronecker product representable transform matrices

$$\mathbf{Q}(n) = \bigotimes_{i=1}^{n} \mathbf{Q}_i(1),$$

where $\otimes$ denotes the Kronecker product and $\mathbf{Q}_i(1)$ are the basic transform matrices.

**Example 1** *For the Walsh, Complex-Hadamard [5], and arithmetic transform, the basic transform matrices are* $\mathbf{W}_i(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $\mathbf{CH}_i(1) = \begin{bmatrix} 1 & j \\ -j & -1 \end{bmatrix}$, $i = \sqrt{-1}$, $\mathbf{A}_i(1) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$, *for* $i = 1, \ldots, n$, *respectively. The Reed-Muller transform is defined over* $GF(2)$ *by the basic transform matrix* $\mathbf{R}_i(1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $i = 1, \ldots, n$. *The Haar transform is a related Kronecker layered transform, and can be expressed in terms of recurrence relations with the basic matrices* $\mathbf{W}(1)$, *and* $\mathbf{I}(1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

Basic transform matrices determine the butterfly operations used in FFT-like algorithms. Basic features of these algorithms can be summarized as follows.

1. In FFT, input data, results of intermediate calculations, and output data are represented by vectors. Therefore, all the calculations are performed over vectors with componentwise performed operations.
2. Efficiency is based on the advantages taken from the decomposition of the domain group into the product of $n$ subgroups. Due to that, a spectral transform of order $2^n$ is performed as a series of $n$ transforms of order 2. Thus, FFT consists of $n$ steps, with operations performed over pairs of data.
3. For a given group $C_2^n$ and a given transform in $C(C_2^n)$, FFT has the same complexity in both of time and space, independently on the peculiar properties a processed function $f$ may possess.
4. However, further advantages may be achieved by taking into account properties of the basic functions in terms of which the considered transform is defined. Due to that, some steps of FFT-like algorithm can be simplified.

**Example 2** *Fig. 1 and Fig. 2 show FFT for the Walsh transform and the Haar transform for* $n = 3$. *The simplification in the case of the Haar transform, follows from definition of the Haar functions and their wavelet structure.*

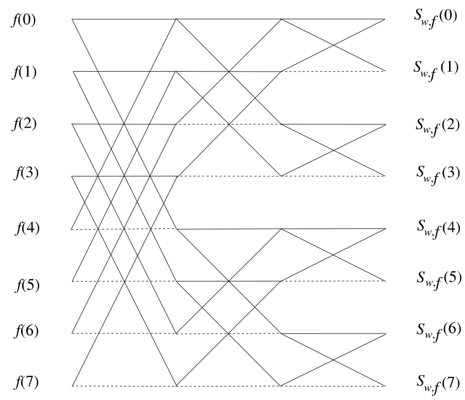


Fig. 1. Fast Walsh transform.

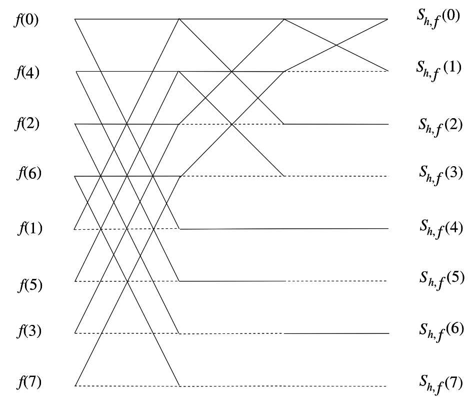


Fig. 2. Fast Haar transform

## 3 DD-methods

### 3.1 Decision diagrams

A decision diagram (DD) is a data structure for efficient representation of discrete functions. Binary DDs (BDDs) [4] are probably the most widely used DDs. They are intended to represent switching functions. Multi-terminal BDDs (MTBDDs) [6] are generalizations of BDDs derived by allowing the integers, the real number, or the complex numbers as the values of constant nodes. Thus, they can be used to represent complex-valued functions defined in $C(C_2^n)$ for large number of variables. Due to that, DDs are used as a basic data structure in related methods for calculation of spectral transforms of large functions. Decision diagrams are derived by the reduction of Decision trees (DTs) [7]. The reduction is performed by deleting isomorphic subtrees, which in the truth-vectors correspond to the equal subvectors of orders $2^k$, $k \leq 2^{n-1}$, where $n$ is the number of variables in the represented function $f$. Therefore, the reduction in a DT is possible iff there are constant or equal subvectors in the truth-vector for $f$. The impact of the deleted nodes can be represented by the cross points defined as points where an edge crosses a level in the DD. In a DD, a level consists of nodes corresponding to the same variable. The notion of DDs will be introduced by the following example.

**Example 3** *Fig. 3.1 shows BDT for a function $f$ given by the truth-vector* $\mathbf{F} = [0,1,0,0,1,1,0,1]^T$. *Fig. 4 shows the BDD for $f$ derived by reduction of the BDT for $f$ by using properties of $f$.*

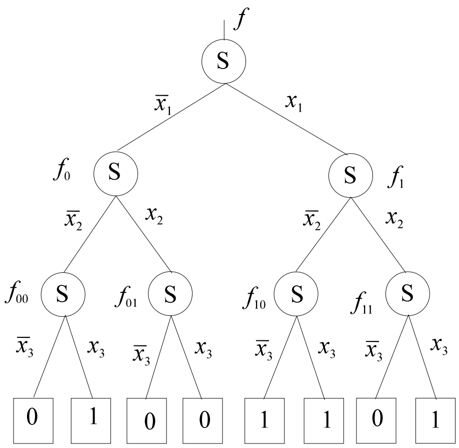


Fig. 3. BDT for $f$ in Example 3

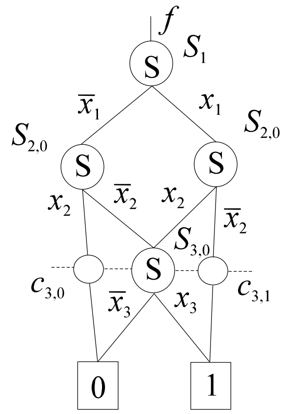


Fig. 4. BDD for $f$ in Example 3.

### 3.2 DD-methods for spectral transforms

In FFT, the spectrum of a given function $f$ with respect to the transform defined by $\mathbf{Q}$ is calculated by performing at the $i$-th step the butterfly operations determined by $\mathbf{Q}_i(1)$ [8]. Similar, in DD-methods, a DD for $f$ is converted in the DD for the spectrum $S_f$ for $f$ by performing at each node an the cross point the basic transform matrix $\mathbf{Q}_i(1)$. Since at the $i$-th level we perform calculations determined by the basic matrix $\mathbf{Q}_i(1)$, levels in the DDs correspond to steps in FFT-like algorithms.

**Example 4** *Fig. 5 shows the correspondence between FFT-like algorithms and MTBDTs for $n = 3$ and explains the calculation of the Complex Hadamard spectrum for the function $f$ in Example 3. The basic operations performed at the nodes of the BDT are determined by* $\mathbf{CH}(1)$.

Basic features of Decision diagram methods for calculation of spectral transforms can be summarized as follows.

1. Unlike FFT, complexity of DD-methods depend on the properties of the processed functions. For different functions of the same number of variables, both space and time complexity may differ very considerably, from polynomial to exponential complexity. Advantages and possibility to process large functions are due to the reduction performed in

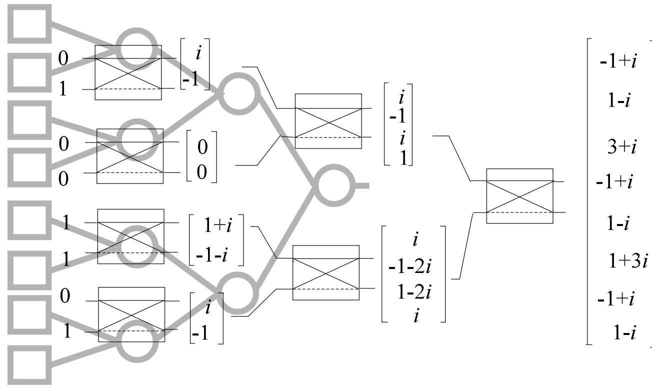


Fig. 5. FFT and MTBDT.

transferring a DT into a DD. The processing of constant subvectors is simplified, since such subvectors are represented by a single constant node in the DD. Further, it is avoided repeated processing of identical subvectors, since they are eliminated from the DD.

2. Input data, results of intermediate calculations, and output data are represented by DDs. Calculations are performed over the nodes and cross points [9] in the DD, and consist of the basic FFT operations [10].

3. The same as in FFT, advantages are taken from the decomposition of the domain group, and algorithms are performed in $n$ steps, each step corresponding to a variable in $f$, or a level in the DD for $f$.

4. The same as FFT, DD-methods permit to take advantages from properties of the basis functions determining the structure of the transform matrix. In FFT, that means avoiding of some calculations in some steps of the algorithm. In DD-methods, we avoid processing of some nodes.

**Example 5** *In FFT-like algorithms, peculiar properties of the transform matrices are exploited to simplify the calculations, as shown in Fig. 1 for the Haar transform. These properties of transform matrices can be also exploited in DD-methods. Due to that, for the Haar transform just the leftmost nodes in MTBDTs are processed by* $\mathbf{W}(1)$*, while all other nodes are processed by* $\mathbf{I}(1)$*, thus, they are unprocessed.*

Table 1. Calculation times.

| $f$ | In | Out | RM | Walsh | Arith | Haar |
|---|---|---|---|---|---|---|
| alu4 | 14 | 8 | 400 | 1590 | 640 | 2.26 |
| apex4 | 9 | 19 | 270 | 370 | 160 | 1.69 |
| misex3 | 14 | 14 | 310 | 1610 | 380 | 2.12 |
| 5xp1 | 7 | 10 | 50 | 20 | 10 | 0.11 |
| sao2 | 10 | 4 | 50 | 30 | 40 | 0.20 |

*In [11], it is derived a DD-method for calculation of the Haar transform by taking advantages from this structure of FFT for the Haar transform. Due to that, the processing at each node and the cross point is reduced to the processing of the first elements in the subfunctions related to the processed node and the cross point. Thanks to that, calculation of the Haar transform is faster than that of the related transforms. Table 1 shows the CPU times in* $10^{-3}$ *seconds for calculation of the Walsh, arithmetic (Arith), Reed-Muller (RM), and the Haar transform through MTBDDs. An important feature is that the related algorithms are implementable on simple hardware. In this experiment, the calculations are performed on a 133MHz Pentium PC with 32MBytes of RAM.*

## 4 Closing Remarks

FFT-like algorithms for discrete spectral transforms allows calculation of discrete spectral transforms efficiently in terms of space and time. Calculations are performed over vectors of function values, and it follows that applicability of these algorithms is restricted by the length of vectors that can be processed at an available hardware. For a given transform, the complexity of a FFT-like algorithm is the same for any function of the same number of variables. Efficiency of the algorithms is achieved by exploiting structure of the transform matrices and properties of the transforms performed.

In DD-methods, calculations are performed over DDs instead over vectors. The structure of transform matrices corresponds to the recursive structure of DDs, and besides properties of transform matrices, the properties a given function $f$ may posses are also exploited, since redundant and repeated calculations over constant and equal subvectors are avoided due to the reduction of isomorphic subtrees.

## References

[1] J. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Math. Computation*, vol. 19, pp. 297–301, 1965.

[2] N. Ahmed and K. R. Rao, *Orthogonal Transform for Digital Signal Processing.* Berlin: Springer-Verlag, 1975.

[3] M. Karpovsky, *Finite Orthogonal Series in the Design of Digital Devices.* Wiley and JUP, 1976.

[4] R. E. Bryant, "Graph-based algorithms for Boolean functions manipulation," *IEEE Trans. Computers*, vol. C-35, pp. 667–691, 1968.

[5] B. Falkowski and S. Rahardja, "Composite complex Hadamard spectra of Boolean functions," in *Proc. 29th IEE Int. Symp. on Circuits and Systems*, (USA), May 1996.

[6] E. Clarke, K. L. Milan, X. Zhao, and M. M. Fujita, "Spectral transforms for extremely large boolean functions," in *Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expression in Circuit Design* (U. Kebschull, E. Schubert, and W. Rosenstiel, eds.), pp. 86–90, Germany: Hamburg, Sep. 1993.

[7] T. Sasao and M. Fujita, *Representations of Discrete Functions.* Massachusetts: Kluwer, 1996.

[8] R. Stanković, M. Stanković, and D. Janković, *Spectral Transforms in Switching Theory, Definitions and Calculations.* Belgrade: Nauka, 1998.

[9] R. Stanković, T. Sasao, and C. Moraga, "Spectral transform decision diagrams." in: [7]. pp. 55-92.

[10] R. Stanković and B. Falkowski, "Fft and decision diagrams based methods for calculation of spectral transforms," *Proc. Int. Conf. on Informatics, Communications and Signal Processing (1st ICICS)*, pp. 241–245, 1997.

[11] M. Stanković, D. Janković, and R. Stanković, "Efficient algorithms for haar spectrum calculation," *Scientific Review*, pp. 171–182, 1996 (No.21-22).