

# Efficient Factorization of DSP Transforms using Taylor Expansion Diagrams

J eremie Guillot, Emmanuel Boutillon  
Laboratoire LESTER  
Universit  de Bretagne Sud  
BP 92116, 56321 Lorient Cedex, France  
jeremie.guillot@univ-ubs.fr

Q. Ren, M. Ciesielski, D. Gomez-Prado, S. Askar  
University of Massachusetts  
Department of Electrical & Computer Engineering  
Amherst, MA 01003  
ciesiel@ecs.umass.edu

## Abstract

*This paper describes an efficient method to perform factorization of DSP transforms based on Taylor Expansion Diagram (TED). It is shown that TED can efficiently represent and manipulate mathematical expressions. We demonstrate that it enables efficient factorization of arithmetic expressions of DSP transforms, resulting in a simplification of the computation.*

## 1. Introduction

This paper addresses the issue of optimization of Digital Signal Processing (DSP) transforms, commonly used in telecommunications and multi-media applications. We present a *systematic* method for performing and optimizing behavioral transformations, based on Taylor Expansion Diagrams (TED) [1]. TED is used as an efficient data structure to optimize the initial algorithmic description.

We illustrate our approach by the following example of the Discrete Cosine Transform (DCT), used frequently in multimedia applications. The DCT of type 2 is defined by:

$$y(j) = \sum_{k=0}^{N-1} x_k \cos\left[\frac{\pi}{N}j\left(k + \frac{1}{2}\right)\right], k = 0, 1, 2, \dots, N-1 \quad (1)$$

It can be computed as  $y = M \cdot x$ , where  $y$  and  $x$  are the output and input vectors and  $M$  is the transform matrix of the cosine terms. In its direct form the computation involves 16 multiplications and 12 additions for the DCT of size 4.

$$M = \begin{bmatrix} A & A & A & A \\ B & C & -C & -B \\ D & -D & -D & D \\ C & -B & B & -C \end{bmatrix} \quad (2)$$

However, by recognizing the dependence between the cosine terms, as shown in matrix 2, and performing common sub-expression elimination (CSE) and factorization, it is possible to factorize the expressions and subsequently reduce the number of the ADD/MPY operations to 6 MPY and 8 ADDs. TED makes it possible to accomplish this in a systematic way.

## 2. Taylor Expansion Diagram

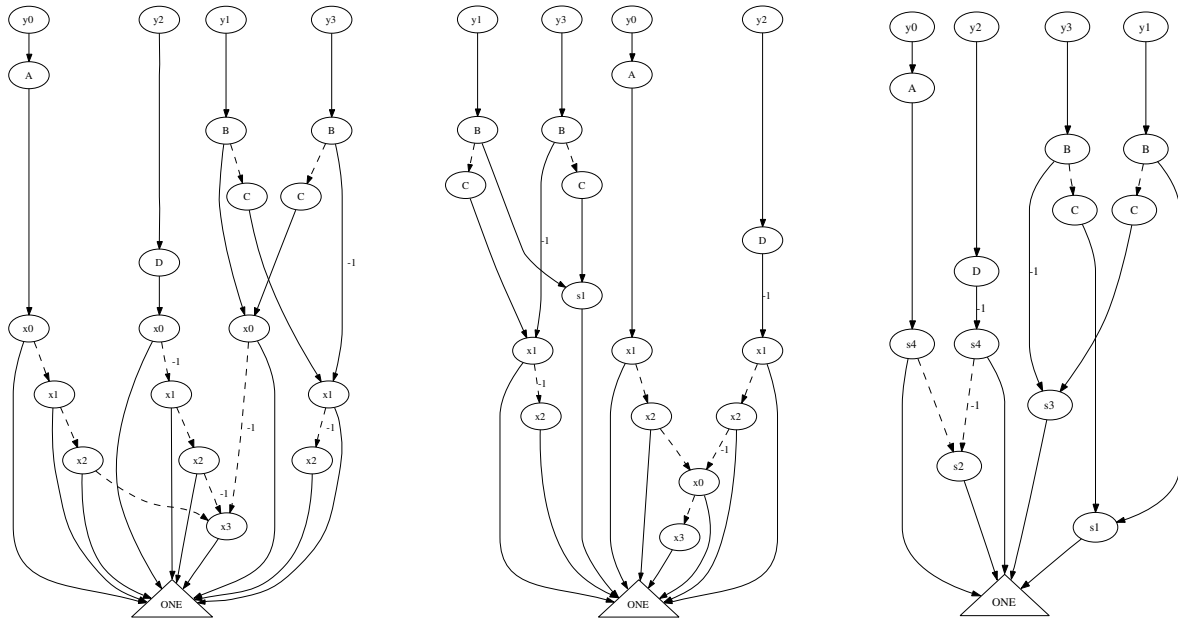
Taylor Expansion Diagram is based on a decomposition scheme using Taylor's series expansion and has the property to be canonical. For multivariate functions the terms of the expansion are successively obtained with respect to the variables, one variable at a time. The resulting decomposition is stored as a directed acyclic graph whose nodes represent the terms of the expansion.

## 3. TED-based Factorization Algorithm

The key to achieve optimum factorization of DSP transform is to represent constants as symbolic variables and place them on top of the diagram. A useful factorization is manifested in a TED by the presence of one or more nodes with multiple parent edges. This means that the subfunction rooted at that node is shared by several expressions, each represented by a path from the TED root to the terminal node 1. We illustrate our factorization algorithm with the example of the DCT2 of size 4. Figure 1 shows the TED graph of DCT2-4, with the constant nodes on top. In the figure several nodes qualify as candidates for factorization: the node with variable  $x_0$ , associated with expression  $(x_0 - x_3)$ ; and the node with variable  $x_1$ , associated with expression  $(x_1 - x_2)$ . These expressions can be extracted from the graph and substituted by new variables  $S_1, S_2, S_3$  and  $S_4$ , following the algorithm defined in Figure 2.

## 4. Results and Conclusions

The TED based factorization described in this paper has been implemented as an experimental program written in C++. The experiments show that for linear transformations, such as DCT, Discrete Wavelet transform (DWT), Walsh-Hadamard transform (WHT), etc., TED-based factorization gives the same results as SPIRAL[2] for common sub-expression elimination(CSE) but with a significantly shorter CPU time. The results are shown in Table 1. It appears that for all linear transformations examined the algorithm presented in Section 3 always arrives at an optimum solution for CSE. TED-based transformation optimization can be used for factorization of general expressions, without an

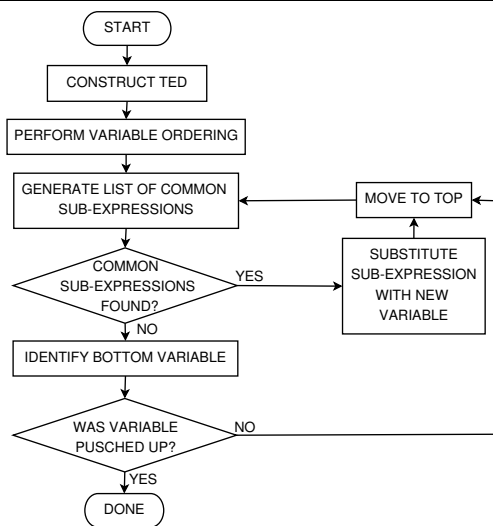


**Figure 1. a) Initial TED of DCT2-4, b) after factoring out  $S_1 = (x_0 - x_3)$  and c) after final factorization with  $S_2 = (x_0 + x_3)$ ,  $S_3 = (x_1 - x_2)$  and  $S_4 = (x_1 + x_2)$**

explicit knowledge of their internal mathematical structure. As such, TED-based optimization offers an attractive alternative to specialized systems, such as SPIRAL, that cannot handle such general cases. This result, to our knowledge, is new and opens an interesting perspective to automatically optimize transformations.

Transform Name	SPIRAL time (s)	TED time (s)	# Nodes init. TED	# Nodes fact. TED
dct4	37	0.01	17	12
dct8	107	0.04	51	32
wht4	42	0.00	12	7
wht8	751	0.03	44	13
fft4	36	0.02	33	21
fft8	106	0.21	104	29
fft16	233	2.12	324	97
fft32	367	25.33	994	353

**Table 1. Comparison of CPU Time of Spiral and TED**



**Figure 2. Flow chart of the CSE algorithm**

## References

- [1] M. Ciesielski, P. Kalla, Z. Zeng, and B. Rouzeyre. Taylor Expansion Diagrams: A Compact Canonical Representation with Applications to Symbolic Verification. In *Design Automation and Test in Europe, DATE-02*, pages 285–289, 2002.
- [2] M. Püschel, J. M. F. Moura, J. Johnson, D. Padua, M. Veloso, B. W. Singer, J. Xiong, F. Franchetti, A. Gačić, Y. Voronenko, K. Chen, R. W. Johnson, and N. Rizzolo. SPIRAL: Code generation for DSP transforms. *Proceedings of the IEEE, special issue on "Program Generation, Optimization, and Adaptation"*, 93(2), 2005.