

An 8 point DCT-II multiplierless architecture based on CMM
Marciano Prates Salbego, Sidinei Ghissoni, João Guilherme Nizer Rahmeier
 {msalbego93,sidineighissoni, joao.nizer}@gmail.com

Federal University of Pampa (UNIPAMPA/Alegrete)
Computer Architecture and Microelectronics Group - GAMA

Abstract

This paper presents an application of multiplication of variables by multiple constants, i.e., the Constant Matrix Multiplications (CMM) that use sharing of partial coefficients in DCT architecture. The use of CMM allows the change of the number of multipliers for addition/subtraction and shifting operations in the shift-add architectures, causing the reduction of area and delay. In this work, an analysis of the 8 point DCT II with 16 bit-width is performed. It was implemented with both multiplier and architecture optimized with CMM using hardware description Verilog language. The results obtained in the synthesis using the Synopsys Design Compiler, while targeting UMC 130nm CMOS technology, allows the observation of a reduction of 15% in area and 8% in power.

1. Introduction

The Discrete Cosine Transform was firstly introduced by [1] in 1974, as the result of an attempt to find a practical approximation of the Karhunen–Loève Transform (KLT), which is considered optimum regarding the compactness of the coefficients. Nowadays, its application includes such a variety of topics in digital signal processing, such as data compression, feature extraction, partial differential equations resolution, among others. Due to its frequency distribution characteristics, which tend to concentrate the signal energy closest to the lower frequency coefficients, the DCT is employed as core element in digital image compression and video standards, such as JPEG and H.264/MPEG-4.

As in other DSP operations, the major bottleneck of DCT computation resides on the highly intensity amount of calculation it performs over the input data set, being most of these operations constituted of repetitive weighted sums of the type $Ax=b$. Thus, the optimization of the multiplication operation scheme, known to be responsible for most part of the chip area on such applications, is crucial to the development of low-power and low-area implementations.

In order to address the problem of hardware realization of weighted sums, a very efficient approach one may take is the CMM (Constant Matrix Multiplication). Using this approach, the multipliers are replaced by successive shift, addition and subtraction operations, organized in a tree-like fashion. Moreover, the products are decomposed into partial sub expressions, allowing the sharing of operations among the multiplicands of the same input variable. The problem of optimizing the numbers of adders/subtractors employed to implement each product is a very well studied topic, with several algorithms proposed. We have used the algorithm proposed by [9] in order to implement the DCT matrix product efficiently.

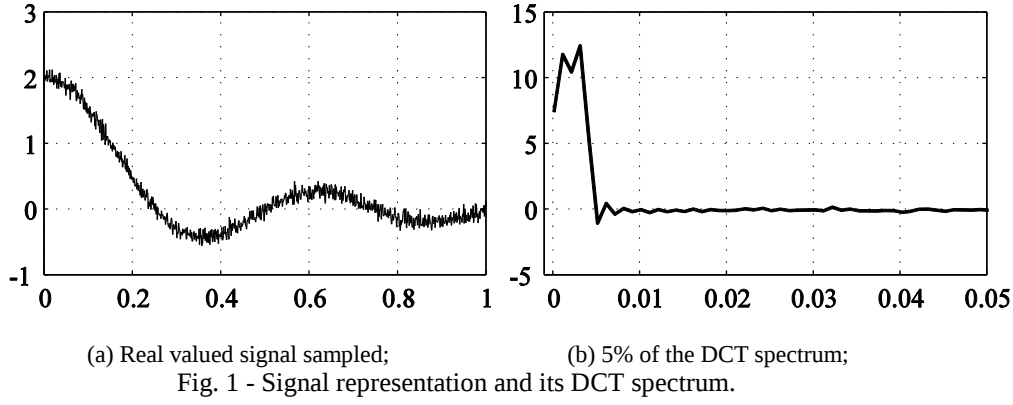
The rest of the paper is organized as follows: Section II presents the mathematical concept of the one-dimensional Discrete Cosine Transform, as well as a discussion on its fast implementations and the selected approach; Section III brings the CMM optimization methodology and finally sections IV and V show, respectively, the practical results of the proposed architecture and the conclusions.

2. The Discrete Cosine Transform

For a given real-valued input sequence X_k , $k = 0, 1, \dots, N-1$, the Discrete Cosine Transform is defined by [8] as

$$G_n = u_n \sum_{k=0}^{N-1} X_k \cdot \cos \pi \frac{(2k+1)n}{2N}, \quad 0 \leq n \leq N-1; \text{ where } u_n = \begin{cases} \frac{1}{\sqrt{N}}, & n = 0 \\ \sqrt{\frac{2}{N}}, & \text{otherwise} \end{cases} \quad (1)$$

The correlation between the signal and its frequency spectrum is shown in the Fig 1. With the purpose of compactness, it is verified that most of the coefficients are close to zero. Such property allows the signal reconstruction with relative fidelity using only a few coefficients. In data compression standards as JPEG, the bidimensional version of DCT is employed combined with entropy algorithms (that deals with the almost zeros entries in an efficient way) in order to attain a compact version of the input data while preserving its characteristics.



Equation (1) can be expressed in matrix form as follows

$$\begin{bmatrix} G_0 \\ G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \\ G_6 \\ G_7 \end{bmatrix} = \begin{bmatrix} C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 \\ C_1 & C_3 & C_5 & C_7 & -C_7 & -C_5 & -C_3 & -C_1 \\ C_2 & C_6 & -C_6 & -C_2 & -C_2 & -C_6 & C_6 & C_2 \\ C_3 & -C_7 & -C_1 & -C_5 & C_5 & C_1 & C_7 & C_3 \\ C_4 & -C_4 & -C_4 & C_4 & C_4 & -C_4 & -C_4 & C_4 \\ C_5 & -C_1 & C_7 & -C_3 & -C_3 & C_7 & C_1 & -C_5 \\ C_6 & -C_2 & C_2 & -C_6 & -C_6 & C_2 & -C_2 & C_6 \\ C_7 & -C_5 & C_3 & -C_1 & C_1 & -C_3 & C_5 & -C_7 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix}; C_n = \frac{1}{2} \cos \frac{n}{16} \pi \quad (2)$$

Over the past decades, several fast algorithms were developed [3-7]. As defined in [8], there is a theoretical lower bound of 11 multiplications for the 8-point DCT-II algorithm. This number was obtained by Loeffler et al [3] by means of a factorization of the transform matrix into 7 sparse matrices, resulting in a total of 24 sums and 11 multiplications. An even faster algorithm was presented by Arai et al [4]. This algorithm requires a total of only 5 multiplications and 29 additions. Saving 6 multiplications is possible due to the fact that each output is actually a rescaled representation of the original DCT-II algorithm output. Other possible approach is the one shown by [6] in which the N-point DCT-II computation is evaluated in terms of a N-point IDFT, after some permutation of the input data. By doing so, the complexity of the problem is transposed to the assessment of the IDFT algorithm.

Based on the decomposition method presented by [5], it is possible to express the equation system (2) as two matrix-vector products separately, in which the new vectors X_{k1} and X_{k2} are linear combinations of the initial input vector. Such reduction is possible due to the symmetry of the matrix. The resultant equation system is shown below

$$\begin{bmatrix} G_0 \\ G_2 \\ G_4 \\ G_6 \end{bmatrix} = \begin{bmatrix} C_4 & C_4 & C_4 & C_4 \\ C_2 & C_6 & -C_6 & -C_2 \\ C_4 & -C_4 & -C_4 & C_4 \\ C_6 & -C_2 & C_2 & -C_6 \end{bmatrix} \begin{bmatrix} X_0 + X_7 \\ X_1 + X_6 \\ X_2 + X_5 \\ X_3 + X_4 \end{bmatrix}; \begin{bmatrix} G_1 \\ G_3 \\ G_5 \\ G_7 \end{bmatrix} = \begin{bmatrix} C_1 & C_3 & C_5 & C_7 \\ C_3 & -C_7 & -C_1 & -C_5 \\ C_5 & -C_1 & C_7 & C_3 \\ C_7 & -C_5 & C_3 & -C_1 \end{bmatrix} \begin{bmatrix} X_0 - X_7 \\ X_1 - X_6 \\ X_2 - X_5 \\ X_3 - X_4 \end{bmatrix} \quad (3)$$

Thus the 8 point DCT-II can be computed in terms of two 4x4 matrices, reducing the number of multiplications required by half. Taking into account the pre-additional stage of the input vector X_k , the total number of additions and multiplications required is 32. But, by observing that in each row of the first matrix there are repeated constants, one can easily reduce the total multiplication number to 22. It is worth noting that, despite the fact that other faster algorithms use fewer multiplications, none of them preserves the density property of the matrices. In the context of the CMM approach, a maximum share of sub-expressions can be obtained when the matrix is dense.

3. The CMM method

In this method, the constant products are decomposed into several addition/subtraction steps. Each sub-expression arising from the previous stage may be shared or not with another sub-expression. Since the adders are structured in a tree-like fashion, the maximum number of addition steps starting from the input defines the overall delay of the operation. The implemented algorithm [9] is intended to find the minimum number of adders that implements the weighted sum under the constraints of minimizing the delay and area. Fig. 2 presented next shows the sharing of sub-expressions in the implementation of a 2x2 matrix-vector product. It can be seen that the sub-expression $17X_2$, as well as the final result Z_1 , are parts of the output Z_2 . By doing so, all 4 multiplications and 2 additions in Fig. 2 (c) are replaced by 5 additions.

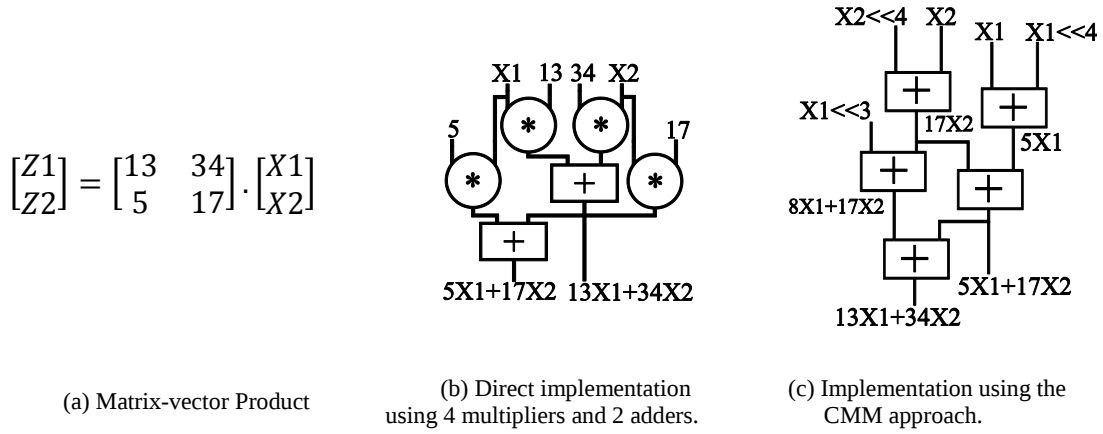


Fig 2. Direct implementation and the CMM approach applied to a 2x2 matrix-vector product.

In the CMM approach, the constants (i.e. $5X_1$) are represented with a pre-defined word length, based on the desired precision of the operation. In the particular case of the Discrete Cosine Transform, where the constants belong to the set $[-1,1]$, the quantization adopted was with a word-length of 16 bits for each constant with 2's complement numerical system. In this context, the constants are represented by their respective quantized form, e.g. $[0.523598, 0.758539, 1]$ are represented as $[17157, 23170, 32768]$. In a recent report, [8] has obtained a gain in power consumption and reducing area by using CMM associated with gate level approaches on the realization of the FFT radix-2 algorithm. Thus the CMM approach was chosen to be applied in this work.

4. Architecture and Synthesis Result

Fig. 3 presents the fully combinational architecture that implements the Discrete Cosine Transform algorithm described by definition (2) using the CMM approach. The constants were represented with a 16-bit long word and 2's complement.

The two matrix-vector products (M1 and M2 blocks) are processed in parallel, since their respective input vectors are independently provided by the previous stage (PRE-ADDER block). The PRE-ADDER block executes the sum of the input vector required due to the matrix factorization in equation (2).

The CMM method was applied in the M1 and M2 blocks. By using the algorithm proposed by [9], the first matrix-vector product was implemented with 29 additions and an maximum number of 6 series additions, the second matrix-vector product was implemented with 52 addition operations and a maximum number of 7- serial additions. Thus, it is possible to verify that the maximum path delay of the DCT architecture results from 8-serial additions, being one of those operations performed at the PRE-ADDER block and the rest at the M2 block. The chosen adder model was the Ripple Carry with two inputs. The ORDER UNSCRAMBLING block is responsible only to reverse the order of the outputs to the natural order $[G_0, G_1, \dots, G_7]^T$.

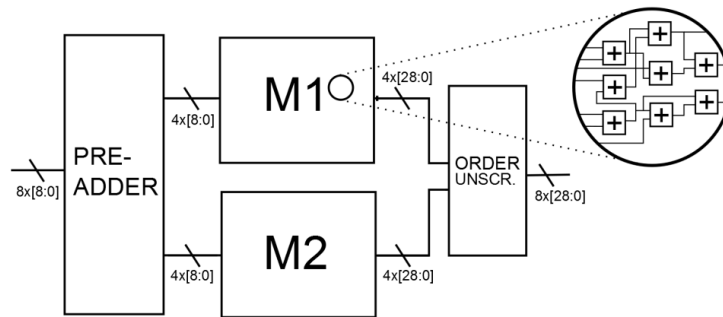


Fig 3. Proposed architecture.

In order to perform a fair comparison, the behavioral model was implemented straightforward from the matrix multiplication in Eq. (3) using the same architecture from Fig. 3. The multiplier architecture employed was the standard one provided by the synthesis tool.

Table 1, shows the main results obtained with the 16 bit-width DCT architecture, which were implemented using Verilog hardware description language. The logic synthesis was performed by Synopsys Design Compiler targeting UMC 0.13 μ m CMOS technology. Power was evaluated using Synopsys Design Power and the average power consumption was estimated with 5000 random input vectors. The maximum frequency operation obtained was 169MHz.

Tab.1 – Architecture DCT-1D 8 points results

Circuits	Area (nm ²)	Power(μW)
DCT 1-D 8 points - CMM	4253	71,37
Behavioral Model	4922	77,91

As can be seen in Table 1, the proposed architecture with CMM approach has achieved a gain of up to 15% and 8% on area and power respectively, when compared with the behavioral model implemented directly using multipliers. These better results presented by our solution are due essentially by the replacement of multipliers by adders/subtractors, which needs a less complex circuit regarding the number of logical gates employed.

5. Conclusions and Future Works

In this work, we have applied a method, namely CMM, for the decomposition of the matrix-vector product of the one-dimensional Discrete Cosine Transform algorithm aiming power and area reduction of the respective hardware implementation. The results demonstrate the efficiency of the method by reducing the overall complexity of the performed operations by means of replacing multiplications by addition/subtraction's.

In future works we intend to use the architecture proposed in this work to implement the two-dimensional version of the Discrete Cosine Transform. Since its kernel has separable variables, it can be easily extended to a 2-D architecture by using the 1-D block already developed. The development of a quantizer block associated to the DCT is also a relevant topic to be explored.

6. References

- [1] Ahmed, N.; Natarajan, T.; Rao, K.R.; "Discrete Cosine Transform," *Computers, IEEE Transactions on* , vol.C-23, no.1, pp.90-93, Jan. 1974.
- [2] Ghissoni, S. ; Costa, E. ; Monteiro, J. C. ; Reis, R. A. L.; "Combination of Constant Matrix Multiplication and Gate-Level Approaches for Area and Power Efficient Hybrid Radix-2 DIT FFT Realization";*18th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*; p.567-570; 2011.
- [3] Loeffler, C.; Ligtenberg, A.; Moschytz, G.S.;"Practical fast 1-D DCT algorithms with 11 multiplications"*Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on* ,pp.988-991 vol.2, 23-26 May 1989.
- [4] Y. Arai, T. Agui, and M. Nakajima, "A Fast DCT-SQ scheme for Images", *Transactions of IEICE*, Vol. E 71, No. 11, pp. 1095-1097, November 1988
- [5] W.-H. Chen, C. H. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform", *IEEE Trans. on Communications* 25, pp. 1004–1009, Sep 1977.
- [6] Narasimha, M.; Peterson, A.; "On the computation of the Discrete Cosine Transform", *IEEE Trans. on Communications* 26, pp. 934 - 936, Jun 1978.
- [7] Duhamel, P.; H'Mida, H.; , "New 2ⁿDCT algorithms suitable for VLSI implementation," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.* , vol.12, no., pp. 1805-1808, Apr 1987.
- [8] Feig, E.; Winograd, S.; , "Fast algorithms for the discrete cosine transform," *Signal Processing, IEEE Transactions on* , vol.40, no.9, pp.2174-2193, Sep 1992.
- [9] Aksoy, L.; da Costa, E.; Flores, P.; Monteiro, J.; , "Exact and Approximate Algorithms for the Optimization of Area and Delay in Multiple Constant Multiplications," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.27, no.6, pp.1013-1026, June 2008.