

A HARDWARE ACCELERATOR FOR THE COMPUTATION OF MODIFIED DISCRETE SINE TRANSFORM

Doru Florin Chiper^{1,2,3}

Abstract: This work presents an efficient hardware implementation of a hardware accelerator for the computation of the Modified Discrete Sine transform (MDST) using a new VLSI algorithm based on a appropriate reformulation of the MDST algorithm using some auxiliary input and output sequences. The obtained hardware implementation is using a low complexity implementation based on only adders/subtractors and has a reduced critical path that can be exploited to obtain a significant reduction of the power consumption.

Keywords: Discrete transforms, Modified Discrete Sine transform, VLSI algorithms, VLSI architectures, data compression

DOI <https://doi.org/10.56082/annalsarsciinfo.2021.1-2.57>

1. Introduction

The Modified Discrete Cosine Transform (MDCT), the Modified Discrete Sine Transform (MDST), and their inverse transforms (IMDCT and IMDST) are used in subband analysis/synthesis approaches [1],[2] that have been used to construct filter-banks used in Dolby Enhanced AC3 (E-AC-3) audio coding standard [3] and some other audio coding standards [4]-[6].

MDCT and MDST are computational intensive as also DCT and DST and efficient software and hardware algorithms and implementations are required for a real-time implementation.

There are several efficient software implementations [7]-[10] and some hardware implementations [11]-[20] but all of these hardware solutions are based on recursive algorithms.

Although it is possible to establish a quite simple relation between the MDCT and the MDST, that allows us to concentrate more on the investigation of a fast MDCT algorithm and implementations, the MDST computation through the MDCT algorithm still takes extra executing time, although both of the MDCT and MDST hardware accelerators could be unified. This indicates that the MDCT and MDST coefficients cannot be simultaneously computed. Since these equations are all dependent, how to efficiently compute the MDCT and MDST coefficients is still a challenging problem.

In this paper we propose a direct method to efficiently implement the MDST algorithm using some auxiliary input and output sequences. Thus, we can

¹Technical University Gheorghe Asachi Iasi

²The Academy of the Romanian Scientists, Romania

³The Technical Sciences Academy of Romania

obtain an efficient VLSI implementation for a hardware accelerator using only adders/subtractors and having a reduced critical path that allows in the same time a high speed implementation using a reduced power consumption.

The rest of the paper is organized as follows:

In Section 2 it is presented the new VLSI algorithm for MDST that allows an efficient hardware implementation for the hardware accelerator that works together with a host computational structure.

In Section 3 it is presented the VLSI implementation of the proposed algorithm and in Section 4 some conclusions are derived.

2. A New VLSI Algorithm for Modified DST

The 1-D MDST for a real input sequence $x(i) : i = 0, 1, \dots, N-1$, is defined as:

$$Y(k) = \sum_{i=0}^{N-1} x(i) \cdot \sin[(2i+1+N/2)(2k+1)\alpha] \quad (1)$$

for $k = 0, \dots, M-1$

where: $M=N/2$

$$\text{and} \quad \alpha = \frac{\pi}{2N} \quad (2)$$

We introduce some restructuring input sequences defined bellow:

$$x_s(i) = x(i) \cdot \sin[(2i+1+N/2)\alpha] \quad (3)$$

$$x_c(i) = x(i) \cdot \cos[(2i+1+N/2)\alpha] \quad (4)$$

$$x_a(i) = (x_c(i) + x_c(N-1-i)) \quad (5)$$

$$x_b(i) = (x_c(i) - x_c(N-1-i)) \quad (6)$$

Using these auxilliary input sequences, it is possible to obtain an efficient implementation of a hardware accelerator for the computation of MDST.

Thus, when the length of the transform is a power of 2 it is possible to significantly reduce the hardware complexity of the designed accelerator. For example, when $N=8$ we have the following equations for our hardware accelerator:

$$T(0) = 0 \quad (7)$$

$$T(2) = -2 \cdot \cos\left(\frac{\pi}{4}\right) [(x_b(0) - x_b(3)) + (x_b(1) - x_b(2))] \quad (8)$$

and

$$T(1) = 2 \cdot \left[\cos\left(\frac{\pi}{N}\right) (x_a(0) - x_a(3)) + \cos\left(3\frac{\pi}{N}\right) (x_a(1) - x_a(2)) \right] \quad (9)$$

$$T(3) = 2 \cdot \left[\cos\left(\frac{\pi}{N}\right) (x_a(1) - x_a(2)) - \cos\left(3\frac{\pi}{N}\right) (x_a(0) - x_a(3)) \right] \quad (10)$$

Using a subexpression technique we can further reduce the number of adders/subtractors. Thus, we can note:

$$x_a(0,3) = x_a(0) - x_a(3) \quad (11)$$

$$x_a(1,2) = x_a(1) - x_a(2) \quad (12)$$

and

$$x_b(0,3) = x_b(0) - x_b(3) \quad (13)$$

$$x_b(1,2) = x_b(1) - x_b(2) \quad (14)$$

Thus, equations (6)-(9) can be rewritten as:

$$T(0) = 0 \quad (15)$$

$$T(2) = -2 \cdot \cos\left(\frac{\pi}{4}\right) [x_b(0,3) + x_b(1,3)] \quad (16)$$

and

$$T(1) = 2 \cdot \left[\cos\left(\frac{\pi}{N}\right) x_a(0,3) + \cos\left(3\frac{\pi}{N}\right) x_a(1,2) \right] \quad (17)$$

$$T(3) = 2 \cdot \left[\cos\left(\frac{\pi}{N}\right) x_a(1,2) - \cos\left(3\frac{\pi}{N}\right) x_a(0,3) \right] \quad (18)$$

Finally, the output sequence can be recursively computed using equations (19) and (20) as follows:

$$Y(0) = \sum_{i=0}^{N-1} x_s(i) \quad (19)$$

$$Y(k) = T(k) - Y(k-1) \quad (20)$$

for $k=1, \dots, M-1$

Thus, using the auxilliary output sequence $\{T(k):k=1, \dots, M-1\}$ we can recursively compute the final output sequence using an accumulator structure.

3. A VLSI Implementation of MDST Using the Proposed VLSI Algorithm

The equations (17) and (18) that are computing the odd samples $T(1)$ and $T(3)$ of the auxiliary output sequence $T(k)$ can be implemented using the hardware structure from Figure 1. It can be seen that $T(1)$ and $T(3)$ can be computed using 4 multiplications with the constants $\cos(\pi/N)$ and $\cos(3\pi/N)$. Due to the fact that we have 4 multiplications where 2 multiplications are using the same constant we can further reduce the number of multipliers at 2. Moreover, the multiplications with a constant can be efficiently implemented using only adders/subtractors and shift operations as will be shown below.

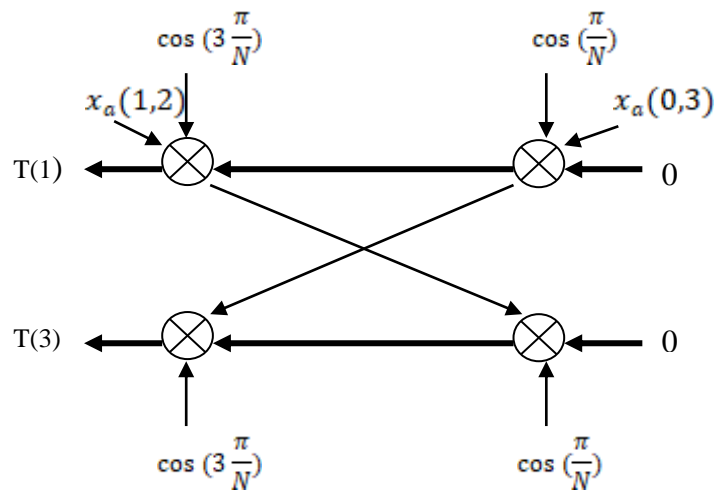


Figure 1. The architecture that implements equations (17) and (18)

The function of the processing elements from Figure 1 is presented in Figure 2. It can be seen that the partial result y is added with the result of the multiplication of the input sequence x with the constant of the multiplier c . The input sequence x is also forwarded to the next processing element as x' .

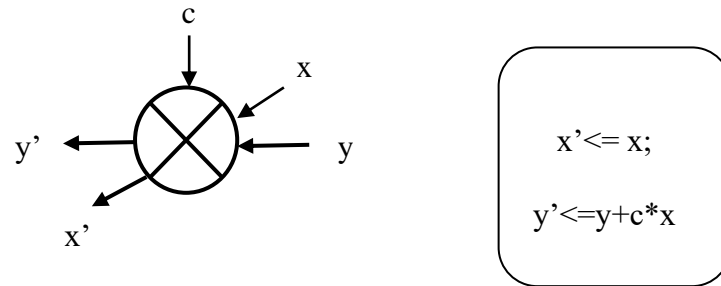


Figure 2. The function of the processing elements from Fig.1

The even samples of the auxiliary output sequence $T(k)$ are computed using the following hardware structure that implements equation (16):

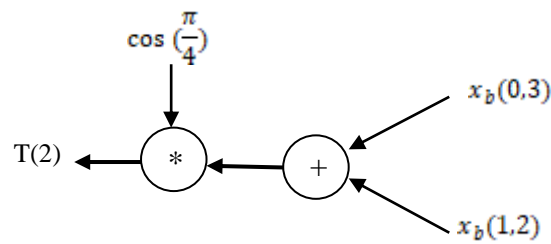


Figure 3. The hardware implementation of equation (16)

It can be seen from Figure 3 that for the computing of the even part of the auxiliary output sequence we are using only a multiplier with a constant and an adder. As already has been mentioned, the multiplication with a constant can be implemented using adders/subtractors and shift operations.

Table 1. Signed Digit representations of the constants used in constant multipliers

Coefficient (C)	Representation	No of adders/subtractors
$c(1)=\cos(\pi/4)$	$2^0 - 2^{-2} - 2^{-4} + 2^{-6} + 2^{-8}$	4
$c(2)=\cos(\pi/8)$	$2^0 - 2^{-4} - 2^{-6} + 2^{-9}$	3
$c(3)=\cos(3\pi/8)$	$2^{-1} - 2^{-3} + 2^{-7} - 2^{-9}$	3

As can be seen from Table I we can efficiently implement the multipliers with a constants from Figure 1 and Figure 3 using a Signed Digit (SD) representation of the constants $\cos(\pi/4)$, $\cos(\pi/8)$ and $\cos(3\pi/8)$. We are using only 3 or 4 adders/subtractors and shift operations. Shift operations can be implemented using appropriate interconnections without any hardware circuits.

The implementations of the 3 multipliers are presented in Figure 4, Figure 5 and Figure 6.

As can be seen from Figure 4 in order to implement the multiplication with the constant $c(1)=\cos(\pi/4)$ we are using 2 subtractors and 2 adders. The shift operations does not involve any suplimentary circuits but only appropriate interconnections. To reduce the critical path of the circuit we are using pipelining. At the intersection between the cut-set lines (represented with dot lines) from Figure 4 with the communication links we are placing the pipeline registers. Thus, the critical path has been reduced at T_a , where T_a is the delay of an adder/subtractor.

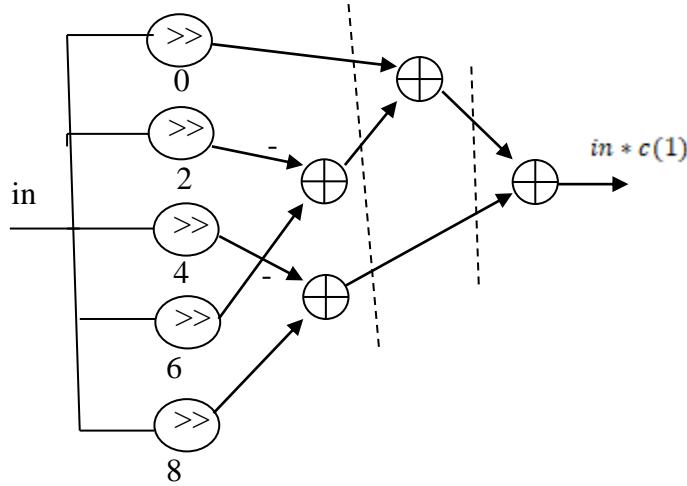


Figure 4. The implementation of the multiplier with the constant $c(1)=\cos(\pi/4)$

As can be seen from Figure 5 in order to implement the multiplication with the constant $c(2)=\cos(\pi/8)$ we are using 2 adders and one subtractor. As has been shown before, the shift operations does not involve any supplementary circuits. We have also reduced the critical path at the value T_a using pipelining. At the intersection between the cut set lines from Figure 5 with the communication links we are placing the pipeline registers.

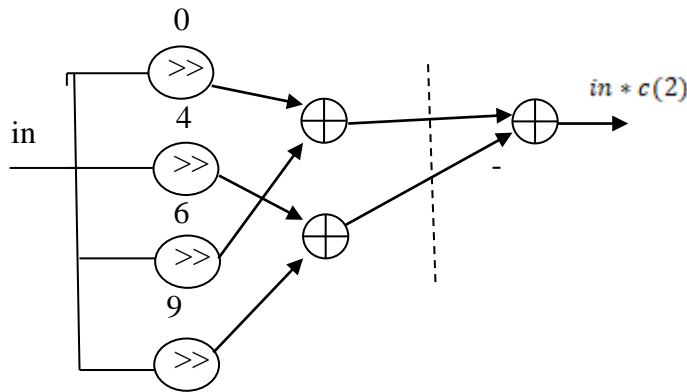


Figure 5. The implementation of the constant multiplier with the constant $c(2)=\cos(\pi/8)$

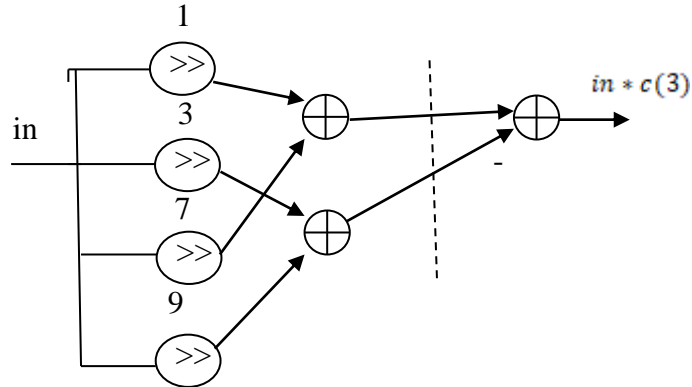


Figure 6. The implementation of the constant multiplier with constant $c(3)=\cos(3\pi/8)$

As can be seen from Figure 6 in order to implement the multiplication with the constant $c(3)=\cos(3\pi/8)$ we are using 2 adders and one subtractor. The shift operations does not involve any supplementary circuits and the critical path have been reduced at the value T_a using pipelining. At the intersection between the cut set lines from Figure 6 with the communication links we are placing the pipeline registers.

Thus, in order to implement the hardware accelerator we are using only 3 multipliers that are implemented using only 11 adders and 4 subtractors and the critical path has been reduced at the value T_a . This, can be used to reduce the power consumption by increasing the delay while reducing the power consumption by reducing the supply voltage.

4. Conclusions

An efficient hardware implementation of Modified Discrete Sine transform (MDST) with a reduced hardware complexity and high speed performances is presented. It uses an appropriate reformulation of the MDST algorithm using some auxiliary input and output sequences. The obtained hardware implementation of the proposed hardware accelerator is using a low complexity implementation using only 11 adders and 4 subtractors and has a reduced critical path that can be used to obtain a further reduction of the power consumption.

REFERENCES

- [1] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 6, pp. 969–978, Jun. 1990.
- [2] J.P. Princen, A.W. Johnson, A.B. Bradley, Sub-band/transform coding using filter bank designs based on time domain aliasing cancellation, in: *Proceedings of the IEEE ICASSP'87*, Dallas, TX, April 1987, pp. 2161–2164.
- [3] Digital Audio Compression Standard (AC-3, E-AC-3), Revision B, Adv. Television Syst. Committee (ATSC), Washington, DC, Jun. 2005, Document A/52B.
- [4] Information Technology—Generic Coding of Moving Pictures and Associated Audio, Part 3: Audio, ISO/IEC JTC1/SC29/WG11 MPEG, International Standard 13818-3 (MPEG-2), 1994.
- [5] S.B. Marovich, Faster MPEG-1 layer III audio decoding, Technical Report HPL-2000-66, Computer Systems and Technology Laboratory, HPL Laboratories, Palo Alto, June 2000.
- [6] M. Bosi, R.E. Goldberg, *Introduction to Digital Audio Coding and Standards*, Springer Science & Business Media, Inc., New York, 2003, pp. 265–332.
- [7] P. Duhamel, Y. Mahieux, J.P. Petit, A fast algorithm for the implementation of filter banks based on time domain aliasing cancellation, in: *Proceedings of the IEEE ICASSP'91*, Toronto, Canada, May 1991, pp. 2209–2212.
- [8] V. Britanak, K.R. Rao, A new fast algorithm for the unified forward and inverse MDCT/MDST computation, *Signal Processing* 82 (3) (2002) 433–459.
- [9] V. Britanak, The refined efficient implementation of the MDCT in MP3 and comparison with other methods, Technical Report II SAS-2002-2, Department of Numerical Methods and Algorithms, Institute of Informatics, Slovak Academy of Sciences, Bratislava, September 2002.
- [10] M.-H. Cheng, Y.-H. Hsu, Fast IMDCT and MDCT algorithms—a matrix approach, *IEEE Transactions on Signal Processing* 51(1) (2003) 221–229.
- [11] H. Li, P. Li, Y. Wang, An efficient accelerator architecture for implementing fast IMDCT computation, *Signal Processing* 90(8) (2010) 2540–2545.
- [12] S.-F. Lei, S.-C. Lai, P.-Y. Cheng, C.-H. Luo, Low complexity and fast computation for recursive MDCT and IMDCT algorithms, *IEEE Transactions on Circuits and Systems—II: Express Briefs* 57(7) (2010) 571–575.
- [13] J. F. Yang and C. P. Fan, "Recursive discrete cosine transforms with selectable fixed-coefficient filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 2, pp. 211–216, Feb. 1999.
- [14] V. Nikolajevic and G. Fettweis, "Computation of forward and inverse MDCT using Clenshaw's recurrence formula," *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 1439–1444, May 2003.
-

- [15] V. Nikolajevic and G. Fettweis, "New recursive algorithms for the unified forward and inverse MDCT/MDST," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 3, pp. 203–208, Jul. 2003.
- [16] C.-H. Chen, B.-D. Liu, and J.-F. Yang, "Recursive architectures for realizing modified discrete cosine transform and its inverse," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 1, pp. 38–45, Jan. 2003.
- [17] Z.-Y. Cheng, C.-H. Chen, B.-D. Liu, and J.-F. Yang, "Unified selectable fixed-coefficient recursive structures for computing DCT, IMDCT and subband synthesis filtering," in *Proc. IEEE ISCAS, Vancouver, BC, Canada, May 2004*, vol. 3, pp. 557–560.
- [18] S.-C. Lai, S.-F. Lei, and C.-H. Luo, "Common architecture design of novel recursive MDCT and IMDCT algorithms for application to AAC, AAC in DRM, and MP3 codecs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 10, pp. 793–797, Oct. 2009.
- [19] S.-F. Lei, S.-C. Lai, P.-Y. Cheng, and C.-H. Luo, "Low complexity and fast computation for recursive MDCT and IMDCT algorithms," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 7, pp. 571–575, Jul. 2010.
- [20] H. Li, P. Li, and Y. Wang, "A compact hardware accelerator structure for realizing fast IMDCT computation," in *Proc. Asia Pacific Conf. Postgraduate Res. Microelectron. Electron., PrimeAsia, 2009*, pp. 317–320.
-