Odd type DCT/DST for video coding: Relationships and low-complexity implementations

Odd Type DCT/DST for Video Coding: Relationships and Low-Complexity Implementations

M. Masera, M. Martina and G. Masera
Department of Electronics and Telecommunications
Politecnico di Torino
Turin 10129, Italy
Email: maurizio.masera@polito.it, maurizio.martina@polito.it, guido.masera@polito.it

***Abstract*—In this paper, we show a class of relationships which link Discrete Cosine Transforms (DCT) and Discrete Sine Transforms (DST) of types V, VI, VII and VIII, which have been recently considered for inclusion in the future video coding technology. In particular, the proposed relationships allow to compute the DCT-V and the DCT-VIII as functions of the DCT-VI and the DST-VII respectively, plus simple reordering and sign-inversion. Moreover, this paper exploits the proposed relationships and the Winograd factorization of the Discrete Fourier Transform to construct low-complexity factorizations for computing the DCT-V and the DCT-VIII of length 4 and 8. Finally, the proposed signal-flow-graphs have been implemented using an FPGA technology, thus showing reduced hardware utilization with respect to the direct implementation of the matrix-vector multiplication algorithm.**

I. Introduction

Transform coding has become one among the fundamental data compression techniques to remove the correlation within image, video, speech and audio. In particular, the type-II Discrete Cosine Transform (DCT-II) [1] has been largely adopted as the core transform in many image and video coding standards, such as JPEG, MPEG-2, AVC and HEVC [2]. Several relationships between the DCT-II, DCT-III and the DCT-IV have been studied, thus leading to low-complexity factorizations for this kind of transforms [3]–[10].

Recently, Saxena *et al.* [11], [12] have shown that transforms other than the DCT-II can better represent signals produced by the intra prediction of an image/video coding scheme. In particular, they demonstrated that the type-VII Discrete Sine Transform (DST-VII) approaches the optimal Karhunen-Loeve Transform in terms of decorrelation for intra-predicted signals. For this reason, the DST-VII was specified in the HEVC standard to code intra-predicted 4×4 luminance blocks.

During these last years, the Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG has been and is still developing the future video coding technology, which will improve significantly the compression efficiency with respect to the current HEVC standard [13]. Novel and enhanced coding tools have been adopted to improve frame partitioning, intra and inter prediction and the transform stage of the future codec. Specifically, an Adaptive Multiple Transform (AMT) scheme, derived from the Enhanced Multiple Transform in [14], has been specified to encode the residual signal for both inter and intra coded blocks. Depending on the coding mode, for each block the encoder chooses the best set of transforms in a pool of previously selected transforms from the DCT/DST families. The newly introduced transforms are the DCT-V, DCT-VIII, DST-I and DST-VII [14]. When the block is intra predicted, a mode-dependent transform candidate selection process is used because of the different residual statistics associated to each mode. On the other hand, only one transform set, composed of the DST-VII and DCT-VIII, is used when the block is inter predicted. Since each set is composed of two transform candidates, which have to be evaluated for both the horizontal and the vertical transform, a total of five different transform candidates (DCT-II plus four multiple transform candidates of the AMT) have to be computed for each block in all the different prediction modes. This raises the issue of very high complexity at the encoder side [15].

Therefore, the need for low-complexity factorizations to efficiently compute the DCT-V and DCT-VIII is an open research topic, which has not been completely addressed in the literature. While several fast algorithms have been proposed for computing the so-called "even type" DCTs and DSTs [3], [4], only few works addressed the problem of finding factorizations of the so-called "odd type" transforms (*i.e.* types V, VI, VII and VIII).

Starting from the knowledge that all the DCT and DST types can be viewed as special cases of the Discrete Fourier Transform (DFT), with proper input mapping and output selection [16], [17], Chivukula and Reznik [18] derived the mapping between type VI-VII DSTs and the DFT. Moreover, they exploited this mapping to propose fast algorithms to compute DST-VI and DST-VII of lengths N = 4, 8. In [19], [20], Reznik showed how to decompose the $2N + 1$-point DCT-II matrix into an $N + 1$-point DCT-VI and an N-point DST-VII. Since the odd-length DCT-II can be viewed as a real-valued DFT of the same length [21], the author showed how to use DFT factorization available in the literature [22] to derive low-complexity algorithms for the $N + 1$-point DCT-VI and the N-point DST-VII. In this paper, we derive low-complexity factorizations and the corresponding Signal-Flow-Graphs (SFGs) for the DCT-V and DCT-VIII of lengths N = 4 and 8. To this purpose, we exploit the relationships between the DFT, DCT-II, DCT-VI and DST-

VII and we define a new class of relationships among the odd type DCTs and DSTs, in particular between the DCT-VI and the DCT-V, and between the DST-VII and the DCT-VIII. Finally, hardware architectures to implement such transforms by means of the proposed low-complexity factorizations have been synthesized on FPGA, thus showing lower complexity with respect to a straightforward implementation based on matrix-vector multiplication (MVM).

The paper is organized as follows. Section II provides definitions of the DCT and DST transforms. The relationships between transforms are presented in Section III, while the derivation of low-complexity factorizations of the 4-point and 8-point DCT-V and DCT-VIII is reported in Section IV. Finally, Section V reports the hardware implementation results and Section VI concludes the paper.

II. Notations and Definitions

First, the transform matrices of the DFT, DCT of types II and the odd type DCTs and DSTs (V, VI, VII and VIII) have to be defined. Differently from the transform definitions provided in [3], normalization constants are not included in the following definitions, as they do not affect transforms factorization. Moreover, for the sake of clarity, the transform definition reported in this paper is not following the notation used in [3], but each definition has been modified so that N is the length of the transform.

DFT:	$[\mathbf{F}_N]_{k,l} = \exp^{-j\frac{2\pi kl}{N}}$	$k,l = [0, N-1]$
DCT-II:	$[\mathbf{C}_N^{II}]_{k,l} = \cos\frac{\pi k(l+\frac{1}{2})}{N}$	$k,l = [0, N-1]$
DCT-V:	$[\mathbf{C}_N^{V}]_{k,l} = \cos\frac{2\pi kl}{2N-1}$	$k,l = [0, N-1]$
DCT-VI:	$[\mathbf{C}_N^{VI}]_{k,l} = \cos\frac{2\pi k(l+\frac{1}{2})}{2N-1}$	$k,l = [0, N-1]$
DCT-VII:	$[\mathbf{C}_N^{VII}]_{k,l} = \cos\frac{2\pi (k+\frac{1}{2})l}{2N-1}$	$k,l = [0, N-1]$
DCT-VIII:	$[\mathbf{C}_N^{VIII}]_{k,l} = \cos\frac{2\pi (k+\frac{1}{2})(l+\frac{1}{2})}{2N+1}$	$k,l = [0, N-1]$
DST-V:	$[\mathbf{S}_N^{V}]_{k,l} = \sin\frac{2\pi (k+1)(l+1)}{2N+1}$	$k,l = [0, N-1]$
DST-VI:	$[\mathbf{S}_N^{VI}]_{k,l} = \sin\frac{2\pi (k+1)(l+\frac{1}{2})}{2N+1}$	$k,l = [0, N-1]$
DST-VII:	$[\mathbf{S}_N^{VII}]_{k,l} = \sin\frac{2\pi (k+\frac{1}{2})(l+1)}{2N+1}$	$k,l = [0, N-1]$
DST-VIII:	$[\mathbf{S}_N^{VIII}]_{k,l} = \sin\frac{2\pi (k+\frac{1}{2})(l+\frac{1}{2})}{2N-1}$	$k,l = [0, N-1]$

Furthermore, it is worth recalling that the following relationships hold true:

$$\mathbf{C}_N^V = (\mathbf{C}_N^V)^T = (\mathbf{C}_N^V)^{-1} \tag{1}$$

$$\mathbf{C}_N^{VIII} = (\mathbf{C}_N^{VIII})^T = (\mathbf{C}_N^{VIII})^{-1} \tag{2}$$

Stemming from these properties of the DCT-V and DCT-VIII, for which the inverse transforms are equal to the corresponding forward ones, one can note that the low-complexity factorizations proposed in this current work for the forward transforms serve for the inverse computations as well.

III. Odd Type Transforms Relationships

In order to derive low-complexity factorizations for the DCT-V and the DCT-VIII, the DCT-II mapping over the DFT and the decomposition of the $2N+1$-point DCT-II are briefly reported. Moreover, this Section introduces a new class of relationships among odd type transforms, which is used to prove the connections that link the DCT-V and the DCT-VIII to the DCT-VI and the DST-VII respectively.

A. Mapping of the DCT-II on the DFT

As explained in [16], it is known that all the DCTs and DSTs can be viewed as special cases of the DFT, with real-valued inputs and limited set of outputs. In particular, Vetterli and Nussbaumer [7] showed how to map the N-point DCT-II over a $4N$-point DFT of real inputs, where the even-indexed inputs are zero and where only one fourth of the outputs is computed.

A more strict relationship presented in [21] shows that an odd-length DCT-II can be derived from the real-valued DFT of the same length, namely:

$$\mathbf{C}_{2N+1}^{II} = \mathbf{H}_1 \cdot \begin{bmatrix} \Re[\mathbf{F}_{2N+1}]_{\text{rows 0,...,N}} \\ \Im[\mathbf{F}_{2N+1}]_{\text{rows N+1,...,2N}} \end{bmatrix} \cdot \mathbf{H}_2, \tag{3}$$

where $\Re(\mathbf{F}_{2N+1})$ and $\Im(\mathbf{F}_{2N+1})$ denote the real and imaginary part of the $2N+1$-point DFT respectively, while $\mathbf{H}_1$ and $\mathbf{H}_2$ are sign-inversion and permutation matrices. Thus, the DCT-II input sequence $x(n)$ must be reordered to generate a new sequence $\hat{x}(n)$, which is the input to the real-valued DFT. Therefore, the $\mathbf{H}_2$ matrix is described by the following permutation:

$$\hat{x}(n) = \begin{cases} x\left(\gamma_{n,N} \cdot n + N\right) & n = [0, N] \\ x\left(\gamma_{n,N} \cdot (2N+1-n) + N\right) & n = [N+1, 2N] \end{cases}, \tag{4}$$

where $\gamma_{n,N} = (-1)^{n+N+1}$. In the following we will refer to the sets of indices defined by (4) as Heidemann indices $n = [0, N]$ and Heidemann indices $n = [N+1, 2N]$ respectively. Then, the DFT of $\hat{x}(n)$ produces $\hat{y}(n)$ and the final DCT-II result ($y(n)$) is extracted from $\hat{y}(n)$ by using the following relation, which implements $\mathbf{H}_1$:

$$\begin{aligned} y(2n) &= (-1)^n \cdot \Re[\hat{y}(n)] & n &= [0, N] \\ y(2N+1-2n) &= (-1)^{n+1} \cdot \Im[\hat{y}(n)] & n &= [1, N] \end{aligned}. \tag{5}$$

According to the mapping proposed in [21], it is possible to efficiently compute the DCT-II using low-complexity DFT algorithms already available in the literature [22]–[24].

B. Decomposition of the DCT-II by means of the DCT-VI and the DST-VII Transforms

The second interesting relationship is the decomposition of the $2N+1$-point DCT-II by means of the $N+1$-point DCT-VI and the N-point DST-VII, which has been shown in [20]:

$$\mathbf{C}_{2N+1}^{II} = \mathbf{Q}_{2N+1} \cdot \begin{bmatrix} \mathbf{C}_{N+1}^{VI} & \\ & \mathbf{S}_N^{VII} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_N & \mathbf{J}_N \\ & 1 \\ -\mathbf{J}_N & \mathbf{I}_N \end{bmatrix}, \tag{6}$$

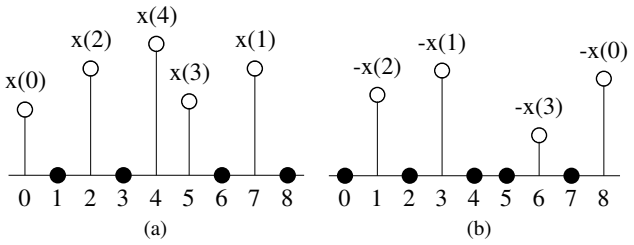

Fig. 1. Input mapping of the 5-point DCT-VI (a) and of the 4-point DST-VII (b) on the 9-point DCT-II.

where $\mathbf{I}_N$ and $\mathbf{J}_N$ are the N-order identity and anti-diagonal identity matrices respectively, while $\mathbf{Q}_{2N+1}$ is a sign alteration and reordering matrix, which acts as in the following:

$$\begin{aligned} \hat{x}(2n) &= x(n) && n = [0, N] \\ \hat{x}(2n+1) &= (-1)^n \cdot x(N+1+n) && n = [0, N-1] \end{aligned} . \quad (7)$$

Reznik [20] derived low-complexity factorizations for the $N+1$-point DCT-VI and for the N-point DST-VII by combining the decomposition of (6) with the mapping of (3) and by relying to the DFT algorithms in [22]–[24].

From these observations, it is straightforward to find the mapping that allows to exactly compute the DCT-VI and the DST-VII transforms from a DCT-II, hence from the DFT. Specifically, the $N+1$-point DCT-VI can be calculated as the $2N+1$-point DCT-II with zeros on the inputs with the Heidemann indices $n = [N+1, 2N]$, and taking the even-indexed outputs. An example of this approach is shown in Fig. 1a, which illustrates the input mapping of a 5-point DCT-VI on a 9-point DCT-II, where the five inputs of the DCT-VI, marked as white dots, are interleaved with zeros, denoted as black dots. On the other hand, the N-point DST-VII can be calculated by means of the $2N+1$-point DCT-II with zeros on the inputs with the Heidemann indices $n = [0, N]$, and taking the odd-indexed outputs. The input mapping of a 4-point DST-VII on a 9-point DCT-II is reported in Fig. 1b.

C. Relationships Between Odd Type Transforms

The aim of this Section is to show a class of relationships which hold among different odd type DCTs and DSTs, with a particular focus on the connections which link the DCT-VI and the DST-VII with the DCT-V and the DCT-VIII respectively. All the relationships belonging to this class are derived by applying the trigonometric identity of the cosine of the difference of angles. Four relationships between the input and the output N-order transform matrices $\mathbf{X}_N$ and $\mathbf{Y}_N$ hold true:

$$\mathbf{Y}_N = \mathbf{D}_N \cdot \mathbf{X}_N \cdot \mathbf{J}_N, \quad (8)$$

$$\mathbf{Y}_N = \mathbf{J}_N \cdot \mathbf{X}_N \cdot \mathbf{D}_N, \quad (9)$$

$$\mathbf{Y}_N = (\mathbf{D}_N \mathbf{J}_N) \cdot \mathbf{X}_N \cdot (\mathbf{D}_N \mathbf{J}_N), \quad (10)$$

$$\mathbf{Y}_N = (\mathbf{J}_N \mathbf{D}_N \mathbf{J}_N) \cdot \mathbf{X}_N \cdot (\mathbf{D}_N \mathbf{J}_N \mathbf{D}_N), \quad (11)$$

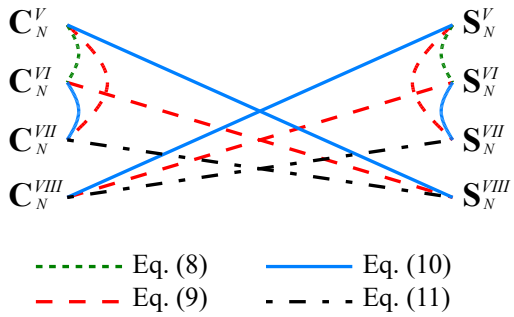

Fig. 2. Relationships between odd type DCT and DST transforms.

where $\mathbf{J}_N$ is the anti-diagonal identity matrix of order N and $\mathbf{D}_N$ is the diagonal matrix implementing sign-alternation:

$$\mathbf{D}_N = \begin{bmatrix} 1 & & & & \\ & -1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & -1 \end{bmatrix} . \quad (12)$$

Fig. 2 shows a diagram, where all the relationships between odd type DCTs and DSTs are highlighted. It is worth noting that all the relationships are bidirectional, thus the output matrix $\mathbf{Y}_N$ can be any of the two transforms linked by one connection. As it can be observed from Fig. 2, the relationship in (10) is obtained as the combinations of (8) and (9), while (11) is derived using (10) and (9).

Example (connection between $\mathbf{C}_N^{VI}$ and $\mathbf{C}_N^V$)**.** *According to (8), the following relationship holds true:*

$$\mathbf{C}_N^V = \mathbf{D}_N \cdot \mathbf{C}_N^{VI} \cdot \mathbf{J}_N. \quad (13)$$

Proof. Let us consider:

$$\alpha = \frac{2\pi k(N - \frac{1}{2})}{2N-1}, \quad \beta = \frac{2\pi k(N - l - \frac{1}{2})}{2N-1} \quad (14)$$

and the trigonometric identity:

$$\cos(\alpha - \beta) = \cos(\alpha) \cdot \cos(\beta) + \sin(\alpha) \cdot \sin(\beta), \quad (15)$$

which becomes (16), where the term $\sin(k\pi)$ is equal to zero when k is integer in the range $[0, N-1]$. Moreover, i) $\cos(k\pi) = (-1)^k$ represents the sign-alternation described by the $\mathbf{D}_N$ matrix and ii) by introducing $m = N - l - 1$ the last cosine term becomes the product between

$$[\mathbf{C}_N^{VI}]_{k,m} = \cos \frac{2\pi k(m + \frac{1}{2})}{2N-1}, \quad (17)$$

with $k, m = [0, N-1]$ and the reverse-order permutation $\mathbf{J}_N$. □

Using the same method, it is straightforward to prove also (9). Then the relationships in (10) and (11) can be easily derived as combinations of the previous ones. According to (11), the following relationship between the DCT-VIII and the DST-VII holds true:

$$\mathbf{C}_N^{VIII} = (\mathbf{J}_N \mathbf{D}_N \mathbf{J}_N) \cdot \mathbf{S}_N^{VII} \cdot (\mathbf{D}_N \mathbf{J}_N \mathbf{D}_N). \quad (18)$$

$$\cos\left(\frac{2\pi kl}{2N-1}\right) = \cos(k\pi)\cdot\cos\left(\frac{2\pi k(N-l-\frac{1}{2})}{2N-1}\right) + \sin(k\pi)\cdot\sin\left(\frac{2\pi k(N-l-\frac{1}{2})}{2N-1}\right)$$
$$= (-1)^k\cdot\cos\left(\frac{2\pi k(N-l-\frac{1}{2})}{2N-1}\right) \qquad k,l=[0,N-1] \tag{16}$$

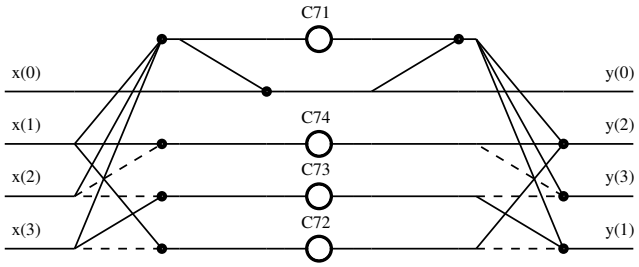

Fig. 4. SFG of 4-point DCT-V.

IV. Low-Complexity Factorizations for Computing DCT-V and DCT-VIII

In this Section we exploit the relationships among DCTs and DSTs introduced in Section III to derive low-complexity factorizations for the DCT-V and DCT-VIII transforms of length N = 4 and 8.

A. DCT-V SFGs

In order to derive the SFG of the 4-point DCT-V, we start from the real-valued 7-point DFT [24] and we apply the decomposition in (3) to obtain the 7-point DCT-II, which is depicted in Fig. 3. Then, the 4-point DCT-VI, denoted by black lines in the figure, is extracted from the 7-point DCT-II SFG according to (6). Finally, the 4-point DCT-V is obtained by reordering the inputs and applying the sign-alternation to the 4-point DCT-VI, as indicated in (13). The resulting SFG is reported in Fig. 4, which needs only 4 multiplications and 13 additions with respect to 9 multiplications and 12 additions required by the direct implementation of the MVM algorithm. Table I reports the coefficients C71-C78 in Fig. 3 and 4, which are the Winograd DFT factors taken from the low-complexity algorithms available at [24].

Concerning the derivation of the 8-point DCT-V, the 15-point DFT and the same length DCT-II are needed. Since 15 is not a prime number, the Prime Factor Algorithm (PFA) in [23], [25] has been employed. Assuming N_1 and N_2 mutually prime numbers, the PFA technique allows to compute a transform of composite length $N = N_1 \cdot N_2$ as the cascade of N_1 and N_2 stages, where the first N_1 stages are N_2-point transforms whereas the second N_2 stages are N_1-point transforms, with proper input mapping and $N_1\cdot N_2 - N_1 - N_2 + 1$ final additions. As shown in Fig. 5, the 15-point DCT-II is derived using the SFG of the 3-point and the 5-point DCT-II [4], which factors are listed in Table I. Observing that the $N+1$-point DCT-VI can be derived from the $2N+1$-point DCT-II with zeros on the inputs corresponding to Heidemann indices $n = [N+1, 2N]$,

TABLE I
Constant Coefficients from DFT Factorizations in [24].

Coefficient	Value	Coefficient	Value
C31	0.86602540	C71	-1.16666667
C32	1.50000000	C72	-0.79015647
		C73	0.05585427
C51	0.95105652	C74	0.73430220
C52	-1.53884180	C75	0.44095855
C53	-0.36327126	C76	-0.34087293
C54	-0.55901699	C77	0.53396936
C55	-1.25000000	C78	0.87484229

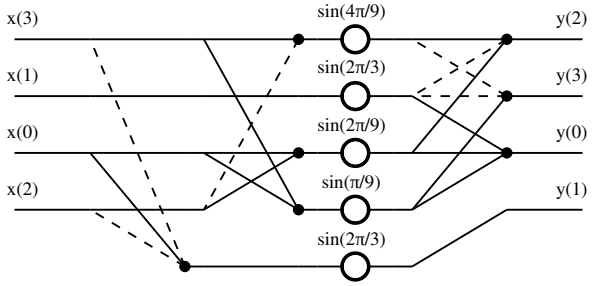

Fig. 6. SFG of 4-point DCT-VIII.

and taking the even-indexed outputs only, we have deleted all the paths coming from the null inputs or connected to unused outputs, which are denoted using gray lines in Fig. 5. The resulting 8-point DCT-VI SFG is highlighted in black in Fig. 5. Finally the 8-point DCT-V could be obtained by applying (13). This implementation requires only 16 multiplications and 36 additions instead of 28 multiplications and 56 additions.

B. DCT-VIII SFGs

The SFGs of the 4-point and the 8-point DCT-VIII are derived directly from the DST-VII SFGs in [18] by applying input and output sign-inversion and reordering, as specified by (18). For brevity, only the SFG of the 4-point DCT-VIII is reported in Fig. 6, where only 5 multiplications and 11 additions are needed with respect to the 11 multiplications and 11 additions required by the reference MVM algorithm. On the other hand, the SFG of the 8-point DCT-VIII requires only 21 multiplications and 77 additions instead of 64 and 56 respectively.

V. Implementation Results

The proposed low-complexity DCT-V and DCT-VIII SFGs have been described in Matlab for functional validation [26]. Then, they have been implemented in VHDL using the HDL Coder toolbox, which translates Matlab scripts into VHDL

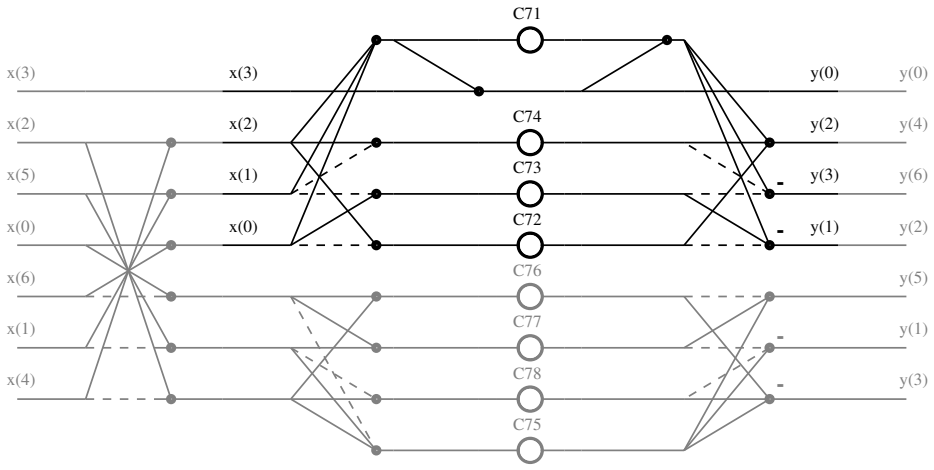

Fig. 3. SFG of 7-point DCT-II and 4-point DCT-VI (highlighted in black). Dashed lines represent inputs to be subtracted.

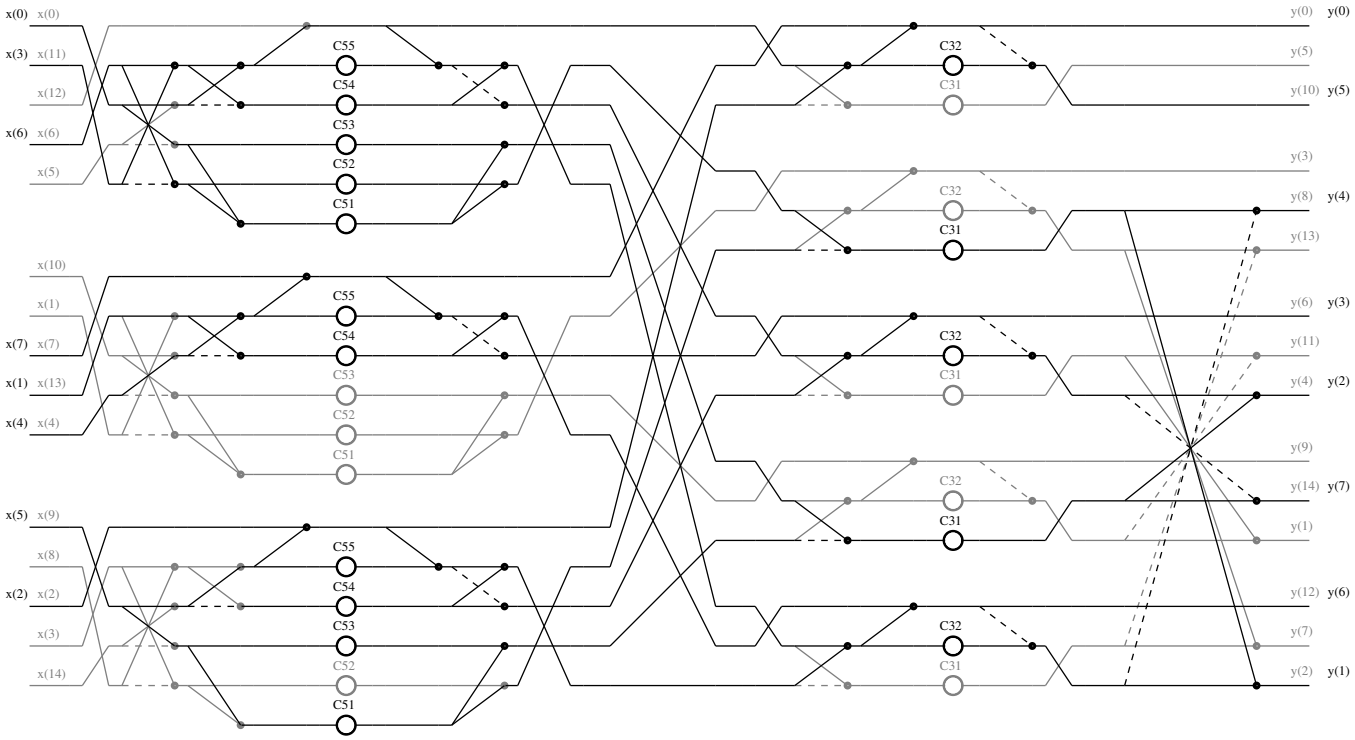

Fig. 5. SFG of 15-point DCT-II and 8-point DCT-VI (highlighted in black).

code, and finally synthesized on an Altera Cyclone II FPGA (device EP2C35F672C6) using Quartus II.

The proposed hardware implementations have been designed to work as accelerators of the one-dimensional transform functions of the reference software developed by the JVET group, named Joint Exploration Model (JEM) [27]. Therefore, the internal signal representation has been chosen in order to be compliant with the reference model. The JEM specifies to use integers on 16 bits to represent the input samples and the output transform coefficients of the designed module and to use 32 bits for the internal signals. To determine the fractional representation of the internal coefficients, we employed the Matlab Fixed-Point Designer, which automatically selects the number of fractional bits for each signal, given the internal word-length.

We compare the implementation results of our proposed architectures with the reference ones, which implements the corresponding MVM algorithm. It is worth noting that the MVM algorithm adopted in this work exploits the symmetries of the coefficients, which compose the transform matrices, indeed reducing the number of additions and multiplications with respect to the $N \cdot (N-1)$ additions and N^2 multipli-

TABLE II
SYNTHESIS RESULTS OF THE MVM BASED IMPLEMENTATION AND THE PROPOSED DCT-V AND DCT-VIII ARCHITECTURES.

Design		Complexity		Synthesis Results		
		Add	Mult	LEs	f_{CK}(MHz)	ADP
DCT-V N=4	Proposed	13	4	1809	47.32	38.23
	MVM	12	9	2926	54.39	53.80
DCT-V N=8	Proposed	36	16	5390	26.21	205.65
	MVM	56	28	9582	42.67	224.56
DCT-VIII N=4	Proposed	11	5	2086	52.78	39.52
	MVM	11	11	4604	57.78	79.68
DCT-VIII N=8	Proposed	77	21	9462	26.7	354.38
	MVM	56	64	20476	45.13	453.71

cations of the MVM approach without optimization. Table II reports the number of additions and multiplications, the number of logic elements (LEs), the maximum achievable clock frequency (f_{CK}) and the Area-Delay Product (ADP) of each implementation. By observing the number of LEs, the proposed schemes nearly halve the required hardware complexity of both 4-point and 8-point DCT-V and DCT-VIII. This because the number of LEs mainly depends on the number of multipliers, which has been significantly reduced with respect to the MVM based implementation. On the other hand, the MVM based architectures achieve higher clock frequency, hence reduced delay. This is due to the fact that the critical path of the MVM implementations is always composed of only one multiplier plus an addition tree, while several multipliers and adders are cascaded in the proposed low-complexity implementations. However, the proposed schemes are more efficient, featuring lower ADP as highlighted in the last column of Table II.

VI. CONCLUSION

In this paper we have introduced a new class of relationships among odd type DCTs and DSTs. These relationships allow to reuse the already known factorizations of the DCT-VI and DST-VII to obtain all the other odd type transforms by simply applying permutations and sign inversions. Then, low-complexity SFGs for the 4-point and 8-point DCT-V and DCT-VIII have been derived resorting to the Winograd DFT factorization and applying the aforementioned relationships. Finally, we have implemented the proposed scheme as accelerators of the transform functions of the future video coding technology, thus showing lower hardware complexity and improved ADP with respect to the reference MVM based implementations. Future works include the design of a reconfigurable unit for different DCT and DST calculations exploiting multiplexers and integer coefficients and the evaluation of the timing and the rate-distortion performance within the video codec.

REFERENCES

[1] N. Ahmed, T. Natarajan, and K. Rao, "Discrete Cosine Transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan 1974.

[2] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze, and M. Sadafale, "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1029–1041, Dec 2013.

[3] V. Britanak, P. C. Yip, and K. R. Rao, *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations*. Elsevier, Sep. 2006.

[4] G. Bi and Y. Zeng, *Transforms and Fast Algorithms for Signal Analysis and Representations*. Springer Science & Business Media, 2003.

[5] W.-H. Chen, C. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. Commun.*, vol. 25, no. 9, pp. 1004–1009, Sep 1977.

[6] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 4, pp. 803–816, Aug 1984.

[7] M. Vetterli and H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal processing*, vol. 6, no. 4, pp. 267–278, 1984.

[8] B. Lee, "A new algorithm to compute the discrete cosine Transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 6, pp. 1243–1245, Dec 1984.

[9] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 10, pp. 1455–1461, Oct 1987.

[10] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, May 1989, pp. 988–991.

[11] J. Han, A. Saxena, and K. Rose, "Towards jointly optimal spatial prediction and adaptive transform in video/image coding," in *Proc. 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 726–729.

[12] A. Saxena and F. C. Fernandes, "DCT/DST-Based Transform Coding for Intra Prediction in Image/Video Coding," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3974–3981, Oct 2013.

[13] J. Chen, Y. Chen, M. Karczewicz, X. Li, H. Liu, L. Zhang, and X. Zhao, "Coding tools investigation for next generation video coding based on HEVC," in *Proc. SPIE 9599, Applications of Digital Image Processing XXXVIII*, no. 95991B, Sept 2015.

[14] X. Zhao, J. Chen, M. Karczewicz, X. Li, and C. Wei-Jung, "Enhanced Multiple Transform for Video Coding," in *Proc. 2016 Data Compression Conference*, 2016, pp. 73–82.

[15] T. Biatek, V. Lorcy, P. Castel, and P. Philippe, "Low-complexity adaptive multiple transforms for post-HEVC video coding," in *Picture Coding Symposium*, 2016, pp. 1–5.

[16] X. Shao and S. G. Johnson, "Type-II/III DCT/DST algorithms with reduced number of arithmetic operations," *Signal Processing*, vol. 88, no. 6, pp. 1553–1564, 2008.

[17] M. Puschel and J. M. F. Moura, "Algebraic Signal Processing Theory: Cooley-Tukey Type Algorithms for DCTs and DSTs," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1502–1521, April 2008.

[18] R. K. Chivukula and Y. A. Reznik, "Fast Computing of Discrete Cosine and Sine Transforms of Types VI and VII," in *Proc. SPIE 8135, Applications of Digital Image Processing XXXIV*, no. 813505, Sept 2011, pp. 1–10.

[19] A. Saxena, F. C. Fernandes, and Y. A. Reznik, "Fast Transforms for Intra-prediction-based Image and Video Coding," in *Proc. 2013 Data Compression Conference*, March 2013, pp. 13–22.

[20] Y. A. Reznik, "Relationship between DCT-II, DCT-VI, and DST-VII transforms," in *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 5642–5646.

[21] M. T. Heideman, "Computation of an odd-length DCT from a real-valued DFT of the same length," *IEEE Trans. Signal Process.*, vol. 40, no. 1, pp. 54–61, Jan 1992.

[22] S. Winograd, "On Computing the Discrete Fourier Transform," *Mathematics of computation*, vol. 32, no. 141, pp. 175–199, 1978.

[23] C. Burrus and P. Eschenbacher, "An in-place, in-order prime factor FFT algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 4, pp. 806–817, Aug 1981.

[24] C. Burrus. (2009, Sept.) Appendix 4: Programs for Short FFTs. OpenStax CNX. [Online]. Available: http://cnx.org/contents/78bef672-dc29-4fb8-86e5-6a21ad72ea0c@4

[25] P. Yang and M. Narasimha, "Prime factor decomposition of the discrete cosine transform and its hardware realization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '85.*, vol. 10, Apr 1985, pp. 772–775.

[26] M. Masera. (2017, May) Type V-VIII DCT Matlab Scripts. [Online]. Available: http://personal.det.polito.it/maurizio.masera/downloads.html

[27] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, *HM-16.6-JEM-4.0 Reference Software Model*. [Online]. Available: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-4.0/