Adaptive Approximated DCT Architectures for HEVC

Adaptive Approximated DCT Architectures for HEVC

Maurizio Masera, *Student Member, IEEE,* Maurizio Martina, *Senior Member, IEEE,* and Guido Masera, *Senior Member, IEEE*

***Abstract*—This paper proposes a flexible and efficient implementation of the two-dimensional N-point Discrete Cosine Transform (DCT) for the High Efficiency Video Coding (HEVC) standard. The DCT is implemented through the Walsh-Hadamard Transform (WHT) followed by Givens rotations. This scheme is exploited to derive an adaptive algorithm, which allows to compute four different approximations ranging from the complete DCT to the WHT, by selectively skipping some rotations. The work shows the statistical analysis of the DCT usage and derives a pre-computation mechanism to adaptively skip rotations. Each approximation, referred to as operating mode, is characterized by a large saving of operations, at the expense of very small quality loss. Then, two 2D-DCT architectures are proposed: the first one is totally unfolded while the second one is folded. The two designs are finally synthesized with a 90-nm standard-cell library for a clock frequency of 250 MHz. Both architectures support real-time processing of 8K UHD video sequences at 64 and 26 fps respectively and show higher throughput and lower gate count compared to state-of-art implementations. Moreover, power saving ranging from 28% to 56% can be achieved by working within the proposed operating modes.**

***Index Terms*—Discrete Cosine Transform (DCT), H.265, High Efficiency Video Coding (HEVC), video coding.**

I. Introduction

THE High Efficiency Video Coding (HEVC) is the latest video coding standard jointly developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) [1]. According to [2], the HEVC standard is able to obtain a bitrate reduction of about the 50% while maintaining the same visual quality produced by the previous Advanced Video Coding (AVC) standard. In order to achieve this saving, the standard exploits a large number of new features and tools such as new structures for recursive partitioning, new intra and inter prediction modes and larger transform unit sizes. However, the resulting improvement in terms of video compression, comes at the expense of increasing the encoder complexity by about 40%-70% with respect to the AVC, due to the exploration of a large space of possible encoder decisions [3], [4]. Moreover, the challenge to design optimized area and power-efficient hardware modules becomes more evident for such devices, as mobile phones and cameras, where the chip has to incorporate a lot of functions and the battery lifetime is limited. In particular, one of the key features of HEVC is the variable size transform computation [5]. The standard exploits the Discrete Cosine Transform (DCT) [6], which can be applied to blocks made of $N \times N$ samples, where N can be: 4, 8, 16 or 32.

Some existing works have proposed hardware architectures for the DCT computation in the context of image and video compression, as the HEVC standard [7]. These works provide both exact and approximated DCT computations, where the quality is traded for a reduction of the computational complexity. Among the architectures proposed in the literature for HEVC transforms, the one introduced by Shen *et al.* [8] uses the multiple constant multiplication (MCM) for the four-point and eight-point DCT while it adopts shared multipliers for DCTs of larger-size. Park *et al.* [9] have exploited the Chen's factorization [10] of the DCT implementing each butterfly operation by means of multiplierless processing elements. Zhu *et al.* [11] proposed a pipelined unit, which is able to compute the forward and inverse DCT as well as the Hadamard Transform (HT) by reusing small-size transform hardware for other larger-size ones. Similarly, Budagavi *et al.* [12] exploited symmetry properties of the forward and inverse HEVC transform matrices to allow resources sharing in a unified architecture. Meher *et al.* [13] proposed a flexible architecture, which is able to compute the DCT for any of the four different N values with the same throughput. Their work exploits the partial butterfly approach, where the one-dimensional N-point DCT can be calculated recursively by means of an $N/2$-point DCT and an $N/2 \times N/2$ matrix multiplication. Another type of hardware architecture is the one proposed by Ahmed *et al.* [14], which is inspired by the factorization proposed in [15], [16], *i.e.* exploiting the Walsh-Hadamard Transform (WHT) followed by a set of Givens rotations. In [17]–[19] an algebraic integer-based scheme, which exploits the Arai's factorization [20], is proposed for the exact computation of the 8×8 DCT. Among the approximated ones, Bouguezel *et al.* [21], [22] provided a parametric 8-point DCT matrix, which allows to generate different transforms with low-complexity, and also defined approximated DCT matrices for $N > 4$ [23]. Bayer *et al.* [24] obtained the eight-point transformation matrix by rounding-off to zero or one each entry of the original DCT matrix. In [25] some entries of the DCT matrix in [24] have been set to zero, thus producing an approximated DCT, which requires 14 additions only. Starting from the matrix of [25], Cintra *et al.* [26] applied frequency-domain pruning to obtain an approximated DCT. In particular, they removed the signal-flows related to those DCT coefficients which are likely

The authors are with the Electronics and Telecommunications Department - Politecnico di Torino, 10129 Torino, Italy (e-mail: maurizio.masera@polito.it; maurizio.martina@polito.it; guido.masera@polito.it).

discarded by the video compression quantization step. All the previous 8×8 DCT approximations have been implemented and compared by Potluri *et al.* in [27], where a novel eight-point DCT matrix has been also introduced.

As pointed out in [13], since the HEVC standard supports DCT of different sizes, a flexible and reusable architecture is required. Furthermore, the challenge to design optimized architectures for area and power has encouraged us to define a new method to trade dynamic power for small losses on the reconstructed video quality. Unlike [21]–[27], where only fixed approximations of the transform matrix of size 8×8 are shown, we propose a content-adaptive DCT scheme, which is applied to all the DCT sizes defined in HEVC. Moreover, four operating modes are defined in order to allow the designer to select a different trade-off between video quality and power consumption.

The contributions of this work are: i) the statistical analysis of the DCT usage during the encoding process; ii) the description of a new algorithm to dynamically choose which rotations will be performed; iii) the design of two flexible architectures, which have been sized resorting to the calculated statistics and a practical method to select a proper folding degree for a target application.

The rest of the paper is organized as follows. In Section II the adopted DCT factorization over different lengths is briefly reported. In Section III the results of the statistical analysis are shown and the proposed operating modes are defined. In particular, Section IV reports the results of the architectural space exploration achieved by resorting to the folding technique [28]. Then, two hardware implementations of the DCT are proposed: the first one is designed to achieve the highest throughput, the second one improves resource utilizations and reduces the required area. Finally, Section V reports the synthesis and the power estimation results while Section VI concludes this paper.

II. The WHT-based DCT

A. DCT Factorization

According to the property of separability the two-dimensional DCT of a matrix of size $N \times N$ pixels (2D-DCTN) can be decomposed in two one-dimensional DCTs of length N, which are performed row-wise and column-wise (or viceversa). Therefore, in the following only the one-dimensional DCT (1D-DCTN) is addressed. According to [29] the 1D-DCTN can be computed as follows:

$$\mathbf{X} = \mathbf{C}_N \cdot \mathbf{x}, \tag{1}$$

where $\mathbf{X} = (X_0, \ldots, X_{N-1})$ is the column vector of output results, $\mathbf{x} = (x_0, \ldots, x_{N-1})$ is the column vector containing input samples and $\mathbf{C}_N$ is the DCT matrix. As suggested in [14], the DCT matrix can be factorized as:

$$\mathbf{C}_N = \frac{1}{\sqrt{N}} \cdot \mathbf{B}_N \cdot \mathbf{T}_N \cdot \mathbf{B}_N \cdot \mathbf{W}_N, \tag{2}$$

where $\mathbf{W}_N$ is the Walsh ordered WHT matrix, which is generated by applying the bit reverse and Gray coding ordering to the row indices of the N-order Hadamard matrix

$$\mathbf{H}_N = \begin{pmatrix} \mathbf{H}_{N/2} & \mathbf{H}_{N/2} \\ \mathbf{H}_{N/2} & -\mathbf{H}_{N/2} \end{pmatrix}, \tag{3}$$

where $\mathbf{H}_1 = 1$. The other two matrices in (2) are the bit-reversal matrix $\mathbf{B}_N$ and a block diagonal matrix $\mathbf{T}_N$, which contains the Givens rotations. The latter can be defined through the following recursion:

$$\mathbf{T}_N = \begin{pmatrix} \mathbf{T}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_{N/2} \end{pmatrix}, \tag{4}$$

where $\mathbf{T}_2$ is the identity matrix of size 2, and $\mathbf{U}_{N/2}$ is the product of two permutation matrices and the Givens rotation matrices, namely

$$\mathbf{U}_{N/2} = \mathbf{B}_{N/2} \cdot \mathbf{V}_{N/2,3} \cdot \ldots \cdot \mathbf{V}_{N/2,q} \cdot \ldots \cdot \mathbf{V}_{N/2,m} \cdot \mathbf{B}_{N/2}, \tag{5}$$

with $m = \log_2 N + 1$, $3 \leq q \leq m$ and

$$\mathbf{U}_2 = \begin{pmatrix} \cos(\pi/8) & \sin(\pi/8) \\ -\sin(\pi/8) & \cos(\pi/8) \end{pmatrix}. \tag{6}$$

The Givens rotation matrices in (5) are defined for $3 \leq q \leq m$ and they are composed of $r = m - q + 1$ sub-matrices placed on the diagonal:

$$\mathbf{V}_{N/2,q} = \begin{pmatrix} \mathbf{V}_{N/2^r,q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{N/2^r,q} \end{pmatrix}, \tag{7}$$

where

$$\mathbf{V}_{N/2^r,q} = \begin{pmatrix} c_{1,q} & 0 & \cdots & \cdots & 0 & s_{1,q} \\ 0 & \ddots & 0 & 0 & \iddots & 0 \\ \vdots & 0 & c_{p,q} & s_{p,q} & 0 & \vdots \\ \vdots & 0 & -s_{p,q} & c_{p,q} & 0 & \vdots \\ 0 & \iddots & 0 & 0 & \ddots & 0 \\ -s_{1,q} & 0 & \cdots & \cdots & 0 & c_{1,q} \end{pmatrix}, \tag{8}$$

and p is an odd positive integer lower than $N/2^r$. The coefficients $c_{p,q}$ and $s_{p,q}$, which are placed in a concentric square way, perform plane rotations and the rotation angle is identified by the couple of indices (p, q) as:

$$c_{p,q} = \cos\left(\frac{p \cdot \pi}{2^q}\right) \qquad s_{p,q} = \sin\left(\frac{p \cdot \pi}{2^q}\right). \tag{9}$$

Noticeably, each rotation can be decomposed in three lifting steps [30] to reduce the computational complexity of the algorithm:

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & \mathcal{P}_\theta \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ \mathcal{U}_\theta & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & \mathcal{P}_\theta \\ 0 & 1 \end{pmatrix}, \tag{10}$$

where $\mathcal{P}_\theta = (1 - \cos\theta)/\sin\theta$ and $\mathcal{U}_\theta = -\sin\theta$.

B. Hardware Oriented Optimization

The computational complexity of a 1D-DCTN, factorized by means of the WHT, is determined by

$$B = \frac{N}{2} \cdot \log_2 N \tag{11}$$

butterfly operators, for the HT, and by

$$R = 1 + \frac{N}{2} \cdot (\log_2 N - 2) \tag{12}$$

TABLE I
ACCURACY MEASURES AND ARITHMETIC COMPLEXITY OF 1D-DCT8 APPROXIMATIONS.

Method	ϵ	MSE ($\times 10^{-2}$)	C_g	η	MULT	ADD	SHIFT
Exact DCT	0.0000	0.0000	8.826	93.991	64	56	0
HM [31]	0.0020	0.0009	8.829	93.825	20	28	2
Proposed DCT [32]	0.0004	0.0003	8.827	93.949	15	39	0
WHT	5.0494	2.5112	7.946	85.314	0	24	0
BAS-2008 [21]	5.9293	2.3783	8.119	86.863	0	18	2
BAS-2011 a=0 [22]	26.8642	7.1040	7.912	85.642	0	16	0
BAS-2011 a=1 [22]	26.8642	7.1025	7.913	85.380	0	18	0
BAS-2011 a=2 [22]	27.9224	7.8318	7.763	84.767	0	18	2
CB-2011 [24]	1.7945	0.9800	8.183	87.430	0	22	0
Modified CB-2011 [25]	8.6592	5.9389	7.333	80.897	0	14	0
Improved Modified CB-2011 [27]	11.3128	7.8987	7.333	80.897	0	14	0

TABLE II
APPROXIMATED VALUES OF LIFTING COEFFICIENTS.

Givens rotation	a	Adders	Shifters	b	Adders	Shifters
$\pi/8$	51	2	2	98	2	3
$\pi/16$	25	2	2	50	2	3
$3\cdot\pi/16$	78	2	3	142	2	3
$\pi/32$	13	2	2	25	2	2
$3\cdot\pi/32$	38	2	3	74	2	3
$5\cdot\pi/32$	64	0	1	121	2	2
$7\cdot\pi/32$	92	2	3	162	2	3
$\pi/64$	6	1	2	13	2	2
$3\cdot\pi/64$	19	2	2	38	2	3
$5\cdot\pi/64$	32	0	1	62	1	2
$7\cdot\pi/64$	44	2	3	86	3	4
$9\cdot\pi/64$	57	2	2	109	3	3
$11\cdot\pi/64$	71	2	2	132	1	2
$13\cdot\pi/64$	85	2	2	152	2	3
$15\cdot\pi/64$	99	2	2	172	3	4

Givens rotations. One butterfly is composed of two adders, while one rotation is implemented by means of the lifting scheme, which is composed of three stages, as in (10), each of which requires one multiplication and one addition. According to the approach suggested in [33], lifting coefficients are expressed as:

$$\frac{1-\cos\theta}{\sin\theta} \approx \frac{a}{2^n}, \qquad \sin\theta \approx \frac{b}{2^n}. \tag{13}$$

The first architecture proposed in this paper relies on an unfolded 1D-DCT data-flow, which takes advantage of a and b values to simplify the multipliers required in the lifting scheme, by exploiting the Reduced Adder Graph (RAG-n) technique [34]. By representing the coefficients with $n = 8$ bits, we first evaluated the matrix proximity metrics and the transform-related measures defined in [24], [29], namely the error energy (ϵ), the mean square error (MSE), the coding gain (C_g) and the transform efficiency (η) for the case 1D-DCT with N = 8. As shown in the first part of Table I, the proposed fixed-point DCT approximates very well the exact DCT and the one adopted in the HEVC reference software [31]. Moreover, as expected, the WHT is less accurate than the proposed DCT. The values assumed by a and b, as well as the number of adders and shifters required to implement each coefficient through the RAG-n representation, are reported in Table II.

C. DCT Algorithm Comparison

Since this work deals with DCT complexity reduction and performance trade-offs, it is important to compare the WHT-based DCT with other exact and approximated algorithms. To the best of our knowledge the architecture proposed in [13] is the best performing one for the HEVC standard, being able to support all the DCT sizes. Thus, Table III compares the computational complexity, in terms of multiplications, additions and shifts, of the 1D-DCTN algorithm in [13] with the WHT-based one, proposed in [14], and its multiplierless version, obtained by applying the RAG-n technique to the lifting scheme coefficients. As it can be observed the WHT-based factorization requires less multiplications than the Partial Butterfly implementation for every DCT size, especially for large-sizes. On the other hand, considering multiplierless implementations, the MCM-based one is better than the WHT-based one for all the DCT sizes, excepting the case N = 32. However, since the WHT can be seen as a very simple approximation of the DCT, the WHT-based DCT features the possibility to adapt the accuracy of the computation to current data, being an interesting alternative for hardware implementation. This property is exploited in this current work to design content-based approximated DCTs

It is known that approximated DCT algorithms trade hardware complexity for accuracy. As discussed in Section II-B and shown in the first part of Table I, the proposed fixed-point implementation of the WHT-based DCT features excellent matrix proximity and transform-related accuracy. As a consequence, it is important to assess the complexity-accuracy trade-off of approximated solutions to make a fair comparison. The second part of Table I extends the matrix proximity metrics and transform-related measures to the solutions proposed in [21], [22], [24], [25], [27]. As it can be observed, the multiplierless approximations in [21], [22], [24], [25], [27] obtain a lower arithmetic complexity than the proposed WHT-based DCT at the cost of lower accuracy. Thus, in the next sections only the most accurate and the two least complex ones, namely [24], [25] and [27] will be further considered.

III. DYNAMIC DCT APPROXIMATION

In this Section we propose a modified algorithm, which leads to relevant rotation reduction. Since the DCT factorization, described in Section II-A, allows to compute the DCT results by means of the WHT and the following rotation scheme,

TABLE III
COMPARISON OF COMPUTATIONAL COMPLEXITY OF 1D-DCTN.

	Partial Butterfly [31]			MCM Based [13]		WHT-based [14]			RAG-n Based	
N	MULT	ADD	SHIFT	ADD	SHIFT	MULT	ADD	SHIFT	ADD	SHIFT
4	4	8	2	14	10	3	11	0	17	10
8	20	28	2	50	30	15	39	0	69	52
16	84	100	2	186	86	49	115	2	213	176
32	340	372	2	682	278	139	307	8	584	503

TABLE IV
TEST SEQUENCES USED FOR SIMULATIONS.

Class	Resolution	Length	Sequence	Frame Rate
A	2560×1600	5 s	*Traffic*	30 Hz
			People On Street	30 Hz
			Nebuta Festival	60 Hz
			Steam Locomotive	60 Hz
B	1920×1080	10 s	*Kimono*	24 Hz
			Park Scene	24 Hz
			Cactus	50 Hz
			BQ Terrace	60 Hz
			Basketball Drive	50 Hz
E	1280×720	10 s	*Four People*	60 Hz
			Johnny	60 Hz
			Kristen and Sara	60 Hz

the key-idea of this modified algorithm is to explore different approximations of the DCT by adaptively reducing the number of rotations to be computed. This space goes from the WHT, where all the Givens rotations are skipped, to the complete DCT where all the computation is performed. It is worth pointing out that the results obtained by removing rotations are approximated. However, in HEVC the coefficients produced by the DCT are quantized, so the injected quantization noise partially hides the accuracy degradation introduced at the transform step. Therefore, if a small quality loss is considered acceptable, then it is unnecessary to compensate the DCT approximation.

In order to determine whether a rotation has to be applied, a pre-computation mechanism is adopted. The idea is to compare the two inputs of one rotation unit with a threshold. Then, the rotation is skipped when the magnitude of both inputs is lower than the threshold or the special signal SKIP is asserted. The effectiveness of this method strongly depends on a proper choice of the thresholds. In this paper we analyse the general approach, which allows to identify appropriate thresholds. Then, four different trade-offs between computation saving and rate-distortion performance have been determined on the basis of the results of such analysis.

A. Experimental Setup

All the simulations have been performed encoding twelve high-resolution video sequences, taken from the set of sequences employed during the HEVC standardization process and referred to as common test conditions (CTC) [35]. These sequences belong to three classes, which differ in terms of resolution, characteristics of the content and application, as reported in Table IV. According to the CTC [35], each class can be encoded with different configurations: *all intra* (AI), *low delay* (LD) and *random access* (RA). AI configuration encodes the video as a sequence of intra frames. The LD configuration is used for interactive applications such as video-conferencing. It is worth noting that it uses only B frames with reference to previous pictures in order to avoid delay due to the encoding computation. The RA configuration is related to entertainment applications and it allows to start decoding from different points in the sequence. This feature is achieved by using a hierarchical Group Of Pictures (GOP) structure made of both I-frames and B-frames.

All the simulations shown in this paper have been performed with the HEVC reference software HM 8.0 [31], which has been modified with the introduction of our DCT and other approximations taken from the literature [32]. Four quantization parameters (QP) were fixed, namely 22, 27, 32 and 37 as suggested in [35]. Finally, rate-distortion curves, which use the combined peak-signal-to-noise-ratio PSNR_{YUV}, as defined in [2], were used as quality measure. The Bjøntegaard method [36] for calculating objective differences (ΔPSNR and ΔRate) between rate-distortion curves has been used as the metric for evaluating quality loss.

B. DCT and Rotation Statistics

From now on, the 2D-DCTN will be referred to as DCTN for brevity. In order to limit quality loss, a statistical analysis of which DCTN are used is required. This information is crucial to understand which DCTs rotation mainly contributes to the quality degradation, as well as to calculate the average throughput of the proposed architectures. As an example Table V reports the usage statistics of each DCT and the corresponding percentage of rotations for three sequences, taken from different classes, encoded with the configurations specified in the CTC [35]. As it can be observed, simulations results with LD or RA configurations point out that all the sequences exhibit similar percentage of usage: the most used DCT is the DCT4 with almost the 70% of the total count, then the DCT8 with about the 20% and the DCT16 with the 5%. The least used one is the DCT32 with a percentage below the 1%. The values are slightly different when the AI configuration is employed; in this case the count for DCT4 decreases to about 58% while larger transforms increase, especially the DCT8 passing from about 20% to about the 34%. The mismatch between statistics of AI and LD, RA configurations is due to the different performance between intra- and inter-prediction, which can remove some TU partitioning from the exhaustive search set.

On the other hand, the highest number of rotations belongs to the DCTs of size 16 and 32, for which it grows more than linearly, as indicated in (12). Together, they cover approximately the 70% of the total, the remaining 30% is due to

TABLE V
DCT AND ROTATION STATISTICS ON SEQUENCES:
(A) TRAFFIC. (B) KIMONO. (C) FOURPEOPLE.

	DCT Statistics			Rotation Statistics		
	AI	LD	RA	AI	LD	RA
DCT4	59.2%	-	72.3%	4.5%	-	7.5%
DCT8	34.0%	-	21.6%	25.9%	-	22.4%
DCT16	5.4%	-	5.3%	27.9%	-	37.5%
DCT32	1.4%	-	0.8%	41.7%	-	32.6%

(A)

	DCT Statistics			Rotation Statistics		
	AI	LD	RA	AI	LD	RA
DCT4	56.8%	72.0%	72.0%	3.9%	7.5%	7.4%
DCT8	35.2%	21.9%	21.8%	24.5%	22.7%	22.5%
DCT16	6.5%	5.3%	5.4%	30.7%	37.3%	37.8%
DCT32	1.5%	0.8%	0.8%	40.9%	32.5%	32.3%

(B)

	DCT Statistics			Rotation Statistics		
	AI	LD	RA	AI	LD	RA
DCT4	60.8%	73.4%	-	4.8%	8.2%	-
DCT8	32.8%	20.9%	-	25.7%	23.3%	-
DCT16	5.0%	5.0%	-	26.6%	37.9%	-
DCT32	1.4%	0.7%	-	43.0%	30.6%	-

(C)

TABLE VI
NUMBER OF ROTATIONS PER ANGLE BREAKDOWN.

	$\pi/8$	$p \cdot \pi/16$	$p \cdot \pi/32$	$p \cdot \pi/64$
DCT4	8	-	-	-
DCT8	48	16	-	-
DCT16	224	96	32	-
DCT32	960	448	192	64

the DCT4 (about 7%) and the DCT8 (about 23%). Therefore, since large-size DCTs require higher computational effort than small-size ones, it is likely that the most of saved operations and quality loss, are due to DCT16 and DCT32. However, some rotations are used across more than one DCT, so the effect on the PSNR of one Givens rotation depends on both the DCT size and the rotation angle. Thus, Table VI shows the number of rotations per angle to compute each DCTN, where p is an odd integer lower than $N/2$.

Results in Table V and VI highlight that the contribution of each rotation to both computational complexity and quality loss depends on the angle and DCT size. Therefore, different thresholds can be assigned to the same rotation module depending on the working conditions.

C. Operating Mode Definition

The proposed method relies on the values assigned to the thresholds. Since the proposed transform module supports four DCT sizes, with many rotation angles, the threshold set ($\mathcal{T}$) contains up to 26 elements, each of which is associated to a rotation of an angle in a DCT. The general optimization problem, which allows to determine the optimal set of thresholds can be written as:

$$\mathcal{T}^* = \begin{cases} \underset{\mathcal{T}}{\text{maximize}} & \Phi(\mathcal{T}) \\ \text{subject to} & \Lambda(\mathcal{T}) < L \end{cases}, \tag{14}$$

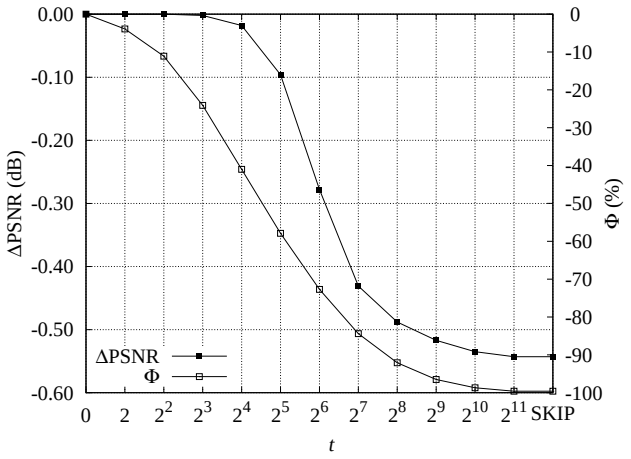


Fig. 1. ΔPSNR and $\Phi(t)$ curves over the threshold value, averaged on all the sequences encoded with AI.

where $\mathcal{T}^*$ is the optimal set of thresholds, $\Phi(\mathcal{T})$ and $\Lambda(\mathcal{T})$ are functions used to model the HEVC encoding process and representing the computational saving and the quality loss, respectively. The maximum allowable quality loss L is measured using the Bjøntegaard difference on the PSNR_{YUV}. In addition, we constrain the threshold values to be powers of 2, in order to simplify the hardware implementation. Since the exhaustive search of a solution is computationally too complex, we reduced the design space by using one threshold t for all the rotation angles, *i.e.* $\mathcal{T} = t$.

Fig. 1 shows the ΔPSNR difference and the percentage of saved rotations (Φ), averaged on all the test sequences encoded with the AI configuration, for threshold values t in the range from 0 to 2048 and in the case of SKIP signal assertion. As expected, both the quality loss and the rotation saving increase with the threshold value up to the limit fixed by asserting the SKIP signal, where all the rotations are skipped and the DCT is approximated by the WHT only. As it can be observed, by choosing values smaller than 8 or larger than 128, the quality loss and the rotation saving are close to either the full DCT or the WHT respectively. Therefore, only the results for thresholds within this range are reported in the left part of Table VII, which shows the quality loss and the rotation reduction, averaged on video sequences reported in Table IV taken from classes (A, B and E) separately or together (All) and encoded with the experimental setup described in Section III-A. Noticeably, the approximations with low threshold values exhibit negligible performance loss with respect to the complete DCT. Nevertheless, on average, they reduce the computational effort of the lifting scheme by more than 40%. Then, the quality loss and the rotation saving grow up to $t = 64$. From that point, a further increase of the threshold leads to large quality degradation without a significant reduction in terms of complexity. This behaviour is observed for all the encoding configurations. Focusing on the quality loss, the AI configuration, which employs only I-frames, is more sensitive to DCT approximation than the LD and RA configurations. This effect is due to the fact that inter-prediction provides better performance than intra-prediction,

TABLE VII
QUALITY-COMPLEXITY TRADE-OFF ANALYSIS: ΔPSNR [DB], ΔRATE [%] LOSS AND COMPUTATIONAL SAVING [%].
(A) ALL INTRA. (B) LOW DELAY. (C) RANDOM ACCESS.

t	8			16			32			64			128			SKIP		
Class	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$
A	-0.002	0.0%	22%	-0.018	0.4%	38%	-0.105	2.5%	55%	-0.389	8.9%	69%	-0.643	15.2%	82%	-0.779	18.8%	100%
B	-0.002	0.0%	20%	-0.020	0.5%	37%	-0.090	2.5%	55%	-0.213	6.3%	71%	-0.311	9.5%	84%	-0.420	13.0%	100%
E	-0.001	0.0%	33%	-0.017	0.4%	52%	-0.093	2.1%	67%	-0.238	5.5%	80%	-0.350	8.1%	89%	-0.439	10.2%	100%
All	-0.002	0.0%	24%	-0.018	0.4%	41%	-0.096	2.4%	58%	-0.278	7.0%	73%	-0.431	11.0%	84%	-0.545	14.3%	100%

(A)

t	8			16			32			64			128			SKIP		
Class	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$
A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B	-0.002	0.1%	22%	-0.007	0.3%	41%	-0.034	1.7%	61%	-0.087	4.1%	78%	-0.133	6.2%	89%	-0.203	9.4%	100%
E	-0.003	0.0%	39%	-0.005	0.1%	59%	-0.040	1.4%	77%	-0.099	3.6%	89%	-0.137	5.1%	95%	-0.186	6.9%	100%
All	-0.002	0.1%	28%	-0.006	0.3%	48%	-0.037	1.6%	67%	-0.091	3.9%	82%	-0.135	5.8%	92%	-0.197	8.5%	100%

(B)

t	8			16			32			64			128			SKIP		
Class	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$	ΔPSNR	ΔRate	$\Phi(t)$
A	-0.001	0.0%	25%	-0.007	0.3%	43%	-0.043	1.9%	61%	-0.163	7.7%	76%	-0.316	16.3%	87%	-0.396	20.6%	100%
B	-0.001	0.1%	23%	-0.008	0.4%	42%	-0.040	2.0%	62%	-0.108	5.1%	78%	-0.167	8.0%	90%	-0.242	11.6%	100%
E	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
All	-0.001	0.1%	24%	-0.008	0.3%	43%	-0.042	1.9%	62%	-0.132	6.3%	77%	-0.233	11.7%	89%	-0.311	15.6%	100%

(C)

by producing smaller residuals and DCT coefficients, which are likely to be quantized near zero. Therefore, the approximation of larger residuals generated by the intra-prediction leads to larger quality loss.

Stemming from the results reported in Fig. 1 and Table VII, we have selected two of the proposed trade-offs as operating modes for our DCT modules, in addiction to the full DCT and the WHT. Four operating modes have been defined as:

- MODE0 is the complete DCT. It computes all the rotations ($t = 0$) and no power saving is achieved.
- MODE1 is the first approximation. The maximum PSNR loss is fixed to 0.02 dB. It leads to a minimum reduction in the rotation computation of 37% with $t = 16$.
- MODE2 is the second approximation. The maximum PSNR loss is 0.1 dB, corresponding to a minimum rotation reduction of about 55% with $t = 32$.
- MODE3 is the WHT computation only. It leads to the maximum quality loss (less than 0.8 dB), but also to the maximum saving (100%) using the SKIP signal.

It is worth noting that the operating modes can be redefined by the designers depending on the application by selecting different trade-offs among the ones reported in Table VII. Moreover, the proposed method can be also used in conjunction with other techniques, such as zero block detection algorithms [37], to further reduce the complexity of the encoder despite of small quality degradation.

TABLE VIII
ΔPSNR [DB] AND ΔRATE [%] OF DCT APPROXIMATIONS IN HEVC.
(A) ALL INTRA. (B) LOW DELAY. (C) RANDOM ACCESS.

Method	AI (default)		AI (max TU 8×8)	
	ΔPSNR	ΔRate	ΔPSNR	ΔRate
CB-2011 [24]	-0.048	1.1%	-0.161	3.7%
Modified CB-2011 [25]	-0.099	2.2%	-0.377	9.1%
Improved Modified CB-2011 [27]	-0.106	2.4%	-0.406	9.9%
MODE1	-0.018	0.4%	-0.022	0.5%
MODE2	-0.096	2.4%	-0.132	3.0%
MODE3	-0.545	14.3%	-0.306	7.3%

(A)

Method	LD (default)		LD (max TU 8×8)	
	ΔPSNR	ΔRate	ΔPSNR	ΔRate
CB-2011 [24]	-0.016	0.6%	-0.041	1.8%
Modified CB-2011 [25]	-0.040	1.6%	-0.167	7.2%
Improved Modified CB-2011 [27]	-0.044	1.8%	-0.190	8.1%
MODE1	-0.006	0.3%	-0.008	0.3%
MODE2	-0.037	1.6%	-0.045	2.1%
MODE3	-0.197	8.5%	-0.110	5.1%

(B)

Method	RA (default)		RA (max TU 8×8)	
	ΔPSNR	ΔRate	ΔPSNR	ΔRate
CB-2011 [24]	-0.019	0.7%	-0.081	3.9%
Modified CB-2011 [25]	-0.046	1.8%	-0.249	11.7%
Improved Modified CB-2011 [27]	-0.049	1.9%	-0.276	12.9%
MODE1	-0.008	0.3%	-0.008	0.4%
MODE2	-0.042	1.9%	-0.069	3.7%
MODE3	-0.311	15.6%	-0.194	9.6%

(C)

D. DCT Approximation Comparison in HEVC

To compare our DCT approximations with other ones proposed in literature, we implemented in the HEVC reference software [31] the CB-2011 [24], the Modified CB-2011 [25] and the Improved Modified CB-2011 [27] 8×8 DCT algorithms, which are respectively the most accurate and the two least complex among the previous approximations presented in Table I[1]. Table VIII reports the average ΔPSNR and ΔRate with reference to the complete DCT, calculated on the rate-distortion curves of all the video sequences encoded with

[1]For the source code of the modified reference software see [32].

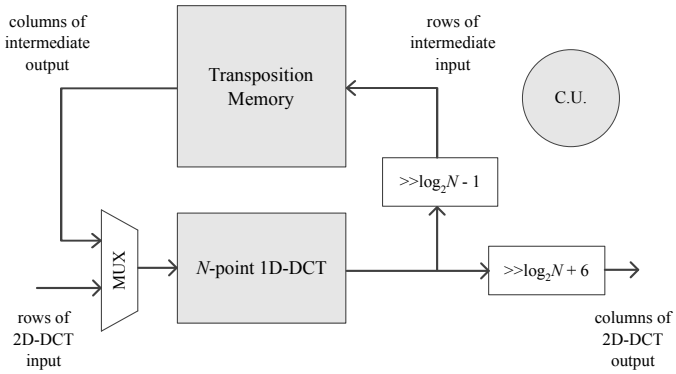


Fig. 2. Folded 2D-DCT architecture.

two configurations. The first one is the default configuration, while the second one is derived from the default configuration by limiting the maximum transform size to 8×8. As shown in Table VIII, the rate-distortion performance with default configurations of our MODE1 approximation is the best one, whereas MODE3 is the worst one. Indeed, in this configuration the results achieved with [24], [25] and [27] methods are affected by DCT8 approximations only. In order to make a fair comparison, we analyzed the custom configuration with maximum TU size limited to 8×8. As it can be observed, the behaviour of our proposed DCT is independent of the transform size, due to its inherent adaptivity. On the other hand, the methods taken from the literature show some degradation of rate-distortion performance when the percentage of usage of the 8×8 DCT grows. In particular, the reduction of complexity provided in [25], [27] and shown in Table I, leads to significant quality loss, much larger than the bound fixed by the proposed MODE3.

IV. Proposed Architectures

In this Section we show the top-level of the proposed architectures and two possible implementations of the 1D-DCT module. The first one is a completely unfolded architecture derived from the one proposed in [14], where all the required operations are mapped to different resources, thus achieving the highest possible throughput for the DCT factorization presented in Section II. The second architecture is a folded one, where the folding technique [28] is exploited to reuse hardware resources during the computation of the different DCTs.

A. 2D-DCT Architecture

Fig. 2 reports the proposed 2D-DCT architecture. It is composed of two main blocks: the 1D-DCT module and the transposition memory, which has the role of transposing the intermediate results. Due to its flexibility, this architecture is able to concurrently perform the DCT computation on multiple blocks, depending on the DCT size. Since the number of input samples is fixed to 32, this module can compute 32/N DCTN, *i.e.* one DCT32, two DCT16, four DCT8 or eight DCT4, thus leading to an efficient usage of the hardware resources. For this reason, in this work the transposition memory is designed in the same way as presented in [13]. It is made of an array of 32×32 registers, required to support the DCT32, and it is able to transpose blocks of different size.

The whole system computation is scheduled as follows. The rows of the input block pixels are fed into the one-dimensional DCT module, which computes the 1D-DCTN. Input pixels are represented with 9 bits, this because they are the result of the difference between the current and the predicted frames. Then, according to the HM 8.0, the produced results are scaled to 16 bits and stored row-wise in the transposition memory, as shown by the $\log_2 N - 1$ right shift block in the feedback path in Fig. 2. Once all the rows have been processed, the multiplexers feed the 1D-DCT with the columns of the intermediate data, which are stored in the transposition memory. Finally, the results are scaled back to 16 bits for compliance with the HM 8.0 (see the $\log_2 N + 6$ shift block in Fig. 2). All the operations are managed by a control unit (C.U.), which generates both the signals for the data selection and storing and for the 1D-DCT block.

B. Unfolded 1D-DCT

The first proposed architecture is the unfolded one. Its data flow is reported in Fig. 3. It is a four stage pipelined datapath composed of two main computational entities: the HT (left side of Fig. 3) and the rotation scheme (right part of Fig. 3). The former block receives 32 samples at each clock cycle and computes the HT by means of the butterfly stages (BUT), which perform the B butterflies indicated in (11). The HT block is followed by a network, which implements i) bit reverse and Gray coding, to obtain Walsh-ordered data, and ii) bit reverse and permutation to reorder the signals for the rotation block.

According to (12), the rotation scheme contains R = 49 modules, depicted as circles and ovals in Fig. 3, to support the worst-case, which is the DCT32. Each rotation receives as input the threshold for the related angle and it is equipped with some logic to implement the pre-computation mechanism introduced in Section III. Fig. 4 reports the block scheme of a generic rotation module. Two comparators are used to determine whether the incoming signals are smaller than the threshold and the enable signal activates only the register bank which is used in the following clock cycle. If one rotation is not calculated, then the data are sampled by register bank 1 and they are not propagated to the rotation logic, thus saving dynamic power. Otherwise, the path which passes through register bank 2 is enabled. A final multiplexer, driven by the same condition signal, chooses the correct output. Moreover, the special SKIP signal (which has been introduced in Section III) is also used to avoid rotations when the DCT size is smaller then 32 or to bypass all the rotations when the module works in MODE3 (WHT only). According to (10), the rotation is computed by means of three lifting steps, each composed of a multiplication, a shift and an addition, as illustrated in Fig. 5, where x_1, x_2 are the input values and y_1, y_2 the output results. The values of the multiplier coefficients (a, b) are the ones reported in Table II. The output of the rotation block is then propagated to a network, which integrates permutation

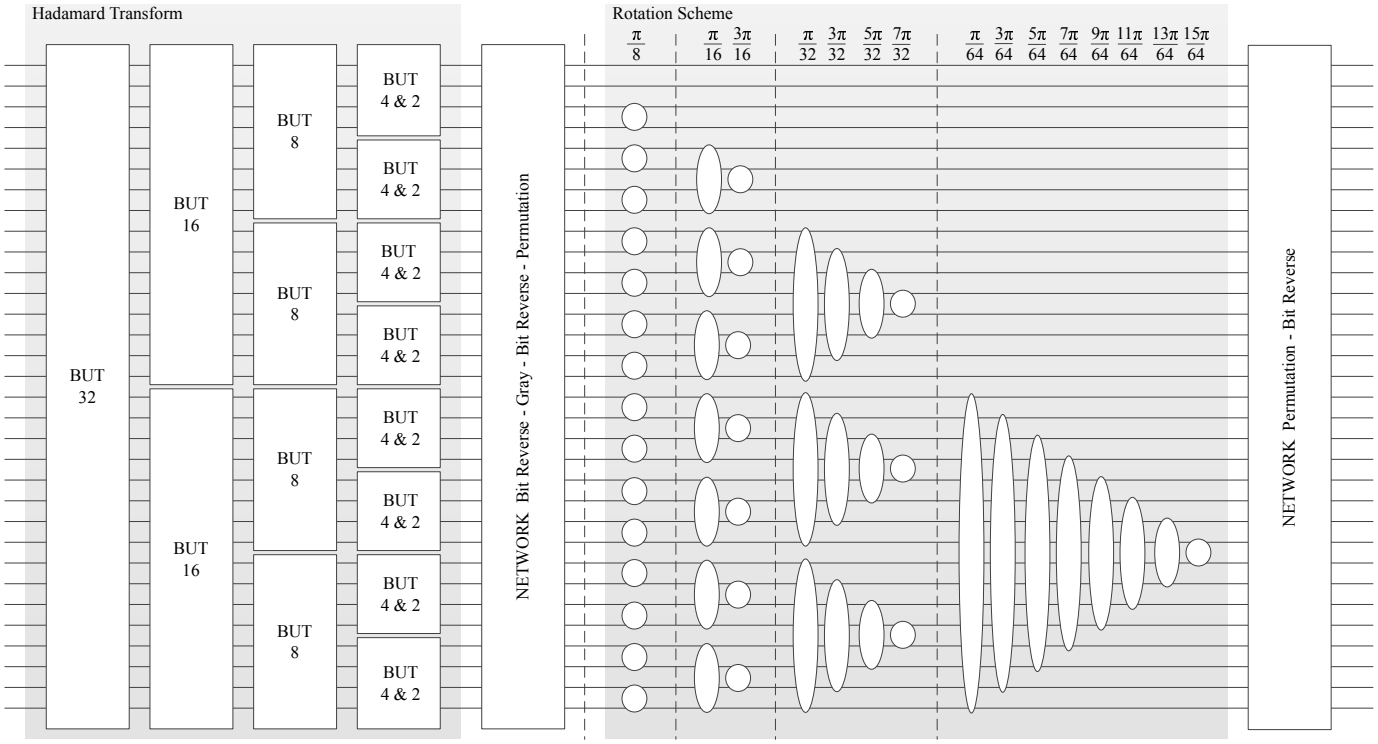


Fig. 3. Unfolded architecture for 1D-DCT.

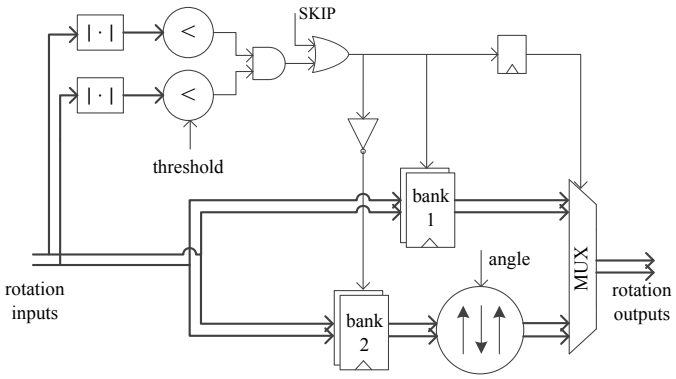


Fig. 4. Block scheme of a rotation module.

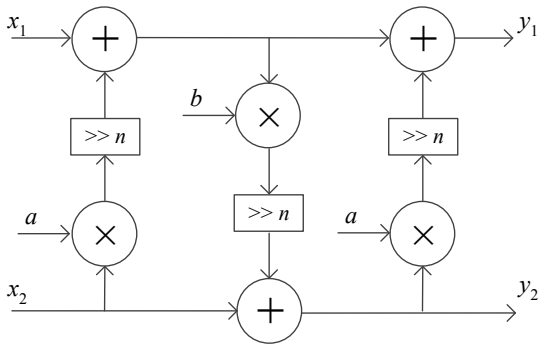


Fig. 5. Lifting scheme circuit.

and bit reverse reordering. The flexibility of the architecture is given by a custom set of connections, which arranges the paths between operators as required for the DCTN.

Since different DCT types require a variable number of rotation stages, the throughput of this architecture, defined as the number of produced results over the time required to produce them, varies with the DCT size. The throughput of the complete 2D-DCT architecture employing the 1D-DCT is

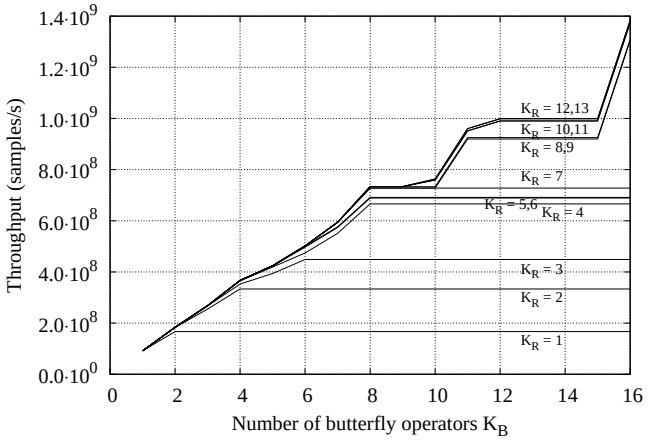
$$T = \frac{32}{N} \cdot \frac{N^2}{\Delta}, \tag{15}$$

where $32/N$ represents the number of $N \times N$ blocks processed concurrently, $\Delta = 2 \cdot (P + N)/f_{CK}$ is the time required to compute the results, $P = \log_2 N - 1$ is the number of pipeline stages required for the computation of a DCTN and f_{CK} is the clock frequency.

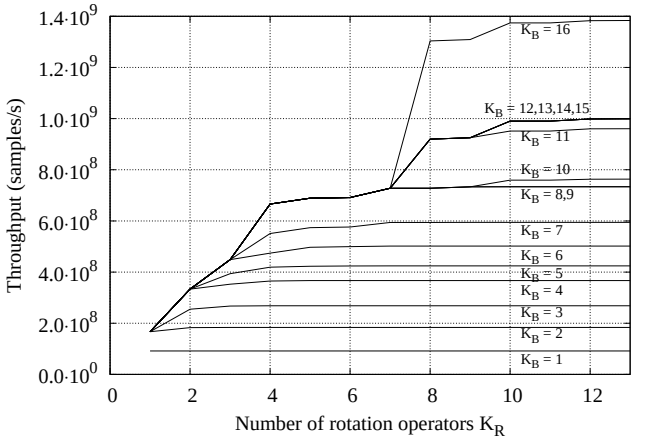
C. Folded 1D-DCT

The second implementation is the folded 1D-DCT, where a set of resources is shared among DCTs of different size. The reuse of such operators can be exploited either to support DCT computations of different lengths or to increase the throughput of small size DCTs. Therefore, the amount of resources and the folding degree define a large design space. The exploration of such a space is detailed in the following paragraphs. The technique is applied to both HT and rotations. Let K_B, K_R be the number of butterfly and rotation resources respectively, and α, β indicate the number of cycles required to compute the WHT and the Givens rotations of $32/N$ 1D-DCT of size N.

Assuming that the architecture works in pipeline and that the number of clock cycles required to compute one 1D-DCTN



(A)



(B)

Fig. 6. Architectural space analysis: (a) throughput as function of K_B (b) throughput as function of K_R.

is M_N, then the time to compute N^2 results is

$$\Delta = \frac{2 \cdot N \cdot M_N + L}{f_{CK}}, \tag{16}$$

where $L = 2 \cdot M_N$ is the latency of the architecture. Since the architecture is folded $M_N = \max\{\alpha, \beta\}$, where α and β depend on the number of available resources, K_B and K_R. Assuming perfect scheduling, α and β can be obtained for each DCT size as $\alpha = B/K_B$ and $\beta = R/K_R$. In order to satisfy the data dependency between the computational stages (see Fig. 3), we assume that each gray shaded block (HT and Givens rotations) is computed in a minimum number of cycles, namely $\log_2 N$ for the HT and $\log_2 N - 1$ for the rotations. Thanks to the concurrent execution of different DCTs, a feasible perfect scheduling of the resources can always be identified. From a detailed analysis of the proposed folded architecture, we discovered that data dependencies occurs with N = 32 in the HT computation only when $8 < K_B < 16$. Moreover, the rotation block requires two cycles: the first one to evaluate the enable condition and the second one to compute the rotation, if needed. The plots of the throughput as function of K_B and K_R are depicted in Fig. 6. In this example the throughput is calculated as in (15) and (16) with a reference clock frequency f_{CK} equal to 250 MHz. As expected, increasing K_B and K_R, the throughput grows up to a maximum value, which depends only on the data dependencies and no longer on the available resources. As it can be observed, there is a relevant increase of throughput when K_B reaches 8, 11 and 16 and when K_R becomes equal to 4 and 8, which correspond to a more efficient usage of the resources. Thus, Fig. 6 is also intended for design purposes. Indeed, depending on the application, the designer can set the throughput and find the minimum number of resources required to achieve it. This work targets a throughput of 1.2 Gsamples/s (7680 $\times$ 4320 $\times$ 24 $\times$ 1.5) *i.e.* the one required for the encoding of 8K ultra-high definition (UHD) video sequences at 24 fps with 4:2:0 YUV sub-sampling, which is one of the HEVC applications. Therefore, the solution with K_B = 16 and K_R = 8 is selected. Such a solution requires α = 2, 3, 4, 5 and β = 2, 6, 10, 14 clock cycles to compute the 1D-DCT of size 4, 8, 16 and 32 respectively. The proposed architecture for the 1D-DCT is depicted in Fig. 7. It is composed of two main blocks: the HT computation (left part of Fig. 7) and the rotation scheme (right side of Fig. 7). Resorting to the folding technique [28], each module shows two selection blocks, used to implement the time-multiplexing of the resources, and a bank of temporary registers to store intermediate results. A pipeline stage separates the HT computation from the rotation scheme, thus allowing concurrent computation of the two parts on successive samples. Two networks for data reordering complete the folded implementation. The first network implements the Walsh ordering and arranges the data for the Givens rotations. The second one applies the permutation and bit-reverse ordering to the results.

TABLE IX
COMPARISON OF DIFFERENT 1D-DCT ARCHITECTURES.

Design	Technology	Gates	f_{CK} (MHz)	T (Gsps)
Shen *et al.* [8]	0.13-μm	134 K	350	1.400
Park *et al.* [9]	0.18-μm	52 K	300	0.638
Meher *et al.* [13]	90-nm	131 K	187	2.992
Unfolded 1D-DCT	90-nm	163 K	250	3.212
Folded 1D-DCT	90-nm	74 K	250	1.302

V. IMPLEMENTATION RESULTS

A. Synthesis Results for 1D-DCT

The proposed architectures have been coded in VHDL, and synthesized with Synopsys Design Compiler using a 90-nm standard cell library for an operating clock frequency equal to 250 MHz. In Table IX the *Unfolded 1D-DCT* and the *Folded 1D-DCT* are compared in terms of gate count, frequency (f_{CK}) and throughput (T) with other existing 1D-DCT architectures. It is important to note that the throughput is calculated as the average on the different DCT sizes weighted with the statistics reported in Section III-B, and it is determined considering the 2D folded structure in Fig. 2.

The proposed *Unfolded 1D-DCT* architecture shows the highest throughput at the expense of larger gate count with respect to the other designs. In particular, the Hadamard

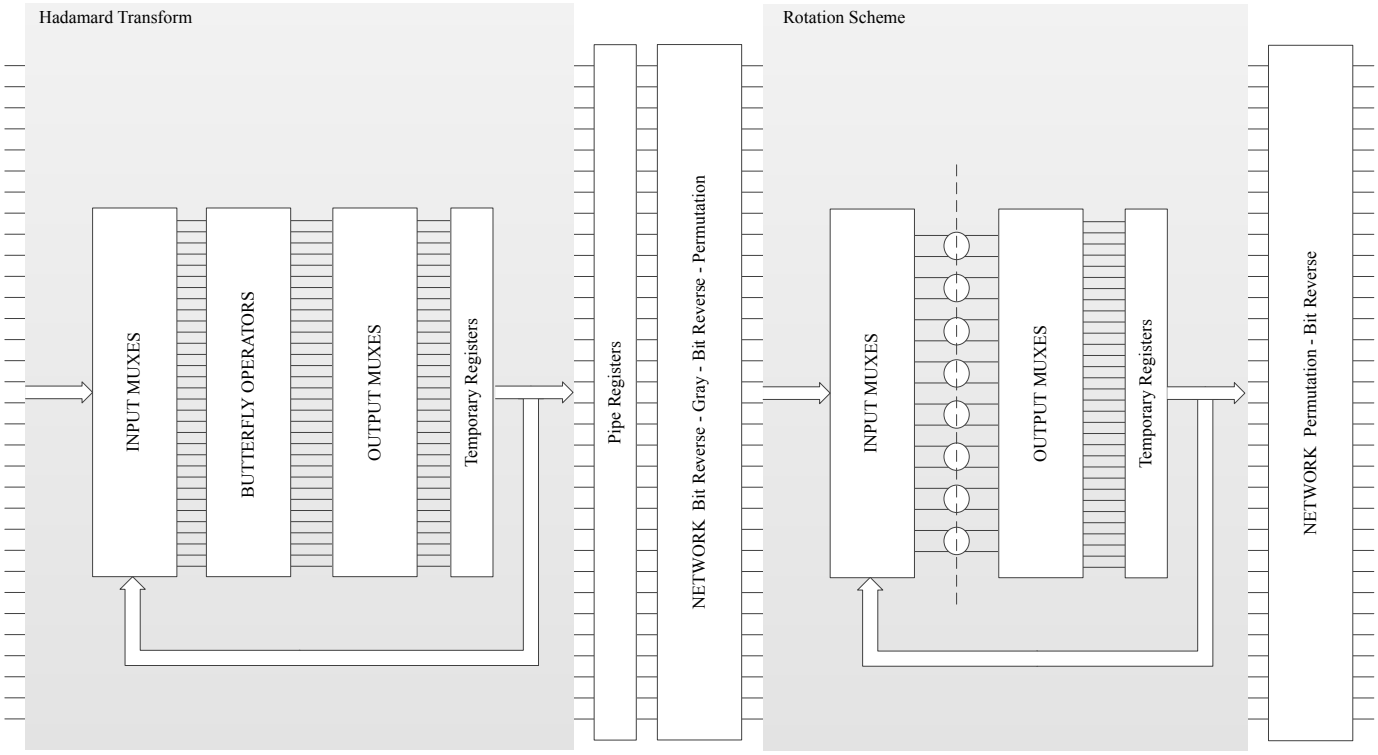


Fig. 7. Proposed folded architecture for 1D-DCT.

TABLE X
COMPARISON OF DIFFERENT 2D-DCT ARCHITECTURES.

Design		N	Technology	Gates	f_{CK} (MHz)	T (Gsps)	P (mW)	EPS (pJ)	P_d (mW/MHz)
BAS-2008 [27]		8	45-nm	78 K	809	6.472	550	84.98	0.679
BAS-2011 a=0 [27]		8	45-nm	74 K	763	6.104	500	81.91	0.655
BAS-2011 a=1 [27]		8	45-nm	68 K	833	6.664	480	72.02	0.576
BAS-2011 a=2 [27]		8	45-nm	68 K	832	6.656	480	72.11	0.576
CB-2011 [27]		8	45-nm	86 K	806	6.448	600	93.05	0.744
Modified CB-2011 [27]		8	45-nm	74 K	849	6.792	550	80.97	0.647
Improved Modified CB-2011 [27]		8	45-nm	74 K	851	6.808	550	80.78	0.646
Meher *et al.* Folded [13]		4, 8, 16, 32	90-nm	208 K	187	2.992	40.04	13.38	0.214
Meher *et al.* Full-parallel [13]		4, 8, 16, 32	90-nm	347 K	187	5.984	67.57	11.29	0.361
Ahmed *et al.* [14]		4, 8, 16, 32	90-nm	149 K	150	0.253	-	-	-
Architecture 1	MODE0	4, 8, 16, 32	90-nm	243 K	250	3.212	51.72	16.10	0.207
	MODE1						35.38	11.01	0.142
	MODE2						30.88	9.61	0.124
	MODE3						22.71	7.07	0.091
Architecture 2	MODE0	4, 8, 16, 32	90-nm	157 K	250	1.302	28.98	22.26	0.116
	MODE1						20.75	15.94	0.083
	MODE2						18.40	14.13	0.074
	MODE3						13.82	10.61	0.055

Transform and the Rotation Scheme block in Fig. 3 requires 22 K and 135 K gates respectively. It is worth noting that the proposed architecture features some hardware overhead compared with the solution provided in [13]. This figure depends on the fact that [13] supports only exact DCT computation, whereas the proposed one includes some logic and registers to support the four operating modes defined in Section III. On the other hand, the proposed *Folded 1D-DCT* shows a very reduced gate count for the computation of the 1D-DCT, even if it supports the four operating modes as well as the unfolded one. Only 51 K gates are needed to implement the Rotation Scheme, while 15 K gates are used for the Hadamard Transform. When compared with [9], where only transforms of size 16 and 32 are implemented, the proposed *Folded 1D-DCT* architecture shows similar gate count, but it can achieve double throughput.

B. Synthesis Results for 2D-DCT

The two proposed 2D-DCT architectures, based on the unfolded and the folded 1D-DCT modules, have been synthesized as well. In the following they will be referred to as *Architecture 1* and *Architecture 2* respectively. Table X lists the technology, gate count, operating frequency (f_{CK}), throughput (T), power consumption (P), energy-per-sample (EPS) and frequency-normalized dynamic power (P_d), which characterize

the two designs and other existing 2D-DCT architectures for HEVC. As it can be observed, all the implementations reported in [27], address the design of DCT of size 8×8 only. They provide very high throughput at the cost of very high power consumption. Besides, the operating frequency of 250 MHz allows the *Architecture 1* to support 8K UHD applications up to 64 fps with 4:2:0 YUV sub-sampling. For such applications, the Full-parallel architecture proposed in [13] achieves the highest throughput with a large gate count as it relies on two 1D-DCT modules. On the contrary, the Folded architecture described in [13] contains one 1D-DCT module, as the *Architecture 1* we propose in this current work. As it can be observed the proposed *Architecture 1* achieves higher throughput with respect to the folded implementation proposed in [13], but it shows slightly larger gate count and power consumption, when the operating mode is set to MODE0, because additional logic is required to support the proposed power reduction algorithm.

On the other hand, *Architecture 2* provides the smallest absolute power consumption, equal to 28.98 mW showing nearly the same gate count as [14] but achieving about five times larger throughput. Indeed, the proposed folded architecture supports 8K UHD applications with a maximum frame-rate equal to 26 fps. Moreover, the folded architecture can be properly sized by choosing the target throughput with the methodology proposed in Section IV-C, thus optimizing area occupation and power consumption.

Finally, it is worth noting that both the proposed architectures outperform the other ones in terms of frequency-normalized dynamic power (P_d) while they provide slightly higher EPS than the implementations in [13]. However, both can be reduced by operating in one of the low-power modes defined in Section III-C, which allow to reduce the power consumption as shown in the following section.

C. Power Consumption Reduction

The power consumption of the proposed architecture can be further reduced by using the proposed operating modes. In order to compute the power consumption reduction achieved by each mode, power estimation has been performed simulating and annotating the switching activities of each node of the gate-level netlist generated by the synthesis tool. A specific testbench has been used to apply values of real samples to the input ports of the designed modules. These samples have been extracted by annotating the DCT inputs during encoding simulation of sequences taken from Table IV and they comply with the DCT usage statistics. Simulations have been performed for each of the four operating modes defined in Section III.

The last two rows of Table X show the power consumption, energy-per-sample and frequency-normalized dynamic power of the two proposed architectures when operating in different modes, while Table XI summarizes the power saving calculated with reference to the MODE0 (complete DCT). As expected, the power saving in *Architecture 1* increases when reducing the computation, namely passing from MODE1 to MODE3 (WHT only), where a power reduction of about 56% can be achieved. The same trend as *Architecture 1* is observed for *Architecture 2*, even though the saving is slightly lower. This is due to the folding technique, which exploits a more effective usage of the resources than the unfolded architecture, and to the instantiation of real multipliers in the lifting scheme, instead of custom add-shift multipliers.

TABLE XI
POWER SAVING OF OPERATING MODES.

Design	MODE0	MODE1	MODE2	MODE3
Architecture 1	ref.	-31.6%	-40.3%	-56.1%
Architecture 2	ref.	-28.4%	-36.5%	-52.3%

VI. CONCLUSION

In this paper two novel DCT architectures for the HEVC standard have been proposed. The proposed 2D-DCTN computation is based on a 1D-DCTN core, where the complete DCT matrix is factorized as the cascade of the WHT and Givens rotations. In order to reduce the number of operations the algorithm has been modified by introducing a pre-computation mechanism, which allows to save rotations and dynamic power at the expense of very small PSNR loss. Then, two flexible and HEVC compliant architectures, able to support the DCT of size N equal to 4, 8, 16, 32 have been proposed. The first one implements the 1D-DCTN in a completely unfolded fashion, while the second one has been selected by identifying the proper folding degree. Moreover, the proposed architectural space exploration provides a method to design such systems by relying on the throughput required by the application. From the implementation results, it is found that the architectures employing the unfolded and folded 1D-DCT module respectively show competitive throughput and gate count with respect to previous existing architectures. Finally, power consumption results show the advantages offered by the proposed operating modes, namely MODE1, MODE2 and MODE3. In particular, MODE1 reduces roughly by 30% the power consumption with negligible quality loss. According to the complexity analysis provided in [3], power saving up to 10% can be achieved for the entire HEVC encoder and decoder by operating with the proposed approximations.

ACKNOWLEDGMENT

The authors would like to thank the HPC@POLITO, a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino (http://www.hpc.polito.it), which have provided the computational resources.

REFERENCES

[1] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[2] J. Ohm, G. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards–Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec 2012.

[3] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC Complexity and Implementation Analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec 2012.

[4] M. Shafique and J. Henkel, "Low power design of the next-generation High Efficiency Video Coding," in *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, Jan 2014, pp. 274–281.

[5] 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.

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

[7] A. Madanayake, R. Cintra, V. Dimitrov, F. Bayer, K. Wahid, S. Kulasekera, A. Edirisuriya, U. Potluri, S. Madishetty, and N. Rajapaksha, "Low-Power VLSI Architectures for DCT/DWT: Precision vs Approximation for HD Video, Biomedical, and Smart Antenna Applications," *IEEE Circuits Syst. Mag.*, vol. 15, no. 1, pp. 25–47, Firstquarter 2015.

[8] S. Shen, W. Shen, Y. Fan, and X. Zeng, "A Unified 4/8/16/32-Point Integer IDCT Architecture for Multiple Video Coding Standards," in *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, July 2012, pp. 788–793.

[9] J. Park, N. W.J., S. Han, and L. S., "2-D Large Inverse Transform (16x16, 32x32) for HEVC (High Efficiency Video Coding)," *Journal of Semiconductor Technology and Science*, vol. 12, no. 2, pp. 203–211, Jun 2012.

[10] 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.

[11] J. Zhu, Z. Liu, and D. Wang, "Fully pipelined DCT/IDCT/Hadamard unified transform architecture for HEVC Codec," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, May 2013, pp. 677–680.

[12] M. Budagavi and V. Sze, "Unified forward+inverse transform architecture for HEVC," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, Sept 2012, pp. 209–212.

[13] P. Meher, S. Y. Park, B. Mohanty, K. S. Lim, and C. Yeo, "Efficient Integer DCT Architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 168–178, Jan 2014.

[14] A. Ahmed, M. U. Shahid, and A. Rehman, "N Point DCT VLSI Architecture for Emerging HEVC Standard," *VLSI Design*, vol. 2012, 2012.

[15] Y.-J. Chen, S. Oraintara, and T. Nguyen, "Video compression using integer DCT," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 2, Sept 2000, pp. 844–845 vol.2.

[16] S. Venkataraman, V. Kanchan, K. Rao, and M. Mohanty, "Discrete transforms via the Walsh-Hadamard transform," *Signal Processing*, vol. 14, no. 4, pp. 371–382, 1988.

[17] V. Dimitrov, K. Wahid, and G. Jullien, "Multiplication-free 8×8 2D DCT architecture using algebraic integer encoding," *Electronics Letters*, vol. 40, no. 20, pp. 1310–1311, Sept 2004.

[18] A. Madanayake, R. Cintra, D. Onen, V. Dimitrov, N. Rajapaksha, L. Bruton, and A. Edirisuriya, "A Row-Parallel 8×8 2-D DCT Architecture Using Algebraic Integer-Based Exact Computation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 6, pp. 915–929, June 2012.

[19] N. Rajapaksha, A. Edirisuriya, A. Madanayake, R. J. Cintra, D. Onen, I. Amer, and V. S. Dimitrov, "Asynchronous Realization of Algebraic Integer-Based 2D DCT Using Achronix Speedster SPD60 FPGA," *Journal of Electrical and Computer Engineering*, vol. 2013, 2013.

[20] Y. Arai, T. Agui, and N. M., "A fast DCT-SQ scheme for images," *Trans. IEICE*, vol. E-71, no. 11, pp. 1095–1097, Nov 1988.

[21] S. Bouguezel, M. Ahmad, and M. Swamy, "Low-complexity 8×8 transform for image compression," *Electronics Letters*, vol. 44, no. 21, pp. 1249–1250, October 2008.

[22] ——, "A low-complexity parametric transform for image compression," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, May 2011, pp. 2145–2148.

[23] ——, "A novel transform for image compression," in *Circuits and Systems (MWSCAS), 2010 53rd IEEE International Midwest Symposium on*, Aug 2010, pp. 509–512.

[24] R. Cintra and F. Bayer, "A DCT Approximation for Image Compression," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 579–582, Oct 2011.

[25] F. Bayer and R. Cintra, "DCT-like transform for image compression requires 14 additions only," *Electronics Letters*, vol. 48, no. 15, pp. 919–921, July 2012.

[26] R. J. Cintra, F. M. Bayer, V. A. Coutinho, S. Kulasekera, and A. Madanayake, "DCT-like Transform for Image and Video Compression Requires 10 Additions Only," *CoRR*, vol. abs/1402.5979, 2014. [Online]. Available: http://arxiv.org/abs/1402.5979

[27] U. Sadhvi Potluri, A. Madanayake, R. Cintra, F. Bayer, S. Kulasekera, and A. Edirisuriya, "Improved 8-Point Approximate DCT for Image and Video Compression Requiring Only 14 Additions," *IEEE Trans. Circuits Syst. I*, vol. 61, no. 6, pp. 1727–1740, June 2014.

[28] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. John Wiley, Jan. 1999.

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

[30] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247–269, 1998.

[31] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, *HM 8.0 Reference Software*, 2012 Jul. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-8.0/

[32] M. Masera, *Adaptive Approximated DCT Architectures for HEVC*, 2015 Dec. [Online]. Available: http://personal.det.polito.it/maurizio.masera/material/HEVCSoftware.zip

[33] Y.-J. Chen, S. Oraintara, T. Tran, K. Amaratunga, and T. Nguyen, "Multiplierless approximation of transforms with adder constraint," *IEEE Signal Process. Lett.*, vol. 9, no. 11, pp. 344–347, Nov 2002.

[34] A. Dempster and M. MacLeod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II*, vol. 42, no. 9, pp. 569–577, Sep 1995.

[35] JCT-VC, "Common test conditions and software reference configurations," *JCTVC-J1100*, Jul. 2012.

[36] G. Bjøntegaard, "Calculation of Average PSNR Differences Between RD Curves," *document VCEG-M33, ITU-T SG 16/Q 6*, Apr. 2001.

[37] K. Lee, H. J. Lee, J. Kim, and Y. Choi, "A Novel Algorithm for Zero Block Detection in High Efficiency Video Coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1124–1134, Dec 2013.

Maurizio Masera (S'15) received the B.S. and M.S. degrees in electronic engineering from Politecnico di Torino, in 2012 and 2014 respectively, where he is currently working towards the Ph.D. degree. His research activities include design of digital VLSI circuits and implementation of algorithms and architectures for multimedia applications.

Maurizio Martina (S'98-M'04-SM'15) was born in Pinerolo, Italy, in 1975. He received the M.Sc. and Ph.D. in electrical engineering from Politecnico di Torino, Italy, in 2000 and 2004, respectively. He is currently an Associate Professor of the VLSI-Lab group, Politecnico di Torino. His research activities include VLSI design and implementation of architectures for digital signal processing and communications.

Guido Masera (SM'07) received the Dr. Ing. Degree (summa cum laude) in 1986 and the Ph.D. degree in electronic engineering from the Politecnico di Torino, Torino, Italy, in 1992. Since 1992, he has been an Assistant Professor and then Associate Professor with the Electronic Department, where he is member of the VLSI-Lab group. His research interests include several aspects in the design of digital integrated circuits and systems, with special emphasis on high-performance architecture development and on-chip interconnect modeling and optimization. He is an associate editor of IEEE Transactions on Circuits and Systems II.