

High Performance Multi-Standard Architecture for DCT Computation in H.264/AVC High Profile and HEVC Codecs

Tiago Dias^{‡§†}, Nuno Roma^{††} and Leonel Sousa^{††}
[‡]INESC-ID Lisbon, [†]IST-TU Lisbon, [§]ISEL-PI Lisbon

Rua Alves Redol, 9

1000-029 Lisbon, Portugal

Email: {Tiago.Dias, Nuno.Roma, Leonel.Sousa}@inesc-id.pt

Abstract—A new high performance architecture for the computation of all the DCT operations adopted in the H.264/AVC and HEVC standards is proposed in this paper. Contrasting to other dedicated transform cores, the presented multi-standard transform architecture is supported on a completely configurable, scalable and unified structure, that is able to compute not only the forward and the inverse 8x8 and 4x4 integer DCTs and the 4x4 and 2x2 Hadamard transforms defined in the H.264/AVC standard, but also the 4x4, 8x8, 16x16 and 32x32 integer transforms adopted in HEVC. Experimental results obtained using a Xilinx Virtex-7 FPGA demonstrated the superior performance and hardware efficiency levels provided by the proposed structure, which outperforms its more prominent related designs by at least 1.8 times. When integrated in a multi-core embedded system, this architecture allows the computation, in real-time, of all the transforms mentioned above for resolutions as high as the 8k Ultra High Definition Television (UHDTV) (7680x4320 @ 30fps).

Keywords—Video coding, AVC / HEVC, Integer DCT, Multi-standard architecture, Systolic array, FPGA.

I. INTRODUCTION

Multimedia centric devices and applications, especially those based on digital video, are nowadays increasingly popular as a result of the latest technological advances. Examples of such systems include the latest generation of mobile devices (already offering video telephony and video conferencing services), video surveillance, IPTV and HDTV equipments, which are nowadays required to support both the latest generation video standards and high definition digital video. Consequently, video codecs have become a fundamental building block of multimedia systems, to make them able to efficiently support the encoding and the decoding of such diverse video content in real-time.

Such vast and varied range of devices and applications, with quite distinct characteristics in terms of performance, memory capacity, communication bandwidth and power consumption, has also motivated the proposal of several different video standards in the last decades (e.g., MPEG-1/2/4,

H.261/3/4 [1], AVS, VC-1, HEVC [2]). Albeit some differences in the adopted coding tools, almost all these standards implement the classical block-based motion compensated transform coding scheme and try to maximize the offered compression efficiency for a given video quality. Data compression is mostly achieved in the *transform & quantization* stage of the codec, where transforms are used to decorrelate the residual data and compact its energy at the low frequency domain. Data compression is then provided by the quantizer, which efficiently exploits the reduced sensibility of the human visual system to higher frequency content by adaptively coding the obtained transform coefficients.

Due to the significant impact of the transform operation in the achieved compression efficiency, transform coding has been an active research topic for several decades. Such studies have addressed not only the definition of new transforms to be included in novel video standards, but also the design of efficient algorithms and circuits for their computation. As a result, various transforms with distinct sizes and coefficients have been presented in the literature, although the majority of such proposals are based on the type-II order-8 Discrete Cosine Transform (DCT) [3]. This is mostly owed to the reduced complexity and the highly satisfactory performance of the DCT in terms of energy compaction, as well as to the existence of fast algorithms for its computation.

Nonetheless, higher order transforms are nowadays also being considered by the most recent video standards (e.g., HEVC and AVS), in order to improve the coding efficiency for high definition and resolution contents. Larger transforms provide better energy compaction and reduce the quantization error, which for such video contents greatly increases the compression rate. However, this higher efficiency comes at the expense of increased computational complexity and data processing rates. This poses several challenges to codec and system designers, especially when real-time operation is demanded. This issue is even more problematic for the latest generation multimedia devices, which are required to support several different video standards, and thus to implement several different video codecs and compute their corresponding transforms. Consequently, the investigation of high performance and area efficient Multi-Standard Transform (MST) architectures is nowadays a very active research area. However, up until now very few MST designs with the capability to compute the order-16 and -32 transforms required by state-of-the-art video standards (e.g., HEVC [2]) have been presented [4], [5].

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under the project "HELIX: Heterogeneous Multi-Core Architecture for Biological Sequence Analysis" with reference PTDC/EEA-ELC/113999/2009 and project PEst-OE/EEI/LA0021/2013 and by the Programa de apoio à formação avançada de docentes do Ensino Superior Politécnico (PROTEC) program funds under the research grant SFRH/PROTEC/50152/2009.

In this paper, it is proposed a unified and scalable architecture for the implementation of high performance MST cores. The proposed structure is based on a two-dimensional systolic array transform core, which was recently proposed for the computation of the 4×4 and 2×2 transforms adopted in the baseline, extended and main profiles of the H.264/AVC standard [6]. Such transform core presents a modular and flexible interconnection architecture that not only allows it to be reconfigured to use different transform coefficients, but it also permits its rescaling to efficiently support transforms of multiple sizes. In this work such properties were extensively exploited, in order to design larger and enhanced transform cores capable of addressing all the profiles of both the H.264/AVC and HEVC standards for the processing of high definition and resolution videos in real-time. To achieve such goal, a new Processing Element with multi-standard computation capabilities is also proposed herein.

The rest of this paper is organized as follows. Section II provides a review of the set of transforms adopted in the H.264/AVC and HEVC standards, while the most relevant related works are introduced and discussed in section III. Section IV presents the proposed MST core. The experimental results considering the implementation of such hardware structure in a Xilinx Virtex-7 FPGA device are presented and discussed in section V, by considering other state-of-the-art designs with similar functionalities. Finally, section VI concludes the presentation.

II. TRANSFORM CODING

The type-II DCT [3] has been extensively used in block-based image and video coding (e.g., in JPEG, H.261/3 and MPEG-1/2/4 standards). This is mostly a consequence of its modest complexity and robust approximation of the statistically optimal Karhunen-Loève transform, which allows to obtain implementations with good performance both in terms of computation time and energy compaction.

Several different algorithms for fast and efficient computation of the DCT have been developed. Many of them take advantage of the relationships between the DCT and various other existing fast transforms, while others are based on the sparse factorization of the DCT kernel. Alternative proposals consist also of recursive implementations like the the row-column decomposition strategy, which poses several advantages in terms of complexity, computation time and design effort. Nevertheless, the computation of the involved simpler one-dimensional (1-D) transforms still presents several drawbacks, since most entries of these transform kernels consist of irrational numbers.

On the one hand, because it requires either floating-point implementations or integer approximations using high-precision coefficient values, which involves expensive and slow arithmetic circuits. On the other hand, because it may cause drift (i.e., mismatch between the decoded data in the encoder and decoder), as a result of the forward and inverse transforms being implemented in different machines with distinct floating-point representations/approximations and rounding. Modern video standards like H.264/AVC and HEVC are very sensitive to prediction drift, since they make extensive use of prediction to increase the compression efficiency [1], [2]. To circumvent this problem, in these standards integer orthogonal transforms

with almost the same symmetry and energy compaction capability of the DCT have been adopted.

Although several different approaches can be used to obtain integer transform kernels, dyadic symmetry schemes have been considered in almost all the state-of-the-art video standards. This approach allows not only to compute the transform in reduced bit-width integer arithmetic (e.g., 16-bit), but also by using exclusively integer multiplications and additions. However, since the basis vectors of these integer transform kernels do not have the same norm, this approach often implies an additional normalization stage in the video coding algorithm. In most video standards, this compensation has been integrated into the quantization stage of the video codec, in order to keep the complexity of the codec as low as possible.

A. H.264/AVC transforms

In H.264/AVC [1], a two-level hierarchical transform path and six different transforms are used in the encoding of each block of residual data, namely, the forward and the inverse 8×8 and 4×4 DCTs, the 4×4 and the 2×2 Hadamard transforms. In the first transform level, the 4×4 DCT is applied to all the 4×4 luma and chroma blocks of residual data. These forward and inverse transform kernels, generically represented in (1), contain only four different values each, as enumerated in Table I. This characteristic greatly minimizes the transforms' complexity, by allowing its computation using only 16-bit integer addition and shift operations.

$$C_{4 \times 4} = \begin{bmatrix} a & a & a & a \\ f & g & -g & -f \\ a & -a & -a & a \\ g & -f & f & -g \end{bmatrix} \quad (1)$$

The second level of the transform operation is based on the Hadamard transform and provides the required means to better exploit the redundancies in smoother areas of a frame, i.e., encoding of the DC coefficients of the transformed blocks computed in the previous level. Two distinct Hadamard transforms are used to process the DC coefficients corresponding to the luma and the chroma blocks. For the macroblocks (MBs) that are encoded using the 16×16 Intra-prediction mode (INTRA 16×16), a 4×4 Hadamard transform is used to process the sixteen DC coefficients of all luma blocks. Conversely, a 2×2 Hadamard transform is used to process the two sets of chroma DC coefficients resulting from the four 4×4 chroma blocks of each MB in all coding modes. The 2×2 Hadamard transform kernel consists of a sub-sampled version of the 4×4 kernel, which considers only the first two entries of rows 0 and 2. The two transforms can be computed by exclusively using integer additions and subtractions, since their kernels contain only the 1 and -1 coefficient values (see Table I). Due to the kernel's symmetric nature, the inverse 4×4 and 2×2 Hadamard transforms use the same kernels of their corresponding forward transforms.

The High profiles defined in the Fidelity Range Extension (FRExt) amendment of the H.264/AVC standard also allow the processing of MBs composed of luma blocks with 8×8 samples [1], for which the transform process consists only of the first transform level. In such cases, an 8×8 integer transform is used to compute all the 64 coefficients of the four luma blocks composing the MBs being encoded either

TABLE I. COEFFICIENTS OF THE H.264/AVC AND HEVC TRANSFORM KERNELS.

Transform	Coefficient Values						
	a	b	c	d	e	f	g
H.264/AVC 4×4 Hadamard	1	-	-	-	-	1	1
H.264/AVC 4×4 Forward IDCT	1	-	-	-	-	1	2
H.264/AVC 4×4 Inverse IDCT	1	-	-	-	-	1	$\frac{1}{2}$
H.264/AVC 8×8 Forward IDCT	8	12	10	6	3	8	4
HEVC 4×4 Forward IDCT	64	-	-	-	-	83	36
HEVC 8×8 Forward IDCT	64	89	75	50	18	83	36

in the INTRA 8×8 or INTER prediction modes. Likewise the Hadamard transforms, the forward and inverse 8×8 integer transforms share the same kernel. Nevertheless, since the involved coefficients are of greater magnitude and more diverse (see (2) and Table I), the complexity of these kernels is greater than in the 4×4 kernels. The dynamic gain of these kernels is also relatively higher, which imposes an intermediate rescaling after the row-wise 1-D transform stage to make its computation possible with 16-bit integer arithmetic.

$$C_{8 \times 8} = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & c & d & e & -e & -d & -c & -b \\ f & g & -g & -f & -f & -g & g & f \\ c & -e & -b & -d & d & b & e & -c \\ a & -a & -a & a & a & -a & -a & a \\ d & -b & e & c & -c & -e & b & -d \\ g & -f & f & -g & -g & f & -f & g \\ e & -d & c & -b & b & -c & d & -e \end{bmatrix} \quad (2)$$

B. HEVC transforms

Although HEVC also follows the same block-based hybrid coding scheme used in all video standards since H.261, it employs several new tools and a very different frame coding structure [2] to increase the compression in about 50% when compared to H.264/AVC.

In this new standard, each frame is divided into a sequence of square units called Largest Coding Units (LCUs), which hold the information of chrominance and luminance. Each LCU is composed of one or more basic Coding Units (CUs) that can be recursively subdivided in four equally sized blocks, starting from the 64×64 samples LCU and going all the way down to a minimum of 8×8 samples. Such generic quadtree segmentation structure also indicates the subdivision of the CUs into Prediction Units (PUs), used for Intra- and Inter-prediction, and Transform Units (TUs), which are the basic units defined for transform and quantization processing. As occurs with the CUs, TUs are also represented as a quadtree structure using squared TU blocks. The dimension of these blocks can vary from 4×4 to 32×32 samples, therefore requiring the implementation of transforms with multiple sizes.

The integer transforms used in HEVC are better approximations of the DCT than the transforms used in H.264/AVC. Such transform kernels were derived by approximating scaled DCT basis functions, under considerations such as limiting the necessary dynamic range for transform computation and maximizing the precision and closeness to orthogonality when the kernel entries are specified as integer values. Consequently, the basis vectors of the HEVC transforms have equal energy and there is no need to compensate for the different norms, as in H.264/AVC. Furthermore, such property also allows to use the same kernels for the computation of both the forward and of the inverse transforms. However, due to the increased

dynamic range of the supported transform kernels, HEVC explicitly inserts rescaling and 16-bit clipping operations after the row-wise transform stage, in order to ensure the possibility to compute all the transforms in 16-bit integer arithmetic.

To reduce the complexity of the encoder and to simplify the computation of the transforms, only one order-32 transform kernel is specified in HEVC [2]. The remaining lower order kernels consist of sub-sampled versions of such kernel. The entries of these order- k kernels (with $k = 4, 8, 16$) consist of the first k values of rows $k \times j$ (with $j = 0 \dots \frac{32}{k} - 1$) of the 32×32 kernel. All the lower order kernels also present key symmetry properties, to enable fast "partially-factored" implementations using very few mathematical operations. As an example of this feature, the HEVC 8×8 transform kernel can be obtained from (2), by considering the corresponding coefficients as enumerated in Table I.

III. RELATED WORK

In the last few years, several different transform cores have been presented in the literature. Although the majority of such structures consist of dedicated architectures targeting efficient VLSI realizations, a couple of reconfigurable solutions have also been presented. Independently of the considered technology, these designs typically implement fast and optimized algorithms for a single transform by using high performance structures, in order to mitigate the involved complexity constraints and to speedup the corresponding computations. Consequently, such architectures are usually denominated as *dedicated transform cores* and are typically used in the design of either the forward or the inverse transform modules of video codecs presenting quite strict performance constraints.

At this respect, direct two-dimensional (2-D) processing structures are usually employed when high performance implementations are desired. Such implementations are characterized by their highly parallel processing capability, which allows them to simultaneously compute several different transform coefficients. However, they are also characterized by requiring a significant design effort and by imposing higher hardware cost and power consumption requirements. As a consequence, not so many direct 2-D transform architectures have been proposed. The most well known of those direct 2-D structures consists of the butterfly architecture proposed by Chen for the computation of the 8×8 DCT, adopted in the JPEG image standard and in the MPEG-1/2/4 and H.261/3 video standards. Other direct 2-D designs targeting more recent standards have also been devised by using quite similar approaches, like the ones presented in [7], [8] for the computation of the 4×4 transforms defined in H.264/AVC.

Nevertheless, the technique introduced by Chen has also been frequently considered in the development of architectures for the computation of 1-D transforms. With some transposition logic, these simpler processing structures are typically employed to compute 2-D transforms by using the row-column decomposition approach. The hardware efficiency of these designs is therefore relatively higher, since the same 1-D transform circuit is used twice in the computation of a 2-D transform. Moreover, such improved hardware efficiency may also contribute to a reduction of the power consumption and hardware cost associated with this type of architectures. Nonetheless, in most practical cases such gains are usually

modest, owing to the usage of quite large and complex memory based circuits to realize the transposition operation. In addition, these circuits also introduce some latency in the computation of the 2-D transforms, which can compromise the implementation of real-time operations. Still, numerous designs of direct 1-D architectures have been proposed to compute 2-D transforms.

Although most of these architectures share the same functionality, they are quite distinct owing to the different optimization techniques that were used to improve their computational efficiency. In particular, CORDIC arithmetic was used in the design of some of these structures not only to avoid the multiplication operations, but also to improve the performance and hardware efficiency of their implementation. Other designs make use of distributed arithmetic to avoid the computation of multiplications. In what concerns the integer transforms adopted by the most recent video standards, such as H.264/AVC, AVS, VC-1 or HEVC, some 1-D structures have also been proposed for the computation of the forward and inverse transforms [4], [9]. Other cost effective designs were described in [10] and the implementation of RNS-based and of systolic array structures have also been reported.

Contrasting with the *dedicated transform cores* previously referred, the majority of the hardware structures that have been proposed for the newest video standards consist of a distinct class of architectures, denominated as *multi-transform cores*. When such more elaborated designs are only capable of computing the set of transforms employed exclusively by a video standard (e.g., all the forward and the inverse integer DCTs and Hadamard transforms defined in H.264/AVC) they are subclassified as *unified transform cores* [11], [12], [13], [14]. However, if they are designed to implement the several different transforms used by various standards, they are subclassified as Multi-Standard Transform (MST) cores [5], [15], [16]. In order to achieve both goals, a couple of different approaches have been considered to devise this type of architectures.

The simplest designs consist of architectures implementing all the required transforms, by adopting a quite straightforward methodology: make use of a distinct, independent and specifically optimized transform core to compute each of the considered transforms and then select the required results, according to the type of transform that is required [17]. Since all the transforms are computed in parallel, the involved hardware cost, memory bandwidth and power consumption values are very high and often prohibitive for implementations targeting mobile or low power video coding systems. As a result, several alternative architectures featuring the sharing of hardware resources have been presented to implement more efficient multi-transform processing structures.

Some of these designs are usually derived by jointly optimizing the algorithms of the considered set of transforms using different techniques (e.g., the Kronecker product [10] and 1-D delta matrix mapping [18]). However, there are other hardware sharing solutions that time multiplex the use of some parts of the computational circuits, in order to realize distinct transforms [19]. Still, such efficiency is not optimal when reconfigurable implementation technologies are considered, because several hardware blocks that are exclusively required to implement a specific transform operation are permanently instantiated. As a result, some reconfigurable architectures with the capability to support the computation of multiple transforms have also been recently proposed [6].

Among all the reviewed transform cores, the design proposed in [6] presents two distinctive properties (i.e., modularity and scalability) that can also be exploited to design high performance and hardware efficient MST cores. As a result, such hardware structure was used as a basis to this work.

IV. UNIFIED ARCHITECTURE FOR MST CORES

The proposed MST architecture was designed to fully address the transform computation requirements of video codecs supporting all the profiles of the H.264/AVC and HEVC standards. Consequently, it is capable of computing not only the forward and the inverse 8×8 and 4×4 integer DCTs and the 4×4 and 2×2 Hadamard transforms defined in the H.264/AVC standard, but also the 4×4 , 8×8 , 16×16 and 32×32 integer transforms adopted in HEVC.

To compute this vast set of 2-D transforms, the hardware structure herein proposed applies the row-column decomposition approach and adopts the same basis structure as the multi-transform core presented in [6]. However, in this work the scalability property of such structure was used to develop larger designs capable of supporting the 8×8 , 16×16 and 32×32 transforms. Furthermore, its modularity property was also used to equip these larger transform cores with a new Processing Element (PE), first presented in this paper, which is capable of efficiently computing all the operations required by the set of transforms previously mentioned.

In the following subsections, the architecture used as a basis to this work is briefly reviewed and the proposed MST architecture is thoroughly presented.

A. Base architecture

The transform core proposed in [6] computes all the 4×4 and 2×2 transforms defined in H.264/AVC, by implementing the row-column decomposition strategy. To achieve such goal, this processing structure makes use of only four distinct functional modules: a *2-D systolic array*, a *transposition switch*, an *input buffer* and a *control unit*, as depicted in Fig. 1.

The heart of this structure consists of a systolic array with 4×4 PEs, which is used to compute all the considered H.264/AVC transforms. Within this structure, the PEs perform the same operations and share the same architecture supporting all the required calculations. Such architecture consists of two main blocks: *i)* an *arithmetic module*, where the transform computations are realized by using an accumulator and a specialized multiplication circuit to compute the involved MAC operations; and *ii)* a *control module*, which is responsible for generating the control signals for the PE, as well as for guaranteeing the correct flow of such signals into the remaining PEs of the array (in the horizontal and vertical directions).

The *transposition switch* is used to implement the row-column transposition of the data processed inside the systolic array. Unlike other transposition units that have been proposed, this circuit does not use additional memory resources and mostly consists of a set of multiplexers that allow a fast and direct row-column transposition of the data.

The *input buffer* is used to feed the PEs of the systolic array with the data to be transformed. Such data consists either of the residues from the Intra- and Inter-predictions when forward transforms are considered, or of the transform

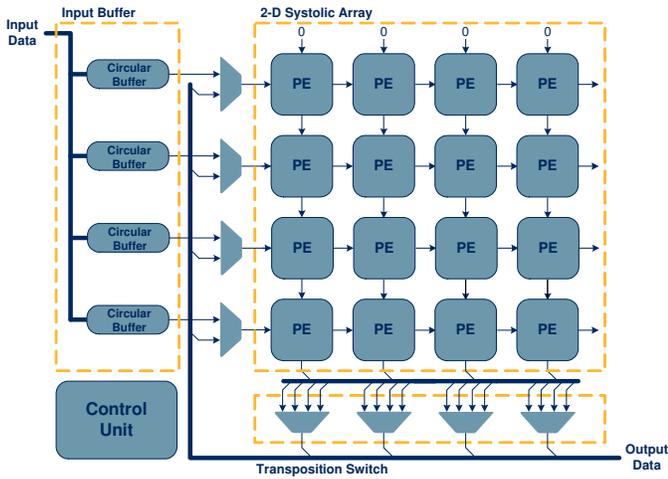


Fig. 1. Block diagram of the architecture presented in [6].

coefficients when inverse transforms must be computed. This unit allows not only to minimize the delays when accessing the external data memories where such data is stored, by efficiently exploiting appropriate cache access patterns, but also to guarantee a regular dataflow within the systolic array. With this module, data values are serially and smoothly transferred to the several rows of the array, therefore supporting the implemented pipelined dataflow.

By adopting this streaming model, the data is processed in a wavefront manner within the array of PEs. Such data is fed into the PEs through the input buffers in the left column of the array, and are then propagated in the horizontal and vertical directions to the remaining PEs. Conversely, the control signals for all the PEs enter the array through the top-left corner PE, which then propagates them to the remaining PEs also in both directions, synchronously with the data propagation. Such signals are also propagated to the transposition switch, where they are used by the control logic to select the proper data to be feedback to each row of the array. With this processing scheme it possible to start the computation of a different transform value in each row of the array on each clock cycle (provided that the input buffers are not empty), which maximizes the data processing rate.

Finally, the *control unit* is responsible for controlling the systolic array, as well as the operation of both the input buffer and the transposition switch. It is also in charge of implementing the necessary synchronization mechanisms between the transform core and the outer video coding system that incorporates this dedicated processing structure. For more information concerning this architecture, the interested reader is referred to [6].

B. Proposed multi-standard architecture

The proposed MST architecture, which consists of a major evolution of the structure presented in [6], is composed also by a 2-D systolic array, a transposition switch, an input buffer and a control unit, which are connected to each other using the same interconnection scheme depicted in Fig. 1. However, the internal structures of all these blocks significantly differ from the ones presented in [6], making it possible to support the computation of order-8, -16 and -32 transforms. Moreover, the dataflow model of the original transform core was also

extended with the definition of a new *PE configuration stage*. Such approach allows to minimize the latency imposed by the PEs and consequently increase the data processing rate of the proposed MST architecture.

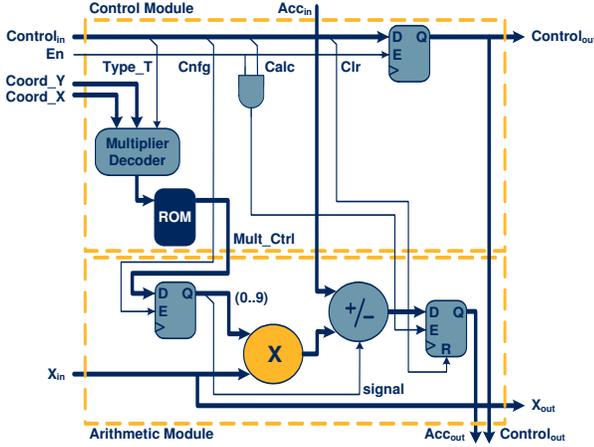
In what concerns the input buffer, the devised block consists of a scalable hardware structure similar to the one represented in Fig. 1, which can now be configured to operate using either 8, 16 or 32 circular buffers. This approach was also adopted to design the new transposition switch, which can now be configured to include either eight 8×1 , sixteen 16×1 or thirty two 32×1 fast switching circuits, depending on the size of the instantiated systolic array.

The major changes implemented in the new systolic array focused not only the scaling of the hardware structure to support arrays with 8×8 , 16×16 and 32×32 PEs, but also the definition of a completely new architecture for such PEs. This results from the significant impact of this circuit in both the performance and the hardware cost of the whole architecture. Similarly to the original PE design, the architecture that was developed for the new PEs is also split into the same two top-level arithmetic and control blocks, as it can be seen in Fig. 2a. Nonetheless, a completely new approach was adopted in the design of this unified architecture for the H.264/AVC and HEVC standards, which supports the computation of all the operations of 16 different transforms, totalizing 70 distinct kernel entries.

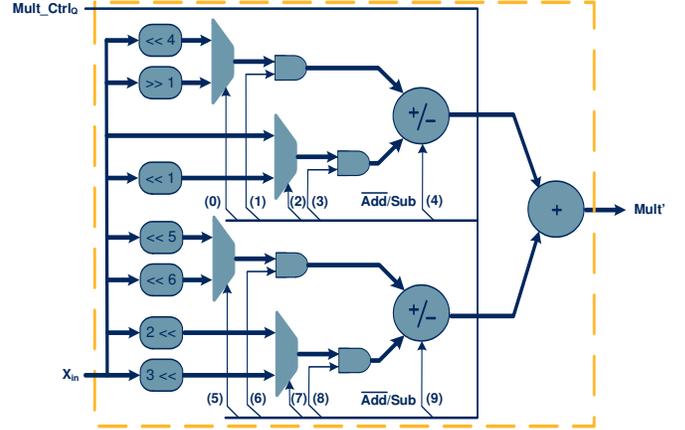
To minimize the complexity of the new PE, the devised control module implements a quite different algorithm to determine the multiplication values used in the MAC operation. The rationale behind this idea results from the observation that only N different basis values exist in any of the considered $N \times N$ transform kernels [1], [2]. Such values consist of the first column-vector of the transform kernel, which correspond to the subset of kernel entries matching the DCT cosine angles in the range of $[0, \frac{\pi}{2}]$. Consequently, they can also be used to generate the remaining $N \times (N - 1)$ values of the transform kernel, provided that the symmetry and periodicity trigonometric properties of the cosine function are properly exploited. To achieve such goal, the new control module makes use of a very simple integer arithmetic circuit (identified as *Multiplier Decoder* in Fig. 2a) and of a small ROM module to generate the multiplication terms for all the considered transform kernels coefficients. The operation of such circuit for a $N \times N$ kernel is the following.

First, the row (r) and column (c) coordinates of the PE inside the systolic array are used to determine the amount α of $\frac{\pi}{2N}$ fractions (e.g., $\alpha = (2i + 1) \times r$) of the cosine angle corresponding to the considered kernel entry, as specified in the generic formulation of the 2-D DCT shown in (3). Although such computations are performed in integer arithmetic using $\lceil \log_2 N \rceil + 2$ bits ($i, j = 0, \dots, N - 1$), only the $\lceil \log_2 N \rceil$ least significant bits of α are of practical interest, because they correspond to the angle value in the range $[0, 2\pi[$. The remaining bits only specify the amount of times the 2π domain was exceeded, and can therefore be ignored.

$$X_{r,c} = \sqrt{\frac{2}{N}} \epsilon_r \epsilon_c \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{i,j} \times \cos \left[(2i + 1) \frac{r\pi}{2N} \right] \times \cos \left[(2j + 1) \frac{c\pi}{2N} \right], \quad r, c = 0, \dots, N - 1 \quad (3)$$



(a) Block diagram.



(b) Architecture of the MCM unit of the arithmetic module.

Fig. 2. Proposed multi-standard PE.

$$\epsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0, \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

Then, the two most significant bits of this constrained result α' are evaluated, in order to determine if the computed result is in the ranges $[\frac{\pi}{2}, \pi[$, $[\pi, \frac{3\pi}{2}[$ or $[\frac{3\pi}{2}, 2\pi[$. In such cases, the proper value of α' is computed, so as to convert the angle into the range $[0, \frac{\pi}{2}]$. This final result is then used to address the ROM, in order to retrieve the corresponding multiplier control word. Such data, which corresponds to the absolute value of the considered transform kernel entry, is used to control the operation of the Multiple Constant Multiplier (MCM) circuit embedded in the PE arithmetic module, as it is described below and represented in Fig. 2b. With such approach, the proposed PE architecture adopts a 64×10 -bits ROM to store all the multiplier control words corresponding to the 32 coefficients of the H.264/AVC transforms (i.e., the 8×8 and 4×4 DCTs and the Hadamard transforms) and the 32 coefficients of the HEVC transforms. These data values are presented in hexadecimal notation in Table II for the entries of the kernels shown in Table I. Finally, the sign information is computed by considering the magnitude of α' . If such value is in the range $[\frac{\pi}{2}, \frac{3\pi}{2}]$, the considered transform kernel entry represents a negative number.

In contrast with the original PE, the obtained multiplier control word and sign information bit are not directly applied to the arithmetic module of the PE. As it can be seen in Fig. 2a, such data is stored in an internal data-standing register in the proposed MST PE, so that it can be used in the arithmetic module to compute the MAC operations in the subsequent clock cycles. This modification allows to significantly improve the processing rate of the PEs, since it greatly reduces the circuit's critical path. This improvement is of the utmost importance for the proposed MST architecture, because all the computational circuits in the new arithmetic module operate

TABLE II. MULTIPLIER CONTROL WORDS FOR THE H.264/AVC AND HEVC TRANSFORM KERNELS (SEE TABLE I).

Kernel Value	$\frac{1}{2}$	1	2	3	4	6	8	10
ROM Word	0x3	0x8	0xC	0x118	0x100	0x10C	0x180	0x18C
Kernel Value	12	18	36	50	64	75	83	89
ROM Word	0xFE	0xE	0x140	0x4E	0x60	0x86	0x17A	0x1EA

in 27-bit integer arithmetic, as a result of the higher dynamic gains imposed by the higher order H.264/AVC and HEVC transform kernels. Nonetheless, this modification also imposes a different dataflow model for the proposed MST architecture, which must take into account an explicit PE configuration stage (prior to the computation of the transform) whenever the MST core is reprogrammed to compute a different transform.

Just like in the original dataflow model, the PEs are configured in a wavefront manner within the array. The command that triggers such event is also sent through the top-left corner PE, which then propagates it to the remaining PEs in both the horizontal and vertical directions. Since this operation requires only one clock cycle and is performed in a pipelined fashion, the new dataflow model makes it possible to start the computation of a new transform operation in each PE on the clock cycle that immediately succeeds its configuration stage. As a result, only when it is necessary to compute transforms that use a kernel different from the one currently in use within the systolic array, does this new PE architecture increase the latency of the proposed MST core in one clock cycle.

In what concerns the arithmetic module, the architecture that is now proposed incorporates a new specialized multiplication circuit, capable of efficiently computing the multiplications involving all the new transform kernel values. As it can be seen in Fig. 2b, the specially developed time-multiplexed MCM realizes such operations using a 2-step shift-and-add procedure, which is based on the factorization of the transform kernel values in powers of two. In the first step, the multiplicand value is shifted by all the possible power of 2 components that compose the values of the considered transform kernel entries. Then, those intermediate results are properly combined using additions and subtractions in a 2-level adder tree structure, by using the multiplier control word that is stored in the PE's internal data-standing register. The multiplication result is only evaluated in conjunction with the accumulation operation, in order to save an extra and redundant adder. Hence, this operation may consist either of an addition or of a subtraction, depending on the sign information that is stored in the PE's internal data-standing register. Compared to a generic (signed) binary multiplier, the devised MCM allows not only to significantly reduce the multiplication computation time, but also to greatly reduce the hardware cost of the PEs.

V. EXPERIMENTAL EVALUATION

In order to experimentally assess the advantages offered by the proposed MST architecture in the design of highly efficient transform cores for video codecs, a dedicated transform coding processor targeting the High Profiles of the H.264/AVC standard was realized. The implemented processing structure incorporates a systolic array with 8×8 PEs and supports the computation of the forward and the inverse 8×8 and 4×4 DCTs, as well as the 4×4 and 2×2 Hadamard transforms.

Naturally, other possible configurations of the proposed MST architecture could have been similarly obtained for both the H.264/AVC and the HEVC standards. In such cases, it would only be required to consider a different configuration for the proposed MST architecture, which for efficiency reasons should include a systolic array at least of the same dimension as the size of the greatest transform to be computed. Nonetheless, in our experiment we have chosen to consider an implementation supporting only the H.264/AVC standard, because more alternative designs can be used in our comparative analysis, owing to the established maturity of this standard.

Nevertheless, it should be highlighted that this design choice does not compromise neither the relevance of the proposed MST architecture, nor the benefits that it offers in terms of performance and functionality when compared with other alternative solutions, as it is discussed in section V-B.

A. FPGA implementation

The considered transform core was synthesized for a Xilinx Virtex-7 XC7VX485T FPGA device, by using the Xilinx ISE 13.2i tool chain and a parameterizable IEEE-VHDL description of the presented MST architecture. Such generic architectural description follows a strict modular approach using independent and self-contained functional blocks, in order to comply with the architecture's scalability requirements and multi-standard transform functionality. Table III presents the results that were obtained for such implementation, which followed a standard synthesis procedure targeting performance optimized circuits.

In what concerns the performance, the maximum clock frequency value presented in Table III reveals that the implemented transform core is able to compute 17.95 GOPS. This processing rate almost doubles the processing rate of the architecture proposed in [6], with a maximum clock frequency of 147 MHz for a Virtex-5 FPGA. Besides the more powerful prototyping technology now considered, the significant improvement that is now attained is owed to two different aspects: *i*) the parallel and highly efficient pipelined datapath implemented by the 2-D systolic array of PEs; and most importantly, to the new highly optimized PE herein presented. As a result, by using the new PE the implemented transform core is able to compute the whole set of H.264/AVC transforms in real-time, for video sequences with resolutions up to 7680×4320 pixels (8k UHDTV resolution) and using frame rates up to 30 fps.

The results presented in Table III also demonstrate the quite reduced hardware cost of the proposed MST architecture, while offering a remarkable processing rate. As it can be seen, less than 30% of the total hardware resources available in the adopted medium-size FPGA device were used in the

TABLE III. IMPLEMENTATION RESULTS OF THE PROPOSED MST CORE WITH AN ARRAY OF 8×8 PEs IN A VIRTEX-7 FPGA.

Processing structure	Registers	LUTs	Max. F.
☐ Proposed MST core	6420	21452	280.6 MHz
☐ Input buffer	1681	1264	344.8 MHz
☐ Array of 8×8 PEs	1501	15727	296.5 MHz
☐ PE	70	254	298.5 MHz
☐ Arithmetic module	0	177	329.8 MHz
☐ Control module	70	78	298.9 MHz
☐ Transposition switch	1462	2161	360.0 MHz
☐ Control unit	97	77	366.4 MHz

realization of the considered transform core. Almost all of these hardware resources (about 85%) are used to implement the 2-D systolic array, with each of its 8×8 MST PEs requiring only about 70 slices (i.e., less than 2% of the total implementation resources). Therefore, it can be observed that while the hardware cost of the proposed MST architecture mostly results from the adopted configuration for its systolic array, its maximum clock frequency is only limited by the critical path of the PEs. Consequently, it can be expected that very similar performance levels can be attained for other configurations of the proposed MST architecture making use of larger systolic arrays (e.g., an array with 32×32 PEs to compute the more complex HEVC transform).

B. Comparative analysis

In order to demonstrate the advantages offered by the proposed MST architecture in the design of video codecs for the most recent video standards, several different alternative designs recently described in the literature for this domain were reviewed. Due to the diverse set of implementation technologies that have been considered, as well as to the distinct considerations that might have been adopted by the authors and the different functionalities presented by each implementation, this analysis focused mostly on a comparison of the functionalities and computational rates offered by the various structures. Table IV summarizes the results of such comparative analysis for the subset of the most related and prominent designs that were evaluated, for which the presented data concerns only to the implementation of the transform computation module(s) of the considered transform cores.

A straightforward analysis of the data presented in Table IV clearly shows that only a few of the reviewed designs are able to compute all the H.264/AVC transforms. In what concerns the HEVC standard, the design presented in [5] is able to compute all its inverse transforms, while the design in [15] is capable of realizing only the 8×8 inverse transform. In fact, to the best of the authors' knowledge, the presented MST architecture is one of the first structures that is able to compute the complete set of transforms adopted in both the H.264/AVC and HEVC standards.

In what concerns to performance, the data that is presented in Table IV clearly shows that the implemented MST core offers one of the highest computation rates (2.2×10^6 samples per second – GSamp/s), despite being one of the few architectures with multi-transform computation capabilities. In fact, it can be observed that this design outperforms almost all the other considered architectures by about 1.8 times. The only exceptions are the designs presented in [6], [14] and [11]. However, these processing structures are able to compute only a quite reduced subset of the transforms supported by the proposed MST core, since they were either specifically

TABLE IV. FUNCTIONAL AND PERFORMANCE COMPARISON WITH OTHER ARCHITECTURES.

Design	Technology	Max. F. [MHz]	Latency [ns]	Throughput [GSamp/s]	Supported Transforms								Applications	
					H.264/AVC				HEVC					
					$DCT_{8 \times 8}$	$DCT_{4 \times 4}$	$H_{4 \times 4}$	$H_{2 \times 2}$	4×4	8×8	16×16	32×32		
[18]	180 nm	146.0	438.4	0.2	Inv									HD 1080p
[8]	180 nm	100.0	20.0	0.8		✓	✓							Digital Cinema 2k
[12]	180 nm	230.9	34.7	1.9	Inv	Inv	✓	✓						8K UHD TV (4320p)
[11]	180 nm	300.0	53.3	2.4	Inv									HD 720p
[13]	180 nm	200.0	20.0	4.1	✓	✓	✓	✓						HD 2160p
[15]	180 nm	211.4	56.8	0.2	Inv					Inv				HD 1080p
[16]	130 nm	384.0	83.3	0.8	Inv	Inv								4K UHD TV (2160p)
[17]	130 nm	100.0	1000.0	0.4	✓	✓								HD 1080p
[14]	130 nm	300.0	70.0	3.7	Fwd	Fwd	✓							4K UHD TV (2160p)
[5]	130 nm	350.0	-	1.4	Inv	Inv	Inv	Inv	Inv	Inv	Inv	Inv	Inv	4K UHD TV (2160p)
[19]	Virtex-4	110.8	144.4	0.9	Inv	✓	✓							4K UHD TV (2160p)
[12]	Virtex-4	133.5	59.9	1.1	✓	✓	✓	✓						4K UHD TV (2160p)
[6]	Virtex-5	317.7	25.2	2.5		✓	✓	✓						8K UHD TV (4320p)
Proposed	Virtex-7	280.6	57.0	2.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	8K UHD TV (4320p)

designed to accelerate the inverse 8×8 DCT [11], optimized for the implementation of the forward transform coding path of a video encoder [14], or only support the computation of the 4×4 and 2×2 transforms [6].

The results presented in Table IV also demonstrate that the processing delay that is imposed by the implemented MST core is small, when compared to the majority of the considered alternative designs. This is a quite important aspect when real-time operation is considered, since the latency that is imposed to the encoding of each video frame may compromise the operation of the codec in "live" mode. Hence, besides its increased functionality (i.e., its ability to compute all the transforms defined in the H.264/AVC and HEVC standards), the MST core herein presented is capable of achieving real-time operation for high-definition video sequences in the 8k UHD TV format ($7680 \times 4320 @ 30$ fps).

VI. CONCLUSIONS

A new high performance multi-standard architecture for the computation of the DCTs defined in several modern video standards was proposed in this paper. To the best of the authors' knowledge, this is one of the first structures capable of computing all the DCT operations that are defined in the H.264/AVC and HEVC standards. Contrasting with other dedicated transform cores, the presented architecture is supported on a completely configurable and scalable structure, capable of providing the processing rates that are required to compute, in real-time, all the transforms in HD video sequences. When prototyped in a Xilinx Virtex-7 FPGA device, the proposed architecture was able to process 8k UHD TV sequences (7680×4320) at a frame-rate of 30 fps.

REFERENCES

- [1] T. Wiegand, G. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [2] K. Ugur *et al.*, "High performance, low complexity video coding and the emerging HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1688–1697, Dec. 2010.
- [3] K. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990.
- [4] J.-S. Park, W.-J. Nam, S.-M. Han, and S. Lee, "2-D large inverse transform (16×16 , 32×32) for HEVC (High Efficiency Video Coding)," *Journal Semicond. Technol. Sci.*, vol. 12, no. 2, pp. 203–211, Jun. 2012.
- [5] 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 *2012 IEEE Int. Conf. Multimedia Expo (ICME)*, 2012, pp. 788–793.
- [6] T. Dias, S. López, N. Roma, and L. Sousa, "Scalable unified transform architecture for advanced video coding embedded systems," *Int. Journal Parallel Programming*, vol. 41, no. 2, pp. 236–260, Apr. 2013.
- [7] C.-P. Fan, "Fast 2-dimensional 4×4 forward integer transform implementation for H.264/AVC," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 3, pp. 174–177, Mar. 2006.
- [8] K.-H. Chen, J.-I. Guo, and J.-S. Wang, "A high-performance direct 2-D transform coding IP design for MPEG-4 AVC/H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 472–483, Apr. 2006.
- [9] L. Ling-Zhi, Q. Lin, R. Meng-Tian, and J. Li, "A 2-D forward/inverse integer transform processor of H.264 based on highly-parallel architecture," in *4th IEEE Int. Workshop System-on-Chip Real-Time Applications*, Jul. 2004, pp. 158–161.
- [10] C.-P. Fan, "Cost-effective hardware sharing architectures of fast 8×8 and 4×4 integer transforms for H.264/AVC," in *2006 IEEE Asia Pacific Conf. Circuits Syst. (APCCAS'06)*, Dec. 2006, pp. 776–779.
- [11] C. Jiang, N. Yu, and M. Gu, "A novel VLSI architecture of 8×8 integer DCT based on H.264/AVC FRext," in *3rd Int. Symp. Knowl. Acquisition Modeling (KAM)*, Oct. 2010, pp. 59–62.
- [12] T. Do and T. Le, "High throughput area-efficient SoC-based forward/inverse integer transforms for H.264/AVC," in *Proc. 2010 IEEE Int. Symp. Circuits Syst.*, Jun. 2010, pp. 4113–4116.
- [13] W. Hwangbo and C.-M. Kyung, "A multitransform architecture for H.264/AVC high-profile coders," *IEEE Trans. Multimedia*, vol. 12, no. 3, pp. 157–167, 2010.
- [14] M. Gua, N. Yu, C. Jiang, and W. Lu, "Hardware prototyping for various transforms in H.264 high profile," *Journal Inform. Comput. Sci.*, vol. 8, no. 1, pp. 119–128, 2011.
- [15] M. Martuza and K. Wahid, "A cost effective implementation of 8×8 transform of HEVC from H.264/AVC," in *25th IEEE Canadian Conf. Elect. Comput. Eng. (CCECE)*, 2012, pp. 1–4.
- [16] T. Ho *et al.*, "A 768 megapixels/sec inverse transform with hybrid architecture for multi-standard decoder," in *IEEE 9th Int. Conf. ASIC (ASICON)*, Oct. 2011, pp. 71–74.
- [17] K. Wang *et al.*, "A reconfigurable multi-transform VLSI architecture supporting video codec design," *IEEE Trans. Circuits Syst. II*, vol. 58, no. 7, pp. 432–436, Jul. 2011.
- [18] K. Wahid, M. Martuza, M. Das, and C. McCrosky, "Resource shared architecture of multiple transforms for multiple video codecs," in *24th Canadian Conf. Elect. Comput. Eng. (CCECE)*, May 2011, pp. 947–950.
- [19] G. Liu, "An area-efficient IDCT architecture for multiple video standards," in *2nd Int. Conf. Information Sci. Eng. (ICISE)*, Dec. 2010, pp. 3518–3522.