

Conference on Electronics, Telecommunications and Computers – CETC 2013

## Exploiting Coarse-Grained Parallelism in Multi-Transform Architectures for H.264/AVC High Profile Codecs

Tiago Dias<sup>a,b,\*</sup>, Nuno Roma<sup>a</sup>, Leonel Sousa<sup>a</sup>

<sup>a</sup>INESC-ID / IST – ULisboa, Rua Alves Redol 9, 1000-029 Lisbon, Portugal

<sup>b</sup>ISEL-PI Lisbon, Rua Conselheiro Emídio Navarro 1, 1959-007 Lisbon, Portugal

---

### Abstract

A parallel Multi-Transform Architecture (MTA) for the computation of the 2-D transforms adopted in modern digital video standards is proposed in this paper. This hardware structure can be dynamically configured to efficiently compute either one transform of size  $N \times N$ , or  $k$  different transforms of size  $\frac{N}{k} \times \frac{N}{k}$  in simultaneous, where  $N \in \mathbb{N}$  and  $k = 2^i$  with  $i = 1, \dots, \log_2 N - 1$ . The advantages offered by the proposed parallel architecture were assessed by implementing in a Xilinx Virtex-7 FPGA a proof-of-concept transform core compliant with the High Profiles of the H.264/AVC standard. The obtained results show that such processing structure is capable of achieving real-time operation for video sequences in the 8k Ultra High Definition Television (UHDTV) format ( $7680 \times 4320$  @ 30 fps). In addition, these results also demonstrate that the proposed parallel MTA allows to, at least, double both the throughput and the hardware efficiency of the implemented transform cores, when compared to the original design of the architecture.

© 2014 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of ISEL – Instituto Superior de Engenharia de Lisboa, Lisbon, PORTUGAL.

**Keywords:** Video coding, Transform, Parallel architecture, H.264/AVC, FPGA.

---

### 1. Introduction

Transform-based coding has been an active research topic in the definition of video standards and its corresponding codecs for several decades. This results not only from the significant impact of this mandatory tool in the compression performance that can be provided by a video codec [1], but also from the need to efficiently implement all the different transforms defined in the several video standards that have been proposed in the latter years [2]. For example, the legacy ITU-T H.261/3 and ISO/IEC MPEG-1/2/4 standards adopted only the type-II order-8 Discrete Cosine Transform (DCT) [3], while the newest video standards (i.e., the H.264/AVC [4] and the H.265/HEVC [5] ITU-T recommendations or the AVS and the VC-1 standards [2]) consider multiple integer orthogonal transforms with different orders (e.g., order-2, 4, 8 and 16) and transformation kernels (e.g., DCT, Hadamard, etc).

As a consequence of this diversity, the research effort has concentrated also in the investigation of efficient architectures for the computation of the transformation procedure. In fact, such investigation line mostly resulted from the

---

\* Corresponding author. Tel.: +351-2131003780 ; fax: +351-213145843.  
E-mail address: [Tiago.Dias@inesc-id.pt](mailto:Tiago.Dias@inesc-id.pt)

latest trends in the design of video consumer and professional electronic systems, which nowadays are required to support several different video applications, such as video telephony and video conferencing, Internet video streaming, video surveillance, IPTV or digital TV broadcasting. These applications are typically implemented using distinct digital video standards, which requires a system to be able to compute several different transforms. Furthermore, currently these multimedia centric devices are already required to deal with high spatial and temporal resolutions (e.g.,  $2560 \times 1440$  Quad-HD @ 30 fps [6]) and provide real-time operation, which involves the processing of huge amounts of data in a reduced amount of time.

Due to all these constraints and requirements, the design of modern video coding systems and applications quite often requires efficient hardware realizations of the involved transform computation algorithms, especially when embedded systems are considered. Such specialized architectures are required to support the high data throughput, the high computational rates and the low latency requirements of the video codecs, as well as to guarantee both hardware and power efficient design approaches. Consequently, several different processing structures have been presented in the literature to optimize the computation of such vast set of transforms.

Most of the proposed designs consist of dedicated architectures targeting efficient VLSI realizations [7–10], although a couple of reconfigurable solutions have also been proposed [11,12]. Independently of the considered technology, the majority of the hardware structures that have been devised for the newest video standards already consist of Multi-Transform Architectures (MTAs), and thus support the computation of several different transforms [13–15]. However, the performance and the hardware efficiency of a large part of these structures is usually not optimal [16,17], since most of its computational units are idle during the processing of the lower order transforms (e.g., the  $4 \times 4$  and the  $2 \times 2$  transforms defined in H.264/AVC [4] and the  $8 \times 8$  and  $4 \times 4$  transforms adopted in H.265/HEVC [5]). This is also the case of the design presented in [18], which is one of the first structures capable of computing all the transforms defined in both the H.264/AVC and H.265/HEVC standards. Nonetheless, this processing structure presents some interesting and important characteristics (i.e., modularity and a flexible interconnection structure) that can be exploited to provide support for *coarse-grain data-level parallelism*, and thus greatly improve the offered performance and hardware efficiency levels.

In this paper, it is presented an efficient parallel Multi-Transform Architecture (MTA) for the implementation of high performance and hardware efficient transform cores, compliant with the requirements of the state-of-the-art digital video standards (e.g., H.264/AVC and H.265/HEVC). The proposed structure consists of a greatly enhanced version of the architecture presented in [18], which is capable of efficiently supporting the simultaneous computation of several transforms within its datapath. By adopting the novel approach herein presented, it becomes possible to use the proposed parallel MTA to calculate a single  $N \times N$  transform, as well as to use the same computational resources to simultaneously process  $k$  smaller transforms of size  $\frac{N}{k} \times \frac{N}{k}$ , where  $N \in \mathbb{N}$  and  $k = 2^i$  with  $i = 1, \dots, \log_2 N - 1$ .

The rest of this paper is organized as follows. Section 2 briefly reviews the architecture that was used as a basis for this work, while the proposed parallel MTA is presented in section 3. The experimental results concerning the implementation of this new hardware structure in a Xilinx Virtex-7 FPGA device are provided and discussed in section 4. Finally, section 5 concludes the presentation.

## 2. Overview of the original transform architecture

The architecture presented in [18] consists of a scalable hardware structure for the computation of all the transforms adopted in the H.264/AVC and H.265/HEVC standards, i.e., the  $8 \times 8$  and  $4 \times 4$  DCTs and the  $4 \times 4$  and  $2 \times 2$  Hadamard transforms defined in H.264/AVC, as well as the  $16 \times 16$ ,  $8 \times 8$  and  $4 \times 4$  integer DCTs adopted by H.265/HEVC. To achieve such goal, this multi-transform processing structure implements the row-column decomposition strategy [3] and makes use of the four functional modules depicted in Fig. 1, i.e., the Transform Array (TA), the Transposition Switch (TS), the Input Buffer (IB) and the Control Unit (CU).

The core of this MTA consists of a two-dimensional (2-D) systolic TA, which is used to compute all the considered transforms. The array is composed of  $N \times N$  Processor Elements (PEs), where  $N$  is the size of the largest supported transform, which perform the same set of operations and share an identical architecture that is capable of supporting all the required calculations [18]. Within the TA, the data is processed in a wavefront manner, as it is illustrated in Fig. 2. The three represented data-sets correspond to the processing of two consecutive  $N \times N$  data blocks: two data-sets of residue values (or transform coefficients, when inverse transforms are being computed), depicted using a

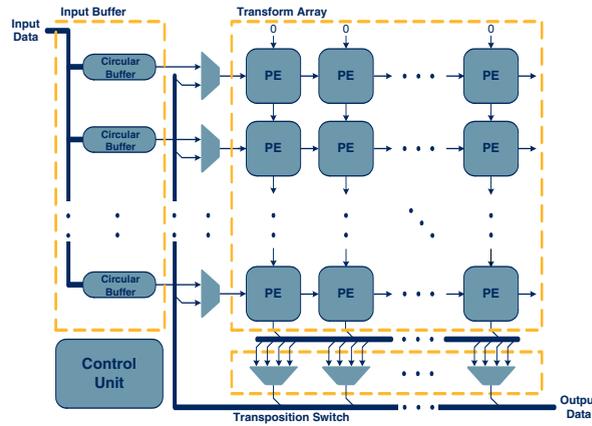


Fig. 1. Generic block diagram of the MTA presented in [18].

solid-line, and one data-set of intermediate values of the row-column decomposition, represented using a dashed-line. As it can be seen, the data is fed into the PE rows through the input buffers in the left column of the array. Then, it is processed by each PE and subsequently propagated in the horizontal and vertical directions to the neighbour PEs inside the array, advancing one PE level (in both directions) at each clock cycle. Conversely, the control signals for all the PEs enter the array through the top-left corner PE and are propagated to the other PEs (also in both directions) synchronously with the data propagation. This processing scheme allows to maximize the data processing rate, since on each clock cycle it is possible to start the computation of a different transform value in each row of the array (provided that the input buffers are not empty).

The TS is used to implement the row-column transposition of the data processed inside the TA. Unlike other transposition units that have been presented in the literature [7,9,13,14,16], this scalable hardware structure does not uses memory resources and is mostly composed by a set of  $(N - 1)$  multiplexers. This approach allows to obtain a fast and direct row-column transposition of the data with a quite reduced hardware cost.

The IB is used to feed the TA 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 coefficients, whenever an inverse transform is being computed. This scalable unit includes  $N$  quite simple circular buffers and allows to greatly enhance the performance of the multi-transform architecture. On the one hand, due to minimizing the delays when

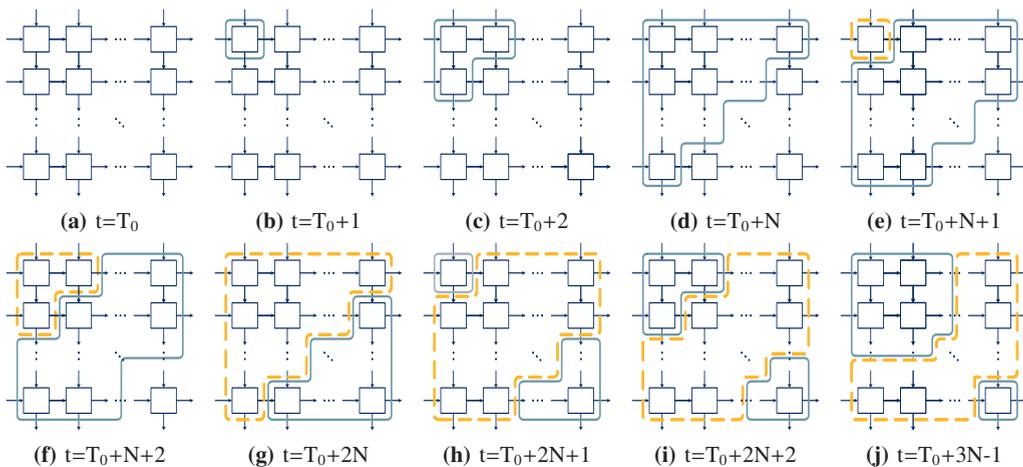


Fig. 2. Dataflow in an  $N \times N$  Transform Array for  $N \times N$  data blocks.

accessing the external data memories where such data is stored, by efficiently exploiting appropriate cache access patterns. On the other hand, because it provides a regular dataflow within the systolic array, thus avoiding stalls in the processing of the data. Consequently, by using this module the data values can be serially and smoothly transferred to the several rows of the array, therefore supporting the adopted streaming model.

Finally, the CU is responsible for controlling the TA, as well as the operation of both the IB and the TS. 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 processing structure, including the architecture of the PEs that are also used in the parallel MTA herein proposed, the interested reader is referred to [18].

### 3. Proposed parallel Multi-Transform Architecture

The MTA proposed in [18] presents several important advantages for the design of state-of-the-art video codecs, such as: *i)* offering high throughput and data processing rates, and thus enable the processing of high definition video in real-time; or *ii)* a modular and scalable hardware structure that can be easily configured to support several different video standards. However, such architecture also features a quite significant drawback for the implementation of multi-transform cores that are required to compute several different transforms with distinct sizes: *poor hardware efficiency*. This is the case of the designs targeting the H.264/AVC standard, which are required to compute not only the forward and the inverse  $4 \times 4$  DCTs, but also the  $4 \times 4$  and the  $2 \times 2$  Hadamard transforms [4]. Furthermore, it is also the case of hardware realizations compliant with the H.265/HEVC standard or the High Profiles of the H.264/AVC standard, which require the computation of larger transforms, e.g.,  $16 \times 16$  [5] and  $8 \times 8$  DCTs [4,5].

In these application domains, the implemented transform cores are typically based on a TA with  $N \times N$  PEs, where  $N$  corresponds to the size of the largest transform to be computed. For the computation of such transforms, this configuration provides the optimal throughput value of  $N$  samples per clock cycle, as well as an occupation rate of 100% for all the computational units [18]. However, such impressive performance and hardware efficiency values are greatly reduced for the computation of all the considered lower order transforms, since only some of the hardware resources available in the TA, the TS and the IB are used in such procedures. More specifically, in a transform core based on a TA with  $N \times N$  PEs, the computation of an  $\frac{N}{k} \times \frac{N}{k}$  transform ( $k = 2^i$  with  $i = 1, \dots, \log_2 N - 1$ ) makes uses of only  $\left(\frac{N}{k}\right)^2$  PEs in the TA and of only  $\frac{N}{k}$  multiplexers and circular buffers in the TS and IB, respectively. Consequently, the offered throughput is constrained to  $k$  samples per clock cycle, while the hardware efficiency is reduced to approximately  $\left(\frac{1}{k}\right)^2$ . For an  $8 \times 8$  TA ( $N = 8$ ) compliant with the High Profiles of the H.264/AVC standard [18], this results in the processing of either 4 or 2 transform coefficients per clock cycle for the computation of the mandatory  $4 \times 4$  ( $k = 2$ ) and  $2 \times 2$  ( $k = 4$ ) transforms, respectively. In addition, the corresponding hardware efficiency values are, approximately, 25% and 6.25% for the computation of such transforms.

The parallel MTA that is herein proposed consists of a greatly enhanced version of the architecture presented in [18], which exploits coarse-grain data-level parallelism techniques to overcome the previously mentioned limitations. Such parallel processing capability consists in allowing the simultaneous computation of several transforms within the datapath of the architecture. For example, with the proposed parallel MTA, a transform core composed of  $N \times N$  PEs is not only capable of efficiently computing an  $N \times N$  transform, but also  $k$  transforms of size  $\frac{N}{k} \times \frac{N}{k}$  simultaneously. This results in an increase of the architecture's hardware efficiency of about  $k$  times, regarding to the original architecture. For the transform core presented in [18], this corresponds to an improvement of its hardware efficiency of, approximately, 100% for the computation of the  $4 \times 4$  transforms and of about 400% for the computation of the  $2 \times 2$  transforms. This can be clearly seen in Fig. 3, which shows the dataflow for the processing, in parallel, of four different  $2 \times 2$  transforms ( $k = 4$ ) within a TA with  $8 \times 8$  PEs ( $N = 8$ ). The two data-sets depicted using a solid-line concern the prediction residue values (or the transform coefficients, when inverse transforms are considered), while the data-sets represented using a dashed-line consist of the intermediate values of the row-column decomposition.

As it can be easily concluded, this new processing mode allows to greatly improve the efficiency of the proposed parallel MTA regarding to the original design presented in [18]. On the one hand, because it not only allows to speedup the transform computation procedure by  $k$  times, but also to always achieve the optimal throughput value of  $N$  samples per clock cycle, as a consequence of  $k$  different transforms being computed simultaneously. On the other

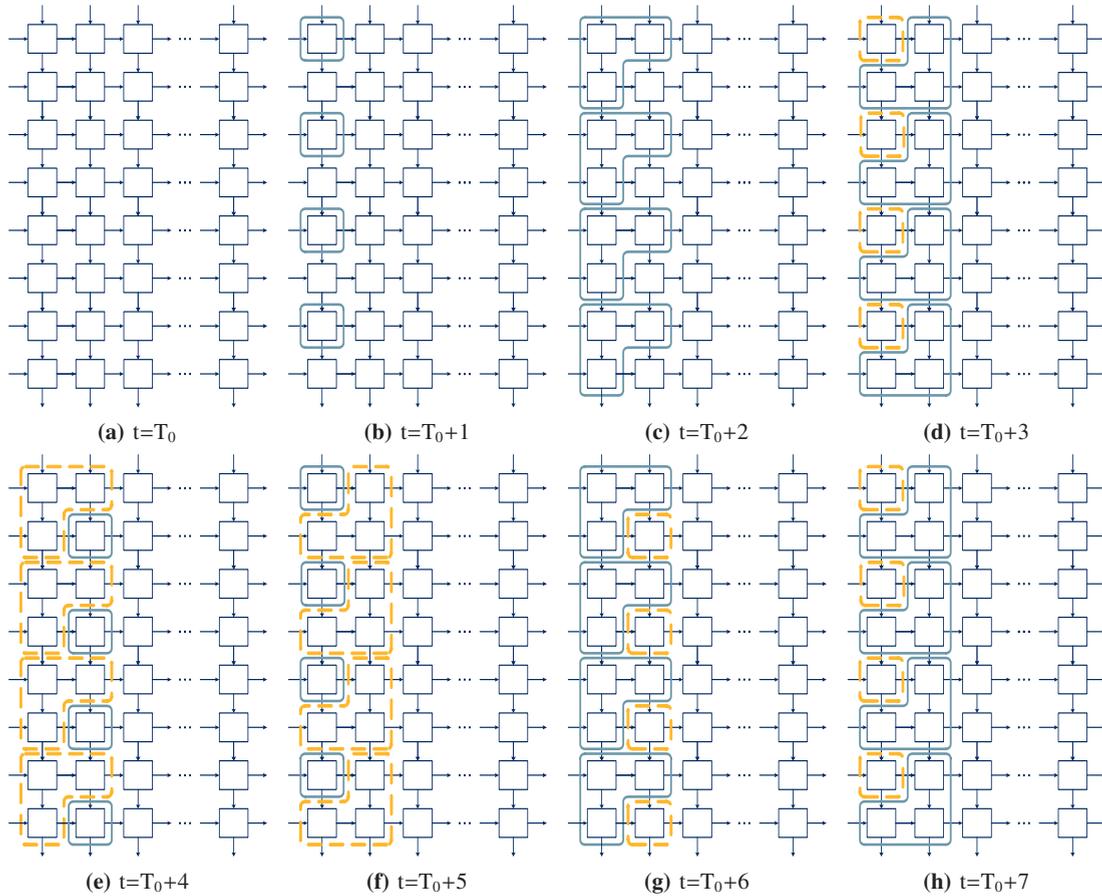


Fig. 3. Dataflow in an  $8 \times 8$  Transform Array for the simultaneous processing of four  $2 \times 2$  data blocks.

hand, due to enabling the use of an additional  $(k - 1) \times (\frac{N}{k} \times \frac{N}{k})$  PEs and of all the multiplexers and circular buffers available in the TS and IB, respectively, for the computation of the smaller  $\frac{N}{k} \times \frac{N}{k}$  transforms. Nevertheless, in order to obtain this improved coarse-grained data-level parallelism functionality, several modifications had to be made to the hardware structure of the original design. In Figure 4 it is presented the generic block diagram of a transform core based on the proposed parallel MTA, where such changes are represented in red color. Such transform core is composed of a TA with  $8 \times 8$  PEs ( $N = 8$ ), which is capable of computing either one  $8 \times 8$  transform, two  $4 \times 4$  transforms ( $k = 2$ ) or four  $2 \times 2$  transforms ( $k = 4$ ).

In what concerns the datapath of the proposed parallel architecture, the TA and the TS were redesigned as sets of multiple independent subunits, each one capable of fully processing an  $\frac{N}{k} \times \frac{N}{k}$  transform. As it can be seen in Figure 4, the TA subunits are composed of  $\frac{N}{k} \times \frac{N}{k}$  PEs with identical processing capabilities. These units can be either cascade interconnected to support the computation of the larger transforms, or operated independently to allow the computation of several smaller transforms in simultaneous within the TA. The TS subunits consist of a set of cascaded multiplexers that allow the transposition of the data under processing back into the TA, either as a whole  $N \times N$  block or in chunks of  $\frac{N}{k} \times \frac{N}{k}$  blocks, when  $k$  transforms are computed in parallel. Obviously, these extra multiplexers impose a penalty in the maximum clock frequency of the implemented parallel transform cores, since they slightly increase the critical path of the design. Nevertheless, it can be shown that the corresponding impact in the global performance of such hardware realizations is almost negligible (see section 4). This is mainly due to the fact that the critical path of the parallel MTA is mostly influenced by the delay imposed by its PEs, as a result of all the arithmetic circuits that they include and the pipelined processing scheme that characterizes this processing structure.

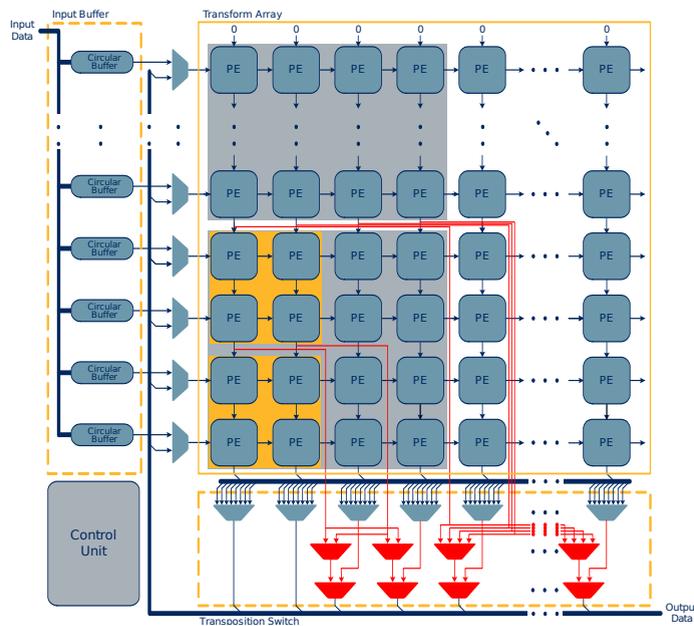


Fig. 4. Block diagram of the parallel MTA for the computation of  $8 \times 8$ ,  $4 \times 4$  and  $2 \times 2$  transforms.

Due to all the previously mentioned modifications, the output interface of the TA and the input interface of the TS were also redesigned to accommodate the new data buses traversing the TA and entering the TS, as it is shown in Figure 4. In addition, the CU was also adjusted, in order to be able to support the new parallel processing mode. As a result of the modular and highly configurable design of the original architecture, such changes mostly consisted in the definition of a broader set of threshold values for the involved auxiliary control circuits.

#### 4. Experimental Evaluation

To evaluate the advantages offered by the proposed parallel MTA, a proof-of-concept transform core targeting the High Profiles of the H.264/AVC standard was designed and implemented in a Xilinx Virtex-7 XC7VX485T FPGA device [19]. This processing structure is based on a TA with  $8 \times 8$  PEs, where the PEs are the same as in [18], and allows the computation of the forward and inverse  $8 \times 8$  and  $4 \times 4$  DCTs, as well as of the  $4 \times 4$  and  $2 \times 2$  Hadamard transforms.

The considered hardware realization was conducted by using the Xilinx ISE 13.2i development tool chain [20] and a fully parameterizable IEEE-VHDL description of the proposed parallel MTA. Such description adopts a strict modular approach, by using independent and self-contained functional blocks. Moreover, it makes an extensive use of generic configuration and parameterization inputs, in order to support both the multi-transform functionality and the parallelism and scalability characteristics of the architecture. Table 1 presents the obtained implementation results, by considering an optimization setup targeting hardware realizations optimized for performance.

As it can be seen, the devised proof-of-concept parallel transform core is capable of operating with a clock frequency of 279.8 MHz. This value is quite similar to the one of the original MTA presented in [18] (280.6 MHz), which allows to conclude that the implementation of the considered coarse-grain parallelism technique has a negligible impact in the clock frequency of the implemented transform cores. This was already expected, since the maximum clock frequency of the architecture is mostly limited by the critical path of the PEs it includes, rather than by the critical path of its TS [18]. Consequently, the considered proof-of-concept hardware realization is able to compute up to 17.91 Giga Operations Per Second (GOPS) by using a clock frequency of 279.8 MHz, while offering a maximum throughput of 8 samples per clock cycle. Such values correspond to the computation of the  $8 \times 8$  transforms, where

Table 1. Implementation results of the proof-of-concept transform core in a Xilinx Virtex-7 XC7VX485T FPGA.

Processing structure	Proposed parallel MTA			Original architecture [18]		
	Registers	LUTs	Max. F.	Registers	LUTs	Max. F.
☐ Parallel MTA	6420	22972	279.8 MHz	6420	21452	280.6 MHz
☒ Input buffer	1681	1254	344.8 MHz	1681	1254	344.8 MHz
☐ Transform array with $8 \times 8$ PEs	1501	15857	294.7 MHz	1501	15727	296.5 MHz
☒ PE	70	254	298.5 MHz	70	254	298.5 MHz
☒ Transposition switch	1462	3047	335.7 MHz	1462	2161	360.0 MHz
☒ Control unit	97	86	365.2 MHz	97	77	366.4 MHz

each one of the 64 PEs that compose the TA computes one different transform operation on each clock cycle. For the computation of the smaller  $4 \times 4$  and  $2 \times 2$  transforms the throughput is also of 8 samples per clock cycle, since 2 different transforms (or 4 transforms in the last case) are simultaneously computed within the datapath of the proposed parallel architecture. However, only half of the PEs (a quarter of the PEs for the  $2 \times 2$  transforms) are used in such computations, which reduces the computational rate of the transform core to 8.95 GOPS and 4.48 GOPS, respectively. Nevertheless, by using the proposed parallel MTA it is possible to compute one  $8 \times 8$  transform in 16 clock cycles, two  $4 \times 4$  transform in 8 clock cycles or four  $2 \times 2$  transform in 4 clock cycles, where the period of the clock signal is 3.57 ns. This results in speedup values of about 1, 2 and 4 times for the computation of the  $8 \times 8$ ,  $4 \times 4$  and  $2 \times 2$  transforms regarding to the original MTA, whose clock cycle period is 3.56 ns. Note that such structure is able to compute only one  $8 \times 8$ ,  $4 \times 4$  and  $2 \times 2$  transform at a time, which also requires 16, 8 and 4 clock cycles, respectively.

The data presented in Table 1 also shows that the hardware cost of the devised proof-of-concept parallel transform core is quite small and also very similar to the one of the original multi-transform design presented in [18]. More specifically, the implemented transform core requires less than 30% of the total hardware resources available in the considered medium-sized FPGA. The minor differences that can be observed mostly concern to the resources required to implement the new TS, as a result of all the extra multiplexers that it includes to support the implemented coarse-grain parallelism technique. Although this causes an increase of about 40% in the hardware cost of the new TS, it only augments the total hardware cost of the whole transform core in about 7%. This results from the fact that the majority of the consumed hardware resources (about 70%) are used in the implementation of the  $8 \times 8$  PEs TA. Conversely, the hardware cost of the TA and of the CU are almost identical to the ones presented in [18], due to the quite negligible impact of the modifications that were implemented in their hardware structures. Note that such modifications mostly consisted in *i*) the addition of some extra buses in the TA, in order to feed the TS with the output data of all the smaller transforms; and *ii*) the definition of earlier stop points in the control algorithm implemented by the CU.

From the previous discussion it can be concluded that the coarse-grain data-level parallel processing capabilities of the implemented proof-of-concept transform core improve its computational performance in about 2 times for the computation of the  $4 \times 4$  transforms, and in about 4 times for the computation of the  $2 \times 2$  transforms. This allows it to compute the whole set of H.264/AVC transforms in real-time for video sequences in the 8k Ultra High Definition Television (UHDTV) format ( $7680 \times 4320 @ 30$  fps). In addition, it can be observed that the hardware efficiency of the implemented parallel multi-transform core is also greatly enhanced, i.e., about 50% for the computation of the  $4 \times 4$  transforms and about 25% for the computation of the  $2 \times 2$  transforms. Such improvements are the result of the application of the considered coarse-grain data-level parallelism technique, which imposes a marginal increase of about 7% in the global hardware cost of the parallel architecture.

## 5. Conclusions

In this paper, it is presented a high performance and hardware efficient parallel Multi-Transform Architecture (MTA) for the computation of the 2-D integer transforms adopted by the state-of-the-art video standards, such as H.264/AVC or H.265/HEVC. This processing structure is based on a high throughput 2-D systolic array, which can be easily configured at run-time to compute multiple transforms with distinct kernels and sizes. To improve the

hardware efficiency and speedup the transform computation procedure, the datapath of the proposed parallel MTA was especially designed to support the computation of several transforms in parallel. The benefits offered by the proposed MTA for the design of modern multi-standard video codecs were experimentally assessed by using a Xilinx Virtex-7 FPGA device. The obtained implementation results demonstrate that the devised parallel architecture is not only capable of achieving real-time operation for video sequences in the 8k UHDTV format ( $7680 \times 4320 @ 30$  fps), but also of improving the hardware efficiency by (at least) 100%, regarding to the original multi-transform design.

## Acknowledgements

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under the project "HELIX: Heterogeneous e 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.

## References

- [1] Z.-N. Li, M. S. Drew, *Fundamentals of Multimedia*, 1st Edition, Prentice Hall, 2003.
- [2] J. Ohm, G. Sullivan, H. Schwarz, T. Keng, T. Wiegand, Comparison of the coding efficiency of video coding standards – including high efficiency video coding (HEVC), *IEEE Transactions on Circuits and Systems for Video Technology* 22 (12) (2012) 1669–1684.
- [3] K. Rao, P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, 1990.
- [4] T. Wiegand, G. Sullivan, G. Bjntegaard, A. Luthra, Overview of the H.264/AVC video coding standard, *IEEE Transactions on Circuits and Systems for Video Technology* 13 (7) (2003) 560–576.
- [5] G. J. Sullivan, J. Ohm, H. Woo-Jin, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, *IEEE Transactions on Circuits and Systems for Video Technology* 22 (12) (2012) 1649–1668.
- [6] [Press release – LG display brings innovation in smart mobile panels to SID 2011](http://lgdnewsroom.com/press_releases/601), LG Display Co, Ltd. (May 2011).  
URL [http://lgdnewsroom.com/press\\_releases/601](http://lgdnewsroom.com/press_releases/601)
- [7] L. Ling-Zhi, Q. Lin, R. Meng-Tian, J. Li, A 2-D forward/inverse integer transform processor of H.264 based on highly-parallel architecture, in: *4th IEEE International Workshop on System-on-Chip for Real-Time Applications*, 2004, pp. 158–161.
- [8] C.-P. Fan, Cost-effective hardware sharing architectures of fast 8x8 and 4x4 integer transforms for H.264/AVC, in: *2006 IEEE Asia Pacific Conference on Circuits and Systems*, 2006, pp. 776–779.
- [9] C. Jiang, N. Yu, M. Gu, A novel VLSI architecture of 8x8 integer DCT based on H.264/AVC FRext, in: *3rd International Symposium on Knowledge Acquisition and Modeling*, 2010, pp. 59–62.
- [10] T. Do, T. Le, High throughput area-efficient SoC-based forward/inverse integer transforms for H.264/AVC, in: *2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 4113–4116.
- [11] C. Wei, H. Hui, L. Jinmei, T. Jiarong, M. Hao, A high-performance reconfigurable 2-D transform architecture for H.264, in: *15th IEEE International Conference on Electronics, Circuits and Systems*, 2008, pp. 606–609.
- [12] C.-C. Lo, S.-T. Tsai, M.-D. Shieh, Reconfigurable architecture for entropy decoding and inverse transform in H.264, *IEEE Transactions on Consumer Electronics* 56 (3) (2010) 1670–1676.
- [13] W. Hwangbo, C.-M. Kyung, A multitransform architecture for H.264/AVC high-profile coders, *IEEE Transactions on Multimedia* 12 (3) (2010) 157–167.
- [14] M. Gu, N. Yu, C. Jiang, W. Lu, Hardware prototyping for various transforms in H.264 high profile, *Journal of Information and Computational Science* 8 (1) (2011) 119–128.
- [15] M. Martuza, K. Wahid, A cost effective implementation of 8x8 transform of HEVC from H.264/AVC, in: *25th IEEE Canadian Conference on Electrical Computer Engineering*, 2012, pp. 1–4.
- [16] S. Shen, W. Shen, Y. Fan, X. Zeng, A unified 4/8/16/32-point integer IDCT architecture for multiple video coding standards, in: *2012 IEEE International Conference on Multimedia and Expo*, 2012, pp. 788–793.
- [17] T. Ho, T. Le, K. Vu, S. Mochizuki, K. Iwata, K. Matsumoto, H. Ueda, A 768 megapixels/sec inverse transform with hybrid architecture for multi-standard decoder, in: *9th IEEE International Conference on ASIC*, 2011, pp. 71–74.
- [18] T. Dias, N. Roma, L. Sousa, High performance multi-standard architecture for DCT computation in H.264/AVC high profile and HEVC Codecs, in: *Conference on Design & Architectures for Signal and Image Processing*, 2013, pp. 14–21.
- [19] Xilinx, Inc., *7 Series FPGAs Overview Data Sheet* (Jul. 2013).
- [20] Xilinx, Inc., *Xilinx ISE In-Depth Tutorial* (Mar. 2011).