Area Efficient DST Architectures for HEVC

Area Efficient DST Architectures for HEVC

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

***Abstract*—This work analyses the actual throughput of the Discrete Sine Transform (DST) stage in a realistic HEVC encoder, which executes the rate-distortion optimization algorithm to achieve high compression quality. Then, a low complexity DST factorization, where all the integer multiplications are substituted with add-and-shift operations, is exploited to design an efficient 1D-DST core. The proposed 1D-DST core is employed to derive two area efficient architectures, namely Folded and Full-parallel, for computing the 4×4 2D-DST in HEVC. Finally, the proposed 2D-DST architectures are synthesized on a 90-nm standard cell technology to support the actual target throughput required to encode 4K UHD @30fps video sequences, showing better area efficiency with respect to existing DST architectures for HEVC.**

I. Introduction

The latest High Efficiency Video Coding (HEVC) standard is an hybrid block-based video compression scheme, based on motion estimation and transform coding [1]. Its aim is to double the rate-distortion performance compared to its predecessor Advanced Video Coding (AVC) standard [2], by involving an increased complexity [3]. Concerning the transform stage, HEVC adopts both Discrete Cosine Transform (DCT) and Discrete Sine Transform (DST) of different lengths [4]. While the DCT has been largely exploited in previous standards to code the inter-predicted blocks, HEVC specifies an integer DST transform for intra-predicted blocks of size 4×4. The benefits of the application of the DST on the residual signal, coming from the directional intra prediction, have been already shown in the literature [5], [6].

Moreover, the issue of real-time video compression, especially for ultra high definition (UHD) content (e.g. 4K UHD @30fps), has posed severe throughput requirements on the transform operation at the encoder side, because of the rate-distortion optimization (RDO) algorithm, as shown in [7]. Therefore, hardware accelerators for computing transforms have been designed in order to meet real-time requirements. However, while several architectures have been recently proposed for the integer DCT specified in HEVC [8]–[10], only few works address the design of hardware architectures to efficiently compute the integer DST. Edirisuriya *et al.* [11] exploited the relationships between the DCT of different sizes and the DST to design a multiplication-free architecture, which is able to perform both the DCT and the DST on a block of size up to 16×16 samples. However, the reconfigurability of the system to support different transforms is paid in terms of large area occupation due to an high number of hardware resources, which are not fully utilized. Therefore, Nam *et al.* [12] proposed an hardware architecture to compute the integer 4×4 DST in HEVC, which combines butterfly operations with 4-2 compressors in order to simplify the circuit and to improve the area efficiency.

Stemming from these observations, the first contribution of this work is the analysis of the actual throughput requirements for the design of a DST architecture taking into account the RDO algorithm of a realistic HEVC encoder. Then, the second contribution is to show a novel architecture to compute the integer 4×4 1D-DST. The proposed architecture exploits the factorization in [13], which reduces the number of hardware resources with respect to the matrix-vector multiplication (MVM). Finally, the 1D-DST core is used to design two 2D-DST HEVC compliant architectures, which outperform existing DST architectures.

The paper is organized as follows. Section II reports the analysis of the actual throughput requirements for the DST design taking into account the RDO algorithm of a realistic HEVC encoder. Then, the proposed low-complexity 1D-DST architecture is described in Section III, while Section IV shows two architectures for the 2D-DST computation in HEVC. Finally, implementation results are presented in Section V, while Section VI concludes the paper.

II. Actual Throughput Analysis

In order to design a DST module for real-time HEVC encoding, it is important to define the throughput of each processing block. This can be calculated by taking into account the resolution and frame rate of the video sequences to be encoded. However, this approach does not take into account the overhead of computations introduced by the RDO process, which is used to choose the set of coding modes that assures the best quality-compression trade-off. Specifically, in HEVC the encoder has to select one among 35 different intra prediction modes for each intra-predicted block. To perform this operation, the encoder evaluates the sum of squared differences (SSD) cost function between the original block and the reconstructed block after transform and quantization. For these reasons, the DST is applied more than once for each block. Although software profiling [3] provides information about the computational burden of encoding tasks, it is not able to catch the actual throughput requirement. On the other hand, throughput can be calculated by counting the number of transform operations that are computed during the whole encoding process [7].

TABLE I
TRANSFORM COMPLEXITY INDEX (C_I) OF 4K UHD SEQUENCES FOR DIFFERENT QPS.

Video Sequence	QP 22	QP 27	QP 32	QP 37
Bund Nightscape	3.07	2.89	2.80	2.74
Campfire Party	3.46	3.06	2.82	2.72
Construction Field	3.64	3.15	2.91	2.77
Fountains	3.36	3.07	2.87	2.76
Library	3.09	2.85	2.76	2.71
Marathon	**3.89**	3.24	2.96	2.82
Residential Building	3.46	3.14	2.95	2.82
Runners	3.86	3.51	3.22	3.00
Rush Hour	3.48	3.11	2.89	2.77
Scarf	3.06	2.76	2.59	2.44
Tall Buildings	3.52	3.26	3.05	2.89
Traffic and Building	3.24	3.01	2.87	2.77
Traffic Flow	3.50	2.97	2.81	2.73
Tree Shade	3.57	3.27	3.05	2.88
Wood	3.66	3.35	3.14	2.97

In this work, the focus is on the design of the DST module in a specific scenario, which is the HEVC intra encoding of 4K UHD @30fps video sequences. Despite an exhaustive test of all coding modes would be optimal for RDO performance, it would be not practical in realistic applications. For this reason the actual throughput analysis has been carried out by using the x265 encoder [14], which has been configured to encode the sequences using only intra frames. All the video sequences of the SJTU dataset [15] have been coded using four different quantization parameters (QPs), namely 22, 27, 32 and 37. For each case the transform complexity has been calculated as in [7], namely as the ratio between the actual throughput ($T_A = 16 \cdot N_{DST} \cdot F_s/N_f$), which considers the RDO algorithm, and the reference throughput ($T_H = W \cdot H \cdot S_c \cdot F_s$), which is computed assuming that each pixel is transformed only once. The Complexity Index (C_I) is calculated by only taking into account the 4×4 DST contribution:

$$C_I = \frac{T_A}{T_H} = \frac{16 \cdot N_{DST}}{W \cdot H \cdot S_c \cdot N_f}, \tag{1}$$

where N_{DST} is the count of 4×4 DST computed by the x265 encoder, and W, H, S_c, F_s and N_f are the width, the height, the chrominance sub-sampling factor, the frame-rate and the number of frames in the video sequence respectively. The values for the 4K UHD video sequences of the SJTU dataset are $W = 3840$, $H = 2160$, $S_c = 1.5$ (4:2:0 format), $F_s = 30$ and $N_f = 300$.

Table I reports the transform complexity index of each sequence for different QPs. As it can be observed, the transform complexity varies across different QPs and it is also dependent on the video content. Specifically, it is higher in such sequences which show high motion and spatial details, since small blocks are mainly used to describe non-uniform areas. Because of the large variety of encoding scenarios, the worst case has been considered in this work ($C_I = 3.89$). Therefore, to have some margin $C_I = 4$ has been chosen as upper bound, which means that the actual target throughput is $T_A = T_H \cdot C_I = 1.493$ Gsps.

III. 1D-DST ARCHITECTURE

Let $x = (x_0, \ldots, x_3)$ and $X = (X_0, \ldots, X_3)$ be the input samples and the output transform coefficients, the HEVC standard specifies the core 4-point integer DST as $X = S \cdot x$, where S is the 1D transform matrix:

$$S = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}, \tag{2}$$

which is derived by upscaling and rounding the coefficients of the 4-point type-VII DST matrix, as shown in [1].

The proposed 1D-DST architecture is based on the factorization suggested in [13], which derives the N-order type-VII DST from the imaginary part of the Discrete Fourier Transform on $2N + 1$ points and reduces the computational complexity from 16 multiplications and 12 additions to only 5 multiplications and 11 additions. According to [13], the 4-point HEVC-compliant DST matrix S can be decomposed by means of three sparse matrices as:

$$S = M_3 \cdot M_2 \cdot M_1, \tag{3}$$

where

$$M_1 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -1 \end{bmatrix}, M_2 = \begin{bmatrix} 84 & 0 & 0 & 0 & 0 \\ 0 & 74 & 0 & 0 & 0 \\ 0 & 0 & 55 & 0 & 0 \\ 0 & 0 & 0 & 29 & 0 \\ 0 & 0 & 0 & 0 & 74 \end{bmatrix},$$

$$M_3 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & -1 & 0 \end{bmatrix}. \tag{4}$$

The proposed 1D-DST architecture is depicted in Fig. 1, where the three main computational blocks, corresponding to matrices M_1, M_2 and M_3, are highlighted. First, input samples x are combined as described by M_1 by using simple additions and subtractions to generate five intermediate results. Then, intermediate results are multiplied by constants implementing M_2. Resorting to the RAG-n technique [16], all the multipliers have been simplified to add-and-shift blocks, thus loosing flexibility but saving hardware costs. Table II details the arithmetic complexity and the computational depth (expressed in number of cascaded adders/subtracters) of each integer coefficient. It is worth noting that the shifts do not contribute to the overall hardware complexity because they are implemented by simple wiring. Finally, the output coefficients X are calculated by the last stage, which implements M_3, by means of adders and subtracters.

The internal parallelism has been chosen according to the HEVC specifications [4]. Since the proposed 1D-DST architecture serves as the core processing block for both the Folded and the Full-parallel architectures, shown in Section IV, the four input samples are represented on 16 bits each, whereas the output samples are on 24 bits. Intermediate results after stage M_1 and M_2 are represented on 18 and 24 bits respectively in order to avoid overflow.

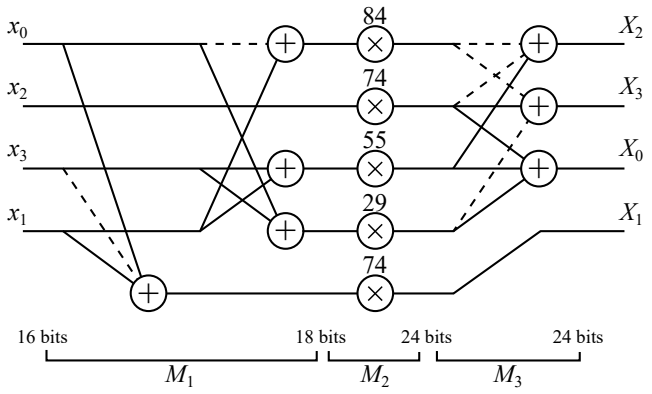


Fig. 1. 1D-DST architecture. Dashed lines represent inputs to be subtracted.

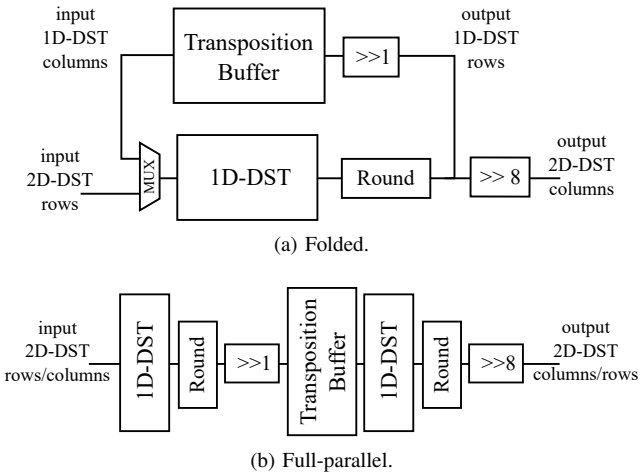
TABLE II
RAG-N REPRESENTATION OF M_2 COEFFICIENTS.

Integer Coefficient	Add/Sub	Shifts	Depth
84	2	3	2
74	2	3	2
55	2	2	2
29	2	2	2

IV. 2D-DST ARCHITECTURES

Thanks to the separability property, the proposed HEVC-compliant 2D-DST architectures rely on the 1D-DST core with no internal pipeline, which is used to perform the transform operation first on the rows and then on the columns of the input block of samples. Two 2D-DST architectures, namely Folded and Full-parallel, have been designed, as in [8] for the 2D-DCT. The former one is depicted in Fig. 2a and uses one 1D-DST module to perform both the row-wise and the column-wise transforms and a transposition buffer to store the intermediate results. The processing rate of this architecture is 2 samples/cycle. On the other hand, the Full-parallel architecture in Fig. 2b is composed of two 1D-DST modules, that allow to achieve double throughput (4 samples/cycle). The transposition buffers have been implemented by means of a 4×4 array of 16-bit registers plus additional logic to read and write either rows or columns, as in Fig. 5b and 6b of [8]. All the control signals for the Folded and the Full-parallel architectures are provided by two different control units, composed of a state machine and a counter. In the Folded structure, the multiplexer (MUX) selects either the input rows or the intermediate columns in a time-multiplexed manner.

Moreover, rounding and scaling operations are required to make the 2D architectures compliant with the HEVC standard [4]. Both architectures are fed with 9-bit input samples, which are extended to 16 bits for row-wise computation. Intermediate results, stored in the transposition buffer, and final results are represented on 16 bits as well. The rounding operation is performed by adding $2^{(B-1)}$, whereas the scaling operation is a right shift of B positions. The value of B is equal to 1 for row-wise computation and to 8 for column-wise computation respectively, as detailed in [4].



(a) Folded.

(b) Full-parallel.

Fig. 2. Proposed 2D-DST architectures.

TABLE III
SYNTHESIS RESULTS OF THE MVM BASED IMPLEMENTATION AND THE PROPOSED 1D-DST ARCHITECTURE.

Design	MVM based	Proposed
Technology	90-nm	90-nm
Delay	0.57 ns	0.81 ns
Gate Count	7615	4311
Power	9.30 mW	3.93 mW
Gates-Delay Product	4340	3491

V. IMPLEMENTATION RESULTS

The proposed architectures have been described in VHDL, verified and synthesized using a 90-nm standard cell library with Synopsys Design Compiler. It is worth noting that the actual implementation of the multiple-operand adders and subtracters, required in the 1D-DST core, has been left to the synthesis tool. This strategy permits to exploit Design Compiler capability to optimize the overall architecture for area and speed by selecting the best arithmetic implementation.

Implementation results of the 1D-DST architecture are reported in Table III, where the delay, the gate count, the power consumption and the gates-delay product, obtained when synthesizing to achieve the maximum operating frequency, are shown. The proposed architecture is compared with the direct implementation of $X = S \cdot x$ as a MVM, where sixteen multipliers and four 4-input adders work concurrently to compute X. It has been observed that the MVM based implementation achieves the minimum delay. Indeed, the critical path is composed of one constant multiplier and one multi-operand adder, implementing the sum-of-products. On the other hand, the proposed architecture occupies a small area and features low power consumption. Moreover, it requires about 20% less gates-delay product than the MVM based one.

As shown in Section II, the actual throughput required for the HEVC intra encoding of 4K UHD @30fps video sequences, considering the RDO algorithm implemented in the x265 model, is 1.493 Gsps. It is worth noting that a throughput of 1.496 Gsps (which is high enough to satisfy the constraint) can be achieved by setting the clock frequency of

TABLE IV
SYNTHESIS RESULTS OF 2D-DST ARCHITECTURES.

Design	Folded	Full-parallel	Edirisuriya *et al.* [11]	Nan *et al.* [12]
Technology	90-nm	90-nm	45-nm	0.13-μm
Operating Frequency	748 MHz	374 MHz	900 MHz	300 MHz
Transform Size	4×4	4×4	4×4/8×8/16×16	4×4
Processing Rate	2 samples/cycle	4 samples/cycle	4/8/16 samples/cycle	4 samples/cycle
Throughput	1.496 Gsps	1.496 Gsps	3.600 Gsps	1.200 Gsps
Supported Video Format	4K UHD @30fps	4K UHD @30fps	4K UHD @72fps	4K UHD @25fps
Gate Count	5910	7167	244360	7000
Throughput/Gates	253130	208734	16045	171428
Power Consumption	7.44 mW	3.19 mW	-	-
Energy-per-Sample	4.97 pJ	2.13 pJ	-	-

the Folded and Full-parallel 2D-DST architectures to 748 MHz and 374 MHz respectively. The first three columns of Table IV summarize the synthesis results achieved for the Folded and Full-parallel architectures, in terms of technology, operating frequency, supported transform sizes, processing rate, throughput, gate count, throughput-over-gates ratio, power consumption and energy-per-sample. Since the Folded structure uses only one 1D-DST module, it saves about 17% gate count with respect to the Full-parallel implementation. On the other hand, it requires double clock frequency to provide the same throughput, thus leading to higher power consumption.

The last two columns of Table IV report the implementation details for the DST architectures proposed in [11], [12] as well. It is worth noting that the values for [11] refer to the implementation with input word length equal to 8 bits and the throughput has been calculated considering 4×4 DST blocks. Moreover, the solution in [11] undergoes a large area overhead due to the reconfigurability to support multiple-size transforms, which are not required for HEVC applications. Finally, the proposed architectures outperform the work in [12] both in terms of performance and area efficiency, showing 24% larger throughput and higher throughput-over-gates ratio. To satisfy the throughput of [12], the Folded and the Full-parallel architectures require only 5632 and 6514 gates respectively.

VI. CONCLUSION

In this work, an analysis of the throughput of the DST transform stage for a realistic HEVC encoder performing the RDO algorithm has been presented. Then, an hardware architecture for the 4-point 1D-DST, which exploits the factorization shown in [13] to reduce the number of adders and multipliers, has been proposed. Finally, two 2D-DST architectures for HEVC encoding, namely Folded and Full-parallel, have been described and implemented. Synthesis results show that the proposed architectures achieve the target actual throughput, showing better area efficiency than state-of-art implementations.

ACKNOWLEDGMENT

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

REFERENCES

[1] G. J. Sullivan, J. R. 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. R. Ohm, G. J. 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. 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.

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

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

[7] M. Masera, L. Re Fiorentin, M. Martina, G. Masera, and E. Masala, "Optimizing the Transform Complexity-Quality Tradeoff for Hardware-Accelerated HEVC Video Coding," in *Proc. 2015 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, Sept 2015, pp. 1–6.

[8] 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.

[9] M. Masera, M. Martina, and G. Masera, "Adaptive Approximated DCT Architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[10] Y. H. Chen and C. Y. Liu, "Area-efficient video transform for HEVC applications," *Electronics Letters*, vol. 51, no. 14, pp. 1065–1067, 2015.

[11] A. Edirisuriya, A. Madanayake, R. J. Cintra, and F. M. Bayer, "A Multiplication-free Digital Architecture for 16×16 2-D DCT/DST Transform for HEVC," in *Proc. 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, Nov 2012, pp. 1–5.

[12] J. Nan, N. Yu, W. Lu, and D. Wang, "A DST Hardware Structure of HEVC," in *Proc. 2015 2nd International Conference on Information Science and Control Engineering*, April 2015, pp. 546–549.

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

[14] MulticoreWare, *x265 HEVC Encoder*. [Online]. Available: https://bitbucket.org/multicoreware/x265/wiki/Home

[15] L. Song, X. Tang, W. Zhang, X. Yang, and P. Xia, "The SJTU 4K video sequence dataset," in *Proc. 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, July 2013, pp. 34–35.

[16] 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.