



# A High-Performance with Low-Resource Utility FPGA Implementation of Variable Size HEVC 2D-DCT Transform

Ying Zhang<sup>1</sup>(✉), Gen Li<sup>2</sup>(✉), and Lei Wang<sup>1</sup>(✉)

<sup>1</sup> Lab 673-2, Institute of Computer, National University of Defense Technology,  
Changsha 410073, China

{zhangying, wanglei}@nudt.edu.cn

<sup>2</sup> Genetalks Biotech. Co., Ltd, Changsha 410073, China  
gen.li@genetalks.com

**Abstract.** High Efficiency Video Coding (HEVC) is a new international video compression standard offering much better compression efficiency than previous video compression standards at the expense of much higher computational complexity. This paper presents a design of two-dimensional (2D) discrete cosine transform (DCT) hardware architecture dedicated for High Efficiency Video Coding (HEVC) in field programmable gate array (FPGA) platforms. The proposed methodology efficiently proceeds 2D-DCT computation to fit internal components and characteristics of FPGA resources. This architecture supports variable size of DCT computation, including  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ , and  $32 \times 32$ , and has been implemented in Verilog and synthesized in various FPGA platforms. Compared with existing related works, our proposed architecture demonstrates significant advantages in the performance improvement with low FPGA resource utility, which are very important for the whole FPGA solution for whole HEVC codec.

**Keywords:** H.265/HEVC · Two-dimensional discrete cosine transform (2D-DCT) · FPGA · Hardware

## 1 Introduction

Rapid advances in consumer electronics have resulted in a variety of video coding applications, such as ultra-high definition (UHD) 4 K/8 K TV [1] or unmanned aerial vehicle (UAV) reconnaissance and surveillance [2, 3], which demands aggressive video compression requirement. ITU and ISO standardization organizations are jointly developing a new international video compression standard H.265/HEVC, which has great potential to improve video compression efficiency by around 50%, while retaining the same video quality as H.264 [5, 6]. As a result, HEVC has been viewed as one of the most promising standards to overcome these challenges [7, 8]. However, High Efficiency Video Coding (HEVC) video compression standard at the expense of much more computational complexity [4–9]. HEVC uses Discrete Cosine Transform (DCT)/Inverse Discrete Cosine

Transform (IDCT). In addition, it uses Discrete Sine Transform (DST)/Inverse Discrete Sine Transform (IDST) for  $4 \times 4$  intra prediction in certain cases. Additionally, HEVC supports more block sizes of DCT and IDCT, including  $32 \times 32$ ,  $16 \times 16$ ,  $8 \times 8$ , and  $4 \times 4$  than H.264 which supports two smaller block sizes (i.e.  $4 \times 4$  and  $8 \times 8$ ). Large DCT and DST have high computational complexity, and they are heavily used in an HEVC encoder [10]. DCT and DST operations account for 11% of the computational complexity of an HEVC video encoder. They account for 25% of the computational complexity of an all intra HEVC video encoder.

Nowadays, computational resources in FPGA makes the study of HEVC FPGA implementation is gaining more and more attention. An efficient 2-D DCT FPGA architecture, used for the whole HEVC FPGA solution should have the following features: supporting all possible sizes in HEVC transform; using as little as possible FPGA implementation cost, i.e. the FPGA resources; gaining high-performance as possible.

In this paper, an FPGA implementation of HEVC 2-D DCT transform is proposed. Our FPGA architecture focuses on the above features. Compared with existing 2D-DCT FPGA architecture, our design can compute all block sizes in HEVC with the same FPGA design, which means the FPGA resources for butterfly PEs can deal with the block sizes of  $32 \times 32$ ,  $16 \times 16$ ,  $8 \times 8$  and  $4 \times 4$ , thus greatly reducing the FPGA implementation cost. Our butterfly PEs, which compute one element for the block size of  $32 \times 32$  per cycle, can compute 2 elements for the block size of  $16 \times 16$  per cycle, 4 elements for the block size of  $8 \times 8$  and 8 elements for the block size of  $4 \times 4$  per cycle, thus gaining high computation efficiency.

The rest of the paper is organized as follows. In Sect. 2, HEVC transform algorithms and related FPGA solutions for 2-D DCT are explained. In Sect. 3, the proposed element computation of HEVC DCT is explained. In Sect. 4, FPGA architecture for HEVC DCT is described. The implementation results are given in Sect. 5. Finally, Sect. 6 presents the conclusion.

## 2 Introduction

### 2.1 HEVC Transform Algorithms

Formula (1) shows the basic function for HEVC 1-D DCT transformation for an  $N \times N$  block, where  $i, j = 0, \dots, N-1$ . HEVC uses  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$  and  $32 \times 32$  TU sizes for DCT [11].

$$T_{i,j} = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right), \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases} \quad (1)$$

HEVC performs 2D transform operation by applying 1D transforms in vertical and horizontal directions. The coefficients in HEVC. 1D transform matrices are derived from DCT basis functions. However, integer coefficients are used for simplicity. HEVC 1D DCT coefficients for  $4 \times 4$  TU size are shown in Fig. 1. The coefficients in the same even row are symmetrical, while those in the odd row are odd-symmetrical, i.e. with the same values but different signals. Thus, the butterfly computation is used to compute every element in the DCT transformation.

$$DCT_{4 \times 4} = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix}$$

**Fig. 1.** HEVC 1D DCT coefficients for  $4 \times 4$  TU size

## 2.2 Related FPGA Architectures

In order to satisfy real-time and high-efficiency coding in these emerging video applications, a few design methodologies and circuit architectures have been developed [11, 18, 19, 20-24]. Usually, they can be partitioned 3 kinds. [12-15] can only compute one or two TU sizes of DCT transformation. The other TU sizes cannot be processed in these FPGA solutions. These architectures are meaningful for research but cannot be used in the really whole HEVC FPGA system solutions. [16, 17] support all block sizes of HEVC, but they build one different FPGA logics. Thus, while the FPGA logics for one block size work, the other three parts of FPGA resources idle. In [18], the FPGA architecture can support all HEVC block sizes, but all PEs computing one element per cycle for  $32 \times 32$  block size can only compute one element for the other smaller sizes. There are also some other FPGA designs with their own features, such as comparing the FPGA implementation of multiplications with LUT or DSP [19], using shift-adds instead of multiplications [12], and so on.

## 3 Proposed FPGA Architecture for Element Computation of HEVC DCT

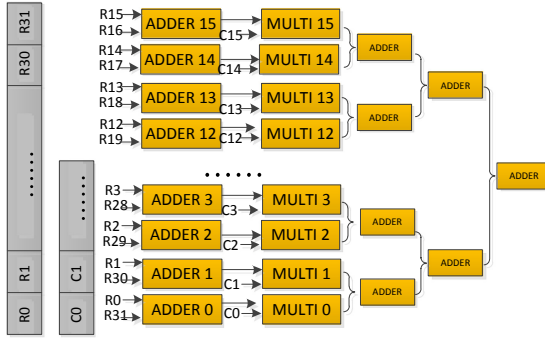
### 3.1 FPGA Design for an Element in $32 \times 32$ Block Size

FPGA design for an element in  $32 \times 32$  block size is explained first in this subsection.

Butterfly transform can be applied several times in DCT. Each time an even data matrix splits into smaller even and odd parts, until down to  $4 \times 4$  size. Even parts may be reused for different DCT sizes, but odd parts are prohibited. Table 1 illustrates how hardware resources for processing one pixel point varies with depth of butterfly transform. It is apparent that more levels of butterfly transforms require less number of multipliers.

We propose to apply butterfly transform only once, instead of three times as in [6]. With the resource overhead of six more multipliers, the benefit of our design is to reuse these multipliers, which need a lot of FPGA resources, in smaller DCT sizes. Thus, the multipliers in our design can compute the multiplication for  $32 \times 32$ ,  $16 \times 16$ ,  $8 \times 8$  and  $4 \times 4$  block sizes, hence greatly improving the efficiency of the FPGA resources, which is greatly important for the whole system of HEVC FPGA solution.

Figure 2 shows the hardware design for an element computation by a row and an even coefficient column, in a  $32 \times 32$  block size. The result of residual data R0 plus R31 and the first coefficient C0 are supplied to the multiplier MUTI0; the result of residual data R1 plus R30 and the first coefficient C1 are supplied to the multiplier MUTI1, and so on.



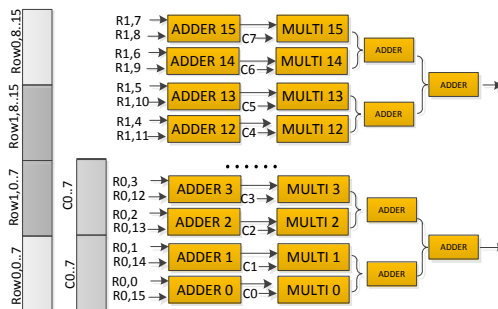
**Fig. 2.** An element computation by a row and an even coefficient column in a  $32 \times 32$  block size

Then the results of MULTI0, MULTI1, ..., and MULTI15 are accumulated together. Thus, an element for 1D-DCT is generated.

With all adders for butterfly computation replaced with subtractors, the hardware design above can generate an element for a row with an odd column coefficient data.

### 3.2 FPGA Architecture for an Element in Other Block Sizes

Then the computation of the element in  $16 \times 16$  Block Size by the Multiplier and Adder/subtractor in Fig. 1 is explained. While 16 multipliers and adders are used to compute an element in  $32 \times 32$  block size, only half of them are needed to compute an element in  $16 \times 16$  block size. To gain high efficiency of the FPGA resources, especially the multipliers which cost high FPGA resources, we compute 2 elements once using the logic in Fig. 1. Thus, all 16 multipliers and adders work at the same time. Figure 3 gives the computation of the elements in  $16 \times 16$  Block Size.



**Fig. 3.** Two elements generated by two rows and an even coefficient column in a  $16 \times 16$  block size

As shown in Fig. 3, two rows,  $r_{0,0}, \dots, r_{0,15}$  and  $r_{1,0}, \dots, r_{1,15}$  are organized as one 32-length row,  $r_{0,0}, \dots, r_{0,7}, r_{1,0}, \dots, r_{1,7}, r_{1,8}, \dots, r_{1,15}, r_{0,8}, \dots, r_{0,15}$ . The 8-length coefficient column is organized as one 16-length coefficient column data,  $c_0, \dots, c_7, c_0, \dots, c_7$ .

Then the organized 32-length row and the 16-length coefficient column are inputted to the multipliers and adders for element computation in  $32 \times 32$  block size. Two elements are outputted by the adders before the last level. With all adders for butterfly computation replaced with subtractors, the hardware design above can generate two elements for a row with an odd column coefficient data.

As analyzed above, four elements of  $8 \times 8$  block size can be generated once by the same FPGA multipliers and adders with four rows and a coefficient column; 8 elements of  $4 \times 4$  block size can be generated once by the same FPGA multipliers and adders with 8 rows and a coefficient column. Figure 4 shows the data organization of the residual rows and coefficient columns for  $8 \times 8$  block size. For  $4 \times 4$  block size, 8 elements, i.e. two rows, are generated once, with similar data organization.

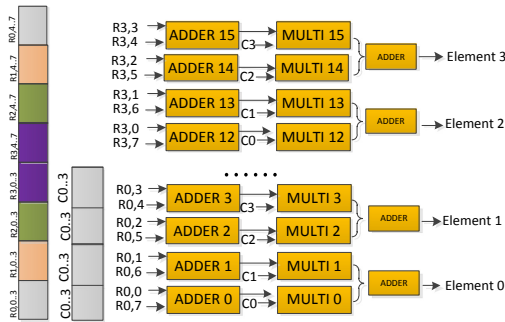


Fig. 4. Four elements generated by four rows and an even coefficient column in an  $8 \times$  block size

As shown in Fig. 2 and Fig. 3, the same computation FPGA logic can compute one  $32 \times 32$  block size element or two  $16 \times 16$  block size elements once.

To increase hardware utilization efficiency, on-chip DSP blocks are preferred to realize multiplication-addition. Here we propose to map butterfly transform into DSP48 blocks in Xilinx xc7vx690 FPGA.

### 4 FPGA Architecture for HEVC DCT

The proposed HEVC 2D-DCT transform hardware for all variable block sizes is shown in Fig. 5. The proposed hardware performs 2D DCT by first performing 1D DCT on the rows of a TU, then reordering the intermediate results to the transpose ram, and performing 1D DCT on the columns of the TU. The resulting transformed coefficients, generated by 1D row DCT, are reordered from row-sequence to row-sequence and stored in a transpose memory, and they are used as input for 1D column DCT. The FPGA architecture for 1D row DCT and 1D column DCT is the same, so only 1D row DCT is described below.

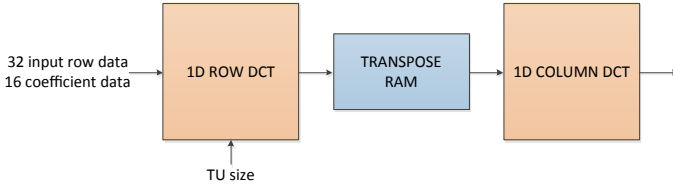


Fig. 5. Hardware design for HEVC 2D-DCT

Figure 6 shows our algorithm and the overall architecture. The 1D-DCT FPGA design includes two even FPGA design shown in Figs. 2, 3, or 4, and two odd FPGA design. Thus, our proposed HEVC design transforms 2 32-length rows and 2 16-length coefficient columns at the same time. Either  $2 \times$  block for  $32 \times 32$  block size,  $4 \times 2$  block for  $16 \times 16$  block size,  $8 \times 2$  block for  $8 \times 8$  block size, or two  $4 \times 4$  blocks for  $4 \times 4$  block size, is generated per cycle.

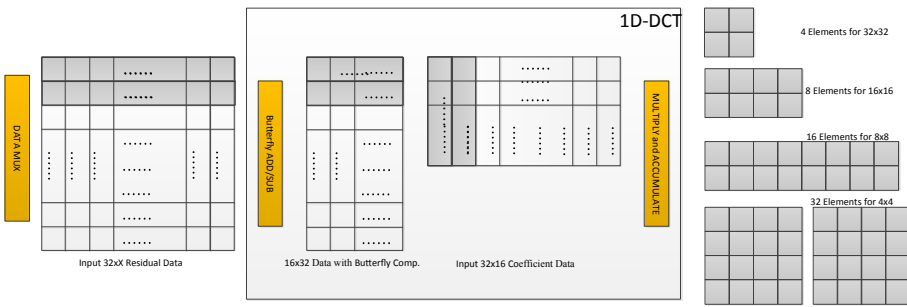


Fig. 6. Proposed algorithm and the overall architecture

Figure 7 shows our proposed architecture for HEVC 1D-DCT. The residual data row is organized as described above according to the TU size. The data mux/organization unit processes the data organization. The organized data is input to 1-D DCT unit, which generates the 1D-DCT coefficients. Finally, the 1D-DCT coefficients are reordered and written to the transpose ram. There are 3 stages in our FPGA design, one for data mux unit and two for 1D-DCT transformation.

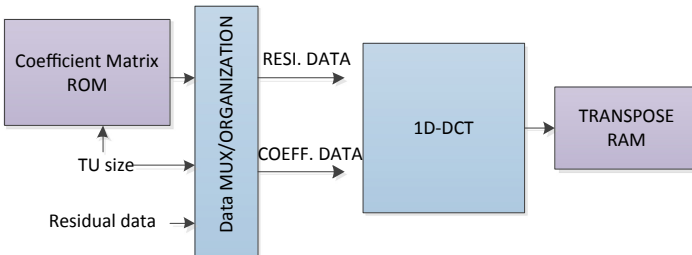


Fig. 7. FPGA architecture for HEVC 1D-DCT

## 5 Experiments

The proposed FPGA architecture for HEVC 2D-DCT transform is implemented using Verilog HDL. The Verilog RTL codes are verified with RTL simulations. RTL simulation results matched results of HEVC 2D-DCT transform implementation in Joint Exploration Test Model (JEM) 4.0 reference software encoder [2]. The Verilog RTL codes are synthesized and mapped to a Xilinx XC7VX690TFFG761-2 FPGA with speed grade 2 using Xilinx ISE. The FPGA implementation is verified to work at 250 MHz by post place and route simulations. Table 1 shows the FPGA resources needed by our design.

**Table 1.** FPGA resources needed by our design

Resource	Utilization	Available	Utilization %
LUT	2900	433200	0.67
FF	302	866400	0.03
BRAM	16	1470	1.09
DSP	64	3600	1.78

As shown in Table 1, our design has low cost of the FPGA resources.

Our design also has high computation efficiency. Table 2 compares the cycles needed for each block size 1D-DCT transformation, with [18] which proposes a design aimed both performance and the FPGA cost.

**Table 2.** Comparison of the cycles needed for each block size 1D-DCT transformation, with [18]

	Our design	Design in [18]
$32 \times 32$	256	256
$16 \times 16$	32	64
$8 \times 8$	4	16
$4 \times 4$	1/2	4

As shown in Table 2, compared with [18], our proposed FPGA design have much higher performance for little block sizes.

As demonstrated in our experiments, our FPGA design can support all possible sizes in HEVC transform, use as little as possible FPGA implementation cost, i.e. the FPGA resources, and gain high-performance as possible. Therefore, our FPGA architecture can be used for the whole HEVC FPGA solutions as the 2-D DCT hardware design.

## 6 Conclusion

Nowadays, computational resources in FPGA makes the study of whole HEVC FPGA implementation is gaining more and more attention. An efficient 2-D DCT FPGA architecture, used for the whole HEVC FPGA solution should have the following features: supporting all possible sizes in HEVC transform; using as little as possible FPGA implementation cost, i.e. the FPGA resources; gaining high-performance as possible. Compared with existing 2D-DCT FPGA architecture, our design can compute all block sizes in HEVC with the same FPGA design, which means the FPGA resources for butterfly PEs can deal with the block sizes of  $32 \times 32$ ,  $16 \times 16$ ,  $8 \times 8$  and  $4 \times 4$ , thus greatly reducing the FPGA implementation cost. Our butterfly PEs, which compute one element for the block size of  $32 \times 32$  per cycle, can compute 2 elements for the block size of  $16 \times 16$  per cycle, 4 elements for the block size of  $8 \times 8$  and 8 elements for the block size of  $4 \times 4$  per cycle, thus gaining high computation efficiency.

**Acknowledgments.** We acknowledge the reviewers for their insightful comments. This work is supported by the HGJ2017 under Grant No.2017ZX01028103 and Grant 2018ZX01029103.

## References

1. Meuel, H., Munderloh, M., Ostermann, J.: Stereo mosaicking and 3D video for single-view HDTV aerial sequences using a low bit rate ROI coding framework. In: International Conference on Advanced Video and Signal Based Surveillance, pp. 1–6 (2015)
2. Bhaskaranand, M., Gibson, J.: Low-complexity video encoding for UAV reconnaissance and surveillance. In: Military Communications Conference, pp. 1633–1638 (2011)
3. Bhaskaranand, M., Gibson, J.: Low complexity video encoding and high complexity decoding for UAV reconnaissance and surveillance. In: International Symposium on Multimedia, pp. 163–170 (2013)
4. Zhang, Q., Chang, H., Huang, X., Huang, L., Su, R., Gan, Y.: Adaptive early termination mode decision for 3D-HEVC using inter-view and spatio-temporal correlations. *Int. J. Electron. Commun.* **70**(5), 727–737 (2016)
5. Bossen, F., Bross, B., Suhring, K., Flynn, D.: HEVC complexity and implementation analysis. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1685–1696 (2012)
6. Kalali, E., Ozcan, E., Yalcinkaya, O., Hamzaoglu, I.: A low energy HEVC inverse transform hardware. *IEEE Trans. Consum. Electron.* **60**(4), 754–761 (2014)
7. Kessentini, A., Samet, A., Ayed, M., Masmoudi, N.: Performance analysis of inter-layer prediction module for H.264/SVC. *Int. J. Electron. Commun.* **69**(1), 344–350 (2015)
8. Samcovic, A.: Mathematical modeling of coding gain and rate-distortion function in multi-hypothesis motion compensation for video signals. *Int. J. Electron. Commun.* **69**(2), 487–491 (2015)
9. Budagavi, M., Fuldseth, A., Bjontegaard, G., Sze, V., Sadafale, M.: Core transform design in the high efficiency video coding (HEVC) standard. *IEEE J. Selected Topics Signal Process.* **7**(6), 1029–1041 (2013)
10. Rao, K.R., Yip, P.: Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press, Inc., Cambridge (1990)
11. Kalali, E., Mert, A.C., Hamzaoglu, I.: A computation and energy reduction technique for HEVC discrete cosine transform. *IEEE Trans. Consum. Electron.* **62**(2), 166–174 (2016)



12. Conceicao, R., Souza, J., Jeske, R., Zatt, B., Porto, M., Agostini, L.: Low-cost and high throughput hardware design for the HEVC  $16 \times 16$  2-D DCT transform. *J. Integr. Circ. Syst.* **9**, 25–35 (2014)
13. Mert, A.C., Kalali, E., Hamzaoglu, I.: An FPGA implementation of future video coding 2D transform. In: *IEEE International Conference on Consumer Electronics - Berlin*. IEEE (2017)
14. Park, J.S., Nam, W.J., Han, S.M., et al.: 2-D large inverse transform ( $16 \times 16$ ,  $32 \times 32$ ) for HEVC (High Efficiency Video Coding). *J. Semicond. Technol. Sci.* **12**(2), 203–211 (2012)
15. Chen, T.H.: A cost-effective  $8 \times 8$  2-D IDCT core processor with folded architecture. *IEEE Trans. Consum. Electron.* **45**(2), 333–339 (1999)
16. Sjovall, P., Viitamaki, V., Vanne, J., et al.: High-level synthesis implementation of HEVC 2-D DCT/DST on FPGA. In: *IEEE International Conference on Acoustics*. IEEE (2017)
17. Huang, J., Parris, M., Lee, J., et al.: Scalable FPGA architecture for DCT computation using dynamic partial reconfiguration. *ACM Trans. Embedded Comput. Syst.* **9**(1), 269–272 (2009)
18. Chen, M., Zhang, Y., Lu, C.: Efficient architecture of variable size HEVC 2D-DCT for FPGA platforms. *Aeu Int. J. Electron. Commun.* **73**, 1–8 (2017)
19. Mert, A.C., Kalali, E., Hamzaoglu, I.: An FPGA implementation of future video coding 2D transform. In: *IEEE International Conference on Consumer Electronics - Berlin*. IEEE (2017)