Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Flexible FPGA 1D DCT hardware architecture for HEVC

Hrvoje Mlinarić, Alen Duspara, Daniel Hofman & Josip Knezović

To cite this article: Hrvoje Mlinarić, Alen Duspara, Daniel Hofman & Josip Knezović (2023) Flexible FPGA 1D DCT hardware architecture for HEVC, Automatika, 64:4, 886-892, DOI: 10.1080/00051144.2023.2228580

To link to this article: https://doi.org/10.1080/00051144.2023.2228580

Published online: 04 Jul 2023.

AUTOMATIKA
2023, VOL. 64, NO. 4, 886–892
https://doi.org/10.1080/00051144.2023.2228580

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Flexible FPGA 1D DCT hardware architecture for HEVC

Hrvoje Mlinarić, Alen Duspara, Daniel Hofman and Josip Knezović

Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia

ABSTRACT

In this work, we propose a flexible 1D DCT hardware design with a constant throughput of 32 pixels per cycle for all transform block (TB) size modes. The design supports all TB sizes defined in the HEVC standard. Flexibility is achieved by multiplexing only the outputs, which results in lower data path delays compared to other proposed designs. Also, the additional partial butterfly units are transferred to the output of transformation cores. The transformation operation is done in a single stage to minimize the latency of the design and reduce hardware usage. The highly parallel input reduces the need for a very high operational frequency, which is suitable for low-power FPGA designs. Four different reusable transformation cores are used that are designed using parallel Multiple Constant Multiplication (MCM) units to further reduce the calculation time. The design was implemented on the Virtex UltraScale + device. The implementation has hardware usage of 21,818 LUTs, and it can reach the maximal throughput of 4.90 Gsps at the working frequency of 153 MHz which is enough to support the video resolutions of up to 8192 × 4320@60fps. Comparison with the other works shows that DCT FPGA implementation without DSPs can reach the performance of the ASICs with trade-offs in power consumption.

ARTICLE HISTORY
Received 21 February 2023
Accepted 19 June 2023

KEYWORDS
HEVC; DCT; FPGA; Multiple Constant Multiplication

Introduction

Standardization organizations ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) created the High-Efficiency Video Coding (HEVC) standard [1,2]. These well-known organizations, ITU-T and ISO/IEC, have developed and enhanced video coding standards over time. ITU-T developed H.261 [3] and H.263 [4], whereas ISO/IEC developed MPEG-1 [5] and MPEG-4 Visual [6]. Moreover, these two organizations worked together to develop the H.262/MPEG-2 Video [7] and H.264/MPEG-4 Advanced Video Coding (AVC) [8] standards. Prior to the HEVC initiative, the most recent video coding standard was H.264/MPEG-4 AVC, which was significantly expanded. H.264/MPEG-4 AVC has been instrumental in enabling digital video in numerous areas that H.262/MPEG-2 did not previously encompass. HEVC was created to address all existing H.264/MPEG-4 AVC applications, with a primary concentration on two issues: increased video resolution and increased utilization of parallel processing architectures.

The HEVC standard aims to achieve several objectives, such as coding effectiveness, simplicity in transport system integration, robustness to data loss, and ability to implement parallel processing architectures.

The HEVC video coding layer employs a hybrid strategy that has been utilized by all video compression standards since H.261 was released. This method predicts data spatially from one region to the next within the same frame without requiring previous or subsequent frame. The initial frame of a video sequence and each random access point are encoded with intra-frame prediction alone. The frame is divided into block-shaped regions, and the decoder is informed of the precise block partitioning. Inter-frame temporally predictive coding modes are utilized for all remaining frames of a sequence between two intra-frame. The encoder calculates motion data consisting of the designated reference image and motion vector (MV) for use in predicting the samples of each block. The residual signal of intra-frame or inter-frame prediction is then subjected to a linear spatial transform. Scaled, quantized, and entropy-encoded transform coefficients are transmitted alongside.

HEVC, a new video encoding standard, has added more transform block sizes to improve the efficiency of encoding 4 K and 8 K video files. During the development of HEVC, the goal was to implement it efficiently in software using processor acceleration functions such as SIMD capabilities, vector operation and parallel processing. In order to improve device compatibility and efficiency, the floating-point discrete cosine transform has been substituted with an integer version, and the quantization procedure has been simplified by converting to scalar multiplication.

CONTACT Hrvoje Mlinarić hrvoje.mlinaric@fer.hr Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb 10000, Croatia

The additional functionality and flexibility of the quad-tree coding structures provide HEVC with many more possible combinations than its predecessors. Therefore, the complexity of the encoder that fully exploits the capabilities of HEVC is expected to be several times higher [9]. This added complexity also has a significant positive effect on rate-distortion performance. In fact, it can achieve up to 50% bitrate reduction while retaining the same video quality compared to AVC [10].

Due to its inherent properties, such as near-optimal decorrelation and symmetry, the Discrete Cosine Transform (DCT) is extensively used in image and video processing. In its original form, the DCT employs floating-point operations, which introduces additional complexity and error between the forward and inverse transformations. The most recent coding standards, including High-Efficiency Video Coding (HEVC) and Advanced Video Coding (AVC), utilize integer transformations, which are integer approximations of the DCT. The HEVC standard specifies core transform kernel matrices for block-based video compression transformations in two dimensions. These matrices are designed as approximations of the DCT with finite precision, allowing embedded structure and preservation of symmetry properties [11]. HEVC also supports a greater number of transform block sizes than its predecessors (4×4, 8×8, 16×16, and 32×32), which enhances data compression, particularly when processing video with a high spatial resolution.

In the process of encoding video to HEVC bitstream, integer DCT is used to transform the residual signal, which is calculated as the difference between the current and the predicted block. The residual signal is divided into square blocks of size N × N called transform blocks (TBs). The DCT property of separability enables the transform to be performed as two separated one-dimensional transforms. Also, the symmetry properties of basis transformation can be used to reduce the total amount of arithmetic operations. The transform operations make a significant amount of total encoding time, which can mostly be attributed to the rate-distortion optimizations (RDO) process. Much effort has been made to optimize the hardware designs of transform operations. Several proposed transform hardware designs used multiplierless Multiple Constant Multiplication (MCM) units constructed as add-shift trees. The MCMs require less hardware than conventional multipliers, if the multiplication of the input value is done in parallel, which suits the lower frequency FPGA designs.

Shen et al. [12] were among the first to propose the integer DCT hardware architecture that supports larger transform sizes (16 and 32 points). This proposal suggests a new architecture for 1D IDCT that works with different video standards and has 4/8/16/32 points. They use SRAM-based transpose memory to save hardware space and reduce power usage. To lower costs even more, they set up the multiplication of 4/8 point IDCT with MCM, while the multiplication of 16/32 point IDCT uses reusable regular multipliers. This design can handle 4 pixels per cycle, and its 5-stage pipelined architecture enables it to work at a maximum frequency of 350 MHz, though it may take up a bit more silicon area. Overall, the design efficiency is 31% better than the previous work based on the normalized criterion [13–15].

Meher and Park [16,17] conducted extensive rese arch on the area and power efficiency of DCT architectures. They presented a fully parallel and folded structure of 2D DCT with generalized and flexible reusable DCT core architectures. They illustrated a generalized design for implementing the transform with different lengths and demonstrated the reusability with various solutions. In their flexible and reusable design, two N/2-point units are used for each N-point unit. When performing the N/2-point transformation, the inputs are multiplexed from the N-point to the second N/2-point unit to maintain throughput. The flexible design is achieved, but the data path delay is also increased because of added input and an output multiplexer.

In the paper [18], the authors present three hardware designs for the HEVC standard's N-point 1D-DCT transformation. Regarding the number of inputs (N) and the number of bits per input (n), these designs offer flexibility and customization. One partial butterfly unit, one $N/2$-point 1D-DCT, $N/2$ MCM units, and $N/2$ adder units constitute them. To increase the operating frequency, the MCM modules use add-shift operations with minimal depth. Additionally, the inputs and outputs of each adder and subtractor use minimal bit representation to conserve hardware resources without jeopardizing the DCT's accuracy and data compression. The proposed 1D-DCT design requires the least amount of area resources and has a higher operating frequency and throughput than other designs presented in the literature, according to the results of the synthesis. The proposed designs are extremely suitable for hardware implementation of the HEVC standard due to their support for 4 K and 8 K UHDV video resolutions.

In a study [19], a new and improved integer DCT architecture was introduced. It is based on optimizing the bit-widths of adders, and the proposed design for different transformation sizes showed significant improvements. Compared to binary multipliers, the optimized multiplier design resulted in an average enhancement of 7% in delay, 33% in gate count, and 29% in consumed power.

In [20], a new set of implementation algorithms for the HEVC transformation were proposed. These DCTs algorithms transform the 4×4, 8×8, 16×16, and 32×32 blocks. Compared to traditional algorithms, these new algorithms can achieve around a 20% reduction in critical path length while using relatively few additions and shift operations. This shorter critical path leads to faster processing times, higher speeds, and increased throughput. By implementing these algorithms in hardware designs, throughput can be increased by 20% while maintaining reasonable resource consumption when compared with other implementation algorithms.

A new method for creating 4-point and 8-point DCT architecture is presented in the paper [21]. This involves making minor adjustments to the 8-point DCT equations, sharing resources more efficiently, and ensuring constant throughput regardless of the DCT type. The paper also introduces a new technique for creating transpose memory, which enables concurrent intermediate data processing. Memory usage during a 4-point DCT operation decreases by 25% when this method is used. The proposed architecture has also been implemented and tested on the FPGA platform.

The paper [22] proposes a new hardware architecture for the development of a high-performance HEVC 2-D DCT transform accelerator. To reduce area and increase performance, the transform matrix is decomposed into multiple matrices that are implemented using processing element hardware with only shift and adder operations. The experimental results demonstrate that this architecture is capable of processing frames with resolutions up to 8 K in real-time, attaining a throughput of 30 frames per second on the Xilinx Zynq-7000 Artix-7 FPGA. A configurable carry-save adder tree-based multiplier is proposed in [23].

This work proposes a flexible 1D DCT hardware architecture design with a constant throughput of 32 pixels per cycle in all size modes. Flexibility is achieved by multiplexing only the outputs, which results in lower data path delays than from the design proposed in work [16]. The architecture is designed in a single stage for lower latency and less resource allocation. The highly parallel input reduces the need for a very high operating frequency, which is suitable for FPGA designs. The implementation of the design on the selected device has enough throughput to support higher levels of HEVC bitrates and resolutions such as 8192 × 4320@60fps.

Flexible DCT model

In HEVC, after the frame residual signal is divided into $N \times N$ blocks, where $N = 2^M$, $N \in$ 4,8,16,32, and M is in the range [9,12], the two-dimensional integer forward DCT (2D DCT) is applied to each block. The 2D DCT is a separable transform and can be divided into two N-point one-dimensional DCT (1D DCT) applied to each row and each column, respectively. Smaller kernel matrices are embedded into the 32 × 32 transform kernel matrix. This property enables efficient hardware sharing between different transform sizes (4 × 4, 8 × 8 and 16 × 16). It defines that for d^N_{ij}, where the d^N_{ij} are the smaller transform matrices ($N \in$ 4,8,16), the next expression is valid.

$$d^N_{ij} = d^{32}_{i(32/N),j}, i,j = 0,\dots,N-1 \tag{1}$$

The embedded structure of the kernel matrices enables creating the flexible DCT design that efficiently reuses hardware. Flexibility can be defined as the ability of the design which enables multiple working modes. Also, reusability can be defined as the design feature which enables the hardware submodule units of the design to have more than one function. To enable runtime flexibility of the DCT with efficient hardware reusability, the next mathematical expressions can be used:

$$\begin{bmatrix} Y(0) \\ Y\left(\frac{N}{4}\right) \\ Y\left(\frac{2N}{4}\right) \\ Y\left(\frac{3N}{4}\right) \end{bmatrix} = \sum_{i=0}^{\left(\frac{N}{4}\right)-1} \left(C_4 \begin{bmatrix} a(i,0) \\ b_4(i,0) \\ a(i,1) \\ b_4(i,1) \end{bmatrix} \right) \tag{2}$$

if $N = 4$

$$\begin{aligned} a(i,j) &= X(4 \cdot i + j) + X(4 \cdot (i+1) - 1 - j) \\ b_n(i,j) &= X(n \cdot i + j) - X((n \cdot (i+1)) - 1 - j) \end{aligned} \tag{3}$$

for $j = 0, 1, \dots, \frac{N}{2} - 1, X = [X(0), X(1), \dots, X(N-1)]$ is the input vector, and $Y = [Y(0), Y(1), \dots, Y(N-1)]$ is N-point DCT of X. $C_N$is N-point integer DCT kernel matrix of size $N \times N$.

For $N = 8$, the additional expression can be constructed (4) using the property described in Equation (1) that defines only the outputs that use the embedded values in the transform kernel matrix:

$$\begin{bmatrix} Y(N/8) \\ Y\left(\frac{3N}{8}\right) \\ Y\left(\frac{5N}{8}\right) \\ Y\left(\frac{7N}{8}\right) \end{bmatrix} = \sum_{i=0}^{\left(\frac{N}{8}\right)-1} \left(M_8 \begin{bmatrix} b_8(i,0) \\ b_8(i,1) \\ b_8(i,2) \\ b_8(i,3) \end{bmatrix} \right) \tag{4}$$

where M_N is defined as:

$$M_N(i,j) = C_N(2i+1,j), for\, 0 \le i,j \le N/2 - 1 \tag{5}$$

The same analogy as in Equation (4) can be used for the sizes $N = 16$, 32 to define the new values in the output vector. If the input matrix size N is considered as the working mode in flexible HEVC DCT design, this model shows how that design can efficiently reuse hardware up to the bottom.

According to (2) and (4), the DCT core functions can be derived as follows:

$$DCT_{K,I} = C_4 \begin{bmatrix} a(I,0) \\ b_4(I,0) \\ a(I,1) \\ b_4(I,1) \end{bmatrix} \text{ for } K = 4 \tag{6}$$

$$DCT_{K,I} = M_K \begin{bmatrix} b_K(I,0) \\ b_K(I,1) \\ \vdots \\ b_K(I,K/2-1) \end{bmatrix} \text{ for } K = 8, 16, 32 \tag{7}$$

Table 1. Unique values of MN/2 matrix.

Size (N)	Unique values
4	83, 36
8	89, 75, 50, 18
16	90, 87, 80, 70, 57, 43, 25, 9
32	90, 88, 85, 82, 78, 73, 67, 61, 54, 46, 38, 31, 22, 13, 4

where K is the size of the DCT core, and possible values of I are defined as follows:

$$I = 0, \ldots, \left(\frac{N}{K}\right) - 1 \tag{8}$$

The number of DCT core operations with specific size K needed for the N-point transformation is defined as follows:

$$G(K) = \frac{N}{K} \tag{9}$$

Then the total number of core transform operations of all sizes needed for the N-point DCT when using this model can be defined as the following expression.

$$T(N) = \begin{cases} G(N) & for\, N = 4 \\ G(N) + T\left(\frac{N}{2}\right) & for\, N = 8, 16, 32 \end{cases} \tag{10}$$

Equations (2) and (4) show how hardware can be reused only in one direction as the DCT operations with $K > N$ are never used. The hardware designed for the N-point transformation, the $\frac{N}{M}, (N \geq M)$ M-point transformation can be performed simultaneously instead of one

N-point. If the flexible design has $T(N)$ cores, then it can perform two $N/2$-point transformations. This property of the model increases the hardware reusability and gives the design ability to have a constant throughput. The transform operation specified in the HEVC standard can be described as the constant matrix multiplication (CMM) problem, where one of the operands has a constant value. Therefore, instead of using fully featured multipliers in design, parallel Multiple Constant Multipliers (MCMs) can be utilized in order to achieve a more performant architecture. The MCM unit replaces the multiplication logic with the add-shift tree structure, which can concurrently calculate products of input with multiple predefined constants. The unique constants values (see Table 1) are derived from kernel matrix $M_N$for each size N.

Proposed architecture

We propose the hardware architecture design of the 32-point flexible DCT described in the model above. The design consists of $T(32) = 15$ DCT cores described in (6) and (7). The Equation (9) shows how many cores with the specific size K are present in the design. The design can perform as 32-point, or 2×16-point, or 4×8-point or 8×4-point vector transforms simultaneously and maintain the 32 pixels per cycle throughput. When performing the N-point transformation where $N < 32$, all $DCT_{K,I}$cores where $K > N$ are unused. Additionally, the architecture is separated, enabling the $\frac{32}{M}$ M-point transforms ($M \leq 32$) by reusing the $DCT_{K,I}$cores where $K \leq M$. The presented model enables wiring the inputs directly to the transformation cores so the flexibility can be achieved without multiplexing the inputs.

Figure 1 presents a high-level overview of the architecture where it is shown how the inputs are wired to the DCT cores. The groups of cores are marked to depict which part of the design is used for a specific transform size mode. In the output adder units, the results are formed using the outputs of the DCT cores as defined in (2) and (4). The DCT cores are implemented according to papers [16] and [17]. The architecture has a shorter data path, but it exploits the symmetry properties of the HEVC kernel matrix only at the DCT core level. The results are scaled and multiplexed depending on the mode selected.

Each of the transformation cores, as shown in Figure 2, consists of one partial butterfly adder, N/2 MCM units, and output adders. The parameters in the figure are $K = 8{,}16{,}32$, $M = \log2K$, $n = 16$, and I is defined in (8). The inputs are brought to the partial butterfly unit, which generates the b_K(i,j) coefficients according to (3), where i is the ordinal number of the core (I), and j is the ordinal input number of each core. The MCM and the adder units implement the matrix product defined in (7) with the size K. When $K = 4$, the butterfly unit must also generate a(i,j) coefficients according to (2), and the matrix product must be performed according to (6).

The parallel MCM units are used instead of the regular multiplicators in order to reduce the calculation time. Each MCM_K unit has implemented the multiplications with constants defined in Table 1 for the $N = K$. In the proposed architecture; those units have minimal depth size and minimal bit representation to increase the operating frequency and reduce hardware usage. All MCM_K designs can be obtained from a multiplier block generator for parallel MCM [24].

Implementation results

The design is synthesized and implemented using the Vivado Design Suite for the device Virtex UltraScale + with speed grade 3. The bit-width is fixed to 16 bits, which is enough to support all existing HEVC profiles. The module is implemented in out-of-context mode with a constrained clock. The implementation process was executed multiple times with different settings until optimal results were achieved. The maximum frequency and area utilization results were obtained from post-implementation reports.

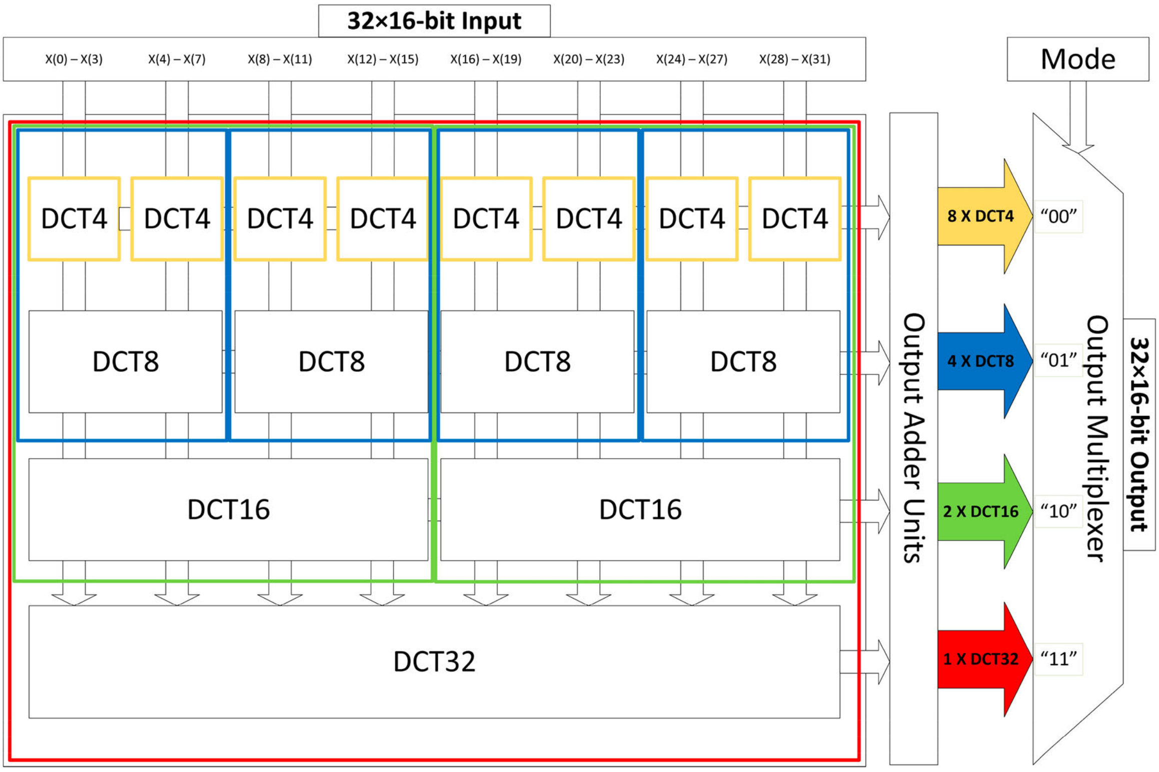

Figure 1. Proposed flexible DCT architecture.

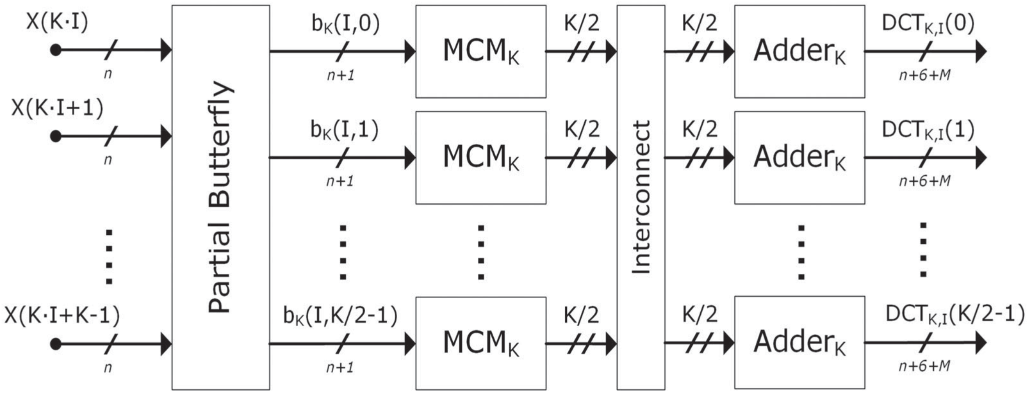

Figure 2. Proposed transformation cores.

In Table 2, the implementation results are presented. The achieved maximum output throughput of 4.90 Gsps allows the design to support resolutions up to 8192 × 4320@60fps. It shows that DCT FPGA designs can reach very high performance without using DSPs. If more performance is needed, the design can also be pipelined. Every pipeline stage requires higher hardware allocation and increases the latency of the device. The longest data path delay of the design is 6.54 ns, of which logic takes 2.16 ns (33.022%) and routing 4.38 ns (66.978%). Power estimation analysis shows that the implementation consumes 4.17 W of total on-chip power, of which 1.09 W is dynamic power. For reference, the consumption of ASIC implementations in other works is under 100 mW. This difference is expected since the higher power consumption is one of the main trade-offs between the reconfigurability of FPGA and the efficiency of ASIC chips implementations.

Table 2. Implementation results.

Technology	Virtex UltraScale+
Type	1D DCT
TB Sizes	4/8/16/32
Bit-width	16
Max. F. [MHz]	153
Utilization	21,818 LUTs
Throughput [Gsps]	4.90
Latency[Clock Cycles]	1

Table 3 presents a comparison of performance with other designs mentioned in this work. The other designs are implemented as an ASIC device, which has different characteristics than FPGA implementations. All designs are flexible and support all of the sizes defined in the HEVC standard. The designs with

Table 3. Comparison with previous works.

Design	Technology	Type	Throughput [Gsps]	Latency [Clock Cycles]
Shen [12]	SMIC 0.13 um	2D IDCT	1.40	5
Park [16]	TSMC 90 nm CMOS	1D DCT	1.50	3
Meher [17]	TSMC 90 nm CMOS	1D DCT	2.99	3
Proposed	Virtex UltraScale+	1D DCT	4.90	1

pipelined structures can work on the higher operating frequency but also has longer latency. Our proposed architecture has a more parallel structure, which results in higher throughput at a lower working frequency. This suits well the FPGA devices in which dynamic power consumption is significantly affected with higher working frequency.

Conclusions

In this work, a flexible 1D DCT hardware architecture is proposed. Design can support all TB sizes defined in the HEVC standard. To minimize the data path delays, flexibility is achieved without an input multiplexer, unlike in other proposed works. Also, the needed additional partial butterfly units are located on the output side. The parallel structure of the design reduces the need for higher working frequencies, which is suitable for FPGA designs. The transformation operation is done in a single stage to minimize the latency of the design and optimize hardware usage. Four different reusable transformation cores are designed using parallel MCM units to reduce the calculation time. The design is implemented on the Virtex UltraScale + device with 16-bit inputs. The implementation has hardware usage of 21,818 LUTs, and it can reach the maximal throughput of 4.896 Gsps at the maximal working frequency of 153 MHz which is enough to support the video resolutions up to 8192 × 4320@60fps. In comparison with the other works, DCT FPGA implementation without DSPs can reach the performance of the ASICs at the cost of increased power consumption.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work has been supported in part by the project KK.01.2.1.02.0054 “Razvoj uredaja za prijenos video signala ultra niske latencije” (Development of ultra low latency video signal transmis-sion device), financed by the EU from the European Regional Development Fund.

References

[1] Sullivan GJ, Ohm JR, Han WJ, et al. Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol. 2012 Dec;22(12):1649–1668. doi:10.1109/TCSVT.2012.2221191.

[2] Bross B, Han W-J, Sullivan GJ, et al. High efficiency video coding (HEVC) text specification draft 9, document JCTVC-K1003, ITU-T/ISO/IEC joint collaborative team on video coding (JCT-VC), Oct. 2012.

[3] Video codec for audiovisual services at px64 kbit/s, ITU-T Rec. H.261, version 1: Nov. 1990, version 2: Mar. 1993.

[4] Video coding for low bit rate communication, ITU-T Rec. H.263, Nov. 1995 (and subsequent editions).

[5] Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s—part 2: video, ISO/IEC 11172-2 (MPEG-1), ISO/IEC JTC 1, 1993.

[6] Coding of audio-visual objects—part 2: visual, ISO/IEC 14496-2 (MPEG-4 visual version 1), ISO/IEC JTC 1, Apr. 1999 (and subsequent editions).

[7] Generic coding of moving pictures and associated audio information-part 2: video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG 2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994.

[8] Advanced video coding for generic audio-visual services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003 (and subsequent editions).

[9] Bossen F, Bross B, Suhring K, et al. HEVC complexity and implementation analysis. IEEE Trans Circuits Syst Video Technol. 2012 Dec;22(12):1685–1696. doi:10.1109/TCSVT.2012.2221255.

[10] Ram C, Panwar S. Performance comparison of high efficiency video coding (HEVC) with h.264 AVC. Dec. 2017. Jaipur, India 2017 13th International Conference on Signal-Image Technology Internet Based Systems (SITIS). 2017;303–310. doi:10.1109/SITIS.2017.58.

[11] Budagavi M, Fuldseth A, Bjntegaard G, et al. Core transform design in the high efficiency video coding (HEVC) standard. IEEE J Sel Top Signal Process. 2013 Dec;7(6):1029–1041. doi:10.1109/JSTSP.2013.2270429.

[12] Shen S, Shen W, Fan Y, et al. A Unified 4/8/16/32-Point Integer IDCT Architecture for Multiple Video Coding Standards.Melbourne, VIC, Australia. 2012 IEEE International Conference on Multimedia and Expo. 2012. pp. 788–793. doi:10.1109/ICME.2012.7.

[13] Wang K, Chen J, Cao W, et al. A reconfigurable multi-transform VLSI architecture supporting video codec design. IEEE Trans Circuits Syst II Exp Briefs. Jul. 2011;58(7):432–436. doi:10.1109/TCSII.2011.2158265.

[14] Qi H, Huang Q, Gao W. A low-cost very large scale integration architecture for multistandard inverse transform. IEEE Trans Circuits Syst II Exp Briefs. Jul. 2010;57(7):551–555. doi:10.1109/TCSII.2010.2048477.

[15] Lee S, Cho K. Architecture of transform circuit for video decoder supporting multiple standards. Electron Lett. Feb. 2008;44(4):274–275. doi:10.1049/el:20083168.

[16] Park SY, Meher PK. Flexible integer DCT architectures for HEVC. Beijing, China. 2013 IEEE International Symposium on Circuits and Systems (ISCAS). 2013. pp. 1376–1370. doi:10.1109/ISCAS.2013.6572111.

[17] Meher PK, Park SY, Mohanty BK, et al. Efficient integer DCT architectures for HEVC. IEEE Trans Circuits Syst Video Technol. 2014 Jan;24(1):168–178. doi:10.1109/TCSVT.2013.2276862.

[18] Bolanos-Jojoa JD, Velasco-Medina J. Efficient hardware design of N-point 1D-DCT for HEVC. Bogota, Colombia. 2015 20th Symposium on Signal Processing, Images and Computer Vision (STSIVA). 2015. pp. 1–6. doi:10.1109/STSIVA.2015.7330449.

[19] Abdelrasoul M, Sayed MS, Goulart V. Scalable Integer DCT Architecture for HEVC Encoder. Pittsburgh, PA, USA. 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). 2016. pp. 314–318. doi:10.1109/ISVLSI.2016.98.

[20] Do Thu T.T., Tan Y.H., Yeo C. High-throughput and low-cost hardware-oriented integer transforms for HEVC. Paris, France. 2014 IEEE International Conference on Image Processing (ICIP). 2014. pp. 2105–2109. doi:10.1109/ICIP.2014.7025422.

[21] Chatterjee S, Sarawadekar K. P. A low cost, constant throughput and reusable 8 × 8 DCT architecture for hevc. Abu Dhabi, United Arab Emirates. 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS). 2016. pp. 1–4. doi:10.1109/MWSCAS.2016.7869994.

[22] Ansari A. E., Mansouri A., Ahaitouf A. An efficient VLSI architecture design for integer DCT in HEVC standard. Agadir, Morocco. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). 2016. pp. 1–5. doi:10.1109/AICCSA.2016.7945747.

[23] Mohamed Asan Basiri M., Noor Mahammad S. K. High Performance Integer DCT Architectures for HEVC.Hyderabad, India. 2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID). 2017. pp. 121–126. doi:10.1109/VLSID.2017.68.

[24] Franchetti F, Low T-M, Popovici T, et al. SPIRAL: extreme performance portability. Proc IEEE. 2018; 106(11):1935–1968. doi:10.1109/JPROC.2018.2873289.