ISSN [Online]: 2583-2654

Effective accelerator for 2D DCT/IDCT using improved Loeffler Architecture

Dr Bhuvanendhiran T , Manas rufus M, Sanjay S ,Vishva D
Department of ECE, Dr. T. Thimmaiah Institute of Technology, KGF, Karnataka, India

Abstract: This paper proposes an effective hardware accelerator for 2D 8x 8 discrete cosine trans- form (DCT) and inverse discrete cosine transform (IDCT) using an improved Loeffler architecture. The accelerator optimizes the data stream of the Loeffler 8-point 1D DCT/IDCT according to the characteristics of image and video processing. An 8-stage pipeline structure greatly improves the processing speed by reasonably dividing the number of clock cycles and simplifying the arithmetic operations in each cycle. The multiplication-free approximation of the DCT coefficients is implemented through adders and shifters,combined with both fixed-point and canonic signed digit (CSD) coding. In particular, the proposed fast parallel transposed matrix architecture achieves the function of row-column coefficient conversion with lower circuit complexity. The FPGA implementation of the proposed architecture uses a Virtex-7 XC7VX330T device, running at 288 MHz with a throughput of 558 M Pixel/sec, and a Full HD real-time frame rate of up to 269 fps. Only 33 cycles are required to complete the 8 x 8 blocks of 2D DCT/IDCT, which can be used as a high- performance hardware accelerator for image andvideo compression encoding.

Keywords: Loeffler algorithm, DCT, IDCT, parallel transpose, hardware accelerator.

1. INTRODUCTION

In recent years portable multimedia devices have experienced a huge demand under the rapid development of computer information technology. These devices require real-time processing of high- resolution, high-quality digital images and video data with stringent requirements on available resources, memory and power consumption. Discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) have a wide range of applications in image and video coding due to their good energy compression performance, such as JPEG [3], MPEG [4]– [6], H.26x [7], [8]. In addition, the new compression scheme high efficiency video coding (HEVC) [9] uses DCT/IDCT integer conversion to achieve efficient compression performance at about half the bit rate needed to maintain the same video quality as H.264. The computation of 8-point 1D DCT/IDCT appeared more fast algorithms in the early stage, most of which require 12-13 multiplications and 29 additions to implement [10]. The Loeffler algorithm, another theoretical cornerstone of this project, draws inspiration from the Fast Fourier Transform (FFT) and the mathematical theory behind butterfly operations. The project's theoretical enhancement of this algorithm seeks to optimize its computational efficiency while maintaining the accuracy and precision inherent in its theoretical foundations. Parallel processing theory, a theoretical pillar of modern computer architecture, forms the bedrock for achieving high throughput in the hardware accelerator. Theoretical principles of parallelism, including data-level parallelism and pipelining, guide the project in maximizing processing speed while minimizing resource utilization, which is crucial for resource-efficient hardware design. Factorization and Loeffler Architecture the Loeffler architecture employs the factorization of the 2D DCT/IDCT into a product of 1D DCT/IDCT operations. This factorization reduces the overall computational complexity, enabling more efficient hardware implementation. By breaking down the2D

ISSN [Online]: 2583-2654

transform into a series of 1D operations, the Loeffler architecture allows for parallelization and pipelining, taking advantage of the parallel processing capabilities of hardware.Parallel Processing is a fundamental concept in accelerating signal processing tasks. In the context of our accelerator, parallelism is exploited at multiple levels.First, within each 1D DCT/IDCT operation, parallelism is achieved by processing multiple data elements simultaneously. Second, at the 2D level, multiple independent 1D operations can be processed concurrently, further enhancing throughput. Precision optimization involves carefully selecting the number of bits used to represent data during computations. By tailoring the precision of arithmetic operations to the specific requirements of the DCT/IDCT algorithms, we can achieve a balance between computational accuracy and the hardware resources required. This is crucial for optimizing performance and minimizing power consumption. Memory management is a critical aspect of hardware accelerators. Optimized memory hierarchies, including on-chip buffers and caching mechanisms, reduce the need for off-chip memory accesses. This minimizes data movement, which is especially beneficial for large-scale multimedia data processing, contributing to lower power consumption and improved overall system efficiency.

The configurability of the accelerator allows users to adapt the hardware to specific application requirements. Parameters such as block size, precision, and performance goals can be customized. This flexibility ensures that the accelerator can be seamlessly integrated into various systems with different constraints and specifications. The accelerator is designed to be integrated into larger systems, particularly SoCs. This integration involves considerations such as communication interfaces, compatibility with existing processing units, and easeof integration into diverse applications. The goal is tocreate a modular and scalable solution that can be easily adopted in different computing environments. Given the precision optimizations and configurable nature of the accelerator, thorough error analysis is conducted to understand the impact of reduced precision on the quality of the transformed signals. Error compensation techniques may be employed to mitigate any loss in quality, ensuring that the accelerator meets the desired performance criteria.Performance evaluation involves metrics suchas throughput, latency, power consumption, and area utilization. These metrics are crucial for assessing the effectiveness of the hardware accelerator in comparison to software-based implementations and other existing hardware solutions.Loeffler Algorithm (1989): The original Loeffler algorithm introduced an efficient method for computing the 1D DCT with only 11 multiplications. This algorithm became a foundation for many subsequent developments in DCT-based compression.JPEG Standard (1992): The JPEG image compression standard adopted the 2D DCT as its core transformation method, and variations of the Loeffler algorithm were widely usedfor practical implementations. Ongoing Optimization Efforts: Over the years, researchers and engineers have continuously worked on optimizing the DCT algorithms for improved speed and efficiency, especially as computing technology advanced.2D DCT Formulation: The 2D DCT is mathematically defined as the product of two 1D DCTs, applied first along the rows and then along the columns of the image matrix. Loeffler Algorithm: The Loeffler algorithm optimally factorizes the 1D DCT into a sequence of simpler operations, minimizing the number of multiplications required for computation. Improved Loeffler Architecture: This could involve enhancements such as parallel processing to compute multiple DCT coefficients simultaneously, pipelining to overlap computation stages, and hardware acceleration using dedicated processing units like GPUs or FPGAs Quantization and Compression: After the DCT, quantization is typically applied to reduce the precision of

ISSN [Online]: 2583-2654

coefficients, and the resulting values are entropy encoded for compression. The acceleration of the 2D DCT is crucial for real-time applications and efficientuse in various devices. Researchers and engineers continue to explore new methods and architectures to further improve the speed and energy efficiency of DCT-based compression techniques.Parallel Processing: Row and Column Parallelism: Break down the 2D DCT computation into parallelizable tasks, allowing simultaneous processing of multiple rows and columns.

SIMD (Single Instruction, Multiple Data): Use vectorization techniques to perform the same operation on multiple data elements simultaneously, improving throughput.Pipelining Overlap of Computation Stages: Introduce pipeline architecture to overlap the computation stages, reducing the overall latency.Efficient Register Usage: Optimize register usage to maximize instruction throughput. Custom Hardware Designs Implement the DCT algorithm using dedicated hardware components such as FPGAs (Field-Programmable Gate Arrays) or ASICs (Application-Specific Integrated Circuits) for high-speed processing. GPU Acceleration Leverage the parallel processing capabilities of GPUs to accelerate DCT computations, especially in graphics and multimedia applications. Quantization and Compression Optimization Adaptive Quantization Schemes: Develop adaptive quantization techniques based on image characteristics, optimizing compression quality. Entropy Coding Improvements: Enhance entropy coding algorithms (e.g., Huffman coding) for more efficient compression of quantized DCT coefficients. Deep Learning Approaches. Explore the use of deep neural networks to learn optimal transformations for DCT, potentially replacing traditional algorithms in specific applications.

II HARDWARE ARCHITECTURE

Fig. 1 shows the hardware architecture of the 8-point DCT with improved Loeffler algorithm. The entire architecture is divided into an 8-stage pipeline structure, where each register in the same stage performs only a single addition, subtraction or multiplication operation. The input data $x0$ $x7$ duringstage 1 are added and subtracted respectively and thenstored in the register of the next stage. The addition of registers 8 and 9 is an important process for implementing critical path computation splitting.

The corresponding constant factors for the Shift-Mul unit in each stage are used in the CSD coding results in Table . Since the constant factor is expanded in theShift-Mul unit, the output result needs to be shifted right to reduce the corresponding multiple during stage 8.

The hardware architecture of the 8-point IDCT is shown in Fig. The input data $y0$- $y7$ are the coefficients after DCT, and the coefficient registers need to be swapped posi- tions before the operation is performed. The data stream of the hardware architecture can be simplified as the reverse operation of DCT (see Fig), transitioning from stage 8 to stage 1. The shift operations in stages 2, 6, 8 are used to match the correct result of the calculation. The output data $x0$-$x7$ are the original input data obtained after IDCT. The transformation result obtained by Loeffler algorithm is 8 times of the original 1D DCT/IDCT, and the result will be enlarged by 8 times after 2D DCT/IDCT. The hardware implementation shifts the result by 3bit to the right to reduce the size by 8 times. Table describes the different valid widths ofthe input and output data of the 8-point 1D DCT/IDCT in the four operating modes, where the row/column 1D DCT and row/column 1D IDCT correspond to the 8-stage pipeline architecture of Fig. 1 and Fig. 2, respectively. When the row transform module is in 1D DCT mode, the input data is the image or the difference of the image, the data range is -255 to 255, and the valid data width is

ISSN [Online]: 2583-2654

9bit. While the row transform is in 1D IDCT mode, the input data is the coefficient after 2D DCT, and the data range is -721 to 721, and the valid data width is 11bit.

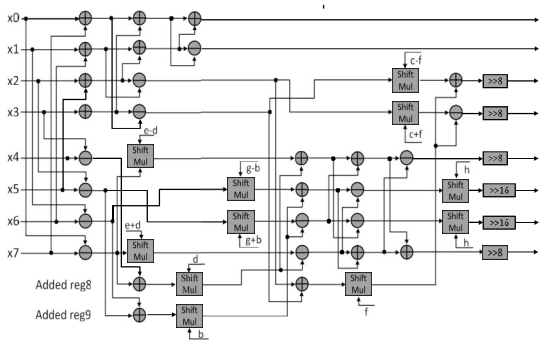

Fig 1

8-point DCT 8-stage pipeline hardware structure diagram.Fig. depicts the hardware architecture of row-column matrix transposition, where *mem*0 *mem*7 are memories generated by LUT and register resources, and each memory has 8-depth and 12-width. The width of the memory depends on the valid width 12bit of the output data of row 1D DCT/IDCT in Table . First, the input data *row*[95:0] is split into 8 independent transformation coefficients by split unit respectively Then the data is written to *mem*0 *mem*7 in parallel and stored in full after 8 cycles. Finally, all the values in the memory are spliced from low to high to form the column data *col*[95:0], through the 8-to-1 data selector, the transposition result is output in parallel cycle-bycycle.Fig. presents the 2D 8 8 DCT/IDCT pipeline architecture.The row/column 1D DCT module uses the same hardware architecture, the difference is in the valid width of the input and output data, as well as the IDCT module. The 8 8 block is the pixel matrix to be transformed, and each row of 8-point is respectively spliced into parallel row data through the SP unit (see Fig). The 64 pixels consist of 8 rows, each row contains 8 independent pixels. Thus, all the pixels of the 8x 8 block are pre-stored in the generated memory

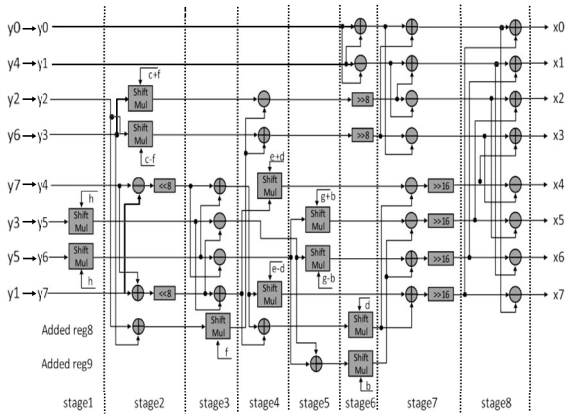

Fig 2

with 8-depth and 88-width. Among them, 8-depth means 8 rows of pixels, and 88-width is composed of 8 independent 11bit pixels, which can simultaneously ensure the valid data width of the row 1D DCT/IDCT module input. First, each rowof pixels is parallel data spliced by 8-point, and eight rows of pixels are output to the row 1D DCT/IDCT module cycle-by-cycle. After 8 cycles, the eighth row of data output is finished, while the row 1D DCT/IDCT module outputs the transformation result of the first row. While the row conversion coefficients are being output cycle-by- cycle, the transpose module is caching the results. Then after 8 cycles, the caching of the transform coefficients of the eight rows is completed, and the parallel row-column transposition is completed in the next cycle. Based on the same principle, when the transpose module outputs column data cycle-by-cycle, the column 1D DCT/IDCT module is running at the same time. After 8 cycles, the eighth column data output is completed, while column 1D DCT/IDCT starts to output the transformation result of the first row to the output 2D DCT/IDCT module. Finally, 8 cycles are needed to obtain the computed 8x 8 block transform coefficient matrix. Efficient processing of the data stream is achieved by pipelining, which requires 33 cycles to complete the 8 8 blocks of 2DDCT/IDCT.

ISSN [Online]: 2583-2654

III. EXPERIMENTAL RESULTS

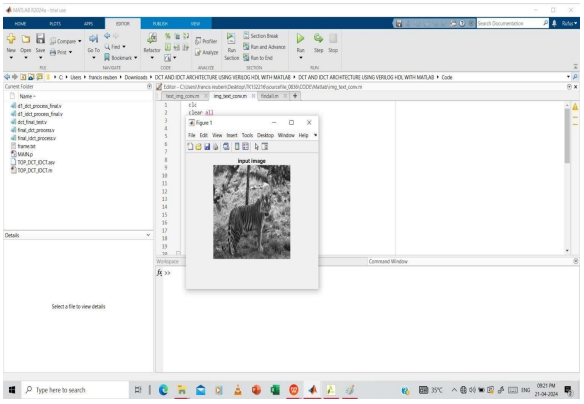

Fig 3 Input image

The above figure shows the input image. Commencingwith the meticulous development of a MATLAB script tailored for input image processing, this endeavor underscores the foundational role of rigorous software engineering in our image compression framework.The script seamlessly interfaces with our DiscreteCosine Transform (DCT) algorithm, demonstrating adept management of diverse image formats and resolutions while maintaining stringent fidelity standards. Noteworthy features include a modular design, robust error handling mechanisms, and intuitive interfaces, all indicative of best practices in software engineering. This MATLAB implementation represents a pivotal element within our comprehensive solution for image compression and transformation tasks, epitomizing the fusion of technical sophistication and user-centric design principles

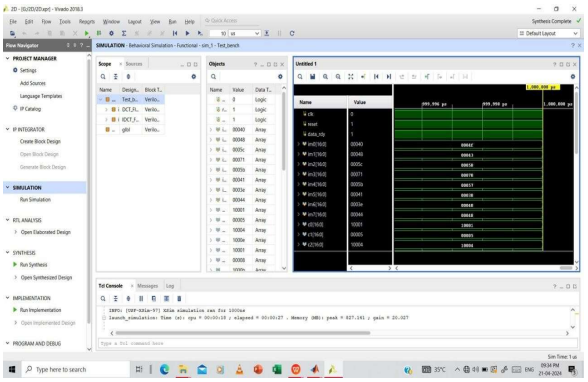

Fig 4 Test bench wave form.

The above figure shows .The refinement of our programs and test bench, we initiated the validation phase using the ModelSim Altera software platform. Utilizing this tool, we meticulously executed our Verilog or VHDL designs alongside their respective test bench environments. Through extensive simulation and analysis, we subjected our implementations to a battery of test scenarios, including corner cases and boundary conditions, to ensure comprehensive validation. The output generated from ModelSim Altera provided detailed insights into the behavior and performance of our designs. Specifically, we scrutinized waveform traces, simulation logs, and assertion results to assess the functionality and correctness of our implementations. Positive outcomes observed in these outputs affirmed the successful execution ofour programs, validating their compliance with specified functional requirements and design constraints.

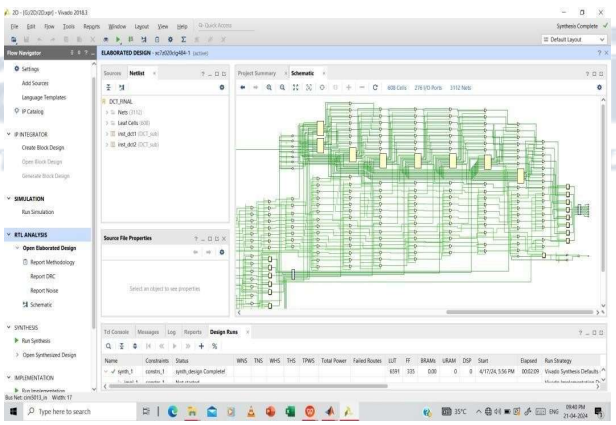

Fig 5 Architecture

The above figure shows following the refinement of our enhanced Loeffler architecture code, we proceeded to execute it within the Xilinx ISE software environment, facilitating hardware synthesis and analysis. Subsequently, the generated output, as depicted in the accompanying figure, provided valuable insights into the performance and functionality of our optimized design. This execution phase served to validate the efficacy of our architectural enhancements, offering tangible results that

ISSN [Online]: 2583-2654

underscored the improvements achieved in terms of computational efficiency and resource utilization.

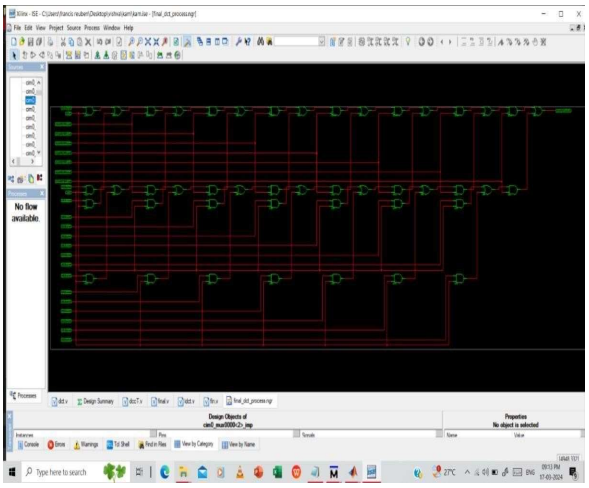

Fig 6 Architecture of cim0

The above figure shows the successful development of our hardware accelerator design within the Xilinx our project journey. It underscore sour commitment to pushing the boundaries of hardware acceleration technology and demonstrates our ability to deliver high-performance computing solutions tailored to specific application requirements.

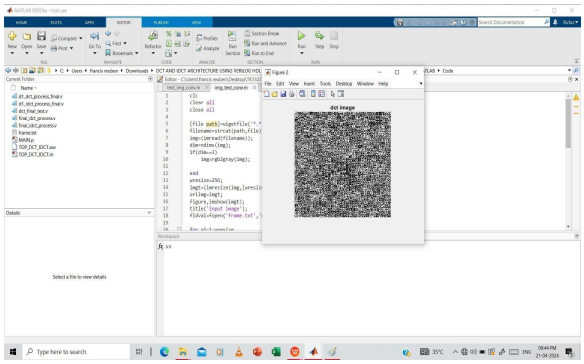

Fig 7 DCT image

The above figure shows the achieve a throughput of252 M Pixels/sec at a clock rate of 256MHz.

Although only 2021 LUTs and 1110 registers are used, the ROM-based storage computing architecture consumes more block RAM resources and data reading and writing is not efficient. The four-stage architecture proposed by supports implementation in a variety of FPGA platforms. Although the processing time of a single 8 × 8 block is 7 cycles, it consumes alarge amount of on-chip DSP resources.

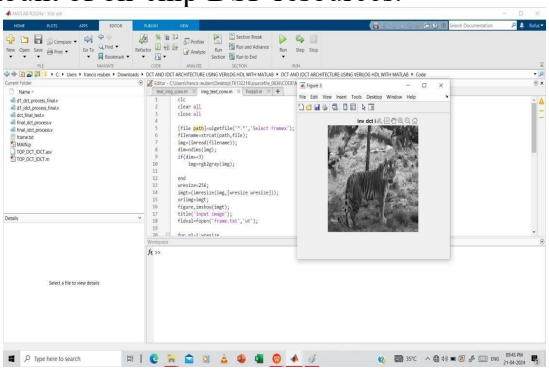

Fig 8 Inverse IDCT image

The above figure shows the 2D 8×8 DCT+IDCT technique in the field of image noise removal. Different levels of multiplicative noise are added to the original image by different sigma values, as shown in Fig. The filter mask size is 8×8, and the heavier the mask filtering, the more high frequency information is removed from the image. Fig. are the images after 2D DCT, which are filtered by light mask, moderate mask and heavy mask, and finally restored by 2D IDCT. It can be seen that the image noise reduction effect is good, as the filter mask gradually increases, the noise removal is obvious, but some detailed information will be lost.

IV CONCLUSION

This paper proposes a fast and efficient hardware architecture for computing 2D 8x8 DCT/IDCT. Optimization of the data stream of the Loeffler 8-point 1D DCT/IDCT greatly improves the processing performance of the 8-stage pipeline structure. In addition, the processing method of approximate DCT coefficients without multiplication and the row-column fast parallel transposition provide superior compression performance under very low circuit complexity. Experimental results show that the accelerator has low resource consumption and high-speed transform performance, and is suitable for

applications where high real-time performance and high bandwidth are required in image and video hardware compression coding.

REFERENCES

[1] M. Siekkinen, E. Masala, and J. K. Nurminen, ''Optimized upload strate- gies for live scalable video transmission from mobile devices,'' IEEE Trans. Mobile Comput., vol. 16, no. 4, pp. 1059–1072, Apr. 2017.

[2] A. Shabani, M. Sabri, B. Khabbazan, and S.Timarchi, ''Area and power- efficient variable-sized DCT architecture for HEVC using muxed-MCM problem,'' IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 68, no. 3,pp. 1259–1268, Mar. 2021.

[3] G. K. Wallace, ''The JPEG still picture compression standard,'' IEEE Trans. Consum. Electron., vol. 38, no. 1, pp. 18–34,Feb. 1992.

[4] K. Brandenburg and G. Stoll, ''ISO/MPEG-1 audio: A generic standard for coding of high-quality digital audio,'' J. Audio Eng. Soc., vol. 42, no. 10,pp. 780–792, 1994.

[5] P. A. A. Assuncao and M. Ghanbari, ''A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams,'' IEEE Trans. Circuits Syst. Video Technol., vol. 8, no. 8, pp. 953–967, Dec. 1998.

[6] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, ''Scalable rate control for MPEG-4 video,''IEEE Trans. Circuits Syst. Video Technol., vol. 10, no. 6,pp. 878–894, Sep. 2000.

[7] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, andS. K. Mitra, ''Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard,'' IEEETrans. Circuits Syst. Video Technol., vol. 6, no. 2, pp. 182–190, Apr. 1996.

[8] H. Schwarz, D. Marpe, and T. Wiegand, ''Overview of the scalable video coding extension of the H.264/AVC standard,''IEEE Trans. Circuits Syst. Video Technol.,vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[9] 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.

[10] W.-H. Chen, C. Smith, and S. Fralick, ''A fast computational algorithm for the discretecosine transform,'' IEEE Trans. Commun., vol. 25, no. 9,pp. 1004–1009, Sep. 1977.

[11] Z. Wang, ''Fast algorithms for the discrete Wtransform and for the discrete Fourier transform,'' IEEE Trans. Acoust., Speech, Signal Process., vol. 32, no. 4, pp. 803–816, Aug. 1984.

[12] B. G. Lee, ''A new algorithm to compute the discrete cosine trans-form,'' IEEE Trans. Acoust., Speech, Signal Process., vol. 32, no. 6,pp. 1243–1245, Dec. 1984.

[13] M. Vetterli and H. J. Nussbaumer, ''Simple FFT and DCT algorithms with reduced number of operations,'' Signal Process., vol.6, no. 4, pp. 267–278, Aug. 1984.

[14] N. Suehiro and M. Hatori, ''Fast algorithms for the DFT and other sinu- soidal transforms,'' IEEE Trans. Acoust., Speech, Signal Process., vol. 34, no. 3, pp. 642–644, Jun. 1986.

[15] H. S. Hou, ''A fast recursive algorithm for computing the discrete cosine transform,'' IEEE Trans. Acoust., Speech, Signal Process., vol. 35, no. 6,pp. 1455–1461, Oct.1987.

[16] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, ''Practical fast 1-D DCT algorithms with 11 multiplications,'' in Proc.Int. Conf. Acoust., Speech, Signal Process., vol. 2, May 1989, pp. 988–991.

[17] M. El Aakif, S. Belkouch, N. Chabini, and M. M. Hassani, ''Low power and fast DCT architecture using multiplier-less method,'' in Proc. Faible Tension Faible Consommation (FTFC), May 2011, pp. 63– 66.

[18] Y. Shen and H. Oh, ''Pipelined implementation of AI-based Loeffler DCT,'' IEICE Electron. Exp., vol. 10, Oct. 2013, Art. no. 20130319.

[19] P. Kitsos, N. S. Voros, T. Dagiuklas, and A.N. Skodras, ''A high speed FPGA implementation of the 2D DCT for ultra high definition video coding,'' in Proc. 18thInt. Conf. Digit. Signal Process. (DSP), Jul. 2013,pp. 1–5.

[20] V. Dimitrov, K. Wahid, and G. Jullien, ''Multiplication-free 8 8 2D DCT architecture using algebraic integerencoding,'' Electron. Lett., vol. 40, no. 20, pp. 1310–1311, Jan. 2004.

[21] A. Madanayake, ''A row-parallel 8 8 2-D DCT architecture using alge- braic integer- based exact computation,'' IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 6, pp. 915–929, Jun. 2012.

[22] A. Edirisuriya, A. Madanayake, R. J. Cintra, V. S. Dimitrov, and N. Rajapaksha, ''A single-channel architecturefor algebraic integer-based 8x 8 2-D DCT computation,'' IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 12, pp. 2083– 2089, Dec. 2013.

[23] D. F. G. Coelho, S. Nimmalapalli, V. S. Dimitrov, A. Madanayake, R. J. Cintra, and A. Tisserand, ''Computation of 2D 8 8 DCT based on the Loeffler factorization using algebraic integer encoding,'' IEEE Trans. Comput., vol. 67, no. 12, pp. 1692– 1702, Dec. 2018.

[24] Y. L. Shen, S. Jung, and H. C. Oh,''Loeffler DCT accelerator for small portable devices,'' IEICE Electron. Exp., vol. 12, no. 13, 2015, Art. no. 20150411.

[25] Z. Wu, J. Sha, Z. Wang, L. Li, and M. Gao, ''An improved scaled DCT architecture,'' IEEE Trans. Consum. Electron., vol. 55, no. 2, pp. 685–689,

[26] May 2009.P. Subramanian and A. S. C. Reddy, ''VLSI implementation of fully pipelined multiplierless 2D DCT/IDCT architecture for JPEG,'' in Proc. IEEE 10th Int. Conf. Signal Process., Oct. 2010, pp. 401–404.

[27] Y. Xing, Z. Zhang, Y. Qian, Q. Li, and Y. He, ''An energy-efficient approximate DCT for wireless capsule endoscopy application,'' in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2018, pp. 1–4.

[28] A. C. Mert, E. Kalali, and I. Hamzaoglu, ''An FPGA implementation of future video coding 2D transform,'' in Proc. IEEE 7th Int. Conf. Consum. Electron.- Berlin (ICCE-Berlin), Sep. 2017, pp. 31– 36.

[29] N. Ahmed, T. Natarajan, and K. R. Rao, ''Discrete cosine transform,'' IEEE Trans. Comput., vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[30] Z. Tang, J. Zhang, and H. Min, ''A high- speed, programmable, CSD coefficient FIR filter,'' IEEE Trans. Consum. Electron., vol. 48, no. 4pp. 834–837, Nov. 2002.

[31] S. Liu, R. Yu, H. Yang, and H. Huang, ''Unified algorithms for computation of different points integer 1-D DCT/IDCT forthe HEVC standard,'' in Proc. Int. Conf. Softw. Intell. Technol. Appl. Int. Conf. Frontiers Internet Things, 2014, pp. 207– 211.

ISSN [Online]: 2583-2654

[32] M. Masera, M. Martina, and G. Masera, ''Adaptive approximated DCT architectures for HEVC,'' IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 12, pp. 2714–2725, Dec. 2017.

[33] S. Chatterjee and K. Sarawadekar, ''An optimized architecture of HEVC coretransform using real-valued DCT coefficients,'' IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 65, no. 12, pp. 2052– 2056, Dec. 2018.

[34] M. Chen, Y. Zhang, and C. Lu, ''Efficient architecture of variable size HEVC 2D- DCT for FPGA platforms,'' AEU-Int. J. Electron. Commun., vol. 73, pp. 1–8, Mar. 2017.

[35] A. C. Mert, E. Kalali, and I. Hamzaoglu, ''High performance 2D trans- form hardware for future video coding,'' IEEE Trans. Consum. Electron., vol. 63, no. 2, pp. 117–125, May 2017.

[36] W. Imen, B. Fatma, M. Amna, and N. Masmoudi, ''DCT -II transform hardware- based acceleration for VVC standard,'' in Proc. IEEE Int. Conf. Design Test Integr. Micro Nano-Syst. (DTS), Jun. 2021, pp. 1– 5.

[37] A. Singhadia, M. Mamillapalli, and I. Chakrabarti, ''Hardware-efficient 2D- DCT/IDCT architecture for portable HEVC-compliant devices,'' IEEE Trans. Consum. Electron., vol. 66, no. 3, pp. 203–212, Aug. 2020.

[38] S. Cichoń and M. Gorgoń, ''H.265 inverse transform FPGA implemen- tation in impulse C,'' in Proc. Federated Conf. Comput. Sci. Inf. Syst., Sep. 2017, pp. 607–611.

[39] E. Kalali and I. Hamzaoglu, ''FPGA implementations of HEVC inverse DCT using high-level synthesis,'' in Proc. Conf.Design Archit. Signal Image Process. (DASIP), Sep. 2015, pp. 1–6.

[40] D. Mukherjee and S. Mukhopadhyay, ''Hardware efficient architecture for 2D DCT and IDCT using Taylor-seriesexpansion of trigonometric functions,'' IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 8,pp. 2723–2735,Aug. 2020.