



An efficient and unified 2D-inverse integer cosine transform (IICT) FPGA-hardware implementation for HEVC standard

Ahmed Kammoun¹ · Fatma Belghith¹ · Hassen Loukil¹ · Nouri Masmoudi¹

Received: 14 March 2018 / Revised: 29 April 2019 / Accepted: 28 May 2019 / Published online: 5 June 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

The HEVC video coding standard supports different transform sizes ranging from 4-point to 32-point. In fact, multiple transform sizes improve coding efficiency, but increase as well the computational complexity. Hardware decoders apply different techniques to satisfy real-time requirements. This paper describes a novel design methodology of a unified 2D inverse core transform IICT. The hardware architecture is based on a 1D-IICT block and a transpose buffer FIFO memory used to store the intermediate values of 1D transform. All this process is controlled in such a way to reduce the hardware and memory resources. To support the different transform sizes, matrix multiplications are simplified based on transform blocks decomposition into fixed-size sub-blocks in previous works. The architecture was developed for an FPGA device. Synthesis results on Startix III FPGA device show that the proposed design, operating at 266 MHz, is sufficient to decode high resolution videos using only 10% of total pins and about 33% of the hardware resources offered.

Keywords High efficiency video coding (HEVC) · 2D-inverse core transform · FIFO

1 Introduction

Within the ISO/IEC (International Standardization Organization/International Electro-technical Commission) and ITU-T (International Telecommunication Union) cooperation [1], the High Efficiency Video Coding (HEVC) standard [2] provided different new features (with respect to its previous one AVC/ H.264 [3]) which we can mention the most important as follows:

- New approach of partitioning (Quad-tree) through 4 depth levels [4, 5].
- More flexible and larger transform unit sizes from 4×4 to 32×32 [2, 6].
- More precision in the Intra prediction process with 33 different directional modes [7].
- The integration of Sample Adaptive Offset filter (SAO) [8].

The introduction of these new features and techniques provided a considerable bitrate gain as it has been reduced by 50% while maintaining the same outputted video quality [9]. However, this gain comes with the cost of significant complexity level especially in terms of execution time and real time implementation.

Researchers are always heading to reduce or/and solve software and hardware implementation problems for different coding chain modules. This paper, particularly, is interested in the Inverse Integer cosine transform [10].

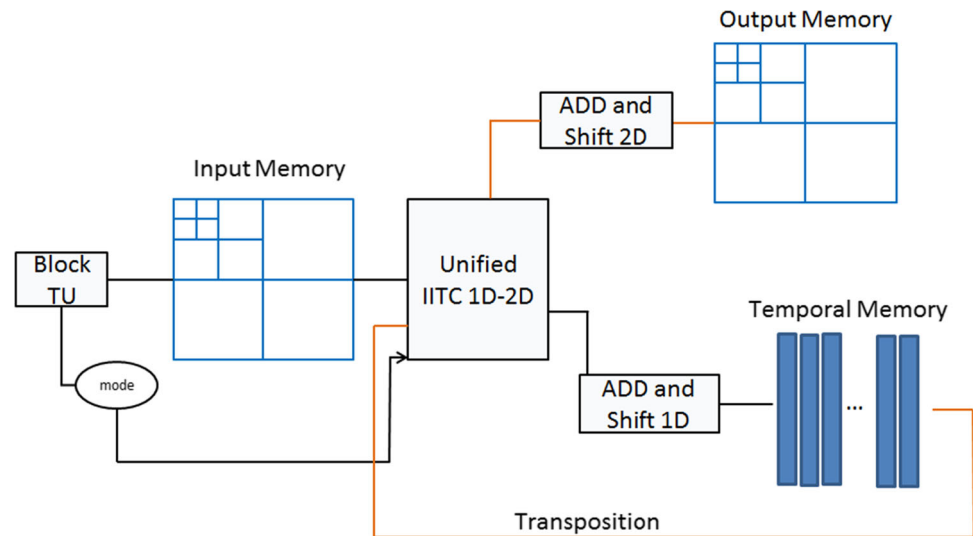
Based on the algorithm and the 1D-inverse core transform implementation proposed in [11], this paper proposes an optimized and unified 2D architecture for all Transform Unit sizes (from 4×4 to 32×32) which offers a good tradeoff between occupied area and operational frequency. The 2D approach will be detailed in this paper while delivering more accurate description details for all designed architectures.

The rest of this paper is organized as follows: In Sect. 2, some related works in the literature, interested in the inverse transform implementation of the HEVC, are presented. The general architecture of the 2D-inverse transform design is specified and detailed in Sect. 3. Section 4 shows the hardware complexity cost of the proposed

✉ Ahmed Kammoun
ahmedkammoun29@yahoo.fr

¹ National School of Engineering Sfax, University of Sfax, Sfax, Tunisia

Fig. 1 General architecture of the proposed 2D-IICT design



architectures and present the implementation synthesis results with comparison to the existing works. Finally Sect. 5 concludes the paper.

2 Related works in the literature

Several hardware implementations of the HEVC standard have been proposed in the literature. Wei Chang et al. [12] proposed a fast algorithm based on hardware-sharing architecture for 4×4 , 8×8 , 16×16 , and 32×32 inverse core transforms. It presented a highly hardware efficient design with an effective cost by using the symmetrical characteristics of the elements in inverse core transform matrices. The proposed 1-D hardware sharing scheme required 115.7 K gate counts to achieve an operational frequency of up to 200 MHz. Shen et al. [13] presented a unified VLSI architecture for 4, 8, 16, and 32-point integer IICTs. The architecture supported MPEG-2/4, H.264, AVS, VC-1, and HEVC video standards. A multiplierless technique was applied to the 4 and 8-point IDCTs. However, regular multipliers with hardware sharing were applied to the 16- and 32-point IICTs. To reduce hardware overheads, the memory was transposed using the SRAM module. The architecture supported $4 \text{ K} \times 2 \text{ K}$ (4096×2048 pixels) at 30 fps real-time decoding at 191 MHz with 93 K gate counts and 18944-bit SRAM. Ahmed et al. [14] proposed a dynamic N-point DCT for HEVC designed for all inverse transform sizes. The hardware architecture is partially folded in order to save the area and improve the speed up of the design. The proposed architecture reached a maximum frequency of 150 MHz which can support 1080 HD video codec.

The work of Manel et al. [15] described a unified hardware architecture for 4×4 , 8×8 , 16×16 , and 32×32 inverse 2D core transform IDCT in HEVC standard. It

eliminated multiplications through addition and shift operations and was based on reusing some coefficients with most occurrences as 2, 4, 9, 18, 36, and 64 to further optimize area consumption. The operating frequency of the hardware design is about 130 MHz. Ercan Kalali [16] proposed a hardware implementation of the 2D Inverse Core transform of the HEVC using High Level Synthesis (HLS) tools: Xilinx Vivado HLS, LegUp and MATLAB Simulink HDL Coder. The proposed design used 4 different cores for each TU size and then all duplicated to perform the 2D approach which affected the occupied FPGA area. The maximum operational frequencies through these HLS tools were respectively 208 Mhz, 143 Mhz and 110 Mhz. Heming Sun and al [17] interested in a reordered parallel-in serial-out (RPISO) scheme for the 2D IDCT core transform hardware implementation in order to reduce the required calculations by minimizing the redundant inputs of the butterfly structures. They also tried to reduce the memory buffer area by adopting a cyclic data mapping scheme and a pipelining schedule.

3 Implementation of the 2D-inverse transform for the HEVC standard

In compression algorithms, Integer Cosine Transform (ICT) is often used to compact the signal energy in a limited number of coefficients. On the other hand, the Inverse Integer Cosine Transform (IICT) restores the signal from the coefficients. The IICT transform process involves 2D separable transforms enabling to perform 1D horizontal transform and 1D vertical transform separately. For an $N \times N$ input block B , the 1D horizontal transform of the N rows of B is computed as given in Eq. (1)

Fig. 2 Storage of 1D outputs in FIFOs for size 4

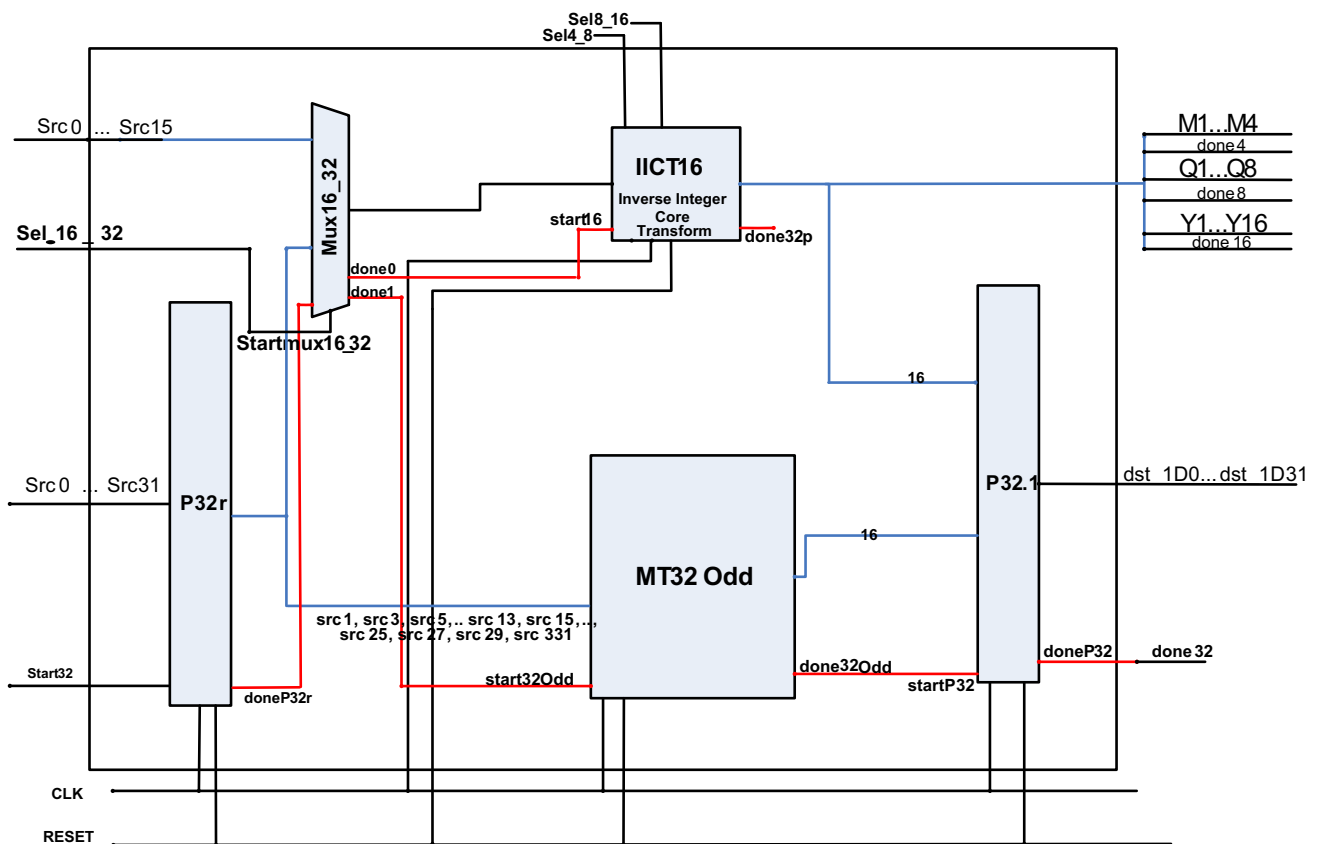
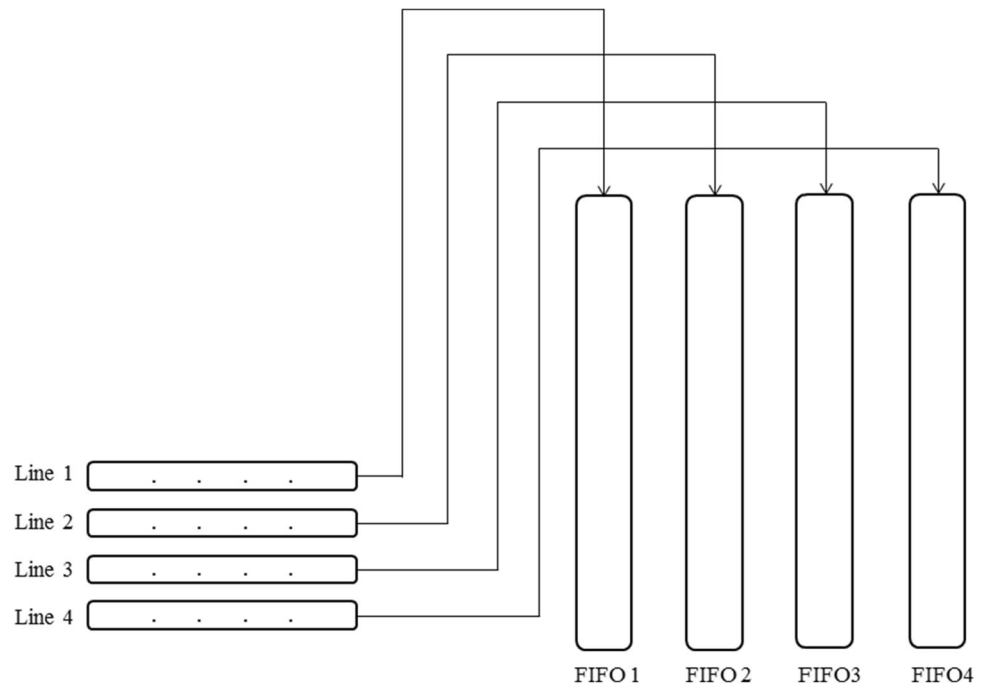


Fig. 3 Unified architecture for the 32-point 1D-IICT

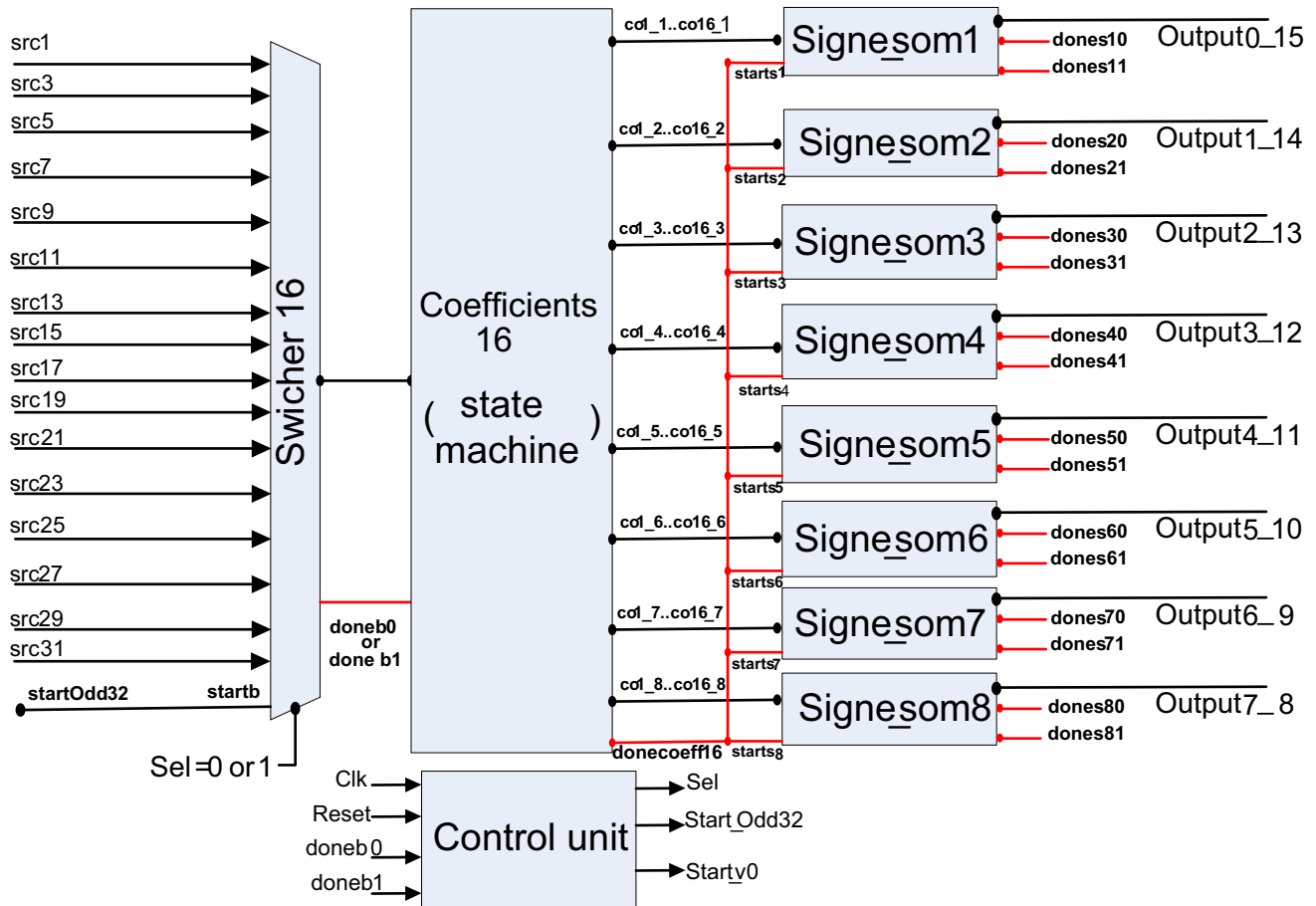


Fig. 4 Architecture of the 32 point IICT Odd part -MT32 Odd

$$Y_{int} = T_H^T \cdot B^T \tag{1}$$

where T_H is the $N \times N$ matrix of the horizontal transform coefficients and \cdot is the matrix multiplication.

The 1D vertical transform of the N columns of Y_{int} is performed by a matrix multiplication between the intermediate output coefficients (Y_{int}) and the matrix of the vertical transform coefficients T_V of size $N \times N$, as given in Eq. (2).

$$Y = T_V^T \cdot Y_{int}^T \tag{2}$$

Equation (3) describes the 2D transform operation by computing the transformed coefficients Y of the input residuals block B .

$$Y = T_V^T \cdot (T_H^T \cdot B^T)^T \tag{3}$$

Figure 1 illustrates the proposed unified 2D-IICT Core transform architecture for all block sizes from 4×4 to 32×32 . The main element is the unified 1D IICT that would be used for both horizontal and vertical transforms separated by several memory blocks, all interconnected by a control module to ensure the process properly.

In the following sections, the 2D hardware implementation will be further detailed and then we will present the synthesis results on the target FPGA device to compare it with the existing works in the literature.

3.1 Memory blocks

For this work, First In First Out (FIFO) memory blocks are used not only as a temporal buffer to store the intermediate 1D results but also as input/output storage memories. All the FIFOs used are independent-technology to ensure that our implementation could be used in different FPGA device families (Altera, Xilinx...).

– Temporal memory buffer:

This block is designed to store the output 1D matrix of the various blocks 4, 8, 16 and 32 after “Add1D” and “Shift1D” operations as presented in Fig. 1. Each row coefficients are stored in a FIFO. Read and write signals are managed according to the appropriate state machine. Since the design is unified for all 4, 8, 16 and 32 orders, depending on the mode-select which defines the desired

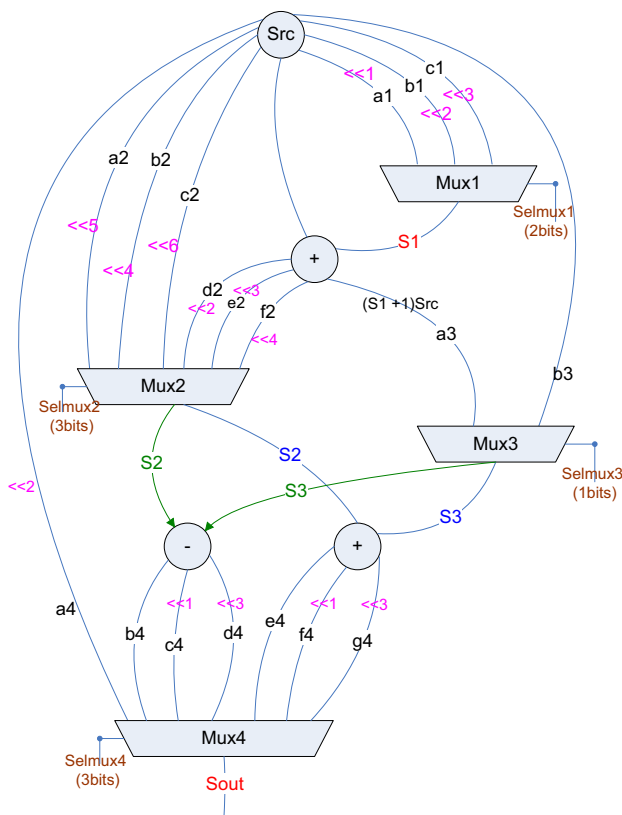


Fig. 5 Architecture of one multiplication matrix coefficient generator

block, the size of this memory appears dynamic. Indeed, we have used up to 32 FIFOs each with 32 boxes of 16 bits which is required for the largest block size. Benefitting from recursion property, when lower order N is selected, only N FIFOs are used for the storage process.

For size 4 we need only 4 FIFOs each with 4 boxes of 16 bits (Fig. 2), for size 8 we will only benefit from 8 FIFOs each with 8 boxes of 16 bits and so on. This selection of the overall size of this storage block-1D is determined by the well-defined conditions through the state machine in order to minimize the area and the execution time.

• Input/output memory:

Two other FIFO blocks are required for input and output signals put on both ends of the design. The input and output bus is defined as 32 bits. As a result, since two inputs or two outputs are concatenated each time, the size of these two memories is 512×32 bits, 128×32 bits, 32×32 bits and 8×32 bits respectively for 32, 16, 8 and 4-point transforms. This is justified by the following reasons:

- To have a generalized and unified configuration for all block sizes (from 4×4 to 32×32).
- To be compliant with different interconnect bus buffers (32 bits).

- To free interconnect buffers while transform block is processed.
- To preserve and minimize the occupied I/O pins of the target device.

3.2 1D/2D: IICT block

The work in [11] explained how the 1D inverse transform block is designed as a unified architecture (as shown in Fig. 3) that supports all transform unit sizes (4×4 to 32×32). As the decomposition of different matrices is recursive, a selection phase will indicate whether if 32-point 1D or 16-point 1D transform is used. If the second condition is set, only the first 16 inputs will be considered and transferred to the even part of the 32-point architecture. The same control path is proceeded to select size 8 or 4. This process is provided by multiplexers according to a definite selection referring to the desired transform block size. Else, all the 32 inputs will be available and 32-point 1D transform is executed through even and odd parts. The 32-IICT Odd part’s architecture is shown in Fig. 4. As explained in [11], it consists in providing eight outputs as a first step and then, by inverting its inputs order and reusing the same hardware cost, the second group of outputs is provided. Indeed, the odd matrix constant coefficients is ensured by a designed bloc generating only one coefficient of the matrix row using only 3 adders, 4 multiplexers and 14 shifters according to a definite selection. Figure 5 illustrates the architecture of the associate block. Subsequently, this latter is used 16 times at once to provide the 16 row coefficients. As a result, this operation, repeated 8 times according to a determinate state-machine, provides the first 8 output results. Finally, by inverting the inputs order and reusing the same hardware shared architecture, all the outputs will be available. This process is secured by the control unit managed by a state machine dealing with the order inverse, start and done signals and the appropriate output assignments. Finally, by inverting the inputs order and reusing the same hardware shared architecture, all the outputs will be available.

3.3 2D inverse transform process

According to a “mode” decision input (“00” referring to IICT4, “01” for IICT8, “10” for IICT16 and “11” for IICT32), the size of the transform is first chosen. After that, data which have been already stored in the input FIFO are read. The number of read clock cycles depends on the inverse transform size.

At each “start₀” signal given, the first dimension is calculated for each row of the IICT block of an appropriate size. Once the outputs are obtained, they will be stored in

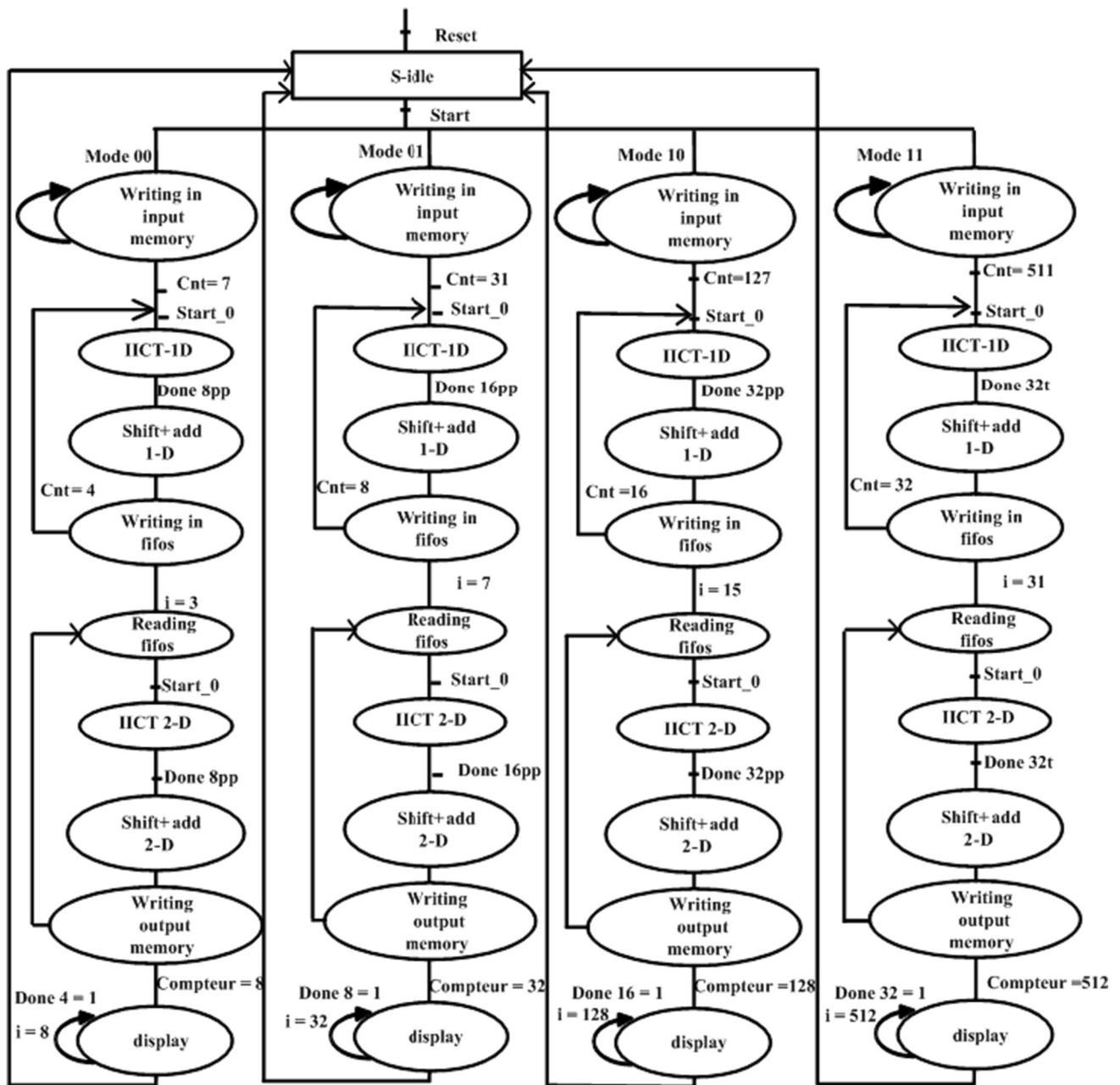


Fig. 6 State machine managing the unified IICT circuit

intermediate FIFOs whose number varies according to the size of the desired transform (32 memories in maximum for size 32).

Next, to start the 2D processing, 1D outputs are read from the FIFOs while scanning the rows if the columns were first scanned and vice versa: that is to say the first inputs vector of the IICT will contain the first values of each FIFO and so on. Indeed the transposition of the values from the FIFOs is done by acting appropriately on the read signals of each FIFO.

Once the 2D-outputs are obtained and shifted, they are stored in the output FIFO 2 by 2 and finally, we proceed with the display. Figure 6 shows all possible states for processing IICT4, IICT8, IICT16 as well as IICT32. For the IICT block, either for the 1st dimension or the 2nd, the process is the same. The difference lies in the size of the transform.

Table 1 Hardware cost of the proposed 2-D inverse core transform architecture at different transform modes

Mode	No. of adders	No. of shifts	No. of multiplexers	No. of multipliers
4 × 4	16	12	–	0
8 × 8	60	36	–	0
16 × 16	160	120	–	0
32 × 32	328	344	64	0

Table 2 Comparison of the proposed low resources contribution with other related works

	No. of adders	No. of shifts	No. of multiplexers	No. of multipliers
Original	372	–	–	344
Original*	1293	1033	–	0
Ahmed [14]	548	249	–	0
Mane [15]	500	239	–	0
Wei Chang [12]	461	256	27	0
Proposed	328	344	64	0

4 Hardware resources and synthesis results of the proposed architecture

4.1 Hardware computational complexity

The hardware sharing of the same circuit (IICT block) used in 1D process[11] provided a complete 2D inverse transform implementation while preserving the same resources. Only memory blocks which are already available in the FPGA platform device are added to both ends to read and display data.

Table 1 shows the hardware complexity required (operations number) for the 2D Inverse Transform implementation. The operation count is the same as for 1D process due to the foalded architectural approach. A comparison with other related works is presented also in Table 2.

This latter shows how the proposed work, by eliminating multiplications and reducing operations number, offers a considerable reduction in computational complexity with respect to the original designs neither multiplierless or not, especially benefitting from recursion property (* in Table 2 refers to multiplierless).

Fig. 7 Functional simulation of the unified 1D-circuit

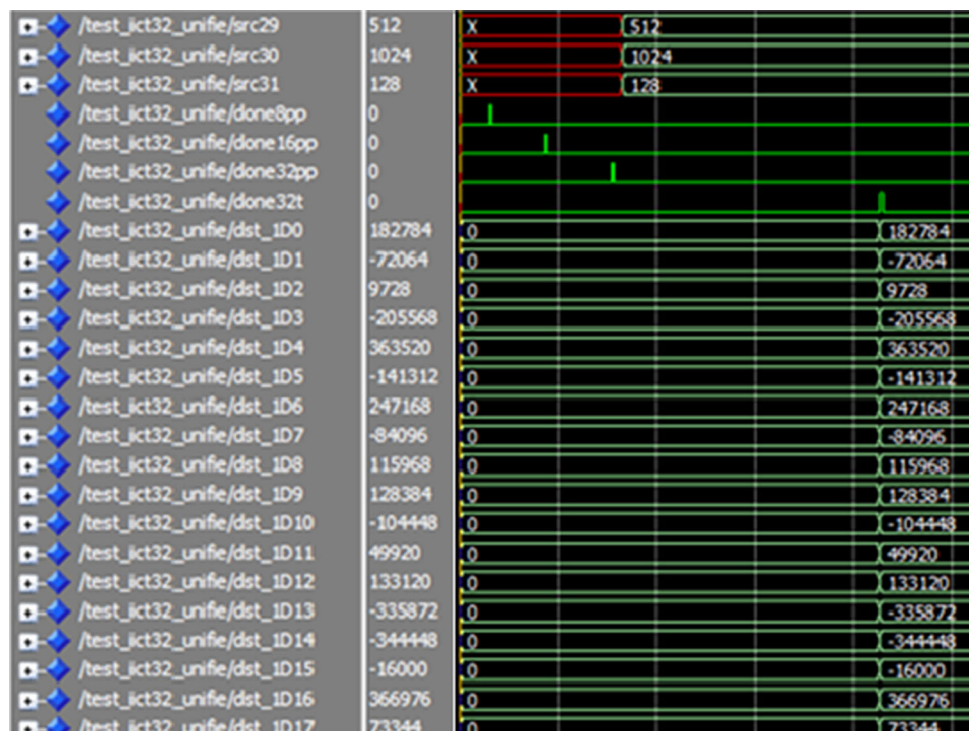


Table 3 Unified IICT32-2D synthesis results

FPGA device	Stratix III
Pins	74/744(10%)
ALUTS	28822/113,600 (25%)
Registers	21512/11,3600 (19%)
Memory (bits)	49152/5,630,972(1%)

4.2 Functional and timing simulation

To test and evaluate our architecture, a functional simulation was performed using the ModelSim simulation tool based on a “testbench” file for each inverse transform size as presented in Fig. 7.

Results are verified using output vectors extracted from HM15 reference [18]. The proposed architectures are implemented and synthesized through the software tool “Quartus II ” [19] to proceed the timing simulation that takes into account the real time constraints, delays and parasites.

The synthesis results of the unified 2D circuit under Stratix-III EP3SL150-F1152C2 FPGA device are shown in

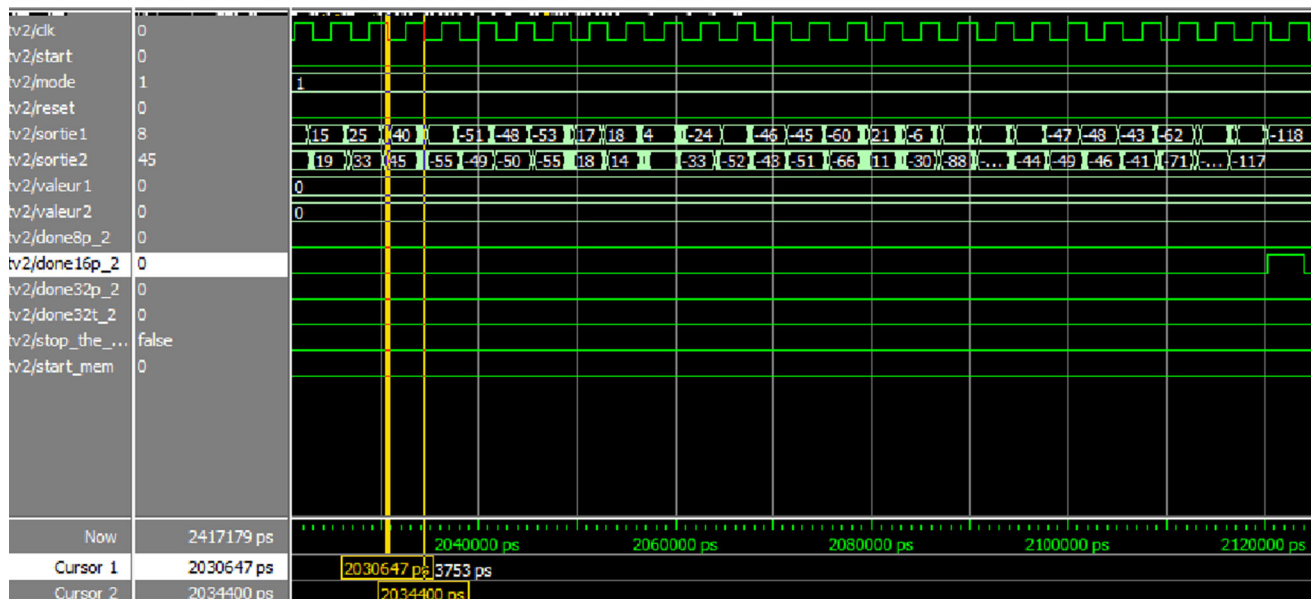
Table 3. It can be noticed that even if the target device does not offer high FPGA resource, the proposed design requires only 25% and 19% of ALUTS and Registers, respectively.

This can be justified by the combination of all the features used: hardware sharing, eliminating multiplications, FIFOs, separable 2D process and recursion.

Figure 8 illustrates an example of timing simulation for the unified circuit operating as IICT8 (i.e the selected mode is “01”). The simulation reveals that the design processes at an operational frequency up to 266 MHz which corresponds to the period $T = 3753$ ps as shown in Fig. 8.

Finally, for further evaluation, the proposed 2D-design performe in terms of throughput and occupied area is presented in Table 4 and compared to other existing works.

Indeed, the hardware sharing used in the proposed design offered a significant optimization in terms of computational complexity and area reduction. It requires about 2 and 3.5 times less hardware resources than works in [14] and [15] respectively. However, it comes with the cost of more additional execution time especially when computing 32×32 transform blocks. Although it can relatively decrease the achievable throughput, the proposed implementation is still able to support real time 2K videos

**Fig. 8** Timing simulation of the unified 2D-IICT circuit**Table 4** Comparison with other related works

	Ahmed[14]	Manel[15]	Wei Chang[12]	Proposed
Technologies	ASIC 90nm	ASIC 180nm	FPGA Stratix III	FPGA Stratix III
Gate/LE	144.8 Kgates	89Kgates	–	40.3 KLE
IICT	2D	2D	1D	2D
Throughput	$1920 \times 1080/30$	$3840 \times 2160/25$	$4 K \times 2 K/30$	$1920 \times 1080/30$
Max frequency	150 MHz	130 MHz	130.6 MHz	266 MHz

decoding at 30 frames per second with an operational frequency up to 266 MHz.

5 Conclusion

This paper proposes an efficient and unified architecture design for the 2D inverse transform used in H.265/HEVC. Hardware resource sharing and eliminating multiplication operations allow to significantly preserve the occupied area on the FPGA target device and optimize the computational complexity. The design methodology supports all the transform sizes from 4×4 to 32×32 . FIFOs memory blocks are used as transposition buffer as well as storing and displaying data to further reduce the logic use. The design required only 33% of the offered hardware area and provided an operational frequency up to 266 MHz which allow the design to sustain real time 2K video decoding.

References

1. ITU-T Recommendation H.265 and ISO/IEC 23008–2 MPEG-H Part 2, High Efficiency Video Coding (HEVC) (2013).
2. Gary, J. S., Woo-Jin, H., & Thomas, W. (2012). Overview of the high efficiency video coding (HEVC) standard. In *Circuits and systems for video technology*.
3. Wiegand, T., Sullivan, G. J., Bjøntegaard, G., & Luthra, A. (2003). Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), 560–576.
4. Seo, Chanwon, & Han, Jongki. (2010). Video coding performance for hierarchical coding block and transform block structures. *Korea Society Broading Engineers Magazine*, 15(4), 23–34.
5. Kim, I.-K., Min, J., Lee, T., Han, W.-J., & Park, J. H. (2012). Block partitioning structure in the HEVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1697–1706.
6. Bossen, F., Bross, B., Suhring, K., & Flynn, D. (2012). HEVC complexity and implementation analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1685–1696.
7. Patel, D., Lad, T., & Shah, D. (2015). Review on intra-prediction in high efficiency video coding (HEVC) standard. *International Journal of Computer Applications*, 975, 8887.
8. Chih-Ming, F., Alshina, E., Alshin, A., Huang, Y.-W., Chen, C.-Y., & Tsai, C.-Y. (2012). Sample adaptive offset in the HEVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1755–1764.
9. Ohm, J.-R., Sullivan, G., Schwarz, H., Tan, T. K., & Wiegand, T. (2012). Comparison of the coding efficiency of video coding standards including high efficiency video coding (HEVC). *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1669–1684.
10. Budagavi, M., Fuldseth, A., Bjøntegaard, G., Sze, V., & Sadafale, M. (2013). Core transform design in the high efficiency video coding (HEVC) standard. *IEEE Journal Of Selected Topics In Signal Processing*, 7(6), 1029–1041.
11. Kammoun, A., Belghith, F., Loukil, H., & Masmoudi, N. (2016). An optimized and unified architecture design for H.265/HEVC 1-D inverse core transform IEEE IPAS'16. In *International Image Processing Applications and Systems Conference*.
12. Chang, C.-W., Hsu, H.-F., Fan, C.-P., Chung-Bin, W., & Robert, C.-H. C. (2016). A fast algorithm-based cost-effective and hardware-efficient unified architecture design of 4×4 , 8×8 , 16×16 , and 32×32 inverse core transforms for HEVC. *Journal of Signal Processing Systems*, 82, 69–89.
13. Shen, S., Shen, W., Fan, Y., & Zeng, X. (2012). A unified 4/8/16/32-point integer IDCT architecture for multiple video coding standards. *IEEE International Conference on Multimedia and Expo (ICME)* (pp. 788–793).
14. Ahmed, A., & Shahid, M. U. (2012). N point DCT VLSI architecture for emerging HEVC standard. *VLSI Design*, 2012, 1–13.
15. Kammoun, M., Maamouri, E., Atitallah, A. B., & Masmoudi, N. (2016). An optimized hardware architecture Of 4×4 , 8×8 , 16×16 and 32×32 inverse transform for HEVC. In *ATSIP: international conference on advanced technologies for signal and image processing*.
16. Kalali, E., & Hamzaoglu, I. (2015). FPGA implementations of HEVC inverse DCT using high-level synthesis. In *IEEE design and architectures for signal and image processing (DASIP), Poland*.
17. Sun, H., Zhou, D., & Goto, S. (2014). A low-cost VLSI architecture of multiple-size IDCT for H265/HEVC. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 97(12), 2467–2476.
18. <https://hevc.hhi.fraunhofer.de/trac/hevc/browser-branches/archived/HM-15.0-dev>.
19. <https://www.altera.com/content/dam/altera-www/global/en-US/pdfs/literature/manual/mn-qts-quick-start.pdf>.



Ahmed Kammoun received the electrical engineering degree from the National Engineering School of Sfax (ENIS), Tunisia in 2016. Since 2017, he has joined the Electronics and Information Technology Laboratory (LETI), Sfax and became a member of VAADER team in Telecommunication and Electronic Institut Rennes (IETR), France where he is currently a Ph.D. student. His research interests include video coding and compression, potential video coding standards and codecs, FPGA hardware implementation.



Fatma Belghith was born in Sfax, Tunisia, in 1988. She received her degree in Electrical Engineering from the National School of Engineering (ENIS), Sfax, Tunisia, in 2012. She received her Ph.D. degree in Electronic Engineering in 2016. She is currently an assistant professor at the faculty of sciences and techniques of Sidi Bouzid (Tunisia) Her current research interests include video coding with emphasis on HEVC standard and beyond, hardware

implementation using FPGA and embedded systems technology.



Hassen Loukil He received electrical engineering degree from the National School of Engineering-Sfax (ENIS) in 2004. He received his M.S. and Ph.D. degrees in electronics from Sfax National School of Engineering in 2005 and 2011 respectively. He is currently an assistant professor at National School Electronic and Telecommunication of Sfax (Tunisia).He is member in the Laboratory of Electronics and Information Technology of

Sfax. His main research activities are focused on image and video signal processing, hardware implementation using FPGA technology, embedded systems.



Nouri Masmoudi received his electrical engineering degree from the Faculty of Sciences and Techniques, Sfax, Tunisia, in 1982, and the DEA degree from the National Institute of Applied Sciences, Lyon, and University Claude Bernard, Lyon, France, in 1984. From 1986 to 1990, he received PhD degree from the National School Engineering of Tunis (ENIT), Tunisia in 1990. He is currently a professor at the Electrical Engineering Department, ENIS.

Since 2000, he has been a group leader “Circuits and Systems” in the Laboratory of Electronics and Information Technology. Since 2003, he has been responsible for the Electronic Master Program at ENIS. His research activities have been devoted to several topics: Design, Telecommunication, Embedded Systems, Information Technology, Video Coding and Image Processing.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.