Adaptive Multiple Transforms Hardware Architecture for Versatile Video Coding

T. Damak, S. Houidi, M. A. Ben Ayed, N. Masmoudi

***Abstract*—**The Versatile Video Coding standard (VVC) is actually under development by the Joint Video Exploration Team (or JVET). An Adaptive Multiple Transforms (AMT) approach was announced. It is based on different transform modules that provided an efficient coding. However, the AMT solution raises several issues especially regarding the complexity of the selected set of transforms. This can be an important issue, particularly for a future industrial adoption. This paper proposed an efficient hardware implementation of the most used transform in AMT approach: the DCT II. The developed circuit is adapted to different block sizes and can reach a minimum frequency of 192 MHz allowing an optimized execution time.

***Keywords*—**AMT, DCT II, hardware, transform, VVC.

I. Introduction

VIDEO data are the fastest growing data type on the Internet, and arguably one of the fastest growing periods. Therefore the video compression standards are evolved rapidly. The VVC [1] is the latest video compression standard under elaboration by the MPEG and the ITU. They have jointly launched the JVET to prepare the next generation of video coding standard. The VVC raises the video compression complexity algorithms in order to reach efficient compression ratio while maintaining the same video quality. One of the new algorithms introduced in the VVC codec is the AMT [2]. Despite the provided coding efficiency, the AMT algorithms raise several issues especially regarding the complexity of the selected set of transforms. Five different kinds of transform are applied depending on the selected prediction mode of the corresponding block.

A recent statistical study on the VVC complexity was done in [3]. This study presented the percentage of use of each transform type for all sizes. Different videos were tested in different QP cases. The results showed that Discrete Cosine Transform (DCT) II is the most used in the AMT module and for more than 99.1% of the cases the size of the transform unit (TU) had never exceeded 64. Based on this statistical study, we have elaborated all sizes of DCT II transform expect for size 128 mainly because of its lack of use and the challenged complexity of its implementation.

In this work, we present hardware architecture description of the block sized: 4, 8, 16, 32, and 64 of DCT II transform for future VVC standard.

The rest of this paper is organized as follows: Section II details the AMT module of the VVC. Section III is reserved to present the proposed hardware architecture of the DCT II module. Results of this implementation and comparison with state of the art are given by Section IV. Finally, a conclusion and perspectives are presented in Section V.

II. The AMT of VVC

Most of the transforms used in standardized video coding schemes belong to the Discrete Trigonometric Transform (DTT) family [4]. Amongst those, the DCT, especially DCT II, has received a considerable amount of attention in the ITU and MPEG transforms, and this is since MPEG-1/H.261.In HEVC [6], additional choices were introduced. In fact, the Discrete Sine Transform (DST) of type VII was adopted [5]. The innovation of the latest standard, the VVC, was the AMT transform. It mixes the DCT and the DST to explore efficiencies of both transforms. Five different equations of transforms are used depending on selected prediction mode of corresponding block. In addition to the DCT II and the DST VII transform used in HEVC, the three other transforms of VVC are: DCTVIII, DST I et DCT V. To select a transform set, the standard defined an algorithm of decision presented by Fig. 1.

III. Proposed Architecture

Based on the statistical study described in [3], the main objective of this work is to present optimized hardware architecture for DCT II transform adopted by VVC. The proposed DCT II architecture supports block sizes from 4 to 64, using multiplexers to select the desired size through the "Sel" input pin. As illustrated in Fig. 2, the global architecture of the proposed DCT II circuit is presented where the block size is selected to activate the corresponding component as:

- Sel = '000', component for bloc size = 4 is active.
- Sel = '001', component for bloc size = 8 is active.
- Sel = '010', component for bloc size = 16 is active.
- Sel = '011', component for bloc size = 32 is active.
- Sel = '100', component for bloc size = 64 is active.

In addition to the "Sel" input, the DCT II architecture reserves four inputs of 16 bits. These "Src_" inputs for our proposed circuit represent the residuals that are obtained from the difference between original pixels and predicted ones. Therefore, it is on 16 bits. For the output, they can attain 24 bits of size due to the consecutive shift operations. In fact, an optimization step is done before hardware implementation to replace multiplication operations by multiple shifts and

T. Damak, S. Houidi and M A. Ben Ayed were with the New Technologies and Telecommunication Systems Research Unit, ENET'COM, University of Sfax, Sfax, Tunisia (e-mail: damak.taheni@gmail.com).

N. Masmoudi was with the Electronics and Information Technology Laboratory, ENIS, University of Sfax, Sfax, Tunisia (e-mail: masmoudi123@ gmail.com).

additions.

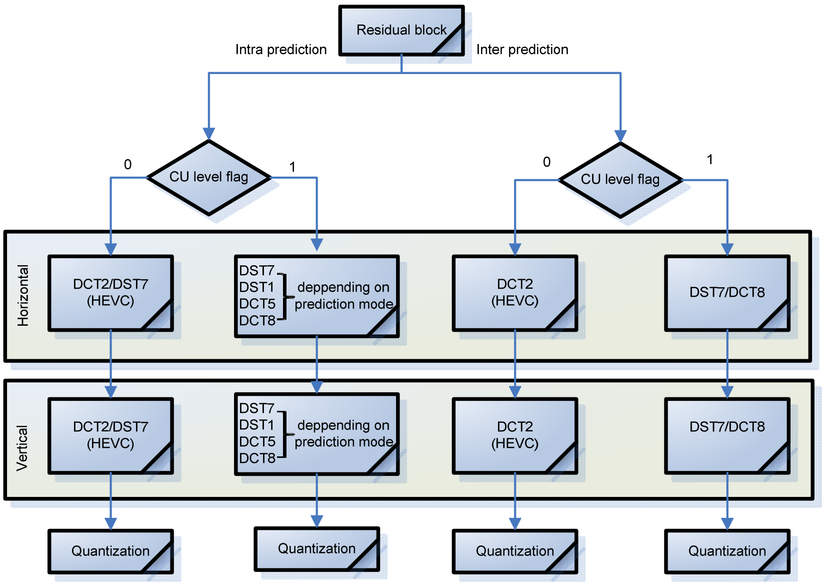

Fig. 1 The AMT transform selection process

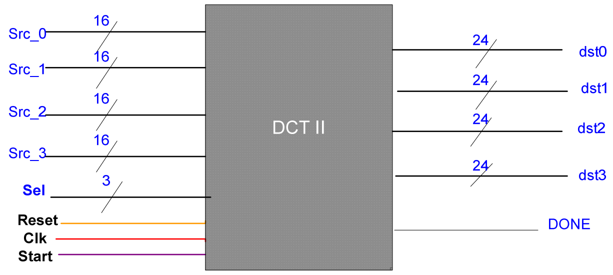

Fig. 2 The proposed DCT II entity

A root state machine illustrated by Fig. 3 manages the proposed DCT II architecture. For each “Sel” value, a specific process is applied depending on block size, as described before. For example, for the value “000” of “Sel” input, the ICT_4_1D component is active directly after receiving the four inputs since it represents the block size 4 that needs only four inputs to start. In opposing, the block size 8 component where Sel = '001', needs two set of four inputs. Therefore, it needs two cycles to start processing and two cycles after finishing transform calculation to generate the two sets of outputs.

The same logic is applied on 16, 32, and 64 block sizes that respectively use 4, 8, and 16 sets of inputs before starting ICT_16_1D, ICT_32_1D and ICT_64_1D components and respectively use also 4, 8, and 16 sets of outputs in order to get the overall output of DCT II component.

Each component used in the DCTII block has its own architecture. But, the operating strategy is the same based on the decomposition of the different matrices recursively. Fig. 4 and Fig. 5 present respectively the ICT_4_1D and ICT_32_1D components as an example to illustrate the internal architecture. In fact, the four source inputs are used to compute a butterfly step. This step prepares inputs coefficients to the decomposition of transform matrix into peer and odd matrix [7]. The first one, the peer part of transform, corresponds at the DCT II of the smaller block size (the block size 2 for the component of block size 4 in Fig. 4 and the block size 16 for the component of block size 16 in Fig. 5). The second part is the odd part of the transform. It is not recursive like peer part of matrix. It is calculated for each block size.

In order to clarify the proposed internal architecture and explain the proposed strategy for implementing different bloc of DCTs. Fig. 6 presents as an example, the odd part of the ICT_4_1D architecture. It computes the “dst_1” and the “dst_3” outputs from scr0, src1, src2 and src3 conforming to (1) and (2):

$$dst\_1= 334 * (src0-scr3)+139* (src1-scr2) \tag{1}$$

$$dst\_3= 139 * (src0-scr3)-334* (src1-scr2) \tag{2}$$

After separating Bloc_DCT2, Bloc_DCT4_odd is based on two constant multiplications and addition. The first constant multiplier is 334 and the second one is 139, as illustrated in (1) and (2). Consequently, a full general-purpose multiplier is not recommended for hardware implementation since it is expensive in terms of cycle time, energy consumption, and hardware resources.

In our case, a constant multiplier can be implemented using an appropriate sequence of additions and shifts operations that compose the desired constant. For example, since the value 334 is equal to 256+64+8+4+2 which is equal to 28+26+23+ 22+21, multiplying any input signal by 334 is equivalent to a multiplication by all power of two constancies that compose 334. This turns out to be just an addition of shifted version of that input signal. In the other words, multiplying any input signal by 334 will be equivalent to >> 8 + >> 7 + >> 3 + >> 2 + >> 1. The same strategy will be applied for the constant multiplier 139 which will be equivalent to: 139 → (>> 7 + >> 3 + >> 1 + 1). The shifts and additions replacing the operators of (1) and (2), are obviously illustrated in Fig. 6.

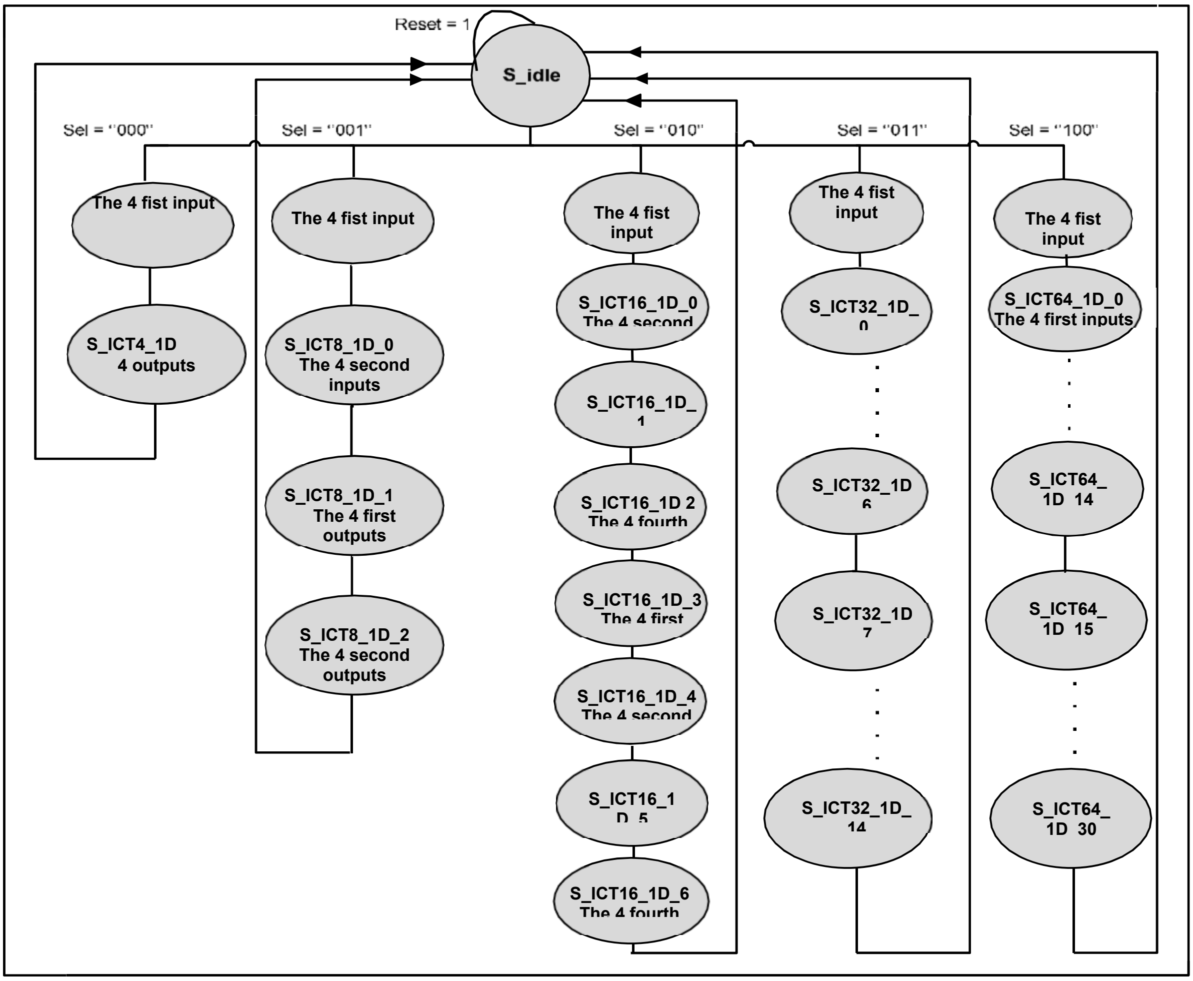

Fig. 3 The state machine of the proposed DCT II

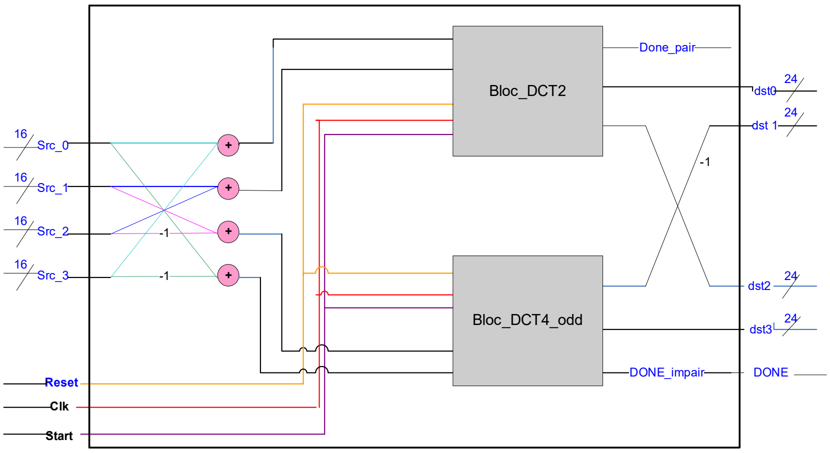

Fig. 4 ICT_4_1D architecture

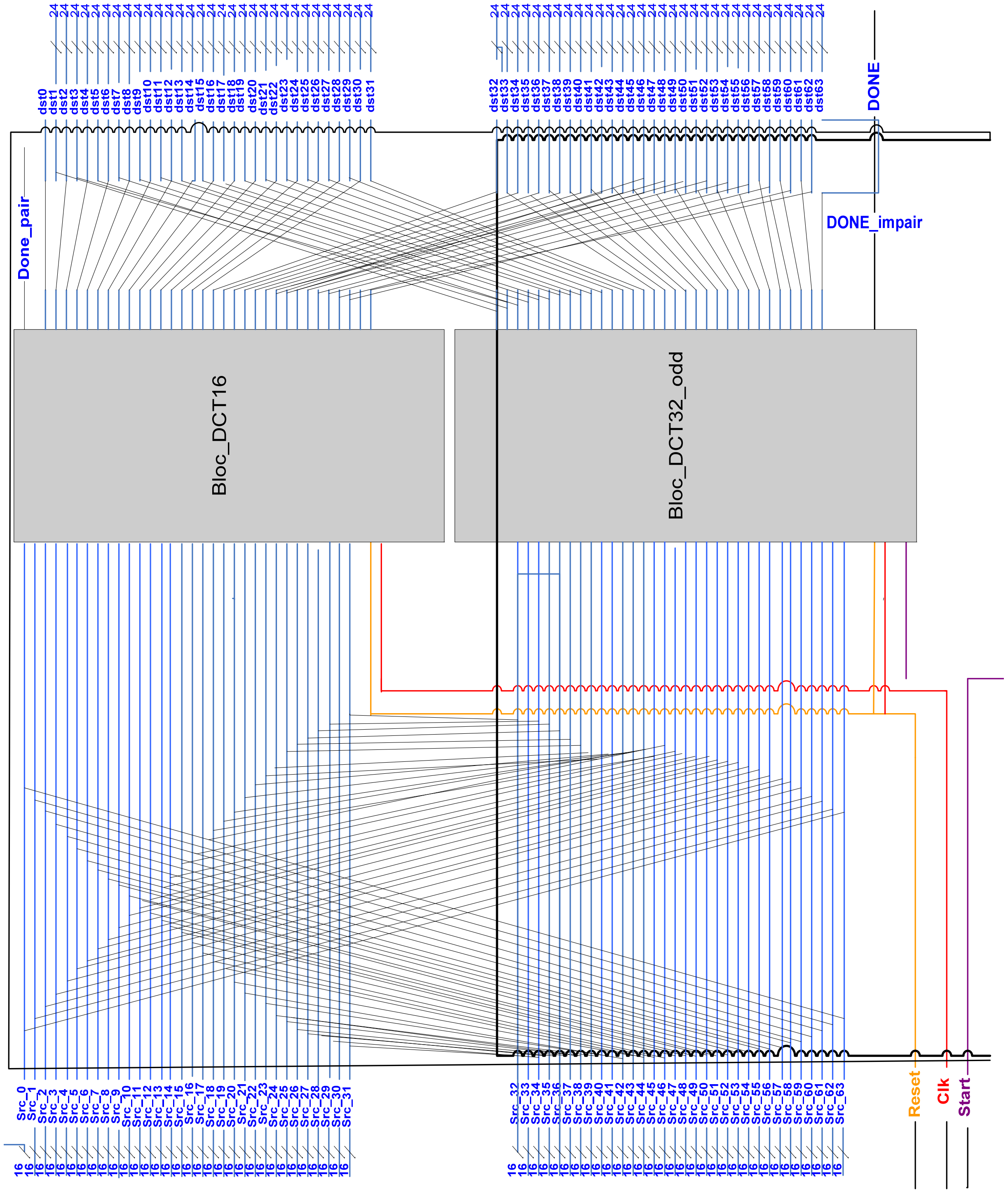

Fig. 5 ICT_32_1D architecture

IV. Results

The proposed work consists of an algorithmic optimization step and then a hardware implementation. The first step is the decomposition into peer and odd matrix for each block size. It consists of converting all DCT II equations to have a recursive matrix, which can be deducted directly from the smaller block size components, and to avoid multiplication operators. In fact, only additions and shifts are permitted to be implemented with hardware description (VHDL). Table I presents the number of operations in terms of multiplications, additions and shifts in the standardized DCT II equations and in the proposed architecture after optimization. As presented in Table I, the proposed architecture replaces multiplications by shifts. In addition, the total number of operation, for each block size, is better for the proposed architecture than the original algorithm of DCT II.

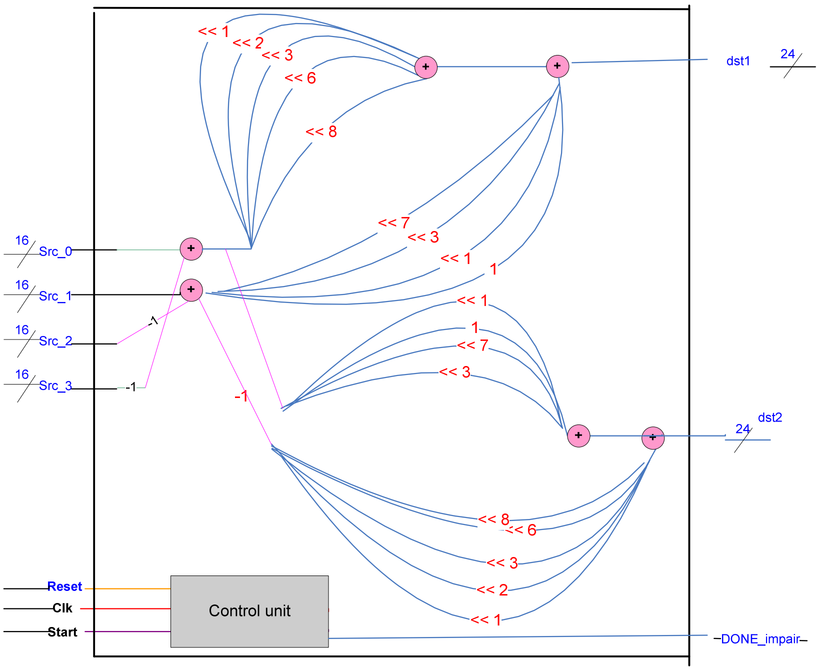

Fig. 6 The peer part of ICT_4_1D architecture: Bloc_DCT4_peer

TABLE I
THE NUMBER OF OPERATIONS IN DCT II EQUATIONS

	Number of operation in DCT II algorithm	Number of operation in proposed architecture
	ICT_4_1D	
Mult	16	0
Add	12	21
Shift	0	16
	ICT_8_1D	
Mult	64	0
Add	56	96
Shift	0	57
	ICT_16_1D	
Mult	256	0
Add	240	351
Shift	0	173
	ICT_32_1D	
Mult	1024	0
Add	992	1342
Shift	0	520
	ICT_364_1D	
Mult	4096	0
Add	4032	5053
Shift	0	1259

Table II shows the synthesis results of the proposed DCT II circuit under Stratix-III EP3SL150F1152C2 [8] FPGA device. The synthesis was done using Quartus II 9.0 [9], and the temporal simulation was verified via ModelSim 6.4a [10]. The occupied area represents only 49% from the total FPGA area. The temporal simulation revealed operation at a period of T equal to 5.2 ns which means a frequency which can reach 192 MHz.

TABLE II
PROPOSED DCT II SYNTHESIS RESULTS

Target	EP3SL340H1152C2
Total Pins	167/ 744 (22%)
Combinational ALUTs	133,279 / 270,400 (49%)
Dedicated Logic register	81,062 / 113,600 (30%)
Total block Memory bits	0 / 16,662,528 (0%)
Frequency (Quartus)	214,41 MHZ
Frequency (temporal simulation)	192 MHZ

TABLE III
TIME EXECUTION OF PROPOSED DCT II

Block size	Number of cycles for one vector input
4	5
8	14
16	19
32	37
64	52

In terms of execution time, Table III presents the number of cycles for each block size. The presented number of cycle includes the input vector reading time and the output vector generation time.

In order to evaluate the proposed architecture, a comparison with a similar implementation is presented [11]. Only the bloc size 4 and 8 components are presented in Table IV because other block sizes are not treated by [11]. For both sizes, our implementation is time saving.

TABLE IV
COMPARISON WITH EXISTING WORK

Block size	Used operators	[11]	Proposed Architecture
4	Add	44	21
	Shift	40	16
8	Add	392	96
	Shift	304	57

V.CONCLUSION

In this work, a hardware implantation is done on Startix-III FPGA. The architecture describes a DCT II transform for the future VVC standard. The architecture supports block size going from 4 to 64. The occupied area was 49% of the overall PFGA area and the minimum provided frequency is 192 Mhz. For future works, the other AMT transform modules will be implemented via hardware description to overcome software complexity.

REFERENCES

[1] Jianle Chen, Ying Chen, Marta Karczewicz, Xiang. Li, Hongbin Liu, Li Zhang, Xin Zhao "Coding tools investigation for next generation video coding based on HEVC" , in Proc. Of SPIE Vol. 9599 95991B-1, Septemb2015.

[2] AdriàArrufat"Multipletransformsforvideo coding", in Thesis defended on 11 December 2015, INSA Rennes, the European University of Brittany, France.

[3] Jallouli.A, Belghith.F,BenAyed.M ,Hamidouche.W, Nezan.J, Masmoudi.N," Statistical analysis of Post-HEVC encoded videos" in IEEE International Workshop on Signal Processing Systems (SiPS), France, october 2017.

[4] Thibaud Biatek, Victorien Lorcy, Pierre Castel, Pierrick Philippe,"Low-Complexity Adaptive Multiple Transforms forpost-HEVC Video Coding", in Picture Coding Symposium (PCS), 4-7 Dec. 2016, Germany.

[5] X Zhao, J Chen, M Karczewicz, L Zhang, X Li and W JungChien "Enhanced Multiple Transform for Video Coding", in 2016 Data Compression Conf.

[6] ITU-T Recommendation H.265 and ISO/IEC, MPEG-H Part 2, High Efficiency Video Coding (HEVC), 2013.

[7] Ahmed Kammoun, FatmaBelghith, HassenLoukil, NouriMasmoudi, "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 November 2016.

[8] https://www.mouser.fr/new/altera/altera-stratixiii; Accessed November 2018.

[9] http ://www.Quartus; Accessed November 2018.

[10] D. Chillet et E. Casseau, "TUTORIAL ModelSim VHDL", in Technopole Anticipa Lannion Rennes university, National School of Applied Science and Technology, November 2008.

[11] Ahmet Can Mert, ErcanKalali, IlkerHamzaoglu, Senior Member, IEEE, "High Performance 2D Transform Hardware for Future Video Coding", in IEEE Transactions on Consumer Electronics, Vol. 63, No. 2, May 2017, P117-125.