Hardware Design and Implementation of Adaptive Multiple Transforms for the Future Video Coding Standard

Ahmed Kammoun, Wassim Hamidouche, Fatma Belghith, Jean-François Nezan, and Nouri Masmoudi

***Abstract*—Future Video Coding (FVC) is the potential next generation video coding standard expected by the end of 2020. Many new contributions have led to better coding efficiency than the High Efficiency Video Coding (HEVC) standard. One of the new tools is the Adaptive Multiple Transform (AMT) as a new approach of the transform core design. The AMT involves five DCT/DST transform types with larger and more flexible partitioning block sizes. The AMT coding efficiency comes with the cost of much higher computational complexity, especially at the encoder side. In this paper, a high performance pipelined hardware implementation of the AMT transform types for 4x4, 8x8, 16x16 and 32x32 sizes is proposed. The architecture design takes advantage of the internal software/hardware resources such as Library of Parametrized Modules (LPM) core IPs and Digital Signal Processing (DSP) blocks of the target FPGA device. The proposed 1D 32-point AMT design is able to process 4K video at 44 frames per second. A unified 2D implementation of the 4, 8, 16 and 32-point AMT process is also presented. It takes into account all the asymmetric 2D block size combinations from 4 to 32. The 2D architecture design is able to sustain 2K video coding at 50 frames per second with an operational frequency up to 147 Mhz.**

***Index Terms*—Future Video Coding, Hardware Implementation, FPGA, Adaptive Multiple Transform, Pipeline, DSP.**

I. Introduction

THE immersive and realistic visual experience in consumer electronic devices (mobile phones, tablets, virtual reality helmets,...) are made possible by the interaction with higher resolution (4K, 8K), 360° videos and High Dynamic Range (HDR) [1] contents. To ensure an efficient storage and delivery of these emerging contents, the latest video coding standard High Efficiency Video Coding (HEVC) released by the Joint Collaborative Team on Video Coding (JCTVC) in early 2013 [2] enables to reduce the bitrate by 50% [3], [4] compared to its predecessor Advanced Video Coding (AVC) standard [5]. To further increase the coding efficiency, the Joint Video Exploration Team (JVET) [6] has launched a Call for Proposals (CFP) on video compression in order to develop the Future Video Coding (FVC) standard with coding performance beyond HEVC. The FVC standard is expected by the end of 2020 [7]. The JVET has first developed the Joint Exploration Model (JEM)software to test the gain of the new coding tools and show the evidence of developing a new video coding standard. The new coding tools in the JEM enable to increase the coding efficiency by 30% compared to HEVC [8]. This gain is the sum of several improvements in the coding chain modules including the transformation process which is one of the key tools of the hybrid codec. A new approach called Adaptive Multiple Transform (AMT) is introduced involving four additional transform types of Discrete Cosine Transform (DCT)/Discrete Sine Transform (DST) family [9], [10].

This coding efficiency is reached at the expense of higher complexity of up to 10x compared to HEVC [11], [12] at both encoder and decoder in inter coding configurations. This complexity increase is one of the main challenge for the development of the FVC, especially for real time implementations on embedded platforms. On the other hand, the hardware implementations are meant to provide some performance accelerations but under the constraints of their resources availability.

In this scenario, the embedded platforms are also witnessing a great progress. Recently, the new created advanced Field-Programmable Gate Array (FPGA) chips enable the implementation of Systems on Chips (SoC) designs. These devices are available for Low End (LE) [13], Middle End (ME) [14] and High End (HE) [15] applications. They are equipped with many soft and hard improvements to make them more adequate for applications requiring high memory and computation resources, such as high resolution video processing. The hybrid platform will enable to perform the sequential video encoding/decoding operations mainly the entropy engine on the software part while the transforms are accelerated on the FPGA part.

Only few works in literature have investigated hardware implementation of the AMT. These works are restricted either to blocks of size 4x4 [16], 8x8 [17] or 1D transform [18] process, due to its high complexity level. In this paper we propose a unified 2D hardware implementation of the AMT on a ME SoC platform. The main contributions of this work are the following :

1) The proposed design methodology takes into account the hardware resources of the target SoC FPGA platform which provides a large number of Digital Signal

Ahmed Kammoun, Wassim Hamidouche and Jean-François Nezan are with INSA Rennes, Institute of Electronic and Telecommunication of Rennes (IETR), CNRS - UMR 6164, VAADER team, 20 Avenue des Buttes de Coesmes, 35708 Rennes, France (E-mails: Ahmed.Kammoun@insa-rennes.fr, whamidou@insa-rennes.fr and Jean-Francois.Nezan@insa-rennes.fr)

Fatma Belghith and Nouri Masmoudi are with Univ Sfax, ENIS, Laboratory of Electronics and Information Technology (LETI), LR99ES37, Sfax Tunisia (E-mail: fatmabelghithenis@gmail.com, Nouri.Masmoudi@enis.rnu.tn)

Manuscript submitted on April 5, 2018.

Processing (DSP)s and reconfigurable multipliers Intellectual Property (IP) Cores, aiming to reduce the logic utilization.

2) A pipelined 1D hardware implementation of the AMT core supporting 4, 8, 16 and 32-point sizes with better performance than those obtained in [17], [18]. It can process HD (1920x1080) and UHD (3840x2160) coding video at 174 frames per second (fps) and 44 fps, respectively.
3) A unified 2D architecture embeds all 1D 4x4, 8x8, 16x16 and 32x32 transform modules and takes into account all asymmetric 2D block size combinations. The design is able to perform 2Kp50 fps video coding.

The rest of this paper is organized as follows. Section II presents a background on the AMT core design and the state of the art on its hardware implementations. In Section III, a brief description of the FPGA target device is given, followed by the detailed hardware implementation approach of the 1D and 2D AMT. The experimental and synthesis results of 1D and 2D implementations are then presented and discussed in Section IV. A comparison with other proposed works is also investigated in this section. Finally, Section V concludes this paper.

II. Related works

A. Background of The AMT Design

The HEVC standard is based on the the well-known DCT type II (DCT-II) as the main transform function and the DST type VII (DST-VII) for Intra blocks of size 4x4. In the the JEM software, the use of trigonometric transforms has been extended with the AMT that includes DCT-II, DCT-V, DCT-VII, DST-I and DST-VII transforms. TABLE I shows the different transform basis functions of the DCT/DST types [10].

TABLE I
TRANSFORM BASIS FUNCTIONS OF DCT-II/V/VIII AND DST-I/VII

Transform Type	Basis function $T_i(j)$, i, j=0, 1,..., $N-1$
DCT-II	$T_i(j) = \omega_0 . \sqrt{\frac{2}{N}} . \cos\left(\frac{\pi . i . (2j+1)}{2N}\right)$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-V	$T_i(j) = \omega_0 . \omega_1 . \sqrt{\frac{2}{2N-1}} . \cos\left(\frac{2\pi . i . j}{2N-1}\right)$, where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$, $\omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} . \cos\left(\frac{\pi . t(2i+1) . (2j+1)}{4N+2}\right)$
DST-I	$T_i(j) = \sqrt{\frac{2}{N+1}} . \sin\left(\frac{\pi . (i+1) . (j+1)}{N+1}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} . \sin\left(\frac{\pi . (2i+1) . (j+1)}{2N+1}\right)$

The AMT algorithm is applied at the block level on intra and inter prediction residuals. A specific *CU-level* flag is added in the bitstream to signal whether single or multiple transforms are used. If the *CU-level* flag is equal to 0, the classic HEVC transforms (DCT-II and DST-VII) are applied, otherwise two additional flags are added to signal the horizontal and vertical transforms used for the current Coding Unit (CU) [10].

For Intra prediction mode, an intra mode-dependent transform candidate selection is applied. According to the selected intra mode, a transform subset is identified as presented in TABLE II.

TABLE II
PRE-DEFINED TRANSFORM CANDIDATE SUBSETS

Transform Set	Transform Candidates
0	DST-VII, DCT-VIII
1	DST-VII, DST-I
2	DST-VII, DCT-V

For inter prediction, DST-VII and DCT-VIII can be used for horizontal and vertical transforms. For both Inter and Intra CU blocks, the JEM encoder encodes with all transforms within the selected subset and then chooses the one that minimizes the rate distortion cost. Related to their magnitude characteristics, the combinations of these transform types contribute efficiently and improve the flexibility of the transform design [19]. However, the fact that five transform types will be excessively evaluated for each CU, comes with the cost of higher computation complexity. This can be an issue for real time implementation.

The AMT involves 2D separable transforms enabling to perform 1D horizontal transform and then 1D vertical transform separably. For the MxN input block B, the 1D horizontal transform of the M rows of B is computed in equation (1)

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

where T_H is the NxN 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 in equation (2) between the intermediate output coefficients (Y_{int}) and the matrix of the vertical transform coefficients T_V of size MxM.

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

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

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

B. Hardware Transform Implementation

Several DCT-II hardware implementations have been proposed in the literature as it is the classic transform used in the previous video coding standards.

Paramud et al. [20] presented an efficient and reusable architectures for the implementation of DCT-II for different lengths using constant matrix multiplication. Moreover, the

proposed architecture can be pruned to reduce the complexity of implementation substantially with only a marginal effect on the coding performance for both folded and full-parallel 2-D DCT-II implementations.

Ahmed et al. [21] propose a dynamic N-point DCT-II for HEVC inverse transform of sizes 4x4, 8x8, 16x16 and 32x32. The hardware architecture is partially folded in order to save the area and improve the speed up of the design. The proposed architecture reaches as maximum frequency of 150 MHz which enables to support real time of 1080p30 video coding.

Chen et al. [22] proposed a 2D hardware implementation of the HEVC DCT transform. The reconfigurable architecture supports all block sizes from 4x4 up to 32x32. It benefits from several hardware resources such as DSP blocks, multipliers and memory blocks to reduce the logic utilization. Their proposed architecture has been synthesized in various FPGA platforms. Synthesis results show that the design could sustain 4Kp30 fps video encoding with reduced hardware cost.

Recently, new works on hardware implementation of the AMT have been published. Mert et al. [17] propose a 2D implementation of AMT including all types for 4x4 and 8x8 sizes by applying two 1D process using adders and shifts instead of multiplication operations. Two hardware methods have been provided. The first ones uses separate datapaths and the second method considers two reconfigurable datapaths for all 1D transforms. Although this work presents 2D hardware implementation of all transform types, it only supports 4x4 and 8x8 block sizes. Knowing that the transform of larger block size (16x16 and 32x32) is more complex and would require higher resources. M.J Garrido et al. propose in [18] a pipelined 1D hardware implementation of the AMT of all block sizes from 4x4 to 32x32. The design has been synthesized for different FPGA chips using multiple Read Only Memory (ROM) blocks to store the matrices of transform coefficients. The synthesis results show that the design can support 2K and 4K video processing with low hardware resources. Although the work proposed in [18] supports all block sizes, it only deals with 1D AMT design. The transform process consists of 2D operations which could normally be more complex. Moreover, this design does not consider the new feature of the AMT of asymmetric block sizes.

This paper proposes a unified and optimized 2D hardware implementation of the AMT using the IP Cores multipliers with the DSPs of the FPGA device. Up to the best of our knowledge, this is the first 2D hardware implementation of the AMT core supporting block sizes from 4x4 to 32x32 and taking into account all the asymmetric block size combinations.

III. The Proposed Hardware Implementation of 2D

In this section a brief description of the target embedded platform is given and then, the proposed design for both 1D and 2D AMT are described in details.

A. The target FPGA SOC device

It is one of the 10^{th} FPGA generation products launched after the union of two FPGA and Geforce Partner Program (GPP) leading manufacturers. Compared to the previous generation family products, several hardware and software improvements have been introduced. As a 20 nm technology platform, it is included in the middle range SoC devices, able to provide the desired high performance while keeping a low energy consumption and an acceptable cost.

Combined with its development kit, it presents a hybrid hardware/software platform that guarantees a faster path to commercialization. It can thus be a good choice for high resolution video processing. In this work, the aim is to benefit from its enhanced hardware features as the most important ones can be mentioned:

- Enhanced FPGA block that can handle more than 500 Mhz frequency performance.
- Large number of DSP blocks (up to 1687) and multipliers (up to 3376). These blocks can perform several constant multiplications between proper constant value as inputs. With a computing capacity of up to 1.5 G Floating-point Operation Per Second (FLOPS), they are dedicated to intensive computational applications.
- Low power consumption with up to 40% lower than previous generation devices.

B. 1D-AMT Hardware implementation

1) 4-point AMT implementation:

- Logic Model

The 4-point 1D-AMT design is summarized in TABLE III. A *start* positive pulse launches the operation while the transform type is set by the *selection* input.

TABLE III
4-POINT 1D INTERFACE DESCRIPTION DESIGN

Signal	I/O	Bits	Description
clk	I	1	Clock system
reset	I	1	Active low
start	I	1	Positive pulse
selection	I	3	Transform types: 0: DCT-II, 1:DST-I, 2: DST-VII, 3: DCT-VIII, 4:DCT-V
src0 .. src3	I	64	Input vector, 4 16 bit inputs
dst0 .. dst3	O	104	Output vector, 4 26 bit outputs
done	O	1	Qualifies output, active high

The input data is provided at a column basis with the start pulse. Four 16-bit inputs must be provided simultaneously. After the design process, the output values are assigned to *dst0 ..dst3* as shown in Fig. 1. Finally, the *done* signal indicates that the outputs are available.

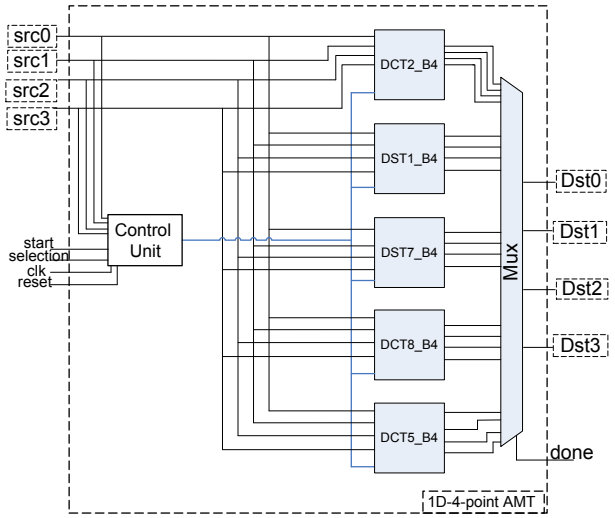

Fig. 1. Proposed 1D 4-point architecture design

- Proposed 4-point AMT architecture

For the DCT-II and DST-I transform types, some preliminary decompositions using efficient butterfly structure are applied in order to reduce the computational complexity of their design as shown in Fig. 2 and Fig. 3.

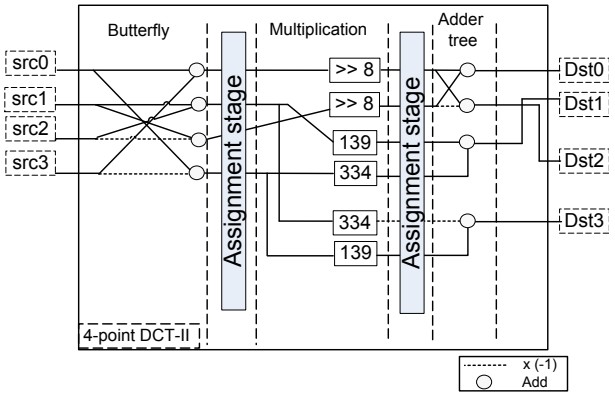

Fig. 2. Proposed 1D 4-point DCT-II architecture (dotted line refers to inverse sign value and add to addition operation)

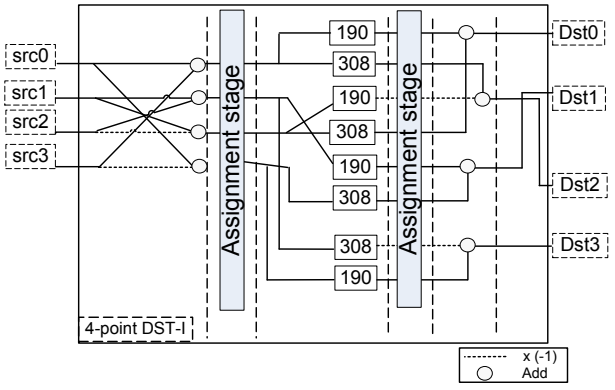

Fig. 3. Proposed 1D 4-point DST-I architecture

After the butterfly stage, all multiplication operations required are performed in parallel at once using the Library of Parametrized Modules (LPM) multipliers [23] of the target platform. The constant values mentioned in Fig. 2 - 6 refer to the multiplication matrix coefficients of each transform type involved in the AMT. Fig. 4 presents a Register Transfer Level (RTL) scheme of the DCT-II (Fig. 2) multiplication stage. LPM instances (green blocks) and shift gates (blue blocks) with appropriate coefficients are placed in parallel to perform multiplication operations.

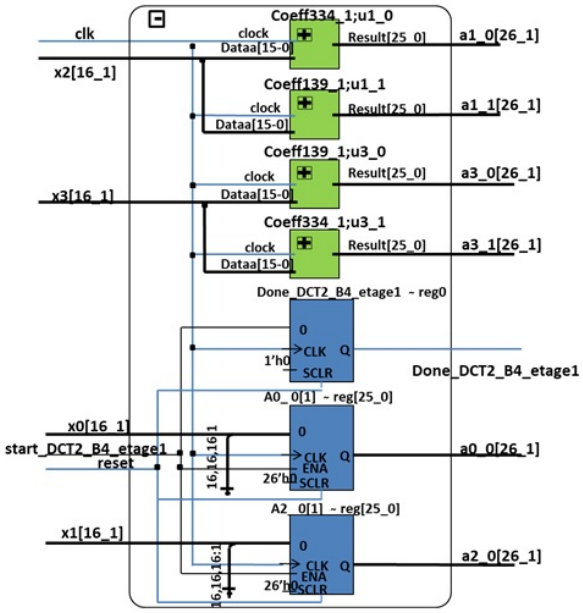

Fig. 4. RTL scheme of the DCT-II multiplication stage; green for LPMs and blue for Shift operators

Finally, an adder tree is applied to provide the 1D four outputs. The dotted vertical line separating two stages is equivalent to a clock cycle in the processing operation. Butterfly decomposition structures can not be applied for the other transform types. Thus, they are computed as forward matrices multiplications. Fig. 5 and Fig. 6 illustrate the proposed architectures for DST-VII and DCT-V, respectively.

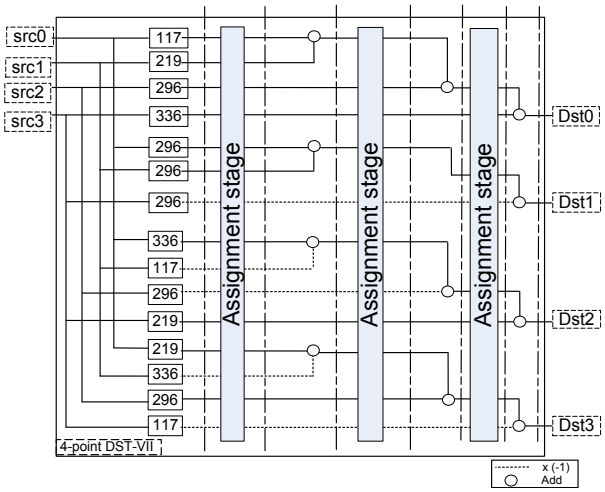

Fig. 5. Proposed 1D 4-point DST-VII architecture

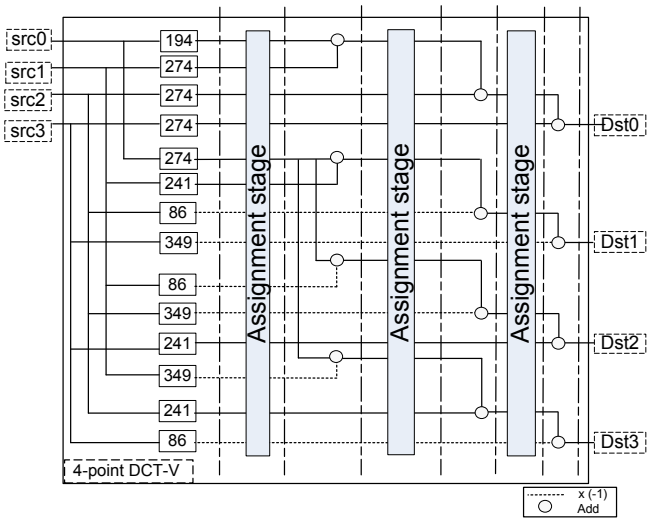

Fig. 6. Proposed 1D 4-point DCT-V architecture

Internal LPMs are used for all required multiplications in parallel. Then, three adder tree stages are placed successively in order to obtain the final outputs.

Compared to the DST-VII matrix, DCT-VIII one has the same coefficients but in inverse order for each row. Therefore, we only inverse the inputs order and assign the appropriate coefficients signs to easily benefit from DST-VII architecture, illustrated in Fig. 5, to implement the DCT-VIII transform type.

- Pipelined architecture design

In order to increase the design performance, the different architectures have been pipelined. Highlighted assignment stage components, as shown in Figures 2, 3, 5 and 6, are added after multiplication stage and also between every two adder tree stages. They are based on registers to store the current results and transfer them to the next stage avoiding data conflicts or loss which may occur in the next clock cycles as inputs are always changing. Fig. 7 shows a timeline presenting a 4x4 block pipeline processing.

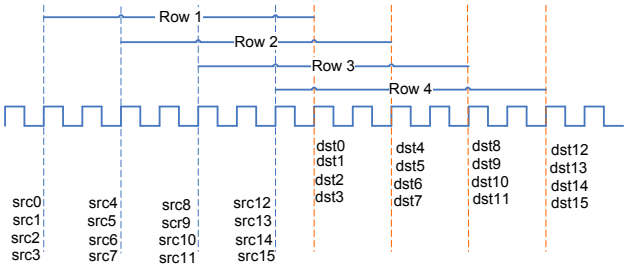

Fig. 7. Timeline for 4x4 block pipeline processing

Each assignment stage introduces one additional cycle to the latency providing the first four outputs. From that, within every two cycles, another four outputs are provided. TABLE IV shows the clock cycles required to compute the first outputs of each transform type. Of course, computing more rows in parallel would increase the performance enabled by the pipeline. In general, we can calculate the clock cycles (C_{Cycles}) required to compute M inputs rows by equation (4).

$$C_{Cycles} = L + (M - 1) . \Delta \tag{4}$$

where L is the number of cycles required to provide the first outputs (latency) and Δ is the pipeline level which refers to the number of cycles required between two outputs. In the example illustrated in Fig. 6, $N = M = 4$, $L = 7$, $\Delta = 2$ and $C_{Cycles} = 13$.

TABLE IV
LATENCY (L) REQUIRED TO PROVIDE THE FIRST OUTPUTS

	DCT-II	DST-I	DST-VII	DCT-VIII	DCT-V
Clock cycles	5	5	7	7	7

2) N-point AMT implementation: For DCT-II and DST-I, as their operations are recursive, an N point 1D transform can be performed by applying two $N/2$-point 1D transforms with additional preprocessing. For the DST-I, the applied $N/2$-point is of type DST-VII as illustrated in Fig. 8. DCT-V and DST-VII do not have the recursivity property. Therefore, they are implemented with matrices multiplications using the LPM multipliers IP Cores as the 4-point case. DCT-VIII transform type is always implemented using the DST-VII with appropriate changes of inputs order and signs.

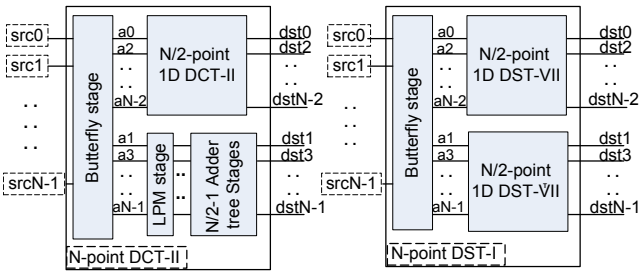

Fig. 8. Architectures of N-point DCT-II and DST-I

It is worth noting that for the 32-point implementation, pipeline is not adopted. This is justified by the fact that using the registers to ensure the pipeline stages for all the 32-point transform types together would require more logic utilization than the available one in the target platform. Instead, in order to preserve the clock cycles for 1D and 2D processes, adder trees were modified to operate two addition operations in one cycle. As a result, clock cycles required to provide 32-point outputs are reduced by half.

To summarize, the clock cycles required to implement one 1D outputs column considering the worst case type are 7, 15, 31 and 15 cycles for 4, 8, 16 and 32-point transforms, respectively. Considering MxN blocks, to calculate the required clock cycles, equation (4) is applied for 4x4, 8x8 and 16x16. For 32-point implementation it is equal to 15*32= 480 cycles since the 32-point transforms are not pipelined.

C. 2D-AMT implementation approach

Using its separable property, an (MxN)-point 2D AMT could be computed by the row-column decomposition technique in two distinct stages:

1) STAGE-1: N-point 1D AMT is computed for each column of the input matrix to generate an intermediate output (Y_{int}).
2) STAGE-2: M-point 1D AMT is computed for each row of the intermediate output matrix to generate desired 2D output.

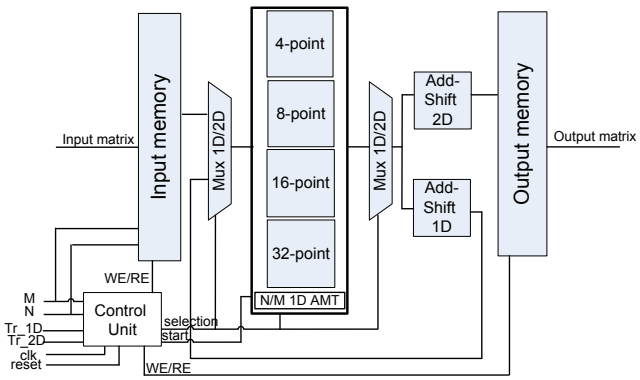

Fig. 9. Proposed 2D AMT architecture

Fig. 9 illustrates the proposed architecture for the 2D AMT approach. Depending on the two block size parameters MxN, the control unit uses the input memory to store the input data. A *start* signal is given to begin the 1D transform. If N = 4, 8 or 16, input columns are read from memory each two cycles within M *start* signals.

N-point transform module operates to provide the 1D outputs. The first output values are available after the latency required according to the transform order (N) and type as explained earlier in TABLE III and TABLE IV. At the next clock cycle, they are stored in temporary registers after the corresponding Add and Shift operations to be rounded and saturated to 16 bits. Once the first N outputs are available, within every two cycles, new outputs are obtained until reaching M rows. When N is equal to 32, *start* signal is given only if the corresponding outputs are available and stored due to the absence of pipeline for the 32-point case.

The final *done-N* signal indicates that 1D intermediate outputs are available and stored in the corresponding registers. Subsequently, the 2D transform process can begin. The 2D transform type is assigned and M-point transform module will operate. The 1D temporary outputs, transposed, will be the inputs of 2D process. The same 1D transform principle explained above is applied only with reversing M and N as block sizes may have asymmetric combinations. Finally, every 2D M-outputs are stored and displayed two by two via First In First Out (FIFO) memory blocks. Delivering and managing the WE/RE signals for the different memories and assigning the appropriate modules, all are guaranteed by a control unit according to a state machine.

IV. Experimental and Synthesis Results

A. Experimental setup

The proposed FVC 2D transform design is implemented using the Verilog HDL description language. The architectures of 1D and 2D processes of different orders have been tested with state of the art simulation and synthesis software tools [24], [25]. Test bench files and JEM4.0 reference vectors were used to validate the output results.

B. Synthesis results of 1D-AMT implementation

The objective is to implement the five AMT transform types with sizes up to 32. Therefore, even if the used platform offers a large number of DSP blocks, it will not obviously cover all the multiplication operations. The LPM multiplier cores IP [23] are characterized to be configurable either to use the default implementation via registers and Aluts or use dedicated circuitry i.e DSP blocks to preserve the logic utilization. With this property we can manage to customize the number of DSPs and avoid exceeding the available resources. All the synthesis are realized with the corresponding software tool [24] under the FPGA target device. TABLE V shows the synthesis results of 4 point module implementation and the DSPs usage of the design. Using only 3% of DSPs (42), logic utilization is reduced by about 30% (Alms & registers). The AMT module of larger size would increase the DSPs usage.

TABLE V
SYNTHESIS RESULTS OF THE PROPOSED 1D 4-POINT AMT DESIGN

	without DSPs	with DSPs
Pins	175	175
Alms	1915	1156
Registers	4597	3222
DSPs	0	42 (3%)
Frequence	550 MHz	532 MHz

Since the 32-point module is the most complex, the LPM multipliers required for the five transform types implementation are configured to use DSPs. However, 4, 8 and 16-point modules are implemented using the default implementation resources (without DSPs). Synthesis results of the 8 and 16 point modules are given in TABLE VI.

TABLE VI
SYNTHESIS RESULTS OF 1D 8 AND 16-POINT AMT DESIGNS

	1D 8-point	1D 16-point
Pins	343	679
Alms	9558	48982
Registers	25525	156328
DSPs	0	0
Frequency	537 MHz	414 MHz

The high number of used registers shown in TABLE VI is mainly due to two reasons: the first one is the use of registers enabling the pipeline through the assignment stages and the second one is the use of the default logic resources through LPMs multipliers. On the other hand, as shown in TABLE VIII, the absence of assignment stages i.e pipeline

TABLE VII
COMPARISON OF PROPOSED 1D AMT TRANSFORM DESIGNS WITH SOLUTION IN [18]

	4-point		8-point		16-point		32-point	
	[18]	Proposed	[18]	Proposed	[18]	Proposed	[18]	Proposed
Alms	501	1915	501	9558	501	48982	501	45865
DSPs	16	0	16	0	16	0	16	1561
Random Access Memory (RAM)	640 Kbit	0	640 Kbit	0	640 Kbit	0	640 Kbit	0
Freq	458	550	458	537	458	414	458	254
2K fps	585	217	294	381	146	559	72	174
4K fps	146	54	73	95	36	140	18	44

(as explained in section III) and benefiting from DSP blocks for the 32-point AMT module reduce the usage of hardware resources.

TABLE VIII
SYNTHESIS RESULTS OF THE PROPOSED 1D 32-POINT AMT DESIGN

Design	Pins	Alms	Registers	DSPs	Frequency
1D-AMT	72	45865	72425	1561	254 Mhz

The 32-point design is adjusted using FIFO memories to provide two by two 16-bit inputs and outputs in order to avoid pin assignment problem. As the DCT-II and DST-I have recursivity property, LPM multipliers of components from lower order modules are reconfigured to use the DSPs blocks in the 32-point implementation. To more evaluate all 1D implementation design performance, TABLE IX summarizes the frame rate in fps that can be processed for 2K and 4K video resolutions.

TABLE IX
PERFORMANCE OF 1D 4, 8, 16 AND 32-POINT DESIGNS

1D-AMT size	Cycles	Frequency	2K fps	4K fps
4-point	13	550 Mhz	217	54
8-point	29	537 Mhz	381	95
16-point	61	414 Mhz	559	140
32-point	480	254 Mhz	174	44

Square block sizes and worst cases are considered for all 1D AMT implementations to compute the frame rate in fps by equation (5).

$$framerate(fps) = (Freq\,.\,M\,.\,N)\,/\,(C_{Cycles}\,.\,Res.\,\frac{3}{2}) \quad (5)$$

where *Freq* is the required operational frequency, $M\,.\,N$ the size of the processed block, C_{Cycles} the clock cycles required for processing the block, *Res* the target video resolution and the term $\frac{3}{2}$ is a factor related to the image color sampling in 4:2:0.

We can notice from TABLE IX that the efficiency of 1D AMT implementation increases with larger block sizes. This is due to the proposed pipeline architecture that enables clock cycles preservation when higher rows are computed. The 16-point AMT design can support 2K and 4K videos at 559 and 140 fps, respectively.

On the other hand, even if the 1D 32-point module is not pipelined, it is still efficient enough to sustain real time coding with 174 and 44 fps for 2K and 4K video resolutions, respectively. This is justified by reducing the adder tree stages and using the internal LPM Cores and DSP blocks offered by the target device.

The proposed architecture offers better performance in terms of processed frame rate with respect to state of the art 1D AMT implementation [18]. For large block sizes 16x16 and 32x32, the proposed design is able to perform more than twice frame rate for 2K and 4K resolutions video as shown in TABLE VII. However, it is worth noting that in terms of logic utilization, the proposed design have higher resource consumption. The work in [18] benefits from RAM memory of 640 Kbit to preserve the logic cost. This would be an objective for our future works. Reducing the number of reserved registers and Aluts can allow the pipeline of the 32-AMT module and further enhance the speed performance.

C. Synthesis results of 2D- AMT implementation

The synthesis results of the unified 2D implementation (Section III-C) are presented in TABLE X. The design reaches an operational frequency of up to 147 Mhz using about 53% of the device logic resources and 93% of the available DSPs.

TABLE X
SYNTHESIS RESULTS OF THE UNIFIED 2D 4, 8, 16 AND 32-POINT AMT DESIGN

Design	Pins	Alms	Registers	DSPs	Frequency
2D-AMT	72	133017 (53%)	274902	1561 (93 %)	147 Mhz

The performance of the unified design is evaluated in TABLE XII. This table presents the frame rate in fps that can be processed for different 2D block size combinations computed using Equation (5). Cycles involved in transform types selection and in intermediate 1D outputs transposition are taken into account in the 2D clock cycles calculation. However, cycles reserved to storing the input data and displaying final 2D output data are not considered.

The proposed design enables high frame rate performance. It should be noted that the larger block size is, the better the results are as long as the pipeline is going deeper with more rows to compute. These numbers are obtained supposing the

TABLE XI
COMPARISON OF DIFFERENT 2D HARDWARE TRANSFORM DESIGNS

Solutions	[20]	[21]	[22]	[17]	[18]	Proposed
Technology	ASIC 90 nm	ASIC 90 nm	28 nm FPGA	40 nm FPGA	ME 20 nm FPGA	ME 20 nm FPGA
ALMs	–	–	–	5292	999	133017
DSPs	0	0	128	–	32	1561
Frequency (Mhz)	187	150	222	167	458	147
Frames/sec	7680x4320p60	1080x720p30	3840x2160p30	3840x2160p30	3840x2160p18	1920x1080p50
Max bit length	25	25	25	27	–	26
Transform unit	4x4, 8x8, 16x16, 32x32	4x4, 8x8, 16x16, 32x32	4x4, 8x8, 16x16, 32x32	4x4, 8x8	4x4, 8x8, 16x16, 32x32	4x4, 8x4, 16x4, 32x4,4x8, 8x8, 16x8, 32x8,4x16, 8x16, 16x16, 32x16,4x32, 8x32, 16x32, 32x32
Transform type	DCT-II	DCT-II	DCT-II	DCT-II, DST-I, DST-VII, DCT-VIII, DCT-V	DCT-II, DST-I, DST-VII, DCT-VIII, DCT-V	DCT-II, DST-I, DST-VII, DCT-VIII, DCT-V
Dimension	2D	2D	2D	2D	1D	2D

TABLE XII
PERFORMANCE OF UNIFIED 2D DESIGN

2D-AMT size	Cycles	2K fps	4K fps
4x4	30	25	7
8x8	62	49	12
16x16	126	96	24
32x32	964	50	13
32x16	337	72	18
16x8	94	64	16
8x32	201	60	15

same size for all transforms. However, in real applications, each frame is encoded with a mix of transform block sizes. Regarding this, the 2D design may have better performance. In addition, in future works, as we intend to reduce the high register number reserved for the pipeline process, the 32-point module can also be pipelined and the 2D design may work at higher operational frequency with less clock cycles.

A fair comparison with other works in literature is quite difficult. Most of works are focusing on the 2D-HEVC DCT-II. Works related to AMT hardware implementation adopt either 2D implementation up to only 8x8 block size [17] or only 1D implementation supporting square block sizes up to 32x32 [18]. TABLE XI summarizes the key parameters to compare the proposed unified design performance with state of the art works.

The proposal presents the union of 4, 8, 16 and 32-point transform modules. It also controls all possible combinations of not only block sizes which can be asymmetric but also transform types which differ from 1D and 2D processes. Furthermore, it manages the Input/Output memory blocks delivering the appropriate WE and RE signals depending on the block sizes. Finally, the whole process is managed by a definite state machine.

The 2D constraints obviously increase the complexity level and the critical paths for the synthesis results adding some internal delays. This may affect the performance in terms of area and frame rate and operational frequency. It is not the case for the 1D process where almost all these constraints do not interfere.

The first purpose of designing a unified circuit involving all 4, 8, 16 and 32-point transform types is preserving the area consumption on the target device. The second one which is more interesting is satisfying the asymmetric combinations of the processed unit size as one of transform core improvements provided by the FVC. Up to the best of our knowledge, this is the first 2D hardware implementation of AMT core supporting 4 up to 32-point transforms and that supports all 2D block sizes combinations.

V. CONCLUSION

In this paper we have proposed a unified 2D hardware implementation of the AMT for the FVC standard. A hardware implementation of 1D 4, 8, 16 and 32-point AMT modules using LPM multiplier core IPs and DSP blocks is presented. The 1D architecture design is able to perform 4K video coding at 44 frames per second. A unified 2D implementation of the AMT is also proposed in this work. This is the first 2D implementation design that takes into account all asymmetric block size combinations from 4 to 32. With an operational frequency of up to 147 Mhz, the unified 2D AMT design is able to sustain 2K video coding at 50 frames per second.

As future work, in order to reach higher performance, logic resources involved in pipeline process can be reduced to allow the pipeline of the 32-point design. As a result, higher operational frequency with less clock cycles can be achieved.

Even though the proposed hardware design is dedicated to the encoder, it can easily be extended to the decoder side by only transposing the transform matrices. Therefore, this solution can be embedded on many electronic devices performing real time video processing such as TVs, cameras, smartphones, virtual reality helmets and tablets.

REFERENCES

[1] Y. Liu, W. Hamidouche, O. Dï¿½forges, and F. Pescador, "A multi-modeling electro-optical transfer function for display and transmission of high dynamic range content," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 350–358, November 2017.

[2] H. I. Recommendation, "High Efficiency Video Coding (HEVC)," *23008-2 MPEG-H Part 2*, 2013.

[3] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards;Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol*, vol. 22, no. 12, pp. 1669–1684, Dec 2012.

[4] F. Pescador, M. Chavarrias, M. J. Garrido, E. Juarez, and C. Sanz, "Complexity analysis of an HEVC decoder based on a Digital signal processor," *IEEE Trans. Consum. Electron.*, vol. 59, no. 2, pp. 391–399, May 2013.

[5] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol*, vol. 13, no. 7, pp. 560–576, July 2003.

[6] Joint-Video-Exploration-Team, "Several jvet meetings," [Onine]. Available: https://jvet.hhi.fraunhofer.de/.

[7] "Joint Call for Proposal on Video Compression with Capability beyond HEVC," *MPEG document N17195, Joint Video Exploration Team (JVET) of ITU-T VCEG (Q6/16) and ISO/IEC MPEG (JTC 1/SC 29/WG 11)*, Oct. 2017.

[8] N. Sidaty, W. Hamidouche, O. Deforges, and P. Philippe, "Compression efficiency of the emerging video coding tools," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sept 2017, pp. 2996–3000.

[9] X. Zhao, J. Chen, M. Karczewicz, A. Said, and V. Seregin, "Joint Separable and Non-Separable Transforms for Next-Generation Video Coding," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2514–2525, May 2018.

[10] "Algorithm Description of Joint Exploration Test Model 7(JEM7)," *MPEG document N17055, Joint Video Exploration Team (JVET) of ITU-T VCEG (Q6/16) and ISO/IEC MPEG (JTC 1/SC 29/WG 11)*, July 2017.

[11] H. Schwarz, C. Rudat, M. Siekmann, B. Bross, D. Marpe, and T. Wiegand, "Coding Efficiency / Complexity Analysis of JEM 1.0 coding tools for the Random Access Configuration," in *Document JVET-B0044 3rd 2nd JVET Meeting: San Diego, CA, USA*, February 2016.

[12] E. Alshina, A. Alshin, K. Choi, and M. Park, "Performance of JEM 1 tools analysis," in *Document JVET-B0044 3rd 2nd JVET Meeting: San Diego, CA, USA*, February 2016.

[13] Cyclon-V-Device-Overview, "Intel 2016," [Onine]. Available: https://www.altera.com/documentation/sam1403480548153.html.

[14] Intel-Arria-10-Device-Overview, "Intel 2017," [Onine]. Available: https://www.altera.com/documentation/sam1403480274650.html.

[15] Intel-Stratix-10-GX/SX-Device-Overview, "Intel 2017," [Onine]. Available: https://www.altera.com/documentation/joc1442261161666.html.

[16] A. Kammoun, S. B. Jdidia, F. Belghith, W. Hamidouche, J. F. Nezan, and N. Masmoudi, "An Optimized Hardware Implementation of 4-point Adaptive Multiple Transform design for post-HEVC," in *International Conference on Advanced Technologies for Signal & Image Processing ATSIP'2018*, 2018.

[17] A. Mert, E. Kalali, and I.Hamzaoglu, "High Performance 2D Transform Hardware for Future Video Coding," *IEEE Trans. Consum. Electron.*, vol. 62, no. 2, May 2017.

[18] M. Garrido, F. Pescador, M.Chavarrias, P. Lobo, and C.Sanz, "A High Performance FPGA-based Architecture for Future Video Coding Adaptive Multiple Core Transform," *IEEE Trans. Consum. Electron.*, March 2018.

[19] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li, and W-J.Chien, "Enhanced multiple transform for video coding," *Data Compression Conference (DCC)*, pp. 73–82, March 2016.

[20] P. K. Meher, S. Y. Park, B. K. Mohanty, K. S. Lim, and C. Yeo, "Efficient Integer DCT Architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol*, vol. 24, no. 1, pp. 168–178, Jan 2014.

[21] A. Ahmed and M. Shahid, "N Point DCT VLSI Architecture for Emerging HEVC Standard," *VLSI Design*, pp. 1–13, 2012.

[22] M.Chen, Y. Zhang, and C. Lu, "Efficient architecture of variable size HEVC 2D-DCT for FPGA platforms," *International Journal of Electronics and Communications*, vol. 73, pp. 1–8, March 2017.

[23] Intel-FPGA-Integer-Arithmetic-IP-Cores-User-Guide, "Intel 2017," [Onine]. Available: https://www.altera.com/en_US/pdfs/literature/ug/ug_lpm_alt_mfug.pdf.

[24] Intel-FPGA-Download-Center, [Onine]. Available: https://www.altera.com/downloads/download-center.html.

[25] Mentor-ModelSim-Functional-Verification-Tool-web, [Onine]. Available: https://www.mentor.com/products/fv/modelsim.

Ahmed Kammoun was born in Tunis, Tunisia in 1992. He 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 PhD student. His research interests include video coding and compression, potential video coding standards and codecs, FPGA hardware implementation.

Wassim Hamidouche was born in Algiers, Algeria, in 1984. He received the Ph. D. Degree in Signal and Image Processing from the University of Poitiers, France in 2010. From 2011 to 2012 he has been a Research Engineer with Canon Research Centre, Rennes, France. He is Associate Professor at INSA Rennes since 2015 and member of the the Institute of Electronics and Telecommunications of Rennes (IETR), UMR CNRS 6164. His research interests focus on video coding, efficient real time and parallel architectures for the new generation video coding standards, multimedia transmission over heterogeneous networks, and multimedia content security.

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.

Jean-François NEZAN is a Professor at the Department of Electrical and Computer Engineering at the National Institute of Applied Sciences (INSA) and the Institute of Electronics and Telecommunications of Rennes (IETR).

He is coauthor or coeditor of more than 75 technical articles including 1 Book, 1 Book chapter, 16 publications in International Journals. He is involved in the French research society *GDR ISIS* and the European Network of Excellence HiPEAC. His research topic is the rapid prototyping of standard video compression on embedded architectures including signal processing systems, architectures, and software; hardware/software co-design; and fast prototyping tools.

Nouri Masmoudi was born in Sfax, Tunisia in 1955. He 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.