World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:7, No:6, 2013

Efficient Hardware Architecture of the Direct 2-D Transform for the HEVC Standard

Fatma Belghith, Hassen Loukil, and Nouri Masmoudi

***Abstract*—**This paper presents the hardware design of a unified architecture to compute the 4x4, 8x8 and 16x16 efficient two-dimensional (2-D) transform for the HEVC standard. This architecture is based on fast integer transform algorithms. It is designed only with adders and shifts in order to reduce the hardware cost significantly. The goal is to ensure the maximum circuit reuse during the computing while saving 40% for the number of operations. The architecture is developed using FIFOs to compute the second dimension. The proposed hardware was implemented in VHDL. The VHDL RTL code works at 240 MHZ in an Altera Stratix III FPGA. The number of cycles in this architecture varies from 33 in 4-point-2D-DCT to 172 when the 16-point-2D-DCT is computed. Results show frequency improvements reaching 96% when compared to an architecture described as the direct transcription of the algorithm.

***Keywords*—**HEVC, Modified Integer Transform, FPGA.

I. Introduction and Previous Works

NOWADAYS, different video standards are widely used in different applications in order to maximize compression capability. Different video standards are developed nowadays such as H.264/AVC [1]. This standard have had a particularly impact into many applications. In order to enhance the coding efficiency, the joint collaboration team on the video coding (JCT-VC) is working on a new video coding standard, known as high efficient video coding (HEVC) [2]. This standard uses the same hybrid approach (inter/intra picture) presented in all previous standards since H.261. The encoding process producing an HEVC congruent bitstream would generally proceed as the Fig. 1 shows.

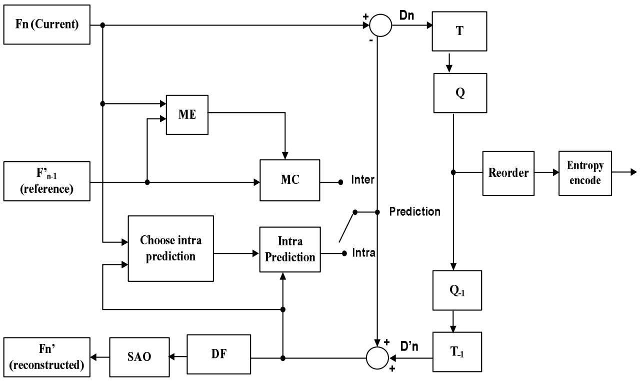

Fig. 1 Typical HEVC video encoder

Each picture is divided into block-shaped regions with the exact block partitioning being transmitted to the decoder. The first picture of a video sequence is always coded using intra-prediction. For the others both intra and inter may be used in prediction. The residual signal which is the difference between the original block and the predicted one is transformed, scaled then quantized and finally entropy coded in order to be transmitted with the prediction information [3]. Concerning the transforming part, two dimensional transforms are computed by applying one-dimensional transforms in both the horizontal and vertical directions. As the first edition of the HEVC standard was expected to be finalized in January 2013, it was important to start exploring the hardware implementation of the transform unit. In recent years, some transform architectures have been proposed for video applications. Both, inverse and direct transforms were considered.

In [4], authors have implemented a fast 16-point DCT using a multiplier-less architecture. The VLSI implementation has been carried out for a 90-nm standard cell technology at a clock frequency of 150 MHZ.

Ricardo's work presented in [5] proposed architecture synthesized using Cyclone II and Stratix III reducing 72% compared to the original architecture.

Martuza's work in [6] proposed unified hybrid architecture to compute the 8x8 integer inverse discrete cosine transform of multiple video codec implemented on FPGA and synthesized in CMOS 0.18 um technology.

In another interesting design [7], Jong Sik proposed 16x16 and 32x32 inverse transform architecture for HEVC based on the hardware reuse. The implementation of the proposed 2D

Fatma Belghith, Hassen Loukil, and Nouri Masmoudi are with University of Sfax, National Engineering School of Sfax, Electronics and Information Technology Laboratory. BP W 3038 Sfax, TUNISIA (e-mail:Fatmabelghithenis@gmail.com).

architecture in 0.18 um technology shows about 300 MHZ frequency.

In this paper, the hardware architecture of the 2D transform was developed for the highly anticipated HEVC standard. The proposed design can be applied to both direct and inverse transformation; however, only the implementation for the direct process was presented. The purpose of this work is the maximum reuse of the matrices using a unified architecture of a 4x4, 8x8 and 16x16 for the two dimensional transformation. This paper is organized as follows. Section II describes the whole architecture of the two dimensional transformation. The hardware implementation and the evaluation results are presented in Section III. Section IV presents conclusions.

II. 2D Transform Architecture

In our previous work [9], a modified transform was presented providing less complexity with comparable quality.

The idea consisted on decomposing the 16x16 transform in two parts: an even part which is the 8x8 and the odd part decomposed also into smaller matrices using only sums and shifts. Concerning the decomposition of the 8x8 matrix, it was divided into the matrix 4x4 which is also decomposed to multiplierless operations. The other part of the 8x8 transform is also decomposed into matrices using only adders and shifts.

The proposed algorithm is suitable especially for low-cost hardware implementation.

The block diagram of the proposed hardware architecture is shown in Fig.2. This architecture contains only four inputs and four outputs, in addition to a clock, a reset and a selection input. Using a multiplexer, which manages the three different transformations. This diagram is composed by two basic modules. The first computes the one-dimensional transform for the three sizes (4x4, 8x8 and 16x16). The second concerns the use of FIFOs in order to compute the two-dimensional transforms.

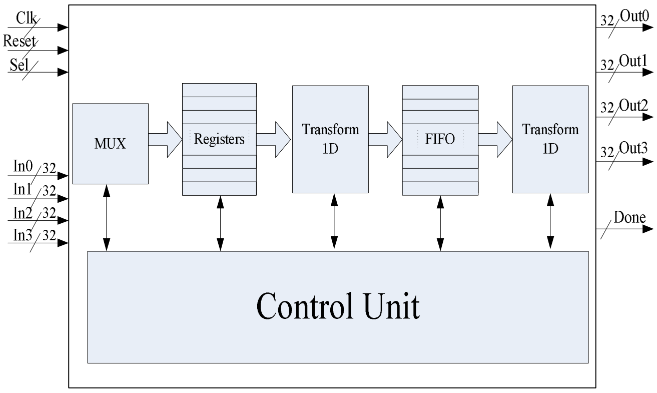

Fig. 2 Block Diagram of the Proposed Architecture

Once the size of the transform unit is chosen, the inputs are read or stored in registers four by four. Whatever the size of the transform, a unique architecture is used containing always 16 inputs and 16 outputs. Outputs of this one-dimensional transform are stored in FIFOs in order to be later considered as the inputs for the second dimension. Using the same architecture, the second dimension is computed. Outputs are also written four by four. The number of reading and writing cycles changes according to the size of the transform unit.

A. Transform 1D Architecture

This unit computes the one-dimensional transform which is applied in the horizontal direction then in the vertical direction.

The choice of the number of inputs and outputs was done considering that the transform 16x16 is the largest implemented.

This block permits to compute one of the three transforms. The three different transforms can be computed using this unified architecture and it takes 12 cycles to compute them regardless the size of the transform.

To compute the 4-point-DCT, we should consider only the first 4th inputs (Src0, Src1, Src2, Src3) and the others are null and the outputs will be (Dst0, Dst4, Dst8, Dst12).The block U1 used to compute this unit needs 10 sums and 12 shifts. Concerning the 8-point-DCT, only 8 inputs (Src0, Src1, Src2, Src3, Src4, Src5, Src6, Src7) are considered and the outputs will be (Dst0, Dst2, Dst4, Dst6, Dst8, Dst10, Dst12, Dst14). To get all these outputs, K1,K2, and K3 were used. These blocks were designed only with adders and shifts [9]. To compute 16-point-DCT, we have 16 inputs and 16 outputs done through the subblocks M1,M, M3 and a shift module. These blocks are also designed using only adders and shifts. To resume, Table I summarizes the number of cycles needed for the three sizes which varies from 14 to 20 cycles when the transform unit 16 is chosen.

TABLE I
Execution Time for the Three One-Dimensional Transforms

Cycles	
4-point-1D-DCT	14 cycles
8-point-1D-DCT	16 cycles
16-point-1D-DCT	20 cycles

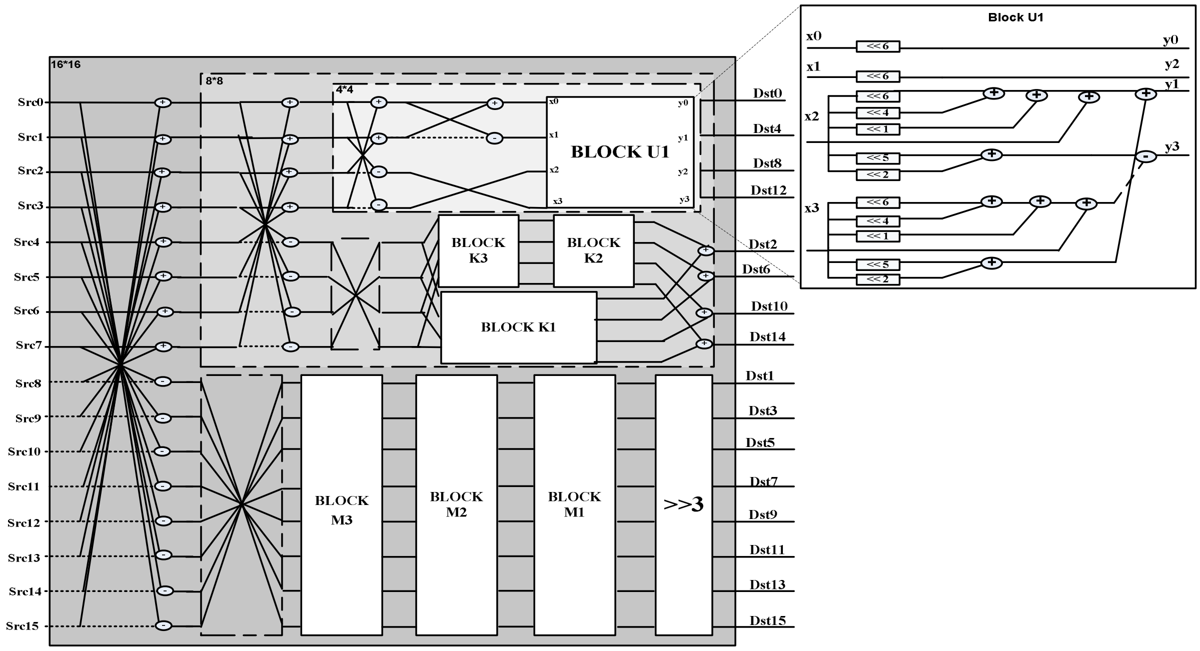

Fig. 3 Transform 1D Architecture

B. FIFOs

Our typical architecture contains 16 FIFOs that are all used when the 16-point-DCT is computed. To compute the 4-point-DCT, only 4 FIFOs are used. Concerning the 8-point-DCT, the first 8th FIFOs are used. In the proposed architecture, the new approach adopted was the use of FIFOs in order to profit from the data transposition. For example to compute the 4-point-2D-DCT as the Fig. 4 shows we proceed by the first row and each computed row is stocked in a FIFO. The process is done row by row until the end of the block. Once the step I is accomplished, we have in each FIFO a row which is transformed. The step III is responsible for the vertical transformation. These steps are fixed by the control unit as the Fig. 6 shows.

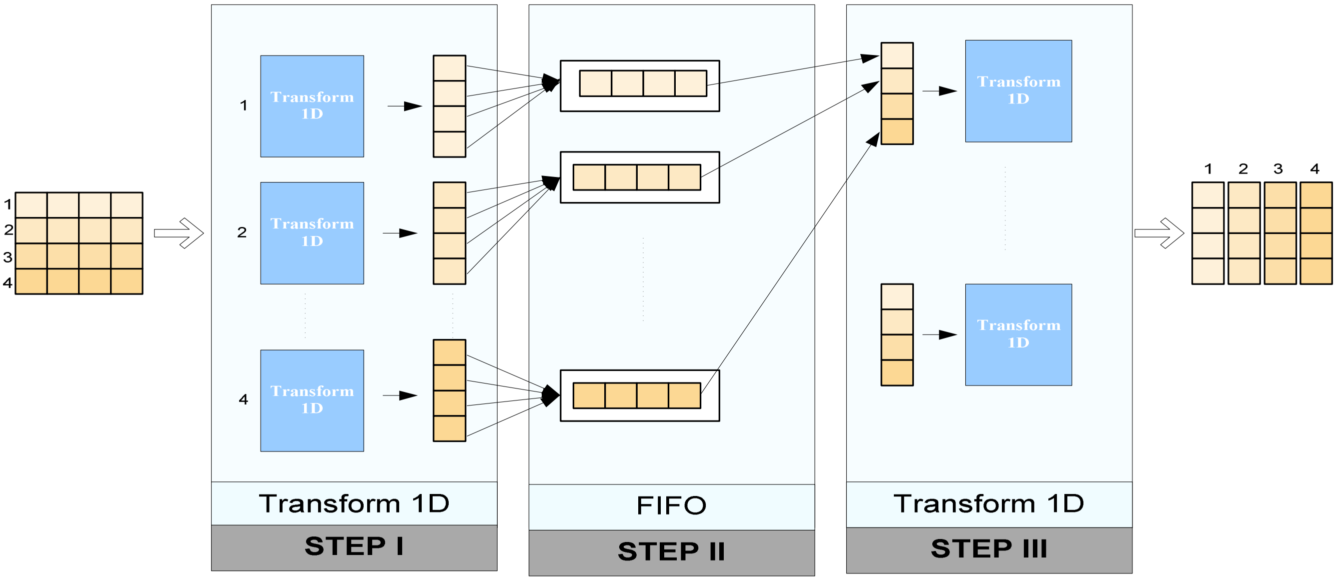

Fig. 4 4-point-2D-DCT Processing

C. Control Unit

Basically, a complete HEVC two-dimensional transform is performed by a sequential three-step approach as shown in Fig.4. The first step is the direct or horizontal transformation. The second is the results transposition using FIFOs and the last step is the vertical transformation.

The control unit permits to control the decision of the Transform Unit size as the Fig. 6 shows. When the 4-point-

2D-DCT is computed, we need one cycle for reading and one for writing the outputs.

When the transform size is eight, we have two cycles for reading and two others for writing. The 16-point-2D-DCT needs 4 cycles for reading and 4 cycles for writing. The control unit serves also to have a shared design for the three different sizes of transformation.

Merits were considerable due to the maximum reductionof the hardware cost while implementing this shared design for three different sizes. As it is proved in Fig. 5, the profit was based on reducing the number of adders from 236 for individual implementations to 160 for shared implemntation. Concerning shifts, 44% of reduction was reached when implementing the unified desing.

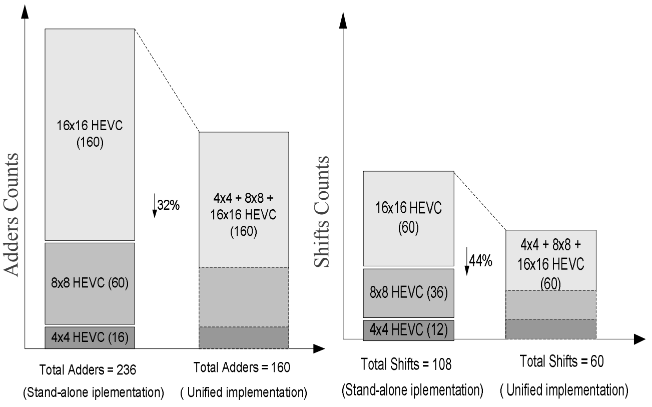

Fig. 5 Cost of the proposed scheme versus the original

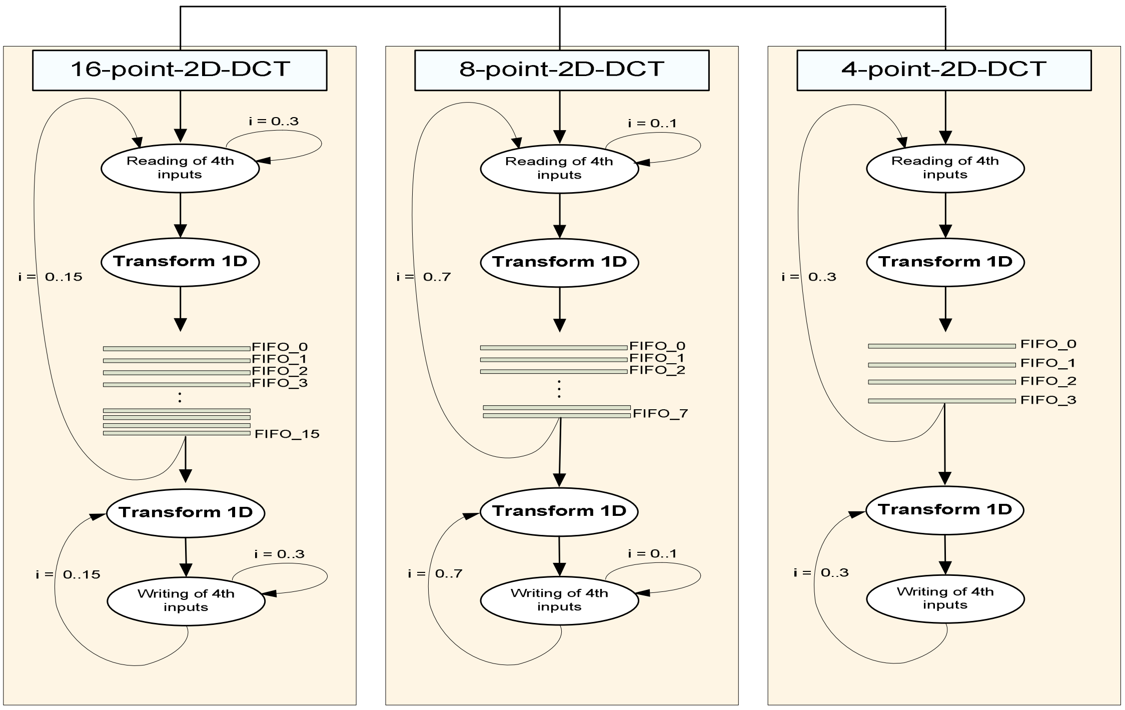

Fig. 6 Control Unit

III. Results

A. One Dimensional Transform

The proposed architecture was implemented in VHDL. The implementation was verified with RTL simulations using ModelSim ALTERA. The VHDL code was then synthesized. The FPGA family selected for the synthesis was Stratix III EP3SL70F780C2 in order to be able to compare our results to similar works that implemented the HEVC transformation for the size 16x16 discussed in the draft [8] and proposed an optimized one using shifts and adders only [5].

TABLE II
Synthesis Results for 1D Transform

Architecture	ALUTs	Freq(MHZ)	Throughput (Msamples/sec)
Original HM [5]	18,484	19.66	314.6
Optimized [5]	5,168	87.6	1401.6
Proposed	**6,091**	**251**	**4016**

As the implementation of the HEVC transforms are not very abundant, we compared our proposed architecture for the three transforms (4, 8 and 16) to another architecture implementing only the one dimensional 16x16 transform [5].

The results show an improvement concerning the hardware resources savings and the frequency. With these results, we can also evaluate the estimated throughput processed by the different architectures considering the Full HD resolution (1920x1080) because the new standard is intended especially for high resolutions [10]. Table II summarizes these results considering that the video uses 4:2:0 color sub sampling [1].

In order to process Full HD resolution at 30 frames per second, considering that it is the lowest frame rate required to establish the sensation of a continuous movement, a throughput of 93.312 million samples per second is required

[5]. As the results show in Table II, our proposed architecture shows obvious improvement comparing to the two others architectures with an estimated throughput reaching 4016 Msamples/sec using the Stratix III family. The optimizations done were crucial to allow the proposed architecture achieving the required throughput in order to process the Full HD in real time.

B. Two Dimensional Transform

As the one dimensional transform is a part from the two dimensional transform, the idea was to divide the complete 2D transform module into several small sub-modules. This architecture was designed to work efficiently in the three different sizes.

Considering all the number of cycles to read all the block and transform it horizontally then vertically, Table III resumes the total execution time for each transform. To the best of your knowledge, in the literature, this is the first hardware work implementing the 2D transform for the HEVC standard.

TABLE III
EXECUTION TIME FOR THE THREE TWO-DIMENSIONAL TRANSFORMS

Cycles	
4-point-2D-DCT	33 cycles
8-point-2D-DCT	69 cycles
16-point-2D-DCT	172 cycles

In order to validate our proposal, this architecture has been synthesized using Alteras Quartus II software adopting as target device the FPGA Stratix III. The following table presents the obtained synthesis results for the proposed architecture.

TABLE IV
SYNTHESIS RESULTS FOR 2D TRANSFORM

Architecture	ALUTs	MEMORY BITS	Freq
Proposed	**14,510**	**8,192**	**251MHZ**

IV. CONCLUSION

In this paper, we proposed a unified architecture of the two-dimensional transform using with the Modified Integer Transform. This design offers less complexity and uses less hardware resources while maintaining the same quality offered by the original algorithm for the modern video codec HEVC. This work was able to reach a considerable throughput even for full HD videos. Therefore, this architecture is capable to process at 240 MHZ in a Stratix III device. The number of cycles to compute the different sizes of the transform unit varies from 33 to 172. As future works, the hardware design for the largest dimension 32x32 is planned.

REFERENCES

[1] Gary J.Sullivan, Pankaj Topiwala, and Ajay Luthra, “The H.264/AVC Advanced Video Coding Standard:Overview and Introduction to the Fidelity Range Extensions” SPIE Conference on Applications of Digital Image Processing XXVII. Special Session on Advances in the New Emerging Standard: H.264/AVC, August, 2004.

[2] Gary J. Sullivan and Jens-Rainer Ohm “Recent developments in standardization of high efficiency video coding (HEVC)”. SPIE Conference on Applications of Digital Image Processing XXVII. Proceeding of SPIE Volume 7798, August,2010.

[3] Gary J.Sullivan, Jens-Rainer Ohm, Woo-Jin Han, “Overview of the High Efficiency Video Coding (HEVC) standard” IEE Trans. On Circuits and Systems for Video Technology, December 2012.

[4] Ashfaq Ahmed, Muhammad Awais, Martina Maurizio and Guido Masera “VLSI Implementation of 16-point DCT for H.265/HEVC using Walsh Hadamard Transform and Lifting Scheme”. IEEE 14th International Multitopic Conference. December 2011.

[5] Ricardo Jeske, José Cláudio de Souza Jr., Gustavo Wrege, Ruhan Conceição, Mateus Grellert, Júlio Mattos and Luciano Agostini “Low Cost and High Throughput Multiplierless Design of a 16 Point 1-D DCT of the New HEVC Video Coding Standard” VIII Southern Programmable Logic Conference (SPL), March 2012.

[6] Muhammad Martuza and Khan A.Wahid “Low Cost Design of a Hybrid Architecture of Integer Inverse DCT for H.264, VC-1, AVS, and HEVC” Hindawi Publishing Corporation ,VLSI Design,Volume 2012.

[7] Jong-Sik Park, Woo-Jin Nam, Seung-Mok Han, and Seongsoo Lee. “2-D Large Inverse Transform (16x16, 32x32) for HEVC (High Efficiency Video Coding)” Journal of Semiconductor Technology and Science, vol.12, June 2012.

[8] Benjamin Bross, Woo-Jin Han Jens Rainer Ohm and Gary J.Sullivan ”JCT-VC-G1103 Working Draft 5 WD 5”, Novembre 2011.

[9] Fatma Belghith, Hassen Loukil and Nouri Masmoudi “Free Multiplication integer Transformation for the HEVC Standard” The 10th International Multi-Conference on Systems, Signals and Devices (SSD) March 2013, in press .

[10] Antonio J. Díaz-Honrubia, José Luis Martínez and Pedro Cuenca, “ HEVC:A Review, Trends and Challenges” 2nd Workshop on MultimediaData Coding and Transmission, September 2011.