

# Design of an 8 Points 1-D IDCT of the Emerging HEVC Video Coding Standard

Ruhan Conceição, J. Cláudio de Souza Jr., Ricardo G. Jeske, Luciano Agostini, Júlio C. B. Mattos

*Group of Architecture and Integrated Circuits (GACI), Federal University of Pelotas (UFPEL)  
Pelotas, Brazil*

{radconceicao,jcdsouza,rgjeske,agostini,julius}@inf.ufpel.edu.br

**Abstract**— This work presents the design implementation of a 8-points 1-D IDCT used in the emerging video coding standard HEVC – High Efficiency Video Coding. The 8-points 1-D IDCT is used in the 8x8 2-D IDCT of the HEVC standard. The IDCT is performed by the video encoder and decoder as well. The hardware design implementation was done in order to enable real time video coding processing. The architecture was designed in a combinational way and multiplierless as well. Results were obtained using VHDL language and the target device was an Stratix V family FPGA. Based on the synthesis results, it was possible to achieve 960 Msamples/s and the implemented architecture – even being purely combinational – is capable to process more than 30 QFHD (3840x2160 pixels) frames per second.

**Keywords**— HEVC, IDCT, Video Coding, Hardware Implementation, Multiplierless

## I. INTRODUCTION

Nowadays, the resolution and the quality of digital videos have been improving in a fast and steady manner. Additionally, such videos are becoming supported by an increasing number of electronic devices (smartphones, set-top-box for digital television, blu-ray players, etc.). As consequence, the study and the improvement of video encoders/decoders is an extremely relevant activity in the current scenario, since the many devices that process digital videos, with their diverse features, must be capable of processing high-resolution videos in real time. For this reason, topics such as compression rate, video quality, computational complexity and energy consumption must be improved, hence they are thoroughly investigated in this area.

Video coding is imperative in applications that handle digital videos, since an uncompressed video requires a prohibitive volume of bits to be represented [1]. H.264/AVC [2] is the latest video coding standard available, presenting significant gains in compressions when compared to the MPEG-2 standard [3]. On January 2010, the JCT-VC (Joint Collaborative Team – Video Coding) was created, composed of experts from ITU-T and ISO/IEC, to start the development of a new video coding standard called HEVC – High Efficiency Video Coding [4]. The goal of the JCT-VC is to increase video compression in 50% while maintaining the same computational complexity. HEVC is on a final stage of development.

A generic video encoder can be represented as a sequence of stages, where each stage is responsible for part of the

coding process. Among the different stages, the transforms hold an important position. Normally by utilizing DCT (Discrete Cosine Transform). The purpose of the transform stage is to concentrate the energy of an image in just a few numerical coefficients. Thereby, the following stages (quantization and entropy coding) can be performed in a much more way.

In addition to these stages, prediction modules hold a prominent position at the video encoder as well. Inter prediction module searches for redundant image information among the frames which compose the video. However, since the process to encode a frame generates data loss and this frame will be used as a reference frame for other frames in the inter prediction stage; the current frame must be decoded ensuring the reference frame used by the inter prediction stage will be the same present at the video. Thus, an inverse transform process it is necessary. In this context the IDCT (Inverse Discrete Cosine Transform) is inserted, performing the inverse process performed by the DCT.

Therefore, the IDCT is performed both in the video encoder and video decoder.

This paper presents a hardware implementation of 8-points 1-D IDCT of HEVC video coding emerging standard. This work aims to process 30 QFHD – Quad Full High Definition (3840x2160 pixels) – frames per second, ensuring movement sensation to the spectator. The designed architecture does not implement multipliers, since these operators are extremely expensive in terms of area and energy consumption. Instead implementing multipliers, the architecture uses sums and shifts, due the fact that adders are not expensive as multipliers. Moreover, shifts have no cost in terms of area and energy consumption.

This paper is organized as follows: in section 2 it will be explained the process to calculate the 2-D IDCT through the 1-D IDCT and related works are also shown. After, in section 3 it will be explained how the designed architecture works and its features. Section 4 shows the results synthesis and finally, section 5 presents work conclusion and future works.

## II. INVERSE DISCRETE COSINE TRANSFORM

DCT process is common used in data compression algorithms, since this method transfers most part of information to the first matrix positions. This process does not generate data loss. Usually after performing DCT, quantization is performed by data compression algorithms. This process performs integer divisions, reducing the data in a

few numbers of coefficients. On the other hand, quantization generates data loss.

The DCT/IDCT modules are novelties of the HEVC standard and their use increases the coder efficiency and complexity. HEVC implements four sizes of IDCT (and DCT as well): 4x4, 8x8, 16x16 and 32x32. The two lower ones are also implemented by the H.264/AVC standard.

In order to calculate the 2-D IDCT, it is necessary to perform 1-D IDCT some times as it is shown in Fig. 1. Firstly, for each column of the input matrix it is performed 1-D IDCT, and the output coefficients is stored in an intermediate matrix row by row. Then, the 1-D IDCT is performed again column by column from the intermediate matrix and the results are stored column by column in the output matrix. Thus, the 2-D IDCT is performed from an input matrix to an output matrix. This process occurs likewise in the forward DCT transforms as well.

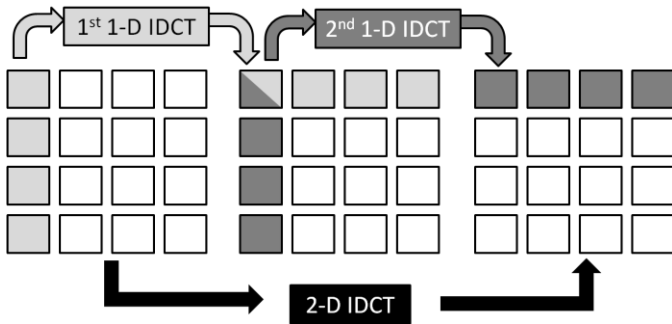


Fig. 1 Demonstration of separability property.

There are few related works about IDCTs of HEVC. The work presented in [6] shows a novel hardware-shared architecture to compute the 8x8 integer IDCTs of the HEVC and the H.264/AVC. The hardware was described in Verilog and it was synthesized in a Xilinx Vertex4 FPGA. The design was also synthesized using 0.18μm CMOS technology. The resource-shared design costs 12.3K gates and 4K standard cells with a maximum operating frequency of 211.4MHz. His architecture is capable to process one sample per clock cycle.

### III. DESIGNED ARCHITECTURE

The 8-points 1-D IDCT was designed in hardware targeting a FPGA device. For this purpose, the architecture was developed in three sequential parts. First one, from the input vector, it is performed multiplications by constants among the input coefficients. However, multipliers are very costly in terms of area and power consumption due the necessary number of logic gates to implement them [5]. Thus, these operations are done indirectly. Instead using multipliers, the implemented architecture performs sums and shifts. Therefore, hardware is saved through this technique.

Table 1 shows the constants which are used by the multiplications and their respective sums and shift operations. In order to facilitate the understanding, it is considered a variable "X" as an input coefficient which will be multiplied by the respective constant.

On a second stage, the architecture performs butterfly operations as it is done by the forward DCT. This process is represented in Fig. 2 and 3. The *Input*'s represent input coefficients already multiplied by the constant on the first stage. Likewise, *Output*'s represent output coefficients already processed by the second stage. Intermediated vector signals are named *E*, *EE*, *EO* and *O*. These labels come from the forward DCT process. As it is able to be seen, the odd positions do not appear in Fig. 2 and Fig. 3 either. It occurs due the fact that these coefficients are used to perform the *O* signals, and the process to obtain them is not the same illustrated in these figures.

TABLE I  
CONSTANT MULTIPLICATIONS AND THEIR RESPECTIVE SUMS AND SHIFT

| Constant | Sums and Shifts                   |
|----------|-----------------------------------|
| 89       | $X \ll 6 + X \ll 4 + X \ll 3 + X$ |
| 75       | $X \ll 6 + X \ll 3 + X \ll 1 + X$ |
| 50       | $X \ll 5 + X \ll 4 + X \ll 1$     |
| 18       | $X \ll 4 + X \ll 1$               |
| 83       | $X \ll 6 + X \ll 4 + X \ll 1 + X$ |
| 36       | $X \ll 5 + X \ll 2$               |
| 64       | $X \ll 6$                         |

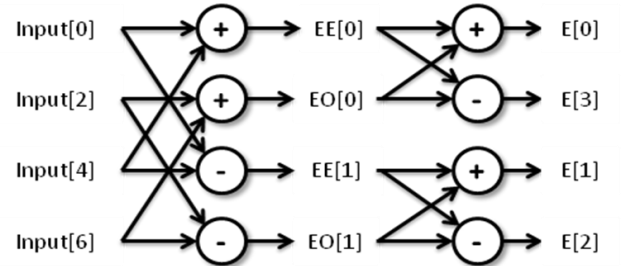


Fig. 2 First stage of Butterfly Operations

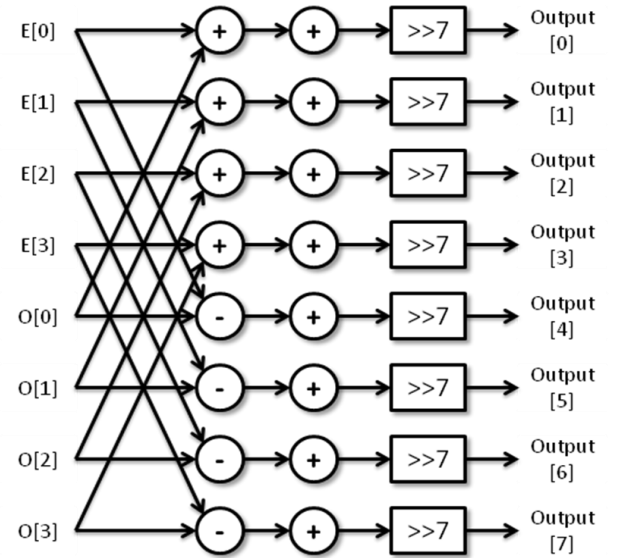


Fig. 3 Second stage of Butterfly Operations

The *O*'s signals are generated by sums done with input coefficients which are also multiplied by the constants

presented in Table 1. These sums are shown in Table 2. All input are represented by variable  $I$ .

In stage three is performed a rounding process which is illustrated in Fig. 2 as well. Firstly it is performed a sum by 64 and then the result is right-shifted by 7. Thus, the result is rounded.

TABLE II  
EQUATIONS FOR EACH O'S SIGNALS

| O's  | Equations                               |
|------|-----------------------------------------|
| O[0] | $89*I[1] + 75*I[3] + 50*I[5] + 18*I[7]$ |
| O[1] | $75*I[1] - 18*I[3] - 89*I[5] - 50*I[7]$ |
| O[2] | $50*I[1] - 89*I[3] + 18*I[5] + 75*I[7]$ |
| O[3] | $18*I[1] - 50*I[3] + 75*I[5] - 89*I[7]$ |

Designed hardware is purely combination, and it is also capable to process 8 samples per clock cycle. Thus, the architecture presents low hardware overhead.

#### IV. SYNTHESIS RESULTS

This section will present the results of the designed architecture. The 8-points IDCT architecture was synthesized targeting a high performance Stratix V Altera FPGA device using Quartus II tool.

Synthesis results are presented in Table 3. Frequency achieved number of Adaptive Logic Modules (ALMs) and register used by the architecture. Register were used only on the input and output of the designed hardware in order to obtain the correct operating frequency.

TABLE III  
SYNTHESIS RESULTS OF DESIGNED ARCHITECTURE

| Frequency (MHz) | ALMs | Registers |
|-----------------|------|-----------|
| 124.95          | 591  | 256       |

Processing 8 samples per second and achieving a operating frequency of 124.95MHz, the architecture is capable to process more than 960Msamples/second. Table 4 presents the number of frames per second that this architecture is able to process for three different video resolutions.

TABLE IV  
ACHIEVED PERFORMANCE FOR DIFFERENT VIDEO RESOLUTIONS

| Resolution | Frames per second |
|------------|-------------------|
| HD 720p    | 694               |
| HD 1080p   | 308               |
| QFHD       | 77                |

As it is able to realize though Table 4, the architecture is able to perform QFHD videos in real time.

Since Martuza [6] has implemented his architecture in two different technologies, Table 5 shows the obtained results of two architectures.

It is possible to be noticed that our architecture uses less number of logic cells than Martuza [6] FPGA design. On the other hand, it would be an unfair comparison since ALMs and LUTs are implemented differently. Even so, comparisons in

terms of performance are possible. Considering the achieved frequency and also the number of samples per second which the architectures are capable to process, it is possible to do a comparison of the number of frames per second reached. Table 6 shows this comparison.

TABLE V  
COMPARISON WITH RELATED WORK

| Features                | Developed | Martuza [6] FPGA | Martuza [6] CMOS |
|-------------------------|-----------|------------------|------------------|
| Frequency (MHz)         | 124.95    | -                | 211.4            |
| Samples per Clock Cycle | 8         | 1                | 1                |
| Gate Count              | -         | -                | 12.3K            |
| LUTs                    | -         | 706              | -                |
| ALMs                    | 591       | -                | -                |

TABLE VI  
COMPARISON AMONG ACHIEVED PERFORMANCES FOR DIFFERENT VIDEO RESOLUTIONS

| Arch        | Frames per Second (4:2:0) |          |      |
|-------------|---------------------------|----------|------|
|             | HD 720p                   | HD 1080p | QFHD |
| Proposed    | 694                       | 308      | 77   |
| Martuza [7] | 152                       | 67       | 16   |

It is possible to notice that our architecture has a high performance than Martuza [6] design, being able to perform QFHD videos in real time (30 frames per second).

#### V. CONCLUSIONS

This work presented a hardware designed of the HEVC 1-D IDCT inverse transform. A multiplierless approach was used in order to reduce hardware in terms of area and power consumption. The architecture was developed targeting a FPGA device.

Synthesis results showed that this work is capable to process more than 30 QFHD frames per second, reaching established goals.

As future works, it is intended to implement all inverse transform sizes stipulated by HEVC and also implement a multi-size architecture which is capable to process all sizes of IDCT.

#### REFERENCES

- [1] AGOSTINI, L. Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas a Compressão de Vídeo Segundo o Padrão H.264/AVC. 2007. 172f. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [2] International Telecommunication Union. "ITU-T Recommendation H.264/AVC (03/05): advanced video coding for generic audiovisual services". 2005.
- [3] International Telecommunication Union. "ITU-T Recommendation H.262 (11/94): generic coding of moving pictures and associated audio information – part 2: video". 1994.
- [4] Joint Collaborative Team on Video Coding (JCT-VC). Available at: <http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx>
- [5] L. Carro, *Projeto e Prototipação de Sistemas Digitais*, 1st ed., Ed. Universidade Federal do Rio Grande do Sul, Brazil: Porto Alegre, 2001.
- [6] M. Martuza, et al. "A cost effective implementation of 8×8 transform of HEVC from H.264/AVC" in *Electrical & Computer Engineering*

(CCECE), 2012 25<sup>th</sup> IEEE Canadian Conference on, Montreal, Quebec,  
pp 1-4.