# An OCP Implementation of the Direct and Inverse Discrete Cosine Transform for HDTV

Solon Ferreira de Lucena Júnior, José Antônio Gomes de Lima,
Leonardo Vidal Batista, Hamilton Soares da Silva and Évisson Fernandes de Lucena.

*Resumo* – **O padrão de vídeo MPEG-2 utiliza a Transformada Discreta de Co-senos (DCT) em blocos de 8x8 pixels para explorar, eficientemente, a correlação espacial entre pixels adjacentes. Este artigo propõe uma implementação em FPGA da DCT bi-dimensional (DCT-2D) dentro do Open Core Protocol (OCP). O cálculo da DCT-2D é obtido ao se aplicar a DCT-1D nas linhas de um bloco 8x8 seguido da aplicação da DCT-1D nas colunas do bloco resultante. A performance alcançada habilita o circuito a ser usado em Codecs para TV's de alta definição.**

*Palavras-Chave* – **Transformada Discreta de Co-senos (DCT), Open Core Protocol (OCP), MPEG, HDTV, VHDL, FPGA**

*Abstract* - **The MPEG-2 video standard uses Discrete Cosine Transform (DCT) in blocks of 8x8 pixels to efficiently explore the space correlation between adjacent pixels. This paper proposes a bi-dimensional DCT (DCT-2D) FPGA implementation according to the Open Core Protocol standard (OCP). The DCT-2D calculation is obtained by applying the DCT-1D to the lines of the 8x8 blocks and then applying the DCT-1D to the columns of the resultant block. Performance evaluation shows that the circuit can be used in High Definition Television codecs.**

*Keywords* - **Discrete Cosine Transform (DCT), Open Core Protocol (OCP), MPEG, HDTV, VHDL, FPGA**

## I. INTRODUCTION

Generally, video sequences contain a substantial amount of statistical redundancy between adjacent pixels and between consecutive frames. The goal of video compression is the reduction of the number of bits to be stored or transmitted [1].

The basic statistical property used in video coding is the correlation between adjacent pixels and between consecutive frames. The MPEG-2 video standard [2] uses bi-dimensional Discrete Cosine Transform (DCT-2D) [1] in blocks of 8x8 pixels to efficiently explore the space correlation between pixels in the same frame. The main objective of DCT is to concentrate the energy of the image in few important coefficients.

The Open Core Protocol standard (OCP) [3] constitutes a set of signals definitions and communication protocols that standardize the interconnection of circuit modules, reducing the project time and maintenance cost of systems on chip (SOC).

This paper presents a high performance and low cost FPGA [4], [5] implementation of the DCT-2D for HDTV [6], [7] following the OCP standard.

Section II introduces the DCT-2D and section III presents the proposed architecture; section IV describes the simulation and synthesis methodology; section V presents experimental results; finally, conclusions are presented in section VI.

Solon Ferreira de Lucena Júnior, José Antônio Gomes de Lima, Leonardo Vidal Batista, Hamilton S. da Silva and Évisson Fernandes Lucena, Departamento de Informática, Universidade Federal da Paraíba, João Pessoa, Brazil, E-mails: solon@lavid.ufpb.br, jose@di.ufpb.br, leonardo@di.ufpb.br, hamilton@di.ufpb.br, evisson@di.ufpb.br. This work was partially financed by FINEP.

## II. THE DISCRETE COSINE TRANSFORM

Normally, video pictures present high correlation between adjacent pixels. In this case, an appropriate transform can be used to concentrate the energy in few coefficients. In MPEG-2, DCT [1] is applied to the intra-frames and also to the images that represent the prediction errors between frames (MPEG) [2]. To increase the performance, the image is divided in small blocks of 8x8 pixels where DCT-2D is applied. A formula for the 8x8 DCT-2D can be written in terms of the pixels values, $f(i, j)$, and frequency domain coefficients, $F(u, v)$:

$$F(u,v) = \frac{1}{4} C(u)C(v) \sum_{i=0}^{7} \sum_{j=0}^{7} f(i,j) * \cos\left[\frac{(2i+1)up}{16}\right] \cos\left[\frac{(2j+1)vp}{16}\right], \qquad (1)$$

$$u = 0,1,...,7$$

$$v = 0,1,...,7$$

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}}, x = 0 \\ 1, x \neq 0 \end{cases}$$

DCT is a separable transform; this means that a multi-dimensional transformation can be carried through by a sequence of one-dimensional transformations. The advantage of this property is the lower complexity of DCT-1D [1].

The DCT-2D calculation can be obtained by the application of the DCT-1D to the lines of the 8x8 blocks and posterior application of the DCT-1D to the columns of the resultant block. The DCT-1D is given by:

$$F(u) = C(u) \sum_{i=0}^{7} f(i) \cos\left[\frac{(2i+1)up}{16}\right], \qquad (2)$$

$$u = 0,1,...,7$$

$$C(x) = \begin{cases} \frac{1}{2\sqrt{2}}, x = 0 \\ \frac{1}{2}, x \neq 0 \end{cases}$$

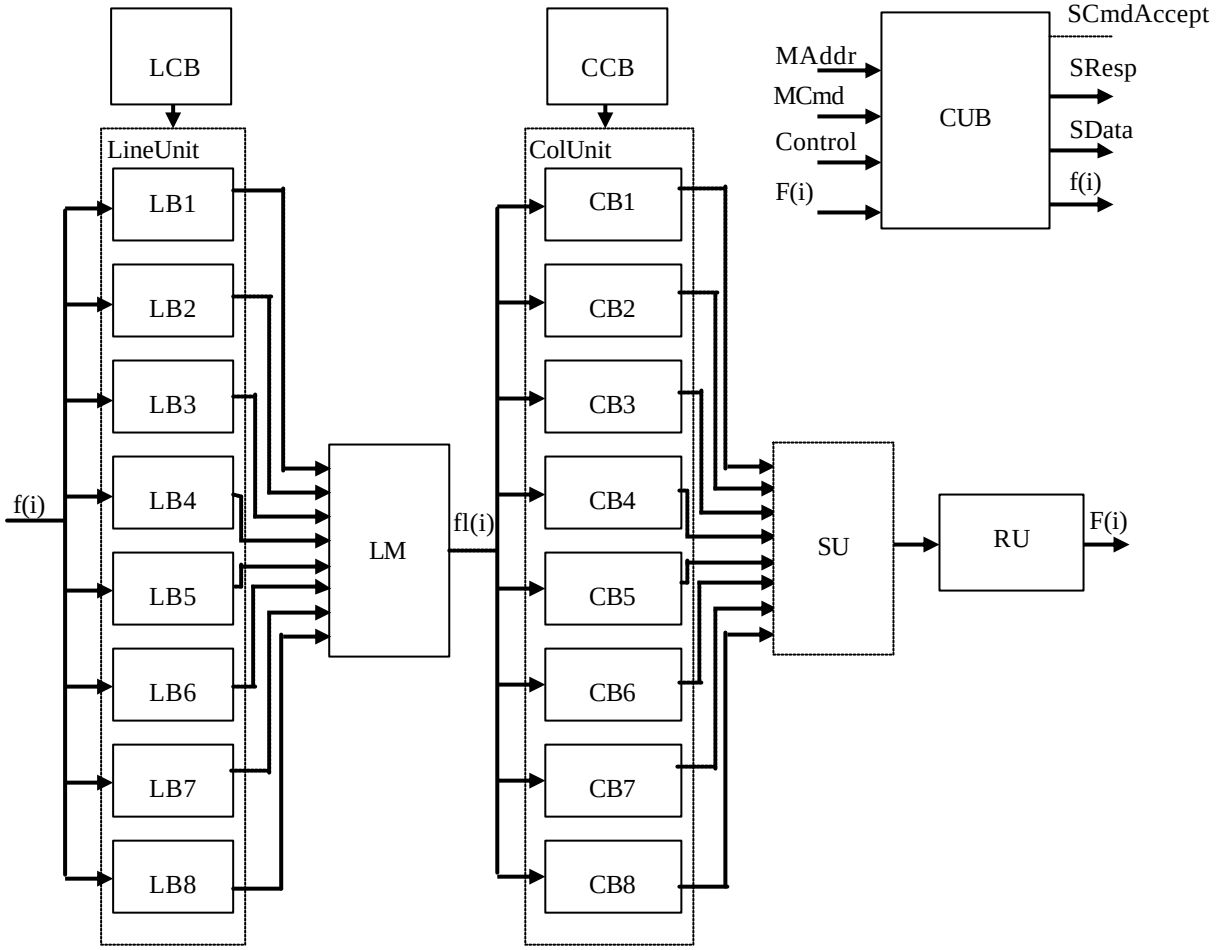


Figure 1 – Architecture of the DCT

## III. DCT/IDCT ARCHITECTURE IMPLEMENTED IN FPGA

Figure 1 shows the DCT/IDCT architecture implemented in FPGA. It is composed of the following main blocks: Line Unit (LineUnit); Line Block (LB); Column Unit (ColUnit); Column Block (CB); Serialization Unit (SU); Line Coefficient Block (LCB); Column Coefficient Block (CCB); Line Multiplexer (LM); Round Unit (RU); and the Control Unit Block (CUB).

The major diference between the DCT and the inverse DCT (IDCT) architecture is the LCB, CCB coefficient values. They are specific for each implementation.

Equation (2) when applied at a line of eight values generates the following relations, where each $F(u)$ coefficient represent a position in resultant line:

$F(0) = 0.354\ [\ f(0) + f(1) + f(2) + f(3) + f(4) + f(5) + f(6) + f(7)\ ]$

$F(1) = 0.490f(0) + 0.416f(1) + 0.278f(2) + 0.098f(3) - 0.98f(4) - 0.278f(5) - 0.416f(6) - 0.490f(7)$

$F(2) = 0.462f(0) + 0.191f(1) - 0.191f(2) - 0.462f(3) - 0.462f(4) - 0.191f(5) + 0.191f(6) + 0.462f(7)$

$F(3) = 0.416f(0) - 0.98f(1) - 0.490f(2) - 0.278f(3) + 0.278f(4) + 0.490f(5) + 0.98f(6) - 0.416f(7)$

$F(4) = 0.354f(0) - 0.354f(1) - 0.354f(2) + 0.354f(3) + 0.354f(4) - 0.354f(5) - 0.354f(6) + 0.354f(7)$

$F(5) = 0.278f(0) - 0.490f(1) + 0.98f(2) + 0.416f(3) - 0.416f(4) - 0.98f(5) + 0.490f(6) - 0.278f(7)$

$F(6) = 0.191f(0) - 0.462f(1) + 0.462f(2) - 0.191f(3) - 0.191f(4) + 0.462f(5) - 0.462f(6) + 0.191f(7)$

$F(7) = 0.98f(0) - 0.278f(1) + 0.416f(2) - 0.490f(3) + 0.490f(4) - 0.416f(5) + 0.278f(6) - 0.98f(7)$.

The $F(u)$ coefficient is obtained from the sum of the products of each incoming f(i) line pixel of the 8x8 matrix pixel and the specific cosine value as showed in equation (2). To obtain the DCT-2D coefficients another operation must be made using the results obtained in the previous operation and cosine values (specific cosine values from CCB).

To simplify and to increase the performance of the circuit two units, Line Unit (LineUnit) and Column Unit (ColUnit) were designed to perform operations in the lines and columns of the blocks, respectively. The LineUnit is composed of a series of 8 independent circuits called Line Block (LB) that perform operations in the block lines. The Column Block (ColUnit) performs operations in the block columns as showed in Figure 1.

*A. Line Unit (LineUnit)*

The Line Unit is composed of eight Line Block (LB). The Line Block is composed by one Multiplier ACCumulator (MACC) [8] and one internal memory (MEM INT) that stores the operation result as showed in Figure 2.

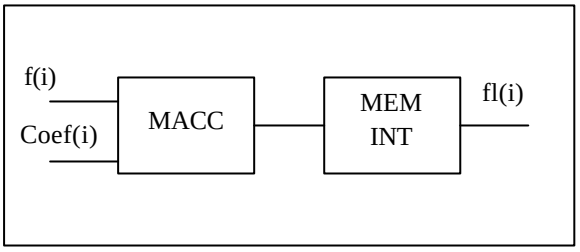


Figure 2 – Architecture of LB

The MACC receives the *f(i)* pixel to be processed and the specific coefficient value Coef(i) from the Line Coefficient Block (LCB) and performs the operation *f*(i) x Coef(i). After eight interactions the final sum of products is stored in MEM INT and the *fl(i)* output is used as input to perform the DCT-2D calculation by the Column Unit (ColUnit).

*B. Column Unit (ColUnit)*

The Column Unit is composed of eight Column Block (CB). The Column Block is composed of one MACC as showed in Figure 3.

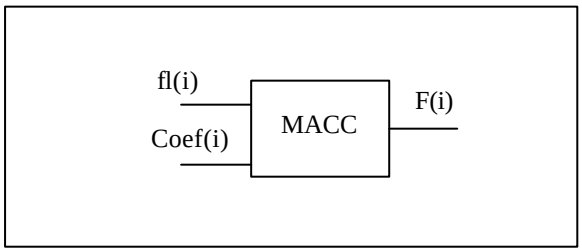


Figure 3 – Architecture of CB

The MACC receives the *fl(i)* output from the Line Unit (LU) to be processed and the specific coefficient value Coef(*i*) from the Column Coefficient Block (CCB) performing the operation fl(i) x Coef(i). Each CB calculates the specific DCT-2D coefficient

After eight interactions the final sum of product is completed and the DCT-2D coefficient output used as input to perform the serialization by the Serialization Unit.

*C. Control Unit Block (CUB)*

The CUB controls the data path and generates all signals to control the other blocks of the proposed DCT-2D circuit.

The CUB generates too the signals that define a point to point interface between two communicating entities such as IP Cores and bus interface modules (OCP).

This protocol delivers the only non-proprietary, open, licensed, core centric protocol that describes the system level integration requirements of intellectual property (IP) cores.

In the proposed architecture, DCT-2D acts as a slave of the OCP instance using the basic signals: Clock, MAddr, MCmd, SCmdAccept, SData and SResp, and the Control signal of the control and status signals group as showed in Figure 4.

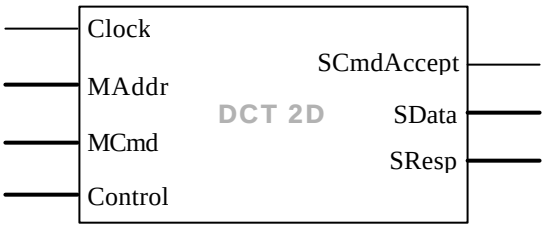


Figure 4 – The DCT-2D as an OCP IP slave

Table I lists the width, the type (data or control) and the function of each OCP signals used.

TABLE I – I/O Signals of the Chip

| **Signal** | **Width** | **I/O** | **D/C** | **Function** |
|---|---|---|---|---|
| Clock | 1 bit | I | C | System Clock |
| MAddr | 14 bits | I | D/C | Transfer data. The last six bits of the MAddr represents the chip address and the first eight bits the incoming pixel. |
| MCmd | 3 bits | I | C | Transfer command (001 – Write ; 010 – Read; 000 – Idle) |
| Control | 6 bits | I | C | DCT-2D address chip - set by user. |
| Sdata | 16 bits | O | D | Output data. The first 12 bits represent the output DCT-2D coefficient, the 13$^{th}$ bit (=1) indicates the first coefficient value and the last three bits are "don't care" bits. |
| SCmdAccept | 1 bit | O | C | Indicates acceptance of write data (SCmAccept = 1). |
| Sresp | 2 bits | O | C | Response field to a transfer request; (00 – No response; 01 – Data valid ). |

*D. Serialization Unit (SU)*

The Serialization Unit (SU) performs a serialization step in DCT-2D coefficient from Column Unit generating one output pixel by clock cycle.

*E. Round Unit (RU)*

The Round Unit (RU) converts the integer value that represents each DCT-2D coefficient in 12 bits value.

## IV. SYNTHESIS AND SIMULATION METHODOLOGY

The DCT-2D was described in VHDL [8]. The general architecture was divided in blocks as showed in Figure 1. The blocks were divided in sub-blocks. At sub-blocks level, behavioral descriptions were made and at block level structural descriptions connecting the sub-blocks were constructed and validated using the Max_Plus [9],[10] and Quartus II Web Edition software from Altera [11].

The DCT-2D acts as a slave of the OCP instance using the basic signals: Clock, MAddr, MCmd, SCmdAccept, SData and SResp, and the "Control" signal of the control and status signals group.

The simulation process was divided in two steps: The first step: "initialization" is necessary to reset the circuit and to synchronize the first input pixel in the next step. The MAddr word format in Figure 5 shows the "Address" field that must contain the DCT-2D address set in the Control input.

As showed in Figure 7 the control word (MCmd = 001) indicates that the last six bits of the MAddr (000001) represents the chip address and it will be accepted at the rising edge of the clock signal (point A of the Figure 7). In this point the DCT output SCmdAccept = 1, indicates this acceptance. This corresponds to the end of the "Initialization" step.

In the second step, the incoming pixel is sent to circuit through the MAddr bus in the first eight bits. The MAddr is read at the rising edge of the clock signal (point B of Figure 7).

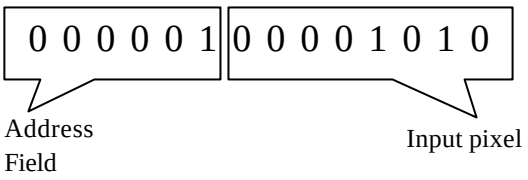


Figure 5 - MAddr word format

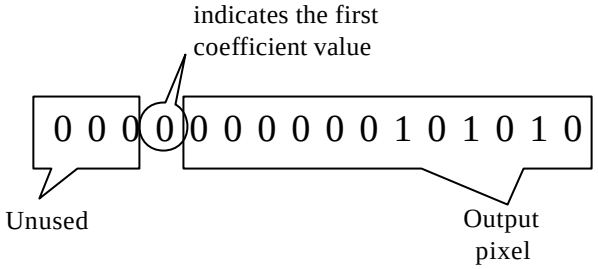


Figure 6 – Sdata word format

After 64 incoming pixel (8x8 matrix pixel) the DCT-2D coefficients are send to the output data Sdata (16 bit bus) in the rising edge of the clock signal, the Sresp output (01) indicates valid output in Sdata bus, as showed in the point B of Figure 8. The first 12 bits of the Sdata word is the output DCT-2D coefficient, the 13$^{th}$ bit (=1) indicates the first coefficient value and the last three bits are "don't care" bits as showed in Figure 6.

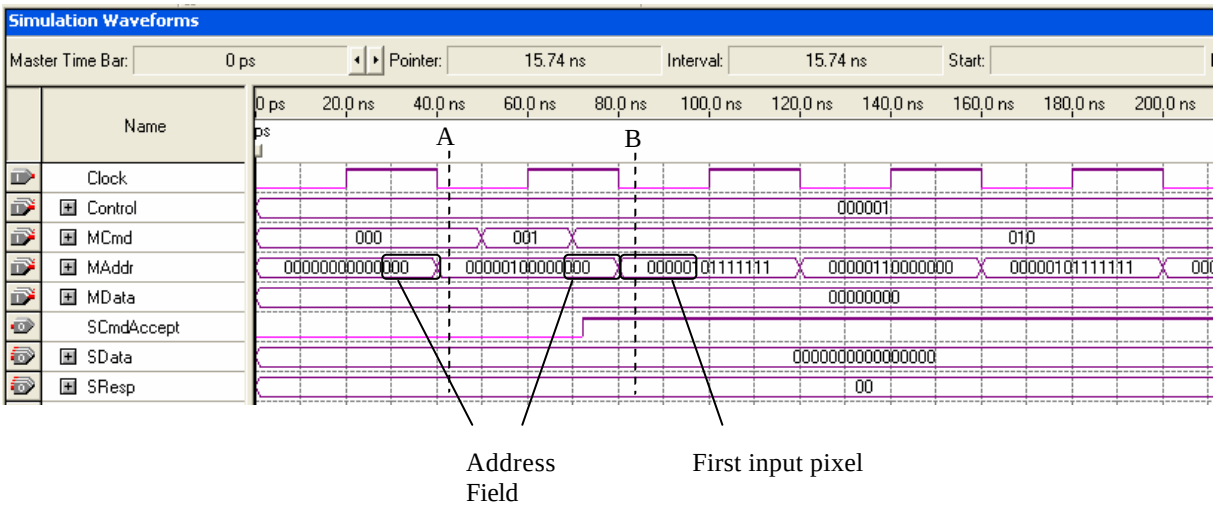


Figure 7 – Initialization step

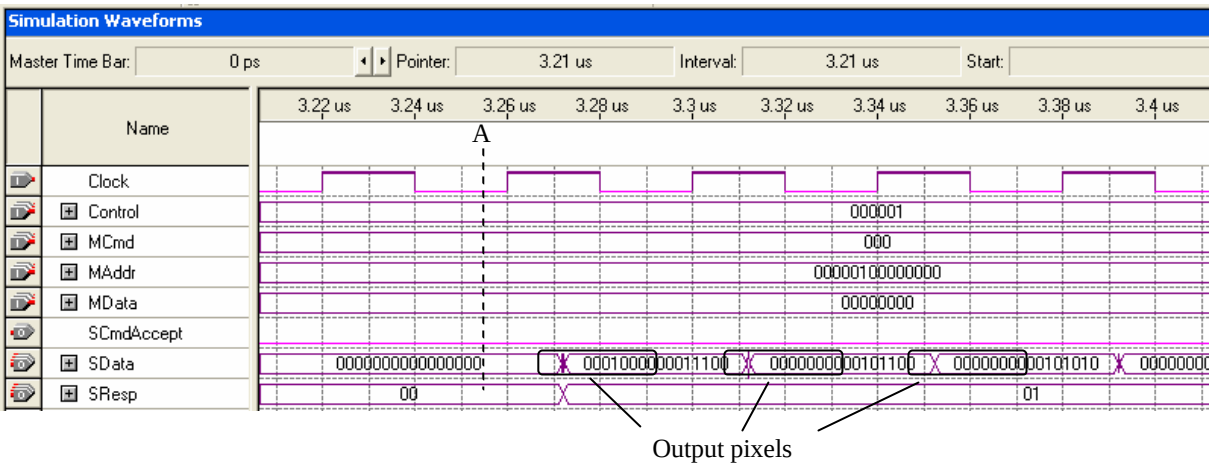


Figure 8 – Read step

## V. EXPERIMENTAL RESULTS

The DCT-2D was synthesized using Altera tools [9], [10], [11]. Results in terms of number of logic cells, memory utilized, frequency of operation and device used are showed in Table II. The maximum frequency of 75.53 MHz for the Stratix Family [11] (EP1S10F484C5) device show that each 8x8 input block is processed in 0.84 µs, allowing a maximum throughput of 75 Mpixel/s, compatible with the HDTV specifications [6], [7]. The synthesis results from the EP1S80B956C6 and EP1C6T144C6 devices show that each 8x8 input block is processed in 0,96 µs and 0,97 µs respectively, with a maximum throughput also compatible with the HDTV specifications.

Figure 9 shows the original and the reconstructed image used to validate the DCT-2D implementation. There are no visual differences between the images.

Figure 10 shows numeric results in one specific block. The application of the DCT in original 8x8 block (a), generates a corresponding DCT block (b). The application of the IDCT in (b) generates a reconstructed block as showed in (c). Figure 10(d) shows the obtained errors between original and reconstructed pixel values.

TABLE II – Synthesis results of the DCT – 2D

| Device | Logic Elements | I/O Pins | Memory Bits | DSP Blocks | Freq. MHz |
|---|---|---|---|---|---|
| EP1S10F484C5 (Stratix Family) | 4.065 (38 %) | 51 | 2.656 (< 1%) | 48 (100 %) | 75.53 |
| EP1C6T144C6 (Ciclone Family) | 4.536 (75 %) | 51 | 4.184 (4 %) | - | 66.20 |
| EP1S80B956C6 (Stratix Family) | 2.209 (2 %) | 51 | 2.624 (< 1 %) | 80 (45 %) | 67.26 |

(a) Original Image (b) Restored Image

Figure 9 – DCT/IDCT result application

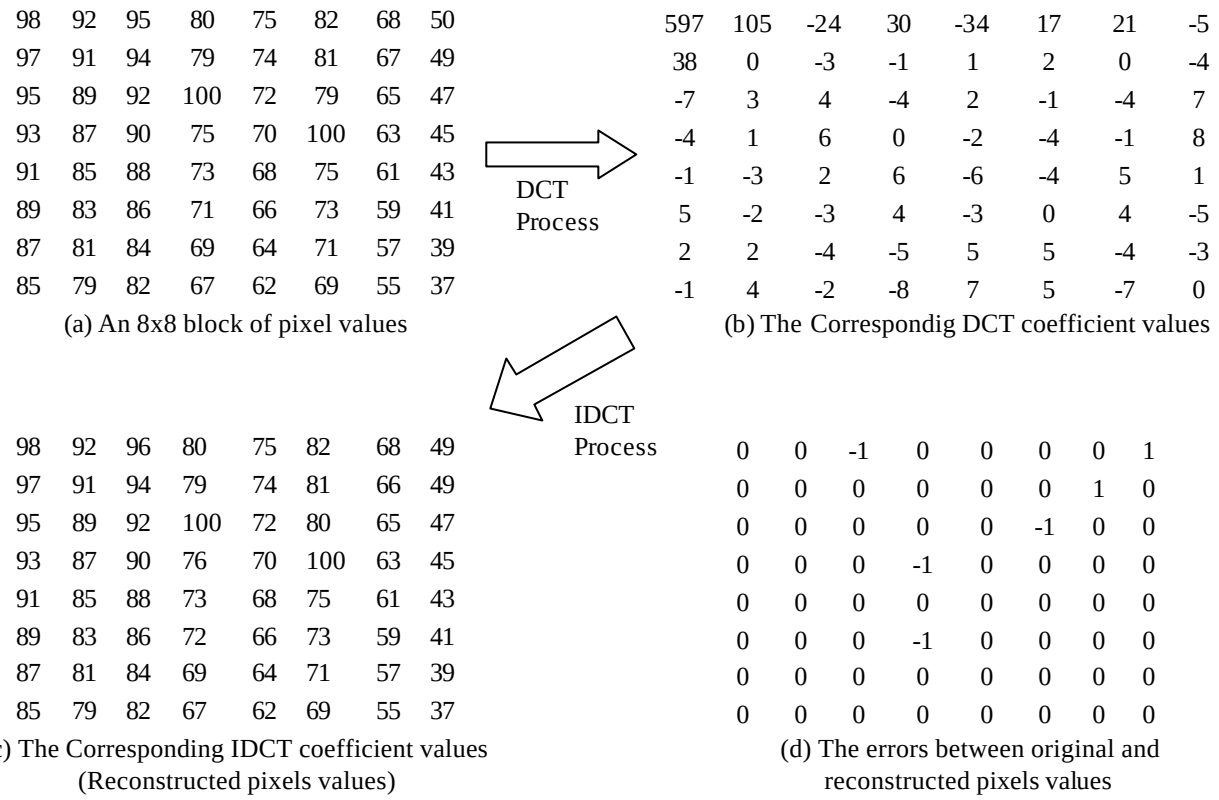


Figure 10 – Results at pixel block level

## VI. CONCLUSIONS AND FUTURE WORKS

The design of an FPGA implementation of bi-dimensional DCT (DCT-2D) according to the OCP standard has been described. The results show that it is possible to implement all the DCT-2D functions in a single FPGA chip compatible with HDTV specifications. The use of FPGA devices opens new possibilities through reconfiguration, which is important to assure a rapid development and prototyping of DCT-2D chips.

New FPGA families appearing in the market, allowing the implementation of more complex cores with significantly improved performance, encourage further researches to optimize DCT-2D project design adding new functionalities for HDTV applications in the same chip.

## REFERENCES

[1] K.R. Rao; and P. Yip: *Discrete Cosine Transform – Algorithms, Advantages, Applications.* Academic Press, 1990.

[2] ISO/IEC 113818-2 (MPEG-2).

[3] *Open Core Protocol Specification 2.0*, Document Revision 1.1, 2003.

[4] M.-B., Uwe: Digital Signal Processing with Filed Programmable Gate Arrays. Springer, 2001.

[5] B. Zeidman: Designing with FPGAs & CPLDs, CMP Books, 2002.

[6] M. Robin and M. Polin: Digital Television Fundamentals: Design and installation of video and audio systems. McGraw-Hill, second edition, 2000.

[7] M. Ercegovac: Introdução aos Sistemas /Digitais, T. Lang and J. H. Moreno: Trad. J. C. B. Santos, Bookman, 2000.

[8] R. K. Dueck Digital Design with CPLD applications and VHDL, Delmar Thomson Learning, 2000.

[9] Altera Corporation. Altera Data Book ,1995.

[10] Altera Corporation. Data Book and Max + Plus II Getting Started ,1997.

[11] Altera Corporation. Using Quartus II Verilog HDL & VHDL Integrated Synthesis, 2002.

[12] M. Orzessek: ATM & MPEG-2: Integrating Digital Video *Into Broadband Networks.* Pretience Hall, 1998.