FAST-DCT MODIFIED ALGORITHMS IMPLEMENTED IN FPGA CHIPS FOR REAL-TIME IMAGE COMPRESSION

Agnieszka Dąbrowska, Kazimierz Wiatr

University of Science and Technology, Institute of Electronics
Mickiewicza 30, 30-059 Cracow, Poland

Abstract: The Discrete Cosine Transform (DCT) is one of the basic varieties of transform coding algorithms. Moreover DCT is used in standard algorithms of compression of still image (JPEG) and video compression algorithms (MPEG, H.26x). In case of compression's images algorithms there are used the blocs: 8×8 pixels. Paper presents problems connected with implementation of DCT algorithms in reconfigurable FPGA structures. Authors implemented DCT transform used Chen's algorithm, Lee's algorithm and Loeffler's algorithm, because these algorithms are most effective for FPGA implementation. Paper presents implementations results in Xilinx chips XCV200BG352. *Copyright 2004 IFAC.*

Keywords: Algorithms, Discrete Cosine Transform, Implementation, Image compression, Real-time.

1. INTRODUCTION

The two-dimensional discrete cosine transform can be divide for two one-dimensional discrete cosine transforms (Bukhari, *et al*). One of them is realized towards rows and the second towards columns. If x_n denotes vector of input values then the vector of transformed values y_n can be marked on the basis of the dependence (1).

$$y_k = \alpha_k \sum_{n=0}^{N-1} x_n \cos\left[\frac{2\pi}{4N}(2n+1)k\right] \qquad (1)$$

where : k=(0, 1, ..., N-1) $\alpha_0 = 1/\sqrt{N}$ for k=0 or $\alpha_k = \sqrt{2/N}$ in remaining cases.

The discrete cosine transform can be represented in a matrix figure. For N = 8 a matrix-vector form is following (2), where the coefficients c_k are equal c_k=cos($k\pi$/16).

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \alpha_k \begin{bmatrix} c_0 & c_0 & c_0 & c_0 & c_0 & c_0 & c_0 & c_0 \\ c_1 & c_3 & c_5 & c_7 & -c_7 & -c_5 & -c_3 & -c_1 \\ c_2 & c_6 & -c_6 & -c_2 & -c_2 & -c_6 & c_6 & c_2 \\ c_3 & -c_7 & -c_1 & -c_5 & c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 & c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 & -c_3 & -c_7 & c_1 & -c_5 \\ c_6 & -c_2 & c_2 & -c_6 & -c_6 & c_2 & -c_2 & c_6 \\ c_7 & -c_5 & c_3 & -c_1 & c_1 & -c_3 & c_5 & -c_7 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \qquad (2)$$

What is visible from the dependence (2) to calculate a single vector of transformed values should be executed 64 multiplications and 56 additions. In case of 8 vectors of input values the number of operations necessary to enumerating transformed values, characteristic for them, makes 1024 multiplications and 896 additions for the two-dimensional discrete cosine transform. The number of executed multiplications may be reduced from 64 to 32 thanks to the symmetry of coefficients' matrix c_k but the number of additions and subtractions will grow up simultaneously. In this case a matrix-vector form is following:

$$\begin{bmatrix} y_0 \\ y_2 \\ y_4 \\ y_6 \\ y_1 \\ y_3 \\ y_5 \\ y_7 \end{bmatrix} = \alpha_k \left[\begin{array}{cccc:cccc} c_0 & c_0 & c_0 & c_0 & 0 & 0 & 0 & 0 \\ c_2 & c_6 & -c_6 & -c_2 & 0 & 0 & 0 & 0 \\ c_4 & -c_4 & -c_4 & c_4 & 0 & 0 & 0 & 0 \\ c_6 & -c_2 & c_2 & -c_6 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & c_1 & c_3 & c_5 & c_7 \\ 0 & 0 & 0 & 0 & c_3 & -c_7 & -c_1 & -c_5 \\ 0 & 0 & 0 & 0 & c_5 & -c_1 & c_7 & c_3 \\ 0 & 0 & 0 & 0 & c_7 & -c_5 & c_3 & -c_1 \end{array}\right] \cdot \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \\ x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \tag{3}$$

Proceeding in the same way it is possible to reduce the number of multiplications to 22. In this case the matrix-vector form is following (4).

$$\begin{bmatrix} y_0 \\ y_2 \\ y_4 \\ y_6 \\ y_1 \\ y_3 \\ y_5 \\ y_7 \end{bmatrix} = \alpha_k \left[\begin{array}{cccc:cccc} c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_2 & c_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_6 & -c_2 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & c_1 & c_3 & c_5 & c_7 \\ 0 & 0 & 0 & 0 & c_3 & -c_7 & -c_1 & -c_5 \\ 0 & 0 & 0 & 0 & c_5 & -c_1 & c_7 & c_3 \\ 0 & 0 & 0 & 0 & c_7 & -c_5 & c_3 & -c_1 \end{array}\right] \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} \tag{4}$$

where vector w_n is (5):

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} = \begin{bmatrix} x_0 + x_7 + x_3 + x_4 + x_1 + x_6 + x_2 + x_5 \\ x_0 + x_7 + x_3 + x_4 - x_1 - x_6 - x_2 - x_5 \\ x_0 + x_7 - x_3 - x_4 \\ x_1 + x_6 - x_2 - x_5 \\ x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \tag{5}$$

2. THE FAST DCT ALGORITHMS

The DCT algorithms in which the number of required operations of multiplications was reduced they are called the fast DCT algorithms.
One of such algorithms is the Chen's algorithm. The number of multiplications was reduced to 16 and addition/subtraction was reduced to 26 operations.

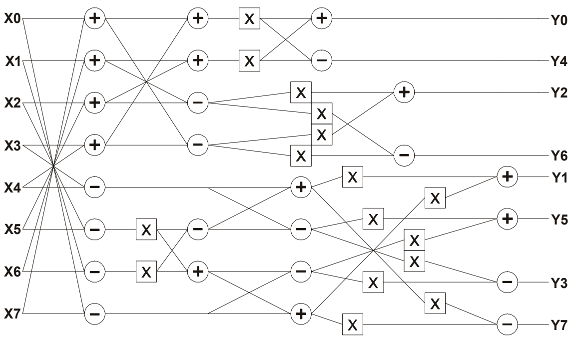

Fig. 1. The graphic scheme of flowing for the Chen's algorithm.

The greater reductions of number of necessary arithmetic operations was received in the Lee's algorithm (Chotin R., *et al.*) in which the number of multiplications is equal 13 and the number of additions and subtractions is equal 29.

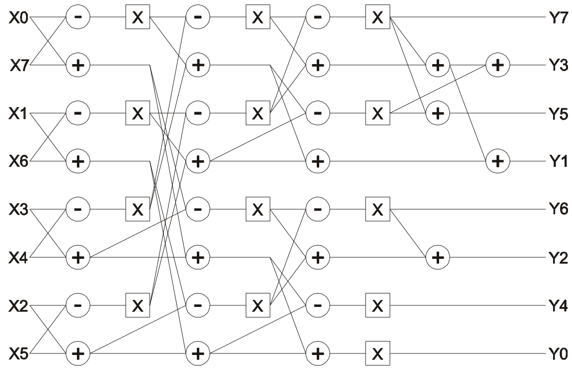

Fig. 2. The graphic scheme of flowing for the Lee's algorithm.

In the Loeffler's algorithm (Fig. 3) the number of multiplications was reduced to 11 and addition/subtraction was reduced to 29 operations.

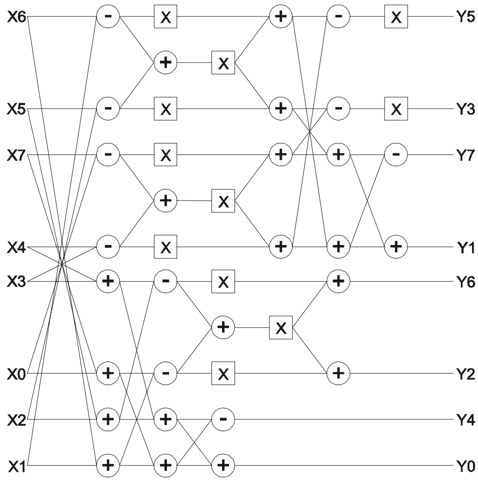

Fig. 3. The graphic scheme of flowing for the Loeffler's algorithm.

3. DCT ALGORITHMS IMPLEMENTATION IN THE FPGA

During an implementation of 1D-DCT algorithms there appears a question: "What accuracy should be accepted for the coefficients of multiplication". We should receive the accuracy in the order of 10^{-2} or maybe there is a necessary to receiving multiplication's coefficients with an accuracy of higher order. The really important question is also: "How the received accuracy does move on values of transformed values?".
For representation's accuracy of multiplication's coefficients to the second place after comma the received maximum difference between the calculated analytically transformed values and the values received from the implemented Chen's algorithm is equal 12,75. The maximum difference appears at maximum values of input pixels (x_0 to x_7 are equal 255). In this case the value of transformation's coefficient y_0 is equal 734 where the value of this coefficient calculated analytically is equal 721,249. Theoretically the maximum difference is equal 20,4

so is larger than value received from the implemented algorithm.

If the multiplication's coefficients were realized with an accuracy to the third place after comma then the maximum difference between the calculated analytically transformed values and received from the implemented algorithm is equal 1,182. The maximum appointed theoretically difference is equal 2.04, thereat accuracy.

There exists the question about sufficient accuracy of representation of multiplication's coefficients. We should answer also for the question: "How does the received differences between y_n received from implemented algorithm and y_n calculated analytically move on pixels' values received thanks to inverse discrete cosine transform (IDCT)?".

Table 1 The consideration for a hypothetical input vector with the accuracy in the order of 10^{-2}.

x_n	y_n calculated analytically	y_n received from the implemented Chen's algorithm	The difference between y_n received and y_n calculated	x_n received from the implemented Chen's algorithm
x_0 =252	y_0 = 645,5885	y_0 = 657	-11,4115	255
x_1 =250	y_1 = 55,33119	y_1 = 55	0,33119	255
x_2 =249	y_2 = 5,037243	y_2 = 5	0,037243	253
x_3 =247	y_3 = -27,3337	y_3 = -28	0,66633	251
x_4 =207	y_4 = -22,6274	y_4 = -23	0,372583	211
x_5 =223	y_5 = -21,849	y_5 = -23	1,151043	226
x_6 =159	y_6 = 36,18185	y_6 = 37	-0,81815	162
x_7 =239	y_7 = -32,817	y_7 = -33	0,183041	244

Table 2 The consideration for a hypothetical input vector with the accuracy in the order of 10^{-3}.

x_n	y_n calculated analytically	y_n received from the implemented Chen's algorithm	The difference between y_n received and y_n calculated	x_n received from the implemented Chen's algorithm
x_0 =252	y_0 = 645,5885	y_0 = 645	0,58849	252
x_1 =250	y_1 = 55,33119	y_1 = 55	0,33119	250
x_2 =249	y_2 = 5,037243	y_2 = 5	0,03724	248
x_3 =247	y_3 = -27,3337	y_3 = -27	-0,33367	247
x_4 =207	y_4 = -22,6274	y_4 = -23	0,37258	206
x_5 =223	y_5 = -21,849	y_5 = -21	-0,848958	223
x_6 =159	y_6 = 36,18185	y_6 = 36	0,181849	159
x_7 =239	y_7 = -32,817	y_7 = -33	0,183041	238

What is visible from the Tab. 1, 2, 3, 4, 5 at the accuracy in the order of 10^{-2} the maximum difference *MD* between the value of the input pixel and the value of the reconstructed pixel by the implemented Chen's algorithm is equal 5. If the representation's accuracy

Table 3 The example for small differences between neighbouring pixels for Chen's algorithm with the accuracy in the order of 10^{-2}.

x_n	y_n calculated analytically	y_n received from the implemented Chen's algorithm	The difference between y_n received and y_n calculated	x_n received from the implemented algorithm
x_0 =191	y_0 = 530,3301	y_0 = 540	-9,67	194
x_1 =190	y_1 = 6,442323	y_1 = 6	0,4423	194
x_2 =189	y_2 = 0	y_2 = 0	0	193
x_3 =188	y_3 = 0,673455	y_3 = 0	0,67345	191
x_4 =187	y_4 = 0	y_4 = 0	0	190
x_5 =186	y_5 = 0,200903	y_5 = 0	0,2009	189
x_6 =185	y_6 = 0	y_6 = 0	0	188
x_7 =184	y_7 = 0,050702	y_7 = 0	0,0507	188

Table 4 The example for small differences between neighbouring pixels for Chen's algorithm with the accuracy in the order of 10^{-3}.

x_n	y_n calculated analytically	y_n received from the implemented Chen's algorithm	The difference between y_n received and y_n calculated	x_n received from the implemented algorithm
x_0 =191	y_0 = 530,3301	y_0 = 530	0,3301	190
x_1 =190	y_1 = 6,442323	y_1 = 6	0,4423	190
x_2 =189	y_2 = 0	y_2 = 0	0	189
x_3 =188	y_3 = 0,673455	y_3 = 0	0,67345	187
x_4 =187	y_4 = 0	y_4 = 0	0	186
x_5 =186	y_5 = 0,200903	y_5 = 0	0,2009	185
x_6 =185	y_6 = 0	y_6 = 0	0	184
x_7 =184	y_7 = 0,050702	y_7 = 0	0,0507	184

Table 5 A case for the maximum difference between y_n received and y_n calculated for Chen's algorithm with the accuracy in the order of 10^{-3}

x_n	y_n calculated analytically	y_n received from the implemented Chen's algorithm	The difference between y_n received and y_n calculated	x_n received from the implemented algorithm
x_0 =253	y_0=393,15137	y_0 = 393	0,15137	253
x_1 =33	y_1=83,52634	y_1 = 83	0,526338	34
x_2 =247	y_2=-25,13239	y_2 = -25	-0,132389	247
x_3 =121	y_3 = 78,72797	y_3 = 78	0,72797	120
x_4 =87	y_4=46,669048	y_4 = 47	-0,330952	87
x_5 =219	y_5=123,22545	y_5 = 123	0,2254516	217
x_6 =123	y_6=157,36061	y_6 = 157	0,3606145	123
x_7 = 29	y_7=41,818001	y_7 = 43	-1,181999	29

of multiplication's coefficients is in the order of 10^{-3} then the maximum difference *MD* between pixels is equal 2.

The implementation of the Chen's DCT with the accuracy in the order of 10^{-3} in the structure XCV200BG352 of Xilinx occupies 574 from 2352 of SLICE blocks (what states 24% of the structure's supplies), 20% LUT blocks (958 from 4704). The maximum attainable frequency for this implementation is equal 76,115MHz. For the implementation of the Chen's DCT with an accuracy of representation of multiplication's coefficients in the order of 10^{-2} this frequency reaches 105,831MHz for the structure XCV200BG352. In this case the SLICE blocks are occupied in 16% (382 from 2352) and LUT blocks in 13% (614 from 4704). From the viewpoint of working speed better is the implementation of algorithm with the accuracy in the order of 10^{-2}. However from the viewpoint of pixels' reconstruction's accuracy better is the implementation of algorithm with the accuracy in the order of 10^{-3}.

There exists some possibility which will be a compromise between speed's working and an accuracy of pixels' reconstruction. It is enough to implement the Chen's DCT algorithm with an accuracy of multiplication's coefficients in the order of 10^{-2} except of multiplication's coefficient occurrents at marking transformed value y_0.

Table 6 The consideration for a hypothetical input vector with the accuracy in the order of 10^{-2} except of coefficients occurrents at marking y_0..

x_n	y_n calculated analytically	y_n received from the implemented Chen's algorithm	The difference between y_n received and y_n calculated	x_n received from the implemented Chen's algorithm
x_0 =253	y_0 = 393,15137	y_0 = 393	0,15137	255
x_1 =33	y_1 = 83,52634	y_1 = 82	1,526338	31
x_2 =247	y_2 = -25,13239	y_2 = -23	-2,13239	247
x_3 =121	y_3 = 78,72797	y_3 = 80	-1,27203	119
x_4 =87	y_4 = 46,669048	y_4 = 47	-0,33095	84
x_5 =219	y_5 = 123,22545	y_5 = 126	-2,77455	220
x_6 =123	y_6 = 157,36061	y_6 = 159	-1,63939	124
x_7 =29	y_7 = 41,818001	y_7 = 42	-0,182	30

What is visible from the Tab. 6 the maximum difference *MD* between the value of the input pixel and the value of the reconstructed pixel by the implemented Chen's algorithm with an accuracy of multiplication's coefficients in the order of 10^{-2} except of multiplication's coefficient occurrents at marking transformed value y_0 is equal 3. The maximum difference between the calculated analytically transformed values and the transformed values received from the implemented algorithm is equal 2,89. In this case the implementation in XCV200BG352 occupies 402 from 2352 SLICE blocks (17% SLICE blocks) and 648 from 4704 LUT blocks (13% LUT). The maximum attainable frequency is equal 93,932MHz

The Lee's 1D-DCT algorithm was implemented with accuracy's representation of multiplication's coefficients in the order of 10^{-3}. Therefore theoretically the maximum difference between the calculated analytically transformed values and the transformed values received from the implemented algorithm shouldn't be larger than 2,04. In this case the maximum difference is equal 2,594. The maximum difference *MD* between the value of the input pixel and the value of the pixel's reconstruction is equal 2.

The implementation of the Lee's 1D-DCT algorithm with the accuracy in the order of 10^{-3} in the structure XCV200BG352 occupies 1277 from 2352 SLICE blocks (54% SLICE blocks) and 2237 from 4704 LUT blocks (47% LUT). The maximum attainable frequency for the implementation of the Lee's 1D-DCT algorithm is equal 35,668MHz.

Table 7 The consideration for a hypothetical input vector for Lee's with the accuracy in the order of 10^{-3}

x_n	y_n calculated analytically	y_n received from the implemented Lee's algorithm	The difference between y_n received and y_n calculated	x_n received from the implemented algorithm
x_0=252	y_0 = 645,5885	y_0 = 646	-0,41151	251
x_1 =250	y_1 = 55,33119	y_1 = 54	1,33119	249
x_2 =249	y_2 = 5,037243	y_2 = 5	0,037243	247
x_3 =247	y_3 = -27,3337	y_3 = -27	-0,33367	247
x_4 =207	y_4 = -22,6274	y_4 = -23	0,372583	207
x_5 =223	y_5 = -21,849	y_5 = -22	0,151043	223
x_6 =159	y_6 = 36,18185	y_6 = 36	0,181849	159
x_7 =239	y_7 = -32,817	y_7 = -22	0,183041	239

Table 8 The example for small differences between neighbouring pixels for Lee's algorithm with the accuracy in the order of 10^{-3}.

x_n	y_n calculated analytically	y_n received from the implemented Lee's algorithm	The difference between y_n received and y_n calculated	x_n received from the implemented algorithm
x_0 =191	y_0 = 530,3301	y_0 = 530	0,330086	190
x_1 =190	y_1 = 6,442323	y_1 = 6	0,442323	190
x_2 =189	y_2 = 0	y_2 = 0	0	189
x_3 =188	y_3 = 0,673455	y_3 = 0	0,673455	187
x_4 =187	y_4 = 0	y_4 = 0	0	186
x_5 =186	y_5 = 0,200903	y_5 = 0	0,200903	185
x_6 =185	y_6 = 0	y_6 = 0	0	184
x_7 =184	y_7 = 0,050702	y_7 = 0	0,050702	184

The Loeffler's 1D-DCT algorithm was implemented with accuracy's representation of multiplication's coefficients in the order of 10^{-3}. In this case the maximum difference between the calculated analytically transformed values and the transformed values received from the implemented algorithm is equal 2,40562. The maximum difference *MD*

between the value of the input pixel and the value of the pixel's reconstruction is equal 2.
The implementation of the Loeffler's 1D-DCT algorithm with the accuracy in the order of 10^{-3} in the structure XCV200BG352 occupies 508 from 2352 SLICE blocks (21% SLICE blocks) and 877 from 4704 LUT blocks (18% LUT). The maximum attainable frequency for the implementation of the Loeffler's 1D-DCT algorithm is equal 79,195MHz.

Table 9 The example for small differences between neighbouring pixels for the implemented Loeffler's algorithm with the accuracy in the order of 10^{-3}.

x_n	y_n calculated analytically	y_n received from the implemented algorithm	The difference between y_n received and y_n calculated	x_n received from the imple-mented algorithm
x_0 =191	y_0 = 530,3301	y_0 = 530	0,330086	190
x_1 =190	y_1 = 6,442323	y_1 = 6	0,442323	189
x_2 =189	y_2 = 0	y_2 = 0	0	190
x_3 =188	y_3 = 0,673455	y_3 = -1	1,673455	189
x_4 =187	y_4 = 0	y_4 = 0	0	186
x_5 =186	y_5 = 0,200903	y_5 = 1	-0,7991	185
x_6 =185	y_6 = 0	y_6 = 0	0	185
x_7 =184	y_7 = 0,050702	y_7 = 0	0,050702	185

Table 10 A case for the maximum difference between y_n received and y_n calculated for the Loeffler's algorithm with the accuracy in the order of 10^{-3}.

x_n	y_n calculated analytically	y_n received from the implemented algorithm	The difference between y_n received and y_n calculated	x_n received from the imple-mented algorithm
x_0 =8	y_0=101,82338	y_0 = 102	-0,17662	8
x_1 = 16	y_1=-51,53858	y_1 = -52	0,461416	14
x_2 = 24	y_2 = 0	y_2 = 0	0	24
x_3 = 32	y_3=-5,387638	y_3 = -4	-1,38764	30
x_4 = 40	y_4 = 0	y_4 = 0	0	41
x_5 = 48	y_5=-1,307223	y_5 = -1	-0,60722	48
x_6 = 56	y_6 = 0	y_6 = 0	0	57
x_7 = 64	y_7=-0,405619	y_7 = 2	-2,40562	63

4. SUMMARY

Comparing the parameters achieved by the implemented Lee's Chen's and Loeffler's algorithms with an accuracy in the order of 10^{-3}, better is the Loeffler's algorithm. At an angle of achieved maximum frequencies better is the Loeffler's algorithm because the maximum frequency for an accuracy's representation of multiplication's coefficients in the order of 10^{-3} is equal 79,195MHz and thereat accuracy for the Lee's algorithm the frequency is equal 35,668MHz and for the Chen's algorithm the frequency is equal 76,115MHz. As regards of an occupied resources the Loeffler's algorithm is more profitable. The implementation of this algorithm occupies 508 SLICE blocks and 877 LUT blocks in case of the accuracy in the order of 10^{-3}. However for the Lee's algorithm there are necessary 1277 SLICE blocks and 2237 LUT blocks. For the Chen's algorithm with the accuracy in the order of 10^{-3} there are necessary 574 SLICE blocks and 958 LUT blocks. For all algorithms with the accuracy in the order of 10^{-3} the accuracy of the value's reconstruction of pixels is the same because the maximum difference between *MD* the value of the input pixels and the value of the reconstructed pixels is equal 2. The argument appeals to us to advantage of the Chen's algorithm is the maximum difference between the calculated analytically transformed values and the transformed values received from the implemented algorithm. This difference is equal 1,182 for the Chen's algorithm. For the Lee's algorithm this maximum difference is equal 2,594 and for the Loeffler's algorithm is equal 2,40562.

REFERENCES

Bukhari K., Kuzmanov G., Vassiliadis S.: *DCT and IDCT Implementations on Different FPGA Technologies*, www.stw.nl

Chotin R., Dumonteix Y., Mehrez H.: *Use of Redundant Arithmetic on Architecture and Design of a High Performance DCT Macro-block Generator*, www.asim.lip6.fr

Heron J., Trainor D., Woods R.: *Image Compression Algorithms Using Re-configurable Logic*, www.vcc.com

Jamro E., Wiatr K.(2002) Dynamic Constant Coefficient Convolvers Implemented in FPGA, Lecture Notes in *Computer Science* – **no 2438**, Springer Verlag, pp. 1110-1113

Trainor D., Heron J., Woods R.: *Implementation of the 2D DCT Using a Xilinx XC6264 FPGA*, www.ee.qub.ac.uk

Woods R., Cassidy A., Gray J.: *VLSI Architectures for Field Programmable Gate Arrays: A Case Study*, www.icspat.com

Wiatr K. (1997) Dedicated Hardware Processors for a Real-Time Image Data Pre-Processing Implemented in FPGA Structure. Lecture Notes in *Computer Science* - **no 1311**, Springer-Verlag , vol. II, pp. 69-75

Wiatr K. (1998) Dedicated System Architecture for Parallel Image Computation used Specialised Hardware Processors Implemented in FPGA Structures. *International Journal of Parallel and Distributed Systems and Networks*, **vol. 1, No. 4**, Pittsburgh, pp. 161-168

Wiatr K., Jamro E. (2001) *Implementation of Multipliers in FPGA Structure*, Proc. of the IEEE Int. Symp. on Quality Electronic Design, Los Alamitos CA, IEEE Computer Society, pp. 415- 420