

A New Approach for Hardware Implementation of the 2D-IDCT Algorithm

Carlos Haas Costa, Jonas Bragagnolo, Ronaldo Hüsemann

PRAV - Laboratório de Projeto de Redes de Alta Velocidade

PIPCA - UNISINOS,

Av. Unisinos, 950

São Leopoldo - Brasil

E-mail: carloshaascosta@gmail.com, rhusemann@gmail.com

Abstract - The current standards of Digital TV (ATSC, DVB and ISDB) use MPEG-2 algorithm to reduce the amount of data stored and transmitted. This algorithm adopts Discrete Cosine Transform (DCT) and Inverse Discrete Cosine Transform (IDCT) to identify spatial redundancy allowing data compression e decompression in coders and decoders respectively. Due to high data rates of high-definition videos (HDTV), IDCT and DCT transforms in MPEG-2 HD decoders until now must be implemented in hardware. Current paper presents a practical structure to perform Loeffler IDCT algorithm in hardware, reducing consumed area and processing time. The results of this proposed structure are compared with traditional approaches showing the gain obtained.

Keywords - IDCT, DCT, FPGA, hardware implementation.

I. INTRODUCTION

Nowadays many current products use the MPEG-2 video compression standard to code and decode video information. DVD players, SDTV or HDTV – standard and high definition digital TV - set-top boxes, digital camcorders and internet video players are some examples of applications found in our lives, which make use of this standard. The idea of MPEG-2 algorithm is to reduce the amount of data, exploiting spatial and temporal redundancy. Pictures where it is only exploited spatial redundancy are compressed as intra frame (i-frames) and pictures with spatial and temporal redundancy are compressed as predicted or bidirectional frames (p or b-frames).

Discrete Cosine Transform (DCT) is used by MPEG-2 coders to identify spatial redundancy, allowing compression of intra frames. Its inverse method, IDCT (Inverse Discrete Cosine Transform) is implemented in MPEG-2 decoders to reverse the operation. Both transforms are substantial performance bottlenecks in their correspondent processes. For example, in a typical MPEG decode process IDCT generally represents 25% of total operations.

In order to support frame rates of up to 19Mbps, MPEG-2 High-Definition (1920x1080 pixels) decoders are implemented in hardware adopting approaches specially defined for that.

IDCT's processing time reduction impacts directly in the MPEG-2 compressing algorithm overall performance. In this paper, it is presented a practical solution to implement DCT

and IDCT methods in hardware reducing processing time and consumed resources, specially the number of multiplications. The proposed approach was implemented in Xilinx FPGA and the obtained results are compared with other traditional solutions.

II. INVERSE DISCRETE COSINE TRANSFORM

The Discrete Cosine Transform and its inverse algorithm were introduced in 1974 by Ahmed et al [1]. The implementation of an IDCT for a one-dimensional vector is given by equation (1), and, for a two-dimensional matrix in equation (2).

$$x_i = \frac{2}{N} \sum_{u=0}^{N-1} K_u X_u \cos \frac{(2i+1)u\pi}{2N} \quad (1)$$

$$x_{i,j} = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 K_u K_v X_{u,v} \cos \frac{(2i+1)u\pi}{16} \times \cos \frac{(2j+1)v\pi}{16} \quad (2)$$

With the large utilization on image processing, filtering and compression achieved by the transform, many algorithms have been proposed in order to minimize number of multiplications and stages, presenting solutions for one-dimensional transforms. However, in MPEG-2, an intra-frame data block is composed by matrices of 8x8 pixels, then, the most common method is to apply a one-dimensional IDCT over the original input matrix, transpose this data results and apply another one-dimensional IDCT. The final data is obtained transposing results of the second 1D-IDCT.

In 1989, Loeffler et al. [2] introduced a new practical fast method. In this new approach, for an 8-point 1D-IDCT, only 11 multiplications and 29 additions are required, and if more precision are necessary, a modification is proposed adding one multiplication in its odd part. It increases the number of multipliers, but results in an algorithm with no multiplications in sequence. It represents a good solution for architectures based on integers and because that it was selected for this implementation.

III. IMPLEMENTATION

The proposed approach was designed for the purpose of reduce area and processing time of IDCT hardware implementations for MPEG-2 (considering 8x8 pixels blocks). The top-level module, called 2D-IDCT, is subdivided in three parts. In data path, the first module performs a Loeffler 1D-IDCT. The input data is directly connected on this module. Based on Loeffler algorithm, it processes just one-dimensional vectors. The data format is a 96 bits vector, divided as eight inputs of signed 12 bits integers. The Loeffler algorithm was implemented in order to perform the calculations in four steps. To reach a better performance, and to exploit the advantages of a dedicated application, each step was adapted to implement a pipelined architecture. Therefore, each calculation step is performed in a pipeline stage, and after five pulses of clock, the job is done. As a gain of this organization, while one column is finally processed, other three columns have being processed in parallel (up to four columns simultaneously) in different pipeline stages.

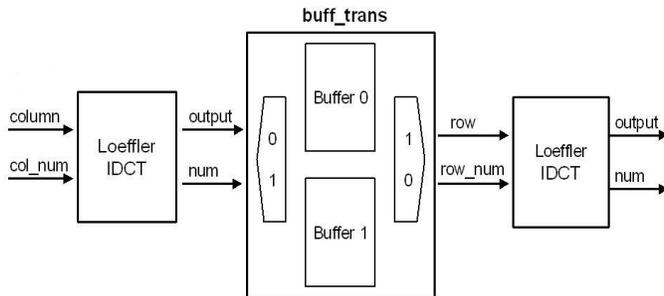


Fig. 1. The three modules – 1D-IDCT – Buffer – 1D-IDCT.

As can be observed in the figure above, the output of the first module feeds a transposition buffer. The goal for this intermediary buffer is to rearrange the first module output data, oriented in columns, in a transposed matrix, oriented in rows, and supply correct input data to the second Loeffler transform. Because buffer stores transposed information, the first line is available just after an entire matrix have been processed. Then, to keep two IDCT modules operating in parallel, writing and reading, a second buffer was included in the project. While the second IDCT module reads data from buffer “0”, the first IDCT module write its processed data on buffer “1”. For next matrix, occurs an inversion of this process (second IDCT reading from buffer “1” and first IDCT writing in buffer “0”), and the system works in a continuous and synchronous process.

The second 1D-IDCT operates as the first one, and has the same internal organization. The differences between both 1D-IDCTs modules are related with precision and data format (considering input and output). Adjustments have been made in order to fit the results of implemented method in accordance with IEEE Reference Implementation [3]. The input of the second IDCT has the same data format of first IDCT output and the same occurs with the transposition buffer (input/output), a 136 bits vector, divided as eight signed integers with 17 bits. The final output is an 80 bits vector, composed by eight signed integers values of 10 bits.

In a global analysis of the 2D-IDCT module functioning, if we assume eight columns for one data block and empty pipeline, the first row is at the output after seventeen pulses, and the next seven rows came in sequence (one row by each clock). It represents a total latency of 24 cycles to process an entire matrix. With full pipeline operation, the system output deliver one row for cycle and a full matrix is available every eight pulses of clock.

IV. RESULTS

The proposed method was developed in VHDL language and implemented in a Xilinx Virtex II Pro XC2VP30 FPGA. After synthesis, a total number of 24 multipliers were used, and, in simulation, the obtained data results had meet IEEE Specifications, in all tests.

Figure 2 presents a comparison among this approach and another IDCT hardware implementations: Chen method [4] and modified Loeffler IDCT [5].

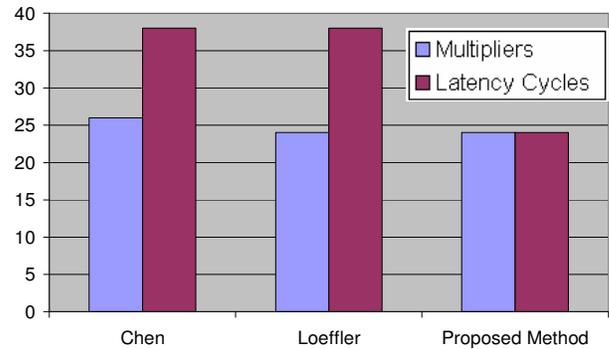


Fig. 2. Graphic comparison.

V. CONCLUSIONS

This paper presents a new approach for hardware implementation of the IDCT for use in Standard and High Definition MPEG-2 decoders. With an optimized pipeline architecture based on Loeffler algorithm, excellent results have been achieved, minimizing area and number of cycles, considering the same data precision. After synthesis and simulation, could be proven the correct functioning of the system, which fulfills MPEG-2 decoder requirements.

REFERENCES

- [1] Ahmed, N.et al.: *Discrete Cosine Transform*. IEEE Transactions on Computer, January 1974, pp.90-93.
- [2] Loeffler, C.; Ligtenberg, A.; Moschytz, G.S.: *Practical Fast ID-DCT Algorithms with 11 multiplications*. IEEE Proc. Int'l Conf. on Acoustics, Speech and Signal Processing 1989 (ICASSP'89), pp. 988-991.
- [3] *IEEE Standard Specifications for the Implementations of 8x8 Inverse Discrete Cosine Transform*. IEEEStd1180-1990, 1991.
- [4] Chen, W.H.; Smith, C.H.; Fralick, S.C.: *A Fast Computational algorithm for the Discrete Cosine Transform*. IEEE Transactions on Communications, Vol. COM-25, No. 9, September 1977, pp. 1004-1009.
- [5] Bukhari et. al: *DCT and IDCT Implementation on Different FPGA Technologies*. ProRISC2002, Netherlands, nov. 2002, pp 232-235