DCT and IDCT Implementations on Different FPGA Technologies

Khurram Bukhari, Georgi Kuzmanov and Stamatis Vassiliadis
Computer Engineering Lab, Delft University of Technology
P.O. Box 5031, 2600 GA Delft, The Netherlands
Phone: +31(0)15 278 7364 Fax: +31(0)15 278 4898
E-mail: G.Kuzmanov@ET.TUDelft.NL

Abstract—**In this paper we investigate hardware implementations of 8x8 DCT and IDCT on different FPGA technologies using the modified Loeffler algorithm. The investigations involved simulations and synthesis of VHDL code utilizing recent FPGA families of Xilinx, Altera, and Lucent. We aimed at achieving the most demanding real-time requirements of some standardized frame resolutions and rates. Synthesis results for 8-point DCT/IDCT implementations indicate operating frequencies of 50 MHz, 60 MHz, and 22 MHz for the investigated Xilinx, Altera and Lucent FPGA chips, respectively. These frequencies allow 2193 SIF and 100 HDTV frames to be processed by the Xillinx FPGA. The resulting frame processing rates for Lucent are 877 and 40 for SIF and HDTV, while for Altera they are 647 and 29, respectively. Results indicate that the investigated FPGA implementations would speed DCT based compression algorithms up to frame rates well above the real-time requirements of SIF, CCIR-TV and HDTV frame formats.**

Keywords— **Discrete Cosine Transform (DCT); VHDL; FPGA; Reconfigurable Processor**

I. INTRODUCTION

The discrete cosine transform (DCT) and the inverse discrete cosine transform (IDCT) are substantial performance bottlenecks in the contemporary visual data compression algorithms (JPEG, MPEG, etc.). Implementing DCT/IDCT, as an ASIC is a design solution, which meets the real time processing requirements, but it lacks flexibility. Another, more flexible solution, still capable to achieve real-time performance, is the reconfigurable realization of the transforms. Such DCT/IDCT implementations mapped on FPGAs will be discussed here.

In this paper we investigate implementations of 8x8 DCT and IDCT hardware units mapped on various FPGA technologies using the modified Loeffler algorithm [1], [2]. We aimed at achieving the most demanding real-time requirements of some standardized frame resolutions such as the Source Input Format (SIF) and the International Consulting Committee on Radio and Television (CCIR-TV). Our particular interest was in performance improvements for the High Definition Television (HDTV) standard. The investigation involved generation, simulation and synthesis of VHDL code using *ModelSim*™ and *LeonardoSpectrum*™ as design environments. During the design process we used VHDL libraries for the recent FPGA families of Xilinx, Altera, and Lucent. Synthesis results for an 8-bit IDCT implementation indicate:

- 214 Configurable Logic Block slices and 22 multipliers in Xilinx Virtex II Technology; 1482 Altera Acex-1K Logical Cells; 1488 Lucent's Orca Look-UP Tables.
- Operating frequencies of 50 MHz for Xilinx, 60 MHz for Altera, and 22 MHz for Lucent.
- 2193 SIF and 100 HDTV frames per second to be processed by Xillinx Virtex II FPGA; 877 SIF and 40 HDTV frames per second processing speed by Lucent's Orca; 647 SIF and 29 HDTV frames per second throughput by Altera's Acex.

Synthesis and simulation results prove that the investigated FPGA implementations can speed up DCT to frame rates well above the real-time requirements of SIF, CCIR-TV and HDTV.

The remainder of the paper is organized as follows. In Section II, we briefly describe some DCT/IDCT theoretical background and the modified Loeffler algorithm. In Section III we discuss the methodology of the experimentation and some hardware implementation issues. Experimental results are reported in Section IV. Finally, concluding remarks are presented in Section V.

II. BACKGROUND

DCT and IDCT have been widely used in video data compression standards. The decorrelation and energy compaction properties of the transform have been exploited to achieve high compression ratios in MPEG and JPEG. The N-point 1-D DCT is defined by [4]:

$$X(k) = \frac{2}{N} c_k \sum_{n=0}^{N-1} y(n) \cos\left[\frac{(2n+1)k\pi}{2N}\right], \; k = 0,1,\ldots, N-1;$$

and the N-point 1-D IDCT is defined by:

$$y(n) = \frac{2}{N} \sum_{k=0}^{N-1} c_k X(k) \cos\left[\frac{(2n+1)k\pi}{2N}\right], \; n = 0,1,\ldots, N-1;$$

where $c_k = \begin{cases} 1/\sqrt{2}, k = 0 \\ 1, k \neq 0 \end{cases}$.

DCT and IDCT are highly computational intensive, which creates prerequisites for performance bottlenecks in systems utilizing them. To overcome this problem, a number of algorithms have been proposed for more efficient computations of these transforms. In our experiments we use an 8-point 1-D DCT/IDCT algorithm, proposed by van Eijdhoven and Sijstermans [1]. This algorithm is a slight modification of the original Loeffler algorithm [2], which provides one of the most computationally efficient 1-D DCT/IDCT calculations. The modified Loeffler algorithm for calculating 8-point 1-D IDCT is illustrated in Figure 1.

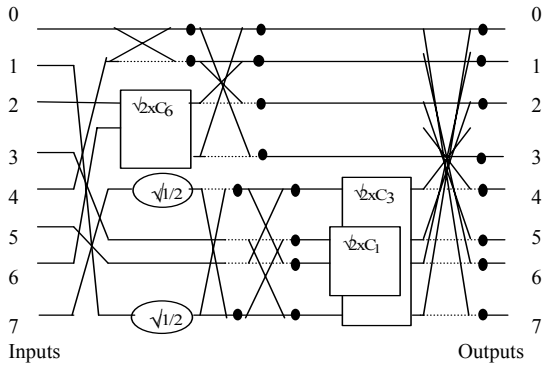
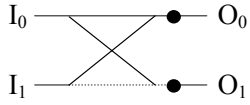


Figure 1 The 8-point IDCT - modified Loeffler algorithm

The round block in Figure 1 signifies a multiplication by √1/2. The butterfly block and the equations associated to it are presented in Figure 2.

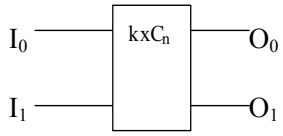


$O_0 = I_0 + I_1$

$O_1 = I_0 - I_1$

Figure 2 The Butterfly

The rectangular block depicts a rotation, which transforms a pair of inputs [I_0, I_1] into outputs [O_0, O_1]. The symbol and associated equations are depicted in Figure 3.



$$O_0 = I_0 k\cos\left[\frac{n\pi}{16}\right] - I_1 k\sin\left[\frac{n\pi}{16}\right] = C'_n I_0 - S'_n I_1$$

$$O_1 = I_0 k\sin\left[\frac{n\pi}{16}\right] + I_1 k\cos\left[\frac{n\pi}{16}\right] = S'_n I_0 + C'_n I_1$$

Figure 3 The rotator and its associated equations

The implementation of the rotator depicted in Figure 4 utilizes four multipliers and two adders to shorten critical path and improve numerical accuracy. This direct implementation has been proven to be ideal for fixed-point arithmetic [5]. Indeed, some other implementations of the rotator are possible, e.g., with three multipliers and three adders. These alternative designs, however, have longer critical paths and involve initial additions, which may lead to overflows and may affect the accuracy of the calculations.

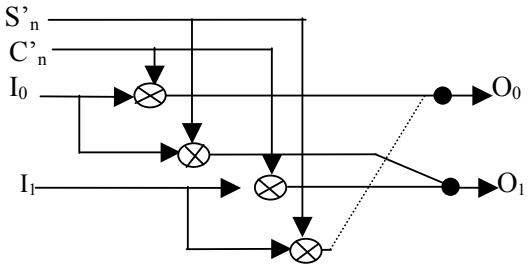


Figure 4 Implementation of the rotator for IDCT

We depict the algorithm of 8-point DCT in Figure 5.

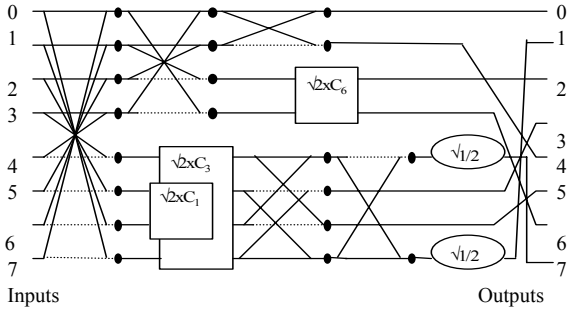


Figure 5 The 8-point DCT - modified Loeffler algorithm

The functionality of the rotator in DCT is slightly different than in IDCT, while the round block and the butterfly are exactly the same.

The DCT rotator block equations are:

$$O_0 = I_0 k\cos\left[\frac{n\pi}{16}\right] - I_1 k\sin\left[\frac{n\pi}{16}\right] = C'_n I_0 - S'_n I_1$$

$$O_1 = -I_0 k\sin\left[\frac{n\pi}{16}\right] + I_1 k\cos\left[\frac{n\pi}{16}\right] = -S'_n I_0 + C'_n I_1$$

In video data compression standards, the 2-D DCT/IDCT is defined. One possible approach to compute the 2-D DCT/IDCT is the standard row-column separation. In this approach, the 1-D transform is applied to each row. On each column of the result 1-D transform is performed again, to produce the final result of the 2-D DCT/IDCT. In our experiments we use this strategy.

III. Methodology of the Experimentation

Our experiments involve processing video data with different frame formats. We have chosen the SIF, CCIR-TV and the HDTV formats, since they have been considered by many video compression standards. The frame resolutions for SIF, CCIR-TV and HDTV are 352x288, 525x720 and 1152x1926, respectively. We have written synthesizable VHDL models of two units, one describing 1-D DCT and the other – 1-D IDCT. The designs have been implemented according to the modified Loeffler algorithm. We both simulated and synthesized the VHDL models for three different FPGA technologies, namely Virtex II, Acex-1K and Orca using the following design tools:

- *ModelSim*™ SE/EE from Model Technology, version 54.b, revision 2000.06, for simulating the VHDL source code;
- *LeonardoSpectrum*™ from Exemplar, version v2000.1a2.75, for the synthesis of VHDL source code.

For the design of DCT we considered 8-bit input data for consistency with the 8-bit color presentation in visual data compression standards like MPEG and JPEG. The output data width was designed to be 10-bit. Similarly, 10-bit inputs and 8-bit outputs were considered for the IDCT design. The row-column separation strategy was used to compute the 2-D DCT/IDCT. As we have used 8-point 1-D DCT and IDCT, the FPGA I/O ports delay, reported by the synthesis software, is in essence the data processing delay for 8 pixels. Implementing matrix transposition without extra delay, we can multiply the 8-point 1-D DCT I/O latency by 16 to calculate the latency of the 8x8 DCT transform. This is in essence the processing latency for one 8x8 pixel block. Given this latency, we can easily calculate the tame, required to transform all 8x8 blocks in any video frame for the selected formats - SIF, CCIR-TV and HDTV. Frame processing rate (frames per second) of the implemented DCT/IDCT was the main criterion used to estimate and compare the FPGA mappings on the three different technologies.

IV. Experimental Results

Synthesis results for 8-point DCT and IDCT units are included in Table I. These results indicate that the Xilinx FPGA implementations of DCT/IDCT can process higher numbers of frames per time unit, compared to the other two FPGA technologies. One reason for this considerable data processing speed is the utilization of coarse-grain reconfigurable resources available in the Virtex II FPGA. In particular, the usage of hardwired multipliers and fast carry chains lead to a severe acceleration of the implemented computations.

TABLE I

SYNTHESIS RESULTS FOR DIFFERENT FPGA TECHNOLOGIES

Implemented function	DCT	IDCT
Lucent ORCA-3C/3T series FPGA		
Max. Clock frequency (MHz)	22	22
No. of LUTs used	1320	1488
No. of SIF frames per second	896	877
No. of CCIR-TV frames per second	1219	214
No. of HDTV frames per second	40	40
Altera Acex-1K series FPGA		
Max.Clock frequency (MHz)	16	16
No. of LCs used	1303	1482
No. of SIF frames per second	647	647
No. of CCIR-TV frames per second	158	158
No. of HDTV frames per second	29	29
Xilinx Virtex-II series FPGA		
Max.Clock frequency (MHz)	54	60
No. of CLBs used	203	214
No. of multipliers used	22	22
No. of SIF frames per second	2193	2469
No. of CCIR-TV frames per second	536	600
No. of HDTV frames per second	100	112

We were particularly interested whether the FPGA implementations of the designs would be fast enough to meet the real time requirements of the selected video formats. For SIF and CCIR-TV, the requirements for real time processing rates are 25 frames per second. It can be observed in Figure 6 that Xilinx FPGA implementations process the highest number of SIF frames per second. The other two FPGA technologies, using finer-grain resources, although slower, are capable of processing SIF frames at speeds, much higher than the required real-time rates (25 frames per second).

Therefore, in this case the advantages of the Virtex II technology can not be utilized efficiently.

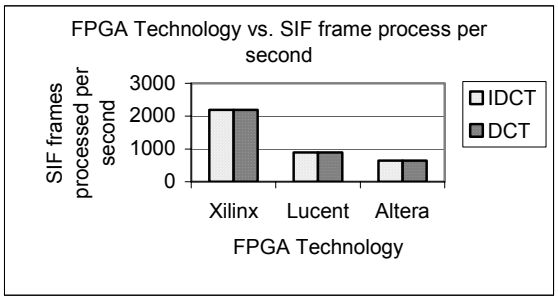


Figure 6 Comparing different FPGA technologies to SIF frames processing per second

Regarding CCIR-TV format, performance results impose similar conclusions (see Figure 7). All FPGA technologies provide DCT/IDCT processing speeds well above the required real time values.

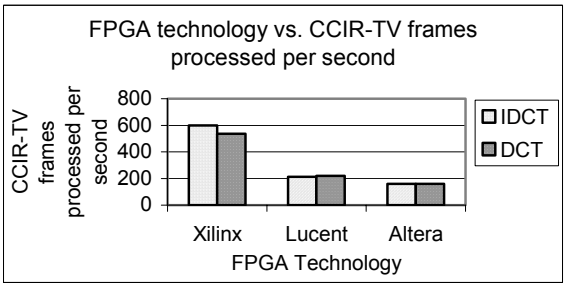


Figure 7 Comparing different FPGA technologies to CCIR-TV frames processed per second

The advantages of using coarse-grain reconfigurable resources for speeding up the DCT and IDCT operations are illustrated in Figure 8. Only Xilinx FPGA is able to process twice the rate required by HDTV, which is 50 frames per second. The other two FPGA can not achieve the requirements for real time processing.

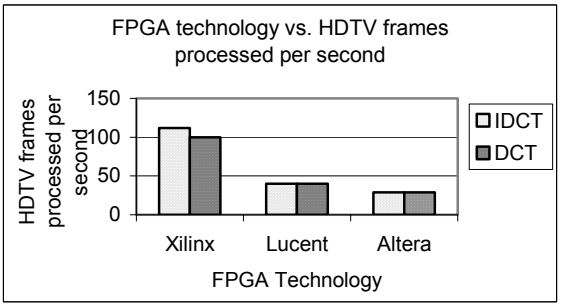


Figure 8 Comparing different FPGA technologies to HDTV frames processed per second

V. CONCLUSIONS AND FUTURE WORK

In this paper, we reported the results from an investigation on reconfigurable implementations of DCT/IDCT mapped on different FPGA technologies. Synthesis and simulation results from the experiments indicate that real-time requirements of SIF, CCIR-TV and even HDTV can be met by the implemented DCT and IDCT designs. From the reported results we can conclude that all investigated FPGA implementations can speed up DCT based compression standards dramatically. However, for computationally intensive algorithms like DCT/IDCT better results can be achieved by coarser-grained reconfigurable logic, like the one realized by the Virtex II technology of Xilinx. In future, we intend to integrate the investigated DCT/IDCT designs into a custom computing machine organization, called MOLEN [3]. The MOLEN processors utilize microcode to control both reconfiguration and execution process of the reconfigurable unit. Our primary goal will be to investigate the influence of FPGA reconfiguration time on the overall performance of the system.

VI. ACKNOWLEDGMENTS

This research has been supported by PROGRESS, the embedded systems research program of the Dutch organization for scientific research NWO, the Dutch Ministry of Economic Affairs, and the Technology Foundation STW (project AES.5021).

REFERENCES

[1] T.J. van Eijndhven and F.W. Sijstermans, "Data Processing Device and method of Computing the Cosine Transform of a Mtrix", *PCT Patent WO 99948025, to Koninklijke Philips Electronics*, World Intellectual Property Organization, International Bureau, 1999.

[2] C. Loeffler and A. Lightenberg, "Practical fast 1-D DCT algorithms with 11 Multiplications," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (ICASSP '89), Scotland, May 1989, pp. 988-991.

[3] S. Vassiliadis, S. Wong and S. Cotofana, "The MOLEN ρμ-coded processor", *Proceedings of the 11th International Conference on Field Programmable Logic and Applications (FPL)*, 2001.

[4] K.R. Rao and P.Yip, "Discrete Cosine Transform. Algorithms, Advantages, Applications", Academic Press, San Diego, California, 1990.

[5] M. Sima, S. Cotafona, S. Vassiliadis and J.T.J. van Eindhoven, " 8x8 IDCT Implementation on an FPGA-augmented Trimedia, "*IEEE Symposium on FPGAs for Custom Computing Machines* (FCCM 2001), California, April 2001.

[6] K.Bukhari. Visual data transforms comparison, M.Sc. thesis, Delft University of Technology, 1-68340-28 (2002)-05