Low-Complexity Multi-Purpose IP Core for Quantized Discrete Cosine and Integer Transform

Chi-Chia Sun†, Philipp Donner and Jürgen Götze
Information Processing Lab, Dortmund University of Technology, Germany
†e-mail: chichia.sun@tu-dortmund.de

***Abstract*— In this paper a low-complexity and highly-integrated IP Core for image/video transformations is presented. It is possible to perform quantized 8×8 DCT and quantized 8×8/4×4 integer transforms on the presented reconfigurable architecture using only shift and add operations. The XVID experimental and FPGA synthesis results show that the proposed architecture not only achieves multiplierless video transformations efficiently, but also retains good transformation quality. It is worth noticing that the proposed IP Core is very suitable for low-complexity and multi-purpose Video CODECs in SoC designs.**

***Index Terms*— DCT, QDCT, integer transform, CORDIC, DCIT, QDCIT, low power, reconfigurable architecture, FPGA**

I. Introduction

As the demand for multi-function devices has been growing explosively in recent years, new challenges have been posed in some emerging design issues, such as low-power, quality awareness and multi-standard integration. Therefore, we present a multi-standard image/video transformation architecture. So far, the Discrete Cosine Transform (DCT) is the main component of many modern Image/Video compression standards and applications (*e.g.,* JPEG, MPEG4, H.26X and so on [1], [2]). Recently, the DCT is replaced by an integer transform in the H.264-Standard which uses block sizes of 4×4 Pixels. For adding profiles for HD-Videos, an integer transform using block sizes of 8×8 has been added [3].

In this paper we present the design of a reconfigurable transformation IP core which can perform the multiplierless 1-D 8-point DCT and the 1-D 8-point/4-point integer transforms for multi-standard Video CODECs by reusing the architecture of our previous CORDIC based Loeffler DCT (CLDCT) [4]. In this work we have successfully implemented a low-power and high-quality CLDCT which not only reduces the computational complexity from 11 multiplier and 29 add operations to 38 add and 16 shift operations but also obtains a transformation quality as good as the original Loeffler DCT. Here, we present the integration of 8-point/4-point integer transforms in this CLDCT, such that a reconfigurable architecture for Discrete Cosine and Integer Transform (DCIT) is obtained, which can be used for multi-functional SoC designs.

Incorporating the quantization into the DCT transformation, resulting in the Quantized DCT (QDCT), has been another important issue to improve the computational complexity. In the literature, Alen Docef et al. [5] proposed a joint implementation of Chen's DCT [6] and Quantization (*i.e.*, a conventional QDCT design). Later, Hanli Wang et al. [7] merged the quantization process represented by a quantization matrix that has variable quantization step sizes into Chen's DCT. They called it Novel QDCT (NQDCT). However, both of these QDCT designs still need multipliers to perform DCT and Quantization. Even if the multiplier could be replaced by Canonical Signed Digit (CSD) representation, the QDCT and NQDCT still need more than 300 add operations in the worst case.

In order to realize an efficient QDCT architecture, we present a CORDIC Scaler (C-Scaler) by involving limited CORDIC compensation iterations. This C-Scaler has four stages and requires 8 add and 10 shift operations. The combination of two 1-D DCITs, a row-column transposition memory and four C-Scalers forms a 2-D Quantized DCIT (QDCIT) which requires only 120 add and 80 shift operations.

The final architecture can not only perform multiplierless 8×8 QDCT but contain reconfigurable modules such that it can also perform 8×8/4×4 quantized integer transforms. Furthermore, it also retains the good transformation quality compared to the Loeffler DCT in terms of PSNR results.

This paper is organized as follows. Section II briefly introduces the algorithms of the DCT, CLDCT and the proposed reconfigurable DCIT for multiple transformations. In Section III, we will further present the proposed C-Scaler architecture for the multiplierless QDCIT transformation. The experimental and synthesis results are shown in Section IV. Section V concludes this paper.

II. DCT algorithms

A. The DCT Background

An 8-point 1-D DCT can transform 8 samples from spatial domain $f(x)$ into frequency domain $F(k)$ as follows:

$$F(k) = \frac{1}{2}C(k)\sum_{x=0}^{7} f(x)\cos[\frac{(2x+1)k\pi}{16}]$$
$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if k = 0} \\ 1 & \text{otherwise.} \end{cases} \quad (1)$$

A commonly used approach to construct 2-D DCT is the row-column decomposition method. The decomposition performs a row-wise 1-D transform followed by another column-wise 1-D transform with the intermediate transposition. This decomposition approach has two advantages. First, the computational complexity is significantly reduced. Second, the

original 1-D DCT can be easily replaced by different DCT algorithms.

B. The CORDIC based Loeffler DCT

Recently, we have presented a simplified 1-D 8-point CLDCT which requires only 38 add and 16 shift operations [4]. We took the original Loeffler DCT as our starting point for optimization, because the theoretical lower bound of the number of multiplications required for the 1-D DCT had been proven to be 11 [8], [9]. Furthermore, the most commonly used DCT-based CODECs for signal processing are usually followed by a quantizer. In this regard we have skipped some CORDIC iterations without losing the visual quality and shifted the compensation steps to the quantization table for reducing the computational complexity.

First of all, let us briefly review the CORDIC algorithm which rotates a vector (x, y) by an angle θ. The circular rotation angle is described as [10], [11]

$$\begin{aligned} \theta \quad &= \sum_i \sigma_i \cdot \tan^{-1}(2^{-i}) \\ &\text{with } \sigma_i \in \{1, -1\},\ i = 0, 1, 2, \ldots \end{aligned} \tag{2}$$

Then, the vector rotation can be performed iteratively as follows:

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i \cdot y_i \cdot 2^{-i} \\ y_{i+1} &= y_i + \sigma_i \cdot x_i \cdot 2^{-i}. \end{aligned} \tag{3}$$

In Eq. 3, only shift and add operations are required to perform the operation. Furthermore, the results of the rotation iterations need to be compensated (scaled) by a compensation factor s. This can be done by using the following iterative approach:

$$\begin{aligned} x_{i+1} \quad &= x_i(1 + \gamma_i \cdot F_i) \\ y_{i+1} \quad &= y_i(1 + \gamma_i \cdot F_i) \\ &\text{with } \prod_i (1 + \gamma_i \cdot F_i) \cong s \\ &\text{and } \gamma_i \in \{0, 1, -1\}, F_i = 2^{-i}. \end{aligned} \tag{4}$$

Eq. 3 and Eq. 4 describe the CORDIC circular rotation and compensation respectively. Implementing Loeffler DCT with the CORDIC circular rotation method by ignoring some unnoticeable iterations and shifting the compensation steps of each angle to the quantizer yields the simplified CLDCT as shown in Fig. 1. Fig. 1 shows the RTL flow graph where the white circles are adders and the dark circles are shifters, including eight scaling factors incorporated into the quantization table. This DCT architecture not only has a similar computational complexity as the binDCT [12] but also keeps the transformation quality as good as Loeffler DCT. More detailed information about this low-power and high-quality DCT can be found in [4].

C. Integrating the 8×8/4×4 integer transforms

In order to combine the integer transforms and the DCT transformation, we have remodeled our CLDCT's flow graph as shown in Fig. 2. Five reconfigurable modules are required for executing both the multiplierless DCT and integer transforms.

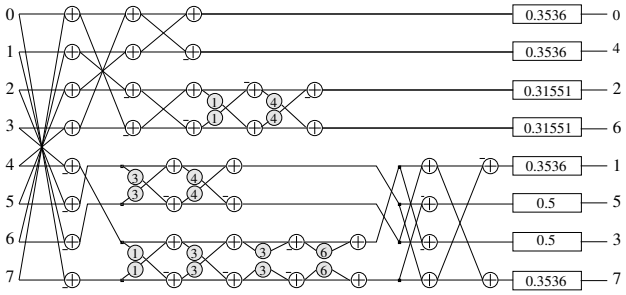


Fig. 1: Flow graph of an 8-point CORDIC based Loeffler DCT architecture.

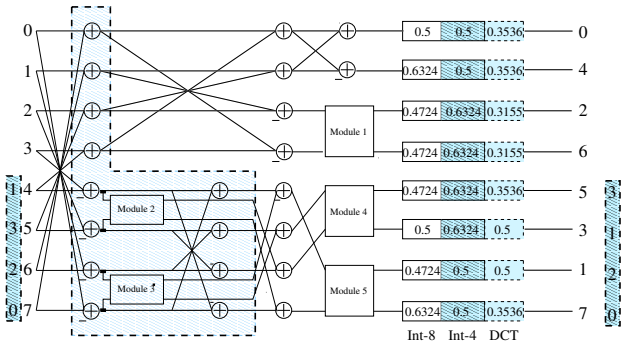
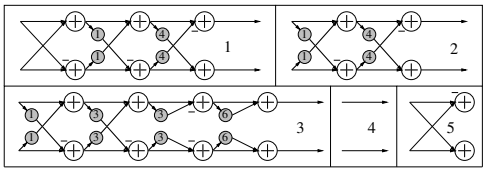


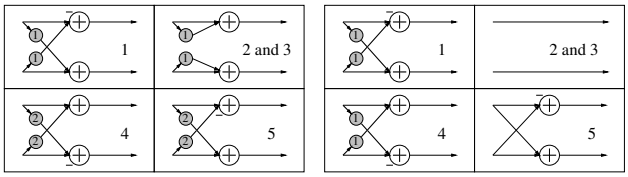
Fig. 2: Flow graph of an 8-point DCIT Transform with five reconfigurable modules for multiplierless DCT and integer transforms.

The proposed DCIT architecture can execute an 8-point DCT transform with a dedicated sub flow graph as shown in Fig. 3(a) where Module 1-3, 5 show the original operations for CLDCT and Module 4 is a bypassing circuit. The required scaling values are still identical to Fig. 1. We have expanded the configurable modules in order to perform the 8-point integer transform on the resulting hardware as shown in Fig. 3(b). In detail, in the upper part of the flow graph, no further operations are required and a simplified Module 1 with two shifters and two adders is used. In the lower part of the flow graph we added six adders and four shifters, where four adders are bordered by the dashed lines in Fig. 2. Another two shifters are added to Module 5, and two shifters/adders are added to Module 4. The needed scaling values for the 8-point integer transform are given in the light boxes at the end of the flow graph.

Next, we have also implemented two 4-point integer transforms by reusing the flow graph which only requires three configurable modules and two bypassing modules. Fig. 3(c) shows the related sub flow graphs for the modules 1-3 of the 4-point integer transform. First of all, the first stage of butterfly operations, modules 2-3 and four additional adders which are bordered by dashed lines in Fig. 2 have to be bypassed. The first 4-point integer transform is executed by the upper part of the flow graph and the second 4-point integer transform is executed by the lower part. Additionally the order of the input values has to be changed too. The different order of the input/output and scaling values are represented by the highlighted numbers and the dark boxes.



(a) Function of the modules for an 8-point CLDCT.



(b) Function of the modules for an 8-point integer transform.

(c) Function of the modules for a 4-point integer transform.

Fig. 3: Three sub flow graphs of the modules of Fig. 2.

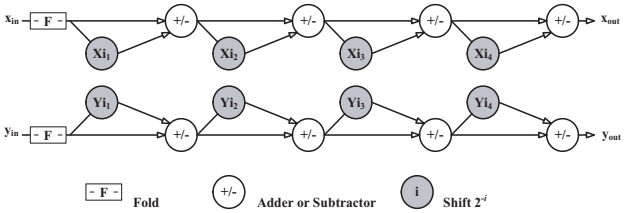


Fig. 4: RTL view of the C-Scaler with four unfolded CORDIC compensation stages.

In short Fig. 2 shows an integration of one 8-point DCIT, two 4-point integer transforms and one 8-point integer transform with five reconfigurable modules as shown in Fig. 3. The proposed architecture is able to perform these integer transforms and the DCT with a very small additional expense of hardware.

III. The proposed C-Scaler

As we mentioned before, QDCT is a rather efficient architecture to reduce the transformation complexity compared to the conventional one (*i.e.*, Quantization follows the 2-D DCT.). However, direct implementation of the corresponding quantizer by multipliers with the CSD representation is not ideal for VLSI implementation. Hence besides CSD representation, there is another rather simple and elegant way to approximate multiplierless QDCT by utilizing the compensation phase of the CORDIC algorithm with sequential shift and add operations [13].

So, let us first consider the CORDIC compensation factors we have shifted at the end of the flow graph of Fig. 1. Basically, they are deduced from Eq. 4, as several sequential radix-2 based shift and add operations are able to approximate the multiplication efficiently. On the other hand, since the quantization phase behind the DCT for video compression are usually fixed steps (*i.e.*, MPEG4 quantization "method 2" has 2-31 steps.), we could first combine the scaling factors and quantization together by utilizing the CORDIC compensation circuit, and then store the pre-computed compensation parameters in a look–up table.

According to the dependence flow graph of the unfolded CORDIC architecture presented in [13], we have implemented

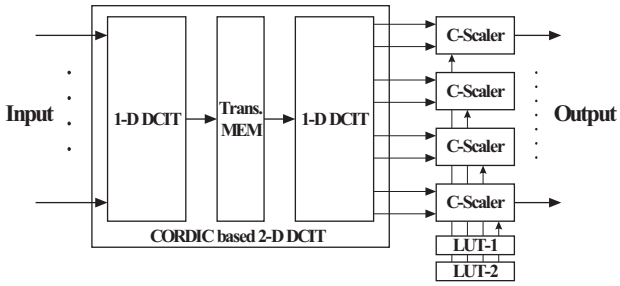


Fig. 5: The proposed framework of our CORDIC based 2-D QDCIT with four C-Scalers and two dedicated look-up tables.

a CORDIC Scaler (C-Scaler) with four unfolded compensation stages as shown in Fig. 4. Each compensation stage has a pair of add and shift operations to perform one CORDIC compensation step in Eq. 4, whereby looking into pre-computed parameters in the look–up table. For example, when the quantization step is 4 from MPEG-4 and scaling factor is $0.31551 \times 0.31551 = 0.09954$ from DCT. The C-Scalar will first fold the scaling value $0.09954/4 = 0.24885$ by shifting two bits and then apply subsequent pre–computed approximated iterations. In this way, the number of iterations could be reduced from 8$\sim$10 times to 4 times in average. Finally, each C-Scaler requires only 8 add and 10 shift operations to approximate the corresponding mul/div operation as illustrated.

Fig. 5 shows a block diagram of a CORDIC based 2-D QDCIT, including four C-Scalers. In this framework, there is a 1-D DCIT followed by a row-column transposition memory, then followed by another 1-D DCIT to perform the 2-D multiplierless image/video transformations. After that, there are four C-Scalers with two dedicated look-up tables (*i.e.* one is for CLDCT and the other one is for integer transform).

IV. Experimental Results

A. FPGA Implementation

Table I summarizes the analyses for each architecture in the number of operations. At first, the conventional Q+DCT requires 40 mul, 52 add and 8 shift operations to perform 2-D QDCT. For the 2-D Novel QDCT 32 mul, 52 add and 8 shift operations are required. However, both methods are unfriendly to VLSI-implementation due to the required multiplications. Even if we use the CSD representation to carry out the multiplications, the Novel QDCT still needs more than 300 add operations. The presented 2-D CLDCT with four C-Scalers reduces the computational complexity dramatically to only 108 add and 72 shift operations. Furthermore, it needs additional 12 add, 10 shift and 56 multiplex operations for integration of the integer transforms. Of course, the QDCIT requires extra multiplexers for performing the multi-functional transformations, but it's own flow graph feature makes it suitable for VLSI implementation.

We have modeled a full-pipelined CORDIC based 2-D QDCIT in VHDL including two 1-D DCITs, four C-Scalers, one transposition memory and two look-up tables. We have also synthesized it into the Virtex-II Pro XC2VP30 FPGA

TABLE I: Complexity for each 2-D QDCT architecture.

Operation / Type	Mul	Add	Shift	Mux
Conventional Q+DCT [5]	40	52	8	0
Novel QDCT [7]	32	52	8	0
Novel QDCT (CSD)	0	308	8	0
CORDIC based QDCT	0	108	72	0
CORDIC based QDCIT	0	120	80	56

device by Xilinx ISE 9.1, where Table II shows the detailed logic utilization of synthesis results. First, the maximal combinational path delay which came from the C-Scaler component is down to $6.73ns$ and results in a maximum frequency around 149 MHz. Second, the integration can reduce the overall area size compared to conventional methodologies. It should be also noticed that the wordlength is 12-bits for 1-D DCT and 14-bits for 2-D DCT in this implementation.

TABLE II: Logical utilization of the 1-D DCIT, four C-Scalers and the 2-D QDCIT in a FPGA implementation.

	1-D DCIT	C-Scaler×4	2-D QDCIT	
Slice Registers	874	796	3736 / 27392	13%
Occupied Slices	982	2050	5729 / 13696	41%
4 input LUTs	1607	3921	10310 / 27392	37%
Frequency	196 Mhz	149 Mhz	149 Mhz	
Maximal Delay	5.10 ns	6.73 ns	6.73 ns	

B. Performance in MPEG4-XVID

We have tested the proposed 2-D QDCIT architecture with the video coding standard MPEG4 by using a publicly available XVID CODEC software [2]. The default DCT algorithm in the CODEC of the selected XVID implementation is based on Loeffler's factorization using floating-point multiplications and MEPG4 Quantization "method 2". In this part we have implemented the NQDCT and our proposed QDCIT into the XVID software, and simulated with some well-known video sequences to further display the ability of the proposed approach.

Fig. 6 illustrates the average PSNR results of the foreman CIF video test sequence from low to high bitrates in XVID. It is very clear that the proposed architecture performs very close to the original Q+DCT design and almost same as the NQDCT.

V. Conclusion

In this paper, a low-complexity and highly-integrated QDCIT based on the CORDIC architecture is presented. The proposed 2-D QDCIT architecture requires only 120 add and 80 shift operations to perform the multiplierless 8×8 QDCT and 8×8/4×4 quantized integer transform by sharing the hardware resources. According to the computational analysis results, the proposed architecture not only reduces more than half of the add operations compared to the conventional Q+DCT and NQDCT architecture, but also performs the integer transforms additionally. On the other hand, the MPEG4 XVID video simulation results show that the proposed architecture can keep the transformation quality as good as the conventional designs. Finally, it is worth noticing that the proposed 2-D QDCIT architecture is very suitable for low-complexity and high-performance CODECs in multi-functional SoC systems.

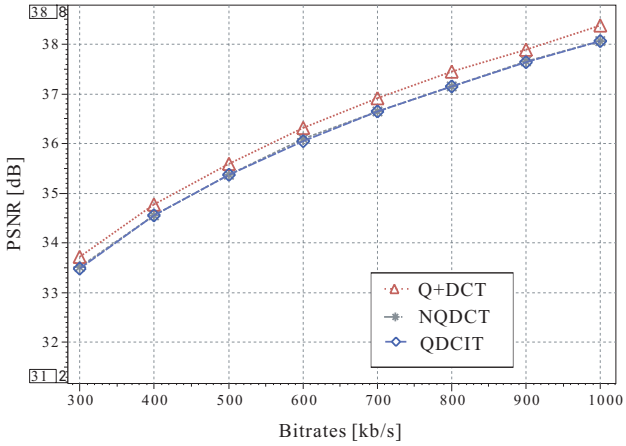


Fig. 6: The average PSNR of the "foreman-cif" from low to high bitrates in XVID.

References

[1] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. Atrium, England: John Wiley & Sons, 2003.

[2] "The XVID Website," www.xvid.org, 2007.

[3] M. K. L. K. H.S. Malvar, A. Hallapuro, "Low-Complexity Transform and Quantization in H.264/AVC," in *IEEE Transactions of circuits and systems for video technology*, vol. 13, 2003, p. 6.

[4] C.-C. Sun, S.-J. Ruan, B. Heyne, and J. Goetze, "Low-power and high-quality Cordic-based Loeffler DCT for signal processing," *IET (IEE) Circuit Devices & System*, vol. 1, pp. 453–461, Dec 2007.

[5] A. Docef, F. Kossentini, K. Nguuyen-Phi, and I. Ismaeil, "The quantized DCT and its application to DCT-based video coding," *IEEE Trans. Image Processing*, vol. 11, no. 3, pp. 177–187, 2002.

[6] W. H. Chen, C. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. Communication*, vol. 25, no. 9, pp. 1004–1009, Sept. 1977.

[7] H. Wang, M.-Y. Chan, S. Kwong, and C.-W. Kok, "Novel quantized DCT for video encoder optimization," *IEEE Signal Processing Letters*, vol. 13, no. 4, pp. 205–208, 2006.

[8] C. Loeffler, A. Lightenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11-multiplications," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Glasgow,UK, May 1989, pp. 988–991.

[9] P. Duhamel and H. H'Mida, "New 2^n DCT algorithms suitable for VLSI implementation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12, Apr. 1987, pp. 1805–1808.

[10] J. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, 1959.

[11] J. Walther, "A unified algorithm for elementary functions," in *Proc. Spring Joint Comput. Conf.*, vol. 38, 1971, pp. 379–385.

[12] T. D. Tran, "The binDCT: fast multiplierless approximation of the DCT," *IEEE Signal Processing Letters*, vol. 7, pp. 141–144, June 2000.

[13] K. K. Parhi and T. Nishitani, *Digial Signal Processing for Multimedia Systems*. New York: MARCEL DEKKER, 1999.