Microelectronics Journal 45 (2014) 1480–1488

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

CORDIC based fast algorithm for power-of-two point DCT and its efficient VLSI implementation

Hai Huang [a,*], Liyi Xiao [b]

[a] School of software, Harbin University of Science and Technology, Harbin, PR China
[b] Microelectronics Center, Harbin Institute of Technology, Harbin, PR China

ARTICLE INFO

Article history:
Received 27 April 2013
Received in revised form
24 June 2014
Accepted 8 July 2014
Available online 30 July 2014

Keywords:
Discrete cosine transform (DCT)
Coordinate rotation digital computer (CORDIC)
Very large-scale integration (VLSI) design
Linear array

ABSTRACT

—In this paper, we present a coordinate rotation digital computer (CORDIC) based fast algorithm for power-of-two point DCT, and develop its corresponding efficient VLSI implementation. The proposed algorithm has some distinguish advantages, such as regular Cooley-Tukey FFT-like data flow, identical post-scaling factor, and arithmetic-sequence rotation angles. By using the trigonometric formula, the number of the CORDIC types is reduced dramatically. This leads to an efficient method for overcoming the problem that lack synchronization among the various rotation angles CORDICs. By fully reusing the uniform processing cell (PE), for 8-point DCT, only four carry save adders (CSAs)-based PEs with two different types are required. Compared with other known architectures, the proposed 8-point DCT architecture has higher modularity, lower hardware complexity, higher throughput and better synchronization.

1. Introduction

Since the discrete cosine transform (DCT) [1] is one of the most widely used transform in signal and image processing, its various fast algorithms are reported in the literature. The existed fast algorithms can be classified into two categories: fixed-length (usually 8-point) DCT and variable-length DCT. Most of the fixed-length DCT algorithms aim at reducing computation complexity and making computation more efficiently, such as matrix factorization [2–5], deduced directly from the signal flow graphs [6–9], indirect computation takes the advantage of existing fast algorithms [10–13] and CORDIC based fast algorithm [14–19]. To reach the specific requirement, these algorithms always make extensive design optimization, which result in some specific drawbacks such as, irregular signal-flow graphs, high control complexity, and hardly extend to more than 8-point DCTs.

With the rapid progress in VLSI technology, variable-length DCT should be possible used more and more often in signal and image processing applications. Thus, it is essential to develop a fast variable-length DCT algorithm to meet the market requirement. Some researchers have investigated the fast algorithms [20–26]. Lee [20] proposed the fast radix DCT algorithm by matrix factorization with simpler structure. Hou [21] developed a recursive algorithm that can generate high-order DCT from low-order DCT. This algorithm requires extra structures for shifting and multiplexing, which damage the regularity of the structure. Kok [22] proposed a radix-2 recursive algorithm with balanced DCT structure for even length sequence. The algorithms in [20–25] are the multiplier-based algorithm, which need real or complex multipliers. Furthermore, these algorithms are not suitable for pipelined VLSI implementation due to irregular signal flow graph and the recursive nature. Hsiao et al. [26] proposed an efficient implementation of DCT computations based on the so-called shifted discrete Fourier transform (SDFT) using the unfolded pipelined CORDIC technique. This implementation has many advantages, such as low complexity, high-throughput, regularity, easy pipelining. However, it is an indirect computation for DCT. Moreover, it has the disadvantage, which most other known CORDIC based algorithms also have, of requiring various rotation angles CORDICs. This leads to the problem that lack synchronization among the various rotation angles CORDICs for the VLSI implementation.

For variable-length DCT VLSI implementation, linear array architecture should be the most suitable, because the linear array features the property of a high degree of pipelining, modularity, regularity and highly synchronized multiprocessing. The existing arrays make use of either the multiplier-based [4,27–30] or the CORDIC-based [13,14,16,26] arithmetic units. CORDIC-based architectures are suitable for VLSI implementation with regularity and simple hardware architecture. In this paper, we present a CORDIC based fast algorithm for power-of-two point DCT, and develop its

* Corresponding author.
E-mail address: ic@hrbust.edu.cn (H. Huang).

http://dx.doi.org/10.1016/j.mejo.2014.07.001
0026-2692/

corresponding efficient VLSI implementation with low hardware complexity, high throughput and good synchronization. The proposed algorithm has some distinguish advantages, such as Cooley-Tukey FFT-like regular data flow, identical post-scaling factor, and arithmetic-sequence rotation angles. To overcome the problem that lack synchronization among the various rotation angles CORDIC, we dramatically reduce the number of the CORDIC types using trigonometric formula. Furthermore, the carry save adders (CSAs)-based modified unfolded CORDIC is used to reduce the hardware complexity and speed up the computation by reducing the number of iterations performed in the conventional CORDIC. Afterwards, we develop an efficient VLSI implementation for 8-point DCT by reusing the uniform processing element (PE). Thus, a low hardware complexity (in terms of PE numbers, PE types and single PE complexity), high throughput, good synchronization and pipelined 8-point DCT is developed.

The rest of this paper is organized as follows. In Section 2, we derive the CORDIC-based fast algorithm. In Section 3, we present the signal flows of proposed DCT/IDCT algorithms and the CSA based modified unfolded CORDIC. The architecture design for fast 8-point DCT based on two types of the PE will be presented in Section 4. Finally, the comparisons and the VLSI implementation can be found in Section 5, and conclusions are drawn in Section 6.

2. CORDIC based fast algorithm for power-of-two point DCT/IDCT

For an N-point signal,$x[n]$, the DCT and IDCT are defined as:

$$\text{DCT}: C[k]=\alpha[k]\sum_{n=0}^{N-1}x[n]\cos\left[\frac{(2n+1)k\pi}{2N}\right],(k=0,....,N-1)\quad \text{and} \tag{1}$$

$$\text{IDCT}: x[n]=\sum_{k=0}^{N-1}\alpha[k]C[k]\cos\left[\frac{(2n+1)k\pi}{2N}\right],(n=0,....,N-1). \tag{2}$$

where$\alpha[0]=1/\sqrt{N}$, and $\alpha[k]=\sqrt{2/N}$for $k>0$.

We can decompose the N-point DCT into two $N/2$-point DCTs based on the CORDIC algorithm, the DCT can be written as [31]:

$$C[k]=\frac{2}{\sqrt{N}}\begin{cases}\tilde{C}_L[0], k=0\\ \tilde{C}_{\hat{H}}[0], k=N/2\\ \begin{bmatrix}\tilde{C}[k]\\ \tilde{C}[N-k]\end{bmatrix}=\begin{bmatrix}\cos\left(\frac{k\pi}{2N}\right) & \sin\left(\frac{k\pi}{2N}\right)\\ -\sin\left(\frac{k\pi}{2N}\right) & \cos\left(\frac{k\pi}{2N}\right)\end{bmatrix}\cdot\begin{bmatrix}\tilde{C}_L[k]\\ \tilde{C}_{\hat{H}}[N/2-k]\end{bmatrix}, k=1,\cdots,N/2-1\end{cases} \tag{3}$$

where

$$\text{CORDIC}\left(-\frac{k\pi}{2N}\right)=\begin{bmatrix}\cos\left(\frac{k\pi}{2N}\right) & \sin\left(\frac{k\pi}{2N}\right)\\ -\sin\left(\frac{k\pi}{2N}\right) & \cos\left(\frac{k\pi}{2N}\right)\end{bmatrix}$$

From (3), for power-of-two point DCT, the proposed algorithm computes the DCT by recursively decomposing it into 2-point DCT. The rotation angles of the CORDICs in the proposed algorithm are arithmetic sequence with a common difference of $-\pi/2N$. Another important aspect is that all outputs ($C[k]$) have uniform post-scaling factor. In addition, since the CORDIC is orthogonal and invertible matrix, the CORDIC-based DCT algorithm can be implemented by lifting-scheme according to (4). Therefore, the proposed algorithm not only can be used as regular DCT computation, but also can be used as approximation DCT computation, such as integer DCT that widely used in video compression standard.

$$\text{CORDIC}(-\theta)=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}$$

$$=\begin{bmatrix}1 & 0\\ -\tan\frac{\theta}{2} & 1\end{bmatrix}\cdot\begin{bmatrix}1 & \sin\theta\\ 0 & 1\end{bmatrix}\cdot\begin{bmatrix}1 & 0\\ -\tan\frac{\theta}{2} & 1\end{bmatrix} \tag{4}$$

Since the rotation angles of the CORDICs in the proposed algorithm are arithmetic sequence, we can use trigonometric formula to reduce the number of the CORDIC type according to Eqs. (5)–(7). In (5), we replace the $\frac{N/2-k}{2N}\pi$angle rotating with$-k\pi/2N$angle rotating followed by one butterfly operation. The general Eq. (7) can be used to realize various transforms for specific requirement. Hence, in theory any power-of-two point DCT can be implemented by only one type of CORDIC.

$$\begin{bmatrix}\boldsymbol{X}_{out}\\ \boldsymbol{Y}_{out}\end{bmatrix}=\text{CORDIC}(-\frac{N/2-k}{2N}\pi)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix}=\frac{\sqrt{2}}{2}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\cdot\text{CORDIC}\left(\frac{k\pi}{2N}\right)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix} \tag{5}$$

$$\begin{bmatrix}\boldsymbol{X}_{out}\\ \boldsymbol{Y}_{out}\end{bmatrix}=\text{CORDIC}\left(-\frac{2k}{2N}\pi\right)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix}=\text{CORDIC}\left(-\frac{k}{2N}\pi\right)\cdot\text{CORDIC}\left(-\frac{k}{2N}\pi\right)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix} \tag{6}$$

$$\begin{bmatrix}\boldsymbol{X}_{out}\\ \boldsymbol{Y}_{out}\end{bmatrix}=\text{CORDIC}(\theta+\alpha)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix}=\text{CORDIC}(\theta)\cdot\text{CORDIC}(\alpha)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix} \tag{7}$$

For 8-point DCT, Eqs. (8) and (9) can be used to transfer one type CORDIC to another type.

$$\begin{bmatrix}\boldsymbol{X}_{out}\\ \boldsymbol{Y}_{out}\end{bmatrix}=\text{CORDIC}\left(-\frac{\pi}{8}\right)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix}=\text{CORDIC}\left(-\frac{\pi}{16}\right)\cdot\text{CORDIC}\left(-\frac{\pi}{16}\right)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix} \tag{8}$$

$$\begin{bmatrix}\boldsymbol{X}_{out}\\ \boldsymbol{Y}_{out}\end{bmatrix}=\text{CORDIC}\left(-\frac{3\pi}{16}\right)\cdot\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix}=\frac{\sqrt{2}}{2}\text{CORDIC}\left(-\frac{\pi}{16}\right)\cdot\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{X}_{in}\\ \boldsymbol{Y}_{in}\end{bmatrix} \tag{9}$$

As shown in Eqs. (8) and (9), for computing 8-point DCT, only one type of CORDIC is required. Subsequently, this leads to an efficient method for overcoming the problem that lack synchronization among the various rotation angles CORDIC.

3. Signal flows for proposed fast DCT/IDCT algorithms and CSA-based modified unfolded CORDIC

In Section 2, we present the CORDIC based fast DCT algorithm. In this section, we will develop the signal flow graph of the proposed fast DCT algorithm. For a certain point DCT, the CORDIC rotation angles are fixed. Hence, the unfolded CORDIC technique is usually used to simplify the DCT computation. To reduce the hardware complexity and speed up the computation, the CSA-based modified unfolded CORDIC is used to implement the DCT.

3.1. Signal flow of proposed fast DCT/IDCT algorithm

As described in (3), the 1-D N-point fast DCT can be realized by the structure consists two basic components: butterfly operator and CORDIC array. The butterfly operator performs the conventional butterfly shuffling and addition/subtraction. The CORDIC array performs the fixed-angle rotations in DCT algorithm. Therefore, the general signal flow graph of the proposed CORDIC-based