CORDIC BASED FAST RADIX-2 DCT ALGORITHM

V.Narayana Reddy¹ · M.Sandhya Sameera²

¹Assoc.Professor, Dept.of ECE, PBRVITS-Kavali, AP, INDIA
vnr.nlr.kvl@gmail.com
²PBRVITS-Kavali, AP, INDIA
sandhyasameera.m@gmail.com

Abstract: This paper proposes an algorithm for computation of discrete cosine transformation (DCT) which uses rotational angles known as coordinate rotation digital computer (CORDIC). This proposed algorithm has less computational complexity compared to other DCT algorithms. Along with reducing computational complexities it has some discriminate advantages, such as Cooley-Tukey fast Fourier transformation (FFT)-like regular data flow, uniform post-scaling factor, in-place computation and arithmetic sequence rotation angles. Furthermore, the proposed algorithm is highly scalable, modular, regular, and suitable for pipelined VLSI implementation.

I. INTRODUCTION

A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from loss compression of audio and images to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient (as described below, fewer functions are needed to approximate a typical signal), whereas for differential equations the cosines express a particular choice of boundary conditions.

In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common. The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT", its inverse, the type-III DCT, is correspondingly often called simply “the inverse DCT” or “the IDCT”.

The DFT, like the Fourier series, implies a periodic extension of the original function. A DCT, like a cosine transform, implies an even extension of the original function. DCT, like a cosine transform, implies an even extension of the original function. Illustration of the implicit even/odd extensions of DCT input data, for N=11 data points (red dots), for the four most common types of DCT (types I-IV). However, because DCTs operate on finite, discrete sequences, two issues arise that do not apply for the continuous cosine transform.

II. COORDINATE ROTATION DIGITAL COMPUTER

CORDIC uses only Shift-and-Add arithmetic with table Look-Up to implement different functions. By making slight adjustments to the initial conditions and the LUT values, it can be used to efficiently implement Trigonometric, Hyperbolic, Exponential functions, Coordinate Transformations etc. using the same hardware. Since it uses only shift-add arithmetic, VLSI implementation of such an algorithm is easily achievable. DCT algorithm has diverse applications and is widely used for Image compression. Implementing DCT using CORDIC algorithm reduces the number of computations during processing, increases the accuracy of reconstruction of the image, and reduces the chip area of
implementation of a processor built for this purpose. This reduces the overall power consumption.

FPGA provides the hardware environment in which dedicated processors can be tested for their functionality. They perform various high-speed operations that cannot be realized by a simple microprocessor. The basic block DCT algorithm with the adaptive procedures and techniques was used. We applied a priori information for choosing proper methods of DCT coefficients quantization and coding (the most efficient for each kind of medical images). Spectral distribution of signal and noise, and HSV (human visual system) contrast sensitivity function was used to the suitable quantization technique selection. The choice of DCT coefficient coding method was based on the analysis of statistic distribution of these data. These quantization and coding methods are as follows DCT coefficients quantization threshold sample selection a single global value of quantization table elements – for scintigraphy images, - HSV contrast sensitivity function (similar to proposed in JPEG standard) - specified normalization array of quantization table elements - for ultrasound, MR and CR images, - adaptive quantization table (this table is varied as a function of DCT coefficient distribution in each block) – for high quality MR and CT images.

III. CORDIC BASED DCT

Among radix algorithms, the radix-2 algorithm is the most popular because of its computational efficiency and structural simplicity. Here in this proposed a CORDIC-based radix-2 fast DCT algorithm. Based on the proposed algorithm, signal flows of DCTs and inverse DCTs (IDCTs) are developed and deduced using their orthogonal properties, respectively. Similar to the Cooley-Tukey fast Fourier transformation (FFT) algorithm, the proposed algorithm can generate the next higher-order DCT from two identical lower-order DCTs. Furthermore, it has some distinguish advantages, such as FFT-like regular data flow, uniform post-scaling factor, in-place computation and arithmetic-sequential rotation angles. By using the unfolding CORDIC technique, this algorithm can overcome the problem of difficult to realize pipeline that in conventional CORDIC algorithms.

For an 8-point signal, \( x(n) \), the DCT is defined as:

\[
C[k] = \alpha[k] \sum_{n=0}^{N-1} x[n] \cos \left( \frac{(2n + 1)k\pi}{2N} \right), \quad k = 0, 1, 2, \ldots, N - 1
\]

Neglecting the post-scaling factor without loss of generality, the main operation of an \( \text{point} \) DCT denoted as can be written as

\[
C[k] = \sum_{n=0}^{N-1} x[n] \cos \left( \frac{(2n + 1)k\pi}{2N} \right), \quad k = 0, \ldots, N - 1
\]

\[
C[k] = 2\cos \left( \frac{\pi k}{2N} \right) \sum_{n=0}^{\left( \frac{N}{2} \right)-1} x_L[n] \cos \left( \frac{(2n + 1)k\pi}{N} \right) + 2\sin \left( \frac{\pi k}{2N} \right) \sum_{n=0}^{\left( \frac{N}{2} \right)-1} x_H[n] \sin \left( \frac{(2n + 1)k\pi}{N} \right)
\]

\[
C[N - k] = -2\sin \left( \frac{\pi k}{2N} \right) \sum_{n=0}^{\left( \frac{N}{2} \right)-1} x_L[n] \cos \left( \frac{(2n + 1)k\pi}{N} \right) + 2\cos \left( \frac{\pi k}{2N} \right) \sum_{n=0}^{\left( \frac{N}{2} \right)-1} x_H[n] \cos \left( \frac{(2n + 1)\left( \frac{N}{2} - k \right)\pi}{N} \right)
\]

IV. IMPLEMENTATION

Hardware requirement and cost of CORDIC processor is less as only shift registers, adders and look-up table (ROM) are required. Number of gates required in hardware implementation, such as an FPGA, is minimum as hardware complexity is greatly reduced compared to other processors such as DSP multipliers. It is relatively simple in design No multiplication and only addition, subtraction and bit-shifting operation ensures simple VLSI implementation Delay involved during processing is comparable to that during the implementation of a division.
or square-rooting operation. Either if there is an absence of a hardware multiplier (e.g. uC, uP) or there is a necessity to optimize the number of logic gates. The algorithm was basically developed to offer digital solutions to the problems of real-time navigation in B-58 bomber. John Walther extended the basic CORDIC theory to provide solution to and implement a diverse range of functions. This algorithm finds use in 8087 Math coprocessor, the HP-35 calculator radar signal processors, and robotics. CORDIC algorithm has also been described for the calculation of DFT DHT, Chirp Z-transforms, filtering, Singular value decomposition, and solving linear systems. Most calculators, especially the ones built by Texas Instruments and Hewlett-Packard use CORDIC algorithm for calculation of transcendental Functions. Sin θ Input angle CORDIC Processor Cos θ

![Block Diagram of a CORDIC processor](image)

**Hardware used:** FPGA

**Software used:** Verification Tool -- Modelsim 6.4c --
ModelSim SE is our UNIX, Linux, and Windows-based simulation and debug environment, combining high performance with the most powerful and intuitive GUI in the industry

*Synthesis Tool -- Xilinx ISE 9.1* -- For two-and-a-half decades, Xilinx has been at the forefront of the programmable logic revolution, with the invention and continued migration of FPGA platform technology. During that time, the role of the FPGA has evolved from a vehicle for prototyping and glue-logic to a highly flexible alternative to ASICs and ASSPs for a host of applications and markets. Today, Xilinx® FPGAs have become strategically essential to world-class system companies that are hoping to survive and compete in these times of extreme global economic instability, turning what was once the programmable revolution into the “programmable imperative” for both Xilinx and our customers.

**V EXPERIMENTAL RESULTS**

The following snapshot shows the 8-point CORDIC based DCT

![Fig: 8-point CORDIC based DCT](image)

![Fig: 8-point CORDIC based IDCT](image)
Device Utilization Summary is compared with the existing method and it shows the less complexity.

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Device Utilization Summary</th>
<th>Number of read LUTs</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Distribution</td>
<td></td>
<td></td>
<td>275</td>
<td>7,750</td>
<td>3%</td>
<td></td>
</tr>
</tbody>
</table>

**Fig:** Existing method Device utilization summary

**VI. CONCLUSION**

From the experimental results Compared to existing DCT algorithms, our proposed algorithm has several distinct advantages, such as low computational complexity, and being highly scalable, modular, regular, and able to admit efficient pipelined implementation. Furthermore, the proposed algorithm also provides an easy way to implement a reconfigurable or unified architecture for DCTs and IDCTs. This algorithm can generate the next higher order DCT from two identical lower-orders DCTs.

**REFERENCES**


