Asian Journal of Computer Science and Technology (AJCST)
Vol.2.No.1 2014 pp 24-28
available at: www.goniv.com
Paper Received :05-03-2014
Paper Published:28-03-2014
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

NOVEL ALGORITHM FOR 8 POINT DCT & IDCT IMPLEMENTATION BASED ON CORDIC

S.Ramesh,R.Kangeyan,
Department of ECE,
Pallavan college of engineering,
Kanchipuram.

ABSTRACT

A novel coordinate rotation digital computer (CORDIC)-based fast radix-2 algorithm for computation of discrete cosine transformation (DCT) . The proposed algorithm has some distinguish 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. Compared to existing DCT algorithms, this proposed algorithm has lower computational complexity. Furthermore, the proposed algorithm is highly scalable, modular, regular, and suitable for pipelined VLSI implementation.

Keyword-Discrete Cosine Transform (DCT), Compression technique, Inverse Discrete Cosine ransform(IDCT),

1. 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".Two related transforms are the discrete sine transforms (DST), which is equivalent to a DFT of real and odd functions, and the modified discrete cosine transforms (MDCT), which is based on a DCT of overlapping data. Like any Fourier-related transform, discrete cosine transforms (DCTs) express a function or a signal in terms of a sum of sinusoids with different frequencies and amplitudes. Like the discrete Fourier transforms (DFT), a DCT operates on a function at a finite number of discrete data points. The obvious distinction between a DCT and a DFT is that the former uses only cosine functions, while the latter uses both cosines and sines (in the form of complex exponentials). However, this visible difference is merely a consequence of a deeper distinction: a DCT implies different boundary conditions than the DFT or other related transforms. The Fourier-related transforms that operate on a

function over a finite domain, such as the DFT or DCT or a Fourier series, can be thought of as implicitly defining an extension of that function outside the domain. That is, once you write a function as a sum of sinusoids, you can evaluate that sum at any , even for where the original was not specified. 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.First, one has to specify whether the function is even or odd at both the left and right boundaries of the domain (i.e. the min-n and max-n boundaries in the definitions below, respectively).

2. THEORY OF CORDIC ALGORITHM

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 sing their orthogonal properties, respectively. Similar to the Cooley-Turkey fast Fourier transformation (FFT) algorithm, the proposed algorithm can generate the next higher- rider 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-sequence rotation angles. By using the unfolding CORDIC technique, this algorithm can overcome the problem of difficult to realize pipeline that in conventional CORDIC algorithms. This results in a pipeline and high-speed

VLSI implementation. Compared to existing DCTs, the proposed algorithm has low computational complexity, and is highly scalable, modular, regular, and able to admit efficient pipelined implementation. In addition, this letter also provides an easy way to implement the reconfigurable or unified architecture for DCTs and IDCTs using the orthogonal property. The general signal-flow graph for the proposed fast DCT algorithm given, while the signal-flow graphs of 2-point DCT, 4-point DCT, and 8-point DCT are respectively represented. Where the angles in the circles are used to represent CORDICs with this rotation angles. There are two separate -point DCTs and one CORDIC array. As mentioned above, the CORDIC array has CORDICs with arithmetic-sequence rotation angles. The inputs are addressed in bit-reverse order and the outputs are addressed in natural order. It also supports in-place computation like the FFT.

For special applications, a double-angle formula can be used to reduce CORDIC types. Hence, the architecture based on the signal flow is highly modular. Furthermore, the modified unfolded CORDIC, which presented in our previous work, can be used to speed up computations and overcome recursive problems in conventional CORDICs. Similarly, the fast algorithm for the -point IDCT can be deduced like the fast DCT algorithm. Alternatively, it can be obtained more easily using their orthogonal property. As is known, the DCT and IDCT are orthogonal transformations, and the signal flow of the -point IDCT can be easily obtained by inverting the transfer function of each building block and reversing the signal flow direction.

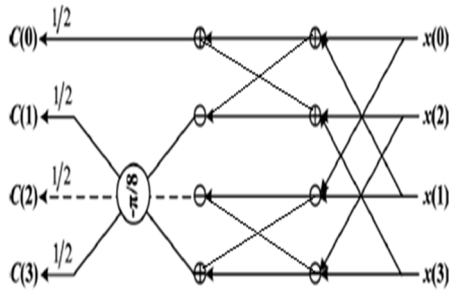


Figure.1.Signal flow of a 4-point fast discrete cosine transformation (DCT)

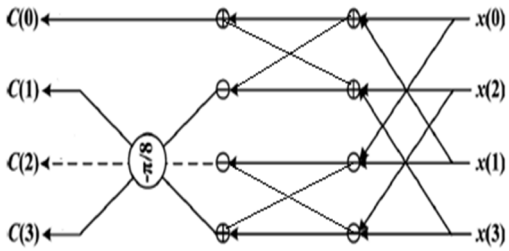


Figure.2.Signal flow of an 4-point fast discrete cosine transformation Without scaling factor

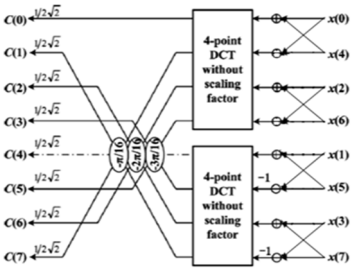


Figure.3.Signal flow of an 8-point fast discrete cosine transformation (DCT)

3. SYSTEM EXPLANATION

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

$$[\ ] = [\ ] \quad [\ ] \quad \frac{(\quad + \quad)}{} \ ,$$

$$(\quad = \ , \ , \ , \ldots , \quad - \ )$$

Neglecting the post-scaling factor without loss of generality, the main operation of an -point DCT denoted as can be written as

$$[\ ] = \quad [\ ] \quad \frac{(\quad + \quad)}{} \ ,$$

$$= \ , \ldots , \quad -$$

$$[\ ]$$

$$= \quad \text{—} \quad - \quad [\ ] \quad \frac{(\quad + \quad)}{}$$

$$+ \quad \text{—} \quad - \quad [\ ] \quad \frac{(\quad + \quad)}{}$$

$$[\ ] = \quad \text{—} \ \Sigma^{-} \quad [\ ] \quad \frac{(\quad)}{} +$$

$$\text{—} \ \Sigma^{-} \quad (-\ ) \quad [\ ] \quad \frac{(\quad) -}{}$$

$$[\ - \ ] =$$

$$- \quad \text{—} \ \Sigma^{-} \quad [\ ] \quad \frac{(\quad)}{} +$$

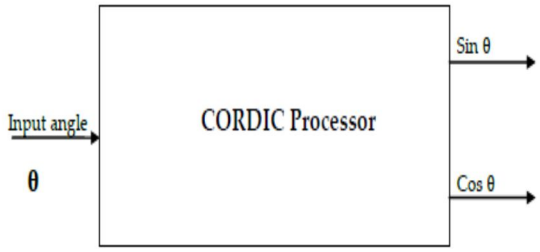
$$\text{—} \ \Sigma^{-} \quad (-\ ) \quad [\ ] \quad \frac{(\quad) -}{}$$

CORDIC or Coordinate Rotation Digital Computer is a simple and hardware-efficient algorithm for the implementation of various elementary, especially trigonometric, functions. Instead of using Calculus based methods such as polynomial or rational functional approximation, it uses simple shift, add, subtract and table look-up operations to achieve this objective. The CORDIC algorithm was first proposed by Jack E Volder in 1959. It is usually implemented in either Rotation mode or Vectoring mode. In either mode, the algorithm is rotation of an angle vector by a definite angle but in variable directions. This fixed rotation in variable direction is implemented through an iterative sequence of addition/subtraction followed by bit-shift operation.

The final result is obtained by appropriately scaling the result obtained after successive iterations. Owing to its simplicity the CORDIC algorithm can be easily implemented on a VLSI system. 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 on 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



Figue. 4Block Diagram of a CORDIC processor

The general signal-flow graph for the proposed fast DCT algorithm given, while the signal-flow graphs of 2-point DCT, 4-point DCT, and 8-point DCT are respectively represented. Where the angles in the circles are used to represent CORDICs with this rotation angles.

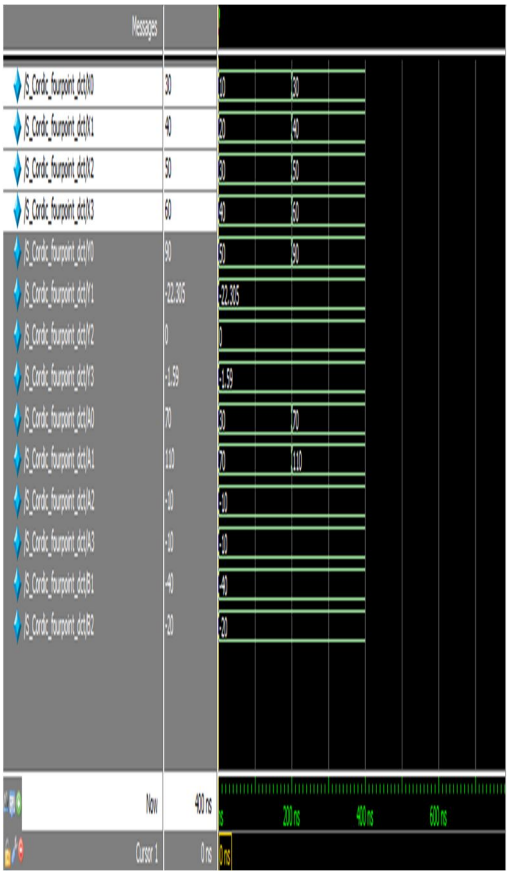
There are two separate -point DCTs and one CORDIC array. As mentioned above, the CORDIC array has CORDICs with arithmetic-sequence rotation angles. The inputs are addressed in bit-reverse Order and the outputs are addressed in natural order. It also supports in-place computation like the FFT. Regular

and pure feed-forward data paths of the signal flow make them suitable for Pipelined VLSI implementation. For special applications, a double-angle formula can be used to reduce CORDIC types. Hence, the architecture based on the signal flow is highly modular.

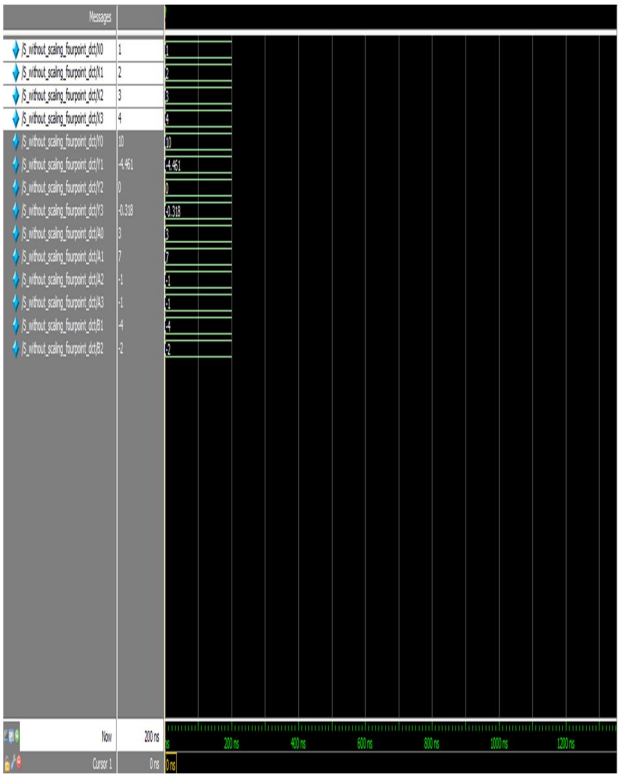
The DCT and IDCT are orthogonal transformations and the signal flow of the -point IDCT can be easily obtained by inverting the transfer function of each building block and reversing the signal flow direction. CORDIC algorithm has also been described for the calculation of DFT, DHT Chirp Z-transforms 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.

4. SIMULATION WAVEFORM

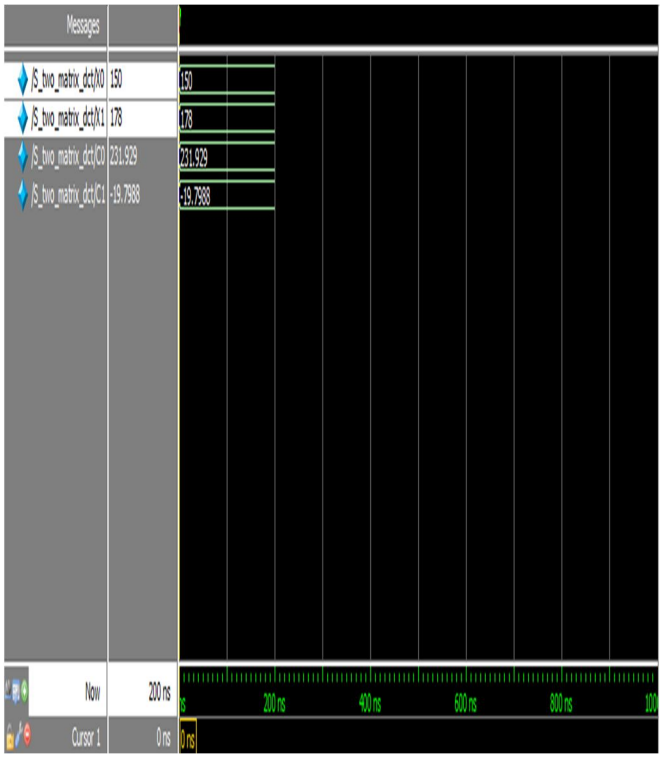
TWO POINT DCT:



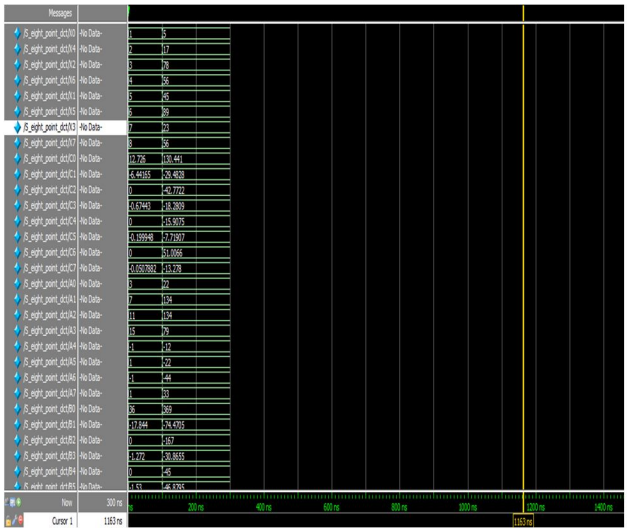
FOUR POINTS WITHOUT SCALING FACTOR:



FOUR POINT DCT:



EIGHT POINT DCT:



5. CONCLUSION

We propose a novel CORDIC-based radix-2 fast DCT algorithm. This algorithm can generate the next higher order DCT from two identical lower-orders DCTs. Compared to existing DCT algorithms
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. The general signal-flow graph for the fast DCT algorithm given, while the signal-flow graphs of 2-point DCT, 4-point DCT, and 8-point DCT are respectively represented. Where the angles in the circles are used to represent CORDICs with this rotation angles. There are two separate -point DCTs and one CORDIC array. As mentioned above, the CORDIC array has CORDICs with arithmetic-sequence rotation angles.

6. FUTURE ENHANCEMENT

Implementation of CORDIC based Inverse Discrete Cosine Transform will be done.

REFERENCES

[1]. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput.,vol. C-23, pp. 90–94, 1974.

[2]. T. D. Tran, "The binDCT: Fast multiplierless approximation of the DCT," IEEE Signal Process. Lett., vol. 7, no. 6, pp. 141–144, 2000.

[3]. W. H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," IEEE Trans. Commun., vol. 25, no. COM-9, pp. 1004–1009, Sep. 1977.

[4]. E. Feig and S.Winograd, "Fast algorithms for the discrete cosine transform,"IEEE Trans. Signal Process., vol. 40, no. 9, pp. 2174–2193, Sep.1992.

[5]. V. Britanak and K. R. Rao, "Two-dimensional DCT/DST universal computational structure for block sizes 2×2 ," IEEE Trans. Signal Process., vol. 45, no. 11, pp. 3250–3255, Nov.2000.

[6]. C. Loeffler, A. Ligtenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in Proc. Int. Conf. Acoust.,Speech, Signal Process., 1989, pp. 988–991.

[7]. J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," IEEE Trans. Signal Process., vol. 49,no. 12, pp. 3032–3044, 2001.

[8]. Y. P. Lee, T. H. Chen, L. G. Chen, M. J. Chen, and C. W. Ku, "A cost effective architecture for 8 8 two-dimensional DCT/IDCT using direct method," IEEE Trans. Circuits Syst. Video Technol., vol. 7, pp.459–467, 1997.

[9]. S.Yu and E. E. Swartzlander, Jr, "DCT implementation with distributed arithmetic," IEEE Trans. Comput., vol. 50, no. 9, pp. 985–991, Sep.2001.

[10]. L. Xiao and H. Huang, "A novel CORDIC based unified architecture for DCT and IDCT," in 2012 Int. Conf. Optoelectronics and Microelectronics (ICOM), 2012, Aug. 2012, pp. 496–500.