Abstract

**Objective:** The objective of this work is to efficiently implement a Co-ordinate Rotation Digital Computer (CORDIC) based fast algorithm for power of two length DCT. **Methods:** The proposed algorithm has some advantages such as regular Cooley-Tukey FFT like data flow, post scaling factor and arithmetic-sequence rotation angles. Using CORDIC algorithm, different types of DCT like four point, eight point and inverse eight point are derived. Usage of trigonometric formula reduces the number of CORDIC. By reusing uniform processing Element cells, architecture for 8-point CORDIC DCT is developed. This is an efficient method for overcoming the problem of lack of synchronization among various rotation angles. **Findings:** When compared with other known architectures, the proposed 8-point DCT architecture is more efficient in terms of power and area. Power analysis and area estimation are performed for the designed architectures and then further power reduction is carried out using clock gating technique. A power reduction of 4.29% is achieved. The area is also reduced by 0.29%. **Conclusion:** An efficient VLSI implementation for 8-point DCT by reusing the uniform processing element cells with low hardware complexity is developed.

Keywords: CORDIC, Discrete Cosine Transform (DCT), Processing Element (PE)

1. Introduction

Discrete Cosine Transform (DCT) is mostly used in signal and image processing. There are several algorithms available for DCT implementation. The existing fast algorithms are classified into two categories. They are fixed-length DCT and variable-length DCT. Fixed-length algorithms are less complex and more efficient, such as matrix factorization, deduced directly from the signal flow graphs and CORDIC-Based fast algorithm. However, the DCT designed using this algorithm has some drawbacks such as irregular signal flow graphs and increased complexity.

With more advancement in VLSI technology, variable-length DCTs are more frequently used in image and signal processing applications. B.G. Lee developed a technique to calculate the Discrete Cosine Transform. This algorithm reduces the number of multipliers. S. Hou developed a recursive algorithm that can generate high-order DCT from low-order DCT which requires few multipliers and adders. C. W. Kok developed fast algorithm for calculating discrete cosine transform, based on the direct decomposition and recursive property of the DCT for even length input sequence generalized from radix-2 algorithm. The algorithms follow composite sequence lengths, directive recursive structures of the DCT/IDCT. Fast radix-q and mixed radix algorithms DCT require real and complex multiplications. These algorithms are not suitable for pipelined VLSI implementation due to recursive property and irregular signal flow graphs. S.F. Hsiao developed an efficient VLSI implementation of fast multiplierless approximated DCT. The efficient implementation of DCT computations is known as Shifted Discrete Cosine Transform (SDCT) which can be implemented using the unfolded pipelined CORDIC technique. This technique has many advantages such as high-throughput, easy pipelining and regularity. The indirect decomposition of DCT is a disadvantage, which other CORDIC based algorithm also possess. This causes lack of synchronization among various rotation angles in VLSI implementation.
Variable length DCT algorithm is mostly suitable for linear architectures because linear architecture possesses high degree of pipelining, high synchronization and modularity in multiprocessing. The existing arrays are either the multiplier based or CORDIC based arithmetic functions. The proposed algorithm has some advantages such as arithmetic-sequence rotation angles, Cooley-Tukey FFT like data flow and post scaling factor. To overcome the lack of synchronization among various rotation angles in CORDIC, the number of the CORDIC types are decreased using trigonometric formulas. Further, the carry save adders based on CORDIC is used to decrease the speed of computation by reducing the number of iterations. An efficient VLSI implementation for 8-point DCT by reusing the uniform Processing Element (PE) cells is developed. The uniform processing elements have low hardware complexity. The ASIC implementation of the design is performed and the area, power and performance are calculated for the designed structure. For power optimization, the clock-gating technique is used.

2. **CORDIC based Fast Algorithm for Power of Two Point DCT/IDCT**

For a P-point signal, x[p] the DCT and IDCT are defined as follows. The P-point DCT is

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

The IDCT of P-point is as follows.

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

Where

\[
 \alpha[k] = \begin{cases} 
 \frac{1}{\sqrt{P}} & \text{for } k = 0 \\
 \frac{2}{\sqrt{P}} & \text{for } k > 0 
\end{cases}
\]

We can divide the N-point DCT into two N/2-point DCTs based on the CORDIC algorithm. Then the DCT can be written as

\[
 C_D[0] \text{ for } p = 0
 C_D[N/2] \text{ for } p = N/2
\]

\[
 \begin{bmatrix} 
 C_1[p] \\
 C_1[N-p] 
\end{bmatrix} = \begin{bmatrix} 
 \cos \left( \frac{p\pi}{2N} \right) & \sin \left( \frac{p\pi}{2N} \right) \\
 -\sin \left( \frac{p\pi}{2N} \right) & \cos \left( \frac{p\pi}{2N} \right) 
\end{bmatrix} \begin{bmatrix} 
 C_1[k] \\
 C_1[N/2-1-p] 
\end{bmatrix} \quad p = 1, \ldots, N/2 - 1
\]

Where

\[
 CORDIC(-\frac{P\pi}{2N}) = \begin{bmatrix} 
 \cos \left( \frac{P\pi}{2N} \right) & \sin \left( \frac{P\pi}{2N} \right) \\
 -\sin \left( \frac{P\pi}{2N} \right) & \cos \left( \frac{P\pi}{2N} \right) 
\end{bmatrix}
\]

For power of two point DCT, the proposed algorithm computes the DCT dividing it into 2-point DCT. The rotation angles of the CORDICs in the proposed algorithm are sequenced with a common difference of \(-\frac{\pi}{2N}\). All the outputs (z[p]) have uniform scaling factor. The CORDIC consists of invertible and orthogonal matrix. The integer DCT used in standard video compression is as follows.

\[
 CORDIC(-\delta) = \begin{bmatrix} 
 1 & 0 \\
 -\tan \left( \frac{\delta}{2} \right) & 1 
\end{bmatrix}
\]

Since the rotation angle of the CORDICs in the proposed algorithm are in arithmetic sequence, the trigonometric formula is used to decrease the number of CORDIC types. Replace the \(\frac{N/2-k}{2N}\) angle rotating with \(-\frac{k\pi}{2N}\) angle rotating by butterfly operation. The power of two point DCT can be implemented by one type of CORDIC.

\[
 \begin{bmatrix} 
 M_{out} \\
 N_{out} 
\end{bmatrix} = CORDIC \left( -\frac{N/2-k}{2N} \pi \right) \begin{bmatrix} 
 M_{in} \\
 N_{in} 
\end{bmatrix}
\]

\[
 \begin{bmatrix} 
 M_{out} \\
 N_{out} \end{bmatrix} = \begin{bmatrix} 
 \frac{1}{\sqrt{2}} & 1 \\
 -1 & 1 
\end{bmatrix} CORDIC \left( \frac{k\pi}{2N} \right) \begin{bmatrix} 
 M_{in} \\
 N_{in} \end{bmatrix}
\]

\[
 \begin{bmatrix} 
 M_{out} \\
 N_{out} 
\end{bmatrix} = CORDIC \left( -\frac{2k}{2N} \pi \right) \begin{bmatrix} 
 M_{in} \\
 N_{in} \end{bmatrix}
\]
Eight-point DCT can be used to transfer one type CORDIC to another type.

\[
\begin{bmatrix}
M_{\text{out}} \\
N_{\text{out}}
\end{bmatrix} = \text{CORDIC}(-\frac{k}{2N}\pi) \cdot \text{CORDIC}(-\frac{k}{2N}\pi) \cdot \begin{bmatrix}
M_{\text{in}} \\
N_{\text{in}}
\end{bmatrix}
\]

\[
\begin{bmatrix}
M_{\text{out}} \\
N_{\text{out}}
\end{bmatrix} = \text{CORDIC}(\theta + a) \cdot \begin{bmatrix}
M_{\text{in}} \\
N_{\text{in}}
\end{bmatrix}
\]

\[
\begin{bmatrix}
M_{\text{out}} \\
N_{\text{out}}
\end{bmatrix} = \text{CORDIC}(\theta) \cdot \text{CORDIC}(a) \cdot \begin{bmatrix}
M_{\text{in}} \\
N_{\text{in}}
\end{bmatrix}
\]

For computing 8-point DCT, one type of CORDIC is required. This leads to an efficient method for overcoming the problem of lack of synchronization among the various rotation angles.

3. Signal Flow Graphs for DCT/IDCT Algorithms

In this section, the signal flow graph of the proposed DCT/IDCT algorithm is presented. For a certain point DCT, the CORDIC angles are fixed. The CORDIC technique is used for DCT computation. The 1-D N-point DCT can be realized by the structure that consists of two parts. They are basically Butterfly operators and CORDIC arrays. The butterfly operator perform the butterfly shuffling subtract and addition. The CORDIC array performs the fixed-angle rotations in DCT algorithm. The general flow graphs of the DCT is shown in Figure 1.

The signal flow graphs can be separated into two major parts, they are butterfly operator and cordin array

with scaling factor. The post scaling factor of 4-point and 8-point CORDIC DCT are:

\[\beta_4 = \frac{1}{2} \text{ and } \beta_8 = \frac{1}{2}\] respectively. The proposed algorithm can be used for both scaled DCT and regular DCT. The 4-point, 8-point and Inverse 8-point CORDIC DCT are shown in Figure 2, Figure 3 and Figure 4 respectively.

The DCT and IDCT are orthogonal to each other, and the signal flow functions of the DCT and IDCT are shown in Table 1. In the table the butterfly operations in

![Figure 1. General flow graph of the CORDIC-DCT.](image1)

![Figure 2. Signal flow graph of four-point CORDIC-DCT.](image2)

![Figure 3. Signal flow graph of eight point CORDIC DCT.](image3)

![Figure 4. Signal flow graph of inverse eight point CORDIC-DCT.](image4)
the DCT/IDCT having same rotation angles but opposite
directions is shown.

The input vector \([M_{in}, N_{in}]\) has been fed in
anticlockwise direction to generate the output vector
\([M_{out}, N_{out}]\) and are expressed as

\[
\begin{bmatrix}
M_{in} \\
N_{in}
\end{bmatrix} = \begin{bmatrix}
\cos(\gamma) & -\sin(\gamma) \\
\sin(\gamma) & \cos(\gamma)
\end{bmatrix} \begin{bmatrix}
M_{out} \\
N_{out}
\end{bmatrix}
\]

The equation can be rewritten as

\[
\begin{bmatrix}
M_{in} \\
N_{in}
\end{bmatrix} = \begin{bmatrix}
\cos(\gamma) & \sin(\gamma) \\
-\sin(\gamma) & \cos(\gamma)
\end{bmatrix} \begin{bmatrix}
M_{out} \\
N_{out}
\end{bmatrix}
\]

Computation is done for clockwise direction \(\theta\), which
is equivalent to anticlockwise direction. The end result
will be achieved when the input and output vectors are
changed. Thus CORDIC implementation is achieved for
DCT/IDCT.

4. Architecture Design of the Proposed CORDIC DCT

The architecture design of the 8-point CORDIC DCT is
shown in Figure 5. In the architecture, there are mainly
two different types of Processing Elements (PE). They
are PE_1 and PE_2. The PE_1 is used to realize the but-
terfly operators and three PE_2s are used to realize the
CORDIC array. The 2-to-1 Mux and the register select the
data to be followed by PE_2. The 4-to-1 muxes are used to
select the right outputs.

The PE_1 has eight inputs and two outputs. The two
outputs can act as inputs of PE_2. The one output will act
as input of mux, other input has two registers and out-
put is controlled by control line (cr). The other output of
PE_2 can be given as input of mux, other input with one
register and output is controlled by control signal. The
PE_1 outputs, PE_2 outputs can be controlled by control
signals (cr1, cr2) and the output is selected. The outputs
of 2 to 1 mux acts as inputs of PE_2 and two outputs.
Now the outputs act as inputs for butterfly structure and

4.1 Architecture of the PE_1 CORDIC DCT

The PE_1 has 8 inputs and two outputs. The PE_1 has two
sub pipeline stages. Each stage has one output. The archi-
tecture of the carry save adders (4,2) are as follows: each
carry save adder has two cascaded CSA (3,2) and two
Carry Look Ahead adders (CLAs). The 2 to 1 mux takes
input and the complemented input and the selection line
selects the data in the mux for PE_1. The architecture of
the PE_1 8-point CORDIC DCT is shown in Figure 7.
4.2 Architecture of the PE_2 CORDIC DCT

The PE_2 has two inputs and two outputs. The PE_2 has three sub pipeline stages. Each stage has two inputs and two outputs. Each pipeline stage has different types of wired-shifters, two carry look ahead adders and registers. The Architecture of the PE_2 8-point CORDIC DCT is shown in Figure 8.

5. Simulation Results and Comparisons

The proposed signal flow graph of the 8-point CORDIC DCT simulation result is shown in Figure 9 and the corresponding layout view is shown in Figure 10. The proposed architecture of the eight point CORDIC DCT simulation result is shown in Figure 11. The corresponding layout view is shown in Figure 12.

The graphs of the different CORDIC DCT and proposed eight point CORDIC DCT is shown in Figure 13. From Figure 13, it is evident that a 4-point DCT consumes 263504.41μW power. 8-point CORDIC DCT consumes 140560.99μW. Inverse eight point CORDIC DCT consumes 134423.26μW and proposed eight point CORDIC DCT consumes 591036.2μW. From Figure 13, it is evident that a 4-point DCT occupies 167508 μm² area. 8-point CORDIC DCT occupies 820413 μm². Inverse eight point CORDIC DCT occupies 818800 μm² and proposed eight point CORDIC DCT occupies 69252 μm².

The graphs of the different CORDIC DCT and proposed eight point CORDIC DCT with clock gating is shown in Figure 14. From Figure 14, it is evident that a 4-point DCT consumes 552335.41μW power. 8-point CORDIC DCT consumes 328015.99μW. Inverse eight point CORDIC
DCT consumes 265405.26μW and proposed eight point CORDIC DCT consumes 278398.2μW. From Figure 14, it is evident that a 4-point DCT occupies 167508 μm² area. 8-point CORDIC DCT occupies 820413 μm². Inverse eight point CORDIC DCT occupies 818800 μm² and proposed eight point CORDIC DCT occupies 71640 μm².

The comparison graph between eight point CORDIC DCT and proposed eight point CORDIC DCT is shown in Figure 15. Comparison graph between eight point CORDIC DCT and proposed eight point CORDIC DCT with clock gating is shown in Figure 16. From Figure 15
it is inferred that there is 95.79% reduction of power and 91.55% reduction of area for proposed eight point CORDIC DCT. From Figure 16 it is inferred that there is 91.52% reduction of power and 91.26% reduction of area for proposed eight point CORDIC DCT.

6. Conclusion

A hardware implementation of CORDIC power of two length DCT/IDCT is presented. CORDIC based fast algorithm for 4-point, 8-point DCT/IDCT are developed. An efficient VLSI implementation for 8-point DCT by using two uniform Processing elements was also developed. CORDIC 8-point DCT is compared with 8-point DCT on various factors such as area, power and gates. By the use of clock gating 4.29% power reduction is achieved and 0.29% reduction in area is obtained.

7. References

1. Ahmed N, Natarajan T, Rao KR. Discrete cosine transform. IEEE Transactions on Computers. 1974; 23(1):90–4.
2. Tugsinavisut S, Hong Y, Kim D, Kim K, Beerl PA. Efficient asynchronous bundled-data pipelines for DCT matrix-vector multiplication. IEEE Transactions on Very Large Scale Integration System. 2005; 13(4):448–60.
3. Guo JI, Li CC. A generalized architecture for the one dimensional discrete cosine and sine transforms. IEEE Transactions on Circuits and Systems for Video Technology. 2001; 11(7):874–81.
4. Loeffler C, Ligtenberg A, Moschytz G. Practical fast 1DDCT algorithms with 11 multiplications. Proceedings of the International Conference Acoustics, Speech, Signal Processing; Glasgow. 1989. p. 988–91.
5. Kaddachi ML, Soudani A, Lecuire V, Makkaoui L, Moureaux JM, Torki K. Design and performance analysis of a zonal DCT-based image encoder for wireless camera sensor networks. Microelectronics Journal. 2012; 43(11):809–17.
6. Hu YH, Wu Z. An efficient CORDIC array structure for the implementation of discrete cosine transform. IEEE Transactions on Signal Processing. 1995; 43(1):331–6.
7. Chen J, RayLiu KJ. A complete pipelined parallel CORDIC architecture for motion estimation. IEEE Transactions on Circuits Systems II: Analog and Digital Signal Processing. 1998; 45(6):653–60.
8. Yu S, Swartzlander EE. A scaled DCT architecture with the CORDIC algorithm. IEEE Transactions on Signal Processing. 2002; 50(1):160–7.
9. Lee BG. A new algorithm to compute the discrete cosine transform. IEEE Transactions on Acoustics Speech and Signal Processing. 1984; 32(6):1243–5.
10. Hou HS. A fast recursive algorithm for computing the discrete cosine transform. IEEE Transactions on Acoustics Speech and Signal Processing. 1987; 35(10):1445–61.
11. Kok CW. Fast algorithm for computing discrete cosine transform. IEEE Transactions on Signal Processing. 1997; 45(3):757–60.
12. Bi G, Yu LW. DCT algorithms for composite sequence lengths. IEEE Transactions on Signal Processing. 1998; 46(3):554–62.
13. Chen CH, Liu BD, Yang JF. Direct recursive structures for computing radix-r two dimensional DST/IDST. IEEE Transactions on Circuits Systems I: Regular Paper. 2004; 51(10):2017–30.
14. Hsu H, Liu C. Fast radix-q and mixed radix algorithms for type-I DCT. IEEE Signal Processing Letters. 2008; 15:910–3.
15. Hsiao SF, Hu YH, Juang TB, Lee CH. Efficient VLSI implementations of fast multiplierless approximated DCT using parameterized hardware modules for silicon intellectual property design. IEEE Transactions on Circuits System-I: Regular Paper. 2005; 52(8):1568–79.