# A Low Power and Fast DCT/IDCT Architecture Using Proposed Loeffler-CSD-CSE Algorithm

Syafiq Al-'Atiiq, Gilang Ilham, and Ardianto Satriawan
Department of Electrical Engineering,
School of Electrical Engineering and Informatics,
Institute Technology of Bandung, Bandung, Indonesia
[syafiq, gilangilham, ardianto_s]@students.itb.ac.id

***Abstract*— Discrete Cosine Transform (DCT) and Inverse Discrete Cosine Transform (IDCT) is widely applied in various international coding standards. In this paper, we present a new design of low power, high speed and low complexity algorithm based in radix-2 DCT algorithm. The approach is based in modification of DCT bases in a bit-wise manner using Canonical Signed Digit (CSD) representation. Thus, by using CSD representation we can reduce amount of computation, with reducing the non zero digit, therefore the number of adders, subtracters, and registers will be minimum. Futhermore, we use Common Subexpression Elimination (CSE) and pipeline technique to reduce computational complexity and increase computational speed. With our design, we can tide the DCT operation at 81.95 MHz of clock frequency. At the same time, our algorithm also achieved low power consumption that only consume 8.42 mW at top design impementation. The pipeline latency of proposed architecture is 44 Clock cycles also at top design implementation.**

## I. INTRODUCTION

Transform coding constitutes an integral component of contemporary image/video processing applications. In the last decade, Discrete Cosine Transform (DCT) has emerged as the de-facto image transformation in most visual systems. DCT has been widely deployed by modern video coding standards, for example, MPEG, JVT, etc [1].

The JPEG image compression standard [2] was developed by Joint Photographic Expert Groups. This standard supports a wide variety for continuous-tone image computation. There are two basic compression way in JPEG standards. The first one called DCT is specified for "lossy" compression and the other is predictive method for "lossless" compression.

In other way, fast and efficient computation for DCT algorithms [3] are important due to reducing the computation complexity, especially in mobile device where the limited computing power or battery supply playing an important roles. This low power DCT algorithm has been researched by many researcher around the world. In [4] the paper present reconfigurable DCT algorithm to minimize power consumption. But, with the minimum computation of DCT, there is always peep out another trade off of PSNR degradation. But, the idea of reducing the complexity by grouping the coefficient is good enough to reduce the computation. Later, we will use this part of method to be implemented in our algorithm.

In VLSI implementation, the multiplication usage is not efficient, especially in DCT impelementation. Many DCT architecture have been proposed to minimize the multiplication usage to improve the trade off between computation time and hardware requirement. Among these, the DCT algorithm proposed by Loeffler [5], has opened a new area in image processing by minimize the number of required multiplication down to 11. This is reached the theoretical limit for an-8 point DCT [6]. In spite of reaching the theoretical limit, these 11 multipliers still burdening the FPGA for our implementation. In all conscience, these operators need many switching events which increase the critical path and transition activity. Hence, using multiplications need more power and decrease the speed.

Because of this, one way to obtain the constant in form of less power and high speed computation, we use Canonical Signed Digit (CSD) representation form [7] and [8]. This representation reduce the non-zero bit so as reduce the number of adder, substracter, and register. Moreover there are some common elements which we can share together to minimize the usage of wire and register. This method is called Common Subexpression Elimination (CSE) taken from [9].

We propose a low-power, high speed and small area 2D-DCT architechture, which is based on efficient trade off between image quality and computational complexity. The low power approach is based on modification of multiplier and adder block in loeffler algorithm with shift and add in CSD form and considerable area reduction can be accomplished using the proposed scheme. Computation of some identical subexpression just implemented once so that the resources necessary of these operations can be shared.

This research has done in [10], but with the approach of grouping the coefficient to reduce the computation, we can reach more speed, decrease more area, and low power consumption. This also our novelty of this paper.

## II. DCT ALGORITHM

### A. DCT Background

The one-dimensional (1-D) DCT transforms one-dimension N points block samples from spatial domain $f(x, y)$ into frequency domain $Z(u)$ as follows :

$$Z(u) = \sum_{x=0}^{N-1} \left( f(x) \left[ \frac{\pi(2x+1)u}{2N} \right] \right) \quad (1)$$

for $u = 0, 1, 2, 3, ..$ where $\alpha(u)$ is defined as :

$$a(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{for } u = 0 \\ \sqrt{\frac{2}{n}} & \text{for } u \neq 0 \end{cases}$$

Similarly, the inverse transformation is defined as :

$$f(x) = \sum_{u=0}^{N-1} a(u) \left( Z(u) \cos \left[ \frac{\pi(2x+1)u}{2N} \right] \right) \quad (2)$$

for $x = 0, 1, 2, ..., N.\alpha(u)$ is defined as :

$$a(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{for } u = 0 \\ \sqrt{\frac{2}{n}} & \text{for } u \neq 0 \end{cases}$$

And, the two dimensional DCT transforms an 8x8 block sample from spatial domain $f(x, y)$ into frequency domain $Z(u, v)$ as follows :

$$Z(u,v) = a(u)a(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left( f_{x,y} \cos \left[ \frac{\pi(2x+1)u}{2N} \right] \cos \left[ \frac{\pi(2y+1)v}{2N} \right] \right) \quad (3)$$

and the inverse transformation is defined as :

$$f(x,y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} a(u)a(v) \left( C_{u,v} \cos \left[ \frac{\pi(2x+1)u}{2N} \right] \cos \left[ \frac{\pi(2y+1)v}{2N} \right] \right) \quad (4)$$

A commonly used approach to reduce the computational complexity is row-column decomposition. This decomposition performs row wise one dimensional (1-D) DCT followed by a column-wise 1-D transform from equation (1). So that, the comoputation for Inverse DCT by using row-column decomposition commonly just replacing equation (1) to equation (2) and equation (3) to equation (4).

### *B. Loeffler DCT*

Since the number of multiplication of Loeffler algorithm [5] reaches the teoretical limit of 11 multiplication of DCT written in [6], our proposed algorithm is based in this algorithm.

As shown in figure 4, loeffler algorithm perform four stages to compute DCT outputs. Each stage consist of some arithmetic operations. Stage-1 only consist of adders and substracters, but in stage-2, there is some multiplication and adder box. We call this "black box" as $a, b, c,$ and $d$, which the value each of them is :

$$\text{MAS1} = \cos(\frac{3\pi}{16})$$

$$\text{MAS2} = \cos(\frac{\pi}{16})$$

$$\text{MAS3} = \sqrt{2}\cos(\frac{6\pi}{16})$$

$$\text{M} = \sqrt{2}$$

Since the "black box" consist of multiplier and adder, it doesn't efficient in power concumption and consume a large area. So that, we will replace every "black box" above with recent technique to decrease power consumption and area.

### *C. Canonical Signed Digit (CSD)*

Canocnical Signed Digit (CSD) is a number system based on radix-2 for encoding a floating point value in a two's complement representation [7]. It can reduce the complexity of the hardware required to realize a DCT and IDCT. This representation containing fewer non-zero digit. Therefore, for constant multipliers, the number of additions and subtractors will be minimum.

## III. PROPOSED ALGORITHM

### *A. One-Dimensional Discrete Cosine Transform*

The one-dimensional discrete cosine transform in our algorithm is based on loeffler algorithm in Fig. (4). Every single black box is replaced with constant values of CSD representation as shown in Fig. (2). Then, to enhance the use of adder and shift operators we use Common Subexpression Elimination (CSE). This method was introduced in [9] and applied in many digital filters. With this technique, we can use resource sharing for different computation. Consequently, this method simplified the harware requirements for complex computation of Discrete Cosine Transform (DCT).

The calculation for one output of DCT using our technique can be shown as follows :

$$\begin{aligned} z(5) &= j \times \sqrt{2} \\ &= (e-h) \times \sqrt{2} \\ &= [(d \times \cos\frac{3\pi}{16}) + (a \times \sin\frac{3\pi}{16}) \\ &- (b \times \cos\frac{\pi}{16}) + (c \times \sin\frac{\pi}{16})] \times \sqrt{2} \\ &= [(\sqrt{2} \times d \times \cos\frac{3\pi}{16}) + (\sqrt{2} \times a \times \sin\frac{3\pi}{16}) \\ &- (\sqrt{2} \times b \times \cos\frac{\pi}{16}) + (\sqrt{2} \times c \times \sin\frac{\pi}{16})] \end{aligned}$$

with every value of each constant shown in Fig (2), we can simplified the calculation by :

$$\begin{aligned} z(5) \quad &= 2^0(c+d-b) + 2^1(a+d) + 2^3(b+d) \\ &+ 2^4(c-a) - 2^5(b) + 2^6(a-b+d) \end{aligned} \quad (5)$$

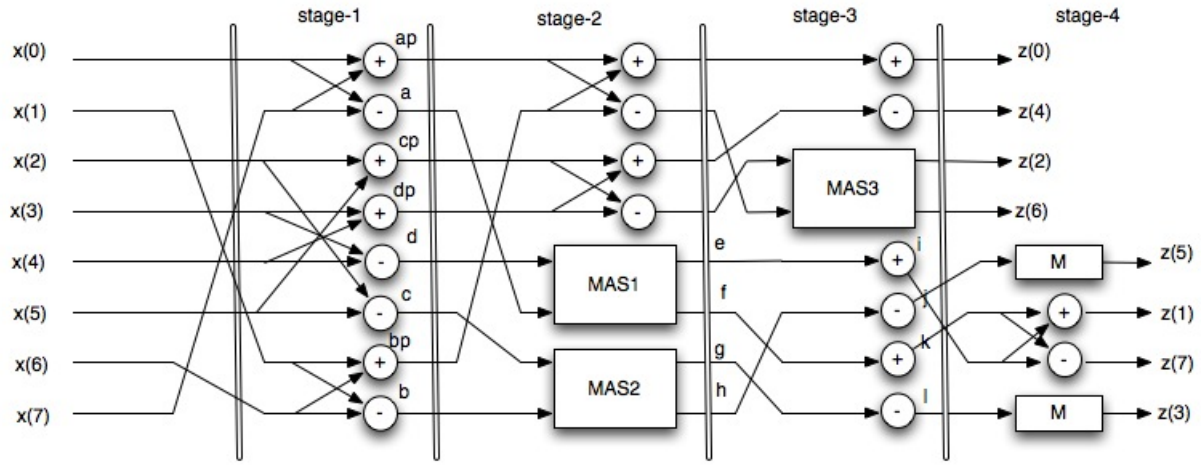


Fig. 1. Loeffler base 1-D DCT


so that, for $z(5)$ output, hence :

$$\begin{aligned} z(5) &= [(c+d-b) >>> 6] + [(a+d) >>> 5] \\ &+ [(b+d) >>> 3] + [(c-a) >>> 2] - [b >>> 1] \\ &+ (a-b+d) \end{aligned} \quad (6)$$

The operator $(>>>)$ represent arithmetic right shift. In the same way, we can get another expression in 1-D DCT as follows :

$$z(0) = (ap + dp + bp + cp) \quad (7)$$

$$\begin{aligned} z(1) &= (a+b+c+d) + [(a-d) >>> 1] \\ &- [(a-b+c+d) >>> 2] + [(a-b+a+c) >>> 4] \\ &+ [(a-b-c+d) >>> 6] + [(a-b+c-d) >>> 7] \end{aligned} \quad (8)$$

$$\begin{aligned} z(2) &= (ap-dp) + [(bp-cp) >>> 1] \\ &+ [(ap-dp) >>> 2] + [(ap-dp) >>> 4] \\ &+ [(bp-cp) >>> 5] + [(bp-cp) >>> 6] \end{aligned} \quad (9)$$

$$\begin{aligned} z(3) &= (a-c-d) - [(c) >>> 1] \\ &+ [(d-b) >>> 2] + [(a+c) >>> 3] \\ &+ [(a-d) >>> 5] + [(a-b-c) >>> 6] \end{aligned} \quad (10)$$

| Real Value | CSD Number |
|---|---|
| $0.5 \times \cos \frac{3\pi}{16}$ | 0011 0101 |
| $0.5 \times \sin \frac{3\pi}{16}$ | 0010 0100 |
| $0.5 \times \cos \frac{\pi}{16}$ | 0100 000$\bar{1}$ |
| $0.5 \times \sin \frac{\pi}{16}$ | 0001 0$\bar{1}$00 |
| $\sqrt{2} \times 0.5 \times \cos \frac{3\pi}{16}$ | 0100 1011 |
| $\sqrt{2} \times 0.5 \times \sin \frac{3\pi}{16}$ | 010$\bar{1}$ 0010 |
| $\sqrt{2} \times 0.5 \times \cos \frac{\pi}{16}$ | 0110 $\bar{1}$001 |
| $\sqrt{2} \times 0.5 \times \sin \frac{\pi}{16}$ | 0001 0001 |
| $\sqrt{2} \times 0.5 \times \cos \frac{6\pi}{16}$ | 0010 0011 |
| $\sqrt{2} \times 0.5 \times \sin \frac{6\pi}{16}$ | 0101 0100 |

Fig. 2. Canonical Signed Digit Representation of Coefficoents

$$z(4) = (ap + dp - bp + cp) \quad (11)$$

$$\begin{aligned} z(5) &= (a-b+d) - [(b) >>> 1] \\ &+ [(c-a) >>> 2] + [(b+d) >>> 3] \\ &+ [(a+d) >>> 5] + [(c+d-b) >>> 6] \end{aligned} \quad (12)$$

$$\begin{aligned} z(6) &= [(ap-dp) >>> 6] + [(ap-dp) >>> 5] \\ &- [(bp-cp) >>> 4] - [(bp-cp) >>> 2] \\ &+ [(ap-dp) >>> 1] - (bp-cp) \end{aligned} \quad (13)$$

$$\begin{aligned} z(7) &= (a-b+c-d) - [(a+d) >>> 1] \\ &- [(a-b-c-d) >>> 2] - [(c+d+b+d) >>> 4] \\ &+ [(a+b-c-d) >>> 6] + [(a+b+c+d) >>> 7] \end{aligned} \quad (14)$$

### B. Two-dimensional Discrete Cosine Transform

Two dimensional discrete cosine transform formula is described in equation (3) and the inverse is described in equation (4). In order to implement the equation, we need two designs of one-dimensional discrete cosine transform and a transposition buffer.

By using a transposition buffer, although it use large amount of memory, it saves a lot of area instead of calculating each row twice and make more number of one-dimensional discrete cosine transform.

The top design of our 2-D DCT can be viewed in Figure 3.


Fig. 3. 2-D DCT top design

The design inputs a matrix row every clock cycle. For 8x8 matrices, each matrix will need 8 clock cycles to be inputted in first 1-D DCT. We also proposed an idea for transposition buffer that can handle continuous multiple matrices input. For example, if two matrices M and N inputted consecutively, the result of the transposition, $M^T$ and $N^T$ will be in consecutive manner. Simulation result of the transposition buffer for a 8x8 matrix can be viewed in Figure 4.

Our ideas for simulation buffer is create a 8x8 register array as temporary storage and a 7x8 register array for transposition processing. Design of our transposition buffer shown in figure 5.

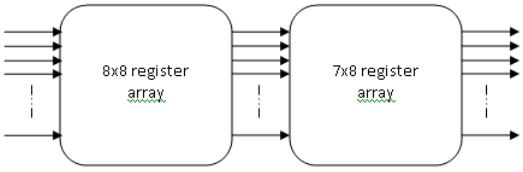


Fig. 5. transposition buffer

In first 8 clock cycle, we input each row every clock cycle, then each clock after that we transpose each row in second 7x8 register array and also inputs each row of second matrix. After that, the transposition result will enter second one-dimensional discrete cosine transform hardware. The output of it will be its two-dimensional discrete cosine transform result. We need total 22 clock cycles to get a two-dimensional discrete cosine transform of a matrix.

## IV. HARDWARE DESIGN IMPLEMENTATION

To measure our design, we implement 2-D Discrete Cosine Transform using our 1-D Discrete Cosine Transform algorithm as described in previous section in verilog hardware design language. First, we create mathematical model of our algorithm using MATLAB®. Second, we create register-transfer-level design using verilog HDL. Then, we simulate our design using ModelSim and synthesize our design in Synopsis software.

We represent each element of 8x8 matrices as fixed points with 20 bits length. First 15 bits represent sign and integer part and the 5 bits LSB represent fraction. With our design, we can tide the DCT operation at 81.95 MHz of clock frequency, achieved low power consumption that only consume 8.42 mW at top design, 44 clock cycles pipeline latency.

## V. CONCLUSION

We present our low-power, high speed, and simple computational algorithm of Discrete Cosine Transform (DCT) function and Inverse Discrete Cosine Transform (IDCT) function based on loeffler algorithm with enhance in multiplierless approach. With this algorithm, we can reduce computational area, complexity and power consumption.

## REFERENCES


[1] N. Ahmed, T. Natarajan, and K. R. Rao "Discrete Cosine Transform," *IEEE Transactions Computers*, pp. 90-93, 1974.

[2] ITU-T recommendation T-81, *Digital Compression of Continuous Tone and Still Images*, September, 1992.

[3] W.H. Chen, C.Smith, S. Fralick "A Fast Computational Algorithm for the Discrete Cosine Transform ," *IEEE Transactions on Communications* 25, pp. 1004-1009, 1977.

[4] J.Park and K.Roy, "A low power reconfigurable DCT architecture to trade off image quality for computational complexity," *IEEE Conference on Acoustics, Speech, and Signal Processing*, V, pp. 17-20, 2004.

[5] C.Loeffler, A.Lightenberg, and G.S. Moschytz "Practical fast 1-D DCT algorithm with 11 multiplication," *IEEE Conference on Acoustics, Speech, and Signal Processing*, pp. 988-991, May 1989.

[6] P. Duhamel, H. H'Mida "New $2^n$ DCT algorithms suitable for VLSI implementation," *IEEE Conference on Acoustics, Speech, and Signal Processing* 12, pp. 18051808, 1987.

[7] R.M. Hewlitt and E.S. Swartzlantler "Canonical signed digit representation for FIR digital filters," *IEEE Workshop on Signal Processing Systems*, pp. 416-426, 2000.

[8] R. Seegal "The canonical signed digit code structure for FIR filters," *IEEE Conference on Acoustics, Speech, and Signal Processing*, 28, pp. 590 - 592, 1980.

[9] R.I. Hartley "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Transactions on Circuits and Systems II : Analog and Digital Signal Processing*, Vol. 43 No. 10, pp. 677 - 688, 1996.

[10] M.Jridi and A.Alfalou, "A Low-Power, High-Speed DCT architecture for image compression: principle and implementation," *VLSI System on Chip Conference (VLSI-SoC)*, pp. 304-309, 2010.

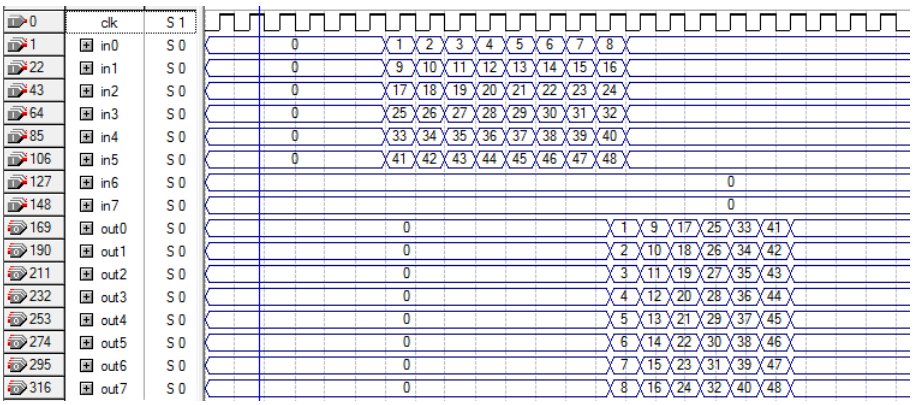


Fig. 4. Transposition Buffer Simulation Result