

An Efficient Hardware Realization of Distributed Arithmetic Based Discrete Cosine Transform

M. Satya Rajeev* and P. Augusta Sophy Beulet

School of Electronics Engineering, VIT University, Chennai – 600048, Tamil Nadu, India;
makka.satyarajeev2013@vit.ac.in, maugustasophyt.p@vit.ac.in

Abstract

Objective: This paper presents an efficient design for the implementation of Discrete Cosine Transform (DCT) which is widely used in still and motion picture compression. In this work cyclic convolution and Grouped Distributed Arithmetic (GDA) technique are used. **Method:** In GDA, Read Only Memory (ROM) size could be reduced compared to conventional DA. For example, 1-D DCT of transform length $N=13$, the proposed design requires 31 Look-Up Tables (LUTs) while conventional DA requires 384 LUTs. **Findings:** From the synthesis results, it is found that proposed design has less area requirement and low power consumption compared to the conventional DA of same length. **Conclusion:** This paper proposes an efficient design for implementation of cyclic convolution based on modified GDA technique.

Keywords: Cyclic Convolution, Discrete Cosine Transform (DCT), Discrete Cosine Transform (DCT), Grouped Distributed Arithmetic (GDA), Read Only Memory (ROM)

1. Introduction

Transform computations such as Fourier Transform, Fast Fourier Transform, Discrete Cosine Transform and Discrete Wavelet Transform are most widely used in various signal and image processing applications. Discrete Cosine Transform (DCT) is one of the widely used transforms in digital signal processing. There is a wide range of applications for DCT such as video coding, image coding, watermarking, face recognition, finger print feature extraction and many other applications in image processing. Several architectures are suggested for efficient computation for DCT. Major Architectures which are proposed for implementing DCT is for a length, which is power of two^{2,4,5,6} and prime length^{1,3,8} DCT. For example, in video processing power of two lengths DCT is used and some other applications like voice multiplexing, power of two lengths DCT is not preferred because of the wide difference between two successive transform lengths. This paper presents an efficient DCT algorithm and its

mapping into a hardware architecture, where transform length is a prime number.

Distributed Arithmetic (DA) is a widely popular and more hardware efficient technique used to design DCT. Main hardware component for DA is Look-Up Table (LUT), which stores the partial products. The major disadvantage of DA technique is ROM size. If transform length of DCT increases, ROM size also increases. Grouped Distributed Arithmetic is used to reduce the ROM size^{1,7,9}. This paper presents architecture with low power consumption and less. In this work, Modelsim is used for simulation and Cadence® RTL Compiler with 180nm technology library is used for synthesis.

The rest of the paper is organized as follows: Section 2 presents the derivation of the proposed algorithm. Section 3 describes the proposed approach with cyclic convolution and DCT architecture. Section 4 discusses the comparison of power consumption and area requirement between GDA based DCT with conventional DCT and DA based DCT and Section 5 gives the conclusion of this proposed work.

*Author for correspondence

2. DCT

The DCT equation for transform length $N=7$ is shown in Equation (1).

$$Y(k) = \sum_{i=0}^{N-1} y(i) \cos \left[\frac{\pi(2i+1)k}{2N} \right] \quad (1)$$

$k = 0, 1, 2, \dots, N-1$

Where $y(i)$ and $Y(k)$ are the input sequence and DCT output sequence respectively. If the transform length is prime number, the DCT equation using cyclic convolution method can be written as in Equation (2).

$$y(0) = \sum_{n=0}^{N-1} y(n)$$

$$y(K) = \{2T(k) + x(0)\} \cos \left[\frac{kn}{2N} \right] \quad (2)$$

$$T(K) = \sum_{n=1}^{N-1} x(i) \cos \left[\frac{\pi ik}{N} \right] \quad (3)$$

Where

And $K \ll g^k \gg N$ and $i \ll g^{n-k+1} \gg N$

Where $k = 0, 1, 2 \dots N-1$

Where $\ll g^k \gg N$ denotes " g^k modulo N " for short, g is primitive number. For example transform length $N=7$, the value of primitive number is 3. $x(i)$ Sequence is defined as follows:

$$\begin{aligned} x(N-1) &= Y(N-1) \\ x(i) &= y(i) - x(i+1) \end{aligned} \quad (4)$$

Where $i = 0, 1, 2 \dots N-2$

By using Equations (3) and (4) with cyclic convolution, prime length DCT based on DA is proposed in the next section.

3. Proposed Design

3.1 Cyclic Convolution

Considering the example of cyclic convolution described in Equation (5). Here $\{v_1, v_2, v_3\}$ are the input sequence, $\{a, b, c\}$ are coefficients and $\{u_1, u_2, u_3\}$ are output sequence.

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} a & b & c \\ c & a & b \\ b & c & a \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad (5)$$

By using commutative property the above matrix can be written as in Equation¹⁰ (6).

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} v_1 & v_2 & v_3 \\ v_3 & v_1 & v_2 \\ v_2 & v_3 & v_1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} \quad (6)$$

From Equation (6) it can be observed that different outputs can be computed using the same input data with rotated order and the same set of coefficients $\{a, b, c\}$.

3.2 Cyclic Convolution based DCT

Considering the cyclic convolution as described in Equation (5) and (6), the DCT for transform length $N=7$ in Equation (3) can be transformed as in Equation (7).

$$\begin{pmatrix} T(3) \\ T(2) \\ T(6) \\ T(4) \\ T(5) \\ T(1) \end{pmatrix} = \begin{pmatrix} c_3 & c_1 & c_2 & -c_3 & -c_1 & -c_2 \\ c_2 & -c_3 & -c_1 & c_2 & -c_3 & -c_1 \\ -c_1 & c_2 & -c_3 & -c_1 & c_2 & -c_3 \\ -c_3 & -c_1 & c_2 & -c_3 & -c_1 & c_2 \\ -c_2 & c_3 & -c_1 & c_2 & -c_3 & c_1 \\ c_1 & -c_2 & -c_3 & -c_1 & c_2 & c_3 \end{pmatrix} \times \begin{pmatrix} x(1) \\ x(5) \\ x(4) \\ x(6) \\ x(2) \\ x(3) \end{pmatrix} \quad (7)$$

Where $C_j = \cos(j\pi/N)$ and $C_j = -C_{(N-j)}$

According to matrix symmetric property the above matrix can be rewritten by interchanging the inputs and coefficients is shown in Equation (8).

$$\begin{pmatrix} T(3) \\ T(2) \\ T(6) \\ T(4) \\ T(5) \\ T(1) \end{pmatrix} = \begin{pmatrix} x(1) & x(5) & x(4) & x(6) & x(2) & x(3) \\ -x(2) & -x(3) & x(1) & x(5) & x(4) & -x(6) \\ -x(4) & -x(6) & x(2) & x(3) & x(1) & -x(5) \\ -x(1) & -x(5) & x(4) & x(6) & x(2) & -x(3) \\ -x(2) & x(3) & -x(1) & -x(5) & x(4) & -x(6) \\ -x(4) & -x(6) & x(2) & -x(3) & -x(1) & x(5) \end{pmatrix} \times \begin{pmatrix} c_3 \\ c_1 \\ c_2 \\ -c_3 \\ -c_1 \\ -c_2 \end{pmatrix} \quad (8)$$

Data elements in matrix (8) can be merged as follows

$$\begin{pmatrix} T(3) \\ T(2) \\ T(6) \\ T(4) \\ T(5) \\ T(1) \end{pmatrix} = \begin{pmatrix} x(1)-x(6) & x(5)-x(2) & x(4)-x(3) \\ -(x(2)+x(5)) & -(x(3)+x(4)) & x(1)+x(6) \\ -(x(3)+x(4)) & -(x(1)+x(6)) & x(2)+x(5) \\ -(x(1)+x(6)) & -(x(2)+x(5)) & x(3)+x(4) \\ x(5)-x(2) & x(3)-x(4) & x(6)-x(1) \\ x(3)-x(4) & x(1)-x(6) & x(2)-x(5) \end{pmatrix} \times \begin{pmatrix} c_3 \\ c_1 \\ c_2 \end{pmatrix} \quad (9)$$

To separate the even and odd outputs given in Equation (9), two smaller and perfect cyclic convolution matrices are obtained as shown in Equation (10) and (11).

$$\begin{pmatrix} T(2) \\ T(6) \\ T(4) \end{pmatrix} = \begin{pmatrix} x(1)+x(6) & x(3)+x(4) & x(2)+x(5) \\ x(2)+x(5) & x(1)+x(6) & x(3)+x(4) \\ x(3)+x(4) & x(2)+x(5) & x(1)+x(6) \end{pmatrix} \times \begin{pmatrix} c_2 \\ c_6 \\ c_4 \end{pmatrix} \quad (10)$$

$$\begin{pmatrix} T(5) \\ T(1) \\ T(3) \end{pmatrix} = \begin{pmatrix} x(6) - x(1) & x(4) - x(3) & x(2) - x(5) \\ x(2) - x(5) & x(6) - x(1) & x(4) - x(3) \\ x(4) - x(3) & x(2) - x(5) & x(6) - x(1) \end{pmatrix} \times \begin{pmatrix} c_2 \\ c_6 \\ c_4 \end{pmatrix} \quad (11)$$

The transformations from Equation (8) to (11) are done using the trigonometric symmetric property of the DCT coefficient. Using this property the input data sequences can merge in the DCT matrix and then separating the matrix into two perfect cyclic forms, which gives an efficient realization of the DCT through the proposed design approach. It is useful to reduce the ROM size when trigonometric symmetry properties of the DCT coefficients are used.

3.3 Proposed Architecture

Figure 1 shows the proposed architecture of 1-D DCT. It consists of the preprocessing stage, the DA processing stage and the post-processing stage. In preprocessing stage, input buffer and preprocessing block are designed by using the bidirectional shifter and accumulator, which is used to get the $x(n)$ sequence from $y(n)$. Figure 2 shows the Distributed Arithmetic stage which is named as Grouped Distributed Arithmetic unit (GDAU) and it is designed with the proposed memory efficient approach to carry out computation of $T(K)$ in the design of seven point DCT. Due to the same content of group ROM, only one group ROM in the GDAU is required to compute the outputs of the separated cyclic

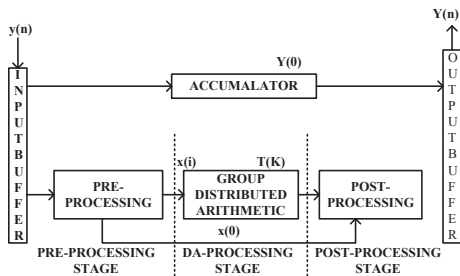


Figure 1. Proposed Architecture.

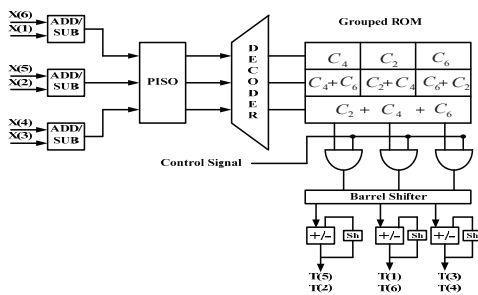


Figure 2. DA Processing Unit for 1-D 7-point DCT.

operations. Inputs are fed to the address decoder and this address decoder will compute the rotating factor and group address by decoding the input vector according to Table 1.

Table 2 shows the original partial product distributions for computing the outputs sequences of DCT matrix under the same input value and it is observed that the rotation relationship between the partial products is available, and then the ROM contents are rearranged in the proposed design as shown in Table 3. The need to store zeros in ROM is not required and this reduces the ROM size. When group address is '2', control signal will be low which disables the AND cells and otherwise control signal is high. Post processing includes multiplication with two followed by addition of $x(0)$ and multiplication with cosine terms.

Table 1. Rotating factor and Group address in the design of seven point DCT

DA input values	Rotating Factor	Group Address
001	0	0
010	1	
100	2	
011	0	1
110	1	
101	2	
000	0	2
111	1	3

Table 2. Partial Product Distribution for different inputs

Inputs	T(2)/T(5)	T(6)/T(1)	T(4)/T(3)	Group Address
000	0	0	0	2
001	C_4	C_2	C_6	0
010	C_6	C_4	C_2	0
011	$C_6 + C_4$	$C_2 + C_4$	$C_2 + C_6$	1
100	C_4	C_6	C_2	0
101	$C_2 + C_4$	$C_2 + C_6$	$C_6 + C_4$	1
110	$C_2 + C_6$	$C_6 + C_4$	$C_2 + C_4$	1
111	$C_2 + C_4 + C_6$			3

Table 3. Reduced ROM with group addresses

Original Address	Group Address	T(2)/T(5)	T(6)/T(1)	T(4)/T(5)
1,2,4	0	C_4	C_2	C_6
3,6,5	1	$C_6 + C_4$	$C_2 + C_4$	$C_2 + C_6$
7	3	$C_2 + C_4 + C_6$		

4. Results

The proposed design is synthesized using 180nm technology of the cadence[®] RTL Compiler. Table 4 and Table 5 shows the power and area results for transform length N=7 and N=13 respectively. The proposed structure has more number of adders and has less numbers of multiplier compared to conventional DCT. Size of ROM is more in DA based DCT compared with proposed design. For example for transform length N=7, proposed method needs 10LUTs while conventional DA needs 24 LUTs in ROM. By observing Table 4 and 5, it is found that proposed design has low power and less area compared to

Table 4. Power and Area calculations for transform length N=7 1-D DCT

	Power Consumption (mW)	Area (sq.μm)
General DCT	12.59	12,21,055
DA based DCT	11.49	11,24,124
Proposed Design	7.48	8,83,442

Table 5. Power and Area calculations for transform length N=13 1-D DCT

	Power Consumption (mW)	Area (sq.μm)
General DCT	14.38	44,79,267
DA based DCT	12.84	34,59,136
Proposed Design	10.75	18,39,556

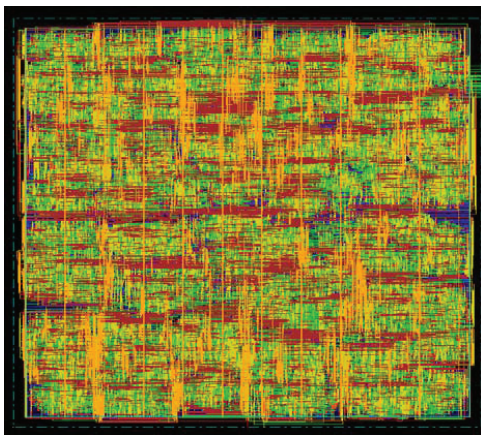


Figure 3. Layout view of 7- point 1-D DCT.

conventional DCT and DA based DCT but GDA based DCT takes longer time to produce the output compared to DA with DCT. The Layout view of the seven point DCT is shown in Figure 3.

5. Conclusion

This paper proposes an efficient design for implementation of cyclic convolution based on modified GDA technique. This GDA is applied to the proposed DCT design. From the synthesis results, it is found that the proposed design involves significantly less area with lower power consumption compared with conventional DCT and DA based DCT. The structure proposed here is highly regular, modular, and therefore, suitable for VLSI realization.

6. Reference

1. Chen H-C, Guo J-I, Chang T-S, Jen C-W. A memory-efficient realization of cyclic convolution and its application to discrete cosine transform. *IEEE Trans Circuits Systems Video Technology*. 2005 Mar; 15(3):445–53.
2. Meher PK, Unified systolic-like architecture for DCT and DST using distributed arithmetic. *IEEE Trans Circuits Systems*. 2006 Dec; 53(12):2656–63.
3. Yin R-X, Siu W-C. A new fast algorithm for computing prime-length DCT through cyclic convolutions. *Signal Processing*. 2001 May; 81(5):895–906.
4. Fanucci L et al., Parameterized and reusable VLSI macro cells for the low-power realization of 2-D Discrete-Cosine-Transform. *Microelectronics J*. 2001; 32(12):1035–45.
5. Yu S, Swartzlander EE. DCT implementation with distributed arithmetic. *IEEE Trans Computers*. 2001 Sep; 50(9):985–91.
6. Xanthopoulos T, Chandrakasan AP. A low-power DCT core using adaptive bit width and arithmetic activity exploiting signal correlations and quantization. *IEEE J Solid-State Circuits*. 2000 May; 35(5):740–50.
7. Guo JI, Liu CM, Jen CW. The efficient memory-based VLSI array designs for DFT and DCT. *IEEE Trans Circuits Syst II, Exp Briefs*. 1992 Oct; 39(10):723–33.
8. Chan YH, Siu WC. On the realization of discrete cosine transform using the distributed arithmetic. *IEEE Trans Circuits Syst I, Reg Papers*. 1992 Sep; 39(9):705–12.
9. Choi JB, Shin SC, Chung JG. Efficient ROM size reduction for distributed arithmetic. *Proc IEEE Int Symp Circuits and Systems*. 2000 May; 1161–4.
10. Agarwal RC, Cooley JW. New algorithms for digital convolution, *IEEE Trans Acoust speech. Signal Process*. 1977 Oct; ASSP-25(5):392–410.