

A SIMPLE ARCHITECTURE FOR IMPLEMENTATION OF 8 - POINT DISCRETE COSINE TRANSFORM

Priyabrata Mohapatra¹, Sudhansu Sekhar Nayak²

¹ Research Scholar, Centurion University of Technology and Management, Odisha, India

² Professor in Physics, Centurion University of Technology and Management, Odisha, India

Abstract

An architecture for implementation of 8 - point discrete cosine transform (DCT) is suggested in this paper. The architecture is simple and suitable for VLSI implementation. The hardware time and area complexities are It has less number of multipliers.

Key words: Discrete cosine transform, VLSI, Systolic architecture, Discrete Fourier transform

Introduction

The discrete cosine transform (DCT) is a real valued orthogonal transform. Its forward transform can be obtained by taking the transpose of its inverse transform. It transforms a signal or image from spatial domain to frequency domain. It is widely used in the field of digital and signal and image processing. So many efficient algorithms have been developed for fast computation of the DCT.

N- point DCT can be computed using 2N-point Fast Fourier transform (FFT) [1]. The DCT could be computed using fast Hartleytransform (FHT) of same length [2]. Several VLSI algorithms and architectures have been proposed for implementation of DCT [3, 4].

A simple architecture for direct implementation of 8-point DCT has beenproposed by us in this paper. The Multipliers used in an architecture increase area. Hardware complexity increases with increased number of multipliers. The multipliers increase the delay in the operation. The proposed architecture uses less number of multipliers, and it is suitable for VLSI implementation.

The algorithm for implementation of DCT is given in Section - II. Section-III deals with the architecture for implementation of DCT. Conclusion remarks are given on the section - IV .

Algorithm

Let $\{x_n, n = 0,1,2,\dots,N-1\}$ be a given input data sequence. Its DCT data sequence $\{X_k; k= 0,1,2, \dots, N-1\}$ is given by the following relation

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

8-pointDCT can be expressed as

$$X_k = \sum_{n=0}^7 x_n \cos \left[\frac{\pi(2n+1)k}{16} \right] \quad (2)$$

It can be shown that

$$X_0 = A+B+C+D \quad (3.a)$$

$$X_1= A' \cos \pi/16 + B' \cos 3\pi/16 + C' \cos 5\pi/16 + D' \cos 7\pi/16 \quad (3.b)$$

$$X_2 = (A-D) \cos 2\pi/16 + (B-C) \cos 6\pi/16 \quad (3.c)$$

$$X_3 = -C' \cos \pi/16 + A' \cos 3\pi/16 - D' \cos 5\pi/16 - B' \cos 7\pi/16 \quad (3.d)$$

$$X_4 = (A - B - C + D) \cos 4\pi/16 \quad (3.e)$$

$$X_5 = -B' \cos \pi/16 + D' \cos 3\pi/16 + A' \cos 5\pi/16 + C' \cos 7\pi/16 \quad (3.f)$$

$$X_6 = (-B' + C) \cos 2\pi/16 + (A - D') \cos 6\pi/16 \quad (3.g)$$

$$X_7 = -D' \cos \pi/16 + C' \cos 3\pi/16 - B' \cos 5\pi/16 + A' \cos 7\pi/16 \quad (3.h)$$

Where,

$$A = x_0 + x_7 \quad A' = x_0 - x_7 \quad (4.a)$$

$$B = x_1 + x_6 \quad B' = x_1 - x_6 \quad (4.b)$$

$$C = x_2 + x_5 \quad C' = x_2 - x_5 \quad (4.c)$$

$$D = x_3 + x_4 \quad D' = x_3 - x_4 \quad (4.d)$$

8-point DCT can be implemented straight forward using equations (3.a) to (3.h).

Proposed Architecture

The proposed architecture for the implementation of 8-point DCT is shown in Fig.1. Our primary objective is to propose a simple architecture for the implementation of 8-point DCT. The proposed architecture consists of 14 adders and 16 Subtractors. In any architecture multipliers introduce significant delay and increase area - and hardware complexities. Using less number of multipliers is always advantageous to the designing of an architecture. So we have used only seven multipliers $M_1, M_2, M_3, M_4, M_5, M_6$ and M_7 . The multiplicands stored in M_4, M_5, M_6 and M_7 are $\cos \pi/16, \cos 3\pi/16, \cos 5\pi/16$ and $\cos 7\pi/16$ respectively.

Multiplicands stored in M_1, M_2, M_3 are $\cos 4\pi/16, \cos 2\pi/16$ and $\cos 6\pi/16$ respectively. Total number of multiplications is equal to 21. The inputs x_1, x_2, \dots, x_7 are available simultaneously. At the end of a first time-step A, A', B, B', C, C', D and D' are obtained. One time-step, T , is the time taken for a multiplication by the multiplier. The first output X is obtained at the end of 4th time - step which needs only three additions. $(A-B-C+D)$ is obtained at the end of 4th time-step. Output X_4 is obtained at the end of 5th time-step. $(A-D)$ and $(B-C)$ are input to M_2 and M_3 , respectively, at the end of a second time-step. So output X_2 is obtained at the end of 4th time-step. $(C-B')$ and $(A-D')$ are input to M_2 and M_3 , respectively, at the end of 3rd time-step. So output X_6 is obtained at the end of 5th time-step. At the end of 1st time Step, A', B', C' and D' reach the buffer. They are fed to the multipliers M_4, M_5, M_6 and M_7 in different orders. The time gap between successive inputs is equal to one time - step. At the end of 2nd time-step, A', B', C' and D' are fed to M_4, M_5, M_6 and M_7 , respectively. At the end of 3rd time-step, C', A', D' and B' are fed to M_4, M_5, M_6 and M_7 , respectively. At the end of 4th time - step B', D', A' and B' are fed to M_4, M_5, M_6 and M_7 , respectively. At the end of 5th time - step $D', C' B'$ and A' are fed to M_4, M_5, M_6 and M_7 , respectively. The successive outputs of M_4, M_5, M_6 and M_7 are added to get X_1, X_3, X_5 and X_7 in the 7th time-step. The outputs X_0, X_7 , can be obtained in an order by using a number of delays.

Table: Comparison of number of multipliers, number of adders and number of subtractors used in the proposed architecture with those of [4] for 8-point DCT.

No. of Multipliers		No. of adders		No. of subtractors	
Architecture [4]	Proposed Architecture	Architecture [4]	Proposed Architecture	Architecture [4]	Proposed Architecture
32	7	36	14	4	16

Conclusion

The architecture proposed in this paper is very simple. It uses 14 adders, 16 subtractors and 7 multipliers. The number of multipliers is very less compared to those of existing architectures. The architecture proposed in [4] requires 32 multipliers for 8 point DCT. Area - and hardware - complexities of the proposed architecture are small compared to those of [4].

References

1. N. Ahmed T. Natarajan and K.R.Rao, "Discrete Cosine transform", IEEE Trans. on Computers, Vol. C-23, pp. 90-93 Jan. 1974.
2. H. S. Halvar, "Fast computation of discrete cosine transform through fast Hartley transform" Electron Lett., Vol. 22, pp. 353 - 354, March 1986.
3. SBoussakta and O. Alshibami, "Fast algorithm for the 3-D DCT - II", IEEE Trans. Signal Processing, vol. 52, pp. 992-1001, April 2004.
4. A. Aggoun, "Three dimensional DCT / IDCT architecture" IJAET, pp. 648-658, May 2013

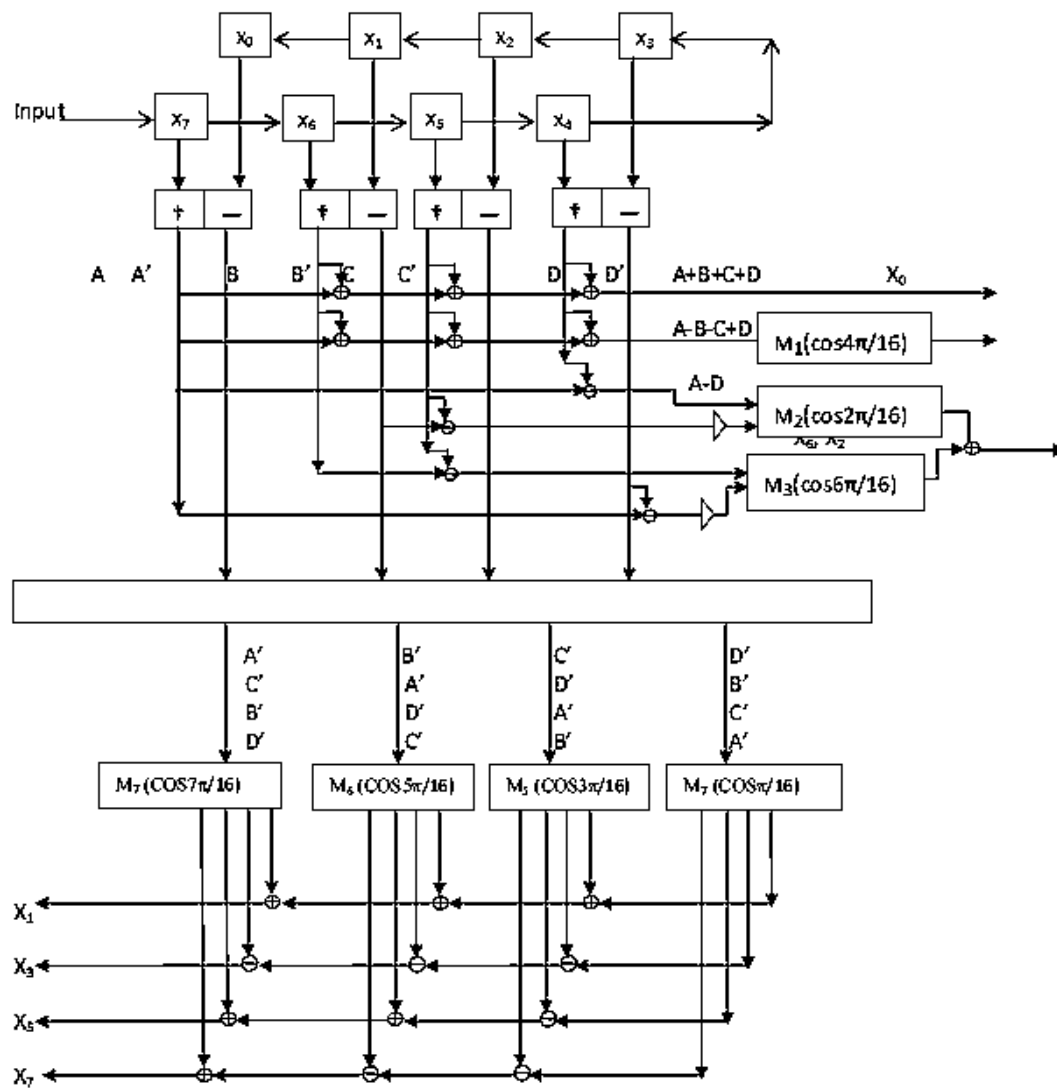


Fig-1: Architecture for implementation of 8-point DCT

▷ --delay