International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 3, Issue 5, May 2014 **ISSN 2319 - 4847**

DCT/IDCT Implementation with Loeffler Algorithm

NEHA V. MAHAJAN1, Dr. J. S. CHITODE2

[1]Research fellow, Dept. of Electronics, BVDU, College of Engineering, Pune, India
[2]Dept. of Electronics, BVDU, College of Engineering, Pune, India

ABSTRACT

The discrete cosine transform (DCT) is a robust approximation of optimal Karhunen-Loeve transform for first order Markov source with large correlation coefficient. DCT is the underlying technique for many image and video compression standards such as JPEG, H.163 and MPEG, due to its high energy compaction capability and also became substantial bottlenecks in the contemporary visual data compression algorithms. The purpose of this paper is to propose a scheme to design 8-point DCT and IDCT with faster implantations by scaling and approximating the coefficients of floating point values to integer values. This can be achieved by multiplying floating point value with 2^s. The design architecture is written in Verilog HDL code using Modelsim Altera and XILINX ISE tools. The architecture is modelled and synthesized using RTL (Register Transfer Level) abstraction.

Keywords: DCT, IDCT, RTL, Loeffler, Verilog.

1. INTRODUCTION

A common research question is range of performance improvements that may be achieved by augmenting general purpose processor with reconfigurable core. The basic idea of such approach is to exploit both general purpose capability and to achieve better performance for large class of applications. FPGA provides that flexibility to implement application specific computations. Such DCT/IDCT implementation mapped on FPGA will discuss here [7].

Transform coding constitutes an integral component of substantial bottlenecks in visual data compression algorithms, filtering and other fields. Many DCT algorithms with efficient hardware software algorithms have been proposed. It has become a heart of international standard such as JPEG, H.26x, and MPEG family.

There are four types of DCT labeled as I-IV [2]. Among them, DCT type II is mostly used. This is used in JPEG and video codec. The theoretical lower bound on the number of multiplications required for 1 D eight point DCT has been proven to be 11. In this sense, the method proposed by Loeffler with 11 multiplications and 29 additions [1] is most efficient solution.

The entire fast algorithm still require floating point multiplication which is slow in both hardware and software implementation. To achieve faster implantation, coefficients can be scaled and approximated by integer such as floating point multiplication can be replaced by integer multiplication [6].

This can be done by rounding floating point value to integer value by multiplying floating point value with 2^s. WhereS can be any integer number? This is called as fixed point arithmetic. The resulting algorithms are much faster than the original version and therefore have wide practical applications.

In this paper we propose a high speed, better accuracy 8 point DCT and IDCT architecture based on Loeffler DCT algorithm. The remainder of the paper is organized as follows. Section II describes the basic theoretical background of DCT and IDCT. Many different algorithms are discussed for implementation DCT and IDCT. Section III is practical implementation of algorithm. This section includes Loeffler algorithm, which is used for implementation. Subsection of section III describes DCT and IDCT details of Loeffler algorithm. It also shows main components that is butterfly unit and rotator unit for both DCT and IDCT. Section IV provides simulation results. Last section concludes the paper.

2. OVERVIEW OF DCT

The DCT transforms data from spatial domain to spatial frequency domain. It de-correlate the data which is highly correlated. Correlation provides much redundant information. The de-correlation and energy compaction properties of transform have been exploited to achieve high compression ratios in MPEG and JPEG. The transformation for N point 1-D DCT is defined as follows :A given data sequence $\{x_n,\ n = 0,1,2,3,\ldots..N-1\}$ is transformed into output sequence $\{y_k, k - 0,1,2\,3,\ldots..N-1\}$ by function given in the equation (1).

$$y(k) = \frac{2}{N} c_k \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)k\Pi}{2N}\right], k = 0,1,\ldots..N-1 \qquad (1)$$

As DCT is used for compression IDCT is used for decompression. IDCT is exact reverse process of DCT. The transformation for N point 1-D IDCT is defined as follows: A given input data sequence $\{y_n,\ n = 0,1,2,3,\ldots\ldots N-1\}$ is transformed into output sequence $\{x_k, k = 0,1,2,3,\ldots\ldots,N-1\}$ by function given in equation (2).

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

where k=0

Where $c_k = \begin{cases} 1/\sqrt{2} \\ 1 \end{cases}$ where k≠0

For DCT and IDCT these must satisfy equation 3 in order to obtain the original signal unscaled after the forward and inverse transformation [1].

$$c_{dct} \cdot c_{idct} = \frac{4}{N^2} \quad (3)$$

DCT and IDCT are highly computational intensive which creates prerequisites for performance bottlenecks in system utilizing them. To overcome this problem number of algorithms has been proposed for more efficient computations of this transform. Table 1 provides the information about multipliers and adders requires for different DCT algorithms [1].

Table 4 Multipliers and adders requires for all different DCT algorithms

Author	Chen	Wang	Lee	Vetterli	Suehiro	Hou	Loeffler
Multiplications	13	13	12	12	12	12	11
Additions	29	29	29	29	29	29	29

Chen proposed a recursive algorithm to factor any N- point DCT-II with N= 2^m,m ≥ 2 into plane rotations and butterflies [2]-[3]. The factorization has very general structure and it is six times as fast as the DFT based DCT algorithm. The method was generalized by Wang with size of power 2 [4]. As multiplications required in Loeffler algorithm is 1 less than other algorithms we are implementing DCT and IDCT with Loeffler algorithm.

3. PRACTICAL IMPLEMENTATION LOEFFLER DCT ALGORITHM

In this paper we will concentrate on the four- point, eight - point transforms since they are most useful in practical applications. Block transforms of other sizes can be design in same fashion. A more elegant factorization for eight point DCT was proposed by Loeffler as shown in figure (1). It also contains the scaled four point DCT. This method needs only 11 multiplications and 29additions. One of its variations is adopted by JPEG group.

3.1 Forward DCT

This algorithm has to be executed sequentially and cannot be evaluated parallel because of data dependencies. However calculations inside stages can be done parallel. This algorithm is divided into four different stages. In first stage inputs are taken and processed as per butterfly structure. In the second stage the algorithm separates out even odd coefficients. Even part of coefficients shows four - point DCT. In stage three again separates out even and odd coefficients. Last stage provides outputs and it uses coefficients c_k in order to obtain exact unscaled output. For DCT implementation notes that factorization requires a uniform scaling factor of $1/\sqrt{8}$ at the end of flow graph to obtain true DCT coefficients.

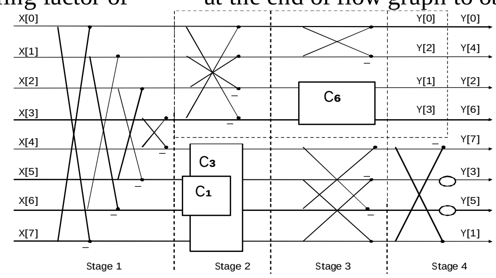


Figure 18 Flow graph for forward DCT with Loeffler's factorization

3.1.1 Butterfly Unit

Butterfly unit is one of the basic units of transforms. Butterfly structure and equations regarding it is presented in the figure 2. I0 and I1 are inputs for unit and outputs are named by O_0andO_1.

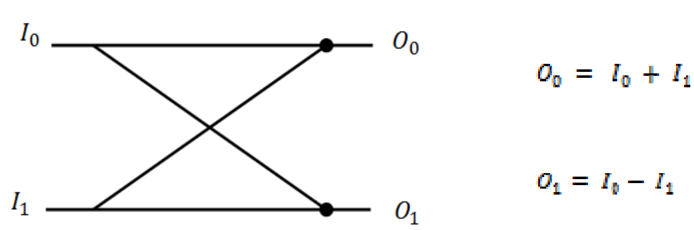


Figure 19 Butterfly unit for DCT

3.1.2 Rotator unit

The rectangular block presents a rotator, which transforms a pair of inputs $[I_0, I_1]$ into outputs $[O_0, O_1]$.There are three types of rotator as shown in following figure 3[7]-[5].

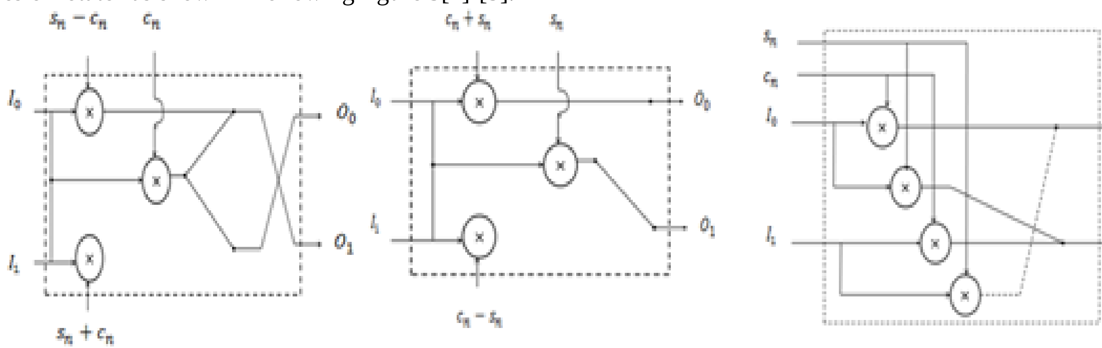


Figure 20 Three possible implementation of rotator

Although an implementation of such rotator with three multiplication and three additions is also possible, we use the direct implementation of rotator with four multiplication and two additions. We have preferred this because it shortens the critical path and improves accuracy. Critical path of implementation with four multiplier needs only two operations, one addition and another is multiplication, but in the case of three multipliers it took three operations, two additions and one multiplication. Also initial addition involve in three multiplier implementation may lead to an overflow when fixed point arithmetic is done. Therefore we used last method for implementation.

Figure 4 shows the rotator unit for our implantation and associated equations for rotator.

$$O_0 = I_0.k.\cos\left[\frac{n\pi}{16}\right] + I_1.k.\sin\left[\frac{n\pi}{16}\right]$$
$$O_1 = I_1.k.\cos\left[\frac{n\pi}{16}\right] - I_0.k.\sin\left[\frac{n\pi}{16}\right]$$

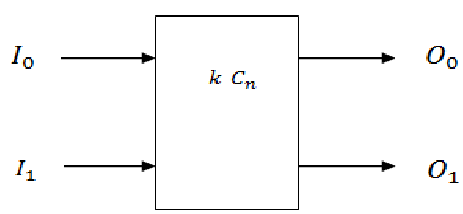


Figure 21 Rotator unit for DCT

3.2 Inverse DCT

In paper [1] it is mentioned that inverse DCT uses exactly the same arithmetic structure as DCT but in reverse order. Outputs thus become input and vice versa. As this is reverse process, in DCT we multiplied every output with uniform coefficient $1/\sqrt{8}$ to obtain unscaled coefficient of DCT now in IDCT all the inputs are multiplied with $\sqrt{8}$first.Then scaling coefficient $\sqrt{2}$was multiplied for obtaining accurate output, here in IDCT we have to multiply scaling coefficients with $1/\sqrt{2}$. This is shown in the figure 5.

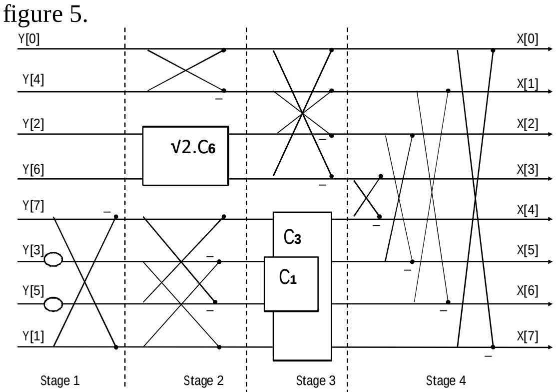
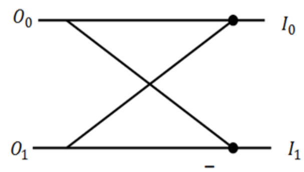


Figure 22 Flow graph for IDCT with Loeffler's factorization

3.2.1 Butterfly Unit

Figure 6 presents the butterfly unit for IDCT. Structural diagram is same as it is in DCT but the equations used for this are different as sown in the figure 6.



$$I_0 = \frac{(O_0 + O_1)}{2}$$

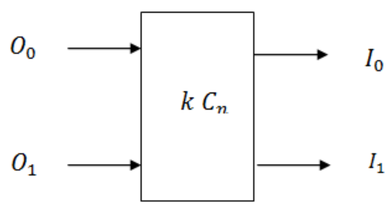
$$I_1 = \frac{(O_0 - O_1)}{2}$$

Figure 23 Butterfly Unit for IDCT

As shown in figure we have to use division operation at the output which may take time, so we are using right shift operation instead of division. As the programming is done using verilog HDL operator >>is used for right shift.

3.2.2 Rotator Unit

Rotator unit structure is also as same as that of DCT but few changes in the equations of rotator. Figure 7 shows structure for rotator of IDCT.



$$I_0 = O_0\,.k.\cos\left[\frac{n\pi}{16}\right] - O_1.k.\sin\left[\frac{n\pi}{16}\right]$$

$$I_1 = O_1\,.k\,.\cos\left[\frac{n\pi}{16}\right] + O_0.\ k\,.\sin\left[\frac{n\pi}{16}\right]$$

Figure 24 Rotator Unit for IDCT

4. SIMULATION RESULTS

We present the design of 8 point DCT and IDCT transforms with Loeffler algorithm. The verilog code has been successfully simulated on ModelSim ALTERA(version 6.3g_p1) Quartus II 8.1 and synthesized using Xilinx ISE (version 13.2).

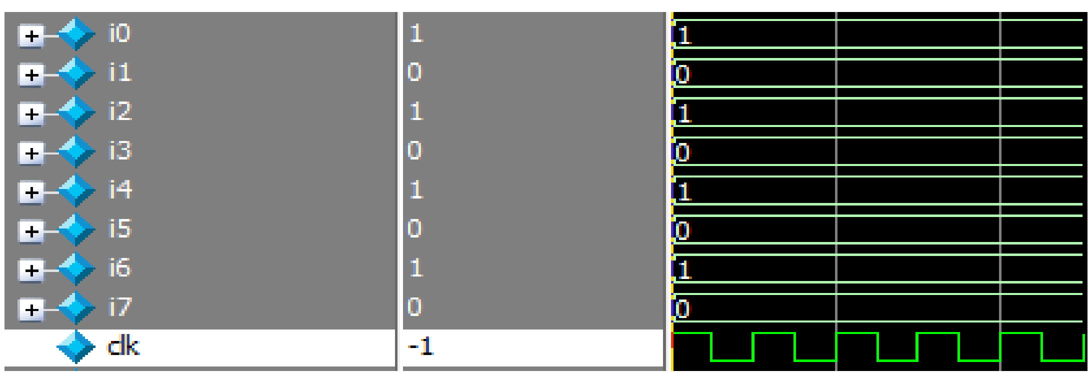


Figure 25 input for DCT

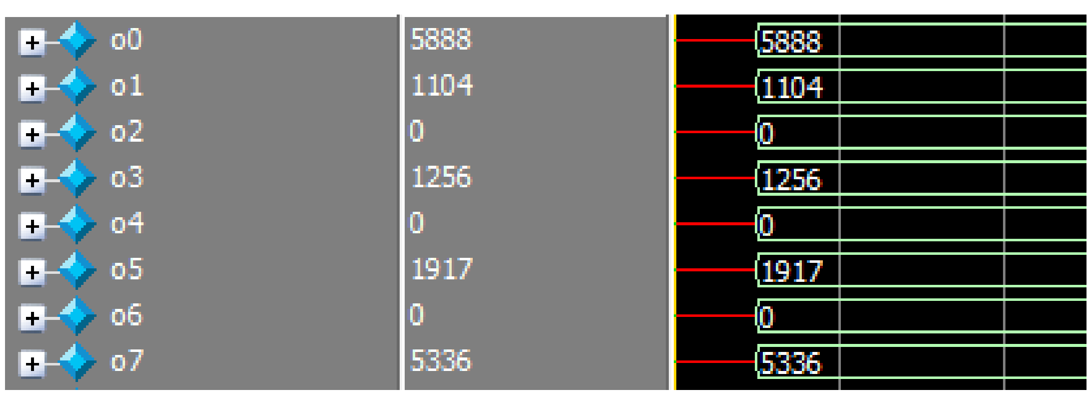


Figure 26 output for DCT

This is verified with MATLAB code for DCT/IDCT. There are total 8 inputs all are named as i0,i1,i2,i3,i4,i5,i6,i7 which is given as {1,0,1,0,1,0,1,0} and output named as o0,o1,o2,o3,o4,o5,o6,o7.The operation is done at positive edge of clock. Figure 8 shows inputs for DCT algorithm. Figure 9 displays output for input provided in figure 8 for DCT.DCT output are in the form of 2^{11} magnitude this is shown in the below table. After dividing output by 2^{12} i.e. 4096. Dividing output by $\sqrt{8}$ is done in the program itself.

Inputs for IDCT are shown in figure 10. Outputs for IDCT for given input in figure10 is shown in figure11. Output of IDCT is in the magnitude of 2^6 i.e.64. Output O1, O2, O5,O6 are again divided by 2 to get the original accurate output.

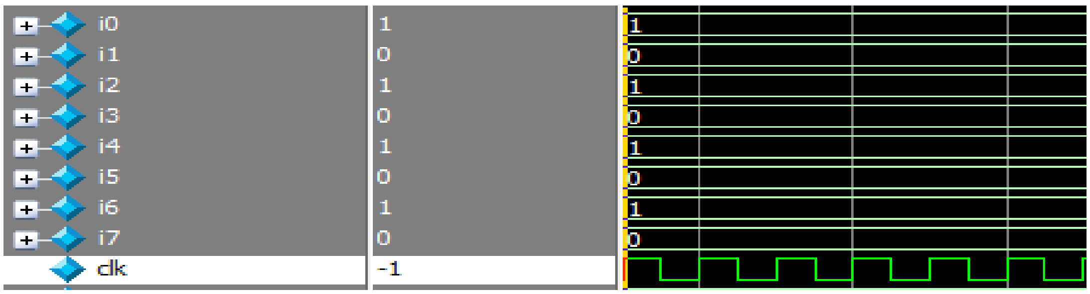


Figure 27 Input for IDCT

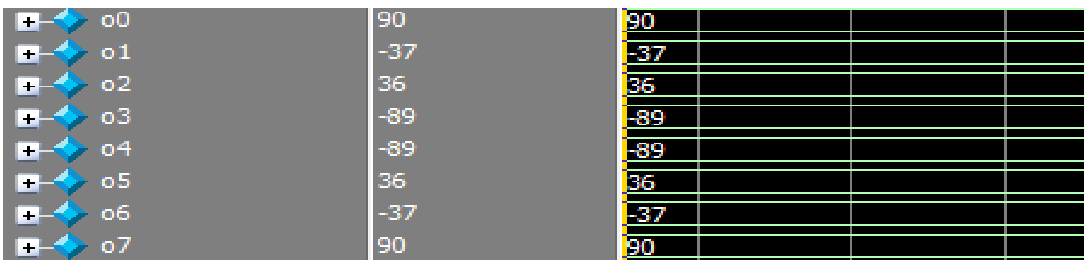


Figure 28 Output for IDCT

5. CONCLUSION

In this paper, a novel 8-point DCT/IDCT processor is implemented using Loeffler factorization. This paper also describes how to avoid floating point arithmetic for implementation of DCT/IDCT. Minimum 11 multiplications are used for implantation. In future, the work can extended to the N bit variable input signals. The implemented design can be used as a basic block for further computation. The pipelined architecture can also be added to DCT and IDCT. The proposed processor can be integrated with other components which can be used as a stand-alone processor for many applications.

REFERENCES

[1] Christoph Loeffler ,Adriaan Ligtenberg Moschytz “Practical fast 1-D DCT algorithm with 11 Multiplications,” Proc. IEEEICASSP,vol 2,pp.988-991,Feb 1989.

[2] K.R.Rao and P.Yip, “Discrete cosine transform : Algorithms, Advantages, Applications,” New York :academic 1990.

[3] W. Chen ,C.H. Smith and S.C.Fralick, “A fast computation algorithm for the discerete cosine transform,” IEEE trans.commun.,vol COMM-25,pp-1004-1009,sept 1977.

[4] Z.wang, “Fast algorithm for discrete W transform and disceret fourier transform,” IEEE Trans .Acoust. , Speech signal processing, Vol ASSP -32 ,pp - 80-816,august 1984.

[5] Khurram bhukari ,George Kuzmanov, Stammatis Vassiliadis ,“DCT and IDCT implementations on different FPGA technology,” Delta University of Technology,netherland.

[6] Syafiq Al-attiq,Gillang IIham, Ardianto Satriawan , “A low power and fast DCT/IDCT architecture using proposed loeffler CSD- CSE algorithm ,” Indonesia.

[7] Mihai sima,Sorin cottofana,Jos T.J. van Eijndhoven, “An 8×8 IDCT implementation on FPGA Augmented Trimedia, ” California. U.S.A. FCCM 2001.