

Implementation of 2D DCT on Xilinx Virtex-5 M1507 FPGA

Sumitra Goswami¹ M Tech Student, BMS College of Engineering, Sri Muktsar Sahib Punjab India

Amit Kumar² Assistant Professor EC Deptt. BMS College of Engineering, Sri Muktsar Sahib Punjab India

Abstract— This paper present two algorithms for Discrete Cosine Transforms used for the compression of still images and motion video. First, background theory on discrete cosine transform (DCT) has been presented, and subsequently a detail study of DCT algorithms is presented, Based on that algorithms one architecture has been designed .It also introduced the concept of Floating point arithmetic operation which is necessary for the implementation. Further 32bit Floating point adder and multiplier is implemented. Main objective of this paper is to explore one of various architectures for optimizing any one or all of the given constraints (area, performance). Given these constraints (area, performance) our explored architecture will be a best suited as per the requirement. Above architecture will be designed and implemented in VHDL and synthesis in Xilinx tools and implementation on FPGA.

Index Terms— Discrete Cosine Transform, Field Programmable Gate Array, Floating Point Number

I. INTRODUCTION

For storage and transmission of signals we prefer a signal for any processing to be represented by minimum number of values. For the reduction of storage and bandwidth we prefer compact signal.

As most of the energy of commonly occurring signals is contained in the lower part of the spectrum, the representation of a signal is more compact in frequency-domain. Although N values can represent a signal in both the time- and frequency-domains.

Applications like desktop publishing, multimedia, teleconferencing, and high-definition Television (HDTV) are growing day by day so need for effective and standardized image compression technique is also increasing. Among the emerging standards are JPEG, for compression of still images; MPEG, for compression of motion video and CCITT H.261 (also known as Px64), for compression of video telephony and teleconferencing. A

basic technique known as the DISCRETE COSINE TRANSFORM (DCT) is employed by all of these three standards.

The DCT works by separating images into parts of differing frequencies. During a step called quantization, where part of compression actually occurs, the less important frequencies are discarded; hence, the technique of using DCT falls under “lossy image compression.” Then only the most important frequencies that remain are used to retrieve the image in the decompression process. As a result, reconstructed images contain some distortion; but these levels of distortion can be adjusted during the compression stage.

II. 1D DISCRETE COSINE TRANSFORM

The discrete cosine transform of a list of N real numbers $x(n)$, $n = 0, \dots, N-1$, is the list of length N given by

$$X(k) = \sqrt{\frac{2}{N}} C(n) \sum_{n=0}^{N-1} \cos \left[\frac{(2n+1)k\pi}{2N} \right] \quad n = 0, \dots, \dots, N-1 \quad \text{equ. (1)}$$

The list $x(n)$ can be recovered from its transform by applying the inverse cosine transform (IDCT)

$$X(n) = \sqrt{\frac{2}{N}} C(k) \sum_{k=0}^{N-1} \cos \left[\frac{(2n+1)k\pi}{2N} \right] \quad k = 0, \dots, \dots, N-1 \quad \text{equ. (2)}$$

III. 2D DISCRETE COSINE TRANSFORM

2D DCT for given sequence is given by

$$X(k_1, k_2) = C k_1 k_2 \frac{2}{\sqrt{N_1 N_2}} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) \cos \left[\frac{\pi(2n_1+1)k_1}{2N_1} \right] \cos \left[\frac{\pi(2n_2+1)k_2}{2N_2} \right] \quad \text{equ. (3)}$$

Where, Where $k_1= 1, 2, \dots, N_1 - 1$, $k_2 = 1, 2, \dots, N_2 - 1$

where F_0, F_1, F_2, F_3 are function blocks.

and

$$C_{k_i} = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } k_i = 0; i = 1, 2 \\ 1, & k_i \neq 0 \end{cases}$$

Let

$$C_{k_1}^{2n_1+1} = \cos \left[\frac{\pi(2n_1 + 1)k_1}{2N_1} \right] \quad \text{equ. (4)}$$

$$C_{k_2}^{2n_2+1} = \cos \left[\frac{\pi(2n_2 + 1)k_2}{2N_2} \right] \quad \text{equ. (5)}$$

$$\tilde{X}(k_1 k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) C_{k_1}^{2n_1+1} C_{k_2}^{2n_2+1} \quad \text{equ. (6)}$$

Using the above relations,

$$X(k_1, k_2) = C_{k_1} C_{k_2} \frac{2}{\sqrt{N_1 N_2}} \tilde{X}(k_1 k_2) \quad \text{equ. (7)}$$

Since the constant is only a scale factor.

Putting values for n_1 and n_2 , we get

$$\begin{aligned} \tilde{X}(k_1 k_2) = & C_{k_1}^1 [C_{k_2}^1 x(0,0) + C_{k_2}^2 x(0,1) + C_{k_2}^3 x(0,2) + \dots + C_{k_2}^{2N_2-1} x(0, N_2 - 1) +] \\ & + C_{k_1}^2 [C_{k_2}^1 x(1,0) + C_{k_2}^2 x(1,1) + C_{k_2}^3 x(1,2) + \dots + C_{k_2}^{2N_2-1} x(1, N_2 - 1) +] \dots \\ & + C_{k_1}^{2N_1-1} [C_{k_2}^1 x(N_1 - 1, 0) + C_{k_2}^2 x(N_1 - 1, 1) + C_{k_2}^3 x(N_1 - 1, 2) \\ & + \dots + C_{k_2}^{2N_2-1} x(N_1 - 1, N_2 - 1)] \quad \text{equ. (8)} \end{aligned}$$

IV. IMPLEMENTATION OF THE ALGORITHM

Let $N_1=N_2=4$ and $n_1=n_2=0, 1, 2, 3$

$$\begin{aligned} \tilde{X}(k_1, k_2) = & c_{k_1}^1 [x(0,0)c_{k_2}^1 + x(0,1)c_{k_2}^2 + x(0,2)c_{k_2}^3 + x(0,3)c_{k_2}^4] \\ & + c_{k_1}^2 [x(1,0)c_{k_2}^1 + x(1,1)c_{k_2}^2 + x(1,2)c_{k_2}^3 + x(1,3)c_{k_2}^4] \\ & + c_{k_1}^3 [x(2,0)c_{k_2}^1 + x(2,1)c_{k_2}^2 + x(2,2)c_{k_2}^3 + x(2,3)c_{k_2}^4] \\ & + c_{k_1}^4 [x(3,0)c_{k_2}^1 + x(3,1)c_{k_2}^2 + x(3,2)c_{k_2}^3 + x(3,3)c_{k_2}^4] \quad \text{equ. (9)} \end{aligned}$$

Put

$$\begin{aligned} c_{k_1}^1 &= a_1 & c_{k_1}^2 &= a_2 & c_{k_1}^3 &= a_3 & c_{k_1}^4 &= a_4 \\ c_{k_2}^1 &= b_1 & c_{k_2}^2 &= b_2 & c_{k_2}^3 &= b_3 & c_{k_2}^4 &= b_4 \end{aligned}$$

Therefore,

$$\begin{aligned} \tilde{X}(k_1, k_2) = & a_1 [b_1 x(0,0) + b_2 x(0,1) + b_3 x(0,2) + b_4 x(0,3)] + a_2 [b_1 x(1,0) + b_2 x(1,1) + b_3 x(1,2) + b_4 x(1,3)] \\ & + a_3 [b_1 x(2,0) + b_2 x(2,1) + b_3 x(2,2) + b_4 x(2,3)] \\ & + a_4 [b_1 x(3,0) + b_2 x(3,1) + b_3 x(3,2) + b_4 x(3,3)] \quad \text{equ. (10)} \end{aligned}$$

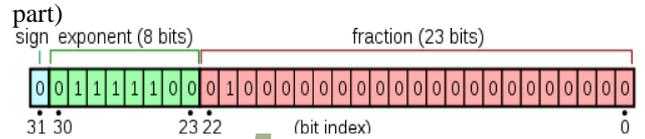
which can be written as:

$$\tilde{X}(k_1, k_2) = a_1 [F_0] + a_2 [F_1] + a_3 [F_2] + a_4 [F_3] \quad \text{equ. (11)}$$

V. ALGORITHM FOR MULTIPLIER

There are 3steps for multiplying two 32-bit binary numbers

1. Calculation of resultant sign bit
2. Calculation of resultant exponent bit
3. Calculation of resultant floating point bits(fractional)



VI. ALGORITHM FOR ADDER

There are five steps for Adder block

Step1: Compare the exponents of two numbers and calculate the absolute value of difference between the two exponents. Take the larger exponent as the tentative exponent of the result.

Step 2: Shift the significant of the number with the smaller exponent, right through a number of bit positions that is equal to the exponent difference. Two of the shifted out bits of the aligned significant are retained as guard (G) and Round (R) bits. So for p bit significant, the effective width of aligned significant must be p + 2 bits. Append a third bit, namely the sticky bit (S), at the right end of the aligned significant. The sticky bit is the logical OR of all shifted out bits.

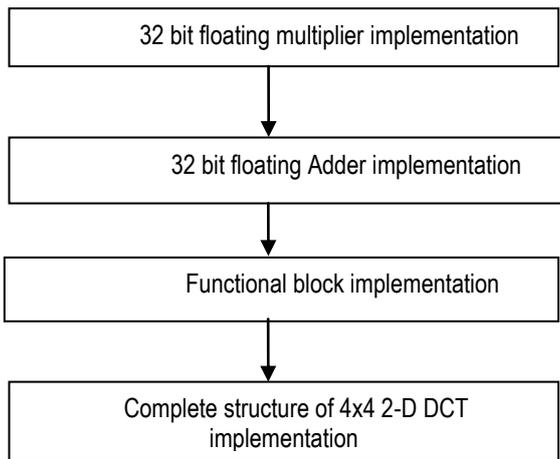
Step 3: Add the two signed-magnitudes significant using a p + 3 bit adder. Let the result of this is SUM.

Step4: Check SUM for carry out (Cout) from the MSB position during addition. Shift SUM right by one-bit position if a carry out is detected and increment the tentative exponent by 1. Evaluate exception conditions, if any.

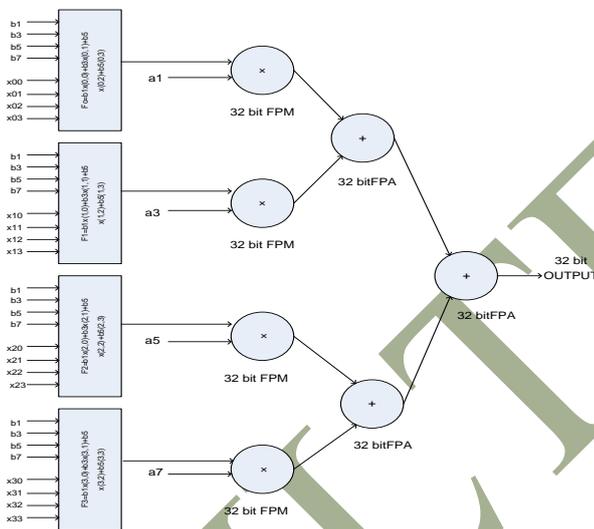
Step 5: Round the result if the logical condition R (M0+ S) is true, where M0 and R represent the pth and (p + 1) st bits from the left end of the normalized significant. New sticky bit (S) is the logical OR of all bits towards the right of the R bit. If the rounding condition is true, a 1 is added at the pth bit (from the left side) of the normalized significant. If p MSBs of the normalized significant are 1's, rounding can generate a carry-out. In that case, normalization (step 4) has to be done again.

VII. FLOW CHART OF 4x4 2D DCT

There are 4 steps in for the designing of 4x4 2-D DCT



VIII. PORPOSED ARCHITECTURE OF 2D DCT



IX. CONCLUSION

The proposed work “IMPLEMENTATION OF 2D DCT ON XILINX VIRTEX-5 ML507FPGA” has been successfully completed. The program codes for the modules have been written in VHDL. For synthesis and simulation, we have used Xilinx ISE 10.1 and Mentor Graphics ModelSim SE 6.3d tool. The modules have been synthesized for Virtex-5 ML507 FPGA (Device: XC5VFX70T package FF1136). The designed architecture work for 4x4 point DCT

X. REFERENCE

1. W. H. Chen, C. H. Smith, S. C. Fralick, "A fast computational algorithm for the discrete Cosine transform," IEEE Transactions on Communications, Vol. 25, Issue9, pp. 1004-1009, Sep. 1977.
2. Mohamed El-Sharkawy and Waleed Eshrawy, "A Fast Recursive Pruned DCT for Image Compression Applications," Proceedings of the 37th Midwest Symposium on Circuits and Systems, 3-5 August 1994.
3. David Salomon, "Data Compression," The Complete Reference, 3rd Edition, Springer-Verlag New York, Inc 2004.
4. L.V. Agostini, I.S. Silva and S. Bampi, "Pipelined Fast 2-D DCT Architecture for JPEG Image Compression," 14th Symposium on Integrated Circuits and Systems Design, 10-15 September 2001.
5. H.I. Helal, A.E. Salama and S. Mashali, "Efficient Realization of FPGA-Based Two Dimensional Discrete Cosine Transform," Proceedings of the Twelfth Annual IEEE International ASIC/SOC Conference, 15-18 September 1999.
6. M.F. Aburdene, J. Zheng and R.J. Kozick, "Computation of Discrete Cosine Transform using Clenshaw's Recurrence Formula," IEEE Signal Processing Letters, vol. 2, no. 8, August 1995.
7. Soumik Ghosh, Soujanya Venigalla, Magdy Bayoumi, "Design and implementaion of a 2D-DCT architecture using Coefficient distributed arithmetic," Proceedings of the IEEE Computer Society Annual Symposium on VLSI, pp. 162-166, 11-12 May 2005.
8. Wendi Pan, "A Fast 2-D DCT Algorithm via Distributed Arithmetic Optimization," Proceedings of the International Conference on Image Processing, 2000.
9. Syed Ali Khayam "The Discrete Cosine Transform (DCT): Theory and Application," Department of Electrical & Computer Engineering, Michigan State University, March 10, 2003.
10. Jiun-In Guo, Rei-Chin Ju, Jia-Wei Chen, "An efficient 2-D DCT/IDCT core design using cyclic convolution and adder-based realization Article Title," IEEE Trans. CAS for Video Technology, Vol. 14, Issue 4, pp. 416-428, April 2004.
11. K.S. Hema Priya, "Design of Various DCT Architecture and its Implementation using VHDL", SASTRA University, March 2011.