International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 08 | Aug 2020 www.irjet.net p-ISSN: 2395-0072
National Conference on Recent Advancements in Communication, Electronics and Signal Processing-RACES'20
Organised by: Department of ECE, Velammal Engineering College, Chennai-66

FPGA IMPLEMENTATION OF VARIABLE SIZE FIXED POINT DCT ARCHITECTURE

Dr.P.Ponsudha[1], Abirami.R[2], Kowsalya.S[3], Swathi.M[4], Tamizhmalar.V[5]

[1-5]Velammal Engineering College, Chennai.

Abstract: This paper proposes a variable size fixed point architecture for the computation of the Discrete Cosine Transform(DCT) which is preferred over Discrete Fourier Transform to remove interpixel redundancy in image compression because of its high energy compaction. This result is obtained by comparing the two different DCT architecture in order to find the most suitable one for image compression. The N=8 point DCT constructed using two N/2=4 point DCTs,is exploited to maximize the hardware reusability while maintaining constant throughput. The proposed architecture supports reconfigurability. Simulation results prove that the proposed architecture has less delay sacrificing some area. Further the proposed DCT architecture can be implemented in HEVC encoding.

Index Terms – Discrete Cosine Transform, DCT, High Efficiency Video Coding

Introduction

Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. There are several different ways in which image files can be compressed. For Internet use, the two most common compressed graphic image formats are JPEG format and GIF format. The JPEG method is more often used for photographs, while the GIF method is commonly used for line art and other images in which geometric shapes are relatively simple. Other techniques for image compression include the use of fractals and wavelets. These methods have not gained widespread acceptance for use on the Internet. However, both methods offer higher compression ratio than the JPEG or GIF methods for some types of images. Another new method that may replace the GIF format is the PNG format. Compressing an image is significantly different than compressing raw binary data. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This means that lossy compression techniques can be used in this area. A text file or program can be compressed without the introduction of errors, but only up to a certain extent. This is called lossless compression. Beyond this point, errors are introduced. In text and program files, it is crucial that compression be lossless because a single error can seriously damage the meaning of a text file, or cause a program not to run. In image compression, a small loss in quality is usually not noticeable. There is no "critical point" up to which compression works perfectly, but beyond which it becomes impossible. When there is some tolerance for loss, the compression factor can be greater than it can when there is no loss tolerance. For this reason, graphic images can be compressed more than text files. One of the important aspects of image storage is its efficient compression. For example, an image, 1024 pixels × 1024 pixels × 24 bit, without compression, would require 3 MB of storage and 7 minutes for transmission, utilizing a high speed, 64Kbits/sec, ISDN line. If the image is compressed at a 10:1 compression ratio, the storage requirement is reduced to 300 Kb and the transmission time drops to under 6 seconds. Seven 1 MB images can be compressed and transferred to a floppy disk in less time than it takes to send one of the original files, uncompressed, over an AppleTalk network. In a distributed environment, large image files remain a major bottleneck within systems. Compression is an important component of the solutions available for creating sizes of manageable and transmittable dimensions. Increasing the bandwidth is another method, but the cost sometimes makes this a less attractive solution. A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task is to find less correlated representation of the image. Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is the removal of redundant data. From a mathematical viewpoint, this amounts to transforming a 2-D pixel array into a statistically uncorrelated data set. The transformation is applied prior to storage and transmission of the image. The compressed image is decompressed at some later time, to reconstruct the original image or an approximation to it. Two fundamental components of

compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image / video). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS).In general, three types of redundancy can be identified:(i) Spatial Redundancy or correlation between neighbouring pixel values.(ii) Spectral Redundancy or correlation between different color planes or spectral bands.(iii) Temporal Redundancy or correlation between adjacent frames in a sequences of images (in video applications).Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible. Two ways of classifying compression techniques are mentioned here (i)Lossy compression(ii)Lossless compression. In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However, lossless compression can only achieve a modest amount of compression. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. However, lossy schemes are capable of achieving higher compression. Under normal viewing conditions, no visible loss is perceived (visually lossless).The information loss in lossy coding comes from quantization of the data. Quantization can be described as the process of sorting the data into different bits and representing each bit with a value. The value selected to represent a bit is called the reconstruction value. Every item in a bit has the same reconstruction value, which leads to information loss (unless the quantization is so fine that every item gets its own bit).In predictive coding, information already sent or available is used to predict future values, and the difference is coded. Since this is done in the image or spatial domain, it is relatively simple to implement and is readily adapted to local image characteristics. Differential Pulse Code Modulation (DPCM) is one particular example of predictive coding. Transform coding, on the other hand, first transforms the image from its spatial domain representation to a different type of representation using some well-known transform and then codes, the transformed values (coefficients). This method provides greater data compression compared to predictive methods, although at the expense of greater computation. A typical lossy image compression system is shown in figure.1. It consists of three closely connected components namely,(i)Source Encoder(ii) Quantizer (iii)Entropy Encoder. Compression is accomplished by applying a linear transform to decorrelate the image data, quantizing the resulting transform coefficients, and entropy coding the quantized values.Over the years, a variety of linear transforms have been developed which include Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and many more, each with its own advantages and disadvantages.

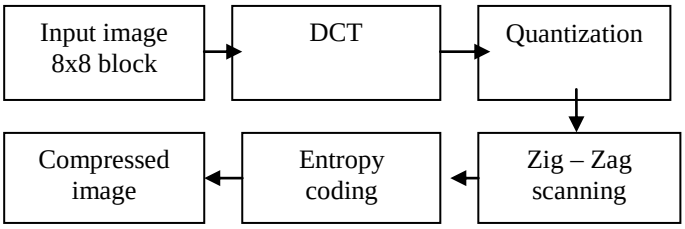


Figure 1. Overall image compression process

A DCT works on a capacity at a limited number of discrete information focuses. The conspicuous qualification between DCT and a DFT is that the previous uses just cosine capacities, while the last uses the two cosines and sines (as intricate exponentials). The guideline favorable position of change is the evacuation of excess between neighboring pixels. This prompts uncorrelated change coefficients which can be encoded independently. The properties of DCT are(i) Decorrelation: Removal of excess between neighboring pixels. (ii)Energy Compaction: Efficiency of a change plan can be legitimately menected images, asured by its capacity to pack input information into as not many coefficients as could be expected under the circumstances. This permits the quantizer to dispose of coefficients with moderately little amplitudes without presenting visual bending in the recreated image.(iii)Separability: The two dimensional DCT can be isolated as two measurements and can be registered as two 1-D DCTs. The focal points of DCT are, DCT as a sinusoidal change all the more intently surmised the data pressing capacity. The most change coding frameworks depend on the DCT, which gives a decent trade off between data pressing capacity and computational intricacy. Actually, the properties of the DCT have end up being of such down to earth esteem that the DCT has become a worldwide changes for change coding frameworks. Contrasted with the other info autonomous changes, it has the upsides of having been actualized in a solitary incorporated circuit, pressing the most data into the least coefficients, and limiting the square like appearance, called blocking ancient rarity, that outcomes when the limits between sub-images become noticeable. This property is especially important.

Method

Initially, an input image is considered. For compressing the image, it has to be subdivided into smaller sub blocks. In most applications, images are subdivided so that the correlation (redundancy) between adjacent sub images is reduced to some acceptable level and so that n is an

integer power of 2, whereas before, n is the sub image dimension. The latter condition simplifies the computation of the sub image transforms. In general, both the level of computational complexity and compression increases as the sub image size increases. The most popular sub image sizes are 8x8 and 16x16. Here it consider the sub-division of images into 8x8 size to ease the process. Also the frequency transformations like DCT is good at compressing rather smooth areas with low frequency content, but quite bad at compressing high frequency content areas. Any matrices of sizes greater than 8X8 are harder to do mathematical operations or not supported by hardware or take longer time. Those are designed so that they can be implemented using parallel hardware. Each block is independent, and can be calculated on a different computing node, or shared out to as many nodes. So, parallel computing on that level was very unusual for JPEG standard. Any matrices less than 8X8 are enough to continue along with the pipeline. And it will consume more coefficients, so we go for dividing 8X8 sub-blocks method. It uses lesser number of bits as compared to the original representation. The subdivided blocks are generally applied directly to the next corresponding steps, that is, the transformation, quantization and the encoding. But in our project, we present a different approach after performing the sub division. We introduce a new method based on computing the input values using DCT. After performing this comparative input method only, the other steps of the compression process are carried out.There are generally 2 methods for computing the 2-D DCT.(i)Direct 2-D computation(ii)Decomposition into two 1-D DCTs.We are adapting the second approach to compute the 2-D DCT. The row transformation is initially applied to obtain a 1-D output and then applying it the next time yields the 2-D output .Initially, as DCT performs the frequency transformation in the form of linear mathematical operations, the DCT equations are considered before transforming into the form of algorithm. The 1-dimensional DCT equations are given by

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

for u=0,1,2…N-1

$$D(u) = \sqrt{\frac{1}{N}} \quad \text{for } u = 0$$

$$D(u) = \sqrt{\frac{2}{N}} \quad \text{for } u = 1,2,3\ldots\ldots,(N-1)$$

Here f(x) is the 1-D row input. The cosine term is the orthonormal function. F(u) gives the 1-D DCT output. D(u) is the normalizing factor.To implement the DCT, we are using the modified Lee algorithm. In this algorithm, The DCT computation is decomposed into 3 steps and mathematical simplifications are applied. This results in the 1-D DCT output. Also the implementation is done using the basic Chen's algorithm and comparison is done to show that the modified Lee algorithm reduces the complexity by 30%.

Existing DCT architecture

Loeffler DCT uses only 11 multiplications and 29 additions for a 8-point vector, achieving the multiplication lower bound, without an increase in addition operations. One of its variations is adopted by the Independent JPEG Group in a popular image compression algorithm JPEG implementation.

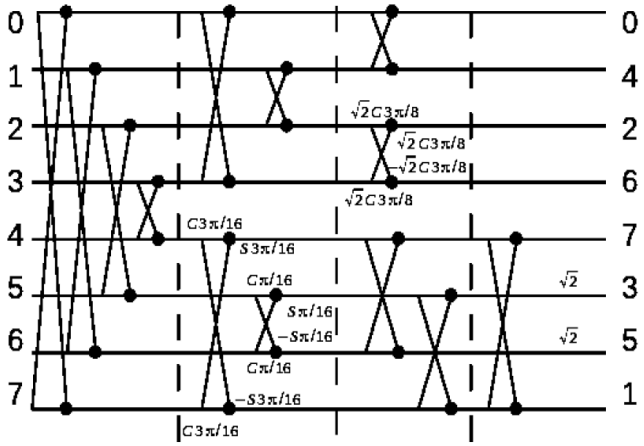
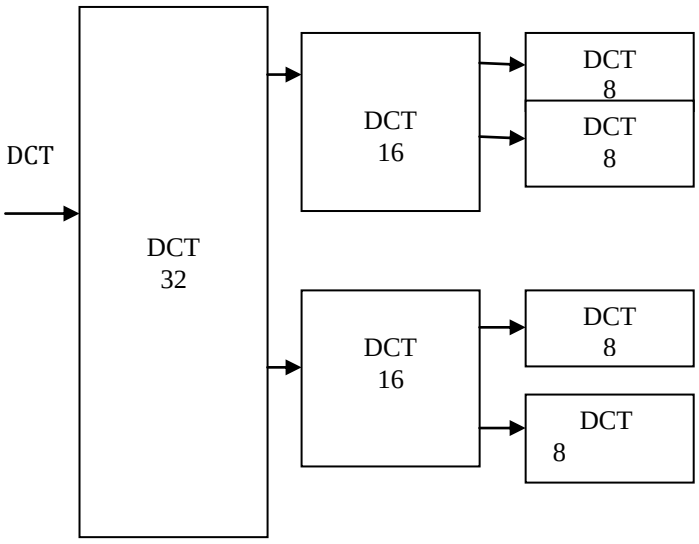


Figure 2 .Signal flow graph of Existing DCT algorithm

The Loeffler Discrete Cosine Transform computation algorithm is sectioned into 4 stages, which are executed in series as in figure 2. Note that this factorization requires a uniform scaling factor at the end of the flow graph to obtain the true DCT coefficients.

Proposed Variable point DCT using Fixed Point DCT

The 64 points modified DCT can be easily reduced to 32 points DCT, which requires 32 x 32 multiplications. The verification shows that it eliminates about 50 per cent of redundancy. The first and last 16 coefficients in the array are identical but with inverted sign. The same occurs with the following 32 coefficients.



$$\begin{pmatrix} Y(0,:) \\ Y(2,:) \\ Y(4,:) \\ Y(6,:) \end{pmatrix} = \begin{matrix} a & a & a & a \\ c & f & -f & -c \\ a & -a & -a & a \\ f & -c & c & -f \end{matrix} \begin{pmatrix} X(0,:)+X(7:,) \\ X(1,:)+X(6:,) \\ X(2,:)+X(5:,) \\ X(3,:)+X(4:,) \end{pmatrix}$$

Figure 3. Variable size fixed point DCT

A 32 points DCT: was implemented using the fast algorithm proposed , Where a N point DCT is divided in two N/2 Point DCT. That operation delivers an even and odd part of N/2 DCTs denoted as C(i) and D(i) respectively.In figure a division scheme of the 32 DCT considering an even and odd part using a 16 DCT is shown. Each one of these modules is divided into an even and odd part with 8-Points DCT respectively. It is important to mention that the odd part must be multiplied by a scalar factor before executing the respective subdivision in even and odd part. Therefore the 8 DCT is divided itself into an even and odd part using a 4-Points DCT.

8 point DCT using 4 point DCT

The algorithm proposed computes the 2-D DCT in terms of two 1-D DCTs. The computation is done as said previously – computing 1-D DCT, transposing, computing 2-D DCT.For 2-D DCT, the 8x8 transformation matrix corresponding to the 8x8 basis function is given by

$$A = \begin{pmatrix} d & d & d & d & d & d & d & d \\ a & c & e & g & -g & -e & -c & -a \\ b & f & -f & -b & -b & -f & f & b \\ c & -g & -a & -e & e & a & g & -c \\ d & -d & -d & d & d & -d & -d & d \\ e & -a & g & c & -c & -g & a & -e \\ f & -b & b & -f & -f & b & -b & f \\ g & -e & c & -a & a & -c & e & -g \end{pmatrix}$$

where $b = C_1$, $c=C_2$, $d=C_3$, $a= C_4$, $e=C_5$, $f=C_6$, $g=C_7$;

$$C_i = 0.5 \times \cos\left(\frac{i\pi}{16}\right)$$

But in Proposed algorithm, the 8x8 transformation matrix is decomposed into two 4x4 matrices. This is done by considering the input values which need to be multiplied with common coefficients (in the transformation matrix). After decomposing the two 4x4 transformation matrices obtained are,

$$\begin{pmatrix} Y(1,:) \\ Y(3,:) \\ Y(5,:) \\ Y(7,:) \end{pmatrix} = \begin{matrix} b & d & e & g \\ d & -g & -b & -c \\ e & -b & g & d \\ g & -e & d & -b \end{matrix} \begin{pmatrix} X(0,:)-X(7:,) \\ X(1,:)-X(6:,) \\ X(2,:)-X(5:,) \\ X(3,:)-X(4,:) \end{pmatrix}$$

The X corresponds to the 1-D input values and Y corresponds to the 1-D output values. The equations are implemented by using the signal flow graph shown in Figure 4. The number of computations involved are (3N/2)(log2 N - 1) +2 real additions and (N log2 (N)) - 3N/2 -t 4 real multiplications. Hence for N=8, it requires 16 multiplications and 26 additions.

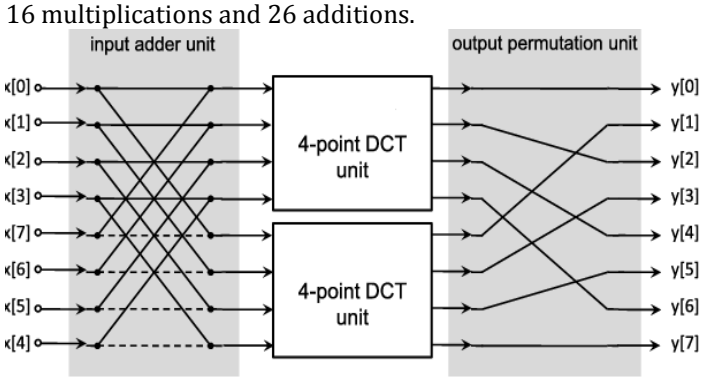


Figure 4. Signal flow graph of proposed Fast algorithm

Results

Simulation is performed using modelsim. 8 inputs are provided each of 8 bits. Therefore 8 outputs are obtained each of 8 bits. Simulation results of the proposed architecture is shown in figure

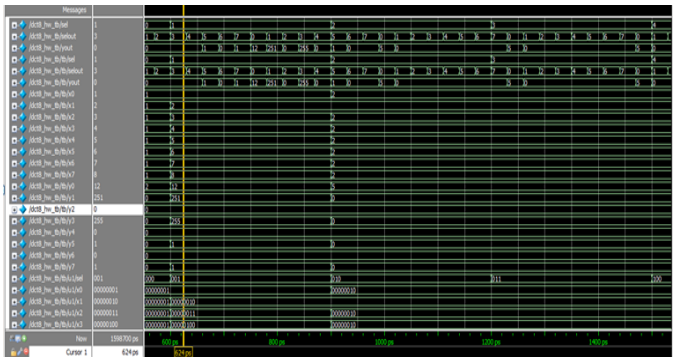


Figure 5.Simulation results of proposed DCT architecture

Synthesis of 8 point DCT using 4 point DCT

The proposed architecture has been described in verilog and synthesized using Xilinx Spartan-6.The delay and area of proposed architecture in comparison with the existing N=8 fixed point DCT architecture has been reported in table I.

As there always exists a tradeoff between area and delay, in the proposed architecture the delay decreases with the increase in area. The area consumed by the proposed DCT architecture is of 7% whereas the area consumed by the existing architecture is of 3%. The delay of the proposed architecture is of 20.019ns whereas the delay is 31.135ns.Therefore the proposed architecture finds its use in applications that require less delay.

s.no	Parameters	Existing system	Proposed system
1	Number of slices	170 out of 4656	364 out of 4656
2	Number of 4 input LUTs	318 out of 9312	676 out of 9312
3	Number of IO's	136	320
4	Number of bonded IOB's	136 out of 190	320 out of 190
5	Number of MULT18X18SIOs	17 out of 20	20 out of 20
6	Maximum combinational path delay	31.135ns	20.019ns

Table I: Comparison between existing (N=8) fixed point architecture and (N/2=4) variable size(N=8)fixed point DCT architecture.

CONCLUSION

In this paper, an variable-size 1D-DCT architecture has been proposed. A comparison among the existing DCT architecture has been presented and exploited to identify the one which minimizes the delay. The proposed architecture can be implemented where reconfigurability is required. It can also be implemented in High efficiency video coding encoding.

REFERENCES

[1] M. BUdagavi, A. Fuldseth, G.Bjontegaard, V. Sze, and M. Sadafale, "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard," IEEE J. Sel. Topics Signal Process, vol. 7, no.6, pp. 1029-1041, Dec 2013.

[2] M. Naccari, A. Gabriellini, M. Mrak, S. G. Blasi, I. Zupancic, and E. Izquierdo,HEVC Coding Optimisation for Ultra High Definition Television Services," in Picture Coding Symposium, May 2015, pp. 20-24.

[3] M. Masera, L.R. Fiorentin, E. Masala, G. Masera, and M. Martina, "Analysis of HEVC Transform Throughput Requirements for Hardware Implementations," Elsevier Signal Processing: Image Communication, vol.57, pp. 17-182, 2017.

[4]A. Ahmed, M.U. Shahid, and A. Rehman, "N point DCT VlSI Architecture for Emerging HEVC Standard," VlSI Design, vol. 2012, Article 752024, pp. 1-13,2012.

[5]M. Masera, M. Martina, and G. Masera, "Adaptive Approximated DCT Architectures for HEVC," IEEE Trans. Circuits Syst. Video Technol., to be published.

[6]M. Budagavi and V. Sze, "Unified Forward + Inverse Transform Architecture for HEVC," in Proc. 19th IEEE Int. Conf. Image Processing, Sept 2012, pp. 209-212

[7]P. Meher, S. Y. Park, B. Mohanty, K.S. Lim, and C. Yeo, "Efficient Integer DCT Architectures for HEVC,"IEEE Trans. Circuits. Syst. Video Technol, vol.24, no.1,pp. 168-178, Jan 2014.

[8]W.Zhao, T. Onoye, and T. Song, "High-Performance Multiplierless Transform Architecture for HEVC," in 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013), May 2013, pp. 1668-1671.