International Journal of Research
Special Issue in UGC Approved Journal
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 11
September 2017

Improved Dct for Image and Video Compression

DR.T.SRINIVAS

Research Scholor(15412335)Dept of ECE,Shri Jagadishprasad Jabharmal Tibrewala University , Jhunjhunu , Rajastan , tspdpl@gmail.com

Abstract— Video processing systems such as HEVC requiring low energy consumption needed for the multimedia market has lead to extensive development in fast algorithms for the efficient approximation of 2-D DCT transforms. The DCT is employed in a multitude of compression standards due to its remarkable energy compaction properties. Multiplier-free approximate DCT Transforms have been proposed that offer superior compression performance at very low circuit complexity. Such approximations can be realized in digital VLSI hardware using additions and Subtractions only, leading to significant reductions in chip area and power consumption compared to conventional DCTs and integer transforms. In this paper, we introduce a novel 8-point DCT approximation that requires only 14 addition operations and no multiplications. The proposed DCT approximation is a candidate for reconfigurable video standards such as HEVC. The proposed transform and several other DCT approximations are mapped to systolic-array digital architectures and physically realized as digital prototype circuits using FPGA Spartan 3 and it are implemented by verilog language.

Keywords- compression , DCT, transform , addition, VLSI, architecture ,approximation

I . INTRODUCTION

A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient (as explained below, fewer are needed to approximate a typical signal), whereas for differential equations the cosines express a particular choice of boundary conditions.In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT"; its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and odd functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of overlapping data.

A.DCT versus KLT/DFT:

At this point it is important to mention the superiority of DCT over other image transforms. More specifically, we compare DCT with two linear transforms:

1) The Karhunen - Loeve Transform (KLT);
2) Discrete Fourier Transform (DFT).

The KLT is a linear transform where the basis functions are taken from the statistical properties of the image data, and can thus be adaptive. It is optimal in the sense of energy compaction, i.e., it places as much energy as possible in as few coefficients as possible. However, the KLT transformation kernel is generally not separable, and thus the full matrix multiplication must be performed. In other words, KLT is data dependent and, therefore, without a fast (FFT-like) pre computation transform. Derivation of the respective basis for each image sub-block requires unreasonable computational resources. Although, some fast KLT algorithms have been suggested ([12], [13]), nevertheless the overall complexity of KLT is significantly higher than the respective DCT and DFT algorithms.

B. VLSI

Very-large-scale integration (VLSI) is the process of creating an integrated circuit by combining thousands of transistors into a single chip. VLSI began in the 1970s when complex semiconductor and communication technolo gies were being developed. The microprocessor is a

VLSI device. Before the introduction of VLSI technology most ICs had a limited set of functions they could perform. An electronic circuit might consist of a CPU, ROM, RAM and other glue logic .Current designs, unlike the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM (static random-access memory) cell, are still designed by hand to ensure the highest efficiency.

C. VLSI Design Styles

Several design styles can be considered for chip implementation of specified algorithms or logic functions. Each design style has its own merits and shortcomings, and thus a proper choice has to be made by designers in order to provide the functionality at low cost.

D. Verilog HDL

The Verilog hardware description language, usually just called Verilog was designed and first implemented by philmoorby at gateway Design automation in 1984 and 1985. It was first used in 1985 and was extended substantially through 1987.The implementation was the verilog simulator sold by gateway. The first major extension was Verilog-XL, which added a few features and implemented the infamous algorithm. This occurred in 1986 and marked the beginning of verilog's growth period. In 1988 synopsis delivered the first logic synthesizer, which used verilog as an input language. This was a major recent as now the top down design methodology could actually be used effectively. The design could be done at the "register transfer level" and then Synopsis design compiler could translate that into gates. With this event, the use of verilog increased dramatically.

Recent years have experienced a significant demand for high dynamic range systems that operate at high resolutions. In particular, high-quality digital video in multimedia devices and video-over-Internet protocol networks are prominent areas where such requirements are evident. Often hardware capable of significant throughput is necessary; as well as allowable area-time complexity. In this context, the discrete cosine transform (DCT) is an essential mathematical tool in both image and video coding [8]. Indeed, the DCT was demonstrated to provide good energy compaction for natural images, which can be described by first-order Markov signals.

Several efficient algorithms were developed and a noticeable literature is available. Although fast algorithm scan significantly reduce the computational complexity of computing the DCT, floating-point operations are still required. Despite their accuracy, floating-point operations are expensive in terms of circuitry complexity and power consumption. There fore, minimizing the number of floating-point operations is a sought property in a fast algorithm. One way of circumventing this issue is by means of approximate transforms.

The aim of this paper is two-fold. First, we introduce a new DCT approximation that possesses an extremely low arithmetic complexity, *requiring only 14 additions*. This novel transform was obtained by means of solving a tailored optimization problem aiming at minimizing the transform computational cost. Second, we propose hardware implementations for several2-D 8-point approximate DCT.

The approximate DCT methods under consideration are (i) the proposed transform for DCT; (ii) the 2008 Bouguezel-Ahmad-Swamy (BAS) DCT approximation(iii) the Cintra-Bayer (CB) approximate DCT based on the rounding-off function. All introduced implementations are sought to be fully parallel time-multiplexed 2-D architectures for 8 ×8 data blocks. Additionally, the proposed designsare based on successive calls of 1-D architectures taking advantage of the separability property of the 2-D DCT kernel.

II . PROPOSED TRANSFORM

The aim at deriving a novel low-complexity approximate DCT. For such end, we propose a search over the 8× 8 matrixspace in order to find candidate matrices that possess low computationcost. Let us define the cost of a transformation matrixas the number of arithmetic operations required for its computation.One way to guarantee good candidates is to restrict the
search to matrices whose entries do not require multiplicationoperations. Thus we have the following optimization problem:

$$T_P = \arg\min_{T} \text{cost(T)}$$

where$T_P$is the sought matrix andcost(T) returns the arithmeticcomplexity ofT .

Additionally, the following constraints were adopted:

1)Elements of matrix T must be in$\{0, \pm 1, \pm 2\}$ to ensurethat resulting multiplicative complexity is null;

2) We impose the following form for matrix T:

$T =$

$$\begin{matrix} a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 \\ a_0 & a_2 & a_4 & a_6 & -a_6 & -a_4 & -a_2 & -a_0 \\ a_1 & a_5 & -a_5 & -a_1 & -a_1 & -a_5 & a_5 & a_1 \\ a_2 & -a_6 & -a_0 & -a_4 & a_4 & a_0 & a_6 & -a_2 \\ a_3 & -a_3 & -a_3 & a_3 & a_3 & -a_3 & -a_3 & a_3 \\ a_4 & -a_0 & a_6 & a_2 & -a_2 & -a_6 & a_0 & -a_4 \\ a_5 & -a_1 & a_1 & -a_5 & -a_5 & a_1 & -a_1 & a_5 \\ a_6 & -a_4 & a_2 & -a_0 & a_0 & -a_2 & a_4 & -a_6 \end{matrix} [\ ]$$

where $a_i = \{0,1,2\}$, for i = 0,1,2 …,6 ;
3) All rows of are non-null;
4) Matrix $T.T^T$ must be a diagonal matrix to ensure orthogonalityof the resulting approximation.Constraint 2) is required to preserve the DCT-like matrixstructure.

Above optimization problem is algebraically intractable Therefore we resorted to exhaustive computational search .As a result, eight candidate matrices were found, including the transform matrix proposed. Among these minimal cost matrices, we separated the matrix that presents the best performance in terms of image quality of compressed images according the JPEG-like technique employed.

An important parameter in the image compression routine is the number of retained coefficients in the transform domain. In several applications ,Retaining a very small number of coefficients is also common for other image block sizes. The number of retained coefficients equal to 10,

C_P

$$= D_P . \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Where$D_P =$
$diag\left({}^{1}/_{\sqrt{8}}, {}^{1}/_{\sqrt{2}}, {}^{1}/_{2}, {}^{1}/_{\sqrt{2}}, {}^{1}/_{\sqrt{8}}, {}^{1}/_{\sqrt{2}}, {}^{1}/_{2}, {}^{1}/_{\sqrt{2}}\right)$
.Matrix$\boldsymbol{T_p}$ hasentries in $\{0, \pm 1\}$and it can be given a sparse factorizationaccording to:$T_P = P_4 . A_{12} . A_{11} . A_1$, where

$$A_{11} = diag\left(\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \mathrm{I}_4\right) \qquad (3.11)$$

$$A_{12} = diag\left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, -1, I_5\right)$$

and P_4 is the permutation (1)(2 5 6 8 4 3 7).

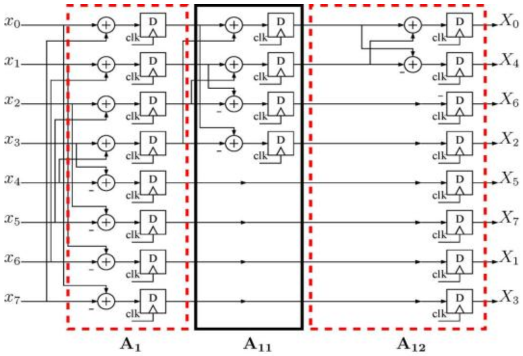


Fig1.Proposed Transform for DCT

This diagram represents the digital architecture of proposed transform matrix This matrix contain only 14 addition operations. This is mainly used for high speed applications.

A. Arithmetic Complexity

The arithmetic complexity as figure of merit for estimating the computational complexity. The arithmetic complexity consists of the number of elementary arithmetic operations (additions/subtractions, multiplications/divisions, and bit shift operations) required to compute a given transformation. In other words, in all cases, we focus our attention to the low-complexity matrices:T_1,T_2 , and the proposed matrixT_P.

For instance, in the context of image and video compression, the complexity of the diagonal matrix can be absorbed into the quantization step ;therefore the diagonal matrix does not contribute towards an increase of the arithmetic complexity , .Because all considered DCT approximations have null multiplicative complexity, we resort to comparing them in terms of their arithmetic complexity assessed by the number of additions/subtractions and bit-shift operations. Table I displays the obtained complexities. We also include the complexity of the *exact* DCT calculated (i) directly from definition and (ii) according to Arai fast algorithm for the exact DCT . We derived a fast algorithm for the proposed transform, employing *only 14 additions*. This is the same very low-complexity exhibited by the Modified CB-2011 approximation . To the best of our knowledge these are DCT approximations offering the lowest arithmetic complexity in literature.

B. Comparative Performance

Method	Multiplication	Addition	Shift	Total
Exact DCT	64	56	0	120
BAS 2008	0	18	2	20
CB 2011	0	22	0	22
Proposed Transform	0	14	0	14

Table.1. Arithmetic complex analysis

C. Digital Architectures and Realizations

In this section we propose architectures for the detailed 1-Dand 2-D approximate 8-point DCT. This section explores the hardware utilization of the discussed algorithms while providing a comparison with the proposed novel DCT approximation algorithm and its fast algorithm realization. Our objective here is to offer digital realizations together with measured or simulated metrics of hardware resources so that better decisions on the choice of a particular fast algorithm and its implementation can be reached.

D. Proposed Architecture

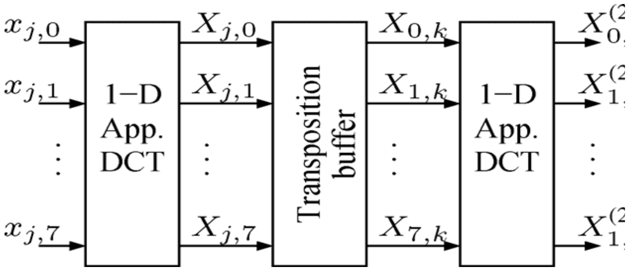


Fig. 2.Two-dimensional approximate DCT

This is propose digital computer architectures that are custom designed for the real-time implementation of the fast algorithm. The proposed architectures employs two parallel realizations of DCT approximation blocks, as shown in Fig. 3.1.

The 1-D approximate DCT blocks implement a particular fast algorithm chosen from the collection described earlier in the paper. The first instantiation of the DCT block furnishes a row-wise transform computation of the input image, while the second implementation furnishes a column-wise transformation of the intermediate result. The row- and column-wise transforms can be any of the DCT approximations detailed in the paper. In other words, there is no restriction for both row- and column-wise transforms to be the same. However, for simplicity, Itadopted identical transforms for both steps.

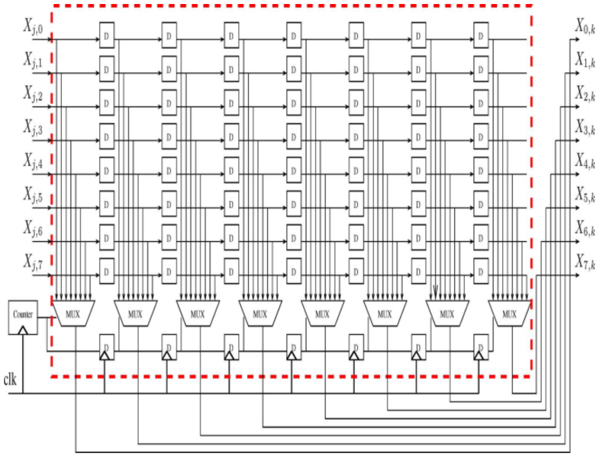


Fig.2. Proposed transposition buffer block

Between the approximate DCT blocks a real-time row-parallel transposition buffer circuit is required. Such block ensures data ordering for converting the row-transformed data from the first DCT approximation circuit to a transposed format as required by the column transform circuit. The transposition buffer block is detailed in Fig. 3.2.The circuitry sections associated to the constituent matrices of the discussed factorizations are emphasized in the figures in bold or dashed boxes.

E.SOFTWARE REQUIREMENT

- Verification Tool
    - Modelsim 6.4c
- Synthesis Tool
    - Xilinx ISE 9.1

F.MODELSIM

ModelSim SE - High Performance Simulation and Debug. ModelSim SE is our UNIX, Linux, and Windows-based simulation and debug environment, combining high performance with the most powerful and intuitive GUI in the industry.

ModelSim provides seamless, scalable performance and capabilities, supports very fast time-tenet-simulation turnarounds while maintaining high performance with its new black box use model, known as bbox. With bbox, non-changing elements can be compiled and optimized once and reused.An intelligently engineered graphical user interface (GUI) efficiently displays design data for analysis and debug. The default configuration of windows and information is designed to meet the needs of most users. However, the flexibility of the ModelSim SE GUI allows users to easily customize it to their preferences. The result is a feature-rich GUI that is easy to use and quickly mastered. The ModelSim advanced code coverage capabilities deliver high performance with ease of use.

G.SYNTHESIS TOOL:XILINX ISE

For two-and-a-half decades, Xilinx has been at the forefront of the programmable logic revolution, with the invention and continued migration of FPGA platform technology. During that time, the role of the FPGA has evolved from a vehicle for prototyping and glue-logic to a highly flexible alternative to ASICs and ASSPs for a host of applications and markets. Today, Xilinx® FPGAs have become strategically essential to world-class system companies that are hoping to survive and compete in these times of extreme global economic instability, turning what was once the programmable revolution into the “programmable imperative” for both Xilinx and our customers.

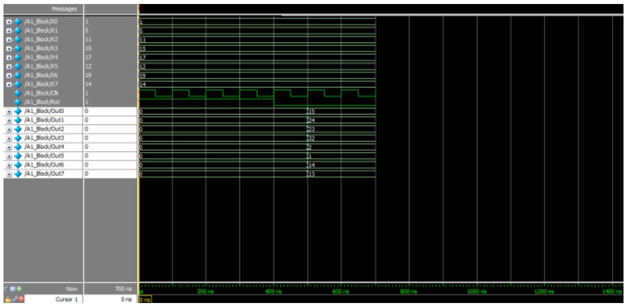
III.CODING IMPLEMENTATION:

Top module Proposed 1D DCT-2014:

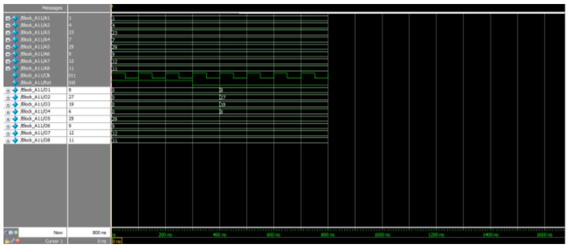
```
Proposed_1D(X0,X1,X2,X3,X4,X5,X6,X7,Clk,Rst,O
ut0,Out4,Out6,Out2,Out5,Out7,Out1,Out3;

  input [7:0]X0,X1,X2,X3,X4,X5,X6,X7;

inputClk,Rst;

output
[7:0]Out0,Out4,Out6,Out2,Out5,Out7,Out1,Out3;

  wire
[7:0]A1,A2,A3,A4,A5,A6,A7,A8,B0,B1,B2,B3,B4,B
5,B6,B7;

A1_Block M1(
        .X0(X0),
        .X1(X1),
        .X2(X2),
        .X3(X3),
        .X4(X4),
        .X5(X5),
        .X6(X6),
        .X7(X7),
        .Clk(Clk),
        .Rst(Rst),
        .Out0(A1),
        .Out1(A2),
        .Out2(A3),
        .Out3(A4),
        .Out4(A5),
        .Out5(A6),
        .Out6(A7),
        .Out7(A8)
        );

A11_Block M2(
        .A1(A1),
        .A2(A2),
        .A3(A3),
        .A4(A4),
        .A5(A5),
        .A6(A6),
        .A7(A7),
        .A8(A8),
        .Clk(Clk),
        .Rst(Rst),
        .O1(B0),
        .O2(B1),
        .O3(B2),
        .O4(B3),
        .O5(B4),
        .O6(B5),
        .O7(B6),
        .O8(B7)
        );

A12_Block M3(
         .B0(B0),
         .B1(B1),
         .B2(B2),
         .B3(B3),
         .B4(B4),
         .B5(B5),
         .B6(B6),
         .B7(B7),
         .Clk(Clk),
         .Rst(Rst),
         .O0(Out0),
.O1(Out4),
.O2(Out6),
.O3(Out2),
.O4(Out5),
.O5(Out7),
.O6(Out1),
.O7(Out3)
);
```

Proposed 1D DCT-2014:

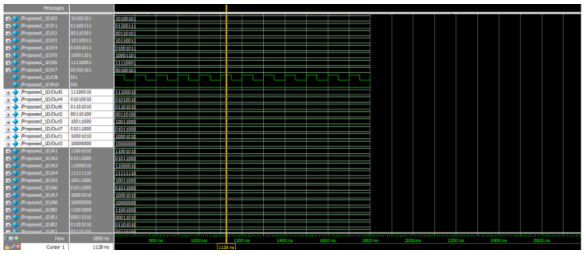
A1Block:



A11 Block:



TOPMODULE:



IV . CONCLUSION

In this we proposed a novel low power 8 point DCT approximation that require only 14 addition operations to computations and hardware implementations for the proposed transform and several other prominent approximate DCT methods including the designs by Bouguezel-Ahmad-Swamy. We obtained that all considered approximate transforms perform very close to the ideal DCT. However ,the modified CB-2011approximation and the proposed transform possess lower computational complexity and are faster than all other approximations under consideration. In terms of image compression ,the proposed transform could outperform the modified CB-2011algorithm . Hence the new proposed transform is the best approximation for the DCT in terms of computational complexity and speed among the approximate transform examined .Introduced implementations address both1-D and 2-D approximate DCT .All the approximations were digitally implemented using both Xilinx FPGA tools and CMOS45 nm ASIC technology. The speeds of operation were much greater using the CMOS technology for the same function word size .Therefore ,the proposed architectures are suitable for image and video processing ,being candidates for improvements in several standards including the HEVC.

V. REFERENCES

[1].A. Madanayake, R. J. Cintra, D. Onen, proposed “A Row-parallel 8 *8 2-D Dct Architecture using Algebraic Integer-Based Exact Computation”. 2008,

[2]. Lengwehasatit, K.proposed “Scalable Variable Complexity Approximate Forward Dct”. 2009,

[3].Bouguezel, S., Ahmad, M.O., Swamy, M.N.S. proposed “A Low-Complexity Parametric Transform for Image Compression”. 2010,

[4].M Bayer, R J Cintra, A Edirisuriya and A Madanayakeproposed “A Digital Hardware Fast Algorithm and Fpga-Based prototype for A Novel 16-Point Approximate Dct for Image Compression Applications”. 2011

[5].Brahimi, N., Bouguezel, S.proposed “An Efficient Fast Integer Dct Transform for Images Compression with 16 additions Only”. 2013

[6].,S. Bouguezel, M.O. Ahmad, M.N.S. Swamy proposed “Low-Complexity 8×8 Transform for Image Compression” 2011

[7].R. J. Cintra, F. M. Bayer, proposed “A Dct approximation for Image Compression”. 2011

F. M. Bayer ,R. J. Cintra ,A. Madanayake, U. S. Potluri,propoed “Multiplier Less approximate 4-Point DctVlsi Architectures for Transform Block Coding”. 2013

[8].Wen-hsiungchen, C. Harrison smith, and s. C. Fralickproposed “A Fast computational algorithm for The Discrete Cosine Transform” 2010

[9].Huei-Yung Lin and Wei-ZheChangproposed"High Dynamic Range Imaging for Stereoscopic scene Representation". 2007

[10].Jie Liang, proposed "Fast Multiplierless Approximations of The Dct with The Lifting Scheme". 2009

[11].S. Marsi, G. Impoco, S. C. A. Ukovich, and G. Ramponiproposed "Video Enhancement and Dynamic Range Control of HDR Sequences for Automotive applications". 2010

[12].S.BhawnaGautam, R.Baliarsinghproposed "Image Compression using Discrete Cosine Transform & Discrete Wavelet Transform". 2008

[13].SarpErtürk proposed "Multiplication-Free One-Bit Transform for Low-Complexity Block-Based Motion Estimation". 2007