

Design of 8-Point DCT Architecture with 12 Addition Only for Image Compression

Dhivyashri K B¹, Saravanan S², Sathishkumar M³, Sonia S⁴

UG Student,

^{1,2,3,4}Electronics & Communication Engg., Aksheyaa College of Engg., Madhuranthagam, , India

Saravanan R⁵

Project Guide,

⁵Electronics & Communication Engg., Aksheyaa College of Engg., Madhuranthagam, , India

Vinothkumar S⁶

Project Co-ordinator

⁶Electronics & Communication Engg., Aksheyaa College of Engg., Madhuranthagam, , India

Abstract-The DCT is engaged in multitude of compression standards due to the remarkable energy compaction properties. Approximate DCT transform have been proposed that offer superior compression performance at very low circuit complexity based on addition and subtraction without any multiplication and shifting operations. In this paper, 8-point approximate DCT is introduced with 12 addition only for image compression. Besides A1CSA is introduced for addition and subtraction operations and transpose memory is also introduced for computing approximate 2D-DCT. A1CSA and transpose memory further reduce the computational complexity, power consumption, chip area and critical time delay and achieves a performance in image compression that is comparable to that of the existing approximated DCT. The proposed approximate DCT is implemented on Xilinx Spartan 3E field programmable gate array (FPGA) device and synthesized with standard cell library.

Index Terms-Approximate 2D-DCT, image compression, critical time delay, FPGA.

I. INTRODUCTION

Discrete cosine transform (DCT) [1] has become one of the basic tools in signal and image processing; the popularity of which is mainly due to its good energy compaction properties. Indeed, DCT has found applications in many image compression standard such as JPEG [2]. During the JPEG process, an image is divided into several 8×8 blocks and then the two-dimensional discrete cosine transform (2D-DCT) is applied for encoding each block. The two-dimensional DCT of order $N \times N$ is defined as

$$F(u, v) = \alpha(u)\alpha(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i, j) \cos \left[\frac{(2i+1)\pi u}{2N} \right] \cos \left[\frac{(2j+1)\pi v}{2N} \right]$$

for $0 \leq i, j, u, v \leq N-1$

Where

$$\alpha(u) = \alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u, v = 0 \\ \sqrt{\frac{2}{N}} & \text{for otherwise} \end{cases}$$

Although fast algorithms can significantly reduce the computational complexity of computing the DCT, floating-point operations are still required [3]. Despite their accuracy, floating-point operations are expensive in terms of circuitry complexity and power consumption. Therefore, minimizing the number of floating-point operations is a sought property in a fast algorithm. For this reason, there has been huge interest in finding fixed point multiplication-free DCT algorithms that can be implemented as low power and area efficient digital circuits. Numerous architectures have proposed a low power, high speed and area efficient hardware implementation for DCT computation [4].

We introduce a new 1D-DCT approximation that possesses an extremely low arithmetic complexity, requiring only 12 addition. In this paper, a low complexity multiplier-less DCT approximation is proposed, which is more essential for hardware realization. All introduced implementations are sought to be fully parallel time-multiplexed 2-D architectures for 8×8 data blocks. Additionally, the proposed designs are based on successive calls of 1-D architectures taking advantage of the separable property of the 2-D DCT kernel.

A1CSA architectures and Transpose memory further provide significant area savings, delay and power consumption reduction and energy efficiency because of fast proposed DCT algorithms [5]. To examine the performance and trade-offs associated with the algorithm, we have coded the proposed algorithms in Verilog HDL, and it is synthesized with Xilinx Spartan 3E XC3S250E-5-PQ208 device.

The rest of the paper is structured as follows. In Section II, the proposed transform and the factors influencing its performance improvements and

computational complexity are compared with the existing methods. In Section III, digital hardware architectures for discussed algorithms are supplied both for 1D and 2D analysis. Hardware resource consumptions using field programmable gate array (FPGA). The hardware implementation for the proposed approximation DCT are detailed and analyzed based AICSA and transpose memory in Section IV. Conclusion and final remarks are given in Section V.

II. PROPOSED APPROXIMATE DCT TRANSFORM

Haweel introduces the approximation DCT method by applying the signum function operator to the DCT element in Equation 1. The DCT is given by

$$F'(u,v) = \frac{1}{\sqrt{N}} \text{sign}\{F(u,v)\}$$

Where $\text{sign } F(u,v) = \{.\}$, which is the signum function defined as follows:

$$\text{sign}\{x\} = \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Signed DCT has many advantages, one of which is apparent from looking at above Equations as all the elements in the transform are 0 or ± 1 , which eradicates the need of a multiplication operation or a transcendental expression. The transform order need not be a specific integer or a power of 2. The proposed DCT also maintains the periodicity and spectral structure of its originating DCT and in turn maintains good de-correlation and energy compaction characteristics. Therefore, proposed DCT is highly preferred for low computation applications.

There have been many recent approaches for reducing the computational complexity of the DCT transform, but the reduction in computational complexity comes at the cost of PSNR. In this paper, a new DCT approximation scheme is developed by reproducing the reported butterfly structures. After reviewing these structures, the common computations are identified and shared to remove the redundancy in DCT matrix. The image compression performance was evaluated based on the PSNR values, the matrix is altered and the procedure is repeated. First, the transform matrix is reduced to 16 addition and then to 14 addition and to 12 addition [6]. The forward and inverse transform matrices are obtained as follows:

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

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} * D$$

where $D = \text{diag}(1, 1, 1, 1, 1, 1, 1, 1) * 1/2$.

It can be seen from above Equations that the entries of T and T^{-1} are $\{0, \pm 1\}$. This indicates that the proposed transform requires only 12 additions, thus avoiding the need for multiplication and bit shift operations. In terms of complexity assessment, the diagonal matrix D may not introduce any computational overhead. In JPEG, the DCT operation is a pre-processing step for a subsequent coefficient quantization procedure. Consequently, the scaling factors in the diagonal matrix D can be merged to the de-quantization matrix.

The common use of additions is reduced without disturbing PSNR in considerable levels. The number of additions, multiplications, and bit-shift operations required for the proposed transform and the evolution of DCT is presented in Table 1.

TABLE I: Arithmetic complexity analysis

METHOD	MULT	ADD	SHIFT	TOTAL
Exact DCT	64	56	0	120
Arai algorithm	5	29	0	34
BAS-2008	0	18	2	20
BAS-2011 with a=0	0	16	0	16
CB-2011	0	22	0	22
Modified CB-2011	0	14	0	14
Approximate DCT	0	24	6	30
App. DCT[4]	0	14	0	14
App. DCT[6]	0	12	0	12

III. DIGITAL ARCHITECTURES AND REALIZATIONS

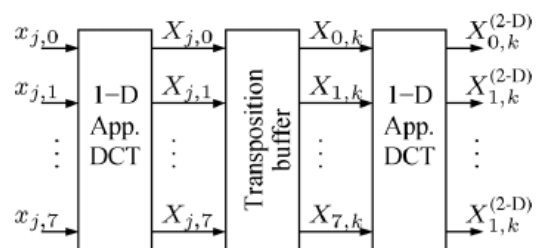


Fig.1. Block diagram of 2D-DCT

The digital architecture of the proposed approximate DCT is shown in Fig.1. The hardware cost is measured by the number of adders, multipliers, and shifters used in the architecture, and the computing time is normalized as clock cycles.

The 1-D approximate DCT blocks (Fig.2) 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.

Between the approximate DCT blocks a real-time row-parallel transposition memory 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 memory consists of an 8x8 register array in which each register is 12 bits long depending on the stage of calculation as shown in Figure 3. The inputs to the transpose can either be 8 12-bit values. Eight clock cycles after receiving the last input, there will either be 8 12-bit outputs.

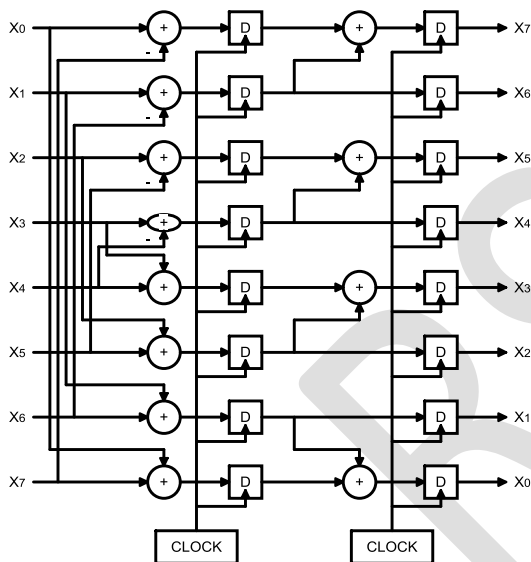


Fig.2. Digital architecture for 1D-DCT

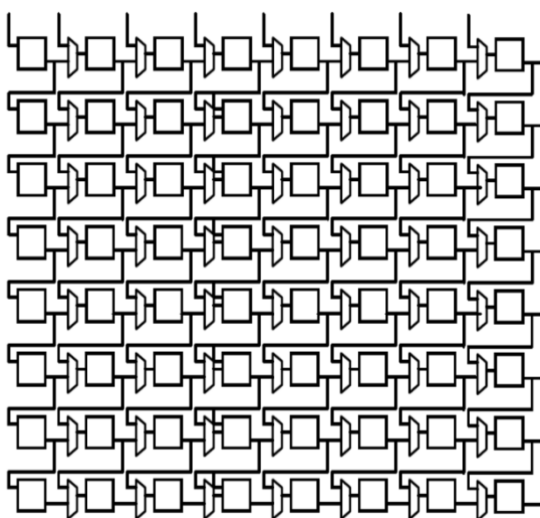


Fig.3. Digital architecture for transpose memory

The above structure acts as a memory transpose unit by taking in 8 inputs at a time, then shifting them down on every cycle. Once all 8 rows have been filled, the select line goes low, and on each clock, the transposed values can be output 1 at a time (for a total of 8 clock cycles to get all 8 values out). The DCT unit controls the data movement through rowed and columned signals that are then mapped to a select line that controls the multiplexers.

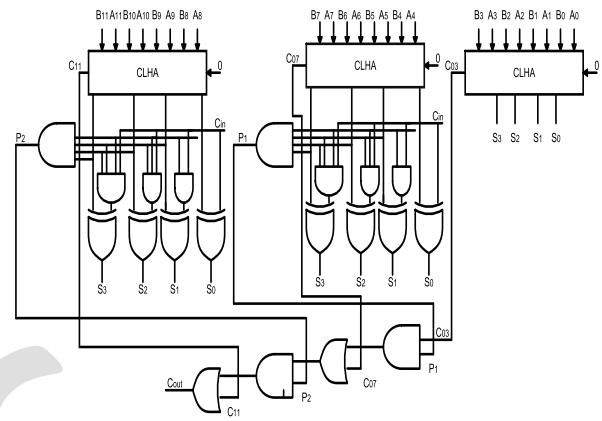


Fig.4. A1CSA fast adder

Only 12 addition are used for calculating one dimensional DCT but for two dimensional DCT totally 24 additions are required. Hence we introduce 12 bit A1CSA fast adders for addition operations of fast DCT algorithm. We use basic 4-bit addition block of carry look ahead adder for high speed carry propagation. As well as A1S is also introduced to generate p_{topb} signal that serves to accelerate the carry propagation, this has an important impact on the circuit critical delay. The architecture for 12-bit A1CSA is shown in Fig.4.

IV. FPGA IMPLEMENTATIONS

Discussed methods were physically realized on a FPGA based rapid prototyping system for various register sizes and tested using on-chip hardware-in-the-loop co-simulation. The architectures were designed for digital realization within the Xilinx with synthesis options set to generic Verilog HDL generation. This was necessary because the auto-generated register transfer language (RTL) hardware descriptions are targeted on both FPGAs. The proposed architectures were physically realized on Xilinx 3E XC3S250E-5-PQ208 device at a clock frequency of 176MHz.

Selected Device: 3s250epq208-5

Number of Slices: 852 out of 2448 34%

Number of Slice Flip Flops: 1138 out of 4896 23%

Number of 4 input LUTs: 1529 out of 4896 31%

Number of IOs: 195

Number of bonded IOBs: 195 out of 158 81%

Number of GCLKs: 1 out of 24 4%

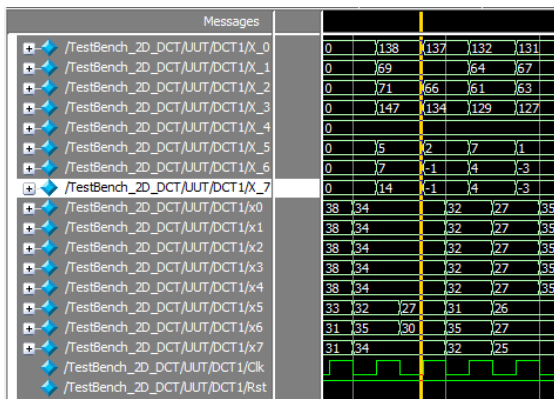


Fig.5. Simulation result

V. CONCLUSION

In this paper, we proposed a low-power 8-point 2D-DCT approximation that require only 24 addition operations thus avoiding the need for multiplication and bit shift operations for computations and hardware implementation to the proposed transform and several other prominent approximate DCT methods. The proposed transform possess lower computational complexity and are faster than all other approximations under the consideration. The proposed transform could outperform in terms of image compression. Hence, the new proposed transform is the best approximation for the DCT in terms of computational complexity, speed among the approximate transform examined and improvement in the peak signal-to-noise ratio. Introduced implementations address both 1D and 2D approximate DCT. All the approximations were digitally implemented using both Xilinx FPGA tools.

REFERENCES

- [1]. N Ahmed, T Natarajan, KR Rao, (1974). 'Discrete cosine transform', IEEE Trans On Computers C-23(1), 90-93
- [2]. WB Pennebaker, JL Mitchell, (1992). 'JPEG Still Image Data Compression Standard', (Van Nostrand Reinhold, New York, NY, USA)
- [3]. V. Britanak, P. Yip, and K. R. Rao, (2007). 'Discrete Cosine and Sine Transforms'. New York, NY, USA: Academic.
- [4]. Uma Sadhvi Potluri, Arjuna Madanayake, Renato J. Cintra, Fábio M. Bayer, Sunera Kulasekera, and Amila Edirisuriya, (2014). 'Improved 8-Point Approximate DCT for Image and Video Compression Requiring Only 14 Additions', IEEE transactions on circuits and systems—I, VOL. 61, NO.6.
- [5]. Jucemar Monteiro, José Luís Güntzel, Santa Catarina Florianópolis, Brazil Luciano Agostini, (2011). 'A1CSA: An Energy-Efficient Fast Adder Architecture for Cell-Based VLSI Design', Group of Architectures and Integrated Circuits (GACI), Federal University of Pelotas, Pelotas, Brazil, IEEE.
- [6]. Vaithiyathan Dhandapani and Seshasayanan Ramachandran, (2013). 'Area and power efficient DCT architecture for image compression', in pceeding of the International Conference on Advanced Computing and Communication Systems (ICACCS) (Coimbatore, Tamil Nadu, India), pp. 1-6.