Speed Optimized 2D DCT Implementation for High Compression Rate Applications

K.Santhosh Kumar[1], B.N. Srinivasa Rao [2],
[1] Pursuing M.Tech in VLSID, Dept. of ECE, Avanthi Institute Of Engineering And Technology, Andhra Pradesh, India
[2] Asst. Professor, Dept. of ECE, Avanthi Institute Of Engineering And Technology, Andhra Pradesh, India,
karothu.santhosh@gmail.com, nagasrinu.b@gmail.com [2]

Abstract: Multimedia applications, and in particular the encoding and decoding of standard image and video formats, are usually a typical target for Systems on Chip (SoC). The bi-dimensional Discrete Cosine Transformation (2D-DCT) is a commonly used frequency transformation in graphic compression algorithms. Many hardware implementations, adopting disparate algorithms, have been proposed for Field Programmable Gate Arrays (FPGA). These designs focus either on performance or area, and often do not succeed in balancing the two aspects. This paper presents two high performance FPGA architectures for the 2D DCT computation for Ultra High Definition video coding systems. Both architectures use Distributed Arithmetic to perform the necessary multiplications instead of traditional multipliers. The first architecture uses 105 clock cycles to transform an 8x8 block and reaches a rate of up to 206 samples per second at a 338.5 MHz frequency, while the second one requires 65 cycles for each 8x8 block and achieves a rate equal to 252 samples per second at 256 MHz's Both architectures have been implemented using VHDL. Virtex7 FPGA of Xilinx has been used for the realization of both implementations.

Keywords—Video Coding, 2D DCT, Distributed Arithmetic, FPGA Implementation, VHDL.

I. INTRODUCTION

High dynamic range (HDR) video and image transmission over digital communication channels is undergoing exponential growth. With the increasing demand for high-definition programming, there exists a strong need for efficient digital video coding (DVC) that provides high data compression ratios which in turn leads to better utilization of network resources. The H.264/AVC standard does not provide the required compression ratios for emerging capture and display technologies such as ultra high definition (UHD), multiview, and auto stereoscopy. To address such emerging needs, the Joint Collaborative Team on Video Coding (JCT-VC) has developed the successor for H.264/AVC, called High Efficiency Video Coding (HEVC). The HEVC standard aims at achieving a 50% reduction in data rate compared with its predecessors while maintaining low complexity computation. Video compression systems operating at high frequencies and resolutions require hardware capable of significant throughput with tolerable area and power requirements. Real-time video compression circuits having high numerical accuracy are needed for next generation video, coding systems, and retina displays.

The two-dimensional (2D) 8×8discrete cosine transform (DCT) is a fundamental operation in real-time video systems, which is adopted in compression standards, such as JPEG, MPEG-1, MPEG-2, H.261, H.263, H.264, and most recentlyH.265/HEVC. The DCT is the de facto standard in transform coding, with extensive applications in modern DVC standards, due to its superior energy compaction on par with the optimum Karhunen–Lo`eve transforms, achieved at reasonably low computational complexity. The circuit realization of the 2D 8×8 DCT affects noise, distortion, circuit area, and power consumption of such compression systems. The 2D DCT implementation is essentially dependent on the one-dimensional (1D) DCT. The 8-point 1D DCT requires multiplications by numbers in the form $c[n] = \cos(n\pi/16)$, $n = 0, 1, \dots, 7$. These constants impose implementation difficulties in terms of their machine representation, because they are irrational values. Fixed-point arithmetic DCT implementations usually employ rounding off to approximate such quantities, which introduces errors.

In this paper two efficient 2D DCT architectures are presented, capable to manage 8x8 image blocks. Both architectures use DA in order to improve the time performance. The architecture of 2D DCT in was used as reference architecture. Although this architecture was optimized for low power applications and implemented in ASIC, there is no need for high level of throughput, since two high speeds FPGA architectures are implemented in this paper.

2. REVIEW AND BACKGROUND

Here we will see different types of cosine transforms which are available for various applications.

Table1: Multiplicands required in the 8-point Chen's fast DCT algorithm.

Exact value	cos(π/32)	cos(π/8))	cos(3π/16	cos(π/4)	cos(5π/16)	cos(3π/4)	cos(3π/8)
Approximation (scaled by √2)	89/64	83/64	75/64	64/64	50/64	36/64	18/64

2.1. HEVC Integer Cosine Transform:

The integer cosine transform (ICT) required in the HEVC specification is a scaled approximation of Chen's fast algorithm for computing the 1D DCT. Since the computational architecture is based on a regular butterfly structure, the ICT has lower circuit complexity at the physical layer. However, the ICT algorithm is not optimal in terms of speed or accuracy [4, 20]. Despite such sub optimality, the ICT algorithm has the advantage of being extensible to larger transform sizes such as 32 × 32 and 64 × 64, leading to its adoption in HEVC.

2.2. 8-Point Chen's Fast DCT Algorithm:

The fast algorithm proposed by Chen et al. for computing the 8-point DCT requires 26 additions and 12 multiplications. The seven *different multiplicands* employed in this algorithm are given in Table 1. The *exact* DCT coefficients can be obtained by scaling the output obtained from the butterfly structure given in by 4. In the suggested reference C++ software implementation [20], Chen's algorithm was implemented in 16-bit arithmetic, using *short* variables to represent intermediate results. Suitable approximations for the irrational constants in the DCT are scaled by a factor of √2, leading to Table 1. In order to compute the 2DDCT, the 1D transform is applied twice: first in a row-wise manner, then column-wisely, with a transposition operation in between. Hence, the errors that are introduced in the rounding operation after the row and column transform calculations propagate through the 2D DCT computation and are present as additive noise at the final quantizer stage. The additive noise injected within the 2D transformation algorithm can affect a visible impact on picture fidelity especially for low levels of compression.

The 2D DCT architecture uses the row–column distributed arithmetic version of the Chen fast DCT algorithm. The first step of the Chen algorithm is a factorization of the DCTII matrix such that the subsequent computation of the even indexed coefficients is fully separated from the computation of the odd indexed coefficients. The 1D DCT

$$\begin{bmatrix} X0 \\ X2 \\ X4 \\ X6 \end{bmatrix} = \begin{bmatrix} A & A & A & A \\ B & C & -C & -B \\ A & -A & -A & A \\ C & -B & -B & C \end{bmatrix} \begin{bmatrix} x0+x7 \\ x1+x6 \\ x2+x5 \\ x3+x4 \end{bmatrix}$$

$$\begin{bmatrix} X1 \\ X3 \\ X5 \\ X7 \end{bmatrix} = \begin{bmatrix} D & E & F & G \\ E & -D & -D & -F \\ F & -G & G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} x0-x7 \\ x1-x6 \\ x2-x5 \\ x3-x4 \end{bmatrix}$$

coefficients *X*k, k=0,1,…,7 for an 8-point input vector *x*n, n=0,1,…,7 can be expressed as follows:

Where A= cos(π/4), B= cos(π/8), C= cos(π/8), D= cos(π/16), E= cos(3π/16), F= cos(3π/16), G= cos(π/16),

The 2-D DCT (8 x 8 DCT) is implemented by the row-column decomposition technique. We first compute the 1-D DCT (8 x 1 DCT) of each column of the input data matrix, after appropriate rounding or truncation, the transpose of the resulting matrix, it is stored in an transpose buffer. We then compute another 1-D DCT (8 x 1 DCT) of each row of the transpose matrix to yield the desired 2-D DCT. A block diagram of the design is shown in Fig 1.

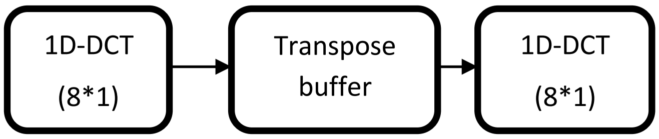


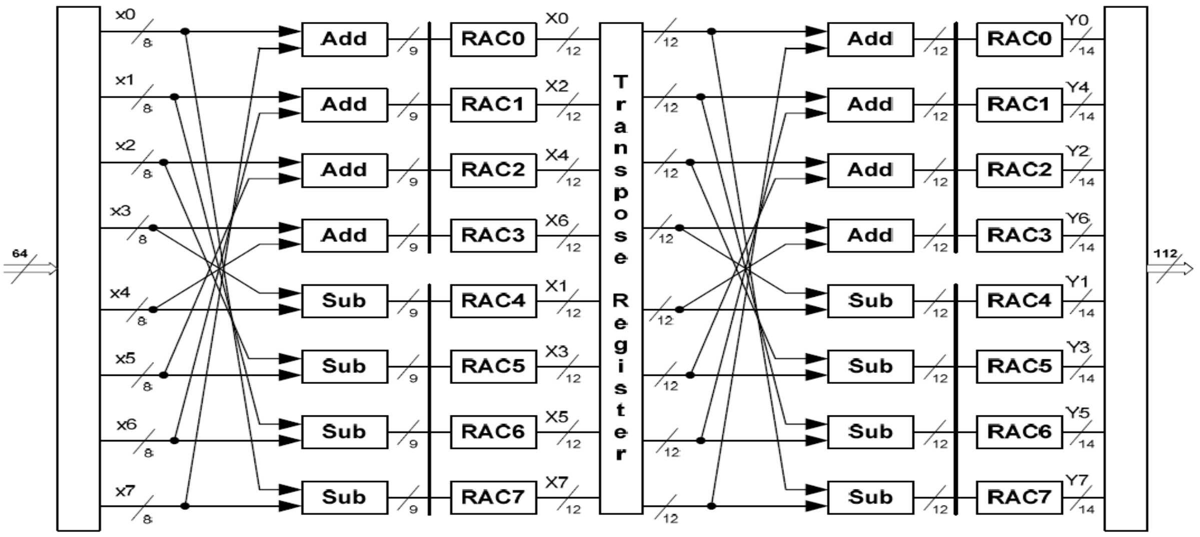
Fig 1: 2D-DCT Architecture

In addition, to allow the use a global 2D-DCT pipeline, a special transpose buffer must be designed, since the first DCT produces row results, and the second DCT needs column values as input. This memory should have ping pong1 features to permit to the first 1D architecture to write different values that could be read by the second 1D architecture. This leads to even more space occupation on FPGA.

III. HARDWARE ARCHITECTURE

The hardware architecture of the 2D DCT is shown in Fig.2. The design has a 64-bit data input and 112-bit output. Each input coefficient is equal to 8 bits. So, the eight coefficients (64-bit) of each row are shifted into the register during the first clock cycle. In the next stage, the adders and subtractors perform the first butterflies. In order to keep full accuracy, the outputs of the butterflies should be 9 bits long. Then, the data are loaded into parallel-in serial-out registers that repackage the data into 4-bit addresses that serially feed the ROM and Accumulators, from RAC0 to RAC7, with MSB first.

Fig. 2. The 8x8 2D DCT architecture



A straightforward architecture of the RAC by using DA structure is depicted in Fig. 3. Although the serial inputs limit the performance of such a structure a better performance can be obtained by using more hardware resources.

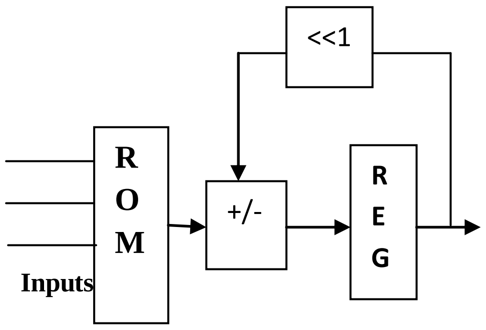


Fig 3. Straight forword ROM Accumulator architecture.

Fig. 4 illustrates two bit-sum that can be computed at a time by duplicating the ROM and adder tree. The first ROM is used by even indexed input bits and the second one is used by odd indexed input bits. The odd bit partials are left shifted to properly weight the result and added to the even partials before accumulating the aggregate. Since two bits are taken at a time, the scaling accumulator has to shift the feedback by two bits on the left. The ROM is made 10-bit wide to accommodate for the largest sum without overflow. ROMs contain the sums of the constant coefficients, for all the possible serial input combinations as shown in Fig. 5. The adder that sums the odd and even partials is 10-bit and the accumulator adder is 20-bit long to accommodate for the continuously increasing operands due to the logical shifting in the feedback path. The results of the first RACs computations are 20-bit 2's-complement and are rounded to 12-bit 2's-complement before being stored in the transpose register. Actually, the results of the first RACs are the row 1D DCT. The resulted coefficients are taken in the order of X0, X2, X4, X6, X1, X3, X5, X7. Every 5 cycles, the final results within each RAC are computed. As it has been mentioned above, two bits per cycle are processed in the RAC. So, in order to balance the outputs of the butterflies, which are 9 bits long, a bit equal to zero is appended at the end of the butterflies' outputs. This does not cost any inaccuracy because the value of the ROM zero address is equal to zero.

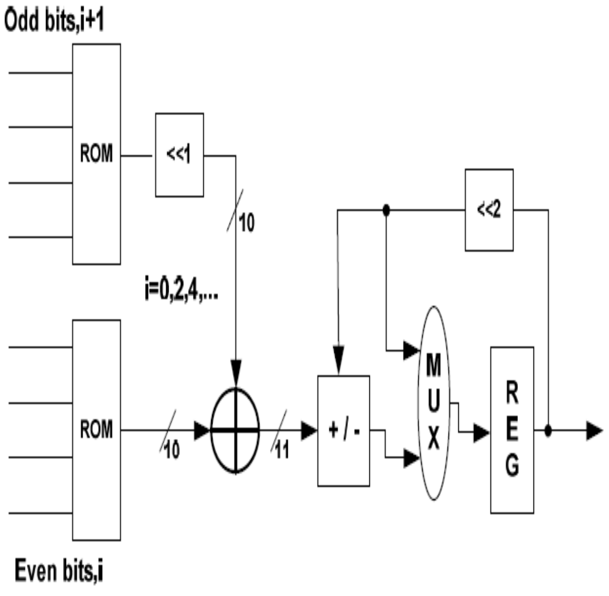


Fig 4. The proposed RAC architecture

The corresponding ROM structure that contains sums of all the constant coefficients, for all the possible serial input combinations are shown in the below figure

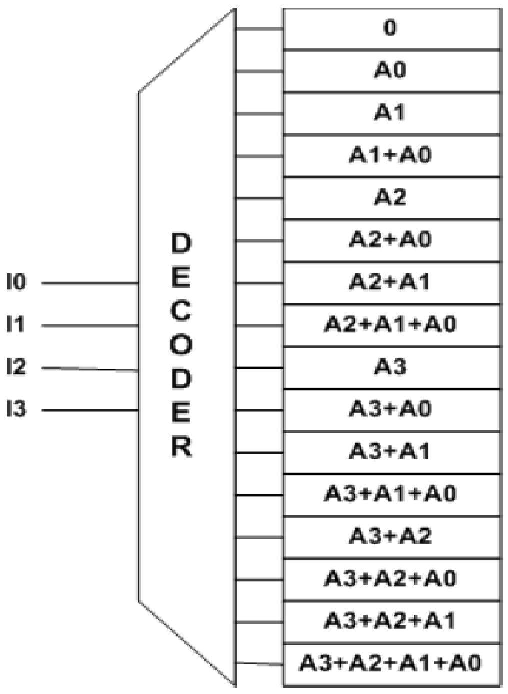


Fig. 5: the content of the ROM

The transpose register consists of eight 96-bit serial-input parallel-output registers and eight 8-to-1 multiplexers. The architecture of the transpose register is depicted in Fig. 6. So, the coefficients that have been computed by the first RACs are rearranged in an ascending order and stored in each register. After 8 executions the transpose register has been filled with the DCT coefficients of the eight rows. Then, each register outputs its content in parallel and enables the eight multiplexers. During the first operation the multiplexers select the most significant byte from each register and feed the second 1D DCT with the first column. During the second operation, the multiplexers feed the second column and so on.

Then, a similar 1D DCT is applied on the eight columns. As shown in Fig. 1, in order to meet high levels of accuracy, the outputs of the butterflies are 12 bits long and the outputs of the RACs are 14 bits long.

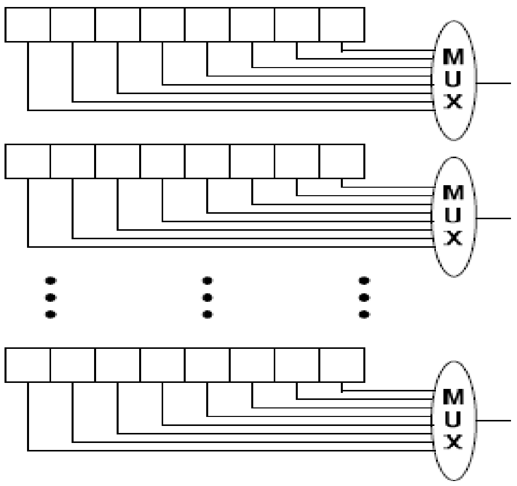
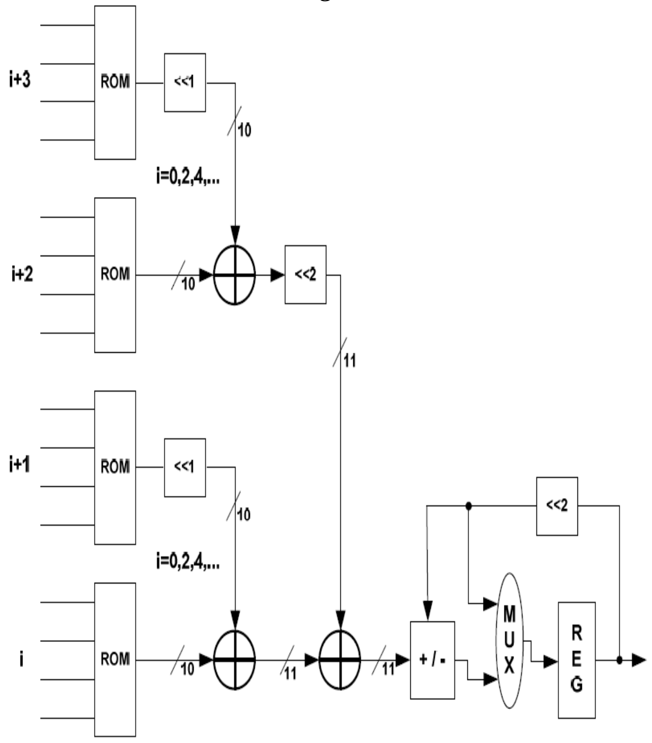


Fig. 6. The Transpose Register Architecture

The final results within each RAC are computed every 6 cycles. This architecture requires a maximum transpose register of 768 1-bit registers and parallelized row and column RAC stages. Additionally, this architecture needs 105 clock cycles in order to process an 8x8 image block. In order to increase the time performance of the architecture, a second option of the RAC architecture is proposed. Specifically, four bit-sums can be computed in parallel using four ROMs. This architecture is illustrated in Fig. 7



.**Fig. 7. Four ROM RAC Architecture**

The first ROM is used by the even indexed input bits, the second one is used by odd indexed input bits, the third by every other even indexed and the fourth by every other odd indexed input bits. This RAC scheme reduces by half the execution cycles compared to the previous one, at the expense of an increase in hardware resources. This second architecture needs 65 clock cycles in order to process an 8x8 image block.

IV. SYNTHESIS RESULTS

Initially, a behavioral model was developed that simulated the behavior of the 2D DCT algorithm. This model has been used in order to verify the correct functionality of the proposed architectures of the 2D DCT. The proposed architectures have been captured using VHDL. The VHDL codes are implemented and synthesized with the Xilinx ISE 13.1 tool. The target FPGA device was XC7VX330T-3FFG1157. The measurements are focused on the design throughput and the consumed FPGA area resources. Table II presents the synthesis results of the proposed architectures. The proposed architecture that uses the RAC structure with 2 ROMs is shown as 2D_DCT_2ROM, while the proposed architecture that uses the RAC structure with 4 ROMs it was symbolized as 2D_DCT_4ROM.

TABLE II. SYNTHESIS RESULTS

FPGA Device	XC7VX330T-3FFG1157	
FPGA Resources	Architectures	
	2D_DCT_2ROM	2D_DCT_4ROM
Slice Registers	1354	1110
Slice LUT's	1786	2021
Freq (MHz)	338.5	256
Bit rate (Samples/sec)	206	252

The 2D_DCT_2ROM architecture consumes less hardware resources (1786 slice LUT) and yields better clock frequency (338.5 MHz) as compared to 2D_DCT_4ROM architecture that consumes 2021 slice LUT and achieves a clock frequency up to 256 MHz The 2D_DCT_2ROM architecture needs 105 clock cycles to manage 64 samples (an 8x8 image), which corresponds to a bit rate equal to 206 Samples per second.

V. CONCLUSIONS

The 2-D DCT and 1D-DCT architectures which adopts algorithmic strength reduction technique to reduce the device utilization pulling the power consumption low have thus been designed. The DCT computation is performed with sufficiently high precision yielding an acceptable quality. Two high speed FPGA architectures for the 2D DCT computation are presented in this paper. Both architectures use Distributed Arithmetic in order to replace the multipliers that are needed by the DCT algorithm. In the 2D_DCT_2ROM architecture two ROMs are used in order to double the performance, while in the 2D_DCT_4ROM four ROMs are used to quadruple the performance. The synthesis results prove that both architectures are very good choices for applications of high time performance requirements.

REFERENCES

[1]. Enas Dhuhri Kusuma, Thomas Sri Widodo "FPGA Implementation of Pipelined 2D- DCT and Quantization Architecture for JPEG Image Compression" Proceedings of 978-1-4244-6716-7/10/$26.00 ©2010 IEEE

[2]. Vijay Kumar Sharma, K. K. Mahapatra and Umesh C. Pati "An Efficient Distributed Arithmetic based VLSI Architecture for DCT"

[3]. Chidanandan, Bayoumi, M, "Area-Efficient NEDA Architecture for The 1-D DCT/IDCT" ICASSP 2006 Proceedings. 2006 IEEE International Conference.

[4]. Zhenyu Liu Tughrul Arslan Ahmet T. Erdogan "A Novel Reconfigurable Low Power Distributed Arithmetic Architecture for Multimedia Applications" Proceedings of 1-4244-0630-7/07/$20.00 C 2007 IEEE.

[5]. S.Saravanan, Dr.Vidyacharan Bhaskar, P.T.Lee ,"A High Performance Parallel Distributed Arithmetic DCT Architecture for H.264 Video Compression" European Journal of Scientific Research ISSN 1450- 216X Vol.42 No.4 (2010), pp.558-564 EuroJournals Publishing, Inc. 2010.

[6]. VLSI Digital Signal Processing Systems: Design And Implementation (Hardcover) By Keshab K Parhi

[7]. Vijaya Prakash.A.M, K.S.Gurumurthy "A Novel VLSI Architecture for Digital Image Compression Using Discrete Cosine Transform and Quantization", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.9, September 2010

[8] J. Lee and H. Kalva, "Video Coding Techniques and Standards", In Furht B. (Ed.) Encyclopedia of Multimedia, Springer-Verlag Berlin Heidelberg, 2008.

[9] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, Issue 7, pp. 560-576, July 2003.

[10] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, December 2012.

[11] S. A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review", IEEE ASSP Magazine, Vol. 6, Issue. 3, pp. 4-19, July 1989.

[12] T. Xanthopoulos, and A. P. Chandrakasan, "A Low-Power DCT Core Using Adaptive Bitwidth and Arithmetic Activity Exploiting Signal Correlations and Quantization", IEEE Journal of Solid-State Circuits, Vol. 35, No. 5, pp. 740-750, May 2000.

[13] W. H. Chen, C. H. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Tranactions on Communications, vol. COM-25, pp. 1004–1009, September 1977.

.