A Simple VLSI Architecture for Computation of 2-D DCT, Quantization and Zig-zag ordering for JPEG

Vijay Kumar Sharma[1] Umesh C. Pati[2] and K. K. Mahapatra[3]
Dept. of Electronics & Comm. Engg.
National Institute of Technology, Rourkela, India-769008
[1]vijay4247@gmail.com, [2]ucpati@nitrkl.ac.in, [3]kmaha2@gmail.com

Abstract— **In this paper, first a comparative simulation study of PSNR is done for two quantization tables, one recommended by JPEG committee and another suitable for hardware simplification. Simulation results indicate that quantization table suitable for hardware simplification can be used for designing JPEG baseline coder circuitry. Then we present a simple finite state machine (FSM) based VLSI architecture and its FPGA implementation from discrete cosine transform (DCT) to zig-zag ordering of transformed coefficients for JPEG baseline coder. 1-D DCT implementation is done for the compressed distributed arithmetic (DA) algorithm reported in previous literature with shifting performed by division operator. Quantizer using only shifter (no adder) and 2-D DCT are combined in single step. Implementation is done on XC2VP30 device on Xilinx Virtex-II Pro FPGA board.**

Keywords— **JPEG, discrete cosine transform (DCT), quantization table, FPGA based design, distributed arithmetic (DA).**

I. Introduction

Digital camera uses JPEG standard to compress the captured image data from sensors so as to reduce the storage requirements [1-5]. JPEG compresses image data in three steps. They are 8x8 block-wise discrete cosine transform (DCT), quantization and entropy coding. DCT transforms the image data from spatial domain into frequency domain which are called DCT coefficients. Most of the visual information is stored in few low frequency DCT coefficients and they are used for further coding while high frequency coefficients are discarded. Quantization is used to bring further compression by representing DCT coefficients with no greater precision that is necessary to achieve desired image quality [6,7]. Quantized DCT coefficients are reordered in zig-zag fashion in increasing order of frequency. Finally entropy coding is done to eliminate the redundancy in quantized data representation [8].

DCT is a computation intensive algorithm and its direct implementation requires a large number of adders and multipliers. Distributed Arithmetic (DA) is a technique that reduces the computation for hardware implementation of digital signal processing algorithm [9]. DCT implementation using DA is done in literatures [10-12]. For the 64 DCT implementation of quantizer, which is done by dividing each coefficients by its corresponding quantizer step-size, a memory module and a divider is required. Quantizer implemented in [15] uses16-bit multiplier and RAM, in [16] 13x10-bit multiplier and RAM are used whereas adders and shifters are used in [13] and [14]. A default quantization table is recommended by JPEG committee. But users are free to use their own quantization table for required perceptual quality and compression [17-20].

In this paper, first a comparative simulation study of PSNR is done for two quantization table one recommended by JPEG committee and another suitable for hardware design simplification. Simulation results show that quantization table suitable for hardware simplification can be used for the design of JPEG baseline circuitry. Then we have presented a simple finite state machine (FSM) based architecture from DCT to zig-zag reordering of quantized coefficients and implemented in FPGA. 1-D DCT implementation is done for the distributed arithmetic (DA) algorithm reported in literature [12] with shifting done by division operator. Quantizer using only shifter (no adder) and 2-D DCT are combined in single step to save additional hardware requirement. Implementation is done on XC2VP30 device on Xilinx Virtex-II Pro FPGA board.

This paper is organized as follows. Section II gives an overview JPEG baseline coding procedure. A comparative simulation for two normalization matrix one suitable for hardware implementation and another recommended by JPEG committee is done in Section III. Section IV presents an architecture for the computation of 2-D DCT, quantization and zig-zag order of quantized coefficients. FPGA implementation and results are discussed in section V. Conclusions are drawn in section VI.

II. JPEG Baseline Coding Procedure Overview

Fig.1 shows the JPEG baseline coder diagram. Image is divided into 8x8 blocks. 2-D DCT is taken to blocks in sequential order from left to right and top to bottom. 8x8 2-D DCT for a set of 2-D data $X(i,j)$ with $0 \le i \le 7$ and $0 \le j \le 7$ is given by,

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{i=0}^{7}\sum_{j=0}^{7} X(i,j) \times cos\left(\frac{(2i+1)u\pi}{16}\right)cos\left(\frac{(2j+1)v\pi}{16}\right) \tag{1}$$

where u, v = 0,1,…,7, and $C(u), C(v) = \sqrt{1/2}$ for $u, v = 0$ and $C(u), C(v) = 1$ otherwise. The 2-D equation (1) is a complex and to implement it require many additions and multiplications. So it is decomposed into two 1-D 8x1 DCT. 1-D DCT is given by,

$$F(u) = \frac{1}{2}C(u)\sum_{i=0}^{7} X(i)cos\left(\frac{(2i+1)u\pi}{16}\right) \tag{2}$$

for $0 \le u \le 7$.

2-D DCT of 8x8 data is obtained by first taking 8 point 1-D DCT to each row and then 8 point 1-D DCT to each column. DCT removes the inter-pixel redundancy from the image data and makes it suitable for coding. Data from DCT output is quantized by a quantizer. Quantization is done by dividing each DCT coefficient by corresponding quantizer step size followed by rounding to nearest integer.

$$F^{Q}(u,v) = Integer\ Round\ \left(\frac{F(u,v)}{Q(u,v)}\right) \tag{3}$$

Quantizer step size is determined by the acceptable visual quality of image. JPEG committee recommends the normalization matrix for quantization shown in table 1. Further scaling of normalization matrix can be done to bring more compression. After the quantization, coefficients are arranged in increasing frequency order. Table 2 shows the zig-zag sequence that is used to re-arrange the transform coefficients for efficient coding. Top left coefficient is also called DC coefficient and rest of the 63 coefficients are called AC coefficient. AC coefficients are coded by entropy coding whereas difference of DC coefficients from previous block is coded by entropy coding. JPEG committee provides a standard table to be used for entropy coding.

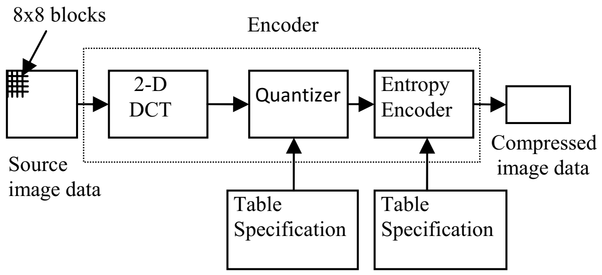


Fig.1. JPEG baseline coder block diagram

Table 1: A typical normalization matrix

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 2: Zig-zag order sequence

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

III. Normalization Matrix for Hardware Simplification

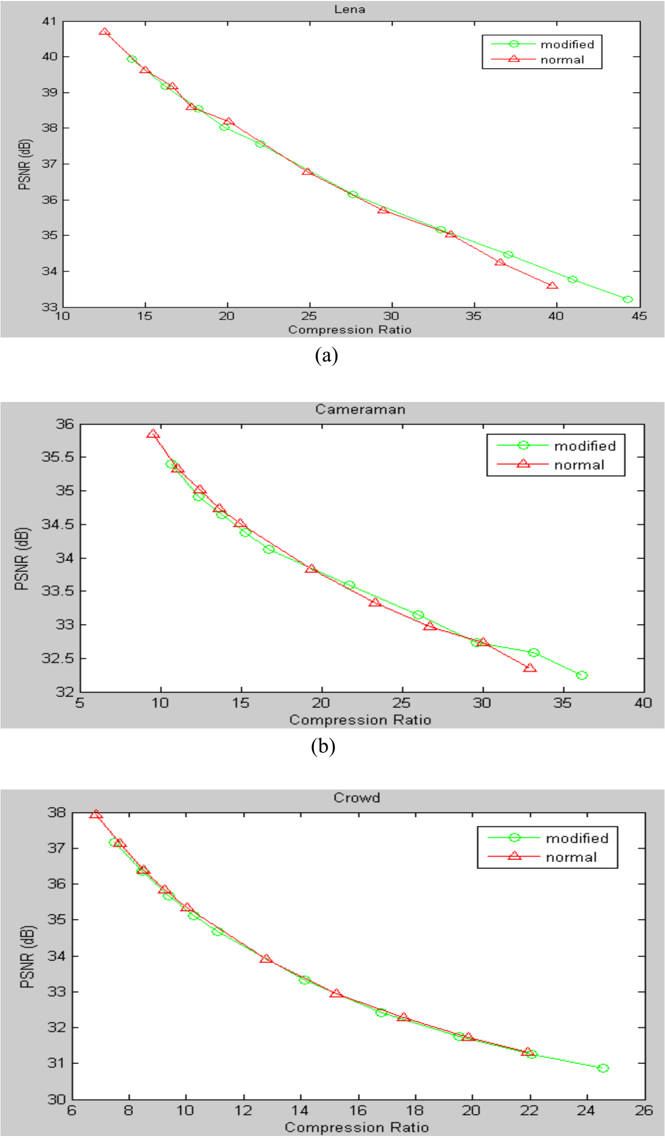
To implement quantization by dividing each coefficient with the corresponding value in normalization matrix in table 1, either shifting and addition or divider and memory is required. If we use the normalization matrix in table 3, only shifting and a control is required. PSNR against compression ratio for three standard images are plotted in fig.2. Fig.3 shows the original and reconstructed images obtained by using both normal and modified table. It is evident from fig.2 and fig.3 (image quality almost same) that modified table which is suitable for hardware implementation can be used in place of normal table.

Table 3: A modified normalization matrix for hardware simplification

16	16	16	16	32	64	64	64
16	16	16	16	32	64	64	64
16	16	16	32	32	64	64	64
16	16	32	32	32	64	64	64
32	32	32	64	128	128	128	128
64	64	64	64	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128

IV. Proposed Hardware Architecture

Architecture for the computation of 2-D DCT, quantization and zig-zag ordering of the quantized coefficients is shown in fig.4. 2-D DCT is computed in row-column decomposition method. When second 1-D DCT is being computed, timing and control will generate the eight addresses to store the eight transform coefficients in

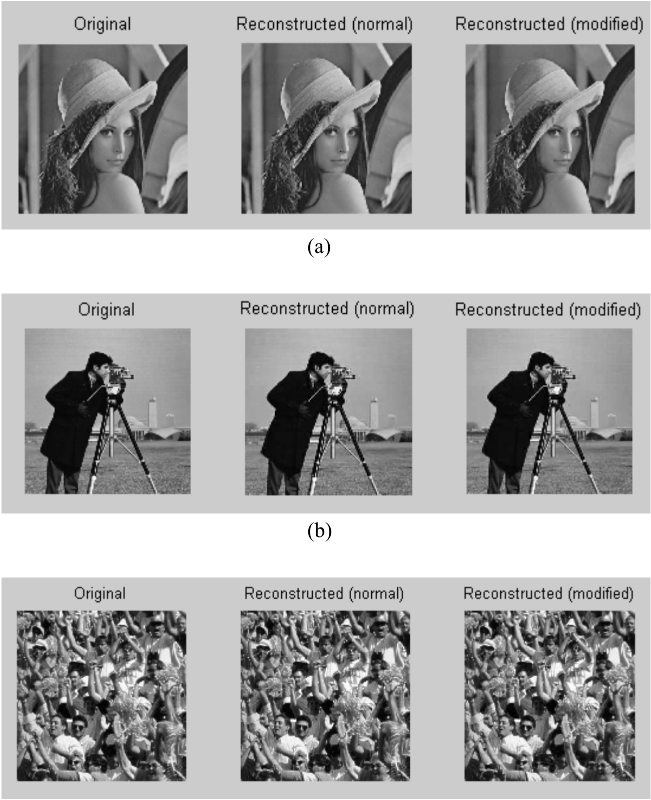


(a)

(b)

(c)

Fig.2. PSNR against compression ratio for (a) 448x448 Lena, (b) 256x256 Cameraman and (c) 512x512 Crowd images



(a)

(b)

(c)

Fig.3. Original and reconstructed images using normal quantization matrix and modified matrix in table 3 (a) 448x448 Lena, (b) 256x256 Cameraman and (c)512x512 Crowd images

appropriate place (zig-zag order) for that coefficients. For example first set of eight addresses generated will be 0, 2, 3, 9, 10, 20, 21, 35. Coefficients are stored in memory. Then a counter will start counting from 0 to 63 and coefficients which are already ordered in zig-zag way will be outputted after conditional shifting one by one in each clock cycle. So total latency of the architecture from 2-D DCT to zig-zag order is 64+latency of 2-D DCT computation.

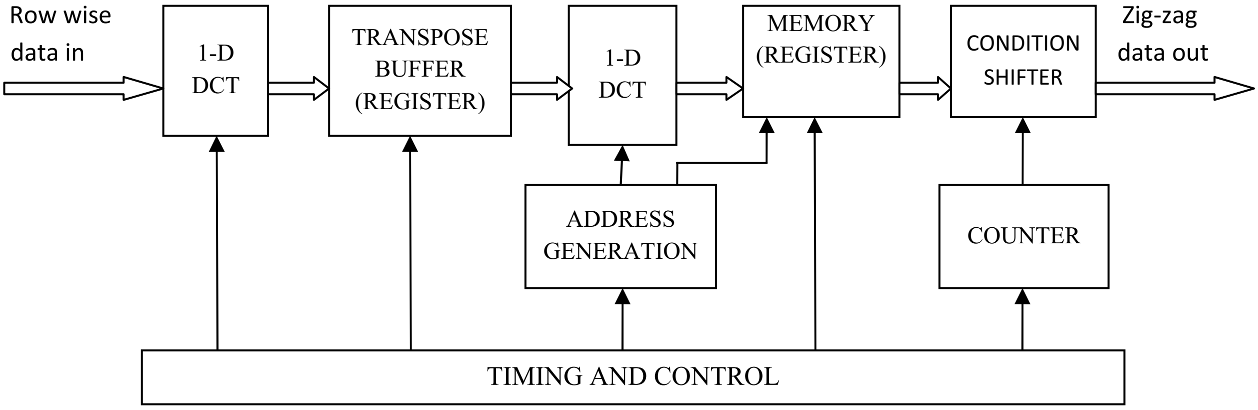


Fig.4. DCT to zig-zag re-order Architecture for JPEG baseline

V. FPGA Implementation Results and Discussions

For the 1-D DCT implementation compressed distributed arithmetic algorithm proposed in [12] is used but shifting is performed by division operation to reduce the error due to sign extension. Quantization is performed by both wire shifting and division operation. 8-bit input, 13-bit internal word representation and 12-bit DA precision is used for the implementation. Table 4 shows the hardware utilization summary implemented in Xilinx XC2VP30 device on Virtex-II Pro board. A considerable hardware savings is achieved when quantization is performed by wired shifting.

Table 4: Hardware utilization summary different implementations

FPGA-chip Xilinx XC2VP30			
	Only 2-D DCT	Zig-zag ordered using division operator for quantization	Zig-zag ordered using wire shifting for quantization
# of 4 input LUTs	4502	5276	4986
# of slices	2435	3070	2856
# of slice Flip Flops	868	1501	1409
Clock Freq. (MHz)	48.4	31.1	40
Power (W)	14.97	16.58	16.53

VI. Conclusions

A comparative simulation study of normalization matrix suitable for hardware implementation and another recommended by JPEG is done. Simulation results show that normalization table that is suitable for hardware simplification can be used in JPEG baseline image compression. Then a simple FSM based architecture for the computation of 2-D DCT, quantization and zig-zag ordering for JPEG image compression is presented and implemented in Xilinx XC2VP30 device. Normalization table used is one for hardware simplification. Implementation result shows a considerable amount of hardware savings when quantization is performed by wired shifting operation.

Acknowledgment

The authors acknowledge to DIT (Ministry of Information & Communication Technology) for the financial support for carrying out this research work.

References

[1] Jin-Maun Ho, Ching Ming Man, "The Design and Test of Peripheral Circuits of Image Sensor for a Digital Camera," *IEEE International Conference on Industrial Technology, 2004. IEEE ICIT '04*, vol.3, pp.1351 – 1356, 8-10 Dec. 2004.

[2] Sang-Yong Lee, Antonio Ortega "A Novel Approach Of Image Compression In Digital Cameras With A Bayer Color Filter Array," *International Conference on Image Processing, 2001.* Proceedings. pp.482 - 485 vol.3, 2001.

[3] Oluwayomi Adamo, Saraju P. Mohanty, Elias Kougianos, "VLSI Architecture and FPGA Prototyping of a Digital Camera for Image Security and Authentication," *IEEE Region 5 Conference*, San Antonio, TX, USA, pp.154 – 158, Apr. 2006.

[4] Junqing Chen, Kartik Venkataraman, Dmitry Bakin, Brian Rodricks, Robert Gravelle, Pravin Rao, Yongshen Ni, "Digital Camera Imaging System Simulation," *IEEE Transactions on Electron Devices,* vol.56(11), pp.2496 – 2505, Nov. 2009.

[5] Y. Nishikawa, S. Kawahito, T. Inoue, "A parallel image compression system for high-speed cameras," *IEEE International* Workshop on Imaging Systems and Techniques, pp. 53 - 57 May 2005.

[6] Gregory K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, vol. 38(I), Feb. 1992.

[7] Digital Compression and Coding of Continuous tone Still Images, Part 1, Requirements and Guidelines. ISO/IEC JTC 1 Draft International Standard T.81, 09/1992.

[8] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 2nd.Ed.,Prentice Hall, 2002.

[9] S. A. White, "Applications of distributed arithmetic to digital signal processing: a tutorial review," *IEEE ASSP Magazine,* vol.6, no.3, pp.4-19, Jul.1989.

[10] M.-T. Sun, T.-C. Chen, A.M. Gottlieb, ''VLSI Implementation of a 16x16 Discrete Cosine Transform," *IEEE Transactions on Circuits and Systems*, vol.36, no. 4, pp. 610 – 617, Apr. 1989.

[11] A. Shams, A. Chidanandan, W. Pan, and M. Bayoumi, "NEDA: A low power high throughput DCT architecture," *IEEE Transactions on Signal Processing,* vol.54(3), Mar. 2006.

[12] Peng Chungan, Cao Xixin, Yu Dunshan, Zhang Xing, "A 250MHz optimized distributed architecture of 2D 8x8 DCT," *7th International Conference on ASIC,* pp. 189 – 192, Oct. 2007.

[13] Luciano Volcan Agostini, Ivan Saraiva Silva and Sergio Bampi, "Multiplierless and fully pipelined JPEG compression soft IP targeting FPGAs," Microprocessors and Microsystems, vol. 31(8), 3 pp.487-497, Dec. 2007.

[14] Zhang Qihui, Chen Jianghua, Zhang Shaohui and Meng Nan, "A VLSI Implementation of Pipelined JPEG Encoder for Grayscale Images," *International Symposium on Signals, Circuits and Systems, ISSCS 2009,* pp.1-4 Jul.2009.

[15] M. Kovac, N. Ranganathan, "JAGUAR: A Fully Pipelined VLSI Architecture for JPEG Image Compression Standard," *Proceedings of the IEEE*, vol.83, no.2, pp. 247-258,Feb.1995.

[16] M. Kovac, N. Ranganathan, M. Zagar, "A prototype VLSI chip architecture for JPEG image compression," *Proceedings European Design and Test Conference, 1995. ED&TC*, pp.2-6. Mar.1995.

[17] B. G. Shedock, A. Nagpal and D. M. Monro, "A Model for JPEG Quantization," *1994 International Symposium on Speech, Image Processing and Neural Networks*, Hong Kong, vol.1,pp.176 - 179,13-16 April 1994.

[18] Long- Wen Chang and Ching- Yang Wang and Shiuh-Ming Lee "Designing JPEG Quantization Tables Based On Human Visual System," Proceedings. *1999 International Conference on Image Processing*, ICIP 99, pp.376 - 380 vol.2, Oct.1999.

[19] L.F. Costa and A.C.P. Veiga, "Identification Of The Best Quantization Table Using Genetic Algorithms," *IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2005. PACRIM*, pp.570 – 573, Aug.2005.

[20] Bruna Arcangelo and Mancuso Massimo, "JPEG Compression Factor Control: A New Algorithm," *International Conference on Consumer Electronics*, ICCE, pp. 206 – 207, Jun-Jul.2001.