http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (7): 1376-1382, 2010
ISSN 1812-5638

An Optimized Pipelined Architecture of LBT for JPEG XR Encoding

Guanghui Ren, Xiaokai Hu, Gangyi Wang and Jinglong Wu
School of Electronics and Information Technology, Harbin Institute of Technology,
Harbin, Heilongjiang 150001, China

Abstract: JPEG XR is a new image coding standard that achieves high compression performance and low complexity. In this study, we propose a novel architecture of Lapped Biorthogonal Transform (LBT) for JPEG XR encoding using an optimized Quadriphase-RAM (QRAM) and a fully pipelined LBT module. In the previous architectures, the whole image data are stored in the main memory so that the waste of memory and the delay for the whole image acquisition must be inevitable. The proposed QRAM minimizes the memory size and improves the speed of image processing and the fully pipelined LBT module achieves higher throughput than that of the previous architectures. Results show the proposed architecture which achieves 121.2 M pixel/sec at 125 MHz can be used for the real-time JPEG XR encoding.

Key words: QRAM, full pipeline, pre-filter, FCT, FPGA

INTRODUCTION

With the rapid development of wireless mobile communication technology, handheld devices (mobile phones, digital cameras, etc.) must provide better multimedia service which requires image compression technology with higher compression rate and lower computation cost. Among the numerous transform algorithms of the image coding standard, the Discrete Cosine Transform (DCT) applied to JPEG coding standard is the most successful transform algorithm so far. However, DCT has inherent shortcomings of blocking effect in the case of high compression rate (Abboud, 2006). JPEG2000 coding standard proposed subsequently improves subsequently the quality of the compressed image (Haseeb and Khalifa, 2006) and its core transform algorithm is the discrete wavelet transform, but it is not favored by the mainstream consumer market due to the high complexity.

JPEG XR (Srinivasan *et al*., 2007) is a new still image coding standard, its core transform algorithm is the lapped biorthogonal transform (LBT), which not only achieves good compression performance twice as high as the JPEG coding standard, but also has advantage over JPEG2000, since it requires much less computational cost than JPEG2000 with slight degradation of image quality. Therefore, the study of LBT is of great significance.

In this study, an optimized pipelined architecture of LBT for JPEG XR encoding is proposed. Some hardware architectures have been proposed (Pan *et al*., 2008; Hattori *et al*., 2009). In these architectures, a frame of the whole image data is stored in the main memory from which the LBT module acquires the image data. So the waste of memory and the delay for the whole image acquisition must be inevitable. Due to this problem, we reconstruct the main memory to Quadriphase-RAM (QRAM) which minimizes the memory size and parallels the processes of the image acquisition and the LBT module without delay. A high-throughput pipelined architecture proposed (Hattori *et al*., 2009), which pipelines all processes of JPEG XR encoding, is bottlenecked by the LBT module which achieve 0.8 pixel/cycle, since the throughput of the other modules of JPEG XR encoding are 1 pixel/cycle. Cooperating with QRAM, fully pipelined architecture proposed in this paper achieves 0.9697 pixel/cycle which is higher than that of the related works.

LBT FOR JPEG XR ENCODING

In order to deal with the problem of blocking effect, the lapped transform firstly proposed by Cassereau in 1985 imposed a long filter which across the adjacent blocks to deal with the correlation between the edge data of the adjacent blocks. Typical lapped transform are based on DCT and then further developed into two categories: first is the frequency-domain lapped transform after DCT, called the post-process lapped transform; second is the time-domain lapped transform before DCT, called pre-process lapped transform.

As post-process, a lapped orthogonal transform (LOT) is proposed based on Cassereau (Malvar and Staelin, 1989), LOT can obviously eliminate the blocking

Corresponding Author: Guanghui Ren, School of Electronics and Information Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

effect of the reconstructed image, two main reason is that: 1) the length of the basis function of LOT is greater than the size of the signal block; 2) the basis function of LOT tends to zero smoothly on the boundary, but not absolutely tends to zeros, H.S.Malvar proposed lapped biorthogonal transform (LBT) after improving the LOT (Malvar, 1998).

As pre-process, T.D.Tran proposed the time-domain lapped transform (TDLT) to which the property of block diagonal matrix is employed (Tran *et al*., 2003). From the perspective of the filter, in the decomposition side, the operator P in TDLT is equivalent to do the pre-filtering to the edge of the adjacent block data; in the reconstruction side, the operator P is equivalent to do the post-filtering to the edge of the adjacent block data.

The coding flow of JPEG XR which has some similarity with JPEG coding flow consists of the following steps: color conversion, Lapped Biorthogonal Transform (LBT) (Tu *et al*., 2008), quantization, inter-block prediction, adaptive coefficient scanning and entropy coding, code stream organization. JPEG XR coding standard employs LBT proposed by Tran *et al*. (2003) as the core transformation algorithm. The flow of LBT in JPEG XR is show in Fig. 1 and can be separated into two stages: 1) first pre-filter and first forward core transform (FCT) ; 2) second pre-filter and second FCT. Before LBT, the image data are firstly tiled and then divided into 16×16 macroblock (MB), the bigger size of tile we chooses, the bigger image size after compression are, as discussed (Pan *et al*., 2008). Pre-filter resolve the problem of blocking effect and FCT transforms the image data to the frequency field.

Pre-filter process will be done before FCT process and there are three modes of optional pre-filter process that we can choose non-overlapping, one-overlapping or two-overlapping operation. Non-overlapping operation is used for fast coding and will generate blocking effect at high compression rate like DCT in JPEG coding standard. One-overlapping operation and two-overlapping operation have higher compression quality without blocking effect. Figure 1 two-overlapping operation is selected.

In the first stage, each macroblock is divided into 16 blocks. Pre-filter process employs pre4x4 filtering to 4×4 data acquired from the edge of four blocks that don't locate at the border of the image and employs pre4 filtering to 2×4 or 4×2 data acquired from the edge of two blocks that locate at the border of the image. FCT process is applied to the 4×4 blocks as shown in Fig. 1. After the first stage process the data in the 4×4 block are decomposed into 1 DC coefficient and 15 HP Coefficients. In the second stage, pre-filter process and FCT process are both applied to the block which is composed of 16 DC Coefficients gathered from 16 blocks in the macroblock. After the second stage, the data in the 4×4 block are decomposed into 1 DC coefficient and 15 LP Coefficients. As a result of two stages, the data in the macroblock are decomposed into 1 DC coefficient, 15 LP Coefficients and 240 HP coefficients.

A separate 2-D transform is typically implemented by doing 1-D transform on each row of the 2-D data, then doing 1-D transform on each column of the 2-D data. A separate 2-D transform can be defined as the following equation:

$$Y = T_1 X T_2' \quad (1)$$

where, T_1 and T_2, respectively represent the row and column transform matrix. An equivalent matrix is generated by Kronecker product (Van Loan, 2000) that doing the 2-D transform in another separate way. After reordering the data matrix X, the transform is given as follow:

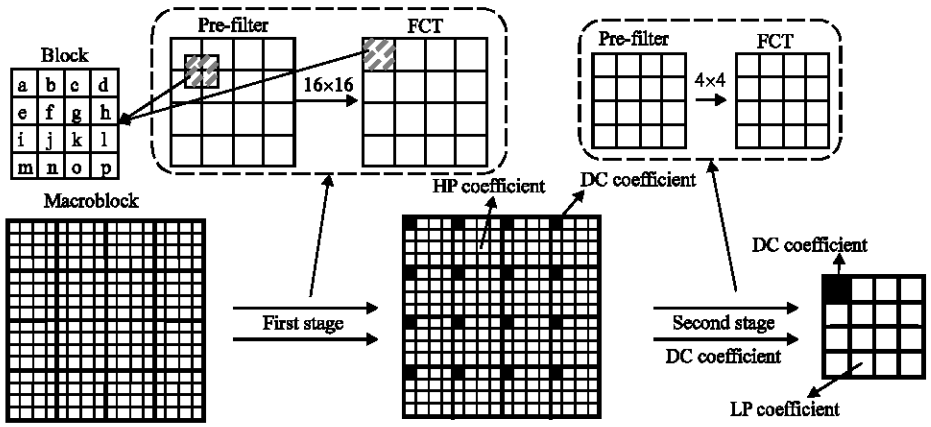

Fig. 1: The flow of LBT in JPEG XR

$$y = Jx \quad (2)$$
$$J = \text{Kron}(T_1, T_2)$$

Furthermore in JPEG XR the row and column 1-D operation can be decomposed into some elementary 2-point operation, so the 2-D transform of pre-filter and FCT is implemented by a cascade of Kronecker product of the elementary 2-point operation of the corresponding 1-D operation. The order of the data in each block is denoted as {a, b, c, d, ... , m, n, o, p} as shown in Fig. 1.

Five steps of pre-filter process are given as follows: (1) Hadamard transform Kron (T_h, T_h): Kronecker product of two 2-point hadamard transform operations (T_h), in the position of {a, d, m, p}, {b, c, n, o}, {e, h, i, l}, {f, g, j, k}; 2) 2-point scaling operation (T_s), in the position of {a, p}, {b, o}, {e, l}, {f, k}; 3) 2-point rotation operation (T_r), in the position of {n, m}, {j, i}, {h, d}, {g, c}; 4) Toddodd rotation Kron (T_r, T_r): Kronecker product of two 2-point rotation operations (T_r), in the position of {k, l, o, p}; 5) Inverse hadamard transform Kron(T_h, T_h): Kronecker product of two 2-point inverse hadamard transform operations (T_h), in the position of {a, d, m, p}, {b, c, n, o}, {e, h, i, l}, {f, g, j, k}.

Four steps of FCT are given as follows: (1) Hadamard transform Kron (T_h , T_h): Kronecker product of two 2-point hadamard transform operations (T_h), in the position of {a, d, m, p}, {b, c, n, o}, {e, h, i, l}, {f, g, j, k}; 2) Hadamard transform Kron (T_h, T_h): Kronecker product of two 2-point hadamard transform operations (T_h), in the position of {a, d, m, p}; 3) Todd rotation Kron (T_h, T_r): Kronecker product of one 2-point hadamard transform operations (T_h) and one 2-point rotation operation (T_r), in the position of {c, d, g, h}, {i, m, j, n}; 4) Toddodd rotation Kron (T_r, T_r): Kronecker product of two 2-point rotation operations (T_r), in the position of {k, l, o, p}.

where, T_h and T_r are given as the following equations:

$$T_h = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix},\quad T_r = \frac{1}{\sqrt{4+2\sqrt{2}}}\begin{bmatrix}1+\sqrt{2} & 1\\ 1 & -(1+\sqrt{2})\end{bmatrix} \quad (3)$$

PROPOSED PIPELINED ARCHITECTURE OF LBT

LBT in JPEG XR is divided into two parts: Pre-filter and FCT that can be designed individually. Lifting structures which only involve integer additions and shifts operations are designed to approximate the transform matrix (Liang and Tran, 2001) and both processes are computing with the 4×4 block without feedback information, so we pipeline two processes of the LBT module. Furthermore we optimize the main memory to QRAM. In this paper, an optimized pipelined architecture of LBT for JPEG XR encoding is proposed.

Pipelined Pre-filter: Pre-filter process is implemented by a cascade of hadamard transform, 2-point scaling operation, 2-point scaling operation, Toddodd rotation and inverse hadamard transform, the lifting structures of which are given by the following. The lifting structure of hadamard transform, as shown in Fig. 2a, can be divided into 2 stages that implemented by pipeline in 2 clock cycles; The lifting structure of 2-point scaling operation, as shown in Fig. 2b, can be divided into 3 stages that implemented by pipeline in 3 clock cycles; The lifting structure of 2-point rotation operation, as shown in Fig. 2c, can be divided into 2 stages that implemented by pipeline in 2 clock cycles; The lifting structure of Toddodd rotation is different with others, as shown in Fig. 2d, it can be divided into 6 states that implemented by finite state machine in 3 clock cycles, because it's only used once in the whole process.

The whole pre-filter process can be implemented by pipeline in three stages. The first stage includes step 1, 2 and 3 of the pre-filter process, the second stage includes step 4 of the pre-filter process, the third stage includes step 5 of the pre-filter process. As shown in Fig. 3, the full pipelined architecture of pre-filter is proposed; First each step is implemented by pipeline in the corresponding stage, in the top module three stages of the pre-filter process are implemented by pipeline. This architecture can archive the coding of a 4×4 block in 10 clock cycles.

Pipelined FCT: FCT process is implemented by a cascade of hadamard transform, Todd rotation, Toddodd rotation, the lifting structures of which are given by the following. The lifting structures of hadamard transform and Toddodd rotation are similar with that in the pre-filter process. The lifting structure of Todd rotation, as shown in Fig. 4, can be divided into 4 stages that implemented by pipeline in 4 clock cycles.

The whole FCT process can be implemented by pipeline in two stages. The first stage includes step 1 of the pre-filter process, the second stage includes step 2, 3 and 4 of the FCT process. As shown in Fig. 5, the full pipelined architecture of FCT is proposed; First each step is implemented by pipeline in the corresponding stage, in the top module two stages of the FCT process are implemented by pipeline. This architecture can archive the coding of a 4×4 block in 6 clock cycles.

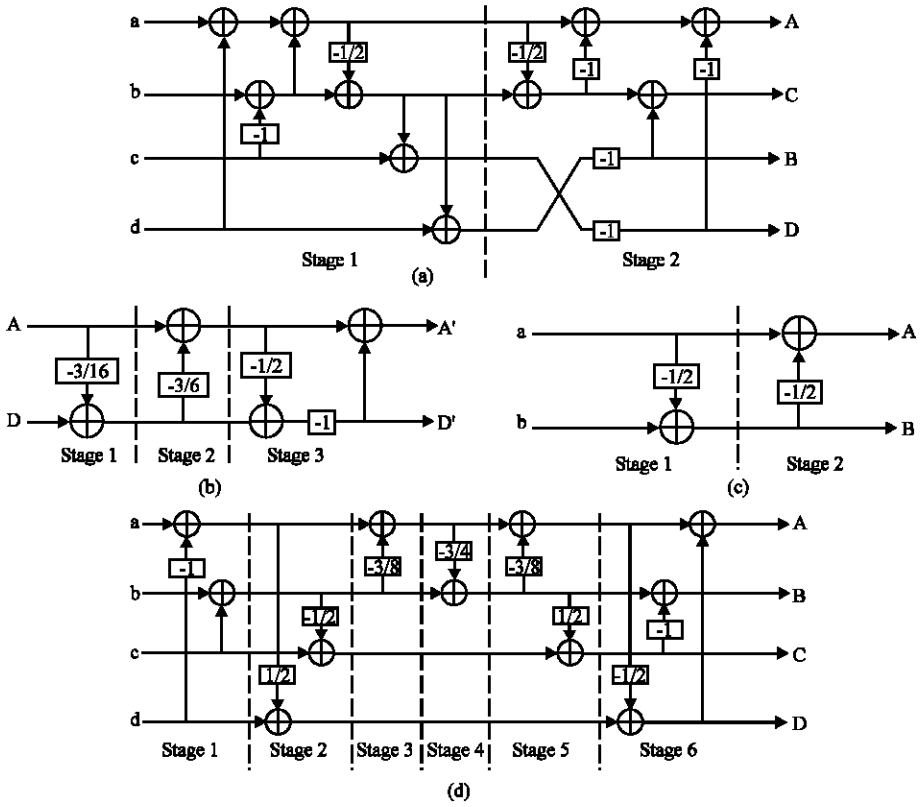

Fig. 2: The lifting structure of the operation: (a) hadamard transform; (b) 2-point scaling; (3) 2-point rotation; (4) Toddodd rotation

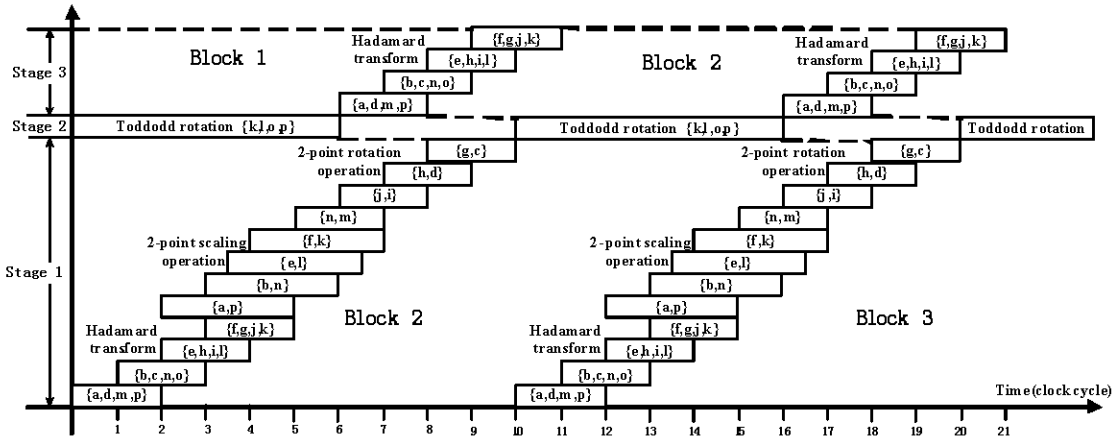

Fig. 3: Full pipelined architecture of pre-filter

In the pipelined architecture, the coding speed is bottlenecked by the stage which has more time-consuming than any other stage. Stage 1 in the pre-filter process which needs 10 clock cycles and stage 2 in the FCT process which needs 6 clock cycles are those bottleneck stages. So, the throughput of the pre-filter process is 16/10≈1.6 pixel/cycle and the throughput of the FCT process is 16/6˜2.7 pixel/cycle.

QRAM: In the real-time JPEG XR encoding, the input analog image data acquired from the digital camera are converted to the digital data by video A/D converter and

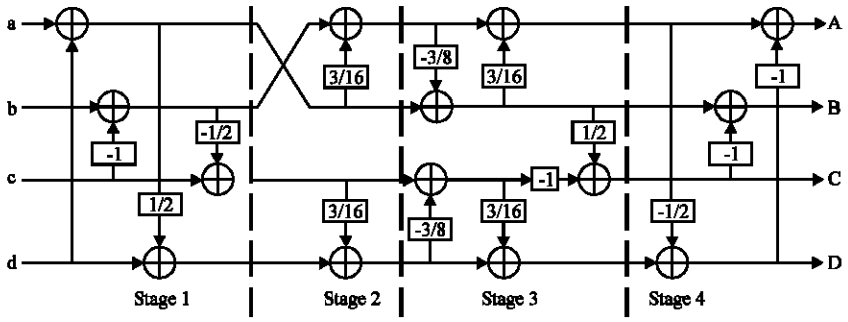

Fig. 4: The lifting structure of Todd rotation

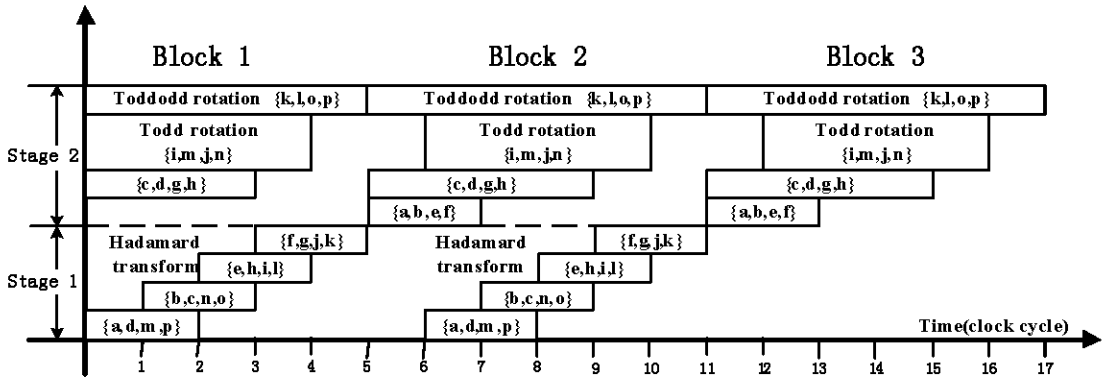

Fig. 5: Full pipelined architecture of FCT

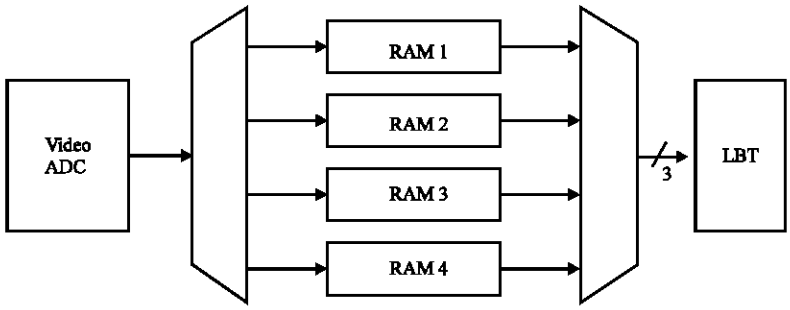

Fig. 6: The architecture of QRAM

the digital image data are fed to the LBT module in raster order. Therefore, we should first store the image data in Onchip-RAM or SRAM row by row. In the previous architectures, the whole image data are stored in the main memory and the encoder doesn't start until a frame of image data is stored. So, the waste of memory and the delay for the whole image acquisition must be inevitable. In this study, Quadriphase-RAM (QRAM) architecture used for the image data acquisition, storing and transfer is designed as shown in Fig. 6 and each RAM stores 4 rows of the image data. the image data acquisition is applied to one RAM, while the image data in the other three RAM are fed to the LBT module, where two rows of one RAM data and two rows of the adjacent RAM are fed to pre-filter process that computes with 4×4 block, four rows of one RAM data are fed to FCT process that computes with 4×4 block. As discussed ahead, the image data acquisition of which the throughput is 1pixel/cycle decides the speed of the whole LBT encoding. The architecture of LBT encoder is given in Fig. 7, where w and h indicate the width and height of the image. After adding the delay of the last several processes, the throughput of the whole LBT module is computed as the following equation:

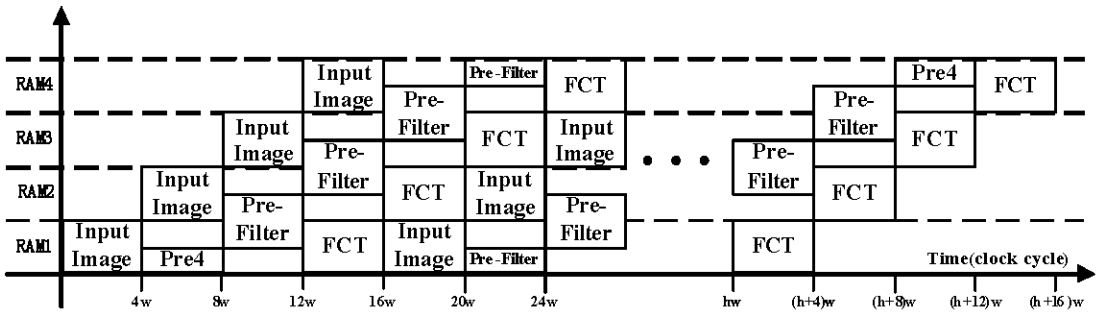

Fig. 7: The architecture of LBT encoder

Table 1: Results of the proposed architecture

Module name	ALUTs	Logic register	Memory (bits)	Maximum frequency (MHz)	Throughput (pixel/cycle)
Pre-filter	1207	1788	--	112.56	1.6
FCT	1552	1720	--	107.25	2.7
QRAM	--	--	256w	--	0.9697

Table 2: Comparison between the previous architecture and the proposed architecture

Architecture	Memory (bits)	Throughput (pixel/cycle)
Hattori *et al.* (2009)	h×w×16	0.8
Proposed	4×4×w×16	0.9697

$$T_{LBT} = \frac{h\times w}{h\times w+16w} = \frac{h}{h+16} \quad (4)$$

For the grey image with the resolution of 512×512, $T_{LBT} \approx$ 0.9697 pixel/cycle, that is 121.2 M pixel/cycle at 125MHz clock frequency.

Evaluation: In order to evaluate the proposed architecture, we implemented the LBT encoder based on FPGA using VHDL language. The design is synthesized by ALTERA QUARTUSII using the StratixII series. The result of the proposed architecture is summarized in Table 1. The results of adaptive look-up tables (ALUTs), logic register and maximum frequency are given by synthesizing the corresponding module. The result of throughput is analyzed 3 based on the grey image with the resolution of 512×512.

The comparison between the previous architecture and the proposed architecture is given in Table 2. A high-throughput pipelined architecture for JPEG XR encoding proposed (Hattori *et al*., 2009) achieves higher throughput than the architecture in the previous work (Pan *et al*., 2008). However in that architecture the input image data is wholly stored in the main memory with the delay of the whole image acquisition and then the image data are transferred from the main memory to the LBT module through the bus MB by MB. So, the size for the main memory is h×w×16 bits, our architecture only needs 4×4×w×16 bits that is much smaller. And that architecture which pipelines all processes of JPEG XR encoding, is bottlenecked by the LBT module which achieve 0.8 pixel/cycle, since the throughput of the other modules in JPEG XR encoder is 1 pixel/cycle. Cooperating with QRAM, our fully pipelined LBT module achieves higher throughput of 0.9697 pixel/cycle than that of 0.8 pixel/cycle (Hattori *et al*., 2009).

CONCLUSIONS

An optimized pipelined architecture of LBT for JPEG XR is proposed in this study. We optimize the memory structure to Quadriphase-RAM which minimizes the memory size and parallels the processes of the image acquisition and LBT without delay and fully pipelines the pre-filter process and the FCT process in LBT module. Compared with the previous architectures, our architecture needs smaller memory size and achieve higher throughput. Therefore the proposed architecture which achieves 121.2M pixel/sec at 125MHz can perform better in the real-time JPEG XR encoding.

REFERENCES

Abboud, I., 2006. Deblocking in BDCT image and video coding using a simple and effective method. Inform. Technol. J., 5: 422-426.

Haseeb, S. and O.O. Khalifa, 2006. Comparative performance analysis of image compression by JPEG 2000: A case study on medical images. Inform. Technol. J., 5: 35-39.

Hattori, K., H. Tsutsui, H. Ochi and Y. Nakamura, 2009. A high-throughput pipelined architecture for JPEG XR encoding. Proceedings of the IEEE/ACM/IFIP 7th Workshop on Embedded Systems for Real-Time Multimedia, Oct. 15-16, IEEE Computer Society, Grenoble, France, pp: 9-17.

Liang, J. and T.D. Tran, 2001. Fast multiplierless approximations of the DCT with the lifting scheme. IEEE Trans. Signal Process., 49: 3032-3044.

Malvar, H.S. and D.H. Staelin, 1989. The LOT: Transform coding without blocking effects. IEEE Trans. Acoust. Speech Signal Process., 37: 553-559.

Malvar, H.S., 1998. Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts. IEEE Trans. Signal Process., 46: 1043-1053.

Pan, C.H., C.Y. Chien, W.M. Chao, S.C. Huang and L.G. Chen, 2008. Architecture design of full HD JPEG XR encoder for digital photography applications. IEEE Trans. Consumer Electron., 54: 963-971.

Srinivasan, S., C. Tu, S.L. Regunathan and G.J. Sullivan, 2007. HD Photo: A new image coding technology for digital photography. Proc. SPIE, 6696: 66960A.1-66960A.19.

Tran, T.D., J. Liang and C.J. Tu, 2003. Lapped transform via time-domain pre- and post-filtering. IEEE Trans. Signal Process., 51: 1557-1571.

Tu, C., S. Srinivasan, G.J. Sullivan, S. Regunathan and H.S. Malvar, 2008. Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR/HD Photo. Proc. SPIE, 70730: 70730C-70730C-12.

Van Loan, C.F., 2000. The ubiquitous Kronecker product. J. Comput. Applied Math., 123: 85-100.