MEMORY EFFICIENT JPEG 2000 ARCHITECTURE WITH STRIPE PIPELINE SCHEME

Hung-Chi Fang, Yu-Wei Chang, Chih-Chi Cheng, Chun-Chia Chen, and Liang-Gee Chen

DSP/IC Design Lab, Graduate of Electronics Engineering and
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
Email:{honchi,wayne,ctchb,chunchia,lgchen}@video.ee.ntu.edu.tw

ABSTRACT

Memory issue is the most critical problem for a high performance JPEG 2000 architecture. The tile memory occupies more than 50% of area in conventional JPEG 2000 architectures. To solve this problem, we propose a stripe pipeline scheme. For this scheme, a Level Switch Discrete Wavelet Transform (LS-DWT) and a Code-block Switch Embedded Block Coding (CS-EBC) are proposed. With small additional memory, the LS-DWT and the CS-EBC can process multiple levels and code-blocks in parallel by an interleaved scheme. As a result of above techniques, the overall memory requirements of the proposed architecture can be reduced to only 8.5% comparing with conventional architectures.

1. INTRODUCTION

JPEG 2000 [1]-[4] is well-known for its excellent coding performance and numerous features [5], such as Region Of Interest (ROI), various kinds of scalabilities, error resilience, and so on. All these powerful tools can be provided by a unified algorithm in a single JPEG 2000 codestream. JPEG 2000 adopts the Discrete Wavelet Transform (DWT) and the Embedded Block Coding with Optimized Truncation (EBCOT) [6] as its core coding algorithms, which is totally different from the algorithms of JPEG [7]. By use of the new coding tools, JPEG 2000 outperforms JPEG by more than 2 *dB* in Peak Signal-to-Noise Ratio (PSNR) [5]. However, the complexity of JPEG 2000 is much higher than that of JPEG. Thus, hardware implementation is a must for real-time JPEG 2000 applications. In this paper, we proposed a stripe pipeline scheduling scheme to reduce the memory requirements of the JPEG 2000 encoder. The stripe pipeline scheduling scheme can reduce the on-chip memory size, which occupies over 50% of chip area in conventional JPEG 2000 encoder, to only 8.5% comparing with conventional architectures.

Since hardware acceleration is a must for real-time JPEG 2000 applications, many architecture for JPEG 2000 has been proposed [8]-[14] All the above architectures focus on how to overcome the computation complexity, especially the Embedded Block Coding (EBC). The solutions can be classified into two categories. The first one is to use multiple EBC engines [8]-[11], which process multiple code-blocks in parallel. The second one is to increase the processing rate of the EBC engine [12]-[14] by processing multiple bit-planes in parallel. The major disadvantage of using multiple EBC engines is that this method needs to use multiple code-block memory, which size is commonly 6 *KB* ($64 \times 64 \times 12$) for a code-block. Fang et al. [15] proposed a parallel EBC architecture, which can greatly increase the processing rate of the EBC while maintain similar hardware cost with architectures that use single EBC engine. On the other hand, memory issues of the DWT are also key factors of a JPEG 2000 design. Block-based scan for DWT [12] [16] is proposed to eliminate the use of tile memory, which size is commonly 96 *KB* ($256 \times 256 \times 12$), at the cost of the increase of memory bandwidth. Although the tile memory is eliminated, the scan order of the block-based scan is not optimized such that the memory requirements are still too high. Therefore, the hardware cost of JPEG 2000 is still too high to let JPEG 2000 take place of JPEG regardless of the large coding gain by use of JPEG 2000.

In this paper, we proposed a stripe pipeline scheme for the DWT and the EBC to solve the above problems. The stripe pipeline scheme takes the throughputs and the dataflows of the DWT and the EBC into joint consideration. The main idea is to match the throughputs and the dataflows of the two modules so that the size of local buffers between the two modules is minimized. Thus, a Level Switch DWT (LS-DWT) and a Code-block Switch EBC (CS-EBC) are proposed. The the CS-EBC can process 13 code-blocks in parallel, and LS-DWT can accomplish multi-level two-dimensional DWT concurrently. As a result of the stripe pipeline scheme, the memory requirements are reduce to only 8.5% comparing with conventional architectures.

This paper is organized as follows. Section 2 gives an overview of JPEG 2000. The proposed architecture is described in Sec. 3. In Section 4, implementation results are compared with the state-of-the-art JPEG 2000 architectures. Finally, Section 5 summaries this paper.

2. JPEG 2000 OVERVIEW

In this section, we'll briefly describe the JPEG 2000 coding system, especially the dataflow. Figure 1 shows the JPEG 2000 coding system. It consists of the Discrete Wavelet Transform (DWT), the Quantization (Q), the Embedded Block Coding with Optimized Truncation (EBCOT), and the Rate Control (RC). The EBCOT is further divided into the Embedded Block Coding (EBC) and the Rate-Distortion Optimization (RDO). Among all the functional blocks, the DWT and the EBC are the most critical ones for their special dataflows and high computational complexities.

Figure 2 shows how an image is decomposed into abstract levels, which include tile, subband, code-block, bit-plane, and coding pass. The original image is partitioned into several rectangular tiles, which are independently coded. The DWT decomposes a tile into N_L levels. Except for the N_L-th level that has only LL band, each level has three subband, which are the HL, LH, and

This work was supported in part by National Science Council, Republic of China, under the grant number 91-2215-E-002-015 and in part by the MediaTek Fellowship.

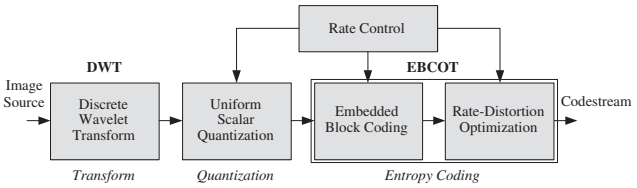

Fig. 1. JPEG 2000 coding system. It adopts the DWT as the transform algorithm and the EBCOT as the entropy coding algorithm.

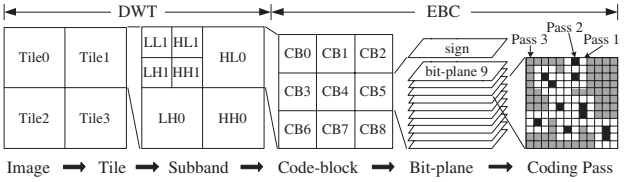

Fig. 2. Decomposition of an image into abstract levels. These abstract levels include tile, subband, code-block, bit-plane, and coding pass.

HH bands. The HL band is high-pass filtered horizontally, and then low-pass filtered vertically, and the LH band is exactly opposite to HL band. Both directions of the HH band are high-pass filtered, and the LL band are low-pass filtered in both directions. In general, the DWT coefficients co-located at each subband are generated consequently. Figure 3 shows an example of a 8×8 tile. Each circle represents a DWT coefficient, and the number within the circle indicates the order it is generated. The order of the DWT coefficients generated within each subband depends on the scan order of the DWT engine. However, the co-located DWT coefficients are always generated consequently.

For the EBC, each subband is further partitioned into code-blocks. The DWT coefficients in the code-block are sign-magnitude represented. The code-block is processed in a bit-plane by bit-plane manner, from the Most Significant Bit (MSB) bit-plane to the Least Significant Bit (LSB) bit-plane. Every bit-plane has three coding passes called Pass 1, Pass 2, and Pass 3. A special coding order called stripe scan is used within any coding pass. A stripe is a $N \times 4$ rectangle, where N is the width of the code-block. The coefficients are scanned stripe by stripe from top to bottom in a coding pass, and column by column from left to right in a stripe.

3. PROPOSED ARCHITECTURE

In this section, the proposed JPEG 2000 architecture is presented. The block diagram of the architecture is shown in Fig. 4. Seven Stripe Buffers (SB) are used for the stripe pipeline, each buffer has 256×11 bits. The Level Switch DWT (LS-DWT) generates 256 DWT coefficients for each subband at every pipeline stage. The resulting coefficients are stored in the SBs, and then processed by the Code-block Switch EBC (CS-EBC).

3.1. Stripe Pipeline Scheme

In this section, a stripe pipeline scheme for JPEG 2000 is proposed. The key concept is to design a scheduling that can minimize the

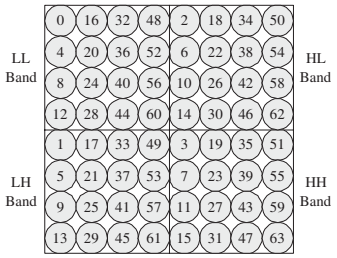

Fig. 3. Output order of DWT coefficients. The co-located DWT coefficients in all subbands are generated consequently.

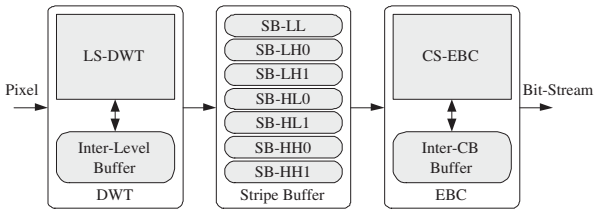

Fig. 4. Proposed JPEG 2000 architecture. No tile memory is required in the proposed architecture.

memory requirement while maintain reasonable complexity and overhead. As described in Sec. 2, the memory issues are arisen from the mismatch between output dataflow of the DWT and the input scan order of the EBC. The mismatch can be solved by using a buffer between the two modules. In conventional architecture [10], the whole tile and three code-block are buffered. Wu et al. [16] proposed a Quad Code-Block (QCB) scheduling scheme that reduces the memory requirements to $\frac{1}{4}$ tile and six code-blocks. However, the memory requirements are still too high. The proposed stripe pipeline scheme can fully eliminate the use of tile memory and code-block memory. It only requires stripe memory, which size is $\frac{1}{16}$ of a code-block memory for a 64×64 code-block.

The detail of the stripe pipeline scheme is shown in Fig. 5. Each rectangle represents a computation state of the LS-DWT or the CS-EBC. The state of LS-DWT is indicated by, for example, $T_k L_i : R_{s-t}\text{-}L$, which means that the LS-DWT is processing left half of the s-th row to the t-th row in the i-th level of the k-th tile. On the other hand, the state of CS-EBC is indicated by, for example, $T_k CB_{s-t} : S_i$, which means that the CS-EBC is processing the i-th stripe of the s-th code-block to the t-th code-block of the k-th tile. The order of execution is from top to bottom. All the computation states require 768 cycles except when the CS-EBC is at $T_k CB_{0-3}$ state, which needs 1024 cycles. In fact, the cycles required in each state is exactly the number of DWT coefficients the CS-EBC must process.

For the stripe pipeline schedule, seven stripe buffers are required as shown in Fig. 4. Each stripe buffer is 256×11 bits, where 11 is the bit-width of a DWT coefficient. While the LS-DWT is writing coefficients into SB-LH0, SB-HL0, and SB-HH0, the CS-EBC is reading coefficients from SB-LH1, SB-HL1, and SB-HH1, or vice versa. When the LS-DWT is processing L_2 or the CS-EBC is processing CB_{0-3}, SB-LL must be accessed by the corresponding module. Therefore, the proposed scheduling

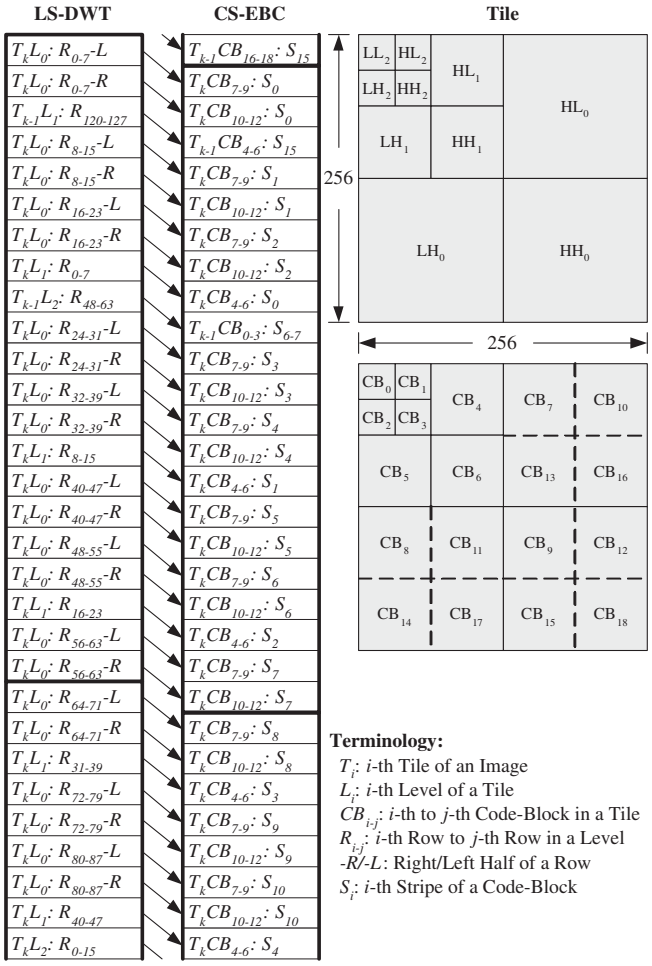

Fig. 5. Stripe pipeline scheme. The LS-DWT and CS-EBC are pipelined at stripe level, and therefore the buffer size is reduced to the same as the stripe size.

only requires about 2.4 Kilo-Bytes (KB) or 1.75 Kilo-Words (KW), equivalently.

3.2. Level Switch DWT

To accomplish the stripe pipeline scheduling described above, a Level Switch DWT (LS-DWT), which can switch between various decomposition levels, is proposed. The block diagram of the LS-DWT is shown in Fig. 6. The filter core is a line-based 2-D DWT architecture. In the following, we will focus on how to extend a general line-based 2-D DWT filter to a LS-DWT by use of inter-level buffer.

There are two kinds of inter-level buffer in the LS-DWT, line buffer for the column 1-D DWT and LL-band buffer for the row 1-D DWT. The line buffer of the LS-DWT is, in fact, the line buffer in conventional line-based 2-D DWT. However, the line buffer for level 1 can not be reused when the LS-DWT is switching to level 2. The same case happens when it is switching to level 3. Thus, each level should have its own line buffer. For the line-based 9/7 filter,

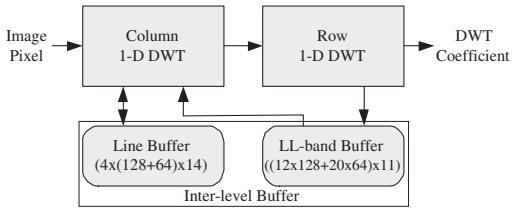

Fig. 6. Block diagram of the proposed LS-DWT. The LS-DWT can transform 3 levels in parallel.

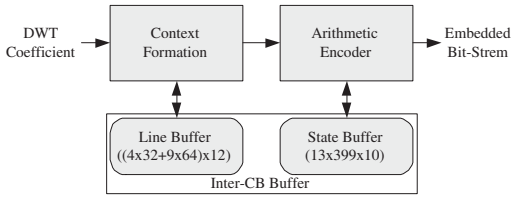

Fig. 7. Block diagram of the proposed CS-EBC. The CS-EBC can process 13 code-blocks in parallel.

four lines are required to be buffered for each level. Therefore, the additional requirements of the line buffer is 10572 ($= 4 \times (128 + 64) \times 14$) bits. On the other hand, a LL-band buffer is required to buffer the LL-bands generated in level 1 and level 2. Since the LS-DWT switches whenever 4096 (16 lines for level 1 and 32 lines for level 2) DWT coefficients are generated, the LL-band buffer for each level requires 1024 words. Due to the latency of DWT, output data for one more stripe pipeline stage in column direction must be buffered. Thus, each level should buffer 4 additional lines. Therefore, the memory requirements for the LL-band buffer for level 1 and level 2 are 16896 ($= 12 \times 128 \times 11$) bits and 14080 ($= 20 \times 64 \times 11$) bits, respectively.

3.3. Code-block Switch EBC

In this section, a Code-block Switch EBC (CS-EBC), which can switch among code-blocks at end of any stripe, is proposed for the stripe pipeline. The block diagram of the CS-EBC is shown in Fig. 7. The processing elements are similar to that in [15]. In order to achieve code-block switch function, additional buffer, which is called inter-CB buffer, is required to store the coding status that is required to resume processing the code-block. The CS-EBC is designed to switch at stripe boundary of code-blocks because the size of inter-CB buffer is minimized in this case. Besides the inter-CB buffer, there is almost no overhead to extend the parallel EBC [15] to the CS-EBC.

The inter-CB buffer can be further divided into the line buffer for the Context Formation (CF) and the state buffer for the Arithmetic Encoder (AE). The line buffer stores the last row in previous stripe of a code-block. Thus, it requires 8448 ($= 4 \times 32 \times 12 + 9 \times 64 \times 12$) bits for the concurrently processing of 13 code-blocks. Although there are 19 code-blocks in a tile, only 13 of them must be processed concurrently. Because CB_{13-18} were processed after CB_{7-12}, the line buffer for CB_{7-12} can be re-used. The same idea is also applied to the state buffer of the AE. For the AE, the state buffer contains the coding status of the bit-stream and the ta-

Table 1. Memory Requirement Comparison. Assume that the tile size is 256×256 and the code-block size is 64×64.

Architecture	Memory Requirement (*KB*)				
	Tile	CB	DWT	EBC	Total
Amphion [9]	128	48	-	-	176
Andra's [10]	128	24	-	-	152
Wu's [16]	32	48	-	-	80
Proposed	2.4	0	5.1	7.5	15

ble index for each context. The coding status has 56 bits, which comprises of an interval register (A, 16 bits), a code register (C, 28 bits), a counter (C_T, 4 bits), and an output buffer (B, 8 bits). The table index has 231 bits, which includes index for Pass 1 (14 contexts, 14×7 bits), Pass 2 (3 contexts, 3×7 bits), and Pass 3 (16 contexts, 16×7 bits). The memory requirements for a bit-plane is 399 ($= 3 \times 56 + 231$) bits, and therefore it requires 3990 bits for a code-block with 10 magnitude bit-planes. Finally, the size of the state buffer is about 6.3 *KB* ($= 13 \times 3990$ bits).

4. COMPARISON

To show the performance on memory reduction of the proposed stripe pipeline scheme, we compare the memory requirements for DWT coefficients between the DWT and the EBC in various architectures. The memory requirements depends on the specification of the architecture. Without loss of generality, we assume that the tile size is 256×256, and the code-block size is 64×64. The memory requirements of various architectures are shown in Table 1. In this table, memory requirements within the DWT and the EBC are not shown since the scheduling will only affect the tile memory requirements. In Table 1, the Tile column is the memory size for storing DWT coefficients, and the CB column is the code-block memory for the EBC. The DWT column and the EBC column are the additional memory requirements for the DWT and the EBC to support the stripe pipeline schedule. By the table, the memory requirements are reduced to only 8.5% comparing with Amphion's architecture [9] and 18.8% comparing with Wu's architecture [16].

There is another important advantage of the proposed stripe pipeline scheme that the memory requirements are proportional to the square-root of the tile size. On the other hand, memory requirements of conventional architectures are proportional to the tile size. Thus, for tile size of 512×512, the memory requirements of the proposed architecture is only 4.2%.

5. CONCLUSION

In this paper, a memory efficient JPEG 2000 architecture with stripe pipeline scheme is proposed. The stripe pipeline scheme takes the dataflow of the DWT and the EBC into joint consideration. By matching the dataflow of both modules, the tile memory is replaced by the stripe memory with additional memory requirements in the DWT and the EBC. The overall memory requirements are reduced to only 8.5% of the conventional architecture. For the stripe pipeline scheme, a LS-DWT and a CS-EBC are proposed. With small additional memory, the LS-DWT can process multi-level in an interleaved order. The CS-EBC can process 13 code-blocks in parallel. The proposed architecture is suitable for large tile size since its memory requirements are proportional to the square-root of the tile size.

6. REFERENCES

[1] *JPEG 2000 Requirements and Profiles*. ISO/IEC JTC1/SC29/WG1 N1271, Mar. 1999.

[2] *JPEG 2000 Verification Model 7.0 (Technical Description)*. ISO/IEC JTC1/SC29/WG1 N1684, Apr. 2000.

[3] *JPEG 2000 Part I: Final Draft International Standard (ISO/IEC FDIS15444-1)*. ISO/IEC JTC1/SC29/WG1 N1855, Aug. 2000.

[4] D. Taubman and M. Marchellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[5] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Mag.*, vol. 18, no. 5, pp. 36–58, Sept. 2001.

[6] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.

[7] *JPEG: Still Image Data Compression Standard.* W. Pennebaker and J. Mitchell, New York: Van Nostrand Reinhold, 1992.

[8] Alma Technologies. (2002, Oct.) JPEG2K_E. [Online]. Available: http://www.alma-tech.com/

[9] AMPHION. (2002, Oct.) CS6510. [Online]. Available: http://www.amphion.com/cs6510.html

[10] K. Andra, C. Chakrabarti, and T. Acharya, "A high-performance JPEG 2000 architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 3, pp. 209–218, Mar. 2003.

[11] Analog Devices. (2004, Mar.) ADV202. [Online]. Available: http://www.analog.com/

[12] H. Yamauchi, S. Okada, K. Taketa, T. Ohyama, Y. Matsuda, T. Mori, S. Okada, T. Watanabe, Y. Matsuo, Y. Yamada, T. Ichikawa, and Y. Matsushita, "Image processor capable of block-noise-free JPEG2000 compression with 30 frames/s for digital camera applications," in *ISSCC Dig. Tech. Papers*, San Francisco, CA, Feb. 2003, pp. 46–47.

[13] H. Yamauchi, K. Mochizuki, K. Taketa, T. Watanabe, T. Mori, Y. Matsuda, Y. Matsushita, A. Kobayashi, and S. Okada, "A 1440×1080 pixels 30frames/s motion-jpeg2000 codec for hd movie transmission," in *ISSCC Dig. Tech. Papers*, San Francisco, CA, Feb. 2004, pp. 326–327.

[14] H.-C. Fang, C.-T. Huang, Y.-W. Chang, T.-C. Wang, P.-C. Tseng, C.-J. Lian, and L.-G. Chen, "81 MS/s JPEG 2000 single-chip encoder with rate-distortion optimization," in *ISSCC Dig. Tech. Papers*, San Francisco, CA, Feb. 2004, pp. 328–329.

[15] H.-C. Fang, T.-C. Wang, C.-J. Lian, T.-H. Chang, and L.-G. Chen, "High speed memory efficient ebcot architecture for JPEG2000," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, Bangkok, Thailand, May 2003, pp. 736–739.

[16] B.-F. Wu and C.-F. Lin, "Analysis and architecture design for high performance JPEG2000 coprocessor," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, Vancouver, Canada, May 2004, pp. 225–228.