High-Performance JPEG 2000 Encoder With Rate-Distortion Optimization

Hung-Chi Fang, Yu-Wei Chang, Tu-Chih Wang, Chao-Tsung Huang, and Liang-Gee Chen, *Fellow, IEEE*

***Abstract*—An 81 MSamples/s JPEG 2000 single-chip encoder is implemented on 5.5 mm^2 area using 0.25-μm CMOS technology. This IC can losslessly encode HDTV 720p resolution at 30 frames/s in real time. Three techniques are adopted: line-based discrete wavelet transform, parallel embedded block coding, and precompression rate-distortion optimization. The line-based discrete wavelet transform achieves the minimum external memory access, while the internal memory is reduced by a proper memory access scheme. The parallel embedded block coding increases the throughput and reduces the memory bandwidth with similar hardware cost comparing to conventional architectures. By accurately estimating bit rates, the precompression rate-distortion optimization reduces the required computational power and processing time of the embedded block coding since the code-blocks are truncated before compression. Experimental results show that this encoder has the highest throughput with the smallest area compared with other designs in the literature.**

***Index Terms*—Discrete wavelet transform, embedded block coding with optimized truncation, HDTV, image compression, JPEG 2000, rate-distortion optimization.**

I. 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. Fig. 1 shows the functional block diagram of JPEG 2000. JPEG 2000 uses discrete wavelet transform (DWT) [6] as the transform algorithm and embedded block coding with optimized truncation (EBCOT) [7], [8] as the entropy-coding algorithm. EBCOT is a two-tiered algorithm. EBCOT Tier-1 is the embedded block coding (EBC), which uses an adaptive binary arithmetic encoder. EBCOT Tier-2 is called the post-compression rate-distortion optimization (RDO), which truncates the codestream at target bit rate to provide optimal image quality. By use of above new coding tools, JPEG 2000 outperforms JPEG by more than 2 dB in peak signal-to-noise ratio (PSNR) over wide range of bit rates [5]. However, the complexity of JPEG 2000 is much higher than that of JPEG. In average, about 550 operations are required to encode a sample by JPEG 2000 [9]. Thus, it requires 42.5 giga-operation per second (GOP) to encode HDTV 1280 × 720 4:4:4 at 30 frames/s (fps). Therefore, dedicated hardware is the only possible solution to achieve high speed JPEG 2000 encoding for real-time applications.

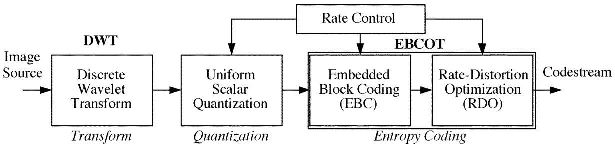

Rate Control
DWT
EBCOT
Image Source
Discrete Wavelet Transform
Uniform Scalar Quantization
Embedded Block Coding (EBC)
Rate-Distortion Optimization (RDO)
Codestream
Transform
Quantization
Entropy Coding

Fig. 1. Functional block diagram of the JPEG 2000 encoder. JPEG 2000 adopts DWT and EBCOT as its transform and entropy coding algorithms, respectively. EBCOT is a two-tiered algorithm, which contains EBC and RDO.

There are three critical issues to design a high throughput JPEG 2000 encoder. First, the DWT requires high memory bandwidth and enormous computational power. Second, the EBC requires extremely complicated control and sequential processing. Third, the RDO requires a large memory for storing the lossless code-stream and rate-distortion information. Moreover, it causes computational power waste since the EBC must losslessly encode all the code-blocks regardless of the compression ratio. All of the above issues require high operating frequency, huge memory size, and high memory bandwidth for chip implementation of a high-speed JPEG 2000 encoder.

Some JPEG 2000 designs have been reported in the literature [10]–[14]. However, they suffer from high operating frequency or large area. The designs in [11], [12], and [14] operate at frequency higher than 150 MHz to provide the throughput of about 50 MSamples/s (MS/s). On the other hand, the design in [12] occupies 289 mm^2 to achieve about 50 MS/s throughput. In order to lower the operating frequency, Yamauchi [13] proposed an architecture, which achieves 21 MS/s at 27 MHz operating frequency. However, the area, which is 25 mm^2, is still too large.

All the above problems come from the EBC. In the above designs, two techniques are adopted to increase the throughput of the EBC: increasing operating frequency and duplicating the EBC module. Increasing operating frequency is the simplest method to increase the throughput without increasing hardware cost. However, this technique directly leads to high power consumption. In order to avoid high power consumption, others

Manuscript received July 12, 2004; revised September 8, 2005. This work was supported in part by the MOE Program for Promoting Academic Excellence of Universities under Grant 89E-FA06-2-4-8, in part by the National Science Council, Taiwan, R.O.C., under Grant 91-2215-E-002-015, and in part by the MediaTek Fellowship. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Suh-Yin Lee.

H.-C. Fang was with Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan. He is now with the MediaTek, Inc., Hsinchu 300, Taiwan, R.O.C (e-mail: honchi@video.ee.ntu.edu.tw).

Y-W. Chang and L.-G. Chen are with DSP/IC Design Lab, Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: wayne@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

T.-C. Wang was with Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. He is now with the Chin Fong Machine Industrial Company, Ltd., Chang Hua 500, Taiwan, R.O.C. (e-mail: eric@video.ee.ntu.edu.tw).

C.-T. Huang was with Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. He is now with Novatech Microelectronics Corporation, Ltd., Hsinchu 300, Taiwan, R.O.C (e-mail: supertg@video.ee.ntu.edu.tw)

Digital Object Identifier 10.1109/TMM.2006.876305

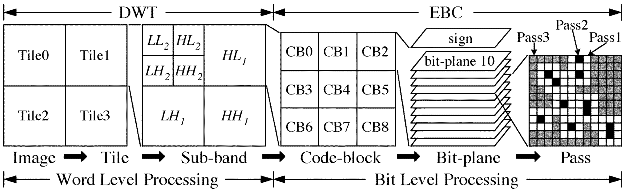


Fig. 2. Various abstract levels in JPEG 2000. An image is divided into tiles, subbands, code-blocks, bit-planes, and coding passes.

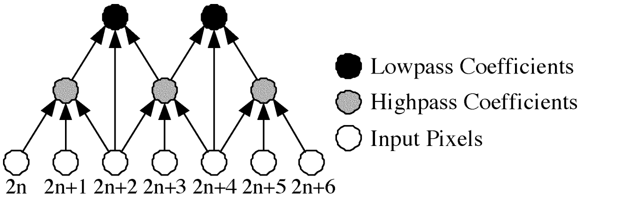


Fig. 3. Signal flow graph of 5-3 filter. Even-indexed coefficients are lowpass coefficients and odd-indexed coefficients are highpass coefficients.

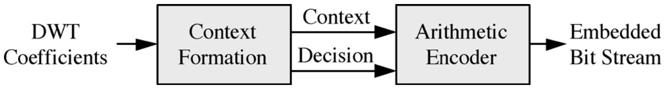


Fig. 4. Block diagram of the EBC algorithm. The CF generates context-decision pairs for the AE to generate the embedded bit streams.

adopt the technique of duplicating the EBC module to increase the throughput of their JPEG 2000 designs. However, this technique results in a large silicon area, and therefore high hardware costs.

In this work, we proposed a high-performance architecture for the JPEG 2000 encoder. For the EBC, we proposed a parallel architecture that increases the throughput without increasing operating frequency or silicon area. To optimize image quality, we also proposed a precompression RDO algorithm and architecture. By deciding truncation points before the EBC, the proposed algorithm can avoid unnecessary computations and eliminate the memory requirement for the rate-distortion information. In order to reduce the I/O access power consumption, a line-based architecture is proposed to achieve two-dimensional (2-D) and two-level DWT operations. This architecture can reduce the memory access to only one read and one write per sample, which is the theoretical lower bound. Combining the above three techniques, a JPEG 2000 encoder with high throughput, high image quality, and small area is proposed.

The remainder of this paper is as follows. Section II provides an overview of JPEG 2000. In Section III, the system architecture of the proposed JPEG 2000 encoder is described. Implementation results and comparisons are shown in Section IV. Finally, Section V concludes this paper.

II. JPEG 2000 Overview

In JPEG 2000, an image is decomposed into various hierarchical levels for coding, as shown in Fig. 2. An image is partitioned into tiles, which are independently coded. By encoding each tile independently, the JPEG 2000 can provide random access to the whole image. Each tile is decomposed by the DWT into some subbands with certain decomposition levels. For example, seven subbands are generated by two levels of decompositions. Resolution scalability can be supported by decoding bitstreams at various decomposition levels. One subband is further divided into code-blocks to be processed by the EBC. The DWT coefficients are represented as sign-magnitude for the EBC.

A. Discrete Wavelet Transform

In JPEG 2000, a multilevel 2-D DWT procedure is defined on a tile. In each decomposition level, the 2-D DWT decomposes the ith LL band (LL_i) into four subbands—LL_{i+1}, LH_{i+1}, HL_{i+1}, and HH_{i+1}. Note that the original tile is viewed as LL_0. A 2-D DWT is accomplished by cascading two one-dimensional (1-D) DWT, vertical 1-D DWT followed by horizontal 1-D DWT. The LL band is obtained by lowpass filtering in both horizontal and vertical directions. On the contrary, the HH band is obtained by highpass filtering in both directions. The HL (LH) band is obtained by highpass filtering in the horizontal (vertical) direction and lowpass filtering in the vertical (horizontal) direction.

For the 1-D DWT, the 5-3 reversible integer wavelet filter [15], also called Le Gall 5-3 filter, is defined using lifting-based filtering. Let $x(\cdot)$ and $y(\cdot)$ denote the input and output signals, respectively. The odd-indexed output, highpass coefficient, is calculated by

$$y(2n+1) = x(2n+1) - \left\lfloor \frac{x(2n) + x(2n+2)}{2} \right\rfloor. \quad (1)$$

Then, the even-indexed output, lowpass coefficient, is calculated by

$$y(2n) = x(2n) + \left\lfloor \frac{y(2n-1) + y(2n+1) + 2}{4} \right\rfloor. \quad (2)$$

The signal flow graph of the 5-3 filter is shown in Fig. 3. The reason why the filter is called 5-3 filter can be observed from the figure. The first number, 5, comes from that every lowpass coefficient is obtained by computations involving five input pixels. Similarly, the second number, 3, comes from that every highpass coefficient is obtained by computations involving three input pixels.

B. Embedded Block Coding

The embedded block coding algorithm is, in fact, a context-adaptive arithmetic encoder. It comprises the context formation (CF) and the arithmetic encoder (AE) as shown in Fig. 4. The CF generates context-decision pairs for the AE to generate the embedded bit streams. The AE encodes the binary decision with the probability adapted by the context. In the following, we will focus on the dataflow of the EBC. For more detailed information about the CF and the AE, the readers can refer to [7] and [8].

The scan order of the EBC is shown in Fig. 5. A magnitude bit-plane is divided into stripes, each of which is four samples high and equal width with the code-block. A column-based scan is used for scanning sample coefficients in each stripe. Three

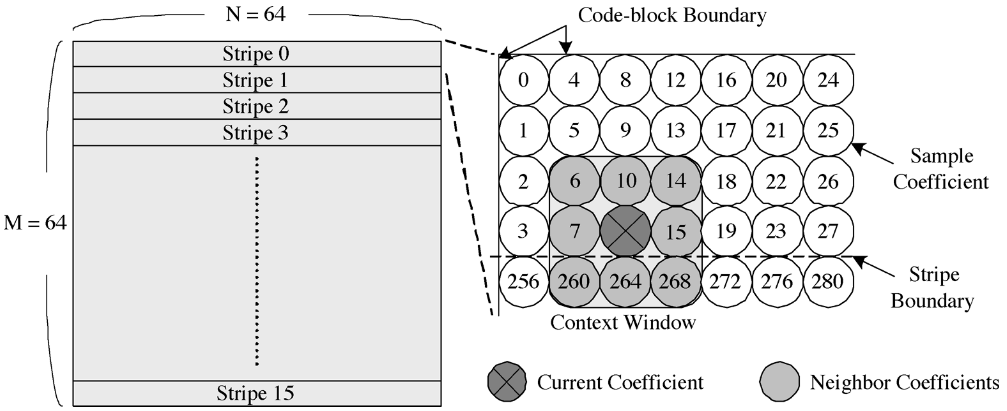


Fig. 5. Stripe-based scan order of the EBC. A 3 × 3 context window is used for the CF. The numbers indicate the scan orders.

TABLE I
CODING PASS CLASSIFICATION

Pass 1	Insignificant sample with at least one significant neighbor
Pass 2	Significant sample
Pass 3	Insignificant sample with all insignificant neighbors

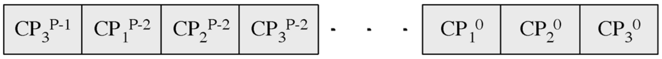


Fig. 6. Organization of embedded bit stream. The embedded bit stream of a code-block is organized from the MSB bit-plane to the LSB bit-plane, and Pass 1 to Pass 3 within a bit-plane.

scans are required for a magnitude bit-plane: significant propagation pass (Pass 1), magnitude refinement pass (Pass 2), and cleanup pass (Pass 3). Each sample coefficient belongs to one of the three coding passes according to the significant state of coefficients in the context window as shown in Fig. 5. The significant state stands for whether there is a nonzero sample coefficient coded in previous scans. When the significant state is "1" for a sample coefficient at, say, position 4, it means that at least a nonzero sample coefficient at position 4 has coded in previous scans. The conditions to classify a sample coefficient into one of the three coding passes are shown in Table I. As shown in Fig. 6, the embedded bit streams of a code-block is organized in the order—$CP_3^{P-1}, CP_1^{P-2}, \ldots, CP_2^0$, and CP_3^0, where P is the number of nonzero bit-planes of the code-block. The bit stream is said to be embedded because it can be truncated at end of any coding pass, and the resulting partial bit stream can also be decoded.

C. Rate-Distortion Optimization

In this section, we will briefly introduce the RDO in the JPEG 2000. Let R_i and D_i denote the rate and distortion of the ith code-block (B_i). Let n_i be a feasible truncation point, i.e. a coding pass, of B_i, and n_i is monotonically increasing from CP_3^0 to CP_3^{P-1} in Fig. 6. Thus, truncating B_i at n_i results in $R_i^{n_i}$ and $D_i^{n_i}$. The bit stream is optimized if the distortion cannot be reduced without the increase of the rate or, equivalently, the rate cannot be reduced without the increase of the distortion. The RDO is to find an optimal truncation point set, $\{n_i^\lambda\}$, that results in the optimized bit stream under rate or distortion constraint.

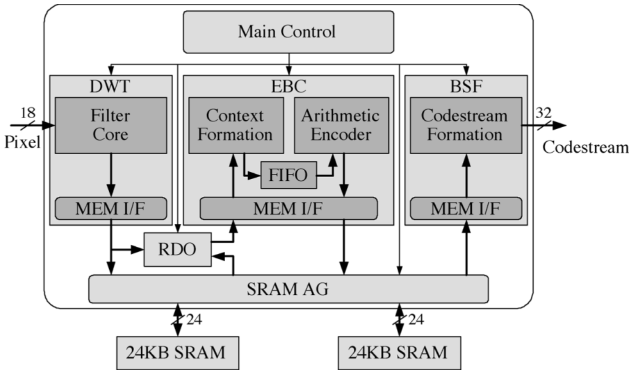


Fig. 7. System block diagram of the proposed JPEG 2000 encoder. The DWT, EBC, and BSF modules are pipelined at tile-level by two off-chip SRAMs.

This problem has been [7] mapped on the rate-constrained Lagrange equation

$$\min\left(D(\lambda) + \lambda R(\lambda)\right) = \min \sum_i \left(D_i^{n_i^\lambda} + \lambda R_i^{n_i^\lambda}\right) \tag{3}$$

where λ is the Lagrange multiplier. For a given λ, n_i^λ is obtained by

$$n_i^\lambda = \min\left\{ n_i \,\middle|\, S_i^{n_i} = \frac{\Delta D_i^{n_i}}{\Delta R_i^{n_i}} \geq \lambda \right\}, \tag{4}$$

where $\Delta D_i^{n_i}$ and $\Delta R_i^{n_i}$ are the reduced distortion and increased bit rate by selecting the coding pass corresponding to n_i. There is an interesting property of this solution that only $\Delta D_i^{n_i}$ and $\Delta R_i^{n_i}$ are the required information to achieve the RDO instead of the D_i and R_i. That is to say, the RDO can be achieved when S_i is sufficiently close to λ even if the full R-D information is unknown.

III. SYSTEM ARCHITECTURE

The architecture of the proposed JPEG 2000 encoder is shown in Fig. 7. There are six functional blocks in this architecture: the main control module, the DWT module, the EBC module, the bit stream formation (BSF) module, the RDO module, and the SRAM address generator (AG) module. The SRAM AG module

TABLE II
COMPARISONS OF VARIOUS 2-D DWT ARCHITECTURES

Architecture	Bandwidth†		Internal Buffer
	Read	Write	(words)
Direct 2-D	2.5	2.5	0
Block-based [13]	$\frac{1.25B}{B-2K}$	1.25	LB
Line-based [16]	1	1	$1.5LN$

† Bandwidth is represented as access times per sample.
B: Block width N: Tile width K,L: Filter dependent constants

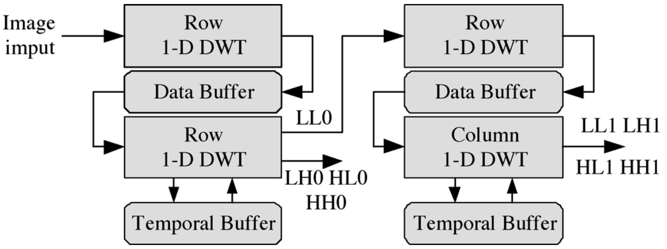


Fig. 8. Block diagram of the proposed two-level 2-D DWT architecture. The architecture can process two samples per cycle.

translates the addresses of the DWT, EBC, and BSF modules into real SRAM addresses. The main control module is a hardwired controller, which is composed of three finite state machines (FSM). The hardwired controller is much more cost-effective than general-purpose processors. The DWT, EBC, and BSF modules are pipelined at tile-level by use of the two off-chip SRAMs. The RDO module, which realizing the proposed RDO algorithm, observes the output of the DWT module and truncates the input of the EBC module. In the subsequent sections, we elaborate the architecture in detail.

A. Discrete Wavelet Transform

Memory issues are the most important for the DWT in JPEG 2000. Unlike the 8 × 8 discrete cosine transform (DCT), the DWT operates on a much larger area. Therefore, it becomes the major goal to reduce the memory requirements, which contain off-chip memory bandwidth and on-chip memory size. Table II shows the comparisons of three 2-D DWT architectures: the direct 2-D, overlapped block-based, and line-based architectures. The B is the block size of the block-based architecture and the N is the tile width. The L and K depend on the filter type. Two levels of decompositions are assumed. The line-based architecture minimizes tile memory bandwidth with some inter buffers. On the other hand, the block-based and direct 2-D architectures reduce the internal buffer size with the increase of the tile memory bandwidth. Note that the increase of tile memory read is proportional to the data recomputation of the retransmitted pixels. These computations are unnecessary, and therefore computational power and processing time are wasted.

Fig. 8 shows the proposed DWT architecture. Two 1-level 2-D line-based DWT modules are cascaded to accomplish the two-level 2-D DWT decomposition with the minimal memory bandwidth, which is one read for original pixel and one write for transformed coefficients. The architecture can process two input pixels and produce two output coefficients, one lowpass coefficient and one highpass coefficient, in a cycle. For the 2-D DWT, the internal buffers can be classified into two categories: the data buffer and the temporal buffer. The data buffer serves as the transpose memory between the 1-D row and column DWT modules. Unlike the conventional implementation in which data buffer requires two lines of pixel data [17], we use only 1.5 lines to implement it by use of a proper data access scheme [16]. To achieve the 128 × 128 tile size, 8512 bits (b) of buffers are required. Three of the six buffers are implemented by two-port SRAM with 128 × 24 b, 64 × 24 b, and 64 × 28 b, respectively. The other three are implemented by register files with 32 × 24 b, 32 × 28 b, and 16 × 28 b, respectively.

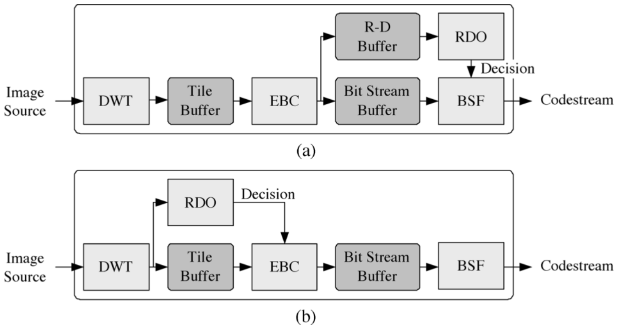


Fig. 9. Comparisons between the two RDO algorithms. (a) Post-compression RDO algorithm performs RDO after the EBC. (b) Precompression RDO algorithm performs RDO before the EBC.

B. Precompression Rate-Distortion Optimization

RDO is one of the important features of JPEG 2000. In the JPEG 2000 verification model (VM) [2], a post-compression RDO algorithm is recommended, which performs RDO after the EBC as shown in Fig. 9(a). There are two drawbacks of the post-compression RDO algorithm. First, the computational power and the processing time are wasted since the source image must be losslessly coded by the EBC but some bit streams are discarded by the post-compression RDO. Second, the large temporary memory is required to buffer the lossless bit streams and RD information for the rate control. To solve these problems, we proposed a precompression RDO algorithm [18], which utilizes the properties of the DWT and EBCOT algorithms, as shown in Fig. 9(b). By performing RDO before the EBC, the computational power and processing time of the EBC are saved and the buffers for RD information are eliminated.

Incremental rate (ΔR) and incremental distortion (ΔD) of embedded bit streams are essential for the RDO. The ΔD can be calculated in DWT domain, i.e., before the EBC, since it only depends on the value of the coefficient. The ΔR, on the other hand, cannot be known before actual coding by the EBC, and therefore the ΔR needs to be estimated. The accuracy of estimation is essential to achieve the RDO before the EBC without significant quality loss. Two features, randomness and propagation, of the EBC algorithm are utilized to accurately estimate the ΔR. Randomness represents the random property of Pass 2. The coding gain of Pass 2 is almost constant for various images, components, subbands, and bit planes as shown in Fig. 10. Each point in the figure shows the coding result of certain Pass 2 bit stream. The "Bits" shows how many bits belong to the coding pass, and the "Rate" is the length of the resulting embedded bit stream of that coding pass. As can be seen, almost all the coding

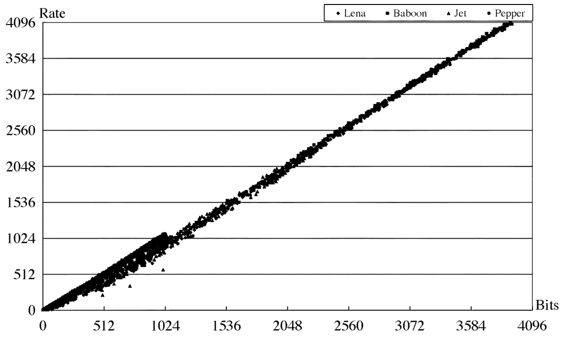


Fig. 10. Randomness property of Pass 2. The coding gain of Pass 2 is almost constant regardless of images, tiles, subbands, code-blocks, and bit-planes.

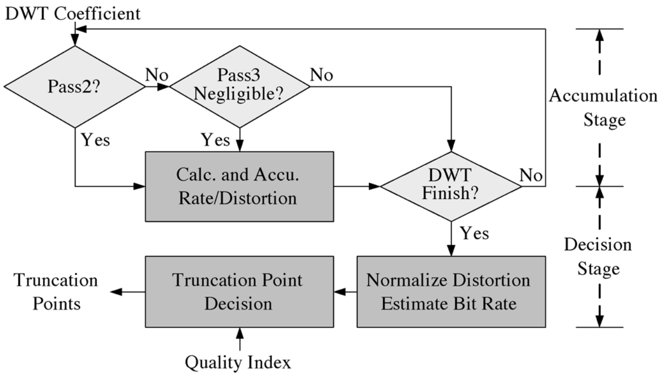


Fig. 11. Flowchart of the precompression RDO algorithm. In the accumulation stage, rate and distortion are calculated and accumulated. The truncation points are decided in the decision stage.

passes lie on a straight line, which means that the coding gain is almost the same for all the coding passes belong to Pass 2. Propagation property means that nonsignificant bits in the lowest two bit-planes will almost belong to Pass 1 due to the significant propagation. Experimental results show that only 0.72% and 9.47% of nonsignificant bits belong to Pass 3 for the lowest two bit planes, respectively. Thus, we can assume that a bit in the lowest two bit-planes is very likely to belong to Pass 1 if it does not belong to Pass 2. Combing the above two properties, the ΔR and the ΔD of the available truncation points ($\overline{\{n_i\}}$), which contain Pass 1 in the lowest two bit-planes and Pass 2 in all bit-planes, can be obtained since determining whether a bit belongs to Pass 2 is possible before the EBC.

The flowchart of the proposed precompression RDO algorithm is shown in Fig. 11. It contains two stages: the accumulation stage and the decision stage. In the accumulation stage, the incremental distortion and bit counts are accumulated for $\overline{\{n_i\}}$. In the decision stage, the weighted distortion and estimated bit rate are calculated, and the truncation points are decided according to the requested quality. Note that, coding passes belong to $\overline{\{n_i\}}$ are feasible truncation points. The hardware of the RDO module is a direct mapping of the algorithm. The distortion of a sample coefficient is obtained by table lookup. The distortion normalization and the bit rate estimation are simply constant multiplications. By comparing the quality index with the rate-distortion value of each truncation point, the truncation point can be decided.

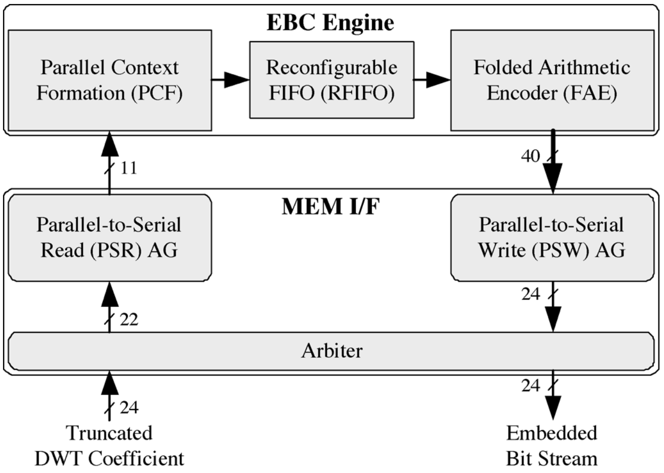

EBC Engine
Parallel Context Formation (PCF)
Reconfigurable FIFO (RFIFO)
Folded Arithmetic Encoder (FAE)
11
40
MEM I/F
Parallel-to-Serial Read (PSR) AG
Parallel-to-Serial Write (PSW) AG
22
24
Arbiter
24
24
Truncated DWT Coefficient
Embedded Bit Stream

Fig. 12. Block diagram of the parallel EBC architecture. The architecture can process all bit-planes of a DWT coefficient per cycle.

C. Embedded Block Coding

The most critical challenge to increase the throughput of a JPEG 2000 design is the EBC, which requires a lot of sequential operations and complicated controls. In our previous work [19], [20], a parallel EBC engine is proposed to process an entire DWT coefficient in a cycle. The major problem to integrate the parallel EBC engine is how to efficiently arrange and form the embedded bit streams. In this work, we proposed an efficient addressing scheme to solve this problem. The block diagram of the parallel EBC architecture is shown in Fig. 12. It contains two parts: the EBC engine and the memory interfaces (MEM I/F). The MEM I/F includes a parallel-to-serial read (PSR) AG, a parallel-to-serial write (PSW) AG, and an arbiter. The arbiter is used to make sure that only one AG is granted to access the SRAM and to prevent deadlocks from happening. In the following, the EBC engine, the PSR AG, and the PSW AG are described in detail.

1) EBC Engine: In [19] and [20], we proposed an EBC engine capable of processing a DWT coefficient in parallel, regardless of bit-width. In this section, we'll briefly review the architecture. Four new techniques are used to achieve the parallel processing. First, the gobang register band (GRB) fits the input coefficients with the dataflow defined in the standard. Secondly, the parallel context formation (PCF) approach is taken, instead of traditional sequential approaches [21], to increase the processing speed. Thirdly, a reconfigurable FIFO (RFIFO) architecture that reduces bubble cycles is obtained by exploiting the features of the EBC algorithm and the DWT algorithm. Finally, a folded arithmetic encoder (FAE) architecture is devised to reduce the area. Combing the three techniques, the parallel architecture is about six times faster than the best techniques described in the literature with similar hardware cost.

The block diagram of the parallel EBC engine is shown in Fig. 13. It can encode an 11-bit DWT coefficient per cycle. Therefore, 28 coding passes must be processed in parallel since each magnitude bit-plane, except the most significant bit (MSB) bit-plane, contains three coding passes. These bit streams are encoded by the FAE, which can process five of them at a time. Thus, there will be at most five output bytes per cycle.

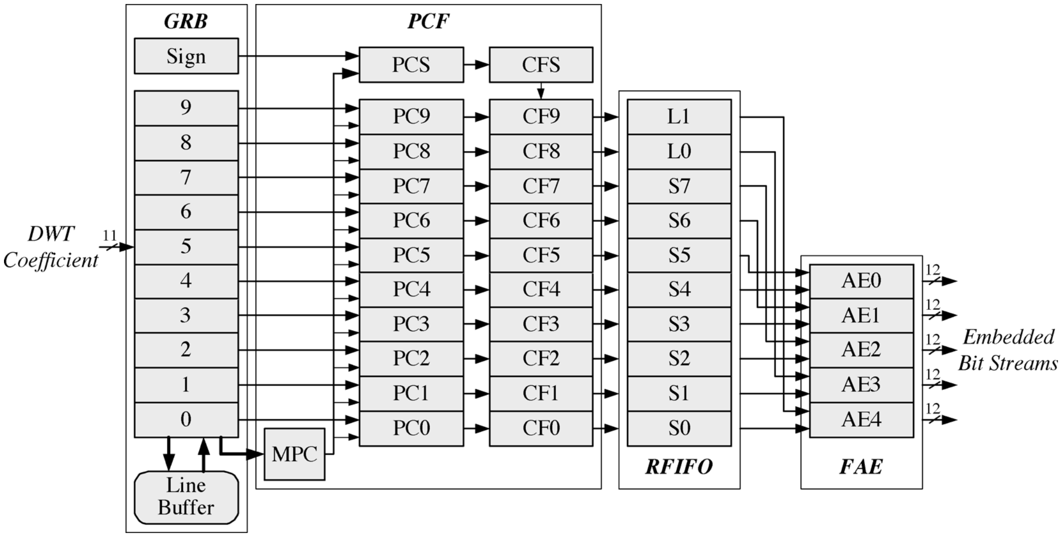


Fig. 13. Block diagram of the EBC engine. This architecture can process 11 bit-planes of a DWT coefficient per cycle.

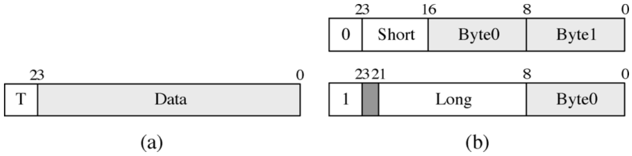


Fig. 14. Adaptive link list addressing scheme. (a) Bit allocation. (b) Two types of data arrangements.

2) PSR AG: The DWT coefficients are stored in the SRAM by a packed pattern. Two coefficients are stored in one word. The PSR AG reads the coefficients, which may be truncated by the RDO, and inputs serially to the EBC engine. Note that, 22 bits of a word is used since one DWT coefficient is 11 bits wide.

3) PSW AG: Due to the parallel processing, the EBC engine will simultaneously process 28 bit streams and outputs five of them in one cycle in the extreme case. To reduce the SRAM requirement, the partial bit streams are stored back to the same memory space as the coefficients. Because the coefficients that have been processed are no longer needed. The bit streams must be buffered since the header depends on the bit stream lengths. The challenge is how to use the limited and uncertain memory spaces to store 28 independent and randomly appeared bit streams. One cannot simply use predefined addressing rules. The address can only be decided after each byte appeared, and the address must be recorded for combining bit streams later.

The link list, in which some bits of a word are used as a pointer to the next word, is a straightforward solution. Since the SRAM has 8 K words, the pointer must have 13 bits. Thus, 11 bits are left and only one byte can be stored. Under this addressing scheme, the compression ratio of the EBC must be higher than $3(=(24/8))$. This is not feasible since the compression ratio of the EBC is typically 2.

To solve this problem, we proposed an adaptive link list (ALL) addressing scheme. As shown in Fig. 14(a), a word is divided into two parts: a type flag and a data segment. The data segment has two configurations indicated by the type flag as shown in Fig. 14(b). When the type flag is 0, the data segment is in short mode, in which the pointer has 7 bits. Otherwise, the data segment is in long mode and the pointer has 13 bits. Thus, one and two bytes can be stored in the short and long modes, respectively. The effectiveness of the ALL scheme lies in the unequal distribution of coding passes across the bit-planes. In general, almost all bytes are stored using short mode for the coding pass that are the majority of the bit-plane. Experimental results show that over 93.3% of memory words are used as the short mode, which means that $1.93(=2\times 0.93+1\times 0.07)$ bytes are stored in one word in average. This packing ratio can support compression ratio as low as $1.55\ (=(24/1.93\times 8))$, which rarely happens. In general, such a low compression ratio only occurs at LL band. Thus, the partial bit streams that are unable to stored back to the original memory space are stored in the other three subbands from the end since the LL band is processed after all other subbands. This effectively solves the problem since the overall compression ratio of a tile is almost guaranteed to higher than 2 for nature images.

In order to provide the random access, the packed packet header (PPH) is generated. The PPH is generated after entire code-block is coded since it requires the length of each partial bit stream. Taking advantage of this, the PPH is stored at the end of memory space, in which the code-block coefficients are stored.

D. Bit Stream Formation

The block diagram of the BSF module is shown in Fig. 15. The main header and tile header are generated by header table according to the state. The packet header analyzer and bit stream analyzer decode the read data from SRAM into packet header and bit stream. They also generate appropriate signals to the packet AG to generate next address. The sequencer combines the header and bit stream into codestream.

The processing time of the BSF module is dominated by the bit stream analyzer. It analyzes the data read from memory and extracts the ALL pointer and partial bit streams. Since two cycles of latency are required to access external SRAM, it can only analyze one word in two cycles. In the extreme case, there will be 8 K

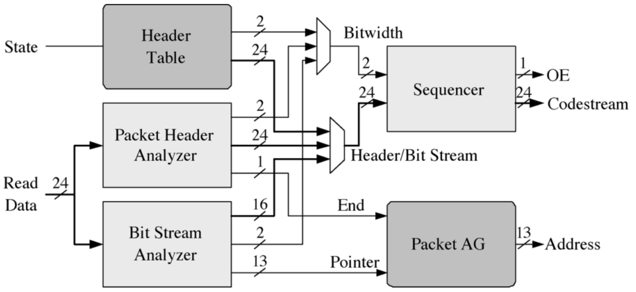


Fig. 15. Block diagram of the BSF module. The sequencer combines main header, packet header, and partial bit streams to form the final codestream.

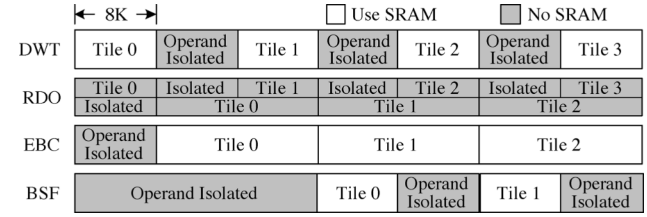


Fig. 16. System scheduling and bandwidth usage. The block in gray and white represent that the operations need and need not access the SRAM.

bytes (KB) of bit streams of a tile assuming a compression ratio of 2. As mentioned in Section III-C3, a memory word can store 1.93 bytes on average. Therefore, the maximum processing time of the BSF module will be $8.29 (= 2 \times (8/1.93))\ K$ cycles.

E. Scheduling

Fig. 16 shows the system scheduling and the bandwidth usage of the off-chip SRAMs. Tile-level pipeline is chosen to achieve high throughput and high utilization. There are three pipeline stages: the DWT, the EBC, and the BSF modules. The RDO module neither forms an individual pipeline stage nor uses any memory bandwidth. It observes the output coefficients of the DWT module to decide the truncation points, and on the other hand it truncates the input coefficients of the EBC module according to the truncation points. Thus, it operates at two pipeline stages for two tiles concurrently. In order to reduce power consumption, the DWT, the RDO, and the BSF modules are operand isolated at idle stages. Therefore, the switching power of the processing elements are saved.

By such efficient scheduling, the memory bandwidth of external SRAM is two operations per cycle at most, which enables the use of two single-port SRAM. The scheduling is achieved by matching the processing rate of the DWT, the EBC, and the BSF modules. The throughput of the EBC module depends on the truncation points, and is about one coefficient per cycle. Adopting lifting scheme, the throughput of the DWT module is two coefficients per cycle. By using the ALL addressing scheme (Section III-D), the BSF module can finish its work in about 8 K cycles, which approximately equal to the speed of the DWT module. Thus, the DWT and the BSF modules are designed to share the memory bandwidth of one SRAM. During a pipeline stage, the BSF module reads bit streams of the $(N-1)$th tile from one SRAM and forms the codestream. After that, the DWT starts to transform the $(N+1)$th tile and stores the transformed coefficients into the same SRAM. Meanwhile, the EBC module reads the DWT coefficients of the Nth tile from the other SRAM and stores the bit streams back to the same SRAM. Therefore, these techniques enable fully utilization of SRAM bandwidth and reduce the number of SRAM by one.

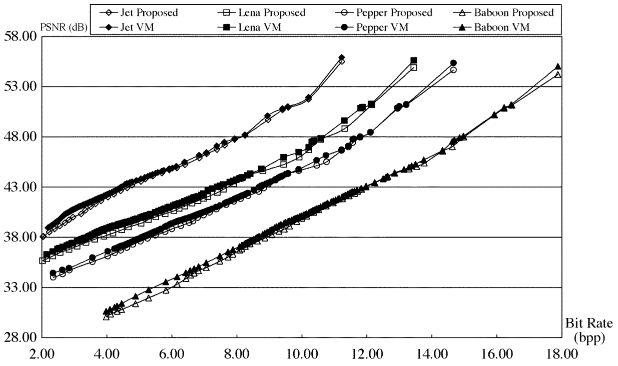


Fig. 17. Rate-distortion curves comparisons. The proposed precompression RDO is about 0.3 dB lower than the post-compression RDO.

TABLE III
THROUGHPUT VERSUS BIT RATE OF LENA

Bit Rate (*bpp*)	PSNR (*dB*)	Cycles (*K*)	Throughput (Sample/Cycle)
14.10	∞	1155	0.66
7.08	42.03	1025	0.75
3.80	38.09	885	0.87
1.80	35.32	839	0.92

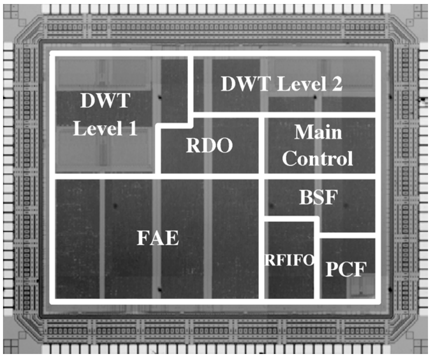


Fig. 18. Micrograph of the prototype chip. The core size is $2.73 \times 2.02\ \mathrm{mm}^2$.

TABLE IV
CHIP SPECIFICATIONS

Technology	TSMC 0.25 μm 1P5M CMOS	
Transistor Counts	914 K	
Core Size	2.73×2.02 mm^2	
Power Consumption	348 mW @ 81 MHz	
Operating Frequency	81 MHz	
Supply Voltage	Internal/External	2.8/3.3 V
Throughput	Lossy/Lossless	81/54 MS/s
On-Chip SRAM	Two-Port/Single-Port	6400/768 b

IV. EXPERIMENTAL RESULTS

A. Performance and Chip Features

The rate-distortion curves of the proposed precompression RDO and the post-compression RDO in VM [2] are shown in Fig. 17. The average quality loss of the proposed precompression RDO is about 0.3 dB. The degradation mainly comes from

TABLE V
COMPARISONS OF VARIOUS JPEG 2000 DESIGNS

Design	[10]	[11]	[12]	[13]	[14]	This Work
Category	Coprocessor	Codec	Codec	Encoder	Encoder	Encoder
Technology (μm)	N/A	0.18	0.18	0.25	0.18	0.25
Area† (mm^2)	N/A	N/A	189_d	25_c	6.0_d	5.5_c
Throughput‡ (MS/s)	-/10	65/40	50/-	21/-	60/-	81/54
Frequency (MHz)	100	150	180	27	150	81
Power (mW)	N/A	N/A	1944	N/A	280	348
Voltage (V)	N/A	1.5	1.8	2.5	1.8	2.8
Tile	160×128	1024 × 1024	256×256	1024×512	128×128	128×128
Code-block	64×64	N/A	N/A	N/A	32×32	64×64
DWT	5-3/9-7	5-3/9-7	5-3/9-7	9-7	5-3/9-7	5-3
RDO	Auxiliary	Auxiliary	No	No	No	Yes

† d/c for die/core area ‡ throughput in lossy/lossless mode

the estimation error and the missing truncation points. However, the degradation is negligible when compared with the benefits brought by the algorithm.

Table III shows the throughput of the encoder at various bit rates. The test image is Jet of size 512 × 512 and 24 bits per pixel (bpp). In lossless mode, the throughput is about $0.66(=(512 \times 512 \times 3)/(1165 \times 1024))$ samples/cycle. The throughput can be greatly increased to 0.99 samples/cycle in lossy mode. The speedup comes from the precompression RDO algorithm that truncates the DWT coefficients before the EBC module. Thus, the data to be processed by the EBC module are reduced. In extreme cases, a code-block is entirely discarded by the RDO module, which greatly reduces the processing time.

The micrograph of the prototype chip is shown in Fig. 18. The DWT area seems somewhat large due to the limitation of available SRAM architectures. The chip size can be further reduced by use of high-density SRAM. Table IV summarizes the chip specifications measured. This chip is fabricated with a $0.25\text{-}\mu\mathrm{m}$ CMOS 1P5 M technology, in which 914 K transistors are integrated on $2.73 \times 2.02\ \mathrm{mm}^2$. It consumes 348 mW at 81 MHz and 2.8 V supply voltage. Operating at 81 MHz, it can encode at least 54 MS/s in lossless mode and 81 MS/s in lossy mode. For natural images, the compression ratio is ranging from 1.8 to 20 corresponding to PSNR higher than 35 dB. Thus, the maximum output data rate of this chip is about 360 megabits per second (Mbps). It can support HDTV 720p (1280 × 720) 4:4:4 at 30 fps in real-time. Table IV also shows the memory requirements and the logic gate counts (2-input NAND gate equivalent) of the chip. Note that the gate counts of the main control module, the BSF module, and the SRAM AG module are 9386, which is much more efficient than a general-purpose processor.

B. Comparisons

A performance index (PI), defined as throughput per mm^2 and per MHz, is used to make a fair comparison to existing work. The PI of this work is $0.182(=(81/5.5 \times 81))\ \mathrm{S/mm}^2$, which means this work can process 0.182 samples per unit area per cycle. The area of the JPEG 2000 core in [13] is estimated as $25\ \mathrm{mm}^2$ and the PI will be $0.030(=(20.7/25 \times 27.4))\ \mathrm{S/mm}^2$. Hence, the developed chip is six times better than [13] using this metric. The improvement is mainly due to the proposed parallel EBC architecture. In order to make a complete comparison among various designs, important parameters for a JPEG 2000 encoder are summarized in Table V. It can be seen that this work achieves the highest throughput with the smallest area. Moreover, this work provides a rate-distortion optimized codestream, which is an important feature of JPEG 2000.

V. CONCLUSION

In this paper, a high-performance JPEG 2000 single-chip is implemented by use of $0.25\text{-}\mu\mathrm{m}$ CMOS technology. It can real-time encode HDTV 720p resolution at 30 fps at 81 MHz. Three techniques are adopted in this paper: the line-based DWT architecture, the parallel EBC architecture, and the precompression RDO algorithm. The line-based DWT architecture minimizes the memory bandwidth with small internal buffer by use of proper data access scheme. The parallel EBC architecture can process all the bit-planes of a DWT coefficient each cycle, which dramatically increases the throughput of the encoder. The precompression RDO algorithm optimizes the image quality by precisely estimating the rate and distortion before the EBC. Experimental results show that this encoder achieves the 81 MS/s processing rate with 5.5 mm^2 area, which are both the best results in the literature.

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/IECFDIS 15444-1)*, ISO/IEC JTC1/SC29/WG1 N1855, Aug. 2000.

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

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

[6] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

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

[8] D. Taubman, E. Ordentlich, M. Weinberger, and G. Serourssi, "Embedded block coding in JPEG 2000," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, BC, Canada, Sep. 2000, vol. 2, pp. 33–36.

[9] P.-R. Schumacher, "An efficient JPEG2000tier-1 coder hardware implementation for real-time video processing," *IEEE Trans. Consumer Electron.*, vol. 49, no. 4, pp. 780–786, Nov. 2003.

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

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

[12] DSPworx Mar. 2002, Cheetah [Online]. Available: http://www.dspworx.com/cheetah.htm

[13] 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.

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

[15] L. Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proc.IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, New York, Apr. 1988, vol. 2, pp. 761–764.

[16] P.-C. Tseng, C.-T. Huang, and L.-G. Chen, "Generic RAM -based architecture for two-dimensional discrete wavelet transform with line-based method," in *Proc. IEEE Asia Pacific Conf. Circuits and Systems*, Singapore, Dec. 2002, vol. 1, pp. 363–366.

[17] K. Andra, C. Chakrabarti, and T. Acharya, "A VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 966–977, Apr. 2002.

[18] Y.-W. Chang, H.-C. Fang, C.-J. Lian, and L.-G. Chen, "Novel pre-compression rate-distortion optimization algorithm for JPEG 2000," in *Vis. Commun. and Image Process.*, San Jose, CA, Jan. 2004, pp. 1353–1361.

[19] 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*, Bangkok, Thailand, May 2003, vol. 2, pp. 736–739.

[20] H.-C. Fang, Y.-W. Chang, T.-C. Wang, C.-J. Lian, and L.-G. Chen, "Parallel embedded block coding architecture for JPEG 2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 1086–1097, Sep. 2005.

[21] C.-J. Lian, K.-F. Chen, H.-H. Chen, and L.-G. Chen, "Analysis and architecture design of block-coding engine for EBCOT in JPEG 2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 3, pp. 219–230, Mar. 2003.

Hung-Chi Fang was born in I-Lan, Taiwan, R.O.C., in 1979. He received the B.S. degree in electrical engineering from National Taiwan University (NTU), Taipei, in 2001, and the Ph.D. degree from NTU in 2005. He was a visiting student at Princeton University, Princeton, NJ, with Prof. Wolf, supported by the "Graduate Students Study Abroad Program" of the National Science Council, Taiwan, in 2005.

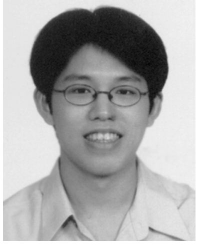
He is currently a Senior Engineer at MediaTek, Inc., Hsinchu, Taiwan. His research interests are VLSI design and implementation for signal processing systems, image processing systems, and video compression systems.

Yu-Wei Chang was born in Taipei, Taiwan, R.O.C., in 1980. He received the B.S. degree in electrical engineering in 2003 from National Taiwan University, Taipei, where he is currently pursuing the Ph.D. degree in the Graduate Institute of Electronics Engineering.

His research interests include algorithm and architecture for image/video signal processing, image coding systems JPEG 2000 and JBIG2, and related VLSI designs.

Tu-Chih Wang was born in Taipei, Taiwan, R.O.C., in 1975. He received the B.S., M.S., and Ph.D. degrees in electrical engineering in 1997, 1999, and 2003, respectively, from the National Taiwan University, Taipei.

His main research interests include video coding technology, DSP architecture, and media processor architecture.



Chao-Tsung Huang was born in Kaohsiung, Taiwan, R.O.C., in 1979. He received the B.S. and Ph.D. degrees in electrical engineering in 2001 and 2005, respectively, from the National Taiwan University, Taipei.

He is now with Novatech Microelectronics Corporation, Ltd., Hsinchu, Taiwan. His major research interests include VLSI design and implementation for 1-D, 2-D, and 3-D discrete wavelet transform.

Liang-Gee Chen (S'84–M'86–SM'94–F'01) was born in Yun-Lin, Taiwan, R.O.C., in 1956. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1979, 1981, and 1986, respectively.

He was an Instructor (1981–1986), and an Associate Professor (1986–1988) in the Department of Electrical Engineering, NCKU. While in the military service during 1987–1988, he was an Associate Professor in the Institute of Resource Management, Defense Management College. In 1988, he joined the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, Taiwan. During 1993–1994, he was a Visiting Consultant with the Digital Signal Processing (DSP) Research Department, AT&T Bell Labs, Murray Hill, NJ. In 1997, he was a Visiting Scholar with the Department of Electrical Engineering, University of Washington, Seattle. During 2001 to 2004, he was the first Director of the Graduate Institute of Electronics Engineering (GIEE), NTU. Currently, he is a Professor with the Department of Electrical Engineering and GIEE at NTU. He is also the Director of the Electronics Research and Service Organization, Industrial Technology Research Institute, Hsinchu, Taiwan. His current research interests are DSP architecture design, video processor design, and video coding systems.

Dr. Chen has served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY since 1996, as Associate Editor of IEEE TRANSACTIONS ON VLSI SYSTEMS since 1999, and as Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II since 2000. He has been the Associate Editor of the *Journal of Circuits, Systems, and Signal Processing* since 1999, and a Guest Editor for the *Journal of Video Signal Processing Systems*. He is also an Associate Editor of the PROCEEDINGS OF THE IEEE. He was the General Chairman of the 7th VLSI Design/CAD Symposium in 1995 and of the 1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He is the Past-Chair of Taipei Chapter of IEEE Circuits and Systems (CAS) Society, and is a member of the IEEE CAS Technical Committee of VLSI Systems and Applications, the Technical Committee of Visual Signal Processing and Communications, and the IEEE Signal Processing Technical Committee of Design and Implementation of Signal Processing Systems. He is the Chair-Elect of the IEEE CAS Technical Committee on Multimedia Systems and Applications. During 2001–2002, he served as a Distinguished Lecturer of the IEEE CAS Society. He received the Best Paper Award from the R.O.C. Computer Society in 1990 and 1994. Annually from 1991 to 1999, he received Long-Term (Acer) Paper Awards. In 1992, he received the Best Paper Award of the 1992 Asia-Pacific Conference on circuits and systems in the VLSI design track. In 1993, he received the Annual Paper Award of the Chinese Engineer Society. In 1996 and 2000, he received the Outstanding Research Award from the National Science Council, and in 2000, the Dragon Excellence Award from Acer. He is a member of Phi Tau Phi.