124 MSamples/s Pixel-Pipelined Motion-JPEG 2000 Codec Without Tile Memory

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

Abstract—A 124 MSamples/s JPEG 2000 codec is implemented on a 20.1 mm² die with 0.18 μm CMOS technology dissipating 385 mW at 1.8 V and 42 MHz. This chip is capable of processing 1920 × 1080 HD video at 30 fps. For previous works, the tile-level pipeline scheduling is used between the discrete wavelet transform (DWT) and embedded block coding (EBC). For a tile with size 256 × 256, it costs 175 kB on-chip SRAM for the architectures using on-chip tile memory or costs 310 MB/s SDRAM bandwidth for the architectures using off-chip tile memory. In this design, a level-switched scheduling is developed to eliminate tile memory and the DWT and the EBC are pipelined at pixel-level. This scheduling eliminates 175 kB on-chip SRAM and 310 MB/s off-chip SDRAM bandwidth. The level-switched DWT (LS-DWT) and the code-block switched EBC (CS-EBC) are developed to enable this scheduling. The codec functions are realized on an unified hardware, and hardware sharing between encoder and decoder reduces silicon area by 40%.

Index Terms—HD video, image compression, JPEG 2000.

I. INTRODUCTION

JPEG 2000 [15]–[17], [21], which is a new still image coding standard, is well known for its excellent coding performance and numerous features [19], such as region of interest (ROI), scalability, error resilience, etc. All these powerful tools can be provided by an unified algorithm in a single JPEG 2000 codestream.

Fig. 1 shows the functional block diagram of the JPEG 2000 encoder. Unlike JPEG [22], JPEG 2000 uses discrete wavelet transform (DWT) as the transformation algorithm and embedded block coding with optimized truncation (EBCOT) as the entropy-coding algorithm. EBCOT is a two-tiered algorithm. Tier-1 is the embedded block coding (EBC), which uses context-adaptive arithmetic coder, and tier-2 is post-compression rate-distortion optimization, which provides optimal image quality at a target bit rate. By use of the above new coding tools, JPEG 2000 outperforms JPEG by more than 2 dB in general [19]. However, the complexity of JPEG 2000 is much higher than that of JPEG.

Several JPEG 2000 codec designs have been reported in the literature [1], [8], [23], [24]. However, they suffer from either high operating frequency or large chip area. Amphion's codec [8] operates at the frequency higher than 150 MHz to provide the throughput of 60 MSamples/s (MS/s) and 20 MS/s for the encoder and decoder, respectively. The design of [1] occupies 144 mm² to achieve about 50 MS/s throughput. Sanyo [23], [24] developed an efficient JPEG 2000 codec architecture, which compromises between the throughput and the silicon area while keeping the operating frequency as low as 54 MHz. However, the SDRAM bandwidth requirement is so high that two buses are needed, and the operating frequency of each bus is two times that of the core.

There are three challenges in the design of efficient JPEG 2000 codec for HD video. First, the large data rate between the DWT and the EBC requires either large on-chip SRAM or high SDRAM bandwidth. Second, complicated control and irregular dataflow of the DWT and the EBC cost large area to meet the high throughput requirement. Third, hardware sharing between the encoder and the decoder is difficult due to different computation characteristics and dataflow. All of the above introduce high operating frequency, huge memory size, and high memory bandwidth for the chip implementation of a high throughput JPEG 2000 codec.

For the conventional architectures, tile-level pipeline scheduling, i.e., DWT and EBC pipelined at tile-level, is used due to two critical problems. First, the dataflow patterns of the DWT and the EBC are quite different; the DWT generates the coefficients in a subband-interleaving manner while the EBC encodes or decodes a code-block within one subband at a time. Second, the DWT is a word-level algorithm while the EBC is a bit-level one. Therefore, a tile memory is usually used for transferring coefficients between the DWT and the EBC. Tile-level pipeline scheduling introduces either high bandwidth for those architectures storing tiles in off-chip memory [23], [24] or high cost for those architectures storing tiles in on-chip memory.

In this work [5], we proposed a level-switched scheduling to solve the above two problems. For a tile size 256 × 256, it eliminates 175 kB SRAM tile memory for those architectures using on-chip tile memory and reduces 310 MB/s memory bandwidth for those architectures using off-chip tile memory. By use of this scheduling, the coefficients between the DWT and the EBC are transferred with a pixel-pipelined dataflow due to the elimination of tile memory. In this dataflow, no buffer is required between the DWT and the EBC. The coefficients generated by the DWT are encoded by the EBC immediately for the encoding flow or the decoded coefficients by the EBC are inverse-transformed immediately by the DWT for the

Y.-W. Chang, C.-C. Cheng, and L.-G. Chen are with the DSP/IC Design Laboratory, 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; ccc@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

C.-C. Chen and H.-C. Fang were with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. They are now with MediaTek Inc., Hsinchu 300, Taiwan, R.O.C. (e-mail: chunchia@video.ee.ntu.edu.tw; honchi@video.ee.ntu.edu.tw).
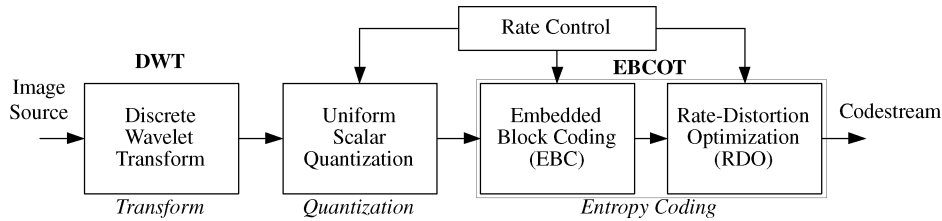
Fig. 1. Functional block diagram of the JPEG 2000 encoder. JPEG 2000 adopts DWT as transform algorithm and EBC as entropy coding algorithm. The RDO maximize image quality at a target bit rate.
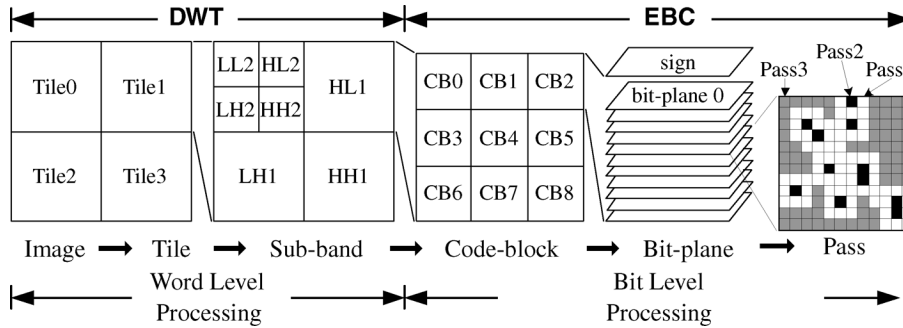

Fig. 2. Data hierarchy in JPEG 2000. An image is decomposed into tiles, sub-bands, code-blocks, bit-planes, and coding passes.

decoding flow. To enable this scheduling, a level-switched DWT (LS-DWT) and a code-block switched EBC (CS-EBC) are developed. The LS-DWT and the CS-EBC process multiple code-blocks in multiple subbands with an interleaving manner to eliminate the tile memory. The encoding and decoding functions are implemented on an unified hardware with little overhead for the control circuits. By use of the above techniques, the codec chip capable of processing 1920×1080 HD 4:2:2 video format at 30 frames per second (fps) is realized on a 20.1 mm^2 die with 0.18 μm CMOS technology dissipating 385 mW at 1.8 V and 42 MHz. Hardware sharing between encoder and decoder reduces silicon costs by 40%.

The organization of this paper is as follows. Section II gives some background information about JPEG 2000. Section III describes the proposed level-switched scheduling and Section IV shows the developed architectures. Implementation results and comparisons with the previous works are shown in Section VI. Finally, Section VII concludes this paper.

II. JPEG 2000 OVERVIEW

In JPEG 2000, an image is decomposed into various abstract levels for coding, as shown in Fig. 2. The image is partitioned into tiles, which are independently coded. Each tile is decomposed by the DWT into subbands with certain decomposition levels. For example, seven subbands are generated with two decomposition levels. Each subband is further partitioned into code-blocks, and each code-block is independently encoded by the EBC.

A. Discrete Wavelet Transform

In JPEG 2000, each tile is transformed by an multi-level and two-dimensional (2-D) DWT. For the forward DWT, an ith LL subband (LL_i) is decomposed into four subbands—LL_{i+1}, LH_{i+1}, HL_{i+1}, and HH_{i+1}. Fig. 3 shows an example that an 8×8 tile is decomposed into four subband. Note that the
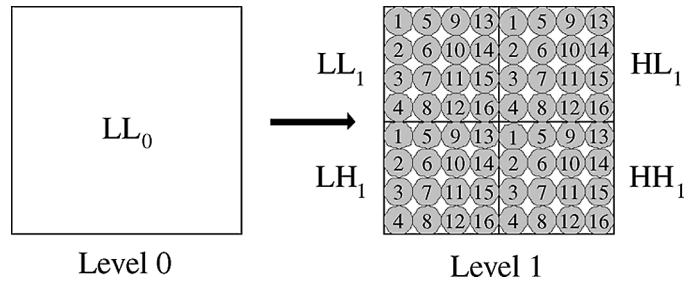

Fig. 3. A 8×8 tile is decomposed into four subbands. Each numbered circle represents the output order of each coefficient in each subband.

LL_0 denotes the original tile and the numbered circles denote the output order of each coefficient in each subband. As can be seen, the output of generated coefficients are interleaved in four subbands. In each level, a 2-D DWT can be factorized with two one-dimensional (1-D) DWT. The 2-D DWT is achieved by using vertical 1-D DWT first then being followed by the horizontal 1-D DWT. The LL band is obtained by low-pass filtering in both horizontal and vertical directions and the HH band is obtained by high-pass filtering in both directions. The HL (LH) band is obtained by high-pass filtering in the horizontal (vertical) direction and low-pass filtering in the vertical (horizontal) direction. For the inverse-transformed DWT, the procedure is a reverse of the procedure for the forward DWT, i.e., LL_{i+1}, LH_{i+1}, HL_{i+1}, and HH_{i+1} compose LL_i subband.

B. Embedded Block Coding

The Embedded Block Coding (EBC) algorithm contains context formation and context-adaptive arithmetic coder. Fig. 4(a) and (b) shows the block diagram of the EBC algorithms for the encoder and decoder, respectively. For the EBC in the encoder, the context formation generates a pair of context and decision bit, and the context-adaptive arithmetic encoder generates embedded bit streams. For the EBC in the decoder,
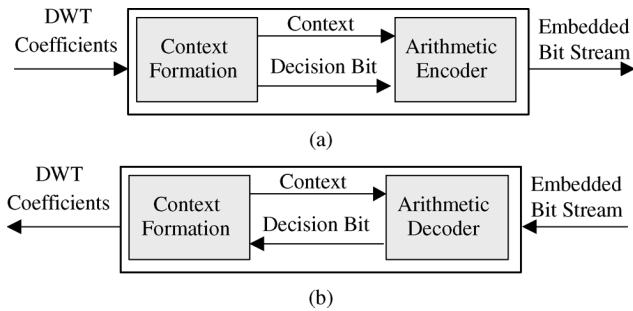
Fig. 4. (a) Block diagram of the EBC algorithm for the encoder. The context formation generates a pair of context and decision bit, and the context-adaptive arithmetic encoder generate embedded bit streams. (b) Block diagram of the EBC algorithm for the decoder. The context formation generates context, and the context-adaptive arithmetic decoder receive it and embedded bit streams to decode decision bit.
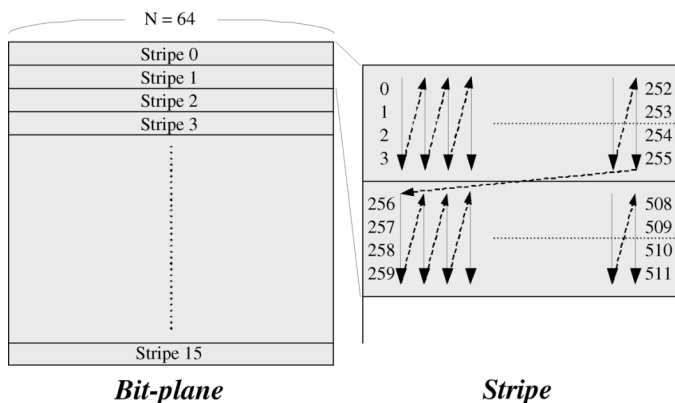

Fig. 5. A spatial scan order, called stripe scan, is used to scan a bit-plane. A stripe has the size of $4 \times N$. A bit-plane is scanned stripe by stripe and column by column from left to right in a stripe.

the context formation generates context, and the context-adaptive arithmetic decoder receives it and embedded bit streams to decode decision bit.

As shown in Fig. 2, the basic coding unit for the EBC is a code-block. The DWT coefficients in a code-block are sign-magnitude represented, and are encoded or decoded from the Most Significant Bit (MSB) bit-plane to the Least Significant Bit (LSB) bit-plane. Each bit-plane is scanned by three coding passes, Pass 1 (significant propagation pass), Pass 2 (magnitude refinement pass), and Pass 3 (clean-up pass). For each coding pass, a special coding order, called stripe scan, is used to scan a bit-plane. A stripe has the size of $4 \times N$, where N is the width of a code-block. Fig. 5 shows the stripe scan. A bit-plane is scanned stripe by stripe and column by column from left to right in a stripe.

III. Level-Switched Scheduling

There are two critical problems to design an efficient and high-throughput JPEG 2000 system. The first one is dataflow mismatch between the EBC and the DWT. The output/input dataflow of the DWT and input/output scan order of the EBC are different in the encoder/decoder. The dataflow of DWT coefficients interleaves in four subbands while the EBC process a code-block within a subband. Besides, the scan order of the EBC is stripe scan, which is different from the scan order of the DWT, which is column by column and row by row in a subband.

The dataflow mismatch introduces large temporal buffer for the dataflow conversion between the DWT and EBC in the codec system . The second problem is throughput mismatch between the DWT and the EBC. The DWT is a word-level processing algorithm while the EBC is a bit-plane sequential processing algorithm. In the encoder, the DWT coefficients of a code-block should be buffered since the EBC processes one bit-plane at a time. Therefore, not only the the EBC is the throughput bottleneck of the entire system but also the multiple memory accesses for DWT coefficients due to EBC's sequential property introduce the waste of power consumption. Due to the two critical problems, the previous architectures [23], [12], [24] use tile-level pipeline scheduling, i.e., the EBC processes the current tile while the DWT processes the next tile in the encoder system by using tile memory. For the target specification 1920×1080 4:2:2 30 fps and tile size 256×256, it costs 175 kB ($= 2 \times 256 \times 256 \times 10$ bits) memory requirement for storing 10-bit DWT coefficients of two tiles and 310 MB/s ($= 1920 \times 1080 \times 2 \times 30 \times 10$) coefficients transmission between the DWT and the EBC. Note that the 310 MB/s only contains the amount of data transmitted between the DWT and the EBC through tile memory. The bandwidth requirement for multi-level transformation of the DWT is not included since it depends on which DWT architecture is adopted.

In this section, a level-switched scheduling is proposed to solve above two mismatch problems. By use of this scheduling, the tile memory can be eliminated at a cost of a little additional memory buffer for the DWT and the EBC. This scheduling eliminates 175 kB SRAM tile memory for those architectures using on-chip tile memory and reduces 310 MB/s memory bandwidth for those architectures using off-chip tile memory.

To enable this scheduling, the parallel mode must be turned on for the EBC (CAUSAL, RESTART, and RESET are enabled [17]). In this mode, the arithmetic coder is terminated at end of each coding pass and the samples that come from the next stripe are considered insignificant. As a result of the two restrictions, the image quality of parallel mode is slightly worse than that of the default mode. The average peak signal-to-noise ratio (PSNR) loss is about 0.15 dB for 64×64 code-block and 0.35 dB for 32×32 code-block at medium bit-rate [20].

The key concept of the level-switched scheduling is to change operational coding flow in a tile to minimize the memory requirement between the DWT and the EBC. As we know, the use of tile memory arises from the dataflow mismatch between the DWT and the EBC. The memory size between the DWT and the EBC is proportional to the data lifetime of the DWT coefficients. If the lifetime for buffering DWT coefficients is shortened, the memory size is also reduced. Therefore, matching dataflow between the DWT and the EBC is a key to reduce memory requirement. As described in Section II, the basic coding unit in a code-block for the EBC is a stripe with size $4 \times N$, where N is code-block size. The processing order for the stripes in a code-block cannot be changed since the order is defined by the standard, but the DWT can change its scan order. Therefore, in the proposed scheduling, the scan order of the DWT is changed to stripe scan to match the scan order of the EBC. Besides, the DWT switches between levels to avoid accumulation of the DWT coefficients due to multi-level decompositions. To
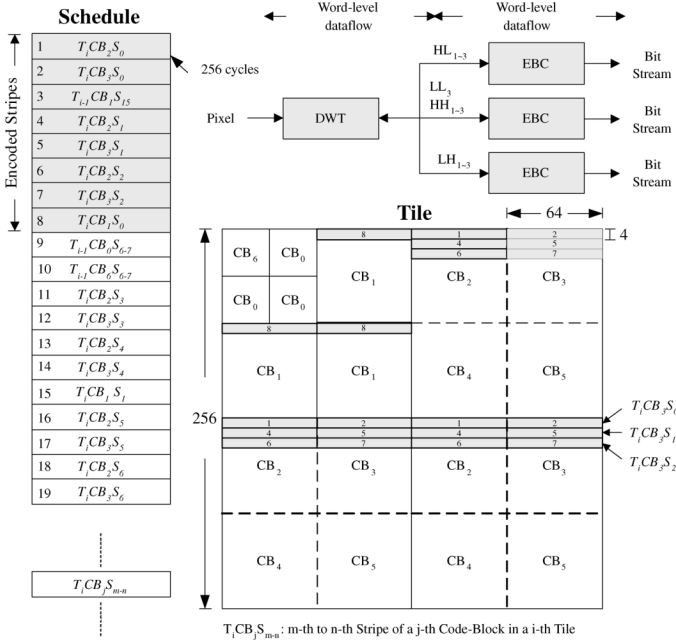
Fig. 6. Level-switched scheduling for JPEG 2000 encoder. Each rectangle on the left side represents a computation state both for the DWT and the EBC, and the number in it indicates the processing order. Each computation state requires 256 cycles to process either one 64×4 stripe or two 32×4 stripes in each subband.
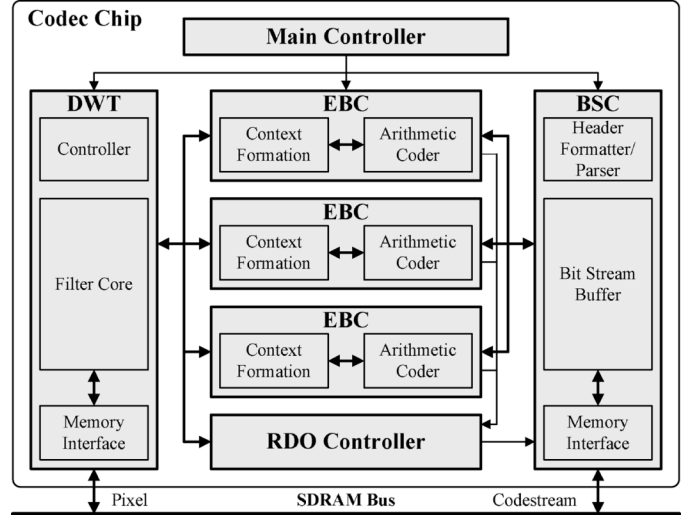

Fig. 7. Block diagram of the JPEG 2000 codec system. It contains a main controller, a 3-level DWT module, three embedded block coding (EBC) modules, a rate-distortion optimization (RDO) controller, and a bit stream controller (BSC).

co-operate with the DWT, the EBC is designed to be capable of switching between code-blocks.

The detail of the proposed scheduling for the encoding flow is shown in Fig. 6. Each rectangle in the left side represents a computation state both for the DWT and the EBC, and the number in it indicates the processing order. The computation state indicated by $T_i CB_j S_{m-n}$ means that the DWT and the EBC process the mth to nth stripes of the code-block with number j in the ith tile. Each computation state requires 256 cycles to process either one 64×4 stripe or two 32×4 stripes in each subband. The dataflow of the DWT is designed to match the stripe scan of the EBC, and the EBC is designed to be capable of processing one coefficient per cycle to match the the word-level throughput of the DWT. Three EBC are used to process three DWT coefficients in three subband. Note that the stripes in CB_6 (LL_3 subband) are processed by one of three EBC, while the other two EBC are idled. The operational sequences for the stripes in Fig. 6 are described as follows. At level 1 decomposition, the DWT generates coefficients in four subbands (computation state $1 \sim 2$ and $4 \sim 7$). The coefficients in the HL_1, LH_1, and HH_1 subbands are processed by three EBC immediately while the coefficients in the LL_1 subband are buffered for the next level decomposition. The DWT and the EBC switch to level 2 decomposition to process computation state 8 as soon as the buffered coefficients are enough for a computation state. After computation state 8 is finished, the DWT switches back to Level 1 to continue the unfinished parts. By use of this scheduling, the buffer between the DWT and the EBC is eliminated by processing stripes with an interleaved manner. To enable this scheduling, the DWT should buffer the unfinished coefficients for each LL band in each level and the EBC should buffer the coding states of each unfinished code-block.

For the scheduling for the decoding flow, the operational sequences for the stripes are opposite to those for the encoding flow. At the beginning to decode a tile, one of three EBC decodes the coefficients in LL_3 subband and these coefficients are buffered. The DWT and the EBC switch to level 2 when the numbers of buffered coefficients are enough. At level 2, three EBC decode the coefficients in the HL_3, LH_3, and HH_3 subbands, and the DWT composes the coefficients in four subbands to generate the coefficients in LL_2. Note that the numbers of buffered coefficients for each LL band in each level are the same as those in the encoding scheduling. The additional buffer to enable the encoding scheduling can be fully shared for the decoding scheduling.

IV. JPEG 2000 CODEC ARCHITECTURE

Fig. 7 shows the block diagram of the codec. It contains a main controller, a 3-level DWT module, three embedded block coding (EBC) modules, a rate-distortion optimization (RDO) controller, and a bit stream controller (BSC). The RDO controller maximizes image quality at a given target bit rate. Both the DWT and the EBC are pixel-pipelined such that no tile memory is required between the DWT and the EBC. Moreover, both the encoding and the decoding are one-pass, that is, no coefficient transmission to or from SDRAM.

To enable the level-switched scheduling, the level-switched DWT (LS-DWT) and the code-block switched EBC (CS-EBC) are developed. The detailed architectures are elaborated in the following sections.

A. Level-Switched DWT Architecture

Fig. 8 shows the architecture of LS-DWT. It contains two 1-D DWT modules, an LL-band buffer, and an inter-level line buffer. The direction of arrows shows the dataflow of the forward transformation. The LS-DWT is designed to be capable of processing four coefficients per cycle. For a computation sate with 256 cycles in the level-switched scheduling, this architecture can decompose a 128×8 or 64×16 block in LL_i and generates 64×4 or 32×8 coefficients, respectively, in each sub-
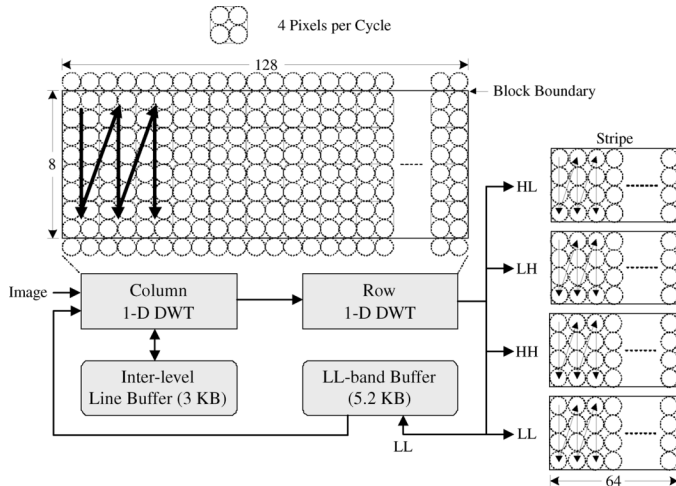
Fig. 8. Dataflow of the LS-DWT for forward transformation. It decomposes a 128×8 block and generates 64×4 coefficients in each subband. The coefficients in *LH*, *HL*, and *HH* are encoded by the CS-EBC as soon as they are generated while the coefficients in *HH* are stored at the LL-band buffer for the next level decomposition.
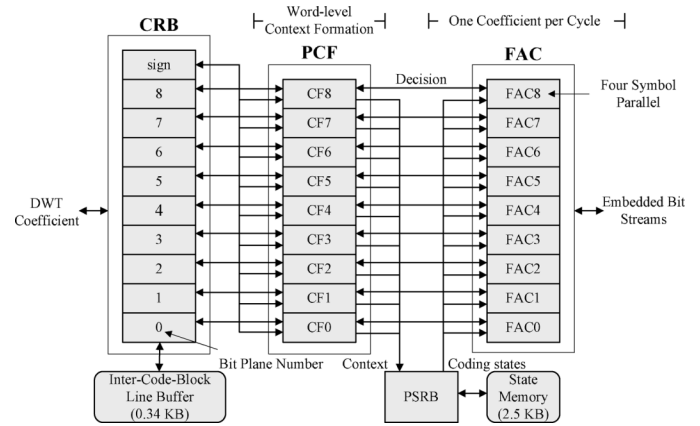

Fig. 9. Word-level EBC codec architecture. It can encode or decode one DWT coefficient per cycle. The coefficient register bank (CRB) is designed to match the scanning data flow of the EBC. The parallel context formation (PCF) process all bit-planes in parallel to generate contexts. The four-symbol arithmetic coder (FAC) processes all the contexts from a bit plane in one cycle.

band (LL_{i+1}, LH_{i+1}, HL_{i+1}, and HH_{i+1}). The coefficients in HL_{i+1}, LH_{i+1}, and HH_{i+1} are encoded by the CS-EBC as soon as they are generated such that no memory buffer is required to buffer these coefficients, while the coefficients in LL_{i+1} band are stored at the LL-band buffer for the next level decomposition. The DWT switches to the next level decomposition as soon as the amount of data in the LL-band buffer are enough for a computation state.

The LS-DWT is based on our previously proposed 2-D DWT architecture [6]. The DWT architecture in [6] uses a line-buffer to buffer the partially transformed coefficients [14] to avoid multiple accesses for the coefficients in the column direction and uses nonoverlapped stripe scan to eliminate the line-buffer in the row direction. By using line-based architecture, only one read for each pixel is required, which is the theoretical lower bound. Based on the analysis about bit width [2], the internal bit width used in this architecture is 14 bits and the output DWT coefficient is reduced to 10 bits. The simulation result shows that the image quality is about $45 \sim 50$ dB, which is not distinguishable by human eyes.

To enable the level-switched scheduling, the inter-level line buffer for the column 1-D DWT and LL-band buffer for the row 1-D DWT are used to buffer the partially transformed coefficients and generated coefficients, respectively, for each level. For the inter-level line buffer, four lines are required for each level since 9/7 filter is supported [14]. Therefore, the memory requirement of the inter-level line buffer is 3 kB ($= 4 \times (256 + 128 + 64) \times 14$ bits). To reduce the memory requirement for the LL-band buffer, 8, 16, and 8 lines are used to buffer LL_3, LL_2, and LL_1 coefficients, respectively. To obtain four basic coding stripes ($= (64 \times 4) \times 4$) in four subbands for level-2 decomposition, it should buffer 8 lines ($= 128 \times 8$) for LL_1. The 8 lines in LL_3 are equal to two stripes and each stripe with size 32×4. Two stripes are matched to two basic coding stripes for the EBC and are matched to a computation state with 256 cycles. The buffer size for the LL_2 ($= 64 \times 16$) is derived from the size

used for the LL_3 ($= 32 \times 8$). To fill buffered lines for LL_2 up, it can be achieved by decomposing the data in LL_1 buffer by four times. Theoretically, this buffer size is the minimal. However, in the actual implementation, additional four lines are used due to the latency of the LS-DWT. Therefore, the total buffer size is 5.2 kB ($= (12 \times 128 + 20 \times 64 + 12 \times 32) \times 14$ bits).

B. Code-Block Switched EBC Architecture

The EBC is the throughput bottleneck of a high-performance JPEG 2000 codec. In [12], a word-level EBC encoder is used to increase the throughput. However, the throughput depends on the complexity of image source and the target image quality. In this work, a word-level EBC codec, which guarantees one coefficient encoding/decoding per cycle, is developed. Fig. 9 shows the block diagram of the EBC codec, which processes a 10-bit DWT coefficient per cycle. The coefficient register bank (CRB) is designed to match the scanning data flow of the EBC. The parallel context formation (PCF) process all bit-planes in parallel to generate contexts. The four-symbol arithmetic coder (FAC) is proposed to encode/decode all the contexts from a bit plane in one cycle.

To match the level-switched scheduling of the DWT, 2.5 kB probability state memory and 0.34 kB inter-code-block line buffer are required for an EBC module to store the coding states of the unfinished code-blocks and the last row in the previous coding stripe for each code-block. The probability state memory is used to buffers the coding states in the probability state register bank (PSRB), which is used to store the coding states for the FAC, when switching to another code-block, and loads the states back to the PSRB before continuing the unfinished code-block. The coding states require 399 bits for a FAC in a bit-plane [10] and total 3990 bits for a code-block with 10-bit magnitude bit-plane. Although there are seven code-blocks (CB_0 to CB_6) should be processed by the EBC, only five of them are switched to each other at a time since CB_4 and CB_5 are processed after CB_2 and CB_3. The probability state memory for CB_2 and CB_3 is re-used for CB_4 and CB_5. Therefore, the probability state memory of EBC requires 19950 bits ($= 3990 \times 5$ bits $= 2.5$ kB). At the same way, the inter-level line buffer requires 0.34 kB
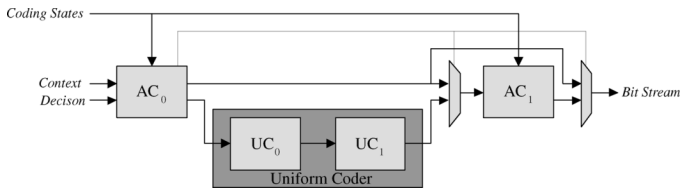
Fig. 10. Four-symbol arithmetic coder. It can operate at one-symbol, two-symbol, or four-symbol mode.

$(= (3 \times 64 + 2 \times 32) \times 11$ bits$)$ for the three code-block with size 64×64 and two code-blocks with size 32×32.

The detailed PCF architecture for the encoder and decoder is described in [11] and [3], respectively. The state memory required in the bit-plane sequential architectures [18], [13], [7], [9] are eliminated due to the parallel processing among all bit-planes.

To ensure that the EBC processes one coefficient per cycle, the four-symbol arithmetic coder (FAC) is designed to be capable of processing all the contexts generated from a bit plane in one cycle. Therefore, the throughput of the pixel-pipelined codec is guaranteed to be constant. The FAC architecture is shown in Fig. 10. It contains two general arithmetic coder (AC_0 and AC_1) and two uniform coder (UC_0 and UC_1). The architecture of general arithmetic coder is modified from the encoder architecture proposed in [4] to achieve codec function by reconfiguring its datapath. The FAC can operate at one-symbol, two-symbol, or four-symbol mode by the multiplexing control. The AC_0 is for magnitude coding and the AC_1 is for sign coding. Two uniform coders are designed for a special and nonadaptive code in run-length coding [17]. The critical path of two uniform coders is shortened by removing the circuits for the adaptive functions. Therefore, the critical path of the two unified coders is the same as that of one general arithmetic coder.

C. Rate-Distortion Optimization

The RDO controller adopts post-compression rate-distortion optimization scheme, which determines truncation points for each code-block at the end of coding a tile according to target bit-rate. In this scheme, the rate and distortion (R-D) for each coding pass of each code-block are accurately calculated. Therefore, the optimal image quality of a tile is guaranteed at target bit-rate.

The RDO controller uses an R-D register bank and an R-D memory to buffer the rate and distortion information for the current code-block and each unfinished code-block, respectively. The control scheme for the register bank and memory is the same as that for the PSRB and state memory in CS-EBC. At each computation state in the level-switched scheduling, the RDO controller receives the same coefficients scanned by the CS-EBC and side information such as coding pass from the CS-EBC to calculate distortion information. At the end of each computation state, the RDO controller receives the rate information of the current code-block from the arithmetic coder and loads the rate information of the next code-block into the arithmetic coder for further accumulation. After the finish of the last computation state for the previous tile, the RDO controller determines the truncation points and passes decisions to the BSC.
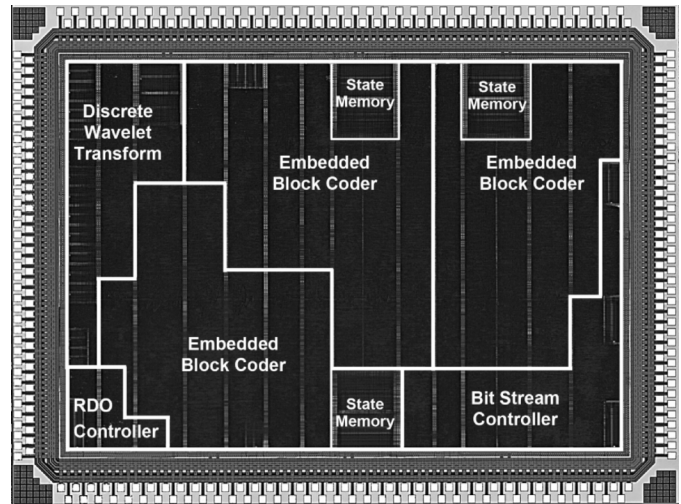

Fig. 11. Die micrograph of the JPEG 2000 codec.

TABLE I
SPECIFICATION OF THE DEVELOPED JPEG 2000 CODEC CHIP

Item	Description
Technology	TSMC 0.18 μm 1P6M CMOS
Pad/Core Voltage	3.3/1.8 V
Core Area	5.37×3.75 mm^2
Logic Gates	1155.5 K (2-input NAND gate)
SRAM	19.9 KB
Operating Frequency	42 MHz
Power Consumption	384 mW for lossless encoding 379 mW for lossless decoding
Package	PGA256
Image Size	Up to 64K$\times$64K
Processing Rate	124 M Sample/sec 1920$\times$1080 Frames/s 4:2:2 (support lossless and lossy coding)
DWT	(5,3), (9,7) with 3-Level Decomposition
Tile Size	256$\times$256
Code-block Size	64$\times$64

V. HARDWARE SHARING TECHNIQUES

To reduce hardware cost, two hardware sharing techniques are developed to design the codec. First, the level-switched scheduling for encoder and decoder have inverse-matched switching characteristics to achieve 100% memory sharing for the LS-DWT and the CS-EBC. The shared memory, including the inter-level line buffer, LL-band buffer, inter-code-block line buffer, and state memory, is 16.7 kB, which occupies 83% of total memory usage of the codec. Second, the filter core sharing processing elements with multiplexed coefficients and arithmetic coder reconfigures its datapath save 489K logic gates. For the 1-D filter core with lifting scheme architecture, the dataflow for the forward and backward transformation is the same but the multiplicators of the multipliers are different. Therefore, a large portion of the processing elements such as multipliers and adders can be shared for the forward and backward transformation by using the multiplexed multiplicators. For the arithmetic coder, many computations are the same between the arithmetic encoder and arithmetic decoder. Therefore, reconfigurability can be achieved with little control overhead. By reconfiguring the datapath, an arithmetic coder can save 17K gates compared with separate arithmetic encoder and arithmetic decoder. Because

TABLE II
LOGIC GATE COUNTS STATISTICS OF THE JPEG 2000 CODEC

| | Main control | LS-DWT | CS-EBC | | | RDO | BSF |
			CRB	CF	FAC		
Gate counts (NAND2)	12300	79753	14271	34907	228943	20889	207545
SRAM bits	0	68096	3072	0	20480	4928	19168

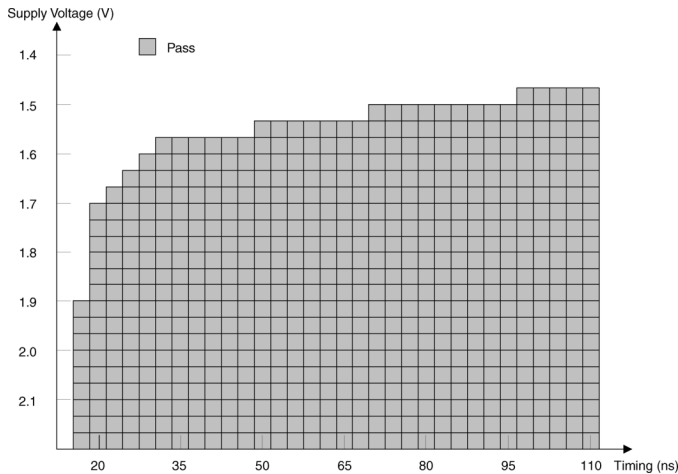

Fig. 12. Shmoo plot of the codec chip. The supply voltage could be scaled down to 1.75 V while maintaining processing performance.
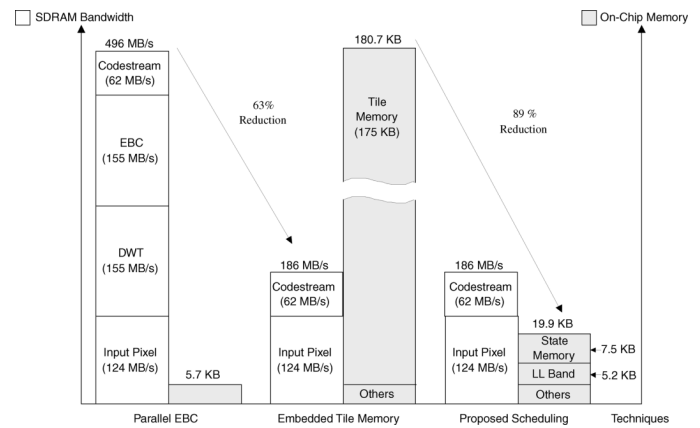

Fig. 13. Effectiveness of SDRAM bandwidth and on-chip memory size reduction. The proposed level-switched scheduling can reduce on-chip SRAM by 89%.
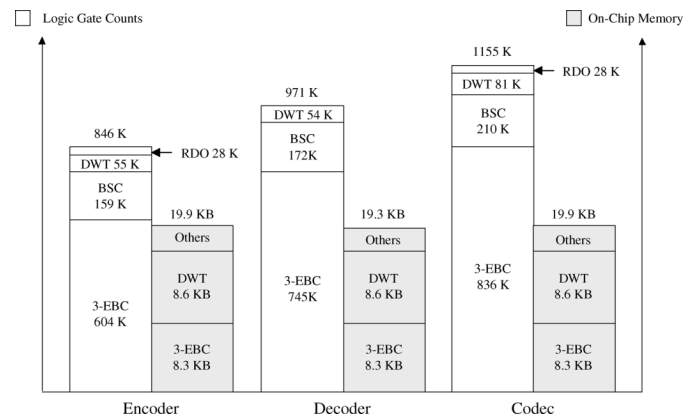

Fig. 14. Effectiveness of cost reduction. It shows the logic gate counts and the memory requirement to implement an encoder, a decoder, and a codec.

of the total 27 arithmetic coders used for the three CS-EBC, tremendous logic gate counts can be saved.

VI. EXPERIMENTAL RESULTS

A. Chip Implementation and Features

The single-chip JPEG 2000 codec is implemented on a 20.1-mm^2 die using TSMC 0.18-μm CMOS one-poly six-metal (1P6M) technology and has been received on September 2005. The die micrograph is shown in Fig. 11 and Table I shows the features of this chip. It contains 1155K logic gates and 19.9 kB of SRAM. This prototype only supports tile size 256×256, code-block size 64×64 and three-level decomposition. For the smaller tile size and fewer decomposition levels, it can be easily achieved by modifying control scheme without any modification for the architectures of LS-DWT and CS-EBC. However, we did not implement other control schemes in this chip. The detailed gates count distribution is shown in Table II, in which gate counts contain logic gates used to realize registers. The power consumption is 385 mW at 1.8 V and 42 MHz for lossless encoding and decoding. The processing rate of this chip is 124 MS/s or, equivalently, 1920×1080 HD video with 4:2:2 format for lossless encoding/decoding.

B. Testing Result

This chip is fully tested by extensive test patterns. The chip works as expected and can correctly encode or decode images. The measured timing versus various supply voltage are shown in Fig. 12. The target working frequency is 42 MHz, which is equivalent to 23.8 ns. By observing Fig. 12, at 1.8 V supply

voltage, the chip can work at a frequency higher than 42 MHz and the supply voltage could be scaled down to 1.7 V while maintaining target specification.

C. Effectiveness of the Level-Switched Scheduling

Fig. 13 shows the effectiveness of the proposed level-switched scheduling on the reduction of memory requirements and external memory bandwidth. The parallel EBC means that only the word-level EBC architecture is used but the level-switched scheduling is not applied. Therefore, the DWT and the EBC are pipelined at tile-level by using off-chip tile memory. The 5.7 kB memory includes the line buffer for one level used in the DWT, state memory, and line buffer for one code-block used in the EBC

TABLE III
COMPARISON OF JPEG 2000 SYSTEMS

Architecture	Technology (μm)	Area (mm^2)	Frequency (MHz)	Throughput (MS/s)	Power (mW)	Tile Size	Code-Block Size	DWT Level	Coding Switches	Performance Index ($MSample/MHz \cdot mm^2$)
ADI [1]	0.18	144	115	65/65[†]	N/A	4096	N/A	6	N/A	0.004/0.004[†]
Amphion [8]	0.18	5.4	150	60/20[†]	280	128	32	5	D,P	0.066/0.022[†]
Sanyo [23][24]	0.18	13/6.5[†]	54	70/35[†]	N/A	4096	32	3	P	0.1/0.1[†]
This work [5]	0.18	20.1	42	124/124[†]	384	256	64	3	P	0.148/0.148[†]

[†] Encoder/Decoder. D:Default mode. P:Parallel mode.

as well as other usages such as bit-streams buffer and rate-distortion buffer for the RDO. Although the target specification can be achieved by word-level EBC, the SDRAM bandwidth is so high due to the DWT coefficients transmission through the external SDRAM. The SDRAM bandwidth can be reduced to 37% of the original one by embedding the tile memory. However, the on-chip memory is too large such that dramatically increases the silicon cost. By use of the proposed level-switched scheduling, the on-chip tile memory is eliminated at the cost of a little on-chip memory while the bandwidth is kept the same.

D. Effectiveness of Hardware Sharing

Fig. 14 shows the effectiveness of cost reduction by using two hardware sharing techniques. It shows the logic gate counts and the memory requirement to implement an encoder, a decoder, and a codec. With sharing techniques, the logic gate counts of the DWT in the codec is about 50% larger than those of the encoder or decoder. The logic gate counts of the three EBC in the codec is 38% and 11% larger than those of the encoder and decoder, respectively. The EBC in the decoder has larger logic gate counts due to the fact that the PCF module is much more complex than that in the encoder. As well, the shared BSC between encoder and decoder also saves 121K logic gates. With the above sharing methods, the resulting logic gates of the codec are 136% (118%) larger than those of the encoder (decoder). The silicon area is reduced by 40% compared to the independent encoder and decoder.

E. Comparison

The comparisons with the previous works are summarized in Table III. The works of ADI and Sanyo use off-chip tile memory. Therefore, the tile size can be up to 4096. Amphion's work uses on-chip tile memory but tile size is only 128×128 since larger tile size cost too much silicon area. For the coding switches, Sanyo uses two bit-plane parallel architecture for the EBC to achieve the listed throughput. Therefore, it only supports parallel coding mode.

It is hard to compare various works since coding parameters are different from each other. However, we use a performance index (PI), defined as throughput per unit area at 1 MHz, to make a comparison for the existing works. The PI is not to justify which design is superior to the others, but to provide an evaluative method for reference. The PI is a good index to know how efficient a design uses area. The higher PI means higher area efficiency. The PI of this chip is 0.148 ($= 124/20 \times 42$). The estimated area of the JPEG 2000 encoder/decoder core in [23] and [24] is 13/6.5 mm^2. Therefore, the PI for the encoder/decoder is 0.100/0.100 ($= (70/13 \times 54)/(35/6.5 \times 54)$). Hence, this codec is 1.48 times more area-efficient than both the encoder

and decoder in [23] and [24]. Moreover, the SDRAM bandwidth of this chip is 280 MB/s less than that of [23] and [24]. For [8], the codec functions are implemented on a unified hardware to achieve 60 MS/s and 20 MS/s for encoder and decoder, respectively. The resulting PI for encoding/decoding is 0.066/0.022 ($= (60/6 \times 150)/(20/6 \times 150)$). As can be seen, the area efficiency of our chip is higher than other works by at least 1.48 times. Also, our chip has lower SDRAM bandwidth than others under the comparison of the same specification since there is no coefficients transmission to or from SDRAM in this chip.

VII. CONCLUSION

In this design, a JPEG 2000 single chip codec is presented. Both encoding and decoding functions achieve 124 MS/s data rate. The level-switched scheduling reduces 175 kB on-chip memory for the architectures using on-chip tile memory, and 310 MB/s SDRAM bandwidth for the architectures using off-chip tile memory. It matches the dataflow and throughput of the LS-DWT and the CS-EBC to eliminate tile memory. The word-level CS-EBC guarantees one coefficient encoding/decoding per cycle by use of developed parallel context formation and four-symbol arithmetic coder. Two hardware sharing techniques reduce silicon area by 40% compared to the independent encoder and decoder. First, the memory in the LS-DWT and CS-EBC is 100% shared between the encoder and the decoder. Second, filter core with multiplexed coefficients and reconfigurable arithmetic coder save 489K logic gates. The experimental results show this chip is high performance, low off-chip memory bandwidth, and low on-chip memory requirement.

REFERENCES

[1] ADV202. Analog Devices, 2004 [Online]. Available: http://www.analog.com/
[2] C.-C. Cheng, C.-T. Huang, J.-Y. Chang, and L.-G. Chen, "Line buffer wordlength analysis for line-based 2-D DWT," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, May 2006, vol. 3, pp. 924–927.
[3] Y.-W. Chang, H.-C. Fang, C.-C. Chen, and L.-G. Chen, "Design and implementation of word-level embedded block coding architecture in JPEG 2000 decoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, May 2006, vol. 2, pp. 449–452.
[4] Y.-W. Chang, H.-C. Fang, and L.-G. Chen, "High performance two-symbol arithmetic encoder in JPEG-2000," in *Proc. IEEE Int. Symp. Consum. Electron.*, Sep. 2004, pp. 101–104.
[5] Y.-W. Chang, H.-C. Fang, C.-C. Cheng, C.-C. Chen, C.-J. Lian, and L.-G. Chen, "124 MSmples/s pixel-pipelined motion-JPEG 2000 codec without tile memory," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 2006, pp. 404–405.
[6] C.-C. Cheng, C.-T. Huang, P.-C. Tseng, C.-H. Pan, and L.-G. Chen, "Multiple-lifting scheme: Memory-efficient VLSI implementation for line-based 2-D DWT," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 2005, vol. 5, pp. 5190–5193.
[7] J.-S. Chiang, Y.-S. Lin, and C.-Y. Hsieh, "Efficient pass-parallel for EBCOT in JPEG 2000," in *Proc. IEEE Int. Symp. Circuits Syst.*, Scottsdale, AZ, May 2002, vol. 1, pp. 773–776.

[8] CONEXANT. CS6590. [Online]. Available: http://www.am-phion.com/CS6590.html

[9] M. Dyer, D. Taubman, and S. Nooshabadi, "Memory efficient pass-parallel architecture for JPEG2000 encoding," in *Proc. IEEE Int. Symp. Signal Process. Applicat.*, Jul. 2003, vol. 1, pp. 53–56.

[10] H.-C. Fang, Y.-W. Chang, C.-C. Cheng, C.-C. Chen, and L.-G. Chen, "Memory efficient JPEG 2000 architecture with stripe pipeline scheme," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Philadelphia, PA, Mar. 2005, vol. 5, pp. 1–4.

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

[12] 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 *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 2004, pp. 328–329.

[13] Y.-T. Hsiao, H.-D. Lin, and C.-W. Jen, "High-speed memory saving architecture for the embedded block coding in JPEG 2000," in *Proc. IEEE Int. Symp. Circuits Syst.*, Scottsdale, AZ, May 2002, vol. 5, pp. 133–136.

[14] C.-T. Huang, P.-C. Tseng, and L.-G. Chen, "Generic RAM-based architectures for two-dimensional discrete wavelet transform with line-based method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 910–920, Jul. 2005.

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

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

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

[18] 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, pp. 219–230, Mar. 2003.

[19] 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.

[20] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, pp. 1158–1170, Jul. 2000.

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

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

[23] H. Yamauchi, K. Mochizuki, K. Taketa, T. Watanabe, T. Mori, Y. Matsuda, Y. Matsushita, A. Kobayashi, and S. Okada, "A 1440 × 1080 pixels 30 frames/s motion-JPEG2000 codec for HD movie transmission," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 2004, pp. 326–327.

[24] H. Yamauchi, S. Okada, K. Taketa, Y. Matsuda, T. Mori, T. Watanabe, Y. Matsuo, and Y. Matsushita, "1440 × 1080 pixel, 30 frames per second motion-JPEG 2000 codec for HD-movie transmission," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 331–341, Jan. 2005.
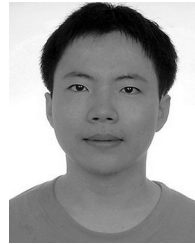
Yu-Wei Chang was born in Taipei, Taiwan, R.O.C., in 1980. In 2003, he received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C, where he is currently working toward the Ph.D. degree in the Graduate Institute of Electronics Engineering.

His research interests include algorithms and architecture for image/video signal processing, image coding system JPEG 2000, JBIG2, and related VLSI designs.
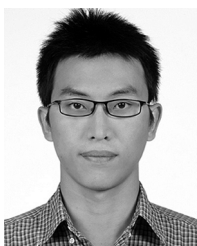
Chih-Chi Cheng was born in Taipei, Taiwan, R.O.C., in 1982. In 2004, he received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C, where he is currently working toward the Ph.D. degree in the Graduate Institute of Electronics Engineering.

His research interests include algorithms and architecture for image/video signal processing, discrete wavelet transform (DWT), and intelligent video processing.

Chun-Chia Chen was born in Changhwa, Taiwan, R.O.C., in 1982. In 2004, he received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C, where he is currently working toward the M.S. degree in the Graduate Institute of Electronics Engineering.

His research interests include algorithms and architecture for JPEG 2000 and JBIG2.

Hung-Chi Fang was born in I-Lan, Taiwan, R.O.C., in 1979. In 2001, he received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, R.O.C. He received the Ph.D. degree in 2005 also from National Taiwan University.

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. Now, he is a Senior Engineer with MediaTek Inc., Hsinchu, Taiwan, R.O.C. His research interests are VLSI design and implementation for signal processing systems, image processing systems, and video compression systems.

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, Tainan, Taiwan, R.O.C., in 1979, 1981, and 1986, respectively.

He was an Instructor (1981–1986), and an Associate Professor (1986–1988) in the Department of Electrical Engineering, National Cheng Kung University. In the military service during 1987 to 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. During 1993–1994, he was a Visiting Consultant of DSP Research Department, AT&T Bell Laboratories, Murray Hill, NJ. In 1997, he was a visiting scholar of the Department of Electrical Engineering, University of Washington, Seattle. During 2001–2004, he was the first Director of the Graduate Institute of Electronics Engineering (GIEE) of National Taiwan University (NTU). Currently, he is a Professor of the Department of Electrical Engineering and GIEE in NTU, Taipei, Taiwan. He is also the Director of the Electronics Research and Service Organization in 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 the 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 SP 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 Tan Phi.