CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Brunel University Research Archive

A VLSI Architecture of JPEG2000 Encoder

Leibo LIU, Ning CHEN, Hongying MENG, Li ZHANG, Zhihua WANG, and Hongyi CHEN

***Abstract*—This paper proposes a VLSI architecture of JPEG2000 encoder, which functionally consists of two parts: Discrete Wavelet Transform (DWT) and Embedded Block Coding with Optimized Truncation (EBCOT). For DWT, a Spatial Combinative Lifting Algorithm (SCLA) based scheme with both 5/3 reversible and 9/7 irreversible filters is adopted to reduce 50% and 42% multiplication computations respectively compared with the conventional Lifting Based Implementation (LBI). For EBCOT, a Dynamic Memory Control (DMC) strategy of Tier-1 encoding is adopted to reduce 60% scale of the on-chip wavelet coefficient storage and a subband parallel-processing method is employed to speed-up the EBCOT Context Formation (CF) process; an architecture of Tier-2 encoding is presented to reduce the scale of on-chip bit-stream buffering from full-tile size down to three-code-block size and considerably eliminate the iterations of the Rate-Distortion (RD) truncation.**

***Index Terms*—DWT, EBCOT, LIFTING, JPEG2000**

I. Introduction

JPEG2000 [1] is intended to create a new image coding system that for different types of still image with different characteristics, allowing different image models, preferably with a unified system. It will provide a set of features vital to many high-end emerging applications by taking advantage of new modern technologies.

JPEG2000 is composed of two major parts: Wavelet transform and EBCOT [2]. Fig. 1 shows the functional block diagram of JPEG2000. Wavelet transform is a subband transform which transfers images from spatial domain to frequency domain. To achieve efficient lossy and lossless compression within a single coding architecture, two wavelet transform kernels are employed by ISO/IEC 15444-1 [1]. The 5/3 reversible and 9/7 irreversible filters are chosen for lossless and lossy compression respectively. After wavelet transform, the coefficients are scalar quantized if lossy compression is chosen. Afterwards, coefficients are entropy encoded by EBCOT, which is a two-tier coding algorithm proposed by David Taubman [2]. Each wavelet subband is then divided into code-blocks and Tier-1 coding engine encodes these code-blocks into independent embedded bit-streams using context-based Arithmetic Encoding (AE) subsequently. Finally, Tier-2 reorders the code-block bit-streams into the final JPEG2000 bit-stream with RD slope optimized property and the features specified by user.

For the wavelet transform part, this paper proposes a novel architecture of 5-level Mallat [3] decomposition, 2-D biorthogonal DWT based on SCLA [4] with both 5/3 and 9/7 filters. SCLA, first proposed by Dr. Hongying MENG, emerges from the lifting scheme where the combinative law of matrix multiplications on the 2-D DWT operator matrix is utilized to combine the vertical and horizontal operations. By utilizing the 5/3 reversible and 9/7 irreversible filters, SCLA substantially reduces the number of multiplications for the 2-D biorthogonal DWT by a ratio of 50% and 42% respectively compared with the conventional LBI [5], [6].

Since JPEG2000 recommends a push-pull model between DWT and EBCOT Tier-1 part, the Tier-1 engine has to preserve a large number of wavelet coefficients. To solve this problem, this paper proposes an efficient memory management strategy named DMC, which can substantially reduce 60% of the scale of the on-chip wavelet coefficients memory and ensure the full memory-reusability. Moreover, a parallel architecture that processes each subband independently is presented to speed-up the entire Tier-1 entropy encoding process.

For the EBCOT Tier-2 part, this paper proposes a novel rate control scheme to execute optimized truncation with the process of AE in parallel. A considerable reduction of computational costs can be achieved without iterative truncations compared with the popular implementations. In addition, bit-stream buffer scale can be reduced from full-tile size to code-block size simultaneously.

The rest of this paper is organized as follows. In Section Ⅱ, Ⅲ,Ⅳ, the VLSI architectures of the three parts are described and analyzed in detail. The experimental results and the performance are depicted in Section Ⅴ. Finally, a conclusion is given in Section Ⅵ.

Manuscript received May 6th, 2003. This work was supported in part by the China National High Technologies Research Program (863) under Grant 2002AA1Z1420.

Leibo LIU is with the Institute of Microelectronics, Tsinghua Univ., Beijing 100084 PRC (phone: 8610-62781991; fax: 8610-62771130; e-mail: llb99@ mails.tsinghua.edu.cn).

Ning CHEN is with the Department of Electronics Engineering, Tsinghua University, Beijing, 100084, PRC (e-mail: cn01@mails.tsinghua.edu.cn).

Hongying MENG is with the Department of Mechanical Engineering, Univ. of Dundee, Scotland UK (e-mail: h.meng@dundee.ac.uk).

Li ZHANG is with the Department of Electronics Engineering, Tsinghua Univ., Beijing 100084 PRC (e-mail:chinazhangli@mail.tsinghua.edu.cn).

Zhihua WANG is with the Institute of Microelectronics, Tsinghua Univ., Beijing 100084 PRC (e-mail:zhihua@tsinghua.edu.cn).

Hongyi CHEN is with the Institute of Microelectronics, Tsinghua Univ., Beijing 100084 PRC (e-mail:chy-ime@tsinghua.edu.cn).

II. ARCHITECTURE OF SCLA BASED DWT

A. Algorithm of SCLA Based DWT with 9/7 filter

1) Overview of SCLA Based DWT with 9/7 filter

For a 2-D image block $M = \{m(i, j) \mid i, j = 0,1,...N-1\}$, the forward wavelet transform of the SCLA with decomposition level one by the 9/7 filter can be represented as (1).

$$\begin{aligned} Y &= T_{NxN} \times M \times T^{t}_{NxN} \\ &= (EDCBA\ \ ) \times M \times (EDCBA\ \ ) \\ &= (E(D(C(B(A \times M \times A^{t})B^{t})C^{t})D^{t})E^{t}) \end{aligned} \quad (1)$$

Where $Y = \{y(i, j) \mid i, j = 0,1,...N-1\}$ is the resulting coefficient vector; T_{NxN} is the associated wavelet transform operator matrix; and *A,B,C,D,E* are the constant matrices associated with each step of the SCLA. The expression of these 5 matrices can refer to [4].

The SCLA computing process starts from the center and extends to the two sides in (1). Fig. 2 (a) and (b) show the corresponding SCLA processing operations on matrices *A* and *B*.

In Fig. 2 (a) and (b), the symbol ⬭ means that the value of the current element does not change during the transform. The operations for the symbol ⯐ include three steps as follows:

1) Sum the 4 horizontal and vertical neighbors of the current element;
2) Multiply the sum of step one by α or β;
3) Add the product of step two to the value of the current element and leave the resulting sum to the current element.

The operations for the symbols ⬍ and ⬌ are similar to those of symbol ⯐. The only difference is that, in step one, only the two vertical or horizontal neighbors are summed. The operations on matrices *C* and *D* are similar to the ones for matrices *A* and *B* with α replaced by γ and β replaced by δ.

Fig. 2 (c) shows the SCLA operation for matrix *E*, where the symbol ⊗ means to multiply the value of the current element by η and symbol ⊕ means to multiply it by $1/\eta$, where $\eta=\xi^2$. The exact value of $\alpha, \beta, \gamma, \delta, \eta$ and ξ can refer to [4].

2) Number of Multiplications of SCLA with 9/7 filter

For NxN (assume $N=2^L$, L is the natural number) image block with decomposition level J (J≤L), the total number of multiplications by the LBI is $2x(4-4^{-J+1})xNxN$ and by the SCLA is $7/6x(4-4^{-J+1})xNxN$. The ratio of the SCLA to LBI is 58%. The comparison is listed in TABLE I.

The analysis of SCLA with 5/3 filter is similar to the 9/7 case. Its multiplication comparison is listed in TABLE II. The ratio of the SCLA to LBI is 50%.

B. Proposed Architecture of SCLA Based DWT

The proposed SCLA based DWT architecture reads in a tile of image serially, followed by the SCLA based DWT and directly outputs of the LH, HL and HH wavelet coefficients at each level of the decomposition to the following EBCOT Tier-1 part. At the same time, the LL coefficients are written back to the processor in preparation for the next level of decomposition. Line based transform [7] is utilized to guarantee the tile is read line by line only once. The proposed architecture is shown in Fig. 3, which consists of three major blocks: DWT filter, Input and output register buffer and On-chip line-buffer memories.

1) Organization of the On-chip Memory

Computing process of LBI with 9/7 filter can be illustrated in Fig. 4 and represented in (2) to (7), in which $X = \{x(l,k) \mid l,k = 0,1,...N-1\}$ is the line based raw input data, $s_l^0, d_l^0, s_l^1, d_l^1$ stand for the temporary data in each processing step, while s_l^2, d_l^2 stand for the target wavelet coefficients. Because (7) can be calculated in-place, it is omitted from the following analysis.

$$s_l^0 = x_{(l,2k)} \quad d_l^0 = x_{(l,2k+1)} \quad (2)$$

$$d_l^1 = d_l^0 + \alpha(s_l^0 + s_{l+1}^0) \quad (3)$$

$$s_l^1 = s_l^0 + \beta(d_l^1 + d_{l-1}^1) \quad (4)$$

$$d_l^2 = d_l^1 + \gamma(s_l^1 + s_{l+1}^1) \quad (5)$$

$$s_l^2 = s_l^1 + \delta(d_l^2 + d_{l-1}^2) \quad (6)$$

$$s_l = \xi s_l^2 \quad d_l = d_l^2 / \xi \quad (7)$$

In Fig. 4, in order to calculate the target wavelet coefficients, for example, s_3^2, d_2^2, there are two feasible methods to reserve the raw input data and temporary data in the line-buffer memories:

1) If d_4^0, s_5^0 are processed simultaneously, it only needs to reserve 2 lines of the raw input data d_4^0, s_5^0, and 4 lines of the temporary data $s_4^0, d_3^1, s_3^1, d_2^2$;
2) If d_4^0, s_5^0 are processed in sequence, (3) to (6) can each split into 2 steps as depicted in (8) to (11). Thus, there are only 1 line of the raw input data d_4^0 or s_5^0 and 4 lines of the temporary data $s_4^0, d_3^1, s_3^1, d_2^2$ to be reserved.

$$\bar{d}_l^1 = d_l^0 + \alpha(s_l^0), d_l^1 = \bar{d}_l^1 + \alpha(s_{l+1}^0) \quad (8)$$

$$\bar{s}_l^1 = s_l^0 + \beta(d_{l-1}^1), s_l^1 = \bar{s}_l^1 + \beta(d_l^1) \quad (9)$$

$$\bar{d}_l^2 = d_l^1 + \gamma(s_l^1), d_l^2 = \bar{d}_l^2 + \gamma(s_{l+1}^1) \quad (10)$$

$$\bar{s}_l^2 = s_l^1 + \delta(d_{l-1}^2), s_l^2 = \bar{s}_l^2 + \delta(d_l^2) \quad (11)$$

According to the above analysis, the LBI scheme needs at least 5 on-chip line-buffer memories. Since SCLA is derived from LBI, its memory organization is the same as LBI's. However, access to such memory organization is inefficient and the corresponding computations are sophisticated. The number of multiplications with different numbers of line-buffer memories are compared in TABLE III (assuming that the tile size is with one decomposition level), indicating that the organization of 6 line-buffer memories can provide the best tradeoff performance.

The 6 line-buffer memories are denoted as *Line0* - *Line5* in Fig. 5 for a decomposition level of 5, where N stands for the image tile width. *Line0 – Line3* buffer memories are used to store intermediate data for all 5 levels. *Line4* and *Line5* buffer memories are split into two pair of segments respectively, denoted as *Line4_Level4*, *Line4_LL* and *Line4_Level4*, *Line4_LL*, to satisfy the access timing constraints. *Line_Level4* and *Line5_Level4* are used for storing the raw image data with level4 decomposition whereas *Line4_LL* and *Line5_LL* are utilized to store the LL output of the previous level decomposition for level0 - level3. Access to these memories is line-based so that the positions of the stored data can be easily located.

2) SCLA based DWT filter control

The DWT filter consists of five processing elements, as shown in Fig. 3. Four elements named as Processing Elements (PE) marked from *A* to *D* are used for the multiplications of the matrices *A* to *D* in (1), each of which has a 3 x 3 working region. These four elements are completely identical except their locations in the filter. The other one is used for multiplications of matrix *E* in (1), covering a 2 x 2 working region.

The numbering of the 3 x 3 working region for PE *A*,*B*,*C* or *D* is given in Fig. 6 (a), in which "X" represents the matrices *A*,*B*,*C* or *D* and block "x" stores the old value of *x10*. Assume $\lambda=\alpha, \beta, \gamma$ or δ, the operation steps on the 3 x 3 working region are depicted in (12).

$$\begin{cases} (1) x21' = x21 + \lambda \times (x20 + x22) \\ (\text{where the apostrophe indicates the new value}) \\ (2) x11' = x11 + \lambda \times (x + x12 + x01' + x21') \\ (\text{where x represents the old value of x10}) \\ (3) x' = x12, x12' = x12 + \lambda \times (x02 + x22) \end{cases} \quad (12)$$

The numbering of the 2 x 2 working region for PE *E* is given in Fig. 6 (b), whose operations are to just multiply the current element by a constant parameter η or $1/\eta$ and leave the result in the current location, hence, it is easy to process matrix *E* in-place.

When the processor starts to work, it first reads 6 lines of data into the input register buffer from the 6 line-buffer memories at a rate of 1 column (including 6 data) per clock tick. After that, the input data is pushed into the DWT filter. The DWT filter processes SCLA algorithm mentioned above on the data in the pipelined processing elements *A, B, C, D, E* and simultaneously pops 6 lines of results into the output register buffer at the same rate as 1 column per clock tick. 4 lines of the results are intermediate data, which are written back into the *Line0* - *Line3* buffer memories for the next step of the current level decomposition. The remaining two lines consist of the LL, HL and LH, HH wavelet coefficients, with the LL written back into the *Line4_LL* or *Line5_LL*, preparing for the next level decomposition, while the LH, HL, as well as HH, directly outputted. At the same time, the raw image data is pushed into the *Line4_Level4* or *Line5_Level4* from outside. This operation is repeatedly pipelined until the last wavelet coefficient of the current tile is processed and outputted from the output register buffer.

3) Implementation Precision of the Wavelet Coefficient and the Constants

Several initial tests were made to determine the wavelet coefficient precision with the constants precision fixed at 32 bits. For different compression ratios, the resulting Peak Signal to Noise Ratios (PSNR) are listed in TABLE IV using the "Lenna" image (512 x 512 x 8 bits).

From the results in TABLE IV, the finite-precision of the DWT coefficient was chosen to be 17 bits. Several similar tests were then made with this coefficient precision to determine the precision of the constants $\alpha, \beta, \gamma, \delta$ and η. The corresponding PSNR results are listed in TABLE V which determines the finite-precision of the constants to be 13 bits.

III. ARCHITECTURE OF EBCOT TIER-1 ENTROPY ENCODING

EBCOT Tier-1 uses the DWT-generated subband samples for further processing. Typically, Mallat is performed as the basic decomposition rule, in which all levels contain three subbands except the coarsest decomposition level level0. The subbands LH, HH and HL form three series, with increasing level index, each of which is called an orientation. Operations on these orientations are almost the same, indicating that three identical encoding cores can be integrated to perform the entire encoding task in parallel and therefore improving the whole encoder performance. The subband of level LL ---- level0, however, can be attached to any of these three orientations without any degrade in coding efficiency. The EBCOT Tier-1 subband parallel architecture is illustrated in Fig. 7, containing four main functional parts: Cleanup pass (CL), Significance Propagation pass (SP), Magnitude Refinement pass (MR) and AE.

According to the Tier-1 algorithm, code-block is the minimum unit in which the original wavelet coefficients to be compressed. Inside each code-block, a key concept of the "fractional bit-plane" is employed in order to acquire a fine embedding, which separates a given quantized bit-plane into three coding passes, i.e. SP, MR and CL. Totally there are four different operations involved in these coding passes, which form the foundation of the embedded block coding strategy. For example, if a sample is not yet significant in the current bit-plane, a combination of the Zero Coding (ZC) and Run-Length Coding (RLC) is used to record whether or not the sample become significant in this bit-plane; otherwise the Sign Coding (SC) is invoked to encode sign of the sample. If the sample is already significant, the Magnitude Refinement (MR) coding is used to code the current bit, refining the sample to a finer precision. During these operations, the sample bits along with their contexts are delivered to the following AE to get further compression.

A. Dynamic Memory Control

JPEG2000 recommends a line-based DWT push-pull model in the analysis filter bank, in which DWT produces wavelet coefficient lines and pushes them into the Tier-1 encoder, per

line at a time. Although this model minimizes the memory needed in transform part, all the responsibilities of reserving and manipulating the coefficients have to be undertaken by the Tier-1 part.

Since the coefficients in the DWT line buffers must be transferred into an on-chip wavelet coefficient memory located between the DWT and EBCOT Tier-1 part and preserved there until the code-block which they belong to is processed by the encoder, there are lots of difficulties in coefficient locating because sample lines in different wavelet decomposition levels have different lengths. Moreover, the reuse of on-chip wavelet coefficient memory is not easy. Motivated by these problems, a DMC strategy is proposed to arrange the access sequence to the coefficient memory. Under DMC scheme, the coefficient memory is divided into blocks with fixed size, same as the maximum code-block size. These blocks are named as Dynamic Memory Block (DMB), which are the minimum units that can be reused. A DMB can only be used by 1 code-block at a time. Even if the wavelet coefficients may not occupy a whole DMB (for instance, the code-block in sub-band L_0 may be much smaller), the remaining area has to be reserved and can't be occupied by others. Each DMB has several flags to indicate its current status. Fig. 8 shows the idea of DMC scheme, in which each box represents a DMB in one of the three decomposition orientations with a specific status.

As shown in Fig. 8, there are totally four kinds of block statuses in DMB as: *S1* Full, under processing; occupied by EBCOT Tier-1; *S2* Full, waiting for processing; in queue; *S3* Not full, data buffering; occupied by DWT; *S4* Empty; not in use.

It is obvious that at most three DMBs can reside in either *S1* or *S3* status, each of which belongs to a decomposition orientation. Since the number of the blocks in status *S2* only depends on the difference between the processing capability of DWT and EBCOT Tier-1, it is easy to make a tradeoff between these two parts to minimize the scale and make a full reuse of the on-chip coefficient memory.

B. Block Encoder

Block encoder, consisting of one context buffer, three coding passes and one block-encoding controller, is the key function unit in the Tier-1 part. Since each orientation contains only one block-encoding engine, there are totally three DMBs under processing at one time. Coefficients in DMBs are scanned during block encoding, starting from the most significant bit-planes to the least significant one, 1 bit-plane at a time. Inside a bit-plane, coefficient scanning begins at the top-left corner, the first four bits of the first column are coded, as shown in Fig. 9, and then the four bits of the next column. During the scanning, the statuses of coefficients are modified according to a certain rule. These statuses are recorded with a context buffer, which is the same size as the DMB. Each context in the buffer has sixteen bits, according to JPEG2000 standard.

When coding a coefficient, at most 1 sample and 8 neighbors are needed. In Fig. 9, if four coefficients in a row defined are coded, four coefficients and thirty-two contexts should be read, which require at least thirty-two read cycles and maybe a lot of write cycles according to the coding results. Apparently, a number of these cycles are redundant because neighbor contexts may be read and write more than once.

To solve this problem, a sixty-four-bit-width data bus for both DMB and context buffer is selected based on this scanning pattern so that a sample row can be read simultaneously and their neighbors no longer need to be read or written multiple times. Although thirty-six contexts will be stored in registers, as shown in Fig. 10, there are only three groups (twelve contexts) should be read in at run time. Because this method pre-reads the contexts of the next row, it not only greatly decreases the memory-accessing rate but also gains a throughput increasing. After all the passes in a DMB are coded, all the context buffers will be cleared to zeros, preparing for a new code-block encoding.

C. Arithmetic Encoder

The final step of Tier-1 encoding is the context-based binary AE. As mentioned earlier, binary decisions and their context labels are generated during the previous bit-plane scanning; and then provided to the AE as inputs. Apart from using on-chip memory, totally 19 registers are used in this AE implementation to represent the EBCOT contexts, with 7 bits each for a faster accessing rate. The lowest bit indicates which is the MPS (Most Probable Symbol) and the other 6 bits represent the index of the probability estimation table. These indices are used for accessing the probability estimation ROM, which is composed of Look Up Tables (LUT). The 2-time table-indexing pattern is shown in Fig. 11.

It is obvious that one of the fundamental advantages of the EBCOT is the optimized truncation; therefore the truncated length must be carefully computed in order to match the requirement of correctly decoding all symbols up to the truncation point. In this implementation, the truncated length of each pass is counted in the data output module, and then provided to the RD slope converter for further rate control logic. To ensure accurate decoding using this directly counted length, parallel termination mode [1] is selected in which every coding pass is flushed in AE, which can easily get the truncation length and add little complexity to the hardware implementation. The termination pattern is shown in Fig. 12.

IV. Architecture of EBCOT Tier-2 Rate Distortion Truncation

Although the conventional RD optimization strategy such as the one proposed by Jin LI [9] attains fairly good performance, it suffers from high computational costs since the coefficient bit modeling and AE process must be completed ahead of starting rate control processing, which demands a large on-chip buffer for the whole image tile storage in order to locate the appropriate truncation point set. Iterative computations are needed under such scheme as well. Current popular JPEG2000 implementations employ two methods to meet the rate control request:

1) To use a quantization coefficients instead of optimized truncation for rate control;
2) To leave the optimized truncation for Micro-Control Unit (MCU).

Apparently, both methods sacrifice flexibility and the second one even imposes much burden on system throughput and costs. Motivated by this problem, a novel rate control architecture, which executes optimized truncation in parallel with the process of AE, is devised. This architecture first stores code-stream and code-block information overheads of a code-block in separate buffers, then estimates RD slope for each truncation point and selects the monotonically decreasing subset. When all the RD slope metrics available, the optimal truncation point for current block can be easily determined. Referring to the information buffer to get truncated block length, the architecture accomplishes rate control by simply shift the buffer address to truncate the block stream. As a result, considerable reduction of computational costs can be achieved with avoiding iterative truncations. At the same time, buffer size is reduced from full-tile size to code-block size.

The proposed architecture for the Tier-2 RD truncation is shown in Fig. 13, which consists of four main functional parts: Code Truncation, Info Truncation, Buffer Arbiter and Packetization.

Data needed to construct a JPEG2000 code-stream can be divided into two categories: code and info. Code means those bytes generated by Tier-1 entropy encoder whereas info stands for code-block information necessary for decoder. Info consists of three parts: zero bit-plane, pass number and cumulative length for each pass in the code-block. RD slope for each pass is also necessary for optimized truncation. It is worth noting that only the truncated block length is necessary not all the cumulative lengths. Such characteristics enable some simplification of the implementation.

In this implementation, separate handlers for code and info simplify and clarify the architecture and minimize memory-locating efforts. The proposed method can be described as follows:

1) When block coding starts, info handler gets *zero_bit_plane* and *pass_number*;
2) For each *code_byte*, code handler writes it into buffer; address increases by 1;
3) When a pass finishes, *rd_slope* gives out the *pass_length* and info handler writes it into the buffer; address increases by 1;
4) When the block finishes, *rd_slope* gives out all slopes for every pass one by one in a monotonously decreasing order. Slopes of those passes not suitable for optimized truncation are set to zero;
5) Info handler compares these slopes with given threshold.
   if the *slope* > threshold then
   optimized_pass_number = *current_pass_number*;
   else if *slope* == zero then next; else if *slope* < threshold then break;
   Since RD slope is not necessary for constructing a code-stream, no memory access is needed;
6) Info handler moves address to the block beginning and writes the *optimized_pass_number*;
7) Info handler increases address by *optimized_pass_number* and gets *optimized_block_length* from buffer and then asserts *optimize_valid*;
8) When optimize_valid asserted, code handler gets optimized_block_length and shifts address to current_address - block_length + optimized_block _length;

Compared with the JPEG2000 Verification Model software architecture, this method provides the same RD performance by choosing the same truncation sets. Moreover, parallel-optimized truncation needs much small buffer and is exempt from searching the entire code-stream. Table VI provides numerical results to illustrate the performance of the proposed architecture under a variety of conditions. Results are presented for the well-known USC images, “Lenna” and “Barbara”, as well as one popular image from JPEG2000 test suite “woman”, which is substantially more complex and less blurred than the USC images.

V. Performance

A. The Chip Implementation of SCLA based DWT

The SCLA based DWT processor was fabricated in DONGBU 0.25 1P4M standard CMOS technology; 25k logic gates plus 93k bits on-chip SRAM were integrated in a 2.8mm x 2.8mm die area, with 0.8mW/MHz power consumption and 150MHz max processing frequency. This processor is implemented with both 5/3 reversible and 9/7 irreversible filters and the maximum tile resolution supported is 512 x 512 x 8 bits. The throughput of the DWT processor can reach 18Mbits/(MHz·s). i.e., under 100MHz system clock, this processor can transform 60 frames per second with image resolution of 1280 x 1024 x 24 bits. This chip had already successfully passed the Printed Circuit Board (PCB) based verification. Fig. 14 illustrates its microphotograph.

B. The Experimental Results and FPGA Verification of EBCOT and JPEG2000 Encoder

Implemented in DONGBU 0.25um 1P4M standard CMOS technology, the total scale of the EBCOT Tier-1 encoding part is 110k logic gates plus 400kbits on-chip SRAM, in which 388kbits are wavelet coefficients and 12kbits are contexts. However, without performing DMC scheme, at least 1Mbits coefficients have to be reserved; therefore, the DMC scheme reduces the 60% scale of the on-chip wavelet coefficient memory. The total scale of Tier-2 RD truncation part is 20k logic gates plus 24kbits on-chip SRAM. The throughput of this EBCOT processor can reach 5.3Mbits/ (MHz·s).

The proposed EBCOT architecture was combined with the chip-verified SCLA based DWT part to form a full JPEG2000 encoder, which had already passed the FPGA based verification and would be fabricated to a chip this year. The estimated scale and other important information of this encoder

are listed in TABLE VII.

Different from other implementations using quantization method such as [10], the proposed architecture adopts the truncation method in order to obtain three targets:

1) Better compression quality, since truncation based on RD slope is more refined compared with traditional quantization;
2) Accurate bit-rate control, which (95%) is higher than H. Yamauchi [10] (80%);
3) Allowance for multi-layer SNR scalable code-stream feature, which is impossible for quantization method.

These three targets are achieved at a sacrifice of the amount of computational efforts which put impedance on high throughput, for example, the throughput of this encoder (5.3Mbits/(MHz·s)) is a bit lower than H. Yamauchi [10] (720*480*16*30/27=6.1Mbits/(MHz·s)). However, compared with quantization method which reduces the entropy of wavelet coefficients before AE, implicit quantization adopted by this architecture leaves much more details and room for truncation to control the quality.

VI. Conclusion

A VLSI architecture of a full JPEG2000 encoder is proposed, which functionally consists of two major parts: SCLA based DWT and EBCOT. The SCLA based 2-D biorthogonal DWT, implemented with both 5/3 reversible and 9/7 irreversible filters, substantially reduces 50% and 42% multiplication computations respectively compared with the conventional LBI. The DWT core had already been fabricated into a chip, which is the first chip-implementation of SCLA in the world. The EBCOT Tier-1 part employs two functional schemes: the DMC scheme reduces 60% on-chip wavelet coefficient storage and the subband parallel-processing scheme greatly shortens the entropy encoding process times; while the Tier-2 part employs a novel architecture to reduce the scale of on-chip bit-stream buffering from full-tile size down to three-code-block size and considerably eliminate the iterations of the truncation. This EBCOT core had already been connected to the chip-verified DWT chip to form a full JPEG2000 encoder and passed the FPGA based verification. The proposed JPEG2000 encoder is fully compatible with ISO/IEC 15444-1. It can be widely used in the applications of next-generation digital camera, broadband PDA, etc.

References

[1] ISO/IEC IS 15444-1, "Information Technology – JPEG2000 image coding system – Part 1: Core coding system," ISO/IEC JTC1/SC29/WG1 (Dec. 2000), AMENDMENT 1: Code-stream restrictions (Mar. 2002). W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[2] D. Taubman, "High Performance Scalable Image Compression with EBCOT," IEEE Trans. Image Processing, Vol. 9, no. 7, pp. 1158-1170, June 2000.

[3] S. G. Mallat, "A theory for multi resolution signal decomposition: The wavelet representation," IEEE Trans. PAMI, Vol.11, pp.674-693, 1989

[4] H. Meng and Z. Wang, "Fast spatial combinative lifting algorithm of wavelet transform using the 9/7 filter for image block compression," Electronics Letters, Vol.: 36 issue: 21, pp. 1766-1767, 12 Oct 2000

[5] I. Daubechies and W.Sweldens, "Factoring wavelet Transforms into Lifting Steps," J.Fourier.Appl., Vol.4, pp.247-269, 1998.

[6] R. C. Calderbank, I. Daubechies, W. Sweldens and Boon-Lock Yeo, "Wavelet Transform that Map Integers to Integers," Applied and Computational Harmonic Analysis (ACHA), Vol.5, No.3 pp.332-369, 1998.

[7] C. Chrysafis and A. Ortega, "Line based, reduced memory, wavelet image compression, " IEEE Trans. On Image Processing. pp. 378-389, March 2000.

[8] C. Christopoulos, "JPEG2000 verification model 7.0 (technical description), " ISO/IEC JTC 1/SC 29/WG 1 N1684, April 25, 2000.

[9] J. Li, S. Lei, "An Embedded Still Image Coder with Rate-Distortion Optimization," IEEE Trans. On Image Processing, Vol. 8, No. 7, July 1999.

[10] H. Yamauchi, et al., "Image processor capable of block-noise-free JPEG2000 compression with 30 frames/s for digital camera applications," ISSCC Digest of Technical Papers, pp. 46-477, 2003.

Leibo LIU was born in Chongqing province, China in 1975. He received his B. S. degree in the Department of Electronics Engineering from Tsinghua University, Beijing, China, in 1999. He is currently pursuing his Ph. D. degree at the Institute of Microelectronic of Tsinghua University, Beijing, China. His research interests include wavelet transform, image compression, coding theory, VLSI design in communication and digital signal processing.

Ning CHEN was born in Guangdong province, China in 1978. He received his B. S. degree in the Department of Electronics Engineering from Tsinghua University, Beijing, China, in 1999. He is currently pursuing his M. S. degree at the Department of Electronics Engineering of Tsinghua University, Beijing, China. His research interests include VLSI for signal processing, VLSI design and testing.

Hongying MENG was born in Xi'an province, China in 1969. He received his B. S., M. S. and Ph. D. degree in the Department of Mathematics from Xi'an Jiaotong University, Xi'an, China, in 1991, 1994 and 1998 respectively. He is now the associate Professor of the Department of Mechanical Engineering in the University of Dundee, Scotland UK. His research interests include wavelet transform, coding theory and digital signal processing.

Li ZHANG was born in Shandong province, China in 1965. He received his B. S. and M. S. degree in the Department of Electronic Engineering from Tsinghua University, Beijing China, in 1987 and 1990 respectively. He became a associate Professor in the Department of Electronic Engineering in 1997. His major research fields are the image processing, signal processing and VLSI design

Zhihua WANG was born in Shandong province, China in 1960. He received his B. S., M. S. and Ph. D. in the Department of Electronic Engineering from Tsinghua University, Beijing China, in 1983, 1985 and 1990 respectively. He became a Professor in the Department of Electronic Engineering and vice Director of the Institute of Microelectronics of Tsinghua University in 1997 and 2000 respectively. He has served as the official member of the commission C of China National Commission of URSI starting from 1998 and the chairman of IEEE Solid-State Circuit Society Beijing Chapter. His major research fields are the design methodology and design automation of integrated circuits and systems, design of integrated circuits for communication and high-speed real-time signal processing.

Hongyi CHEN was born in Chongqing, China in 1941. He received his B. S in the Department of Electronic Engineering and M. S. in the Department of Computer Science both from Tsinghua University, Beijing China, in 1965 and 1982 respectively. He is the Director and Professor of the Institute of Microelectronics of Tsinghua University. He is also the senior member of China Institute of Electronic (CIE). His research interest includes a variety of the teaching and research activities in the field of VLSI design in communication, image coding, DSP algorithm, and the design of multimedia integrated circuits.

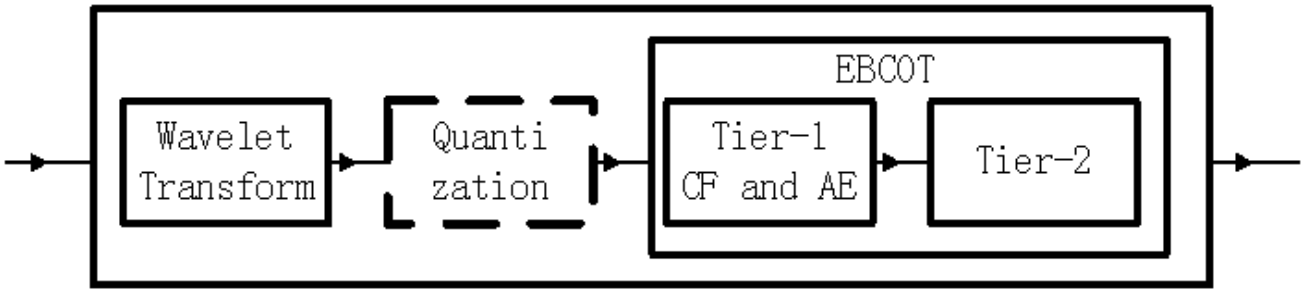

Fig. 1 JPEG2000 block diagram

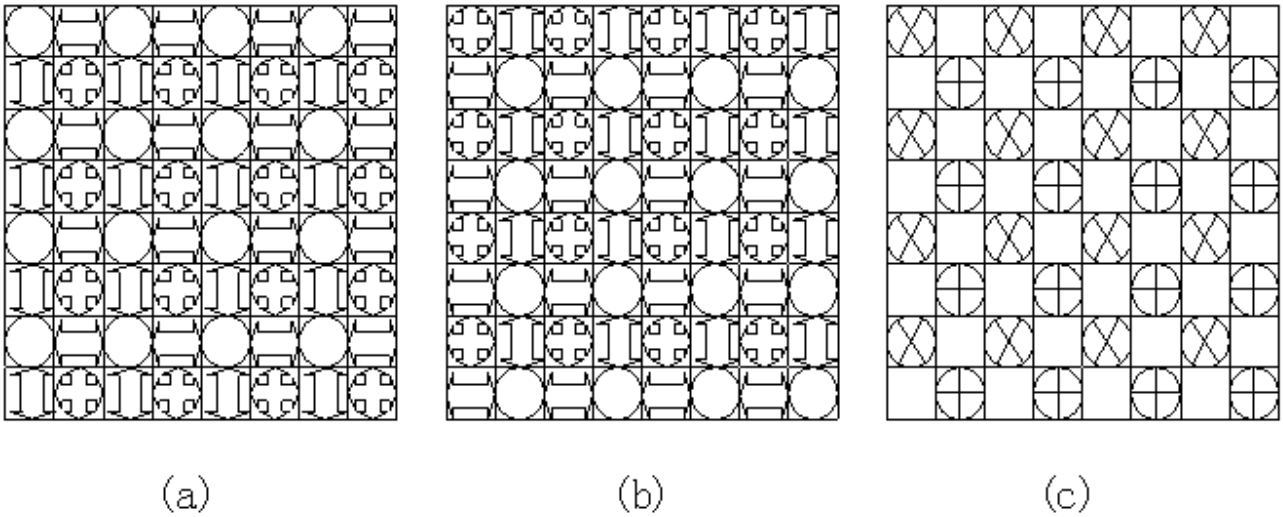

Fig. 2 SCLA operations on matrices *A,B,C,D,E*

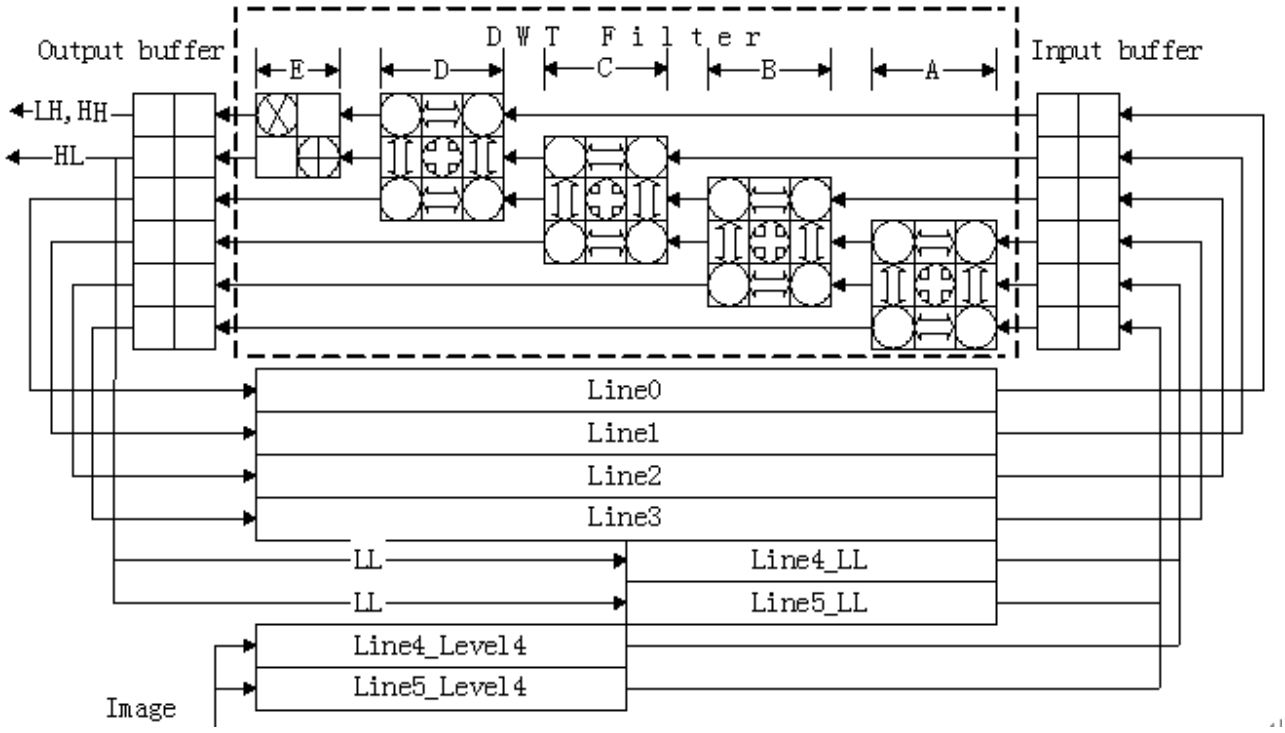

Fig. 3 Architecture the SCLA based DWT processor

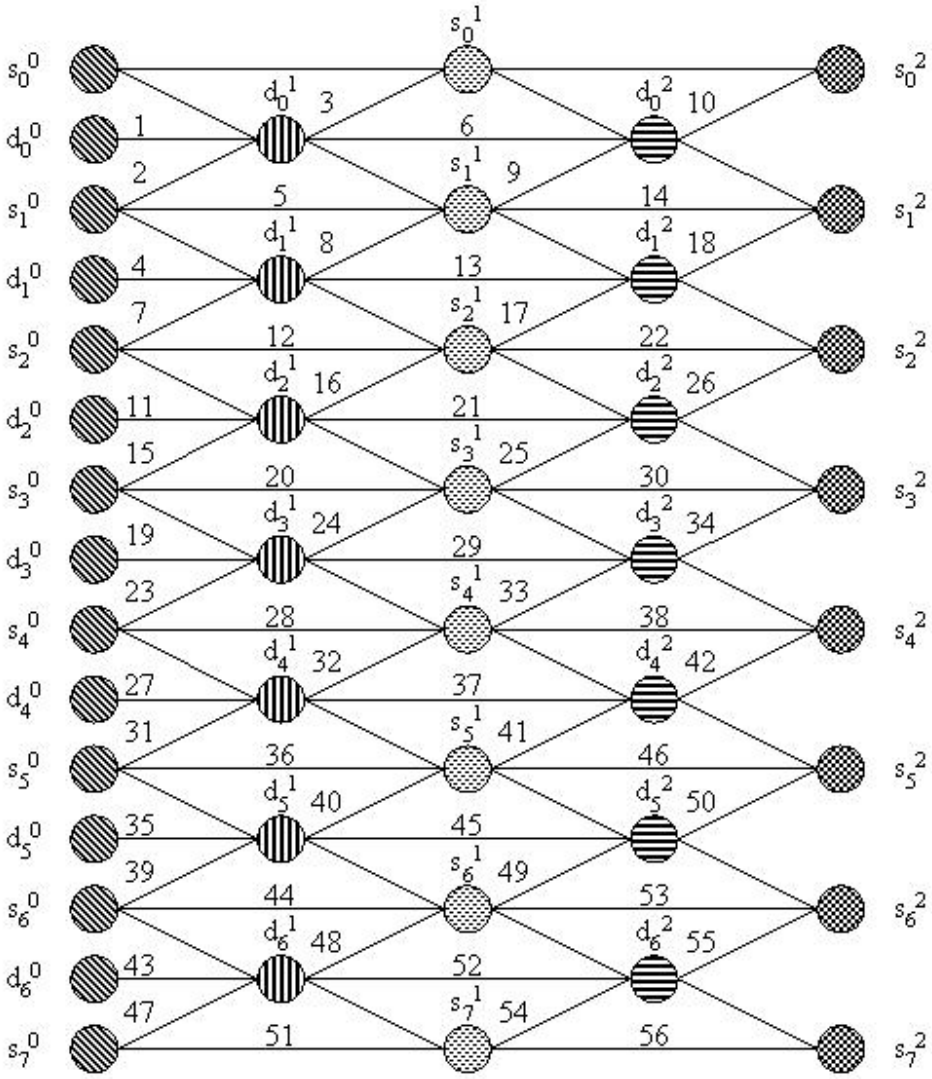

Fig. 4 Computing process of LBI

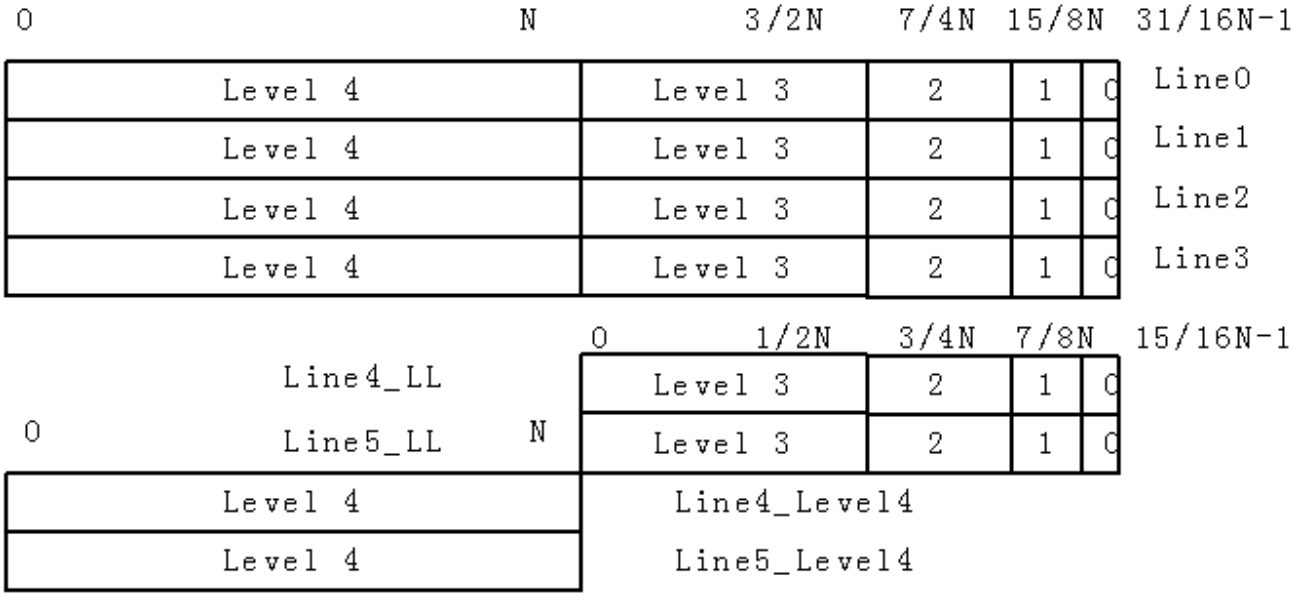

Fig. 5 Organization of the 6 line-buffer memories

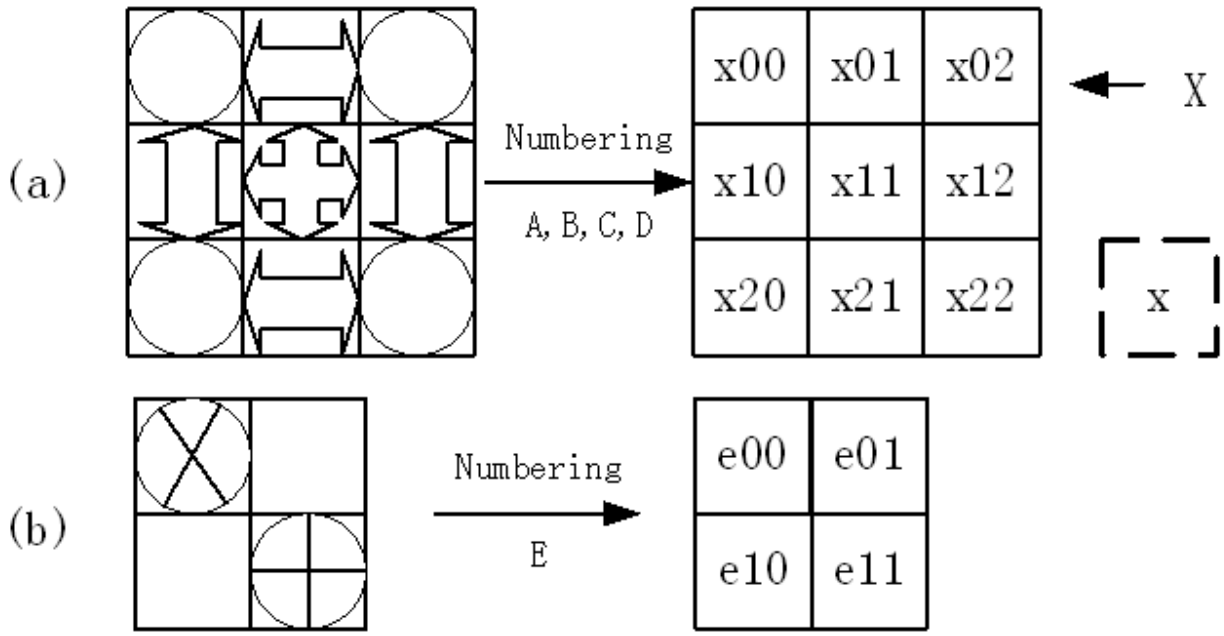

Fig. 6 The numbering of PE

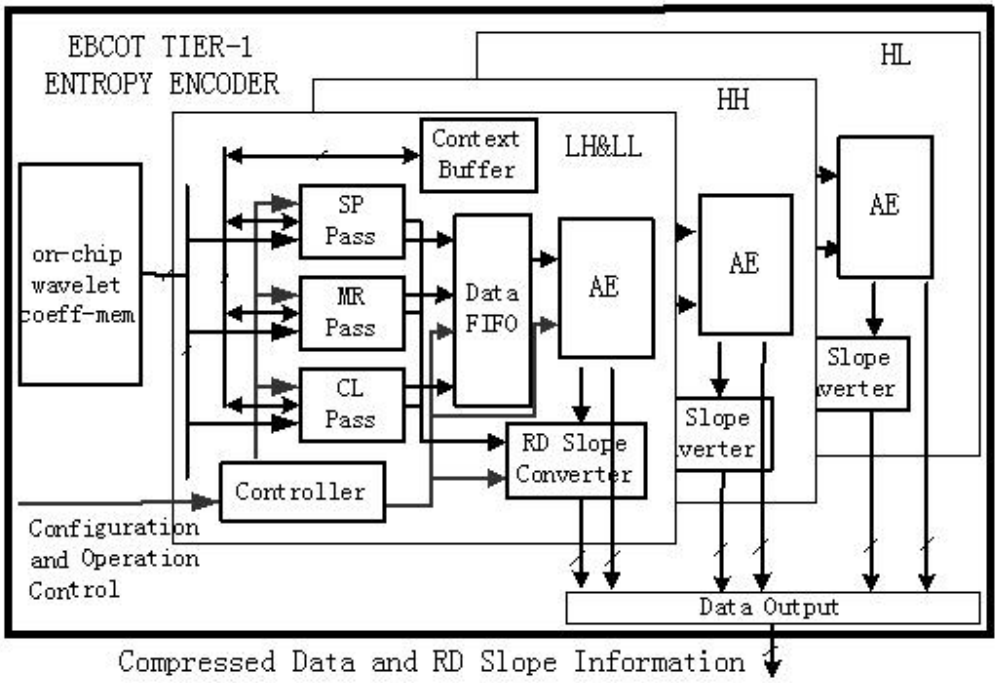

Fig. 7 Block diagram of Tier-1 entropy encoder

Full Waiting	Full Processing	Full Waiting	Full Waiting	Full Processing
Full Processing	Empty	Empty	Full Waiting	Not Full Buffering
Not Full Buffering	Empty	Full Waiting	Full Waiting	Full Waiting
Empty	Empty	Empty	Not Full Buffering	Empty

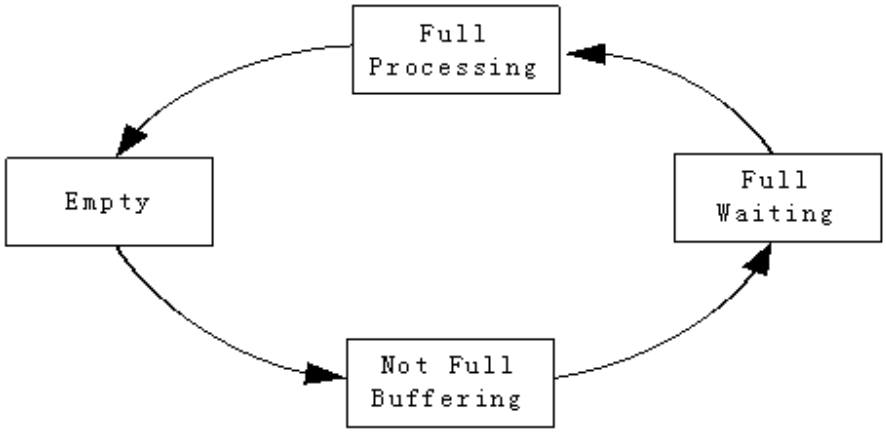

Fig. 8 DMC scheme

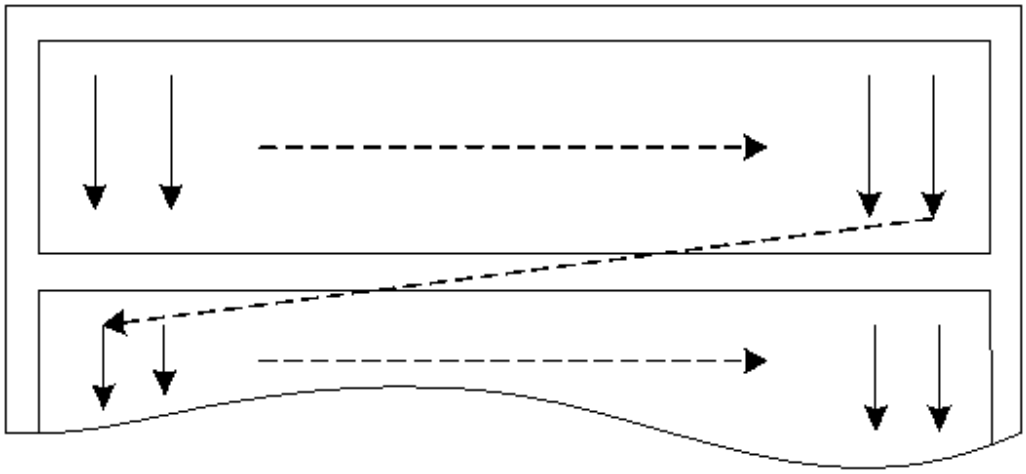

Fig. 9 Code-block scanning pattern in Tier-1

Fig. 10 Context buffer and read pattern

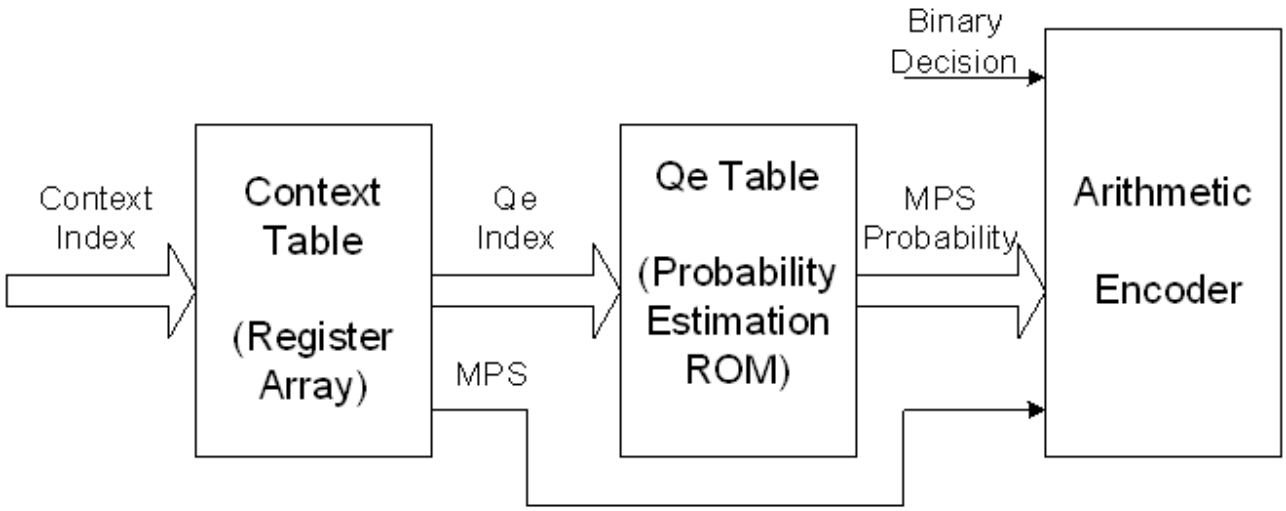

Fig. 11 Two-time table indexing in probability estimation

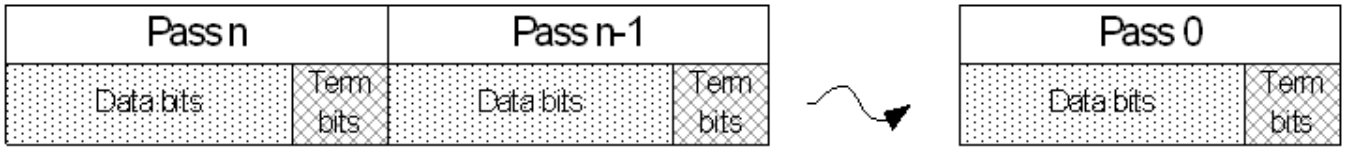

Fig. 12 Arithmetic encoder termination pattern

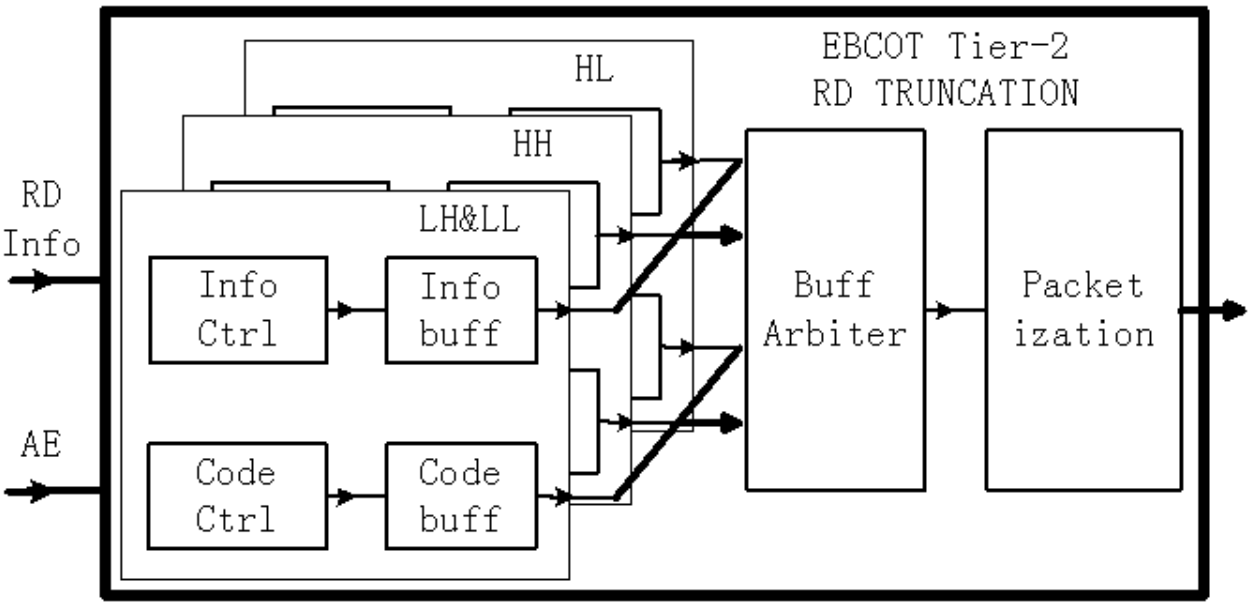

Fig. 13 Block diagram of Tier-2 RD Truncation

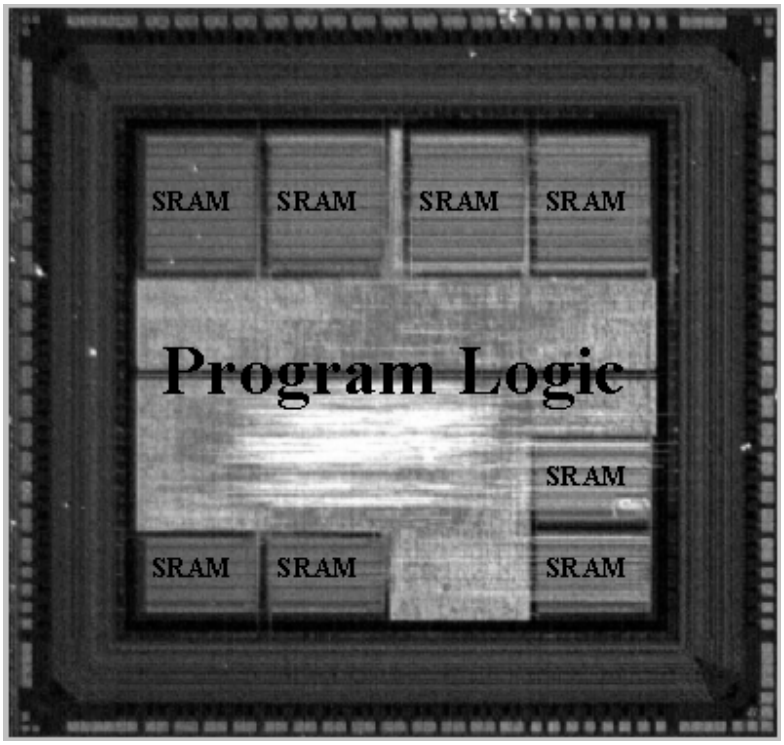

Fig. 14 SCLA based DWT chip microphotograph

Figure Caption

TABLE I

THE NUMBER OF MULTIPLICATIONS REQUIRED BY MALLAT, LBI AND SCLA WITH 9/7 FILTER

Algorithm	Level=1	Level=J
Typical Mallat	((N/2)x9+(N/2)x7))x2N =16NxN	$16/3x(4-4^{-J+1})xNxN$
Symmetrical Mallat	((N/2)x5+(N/2)x4))x2N =9N x N	$3x(4-4^{-J+1})xNxN$
LBI	((N/2)x4+N)x2N=6NxN	$2x(4-4^{-J+1})xNxN$
SCLA	(NxN)x3+Nx(N/2)=3.5NxN	$7/6x(4-4^{-J+1})xNxN$

TABLE II

THE NUMBER OF MULTIPLICATIONS REQUIRED BY MALLAT, LBI AND SCLA WITH 5/3 FILTER

Algorithm	Level=1	Level=J
Typical Mallat	(N/2x5+N/2x3)x2N=8NxN	$8/3x(4-4^{-J+1})xNxN$
Symmetrical Mallat	(N/2x3+N/2x2)x2N=5NxN	$5/3x(4-4^{-J+1})xNxN$
LBI	(N/2x2+N)x2N=4NxN	$4/3x(4-4^{-J+1})xNxN$
SCLA	3/2NxN+NxN/2=2NxN	$2/3x(4-4^{-J+1})xNxN$

TABLE III
NUMBER OF MULTIPLICATIONS WITH DIFFERENT LINE-BUFFER MEMORIES

Algorithm	6 line-buffer	5 line-buffer
LBI	6xNxN	10xNxN
SCLA	3.5xNxN	5.5xNxN

TABLE IV

PSNR (DB) COMPARISON WITH DIFFERENT COEFFICIENT PRECISIONS

(CONSTANTS PRECISION=32BITS)

Precision	Bit-Rate=0.5bpp	Bit-Rate=0.25bpp	Bit-Rate=0.125bpp
JPEG2000 VM7.0 [8] software	37.2687	34.0825	30.9460
19 bits	37.2788	34.0783	30.9446
18 bits	37.2383	34.0717	30.9649
17 bits	37.1733	34.0406	30.9286
16 bits	36.8708	33.8945	30.8604

TABLE V

PSNR (DB) COMPARISON WITH DIFFERENT CONSTANTS PRECISIONS

(WAVELET COEFFICIENT PRECISION=17BITS)

Precision	Bit-Rate=0.5bpp	Bit-Rate=0.25bpp	Bit-Rate=0.125bpp
32 bits	37.1733	34.0406	30.9286
14 bits	37.1288	34.0185	30.9417
13 bits	37.1630	34.0357	30.9537
12 bits	36.2269	33.5604	30.6938

TABLE VI

PSNR(DB) RESULTS, FOR VARIOUS IMAGES (512 X 512 X 8) AND BIT-RATES(BPP)

Bit-Rate	Lenna		Barbara		Woman	
	VM7.0 Software	Proposed Architecture	VM7.0 Software	Proposed Architecture	VM7.0 Software	Proposed Architecture
0.0625	28.2892	28.2576	23.4327	23.3903	25.7014	25.6745
0.125	30.9460	30.9537	25.5735	25.4935	27.4922	27.3895
0.25	34.0825	34.0357	28.5832	28.4929	30.1825	30.1283
0.5	37.2687	37.1630	32.4846	32.4047	33.8417	33.7725
1.0	40.6013	40.5733	37.3729	37.3092	38.6913	38.6092

TABLE VII

ESTIMATED SCALE AND OTHER IMPORTANT INFORMATION OF THE JPEG2000 ENCODER

Technology/Library	DONGBU 0.18um 2P5M standard CMOS technology / Artisan
Logic Gates	180k logic gates
On-Chip Memory	550k bits SRAM
Die Area	Less than 20mm^2
Power Consumption	Less than 4.5mW/MHz
Peak Frequency	DWT: 100MHz, EBCOT: 200MHz
Throughput	5.3Mbits/ MHz • s (Bottleneck: EBCOT) (17fames/s can be reached with image resolution of 1280 x 1024 x 24bits, if DWT employs 50MHz and EBCOT employs 100MHz working clock)
Wavelet Filters	(5,3) and (9,7) Daubechies filters
Decomposition Method	Typical 5-level Mallat
Tile Resolution	512 x 512 x 8bits
Code Block Size	32 x 32, 16 x 16
Max Image Resolution	1800x 1600 x 24bits
Compression Ratio	Arbitrary

TABLE CAPTION

TABLE I THE NUMBER OF MULTIPLICATIONS REQUIRED BY MALLAT, LBI AND SCLA WITH 9/7 FILTER
TABLE II THE NUMBER OF MULTIPLICATIONS REQUIRED BY MALLAT, LBI AND SCLA WITH 5/3 FILTER
TABLE III NUMBER OF MULTIPLICATIONS WITH DIFFERENT LINE-BUFFER MEMORIES
TABLE IV PSNR (DB) COMPARISON WITH DIFFERENT COEFFICIENT PRECISIONS (CONSTANTS PRECISION=32BITS)
TABLE V PSNR (DB) COMPARISON WITH DIFFERENT CONSTANTS PRECISIONS (WAVELET COEFFICIENT PRECISION=17BITS)
TABLE VI PSNR(DB) RESULTS, FOR VARIOUS IMAGES (512 X 512 X 8) AND BIT-RATES(BPP)
TABLE VII ESTIMATED SCALE AND OTHER IMPORTANT INFORMATION OF THE JPEG2000 ENCODER