

Area Efficient Architecture for the Embedded Block Coding in JPEG 2000

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

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

Abstract— An area efficient architecture for the embedded block coding in JPEG 2000 is implemented on a 1.23 mm^2 die using $0.18 \text{ }\mu\text{m}$ CMOS technology. This chip can support 16.7 MS/s lossless encoding. The area of the proposed architecture is only $\frac{1}{6}$ of the conventional architectures while the throughput is the same as others. The proposed architecture has the highest performance comparing with other existing architectures according to the experimental results.

I. INTRODUCTION

JPEG 2000 [1] [2] [3] [4] is well-known for its excellent coding performance and numerous features [5], such as region of interest, scalability, error resilience, etc. All these powerful tools can be provided by a unified algorithm in a single JPEG 2000 codestream. For example, an image can be losslessly coded for storage and then retrieved at different bit-rates by transcoding. Transcoding of the JPEG 2000 codestream can be done by parsing, reordering, and truncating the original codestream. However, the high computational complexity that gives such excellent performance and rich features correspondingly restricts real-time applications of JPEG 2000. In this paper, we proposed an area efficient architecture for the embedded block coding in JPEG 2000.

JPEG 2000 is a new still image coding standard, which is entirely different from the JPEG [6]. The functional block diagram of the JPEG 2000 encoder is shown in Fig. 1. The Discrete Wavelet Transform (DWT) is adopted as the transform algorithm of JPEG 2000. The DWT has several features that are better than the Discrete Cosine Transform (DCT), such as better coding performance, easy rate control, fully embedded coding, etc. After the DWT, a uniform scalar quantization is applied to the transformed coefficient. The entropy coding algorithm of JPEG 2000 is the Embedded Block Coding with Optimized Truncation (EBCOT) [7] [8]. It is a two-tiered algorithm, in which the Embedded Block Coding (EBC) is the tier-1 and the Rate-Distortion Optimization (RDO) is the tier-2. The EBC is based on a context-adaptive binary Arithmetic Encoder (AE). By optimized truncation of the embedded bit streams, the RDO optimizes the coded image quality at a given target bit rate.

The Embedded Block Coding (EBC) is the most complicated part of JPEG 2000 [9] and is the bottleneck for real-time applications. Therefore, many EBC architectures are proposed [9] [10] [11] [12] to solve the problem. Lian et al. [9] proposed the first EBC architecture, which implements the default mode of the EBC algorithm. In this architecture, three techniques are used to skip unnecessary check-point, and the processing cycles are reduced by 60% comparing to [2]. To reduce the hardware cost, Hsiao et al. [10] proposed a memory-saving architecture that reduces the memory requirement by 4 *Kbits* (*Kb*). On the other hand, Chiang et al. [11] proposed a pass-parallel architecture to increase the processing rate based on the parallel mode. The processing cycles are reduced by 67% comparing to [2]. The above three architectures process a code-block bit-plane

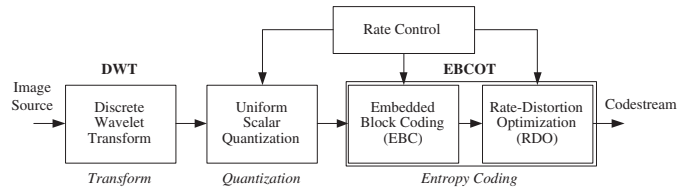


Fig. 1. Functional block diagram of the JPEG 2000 encoder. The JPEG 2000 encoder comprises the discrete wavelet transform, the uniform scalar quantization, and the embedded block coding with optimization truncation algorithm.

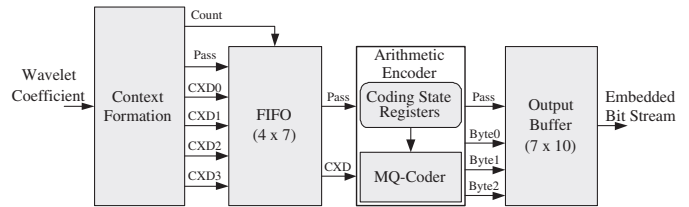


Fig. 2. Block diagram of the proposed EBC architecture. There are three major modules: the context formation module, the FIFO module, and the arithmetic encoder.

by bit-plane. Fang et al. proposed a parallel architecture to process a coefficient per cycle. All the above architectures occupies more than 5.0 mm^2 silicon area in $0.35 \text{ }\mu\text{m}$ technology, which is too large.

In this design, we proposed an area efficient EBC architecture for JPEG 2000. This architecture is based on the new context formation algorithm, which can accomplish the context formation without storing any state variables. All the state variables are computed on-the-fly while a coefficient is read. Besides, the data flow and controls are simplified by using the proposed algorithm. This architecture can encode all the three coding passes in a bit-plane in one scan. Therefore, it features high throughput and low area cost for the embedded block coding in JPEG 2000.

II. PROPOSED ARCHITECTURE

Figure 2 shows the block diagram of the low cost Embedded Block Coding (EBC) architecture. It contains four main modules: the Context Formation (CF) module, the First-In First-Out (FIFO) module, the Arithmetic Encoder (AE) module, and the Output Buffer (OB) module. The input is the wavelet coefficient and the output is the embedded bit stream. The OB module is used to reduce the number of output ports while maintaining the same throughput. To prevent the buffer overflow, the number of the registers are chosen as 7.

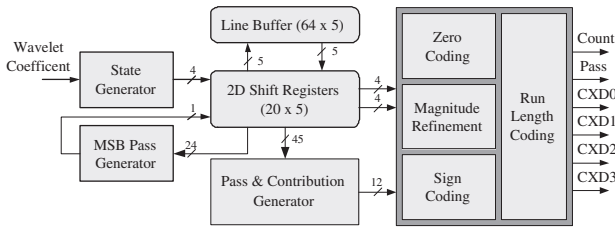


Fig. 3. Block diagram of the context formation module. A 2D shift register bank is used to fit the dataflow with the scan order defined in the JPEG 2000 encoder.

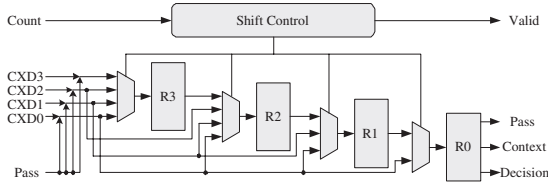


Fig. 4. Block diagram of the FIFO module. There will be 0 ~ 4 inputs and one output per cycle.

A. Context Formation

Based on the algorithm we proposed in [13], the low cost CF module is obtained as shown in Fig. 3. The state variables, usually implemented as 8 Kb memory in conventional CF architecture, are computed on-the-fly by the state generator while reading the wavelet coefficient. Therefore, the 8Kb state memory are eliminated. The resulted state variables as well as sign and magnitude bit of coefficients are fed into the 2D shift registers to fit the scan order defined in the JPEG 2000 standard. The MSB coding pass of the scanned coefficient is then generated by the MSB pass generator, and merged into the data flow in the 2D shift registers. A line buffer with size 64×5 is required to store the last row of previous stripe.

The coding pass and significant contributions are generated in the pass & contribution generator for the context formation. In order to cope with the special run-length code, the contexts generated by the zero coding, magnitude refinement, and sign coding modules are buffered for three cycles. After deciding whether the run-length code is used or not, the final ConteXt Decision (CXD) pairs are generated by the run length coding module. Note that various number of CXD pairs may output in one cycle. The extreme case occurs in the first sample coefficient of a column when the run-length coding fails. Four CXD pairs are generated in this case: one run-length CXD pair, two uniform CXD pairs, and one sign coding CXD pair. The coding pass information is also required since the three coding passes are processed in parallel.

B. FIFO

The FIFO module is used to smoothen the input data flow of the AE module. This is because the CF module generates various number of CXD pairs, from 0 to 4, per cycle. However, the AE module can only process one CXD pair per cycle. Thus, the use of the FIFO module can alleviate the problem arisen from the throughput mismatch between the CF and AE modules. As shown in Fig. 4, there are four registers in the FIFO, in which each register has seven bits comprising two bits of coding pass and five bits of CXD pair.

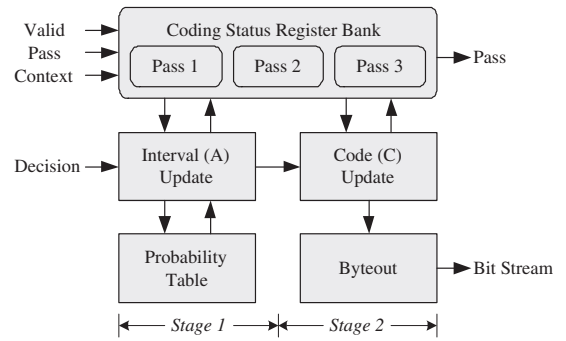


Fig. 5. Block diagram of the AE module. It has three suits of the coding status registers and one suit of processing elements.

C. Arithmetic Encoder

In the proposed architecture, the three coding passes in a bit-plane is proposed in parallel. Thus, there are three embedded bit streams to be processed by the AE in parallel. Therefore, the Pass Switching AE (PSAE) [11] is adopted. By using the PSAE architecture, only one suit of processing unit is required to encode three coding passes in parallel as shown in Fig. 5. Two stages of pipeline are used in the proposed architecture. In this architecture, the index of the probability table can be updated in the first stage of pipeline. Thus, no probability look ahead is required and the hardware cost is reduced. Moreover, the re-normalization and the byteout operation can be finished in one cycle, which can ensure that one CXD pair can be consumed by the AE module.

III. CHIP IMPLEMENTATION

The EBC architecture is implemented on a 1.23 mm^2 die using $0.18 \mu\text{m}$ CMOS 1P6M technology. The detailed design flow and test considerations are elaborated in the following sections.

A. Design Flow

Figure 6 shows the design flow for the chip. It's quite similar to a standard cell-based design flow. For the architecture design, we use Verilog Hardware Description Language (HDL) to describe the hardware. After a plenty of Verilog-XL simulations, we synthesize the design by using Synopsys Design Compiler with the Artisan cell library. The total gate count is 10 K gates and the on-chip SRAM requirement is 320 bits. The detailed gate count distribution is shown in Table I. The synthesis results are compared with the target specification to see whether it is met or not. After confirming that the specification is met, various Design for Testability (DfT) techniques are considered and applied to the design, which will be described in more detail in Sec. III-B.

After the DfT stage, we use Verilog-XL to perform the gate-level simulation to make sure that the target specification is met. Moreover, the power consumption is also estimated at this stage by using the Synopsys PrimePower. The estimated power consumption is about 26.4 mW at 100 MHz . By use of the estimated power consumption, we could perform the power-plan of the chip to ensure that the power supply is enough and the power density is equally distributed on the chip.

We use Synopsys Astro as the backend tool that performs automatic place-and-route. In order to guarantee the timing specification is met, we use timing driven place-and-route of Astro. This is because the wire delay becomes non-negligible in $0.18 \mu\text{m}$ technology. Again, we use Verilog-XL to perform post-layout gate-level simulation to

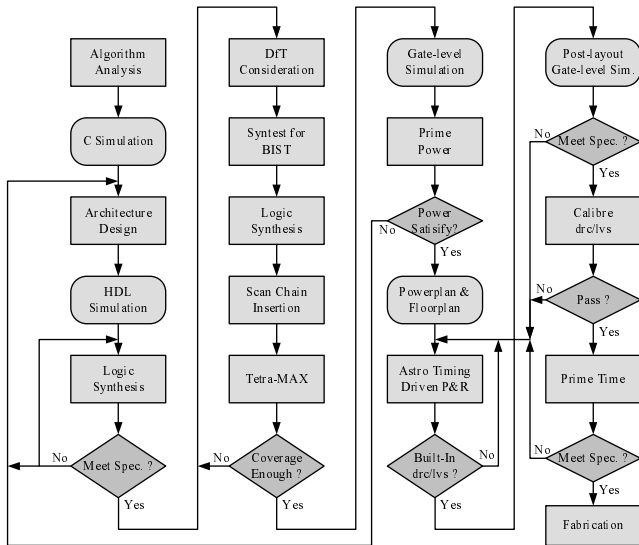


Fig. 6. Design flow for the EBC chip.

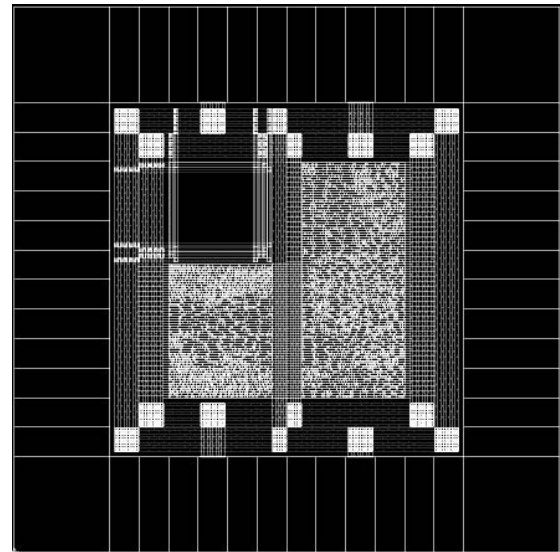


Fig. 7. Layout view of the chip.

TABLE I
HARDWARE REQUIREMENTS OF THE PROPOSED ARCHITECTURE

Module	Gate Count (NAND2)	Memory (bits)
CF	1937	320
FIFO	400	0
AE	6596	0
OB	865	0
Control	658	0
Total	10456	320

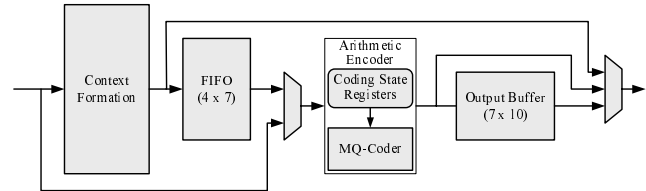


Fig. 8. Diagram of ad-hoc DfT technique.

confirm the functionality and the timing after clock tree synthesis, gate sizing, and buffer insertion done by Astro. After that, to check the design rule and electric connectivity, we use Mentor Graphics Calibre to perform DRC and LVS. As the last step, we use Synopsys PrimeTime to evaluate the timing by using static timing analysis. Figure 7 shows the layout view. The core size and the target operating frequency of the chip are 0.22 mm^2 and 100 MHz , respectively.

B. Test Considerations

Besides the architecture, Design for Testability (DfT) is very important for chip implementation. In this design, we use three techniques: ad-hoc, Build-In-Self-Test (BIST), and scan chain.

The ad-hoc technique can increase the observability and controllability of the design. In the ad-hoc mode, we make input signals directly connected to inputs of certain module and observe the output signals from the output port as shown in Fig. 8. By doing this, the module can be fully controlled and tested to see whether it is functionally work or not. In the ad-hoc modes, we can control and observe the CF and the AE modules. Therefore, if the chip does not function correctly, we can find the module in which error occurs. A few multiplexers and registers are spent for the ad-hoc testing mode, which is quite efficiency.

It is very efficient to use the BIST for the testing of SRAM. For the BIST algorithm, we adopted MARCH algorithm [14] due to its effectiveness. We use one BIST controller for the 64×5 bits single port SRAM. The total gate counts for the BIST resources is only about 330 gates.

The most important DfT technique is the scan test. By the scan

test, a fault can almost always detectable if the fault coverage is high enough. We use one scan path to connect 760 registers since the number of register is not large. Besides, the SRAM are bypassed at scan mode to increase the controllability of logics following them. According to the report of Tetra-MAX, the number of faults are 37878 and the test coverage is 99.80%.

IV. EXPERIMENTAL RESULTS

A. Chip Feature

The chip has been fabricated and received in May 2004 by the UMC. Figure 9 shows the die micrograph. After testing, the chip is fully functional as expected. Moreover, the power consumption is 18.2 mW @ 1.8 V supply voltage at 100 MHz operating frequency, which is lower than estimated. The supply voltage can be scaled down to 1.4 V and the power consumption is 10.7 mW . The chip can support $16.7 \text{ M Samples/sec}$ encoding at 100 MHz operating frequency. The detailed specification of the chip is shown in Table II.

B. Comparison

In this section, we compare the proposed EBC architecture with others. The hardware requirement of various EBC architectures are summarized in Table III. Except Fang's architecture [12], all the architectures are sequential architectures that process a code-block in a bit-plane by bit-plane manner. Fang's architecture is a parallel architecture that process a DWT coefficient per cycle. For the sequential architectures, the processing rate depends on the number of non-zero bit-planes of a code-block. In this comparison, the number of non-zero bit-planes is assumed to be six, which is an average value

TABLE III
COMPARISON OF VARIOUS EMBEDDED BLOCK CODING ARCHITECTURES

	Technology (μm)	Gate Count (NAND2)	Memory (bits)	Throughput ($\frac{S}{\text{cycle}}$)	Area (mm^2)	Performance Index ($\frac{S}{\text{cycle}\cdot\text{mm}^2}$)
Lian's [9]	0.35	19000	12288	0.128	6.49	0.0197
Hsiao's [10]	0.35	21589	8192	0.128	5.52	0.0232
Chiang's [11]	0.35	23927	8192	0.167	5.20	0.0321
Fang's [12]	0.25	91758	768	1.000	5.50 [†]	0.1818
Proposed	0.18	10052	320	0.167	0.92	0.1875

[†] Normalized by doubling per technology generation.

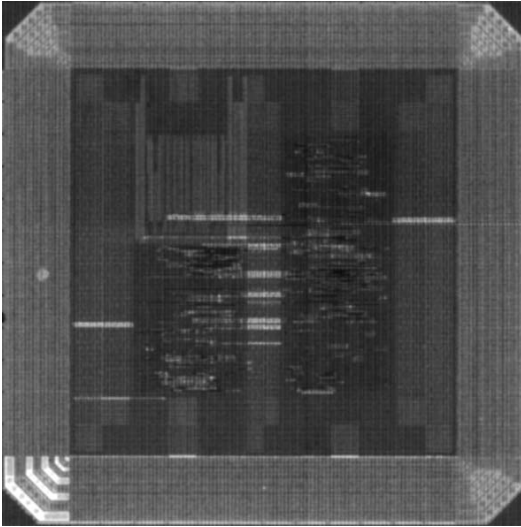


Fig. 9. Die micrograph.

TABLE II
SPECIFICATION OF THE DEVELOPED EMBEDDED BLOCK CODING CHIP.

Item	Description
Technology	UMC 0.18 μm 1P6M CMOS
Pad/Core Voltage	3.3/1.4 V
Core Area	0.48 \times 0.47 mm^2
Logic Gates	10 K (2-input NAND gate)
SRAM	64 \times 5 bits
Operating Frequency	100 MHz
Power Consumption	10.7 mW
Package	CLCC68
Processing Rate	16.7 M Sample/sec

of nature images. The unit of the processing rate is defined as Samples per cycle (S/cycle). By Table III, the gate count of the proposed architecture is half of that of the other sequential architectures and is only $\frac{1}{9}$ of that of the parallel architecture. The memory requirement of the proposed architecture is only 4% of that of the other sequential architectures.

In order to make a fair comparison, the Performance Index (PI) defined in [15] is adopted to compare these architectures. The PI is defined as processing rate per unit area ($\frac{S}{\text{cycle}\cdot\text{mm}^2}$). For various technologies, the area is normalized by doubling the area per technology generation. Table III summarizes the comparisons of various EBC architectures by this metric. By Table III, the proposed architecture is six times better than other sequential architectures and is comparable to the parallel architecture. This mainly comes from the low cost context formation.

V. CONCLUSION

In this design, an area efficient chip for the embedded block coding in JPEG 2000 is presented, which can support 16.7 M Sample/sec encoding on a 1.23 mm^2 die with 0.18 μm process. A new scheme is proposed to accomplish context formation by computing all the state variables on-the-fly. Therefore, a total number of 8 Kb state variable memory is eliminated. The area of the proposed architecture is only $\frac{1}{6}$ of other sequential architectures while the throughput is the same as others. According to the experimental results, the proposed architecture is the most cost-effective among existing architectures.

REFERENCES

- [1] *JPEG 2000 Part 1: Final Draft International Standard (ISO/IEC FDIS15444-1)*. ISO/IEC JTC1/SC29/WG1 N1855, Aug. 2000.
- [2] *JPEG 2000 Verification Model 7.0 (Technical Description)*. ISO/IEC JTC1/SC29/WG1 N1684, Apr. 2000.
- [3] *JPEG 2000 Requirements and Profiles*. ISO/IEC JTC1/SC29/WG1 N1271, Mar. 1999.
- [4] D. Taubman and M. Marchellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.
- [5] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Mag.*, vol. 18, no. 5, pp. 36–58, Sept. 2001.
- [6] *JPEG: Still Image Data Compression Standard*. W. Pennebaker and J. Mitchell, New York: Van Nostrand Reinhold, 1992.
- [7] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.
- [8] *EBCOT: Embedded Block Coding with Optimized Truncation*. ISO/IEC JTC1/SC29/WG1 N1020R, Oct. 1999.
- [9] C.-J. Lian, K.-F. Chen, H.-H. Chen, and L.-G. Chen, "Analysis and architecture design of block-coding engine for EBCOT in JPEG 2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 3, pp. 219–230, Mar. 2003.
- [10] 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 and Systems*, vol. 5, Scottsdale, Arizona, May 2002, pp. 133–136.
- [11] J.-S. Chiang, Y.-S. Lin, and C.-Y. Hsieh, "Efficient pass-parallel for EBCOT in JPEG 2000," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 1, Scottsdale, Arizona, May 2002, pp. 773–776.
- [12] H.-C. Fang, T.-C. Wang, C.-J. Lian, T.-H. Chang, and L.-G. Chen, "High speed memory efficient ebcot architecture for JPEG2000," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, Bangkok, Thailand, May 2003, pp. 736–739.
- [13] H.-C. Fang, Y.-W. Chang, and L.-G. Chen, "Area efficient architecture for the embedded block coding in JPEG 2000," in *Proc. IEEE International Midwest Symposium on Circuits and Systems*, Hiroshima, Japan, July 2004.
- [14] K.-L. Cheng, M.-F. Tsai, and C.-W. Wu, "Neighborhood pattern-sensitive fault testing and diagnostics for random-access memories," *IEEE J. Technol. Computer Aided Design*, vol. 21, no. 11, pp. 1328–1336, 11 2002.
- [15] H.-C. Fang, C.-T. Huang, Y.-W. Chang, T.-C. Wang, P.-C. Tseng, C.-J. Lian, and L.-G. Chen, "81 MS/s JPEG 2000 single-chip encoder with rate-distortion optimization," in *ISSCC Dig. Tech. Papers*, San Francisco, CA, Feb. 2004, pp. 328–329.