5

Full HD JPEG XR Encoder Design for Digital Photography Applications

Ching-Yen Chien*, Sheng-Chieh Huang*, Chia-Ho Pan, and Liang-Gee Chen
DSP/IC Design Lab, Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University
**Sense/TCM SOC Lab, Department of Electrical and Control Engineering, National Chiao-Tung University*
Taiwan, R.O.C.

1. Introduction

Multimedia applications, such as radio, audio, camera phone, digital still camera, camcoder, and mobile broadcasting TV, are more and more popular in our life as the progress of image sensor, communication, VLSI manufacture, and image/video coding standards. With rapid progress of image sensor, display devices, and computing engines, image coding standards are used in the digital photography application everywhere. It has been merged together with our life such as camera phone, digital still camera, blog and many other applications.

Many advanced multimedia applications require image compression technology with higher compression ratio and better visual quality. High quality, high compression rates of digital image and low computational cost are important factors in many areas of consumer electronics, ranging from digital photography to the consumer display equipments such as digital still camera and digital frame. These requirements usually involve computationally intensive algorithms imposing trade-offs between quality, computational resources, and throughput.

For high quality of digital image applications, the extension of color range has becoming more important in the consumer product. In the past, the digital cameras and the display equipments in the consumer market typically had 8 bits information per channel. Today the condition is quite different. In the consumer market, digital cameras and the desktop display panels also have at least 12 bits of information per channel. If the information per channel of digital image is still compressed into 8 bits, 4 or more bits of information per channel are lost and the quality of the digital image is limited. Due to the improvement of the display equipments, the JPEG XR is designed for the high dynamic range (HDR) and the high definition (HD) photo size. JPEG XR which is already under organized by the ISO/IEC Joint Photographic Experts Group (JPEG) Standard Committee is a new still image coding standard and derived from the window media photo (Srinivasan et al., 2007; Srinivasan et al., 2008; Schonberg et al., 2008). The goal of JPEG XR is to support the greatest possible level

of image dynamic range and color precision, and keep the device implementations of the encoder and decoder as simple as possible.

For the compression of digital image, the Joint Photographic Experts Group, the first international image coding standard for continuous-tone natural images, was defined in 1992 (ITU, 1992). JPEG is a well-known image compression format today because of the population of digital still camera and Internet. But JPEG has its limitation to satisfy the rapid progress of consumer electronics. Another image coding standard, JPEG2000 (ISO/IEC, 2000), was finalized in 2001. Differed from JPEG standard, a Discrete Cosine Transform (DCT) based coder, the JPEG2000 uses a Discrete Wavelet Transform (DWT) based coder. The JPEG2000 not only enhances enhances the compression, but also includes many new features, such as quality scalability, resolution scalability, region of interest, and lossy/lossless coding in a unified framework. However, the design of JPEG2000 is much complicated than the JPEG standard. The core techniques and computation complexity comparisons of these two image coding standard are shown in (Huang et al., 2005).

For satisfaction of the high quality image compression and lower computation complexity, the new JPEG XR compression algorithm is discussed and implemented with the VLSI architecture. JPEG XR has high encoding efficiency and versatile functions. The XR of JPEG XR means the extended range. It means that JPEG XR supports the extended range of information per channel. The image quality of JPEG XR is nearly equal to JPEG 2000 with the same bit-rate. The computation complexity is much lower than JPEG2000 as shown in Table 1.

The efficient system-level architecture design is more important than the module design since system-level improvements make more impacts on performance, power, and memory bandwidth than the module-level improvements. In this chapter, the new JPEG XR compression standard is introduced and the analysis and architecture design of JEPG XR encoder are also proposed. Comparison was made to analyze the compression performance among JPEG2000 and JPEG XR. Fig. 1 is the peak signal-to-noise ratio (PSNR) results under several different bitrates. The test color image is 512x512 Baboon. The image quality of JPEG XR is very close to that of JPEG2000. The PSNR difference between JPEG XR and JPEG2000 is under 0.5dB. Fig. 2 shows the subjective views at 80 times compression ratio. The block artifact of JPEG image in Fig. 2 is easily observed, while the JPEG XR demonstrates acceptable qualities by implementing the pre-filter function. For the architecture design, a 4:4:4 1920x1080 JPEG XR encoder is proposed. From the simulation and analysis, entropy coding is the most computationally intensive part in JPEG XR encoder. We first proposed a timing schedule of pipeline architecture to speed up the entropy encoding module. To optimize memory bandwidth problem and maximize the silicon area efficiency, we also proposed a data reuse skill to solve this problem. The data reuse skill can reduce 33% memory bandwidth form the memory access. The hardware design of JPEG XR encoder has been implemented by cell-based IC design flow. The encoder design is also verified by FPGA platform. This JPEG XR chip design can be used for digital photography applications to achieve low computation, low storage, and high dynamical range features.

Technologies	Operations(GOPs)
JPEG2000	4.26
JPEG XR	1.2

Table 1. Machine cycles comparison between JPEG XR and JPEG 2000.

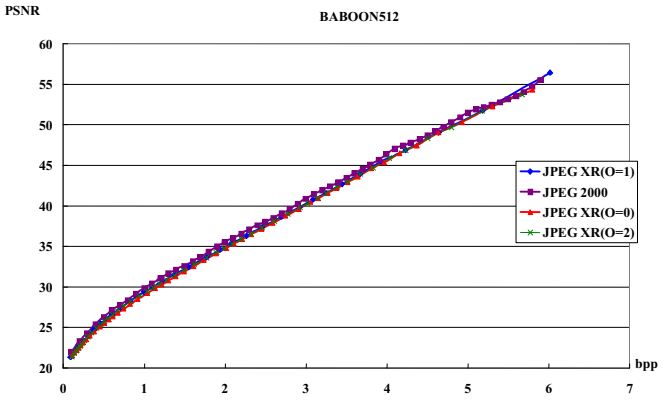

Fig. 1. PSNR Comparison between different image standard.

(a) (b)

Fig. 2. Image Quality Comparison with (a) JPEG XR and (b) JPEG.

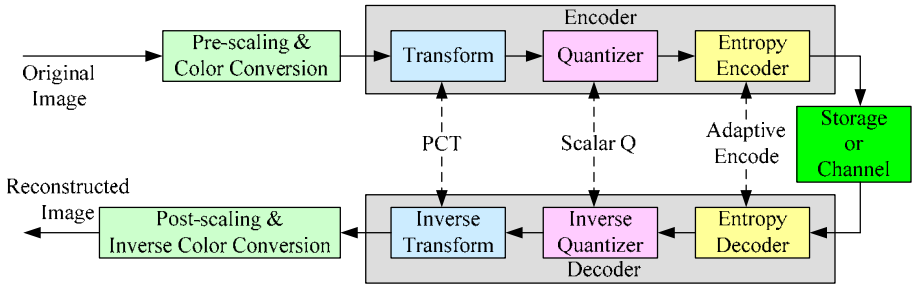

Fig. 3. JPEG XR Coding Flow.

The coding flow of JPEG XR is shown in the Fig. 3. The JPEG XR has lower computation cost in each module and simple coding flow while maintaining similar PSNR quality at the same bitrate as compared with the coding flow of JPEG2000. Hence, the JPEG XR is very suitable to be implemented with the dedicated hardware to deal with HD photo size image for the HDR display requirement. The JPEG XR encoder architecture design is presented in the following section.
This paper is organized as follows. In Section 2, we present the fundamentals of JPEG XR. Section 3 describes the characteristics of our proposed architecture design of JPEG XR encoder. Section 4 shows the implementation results. Finally, a conclusion is given in Section 5.

2. JPEG-XR

The JPEG XR image compression standard has many options for different purposes. In the following section, the fundamentals of JPEG XR are introduced.

2.1 Image Partition and Windowing

Figure 4 shows the partition and windowing example of JPEG XR standard. As the different tile size makes the different compression result, the compression results under the same quantization with different tile size for four test 512x512 benchmark images are discussed in (Pan et al., 2008). When the tile function is used, the compressed image size is increased with the tiles number when the small tiles size has been chosen.
When the image has been divided into the different tiles, each tiles are processed independently which are more suitable for the design of hardware implementation and memory buffer size. The pixels acrossing the boundary of the tiles have to be processed without any data dependency. The data dependancy must be changed accordingly, and the scan order has to be rebuilt as well. This characteristic overcome the error impact of data independency when the error occurs. Although the division of tiles is helpful for the hardware implementation and data reserving in error environment, but it decreases the data compression efficiency. However, it depends on the tradeoff between hardware design considerations and the data compression efficiency.

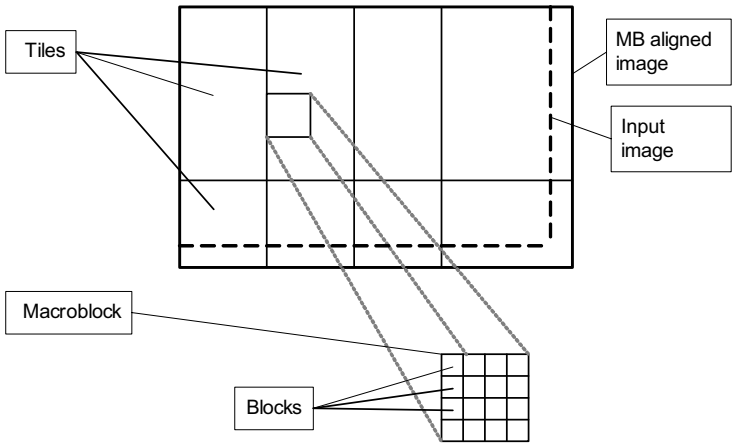

Fig. 4. Example of Image Partitions and Windowing.

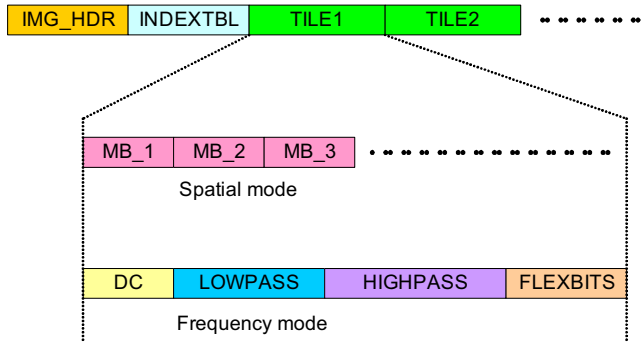

Fig. 5. Layout of Bitstream.

2.2 Operation Modes

In the Fig. 5, there are two modes of bitstream in JPEG XR which are named as spatial mode and frequency mode. In the spatial mode, a single tile packet carries the bitstream in macroblock raster order scanning from left to right and top to bottom. In the frequency mode, the bitstream of each tile is carried in multiple tile packets, where each tile packet carries transform coefficients of each frequency band in the tile. The freqency mode transmits the DC coefficients at first. When the user receive the bitstreams, images are decoded progressively. The frequency mode is very suitable for limited bandwidth applications such as web browsing.

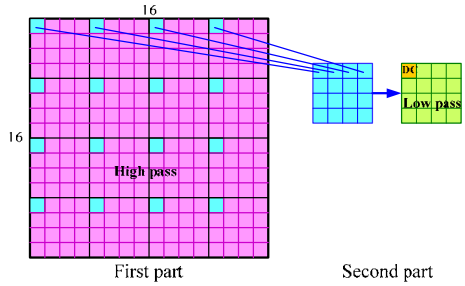

Fig. 6. The 1st part and 2nd part process of PCT.

2.3 Color Conversion

The purpose of color conversion is to reflect the color sensivity according to the characteristics of human eyes. The color space is converted from RGB to YUV. Then, the important energies are concentrate on Y channel. The color conversion is reversible, in other words, this color conversion is a lossless conversion. The color conversion equation is

$$\begin{aligned} V &= B - R \\ U &= -\left[R - G + \left\lfloor \frac{V}{2} \right\rfloor \right] \\ Y &= G + \left\lfloor \frac{U}{2} \right\rfloor - offset \end{aligned} \tag{1}$$

where offset is 128.

2.4 Pre-filter

There are three overlapping choices of the pre-filter function: non-overlapping, one-level overlapping and two-level overlapping. (Pan et al., 2008) discuss different trade-offs of three overlapping choices. The non-overlapping condition is used for fastest encoding and decoding. It is efficient in low compression ratio mode or lossless mode. However, this mode potentially introduces blocking effect in low bitrate images. The one-level overlapping function, compared to the non-overlapping, has higher compression ratio but it needs additional time for the overlapping operation. The two-level overlapping has the highest computation complexity. The PSNR and objective image quality of two-level overlapping are better than the other two at low bitrate area. The pre-filter function is recommended for both high image quality and further compression ratio considerations at high quantization levels. For high image quality issue, the pre-filter function eliminate the block effect which is sensitive of visual quality.

2.5 Photo Core Transform

Photo Core Transform (PCT) function transforms the pixel values after pre-filter computing to the lowest frequency coefficient DC, low frequency coefficients AD, and high frequency coefficients AC. There are two parts process for the PCT as shown in Fig. 6. The macroblock (MB) is partitioned into 16 4x4 blocks. In each part process, each 4x4 block is pre-filtered and then transformed by 4x4 PCT. A 2x2 transform is applied to four coefficients by four times

for each 4x4 block in first part process. The low frequency coefficient of these four coefficients is processed to the top-left coefficient. After first part process, the DC coefficients of 16 4x4 blocks can be collected as a 4x4 DC block. The second part process is for the 4x4 DC block from the first part process. The second part 2D 4x4 PCT is built by using the three operators: 2x2 T_h, T_odd and T_odd_odd. After second part process, the 16 DC coefficients are processed as DC and AD coefficients.

2.6 Quantization

The quantization is the process of rescaling the coefficients after applying the transform process. The quantization uses the quantized value to divide and round the coefficients after PCT transformed into an integer value. For the lossless coding mode, the quantized value = 1. For the lossy coding mode, the quantized value > 1. The quantization of JPEG XR use integer operations. The advantage of integer operation keeps the precision after scaling operations and only uses shift operation to perform the division operations. The quantization parameter is allowed to differ across high pass band, low pass band, and DC band. It varies to different values according to the sensitivity of human vision in different coefficient bands. Fig. 7 shows a example that pixels transformed by 2 stage PCT and quantized by the quantization process.

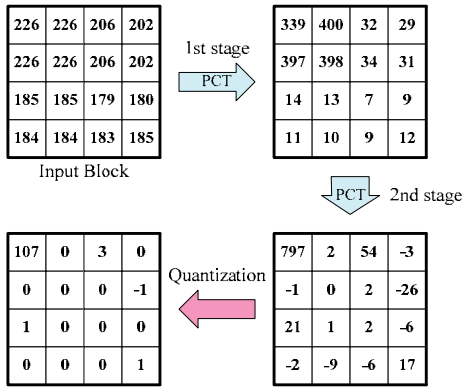

Fig. 7. Example of PCT.

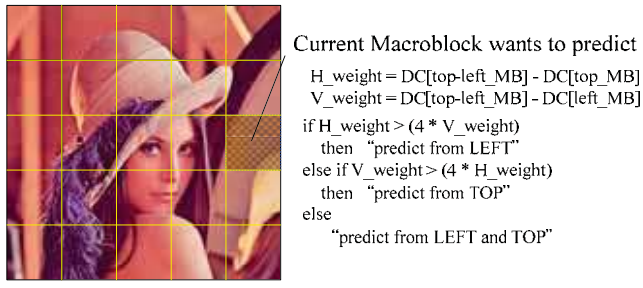

Fig. 8. DC prediction model.

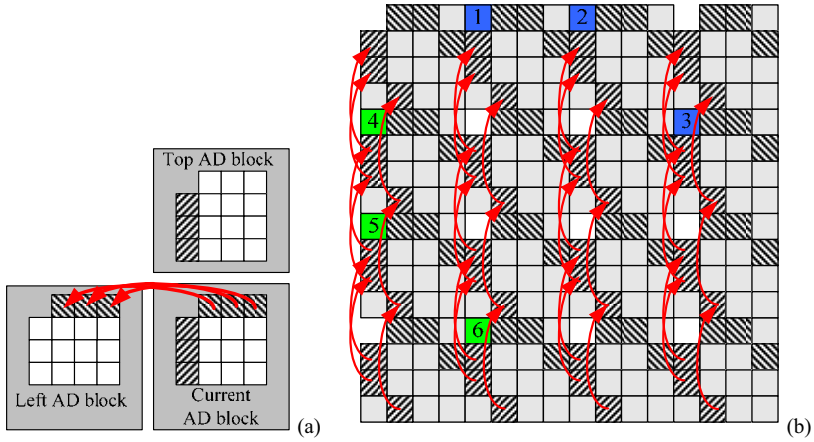

Fig. 9. Example of (a) Prediction of AD coefficients. (b) Prediction model and AC coefficients.

2.7 Prediction

There are three directions of DC prediction: LEFT, TOP, and LEFT and TOP. As shown in Fig. 8, the JPEG XR prediction rules of DC coefficient use the DC coefficient of left MB and top MB to decide the direction of DC prediction. The DC predicted direction can be decided by the pseudo code described in Fig. 8. After comparing the H_weight and the V_weight, the predicted direction can be decided. At the boundary MBs of image, the purple MBs only predict from left MB and the gray MBs only predict from top MB.

The AD/AC block can be predicted from its TOP or LEFT block as Fig. 9. If the predicted direction is LEFT, the prediction relationship example of AD coefficients is shown as Fig. 9(a). The AD predicted direction follows DC prediction model as described in Fig. 8. The computation after prediction judgment of AC is a little similar to AD that can reduce the coefficient value of the block. The Fig. 9(b) shows the prediction relationship example of AC coefficients when the prediction direction is TOP.

2.8 Entropy Encoding

The adaptive scan is used for entropy coding which is based on the latest probability of non-zero coefficients. Fig. 10 shows an example that the previous scan order is changed to update scan order based on the probability. The most probable non-zero coefficients are scanned firstly and the probability is counted by numbers of the non-zero coefficients. If the non-zero probability is larger than the previous scanned coefficient, the scan order is exchanged. By doing so, the non-zero coefficients are collected together and processed in an orderly fashion.

After the coefficients of current block are scanned by adaptive scan order, JPEG XR uses two entropy coding schemes. Fig. 11 demonstrates an example of coding a 4x4 block when ModelBits is 4 bits. The coefficients of 4x4 block are scanned by adaptive scan order first, then the ModelBits is updated by the total overhead coefficients in each band per channel. The coefficients which can be represented under the ModelBits are encoded by the FlexBits table. For the extra bit such as the 17 can not be represented in four bits, the encoding block

will increase extra bit for the new Run-Level Encode (RLE) coding. The RLE function is added into the bitstream length for the overhead coefficient. The Levels, the Runs before nonzero coefficients, and the number of overhead coefficients are counted for RLE. Then the RLE block is encoded firstly while different size of Huffman table is used to make the bit allocation in optimization. After processing the RLE algorithm, the RLE results and FlexBits are packetized.

3. Architecture Design of JPEG XR

In the following section, a JPEG XR encoder architecture design is presented. For the system requirement, the consideration of functional block pipelining, and the architecture design of the pre-filter, transform, quantization, prediction, and the entropy coding modules are all discussed in the following sections.

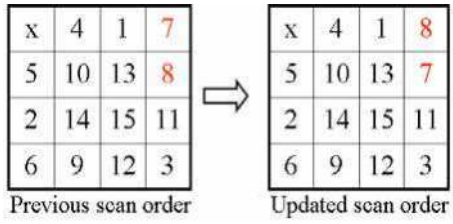

Fig. 10. Adaptive scan order updating example.

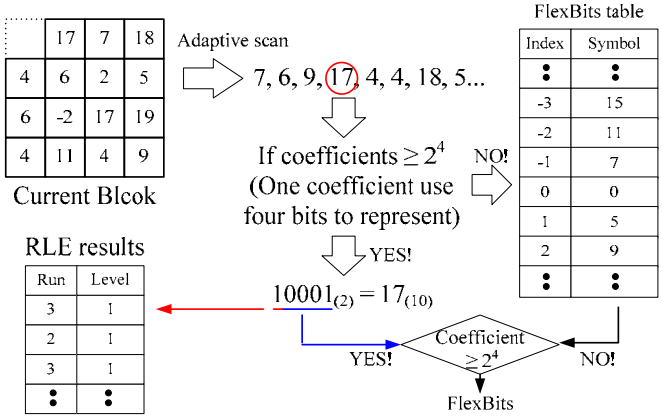

Fig. 11. Adaptive scan and Run-Level encode example.

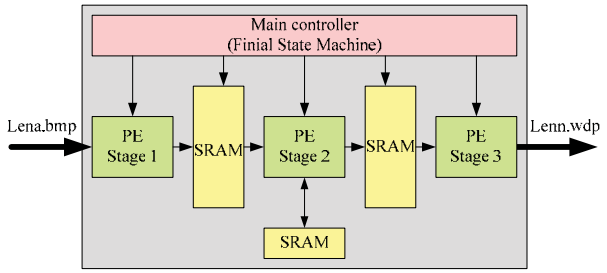

Fig. 12. Pipeline stage consideration of JPEG XR.

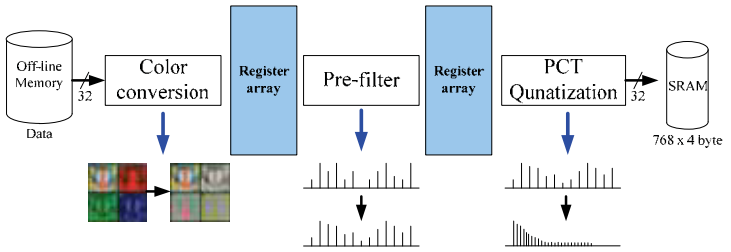

Fig. 13. Data flow of JPEG XR stage 1.

3.1 System Architecture and Functional Block Pipelining

The pipeline stages of this JPEG XR encoder design can be divided into three steps, as shown in Fig. 12. Since the color conversion, pre-filter and the PCT modules are computed with the 4x4 block matrix style with no feedback information, they are arranged into the same stage at the beginning. The prediction unit is used as second stage for different direction comparisons with the feedback processed pixels for the DC/AD/AC region. Then, the entropy encoding module with high data dependency are divided as the third stage.

3.2 Color Conversion, Pre-filter, PCT, and Quantization

Fig. 13 shows the data flow of stage 1. A 32 bits external bus can transmit 2 R/G/B coefficients in one clock cycle. The pre-filter and PCT/quantization modules process the coefficients after color conversion. This paper uses a memory reused method to reduce the memory access bandwidth for the color conversion, pre-filter, and PCT module. The black line in Fig. 14 is the boundary of MB. When the blue range is to be processed after the yellow range, the pre-filter function only have to deal with the amount of new data instead of the entire block data into the register. This is due to the presence of previously stored column coefficients of blocks in the registers and the capability to be utilized as the coefficients in the yellow range. Therefore, the memory buffer size and the memory access for pre-filter are reduced. Otherwise, the additional SRAM will be needed to store the data of the entire block from off-chip memory and the execution time will be increased. The detail simulations is discussed in (Pan et al., 2008).

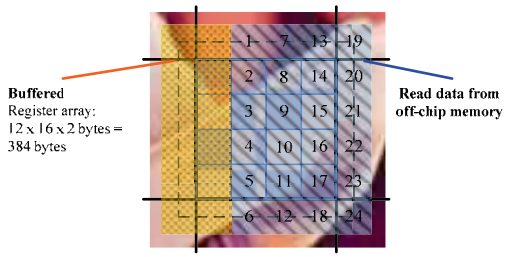

Fig. 14. Memory reused method.

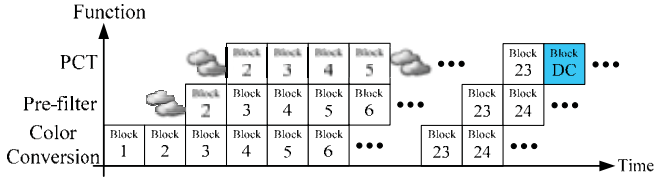

Fig. 15. DC block insertion on the block pipeline.

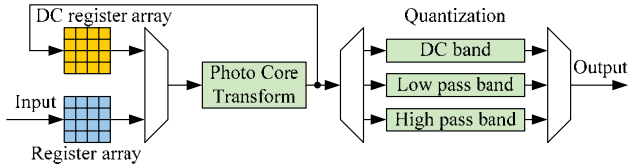

Fig. 16. Architecture of PCT, and Quantization.

Because the functions of stage 1 are computing with the 4x4 block style with no feedback information, the three pipeline architecture for stage1 is used: color conversion (CC), pre-filter, PCT (include quantization). For the memory allocation, the color conversion requires 6 blocks per column, the pre-filter requires 5 blocks, and the PCT including quantization uses 4 blocks to execute the function. At the end of PCT in Fig. 15, DC block can be processed to reduce one pipeline bubble when the next new pipeline stage starts. Hence, the well arranged timing schedule eliminated the access of additional clock cycle to process the DC block.

The architecture for PCT including Quantization is shown in Fig. 16. Additional registers are implemented to buffer the related coefficients for the next pipeline processing. The left multiplex selects the inputs for two parts PCT process. Initially, the input pre-filtered coefficients are selected to process the PCT algorithm. Then, the yellow block (DC) will be processed after the 16 blocks have been computed. The quantization stage de-multiplex the DC, low pass band AD and high pass band AC coefficients to suitable process element. The processed data are arranged into the quantized coefficients of Y, U, V SRAM blocks for prediction operation.

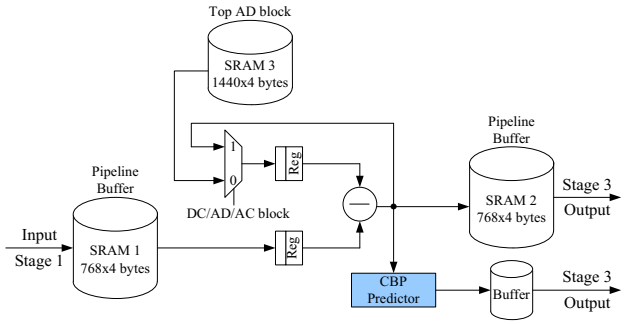

Fig. 17. Prediction architecture.

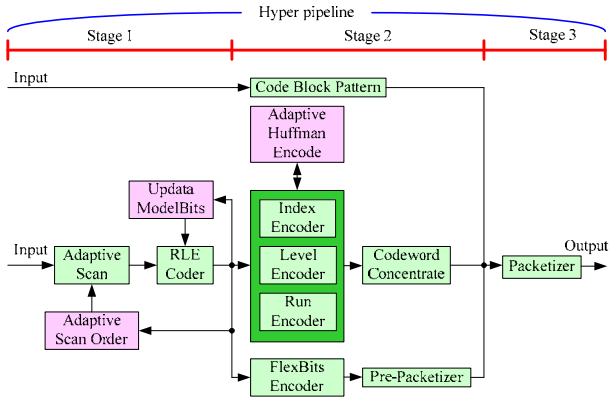

Fig. 18. Adaptive encode architecture.

3.3 Prediction

The quanitzed data stored in Y, U, V SRAM blocks are processed with the subtract operation as the prediction algorithm. Three SRAM blocks are used in the this design. One 1440x4 byte SRAM is used to buffer the 1 DC and 3 AD coefficients for the TOP AD prediction coefficients in the prediction judgement, so that the regeneration of these data are unnecessary when they are selected in the prediction mode. In Fig. 17, two 768x4 bytes SRAMs are used to save the quantized coefficients of current block and the predicted coefficients for current block.

3.4 Entropy Encoding

There are three complex data dependency loops in the entropy encoding module as shown in Fig. 18. The first one is the adaptive-scan-order block used to refresh the scan order. The second is the updated ModelBits block, which decides how many bits are necessary to represent a coefficient. Third, the adaptive Huffman encode block to choose the most efficient Huffman table.

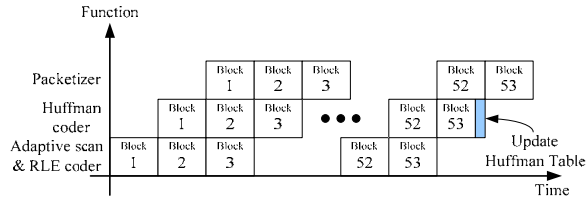

Fig. 19. Timing scheduling of entropy encoding.

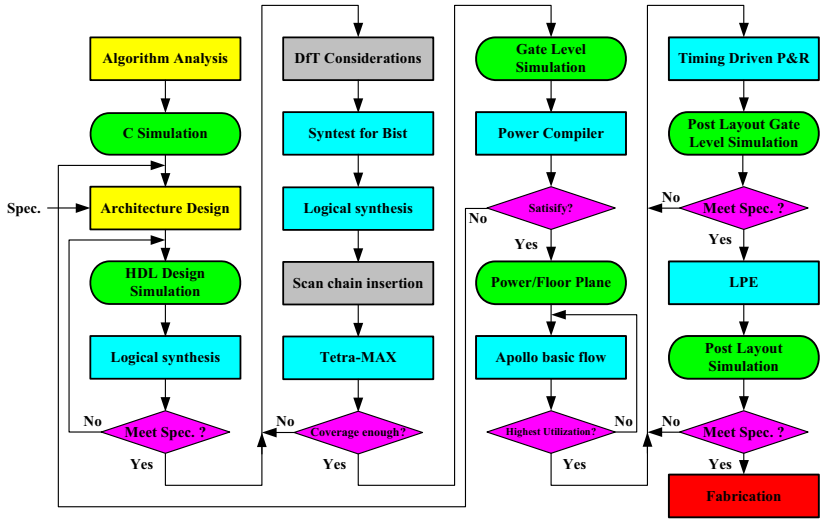

Fig. 20. Chip design flow.

In order to increase the throughput and decrease the timing of critical patch, we use three sub-pipeline stages architecture to implement the entropy encoding module as described in (Chien et al., 2009). Because of the feedback patch, stage 1 includes the modules of feedback path and the feedback information generator. Stage 2 includes the modules that encode the RLE information to symbol-based data. The bit-wise packetizer module of stage 3 processes the symbol-based data bit by bit. By doing this, we can increase the throughput about 3 times by well arranged pipeline timing schedule as shown in Fig. 19.

The design of adaptive scan block counts the numbers of the non-zero coefficients to decide whether the scan order should be exchanged or not. After the processing of the adaptive scan block, the coefficients which can be represented under ModelBits are coded by the FlexBits table. The other coefficients change to generate the Flexbits and Level. After the processing of the RLE algorithm, the Level and Run choose the suitable Huffman table to generate the RLE codeword. The RLE results are translated into the codewords by different Huffman tables. The Huffman encoder in JPEG XR is different from the other standards. It is composed of many small Huffman tables and can adaptively choose the best option in these tables for the smaller codesize for Run and Level. Many codewords are produced after the Huffman encoding. In order to increase the throughput, the codeword concentrating

architecture is proposed. Linked to the output of the above operation, the whole RLE codeword and RLE codesize will be produced to the packetizer. The packetizer architecture is modified from the (Agostini et al., 2002) architecture by combining the RLE codeword and the FlexBits for generating the JPEG XR compressed file. More detail design is described in (Pan et al., 2008).

4. Implementations

This design is implemented to verify the proposed VLSI architecture for JPEG XR encoder. And it is also verified by FPGA platform. The detail information about implementation result of each module by the FPGA prototype system is shown in (Pan et al., 2008). It is used to test the conformance bitstreams for the certification.

A three-stage MB pipelining of 4:4:4 lossless JPEG XR encoder was proposed to process the capacity and hardware utilization. In our design, the extra registers are used to increase the pipeline stages for achieving the specification, such as the color conversion, PCT/ quantization and the adaptive encode block. And the on-chip SRAM blocks are used to store the reused data processed with the prediction module to eliminate the memory access. For the entropy encoding module, the timing schedule and pipelining is well designed. The proposed architecture of entropy encoding module increases the total throughput about three times. We use 0.18um TSMC CMOS 1P6M process to implement the JPEG XR encoder. Our design flow is standard cell based chip design flow. The design flow of our design is shown as Fig 20. Test consideration is also an important issue in chip design. Therefore, the scan chain and the built-in self-test (BIST) are considered in our chip. The chip synthesis layout is shown as Fig. 21. The implementation results are shown as the Table 2. The power dissipation distribution in shown as Fig. 22.

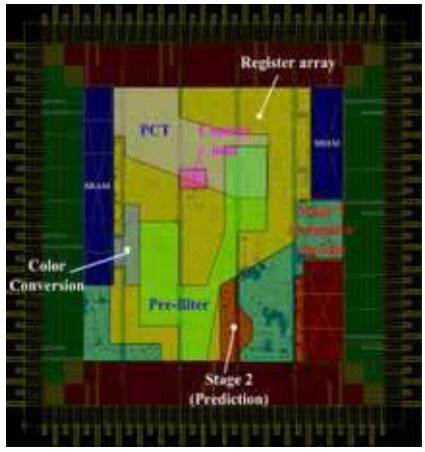

Fig. 21. Chip synthesis layout.

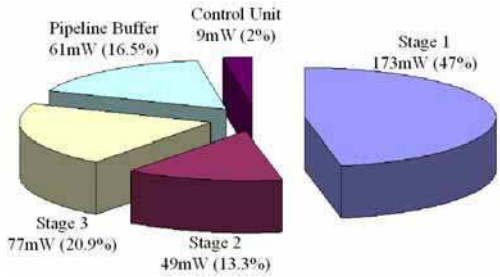

Fig. 22. Chip power dissipation distribution.

Technology	TSMC 0.18um CMOS 1P6M
Core Size	3.18 mm x 3.18 mm (9.3025 mm^2)
Die Size	4.64 mm x 4.64 mm (20.47 mm^2)
Gate Count	651.7 K
Work Clock Rate	62.5 MHz
Power Consumption	368.7 mW@62.5MHz，1.8V
On-Chip Memory	256x32 SRAM x 6 (single port) 768x32 SRAM x 2 (single port) 480x32 SRAM x 3 (single port) Total: 144,384 Bits = 18,047 Bytes
Processing Capability	34.1 Mega pixels within one second 5.45 fps for 4:4:4 HDTV(1920x1080) @62.5MHz 32.9 fps for 4:4:4 VGA(640x480) @62.5MHz 112 fps for 4:4:4 CIF(352x288) @62.5MHz
Input Pad	37
Output Pad	34

Table 2. Chip specification.

5. Conclusion

Compared with the JPEG2000, the coding flow of the JPEG XR is simple and has lower complexity in the similar PSNR quality at the same bit rate. Hence, the JPEG XR is very suitable for implementation with the dedicated hardware used to manage HD photo size images for the HDR display requirement. In this paper, we initially analyzed the comparison of JPEG XR with other image standard, and then a three-stage MB pipelining was proposed to process the capacity and hardware utilization. We also made a lot of efforts on module designs. The timing schedule and pipelining of color conversion, pre-filter, PCT & quantization modules are well designed. In order to prevent accessing the coefficients from off-chip memory, an on-chip SRAM is designed to buffer the coefficients for the prediction module with only some area overhead. The pre-filter and PCT function was designed to reduce 33.3% memory access from off-chip memory.For the entropy coding, we designed a codeword concentrating architecture for the throughput increasing of RLE algorithm. And the adaptive encode and packetizer modules efficiently provide the coding information required for packing the bitstream. Based on this research result, we contribute

a VLSI architecture for 1920x1080 HD photo size JPEG XR encoder design. Our proposed design can be used in those devices which need powerful and advanced still image compression chip, such as the next generation HDR display, the digital still camera, the digital frame, the digital surveillance, the mobile phone, the camera and other digital photography applications.

6. References

B. Crow, Windows Media Photo: A new format for end-to-end digitalimaging, *Windows Hardware Engineering Conference*, 2006.

C.-H. Pan; C.-Y. Chien; W.-M. Chao; S.-C. Huang & L.-G. Chen, Architecture design of full HD JPEG XR encoder for digital photography applications, *IEEE Trans. Consu. Elec.*, Vol. 54, Issue 3, pp. 963-971, Aug. 2008.

C.-Y. Chien; S.-C. Huang; C.-H. Pan; C.-M. Fang & L.-G. Chen, Pipelined Arithmetic Encoder Design for Lossless JPEG XR Encoder, *IEEE Intl. Sympo. on Consu. Elec.*, Kyoto, Japan, May 2009.

D. D. Giusto & T. Onali. Data Compression for Digital Photography: Performance comparison between proprietary solutions and standards, *IEEE Conf. Consu. Elec.*, pp. 1-2, 2007.

D. Schonberg; S. Sun; G. J. Sullivan; S. Regunathan; Z. Zhou & S. Srinivasan, Techniques for enhancing JPEG XR / HD Photo rate-distortion performance for particular fidelity metrics, *Applications of Digital Image Processing XXXI, Proceedings of SPIE*, vol. 7073, Aug. 2008.

ISO/IEC JTC1/SC29/WG1. JPEG 2000 Part I Final Committee Draft, Rev. 1.0, Mar. 2000.

ITU. T.81 : Information technology - Digital compression and coding of continuous-tone still images. 1992.

L.V. Agostini; I.S. Silva & S. Bampi, Pipelined Entropy Coders for JPEG compression, *Integrated Circuits and System Design*, 2002.

S. Groder, Modeling and Synthesis of the HD Photo Compression Algorithm, Master Thesis, 2008.

S. Srinivasan; C. Tu; S. L. Regunathan & G. J. Sullivan, HD Photo: a new image coding technology for digital photography, *Applications of Digital Image Processing XXX, Proceedings of SPIE*, vol. 6696, Aug. 2007.

S. Srinivasan; Z. Zhou; G. J. Sullivan; R. Rossi; S. Regunathan; C. Tu & A. Roy, Coding of high dynamic range images in JPEG XR / HD Photo, *Applications of Digital Image Processing XXXI, Proceedings of SPIE*, vol. 7073, Aug. 2008.

Y.-W. Huang; B.-Y. Hsieh; T.-C. Chen & L.-G. Chen, Analysis, Fast Algorithm, and VLSI Architecture Design for H.264/AVC Intra Frame Coder, IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 3, pp. 378-401, Mar. 2005.

VLSI

Edited by Zhongfeng Wang

ISBN 978-953-307-049-0
Hard cover, 456 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

The process of Integrated Circuits (IC) started its era of VLSI (Very Large Scale Integration) in 1970's when thousands of transistors were integrated into one single chip. Nowadays we are able to integrate more than a billion transistors on a single chip. However, the term "VLSI" is still being used, though there was some effort to coin a new term ULSI (Ultra-Large Scale Integration) for fine distinctions many years ago. VLSI technology has brought tremendous benefits to our everyday life since its occurrence. VLSI circuits are used everywhere, real applications include microprocessors in a personal computer or workstation, chips in a graphic card, digital camera or camcorder, chips in a cell phone or a portable computing device, and embedded processors in an automobile, et al. VLSI covers many phases of design and fabrication of integrated circuits. For a commercial chip design, it involves system definition, VLSI architecture design and optimization, RTL (register transfer language) coding, (pre- and post-synthesis) simulation and verification, synthesis, place and route, timing analyses and timing closure, and multi-step semiconductor device fabrication including wafer processing, die preparation, IC packaging and testing, et al. As the process technology scales down, hundreds or even thousands of millions of transistors are integrated into one single chip. Hence, more and more complicated systems can be integrated into a single chip, the so-called System-on-chip (SoC), which brings to VLSI engineers ever increasingly challenges to master techniques in various phases of VLSI design. For modern SoC design, practical applications are usually speed hungry. For instance, Ethernet standard has evolved from 10Mbps to 10Gbps. Now the specification for 100Mbps Ethernet is on the way. On the other hand, with the popularity of wireless and portable computing devices, low power consumption has become extremely critical. To meet these contradicting requirements, VLSI designers have to perform optimizations at all levels of design. This book is intended to cover a wide range of VLSI design topics. The book can be roughly partitioned into four parts. Part I is mainly focused on algorithmic level and architectural level VLSI design and optimization for image and video signal processing systems. Part II addresses VLSI design optimizations for cryptography and error correction coding. Part III discusses general SoC design techniques as well as other application-specific VLSI design optimizations. The last part will cover generic nano-scale circuit-level design techniques.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ching-Yen Chien, Sheng-Chieh Huang, Chia-Ho Pan and Liang-Gee Chen (2010). Full HD JPEG XR Encoder Design for Digital Photography Applications, VLSI, Zhongfeng Wang (Ed.), ISBN: 978-953-307-049-0, InTech, Available from: http://www.intechopen.com/books/vlsi/full-hd-jpeg-xr-encoder-design-for-digital-photography-applications

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821