IEICE Electronics Express, Vol.7, No.5, 332–338

Fast implementation of H.264 4x4 intra prediction

Jianjun Li[a)] **and Esam Abdel-Raheem**[b)]

Department of Electrical and Computer Engineering

University of Windsor, Windsor, Ontario N9B 3P4, Canada

a) *li1118@uwindsor.ca*

b) *eraheem@uwindsor.ca*

Abstract: This letter proposes a fast parallel architecture and redundancy reduction algorithm for H.264/AVC intra4x4 prediction to speed up intra frame coding. A significant reduction in execution time is achieved without losing video quality. Only 204 cycles are required to process a macroblock (MB). Compared with the dedicated intra prediction [1], processing speed is enhanced by 79%.

Keywords: H.264/AVC, intra prediction, parallel processing

Classification: Science and engineering for electronics

References

[1] Y.-W. Huang, B.-Y. Hsieh, T.-C. Chen, and 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, March 2005.

[2] "Joint Video Team, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC," May 2003.

[3] G. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," *SPIE Conference on Applications of Digital Image Processing*, Aug. 2004.

[4] W. Lee, S. Lee, and J. Kim, "Pipelined Intra Prediction Using Shuffled Encoding Order for H.264/AVC," *TENCON 2006*, pp. 14–17, Nov. 2006.

[5] K. Suh, S. Park, and H. Cho, "An Efficient Hardware Architecture of Intra Prediction and TQ/IQIT Module for H.264 Encoder," *ETRI Journal*, vol. 27, no. 5, pp. 511–524, Oct. 2005.

[6] G. Jin and H.-J. Lee, "A Parallel and Pipelined Execution of H.264/AVC Intra Prediction," *IEEE International Conference on Computer and Information Technology, CIT'06*, pp. 246–250, Sept. 2006.

[7] J. Li and M. Ahamdi, "Realizing High Throughput Transforms of H.264/AVC," *IEEE International Symposium on Circuits and Systems*, pp. 840–843, May 2008.

[8] K. Roman and S. Shahram, "On Hardware Implementations Of DCT and Quantization Blocks for H.264/AVC," *The Journal of VLSI Signal Processing*, vol. 47, no. 2, pp. 93–102, May 2007.

[9] [Online] http://iphome.hhi.de/suehring/tml/

DOI: 10.1587/elex.7.332
Received October 24, 2009
Accepted February 12, 2010
Published March 10, 2010

1 Introduction

The Joint Video Team (JVT) of ISO/IEC MPEG and ITU-VCEG jointly developed a new video compression standard H.264/AVC [2]. Compared to previous standards, such as MPEG-2, H.263 and MPEG-4, aggressive compression techniques are employed in H.264/AVC standard. As a result, its performance is greatly improved in terms of the compression efficiency, however, this is achieved at the expense of increasing the computational complexity.

Intra prediction utilizes the spatial correlation in an image to predict the block being encoded from its nearby pixels. It is recognized to be one of the main factors contributing the success of H.264/AVC [2]. To select the best prediction mode, the encoder has to search all possible prediction modes exhaustively in order to encode blocks. As a result, the computational complexity in H.264/AVC is extremely high. Some previous approaches to reduce computation complexity of H.264/AVC intra prediction by optimizing and speeding-up are presented in [1, 4, 5, 6].

This letter provides a novel parallel architecture to achieve fast intra prediction. The rest of the paper is organized as follows. In section 2, the H.264/AVC intra prediction is briefly introduced. Section 3 presents the proposed architecture. Section 4 provides simulation results and conclusions are addressed in Section 5.

2 H.264/AVC Intra Prediction

In H.264/AVC baseline intra coding, two intra prediction modes for luminance component are supported in each profile [2]. One is intra4x4 mode and the other is intra16x16 mode. The intra8x8 is a new prediction type defined in H.264/AVC FREext [3]. For intra4x4 mode, the macrocblock (MB) is divided into sixteen non-overlapping 4x4 blocks. Each block can select one of nine prediction modes. For intra16x16 mode, four prediction modes are available for each macroblock. Chroma intra prediction is independent to luminance prediction mode. Two chroma components are simultaneously predicted with the same mode. The intra4x4 is more accurate than intra16x16, however it requires more bits to be coded. Hence, intra4x4 is used for highly textured regions while intra16x16 is used for plain regions.

From the complexity perspective, H.264/AVC encodes MBs by iterating all the luminance intra decisions for each possible chroma intra prediction mode to achieve the best coding efficiency. Therefore, the number of mode combinations for luminance and chroma components in an MB is $C8 \times (L4 \times 16 + L16)$, where C8, L4, and L16 represent the number of modes for chroma prediction, 4x4 luminance prediction, and 16x16 luminance prediction respectively. This means that, $4 \times (9 \times 16 + 4) = 592$ different costs have to be performed before a best mode can be determined. If the target application is high-definition TV (HDTV), each frame needs $(1920 \times 1080 \times 592)/256 = 4{,}795{,}200$ calculations, which is not feasible for real-time implementation. Thus, speeding up intra coding process is essen-

DOI: 10.1587/elex.7.332
Received October 24, 2009
Accepted February 12, 2010
Published March 10, 2010

tially required.

The 4x4 intra prediction mode has one DC mode (mode2) and eight directional modes, e.g., horizontal (mode0), vertical (mode1), diagonal down left (mode3), diagonal down right (mode4), vertical right (mode5), horizontal down (mode6), vertical left (mode7) and horizontal up (mode8). To reflect the edge trend of the block, the prediction for the current 4x4 block is calculated using the boundary pixels of the previously decoded blocks above and to the left of it. Since the pixels along the direction of the local edge have similar values, an accurate prediction can be achieved if the direction of the prediction mode is the same as the edge direction of the block. Therefore, in the intra4x4 prediction, the block in the upper left corner is processed at first and the down right corner is processed at last. The intra prediction for each block uses the pixels in its left and top sides as reference pixels. A block thus can not be predicted until its previous block has been reconstructed. The reconstruction includes DCT, Quantization, Inverse Quantization and inverse DCT (DCT, Q, IQ and IDCT).

3 Proposed Parallel Processing

The original process handles blocks in serial, which is not efficient as illustrated in Fig. 1 (a). Efficient architectures have been reported in [1, 4, 5, 6], however, they all have drawbacks either with the pipelining architecture or in compression gains. Huang's work [1] has bubbles between Intra4x4 predictions because of the low throughput of reconstruction process so that the prediction has to wait for the completion of reconstruction. Lee's work [4] perfectly pipelines the intra prediction and reconstruction process shown in Fig. 1 (b), however, it requires that both intra prediction and reconstruction have exact equal processing cycles. It also reduces some prediction modes in some blocks in order to enforce pipelining, hence, the video quality is de-

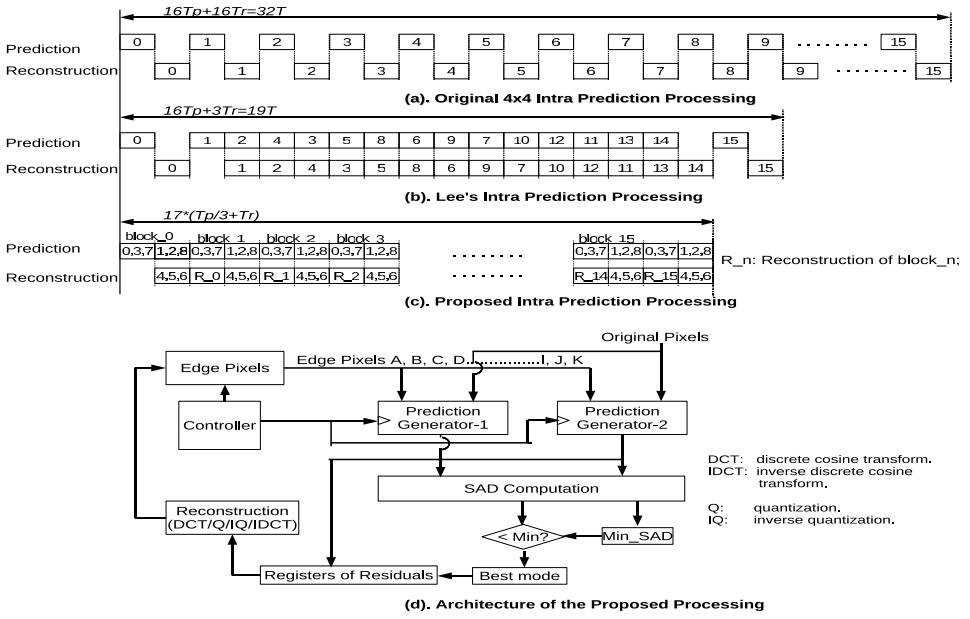

Fig. 1. Intra4x4 Prediction Process and Architecture.

DOI: 10.1587/elex.7.332
Received October 24, 2009
Accepted February 12, 2010
Published March 10, 2010

graded. Suh's work [5] is similar to Huang's work, which takes 34 cycles to process each block. Jin's work [6] proposes both partially and fully pipelined architectures for intra4x4 prediction and has the same drawback as the approach in [4]. Moreover, the architectures add dependency graph process in order to improve gains, however, this increases hardware overhead. It takes 25 cycles to process each block, which is too long for high throughput reconstruction. This letter proposes an efficient parallel architecture followed by a redundancy reduction algorithm to speed up the intra4x4 prediction.

3.1 Parallel Architecture

Although a data dependency truly exists among blocks in intra4x4 prediciton, we can state, after careful observation, that such data dependency does not exist in some intra4x4 prediction modes. Therefore, they are able to be processed without waiting for their previous blocks to be reconstructed, i.e., mode0, 3, and 7 of the current block can be simultaneously predicted when its previous block is being reconstructed. After the reconstruction of its previous block is complete, the rest of modes, i.e., mode1, 2, 8 and mode4, 5, 6 of the current block, shown as Fig. 1 (c) can be predicted in parallel. The same procedure follows in the rest of the blocks. To sum up, we divide nine prediction modes into three groups, and each group has three prediction modes.

The proposed architecture has four advantages compared to previous works. The first advantage is that the proposed process does not ignore any prediction modes. The second advantage is that the encoder follows that order specified in H.264/AVC standard to guarantee the consistency between the encoder and the decoder. The third advantage is that it does not require the processing cycles of intra prediction and reconstruction to be exactly the same. The last advantage is that the proposed architecture can reduce total processing time of each MB to $17\times(1/3T_p+T_r)$ if high throughput reconstruction is adopted, where T_p is prediction time and T_r is reconstruction time. As shown in Fig. 1 (d), the luma4x4 prediction unit mainly consists of five functional blocks for Prediction Generator-1, Prediction Generator-2, SAD (sum of absolute difference) Computation, Reconstruction, and Controller. Prediction Generator-1 and Prediction Generator-2 calculate the predicted pixel values for all the intra modes. SAD Computation block calculates sum of absolute difference values for each mode in order to make mode decision. Reconstruction block recovers the prediction pixels of the best mode by the reconstruction process (DCT, Q, IQ, and IDCT). The Controller block selects the right pixels from Edge Pixels buffer and feeds them into Prediction Generator blocks.

In the first step, the Generator-1 is parallel with the reconstruction process and the first group (mode0, 3 and 7) are predicted in this step. In the second step, the Generator-1 and Generator-2 are parallel to process the second group (mode1, 2, and 8) and the third group (mode4, 5 and 6). The output of Edge Pixels unit are selected by the controller. Each group has its own best mode by calculating SAD. The final best mode is obtained by

DOI: 10.1587/elex.7.332
Received October 24, 2009
Accepted February 12, 2010
Published March 10, 2010

comparing the best mode of each group. Meanwhile, the residuals of this block with the best mode are written into the register. The reconstruction process implements DCT, Q, IQ and IDCT based on the residuals. After added to the values of prediction, the reconstructed edge pixels are stored in the buffer for being used for predicting the next block.

3.2 Redundancy Reduction Algorithm

Considering the formulas of the nine intra4x4 prediction modes [2], there are some identical parts in calculating the predicted values. It is possible to reduce memory access and improve prediction time by eliminating these redundancy computations. Fig. 2 illustrates how to calculate 6 prediction modes (except DC, vertical and horizontal prediction modes) with the common parts.

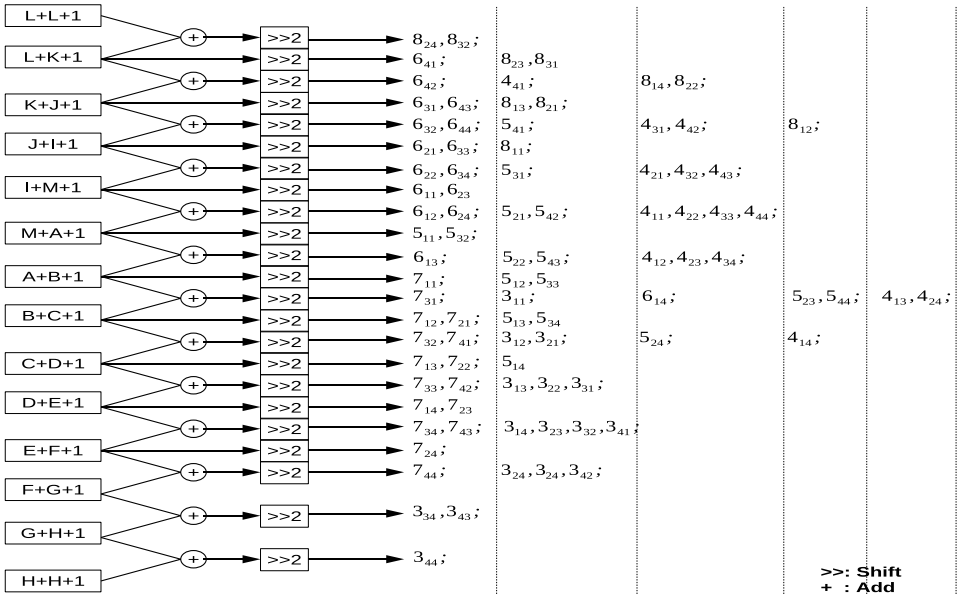

Fig. 2. Redundancy Reduction Algorithm.

The 14 common parts are listed in the left side of Fig. 2. The numbers N_{xy} in Fig. 2 refer to mode N in position (x, y). For example, the predicted value of pixel (1, 1) in diagonal down left mode, (A+2B+C+2)>>2, can be calculated by (A+B) and (B+C). Moreover, identical prediction equations not only exist in different pixels of the same mode, but also in different pixels of different modes, for example, the predicted value of pixels (1, 0) and (0, 1) in diagonal down left prediction mode is equal to that of pixel (3, 0) in diagonal down right prediction mode. By analysis, only 14 common parts and 23 equations of their combination are required for intra4x4 prediction calculations, which can be implemented by 27 adders and 23 shifts. Moreover, the DC prediction mode is very straightforward, which only requires 3 adders and 1 shift. For vertical, horizontal and part of horizontal up prediction modes, the predicted values can be obtained only by propagating the values of edge pixels. Therefore, for total intra4x4 mode prediction, the proposed

DOI: 10.1587/elex.7.332
Received October 24, 2009
Accepted February 12, 2010
Published March 10, 2010

algorithm requires 30 adders and 24 shifts. It can be completed within one cycle.

To achieve fast intra4x4 prediction, a high throughput reconstruction process is also required. The reconstruction process (DCT, Q, IQ, and IDCT) is implemented in parallel with three intra4x4 prediction modes (mode0, 3, and 7). Our previous work [7] of a high throughput realization of DCT and IDCT is adopted in this implementation. It takes one cycle to process DCT and IDCT separately. Fast quantization and its inverse have been implemented using the approach in [8], in which a lookup table is utilized to obtain quantization in one cycle. In the proposed design, both fast DCT/IDCT and Quantization/IQ are utilized in order to achieve fast implementation. The total reconstruction time to process an MB is 6 cycles (2 cycles for control).

4 Experimental Results and Analysis

We use JM10.2 [9] reference software to evaluate the compression efficiency and execution speed of our proposed parallel algorithm. The experiments conditions are: baseline profile, CIF frame size, 30frame/second, I frame only, rate distortion optimization (RDO) off, frame number equal to 100. Several sequences have been verified including Foreman, Akiyo, News and Calendar.

Table I. Execution cycles for each MB.

Methods	Cycles/Block	Cycles/MB	Full Modes	Savings
Huang. [1]	60	960	Yes	79%
Suh. [5]	34	544	Yes	63%
Jin. [6]	25	475	No	57%
Proposed	12	204	Yes	- -

Table I shows the results compared with the previous works. The proposed architecture reduces complexity up to 79% compared to [1], 63% compared to [5] and 57% compared to [6]. By comparing average peak signal-to-noise ratio (PSNR) of the proposed approach with H.264/AVC reference module at various bit rates of sequences, we find no significant quality degradation. The proposed parallel architecture is implemented in a Xilinx Virtex-4 FPGA using Xilinx ISE Series 9.1i. The implementation is verified with RTL simulations using Mentor Graphics ModelSim SE 6.1.

5 Conclusions

A fast parallel architecture and redundancy reduction algorithm of H.264/AVC intra4x4 prediction have been proposed in this letter. This parallel execution cuts down part of data dependency. It has adopted high-throughput DCT/IDCT and Q/IQ to approach a fast implementation, and has resulted in reducing the intra prediction execution time up to 79% compared with the previous works. Meanwhile, no any prediction modes are ignored. Software

DOI: 10.1587/elex.7.332
Received October 24, 2009
Accepted February 12, 2010
Published March 10, 2010

simulation shows no significant performance degradation. In order to verify the proposed design, it has been implemented in Xilinx Virtex-4 FPGA.

Acknowledgments

This work is supported by University of Windsor, Ontario, Canada (2009).

DOI: 10.1587/elex.7.332
Received October 24, 2009
Accepted February 12, 2010
Published March 10, 2010