A 61MHz 72K GATES 1280x720 30FPS H.264 INTRA ENCODER

De-Wei Li, Chun-Wei Ku, Chao-Chung Cheng, Yu-Kun Lin, and Tian-Sheuan Chang

Dept. Electronics Engineering National Chiao-Tung University, Hsinchu, Taiwan, R.O.C. {sunbreak, sjerry, fury, yklin, tschang}@twins.ee.nctu.edu.tw

ABSTRACT

Abstract—This paper presents an HD720p 30 frames per sec H.264 intra encoder operated at 61MHz with just 72K gate count. We achieve the low cost and low operating frequency with the highly utilized variable pixel scheduling, and a modified three-step fast algorithm. Thus, the resulted design only needs half of operating frequency and reduces 30% of area cost compared to the previous HD720p intra encoder design.

Index Terms—Intra prediction, H.264,

1. INTRODUCTION

The latest video coding standard for higher video quality and lower bit-rate, known as H.264 or MPEG-4 Part 10 Advanced Video Coding (MPEG-4 AVC) [1], is developed by the Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG as the next generation video compression standard. It introduces various new tools to improve the coding efficiency by up to 50% [2]. In which, the intra frame prediction technique is one of such tools that use neighboring pixels to predict current coding block. With this, the coding efficiency is much higher than that in the previous standards and even competitive with the latest still image coding standard, JPEG2000 [3]. This intra frame only coding and decoding is very suitable for applications like digital video recorder and digital still camera that do not need or cannot afford the inter prediction capability because of lower cost and power issue. Besides, the intra-only profile may be adopted in H.264 spec, the intra frame only encoder and decoder can be the new topic in the feature.

In this paper, we present an H.264 intra frame codec IP design improved from our previous work [4]. Though the design in [4] is quite suitable for video application products such as digital camera or digital video recorder, it still lacks the consideration for low power consumption. We improve our design by with the variable-pixel parallel architecture and modified three-step fast prediction algorithm to reduce nearly half clock rate. The variable-pixel parallel datapath can save almost half of the prediction cycles efficiently, and the fast algorithm can speedup the prediction flow of 4x4 modes with only negligible quality degradation. This work supports HD size 720p encoding process with only half of

frequency requirement with [4] and lower gate count, and thereby reduces the power consumption.

The rest of the paper is organized as follows. In Section 2, we first overview H.264 intra coding. Then, we present the modified three-step intra prediction algorithm and variable pixel parallel architecture for fast intra coding in Section 3. Section 4 shows the implementation result and design comparison. Finally, a conclusion is made in Section 5.

2. OVERVIEW OF H.264 INTRA FRAME ENCODING

Fig. 1 shows the intra encoding flow of H.264. Each input macroblock data will be intra-predicted by the reconstructed boundary data of neighboring blocks. These predictions can be one of nine kinds of 4x4 luma prediction modes as in Fig. 2, four kinds of 16x16 luma prediction mode, and four kinds of 8x8 chroma prediction mode. Then the prediction mode with the minimum cost value is selected as the best mode. The residuals after the prediction are further processed by transform and quantization, and reconstructed by inverse quantization and inverse transform, to be the reference of next macroblock. The coefficients after quantization and mode information are encoded by entropy coding for final bitstream output. Details can be referred in [1].







Fig. 2 Nine modes for intra 4x4 prediction

3. THE PROPOSED ALGORITHM AND ARCHITECTURE FOR FAST INTRA CODEC

3.1 Survey of Fast Intra Prediction Algorithm

In the intra coding of H.264, intra prediction and SATD (sum of absolute transformed difference) cost function for mode decision takes almost 77% of computation in all functions [5]. This result is reasonable since each 4x4 block has to check 13 modes for luma prediction and four for chroma prediction. Thus, if we can save some unnecessary prediction modes by a fast algorithm, we can reduce the computation of intra prediction and its related SATD transform.

Various fast algorithms for intra predictions are published to decrease prediction modes efficiently with acceptable performance loss. For example, some algorithms define thresholds for RDO cost [6] or for SAD cost [7] to select the modes to be skipped. The other algorithms use special methods to predict the appropriate modes such as macroblock properties prediction [8], edge map and local edge direction histogram [9], and feature-based mode filtering [10]. They take advantage of the correlation of block pixels to speedup the decision process. Further solution to reduce computation by combining intra prediction with transform [11] is also proposed. However, these researches are mostly developed for software optimization only, but do not consider for hardware design issues like high data dependency [8], expensive hardware gate count cost [9]. As a result, most of the algorithms are hard to be implemented in hardware.



Fig. 3 Decision flow of fast three-step algorithm

3.2 Modified Three-Step Algorithm and Scheduling

A suitable choice for both simple hardware design and low extra computation is to use our previous proposed fast three step intra prediction algorithm [12]. In Fig 3, we first compare the vertical and horizontal modes, and then compare the two neighboring modes around the winner of previous step at the second and third step. Finally, costs of these winners are compared and the minimum one is selected. Though the three-step algorithm is more suitable for hardware design than the other software-based ones, it still has much room for improvement in practice. Directly applying the algorithm to the hardware will result in pipeline bubble and performance loss. To fit the pipeline architecture and schedule in our design, the order of the decision flow should be properly modified.

Fig. 5 (a) illustrates a pipeline schedule applying the three-step algorithm from [12]. In the pipeline stage diagram, each block takes 8 cycles latency to complete a mode prediction process including intra prediction, SATD calculation, and mode decision. In Fig. 5 (a), the first step will take 12 cycles for three modes, and the second step can be immediately executed in the 11th cycle since the comparison results of mode 0 and mode 1 is finished in the 10^{th} cycle. However, this scheduling leads to four cycle bubbles marked in Fig 5 (a). The same situation also occurs between the second step and the third step, and six cycles latency are generated. Therefore, total 28 cycles are required to predict a block with the original fast algorithm.

A solution to conceal the bubble cycles is to adjust the order of prediction modes in the scheduling. Comparing with [12], since the third step in Fig. 5 (a) has to predict either mode 3 or mode 4 no matter which branch is chosen, we modify it by predicting both mode 3 and 4 in the second step and moving the original second step to the third step in order to fill the transition bubbles. Fig. 5 (b) shows the modified three step algorithm. Though the number of prediction modes increase from six to seven, the total cycles to predict a block are reduced to 20 and no bubble cycle exists. Thus, pipeline operations can be executed successively without redundant cycle. Fig. 5 (b) shows the pipeline scheduling .for the modified three step algorithm and Fig. 6 illustrates the decision flow of modified three-step algorithm.



Fig.4 Comparison between proposed modified three-step and the reference software JM8.6 in [1].



Fig. 5 (a) the three step algorithm (b) the modified one.



Fig. 6 Decision flow of modified three-step algorithm

Fig. 4 shows the comparison results between the original mode decision method and the modified ones for four common sequences. It can be observed that using the modified three-step algorithm only increases 0.68% of bitrate in average comparing with JM8.6. Though the bitrate is increased, the number of 4x4 prediction modes can be reduced from nine to seven with 23% of computation saving and the quality of PSNR is almost unchanged.

3.3 Proposed Variable Pixel Parallel Architecture

The proposed intra frame encoder design with the modified fast algorithm and variable-pixel parallel architecture is shown in Fig. 7. This design is mainly partitioned into four phases: prediction phase, reconstruction phase, quantization phase, and bitstream phase. In this design, we adopt variable pixel parallelism instead of constant four-pixel parallelism as in other and our previous design. In the whole design, the prediction phase needs the longest latency and is the target for speedup. Thus, we adopt eight-pixel instead of fourpixel parallelism in the prediction phase. However, to save the cost, we keep the quantization and reconstruction phase in four-pixel parallelism, and are separated from the prediction phase.

The eight-pixel parallel prediction phase significantly improves the throughput and reduces almost half of computation cycles. This improvement will increase a little area overhead because we only add one more intra prediction engine, two 1-D four-point transform, and a few small buffers. Since only blocks with the best mode are passed to the quantization phase and reconstruction phase, these two phases adopt four-pixel parallel architecture to save area. To allow smooth data flow between different data parallelism, we use the current block and best block registers in the quantization phase and the FIFO registers to buffer the data.



Fig. 7 Proposed architecture of encoder with fast algorithm

4. IMPLEMENTATION AND DESIGN COMPARISON

The proposed variable-pixel parallel intra frame encoder is designed by Verilog HDL and implemented using UMC 0.18um 1P6M CMOS technology. When synthesizing at 62.5MHz, the total gate count is about 72K excluding the memory area. Though the usage of eight-pixel parallelism increases the number of necessary functional units, the final gate count does not increase due to the reduction of pipeline stage number and relaxed critical path when compared to our previous design [4].

The chip layout is shown in Fig. 8 with a core size of $1.20 \times 1.20 \text{ mm}^2$, which is 13% smaller than the previous design. Though the 48x64-bit SRAM is larger than the 96x32-bit one used in [4], the total area of three SRAMs is only about 0.27 mm² instead of original 0.30 mm² and has

10% of reduction. The chip has highest frequency at 62.5MHz and exceeds the requiring 61MHz for HD 720p encoding. Table 1 lists the chip information.



Fig. 8 Layout of the fast encoder chip

Table 1 Information for the encoder chip			
Technology	UMC 0.18 um 1P6M CMOS		
Core Voltage	1.8V		
I/O Voltage	3.3V		
Core Size	$1.20 \text{x} 1.20 \text{ mm}^2$		
Package	144 pin CQFP		
On-chip Memory	Single-port 104 x 48-bit x 2		
	banks		
	Single-port 48 x 64-bit x 1 bank		

Design Feature	This Work	[4]	[5]
Max operation freq.	62.5MHz	125MHz	55MHz
System pipeline	MB-based	MB-based	MB-based
Pixel parallelism	8 pixels/4 pixels	4 pixels	4 pixels
CMOS technology	UMC 0.18µm	UMC 0.18 µm	TSMC0.25µm
Gatecount	72K	103K	85K
Chipcoresize	1.20x1.20 mm ²	1.28x1.28 mm ²	1.86x1.86 mm ²
On-chip memory usage	Single 48x64(x1)	Single 96x32(x1)	Single 96x32(x2)
	Single104x48(x2)	Single104x64(x2)	Single 64x32(x1)
			Dual 96x16(x4)
Max target size	HD 1280x720	HD 1280x720	HD 720x480
Freq. for HD 720 p	61MHz	117MHz	N/A
Freq. for SD	23MHz	43MHz	54MHz
Freq. for CIF	6.7MHz	12.8MHz	15.8MHz
Processing cycles/MB	< 560 cycles	< 1080 cycles	< 1300 cycles

Table 2 Comparison with previous designs

Table 2 shows the comparison to other designs. The proposed fast intra frame encoder can support the same size of HD 720p encoding but with only 61MHz. This design can reduce 30% of gate count and 48% of operating frequency compared to our previous design in [4]. Compared to the SD sized encoder in [5], this design can

reduce 58.4% of chip area, 57.6% of operating frequency and 71.6% memory storage bits but with HD sized support.

These improvements are from the proposed optimization such as modified three-step algorithm and variable pixel parallel scheduling.

5. CONCLUSION

A H.264/AVC intra frame encoder with modified fast prediction algorithm and variable-pixel parallel architecture is proposed. Compared to previous designs, this work can reduce 30% of gate count but only with 52% of operating frequency. With such low power techniques, this design is very suitable for the products with demand of low power and portable issues, such as digital video recorder or digital still camera.

6. REFERENCES

- Draft ITU-T Recommendation and Final Draft International Standard of Joint Vedio specification (ITU-T Rec. H.264/ ISO/ IEC 14496-10 AVC), Mar. 2003.
- [2] A. Puri, X. Chen, A. Luthra, "Video Coding Using the H.264/MPEG-4 AVC Compression Standard," Signal Proc. Image Communication, vol. 19, pp.793-849, 2004
- [3] T. Halbach, "Performance comparison: H.26L intra coding vs. JPEG2000", *JVT-D039*, July, 2002.
- [4] C.-C. Cheng, C.-W. Ku, and T.-S. Chang, "A 1280x720 pixels 30 frames/s H.264/MPEG-4 AVC intra encoder," *Proc. IEEE International Symposium onCircuits and Systems*, May. 2006
- [5] 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 Transactions on Circuits and Systems for Video Technology*, pp. 378-401, vol. 15, no. 3, 2005
- [6] F. Fu, X. Lin, and L. Xu, "Fast intra prediction algorithm in H.264/AVC," Proc. IEEE International Conference on Signal Processing, vol. 2, pp. 1191-1194, Aug. 74 2004.
- [7] B. Meng, O. C. Au, C.-W. Wong, H.-K. Lam, "Efficient intraprediction algorithm in H.264," *Proc. IEEE International Conference on Image Processing*, vol. 3, pp. 837-840, Sep. 2003
- [8] C.-L. Yang, L.-M. Po, W.-H. Lam, "A fast H.264 intra prediction algorithm using macroblock properties," *Proc. IEEE International Conference on Image Processing*, vol. 1, pp. 461-464, Sep. 2004.
- [9] F. Pan, X. Lin, S. Rahardja, K.-P. Lim, Z.-G. Li, D. Wu, and S. Wu, "Fast intra mode decision algorithm for H.264/AVC video coding," *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 781-784, Oct. 2004.
- [10] C. Kim, H.-H. Shih, C.-C. J. Kuo, "Feature-based intraprediction mode decision for H.264," *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 769-772, Oct. 2004
- [11] C. Chen, P.-H. Wu, H. Chen, "Transform-domain intra prediction for H.264," Proc. IEEE International Symposium on Circuits and Systems, vol.2, pp. 1497-1500, May. 2005
- [12] C.-C. Cheng, T.-S. Chang, "Fast three step intra prediction algorithm for 4x4 blocks in H.264," Proc. *IEEE International Symposium on Circuits and Systems*. vol.2, pp. 1509-1512, May. 2005