ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.10, No.15, 1–6

An implementation of intra prediction with transform for H.264/AVC

Eunchong Lee, Jeongwoo Yoo, Sangwoo Ye, and Youpyo Hong[a)]

Division of Electronics and Electrical Engineering, Dongguk University-Seoul, 26 3-Ga Pil-Dong Jung-Gu Seoul Korea

a) *yhong@dgu.edu*

Abstract: Tremendous efforts have been made to reduce the cycles per macroblock for intra prediction and transform function blocks for H.264/AVC encoders. We propose a high performance intra prediction and transform logic that can be used for all prediction cases, even when considering the interface with inter prediction results; the overall performance of the encoder is therefore ensured for real-time operation for UHD images.

Keywords: H.264/AVC, intra prediction, inter prediction, transform

Classification: Integrated circuits

References

[1] Y. Huang, B. Hsieh, T. Chen and L Chen: IEEE Trans. Circuits Syst. Video Technol. **15** [3] (2005) 378.

[2] C.-W. Ku, C.-C. Cheng, G.-S. Yu, M.-C. Tsai and T.-S. Chang: IEEE Trans. Circuits Syst. Video Technol. **16** [8] (2006) 917.

[3] C.-C. Cheng, C.-W. Ku and T.-S. Chang: Proc. IEEE International Symposium on Circuits and Systems (2006) 5335.

[4] D. Li, C. Ku, C. Cheng, Y. Lim and T. Chang: ICASSP (2007) II-801.

[5] Y. Lin, C. Ku, D. Li and T. Chang: IEEE Trans. Circuits Syst. Video Technol. **19** [3] (2009) 432.

[6] H. Lin, K. We, B. Liu and J. Yang: IEEE Trans. Circuits Syst. Video Technol. **20** [6] (2010) 894.

[7] G. Jin, J. Jung and H. Lee: IEEE International Symposium on Circuits and Systems (ISCAS) (2007) 1605.

[8] K. Suh, S. Park and H. Cho: ETRI J. **27** [5] (2005) 511.

[9] W. Lee, S. Lee and J. Kim: TENCON 2006 (2006) 1.

[10] W. Lee, Y. Jung, S. Lee and J. Kim: IEEE Trans. Consum. Electron. **53** [4] (2007) 1577.

1 Introduction

Intra prediction is a core function of an H.264/AVC encoder; its efficient hardware implementation is thus essential for many real-time applications. Two approaches can be taken to improve the performance of intra prediction; one approach is to reduce the complexity of intra prediction by

DOI: 10.1587/elex.10.20130510
Received July 01, 2013
Accepted July 16, 2013
Publicized July 25, 2013
Copyedited August 10, 2013

skipping some of the modes at the algorithmic level and the other approach is to increase throughput by using parallelism or a pipelining technique.

Huang et al. presented a VLSI architecture that skips unlikely prediction modes with a reconfigurable predictor generator [1]. Ku et al. presented an intra prediction scheduling technique with macroblock pipelining to avoid idle cycles and improve throughput [2]. Cheng et al. presented an implementation of an H.274/AVC intra encoder by disabling the plane mode [3] and later improved their previous work with the variable-pixel parallel architecture to increase speed while reducing power consumption [4]. Lin et al. also proposed an H.264 intra encoder using a modified three-step fast intra prediction technique to reduce cycle count while keeping the quality close to full search [5]. Lin et al. presented an efficient mode decision algorithm that finds the most probable mode with less image quality degradation [6]. Jin et al. presented a highly pipelined architecture for H.264/AVC intra prediction [7] and Suh et al. presented an efficient architecture for H.264 with full-mode search particularly focused on reducing the execution cycles for intra 16×16 (I16) predictions [8]. Lee et al. proposed rearranging the encoding order of intra 4×4 (I4) predictions to increase the throughput of the pipelined process of intra prediction and transform [9]. Later, they increased the performance of the pipelined I4 prediction and transform by partially overlapping the two operations when possible so that intra prediction of a block can start the operation while the previous block is being transformed [10]. In this paper, a full intra prediction implementation including transform and inter prediction interface is presented. The key contribution is the efficient allocation of dedicated logics and common logics for I4, I6, chroma, and inter prediction.

2 Proposed intra prediction and transform architecture

2.1 Overall architecture

Two intra prediction types for luma data are defined in the H.264/AVC standard: I4 and I16 predictions. The prediction is performed in units of 4×4 blocks in I4 and there are nine modes for I4 depending on the reference pixel positions. Implementing I4 is difficult because a prediction of a 4×4 block may need the prediction results of neighboring blocks belonging to the same macroblock. That is, not only the intra prediction but also the transform for a 4×4 block may need to be finished before starting the prediction of the next 4×4 block, which leads to the cycle increase for I4 prediction of a macroblock. I16 prediction is performed in a unit of a macroblock and the I16 prediction and transform are completely separated. This suggests a relatively straightforward hardware implementation of I16 compared to I4.

Because the computation function of the transform is common for I4 luma, I16 luma, chroma, and inter prediction residuals, many combinations can possibly share transform logics to reduce logic sizes or to reduce cycles for the entire prediction and transform function. We can consider two extreme approaches to handle both luma and chroma data: one that shares logic maximally to reduce circuit size and another that utilizes dedicated logics for I4I, I16, and chroma data to maximize performance. Our primary goal is to maximize the performance of intra prediction and transform to be able to handle UHD images using a reasonable circuit size. Therefore, even

DOI: 10.1587/elex.10.20130510
Received July 01, 2013
Accepted July 16, 2013
Publicized July 25, 2013
Copyedited August 10, 2013

if it is possible to share one transform logic for both I4 and I16, we decided to use separate transform logics for I4 and I16 and share the transform logic for I16 and chroma because I16 requires fewer cycles than I4.

The encoding mode of a chroma block follows the encoding mode of the corresponding luma block according to the H.264/AVC standard. That is, if a luma block is intra-coded then the corresponding chroma block is also intra-coded. Depending on the mode decision, the transform logic of chroma may take input residuals from intra prediction or inter prediction. Therefore, chroma prediction logic may need to wait until the mode decision of the corresponding luma is finished for straightforward implementation. However, to better utilize transform logic and to reduce cycle counts per MB, chroma intra prediction and transform can be performed while I16 prediction is performed. Also, the transform of the residuals from the inter prediction result can be performed while chroma prediction is performed. After all the I16 and I4 predictions are complete, a transform result of the chroma block with the consistent mode will be chosen. The exact sequence and execution cycle of each operation are shown in Fig. 1.

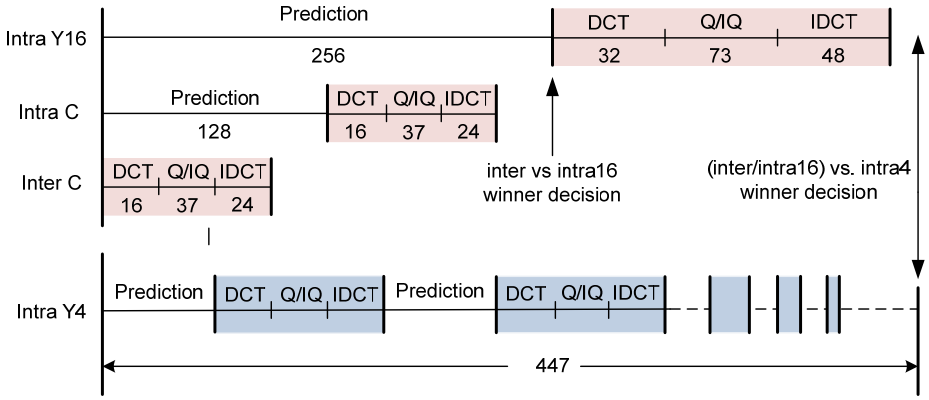

Fig. 1. Timing chart for entire intra prediction and transform

2.2 I4 implementation

Sum-of-absolute-difference (SAD) is one of the most widely used criteria to determine the best mode for prediction due to its simplicity and effectiveness. Two methods can be used for SAD logic configuration for I4 prediction. One method is to reuse single SAD computation logic sixteen times with different reference pixels for each mode which requires nine SAD logics with 16 cycles/block. The other approach is to reuse sixteen SAD logics for all modes which require sixteen SAD logics with 9 cycles/block. The efficiency of overlapping of intra prediction and transform increases if their computation cycles are closer. The computation cycle of transform in our design is 20 cycles/block; therefore, the first approach with 16 cycles/block is chosen for our I4 prediction logic as shown in Fig. 2.

Coded pixel data, i.e. the IDCT outputs, of the last row of all marcoblocks are used as reference pixels for the intra prediction of following macroblocks, so they are stored in SRAMs until they are no longer needed. For the concurrent intra prediction computation, all relevant reference pixels must be provided concurrently. Therefore, we partitioned the

DOI: 10.1587/elex.10.20130510
Received July 01, 2013
Accepted July 16, 2013
Publicized July 25, 2013
Copyedited August 10, 2013

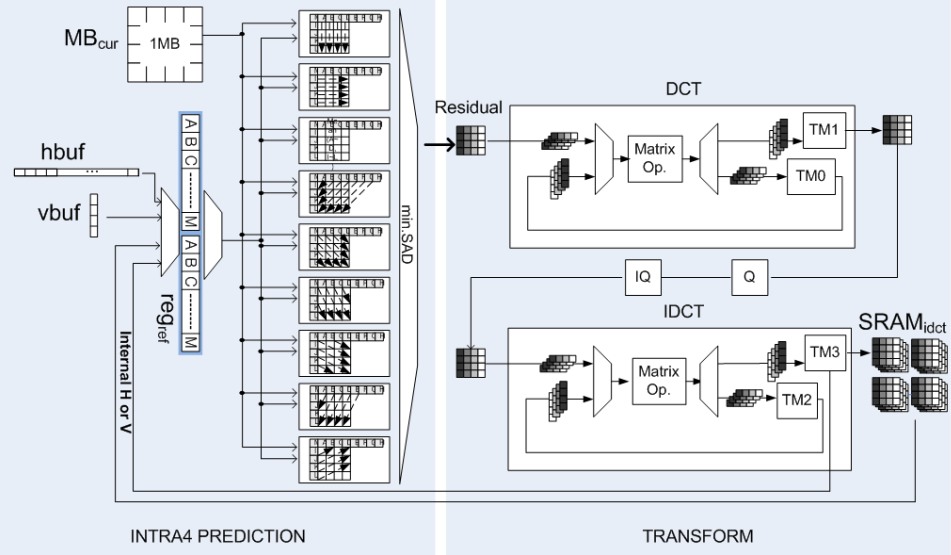

Fig. 2. Intra 4 prediction logic diagram

SRAMS, *hbuf*, into four SRAMs as shown in Fig. 3: $hbuf_0$, $hbuf_1$, $hbuf_2$, and $hbuf_3$. Coded pixel data of the right-most column of a macroblock are stored in a register, *vbuf*, as it does not occupy a large area.

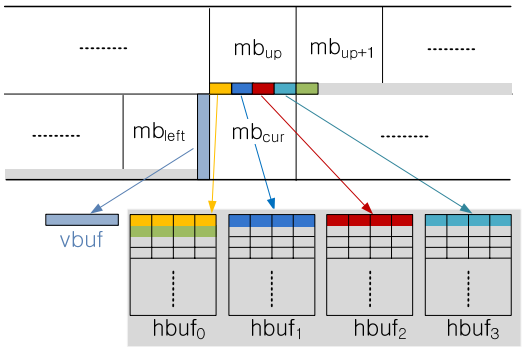

Fig. 3. Buffering scheme for intra4 reference pixels

Most blocks, except the top-left block of a macroblock, need coded pixels from previously processed blocks as reference pixels for I4 predictions. These coded pixels are available from $SRAM_{idct}$, where all IDCT results for a macroblock are stored as shown in Fig. 2. However, if the reference pixels are taken from the SRAM, extra cycles are consumed for the storing to and reading from the SRAM for the prediction. In our design reference, pixels are directly read from the transpose memory TM3 as it consists of registers which allow concurrent multiple read operations for $SRAM_{idct}$ and REG_{ref}, which is a register for the prediction logic. REG_{ref} feeds reference pixels to SAD computation logic and it consists of two register sets so that the read and write operations for the current and next block can be overlapped to save cycles in ping-pong style. That is, while a prediction is being performed for a block using a set of registers in REG_{ref}, the reference pixels for the next block are copied to another set of registers in REG_{ref}.

The detail computation cycles for I4 prediction and transform are shown in Fig. 4. The processing order of the blocks is the same as that proposed by Lee et al. in [9]. In [10], which shows an improved version of that in [9], the total cycle of I4 prediction for a macroblock is reported to be

DOI: 10.1587/elex.10.20130510
Received July 01, 2013
Accepted July 16, 2013
Publicized July 25, 2013
Copyedited August 10, 2013

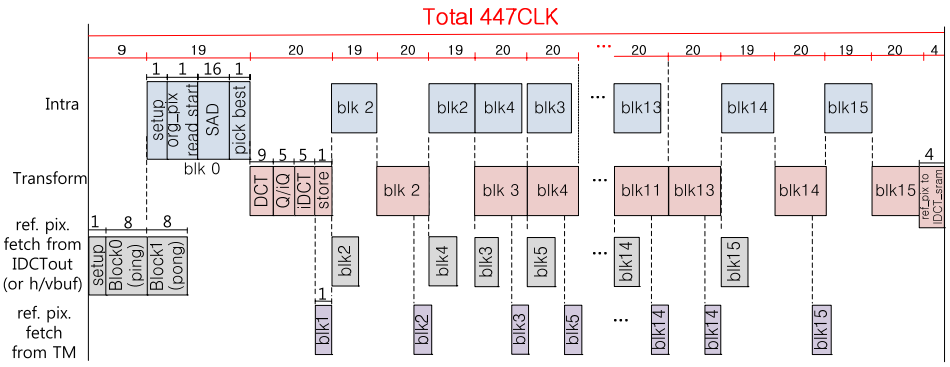

Fig. 4. Timing chart for intra4

522, with a prediction cycle of 25 and transform cycle of 28 for a 4×4 block. In our design, based on the optimizations described in this section, I4 prediction requires 19 cycles, the transform requires 20 cycles for a 4×4 block, and a macroblock requires 447 cycles for the entire I4 prediction.

2.3 I16 and chroma prediction implementation

Because I4 prediction is the limitation in terms of cycles per macroblock, the design goal of I16 prediction is to minimize the circuit size, provided the cycle is fewer than 294, which is the difference between I4 prediction including transform and I16 transform. Therefore, our I16 prediction logic takes one pixel at each clock and computes SADs for five modes in 256 cycles.

3 Results and discussion

We implemented the proposed intra prediction and transform function using Verilog RTL and integrated them into an H.264/AVC encoder. According to simulation results, the proposed intra prediction and transform logic runs at 133 MHz maximum and the throughput is 447 cycles/MB.

Previous works on intra prediction and transform hardware implementation were summarized. Most of these works focused on the intra coder design where the transform of inter prediction residual is not considered. If we take such intra prediction and transform logic for an H.264/AVC encoder with inter prediction function, the latency is increased because the transform logic must be used for chroma transform once the luma prediction is finished. Most of such works [4, 5, 6, 7] used modified intra prediction algorithms to accelerate intra prediction at the cost of intra prediction quality. The transform logic is shared in [7], but the cycles/MB is very low even though I4 and I16 predictions are interleaved due to their fast prediction algorithm. The design presented in [10] shows excellent cycles/MB, but the work only considered I4 prediction.

The design in [8] considered transform for residual from inter prediction after intra prediction is complete. Because the chroma transform waits until intra prediction is completely finished, this design showed 927 cycles/MB. With the same design specification of full intra prediction, considering the inter prediction case, our design shows less than half cycles/MB compared to the design in [8]. 2 K UHD ($2{,}048 \times 1{,}080$) requires less than 513 cycle/MB with 133 MHz and our design is the only one that meets the

DOI: 10.1587/elex.10.20130510
Received July 01, 2013
Accepted July 16, 2013
Publicized July 25, 2013
Copyedited August 10, 2013

Table I. Comparisons results for intra prediction and transform

Feature	Huang [1]	Ku [2]	Suh [8]	Li [4]	Jin [5]	Y. Lin [6]	H. Lin [7]	Lee [10]	Proposed
Intra Pred. Algorithm	fast	fast	full	fast	fast	fast	fast	full	full
Tech. (μm)	0.25	0.18	0.35	0.18	N/A	0.13	0.18	N/A	0.13
Max.Clock (MHz)	55	125	N/A	62.5	N/A	N/A	100	N/A	133
Gate Count (KGate)	85	103	192	72	N/A	95	121	N/A	198
Function	I4 I16 UV	I4 I16 UV	I4 I16 UV inter	I4 I16 UV	I4 I16 UV	I4 I16 UV	I4 I16 UV	I4	I4 I16 UV, inter
Cycle/MB	1,280	1,080	927	560 <	548	560	556	522	447

requirement among the designs in the table.

4 Conclusion

In this paper, we presented an implementation of intra prediction and transform for an H.264/AVC encoder. Key contributions are the efficient resource allocation of intra prediction for luma 4×4, luma 16×16, and chroma blocks to reduce the cycle per macroblock while maintaining a reasonable size. In addition, the proposed design considers the interface with inter prediction results, which are neglected in most of the existing work. The experimental results show that our design achieved the lowest cycles per macroblock.

DOI: 10.1587/elex.10.20130510
Received July 01, 2013
Accepted July 16, 2013
Publicized July 25, 2013
Copyedited August 10, 2013