Algorithm and Architecture Optimization for Full-mode Encoding of H.264/AVC Intra Prediction

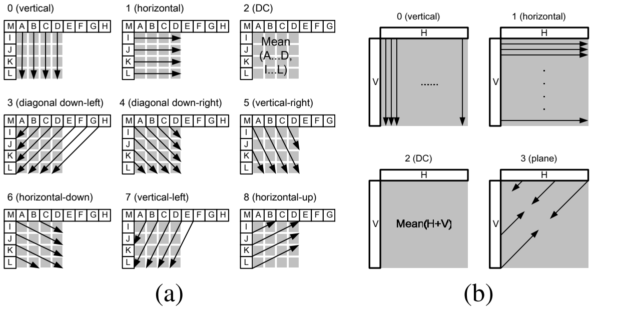
Chen-Han Tsai, Yu-Wen Huang, and Liang-Gee Chen
DSP/IC Design Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan
{chtsai, yuwen, lgchen}@video.ee.ntu.edu.tw

Abstract— **In this paper, we designed a four-parallel intra prediction architecture applied with four optimization schemes. Category-Level Interleaved Scheme (CLIS) eliminates the bubble cycles of I4MB reconstruction. Mode-Level Scheduling (MLS) and Early Data Preparation Scheme (EDPS) rearrange the processing sequence of intra modes. The hardware resource of earlier low-complexity modes is used to deal with the computation of later high-load modes. Not only the hardware utilization is increased but the processing cycles is reduced. Furthermore, Stage-Level Partial Distortion Elimination (SLPDE) is induced to skip the calculation of unnecessary intra modes. The architecture has been integrated into an H.264/AVC baseline encoder for HDTV applications and has been verified to be feasible under system consideration.**

I. Introduction

H.264/AVC [1] is the latest video coding standard. It can attain the same quality with relatively low bit-rate, at only 63% compared to MPEG-4 ASP and 36% compared to MPEG-2. Intra Prediction plays an important role in H.264 coding flow. H.264/AVC intra frame coder is competitive with the latest image coding standard, JPEG2000, in coding performance. Intra prediction is the most critical tool among those of an intra frame coder. Besides, intra prediction is essential in inter-frame also because some inter-frame macroblocks will choose intra type as coding mode.

In H.264/AVC intra coding, two intra macroblock modes are supported. One is intra 4×4 prediction mode, denoted as I4MB, and the other is intra 16×16 prediction mode, denoted as I16MB. For I4MB, each 4×4-luma block can select one of nine prediction modes. In additional to DC prediction, eight directional prediction modes are provided. Fig. 1 (a) shows the illustration of 4×4-luma prediction modes. The 13 boundary pixels from previously coded blocks are used for predictor generation. Fig. 1(b) shows the illustration of 16×16-luma prediction modes. For I16MB, each 16×16-luma macroblock can select one of the four modes. Mode 3 is called plane prediction, which is an approximation of bilinear transform with only integer arithmetic. Chroma intra prediction is independent of luma. Two chroma components are simultaneously predicted by one mode only. The provided chroma prediction modes are very similar to those of I16MB except different block size (chroma is 8×8) and some modifications of DC prediction mode. The mode information of I4MB requires more bits to represent than that of I16MB. I4MB tends to be used for highly textured regions while I16MB tends to be chosen for flat regions.



(a) (b)

Fig. 1. Illustration of (a) nine 4×4, (b) four 16×16 luma prediction modes.

The rest of the paper is organized as follows. In Section II, we will introduce the challenge of H.264/AVC intra prediction. Then, four schemes in different levels will be stated in Section III to enhance hardware efficiency. Section IV shows the implementation results, and we finally conclude our work in Section V.

II. Challenge of H.264 Intra Prediction

Intra prediction is a new coding tool of H.264/AVC video coding standard. It contains thirteen luma and four chroma prediction modes. According to the predicted block size, thirteen luma modes can be divided into two categories, nine 4×4 block-based (I4MB) and four 16×16 block-based modes (I16MB). Due to the variety of modes, it is hard to design an efficient intra prediction engine to support all modes. To solve this, our previous work designed a reconfigurable hardware capable of calculating all prediction modes [2]. However, there are still some problems which lower the hardware efficiency:

A. I4MB Reconstruction Loop

In I4MB mode, for each 4×4 blocks, 13 boundary pixels of reconstructed blocks are required for intra prediction. Due to the processing flow defined by H.264/AVC, intra prediction of 16 4×4 blocks will be processed by zig-zag scan order as shown in Fig. 2. According to these, we can not calculate intra predicted pixels of 16 blocks in parallel. For example, when block 1 in Fig. 2 is processed, its left four boundary pixels belong to block 0. That is, we can not do the intra

0-7803-9197-7/05/$20.00 © 2005 IEEE.

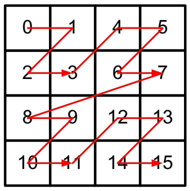


Fig. 2. Zig-zag processing order of 16 4×4 blocks in a macroblock.

prediction of block 1 until the reconstruction of block 0 is done. Unfortunately, we also can not do the reconstruction of block 0 if mode decision of this block is not finished. Mode decision will need the costs of all intra prediction modes. So, a loop of I4MB reconstruction occurs and that introduces bubble cycles of the prediction engine. Then the processing time of intra prediction will be prolonged and the hardware utilization will be lowered.

B. Computation Load Difference

There are 17 intra prediction modes in a macroblock. In our previous work [3], these 17 modes are divided into four categories by the configuration of processing element (PE): bypass configuration for vertical and horizontal modes, cascading configuration for DC modes, recursive configuration for plane prediction mode and normal configuration for the remaining I4MB modes. The computational complexity of these four configurations is quite different. For example, I16MB vertical mode just passes the upper boundary pixels as predicted pixels, but I16MB plane prediction needs 14 multiplications and 18 additions, while most I4 modes require three additions only. Additional hardware resource or processing cycles are needed for plane prediction while almost whole engine sleeps when processing vertical mode. This computation difference makes the utilization of prediction engine further decreased.

C. I16MB Plane Prediction Mode

Several previous works eliminated plane prediction mode because of its high computation load and relative low occurring percentage in small size sequences. However, for larger size or smooth images, plane prediction mode can greatly improve the R-D curve performance. Thus plane prediction is very important for high-end digital camera and HDTV video applications. But its computation complexity is really high: Fig. 3 shows the diagram and formulas of I16MB plane prediction. 18 multiplications and 17 additions are required for it of a macroblock. Besides, the computation type of this mode is quite different from other modes. The hardware resource sharing becomes a tough task and it is harmful to the hardware utilization also.

III. Optimization Schemes

The problems of intra prediction mentioned above lower the hardware utilization or increase the processing cycles. In this section, based on the four-parallel reconfigurable prediction engine in our previous work [3], we will further introduce the proposed optimization schemes in different levels to improve both hardware efficiency and processing time.

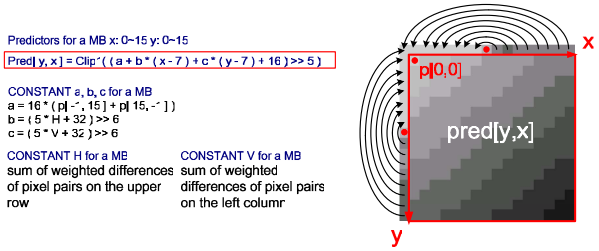

Predictors for a MB x: 0~15 y: 0~15
Pred[y, x] = Clip1((a + b * (x - 7) + c * (y - 7) + 16) >> 5)
CONSTANT a, b, c for a MB
a = 16 * (p[-1, 15] + p[15, -1])
b = (5 * H + 32) >> 6
c = (5 * V + 32) >> 6
CONSTANT H for a MB
sum of weighted differences of pixel pairs on the upper row
CONSTANT V for a MB
sum of weighted differences of pixel pairs on the left column
p[0,0]
pred[y,x]
x
y

Fig. 3. Computation of I16MB plane prediction.

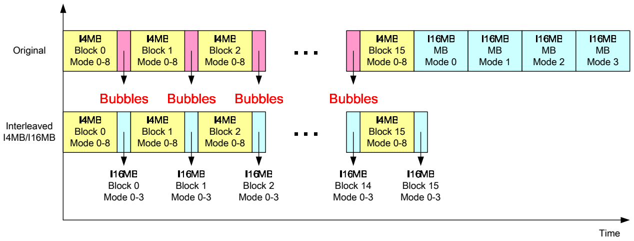


Fig. 4. Cycle-reduction by Category-Level Interleaved Scheme.

A. Category-Level Interleaved Scheme (CLIS)

I4MB reconstruction loop brings bubble cycles. That is, a new 4 × 4 block should wait for the reconstruction process of previous 4 × 4 block. Except I4MB modes, there are still four I16MB prediction modes to be calculated. Therefore, to eliminate these bubbles, we divide I16MB prediction into 16 spatial parts, 16 4 × 4 blocks, and insert them into the corresponding 4 × 4-block I4MB bubble cycles. The two categories, I4MB and I16MB, will be interleaved by this scheme. Fig. 4 shows how Category-Level Interleaved Scheme eliminates I4MB bubble cycles.

The processing cycles will be reduced by applying CLIS. However, DC coefficients of the four I16MB modes have to be stored in registers because the distortion cost of I16MB modes in the reference software requires to further apply 4 × 4 2-D Hadamard transform on the 16 DC values. This is a tradeoff between speed and area.

B. Mode-Level Scheduling (MLS)

In CLIS, I4MB bubble cycles are used to calculate I16MB predictions. But the number of I16MB cycles is not exactly the same as that of I4MB bubble cycles. Now we further reduce both of them to improve processing time of whole intra prediction.

I4MB bubble cycles come from the data dependency between current block and left or upper adjacent block. According to zig-zag scan order, left boundary pixels are more critical than upper ones because no two sequentially processed blocks will be vertically adjacent. So we focus on left boundary data dependency. Among nine I4MB prediction modes, only vertical prediction mode needs no left boundary pixels to predict.

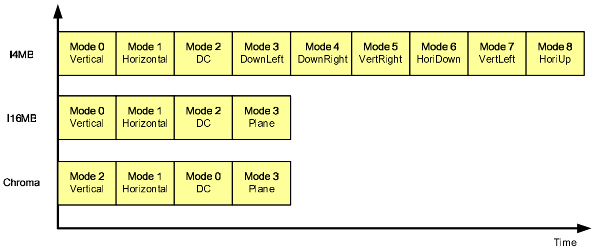


Fig. 5. Mode-Level Scheduling of I4MB, I16MB and Chroma Prediction Modes.

TABLE I
COMPUTATIONAL LOAD OF EACH LUMA MODE.

	Vertical	Horizontal	I4 Normal	I16 DC	I16 Plane
addition	0	0	3	31	17
muplication	0	0	0	0	18

That is, the computation of this mode will not be restricted by the reconstruction process of previous block. Therefore, we rearrange the processing sequence of modes. Those who have no data dependency or have lower computational load will be calculated earlier. Fig. 5 shows Mode-Level Scheduling of I4MB, I16MB and Chroma prediction modes. We can see vertical mode is first calculated. Following is horizontal mode, and then normal configuration I4MB modes are processed. Plane prediction mode is always the last mode to be dealt with.

C. *Early Data Preparation Scheme (EDPS)*

Table I lists the number of additions and multiplications required by each luma mode. Most modes require only three additions and bypass modes need no computation. But the requirement of DC or plane mode is much higher than others. An intuitive solution is to design a hardware engine capable of processing most modes, and additional time or other area is spent for DC and plane mode. However, this method may have area overheads, timing overheads or both.

To expound EDPS more clearly, we assume the parallelism of prediction engine and reconstruction hardware is four. And we apply the requirement of most modes, three adders, for each processing element of intra prediction engine. The hardware can compute four 4-to-1 additions in each cycle. For those two high-complexity mode, I16 DC mode needs a 32-to-1 addition and I16 plane mode needs to realize a, b and c in the formulas of Fig. 3. After applying MLS, the first two modes, vertical and horizontal, need no computation. So we can utilize these cycles to compute partial results of DC and plane mode. Fig. 6 shows how EDPS reduces the I16MB processing cycles by balancing the computation load of each mode.

D. *Stage-Level Partial Distortion Elimination (SLPDE)*

I4MB modes need to do reconstruction of current 4×4 block before the next block starts to calculate, so intra stage is usually combined with final mode decision and the reconstruction engine. That is, intra prediction is usually processed later than inter prediction. According to the four-stage macroblock pipelining in our previous work [4], if we partition intra and inter predictions into different pipeline stages, the best inter cost will be calculated before the intra prediction process of this macroblock starts. Therefore, when the accumulated costs of all intra prediction modes are larger than the best inter cost, we can skip intra prediction and choose inter type as the coding mode of this macroblock. Furthermore, inter prediction is usually better than intra prediction. That is, the cost of best inter prediction is usually much smaller than the best inter prediction if the motion is correctly found. So SLPDE is very powerful for processing-cycle reduction.

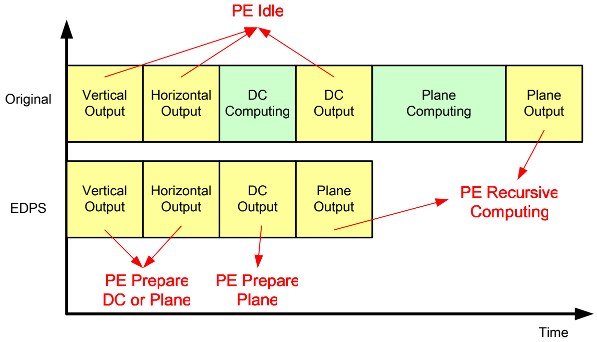


Fig. 6. Early Data Preparation Scheme of I16MB.

TABLE II
HARDWARE INFORMATION OF LUMA INTRA PREDICTION ENGINE.

Technology	UMC 0.18um 1P6M CMOS
Logic Gate Count	10483 gates
Max. Clock Rate	120 MHz
Processing Capability	33 fps for 4:2:0 HDTV 720p (1280x720)
	88 fps for 4:2:0 SDTV (720x480)
	30.72 Mega-Pixels within 1 sec

IV. IMPLEMENTATION RESULTS

Applied with the four schemes mentioned above, we developed a four-parallel reconfigurable intra prediction generator to achieve resource sharing between all kinds of prediction modes. Below shows the hardware implementation results and the performance of each scheme:

A. *Hardware Implementation Results*

Fig. 7 depicts the architecture of our new reconfigurable intra prediction generator with four proposed schemes. Table. II lists the hardware information of the proposed luma intra prediction engine. Noted that MLS and EDPS have no area overheads. Registers of intra prediction engine can be reused for the required data buffer of CLIS. The hardware overheads are only two comparators (13-bit and 17-bit) and the skip control circuits for SLPDE.

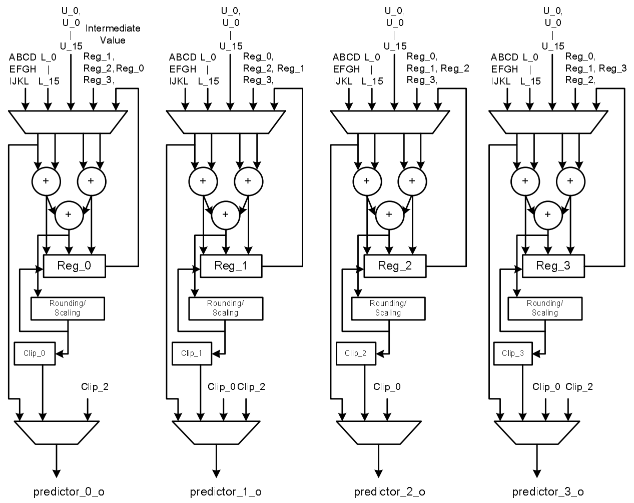


Fig. 7. The architecture of four-parallel luma intra prediction engine with four proposed schemes.

TABLE III

THE PERFORMANCE OF FOUR SCHEMES.

	Processing Cycles per MB (cycle)	Reduction Rate (%)
Original	(36+4+20)*16 + (26)*16 = 1376	0
CLIS only	(36+4)*16 + (26)*16 = 1056	23.5
MLS only	(36+4+16)*16 + (26)*16 = 1312	4.7
EDOS only	(36+4+20)*16 + (16)*16 = 1216	11.6
Apply Three schemes	(36+4+16)*16 = 896	34.9

B. Scheme Performance

The number of luma prediction cycles of a macroblock can be calculated by the following equation:

$$C_{MB} = (C_{9I4Modes} + C_{BestI4Mode} + C_{Rec}) \times 16 + (C_{4I16Modes}) \times 16; \qquad (1)$$

where $C_{9I4Modes}$ and $C_{BestI4Mode}$ stand for the cycle number of intra prediction for all I4MB modes and for the mode which has the smallest cost, respectively. C_{Rec} is the required cycles of a 4×4-block reconstruction process, and $C_{4I16Modes}$ means the cycles for all I16MB intra prediction modes.

Here we evaluate the performance of four schemes: For a four-parallel architecture without any scheme applied, all I4MB modes require four cycles, while I16MB DC and I16MB plane modes cost eight and ten cycles to predict a 4×4 block, respectively. CLIS inserts I16MB process into I4MB reconstruction bubbles, so only the longer one, I16MB process, is remained. MLS reduces the processing cycles by earlier computing the vertical mode. EDPS optimizes the number of needed cycles for each I16MB mode to four by balancing the computation load. Table. III calculates the cycle-reduction performance of four proposed schemes. The performance column lists the corresponding parameters of (1).

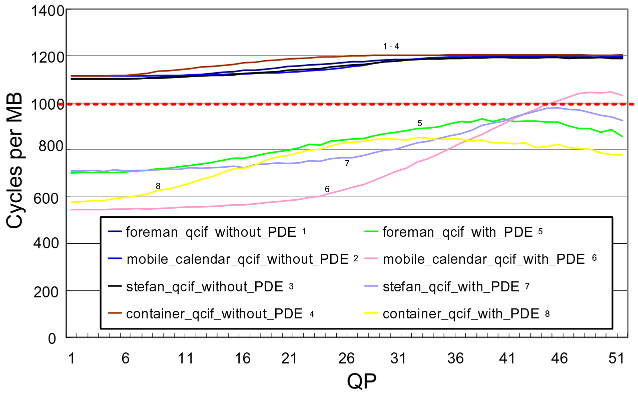


Fig. 8. Cycle reduction performance of Stage-Level Partial Distortion Elimination.

Besides these three schemes, SLPDE utilizes the best inter cost as a upper threshold. When the accumulated costs of all intra prediction modes are greater than that, we will terminate the intra prediction to reduce processing cycles. Fig. 8 shows the performance of Stage-Level Partial Distortion Elimination. The effect of SLPDE differs sequence by sequence. Generally, it can further reduce 20% to 50% of processing time.

If we apply these four schemes together, the hardware can achieve 47.9% to 67.5% cycle-reduction with near 100% utilization. It should be noted that these four schemes will not skip any possible intra prediction mode so that they will not introduce any quality loss.

V. CONCLUSION

This paper presents four schemes to enhance the hardware efficiency. We analyze the problems causing unnecessary processing cycles or hardware resource and then propose corresponding solutions in different levels. A four-parallel intra prediction engine applied with these four schemes has been proposed and integrated into an H.264/AVC baseline encoder for HDTV applications [5]. It can achieve 47.9% to 67.5% cycle-reduction with near 100% utilization under real chip environment.

REFERENCES

[1] 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.

[2] Y.-W. Huang, B.-Y. Hsieh, T.-C. Chen, and L.-G. Chen, ”Hardware architecture design for H.264/AVC intra frame coder,” in *Proceedings of 2004 IEEE International Symposium on Circuits and Systems 2004*.

[3] 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, vol. 15, pp. 378-401, March 2005.

[4] T.-C Chen, Y.-W. Huang, and L.-G. Chen, ”Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture,” in *Proceedings of 2004 IEEE International Symposium on Circuits and Systems*.

[5] Y.-W. Huang, T.-C. Chen, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, C.-S. Chen, C.-F. Shen, S.-Y. Ma, T.-C. Wang, B.-Y. Hsieh, H.-C. Fang, and L.-G. Chen, ”A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications,” in *Proceedings of 2005 IEEE International Solid-State Circuits Conference*.