Low Complexity Architecture Design of H.264 Predictive Pixel Compensator for HDTV Application

Jia-Wei Chen[1], Chien-Chang Lin[2], Jiun-In Guo[2], and Jinn-Shyan Wang[1]
[1]Department of Electrical Engineering, National Chung-Cheng University
[2]Department of Computer Science and Information Engineering, National Chung-Cheng University
Chia-Yi, Taiwan, Republic of China
92jiawei@vlsi.ee.ccu.edu.tw, lcc90@cs.ccu.edu.tw, jiguo@cs.ccu.edu.tw, ieegsw@ccu.edu.tw

Abstract—**In this paper, we propose a low-complexity architecture design of H.264 predictive pixel compensator (PPC) for HDTV application. In intra prediction, we propose a shared adder-based architecture style that supports all of the 17 intra prediction modes, and reduce computational complexity in the I4MB prediction mode 3~8 up to 50% computation. Besides, we have also proposed the distributed memory access to improve the HW usage. As well as, it can used to reduce the memory size for buffering the neighboring pixels. In inter prediction, we can save about 48% of external memory bandwidth by the data reused through the hybrid block size memory access. Adopting the mixed six-tap FIR filter architecture to design luma interpolation, we can efficiently reduce the hardware cost up to 27%. The implemental result shows the hardware cost of the proposed design is about 60854 gates under a TSMC 0.18μm CMOS technology, which achieves the real-time processing requirement for HD-1080 format video@30Hz at the working frequency of 87 MHz.**

I. INTRODUCTION

The newest video standard, H.264 [1], is developed by ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) jointly for next generation multimedia coding applications. Compared to the previous MPEG standards, H.264 provides at least two times compression ratio for higher video coding quality [2]. H.264 supports several efficient coding tools, including inter prediction for motion compensated video frames (i.e. P-frame and B-frame) and intra prediction for I-frame. In intra prediction, the macroblock pixel predictor is generated by neighboring reconstructed samples. Compared with JPEG-2000 standard applied on the compression of still images, H.264 intra coding possesses about 0.1~0.8 dB gain on PSNR as compared to JPEG-2000 DWT 53 [3].

Another efficient tool in H.264 is the variable block size motion estimation that contributes the main coding gain [4] in temporal prediction. Considering the high resolution video decoder applications in the real-time processing requirement, the memory bandwidth becomes the bottleneck for variable block size motion compensation. According to the performance of processor-based H.264 video decoders [5][6][7], the computational complexity of H.264 Predictive Pixel generator (intra + inter) is about 50% of the total decoding time and the memory bandwidth requirement is up to 528MB/s [10]. On the other hand, the intra/inter prediction and memory bandwidth are critical to influence the system performance in H.264 video decoder. According to the bandwidth analysis, the extensive frame-samples are needed to reconstruct the predictors in inter prediction. There are 64% memory bandwidth needed for inter prediction. The system frequency of 200MHz is needed to fit this requirement when single external memory interface (32-bit SDRAM) is equipped. So, the complexity and memory bandwidth are the keys to design an efficient Predictive Pixel Compensator (PPC) for H.264 video decoding system. By exploiting the characteristic in reference data, if the Luma and Chroma reference frame buffer are separated to different memory and dual memory interface are adopted, the working frequency can be reduced to 140MHz, but it’s still too high in the chip implementation. Therefore, hybrid block size memory access is proposed in this paper for inter prediction to reduce the memory bandwidth and improve the system throughput.

The estimation of computational complexity is necessary for optimizing the hardware cost in intra prediction. If the throughput of system performance is ‘1’ (sample/cycle), the working frequency is only 1.2 MHz to support the real-time decoding on QCIF video @30Hz. Therefore, in the proposed design, the target specification is to support the real-time processing on HD1080 video @30Hz. The system only needs to work at 95 MHz to achieve the performance of 1 pixel/cycle throughput.

In order to achieve the goals of efficient hardware implementation, we propose a shard adder-based architecture to support all 17 kinds of intra prediction modes by exploiting the concept of shard terms. Besides, the distributed memory access is proposed to achieve the maximum memory utilization. On the other hand, a novel interpolation based on separate 1-D approach with the proposed mixed six-tap FIR is implemented. According to these proposed techniques, we can provide high performance and low cost PPC design for H.264 baseline/main profile video decoding targeting at HDTV applications.

The rest of this paper is organized as follows. In Section Ⅱ, we describe the proposed low complexity PPC design. In Section Ⅲ, we show the implementation results of proposed PPC design. Finally, we concludes this paper in Section Ⅳ.

II. PROPOSED DESIGN

Because the neighbor reconstructed pixel dependency of intra prediction, inter prediction, and compensation are combined in the PPC module to reduce the complexity in data processing and control flow, as illustrated in Fig.1. The decoding information of one macroblock can be shared by intra and inter prediction to reduce hardware cost. In intra prediction mode, we propose the

1-4244-0469-X/06/$20.00 ©2006 IEEE

distributed memory access flow to optimize the memory usage. In order to reduce the computational complexity, we propose the shared adder-based technique to reduce the calculating operation amount up to 50% in the I4MB prediction mode 3~8 as compared to original design.

As for inter prediction, to reduce external memory bandwidth and working frequency are important for reducing the working frequency and optimizing the power consumption. In order to reduce the memory bandwidth, by exploiting data reuse in inter prediction, the hybrid block-size memory access is adopted. In addition, the interpolator of mixed six-tap FIR filter based on the shared terms concept is proposed to reduce hardware cost in the Luma interpolation mode. In the Chroma interpolation mode, we propose a two-level chroma interpolation architecture to improve the performance. And the WP tool is also supported for main profile. Finally, the reconstructed pixels are generated and write back the predictors to the buffer for further reused.

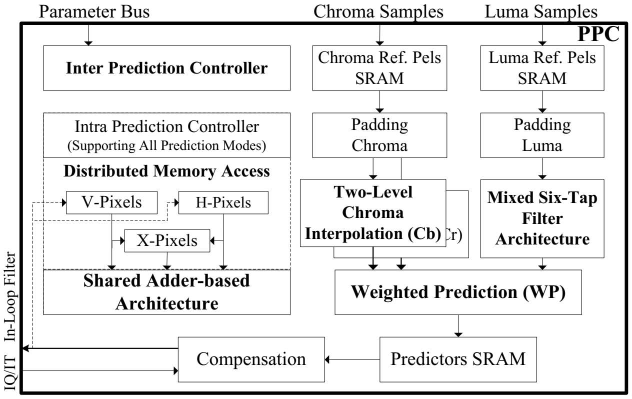
Parameter Bus
Chroma Samples
Luma Samples
PPC
Inter Prediction Controller
Chroma Ref. Pels SRAM
Luma Ref. Pels SRAM
Intra Prediction Controller (Supporting All Prediction Modes)
Distributed Memory Access
V-Pixels
H-Pixels
X-Pixels
Shared Adder-based Architecture
Padding Chroma
Padding Luma
Two-Level Chroma Interpolation (Cb)
(Cr)
Mixed Six-Tap Filter Architecture
Weighted Prediction (WP)
Compensation
Predictors SRAM
IQ/IT In-Loop Filter

Fig. 1 Block diagram of the proposed PPC module

A. Distributed Memory Access Flow

In the intra prediction, traditional design must use a lot of memory to save the neighboring pixels as well as do not have the solutions to store the predictor-Q effectively in the memory, as illustrated in Fig. 2. In order to optimize the memory access in intra prediction, we proposed a distributed memory access flow, which will store each predictor-Q in 4x4-block to the X-memory and make full use of the H-memory and V-memory to reduce the memory size and the complexity. According to this technique, we only need 320-bits and 640-bits memory size for supporting H.264 baseline profile and main profile.

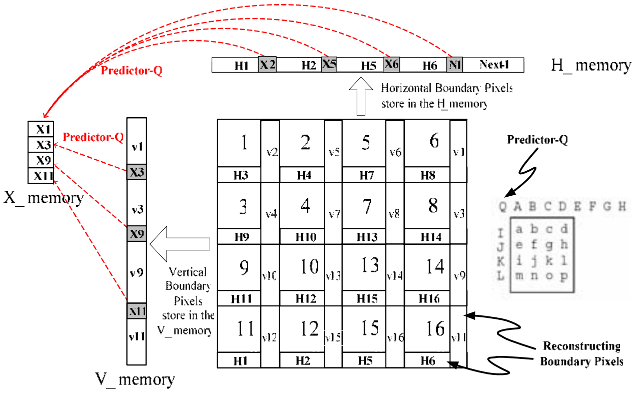

Fig. 2 Illustration of distributed memory access

B. Prediction Mode Optimization

By exploiting the regularity of I4MB prediction modes 3~8 operations, three facts are implied, take mode N for example as illustrated in Fig. 3. First, some output samples have the same results (e.g. samples b and e are the same) which can be used to reduce the complexity. Second, the common terms can be shared in computing different output samples (e.g. term (B+C) can be shared). Finally, computing different output samples can be viewed as the same filtering operation on shifted input samples. So that we can use the same hardware circuit to carry out all the filter operations for reducing hardware cost. Exploiting the regularity existed in prediction operations can contribute the 50% complexity reduction in PPC. As well as, through the proposed shared adder-based generator as showed in Fig. 4, we can support all 17 kinds of intra prediction modes. In the horizontal and vertical prediction modes, the predictors are bypassed from the input to the output. Besides, the predictors can be constructed by an accumulator in the DC prediction mode. Finally, due to the computation complexity that is too high in the plane prediction mode, the input port, a3, is used to increase the performance in the proposed architecture.

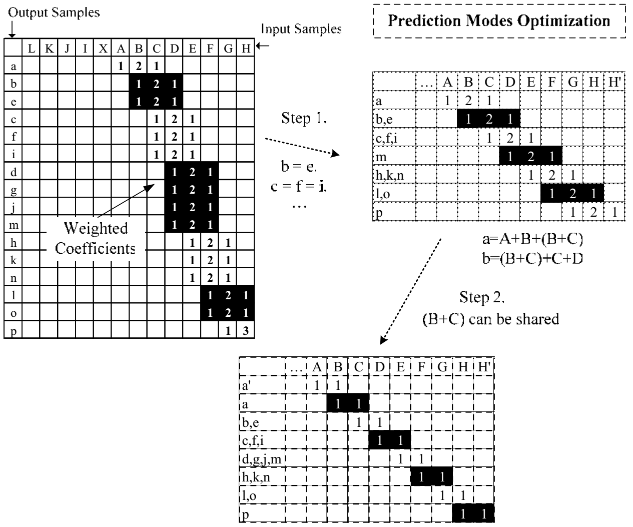

Fig. 3 Example of operation sharing in Diagonal_Down_Left Mode

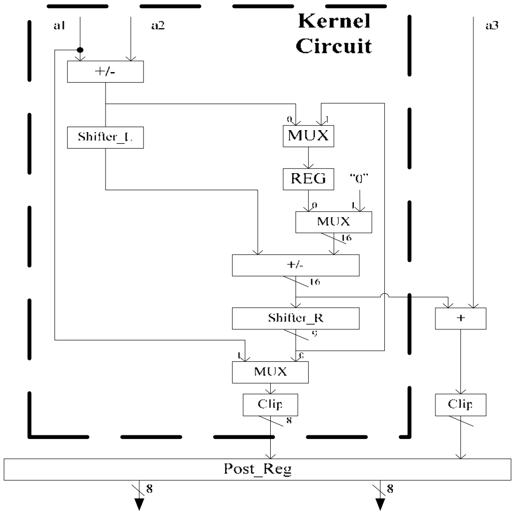

Fig. 4 Proposed shared adder-based intra prediction generator

C. Exploiting Data Reuse in Hybrid Bblock Ssize Memory Access Operations

According to analysis, the memory bandwidth is quite huge if the inter prediction is 4x4 block based. So, it is the bottleneck in inter prediction for H.264 video decoding system. In order to make full use of overlapped data in neighboring block, the hybrid block size operation is proposed to decrease the access counts in external frame memory. Considering the Luma and Chroma interpolation, the *M*x*N* {M, N | M, N∈2, 3, 4, 5, 8, 9, 13} variable block size mode operation is equipped. That support Luma and Chroma operation from 4x4 up to 8x8 block size and from 2x2 up to 4x4 block size respectively in integer or fractional pels. Overlapped reference frame pixels of larger than block-4x4s can be shared and reused to reduce the memory bandwidth. As Fig.5 example, the 9x5 overlapped shaded data is reused in 4x8 block size with fractional pel MV, and 5x9 overlapped data is reused in 8x4 block size with fractional pel MV. According to the experimental results, the memory bandwidth can be contributed up to 48% reduction.

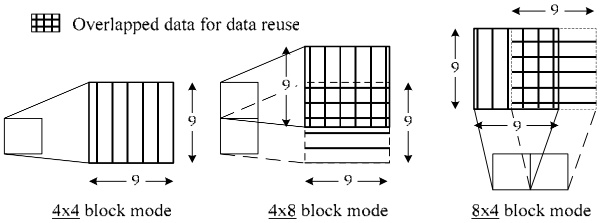

Fig. 5 Illustration of 4x4, 4x8, and 8x4 block mode with 1/4 MV

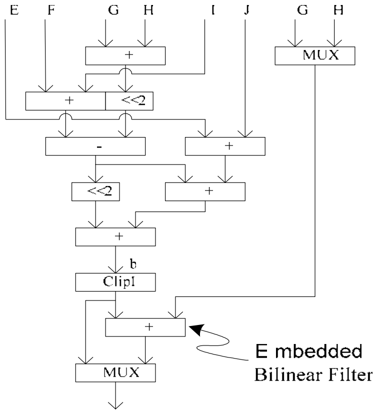

Fig. 6 The interpolator of mixed six-tap FIR filter architecture

D. Interpolator of Mixed Six-Tap FIR Filter Architecture

The detailed definition of inter prediction for half-pixel and quarter-pixel is described as follows, respectively.

$$\begin{aligned} b &= (E - 5F + 20G + 20H - 5I + J + 16) >> 5 \\ &= \{\{(E + J) + [(G + H)*4 - (F + I)]\} + \\ &\qquad [(G + H)*4 - (F + I)]*4\} >> 5 \end{aligned} \qquad (1)$$

$$a = (G + b)/2 \qquad (2)$$

$$c = (H + b)/2 \qquad (3)$$

In Eq. (1), the value "[(G+H)*4-(F+I)]" is the shared term that can be used to reduce the hardware cost. In addition, the quarter-pixel samples "a" or "c" are generated with values "b", "G", and "H" by using bilinear interpolation specified in Eq. (2) and (3). The integer samples "G" and "H" have been used to calculate the half-pixel sample "b". Therefore, we can share the integer samples "G" and "H" to generate half-pixel sample "b" first and produce the quarter-pixel samples "a" and "c" at the same time, respectively. According to the evaluation, a new Luma interpolation architecture based on mixed six-tap FIR filter architecture is proposed to reduce the hardware cost, as shown in Fig. 6.

E. Interleaved Chroma pixel and Two-Level Chroma Interpolation Architecture

In addition to variable block size operation adopted to reduce the memory bandwidth, the interleaved Chroma data, Cb and Cr, is also used for memory bandwidth reduction. Because the MVs of Cb and Cr are the same, the Cb and Cr are placed interleaved in the frame buffer. That result in only one memory read command is needed to grab a chroma row data from external memory, and 50% chroma memory read command can be easily reduced.

According to analysis for chroma interpolator, the neighboring components "X0" and "X1" in the vertical direction are generated by the following equations:

$$\begin{aligned} X0 = ((\alpha' * A0) * \beta' + (\alpha * A1) * \beta' + \\ (\alpha' * B0) * \beta + (\alpha * B1) * \beta) / 64 \end{aligned} \qquad (4)$$

$$\begin{aligned} X1 = ((\alpha' * B0) * \beta' + (\alpha * B1) * \beta' + \\ (\alpha' * C0) * \beta + (\alpha * C1) * \beta) / 64 \end{aligned} \qquad (5)$$

then, α = dx, α' = 8-dx, β = dy, and β'= 8-dy.

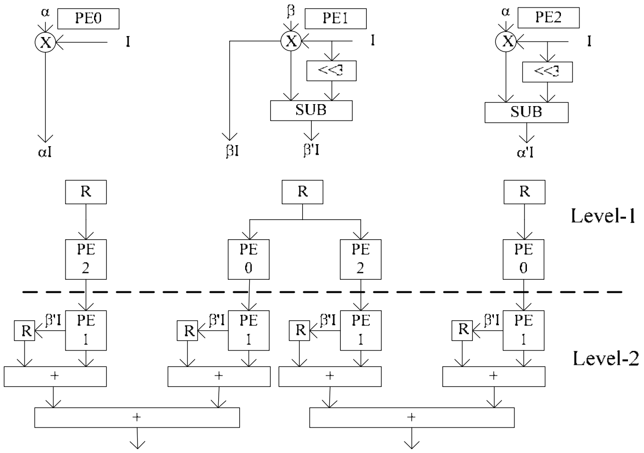

Fig. 7 Two-level chroma interpolation architecture

In the proposed design, the shared terms "(α*B0)" and "(α'*B0)" can be constructed by PE0 and PE2 as shown in Fig. 7. By the PE1, the values "(α*B0)*β" and "(α*B1)*β" can be generated with multiplying the value "β", respectively. On the other hand, the PE1 can also provide the value "β'". Therefore, the values "(α*B0)* β'" and "(α*B1)*β'" can be generated at the same time with multiplying the component "β'", respectively. As well as, it can be stored in the register for constructing next predictor. Therefore, constructing four predictors only need three cycles in the proposed design. Finally, we use the three adders to generate each predictor, as shown in Fig. 7. Interpolation of each chroma component in each 8x8-MB needs 48 cycles in worst case.

F. *Proposed Weighted Prediction Architecture*

The weighted prediction is efficient to increase the video quality in H.264 video decoding. In the proposed design, a four-parallel WP architecture is proposed to reduce the memory requirement between inter prediction and WP tool. Take List0 only for example, the equation of WP tool is shown in the following

$$\text{If}\,(\text{logWD}>=1)$$

$$\text{SumLx} = \text{Clip1}(((\text{Inter}[x, y] * \text{w0} + 2^{\text{logWD-1}}) >> \log WD) + o_x) \quad (6)$$

$$\textit{Else}$$

$$\text{SumLx} = \text{Clip1}(\text{Inter}[x, y] * \text{w0} + o_x) \quad (7)$$

Where, input data "Inter" is decoded from the inter prediction. The parameters, logWD, Wx, and Ox are generated from the sequence parameter set, and w0 are generated from software. As well as, they will be stored into the WP table buffer. The component of $2^{\text{logWD-1}}$ and shift right logWD operation can be combined and be viewed as a rounding operation. Besides, a buffer is required for B MB decoding and the proposed architecture is shown in Fig. 8.

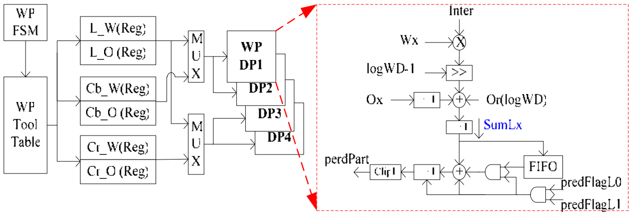

Fig. 8 Weighted prediction architecture

III. IMPLEMENTATION RESULTS AND COMPARISON

The implementation results are shown in Table 1. The neighboring samples of intra prediction are stored through the technique of proposed distributed memory access to increase the memory usage. In the inter prediction, the reference samples will be pre-stored by interlaced schedule to reduce the access latency as illustrated in Fig.1. The predictors SRAM are used to save the predictors after calculating the interpolation for supporting the variable block sizes. In order to support the WP, we need a SRAM to save the parameters form the bitstream parser. The other SRAM is used for supporting WP in B-slice. Finally, the implement results show the proposed design needs 60854 gates and 8.7K bits SRAM at the working frequency of 150 MHz.

From the comparison results listed in Table 2, we find that the proposed design outperforms the existing inter prediction decoder design [4] to save about 13% reduction in required working frequency for HD1080 video format and 7% reduction in inter prediction total gate count. Compared to the other design [4], the proposed design can save about 27% reductions in interpolator gate count. On the other hand, the proposed interpolator of mixed six-tap FIR filter is efficient to reduce the hardware cost.

IV. CONCLUSION

This paper presents a VLSI architecture to combine the intra and inter prediction in the PPC module for H.264 video decoding. We provide the analysis to estimate the computational complexity under HDTV 1080 specification. The distributed memory access and shared adder-based method are used to reduce the hardware cost and computation. The exploiting data reuse in supporting variable block-size operations and the interpolator of mixed six-tap FIR filter are used to improve the performance. Our implement is capable of decoding one macroblock within 320 cycles in average. The HW cost is 60854 gates. As well as the max speed archive the 150MHz in the proposed design under TSMC 0.18um CMOS 1P6M.

REFERENCES

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, May 2003. Joint Video Team.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on CSVT, July 2003.

[3] 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," Circuits and Systems for Video Technology, IEEE Transactions on, March 2005 pp. 378-401.

[4] S. Z. Wang, T. A. Lin, T. M. Liu and C. Y. Lee, "A new motion compensation design for H.264/AVC decoder," Proc. ISCAS 2005.

[5] Moshe Y., Peleg N., "Implementations of H.264/AVC baseline decoder on different digital signal processors," ELMAR, 2005. 47th International Symposium, 8-10 June 2005 pp. 37 – 40.

[6] V. Lappalainen, A. Hallapuro, T. D. Hamalainen, "Complexity of Optimized H.26L Video Decoder Implementation," IEEE Trans. Circ. And Syst. for Video Technol. vol. 13, pp. 717-725, July 2003.

[7] X. Zhou, E. Q. Li, Y. K. Chen, "Implementation of H.264 Decoder on General-Purpose Processors with Medio Instructions," SPIE Conf. on Image and Video Comm. And Process., pp. 224-235, May 2003

[8] Joint Video Team (JVT) reference software JM8.2a.

[9] J. M. Boyce, "Weighted prediction in the H.264/MPEG AVC video coding standard," Proc. ISCAS 2004.

[10] T. W. Chen, Y. W. Huang, T. C. Chen, Y. H. Chen, C. Y Tsai, L. G. Chen, "Architecture Design of H.264/AVC Decoder with Hybrid Task Pipelining for High Definition Videos," Proc. ISCAS 2005.

Table 1: Implementation results of the proposed design under TSMC 0.18um CMOS 1P6M

PPC Components	Gate Count	Memory (bit)
Intra Prediction	12785	~0.7 K
Inter Prediction	41162	6.5 K
Weighted Prediction	4152	1.5 K
PPC Controller	2758	No
Total	60854	8.7 Kb

Table 2: Comparison of the proposed design with other interpolation architectures

	ISCAS'2005 [4]	**Proposed Design**
MVP	Incomplete MVP	Software
Needed Interpolator Components	Horizontal x 9 + Vertical x 4 + Bilinear	Horizontal x 4 + Vertical x 8
Interpolator Gate Count	20686 (0.18 um)	**15000 (0.18um)**
Interpolation Execute time	560 cycles/MB	**320 cycles/MB**
Inter Prediction Total Gate Count	43k	40k
Required Working Frequency for HD 1080 Video	100MHz	**87MHz**