A Flexible and High-Performance Hardware Video Encoder Architecture

Kaijin Wei, Shanghang Zhang, Huizhu Jia, Don Xie, and Wen Gao, *Fellow IEEE*
National Engineering Laboratory for Video Technology
Peking University
Beijing, China

***Abstract*—This paper presents a new video encoder architecture for H.264 and AVS, which adopts a novel macroblock (MB) encoding order. As a replacement of Level C+ zigzag coding order, the so-called Level C+ slash scan coding order with NOP insertion is used as MB scheduling to remove MB-level data dependency of the pipeline so that the left MB's coded results such as motion vector (MV) and reconstructed pixels can be obtained early in motion estimation (ME) stages. As a result, by sharing the reconstruction (REC) loop, sequential intra prediction (INTRA) can be split into multiple pipeline stages to explore more block-level parallelization and rate distortion optimization (RDO) based mode decision is apt to implement. The exact MV predictors (MVP) obtained in motion estimation can not only improve coding performance but also make pre-skip ME algorithm able to be applied into this architecture for low power applications. Since the proposed scheme is attributed to Level C+ data reuse, the bandwidth is decreased greatly. A real-time high-definition (HD) 1080P AVS encoder implementation on FPGA verification board with search range [-128, 128]×[-96, 96] and two reference frames at an operating frequency of 160 MHz validates the efficiency of proposed architecture .**

I. INTRODUCTION

Many complex video coding standards have been developed in recent years such as H.264/AVC, Audio Video Coding Standard (AVS). Software encoders cannot usually achieve real-time processing performance for high resolutions. A hardware implementation is preferred. However, for inter and intra predictions, current MB needs the MV and reconstructed pixels of left and top coded MBs. With normal raster coding order, the data dependency of left MB makes it inconvenient for parallel processing and MB pipelining.

Much simplification and modification [1] [2] [3] [4] are adopted to avoid the data dependency of left MB while the coding performance inevitably degrades. Bandwidth is another problem. Conventional Level C data reuse is not sufficient for high resolutions [6]. According to our analysis, a new Level C+ slash coding order is proposed as a replacement of zigzag coding order in paper [6]. By inserting a few NOP operations into the pipeline to keep a single slash scan pattern for the whole frame, data dependency can be removed substantially. This scheme not only reduces the required bandwidth but also makes the pipeline more flexible and efficient than Level C+ zigzag coding scheme. Many algorithms such as fast skip can be directly mapped into this framework.

The remainder of this paper is organized as follows. Section II presents the MB-level data dependency and reviews some popular hardware encoder pipelines proposed in the literature. Section III proposes our Level C+ slash scan coding order and the corresponding flexible and high-performance pipeline architecture for various function requirements while Section IV presents an implementation of HD video encoder with this framework. Finally, Section V concludes this paper.

II. REVIEWS OF PREVIOUS PIPELINING SOLUTIONS TO DATA DEPENDENCY IN H.264

A. MVP dependency for RD optimized ME in H.264

In order to improve the coding performance, RD optimized method is applied in ME to check each MV candidates by considering the consumed bits for them. MV predictor (MVP) for current block is referenced to calculate MV difference. The standard MVP is the best choice for the RD performance. However, the standard MVP is dependent on the MV of left and top MBs after mode decision as Fig. 1(a). In raster coding order, the left MB's MV is difficult to obtain in general pipelines. Because integer ME (IME) usually performs one stage earlier than fractional ME (FME), the left MB's MV is harder for IME to get. In paper [1], four-stage pipeline is proposed, as Fig. 2. The left MB's MV is unavailable for both IME and FME. A simplified MVP algorithm for IME was thus proposed there where the medium of MVs from top-left, top and top-right MB is applied to all of the 41 blocks in current MB [1]. For FME, MVP algorithm adopts the left incomplete MV (before intra/inter mode decision). The same simplified MVP algorithm is used in [2] for IME. Obviously, the coding performance degrades by 0.1-0.2dB due to the inaccurate MVP.

B. Reconstructed pixels dependency for intra prediction in H.264

For intra coding, the needed reconstructed pixels from left and top neighboring blocks result in MB-level and block-level data dependency as in Fig. 1(b). It needs to calculate nine 4x4 and four 16x16 intra modes and the 4x4 blocks should sequentially perform reconstruction loop. This intra dependency seriously influences MB pipelining. In paper [1], INTRA and DPCM (DCT/Q/IQ/IDCT) are installed in the same stage. The sequential intra prediction results in long computation path. Significant logic resource is consumed to achieve real-time processing. In [3], intra prediction is split into two parts and crosses two stages. The first part performs intra prediction and intra mode decision using original pixels while the second part performs prediction with reconstructed pixels for the selected best mode. However, sometimes this modification causes severe quality degradation especially for the sequences which needs many intra MBs, like an action movie in which motion estimation often fails to find a good match [1].

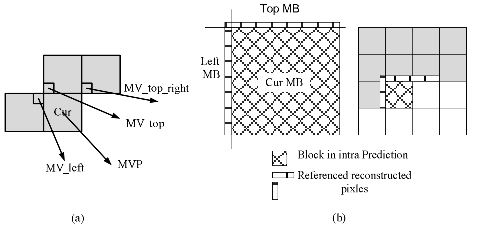

Figure 1. (a)MVP dependency in MB level. (b)MB-level and block-level reconstructed pixels dependency for intra prediction.

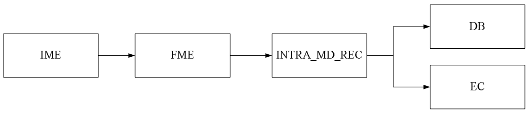

Figure 2. The conventional four-stage pipeline.

C. *MVP for skip mode/fast skip in H.264 in low power applications*

Simplified MVP used in ME is just to decide the best motion candidates, while for the skip mode, standard MVP must be used. Because the left MB's MV is not available in IME in general pipeline structures, skip mode has not been mentioned in most papers at all. But skip mode can increase coding performance and usually works as the fast algorithm in low power applications. Checking the skip mode before IME or FME can save much computing power. In paper [5], modified fast skip mode is produced from the 16x16 mode after IME to reduce power by checking the resulted MV and the residual value. If skip mode is selected, FME and INTRA will both be skipped. FME, INTRA and mode decision (MD) are together located in the same pipeline stage [7] so that the left MB's MV can be obtained to calculate the fast MVP skip mode for current MB. The disadvantage of this structure is the long processing chain. Paper [4] proposed a ME pre-skip algorithm which can skip both IME and FME. Hardware-software co-design is utilized to implement a dynamic reconfigurable pipeline structure. FME can be placed either in the same stage as IME or in the subsequent stage. Sophisticated task scheduling is performed to control the two level MB pipeline. If skip MV before ME is tested true, ME engines are turned off and inter mode decision is directly preformed. Nevertheless, this paper did not mention how to obtain the standard MVP in IME at all.

D. *Bandwidth problem*

For low resolution pictures, adopting Level C data reuse scheme to reuse the horizontal overlapped search window is sufficient. For HD video, a larger search window is required to cover the possible motion range. Therefore real-time HD encoder will consumes much more bandwidth for IME than the low definition image. If with multiple reference frames, the bandwidth may become a bottleneck especially in low power applications. In paper [6], Level C+ data reuse scheme and corresponding zigzag coding order were proposed to reduce the required external memory bandwidth. It can reuse two-dimensional overlapped search window among adjacent MBs. For a given HFmVn [6] zigzag scheme in which parameter m indicates the width of the stable zigzag line and n indicates its height, the bandwidth is decided by n and vertical search range (SRV) while the buffer size is decided by m, n, horizontal search range (SRH) and SRV. A larger n means that more bandwidth will be saved while more on-chip memory is consumed. For tradeoff between bandwidth and RAM, it is appropriate for general pipeline structure if n is given by 2 or 3. A HF5V3 zigzag stitch scheme can be used for the conventional pipeline in Fig. 2. Therefore, one picture will be split into many stitches as illustrated in Fig. 3. And the bottom stitch may be three-MB-row, two-MB-row or even one-MB-row according to the picture height. The coding order in one HF5V3 stitch is illustrated in Fig. 4. In fact, it consists of 3 kinds of slash patterns that are slash HF5V3 (the stable zigzag line), slash HF3V2 and slash HF1V1 (the raster order).

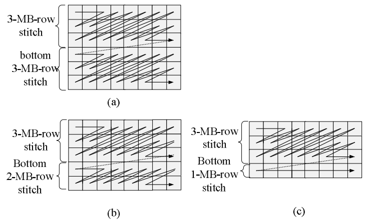

Figure 3. (a)Bottom stitch is 3 MB height. (b)Bottom stitch is 2 MB height. (c)Bottom stitch is 1 MB height.

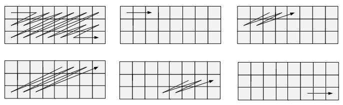

Figure 4. One HF5V3 zigzag stitch is composed of slash pattern HF5V3, HF3V2 and HF1V1.

III. OUR PROPOSED SLASH SCAN CODING SCHEME

A. *The problem of zigzag coding order*

Slash scan pattern analysis in Fig. 4 reveals that a HF5V3 zigzag stitch is a combination of three slash patterns. For the common pipeline structure in Fig. 2, in slash pattern HF5V3, the constraint towards IME and FME imposed by MVP and the constraint towards reconstructed pixels imposed by INTRA can be alleviated substantially. It is because the left MB is encoded 3 MB periods earlier than current MB. Consequently, the left MB's coded results such as MV and reconstructed pixels can be got in time even in IME stage. However, in slash pattern HF3V2, the standard MVP can be obtained only in FME, not in IME, and in slash pattern HF1V1, the standard MVP can not be obtained in either IME or FME and the constraint on the left MB for intra prediction is still strict. Nevertheless, the undesired slash HF3V2 and slash HF1V1 merely appear in the following three exceptional cases. The first case is in the initialization of the pipeline at the beginning of an image; the second is during the switch between two stitches, from the end of one stitch to the beginning of the next stitch. The third case is for the last stitch, when the MB number in vertical direction of the picture can not be divided exactly by 3, the bottom stitch may be 2-MB-row or 1-MB-row. Though this zigzag coding order can reduce memory bandwidth, varied slash patterns makes no improvement to the pipeline dependency in those three cases.

B. *Our 3-MB-row scheme with slash pattern HF5V3*

In order to remove the MB-level data dependency fundamentally, an improved slash scan coding order is proposed to deal with the preceding three exceptional cases, making the whole image consists of just one slash pattern HF5V3. As illustrated in Fig. 5, for the first pipeline initialization case and the third case that bottom stitch is not 3-MB-row, NOP operations are inserted to meet the requirement of data constraint. Six NOP operations are added at the beginning of the frame. If the bottom stitch has only 2 rows of MB such as 1080p (1920x1080), one row of NOP is appended to this stitch while if the bottom stitch has only one MB row like a filed of 1080i (1920x540), 2 rows of NOP are added. For the second switching condition, interlaced encoding order is preferred compared to respectively encoding between two stitches as in [6]. By all these adjustments, the data dependency constraints can be all satisfied for MVP and INTRA, making the pipeline concise and MB scheduling regular. The added NOP operations yield no more than 6% bubbles increment in the worst case.

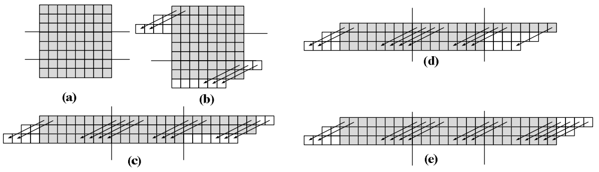

Figure 5. (a)Picture is split into stitches. (b)Add NOP operations in the beginning and the end. (c)Regular slash coding order with bubbles when bottom stitch is 2 MB rows. (d)Slash coding order when bottom stitch is 1 MB row. (e)Slash coding order when bottom stitch is 3 MB rows.

C. *Flexible pipeline structure with slash HF5V3 scheme*

Although NOP operations will cause negligible pipelining performance degradation, the advantage is obvious that the MB-level data dependency can be substantially removed and more parallelization can be explored. With 3-MB-row slash scheme, the left MB is encoded 3 MB periods before the current MB, so the left MB's coded results such as MV and reconstructed pixels can be got in time in the IME stage.

With the available reconstructed pixels of left MB in IME stage, INTRA will have 3 MB periods' time to perform. Consequently, by sharing DPCM as illustrated in Fig. 6(a), INTRA can be at most pipelined into 3 stages to alleviate reliance between intra blocks in a MB to increase processing performance. Thus, complicated and sequential intra coding is endurable. In addition, with INTRA partitioned from the third stage, RDO based mode decision is apt to implement in this stage as Fig. 6(b).

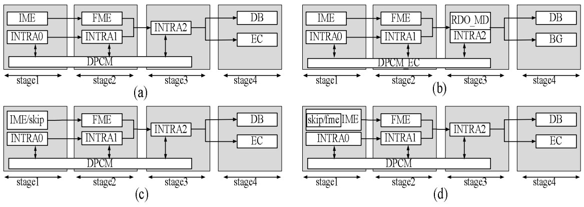

Figure 6. Extended pipeline with slash HF5V3. (a) INTRA can be at most separated into 3 stages by sharing DPCM. (b)RDO based mode decision. (c)Skip mode support in IME stage. (d)Fast skip in IME.

Because standard MVP can be obtained in IME and FME, exact R-D optimized Lagrange cost calculation for IME and FME is possible, which can improve coding efficiency by 0.1-0.2dB. Furthermore, many algorithms can be mapped into this framework without too much effort. For example, skip mode can be supported by this slash coding framework as in Fig .6(c). Like normal modes, this skip mode can be processed in the way that the skip block is obtained first in IME by standard MVP, afterwards necessary interpolations will be done in FME and at last mode decision is performed in MD. For low power application, fast skip early termination algorithm can be easily installed in IME as in Fig. 6(d). However, a FME module should be equipped in IME in case that the MVP falls in the sub-pixel position.

D. *Bandwidth and RAM resource considerations*

In Level C data reuse scheme, after coding one MB, one MB column of search window, about (2×SRV+1) MBs, will be loaded from external memory. While in Level C+ slash HF5V3, among coding three MB, one MB column of buffer window, about (2 × SRV+3) MBs, will be loaded from external memory. During each MB period, only (2 × SRV+3)/3 MBs need to be loaded. The bandwidth is reduced to (2 × SRV+3)/(3 × (2 × SRV+1)) of the Level C style. Because the buffer window is enlarged by 4 MBs in horizontal direction and 2 MBs in vertical direction, the RAM resource becomes (2 × SRV+3)(2 × SRH+5) from (2 × SRV+1)(2 × SRH+1) of Level C. If with a larger search window, the total buffer may become unacceptable because of the two-dimensional expanding. For the tradeoff between RAM resource and bandwidth, 2-MB-row stitched scheme is considerable. With this 2-MB-row scheme, bandwidth turns into (2×SRV+2)/(2×(2×SRV+1)) of Level C while the on-chip RAM is reduced to (2×SRV+2)(2×SRH+3) from that of 3-MB-row stitched scheme. Slash scan coding order with NOP insertion is still used as MB scheduling and the single slash pattern HF3V2 is adopted for the whole picture. However, for HF3V2, the left MB is coded two MB periods before current MB. So for the pipeline in Fig. 6, IME can not get the standard MVP in time while FME can still obtain it. Therefore, skip mode can be at most implemented in FME stage. INTRA can be at most expanded into two pipeline stages. In a word, adopting 2-MB-row stitches degrades the pipeline's flexibility while more RAM resource can be saved.

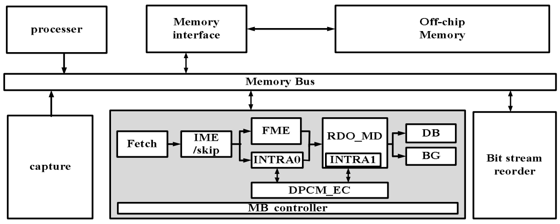

Figure 7. System architecture of AVS HD encoder.

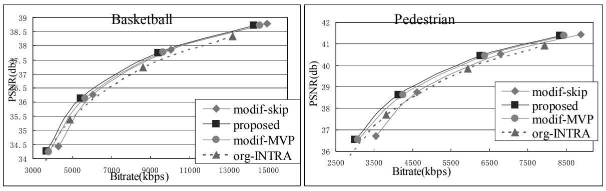

Figure 8. The rate distortion curves for two 1080p test sequences.

IV. Simulation and Implementation

The proposed architecture is verified by implementing a high-performance AVS HD 1080p encoder, as AVS has the same coding structure used in H.264. This encoder adopts a large search window by [-128, 128]×[-96, 96] and supports two reference frames. Level C+ HF5V3 slash coding order is employed as MB scheduling. The same Level C+ scheme is adopted for chroma compensation. The system architecture is illustrated as Fig. 7. Integer motion estimation is performed with the standard MVP and skip mode is supported in IME stage instead of being simplified from 16x16 mode [5]. Approximate RDO based mode decision is implemented in the fourth stage. For I frame only four luminance 8x8 blocks are performed with RDO and for P frame only the selected intra mode and the selected three inter modes including total 24 8x8 blocks are processed with RDO to reduce the throughput. INTRA is just expanded into adequate 2 stages by sharing DPCM and entropy coding (EC). Two luminance 8x8 blocks and two chroma 8x8 blocks are processed in first stage while the other two luminance 8x8 blocks are processed in second stage. The test conditions are IPPP and two reference frame for 90 frames of 1080p. The performance testing results are illustrated in Fig. 8 and in Table I. "Modif-skip" means that the encoder adopts a modified skip mode generated from the 16x16 mode like paper [5]. "Modif-MVP" means the MVP for IME is simplified like in general level C pipeline. "Org-INTRA" means that the intra prediction and mode decision are performed with original pixels instead of reconstructed pixels. No more than 1.5% bubbles inserted in the pipeline for HD 1080p can almost be ignored in contrast to this high performance of the proposed architecture. This hardware encoder system is implemented on Xilinx FPGA V6 verification platform with one 64-bit width SDRAM controller to achieve a real-time compression of 1080P at 30f/s. The frequency of system memory can be reduced to 160 MHz to meet our bandwidth requirement due to Level C+ slash HF5V3 data reuse. The same as using Level C+ zigzag coding order [6], our system using the proposed slash scheme reduces the bandwidth for reference data by 62% compared to Level C. Though the on-chip buffer is increased by about 42%, a Level C+ slash HF3V2 as mentioned above can be used to make this increment optimized to 20%.

TABLE I. TESTING RESULTS OF 1080P

Sequence	Proposed VS Midif-MVP		Proposed VS Midif-skip		Proposed VS Org-INTRA	
	PSNR Gain (dB)	*Bit-rate change (%)*	*PSNR Gain (dB)*	*Bit-rate change (%)*	*PSNR Gain (dB)*	*Bit-rate change (%)*
Basketball	0.11	-2.8	0.18	-5.2	0.32	-11.2
Pedestrian	0.05	-1.4	0.17	-5.1	0.35	-9.2
Tractor	0.2	-4.2	0.2	-3.9	0.03	-0.4
Old_town	0.02	-0.7	0.18	-10.3	0.11	-6.5
Mobcal_ter	0.05	-1.7	0.28	-9.2	-0.1	2.8
Average	0.086	-2.16	0.202	-6.47	0.142	-4.9

V. Conclusion

This paper presents a new flexible and high-performance hardware encoder architecture for H.264 and AVS, which adopts an improved Level C+ slash coding order with NOP insertion to solve MB-level data dependency. As a replacement of Level C+ zigzag coding order, it not only reduces the required bandwidth but also leads to highly efficient MB pipelining and simple MB scheduling. By sharing the DPCM, INTRA can be pipelined into multiple stages to explore more block-level parallelization. Moreover, the standard MVP obtained in ME can improve coding performance. Similarly, fast skip algorithm can also be easily and directly mapped into this framework in low power applications. A real-time HD 1080P AVS encoder implementation with search range [-128, 128]×[-96, 96] and two reference frames at an frequency of 160 MHz validates the efficiency of the proposed architecture.

References

[1] Tung-Chien Chen, Yu-Wen Huang, and Liang-Gee Chen, "Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture", IEEE ICASS, Hong Kong, China, April, 2004.

[2] T. C.Wang, Y.W. Huang H. C. Fang, and L. G. Chen, "Performance analysis of hardware oriented algorithm modifications in H.264," in Proc. of ICASSP, 2003.

[3] S. Mochizuki, T. Shibayama, M. Hase, et al., "A 64 mW high picture quality H.264/MPEG-4 video codec IP for HD mobile applications in 90 nm CMOS," IEEE Journal of Solid-State Circuits, vol. 43, no. 11, pp. 2354–2362, 2008.

[4] Yu-Han Chen, Tung-Chien Chen, Liang-Gee Chen, "Power-scalable algorithm and reconfigurable macro-block pipelining architecture of H.264 encoder for mobile application," IEEE International Conference on Multimedia and Expo, pp. 281 - 284, 2006.

[5] Hyun Kim, Chae-Eun Rhee, Jin-Sung Kim, Sunwoong Kim, Hyuk-Jae Lee, "Power-aware design with various algorithms for an H.264/AVC encoder," ISCAS 2011: 571-574

[6] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, and L.-G. Chen, "Level C+ data reuse scheme for motion estimation with corresponding coding orders," IEEE Trans. Circuits Syst. Video Technol., vol. 16, pp. 553–558, Apr.2006.

[7] Zhenyu Liu, Yang Song, Ming Shao, Shen Li, Lingfeng Li, Shunichi Ishiwata, Masaki Nakagawa, Satoshi Goto, and Takeshi Ikenaga, "HDTV1080p H.264/AVC Encoder Chip Design and Performance Analysis", IEEE Journal of Solid-State Circuits, Vol. 44, No. 2, pp 594-608, Feb. 2009.