ANALYSIS AND DESIGN OF MACROBLOCK PIPELINING FOR H.264/AVC VLSI ARCHITECTURE

Tung-Chien Chen, Yu-Wen Huang, and Liang-Gee Chen

DSP/IC Design Lab., Graduate Institute of Electronics Engineering and
Department of Electrical Engineering, National Taiwan University
{djchen, yuwen, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

This paper presents a new macroblock (MB) pipelining scheme for H.264/AVC encoder. Conventional video encoders adopt two-stage MB pipelines, which are not suitable for H.264/AVC due to the long encoding path, sequential procedure, and large bandwidth requirement. According to our analysis of encoding process, an H.264/AVC accelerator is divided into five major functional blocks with four-stage MB pipelines to highly increase the processing capability and hardware utilization. By adopting shared memories between adjacent pipelines with sophisticated task scheduling, 55% of the bus bandwidth can be further reduced. Besides, hardware-oriented algorithms are proposed without loss of video quality to remove data dependencies that prevent parallel processing and MB pipelining. The H.264/AVC Baseline Profile Level Three encoder, which requires computational complexity of 1.8 tera-instructions per second (TIPS), is successfully mapped into hardware with our MB pipeline scheme at 100 MHz.

1. INTRODUCTION

The new video coding standard, H.264/AVC, developed by Joint Video Team (JVT) significantly outperforms previous standards in compression due to the new features including motion estimation (ME) with variable block sizes and multiple reference frames, intra prediction, CAVLC, CABAC, in-loop deblocking filter and more [1][2]. Compared with MPEG-4 Simple Profile, up to 50% bitrate reduction is achieved with more than four times of computational complexity. Therefore, hardware acceleration is a must for real-time applications. However, the reference software [3] adopts sequential processing of each block in the MB and creates data dependencies that prevent parallel processing.

We have addressed these difficulties [4] and proposed hardware architectures with modified algorithms[5][6][7]. However, it is still challenging to integrate each module into a complete encoder. The traditional two-stage MB pipelines, prediction (ME) and block engine (BE=MC+DCT+Q+IQ+IDCT+VLC), cannot be applied to H.264/AVC anymore because of the long critical path (in unit of cycles) and feedback loop. According to our analysis, five major functions are extracted and mapped into four MB pipeline stages with hardware-oriented algorithms to enable parallelization and MB pipelining. The bandwidth is also reduced by utilizing shared memories between adjacent pipelines with sophisticated task scheduling.

The rest of this paper is organized as follows. In Section 2, we describe the traditional pipeline schedule and clarify the problems for H.264/AVC when adopted. Section 3 proposes the analysis and decomposition of encoding process, as well as our MB pipeline scheme and schedule. Afterward, in Section 4, modified algorithms are applied to enable parallel processing and MB pipelining. Finally, Section 5 shows the coding performance, and Section 6 gives a conclusion.

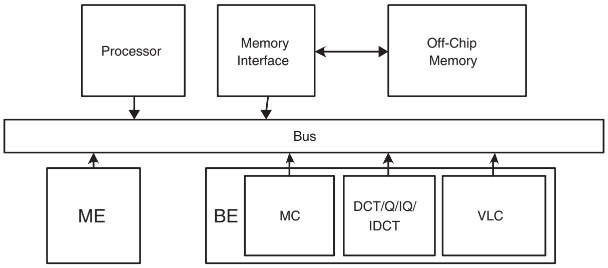


Fig. 1. Traditional platform-based system with hardware accelerators.

2. PROBLEM STATEMENT

Our design target is an encoding system for H.264/AVC Baseline Profile Level Three with ±64 horizontal and ±32 vertical search range and four reference frames. A typical platform-based system with hardware accelerators to achieve traditional video coding functionalities is depicted in Fig. 1. The processor handles MB-level hardware controlling and other high level procedures. Accelerators such as ME, MC, DPCM loop (DCT/Q/IQ/IDCT), and VLC are connected to the bus to speed up the coding operations. To reduce the bus traffic, local memories are shared by MC, DPCM loop, and VLC modules. The ME module occupies the dominant percentage of computation and bandwidth. As for scheduling, ME module processes each MB in raster order. After ME finishes a MB, other accelerators take over the encoding of this MB while ME keeps on processing the next MB simultaneously.

The two-stage MB pipelining works well for traditional video encoders. However, many problems are encountered for hardware implementation of H.264 with two-stage MB pipelines. First, the entire path of the encoding process includes inter/intra prediction/compensation, entropy coding, and deblocking. If two-stage MB pipelining is applied, the critical path will be too long, and the utilization of accelerators will be significantly decreased. Second, the bandwidth requirement increases abruptly due to multiple reference frames and deblocking. Separated accelerators with

0-7803-8251-X/04/$17.00 ©2004 IEEE

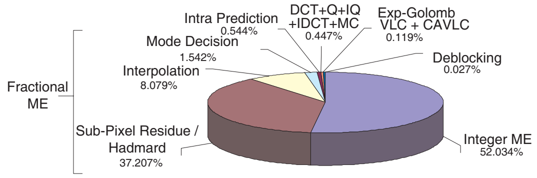


Fig. 2. Run time analysis of encoding of H.264/AVC

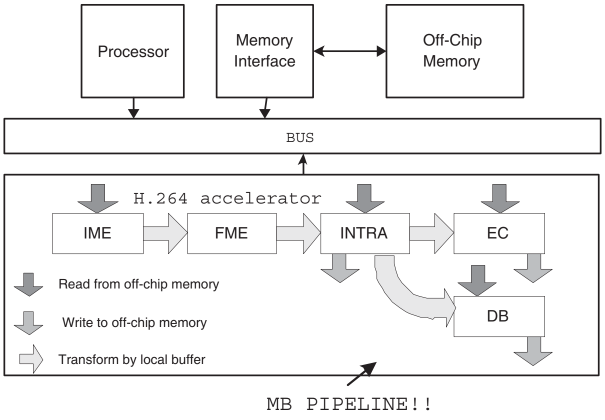


Fig. 3. H.264 MB engine with pipeline scheme for the platform-based system.

communication via bus increase the traffic even more. Third, the Lagrangian mode decision in reference software adopts sequential processing of each block in MB, which can improve the coding performance. Nevertheless, it causes data dependencies between neighboring blocks that make parallel processing and MB pipelining a tough job.

3. PROPOSED MB PIPELINE SCHEME IN H.264

3.1. Decomposition of Coding Process and Pipelining

The prediction procedure, in which only ME was involved before, includes integer ME (IME), fractional ME (FME), and intra prediction (INTRA). The computational complexities are shown by run time analysis as Fig. 2. IME stage has the highest computation complexity in the encoding system. It must be highly parallelized to support the functions of variable block sizes and the multiple reference frames and to meet the real-time requirement. Unlike traditional mode decision algorithm, the Lagrangian mode decision is done after FME for all kinds of block sizes and reference frames, which improves the PSNR by 0.5-2.0 dB. However, a large increase in computational complexity and complex sequential flow are demanded. Hence, FME is separated from IME as one pipeline stage. Intra prediction, which uses the reconstructed data of current frame as predictors, must be integrated with DPCM loop. Although, the computational complexities of intra prediction and reconstruction procedure are not high, data dependency between sub-blocks results in unavoidable sequential flow. If we arrange intra prediction together with fractional ME, the critical path of this MB pipeline stage will be too long (FME, inter mode decision, intra prediction, intra mode decision, intra/inter mode selection, and DPCM loop). Therefore, intra prediction with DPCM loop should be separated as the third stage, INTRA. Deblocking, which requires considerable bandwidth, uses reconstructed data after DPCM loop. Entropy coding, which requires sequential bit-level operations, encodes data after mode decision and residue generation. Because of the sequential procedure and unique features that are difficult to achieve resource sharing with other accelerator, deblocking (DB) and entropy coding (EC) are proposed as the fourth stage to get higher utilization of hardware.

In sum, as shown in Fig. 3, five main functions including integer motion estimation (IME), fractional motion estimation (FME), intra prediction (INTRA), entropy coding (EC), and deblocking (DB) are exacted from encoding process. Each function is mapped to different architectures with individual features, and MB-pipelined to improve hardware utilization.

3.2. Integration of Other Functions

In addition to the five main functions, there are still some other modules needed in the H.264/AVC encoding system. They must be carefully arranged in each stage to adapt to the MB pipeline scheme discussed above. Luma motion compensation (MC) is placed in FME stage for reuse of search window buffer and interpolation circuits. These compensated data are transmitted to INTRA stage for generation of residues after intra/inter mode selection. As for the chroma MC, it is arranged in the INTRA stage. Because inter mode decision does not consider the chroma distortion (low complexity mode decision), we get reference chroma pixels from DRAM after intra/inter mode selection to reduce the local buffer size and bus bandwidth. DPCM loop is integrated with INTRA to get the correct reconstructed data needed for prediction, which have been discussed before.

3.3. Bandwidth Consideration

The required bandwidth in H.264 encoding system with MB pipeline scheme is much larger than before. Due to multiple reference frames ME, several times of traffic from of-chip memory are demanded by IME for search window loading and by FME for interpolation of fractional pixels. Reconstructed frames in DRAM must also be loaded to DB and written back after processed. Mode decision and entropy coding have data dependencies on previous MB's. Necessary information such as best modes, motion vectors (MV's), and so on, must be received through bus, which contributes considerable bandwidth as well. The total requirement is 620 MByte/sec.

We integrate all accelerators into one MB engine and place shared buffers between adjacent pipelines. Reusable data between each stage are shared or transmitted locally. For example, search window buffer are used by both IME and FME with a rotation model, and the reconstructed MB without DB is transmitted directly from INTRA to DB stage. The coding information and reconstructed pixels of the previous (left) MB are locally buffered for next MB procedure, and those of upper MB's are transmitted via bus. By these techniques, bandwidth is reduced from 620 MBytes/sec to 280 MBytes/sec.

3.4. Scheduling of MB Pipelines

When a frame starts, a short initialization is required, and then the H.264/AVC MB engine processes MB's in raster order with the

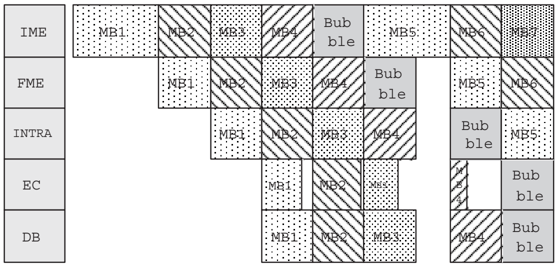
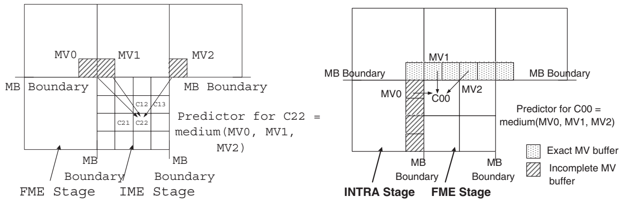


Fig. 4. MB model(schedule) of MB pipeline (assume frame width is four MB length).



(a) (b)

Fig. 5. (a)Global modified MV predictors used in IME stage. (b)Incomplete MV predictors used in FME stage.

previously mentioned MB pipelines like Fig. 4. While MB4 is in IME stage, MB3 is in FME stage, MB2 is predicted by INTRA, and MB1 is processed by EC and DB. Under this MB pipelining model, we should pay attention to the following points. First, when first MB of a row is processed in IME stage, it takes more time to reload the whole search window buffer, while others need less time because we can reuse the overlapped search area data that are already in search window buffer. Second, each stage is designed to have about the same operation time with different degrees of parallelism except for the EC module whose operation time depends on the residues. Third, there will be one bubble for every MB row because IME and FME share the reference data in the same search window buffer. When FME processes the last MB in a MB row, search window buffer cannot be refreshed immediately by IME to process the first MB in next MB row. Therefore, a bubble is inserted to wait for FME. The bubble rate is 2.5% under our specification.

4. HARDWARE-ORIENTED ALGORITHM

Lagrangian mode decision improves coding performance but results in many block-level loops, which cause data dependencies between neighboring MB's and neighboring sub-blocks and prevent parallel processing and MB pipelining. Therefore, some hardware-oriented modifications are applied for hardware implementation with MB pipeline scheme. Our target here is to map H.264 sequential algorithm into the MB pipeline system with regular and smooth flow and to maintain the compression performance.

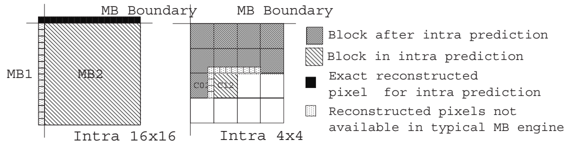

MB Boundary
MB Boundary
MB1
MB2
C01
C12
Intra 16x16
Intra 4x4
Block after intra prediction
Block in intra prediction
Exact reconstructed pixel for intra prediction
Reconstructed pixels not available in typical MB engine

Fig. 6. Reconstructed pixels for intra prediction.

4.1. IME Stage

The MV's of each block in MB are generally medium predicted by left, top, and top-right neighbor blocks. The cost function can be computed only after prediction modes of neighboring blocks are determined. Such dependency between neighboring blocks causes inevitable sequential processing, which conflicts with the required high parallelism in IME stage. Moreover, for a MB in IME stage, its left MB is still in FME stage, and both the prediction mode and the best MV's are not available. Therefore, the exact MV predictors (MVP's) are not available, too. To solve these problems, the exact MVP's are replaced by modified global MVP, which is the medium of MVs from top-left, top, and top-right MB and is applied to all of the 41 blocks in MB, as shown in Fig.5(a). For example, the exact MV cost of the C22 4x4 block is related to the MV's of C12, C13, and C21. During the IME phase, we change the MVP's of all 41 blocks to the medium of MV0, MV1, and MV2 in order to facilitate the parallel processing and MB pipeline scheme.

4.2. FME Stage

The sub-pixel ME refinement is performed around the best integer search position of 41 blocks at all reference frames. Inter mode decision will be finished according to the sum of absolute transformed differences (SATD) and the exact cost of side information in this stage. The compression performance here is more sensitive with accuracy of MVP's than in IME stage, and the modified global MVP will lead to considerable quality degradation. Fortunately, its computational complexity is not as high as IME stage. We can carefully design an architecture to implement the sequential flow as the reference software. However, a MB pipeline problem still exits. For a MB in FME stage, its left MB is in INTRA stage, and the MV's from left MB are incomplete (before intra/inter mode selection). For quality consideration and compatibility with MB pipeline scheme, exact left MV's are replaced by incomplete ones. As Fig.5(b) shows, exact MV0 is not available in FME stage and is replaced by its incomplete version to estimate the MVP of C00. Of course, the exact MVP's must be calculated in the sequential order defined by standard for EC stage after intra/inter mode selection in INTRA stage.

4.3. INTRA Stage

The intra prediction, which contains nine 4x4 and four 16x16 intra prediction modes, requires reconstructed pixels from left and top neighboring blocks. However, reconstructed data cannot be obtained until the neighboring blocks are transformed, quantized, inverse quantized, inverse transformed, and then reconstructed. In [4], original frame pixels instead of reconstructed pixels are used as predictors and an error term is added to compensate the estimation inaccuracy. However, sometimes this modification causes

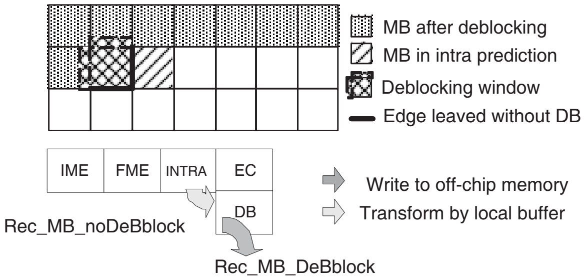


Fig. 7. DB window for MB pipeline scheme

severe quality degradation especially for sequence which needs many intra MB's, like an action movie in which motion estimation often fails to find a good match. In order to get high compression performance, we change the solution from algorithm domain to architecture domain. That means the block engine must be integrated with intra prediction. As shown in Fig.6, C02 should be reconstructed before C12 in intra prediction. MB1 should also be reconstructed right after mode decision so that MB2 can get the correct intra predictors. The methodology of transform and quantization in H.264 are both very simple and are suitable for hardware implementation [8]. Therefore, such integration should not cost much.

4.4. EC/DB Stage

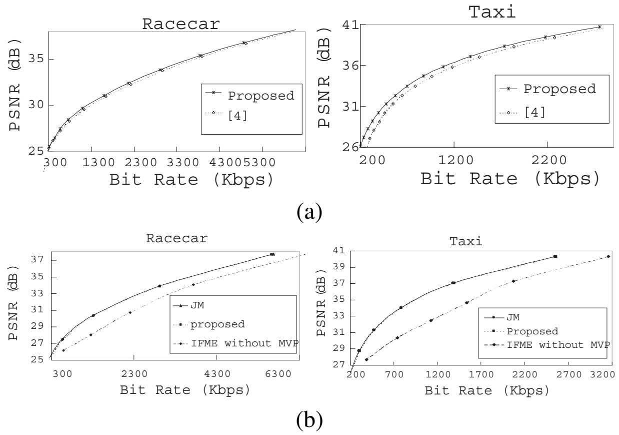
As for EC/DB stage, modification of algorithm will influence the consistence between encoder and decoder. Fortunately, MB pipeline problem caused by data dependencies with left MB does not exist. Parallelism is not the most important issue here because of the low computational complexity. However, sophisticated task scheduling is still demanded for DB to be done in parallel with other stages MB by MB in raster order. We arrange deblocking window as Fig 7 and leave procedure of the right and bottom edge to MB schedule afterward. We also reserve a frame-width-size space in off-chip memory for a row of reconstructed pixels without DB which are needed by intra prediction in MB's below. By these ways, DB can be processed right after INTRA without any MB delay, and the bandwidth is also reduced.

5. SIMULATION

The test conditions are IPPP..., one reference frame, CAVLC, low-complexity mode decision, ±64 horizontal search range, ±32 vertical search range, Hadamard transform opened, and (0,0) search range center. The sizes of two sequence, Racecar and Taxi, are 720x288x220 and 672x288x110 in 30Hz. Figure 8(a) shows the importance of integration between intra prediction and DPCM loop. Intra MB rate is 60% in Taxi and 30% in Racecar. Fig.8(b) shows the proposed algorithm to adopt modified global MVP's for IME and incomplete MVP's for FME, which results in almost no PSNR degradation.

6. CONCLUSION

We proposed an H.264 MB engine with novel MB pipeline scheme for platform-based system. Five main functions extracted from H.264 coding process are mapped into four MB pipeline stages. Hardware-oriented algorithms, which maintain the same coding performance compared with reference software, are proposed to remove the data dependencies that prevent parallel processing and MB pipelining. By transmitting reusable data through local buffer between adjacent stages, bandwidth is reduced from 620 to 280 MByte/Sec, and high utilization of accelerators is achieved. The scheduling of MB pipeline is very regular. H.264/AVC encoding process with computational complexity of 1.8 TIPS is successfully mapped in hardware with MB pipeline scheme at 100 MHz.



(a)

(b)

Fig. 8. The rate distortion curves of various standard sequences generated by JM7.3 and our proposed software.

7. 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] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on CSVT*, vol. 13, no. 7, pp. 560–576, July 2003.

[3] *Joint Video Team Reference Software JM7.3*, http://bs.hhi.de/ suehring/tml/download/, Aug. 2003.

[4] 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.

[5] T. C. Wang, Y. W. Huang H. C. Fang, and L. G. Chen, "Parallel 4x4 2D transform and inverse transform architecture for MPEG-4 AVC/H.264," in *Proc. of ISCAS*, 2003.

[6] Y. W. Huang, T. C. Wang, B. Y. Hsieh, and L. G. Chen, "Hardware architecture design for variable block size motion estimation estimation in MPEG-4 AVC/JVT/ITU-T H.264," in *Proc. of ISCAS*, 2003.

[7] Y. W. Huang, T. C. Wang, B. Y. Hsieh, T. C. Wang, T. H. Chang, and L. G. Chen, "Architecture design for deblocking filter in H.264/JVT/AVC," in *Proc. of ICME*, 2003.

[8] H. S. Malvar, A. Hallapuro, M. Karczewicz, and Louis Kerosfsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. on CSVT*, vol. 13, no. 7, pp. 598–603, July 2003.