FRAME-PARALLEL DESIGN STRATEGY FOR HIGH DEFINITION B-FRAME H.264/AVC ENCODER

Yi-Hau Chen, Tzu-Der Chuang, Yu-Han Chen, Chen-Han Tsai, and Liang-Gee Chen

DSP/IC Design Lab.,
Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan
Email: {ttchen, peterchuang, doliamo, phenom, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

High Definition (HD) H.264/AVC video compression is the emerging necessity on nowadays home entertainment environment and so on. However, Although B-frame coding scheme provides better quality, only P-frame encoders are presented due to too high complexity and memory requirement. In this paper, a frame-parallel encoding scheme based on B-frame's data independency is proposed. It can largely reduce the system memory bandwidth and improve the processing capability. Then, the proposed IME and FME scheduling can further enhance the hardware utilization for frame-parallel scheme. Finally, a case study is given to show that the proposed scheme can largely reduce 66% system bandwidth compared to direct implementation from previous P-frame encoder.

1. INTRODUCTION

With the trend of high quality digital video services, Joint Video Team (JVT) developed H.264/AVC as the next generation video coding standard in 2003. In H.264/AVC, many important coding tools are included to achieve great coding efficiency[1, 2]. However, the computation complexity and memory requirement of the whole video encoding system are also greatly increased especially for the high definition (HD) video specification. It deeply increases the difficulty to achieve the HD specification for a newest H.264/AVC encoder.

Previously, several works propose HD video encoders. Huang [3] firstly adopts four-stage MB-pipelined architecture to achieve HD 720p at 108MHz. Recently, Liu [4] proposes an HD 1080p encoder at 200 MHz with higher parallelism and system-in-silicon (SiS) DRAM. However, these works all target at H.264/AVC Baseline profile and only IPPP scheme is supported. Compared to B-frame coding structures such as IBPBP, IBBP and Hierarchical B-frame, IPPP scheme has 0.5 to 2 dB quality degradation [5]. But for B-frame schemes, computation complexity and memory requirement of motion estimation will be doubled due to forward and backward predictions. Modules in the previous designs [3, 4] should be redesigned to meet B-frame specification.

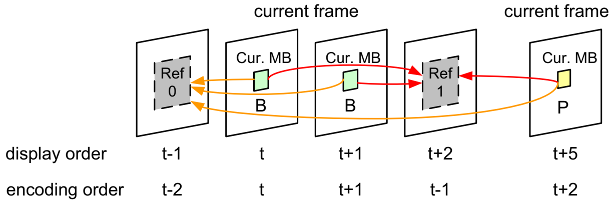


Fig. 1. Data dependency among IBBP coding scheme. Orange arcs represent forward prediction and red arcs represent backward prediction.

In this paper, we introduce a frame-parallel design strategy for an HD H.264/AVC encoder with B-frame coding. The proposed strategy can extend the computation parallelism in the frame-level by use of the data independency among B-frames. It can reduce system memory bandwidth which is always the biggest bottleneck for an HD H.264/AVC encoder chip design. Besides, the required operating frequency will not have apparent increase while supporting HD specification. The corresponding hardware architecture and scheduling of integer motion estimation (IME) and fractional motion estimation (FME) are also discussed. Then, a case study is used to evaluate the performance of proposed design strategy. This paper is organized as follows. In section 2, the main design issues of an HD H.264/AVC encoder will be introduced. The main concepts and examples of the proposed frame-parallel strategy and corresponding module design are presented in section 3 and 4, respectively. A case study will analyze the performance in section 5. Section 6 will conclude this paper.

2. EXISTING WORKS AND PROBLEM STATEMENT

The two HD H.264/AVC encoder designs[3, 4] mentioned in section 1 both adopt multi-stage pipelined architecture to balance their computation complexities between each pipeline stage. The required operating cycles for each stage will be similar and the hardware utilization will be higher so the op-

978-1-4244-1684-4/08/$25.00 ©2008 IEEE

erating frequency can be lowered. In these two designs, the average cycles per macroblock (MB) is only around 800 to 1000 cycles for various tasks in one stage. However, when B-frame is adopted, the available operating cycles of previous works will be insufficient.

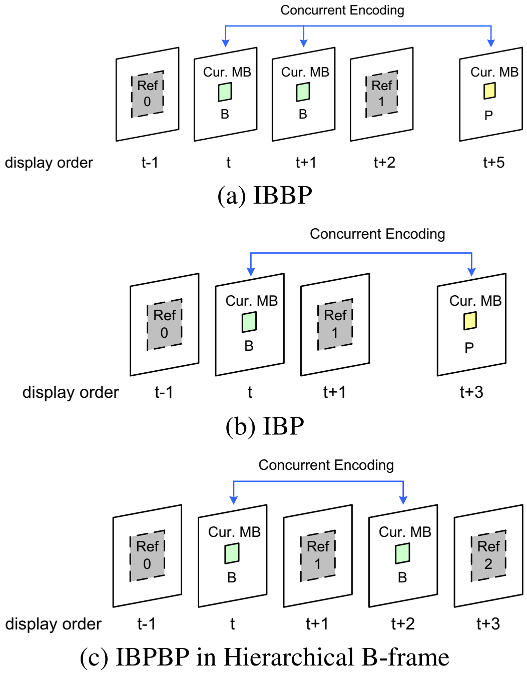
The main difference between B-frame and P-frame coding is the reference frame number. In general, the computation complexity and system memory bandwidth of IME and FME are directly proportional to their reference frame number. If our target specification is an HD 1080p encoder with B-frame schemes, the number of reference frames is twice compared to previous works [3, 4]. Therefore, the computation complexity between original multi-stage pipeline will become quite unbalanced due to doubled IME and FME computation. Cutting pipeline at original IME and FME stage will cause very large hardware cost due to buffering intermediate processed data. Besides, it also induces accessing confliction on internal searching range (SR) memory since the port of internal SRAM is always limited up to 2. Increasing parallelism of original module designs on candidate-level or higher level would be a better approach. If no more parallelism is applied, the encoder should be operated at a much higher frequency and the original processing unit design will be re-designed.

For HD encoder, the doubled system memory bandwidth of loading B-frame SR data could be more critical in system design. It not only increases the accessing frequency of external memory but also doubles the cycles of loading SR in IME stage in [3, 4]. The doubled cycles of loading SR would largely decrease the available cycles for IME data processing and result in higher parallelism requirement. For P-frame, the MB-level data reuse scheme has been maturely developed [6]. Therefore, we need a further parallelism scheme which can take care of computation complexity and system memory bandwidth simultaneously.

3. PROPOSED FRAME-PARALLEL ENCODING SCHEME

B-frames are pictures that are encoded by using both past and future pictures as reference frames. The prediction results are obtained by linear combinations of forward and backward prediction signals from motion compensation. By utilizing B-frame in the encoder design, the data dependency with the coding scheme of IBBP can be shown in Fig. 1 for one P picture and two B pictures. The encoding order is labeled from $t-2$ to $t+2$. At $t-2$ and $t-1$, two reference frames are encoded and reconstructed. Then the two B pictures and P picture can be encoded sequentially.

Since these three current frames has no data dependency among each other. It is possible to simultaneously encode the three current MBs which all locate in the same position. The proposed frame-parallel encoding schemes are shown in Fig. 2. As shown in Fig. 2 (a), the three current frames in Fig. 1 can now be encoded at the same time instead of coding these pictures in the frame by frame order. With proposed scheme, the loaded SR data on Ref_0 and Ref_1's gray region in Fig. 1 and 2 (a) can be easily shared by three encoded MBs to reduce the SR data bandwidth. Besides, since three MBs of different frames are processed simultaneously for all pipeline stage, the original modules in previous designs can be easily duplicated to triple the processing capability without modifying the detailed processing unit design. If the target operating frequency is the same as previous designs, the cycle number of each pipeline is now tripled and the problem of loading SR data mentioned in Sec. 2 can be solved.



(a) IBBP

(b) IBP

(c) IBPBP in Hierarchical B-frame

Fig. 2. Proposed frame-parallel encoding scheme.

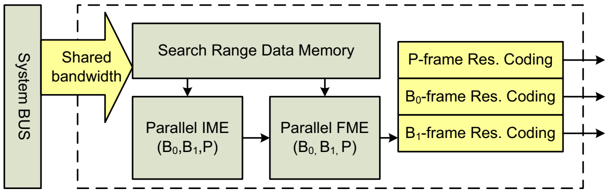


Fig. 3. Frame-parallel encoder architecture for IBBP scheme.

The proposed frame-parallel encoding scheme can be also applied on other B-frame schemes as shown in Fig. 2 (b) and (c) for IBP and IBPBP of Hierarchical B-frame schemes, respectively. The system memory bandwidth of loading SR data can be largely reduced for these schemes. Assume that the P-frame in Fig. 2 has two forward reference frames, by introducing frame-parallel scheme, the system bandwidth of schemes in Fig. 2 can be reduced by 66%, 50% and 25%, respec-

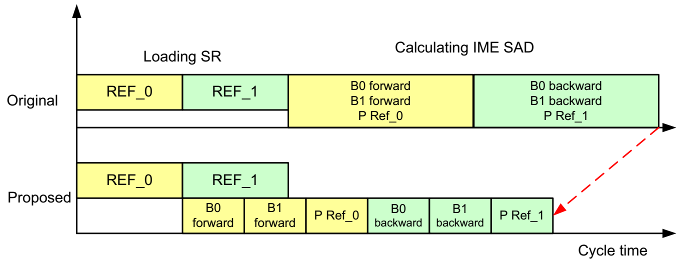


Fig. 4. IME scheduling for original and proposed frame-parallel encoder. The symbols are the same as Fig. 1.

tively. Please note that the proposed frame-parallel scheme can cooperate with other existing data reuse methods such as level C to further reduce system bandwidth.

4. PROPOSED ARCHITECTURES FOR FRAME-PARALLEL PROCESSING MODULES

In Sec. 3, the concept of frame-parallel scheme for B-frame coding is presented. The whole block diagram of proposed frame-parallel encoder can be depicted as Fig. 3. In Fig. 3, the IME and FME modules are shared by different frames' MBs and each frame has its own residual coding processor. In this section, the corresponding frame-parallel architectures for IME and FME will be introduced and the scheduling of accessing SR memory will be explained.

4.1. Integer Motion Estimation

In integer motion estimation stage, the total processing cycle time can be divided into loading SR data and sum-of-absolute-difference (SAD) calculating. Assume that the well-known Level C data reuse scheme [6] is adopted, the cycle time of loading SR data is decided by vertical searching range and system bus bitwidth, and the cycle time of SAD calculating is proportional to (# of search candidate)/(# of IME parallelism). Although the proposed frame-parallel scheme enables three current MBs to perform IME simultaneously, we recommend to process one current MB at a time as shown in Fig. 4 and increase the parallelism in candidate-level such as doubling the number of SAD tree. It is because that loading one reference frame's SR should be completed before accessing its data for SAD calculation. By the proposed schedule in Fig. 4, the cycle time of IME stage will be shorter and the control unit will be simpler. Besides, this approach can still work while any fast or predictive IME search algorithm is applied.

4.2. Fractional Motion Estimation

In FME module, since the three current MBs may have different integer motion vectors as their starting points for refinement, it causes confliction during accessing SR memory data and three current MBs cannot perform FME simultaneously. Besides, the computation complexity of B-frame equals to P-frame of three reference frames if one iteration refinement of B-frame is included. Based on above reasons, we propose an interleaved FME architecture and scheduling not only to improve the utilization of FME processing units but also to support B-frame iteration refinement.

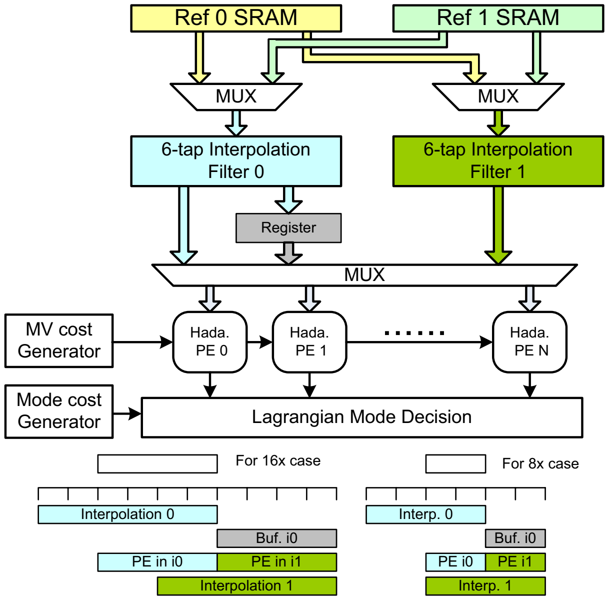


Fig. 5. The proposed FME architecture with interleaved PE scheduling. The lower part represent the interpolation and transform schedule for vertical 8 and 16 pixel, respectively. Hada. represents Hadamard transform.

In [3, 7], although the interpolation unit can reach 100% utilization, the Hadamard transform processing elements (PE) has only 64% utilization and the latter one limits the processing capability of FME. Therefore, for B-frame coding, we propose a fully-utilized FME architecture with interleaved PE scheduling as shown in Fig. 5, and the whole FME schedule for IBBP scheme is shown in Fig. 6. In Fig. 5 and 6, the blue block represents the data is from interpolation filter 0, the deep-green block represents the data is from interpolation filter 1. The data of gray blocks are buffered for iteration refinement with data from interpolation filter 1. The two interpolation filters in Fig. 5 are responsible for different reference frames in the same time, and the loading schedule are interleaved so that the Hadamard transform PE can be fully utilized. The proposed scheduling in Fig. 6 also avoid the confliction of accessing SR memory. With proposed schedule, the total cycle of FME stage is four times compared to original FME for P-frame in [7]. Since the cycles for each stage is only tripled, it is recommended to increase FME processing parallelism from 4 pixels to 8 pixels as in [4] or adopt the

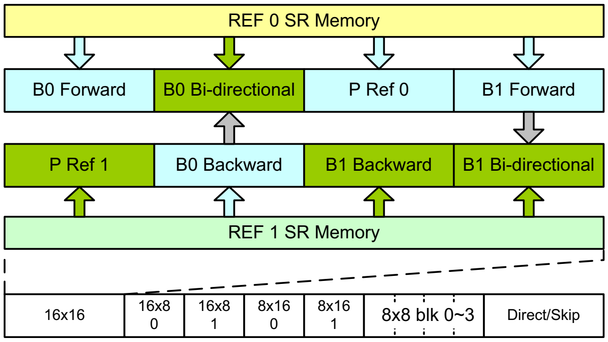


Fig. 6. The FME scheduling of SR memory access and processed operations for proposed frame-parallel encoder. The symbols of frames and prediction directions are the same as Fig. 1.

one-pass algorithm in [8] to further improve the processing capability.

5. CASE STUDY

In this section, we take [3] as original design and set a target specification of HD 720p with IBBP coding scheme for case study. For simplicity, only performances of IME and FME are evaluated. Based on [3], 32-bit bus for loading SR data is used and Max. SR size is 128×64. For IME, 64×32 search range is estimated and 8 candidates are calculated per cycle. Assume that FME operation for P-frame requires 1000 cycles and the P-frame in IBBP scheme has two reference frames. The results of performance evaluation are listed in Table 1. The "Benchmark" is direct implementation from [3]. Please note that the parallelism for benchmark and proposed scheme are the same for IME and FME. As shown in Table 1, the proposed scheme can largely reduce 66% system bandwidth and the required operating frequency is much lower than that of benchmark. Thus, the proposed architecture can be easily developed from existing works.

6. CONCLUSION

A frame-parallel design strategy for HD H.264/AVC encoder for B-frame is presented in this paper. The proposed frame-parallel scheme utilizes the data independency of B-frame to overcome the computation complexity and memory bandwidth from B-frame, especially for high definition specification. The corresponding IME and FME architecture and scheduling are also introduced. By case study, the proposed design methodology can largely reduce 66% system bandwidth and only require 144MHz for 720p with IBBP coding scheme.

Table 1. Performance evaluation of proposed frame-parallel encoding schemes

	Huang's[3]	Bechmark[3]	Proposed
Specification	720p IPP	720p IBBP	720p IBBP
Frame parallelism	1	1	3
Bandwidth(MByte/s)	138.24	276.48	92.16
IME stage			
Loading SR cycle	320	640	640
SAD calculating cycle	512	1024	3072
Total IME cycle	832	1664	3392
FME stage			
FME operation stage	1	2	4
Total FME cycle	1000	2000	4000
Cycle/per stage	1000	2000	4000
Required Freq.(MHz)	108	216	$144^{\dagger}$
Design effort		High	Low

† : 4000×3600MB×30frame÷3=144MHz

7. REFERENCES

[1] Joint Video Team, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, 2003.

[2] T. Wiegand, G.J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–577, 2003.

[3] Y.-W. Huang and et al., "A 1.3tops H.264/AVC single-chip encoder for HDTV applications," in *Proc. of IEEE ISSCC*, 2005, pp. 128–588.

[4] Z. Liu and et al., "A 1.41w H.264/AVC real-time encoder SOC for HDTV1080P," in *2007 Symposium on VLSI Circuits Digest of Technical Papers*, 2007, pp. 2–1 12–13.

[5] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. IEEE International Conference on Multimedia and Expo*, 2006, pp. 1929–1932.

[6] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. CSVT*, vol. 12, no. 1, pp. 61–72, Jan. 2002.

[7] T.-C Chen, Y.-W. Huang, and L.-G. Chen, "Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC," in *Proceedings of IEEE ICASSP*, May 2004, pp. V–9–V–12.

[8] T.-C. Chen and et al., "2.8 to 67.2mW low-power and power-aware h.264 encoder for mobile applications," in *2007 Symposium on VLSI Circuits Digest of Technical Papers*, 2007, pp. 21–4 222–223.