A VLSI Implementation for Inter-Prediction Module in H.264/AVC Encoders

Nam-Khanh Dang, Van-Mien Nguyen, Xuan-Tu Tran†

SIS Laboratory, VNU University of Engineering and Technology – 144 Xuan Thuy road, Hanoi, Vietnam

Email: †tutx@vnu.edu.vn

Abstract The H.264/AVC is known as the emerging video coding standard which provides better video quality at a lower bit-rate than the previous ones thanks to many advances in coding technology. These prominent techniques are applied to remove efficiently spatial and temporal redundancies. However, because many coding tools have been adopted it makes the encoding system much more complex, especially the inter-prediction part of the coding system. This paper presents an efficient VLSI implementation for Inter-Prediction in the H.264/AVC encoders with three key proposals: full search algorithm with bandwidth efficient technique, pipelining technique, and data reuse strategy. With this approach, the Inter-Prediction has been efficiently implemented with a low cost in terms of latency, hardware area and memory bandwidth.

Keywords H.264/AVC, Inter-Prediction, Motion Estimation, Motion Compensation, Encoder, Video Coding

1. Introduction

The H.264 Advanced Video Coding (H.264/AVC) is known as the emerging video coding standard which provides better video quality at a lower bit-rate than previous standards [3]. The standard is jointly developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). Compared with the previous standards such as MPEG-4, H.263, and MPEG-2, the H.264/AVC can achieve respectively 39%, 49%, and 64% of bit-rate reduction [4] thanks to many advances in coding technology equipped with the standard.

A conventional Inter-Prediction in the H.264/AVC is composed of three main components: Integer Motion Estimation (IME), Fractional Motion Estimation (FME), and Motion Compensation (MC). The variable block-size IME predictes current macroblock (MB) from search windows to finds 41 motion vectors (MVs) of 41 sub-blocks. The FME refines 41 MVs with fractional components by interpolation and then chooses the best mode of MB base on MVs and distortion values. The MC block calculates predicted and residual MBs by using the selected mode and MVs after motion estimation step. Moreover, the Inter-Prediction also communicates with reference and current frames for prediction data and also encapsulates the information in encoding process.

Many VLSI implementations of the inter-prediction of H.264/AVC encoding systems have been recently proposed to get high-throughput design for real-time high-definition (HD) video applications such as in [1, 5, 6]. A conventional implementation is normaly composed of Motion Estimation (including Integer Motion Estimation and Fractional Motion Estimation) and Motion Compensation (MC). Related to the Inter-Prediction, several works presented in [1, 2, 6, 7, 9] had been implemented for IME, and the designs of FME were appeared in [6, 8, 10]. However, most of existing implementations do not explore efficiently the relationship between these components.

In our work, we first explore the relationship between these three main components in the inter-prediction. Then, we define a set of solutions such as full search algorithm, pipelining technique, and data reuse strategy to propose an efficient hardware architecture for the inter-prediction. The architecture has been implemented with a CMOS $0.18\mu m$ technology from ams AG and can encode CIF resolution video at an operating frequency of $24MHz$ with an area cost of $330KGates$. For encoding HDTV video, the system requires operating frequency of $215MHz$.

The remaining part of the paper is organized as follows. Section 2 is presented our proposed approach. An efficient hardware architecture of the inter-prediction will be also presented in this section. We also address the key techniques applied to the architecture

ISBN: 978-4-88552-282-6

Section 3 will show the implementation results as well as the comparisons with the previous works. Finally, conclusions and remarks will be given in Section 4.

2. Proposed Approach and Hardware Architecture

2.1. System Block Diagram

Figure 1 presents the block diagram of the full Inter-Prediction module which consists of IME, FME, CMC (Chroma Motion Compensation), EEI (Encapsulating Encoding Information), and several memory buffers such as: Current MB RAM, Search Window RAM, Motion Vectors Memory, and Residual/Predicted MB RAM. Because the *YUV 4:2:0* raw video format down-samples chroma components, only luma values are used for motion estimation.

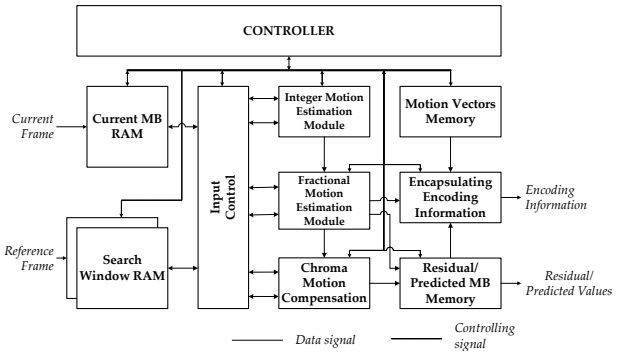

Figure 1: Block Diagram of Inter-Prediction.

The IME adopts the full-search algorithm for getting high accuracy in motion estimation. Moreover, the full-search algorithm also supports parallel variable block-size motion estimation for all modes in the H.264/AVC standard. In the proposed design, we employ two memory optimization techniques to obtain low-bandwidth solution. We also applied mode decision in IME to avoid memory requirements for motion vectors or predicted pixels and allow low-cost integrated chroma motion compensation in FME.

As the mode decision is integrated in IME, the FME only has to refine motion vectors with fractional components. The interpolated pixels are stored to be reused as predicted values. Therefore, the motion compensation of luma components is integrated after the motion estimation has been completed. In comparison with conventional method (the motion compensation of luma components is processed separately with the motion estimation), we are able to reduce the latency of re-generating sub-pixels and therefore optimize the memory capacity.

Because the luma compensation is predicted by FME, the chroma motion compensation only rebuilds the chroma components from search windows. As the chroma is down-sampled, the predicted chromas are generated via linear filter with three least significant bits of motion vectors which defined in the standard [3].

In Encapsulating Encoding Information module, the motion vector difference and Macroblock Mode are packed as the standard, and then are transferred to Entropy Coding block.

For memory organization, a data-reuse strategy is defined to optimize on-chip and off-chip data bandwidth. Both optimization techniques exploit the similarity between two regions to avoid re-reading data.

With all the above considerations, the Inter-Prediction can be efficiently implemented with a low cost in terms of latency, hardware area and memory bandwidth. The followed section will describe the methodology and the design of sub-modules inside Inter-Prediction.

2.2. Full Search Variable Block-size Integer Motion Estimation

The IME executes exhaustive search in windows which are mapped from current MB to reference frames. The design also supports parallel variable block-size motion estimation and mode decision.

The proposed design employs the full-search algorithm with a snake scanning movement as illustrated in Figure 2. The moving

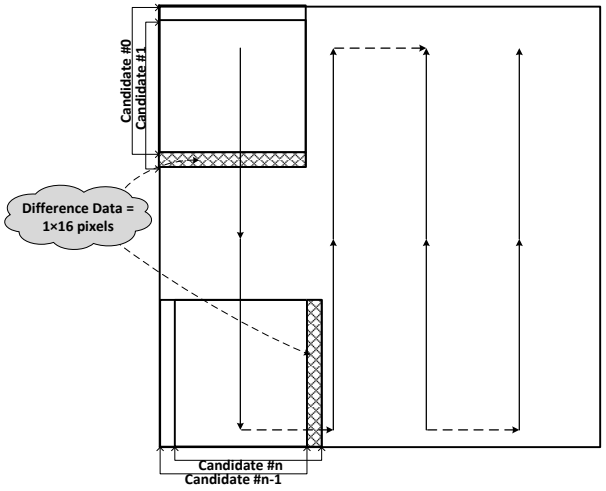

Figure 2: Full Search Algorithm: *fast switching between two neighboring candidates in full search algorithm by read a row/column* 1×16 *pixels.*

strategy includes three kinds of candidate shifting : down, right and up. By scanning column-by-column, this strategy covers all possible positions of the candidates with theirs Sum of Absolute Dif-

ISBN: 978-4-88552-282-6

ference (SAD) values. Moreover, this strategy is also suitable for on-chip memory optimization with overlapped regions.

With each candidate of searching, the IME calculates SAD value for each 4×4 block-size. The further block-sizes' SAD are obtained by accumulating of various blocks 4×4. Therefore, we obtain all SAD values of all block-sizes. With SAD values, IME is able to decide best motion vectors of block-sizes by indicating the smallest SAD candidate.

After completing search process, all SAD values and MVs are used for mode decision. Because the full search has the best accuracy, in our design the mode decision is integrated into the IME. The best mode and its motion vectors are then sent to FME for refining purpose.

The complete architecture of IME is presented in Figure 3. This architecture includes the memories for Search Windows as described in Section 2.5. The proposed searching strategy (as in Figure 2) requires a caching module to re-order the data. In coding flow, the caching module receives the data then send them to SAD matrix in three directions of caching. Current MB and Search Windows data are used to calculate SAD values in a matrix of 16×16 processing element (PE). These SADs (distortion value) are grouped in block-size of 4×4 and further block's SADs are obtain as summary of various 4×4 blocks. Base on these SADs, *minimum* module selects the smallest SAD values of all possible candidates to indicate the minimum distortion motion vectors. The MVs and theirs SADs after the searching are used for deciding the mode of MB via two stages: sub-partition mode and macroblock mode, as illustrated in H.264/AVC standard.

2.3. Fractional Motion Estimation with Integrated Luma Motion Compensation

Because the IME already decided the mode of MB, we have proposed a Fractional Motion Estimation (FME) architecture which only refines integer motion vectors with fractional elements. In addition, we integrates the motion compensation for luma components into this module.

As in H.264/AVC standard, the FME interpolates sub-pixels and refines in two steps using FIR model [3]. The first iteration generates 8 half-pixels and compares their SAD values to select the smallest SAD positions. After half-pixel generating stage, next iteration continues generating 8 quarter-pixels around a seleted position by reusing the previous sub-pixels. The best matched position also has the smallest SAD value. In addition, the sub-pixels after the second generating step is stored in RAM and will be reused as predicted values. Therefore, we integrated motion compensation for luma components inside FME.

In the other hand, if we put mode decision after FME, the integrated motion compensation requires all block-size sub-pixels values. Because the H.264/AVC standard supports 7 block-sizes, we can save 7 times of RAM capacity for motion compensation. In the other hand, the separate motion compensation also requires the same interpolating function which costs similar latency with only-refine FME and more memory space than our design for motion compensation.

Figure 4 shows the proposed FME architecture with three stages: half-pixel, quarter-pixel, and compensation. In half and quarter-pixel stages, the pixels are read, and then interpolated to obtain sub-pixels. The integer-pixels and half-pixels will be used to interpolate half-pixels and quarter-pixels respectively. Because FME only refines the motion vectors, we integrated the motion compensation for luma components at the last stage.

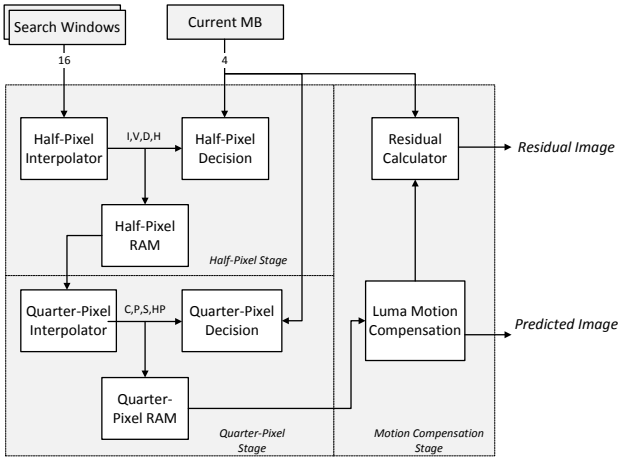

Figure 4: FME block diagram.

The result of FME includes residual/predicted pixels and the prediction information are packaged and transmitted to the following block in the coding flow. The chroma values (residual and predicted) are also calculated and transmitted via chroma motion compensation module.

2.4. Chroma Motion Compensation and Encapsulating Encoding Information

The chroma motion compensation generates predicted chroma elements for current MB base on mode and motion vectors from FME. In our design, a linear filter has been used in motion com-

ISBN: 978-4-88552-282-6

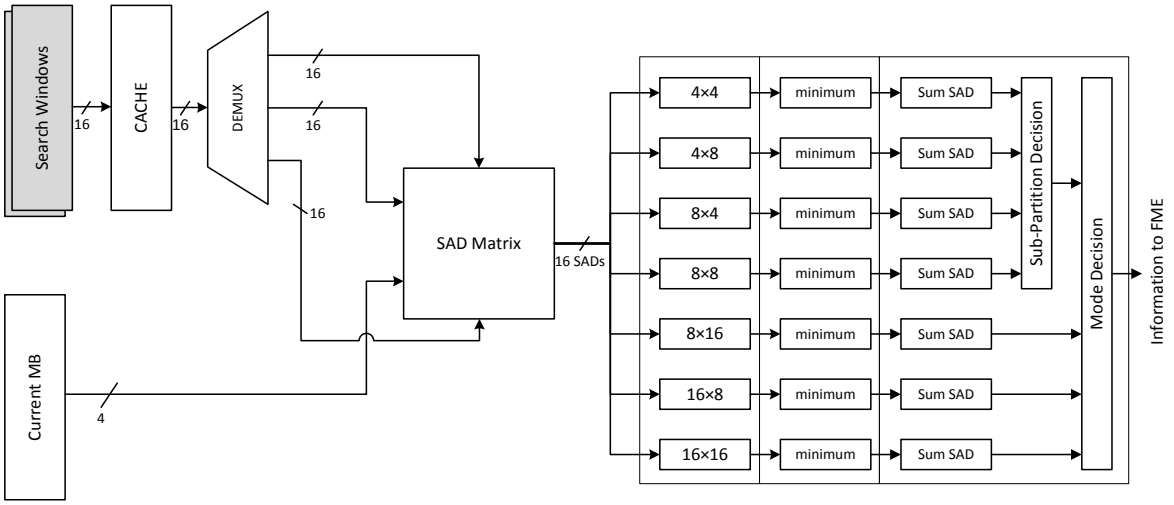

Figure 3: IME block diagram.

pensation to calculate sub-pixels as depicted in H.264/AVC standard [3].

In order to interface the Inter-Prediction with another functional modules of the H.264/AVC encoder, the Encapsulating Encoding Information (EEI) module communicates inside encoder and transmits the information to the following block in the coding flow. This module defines the prediction types of MBs, the position of MBs from the information stored in system's registers, and then encapsulates the residual/predicted values and prediction information. The motion vectors and residual/predicted values are stored in memory to support the pipeling technique in encoder system.

2.5. Memory Organization

One of the essential aspects of our proposal is the optimization for memory space of Inter-Prediction. We have considered to two optimizing techniques: off-chip technique which transfers data between the integrated circuit (IC) and an external DDR memory, and on-chip which communicates inside H.264/AVC encoder IC.

With the off-chip memory cost, an analysis in [1] points out that the Inter-Prediction costs more than 90% of data-bandwidth from RAM to encoding process and leads to bottle-neck affect inside the encoding system. In our proposal, we has defined search windows by a centerized mapping method from the current MB to the reference frame. Thus, the search windows of two neighboring MBs has overlapped area which is exploited to decrease the off-chip bandwidth.

The Figure 5 shows the overlapped area between two neighboring MBs (#1 and #2). In general, the search window is defined as $SR_H \times SR_W$ pixels with SR_H is height of search range and SR_W is width of search range. As shown in the figure, the search windows #2 can be obtained by reusing the region of $SR_H \times (SR_W - 16)$ pixels from search windows #1 and read a new $SR_H \times 16$ pixels. For example, with a search range of 48×48, we can save at least 60% of the off-chip data bandwidth while only extend 33% memory capacity.

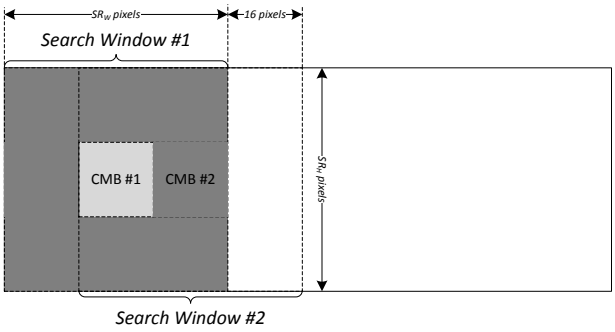

Figure 5: Overlapped region between two neighboring search windows.

To optimize on-chip bandwidth, the search engine employs the caching between two candidates of a search as illustrated in Figure 2. To obtain the overlapped region between two candidates, we use a scanning method with only one pixel movement. With this scanning, we can switch between the previous candidate to the current candidate by reading only a row or a column of 1×16 pixels. Therefore, IME can obtain a maximum speed of search if the data bandwidth from internal RAM to search engine obtain 16 pixels/second.

ISBN: 978-4-88552-282-6

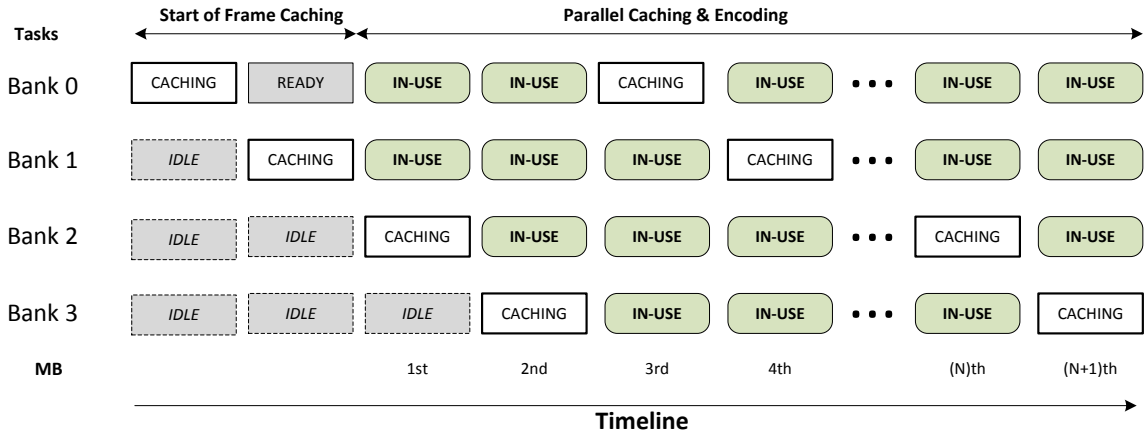

Figure 6: Caching strategy for Search Window RAM.

The strategy of memory reusage for inter-prediction is shown in Figure 6. In order to overlap the search windows, system uses model of "circular shift memory" as shown in Figure 5. In our design specification, the search windows is 48×48 which is equivalent to 3×3 macroblocks. Therefore, we use 4 banks of memory to cach data, in which 3 banks is used for prediction and 1 bank is used to read the next different pixels between 2 serial search windows.

3. Implementation Results and Comparisons

The proposed design has been modeled in VHDL at RTL level, and then synthesized with a CMOS $0.18\mu m$ technology from ams AG. With the search range of 48×48 and 1 reference frame per list, maximum computing capability is a real-time *Main Profile* CIF video encode at a frequency of 24*MHz* or an HD video at a frequency of 215*MHz*. Table 1 shows the total hardware cost of our design compared with Cheng et al.'s [1] and Lin et al.'s [6] designs. Our design implementation consists of both motion estimation steps, motion compensation, memories, and interfacing modules while the implementation results of both mentioned designs only include motion estimation. Moreover, our design can provide *Main Profile* encoding process with *bi-predictive*, which requires double RAMs for Inter-Prediction and also costs double time for searching. In comparison, the proposed design costs medium area which is equivalent to a half of Cheng et al's design and approximates Lin et al.'s design while we integrated additional functions inside. With the off-chip memory optimization and fast mode decision techniques, the proposed design costs only 16.7*KBytes* for full search while Cheng et al.'s required 27*Kbytes*. The design of Lin et al.'s required 8*Kbytes* approximately with multi-resolution search and 2-candidates on FME but it lacks motion compensation and residual/predicted memories. Moreover, the algorithm of Lin et al.'s design lacks of the accuracy in comparison with the full-search algorithm. In summary, our proposed Inter-Prediction design can achieve a low area cost, high accuracy, and the design is suitable for mobile applications. The design has been finally integrated into a hardware H.264 video encoding system.

Table 1: Comparison of Inter-Prediction module

Spec	**[1]**	**[6]**	**This work**
Technology	$0.18\mu m$	$0.13\mu m$	$0.18\mu m$
Freq (*MHz*)	81/108	28.5/128.8	24/215
Gate Count	$700K$	$208.6/282.6K$	$330K$
RAM	27 *KBytes*	7.78/8.54 *KBytes*	16.7 *KBytes*
IME	Full Search	Multi-Resolution	Full Search
FME	17 candidates 2-iteration interpolation	6 candidates 1-iteration interpolation	17 candidates 2-iteration interpolation
Resolution	SDTV/HDTV	720p/1080p	CIF/HDTV
Profile	Baseline (4/1 ref.(s))	Baseline (1 ref.)	Main Profile (2 lists $\times$ 1 ref.)

4. Conclusions

This paper represents a VLSI implementation for Inter-Prediction in H.264/AVC encoders. We applied the full-search algorithm with bandwidth efficient techniques. In addition, the fast mode decision provides better performance and leads to integrated motion compensation inside estimation block. With search range of 48×48 and bi-predictive support, the proposed architecture costs 300*KGates* and 16.7*KBytes* RAM capacity. With this architecture,

ISBN: 978-4-88552-282-6

the real-time *HD* and *CIF* video encoding can be obtained at frequencies of $215MHz$ and $24MHz$, respectively. Therefore, the proposed architecture provides an efficient solution for Inter-Prediction in H.264/AVC encoders with low bandwidth, high performance, and high accuracy.

Acknowledgment

This work has been done in the framework of project No.QGDA.10.02 (VENGME), funded by Vietnam National University, Hanoi. The project aims at developing a hardware for video encoding system based on the H.264/AVC standard, targeted to mobile applications. We would like to express special thanks to Synopsys for providing EDA tools, CMP and AMS for providing CMOS $0.18\mu m$ technology libraries.

References

[1] Tung-Chien Chen et al. Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(6):673–688, 2006.

[2] Tung-Chien Chen et al. Fast algorithm and architecture design of low-power integer motion estimation for H. 264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(5):568–577, 2007.

[3] ITU. ITU-T Recommendation H.264: Advanced video coding for generic audiovisual services, 2006.

[4] A. Joch et al. Performance comparison of video coding standards using lagragian coder control. In *IEEE International Conference on Image Processing, 2002*, pages 501–504. IEEE, 2002.

[5] Jae Hun Lee and Nam Suk Lee. Variable block size motion estimation algorithm and its hardware architecture for H. 264/AVC. In *Proceedings of the 2004 International Symposium on Circuits and Systems, 2004*, volume 3, pages III–741. IEEE, 2004.

[6] Yu-Kun Lin et al. A Hardware-Efficient H.264/AVC Motion-Estimation Design for High-Definition Video. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(6):1526–1535, 2008.

[7] Marcelo Schiavon Porto et al. An efficient ME architecture for high definition videos using the new MPDS algorithm. In *Proceedings of the 24th symposium on Integrated circuits and systems design*, SBCCI '11, pages 119–124, New York, NY, USA, 2011. ACM.

[8] G.A. Ruiz and J.A. Michell. An Efficient VLSI Architecture of Fractional Motion Estimation in H.264 for HDTV. *Journal of Signal Processing Systems*, 62(3):443–457, 2011.

[9] G.A. Ruiz and J.A. Michell. An efficient VLSI processor chip for variable block size integer motion estimation in H.264/AVC. *Signal Processing: Image Communication*, 26(6):289 – 303, 2011.

[10] Changqi Yang, S. Goto, and T. Ikenaga. High performance VLSI architecture of fractional motion estimation in H.264 for HDTV. In *2006 IEEE International Symposium on Circuits and Systems*, pages 4 pp.–, 2006.