

An Efficient Hardware Architecture for Inter-Prediction in H.264/AVC Encoders

Nam-Khanh Dang*, Xuan-Tu Tran*, and Alain Merirot†

*SIS Laboratory, VNU University of Engineering and Technology, 144 Xuan Thuy road, Cau Giay, Hanoi, Vietnam.

†University Paris Sud XI, 91192 Gif-sur-Yvette Cedex, France.

Corresponding author's email: tutx@vnu.edu.vn

Abstract—In this paper, we propose a design methodology for the inter-prediction in H.264/AVC codecs by addressing the relationship between its main processes. The target of this methodology is to optimize the design in order to get better performance while keeping a reasonable design cost. An efficient hardware architecture for the inter-prediction in H.264/AVC codecs is then proposed with three key techniques: a modified full search algorithm with bandwidth efficiency, pipelining technique, and data reuse strategy. With this approach, the inter-prediction has been successfully designed and implemented with a CMOS 180nm technology which provides low cost in terms of latency, hardware overhead and memory bandwidth. The design is initially targeted to CIF video format; however, it is obviously suitable for real-time HD 1080p video format.

I. INTRODUCTION

The H.264/AVC (Advanced Video Coding) is known as one of the latest and most efficient video coding standards which provides better video quality at a lower bit-rate than previous standards [1]. Even the new video coding standard, named HEVC, has been recently introduced in 2013, the efficient hardware implementation of the H.264/AVC still plays an important role in current multimedia devices to get real-time, efficient video coding/decoding applications. The H.264/AVC standard is jointly developed by the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. 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 [2] thanks to many advances in coding technology equipped with the standard. These prominent techniques are Context Adaptive Variable Length Coding (CAVLC), variable block size motion detection, in-loop deblocking filter which 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 system.

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 [3]–[6]. A conventional implementation is normally composed of Motion Estimation (including Integer Motion Estimation (IME) and Fractional Motion Estimation (FME)) and Motion Compensation (MC). Most of existing implementations do not explore efficiently the relationship between these components. In our work, we first investigate in exploring the relationship between these three main components of the inter-prediction. Then, we define a set of solutions such as a modified full search algorithm for better bandwidth efficiency, pipelining technique, and data reuse strategy to propose an efficient hardware architecture for the inter-prediction. The architecture has been implemented

with CMOS 180nm technology from ams AG and can encode CIF resolution video at an operating frequency of 24MHz with an area cost of 330K Gates (it is also able to encode HDTV video at the operating frequency of 215MHz).

The remaining part of the paper is organized as follows. Section II will address the main principles of the inter-prediction and the state of the art on its hardware implementations. In Section III, we will describe our proposed approach and methodology to develop an efficient inter-prediction hardware architecture. Key proposed techniques will be addressed and discussed in this section. Section IV presents the VLSI hardware architecture for the inter-prediction and addresses the implementation results as well as some comparisons with the previous works. Finally, conclusions and remarks will be given in Section V.

II. REVIEW OF INTER-PREDICTION IN H.264/AVC ENCODERS

A. Inter-Prediction in H.264/AVC

A conventional Inter-Prediction in H.264/AVC is composed of three main components: IME, FME, and MC. The variable block-size IME predicts current macroblock (MB) from search windows to find 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 getting prediction data and also encapsulates the information in encoding process.

In H.264/AVC, the IME processes seven kinds of block size. A MB can be predicted with one of four MB modes: 16×16 to 8×8 . When mode 8×8 is selected, each partition 8×8 can be predicted with a sub-macroblock mode from 8×8 to 4×4 . Therefore, there are totally 41 MVs corresponding to 41 block-types. The FME interpolates half and quarter pixels from reference frames, then predicts fractional components. The MVs and their distortions are used to select mode for each MB. The MB's mode, reference indexes, MVs are sent to MC and followed units for encapsulating the encoding information. The MC receives the information related to the selected mode, reference indexes and MVs from motion estimation modules to re-build the predicted MB. Base on this information, MC obtains the prediction of current MB base on values on reference frames. In addition, the residual values are also calculated for next encoding process.

The information of prediction is then encapsulated in structure which defined in H.264/AVC standard. The encapsulated

data usually consists of MVs' difference, MB's type, sub-macroblock's type, and reference indexes.

B. State of the Art

In order to get high encoding performance, various hardware implementation approaches for inter-prediction have been recently reported in literature. The designs presented in [3], [7]–[9] use full search as block-matching algorithm which is easy to be implemented in hardware and it has the best accuracy. However, the full search algorithm is not suitable for larger search ranges and very expensive in hardware implementation and computation power. Other designs used Three Step Search (3SS) algorithm [10] or diamond-based search algorithm [11], [12] but they do not support variable block-sizes. Another method is multi-resolution search [13]–[15] which provides low latency for prediction. Although multi-resolution obtains best performance of coding, the motion vector may lack of accuracy and this algorithm is only suitable for high-definition video which has high similarity between neighboring MBs. In our case, the CIF resolution is intently focused for mobile devices. Therefore, the full search algorithm is applied with some proposed techniques for decreasing hardware implementation cost, memory bandwidth and latency, while keeping the accuracy of the algorithm.

In the other hand, almost previous designs implement motion estimation without exploiting the relationship between motion estimation and motion compensation. In our proposal, the design integrates motion compensation for luma components inside FME to reduce latency and area.

III. PROPOSED APPROACH FOR INTER-PREDICTION IMPLEMENTATION

A. Methodology

Our approach consists of three parts: motion estimation which includes integer and fractional steps, motion compensation, and data-reuse strategy. The design methodology intently focuses on best-accuracy video encoding while reducing area-cost and memory bandwidth. In addition, the proposed approach also exploits the relationship between motion estimation and motion compensation in order to optimize the whole encoding process.

In our encoder, video specification of encoder is $YUV\ 4:2:0$. With $4:2:0$, chroma components is half down-sampled. Therefore, chroma components are insufficient for motion estimation. We only perform motion estimation for luma components. The chroma components are only compensated.

The IME adopts the full-search algorithm. Although this algorithm costs expensive in computation, it has best accuracy for motion estimation. In the other hand, the full-search algorithm also supports parallel variable block-size motion estimation for all modes in the H.264/AVC standard. By centering search windows with current MB, the overlapping rule of two neighboring search windows saves at least 66% data bandwidth for reading data from external RAM with search range 48×48 . In addition, the on-chip bandwidth is intently optimized to obtain the best pipelining computation speed. We also applied mode decision in IME which avoids memory requirement for motion vector or predicted pixels and allows low-cost integrated chroma MC in FME.

With mode decision integrated in IME, the FME only has to refine the motion vector with fractional components. The interpolated pixels are stored to be reused as predicted values. Therefore, we integrate the MC of luma component in after finish FME process. Thus, we reduce the latency of re-generating sub-pixels and optimize the memory capacity.

Because the luma compensation is predicted inside FME, the chroma motion compensation only rebuilds the chroma components from search windows. The motion vector difference, Macroblock Mode is also packed as the standard and then is transferred to Entropy Coding block.

The data-reuse strategy is defined to optimize on-chip and off-chip data bandwidth. Both optimizations exploit the similarity between two regions to avoid re-reading data. Therefore, this technique maximizes the calculating speed of motion estimation and motion compensation.

With all the above considerations, the Inter-Prediction can be efficiently implemented with low cost in terms of latency, hardware area and memory bandwidth. This is suitable for mobile applications.

B. Full Search Variable Block-size Integer Motion Estimation

As mentioned above, in our approach, the IME executes exhaustive search in windows which are mapped from current MB to reference frames. The IME is also designed to support parallel variable block-size motion estimation and mode decision. Fig. 1 presents the proposed design using full-search algorithm with scanning movement.

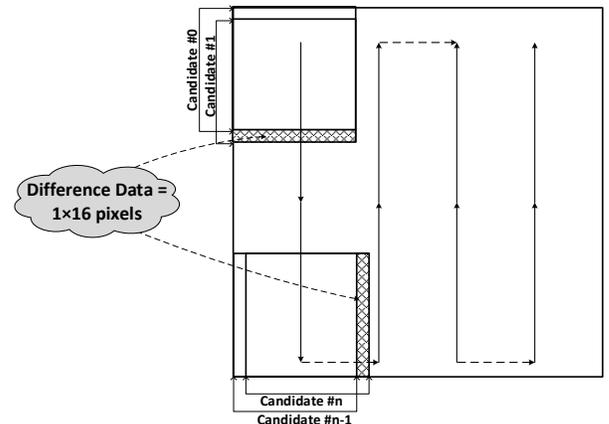


Fig. 1. Full Search Algorithm: fast switching between two neighboring candidates in full search algorithm by read a row/column 1×16 pixels.

The moving strategy includes three kinds of candidate shifting: *down*, *right*, and *up*. By scanning column-by-column, this algorithm covers all possible positions of the candidate with theirs Sum of Absolute Differences (SAD) values. Moreover, this strategy is also suitable for on-chip memory optimization with overlapped region.

With each candidate of searching, IME calculates the SAD value for each 4×4 block-size. The further block-sizes' SAD values are obtained by accumulating of various blocks 4×4 . Therefore, we obtain all SAD values of all block-sizes.

With SAD values, IME decides best MVs of block-sizes by indicating the smallest SAD candidate.

After finishing this search, all SAD values and MVs are used for mode decision. Because the full search algorithm has best accuracy, we decide put mode decision on IME. The best mode and its MVs are sent to FME to be refined.

C. Fractional Motion Estimation with Integrated Luma Motion Compensation

Because the IME already decided the mode of MB, in our proposal, the FME only refines integer MVs with fractional elements.

The FME interpolates sub-pixel and refines in two steps using FIR models as defined in the standard [1]. The first iteration generates 8 half-pixels and compares their SAD values to select the smallest SAD position. After half-pixel stage, the next iteration continues generating 8 around quarter-pixels and choosing position by reusing the previous sub-pixels. The best matched position also has smallest SAD value. In addition, the sub-pixels got from the second step is stored in RAM and can be reuse as predicted value. Therefore, we integrate motion compensation for luma components inside FME.

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

D. Data-reuse strategy

To optimize memory space and minimize the data exchange between the memories and Inter-Prediction unit, we proposed a data-reused strategy. In fact, the Inter-Prediction unit communicates with off-chip memories for getting data from reference frames before starting a new encoding process. Then, this data is stored in on-chip memories for further process during encoding. To optimize off-chip memory exchange, an analysis of JM reference software [16] in [3] points out that the Inter-Prediction costs more than 90% of data-bandwidth from RAM for encoding process and leads to bottle-neck affect inside the encoding system. In our proposal, we define search windows by a centralized mapping method from current MB to reference frames. Thus, the search windows of two neighboring MBs have an overlapped area which is intently exploited to decrease the off-chip bandwidth.

The Fig. 2 shows the overlapped area between two neighboring search windows (SW#1 and SW#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 Fig. 2, the SW#2 can be obtained by reusing region of $SR_H \times (SR_W - 16)$ pixels from SW#1 and reading new $SR_H \times 16$ pixels. In comparison, Table I shows the advantages of the overlapping technique. For example, with the proposed search range of 48×48 , we can save at least 66% of off-chip data bandwidth while only extend 33% memory capacity.

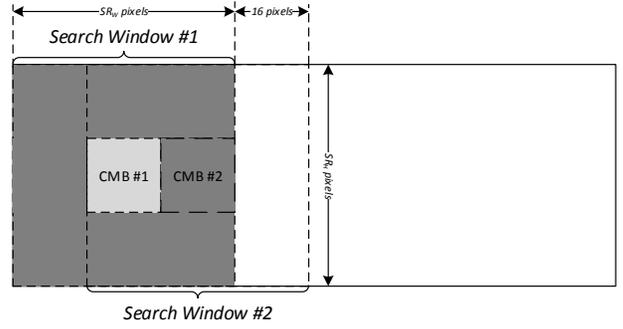


Fig. 2. Overlapped region between two neighboring search windows.

TABLE I. COMPARISON OF BANDWIDTH OF READING SW

Memory type	Amount (pixels)	Bandwidth (pixels/MB)
Direct Design	$SR_H \times SR_W$	$SR_H \times SR_W$
Proposed Method	$SR_H \times (SR_W + 16)$	$SR_H \times 16$
Changing	$+SR_H \times 16$	$-SR_H \times (SR_W - 16)$

To optimize the on-chip bandwidth, the search engine employs a caching technique between two searching candidates as illustrated in Fig. 1. To obtain the overlapped region between two candidates, we use a scanning method with only one different pixel. With this scanning technique, we can switch from the previous candidate to the current candidate by reading only one row or one column 1×16 pixels. Therefore, IME can achieve the maximum searching speed if the reading process for the additional data can be done in one cycle.

IV. VLSI ARCHITECTURE AND IMPLEMENTATION

From the proposed approach, the block diagram of our Inter-Prediction unit is depicted in Fig. 3. The design is composed of IME (Integer Motion Estimation), FME (Fractional Motion Estimation), CMC (Chroma Motion Compensation), EEI (Encapsulating Encoding Information), and several memory buffers: CMB RAM (Current MB RAM), SW RAM (Search Window RAM), MV MEM (Motion Vectors Memory), and RES/PRED MEM (Residual/Predicted Memory).

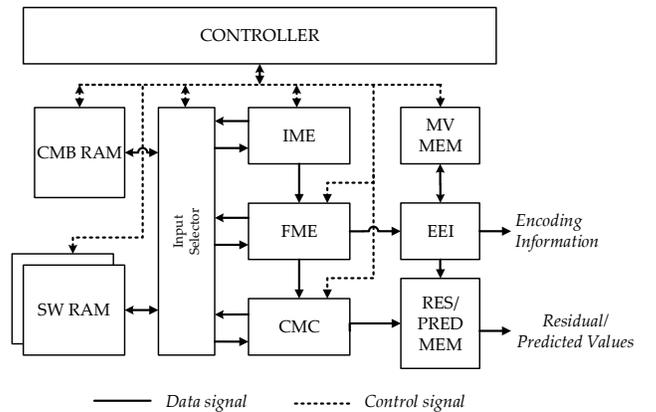


Fig. 3. Block Diagram of Inter-Prediction.

In order to interfaces Inter-Prediction with another modules inside the H.264/AVC encoder, the EEI (Encapsulating En-

coding Information) module communicates inside encoder and also transmits information to followed block in coding flow. This module defines the prediction type of MB, the position of MB from information of system's register and encapsulates the residual/predicted value and prediction information. The MVs and residual/predicted are stored in memory to support pipelining technique in encoder system.

The proposed design has been implemented by VHDL and synthesized by the 180nm CMOS technology from ams AG. Maximum computing capability is real-time encode *Main Profile* CIF video at 24MHz and HD video at frequency 215MHz. Table II shows the total hardware cost of our design and comparison with Cheng et al.'s [3] and Lin et al.'s [15] design. Our design implementation consists of both motion estimation steps, motion compensation, memory of design and interfacing modules while the implementation result of both existing designs only include motion estimation. Moreover, our design can provide *Main Profile* encoder with *bi-predictive*, which requires double RAMs for Inter-Prediction and also cost double time for searching. In comparison, the proposed design cost medium area which is equivalent to a half of Cheng et al.'s design and approximate to Lin et al.'s design while we integrated additional functions inside. With off-chip memory optimization and fast mode decision, the proposed design cost only 16.7Kbytes for full search while Cheng et al.'s required 27Kbytes. The design presented by Lin et al. requires 7.78/8.54Kbytes with multi-resolution search and 2-candidates on FME but it lacks motion compensation and residual/predicted memory. Moreover, the algorithm of Lin et al.'s design lack of accuracy in comparison with full-search algorithm. In summary, the proposed Inter-Prediction design can archive low area cost and high accuracy. In addition, this design supports for Main Profile, which can be easily extended for other profiles, and is suitable for mobile applications.

TABLE II. COMPARISON OF INTER-PREDICTION DESIGNS.

Specification	[3]	[15]	Proposed
Technology	180nm	130nm	180nm
Freq. (MHz)	81/108	28.5/128.8	24/215
Area (KGates)	700	208.6/282.6	330
RAM (Kbytes)	27	7.78/8.54	16.7
IME Algorithm	Full Search	Multi-Resolution	Full Search
FME Algorithm	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 × 1 ref.)

V. CONCLUSIONS

We have presented in this paper an efficient hardware design and implementation for inter-prediction in H.264/AVC encoders. The modified full-search algorithm with bandwidth efficiency technique, pipelining technique, and data reuse strategy are applied to improve the encoding performance and reduce the implementation costs in terms of latency, hardware overhead, memory bandwidth. In addition, the fast mode decision makes better performance and leads to the integration of motion compensation block inside estimation block. The proposed architecture has been fully modeled, verified,

and synthesized using CMOS 180nm technology from ams AG. The design occupies 330KGates and 16.7Kbytes RAM capacity. With the search range of 48 and bi-predictive support, the proposed architecture is able to encode CIF resolution video at an operating frequency of 24MHz (as the target of the project); however, it is also able to encode HD1080p video at an operating frequency of 215MHz).

ACKNOWLEDGMENT

The authors would like to thank Nafosted for travel grant.

REFERENCES

- [1] *ITU-T Recommendation H.264: Advanced video coding for generic audiovisual services*, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG Std., 2006.
- [2] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. Sullivan, "Performance comparison of video coding standards using lagragian coder control," in *Image Processing, 2002. ICIP 2002. IEEE Int'l Conf. on*. IEEE, 2002, pp. 501–504.
- [3] T.-C. Chen and et al., "Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 6, pp. 673–688, 2006.
- [4] G. Ruiz and J. Michell, "An Efficient VLSI Architecture of Fractional Motion Estimation in H.264 for HDTV," *Journal of Signal Processing Systems*, vol. 62, no. 3, pp. 443–457, 2011.
- [5] C. Yang, S. Goto, and T. Ikenaga, "High performance VLSI architecture of fractional motion estimation in H.264 for HDTV," in *Proceedings 2006 the IEEE Int'l Symposium on Circuits and Systems*, 2006.
- [6] R. Porto, L. Agostini, and S. Bampi, "Hardware Design of the H.264/AVC Variable Block Size Motion Estimation for Real-Time 1080HD Video Encoding," in *Proceedings of the 2009 Symposium on VLSI*, 2009, pp. 115–120.
- [7] S.-M. Pyen and et al., "An Efficient Hardware Architecture for Full-Search Variable Block Size Motion Estimation in H.264/AVC," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science. Springer, 2006, vol. 4292, pp. 554–563.
- [8] M. Kim, I. Hwang, and S.-I. Chae, "A fast VLSI architecture for full-search variable block size motion estimation in MPEG-4 AVC/H. 264," in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*. ACM, 2005, pp. 631–634.
- [9] G. Ruiz and J. Michell, "An efficient VLSI processor chip for variable block size integer motion estimation in H.264/AVC," *Signal Processing: Image Communication*, vol. 26, no. 6, pp. 289–303, 2011.
- [10] T.-C. Chen and et al., "Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC," *Trans. on Circuits and Systems for Video Techno.*, vol. 17, no. 5, pp. 568–577, 2007.
- [11] M. S. Porto and 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*, ser. SBCCI '11. New York, NY, USA: ACM, 2011, pp. 119–124.
- [12] G. Sanchez and et al., "DMPDS: A Fast Motion Estimation Algorithm Targeting High Resolution Videos and Its FPGA Implementation," *International Journal of Reconfigurable Computing*, vol. 2012, 2012.
- [13] J. H. Lee and N. S. Lee, "Variable block size motion estimation algorithm and its hardware architecture for H. 264/AVC," in *Proceedings of the 2004 IEEE Int'l Symposium on Circuits and Systems*, vol. 3. IEEE, 2004, pp. III–741.
- [14] H. Yin and et al., "A Hardware-Efficient Multi-Resolution Block Matching Algorithm and its VLSI Architecture for High Definition MPEG-Like Video Encoders," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 9, pp. 1242–1254, 2010.
- [15] Y.-K. Lin and et al., "A Hardware-Efficient H.264/AVC Motion-Estimation Design for High-Definition Video," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 55, no. 6, pp. 1526–1535, 2008.
- [16] Joint Video Team, "Reference Software JM 7.3," Aug. 2003. [Online]. Available: <http://bs.hhi.de/suehring/tml/download/>