An H.264/AVC MP@L4.1 Quarter-Pel Motion Estimation Processor VLSI for Real-Time MBAFF Encoding

Kosuke Mizuno, Junichi Miyakoshi, Yuichiro Murachi, Masaki Hamamoto, Takahiro Iinuma,
Tomokazu Ishihara, Fang Yin, Jangchung Lee, Hiroshi Kawaguchi, and Masahiko Yoshimoto
Department of Computer Science and Systems Engineering
Kobe University
Kobe, Japan

Abstract— **This paper describes an H.264/AVC MP@L4.1 quarter-pel motion estimation processor core for a low power video encoder. It supports macro block adaptive frame field (MBAFF) encoding and bi-directional prediction for a resolution of 1920 × 1080 pixels at 30 fps which haven't been realized by conventional methods yet. The proposed processor consists of four modules for low power consumption: a module for an integer-pel motion estimation, a segmentation-free rectangle-access search window buffer, a module for quarter-pel motion estimation, and a module reducing candidate motion vectors. We propose an adaptive algorithm that reduces a workload and power in quarter-pel motion estimation. The algorithm and architecture for the candidate motion vectors reduction suppress a workload of the following process. The processor core has been designed in a 90 nm CMOS technology. The core size is 6.0 × 6.0 mm^2. With this core, two reference frame can be handled, and 160.1 mW is consumed at 1.0 V.**

I. Introduction

In H.264/AVC, more than double workload of the conventional MPEG-2 is necessary for higher picture quality and lower bitrate [1]. Several H.264/AVC motion estimation (ME) processor cores have been developed to save the workload and power. ME is composed of an integer-pel ME (IME) and a fractional pel ME (FME). The IME finds integer-pel accuracy motion vectors (MVs), and then the FME calculates quarter-pel accuracy MVs using the integer-pel accuracy MVs. Because the workload of the ME accounts for over 80~90% in an H.264/AVC encoder, power reduction in the ME is essential to suppress the total power. It is also crucial to reduce a workload in a motion vector extractor (MVE) which selects the optimum vector from results of ME (Fig. 1). We propose the following two techniques to reduce the power in the ME and the MVE while keeping high picture quality:

- An FME algorithm that uses results out of our proposed IME[2]. This ME algorithm adaptively executes frame mode prediction or field mode prediction for macro block adaptive frame field (MBAFF) encoding, which reduces a workload and power with low picture degradation.
- An algorithm and architecture for candidate MV reduction in the MVE. The MVE cuts down a workload by refining motion vectors with an SATD index.

We implemented these techniques and designed an ME processor in a 90-nm process technology. The processor supports MBAFF encoding at 1920 × 1080 pixels and 30 fps, which haven not been realized by conventional methods [3-4].

This work was supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Cadence Design Systems, Mentor Graphics and Synopsys, Inc.

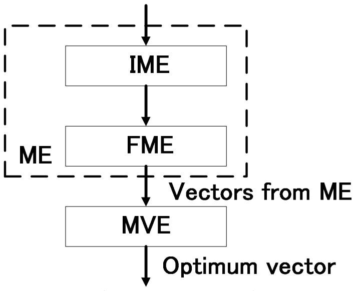

Fig. 1 Processing to extract motion vector.

II. Algorithm

A. IME Algorithm

The algorithm is a hierarchical search algorithm consisting of a coarse search and two fine searches. The flow chart is shown in Fig. 2.

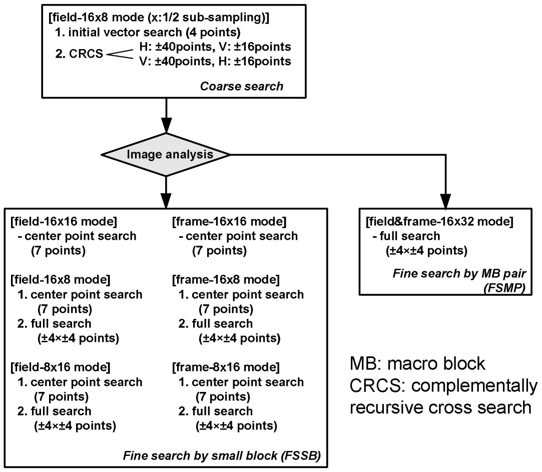

Fig. 2 IME algorithm flow.

The coarse search finds suboptimal MVs over a wide area in a search window, and analyzes the distribution of MVs for variable size of blocks. As the coarse search, a one-dimensional search method with low complexity and high picture quality is conducted, named complementally recursive cross search (CRCS). The method of the CRCS is illustrated in Fig. 3. In CRCS, 2 RCSs (recursive cross search) are executed in parallel with the same initial point. When search is over, the vector with smaller SAD between the two vectors of RCSs is chosen as result of CRCS.

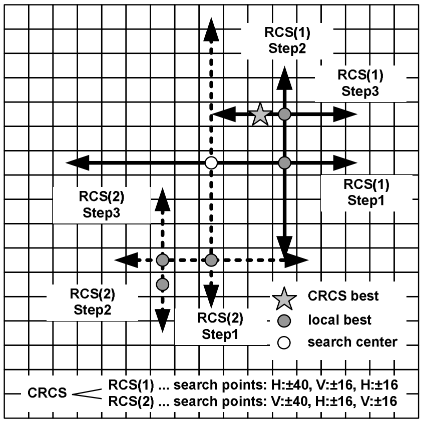

Fig. 3. Complementally recursive cross search (CRCS).

Then, either of two fine searches is carried out. The two fine searches comprise a fine search by small block (FSSB) and a fine search by macro block pair (FSMP), one of which is chosen according to the result of image analysis. In FSSB, center point searches are carried out for 2 MBAFF modes and 3 block sizes. The purpose of the center point searches is to adjust the coarse search results to each mode's optimum. Then full searches are conducted to increase picture quality. In FSMP, a full search is conducted at MB pair size.

B. *FME Algorithm*

We propose a new FME search algorithm to reduce the workload. The proposed FME algorithm firstly determines relative merits of both MBAFF modes comparing SAD result of IME. The MBAFF mode of the smaller SAD is called superior MBAFF mode and the other inferior is called inferior MBAFF mode. Superior MBAFF mode is searched for 16 × 16, 8 × 16, and 8 × 8 block sizes to conduct high accurate estimation, while the inferior MBAFF mode is only for a 16 × 16 block size to reduce the workload as shown in Fig. 4(a). For each above block size and MBAFF mode, neighbor search algorithm (NSA) is conducted as shown in Fig. 4(b). NSA is a 2 step search algorithm. In step 1, IME result and its neighbor 8 half-pels are searched and one best point is determined. In step 2, neighbor 8 quarter-pels of the best point is searched. Search cost of the NSA is sum of the absolute transferred difference (SATD) with 4 × 4 Hadamard transform. The search cost is reused to calculate SATD of smaller block sizes in the same search point. By the SATD reusing scheme, the proposed FME algorithm finds SATD and MV of all 4 block sizes and both MBAFF mode is searched in the only 33% workload compared to the conventional all mode search.

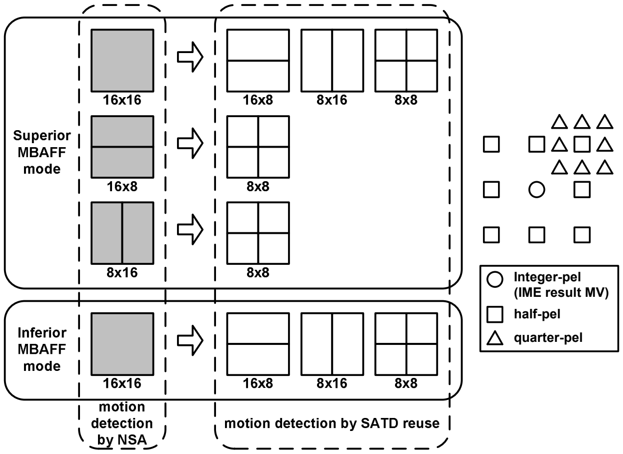

Fig. 4. (a) FME search strategy (b) neighbor search algorithm (NSA).

C. *Candidate-MVs Reduction Algorithm*

According to FME algorithm, 162 motion vectors are outputted for P frame and B frame. All the motion vectors were transmitted to external ME (MVE) in conventional way. In MVE, the best motion vector is decided by comparing the cost of each motion vector. Multiplication in MVCOST calculation causes increase of power. So it is necessary to reduce the number of motion vectors before transmitting to MVE. We proposed the Algorithm selecting motion vectors which could reduce the number of motion vectors to 14 by SATD.

Figure 5 showed proposed algorithm flow. We select vectors by comparing SATD. Firstly reference picture selection is performed. Secondly L0 prediction or L1 prediction is decided for B picture. Then frame prediction or field prediction is decided. At last, block size is selected and the motion vectors can be reduced to 14.

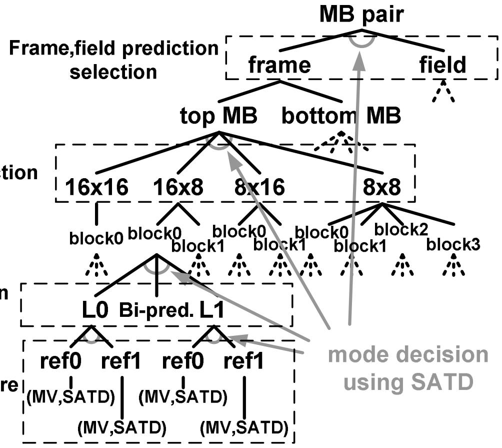

Fig. 5. Candidate-MVs reduction algorithm.

D. *Simulation Result*

The parameters in the algorithm simulation are as follows: a resolution is HD (1920 × 1080 interlace), a frame rate is 30 fps, the number of reference frames are two for a P frame and two for a B frame, MBAFF is supported, and search range is ±128 × ±64. We used 13 test sequences including quick motion and high activity picture (Bronze with Credits, Church, European Market, Whale Show, Soccer Action, Track with Credits, Buildings along the Canal, View from Sky with Credits, Intersection, Streetcar, Yachting, Japanese Room, and Yacht Harbor). Fig. 6 shows comparisons of average workload and average PSNR over the above sequences between the proposed and conventional algorithms. The workload of the proposed algorithm is reduced to 9.7% and the picture degradation is only 0.098dB, compared with the conventional hybrid unsymmetrical cross multi hexagon grid search (UMHS) [5].

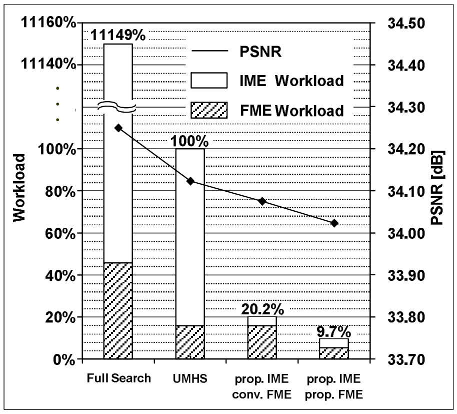

Fig. 6. Workload and PSNR of the proposed ME algorithm.

III. Architecture

A. Architecture Overview

A whole diagram of the proposed ME processor is illustrated in Figure. 7. One core has IME, FME, image cache, and other controllers. ME with two reference picture is conducted by using two cores. SATD comparator executes selection of motion vectors after getting results of ME from two cores. A search window buffer RAM (SWRAM: dual port = one read / one write) provides reference image data to the IME_PU and SME_DATAPATH through a cross path circuit. The cross path circuit rotates pixel data on demand. A template-block buffer (TB buffer) is a register file that stores current image data.

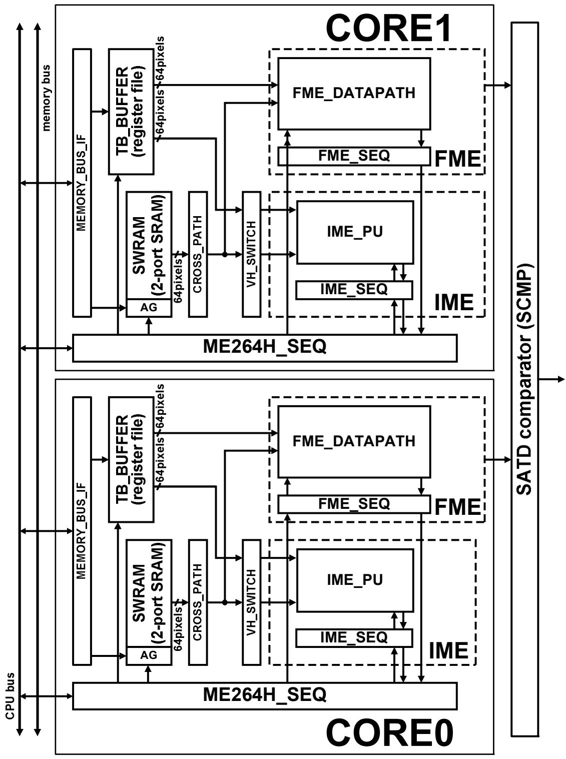

Fig. 7. A block diagram of the proposed ME processor core.

B. IME Architecture

Fig. 8 portrays the IME block diagram and the detail drawing of the reconfigurable ring-connected systolic array (RRSA) inside it. The RRSA is comprised of eight sub block systolic arrays (SBSAs) and registers for vertical shift (REG_VS).

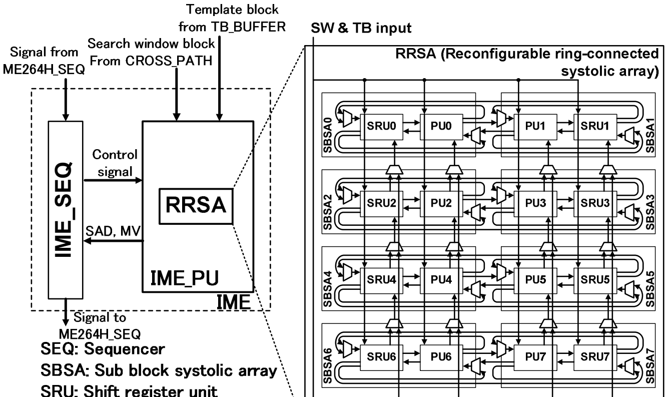

Fig. 8. A block diagram of the IME architecture.

By changing a combination of SBSAs, the size of a processing unit (PU) arrays can be configured, ranging from 8 × 8 to 16 × 32. Hence, SAD calculations in any block types (frame / field MBAFF modes, and 8 × 8 / 16 × 8 / 8 × 16 / 16 × 16 / 16 × 32 block sizes) can be performed by the RRSA architecture.

The PU and shift register unit (SRU) have a direct access path for initial load and buffered access path through REG_VS for a full search. Furthermore the PU and SRU can search at continuous points with less reload of pixels from the SWRAM by pixel reuse. The SBSA has an inter- and intra-connection (ring connection) with multiplexers to exchange 1 × 8 pixels, along the horizontal direction. In addition, the SBSA has vertical connections to transfer 8 × 1 pixels to the upper SBSA. The function of the PU is to calculate a block SAD. Meanwhile, the function of the SRU is to buffer search-window pixels for the PU. Using these configurations, a vertical shift operation, left shift operation, right shift operation, and 8 × 8 rectangular-block loading (initial load) can be supported. So, a block is searched along the left, right, and lower directions on demand without pixel reload. In other words, the flexible systolic array enables efficient one-dimensional search and FS. The ring-shaped connection is thus effective for the pixel reuse in the one-dimensional search and full search because the connection does not dispose pixels in search window.

C. FME Architecture

The FME architecture is comprised of FME quarter-pel generators (FME_SPGs), FME processing units (FME_PUs), and FME sequencer (FME_SEQ), as shown in Fig. 9. A pair of SME_SPG and SME_PU is able to process 8 × 8 pixels. The FME_SPGs access the SWRAM only while the IME_PUs do not access it. In consideration of this access restriction, the FME architecture has a 512-way parallelism. With the proposed architecture, an NSA with 16 × 16 pixels is carried out in 85 cycle times, and the total access time of the proposed FME algorithm is 404.

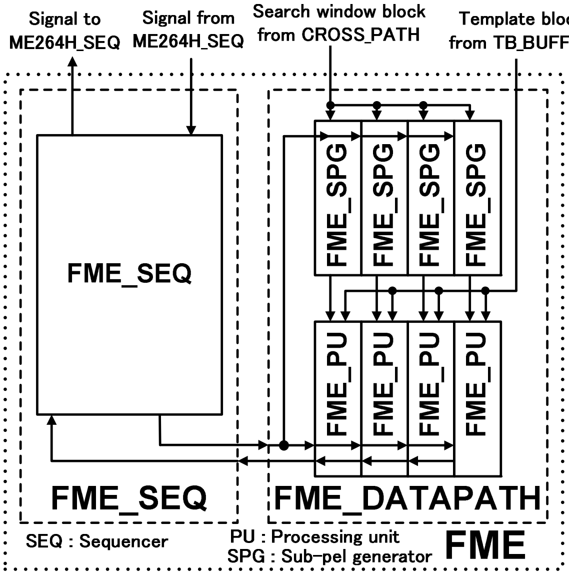

Fig. 9. A block diagram of the FME architecture.

D. Candidate-MVs Reduction Architecture

The candidate motion vectors reduction architecture is illustrated in Fig. 10. The SATD comparator corresponds to “SCMP”, which refines motion vectors. The SCMP is comprised of an Enable buffer, SATD buffer, Motion vector buffer, Reference-selector, List-selector, Frame/field-selector, and Block-size-selector. Each selector has a comparator and decoder. All registers of Enable buffer is initially set valid, and subsequently part of registers are set invalid through the each selector.

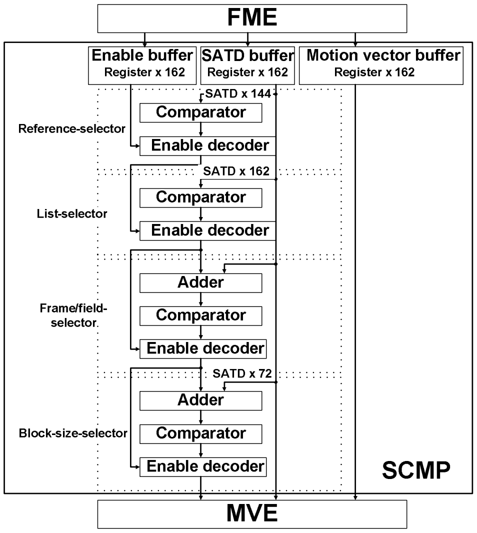

Fig. 10. A block diagram of the SCMP architecture.

IV. VLSI Implementation

Fig. 11 is a chip layout of the ME processor core in a 90-nm CMOS technology. A 410-Kb SWRAM on one core can handles one reference frame. The proposed architecture was designed on a standard cell base (logic synthesis). The processor occupies 6.0 × 6.0 mm^2. The maximum operating frequency is 150 MHz at a nominal supply voltage of 1.0 V. The chip specification is summarized in TABLE I. A 100MHz operation supports two reference frames at a resolution of a 30-fps HDTV (1920 × 1080 interlace), in which the power is 160.1mW at 1.0 V.

Fig. 12 exhibits the power consumption between the conventional and proposed ME methods. The proposed processor dramatically reduces the power consumption to 7.6 % of the conventional method.

TABLE I. Chip specification

Technology	90nm
Chip size	6.0 x 6.0 mm^2
Voltage	1.0 V
Max frequency	150 MHz @ 1.0V
Search range	±128 × ±64
Memory size	SWRAM: 410 kb / one core
Power	160.1 mW @100MHz 1.0V

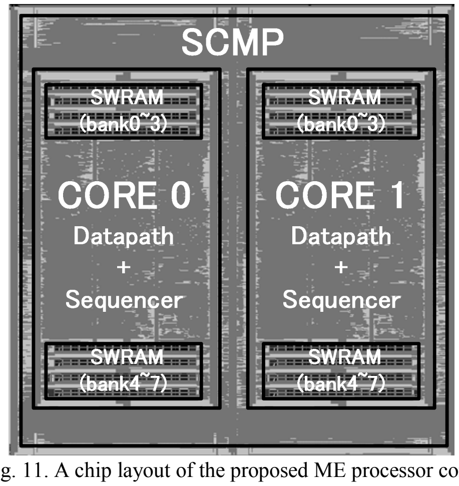

Fig. 11. A chip layout of the proposed ME processor core.

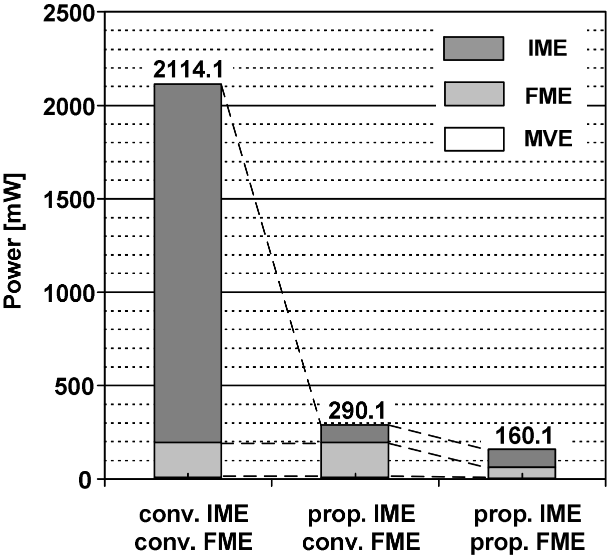

Fig. 12. Power reduction by the proposed ME processor core.

V. Summary

We described the novel algorithm and architecture for the IME, FME, and candidate motion vectors reduction. FME search strategy is proposed to reduce the workload and power in FME. The proposed FME architecture executes the proposed algorithm in 464 cycle times. The candidate-MVs reduction algorithm and SCMP architecture are proposed to save the workload in MVE by reducing the number of MVs from 162 to 14.

We designed an H.264/AVC main profile motion estimation processor core for MBAFF encoding. 16 × 16, 16 × 8, 8 × 16, and 8 × 8 block sizes are supported for an HDTV resolution video (1920 × 1080 interlace). The processor provides fractional-accuracy motion vectors in real-time operation. The proposed processor reduces power consumption to 7.6 % of the conventional method.

References

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

[2] Y. Murachi, J. Miyakoshi, M. Hamamoto, T. Iinuma, T. Ishihara, F. Yin, J. Lee, H. Kawaguchi, and M. Yoshimoto, "A sub 100 mW H.264 MP@L4.1 Integer-pel Motion Estimation Processor Core for MBAFF Encoding with Reconfigurable Ring-connected Systolic Array and Segmentation-free, Rectangle-access Search-window Buffer," IEICE Trans. Electron., April 2008 (in press).

[3] K. Kumagai “Systen-in-silicon architecture and its application to H.264/AVC motion estimation for 1080HDTV”, Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International.

[4] S. Warrington “Scalable high-throughput architecture for H.264/AVC variable block size motion estimation”, Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on.

[5] ISO/IEC | ITU-T VCEG, Fast Integer Pel and Fractional Pel Motion Estimation for JVT, JVT-F017, 2002