3-2

An 800-μW H.264 Baseline-Profile Motion Estimation Processor Core

Takahiro Iinuma, Junichi Miyakoshi, Yuichiro Murachi, Tetsuro Matsuno, Masaki Hamamoto, Tomokazu Ishihara, Hiroshi Kawaguchi, Masahiko Yoshimoto
The Graduate School of Science and Technology
Kobe University
Kobe, Hyogo, Japan
e-tak@cs28.cs.kobe-u.ac.jp

Masayuki Miyama
The Graduate School of Science and Technology
Kanazawa University
Kanazawa, Ishikawa, Japan
miyama@t.kanazawa-u.ac.jp

***Abstract*— This paper describes an 800-μW H.264 baseline-profile motion estimation processor for portable video applications. It features a VLSI-oriented block partitioning strategy, a reconfigurable SIMD/systolic-array datapath architecture and a power-efficient novel SRAM circuit with a segmentation-free and horizontal/vertical accessibility. The proposed architecture can reconfigure datapath to either an SIMD or systolic array depending on processing flow. The segmentation-free access means concurrent accessibility to arbitrary consecutive pixels. The processor supports all the seven kinds of block modes, and can handle three reference frames for a VGA (640 × 480) 30-fps to QCIF (176 × 144) 15-fps sequences with a quarter-pixel accuracy. It integrates 3.3 million transistors, and occupies 2.8 × 3.1 mm^2 in a 130-nm CMOS technology. The proposed processor achieves a power of 800 μW for QCIF 15-fps with one reference picture.**

I. INTRODUCTION

H.264/AVC (hereafter, H.264) [1] provides two times higher coding efficiency than the previous standards such as H.263 and MPEG-4, while a computational cost of H.264 reaches a dozen-fold workload of the previous standards. This is because motion estimation in H.264 employs seven kinds of motion compensations whose block sizes range from 16 ×16 to 4 × 4 pixels with a quarter-pel motion vector accuracy. Therefore, the H.264 algorithm requires complicated hardware, which causes power increase.

This paper describes a low-power motion estimation processor for the H.264 baseline profile with a VLSI-oriented block partitioning algorithm, SIMD/systolic-array architecture and flexible-access SRAM for video processing.

II. VLSI-ORIENTED ALGORITHM

The conventional algorithm named UMHexagonS (hybrid unsymmetrical-cross multihexagon-grid search) [2] is an eminent motion estimation algorithm reducing an average workload by using an early-termination, and thus adopted in the H.264 joint model JM [3]. This is, however, not suitable for hardware implementation because it can not reduce a worst case workload, and performs adaptive motion searches that can not carry out a next step until a current step is completed by branch processes.

A. Block-Matching Methods

The motion estimation processor core (ME) employs the one-dimensional diamond search (1D-DS) [4] for an integer-pel motion search which is a kind of gradient methods. Its flowchart is shown in Fig. 1. Step 1 of the integer-pel search is an initial vector search where an initial motion vector is chosen from 0-vector and vectors of the neighboring MBs. At Step 2, a search direction is selected out of four points (diamond shape) surrounding the initial vector, according to the minimum sum of absolute difference (SAD). Then, at Step 3, a one-dimensional search is carried out toward the search direction, where eight SADs are computed at eight pixels and the vector that has the smallest SAD is selected as the minimum vector. Next, the minimum vector is compared with the initial vector. If the vectors are not equal, the initial vector is reset to the minimum vector, and the 1D-DS are iterated. Alternatively, if it is equal, the 1D-DS is terminated. The motion vector which indicates the smallest SAD is obtained in this search. The 1D-DS reduces a worst-case workload to 18% of UMHexagonS. Furthermore, it reduces branch processes to 51% by decreasing the number of steps compared to the UMHexagonS which has seven steps. In other words, the 1D-DS is suitable for VLSI implementation, since it can reduce the worst-case workload and operating cycles by less branch processes.

Next, the 35-point full search method is adopted to obtain a motion vector of a quarter-pel accuracy. This method performs a full search in a narrow search area (±0.5 × ±0.75) whose center is a resultant vector in the 1D-DS.

B. Block Partitioning Strategy

It is important to choose the optimum mode and its optimum vector as early as possible to suppress computation

0-7803-9735-5/06/$20.00 ©2006 IEEE

cost. In this subsection, the fast block mode decision algorithms for the integer-pel search method (1D-DS) and sub-pel search method (35-point full search) are proposed.

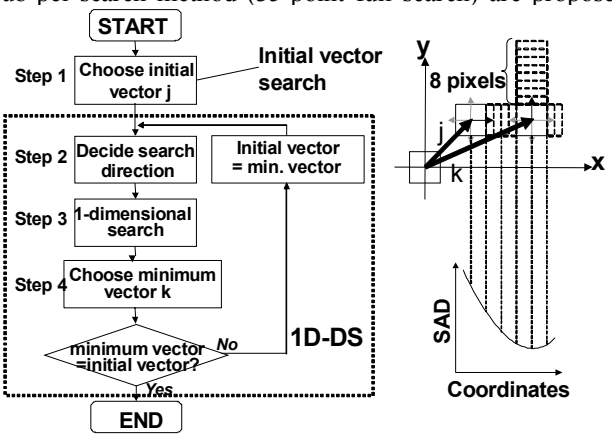

Figure 1. Flowchart of integer-pel search with initial vector search and 1D-DS method

1) Fast Search for Larger Blocks (FSLB) with Integer-Pel Accuracy

The fast search for larger blocks (FSLB) is the proposed algorithm for an integer-pel motion search. The four rectangles in Fig. 2 indicate block segmentations in Mode 2 and Mode 3. At first, the integer-pel motion search finds four motion vectors (MVs) in the four blocks using the 1D-DS method. The four blocks are top Block A in Mode 2 (MV_A), bottom Block B in Mode 2 (MV_B), left Block C in Mode 3 (MV_C), and right Block D in Mode 3 (MV_D). Then, the MV in Mode 1 (MV_{Mode1}) should be founded. The algorithm supposes that the MV_{Mode1} is inside the quadrilateral pointed by MV_A, MV_B, MV_C, and MV_D, as depicted in Fig. 2. In concrete terms, MV_{Mode1} is selected from the set of eight candidate MVs (CMVs), which means that the integer-pel motion search in Mode 1 is carried out only around the eight points. Here, the CMVs are a predicted vector, MV_A, MV_B, MV_C, MV_D, and three vectors of each average between MV_A and MV_B, MV_C and MV_D, and all four vectors.

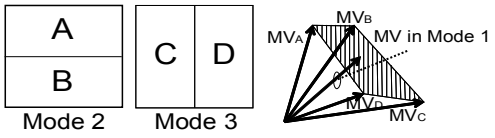

Figure 2. Optimum MV for mode 1 by FSLB.

2) Fast Search for Smaller Blocks (FSSB) with Sub-pel Accuracy

Next, we proceed to the sub-pel motion search with the results of the FSLB. That is, the 35-point full searches are carried out around the five integer-pel MVs in Modes 1-3 (MV_A, MV_B, MV_C, MV_D, MV_{Mode1}). The salient feature in the proposed algorithm for the sub-pel motion search (FSSB: fast search for smaller blocks) is that the sub-pel motion search even in Modes 4-7 is simultaneously performed during the motion search in Modes 1-3. This implies that there is no additional computation cost to obtain the sub-pel MVs in Modes 4-7.

C. Simlation Results

We compared the proposed algorithms with the conventional algorithms that have been adopted in the JM, namely full search (FS) and UMHexagonS. The simulation conditions are as follows; baseline-profile, CIF (352 × 288 pixels) resolution, 30-fps frame-rate, three reference frames, no R-D optimization, and no rate control. The R-D curves in Fig. 3 indicate that, in a low-bitrate region, the picture quality degradation in the proposed algorithm is within 0.10 dB compared with Conventional-1. On the other hand in a high-bitrate region, the picture qualities are almost equal. As shown in Fig. 4, the worst-case workload in Proposed-1 with FSLB and FSSB is 10.5% of that of Conventional-1. Furthermore, by using the 1D-DS method, the worst-case workload of Proposed-2 is lowered to 4.5% of the Conventional-1, allowing a low power VLSI implementation.

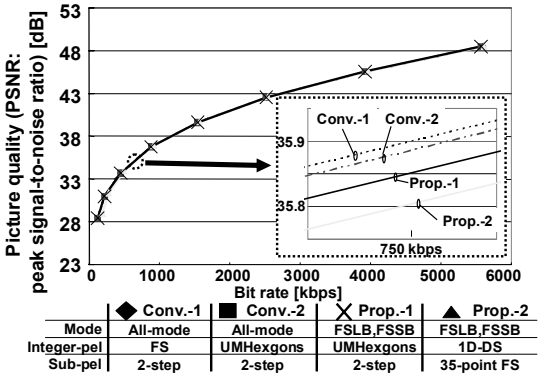

Figure 3. R-D Curves in the conventional and proposed algorithms.

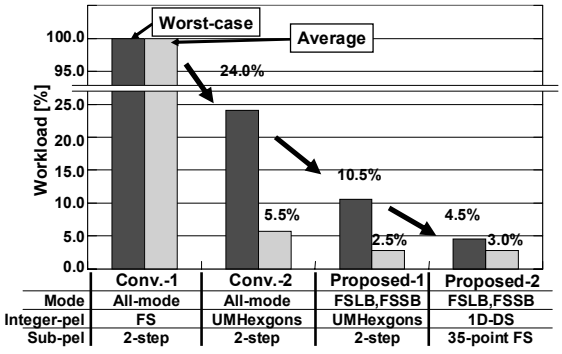

Figure 4. Workload comparison of the algorithms.

III. ADAPTIVE SIMD/SYSTOLIC-ARRAY ARCHITECTURE

The architecture of the H.264 motion estimation processor (ME) to which the proposed algorithm is implemented is illustrated in Fig. 5. An integer-pel motion estimation (IME) processor performs the initial vector search and the 1D-DS method with an integer accuracy. The sub-pel motion estimation (SME) processor executes the 35-

point full search with a quarter-pel accuracy. Two three-port (two read ports + one write port) SRAMs are also implemented as image data caches. One is for reference pictures (SWRAM: search window RAM), and the other is for a current picture (TBRAM: template block RAM). The IME and SME processors share these two SRAMs. By adaptive assignment of the read ports to the two processors, they can operate in parallel.

A novel architecture which can reconfigure datapath to either SIMD or systolic array (SA) is proposed for the IME processor. The proposed architecture named SIMD/SA in the IME in Fig. 5 is comprised of two IPUs (integer-pel processing units) including an eight-way SIMD module each. The SIMD/SA works as a low operating-cycle 16-way SIMD [5] during random block-matching such as the initial vector search. In that case, the MUX routes outputs of read-port 1 in both the SRAMs to IPU1. The both IPU 0 and IPU 1 are connected to the read ports of both TBRAM and SWRAM, and thus both IPU0 and IPU1 operate in parallel. On the other hand, it changes the datapath to a little memory-access SA [5] during a serial block-matching such as the 1D-DS method. The MUX routes IPU0 outputs to IPU1. The IPU 0 forwards TBRAM and SWRAM data to the IPU 1, and both IPUs perform the 1D-DS.

Figure 6 illustrates timing schedules of ME processors of the conventional ones and the proposed. In the 16-way SIMD mode, the IME occupies both the read ports #0 and #1 to perform a parallel processing during the initial vector search, so that the IME and SME can not operate in parallel as shown in Fig. 6. Furthermore, when the processor configure SA mode, though the IME and SME can operate in parallel to execute the 1D-DS and 35-point full search, it takes 1098 cycles to complete the process. Figure 6 (c) shows the case of the proposed SIMD/SA configuration which demonstrates the minimum cycle counts, 878 cycles. The concurrent operation by the proposed SIMD/SA architecture attains the minimum power since the required cycle is less, and thus an operating frequency can be lowered. The operating frequencies in three types of configuration are shown in Fig. 7. It is seen that 25% to 34% reduction of the operating frequency in the proposed is achieved with the minimum count of the required pixels compared with the SIMD and the SA configuration.

IV. Segmentation-Free- and Horizontal/Vertical-Access SRAM Architecture

A parallel datapath and an adaptive motion search generally require a concurrent access to arbitrary consecutive pixels on a horizontal/vertical line. The conventional SRAM techniques can not achieve these flexible accesses without dividing the SRAM [4]-[5] which increases the power due to extra X-decoders in each divided SRAM. To solve this issue, we propose a novel SRAM architecture with a spirally-connected local-wordline (LWL) select line and a multi-selection scheme in global-wordlines (GWL), which is suitable for the parallel video processing. Figure 8 shows an access method of the proposed SRAM. The X-decoder which is not shown in the figure, selects multiple global-wordlines, and then the LWL are selected by local-AND of the selected GWL and the selected LWL select line (LWLS) which are spirally connected.

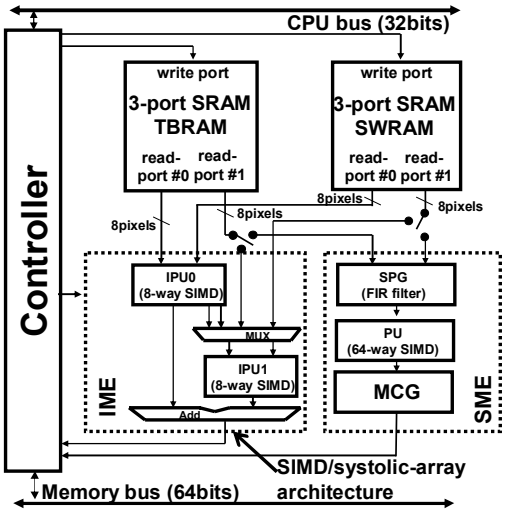

Figure 5. A block diagram of the proposed ME processor core incluing SIMD/systolic-array datapath.

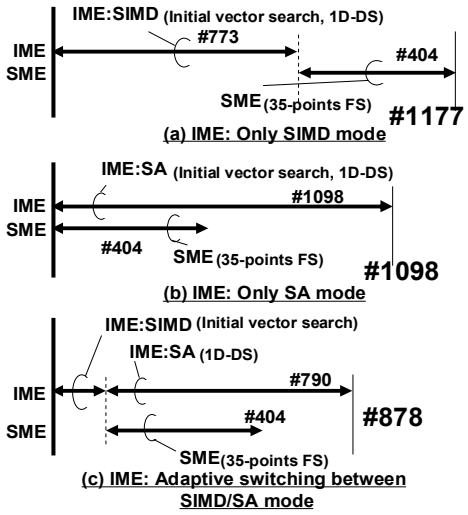

Figure 6. Comparison of timing schedule for the entire motion estimation process among three kinds of IME architecture

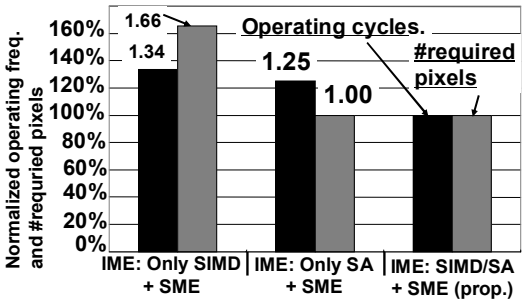

Figure 7. Operating cycles and required pixel counts for the entire motion estimation process

Consequently, the proposed SRAM can eliminate extra X-decoder by using LWL selector, so that is reduces the amount of area and power consumed by extra X-decoder. The power comparison between the conventional and proposed SRAM is shown in Fig. 9. It can be seen that 42% power reduction is attained.

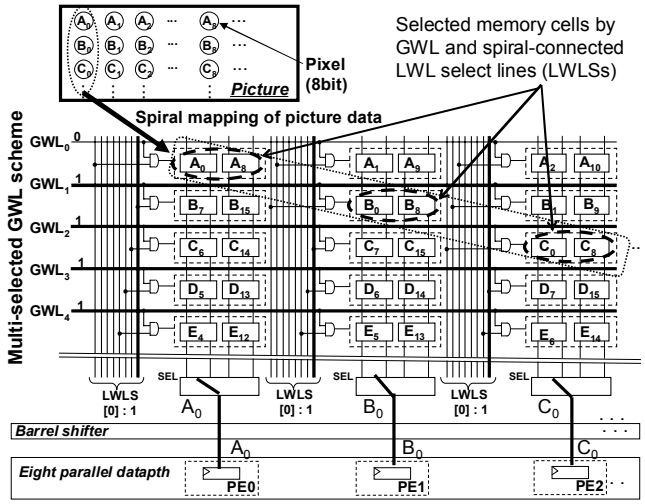

Figure 8. An access method of the proposed SRAM. (accessing A_0 to I_0 pixels in a vertical direction)

V. Power Estimation and VLSI Implementation

A comparison of the power consumption for the algorithms, the architectures, and the SRAMs is shown in Fig. 10. The power consumption has been reduced 75% compared with the conventional one whose power was estimated by technology scaling and resolution adjustment from Ref. [6]. This is achieved by exploiting the VLSI-oriented algorithm, the SIMD/SA datapath as IME architecture and the SRAM circuit with the segmentation-free and horizontal/vertical accessibility. The power consumption for QCIF (176 × 144) 15-fps is 800 μW in 130-nm CMOS technology. The chip photograph and layout of the ME processor are shown in Fig. 11. The area is 2.8 × 3.1 mm^2.

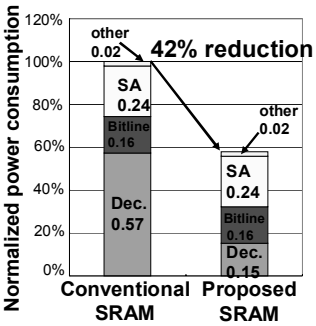

Figure 9. Power conparison of SRAMs

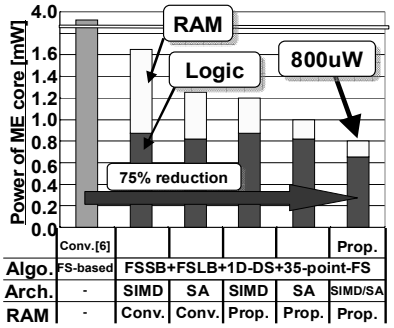

Figure 10. Power of the ME processor (processing QCIF 15-fps)

VI. Summary

This paper proposed an 800-μW H.264 baseline profile motion estimation processor core. The processor supports all the seven kinds of block modes, and can handle three reference frames for a VGA (640 × 480) 30-fps to QCIF (176 × 144) 15-fps sequences with a quarter-pixel accuracy. The processor features the adaptive block-matching algorithm and novel block partitioning strategy to reduce both the worst-case workload and operating frequency. By applying the SIMD/SA architecture to the integer-pel search, pixels read from SRAMs and operating cycles for motion estimation can be lowered. Furthermore the segmentation-free- and horizontal/vertical-access SRAM reduces the power and area by eliminating extra X-decoder. It is expected to be applicable to an SoC with H.264 codec function for portable video terminals.

Acknowledgment

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.

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

[3] JM 8.5, http://iphome.hhi.de/suehring/tml/.

[4] Yuichiro MURACHI, Koji HAMANO, Tetsuro MATSUNO, Junichi Miyakoshi, Masayuki MIYAMA, and Masahiko YOSHIMOTO,"A 95 mW MPEG2 MP@HL Motion Estimation Processor Core or Portable High-Resolution Video Application", IEICE Trans. Fundamentals, VOL.E88-A, NO.12, pp.3492-3499, December 2005.

[5] Junichi MIYAKOSHI, Yuichiro MURACHI, Koji HAMANO, Tetsuro MATSUNO, Masayuki MIYAMA, and Masahiko YOSHIMOTO, "A Low-Power Systolic Array Architecture for Block-Matching Motion Estimation" , IEICE Trans. Electoronics, VOL.E88-C, NO.4, pp.559-569, April 2005

[6] Y-W. Huang, T-C. Chen, C-H.Tsai, C-Y.Chen, T-W. Chen, C-S. Chen, C-F. Shen, S-Y. Ma, T-C. Wang, B-Y. Hsieh, H-C. Fang, L-G. Chen, "A 1.3TOPS H.264/AVC Single-Chip Encoder for HDTV Applications", IEEE International Solid-state Circuit Conference, pp128-129, January 2005

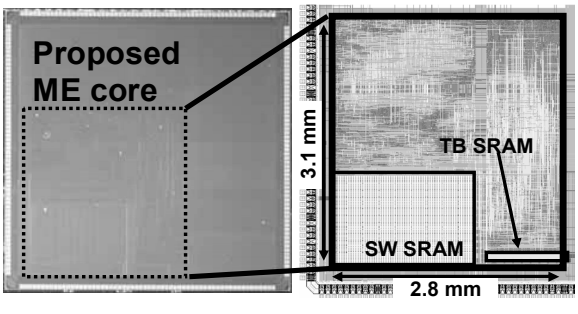

Figure 11. Photograph and layout of the propsoed ME processor