A Sub-mW MPEG-4 Motion Estimation Processor Core for Mobile Video Application

A Sub-mW MPEG-4 Motion Estimation Processor Core for Mobile Video Application

Masayuki Miyama, *Member, IEEE*, Junichi Miyakoshi, Yuki Kuroda, Kousuke Imamura, Hideo Hashimoto, *Member, IEEE*, and Masahiko Yoshimoto, *Member, IEEE*

***Abstract*—This paper describes a sub-mW motion estimation processor core for MPEG-4 video encoding. It features a gradient descent search (GDS) algorithm that reduces required computational complexity to 15 MOPS. The GDS algorithm combined with a sub-block search method upgrades picture quality. The quality is almost equal to that of a full search method. An SIMD datapath architecture optimized for the algorithm decreases a clock frequency and supply voltage. A dedicated three-port SRAM macro for image data caches of the processor is newly designed to reduce power consumption. It has been fabricated with 0.18-μm five-layer metal CMOS technology. The VLSI processing QCIF 15-f/s video consumes 0.4-mW power at 0.85-MHz clock frequency with 1.0-V supply voltage. It is applicable to mobile video applications.**

***Index Terms*—Gradient-based method, low power, motion estimation, MPEG, SIMD.**

I. Introduction

A MOBILE terminal by which people can visually communicate with others continues to gain popularity. To realize an ultra-low-power and high-quality real-time MPEG-4 video codec in the terminal, a highly efficient motion estimation processor is essential.

The motion estimator with a conventional full search (FS) shares more than 70% of the total computational complexity in the MPEG-4 encoder. The FS algorithm requires about 200-MOPS computation complexity for QCIF 15-f/s motion estimation. Many MPEG codec LSIs that perform motion estimation with the FS method have been reported [1], [2]. Power consumption of a motion estimation processor using 0.18-μm technology is about 20 mW. This power consumption is prohibitively large for an IP core in the mobile terminal.

Many fast motion estimation algorithms has been investigated [3]. Hierarchical algorithm predicts an approximate motion vector in a coarse resolution image, and refines it in a finer resolution image. The Three Step Search (TSS) algorithm is the most popular one as a fast motion estimation algorithm. The Cote algorithm is known as a gradient-based method and faster than the TSS algorithm. Low-power motion estimation circuits with these methods have been reported [4]–[6]. Unfortunately, predicted picture quality produced by these algorithms is degraded for high motion video because of a local minimum problem.

This paper describes a sub-mW motion estimation processor core (ME core) for MPEG-4 video encoding which solves these problems. Section II describes the GDS algorithm. The GDS algorithm attains higher picture quality than the other fast motion estimation algorithms, even though the computational complexity is quite low. Section III describes an SIMD datapath architecture optimized for the GDS algorithm. Section IV describes a low-power image data cache dedicated for the ME core. Section V describes evaluation results of the ME core implemented with 0.18-μm technology. The ME core processing QCIF 15-f/s video consumes only 0.4 mW at 0.85 MHz under 1.0-V condition. Section VI concludes this paper.

II. Algorithm

A. GDS Algorithm

The GDS algorithm shown in Fig. 1 is a gradient-based method using the steepest descent method [7]. The criterion of a distortion function used in the GDS algorithm is a mean square error (MSE) of a macro block (MB) indicated by a motion vector. One of the four motion vectors shown in Fig. 1 is decided as a start vector. Next, a search direction is calculated by differential coefficients of the function at the point indicated by the vector. The MBs are evaluated toward the search direction step by step in one pixel width until the MSE increase. This search process is called a one-dimensional search (1-DS). A search direction is calculated again at the point indicated by the vector that has the lowest MSE in the previous 1-DS. This procedure is repeated several times to reach the minimum distortion. An integer-pel motion vector whose MSE is the lowest among search points until now is a temporal solution. This is followed by a 3×3 neighbor half-pel search (3×3-NHS), at the position indicated by the temporal solution. Hence, we can reach a final solution with half-pel accuracy.

The GDS algorithm introduces a hierarchical search method and a lump search method, not to fall into a local minimum. The hierarchical search method generates multi-resolution images. It predicts a large-scale motion vector in a coarse resolution layer and refines the vector in a finer resolution layer. The lump search method fixes the number of points to evaluate in a 1-DS, regardless of the MSE increase.

Manuscript received December 16, 2003; revised April 6, 2004. This work was supported by STARC (Semiconductor Technology Academic Research Center) and the VLSI Design and Education Center (VDEC), University of Tokyo.

The authors are with the Faculty of Engineering, Kanazawa University, Kanazawa 920-8667, Japan (e-mail: miyama@t.kanazawa-u.ac.jp).

Digital Object Identifier 10.1109/JSSC.2004.831461

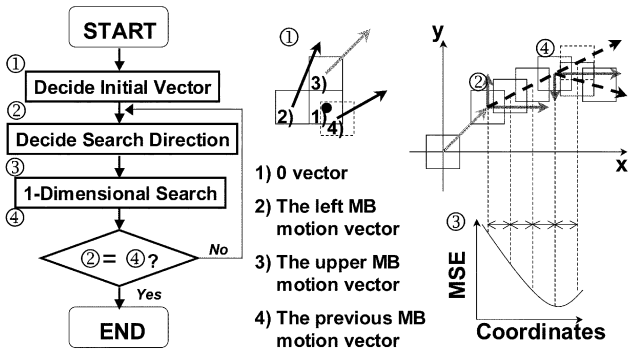

Fig. 1. GDS algorithm.

Fig. 2. Sample pictures.

B. Optimization for VLSI Implementation

The GDS algorithm was optimized for VLSI implementation about the following items:

- search range;
- the number of hierarchies;
- the number of times to iterate a one-dimensional search;
- the number of points in a lump search;
- search direction rounding.

The search range should be minimized because search window (SW) RAM accounts for a significant amount of power consumption. The minimum search range maintaining picture quality within a mean PSNR drop of -0.1 dB was obtained by simulation. Simulation conditions are summarized as:

- sample pictures (Fig. 2);
  - salesman (sale);
  - susie (ssie);
  - mobile and calendar (mbcl);
  - flower garden (flow);
  - bus (bus1);
- resolution and frame rate: QCIF 15 f/s, CIF 30 f/s;
- the number of frames: 75 (QCIF 15 f/s), 150 (CIF 30 f/s);
- forward, half-pel prediction.

The simulation result indicates that the minimum search range maintaining a mean PSNR drop of -0.1 dB is $[-16 : +15.5]$ pixels.

The optimum number of hierarchies was obtained by simulation. The simulated number of hierarchies was 1 (GDS_mb_h1), 2(GDS_mb_h2), and 3(GDS_mb_h3). The simulation result represents that the GDS_mb_h1 yields the best picture quality, even though its computational complexity is the lowest. It has been confirmed that the hierarchical method has a good effect on the motion estimation using an interlaced format such as CCIR601 [7], [8]. The hierarchical method for CIF and QCIF has no effect because they are noninterlaced formats. The GDS_mb_h1 algorithm does not require extra memory space including the hierarchical image. A VLSI based on the algorithm does not contain a circuit to generate low-resolution images.

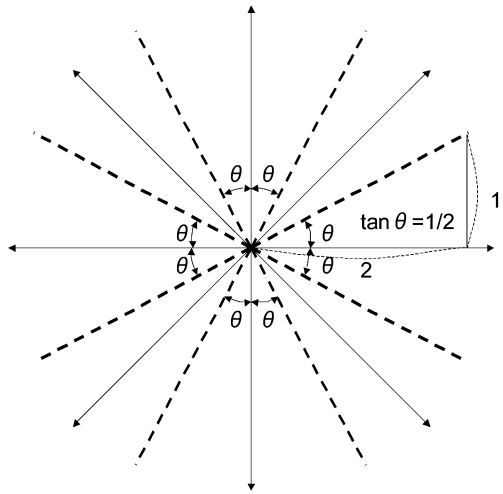

Fig. 3. Direction rounding.

The minimum number of times to iterate a one-dimensional search and the minimum number of points in a lump search were investigated by simulation. They were obtained on the condition that the picture quality was maintained within a mean PSNR drop of -0.1 dB. The simulation results show the numbers are 2 and 3, respectively.

The search direction is rounded off to one of 8 directions to simplify address generation. A boundary definition to round off the search direction is depicted in Fig. 3. Simulation results show that the rounding operation upgrades the picture quality. The horizontal and vertical motion is often the preferred motion direction. The rounding operation probably prevents the algorithm from choosing a wrong search direction.

C. Sub-Block Search Method

The GDS algorithm is combined with a sub-block (SB) search method to enhance picture quality. The SB search method is illustrated in Fig. 4. First, an MB (16×16 pixels) indicated by the start vector is divided into 4 SBs (8×8 pixels). Next, the 1-DS for each SB are executed toward a search direction indicated by its own differential coefficients. As a result of the 1-DS, four vectors are obtained as temporal solutions. An MSE of an MB indicated by each vector is calculated. A final motion vector is chosen from five vectors obtained by an MB search (V) and four SB searches (Va, Vb, Vc, Vd). The GDS algorithm with the SB search decides a motion vector that indicates an MB having the lowest MSE during the MB search and the SB search. Therefore, the algorithm always attains higher or equal picture quality comparing with the original algorithm.

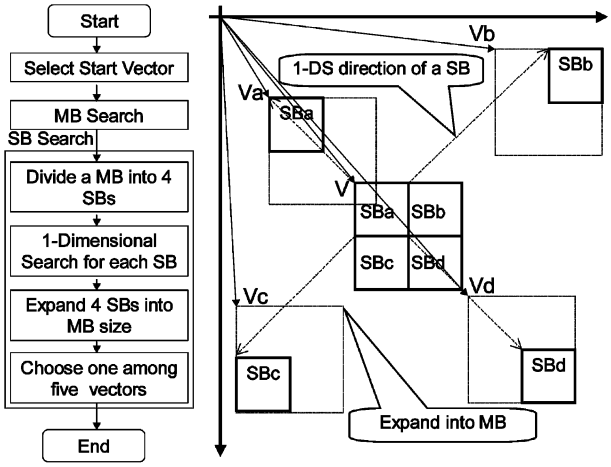

Fig. 4. Sub-block search method.

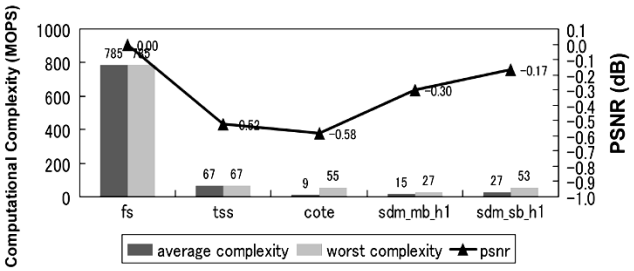

Fig. 5. ME algorithms comparison (QCIF 15 f/s).

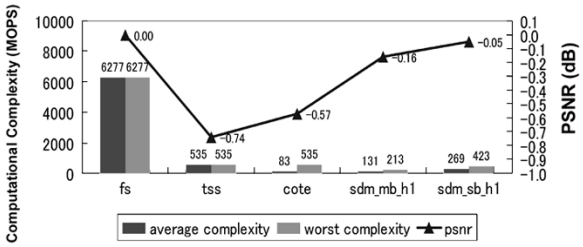

Fig. 6. ME algorithms comparison (CIF 30 f/s).

D. Simulation Results

The GDS algorithm and the other algorithms were simulated to analyze computational complexity and picture quality. The simulation conditions were the same as above. The algorithms simulated here were as follows:

- FS;
- TSS;
- Cote;
- GDS_mb_h1;
- GDS_sb_h1 (GDS_mb_h1 combined with a sub-block method).

The FS, TSS, and Cote algorithms search integer-pel points first, then 8 half-pel points surrounding the minimum integer-pel point. The distortion function of the FS, TSS, and Cote algorithm is a mean absolute error. The VLSI based on each algorithm usually adopts these search methods. The search range is $[-16 : +15.5]$.

Figs. 5 and 6 depict the relation between the computational complexity and PSNR obtained by the simulation. They represent that the GDS algorithm attains both higher picture quality and lower complexity than the TSS algorithm. The GDS algorithm also attains higher picture quality than the Cote algorithm, and the average complexity is a little bit higher than that of the Cote algorithm. The worst complexity is lower than that of the Cote algorithm owing to the optimization as mentioned above.

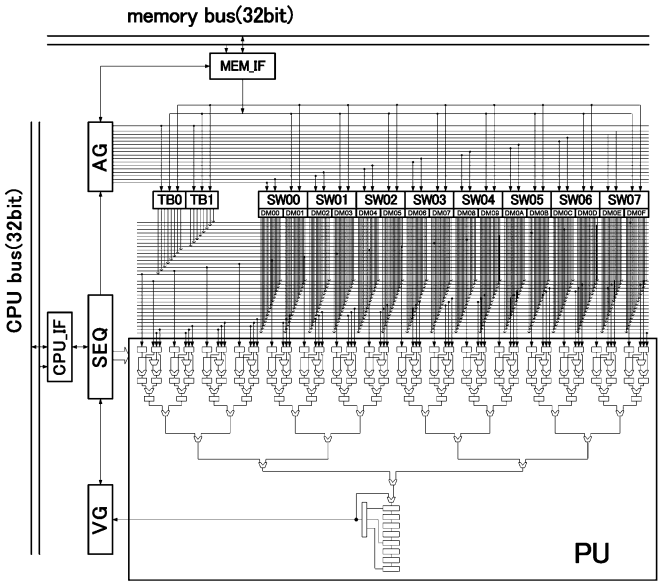

Fig. 7. ME core block diagram.

III. Architecture

Fig. 7 shows a block diagram of the ME core. The ME core is divided into an SIMD datapath and a control part. The SIMD datapath contains two template block (TB) RAMs, eight search window (SW) RAMs, and a processor unit (PU). The TB RAM has three-port access capability (2 read/1 write) and 64 words by 64 bit configuration. The SW RAM has three-port access capability (2 read/1 write) and 512 words by 8 bit configuration. The PU contains 16 processor elements (PEs), an adder tree (AT), and an accumulator (ACC). The control part consists of a sequencer (SEQ), a vector generator (VG), and an address generator (AG).

The TB buffer and the SW buffer have 16 read ports each. They keep supplying pixel data to 16 PEs so that pipeline operation can be maintained continuously. The TB buffer, the SW buffer, and 16 PEs are connected by a cross path (CP) to sort pixel data. The pipelined PE executes four operations for one pixel calculation per one clock cycle; so 16 PEs can execute 64 operations for one row of an MB per one clock cycle. The 16 PEs are followed by the AT, which completes the summation. Thus, the ME core can efficiently execute the GDS algorithm in the highly parallel and pipelined way. The performance of the ME core operating at 13.5 MHz is 864 MOPS. The ME core can operate at low frequency and voltage, giving very low power consumption.

A. Memory Configuration and Data Mapping

Fig. 8 shows a configuration of the SW buffer. It also illustrates a data mapping method. The SW buffer is configured by

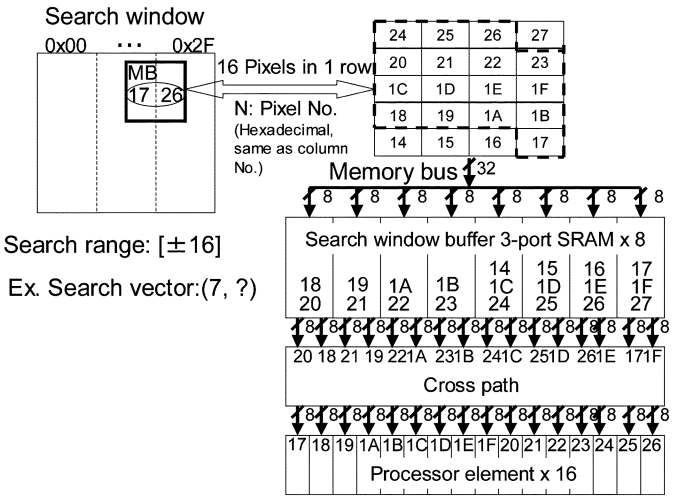

Fig. 8. Memory data mapping.

eight SRAMs. The SRAM has two read ports and one write port. The memory bus with 32-bit width feeds image data to the SW buffer. Adjacent pixels are stored into adjacent SRAMs as illustrated in Fig. 8. Hence, the SW buffer can feed 16 pixels corresponding to one row of an MB to 16 PEs simultaneously. The cross path sorts 16 pixels from the SW buffer, according to the sequence of 16 pixels from the TB buffer. When the motion estimation for the current MB is being executed, additional pixels to the current SW are written to the SW buffer concurrently, to prepare the motion estimation for the next MB. A required bandwidth processing QCIF 15-f/s video is 1.14 MB/s. It is 4.56 MB/s for CIF 30-f/s video.

B. Processing Element

Fig. 9 shows a block diagram of the PE. The PE can calculate both an MSE and differential coefficients. Fig. 9(a) depicts the PE operation for the MSE calculation. The PE receives each pixel data from the TB buffer and the SW buffer. Fig. 9(b) depicts the PE operation for calculating the differential coefficient in x direction. In this case, the PE receives one pixel data from the TB buffer. The right, center, and left pixel data are received from the SW buffer. Fig. 9(c) depicts the PE operation for calculating the differential coefficient in y direction. The PE can also execute the 3×3-NHS by using a half-pixel blender (HPB). Fig. 9(d) depicts the PE operation for a 3×3-NHS calculation. The HPB generates half-pel data by filtering operation among integer-pel data.

Fig. 10(a) and (b) shows a timing chart of an MSE calculation and that of the 3×3-NHS, respectively. Texts appended to data in Fig. 10 correspond to those appended to pixels in Fig. 11. Fig. 10(a) indicates that 16 pixels in one row can be calculated per one clock cycle by introducing a pipeline dataflow. Then one MB can be calculated per 16 clock cycles.

The 3×3-NHS is illustrated in Fig. 10(b). Eight surrounding integer-pels are serially loaded into PEs for an MSE calculation of each half-pel MB. The AT and the ACC accumulate the result for each MB with serially connected nine registers (reg0-8) corresponding to eight half-pel MBs. These operations are iterated 16 times.

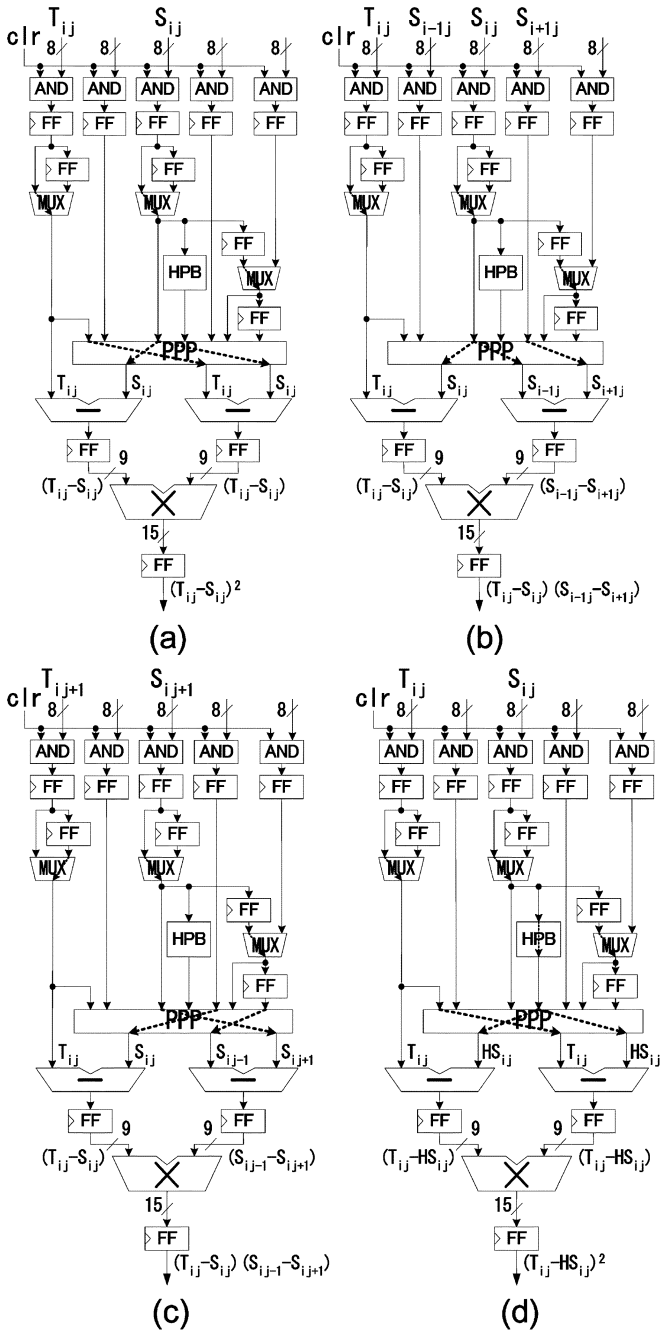

Fig. 9. PE block diagram.

C. Hierarchical Sequencer

Fig. 12 shows a block diagram of the SEQ. The SEQ consists of three sub-sequencers. The P_SEQ controls a sequence of instructions stored in the INST_REG. The P_COUNTER stores the instruction number that is currently executing.

The I_SEQ controls execution of the instruction given by the P_SEQ. There are two types of instruction, which are a basic instruction and a macro instruction. The basic instruction includes a calculation of the evaluation value, the x differential coefficient, the y differential coefficient, and the 3×3-NHS. A combination of the basic instructions makes the macro instruction such as the initial vector decision and the 1-DS. The I_SEQ breaks a macro instruction into a sequence of basic instructions,

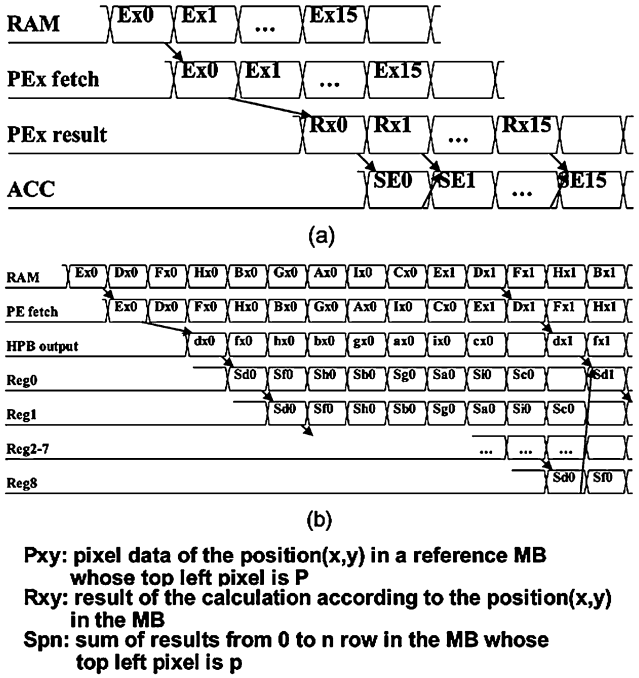

Fig. 10. Timing diagram. (a) MSE calculation. (b) 3 × 3 NHS.

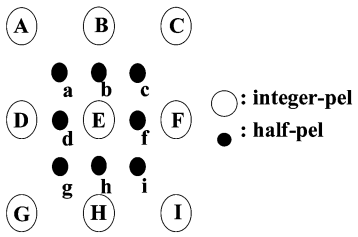

Fig. 11. Half-pel pixel generation.

and controls the sequence. The I_SEQ gives the VG control signals to store the calculation result at the completion of the basic instruction. The I_COUNTER counts the number of basic instructions.

The C_SEQ controls execution of the basic instruction given by the I_SEQ. The C_SEQ produces control signals to the AG and the PU. The C_SEQ stores them into the PIPELINE_REG. The C_COUNTER counts the number of clock cycles.

The PIPELINE_REG consists of pipelined registers. Each register corresponds to the pipeline stage of the SIMD datapath. It includes control signals for the stage. The control signals for all stages are produced by the C_SEQ simultaneously. They are bound to image data in one row of an MB to be calculated. They move in the PIPELINE_REG with the same pace as the image data in the SIMD datapath. This control method simplifies a design of the SEQ.

D. Vector Generator and Address Generator

Fig. 13 shows a block diagram of the VG. The VG receives the *x* differential coefficient and the y differential coefficient from the PU. The next search direction is generated from them. If the direction is the same as the previous direction, then the VG gives a signal to finish the current instruction to the SEQ. The next search vector is generated from the current search vector, a

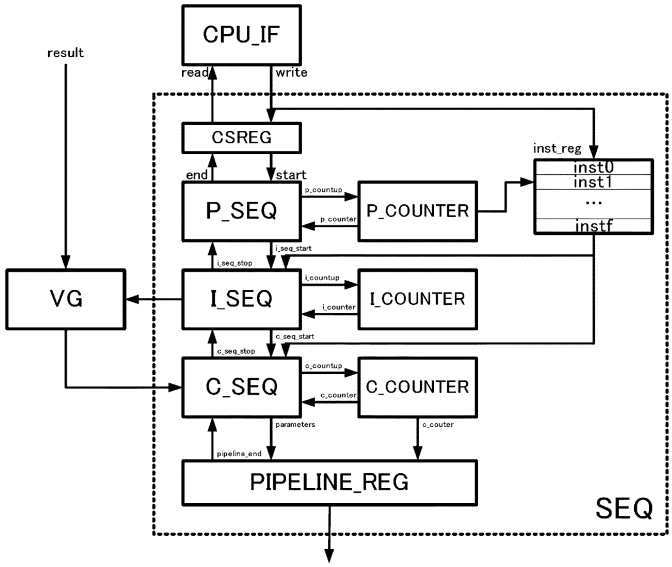

Fig. 12. Hierarchical sequencer.

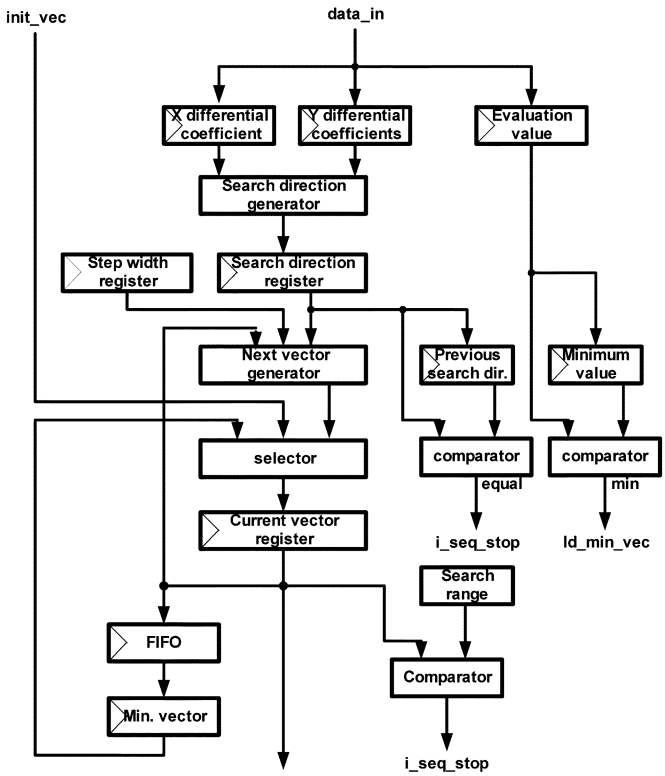

Fig. 13. Vector generator.

step width, and the next search direction. The current vector is chosen from three vectors as follows:

- one of predicted vectors;
- a vector having the minimum evaluation value;
- a vector generated by the next vector generator.

At the initial vector decision, one of the predicted vectors is chosen. At the start of the 1-DS, the vector that has the smallest evaluation value until now is chosen. In the middle of the 1-DS, the next search vector is chosen. The selected vector is stored in the current vector register. If the current search vector exceeds

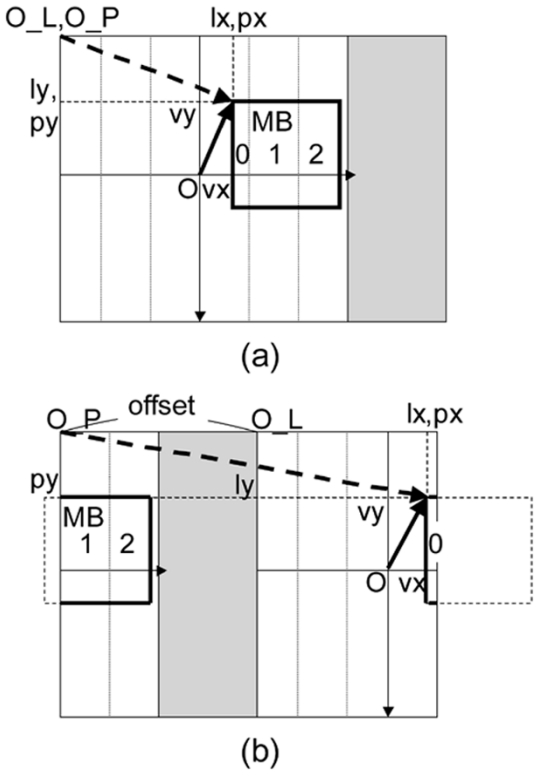

Fig. 14. Search window buffer. (a) Offset = 0. (b) Offset = 32.

the search range, then the VG gives a signal to finish the current instruction to the SEQ.

The VG also receives the evaluation value from the PU. If the evaluation value is smaller than the minimum value, then it is stored in the minimum value register, and the corresponding vector is stored in the minimum vector register. The 1-DS executes calculations of the evaluation value for each vector continuously, so a calculation for the subsequent vector is executing at the completion of the previous vector. The FIFO is necessary to store the vector corresponding to the minimum evaluation value at that time.

Fig. 14 illustrates how to generate an address of the SW buffer. Fig. 14(a) illustrates the SW buffer that an offset equals 0. The offset is the start address of the SW, not of the SW buffer. The white area is the SW and the gray area is a part of the SW buffer to add reference image data for the next MB. A procedure to generate an address corresponding to the search vector (vx, vy) is as follows. The logical address (lx, ly) is calculated from the search vector (vx, vy). The lx and ly are x and y coordinates whose origin is O_L. The O_L is the top left pixel of the SW. Next the physical address (px, py) is calculated from the logical address (lx, ly). The px and py are x and y coordinates whose origin is O_P. The O_P is the top left pixel of the SW buffer. The physical address is equivalent as the logical address when the offset equals 0. Then the SW buffer addresses (addr0, addr1, addr2) are calculated from the physical address (px, py). The SW buffer consists of eight SW RAMs, so an address boundary exists every eight pixels. Therefore, three addresses exist for one row of an MB. The SW buffer address can be obtained by a concatenation of px and py, basically.

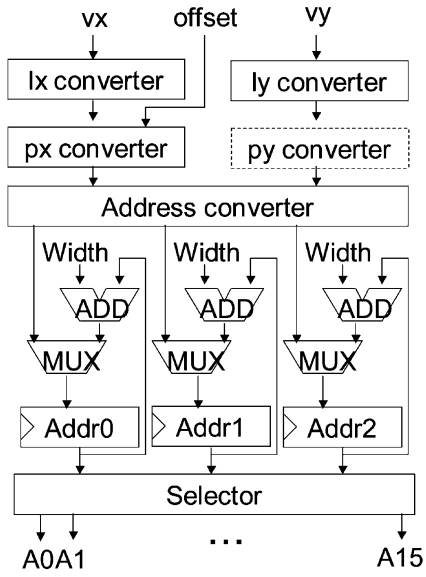

Fig. 15. Address generator.

Fig. 14(b) illustrates the SW buffer that the offset equals 32. The lx and px are different because the logical origin O_L and the physical origin O_P are different this time. On the other hand, the ly and py are always the same. In this case, the address addr1 corresponding to the area 1 is calculated from the physical address (0, py), because the right side of the MB exceeds the right boundary of the SW buffer.

Fig. 15 depicts a block diagram of the AG. The AG generates a SW buffer address indicated by a search vector given by the VG. The data flow from the search vector (vx, vy) to the address (addr0, addr1, addr2) is the same as described above. The next address can be obtained by adding the one line width of the SW buffer (Width) to the current address. Each address (from A0 to A15) chooses one of three addresses (addr0, addr1, addr2) according to the area that the corresponding pixel belongs to.

IV. Circuit Design

A. Three-Port SRAM

The ME core integrates eight pieces of the three-port SRAM macro for about 32-kb storage as SW RAMs. The area of SW RAMs is almost one-half of the ME core. A low power design for the three-port SRAM macro is essential to realize the sub-mW ME core. The three-port SRAM macro has three major features to reduce power dissipation, as follows:

- full divided wordline structure;
- write-disturb-free memory cell arrangement;
- symmetric three-port memory cell layout.

The divided wordline structure for the entire three-port circuit drastically reduces the bitline current. The bitline occupies a significant amount of total power consumption of the macro. The structure reduces the power consumption of the bitline to 1/2 of its previous value.

A write-disturb problem frequently appears in operation of a conventional multi-port RAM. It is completely eliminated by a newly developed cell-array arrangement. This is realized by a

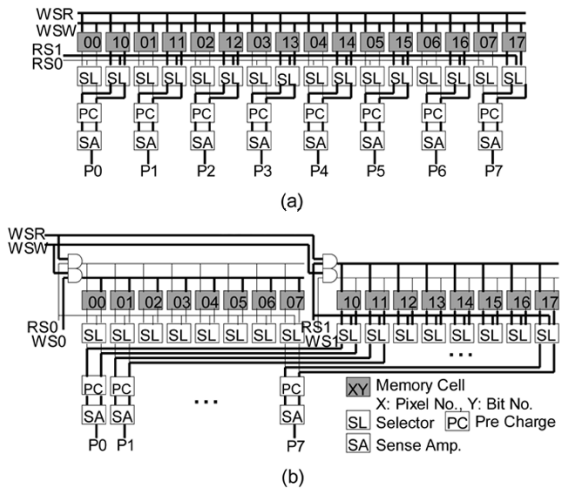

Fig. 16. Write-disturb-free memory cell arrangement. (a) Conventional method. (b) Proposed method.

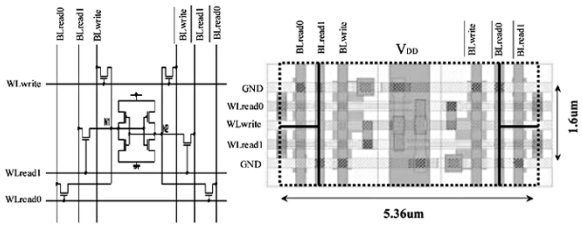

Fig. 17. Symmetric memory cell layout.

combination of the full divided wordline structure and a wordline composition scheme. The wordline is connected to only one row composed of eight memory cells corresponding to one pixel.

The write-disturb-free memory cell arrangement is illustrated in Fig. 16. The conventional method arranged more than one pixel in the same row. A collision writing to and reading from the same row is possible to occur with this method. The proposed method avoids this collision by the arrangement that places one pixel into one divided wordline. It is notable that simultaneous write and read accesses to the same pixel can never happen in motion estimation. It is possible to design a smaller memory cell to operate under low-voltage condition with this method.

A symmetric three-port memory cell layout (Fig. 17) has been introduced to avoid influence to the transistor ratio within the cell by process issues such as mask misalignment. This enhances cell stability, particularly under low-voltage condition less than 1 V. This feature enables 1-V operation, allowing low-power characteristic.

The block diagram of the three-port SRAM macro, which is utilized for SW RAMs, is shown in Fig. 18. The macro has concurrent three-port access capability (2R1W) and a 512-word by 8-bit configuration.

Simulation results of the three-port SRAM macro is shown in Fig. 19. It represents that access time is 23 ns and cycle time is 73 ns. The maximum frequency is about 21 MHz under 1-V supply voltage condition. The macro with 512×8 bit consumes only 0.21 mW at 1 V under 13.5-MHz operation condition. Hence, the power dissipation of the SW buffer is suppressed to 1.68 mW under 1-V operation.

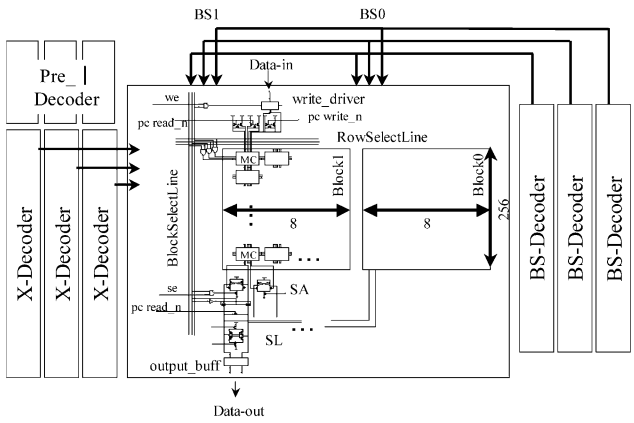

Fig. 18. Three-port SRAM macro block diagram.

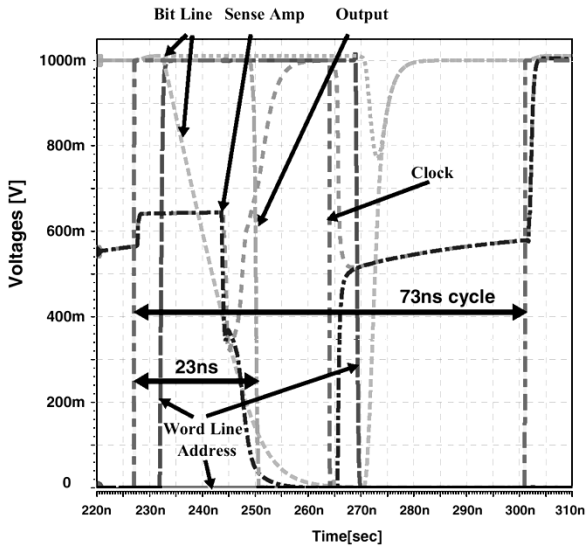

Fig. 19. Three-port SRAM simulation waveform.

B. Gated Clock

In the case of the GDS algorithm, the average computational complexity is about one-half of the worst complexity. The worst complexity decides an operating frequency of the ME core. After the completion of the motion estimation, the ME core stops clocking to all circuits except I/O by the gated clock technique. An extensile use of the gated clock technique minimizes power dissipation.

V. VLSI Implementation and Evaluation

A. Evaluation

The ME core was implemented using 0.18-μm five-layer metal CMOS technology. A photomicrograph of the ME core is shown in Fig. 20. The ME core incorporates about 1M transistors. Ten pieces of 4-kb three-port SRAMs are integrated in the core as SW RAMs and TB RAMs.

A shmoo plot obtained by a VLSI tester is shown in Fig. 21. It indicates that the ME core can operate under 0.7-V condition at 0.85 MHz. The ME core can also operate under 1.2-V condition at 13.5 MHz.

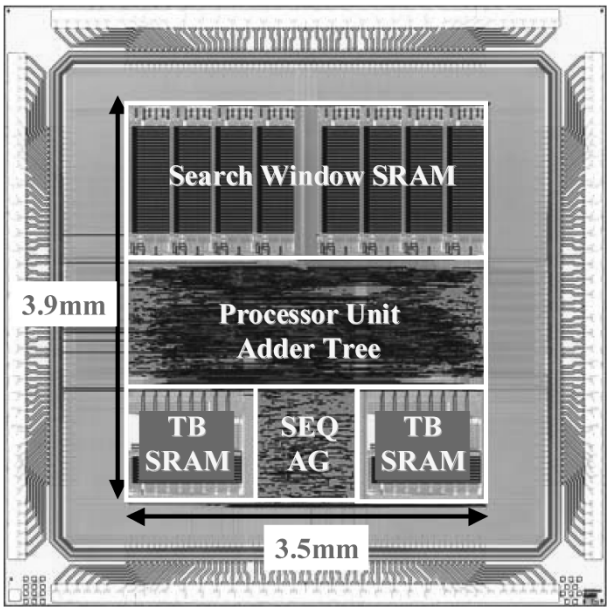

Fig. 20. ME core photomicrograph.

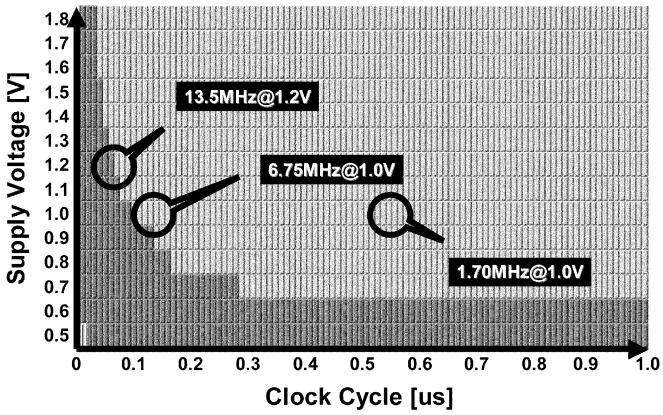

Fig. 21. Shmoo plot.

Fig. 22 shows power consumption measured using picture data. The power consumption in the GDS without SB search is 0.4 mW under 0.85 MHz at 1.0 V. In the case of the GDS combined with the SB search method, the power dissipation is 0.9 mW under 1.70 MHz at 1.0 V.

B. Characteristics

Characteristics of the ME core are summarized as follows:

- technology: 0.13-μm CMOS, five metal layers;
- core size: 3.9 mm $\times$ 3.5 mm;
- number of transistors: 1 million;
- function: motion estimation, half-pel precision;
- resolution: QCIF, CIF;
- frame rate: 15 f/s, 30 f/s;
- search range: $[-16 : +15.5]$;
- clock frequency: 13.5 MHz;
- power supply: 1.0 V;

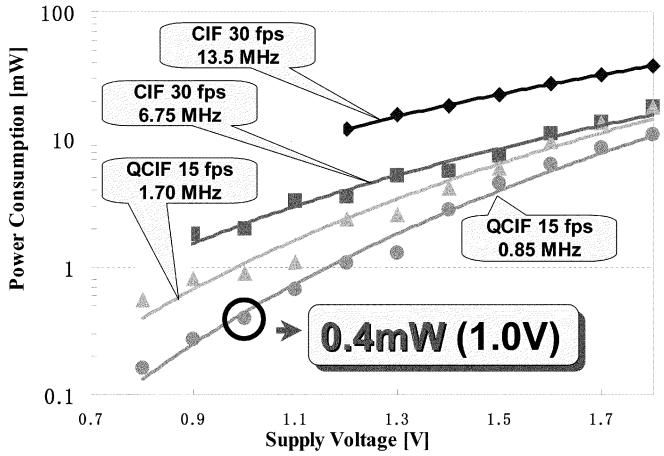

Fig. 22. ME core power consumption.

- power consumption:
  0.4 mW (QCIF 15 f/s, 0.85 MHz);
  0.9 mW (QCIF 15 f/s, 1.70 MHz);
  2.5 mW (CIF 30 f/s, 6.75 MHz);
  12 mW (CIF 30 f/s, 13.5 MHz at 1.2 V).

VI. Conclusion

Highly efficient motion estimation is essential to produce a low-power MPEG-4 video codec with superior visual quality. A motion estimation processor for MPEG-4 video encoding is described in this paper. The GDS algorithm reduces the computing power approximately to 7% comparing with the conventional full search method, and produces higher picture quality than the other fast motion estimation algorithms. The ME core contains 16-way SIMD datapath and low-power three-port SRAMs for highly parallel operation. A clock frequency and an operating voltage were reduced by the above techniques. Hence, the ME core attains ultra-low-power dissipation less than 1 mW at a QCIF 15 f/s with high picture quality. Also, the ME core supports a wide range of resolution from QCIF 15 f/s to CIF 30 f/s. It is applicable to the MPEG-4 mobile video applications. This algorithm and architecture will be easily extended to process finer resolution video with low power consumption.

Acknowledgment

The VLSI chip in this study was designed with Cadence and Synopsys CAD tools.

References

[1] T. Matsumura, S. Kumaki, H. Segawa, K. Ishihara, A. Hanami, Y. Matsuura, S. Scotzniovsky, H. Takata, A. Yamada, S. Murayama, T. Wada, H. Ohira, T. Shimada, K. Asano, T. Yoshida, M. Yoshimoto, K. Tsuchihashi, and Y. Horiba, "A single-chip MPEG2 422@ML video, audio, and system encoder with a 162 MHz media-processor and dual motion estimation cores," *IEICE Trans. Electron.*, vol. E84-C, no. 1, pp. 202–211, Jan. 2001.

[2] S. Uramoto, A. Takabatake, M. Suzuki, H. Sakurai, and M. Yoshimoto, "A half-pel precision motion estimation processor for NTSC-resolution video," *IEICE Trans. Electron.*, vol. E77-C, no. 12, pp. 1930–1936, Dec. 1994.

[3] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation*. Boston, MA: Kluwer, 1999.

[4] H. Ohira, T. Kamemaru, H. Suzuki, K. Asano, and M. Yoshimoto, "A low power media processor core performable CIF30 fr/s MPEG4/H26x video codec," *IEICE Trans. Electron.*, vol. E84-C, no. 2, pp. 157–165, Feb. 2001.

[5] M. Takahashi, T. Nishikawa, M. Hamada, T. Takayanagi, H. Arakida, N. Machida, H. Yamamoto, T. Fujiyoshi, Y. Ohashi, O. Yamagishi, T. Samata, A. Asano, T. Terazawa, K. Ohmori, Y. Watanabe, H. Nakamura, S. Minami, T. Kuroda, and T. Furuyama, "A 60-MHz 240-mW MPEG-4 videophone LSI with 16-Mb embedded DRAM," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1713–1721, Nov. 2000.

[6] H. Nakayama, T. Yoshitake, H. Komazaki, Y. Watanabe, H. Araki, K. Morioka, J. Li, L. Peilin, S. Lee, H. Kubosawa, and Y. Otobe, "An MPEG-4 video LSI with an error-resilient codec core based on a fast motion estimation," in *Proc. IEEE ISSCC*, 2002, p. 296.

[7] M. Takabayashi, K. Imamura, and H. Hashimoto, "A fast motion vector detection based on gradient method,", Tech. Rep. of IEICE, IE2001-74, Sept. 2001.

[8] M. Miyama, O. Tooyama, N. Takamatsu, T. Kodake, K. Nakamura, A. Kato, J. Miyakoshi, K. Imamura, H. Hashimoto, S. Komatsu, M. Yagi, M. Morimoto, K. Taki, and M. Yoshimoto, "An ultra low power motion estimation processor for MPEG2 HDTV resolution video," *IEICE Trans. Electron.*, vol. E86-C, no. 4, pp. 561–569, Apr. 2003.

Masayuki Miyama (M'04) was born on March 26, 1966. He received the B.S. degree in computer science from University of Tsukuba, Japan, in 1988 and the M.S. degree in computer science from Japan Advanced Institute of Science and Technology in 1995.

He joined PFU Ltd. in 1988, and Innotech Company in 1996. He is currently a Research Assistant in the Department of Electrical and Electronic Engineering at Kanazawa University, Japan. His current research focus is low-power design techniques for multimedia VLSI.

Junichi Miyakoshi was born on February 22, 1980, in Niigata Prefecture, Japan. He received B.E. degrees in electrical and information engineering from Kanazawa University, Ishikawa, Japan, in 2002. He is currently a masters degree student at Kanazawa University.

His research interests include system VLSI design and implementation of multimedia communication system.

Yuki Kuroda was born on October 3, 1980, in Japan. He received the B.S. degree in electrical and computer engineering from Kanazawa University, Ishikawa, Japan, in 2003. He is a masters degree student at Kanazawa University.

His current research interests are in low-power design technology and a multimedia processor for mobile communications and Intelligent Transport Systems.

Kousuke Imamura received the B.S., M.S., and Dr. Eng. degrees in electrical engineering and computer science from Nagasaki University, Japan, in 1995, 1997, and 2000, respectively.

He is a Research Assistant in the Department of Information and Systems Engineering, Kanazawa University, Japan. His research interests are high-efficiency image coding and image processing.

Hideo Hashimoto (M'77) received the B.S., M.S., and Dr. Eng. degrees in electronic engineering from Osaka University, Japan, in 1968, 1970, and 1975, respectively.

He joined the Electrical Communication Laboratories of Nippon Telegraph and Telephone Corporation (NTT) in 1975. Since 1993, he has been a Professor of information and systems engineering at Kanazawa University, Japan. His research interests are video coding, moving object segmentation, and visual communication.

Masahiko Yoshimoto (M'04) received the B.S. degree in electronic engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1975, the M.S. degree in electronic engineering from Nagoya University in 1977, and the Ph.D. degree in electrical engineering from Nagoya University in 1998.

He joined the LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan, in April 1977. From 1978 to 1983, he was engaged in the design of NMOS and CMOS static RAM including a 64K full CMOS RAM with divided-wordline structure. Since 1984, he has been involved in the research and development of a digital NTSC decoder LSI with adaptive filtering called VSP, an image compression DSP called DISP, a 100-MHz DCT processor, MPEG2 video encoder/decoder LSIs, 3-D graphics processor, multimedia ULSI systems for the digital broadcasting and the digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology and so on. Since 2000, he has been a Professor in the Department of Electrical and Electronic System Engineering, Kanazawa University, Japan. His current activity is focused on the research and development of multimedia and ubiquitous media VLSI systems including an ultra-low-power image compression processor and a low-power wireless interface circuit. He holds 70 registered patents.

Dr. Yoshimoto is a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE). He served on the program committee of the IEEE International Solid-State Circuit Conference from 1991 to 1993. Also, he served as Guest Editor for special issues on Low-Power System LSI and IP and Related Technologies of IEICE Transactions in 2004. He received the R&D100 Awards from the R&D magazine for the development of the DISP and the development of the realtime MPEG2 video encoder chipset in 1990 and 1996, respectively.