A 1.2-W Single-Chip MPEG2 MP@ML Video Encoder LSI Including Wide Search Range (H: ±288, V: ±96) Motion Estimation and 81-MOPS Controller

Eiji Ogura, Masatoshi Takashima, Daisuke Hiranaka, Toshiro Ishikawa, Yukio Yanagita, Shuji Suzuki, Tokuya Fukuda, and Toshiyuki Ishii

Abstract— An MPEG2 MP@ML video encoder large-scale integrated circuit (LSI) has been developed including an 81 MOPS controller and motion estimator. By using two adaptive algorithms, a wide motion-estimation search area (±288 pixels horizontal and ±96 pixels vertical) was achieved with computation complexity of only 0.5% (20 GOPS) of full search block-matching algorithm. By using this expanded motion-estimation search area, there is a significant improvement in picture quality for coding fast motion sequences. The power consumption was reduced by using an efficient pipeline architecture and optimizing the circuitry, especially in the motion-estimation block and the data transfers for the external SDRAM. The 13.7 × 12.4 mm², 4.5-M transistor device using 0.4-µm CMOS technology dissipates 1.2 W at 3.3 V.

Index Terms— Large-scale integration, motion compensation, video signal processing.

I. INTRODUCTION

To design a low-cost MPEG2 encoder large-scale integrated circuit (LSI) for consumer applications, all functions have to be integrated into a single-chip and meet the following three requirements.

First, cost-effective motion estimation has to be included. Some of the conventional MPEG2 video encoder chipsets use the full-search block-matching algorithm (FSBMA) [1], [2] for motion estimation, which requires large computational power and leads to a huge hardware cost. Thus, reducing the amount of motion-estimation hardware is the key to designing a practical, cost-effective, single-chip encoder. There are several approaches to reducing the amount of hardware, such as using subsampling [3], two-step hierarchical search [4], [5], [9], or three-step hierarchical search [6], [7]. The telescopic search technique [2], [5] can also be combined with one of the other techniques to obtain a wider search area. However, this technique requires huge memory bandwidth and complicated controls. Hence, it may not be suited for a single-chip approach where the memory bandwidth is limited. We also have developed a cost-effective motion-estimation LSI [8] in the past by using an integral projection technique [11], [12]. However, multiple motion-estimation LSI’s had to be used to achieve reasonable range of search area.

Second, a programmable controller must be included in the chip, allowing the bit stream to be generated without an external host controller. Third, low power consumption and low external memory requirements have to be achieved. Previous single-chip MPEG2 encoder LSI’s [9] did not have an internal controller, and therefore an extra host controller was necessary along with external SDRAM’s.

To meet these requirements, we have developed two adaptive algorithms for motion estimation and implemented them to achieve wide search area with less hardware than required by FSBMA. In addition, by using an efficient pipeline architecture, optimizing the circuitry and data transfers between the external memory, reduced external memory requirements and low power consumption have been achieved.

The chip architecture is briefly discussed, including the multiclock system in Section II. In Section III, the two adaptive algorithms implemented for motion estimation are described, showing how the number of operations required was reduced compared to FSBMA. In Section IV, the design methodology, focusing mainly on achieving low power consumption, is discussed by showing some architecture and circuit examples.

II. CHIP ARCHITECTURE

Fig. 1 shows the architecture of the LSI. Using a 27-MHz clock input, the internal phase-locked-loop unit generates multiple system clocks such as 13.5, 27, 22.5, 45, and 67.5 MHz. The operating clock frequency for each function was selected to match the necessary operation power as close as possible to reduce the idle time. As a result, most of the circuits operate at more than 97% of the whole period, and the use of gated clock [9], [10] was not necessary for reducing the power consumption. The “PRE” unit block, which operates at 27 and 13.5 MHz, performs functions such as 4:2:2:4:2:0 chroma-format conversion, temporal noise reduction, and scene change detection. Full-pel motion estimation (ME), half-pel ME, and motion compensation (MC) are executed in the “full-pel ME” unit and “half-pel ME/MC” unit, respectively, running at 45 MHz. The discrete cosine transform (DCT) and quantization block and the variable-length coding (VLC) block operate at 22.5 MHz. The internal 81-MOPS digital signal processing (DSP) controls all the other units and performs functions such as rate control and memory management. This allows
encoded video elementary stream to be generated independent of an external host controller. An external host’s purpose is then reduced to setting encoding parameters and reading information from the chip. External 32-Mb SDRAM memory is connected via a 32-bit data bus running at 67.5 MHz. The SDRAM is used to store original pictures, local decoded pictures, encoded bit stream, and DSP instructions. The details of the SDRAM data transfers are discussed in Section IV.

III. MOTION ESTIMATION

The full-search block matching is the most widely used algorithm for motion estimation. In this algorithm, the current frame is divided into small macro blocks, as shown in Fig. 2. A candidate block of the search frame is shifted along all the positions within the search area to find the best match to the current macro block. The mean absolute error (MAE) is commonly used as the matching criterion. The problem using this algorithm is the high computational requirement, which leads to a huge hardware cost. For example, for a wide search area of 288 horizontal and 96 vertical pixels, 4.5 teraoperations per second (TOPS) are required. Therefore, it is hard to implement motion estimation with a wide search area in a single chip using this straightforward approach.

To reduce the amount of computation, two algorithms were combined for full-pel precision motion estimation. The first algorithm is the 8 : 1 adaptive subsampling using full search and the second algorithm is the adaptive search area control.

A. 8 : 1 Adaptive Subsampling

By optimizing the tradeoff between the performance and the hardware cost, 8 : 1 adaptive subsampling was selected. As shown in Fig. 3, a sampling point is adaptively selected by choosing maximum and minimum pixel values for each vertical line in the current macro block. This is like choosing pixels that have strong characteristics within the current macro block. These selected pixels are then used to calculate the MAE’s. To obtain the field motion vectors, these processes are done on a field-by-field basis. As a result, the number of pixel data used for MAE calculation is reduced from 256 to 32, and an 8 : 1 reduction in computation is obtained. By using this algorithm, average loss of luminance signal-to-noise ratio (SNR) was 0.2 dB less than FSBMA when coding four MPEG test sequences such as mobile and calendar, table tennis, flower garden, and cheerleader. This 8 : 1 adaptive subsampling algorithm is very simple and suited for hardware implementation.

Fig. 4 compares the performance of the conventional fixed subsampling [Fig. 4(a)] and proposed adaptive subsampling algorithm [Fig. 4(b)]. In Fig. 4(a), macro block size is shown as 4 × 4, with pixel values inside the circles. The shaded pixel is the subsampled pixel and is used to calculate MAE for each candidate block. Based on the two candidate blocks shown, “candidate block 1” is the perfect match, and this block must be selected. “Candidate block 2” is a flat image with all pixel values of one. When we calculate MAE for the two candidate blocks using conventional fixed subsampling,
shown in Fig. 4(a), both MAE1 and MAE2 become zero. This means that it is impossible to distinguish the two different candidate blocks using this algorithm, and candidate block 2 may be wrongly selected.

Fig. 4(b) shows what happens when the proposed adaptive algorithm is used for the same condition. First, the current macro block is divided into two small blocks, with the pixels having the minimum and maximum values in each of the small blocks selected as subsampling points. This example uses two horizontal pixel lines instead of a vertical line as a group of subsampling choice. The number of sampling points is the same as before. Next, two MAE’s are calculated for each candidate block. In this case, MAE1 is smaller than MAE2. This means that candidate block 1 is always selected. Both algorithms have exactly the same number of operations to calculate MAE. However, the proposed adaptive subsampling algorithm has better performance.

By comparing the coded average luminance SNR of the two subsampling algorithms using real image sequences, the performance of the proposed 8:1 adaptive subsampling was equivalent to the 4:1 fixed subsampling. And by coding some image sequences, such as mobile and calendar MPEG test sequence, the proposed 8:1 adaptive subsampling even outperformed the 4:1 fixed subsampling. By comparing the 8:1 adaptive subsampling with the conventional 8:1 fixed subsampling, there was a 0.2-dB improvement in luminance SNR.

B. Adaptive Search Area Control

Encoding with a small motion-estimation search area results in picture-quality loss in fast motion sequences, especially with fast background moving images. On the other hand, encoding with an expanded motion-estimation search area results in better picture quality due to the improvement in coding efficiency. However, it demands more hardware, making the motion-estimation block difficult to design a single chip having a reasonable chip size. To expand the motion-estimation search area without increasing the amount of hardware, a second algorithm was implemented. This is probably the most important functional element of the chip. This algorithm will be referred as the adaptive search area control.

Fig. 5 shows how the adaptive search area control works. A wider search area is obtained by adaptively changing the center point of the search area. In the case of P-pictures, two independent motion-estimation blocks designated as ME1 and ME2 are used concurrently. Each motion-estimation block has a search area of −32 to +31.5 horizontal pixels and −16 to +15.5 vertical pixels with full search. By setting the two offsets of the ME block based on a distribution history of the motion vectors, the search area can be expanded up to −288 to +287.5 horizontal pixels and −96 to +95.5 vertical pixels, as shown in Fig. 5. The offset values are determined by the internal DSP. Using this second algorithm, computational complexity is reduced 27:1 without an increase in hardware. For example, as shown in Fig. 5(a), when there is no large motion in the video, the two motion-estimation blocks can be placed beside each other. However, if there is a fast horizontal motion, the ME1 block can be set in the middle to cover the small motion vectors, and the ME2 block can be set at a corresponding position to cover the large horizontal motion, as shown in Fig. 5(b).

Fig. 6 shows an example of how the adaptive search area control works effectively for coding images when the background is changing rapidly. Fig. 6(a) shows a person running with a fast camera pan trying to follow the motion. This results in a fast moving background. This example is one of the most critical cases when encoding with a conventional small motion-estimation search area simply because most of the motion vectors are outside the search area, as shown in Fig. 6(b). The result would be critical picture-quality loss in the image background. In this example, most of the motion vectors are located at MV2, the motion vectors for the background. The motion vectors for the person are located at MV1. In most encoding situations, motion vectors are concentrated in a few places, mostly at two places such as MV1 and MV2. Therefore, as shown in Fig. 6(c), it is possible to cover most of the motion vectors by using two independent motion-estimation blocks and achieve a significant improvement in picture quality in the background.

C. Picture-Quality Improvement by Widening the Motion-Estimation Search Area

To measure the picture-quality improvement by widening the motion-estimation search area, a luminance SNR of the volleyball sequence was compared by encoding with different
motion-estimation search areas at 4 Mbps. The results are shown in Fig. 7. In this sequence, there is a fast horizontal background motion as a ball is thrown from one side of the court to the other with the camera panning to follow the ball. The SNR encoded by a conventional motion-estimation search area, which is ±96 horizontal pixels and ±48 vertical pixels, is shown. FSBMA was used, requiring 746 gigaoperations per second (GOPS). This is compared to the luminance SNR (YSNR) encoded by a wide motion-estimation search area using the proposed motion-estimation algorithms, which required only 20 GOPS. During frame numbers 0–10, the motion is very small and there is no difference in YSNR. From frame number 10 to frame number 40, there is a fast-pan camera movement. During this period, the YSNR of the proposed wide search area shows a maximum of 4-dB improvement over the YSNR of the conventional small search area. The bottom of Fig. 7 shows how the search areas of the two motion-estimation blocks change to follow the fast horizontal motion. Furthermore, the YSNR using the proposed wide search area at 4 Mbps was measured to be equivalent to the YSNR using conventional small search area at a higher bit rate of 6 Mbps. From these results, it is clear that widening the motion-estimation search area has a significant improvement on encoded picture quality, especially for fast motion sequence.

D. The Full-Pel Precision Motion-Estimation Architecture

Fig. 8 shows the full-pel precision motion-estimation block architecture. Picture data stored in the external SDRAM is sent to the first in, first out (FIFO) in the SDRAM interface. The current macro-block data and search data are sent to the two motion-estimation circuits, designated as ME1 and ME2. The sampling point of the current macro block is adaptively selected by the “Data Selection” block. Selected values are stored in the “Current MB storage” block with the position information. Each processor element (PE) performs subtract and absolute operations to calculate the MAE’s. Minimum MAE’s are selected in the “MIN” block, and the MAE values are compared in the “COMP” block. The DSP obtains the final results from the motion estimation, calculates a distribution for the motion vectors, and then sets the offsets of each motion-estimation circuit.

E. Motion-Estimation Summary

Fig. 9 summarizes the reduction in the computational complexity for motion estimation. The first motion-estimation algorithm, using 8:1 adaptive sampling, reduces computational complexity down to 12.5%. By combining the second adaptive search area control algorithm, computational complexity is further reduced down to 0.5%, which is a 27:1 reduction. In total, a 216:1 reduction in computational complexity can be achieved. The number of operations required is reduced from 4.5 TOPS down to only 20 GOPS, and therefore, motion estimation with a wide search area could be implemented in a single chip.
IV. Design Methodology

The LSI has been developed based on 0.4-μm application-specific integrated-circuit CMOS standard-cell library. No special cells or multiple power voltages were used. Low-power design strategy was achieved in system-level design, algorithm optimization, architecture optimization, and logic-level design.

First, starting from the system design, each task was allocated among suitable hard-wired logic or software functions. Tasks higher than the macro block layer, such as motion-compensation mode selection and rate control, were allocated to software functions in the DSP controller. Tasks lower than macro-block layer such as motion estimation, DCT, and VLC were implemented in hard-wired logic.

Second, as previously explained, algorithm optimization was performed for motion estimation to reduce the amount of hardware.

Third, to optimize the architecture, clock distribution, data operation, and data representation were taken into account. As well, correlation between data was checked and used to optimize the data-path design. The low-power circuit was mainly considered for the motion estimation and SDRAM interface blocks, which operate at higher frequency clocks, such as 45 and 67.5 MHz, respectively. Also the overall clock was lowered 20% to reduce the power consumption and idle time for each block. For example, 81- and 27-MHz clock was lowered to 67.5 and 22.5 MHz, respectively. One example of hardware optimizing is in the data transfers between external memory and the LSI. MPEG2 encoding requires large amounts of data transfers between external memory and the LSI. However, the data-bus clock frequency should be kept as low as possible for reducing power consumption. In our design, we used a 32-bit data bus at a low clock speed of 67.5 MHz, which gives a maximum performance of 270 MB/s. During the macro-block processing, when peak transfer rates occur, we adopted a scheduled transfer architecture for maximum bus performance along with optimized memory mapping. Each macro-block period has 1536 cycles at 67.5 MHz. There will be a four-cycle penalty when switching between read and write SDRAM data transfers. To avoid this penalty, read or write data transfers are scheduled by using hardware state machines to combine the same type of transfer as much as possible. For B-picture encoding, where a maximum transfer of 266 MB/s occurs, bus utilization of 98.4% can be achieved. If this optimization is not used, faster bus clock such as 81 MHz [9], [10] becomes necessary, which leads to higher power consumption. During frame processing, where the required transfer rate is lower than during macro-block processing, arbitration was used for SDRAM data transfers for more flexibility.

In the motion-estimation block, a power-saving circuitry was also applied. Fig. 10(a) shows a conventional circuit with an eight-step shift register with data width of 12 bits running at 45 MHz. Fig. 10(b) shows the implemented circuit, which replaces the shift registers with transparent latches, a rotation barrel shifter, and a 3-bit counter. The clock is supplied only to the three flip-flops of the counter as opposed to 96 flip-flops in the conventional circuit. The power consumption of the implemented circuit is estimated to be 30% of the conventional circuit.

Last, we considered logic-level design. Image-compression chips tend to use many small FIFO’s with wide data bits and short words, which leads to high power consumption and layout design constraints. To avoid this, small FIFO’s were implemented using transparency latches with logic gates as shown in Fig. 11. This figure shows an example of a 32-bit, 128-word FIFO operating at 67.5 MHz. To avoid the common glitching problem that occurs during address decoding, a gray code decoder was used for FIFO address decoding circuits. As shown in the timing diagram of Fig. 11, the address generated by the gray code counter has no glitch as it counts up, and this results in no glitching in the final output data. The power consumption of this FIFO is estimated to be 80% of the conventional hard macro RAM. As well, by using soft macro FIFO’s, there are fewer layout design constraints compared to using hard macro RAM’s. Another advantage of using this design approach is that memory testing can be simplified.

![Fig. 10. Example of power-saving circuit in ME.](image-url)
Table I shows the specifications of the LSI. A die photograph of the LSI is shown in Fig. 12. The function of the LSI is MPEG2 MP@ML real-time encoding for both NTSC and PAL images up to 95.5 [two bidirectional (B) pictures is supported]. The chip size is 13.7 x 12.4 mm, integrating 4.5 million transistors using 0.4-μm CMOS technology. The power consumption is 1.2 W at 3.3 V.

VI. CONCLUSION

A single-chip MPEG2 encoder LSI has been developed, including a cost-effective motion estimation and a programmable controller. Motion estimation has been implemented by combining two adaptive algorithms, reducing the computational complexity down to 0.5% (20 GOPS) of the full search block matching algorithm. By using the expanded motion-estimation search area, there is a significant improvement in picture quality for coding fast motion sequences. Low power consumption and low external memory requirements have been achieved. We believe this LSI will help in building low-cost, real-time MPEG2 encoding systems and applications.

ACKNOWLEDGMENT

The authors would like to thank Y. Itoh, T. Miura, M. Shiraga, T. Ohtsuki, C. Ko, S. Hasama, K. Aoki, T. Fukushima, E. Ozeki, A. Nakayama, and K. Tsunoji for their great contributions to this work. They also would like to thank G. Wei for helping to prepare this manuscript.

REFERENCES

Toshiro Ishikawa received the B.S. degree in electrical engineering from Kagoshima University, Japan, in 1986. He joined NTT Electronics Technology, where he worked on designing signal-processing LSI’s. He joined Sony Corp., Tokyo, Japan, in 1991 and has been working on LSI design. He is now Assistant Manager of System LSI Division, Semiconductor Company. His interests lie in high-performance and lower power consumption LSI.

Yukio Yanagita received the B.S. degree in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1993. In the same year, he joined Sony Corp., System LSI Division, Semiconductor Company, Tokyo, and has been working on LSI design.

Shuji Suzuki received the B.S. degree in theoretical physics from St. Paul’s University, Tokyo, Japan, in 1991. He joined Sony Corp., System LSI Division, Semiconductor Company, Tokyo, in 1991 and has been engaged in LSI design and the development of LSI CAD technology.

Tokuya Fukuda received the B.S. degree in electronic engineering from Keio University, Japan, in 1977. He joined Sony Corp., Tokyo, Japan, the same year and has participated in the development of image- and sound-processing systems for new video standards. He has been the General Manager of the System LSI Division, Semiconductor Company, and is engaged in the development of MPEG2 encoding and decoding LSI’s. He has previously contributed three papers to ISSCC and two papers to ICCE.

Toshiyuki Ishii received the B.E. degree in electrical engineering from Waseda University, Japan, in 1979. He joined Sony Corp., Tokyo, Japan, in 1980 and is Manager of the Media Processing Laboratories. He has been engaged in the development of various systems, such as the personal computer, CD-I, MPEG application, video-CD, and DVD systems.

Eiji Ogura received the B.E. and M.E. degrees in electrical engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1987 and 1989, respectively. He joined Sony Corp. and is Assistant Manager of the Media Processing Laboratories, Tokyo. He has been engaged in the development of various image-processing systems, algorithms, and LSI’s for HDTV and MPEG. From 1990 to 1992, he was a Research Affiliate at the Media Laboratory, Massachusetts Institute of Technology, Cambridge.

Masatoshi Takashima received the B.E. and M.E. degrees in electrical engineering from Musashi Institute of Technology, Japan, in 1984 and 1986, respectively. He joined Sony Corp., Tokyo, Japan, in 1986 and is Manager of Media Processing Laboratories. He has been engaged in the development of various image-processing systems.

Daisuke Hiranaka was born in Sapporo, Japan. He received the M.S. degree in information science from Osaka University, Osaka, Japan, in 1994. In the same year, he joined Sony Corp. Media Processing Laboratories, Tokyo, Japan, and has been developing the MPEG2 video encoder LSI and its application system.


Daisuke Hiranaka was born in Sapporo, Japan. He received the M.S. degree in information science from Osaka University, Osaka, Japan, in 1994. In the same year, he joined Sony Corp. Media Processing Laboratories, Tokyo, Japan, and has been developing the MPEG2 video encoder LSI and its application system.