Motion Estimator LSI for MPEG2 High Level Standard

Li JIANG, Dongju LI, Shintaro HABA, Chawalit Honsawek, and Hiroaki KUNIEDA
Dept. of EE Eng.,Tokyo Institute of Technology, Tokyo

Abstract

In this design, a dedicated motion estimation LSI of MPEG2 is presented. Combining our bits truncation adaptive pyramid (BTAP) algorithm with Window-MSPA architecture as well as by using custom cell and full custom design methods, the chip size becomes 4.8*mm* × 4.8*mm* small with 0.5u 2-level metal CMOS technology. The test chip which works at 41.5 MHz, possesses a search range of ±67 for image size of 1920 × 1152 and achieves video rate of 30 field/s.

I. Introduction

Motion estimation is the most time consuming task in encoding video of today's hybrid video coding standards such as MPEG. As the interests for HDTV is growing, the high level standard requirement is coming into place. However, the existed motion estimation algorithms and hardware design methods are not well fitted into high level design smoothly[1, 2, 3]. In [4], a design with each chip consisting of 1024 PEs working at 200 MHz has been considered. In [3], a chip of 256 PEs working at 133 MHz has been proposed. They suffer from not only large chip area cost, but also the clock skew problems in high speed operation of large search array. In order to increase the cost-efficiency and reduce hardware size, as well as to enable the single chip realization of HL@MP, we have proposed *Bits Truncation adaptive Pyramid algorithm* and Window-MSPA architecture[1, 2] which results in very high cost-efficiency and good performance.[2].

In this design, we present a dedicated motion estimator LSI for HDTV purpose using proposed bits truncation algorithm and excellent Window-MSPA[6] architecture as well as custom cell and full custom design methods. The chip size is 1/16 times as small as conventional design such as [3] with better PSNR performance[2]. The test chip is estimated to work at 41.5 MHz for processing video rate of 30-field/s. If sufficient number of pins is available, the core of the chip will also work at the clock rate of 83 MHz for 60-field/s video rate. It makes the single chip realization of HDTV system possible. The layout of the chip is shown in Fig. 1.

II. Algorithm

The hierarchical search scheme is as follows. First, from raw image, we use 1/64 mean pyramid[1] downsampling to construct level 3 search window(SW) with search range ±7 and block of size 4 × 4. The SW and block are saved into ring buffer configured memory for renewing search window data of next blocks search. Secondly, we use 1/16 mean pyramid downsampling to construct level 2 search window with search range ±2 and block of size 8 × 8, and are also saved in local memory. The blocks in above two high levels are four MB' s down-sampled large block to increase performance[3]. Lastly, the raw image is used as level 1 with search range ±3 and block size 16 × 16(or low level). This data is loaded directly from SDRAM via SDRAM interface controller(IFC), and goes through register chain to generate parallel input data for search array I.

III. Overall architecture

The overall architecture is shown in the dashed line part of Fig. 3 as an embedded search engine in MPEG2 system. The *Pyramid generator*(PYG) generates the mean pyramid[1] data in different levels for search window and macroblock. These data are then saved into *local memory* and fed into *Search Array I*(SA1) for two high pyramid level search performing the 8-bit MAD(mean absolute difference) matching. The *Register File*(RF) feeds data directly to *Search Array II*(SA2) for low level search performing 1-bit truncation search.

The mean pyramid algorithm has two problems for hardware design. One problem is the I/O bandwidth, the other is local memory size. We will discuss these two problems first and then discuss our search array architecture.

A. I/O bandwidth

The I/O bandwidth is required as wide as 36 Gbps(search range ±63) for HL@MP mean pyramid algorithm. Even 64-bit data access to external frame buffer can only provide 5.3 Gbps of bandwidth(at 83 MHz). To decrease I/O bandwidth requirement, design methods such as clustering search[3] and reusing SW area are considered in our design as shown in Fig. 7. Only the dark part of search area is needed to be renewed and the renewing is for every 16 MBs since each clustering block contains four MBs. The I/O bandwidth for highest level is thus reduced from more than 30 Gbps to 1.44 Gbps, and total bandwidth requirement to be 3.98 Gbps in our design, which makes mean pyramid algorithm possible. This scheme allows us to use parallel pyramid level generator as shown in Fig. 8. Pipeline stage is inserted in between adders to increase further speed. After the access of SDRAM, we average the obtained 8 pixels data.

B. Local memory size

In three level mean pyramid algorithm, to generate one pixel for search window of level 3, we usually generate the search window of level 2 at first. The level 2 data has to be saved before we get a coarse motion vector in high level search and estimate this middle level search area. This results in too large middle level memory size for ASIC design. Our strategy is to directly generate the 3rd pyramid level search window without the middle level 's generation. After the search of highest level finishes, we generate the search window for middle level according to the coarse motion vector. The local memory size therefore is reduced to 4 times as small as conventional one.

C. Scheduling

The pipelined operation of pyramid level generation and search is shown in Fig. 6. $G3_0$ means the basic SW pyramid level generation as shown in Fig. 7(b). Afterwards, the renewal of pyramid level 3 $G3$ is taken place within 16 MB search time. $G2$ indicates the generation of middle pyramid level. It is performed after the pyramid level 3 search $S3$ is completed. $S2$ and $S1$ are the searches for middle level and low level. Due to clustering search, $G2$ or $S3$ and $S2$ spend 4 MB 's search time. Four clustering MBs in low level image use the same coarse motion vector of high level search.

D. Search Array Architecture

We adopted Window-MSPA[5, 6] for search array, which has been proposed for performing general repetitive window operations over the area in image data space. Its excellent features are minimum I/O, high parallel efficiency and free to select the number of PEs.

D.1. Search Array I Architecture

5 PEs are used for both level 3 and level 2 pyramid image search with both 100% PE utilization. Reconfigurable feature of Window-MSPA architecture allows us to use the same array for search in different levels. Figure 4 shows the search array structure and the local memory for feeding pyramid level data into search array I. The calculation of threshold value *sep* is performed at the same time. Fig. 9(a) shows internal structure of each PE.

D.2. Search Array II Architecture

A parallel search scheme of window MSPA is used here for finishing $(2p + 1) = 49$ search points as shown in Fig. 2. Data is inputed into register file by direct access to 64-bit SDRAM data bus in 40 clock cycles. The search costs 256 clock cycles by serially feeding data through register chain into search array. The low level search is finished within 296 cycles. Fig. 9(b) shows 1-bit simple operation in each PE.

In the register file, two ports $c11$ and $c12$ feed search window data to PEs array for first row search points. $c21$ and $c22$ operate for second row search points, and so on. During search, the data serially flows in the register chain. But before the search, 7 lines of data must be prepared into these registers by parallel access of SDRAM.

IV. VLSI Implementation and Experimental Results

We use 0.5u 2-level metal CMOS technology. Overall high level simulation are performed by Synopsys. Since the search array are are very regular, we use full custom design with Cadence to result in compact layout. Control part are irregular so that we directly synthesize it into custom cell and layout with Cadence Cell Ensemble to reduce design time. Memory are designed with Mentor Memory Builder. At last, all part are assembled by Cadence Block Ensemble. The design costs 11 month× person. The chip area for each component is shown in Table I. Because of bits truncation, the area of PE2 is four times as small as that of PE1. Our chip size is 1/16 times as small as the conventional one[3] with better performance. The timing chart is shown in Fig. 5. On the average of 296 cycle, we can finish one motion vector.

V. Conclusion

In this design, we present a dedicated motion estimator LSI for HDTV purpose. The results indicate that it has very high cost-efficiency and good performance. We will use the similar design method to include half-pel search. The result in this design shows our promising future to realize one chip HDTV MPEG2 encoder.

References

[1] Li Jiang, Kazuhito Ito, Hiroaki Kunieda, "Bits Truncation adaptive Pyramid algorithm for Motion Estimation of MPEG", *IEICE Trans. Fundamentals,*, Vol. E80-A, NO.8, pp.1438-1445, Aug. 1997.

[2] Li Jiang, Dongju LI, Shintaro HABA, Chawalit Honsawek and Hiroaki KUNIEDA, "Dedicated Design of Motion Estimator with Bits Truncation Fast Algorithm", *IEICE Trans. Fundamentals,*, Vol. E80-A, NO.8, Aug. 1998.

[3] T.Onoye, G.Fujita, I. Shirakawa, "Single chip implementation of motion estimator dedicated to MPEG2 MP@HL", *IEICE Trans. Fundamentals,*, Vol.E79-A, No.8, pp.1210-1216, Aug. 1996.

[4] J.P.Berns, T.G.Noll, "A Cascadable 200 GOPS Motion Estimation Chip for HDTV Applications", *Proceedings of CICC.*, pp.355-358, 1996.

[5] Dongju Li and Hiroaki Kunieda, "Memory sharing processor array(MSPA) Architecture", *IEICE Trans. Fundamentals,* , Vol.E79-A, No.12, Dec. 1996.

[6] Dongju Li, Li Jiang, Tsuyoshi Isshiki, and Hiroaki Kunieda, "Array Architecture and Design for Image Window Operation Processing ASICs", *ISCAS'98.*, May, 1998.

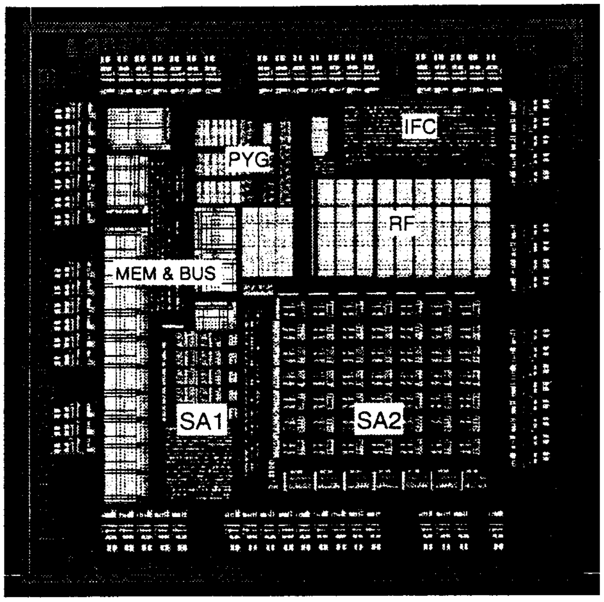


Fig. 1. : The final layout of single chip motion estimator

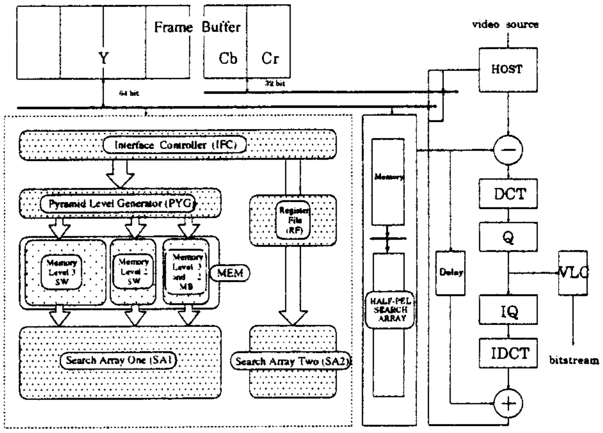


Fig. 3. : Overview of motion estimator

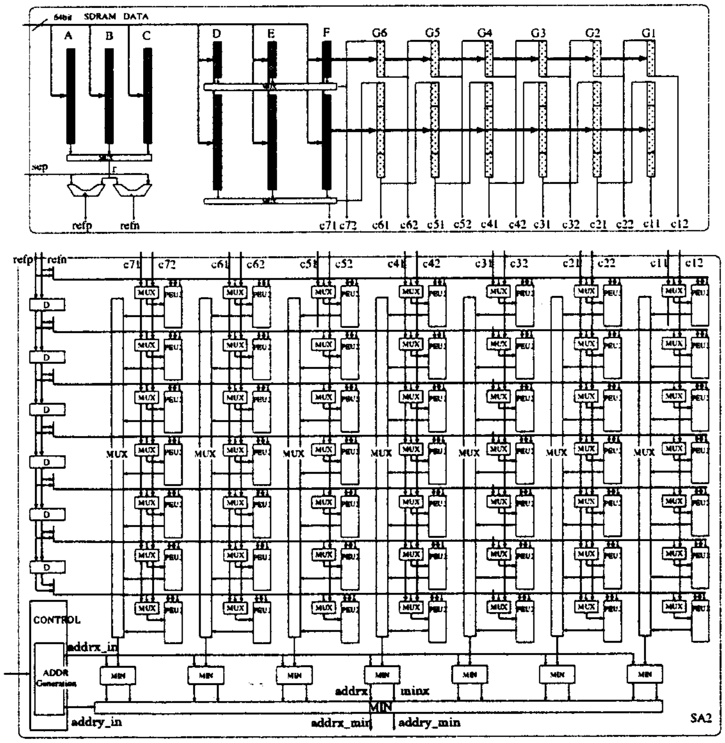


Fig. 2. : Search Array II Architecture

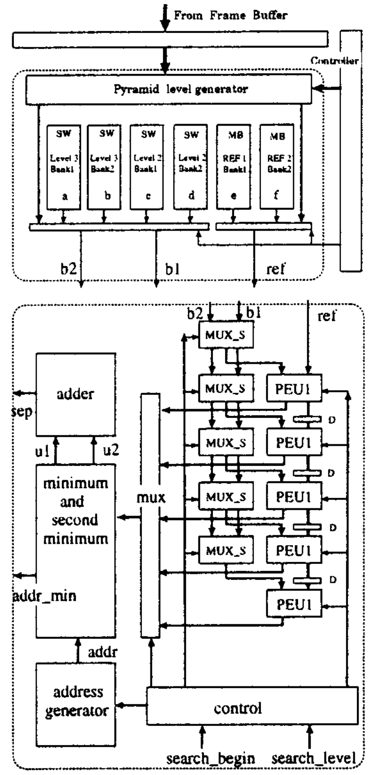


Fig. 4. : Search Array I Architecture

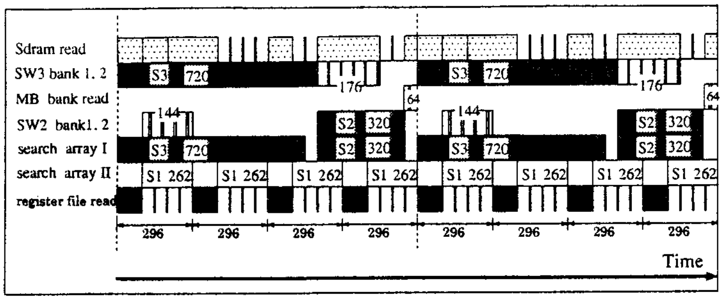
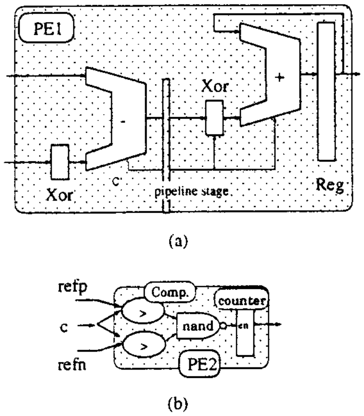


Fig. 5. : The overall timing chart



(a)

(b)

Fig. 9. : PE1 and PE2

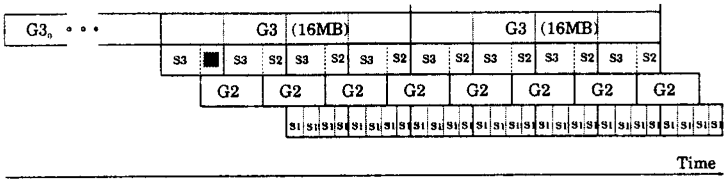


Fig. 6. : Scheduling of pyramid search

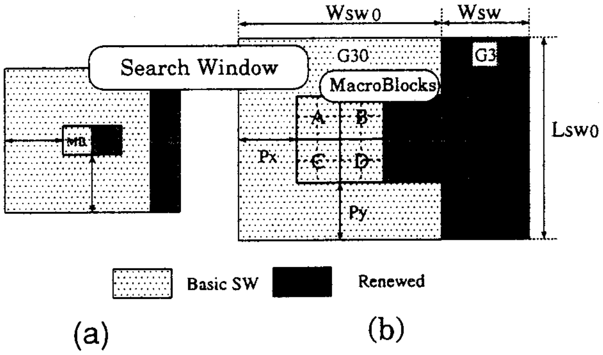


Fig. 7. : The reuse of search window among neighboring MBs

Table. I: AREA FOR EVERY PART OF CHIP

search array I	750mm × 1470mm (PE1: 700 × 150)
search array II	2200mm × 1750mm (PE2: 175 × 150)
memory level 4	400mm × 2900mm (780 Bytes)
memory level 3	400mm × 815mm (180 Bytes)
memory ref	400mm × 560mm (160 Bytes)
register file	900mm × 1850mm (246 Bytes)
Interface controller	640mm × 1500mm
Pyramid generator	850mm × 800mm

Table. II: MAIN FEATURES OF THE MOTION ESTIMATOR

Die size	4.8mm × 4.8mm
Clock rate	Core : 41.5 MHz, I/O : 83 MHz
NOP	5 GOP(41.5 MHz), 10 GOP(83 MHz)
Transistors	198,200
Package	108 pins (76 signal pins)
	input 35 pins, output 35 pins
Image size	1920 × 1152
Frame rate	30 field / s
Search range	± 67

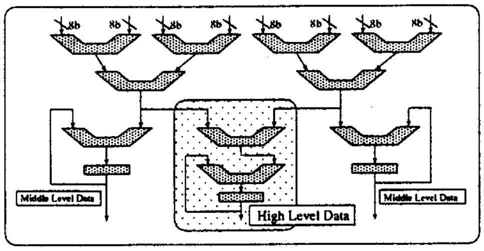


Fig. 8. : Pyramid Level Generator