

SCALABLE MODULE-BASED ARCHITECTURE FOR MPEG-4 BMA MOTION ESTIMATION

Mei-Yun Hsu, Hao-Chieh Chang, Yi-Chu Wang and Liang-Gee Chen

DSP/IC Design Lab
Department of Electrical Engineering, National Taiwan University
1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan
Email : {yun,howard,lgchen}@video.ee.ntu.edu.tw

ABSTRACT

In this paper, we present a scalable module-based architecture for block matching motion estimation algorithm of MPEG-4. The basic module comprises one set of processing elements based on one-dimensional systolic array architecture. To support various applications, modules of processing elements can be configured to form the processing element array to meet the requirements, such as variable block size, search range and computation power. And this proposed architecture has the advantage of few I/O port counts. Based on eliminating unnecessary signal transitions in the processing element, power dissipation of datapath can be reduced to about half without decreasing the picture quality.

1. INTRODUCTION

In video systems, motion estimation is a widely adopted technique to explore the temporal redundancy of sequences. Full search block matching algorithm is commonly used in motion estimation. Because of the requirement of high computation power, dedicated hardware is usually employed.

For various video applications in the present and future, the architecture of motion estimation should be more flexible to support different requirements. Many previous works have reported the similar approaches. Some designs aim to support different block size and search range by modifying architecture parameters and cascading [1]-[8]. Some of these architectures are based on two-dimensional systolic array [6][8]. In the condition of larger block size, these designs have to spend much more resources. In addition, two-dimensional systolic array often needs to access many data elements once at some cycles, hence the wordwidths or ports of memory would be large. The increment of memory ports would influence delay and area of memory significantly [9]. Some are based one-dimensional systolic array [10] [5] [7], but they may require various processing elements designed with irregular data flow or use more registers. This will lead to more power consumption and larger chip area. As a result, a scalable architecture based on one-dimensional systolic array module with fewer registers and regular data flow is proposed, and the port number can be reduced by well-arranged data flow. Besides, unnecessary switchings of circuits in processing elements are eliminated to reduce the power consumption of datapath.

The organization of this paper is as follows. In section II, we briefly review the MPEG-4 motion estimation and its computation analysis. In section III, the scalable module-based architecture is presented. In section IV, the comparison results of the proposed

Table 1: Evaluation of Computation Load

sequence	giga operation per second (GOPs)	operation ratio(%) (/rectangular)	% of opaque MB	% of boundary MB
rectangular	18.398	100	100	0
weather	12.787	69.50	46.03	28.16
news	11.211	60.93	39.02	26.29
children	6.435	34.98	7.37	33.12

architecture with other designs found in the literature are represented. Finally, section V concludes this paper.

2. MPEG-4 MOTION ESTIMATION

In MPEG-4 [11], content-based representation is employed. For motion estimation of arbitrarily shape video object (VO), the SAD calculation of block matching has to be modified. Only the errors that locate inside video object are accumulated. The formula is as follows,

$$SAD(x, y) = \sum_{i=1}^N \sum_{j=1}^N |current - reference| \times (\alpha_{original} \neq 0)$$

where N is block size,

$\alpha = 1$ (inside object) or 0 (outside object).

Motion Estimation is performed for macroblock that is entirely inside object or lies on the boundary of object. In the following, the computation requirements of three video objects are calculated for MPEG-4 Core Profile Level 2 (CPL2). In MPEG-4 CPL2, the maximum number of macroblock per second is 23760, and the typical visual session size is CIF (352×288). Assume that the search range is [-16,15], and the type of macroblock is already known. The calculation of SAD is counted three operations. In boundary macroblock there is an additional operation to check whether the pixel lies in the object or not. If the pixel is not in the object, operations of SAD are not counted. The result of operation analysis for four sequences is shown in Table 1. In general, the computation load of object based motion estimation would be lower than the one of rectangular frame. The percentage of reduction depends on the characteristic of video object. However, there may be multiple visual objects in a scene, so total computation load would depends on the object number, too.

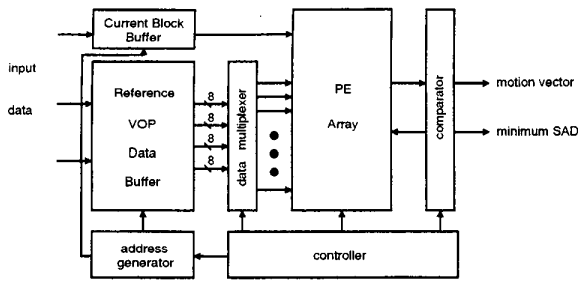


Figure 1: Block Diagram of Motion Estimation Core

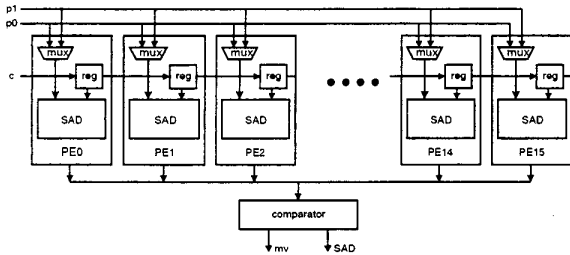


Figure 2: Block Diagram of One-Dimensional Systolic Array

3. MODULE-BASED ARCHITECTURE FOR MPEG-4 MOTION ESTIMATION

3.1. MOTION ESTIMATION CORE

Fig. 1 depicts the proposed architecture of motion estimation core. This architecture mainly includes two data buffers, processing element (PE) array, flexible address generator, controller, data multiplexer and comparator. Two data buffers are utilized to store current block and reference data, respectively. The current block buffer stores both texture and shape of current block. The reference VOP data buffer exports at most four different pixels at each clock cycle, and data multiplexer outputs proper data to every PE module according to the configuration of PE array. The comparator finds minimum of accumulated errors from PE array and calculates the corresponding motion vector.

3.2. PROCESSING ELEMENT MODULE

The processing element module is based on one-dimensional systolic array [12]. The block diagram of one-dimensional systolic array with 16 PEs is shown in Fig. 2. Reference data, p_0 and p_1 , are broadcasted to every PE, and current data propagate through PE array every clock cycle. Every PE calculates SAD of one specific candidate of motion vector in the search range. A PE module is responsible for one row of candidates. Fig. 3 illustrates which rows of the reference VOP data are accessed every clock cycle.

3.3. SCALABLE ARCHITECTURE

For various applications, motion estimation core needs to be scalable to support different block size, search area, and operation frequency. In the following, we show that how the module can be cascaded to support various requirements. Assume that a PE module can handle $N \times N$ block and the search range is $[-P, P - 1]$.

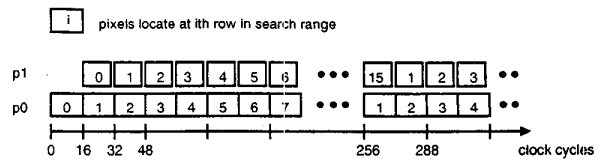


Figure 3: Illustration of Reference Data (p_0 and p_1) Source for One PE Module

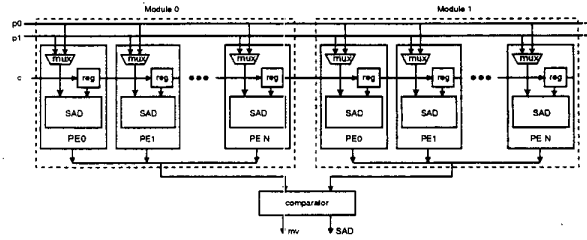


Figure 4: Case I: larger block size

3.3.1. Case I: larger block size

If the block size increases from $N \times N$ to $2N \times 2N$, two PE modules can be connected to keep data flow the same. Fig. 4 shows the connected module architecture. The reference data of two PE modules are the same, and the current block data propagate through modules. If the search range, frame size, and frame rate remain the same, the change of block size would not influence the total operation amount per second. Because the number of PE is double, the frequency of operation would become half of the original one for the same throughput.

3.3.2. Case II: increasing search range

For search range ($2P$) that is a multiple of N , one motion vector is generated every $(2P)^2 \times N$ cycles. Now the search range is increased from $[-P, P - 1]$ to $[-2P, 2P - 1]$. If only one PE module is used, the cycles of generating a motion vector would become $(4P)^2 \times N$. If we want to maintain the same operation frequency and throughput, the number of PE modules has to increase to four. Every module is responsible for different rows of motion vector candidates. The cascaded module architecture is depicted in Fig. 5. And Fig. 6 shows the distribution of motion vector candidates in every PE module under the assumption of $N = 4$, $P = 2$. The timing of each PE module would delay 16 cycles after the for-

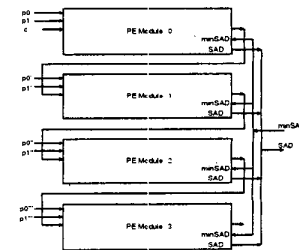


Figure 5: Case II: increase search range

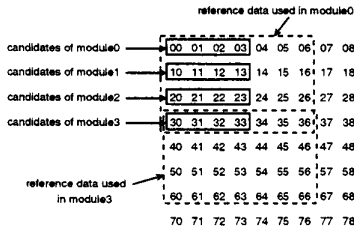


Figure 6: Distribution of motion vector candidates.

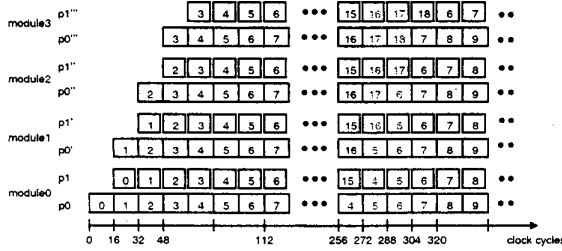


Figure 7: Illustration of Reference Data Source for Four PE Modules

mer PE module for $N=16$. Fig. 7 depicts the rows that reference VOP data locate every cycle for four modules. At each cycle, PE array accesses at most four different pixels. The maximum number of pixels accessed at a clock cycle can be limited to four if the number of PE modules is not larger than N .

3.3.3. Case III: lower operation frequency

Through increasing the number of PE module, the operation frequency can be lowered. As mentioned above, we can get one motion vector every $(2P)^2 \times N$ cycles for one PE module. For the same amount of operation, doubling the number of PE module would decrease the operation frequency to half, and every $(2P)^2 \times N/2$ cycles a motion vector is generated.

3.4. PE WITH POWER-SAVING OPERATION

According to the block matching criterion, the candidate with the smallest SAD would be selected. If the accumulating SAD of the candidate were larger than the present minimum SAD, we can stop calculating this candidate further. By eliminating the unnecessary operations, the operation amount of motion estimation can decrease while preserving the optimal picture quality. The experiment result is shown in Table 2 and Fig. 9. The amount of operation reduces to about 40% to 60% of original one. For the larger search range, the percentage of remaining operations would be lower. For sequence with lower amount of movement and lower spatial detail, like "hall" and "mother and daughter", the amount of eliminated operations would be larger.

This power-saving concept is implemented in the PE design. As shown in Fig. 8, two registers (shaded blocks) are employed in the PE to store current block and reference data. In normal mode, these registers are transparent. Input data are directly bypassed to the circuit for SAD. As the accumulated SAD is larger than the present minimum SAD, PE would receive the "gated" signal

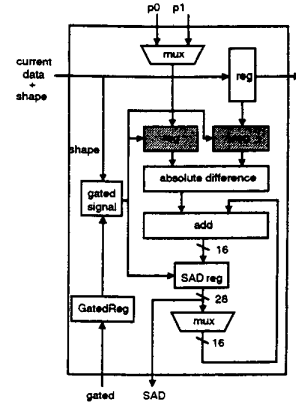


Figure 8: Block Diagram of Processing Element

Table 2: Percentage of Remaining Operations

sequence	operation%/(original)	
	search range	
	[-8,7]	[-16,15]
hall	49.48	40.29
mother&daughter	49.69	39.34
foreman	54.16	43.04
football	65.46	55.99
bream	52.57	43.56
children	50.43	37.75
weather	40.65	34.16

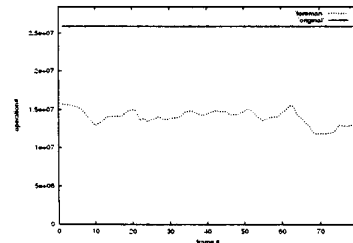


Figure 9: Operation # of Foreman with Search Range [-8, 7]

from the comparator in PE module. This signal would be stored in the register, GatedReg. The SAD register and block data registers would be gated. For the remaining clock cycles of this candidate, the circuit for calculating SAD would not switch any more. Therefore, the power dissipation in PE can be reduced. When the PE begins to process new candidate, GatedReg will be clear, and let the circuit come back to work.

To support MPEG-4 polygon matching, the shape information of current block is required during processing boundary macroblock. When the pixel doesn't locate in video object, the two extra registers will hold the previous input data and the register of SAD will hold the value of previous accumulated sum.

4. PERFORMANCE COMPARISON AND DISCUSSION

In this section, we present some comparisons among the proposed architecture and other designs. Because no bubble cycles are re-

Table 3: The Wordlength of Register Used in a PE Module with N Processing Elements (K is search range)

	register (bits)
Nam [5]	$N \times 4 \times 8 + K \times 16$
He [6]	$T_h \times 16 + N \times 2 \times 8 + T_h \times N \times 20$
Chang [4]	$N \times 8 + N \times 12 + K \times 16$
ours	$N \times 16 + N \times 8$

Table 4: The Amount of Memory Access for a Block based on N PEs ($N = 2P$ and K is search range)

design	memory access for one block (bytes)	
	$K = 2P$	$K = 4P$
Nam [5]	$(K + N - 1) \times N \times K$	
He [6]	$K \times K \times (N + T_h - 1) \times (N + T_v - 1) / (T_h \times T_v)$	
Chang [4]	$(K + N - 1) \times N \times K$	
ours	$(2N - 1) \times N \times K$	$(2N - 1) \times N \times K \times 2$

quired when changing candidates or blocks in these architectures, the cycles needed to produce one motion vector would be the same under the same number of PEs. However, for different strategies of data flow, the number of registers used in a PE module would be very different. For general operations, the two extra registers are not included in our PE, and the wordlength of SAD register would be 16. The total wordlength of registers used in a PE module with N processing elements are compared. Table 3 lists the analysis result. T_v and T_h are the vertical and horizontal dimensions of the tile respectively defined in [6]. Table 4 shows the analysis result of memory access from on-chip buffer to PE array based on 16 PEs. Because of flexible designs, two specifications are used to compare the performance of these designs. One is 16 PEs with search range [-8,7]. The other is 64 PEs with search range [-16,15]. Table 5 shows the result. We can find that the proposed architecture can use fewer registers in both cases. And the amount of memory access is acceptable. For larger search range, the amount of memory access wouldn't increase substantially.

5. CONCLUSION

In this paper, we have presented a scalable motion estimation architecture for MPEG-4. The architecture is based on modules of one-dimensional systolic array. Through cascading of multiple modules, different processing element arrays can be constructed to meet various applications, such as larger block size, search range and various operation frequencies. Through well-arranged data flow, the number of I/O ports is reduced. Using simple termination judgment to eliminate unnecessary switching of circuits, processing element with power saving is achieved. The power dissipation

Table 5: Comparison of Designs for Two Specifications

design	16 PEs with SR [-8,7]		64 PEs with SR [-16,15]	
	register# (bits)	BW (bytes)	register# (bits)	BW (bytes)
Nam [5]	768	7936	4096	6016
He [6]	1600	19456	6400	23104
Chang [4]	576	7936	3328	6016
ours	384	7936	1536	7936

in datapath becomes about 50% of that of the conventional systolic array. And it can support polygon matching in MPEG-4 standard.

6. REFERENCES

- [1] L. D. Vos and M. Stegherr, "Parameterizable VLSI architectures for the full-search block-matching algorithm," *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 1309-1316, Oct. 1989.
- [2] H. Yeo and Y. H. Hu, "A novel modular systolic array architecture for full-search block matching motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 407-416, Oct. 1995.
- [3] L. D. Vos and M. Schobinger, "VLSI architecture for a flexible block matching processor," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 417-428, Oct. 1995.
- [4] S. Chang, J.-H. Hwang, and C.-W. Jen, "Scalable array architecture design for full search block matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 332-343, Aug. 1995.
- [5] S. H. Nam and M. K. Lee, "Flexible VLSI architecture of motion estimator for video image compression," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, vol. 43, pp. 467-470, June 1996.
- [6] Z. L. He and M. L. Liou, "Cost effective VLSI architecture for full-search block-matching motion estimation algorithm," *Journal of VLSI Signal Processing*, vol. 17, pp. 225-240, Nov. 1997.
- [7] S. H. Nam and M. K. Lee, "High-throughput block-matching VLSI architecture with low memory bandwidth," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, vol. 45, pp. 508-512, Apr. 1998.
- [8] Y.-H. Yeh and C.-Y. Lee, "Cost-effective VLSI architectures and buffer size optimization for full-search block matching algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 345-358, Sept. 1999.
- [9] S. Dutta, K. J. O'Connor, W. Wolf, and A. Wolfe, "A design study of a 0.25 μ m video signal processor," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 501-519, Aug. 1998.
- [10] S. H. Nam, J. S. Baek, and M. K. Lee, "Flexible VLSI architecture of full search motion estimation for video applications," *IEEE Transactions on Consumer Electronics*, vol. 40, pp. 176-184, May 1994.
- [11] *JTC1/SC29/WG11, N2502a, Generic Coding of Audio-Visual Objects: Visual 14496-2, Final Draft IS*. Atlantic City: ISO/IEC, 1998.
- [12] K.-M. Yang, M.-T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 1317-1325, Oct. 1989.