A NOVEL VLSI ARCHITECTURE FOR BLOCK MATCHING ALGORITHMS*

Chen-Yi Lee

Dept. of Electronics Engineering, National Chiao Tung University,
1001, University Road, Hsinchu 300, Taiwan, ROC
Tel: 886-35-731849; Email: cylee@cc.nctu.edu.tw

ABSTRACT

This paper presents a new VLSI architecture for full search block matching motion estimation (ME) algorithm. The proposed VLSI architecture has three specific features: (1) it has a processor element (PE) array which provides sufficient computational power and achieves 100% hardware efficiency, where PE's work in a systolic style; (2) it contains stream memory banks which provide scheduled data flow needed in PE for computing mean absolute error (MAE); and (3) it has minimal memory access bandwidth to save I/O pin-count. As a result, the proposed architecture allows to reach cost-effective ME hardware solution.

1. INTRODUCTION

Block matching algorithms (BMA) are often found in image and video applications for pattern analysis, motion detection, data compression, ... etc. Inherent computational complexity in these algorithms often demand special hardware to meet real-time performance. Motion estimation (ME) based on full-search BMA (FSBMA), one of the key techniques in video coding to remove redundancy, is a good example for illustration of such design complexity. Many research results on VLSI architectures for FSBMA were proposed in the past [1, 2, 3, 4, 5, 6, 7, 8, 9], where research efforts have moved from meeting computational requirements to reducing memory bandwidth as well as providing multifunctions specified in standard [10]. This trend results from the consideration that, as VLSI technology advances, more devices can be integrated on-chip to cope with computational requirement while cost of chip I/O still remains very high. To cope with computational complexity, systolic array (SA) architecture [11, 12] is often exploited. The number of processor elements (PE) is often dependent on the size of reference block and pixel rate. To reduce I/O bandwidth, search data are often organized in a snake-like form to run through processor element array as found in [2, 3, 4, 13] or in an interlacing style as in [6]. Although I/O pin-count is reduced, the time to initialize both reference and search data (i.e. before PE array can be fully operated) is too long and hence overall performance becomes degraded further. For example, for 16X16 reference block and [-16,+15] search area, it may take more than half of the total cycle count for data initialization [3, 4]. In addition, for block matching at search boundary, some PE's may become idle because of pipeline filling [2], which also leads to degrade system performance.

*WORK SUPPORTED BY THE NATIONAL SCIENCE COUNCIL OF TAIWAN, ROC.

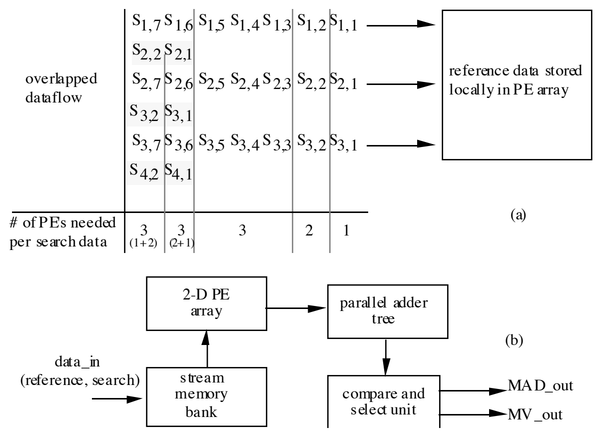

Fig.1. (a) shows an overlapped search data flow and (b) shows the block diagram for the ME processor.

In this paper, we present a novel VLSI architecture to solve these problems. By partitioning search area and including stream memory banks, the above problems can be solved. In section 2, we first describe our previous work on ME archiecture which is based on semi-systolic array (SSA). Then we present a new SA-based ME architecture which realize same search data flow in section 3. A systematic design approach is derived in section 4, where both SSA and SA can be derived for different mapping schemes.

2. PREVIOUS WORK ON ME PROCESSOR

Previously we presented a semi-systolic array (SSA) style for realizing an overlapped data flow of search area [14] (as shown in Figure 1(a)), where a stream memory bank is designed to buffer part of the search area so that both I/O pin-count and data initialization delay can be minimized. A dedicated memory module for such an application was also developed [13]. In SSA-based architecture, reference data are locally stored in the PE array, while stream memory bank provides the

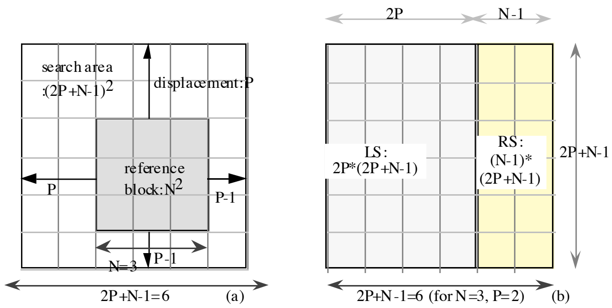

Fig.2. Illustration example for the proposal architecture: (a) 3X3 reference block (N=3) and 6X6 search area (P=2); (b) search area is partitioned into LS and RS two regions.

required search data which are scheduled to meet computational requirement.

The features of this architecture are summarized as follows:

- include stream memory bank to enhance data usage as well as to reduce memory bandwidth;
- search data are broadcast while partial MAE are locally communicated within PE array;
- apply double buffer strategy to minimize initialization phase and double data buses to overcome idle operations when boundary candidate is detected.

3. THE PROPOSED SA ME ARCHITECTURE

However we have recently developed a new systolic architecture to implement the overlapped data flow. We will use the example (N=3 and P=2) shown in Figure 2(a) to illustrate the mapping process. From the overlapped data flow shown in Figure 1(a), it can be found that search data from two different rows (or columns) are needed as boundary candidate is detected. This implies that two input data buses are needed, where one is always busy and the other is needed only when boundary candidate is detected. Therefore how to schedue search data into the PE array so that MAE at each PE can always be calculated becomes an important issue in systolic array (SA) based realization. This problem can easily be solved by taking into account data flow needed in the PE array.

Figure 3 shows the search data needed in each PE for the first 6 cycles. The marked data items indicate they are from different row (or column) and thus two input buses are requested. If we can map search data into PE array in that way, all PEs are working on correct data items, resulting in $(2P)^2$ cycles in obtaining an MV. This also implies that search area should be partitioned into two regions, which are named LS and RS respectively as shown in Figure 2(b). Both regions have the size of (2P)*(2P+N-1) and (N-1)*(2P+N-1) respectively. This partitioning method is very straightforward because it is based on the reference block size N only. Thus all search data are only accessed once and then buffered in stream memory bank for other necessary computations. This method is different from overlapped partitioned regions [4] which may increase I/O bandwidth. The stream stream used here can be either shifter register array or pointer addressable memory (PAM) [13]. The overall SA-based architecture is depicted in Figure 4. Note that the required skew for search data is achieved by both steam memory and PE array since search data on the lower row is always one clock ahead of its upper row. In that case, partial MAE's of each candidate, which are obtained row by row, can be matched.

cycle	1	2	3	4	5	6
requested search data in PE array	X, X, 11,	X, 11, 12,	11, 12, 13	12, 13, 14,	13, 14, 15	14, 15, 16,
	X, 21, 22,	21, 22, 23,	22, 23, 24	23, 24, 25	24, 25, 26	31, 32, 33
	31, 32, 33	32, 33, 34	33, 34, 35	34, 35, 36	41, 42, 43	42, 43, 44
candidate distortion available at cycle#					(-2,-2)	(-2, -1)

2 cycles delay *

* 2 cycles delay at Parallel adder tree, compare and select

Fig.3. An overlapped search data flow is developed to allow each PE handling correct data items based on systolic array architecure.

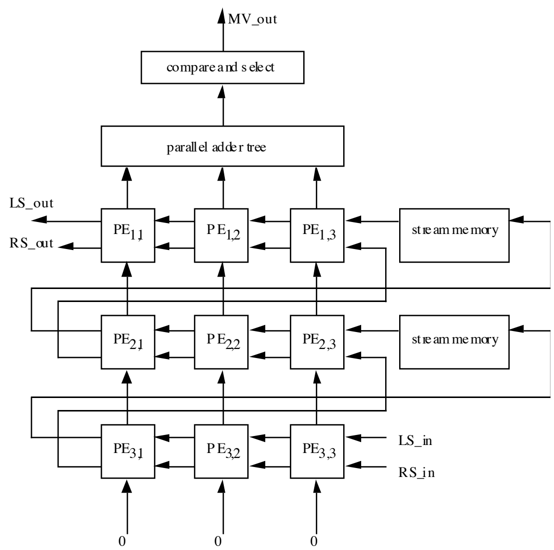

Fig.4. Systolic array architecture for the FSBMA.

PE Design: To meet real-time performance in video coding, very often 2-D array structure is requested for ME processor. Based on the above description, we propose a double-port (LS and RS) structure as shown in Figure 5. Here search data run horizontally, MAE and control signal ($cmux$) run vertically. In the initialization phase, all internal registers except MAE accumulation register acc are defined appropriately through two data buses. The control register sel is used to select either LS or RS for MAE computation.

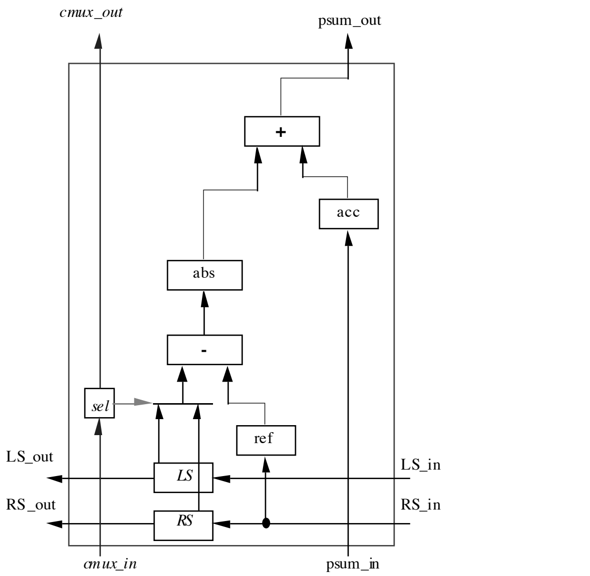

Fig.5. The structure of processor element for systolic array based architecture.

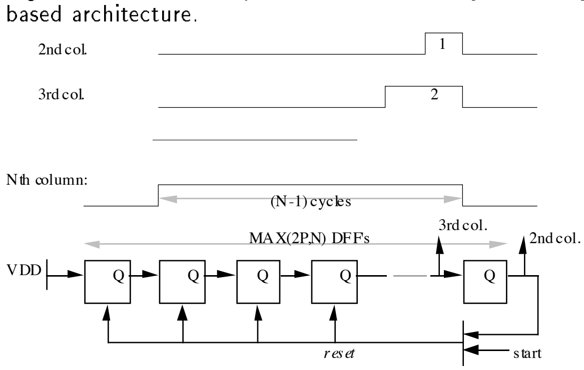

Fig.6. Ring-counter based controller for the PE array of systolic array architecture.

Controller Design: It can be found that from Figure 3, the control signals needed for each PE are stair-like waveform which can easily be obtained by a ring-counter as shown in Figure 6. As MV search is activated, this ring-counter generates the required control signals which are then connected to (N-1) PE columns. Note that this ring counter is reset every 2P cycles.

Performance Evaluation: Here we summarize the performance of the SA-based ME architecture, where reference size of NXN and search range of [-P,P-1] are assumed.

- initialization phase: it takes 2P*(N-1) cycles to preload both reference data and search data;
- execution phase: it takes $(2P)^2$ cycles to find one MV, thus the PE array efficiency[1] becomes 68% for N=P=16;

[1] The efficiency can achieve 100% if double buffering is exploited, i.e. use another buffer to preload next reference and search data. However this needs to revise PE structure and makes the controller more complex.

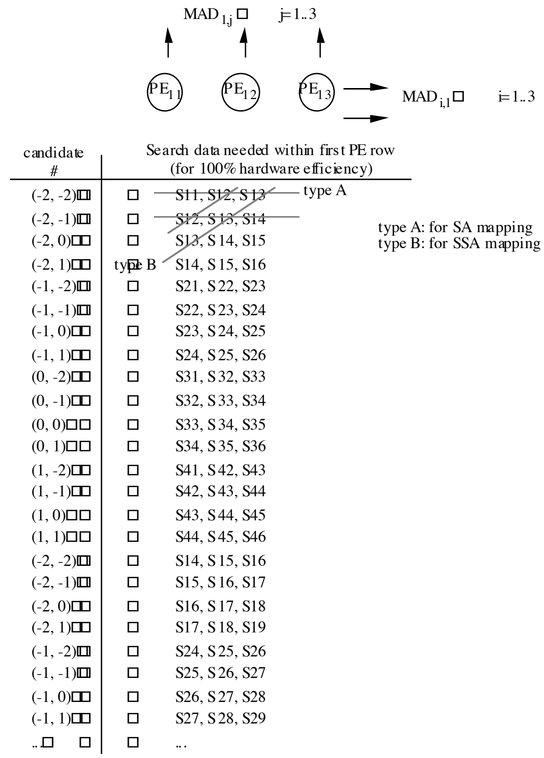

Fig.7. A systematic mapping procedure for FSBMA ME processor design.

- stream memory size: it only needs (N-1)x(2P-N+1) storage space to buffer part of LS search area[2];
- pipelinable design: pipeline register can be directly inserted to enhance clock rate.

4. DISCUSSION

Compared to other ME architectural solutions found in the literature [1, 2, 4], our proposal not only achieves optimal hardware efficiency within PE array but also requires minimal memory bandwidth. Also the proposed architecture which combines systolic array and stream memory makes it very suitable for high sample rate video applications.

From the description of algorithm mapping and architecture design, it can be found that good data partitioning may result in reducing memory bandwidth and I/O pin-count without degrading overall system performance. In addition, the same data flow can be mapped onto different architectures as shown in Figure 7, where idle operations can be avoided if both initialization delay and boundary search data are handled appropriately. Below we summarize the key procedures for designing ME processors based on FSBMA:

- **search area partitioning:** The first step is to partition search area such that all PE's work all

[2] Part of the RS and LS search area is buffered in the PE array.

the time once MV search is activated. This is often found in boundary candidate whose search data are reused less than N times. Therefore these search data have to be overlapped with other search data to ensure that all PE's are working on correct data items. In principle, search area is partitioned into two regions whose size are $(2P+N-1)\times(2P)$ and $(2P+N-1)\times(N-1)$ respectively.

- **ME processor structure:** The ME processor mainly consists of 4 blocks, namely stream memory bank, PE array, parallel-adder-tree, and compare-select unit as shown in Figure 1(b). If search data flow and partial MAE accumulation are running in same direction, the SSA architecture solution is proposed. On the other hand, if they run orthogonally, the SA architecture is proposed. Total storage space[3] needed for both search area and reference block is $4N^2$ + (N-1)(2P+2-N). Without counting the initialization delay, each MV can be obtained every $(2P)^2$ cycles. As for the controller, ring-counter-based structure is proposed.
- **Data initialization phase:** In practical applications, we have to take the initialization delay into account because only when both search and reference data are placed appropriately can MV search be activated. If the initialization phase takes too many cycles, the overall performance becomes degraded. In our proposals, both architectures need (N-1)*(2P+1) cycles to preload search and reference data through two input buses. Thus it totally takes $(2P)^2$ + (N-1)*(2P+1) cycles to find one MV. For N=P=8, the PE array becomes idle about one third of the total cycle count. To solve this initializaton delay problem, a double buffering scheme is suggested. Although this result in the need of doubling storage space and three input buses, i.e. one for reference block and the other two for search area, the cycle-count needed to obtain one MV still remains $(2P)^2$ cycles.

5. CONCLUSION

In this paper, we have presented a novel VLSI architecture for optimally implementing full-search block matching algorithm. The proposed architecture mainly consists of (1) PE-array which is a systolic array structure to offer computation power and (2) stream memory bank which offers a scheduled search data flow so that 100% hardware efficiency within the PE-array can be achieved. As a result, both memory bandwidth and I/O pin-count can be minimized. In addition, a systematic approach for the design of FSBMA ME processor has also been derived. We are currently investigating the possibility of hierarchical search and scalable solution for large search area based on this proposed architecture.

REFERENCES

[1] *"VLSI Architectures for Estimation of Translatory Motion"*, Special Issue on VLSI Implementation for Digital Image and Video Processing Applications, IEEE Trans. on CAS, Vol. 36, No. 10, Oct. 1989.

[2] C.H. Hsieh and T.P. Lin, *"VLSI Architecture for Block-Matching Motion Estimation Algorithm"*, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 2, No. 2, June 1992, pp. 169-175.

[3] S.I. Uramoto, A. Takabatake, M. Suzuki, H. Sakurai, and M. Yoshimoto, *"A Half-Pel Precision Motion Estimation Processor For NTSC-Resolution Video"*, In IEEE 1993 Custom Integrated Circuits Conference, San Diego, CA, May 9-12, 1993, pp. 11.2.1-11.2.4.

[4] J. Baek, S. Nam, M. Lee, C. Oh, and K. Hwang, *"A Fast Array Architecture for Block Matching Algorithm"*, in Proc. of ISCAS'94, May 30 - June 3, 1994.

[5] K. Ishihara, S. Masuda, S. Hattori, H. Nishikawa, Y. Ajioka, T. Yamada, H. Amishiro, and M. Yoshimoto, *"A Half-Pel Precision MPEG2 Motion Estimation Processor with Concurrent Three-Vector Search"*, in Proc. of ISSCC'95, Feb. 1995, pp. 288-289.

[6] C.L. Wang, K.M. Chen, and J.M. Hsiung, *"A High-Throughput and Flexible VLSI Architecture for Motion Estimation"*, in Proc. of ICASSP'95, Detroit, May 8-12, 1995, pp. 3295-3298.

[7] S. Chang, J.H. Hwang, and C.W. Jen, *"Scalable Array Architecture Design for Full Search Block Matching"*, IEEE Trans. on CAS for Video Technology, Vol. 5, No. 4, Aug. 1995, pp. 332-343.

[8] H. Yeo and Y.H. Hu, *"A Novel Modular Systolic Array Architecture for Full-Search Block Matching Motion Estimation"*, IEEE Trans. on CAS for Video Technology, Vol. 5, No. 5, Oct. 1995, pp. 407 - 416.

[9] L. De Vos and M. Schobinger, *"VLSI Architecture for a Flexible Block Matching Processor"*, IEEE Trans. on CAS for Video Technology, Vol. 5, No. 5, Oct. 1995, pp. 417 - 428.

[10] *"International Organization for Standardization Coding of Moving Pictures and Associated Audio"*, ISO/IEC JTC1/SC29/WG11/N702, March 25, 1994.

[11] H. T. Kung, *"Why Systolic architectures"*, IEEE Computer, Vol. 15, no 1, Jan., 1982.

[12] S.Y. Kung, *"VLSI Array Processors"*, Prentice-Hall International Editions, 1988.

[13] G.L. Tzeng and C.Y. Lee, *"An Efficient Memory Architecture for Motion Estimation Processor Design"*, in Proc. of ISCAS'95, April 29 - May 3, 1995.

[14] M.C. Lu and C.Y. Lee *"Semi-Systolic Array Based Motion Estimation Processor Design"*, in Proc. of ICASSP'95, Detroit, May 18-12, 1995.

[3]The storage space includes registers within PE array and the stream memory bank.