A new concept of a mono-dimensional SIMD/MIMD parallel architecture based in a content addressable memory

Domingo Torres[1], Hervé Mathias[2], Hassan Rabah[2], and Serge Weber[2]

[1]Programa de Graduados e Investigación en Ingeniería Eléctrica
Instituto Tecnológico de Morelia, México
Av. Tecnológico 1500 Morelia Mich., MEXICO C.P.58600

[2]Laboratoire d'Instrumentation Electronique
Nancy (L.I.E.N), University of Nancy I France,
BP 239, 54506, Vandoeuvre Cedex, FRANCE

Abstract. A new concept of parallel architecture for image processing is presented in this paper. Named LAPMAM (Linear array processors with Multi-mode Access Memory), this architecture, for a 512 x 512 image, has 512 processors and four memory planes each of 512^2 memory modules. One important characteristic of this architecture is its memories modules that allow different access modes: RAM, FIFO, normal CAM and interactive CAM. This particular memory and a linear structure of RISC processors are combined with a tree interconnection network to obtain very efficient 1-d architecture suitable for real time image processing. The processor works in SIMD mode and allows a restricted MIMD mode without requiring a great hardware complexity. A hardware simulation of an architecture prototype has been accomplished to test its performance in low and intermediate level vision tasks. The performance of the LAPMAM is compared with that of different architectures.

Key-words: Parallel architecture, image processing, content addressable memory, SIMD processors, MIMD processors, interconnection network.

1. Introduction

The computational demands of image processing have generated a large number of research work and led to numerous architectures and algorithms [1, 2, 3]. Sequential machines, which require an excessive amount of time, are not well suited to solve these problems. Consequently, the potential of parallelism has been highly exploited. Among the existing topologies, the most attractive in terms of processor x time complexity [4] is the 1-d array processor family. The 1-d architectures are also, en general, the cheapest topologies. However, because of a lack of parallelism in the memory rows, they have a lower performance than more complex architectures (2-d architectures example). To solve this problem, we have developed systems based on CAM memories: an efficient architecture dedicated to labeling and a more general architecture called LAPCAM (Linear Array Processors with Content-Addressable Memory) [5]. We have redesigned the LAPCAM architecture, particularly the Multi-mode Access Memory (MAM) modules to enable the hardware implementation. The CAM mode of this memory enhances the parallelism in the memory rows of the linear arrays. In this architecture, a new concept of SIMD/restricted MIMD processors is also proposed. The processor has the SIMD structure with its typical advantages (simple implementation, high performance and no memory access conflicts), but also can work like a MIMD processor, taking a limited conditional decision with a simple control logic.

This paper describes the new linear array processor architecture based on MAM (LAPMAM), its Multi-mode Access Memory, the structure of the SIMD/restricted MIMD processing element, its interconnection network and the performance results of the hardware simulation of a prototype. A comparison between LAPMAM and others image processing architectures is finally given.

2. LAPMAM architecture

The LAPMAM is a linear array of RISC SIMD/restricted MIMD processors with a Multi-

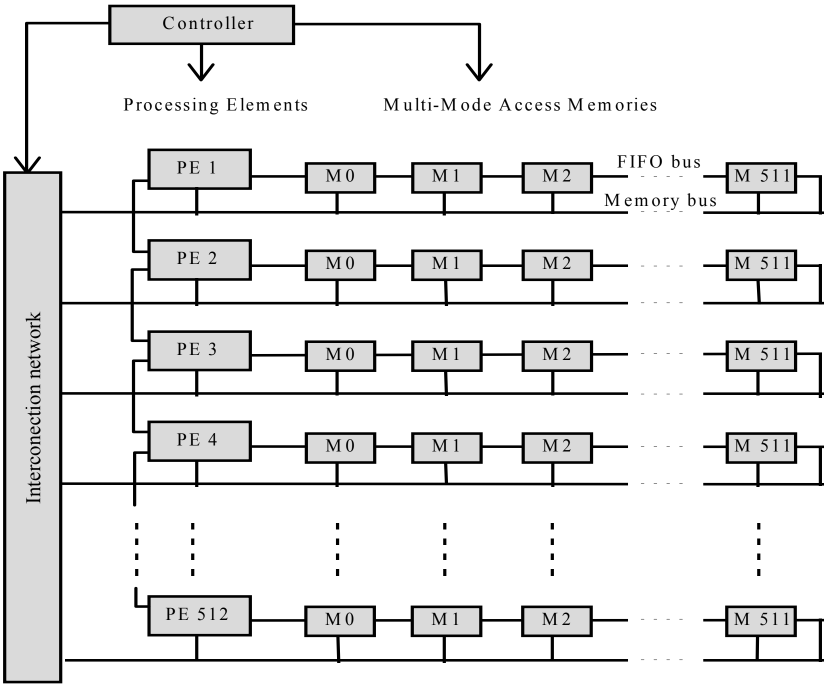

Figure 1: The LAPMAM architecture

mode Acces Memory. The LAPMAM has four memory planes that are dependent tanks to the bi-directional heteroassositive CAM property. A controller gives the instructions word to each PE. The PE-PE and PE-MAM communications are carry out by a tree interconnection network and by a local bus (Memory bus) between the PEs and its corresponding row of memory modules.

The LAPMAM architecture for a 512 x 512 image (n=512) is shown on Figure 1. It features n processors organized in a linear array. Each of them is connected to a row of n memory modules. A special interconnection network allows every processor to reach any of the other processors and their associated memory rows. This network presents a tree structure and ensures global communication in O(log n) units of propagation time.

LAPMAM has four identical memory planes of log 512 bits denoted $M_{A1}[i,j]$, $M_{A2}[i,j]$, $M_{B1}[i,j]$ and $M_{B2}[i,j]$ (0 ? i,j ? 511). Each plane consists of 512 rows, each containing 512 memory modules. The four planes can be turned into two planes $M_A[i,j]$ and $M_B[i,j]$ of 2 log 512 bits. On Figure 1 the planes M_{A1}, M_{A2}, M_{B1}, M_{B2} are represented by the memory modules (M).

3. Multi-mode Access Memory

This Multi-mode Access Memory (MAM) module is basically a modified CAM. The CAM is a memory with addressing based on its *content*. This is an excellent solution in some applications where the RAM, with addressing based on its location, shows limited performance. The main advantage of the CAM is its capability to write/read a data to/from multiple locations in only one clock cycle or *O*(1) time. The concept of CAM was introduced in 1956 [6]. Despite its relatively high cost, CAM has found since then enormous importance in various applications like data base management [7, 8] and image processing [9, 10].

The CAM enhances the parallelism of this architecture because this memory works inherently in parallel. However, its utilization reduces the processing flexibility since the CAM can not be addressed by its position and the CAM reading is difficult. We have designed a CAM based memory with the possibility RAM and FIFO to solve the limitation of the CAM pure, it was called Multi-Mode Access Mem-

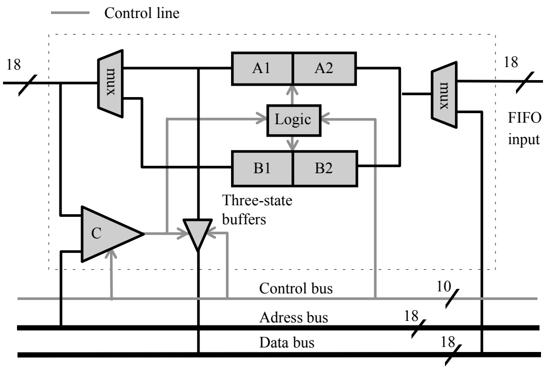

Figure 2: Memory module bloc diagram

ory. The bloc diagram of the MAM is shown on Figure 2.

The MAM modules constitute either four log n bits wide or two 2 log n bits wide memory planes. The four planes enable the architecture to work with algorithms that need to store intermediate results. The image loading procedure is also made simpler thanks to this new possibility: an image frame may be stored in one memory plane while the previous is still under processing. The size of the memory words depends on the algorithms being run (18 bits for labeling and 9 bits for median filtering for example). The CAM and RAM operation can be carried out in a whole plane, in a row (PE-MAMs) or in several rows of a plane. The FIFO operation is only carried out in the couples PE-MAMs.

The functional diagram of a MAM module (Figure 4) consists of four log n bits registers, a little control logic made of several elementary gates, one comparator (C) and two multiplexers. The comparator makes a comparison between the *content* of any of the four registers and the data issued from the address bus. The writing plane is defined by the comparator result and the write control signals.

4. The processing element (PE)

The LAPMAM processing element was designed to exploit the Multi-mode Access Memory possibilities. The philosophy design of the PE was develop a simple RISC processor that allows exploit all the MAM possibilities trying of obtain a small element to integrate in a chip the greatest number of couples PE - row of MAM. Taking the above consideration, we propose a SIMD processor for our architecture with the possibility of take some decisions. Each PE can be activated or deactivated independently. The PE is able to compute a basic logical or arithmetic operation in $O(1)$ time. It can communicate in $O(1)$ units of propagation time with its adjacent PEs or with its associated memory modules. Furthermore, it can communicate in $O(\log n)$ units of propagation time either with its non-adjacent PEs or its non-adjacent memory modules through the interconnection network.

4.1 Processor architecture

Each processor features sixteen 9-bits registers, an 18 bits arithmetical and logical unit (ALU), a 28 bits x 4 internal memory, flag registers and a small decoder. It distributes the signal controls to the different elements of PE. The processor has reduced instruction-set computer (RISC) load/store architec-

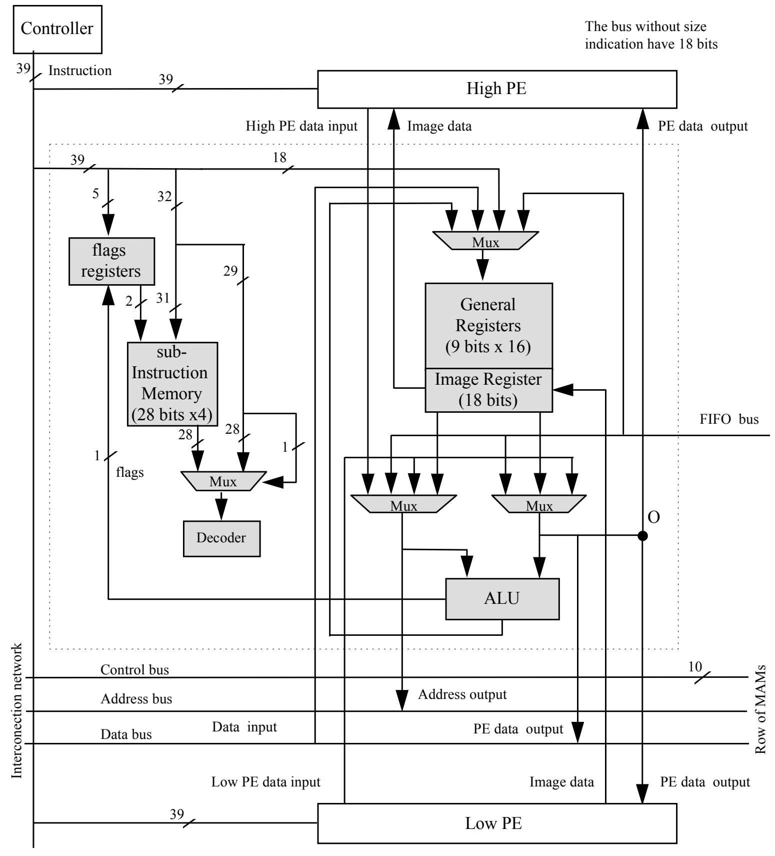

Figure 3: Processor architecture

ture. All operations, including memory load/store operations, are carried out in one clock cycle. The processor is shown on Figure 3.

When the processor is connected directly to its memory row, the access to the data contained in this line is accomplished by means of the FIFO, RAM and CAM modes. In the FIFO mode the data of the PE are transferred to the last MAM module of the row.

Using the tree interconnection network, a processor can be connected to several memory rows or even to all memory rows. This depends on the interconnection network programming. To enable the communication between PEs, each PE has a data output toward its adjacent PEs (upper PE, lower PE). The data output of each PE is taken at the point O on Figure 3. The data output of each processor can be activated or deactivated independently: the PEs have a particular address in the register 0, which is compared to the addressee sent by the general controller, thus allowing to select which PEs are active. This is quite useful when a PE needs to take control of several memory rows. To accomplish this, the remaining PEs are deactivated.

The PE can work with 9 or 18 bit words, their inner registers being addressable either as a 9 or 18 bits register. This possibility used when the LAPMAM architecture process algorithms that need to execute data operation with more than 9 bits like our algorithms of labeling and area computing. The PE has also a 18 bits shift register that allows the data transfer to the video interface or vice versa.

4.2 Restricted MIMD mode

A SIMD PE is characterized by its reduced size. But, because it does not have a unit control, these types of PEs can not take internal decisions. Then, to execute different operations on different data, an architecture SIMD has to connect and disconnect the processor as many times as the number of different operations. On the other hand, the MIMD processor, that has a unit control, can take internal decision, but they are very much complex. It limits the number of PEs in an integrated circuit. We are designed a SIMD processor that can take some internal decisions, this possibility increment the flexibility of the LAPMAM architecture avoiding the connection and disconnection of PEs pour perform different instructions, reducing the computing time. This particular characteristic will be called restricted MIMD mode because the processor can only take a few decisions. For working in this way each processor stores in its internal memory a limited number of sub-instructions (4). The selection of one of these sub-instructions depends on the flags previous results that are stored in the flag registers, they are the input address of the sub-instruction memory. The controller has a bit that allows the output of the sub-instruction memory to the instruction decoder. In the SIMD mode the instructions come directly of the instructions word, but in the restricted MIMD mode the instructions arrive from the sub-instructions memory. In this last mode each PE executes an instruction that depend of its internal flags registers. Then, the LAPMAM can execute different instructions on different data.

5. The interconnection network

We have design a tree interconnection network type that complements the couple PE-MAMs. This network allow performing the communications PE-PE or PE-MAMs in O(log n), but in some case it can be executed in O(1), possibility that we are exploited in our algorithms. Moreover, this network has the characteristics of modularity and extensibility that allow to the network to be constructed from a small set of basic modules and to be extended to a larger size [11]. These possibilities are very interesting for a VLSI implementation.

The interconnection network is reconfigurable by n+(3n/4)-1 switch modules denoted S_g. Each switch S contains (4 log 512 + 2) three states buffers. The PE-PE and PE-MAMs connections can be carried out in regions. Some PEs can be connected to a region of 2, 4, 8 etc. elements (PEs or rows of MAMs). This connection allows certain PEs to do a regional or global communication in O(1) with a propagation delay of O(log n). But, in general, a global communication time of O(log n) is obtained with this type of network [12]. Furthermore, its special form allows all processors P_j to simultaneously write/read directly to/from their current row_j in only O(1) time. A tree interconnection network for architecture with eight PE is presented on Figure 4.

The network has a system protection to avoid possible conflicts between PE outputs because a program mistake may connect two PE to the same memory bus. To solve this problem, the PEs with output activated disable the network data input to their memory bus automatically. Thus, each PE controls the closest switch module of the network as it is shown on Figure 4.

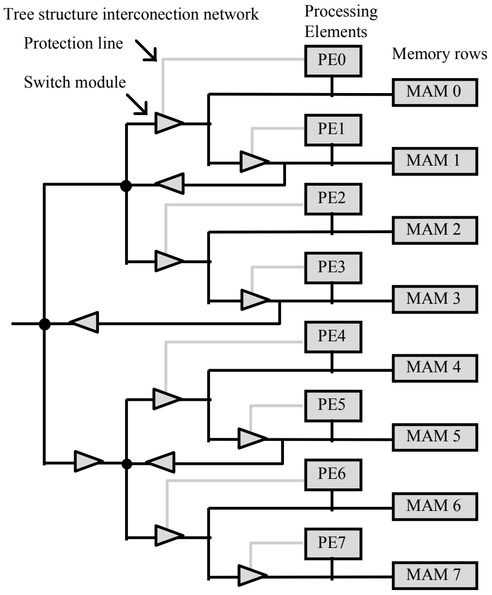

Figure 4: Tree structure interconnection network for LAPMAM architecture with 8 PE and 8x8 MAMs.

6. Performance

In order to validate the LAPMAM architecture, we have performed a hardware simulation of a prototype of 8 PEs with 8 MAM rows. The algorithms of labeling, area and perimeter computation, histogramming and median filter were implemented on this prototype. These algorithms use the architecture properties to obtain a quasi-optimal complexity [3]. All these algorithms have a O(n log n) complexity, except the median filter that has the optimal O(n) complexity for this type of architectures. The hardware simulation shows that the architecture can work with a 50 MHz frequency. All the algorithms were simulated at this frequency and the results, which were extended to 512 PEs architecture, are shown and compared with others architectures in Table 1. LAPMAM computes these low and intermediate-level image processing algorithms much faster than the video rate.

The best performance results of the DARPA II image understanding benchmark [13] for the algorithms evaluated are compared in the first part of the Table 1. The architectures included are the Connection Machine (CM) with 64 K of PE, the Associative String Processor (ASP) that has 262,144 processors and the Image Understanding Architecture (IUA) that consists of three difference processors: low level SIMD PEs (processor-per-pixel), 4096 intermediate level SIMD/MIMD 16 bits processors, and one high level multiprocessor. For the tasks compared, our architecture is among the best ones while being the least complex. On this benchmark, only IUA has better results for labeling. However, it features for many more processors than our architecture. Otherwise, LAPMAM has the best computation times. This does not necessary mean that our architecture is much better than the others, since these architectures are very different and the technology evolution is not considered. Nevertheless, it gives a good idea of the LAPMAM's potential in low and intermediate level tasks.

In the second part of the Table 1, the LAPMAM estimated performances are compared with architectures that are more similar to LAPMAM: VIP [14], SliM-II [14] and IMAP VISION [15]. In this comparison, our architecture has the best results for these algorithms. Its enhanced parallelism allows the reduction of the algorithms complexities. The use of CAM and the tree structure of switches in interconnection network make the LAPMAM extremely efficient in terms of connected component analysis and median filtering tasks [16]. However, because of the MAM modules, the architecture is more complex than the ones that use RAM. LAPMAM thus necessitates a full custom approach for its hardware implementation.

Table 1: The LAPMAM estimated time results compared with others architectures (time in ms)

Algorithm	DARPA II Benchmark results for a 512x512 image			LAPMAM similar architectures			LAPMAM
	CM 64 K	ASP	IUA	VIP 1024 PEs, 50 MHz	SliM-II 512 PEs, 40 MHz	IMAP-VISION 512 PEs, 40 MHz, 256x240 image	50 MHz, 512 PEs, 512x512 image
Labeling	100	22.8	0.0596	-	-	19.5*	0.614
Median filter	15	0.72	0.5625	3.672	2.525	1.07	0.184
Histogram	-	-	-	-	3.313	1.33	0.420

* Worst case example

7. Conclusion

The LAPMAM architecture and its estimated performance have been presented. This is a new design of linear array processors using content addressable memory. The use of CAM modules increases the row architecture parallelism. The typical function of the CAM has been modified to obtain a more flexible memory with a multi-mode access featuring RAM, FIFO and CAM. The algorithms for labeling, area, perimeter, histogramming and 3x3 median filtering have been implemented in a LAPMAM prototype to test their performance. The Hardware simulation results demonstrate the efficacy of LAPMAM for low and intermediate level vision tasks. The use of a tree structure of switches interconnection network has also proved to be an excellent solution to decrease the data propagation time.

Globally, the architecture presents very good performance for image processing. It will be confirmed with the development of other algorithms and the LAPMAM hardware implementation.

References

[1] V. K. P. Kumar, *Parallel Architectures and Algorithms for Image Understanding*: Academic Press INC., 1991.

[2] A. Downton, and D. Crookes, "Parallel Architectures for Image Processing," *Electronics & Communication Engineering Journal*, vol. 10, pp. 139-151, 1998.

[3] Byeong-Moon Jeon, Kyu-Yeol Chae and Chang-Sung Jeong, "Reconfigurable Mesh Algorithm for Enhanced Median Filter", 4th International Meeting on Vector and Parallel Processing VECPAR'2000, Porto Portugal, 2000.

[4] H. M. Alnuweiri, "Parallel architectures and algorithms for image component labeling," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 14, pp. 1014-1034, 1992.

[5] E. Mozef, S. Weber, J. Jaber, and E. Tisserand, "Design of Linear Array Processors with Content Addressable Memory for Intermediate Level Vision," presented at ISCA 9th International Conference on Parallel and Distributed Computing Systems, Dijon, France, 1996.

[6] A. Slade, H. O. McMahon, "A Cryotron Catalog Memory System," presented at EJCC, 1956.

[7] S. S. Yau, and H.S. Fung, "Associative Processor Architecture - A Survey," *ACM Comp. Surveys*, vol. 9, pp. 3-28, 1977.

[8] L. Chisvin, and Duckworth R. J., "Content-Addressable and Associative Memory: Alternatives to the Ubiquitous RAM," *Computer*, vol. 1, pp. 51-64, 1989.

[9] Y. C. Shin, et al., "A Special-Purpose Content Addressable Memory Chip for Real-Time Image Processing," *IEEE Journal of Solid-State Circuit*, vol. 27, pp. 737-744, 1992.

[10] Y. Fujino, T. Ogura, and T. Suchiya, "Facial Image Tracking System Architecture Utilizing Real-Time Labeling," *SPIE - Int. Soc. Opt. Eng (USA)*, vol. 2094, pp. 2-11, 1993.

[11] H. J. Siegel, Robert J. McMillen, and Philip T. Mueller Jr., "A survey of interconection methods for reconfigurable parallel processing systems," presented at AFIPS Conference Proceeding 1979, National Computer Conference, 1979.

[12] G. S. Almasi, and Allan Gottlieb, *Highly Parallel Computing*. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc., 1989.

[13] C. Weems, Riseman Edward, and Hanson Allen, "*The DARPA Image Understanding Benchmark for Parallel Computers*," *Journal of Parallel and Distributed Computing*, vol. Vol. II, fasc. 1, p.p. 1-24, 1991.

[14] H. C. Chang, Soohwan Ong, and Myung H. Sunwoo, "A linear Array Parallel Image Processor: SliM-II," presented at Proceeding IEEE, International Conference on Aplications-Specific Systems, Architectures and Processors, Zurich, Switzerland, 1997.

[15] Y. Fujita, N. Yashamita, and S. Okazaki, "IMAP-VISION: An SIMD Processor with High-Speed On-Chip Memory and Large Capacity External Memory," presented at MVA'96 IAPR Workshop on Machine Vision Application, 1996.

[16] Domingo Torres, Hervé Mathias, Hassan Rabah, and Serge Weber, "Efficient low and intermediate level vision algorithms for the LAPMAM image processing parallel architecture", 4th International Meeting on Vector and Parallel Processing VECPAR'2000, Porto Portugal, 2000.