A PROCESSING ELEMENT ARCHITECTURE FOR HIGH-DENSITY FOCAL PLANE ANALOG PROGRAMMABLE ARRAY PROCESSORS

G. Liñán-Cembrano, S. Espejo, R. Domínguez-Castro and A. Rodríguez-Vázquez

Instituto de Microelectrónica de Sevilla – CNM-CSIC
Edificio CICA-CNM, C/Tarfia s/n, 41012- Sevilla, SPAIN
Phone: +34 95 4239923, Fax: +34 95 4231832, E-mail: linan@imse.cnm.es

ABSTRACT[1]

The architecture of the elementary Processing Element -PE- used in a recently designed 128x128 Focal Plane Analog Programmable Array Processor is presented. The PE architecture contains the required building blocks to implement bifurcated data flow vision algorithms based on the execution of 3×3 convolution masks. The vision chip has been implemented in a standard 0.35μm CMOS technology. The main PE related figures are: 180 cells/mm^2, 18 MOPS/cell, and 180 μW/cell.

1. INTRODUCTION

Among all the possible developments of the sight sense, evolution selected those individuals -and their vision systems-, in which the early image processing -involving calculations on a huge data space- were performed at the sensory plane. In these systems, a special part of the brain, *the retina*, was moved to the sensing device, *the eye*, reducing the amount of information to be transmitted -and processed- to the brain [1].

Inspired by this efficiency, massively parallel focal plane image processing architectures have been proposed since the late 80s [2]. In these systems, each light-sensing device is surrounded by special circuitry for early image processing, constituting a so-called *Processing Element* PE. The moderate accuracy constraints, as well as the low power consumption and high density of integration -PEs per chip- requirements, has pushed the design of these PEs to the analog side.

The optimum design of the PE is the crucial issue in any type of massively parallel processing architecture. The reason is quite simple; these systems consist of the spatial replication of a fundamental building block, the PE. Hence, since the processing performances of the system depend, directly, on the number of processing nodes -PEs-, the maximization of the number of PEs, is the major design objective[2].

Unfortunately, not only the processing capabilities are directly proportional to the number of PEs. In practice, most of the total power consumption, as well as the final die area, are also a direct consequence of those at the PE level. This renders the design of the PE as the fundamental problem to be faced. It is an iterative optimization process in which every single micrometer, and microwatt, counts. Therefore, additional design efforts to reduce the power consumption and area occupation, while keeping the processing performances, commonly worth it.

In what that follows, we will describe the architecture, and main design techniques, used to implement the PE in the so called ACE16K chip [3]. This is a 128×128 Focal Plane Analog Programmable Array Processor -FPAPAP- designed in a 5M-1P 0.35μm CMOS process. Global performance figures[3] are 0.33TeraOPS, 0.18 TeraOP/J, 3.8GOPS/mm^2, while advanced competitors [4], [5] provide 2.8 GOPS-9GOP/J-65MOP/mm^2, and 0.5GOPS-13GOP/J-0.11GOPS/mm^2, thus positioning the ACE16K in a top position in this field.

2. ARCHITECTURAL CONSIDERATIONS

The first step, in this optimization process, runs at the architectural level rather than at the circuit or transistor ones. The identification of the required functional blocks for low-level image processing applications, the design of

1. This work has been partially funded by ONR-NICOP N68171-98-C-9004, EU-IST DICTAM IST-1999-19007 and Spanish-CICYT TIC1999-0826

2. A serial computation -pixel-wise- architecture, must be N-times faster than each PE -N=number of PEs- in order to provide the same Frame rate.

3. Figures refer to 8-b accuracy operations

0-7803-7448-7/02/$17.00 ©2002 IEEE

a proper -simple, fast, and accurate- communication scheme among blocks, and the supply of flexible enough -high programmability and versatility- circuit and system topologies, are the guidelines for the selection of the PE architecture.

Probably, the first consideration refers to the accuracy requirements. Low-Level image processing tasks can be efficiently carried out by using 6-8b precision calculations [5], [6]. Hence, any attempt to further increase the accuracy of the different blocks in the PE -either by augmenting the device's area and hence reducing the mismatching level, or by adding special callibration schemes- does not commonly worth it.

In addition to that, PEs architectures must be designed as the modular aggregation of functional blocks, instead of as a sea of analog transistors. Hence, those blocks must be completely compatible from the I/O point of view, in order to allow the flow of information between them. This fully-communicated inter-function scheme, provides high algorithmic capabilities, due to the large number of possible data transferences and operations. On the other hand, making the processing blocks easily programmable, increases the versatility of the system to easily -by reprogramming- fit into very different applications and environments.

Finally, selecting which functional blocks must be included at the PE level, is an open question that depends on concerns of different nature. Most of these considerations, refer to the environment in which the chip is to be used and on the intended application. However, due to both, the size of these chips, and the high fabrication costs in modern technologies -about 80k Euro/cm^2 for a digital 5M-1P 0.25μm process-, the design of a special purpose vision chip is confined to those cases in which the market niche -and sales- is ensured. Otherwise, the PE architecture must be flexible enough to guarantee the execution of different vision algorithms -also in different illumination conditions. Thus, if we consider that most early vision process consist of the application of convolutions masks, and the combination -either by boolean operations, in the case of B/W images, or by a local analog arithmetic operator- of their results in a bifurcated flow algorithm, the following operators should be included at the PE level.

- Multipliers and Adders: For the convolution operation.
- Analog Registers: To allow for the storage of various image processing results at the local level.
- Arithmetic Operator and/or Binary Operator; To combine previously obtained results.
- Local Masks; To allow for the conditional execution of certain operations at the PE level depending on some locally defined value.
- Optical Input module; To permit the light sensing capability, and, hence, to avoid the bottle-neck existing in data transmission from the sensory to the processing plane.

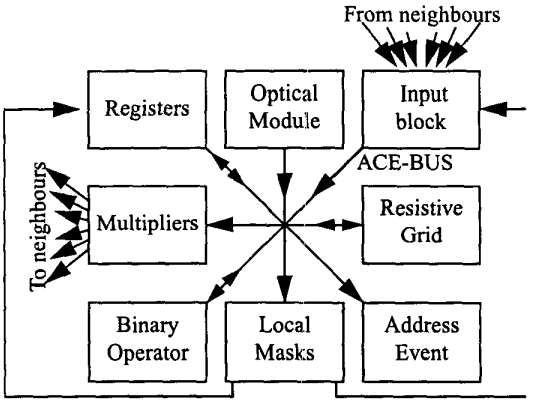

Figure 1. Block Diagram of the PE in ACE16K.

3. THE PE IN ACE16K

3.1. Block Diagram

Fig. 1 shows the block diagram of the PE in ACE16K. Arrows indicate how information flows. It contains 8 fundamental building blocks that communicate to each other by means of the so-called *ACE-BUS*. Data transferences are always carried out in the same way; some block -the data source- drives the ACE-BUS while another one -the data destination-, at the same time acquires this information from the ACE-BUS. Since the processing is done in the analog domain, this bus is, in practice, a single wire, thus reducing the amount of area and complexity required for the interconnection as compared to a possible digital design of the PE.

3.2. Multipliers

A bank of programmable analog multipliers is used to implement the neighborhood operations required in low-level image processing [7]. It connects the PE with its 8 nearest neighbors and with the PE itself. Multipliers are designed using a one transistor technique [8], which, in addition to the intended product, also generates a signal independent current -offset- that must be cancelled afterwards. Both, pixel and scaling coefficient variables, are codified in voltage form, while multiplier' output is provided as a current -thus facilitating the computation of the required addition for the convolution. Multipliers, in Fig. 2, are driven by three different pixel values, P_A, P_B and P_C in such a way that the current which flows to the processing core is expressed as,

$$I_{in} = \mathbf{A} \bullet \mathbf{P_A} + b \cdot P_B + c \cdot P_C + z \qquad (1)$$

where the $\mathbf{A}$ and $\mathbf{P_A}$ matrices are defined as[4],

4. Indexes *r, l, c, t, b,* are used to specify the position of the neighbor right, left, centre, top, bottom. Hence, for instance, the contribution to the top-left neighbor is $a_{tl} \cdot P_A$.

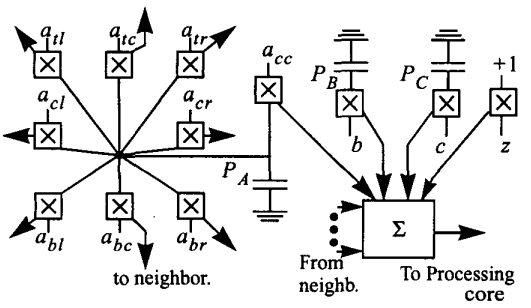

Figure 2. Distribution of Multipliers

$$\mathbf{A} = \begin{bmatrix} a_{br} & a_{bc} & a_{bl} \\ a_{cr} & a_{cc} & a_{cl} \\ a_{tr} & a_{tc} & a_{tl} \end{bmatrix} \qquad \mathbf{P_A} = \begin{bmatrix} P_{A_{tl}} & P_{A_{tc}} & P_{A_{tr}} \\ P_{A_{cl}} & P_{A_{cc}} & P_{A_{cr}} \\ P_{A_{bl}} & P_{A_{bc}} & P_{A_{br}} \end{bmatrix} \quad (2)$$

3.3. Current Processing Block

The currents, generated by the one transistor multipliers, are collected by the input block of the PE -in Fig. 3 -schematic is displayed in Figure 4. Due to the low output impedance of the one-transistor multipliers, a virtual ground -with the appropriate voltage value- must be provided -by a class II current conveyor- for that purpose. The non-desired offset contribution generated by the multiplier topology, is substracted from the total input current, by using a high accuracy current memory block based on a s^3I memorization scheme. Afterwards, I_{in}, can be either directly steered to the ACE-BUS or sent to the input of a current comparator, whose output can be also connected to the ACE-BUS. When the PE is operated in grey-scale mode[5], the input current is allowed to flow into any -user selectable- of the capacitors associated to the pixel. Depending on this selection, different processing paradigms are achieved. Thus, for instance, to run a Sobel operator, we would define the operator in the **A** matrix, the image to be processed would be loaded to the P_A pixel, and we would use $c = z = 0$, and $b = -1$, to obtain,

$$C_B \frac{dP_B}{dt} = -P_B + \mathbf{A} \bullet \mathbf{P_A} \quad (3)$$

whose steady state solution is $P_B = \mathbf{A} \bullet \mathbf{P_A}$.

If the capacitor which is allowed to be updated is C_A, then PEs are dynamically coupled, and we get CNN-like

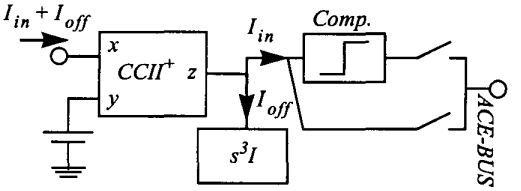

Figure 3. Input block of the PE

behavior [9]. In addition, by allowing the current to flow into C_A and by defining $a_{cc} = -1$ and $a_{ij} = 0$, the steady state solution is,

$$P_A = b \cdot P_B + c \cdot P_C + z \quad (4)$$

thus providing grey-scale arithmetic operations.

3.4. Analog Registers

The analog register stores 8 grey-scale pixel values with an equivalent resolution of 8-b. Memorization relies on a bottom-plate sampling technique [10], which avoids the introduction of signal dependent feedthrough error. Moreover, this technique is not sensitive to offset voltage of the required *opamp*. Hence, it is very suitable for spatial uniformity issues.

3.5. Optical Module.

The optical input module, consists of a multimode sensor in which both the physical device used as sensor and the transduction mechanism are programmable. A P-diff/N-well diode, a N-well/P-subs diode, or a P-diff/N-well/P-subs phototransistor, are available. Selection is done by using global programming instructions. Furthermore, the phototransduction scheme is also programmable. Both linear integration of the photogenerated current, or logarithmic compression

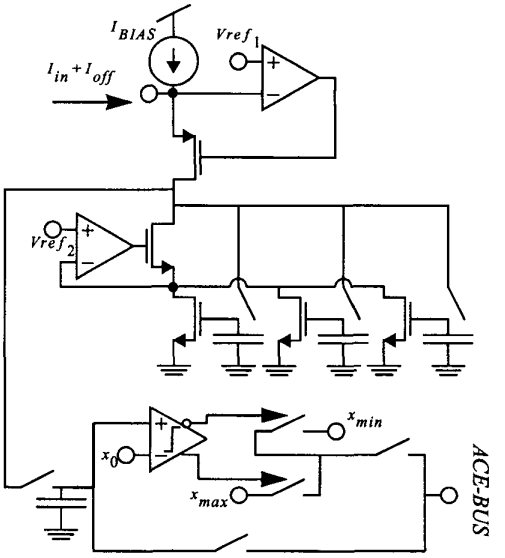

Figure 4. Schematic of the Image Processing Kernel

5. When the comparator is used, the same processing tasks are available. However, in this case, only B/W are produced.

sensing, are also available by proper definition of the digital instructions controlling the chip operation. This allows for the chip to be used in very different illumination conditions.

3.6. Binary Operator

A fully programmable 2-input 1-output binary operator has been also included. It allows for the execution of any 2-variables boolean function. Operands are acquired from the ACE-BUS, while the logic operation to be performed is defined by four global digital instruction signals -in truth-table form.

3.7. Local masks & Address Event Blocks

Two masks, whose use is optional -they must be enabled by the user-, have been also incorporated to the PE. The first one, called *freeze* mask, is a transient enabling-disabling mask. When active, it interrupts the flow of current from the input block to the ACE-BUS, hence, those PE's in which the content of the mask -which has been previously acquired from the ACE-BUS- is +1 remain unchanged. The second one, the *writing* mask, operates in a similar way. It disables the possibility of updating any analog register. Therefore, it allows for selective updating of content of the memory.

The address event circuitry is an independent module which allows for fast downloading of sparse images. In this case, the chip provides the addresses of those pixels in which activity is detected. Active cells are those cells having a low-logic level at the ACE-BUS when the address event downloading is started.

3.8. Resistive Network

A resistive grid, connecting the PE to its right and top neighbor has been included. It allows for the execution of conventional low-pass diffusion process.

4. Relevant Data

Fig. 5 displays the layout and floor planing of the PE in the ACE16K chip, containing 198 transistors. PE size is $75.5 \times 73.7\ \mu m^2$ -180 cells/mm^2. Maximum power consumption of the cell is $180\mu W$. Time constant for linear convolutions is 160ns, yielding peek performances of about 18 MOPS per cell -9 products and 8 additions performed in 960 ns -six time constant units.

Finally, Fig. 6 shows, as an example, the execution of the Sobel operator on the famous Lena's picture. Image loading and downloading requires $130\mu s$ each, while total processing time is only $3\mu s$ -including internal self-callibration processes-.

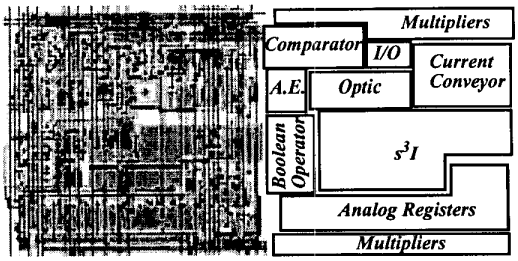

Figure 5. Layout and Floorplaning of the PE

5. REFERENCES

[1] B. Roska and F. Werblin, "Vertical Interactions Across Ten Parallel, Stacked Representations in the Mammalian Retina". *Nature*, No. 410, pp. 583-587, March 2001.

[2] A. Moini, *Vision Chips*. Kluver Academic Publishers, ISBN 0-7923-8664-7, 2000.

[3] G. Liñán, R. Domínguez-Castro, S. Espejo and A. Rodríguez-Vázquez, "ACE16K: an Advanced Focal-Plane Analog Programmable Array Processor". *European Solid State Circuit Conference,* Villach, Austria, Sept. 2001.

[4] J. C.Gealow and C. G. Sodini, "A Pixel-Parallel Image Processor Using Logic Pitch-Matched to Dynamic Memory". *IEEE J. of Solid State Circuits,* pp. 831- 839, Vol. 34, No. 6, June 1999.

[5] P. Dudek, *A Programmable Focal-Plane Analogue Processor Array.* Ph. D. Dissertation, University of Manchester Institute of Science and Technology, May 2000.

[6] D. A. Martin, H. S Lee, and I. Masaki, "A Mixed Signal Array Processor with Early Vision Applications". *IEEE J. of Solid State Circuits,* Vol. 33, No. 3, pp. 497-502, March 1998.

[7] B. Jahne, H. Haubecker and P. Geibler (Eds.), *Handbook of Computer Vision and Applications Volume II: Signal Processing and Pattern Recognition.* Academic Press, London, ISBN 0-12-379771, 1999.

[8] G. Liñán, *Design of Programmable Vision Chips with Low Power Consumption Levels.* Ph. D. Thesis, University of Seville, to be published in 2002.

[9] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory". *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, Vol. 35, No. 10, pp. 1273-1290, October 1988.

[10] R. Carmona, A. Rodríguez-Vázquez, S. Espejo, R. Domínguez-Castro, T. Roska, T. Kozek and L. O. Chua, "A 0.5um CMOS Analog Random Access Memory Chip for TeraOPS Speed Multimedia Video Processing". *IEEE Transactions on Multimedia*, Vol. 1, No. 2, pp.121-135. June 1999.

Figure 6. Example of Image Processing with ACE16K