A Parametrized Real Time Image Convolver

M. Aberbour, A. Houelle, H. Mehrez , N. Vaucher
G. Besencenet*, E. Dupont-Nivet*, F. Durbin*, T. Garrie*, A. Tissot*,

Laboratoire MASI/CAO-VLSI
Université Pierre et Marie Curie (PARIS VI)
4, Place Jussieu
75252 Paris Cedex 05 FRANCE
e-mail: mourad.ABERBOUR@masi.ibp.fr

*CEA-DAM, Centre d'études de Bruyères-Le-Chatel
Service d'électronique, BP N° 12
91680 Bruyères-Le-Chatel
FRANCE

Abstract

This paper presents a design of a real time image convolver chip set, for use specifically in pattern recognition applications. The main features of this design are : use of systolic arrays, data vectorization, cascadability and reconfigurability.
Several major goals guided this work. It was desired to produce an image convolver that can use big size masks, like the Gabor masks (32x32), and that could be configurable and cascadable to satisfy many real-time performance requirements.
A hardware/software technique based on the module generators concept has been used in this design. The chips have been successfully fabricated with a 0.7 µm CMOS technology. The circuit performs a convolution of an 256x256 image with a 32x32 Gabor mask in 265 ms.

Introduction

One of the basic operations used in image processing is the convolution. The computation throughput and speed demand seem to be huge in image processing applications. A serious bottleneck for speed is the number of I/O operations. Systolic arrays enlighten the burden of I/O exchanges, since an advantage of these arrays is that each input data item can be used a number of times once it is accessed. Other advantages of systolic arrays are configurability, modularity and cascadability. Moreover systolic arrays make use of simple and regular Processing Elements (PEs) that are easily interconnectable. All these schemes make the VLSI implementation more practical and desirable.
The application targeted by our convolvers is target recognition. The task of recognizing an object in a complex scene becomes harder when considering multiple invariance constraints such as rotation, scale and translation. Gabor functions or wavelets seem to be very suited for such an environment. In this case the filters implemented, said to be Gabor filters, are of big sizes (32x32). The processing should be done in real time, thus the need for a very high performance convolver. The recognition system is said to be "of GABOR", it is based on a neural network supplied by different signatures of the image to be processed. Where the signature is a primary representation of the input image enhancing certain features obtained by the Gabor filtering [16,17,18].
The architecture of the convolver implements a number of performant and flexible concepts : use of systolic arrays, data vectorization, cascadability and reconfigurability. The elementary processors, or systoles, carry out elementary operations. One of these operations is

multiplication. The multipliers occupy a large surface in each PE. This fact leads to an unrealistic implementation of the systolic convolver. A surface/latency compromise has been investigated. So the concept of associating an elementary systole to each filter coefficient has been abandoned in preference of a vectorial design of the systolic system. Thus, each convolution line corresponds to a vectorized systole.

This paper investigates all the architectural schemes summarized here as well as the VLSI implementation based on the module generators concept [1,2,3]. This concept is of major interest. The generator is developed over an environment which allows an independence of the generators from the cells library used and from the CAD framework (Cadence, Compass, Alliance [a set of CAD tools for teaching VLSI developed at the university of Paris],...) at hand [4].

Architectural issues

In order to carry out a convolution operation satisfying real-time constraints, there exist many different types of architectures. One of the simplest is to hard-wire completely the convolution equation in terms of adders and multipliers to implement the sums of products operations. Even though this architecture is very fast, it is only possible for convolutions with small size filters, because this architecture being fully parallel requires a large number of arithmetic operators. It is possible to make use of more complex operators such as the ones implementing the Discrete Fourrier Transform (DFT). This architecture is more expensive than a systolic architecture, it occupies a surface of the order of O(N log(N)), where N is the filter size. A systolic architecture is a net of elementary operators locally interconnected. The cells operate in parallel so the synchronization is total. There exist also many types of systolic architectures. After a study of these different architectures (classical, symmetric systolic, transversal and diffused systolic), [5,9,11,12], we chose an architecture of the type of diffused systolic illustrated in figure 1.

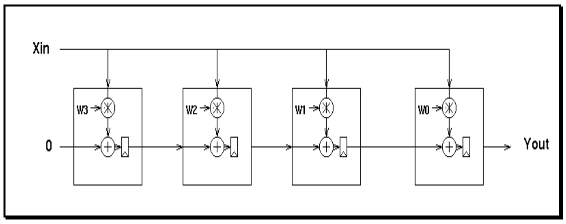

Figure 1 :Diffused systolic architecture

The advantage of a systolic architecture resides in the homogeneous use of the Silicon surface in the VLSI design by a planar routing of the data. Moreover this architecture is readily cascadable, allowing the connection of many circuits together in order to widen the size of masks and to achieve higher computation speeds. Finally, this architecture allows the reconfiguration of the systolic net to compute convolutions with masks of different size and of different coefficients.

Two dimensional convolution

The two dimensional convolution architecture presents two main features, one consists of the interconnected elementary processors and the other is the delay lines (shift registers) to store the intermediate results. These registers are implemented using a commercial FIFO circuit, figure 2.

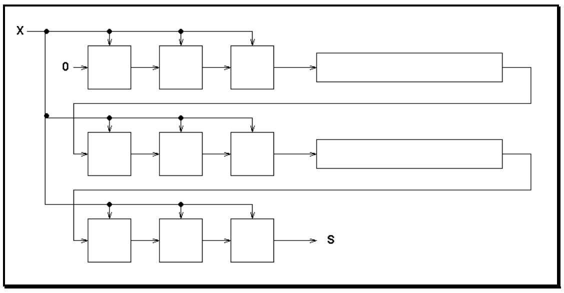

Figure 2 :Two-dimensional systolic architecture

Vectorization

The maximum size of the masks that we intend to use in the convolution computations is of the order of 32x32 coefficients. The design of such an operator requires a large number of arithmetic operators (multipliers and adders). To reduce the number of these operators we vectorized the initial systolic architecture, figure 3. According to a delay-surface compromise we chose to implement one systole for each column.

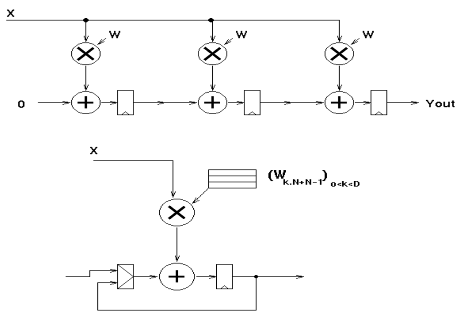

Figure 3 : Vectorization

Cascadability

In order to compute convolutions with large size masks, we make use of the cascadability of the circuit. This cascadability can be implemented in either software or hardware.

The software cascadability is achieved by iterating several times on the image with the different parts building up the complete mask.

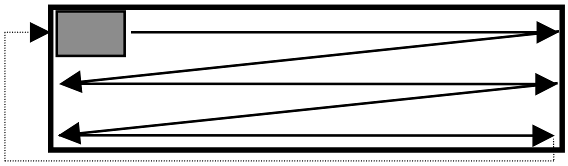

Figure 8 : Software cascadability

The hardware cascadability, figure 5, is achieved by interconnecting a set of circuits together.

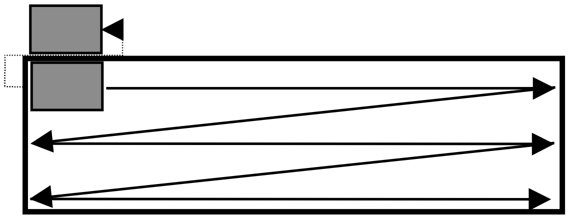

Figure 5 : Hardware cascadability

The systoles and the convolver core

The systole contains a file of 32 registers storing the different values on 13 bits of the filters coefficients. The systole consists also of a signed or non-signed multiplier and an adder-accumulator computing the sum of products, figure 6. The systole possesses horizontal communication ports (Yin, Yout) for the flow of the intermediate results, and a vertical port (DW) to bring the filter coefficients and the pixel (signal) values. The systole computes signed or non-signed convolutions according to the type of data (images, speech, ...).

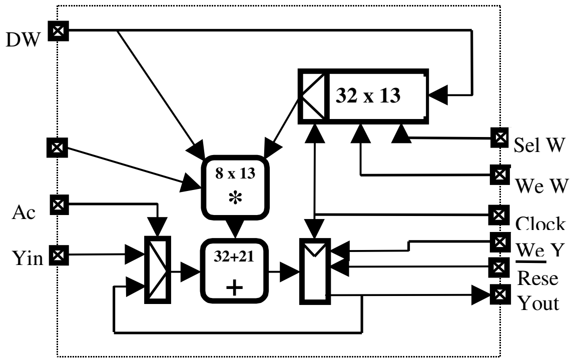

Figure 6: Architecture of a systole

The designed chip contains 8 systoles. It computes convolutions with masks of sizes up to 32x8 coefficients. The cascadability of the circuit makes it possible to realize convolutions with larger size masks.

The architecture

The heart of the chip is the convolution core comprising 8 systoles as shown in figure 7. The multiplexers at each side of the core get in and deliver out the intermediate data. The programmable line and column counters set up the size of the convolution mask. the chip contains a status register indicating the functioning mode of the circuit. Finally there is all the combinatory logic for the control of the circuit and the external FIFO simulating the delay lines.

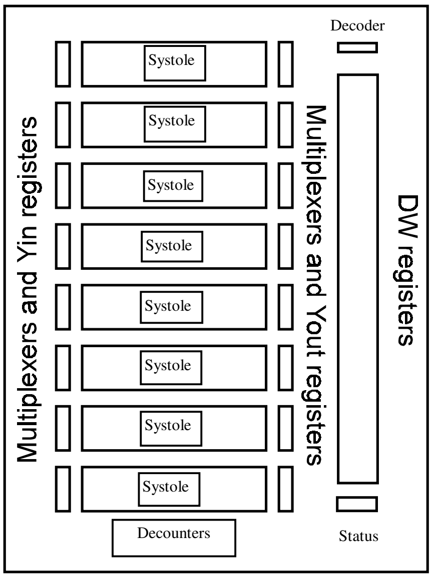

Figure 7: Convolver block diagram

VLSI implementation methodology

A set of generators has been developed separately during the design of fully parametrized libraries of arithmetic operators (fixed and floating point) for signal processing [19]. This set of generators has been intensively used in the design of the image convolver. The macro-blocks generated are surface and delay optimized.

The chip was designed hierarchically using the generated macro-blocks which in their turn use smaller macro-blocks and macrocells. Manual placement of the blocks and routing of certain signals (power signals) permitted a large gain in space.

Performance of the chips

Technology	ES2 ecpd07
Transistors	240000
Surface	84 mm²
Frequency	40 MHz
Instruction cycle	25 ns
Number of pins	184
Package	PGA 184 pins

Table 1. Characteristics of the chip

A micrograph of the chip is shown in the figure below.

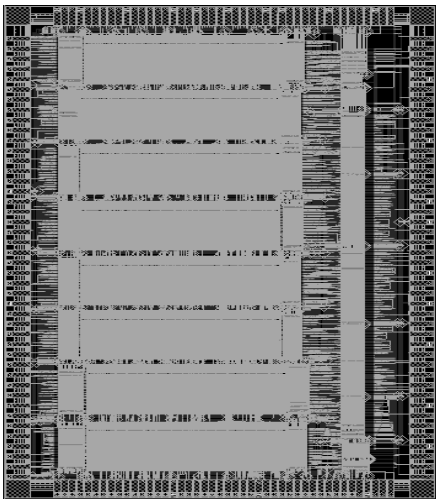

Figure 8: microgph of the 8D32 image convolver

Comparative results

We provided here in table 2 some results related to the performance of the fabricated chip relatively to available commercial circuits.

Image size 256x256
Mask size 31 x 31
Coefficients size 13 bits
Pixels size 8 bits
Results size 32 bits

Convolving circuit	Time
TMS320C50 (35 ns)	6,9 s
Convolver chip 8D32 (25 ns)	228 ms
Convolver chip 32D32 (25 ns)	57 ms

Table 2. Performance of the chip

Conclusion

We have implemented a very powerful chip to use in the computation of convolutions of images with large size kernels like the Gabor filters [16]. The use of this chip will be of great contribution in the development of performant systems for real-time pattern recognition.

The use of the module generators concept and a hierarchical design made the VSLI implementation efficient and flexible. The design with generators eases the convolver parametrization (coefficients precision, number of bits of pixels, ...). These generators have been developed with the assistance of a CAD tool (GenOptim) developed in the laboratory [2]. This tool allows the design of optimized macro-blocks. Moreover these generators are portable on different VLSI CAD tools (Cadence, Compass, Alliance) and are mappable on different cells libraries (ES2, Alliance, Xilinx).

BIBLIOGRAPHY

[1] Houelle A., Mehrez H., Vaucher N. "*On portable Macro-cell generators using the fully 754-IEEE standard* ", proc. ICSPAT 94, Dallas Texas USA, Oct. 18-21, 1994, V2 pp1749-1754.

[2] Houelle A., Mehrez H., Vaucher N., *Méhodologie de conception de générateurs portables de macro-blocs fonctionnels optimiseé en surface et en performance*, MASI report, December 1993.

[3] D. Johansen, "*Bristle blocks: a silicon compiler* " proc 16th ACM IEEE DAC, San Diego, June 1979.

[4] A. Greiner, F. Pêcheux "*ALLIANCE: A complete Set of CAD Tools for teaching VLSI Design* ", proc. Third Eurochip Workshop, p 230-237, 1992.

[5] Mehrez H., *Algorithmes et Architectures VLSI pour le Traitement Numerique*, UPMC, Laboratoire MASI/CAO-VLSI, 1992.

[6] Renver D., Klar H., *A configurable convolution chip with programmable coefficients*, IEEE journal of solid state circuits, vol. 27, no. 7, July 1992

[7] Hall E., Wise G.L., *Some aspects of multidimensional convolution*, ICASSP 1991, vol. 4, pp. 2317

[8] Jutand F., Demassieux N., Artieri A., *A new VLSI architecture for large kernel real convolution*, ICASSP 1990, vol. 2, pp. 921

[9] Kung H.T., *Why systolic architectures?, Computer magazine*, vol. 15, No. 1, Jan. 1982, pp. 37-46

[10] Cohen H.A., *Symmetry considerations applied to hardware convolcvers for image filtering*, Proc. of the 1988 ICSMC, vol. 2, pp. 1128-1131

[11] Shirazi B., Mukherjee P., Hu Y.H., *VLSI implementation of a pipelined multiplier for systolic convolution*, International journal on Computer Aided VLSI., Design 1- 1989

[12] Chester D., Young R., Petrowsky M., *A fully systolic adaptive filter implementation*, ICASSP 1991, vol. 2, pp. 917

[13] Leighton T., Leiserson C.E., *Wafer-scale integration of systolic arrays*, IEEE transactions on computers, vol. C-34, No. 5, 1985, pp. 448-461

[14] Denyer P., Renshaw D., *VLSI signal processing: a bit-serial approach*, MicroElectronics Systems Design

[15] Ruetz P., Brodersen R.W., *Architectures and design techniques for real-time image processing IC's, IEEE Journal of Solid-state cicuits*, vol. Sc-22, No. 2, April 1987, pp. 233-250

[16] Orr R. S., *The order of computaion for finite discrete Gabor transforms*, IEEE transactions on signal processing, vol. 41, No. 1, January 1993

[17] Clark G.A., Hernandez J.E., Lawver B.S., Sherwood R.J., *Gabor transforms and neural networks for automatic target recognition*, Lawrence Livermore National Laboratory, Report UCRL-JC-105362, December 1990

[18] Flaton K.A., Toborg S.T., *An approach to image recognition using sparse filter graphs*, IJCNN 1, Jun. 1989, pp. 1313-1320

[19] Aberbour M., Houelle A., Mehrez H., Vaucher N., *A Fully Parametrized IEEE Floating Point Operators Library For Use In Digital Signal Processing*, ICSPAT 95, Boston MA USA, October 1995,

[20] Gabor D., *Theory of Communication*, Journal of the Institution of Electrical Engineers 93, part III., No. 26, Nov, pp. 429-457

[21] Clark M., Bovik A.C., Texture Discrimination Using a model of Visual cortex, 1986 International conference on Systems, Man, and Cybernetics, Atlanta, GA, Oct 14-17