

Image Convolution Circuit: Parallel and Parameterized Architecture and FPGA Implementation

Alexandre Marques Amaral

Pontifícia Universidade Católica de Minas Gerais
Av. Dom José Gaspar, 500 – Coração Eucarístico
Belo Horizonte – MG. CEP: 30535-610
55-31-3319-4305
alexmarques@ieee.org

Carlos Augusto Paiva da Silva Martins

Pontifícia Universidade Católica de Minas Gerais
Av. Dom José Gaspar, 500 – Coração Eucarístico
Belo Horizonte – MG. CEP: 30535-610
55-31-3319-4305
capsm@pucminas.br

ABSTRACT

The sequentially execution of image convolution operation does not supply some performance demands. We developed this research to propose a high performance implementation of this operation. In this paper, we present an image convolution circuit, with a parameterized and parallel architecture, implemented using a FPGA (Field Programmable Gate Array). We also describe the circuit architecture using VHDL - VHSIC HDL (Very High Speed Integrated Circuit Hardware Description Language), to simulate, synthesize and test its behavior. Analyzing the results, we notice that the circuit performance/cost is much better than the sequential processors and some related circuit researches, with increasing of the speed-ups whenever the kernel becomes bigger.

1. INTRODUCTION

The Digital Image Processing (DIP) is continuously growing in industrial, commercial and domestic applications. In lots of them, the quality and high performance demands of DIP operations always increase, especially in real-time applications. These facts contribute for the continuous evolution of DIP area. Among DIP operations, the convolution is one of the most important [1][2][3].

The execution of convolution operation is a critical performance point. Whenever implemented in sequential software and executed in a GPP (General Purpose Processor) or a DSP (Digital Signal Processor), its performance usually does not supply the demand of some applications [2][3][4][5]. This happens, since such operation has a high workload, with lots of sub operations to be executed for each resulting pixel. Even when optimized, sequential software usually does not explore the implicit spatial and temporal parallelisms of the image convolution operation. Therefore, its sequential execution has a performance problem.

Fourier transform properties state that the convolution in the space domain is equivalent to the multiplication in the frequency domain [1]. Thus, the discrete convolution is used to implement image filtering operations in discrete-space domain. Each convoluted pixel is obtained by a summation of several pixel-weight products. Then, the resulting pixels compose the resulting image.

There is a large number of efforts to create and to optimize circuits and algorithms that implement image convolution operation with high performance [2][3][4][5]. Analyzing these and others researches, we conclude that there is not an implementation that has its architecture with high performance and high flexibility, exploring all the inherent features of the convolution operation.

Motivated by the mentioned performance problems, we developed this research, which the main objective is to design and

implement a parallel and parameterized circuit that execute digital image convolution with high performance and high flexibility of its implementation. We developed the parallel and parameterized circuit architecture to explore these features and improve the overall performance. In this paper, we present and verify our image convolution circuit with a VHDL description and FPGA implementation.

2. PROPOSED CIRCUIT

In Fig. 1, we present the block diagram of our circuit architecture. In this architecture, the input and output memories store the input and the output images, respectively. The addresser modules 1 and 2 address the input and output memories, in a predefined sequence, correctly loading and storing the pixels. The register bank optimizes the memory loads, working as a parallel source of the image pixels. The enable controller module controls the right enable of the registers, for the correct storage/reading.

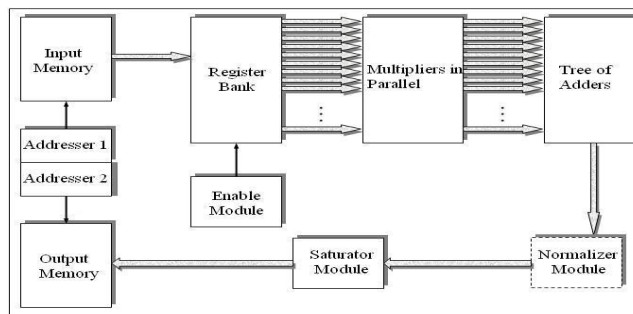


Figure 1. Circuit Architecture.

The circuit architecture has spatial and temporal parallelisms (SP and TP), exploring the convolution operation parallelism feature. The TP or pipeline is performed through the parallel execution of different steps of the overall operation. The TP is implemented in the adders disposed in a Wallace tree structure. The SP is performed through the parallel execution of a same sub-operation of the pipelined structure. The normalizer module, whenever required, makes the normalization of the resulting pixels. The saturator module makes the saturation, whenever required; making sure the pixel value is in the permitted range. The feature of parameterization was developed based on the adaptation aspect of the architecture. There are many possible parameters to be considered, making the architecture more flexible, eg., kernel size, communication widths, register bank depth, multiplication, addition and normalization operations and their implementations.

The convolution circuit architecture presented was designed, described and simulated with the structural mode VHDL language. Three variations of the kernel size parameter were described, which is 3x3, 5x5 and 7x7. Our circuit synthesis, map, place and route were done for the XC2V1500 device produced by

Xilinx. The choice of the FPGA implementation is based on several advantages of these devices [6]. Hence, our circuit implementation has lots of advantages over others hardware implementation technologies. Besides that, some optimizations for the convolution circuit whenever implemented, the performance is widely increased. These optimizations, implemented in our circuit, are: LUT-multiplication, kernel weights normalization, tree of adders' structure and more than one memory hierarchy for pixel storages. We chose LUT-multiplier modules core, since accessing a RAM-based LUT is much faster and costs fewer device resources, than executing the multiplication in a full multiplier module. The choice of adder cores is due to their performance and area optimizations. These modules have a clock pulse input for operations synchronism.

3. RESULTS

In this section, we present the main circuit results obtained with the ISE and ModelSim package softwares. We also present timing simulated data compared to others implementations.

After observing the behavior simulation results, we notice that the circuit behavior is exactly the predicted, presenting the correct results in a predicted time. Since the circuit has a pipelined structure, there is initial response latency. As soon as the latency time passes, one output pixel is resulted in each clock pulse. With these performance times we can estimate the circuit overall performances, for different image and kernel sizes. The implemented circuit has a maximum clock frequency of 125 MHz.

In table 1 and figure 2, we present the temporal results of a 512x512 and 1024x1024 image convolution operations, respectively, executed in the circuit and other implementations. The processors results were analytically obtained with Sandra benchmark data, hence the overhead of other jobs were not considered. The others system data were obtained from [3].

System	Architecture	Kernel	Time (ms)
DECChip 21064	Multiprocessor	5x5	220,00
Alacron's AI-860	I860 processor	8x8	66,10
TMS320C80	Multiprocessor	5x5	40,00
UWVSP5	DSP-Based	3x3	19,00
LSI	Hardwired ASIC	8x8	13,11
Cvwp	Sistolic	7x7	8,35
MAP1000	Media Processor	7x7	7,90
Blue Wave System	DSP-Based	3x3	7,20
PDSP16488	Hardwired ASIC	8x8	6,56
Circuit 1	Proposed Architecture	3x3	2,08
Circuit 2	Proposed Architecture	5x5	2,07
Circuit 3	Proposed Architecture	7x7	2,05
Circuit 4	Proposed Architecture	8x8	2,04

Table 1. Different implementation performances

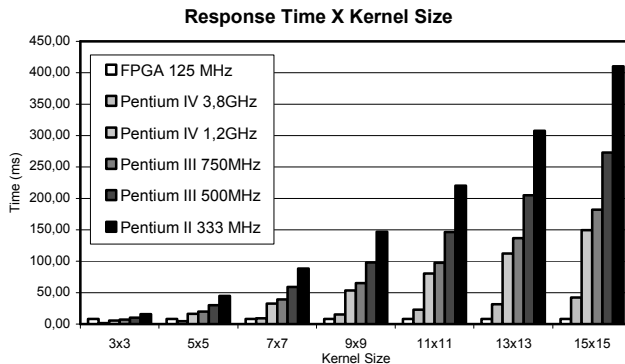


Figure 2. Execution time of 1024x1024 image convolution

Observing the table 1, we can compare our circuit performance with some other related and dedicated implementations.

Analyzing this table, we notice that our circuit execution time results are much better than all of the others. Considering the fact that these implementations are the most recent in the research literature, our circuit has contributions over them. With this performance, our circuit can be used as accelerator core into real-time systems. Observing the figure 2, we notice that our circuit executes the convolution with a much better performance over the GPPs, for kernels bigger than 7x7, although it is synchronized by a lower clock frequency. This difference becomes bigger when the kernel size increases. For smaller kernels our circuit has a great performance, in spite of being lower than the processors'. This happens because of the circuit's lower clock frequency, besides the additional resources into the processors' chip, e.g. cache memories. However, this lower performance is enough for high speed demands. We also observed that increasing the kernel size for the same image size, our circuit performance has a little increase, while the GPPs' one has a great decrease. This performance gain is because of our circuit parallel architecture and the reduction of the kernel number of iterations with the kernel size increasing. Thus, there are more instructions to be sequentially executed in the processor implementations.

4. CONCLUSIONS

After analyzing the results presented in the last section, we can conclude that we designed and implemented a circuit that executes image convolution with high performance, high flexibility of implementation and low cost, compared to commercial processors and related ASICs. The main contribution of this work is the presented parallel and parameterized circuit, its architecture, the low cost and optimized FPGA implementation.

Some of the future works are: the design and implementation of a reconfigurable image convolution circuit; a performance analysis of some architecture and implementation optimizations and their impacts on the resolution and precision; the design and implementation of an image coprocessor executing other operations, in reconfigurable devices; and design and implementation of its partial and dynamic reconfiguration.

5. ACKNOWLEDGMENTS

We would like to thank FIP/PUC Minas for the financial support.

6. REFERENCES

- [1] R.C. Gonzalez, R. E. Woods. Digital Image Processing. 2.ed. Prentice Hall, São Paulo, 2002. pp. 115-152.
- [2] S. Perri, M. Lanuzza, P. Corsonello, G. Cocorullo. A High Performance Fully Reconfigurable FPGA-based 2-D Convolution Processor, Microprocessor and Microsystem, The Netherlands 2004.
- [3] C. T. Huitzil, M. Arias-Estrada. Real-time Image Processing with a Compact FPGA-based Systolic Architecture, Journal of Real-time Imaging, Elsevier, Vol. 10, 2004. pp 177-187.
- [4] H. Jiang, V. Öwall. FPGA Implementation of Real-time Image Convolutions with Three-Level of Memory Hierarchy, FPT, Tokyo, Japan, 2003.
- [5] B. Bosi, G. Bois, Y. Savaria. Reconfigurable Pipelined 2D Convolver for Fast Digital Signal Processing. IEEE VLSI Systems Transactions, 1999. pp 299-308.
- [6] Xilinx: The Programmable Logic Company. <http://www.xilinx.com/>. August, 2005.