FPGA-BASED DIGITAL IMAGE PROCESSING ALGORITHMS IMPLEMENTATION OVERVIEW

Оверченко А.С., Момотова Ю.О.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Рогачевская А.И. – ст. преп.

Today FPGAs are often used for real-time image processing acceleration. This paper describes digital image processing (DIP) algorithms implementations. FPGAs and SoCs use these implementations as their main basis and acceleration.

Devices, that FPGAs use, are very popular today. These devices may contain FPGA as a part of system-on-a-chip (SoC) or as an independent integrated circuit.

FPGA (field-programmable gate array) is a digital integrated circuit consisting of programmable logic blocks and connections between them [1]. There are some main reasons for FPGA usage: acceleration, response time, advanced pipelining and parallelism of data processing, good price/performance ratio.

The purpose of this paper is to review digital image processing FPGA implementations, that have very high effectiveness and acceleration. This implementations can help the authors to create good FPGA pipeline for embedded and server image processing. The source [2] presents recent advances in FPGA image processing, such as filtering, segmentation, clustering, and compression. It demonstrates the possibilities of using FPGAs as an effective component, which can be quickly reconfigured to meet changing environmental demands in some cases.

There are few groups of DIP algorithms, that can be used for effective FPGA implementation. We will review window function, Fourier transform and neural networks.

Window function image processing includes most of the classic image processing algorithms. Their main idea is to use a convolution operation when the necessary filter is applied to each pixel of the image for getting the desired effect. The filter (window function) size usually varies from 2x2 to 7x7. This group of algorithms is usually used for median, edge detector filtering and high- and low-frequency filtering.

Some FPGA implementation examples were described in the paper [3]. The authors or that article used board with Spartan-6 chip. Xilinx ISE was used as a synthesizer with its various presets. Authors also compared memory usage, maximum operating frequency and the heat output of the circuit for 585x450 pixels images processing.

The Fourier transform is one of the few ways to obtain a signal frequency representation. Typically, that representation is used for image analysis or compression. For signal and image processing, a discrete version of the transformation is used.

In the paper [4] and [5], a fast Fourier transform (FFT) was implemented on the basis of Xilinx FPGAs. For implementation [4], floating-point numbers with reduced number width were used. That optimization allowed the author to get 2.5 times gain in the amount of computing blocks occupied on the chip without losing performance with up to 4 times reduction of the used memory.

The paper [5] has an integer implementation of the FFT. This implementation is a fully configurable, open-source kernel that can perform calculations at approximately 375 MHz for Xilinx Kintex 7 chips.

The neural networks are used in conjunction with classic image processing algorithms both to increase the speed of their operation and to find and describe implicit relationships that can improve the accuracy and efficiency of usual DIP algorithms. For example, in the article [6], the authors describe the results of implementing an optimized design for training convolutional neural networks using Xilinx Zynq 7 chip. This implementation shows excellent results of the training time in comparison with the CPU and almost identical to the GPU. But unlike its competitors, this implementation is energy efficient.

All the above mentioned articles and papers show that the use of FPGA as an accelerator is justified. This is due to the fact that well-written code with the necessary optimizations works faster than processing on the CPU, even taking into account data transfer from and to the FPGA.

References:

1. Maxfield, Clive. The Design Warrior’s Guide to FPGAs: Devices, Tools and Flows / Clive Maxfield. – NEWNES, 2004. – P. 182–197.
2. Bailey, Donald G. Image Processing Using FPGAs / Donald G. Bailey // Journal of Imaging. – 2019. – Vol. 5, no. 5. – 53 P. – Access mode : http://dx.doi.org/10.3390/jimaging5050053.
3. AlAli, M. I. Implementing image processing algorithms in FPGA hardware / M. I. AlAli, K. M. Mhaidat, I. A. Aljarrah // 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). – 2013. – P. 1–5.
4. FPGA-based floating point FFT implementation / Habr [Electronic resource]. – Access mode : https://habr.com/ru/post/322728/. – Access date 16.01.2020.
5. FPGA-based integer FFT implementation / Habr [Electronic resource]. – Access mode : https://habr.com/ru/post/420517/. – Access date : 16.01.2020.
6. Design of Convolutional Neural Network Based on FPGA / Sheping Zhai [et al.] // Journal of Physics: Conference Series. – 2019. – Vol. 1168, no. 6.