University of Applied Sciences and Arts
Northwestern Switzerland

Distributed FPGA for Enhanced Image Processing

Technical Requirements

Noah Hütter, Jan Stocker

Windisch, 21.02.2018

IME – Institute of Microelectronics
Steinackerstrasse 5
5210 Windisch
noah.huetter@students.fhnw.ch
jan.stocker@students.fhnw.ch

Table of contents

1 Dates

Project start	19.02.2018
Submission of planning	09.03.2018
Submission of documentation	31.08.2018
Presentation	14.09.2018

2 Contact

Noah Hütter
Student
noah.huetter@students.fhnw.ch
T +41 76 562 07 19

Jan Stocker
Student
jan.stocker@students.fhnw.ch
T +41 79 890 71 23

Michael Pichler
Coach
michael.pichler@fhwn.ch
T +41 56 202 75 26

Nomoko
matteo@nomoko.ch
quentin@nomoko.ch
kevin@nomoko.ch
T +41 44 309 81 11

3 Work Environment

The team Jan Stocker and Noah Hütter are working at FHNW Brugg-Windisch. They are doing a weekly digest by Mail to Nomoko.

4 Documentation

4.1 Organizational Requirements Sheet

A Rough time plan is created showing the milestones and the 4 tasks:

1. Research and feasibility check (20%)
2. Investigate and evaluate computer vision algorithms (30%)
3. Implement the distributed FPGA algorithm (40%)
4. Present you achievement (10%)

4.2 Technical Documentation

A technical report is written showing the problems and solutions. The code is commented. One documentation is sufficient for both companies.

4.3 Presentation

The team will perform a presentation about the results achieved at FHNW and Nomoko.

5 Goals

The project is divided into two parts.

5.1 First Semester

By the 12/01/2018 a working platform is implemented with the following features:

- Data transmission from PC to FPGA and back
- Basic image processing task performed by the FPGA
- Keep the possibility open to distribute the workload to multiple FPGAs

5.2 Second Semester (Bachelor Thesis)

- Distribute the workload onto multiple FPGAs
- Implement a more demanding image processing task
- Benchmark the performance against a CPU/GPU

6 Technical specifications

6.1 Data Transmission

The data is transmitted over Ethernet to the FPGA system. The outcome of project 5 (the UDP file transfer protocol) serves as a base and is further improved in the following aspects:

- Implementation of reliable transfer
- Make use of the entire gigabit LAN connection
- Write a computer side cross-platform implementation to transfer data

Furthermore, the subsequent modifications are investigated on their impact on performance:

- The use of jumbo frames (packet size larger than 1600 bytes)
- Heartbeat from the FPGA to the computer instead of packet acknowledgment

6.2 Data Format

The images have the data format of tiff and for the raw images dng.

6.3 Data Size

The input images are in the order of 1500M Pixel.

6.4 Image Processing Algorithm

The following image processing algorithms are considered:

- Debayering
- Color space transform
- Local contrast enhancement
- Edge detection

Furthermore, multiple different algorithms may be implemented to perform multiple tasks with or without reconfiguring the FPGA:

- Multiple algorithms on the FPGA, select through Ethernet
- Different bit streams with different algorithms, partial reconfigure by the FPGA itself

6.5 Benchmark

The implemented algorithm(s) are compared with a solution based on CPU/GPU in the following criteria:

- Throughput
- Latency
- Image quality
- Maximum clock frequency

The communication part is also benchmarked in terms of:

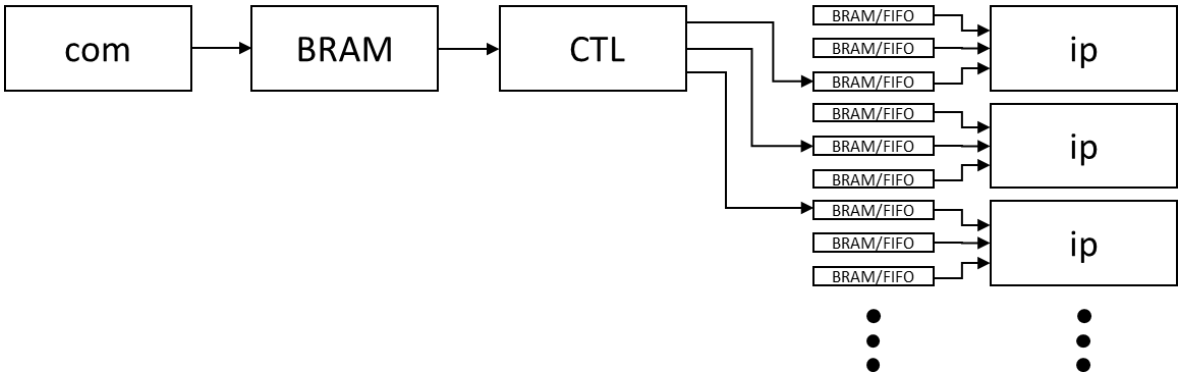
- Throughput
- Latency
- Packet loss

6.6 Distributed Processing

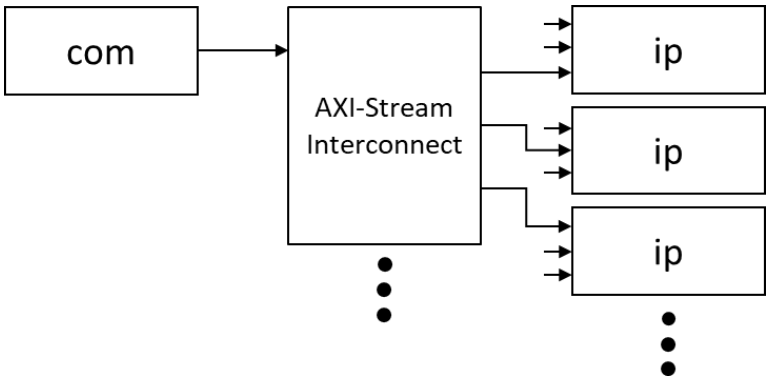
6.6.1 Inside FPGA

Data transmission inside the FPGA must be defined. Possible solutions are FiFo, Block RAM and Streaming. This depends on the image processing algorithm and what image data it requires. The following approaches are examined and compared.

Block RAM
Every image processing core is connected to block memory. A control block distributes the incoming data among the cores and handles the data flow.



Streaming
The connection inside the FPGA is over a stream interface. The data are distributed to the image processing core by a AXI-Stream interconnect.



6.6.2 Across FPGA

Develop a PC application that takes the image data and handles the communication with the FPGAs. A minimal console application serves the purpose. Cross-platform compatibility (Mac/Windows/Linux) is necessary. The FPGAs don't communicate among themselves.

6.7 Platform

The platform used is a Xilinx AC-701 board with Artix 7 series FPGA provided by Nomoko. It features:

- DDR3 SODIMM 1GB up to 533MHz / 1066Mbps
- 10/100/1000 Mbps Ethernet (RGMII)
- Onboard JTAG configuration circuitry to enable configuration over USB

6.8 Design Entry

6.8.1 Image Processing

The image processing algorithm is programmed in C/C++ and then compiled to HDL using the Vivado HLS environment. VHDL is used as hardware description language.
Develop a deep understanding on the Vivado HLS workflow.

6.8.2 Communications

The communication is implemented using configurable IP blocks provided by Xilinx or others. VHDL is used as hardware description language.