

Distributed FPGA for Enhanced Image Processing

Local contrast enhancement using a Wallis filter on multiple FPGAs with images of a resolution of 1500MP for use in virtual reality

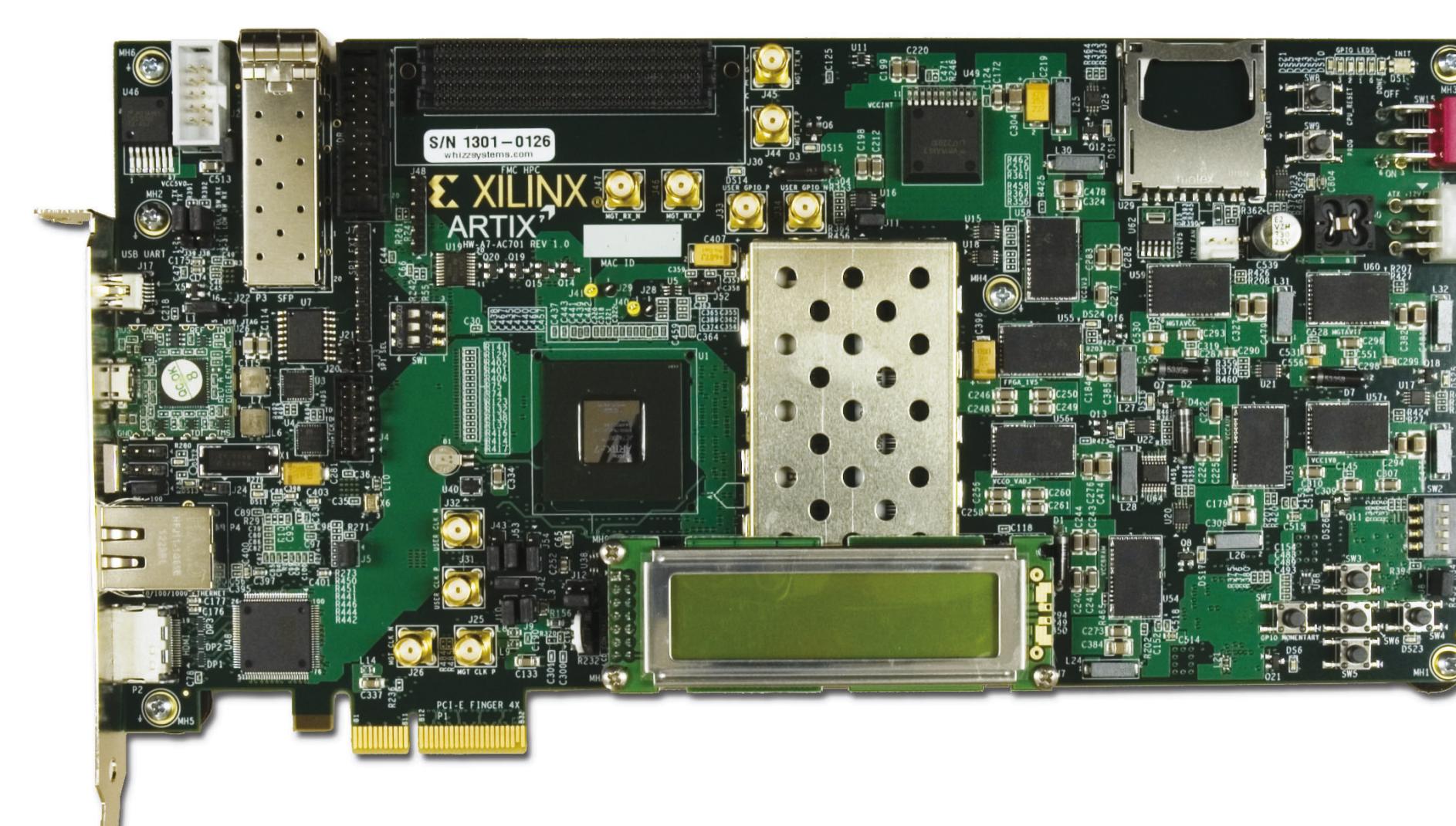


Fig. 1-3: Original image processed with the Wallis filter on the FPGA produces the output image with a local contrast enhancement

Introduction

A dedicated hardware image processing algorithm using FPGA was implemented that is scalable onto multiple FPGAs. In a first run High Level Synthesis was used to describe a Wallis local contrast enhancement filter in C/C++ language that was then synthesized to hardware description language. To further improve throughput a VHDL solution was implemented.

Image Processing Benchmark

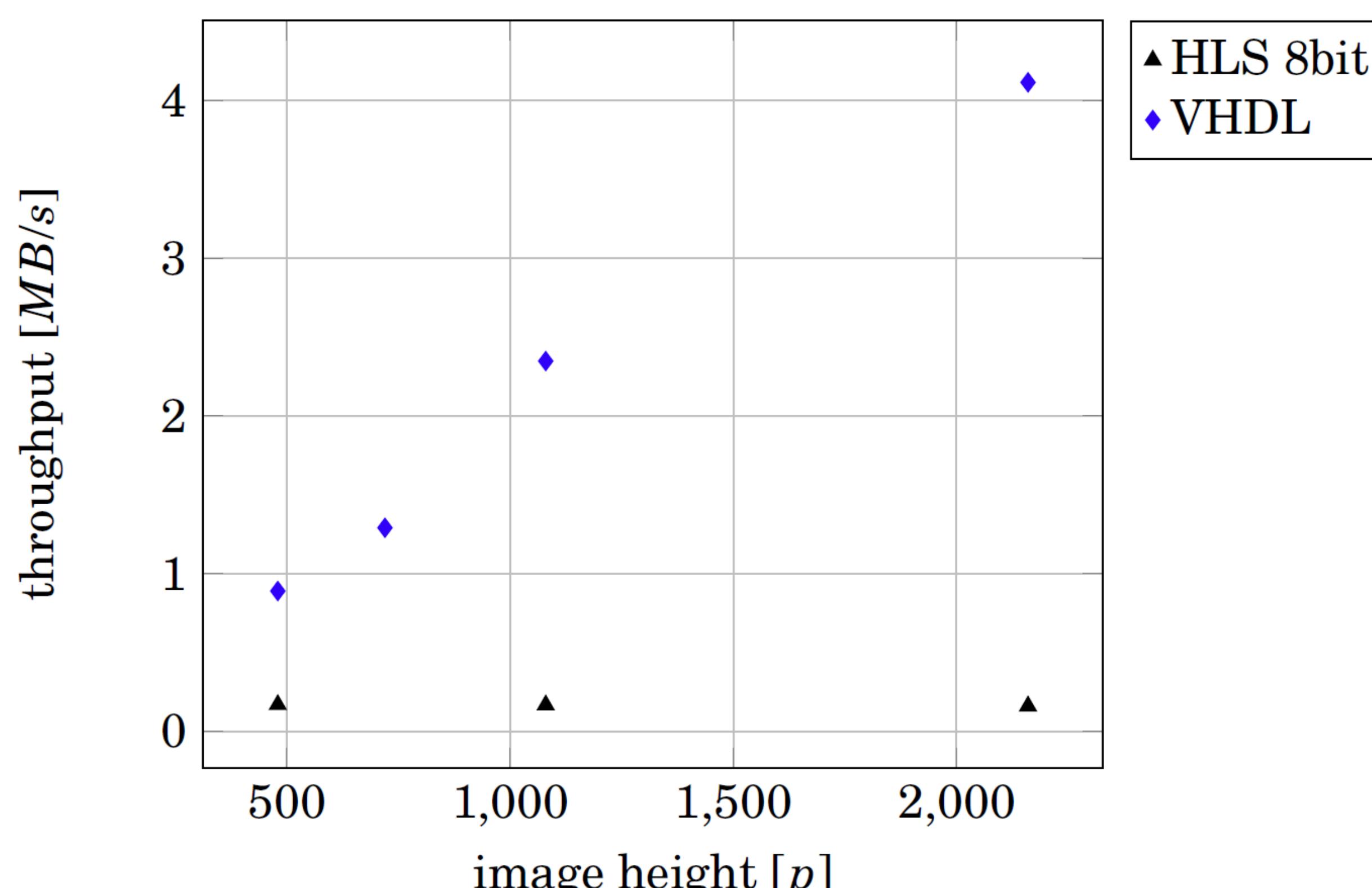


Fig 4: Throughput of the FPGA implementations

Results

The result is a complete image processing pipeline that begins on a PC where the input image is sent via Ethernet to the FPGA where it is processed and sent back to the PC. The achieved image throughput is 4.1MB/s primarily limited by the image processing core (fig. 4). The Wallis filter core alone is capable of processing up to 125Mp/s on the input. Concepts on scalability show how the processing power of FPGAs can be exploited if multiple image processing cores were implemented on one FPGA and also multiple FPGAs would work on a network (fig. 5).

Students: Noah Hüttler & Jan Stocker

Customer: Nomoko AG

Expert: Dr. Jürg M. Stettbacher

Examiner: Michael Pichler

michael.pichler@fhnw.ch

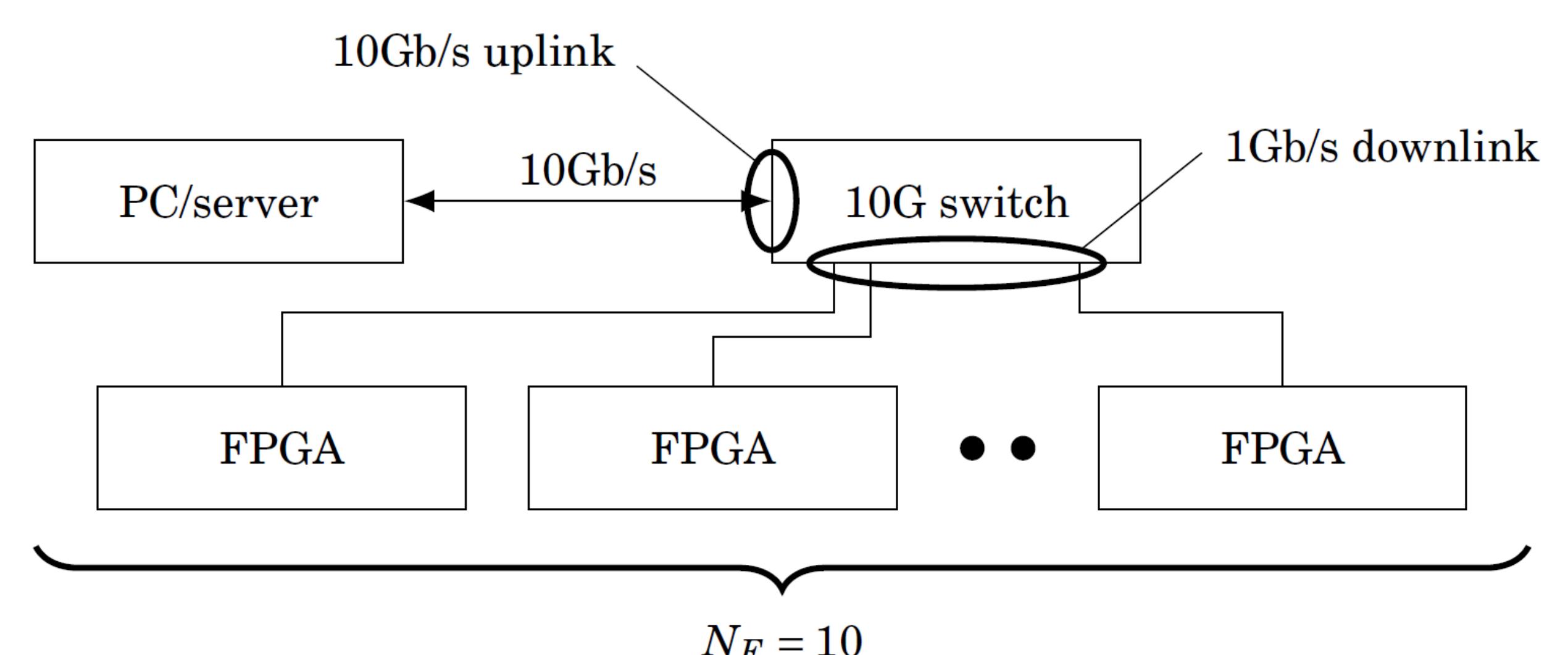


Fig 5: Scalability of an FPGA network with 10 FPGAs

FPGA vs. CPU

A benchmark shows the performance differences between a CPU based and FPGA based solution (fig. 6). The throughput of both FPGA implementations (VHDL and HLS 256bit) can be improved by implementing the Wallis filter multiple times on one FPGA. By implementing the HLS solution twice or the VHDL solution three times, a regular CPU could already be outrunned by one FPGA.

Theoretical Maximum vs CPU

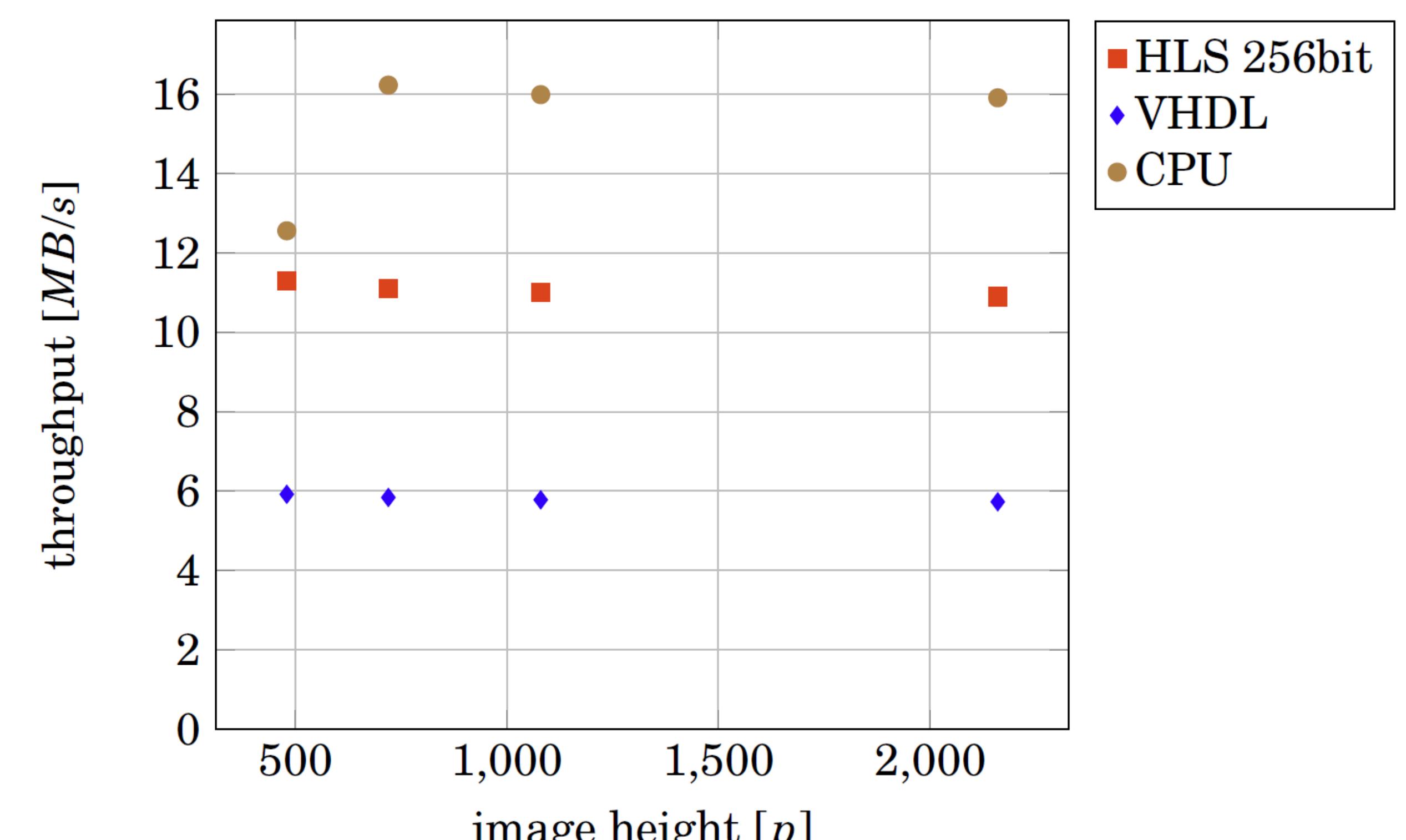


Fig 6: Theoretical throughput of a CPU solution and two FPGA solutions

- FPGA programmed using C/C++
- 1500MP processing within 3s
- Throughput: 500Mp/s (4 FPGAs)