High-Performance Computing Group

Prof. Dr. Christian Plessl

Computer Science Department and

Paderborn Center for Parallel Computing

Analysis of Bit-level Operations on FPGAs

Bachelor Thesis

At a glance

- Research and model theoretical throughput of bit-level operations on FPGAs
- Investigate resource efficiency of building blocks with different specifications
- Integrate blocks into complete demonstrator application
- Analyze practical limits and integration overhead

Several applications, most prominently from the bioinformatics domain, are performance limited by highly parallel and often specialized bit-level operations. One example of these operations is the commonly used population count (*popcount*) operation, that counts the number of 1 bits in a binary word and is used in different applications. FPGAs are an excellent hardware architecture for such operations, because each of their hundreds of thousands of lookup tables (LUTs) can be configured to perform any bit-level operations for one output bit given a few input bits. However, in practice it can be challenging to harness the full potential of the hardware when implementing such operations with high-level synthesis (HLS) tools [1,2]. The goal of this thesis is to understand these limitations better and explore ways to overcome them.

x3	x2	x1	x0	c1	c0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
…	…	…	…	…	

Table 1: Truth table for two least significant output bits (c1, c0) of 4-input popcount(x) to be implemented in FPGA LUTs.

```
1   char cnt = 0;
2   for (char i = 0; i < 64; i++)
3       cnt += (x >> i) & 0x1;
```

```
1   cnt = std::popcount(x);
```

Listing 1: (Top) Naïve implementation of popcount. (Bottom) Library version (e.g. oneAPI). Will it be more efficient?

In this thesis project, you will first research existing publications on theoretical potential and practically achieved results for bit-level operations on FPGAs. For a few selected operations, including the popcount, you will then investigate reported resource consumption values when implementing them with high-level synthesis (HLS) tools or exemplarily with hardware design languages (HDL). By integrating multiple different or identical blocks into a larger HLS-based demonstrator application, you will be able to analyze scalability challenges and resource overheads and explore best practices to overcome those.

Further reading:

- [1] Z. Jin and H. Finkel. Population count on intel® CPU, GPU and FPGA. In Proc. Int. Symp. on Parallel and Distributed Processing (IPDPS), 2020. https://ieeexplore.ieee.org/abstract/document/9150362
- [2] J.-O. Opdenhövel, C. Plessl, and T. Kenter. Mutation tree reconstruction of tumor cells on FPGAs using a bit-level matrix representation. In Proc. Int. Symp. Highly-Efficient Accelerators and Reconfigurable Technologies (HEART), 2023. https://dl.acm.org/doi/10.1145/3597031.3597050

Contact:

Tobias Kenter, Phone: 05251/60-4340 E-Mail: kenter@uni-paderborn.de

For information on theses offered by the HPC group or PC2 please visit:
https://en.cs.uni-paderborn.de/hpc/teaching/open-theses