*September 2017*

# Highly Parallel Lattice QCD Wilson Dirac Operator with FPGAs

Thomas Janson [a,1] and Udo Kebschull [a]
[a] *IRI, Goethe-Universität Frankfurt am Main, Germany*

**Abstract.** In this paper, we discuss a highly parallel implementation of the lattice QCD Wilson Dirac operator on FPGAs. The operator is described as a data-flow graph using OpenSPL and acts on a four-dimensional lattice collecting all next neighbour terms. The so implemented kernel fits on an Altera Stratix V FPGA using fixed-point arithmetic. This allows us to compute all arithmetic operations simultaneously. In addition, the OpenSPL language shows also an implicit optimized locality expressed by the offset operator which is studied on the presented implementation. The lattice is held in DDR3 memory on the FPGA accelerator card and streamed into the FPGA over six memory channels where we use the MAX4 card and the software framework from Maxeler. The operator is memory bound and has an equivalent arithmetic intensity of 0.92 FLOPs/Byte. With a clock frequency of 133 MHz, we get an equivalent theoretical peak performance of 176 GFLOP/s. Therefore, we also address the memory interface and memory access pattern.

**Keywords.** FPGA, parallel computing, lattice QCD, high performance computing

## 1. Introduction

Lattice QCD is one of the most compute intense simulations to solve problems of quantum chromodynamics (QCD) which is a theory of the strong interactions between quarks and gluons. In this paper, we do not concentrate on the physical aspect, but we take one of the algorithms from this simulation framework for an implementation on FPGAs. Here, the most compute intense algorithm is the Wilson Dirac operator $\not{D}$, which is introduced at first, where we use only the matrix-vector notation. We investigate the possibility by using the presented approach to fully enable parallelism and next neighbour search of the Dslash operator. The algorithm is memory bound on almost all compute platforms, e.g. Xeon Phi or GPUs [9,11]. It turns out that the implementation shown in this work is also memory bound. We have developed and tested the operator using a compute framework from Maxeler Technologies [6]. Therefore, a short overview of the compute framework is given, before the data-flow graph of the Dirac operator is shown.

There is also a former work from Owen Callanan [1,2] who reached reasonable results compared to CPU based solutions in 2006. The implementation is done on an Alpha Data ADM-XRC-II board equipped with a Xilinx Virtex-II FPGA device using Handel-C and logarithmic arithmetic [1]. It performs with 1320 MFLOP/s. We reach a significant

[1]Corresponding Author: Thomas Janson, Goethe-Universität Frankfurt am Main, Senckenberganlage 31, 60325 Frankfurt am Main, Germany; E-mail: janson@iri.uni-frankfurt.de.