Lattice Quantum Chromodynamics on FPGA hardware

William Detmold and Phiala Shanahan

Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

September 1, 2020

Field-programmable gate arrays (FPGAs) offer a computational paradigm that can in principle provide a low-cost, high-performance alternative to general-purpose high-performance computing and accelerator architectures such as graphics processing units (GPUs). These devices comprise user-configurable connections amongst a predefined set of logical units and can be optimized at the hardware level for a specific calculation; the space of possible optimizations in computational representations and data localization in particular is large, and by exploiting them, calculations can be performed orders of magnitude more efficiently than on regular CPU hardware. While FPGAs have existed for many years and have seen much use in custom electronics such as trigger implementations in detector experiments and inference [1], it is only recently that the available hardware has evolved to the point where they provide a viable resource for general scientific computing. Rapid developments in hardware driven by large-scale uptakes in industry (e.g. Microsoft's cloud platform and Bing search infrastructure), and the availability of higher-level programming models that abstract away hardware details, have precipitated this transition and offer a positive outlook for future rapid evolution. Recently, FPGAs are appearing in HPC systems such as the `cygnus` computer at the University of Tsukuba in Japan. For the purposes of this letter, FPGAs can be considered as accelerator cards added to a CPU-based system.

Lattice Quantum Chromodynamics (LQCD) [2, 3, 4] is the only known method to address low energy hadronic and nuclear physics calculations in the Standard Model (SM). At intermediate stages, this numerical approach proceeds by defining a 4-dimensional spacetime grid (lattice) on which the quark and gluon degrees of freedom are defined. Physical results are defined as integrals over these degrees of freedom, $N_{\rm dof} = \mathcal{O}(10^9{-}10^{12})$ variables in state-of-the art computations, in the limit that this discretization is removed. LQCD calculations therefore amount to the evaluation of such integrals using importance sampling Monte-Carlo methods. LQCD presents a extreme computational challenge and a large sustained algorithmic and software development effort over the last decades has brought the field to a point where many relatively simple properties of hadrons such as the proton and heavy mesons can be computed with high fidelity and rigorously controlled uncertainties. These calculations underlie the critical role that LQCD plays in high-energy physics (HEP), amongst other things enabling determinations of many of the CKM matrix elements and providing the most precise value of the strong coupling [5, 6, 7, 8, 9, 10, 11]. However, there are many new and emerging opportunities for LQCD to contribute to the HEP mission, many of which require more challenging calculations than are presently possible, either due to precision requirements or to the complexity of the process under consideration. For example, a high precision determination of the proton radius would provide a SM benchmark to confront discrepant measurements from electronic and muonic probes of hydrogen. Similarly, in intensity frontier experiments such as those seeking to directly detect dark matter, nuclear targets are essential and SM predictions require calculations of nuclear matrix elements of operators arising from BSM–SM couplings.

The dominant computational task in LQCD is the evaluation of inverses and determinants of the "Dirac operator", a sparse matrix of size $N_{\text{dof}} \times N_{\text{dof}}$ that appears in the requisite integrals. Fundamentally, these tasks amount to solving very large linear systems. First attempts to address LQCD computations by implementation of these linear systems on FPGAs date back to 2006 [12] with more recent work in Refs. [13, 14, 15, 16]. However the LQCD community has a long tradition of developing and utilizing novel and custom hardware for its computational needs, with notable machines based on purpose-built ASICs such as QCDOC [17]. So far, FPGA implementations of one of the simplest LQFT algorithms have been developed, and these are not yet sufficient to provide a viable alternative to mainstream high-performance computing. The most sophisticated work is that in Ref. [16] which implements the Conjugate Gradient (CG) linear solver algorithm, offloading the most computationally expensive kernel for execution on the FPGA and makes use of hardware kernels at multiple different precisions and cyclic buffers. The performance achieved in this implementation on a single Xilinx Alveo U280 accelerator card reached 600 GFlops, similar to latest generation GPUs using the QUDA lattice field theory library[18]. This is impressive and substantial progress, but the CG algorithm is far from the state-of-the-art and significant advances are necessary in order for FPGAs to be considered feasible for large-scale LQCD calculations. Over the timeframe of the ongoing planning process, we anticipate that these developments will be undertaken and an FPGA-based approach to LQCD may be a viable alternative to the current paradigm. Some particular directions in which advances are necessary are:

- *State-of-the-art inverter algorithms*

  The dominant computational task in LQFT is the inversion of very large, but very sparse matrices. There are many numerical methods that can be used for this task; the conjugate-gradient (CG) algorithm is one of the simplest of these methods and is the one for which FPGA implementations have already been developed. However, more efficient algorithms exist and are used for state-of-the-art LQFT calculations on traditional high-performance computing platforms and GPU systems. In particular, *algebraic multigrid* (AMG) algorithms, which exploit algebraic structure within the matrix, are proven to be optimal at this task but are considerably more complex to implement and depend in greater detail on the particular matrix being inverted. As such, to be competitive with existing approaches, an FPGA based computer must have an efficient implementation of AMG.

- *Parallel implementation of inverter algorithms*

  Due to the four-dimensional nature of spacetime, the size of the matrices that must be inverted in state-of-the-art LQCD calculations is large, as reported above. This is far beyond the scale of problem that can fit in a single FPGA and thus must rely on parallel algorithms ultimately implemented across hundreds of FPGAs. A natural approach to parallelization that is used in more traditional HPC systems is to split the spacetime geometry into regions and assign each to a different computational unit. Because of the structure of the underlying physical interactions, parallelization is non-trivial as different FPGAs must exchange information. This requires efficient kernels for data exchanges and optimizing their interplay with the computational kernels. However, FPGA vendors have invested deeply in developing low-latency high-performance network implementations (FPGAs are extensively used in network switches) and there is reason to believe this will be transferable to LQCD.

- *Force calculations and other LQCD algorithms*

  Beyond the sparse-matrix inversion task, LQCD calculations involve a range of other operations that must be performed including the calculation of the gauge force terms used in the Hybrid Monte-Carlo algorithm that is used to generate the gluon field configurations used in LQCD. While not the dominant cost in current CPU implementations of LQCD, with accelerated inverters, the computational cost of such algorithms becomes relevant and will need to be ported to make use of FPGA hardware.

Topical Groups: ■ (CompF2) Theoretical Calculations and Simulation
■ (TF05) Lattice Gauge Theory
■ (CompF4) Storage and processing resource access (Facility and Infrastructure R&D)

Contact Information:
William Detmold: wdetmold@mit.edu

References

[1] Javier Duarte et al. "Fast inference of deep neural networks in FPGAs for particle physics". In: *JINST* 13.07 (2018), P07027. DOI: 10.1088/1748-0221/13/07/P07027. arXiv: 1804.06913 [physics.ins-det].

[2] Kenneth G. Wilson. "Confinement of Quarks". In: (Feb. 1974). Ed. by J.C. Taylor, pp. 45–59. DOI: 10.1103/PhysRevD.10.2445.

[3] Thomas DeGrand and Carleton E. Detar. *Lattice methods for quantum chromodynamics*. 2006.

[4] C. Gattringer and C. B. Lang. *Quantum Chromodynamics on the Lattice*. 3rd ed. Springer, Berlin, Heidelberg, 2010.

[5] Richard C. Brower et al. "Lattice Gauge Theory for Physics Beyond the Standard Model". In: *Eur. Phys. J. A* 55.11 (2019), p. 198. DOI: 10.1140/epja/i2019-12901-5. arXiv: 1904.09964 [hep-lat].

[6] Bálint Joó et al. "Status and Future Perspectives for Lattice Gauge Theory Calculations to the Exascale and Beyond". In: *Eur. Phys. J. A* 55.11 (2019), p. 199. DOI: 10.1140/epja/i2019-12919-7. arXiv: 1904.09725 [hep-lat].

[7] Alexei Bazavov et al. "Hot-dense Lattice QCD: USQCD whitepaper 2018". In: *Eur. Phys. J. A* 55.11 (2019), p. 194. DOI: 10.1140/epja/i2019-12922-0. arXiv: 1904.09951 [hep-lat].

[8] Vincenzo Cirigliano et al. "The Role of Lattice QCD in Searches for Violations of Fundamental Symmetries and Signals for New Physics". In: *Eur. Phys. J. A* 55.11 (2019), p. 197. DOI: 10.1140/epja/i2019-12889-8. arXiv: 1904.09704 [hep-lat].

[9] Christoph Lehner et al. "Opportunities for Lattice QCD in Quark and Lepton Flavor Physics". In: *Eur. Phys. J. A* 55.11 (2019), p. 195. DOI: 10.1140/epja/i2019-12891-2. arXiv: 1904.09479 [hep-lat].

[10] William Detmold et al. "Hadrons and Nuclei". In: *Eur. Phys. J. A* 55.11 (2019), p. 193. DOI: 10.1140/epja/i2019-12902-4. arXiv: 1904.09512 [hep-lat].

[11] Andreas S. Kronfeld et al. "Lattice QCD and Neutrino-Nucleus Scattering". In: *Eur. Phys. J. A* 55.11 (2019), p. 196. DOI: 10.1140/epja/i2019-12916-x. arXiv: 1904.09931 [hep-lat].

[12] O. Callanan et al. "High Performance Scientific Computing Using FPGAs with IEEE Floating Point and Logarithmic Arithmetic for Lattice QCD". In: *2006 International Conference on Field Programmable Logic and Applications*. Aug. 2006, pp. 1–6. DOI: 10.1109/FPL.2006.311191.

[13] T Janson and U Kebschull. "Highly Parallel Lattice QCD Wilson Dirac Operator with FPGAs". In: *Parallel Computing is Everywhere* 32 (), pp. 664–672. DOI: 10.3233/978-1-61499-843-3-664.

[14] Grzegorz Korcyl and Piotr Korcyl. "Towards Lattice Quantum Chromodynamics on FPGA devices". In: (2018). DOI: 10.1016/j.cpc.2019.107029. arXiv: 1810.04201 [cs.DC].

[15] G. Korcyl and P. Korcyl. "Investigating the Dirac operator evaluation with FPGAs". In: (2019). DOI: 10.14529/jsfi190204. arXiv: 1904.08616 [cs.DC].

[16] G. Korcyl and P. Korcyl. "Optimized implementation of the conjugate gradient algorithm for FPGA-based platforms using the Dirac-Wilson operator as an example". In: (Jan. 2020). arXiv: 2001.05218 [cs.DC].

[17] D. Chen et al. "QCDOC: A 10-teraflops scale computer for lattice QCD". In: *Nucl. Phys. B Proc. Suppl.* 94 (2001). Ed. by T. Bhattacharya, R. Gupta, and A. Patel, pp. 825–832. DOI: 10.1016/S0920-5632(01)01014-3. arXiv: hep-lat/0011004.

[18] Ronald Babich, Michael A. Clark, and Balint Joo. "Parallelizing the QUDA Library for Multi-GPU Calculations in Lattice Quantum Chromodynamics". In: *SC 10 (Supercomputing 2010)*. Nov. 2010. arXiv: 1011.0024 [hep-lat].