

# Acceleration of the Lattice Boltzmann Method for Fluid Flow

Kentaro Sano<sup>a</sup>, Oliver Pell<sup>b</sup> and Wayne Luk<sup>c</sup>

<sup>a</sup> *Department of Computer and Mathematical Sciences  
Graduate School of Information Sciences, Tohoku University*

<sup>b</sup> *Maxeler Technologies*

<sup>c</sup> *Department of Computing, Imperial College London*

## Overview

FPGAs (Field-Programmable Gate Arrays) are becoming increasingly attractive for high-performance scientific computing. FPGAs are high volume, off-the-shelf semiconductor devices containing programmable logic components, embedded arithmetic units, embedded memories and a programmable interconnection network. FPGAs have remarkably increased their potential for high-performance computing by integrating much more programmable hardware resources and increasing their operating frequency, and therefore recent leading-edge FPGAs can have peak floating-point computation performance surpassing that of typical microprocessors [1]. By designing a custom computing machine (CCM) on FPGAs, the properties, e.g., parallelism, regularity and homogeneity, of a specific application can be efficiently exploited by customized data-paths, customized arithmetic units and customized memory systems.

Although FPGAs are programmable, programming FPGAs requires designing hardware. Therefore, it is very difficult for software programmers to implement CCMs for specific applications on FPGAs without knowledge of hardware design. Maxeler Technologies' "A Stream Compiler" (ASC) [2] solves this designing problem for FPGAs. By automating the production of CCMs that process streamed data, ASC allows users to write code with statements similar to the C language [2][3]. ASC also supports floating-point computations with flexible precisions, which are very suitable for efficient resource utilization on FPGAs.

This paper shows that the lattice Boltzmann method (LBM) for computational fluid dynamics is suitable for stream processing. Using a state-of-the-art FPGA platform from Maxeler Technologies we design a stream accelerator running at only 67MHz, initially implemented with the x1 transfer rate of PCI-Express, which achieves 1.15 times faster LBM computation than a 2.2GHz Opteron processor. We estimate the speedup of an FPGA-based stream accelerator with the x8 transfer rate at 7.68x.

Related work has shown that FPGAs have significant potential for computational fluid dynamics. For instance, a single FPGA implementation of a 3D lattice gas model [4] can run 200 times faster than a software version on a 1.8GHz Athlon processor. It has also been reported that FPGA-based accelerators for computational fluid dynamics [5] promise large improvement in sustained performance at better price-performance ratios with lower overall power consumption than conventional processors.

## Implementation and Results

The lattice Boltzmann method [6] models fluids with fictive particles performing propagation and collision processes over a discrete lattice mesh. Although relatively large data-sets are necessary to define multiple particle distribution functions on each grid-point, the algorithm for LBM has simplicity and parallelism among grid-points.

Here, we implement the lattice Boltzmann method for the 2D orthogonal 9-speed (2D9V) model using ASC and a MAX-1 PCI-Express FPGA card. Fig. 1 shows the block diagram of MAX-I, where we have a Xilinx Virtex-4 FPGA, two DDR2 RAMs, and a PCI-Express interface with up to x8 transfer rate.

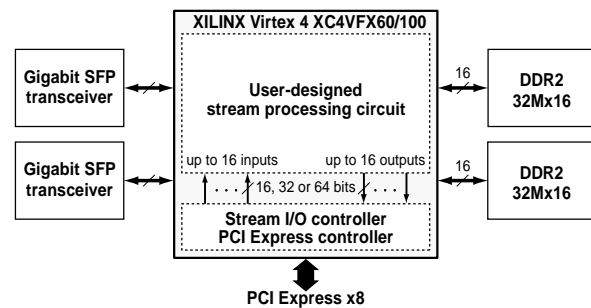


Fig.1 The block diagram of the Maxeler MAX-I FPGA card.

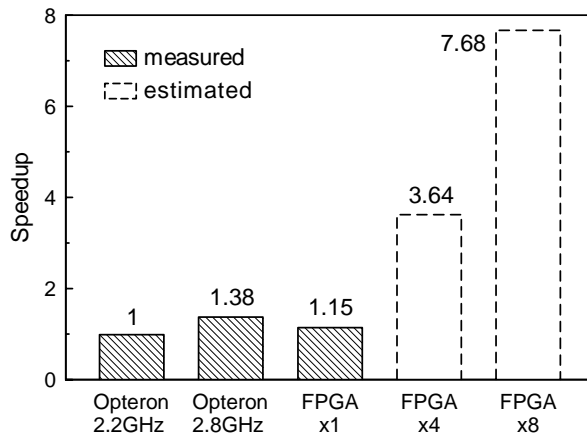


Fig.2 The measured and estimated performance comparison.  
FPGA acceleration has x1, x4 or x8 PCI-Express transfer modes.

We implement the prototype of the stream accelerator with the x1 mode of PCI-Express. This circuit operates at 67MHz consumes 86% of logic resources of the FPGA. For comparison, we also implement the stream computation in C++ for the Opteron processor, compiled using gcc with full optimization. For both FPGA and software computations, the code for the remaining translation and termination evaluation stages (including boundary computations) is written in C++ to be executed by the host PC. The host PC performs all the floating-point computations in single precision.

In the case of software computation, the Opteron processor running at 2.2GHz takes 0.0190 seconds per timestep on average, while an Opteron processor running at 2.8GHz takes 0.0137 seconds. On the other hand, the FPGA-based computation takes 0.0164 seconds. This means that with the x1 mode, we achieve 1.15 and 0.833 times faster computation by FPGA than by using the AMD Opteron processors running at 2.2GHz and 2.8GHz respectively. Based on these figures estimated speedups are 3.64x for PCI Express x4 mode and 7.68x for the PCI Express x8 mode compared to the 2.2GHz processor. Implementation of the PCI Express x4 and x8 accelerator circuits requires further optimization to fit the circuit into available logic resources and increase clock frequency. Fig. 2 summarizes the above performance results and estimation.

## Conclusion

We have shown a FPGA-based stream accelerator for the 2D9V lattice Boltzmann method. Although currently only implemented using PCI Express x1, the FPGA achieves faster computation than a 2.2Ghz AMD Opteron processor. Throughput will be improved in the optimized code for the same FPGA with the x8 transfer mode.

## Acknowledgements

This briefing note is a summary of a paper presented at the International Conference on Parallel Computational Fluid Dynamics [7].

## References

- [1] K. Underwood, "FPGAs vs. CPUs: Trends in peak floating-point performance," Proceedings of the International Symposium on Field-Programmable Gate Arrays, pp. 171–180, 2004.
- [2] Oskar Mencer, "ASC: A Stream Compiler for computing with FPGAs," IEEE Trans. on CAD of Integrated Circuits and Systems, vol.25, no. 9, pp.1603-1617, 2006.
- [3] Evgeny Fikshan, Yitzhak Birk and Oskar Mencer, "ASC-Based Acceleration in an FPGA with a Processor Core Using Software-Only Skills," Procs. of IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM2006), pp.271-272, 2006.
- [4] Tomoyoshi Kobori and Tsutomu Maruyama, "A High Speed Computation System for 3D FCHC Lattice Gas Model with FPGA," Procs. of International Conference on Field-Programmable Logic and Applications (FPL2003), pp.755-765, 2003.
- [5] William D. Smith and Austars R. Schnore, "Towards an RCC-Based Accelerator for Computational Fluid Dynamics Applications," The Journal of Supercomputing, vol. 30, no. 3, pp.239-261, 2004.
- [6] P. A. Skordos, "Initial and boundary conditions for the lattice Boltzmann method," Physical Review E, vol. 48, no. 6, pp.4823-4842, 1993.
- [7] Kentaro Sano, Oskar Mencer and Wayne Luk. "FPGA-based Acceleration of the Lattice Boltzmann Method," Intl. Conf. Parallel Computational Fluid Dynamics 2007 (parCFD2007), 2007.