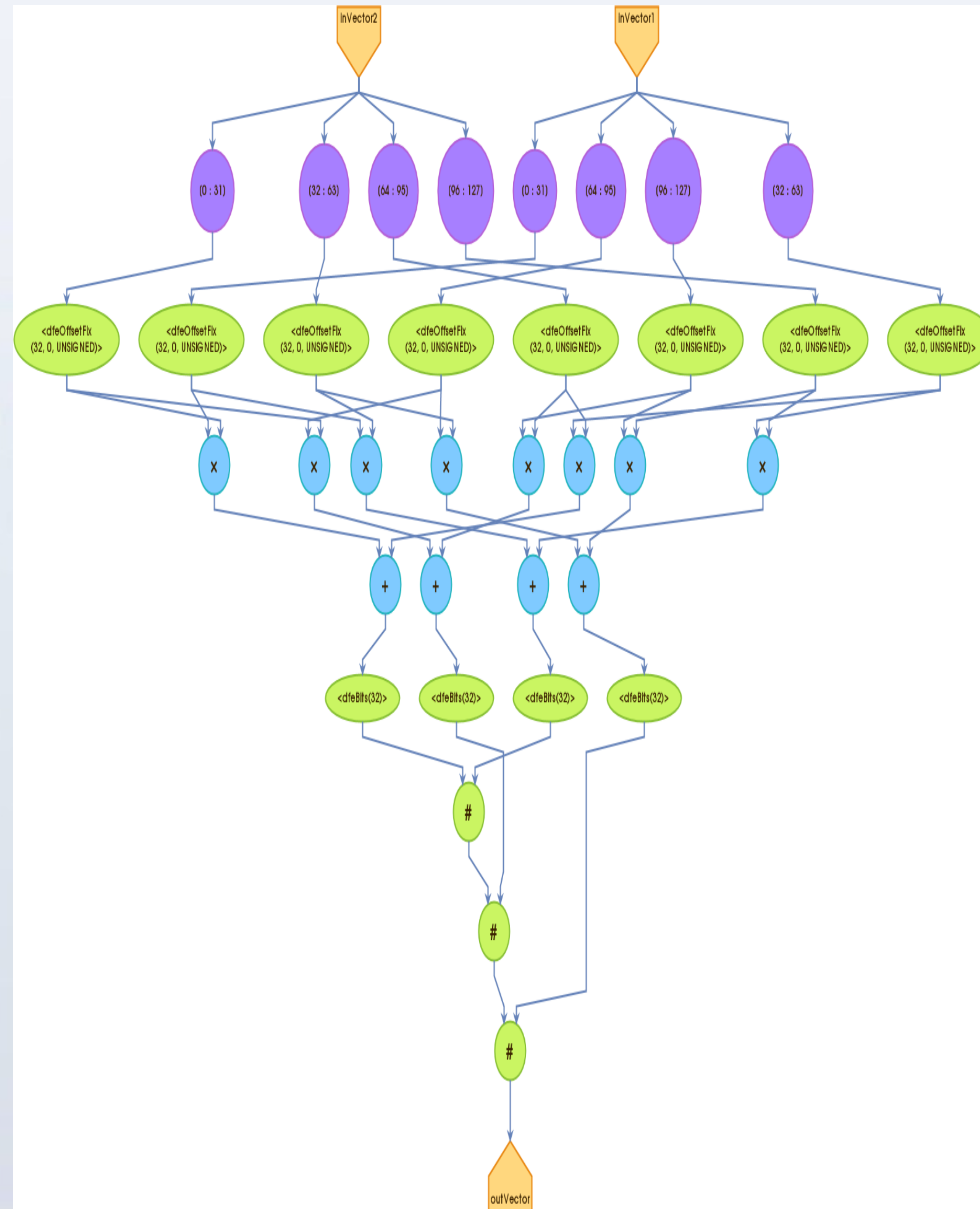


Accelerating Matrix Multiplication and Bubble Sort using Maxeler Dataflow Approach

Kristy Yoshimoto, Veljko Milutinovic (Mentor)

UROC, Luddy School of Informatics, Computing, and Engineering, Indiana University Bloomington

Matrix Multiplication



References

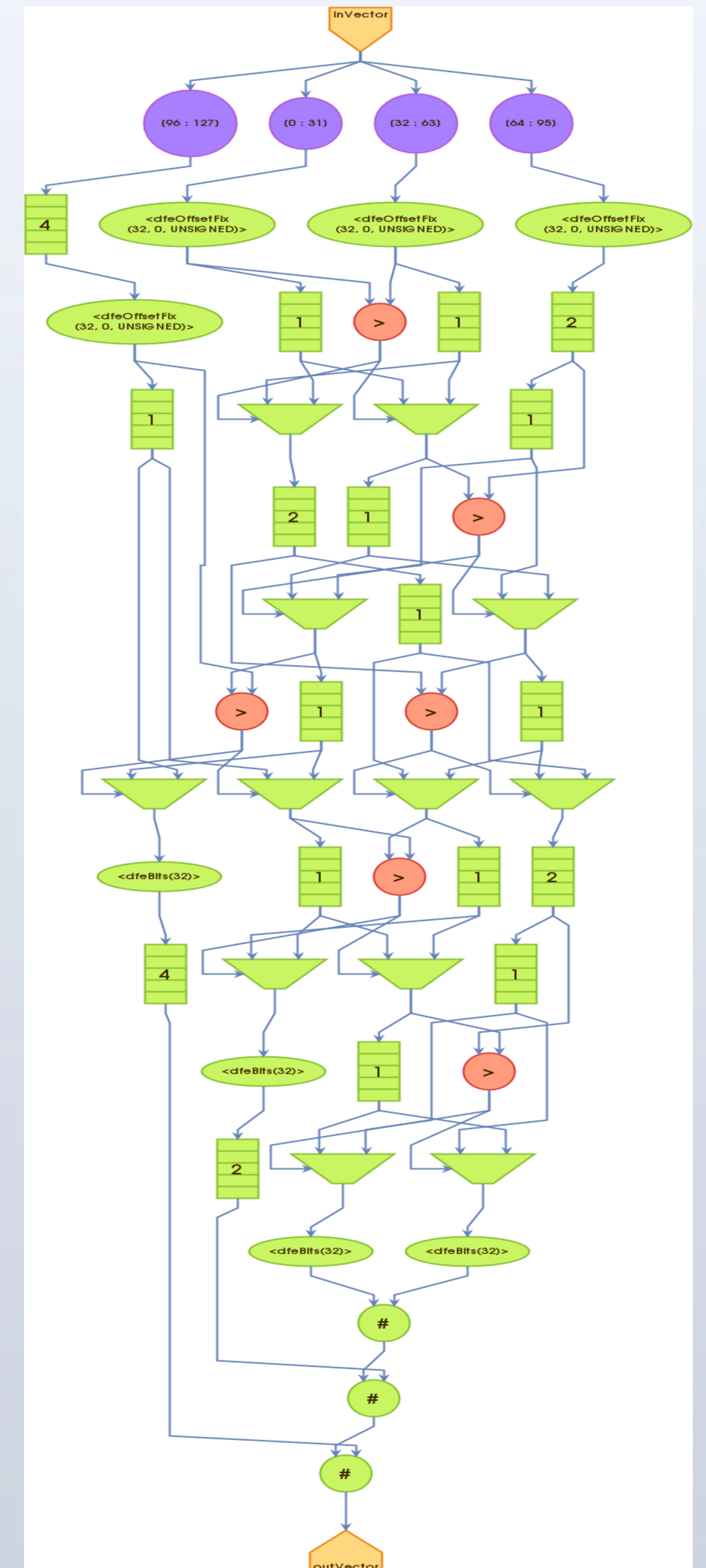
- Korolija, N., Djukic, T., Milutinovic, V., & Filipovic, N. (2013) Accelerating Lattice-Boltzman Method Using Maxeler DataFlow Approach. *The IPSI BgD Transactions on Internet Research*, 9(2), 34-41.
- Maxeler Technologies, (2015), Multiscale Dataflow Programming. *Maxeler Technologies*, 2015.1.1(1), 1-4.
- Milutinovic, V. (2017) A Dataflow course: Dataflow supercomputing. *Mediterranean Conference on Embedded Computing*, 6(1), 1-1.
- Milutinovic, V., Kotlar, M., Ratkovic, I., Korolija, I., Djordjevic, M., Yoshimoto, K., Klem, E., & Valero, M. (2018) The Ultimate DataFlow for Ultimate SuperComputers-on-a-Chips. *Supercomputing: Methodologies and Applications*, 1(1), 369-374.
- Milutinović, V., Trifunović, N., Korolija, N., Popović, J., & Bojić, D. (2017) Accelerating program execution using hybrid control flow and dataflow architectures. *Telecommunication Forum (TELFOR)*, 25(1) 1-4.
- Ranković, V., Kos, A., & Milutinović, V. (2013) Bitonic Merge Sort Implementation on the Maxeler Dataflow Supercomputing System. *The IPSI BgD Transactions on Internet Research*, 9(2), 5-10.
- Stojanović, S., Bojić, D., & Milutinović, V. (2013) Solving Gross Pitaevskii Equation Using Dataflow Paradigm. *The IPSI BgD Transactions on Internet Research*, 9(2), 17-22.
- Trifunovic, N., Milutinovic, V., Salom, J. & Kos, A. (1970). Paradigm Shift in Big Data SuperComputing: DataFlow vs. ControlFlow. *Journal of Big Data*, 2(1).
<https://doi.org/10.1186/s40537-014-0010-z>

Overview

Two programs were written and run on the dataflow machine at the Mathematical Institute of the Serbian Academy of Sciences. The first program performs matrix multiplication on two input streams and returns a single stream. All streams in this program are one-dimensional vectors of the same fixed length. In order to change the length, the size needs to be modified in the kernel, kernel manager, and host. Both inputs are created in the host and passed to the dataflow engine (DFE) at runtime. After all the calculations are performed in the engine, the output steam is passed back to the host. The second program implements Bubble Sort. Bubble Sort uses nested for-loops and the idea is to swap two neighboring indexes if it is out of order. In the host code, this is easily done by having a condition that checks if the item at the current index is bigger than the next item. If it is, the values stored in both indexes are swapped. Bubble sort is implemented in a very similar manner in the kernel.

Both programs demonstrate that simple control-flow solutions can be converted for dataflow programming. The benefits of using dataflow over control flow systems are faster runtimes and power reductions (Milutinovic, 2018, p 370). This is achievable due to the way DFE's are designed. A single DFE contains thousands of cores and each core computes a single type of arithmetic operation (Maxeler, 2015, p. 2) Multiple cores are able to process on neighboring streams simultaneously, and this is often referred to as computing in space (Maxeler, 2015, p. 2). While the difference in performance with small sets of data may be minute, it is on BigData and Machine Learning problems where dataflow programming noticeably triumphs control-flow implementations.

Bubble Sort



Contact Information: kyoshimo@iu.edu