Overview. My passion for system design brought me into research under Professor John Wawrzynek, who introduced me to a multi-university effort centered around high-performance computing and systems biology (2009 – 2010). Over time, my research interests generalized to reconfigurable computing (2010).

Systems Biology. From Spring 2009 to Spring 2010, I worked with Professors John Wawrzynek (at U.C. Berkeley), Wing Wong, and Garry Nolan (both at Stanford University) to create scalable Bayesian network inference accelerators.

The project's motivating application is to learn the structure of *Signal Transduction Networks* (STNs). STNs are chains of interconnected proteins that carry and amplify signals, detected at a cell's membrane, to intercellular components such as the nucleus or mitochondria. STNs are important in medicine because differences between individuals' STNs have been correlated to how effective clinical drugs have been in treating cancer and other human diseases. Professor Nolan's lab is investigating how to transform STNs into *Bayesian networks* (BNs). BNs are probabilistic graphical models that can illuminate how proteins within an STN interact. Professor Wong's lab is focused on how to *efficiently* infer a BN's structure from data that describes an STN. My, and Professor Wawrzynek's, interest is how best to map efficient BN inference algorithms (which I will refer to as *BN kernels*) to compute platforms such as *field-programmable gate arrays* (FPGAs).

My first contribution to the systems biology project was to integrate a set of input/output facilities into a prototype BN inference kernel so that the system could be used by biologists. FPGAs were seen as a compelling implementation platform because of their reconfigurability (allowing them to adapt to different networks) and gate-level customizability (allowing them to directly implement bit-level tasks within the algorithm). To connect the kernel to the outside world, I implemented a scalable stream, remote memory transfer, and control-level communication solution to and from a host general purpose processor. On top of transferring bits, I designed a script interface that allowed the FPGA to communicate with an internet application used by biologists. Throughout this project, I learned how to carefully communicate technical nuances with both engineers and biologists, which was necessary as my implementation was to be the "glue" between these two worlds. I also gained a deeper appreciation for heterogeneous systems, and how rate matching between pairs of interacting components can drive the design process. Upon completion, I gave a tutorial-style talk on the prototype system to the Berkeley-Stanford groups to facilitate use.

My next contribution to the systems biology effort was to optimize and scale the BN kernel itself. BN inference is a compute-bound problem that presents fine and course-grained parallelism. From these observations, I designed and implemented a multi-threaded, multi-core architecture customized for the BN application, that could scale to multiple FPGAs over a general-purpose mesh interconnect. In Fall 2009, I gave a talk on this scalability approach at the annual GigaScale Systems Research Center (GSRC) symposium and later demonstrated the system, running across four FPGAs, at the semi-annual Berkeley Wireless Research Center (BWRC) retreat. In Spring 2010, I was co-lead author on a publication surveying the system (which won *Best Student Paper Award* at the International Conference on Supercomputing, ICS'10) [1], and gave the paper talk in Japan during the conference proceedings. This project taught me the tradeoffs in custom-architecting a high-performance computing application. Moreover, I became more mature

in planning a long term research effort and exploring the right design alternatives so that the result was both tractable and of high quality.

Reconfigurable Computing. In Summer 2010, after my graduation, I continued my Berkeley-Stanford collaboration in order to answer research questions which had arisen amidst publishing at ICS'10.

The first question, which was a concurrent research endeavor by other members of Professor Wawrzynek's group, asked how best to raise the level of abstraction in FPGA computing. This problem was exemplified in my work submitted to ICS'10: the entire multi-FPGA system was custom designed at the *gate* level, and therefore took four months to implement. Professor Wawrzynek's group's approach to this problem is to look at an FPGA as a *micro-architectural template* which can be customized on a per-application basis. To explore this idea, I helped map a part of the BN inference kernel into the template-based framework in order to show the performance/area delta between my hand-optimized solution and the template-based approach. (The study was accepted for publication and will appear in [2]). From this project, I started viewing hardware systems as collections of micro-architectural patterns.

The second question examined the benefits of template-based architectures, built with application-specific knowledge, over programmable architectures such as *general-purpose graphics processing units* (GPGPUs). In order to minimize the BN kernel's off-chip memory footprint, I led an effort to reformulate the BN algorithm into a form that naturally mapped well to a GPGPU. Given this algorithm, the resulting FPGA implementation itself resembled an application-specific GPGPU. With two similar architectures executing the same algorithm, we were able to characterize normalized core performance across the two platforms. Our findings (to appear in [3]) quantitatively showed how reconfigurable logic placed at each GPGPU core could benefit compute-bound applications.

Working on [3] developed my engineering and scientific sophistication beyond that of my previous projects. Comparing GPGPUs to FPGAs requires expertise across both platforms, where each has a different set of design tradeoffs. Furthermore, I had to be very careful to design experiments that fairly represented equivalent operations across each device. This project, however, only studied workloads within the scope of BN inference. As my next project (which I discuss in my research proposal essay), I want to learn how techniques developed in this and similar projects can benefit larger classes of applications.

References

[1] N. B. Asadi, **C. W. Fletcher**, G. Gibeling, E. N. Glass, K. Sachs, D. Burke, Z. Zhou, J. Wawrzynek, W. H. Wong, and G. P. Nolan. *Paralearn:* a massively parallel, scalable system for learning interaction networks on fpgas. *Proceedings of the 24th International Conference on Supercomputing (ICS)*, 2010. New York, NY, USA: ACM, 2010, pp. 83-94. **Best Student Paper Award.**

[2] I. Lebedev, S. Cheng, A. Doupnik, J. Martin, **C. W. Fletcher**, D. Burke, M. Lin, J. Wawrzynek. MARC: A Many-Core Approach to Reconfigurable Computing. *To appear in the Proceedings of the 6th International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, 2010.

[3] **C. W. Fletcher**, I. Lebedev, N. B. Asadi, D. Burke, J. Wawrzynek. Bridging the GPGPU-FPGA Efficiency Gap. *To appear as a short paper in the Proceedings of the 19th International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2011.