JAMES COOLE

Raleigh, NC

[LinkedIn](#) | [Google Scholar](#)

EXPERIENCE

Cisco — Jan 2020 - Apr 2024
Technical Lead, Silicon One P4 Compiler & Simulator — *Remote*

· Contributed to the development of the P4 compiler and simulator for Silicon One programmable ASICs, underpinning Cisco 8000 routers, Catalyst 9600 and Nexus 9800 switches, and future products.

· Enabled support for new devices and device families, hardware blocks, and applications, collaborating closely with ASIC and P4 application/dataplane engineers.

· Responsible for optimizations to reduce the resource utilization of dataplane code, ensuring Cisco's growing applications fit within device constraints and improving quality of results (QoR) across devices.

· Led the evolution and modernization of Cisco's internal variant of the P4 language, supporting the refactoring of Cisco's P4 applications to better scale across multiple markets and devices.

Cisco — Jun 2016 - Jan 2020
Senior Software Engineer, Data Center CTO Office — *Research Triangle Park, NC*

· Worked on SmartNICs and P4 hardware synthesis in the Data Center CTO and CTAO offices, developing P4-to-RTL compilers targeting FPGA, eFPGA + ASIC, and ASIC technologies.

· Implemented as backends for The Linux Foundation's P4-16 reference compiler with hardware and technology-specific optimizations. Also investigated implementations leveraging Altera's OpenCL SDK.

University of Florida and NSF SHREC — Aug 2009 - May 2016
Graduate Researcher/Fellow — *Gainesville, FL*

· Developed reconfiguration contexts, an approach to FPGA high-level synthesis that provides very fast compilation and portability of kernel datapaths using coarse-grained FPGA overlays. Created overlay architectures based on CGRAs and datapath merging, along with mapping and PnR algorithms, and tools to automatically design overlay libraries based on an analysis of expected kernels.

· Created an FPGA high-level synthesis tool for OpenCL 1.1 using Clang and LLVM, targeting libraries of reconfiguration contexts to provide O(seconds)/kernel compile times. Implemented novel optimizations for OpenCL synthesis, including inference of sliding-window buffer hardware.

· Developed block-based place and route (PnR) techniques, enabling fast PnR of high-level designs by assembling netlists from a library of pre-synthesized coarse-grained blocks. Created a block-based variant of VPR and tools for generating block libraries via floorplaning and manipulation of full-detail PnR solutions. Included reverse-engineering portions of the Altera Cyclone III's routing architecture.

NASA Goddard Space Flight Center — May 2014 - May 2016
Computer Engineer — *Greenbelt, MD*

· Extended OpenCL high-level synthesis tool developed at UF by implementing the OpenCL 1.1 runtime, enabling dynamic synthesis of kernel source, overlay swapping via partial reconfiguration, and copyless buffer sharing with kernels via a custom Linux kernel module. Supported Xilinx Zynq SoCs.

· Integrated this framework into a CubeSat built around the Zynq-based SHREC Space Processor for an experimental mission on the International Space Station.

· Developed part of a real-time hyperspectral image processing pipeline using this framework, for use in Zynq-based earth observation satellite and UAV projects.

University of Florida   Aug 2008 - Aug 2009
Research Assistant   *Gainesville, FL*

· Researched and developed traversal caches, a hardware/software framework designed to improve memory performance for data structures requiring non-sequential accesses (e.g., tree traversal).
· Applied this framework to develop efficient FPGA accelerators for n-body simulation.

EDUCATION

University of Florida   *May 2016*
Ph.D. in Electrical & Computer Engineering
Dissertation: *FPGA Overlays and Runtime Synthesis for Flexibility and Productivity*

University of Florida   *May 2012*
M.S. in Electrical & Computer Engineering

University of Florida   *May 2008*
B.S. in Computer Engineering, *Magna Cum Laude*

TECHNICAL SKILLS

Languages & Libraries   C++, Python, P4, Verilog, LLVM
Hardware & Tools   Vivado, Quartus

SELECTED PUBLICATIONS AND PATENTS

· J. Coole and G. Stitt. Intermediate Fabrics: Virtual Architectures for Circuit Portability and Fast Placement and Routing. In *Proceedings of IEEE/ACM/IFIP Conference on Hardware/Software Codesign and System Synthesis*, CODES+ISSS, 2010.

· G. Stitt and J. Coole. Intermediate Fabrics: Virtual Architectures for Near-Instant FPGA Compilation. *IEEE Embedded Systems Letters*, 3(3), September 2011.

· J. Coole and G. Stitt. BPR: Fast FPGA Placement and Routing Using Macroblocks. In *Proceedings of IEEE/ACM/IFIP Conference on Hardware/Software Codesign and System Synthesis*, CODES+ISSS, 2012.

· J. Coole and G. Stitt. Fast, Flexible High-Level Synthesis from OpenCL using Reconfiguration Contexts. *Micro, IEEE*, 34(1), Jan 2014.

· J. Coole and G. Stitt. Adjustable-Cost Overlays for Runtime Compilation. In *Field-Programmable Custom Computing Machines, IEEE 23rd International Symposium on*, FCCM, 2015.

· C. Wilson, J. Stewart, P. Gauvin, J. MacKinnon, J. Coole, J. Urriste, A. George, G. Crum, E. Timmons, J. Beck, et al. CSP hybrid space computing for STP-H5/ISEM on ISS. In *Proceedings of the AIAA/USU Conference on Small Satellites*, 2015.

· J. Coole, J. Wernsing, and G. Stitt. A Traversal Cache Framework for FPGA Acceleration of Pointer Data Structures: A Case Study on Barnes-Hut N-body Simulation. In *Proceedings of the International Conference on Reconfigurable Computing and FPGAs*, ReConFig, 2009.

· D. Wilson, G. Stitt, and J. Coole. A Recurrently Generated Overlay Architecture for Rapid FPGA Application Development. In *International Symposium on Highly-Efficient Accelerators & Reconfigurable Technologies*, 2018.

· J. Coole and G. Stitt. Overlay architecture for programming FPGAs, 2019. US Patent 10516396.

· J. Coole. Implementing configurable packet parsers for field-programmable gate arrays using hardened resources, 2021. US Patent 11095760.

PROFESSIONAL SERVICE

Technical Program Committee Member, IEEE International Symposium On Field-Programmable Custom Computing Machines (FCCM), 2021-2024