

Archibald Samuel Elliott

BSc (Hons) St And, M.S. Washington

Cambridge, UK

sam@lenary.co.uk

lenary.co.uk

Known as: Sam

Pronouns: He/Him

I am a professional compiler engineer and have worked on programming languages and compilers since 2014.

My primary focus since 2019 has been working with the Clang/LLVM project, where I have worked across many layers of the compiler, including the frontend, the instruction selection systems, backend passes, and improvements to assembly support. My research into solver-aided compilation involved writing compilers for simple domain-specific languages (DSLs) and intermediate representations (IRs) to test out potential optimisation techniques.

Experience

Apr 2023–Present

Staff Software Engineer, *Qualcomm*, Cambridge, UK

I am a Staff Software Engineer in Qualcomm's Compiler Labs. I am working on a project to make it easier for third party developers to extend Qualcomm's proprietary Neural Network compilers and software for the Hexagon platform.

Apr 2023–Apr 2023

Staff Software Engineer, *Arm*, Cambridge, UK

Jan 2021–Mar 2023

Senior Software Engineer, *Arm*, Cambridge, UK

I was a member of the Arm Compiler for Embedded team, which maintains a proprietary LLVM-based toolchain for embedded Arm CPUs, including the official Arm Architecture assembler. I was also involved in the team's work to ensure that upstream LLVM support for the Arm architecture is complete and well-maintained.

- I led the work to add support for the 2022 Arm A-profile Architecture Extensions to our toolchains. This included: analysing new versions of extension specifications for their impact on compilers, linkers, and libraries; providing feedback on the specifications; implementing support for new architectural features; proposing and implementing changes to the relevant software specifications; and coordinating and delivering these changes to internal customers and external partners. This work is important because it happens early enough that it does lead to improvements in the released specifications.
- I participated in discussions about the Arm Architecture ABI, including working in the group defining the ABI for Guarded Control Stacks, a complex new feature added in the 2022 A-profile Extensions.
- I worked on projects to engage with the architecture group earlier, to provide additional feedback for their proposals.
- I was the engineer leading the release process for Arm Compiler for Embedded 6.17 in Autumn 2021, which involved balancing the needs of our users, the wish to ship new features, and a hard release deadline.
- I have taken responsibility for Diversity, Equity, and Inclusion efforts within our team.

Apr 2020–Jan 2021

Lead Software Engineer, *lowRISC*, Cambridge, UK

Jun 2019–Apr 2020

Software Engineer, *lowRISC*, Cambridge, UK

I managed the 3–4-person software team at lowRISC, and led the team of software developers contributing to Google's OpenTitan project. During my time at lowRISC:

- I contributed to LLVM and to OpenTitan's software stack—the latter included designing the initial (ROM) secure boot stage and devising a strategy to write low-level, chip-agnostic software interfaces for the hardware peripherals.
- I helped to define quarterly roadmaps and I planned software deliverables for OpenTitan and lowRISC's LLVM work.
- I contributed and reviewed changes to the RISC-V ELF psABI and Rust's RISC-V targets.
- I transitioned OpenTitan from GCC to a Clang-based RISC-V toolchain.
- I led recruiting for my team and coauthored lowRISC's technical interview system.

Jun 2018–Aug 2018

Research Intern, *NVIDIA*, Redmond, WA, USA

I worked for Vinod Grover on Fireiron, a Halide-like scheduling DSL for dense linear algebra on GPUs.

Jun 2017–Sep 2017

Research Intern, *Microsoft*, Redmond, WA, USA

Jan 2017–Apr 2017

Research Intern, *Microsoft*, Redmond, WA, USA

During two internships, I worked with David Tarditi to design the bounds propagation algorithm and the run-time bounds checking for Checked C, a C language extension that aims to make C safer by adding bounds-checked pointer types.

2011–2015

Freelance Software Developer, UK

2013–2014

Junior Software Developer, *Basho Technologies*, Remote (UK & USA)

Summer 2012

Research Intern, *University of St Andrews*, St Andrews, UK

2010–2011

Software Developer, *55degrees*, Glasgow, UK

Education

- Sep 2015–Dec 2017 **M.S. Computer Science & Engineering**, *University of Washington*, Seattle, WA, USA
I worked in Rastislav Bodik's group (until December 2018) on projects applying program synthesis and solver technologies to optimizing compilers, especially domain-specific compilers for dense linear algebra programs.
Putting the Checks into Checked C, *Master's Project*, Advised by Rastislav Bodik and Dan Grossman
This covered my work designing Checked C, a language extension that adds bounds-checked pointer types to C.
- Sep 2011–Jun 2015 **BSc (Hons) Computer Science**, *University of St Andrews*, St Andrews, UK, First Class Honours
With Year Abroad (2013–2014) at the *University of Virginia*, Charlottesville, VA, USA.
A Concurrency System for Idris & Erlang, *Bachelor's Dissertation*, Advised by Edwin Brady
I explored how Idris, a dependently-typed programming language, can be used to reason about the behaviour of concurrent Erlang programs. I won the *Lockheed Martin Award for Software Engineering* for this work.

Publications

- PACT 2020 **Fireiron: A Scheduling Language for High-Performance Linear Algebra on GPUs**, Bastian Hagedorn, [Archibald Samuel Elliott](#), Henrik Barthels, Rastislav Bodik, and Vinod Grover. In: *Parallel Architectures and Compilation Techniques*. Sep 2020.
- ASPLOS 2019 **Swizzle Inventor: Data Movement Synthesis for GPU Kernels**, Phitchaya Mangpo Phothilimthana, [Archibald Samuel Elliott](#), An Wang, Abhinav Jangda, Bastian Hagedorn, Henrik Barthels, Samuel J. Kaufman, Vinod Grover, Emina Torlak, and Rastislav Bodik. In: *Architectural Support for Programming Languages and Operating Systems*. Apr 2019.
- IEEE SecDev 2018 **Checked C: Making C Safe by Extension**, [Archibald Samuel Elliott](#), Andrew Ruef, Michael Hicks, and David Tarditi. In: *IEEE Cybersecurity Development Conference*. Sep 2018.
- IJPP 42.4 **Cost-Directed Refactoring for Parallel Erlang Programs**, Christopher Brown, Marco Danelutto, Kevin Hammond, Peter Kilpatrick, and [Archibald Elliott](#). In: *International Journal of Parallel Programming* 42.4 (Aug 2014).
- PaPEC 2014 **Riak DT Map: A Composable, Convergent Replicated Dictionary**, Russell Brown, Sean Cribbs, Christopher Meiklejohn, and [Sam Elliott](#). In: *Principles and Practice of Eventual Consistency*. Apr 2014.

Technical Reports

- Synthesizing Number Generators for Stochastic Computing using Mixed Integer Programming**, Vincent T. Lee, [Archibald Samuel Elliott](#), Armin Alaghi, and Luis Ceze. In: *arXiv e-prints*, arXiv:1902.05971 (Feb 2019).
- Checked C TR02 **[Archibald Samuel Elliott](#), Putting the Checks into Checked C**. Checked C Technical Report 2. Paul G. Allen School of Computer Science and Engineering, University of Washington, Oct 2017.

Dissertations

- BSc (Hons) **[Archibald Samuel Elliott](#), A Concurrency System for Idris & Erlang**. Bachelor's Dissertation. School of Computer Science, University of St Andrews, Apr 2015.

Selected Talks

- May 2020 **The RISC-V psABI**, *UW CSE 481A: Operating Systems Capstone*, Seattle, WA, USA
- May 2017 **Putting the Checked into Checked C**, *Microsoft Research*, Redmond, WA, USA
- Oct 2013 **CRDTs: An Update (or maybe just a PUT)**, *RICON West*, San Francisco, CA, USA
- Sep 2012 **Skel: A Streaming Process-based Skeleton Library for Erlang**, *24th Symposium on Implementation and Application of Functional Languages (IFL 2012)*, Oxford, UK

Open Source

I have significant experience working on the Clang and LLVM open source projects, which underlie many commercial compilers. I have experience working on other open source projects, including Rust, Riak, and Google's OpenTitan. I mentored a student during the 2020 Google Summer of Code.

I have profiles on both [GitHub \(lenary\)](#) and LLVM's [Phabricator \(lenary\)](#).