# Build and Run SingleSource Benchmarks Using ClangIR

Google Summer of Code 2023 Proposal.

Takemaru Kadoi(diohabara@gmail.com)

## Overview

- **Project size: Medium**
- **Difficulty: Medium**
- **Confirmed Mentors: @bcardosolopes @lanza**

ClangIR[1] is an intermediate representation that Clang emits from AST visitors. It is a high-level representation that is easier to analyze and optimize than the source code itself. The ClangIR project aims to improve the quality of code generation in Clang by emitting ClangIR instead of LLVM IR, and this project is focused on building and running SingleSource[2] benchmarks using ClangIR.

Clang codegen works by emitting LLVM IR through AST visitors. In the ClangIR project, we emit ClangIR from AST visitors instead, which can then be lowered to

- (a) LLVM IR directly or, alternatively,
- (b) MLIR[3] in-tree dialects.

Lowering is still a quite immature and lacks many instructions, attributes and metadata support. By improving it and pursuing parity with AST→LLVM IR codegen quality, ClangIR can incrementally bridge correctness and performance testing, providing a baseline for future higher level optimizations for C/C++ and/or mixing of MLIR dialects to improve specific workloads. A good starting point is to build and run simple benchmarks, measuring both generated code and build time performance. LLVM's llvm-test-suite contains scripts and machinery that easily allows checking correctness and collecting perf related data and its SingleSource 11 collection provide a set of simpler programs to build.

[1] https://llvm.github.io/clangir/

[2] https://github.com/llvm/llvm-test-suite/tree/main/SingleSource

[3] MLIR stands for Multi-Level Intermediate Representation. It is a compiler infrastructure that provides a common interface for representing and transforming code at different levels of abstraction. https://mlir.llvm.org/docs/LangRef/

## Objectives

The objectives of this project are:

- To build and run programs from the SingleSource subdirectory from the llvm-test-suite using ClangIR
- To collect and present performance and build time data comparing ClangIR to regular (upstream) Clang codegen
- To identify areas for improvement in the ClangIR lowering process and implement improvements as needed

## Roadmap

The project will be divided into the following phases:

### Week 1-2

- Research and familiarize with the ClangIR project and the Clang compiler codebase
- Set up a development environment for the project
- Implement the ClangIR lowering process for SingleSource benchmarks

**Deliverable: ClangIR lowering process implemented and a few SingleSource benchmarks built and run using ClangIR.**

### Week 3-5

- Build and run a larger set of SingleSource benchmarks using ClangIR
- Collect and present performance and build time data comparing ClangIR to regular (upstream) Clang codegen
- Analyze the data and identify areas for improvement in the ClangIR lowering process

**Deliverable: Performance and build time data collected and presented, and areas for improvement in the ClangIR lowering process identified.**

### Week 6-8

- Implement improvements to the ClangIR lowering process as needed
- Re-run benchmarks using the improved ClangIR lowering process
- Collect and present performance and build time data comparing the improved ClangIR lowering process to regular (upstream) Clang codegen

**Deliverable: Improved ClangIR lowering process implemented and performance and build time data collected and presented.**

### Week 9-11

- Implement any remaining improvements to the ClangIR lowering process

- Test the ClangIR lowering process with a variety of SingleSource benchmarks
- Collect and present final performance and build time data comparing ClangIR to regular (upstream) Clang codegen

**Deliverable: Final performance and build time data collected and presented, and final report summarizing the project findings and recommendations for further improvements to the ClangIR lowering process.**

#### Week 12-13

- Refine and improve project documentation, including code comments and user guides
- Prepare the project for submission to Google Summer of Code

**Deliverable: Final code for the project hosted on GitHub, along with all relevant documentation and materials.**

#### Conclusion

- This project will help improve the quality of code generation in Clang by building and running SingleSource benchmarks using ClangIR. The results of this project will help identify areas for improvement in the ClangIR lowering process and provide a baseline for future higher-level optimizations for C/C++ and/or mixing of MLIR dialects to improve specific workloads.

## About Me

#### Personal Information

- Name: Takemaru Kadoi
- Email: diohabara@gmail.com
- LinkedIn: https://www.linkedin.com/in/takemaru-kadoi/
- GitHub: diohabara
- Discourse on LLVM: diohabara
- Discord on LLVM: diohabara
- Residency: CST(UTC - 06:00)
- Availability over the project
  - Up to 40 hours per week during `2023/05/13-2023/08/20`
  - No courses, no internships
- Operating systems and hardware
  - Linux(amd64), macOS(AArch64)

#### Why Me?

I believe that I am a good fit for the Clang-IR project for three main reasons: my technical skills, soft skills, and personal motivations.

Firstly, I have a strong background in compilers[4] and low-level programming[5], which has prepared me well for this project. I have experience working with LLVM IR[6], and I am familiar with the pipeline of language and code generation. Additionally, I have experience with C++ programming, which is essential for working with Clang[gccrs1][gccrs2].

Secondly, I have strong soft skills that will be valuable for this project. I have experience writing technical documents[7], and I am comfortable communicating with others. My experience as an officer of the computer security group at my university is a compelling example[8].

Finally, I am highly motivated to contribute to the Clang-IR project. I have a passion for programming languages and their associated technical stacks, and I believe that this project will be an excellent opportunity for me to apply my skills and knowledge to a widely used and important project. I am committed to dedicating my summer to this project and making a valuable contribution to the Clang-IR project.

Overall, I believe that my technical skills, soft skills, and personal motivations make me a good fit for the Clang-IR project, and I am excited about the opportunity to work on it.

[4] My toy C compiler written in C. https://github.com/diohabara/ccc

[5] Write-ups for CTF challenge hosted by NSA. https://github.com/diohabara/nsa-codebreaker-challenge2022

[6] Repository where I wrote for LLVM https://github.com/diohabara/Kaleidoscope

[7] I have worked at several companies as intern and learned how to write technical documents.

[8] https://www.linkedin.com/company/utdcsg