

## Contents

<b>Compiler Explorer for the EPAC - User Guide</b>	<b>1</b>
Basic usage . . . . .	1
Examining the generated assembly . . . . .	4
Execution . . . . .	4

## Compiler Explorer for the EPAC - User Guide

This is a setup of a Compiler Explorer for an LLVM-based compiler that targets the RISC-V and the V-extension (still draft) architecture.

This tool is intended for examination of the compiler code generation when targeting the V-extension. This gives us valuable information in co-design as it can quickly expose pain points in the code generation. These pain points may suggest changes in the V-extension architecture or require new code generation strategies or optimizations in the compiler.

Compiler Explorer is intended for small programs or snippets and it is not intended for large applications.

Compiler Explorer website is <https://repo.hca.bsc.es/epic/>

### Basic usage

Compiler explorer is mostly a tool to look at the assembler generated by the compiler. The environment is highly configurable by means of windows. This guide describes the basic operation.

By default the interface shows two windows. If you want to reset the environment go to the top of the page and click More button and then Reset Code and UI Layout.

The left window is the **source code window**. It is editable and contains the snippet or small program we want to generate code. The source code window accepts C or C++. This can be changed in a drop-down button found at the top of the source code window.

Testing source code can be done by typing it in the source code window or by loading one of the examples provided. To load an example in the source code window, use the button Save/Load found at the top of the source code window.

The right window is the **assembly output window**. The assembly output window is updated every time the source code window is updated (either by updating the source code or



loading an example). To do so, Compiler Explorer will invoke the compiler. By default no special flags are passed (except the minimal ones that enable the V-extension support) but new flags can be specified in the Compiler options edit box, found at the top of the assembly output window. Because clang defaults to -O0 (no optimization) you will most of the time want to put -O2 here.

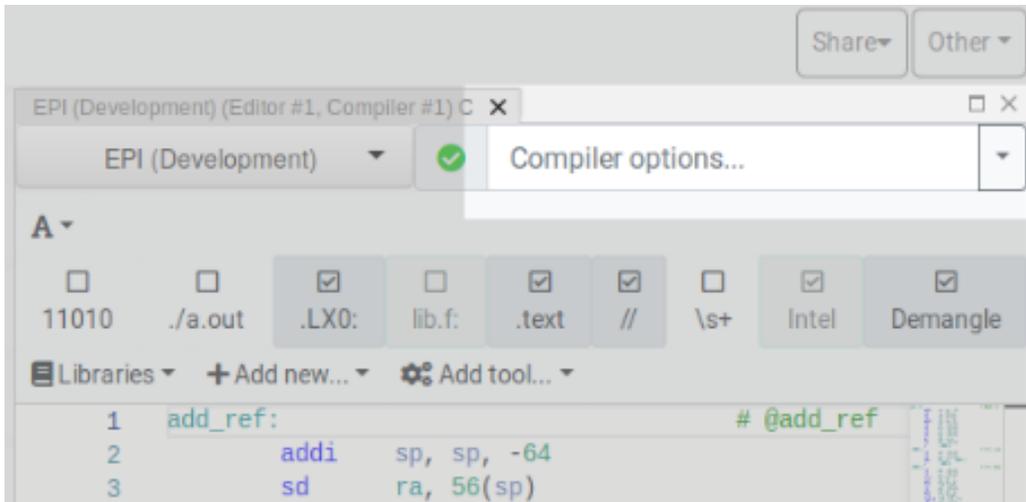


Figure 2: Compiler options

If the compilation fails the assembly output window will only show <Compilation failed>.

It is possible to obtain the output of the compiler to know what went wrong. To do so click on the Output button found at the bottom of the assembly output window.

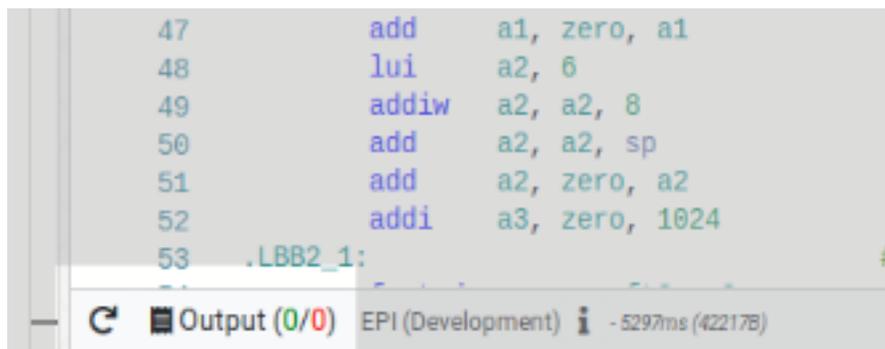


Figure 3: Output button to open output window

This will open a third window: the **command output window**. The command output window shows the output of the compiler, which is usually empty if the compilation succeeds.

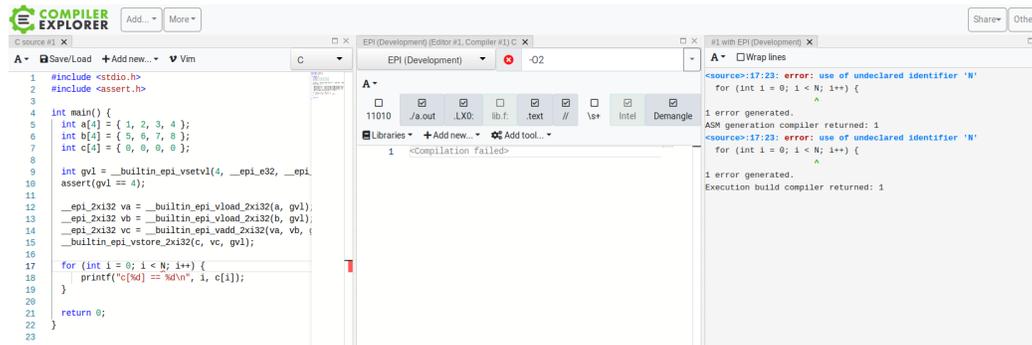


Figure 4: Output window showing an error

## Examining the generated assembly

Compiler Explorer is able to link the source code with the assembly output. This happens when hovering the mouse over the source code window or the assembly output window. Compiler Explorer highlights the corresponding assembly or source code.

Note that, if the corresponding source or assembly is not currently visible nothing will happen. This is because Compiler Explorer is highlighting something that is not in the current view. To address this either scroll to the specific place or, better, right click on the source code window to display a popup menu and then choose **Reveal linked code**. This operation can also be done on the assembly output window but in this case the menu option is called **Scroll to source**.

## Execution

It is possible to execute small programs that use the V-extension instructions directly from the Compiler Explorer page. We do this using the user-space emulator `vehave` which traps the vector instructions and emulates them.

Execution is not enabled by default. To enable it click on the `./a.out` button found at the top of the assembly output window. This will link the program and execute it under `qemu-user` plus `vehave`.

The output of the program is shown in the command output window. If you don't have it open, click on the `Output` button at the bottom of the assembly output window.

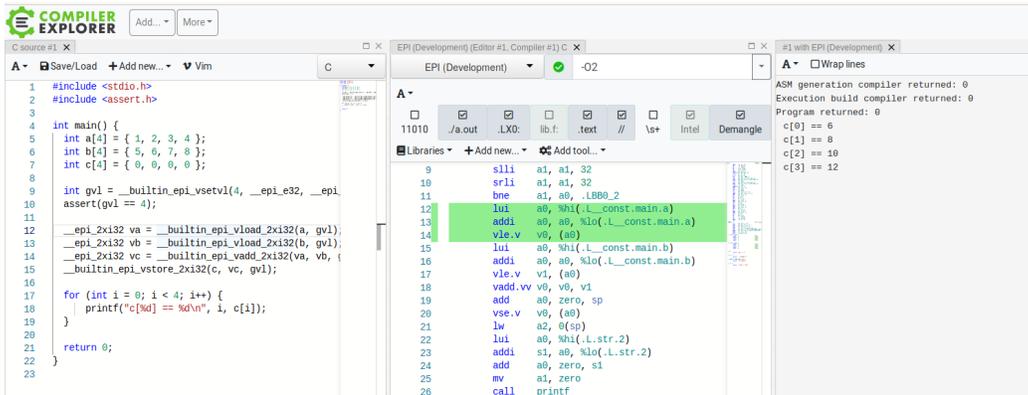


Figure 5: Execution of a simple vector addition