

RISCV 工具链数据报 第三期

本期 RV 数据报，我们将首先对在前两期数据报中热心读者提出的问题进行分析 and 勘误；然后对 SiFive FreedomStudio 和 SiFive prebuilt toolchain、以及平头哥 C910 所提供的 buildroot 所包含的工具链进行简单介绍和评测。

关于 GCC/Clang 的 Os 和 O2/O3 选项的混合使用的效果、数据合理性分析和后续勘误计划

在查阅参考文档[1]，[2]后，我们确认了如下的命令行规则：

GCC: "If you use multiple -O options, with or without level numbers, the last such option is the one that is effective."
Clang: 经过阅读代码和实验结果，多个-O 选项混合使用的效果，跟 GCC 一致。

下表说明了我们在前两期数据报中采用的测试选项和等价的实际效果，以及测试对比结果的有效性和后续勘误计划。

报告期数	选项	实际等价效果	报告数据有效性说明
第一期	gcc -O3 -Os	gcc -Os	报告中“RV32/RV64 codesize 对比”（图一），和“GCC/Clang 的相同选项下 codesize 对比”（图三）依然合理；但图二需要改进，改进原因见表后说明（*）
	clang -O3 -Os	clang -Os	
第二期	gcc -O3 -Os	gcc -Os	由于报告中并未做 O3 和 Os 选项数据之间的对比，因此所报告数据依然是合理的，后续会添加 O2 的数据

* 由于 gcc 和 clang 开启-O3 时，会开启若干可能引起代码膨胀的性能优化选项，而-O2 是只开启那些不引起代码膨胀的优化选项，因此使用-O3 与-Os 来比较 gcc 和 clang 编译器的 codesize 优化能力是不合适的，应该用-O2 和-Os 进行对比，我们将在下一次报告中补充数据，进行此项评测的修正。

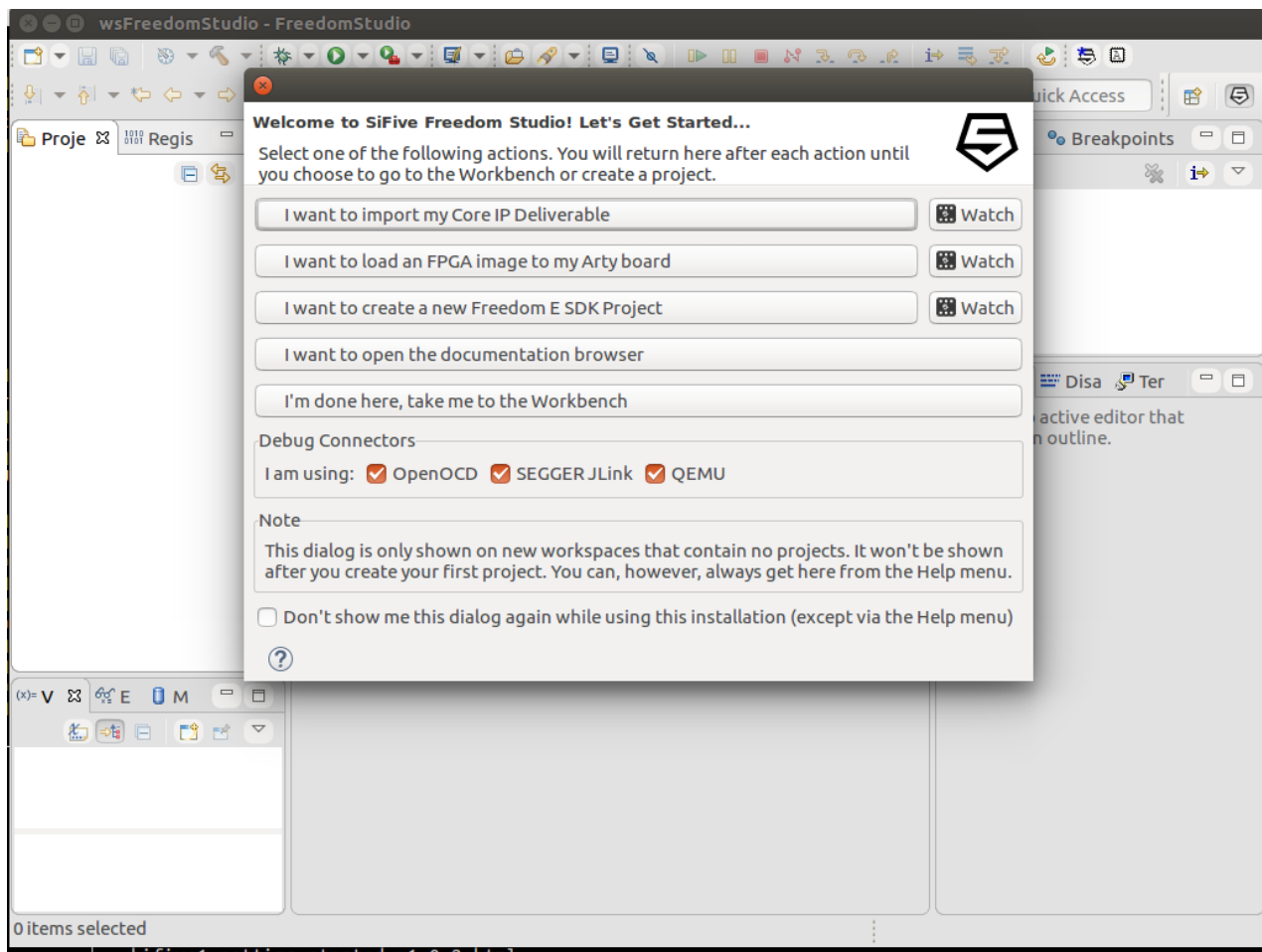
SiFive FreedomStudio 和平头哥 C910 buildroot toolchain 评测

本期我们将对 RISCV 开发者可选的另一个免费 IDE——来自于在 RISCV 产业届具执牛耳地位的 SiFive 公司的 FreedomStudio 进行评测。

我们从 <https://sifive.com/boards/> 页面下载了的 FreedomStudio v2019.08.2 的最新 Linux 版本（该 IDE 支持 Windows/macOS/Linux 三种平台），也下载了 Prebuilt RISC-V GCC toolchain 的 Linux 版本（网站提供 Windows/macOS/CentOS/Ubuntu 四种环境的下载）。

第一步是对 FreedomStudio 进行了安装使用和评测，该安装包 521Mbyte 大小，解压后，直接执行 FreedomStudio 就可以在 Ubuntu16.04 的环境下直接启动基于 Eclipse 扩展的 IDE 环境。

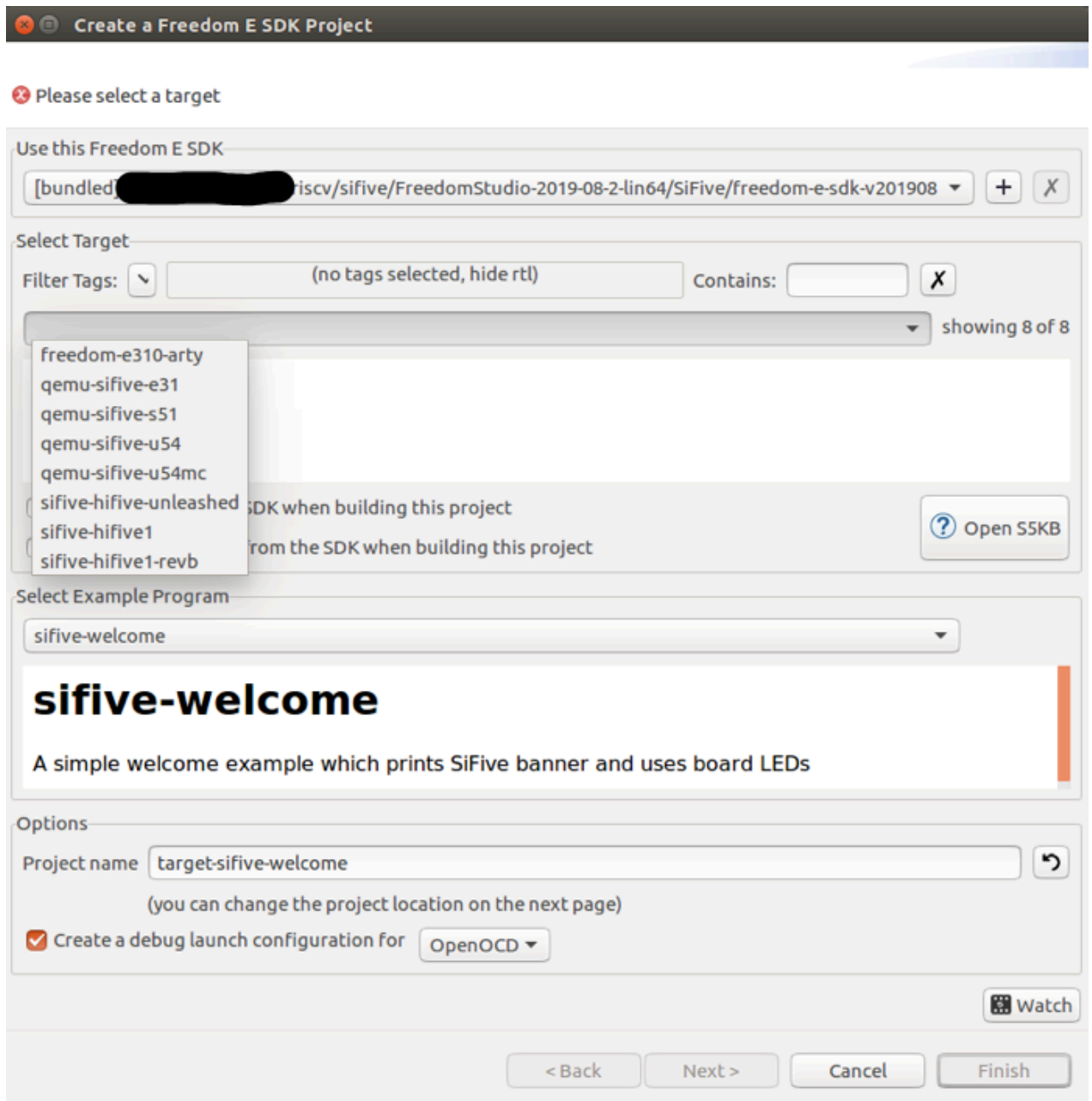
启动时的界面截图如下：



这里可以看到界面提示有 3 类 Project 可以被创建，如果因为没有接触过 SiFive 的开发环境而不了解这些术语，点击右边的 Watch 按钮，就可以直接跳转到浏览器中观看讲解视频。

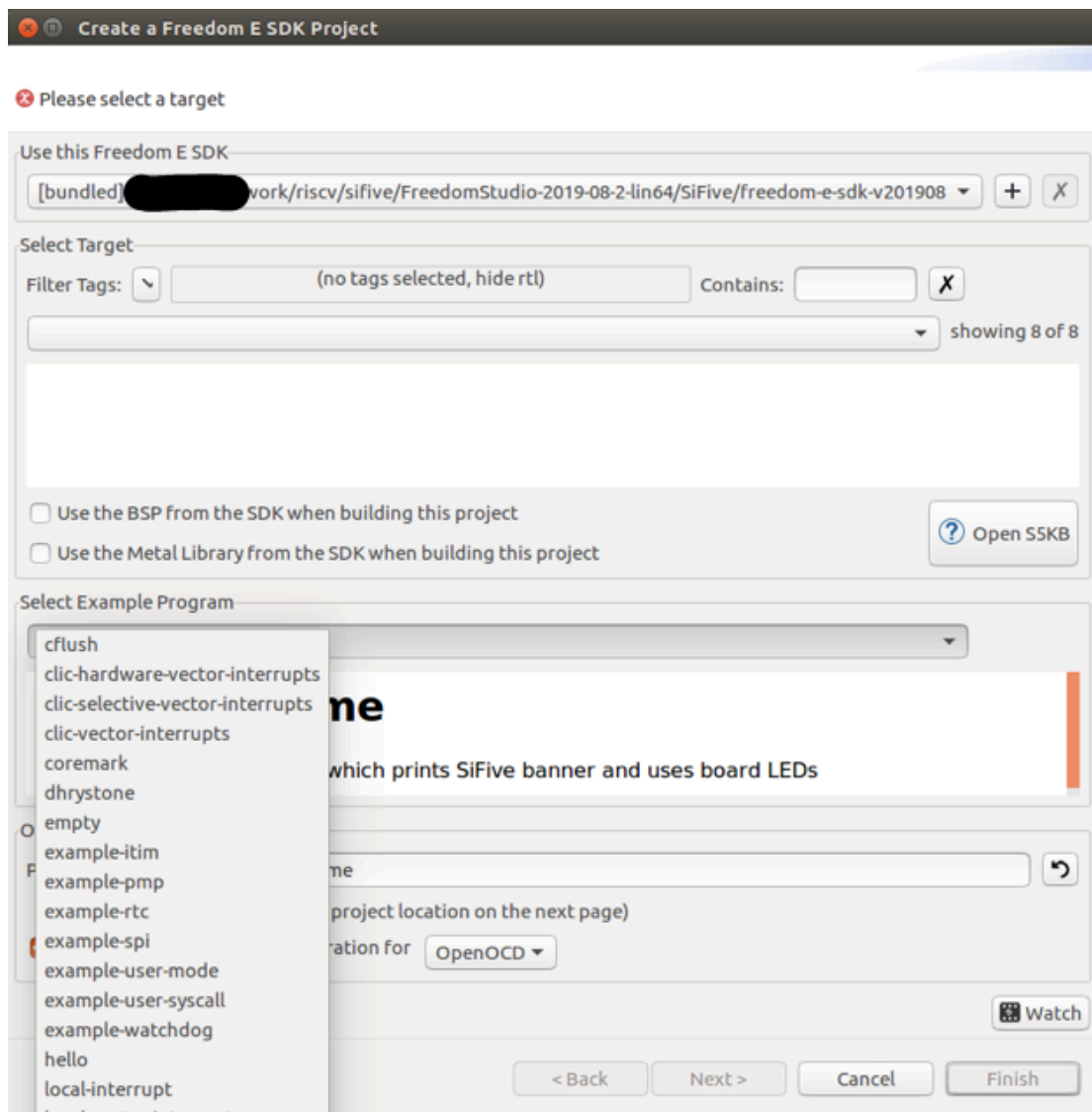
由于我们的评测过程并没有 Core IP，也没有 FPGA 环境，所以我们选择了 Freedom E SDK Project 的项目类别。

在 Select Target 的下拉菜单中，选择 qemu 的 target，这里我们选择了 **qemu-sifive-e31 target**：

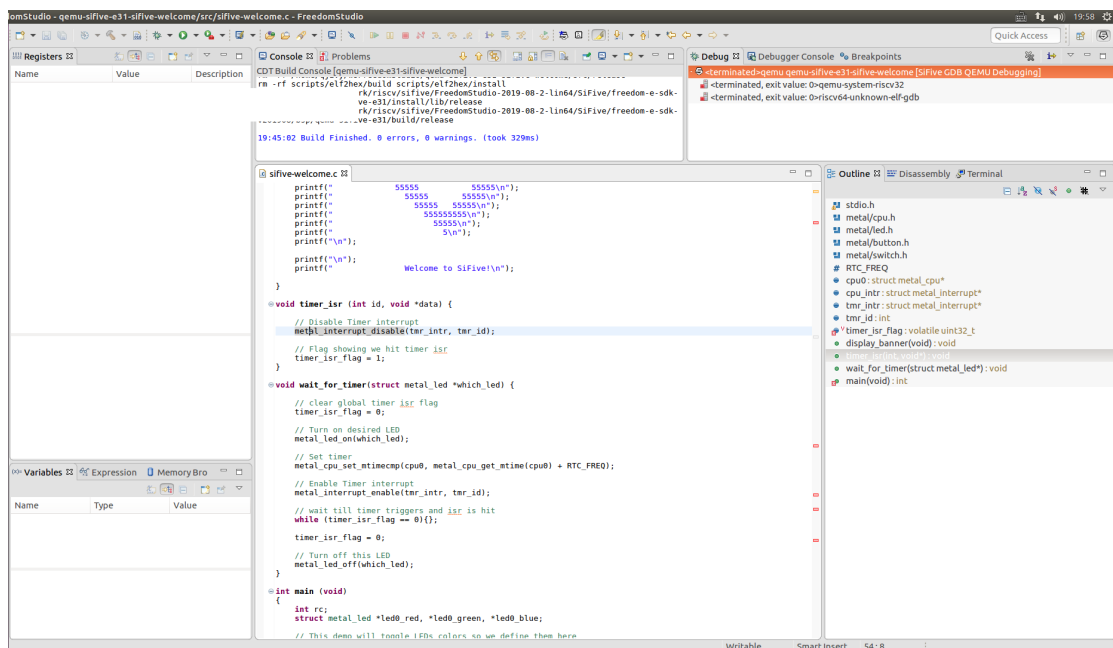


在其下方的 Select Example Program 下拉菜单中，IDE 提供了 25 类该平台上的示例工程，包含的类别很广，从 CPU cache 管理，到外设驱动（SPI/watchdog/RTC），到更为上层的 dhrystone/coremark 等（列表太长项目太多，截图已经无法截全了！）。

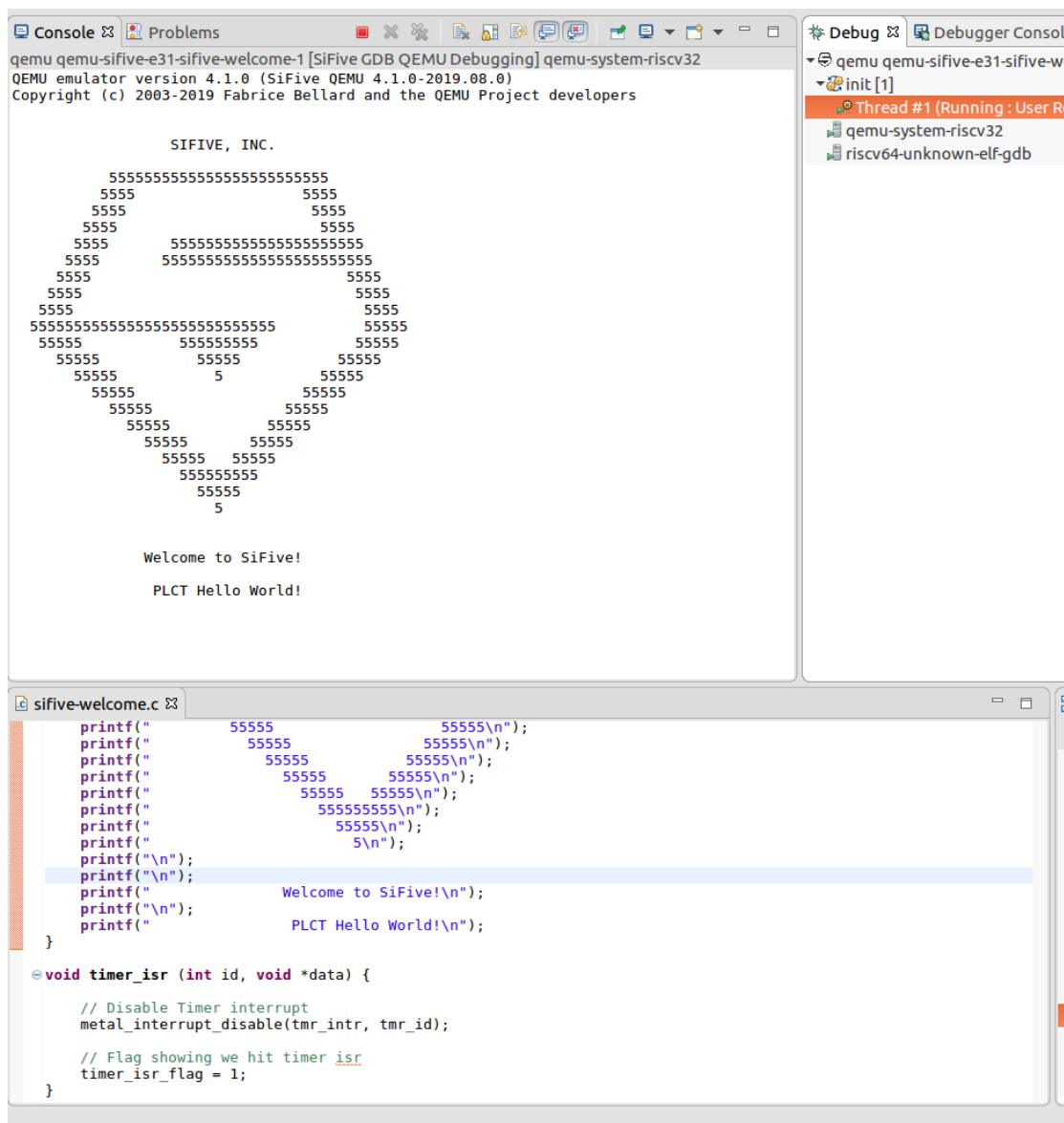
这里我们选择了“sifive-welcome”。



点击确认，IDE 默认建立名为 qemu-sifive-e31-sifive-welcome 的工程后，直接会 build 和 run 一遍这个工程，这意味着所有的工具链、模拟器路径和参数都已经默认设置好了，该界面如图所示：



可以看到这是一个较为经典的 C/C++ Eclipse 开发和 debug 视图，这个 demo 程序的功能是输出 SiFive 的 LOGO。为 coding 的 demo，我们在输出 LOGO 下方做少许改动，再次点击 build 和 run 按钮以后，在 console 中输出如下：



同时，我们也操作了源代码的变量高亮、定义位置提示和文件间跳转、breakpoint、反汇编、查看 registers、memory browser 等 IDE 开发环境中常见和必备的功能，感觉简洁流畅而明确。

至此，第一步评测完成，在网络连接畅通的情况下，SiFive FreedomStudio 可以下载到用户手册、观看范例视频，上手和使用较为便利；但如果使用者场景是只有一个安装包且无法联网，或许可以在安装包顶层目录中，加入 README 简单介绍启动方式，并对 doc 目录下包含用户手册文档和

《custom-core-software-getting-started-guide》文档进行简单提示，这将会使这种场景下的上手更加容易。

接下来的评测将还是针对 toolchain 的模块，评测指标还是 CSiBE codesize (<http://szeged.github.io/csibe/>)。

对于 SiFive FreedomStudio 所包含的 GNU toolchain，其用户手册中已经详细而完备地说明了源代码未经修改地完全来自 github 上开源的 riscv/riscv-gcc repo，目前集成的版本是 gcc: 8.3.0，binutils: 2.32。

另外，本周平头哥在“玄铁 910 交流群”中对高性能 RISCv64 处理器核 C910 开展了的第一场讲座，同时公布了相关开发资源的下载方式，我们从其 buildroot 包 (https://occ.t-head.cn/community/download_detail?id=575996958993285120) 中找到了其发布的交叉工具链 riscv64-buildroot-linux-gnu-gcc。在“buildroot_1.0.7.1/buildroot/thead_910_compat_5.1_glibc_br_defconfig/host/”目录中的 readme 文件描述了其各个 GNU 组件的版本，其中 gcc: 8.3.0, binutils: 2.31.1。

在这里我们将 C910 buildroot toolchain 与SiFive提供的riscv64-unknown-elf-gcc进行了CSiBE的评测，结果如下表。可以看出两者在codesize指标上差别不大。

需要强调的是，一方面，由于这两套工具链源代码都来自于开源社区，一个是 Linux target，另一个是 baremetal target，且代码小版本并不完全一致（或完全明确）；另外一方面，这两套工具链的发布者都没有额外提到过有codesize方面的优化，所以这里的delta值只作为参考，并不能作为工具链质量的判断标准。

编译选项: -Os -march=rv64imafdc	SiFive- 8.3.0	C910- 8.3.0	delta
OpenTCP-1.0.4	1471	1476	-0.4%
teem-1.6.0-src	2584	2605	-0.8%
jikespg-1.3	12525	12540	-0.1%
zlib-1.1.4	2230	2219	0.5%
bzip2-1.0.2	6228	6225	0.0%
cg_compiler_opensrc	5274	5284	-0.2%
mpgcut-1.1	8602	8660	-0.7%
libpng-1.2.5	5143	5170	-0.5%
mpeg2dec-0.3.1	1625	1521	6.8%
flex-2.5.31	9823	9813	0.1%
compiler	4178	4174	0.1%
unrarlib-0.4.0	4779	4794	-0.3%
replaypc-0.4.0.preproc	2053	2053	0.0%

Reference:

- [1] <https://gcc.gnu.org/onlinedocs/gcc-9.2.0/gcc/Optimize-Options.html#Optimize-Options>
- [2] <https://clang.llvm.org/docs/CommandGuide/clang.html>