# 

# RISC-V Rocket-chip模拟器 Gem5模拟器

# 搭建及配置文档

版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 编写人 | 日期 | 备注 |
| 0.1 | 高铮、孙靖渊 | 2017年7月15日 | 建立文档 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

一、rocket-chip模拟器

**1.1 软硬件配置**

硬件基本配置为：

CPU: i7-2640M @ 2.80GHz

Mem: 2G

软件环境为：

OS：Ubuntu 14.04

Linux： 3.19.0-25

**1.2 安装环境说明**

所有源码包放置在/home/gaozheng/riscv目录下，其中包括：

|  |  |
| --- | --- |
| rocket-chip | rocket模拟器 |

所有工具链安装路径为/opt/riscv，其中包括：

|  |  |
| --- | --- |
| riscv | 包含glibc的工具链 |

**1.3 具体内容**

1）下载rocket-chip源码

|  |
| --- |
| $ git clone https://github.com/ucb-bar/rocket-chip.git  $ cd rocket-chip  $ git submodule update --init |

2）安装依赖软件

|  |
| --- |
| $ sudo apt-get install autoconf automake autotools-dev curl device-tree-compiler libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc zlib1g-dev |

3）安装工具链

|  |
| --- |
| $ export RISCV=/opt/riscv/  $ cd rocket-chip/riscv-tools  $ git submodule update --init --recursive  $ export RISCV=/path/to/install/riscv/toolchain  $ export MAKEFLAGS="$MAKEFLAGS -jN" # Assuming you have N cores on your host system  $ ./build.sh |

3）安装C模拟器

|  |
| --- |
| $ cd emulator  $ make |

4）运行模拟器测试

|  |
| --- |
| $ make -jN run-asm-tests  $ make -jN run-bmark-tests |

**1.4工具链及模拟器使用方法**

1）工具链使用

|  |
| --- |
| $ riscv64-unknown-elf-gcc <file> |

部分程序需要特殊的链接方式，才能使用一些特定的系统调用,如printf//TODO

2）模拟器使用

|  |
| --- |
| $ emulator/emulator-rocketchip-DefaultConfig +max-cycles=100000000 +verbose <file> |

输出说明：

文件执行结果在**标准输出**中

模拟器执行trace结果在**标准错误**中

**1.5 使用Dhrystone进行简单测试**

1）使用工具链编译Dhrystone

|  |
| --- |
| $ cd riscv-tools/riscv-tests/benchmarks  $ make |

2）使用模拟器执行

|  |
| --- |
| $ ../../../emulator/emulator-rocketchip-DefaultConfig +max-cycles=100000000 +verbose dhrystone.riscv 2>dhrystone.result |

修改runs数需要修改文件dhrystone.c

默认dhrystone没有debug输出，打印debug信息需要补充debug\_print函数如下：

|  |
| --- |
| void debug\_printf(const char\* str, ...)  {  printf(str);  } |

二、Gem5模拟器

**1.1 软硬件配置**

硬件基本配置为：

CPU: i7-4770HQ @ 2.2GHz

Mem: 4G

软件环境为：

OS：Ubuntu 17.04

Linux： //TODO

**1.2 具体内容**

1）安装依赖软件

|  |
| --- |
| sudo apt-get install autoconf automake autotools-dev curl device-tree-compiler libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc libusb-1.0 scons python-dev zlib1g-dev protobuf-compiler libprotobuf-dev |

2）下载Gem5源码

|  |
| --- |
| git clone https://github.com/riscv/riscv-tools.git  cd riscv-tools  git submodule update --init --recursive |

3）安装工具链

|  |
| --- |
| cd riscv-gnu-toolchain  ./configure --with-arch=rv64g  sudo make |

4）安装Gem5

|  |
| --- |
| git clone https://gem5.googlesource.com/public/gem5  cd gem5  scons build/RISCV/gem5.opt |

**1.3 工具链及模拟器使用方法**

使用工具链编译文件

|  |
| --- |
| $ riscv64-unknown-elf-gcc -march=rv64g <file> |

使用Gem5运行模拟:

|  |
| --- |
| $./build/RISCV/gem5.opt --debug-flags=O3PipeView --debug-file=trace.out ./configs/example/se.py --cpu-type=DerivO3CPU --caches -c <file> |

**1.4使用Dhrystone进行简单测试**

在makefile中修改gcc,使用工具链编译dhrystone

|  |
| --- |
| $ riscv64-unknown-elf-gcc -march=rv64g <file> |

使用Gem5运行dhrystone

|  |
| --- |
| $./build/RISCV/gem5.opt --debug-flags=O3PipeView --debug-file=trace.out ./config/example/se.py --cpu-type=DerivO3CPU --caches -c dhryreg |

运行结果无法计算时间，经测试，时间函数time()存在问题，//TODO