计算机体系结构 lab1

PB20111704 张宇昂

实验过程

1. 安装&编译Gem5, 该过程按照官网上教程安装即可

```
zya1412@ubuntu: ~/ca2023-labs/gem5-stable/gem5-stable
zya1412@ubuntu:~/ca2023-labs/gem5-stable/gem5-stable$ build/X86/gem5.opt
gem5 Simulator System. http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.
gem5 version 21.2.1.0
gem5 compiled Mar 23 2023 23:09:30
gem5 started Mar 24 2023 00:34:27
gem5 executing on ubuntu, pid 13390
command line: build/X86/gem5.opt
Usage
____
 gem5.opt [gem5 options] script.py [script options]
gem5 is copyrighted software; use the --copyright option for details.
Options
____
                        show this help message and exit
 -help, -h
 -build-info, -B
                        Show build information
                        Show full copyright information
 -copyright, -C
 -readme, -R
                        Show the readme
 --outdir=DIR, -d DIR
                        Set the output directory to DIR [Default: m5out]
--redirect-stdout, -r Redirect stdout (& stderr, without -e) to file
```

- 2. 两个章节的实现,这里记录一下相关内容,以便后续实验用到,接下来是一个创建模拟系统的过程 Gem5二进制文件需要一个python脚本作为参数去启动模拟,在脚本中你需要创建一个待模拟的系统(包括各种参数和组成部分),这个脚本完全由用户自己编写
 - 1. 先创建一个System对象,这个System对象包含了很多功能性信息(物理内存空间,特权时钟域等)
 - 2. 设置系统时钟,设置频率域其他参数如电压参数等
 - 3. 用 timing mode 设置模拟系统的内存(如程序所示,内存大小为512M)
 - 4. 为系统创建一个 timing-based CPU ,每个时钟执行除内存访问外的一条指令
 - 5. 创建总线,并将总线与缓存端口相连,在我们的实例中只需要直接将I缓存和D缓存接口直接 连到总线上即可
 - 6. 为了连接系统的所有组成部分,Gem5使用端口抽象,包括**请求端口**和**响应端口**,连接端口保证模拟器功能的正确性,我们连接的特殊端口是用来允许系统向内存中进行读写操作的
 - 7. 创建内存管理器并连接到总线上,这个案例中我们使用基本的DDR3管理器
- 3. 系统创建完成后功能模拟过程如下:
 - 1. 创建一个process,设置好参数(格式类似argv)
 - 2. 启动模拟,创建Root对象

```
zya1412@ubuntu: ~/ca2023-labs/gem5-stable/gem5-s...
                                                     Q
 build/X86/python/m5/main.py(442): main
ya1412@ubuntu:~/ca2023-labs/gem5-stable/gem5-stable$ build/X86/gem5.opt c
onfigs/tutorial/part1/simple.py
gem5 Simulator System. http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.
gem5 version 21.2.1.0
gem5 compiled Mar 24 2023 06:32:22
gem5 started Mar 24 2023 06:39:29
gem5 executing on ubuntu, pid 6832
command line: build/X86/gem5.opt configs/tutorial/part1/simple.py
Global frequency set at 1000000000000 ticks per second
warn: No dot file generated. Please install pydot to generate the dot file
and pdf.
puild/X86/mem/mem interface.cc:791: warn: DRAM device capacity (8192 Mbyte
s) does not match the address range assigned (512 Mbytes)
0: system.remote_gdb: listening for remote gdb on port 7000
Beginning simulation!
puild/X86/sim/simulate.cc:194: info: Entering event queue @ 0. Starting s
imulation...
Hello world!
Exiting @ tick 454646000 because exiting with last active thread context
zya1412@ubuntu:~/ca2023-labs/gem5-stable/gem5-stable$
```

最后simple.py的运行结果如上图所示

Cache SimObject定义及其参数可以在 src/mem/cache/Cache.py 中找到,Cache 是从 BaseCache 中继承得到的(用C++实现的),BaseCache 中的很多参数是没初始化的,所以在调用m5.instantiate() 前需要初始化

我们在 caches.py 中创建缓存并设置特定参数

给configuration脚本加cache的步骤如下:

- 1. 创建 L1Cache 类并设置 BaseCache 的参数,并设置ICache和DCache两个子类,按照相同的方式创建 L2Cache
- 2. 有了设置好参数的Cache,我们只需要将初始化好的子类进行连接,但是连接众多Cache很麻烦, 我们需要设置两个辅助函数 connectCPU 和 connectBus ,并将其加入 L1Cache 类中
- 3. 接下来定义一个独立的 connectCPU 函数供指令和数据Cache使用
- 4. 将 simple.py 用cp命令拷贝到新创建的 two_level.py 中,并导入caches的内容
- 5. 创建L1 Cache并连接到CPU端口
- 6. 由于L2 Cache只能连接一个端口,所以我们不能直接将L1 Cache和L2 Cache相连,而应该用我们 定义的辅助函数进行连接

在Gem5中进行实验时,我们不需要每次都编辑configuration脚本,直接添加command-line参数即可,我们可以用Python的库去支持这种参数添加方式

```
zya1412@ubuntu: ~/ca2023-labs/gem5-stable/gem5-s...
                                                     Q
  build/X86/python/m5/main.py(442): main
zya1412@ubuntu:~/ca2023-labs/gem5-stable/gem5-stable$ build/X86/gem5.opt c
onfigs/tutorial/part1/two_level.py
gem5 Simulator System. http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.
gem5 version 21.2.1.0
gem5 compiled Mar 24 2023 06:32:22
gem5 started Mar 24 2023 08:32:28
gem5 executing on ubuntu, pid 8042
command line: build/X86/gem5.opt configs/tutorial/part1/two_level.py
Global frequency set at 1000000000000 ticks per second
warn: No dot file generated. Please install pydot to generate the dot file
 and pdf.
build/X86/mem/mem_interface.cc:791: warn: DRAM device capacity (8192 Mbyte
s) does not match the address range assigned (512 Mbytes)
0: system.remote gdb: listening for remote gdb on port 7000
Beginning simulation!
build/X86/sim/simulate.cc:194: info: Entering event queue @ 0. Starting s
imulation...
Hello world!
Exiting @ tick 57873000 because exiting with last active thread context
zya1412@ubuntu:~/ca2023-labs/gem5-stable/gem5-stable$
```

见到的报错

- 1. 目录在gem5所有文件的上一层时执行编译:会报一个在 find_flags() 函数中传入了三个 (需要两个)参数 (在stackoverflow上查找的结果是可能存在不同版本的scons)
- 2. 内存没开够(<8G)在编译过程中报错(报错信息: C++: fatal error: κilled signal terminated program cc1plus),不确定具体原因,但是内存开到8G就编译成功了
- 3. ubuntu默认的python是python2,我的解决方案是加一个命令: alias python=python3
- 4. 按照官网上的教程进行实验时, simple.py 的 x86TimeSimulationCPU API不可用,按照助教的实验文档上修改后可以运行