**计算机系统原理实验三**

**Gem5仿真器及Cache行为仿真**

**一 实验目的**

学习仿真器的基本功能，理解cache的工作原理，以及相应参数与算法对程序性能的影响。

**二 实验内容**

1. 通过gem5获得**快速数论变换NTT**程序的内存地址访问序列（trace）。
2. 编写cache模拟器，就是用软件的方法实现cache的工作原理，比如命中检查机制、内存地址和cache块的映射（全相联、组相联、直接映射）的实现和一些替换算法（Random,LRU）等。
3. 自行编写Cache模拟器，以获得的内存地址访问序列文件为输入，统计miss/hit情况。调整相关参数（cache块大小，cache大小）和算法（映射方式，替换算法），查看miss/hit的变化情况，并做出相应解释。在报告里尝试通过实验回答以下问题或者其他你认为有价值的问题:
4. 什么是仿真器，仿真器有什么作用/价值；
5. 如何验证你编写的cache仿真器的正确性（micro-benchmarking）；
6. Cache的大小，组相连等配置对命中率的影响；
7. LRU是否是一个好的cache替换算法；
8. 不同的代码实现与优化方法对cache命中率有怎样的影响；

**三、实验报告要求**

**1. 不超过6人一组完成，提交一份实验报告；**

**2. 提交形式为.zip压缩文件，命名方式为“张三-李四-王五-赵六-实验一报告.pdf”（命名方式不正确会扣分）。压缩包中需中包括修改后的代码文件和实验报告文件，实验报告格式为word或pdf，并在报告开头写上全组成员名字学号。发送到山大网盘：**

<https://icloud.qd.sdu.edu.cn:7777/link/A0E0C09A98A4C61AD3515CC3421C5A15>

**3. 实验报告的得分不取决于长度，在于实验设计的合理性，实验结果的展现效果（图表的设计制作），结论的分析深度，小组人数和工作量的对应关系。**

**4. 提交报告的截止时间为5月9日晚上12点前。晚交的报告将会被扣分。**

**5. 所有雷同的实验报告（包括和之前高年级提交的实验报告雷同），本次实验不得分。同一学生本学期出现2次及以上雷同的实验报告，本学期实验为0分。**

附：

**（实验室的Ubuntu系统上已安装GEM5，可直接使用，也可在自己的电脑上安装）**

GEM5安装指南与使用手册（Linux）

下载最新gem5源码

Github：https://github.com/gem5/gem5

若连不上Github，Gitee：<https://gitee.com/mirrors/gem5source>

解压源码到gem5目录下

cd gem5

sudo apt-get install mercurial scons swig gcc m4 python python-dev libgoogle-perftools-dev g++ libprotobuf-dev

sudo apt-get install build-essential

sudo apt-get install libboost-dev

依赖安装完成之后开始编译gem5源代码

scons build/ARM/gem5.opt

这一步会耗费一定时间（1个小时左右 i5 6400），可以用-j命令开多线程编译

scons build/ARM/gem5.opt -j8

将NTT程序编译为ARM架构的可执行程序ntt

接下来运行gem5

1. 生成dram.out文件

./build/ARM/gem5.opt --outdir=memaccess --debug-flag=DRAM --debug-file=dram.out ./configs/deprecated/example/se.py -c ntt

如使用gem5自带的hello程序

./build/ARM/gem5.opt --outdir=memaccess --debug-flag=DRAM --debug-file=dram.out ./configs/deprecated/example/se.py -c ./tests/test-progs/hello/bin/arm/linux/hello

dram.out文件包含了每条指令的地址及其类型（读或者写）

1. 生成mmu.out文件

./build/ARM/gem5.opt --outdir=memaccess --debug-flag=MMU --debug-file=mmu.out ./configs/deprecated/example/se.py -c ntt

mmu.out是程序加载过程中“gem5简单页表”的加载过程，即trace中的虚拟地址与物理地址的映射关系

1. 生成MemoryAccess.out文件

./build/ARM/gem5.opt --outdir=memaccess --debug-flag=MemoryAccess --debug-file= MemoryAccess.out ./configs/deprecated/example/se.py -c ntt

MemoryAccess.out文件为计算机执行指令的信息，包含了每条指令的操作类型、地址以及操作数据的值