Lab1 实验报告

gem5 简介

gem5是一个模块化的离散事件驱动的计算机模拟器平台。

- gem5 的组件可以很容易地重新排列、参数化、扩展或替换。
- 它将时间的流逝模拟为一系列离散事件。
- 它的预期用途是以各种方式模拟一个或多个计算机系统。
- 它是一个模拟器平台，可以根据需要使用尽可能多的预制组件来构建自己的模拟系统。

gem5的安装和编译

- ubuntu 安装
  - ubuntu 22.04.2-desktop-amd64 。
  - 内存 8G（8192MB）。处理器2CPU。硬盘25GB。
- 安装依赖项
  - 修改虚拟机下载源为ustc镜像

- `sudo apt install build-essential git m4 scons zlib1g zlib1g-dev libprotobuf-dev protobuf-compiler libprotoc-dev libgoogle-perftools-dev python-dev python`

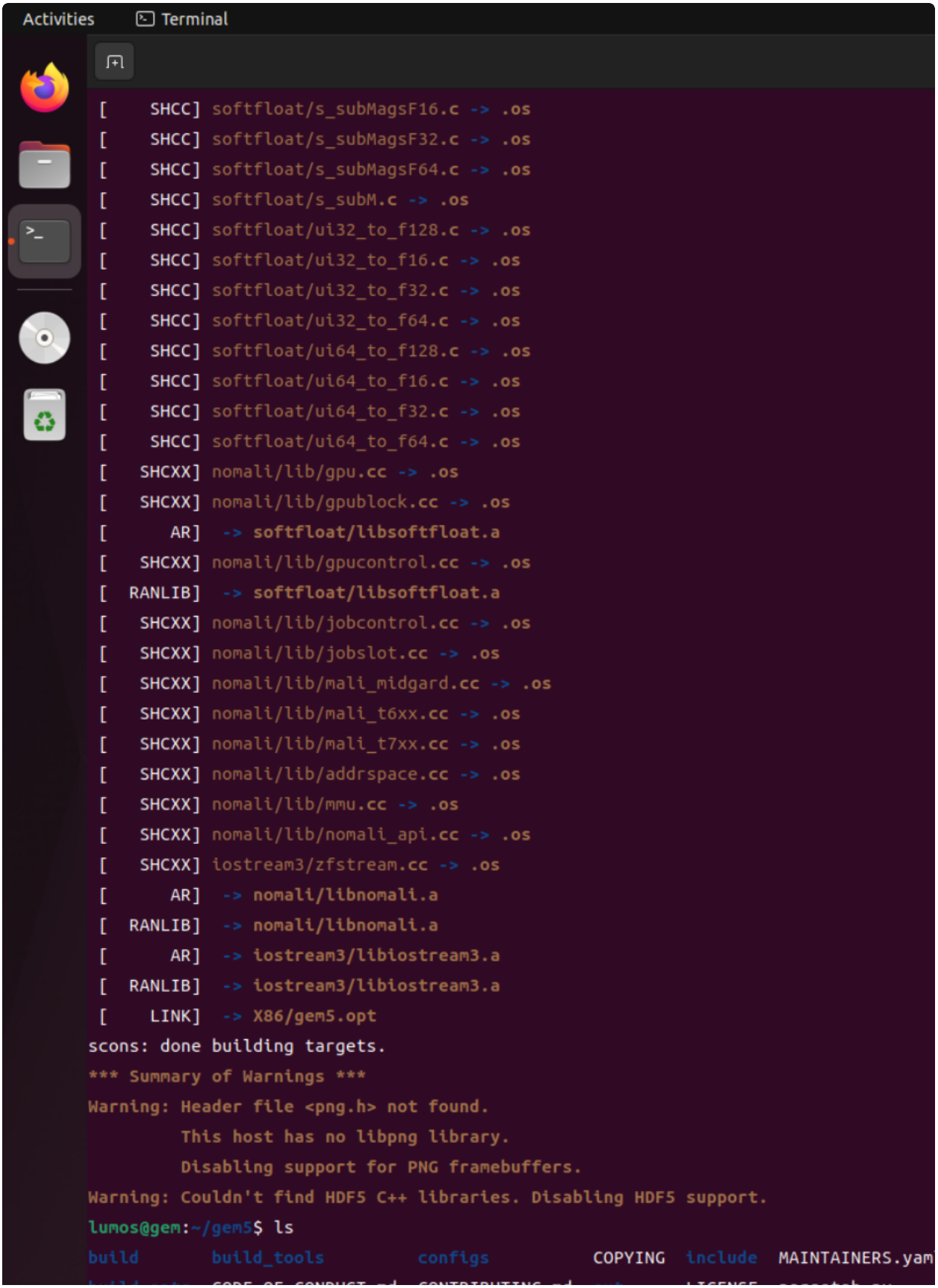
- 获取代码
  - 不要用官方文档中的方法
  - 从 `https://git.lug.ustc.edu.cn/FuWei/ca2023lab/-/tree/main/lab1` 中下载gem5压缩包。解压
  - 也可以从gitee中clone： `git clone https://gitee.com/mirrors/gem5.git`
- 编译
  - 进入 `gem5` 目录。注：下载的安装包解压后目录名为 `gem5-stable` 被我修改为了 `gem5` 。目录结构如下：

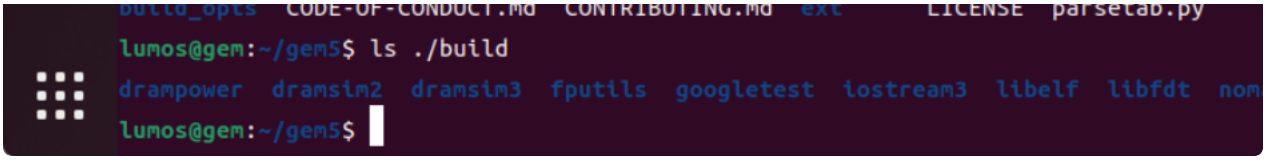
```
lumos@gem:~$ tree -L 2
.
├── Desktop
├── Documents
├── Downloads
├── gem5
│   ├── build
│   ├── build_opts
│   ├── build_tools
│   ├── CODE-OF-CONDUCT.md
│   ├── configs
│   ├── CONTRIBUTING.md
│   ├── COPYING
│   ├── ext
│   ├── include
│   ├── LICENSE
│   ├── MAINTAINERS.yaml
│   ├── README
│   ├── RELEASE-NOTES.md
│   ├── SConstruct
│   ├── site_scons
│   ├── src
│   ├── system
│   ├── TESTING.md
│   ├── tests
│   └── util
├── Music
```

  - `scons build/X86/gem5.opt -j3 CPU_MODELS=AtomicSimpleCPU,TimingSimpleCPU,O3CPU,MinorCPU` 该指令中的 `-j3` 是因为我的虚拟机分配了2个CPU。

编译成功后结果显示如下：

有两个warning，网上查了下。可以用 `sudo apt-get install libhdf5-dev ;sudo apt-get install libpng-dev` 修复。我执行了这两条命令。具体修复了没有，我也不知道。

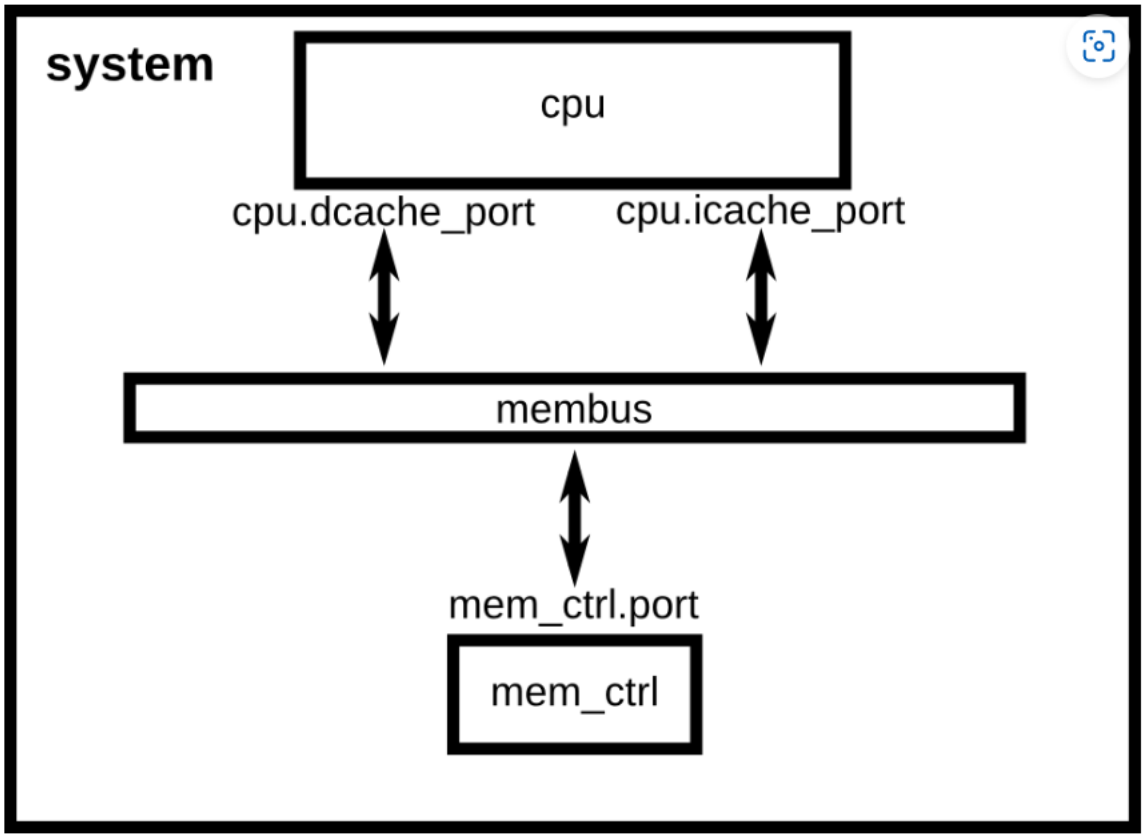




成功编译gem5截图

创建配置脚本

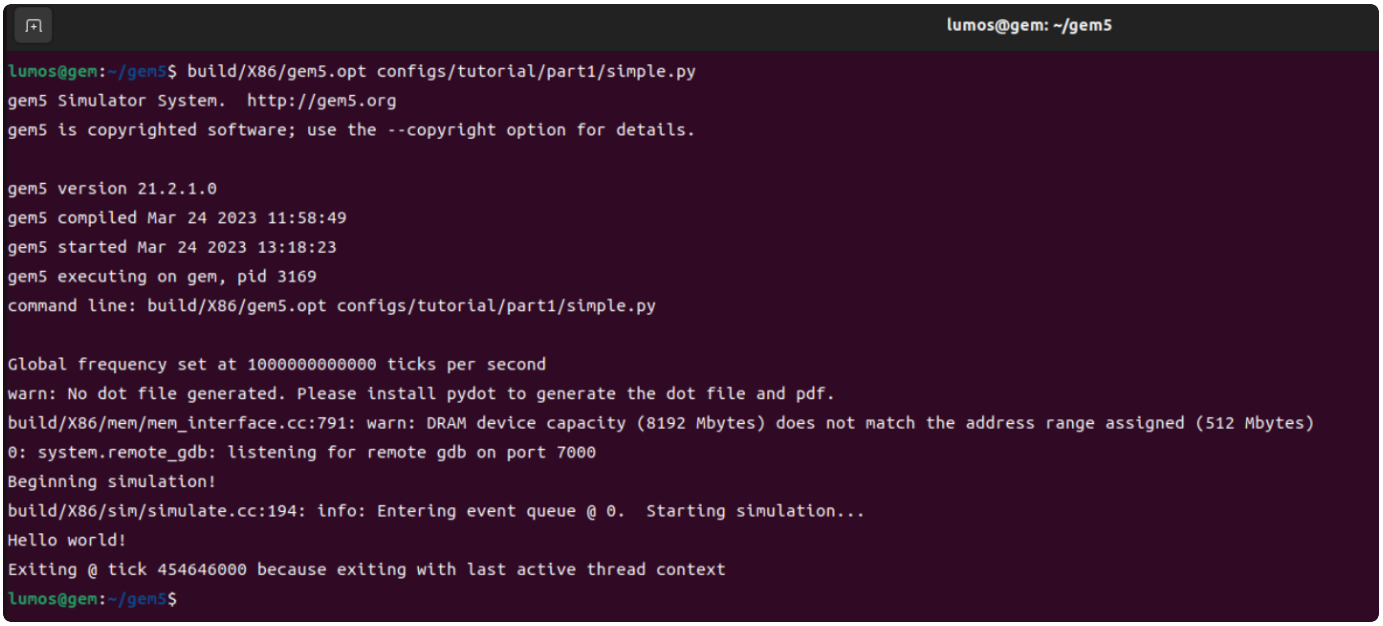
脚本文件及其注释见 `simple.py`



和官方文档的不同处：

`system.cpu = X86TimingSimpleCPU()` ==> `system.cpu = TimingSimpleCPU()`

运行结果如下：

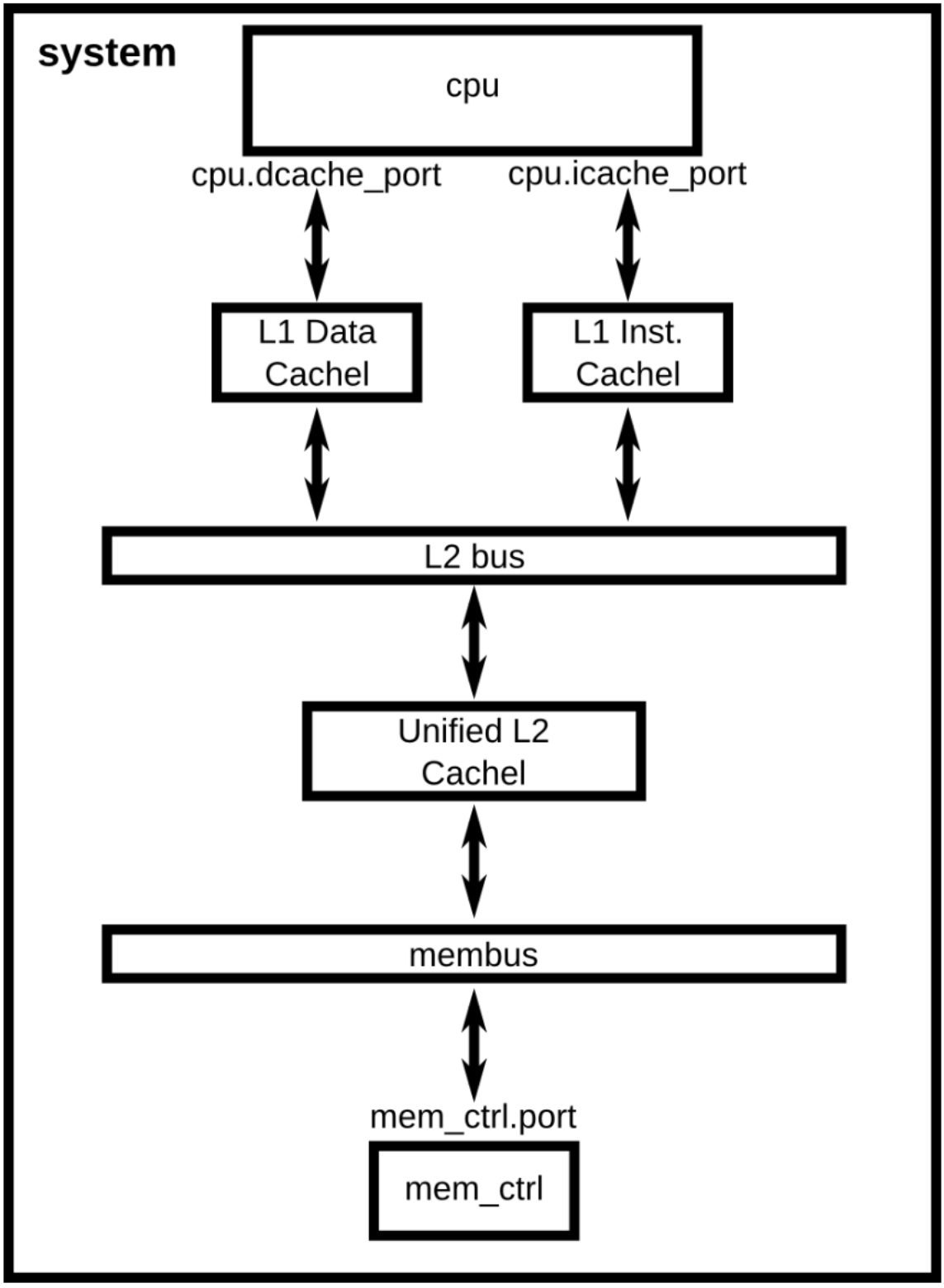


运行simple.py截图

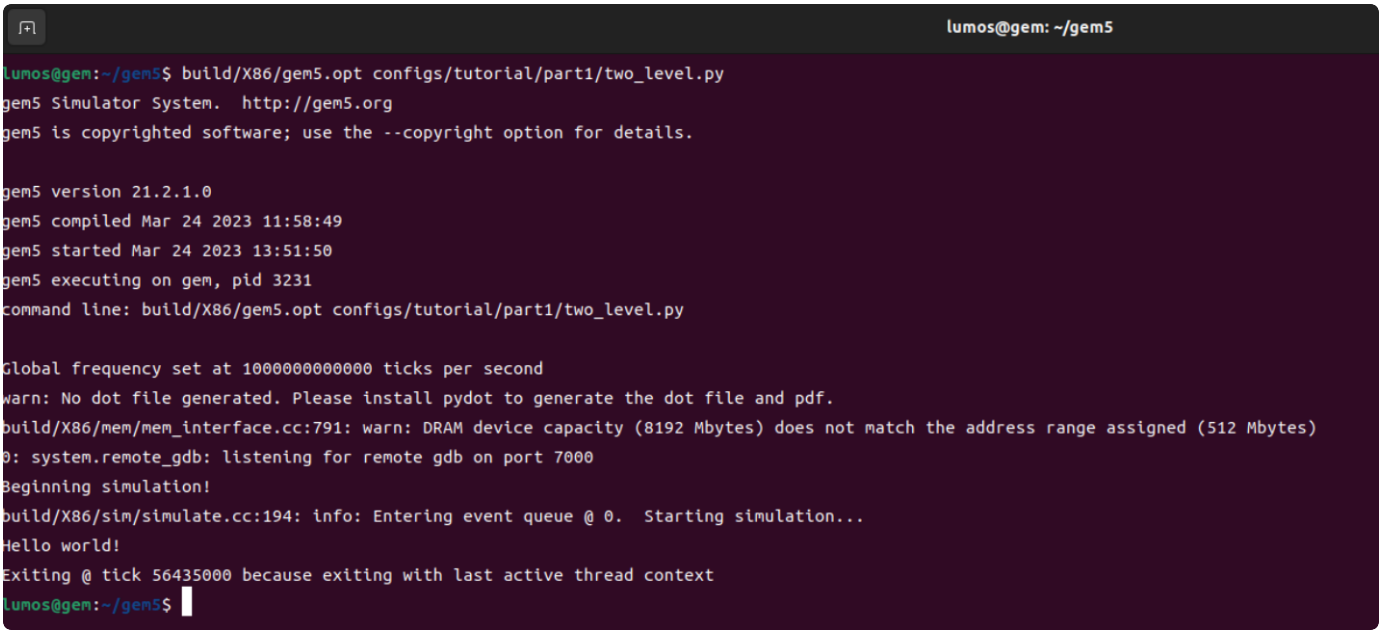
将缓存添加到配置脚本

脚本文件见 `cache.py` 和 `two_level.py`

此时系统模型为：


system
cpu
cpu.dcache_port
cpu.icache_port
L1 Data Cachel
L1 Inst. Cachel
L2 bus
Unified L2 Cachel
membus
mem_ctrl.port
mem_ctrl

运行结果如下：



运行teo_level.py截图

gem5统计信息和输出

运行gem5后，在名为m5out的目录中还生成了三个文件

- `config.ini` 包含模拟创建的每个SimObject的列表及其参数值
- `config.json` 与 `config.ini` 相同，但使用json格式
- `stats.txt` 为模拟注册的所有 gem5 统计信息的文本表示

config.ini

使用 `grep -E "^\\[\w*\\]$|^\\[\w*\\.\w*\\]$" -A 2 config.ini` 命令处理config.ini文件，结果如下，可以从下面的文件中看出很多信息。

比如root类有一个system子类，而系统的部件都是system的子类。这些子类包括 `clk_domain cpu dvfs_handler l2bus l2cache mem_ctrl membus workload` 。这些都是我们在配置文件中定义和使用的。

如果要查看这些子类部件有包括了哪些子类，或者想查看更多详细信息，可以直接查看config.ini文档。或者拓展上面的grep正则筛选更多有用信息。

```
lumos@gem:~/gem5/m5out$ grep -E "^\\[\w*\\]$|^\\[\w*\\.\w*\\]$" -A 2 config.ini
[root]
type=Root
children=system
--
[system]
type=System
children=clk_domain cpu dvfs_handler l2bus l2cache mem_ctrl membus workload
--
[system.clk_domain]
type=SrcClockDomain
children=voltage_domain
--
[system.cpu]
type=TimingSimpleCPU
children=dcache decoder icache interrupts isa mmu power_state tracer workload
--
[system.dvfs_handler]
type=DVFSHandler
domains=
--
[system.l2bus]
type=CoherentXBar
children=power_state snoop_filter
--
[system.l2cache]
type=Cache
children=power_state replacement_policy tags
--
[system.mem_ctrl]
type=MemCtrl
children=dram power_state
--
[system.membus]
type=CoherentXBar
children=power_state snoop_filter
--
[system.workload]
type=X86EmuLinux
eventq_index=0
lumos@gem:~/gem5/m5out$
```

stats.txt

gem5的统计生成文件中将存储所有SimObjects的状态。它包括有关执行的一般统计信息和SimObjects的统计数据。例如CPU统计信息，其中包含有关系统调用数量的信息、缓存系统和翻译缓冲区的统计信息等。文件的最后还有内存控制器统计信息。其中包含每个组件读取的字节数以及这些组件使用的平均带宽等信息。

使用linux命令，精简出stats.txt的部分信息，可以更方便从整体上了解stats.txt所包含的内容。

命令为：

```
cut -d "." -f 1-3 stats.txt|uniq > tmp.txt
```

```
cut -d ":" -f 1 tmp.txt |uniq
```

部分结果如下：

```
1
2   ---------- Begin Simulation Statistics ----------
3   simSeconds                                   0.000056
      # Number of seconds simulated (Second)
4   simTicks                                     56435000
      # Number of ticks simulated (Tick)
5   finalTick                                    56435000
      # Number of ticks from beginning of simulation (restored from checkpoint
    s and never reset) (Tick)
6   simFreq                                1000000000000
       # The number of ticks per simulated second ((Tick/Second))
7   hostSeconds                                     0.01
      # Real time elapsed on the host (Second)
8   hostTickRate                             3946576908
      # The number of ticks simulated per host second (ticks/s) ((Tick/Secon
    d))
9   hostMemory                                    651228
      # Number of bytes of host memory used (Byte)
10  simInsts                                        5701
      # Number of instructions simulated (Count)
11  simOps                                         10302
      # Number of ops (including micro ops) simulated (Count)
12  hostInstRate                                  396095
      # Simulator instruction rate (inst/s) ((Count/Second))
13  hostOpRate                                    715480
      # Simulator op (including micro ops) rate (op/s) ((Count/Second))
14  system.clk_domain.clock                         1000
      # Clock period in ticks (Tick)
15  system.clk_domain.voltage_domain
16  system.cpu.numCycles                           56435
      # Number of cpu cycles simulated (Cycle)
17  system.cpu.numWorkItemsStarted                     0
      # Number of work items this cpu started (Count)
18  system.cpu.numWorkItemsCompleted                   0
      # Number of work items this cpu completed (Count)
19  system.cpu.dcache
20  system.cpu.exec_context
21  system.cpu.icache
22  system.cpu.interrupts
23  system.cpu.mmu
24  system.cpu.power_state
25  system.cpu.thread_0
26  system.cpu.workload
27  system.l2bus.transDist
28  system.l2bus.pktCount_system
```

```
...
system.l2bus.respLayer0
system.l2bus.respLayer1
system.l2bus.snoop_filter
system.l2cache.demandHits
system.l2cache.overallHits
system.l2cache.demandMisses
...
system.l2cache.ReadSharedReq
system.l2cache.power_state
system.l2cache.tags
system.mem_ctrl.avgPriority_cpu
system.mem_ctrl.priorityMinLatency       0
system.mem_ctrl.priorityMaxLatency       0
...
system.mem_ctrl.dram
system.mem_ctrl.power_state
system.membus.transDist
system.membus.pktCount_system
...
system.membus.respLayer0
system.membus.snoop_filter
system.workload.inst

---------- End Simulation Statistics   ----------
```

使用gem5自带的配置文件进行模拟

gem5附带许多配置脚本，所有的配置文件都在config目录中。

主要尝试和学习se.py。这个配置文件在 `configs/example/` 下。

尝试无参数运行 `se.py`

`build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello`

结果显示 `tick=5943000`

```
lumos@gem:~/gem5$ build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello
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 11:58:49
gem5 started Mar 24 2023 15:47:58
gem5 executing on gem, pid 4466
command line: build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello

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 Mbytes) does not match the address range assigned (512 Mbytes)
0: system.remote_gdb: listening for remote gdb on port 7000
**** REAL SIMULATION ****
build/X86/sim/simulate.cc:194: info: Entering event queue @ 0.  Starting simulation...
Hello world!
Exiting @ tick 5943000 because exiting with last active thread context
```

查看config.ini中cpu信息

`grep -E "^\\[system.cpu\\]$" -A 5 "./m5out/config.ini"`

```
lumos@gem:~/gem5$ grep -E "^\\[system.cpu\\]$" -A 5 "./m5out/config.ini"
[system.cpu]
type=AtomicSimpleCPU
children=decoder interrupts isa mmu power_state tracer workload
branchPred=Null
checker=Null
clk_domain=system.cpu_clk_domain
lumos@gem:~/gem5$
```

可以看到，此时配置文件默认使用了原子CPU。

指定CPU类型和L1缓存大小

1. 运行

`build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello --cpu-type=TimingSimpleCPU --l1d_size=64kB --l1i_size=16kB`

2. 结果

```
lumos@gem:~/gem5$ build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello --cpu-type=TimingSimpl
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 11:58:49
gem5 started Mar 24 2023 15:52:28
gem5 executing on gem, pid 4471
command line: build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello --cpu-type=TimingSimpleCPU

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 Mbytes) does not match the address range assigned (512 Mbytes
0: system.remote_gdb: listening for remote gdb on port 7000
**** REAL SIMULATION ****
build/X86/sim/simulate.cc:194: info: Entering event queue @ 0.  Starting simulation...
Hello world!
Exiting @ tick 454646000 because exiting with last active thread context
lumos@gem:~/gem5$
```

结果显示 `tick=454646000`

结果与预期似乎不符。

3. 分析

查看下CPU相关信息

`grep -E "^\\[system.cpu\\]$" -A 5 "./m5out/config.ini"`

```
lumos@gem:~/gem5$ grep -E "^\\[system.cpu\\]$" -A 5 "./m5out/config.ini"
[system.cpu]
type=TimingSimpleCPU
children=decoder interrupts isa mmu power_state tracer workload
branchPred=Null
checker=Null
clk_domain=system.cpu_clk_domain
```

CPU设置成功了。那可能是缓存的原因。查找config.ini中缓存相关信息

`grep -E "cache" -A 5 "./m5out/config.ini"`

```
lumos@gem:~/gem5$ grep -E "cache" "./m5out/config.ini"
cache_line_size=64
dcache_port=system.membus.cpu_side_ports[2]
icache_port=system.membus.cpu_side_ports[1]
cpu_side_ports=system.system_port system.cpu.icache_port system.cpu.dcache_port system.cpu.mmu.itb.walker.port sys
lumos@gem:~/gem5$
```

`cache` 的搜索结果居然只有这么一点，显然，我们并没有成功使用缓存。

事实上，上面的命令设置了缓存的相关信息，但并没有真正的使用缓存。

指定CPU类型并使用缓存

1. 运行

`build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello --cpu-type=TimingSimpleCPU --l1d_size=64kB --l1i_size=16kB --caches`

2. 结果

```
lumos@gem:~/gem5$ build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello --cpu-type=TimingSimpleCPU
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 11:58:49
gem5 started Mar 24 2023 16:00:26
gem5 executing on gem, pid 4483
command line: build/X86/gem5.opt configs/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello --cpu-type=TimingSimpleCPU --l1

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 Mbytes) does not match the address range assigned (512 Mbytes)
0: system.remote_gdb: listening for remote gdb on port 7000
**** REAL SIMULATION ****
build/X86/sim/simulate.cc:194: info: Entering event queue @ 0.  Starting simulation...
Hello world!
Exiting @ tick 31680000 because exiting with last active thread context
lumos@gem:~/gem5$
```

此时 `tick=3168000` 。这次设置正确了。

3. 分析

查看下CPU和Cache

```
lumos@gem:~/gem5$ grep -E "^\\[system.cpu\\]$" -A 5 "./m5out/config.ini"
[system.cpu]
type=TimingSimpleCPU
children=dcache decoder dtb_walker_cache icache interrupts isa itb_walker_cache mmu power_state tracer workload
branchPred=Null
checker=Null
clk_domain=system.cpu_clk_domain
lumos@gem:~/gem5$
```

```
lumos@gem:~/gem5$ grep -E "^\\[system\\.cpu\\.(dcache|icache)\\]$" -A 3 "./m5out/config.ini"
[system.cpu.dcache]
type=Cache
children=power_state replacement_policy tags
addr_ranges=0:18446744073709551615
--
[system.cpu.icache]
type=Cache
children=power_state replacement_policy tags
addr_ranges=0:18446744073709551615
lumos@gem:~/gem5$ grep -E "^\\[system\\.cpu\\.(dcache|icache)\\]$" -A 3 "./m5out/config.ini"
[system.cpu.dcache]
type=Cache
children=power_state replacement_policy tags
addr_ranges=0:18446744073709551615
--
[system.cpu.icache]
type=Cache
children=power_state replacement_policy tags
addr_ranges=0:18446744073709551615
lumos@gem:~/gem5$
```

重要选项

- `--help` 获取帮助
- `--cpu-type` CPU类型，默认为原子
- `--sys-clock` 系统时钟
- `--mem-type` 内存类型

从help文档中筛选出上述三个参数的可选参数如下：

```
lumos@gem:~/gem5$ build/X86/gem5.opt configs/example/se.py --help|grep -E "(\\[\\-\\-cpu-type)|(\\[\\-\\-sys-clock)|(\\[\\-\\-mem-type)"
            [--sys-clock SYS_CLOCK] [--list-mem-types]
            [--mem-type {CfiMemory,DDR3_1600_8x8,DDR3_2133_8x8,DDR4_2400_16x4,DDR4_2400_4x16,DDR4_2400_8x8,DRAMInterface,GDDR5_4000_2x32,HBM_
1000_4H_1x128,HBM_1000_4H_1x64,HMC_2500_1x32,LPDDR2_S4_1066_1x32,LPDDR3_1600_1x32,LPDDR5_5500_1x16_8B_BL32,LPDDR5_5500_1x16_BG_BL16,LPDDR5_550
0_1x16_BG_BL32,LPDDR5_6400_1x16_8B_BL32,LPDDR5_6400_1x16_BG_BL16,LPDDR5_6400_1x16_BG_BL32,NVMInterface,NVM_2400_1x64,QoSMemSinkInterface,Simpl
eMemory,WideIO_200_1x128}]
            [--cpu-type {AtomicSimpleCPU,DerivO3CPU,MinorCPU,NonCachingSimpleCPU,O3CPU,TimingSimpleCPU,TraceCPU,X86KvmCPU}]
lumos@gem:~/gem5$
```

- `--caches` 使用经典缓存执行模拟
- `-l2cache` 如果使用经典缓存，则使用L2缓存执行模拟
- `--ruby` 使用Ruby替代经典缓存作为缓存系统模拟
- `-m TICKS, --abs-max-tick=TICKS`
    - 运行到指定的绝对模拟滴答声，包括来自已恢复检查点的滴答声。如果只想模拟一定量的模拟时间，这将很有用。
- `-I MAXINSTS, --maxinsts=MAXINSTS`

- 要模拟的指令总数（默认值：永远运行）。如果想在执行了一定数量的指令后停止仿真，这很有用。