<http://blog.csdn.net/fandroid/article/details/45786885>

# 浏览了一个网站 关于[gem5三级缓存配置，其中L1和L2为cpu独享](http://blog.csdn.net/fandroid/article/details/45786885) 。发现与BATMAN那篇论文有相似之处，我认为可以借鉴。

缓存架构参数配置：

cpu: four core

private L1 dcache: 32KB/8-way; icache: 32KB/8-way

private L2 cache: 256KB/8-way

shared L3 cahce: 4MB/16-way

实现方法：

1.实现三级缓存配置涉及三个文件：configs/common/Caches.py,CacheConfig.py和configs/example/fs.py

2.通常实现自己的功能不直接在源代码上修改，而是生成一份新的副本，如：l3Caches.py, l3CacheConfig.py和l3fs.py

3. 分别修改上述文件

l3Caches.py中增加:

class L3Cache(BaseCache):

assoc = 16

block\_size = 64

hit\_latency = 20

response\_latency = 20

mshrs = 512

tgts\_per\_mshr = 20

write\_buffers = 256

l3fs.py 中对Caches和CacheConfig模块的引用改为l3Caches和l3CacheConfig

l3CacheConfig.py文件的修改如下：

def config\_cache(options, system):

if options.cpu\_type == "arm\_detailed":

try:

from O3\_ARM\_v7a import \*

except:

print "Did you compile the O3 model?"

sys.exit(1)

dcache\_class, icache\_class, l2\_cache\_class, l3\_cache\_class = \

O3\_ARM\_v7a\_DCache, O3\_ARM\_v7a\_ICache, O3\_ARM\_v7aL2, O3\_ARM\_v7aL3

else:

dcache\_class, icache\_class, l2\_cache\_class, l3\_cache\_class = \

L1Cache, L1Cache, L2Cache, L3Cache

# Set the cache line size of the system

system.cache\_line\_size = options.cacheline\_size

# set the shared l3 cache

#配置三级共享换成，设置方法和源代码中二级缓存设置一样，只要更改bus连接方式

if options.l3cache:

system.l3=l3\_cache\_class(clk\_domain=system.cpu\_clk\_domain,

size=options.l3\_size,

assoc=options.l3\_assoc)

system.tol3bus=CoherentBus(clk\_domain=

system.cpu\_clk\_domain,

width = 32)

system.l3.cpu\_side = system.tol3bus.master

system.l3.mem\_side = system.membus.slave

for i in xrange(options.num\_cpus):

if options.caches:

icache = icache\_class(size=options.l1i\_size,

assoc=options.l1i\_assoc)

dcache = dcache\_class(size=options.l1d\_size,

assoc=options.l1d\_assoc)

if buildEnv['TARGET\_ISA'] == 'x86':

system.cpu[i].addPrivateSplitL1Caches(icache, dcache,

PageTableWalkerCache(),

PageTableWalkerCache())

else:

system.cpu[i].addPrivateSplitL1Caches(icache, dcache)

system.cpu[i].createInterruptController()

#配置二级私有缓存

if options.l2cache:

system.cpu[i].l2=l2\_cache\_class(clk\_domain=system.cpu\_clk\_domain,

size=options.l2\_size,

assoc=options.l2\_assoc)

system.cpu[i].tol2bus = CoherentBus()

system.cpu[i].l2.cpu\_side = system.cpu[i].tol2bus.master

system.cpu[i].l2.mem\_side = system.tol3bus.slave

if options.l3cache:

#连接所有端口，主要是cpu上缓存端口和主存端口的连接，要理解该函数可以查看./src/cpu/BaseCPU.py源代码中对于connectAllPorts的定义

system.cpu[i].connectAllPorts(system.cpu[i].tol2bus, system.membus)

else:

if options.l2cache:

system.cpu[i].connectAllPorts(system.tol2bus, system.membus)

else:

system.cpu[i].connectAllPorts(system.membus)

return system

4.配置好后用脚本命令执行

build/ALPHA/gem5.debug configs/example/l3fs.py --cpu-clock=3GHz --kernel=/dawnfs/users/me/dist/m5/system/binaries/vmlinux\_2.6.27-gcc\_4.3.4 -n 4 --script=benchmark/blackscholes\_16c\_simsmall\_ckpts.rcS --cpu-type=detailed --caches --l1d\_size=32kB --l1d\_assoc=8 --l1i\_size=32kB --l1i\_assoc=8 --l2cache --l2\_size=256kB --l2\_assoc=8 --l3cache --l3\_size=4MB --l3\_assoc=16

5.运行成功，可以在config.ini和config.dot中查看配置及体系架构信息。