計組 期末 project demo

Q0(40%):

編譯環境

Q1(15%):

enable L3 跑 hello world 修改 option, cache, cache_config, xBar.py

- Cache:

```
class L3Cache(Cache):
    assoc = 8
    tag_latency = 20
    data_latency = 20
    response_latency = 20
    mshrs = 20
    tgts_per_mshr = 12
    write_buffers = 8
```

- Cache Config:

```
    dcache_class, icache_class, l2_cache_class, walk_cache_class, l3_cache_class = \
        O3_ARM_v7a_DCache, O3_ARM_v7a_ICache, O3_ARM_v7aL2, \
        O3_ARM_v7aWalkCache, O3_ARM_v7aL3
else:
    dcache_class, icache_class, l2_cache_class, walk_cache_class, l3_cache_class = \
        L1_DCache, L1_ICache, L2Cache,None, L3Cache

    if buildEnv['TARGET_ISA'] == 'x86':
        walk_cache_class = PageTableWalkerCache
```

```
if options.l2cache and  options.l3cache:
    system.l2 = l2_cache_class(clk_domain=system.cpu_clk_domain,
                               size=options.l2_size,
                               assoc=options.l2_assoc)
    system.l3 = l3_cache_class(clk_domain=system.cpu_clk_domain,
                               size=options.l3_size,
                               assoc=options.l3_assoc)

    system.tol2bus = L2XBar(clk_domain = system.cpu_clk_domain)
    system.tol3bus = L3XBar(clk_domain = system.cpu_clk_domain)

    system.l2.cpu_side = system.tol2bus.master
    system.l2.mem_side = system.tol3bus.slave

    system.l3.cpu_side = system.tol3bus.master
    system.l3.mem_side = system.membus.slave
elif options.l2cache:
    # Provide a clock for the L2 and the L1-to-L2 bus here as they
    # are not connected using addTwoLevelCacheHierarchy. Use the
    # same clock as the CPUs.
    system.l2 = l2_cache_class(clk_domain=system.cpu_clk_domain,
                               size=options.l2_size,
                               assoc=options.l2_assoc)

    system.tol2bus = L2XBar(clk_domain = system.cpu_clk_domain)
    system.l2.cpu_side = system.tol2bus.master
    system.l2.mem_side = system.membus.slave
```

BaseCPU:

```
from XBar import L3XBar

import sys
```

```
def addThreeLevelCacheHierarchy(self, ic, dc, l3c, iwc = None, dwc = None):
    self.addPrivateSplitL1Caches(ic, dc, iwc, dwc)
    self.toL3Bus = L3XBar()
    self.connectCachedPorts(self.toL3Bus)
    self.l3cache = l3c
    self.toL2Bus.master = self.l3cache.cpu_side
    self._cached_ports = ['l3cache.mem_side']
```

XBar:

```
class L3XBar(CoherentXBar):
    # 256-bit crossbar by default
    width = 32

    # Assume that most of this is covered by the cache latencies, with
    # no more than a single pipeline stage for any packet.
    frontend_latency = 1
    forward_latency = 0
    response_latency = 1
    snoop_response_latency = 1

    # Use a snoop-filter by default, and set the latency to zero as
    # the lookup is assumed to overlap with the frontend latency of
    # the crossbar
    snoop_filter = SnoopFilter(lookup_latency = 0)

    # This specialisation of the coherent crossbar is to be considered
    # the point of unification, it connects the dcache and the icache
    # to the first level of unified cache.
    point_of_unification = True
```

Option.py

```
parser.add_option("--l3cache", action="store_true")
```

Q2(15%):

full:

./build/X86/gem5.opt configs/example/se.py -c ./quicksort --cpu-type=TimingSimpleCPU --caches --l2cache --l3cache --l3_assoc=1 --l1i_size=32kB --l1d_size=32kB --l2_size=128kB --l3_size=1MB --mem-type=NVMainMemory --nvmain-config=../NVmain/Config/PCM_ISSCC_2012_4GB.config

2 way

./build/X86/gem5.opt configs/example/se.py -c ./quicksort --cpu-type=TimingSimpleCPU --caches --l2cache --l3cache --l3_assoc=2 --l1i_size=32kB --l1d_size=32kB --l2_size=128kB --l3_size=1MB --mem-type=NVMainMemory --nvmain-config=../NVmain/Config/PCM_ISSCC_2012_4GB.config

去 stats.txt 看 L3 overall cache miss rate

去 terminal 看 active_energy

都要截圖。(共 4 張)

Two:

```
system.l3.ReadSharedReq_miss_rate::cpu.data     0.514880                       # miss rate for ReadSha
system.l3.ReadSharedReq_miss_rate::total     0.515260                       # miss rate for ReadShared
system.l3.demand_miss_rate::cpu.inst              1                       # miss rate for demand acc
system.l3.demand_miss_rate::cpu.data         0.572914                       # miss rate for demand acc
system.l3.demand_miss_rate::total            0.573207                       # miss rate for demand acc
system.l3.overall_miss_rate::cpu.inst             1                       # miss rate for overall ac
system.l3.overall_miss_rate::cpu.data        0.572914                       # miss rate for overall ac
```

```
i0.defaultMemory.channel0.FRFCFS-WQF.channel0.rank0.bank3.subarray0.subArrayEner
gy 2.65416e+06nJ
i0.defaultMemory.channel0.FRFCFS-WQF.channel0.rank0.bank3.subarray0.activeEnergy
 2436.41nJ
i0.defaultMemory.channel0.FRFCFS-WQF.channel0.rank0.bank3.subarray0.burstEnergy
120715nJ
```

Full :

```
system.l3.demand_miss_rate::cpu.inst                1
system.l3.demand_miss_rate::cpu.data         0.519218
system.l3.demand_miss_rate::total            0.519548
system.l3.overall_miss_rate::cpu.inst               1
system.l3.overall_miss_rate::cpu.data        0.519218
system.l3.overall_miss_rate::total           0.519548
system.l3.ReadExReq_avg_miss_latency::cpu.data 289832.125854
system.l3.ReadExReq_avg_miss_latency::total 289832.125854
system.l3.ReadSharedReq_avg_miss_latency::cpu.inst 154757.022472
system.l3.ReadSharedReq_avg_miss_latency::cpu.data 409457.429476
```

```
o {}
i0.defaultMemory.channel0.FRFCFS-WQF.channel0.rank0.bank3.bankEnergy 2.27436e+06
nJ
i0.defaultMemory.channel0.FRFCFS-WQF.channel0.rank0.bank3.activeEnergy 2014.9nJ
i0.defaultMemory.channel0.FRFCFS-WQF.channel0.rank0.bank3.burstEnergy 103544nJ
i0.defaultMemory.channel0.FRFCFS-WQF.channel0.rank0.bank3.refreshEnergy 0nJ
i0.defaultMemory.channel0.FRFCFS-WQF.channel0.rank0.bank3.bankPower 1.03595e+07W
```

結論：full miss rate 比較低。

Q3(15%):

replacement_policy = Param.BaseReplacementPolicy(LFURP(),"Replacement policy")

編譯 scons 執行 quicksort. 利用 quick sort 去比較預設(LRU)跟 LFU 的差別，比較 miss rate。

- 執行指令：

./build/X86/gem5.opt configs/example/se.py -c ./quicksort --cpu-type=TimingSimpleCPU --caches --l2cache --l3cache --l1i_size=32kB --l1d_size=32kB --l2_size=128kB --l3_size=1MB --mem-type=NVMainMemory --nvmain-config=../NVmain/Config/PCM_ISSCC_2012_4GB.config

- 預設(LRU):
  - 修正程式碼：

```
class L3Cache(Cache):
    assoc = 8
    tag_latency = 20
    data_latency = 20
    response_latency = 20
    mshrs = 20
    tgts_per_mshr = 12
    write_buffers = 8
    #replacement_policy = Param.BaseReplacementPolicy(LFURP(),"Replacement policy")
```

```
system.l3.overall_accesses::total              518718                       # number of overall (read+write) accesses
system.l3.ReadExReq_miss_rate::cpu.data      0.989091                       # miss rate for ReadExReq accesses
system.l3.ReadExReq_miss_rate::total         0.989091                       # miss rate for ReadExReq accesses
system.l3.ReadSharedReq_miss_rate::cpu.inst            1                    # miss rate for ReadSharedReq accesses
system.l3.ReadSharedReq_miss_rate::cpu.data     0.492719                    # miss rate for ReadSharedReq accesses
system.l3.ReadSharedReq_miss_rate::total     0.493115                       # miss rate for ReadSharedReq accesses
system.l3.demand_miss_rate::cpu.inst                1                       # miss rate for demand accesses
system.l3.demand_miss_rate::cpu.data         0.553463                       # miss rate for demand accesses
system.l3.demand_miss_rate::total            0.553769                       # miss rate for demand accesses
system.l3.overall_miss_rate::cpu.inst               1                       # miss rate for overall accesses
system.l3.overall_miss_rate::cpu.data        0.553463                       # miss rate for overall accesses
system.l3.overall_miss_rate::total           0.553769                       # miss rate for overall accesses
system.l3.ReadExReq_avg_miss_latency::cpu.data 289842.843026                # average ReadExReq miss latency
```

- 更改更改 policy 後的(LFU)
  - 修正程式碼：

```
class L3Cache(Cache):
    assoc = 8
    tag_latency = 20
    data_latency = 20
    response_latency = 20
    mshrs = 20
    tgts_per_mshr = 12
    write_buffers = 8
    replacement_policy = Param.BaseReplacementPolicy(LFURP(),"Replacement policy")
```

```
system.l3.ReadSharedReq_miss_rate::total     0.672610                       # miss rate for ReadSharedReq access
system.l3.demand_miss_rate::cpu.inst                1                       # miss rate for demand accesses
system.l3.demand_miss_rate::cpu.data         0.711455                       # miss rate for demand accesses
system.l3.demand_miss_rate::total            0.711653                       # miss rate for demand accesses
system.l3.overall_miss_rate::cpu.inst               1                       # miss rate for overall accesses
system.l3.overall_miss_rate::cpu.data        0.711455                       # miss rate for overall accesses
system.l3.overall_miss_rate::total           0.711653                       # miss rate for overall accesses
```

去 config.ini 看設置 L3 replacement policy.

```
[system.cpu.dcache.replacement_policy]
type=LRURP
eventq_index=0

[system.cpu.dcache.tags]
type=BaseSetAssoc
assoc=2
block_size=64
```

更改前

```
[system.l3.replacement_policy]
type=LFURP
eventq_index=0
```

更改後

一開始預設的 LRU 的 miss rate 會比較低(0.5 多)，到後面會被更改到 0.7 多。

Q4(15%):

我不會 qaq