**目录**

[1. Xml文件中的标签分类 1](#_Toc532048053)

[2. 不同组件的参数信息 2](#_Toc532048054)

[Predictor，name=PBT (以alpha 21264为例) 2](#_Toc532048055)

[BTB，name=BTB 2](#_Toc532048056)

[Itlb/Dtlb, name=itlb/dtlb 2](#_Toc532048057)

[Icache/Dcache, name=icache/dcache 2](#_Toc532048058)

[Core，name=core0/n 2](#_Toc532048059)

[L1Directory/L2Directory，name= L1Directory0/ L2Directory0 4](#_Toc532048060)

[L2/L3, name= L20/L30 5](#_Toc532048061)

[NoC, name=noc0 5](#_Toc532048062)

[MC，name=mc (Memory controllers are for DDR(2,3...) DIMMs) 6](#_Toc532048063)

[NIU,name= niu (On chip 10Gb Ethernet NIC, including XAUI Phy and MAC controller) 6](#_Toc532048064)

[PCIe,name= pcie (On chip PCIe controller, including Phy) 7](#_Toc532048065)

[Flashc,name=flashc 7](#_Toc532048066)

[System,name=system 7](#_Toc532048067)

[3. 各个组件在运行时需要收集的状态信息 8](#_Toc532048068)

[BTB 8](#_Toc532048069)

[Itlb/dtlb 8](#_Toc532048070)

[Icache 8](#_Toc532048071)

[Dcache 9](#_Toc532048072)

[Core (如果是x86，则所有的指令指代的都是微操作) 9](#_Toc532048073)

[L1Directory (l1 cache coherent)/ L2Directory (l2 cache coherent) 11](#_Toc532048074)

[L2 (L2 cache)/ L3 (L3 cache) 11](#_Toc532048075)

[NoC (Network on Chip) 11](#_Toc532048076)

[MC (Memory controller) 11](#_Toc532048077)

[NIU (NIC unit)/ PCIe/Flashc 11](#_Toc532048078)

[System 12](#_Toc532048079)

[4. 其它信息 12](#_Toc532048080)

[RAT的两种设计方法：RAM/CAM 12](#_Toc532048081)

[memory\_ports 12](#_Toc532048082)

# Xml文件中的标签分类

* Component：表示着实际的硬件结构(root除外)。

参数ID：用于唯一表示结构的名称，同时包含着从system开始到归属信息

参数name：表示着硬件结构的名称

|  |  |  |
| --- | --- | --- |
| System | Core | Predictor |
| BTB |
| itlb |
| dtlb |
| Icache |
| Dcache |
| L1Directory (l1 cache coherent) | |
| L2Directory (l2 cache coherent) | |
| L2 (L2 cache) | |
| L3 (L3 cache) | |
| NoC (Network on Chip) | |
| MC (Memory controller) | |
| NIU (NIC unit) | |
| PCIe | |
| Flashc (flash controller) | |

* Param：用于指示组件结构的某些固定的参数值，一部分实现根据实际情况设定，另一部分根据gem5的config文件中的设置得到

参数name：指明参数代表的含义

参数value：指明参数的具体数值。如果需要gem5的config文件指明，则需要使用类似于[value="REPLACE{config.system.cpu.numThreads}"]的语句，给出对应的在gem5的config文件中的属性名称

* Stat：用于指示组件在运行期间的一些统计信息，需要从gem5的stats文件中获取得到

参数name：指明统计信息的含义

参数value：由gem5指明，格式类似于

[value="REPLACE{stats.system.cpu.iq.iqInstsIssued}"]

# 不同组件的参数信息

## Predictor，name=PBT (以alpha 21264为例)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| local\_predictor\_size | 10,3 | 第一级，第二级表项大小 |
| local\_predictor\_entries | 1024 | 第一级和第二级一样 |
| global\_predictor\_entries | 4096 | 表项数 |
| global\_predictor\_bits | 2 | 每个表项的大小 |
| chooser\_predictor\_entries | 4096 | 表项数 |
| chooser\_predictor\_bits | 2 | 每个表项的大小 |

## BTB，name=BTB

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| BTB\_config | 4096,4,2,1, 1,3 | Size,line,assoc,banks, thoughtput(n/T), latency(T) |

## Itlb/Dtlb, name=itlb/dtlb

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| number\_entries | config.system.cpu.itb.size | Gem5中的size即为表项数 |

## Icache/Dcache, name=icache/dcache

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| icache\_config | config.system.cpu.icache.size,  config.system.cpu.icache.tags.block\_size,  config.system.cpu.icache.assoc,1,1, config.system.cpu.icache.hit\_latency,  config.system.cpu.icache.tags.block\_size,0 | Size,line,assoc,  Banks,  throughput(n/T),  latency(T),  -,cache\_policy  (WT=0,WB=1) |
| buffer\_sizes | config.system.cpu.icache.mshrs,  config.system.cpu.icache.mshrs,  config.system.cpu.icache.mshrs,  config.system.cpu.icache.mshrs | 通过mshrs的个数计算cache buffer的大小 |

## Core，name=core0/n

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| clock\_rate | 1e-6/  config.system.cpu\_clk\_domain  .clock | 时钟频率，gem5的cpu clock以s为单位给出 |
| vdd | 0  (system.cpu\_voltage\_domain) | 0=ITRS default vdd |
| power\_gating\_vcc | -1 | 代表默认值 |
| opt\_local | 0 | **含义不明** |
| instruction\_length | 32 | 指令长度 |
| opcode\_width | 7 | 指令中操作数字段长度 |
| x86 | 0/1 | 是/否 |
| micro\_opcode\_width | 8 | 微指令操作数字段长度 |
| machine\_type | 0/1 | OOO/INO |
| number\_hardware\_threads | config.system.cpu.numThreads | Cpu支持的线程数 |
| fetch\_width | config.system.cpu.fetchWidth | 取决于L1 cache的cacheline的大小 |
| number\_instruction  \_fetch\_ports | 1  (代表icache的端口数目,同时BTB的端口数与其一致) | 在单线程CPU中始终为1，在SMT中可能>1 |
| decode\_width | config.system.cpu.decodeWidth | 译码宽度会决定RAT的端口数 |
| issue\_width | config.system.cpu.issueWidth  (issueWidth=dispatchWidth) | 发射宽度 |
| peak\_issue\_width | config.system.cpu.issueWidth | Gem5中为提供峰值的issue width |
| commit\_width | config.system.cpu.commitWidth | 提交宽度决定了寄存器堆的端口数量 |
| fp\_issue\_width | 2 | **Gem5中未明确指明** |
| prediction\_width | 1 | 分支预测器可同时预测的分支指令数目  **Gem5中未指明** |
| pipelines\_per\_core | 1,1  (mcpat未区分整数和浮点数流水线) | 1整数，1浮点流水线  **configs中未指明** |
| pipeline\_depth | 8，8  (如果不支持浮点流水线，第二个参数代表浮点操作的平均周期数) | 流水线的宽度  **configs中未指明** |
| ALU\_per\_core | 6  (包括adder,shifter,logical) | Configs包含此信息 |
| MUL\_per\_core | 1  (包括mul,div) | Configs包含此信息 |
| FPU\_per\_core | 2 | Configs包含此信息 |
| instruction\_buffer\_size | 32  (configs = fetchBufferSize) | IF和ID阶段之间的buffer大小 |
| decoded\_stream\_  buffer\_size | 16 | ID和sche/exe之间的buffer的大小  **configs中未指明** |
| instruction\_  window\_scheme | 0/1  (OOO, PHYREG/RS) | **configs中未指明**  可以根据参数看出 |
| instruction\_window\_size | config.system.cpu.  numIQEntries | **Configs未区分FP/INT** |
| fp\_instruction\_  window\_size | config.system.cpu.  numIQEntries |  |
| ROB\_size | config.system.cpu  .numROBEntries |  |
| archi\_Regs\_IRF\_size | 32 | **Config文件中未给出架构寄存器数目** |
| archi\_Regs\_FRF\_size | 32 |
| phy\_Regs\_IRF\_size | config.system.cpu.  numPhysIntRegs | 物理寄存器堆的大小 |
| phy\_Regs\_FRF\_size | config.system.cpu.  numPhysFloatRegs |  |
| rename\_scheme | 0/1  (RAM based(0)/CAM based(1)) | RAT的两种设计方法  **configs中未指明** |
| register\_windows\_size | 0  (sun处理器中有此信息) | 0代表无 |
| LSU\_order | inorder/OoO/out-of-order | **configs中未指明** |
| store\_buffer\_size | config.system.cpu.SQEntries |  |
| load\_buffer\_size | config.system.cpu.LQEntries | Inorder无LQ |
| memory\_ports | 2  (代表可并发的访存操作数)  (usually In-order=1 / OOO>=2) | 决定了  LB/SB/Dcache的端口数  **Configs未明确说明** |
| RAS\_size | config.system.cpu.  branchPred.RASSize | RAS的大小 |

## L1Directory/L2Directory，name= L1Directory0/ L2Directory0

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| Directory\_type | 0 = cam based shadowed tag  1 = directory cache | **Configs未明确说明** |
| Dir\_config | 4096,2,0,1,100,100, 8  (capacity,block\_width, assoc, banks,throughput,latency) | **Configs未明确说明** |
| buffer\_sizes | 8, 8, 8, 8  (miss\_buffer\_size(MSHR),  fill\_buffer\_size,  prefetch\_buffer\_size,  wb\_buffer\_size) | **Configs未明确说明** |
| clockrate | 3400  (应该是CPU时钟) | **Configs未明确说明** |
| vdd | 0(ITRS的默认vdd) | **Configs未明确说明**  **system.voltage\_domain** |
| power\_gating\_vcc | -1(默认) | **Configs未明确说明** |
| ports | 1,1,1  (r,w,rw search ports) | **Configs未明确说明** |
| device\_type | 0/1/2/3/4  (itrs-hp, itrs-lstp, itrs-lop  lp-dram, comm-dram) | **Configs未明确说明** |

## L2/L3, name= L20/L30

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| L2\_config | 1048576,32, 8, 8, 8, 23, 32, 1  (capacity,block\_width, assoc, banks,throughput,latency) | **Config如果设置了L2，则会有描述** |
| buffer\_sizes | 16, 16, 16, 16  (miss\_buffer\_size(MSHR),  fill\_buffer\_size,  prefetch\_buffer\_size,  wb\_buffer\_size) | **Config如果设置了L2，则会有描述** |
| clockrate | 3400  (应该是CPU时钟) | **Config中为**  **system.cpu\_clk\_domain** |
| vdd | 0(ITRS的默认vdd) | **Configs未明确说明**  **system.voltage\_domain** |
| power\_gating\_vcc | -1(默认) | **Configs未明确说明** |
| ports | 1,1,1  (r,w,rw search ports) | **Configs未明确说明** |
| device\_type | 0/1/2/3/4  (itrs-hp, itrs-lstp, itrs-lop  lp-dram, comm-dram) | **Configs未明确说明** |

## NoC, name=noc0

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| clockrate | 3400  (system.cpu\_clk\_domain) | CPU时钟 |
| vdd | 0(ITRS的默认vdd)  (system.voltage\_domain) | CPU电压 |
| power\_gating\_vcc | -1(默认) | **Configs未明确说明** |
| type | 0(bus)/1(NoC) | **Gem5一般为bus** |
| horizontal\_nodes | 1 | 对于type=1，需要设置节点个数. **Configs未明确说明** |
| vertical\_nodes | 1 |
| has\_global\_link | 0(No)/1(Yes) | **Configs未明确说明** |
| link\_throughput | 1(/T) | **Configs未明确说明** |
| link\_latency | 1(/T) | **Configs未明确说明** |
| input\_ports | 1 | 对于type=0，两者为1. **Configs未明确说明** |
| output\_ports | 1 |
| flit\_bits | 64 | **Configs未明确说明** |
| chip\_coverage | 1  (存在多个NoC，分别覆盖一部分) | **Configs未明确说明** |
| link\_routing\_  over\_percentage | 0.5  (默认值) | **Configs未明确说明** |

## MC，name=mc (Memory controllers are for DDR(2,3...) DIMMs)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| type | 0/1  (low power/high perf) | **Configs未明确说明** |
| mc\_clock | 200  (DIMM IO bus clock rate MHz) | **Configs文件中给出了clk\_domain和一些其它时间** |
| vdd | 0(ITRS的默认vdd)  (system.voltage\_domain) | CPU电压 |
| power\_gating\_vcc | -1(默认) | **Configs未明确说明** |
| peak\_transfer\_rate | 3200 MB/S | **可以根据Config计算得到** |
| block\_size | 64 B  (system.mem\_ctrls.  read\_buffer\_size) | **一次memory burst transfer的单位** |
| number\_mcs | 0  (目前McPAT只支持同构的MC) | **Configs未明确说明** |
| memory\_channels  \_per\_mc | 1  (system.mem\_ctrls.channels) | Channel数 |
| number\_ranks | 2 (system.mem\_ctrls. ranks\_per\_channel) | Channel的rank数 |
| withPHY | 0/1(No/Yes) | **Configs未明确说明** |
| req\_window\_size  \_per\_channel | 32  (system.mem\_ctrls.  read\_buffer\_size) | **不确定** |
| IO\_buffer\_size  \_per\_channel | 32  (system.mem\_ctrls.  read\_buffer\_size) | **不确定** |
| databus\_width | 128 | **Configs未明确说明** |
| addressbus\_width | 51 | **Configs未明确说明** |

## NIU,name= niu (On chip 10Gb Ethernet NIC, including XAUI Phy and MAC controller)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| type | 0/1  (low power/high perf) | **Configs未明确说明** |
| clockrate | 350 MHz | **Configs未明确说明** |
| vdd | 0(ITRS的默认vdd) | **Configs未明确说明** |
| power\_gating\_vcc | -1(默认) | **Configs未明确说明** |
| number\_units | 0 (each Ethernet  controller only have one port) | **Configs未明确说明** |

## PCIe,name= pcie (On chip PCIe controller, including Phy)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| type | 0/1  (low power/high perf) | **Configs未明确说明** |
| withPHY | 0/1(on chip) | **Configs未明确说明** |
| clockrate | 350 | **Configs未明确说明** |
| vdd | 0(ITRS的默认vdd) | **Configs未明确说明** |
| power\_gating\_vcc | -1(默认) | **Configs未明确说明** |
| number\_units | 0 | **Configs未明确说明** |
| num\_channels | 2/4/8/16/32 | **Configs未明确说明** |

## Flashc,name=flashc

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| type | 0/1  (low power/high perf) | **Configs未明确说明** |
| number\_flashcs | 0 | **Configs未明确说明** |
| vdd | 0(ITRS的默认vdd) | **Configs未明确说明** |
| power\_gating\_vcc | -1(默认) | **Configs未明确说明** |
| withPHY | 0/1 | **Configs未明确说明** |
| peak\_transfer\_rate | 200 MB/s | **Configs未明确说明** |

## System,name=system

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| number\_of\_cores | 1 | 核数 |
| number\_of\_L1  Directories | 0 | L1 cache有几个目录协议 |
| number\_of\_L2  Directories | 0 | L2 cache有几个目录协议 |
| number\_of\_L2s | 1 | L2 cache的个数 |
| Private\_L2 | 1(Private)/0(shared/coherent) | 是否私有的L2 |
| number\_of\_L3s | 0 | L3 cache的个数 |
| number\_of\_NoCs | 0 | NoC的个数 |
| homogeneous\_cores | 0/1(Yes) | 是否为同构的 |
| homogeneous\_L2s | 0/1(Yes) |
| homogeneous\_L1  Directories | 0/1(Yes) |
| homogeneous\_L2  Directories | 0/1(Yes) |
| homogeneous\_L3s | 0/1(Yes) |
| homogeneous\_ccs | 0/1(Yes) | cache coherence |
| homogeneous\_NoCs | 0/1(Yes) | 是否为同构的 |
| core\_tech\_node | 64(nm) | 制造工艺 |
| target\_core\_  clockrate | 1e-6/  config.system.cpu\_clk\_domain.clock | Mhz |
| temperature | 380 (Kelvin) |  |
| number\_cache\_levels | 1 | Cache的层数 |
| interconnect\_  projection\_type | 0: aggressive wire technology;  1: conservative wire technology | 互连的技术类型 |
| device\_type | 0: HP(High Performance Type);  1: LSTP(Low standby power)  2: LOP (Low Operating Power) | 设备类型 |
| longer\_channel\_  device | 0: no use;  1: use when possible |  |
| power\_gating | 0: not enabled;  1: enabled | 电源/功率门控是否使用 |
| machine\_bits | 64 | 机器位数 |
| virtual\_address  \_width | 64 | 地址宽度将会决定cache, LSQ, cache控制器中的buffer中的tag的大小。如果没有设置，默认为机器位数 |
| physical\_address  \_width | 52 |
| virtual\_memory  \_page\_size | 4096 |

# 各个组件在运行时需要收集的状态信息

## BTB

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| read\_accesses | stats.system.cpu.branchPred.BTBLookups |  |
| write\_accesses | stats.system.cpu.commit.branches | 每个分支都会写BTB |

## Itlb/dtlb

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| total\_accesses | stats.system.cpu.itb\_walker\_cache  .tags.tag\_accesses | 总访问次数 |
| total\_misses | stats.system.cpu.itb\_walker\_cache  .no\_allocate\_misses | 发生miss的次数 |
| conflicts | 0  stats.system.cpu.itb\_walker\_cache  .tags.replacements | 冲突次数(都为0) |

## Icache

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| read\_accesses | stats.system.cpu.icache  .ReadReq\_accesses::total |  |
| read\_misses | stats.system.cpu.icache  .ReadReq\_misses::total |  |
| conflicts | stats.system.cpu.icache.  tags.replacements |  |

## Dcache

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| read\_accesses | stats.system.cpu.dcache  .ReadReq\_accesses::total |  |
| read\_misses | stats.system.cpu.dcache  .ReadReq\_misses::total |  |
| write\_accesses | stats.system.cpu.dcache  .ReadReq\_misses::total |  |
| write\_misses | stats.system.cpu.dcache  .WriteReq\_misses::total |  |
| conflicts | stats.system.cpu.dcache  .tags.replacements |  |

## Core (如果是x86，则所有的指令指代的都是微操作)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| total\_instructions | stats.system.cpu.iq.iqInstsIssued | 定义了模拟的周期信息 |
| int\_instructions | stats.system.cpu.iq.FU\_type\_0::No\_OpClass + stats.system.cpu.iq.FU\_type\_0::IntAlu + stats.system.cpu.iq.FU\_type\_0::IntMult + stats.system.cpu.iq.FU\_type\_0::IntDiv + stats.system.cpu.iq.FU\_type\_0::IprAccess |
| fp\_instructions | stats.system.cpu.iq.FU\_type\_0::FloatAdd + stats.system.cpu.iq.FU\_type\_0::FloatCmp + stats.system.cpu.iq.FU\_type\_0::FloatCvt + stats.system.cpu.iq.FU\_type\_0::FloatMult + stats.system.cpu.iq.FU\_type\_0::FloatDiv + stats.system.cpu.iq.FU\_type\_0::FloatSqrt |
| branch\_instructions | stats.system.cpu.branchPred.condPredicted |
| branch\_  mispredictions | stats.system.cpu.branchPred.condIncorrect |
| load\_instructions | stats.system.cpu.iq.FU\_type\_0::MemRead + stats.system.cpu.iq.FU\_type\_0::InstPrefetch |
| store\_instructions | stats.system.cpu.iq.FU\_type\_0::MemWrite |
| Committed  \_instructions | stats.system.cpu.commit.committedInsts |
| Committed  \_int\_instructions | stats.system.cpu.commit.int\_insts |
| Committed  \_fp\_instructions | stats.system.cpu.commit.fp\_insts |
| pipeline\_duty\_cycle | 1 (runtime\_ipc/peak\_ipc） | **Config**  **未指明** |
| total\_cycles | stats.system.cpu.numCycles |  |
| idle\_cycles | stats.system.cpu.idleCycles |  |
| busy\_cycles | stats.system.cpu.numCycles - stats.system.cpu.idleCycles |  |
| ROB\_reads | stats.system.cpu.rob.rob\_reads | ROB stats |
| ROB\_writes | stats.system.cpu.rob.rob\_writes |
| rename\_reads | stats.system.cpu.rename.int\_rename\_lookups |  |
| rename\_writes | int(  stats.system.cpu.rename.RenamedOperands \* stats.system.cpu.rename.int\_rename\_lookups / stats.system.cpu.rename.RenameLookups) | update dest regs |
| fp\_rename\_reads | stats.system.cpu.rename.fp\_rename\_lookups |  |
| fp\_rename\_writes | int(stats.system.cpu.rename.RenamedOperands \* stats.system.cpu.rename.fp\_rename\_lookups / stats.system.cpu.rename.RenameLookups) |  |
| inst\_window\_reads | stats.system.cpu.iq.int\_inst\_queue\_reads | Inst window stats |
| inst\_window\_writes | stats.system.cpu.iq.int\_inst\_queue\_writes |
| inst\_window\_  wakeup\_accesses | stats.system.cpu.iq  .int\_inst\_queue\_wakeup\_accesses |
| fp\_inst\_window  \_reads | stats.system.cpu.iq.fp\_inst\_queue\_reads |
| fp\_inst\_window  \_writes | stats.system.cpu.iq.fp\_inst\_queue\_writes |
| fp\_inst\_window  \_wakeup\_accesses | stats.system.cpu.iq  .fp\_inst\_queue\_wakeup\_accesses |
| int\_regfile\_reads | stats.system.cpu.int\_regfile\_reads | RF accesses |
| float\_regfile\_reads | stats.system.cpu.fp\_regfile\_reads |
| int\_regfile\_writes | stats.system.cpu.int\_regfile\_writes |
| float\_regfile  \_writes | stats.system.cpu.fp\_regfile\_writes |
| function\_calls | stats.system.cpu.commit.function\_calls |  |
| context\_switches | stats.system.cpu.workload.num\_syscalls |  |
| ialu\_accesses | stats.system.cpu.iq.int\_alu\_accesses |  |
| fpu\_accesses | stats.system.cpu.iq.fp\_alu\_accesses |  |
| mul\_accesses | **可以只包括前两类，如果mul在前两者中** |  |
| cdb\_alu\_accesses | **为理解，一些情况下，与上三者一样** |  |
| cdb\_mul\_accesses |  |
| cdb\_fpu\_accesses |  |
| IFU\_duty\_cycle | **AF for max power computation.**  **Do not change them, unless you understand them** | 工作周期 |
| LSU\_duty\_cycle |  |
| MemManU\_I\_duty  \_cycle |  |
| MemManU\_D\_duty  \_cycle |  |
| ALU\_duty\_cycle |  |
| MUL\_duty\_cycle |  |
| FPU\_duty\_cycle |  |
| ALU\_cdb\_duty\_cycle |  |
| MUL\_cdb\_duty\_cycle |  |
| FPU\_cdb\_duty\_cycle |  |

## L1Directory (l1 cache coherent)/ L2Directory (l2 cache coherent)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| read\_accesses | 800000 | **Config**  **未指明** |
| write\_accesses | 27276 |
| read\_misses | 1632 |
| write\_misses | 183 |
| conflicts | 20 |

## L2 (L2 cache)/ L3 (L3 cache)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| read\_accesses | stats.system.l2.ReadReq\_accesses::total | **不确定** |
| write\_accesses | stats.system.l2.Writeback\_accesses::total |
| read\_misses | system.l2.ReadReq\_hits::total |
| write\_misses | system.l2.Writeback\_hits::total |
| conflicts | system.l2.tags.replacements |
| duty\_cycle | 1.0 | **占空比** |

## NoC (Network on Chip)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| total\_accesses | 10000 | **Config**  **未指明** |
| duty\_cycle | 1 |

## MC (Memory controller)

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| memory\_accesses | 33333 | **Config**  **未指明** |
| memory\_reads | 16667 |
| memory\_writes | 16667 |

## NIU (NIC unit)/ PCIe/Flashc

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| duty\_cycle | 1 | **Config**  **未指明** |
| total\_load\_perc | 0.7 |

## System

|  |  |  |
| --- | --- | --- |
| Name | Value | Explain |
| total\_cycles | stats.system.cpu.numCycles |  |
| idle\_cycles | stats.system.cpu.idleCycles |  |
| busy\_cycles | stats.system.cpu.numCycles - stats.system.cpu.idleCycles |  |

# 其它信息

## RAT的两种设计方法：RAM/CAM

1. McPAT支持基于RAM的和基于CAM的两种重命名设计逻辑，分别可以在Intel和Alpha的处理器中看到这两种设计
2. 基于RAM的RAT（Register alias table）：实际的架构有Intel P6。这种设计中RAT被建模为一个表项数为体系结构寄存器数目的Array。每一个表现利用指令中的寄存器标号索引，对应的物理寄存器标号存储在表项之中
3. 基于CAM的RAT设计：实际架构有Alpha的处理器。此时RAT被建模为电路级的CAM阵列模型，表项数为物理寄存器的数量。每个表项两个字段：映射到该物理寄存器的架构寄存器的标号；有效位。

## memory\_ports

未找到gem5设置端口的位置

附录：template\_xeon.xml

|  |
| --- |
| <?xml version=**"1.0"** ?>  <component id=**"root"** name=**"root"**>  <component id=**"system"** name=**"system"**>  <!--McPAT will skip the components if number is set to 0 -->  <param name=**"number\_of\_cores"** value=**"1"**/>  <param name=**"number\_of\_L1Directories"** value=**"0"**/>  <param name=**"number\_of\_L2Directories"** value=**"0"**/>  <param name=**"number\_of\_L2s"** value=**"1"**/><!-- This number means how many L2 clusters in each cluster there can be multiple banks/ports -->  <param name=**"Private\_L2"** value=**"0"**/><!--1 Private, 0 shared/coherent -->  <param name=**"number\_of\_L3s"** value=**"0"**/><!-- This number means how many L3 clusters -->  <param name=**"number\_of\_NoCs"** value=**"0"**/>  <param name=**"homogeneous\_cores"** value=**"1"**/><!--1 means homo -->  <param name=**"homogeneous\_L2s"** value=**"1"**/>  <param name=**"homogeneous\_L1Directories"** value=**"1"**/>  <param name=**"homogeneous\_L2Directories"** value=**"1"**/>  <param name=**"homogeneous\_L3s"** value=**"1"**/>  <param name=**"homogeneous\_ccs"** value=**"1"**/><!--cache coherence hardware -->  <param name=**"homogeneous\_NoCs"** value=**"1"**/>  <param name=**"core\_tech\_node"** value=**"65"**/><!-- nm -->  <param name=**"target\_core\_clockrate"** value=**"1e-6/config.system.cpu\_clk\_domain.clock"**/><!--MHz -->  <param name=**"temperature"** value=**"380"**/><!-- Kelvin -->  <param name=**"number\_cache\_levels"** value=**"1"**/>  <param name=**"interconnect\_projection\_type"** value=**"0"**/><!--0: aggressive wire technology; 1: conservative wire technology -->  <param name=**"device\_type"** value=**"0"**/><!--0: HP(High Performance Type); 1: LSTP(Low standby power) 2: LOP (Low Operating Power) -->  <param name=**"longer\_channel\_device"** value=**"0"**/><!-- 0 no use; 1 use when possible -->  <param name=**"power\_gating"** value=**"1"**/><!-- 0 not enabled; 1 enabled -->  <param name=**"machine\_bits"** value=**"64"**/>  <param name=**"virtual\_address\_width"** value=**"64"**/>  <param name=**"physical\_address\_width"** value=**"52"**/>  <param name=**"virtual\_memory\_page\_size"** value=**"4096"**/>  <!-- address width determines the tag\_width in Cache, LSQ and buffers in cache controller  default value is machine\_bits, if not set -->  <stat name=**"total\_cycles"** value=**"stats.system.cpu.numCycles"**/>  <stat name=**"idle\_cycles"** value=**"stats.system.cpu.idleCycles"**/>  <stat name=**"busy\_cycles"** value=**"stats.system.cpu.numCycles - stats.system.cpu.idleCycles"**/>  <!--This page size(B) is complete different from the page size in Main memo section. this page size is the size of  virtual memory from OS/Archi perspective; the page size in Main memo section is the actual physical line in a DRAM bank -->  <!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* cores \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->  <component id=**"system.core0"** name=**"core0"**>  <!-- Core property -->  <param name=**"clock\_rate"** value=**"1e-6/config.system.cpu\_clk\_domain.clock"**/>  <param name=**"vdd"** value=**"1.25"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"opt\_local"** value=**"0"**/><!-- for cores with unknown timing, set to 0 to force off the opt flag -->  <param name=**"instruction\_length"** value=**"32"**/>  <param name=**"opcode\_width"** value=**"16"**/>  <param name=**"x86"** value=**"1"**/>  <param name=**"micro\_opcode\_width"** value=**"8"**/>  <param name=**"machine\_type"** value=**"0"**/>  <!-- inorder/OoO; 1 inorder; 0 OOO-->  <param name=**"number\_hardware\_threads"** value=**"config.system.cpu.numThreads"**/>  <!-- number\_instruction\_fetch\_ports(icache ports) is always 1 in single-thread processor,  it only may be more than one in SMT processors. BTB ports always equals to fetch ports since  branch information in consecutive branch instructions in the same fetch group can be read out from BTB once.-->  <param name=**"fetch\_width"** value=**"config.system.cpu.fetchWidth"**/>  <!-- fetch\_width determines the size of cachelines of L1 cache block -->  <param name=**"number\_instruction\_fetch\_ports"** value=**"1"**/>  <param name=**"decode\_width"** value=**"config.system.cpu.decodeWidth"**/>  <!-- decode\_width determines the number of ports of the  renaming table (both RAM and CAM) scheme -->  <param name=**"issue\_width"** value=**"config.system.cpu.issueWidth"**/>  <param name=**"peak\_issue\_width"** value=**"config.system.cpu.issueWidth"**/>  <!-- issue\_width determines the number of ports of Issue window and other logic  as in the complexity effective processors paper; issue\_width==dispatch\_width -->  <param name=**"commit\_width"** value=**"config.system.cpu.commitWidth"**/>  <!-- commit\_width determines the number of ports of register files -->  <param name=**"fp\_issue\_width"** value=**"2"**/>  <param name=**"prediction\_width"** value=**"1"**/>  <!-- number of branch instructions can be predicted simultaneously-->  <!-- Current version of McPAT does not distinguish int and floating point pipelines  Theses parameters are reserved for future use.-->  <param name=**"pipelines\_per\_core"** value=**"1,1"**/>  <!--integer\_pipeline and floating\_pipelines, if the floating\_pipelines is 0, then the pipeline is shared-->  <param name=**"pipeline\_depth"** value=**"31,31"**/>  <!-- pipeline depth of int and fp, if pipeline is shared, the second number is the average cycles of fp ops -->  <!-- issue and exe unit-->  <param name=**"ALU\_per\_core"** value=**"6"**/>  <!-- contains an adder, a shifter, and a logical unit -->  <param name=**"MUL\_per\_core"** value=**"1"**/>  <!-- For MUL and Div -->  <param name=**"FPU\_per\_core"** value=**"2"**/>  <!-- buffer between IF and ID stage -->  <param name=**"instruction\_buffer\_size"** value=**"32"**/>  <!-- buffer between ID and sche/exe stage -->  <param name=**"decoded\_stream\_buffer\_size"** value=**"16"**/>  <param name=**"instruction\_window\_scheme"** value=**"0"**/><!-- 0 PHYREG based, 1 RSBASED-->  <!-- McPAT support 2 types of OoO cores, RS based and physical reg based-->  <param name=**"instruction\_window\_size"** value=**"config.system.cpu.numIQEntries"**/>  <param name=**"fp\_instruction\_window\_size"** value=**"config.system.cpu.numIQEntries"**/>  <!-- the instruction issue Q as in Alpha 21264; The RS as in Intel P6 -->  <param name=**"ROB\_size"** value=**"config.system.cpu.numROBEntries"**/>  <!-- each in-flight instruction has an entry in ROB -->  <!-- registers -->  <param name=**"archi\_Regs\_IRF\_size"** value=**"16"**/><!-- X86-64 has 16GPR -->  <param name=**"archi\_Regs\_FRF\_size"** value=**"32"**/><!-- MMX + XMM -->  <!-- if OoO processor, phy\_reg number is needed for renaming logic,  renaming logic is for both integer and floating point insts. -->  <param name=**"phy\_Regs\_IRF\_size"** value=**"config.system.cpu.numPhysIntRegs"**/>  <param name=**"phy\_Regs\_FRF\_size"** value=**"config.system.cpu.numPhysFloatRegs"**/>  <!-- rename logic -->  <param name=**"rename\_scheme"** value=**"0"**/>  <!-- can be RAM based(0) or CAM based(1) rename scheme  RAM-based scheme will have free list, status table;  CAM-based scheme have the valid bit in the data field of the CAM  both RAM and CAM need RAM-based checkpoint table, checkpoint\_depth=# of in\_flight instructions;  Detailed RAT Implementation see TR -->  <param name=**"register\_windows\_size"** value=**"0"**/>  <!-- how many windows in the windowed register file, sun processors;  no register windowing is used when this number is 0 -->  <!-- In OoO cores, loads and stores can be issued whether inorder(Pentium Pro) or (OoO)out-of-order(Alpha),  They will always try to execute out-of-order though. -->  <param name=**"LSU\_order"** value=**"inorder"**/>  <param name=**"store\_buffer\_size"** value=**"config.system.cpu.SQEntries"**/>  <!-- By default, in-order cores do not have load buffers -->  <param name=**"load\_buffer\_size"** value=**"config.system.cpu.LQEntries"**/>  <!-- number of ports refer to sustain-able concurrent memory accesses -->  <param name=**"memory\_ports"** value=**"2"**/>  <!-- max\_allowed\_in\_flight\_memo\_instructions determines the # of ports of load and store buffer  as well as the ports of Dcache which is connected to LSU -->  <!-- dual-pumped Dcache can be used to save the extra read/write ports -->  <param name=**"RAS\_size"** value=**"config.system.cpu.branchPred.RASSize"**/>  <!-- general stats, defines simulation periods;require total, idle, and busy cycles for sanity check -->  <!-- please note: if target architecture is X86, then all the instructions refer to (fused) micro-ops -->  <stat name=**"total\_instructions"** value=**"stats.system.cpu.iq.iqInstsIssued"**/>  <stat name=**"int\_instructions"** value=**"stats.system.cpu.iq.FU\_type\_0::No\_OpClass + stats.system.cpu.iq.FU\_type\_0::IntAlu + stats.system.cpu.iq.FU\_type\_0::IntMult + stats.system.cpu.iq.FU\_type\_0::IntDiv + stats.system.cpu.iq.FU\_type\_0::IprAccess"**/>  <stat name=**"fp\_instructions"** value=**"stats.system.cpu.iq.FU\_type\_0::FloatAdd + stats.system.cpu.iq.FU\_type\_0::FloatCmp + stats.system.cpu.iq.FU\_type\_0::FloatCvt + stats.system.cpu.iq.FU\_type\_0::FloatMult + stats.system.cpu.iq.FU\_type\_0::FloatDiv + stats.system.cpu.iq.FU\_type\_0::FloatSqrt"**/>  <stat name=**"branch\_instructions"** value=**"stats.system.cpu.branchPred.condPredicted"**/>  <stat name=**"branch\_mispredictions"** value=**"stats.system.cpu.branchPred.condIncorrect"**/>  <stat name=**"load\_instructions"** value=**"stats.system.cpu.iq.FU\_type\_0::MemRead + stats.system.cpu.iq.FU\_type\_0::InstPrefetch"**/>  <stat name=**"store\_instructions"** value=**"stats.system.cpu.iq.FU\_type\_0::MemWrite"**/>  <stat name=**"committed\_instructions"** value=**"stats.system.cpu.commit.committedInsts"**/>  <stat name=**"committed\_int\_instructions"** value=**"stats.system.cpu.commit.int\_insts"**/>  <stat name=**"committed\_fp\_instructions"** value=**"stats.system.cpu.commit.fp\_insts"**/>  <stat name=**"pipeline\_duty\_cycle"** value=**"1"**/><!--<=1, runtime\_ipc/peak\_ipc; averaged for all cores if homogeneous -->  <!-- the following cycle stats are used for heterogeneous cores only,  please ignore them if homogeneous cores -->  <stat name=**"total\_cycles"** value=**"stats.system.cpu.numCycles"**/>  <stat name=**"idle\_cycles"** value=**"stats.system.cpu.idleCycles"**/>  <stat name=**"busy\_cycles"** value=**"stats.system.cpu.numCycles - stats.system.cpu.idleCycles"**/>  <!-- instruction buffer stats -->  <!-- ROB stats, both RS and Phy based OoOs have ROB  performance simulator should capture the difference on accesses,  otherwise, McPAT has to guess based on number of committed instructions. -->  <stat name=**"ROB\_reads"** value=**"stats.system.cpu.rob.rob\_reads"**/>  <stat name=**"ROB\_writes"** value=**"stats.system.cpu.rob.rob\_writes"**/>  <!-- RAT accesses -->  <stat name=**"rename\_reads"** value=**"stats.system.cpu.rename.int\_rename\_lookups"**/><!--lookup in renaming logic -->  <stat name=**"rename\_writes"** value=**"int(stats.system.cpu.rename.RenamedOperands \* stats.system.cpu.rename.int\_rename\_lookups / stats.system.cpu.rename.RenameLookups)"**/><!--update dest regs. renaming logic -->  <stat name=**"fp\_rename\_reads"** value=**"stats.system.cpu.rename.fp\_rename\_lookups"**/>  <stat name=**"fp\_rename\_writes"** value=**"int(stats.system.cpu.rename.RenamedOperands \* stats.system.cpu.rename.fp\_rename\_lookups / stats.system.cpu.rename.RenameLookups)"**/>  <!-- decode and rename stage use this, should be total ic - nop -->  <!-- Inst window stats -->  <stat name=**"inst\_window\_reads"** value=**"stats.system.cpu.iq.int\_inst\_queue\_reads"**/>  <stat name=**"inst\_window\_writes"** value=**"stats.system.cpu.iq.int\_inst\_queue\_writes"**/>  <stat name=**"inst\_window\_wakeup\_accesses"** value=**"stats.system.cpu.iq.int\_inst\_queue\_wakeup\_accesses"**/>  <stat name=**"fp\_inst\_window\_reads"** value=**"stats.system.cpu.iq.fp\_inst\_queue\_reads"**/>  <stat name=**"fp\_inst\_window\_writes"** value=**"stats.system.cpu.iq.fp\_inst\_queue\_writes"**/>  <stat name=**"fp\_inst\_window\_wakeup\_accesses"** value=**"stats.system.cpu.iq.fp\_inst\_queue\_wakeup\_accesses"**/>  <!-- RF accesses -->  <stat name=**"int\_regfile\_reads"** value=**"stats.system.cpu.int\_regfile\_reads"**/>  <stat name=**"float\_regfile\_reads"** value=**"stats.system.cpu.fp\_regfile\_reads"**/>  <stat name=**"int\_regfile\_writes"** value=**"stats.system.cpu.int\_regfile\_writes"**/>  <stat name=**"float\_regfile\_writes"** value=**"stats.system.cpu.fp\_regfile\_writes"**/>  <!-- accesses to the working reg -->  <stat name=**"function\_calls"** value=**"stats.system.cpu.commit.function\_calls"**/>  <stat name=**"context\_switches"** value=**"stats.system.cpu.workload.num\_syscalls"**/>  <!-- Number of Windows switches (number of function calls and returns)-->  <!-- Alu stats by default, the processor has one FPU that includes the divider and  multiplier. The fpu accesses should include accesses to multiplier and divider -->  <stat name=**"ialu\_accesses"** value=**"stats.system.cpu.iq.int\_alu\_accesses"**/>  <stat name=**"fpu\_accesses"** value=**"stats.system.cpu.iq.fp\_alu\_accesses"**/>  <stat name=**"mul\_accesses"** value=**"0"**/>  <stat name=**"cdb\_alu\_accesses"** value=**"0"**/>  <stat name=**"cdb\_mul\_accesses"** value=**"0"**/>  <stat name=**"cdb\_fpu\_accesses"** value=**"0"**/>  <!-- multiple cycle accesses should be counted multiple times,  otherwise, McPAT can use internal counter for different floating point instructions  to get final accesses. But that needs detailed info for floating point inst mix -->  <!-- currently the performance simulator should  make sure all the numbers are final numbers,  including the explicit read/write accesses,  and the implicit accesses such as replacements and etc.  Future versions of McPAT may be able to reason the implicit access  based on param and stats of last level cache  The same rule applies to all cache access stats too! -->  <!-- following is AF for max power computation.  Do not change them, unless you understand them-->  <stat name=**"IFU\_duty\_cycle"** value=**"0.25"**/><!--depends on Icache line size and instruction issue rate -->  <stat name=**"LSU\_duty\_cycle"** value=**"0.25"**/>  <stat name=**"MemManU\_I\_duty\_cycle"** value=**"0.25"**/>  <stat name=**"MemManU\_D\_duty\_cycle"** value=**"0.25"**/>  <stat name=**"ALU\_duty\_cycle"** value=**"1"**/>  <stat name=**"MUL\_duty\_cycle"** value=**"0.3"**/>  <stat name=**"FPU\_duty\_cycle"** value=**"0.3"**/>  <stat name=**"ALU\_cdb\_duty\_cycle"** value=**"1"**/>  <stat name=**"MUL\_cdb\_duty\_cycle"** value=**"0.3"**/>  <stat name=**"FPU\_cdb\_duty\_cycle"** value=**"0.3"**/>  <param name=**"number\_of\_BPT"** value=**"2"**/>  <component id=**"system.core0.predictor"** name=**"PBT"**>  <!-- branch predictor; tournament predictor see Alpha implementation -->  <param name=**"local\_predictor\_size"** value=**"10,3"**/>  <param name=**"local\_predictor\_entries"** value=**"1024"**/>  <param name=**"global\_predictor\_entries"** value=**"4096"**/>  <param name=**"global\_predictor\_bits"** value=**"2"**/>  <param name=**"chooser\_predictor\_entries"** value=**"4096"**/>  <param name=**"chooser\_predictor\_bits"** value=**"2"**/>  <!-- These parameters can be combined like below in next version  <param name="load\_predictor" value="10,3,1024"/>  <param name="global\_predictor" value="4096,2"/>  <param name="predictor\_chooser" value="4096,2"/>  -->  </component>  <component id=**"system.core0.itlb"** name=**"itlb"**>  <param name=**"number\_entries"** value=**"config.system.cpu.itb.size"**/>  <stat name=**"total\_accesses"** value=**"stats.system.cpu.itb\_walker\_cache.tags.tag\_accesses"**/>  <stat name=**"total\_misses"** value=**"stats.system.cpu.itb\_walker\_cache.no\_allocate\_misses"**/>  <stat name=**"conflicts"** value=**"0"**/>  <!-- there is no write requests to itlb although writes happen to itlb after miss,  which is actually a replacement -->  </component>  <component id=**"system.core0.icache"** name=**"icache"**>  <!-- there is no write requests to itlb although writes happen to it after miss,  which is actually a replacement -->  <param name=**"icache\_config"** value=**"config.system.cpu.icache.size,config.system.cpu.icache.tags.block\_size,config.system.cpu.icache.assoc,1,1,config.system.cpu.icache.response\_latency,config.system.cpu.icache.tags.block\_size,0"**/>  <!-- the parameters are capacity,block\_width, associativity, bank, throughput w.r.t. core clock, latency w.r.t. core clock,output\_width, cache policy, -->  <!-- cache\_policy;//0 no write or write-though with non-write allocate;1 write-back with write-allocate -->  <param name=**"buffer\_sizes"** value=**"config.system.cpu.icache.mshrs,config.system.cpu.icache.mshrs,config.system.cpu.icache.mshrs,config.system.cpu.icache.mshrs"**/>  <!-- cache controller buffer sizes: miss\_buffer\_size(MSHR),fill\_buffer\_size,prefetch\_buffer\_size,wb\_buffer\_size-->  <stat name=**"read\_accesses"** value=**"stats.system.cpu.icache.ReadReq\_accesses::total"**/>  <stat name=**"read\_misses"** value=**"stats.system.cpu.icache.ReadReq\_misses::total"**/>  <stat name=**"conflicts"** value=**"stats.system.cpu.icache.tags.replacements"**/>  </component>  <component id=**"system.core0.dtlb"** name=**"dtlb"**>  <param name=**"number\_entries"** value=**"config.system.cpu.dtb.size"**/><!--dual threads-->  <stat name=**"total\_accesses"** value=**"stats.system.cpu.dtb\_walker\_cache.tags.data\_accesses"**/>  <stat name=**"total\_misses"** value=**"stats.system.cpu.dtb\_walker\_cache.no\_allocate\_misses"**/>  <stat name=**"conflicts"** value=**"0"**/>  </component>  <component id=**"system.core0.dcache"** name=**"dcache"**>  <!-- all the buffer related are optional -->  <param name=**"dcache\_config"** value=**"config.system.cpu.dcache.size,config.system.cpu.dcache.tags.block\_size,config.system.cpu.dcache.assoc,1,1,config.system.cpu.dcache.response\_latency,config.system.cpu.dcache.tags.block\_size,0"**/>  <param name=**"buffer\_sizes"** value=**"config.system.cpu.dcache.mshrs,config.system.cpu.dcache.mshrs,config.system.cpu.dcache.mshrs,config.system.cpu.dcache.mshrs"**/>  <!-- cache controller buffer sizes: miss\_buffer\_size(MSHR),fill\_buffer\_size,prefetch\_buffer\_size,wb\_buffer\_size-->  <stat name=**"read\_accesses"** value=**"stats.system.cpu.dcache.ReadReq\_accesses::total"**/>  <stat name=**"write\_accesses"** value=**"stats.system.cpu.dcache.WriteReq\_accesses::total"**/>  <stat name=**"read\_misses"** value=**"stats.system.cpu.dcache.ReadReq\_misses::total"**/>  <stat name=**"write\_misses"** value=**"stats.system.cpu.dcache.WriteReq\_misses::total"**/>  <stat name=**"conflicts"** value=**"stats.system.cpu.dcache.tags.replacements"**/>  </component>  <param name=**"number\_of\_BTB"** value=**"2"**/>  <component id=**"system.core0.BTB"** name=**"BTB"**>  <!-- all the buffer related are optional -->  <param name=**"BTB\_config"** value=**"5120,4,2,1, 1,3"**/><!--should be 4096 + 1024 -->  <!-- the parameters are capacity,block\_width,associativity,bank, throughput w.r.t. core clock, latency w.r.t. core clock,-->  <stat name=**"read\_accesses"** value=**"stats.system.cpu.branchPred.BTBLookups"**/><!--See IFU code for guideline -->  <stat name=**"write\_accesses"** value=**"stats.system.cpu.commit.branches"**/>  </component>  </component>  <component id=**"system.L1Directory0"** name=**"L1Directory0"**>  <param name=**"Directory\_type"** value=**"0"**/>  <!--0 cam based shadowed tag. 1 directory cache -->  <param name=**"Dir\_config"** value=**"4096,2,0,1,100,100, 8"**/>  <!-- the parameters are capacity,block\_width, associativity,bank, throughput w.r.t. core clock, latency w.r.t. core clock,-->  <param name=**"buffer\_sizes"** value=**"8, 8, 8, 8"**/>  <!-- all the buffer related are optional -->  <param name=**"clockrate"** value=**"3400"**/>  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"ports"** value=**"1,1,1"**/>  <!-- number of r, w, and rw search ports -->  <param name=**"device\_type"** value=**"0"**/>  <!-- although there are multiple access types,  Performance simulator needs to cast them into reads or writes  e.g. the invalidates can be considered as writes -->  <stat name=**"read\_accesses"** value=**"800000"**/>  <stat name=**"write\_accesses"** value=**"27276"**/>  <stat name=**"read\_misses"** value=**"1632"**/>  <stat name=**"write\_misses"** value=**"183"**/>  <stat name=**"conflicts"** value=**"20"**/>  </component>  <component id=**"system.L2Directory0"** name=**"L2Directory0"**>  <param name=**"Directory\_type"** value=**"1"**/>  <!--0 cam based shadowed tag. 1 directory cache -->  <param name=**"Dir\_config"** value=**"1048576,16,16,1,2, 100"**/>  <!-- the parameters are capacity,block\_width, associativity,bank, throughput w.r.t. core clock, latency w.r.t. core clock,-->  <param name=**"buffer\_sizes"** value=**"8, 8, 8, 8"**/>  <!-- all the buffer related are optional -->  <param name=**"clockrate"** value=**"3400"**/>  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"ports"** value=**"1,1,1"**/>  <!-- number of r, w, and rw search ports -->  <param name=**"device\_type"** value=**"0"**/>  <!-- altough there are multiple access types,  Performance simulator needs to cast them into reads or writes  e.g. the invalidates can be considered as writes -->  <stat name=**"read\_accesses"** value=**"58824"**/>  <stat name=**"write\_accesses"** value=**"27276"**/>  <stat name=**"read\_misses"** value=**"1632"**/>  <stat name=**"write\_misses"** value=**"183"**/>  <stat name=**"conflicts"** value=**"100"**/>  </component>  <component id=**"system.L20"** name=**"L20"**>  <!-- all the buffer related are optional -->  <param name=**"L2\_config"** value=**"1048576,32, 8, 8, 8, 23, 32, 1"**/>  <!-- the parameters are capacity,block\_width, associativity, bank, throughput w.r.t. core clock, latency w.r.t. core clock,output\_width, cache policy -->  <param name=**"buffer\_sizes"** value=**"16, 16, 16, 16"**/>  <!-- cache controller buffer sizes: miss\_buffer\_size(MSHR),fill\_buffer\_size,prefetch\_buffer\_size,wb\_buffer\_size-->  <param name=**"clockrate"** value=**"3400"**/>  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"ports"** value=**"1,1,1"**/>  <!-- number of r, w, and rw ports -->  <param name=**"device\_type"** value=**"0"**/>  <!--<stat name="read\_accesses" value="stats.system.l2.ReadReq\_accesses"/>-->  <!--<stat name="write\_accesses" value="stats.system.l2.ReadExReq\_accesses"/>-->  <!--<stat name="read\_misses" value="stats.system.l2.ReadReq\_misses"/>-->  <!--<stat name="write\_misses" value="stats.system.l2.ReadExReq\_misses"/>-->  <!--<stat name="conflicts" value="stats.system.l2.replacements"/> -->  <stat name=**"duty\_cycle"** value=**"0.5"**/>  </component>    <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->  <component id=**"system.L30"** name=**"L30"**>  <param name=**"L3\_config"** value=**"16777216,64,16, 16, 16, 100,1"**/>  <!-- the parameters are capacity,block\_width, associativity,bank, throughput w.r.t. core clock, latency w.r.t. core clock,-->  <param name=**"clockrate"** value=**"850"**/>  <param name=**"ports"** value=**"1,1,1"**/>  <!-- number of r, w, and rw ports -->  <param name=**"device\_type"** value=**"0"**/>  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"buffer\_sizes"** value=**"16, 16, 16, 16"**/>  <!-- cache controller buffer sizes: miss\_buffer\_size(MSHR),fill\_buffer\_size,prefetch\_buffer\_size,wb\_buffer\_size-->  <stat name=**"read\_accesses"** value=**"11824"**/>  <stat name=**"write\_accesses"** value=**"11276"**/>  <stat name=**"read\_misses"** value=**"1632"**/>  <stat name=**"write\_misses"** value=**"183"**/>  <stat name=**"conflicts"** value=**"0"**/>  <stat name=**"duty\_cycle"** value=**"1"**/>  </component>  <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->  <component id=**"system.NoC0"** name=**"noc0"**>  <param name=**"clockrate"** value=**"3400"**/>  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"type"** value=**"0"**/>  <!--0:bus, 1:NoC , for bus no matter how many nodes sharing the bus  at each time only one node can send req -->  <param name=**"horizontal\_nodes"** value=**"1"**/>  <param name=**"vertical\_nodes"** value=**"1"**/>  <param name=**"has\_global\_link"** value=**"0"**/>  <!-- 1 has global link, 0 does not have global link -->  <param name=**"link\_throughput"** value=**"1"**/><!--w.r.t clock -->  <param name=**"link\_latency"** value=**"1"**/><!--w.r.t clock -->  <!-- throughput >= latency -->  <!-- Router architecture -->  <param name=**"input\_ports"** value=**"1"**/>  <param name=**"output\_ports"** value=**"1"**/>  <!-- For bus the I/O ports should be 1 -->  <param name=**"flit\_bits"** value=**"256"**/>  <param name=**"chip\_coverage"** value=**"1"**/>  <!-- When multiple NOC present, one NOC will cover part of the whole chip.  chip\_coverage <=1 -->  <param name=**"link\_routing\_over\_percentage"** value=**"0.5"**/>  <!-- Links can route over other components or occupy whole area.  by default, 50% of the NoC global links routes over other  components -->  <stat name=**"total\_accesses"** value=**"100000"**/>  <!-- This is the number of total accesses within the whole network not for each router -->  <stat name=**"duty\_cycle"** value=**"1"**/>  </component>  <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->  <component id=**"system.mc"** name=**"mc"**>  <!-- Memory controllers are for DDR(2,3...) DIMMs -->  <!-- current version of McPAT uses published values for base parameters of memory controller  improvements on MC will be added in later versions. -->  <param name=**"type"** value=**"0"**/><!-- 1: low power; 0 high performance -->  <param name=**"mc\_clock"** value=**"200"**/><!--DIMM IO bus clock rate MHz-->  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"peak\_transfer\_rate"** value=**"3200"**/><!--MB/S-->  <param name=**"block\_size"** value=**"64"**/><!--B-->  <param name=**"number\_mcs"** value=**"0"**/>  <!-- current McPAT only supports homogeneous memory controllers -->  <param name=**"memory\_channels\_per\_mc"** value=**"1"**/>  <param name=**"number\_ranks"** value=**"2"**/>  <param name=**"withPHY"** value=**"0"**/>  <!-- # of ranks of each channel-->  <param name=**"req\_window\_size\_per\_channel"** value=**"32"**/>  <param name=**"IO\_buffer\_size\_per\_channel"** value=**"32"**/>  <param name=**"databus\_width"** value=**"128"**/>  <param name=**"addressbus\_width"** value=**"51"**/>  <!-- McPAT will add the control bus width to the address bus width automatically -->  <stat name=**"memory\_accesses"** value=**"33333"**/>  <stat name=**"memory\_reads"** value=**"16667"**/>  <stat name=**"memory\_writes"** value=**"16667"**/>  <!-- McPAT does not track individual mc, instead, it takes the total accesses and calculate  the average power per MC or per channel. This is sufficient for most application.  Further track down can be easily added in later versions. -->  </component>  <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->  <component id=**"system.niu"** name=**"niu"**>  <!-- On chip 10Gb Ethernet NIC, including XAUI Phy and MAC controller -->  <!-- For a minimum IP packet size of 84B at 10Gb/s, a new packet arrives every 67.2ns.  the low bound of clock rate of a 10Gb MAC is 150Mhz -->  <param name=**"type"** value=**"0"**/><!-- 1: low power; 0 high performance -->  <param name=**"clockrate"** value=**"350"**/>  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"number\_units"** value=**"0"**/><!-- unlike PCIe and memory controllers, each Ethernet controller only have one port -->  <stat name=**"duty\_cycle"** value=**"1.0"**/><!-- achievable max load <= 1.0 -->  <stat name=**"total\_load\_perc"** value=**"0.7"**/><!-- ratio of total achieved load to total achieve-able bandwidth -->  <!-- McPAT does not track individual nic, instead, it takes the total accesses and calculate  the average power per nic or per channel. This is sufficient for most application. -->  </component>  <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->  <component id=**"system.pcie"** name=**"pcie"**>  <!-- On chip PCIe controller, including Phy-->  <!-- For a minimum PCIe packet size of 84B at 8Gb/s per lane (PCIe 3.0), a new packet arrives every 84ns.  the low bound of clock rate of a PCIe per lane logic is 120Mhz -->  <param name=**"type"** value=**"0"**/><!-- 1: low power; 0 high performance -->  <param name=**"withPHY"** value=**"1"**/>  <param name=**"clockrate"** value=**"350"**/>  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <param name=**"number\_units"** value=**"0"**/>  <param name=**"num\_channels"** value=**"8"**/><!-- 2 ,4 ,8 ,16 ,32 -->  <stat name=**"duty\_cycle"** value=**"1.0"**/><!-- achievable max load <= 1.0 -->  <stat name=**"total\_load\_perc"** value=**"0.7"**/><!-- Percentage of total achieved load to total achieve-able bandwidth -->  <!-- McPAT does not track individual pcie controllers, instead, it takes the total accesses and calculate  the average power per pcie controller or per channel. This is sufficient for most application. -->  </component>  <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->  <component id=**"system.flashc"** name=**"flashc"**>  <param name=**"number\_flashcs"** value=**"0"**/>  <param name=**"type"** value=**"1"**/><!-- 1: low power; 0 high performance -->  <param name=**"withPHY"** value=**"1"**/>  <param name=**"peak\_transfer\_rate"** value=**"200"**/><!--Per controller sustain-able peak rate MB/S -->  <param name=**"vdd"** value=**"0"**/><!-- 0 means using ITRS default vdd -->  <param name=**"power\_gating\_vcc"** value=**"-1"**/><!-- "-1" means using default power gating virtual power supply voltage constrained by technology and computed automatically -->  <stat name=**"duty\_cycle"** value=**"1.0"**/><!-- achievable max load <= 1.0 -->  <stat name=**"total\_load\_perc"** value=**"0.7"**/><!-- Percentage of total achieved load to total achieve-able bandwidth -->  <!-- McPAT does not track individual flash controller, instead, it takes the total accesses and calculate  the average power per fc or per channel. This is sufficient for most application -->  </component>  <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->  </component>  </component> |