目录

[1. Core类介绍 2](#_Toc532840793)

[2. Core类声明 3](#_Toc532840794)

[3. 定义Core的对象 4](#_Toc532840795)

[4. Core的构造函数 5](#_Toc532840796)

[5. ComputerEnergy函数 6](#_Toc532840797)

# Core类介绍

* Core中定义的对象

|  |  |  |  |
| --- | --- | --- | --- |
| Class | 内部定义的对象 | | |
| Core | ifu (InstFetchU)  exu(EXECU)  undiffCore(UndiffCore) | lsu(LoadStoreU)  rnu(RENAMINGU)  l2cache(SharedCache) | mmu(MemManU)  corepipe(Pipeline) |
| InstFetchU | cache\_p(Cache\_policy)  BTB(ArrayST)  ID\_operand(inst\_decoder) | icache(InstCache)  BPT(BranchPredictor)  ID\_misc(inst\_decoder) | IB(ArrayST)  ID\_inst(inst\_decoder) |
| BranchPredictor | globalBPT(ArrayST)  L1\_localBPT(ArrayST) | loaclBPT(ArrayST)  L2\_localBPT(ArrayST) | choose(ArrayST) RAS(ArrayST) |
| LoadStoreU | dcache(DataCache) | LSQ(ArrayST) | LoadQ(ArrayST) |
| MemManU | itlb(ArrayST) | dtlb(ArrayST) |  |
| EXECU | rfu(RegFU)  fp\_u(FuctionalUnit)  int\_bypass  (interconnect)  intTagBypass  (interconnect) | scheu(SchedulerU)  exeu(FuctionalUnit)  int\_mul\_bypass  (interconnect)  intTag\_mul\_Bypass  (interconnect) | mul(FuctionalUnit)  fp\_bypass  (interconnect)  fpTagBypass  (interconnect) |
| RegFU | IRF(ArrayST) | FRF(ArrayST) | RFWIN(ArrayST) |
| SchedulerU | int\_inst\_window  (ArrayST)  instruction\_selection  (selection\_logic) | fp\_inst\_window  (ArrayST) | ROB  (ArrayST) |
| RENAMING | iFRAT(ArrayST)  fFRAT(ArrayST)  idcl(dep\_resource  \_conflict\_check) | iRRAT(ArrayST)  FRRAT(ArrayST)  fdcl(dep\_resource  \_conflict\_check) | ifreeL(ArrayST)  ffreeL(ArrayST) |
| 未在core.h中定义的类 | | | |
| FuctionalUnit | logic.h 189 | local\_result(uca\_org\_t)  l\_ip(InputParameter)  ithCore(int)  FU\_height,clockRate  executionTime(double)  num\_fu(double)  energy, base\_energy,  per\_access\_energy, leakage, gate\_leakage  (double) | coredynp  (CoreDynParam)  tdp\_stats(statsDef)  rtp\_stats(statsDef)  stats\_t(statsDef)  power\_t(powerDef)  fu\_type(FU\_type) |
| inst\_decoder | logic.h 95 | local\_result(uca\_org\_t)  l\_ip(InputParameter)  opcode\_length(int)  num\_decoders(int)  num\_decoder\_segments  (int)  final\_dec(Decoder)  pre\_dec(Predec) | num\_decoded\_signals  (int)  device\_ty(Device\_ty)  core\_ty(Core\_type)  x86(bool)  tdp\_stats(statsDef)  rtp\_stats(statsDef)  stats\_t(statsDef)  power\_t(powerDef) |
| dep\_resource  \_conflict\_check | logic.h 72 | local\_result(uca\_org\_t)  WNORn, WNORp, Wevalinvp, Wevalinvn, Wcompn, Wcompp, Wcomppreequ(double)  compare\_bits(int) | l\_ip(InputParameter)  coredynp  (CoreDynParam)  tdp\_stats(statsDef)  rtp\_stats(statsDef)  stats\_t(statsDef)  power\_t(powerDef) |
| selection\_logic | logic.h 52 | local\_result(uca\_org\_t)  win\_entries(int)  issue\_width(int)  num\_threads(int) | device\_ty(Device\_ty)  core\_ty(Core\_type)  l\_ip(InputParameter) |
| interconnect | interconnect.h 48 |  |  |
| FU\_type | basic\_components.h 41 | enum FPU, ALU, MUL |  |
| Renaming\_type | basic\_components.h 47 | enum RAMbased | CAMbased |
| Scheduler\_type | basic\_components.h 52 | enum PhysicalRegFile | ReservationStation |
| cache\_level | basic\_components.h 57 | enum L2, L3 | L1Directory  L2Directory |
| Cache\_policy | basic\_components.h 77 | enum Write\_through,0 | Write\_back,1 |
| Device\_ty | basic\_components.h 82 | enum Core\_device  LLC\_device | Uncore\_device |
| Core\_type | basic\_components.h 88 | Enum OOO, InOrder |  |
| statsComponents | basic\_components.h 93 | Access, hit, miss  (double) |  |
| InstCache | array.h 70 | caches(ArrayST)  missb(ArrayST)  power\_t(powerDef) | Ifb(ArrayST)  prefetchb(ArrayST) |
| DataCache | array.h 90 | wbb(ArrayST) | 继承于InstCache |
| SharedCache | sharedcache.h 42 | l\_ip(InputParameter)  ithCore(int)  unicache(DataCache)  cachep  (CacheDynParam) | homenode\_tdp\_stats  homenode\_rtp\_stats  homenode\_stats\_t  (statsDef)  dir\_overhead(double)  executionTime  scktRatio(double) |
| Pipeline | logic.h 151 | local\_result(uca\_org\_t)  l\_ip(InputParameter)  is\_core\_pipeline, is\_default  process\_ind(bool)  load\_per\_pipeline\_stage  (double) | coredynp  (CoreDynParam)  device\_ty(Device\_ty)  num\_piperegs  WNANDn, WNANDp(double) |
| UndiffCore | logic.h 213 | local\_result(uca\_org\_t)  l\_ip(InputParameter)  ithCore(int)  clockRate  executionTime(double)  core\_ty(Core\_type)  opt\_performance, embedded(bool) | scktRatio, chip\_PR\_overhead, macro\_PR\_overhead  pipeline\_stage,  num\_hthreads,  issue\_width  (double) |
| ArrayST | array.h 45 | l\_ip(InputParameter)  local\_result(uca\_org\_t)  name(string)  opt\_local(bool)  power\_t(powerDef) | device\_ty(Device\_ty)  core\_ty(Core\_type)  tdp\_stats(statsDef)  rtp\_stats(statsDef)  stats\_t(statsDef) |

# Core类声明

|  |
| --- |
|  |

# 定义Core的对象

* Core在Core的构造函数中定义，只传入两个参数，其余均使用默认值

|  |
| --- |
|  |

* Core中函数使用到的外部参数，大多数在估算流水线需要增加的位数时使用

# Core的构造函数

|  |
| --- |
|  |

# ComputerEnergy函数

* ComputerEnergy函数包括一个参数is\_tdp，bool类型
* 当is\_tdp为true时，power = energy\_per\_cycle\* clock\_rate。在该函数中只计算得到每个周期该组件会消耗的能量energy\_per\_cycle，在displayEnergy函数中，将会使用该公式计算得到峰值power，即每个周期都在工作时的功率。此时的计算结果保存在power中
* 当is\_tdp为false时，power = total energy / Total execution time。同样该函数中只计算该组件在整个执行过程中会消耗的所有能量（使用组件的访问次数等计算），在displayEnergy函数中，将其除以整体的执行时间（cycle count / clock rate），得到运行时的动态功耗。此时的计算结果保存在rt\_power中

|  |
| --- |
|  |