1. interface\_ip是许多McPAT的类中定义的对象，主要用处是用于在McPAT和cacti之间传递参数，然后将某些计算过程调用Cacti中的函数完成
2. Cacti： Architecture-Level modeling for sram-based structures with advanced leakage reduction techniques
3. interface\_ip的类是cacti文件夹中的cacti\_interface.h文件的117行定义的：InputParameter类
4. InputParameter类中的方法

|  |  |
| --- | --- |
| 名称 | 作用 |
| InputParameter(); | 构造函数，初始化了一些参数  array\_power\_gated(false),  bitline\_floating(false),  wl\_power\_gated(false),  cl\_power\_gated(false),  interconect\_power\_gated(false),  power\_gating(false),  perfloss(0.01),  cl\_vertical (true),  long\_channel\_device(false)  dvs\_voltage = std::vector<double>(0); |
| void parse\_cfg(const string & infile); | 解析Catic原本的输入文件，例如cache.cfg |
| bool error\_checking(); | 判断当前的输入参数是否相互之间问题 |
| void display\_ip(); | 显示InputParameter中的参数 |

1. InputParameter中的属性值和解释

|  |  |
| --- | --- |
| 名称 | 作用 |
| unsigned int cache\_sz | 单位byte，整个cache的大小 |
| unsigned int line\_sz | 单位byte，每个cache line的大小 |
| unsigned int assoc | cache的相连度 |
| unsigned int nbanks | cache的bank个数 |
| unsigned int out\_w | 单位bit，input/output的bus宽度，cache line的宽度 |
| bool specific\_tag | 用于指定tag的宽度是否需要cacti自己计算  为false，则cacti计算宽度  为true，则直接外部指令，不再计算 |
| unsigned int tag\_w | 单位bit，tag的宽度 |
| unsigned int access\_mode | 访问模式：  fast(2): 数据和tag同时访问  sequential(1)：先访问tag，然后再访问data  normal(0): 数据查找和tag访问同时进行，然后通过tag的信号，确定数据 |
| unsigned int obj\_func\_dyn\_energy  obj\_func\_dyn\_power  obj\_func\_leak\_power  obj\_func\_cycle\_t | 未找到在cacti的使用 |
| double F\_sz\_nm | 单位nm，工艺尺寸 |
| double F\_sz\_um | 单位um，工艺尺寸，F\_sz\_nm=1000\*F\_sz\_um |
| bool specific\_hp\_vdd  double hp\_Vdd | 用于指定hp vdd是否使用ITRS预测的值  false为不指定，true是指定  HP：高性能类型 |
| bool specific\_lstp\_vdd  double lstp\_Vdd | 类似于LSTP：低待机功耗 |
| bool specific\_lop\_vdd  double lop\_Vdd | 低工作功耗 |
| bool specific\_vcc\_min  double user\_defined\_vcc\_min | 用于决定是否使用用户指定的vcc\_min来实现功率门控，false不使用，true使用 |
| bool user\_defined\_vcc\_underflow | 标志，当用户指令的vcc太低以至于无法保持电路稳定时，给出指示表示 |
| unsigned int num\_rw\_ports | 读写端口数 |
| unsigned int num\_rd\_ports | 独占的读端口数 |
| unsigned int num\_wr\_ports | 独占的写端口数 |
| unsigned int num\_se\_rd\_ports | 单端读端口数，一般为0 (single ended read ports) |
| unsigned int num\_search\_ports | CAM存储类型中的搜索端口的数目 |
| bool is\_main\_mem  bool is\_cache  bool pure\_ram  bool pure\_cam | cache type: cache, ram, cam, main memory  是否是主存，以决定是否按照主存建模  array的类型是否是cache类型的  array的类型是暂存器，类似于寄存器  array的类型是CAM |
| bool rpters\_in\_htree | 解析函数中，设置为true |
| unsigned int ver\_htree\_wires\_over\_array  unsigned int broadcast\_addr\_din\_over  \_ver\_htrees | 在mcpat中没有看到赋值的地方 |
| unsigned int temp | 工作温度 |
| unsigned int ram\_cell\_tech\_type | itrs-hp(0),lstp(1),lop(2),lp\_dram(3),comm-dram(4) |
| unsigned int peri\_global\_tech\_type | 未设置 |
| unsigned int data\_arr\_ram\_cell\_tech\_type  unsigned int tag\_arr\_ram\_cell\_tech\_type | itrs-hp(0),lstp(1),lop(2),lp\_dram(3),comm-dram(4) |
| unsigned int data\_arr\_peri\_global\_tech\_type  unsigned int tag\_arr\_peri\_global\_tech\_type | Data array peripheral type(数据阵列外设类型)  itrs-hp(0),lstp(1),lop(2) |
| unsigned int burst\_len  unsigned int int\_prefetch\_w  unsigned int page\_sz\_bits | 这三个参数只对主存意义  突发长度：连续传输的周期数就是突发长度  内部预取宽度  单位bit，页面大小 |
| unsigned int ic\_proj\_type | 互连规划设计的类型: aggressive(0), conservative(1) |
| unsigned int wire\_is\_mat\_type | mat内部的连线类型：global(2), local(0), else(1) |
| unsigned int wire\_os\_mat\_type | mat外部的连线类型，global(2), else(1) |
| enum Wire\_type wt | 导线的类型  Global : gloabl wires with repeaters  Global\_5 : 5% delay penalty  Global\_10 : 10% delay penalty  Global\_20 : 20% delay penalty  Global\_30 : 30% delay penalty  Low\_swing : differential low power wires with high area overhead  Semi\_global : mid-level wires with repeaters  Transmission : tranmission lines with high area overhead  Optical : optical wires  Invalid\_wtype |
| int force\_wiretype | 除了当wt为global时为0，其余均为1 |
| bool print\_input\_args | 是否输出输入的参数信息 |
| unsigned int nuca\_cache\_sz | 未设置 |
| int ndbl, ndwl, nspd, ndsam1, ndsam2, ndcm  bool force\_cache\_config | 强迫使用用户提供的一些cache优化的参数  决定是否使用用户提供的参数 |
| int cache\_level | L2(0),L3(1) |
| int cores | 核数，不能够大于16 |
| int nuca\_bank\_count | NUCA的bank数量 |
| int force\_nuca\_bank | 0/1(nuca\_bank\_count不为零，则都设置为1) |
| int delay\_wt, dynamic\_power\_wt, leakage\_power\_wt,  cycle\_time\_wt, area\_wt | weight delay, dynamic power, leakage power, cycle time, area |
| int delay\_wt\_nuca, dynamic\_power\_wt\_nuca, leakage\_power\_wt\_nuca,  cycle\_time\_wt\_nuca, area\_wt\_nuca | NUCA的weight delay, dynamic power, leakage power, cycle time, area |
| int delay\_dev, dynamic\_power\_dev, leakage\_power\_dev,  cycle\_time\_dev, area\_dev | deviate背离，  delay, dynamic power, leakage power, cycle time, area |
| int delay\_dev\_nuca, dynamic\_power\_dev\_nuca, leakage\_power\_dev\_nuca,  cycle\_time\_dev\_nuca, area\_dev\_nuca | NUCA的deviate，delay, dynamic power, leakage power, cycle time, area |
| int ed | ed=1(ED),ed=2(ED^2)，优化的目标参数 |
| int nuca | 0(UCA), 1(NUCA) |
| bool fast\_access | 在error\_checking函数中，根据access\_mode设置  access\_mode=normal/seq, fast\_access=false  access\_mode=fast, fast\_access=true |
| unsigned int block\_sz | 在error\_checking函数中设置  block\_sz=line\_sz |
| unsigned int tag\_assoc | 在error\_checking函数中设置  如果是全相连，tag\_assoc=cache\_sz/nbanks/line\_sz  如果不是全相连，tag\_assoc=assoc |
| unsigned int data\_assoc | 在error\_checking函数中设置  如果是全相连，data\_assoc=1  如果不是全相连并且access mode=seq, 则data\_assoc=1  否则data\_assoc=tag\_assoc |
| bool is\_seq\_acc | 在error\_checking函数中设置  如果access mode=seq,则为true，否则false |
| bool fully\_assoc | 在error\_checking函数中设置  如果是cache，并且assoc=0，则是全相联 |
| unsigned int nsets | 在error\_checking函数中计算得到的组数 |
| int print\_detail | Detailed(1),Concise(0) |
| bool add\_ecc\_b\_ | 是否有ECC校验码 |
| double throughput  double latency  bool pipelinable  int pipeline\_stages  int per\_stage\_vector  bool with\_clock\_grid | 设计约束的参数，cacti中未设定  流水线中每个阶段需要的vector/寄存器的数量  false：全局时钟不驱动本地终端节点 |
| bool array\_power\_gated  bool bitline\_floating  bool wl\_power\_gated  bool cl\_power\_gated  bool interconect\_power\_gated  bool cl\_vertical | 只在cacti中进行了设置  是否支持array的功率门控  是否支持位线浮动  是否支持wordline功率门控  是否支持cacheline功率门控  是否支持互连结构功率门控  CLDriver vertical |
| bool power\_gating  double perfloss | 是否支持门控技术  门控所带来的性能损失 |
| std::vector<double> dvs\_voltage | 动态电压调节，假设无论每个阵列的设备类型如何，都将相同的电压应用于标签和数据阵列 |
| bool long\_channel\_device | 是否使用长（10％）通道设备（true/false，当false时,假设90％的设备（非时间关键）可以是长通道设备） |