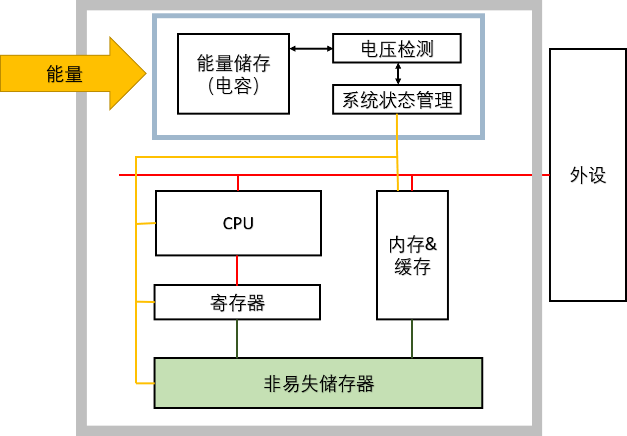
背景介绍

NVP（Non-Volatile Processor）为非易失处理器的简称，非易失处理器相比传统处理器多了能量采集和储存模块、电压检测模块、系统状态管理模块、非易失储存器模块，本作业所使用的NVP模型如下：

非易失处理器的电源供给是外界环境， 外界能量被能量采集模块收集到系统的能量缓存（电容）中，电压检测模块随时检测电容的电压，如果电压检测模块发现电压过低有关机风险，则通知系统中的寄存器和内存备份当前状态到非易失储存器中（可以认为是不会随着断电丢失内容的内存，类似硬盘，但远比硬盘块，读写速度和内存同等级），并通知系统进入关机状态，关机时当电压检测模块发现电容电压超过一个阈值后，将通知寄存器和内存从非易失储存器中恢复原有数据，并让系统进入开机状态。

DFS（Dynamic Frequency Selection）为动态频率选择系统的简称，这种处理器会随着工作时的负载和外界能量变化而改变自身的工作频率，在非易失处理器领域，DFS处理器的意义在于，当外界能量强度不够高（但也不至于过低导致大部分时间都让非易失处理器在休眠），不足以支持固定频率的处理器长时间持续运行时，处理器会面对频繁的掉电、上电，断电重启的恢复时间会导致系统运行的效率过低，如果系统能够在能量较少时使用降低频率的方式来降低功耗，则有可能避免频繁的断电重启，从而增加完成任务的效率。

本文需要实现的动态频率系统状态转移图如下：

关机

低频率工作

高频率工作

有关Gem5-NVP仿真器的知识，请参阅github.com/zlfben/gem5/wiki页面的内容。

Gem5-NVP仿真器下载地址：https://github.com/zlfben/gem5/wiki/Getting-Started

大作业内容

为了使理解变得更加容易，本次作业中可以进行如下假定：忽视电容的电容值、电压等参数，认为电容是一个理想的能量储存器，从Energy Profile中获取的能量可以完全被储存在电容中，我们定义能量单位为“单位”，可以认为这个单位是“uW\*us”，但是这不重要。这个能量储存器的储存有一定上限，为50000单位。在这个假设下，电压检测模块直接检测电容的能量值而不是电压值。这个假定可以大大降低建模困难。

关于Energy Profile（描述从外界收集能量的文件，内容为换行分隔的浮点数）：可以自行生成一些Energy Profile，比如有一定占空比的方波，方波的低电平也不一定要是0，可以为大于0的某个数值，总之大家在此次设计中可以尽可能多的尝试DFS系统在各种外界能量条件下的行为。

必做内容：

1. （只涉及gem5）AtomicSimpleCPU在仿真运行时会在EventQueue中触发哪些事件，这些事件是在运行时由哪些函数触发的？这些事件被调用时运行的函数是什么？
2. （只涉及gem5）AtomicSimpleCPU中，访问内存的指令和不访问内存的指令是否消耗相同的时间？使用AtomicSimpleCPU任意仿真一个程序，并使用SimpleCPU这个debug flag输出单周期处理器时间信息，从输出中观察每一个指令消耗的周期。观察AtomicSimpleCPU::tick()函数，找到统计访问内存时间的部分，为什么虽然AtomicSimpleCPU计算了访存时间但是在仿真中没有体现？如果要在仿真中体现访问内存花费的时间，需要怎么做？
3. 使用gem5-nvp的默认脚本simple\_engy.py仿真任意程序，观察系统状态机是如何通知CPU模块系统发生了状态变化（需要看engy/engy\_mgmt.\*），可以添加仿真Debug Flag：EnergyMgmt并观察输出结果。需要关注EnergyMgmt如何发出这些信息，AtomicSimpleCPU如何接收这些信息
4. 修改simple\_engy.py，使用TwoThresSM状态机（参考<https://github.com/zlfben/gem5/wiki/Concepts-and-Structure#Energy-Management-Module-State-Machine>的内容，设置energy\_mgmt的state\_machine变量为TwoThresSM()）进行仿真，仿真程序可以是任意的，比如ARM编译的八皇后，为状态机设定20000单位的上阈值（开机阈值）和10000单位的下阈值（关机阈值），观察仿真结果，设定方式可以查看TwoThresSM类的python端定义。
5. 请完成上文所述DFS系统，设定为10000单位，为20000单位，你需要创建一个新的状态机模块，关于状态机模块，请参考<https://github.com/zlfben/gem5/wiki/Concepts-and-Structure#Energy-Management-Module-State-Machine>中的内容，有关如何添加新模块，请参考<https://github.com/zlfben/gem5/wiki/Example-1>中描述TwoThresSM创建的部分，当前TwoThresSM已经集成到Gem5-NVP中，不需要用户创建，你需要自行创建一个满足DFS系统要求的状态机。除此之外，还需要在AtomicSimpleCPU中编写接收到状态转移信息时的操作，在接收到响应操作时修改系统频率。为了测试上的容易，可以把高频率设定为1MHz，低频率设定为500kHz
6. 在各种energy\_prof下测试这种系统和原系统的性能（不同占空比，不同高电平和低电平），这种系统（DFS）在什么情况下的性能会有很大提高？原系统的开机阈值为20000单位，关机阈值为10000单位，频率为1MHz，其余参数二者应相同。

要求

分组：2人一组

分值：算法实现\*60% + 实验分析\*25% + 报告质量 \*15% + 创新性\*10% =110分

截止时间：第19周周日（7月2日）晚24点

报告要求：

科技论文格式，包括：摘要+系统设计+系统实现方法+实验结果与分析+总结，鼓励提出改进方案，在报告中明确出哪些是原创的方法

提交要求：报告+修改过的gem5文件+readme文件说明

提示

* 要了解Gem5的event queue
* 了解Simple CPU的工作流程
  1. <http://gem5.org/SimpleCPU>
* 了解AtomicSimpleCPU如何使用周期间隔
* 要了解Gem5模块如何建立
* 了解模块如何在Gem5-NVP下消耗能量和在收到控制信号时做出反应
* 可以只考虑CPU消耗能量
* 不需要考虑具体的备份策略，可以认为系统完全非易失
* 为了防止每次都要修改状态机，完成DFS开发后测试原有系统时可以把高频和低频都设定成1MHz，这样行为就和原有系统一致了