2016/1/1

macbook系统功耗限制内核模块

# 本文简介

本模块仅为osx系统内核驱动模块，本模块生成后，由于操作超出正常允许操作的范围以及功能，加之没有开发者证书，所以osx 10.11系统以上未关闭rootless功能会导致加载失败，关闭rootless的坏处自行决定，方法在附录中有介绍。理论上来讲win系统也可使用本代码，只是把加载与卸载部分代码自行修改下即可，至于cpu型号支持，请自行进行兼容性判断。Intel手册中可查询。

本模块未做用户与内核层通讯，至于原因，说出来汗颜，我才接触osx开发，不是很熟，就是想偷个懒，平台相关代码写起来太麻烦，有兴趣加入ui界面的，希望能秉承开源思想，主要是别嫌麻烦，放出来方便大家下，顺便求个@。

由于作者技术有限，读此文时请带着怀疑的态度去读，有错误欢迎指正，但求不撕逼。

Name：Erriy

Email：[Erriy@msn.com](mailto:Erriy@msn.com)

Git地址：<https://github.com/Erriy/cpu_power_limit>

Git库链接：https://github.com/Erriy/cpu\_power\_limit.git

# 本机测试环境简介以及免责声明

机型：a1398（2014年中）

cpu：Intel Core i7-4770HQ CPU @ 2.20GHz

系统：64bit Mac OS X 10.11.2 15C50

内核：x86\_64 Darwin 15.2.0

xcode：7.2 Build version 7C68

rootless已关闭

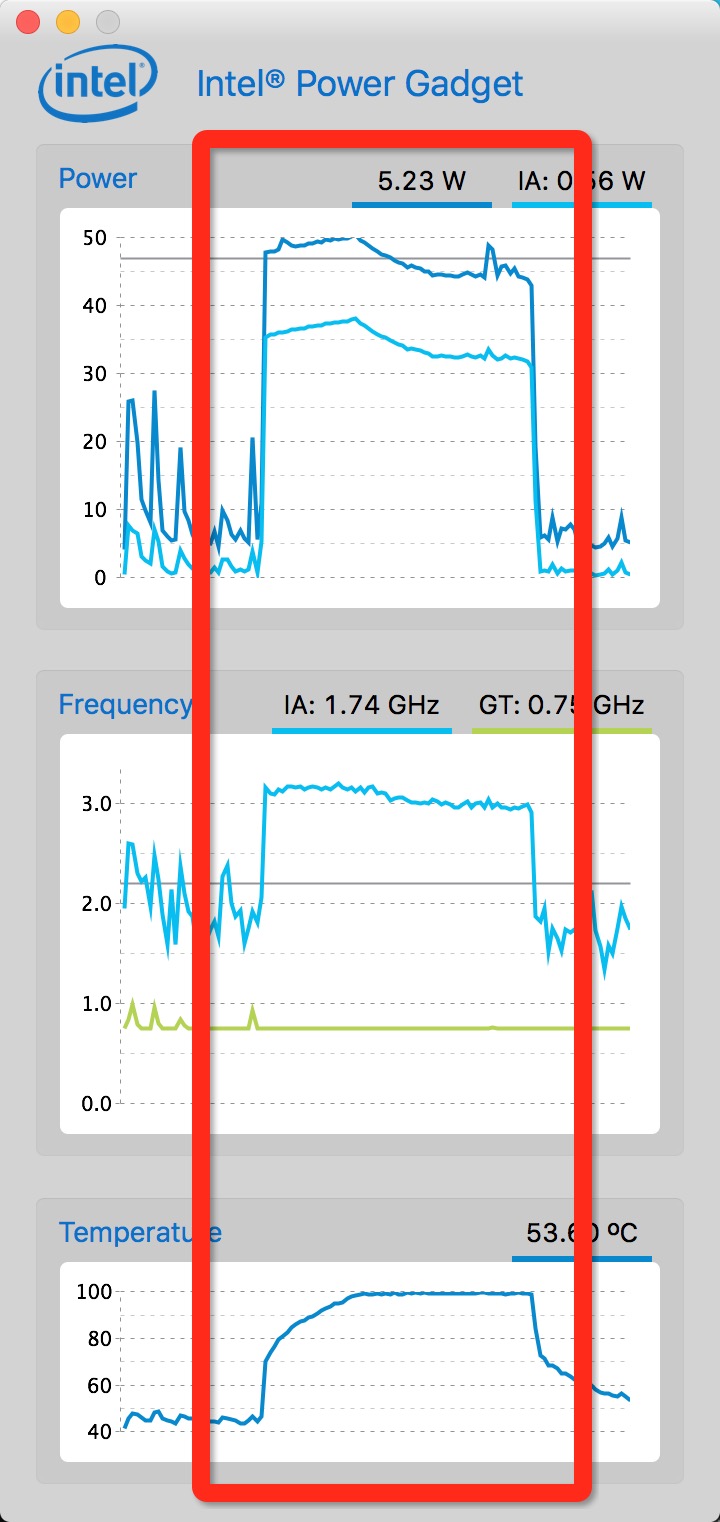
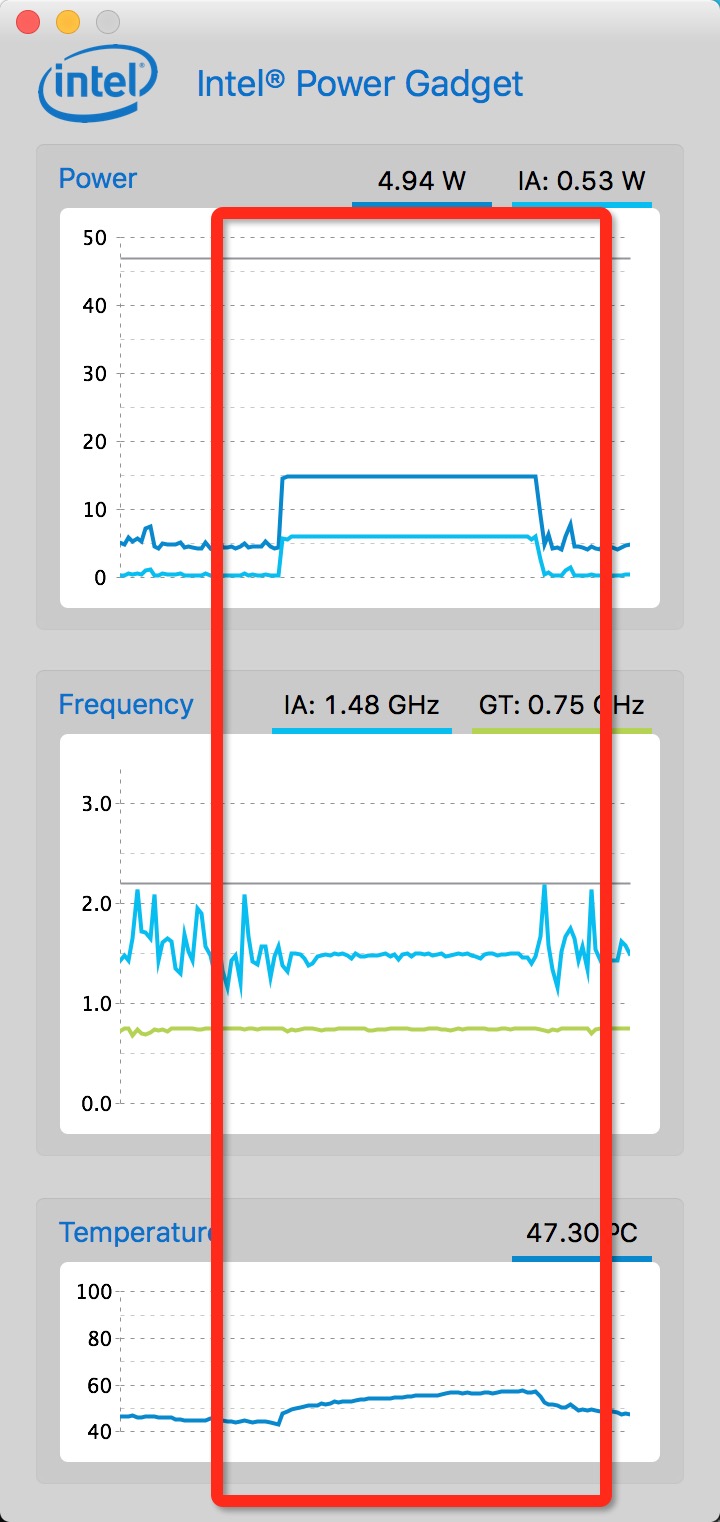
非此对应版本的环境并不保证一定运行成功，未做兼容性考虑，本文内将主要实现原理已做说明，此份代码并不保证一定在您的机器上如期运行，请根据原理自行判断是否支持。

代码使用前，请用xcode添加开发者账号。

项目目录内写了个简单的makefile，xcodebuild使用不熟，所以直接每次强制删除模块重新生成，执行测试直接make即可。

# 实际测试功耗截图

左图为系统电源管理，右图为开启功耗限制15w。

# 已实现功能

## 功耗限制

## 关闭cpu turbo boost睿频加速功能

# 模块优缺点简介

mac笔记本的电源管理非常“人性化”，系统自动控制，但是不知道正在读这篇文档的你怎么想，我的想法是这样的，作为一个开发者，大部分情况下我们不玩游戏，不做大量计算操作，对计算机性能要求很低，只需运行一些低功耗处理操作即可。而macbook自身为了系统流畅性的原因，会短时间通过cpu睿频来进行更迅速的操作，但就我个人情况来讲，此睿频加速带给我的性能提升体验，在大部分情况下并非必要，然而为了仅有的我并不需要的性能提升，牺牲的是电池续航以及整体温度的提升。

## 好处：

### 温度控制（限制20w，最高温度70度，限制15w，最高温度不到60度）

### 待机时间控制（限制20w，正常高强度使用大约5小时，限制15w，正常高强度使用大约6～7小时）

## 缺点：

### 网页视频播放性能降低（b站带弹幕测试，html5播放器以及flash播放器均超低帧数）

### 系统动画性能略卡顿（四指内抓显示lanuchpad时动画最明显）

### 其他大量数据运算性能降低

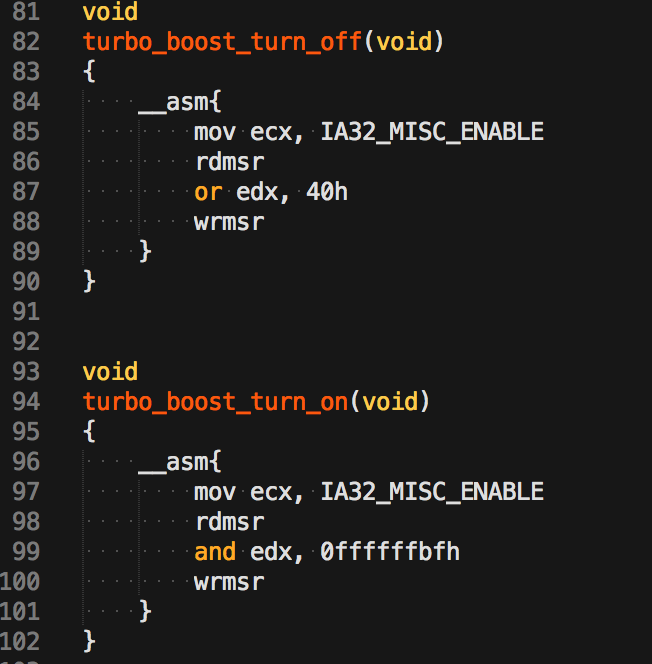
# 功能实现原理

## 关闭turbo boost

turbo boost即睿频加速功能，此功能在msr寄存器中注册名为IA32\_MISC\_ENABLE，注册地址为0x1a0的表值中的第38位决定，将其置1则禁止睿频加速。

需要注意的是msr寄存器中均为64位字节，但返回值并不是使用64位通用寄存器，而采用的高32位存在edx中，低32位存于eax中（猜测是为了32位64位通用？）。

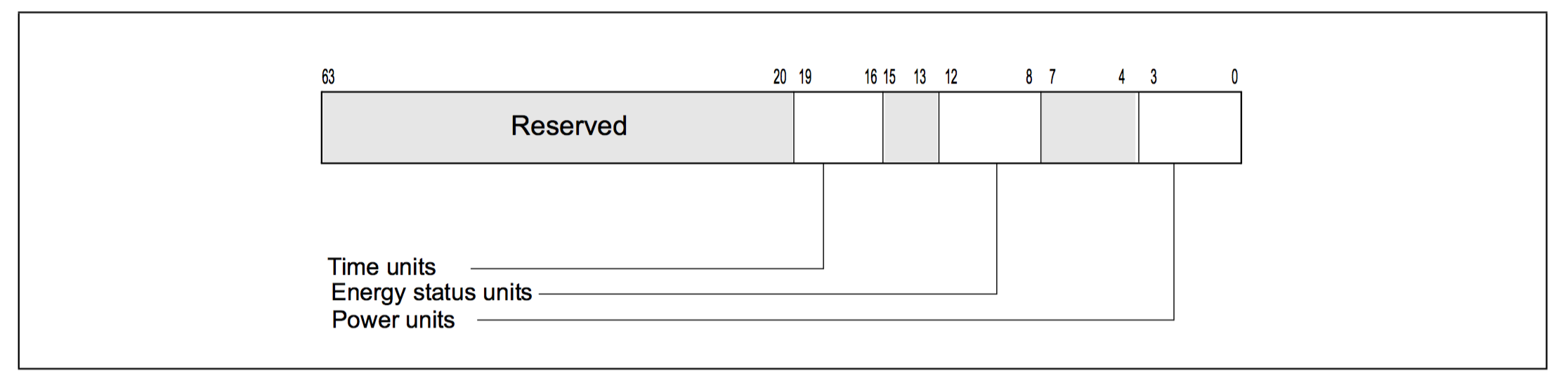
实现代码如下图所示：



## 功耗限制

功耗控制上略微麻烦那么一丢丢，从读一个表单变为读两个表单。

### MSR\_RAPL\_POWER\_UNIT



看名识功能，RAPL英文全称为Running Average Power Limit，直译则为运行时功耗平均数上限，power\_unit，能量单位，此64位值中，仅低32位中为有意义的值，分别为：

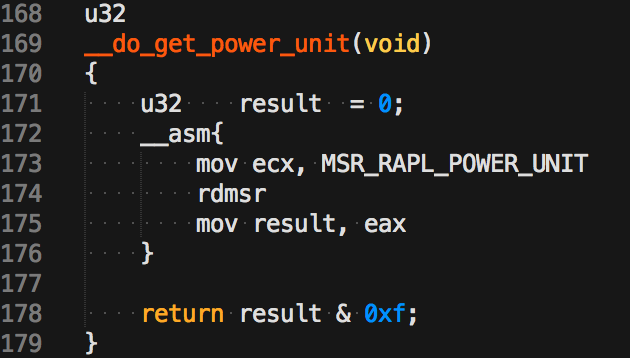
#### Power Units (bits 3:0)

本次功能实现只关注power units，单个单位并非是直接使用，而是1/2^pu，2的power units次幂分之一，单位为watts，是以下即将使用的功能单位。至于剩下的两个单位，略过不谈，有兴趣的可以去读intel软件开发手册手册，章节14.9.1，RAPL Interfaces

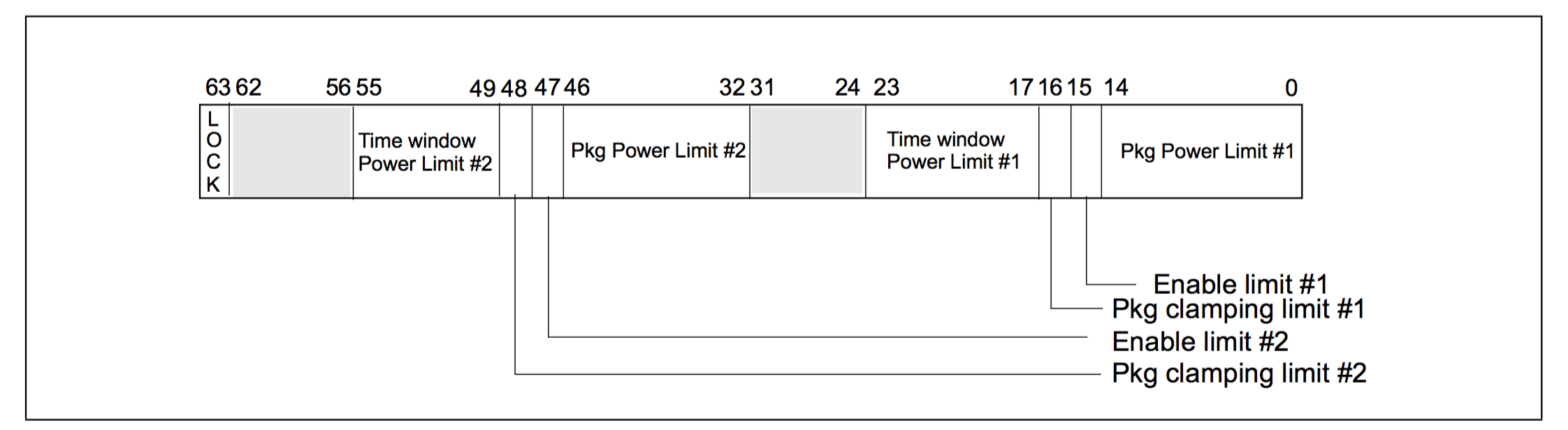
#### Energy Status Units (bits 12:8)

#### Time Units (bits 19:16)

实现代码如下图所示：



### MSR\_PKG\_POWER\_LIMIT



PKG，代表package，此处package的意思为整个cpu，包括ia（cpu部分）和gt（gpu部分），此64位数据掌管整个cpu的功耗上限，废话不多说，包含值如下：

#### Package Power Limit #1(bits 14:0)

在4.时间限制范围内的平均功耗上限，限制的功耗为此值＊power unit，此处功耗限制即为pl1\*（1/2^pu）= pl1/2^pu w

#### Enable Power Limit #1(bit 15)

打开1号功耗上限控制

#### Package Clamping Limitation #1 (bit 16)

clamping为夹紧的意思，此标志位意为严格限制，不打开此标志位则处理器可能会按照自身的建议设置进行处理（比如我设置15w最高限制，但未将此位设置为真，则cpu自动将功耗限制位20w）

#### Time Window for Power Limit #1 (bits 23:17)

1.中的时间条件，此处个人觉得不太好说，可能也是由于理解的不是很清晰。按RAPL这个全称的意思我理解为在这个值的描述时间内，平均功耗的值不能超过power limit ＃1的限制。此处并不关注此值，详情去看intel手册14.9.3，Package RAPL Domain

顺便在此解释下为什么有两套限制，我在前期的实验中进行过读取此值，1号为28秒左右，2号则为非常短暂的几百毫秒，具体数据遗忘，结合查到的[玩转BIOS设置 史上最强BIOS设置大解析 下](http://tech.hexun.com/2011-05-03/129218470.html)中的Long Duration power limit长时间功耗限制和Short Duration power limit段时间功耗限制，此处双份限制为＃1长时间的功耗限制，另一段＃2则为短时间＃2范围内可超出＃1的功耗限制的功耗限制

#### Package Power Limit #2(bits 46:32)

同1

#### Enable Power Limit #2(bit 47)

同2

#### Package Clamping Limitation #2 (bit 48)

同3

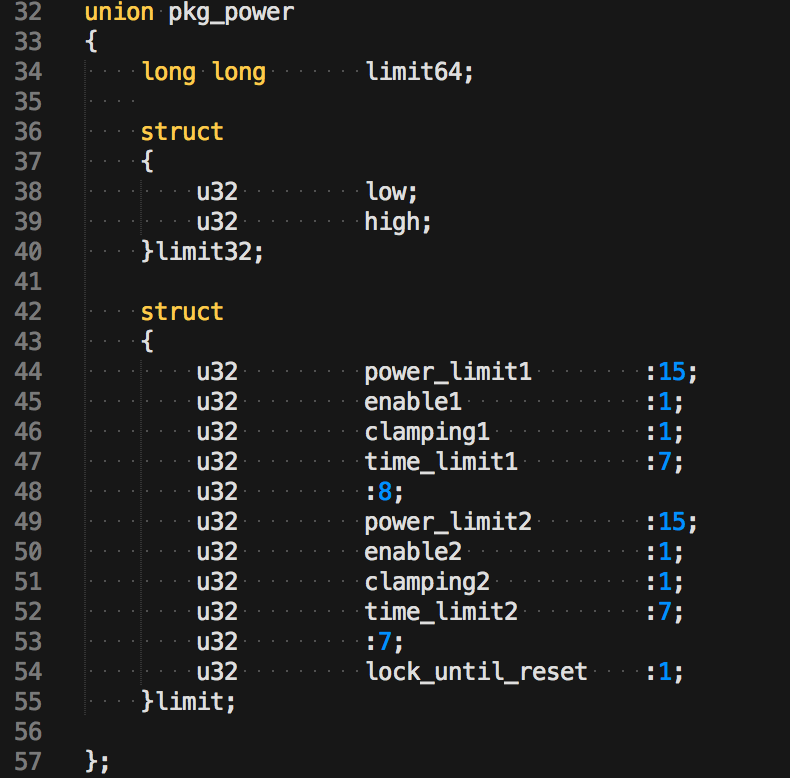
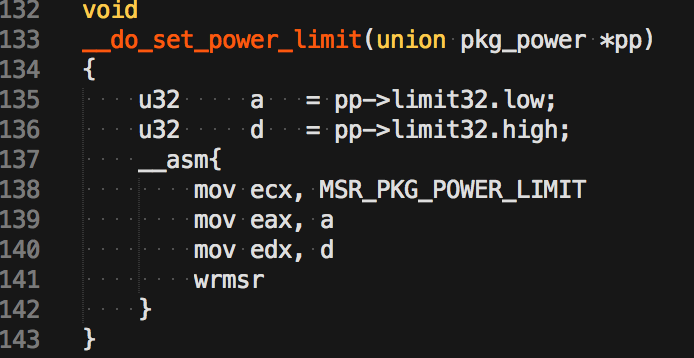
#### Time Window for Power Limit #2 (bits 55:49)

同4

#### Lock (bit 63)

锁定位，此位置为1后，则再下次cpu重新设置之前均无法更改此处的功耗限制，修改不成功但并不会死机

实现代码如下图所示：

# 附录

## 满载压力模拟&关闭命令

for((i=8;i--;))do yes > /dev/null & done

killall yes

## bilibili测试链接[【7月】没有黄段子存在的无聊世界 01【独家正版】](http://www.bilibili.com/video/av2524522/)

## [英特尔® 64 和 IA-32 架构软件开发人员手册](http://www.intel.cn/content/www/cn/zh/processors/architectures-software-developer-manuals.html)

## [Intel® Power Gadget](https://software.intel.com/en-us/articles/intel-power-gadget-20)

## rootless 关闭方式

### 开机时按cmd＋r 进入安全模式

### 找到实用工具->terminal

### csrutil disable #关闭（PS：enable #打开）

### restart