

# RPM 硬件固化与调试概述

80-P9301-16SC 版本 F

**机密和专有信息 – Qualcomm Technologies, Inc.**

**禁止公开披露：**如若发现本文档在公共服务器或网站上发布，请报告至：[DocCtrlAgent@qualcomm.com](mailto:DocCtrlAgent@qualcomm.com)。

**限制分发：**未经 Qualcomm 配置管理部门的明确批准，不得向 Qualcomm Technologies, Inc. 或其关联公司的员工之外的任何人分发。



**机密和专有信息 – Qualcomm Technologies, Inc.**

**禁止公开披露：**如若发现本文档在公共服务器或网站上发布，请报告至：[DocCtrlAgent@qualcomm.com](mailto:DocCtrlAgent@qualcomm.com)。

**限制分发：**未经 Qualcomm 配置管理部门的明确批准，不得向 Qualcomm Technologies, Inc. 或其关联公司的员工之外的任何人分发。

未经 Qualcomm Technologies, Inc. 的明确书面许可，不得使用、复印、复制或修改其全部或部分内容，或以任何方式向其他人泄露其内容。

本文中提到的所有 Qualcomm 产品是 Qualcomm Technologies, Inc. 和/或其子公司的产品。

Qualcomm 和 FSM 是 Qualcomm Incorporated 在美国和其他国家/地区所注册的商标。其他产品和品牌名称可能是其各自所有者的商标或注册商标。

本技术资料可能受美国和国际出口、再出口或转让（统称“出口”）法律的约束。严禁违反美国和国际法律。

Qualcomm Technologies, Inc.  
5775 Morehouse Drive  
San Diego, CA 92121  
U.S.A.

© 2017–2018 Qualcomm Technologies, Inc. 和/或其子公司。保留所有权利。

# 修订历史记录

| 版本 | 日期              | 说明  |
|----|-----------------|---|
| A  | 2017 年 5 月 19 日 | 初始版本  |
| B  | 2017 年 5 月 19 日 | 增加了幻灯片 27 到 55 以说明 RPMh 硬件模块的调试详情   |
| C  | 2017 年 8 月      | <ul style="list-style-type: none"><li>在幻灯片 43、47、48 和 49 中增加了不同硬件模块的 CMM 脚本</li><li>更新了幻灯片 46，其中增加了有关 CMM 脚本的详细信息</li></ul> |
| D  | 2017 年 10 月     | <ul style="list-style-type: none"><li>增加了幻灯片 30 和 31，介绍解析 RAM 转储中调试信息的 Hansei 解析器工具</li><li>更新了本演示文稿的标题</li></ul>           |
| E  | 2018 年 6 月      | <ul style="list-style-type: none"><li>增加了幻灯片 11 和 12 以介绍直接资源表决器和 RPM 和 RPMh 中的任务</li></ul>                                  |
| F  | 2018 年 8 月      | <ul style="list-style-type: none"><li>增加了有关了解 Hansei 输出的幻灯片 33、34、35 和 36</li></ul>   |

# 议程

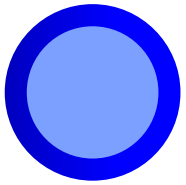
---

|                           |                           |
|---------------------------|---------------------------|
| 目标                        | <a href="#"><u>5</u></a>  |
| RPM 与 RPMh 架构对比           | <a href="#"><u>6</u></a>  |
| 调压器管理                     | <a href="#"><u>13</u></a> |
| 频率管理                      | <a href="#"><u>19</u></a> |
| 休眠管理                      | <a href="#"><u>25</u></a> |
| 调试 AOP RAM 转储             | <a href="#"><u>30</u></a> |
| 调试 ARC                    | <a href="#"><u>38</u></a> |
| 调试 BCM                    | <a href="#"><u>44</u></a> |
| 调试 PDC                    | <a href="#"><u>51</u></a> |
| 调试 VRM                    | <a href="#"><u>57</u></a> |
| 调试客户端驱动程序 – 子系统 RPMh 驱动程序 | <a href="#"><u>60</u></a> |
| SoC 休眠                    | <a href="#"><u>68</u></a> |

# 目标

---

- 本演示文稿结束后，您将了解以下内容：
  - RPM 硬件固化 (RPMh) 架构
  - RPM 和 RPMh 在以下用例中的操作差异：
    - 调压器管理
    - 频率管理
    - 休眠管理
  - 调试 RPMh 硬件块



## 第 1 节

# RPM 与 RPMh 架构对比

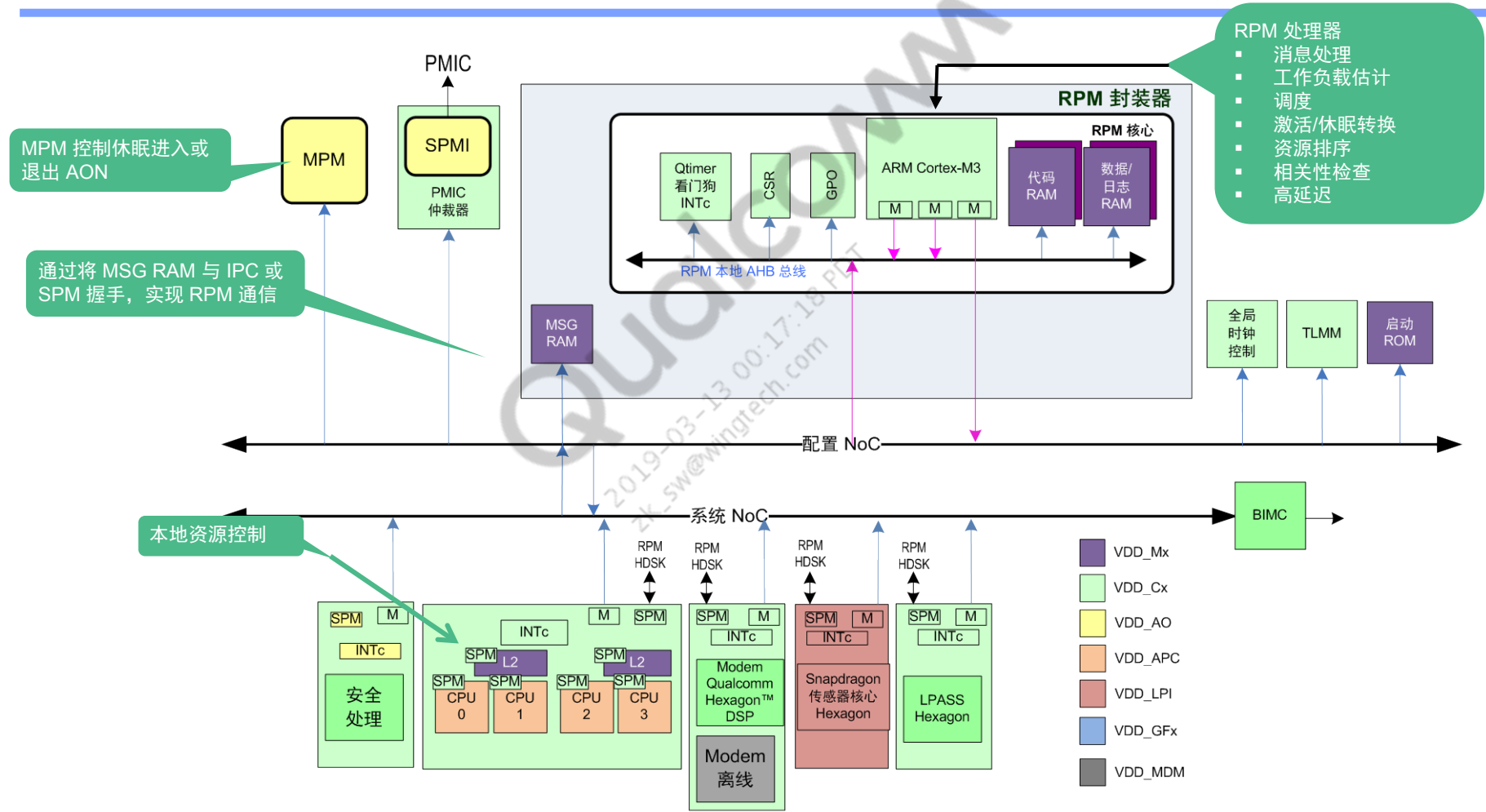
Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

# 关于 RPMh

---

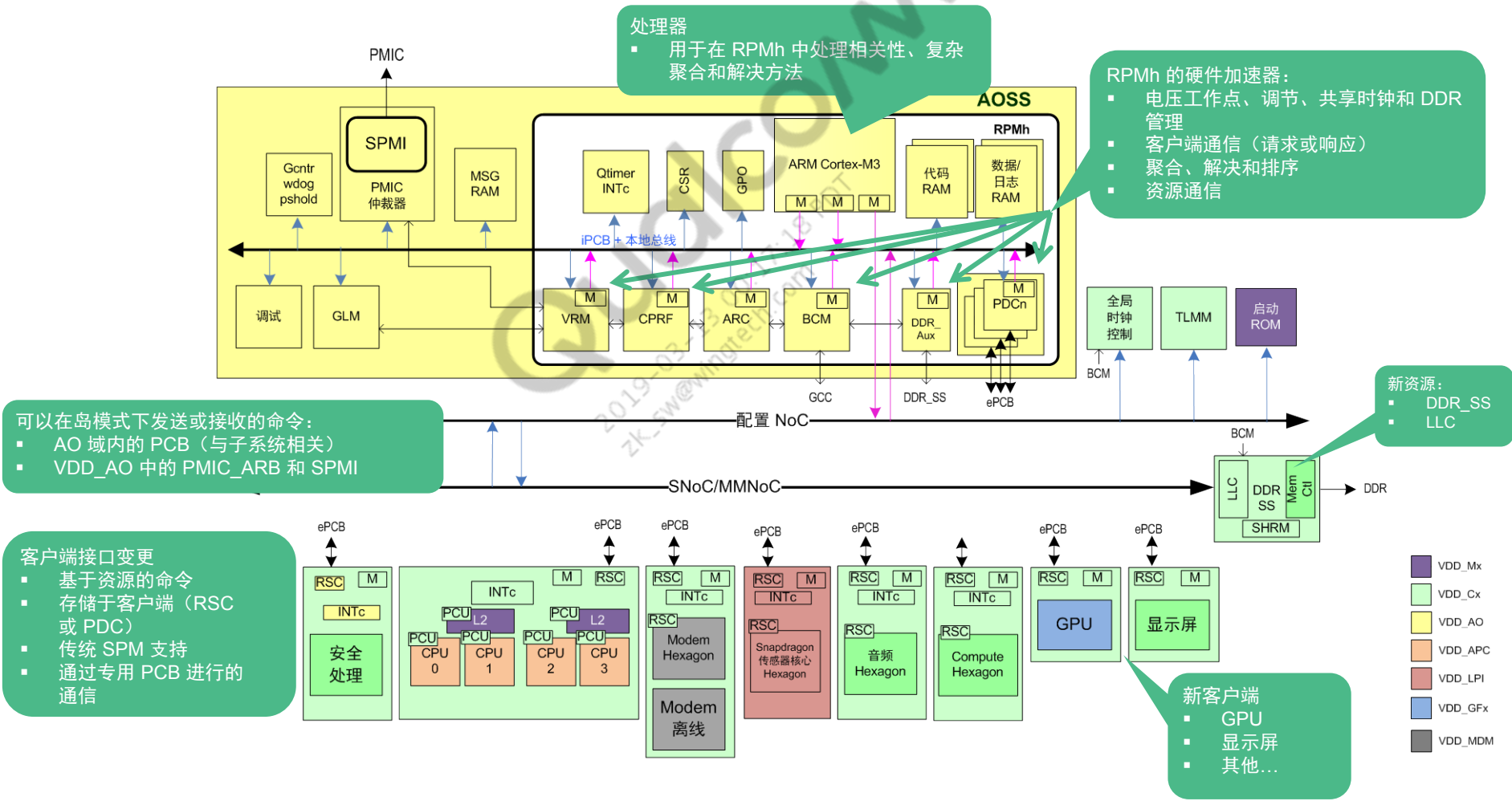
- RPMh 是一种以硬件为基础，软件为辅助的解决方案
- 凭借硬件加速转换，RPMh 显著缩短传输延迟
  - RPMh 解决方案具备以下优势：
  - 使系统加速 3 到 10 倍，缩短现有转换时间线
  - 使用时间 (DoU) 延长 2.5%
  - 使用实时响应处理器 (AOP) 帮助调试

# 硬件固化前的 RPM 架构





# RPMh 架构



# 将 RPMh 架构映射到 RPM 架构

|    | RPMh                   | 功能                      | RPM                        |
|----|------------------------|-------------------------|----------------------------|
| 本地 | 解算器（仅适用于传感器和显示屏）(SLVR) | 休眠模式决策                  | 各客户端的休眠软件                  |
|    | 资源状态协调器 (RSC)          | 子系统资源控制的休眠和唤醒状态         | 子系统电源管理器 (SPM) 和 RPM 软件    |
|    | 电源域控制器 (PDC)           | 子系统休眠和唤醒（包括中断管理）        | Modem 电源管理器 (MPM) 和 RPM 软件 |
|    | 聚合资源控制器 (ARC)          | 共享电压工作点控制               | SPM、MPM 和 RPM 软件           |
| 全局 | 总线时钟管理器 (BCM)          | 总线和存储器时钟控制              | RPM 软件                     |
|    | 调压器管理器 (VRM)           | 调压器、晶体振荡器 (XO) 缓冲区和晶体控制 | MPM 和 RPM 软件               |
|    | 电源控制总线 (PCB)           | 资源和电源控制消息传输             | 片上网络 (NoC) 和处理器间通信 (IPC)   |
|    | AOP                    | 复杂聚合或相关性处理以及解决方法        | RPM 软件                     |

# 直接资源表决器

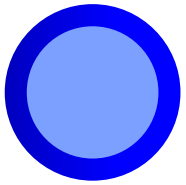
- 基于 RPMh 的主机
  - 应用程序 (APPS)
    - TrustZone (TZ)
    - Hypervisor
    - HLOS
  - Modem
    - 软件
    - 硬件
  - LPASS
  - 传感器
  - 安全处理器
  - AOP
  - GPU
  - 显示屏

| 资源                 | 基于 RPM      | 基于 RPMh   |
|--------------------|-------------|-----------|
| CX、MX              | RPM 软件控制的资源 | ARC 处理的资源 |
| XO 时钟              | RPM 软件控制的资源 | ARC 处理的资源 |
| GFX                | APPS 控制的资源  | ARC 处理的资源 |
| Modem 子系统 (MSS)    | MPSS 控制的电源轨 | ARC 处理的资源 |
| 共享时钟 (NoC、DDR)     | RPM 软件控制的资源 | BCM 处理的资源 |
| PMIC 资源 (SMPS、LDO) | RPM 软件控制的资源 | VRM 处理的资源 |

- 直接资源表决器 (DRV) 符合数字电源轨的资源相关性。  
例如,  $MX \geq CX$
- 资源必须使用相应加速器的地址进行表决。
  - CmdDB 驱动程序支持使用资源名称对加速器映射进行寻址。
  - CmdDB 位于 DDR 共享内存中。对于所需信息, 所有子系统均具备 CmdDB API。

# RPM 和 RPMh 中的任务

| 任务                | 基于 RPM 软件                                  | 基于 RPMh   |
|-------------------|--|---|
| 立即请求资源            | 写入 MSG RAM 并发送 IPC（所有子系统<br>中的 RPM 软件驱动程序） | 写入 RSC TSC 寄存器并向 AOP 发送数据包（所有子<br>系统中的 RPMh 驱动程序） |
| 关断或唤醒请求           | 子系统 SPM 到 RPM 软件                           | 子系统 RSC > 子系统 PDC                                 |
| 唤醒中断功能支持          | MPM 中断控制器                                  | PDC 中断控制器   |
| 激活或休眠设置信息         | 存储在 RPM 端                                  | 存储在子系统端（所有子系统<br>中的 RPMh 驱动程序）                    |
| 电源轨驱动程序           | RPM 电源轨驱动程序                                | 子系统 PDC 接收数据包并将其路由到 ARC<br>基于硬件的 ARC 聚合和处理        |
| CPR 驱动程序          | RPM CPR 驱动程序                               | 基于硬件的 CPR 中断处理                                    |
| 时钟或总线驱动程序         | RPM 时钟或总线驱动程序                              | 基于硬件的 BCM 聚合和处理<br>(RSC > PDC > BCM)              |
| PMIC 驱动程序         | RPM PMIC 驱动程序                              | 基于硬件的 VRM 聚合和处理<br>(RSC > PDC > VRM)              |
| DDR 低功耗模式         | RPM DDR 驱动程序                               | 向 SHRM 发送的基于硬件的 DDR_AUX 通知                        |
| 资源的最新状态           | RPM 数据结构                                   | 硬件加速器寄存器配置<br>(ARC、CPRF、BCM 和 VRM)                |
| Vdd 最小化           | RPM 休眠驱动程序                                 | 基于硬件的 ARC 聚合和处理                                   |
| 实时响应子系统 (AOSS) 休眠 | —  | AOP 软件处理  |

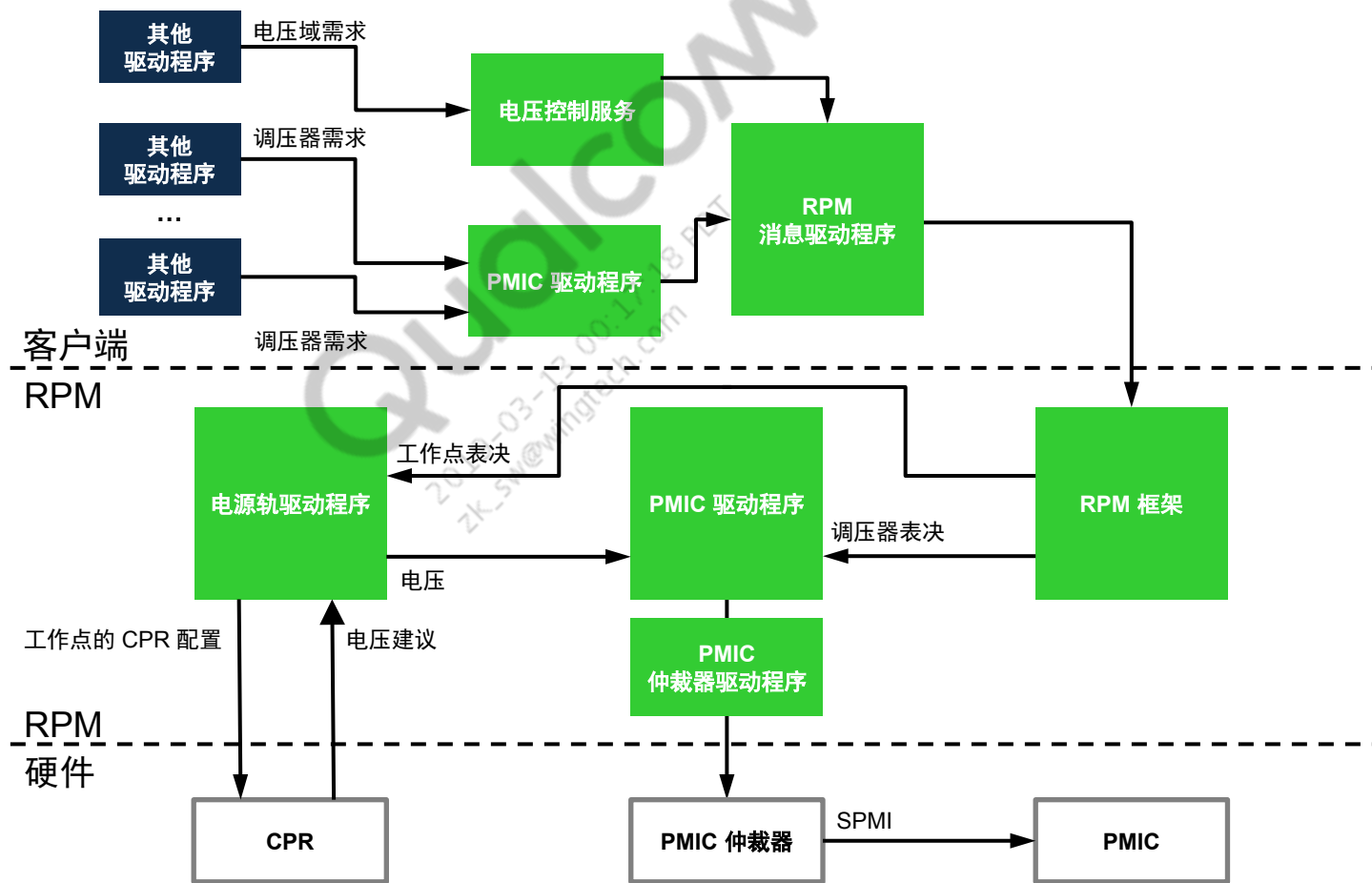


## 第 2 节

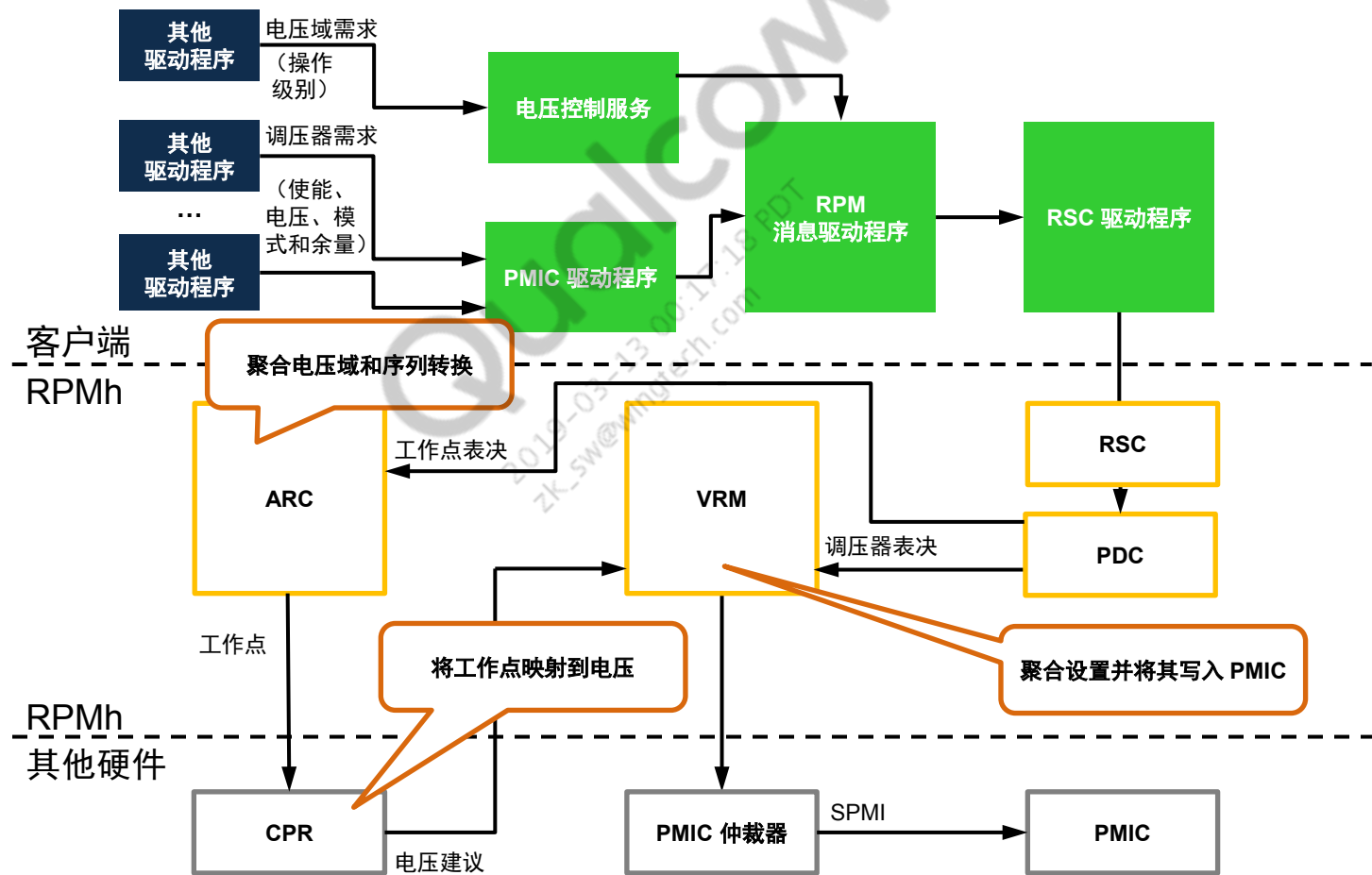
# 调压器管理

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

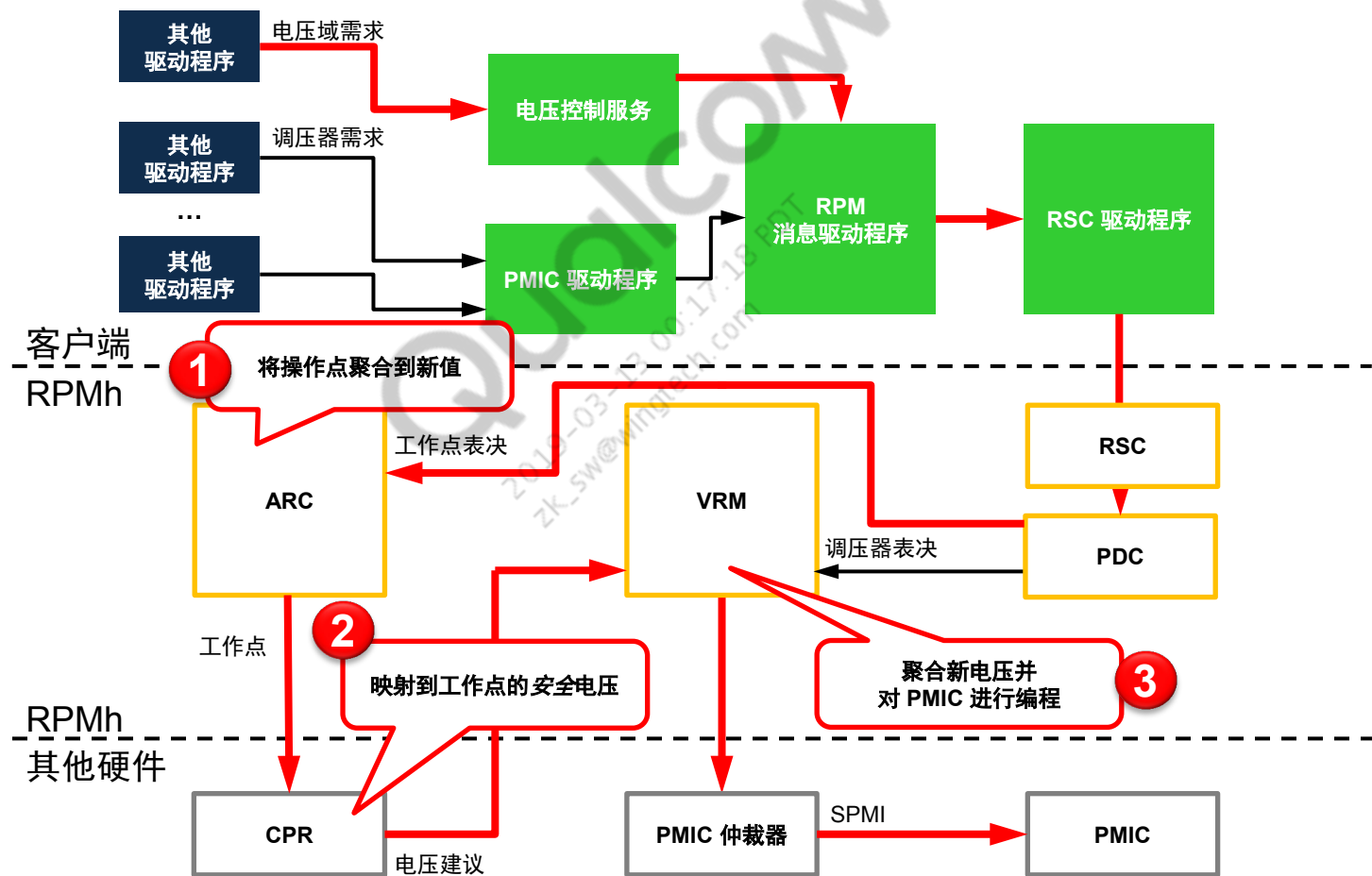
# RPM 调压器控制



# RPMh 调压器控制

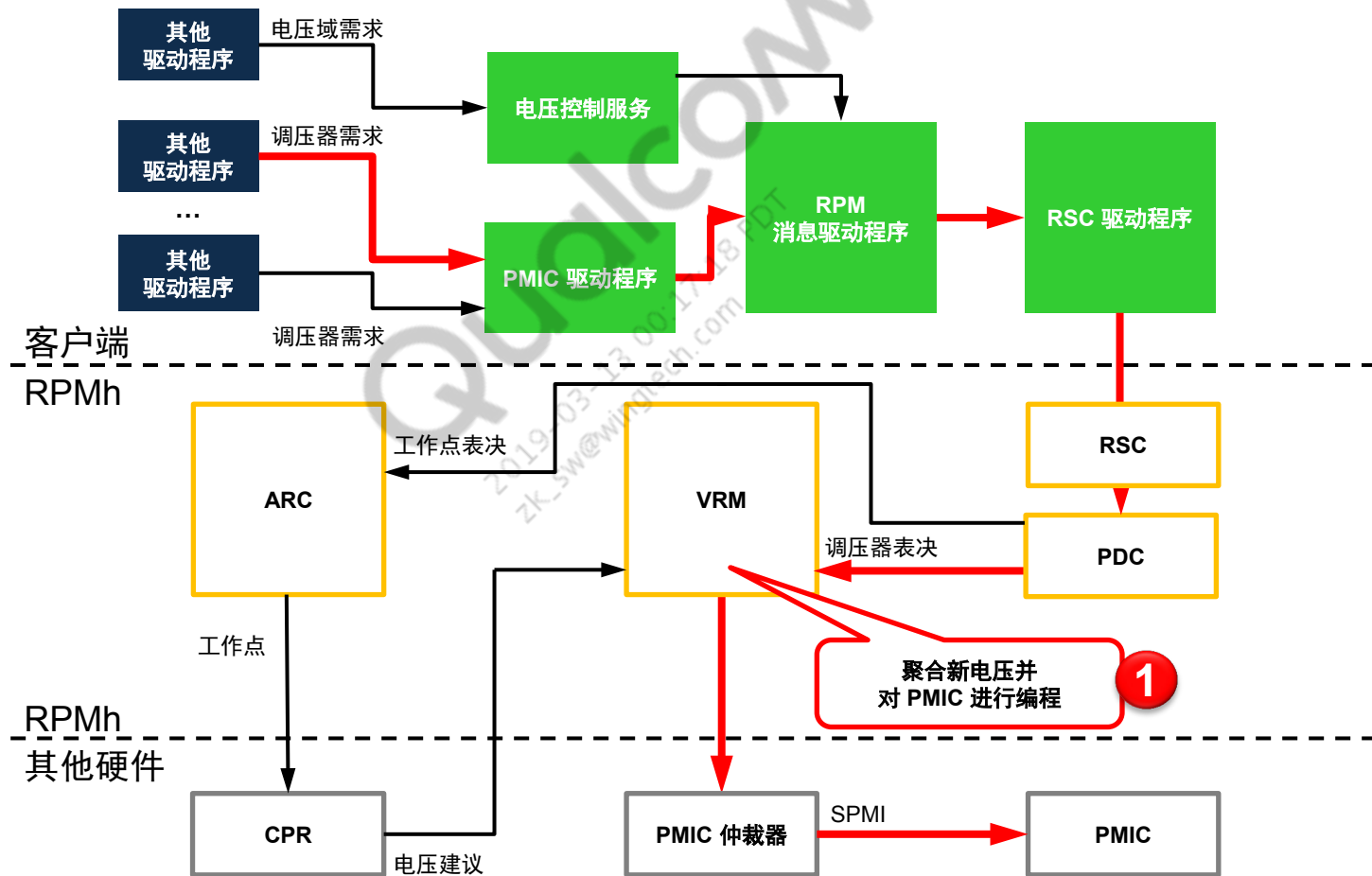


# RPMh 调压器控制 – 表决数字电压域

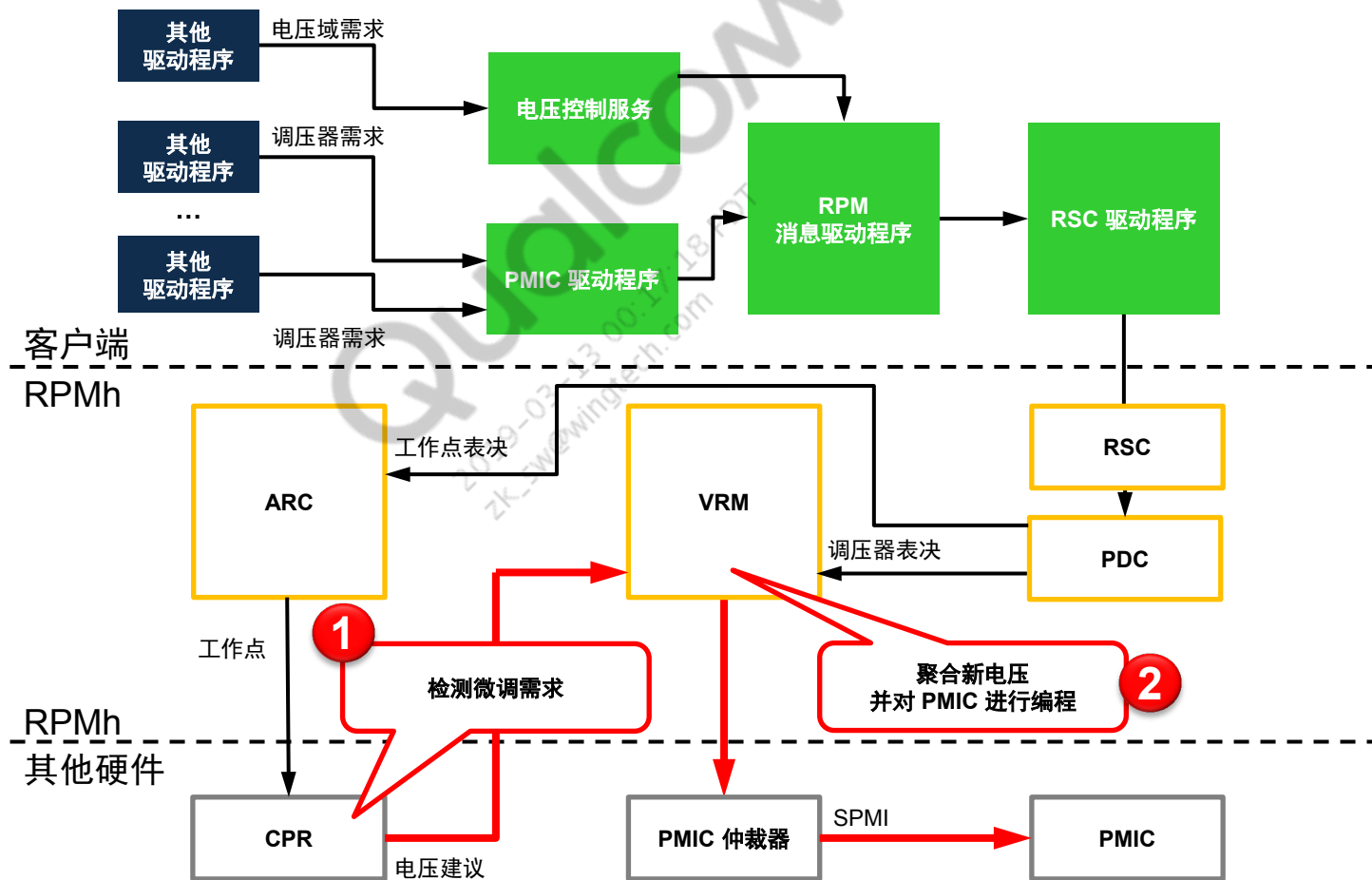


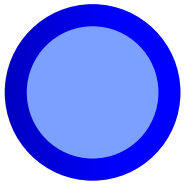


# RPMh 调压器控制 – 表决调压器



# RPMh 调压器控制 – CPR 微调



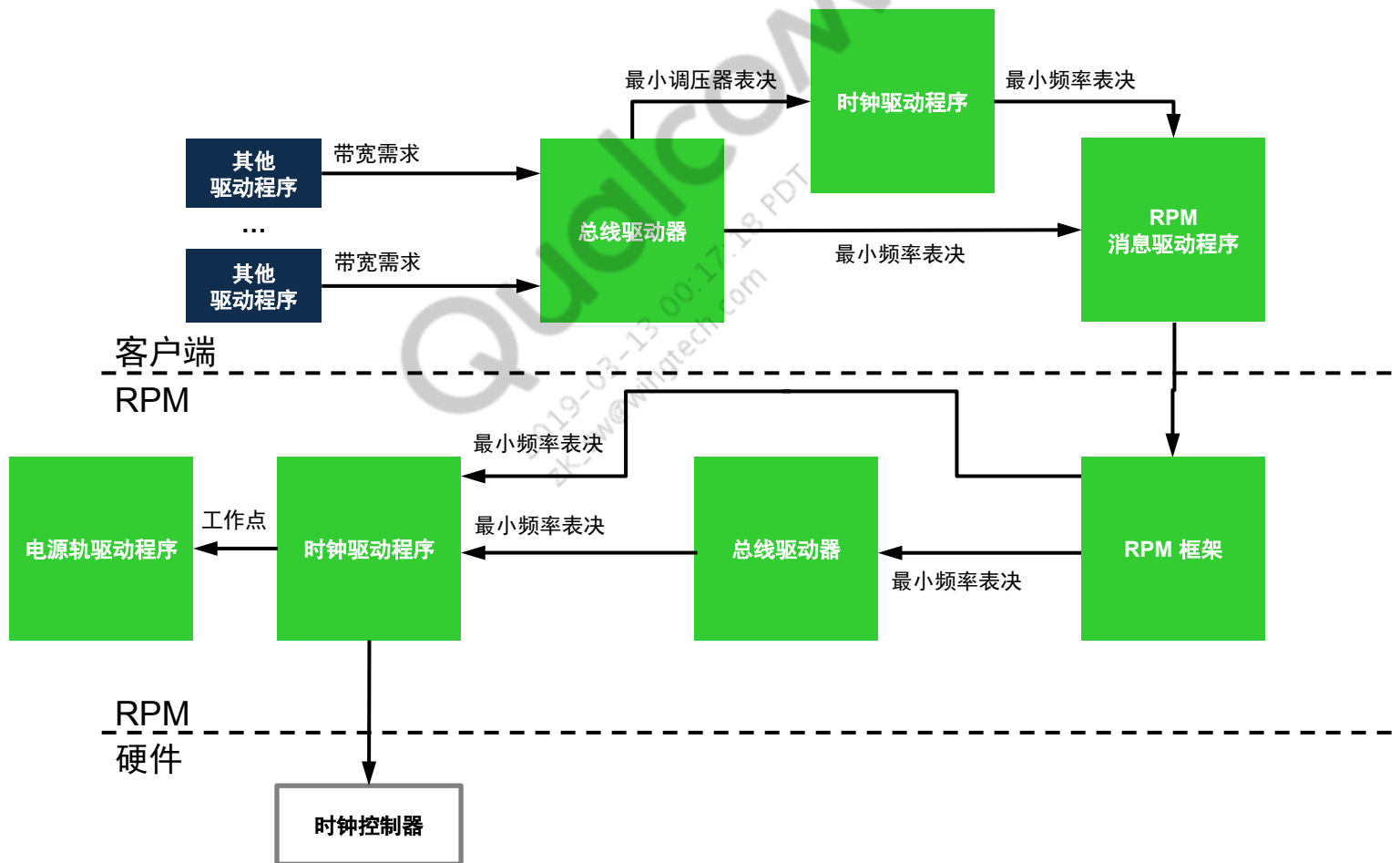


### 第 3 节

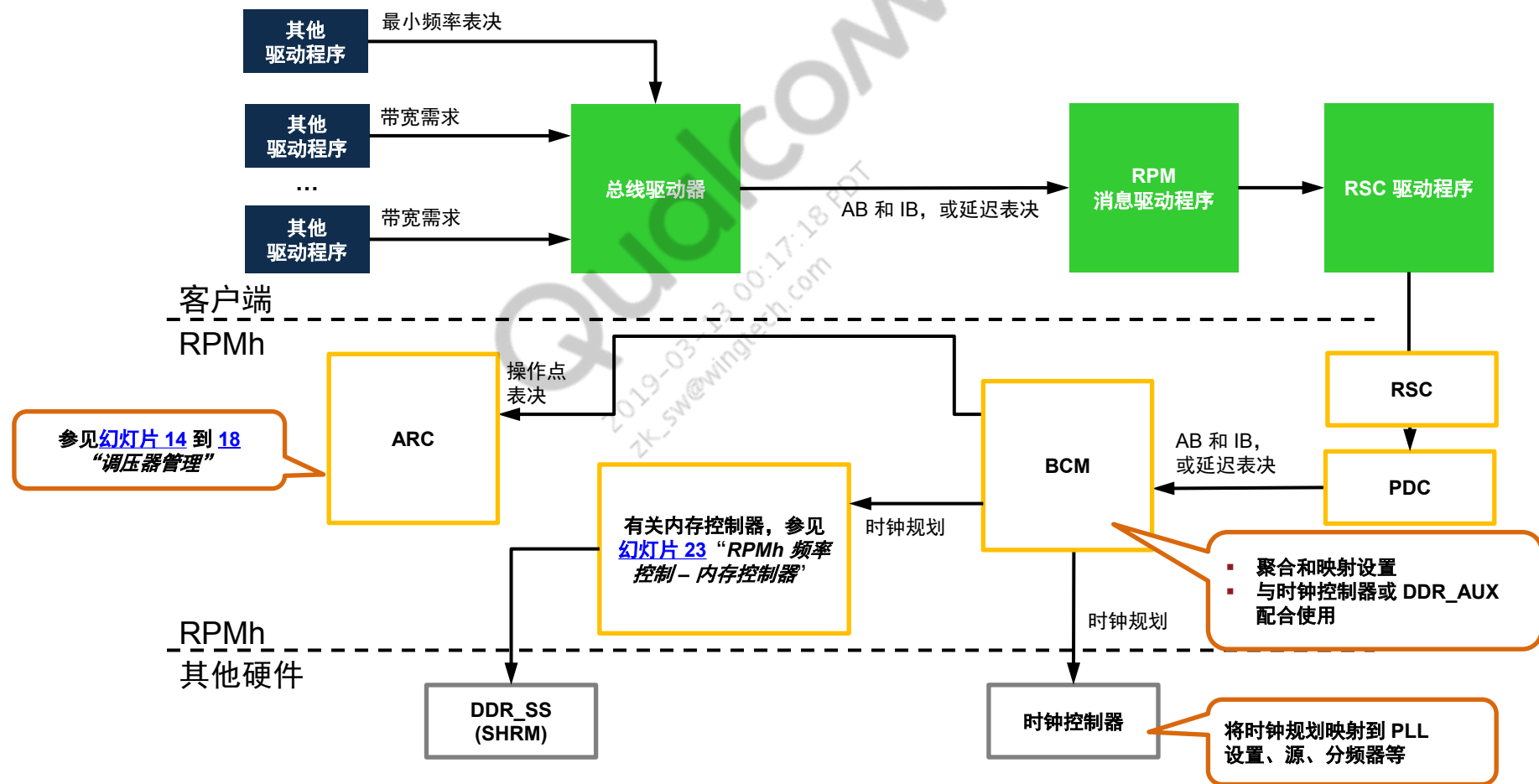
## 频率管理

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

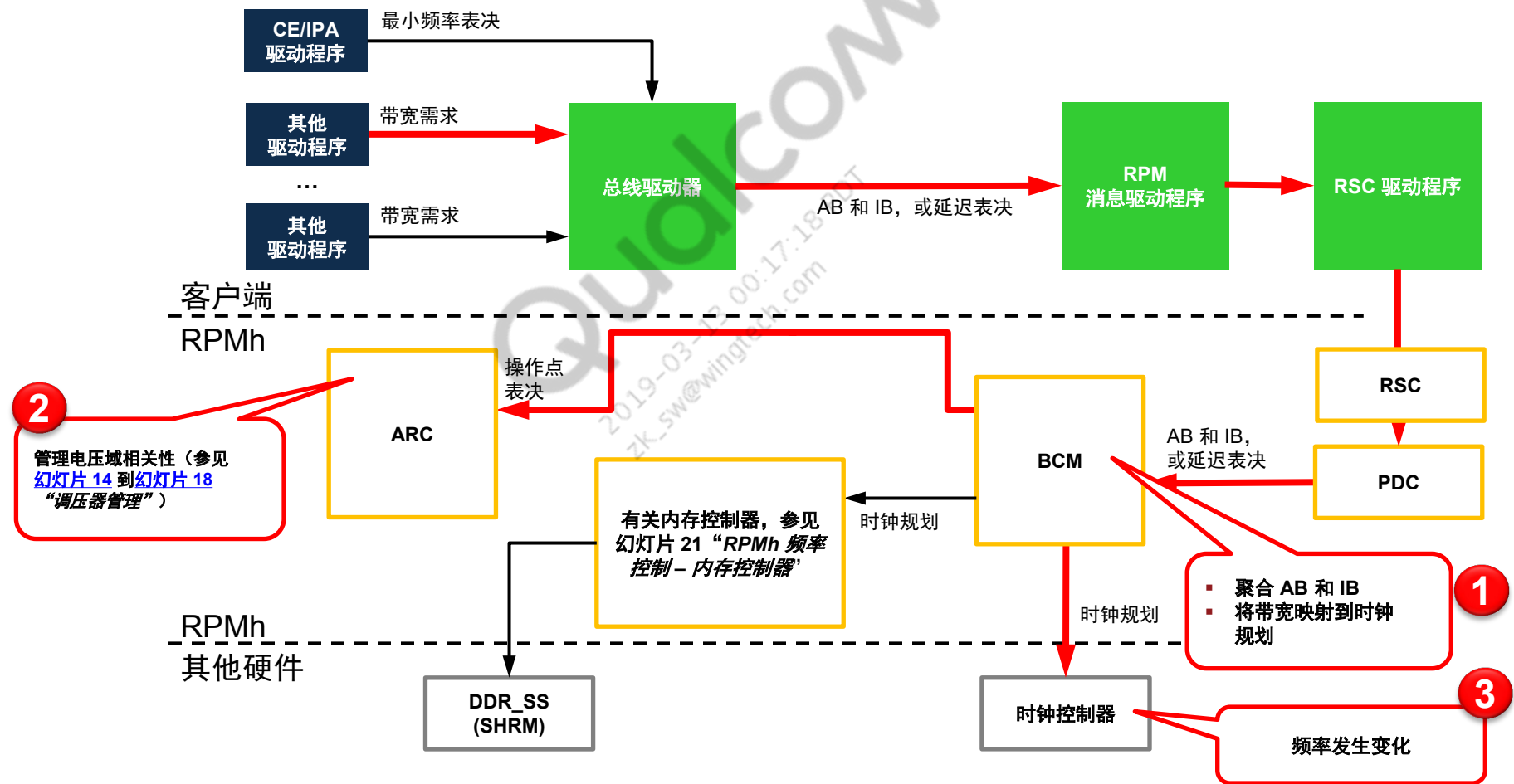
# RPM 频率控制



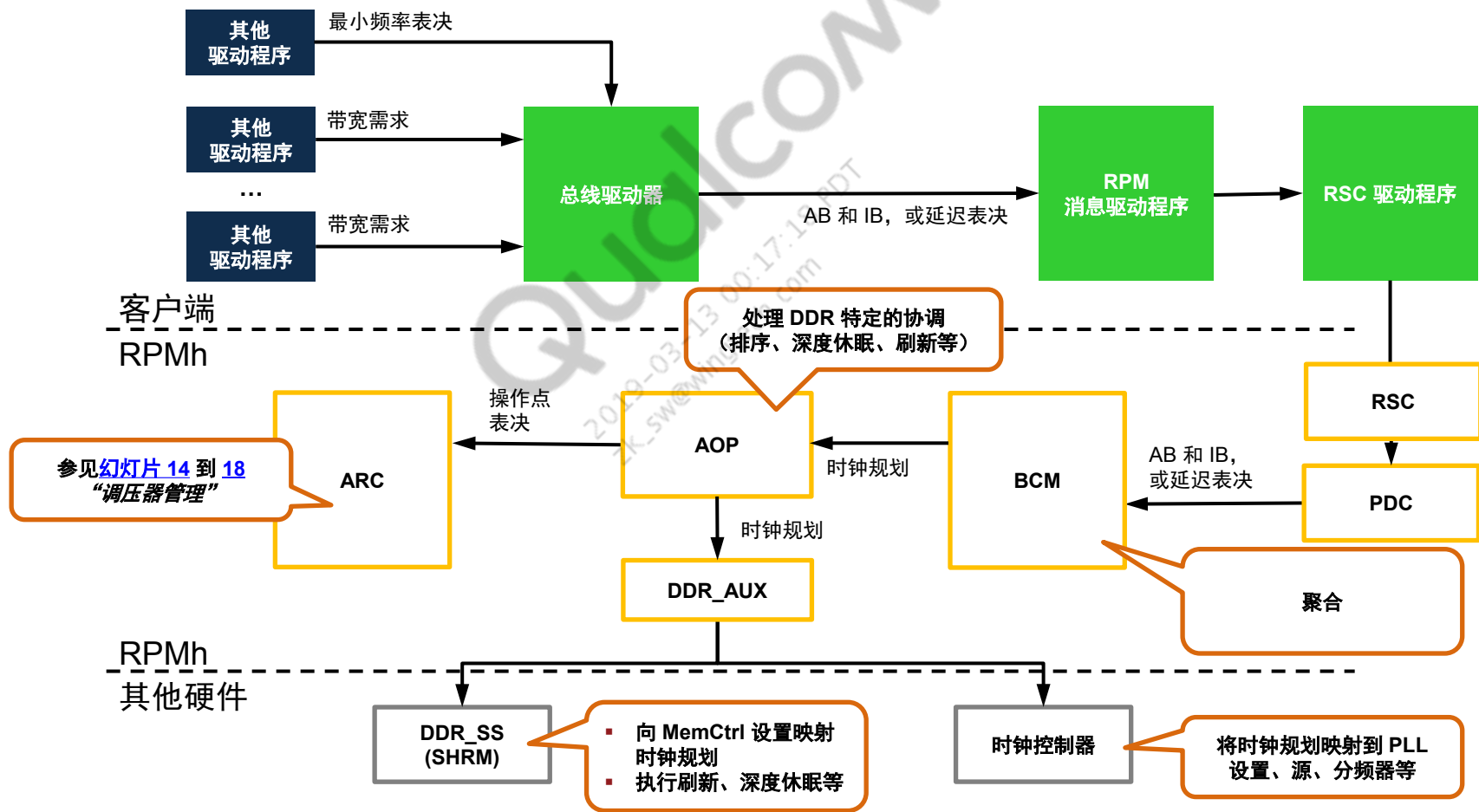
# RPMh 频率控制



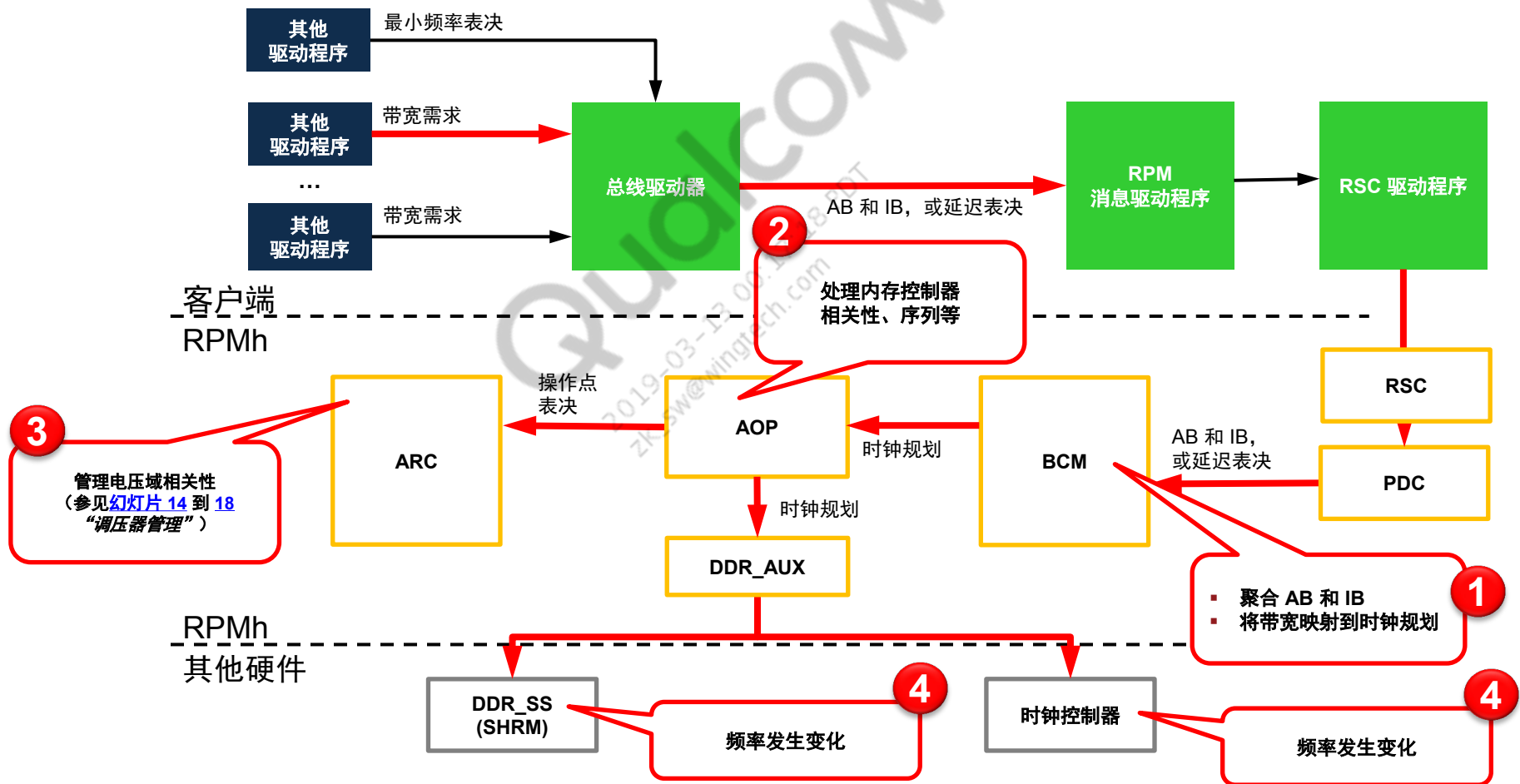
# RPMh 频率控制 – 带宽表决



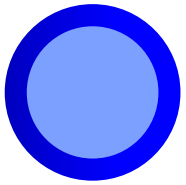
# RPMh 频率控制 – 内存控制器



# RPMh 频率控制 – 内存控制器 – 带宽表决





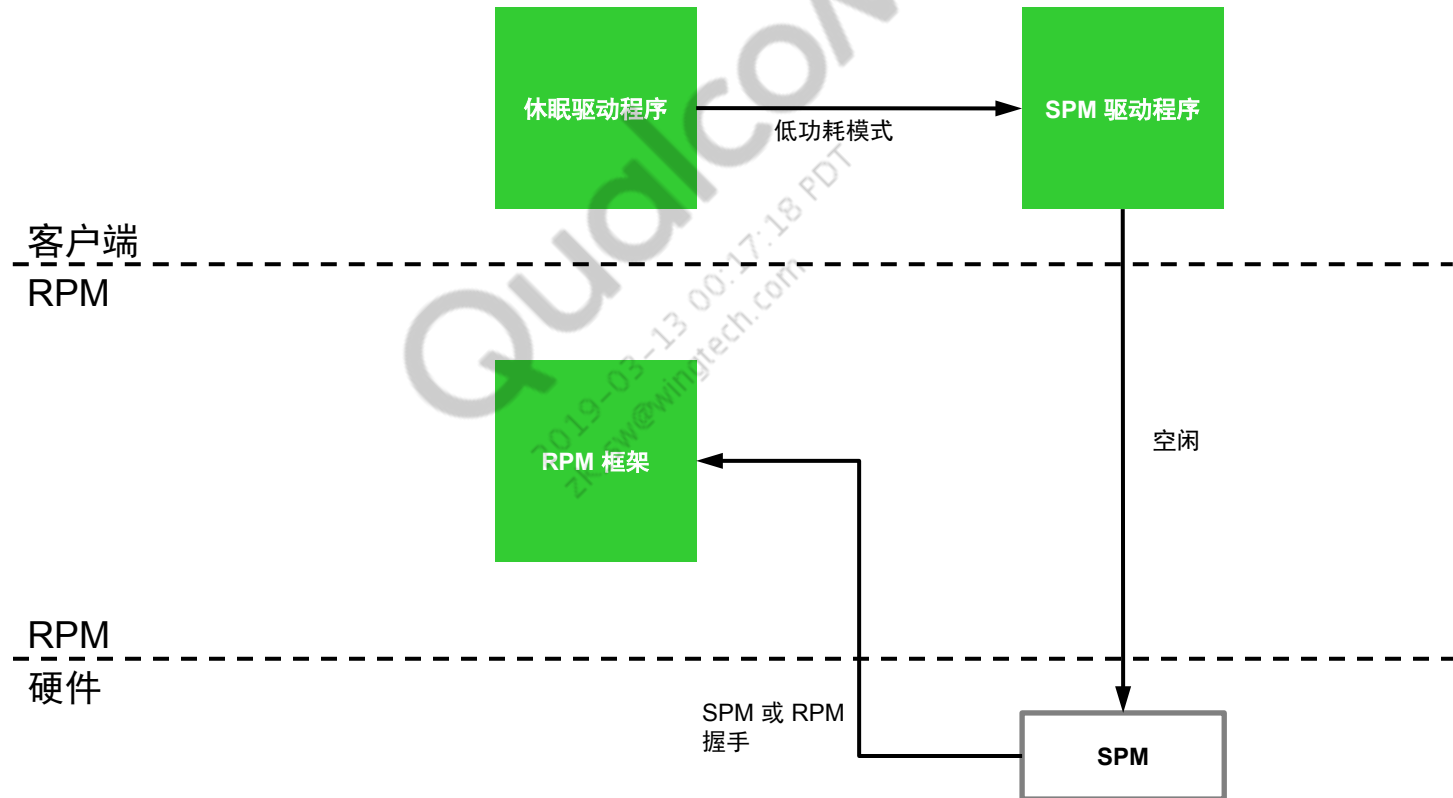


## 第 4 节

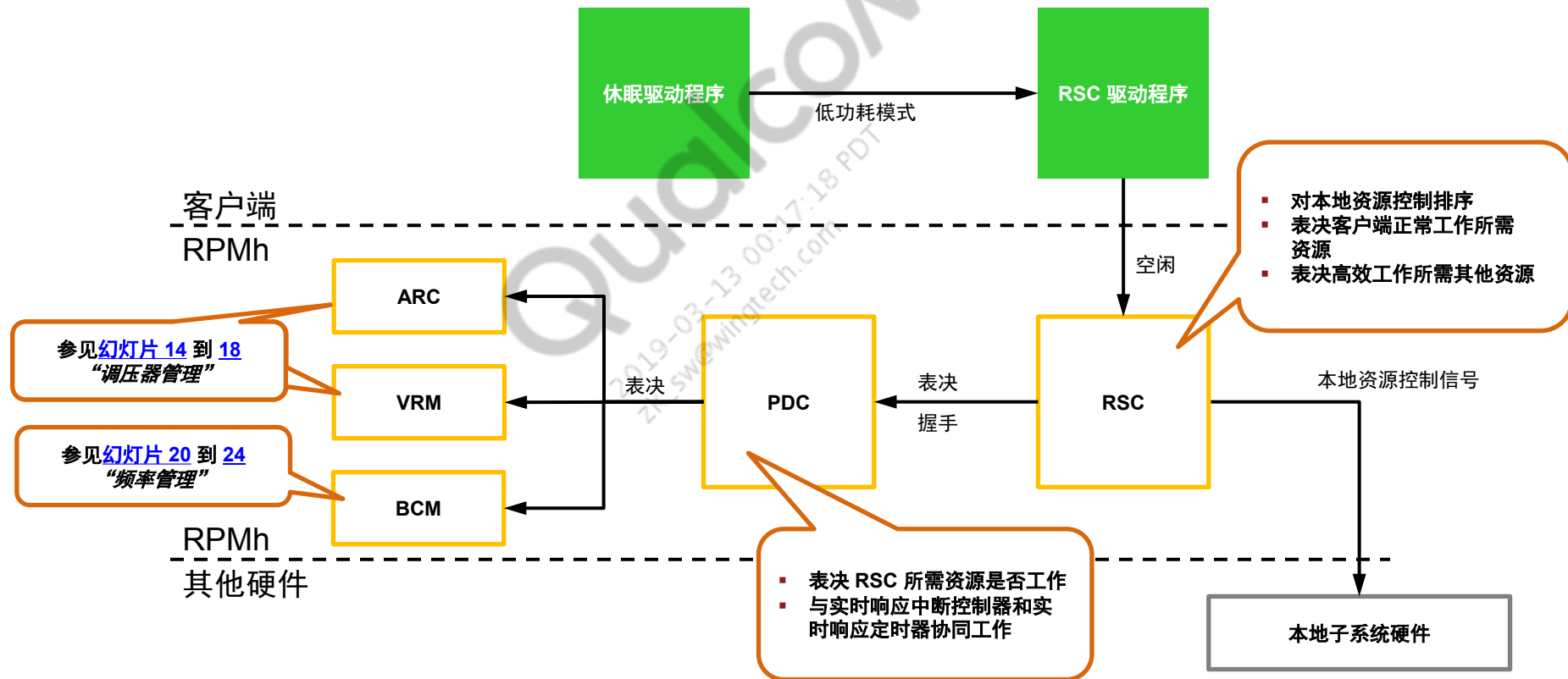
# 休眠管理

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

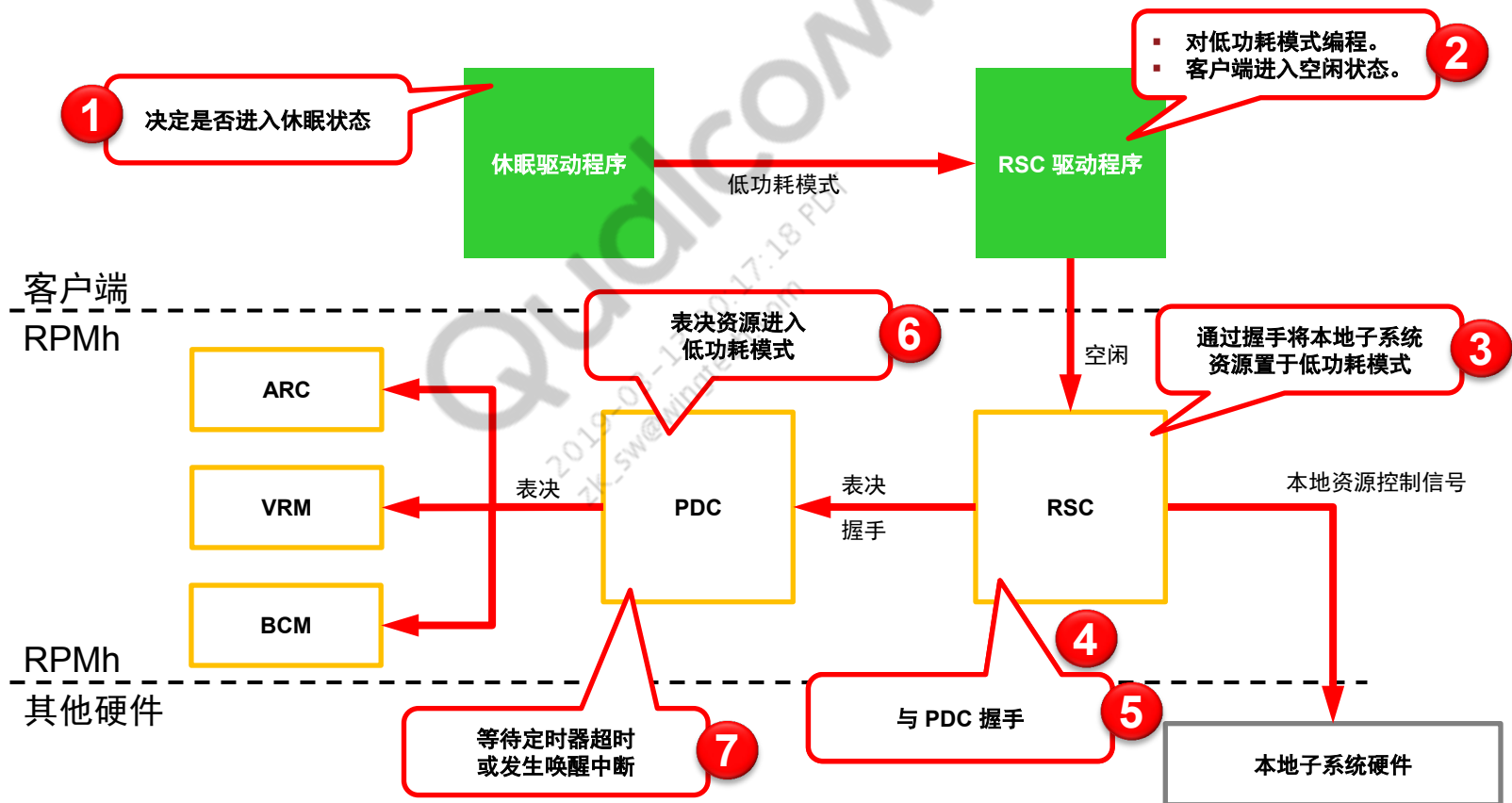
# RPM 休眠管理



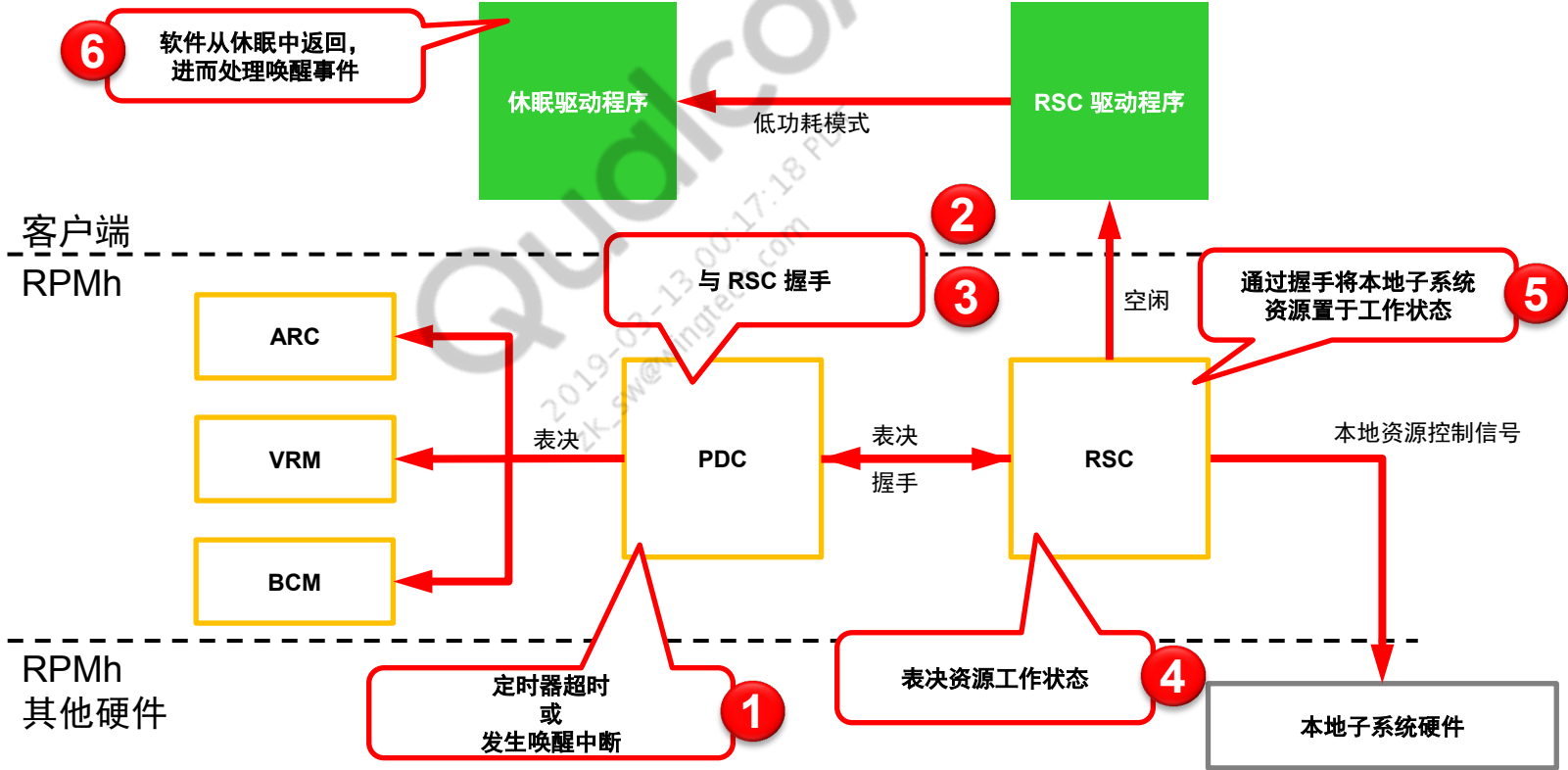
# RPM 休眠管理

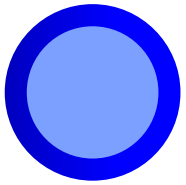


# RPMh 休眠管理 – 进入休眠



# RPMh 休眠管理 – 退出休眠





## 第 5 节

# 调试 AOP RAM 转储

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

# AOP 调试脚本

- 保存转储

- 在 AOP TRACE32 窗口的实时调试设置中，停止进程并通过运行以下命令保存转储。

```
do <aop build>\aop_proc\core\bsp\aop\scripts\aop_dump.cmm
```

- AOP 转储也是系统崩溃后收集的 RAM 转储的一部分。

- 加载转储

- 运行以下命令，在 TRACE32 仿真器中加载 AOP 转储：

```
do <aop build>\aop_proc\core\bsp\aop\scripts\aop_load_dump.cmm <dump path>
```

- 解析 AOP 用户日志

1. 通过 AOP TRACE32 以实时方式或使用仿真器运行以下命令：

```
do <aop build>\aop_proc\core\power\ulog\scripts\aop_ulogdump.cmm <output path>
```

2. 从 MS-DOS 控制台运行以下命令：

```
<aop build>\aop_proc\core\bsp\aop\scripts\aop_log_hfam.py -f "<path>\AOP External Log.ulog"  
-tbl <aop build>\aop_proc\core\api\debugtrace\tracer_event_tbl.h > AOP_ulog_parsed.txt
```

# Hansei RAM 转储解析器

- Hansei 工具用于解析 RAM 转储输出的调试信息
- 安装
  1. 访问 <https://www.python.org/> 下载 Python v2.7.x。  
如需检查 Python 的当前版本，输入以下命令：  

```
python -V
```
  2. 安装支持 ARM 编译器的 pyelftools 库，使用 <https://bitbucket.org/pplesnar/pyelftools-pp> 中提供的版本  
输入以下命令开始安装：  

```
python setup.py install
```
- Hansei 脚本已随 AOP 共同发布
  - 位置 – aop\_proc\core\bsp\aos\scripts\hansei\
  - Hansei 引用编译版本中的多份文件，需要从 AOP 编译版本执行 Hansei
- 如需使用脚本进行 RAM 转储解析，输入以下命令：  

```
aop_proc\core\bsp\aos\scripts\hansei\hansei.py --elf <aop>.elf -o . -t 845  
dumpfile <ramdump folder>
```



# 了解 Hansei 输出

- **Aop-summary.txt 文件**

提供 AOP 固件的平台信息和状态，例如，是正在运行还是处于致命场景。如果由于故障（例如，总线、使用、硬件/内存管理故障）而处于致命场景，则将汇总故障详细信息。

- **.BIN 文件夹**

包含所需二进制文件（CODERAM.BIN、DATARAM.BIN 和 MSGRAM\*.BIN）和 AOP elf 的备份，以供将来参考。还包含 CMM 脚本，可用于加载二进制文件，以供进一步调试。

- **Requests\_By\_Master 文件夹**

每个子系统有一个单独的文件夹，其中包含每个资源的表决。例如，AOP\_drv6.txt、APPS\_drv2.txt 等等。

**注意：**如果 SDI 无法备份 RPMh 二进制文件，则不会生成此文件夹。

- **Requests\_For\_Resource 文件夹**

- arc\_vt.txt: 包含电源轨资源（例如：CX 和 MX）表决的列表以及转储收集时每个资源的状态。
- bcm\_vt.txt: 包含时钟资源（例如：DDR、SNOC 和 CNOC）表决的列表以及转储收集时每个时钟的状态。
- vrm\_vt.txt: 包含 PMIC 资源（例如：s1a、s2a、l2a 等等）表决的列表以及收集转储时每个 SPMS/LDO 的状态。
- cprf.txt: 包含每个电源轨的 CPRF 控制设置和电压表。
- rpmh\_summary.txt: 包含每个 RPMh 资源管理器的忙碌/空闲状态。

**注意：**如果 SDI 无法备份 RPMh 二进制文件，则不会生成此文件夹。

# 了解 Hansei 输出（续）

- **.Ulog 日志二进制文件**

从转储中提取的“AOP DDR Log.ulong”和“AOP External Log.ulong”记录二进制文件。

- aop-rawts.txt（定时器节拍（19.2 MHz 时钟）的时间戳）和 aop-log.txt（以秒为单位的时间戳）通过“AOP External Log.ulong”来解析，并且包含 RPMh 驱动程序、进入休眠/退出休眠和异常中的总体 AOP 固件记录。
- ddr-rawts.txt（定时器节拍（19.2 Mhz 时钟）的时间戳）和 ddr-log.txt（以秒为单位的时间戳）通过“AOP DDR Log.ulong”来解析，并且包含 ddr\_mgr 驱动程序中关于 Qualcomm® FSM™ 平台的电压和频率事件的记录。

- **Npa-dump.txt 文件**

包含向 NPA 注册的资源状态（如果 AOP 固件中包含多个客户端）。还包含来自每个客户端的请求。

- **Task\_info.txt 文件**

包含一个列表，其中列出 AOP 固件中运行的任务及其优先级、等待事件、等待信号和状态（是否已暂停）。

- **Cmd\_db.txt 文件**

包含 RPMh 管理的资源及其表决地址的列表。

- **Sleep\_stats.txt 文件**

包含 AOP/AOSS 低功耗统计信息（sleep\_count（进入低功耗模式的次数）、last\_entered\_at、last\_exited\_at、accumulated\_duration（系统持续处于 LPM 模式的总累计时间））。

- aosd: AOSS 深度休眠
- cxsd: CX 深度睡眠

# 了解 Hansei 输出（续）

- **Aop\_serv\_msgram\_parse.txt 文件**

包含与 AOP 服务驱动程序进行的中断处理相对应的日志记录。作为 ISR 处理的一部分，将记录以下内容：

- \*ISR 编号和收到的时间戳
- \*向 AOP 服务发出的相应信号
- \*资源/信号转换：当前状态和所需状态
- \*中断处理过程中的其他参数和阶段

- **ddr\_mgr\_msgram\_parse.txt RAM 日志**

包含与 ddr\_mgr 驱动程序进行的中断处理相对应的日志记录

（主要用于 MC/Shub 切换。）作为 ISR 处理的一部分，将记录以下内容：

- \*ISR 编号和收到的时间戳
- \*ISR 处理的多个阶段
- \*MC/Shub 频率等级转换：当前状态和所需状态
- \*依赖于 CX/MX/VDDA 转换（如果有）
- \*中断处理过程中的其他参数和阶段

# 了解 Hansei 输出（续）

- **PDC 文件夹**

每个子系统有一个单独的文件夹，其中包含相应子系统的 pdc 状态/配置。例如：PDC\_AOP.txt、PDC\_APPS.txt 等等。

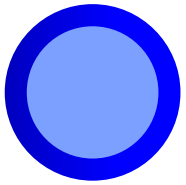
- PDC\_MODE\_STATUS：在直通模式下，子系统处于活动状态；在定序器模式下，子系统处于休眠状态。
- ENABLE\_PDC：是否启用 PDC。
- TIMER\_MATCH\_VALUE\_HI 和 TIMER\_MATCH\_VALUE\_Lo：相应子系统的下一次唤醒时间。
- IRQ\_CFG、IRQ\_ENABLE 和 IRQ\_STATUS：当子系统中端控制器无法正常工作时，PDC 代表子系统监视唤醒中断。其中一部分中断是对每个 PDC 均可见的 GP 中断，另一部分中断特定于此 PDC。
  - CFG：配置中断的触发方式（例如，电平/边沿/上升沿/下降沿触发）。
  - Enable：启用 PDC 正在监视的中断并唤醒子系统。
  - STATUS：显示中断的状态（是否发生）。

**注意：**如果 SDI 无法备份 RPMh 二进制文件，则不会生成此文件夹。

# RPMh 调试

---

- 调试基于硬件寄存器和状态
- RAM 转储捕获 RPMh 硬件寄存器和状态
- 从 AOP 版本运行 CMM 脚本，以解析 RAM 转储



## 第 6 节

# 调试 ARC

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

# ARC 资源

---

- 处理需要额外协调的复杂硬件模块
- 可根据需要协调 ARC 和其他加速器（CPRF、VRM、DDR\_Aux 等）
- 表决单元处于操作级别
- ARC 资源
  - 电压轨
    - CX 或 MX
    - SSC\_CX 或 SSC\_MX
    - VDDMSS
    - EBI
    - GFX
  - 带有电源状态的常规资源
    - XO
    - DDR\_SS

# ARC 调试

---

- 解析各资源时（例如电源轨），CMM 脚本检查以下事项：
  - 常规状态
  - 各客户端的表决
  - 内部状态
- 示例调试方案
  - ARC 尚未转换资源状态 – 通过检查资源的内部状态完成调试
  - ARC 未达到预期状态
    - 通过检查各客户端的表决完成调试
    - 通过检查资源内部状态完成深入调试



# ARC 转储脚本

- `arc_dump.cmm [op=<output file path>]`

- 转储每项 ARC 资源的所有信息
- 如果未指定输出文件，则在 TRACE32 区域中转储输出
- 针对 RM 列表和 DRV 列表，查看生成的 ARC 转储

```
RM0 : cx
      Status
      Enable: 1
      OLS
      Curr : 0x2
      Agg : 0x1
      Dest : 0x1
      Solved: 0x0
      Seq : 0x1
      Sequencer
      Busy : 1 ← Seq busy- 'hang'
      PC : 108
      Instr : 0x400F
      Votes
      DRV0 : 0x0
      DRV1 : 0x0
      DRV2 : 0x1* ← Decider DRV
      ... ← Other DRV votes
      DRV29 : 0x0

RM1 : mx
... ← Other RM details
ARC_RM Map
ARC_DRV Map
```

**注意：**OL – 操作级别；DRV – 表决器

# ARC 转储脚本 DRV 过滤器

- `arc_dump.cmm drvs=mss,aop`  
    `[op=<output file path>]`
  - 可以过滤输出中的特定 DRV
- MSS\_SW 和 AOP 转储适用以下内容：
  - 各 ARC 资源的表决
  - 当前和目标操作级别
  - 定序器状态

DRV10 : mss

RM0 : cx

vote : 0x0

curr\_ol : 0x6

dest\_ol : 0x6

seq\_busy: 0x0

...

← Intermediate RMs

RM6 : mss

**vote : 0x4\*** ← Decider vote

curr\_ol : 0x4

dest\_ol : 0x4

seq\_busy: 0x0

...

DRV6 : aop

RM0 : cx

**vote : 0x6\*** ← Decider vote

curr\_ol : 0x6

dest\_ol : 0x6

seq\_busy: 0x0

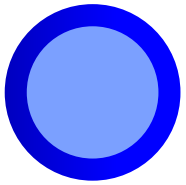
...

← Rest of RMs

# ARC 转储脚本 RM 过滤器

- `arc_dump.cmm rms=mx,mss`  
[op=<output file path>]
  - 可以过滤输出中的特定 RM
- CX 和 MSS 资源转储适用以下内容：
  - 各种操作级别
  - 定序器状态
  - DRV 表决

RM1 : mx  
Status  
Enable: 1  
OLs  
Curr : 0x4  
Agg : 0x4  
Dest : 0x4  
Solved : 0x0  
Seq : 0x4  
Sequencer  
Busy : 0  
PC : 36  
Instr : 0x5005  
Votes  
DRV0 : 0x0  
DRV1 : 0x0  
DRV2 : 0x4\* ← Decider DRV  
... ← Other DRV votes  
DRV29 : 0x0  
RM6 : mss  
... ← MSS details



## 第 7 节

# 调试 BCM

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

# BCM 资源

---

- 处理共享时钟、电源域和内存控制器提示
- 表决单元特定于域：
  - 带宽
  - 延迟
  - 其他
- BCM 资源
  - 系统总线时钟和域
    - 系统 NoC 和聚合 NoC
    - 内存 NoC
    - 多媒体 NoC
    - 配置 NoC
  - 其他共享时钟资源
    - Qualcomm 通用外设 (QUP) v3
    - IPA
    - Crypto

# BCM 调试

---

- 解析各资源时（例如 BCM 或虚拟时钟域 (VCD)），CMM 脚本检查以下事项：
  - 常规状态
  - 各客户端的表决
  - 内部状态
- 示例调试方案
  - BCM 卡断 – 检查 VCD 的 BCM 状态
  - BCM 未达到预期聚合状态
    - 检查 VCD 的聚合状态
    - 检查由 VCD 驱动的 BCM 表决表

# BCM 转储脚本

- BCMDump.cmm [<output dir>]
  - 转储各 VCD 和 BCM 实例的所有信息
  - 如果未指定输出文件，将输出转储到 TRACE32 区域
  - 以下是 VCD 列表：

|         |          |
|---------|----------|
| MC      | 0        |
| SHUB    | 1        |
| SHRM    | 2        |
| SNoC    | 3        |
| MMNoC   | 4        |
| CNoC    | 5        |
| Crypto  | 6        |
| IPA     | 7        |
| QUP     | 8        |
| DDR_SS  | 10       |
| 活动客户端向量 | 11       |
| 未使用     | 9, 12-15 |

BCM Front End:

VCD0: AGG\_BW: 0x1533 FINAL\_CP: 0x8 AGG\_CP 0x8  
...

BCM SNDs:

SND0: SEL\_CP: 0x0  
...

BCM Back End:

VCD0: CLK\_DEST\_STATE: 0x8 COMBINED\_CP: 0x8 SW\_CP\_SNAP: 0x0  
WRITTEN\_CP: 0x8 CURR\_CP: 0x8  
...

BCM Back End Sequencers:

VCD0: IDLE: 0x1 CURR\_PC: 0x3E  
...

Vote Table for DRV ID: 0

Vote Table for DRV ID: 1

Vote Table for DRV ID: 2

BCM0: VALID: 0x1 VOTE\_X: 0x0 VOTE\_Y: 0x1533

BCM3: VALID: 0x1 VOTE\_X: 0x0 VOTE\_Y: 0x8

...

注意：

- CP – 时钟规划
- VCD – 虚拟时钟域
- SND – SNoC 域

# BCM 前端

- 包含有关 VCD 状态的信息：

- 聚合带宽值 (AGG\_BW)
- 最终选定的时钟规划 (FINAL\_CP)、SND CP 聚合以及聚合的 CP (AGG\_CP)；AGG\_CP 选择基于 AGG\_BW

VCD0: AGG\_BW: 0x1533 FINAL\_CP:  
0x8 AGG\_CP 0x8

VCD1: AGG\_BW: 0x1900 FINAL\_CP:  
0x5 AGG\_CP 0x5

- BCM 特殊节点 CP 选择

- SEL\_CP: 根据延迟表决和当前表决负载选择的 CP

...

BCM SNDs:

SND0: SEL\_CP: 0x0

SND1: SEL\_CP: 0x0

...



# BCM 后端

- 根据 VCD 的状态决定的后端：
  - CLK\_DEST\_STATE – 下一个选定的 CP
  - COMBINED\_CP – 结合软件和硬件选择的 CP
  - SW\_CP\_SNAP – 软件覆盖 CP 选择（不常见）
  - WRITTEN\_CP – 由 BCM 写入的当前 CP
  - CURR\_CP – 当前确认的 CP
- 后端定序器状态
  - 空闲
  - $0 \times 1$  = 不繁忙
  - CURR\_PC
  - 超过最后一个运行指令的单一指令

## BCM 后端：

VCD0: CLK\_DEST\_STATE: 0x8  
COMBINED\_CP: 0x8 SW\_CP\_SNAP: 0x0  
WRITTEN\_CP: 0x8 CURR\_CP: 0x8  
VCD1: CLK\_DEST\_STATE: 0x5  
COMBINED\_CP: 0x5 SW\_CP\_SNAP: 0x0  
WRITTEN\_CP: 0x5 CURR\_CP: 0x5  
...

## BCM 后端定序器：

VCD0: IDLE: 0x1 CURR\_PC: 0x3E  
VCD1: IDLE: 0x1 CURR\_PC: 0x3E  
...

# BCM 表决表

- 每个 DRV 的 BCM 表决状态:

- VALID – 表决有效性
- VOTE\_X – 与资源相关, 通常为 AB
- VOTE\_Y – 与资源相关, 通常为 IB

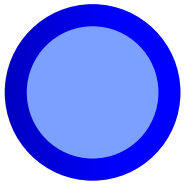
Vote Table for DRV ID: 0

Vote Table for DRV ID: 1

Vote Table for DRV ID: 2

BCM0: VALID: 0x1 VOTE\_X: 0x0  
VOTE\_Y: 0x1533

BCM3: VALID: 0x1 VOTE\_X: 0x0  
VOTE\_Y: 0x8



## 第 8 节

# 调试 PDC

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

# 关于 PDC

---

- 运行休眠序列，关闭子系统的关键资源
- 运行唤醒序列，唤醒关键资源，使子系统 RSC 进入运行状态
- 当子系统处于 RPMh 辅助电源深度休眠状态时，处理唤醒中断

# PDC 调试

---

- 解析各资源时（例如全局或子系统），CMM 脚本检查以下事项：
  - 常规状态
  - 中断请求 (IRQ) 配置
- 示例调试方案
  - PDC 卡断 – 检查 DRV 的 PDC，然后检查 DRV
  - 子系统唤醒出现问题
    - 检查 PDC IRQ 状态
    - 检查预期中断的 IRQ 配置

# 子系统的 PDC 用户日志

- 在非 HLOS 中，运行 `uLogDump.cmm` 解析 PDC 日志：

`ULogDump.cmm <output file path> PDC Log`

- 示例日志

```
Content-Type: text/pdc-driver-1.0; title=PDC Driver
0x2F92FC02: Initializing target
0x2F93089E: Initializing main driver
0x2F930B86: Interrupt configuration (Number of Interrupts: 15)
0x2F930DA4: GPIO configuration (Number of GPIOs: 96) (Number of MUXs: 20)
0x2F9F7231: TCS Resource lookup: (Name: cx.lvl) (Base address: 0x30000)
0x2F9F77ED: TCS Resource lookup: (Name: mx.lvl) (Base address: 0x30010)
0x2F9F8287: TCS Resource lookup: (Name: xo.lvl) (Base address: 0x30080)
0x2F9F89DA: TCS write (Resource: xo.lvl) (TCS.Cmd: 0.0) (hlvl: 1)
0x2F9F8BC0: TCS write (Resource: cx.lvl) (TCS.Cmd: 0.1) (hlvl: 1)
0x2F9F8DE2: TCS write (Resource: mx.lvl) (TCS.Cmd: 0.2) (hlvl: 1)
0x2F9F9195: TCS write (Resource: xo.lvl) (TCS.Cmd: 1.0) (hlvl: 1)
0x2F9F9381: TCS write (Resource: cx.lvl) (TCS.Cmd: 1.1) (hlvl: 0)
0x2F9F969E: TCS write (Resource: mx.lvl) (TCS.Cmd: 1.2) (hlvl: 1)
0x2F9F9903: TCS write (Resource: mx.lvl) (TCS.Cmd: 2.0) (hlvl: 2)
0x2F9F9C0F: TCS write (Resource: cx.lvl) (TCS.Cmd: 2.1) (hlvl: 2)
0x2F9F9ED1: TCS write (Resource: xo.lvl) (TCS.Cmd: 2.2) (hlvl: 3)
0x2F9FA04A: TCS registers programmed successfully
0x2FB5EF50: Config subsystem int 475 (PDC bit: 0) (trigger: 4)
0x2FD0A43E: Successfully enabled 4 of 5 profiling units
```

# PDC 转储脚本（第 1 页，共 2 页）

- `pdc_dump.cmm ss=<subsystem name>`  
`[op=<output file path>]`

- 各 PDC 有关以下内容的转储信息：

- 定序器
- 触发命令集 (TCS)
- IRQ
- 参数

**注意：**子系统名称可以是 APPS、Modem、音频、传感器、AOP、调试、GPU、显示和计算

- 定序器

```
Start: 0x0
Busy: 0
PC: 0x0
Instr: 0xE1
Timer: 0x0
Pwr Override: (0x0, 0x0)
Wait Override: (0x0, 0x0)
Br Override: (0x0, 0x0)
Branches: (0x0, 0x0, 0x0, 0x0)
Delays: (0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0)
```

- TCS0

```
AMC Mode: 0
En Mask: 0x7
Wait Mask: 0
Idle: 0
CMD0
    slave: 3, addr: 0x80, data: 0x1
    resp req: 0, triggerd: 0, issued: 0,
ackd: 0
...
CMDc
...
TCSn
```

# PDC 转储脚本（第 2 页，共 2 页）

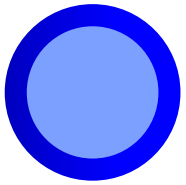
- IRQ（软件）

```
Num: (ss_irq, gpio, en, status, cfg, owner)
0:   (475,   -,   0,   0,   high,   0)
1:   (476,   -,   0,   0,   falling, 0FFFFFFF)
2:   (477,   -,   0,   0,   falling, 0FFFFFFF)
3:   (478,   -,   0,   0,   falling, 0FFFFFFF)
4:   (479,   -,   0,   0,   falling, 0FFFFFFF)
5:   (480,   -,   0,   0,   falling, 0FFFFFFF)
6:   (481,   -,   0,   0,   falling, 0FFFFFFF)
...
IRQn
```

- 参数

- 版本
  - 主要 – 1
  - 次要 – 0
- 资源
  - DRV – 2
  - TCS – 3
  - cmds/tcs – 4
  - ProfUnit – 5
  - Timestamp – 0
- 定序器
  - CMD – 96
  - XPwrCtrl – 1
  - XWaitInput – 1
  - AOP IRQ – 2
- IRQ 和 GPIO
  - IRQ – 15
  - GPIO – 96
  - GP SEL – 30





## 第 9 节

# 调试 VRM

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

# 关于 VRM

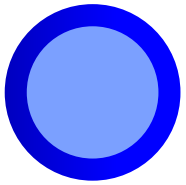
---

- VRM 与 AOP 配合，管理 PMIC 调压器和时钟缓冲区中表决的执行
- 支持对以下资源类型进行表决：
  - 调压器（使能、模式、电压和余量）
  - XOB（使能）
  - XO（使能）
- 包含当前资源状态 (CRS) 寄存器，指示处于稳定状态的最终 VRM 资源设置

# VRM 调试

---

- CMM 脚本
  - `vrn.cmm`: 显示 DRV 表决、CRS 状态和 PMIC 状态
  - `vrn_dump.cmm`: 转储 RPMH VRM 寄存器
- 有关调试方案, 参见 *Non-HLOS PMIC Voltage Regulator and Clock Software User Guide* (80-P9301-78)



## 第 10 节

# 调试客户端驱动程序 – 子系统 RPMh 驱动程序

# RPMh 驱动程序功能

---

- 管理运行系统客户端表决配置
  - 发送或管理活动或立即的表决集
  - 配置休眠或唤醒的表决集
- 跟踪软件资源表决
- 使用单一软件处理多个 DRV 和客户端

# RPMhDriverLog

---

- 由 [Qualcomm 崩溃分析门户 \(QCAP\)](#) 生成
- 转储 RPMh 驱动程序活动：
  - 发出表决集
  - 等待表决集完成
  - 管理休眠进入和退出活动

**注意：** CMM 脚本仅适用于非 HLOS。HLOS 具有专属日志。例如 Linux 使用 Ftrace。

# 示例 RPMhDriverLog

0xD917234: rpmh\_issue\_command\_set (CLIENT: ICB) (DRV\_ID: 0x0000000A) (SET: 0) (NUM\_CMDS: 1) ← DRV 0xA VOTE SET from ICB

0xD917EFD: resource\_command (ADDRESS: 0x000500A0) (DATA: 0x40000000) ) ← Resource vote information

0xD918ABB: configuring TCS (DRV\_ID: 10) (TCS\_ID: 0) (AMC?: 1)

0xD918AE3: (CLIENT: ICB) (REQ\_ID: 7)

0xD918F60: TCS SENT (TCS INDEX: 0) (TCS\_ID: 0) ← Vote triggered in hardware

0xD918F8A: req\_id (REQ\_ID: 7) ← ID for this request is 7

0xD918FD4: rpmh\_in\_flight sleep update (CMDQ\_EMPTY? == 0)

0xD919102: rpmh\_finish\_amc (HW\_DRV: 0x0000000A) (SW\_DRV: 0x0000000A) ← Callback to complete vote

0xD9191C1: TCS IDLE (TCS\_ID: 0)

0xD9191E6: finished req: (CLIENT: ICB) (REQ\_ID: 7) ← Finished ICB client request ID 7

# RPMh 驱动器调试方案

- 一组表决卡断，可能导致以下错误：
  - rpmh\_barrier() 卡断
  - RPMhDriverLog 中写入 !!!!VOTE\_SET\_IS\_STUCK!!!!
  - RPMhDriverLog 中写入 BUS\_TIMEOUT
- 如需在一组表决卡断时调试方案：
  1. 检查 RPMhDriverLog，确定卡断的请求。
  2. 查看 RSC 中表决集的内容，确认以下事项：
    - 存在未完成的命令。
    - 针对给定子系统将 PDC\_EN 位置 1。
  3. 根据发出请求的地址或客户端，建立命令适用的 RPMh 块。
  4. 为该硬件块（ARC、VRM 或 BCM）检查 RPMh 块
    - 有关示例，参见[幻灯片 62](#) 到 [幻灯片 64](#) “示例调试程序 – 表决组卡断”。



# 示例调试程序 – 表决组卡断（第 1 页，共 3 页）

```
0xD917234: rpmh_issue_command_set (CLIENT: ICB) (DRV_ID: 0x0000000A)
0xD917EFD: resource_command (ADDRESS: 0x000500A4) (DATA: 0x40000000) )
0xD918ABB: configuring TCS (DRV_ID: 10) (TCS_ID: 0) (AMC?: 1)
0xD918AE3: (CLIENT: ICB) (REQ_ID: 7)
0xD918F60: TCS SENT (TCS INDEX: 0) (TCS_ID: 0)
0xD918F8A: req_id (REQ_ID: 7)
0xD918FD4: rpmh_in_flight sleep update (CMDQ_EMPTY? == 0)
0xD919102: rpmh_barrier_all (CLIENT: ICB) (DRV_ID: 0x0000000A) (REQ_ID: 7)
```

1. 检查 RPMhDriverLog 中的以下几项，确定卡断的请求。

- 客户端
- 表决
- 调用者的跟踪 ID
- 等待跟踪 ID

## 示例调试程序 – 表决组卡断（第 2 页，共 3 页）

DRV: mss\_sw

**PDC ENABLE : 0x1**

NUM TCS: 0x6

CMDs per TCS : 0x10

TCS NUM : 0x0

TCS AMC\_MODE\_EN: 0x1

TCS AMC\_MODE\_TRIGGER: 0x1

CMD: 0x0

**ADDRESS: 0x000500A4**

DATA: 0x40000000

TYPE: WRITE

**RESPONSE REQUIRED? : 0x1**

MSG\_LENGTH : 0x8

TRIGGERED? : 0x1

ISSUED? : 0x1

**COMPLETED (only for RR cmds)? : 0x0**

CMD: 0x1 NOT ENABLED

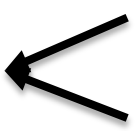
### 2. 查看表决集内容，确认以下事项：

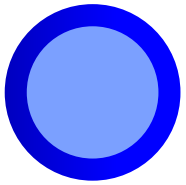
- 针对给定子系统将 PDC\_EN 置位
- 表决匹配
- 存在未完成的表决

## 示例调试程序 – 表决组卡断（第 3 页，共 3 页）

---

0xD917234: rpmh\_issue\_command\_set (CLIENT: ICB) (DRV\_ID:  
0x0000000A)

- 
- 3. 根据发出请求的地址或客户端，建立命令适用的 RPMh 块
  - 4. 检查 BCM 块的 RPMh 块



## 第 11 节

# SoC 休眠

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

# RPM – SoC 休眠

---

- 系统休眠与休眠模式紧密相关：
  - XO – 关断
  - VDD-MIN

Qualcomm  
2019-03-13 00:17:18 PDT  
zk\_sw@wingtech.com

# RPMh – SoC 休眠

---

- SoC 资源休眠状态是独立的
- 资源具备休眠状态，不具备休眠模式
  - XO – 关闭
  - VDD-CX
    - 保持
    - 深度休眠（关闭）
  - VDD-MX
    - 保持
- 子系统进入低功耗状态时，显示 AOSS 的休眠状态

# 参考资料

| 标题  | 文档号              |
|---|------------------|
| <b>Qualcomm Technologies, Inc.</b>  |                  |
| <i>Non-HLOS Voltage Regulator Software User Guide</i>   | 80-P9301-78      |
| <i>DDR Debugging and Test Guide</i>   | 80-NR123-1       |
| <i>SDM845 NON-HLOS PMIC Software Overview</i>   | 80-P9301-26      |
| <i>SDM845 RPM FAQs</i>  | KBA-170707163551 |
| <b>资源</b>   |                  |
| <a href="https://www.python.org/">https://www.python.org/</a>   | —                |
| <a href="https://bitbucket.org/pplesnar/pyelftools-pp">https://bitbucket.org/pplesnar/pyelftools-pp</a> | —                |
| <a href="https://cap.qti.qualcomm.com/">https://cap.qti.qualcomm.com/</a>                               | —                |

# 缩略词

| 缩略词或术语 | 定义  |
|--------|---|
| AMC    | 激活模式控制器 (Active mode controller)                        |
| ARC    | 聚合资源控制器 (Aggregated resource controller)                |
| AOP    | 实时响应处理器 (Always-on processor)                           |
| AOSS   | 实时响应子系统 (Always-on subsystem)                           |
| BCM    | 总线时钟管理器 (Bus clock manager)                             |
| CPRF   | 核心功率降低辅助器 (Core power reduction facilitator)            |
| CRS    | 当前资源状态 (Current resource state)                         |
| DRV    | 直接资源表决器或客户端 (Direct resource voters or clients)         |
| IRQ    | 中断请求 (Interrupt request)                                |
| NPA    | 节点电源架构 (Node power architecture)                        |
| PCB    | 电源控制总线 (Power control bus)                              |
| PDC    | 电源域控制器 (Power domain controller)                        |
| RBCPR  | Rapid Bridge 核心功率降低 (Rapid Bridge core power reduction) |
| RSC    | 资源状态协调器 (Resource state coordinator)                    |
| SLVR   | 休眠模式解算器 (Sleep mode solver)                             |



# 缩略词（续）

| 缩略词或术语 | 定义   |
|--------|--|
| TCS    | 触发命令集 (Triggered command set)                    |
| VCD    | 虚拟时钟域 (Virtual clock domain)                     |
| VRM    | 调压器管理器 (Voltage regulator manager)               |
| QCAP   | Qualcomm 崩溃分析门户 (Qualcomm crash analysis portal) |
| QUP    | Qualcomm 通用外设 (Qualcomm universal peripheral)    |

# 问题？

欲了解更多信息或者存在技术问题，可访问：<https://createpoint.qti.qualcomm.com>