**定屏死机问题分析**

Version: 0.1

Date: 2014-12-07



[www.spreadtrum.com](http://www.spreadtrum.com)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Author | Description |
| 0.1 | 2014-12-07 | Congfu.zhao | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

目录

[Revision History 2](#_Toc406424266)

[Contents 3](#_Toc406424267)

[一、 定屏死机类型及checkpoint 5](#_Toc406424268)

[1 定屏死机类型 5](#_Toc406424269)

[2 checkpoint 5](#_Toc406424270)

[3 T32常用命令说明 5](#_Toc406424271)

[二、 定屏死机分析 6](#_Toc406424272)

[1、 checkpoint解析 6](#_Toc406424273)

[2、 分析思路 12](#_Toc406424274)

[三、 案例分析 12](#_Toc406424275)

[1 DDR通道busy导致的定屏 12](#_Toc406424276)

[2 访问IO挂死 13](#_Toc406424277)

[3 phy掉电失效芯片 14](#_Toc406424278)

[4 中断频繁导致定屏 15](#_Toc406424279)

[5 进程UN状态导致定屏 15](#_Toc406424280)

[6 mutex\_lock死锁导致的定屏 16](#_Toc406424281)

[7 FM后台运行，唤不醒的问题 16](#_Toc406424282)

[8 DCDC core电压问题 17](#_Toc406424283)

[四、 附录 18](#_Toc406424284)

[1 CoreSight Debug Access Port ”DAP” 18](#_Toc406424285)

[2 Wakeup流程图 19](#_Toc406424286)

## 一、 定屏死机类型及checkpoint

### 1 定屏死机类型

基本可以归为三类：

1. 硬件或芯片缺陷，芯片相关配置操作不完善导致的定屏死机；
2. Kernel bug导致的定屏死机；
3. Android 上层导致的定屏；

### 2 checkpoint

|  |  |
| --- | --- |
| 1 | 确定adb是否可用 |
| 2 | 确定串口shell是否可用 |
| 3 | Power键是否可以亮灭屏 |
| 4 | 连接t32，若没有t32手动sysdump，跳过下面步骤，直接分析sysdump |
| 5 | 连接t32，在prepare状态下初步判断pc运行的位置 |
| 6 | 查看DDR controller寄存器0x300003fc，确定各通道未挂住 |
| 7 | 大体判断一级页表内核部分是否正常 |
| 8 | 查看代码段是否正常，内存中代码段与vmlinux代码对比 |
| 9 | 查看sprd\_debug\_last\_regs\_access，判断是否IO挂死 |
| 10 | 查看sprd\_debug\_log，判断中断是否正常 |
| 11 | 检查cpsr/spsr/sp是否有明显异常 |
| 12 | 通过sysdump或t32分析具体原因 |

### 3 T32常用命令说明

1. edap/eahb和直接输入地址单元访问地址的区别

edap主要用来查看pc运行情况，目前只知道80030084地址对应pc运行位置(也即运行到code哪条指令上)。eahb可以在线访问内存和寄存器地址空间的内容，所谓在线就是系统运行的同时，可以同时访问地址空间，可以直接访问物理地址空间。当attach后break下来，可以通过arm访问虚拟地址空间，只能访问虚拟地址，这个访问是经过mmu的。

1. 关于prepare/up/attach的区别

Prepare : 初始化JTAGA interface，但是不连接arm core。

Up ：初始化JTAGA interface，设置CPU到debug 模式，并将cpu停止在reset vector。

Attach: 初始化JTAGA interface，设置CPU到debug 模式，但是不停止cpu，cpu仍然在运行。

## 二、 定屏死机分析

### 1、 checkpoint解析

1.1 adb是否可用

adb连接上后，可以先把slog pull出来，分析log。一般adb可以连接基本上定屏属于android framework及以上的问题。

执行: adb shell bugreport > pc目录

1.2 串口shell是否可用

可以用串口shell在线查看log，查看进程，内存等信息，查用命令

shell@scx35\_sp7731gea:/ $ su

shell@scx35\_sp7731gea:/ # dmesg //查看kernel log

shell@scx35\_sp7731gea:/ # logcat -v time //查看android log

shell@scx35\_sp7731gea:/ # ps //查看进程信息

shell@scx35\_sp7731gea:/ # cat proc/meminfo //查看内存信息

shell@scx35\_sp7731gea:/ # procrank

shell@scx35\_sp7731gea:/ # top

shell@scx35\_sp7731gea:/ # debuggerd -b 596 //dump 进程栈信息

1.3 按powerkey是否可以亮灭屏

分为三情况，a.可以亮灭屏，应该只是上层某个地方阻塞；b.不能亮灭屏，但是按powerkey后可以在kernel log中看到对应的log，说明kernel是正常的；c.不能亮灭屏，kernel log中没有对应的log，整个系统都挂死了。

1.4 连接t32或手动sysdump

若具备条件，建议先用t32连上分析，若没有条件，需要手动sysdump。

手动sysdump分两种情况：a.定屏时，若kernel还能响应中断，组合键(同时按上音量键+下音量键，点击两次powerkey)产生sysdump，此时会走正常sysdump流程，保存一些关键信息。b.长按powerkey，7s chip reset，这是硬件重启，sysdump仅包含内存信息。

1.5 连接t32 prepare状态下初步判断pc的位置

在t32的command line下输入，查看pc运行的位置

d.dump edap:80030084 //core0

d.dump edap:80032084 //core1

d.dump edap:80034084 //core2

d.dump edap:80036084 //core3

如下是不同运行阶段对应的地址空间，我们可以通过这些地址范围，大概确定目前系统跑到什么阶段，具体运行到的函数，可以通过map文件或者符号表找到：

rom code段：0xffff 0000 ~ 0xffff 3d50

fdl1 code段：0x5000 04c0 ~ 0x5000 4fbc

fdl2 code 段：0x8000 0000 ~ 0x8002 b934

spl code段: 0x5000 5000 ~ 0x5000 9b40

uboot code段：0x8f80 0000 ~ 0x8f87 40dc

kernel code 段：0xc000 9000 ~ 0xc05b ca9c

Arm address map:

Arm 内部ram(ap)区域：0x0000 0000 ~ 0x0000 ffff

Arm external memory space: 0x8000 0000 ~ 0xffdf ffff

Ap cp share iram: 0x5000 0000 ~ 0x5fff ffff

Ap sleep code 区域: 0x400 ~ 0xc00

Ap dfs code区域: 0x1000 ~ 0x1c00

1.6 查看DDR通道状况

在t32的command line中输入，查看ddr通道的状态：

d.dump eahb:0x3000 03fc

若其中任意bit一直为1则，说明DDR通道挂住了。其中高16bit是写通道，低16bit是读通道，若只是写通道挂住只会导致定屏，而读通道挂住会导致白屏。

读通道

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 |
| mm/dcam/vsp | GPU | Display/gsp | CA7 | CPx DSP | CP0W | CP0 ARM | AP matrix |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 |
| CP1 ARM | CP2 |  |  |  |  |  |  |

写通道

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 |
| mm/dcam/vsp | GPU | Display/gsp | CA7 | CPx DSP | CP0W | CP0 ARM | CP1 ARM |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 |
| CP1 ARM | CP2 |  |  |  |  |  |  |

产生DDR通道busy的原因：上述任意一个通道正在访问DDR时，该通道的控制器被下电或者被reset会导致这个通道访问DDR挂死，这时若其它通道再访问DDR也会被挂住。

**注意：**

当你发现DDR通道挂住时，此时你不能用t32 eahb命令访问内存区域，否则会导致整个地址空间都不能访问，都会变成问号。虽然不能访问内存，但是仍然可以访问寄存器。这时可以看一下PMU寄存器各core的上电和sleep情况：

各core和模块的上电情况：

0x402b 00bc ~ 0x402b 00c8 //0表示上电，7表示下电

各core的sleep情况：

0x402b 00d4 //5表示wakeup，0表示sleep

CA7 standby 状态：

0x20d0 3034 //每个core对应1个bit，1为 对应的standby状态

具体内容请参考内网svn tshark spec.

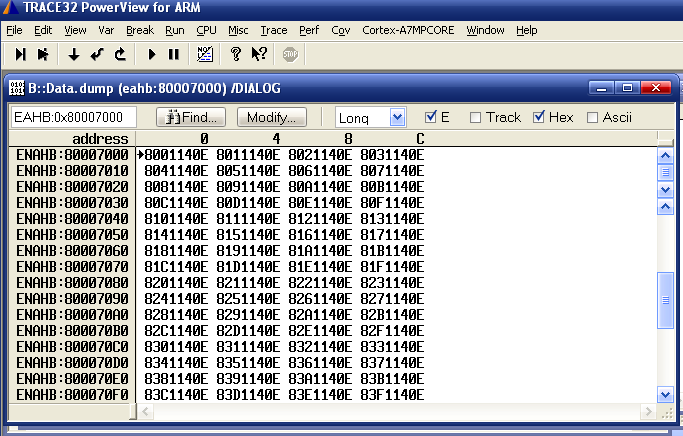
1.7 查看内存中一级页表是否正常

目前一级页表起始地址是0x80007000，对于512m的内存，其页表范围为，

0x8000 7000 ~ 0x8000 7800

在t32的command line中输入，如下图所示，正常的页表是规律递增的。

d.dump eahb:0x8000 7000



对于页表被破坏的情况，需要具体问题具体分析。

1.8 查看内存代码段与符号表是否一致

执行如下步骤即可以对比出内存代码段与符号表是否一致

a.arm-eabi-objdump -h vmlinux | less //得到text段的size大小，将size除以4；

b. 在crash中>rd 0xc0009000 size/4 > textfromelf；//从内存中读出code段内容

c. cp textfromelf到memory dump的同一目录下；

d. 运行sprdumpseg脚本./sprdumpseg text；

e. vimdiff text\*



对比代码段是一个比较有效的手段，建议无论在分析sysdump或者现场分析问题，都可以执行这个步骤，确定是否有bitflip或者内存覆盖。

1.9查看sprd\_debug\_last\_regs\_access，判断是否IO挂死

IO挂死的原因，当模块控制器的clk被关闭后，再操作模块寄存器就会导致IO挂死，这时整个ap都会hang住，看pc已经不动了。

如下结构体说明了，每次操作寄存器我们都会记录的内容，当任意一个cpu调用\_\_raw\_writel，\_\_raw\_readl操作寄存器时，会将如下信息记录到sprd\_debug\_last\_regs\_access中，在操作寄存器前将status状态清0，当寄存器操作结束后，会将status置1，因此若某个寄存器未访问完成其status=0，我们可以从vaddr中发现是操作什么寄存器导致的IO挂死。sprd\_debug\_last\_regs\_access是个全局变量，直接在t32或crash打印这个变量即可。

struct sprd\_debug\_regs\_access{

u32 vaddr;

u32 value;

u32 stack;

u32 pc;

unsigned long time;

unsigned int status;

};

1.10 查看sprd\_debug\_log，分析调度、中断、work是否正常

如下是sprd\_debug\_log的结构体，可以看到这个结构体记录了task调度信息，中断信息，work调度，hrtimers调度信息。

对与定屏，遇到最多的是某个中断一直往上报，导致其它任务无法执行，这时可以查看sprd\_debug\_log的irq信息，若一直记录的是一个中断，这个中断可能就是引起定屏的原因。

struct sched\_log {

struct task\_log {

unsigned long long time;

#ifdef CP\_DEBUG

unsigned long sys\_cnt;

#endif

char comm[TASK\_COMM\_LEN];

pid\_t pid;

} task[NR\_CPUS][SCHED\_LOG\_MAX];

struct irq\_log {

unsigned long long time;

#ifdef CP\_DEBUG

unsigned long sys\_cnt;

#endif

int irq;

void \*fn;

int en;

} irq[NR\_CPUS][SCHED\_LOG\_MAX];

struct work\_log {

unsigned long long time;

#ifdef CP\_DEBUG

unsigned long sys\_cnt;

#endif

struct worker \*worker;

struct work\_struct \*work;

work\_func\_t f;

} work[NR\_CPUS][SCHED\_LOG\_MAX];

struct hrtimer\_log {

unsigned long long time;

#ifdef CP\_DEBUG

unsigned long sys\_cnt;

#endif

struct hrtimer \*timer;

enum hrtimer\_restart (\*fn)(struct hrtimer \*);

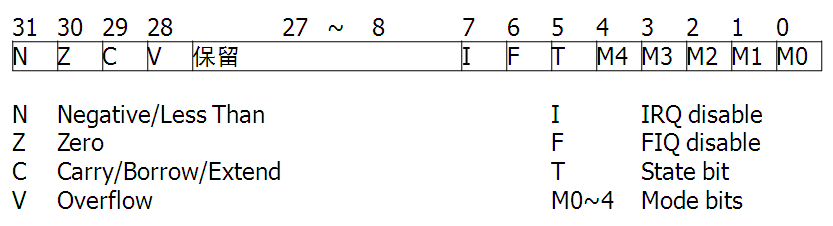
int en;

} hrtimers[NR\_CPUS][8];

};

1.11检查cpsr/spsr/sp是否有明显异常

从cpsr/spsr中我们至少可以知道当前是什么模式，前面一个是什么模式，并且当前中断有没有关闭；可以初步看一下sp对应的内存是否正常。



|  |  |  |
| --- | --- | --- |
| Mode bits | 工作模式 | 对应的寄存器 |
| 1 0000 | User模式 | PC,R14~R0,CPSR |
| 1 0001 | FIQ模式 | PC,R14\_FIQ-R8\_FIQ,R7~R0,CPSR,SPSR\_FIQ |
| 1 0010 | IRQ模式 | PC,R14\_IRQ-R13\_IRQ,R12~R0,CPSR,SPSR\_IRQ |
| 1 0011 | SUPERVISOR模式 | PC,R14\_SVC-R13\_SVC,R12~R0,CPSR,SPSR\_SVC |
| 1 0111 | ABORT模式 | PC,R14\_ABT-R13\_ABT,R12~R0,CPSR,SPSR\_ABT |
| 1 1011 | UNDEFINEED 模式 | PC,R14\_UND-R8\_UND,R12~R0,CPSR,SPSR\_UND |
| 1 1111 | SYSTEM 模式 | PC,R14-R0,CPSR |

1.12通过t32或sysdump分析具体原因

主要查看栈信息是否由于某些原因导致某个循环出不来，看\_\_log\_buf的信息是否有明显的错误log信息。

1.14 tshark脚本

如下是tshark的脚本，运行这个脚本可以完成5，6步的检查。



运行打印出来的信息主要有PLL配置，CA7配置，VDDARM，与VDDCORE电压，CA7 PC运行情况，DDR通道状态。

------------TShark CHECK V0.1------------------

------------------ PLL CHECK ------------------

MPLL 1199999 KHz

DPLL 931999 KHz

TDPLL 768000 KHz

CPLL 624000 KHz

WIFIPLL1 880000 KHz

WIFIPLL2 959999 KHz

WPLL 921599 KHz

------------------ CA7 CHECK ------------------

MCU SELECT MPLL

MCU DIV is 1/1 1199999 KHz

AXI DIV is 1/3 399999 KHz

DBG DIV is 1/6 199999 KHz

------------------ VOL CHECK ------------------

VDDARM is 956 mv (NOTE: calibration will cause some offset)

VDDCORE is 865 mv (NOTE: calibration will cause some offset)

------------------ CA7 CHECK ------------------

CPU0 Running

CPU1 Running

CPU2 Running

CPU3 Running

------------------ MEM CHECK ------------------

EMC SELECT DPLL

EMC DIV is 1/1 931999 KHz

DDR IN Normal

对于定屏比较常见的情况是：

a. pc停止不动，对应脚本打印出来的信息如下，4个core分别挂在0xC03D2EC0，0xC005FD14，0xC01FE54C，0xC059C54C的指令处。

------------------ CA7 CHECK ------------------

CPU0 Hold? PC=0x0C03D2EC0

CPU1 Hold? PC=0x0C005FD14

CPU2 Hold? PC=0x0C01FE54C

CPU3 Hold? PC=0x0C059C54C

b.DDR 通道挂住，对应打印的信息如下，

------------------ MEM CHECK ------------------

EMC SELECT DPLL

EMC DIV is 1/1 931999 KHz

DDR Port Busy? 0x03FC=0x2D90000

### 2、 分析思路

通过1,2,3步基本可以将android framework及上层的问题过滤掉大部分。对于能连t32问题，通过pc位置可以大概判断系统运行到什么阶段，对于pc不动的情况，主要有6，9。除了这两种情况还有就是停在while(1)或wfe，这时就需要加符号表分析。对于不能连接t32的问题，只能手动sysdump，首先用组合键，不行再按powerkey 7s reset。对于分析sysdump，需要check 7，8，9，10和log信息。

## 三、 案例分析

### 1 DDR通道busy导致的定屏

[Bug 333658](http://bugzilla.spreadtrum.com/bugzilla/show_bug.cgi?id=333658) - [PSST][7731GGA][Tshark][Full]进行bonnie+restart测试，手机出现定屏

这个bug是因为在ap实现了cp2上下电的功能，但是没有保证在cp2 deepsleep状态，然后对cp2进行了 reset，导致cp2的通道堵住，进而影响了其它通道。

------------TShark CHECK V0.1------------------

------------------ PLL CHECK ------------------

MPLL 1199999 KHz

DPLL 931999 KHz

TDPLL 768000 KHz

CPLL 624000 KHz

WIFIPLL1 880000 KHz

WIFIPLL2 959999 KHz

WPLL 921599 KHz

------------------ CA7 CHECK ------------------

MCU SELECT MPLL

MCU DIV is 1/1 1199999 KHz

AXI DIV is 1/3 399999 KHz

DBG DIV is 1/6 199999 KHz

------------------ VOL CHECK ------------------

VDDARM is 962 mv (NOTE: calibration will cause some offset)

VDDCORE is 862 mv (NOTE: calibration will cause some offset)

------------------ CA7 CHECK ------------------

CPU0 Hold? PC=0x0C027DE20

CPU1 Hold? PC=0x0B6F8F214

CPU2 Hold? PC=0x0B6F3D031

CPU3 Hold? PC=0x0C02A4FFC

------------------ MEM CHECK ------------------

EMC SELECT DPLL

EMC DIV is 1/1 931999 KHz

**DDR Port Busy? 0x03FC=0x2D80000 //cp2\ap matrix\cp0 arm\cpx dsp\ca7通道busy**

DDR 0x0004=0x1

DDR 0x0308=0x200A0000

DDR 0x0004=0x1

DDR 0x0308=0x300A0000

DDR 0x0004=0x1

DDR 0x0308=0x300A0000

DDR 0x0004=0x1

DDR 0x0308=0x300A0000

### 2 访问IO挂死

[Bug 305618](http://bugzilla.spreadtrum.com/bugzilla/show_bug.cgi?id=305618) - [PSST][8830GA][4.4\_3.10] 进行DC/DV压力测试，出现定屏

多媒体域的clk已经关闭，但是还有code访问多媒体模块，导致系统挂住

找到sprd\_debug\_last\_regs\_access的地址，然后dump出来，最后访问的寄存器，可以看到最后访问0xf534 0000时status=0，未完成。

\_\_\_\_\_\_\_\_address|\_\_\_\_\_\_\_\_0\_\_\_\_\_\_\_\_4\_\_\_\_\_\_\_\_8\_\_\_\_\_\_\_\_C

ENAHB:9AC55000| F5242024 00000000 C09E1EA0 C002CF80

ENAHB:9AC55010| 0011B4E8 00000001 **F5340000** 00000000

ENAHB:9AC55020| C375DCD8 C002A504 0011B4E8 **00000000**

ENAHB:9AC55030| F5242024 00000000 DBA37ED8 C002CF80

ENAHB:9AC55040| 0011B4E8 00000001 F51F4044 00000000

ENAHB:9AC55050| C692BF00 C007F910 0011B4E8 00000001

0x402e 0000 bit25=mm\_eb=0，可以看到多媒体域的clk已经关了

\_\_\_\_\_\_\_\_address|\_\_\_\_\_\_\_\_0\_\_\_\_\_\_\_\_4\_\_\_\_\_\_\_\_8\_\_\_\_\_\_\_\_C

ENAHB:402E0000|>**FDF1DF08** 00000F31 00000000 00000000

### 3 phy掉电失效芯片

[Bug 326520](http://bugzilla.spreadtrum.com/bugzilla/show_bug.cgi?id=326520) - [PSST][7730G][DT][Tshark]进行灭屏唤醒测试，手机出现黑屏

pc停止在用户空间，eahb访问都是问号，从log上看已经进入deep sleep

EDAP:80030080| 00000000 **40880170** 00000002 00000000 ....p..@........

EDAP:80030090| 00000000 00000008 00000000 00000000 ................

EDAP:800300A0| 40880170 FFFFFFEB 800000CA 00000000 p..@............

EDAP:800300B0| 00000000 00000000 00000000 00000000 ................

EDAP:800300C0| 00000000 00000000 00000000 00000000 ................

EDAP:800300D0| 00000000 00000000 00000000 00000000 ................

EDAP:800300E0| 00000000 00000000 00000000 00000000 ................

EDAP:800300F0| 00000000 00000000 00000000 00000000 ................

EDAP:80030100| 08581040 0001A024 01814860 00004100 @.X.$...`H...A..

EDAP:80030110| FFFF7FFF FFFDFFFF 00000000 00000000 ................

EDAP:80030120| 00000000 00000000 00000000 00000000 ................

EDAP:80030130| 00000000 00000000 00000000 00000000 ................

EDAP:80030140| 00F04180 00E0A004 00A04180 00E02000 .A.......A... ..

EDAP:80030150| 00F50000 00E02066 00000000 00000000 ....f ..........

EDAP:80030160| 00000000 00000000 00000000 00000000 ................

EDAP:80030170| 00000000 00000000 00000000 00000000 ................

\_\_\_\_\_\_\_\_address|\_\_\_\_\_\_\_\_0\_\_\_\_\_\_\_\_4\_\_\_\_\_\_\_\_8\_\_\_\_\_\_\_\_C\_0123456789ABCDEF

ENAHB:300003F0| ???????? ???????? ????????>???????? ????????????????

ENAHB:30000400| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000410| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000420| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000430| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000440| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000450| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000460| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000470| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000480| ???????? ???????? ???????? ???????? ????????????????

ENAHB:30000490| ???????? ???????? ???????? ???????? ????????????????

ENAHB:300004A0| ???????? ???????? ???????? ???????? ????????????????

ENAHB:300004B0| ???????? ???????? ???????? ???????? ????????????????

### 4 中断频繁导致定屏

[Bug 346173](http://bugzilla.spreadtrum.com/bugzilla/show_bug.cgi?id=346173) - [PSST][7731GEA][Tshark][Native]进行开关机测试,183#出现定屏

从栈的信息和sprd\_debug\_log中的irq项都可以看到，70号中断不停的产生，导致系统一直在处理中断，不能继续执行

-000|\_\_sprd\_debug\_irq\_log(

-001|handle\_irq\_event\_percpu(

| irq = 70

-002|handle\_irq\_event(

-003|handle\_fasteoi\_irq(

-004|generic\_handle\_irq(

| irq = 70)

-005|handle\_IRQ(

| irq = 70,

| ?)

-006|gic\_handle\_irq(

-007|\_\_irq\_svc(asm)

-->|exception

-008|start\_kernel()

-009|ZSR:0x8000806C(asm)

|

---|end of frame

### 5 进程UN状态导致定屏

[Bug 339673](http://bugzilla.spreadtrum.com/bugzilla/show_bug.cgi?id=339673) - [PSST][7731GEA][Tshark]进行开关机测试，手机出现黑屏

现场用t32看到cpu0,2,3都处于idle状态，说明已经没有任务可以调度了 ，看mem很多进程都处于freezing状态，而进程113(suspend的线程)则被调度出去了。使用crash分析时只需要执行ps | grep UN就可以把UN状态的进程打印出来。

596 153 0 d94b5c80 UN 8.4 391556 44176 system\_server

606 153 2 d9548000 UN 8.4 391556 44176 FinalizerWatchd

611 153 2 d954af80 UN 8.4 391556 44176 WindowManager

612 153 3 d9548e40 UN 8.4 391556 44176 ActivityManager

618 153 0 d9549c80 UN 8.4 391556 44176 CpuTracker

619 153 3 d954a140 UN 8.4 391556 44176 FileObserver

629 153 3 d94b57c0 UN 8.4 391556 44176 system\_server

630 153 3 d94b6140 UN 8.4 391556 44176 AlarmManager

637 153 0 d95ec980 UN 8.4 391556 44176 InputReader

### 6 mutex\_lock死锁导致的定屏

[Bug 322895](http://bugzilla.spreadtrum.com/bugzilla/show_bug.cgi?id=322895) - [PSST][7730G][DT][Tshark]进行camera录像回放删除压力测试，手机出现定屏

在dt版本中clock要求以公共框架实现，其中clk\_prepare\_enable()有mutex\_lock操作，不能关闭中断。

栈信息：

-000|\_\_mutex\_lock\_slowpath(

-001|mutex\_lock(

-002|clk\_prepare\_lock()

-003|clk\_unprepare(

-004|sprd\_sdhci\_host\_enable\_clock(

-005|sprd\_sdhci\_host\_runtime\_suspend(

-006|pm\_generic\_runtime\_suspend(

-007|\_\_rpm\_callback(

-008|rpm\_callback(

-009|rpm\_suspend(

-010|pm\_runtime\_work(

-011|process\_one\_work(

-012|worker\_thread(

-013|kthread(

-014|ret\_from\_fork(asm)

### 7 FM后台运行，唤不醒的问题

[Bug 363689](http://bugzilla.spreadtrum.com/bugzilla/show_bug.cgi?id=363689) - [PSST][7731GEA][Tshark][Full][2351]FM后台播放&反复灭屏唤醒，测试20小时以内出现唤不醒现象

现场情况：

1.#279

Edap:80030000，都是问号

Eahb:300003fc，可以访问，ddr通道未挂住

2.#222

Edap:80030000，pc停在0xc05b9054，对应函数hrtimer\_cpu\_notify的cmp指令

Eahb:300003fc，可以访问，ddr通道未挂住

3.#257

Edap:80030000，pc停在0x2c，iram区域

Eahb:300003fc，可以访问，ddr通道未挂住

4.#208

Edap:80030000，pc停在0x4c，iram区域

Eahb:300003fc，可以访问，ddr通道未挂住

原因：Tshark 由于之前LVDS PLL漏电问题，提出解决方案凡是AP可以sleep的场景，比如FM和通话都是采用AP sys 打开来减少LVDS PLL漏电。所以在这两个场景下为了功耗问题，workaround方法ap sys power on, CA7 TOP power down。这种设置会导致 AP sys 在某些corner下内部的某些logic不能在CA7 TOP power on时完全reset干净。后续芯片会修改reset相关设计，保证AP sys power on，CA7 top power down下配置能正常工作。

### 8 DCDC core电压问题

[Bug 346039](http://bugzilla.spreadtrum.com/bugzilla/show_bug.cgi?id=346039) - [PSST][7731GEA][Tshark][Native]进行sleep测试,179#出现黑屏

现场情况：

Pc跑在iram:0x568~0x574之间

Eahb访问寄存器和内存是问号

原因：测试179手机是AB芯片，当配置AB芯片DCDC core电压 sleep降压使能，wakeup升压使能，但由于芯片问题，会出现概率升压不成功的问题，导致黑屏唤不醒。所以对于AB芯片配置sleep后DCDC core是不降压。

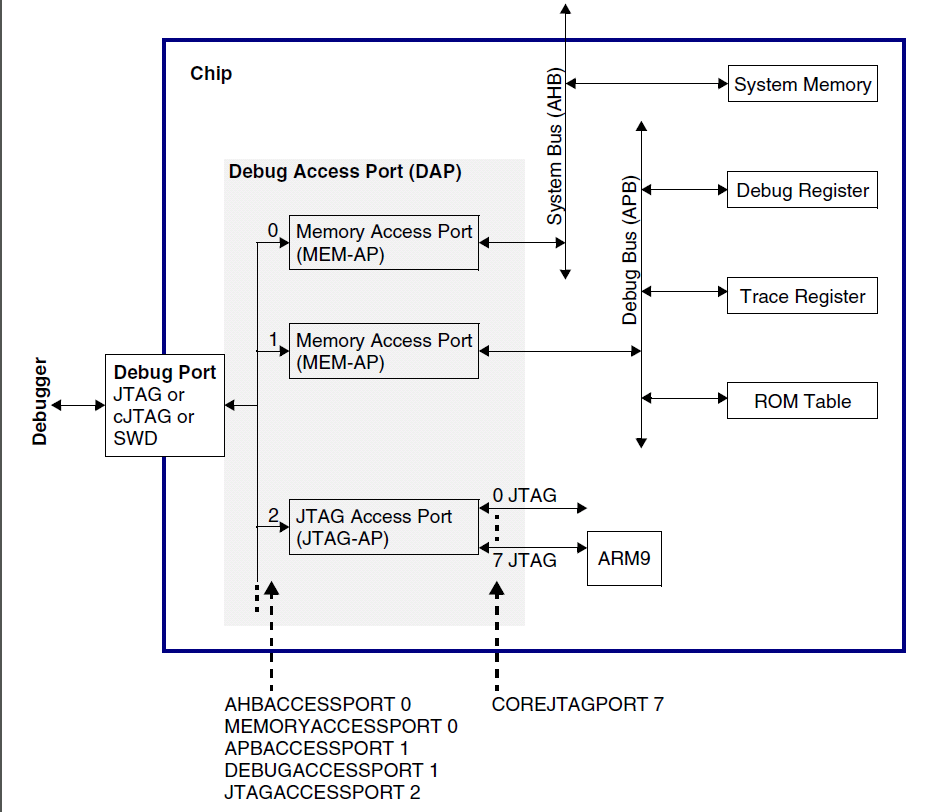
## 四、 附录

### 1 CoreSight Debug Access Port ”DAP”

A Debug Access Port (DAP) is a CoreSight module from ARM which provides access via its debugport(JTAG, cJTAG, SWD) to:

1. Different memory busses (AHB, APB, AXI). This is especially important if the on-chip debug register needs to be accessed this way. You can access the memory buses by using certain access classes with the debugger commands: “AHB:”, “APB:”, “AXI:, “DAP”, “E:”. The interface to these buses is called Memory Access Port (MEM-AP).

2. Other, chip-internal JTAG interfaces. This is especially important if the core you intend to debug is connected to such an internal JTAG interface. The module controlling these JTAG interfaces is called JTAG Access Port (JTAG-AP). Each JTAG-AP can control up to 8 internal JTAG interfaces. A port number between 0 and 7 denotes the JTAG interfaces to be addressed.



### 2 Wakeup流程图

