watchdog

系统stability检测工具

Agenda

- •1.总体流程
- 2.设备树中解析watchdog参数
- 3.设置定时器pet_timer
- 4.唤醒msm_watchdog线程
- 5.向其他CPU发送IPI中断
- 6. 狗叫

watchdog总体流程

正常喂 init_watchdog platform_driver_register 狗 msm watchdog probe msm wdog dt to pdata 从设备树中解析watchdog参数 创建msm watchdog线程 create kthread 注册bark_irq,设置watchdog寄存器的bark timer和bite time init_watchdog_data 设置定时器pet_timer,下次喂狗时间 watchdog kthread watchdog线程运行之后会进入睡眠,等待定时器时间到 给其他CPU发送ipi中断,并且等待对方回应。 ping other cpus 如果对方及时回应,说明这个CPU没有卡住 pet_watchdog 喂狗--向寄存器WDOG0 RST写1,复位watchdog计数器 mod timer 设置下次喂狗时间

解析设备树中watchdog参数

msm_wdog_dt_to_pdata

```
wdog: qcom,wdt@17c10000{
         compatible = "qcom,msm-watchdog";
         reg = \langle 0x17c10000 0x1000 \rangle; \rightarrow wdog_data.phys_base = 0x17c10000, wdog_data.size = 0x1000
         reg-names = "wdt-base";
         interrupts = <0 0 0>, <0 1 0>; →bark irg hwirg = 0x20, bite irg hwirg = 0x21(wdog data中的bark irg和bite irg是
虚拟中断号)
         qcom,bark-time = <11000>;
                                     →wdog data.bark time = 11s
         qcom,pet-time = <9360>;
                                     →wdog_data.pet_time = 9.36s
                                     →wdog data.do ipi ping = true
         qcom,ipi-ping;
         qcom,wakeup-enable;
                                     →wdog data.wakeup irq enable = true
         qcom, scandump-size = <0x10100 0x10100 0x10100 0x10100
         0x18100 0x18100 0x18100 0x18100>;
```

设置pet_timer

init_watchdog_data

init_watchdog_data

devm request irq

注册bark irq中断处理函数wdog_bark_handler

set bark_time

write bark_time to register WDT0_BARK_TIME

set pet_timer

wdog_data->pet_timer.data = wdog_data

wdog_data->pet_timer.function = pet_task_wakeup

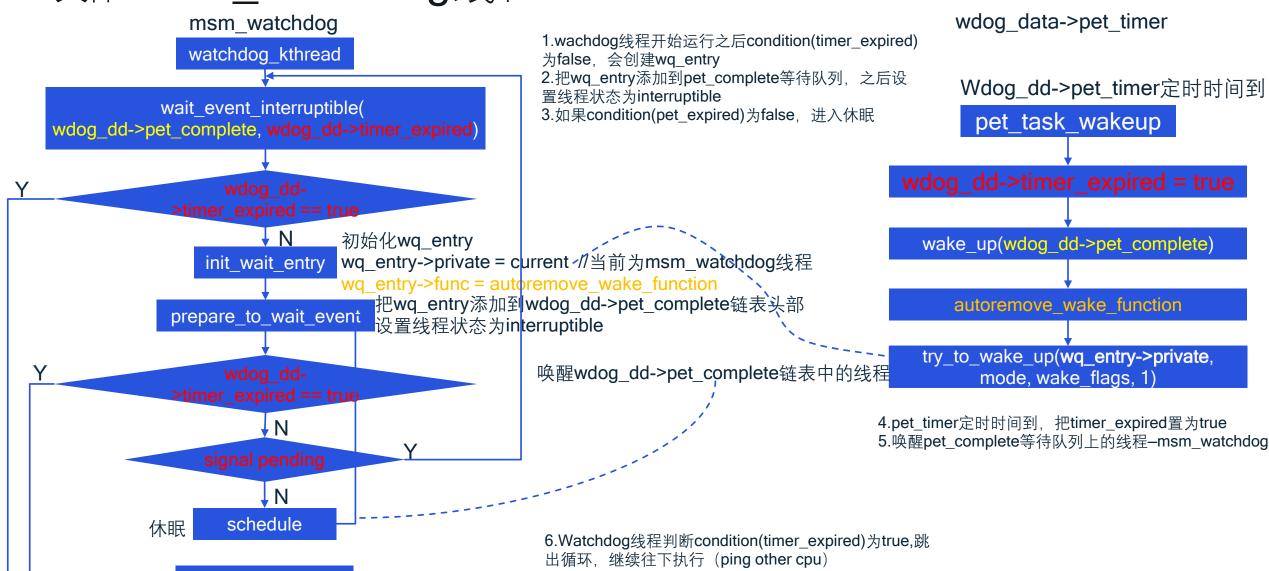
wdog_data->pet_timer.expires = jiffies+msecs_to_jiffies(wdog_data->pet_time)

first time to pet wdog

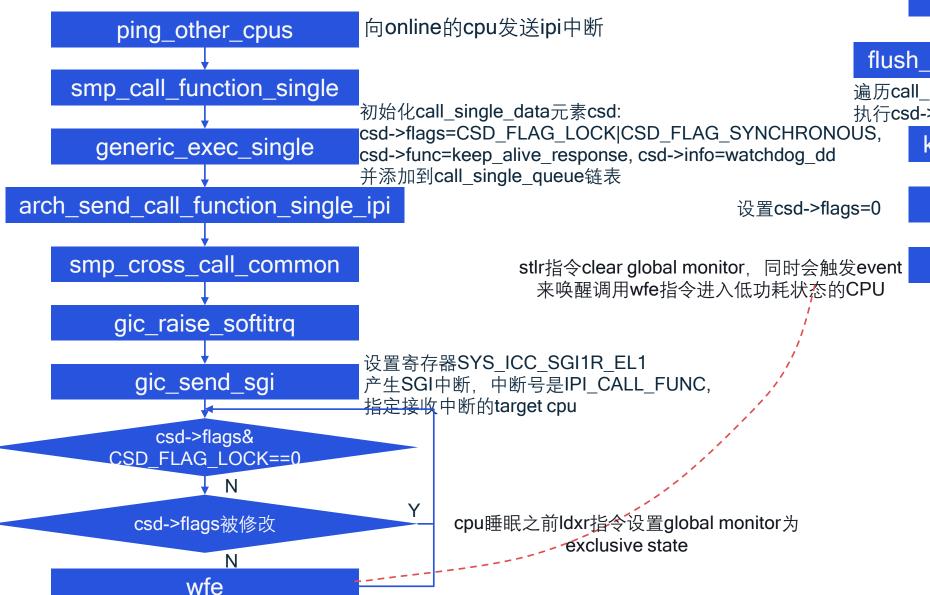
write 1 to register WDT0_RST

唤醒msm_watchdog线程

Going on



向其他cpu发送ipi中断



target CPU接收到中断

watchdog bark

wdog_bark_handler

