Linux CPUIDLE
开发指南

版本号: 1.0
发布日期: 2021.05.18

深圳市奥谷奇技术有限公司OCOCCI

版本历史

版本号	日期	制/修订人	内容描述
1.0	2021.05.18	AWA1442	1. 添加初始版本

深圳市奥谷奇技术有限公司OCOCCI

版权所有 © 珠海全志科技股份有限公司。保留一切权利

目　录

深圳市奥谷奇技术有限公司OCOCCI

版权所有 © 珠海全志科技股份有限公司。保留一切权利

1 前言

1.1 文档简介

介绍 CPUIDLE 使用方法。

1.2 目标读者

CPUIDLE 驱动开发维护人员及需要使用到 cpuidle 功能的工程师。

1.3 适用范围

表 1-1：适用产品列表

内核版本	驱动文件
Linux-4.9	drivers/cpuidle/* kernel/sched/idle.c
Linux-5.4	drivers/cpuidle/* kernel/sched/idle.c

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2 模块介绍

2.1 模块功能介绍

CPUIDLE 能让 cpu 在空间时进入低功耗模式，达到节省功耗的目的。

2.2 相关术语介绍

表 2-1: 术语介绍

术语	说明
Sunxi	指 Allwinner 的一系列 SOC 硬件平台。
CPUIDLE	让 cpu 进入低功耗状态的一种方法

2.3 模块配置介绍

2.3.1 Device Tree 配置说明

设备树中存在的是该类芯片所有平台的模块配置，设备树文件的路径为：kernel/linux-4.9/arch(RISCV 平台为 riscv)/arm64（32 位平台为 arm）/boot/dts/sunxi/CHIP.dtsi(CHIP 为研发代号，如 sun50iw10p1 等)。

- cpu 节点

```
cpu0: cpu@0 {
        device_type = "cpu";
        compatible = "arm,cortex-a53","arm,armv8";
        reg = <0x0 0x0>;
        enable-method = "psci";
        clocks = <&clk_pll_cpu>;
        clock-latency = <2000000>;
        clock-frequency = <1320000000>;
        dynamic-power-coefficient = <190>;
        operating-points-v2 = <&cpu_opp_l_table>;
        cpu-idle-states = <&CPU_SLEEP_0 &CLUSTER_SLEEP_0>;  //引用定义好的idle的状态
        #cooling-cells = <2>;
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
};

cpu@1 {
        device_type = "cpu";
        compatible = "arm,cortex-a53","arm,armv8";
        reg = <0x0 0x1>;
        enable-method = "psci";
        clocks = <&clk_pll_cpu>;
        clock-frequency = <1320000000>;
        operating-points-v2 = <&cpu_opp_l_table>;
        cpu-idle-states = <&CPU_SLEEP_0 &CLUSTER_SLEEP_0>;  //引用定义好的idle的状态
        #cooling-cells = <2>;
};
```

- psci 节点：

```
psci {
        compatible = "arm,psci-1.0";
        method = "smc";
}

cpuidle的实现需要通过psci，如果没有定义psci节点，cpuidle功能就无法实现。
```

- idle-states 节点

```
idle-states {
        entry-method = "arm,psci";                      //说明通过psci方式进入退出cpuidle

        CPU_SLEEP_0: cpu-sleep-0 {
                compatible = "arm,idle-state";          //匹配psci-idle或arm-idle驱动
                arm,psci-suspend-param = <0x0010000>;   //PSCI传递参数，存储了power_state信息，
                                                        对cpuidle来说，bit24用于区分哪种掉电方式

                entry-latency-us = <46>;                //进入该cpuidle状态的时间，
                                                        由软件进入时间和硬件进入时间组成

                exit-latency-us = <59>;                 //退出该cpuidle状态的时间，
                                                        由软件退出时间和硬件退出时间组成

                min-residency-us = <3570>;              //在该cpuidle状态的最小驻留时间，
                                                        一旦小于时间idle反而会增加功耗

                local-timer-stop;                       //指示在进入cpuidle时，
                                                        是否需要关闭本地的timer
        };

        CLUSTER_SLEEP_0: cluster-sleep-0 {
                compatible = "arm,idle-state";
                arm,psci-suspend-param = <0x1010000>;
                entry-latency-us = <47>;
                exit-latency-us = <74>;
                min-residency-us = <5000>;
                local-timer-stop;

        }
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
}
```

- timer 节点

```
timer@3009000 {
        compatible = "allwinner,sun4i-a10-timer";
        /*
         * FIXME: After using sunxi timer driver, the number
         * of CPU entering idle becomes less?
         * "allwinner,sunxi-timer";
         */
        reg = <0x0 0x03009000 0x0 0x90>;
        interrupts = <GIC_SPI 51 IRQ_TYPE_LEVEL_HIGH>;
        clocks = <&dcxo24M>;
};

cpu进入idle后需要不定期通过tick进行唤醒，但是如果在idle-state节点中定义了local-timer-stop属性就会导致
    cpu本地的timer被关闭，出现没有外部中断来临就无法退出中断的情况，这种情况下就需要将一个timer变为
    broadcast-timer，用来一段时间后让cpu退出idle状态。SUNXI平台使用timer（部分soc平台中叫soc_timer）
    来作为broadcast-timer，所以使用cpuidle功能需要配置timer节点并加载timer驱动。
```

2.3.2 board.dts 配置说明

board.dts 用于保存每一个板级平台的设备信息（如 demo 板，perf1 板等），里面的配置信息会覆盖上面的 Device Tree 默认配置信息。

cpuidle 模块在 board.dts 中无用户可用配置。

2.3.3 sysconfig 配置说明

cpuidle 模块在 sysconfig 中无用户可用配置。

2.3.4 kernel menuconfig 配置说明

linux-4.9 内核版本，进入 linux 目录，执行：make ARCH=arm64 menuconfig(32 位系统为 make ARCH=arm menuconfig) 进入配置主界面
Linux-5.4 内核版本，进入 longan 目录，执行：./build.sh menuconfig 进入配置主界面，并按以下步骤操作。

```
CPU Power Management --->
    CPU Idle --->
        [*] CPU idle PM support

CPU Power Management --->
    CPU Idle --->
        ARM CPU Idle Drivers --->
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
        [*] PSCI CPU idle Driver
```

最终配置效果如下图：

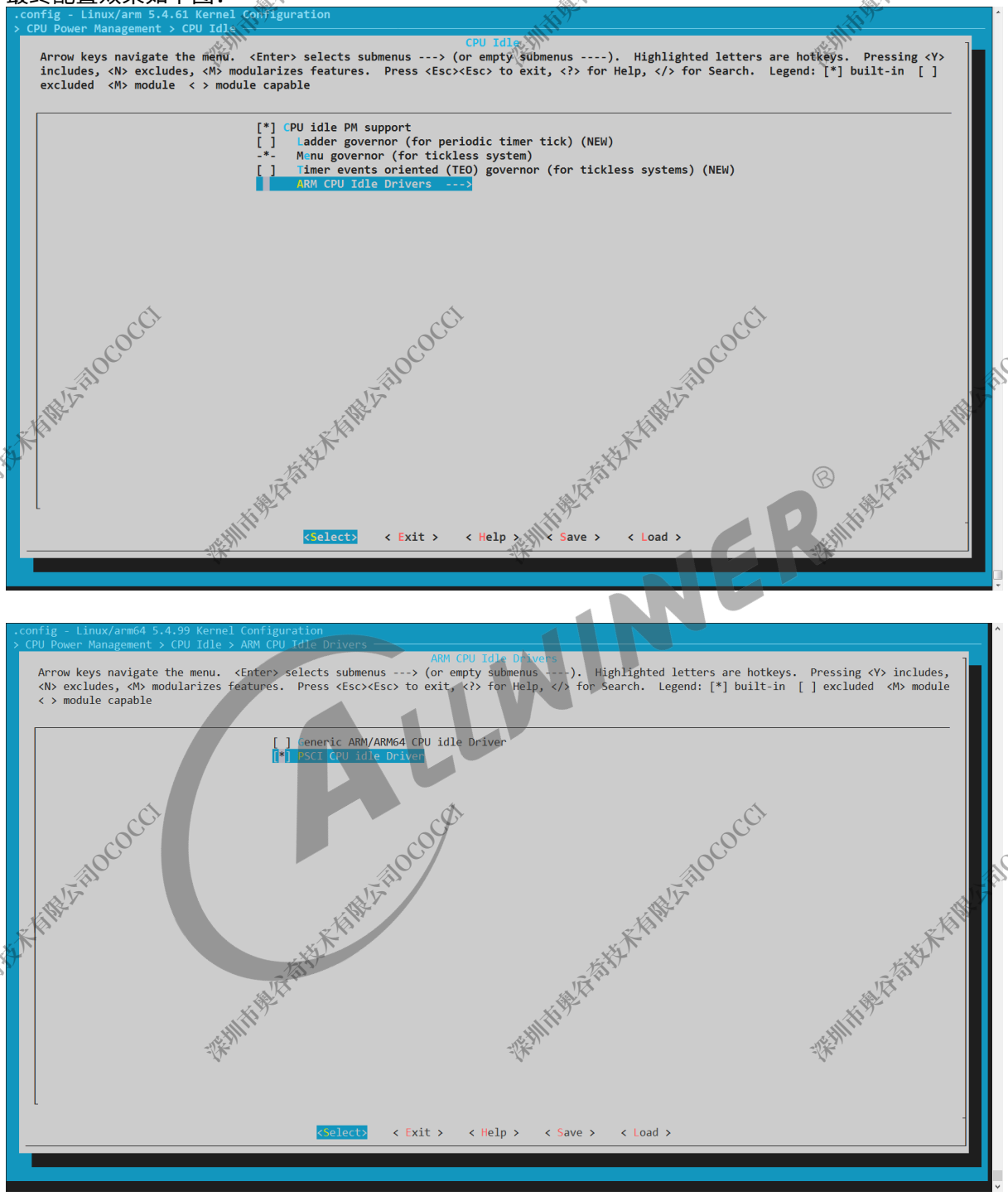

图 2-1: cpuidle 配置-2

2.4 源码结构介绍

CPUIDLE 的源代码位于内核 drivers/cpufreq/目录下：

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
drivers/cpuidle
├── cpuidle.c            //cpuidle的core代码
├── cpuidle-psci.c       //依赖psci来实现的cpuidle驱动
├── governor.c           //为governor提供公共API的代码
└── governors/           //该目录里面存放了内核主要用到的几个cpuidle governor，目前SUNXI平台使用menu
```

2.5 驱动框架介绍

无。

深圳市奥谷奇技术有限公司OCOCCI

3 FAQ

3.1 调试方法

3.1.1 调试节点

节点	权限	说明
desc	R	描述当前 idle state
name	R	当前 idle state 的名字
disable	R/W	当前 idle-state 的使能状态，默认为 0，写 1 为不使能
latency	R	延时，读出来的数值为进入延时和退出延时之和
residency	R	驻留时间，读出来的数值为该 state 的最小驻留时间
time	R	当前 cpu 进入该 idle-state 的时间总和
usage	R	当前 cpu 进入该 idle-state 的总次数
power	R	该 idle-state 下 cpu 硬件消耗的功耗，默认为 0

```
节点位于/sys/devices/system/cpu/cpuX/cpuidle/stateX下，每个cpu的每个state都有自己的一套节点，用于描述该idle的信息和状态
```

3.2 常见问题

3.2.1 cpuidle 中的 usage 计数不会增长

该问题出现大多由于 dts 配置错误或 timer 驱动支持异常导致。

3.2.1.1 dts 配置错误

常见的 dts 配置错误有：

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- soc_timer 时钟源引用错误或与实际硬件没对上，常见 FPGA 版型上最高支持 32K 时钟，此时就不能引用 24M 的时钟作为时钟源
- 32 位平台中，timer_arch 中未加上 **arm,cpu-registers-not-fw-configured;**
- timer_arch 未加上 **arm,no-tick-in-suspend;**
- idle-states 中的 entry-latency-us、exit-latency-us、min-residency-us 属性大小配置错误，与当前时钟源不匹配。在调试阶段可人为调大调小这部分参数进行验证。

3.2.1.2 timer 驱动支持异常

常见的 timer 驱动支持异常有：

- timer 驱动未支持上
- timer 驱动类型错误，当前的 timer 驱动有 sun4i_timer 和 sunxi_timer 两种，需与负责 timer 驱动的同事确保 timer 驱动正常加载
- 时钟或模块总线被 gating 住，使得挂在总线上的 timer 无法正常使用，idle 无法获知下一个来临的 tick，导致进出 idle 异常，需找负责 timer 驱动的同事进行确认

3.2.2 如何关闭 cpuidle

cpuidle 的每个 state 都提供了 diable 节点，往 disable 节点写 1 即可关闭当前 idle-state。如需关闭所有 cpu 的 state，需要手动依次对各个 state 目录下的 disable 节点写 1。**需要特别注意的是，WFI（即 state0）默认无法关闭，即使对 disable 写 1 也无法关闭。**

- 关闭 cpu1 的 idle state1

```
echo 1 > sys/devices/system/cpu/cpu1/cpuidle/state1/disable
```

- 关闭 cpu1 的全部 idle state

```
echo 1 > sys/devices/system/cpu/cpu1/cpuidle/state*/disable
```

- 关闭全部 cpu 的全部 idle state

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
echo 1 > sys/devices/system/cpu/cpu*/cpuidle/state*/disable
```

- 获取系统当前 cpu1 的 idle state1 状态，为 0 即该 idle state 开启，为 1 即该 idle state 关闭

```
# cat sys/devices/system/cpu/cpu1/cpuidle/state1/disable
0
```

- 获取系统当前所有 cpu 的 idle state 状态，为 0 即该 idle state 开启，为 1 即该 idle state 关闭

```
# cat sys/devices/system/cpu/cpu*/cpuidle/state*/disable
1           //cpu0的state0虽然写了关闭，但是功能实际还是开启
1           //cpu0的state1关闭
1           //cpu0的state2关闭
0           //cpu1的state0打开
1           //cpu1的state1关闭
1           //cpu0的state2关闭
0
0
0
0
0
0
```

深圳市奥谷奇技术有限公司OCOCCI

著作权声明

版权所有 © 2021 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明


、 
、全志科技、  （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。