Tina Linux 调频系统
使用指南

版本号: 1.4
发布日期: 2024.11.06

版本历史

版本号	日期	制/修订人	内容描述
1.0	2022.11.16	AWA1812	初始版本。
1.1	2023.04.25	AWA1812	**第一章 前言** 1.3 小节，适用范围增加 MR527。
1.2	2023.11.22	AWA1812	**第一章 前言** 1.3 小节，适用范围增加 AI985。 **第二章 模块介绍** 2.3 小节，更新调频模块内核配置说明。
1.3	2024.10.16	AWA1812	**第一章 前言** 1.3 小节，适用范围增加 V821，修改列表排版。 **第四章 FAQ** 4.5 小节，新增一号通 FAQ 列表小节。
1.4	2024.11.06	AWA2253	**第二章 模块介绍** 2.1 小节，增加模块框架介绍内容。 2.2 小节，完善 Device Tree 相关描述。

目　录

1 前言 **1**
1.1 文档简介 . . . 1
1.2 目标读者 . . . 1
1.3 适用范围 . . . 1
1.4 文档约定 . . . 2
1.4.1 标志说明 . . . 2
1.4.2 地址与数据描述方法约定 . . . 2
1.4.3 数值单位约定 . . . 2
1.5 相关术语介绍 . . . 3

2 模块介绍 **4**
2.1 模块框架介绍 . . . 4
2.2 模块配置介绍 . . . 6
2.2.1 Device Tree 配置说明 . . . 6
2.2.1.1 OPP table 类型 1 . . . 7
2.2.1.2 OPP table 类型 2 . . . 8
2.2.1.3 OPP table 类型 3 . . . 9
2.2.1.4 CPU 配置 . . . 11
2.2.2 kernel menuconfig 配置说明 . . . 12
2.2.2.1 Tina 4.0 . . . 12
2.2.2.2 Tina 5.0 . . . 12
2.3 源码结构介绍 . . . 13

3 调试方法 **14**
3.1 调试节点 . . . 14

4 FAQ **15**
4.1 更改调频策略 . . . 15
4.2 用户空间调频 . . . 15
4.3 获取当前使用的 VF 表 . . . 16
4.4 更改 VF 表 . . . 16
4.5 一号通 FAQ 列表 . . . 17

表　格

表 1-1　适用产品列表 1
表 1-2　地址与数据描述方法约定 2
表 1-3　数值单位约定 2
表 1-4　术语介绍 3
表 2-1　cpufreq 调频策略说明 6
表 3-1　cpufreq 调试节点说明前缀是 scaling 的属性文件表示软件可调节的几种属性，前缀是 cpuinfo 的属性文件表示硬件支持的几种属性。 14

1 前言

1.1 文档简介

此文档介绍 Tina Linux 调频系统的功能、配置及调试方法，以便于相关人员查看。

1.2 目标读者

Tina Linux 平台的客户及相关技术人员。

1.3 适用范围

表 1-1: 适用产品列表

产品名称	内核版本
R328	Linux-4.9
R818	Linux-4.9
R818B	Linux-4.9
MR813	Linux-4.9
MR813B	Linux-4.9
R329	Linux-4.9
R528	Linux-5.4
T113	Linux-5.4
D1-H	Linux-5.4
V833	Linux-4.9
V853	Linux-4.9
R853	Linux-4.9
MR527	Linux-5.15
AI985	Linux-5.15
MR536	Linux-5.15
V821	Linux-5.4

1.4 文档约定

1.4.1 标志说明

⚠ 注意

- **提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。**

说明

为准确理解文中指令、正确实施操作而提供的补充或强调信息。

技巧

一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

1.4.2 地址与数据描述方法约定

本文档在描述地址、数据时遵循如下约定：

表 1-2: 地址与数据描述方法约定

符号	例子	说明
0x	0x0200	地址或数据以 16 进制表示。
0b	0b010	数据采用二进制表示 (寄存器描述除外)。
X	00X	数据描述中，X 代表 0 或 1，例如，00X 代表 000 或 001。

1.4.3 数值单位约定

本文档在描述数据容量（如 NAND 容量）时，单位词头代表的是 1024 的倍数；描述频率、数据速率等时则代表的是 1000 的倍数。具体如下：

表 1-3: 数值单位约定

类型	符号	对应数值
数据容量	1 K	1024
数据容量	1 M	1 048 576
数据容量	1 G	1 073 741 824
频率，数据速率等	1 K	1000
频率，数据速率等	1 M	1 000 000
频率，数据速率等	1 G	1 000 000 000

1.5 相关术语介绍

表 1-4: 术语介绍

术语	说明
sunxi	指 Allwinner 的一系列 SOC 硬件平台
cpufreq	cpu 频率电压动态调整
DVFS	动态频率电压调整
VF 表	电压-频率表
OPP	Operating Performance Points，描述一种频率和电压配置下的性能情况
QoS	Quality of Service，管理资源申请与分配的机制，让模块在共享资源时能按需获得合适服务质量

2 模块介绍

cpufreq 功能负责系统运行过程中 CPU 频率和电压的动态调整。

cpufreq 通过 cpufreq core、cpufreq governor 内部模块统筹频率电压调节逻辑，借助 sysfs 向用户空间提供 CPU 频率的查询与控制接口，并通过内核的 notifier 机制通知关心频率变化的驱动。而对频率及电压的控制则是基于内核的 cpu subsystem、OPP、clock framework、regulator framework 以及 QOS 的 API，由 cpufreq driver 完成。

2.1 模块框架介绍

cpufreq 框架主要由三部分组成：cpufreq core、cpufreq governor、cpufreq driver。

cpufreq framework 抽象出了 policy、driver、governor 等软件实体，在运作过程中，policy 给出频率范围，governor 选取频率，driver 操作硬件的频率及电压。

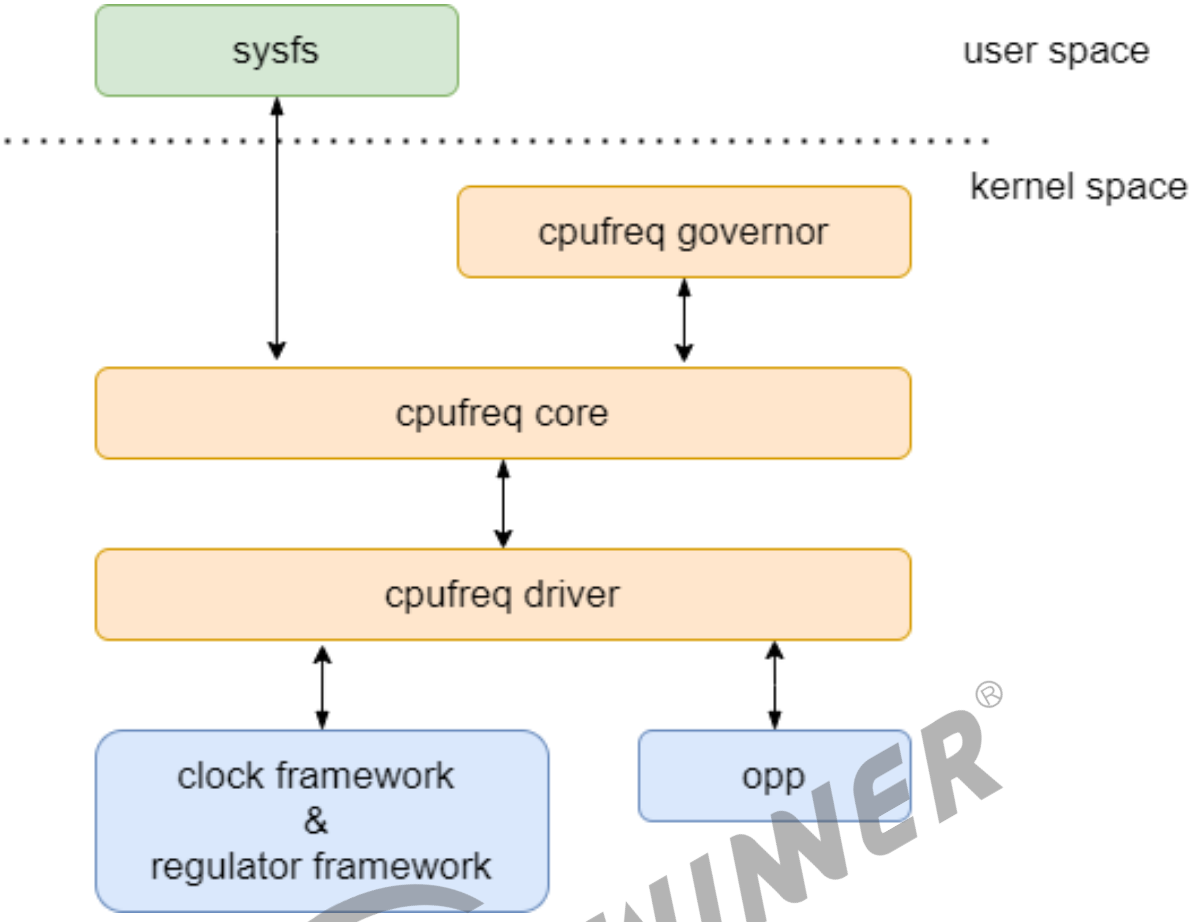

图 2-1: cpufreq 运作框架

- cpufreq core：

cpufreq 功能核心逻辑层，提供数据结构定义及 API，是 cpufreq 运作的框架，core 层协调各层实现调频调压功能，并通过 sysfs 与用户空间交互。

- cpufreq driver

负责与硬件相关的频率电压设置、调用其他层提供的接口实现调频调压功能的驱动，具有平台差异，一般是通过 OPP interface 初始化 VF 表，使用 clk framework 接口设置 CPU 频率，以及调用 regulator framework 接口设置 CPU 电压。

- cpufreq governor：

统筹调频策略，在 policy 指定的频率范围中，计算并选择 cpu 运行频率，传递给 driver 层进行实际调频调压动作。

常用的调频策略如下：

表 2-1: cpufreq 调频策略说明

governor	说明
powersace	节能，始终选择最低频率
performance	性能，始终选择最高频率
userspace	由用户控制，用户通过对应接口调节频率
ondemand	按需，不限制调频幅度
conservative	保守，平滑升降
schedutil	利用调度器提供信息调频，与 EAS 调度器配合使用

- powersave 和 performance 这两种 governors 都属于静态 governor ，在使用它们时 CPU 的运行频率不会根据系统运行时负载的变化动态作出调整。这两种 governors 对应的是两种极端的应用场景，使用 performance governor 体现的是对系统高性能的最大追求，而使用 powersave governor 则是对系统低功耗的最大追求。
- ondemand 快速响应、定期检查负载，自动根据负载来调节频率。具体来说，当负载超过一定程度时，会将 CPU 的频率调节到最高，否则会按照负载比例调节相应频率。由于是快速响应，如果负载变化过快，可能会导致 CPU 频繁地升频降频，进而造成性能抖动。
- conservative 跟 ondemand 方式类似，不同之处在于 conservative 会平滑地调整 CPU 频率，频率是渐进式的升降。这种降频策略的主导思想是尽量减小对系统性能的负面影响，从而不会使得系统性能在短时间内迅速降低以影响用户体验。
- performance、powersave 和 userspace 都是定频执行，只有 ondemand 和 conservative 两个频率调节器是根据负载动态调节频率的。
- ondemand、conservation 都需要定期采样以计算 CPU 负载，具有一定的滞后性且精度有限，而 schedutil 直接追踪 CPU 负载信息，省去了采样，使调频能更快速。另外，schedutil 还支持中断上下文直接切换频率的机制，可以进一步缩短调频时延。

2.2 模块配置介绍

2.2.1 Device Tree 配置说明

frequency table 是 CPU core 可以正确运行的一组频率/电压组合，一般可以称为 OPP table。OPP table 是频率和电源之间的一个一一对应组合。cpufreq driver 可以在 “调频” 的同时，通过 table 取出和频率对应的电压，进行修改 CPU core 电压，实现 “调压” 的功能。

设备树需要为调频模块配置 opp 描述信息，一个频点可能对应多个可选电压，平台调频驱动会根据硬件平台与性能从多个可选电压中最终选择一个电压作为该频点的对应电压。

根据内核标准，目前全志平台的 dts 文件中 OPP table 有三种配置类型：

（1）类型 1：同一个 opp 节点中不含多个可选电压；

（2）类型 2：同一个 opp 节点中含有多个可选电压，提供给驱动匹配硬件的信息体现在 opp-supported-hw 属性中；

（3）类型 3：同一个 opp 节点中含有多个可选电压，提供给驱动匹配硬件的信息体现在 opp-microvolt-xx 属性中。

这部分信息在驱动开发时已完成配置，不需要额外配置。

需要注意的是，多个 opp 中的频率电压对应关系组合起来即 VF 表，通常由厂家提供，改动 opp 信息可能会对系统稳定性或硬件性能、寿命产生影响，不建议随意修改。

2.2.1.1 OPP table 类型 1

同一个 opp 节点中不含多个可选电压。

```
compatible = "operating-points-v2";：用于匹配驱动的属性。
opp-shared：共享标志，表示多个设备可以使用这个操作点表。
opp-hz：某个频点的频率，单位hz。
opp-microvolt：频率对应的电压，单位uV。
clock-latency-ns：表示CPU从一个频率切换到另一个频率所需的时钟延迟，单位ns。

cpu_opp_l_table: opp_l_table {
    compatible = "operating-points-v2";
    opp-shared;

    opp@720000000 {
      opp-hz = /bits/ 64 <720000000>;
      opp-microvolt = <820000>;
      clock-latency-ns = <244144>;
    };
    opp@936000000 {
      opp-hz = /bits/ 64 <936000000>;
      opp-microvolt = <820000>;
      clock-latency-ns = <244144>;
    };
    opp@1104000000 {
      opp-hz = /bits/ 64 <1104000000>;
      opp-microvolt = <900000>;
      clock-latency-ns = <244144>;
    };
    opp@1200000000 {
      opp-hz = /bits/ 64 <1200000000>;
      opp-microvolt = <950000>;
      clock-latency-ns = <244144>;
    };
    opp@1320000000 {
      opp-hz = /bits/ 64 <1320000000>;
      opp-microvolt = <1000000>;
      clock-latency-ns = <244144>;
    };
    opp@1416000000 {
      opp-hz = /bits/ 64 <1416000000>;
      opp-microvolt = <1050000>;
      clock-latency-ns = <244144>;
    };
```

```
    opp@1512000000 {
      opp-hz = /bits/ 64 <1512000000>;
      opp-microvolt = <1050000>;
      clock-latency-ns = <244144>;
    };
  };
```

2.2.1.2 OPP table 类型 2

同一个 opp 节点中含有多个可选电压，提供给驱动匹配硬件的信息体现在 opp-supported-hw 属性中。

```
compatible = "allwinner,sun50i-operating-points";：用于匹配驱动的属性。
nvmem-cells：表示指定的与非易失性内存(NVMEM)相关的单元。
nvmem-cell-names：指定的NVMEM单元的名称。
opp-hz：某个频点的频率。
opp-microvolt：频率对应的电压。
opp@480000000-0、opp@480000000-1：后缀的0、1仅用于区分不同节点名字，以免报错。
opp-supported-hw：选择该频点所支持的芯片版本。如“opp-supported-hw = <0x3>;”，表示该频点支持bit0、bit1所表示
      的芯片版本。详见内核文档Documentation/devicetree/bindings/opp/opp.txt关于opp-supported-hw的说明。
需要注意，不能存在多个相同频率的频点，所以要避免相同频率的频点都被选择的情况，如opp@480000000-0、
      opp@480000000-1不能被同一个芯片版本选择。

   cpu_opp_l_table: opp_l_table {
    compatible = "allwinner,sun50i-operating-points";
    nvmem-cells = <&speedbin_efuse>;
    nvmem-cell-names = "speed";
    opp-shared;

    opp@480000000-0 {
      opp-hz = /bits/ 64 <480000000>;
      opp-microvolt = <820000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-supported-hw = <0x3>;
    };
    opp@480000000-1 {
      opp-hz = /bits/ 64 <480000000>;
      opp-microvolt = <880000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-supported-hw = <0x4>;
    };
    opp@600000000-0 {
      opp-hz = /bits/ 64 <600000000>;
      opp-microvolt = <820000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-supported-hw = <0x3>;
    };
    opp@600000000-1 {
      opp-hz = /bits/ 64 <600000000>;
      opp-microvolt = <880000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-supported-hw = <0x4>;
    };
    opp@792000000-0 {
      opp-hz = /bits/ 64 <792000000>;
      opp-microvolt = <860000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
```

```
            opp-supported-hw = <0x3>;
        };
        opp@792000000-1 {
            opp-hz = /bits/ 64 <792000000>;
            opp-microvolt = <940000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-supported-hw = <0x4>;
        };
        opp@1008000000-0 {
            opp-hz = /bits/ 64 <1008000000>;
            opp-microvolt = <900000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-supported-hw = <0x3>;
        };
        opp@1008000000-1 {
            opp-hz = /bits/ 64 <1008000000>;
            opp-microvolt = <1020000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-supported-hw = <0x4>;
        };
        opp@1200000000-0 {
            opp-hz = /bits/ 64 <1200000000>;
            opp-microvolt = <960000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-supported-hw = <0x3>;
        };
        opp@1200000000-1 {
            opp-hz = /bits/ 64 <1200000000>;
            opp-microvolt = <1100000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-supported-hw = <0x4>;
        };
        opp@1296000000 {
            opp-hz = /bits/ 64 <1296000000>;
            opp-microvolt = <1100000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-supported-hw = <0x2>;
        };
        opp@1344000000 {
            opp-hz = /bits/ 64 <1344000000>;
            opp-microvolt = <1120000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-supported-hw = <0x4>;
        };
        opp@1512000000 {
            opp-hz = /bits/ 64 <1512000000>;
            opp-microvolt = <1100000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-supported-hw = <0x1>;
        };
    };
```

2.2.1.3 OPP table 类型 3

同一个 opp 节点中含有多个可选电压，提供给驱动匹配硬件的信息体现在 opp-microvolt-xx 属性中。

compatible = "allwinner,sun50i-operating-points";： 用于匹配驱动的属性。
opp-hz： 某个频点的频率。
opp-microvolt-x： 该频率下，x类型bin对应的电压。详见内核文档Documentation/devicetree/bindings/opp/opp.txt关于opp-microvolt-<name>的说明。

```
cpu_opp_l_table: opp_l_table {
 compatible = "allwinner,sun50i-operating-points";
 nvmem-cells = <&speedbin_efuse>, <&cpubin_efuse>;
 nvmem-cell-names = "speed", "bin";
 opp-shared;

 opp@408000000 {
      opp-hz = /bits/ 64 <408000000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-microvolt-a0 = <900000>;
      opp-microvolt-a1 = <900000>;
      opp-microvolt-a2 = <900000>;
      opp-microvolt-b0 = <900000>;
      opp-microvolt-b1 = <900000>;
 };

 opp@600000000 {
      opp-hz = /bits/ 64 <600000000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-microvolt-a0 = <900000>;
      opp-microvolt-a1 = <900000>;
      opp-microvolt-a2 = <900000>;
      opp-microvolt-b0 = <900000>;
      opp-microvolt-b1 = <900000>;
 };

 opp@816000000 {
      opp-hz = /bits/ 64 <816000000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-microvolt-a0 = <940000>;
      opp-microvolt-a1 = <900000>;
      opp-microvolt-a2 = <900000>;
      opp-microvolt-b0 = <900000>;
      opp-microvolt-b1 = <900000>;
 };

 opp@1008000000 {
      opp-hz = /bits/ 64 <1008000000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-microvolt-a0 = <1020000>;
      opp-microvolt-a1 = <980000>;
      opp-microvolt-a2 = <950000>;
      opp-microvolt-b0 = <980000>;
      opp-microvolt-b1 = <950000>;
 };

 opp@1200000000 {
      opp-hz = /bits/ 64 <1200000000>;
      clock-latency-ns = <244144>; /* 8 32k periods */
      opp-microvolt-a0 = <1100000>;
      opp-microvolt-a1 = <1020000>;
      opp-microvolt-a2 = <1000000>;
      opp-microvolt-b0 = <1020000>;
      opp-microvolt-b1 = <1000000>;
 };
```

```
        opp@1320000000 {
            opp-hz = /bits/ 64 <1320000000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-microvolt-a0 = <1160000>;
            opp-microvolt-a1 = <1060000>;
            opp-microvolt-a2 = <1030000>;
            opp-microvolt-b0 = <1060000>;
            opp-microvolt-b1 = <1030000>;
        };

        opp@1416000000 {
            opp-hz = /bits/ 64 <1416000000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-microvolt-a0 = <1180000>;
            opp-microvolt-a1 = <1180000>;
            opp-microvolt-a2 = <1130000>;
            opp-microvolt-b0 = <1100000>;
            opp-microvolt-b1 = <1070000>;
        };

        opp@1512000000 {
            opp-hz = /bits/ 64 <1512000000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-microvolt-b0 = <1180000>;
            opp-microvolt-b1 = <1130000 1130000 1140000>;
        };

        opp@1608000000 {
            opp-hz = /bits/ 64 <1608000000>;
            clock-latency-ns = <244144>; /* 8 32k periods */
            opp-microvolt-b0 = <1180000>;
            opp-microvolt-b1 = <1130000 1130000 1140000>;
        };
    };
```

2.2.1.4 CPU 配置

```
    cpu0: cpu@0 {
      ...
      operating-points-v2 = <&cpu_opp_l_table>;
      ...
    };
```

“operating-points-v2 = <&cpu_opp_l_table>;” ： 该cpu使用的opp table配置项。

2.2.2 kernel menuconfig 配置说明

2.2.2.1 Tina 4.0

```
CPU Power Management
> CPU Frequency scaling
  [*] CPU Frequency scaling                    # cpufreq功能
  [*]   CPU frequency time-in-state statistics      # 导出cpu处于各频点的时间统计信息
      Default CPUFreq governor (performance)  --->  # 使用的默认调频策略
  -*-   'performance' governor                  # 调频策略
  <*>   'powersave' governor
  <*>   'userspace' governor for userspace frequency scaling
  <*>   'conservative' cpufreq governor
  <*>   'interactive' cpufreq policy governor
  <*>   Generic DT based cpufreq driver           # 平台调频驱动
  <*>   Allwinner nvmem based SUN50I CPUFreq driver   # 平台信息匹配驱动，保持默认状态即可
  [ ]   SUNXI PWM CPUFreq support                 # 不使用标准pwm regulator 驱动的平台的调频动作驱动，保持默认即可
```

2.2.2.2 Tina 5.0

```
CPU Power Management
> CPU Frequency scaling
  [*] CPU Frequency scaling                    # cpufreq功能
  [*]   CPU frequency transition statistics        # 导出频率统计信息
  [*]   CPU frequency time-in-state statistics      # 导出cpu处于各频点的时间统计信息
      Default CPUFreq governor (performance)  ---> # 使用的默认调频策略
  -*-   'performance' governor                  # 调频策略
  <*>   'powersave' governor
  <*>   'userspace' governor for userspace frequency scaling
  <*>   'ondemand' cpufreq policy governor
  <*>   'conservative' cpufreq governor
  [*]   'schedutil' cpufreq policy governor
      *** CPU frequency scaling drivers ***
  < >   Generic DT based cpufreq driver
  < >   Dummy CPU frequency driver

Allwinner BSP
> Device Drivers
  > CPU Frequency Drivers
    <*> Generic DT based cpufreq driver        # 平台调频驱动
    < > Generic test for cpufreq driver
      sunxi cpufreq Drivers  --->
        <*> Allwinner nvmem based SUN50I CPUFreq driver # 平台信息匹配驱动，保持默认状态即可
```

2.3 源码结构介绍

```
1. Tina 4.0
linux根目录 drivers/cpufreq/
├──── cpufreq.c              # cpufreq core，提供运作框架，协调各层实现cpufreq功能
├──── cpufreq-dt.c           # cpufreq driver，实现调频调压功能
├──── cpufreq-dt-platdev.c   # cpufreq device，匹配平台driver
├──── cpufreq_governor.c     # cpufreq governor，调频策略统筹
├──── freq_table.c           # frequency table整理
└──── sun50i-cpufreq-nvmem.c  # sunxi平台芯片版本信息匹配

2. Tina 5.0
（1）linux根目录 drivers/cpufreq/ # 存放内核cpufreq框架相关代码
├──── cpufreq.c              # cpufreq core，提供运作框架，协调各层实现cpufreq功能
├──── cpufreq_governor.c     # cpufreq governor，调频策略统筹
└──── freq_table.c           # frequency table整理
（2）bsp/drivers/cpufreq/    # 存放sunxi平台cpufreq驱动代码
├──── cpufreq-linux内核版本
   ├──── cpufreq-dt.c        # cpufreq driver，实现调频调压功能
   └──── cpufreq-dt-platdev.c  # cpufreq device，匹配平台driver
└──── sun50i-cpufreq-nvmem.c  # sunxi平台芯片版本信息匹配
```

3 调试方法

3.1 调试节点

调试节点目录：

```
/sys/devices/system/cpu/cpufreq/policy0
```

表 3-1: cpufreq 调试节点说明前缀是 scaling 的属性文件表示软件可调节的几种属性，前缀是 cpuinfo 的属性文件表示硬件支持的几种属性。

节点	权限	说明
scaling_setspeed	R/W	设置频率，仅当调频策略为 userspace 时可用
scaling_governor	R/W	调频策略设置
scaling_max_freq	R/W	软件调频支持的最大频率
scaling_min_freq	R/W	软件调频支持的最小频率
affected_cpus	R	受到该调频策略影响且在线的 cpu
related_cpus	R	受到该调频策略影响的所有 cpu
scaling_cur_freq	R	linux kernel 缓存的 cpu 当前频率
scaling_driver	R	调频驱动名称
scaling_available_governors	R	可用调频策略
cpuinfo_transition_latency	R	频率转换延迟
cpuinfo_max_freq	R	硬件支持的最大频率
cpuinfo_min_freq	R	硬件支持的最小频率
cpuinfo_cur_freq	R	当前硬件实际运行频率

4 FAQ

4.1 更改调频策略

步骤：

1. 查看可用调频策略：

```
cat /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
```

2. 选择调频策略写入 scaling_governor 节点，例如，选择 performance 策略：

```
echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
```

4.2 用户空间调频

步骤：

1. 查看可用频率（kHz）：

```
cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies
```

2. 切换为 userspace 策略：

```
echo userspace > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
cat /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
```

3. 将可用频率写入 scaling_setspeed：

```
echo 可用频率 > scaling_setspeed
```

4. 确认当前频率：

```
cat /sys/devices/system/cpu/cpufreq/policy0/cpuinfo_cur_freq
```

4.3 获取当前使用的 VF 表

方式一：使用内核原生的 OPP 节点

- 对于 Linux-4.x

```
# 获取所有频点(Hz)
cat /sys/kernel/debug/opp/cpu0/opp*/rate_hz

# 获取所有频点的电压(mV)
 cat /sys/kernel/debug/opp/cpu0/opp*/u_volt_target

PS：若没有/sys/kernel/debug目录，可输入以下命令挂载该目录：
mount -t debugfs none /sys/kernel/debug/
```

- 对于 Linux-5.x

```
# 获取所有频点(Hz)
cat /sys/kernel/debug/opp/cpu0/opp*/rate_hz

# 获取所有频点的电压(mV)
 cat /sys/kernel/debug/opp/cpu0/opp*/supply-0/u_volt_target
```

方式二：使用 sunxi 自定义节点 cpufreq_table

```
 cat /sys/kernel/debug/cpufreq_table

PS：该节点需要kernel编译 drivers/soc/sunxi/vf-table.c
```

4.4 更改 VF 表

方式一：直接修改 dts 文件中 VF 表的电压频率配置值，重新编译并烧录固件；

方式二：在 uboot 阶段临时修改 VF 表，该方法无需重烧固件，但对于安全固件，修改不会保存，重启后失效，修改方法如下：

1. 进入 uboot 命令行:

```
可通过开机时持续输入字符进入，例如，串口连接电脑，按住键盘"s"按键，重启系统。
```

2. 修改 VF 表信息，示例如下:

```
 fdt list命令查看节点信息
=> fdt list /opp_l_table
opp_l_table {
      compatible = "allwinner,sun50i-operating-points";
      nvmem-cells = <0x000000ff 0x00000100>;
      nvmem-cell-names = "speed", "bin";
      opp-shared;
      linux,phandle = <0x000000fc>;
      phandle = <0x000000fc>;
      opp@408000000 {
      };
      opp@600000000 {
      };
      opp@816000000 {
      };
      opp@1008000000 {
      };
      opp@1200000000 {
      };
      opp@1320000000 {
      };
      opp@1416000000 {
      };
      opp@1464000000 {
      };
      opp@1512000000 {
      };
      opp@1608000000 {
      };
};

# 使用fdt rm 或fdt set等命令修改vf表，例如删除408MHz频点
 fdt rm /opp_l_table/opp@408000000

# 保存修改
fdt save
PS: 安全固件不支持保存

# 启动系统，内核启动后查看节点确认修改是否生效
 boot
 cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies
```

4.5 一号通 FAQ 列表

更多 FAQ 请登录 allwinner 一号通平台，在常见问题中搜索 “调频”、“cpufreq” 关键词获取。

典型 FAQ：

【FAQ2410】linux4.9 内核中使用 tbl 表的时钟调频失败问题

【FAQ1616】介绍通过温度调节 CPU 频率规则

著作权声明

商标声明

免责声明