

RK356X Linux PCIe 开发指南

文件标识：RK-KF-YF-141

发布版本：V1.2.0

日期：2021-01-15

文件密级：☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2021 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

前言

概述

产品版本

芯片名称	内核版本
RK356X	4.19

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

修订记录

日期	版本	作者	修改说明
2021-01-15	V1.0.0	林涛	初始版本
2021-01-22	V1.1.0	林涛	增加PCIe 3.0控制器异常情况的检查信息
2021-01-26	V1.2.0	林涛	增加PCIe 2.0 Combo phy异常排除信息

目录

RK356X Linux PCIe 开发指南

- 芯片资源介绍
- DTS 配置
- menuconfig 配置
- 常见应用问题
- 芯片互联功能
- 异常排查

芯片资源介绍

RK3566

资源	模式	支持芯片互联	支持lane拆分	备注
PCIe Gen2 x 1 lane	RC only	否	否	内部时钟

RK3568

资源	模式	支持芯片互联	支持lane拆分	备注
PCIe Gen2 x 1 lane	RC only	否	否	内部时钟
PCIe Gen3 x 2 lane	RC/EP	是	1 lane RC+ 1 lane RC	外置晶振时钟

DTS 配置

RK3566

资源	模式	参考配置	控制器节点	PHY节点
PCIe Gen2 x 1 lane	RC	rk3566-evb1-ddr4-v10.dtsi	pcie2x1	combphy2_psq

RK3568

资源	模式	参考配置	控制器节点	PHY节点
PCIe Gen2 x 1 lane	RC	rk3568-evb2-lp4x-v10.dtsi	pcie2x1	combphy2_psq
PCIe Gen3 x 2 lane	RC	rk3568-evb1-ddr4-v10.dtsi	pcie3x2	pcie30phy
PCIe Gen3 拆分1 lane + 1 lane	RC	rk3568-evb6-ddr3-v10.dtsi	pcie3x2 pcie3x1	pcie30phy
PCIe Gen3 x 2 lane	EP	rk3568-iotest-ddr3-v10.dts	pcie3x2	pcie30phy

1. compatible = "rockchip,rk3568-pcie", "snps,dw-pcie";

可选配置项：此项目设置PCIe接口使用的是RC模式还是EP模式。作为RC功能时，需要配置成 compatible = "rockchip,rk3568-pcie", "snps,dw-pcie"; 而如果需要修改成EP模式，则需要修改为 compatible = "rockchip,rk3568-pcie-ep", "snps,dw-pcie";

2. reset-gpios = <&gpio3 13 GPIO_ACTIVE_HIGH>;`

必须配置项：此项是设置 PCIe 接口的 PERST#复位信号；不论是插槽还是焊贴的设备，请在原理图上找到该引脚，并正确配置。否则很有可能将无法稳定完成链路建立。

3. num-lanes = <4>;

无需配置项：此配置设置 PCIe 设备所使用的 lane 数量，已在rk3568.dtsi中配置，默认不需要调整，软件可以自己探测并关闭不需要的 lane 以节省功耗。

4. max-link-speed = <2>;

无需配置项：此配置设置 PCIe 的带宽版本，1 表示 Gen1，2 表示 Gen2，3表示Gen3。需要注意，此配置与芯片相关，原则上不需要每个板子配置，因此我们在SoC的rk3568.dtsi中已配置，仅仅是做为一个测试手段，或者客户板子设计异常后的降级手段。

5. status = <okay>;

必须配置项：此配置需要在 pcie控制器节点和对应的 phy 节点同时使能。

6. vpcie3v3-supply = <&vdd_pcie3v3>;

可选配置项：用于配置 PCIe 外设的 3V3 供电(原则上我司的硬件参考原理图上将PCIe插槽的12V电源和 3V3电源合并控制，所以配置3v3的电源之后，12V电源一并控制)。如果板级针对 PCIe 外设的 3V3 需要控制使能，则如范例所示定义一组对应的 regulator，regulator 的配置请参考 Documentation/devicetree/bindings/regulator/。另需要注意，如果是PCIe3.0的控制器，一般需要外接100M晶振芯片，那么该晶振芯片的供电原则上硬件设计与PCIe外设的3V3共用。所以配置了该项之后，除了确认外设3V3供电之外，还需要确认外置晶振芯片的时钟是否输出正常。

7. rockchip,bifurcation;

可选配置项：可以将pcie3x2的2个lane 拆成两个1个lane的控制器来使用。具体的配置方法就是dts中 pcie3x1和pcie3x2控制器节点和pcie30phy都使能，并且pcie3x2和pcie3x1节点中都添加 rockchip,bifurcation属性。可参考rk3568-evb6-ddr3-v10.dtsi。否则默认情况下，pcie3x1控制器无法使用。

此时lane0是由pcie3x2控制器使用，lane1是由pcie3x1控制器使用，硬件布板上严格按照我司原理图。另注意，此模式下两个1-lane的控制器必须同时工作在RC模式下。

8. rockchip,ext-refclk

特殊调试配置：首先请注意此配置仅仅针对PCIe2x1控制器所对应combphy2_psq。默认combphy2_psq使用SoC内部时钟方案，可参阅rk3568.dtsi节点，默认使用24MHz时钟源。除了24MHz时钟源，还支持25M和100M，仅需要调整assigned-clock-rates = <24000000>数值为所需频率即可。内部时钟源方案成本最优，所以作为SDK默认方案，但combphy2_psq仍然预留了外部晶振芯片的时钟源输入选择。如果PCIe2x1确实需要使用外部时钟晶振芯片提供时钟的方案，请在板级的dts的combphy2_psq中加入rockchip,ext-refclk，且需要注意在节点中加入assigned-clock-rates = <时钟频率> 来指定外部时钟芯片输入的频率，仍然只支持24M,25M,100M三档。

9. rockchip,lpbk-master

特殊调试配置：此配置是针对loopback信号测试，使用PCIe控制器构造模拟loopback master环境，让待测试对端设备进入slave模型，非模拟验证实验室的RX环路需求请勿配置。另注意，Gen3控制器可能需要配置compliance模式，才可以loopback slave模式。如果阅读者不理解什么是loopback测试，说明这不是你要找的配置，请勿针对此配置提问。

10. rockchip,compliance-mode

特殊调试配置：此配置是针对compliance信号测试，使用PCIe控制器强制进入compliance测试模式。默认TX测试应该使用测试SMA夹具进入compliance，而不需要强制进入。预留此配置是为了测试Gen3模式的loopback slave，因为实验室测试可能Gen3的loopback测试需要进compliance模式。如果阅读者不理解什么是compliance测试，说明这不是你要找的配置，请勿针对此配置提问。

menuconfig 配置

1. 需要确保如下配置打开，方可正确的使用 PCIe 相关功能

```
CONFIG_PCI=y
CONFIG_PCI_DOMAINS=y
CONFIG_PCI_DOMAINS_GENERIC=y
CONFIG_PCI_SYSCALL=y
CONFIG_PCI_BUS_ADDR_T_64BIT=y
CONFIG_PCI_MSI=y
CONFIG_PCI_MSI_IRQ_DOMAIN=y
CONFIG_PHY_ROCKCHIP_SNPS_PCIE3=y
CONFIG_PHY_ROCKCHIP_NANENG_COMBO_PHY=y
CONFIG_PCIE_DW=y
CONFIG_PCIE_DW_HOST=y
CONFIG_PCIE_DW_ROCKCHIP=y
CONFIG_PCIEPORTBUS=y
CONFIG_PCIE_PME=y
CONFIG_GENERIC_MSI_IRQ=y
CONFIG_GENERIC_MSI_IRQ_DOMAIN=y
CONFIG_IRQ_DOMAIN=y
CONFIG_IRQ_DOMAIN_HIERARCHY=y
```

2. 使能 NVMe 设备(建立在 PCIe 接口的 SSD)，PCIe转接AHCI设备（SATA），PCIe转接USB设备（XHCI）均已在默认config中打开，烦请确认。其他转接设备例如以太网卡，WiFi等请自行确认相关config配置。

```
CONFIG_BLK_DEV_NVME=y
CONFIG_SATA_PMP=y
CONFIG_SATA_AHCI=y
CONFIG_SATA_AHCI_PLATFORM=y
CONFIG_ATA_SFF=y
CONFIG_ATA=y
CONFIG_USB_XHCI_PCI=y
CONFIG_USB_XHCI_HCD=y
```

特别说明，默认 4.19 开源内核仅支持 drivers/ata/ahci.c 中列表内的PCIe转接SATA设备，超出部分请找原厂或者代理商支持。

常见应用问题

Q1： 客户走线的时候不好走，问不同 lane 之间能否交织？

A1： 理论上可以交织，RC 的 lane[1-4]与 EP/switch 的 lane[1-4]随意对应，属于硬件协议行为，软件不需要改动。但我司EVB未验证，请谨慎使用，把控风险。

Q2： 同一个 lane 的差分信号能否交织？比如 RC 的 lane1 的 RX+ 与 EP/Switch 的 RX-对应，TX+与 EP/Switch 的 TX-对应。或者 RX 正负对应，TX 正负对应等等情况，怎么处理？

A2： 理论上可以任意接，软件上不需要再额外处理。PCIe 的探测状态机已经考虑了这些所有情况。但我司EVB未验证，请谨慎使用，把控风险。

Q3： RK356X的只有3.0的 RC有2个lane，能不能支持把这2个 lane 拆分成1+1模式？

A3： 可以，详细配置请看下DTS配置的第六点

Q4： RK356X 芯片支持分配的BAR空间地址域有多大？

A4： PCIe2.0控制器支持1GB的64-bit memory空间(不支持预取)和1MB的IO空间。PCIe3.0控制器如果是两个lane同时使用，则PCIe3x2支持1GB的64-bit memory空间(不支持预取)和1MB的IO空间。PCIe3.0控制器如果拆分成两个1-lane的控制器，则PCIe3x1和PCIe3x2分别都支持1GB的64-bit memory空间(不支持预取)和1MB的IO空间。

Q5： 是否支持PCIe switch？贵司有没有推荐？

A5： 理论上支持，不需要任何补丁，且没有推荐列表。为了把控风险，请联系供应商借评估板，插在我司EVB上验证后再采购。

Q6： 在系统中如何确定控制器与设备的对应关系？

A6： PCIe2x1控制器给外设分配的Bus地址介于0x0~0xf，PCIe3x1控制器给外设分配的bus地址介于0x10~0x1f，PCIe3x2控制器给外设分配的bus地址介于0x20~0x2f。从lspci输出的信息中可以看到各设备分配到的bus地址（高位），即可确定对应关系。第二列Class是设备类型，第三列VID:PID。Class类型请参考<https://pci-ids.ucw.cz/read/PD/>，厂商VID和产品PID请参考 <http://pci-ids.ucw.cz/v2.2/pci.ids>

```
console:/ # lspci
21:00.0 class 0108: 144d:a808
20:00.0 class 0604: 1d87:3566
11:00.0 class 0c03: 1912:0014
10:00.0 class 0604: 1d87:3566
01:00.0 class 0c03: 1912:0014
00:00.0 class 0604: 1d87:3566
```

我们可以看到每个控制器下游预留了16级bus来接设备，意味着每个控制器下游可以接16个设备(含 switch)，一般可以满足需求，阅读者可以跳过下面的说明。如果确属需要调整，请调整rk3568.dtsi中三个控制器的bus-range分配，且务必确保不要重叠。另外，调整bus-range将导致设备的MSI(-X) RID区间变化，请同步调整msi-map。

```
bus-range = <起始地址    结束地址>

msi-map = < bus-range中的起始地址 << 16
           &its
           bus-range中的起始地址 << 16
           bus-range中分配的总线总数 << 16>
```

例如bus-range调整为0x30 ~ 0x60，即该控制器下游设备分配的bus地址从0x30 到0x60，总线总数 0x30 个

则可配置 msi-map = <0x3000 &its 0x3000 0x3000>

依此类推，且一定要保证三个控制器的bus-range和msi-map互不重叠，且bus-range和msi-map相互适配。

Q7：如何确定PCIe设备的链路状态？

A7: 请使用服务器发布的lspci工具，执行lspci -vvv，找到对应设备的linkStat即可查看；其中Speed为速度，

Width即为lane数。如需要解析其他信息，请查找搜索引擎，对照查看。

芯片互联功能

RK3568芯片的PCIe Gen3 x 2 lane的接口支持EP或者功能，用于芯片间互联。RK3566芯片和RK3568芯片的PCIe Gen2 x 1 lane接口不可用于芯片间互联。

1. 请确保内核配置项打开下列项，其中作为EP板子的rk3568.dtsi中配置所需使用的控制器的compatible字段为compatible = "rockchip,rk3568-pcie-ep"；作为RC的板子的所使用控制器的配置不变。

```
CONFIG_ROCKCHIP_PCIE_DMA_OBJ=y
CONFIG_DEBUG_FS=y
```

2. 然后在**两个板子**的rk3568.dtsi中都预留一段内存做为通信数据空间，并加到所用控制器的节点中，例如

```
作为EP板子的rk3568芯片配置如下，我们以pcie3x2做为接口为例
reserved-memory {
    #address-cells = <2>;
    #size-cells = <2>;
    ranges;
    dma_trans: dma_trans@3c000000 {
        reg = <0x0 0x3c000000 0x0 0x04000000>; //保留了0x3c000000 到0x40000000的内
存
    };
};

&pcie3x2 {
    compatible = "rockchip,rk3568-pcie-ep"; //pcie3x2做为EP
    memory-region = <&dma_trans>; //这段内存给pcie3x2控制器用，做为互联时候通信的内
存
```

```
busno = <1>; //作为EP需分配bus 1
};
```

作为RC板子的rk3568芯片配置如下，我们以pcie3x2做为接口为例

```
reserved-memory {
    #address-cells = <2>;
    #size-cells = <2>;
    ranges;
    dma_trans: dma_trans@3c000000 {
        reg = <0x0 0x3c000000 0x0 0x04000000>; //保留了0x3c000000 到0x40000000的内
存
    };
};

&pcie3x2 {
    compatible = "rockchip,rk3568-pcie"; //pcie3x2做为RC
    memory-region = <&dma_trans>; //这段内存给pcie3x2控制器用，做为互联时候通信的内
存
    busno = <0>; //作为RC分配bus 0
};
```

3. 内部开发工程师如需运行互联模式的程序以及参考代码，可以直接访问<https://redmine.rock-chip.com/issues/281070>。客户需取得redmine中对应项目的权限后，联系FAE中心获取。其中 test-pcie-ep-new是一个daemon程序，用于互联传输协议的维护。test-pcie 是实时数据发送程序，用于数据的实际传输。

4. 将 test-pcie-ep-new 和 test-pcie 拷贝到RC 和EP板子中

首先RC和EP的板子都运行以下命令用于应答 ./test-pcie-ep-new 500 &

其次RC发送命令,发送10000包数据，每包1M ./test-pcie 1 10000

EP发送命令,发送10000包数据，每包1M ./test-pcie 2 10000

最后如果正常结束，在RC和EP端都能看到类似以下log：DMA: To bus: 1541MB/s

5. 互联模型的异常debug问题请提供下列两个信息：

```
cat /sys/kernel/debug/pcie/pcie_trx
cat /proc/interrupts | grep pcie
```

异常排查

1. training 失败

PCIe Link Fail的log如下一致重复，LTSSM状态机可能不同

```
rk-pcie 3c0000000.pcie: PCIe Linking... LTSSM is 0x0
rk-pcie 3c0000000.pcie: PCIe Linking... LTSSM is 0x0
rk-pcie 3c0000000.pcie: PCIe Linking... LTSSM is 0x0
```

如果link成功，应该可以看到类似log，LTSSM状态机可能不同，重点看到link up了

```
[ 2.410536] rk-pcie 3c0000000.pcie: PCIe Link up, LTSSM is 0x130011
```

异常原因：training 失败，外设没有处于工作状态或者信号异常。首先检测下 reset-gpios 这个是否配置对了。其次，检测下外设的3V3供电是否有，是否足够，部分外设需要12V电源。最后测试复位信号与电源的时序是否与此设备的spec冲突。如果都无法解决，大概率需要定位信号完整性，需要拿出测试眼图和PCB给到我司硬件，并且最好我们建议贵司找实验室提供一份测试TX兼容性信号测试报告。

另外还建议客户打开pcie-dw-rockchip.c中的RK_PCIE_DBG，抓一份log以便分析。请阅读者注意，如果有多个控制器同时使用，抓log前请先把不使用或者没问题的设备对应的控制器disable掉，这样log会好分析一点。

2. PCIe3.0控制器初始化设备系统异常

```
[ 21.523506] rcu: INFO: rcu_preempt detected stalls on CPUs/tasks:
[ 21.523557] rcu:      1-...0: (0 ticks this GP) idle=652/1/0x4000000000000000
softirq=30/30 fqs=2097
[ 21.523579] rcu:      3-...0: (5 ticks this GP) idle=4fa/1/0x4000000000000000
softirq=35/36 fqs=2097
[ 21.523590] rcu:      (detected by 2, t=6302 jiffies, g=-1151, q=98)
[ 21.523610] Task dump for CPU 1:
[ 21.523622] rk-pcie          R  running task          0    55        2 0x0000002a
[ 21.523640] Call trace:
[ 21.523666] __switch_to+0xe0/0x128
[ 21.523682] 0x43752cfcfe820900
[ 21.523694] Task dump for CPU 3:
[ 21.523704] kworker/u8:0      R  running task          0     7        2 0x0000002a
[ 21.523737] workqueue: events_unbound enable_ptr_key_workfn
[ 21.523751] Call trace:
[ 21.523767] __switch_to+0xe0/0x128
[ 21.523786] event_xdp_redirect+0x8/0x90
[ 21.523816] rcu: INFO: rcu_sched detected stalls on CPUs/tasks:
[ 21.523840] rcu:      1-...0: (50 ticks this GP) idle=652/1/0x4000000000000000
softirq=7/30 fqs=2099
[ 21.523859] rcu:      3-...0: (55 ticks this GP) idle=4fa/1/0x4000000000000000
softirq=5/36 fqs=2099
[ 21.523870] rcu:      (detected by 2, t=6302 jiffies, g=-1183, q=1)
[ 21.523887] Task dump for CPU 1:
[ 21.523898] rk-pcie          R  running task          0    55        2 0x0000002a
[ 21.523915] Call trace:
[ 21.523931] __switch_to+0xe0/0x128
[ 21.523944] 0x43752cfcfe820900
[ 21.523955] Task dump for CPU 3:
[ 21.523965] kworker/u8:0      R  running task          0     7        2 0x0000002a
[ 21.523990] workqueue: events_unbound enable_ptr_key_workfn
[ 21.524004] Call trace:
```

异常原因：如果系统卡住此log附近，则表明PCIe3.0的PHY工作异常。请依次检查

- 外部晶振芯片的时钟输入是否异常，如果无时钟或者幅度异常，将导致phy无法锁定。
- 检查 PCIE30_AVDD_0V9 和PCIE30_AVDD_1V8电压是否满足要求。

3. PCIe2.0控制器初始化设备系统异常

```
[ 21.523870] rcu:      (detected by 2, t=6302 jiffies, g=-1183, q=1)
[ 21.523887] Task dump for CPU 1:
[ 21.523898] rk-pcie          R  running task          0    55        2 0x0000002a
[ 21.523915] Call trace:
[ 21.523931] __switch_to+0xe0/0x128
[ 21.523944] 0x43752cfcfe820900
[ 21.523955] Task dump for CPU 3:
[ 21.523965] kworker/u8:0      R  running task          0     7        2 0x0000002a
[ 21.523990] workqueue: events_unbound enable_ptr_key_workfn
[ 21.524004] Call trace:
```


异常原因：如果系统卡住此log附近，则表明PCIe2.0的PHY工作异常。请依次检查

- 检查 PCIE30_AVDD_0V9 和PCIE30_AVDD_1V8电压是否满足要求。
- 修改combphy2_psq的驱动phy-rockchip-naneng-combphy.c，在rockchip_combphy_init函数的末尾增加如下代码，检查PHY内部的一些配置：

```
val = readl(priv->mmio + (0x27 << 2));
dev_err(priv->dev, "TXPLL_LOCK is 0x%x PWON_PLL is 0x%x\n",
val & BIT(0), val & BIT(1));
val = readl(priv->mmio + (0x28 << 2));
dev_err(priv->dev, "PWON_IREF is 0x%x\n", val & BIT(7));
```

首先查看TXPLL_LOCK是否为1，如果不是，表明PHY没有lock完成。其次查看PWON_IREF是否为1，如果不为1，则表明PHY时钟异常。此时尝试切换combophy的时钟频率，修改rk3568.dtsi中的combphy2_psq的assigned-clock-rates，依次调整为25M或者100M进行尝试。

- 如果调整以上步骤均无效，请将PHY内部的时钟bypass到refclk差分信号脚上，进行测量。bypass加在rockchip_combphy_pcie_init函数的末尾，设置如下代码所示

```
u32 val;
val = readl(priv->mmio + (0xd << 2));
val |= BIT(5);
writel(val,priv->mmio + (0xd << 2));
```

设置完成后，请依次配置combphy2_psq的时钟频率为24M,25M以及100M，用示波器从PCIe的refclk差分信号脚上测量时钟情况，检查频率和幅值、抖动是否满足要求。