RK356X Linux PCIe 开发指南

文件标识：RK-KF-YF-141

发布版本：V1.9.0

日期：2021-03-16

文件密级：□绝密 □秘密 □内部资料 ■公开

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

软件开发工程师

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
2021-02-04	V1.3.0	林涛	增加MSI和MSI-X支持数量的问题描述
2021-02-05	V1.4.0	林涛	增加地址分配异常信息
2021-02-06	V1.5.0	林涛	增加PCIe2x1的PHY支持SSC说明
2021-02-23	V1.6.0	林涛	增加MSI/MSI-X调试支持和运行态设备异常说明
2021-02-26	V1.7.0	林涛	增加Legacy INT的说明
2021-02-27	V1.8.0	林涛	增加标注EP功能件开发说明
2021-03-16	V1.9.0	林涛	增加FW存在异常设备的说明

目录

1. 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

2. 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,enable-ssc`

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。默认情况下，PCIe2x1的PHY输出时钟不开启展频。如果用户需要规避一些EMI问题，可尝试combphy2_psq节点加入此配置项，开启SSC。

10. `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。

11. `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。

3. 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。

4. 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。

Q8：如何确定SoC针对PCIe设备可分配的MSI或者MSI-X数量？

A8: SoC针对每个PCIe设备可分配的数量由中断控制器的资源决定。3566和3568上，针对PCIe2.0和PCIe3.0控制器的下游设备，可分配的MSI或者MSI-X总数均是65535个。

Q9：是否支持Legacy INT方式？如何强制使用Legacy INTA ~ INTD的中断？

A9: 支持legacy INT方式。但Linux PCIe协议栈默认的优先级是MSI-X, MSI, Legacy INT，因此常规市售设备不会去申请Legacy INT。若调试测试需要，请参考内核中Documentation/admin-guide/kernel-parameters.txt文档，其中"pci=option[,option...] [PCI] various PCI subsystem options."描述了可以在cmdline中关闭MSI，则系统默认会强制使用Legacy INT分配机制。以RK356X安卓平台为例，可在arch/arm64/boot/dts/rockchip/rk3568-android.dtsi的cmdline参数中额外添加一项pci=nomsi，注意前后项需空格隔开：

```
bootargs = "...... pci=nomsi ......";
```

如果添加成功，则lspci -vvv可以看到此设备的MSI和MSI-X都是处于关闭状态(Enable-)，而分配了INT A中断，中断号是80。cat /proc/interrupts可查看到80中断的状态。

```
01:00.0 Class 0108: Device 14a4:22f1 (rev 01) (prog-if 02)
        Subsystem: Device 1b4b:1093
...
        Interrupt: pin A routed to IRQ 80
...
        Capabilities: [50] MSI: Enable- Count=1/1 Maskable+ 64bit+
                Address: 0000000000000000  Data: 0000
                Masking: 00000000  Pending: 00000000
...
        Capabilities: [b0] MSI-X: Enable- Count=19 Masked-
                Vector table: BAR=0 offset=00002000
                PBA: BAR=0 offset=00003000
```

5. 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-chips.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deamon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/interrups | grep pcie
```

6. 标准EP功能件开发

将RK3568 PCIe接口作为EP设备与任意RK芯片互联，推荐使用我司提供的互联模型，性能与稳定性等均可得到有效的保证，详情可查看“芯片互联功能”章节。若熟悉PCIe EP设备驱动开发的技术人员需要使用RK3568对接封闭芯片系统(如x86)，或者希望自行开发标准EP业务流程的，可参考本节内容进行二次开发。

标准EP功能件开发需要三个功能组件：

- RC端运行的针对RK3568 EP设备的function driver, 其功能是负责在RC系统中申请bar空间对应的虚拟内存，管理数据业务，注册处理各类中断，提供更上层业务态的业务接口。
- EP端(本例指RK3568芯片)运行的firmware driver, 其功能是负责配置bar内存的inbound和outbound，提供DMA完成EP端与RC端内存数据搬移等功能。
- 快速建立链路的loader: 负责配置class code, ID, 修改bar需求大小以及迅速建立链路连接；因为例如x86的BIOS扫描总线时间较短，需要提前准备链路。

由于开发技术难度较高，我们提供了可运行的全部demo，希望降低阅读者二次开发的难度。此demo可以将RK3568芯片模拟成一个memory controller, 可对接任何芯片平台的linux系统。以x86为例，执行sudo lspci可以看到我司设备：

```
lt-HP-ProDesk-400-G5 -NT-ID5-APD:~$ sudo lspci
[sudo] password for lt:
00:00.0 Hest bridge: Intel Corporaton 8th Gen Core Processor Host Brldge/RAN
Regtsters
00:02.0 VGA compatible controller: Intel Corporation UHD Graphics 630 (Desktop)

...

02:00.0 Memory controller: Fuzhou Rockchip Electronics co. Ltd Device 356a Crev
01)
```

在RC端加载function driver模块之后，将出现/dev/rk-rmd设备节点，可使用echo/cat访问此节点。访问该节点实际将访问到EP端(本例指RK3568)的内存，而EP端被访问的内存地址在EP端的firmware driver中可配置。利用本demo可以实现最原始的数据交互，为封装更上层业务态提供支持。内部开发工程师如需运行标准EP功能件的程序以及参考代码，可以直接访问https://redmine.rock-chips.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。

注意事项：

- 对接x86设备时需要注意，由于较多市售x86设备主板的x16的插槽默认不支持低于4-lane的设备，烦请设计成x1的金手指接入其x1的槽。
- EP端的系统供电通过金手指由RC的PCIe插槽上提供，并将金手指的#PERST接到EP设备主控的PMU复位信号上，使得RC端可以控制它插槽上的#PERST信号，对EP进行芯片级的复位控制。
- EP端金手指的#PRSENT信号需要正确布置成x1模式。

7. 异常排查

1. **trainning** 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：trainning 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。

4. **PCIe外设资源分配异常**

```
3.286864] pci 0002:20:00.0: bridge configuration invalid ([bus 01-ff]),
reconfiguring
3.286886] scanning [bus 00-00] behind bridge, pass 1
3.288165] pci 0002:21 :00.0: supports D1 D2
3.288170] pci 0002:21 :00.0: PME# supported from DO D1 D3hot
3.298238] pci bus 0002:21: busn res: [bus 21-2f] end is updated to 21
3.298441] pci 0002:21:00.0: BAR 1: no space for [mem size 0xe0000000 ]
3.298456] pci 0002:21:00.0: BAR 1: failed to assign [mem size 0xe0000000 ]
3.298473] pci 0002:21:00.0: BAR 2: assigned [mem 0x380900000- 0x38090ffff pref ]
3.298488] pci 0002:21:00.0: PCI bridge to [bus 21]
```

如常用应用问题Q4所述，RK356X的PCIe地址空间有限制。此log表明21号总线外设向RK356X申请3GB的64bit memory空间，超出了限制导致无法分配资源。若为市售设备，将不受RK356X芯片支持；若为定制设备，请联系设备vendor确认是否可以修改其BAR空间容量编码。

5. **MSI/MSI-X无法使用**

在移植外设驱动的开发过程中(主要指的是WiFi)，认为主机端的function driver因无法使用MSI或者MSI-X中断而导致流程不正常，按如下流程进行排查

- 确认前述menuconfig 中提到的配置，尤其是MSI相关配置是否都有正确勾选
- 确认rk3568.dtsi中，its节点是否被设置为disabled
- 执行lspci -vvv，查看对应设备的MSI或者MSI-X是否有支持并被使能。以此设备为例，其上报的capabilities显示其支持32个64 bit MSI，目前仅使用1个，但是 Enable-表示未使能。若正确使能应该看到Enable+， 且Address应该能看到类似为0x00000000fd4400XX的地址。此情况一般是设备驱动还未加载或者加载时申请MSI或者MSI-X失败导致，请参考其他驱动，使用pci_alloc_irq_vectors等函数进行申请，详情可结合其他成熟的PCIe外设驱动做法以及参考内核中的Documentation/PCI/MSI-HOWTO.txt文档进行编写和排查异常。

```
Capabilities: [58] MSI: Enable- Count=1/32 Maskable- 64bit+
                Address: 0000000000000000  Data: 0000
```

- 如果MSI或者MSI-X有正确申请，可用如下命令导出中断计数，查看是否正常： cat /proc/interrupts 。在其中找到对应驱动申请的ITS-MSI中断(依据最后一列申请者驱动名称，例如此处为xhci_hcd驱动申请了这些MSI中断)。理论上每一笔的通信传输都会增加ITS，如果设备没有通信或者通信不正常，就会看到中断计数为0，或者有数值但发起通信后不再增加中断计数的情况。

```
229: 0 0 0 0 0 0 ITS-MSI 524288 Edge xhci_hcd
```

- 如果是概率性事件导致function driver无法收到MSI或者MSI-X中断，可以进行如下尝试。首先执行cat /proc/interrupts 查看相应中断号，以上述229为例，将中断迁移导其他CPU测试。例如切换导CPU2，则使用命令echo 2 > /proc/irq/229/smp_affinity_list 。
- 使用协议分析仪器抓取协议信号，查看流程中外设是否有概率性没有向主机发送MSI或者MSI-X中断，而导致的异常。需注意，目前协议分析仪一般都难以支持焊贴设备的信号采集，需向设备vendor购买金手指的板卡，在我司EVB上进行测试和信号采集。另需注意我司EVB仅支持标准接口的金手指板卡，若待测设备为M.2接口的设备(常见key A, key B, key M三种类型)，请采购使用对应型号的转接板。

6. **外设枚举后通信过程中报错**

以下是NVMe在RK3566-EVB2上进行正常枚举之后，通信过程中突然设备异常报错的log。不论是什么设备，如果可以正常枚举并使能，则可以看到类似nvme 0000:01:00.0: enabling device (0000 -> 0002)的log。此后通信过程中设备报错，需要考虑如下三个方面：

- 利用示波器测量触发外设的电源，排除是否有跌落的情况发生
- 利用示波器测量触发外设的#PERST信号，排除是否被人误操作导致设备被复位的情况发生
- 利用示波器测量触发PCIe PHY的0v9和1v8两路电源，排除是否PHY的电源异常

特别提醒：RK EVB有较多的信号复用，利用拨码将PCIe的#PERST控制信号和其他外设的IO进行复用，请配合硬件重点确认。例如目前已知有部分RK3566-EVB2的拨码有异常，需要修正。

```
[    2.426038] pci 0000:00:00.0:   bridge window [mem 0x300900000-0x3009fffff]
[    2.426183] pcieport 0000:00:00.0: of_irq_parse_pci: failed with rc=-22
[    2.427493] pcieport 0000:00:00.0: Signaling PME with IRQ 106
[    2.427712] pcieport 0000:00:00.0: AER enabled with IRQ 115
[    2.427899] pcieport 0000:00:00.0: of_irq_parse_pci: failed with rc=-22
```

```
[    2.428202] nvme nvme0: pci function 0000:01:00.0
[    2.428259] nvme 0000:01:00.0: enabling device (0000 -> 0002)
[    2.535404] nvme nvme0: missing or invalid SUBNQN field.
[    2.535522] nvme nvme0: Shutdown timeout set to 8 seconds
...
[   48.129408] print_req_error: I/O error, dev nvme0n1, sector 0
[   48.137197] nvme 0000:01:00.0: enabling device (0000 -> 0002)
[   48.137299] nvme nvme0: Removing after probe failure status: -19
[   48.147182] Buffer I/O error on dev nvme0n1, logical block 0, async page read
[   48.162900] nvme nvme0: failed to set APST feature (-19)
```

7. 外设枚举过程报**FW**异常

如设备在枚举过程分配BAR空间报如下错误，一般问题是设备的BAR空间与协议不兼容，需要特殊处理。需要修改drivers/pci/quirks.c中，增加对应quirk处理。具体信息应该咨询设备厂商。

```
[    2.379768] rk-pcie 3c0000000.pcie: PCIe Link up, LTSSM is 0x30011
[    2.380155] rk-pcie 3c0000000.pcie: PCI host bridge to bus 0000:00
[    2.380187] pci_bus 0000:00: root bus resource [bus 00-0f]
[    2.380204] pci_bus 0000:00: root bus resource [??? 0x300000000-0x3007fffff
flags 0x0] (bus address [0x00000000-0x007fffff])
[    2.380217] pci_bus 0000:00: root bus resource [io  0x0000-0xfffff] (bus
address [0x800000-0x8fffff])
[    2.380230] pci_bus 0000:00: root bus resource [mem 0x300900000-0x33fffffff]
(bus address [0x00900000-0x3fffffff])
[    2.394983] pci 0000:01:00.0: [Firmware Bug] reg 0x10: invalid BAR (can't
size)
```