



**Hi309a**

## **FAQ 使用指南**

文档版本 01

发布日期 2025-10-30

版权所有 © 海思技术有限公司2026。保留一切权利。

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。

## 商标声明



**HISILICON**、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受海思公司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的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 海思技术有限公司

地址：上海市青浦区虹桥港路2号101室 邮编：201721

网址：<https://www.hisilicon.com/cn/>

客户服务邮箱：[support@hisilicon.com](mailto:support@hisilicon.com)



# 前言

## 概述

本文档是为使用海思Hi309a芯片进行二次开发的工程师编写，提供了开发过程中可能遇到的Hi309a使用常见问题解答（FAQ）。

## 读者对象

本文档主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

## 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

| 符号                                                                                            | 说明                                                                    |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
|  <b>危险</b> | 表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。                                        |
|  <b>警告</b> | 表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。                                        |
|  <b>注意</b> | 表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。                                        |
|  <b>须知</b> | 用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。<br>“须知”不涉及人身伤害。 |
|  <b>说明</b> | 对正文中重点信息的补充说明。<br>“说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。                       |



## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

| 修订日期       | 版本 | 修订说明       |
|------------|----|------------|
| 2025-10-30 | 01 | 第1次正式版本发布。 |



# 目录

|                                   |    |
|-----------------------------------|----|
| 前言.....                           | i  |
| 1 DTS 配置参考.....                   | 1  |
| 1.1 serdes 配置.....                | 1  |
| 1.2 local bus 配置.....             | 3  |
| 1.3 sdcard 配置.....                | 3  |
| 1.4 pinctrl 配置与使用.....            | 3  |
| 2 网络.....                         | 5  |
| 2.1 网络相关 DTS 配置.....              | 5  |
| 2.1.1 XGE 模式.....                 | 6  |
| 2.1.2 QSGMII 模式.....              | 7  |
| 2.1.3 GE 模式.....                  | 9  |
| 2.2 DPDK.....                     | 10 |
| 2.2.1 编译环境准备.....                 | 10 |
| 2.2.2 DTS 配置.....                 | 10 |
| 2.2.3 DPDK 编译.....                | 18 |
| 2.2.4 dpdk-l2fwd Sample 使用说明..... | 18 |
| 2.2.5 dpdk-l3fwd Sample 使用说明..... | 19 |
| 2.3 网络驱动.....                     | 20 |
| 2.3.1 业务网口.....                   | 20 |
| 2.3.2 调试网口.....                   | 21 |
| 2.4 网络 RSS.....                   | 22 |
| 3 串口 UART.....                    | 23 |
| 4 开机启动自动插入 ko.....                | 25 |
| 5 BIOS 单板适配修改.....                | 26 |
| 6 BIOS 下 USB 驱动接口使用说明.....        | 28 |
| 7 SD 卡使用说明.....                   | 30 |



## 插图目录

|                            |    |
|----------------------------|----|
| 图 2-1 网口 ping 图.....       | 21 |
| 图 2-2 调试网口 ping 图.....     | 22 |
| 图 4-1 系统默认插入 ko 图.....     | 25 |
| 图 5-1 BIOS 切换串口图.....      | 26 |
| 图 5-2 BIOS 切换 DDR 主频图..... | 27 |
| 图 5-3 DDR 频率参考图.....       | 27 |
| 图 6-1 BIOS 修改图.....        | 28 |
| 图 6-2 文件目录图.....           | 29 |



## 表格目录

|                                            |   |
|--------------------------------------------|---|
| 表 1-1 serdes 常用频点（type）和时钟（reclk）对应关系..... | 2 |
| 表 2-1 Serdes 复用表-MAC 接口排布表.....            | 5 |



# 1 DTS 配置参考

DTS文件位于“mpu\_hi309a\_solution/build/build\_hi309a”目录下。

## 1.1 serdes 配置

serdes相关的DTS配置位于“mpu\_hi309a\_solution/build/build\_hi309a”目录下中。

```
serdes {
    compatible = "hisilicon,hilink32LRT12V101";
    macro_num = <0x03>;
    all_lan_num = <0x0a>;
    status = "okay";
    mask = <0x3ff>;
    macro_mask = <0x07>;
    delay_init = <0x00>;
    /* cs_id : pll选择 0: pll0, 1: pll1*/
    cs_id = <0x01 0x01 0x00 0x00 0x00 0x01 0x00 0x00 0x01 0x01>;
    /* refclk: 参考时钟选择 0=156M, 1=100M*/
    refclk = <0x00 0x00 0x00 0x00 0x01 0x01 0x00 0x00 0x00 0x00>;
    /* type: rate 速率 参考0x02: SGMII 0x6: QSGMII, 0x12:XGE */
    type = <0x06 0x06 0x12 0x12 0x11 0x0d 0x12 0x12 0x06 0x06>;
    /*mac_id: 序号, 同一种模式有多种选择, 比如GE0/复用GE10 填0选择配置为GE0 填10配置为GE10*/
    mac_id = <0x02 0x03 0x00 0x01 0x00 0x00 0x02 0x03 0x04 0x05>;
    pcie_mode = <0x00 0x00 0x00 0x00 0x00 0x02 0x00 0x00 0x00 0x00>;
    pcie_width = <0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
    tx_pn = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00>;
    rx_pn = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00>;
    pcie_gen4_link_mode = <0x00>;
    use_dts_tx_rx_para = <0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00>;
    adapt_en = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00>;
    flexmii_flag = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00>;
    lane0_tx_para = <0x01 0x00 0x01 0x00 0x00 0x04 0x01 0x00 0x3b 0x00>;
    lane0_rx_para = <0x07 0x05 0x07 0x02 0x07 0x00 0x00 0x00 0x00 0x00 0x01 0x01
0x00 0x04 0x04 0x00 0x00 0x00>;
};
```



表 1-1 serdes 常用频点（type）和时钟（reclk）对应关系

| type                   | 值  | 参考时钟                                                                 |
|------------------------|----|----------------------------------------------------------------------|
| E1_SGMII_3125000       | 1  | 速率3125000kHz 参考时钟156M                                                |
| E2_SGMII_1250000       | 2  | 速率1250000kHz 参考时钟156M                                                |
| E4_SGMII_3125000       | 4  | 速率3125000kHz 参考时钟156M                                                |
| E5_SGMII_1250000       | 5  | 速率1250000kHz 与E4_SGMII_3125000可共用PLL 参考时钟156M                        |
| E6_QSGMII_5000000      | 6  | 速率5000000kHz 与E7_SGMII可共用PLL 参考时钟156M                                |
| E7_SGMII_1250000       | 7  | 速率1250000kHz 与E6_QSGMII可共用PLL 参考时钟156M                               |
| E8_QSGMII_5000000      | 8  | 速率5000000kHz 可与PCIE gen1 gen2共用PLL的QSGMII 频点 参考时钟为100M               |
| E9_SGMII_1250000       | 9  | 速率1250000kHz 与PCIE gen1 gen2共用PLL的GE频点 参考时钟为100M                     |
| E10_PCIE_GE_N1_2500000 | 10 | 速率2500000kHz 不开展频 可与E8_QSGMII_5000000、E9_SGMII_1250000可共用PLL参考时钟156M |
| E11_PCIE_GE_N2_5000000 | 11 | 速率5000000kHz 不开展频 可与E8_QSGMII_5000000、E9_SGMII_1250000可共用PLL参考时钟156M |
| E12_PCIE_GE_N1_2500000 | 12 | 速率2500000kHz 可开展频 不可共用PLL 参考时钟156M                                   |
| E13_PCIE_GE_N2_5000000 | 13 | 速率5000000kHz 可开展频 不可共用PLL 参考时钟156M                                   |
| E14_PCIE_GE_N3_8000000 | 14 | 速率8000000kHz 占用两个PLL 参考时钟156M                                        |
| E15_SAS_SAT_A_1500000  | 15 | 速率1500000kHz 参考时钟156M                                                |
| E16_SAS_SAT_A_3000000  | 16 | 速率3000000kHz 参考时钟156M                                                |
| E17_SAS_SAT_A_6000000  | 17 | 速率6000000kHz 参考时钟156M                                                |
| E18_10GE_10312500      | 18 | 速率10312500kHz 参考时钟156M                                               |
| E19_5GE_5156250        | 19 | 速率5156250kHz 参考时钟156M                                                |



## 1.2 local bus 配置

local-bus相关的DTS配置位于“mpu\_hi309a\_solution/build/build\_hi309a”目录下hi309a\_common\_drivers.dtsi中。

```
localbus {
    compatible = "lbc";
    reg = <0x00 0xfa080000 0x00 0x10000>;
    reg-names = "reg";
    interrupts = <0x00 0x62 0x04>;
    bus_num=<0x00>;
    clocks = <0x05 0x18>;
    clock-names = "clk_lbus";
    resets = <0x05 0x18>;
    reset-names = "RST_LBUS_RST0";
    cnt = <0x01>;//片选数量
    cfg0 = <0x00 0x00 0xfa080000 0x00 0x10000000>;//片选0的基地址和大小
};
```

## 1.3 sdcard 配置

sdcard相关的DTS配置位于“mpu\_hi309a\_solution/build/build\_hi309a”目录下hi309a\_common\_drivers.dtsi中。

```
sdhci@fb400000 {
    compatible = "hisilicon,sdhci-hi309a";
    reg = <0xfb400000 0x10000>;
    #address-cells = <0x01>;
    #size-cells = <0x00>;
    interrupts = <0x00 0x61 0x04>;
    bus-width = <0x04>;
    max-frequency = <12000000>;
    non-removable;
    wp-inverted;
    use-pio;
    sdc_ctrl_reg = <0xfa053238>;
    hc-type = <1>;

    mmcard@0 {
        reg = <0x00>;
        compatible = "mmc-card";
        broken-hpi;
    };
};
```

## 1.4 pinctrl 配置与使用

pinctrl相关的DTS配置位于“mpu\_hi309a\_solution/build/build\_hi309a”目录下hi309a\_common\_drivers.dtsi中。pinctrl节点对应的复用功能（具体复用可参考芯片手册）。

```
//pinctrl节点示例配置
pinctrl@FA860000 {
    compatible = "hisi,hi309a-pinctrl";
    reg = <0x00 0xfa860000 0x00 0x10000>;
    status = "okay";
    hisi309a-i2c-basic {
```



```
        i2c1_default {  
            hisi,pins = <0x298 0x30000 0xab8 0x300fb 0x00 0xffffffff 0x29c 0x30000 0xabc  
0x300fb 0x00 0xffffffff>;  
            phandle = <0x0d>;  
        };  
        i2c1_gpio {  
            hisi,pins = <0x298 0x30000 0xab8 0x300fb 0x06 0xffffffff 0x29c 0x30000 0xabc  
0x300fb 0x06 0xffffffff>;  
            phandle = <0x0e>;  
        };  
    };  
};
```

配置需要使用复用的节点。

```
//需要使用pinctrl的节点  
test_pinctrl {  
    compatible = "dp,i2c_gpio_exchage";  
    pinctrl-names = "default\0gpio";  
    pinctrl-0 = <0x0d>;//i2c模式  
    pinctrl-1 = <0x0e>;//gpio模式  
    status = "okay";  
};
```



# 2 网络

## 2.1 网络相关 DTS 配置

表 2-1 Serdes 复用表-MAC 接口排布表

| Port # | SGMII | 10GE    | QSGMII  |
|--------|-------|---------|---------|
| Port0  | GE0   | 10GE2   | QSGMII0 |
| Port1  | GE1   | NotUsed | NotUsed |
| Port2  | GE2   | NotUsed | NotUsed |
| Port3  | GE12  | NotUsed | NotUsed |
| Port4  | GE3   | 10GE3   | QSGMII1 |
| Port5  | GE13  | NotUsed | NotUsed |
| Port6  | GE14  | NotUsed | NotUsed |
| Port7  | GE15  | NotUsed | NotUsed |
| Port8  | GE4   | -       | QSGMII2 |
| Port9  | GE16  | -       | NotUsed |
| Port10 | GE17  | -       | NotUsed |
| Port11 | GE18  | -       | NotUsed |
| Port12 | GE5   | -       | QSGMII3 |
| Port13 | GE19  | -       | NotUsed |
| Port14 | GE20  | -       | NotUsed |
| Port15 | GE21  | -       | NotUsed |
| Port16 | GE6   | -       | QSGMII4 |



| Port # | SGMII | 10GE  | QSGMII  |
|--------|-------|-------|---------|
| Port17 | GE22  | -     | NotUsed |
| Port18 | GE23  | -     | NotUsed |
| Port19 | GE24  | -     | NotUsed |
| Port20 | GE7   | -     | QSGMII5 |
| Port21 | GE8   | -     | NotUsed |
| Port22 | GE9   | -     | NotUsed |
| Port23 | GE25  | -     | NotUsed |
| Port24 | GE10  | 10GE0 | -       |
| Port25 | GE11  | 10GE1 | -       |

#### 约束条件

1. 上述表格总的同一行中，同时只能支持一种速率模式。
2. NotUsed：表示当使用某种速率模式时，其他对应的port无法工作。
3. GE：该端口通过QSGMII来实现，不能作为单独GE口实现。
4. 如果需要使用QSGMII模式，比如使用QSGMII5，那么对应四个网口可用，port号分别为：port20、port21、port22、port23。

### 2.1.1 XGE 模式

如果需要使用XGE模式，请参考如下进行修改：

```
/* 配置 10GE1 M1L1*/
mac_9: mac@9 {
    id = <0x19>;
    port_mode = <0x05>;
    mac_port_speed = <0x06>; /* mac端口速率 0:10M 1: 100M 2: 1000M, 6: invalid */
    sds_speed = <0x06>;
    neg_mode = <0x00>;
};

poe_9: poe@9 {
    vcpu_id = <0x00>;
    add_pkt_grp = <0x00>; /* 采用默认配置：dts中每个网口不能够相同：配置范围：0-15 */
    grp_cfg = <0x01 0x00 0x400>;
};

ppe_9: ppe@9 {
    cport_id = <0x18>;
    dir_mode_en;
    fw_tnl = <0x08 0x08 0x40 0x00 0x00>;
    pool = <0x08 0xc0 0x40 0x02 0x00>;
    set_promisc = <0x00 0x00>;
    fw1_flow_num = <0x02>; // 如果使用网桥模式， fw1_flow_num = <0x03>;
    /**
     * fw1_flow0_key、fw1_flow1_key中需要注意点如下：
     * 1、0x8180000中bit16-bit23（数值0x18）表示网口的port号，port号与cport_id保持一致。
     * 2、其他参数保持不变。
     */
};
```



```
*/
fw1_flow0_key = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x81800000 0x00 0x00
0x00>;
fw1_flow0_mask = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff0000 0x00
0x00 0x00>;
fw1_flow0_ad = <0x00 0x00 0x00 0x08 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
fw1_flow0_update_dmac = <0x01 0x00>;
/**
 * fw1_flow0_ad、fw1_flow1_ad 中需要注意点如下：
 * 比如：0x08与fw_tnl中第一个参数保持一致，其他值保持不变。
 */
fw1_flow1_key = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x81800000 0x00
0x00 0x00>;
fw1_flow1_mask = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff0000 0x00
0x00 0x00>;
fw1_flow1_ad = <0x00 0x00 0x00 0x08 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
fw1_flow1_update_dmac = <0x00 0x00>;
/**
 * 如果需要使用网桥模式，需要复制fw1_flow1_key、fw1_flow1_mask、fw1_flow1_ad、
fw1_flow1_update_dmac，并且修改为
 * fw1_flow2_key、fw1_flow2_mask、fw1_flow2_ad、fw1_flow2_update_dmac
 * 修改fw1_flow2_key、fw1_flow2_mask中参数0xffffffff修改为0x0 0xffff修改为0x0
 */
fw1_flow2_key = <0x0 0x0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x81800000 0x00 0x00
0x00>;
fw1_flow2_mask = <0x0 0x0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff0000 0x00 0x00
0x00>;
fw1_flow2_ad = <0x00 0x00 0x00 0x08 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
fw1_flow2_update_dmac = <0x00 0x00>;
};

ethernet@9 {
compatible = "hi1230_eth,dev";
mac = <&mac_9>; //与mac配置保持一致
poe = <&poe_9>; //与poe配置保持一致
ppe = <&ppe_9>; //与ppe配置保持一致
ifnet-name = "xge1";
flowctrl_rx_pause_en = <0x1>;
};
```

#### 须知

- 在配置网口复用时，同一个MACRO两个pll同时只支持两种速率共存。
- 同一个MACRO里面，MAC中port\_mode模式不一样时，需要配置不同的pll，比如同一个MACRO里面存在两个不同速率，则需要配置不同的pll，10GE可配置pll1，SGMII配置可pll0。

## 2.1.2 QSGMII 模式

如果需要使用QSGMII模式，请参考如下进行修改。

```
/* 例如：需要使用QSGMII 2 */
/* 配置 QSGMII 模式*/
mac_1: mac@1 {
id = <0x08>; /* mac_port id 查看对应的serdes复用表格 */
port_mode = <0x03>;
mac_port_speed = <0x02>; /* mac端口速率 0:10M 1: 100M 2: 1000M */
sds_speed = <0x00>;
neg_mode = <0x01>;
```



```
};
poe_1: poe@1 {
    vcpu_id = <0x00>;
    add_pkt_grp = <0x00>; /* 采用默认值 */
    grp_cfg = <0x01 0x00 0x400>;
};
ppe_1: ppe@1 {
    cport_id = <0x08>;
    dir_mode_en;
    fw_tnl = <0x00 0x00 0x00 0x00 0x00>;
    pool = <0x00 0xc0 0x40 0x02 0x00>;
    set_promisc = <0x00 0x00>;
    fw1_flow_num = <0x02>; // 如果使用网桥模式， fw1_flow_num = <0x03>;
    fw1_flow0_key = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80800000 0x00 0x00
0x00>;
    fw1_flow0_mask = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff00000 0x00
0x00 0x00>;
    fw1_flow0_ad = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
    fw1_flow0_update_dmac = <0x01 0x00>;
    fw1_flow1_key = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80800000 0x00
0x00 0x00>;
    fw1_flow1_mask = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff00000 0x00
0x00 0x00>;
    fw1_flow1_ad = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
    fw1_flow1_update_dmac = <0x00 0x00>;
    /**
     * 如果需要使用网桥模式，需要复制fw1_flow1_key、fw1_flow1_mask、fw1_flow1_ad、
fw1_flow1_update_dmac，并且修改为
     * fw1_flow2_key、fw1_flow2_mask、fw1_flow2_ad、fw1_flow2_update_dmac
     * 修改fw1_flow2_key、fw1_flow2_mask中参数0xffffffff修改为0x0 0xffff修改为0x0
     */
    fw1_flow2_key = <0x0 0x0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80800000 0x00 0x00
0x00>;
    fw1_flow2_mask = <0x0 0x0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff00000 0x00 0x00
0x00>;
    fw1_flow2_ad = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
    fw1_flow2_update_dmac = <0x00 0x00>;
};
ethernet@1 {
    compatible = "hi1230_eth,dev";
    phy-handle = <&mdio0_phy14>; // 如果单板存在phy芯片时，需要加入该字段，其中
mdio1_phy14取与dts里面的mdio下挂的phy的编号
    mac = <&mac_1>; //与mac配置保持一致
    poe = <&poe_1>; //与poe配置保持一致
    ppe = <&ppe_1>; //与ppe配置保持一致
    ifnet-name = "eth1";
    flowctrl_rx_pause_en = <0x1>;
};
```

#### 须知

- 必须正确配置相应的QSGMII父节点，例如使用QSGMII 2时，需配置与Port8相关的MAC、PPE和POE及网口模块。
- 在同一MACRO中，两个PLL仅支持两种速率的同时共存。若在同一MACRO内采用不同模式，则需为不同的PLL配置不同的频率点。



## 2.1.3 GE 模式

如果需要使用GE模式，请参考如下进行配置。

```
/* 例如：需要使用GE4，对应的port口为8*/
/* 配置 GE 模式*/
mac_8: mac@8 {
    id = <0x08>; /* mac_port id 查看对应的serdes复用表格 */
    port_mode = <0x02>;
    mac_port_speed = <0x02>; /* mac端口速率 0:10M 1: 100M 2: 1000M */
    sds_speed = <0x00>;
    neg_mode = <0x01>;
};

poe_8: poe@8 {
    vcpu_id = <0x00>;
    add_pkt_grp = <0x00>; /* 采用默认值 */
    grp_cfg = <0x01 0x00 0x400>;
};

ppe_8: ppe@8 {
    cport_id = <0x08>;
    dir_mode_en;
    fw_tnl = <0x08 0x08 0x40 0x00 0x00>;
    pool = <0x08 0xc0 0x40 0x02 0x00>;
    set_promisc = <0x00 0x00>;
    fw1_flow_num = <0x02>; // 如果使用网桥模式， fw1_flow_num = <0x03>;
    fw1_flow0_key = <0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80800000 0x00 0x00
0x00>;
    fw1_flow0_mask = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff00000 0x00
0x00 0x00>;
    fw1_flow0_ad = <0x00 0x00 0x00 0x08 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
    fw1_flow0_update_dmac = <0x01 0x00>;
    fw1_flow1_key = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80800000 0x00
0x00 0x00>;
    fw1_flow1_mask = <0xffffffff 0xffff 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff00000 0x00
0x00 0x00>;
    fw1_flow1_ad = <0x00 0x00 0x00 0x08 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
    fw1_flow1_update_dmac = <0x00 0x00>;
    /**
     * 如果需要使用网桥模式，需要复制fw1_flow1_key、fw1_flow1_mask、fw1_flow1_ad、
     fw1_flow1_update_dmac，并且修改为
     * fw1_flow2_key、fw1_flow2_mask、fw1_flow2_ad、fw1_flow2_update_dmac
     * 修改fw1_flow2_key、fw1_flow2_mask中参数0xffffffff修改为0x0 0xffff修改为0x0
     */
    fw1_flow2_key = <0x0 0x0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80800000 0x00 0x00
0x00>;
    fw1_flow2_mask = <0x0 0x0 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xffff00000 0x00 0x00
0x00>;
    fw1_flow2_ad = <0x00 0x00 0x00 0x08 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00>;
    fw1_flow2_update_dmac = <0x00 0x00>;
};

ethernet@8 {
    compatible = "hi230_eth,dev";
    // phy_handle = <&mdio1_phy1>; // 如果单板存在phy芯片时，需要加入该字段，其中
mdio1_phy1取决与dts里面的mdio下挂的phy的编号
    mac = <&mac_8>; //与mac配置保持一致
    poe = <&poe_8>; //与poe配置保持一致
    ppe = <&ppe_8>; //与ppe配置保持一致
    ifnet-name = "eth9";
    flowctrl_rx_pause_en = <0x1>;
};
```



## 2.2 DPDK

### 2.2.1 编译环境准备

编译环境：

- Python 3.4+、OpenSSL 1.1.1f、CMake 3.20.5+，GLIBC\_2.35及以上。
- 安装工具meson 0.61.2+、ninja 1.11.1+。

**步骤1** 从以下路径下载openEuler-24.03-LTS交叉编译链到~/hi309A\_tool目录。

[https://mirror.truenetwork.ru/openeuler/openEuler-24.03-LTS/embedded\\_img/aarch64/qemu-aarch64/openeuler-glibc-x86\\_64-openeuler-image-aarch64-qemu-aarch64-toolchain-24.03-LTS.sh](https://mirror.truenetwork.ru/openeuler/openEuler-24.03-LTS/embedded_img/aarch64/qemu-aarch64/openeuler-glibc-x86_64-openeuler-image-aarch64-qemu-aarch64-toolchain-24.03-LTS.sh)

**步骤2** 执行如下命令。

```
sh openeuler-glibc-x86_64-openeuler-image-aarch64-qemu-aarch64-toolchain-24.03-LTS.sh -d /home/${USER}/hi309A_tool/toolchain/ -y \
```

**步骤3** 配置openEuler交叉编译环境，执行如下命令。

```
./home/${USER}/hi309A_tool/toolchain/environment-setup-aarch64-openeuler-linux
```

----结束

#### 须知

- 路径前是“点(.) 空格( )”，点和空格都不能少。
- 在分模块编译过程中，编译模块所在服务器的CMake需要在3.20.5版本及以上。

### 2.2.2 DTS 配置

网口配置请参考如下进行修改。

```
dtpdketh@0 {
    compatible = "udrvdpdk,net_pmd";
    id = <0>; /* DPDK使用端口索引 */
    port_id = <0x8>; /* mac_port id 查看对应的serdes复用表格 */
    port_mode = <0x02>; /* 2:SGMII 3: QSGMII 5: 10GBASE_R */
    port_speed = <0x02>; /* 2: 1000M 5: 10G */
    mac_auto_neg = <0x01>; /* SGMII、QSGMII模式保留，XGE模式删除此字段 */
    mac_duplex = <0x01>; /* SGMII、QSGMII模式保留，XGE模式删除此字段 */

    /* serdes_lane_id serdes_lane_num
    * serdes_lane_id：取值范围：0-9
    * serdes_lane_num
    */
    serdes_lane = <0x00 0x01>;
    rss_rxq_num = <0>; /* 0:dpdk转发非rss方案, 8:dpdk rss方案，二层三层转发示例使用8个接收队列. */
    fd_fw_tmplt_id = <0x01>; /* 采用默认值 */
    fw_fd_res = <0x40 0x01>; /* 根据需求修改，范围0x32~0x53 */

    /* fw_tnl*以及fw_tnl_cnt只需要在dts中存在一份，dts中其他dtpdketh节点中不需要配置fw_tnl_cnt以及fw_tnl*字段 */
}
```



```
/** fw_tnl_id pool_id base_vqi qos_sel default_qos
    poe_qos_drop_en poe_qos_bp_grp order_type order_tag_sel cache_alloc_len
    * pool_id: 范围0x0~0x0f
    * base_vqi: 范围0x0~0x98
    */

fw_tnl_cnt = <0x0c>;
fw_tnl0 = <0x40 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x0d>;//0x0d
fw_tnl1 = <0x41 0x01 0x08 0x00 0x01 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl2 = <0x42 0x02 0x10 0x00 0x02 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl3 = <0x43 0x03 0x18 0x00 0x03 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl4 = <0x44 0x04 0x20 0x00 0x04 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl5 = <0x45 0x05 0x28 0x00 0x05 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl6 = <0x46 0x06 0x30 0x00 0x06 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl7 = <0x47 0x07 0x38 0x00 0x07 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl8 = <0x48 0x08 0x40 0x00 0x00 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl9 = <0x49 0x09 0x48 0x00 0x01 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl10 = <0x4a 0x0a 0x50 0x00 0x02 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl11 = <0x4b 0x0b 0x58 0x00 0x03 0x01 0x00 0x00 0x00 0x0d>;

checksum_tmplt_num = <0x01>; /* 该字段只需要在dts中存在一份，dts中其他dpdketh节点
中不需要配置该字段*/

fw_fd@0 {
    src_port = <8 0x1f>;
    ad_next_step = <0x00>;
    ad_fw_tnl = <0x40>; /* 根据需求填写fw_tnl*中fw_tnl_id 值 */
    ad_stats_cnt_id = <0x01>;
};

/* checksum_tmplt节点只需要在dts中存在一份，dts中其他dpdketh节点中不需要配置
checksum_tmplt节点*/
checksum_tmplt@0 {
    sctp_check_en = <0x00>;
    udp_check_en = <0x00>;
    tcp_check_en = <0x01>;
    l3_check_en = <0x01>;
};
};
```

SDK包中默认DPDK使用非RSS方案，可修改DTS文件切换为RSS方案，DTS修改点参考如下。

修改点1:

```
ppe {
    compatible = "hisi, ppe-v110";
    #address-cells = <0x02>;
    #size-cells = <0x02>;
    bd_version = <0x00>;
    status = "okay";

    pf {
        reg = <0x00 0xf0000000 0x00 0x800000>;
        vf_num = <0x01>;
        deft_fw_mod = <0x01>;
        deft_fw_tnl = <0x00>;

        /* 使用rss方案时释放enable rss use for dpdk path注释域内容，删除默认重复字
        段。*/
        /* enable rss use for dpdk path
        enable_rss = <0x01>;
        mam_pool_num_vf0 = <0x08>;
        mam_pool_depth_vf0 = <512 0x40 0x40 0x40 0x40 0x40 0x40 0x40>;
```



```
*/  
  
/* 使用rss方案时删除如下默认字段。 */  
enable_rss = <0x00>; // 0 means disable rss  
mam_pool_num_vf0 = <0x10>;  
mam_pool_depth_vf0 = <0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40>;  
  
ppe_cport_alloc_num = <0x1a>;  
smmu_type = <0x00>;  
fw_grp_num_per_vf = <0x10>;  
bd_endian = <0x00>;  
fw1_flow_res_range = <0x00 0x00>;  
fw1_flow_bit_mode = <0x01>;  
fw2_flow_res_range = <0x01 0x01>;  
fw2_flow_bit_mode = <0x01>;  
ig1_flow_res_range = <0x00 0x00>;  
ig1_flow_bit_mode = <0x02>;  
ig2_flow_res_range = <0x01 0x01>;  
ig2_flow_bit_mode = <0x02>;  
eg1_flow_res_range = <0x02 0x02>;  
eg1_flow_bit_mode = <0x02>;  
  
clk_ppe_group_0 {  
    clk_group_id = <0x00>;  
    clocks = <0x04 0x01>;  
    clock-names = "CLK_PPE_CLK0";  
    at_user = <0x01>;  
};  
  
rst_ppe_group_0 {  
    rst_group_id = <0x00>;  
    resets = <0x04 0x01 0x04 0x02>;  
    reset-names = "RST_PPE_RST0\0RST_PPE_RST1";  
    at_user = <0x01>;  
};  
  
vf0 {  
    reg = <0x00 0xf0800000 0x00 0x100000>;  
    buf_endian_mode = <0x00>;  
    buf_adp_en = <0x00>;  
    buf_big_adp_en = <0x00>;  
};  
};  
  
修改点2:  
dpdketh@0 {  
    compatible = "udrvdpdk,net_pmd";  
    id = <0>;  
    port_id = <0x8>;  
    port_mode = <0x02>;  
    port_speed = <0x02>;  
    mac_auto_neg = <0x01>;  
    mac_duplex = <0x01>;  
    serdes_lane = <0x00 0x01>;  
  
    /* 8:dpdk rss方案，二层三层转发示例使用8个接收队列。 */  
    rss_rxq_num = <8>;  
    fd_fw_tmplt_id = <0x01>;  
    fw_fd_res = <0x40 0x01>;  
  
    /* fw_tnl*以及fw_tnl_cnt只需要在dts中存在一份，dts中其他dpdketh节点中不需要配置
```



```
fw_tnl_cnt以及fw_tnl*字段 */
/* 使用rss方案时释放enable rss use注释域内容，删除默认重复字段。 */
/* enable rss use
fw_tnl_cnt = <256>;
fw_tnl_rss0 = <0x00 0x00 0x00 0x00 0 0 0x00 0x00 0x00 0>;
fw_tnl_rss1 = <0x00 0x00 0x08 0x00 0 0 0x00 0x00 0x00 0>;
fw_tnl_rss2 = <0x00 0x00 0x10 0x00 0 0 0x00 0x00 0x00 0>;
fw_tnl_rss3 = <0x00 0x00 0x18 0x00 0 0 0x00 0x00 0x00 0>;
fw_tnl_rss4 = <0x00 0x00 0x20 0x00 0 0 0x00 0x00 0x00 0>;
fw_tnl_rss5 = <0x00 0x00 0x28 0x00 0 0 0x00 0x00 0x00 0>;
fw_tnl_rss6 = <0x00 0x00 0x30 0x00 0 0 0x00 0x00 0x00 0>;
fw_tnl_rss7 = <0x00 0x00 0x38 0x00 0 0 0x00 0x00 0x00 0>;
*/

/* 使用非rss方案时，使用此区域配置。 */
fw_tnl_cnt = <0x0c>;
fw_tnl0 = <0x40 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x0d>;//0x0d
fw_tnl1 = <0x41 0x01 0x08 0x00 0x01 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl2 = <0x42 0x02 0x10 0x00 0x02 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl3 = <0x43 0x03 0x18 0x00 0x03 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl4 = <0x44 0x04 0x20 0x00 0x04 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl5 = <0x45 0x05 0x28 0x00 0x05 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl6 = <0x46 0x06 0x30 0x00 0x06 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl7 = <0x47 0x07 0x38 0x00 0x07 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl8 = <0x48 0x08 0x40 0x00 0x00 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl9 = <0x49 0x09 0x48 0x00 0x01 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl10 = <0x4a 0x0a 0x50 0x00 0x02 0x01 0x00 0x00 0x00 0x0d>;
fw_tnl11 = <0x4b 0x0b 0x58 0x00 0x03 0x01 0x00 0x00 0x00 0x0d>;

checksum_tmplt_num = <0x01>; /* 该字段只需要在dts中存在一份，dts中其他dpdketh节点
中不需要配置该字段*/

fw_fd@0 {
    src_port = <8 0x1f>;
    ad_next_step = <0x00>;
    ad_fw_tnl = <0x40>;
    ad_stats_cnt_id = <0x01>;
};

/* checksum_tmplt节点只需要在dts中存在一份，dts中其他dpdketh节点中不需要配置
checksum_tmplt节点*/
checksum_tmplt@0 {
    sctp_check_en = <0x00>;
    udp_check_en = <0x00>;
    tcp_check_en = <0x01>;
    l3_check_en = <0x01>;
};

修改点3:
dpdkevent@0 {
    compatible = "udrvidpk,event";
    id = <0x00>;
    group_cfg_cnt = <0x0c>;
    port_cnt = <0x0c>;
    group_cfg_temp0 = <0x00 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0x00 0x01 0x02 0x03 0x04
0x05 0x06 0x07>;

    group@0 {
        group_id = <0x00>;
        bp_thrd = <0xffff>; // 512 can solve performance drop
        cfg_temp_id = <0x00>;
    };
};
```



```
group@1 {  
    group_id = <0x01>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@2 {  
    group_id = <0x02>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@3 {  
    group_id = <0x03>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@4 {  
    group_id = <0x04>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@5 {  
    group_id = <0x05>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@6 {  
    group_id = <0x06>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@7 {  
    group_id = <0x07>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@8 {  
    group_id = <0x08>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@9 {  
    group_id = <0x09>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@10 {  
    group_id = <0x0a>;  
    bp_thrd = <0xffff>;  
    cfg_temp_id = <0x00>;  
};  
  
group@11 {  
    group_id = <0x0b>;
```



```
        bp_thrd = <0xffff>;
        cfg_temp_id = <0x00>;
    };

    /* 使用rss方案时释放enable rss use注释域内容，删除默认重复字段。 */
/* eanble rss use
    port@0 {
        id = <0x00>;
        port_type = <0x00>;
        lcpu = <0x00>;
        group = <0x00>;
    };

    port@1 {
        id = <0x01>;
        port_type = <0x00>;
        lcpu = <0x00>;
        group = <0x00>;
    };

    port@2 {
        id = <0x02>;
        port_type = <0x00>;
        lcpu = <0x00>;
        group = <0x00>;
    };

    port@3 {
        id = <0x03>;
        port_type = <0x00>;
        lcpu = <0x00>;
        group = <0x00>;
    };

    port@4 {
        id = <0x04>;
        port_type = <0x00>;
        lcpu = <0x00>;
        group = <0x00>;
    };

    port@5 {
        id = <0x05>;
        port_type = <0x00>;
        lcpu = <0x00>;
        group = <0x00>;
    };

    port@6 {
        id = <0x06>;
        port_type = <0x00>;
        lcpu = <0x00>;
        group = <0x00>;
    };

    port@7 {
        id = <0x07>;
        port_type = <0x00>;
        lcpu = <0x00>;
        group = <0x00>;
    };
};
```



```
port@8 {
    id = <0x08>;
    port_type = <0x00>;
    lcpu = <0x00>;
    group = <0x00>;
};

port@9 {
    id = <0x09>;
    port_type = <0x00>;
    lcpu = <0x00>;
    group = <0x00>;
};

port@10 {
    id = <0x0a>;
    port_type = <0x00>;
    lcpu = <0x00>;
    group = <0x00>;
};

port@11 {
    id = <0x0b>;
    port_type = <0x00>;
    lcpu = <0x00>;
    group = <0x00>;
};
*/
/* 使用非rss方案时，使用此区域配置。 */
port@0 {
    id = <0x00>;
    port_type = <0x00>;
    lcpu = <0x00>;
    group = <0x00>;
};

port@1 {
    id = <0x01>;
    port_type = <0x00>;
    lcpu = <0x01>;
    group = <0x01>;
};

port@2 {
    id = <0x02>;
    port_type = <0x00>;
    lcpu = <0x02>;
    group = <0x02>;
};

port@3 {
    id = <0x03>;
    port_type = <0x00>;
    lcpu = <0x03>;
    group = <0x03>;
};

port@4 {
    id = <0x04>;
    port_type = <0x00>;
    lcpu = <0x04>;
    group = <0x04>;
};
```



```
};

port@5 {
    id = <0x05>;
    port_type = <0x00>;
    lcpu = <0x05>;
    group = <0x05>;
};

port@6 {
    id = <0x06>;
    port_type = <0x00>;
    lcpu = <0x06>;
    group = <0x06>;
};

port@7 {
    id = <0x07>;
    port_type = <0x00>;
    lcpu = <0x07>;
    group = <0x07>;
};

port@8 {
    id = <0x08>;
    port_type = <0x00>;
    lcpu = <0x08>;
    group = <0x08>;
};

port@9 {
    id = <0x09>;
    port_type = <0x00>;
    lcpu = <0x09>;
    group = <0x09>;
};

port@10 {
    id = <0x0a>;
    port_type = <0x00>;
    lcpu = <0x0a>;
    group = <0x0a>;
};

port@11 {
    id = <0x0b>;
    port_type = <0x00>;
    lcpu = <0x0b>;
    group = <0x0b>;
};
};
```

#### 须知

1. SDK版本包中hi309a\_m900\_uart0\_dpdk.dts默认RSS方案，不支持非RSS方案。
2. SDK版本包中hi309a\_demo\_uart0\_dpdk.dts默认非RSS方案，支持修改DTS文件切换为RSS方案。



## 2.2.3 DPDK 编译

### 步骤1 执行编译脚本。

```
cd mpu_hi309a_solution/build  
./build.sh -f tmpfs -e dpdk -u uart0 -b demo
```

参数说明：

- f：表示编译出的文件系统类型，可选择tmpfs（内存文件系统）、ext4、mini
- u：表示编译出的文件系统和设备采用的串口，可选择uart0、uart1
- e：表示编译出来的设备树，可选择eth（内核态模式）、dpdk（dpdk模式）、eth\_dpdk（内核态和dpdk网口共存模式）
- b：表示单板类型，可选择demo、m900

### 步骤2 编译生成的文件在“mpu\_hi309a\_solution/build/output”目录下。

- hi309a\*\*.dtb：设备树。
- zImage：升级的系统镜像。
- initrd\_boot.cpio.gz：文件系统。

----结束

## 2.2.4 dpdk-l2fwd Sample 使用说明

前提条件：根据所需按照[DTS配置](#)、[DPDK编译](#)说明编译对应设备树、系统镜像、文件系统。

### 步骤1 openEuler系统升级完成。

### 步骤2 进入openEuler系统后，执行如下命令：

```
cd /lib/udrivers  
insmod motorcomm.ko  
insmod hns_mdio.ko  
insmod hi309a_urm.ko  
insmod hi309a_urm_adapt.ko  
insmod hi309a_poe.ko  
insmod hi309a_ppe.ko  
insmod hi309a_mag.ko  
insmod igb_uio.ko  
insmod rte_kni.ko  
insmod hi309a_kdrv_dfx.ko  
insmod dpdk_phy.ko  
mkdir -p /dev/hugepages  
mountpoint -q /dev/hugepages || mount -t hugetlbfs nodev /dev/hugepages  
echo 1024 > /proc/sys/vm/nr_hugepages
```

### 步骤3 单板与网络相关测试仪器对接后，根据系统中网口的数量，动态修改执行的参数。

非RSS方案网口数据为8个时，可以执行如下命令：

```
非RSS方案  
dpdk-l2fwd -l 0,1,2,3,4,5,6,7 -n 1 -- -q 1 -p 0xff -T 2000
```

RSS方案仅支持2个网口，RSS方案可以执行如下命令：

```
RSS方案  
dpdk-l2fwd -l 0,1,2,3,4,5,6,7 -n 1 -- -q 1 -p 0x3 -T 2000
```

参数说明：

- l 2,3 绑定在2 和 3号核上。
- n 每个CPU的内存通道数。
- q 每个CPU管理的队列数，这里设置为一个队列。
- p 使用的端口的掩码：0x1 表示2的0次方使用的是port 0。  
使用的是port 1是2的一次方，也就是2 即0x2。



使用的是port 2，就是2的2次方，也就是4，即0x4。  
使用的是port 3，就是2的3次方，也就是8，即0x8。  
则此时-p的值是，2的0次方+2的1次方，也就是3，即此时参数为：-p 0x3  
-T 100 每隔100s显示一下转发的统计计数，此处可以设置大一些，防止频繁打印导致性能下降。  
--portmap 指定端口映射关系。比如--portmap="(1,2),(0,3,4,5,6,7)" 第一个括号(1,2)表示端口1，2绑定到第一个核心(核心1)；第二个括号(0,3,4,5,6,7)表示其他端口绑定到第2个核心(核2)

如果需要结束运行，则按CTRL+C结束DPDK的运行。

----结束

#### 须知

1. 需要phy相关的驱动时，客户需要提供phy相关驱动源码。在文件制作过程中，客户需要把phy的源码编译出来ko放入文件系统里面。
2. RSS方案需要在“mpu\_hi309a\_solution/open\_source/dpdk/examples/l2fwd/main.c”文件中定义RSS\_ENABLE 1宏定义。非RSS方案需要删除此宏定义。
3. RSS方案进行转发性能测试时，输入报文需要进行源IP变化使得报文可以散列到不同队列上，让不同核心进行转发处理，如果固定报文测试，性能与单核一致。
4. RSS方案转发端口仅支持设备树中dpdketh中id=0、id=1两个端口之间数据转发。
5. 使用hi309a\_m900\_uart0\_dpdk.dts测试l2fwd示例时，必须使能l2fwd示例中的RSS\_ENABLE宏。

## 2.2.5 dpdk-l3fwd Sample 使用说明

前提条件：三层转发示例仅支持RSS方案，按照[DTS配置](#)、[DPDK编译](#)说明编译RSS方案对应DPDK的设备树、系统镜像、文件系统。DTS路径位于“mpu\_hi309a\_solution/build/build\_hi309a”目录下。

**步骤1** openEuler系统升级完成。

**步骤2** 进入openEuler系统后，执行如下命令：

```
cd /lib/udrivers
insmod motorcomm.ko
insmod hns_mdio.ko
insmod hi309a_urm.ko
insmod hi309a_urm_adapt.ko
insmod hi309a_poe.ko
insmod hi309a_ppe.ko
insmod hi309a_mag.ko
insmod igb_uio.ko
insmod rte_kni.ko
insmod hi309a_kdrv_dfx.ko
insmod dpdk_phy.ko
mkdir -p /dev/hugepages
mountpoint -q /dev/hugepages || mount -t hugetlbfs nodev /dev/hugepages
echo 1024 > /proc/sys/vm/nr_hugepages
```

**步骤3** 单板与网络相关测试仪器对接后，执行l3fwd启动参数。

RSS方案  
dpdk-l3fwd -l 0,1,2,3,4,5,6,7 -n 1 -- -p 0x3 --config="(0,0,0),(0,1,1),(0,2,2),(0,3,3),(0,4,4),(0,5,5),(0,6,6),(0,7,7)" --parse-ptype

参数说明：

-l 2,3 绑定在2 和 3号核上。  
-n 每个CPU管理的队列数，这里设置为一个队列。  
-p 使用的端口的掩码：0x1 表示2的0次方使用的是port 0。  
使用的是port 1是2的一次方，也就是2 即0x2。



使用的是port 2，就是2的2次方，也就是4，即0x4。  
使用的是port 3，就是2的3次方，也就是8，即0x8。  
则此时-p的值是，2的0次方+2的1次方，也就是3，即此时参数为：-p 0x3  
-config 三元组（端口id,核id,队列id），队列id必须从0开始。  
--parse-ptype 使能包类型解析功能。

如果需要结束运行，则按CTRL+C结束DPDK的运行。

----结束

#### 须知

1. 需要phy相关的驱动时，客户需要提供phy相关驱动源码。在文件制作过程中，客户需要把phy的源码编译出来ko放入文件系统里面。
2. RSS方案进行转发性能测试时，输入报文需要进行源IP变化使得报文可以散列到不同队列上，让不同核心进行转发处理，如果固定报文测试，性能与单核一致。
3. RSS方案转发端口仅支持设备树中dpdketh中id=0、id=1两个端口之间数据转发。ip与端口绑定关系是port0-ip为198.18.0.xx，port1-ip为198.18.1.xx。
4. 路由算法使用示例中默认LPM。
5. l3fwd默认支持RSS方案，不支持非RSS方案。

## 2.3 网络驱动

### 2.3.1 业务网口

**步骤1** openEuler系统升级完成。

#### 须知

- 如果DTS配置多个网口时，需要在执行**步骤2**之前，在对应网口的lan口中插入对应的速率的光模块，保证两端对接的光模块支持最高速率一致的。
- 如果客户单板需要phy相关的驱动时，客户需要提供phy相关驱动源码。在文件制作过程中，客户需要把phy的源码编译出来ko放入文件系统里面。

**步骤2** 进入openEuler单板如果需要使用phy驱动，比如phy驱动的motorcomm.ko位于“/lib/udrivers”目录下，则执行如下命令。

```
cd /lib/udrivers/  
insmod motorcomm.ko  
insmod hns_mdio.ko
```

**步骤3** 进入openeuler系统起来后，执行如下命令。

```
cd /lib/udrivers/  
insmod hi309a_urm.ko  
insmod hi309a_poe.ko  
insmod hi309a_ppe.ko  
insmod hi309a_mag.ko  
insmod hi309a_eth_drv.ko
```

**步骤4** 加载完驱动后，执行命令ifconfig -a，出现网口，例如出现eth8网口。

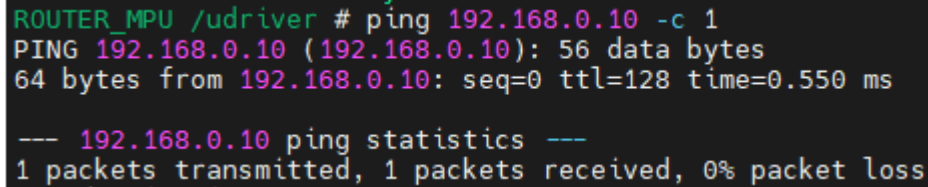


**步骤5** 配置网口的网络，比如配置eth8网口，eth8的网段与pc直连网线的以太网处于同一个网段，执行如下命令。

```
ifconfig eth8 192.168.0.12
ifconfig eth8 up
ethtool eth8
```

**步骤6** 配置完成后，在串口上ping直连pc的IP地址。

图 2-1 网口 ping 图



```
ROUTER_MPU /udriver # ping 192.168.0.10 -c 1
PING 192.168.0.10 (192.168.0.10): 56 data bytes
64 bytes from 192.168.0.10: seq=0 ttl=128 time=0.550 ms

--- 192.168.0.10 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
```

----结束

## 2.3.2 调试网口

**步骤1** openEuler系统升级完成。

### 须知

- 在执行插入调试相关网口的ko之前，首先查看系统是否已经插入调试网口所依赖的ko，避免重复插入。
- 使用调试网口时，不支持使用业务网口（即不加载业务网口相关驱动）。
- 客户需要提供phy相关驱动源码。在文件制作过程中，客户需要把phy的源码编译出来ko放入文件系统里面。

**步骤2** 进入openEuler单板如果需要使用phy驱动，比如phy驱动的motorcomm.ko位于“/lib/udrivers”目录下，则执行如下命令。

```
cd /lib/udrivers/
insmod motorcomm.ko
insmod hns_mdio.ko
```

**步骤3** 执行如下命令加载调试网口。

```
cd /lib/udrivers/
insmod hi309a_urm.ko
insmod hi309a_poe.ko
insmod hi309a_ppe.ko
insmod hi309a_mag.ko
insmod hi309a_mii_mac.ko
```

**步骤4** 加载完驱动后，执行命令ifconfig -a，出现调试网口，例如出现ethmii网口。

**步骤5** 配置网口的网络，比如配置ethmii网口，ethmii的网段与pc直连网线的以太网处于同一个网段，执行如下命令。

```
ifconfig ethmii 192.168.0.12
ifconfig ethmii up
ethtool ethmii
```

**步骤6** 配置完成后，在串口上ping直连PC的IP地址，如图2-2所示。



图 2-2 调试网口 ping 图

```
ROUTER_MPU /udriver # ping 192.168.0.10 -c 1
PING 192.168.0.10 (192.168.0.10): 56 data bytes
64 bytes from 192.168.0.10: seq=0 ttl=128 time=0.550 ms

--- 192.168.0.10 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
```

----结束

## 2.4 网络 RSS

目前内核态模式的网口常用默认值，如果需要使用RSS功能，修改如下。

```
ppe {
    compatible = "hisi, ppe-v110";
    #address-cells = <0x02>;
    #size-cells = <0x02>;
    bd_version = <0x00>;
    status = "okay";
    pf {
        reg = <0x00 0xf0000000 0x00 0x800000>;
        vf_num = <0x01>;
        deft_fw_mod = <0x01>;
        deft_fw_tnl = <0x00>;
        /* 如果需要打开RSS功能，打开注释
        enable_rss = <0x01>;
        mam_pool_num_vf0 = <0x04>;
        mam_pool_depth_vf0 = <0x100 0x100 0x100 0x100>;
        */
        /* 默认是打开该注释
        enable_rss = <0x00>;
        mam_pool_num_vf0 = <0x10>;
        mam_pool_depth_vf0 = <0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40 0x40
        0x40 0x40 0x40 0x40>;
        */
    };
};
```



# 3 串口 UART

目前UART串口默认使用UART0，如果系统的输出使用其他的UART串口时，需要修改如下地方。

前提条件：

- 文件系统的重新构建，需要添加管理员权限。
- DTS位于mpu\_hi309a\_solution/build/build\_hi309a。

修改处如下：

1. 修改DTS里面的bootargs节点，比如系统需要使用UART1，那么修改如下。

修改前：

```
chosen {  
    bootargs = "root=/dev/ram0 console=ttyS0,115200 vmalloc=512M  
earlycon=uart8250,mmio32,0xfa510000 rdinit=/sbin/init maxcpus=8 initrd=0x6020000,64M  
ramdisk_size=100000 ekbox=0x200000$0x10300000 kbox_mem=0x1E00000@0x12B00000  
oops=panic min_loop=32 swiotlb=1 iomem=relaxed";  
};
```

修改后：

```
chosen {  
    bootargs = "root=/dev/ram0 console=ttyS1,115200 vmalloc=512M  
earlycon=uart8250,mmio32,0xfa520000 rdinit=/sbin/init maxcpus=8 initrd=0x6020000,64M  
ramdisk_size=100000 ekbox=0x200000$0x10300000 kbox_mem=0x1E00000@0x12B00000  
oops=panic min_loop=32 swiotlb=1 iomem=relaxed";  
};
```

执行如下命令重新生成dtb文件。

```
dtc -I dts -O dtb -o hi309a.dtb hi309a.dts
```

DTB文件位于mpu\_hi309a\_solution/build/build\_hi309a。

## 须知

- UART0对接ttyS0，UART1对接ttyS1。
  - 文件系统的制作需要添加管理员权限。
2. 根据DTS（设备树）修改文件系统中ttyS\*，比如DTS中bootargs 中uart1对应着ttyS1，文件系统initrd\_boot.cpio.gz位于“mpu\_hi309a\_solution/build/output”目录下，需要执行如下命令。



```
cd mpu_hi309a_solution/build/output
mkdir rootfs
cd rootfs
zcat ../initrd_boot.cpio.gz | cpio -dim
sed -i "s/115200 ttyS0/115200 ttyS1 xterm/g" "etc/inittab"
sed -i "s/ttyS0/ttyS1:/g" "etc/inittab"
sudo chown -R root:users .
sudo bash -c "find . | cpio -H newc -o --quiet | gzip -c --quiet > ../initrd_boot.cpio.gz"
```

重新生成的“initrd\_boot.cpio.gz”存放在“mpu\_hi309a\_solution/build/output”目录下。



# 4 开机启动自动插入 ko

**步骤1** 进入“mpu\_hi309a\_solution/build/build\_fs”目录，修改rc.sysinit文件添加需要开机自动插入的ko，如图4-1所示。

图 4-1 系统默认插入 ko 图

```
24 #insmod
25 if [ -d /lib/udrivers/ ]
26 then
27     echo "ko insmod start:"
28     insmod /lib/udrivers/ksecrec.ko
29     insmod /lib/udrivers/hi309a_subctrl.ko
30     insmod /lib/udrivers/hi309a_pinctrl.ko
31     insmod /lib/udrivers/hi309a_gpio.ko
32     insmod /lib/udrivers/hi309a_spi.ko
33     insmod /lib/udrivers/hi309a_i2c.ko
34     insmod /lib/udrivers/hi309a_pmbus.ko
35     insmod /lib/udrivers/hi309a_sfc.ko
36     insmod /lib/udrivers/hi309a_serdes.ko
37     insmod /lib/udrivers/hi309a_pcie.ko
38     insmod /lib/udrivers/hi309a_sata.ko
39     insmod /lib/udrivers/hi309a_log.ko
40     insmod /lib/udrivers/hi309a_tsensor.ko
41     insmod /lib/udrivers/hi309a_lbc.ko
42     insmod /lib/udrivers/usbcore.ko
43     insmod /lib/udrivers/xhci-hcd.ko
44     insmod /lib/udrivers/hi309a_usb_hisi.ko
45     insmod /lib/udrivers/xhci-plat-hcd.ko
46     insmod /lib/udrivers/scsi_mod.ko
47     insmod /lib/udrivers/crc10dif_common.ko
48     insmod /lib/udrivers/crc-t10dif.ko
49     insmod /lib/udrivers/t10-pi.ko
50     insmod /lib/udrivers/sd_mod.ko
51     insmod /lib/udrivers/usb-storage.ko
52     insmod /lib/udrivers/libata.ko
53     insmod /lib/udrivers/libahci.ko
54     insmod /lib/udrivers/libahci_platform.ko
55     insmod /lib/udrivers/ahci_platform.ko
56     insmod /lib/udrivers/hi309a_pcie_api_test.ko
57     echo "ko insmod end"
58 fi
```

**步骤2** 重新执行“./build\_hi309a\_cpio.sh”打包文件系统。

----结束



# 5 BIOS 单板适配修改

参考如下说明修改BIOS相关代码。

目前BIOS支持Demo板，DDR主频为2133M，调试串口为UART0，若要有单板进行适配，做如下修改。

1. 单板调试串口切换为UART1，如[图5-1](#)所示。文件路径：  
mpu\_hi309a\_solution/src/BIOS/BIOS/HwPkg/UEFI/Products/Network/  
Platforms/Hi309a/Hi309APkg.dsc。

图 5-1 BIOS 切换串口图

```
209  ## Serial Terminal
210  gEfiMdeModulePkgTokenSpaceGuid.PcdSerialRegisterBase|0xFA520000
211  gEfiMdePkgTokenSpaceGuid.PcdUartDefaultBaudRate|9600
212  gArmTokenSpaceGuid.PcdVFPEnabled|0
213  gHwTokenSpaceGuid.PcdUartClkInHz|200000000
214  gHwTokenSpaceGuid.PcdSerialPortSendDelay|500000
215
```

2. 修改DDR主频，如[图5-2](#)所示。文件路径：  
mpu\_hi309a\_solution/src/BIOS/BIOS/HwPkg/UEFI/Products/Network/script/  
hi309a/fw\_hi309a.pcfx。  
在里面先找到对应的board\_id，然后修改对应配置项里的ddrfreqm。



图 5-2 BIOS 切换 DDR 主频图

```
38 #CAPP_BT_1
39 @board_id : 0xFFFD {
40     boardtype=0xFFFD
41
42     #cpu频率: 单位M 1600=0x640; 2000=0x7D0; 2200=0x898
43     cpufreqm=0x898
44
45     #串口波特率: 9600=0x2580; 38400=0x9600; 115200=0x1C200
46     baudrate=0x1C200
47
48     #DDR参数配置 0xC代表2133
49     ddrfreqm = 0xC
50     #DDR in rank mirror使能配置
51     inrankmirroren = 1
52
53     kernelloaderaddr = 0xA80000
54     kerneladdr = 0x20
55     kernelsize = 0x1FFC000
56     bootcmdloaderaddr = 0x6000000
57     bootcmdaddr = 0x10000
58     bootcmdsize = 0x10000
59     initrdloaderaddr = 0x6020000
60     initrdaddr = 0x10080
61     initrdsize = 0x4000000
62
63     #MII调试网口硬件配置,如果没有enable参数或者enable=0,则网口不初始化,其他参数不用配置
64     #mode(0:MII 1:RVMMII 2:RGMMII) speed(#0:10M 1:100M 2:1000M, only RGMMII support 1000M)
65     #duplex(#0:MMAC_DUPLEX_HALF 1:MMAC_DUPLEX_FULL) neg_mode(#0:MMAC_NEG_DISABLE 1:define MMAC_NEG_ENABLE)
66     miidrv.enable = 1
67     miidrv.mode = 2
68     miidrv.speed = 2
69     miidrv.duplex = 1
70     miidrv.neg_mode = 1
71 }
```

参考频率如图5-3所示。

图 5-3 DDR 频率参考图

```
DDR_FREQ_1333,
DDR_FREQ_1400,
DDR_FREQ_1600,
DDR_FREQ_1800,
DDR_FREQ_1866,
DDR_FREQ_2000, // 19
DDR_FREQ_2100,
DDR_FREQ_2133,
DDR_FREQ_2200,
DDR_FREQ_2400,
DDR_FREQ_2600,
DDR_FREQ_2666,
DDR_FREQ_2800,
DDR_FREQ_2933,
DDR_FREQ_3000,
DDR_FREQ_3200, // 29
DDR_FREQ_3400,
DDR_FREQ_3600,
DDR_FREQ_3800,
DDR_FREQ_4000,
DDR_FREQ_4200,
DDR_FREQ_4400,
DDR_FREQ_4600,
DDR_FREQ_4800,
DDR_FREQ_5000,
DDR_FREQ_5200, // 39
DDR_FREQ_5400,
DDR_FREQ_5600,
DDR_FREQ_5800,
DDR_FREQ_6000,
DDR_FREQ_6200,
DDR_FREQ_6400,
DDR_FREQ_6800,
DDR_FREQ_7200,
DDR_FREQ_7600,
DDR_FREQ_8000, // 49
DDR_FREQ_8400,
DDR_FREQ_8800,
DDR_FREQ_MAX
DDR_FREQUENCY_INDEX;
```



# 6 BIOS 下 USB 驱动接口使用说明

**步骤1** 目前BIOS下支持通过u盘启动系统，文件系统为ext4。客户做二次开发，可参考mpu\_hi309a\_solution/src/BIOS/BIOS/HwPkg/UEFI/Common/Library/HwBdsLib/HwBdsLib.c，如图6-1所示。

图 6-1 BIOS 修改图

```
STATIC BOOLEAN CopyOSFromUsb(Cfg_Type *AddrCfg)
{
    ... EFI_STATUS Ret;
    ... USB_PROTOCOL *Usb = NULL;
    ... Ret = gBS->LocateProtocol(&gUsbProtocolGuid, NULL, (VOID *)&Usb);
    ... if (EFI_ERROR(Ret)) {
    ...     DEBUG((EFI_D_ERROR, "Protocol failed when get gNorFlashProtocolGuid failure\n"));
    ...     return FALSE;
    ... }

    ... Ret = Usb->Init(Usb);
    ... if (EFI_ERROR(Ret)) {
    ...     DEBUG((DEBUG_ERROR, "Init usb failed(%d)\r\n", Ret));
    ...     return FALSE;
    ... }

    ... // zImage
    ... Ret = Usb->Read(Usb, 1, (UINT8 *) (AddrCfg->Kernel.Dst)); /* 1 is kernel */
    ... if (EFI_ERROR(Ret)) {
    ...     DEBUG((DEBUG_ERROR, "Copy kernel from usb failed(%d)\r\n", Ret));
    ...     return FALSE;
    ... }

    ... // bootcmd
    ... Ret = Usb->Read(Usb, 2, (UINT8 *) (AddrCfg->Bootcmd.Dst)); /* 2 is dtb */
    ... if (EFI_ERROR(Ret)) {
    ...     DEBUG((DEBUG_ERROR, "Copy bootcmd from usb failed(%d)\r\n", Ret));
    ...     return FALSE;
    ... }

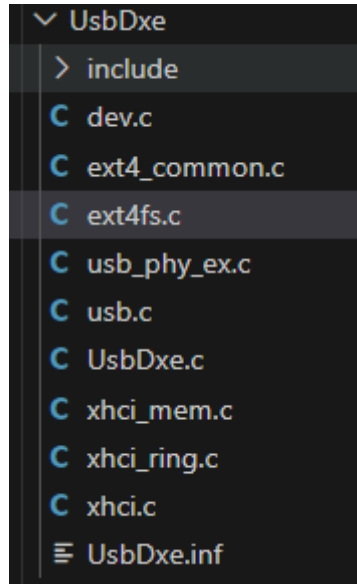
    ... // initrd
    ... Ret = Usb->Read(Usb, 3, (UINT8 *) (AddrCfg->Initrd.Dst)); /* 3 is rootfs */
    ... if (EFI_ERROR(Ret)) {
    ...     DEBUG((DEBUG_ERROR, "Copy initrd from usb failed(%d)\r\n", Ret));
    ...     return FALSE;
    ... }

    ... return TRUE;
}
```



**步骤2** USB驱动和文件系统的具体实现在mpu\_hi309a\_solution/src/BIOS/BIOS/HwPkg/UEFI/Common/Drivers/UsbDxe，如图6-2所示。

图 6-2 文件目录图



**步骤3** 代码中绑定了系统启动的相关文件名和步骤，客户可参考，亦可以调用如下接口进行封装。

1. int host\_self\_usb\_check(void);
2. int ext4fs\_size(const char \*filename, unsigned int \*size);
3. int ext4\_read\_file(const char \*filename, void \*buf, unsigned int offset, unsigned int len, unsigned int \*len\_read);

----结束



# 7 SD 卡使用说明

eMMC与SD卡共用同一控制器，不能同时使用，使用时通过DTS配置区分，当前默认为eMMC DTS配置，SD卡DTS配置参考[sdcard配置](#)，启动后插入hi309a\_emmc.ko即可使用。