## RTL8367RB SWITCH调试

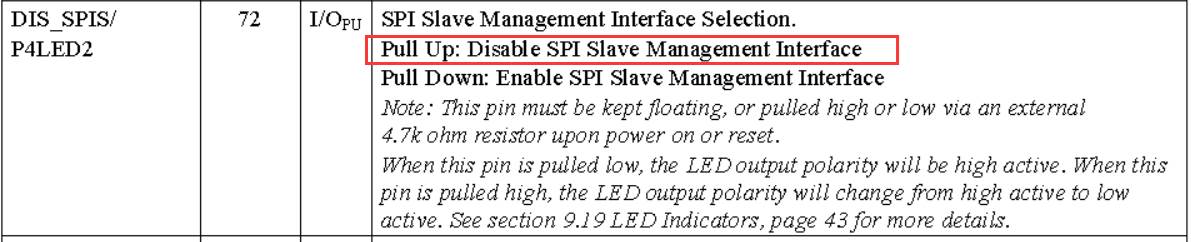
## 一、硬件配置注意事项：

1、swtich关键pin 默认状态

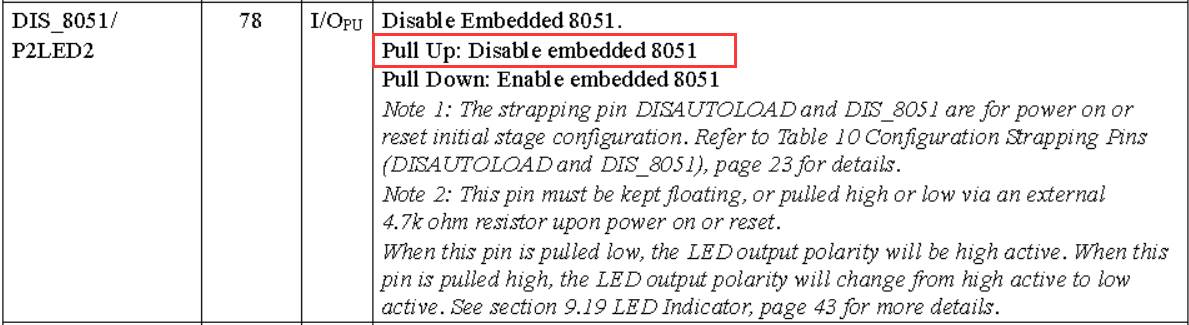
8367rb默认可以通过3种方式去配制switch 寄存器1: (gpio 模拟SCK/SDA)、 2: SPI 3:(MDC/MDIO) . 我们目前配合软件调试用的是第3种。

原厂建议这几个管脚请硬件核查确认：

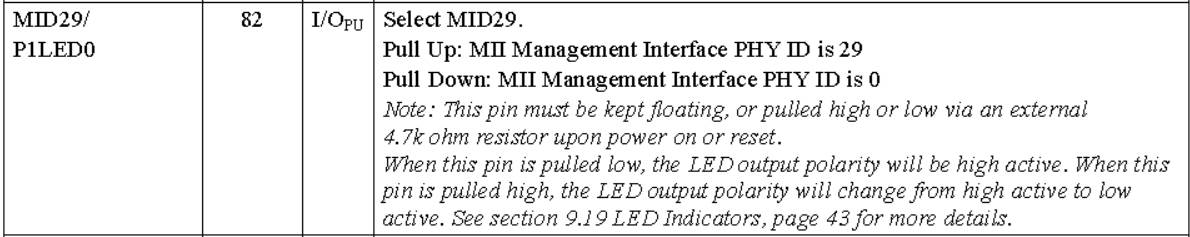
Pin72 我们是通过mdc/mdio 去配制switch, 所以关闭这个接口, 默认接上拉电阻



Pin78原本这个switch可以外接C51去驱动，如果不通过C51,使用我们RK的主控驱动需要有上拉4.7K电阻



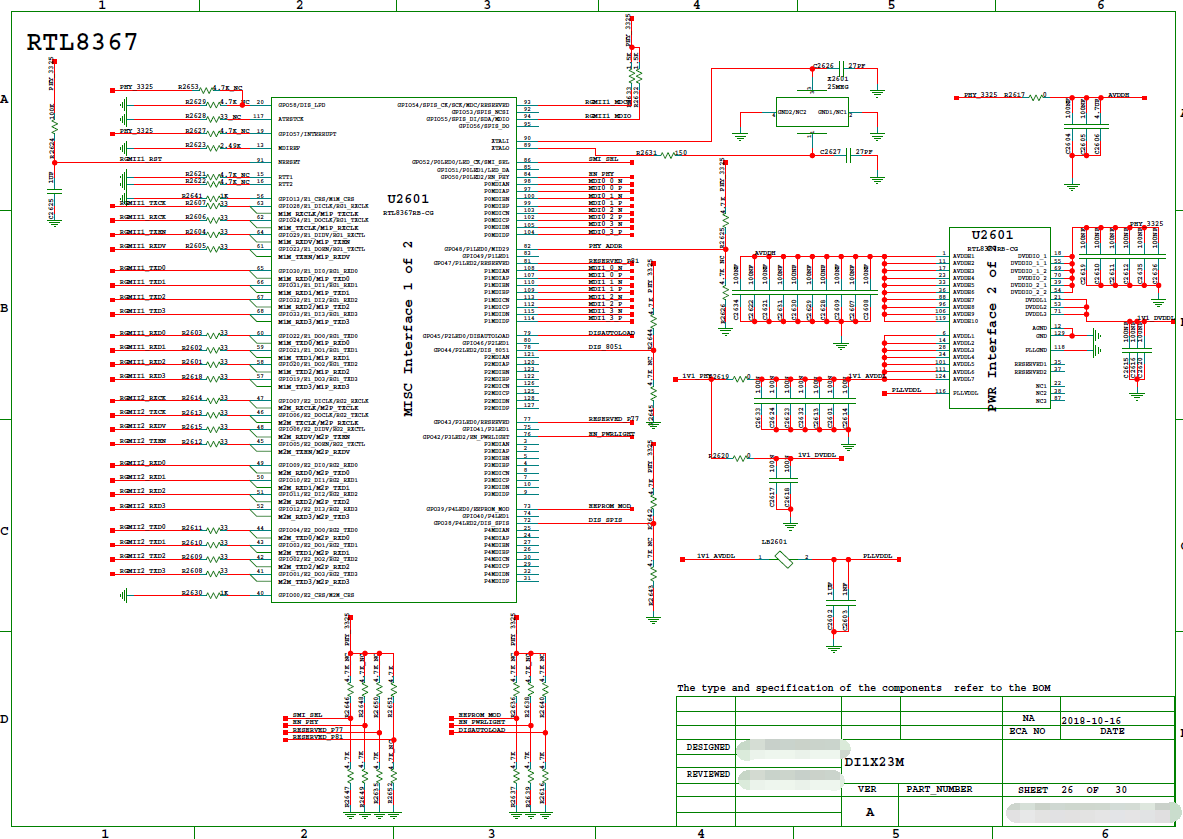
Ping82 是PHY\_ADDR选择， 上拉4.7K phy\_id 定为29， 下拉4.7K phy\_id 0；注意软件中的设定要与硬件匹配，否则无法识别, 目前提供的代码默认按上拉4.7K处理。后面软件部分说明。

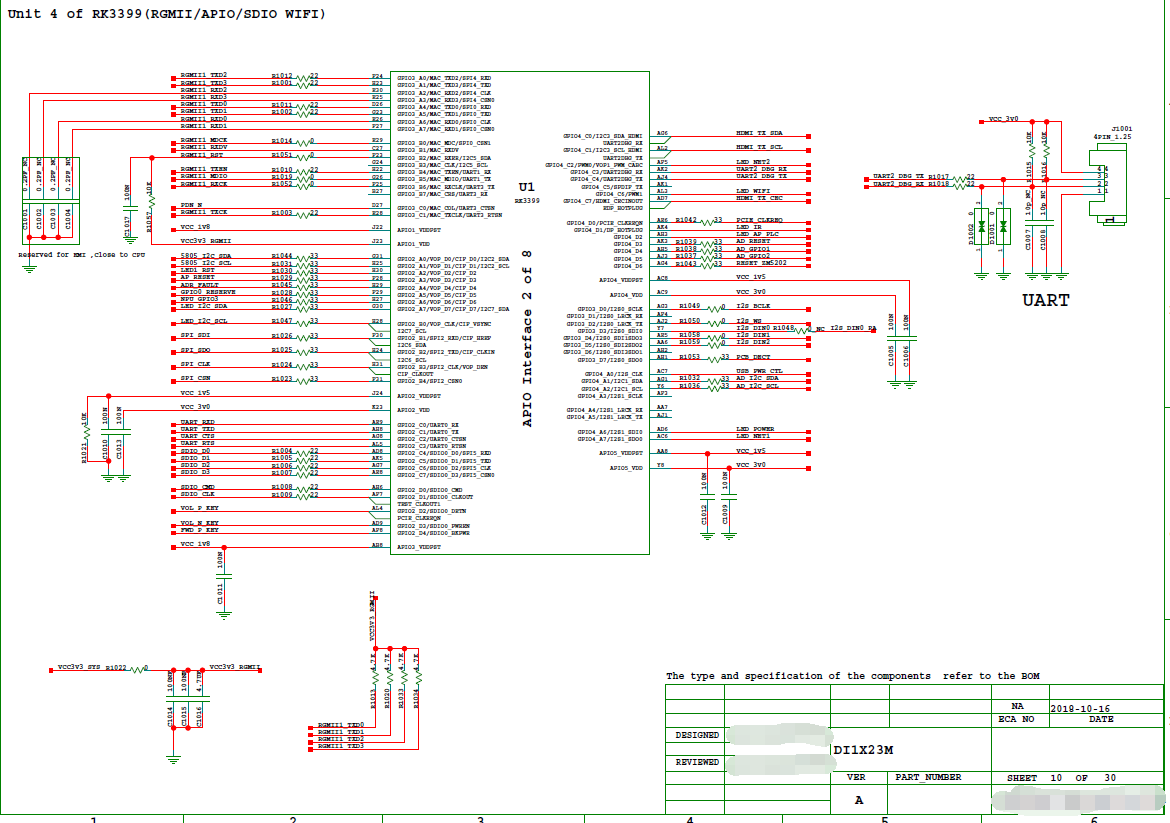


Pin86 配制switch 参数，这颗芯片不带EEPROM，需要软件配制默认参数，默认SMI\_SEL低。

2、硬件接法

8367RB是自带MAC的，相当于主控MAC 与switch MAC RGMII 对接 ，所以硬件上不需要交叉接，即switch端的RGMII\_TXD0接到RK主控端的MAC\_TXD0，switch端的RGMII\_RXD0接到主控端的MAC\_RXD0.具体的硬件接法见图。



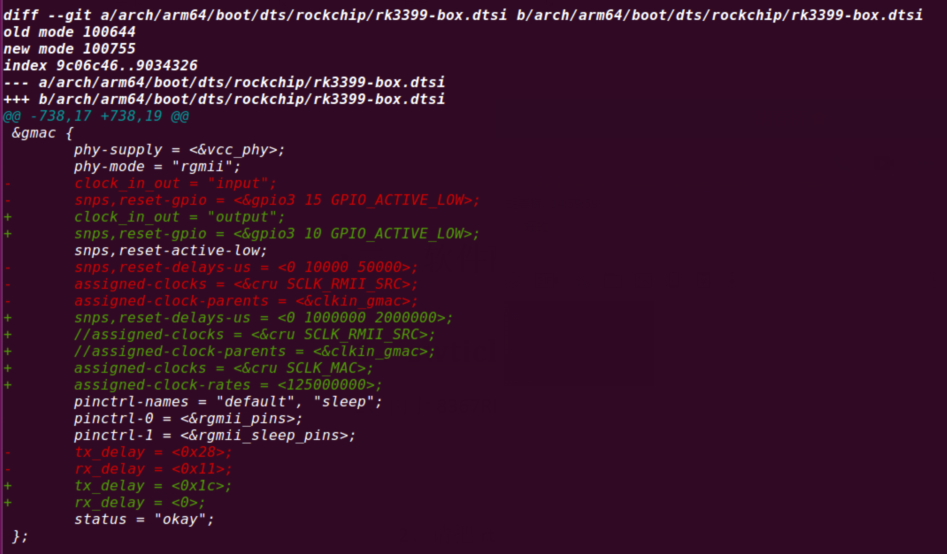


## 二、软件配制注意事项

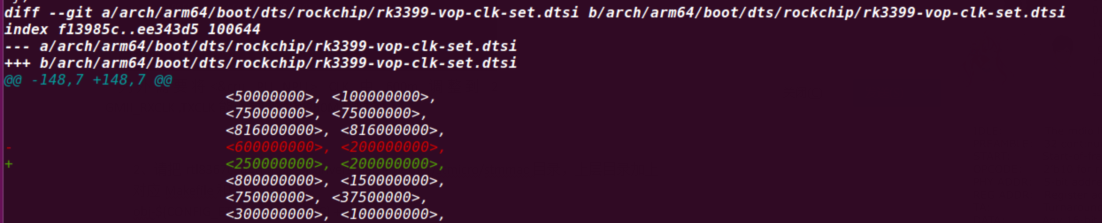
目前调试以RK3399 ANDROID8.1 代码为例，调试RTL8367RB,需要注意以下一些内容。

1、swtich相关口线配置

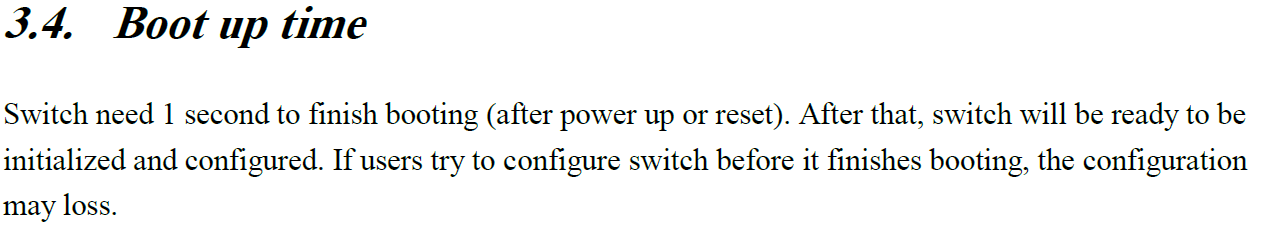
1、由于8367RB不提供125M mac\_clk, 所以需要主控配制125M tx clock

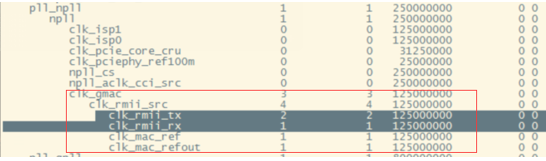


另外需要将<&cru PLL\_NPLL> CLK 由600m调整到250M,保障输出的NPLL GMII\_RXCLK ,TXCLK能正常的输出250M的时钟。



2、需要确定RTL8367RB的复位管脚，以及复位延迟。按照RTL要求复位管脚低电平有效，然后拉高 延迟1到2s 才会开始通过MDIO/MDC去读写寄存器。

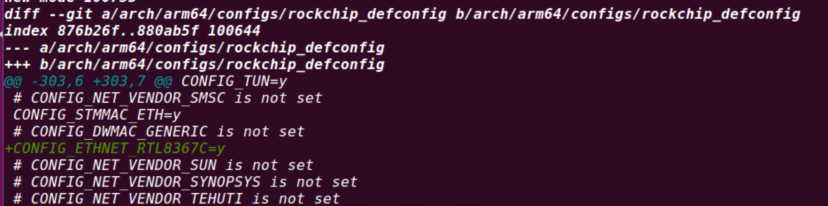


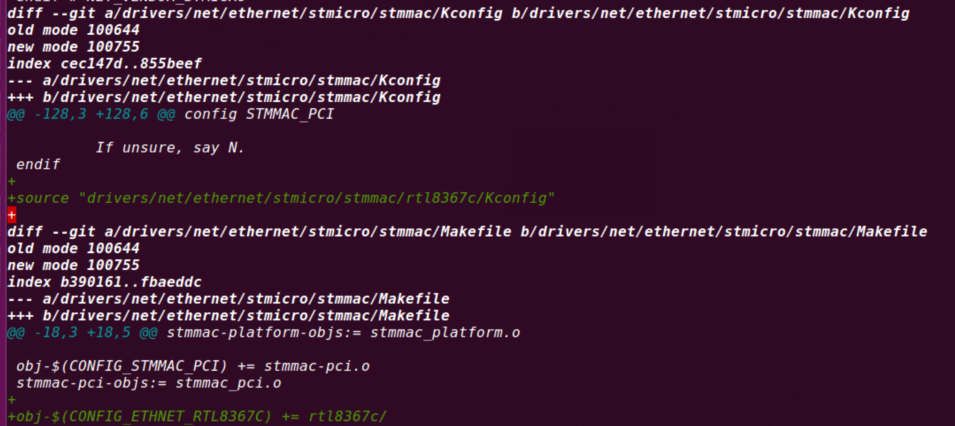
做完以上1 和2 之后，可以通过串口命令cat /d/clk/clk\_summary查看

其次用示波器量一下主控端的TXCLK,RXCLK是否输出150M正确的波形出来，确认复位信号正确的复位。确认好之后，可以通过PC端设置千兆网口，然后通过网线去连接到盒子端，确认pc端是否显示正确的已连接的状态，如果不行就检查硬件口线，以及clk tree。按道理软件不需要做过多的配置，都会正确的显示已连接的状态，就可以确定硬件工作是正常的。

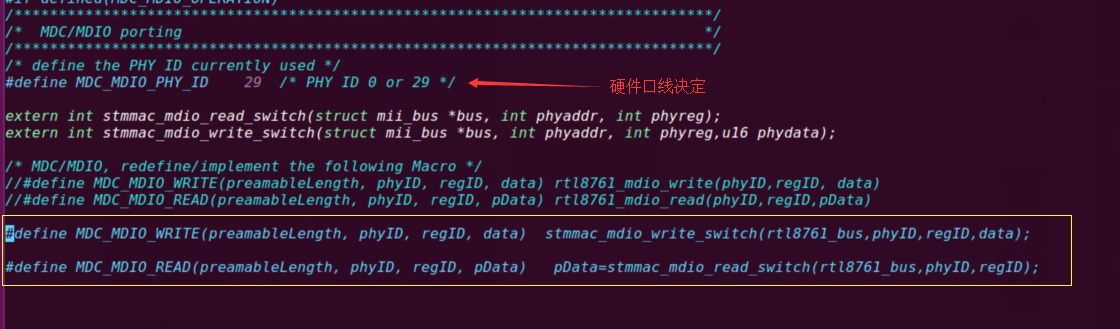
2、确认switch的读写

1、确认完前面的第1,2之后，就可以开始调试rtl8367rb的driver了。由于RTL switch厂商提供的driver本身都是一些.c和.h文件，并没有makefile 和kconfig 所以这部分需要自己手动完成。目前我自己提供的针对RTL8367RB 的driver见压缩包rtl8367c.tgz, 将其解压到drivers/net/ethernet/stmicro/stmmac/目录下。然后需要配置以下截图，使其可以正常编译。



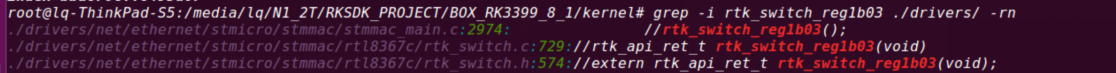


2、确认前面谈到的硬件接法PIN82 是PHY\_ADDR选择，目前switch代码这块位于drivers/net/ethernet/stmicro/stmmac/rtl8367c/smi.c 中的MDC\_MDIO\_PHY\_ID,另外switch自己封装了一层读写接口，见黄色框，后面使用的是RK 封装的读写接口，确认可以正常的读写。



3、确认好这些读写接口之后，需要将我们sdk中默认的phy的MDC/MDIO的操作接口关闭，否则会有影响，导致无法正确的拿到数据，这个比较重要，后面会谈到。确认好这些之后，我们可以尝试一些接口，去读一下0x1b03的寄存器看能否拿到正确的数据，default 默认值为0x0432。

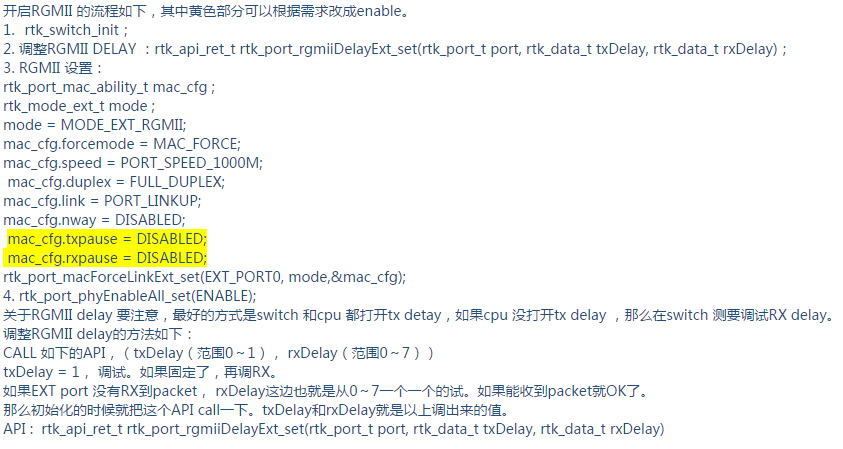
drivers/net/ethernet/stmicro/stmmac/stmmac\_main.c里面stmmac\_init\_phy函数中打开rtk\_switch\_reg1b03()这个函数；./drivers/net/ethernet/stmicro/stmmac/rtl8367c/rtk\_switch.c中打开这个函数实体；在./drivers/net/ethernet/stmicro/stmmac/rtl8367c/rtk\_switch.h中打开这个声明。



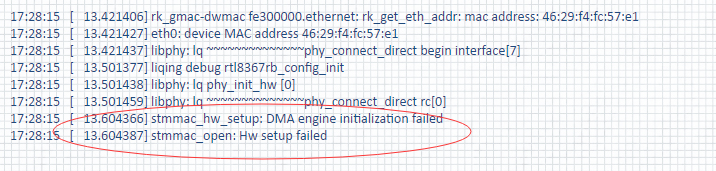
打开这些函数，通过串口打印看能否读取到0x0432，如果能正确的读取，则MDC/MDIO通讯没有问题。

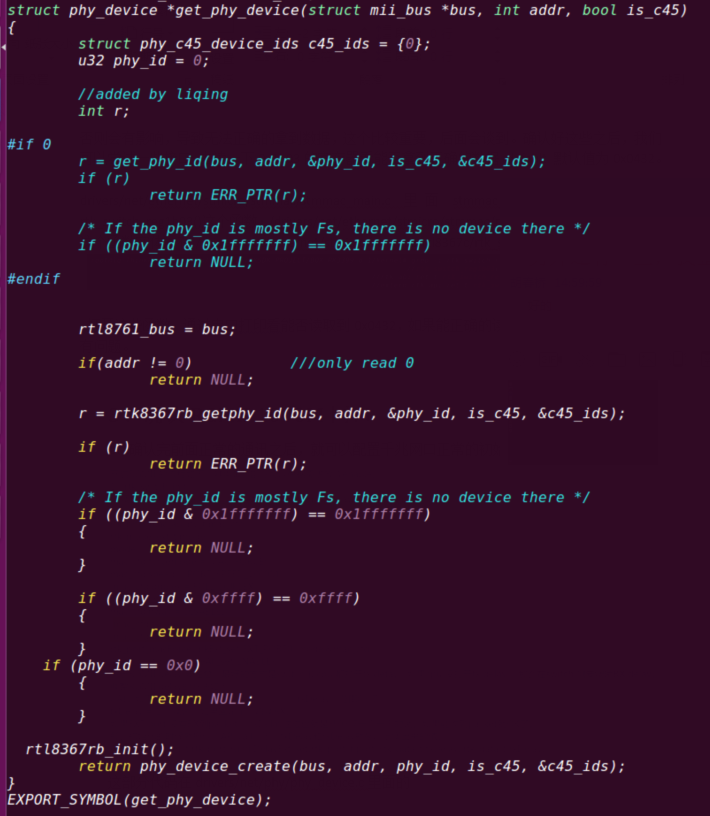
3、确认RGMII 1000M网口

1、确认完前面正常的通讯之后，就可以配置千兆网口正常的初始化动作。



这部分代码在drivers/net/phy/phy\_device.c里面的去获取phy\_ID的位置重新改写。确认完可以正常通过switch的初始化，确认程序可以继续往下走的话，就可以拿到eth0的相关信息，比方busybox ifconfig等内容。Log上出现下面截图的错误就和这个关系，以及clk tree有关。

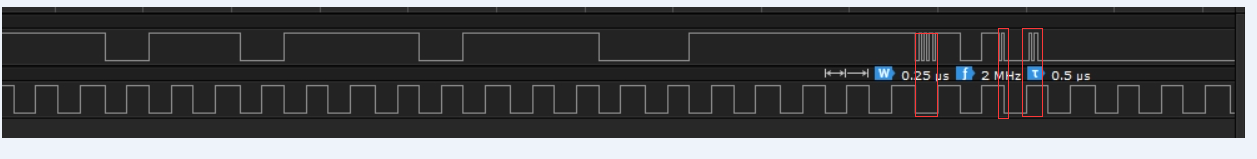




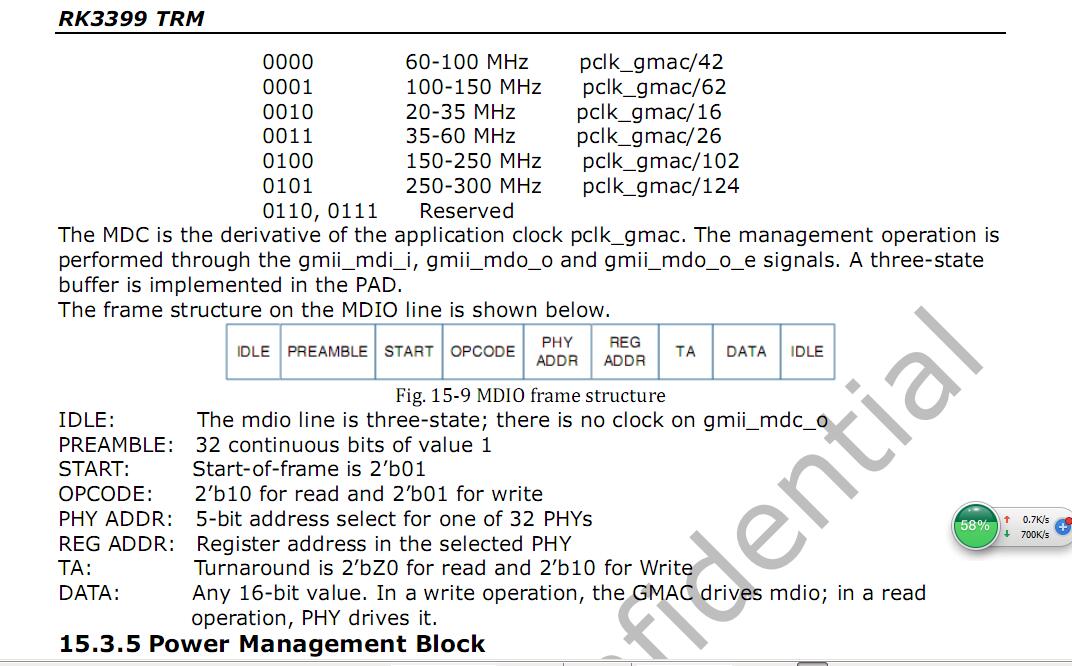
## 三、可能遇到的问题点

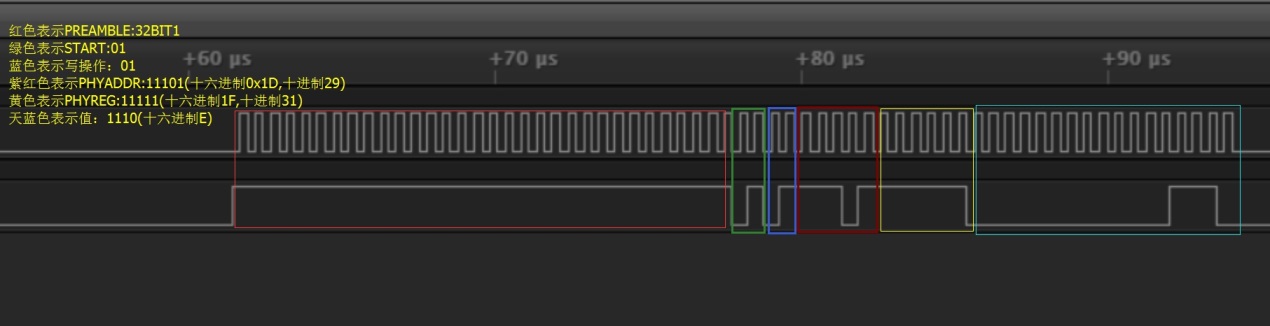
#### 问题点1 MDIO/MDC出现问题

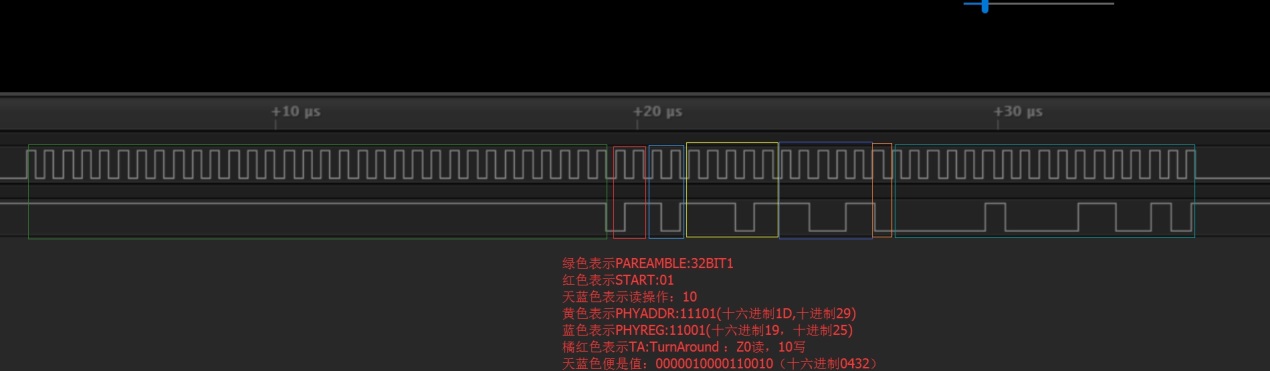
用逻辑分析仪录的mdc/mdio 数据异常，一笔clock 里有好几笔数据，不知道是模块回的不正确，还是主控导致模块异常：

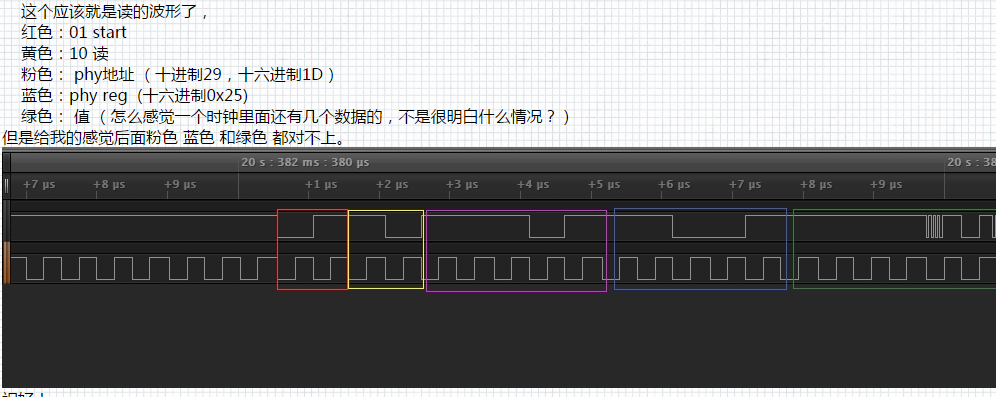


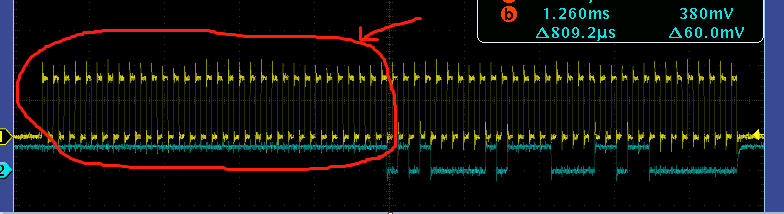
正常mdc/mdio波形应该如下：



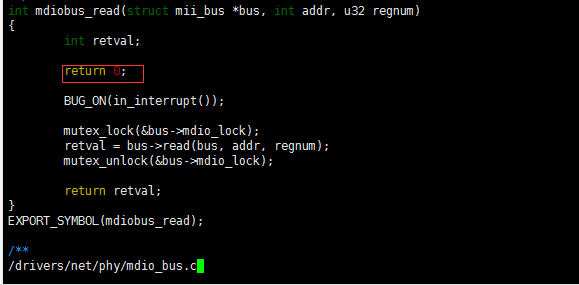


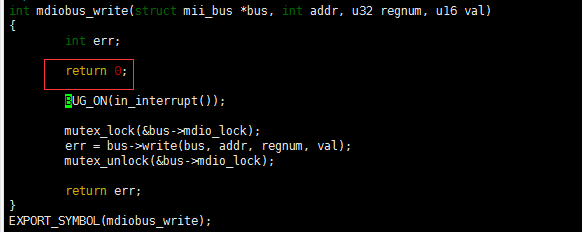




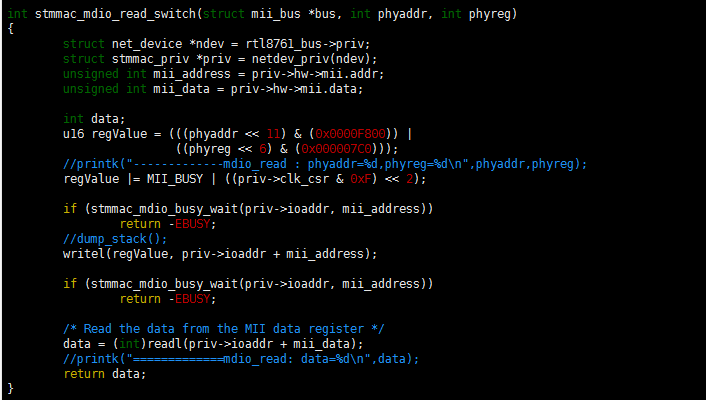


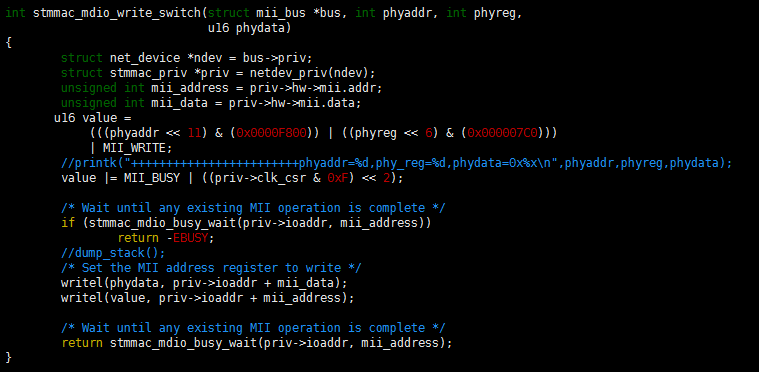
后经过查明是因为初始化时有phy 寄存操作导致，对于switch而言需要干净寄存器操作，我们做如下修改，把原本寄存操作直接返回，会对switch 专门提供寄存器操作函数:





增加单独接口，其实这个呼叫以前接口就可以了，只是调试时候改成这样了。

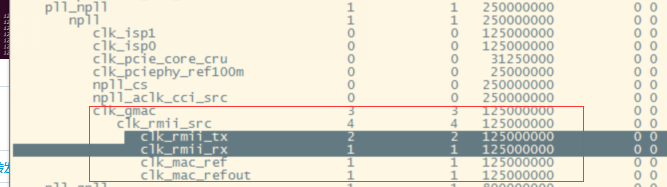




#### 问题点2

量不到125M clock, 请确认主控 clock tree 有设置起来,正常如下

Cat /d/clk/clk\_summary



正常的话会出来这样的波形：



#### 问题点3

千兆PHY有可能出现tx/rx 丢包问题，主要是调整主控或者switch delay, 保证时序在2ns以内可正常锁存数据，调整RGMII delay的方法如下：

CALL 如下的API，（txDelay（范围0～1）， rxDelay（范围0～7））

txDelay = 1， 调试。如果固定了，再调RX。

如果EXT port 没有RX到packet， rxDelay这边也就是从0～7一个一个的试。如果能收到packet就OK了。

那么初始化的时候就把这个API call一下。txDelay和rxDelay就是以上调出来的值。

API : rtk\_api\_ret\_t rtk\_port\_rgmiiDelayExt\_set(rtk\_port\_t port, rtk\_data\_t txDelay, rtk\_data\_t rxDelay), 目前已固定

也可以打上GMAC tx rx delay动态调整补丁V2.0 回环测试补丁（附件kernel4.4.zip 只做验证通路rx、tx data数据），通过命令设置switch为回环模式(register0 bit[14]置1，节点路径请通过find 命令查找，也可以通过后面调试的log方式获得。

busybox find –name rtl\_phy\_reg

1 、enable loopback

echo 0 > rtl\_phy\_reg

cat rtl\_phy\_regValue

echo xxxx > rtl\_phy\_regValue //bit[14]写1

2 、清空realtek mib 统计

echo 1 > rtl\_phy\_Rst

3 、执行一次测试

cat delay\_test

4 、查看realtek mib 与主控发包数对比

ifconfig eth0

cat 1 > rtl\_phy\_Mib

回环测试是RGMII，原理如下：



问题4 软件设定了一些抓取log的调试方法：



realtek mib 这边主要看下面统计：

EXT\_PORT0 是switch MAC端统计， UTP\_PORT0 代表phy 物理接口统计：

[ 5976.088116] port 0: ifInBroadcastPkts            [60]:                    7906358

[ 5976.150769] port 16: ifOutBroadcastPkts       [32]:                    7906358

ifInBroadcastPkts   代表输入广播包多少笔，ifOutBroadcastPkts   代表输出广播包多少笔

另外rtl还有很多相关接口，具体使用方法还是需要咨询rtl switch厂商获得更多相关信息，RK 这边只负责配合跑通芯片，SWITCH API由rtl提供技术支持。

附件中：

Patch\_rk3399\_switch.patch ----针对rk3399 android8.1 搭配rtl8367rb switch 底层相关修改（必须参考打上）

rtl8367c.tgz ----针对rtl8367rb switch driver 这个是rtl提供的driver并做了修改调整（必须打上）

Realtek\_Unmanaged\_Switch\_API\_Document.pdf --- rtl8367rb switch 厂商提供的相关文档

Realtek\_Unmanaged\_Switch\_ProgrammingGuide.pdf --- rtl8367rb switch 厂商提供的相关文档

Realtek\_Unmanaged\_Switch\_ReleaseNote.pdf --- rtl8367rb switch 厂商提供的相关文档

RTL8367RB-CG\_DataSheet\_1.2.pdf --- rtl8367rb switch 厂商提供的datasheet

Kenrel4.4.zip ------ RK 提供的用来验证回环测试的方法补丁包（正式版本无需打上，只做验证需要）

stmmac\_mdio.c --- 可以用来确认switch MDC/MDIO 读写接口。（只需了解，无需更新。）