[分享] Xilinx MPSoC以太网调试思路

发布于 2020-07-16 16:53:06 1.9K 0 举报

在嵌入式系统里，以太网是一个基本的接口，既用于调试，也用于数据传输。所以在单板调试过程中，以太网是一个基本的任务。如果以太网工作正常，也可以说是一个重要的里程碑。
Xilinx MPSoC支持多个网卡，应用成熟，下面是常见的调试思路。

1. 以太网硬件

以太网的硬件，分为两块，第一是MAC，第二是PHY。当然，在调试以太网以前，CPU和DDR、相关总线都要工作正常。MAC和PHY之间，有两个接口，第一是数据接口，可能是MII、GMII、RGMII、SGMII等；第二是管理接口，MDIO总线。数据接口用于传输数据。对RGMII而言，如果以太网工作在1000M，频率是125MHz；如果以太网工作在100M，频率是25MHz。MDIO是类似IIC的总线，MAC提供时钟MDC，数据线MDIO是双向的，既可以读PHY的寄存器，也可以写PHY的寄存器。
!Xilinx MPSoC RGMII信号](https://upload-images.jianshu.io/upload_images/22911878-ecedcfc816a786bd.png?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)

2. 软件

2.1. Standalone代码

Xilinx在Xilinx\SDK\2018.3\data\embeddedsw\XilinxProcessorIPLib\drivers\emacps_v3_8\examples提供了测试代码xemacps_example_intr_dma.c。2018.3是版本号，请根据自己情况更改。xemacps_example_intr_dma.c的功能是初始化MAC和PHY，设置PHY为自环，发送一个包，再接收一个包，最后检查数据是否正确。如果代码不能退出，可能是发送失败，或者没有收到包。代码缺省配置MAC和PHY为1000M。

2.2. UBoot代码

UBoot下，MAC的驱动代码是drivers\net目录下的macb.c。
PHY的驱动代码是drivers\net\phy目录下的phy.c，以及厂家相关代码，比如ti.c。

2.3. Linux代码

Linux下，MAC的驱动代码是drivers\net\ethernet\cadence目录下的macb_main.c、macb_ptp.c。
PHY的驱动代码是drivers\net\phy目录下的phy.c，以及厂家相关代码，比如dp83867.c。

2.4. 设备树

UBoot/Linux的驱动代码需要设备树提供一些参数。其中一个必须的参数是PHY的地址。下面代码是U-Boot 2019.1里zynqmp-zcu102-revB.dts文件里关于Phy的设置。它指定了PHY的地址0xc。其它参数是Phy的参数，设置原因请参考PHY手册。

```
&gem3 {
    phy-handle = <&phyc>;
    phyc: phy@c {
        reg = <0xc>;
        ti,rx-internal-delay = <0x8>;
        ti,tx-internal-delay = <0xa>;
        ti,fifo-depth = <0x1>;
```

```
                ti,dp83867-rxctrl-strap-quirk;
                /* reset-gpios = <&tca6416_u97 6 GPIO_ACTIVE_LOW>; */
            };
            /* Cleanup from RevA */
            /delete-node/ phy@21;
        };
```

3. 调试流程

调试时，先确保硬件工作正常。需要软件配合时，Standalone代码最简单，因此最好修改Standalone代码来配合硬件调试。

3.1. 检查MDIO

让软件发起PHY寄存器的读写操作，检查MDC/MDIO是否有跳变及其信号质量。

3.2. 检查PHY

让软件读PHY的ID等寄存器，对照手册，看寄存器值是否正确。如果不对，可能是PHY的地址错误。也可以从0到31尝试PHY的地址，读取PHY的ID。读到正确的ID，就说明PHY的地址对了。

3.3. 测试自协商

连接单板和电脑，电脑分别配置成自协商、1000M、100M、10M。让软件读PHY寄存器的自协商结果寄存器，检查单板侧PHY自协商的结果。

3.4. RGMII时钟

对RGMII而言，TX_Clk是MAC发出的，RX_Clk是PHY发出的。如果没有时钟，MAC自环也会有问题。需要确保TX_Clk、RX_Clk正常。

3.5. MAC自环测试

network_control寄存器的bit 1是loopback_local，用于使能本地自环。在测试数据收发之前，可以先设置loopback_local，再进行发送、接收测试。
在Standalone代码下,调用函数EmacPsUtilEnterLocalLoopback()可以完成这个操作。
建议创建一个Standalone的工程，把xemacps_example_intr_dma.c复制到工程中。再在xemacps_example_intr_dma.c中调用EmacPsUtilEnterLocalLoopback()设置本地自环，完成MAC自环测试。

3.6. PHY自环测试

在Standalone代码下,函数EmacPsUtilEnterLoopback()会设置PHY为自环模式。它调用XEmacPsDetectPHY()检测PHY的地址，在读取PHY的ID，并根据ID判断PHY的厂家是Marvell或者Ti，再调用对应的函数设置PHY为自环模式。对于ZCU102单板，会调用EmacPsUtilTiPhyLoopback()。
建议创建一个Standalone的工程，把xemacps_example_intr_dma.c复制到工程中。如果PHY和Xilinx开发板型号一样，xemacps_example_intr_dma.c能够正常完成PHY自环测试。

3.7. 检查MAC统计计数寄存器

MAC提供了寄存器统计MAC的运行状态。其中最重要的两个寄存器是frames_txed_ok、frames_rxed_ok。frames_txed_ok的偏移是0x0108，表示成功发送的帧的数目。frames_rxed_ok的偏移是0x0158，表示成功接收的

帧的数目。
在调试时，检查者两个寄存器，可以检查MAC的发送和接收的状态。

3.8. UBoot测试

在MAC自环和PHY自环测试成功后，可以在UBoot测试以太网，比如使用简单的ping命令。可以把对端的电脑，分别设置成1000M、100M、10M的情况，检查是否能够成功ping对端的电脑。下面是一个ping命令的例子。

```
uboot> setenv ipaddr 192.168.0.111
uboot> setenv serverip 192.168.0.100
uboot> ping 192.168.0.100
host 192.168.0.100 is alive
```

3.9. Linux测试

在MAC自环和PHY自环测试成功后，可以在Linux测试以太网，比如可以检查Linux启动后，能否通过DHCP得到IP地址，能否成功ping其它主机。同UBoot一样，也可以把对端的电脑，分别设置成1000M、100M、10M的情况，检查是否能够成功ping对端的电脑。