ZYNQ UltraScale uboot SGMII网口调试 TFTP

原创 Kuens 于 2019-12-31 15:17:41 发布 2275 收藏 16 版权

分类专栏： ZYNQ开发 文章标签： linux 嵌入式

ZYNQ UltraScale uboot SGMII网口调试

1. 软硬件平台

ZYNQ Ultrascale MPSOC XCZU19EGFFVC1760

Petalinux 2019.1

design_1_wrapper_hw_platform_1 Hardware Platform Specification

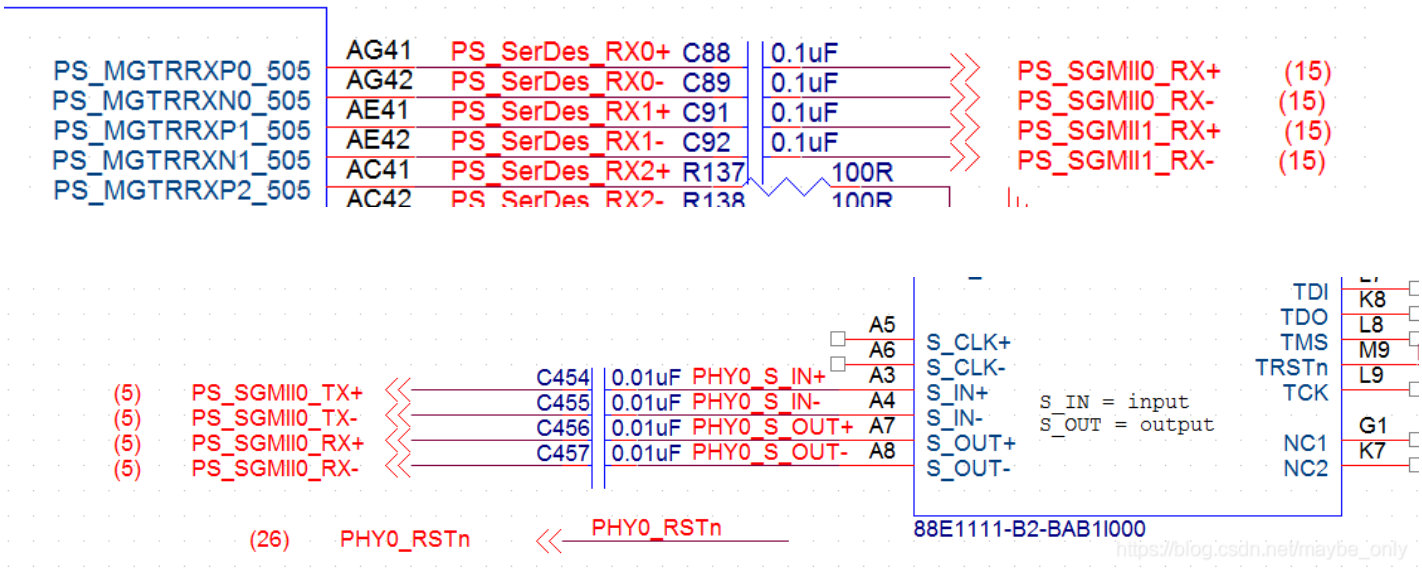
Design Information

Target FPGA Device: xczu19eg
Part: xczu19eg-ffvc1760-2-i
Created With: Vivado 2019.1
Created On: Tue Sep 17 10:49:42 2019

Address Map for processor psu_cortexa53_[0-3]

```
1 我使用petalinux2019.1 在uboot阶段网口不通，进入内核可以正常通信。
```

2. 硬件电路设计



3. uboot 调试笔记

3.1 uboot 设备树 及其本地源码路径配置

```
1 该平台默认使用的设备树为uboot
2 zynqmp-zcu102.rev1.0.dts
3 zynqmp-zcu102.revA.dts
4 zynqmp-zcu102.revB.dts
```

源码位于：

/opt/pkg/source_code/u-boot-xlnx-master/arch/arm/dts/zynqmp-zcu102-rev1.0.dts

在Petalinux下uboot源码是很难找到. 可去xilinx官方下载相关源码

Xilinx源码官方git:

链接: link.

petalinux-config

```
-*- ZYNQMP Configuration
    Linux Components Selection  --->
    Auto Config Settings  --->
-*- Subsystem AUTO Hardware Settings  --->
    DTG Settings  --->
    ARM Trusted Firmware Compilation Configuration  --->
[ ] Power Management kernel configuration
    FPGA Manager  --->
    u-boot Configuration  --->
    Image Packaging Configuration  --->
    Firmware Version Configuration  --->
    Yocto Settings  --->
```



```
        [*] First Stage Bootloader
        [*]   Auto update ps_init
        [*] PMU Firmware
            u-boot (ext-local-src)  --->
            External u-boot local source settings  --->
            arm-trusted-firmware (arm-trusted-firmware)  --->
            linux-kernel (ext-local-src)  --->
            External linux-kernel local source settings  --->
```


![在这里插入图片描述](https://img-blog.csdnimg.cn/20191231135833497.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L21heWJlX29ubHk=,size_16,color_FFFFFF,t_70

3.2 uboot网卡源码分析

源码路径:

/opt/pkg/source_code/u-boot-xlnx-master/drivers/net/zynq_gem.c

系统启动后FSBL->Uboot

通过zynq_gem_ofdata_to_platdata()获取设备树相关节点信息:

例如: phy-handle

reg

max-speed

phy-mode

####特别说明#### :

初次调试使用的是修改uboot源码下的设备树文件,配置gem0为默认的网络调试端口:

如下:

file: zynqmp-zcu102.revA.dts

```
aliases {
        ethernet0 = &gem0;      //kuens
                gpio0 = &gpio;
                i2c0 = &i2c0;
                i2c1 = &i2c1;
                mmc0 = &sdhci1;
                rtc0 = &rtc;
                serial0 = &uart0;
                serial1 = &uart1;
                serial2 = &dcc;
                spi0 = &qspi;
                //usb0 = &usb0;
};
```

```
        file:zynqmp-zcu102.revB.dts
```

```
//kuens
&gem0{
        local-mac-address = [00 40 85 44 00 E0];
        status = "okay";
        //is-internal-pcspma = <0x1>;
    phy-handle = <&phy3>;
        phy-mode = "sgmii";
        //pinctrl-names = "default";

        mdio {
                #address-cells  = <0x1>;
                #size-cells     = <0x0>;
                phy3: phy@3 {
                        compatible = "marvell,88e1111";
                        device_type = "ethernet-phy";
                        reg = <3>;
                        //linux,phandle = <0x4>;
                        //phandle = <0x4>;
                };
        };
};
```

如上图配置并无异常，但是在uboot阶段通信异常，显示A R P错误。

3.3 uboot阶段网卡info查询

```
ZynqMP> mii info
PHY 0x01: OUI = 0x5043, Model = 0x0C, Rev = 0x02, 1000baseX, HDX
PHY 0x02: OUI = 0x5043, Model = 0x0C, Rev = 0x02, 1000baseX, HDX
PHY 0x03: OUI = 0x5043, Model = 0x0C, Rev = 0x02, 1000baseX, FDX
PHY 0x04: OUI = 0x5043, Model = 0x0C, Rev = 0x02, 1000baseX, HDX
ZynqMP>
```

```
uboot系统使用phy 3， mii info查询接口信息为：1000baseX,FDX。接口配置为sgmii  to copper。
应该显示为：1000baseT, FDX.（这个问题困扰许久）
```

查询datasheet

When the copper interface is running in 1000BASE-T mode, the serial 1.25 GHz SGMII encoding is identical to that found in 1000BASE-X.

In 100BASE-TX and 10BASE-T modes, the SGMII interface still runs at 1.25 GHz using 1000BASE-X encoding. However, each byte of data in the packet is repeated 10 or 100 times, respectively. The synchronizing FIFOs are automatically enabled in these modes for both the transmit and receive paths.

The SGMII interface implements a modified 1000BASE-X Auto-Negotiation to indicate link, duplex, and speed to the MAC. The result of the Auto-Negotiation exchange on the copper side is encoded onto the serial interface via the modified Auto-Negotiation so that multi-port devices can adjust to the correct operating speed.

Figure 16 is an example of an 88E1111 device using the SGMII interface.

3.1 网卡初始化过程

```
        zynq_gem_probe(struct udevice *dev)函数:
         //分配tx_bd rx_bd的空间
      priv->tx_bd = (struct emac_bd *)bd_space;
         priv->rx_bd = (struct emac_bd *)((ulong)bd_space + BD_SEPRN_SPACE);
         //分配mdio  bus
     priv->bus = mdio_alloc();
     //初始化  读写函数
     priv->bus->read = zynq_gem_miiphy_read;
```

```
priv->bus->write = zynq_gem_miipny_write;
//注册
ret = mdio_register(priv->bus);
return zynq_phy_init(dev);

 zynq_phy_init(struct udevice *dev)函数:
          writel(ZYNQ_GEM_NWCTRL_MDEN_MASK, &regs->nwctrl);
          ret = phy_detection(dev);
          priv->phydev = phy_connect(priv->bus, priv->phyaddr, dev,priv->interface);
```

//查看定义

#define ZYNQ_GEM_NWCTRL_MDEN_MASK 0x00000010 /* Enable MDIO port */

man_port_en	4	rw	0x0	Management port enable - set to one to enable the management port. When zero forces mdio to high impedance state and mdc low.

主要初始化gem寄存器,然后通过mdio配置phy

zynq_gem_init() 函数分析:

```
/* Device registers */
                struct zynq_gem_regs {
                        u32 nwctrl; /* 0x0 - Network Control reg */
                        u32 nwcfg; /* 0x4 - Network Config reg */
                        u32 nwsr; /* 0x8 - Network Status reg */
                        u32 reserved1;
                        u32 dmacr; /* 0x10 - DMA Control reg */
                        u32 txsr; /* 0x14 - TX Status reg */
                        u32 rxqbase; /* 0x18 - RX Q Base address reg */
                        u32 txqbase; /* 0x1c - TX Q Base address reg */
                        u32 rxsr; /* 0x20 - RX Status reg */
                        u32 reserved2[2];
                        u32 idr; /* 0x2c - Interrupt Disable reg */
                        u32 reserved3;
                        u32 phymntnc; /* 0x34 - Phy Maintaince reg */
                        u32 reserved4[18];
                        u32 hashl; /* 0x80 - Hash Low address reg */
                        u32 hashh; /* 0x84 - Hash High address reg */
                #define LADDR_LOW       0
                #define LADDR_HIGH      1
                        u32 laddr[4][LADDR_HIGH + 1]; /* 0x8c - Specific1 addr low/high reg */
                        u32 match[4]; /* 0xa8 - Type ID1 Match reg */
                        u32 reserved6[18];
                #define STAT_SIZE       44
                        u32 stat[STAT_SIZE]; /* 0x100 - Octects transmitted Low reg */
                        u32 reserved9[20];
                        u32 pcscntrl;
                        u32 rserved12[36];
                        u32 dcfg6; /* 0x294 Design config reg6 */
                        u32 reserved7[106];
                        u32 transmit_q1_ptr; /* 0x440 - Transmit priority queue 1 */
                        u32 reserved8[15];
                        u32 receive_q1_ptr; /* 0x480 - Receive priority queue 1 */
                        u32 reserved10[17];
                        u32 upper_txqbase; /* 0x4C8 - Upper tx_q base addr */
                        u32 reserved11[2];
                        u32 upper_rxqbase; /* 0x4D4 - Upper rx_q base addr */
                };
```

```
        该结构体是gem寄存器相关定义。
```

注意部分:

#define ZYNQ_GEM_NWCFG_INIT (ZYNQ_GEM_DBUS_WIDTH |

ZYNQ_GEM_NWCFG_FDEN |
ZYNQ_GEM_NWCFG_FSREM |
ZYNQ_GEM_NWCFG_MDCCLKDIV)
nwconfig = ZYNQ_GEM_NWCFG_INIT;

```
                /*
                 * Set SGMII enable PCS selection only if internal PCS/PMA
                 * core is used and interface is SGMII.
                 */
                if (priv->interface == PHY_INTERFACE_MODE_SGMII &&
                    priv->int_pcs) {
                        nwconfig |= ZYNQ_GEM_NWCFG_SGMII_ENBL |
                                    ZYNQ_GEM_NWCFG_PCS_SEL;
        #ifdef CONFIG_ARM64
                        writel(readl(&regs->pcscntrl) | ZYNQ_GEM_PCS_CTL_ANEG_ENBL,
                               &regs->pcscntrl);
        #endif
                }
```

查看源码可知:
使能SGMII模式，priv->int_pcs 必须为１，在上述源码手动修改：
priv->int_pcs = 1;
系统依然通信异常。
解决办法：
１.在zynq_gem_ofdata_to_platdata() 函数中，修改下面代码:
priv->int_pcs = dev_read_bool(dev, “is-internal-pcspma”);
２.在设备树中添加手动指定is-internal-pcspma节点值。

gem中使能sgmii模式的寄存器如下：

Field Name	Bits	Type	Reset Value	Description
uni_direction_enable	31	rw	0x0	Uni-direction-enable. When low the PCS will transmit idle symbols if the link goes down. When high the PCS can transmit frame data when the link is down.
ignore_ipg_rx_er	30	rw	0x0	Ignore IPG rx_er. When set rx_er has no effect on the GEMs operation when rx_dv is low. Set this when using the RGMII wrapper in half-duplex mode.
nsp_change	29	rw	0x0	Receive bad preamble. When set frames with non-standard preamble are not rejected.
ipg_stretch_enable	28	rw	0x0	IPG stretch enable - when set the transmit IPG can be increased above 96 bit times depending on the previous frame length using the IPG stretch register.
sgmii_mode_enable	27	rw	0x0	SGMII mode enable - changes behaviour of the auto-negotiation advertisement and link partner ability registers to meet the requirements of SGMII and reduces the duration of the link timer from 10 ms to 1.6 ms.
ignore_rx_fcs	26	rw	0x0	Ignore RX FCS - when set frames with FCS/CRC errors will not be rejected. FCS error statistics will still be collected for frames with bad FCS and FCS status will be recorded in frame's DMA descriptor. For normal operation this bit must be set to zero.
en_half_duplex_rx	25	rw	0x0	Enable frames to be received in half-duplex mode while transmitting.
receive_checksum_offload_enable	24	rw	0x0	Receive checksum offload enable - when set, the receive checksum engine is enabled. Frames with bad IP, TCP or UDP checksums are discarded.
disable_copy_of_pause_frames	23	rw	0x0	Disable copy of pause frames - set to one to prevent valid pause frames being copied to memory. When set, pause frames are not copied to memory regardless of the state of the copy all frames bit; whether a hash match is found or whether a type ID match is identified. If a destination address match is found the pause frame will be copied to memory. Note that valid pause frames received will still increment pause statistics and pause the transmission of frames as required.
data_bus_width	22:21	rw	0x1	Data bus width - set according to AMBA AXI or external FIFO data bus width

根据寄存器定义，寄存器0xFF0B0004 bit27 为１
启动uboot，使用md命令读取寄存器参数：
md指令用法:
uboot下输入指令md,会提示md的用法，memory display，即内存显示。
U-Boot-PetaLinux> md
md - memory display
Usage:
md [.b, .w, .l] address [# of objects]
b:8位
w:16位
l:32位（默认值）
例如：

```
ZynqMP> <INTERRUPT>
ZynqMP>
ZynqMP>
ZynqMP>
ZynqMP>
ZynqMP>
ZynqMP>
ZynqMP>
ZynqMP> md.l 0xff0b0004 1
ff0b0004: 08320c02                              ..2.
ZynqMP>
ZynqMP> ping 192.168.1.77
Using ethernet@ff0b0000 device
host 192.168.1.77 is alive
ZynqMP>
```

设置服务器ip:

ZynqMP> print serverip

ZynqMP> set serverip ; saveenv

启动:

ZynqMP> run netboot

```
ZynqMP> run netboot
Using ethernet@ff0b0000 device
TFTP from server 192.168.1.92; our IP address is 192.168.1.80
Filename 'image.ub'.
Load address: 0x10000000
Loading: ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################
         6.9 MiB/s
done
```


总结：设备树修改建议不要修改源码路径下的设备树。包括：uboot与内核下的设备树，建议修改system-user.dtsi

参考:

ug1144.pdf

https://www.xilinx.com/html_docs/registers/ug1087/ug1087-zynq-ultrascale-registers.html