## **Part 1: Document introduction**

This article describes the AX7021 FPGA development board, Multiple Ethernet for the FL9031 Ethernet Module in the SDK test IwIP Echo Server function, has petalinux 2017.4 driver configuration, device tree configuration, and simple application

This example requires Vivado 2017.4, Petalinux 2017.4, other versions may have unpredictable problems, please solve it yourself.

## Part 2: VIVADO hardware Project Establishment

How to use VIVADO to build a project is not the focus of this article. ALINX provides the already completed vivado project.

#### 2.1: Export SDK



Figure 2-1: Export SDK

## Part 3: IwIP Echo Server test

This section describes how to test the Ethernet on the PL and PS side

under sdk and build a simple test using the SDK's own lwIP Echo Server template.

#### 3.1 LWIP library modification

Since the built-in LWIP library can only identify part of the phy chip, if the phy chip used by the development board is not within the default support, modify the library file. You can also replace the original library directly with the modified library.

1) Find the library file directory

"X:\Xilinx\SDK\2017.4\data\embeddedsw\ThirdParty\sw\_services"

本地磁盘 (C:) ▶ Xilinx ▶ SDK →	2017.4 I data I embedde	edsw ♦ ThirdPa	rty • sw_services
共享 🔻 新建文件夹			
名称	修改日期	类型	大小
🍌 libmetal_v1_0	2018/1/15 15:41	文件夹	
📕 libmetal_v1_1	2018/1/15 15:43	文件夹	
📙 libmetal_v1_2	2018/1/15 15:41	文件夹	
🍶 libmetal_v1_3	2018/1/15 15:41	文件夹	
🔒 lwip141_v1_0	2018/1/15 15:41	文件夹	
iwip141_v1_6	2018/1/15 15:41	文件夹	
퉬 lwip141_v1_7	2018/1/15 15:43	文件夹	
퉬 lwip141_v1_8	2018/1/15 15:41	文件夹	
🍌 lwip141_v1_9	2018/1/15 <mark>1</mark> 5:43	文件夹	
📕 lwip141_v2_0	2018/1/15 15:41	文件夹	
🍌 openamp_v1_0	2018/1/15 15:43	文件夹	
🍌 openamp_v1_1	2018/1/15 15:43	文件夹	
🍌 openamp_v1_2	2018/1/15 15:41	文件夹	
🍌 openamp_v1_3	2018/1/15 15:41	文件夹	
🍌 openamp_v1_4	2018/1/15 15:40	文件夹	

Figure 3-1: Find the library file directory

2) Locate the files "xaxiemacif\_physpeed.c" and "xemacpsif\_physpeed.c" in the file directory "lwip141\_v2\_0\src\contrib\ports\xilinx\netif" to be modified. The v2\_0 version is modified here. If it is another version of Vivado, the version of the lwip library will also change.

#### Generally, the latest version is selected to be modified.

2017.4 • data • embeddedsw •	ThirdParty • sw_services •	• lwip141_v2_0	c Contrib	ports 🕨 xilinx 🕨 neti
共享 ▼ 新建文件夹				
名称 ^	修改日期	类型	大小	
c xadapter.c	2017/12/14 9:27	C Source file	7 KB	
xaxiemacif.c	2017/12/14 9:27	C Source file	15 KB	
xaxiemacif_dma.c	2017/12/14 9:27	C Source file	30 KB	
• xaxiemacif_fifo.c	2017/12/14 9:27	C Source file	12 KB	
h xaxiemacif_fifo.h	2017/12/14 9:27	C++ Header file	1 KB	
• xaxiemacif_hw.c	2017/12/14 9:27	C Source file	5 KB	
xaxiemacif_hw.h	2017/12/14 9:27	C++ Header file	2 KB	
xaxiemacif_physpeed.c	2017/12/14 9:27	C Source file	28 KB	
🖻 xemacliteif.c	2017/12/14 9:27	C Source file	24 KB	
xemacpsif.c	2017/12/14 9:27	C Source file	13 KB	
xemacpsif_dma.c	2017/12/14 9:27	C Source file	27 KB	
xemacpsif_hw.c	2017/12/14 9:27	C Source file	8 KB	
xemacpsif_hw.h	2017/12/14 9:27	C++ Header file	2 KB	
c xemacpsif_physpeed.c	2017/12/14 9:27	C Source file	34 KB	
c xpqueue.c	2017/12/14 9:27	C Source file	3 KB	

Figure 3-2: Find the file

3) Modify the "xaxiemacif\_physpeed.c" file to add related macro definitions

#define	IEEE_MMD_ACCESS_CTRL_DEVAD_MASK	Ox1F
#define	IEEE MMD ACCESS CTRL PIDEVAD MASK	Ox801F
#define	IEEE_MMD_ACCESS_CTRL_NOPIDEVAD_MASK	0x401F
#define	PHY_R0_ISOLATE	0x0400
#define	PHY DETECT REG	1
#define	PHY IDENTIFIER L REG	2
#define	PHY IDENTIFIER 2 REG	3
#define	PHY DETECT MASK	0x1808
#define	PHY MARVELL IDENTIFIER	Ox0141
#define	PHY_TI_IDENTIFIER	0x2000
/* Marv	el PHY flags */	
#define	MARVEL_PHY_IDENTIFIER	0x141
#define	MARVEL PHY MODEL NUM MASK	Ox3F0
#define	MARVEL PHY 88E1111 MODEL	OxC0
#define	MARVEL PHY 88E	Ox240
#define	PHY_88E1111_RGMII_RX_CLOCK_DELAYED_MASK	0x0080
/* TI P	HY Flags */	
#define	TI PHY DETECT MASK	0x796I
#define	TI PHY IDENTIF	0x2000
#define	TI PHY DP83867 MODEL	0xA231
#define	DP83867 RGMII CLOCK DELAY CTRL MASK	0x0003
#define	DP83867 RGMII TX CLOCK DELAY MASK	0x0030
#define	DP83867_RGMII_RX_CLOCK_DELAY_MASK	0x0003
/* TI D	P83867 PHY Registers */	
#define	DP83867_R32_RGMIICTL1	0x32
#define	DP83867_R86_RGMIIDCTL	0x86
#define	MICREL_PHY_IDENTIFIER	0x22
#define	MICREL PHY KSZ9031 MODEL	0x220
#define	TI PHY REGCR 0xD	
#define	TI PHY ADDDR 0xE	
tdofino	TT DEV DEVCTOI 0110	

Figure 3-3: Add Related Macro Definitions

#### 4) Add phy get function

```
unsigned int get_phy_speed_ksz9031(XAxiEthernet *xaxiemacp, u32 phy_addr)
{
       ul6 control;
       ul6 status;
       ul6 partner_capabilities;
       xil_printf("Start PHY autonegotiation \r\n");
       XAxiEthernet_PhyWrite(xaxiemacp,phy_addr, IEEE_PAGE_ADDRESS_REGISTER, 2);
       XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_CONTROL_REG_MAC, &control);
       //control |= IEEE_RGMII_TXRX_CLOCK_DELAYED_MASK;
       control &= ~(0x10);
       XAxiEthernet_PhyWrite(xaxiemacp, phy_addr, IEEE_CONTROL_REG_MAC, control);
       XAxiEthernet_PhyWrite(xaxiemacp, phy_addr, IEEE_PAGE_ADDRESS_REGISTER, 0);
       XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_AUTONEGO_ADVERTISE_REG, &control);
       control |= IEEE_ASYMMETRIC_PAUSE_MASK;
       control |= IEEE_PAUSE_MASK;
       control |= ADVERTISE_100;
       control |= ADVERTISE_10;
       XAxiEthernet_PhyWrite(xaxiemacp, phy_addr, IEEE_AUTONEGO_ADVERTISE_REG, control);
       XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_1000_ADVERTISE_REG_OFFSET,
                                               &control);
       control |= ADVERTISE_1000;
       XAxiEthernet_PhyWrite(xaxiemacp, phy_addr, IEEE_1000_ADVERTISE_REG_OFFSET,
                                               control);
       XAxiEthernet_PhyWrite(xaxiemacp, phy_addr, IEEE_PAGE_ADDRESS_REGISTER, 0);
       XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_COPPER_SPECIFIC_CONTROL_REG,
```



```
&control);
control |= (7 << 12); /* max number of gigabit attempts */</pre>
XAxiEthernet_PhyWrite(xaxiemacp, phy_addr, IEEE_COPPER_SPECIFIC_CONTROL_REG,
                               control);
XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_CONTROL_REG_OFFSET, &control);
control |= IEEE_CTRL_AUTONEGOTIATE_ENABLE;
control |= IEEE_STAT_AUTONEGOTIATE_RESTART;
XAxiEthernet_PhyWrite(xaxiemacp, phy_addr, IEEE_CONTROL_REG_OFFSET, control);
XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_CONTROL_REG_OFFSET, &control);
control |= IEEE_CTRL_RESET_MASK;
XAxiEthernet PhyWrite(xaxiemacp, phy addr, IEEE CONTROL REG OFFSET, control);
while (1) {
       XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_CONTROL_REG_OFFSET, &control);
       if (control & IEEE_CTRL_RESET_MASK)
               continue:
       else
               break;
}
xil_printf("Waiting for PHY to complete autonegotiation.\r\n");
XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_STATUS_REG_OFFSET, &status);
while ( !(status & IEEE_STAT_AUTONEGOTIATE_COMPLETE) ) {
       sleep(1);
       XAxiEthernet_PhyRead(xaxiemacp, phy_addr, IEEE_STATUS_REG_OFFSET,
                              &status);
        }
xil_printf("autonegotiation complete \r\n");
```

```
XAxiEthernet_PhyRead(xaxiemacp, phy_addr, 0x1f, &partner_capabilities);

if ( (partner_capabilities & 0x40) == 0x40)/* 1000Mbps */

    return 1000;

else if ( (partner_capabilities & 0x20) == 0x20)/* 100Mbps */

    return 100;

else if ( (partner_capabilities & 0x10) == 0x10)/* 10Mbps */

    return 10;

else

    return 0;

}
```

5) Modify the function "get\_IEEE\_phy\_speed" to add support for KSZ9031

```
unsigned get_IEEE_phy_speed(XAxiEthernet *xaxiemacp)
{
    ul6 phy_identifier;
    ul6 phy_model;
    u8 phytype;
#ifdef XPAR_AXIETHERNET_0_BASEADDR
    u32 phy_addr = detect_phy(xaxiemacp);
    /* Get the PHY Identifier and Model number */
    XAxiEthernet_PhyRead(xaxiemacp, phy_addr, PHY_IDENTIFIER_1_REG, &phy_identifier);
    XAxiEthernet_PhyRead(xaxiemacp, phy_addr, PHY_IDENTIFIER_2_REG, &phy_model);
/* Depending upon what manufacturer PHY is connected, a different mask is
 * needed to determine the specific model number of the PHY. */
    if (phy_identifier == MARVEL_PHY_IDENTIFIER) {
        phy_model = phy_model & MARVEL_PHY_MODEL_NUM_MASK;
    }
}
```

if (phy_model == MARVEL_PHY_88E1116R_MODEL) {
<pre>return get_phy_speed_88E1116R(xaxiemacp, phy_addr);</pre>
<pre>} else if (phy_model == MARVEL_PHY_88E1111_MODEL)</pre>
<pre>{ return get_phy_speed_88E1111(xaxiemacp,</pre>
phy_addr);
3
<pre>} else if (phy_identifier == TI_PHY_IDENTIFIER) {</pre>
<pre>phy_model = phy_model &amp; TI_PHY_DP83867_MODEL;</pre>
<pre>phytype = XAxiEthernet_GetPhysicalInterface(xaxiemacp);</pre>
if (phy_model == TI_PHY_DP83867_MODEL && phytype == XAE_PHY_TYPE_SGMII)
<pre>{ return get_phy_speed_TI_DP83867_SGMII(xaxiemacp, phy_addr);</pre>
}
<pre>if (phy_model == TI_PHY_DP83867_MODEL) {</pre>
<pre>return get_phy_speed_TI_DP83867(xaxiemacp, phy_addr);</pre>
}
}
<pre>else if(phy_identifier == MICREL_PHY_IDENTIFIER)</pre>
ł
<pre>xil_printf("Phy %d is KSZ9031\n\r",phy_addr);</pre>
<pre>get_phy_speed_ksz9031(xaxiemacp, phy_addr);</pre>
}
else {
LWIP_DEBUGF (NETIF_DEBUG, ("XAxiEthernet get_IEEE_phy_speed: Detected PHY with unknown identifier/model.\r\n"));
3
#endif
#ifdef PCM_PMA_CORE_PRESENT
<pre>return get_phy_negotiated_speed(xaxiemacp, phy_addr);</pre>
#endif
}

6) Modify the "xemacpsif\_physpeed.c" file to add a macro definition

#define IEEE_CTRL_1GBPS_LINKSPEED_MASK       0x2040         #define IEEE_CTRL_LINKSPEED_MASK       0x0040         #define IEEE_CTRL_LINKSPEED_100M       0x2000         #define IEEE_CTRL_LINKSPEED_100M       0x2000         #define IEEE_CTRL_LINKSPEED_10M       0x2000         #define IEEE_CTRL_LINKSPEED_10M       0x0000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_STAT_AUTONEGOTIATE_CMPLETE       0x0000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0000         #define IEEE_RGMIT_TXRX_CLOCK_DELAYED_MASK       0x0800         #define IEEE_RGMIT_TXRX_CLOCK_DELAYED_MASK       0x0800         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x100         #define PHY_DETECT_MASK       0x1011         #define PHY_DETECT_MASK       0x2000         #define PHY_XILINX_PCS_PMA_ID1	AGETTHE TEFE LAGE ADDRESS KEGISTER	22
#define IEEE_CTRL_LINKSPEED_MASK       0x0040         #define IEEE_CTRL_LINKSPEED_1000M       0x0040         #define IEEE_CTRL_LINKSPEED_100M       0x0000         #define IEEE_CTRL_LINKSPEED_10M       0x0000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_MASK       0x0000         #define IEEE_SPEED_1000       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_SPEED_100       0x4000         #define IEEE_SPEED_100       0x4000         #define IEEE_SPEED_TAT_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_RGMIT_TXRX_CLOCK_DELAYED_MASK       0x00030         #define IEEE_PAUSE_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x0400         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x18000         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_MARVELI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174	#define IEEE CTRL 1GBPS LINKSPEED MASH	0x2040
#define IEEE_CTRL_LINKSPEED_100M       0x0040         #define IEEE_CTRL_LINKSPEED_10M       0x2000         #define IEEE_CTRL_LINKSPEED_10M       0x0000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_1000       0x8000         #define IEEE_SPEED_1000       0x4000         #define IEEE_SPEED_100       0x4000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_SPEED_100       0x4000         #define IEEE_SPEED_100       0x4000         #define IEEE_SPEED_100       0x4000         #define IEEE_STAT_AUTONEGOTIATE_CMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_ASYMMETRIC_PAUSE_MASK       0x0800         #define IEEE_AUTONEG_ERROR_MASK       0x08000         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x1800         #define PHY_DENTIFIER_R_REG       3         #define PHY_DENTIFIER_1 REG       2         #define PHY_DENTIFIER_1 REG       0x2000         #define PHY_MARVELL IDENTIFIER       0x2000	#define IEEE CTRL LINKSPEED MASK	0x0040
#define IEEE_CTRL_LINKSPEED_10M       0x2000         #define IEEE_CTRL_LINKSPEED_10M       0x0000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_1000       0x8000         #define IEEE_SPEED_1000       0x4000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_CTRL_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0000         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_ASYMMETRIC_PAUSE_MASK       0x08000         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x18000         #define PHY_DENTIFIER 0x101       0x0141         #define PHY_DETECT_MASK       0x1800         #define PHY_DENTIFIER 0x2000       #define PHY_NARVELL IDENTIFIER 0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0141         #define PHY_XILINX_PCS_PMA_ID2       0x2200         #define MICREL PHY_IDENTIFIER 0x2000       0x2200 <td>#define IEEE CTRL LINKSPEED 1000M</td> <td>0x0040</td>	#define IEEE CTRL LINKSPEED 1000M	0x0040
#define IEEE_CTRL_LINKSPEED_10M       0x0000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_MASK       0x0000         #define IEEE_SPEED_1000       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_TRL_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_STAT_AUTONEGERROR_MASK       0x08000         #define IEEE_PAUSE_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_IDENTIFIER_1 REG       2         #define PHY_IDENTIFIER_2 REG       3         #define PHY_IDENTIFIER 0x2000       #define PHY_MARVELL_IDENTIFIER       0x0141         #define PHY_XILINX_PCS_PMA_ID1       0x0174       #define PHY_XILINX_PCS_PMA_ID2       0x22000         #define MICREL_PHY_IDENTIFIER       0x220       0x22000       #define MICREL_PHY_KS29031 MODEL       0x22000         #define XEMACPS_GMIL2RGMII_SPEED1000_FD       0x140       0x22000	#define IEEE CTRL LINKSPEED 100M	0x2000
#define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_SPEED_MASK       0x0000         #define IEEE_SPEED_100       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_SPEED_100       0x4000         #define IEEE_SPEED_100       0x4000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_STAT_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_RGMIT_TXRX_CLOCK_DELAYED_MASK       0x0030         #define IEEE_ASYMMETRIC_PAUSE_MASK       0x08000         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x10141         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0200         #define MICREL_PHY_IDENTIFIER       0x22         #define MICREL_PHY_IDENTIFIER       0x22         #define XEMACPS_GMIL2RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED1000	#define IEEE CTRL LINKSPEED 10M	0x0000
#define IEEE_SPEED_MASK       0xC000         #define IEEE_SPEED_100       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_CTRL_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0030         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0030         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0030         #define IEEE_ASYMMETRIC_PAUSE_MASK       0x08000         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x18000         #define PHY_DETECT_MASK       0x1800         #define PHY_DENTIFIER_1 REG       2         #define PHY_DENTIFIER_2 REG       3         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_NERVEL_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x2200         #define MICREL PHY_KSZ90	#define IEEE_CTRL_RESET_MASK	0x8000
#define IEEE_SPEED_1000       0x8000         #define IEEE_SPEED_100       0x4000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_CTRL_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x00200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x00200         #define IEEE_ASYMMETRIC_PAUSE_MASK       0x0000         #define IEEE_AUTONEG_ERROR_MASK       0x0400         #define PHY_DETECT_REG       1         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x1808         #define PHY_TI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0141         #define PHY_XILINX_PCS_PMA_ID2       0x2200         #define MICREL PHY_IDENTIFIER       0x220         #define MICREL PHY_KS29031 MODEL	#define IEEE SPEED MASK 0xC000	
#define IEEE_SPEED_100       0x4000         #define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_CTRL_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_RGMIT_TXRX_CLOCK_DELAYED_MASK       0x0030         #define IEEE_AUTONEG_ERROR_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_DETECT_REG       1         #define PHY_IDENTIFIER_1 REG       2         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_TI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x2200         #define MICREL_PHY_SEGMIL_SPEED1000_FD       0x140         #define XEMAC	#define IEEE SPEED 1000 0x8000	
#define IEEE_CTRL_RESET_MASK       0x8000         #define IEEE_CTRL_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0200         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_ASYMMETRIC_PAUSE_MASK       0x0000         #define IEEE_AUTONEG_ERROR_MASK       0x0400         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x1808         #define PHY_IDENTIFIER_2REG       3         #define PHY_DETECT_MASK       0x1808         #define PHY_MARVELL_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0000         #define MICREL_PHY_IDENTIFIER       0x2220         #define MICREL_PHY_KSZ9031 MODEL       0x2200         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x2100         #define XEMACPS_GMIL2RGMII_SPEED100_FD	#define IEEE_SPEED_100 0x4000	
#define IEEE_CTRL_AUTONEGOTIATE_ENABLE       0x1000         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_RGMII_TXRX_CLOCK_DELAYED_MASK       0x00800         #define IEEE_AUTONEG_ERROR_MASK       0x04000         #define IEEE_AUTONEG_ERROR_MASK       0x04000         #define PHY_DETECT_REG       1         #define PHY_DETECT_REG       1         #define PHY_IDENTIFIER_1_REG       2         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_MARVELL_IDENTIFIER       0x2000         #define PHY_TI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID2       0x0000         #define MICREL PHY_IDENTIFIER       0x2220         #define MICREL PHY_KSZ9031 MODEL       0x2200         #define XEMACPS_GMIL2RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x100         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x	#define IEEE CTRL RESET MASK	0x8000
#define IEEE_STAT_AUTONEGOTIATE_COMPLETE       0x0020         #define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE_RGMII_TXRX_CLOCK_DELAYED_MASK       0x0030         #define IEEE_RGMII_TXRX_CLOCK_DELAYED_MASK       0x00400         #define IEEE_PAUSE_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_DETECT_REG       1         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_MARVELL_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0000         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_KSZ9031_MODEL       0x220         #define XEMACPS_GMIL2RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x2100         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x100         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x100	#define IEEE CTRL AUTONEGOTIATE ENABLE	0x1000
#define IEEE_STAT_AUTONEGOTIATE_RESTART       0x0200         #define IEEE RGMII_TXRX_CLOCK_DELAYED_MASK       0x0030         #define IEEE_ASYMMETRIC_PAUSE_MASK       0x0800         #define IEEE_AUTONEG_ERROR_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_DETECT_REG       1         #define PHY_IDENTIFIER 1_REG       2         #define PHY_DETECT_MASK       0x1808         #define PHY_MARVELL_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0141         #define PHY_XILINX_PCS_PMA_ID2       0x0200         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x220         #define XEMACPS_GMIL2RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x100         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x100	#define IEEE STAT AUTONEGOTIATE COMPLE	TE 0x0020
#define IEEE_RGMII_TXRX_CLOCK_DELAYED_MASK       0x0030         #define IEEE_ASYMMETRIC_PAUSE_MASK       0x0800         #define IEEE_PAUSE_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_IDENTIFIER_1_REG       2         #define PHY_IDENTIFIER_2_REG       3         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_TI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0000         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x220         #define XEMACPS_GMIL2RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x140	#define IEEE STAT AUTONEGOTIATE RESTAR	0x0200
#define IEEE_ASYMMETRIC_PAUSE_MASK       0x0800         #define IEEE_PAUSE_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_IDENTIFIER_1_REG       2         #define PHY_IDENTIFIER_2_REG       3         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x0141         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1000         #define PHY_MARVELL_IDENTIFIER       0x2000         #define MICREL_PHY_IDENTIFIER       0x2000         #define MICREL_PHY_IDENTIFIER       0x222         #define MICREL_PHY_KSZ9031 MODEL       0x2200         #define XEMACPS_GMIL2RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMIL2RGMII_SPEED100_FD       0x100         #define XEMACPS_GMIL2RGMII_SPEED10_FD       0x100         #define XEMACPS_GMIL2RGMIT_DEC_NIM       0x10	#define IEEE RGMII TXRX CLOCK DELAYED	MASK 0x0030
#define IEEE_PAUSE_MASK       0x0400         #define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_IDENTIFIER_1 REG       2         #define PHY_IDENTIFIER_2 REG       3         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x0141         #define PHY_MARVELL_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0000         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_KSZ9031 MODEL       0x220         #define XEMACPS_GMI12RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100	#define IEEE ASYMMETRIC PAUSE MASK	0x0800
#define IEEE_AUTONEG_ERROR_MASK       0x8000         #define PHY_DETECT_REG       1         #define PHY_IDENTIFIER_1_REG       2         #define PHY_IDENTIFIER_1_REG       3         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_DETECT_MASK       0x1808         #define PHY_MARVELL_IDENTIFIER       0x0041         #define PHY_TI_IDENTIFIER       0x02000         #define PHY_XILINX_PCS_PMA_ID2       0x0C00         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_KS29031_MODEL       0x220         #define XEMACPS_GMI12RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x140         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100	#define IEEE PAUSE MASK	0x0400
#define       PHY_DETECT_REG       1         #define       PHY_IDENTIFIER_1_REG       2         #define       PHY_IDENTIFIER_2_REG       3         #define       PHY_DETECT_MASK       0x1808         #define       PHY_DETECT_MASK       0x1808         #define       PHY_DETECT_MASK       0x0141         #define       PHY_MARVELL_IDENTIFIER       0x2000         #define       PHY_TI_IDENTIFIER       0x0174         #define       PHY_XILINX_PCS_PMA_ID1       0x0174         #define       PHY_XILINX_PCS_PMA_ID2       0x0000         #define       MICREL_PHY_IDENTIFIER       0x220         #define       MICREL_PHY_KSZ9031 MODEL       0x220         #define       XEMACPS_GMI12RGMII_SPEED1000_FD       0x140         #define       XEMACPS_GMI12RGMII_SPEED100_FD       0x100         #define       XEMACPS_GMI12RGMII_SPEED10_FD       0x100	#define IEEE_AUTONEG_ERROR_MASK	0x8000
#define       PHY_IDENTIFIER_1 REG       2         #define       PHY_IDENTIFIER_2 REG       3         #define       PHY_DETECT_MASK       0x1808         #define       PHY_DETECT_MASK       0x0141         #define       PHY_MARVELL_IDENTIFIER       0x2000         #define       PHY_TI_IDENTIFIER       0x2000         #define       PHY_XILINX_PCS_PMA_ID1       0x0174         #define       PHY_XILINX_PCS_PMA_ID2       0x0C00         #define       MICREL_PHY_IDENTIFIER       0x220         #define       MICREL_PHY_IDENTIFIER       0x220         #define       MICREL_PHY_KS29031 MODEL       0x220         #define       MICREL_PHY_KS29031 MODEL       0x220         #define       XEMACPS_GMIL2RGMII_SPEED1000_FD       0x140         #define       XEMACPS_GMIL2RGMII_SPEED100_FD       0x2100         #define       XEMACPS_GMIL2RGMII_SPEED100_FD       0x100         #define       XEMACPS_GMIL2RGMII_SPEED100_FD       0x140	#define PHY_DETECT_REG	1
#define PHY_IDENTIFIER_2 REG       3         #define PHY_DETECT_MASK       0x1808         #define PHY_MARVELL_IDENTIFIER       0x0141         #define PHY_TI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0C00         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_KSZ9031 MODEL       0x2200         #define XEMACPS_GMI12RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100         #define XEMACPS_GMI12RGMII_SPEED10_FD       0x100	#define PHY_IDENTIFIER_1_REG	2
#define PHY_DETECT_MASK       0x1808         #define PHY_MARVELL_IDENTIFIER       0x0141         #define PHY_TI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0C00         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_KSZ9031 MODEL       0x220         #define XEMACPS_GMI12RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100         #define XEMACPS_GMI12RGMII_SPEED10_FD       0x100	#define PHY_IDENTIFIER_2_REG	3
#define PHY_MARVELL_IDENTIFIER       0x0141         #define PHY_TI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0000         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_KS29031_MODEL       0x220         #define XEMACPS_GMI12RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100         #define XEMACPS_GMI12RGMII_SPEED10_FD       0x100         #define XEMACPS_GMI12RGMII_SPEED10_FD       0x100	#define PHY DETECT MASK	0x1808
#define PHY_TI_IDENTIFIER       0x2000         #define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0000         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_KSZ9031_MODEL       0x220         #define XEMACPS_GMI12RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x2100         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100         #define XEMACPS_GMI12RGMII_SPEED100_FD       0x100	#define PHY MARVELL IDENTIFIER	0x0141
#define PHY_XILINX_PCS_PMA_ID1       0x0174         #define PHY_XILINX_PCS_PMA_ID2       0x0C00         #define MICREL_PHY_IDENTIFIER       0x220         #define MICREL_PHY_KSZ9031_MODEL       0x220         #define MICREL_PHY_KSZ9031_MODEL       0x220         #define XEMACPS_GMII2RGMII_SPEED1000_FD       0x140         #define XEMACPS_GMII2RGMII_SPEED100_FD       0x2100         #define XEMACPS_GMII2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMII2RGMII_SPEED100_FD       0x140         #define XEMACPS_GMII2RGMII_SPEED100_FD       0x100	#define PHY_TI_IDENTIFIER	0x2000
<pre>#define PHY_XILINX_PCS_PMA_ID2 0x0C00 #define MICREL PHY_IDENTIFIER 0x22 #define MICREL PHY_KSZ9031 MODEL 0x220 #define XEMACPS_GMI12RGMII_SPEED1000_FD 0x140 #define XEMACPS_GMI12RGMII_SPEED100_FD 0x2100 #define XEMACPS_GMI12RGMII_SPEED10_FD 0x100 #define XEMACPS_GMI12RGMII_SPEED10_FD 0x100</pre>	#define PHY XILINX PCS PMA ID1	0x0174
#define MICREL PHY IDENTIFIER       0x22         #define MICREL PHY KSZ9031 MODEL       0x220         #define XEMACPS GMII2RGMII SPEED1000 FD       0x140         #define XEMACPS GMII2RGMII SPEED100 FD       0x2100         #define XEMACPS GMII2RGMII SPEED100 FD       0x100         #define XEMACPS GMII2RGMII SPEED10 FD       0x100	#define PHY_XILINX_PCS_PMA_ID2	0x0C00
#define MICREL PHY KSZ9031 MODEL     0x220       #define XEMACPS_GMI12RGMII_SPEED1000_FD     0x140       #define XEMACPS_GMI12RGMII_SPEED1000_FD     0x2100       #define XEMACPS_GMI12RGMII_SPEED100_FD     0x100       #define XEMACPS_GMI12RGMII_SPEED10_FD     0x100	<pre>#define MICREL_PHY_IDENTIFIER</pre>	0x22
#define XEMACPS_GMI12RGMII_SPEED1000_FD         0x140           #define XEMACPS_GMI12RGMII_SPEED100_FD         0x2100           #define XEMACPS_GMI12RGMII_SPEED100_FD         0x100           #define XEMACPS_GMI12RGMII_SPEED10_FD         0x100	#define MICREL PHY KSZ9031 MODEL	0x220
#define XEMACPS_GMII2RGMII_SPEED1000_FD         0x140           #define XEMACPS_GMII2RGMII_SPEED100_FD         0x2100           #define XEMACPS_GMII2RGMII_SPEED10_FD         0x100           #define XEMACPS_GMII2RGMII_SPEED10_FD         0x100		
#define XEMACPS_GMII2RGMII_SPEED100_FD         0x2100           #define XEMACPS_GMII2RGMII_SPEED10_FD         0x100           #define XEMACPS_CMII2RGMII_DEC_NUM         0x100	<pre>#define XEMACPS_GMII2RGMII_SPEED1000_F</pre>	D 0x140
#define XEMACPS_GMII2RGMII_SPEED10_FD 0x100 #define XEMACPS_CMII2DCMII_DEC_NUM 0x10	<pre>#define XEMACPS_GMII2RGMII_SPEED100_FI</pre>	0x2100
#dofine VEMACRE CMITARCMIT DEC MIM 0w10	<pre>#define XEMACPS_GMII2RGMII_SPEED10_FD</pre>	0x100
	#dofine VEMACHE CMITADOMIT DEC MUM	01110

Figure 3-4: Add a macro definition

7) Add phy speed get function



```
XEmacPs_PhyWrite(xemacpsp,phy_addr, IEEE_PAGE_ADDRESS_REGISTER, 2);
XEmacPs_PhyRead(xemacpsp, phy_addr, IEEE_CONTROL_REG_MAC, &control);
control |= IEEE_RGMII_TXRX_CLOCK_DELAYED_MASK;
XEmacPs_PhyWrite(xemacpsp, phy_addr, IEEE_CONTROL_REG_MAC, control);
XEmacPs_PhyWrite(xemacpsp, phy_addr, IEEE_PAGE_ADDRESS_REGISTER, 0);
XEmacPs_PhyRead(xemacpsp, phy_addr, IEEE_AUTONEGO_ADVERTISE_REG, &control);
control |= IEEE_ASYMMETRIC_PAUSE_MASK;
control |= IEEE PAUSE MASK;
control |= ADVERTISE 100;
control |= ADVERTISE_10;
XEmacPs_PhyWrite (xemacpsp, phy_addr, IEEE_AUTONEGO_ADVERTISE_REG, control);
XEmacPs_PhyRead(xemacpsp, phy_addr, IEEE_1000_ADVERTISE_REG_OFFSET,
                                &control);
control |= ADVERTISE_1000;
XEmacPs_PhyWrite(xemacpsp, phy_addr, IEEE_1000_ADVERTISE_REG_OFFSET,
                                control);
XEmacPs_PhyWrite(xemacpsp, phy_addr, IEEE_PAGE_ADDRESS_REGISTER, 0);
XEmacPs PhyRead (xemacpsp, phy addr, IEEE COPPER SPECIFIC CONTROL REG,
                                &control);
control |= (7 << 12); /* max number of gigabit attempts */</pre>
control |= (1 << 11); /* enable downshift */</pre>
XEmacPs_PhyWrite(xemacpsp, phy_addr, IEEE_COPPER_SPECIFIC_CONTROL_REG,
                                control);
XEmacPs_PhyRead(xemacpsp, phy_addr, IEEE_CONTROL_REG_OFFSET, &control);
control |= IEEE_CTRL_AUTONEGOTIATE_ENABLE;
control |= IEEE_STAT_AUTONEGOTIATE_RESTART;
XEmacPs_PhyWrite(xemacpsp, phy_addr, IEEE_CONTROL_REG_OFFSET, control);
XEmacPs_PhyRead(xemacpsp, phy_addr, IEEE_CONTROL_REG_OFFSET, &control);
```

```
control |= IEEE CTRL RESET MASK;
XEmacPs_PhyWrite(xemacpsp, phy_addr, IEEE_CONTROL_REG_OFFSET, control);
while (1) {
        XEmacPs_PhyRead(xemacpsp, phy_addr, IEEE_CONTROL_REG_OFFSET, &control);
        if (control & IEEE CTRL RESET MASK)
                continue;
        else
                break;
}
XEmacPs_PhyRead(xemacpsp, phy_addr, IEEE_STATUS_REG_OFFSET, &status);
xil_printf("Waiting for PHY to complete autonegotiation.\r\n");
while ( !(status & IEEE STAT AUTONEGOTIATE COMPLETE) ) {
        sleep(1);
        XEmacPs_PhyRead(xemacpsp, phy_addr,
                                        IEEE COPPER SPECIFIC STATUS REG 2, &temp);
        timeout_counter++;
        if (timeout_counter == 30) {
                xil_printf("Auto negotiation error \r\n");
                return;
        }
        XEmacPs_PhyRead(xemacpsp, phy_addr, IEEE_STATUS_REG_OFFSET, &status);
}
xil_printf("autonegotiation complete \r\n");
XEmacPs PhyRead(xemacpsp, phy addr, 0x1f,
                                &status_speed);
if ( (status speed & 0x40) == 0x40)/* 1000Mbps */
        return 1000;
```

```
else if ( (status_speed & 0x20) == 0x20)/* 100Mbps */
        return 100;
else if ( (status_speed & 0x10) == 0x10)/* 10Mbps */
        return 10;
else
        return 0;
return XST_SUCCESS;
}
```

#### 8) Modify the function "get\_IEEE\_phy\_speed" to add support for KSZ9031



### 3.2 Create an APP based on the LWIP template



Figure 3-5: Create an APP based on the LWIP template

#### 3.3 Download debugging

If the system has both a PS Ethernet controller and a PL AXI Ethernet controller, the LWIP template will select the PL AXI Ethernet controller by default. We first test the PL port Ethernet. The test environment requires a router that supports "dhcp". The FPGA development board connects to the router to

automatically obtain the IP address. The experiment host and the FPGA development board are in a network and can communicate with each other.

#### 3.4 PL port Ethernet Test

- Connect the serial port to open the serial debugging terminal, connect the PL Ethernet cable to the router (ETH1), and only use ETH1 in the VIvado project, so other ports are not available.
- ➢ Run the SDK

Run or Debug a program usin	g System Debugger.			
	Name: System Debugg	er using Debug_lwip.elf on Local		
type filter text	🧿 Target Setup 📑	Application 🗱 Arguments 🖾 Er	nvironment 🔂 Symbo	ol Files »3
get Communication Framewo nx C/C++ application (GDB) nx C/C++ application (Systen nx C/C++ application (Systen	Hardware Platform: Bitstream File:	design_1_wrapper_hw_platform_t design_1_wrapper.bit	D 🔹	Browse
System Debugger using Deb	Initialization File:	ps7_init.tcl	Search	Browse
	FPGA Device:	Auto Detect	Select	
	PS Device:	Auto Detect	Select	
		Summary of operations to	be performed	
	<ul> <li>Reset entire syste</li> <li>Program FPGA</li> <li>Run ps7_init</li> <li>Run ps7_post_cord</li> </ul>	<ul> <li>Following operations will I</li> <li>Resets entire system. Cl</li> <li>Program FPGA fabric (P</li> <li>Runs ps7_init to initialize</li> <li>Runs ps7_post_config. E</li> </ul>	be performed before ears the FPGA fabric () L). e PS. inables level shifters fr	launching the de PL). rom PL to PS. (Re
	4	III		•
۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰			Revert	Apply
	fe		Rup	Close

Figure 3-6: Run Configuration

You can see that the serial port prints some information, you can see that the address is automatically obtained as "192.168.1.68", the connection speed is 1000Mbps, and the tcp port is 7



Figure 3-7: Board IP

Connect using telnet

tegory.	1.1	
Session	Basic options for your Pu	JTTY session
Logging     Terminal     Keyboard     Bell     Features     Window     Appearance     Behaviour     Translation     Solution	Specify the destination you want t Host Name (or IP address) 192 168 1.68	o connect to Port 7
	Connection type: Raw I Telnet Riogin	SSH Serial
	Load, save or delete a stored sess Saved Sessions	sion
Colours Connection Data Proxy Telnet Riogin ccu scu	Default Settings com3	Load Save Delete
terial	Close window on exit: Always Never O O	nly on clean exit

Figure 3-8: Board IP

The development board returns the same character when entering a character

🗗 192.168.1.68 - PuTTY	STATISTICS.	
hheelllllllllll122		*
		+

Figure 3-9: Returns the same character when entering a character

#### 3.4.1: Test other channels

By default, the program uses the first network port of the PL end, and can test other channels by modifying the file platform\_config.h.

#define PLATFORM\_EMAC\_BASEADDR XPAR\_AXI\_ETHERNET\_0\_BASEADDRTest channel 1 #define PLATFORM\_EMAC\_BASEADDR XPAR\_AXI\_ETHERNET\_1\_BASEADDRTest channel 2

#define PLATFORM\_EMAC\_BASEADDR XPAR\_AXI\_ETHERNET\_2\_BASEADDR Test channel 3 #define PLATFORM\_EMAC\_BASEADDR XPAR\_AXI\_ETHERNET\_3\_BASEADDR Test channel 4

Due to the LWIP library itself, it is not possible to test multiple channels at the same time. If you use a Linux system, you can easily use multiple channels.



Figure 3-10: Test other channels by modifying the file platform\_config.h

#### 3.5 PS port Ethernet Test

Modify BSP settings



Figure 3-11: Modify BSP settings

"use\_axieth\_on\_zynq" is changed to 0, using PS Ethernet

oard Support Package Set	t <b>ings</b> your Board Support Package.				
Overview standalone	Configuration for library: wip14	41			
lwip141	Name	Value	Default	Туре	Description
drivers	api_mode	RAW API (RAW_API)	RAW_API	enum	Mode of operation for IwIP
ps7_cortexa9_0	lwip_tcp_keepalive	false	false	boolean	Enable keepalive processing
	no_sys_no_timers	true	true	boolean	Drops support for sys_timed
	socket_mode_thread_prio	2	2	integer	Priority of threads in socket
	use_axieth_on_zynq	0	1	integer	Option if set to 1 ensures as
	use_emaclite_on_zynq	1	1	integer	Option if set to 1 ensures er
	arp_options	true	true	boolean	ARP Options
	debug_options	true	true	boolean	Turn on IwIP Debug?
	b dhcp_options	true	true	boolean	Is DHCP required?
	icmp_options	true	true	boolean	ICMP Options
	igmp_options	false	false	boolean	IGMP Options
	Iwip_ip_options	true	true	boolean	IP Options
	Iwip_memory_options				Options controlling lwIP me
	b pbuf_options	true	true	boolean	Pbuf Options
	stats_options	true	true	boolean	Turn on lwIP statistics?
	tcp_options	true	true	boolean	Is TCP required ?
	temac_adapter_options	true	true	boolean	Settings for xps-ll-temac/Ax
	udp_options	true	true	boolean	Is UDP required ?

Figure 3-12: "use\_axieth\_on\_zynq" is changed to 0

Modify the "platform\_config.h" file

1. A 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	hdt	system.mss	[C] main.c	b platform	_contig.h 🛛	
#ifnd	lef	PLATFORM_CONFIG	<u>H</u>			
#defi	ne US	E_SOFTETH_ON_ZY	NQ Ø		- Alf fairly for one on an	
//#de	tine	PLATFORM EMAC B/	ASEADDR XPAR A	XI ETHERNET	0 BASEADD	R
#defi	ne PL	ATFORM_EMAC_BASI	EADDR XPAR_PS7	ETHERNET 0	BASEADDR	-
#defi	ne PL	ATFORM ZYNQ				N

Figure 3-13: Modify the "platform\_config.h" file

- > Network cable connects PS end Ethernet to router
- Run the program and observe the serial output

Phy 1 is KSZ9031	
Start PHY autonegotiation	
Waiting for PHY to complete autonegotiation.	
autonegotiation complete	
auto-negotiated link speed: 1000	
Board IP: 192.168.1.68	
Netmask : 255.255.255.0	
Gateway : 192.168.1.1	
TCP echo server started @ port 7	
lwIP TCP echo server	
lwIP TCP echo server TCP packets sent to port 6001 will be echoed WARNING: Not a Marvell or TI Ethernet PHY. P ence Start PHY autonegotiation	l back Please verify the initialization sequ
lwIP TCP echo server TCP packets sent to port 6001 will be echoed WARNING: Not a Marvell or TI Ethernet PHY. P ence Start PHY autonegotiation Waiting for PHY to complete autonegotiation.	l back Please verify the initialization segu
lwIP TCP echo server TCP packets sent to port 6001 will be echoed WARNING: Not a Marvell or TI Ethernet PHY. P ence Start PHY autonegotiation Waiting for PHY to complete autonegotiation. autonegotiation complete	l back Please verify the initialization segner
lwIP TCP echo server TCP packets sent to port 6001 will be echoed WARNING: Not a Marvell or TI Ethernet PHY. P ence Start PHY autonegotiation Waiting for PHY to complete autonegotiation. autonegotiation complete link speed for phy address 1: 1000	l back Please verify the initialization segner
lwIF TCP echo server TCP packets sent to port 6001 will be echoed WARNING: Not a Marvell or TI Ethernet PHY. P ence Start PHY autonegotiation Waiting for PHY to complete autonegotiation. autonegotiation complete link speed for phy address 1: 1000 Board IP: 192.168.1.68	l back Please verify the initialization segu
lwIP TCP echo server TCP packets sent to port 6001 will be echoed WARNING: Not a Marvell or TI Ethernet PHY. P ence Start PHY autonegotiation Waiting for PHY to complete autonegotiation. autonegotiation complete link speed for phy address 1: 1000 Board IP: 192.168.1.68 Netmask : 255.255.255.0	l back Please verify the initialization sequ
lwIP TCP echo server TCP packets sent to port 6001 will be echoed WARNING: Not a Marvell or TI Ethernet PHY. P ence Start PHY autonegotiation Waiting for PHY to complete autonegotiation. autonegotiation complete link speed for phy address 1: 1000 Board IP: 192.168.1.68 Netmask : 255.255.255.0 Gateway : 192.168.1.1	l back Please verify the initialization sequ

Figure 3-14: Run the program and observe the serial Output

## Part 4: Application under petalinux

This section describes how to configure multiple Ethernet drivers in petalinux.

### 4.1 Create a petalinux project

Copy the \*.sdk folder to the linux system and use it for petalinux

.xil	2018/10/10 17:41	文件夹	
ax7021_multi_eth.cache	2018/10/10 15:13	文件夹	
ax7021_multi_eth.hw	2018/10/10 15:13	文件夹	
ax7021_multi_eth.ip_user_files	2018/10/10 15:14	文件夹	
ax7021_multi_eth.runs	2018/10/10 15:17	文件夹	
ax7021_multi_eth.sdk	2018/10/10 17:42	文件夹	
ax7021_multi_eth.sim	2018/10/10 15:13	文件夹	
ax7021_multi_eth.srcs	2018/10/10 15:13	文件夹	
ax7021_multi_eth.xpr	2018/10/10 15:44	Vivado Project Fi	8 KB
vivado.jou	2018/10/10 17:40	JOU 文件	3 KB
vivado.log	2018/10/10 17:40	LOG 文件	32 KB
vivado_pid9456.str	2018/10/10 15:13	STR 文件	119 KB

Figure 4-1: Copy the \*.sdk folder to the linux system

Open

terminal



#### Figure 4-2: Open the Terminal

Enter the command in Figure 4-3 for petalinux and VIVADO software environment variable settings

source /opt/pkg/petalinux/settings.sh

source /opt/Xilinx/Vivado/2017.4/settings64.sh

Figure 4-3: Software Environment Variable Settings

petalinux-create --type project --template zynq --name axi\_eth

Figure 4-4: Create a Petalinux Project

After the project is created, petalinux will automatically create a project directory "axi eth" and run the command to enter the directory.

cd axi\_eth

Figure 4-5: Run theCcommand to enter the directory "axi\_eth"

#### 4.2 Configure petalinux hardware related information

Run the command in Figure 4-6 to configure the hardware information of the petalinux project. "../\*.sdk/" is the hardware information directory exported by VIVADO. After the command is run, a configuration interface will pop up.

Here you can mainly configure the startup method. The following describes how to configure the startup mode. The default configuration starts from the startup of the SD card. If you need to modify it, you can directly "Save" and then "Exit" to exit.

"sdk" to be modified according to the actual file name

```
petalinux-config --get-hw-description ../*.sdk/
```

Figure 4-6: Configure the hardware information of the petalinux project

Then petalinux started the automatic configuration process, there are some warnings, no errors.

#### 4.3 Modify the device tree

Find the "system-top.dts" file in the project directory, which is the top-level file for the device tree.



Figure 4-7: "system-top.dts" file in the Project Directory

Modify the system-user.dtsi content to:

```
/include/ "system-conf.dtsi"
/{
       model = "Zynq ALINX Development Board";
       compatible = "alinx,axi eth", "xlnx,zynq-7000";
       usb_phy0: usb_phy@0 {
               compatible = "ulpi-phy";
               #phy-cells = <0>;
               reg = <0xe0002000 0x1000>;
               view-port = <0x0170>;
               drv-vbus;
       };
};
&usb0 {
       usb-phy = <&usb_phy0>;
};
```

```
&sdhci0 {
       u-boot,dm-pre-reloc;
};
&uart1 {
       u-boot,dm-pre-reloc;
};
&flash0 {
       compatible = "micron,m25p80", "w25q256", "spi-flash";
};
&gem0 {
       phy-handle = <&ethernet_phy>;
       ethernet_phy: ethernet-phy@1 {
               reg = <1>;
               device_type = "ethernet-phy";
       };
};
&axi_ethernet_0 {
       local-mac-address = [00 0a 35 00 03 22];
       phy-handle = <&phy1>;
       xlnx,has-mdio = <0x1>;
```

```
phy-mode = "rgmii";
       mdio {
               phy1:phy@1{
                       device_type = "ethernet-phy";
                       reg = <1>;
               };
       };
};
&axi_ethernet_1 {
       local-mac-address = [00 0a 35 00 03 23];
       phy-handle = <&phy2>;
       xlnx,has-mdio = <0x1>;
       phy-mode = "rgmii";
       mdio {
               phy2:phy@1{
                       device_type = "ethernet-phy";
                       reg = <1>;
               };
       };
};
&axi_ethernet_2 {
       local-mac-address = [00 0a 35 00 03 24];
       phy-handle = <&phy3>;
       xlnx,has-mdio = <0x1>;
```

```
phy-mode = "rgmii";
        mdio {
               phy3: phy@1 {
                       device_type = "ethernet-phy";
                       reg = <1>;
               };
       };
};
&axi_ethernet_3 {
       local-mac-address = [00 0a 35 00 03 25];
        phy-handle = <&phy4>;
        xlnx,has-mdio = <0x1>;
        phy-mode = "rgmii";
        mdio {
               phy4: phy@1 {
                       device_type = "ethernet-phy";
                       reg = <1>;
               };
       };
};
```

### 4.4 Configuring the kernel

Enter the command in Figure 4-8 to configure the kernel.

petalinux-config -c kernel

#### Figure 4-8: Command to configure the kernel

#### 4.4.1: Configure phy driver

Select "<\*> Micrel PHYs" in the "Device Drivers > Network device support > PHY Device support and infrastructure", then "Save", and finally "Exit" to exit the configuration interface.



Figure 4-9: Configure phy driver

#### 4.5 Configuring the root file system

Do not make any changes to the linux root file system. If you need to modify the root file system, you can run the following command in Figure 4-10:

```
petalinux-build
```

Figure 4-10: Modify the Root File System

#### 4.6 Build petalinux

Run the command in Figure 4-11 to compile and build the petalinux project.

petalinux-build

Figure 4-11: Compile and build the petalinux project

#### 4.7 Start petalinux from SD card

Run the command in Figure 4-12 to generate the startup file.

petalinux-package --boot --fsbl ./images/linux/zynq\_fsbl.elf --fpga --uboot --force

Figure 4-12: Generate the startup file

Copy the BOOT.BIN and image.ub files to the FAT32 partition of the SD card

#### 4.8 Configuring the Ethernet port

First use the serial terminal tool to log in to the system, use the root account, password root

Enter the following configuration Ethernet IP address and start it as follows

ifconfig eth1 192.168.1.101 netmask 255.255.255.0 up

ifconfig eth2 192.168.1.102 netmask 255.255.255.0 up

ifconfig eth3 192.168.1.103 netmask 255.255.255.0 up

ifconfig eth4 192.168.1.104 netmask 255.255.255.0 up

Figure 4-13: Configuration Ethernet IP address

```
PetaLinux 2017.4 axi_eth /dev/ttyPS0
axi_eth login: root
Password:
root@axi_eth:~ # ifconfig eth1 192.168.1.101 netmask 255.255.255.0 up
IPv6: ADDRCONF (NETDEV_UP): eth1: link is not ready
root@axi_eth:~# ifconfig eth2 192.168.1.102 netmask 255.255.255.0 up
IPv6: ADDRCONF(NEIDEV_UP): eth2: link is not ready
root@axi_eth:~ # ifconfig eth3 192.168.1.103 netmask 255.255.255.0 up
IPv6: ADDRCONF (NETDEV_UP): eth3: link is not ready
root@axi_eth:~ # ifconfig eth4 192.168.1.104 netmask 255.255.255.0 up
IPv6: ADDRCONF(NETDEV_UP): eth4: link is not ready
root@axi_eth:~ # xilinx_axienet 41000000.ethernet eth1: Link is Down
xilinx_axienet 41040000.ethernet eth2: Link is Down
xilinx_axienet 41080000.ethernet eth3: Link is Down
xilinx_axienet 410c0000.ethernet eth4: Link is Down
xilinx_axienet 41040000.ethernet eth2: Link is Up - 1Gbps/Full - flow control rx/tx
IPv6: ADDRCONF(NETDEV_CHANGE): eth2: link becomes ready
root@axi_eth:~#
```



Run "ifconfig" command to view status

LOOC@AX1_6	eth: # liconing
ethO	Link encap:Ethernet HWaddr 00:0A:35:00:1E:53 inet addr:192.168.1.46 Bcast:192.168.1.255 Mask:255.255.255.0 inet6 addr: fe80::20a:35ff:fe00:1e53%lo/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:10 errors:0 dropped:1 overruns:0 frame:0 TX packets:8 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:3419 (3.3 KiB) TX bytes:1192 (1.1 KiB) Interrupt:27 Base address:0xb000
eth1	Link encap:Ethernet HWaddr 00:0A:35:00:03:22 inet addr:192.168.1.101 Bcast:192.168.1.255 Mask:255.255.255.0 UP BROADCAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
eth2	Link encap:Ethernet HWaddr 00:0A:35:00:1E:53 inet addr:192.168.1.102 Bcast:192.168.1.255 Mask:255.255.255.0 inet6 addr: fe80::20a:35ff:fe00:1e53%lo/64 Scope:Link UP BROADCAST MTU:1500 Metric:1 RX packets:532 errors:0 dropped:0 overruns:0 frame:0 TX packets:13 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:77058 (75.2 KiB) TX bytes:998 (998.0 B)
eth3	Link encap:Ethernet HWaddr 00:0A:35:00:03:24 inet addr:192.168.1.103 Bcast:192.168.1.255 Mask:255.255.255.0 UP BROADCAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
eth4	Link encap:Ethernet HWaddr 00:0A:35:00:03:25 inet addr:192.168.1.104 Bcast:192.168.1.255 Mask:255.255.255.0 UP BROADCAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
10	Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 inet6 addr: ::1%1/128 Scope:Host UP LOOPBACK RUNNING MTU:65536 Metric:1 RX packets:2356 errors:0 dropped:0 overruns:0 frame:0 TX packets:2356 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1 RX bytes:685456 (669.3 KiB) TX bytes:685456 (669.3 KiB)

Figure 4-15: Configuring the Ethernet port

## **4.9 Test the Ethernet port**

Plug the first PL port Ethernet port (ETH1) into the cable, Use eth1 to ping

192.168.1.10 in the LAN

ping -l eth1 192.168.1.10

ro	ot@axi_	eth:	°# pi	ng -	·I	ethi 19	92.168.	1.10	
ΡI	NG 192.	168.1	1.10	(192.	168.	1.10):	56 dat:	a bytes	
64	bytes	from	192.	168.1	.10:	seq=0	ttl=64	time=0.382	ms
64	bytes	from	192.	168.1	.10:	seq=1	ttl=64	time=0.207	ms
64	bytes	from	192.	168.1	.10:	seq=2	ttl=64	time=0.197	ms
64	bytes	from	192.	168.1	.10:	seq=3	ttl=64	time=0.183	ms
64	bytes	from	192.	168.1	.10:	seq=4	ttl=64	time=0.177	ms

Figure 4-16: Test the Ethernet port

## Part 5: Q&A

#### 5.1 Petalinux Startup Error

If the error shown in Figure 5-1 appears, you can format the sd card and try again.

```
U-Boot 2015.07 (Oct 12 2017 - 13:33:23 +0800)
DRAM: ECC disabled 1 GiB
MMC: zynq_sdhci: 0
SF: Detected W25Q256 with page size 256 Bytes, erase size 4 KiB, total 32 MiB
*** Warning - bad CRC, using default environment
In: serial
Out: serial
Err: serial
Net: Gem.e000b000
U-BOOT for ax7021_multi_eth
Gem.e000b000 waiting for PHY auto negotiation to complete...... TIMEOUT !
Gem.e000b000: No link.
Hit any key to stop autoboot: 0
Device: zynq_sdhci
Manufacturer ID: 3
OEM: 5344
Name: SL08G
Tran Speed: 50000000
Rd Block Len: 512
SD version 3.0
High Capacity: Yes
Capacity: 7.4 GiB
Bus width: 4-bit
Erase Group Size: 512 Bytes
reading image.ub
Invalid FAT entry
77824 bytes read in 16 ms (4.6 MiB/S)
## Loading kernel from FIT Image at 01000000 ...
Bad FIT kernel image format!
ERROR: can't get kernel image!
U-BOOT-PetaLinux>
```

Figure 5-1: Petalinux Startup Error