高级练习8 RGMII 接口和 IDDR 原语的使用和千兆 PHY 数据接收

一、练习内容

1.总体项目要求

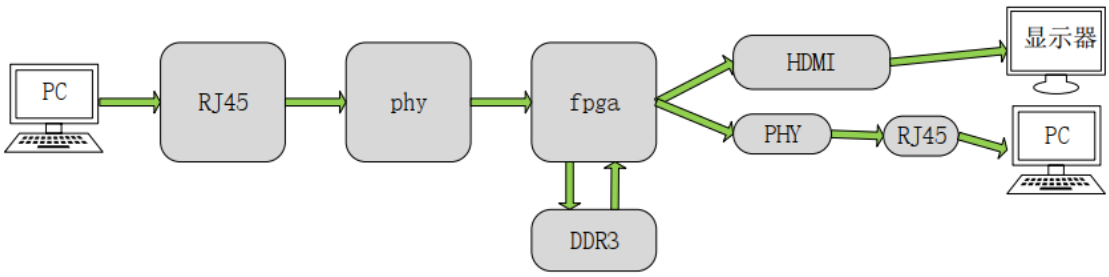
使用 PC 机将图像数据信息通过千兆以太网发送给 FPGA，存储到DDR3 中， HDMI 将 DDR3 中数据读出显示到显示器上，同时将图像数据回传到PC 机对应的上位机上。

2.RX端项目要求

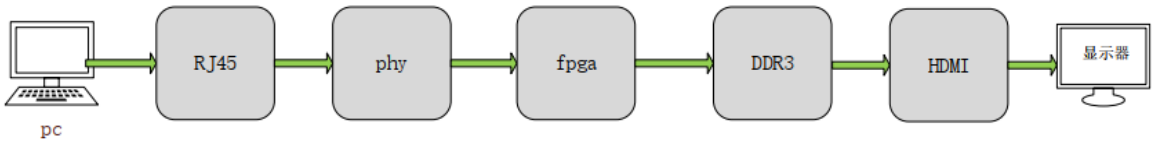
通过 PC 机将图像信息发送到 FPGA， FPGA 将数据处理后存储到 DDR3 中， HDMI 读出 DDR3 中的数据显示在显示器上。

二、系统框图

1.总体系统框图



2. RX端系统框图



三、设计分析

1.PHY芯片配置

PHY芯片有两种配置方式，一是通过配置寄存器到达配置效果，另一个就是通过配置端口电阻达到不同的效果

（1）CMODE模式配置

通过配置CMODE引脚连接不同的电阻，达到不同的配置效果。不同端口的引脚电阻和电平配置控制了相应的电平变量。

Table 49. CMODE Configuration Pins and Device Functions

CMODE Pin	Bit 3 (MSB) Control	Bit 2 Controls	Bit 1 Controls	Bit 0 (LSB) Controls
3	PHY address [3]	PHY address [4]	MAC calibration setting[1]	MAC calibration setting[0]
2	PHY address [2]	ActiPHY	RGMII clock skew[1]	RGMII clock skew[0]
1	PHY address [1]	Link speed downshift	Speed/Duplex Modes [1]	Speed/Duplex modes [0]
0	PHY address [0]	CLKOUT enable	Advertise asymmetric pause	Advertise symmetric pause

具体来说，参考下面的说明。

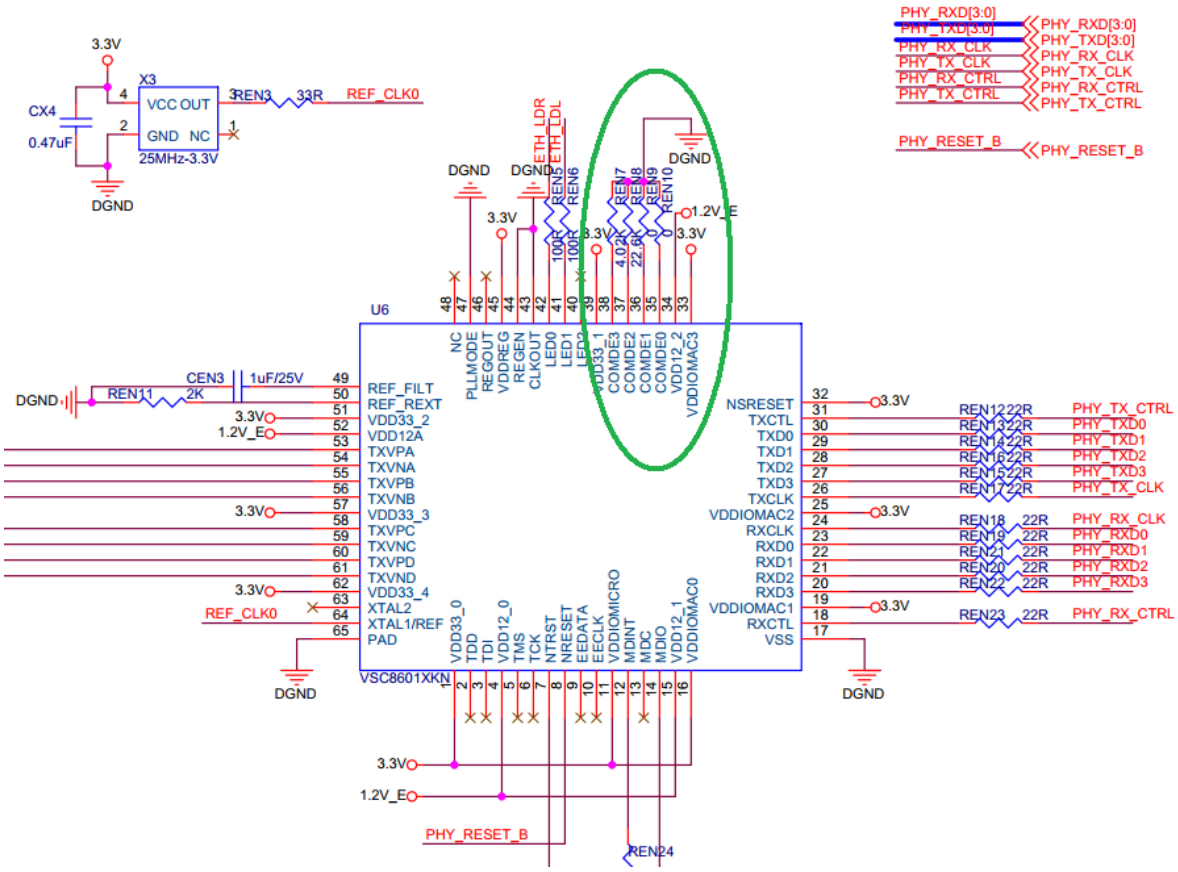
Table 51. CMODE Resistor Values and Resultant Bit Settings

With CMODE Pin Tied To	With 1% Resistor Value	Set Bit 3 (MSB) to:	Set Bit 2 to:	Set Bit 1 to:	Set Bit 0 (LSB) to:
VSS	0	0	0	0	0
VSS	2.26 kΩ	0	0	0	1
VSS	4.02 kΩ	0	0	1	0
VSS	5.90 kΩ	0	0	1	1

Table 51. CMODE Resistor Values and Resultant Bit Settings *(continued)*

With CMODE Pin Tied To	With 1% Resistor Value	Set Bit 3 (MSB) to:	Set Bit 2 to:	Set Bit 1 to:	Set Bit 0 (LSB) to:
VSS	8.25 kΩ	0	1	0	0
VSS	12.1 kΩ	0	1	0	1
VSS	16.9 kΩ	0	1	1	0
VSS	22.6 kΩ	0	1	1	1
VDD33	0	1	0	0	0
VDD33	2.26 kΩ	1	0	0	1
VDD33	4.02 kΩ	1	0	1	0
VDD33	5.90 kΩ	1	0	1	1
VDD33	8.25 kΩ	1	1	0	0
VDD33	12.1 kΩ	1	1	0	1
VDD33	16.9 kΩ	1	1	1	0
VDD33	22.6 kΩ	1	1	1	1

本次练习的电路板的连接图如下图所示。



故相应引脚对应的电平为CMOD0对应0000，CMOD1对应0000，CMOD2对应0111，CMOD对应0010，实际对应的配置就是PHY的地址为00000，使能千兆以太网，然后PHY芯片的接收时钟RX_CLK和TX_CLK之间有2us的差值。

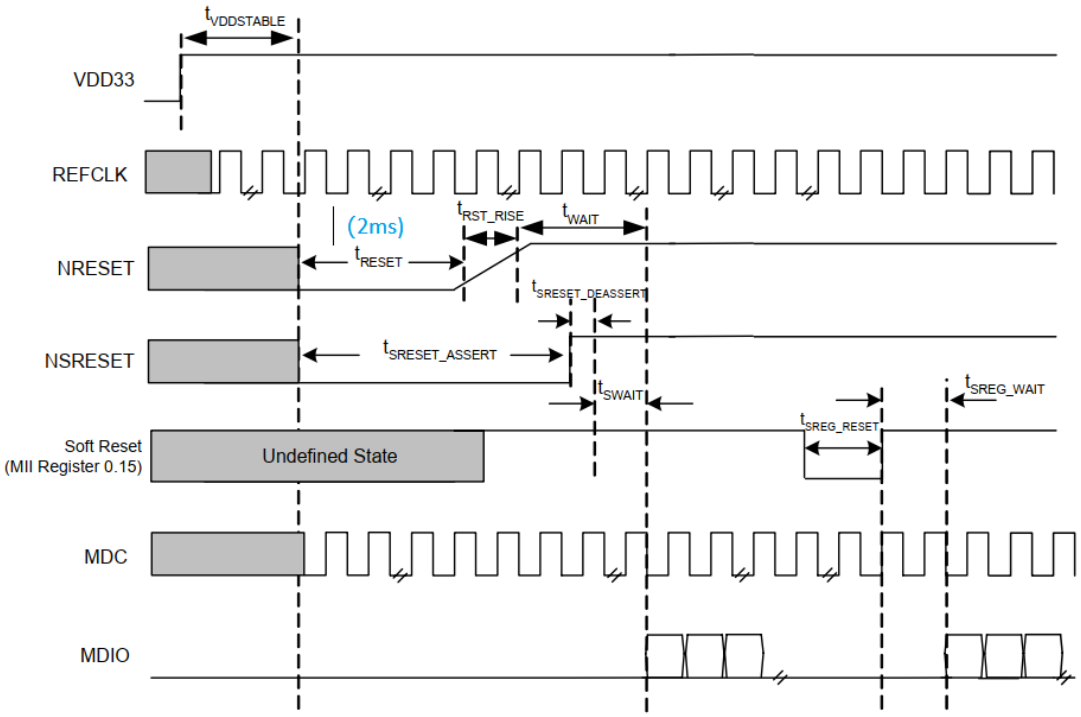
Table 50. Device Functions and Associated CMODE Pins

Function	CMODE Pin	Bit	Associated Register, Bit	Result
PHY Address [4:0]	3 to 0	3 and 2		Sets the PHY address.
Link speed downshift	1	2	Register 20E, bit 4	0 = Link only according to the auto-negotiation resolution. 1 = Enable link speed downshift feature.
Speed and duplex	1	1 and 0	Register 4, bits 8:5 and Register 9, bits 9:8	00 = 10/100/1000BASE-T FDX/HDX. 01 = 10/100/1000BASE-T FDX; 10/100BASE-T HDX. 10 = 1000BASE-T FDX only. 11 = 10/100BASE-T FDX/HDX.
RGMII clock skew	2	1 and 0	Register 23, bit 8	00 = No skew on RX_CLK and TX_CLK. 01 = 1.4 ns skew on RX_CLK and TX_CLK. 10 = 1.7 ns skew on RX_CLK and TX_CLK. 11 = 2.0 ns skew on RX_CLK and TX_CLK.
Advertise asymmetric pause	0	1	Register 4, bit 11	0 = Not advertised. 1 = Advertised.
Advertise symmetric pause	0	0	Register 4, bit 10	0 = Not advertised. 1 = Advertised.
CLKOUT enable	0	2	Register 18, bit 0	0 = Disabled. 1 = Enabled.
ActiPHY	2	2	Register 23, bit 5	0 = Disabled. 1 = Enabled.
MAC resistor calibration setting	3	1 and 0	Register 19E, bits 15:14	00 = 50 Ω. 01 = 60 Ω. 10 = 30 Ω. 11 = 45 Ω.

(2) PHY芯片的初始化配置

具体时序查考下面的时序图。

Figure 20. Reset Timing

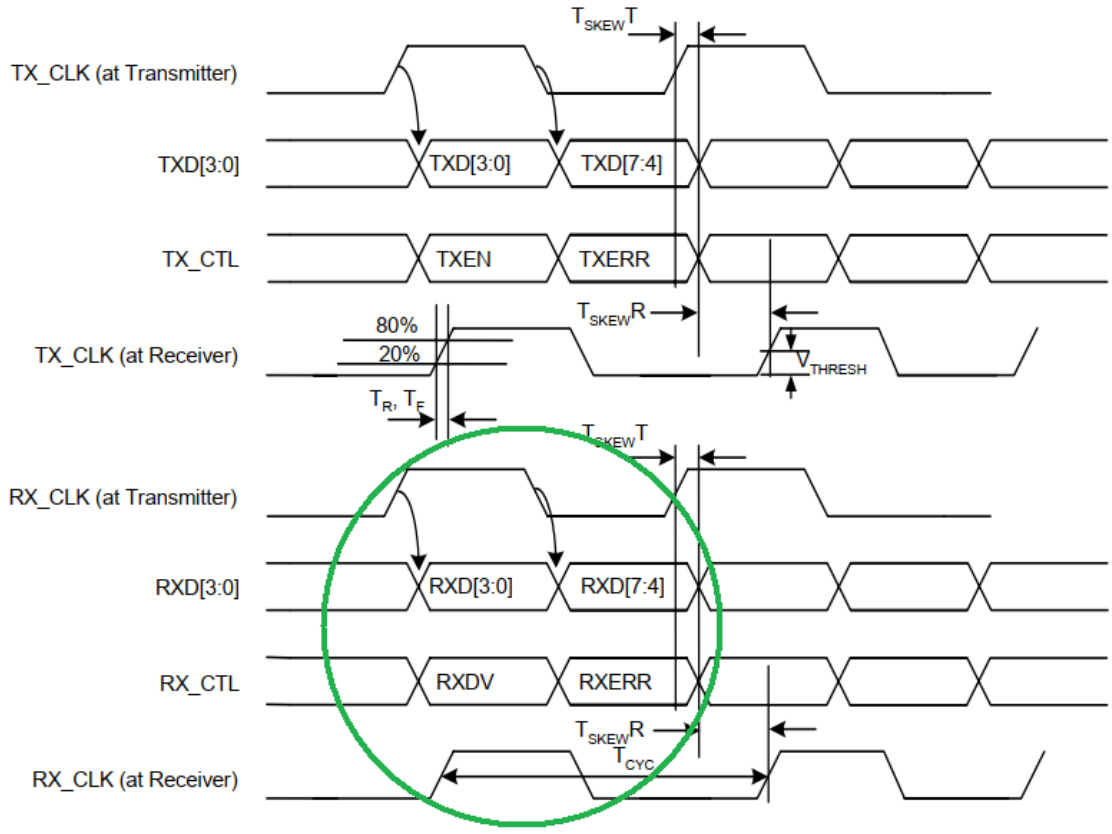


这里只要控制NRESET的电平变化即可，如上图所以，要求其拉低4ms以上再拉高。

2. IDDR原语配置

（1）无补偿工作时序

Figure 21. RGMII Uncompensated Timing



对于接收端来说，PHY接收到的数据是先发送低4位，再发送高4位，顺序是颠倒的，而且还是上升沿和下降沿沿通过输出数据，对于FPGA来说，无法直接实现双沿接收，这就需要将其改变为单沿触发，便于FPGA的实现，这就是下面所说的IDDR。

（2）IDDR实现

具体来说，IDDR就是将双沿触发的数据变为单沿触发，可以将2bit的双沿数据分别从两个端口输出，具体可以参考Vavado的template 直接调用。

从上面的时序图，我们可以发现这里有4bit的数据输出和1bit的RX_CTL输出。这样的话，总共需要5次原语的调用以解决这个问题。

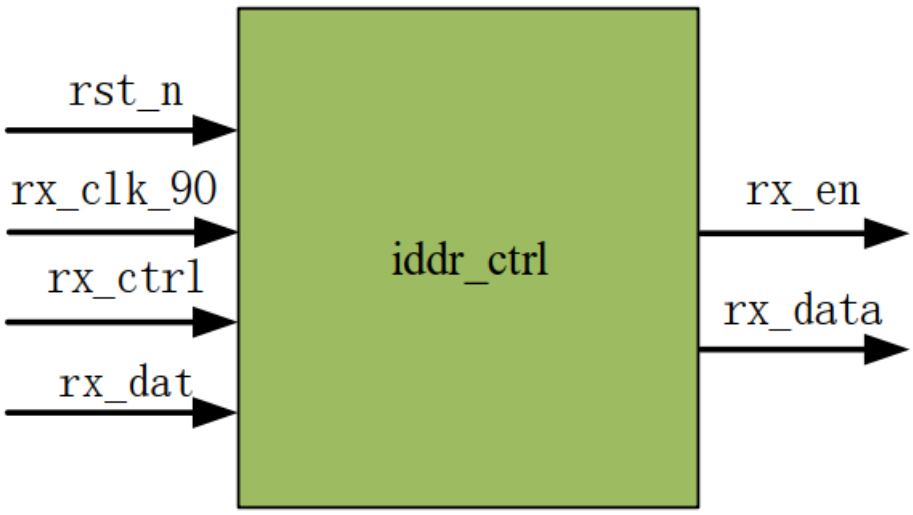
四、练习步骤

1.verilog代码实现

（1）PHY芯片复位

NRESET引脚拉低4ms后拉高

（2）IDDR原语调用



2. ILA抓取

抓取信号，看是否能否接收到0x55和0xD5两个数据。

3.查看结果

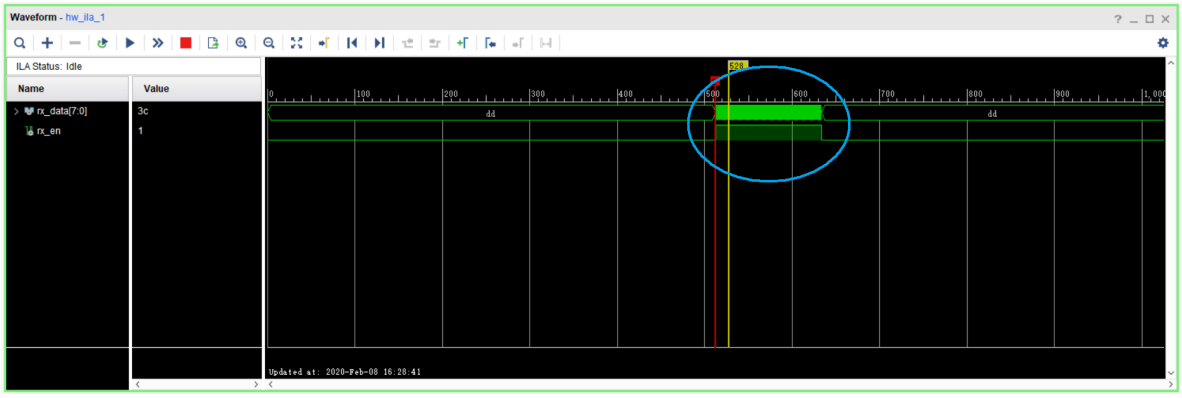
PHY芯片初始化成功后，可以通过电脑的连接端看到正确1Gbit的网络连接。

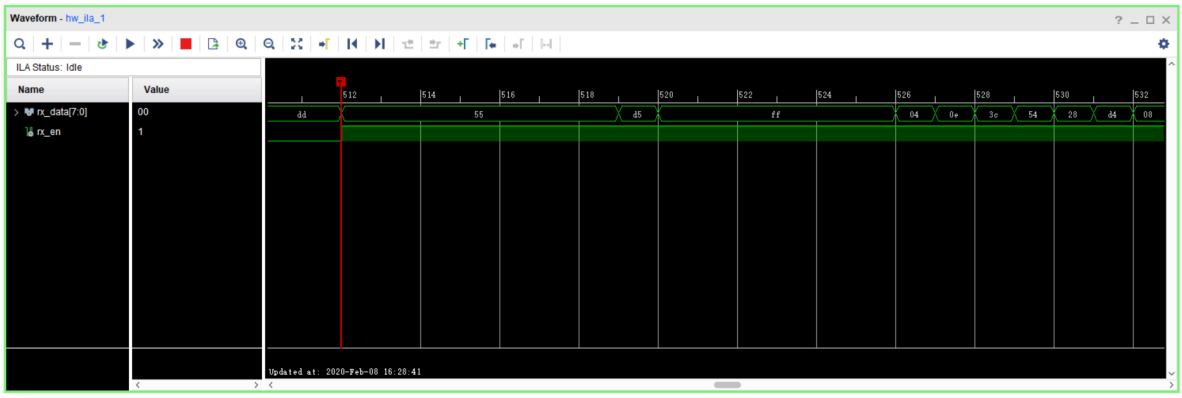
可以看到正确连通后，板卡和电脑的网卡协商后，达到了1Gbps，这个速度还是挺快的。

五、实际波形仿真

以rx_en的上升沿作为触发条件，观察rx_data的数据情况。

这里主要从两个方面进行查看，一是rx_en完全包裹了rx_data的所有数据范围；





二是每次数据的开始都是7个字节的0x55后，紧跟着0xD5，这是一帧数据包的帧头。

这两部分都有了，那么本次的实验也算成功啦。

六、总结与讨论

1.phy_rst_n的简单的实现方式

```
always @(posedge clk)
begin
    if (rst == 1'b1)
        phy_rst_cnt <= 'd0;
    else if (phy_rst_cnt [18] == 1'b0)
        phy_rst_cnt <= phy_rst_cnt + 1'b1;
end
assign phy_rst_n = phy_rst_cnt[18];
```

这种方式不需要两次always代码块，一次就行了，而且很好了利用了时序逻辑和组合逻辑的特点，值得推广。

2. phy双沿触发转换为单沿触发的注意事项

- 需要特别注意rx_ctrl也需要经过此类变化，不然也是会出问题的
- 不论是模块和原语，输入必须要有信号，虽然输出可以没有，但这的确值得注意。
- 虽然在硬件电路上，rx_clk已经现对与tx_clk有2us的偏差，即刚好在数据的中间有效位置，但是由于受到电路和内部总线等的影响，最后输出的rx_clk相对于数据有一定的偏差。这样的话，在rx_clk增加一定的相移就很有必要啦。

3.phy_rst_n信号输出的时间问题

前面要求是需要4ms，实际上根本不需要这么长，4ms是软复位所需的时间，实际外面的硬件引脚达到100ns就够啦。

Table 73. AC Characteristics for Device Reset

Parameter	Symbol	Minimum	Maximum	Unit	Condition
NRESET assertion time	t_{RESET}	100		ns	
Wait time between NRESET de-assert and access of the SMI interface	t_{WAIT}	20		ms	Register 21E.14 = 0
		220		ms	Register 21E.14 = 1

Table 73. AC Characteristics for Device Reset *(continued)*

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Soft reset (pin) assertion	t_{SRESET_ASSERT}	4		ms	
Soft reset (pin) de-assertion	$t_{SRESET\_DEASSERT}$	4		ms	
Reset rise time	t_{RST_RISE}	0 25		ms ms	If REG_EN pin = 0 If REG_EN pin = 1 Measured from a 10% level to a 90% level
Supply stable time	$t_{VDDSTABLE}$	10		ms	
Wait time between soft reset pin de-assert and access of the SMI interface	t_{SWAIT}	4 300 200 200		µs µs ms ms	Registers 28.1 = 1, 21E.14 = 0 Registers 28.1 = 0, 21E.14 = 0 Registers 28.1 = 0, 21E.14 = 1 Registers 28.1 = 1, 21E.14 = 1
Soft reset MII register 0.15 assertion	t_{SREG_RESET}	100		ns	
Wait time between Soft Reset (MII Register 0.15) de-assert and access to the SMI interface	t_{SREG_WAIT}	4 300 200 200		µs µs ms ms	Registers 18.1 = 1, 21E.14 = 0 Registers 18.1 = 0, 21E.14 = 0 Registers 18.1 = 0, 21E.14 = 1 Registers 18.1 = 1, 21E.14 = 1