

操作系统网络框架

萧络元

清华大学

网络模型

- 操作系统
 - Linux Socket (BSD Socket)
 - zCore Socket



目录

一、网卡

二、RJ-45接口

三、隔离变压器

四、PHY

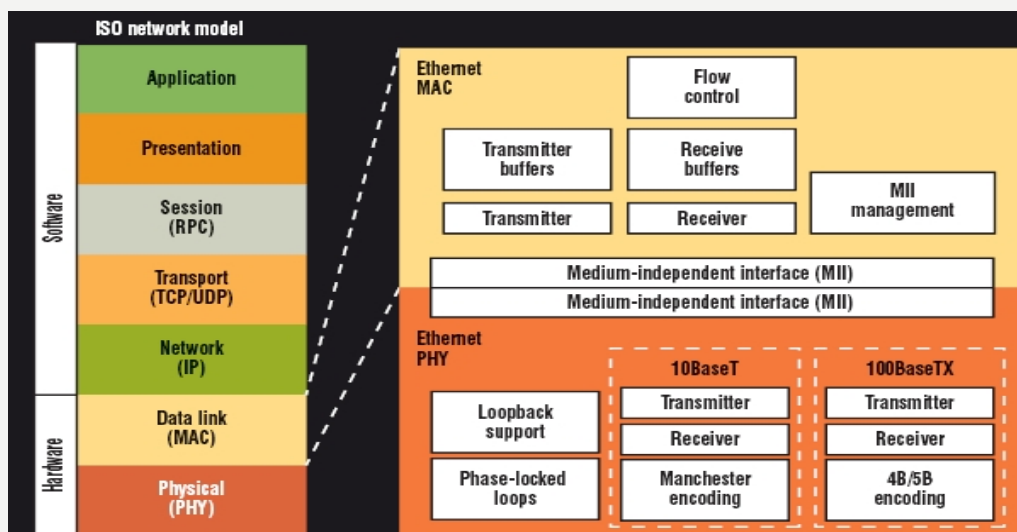
五、MAC

六、MII与RMII

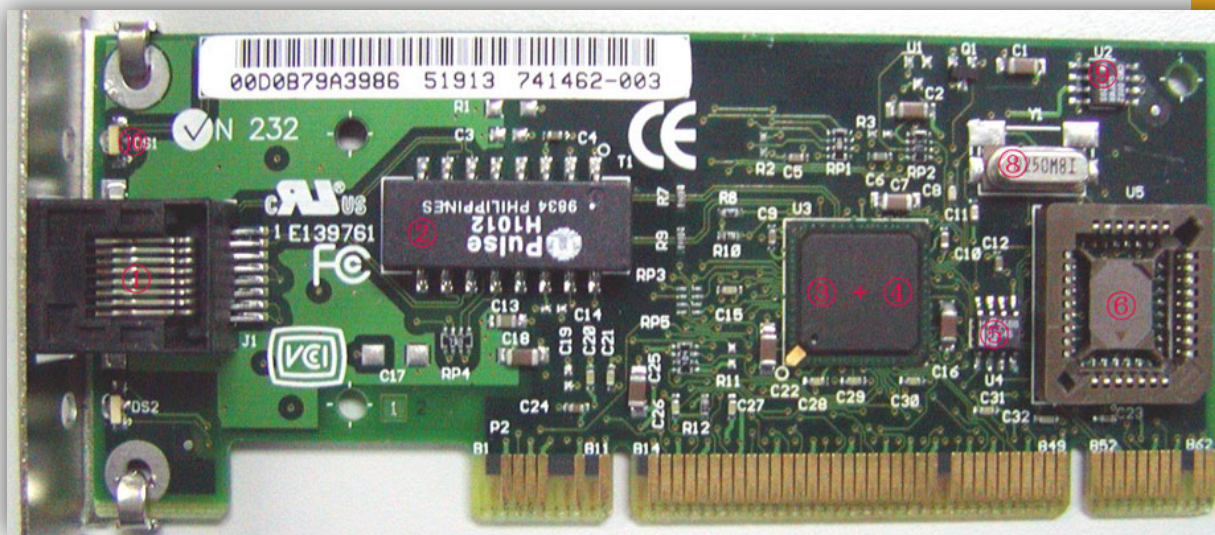
七、总结

网卡

- 网卡工作在ISO/OSI模型的最底下两层：物理层(PHY)和数据链路层(MAC)。
- 物理层定义了数据传送与接收所需要的电气信号、线路状态、时钟基准、数据编码和电路等，并向数据链路层设备提供标准接口。工作在该层的芯片称为网络PHY。
- 数据链路层则提供寻址机制、数据帧的构建、数据差错检查、传送控制等功能，并向网络层提供标准接口。工作在该层的芯片称为MAC控制器。



1. The Ethernet MAC and PHY implement the bottom two layers of the International Organization for Standardization/Open System Interconnect (ISO/OSI) stack. The MAC interfaces with the PHY through an MII. The typical 10/100 PHY Ethernet implementation incorporates separate 10BaseT and 100BaseTX interfaces.



RJ-45接口

网线插头

RJ45型网线插头又称“水晶头”，含有8根线。

RJ45型插头和网线有两种连接方法，分别称作T568A线序和T568B线序。

（右上图展示了两种线序对应的差分线的绕法，下图展示了两种线序的接头）

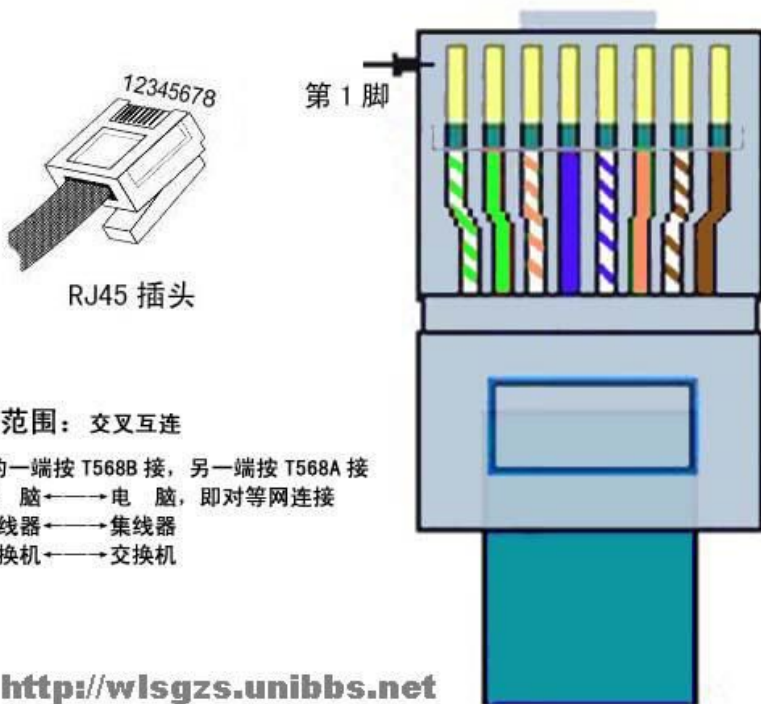
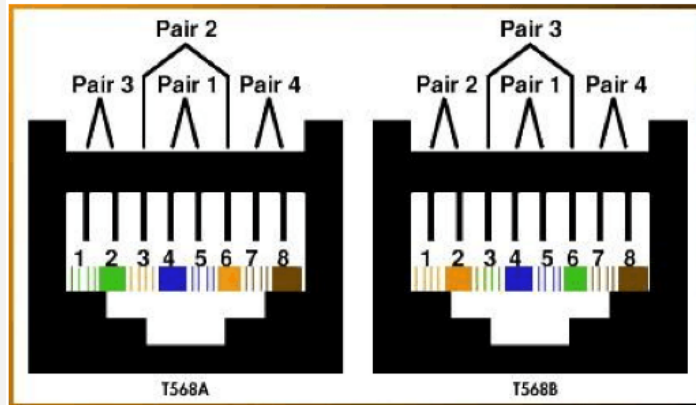


图 1：RJ45 型网线插头的 T568A 线序

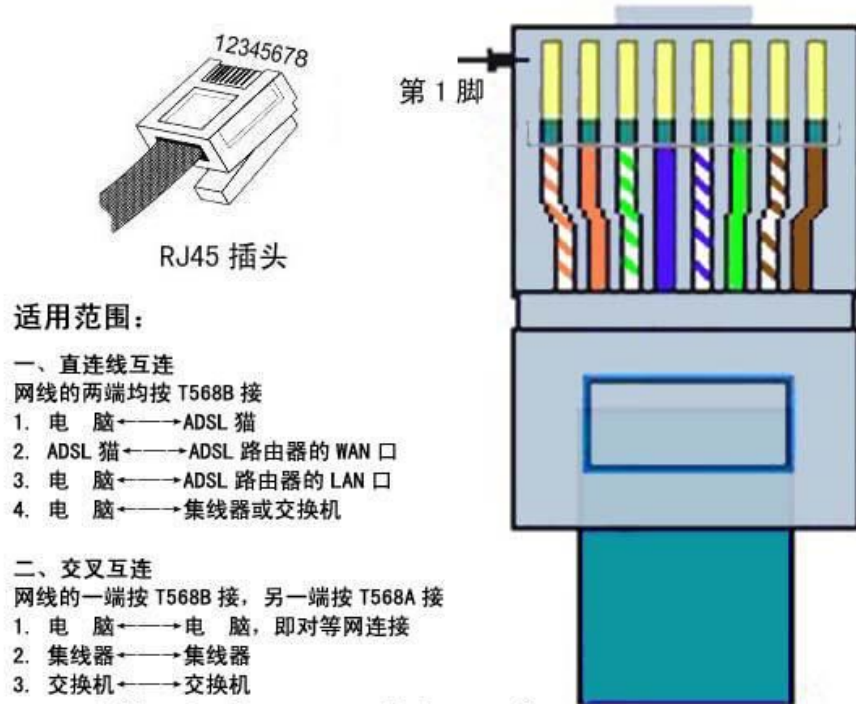
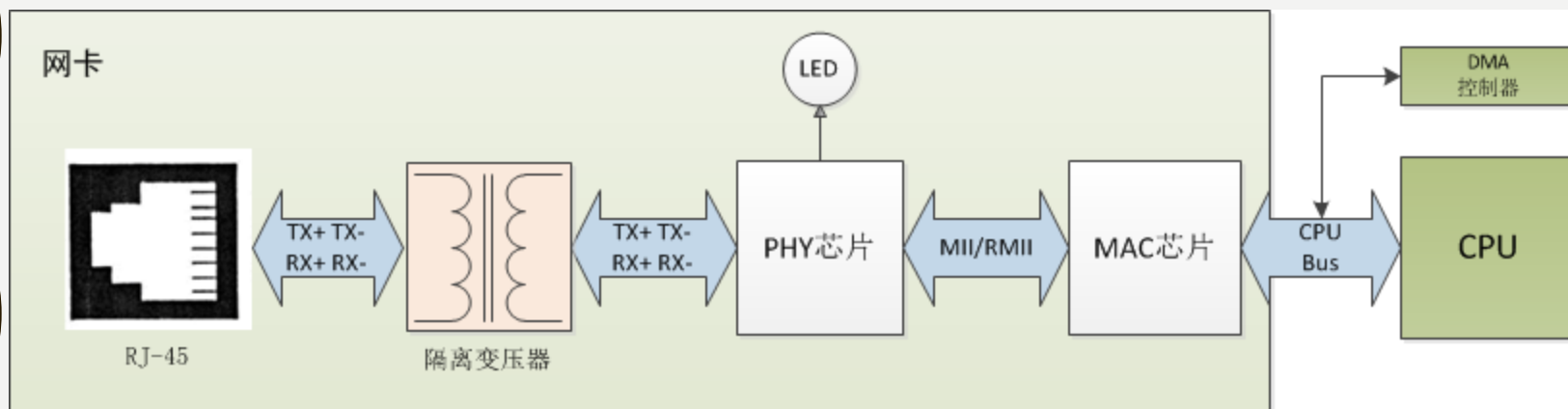


图 2：RJ45 型网线插头的 T568B 线序

网卡

网卡框图



图中各部分不一定都是独立的芯片。

根据组合形式，可分为下列几种类型：

- CPU集成MAC与PHY
- CPU集成MAC，PHY采用独立芯片
- CPU不集成MAC与PHY，MAC与PHY均采用集成芯片

[注] DMA控制器通常属于CPU的一部分，这里单独画出是为了表示DMA控制器可能会参与到网口数据传输中。

网卡MAC和PHY

问：网卡的MAC和PHY之间的关系？

答：网卡工作在ISO/OSI的最下两层：物理层和数据链路层。

物理层定义了数据传送与接收所需要的电与光信号、线路状态、时钟基准、数据编码和电路等，并向数据链路层设备提供标准接口。物理层的芯片称之为PHY。

数据链路层则提供寻址机构、数据帧的构建、数据差错检查、传送控制、向网络层提供标准的数据接口等功能。以太网卡中数据链路层的芯片称之为MAC控制器。

很多网卡的这两个部分是做到一起的，他们之间的关系是PCI总线接MAC，MAC接PHY，PHY接网线（当然也不是直接接上的，还有一个隔离变压器）。

问：PHY和MAC之间是如何传送数据和相互沟通的？

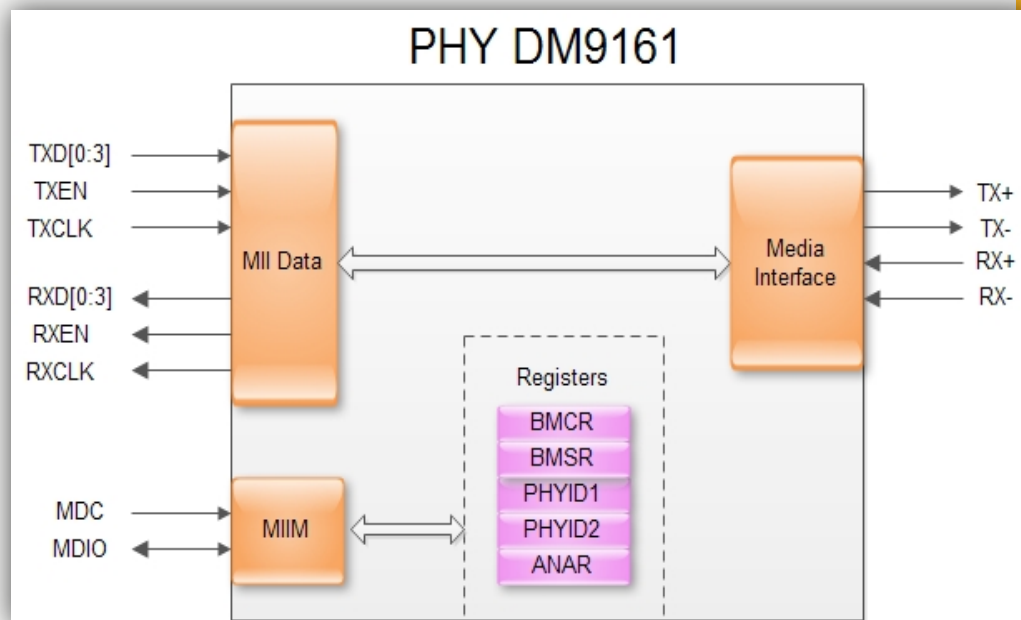
答：通过IEEE定义的标准MII/GigaMII (Media Independent Interface, 介质独立接口)。MII接口传递了网络的所有数据和控制信息。ETHERNET的接口实质是MAC通过MII总线控制PHY的过程。

PHY

- PHY (Physical) 是物理接口收发器，位于隔离变压器和MAC芯片之间。
- PHY工作在ISO/OSI七层模型的最底层（物理层）。
- PHY的功能：数据收发、串并行转换、物理层编解码等。
- PHY还实现了CSMA/CD的部分功能。

CSMA/CD（带冲突检测的载波监听多路访问技术），是一种争用型介质访问控制协议。“先听再发，边发边听，冲突停发，随机延迟后重发。”

- PHY提供了和对端设备连接的重要功能——自动协商 (AutoNegotiation或NWAY)，以确定双方支持的最大速度和双工模式。
- PHY可以连接LED灯显示出目前的连接状态。

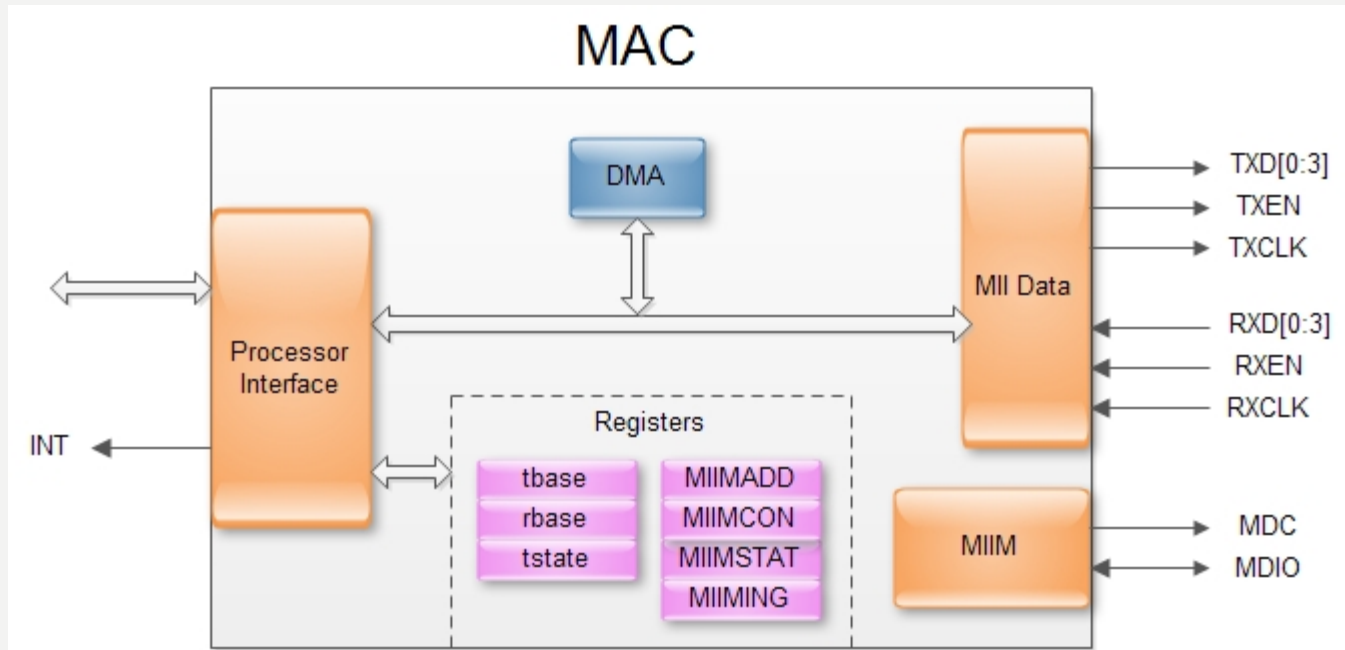


MAC

- MAC (Media Access Control) , 即介质访问控制子层。它工作在ISO/OSI七层模型的数据链路层的下半部分。
- MAC芯片位于PHY和CPU之间, 一端通过MII/RMII接口连接PHY芯片, 一端通过PCI等接口连接CPU。
- 以太网数据链路层其实包含MAC (介质访问控制) 子层和LLC (逻辑链路控制) 子层。但对于以太网卡的MAC芯片来说, 它不但要实现MAC子层和LLC子层的功能, 还要提供符合规范的PCI界面以实现和主机的数据交换。
- MAC的功能: 数据收发、打包解包……
- 发送数据时, MAC协议先判断是否可以发送, 如果可以则给数据加上控制信息, 以规定的格式发送到物理层 (PHY) ; 接收数据时, MAC协议先判断是否有传输错误, 如果没有则去掉控制信息上传至LLC层。
- MAC芯片从PCI总线收到数据包后, 将之拆分并重新打包成帧 (每帧64Byte~ 1518Byte , 包含目标MAC地址、自己的MAC地址、数据包协议类型、CRC码) 。
- 还有一些控制帧 (如流控帧) 也需要MAC直接识别并执行相应的行为。
- 有些MAC芯片是内置在CPU/MCU内部的, 无需外接。

MAC

- MAC芯片接口框图：



MAC地址

- MAC地址简介：
 - 在网络底层的物理传输过程中，设备是通过MAC地址来相互识别的。因此MAC地址一定是全球唯一的。
 - MAC地址长度为6个字节（48bit），按组合的个数来看一共有 2^{48} 个。换句话说，MAC地址个数也是有限的，只是现时还不用考虑耗尽的问题。
 - MAC地址前3个字节称为组织唯一标志符(OUI)，由IEEE分配给网络设备生产厂商。后3个字节称为扩展标志符，由厂家分配给产品。

MAC地址格式

OUI	厂家自己分配
00-12-13	34-20-19

一个厂家可以拥有多个OUI

MAC地址

- MAC地址与IP地址：
 - IP地址负责表示计算机的**网络层**地址，网络层设备（如路由器）根据IP地址来进行操作；
 - MAC地址负责表示计算机的**数据链路层**地址，数据链路层设备（如交换机）根据MAC地址来进行操作。
 - IP和MAC地址这种映像关系由**ARP**（Address Resolution Protocol，地址解析协议）协议完成。（实际在smoltcp中处理）

网卡结构

- 网卡结构图（内部）：

MAC集成在CPU中。PHY芯片通过MII接口与CPU上的MAC连接。

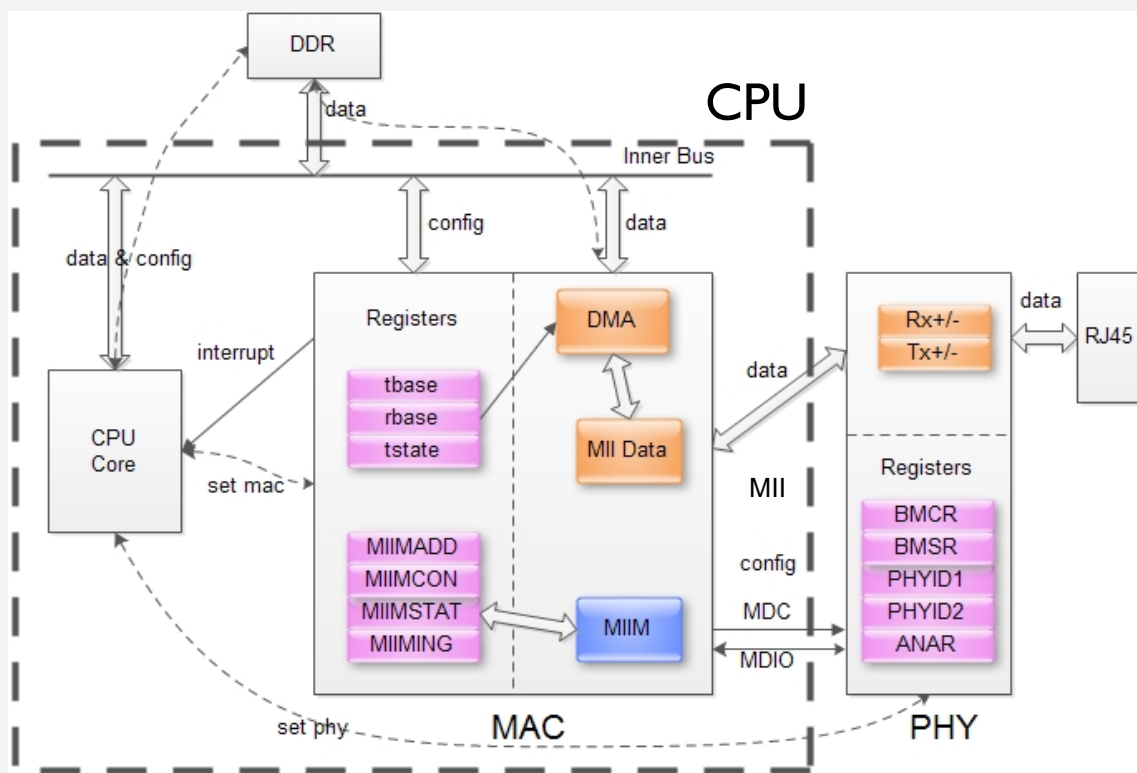
- 软件对网口的操作步骤：

为数据收发分配内存，

初始化MAC寄存器，

初始化PHY寄存器，

启动收发。



MII与RMII

1. MII

- MII (Media Independent Interface)，媒体独立接口，也叫介质无关接口。它是IEEE-802.3定义的以太网行业标准。
- 用于PHY与MAC之间的通信。它包括一个数据接口和一个管理接口。

类型	信号名称	数量	Direction	Description	Belong To
数据接口	TX_CLK	1	Input	Transmit Clock	MAC 层到物理层的发送数据接口
	TXD[0:3]	4	Output	Transmit Data	
	TX_ER	1	Output	Transmit Coding Error	
	TX_EN	1	Output	Transmit Enable	
	RX_CLK	1	Input	Receive Clock	物理层到MAC 层的接收数据接口
	RXD[0:3]	4	Input	Receive Data	
	RX_ER	1	Input	Receive Error	
	RX_DV	1	Input	Receive Data Valid	
	COL	1	Input	Collision Detected	物理层到 MAC 层状态指示信号
	CRS	1	Input	Carrier Sense	
	Total bus Width	16	-----	-----	-----
管理接口	MDC	1	Output	Management Clock	MAC 和物理层控制和状态信息
	MDIO	1	I/O	Management Data I/O	

MII与RMII

- zCore的网卡初始化中，根据RGMII的phy模式来初始化时钟，以匹配相应的网卡速率

```
// clk_prepare_enable()

let mut clk_value: u32 = read_volatile((self.base_phy + EMAC_EPHY_CLK_REG0) as *mut u32);
info!("clk enable, Read PHY CLK: {:#x}", clk_value);
// RGMII接口, 支持10/100/1000 Mbps速率
if self.phy_mode == RGMII {
    clk_value |= 0x00000004; // set RGMII
} else {
    clk_value &= !0x00000004;
}

clk_value &= !0x00002003; // clear RMII_EN, ETCS

if (self.phy_mode == RGMII) || (self.phy_mode == GMII) {
    clk_value |= 0x00000002; // set ETCS=2

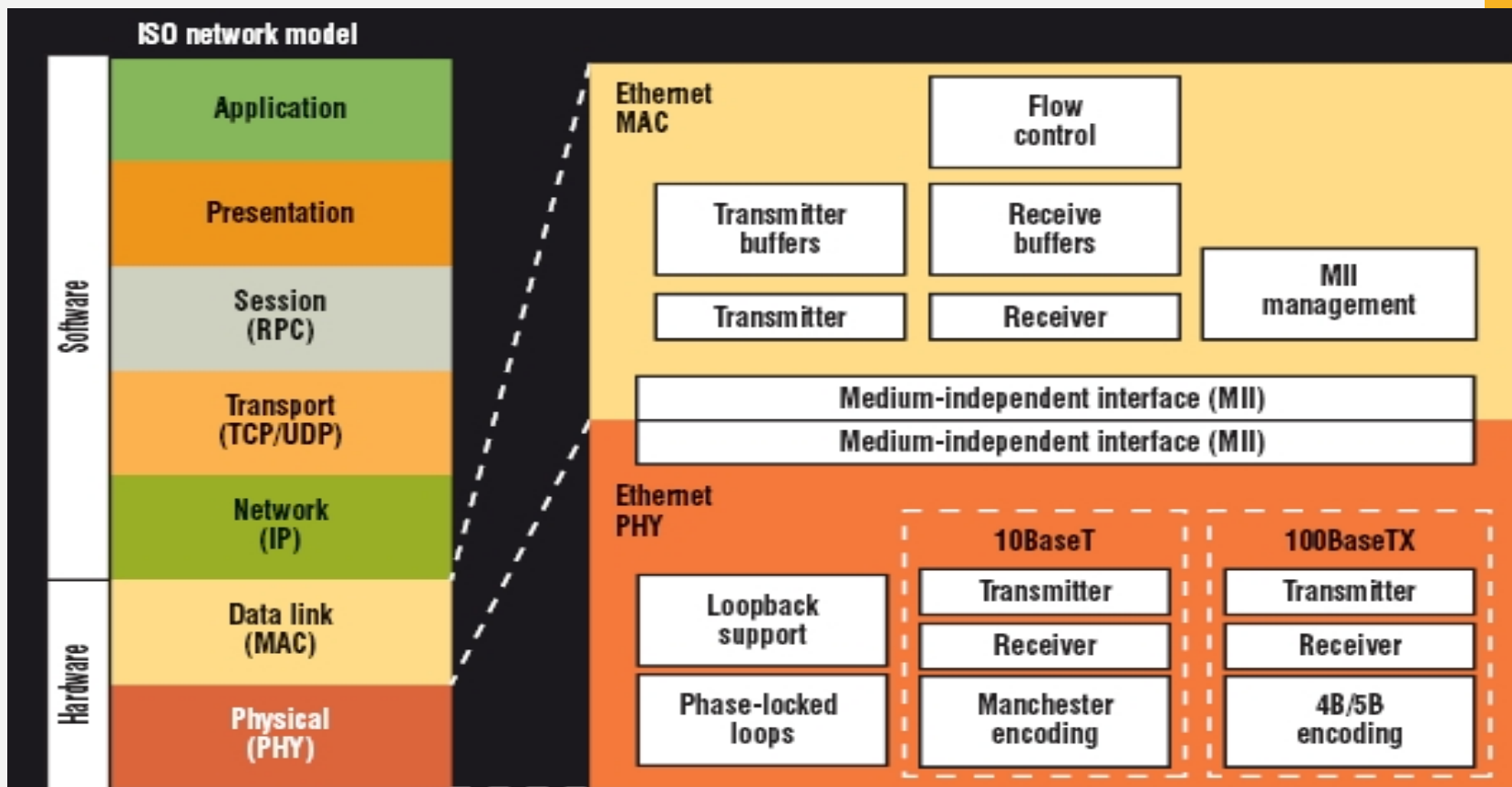
    // RMII接口, 支持10/100 Mbps速率
} else if self.phy_mode == RMII {
    clk_value |= 0x00002001;
}

// Adjust Tx/Rx clock delay
clk_value &= !(0x07 << 10);
clk_value |= (self.tx_delay & 0x07) << 10;
clk_value &= !(0x1F << 5);
clk_value |= (self.rx_delay & 0x1F) << 5;

info!("clk enable, write clk_value: {:#x}", clk_value);
```

网卡总结

- PHY和MAC在ISO/OSI七层模型中的位置：



1. The Ethernet MAC and PHY implement the bottom two layers of the International Organization for Standardization/Open System Interconnect (ISO/OSI) stack. The MAC interfaces with the PHY through an MII. The typical 10/100 PHY Ethernet implementation incorporates separate 10BaseT and 100BaseTX interfaces.

网卡总结

- 创建网卡驱动对象, 准备好接受发送Buffer, 对外提供收发及中断处理接口;
- 初始化驱动之前通过pinctrl设置GPIO引脚;
- 设置网卡时钟;
- 初始化mdio, 即RGMII操作接口;
- 初始化网卡寄存器, 设置配置参数, 匹配网络速率等
- 整个过程比较繁琐, 对照datasheet来操作;



The End