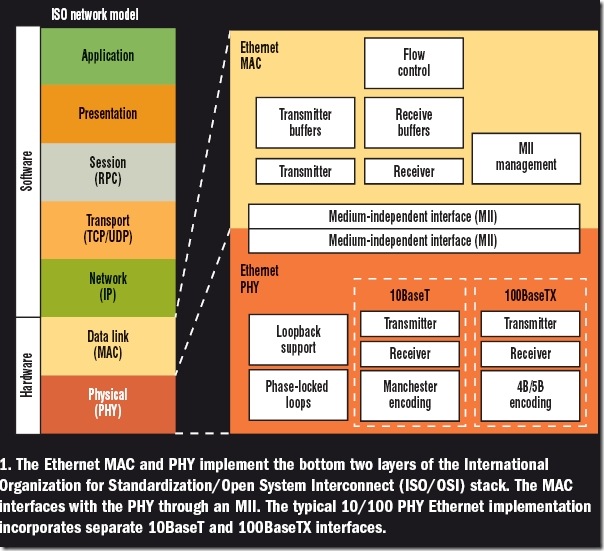
此文章已于 19:30:09 2016/6/23 重新发布到 MADHEX

以太网PHY和MAC

以太网PHY和MAC对应OSI模型的两个层——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（RGMII / GMII / MII）。

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。

[](http://images.cnblogs.com/cnblogs_com/shengansong/201209/20120901002711977.jpg)

**问：以太网PHY是什么？**

答：PHY是物理接口收发器，它实现物理层。IEEE-802.3标准定义了以太网PHY。它符合IEEE-802.3k中用于10BaseT(第14条)和100BaseTX(第24条和第25条)的规范。

**问：以太网MAC是什么？**

答：MAC就是媒体接入控制器。以太网MAC由IEEE-802.3以太网标准定义。它实现了一个数据链路层。最新的MAC同时支持10/100/1000Mbps速率。通常情况下，它实现MII/GMII/RGMII接口，来同行业标准PHY器件实现接口。

**问：什么是MII？**

答：MII（Medium Independent Interface）即媒体独立接口。它是IEEE-802.3定义的以太网行业标准。它包括一个数据接口，以及一个MAC和PHY之间的管理接口。数据接口包括分别用于发送器和接收器的两条独立信道。每条信道都有自己的数据、时钟和控制信号。MII数据接口总共需要16个信号。管理接口是个双信号接口：一个是时钟信号，另一个是数据信号。通过管理接口，上层能监视和控制PHY。

MII标准接口 用于连快Fast Ethernet MAC-block与PHY。“介质无关”表明在不对MAC硬件重新设计或替换的情况下，任何类型的PHY设备都可以正常工作。在其他速率下工作的与 MII等效的接口有：AUI（10M 以太网）、GMII（Gigabit 以太网）和XAUI（10-Gigabit 以太网）。

此外还有RMII(Reduced MII)、GMII(Gigabit MII)、RGMII（Reduced GMII）SMII等。所有的这些接口都从MII而来，MII是(Medium Independent Interface）的意思，是指不用考虑媒体是铜轴、光纤、电缆等，因为这些媒体处理的相关工作都有PHY或者叫做MAC的芯片完成。   
MII支持10兆和100兆的操作，一个接口由14根线组成，它的支持还是比较灵活的，但是有一个缺点是因为它一个端口用的信号线太多，如果一个8端口的交换机要用到112根线，16端口就要用到224根线，到 32端口的话就要用到448根线，一般按照这个接口做交换机，是不太现实的，所以现代的交换机的制作都会用到其它的一些从MII简化出来的标准，比如 RMII、SMII、GMII等。   
RMII是简化的MII接口，在数据的收发上它比MII接口少了一倍的信号线，所以它一般要求是50兆的总线时钟。RMII一般用在多端口的交换机，它不是每个端口安排收、发两个时钟，而是所有的数据端口公用一个时钟用于所有端口的收发，这里就节省了不少的端口数目。RMII的一个端口要求7个数据线，比MII少了一倍，所以交换机能够接入多一倍数据的端口。和 MII一样，RMII支持10兆和100兆的总线接口速度。   
SMII是由思科提出的一种媒体接口，它有比RMII更少的信号线数目，S表示串行的意思。因为它只用一根信号线传送发送数据，一根信号线传输接受数据，所以在时钟上为了满足100的需求，它的时钟频率很高，达到了125兆，为什么用125兆，是因为数据线里面会传送一些控制信息。SMII一个端口仅用4根信号线完成100信号的传输，比起RMII差不多又少了一倍的信号线。SMII在工业界的支持力度是很高的。同理，所有端口的数据收发都公用同一个外部的125M时钟。   
GMII是千兆网的MII接口，这个也有相应的RGMII接口，表示简化了的GMII接口。   
  
  
**MII总线**在IEEE802.3中规定的MII总线是一种用于将不同类型的PHY与相同网络控制器（MAC）相连接的通用总线。网络控制器可以用同样的硬件接口与任何PHY 。

**GMII (Gigabit MII)**GMII采用8位接口数据，工作时钟125MHz，因此传输速率可达1000Mbps。同时兼容MII所规定的10/100 Mbps工作方式。   
GMII接口数据结构符合IEEE以太网标准。该接口定义见IEEE 802.3-2000。   
**发送器**：   
◇ GTXCLK——吉比特TX..信号的时钟信号（125MHz）   
◇ TXCLK——10/100M信号时钟   
◇ TXD[7..0]——被发送数据   
◇ TXEN——发送器使能信号   
◇ TXER——发送器错误（用于破坏一个数据包）   
注：在千兆速率下，向PHY提供GTXCLK信号，TXD、TXEN、TXER信号与此时钟信号同步。否则，在10/100M速率下，PHY提供TXCLK时钟信号，其它信号与此信号同步。其工作频率为25MHz（100M网络）或2.5MHz（10M网络）。   
**接收器：**   
◇ RXCLK——接收时钟信号（从收到的数据中提取，因此与GTXCLK无关联）   
◇ RXD[7..0]——接收数据   
◇ RXDV——接收数据有效指示   
◇ RXER——接收数据出错指示   
◇ COL——冲突检测（仅用于半双工状态）   
**管理配置**◇ MDC——配置接口时钟   
◇ MDIO——配置接口I/O   
管理配置接口控制PHY的特性。该接口有32个寄存器地址，每个地址16位。其中前16个已经在“IEEE 802.3,2000-22.2.4 Management Functions”中规定了用途，其余的则由各器件自己指定。

**RMII: Reduced Media Independant Interface**   
简化媒体独立接口   
是标准的以太网接口之一，比MII有更少的I/O传输。

关于RMII口和MII口的问题

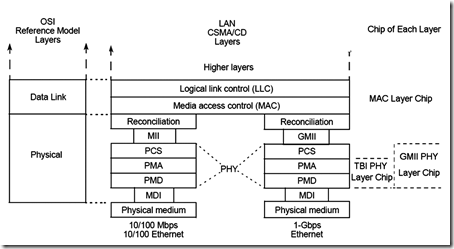
RMII口是用两根线来传输数据的，   
MII口是用4根线来传输数据的，   
GMII是用8根线来传输数据的。

MII/RMII只是一种接口，对于10M线速,MII的速率是2.5M，RMII则是5M；对于100M线速，MII的速率是25M，RMII则是50M。   
  
MII/RMII用于传输以太网包，在MII/RMII接口是4/2bit的，在以太网的PHY里需要做串并转换、编解码等才能在双绞线和光纤上进行传 输，其帧格式遵循IEEE 802.3(10M)/IEEE 802.3u(100M)/IEEE 802.1q(VLAN)。

以太网帧的格式为：前导符+开始位+目的mac地址+源mac地址+类型/长度+数据+padding(optional)+32bitCRC   
如果有vlan，则要在类型/长度后面加上2个字节的vlan tag，其中12bit来表示vlan id，另外4bit表示数据的优先级！

**吉比特以太网物理层协议及接口**

吉比特以太网协议的数据链路层与传统的10/100Mb/s以太网协议相同，但物理层有所不同。三种协议与OSI七层模型的对应关系如图所示。

[](http://images.cnblogs.com/cnblogs_com/shengansong/201209/201209010027151694.gif)

从图可以看出，吉比特以太网协议与10/100Mb /s以太网协议的差别仅仅在于物理层。图中的PHY表示实现物理层协议的芯片；协调子层（Reconciliation sublayer）用于实现指令转换；MII（介质无关接口）／GMII（吉比特介质无关接口）是物理层芯片与实现上层协议的芯片的接口；MDI（介质相关接口）是物理层芯片与物理介质的接口；PCS、PMA和PMD则分别表示实现物理层协议的各子层。在实际应用系统中，这些子层的操作细节将全部由PHY 芯片实现，只需对MII和MDI接口进行设计与操作即可。

吉比特以太网的物理层接口标准主要有四种：GMII、 RGMII（Reduced GMII）、TBI（Ten-Bit Interface）和RTBI（Reduced TBI）。GMII是标准的吉比特以太网接口，它位于MAC层与物理层之间。对于TBI接口，图1中PCS子层的功能将由MAC层芯片实现，在降低PHY 芯片复杂度的同时，控制线也比GMII接口少。RGMII和RTBI两种接口使每根数据线上的传输速率加倍，数据线数目减半。

**网卡**

PHY和MAC是网卡的主要组成部分，网卡一般用 RJ-45插口，10M网卡的RJ-45插口也只用了1、2、3、6四根针，而100M或1000M网卡的则是八根针都是全的。除此以外，还需要其它元件，因为虽然PHY提供绝大多数模拟支持，但在一个典型实现中，仍需外接6、7只分立元件及一个局域网绝缘模块。绝缘模块一般采用一个1：1的变压器。这些部件的主要功能是为了保护PHY免遭由于电气失误而引起的损坏。

网卡的功能主要有两个:一是将电脑的数据封装为帧，并通过网线(对无线网络来说就是电磁波)将数据发送到网络上去;二是接收网络上其它设备传过来的帧，并将帧重新组合成数据，发送到所在的电脑中。网卡能接收所有在网络上传输的信号，但正常情况下只接受发送到该电脑的帧和广播帧，将其余的帧丢弃。然后，传送到系统CPU做进一步处理。当电脑发送数据时，网卡等待合适的时间将分组插入到数据流中。接收系统通知电脑消息是否完整地到达，如果出现问题，将要求对方重新发送。