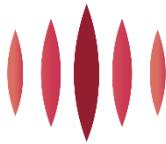


NOVEMBER 26, 2020



裕太微电子
Motorcomm

YT8512H YT8512C 应用说明

单口 10/100 以太网物理层收发器

版本 2.3

裕太微电子股份有限公司

苏州市高新区科技城科灵路 78 号 4 号楼 201 室

上海市浦东新区盛夏路 570 号 201 室

www.motor-comm.com



裕太微电子
Motorcomm

版权声明

Copyright Statement

本文档版权归苏州裕太微电子有限公司（以下简称“裕太微电子”）所有，并保留一切权利。未经裕太微电子书面许可，任何公司和个人不得将此文档中的任何部分复制、传播、披露或以其他方式散发给第三方。否则裕太微电子将保留追究其法律责任的权利。

This document is copyright of Suzhou Motorcomm Electronic Technology Co., Ltd. ("Motorcomm"). All rights reserved. No company or individual may copy, disseminate, disclose or otherwise distribute any part of this document to any third party without the written consent of Motorcomm. If any company or individual so does, Motorcomm reserves the right to hold it or him liable therefor.

免责声明

Disclaimer

本文档仅提供阶段性信息，所含内容将/可根据产品的实际情况随时更新，恕不另行通知。如因文档使用不当造成直接或间接损失，裕太微电子不承担任何责任。

This document only provides periodic information, and its contents will/may be updated from time to time according to actual situation of Motorcomm's products without further notice. Motorcomm will not take any responsibility for any direct or indirect losses caused due to improper use of this document.

Revision History

Revision	Release Date	Summary
1.5	2020.05.27	修改文档格式
1.6	2020.05.28	更正书写错误: "RGMII"改为"MI/REMII/RMII"; 更正 template 测试寄存器 0x27 改为 0x2027 更正 template10bte 测试寄存器 0x200a 的值
1.7	2020.06.15	更正 4.8.2.2: RMII1 的 TXC 应为输入, 并增加上电复位时序提醒;
1.8	2020.07.03	添加 sleep 功能介绍
1.9	2020.08.03	修改描述
2.0	2020.08.20	加入关于 phyaddr0 和 broadcast address 的说明。
2.1	2020.11.09	更正 4.8.2.2 章 RMII 外灌 50MHz 应用电路连接, 并加入更多说明 新增 4.9 章 电路设计时避免进入 SCAN 模式
2.2	2020.11.26	更正 4.8.2.2 章 RMII 外灌 50MHz 应用电路连接并建议通过 XTAL_IN 输入时钟
2.3	2024.01.03	更正 6.2.3 章 loopback 框图, 更正 6.4 章 LED1 寄存器

1. 简介	5
2. 缩略语	5
3. 系统框图	5
4. 硬件电路相关	6
4.1 电源	6
4.2 上电时序	6
4.3 时钟	8
4.4 RBIAS	8
4.5 POS	8
4.6 LED	9
4.7 SMI (MDIO) 接口	10
4.8 MII, REMII, RMII接口	10
4.8.1 MII与REMII模式	10
4.8.2 RMII1与RMII2模式	10
4.8.2.1 RMII1/2 (25MHz时钟)	11
4.8.2.2 RMII1/2 (50MHz时钟)	11
4.9 避免进入SCAN模式	12
4.10 MDI接口	12
4.11 硬复位后延时进行MDIO操作要求	13
4.12 外接参考时钟的抖动要求	13
4.13 外接参考时钟源切换操作要求	13
4.14 数据传输通路延时	13
5. 寄存器访问方式	14
5.1 寄存器的类型	14
5.2 通过MII寄存器访问其它类型寄存器	15
5.3 读写寄存器的某几位的表示方法	16
5.4 PHYADDR0和BDCST_ADDR的使用说明	16
6. 寄存器功能配置	17
6.1 初始化配置及不支持的功能	17
6.1.1 MII模式下百兆半双工模式不通	17
6.1.2 不插网线误连接到10BT	17
6.2 UTP (电口) 配置	17
6.2.1 10/100BT的速度和双工	17
6.2.2 power down设置	18
6.2.3 Loopback (回环) 模式	18
6.2.4 Smart downgrade (自动降速) 配置	20
6.2.5 Sleep (自动睡眠) 配置	20
6.2.6 包生成器和收发包统计	21
6.2.7 SNR读取 (当前连接质量指示)	21
6.2.8 Template (电口指标) 配置	22
6.3 MII/RMII配置	22
6.3.1 驱动能力的设置	22

6.4	LED灯配置.....	23
6.5	INTERRUPT（中断）配置	23
6.6	其它维测寄存器	24
6.6.1	用户自配置寄存器	24
6.6.2	PLL锁定指示	24
6.6.3	Clock drift	25

Motorcomm confidential for 裕太

1. 简介

YT8512 是一款单口、低功耗 10M/100M 以太网 PHY 收发器，支持 MII、RMII 两种接口与 MAC 相连。主要由电源、复位、时钟、数据接口、LED 输出等部分组成，其中数据接口包括了 xMII, SMI, MDI 三个模块。本文主要介绍硬件电路的一些注意事项和软件相关配置。

2. 缩略语

YT: 裕太微电子，代指本公司。

PHY: physical layer, 物理层芯片，本文用来代指以太网物理层芯片，即 YT8512 芯片。

MAC: Media Access Control, 数据链路层，本文用来代指与 PHY 相连的上层接口芯片，通常为集成 MAC 功能的 switch 芯片或 CPU 芯片。

DUT: Device under test, 被测对象，本文一般用来代指 YT8512 芯片。

LP: Link Partner, 对端相连芯片，本文一般用来代指与 DUT 通过网线，光纤或 SMA 等线相连接的对端芯片。

UTP: un-twisted pair, 非屏蔽双绞线，用来与其它 PHY 芯片相连的接口，本文用来代指以太网电口（或称 RJ45），包括 100/10Mbps 两种速率。

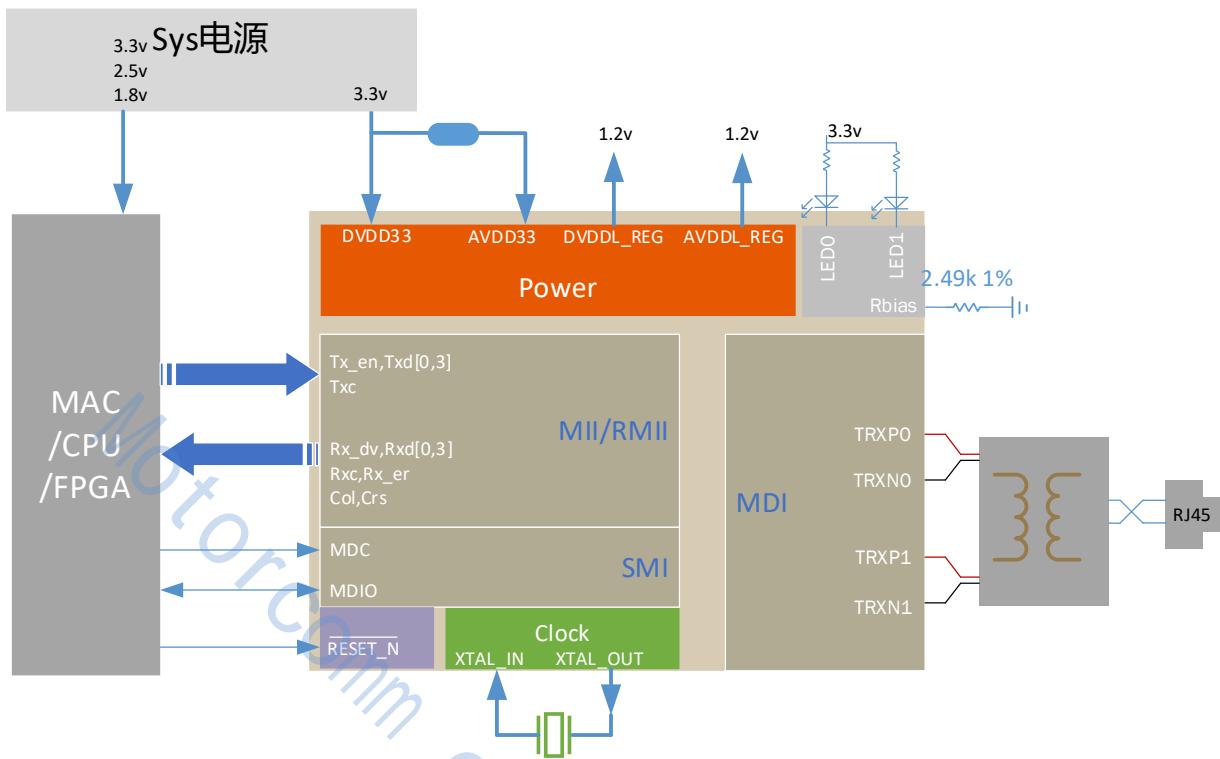
MDIO/SMI: Management Data Input Output, 是由 MDC/MDIO 组成的接口，用来读写 PHY 的内部寄存器。

Mii: Media independent interface, 以太网 PHY 芯片与 MAC 相连的接口，只适用于百兆/十兆速率。802.3 中规定的 PHY 标准寄存器命名为 MII 寄存器，所以本文中沿用称呼标准寄存器为 MII 寄存器，记为 mii_reg。

Ext: 由于 MII 寄存器数量有限，仅有 32 个。为满足更多功能需求，phy 芯片配置了更多的寄存器，称为扩展寄存器 extended register, 记为 ext_reg。

MMD: MDIO Managable Device, 通过 IEEE802.3 标准 Clause45 规定的 MDIO 接口进行管理的寄存器，称为 MMD 寄存器，记为 mmd_reg。

3. 系统框图



4. 硬件电路相关

4.1 电源

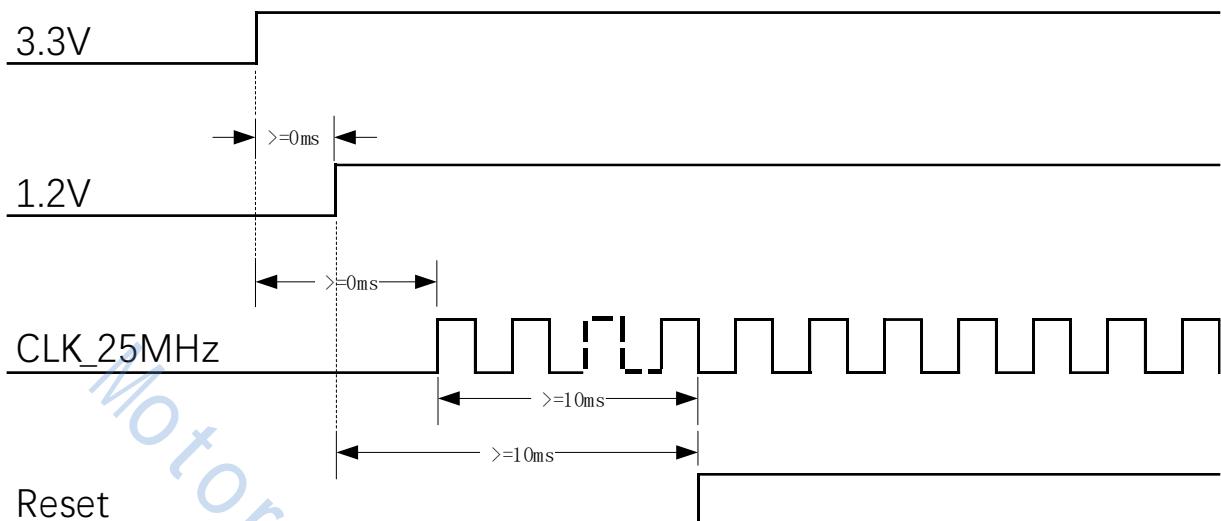
管脚 DVDD33 (pin14) 与 AVDD33 (pin7) 外接 3.3v@80mA 电压，分别向 YT8512 的数字部分与模拟部分供电，两者之间最好接一磁珠，且靠近管脚各自接 1uF 并 0.1uF 的电容。

DVDDL_REG (pin29) 与 AVDDL_REG (pin2) 是 YT8512 内部产生的分别向数字与模拟部分供 1.2v 电压的输出管脚。此两管脚在使用中，建议各自靠近管脚外接 1uF 并 0.1uF 电容。

DVDDL_REG 与 AVDDL_REG 仅供 YT8512 内部电路供电，不可对外给其他器件供电。

4.2 上电时序

RESET_N(pin21)实现硬件复位功能，低电平有效。当 RESET_N 开始动作，呈现低电平，此期间电源管脚电压与时钟信号从无到有，且保持稳定后，才可释放 RESET_N，一般 RESET_N 维持 10ms 低即可实现可靠地复位。执行硬复位后 YT8512 内寄存器值均恢复默认值。具体的时序关系如下：

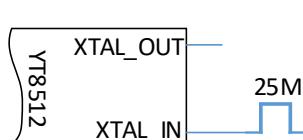


说明:

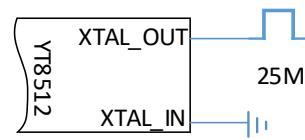
- 1.
2. 若 1.2V 或 CLK_25MHz 由芯片内部电源/晶体振荡电路产生，则它们与 3.3V 的时序由芯片内部控制，满足芯片内部时序要求。其中可能由于 3.3V 上升慢，而 1.2V 上升快，导致 1.2V 比 3.3V 先稳定，这是正常现象，无问题。或者 CLK_25MHz 在 3.3 上电未稳定时即有输出也是正常现象。
3. 若 1.2V 或 CLK_25MHz 由外灌，则需遵守上述时序。其中 1.2V 和 CLK_25MHz 并不要求先后，但要求 1.2V 和 CLK_25MHz 都稳定并保持至少 10ms 后再释放 Reset 信号。
4. 一般地，电源的爬坡时间至少需 $>100\mu\text{s}$ ，以防止芯片内部产生过压。

4.3 时钟

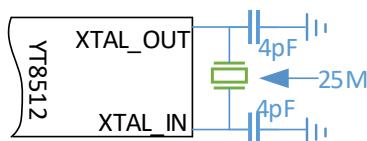
YT8512 可采用 25M 晶振，或外接 3.3v 的 25MHz 时钟信号。



XTAL_IN 接 25M 时钟源



XTAL_OUT 接 25M 时钟源



采用 25M 晶振

YT8512 亦可接 50MHz 时钟信号(将上图中的时钟源或晶振替换为 50MHz)。如果使用 50MHz 时钟，在系统上电后，需对以下寄存器做初始化设置：

```
Write_ext_reg0x50[6]: 1'b1 //将扩展寄存器 0x50 的 Bit6 置 1,选择时钟分频系数
```

```
Write_mii_reg0x0[15]: 1'b1 //基本寄存器 0x00 的 Bit15 置 1, 执行软复位操作
```

详细的 50MHz 时钟连接方式，请参考后叙的“RMII1/2(50MHz 时钟)”章节。

4.4 Rbias

Rbias(pin1)管脚需外接一 2.49k(1%精度)的电阻到地，YT8512 内部的部分电参数与此电阻相关。

4.5 POS

LED0、LED1、RX_DV、RXD1、RXD3 是复用管脚，除了正常状态下的功能外，在上电初始阶段，YT8512 检测此 5 位管脚的状态，据其对芯片做一些基本配置，这个过程称为 Power on strapping(POS)。

这 5 位管脚在 IC 内部均有弱下拉电阻。若外部不做配置，在 POS 阶段，默认是 0 状态。实际应用中，强烈建议外部接上拉或下拉 4.7k 电阻，不要依赖内部的弱上/下拉电阻。

LED0(pin24)\LED1(pin25):

LED0 与 LED1 在 POS 阶段的状态，确定了 YT8512 的 phy 地址

管脚名称_phy 地址位		phy 地址
Led1_addr1	Led0_addr0	
00		0
01		1
10		2
11		3

需要注意的是：同一个 MCU 或 MAC 管理多个 phy 时，不要使用 0 地址。由于 0 地址是广播地址，对地址为 0 的 phy 操作时，所有的 phy 都会响应。

RXD1(pin10):

POS 阶段 RXD1 的状态，决定了 LED0 管脚是作为 LED 还是 WOL 输出

RXD1—0: LED0 做为 LED 输出管脚

RXD1—1: LED0 作为 WOL 中断输出(open-drain 模式，只能 sink 电流)

RX_DV(pin8):

RX_DV—0: MII 模式

RX_DV—1: RMII 模式

RXD3(pin12):

RXD3—0: TXC 为输出

RXD3—1: TXC 为输入

4.6 LED

YT8512 有两个 LED 管脚输出，分别是 pin24—LED0, pin25---LED1。两管脚内部都有弱下拉电阻。

除了当做 LED 输出管脚外，它们在 power on strapping 阶段，还当做 phy 地址配置管脚用，所以有时这两个管脚外部会有强上拉或强下拉 (4.7k)。

LED 管脚输出极性（即高有效还是低有效）与其管脚上所接的上拉电阻还是下拉电阻有关。（有外部上下拉电阻，则以外部为准；无外部上下拉电阻，则依赖内部默认上下拉电阻）

有上拉电阻，则为低有效（即需要外接 LED 灯的阴极）；有下拉电阻，则为高有效（外接 LED 灯的阳极）。

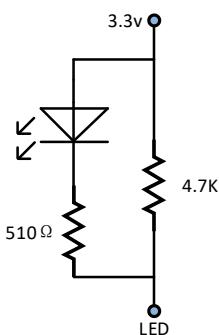


图4 a 上拉、sink模式

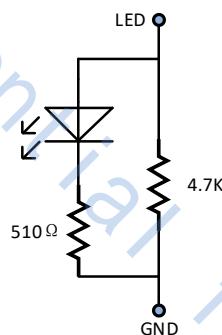


图4 b 下拉、source模式

如上图所示，

1. 在 power strapping 期间，LED0 与 LED1 被上拉，则工作模式下时，LED 低有效，sink 电流，默认的 LED 行为如下：

LED0: 10M link up, 则灯亮，输出为低；10M 数据传输，则灯闪；其余状态灯灭。

LED1: 100M link up, 则灯亮，输出为低；100M 数据传输，则灯闪；其余状态灯灭。

2. 在 power strapping 期间，LED0 与 LED1 下拉(芯片内部默认下拉)，则工作模式下时，LED 高有效，source 电流，默认的 LED 行为如下：

LED0: 10M link up, 则灯亮，输出为高；10M 数据传输，则灯闪；其余状态灯灭。

LED1: 100M link up, 则灯亮，输出为高；100M 数据传输，则灯闪；其余状态灯灭。

如要改变 LED 的行为，则需配置内部寄存器，详见 LED 灯配置章节。

4.7 SMI (MDIO) 接口

MDIO 接口包括两根信号线: MDC(pin22)和 MDIO(pin23), YT8512 的 MDC 所支持数据时钟频率达到 12.5MHz, MDIO 支持双向数据交换。需要说明的是, MDC 时钟与 TXC、RXC 时钟无关联性。

尽管内部有上拉, MDIO 在使用中, 外部需采用 4.7k 电阻上拉至 DVDD33。

4.8 MII, REMII, RMII 接口

YT8512 支持 MII/RMII 两种接口与 MAC 相连。YT8512 在上电初始期间, 检测 (Power Strapping) RX_DV(pin8) 管脚, 若 RX_DV 为低电平, 则设置为 MII; 否则为 RMII。另外由于 RX_DV 管脚内部为弱下拉, 如果外部电路不做设置, 则 YT8512 默认就是 MII 接口。但强烈建议实际应用中, 在外部加上/下拉电阻, 不依赖内部的弱上/下拉。

YT8512 也会在上电期间检测管脚 RXD3/CLK_CTL(pin12, 内部弱下拉), 根据其电平状态决定 TXC 的时钟方向。低电平时 TXC 为输出; 高电平时 TXC 为输入。

这两个 POS 的组合产生的模式如下:

RXDV	RXD3	模式	备注	应用场合
0	0	MII 模式	TXC,RXC 由 PHY 输出	与 MAC MII 接口相连
0	1	REMII 模式	TXC,RXC 由对端输出给 PHY	一般不与 MAC 相连, 而是与另一颗 PHY
1	0	RMII2	TXC 由 PHY 输出给 MAC	与 MAC RMII (TXC 输入) 接口相连
1	1	RMII1	TXC 由 MAC 输出给 PHY	与 MAC RMII (TXC 输出) 接口相连

4.8.1 MII 与 REMII 模式

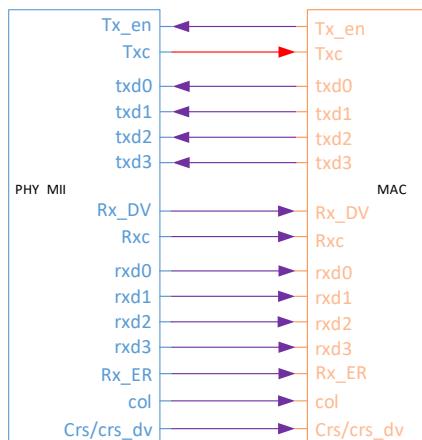


图 5a YT8512 工作在 MII 状态
(Rx_DV、RXD3 外接 4.7k 电阻到地)

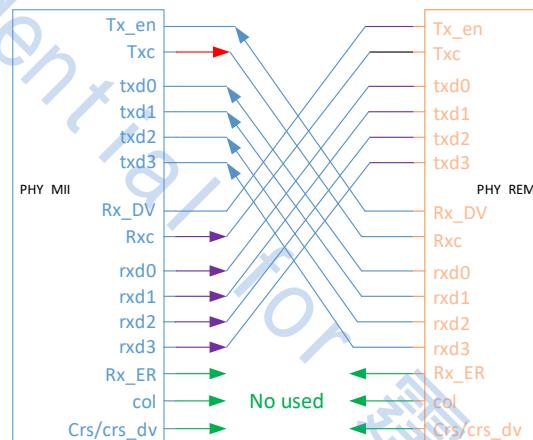


图 5b YT8512 工作在 REMII 状态
(Rx_DV 外接 4.7k 电阻到地、RXD3 接 4.7k 到 3.3v)

其中图 5a 为典型的 Phy 工作于 MII 模式下, 配合 MAC 的连接图。图 5b 一般用于两个 phy 背靠背相连的应用中。

4.8.2 RMII1 与 RMII2 模式

PHY 的 RMII 模式分两种 RMII1, RMII2。区别在于 TXC 时钟的方向。

RMII1 模式, TXC 是输入。

RMII2 模式, TXC 是输出。

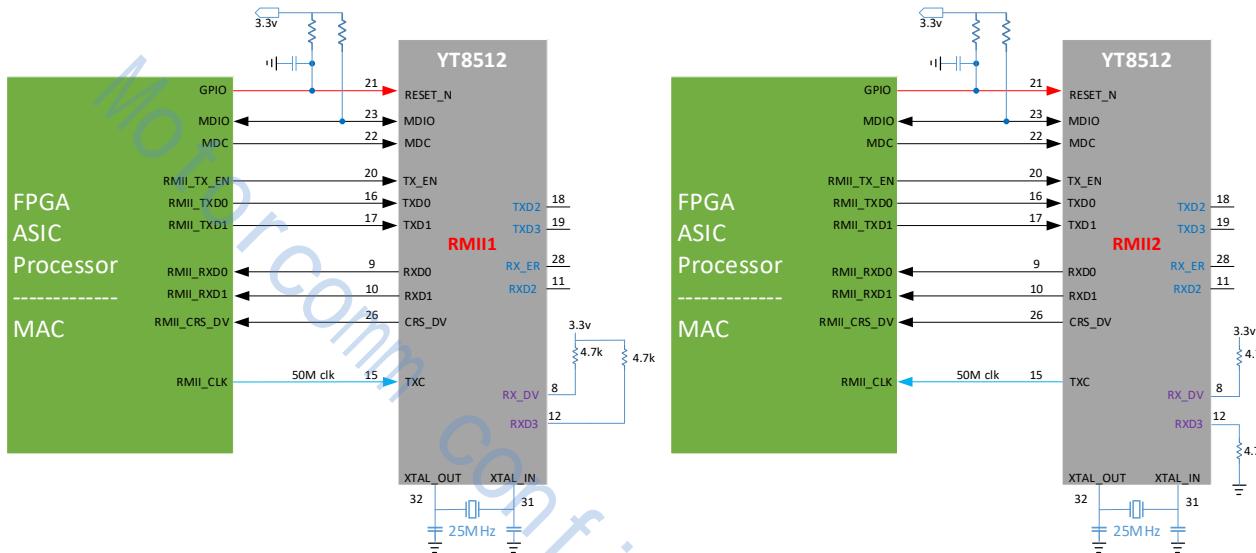
MAC 端的 RMII 模式要与 PHY 的模式相对应, 即 PHY RMII1 + MAC RMII2, 或 PHY RMII2 + MAC RMII1。

一般的，因为 RMII 模式用到 50MHz 的 RMII TXC，客户为省 PHY 的晶体，可能把 MAC 发出的 RMII TXC 直接连到 PHY 作参考时钟使用。因此根据不同的时钟源，RMII 模式的电路连接方式包括：RMII1/2 (25MHz 时钟) 和 RMII1 (外灌 50MHz 时钟)

4.8.2.1 RMII1/2 (25MHz 时钟)

25MHz 作参考时钟的情况下，只需 POS 设置到正确模式即可，不需要软件配置。

下图以 25MHz 晶体作参考时钟为例（外灌 25MHz 时钟信号与此类似，请参考前文的时钟章节），展示 PHY 与 MAC 的连接方式：



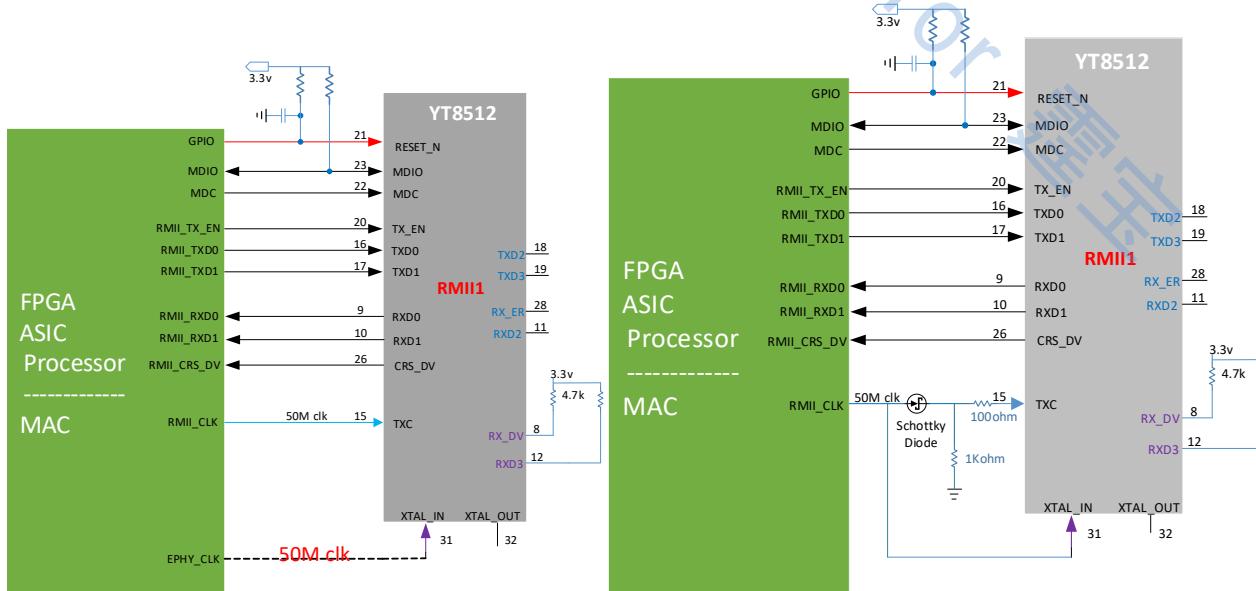
注意：

1. 上图中 PHY 的 CRS_DV 与 MAC 相连，而非 RXDV。

4.8.2.2 RMII1/2 (50MHz 时钟)

50MHz 作参考时钟的情况下，除 POS 设置到正确模式外，还需要 PHY 在上电初始化时作相应软件配置。

通常用 50MHz 作参考时钟时，是由 MAC 提供的 50MHz 时钟：可以是 MAC 的 TXC，或者 MAC 有单独的 50MHz 时钟信号输出。此时 PHY 用在 RMII1 模式下。下图以这两种情况为例（用 50MHz 晶体与此类似，请参考前文的时钟章节），展示 PHY 与 MAC 的连接方式：



硬件 POS 设置:

RXDV, RXD3 外接上拉

硬件 50MHz 时钟一分二时:

需要肖特基二极管和两个电阻作隔离和限流作用。绝大多数的肖特基二极管都可用, 比如 SD103AWS。

软件寄存器设置:

`Write_ext_reg0x50[6]: 1'b1 //选择 50MHz 参考时钟`

`Write_ext_reg0x4000[4]: 1'b1 //在 RMII 模式 default 为 CRS_DV, 如果需要配置为 RX_DV, 需要置"1"`

`Write_mii_reg0x0[15]: 1'b1 //软复位`

注意:

1. 上图中 PHY 的 CRS_DV 与 MAC 相连, 而非 RXDV。

2. 上述的软件寄存器设置, 只要是 50MHz, 无论是外灌/晶体, 无论 RMII1/RMII2 都需要设置。

3. 上电时的电压, 时钟, 复位时序需达到“4.2 上电时序”章节提的要求。因为时钟由 CPU 提供, 所以 PHY 的复位管脚也应该由 CPU 控制, 方便调节先后来满足上电时序要求。

4. 上述外灌 50MHz 应用场景下的电路与对标产品不一致, 在替换方案中需要特别注意。

5. 外灌时钟时, 推荐从 XTAL_IN 输入, 因为 XTAL_IN 为输入高阻态, 不会影响输入时钟幅度。

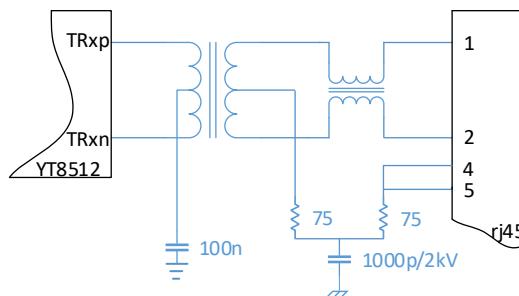
4.9 避免进入 SCAN 模式

SCAN 模式是 YT8512 的一种测试模式。用于芯片出厂测试使用, 而在正常使用时不能进入此模式, 否则会导致芯片无法正常工作, 甚至 MDIO 都无法访问。YT8512 通过上电复位时 CRS_DV 管脚的输入高电平来使芯片进入 SCAN 模式。此管脚在芯片内部有弱下拉, 并且为输出状态。理论上, 上电复位时为低电平且相连的对端管脚应为输入态, 若对端自带拉或者其在 PHY 上电复位时将此管脚强拉为高, 则 YT8512 会误进入 SCAN 模式, 导致功能异常。

如何避免进入 SCAN 模式:

1. 确认与 YT8512 CRS_DV 管脚相连的另一端为输入态且无上拉电阻, 则无需额外电路。
2. 若另一端有上拉电阻, 则需额外在 CRS_DV 加下拉 4.7Kohm 电阻, 以确保是下拉状态。
3. 实测上电硬复位信号释放到之后至少 10us 之内, CRS_DV 应一直保持为低电平。

4.10 MDI 接口



YT8512 内置端口自转功能, 能自动识别直连线或交叉网线。YT8512 内置 100Ω 端接电阻、电压型输出, 因此变压器芯片侧中心抽头只需 100n 电容接地。

注意:

1. 上图中只列出一对 MDI 线的接法, 另一对 MDI 线接法相同。
2. 由于百兆只用两对线, 对于 RJ45 上另外两对线, 可按上图 RJ45 pin 4, 5 的接法。

4.11 硬复位后延时进行 MDIO 操作要求

硬复位从低变高后，芯片需要在一段时间内完成内部的复位和 power-on-strapping 工作，因此一般要求在硬复位释放至少 100ms 后再进行 MDIO 操作。

4.12 外接参考时钟的抖动要求

通常可以接无源晶体在 XTL_IN 和 XTL_OUT 管脚，产生 DUT 的参考时钟。若要外接时钟信号，具体接法参见 datasheet。要求外灌 25MHz 的时钟抖动范围为：

Peak to peak jitter < 200ps;

25kHz~25MHz rms jitter < 3ps;

broadband rms jitter < 9ps

(其它的要求，比如幅度，占空比等参见 datasheet)

4.13 外接参考时钟源切换操作要求

在外接参考时钟的情况下，如果有多个时钟源，并且有时钟源切换的要求，那对切换过程的要求如下：

1. 因为切换过程中，时钟的行为及对芯片造成的影响无法预测，而硬复位比软复位更为彻底，从系统可靠性角度看，建议切换时钟源后用硬复位将 PHY 复位。

2. 硬复位的时序，请参考本应用说明的复位时序要求(即 RESET 信号应该在时钟稳定后保持为低至少 10ms，再拉高/释放)。

3. 硬复位释放后，会清掉之前软件所配的所有的寄存器设置。MDIO 需要等至少 100ms 之后再进行操作，软件可通过 MDIO 将所需的寄存器重新配置（一般与上电初始化的寄存器配置相同）。

4.14 数据传输通路延时

系统实现精准时钟同步功能时，需要知道数据从 MDI 接口到 MII/REMII/RMII 接口，或从 MII/REMII/RMII 到 MDI 接口，所花的时间。包括时间的绝对值，和时间的抖动范围(Variation)。在各模式下，其值如下：

mode	direction	speed	basic delay(ns)	variation(ns)
mii_mode	mii_tx->utp_tx	100M	232	0
	utp_rx ->mii_rx	100M	204	40
	mii_tx->utp_tx	10M	1170	0
	utp_rx ->mii_rx	10M	7060	400
rmii_mode	rmii_tx->utp_tx	100M	392	40
	utp_rx ->rmii_rx	100M	424	80
	rmii_tx->utp_tx	10M	2770	400
	utp_rx ->rmii_rx	10M	9460	800
remii_mode	remii_tx->utp_tx	100M	472	40
	utp_rx ->remii_rx	100M	424	80
	remii_tx->utp_tx	10M	3570	400
	utp_rx ->remii_rx	10M	9460	800

注：

上表中的数值有一定误差，具体数据以实测结果为准。

5. 寄存器访问方式

系统对 YT8512 的配置管理、状态获取均依赖 MDIO 接口通信读写 YT8512 内部寄存器实现。

5.1 寄存器的类型

YT PHY 内部有三种类型的寄存器：MII 寄存器（MII register，以下简写为 mii_reg），扩展寄存器（extended register，以下简写为 ext_reg）和 MMD 寄存器（简写为 mmd_reg）。三者各自编址，地址都从 0x00 开始。

mii_reg: 寄存器地址为 0~0x1f，遵从 802.3 clause 22 的标准定义方式进行访问。其 MDC/MDIO 协议如下：

Table 22-12—Management frame format

	Management frame fields							IDLE
	PRE	ST	OP	PHYAD	REGAD	TA	DATA	
READ	1...1	01	10	AAAAAA	RRRRR	Z0	DDDDDDDDDDDDDDDDDD	Z
WRITE	1...1	01	01	AAAAAA	RRRRR	10	DDDDDDDDDDDDDDDDDD	Z

ext_reg: 由于 mii_reg 只有 0~0x1f，即 32 个。不能满足产品对寄存器数目的要求，就通过访问两个 mii_reg 0x1e, 0x1f 来扩展了更多的寄存器。访问方式为：

将要访问的 ext_reg 地址，写入 mii_reg 0x1e.

读 mii_reg 0x1f 得到值，即为 ext_reg 内的值. 写值到 mii_reg 0x1f，即将 ext_reg 寄存器内容改为写入的值.

例如：读取 ext_reg0x1000 的值： write mii_reg0x1e 0x1000;

read mii_reg0x1f;

写 ext_reg0x1000 的值为 0x3456:

write mii_reg0x1e 0x1000;

write mii_reg0x1f 0x3456;

mmd_reg: 为了扩展更多的寄存器，以适应更高速率的以太网。MMD 寄存器被创造出来。遵从 802.3 clause 45 的标准定义方式进行访问。MMD 寄存器有两种访问方式：间接访问和直接访问。

直接访问：其协议与 MII 访问不同，它是通过两条指令实现 MMD 寄存器的读或写，即先发 address 指令，再发 Write 或 Read 指令实现，具体协议如下：

Table 45-153—Extensions to management frame format for indirect access

Frame	Management frame fields							IDLE
	PRE	ST	OP	PRTAD	DEVAD	TA	ADDRESS / DATA	
Address	1...1	00	00	PPPPP	EEEEEE	10	AAAAAAAAAAAAAAA	Z
Write	1...1	00	01	PPPPP	EEEEEE	10	DDDDDDDDDDDDDDDD	Z
Read	1...1	00	11	PPPPP	EEEEEE	Z0	DDDDDDDDDDDDDDDD	Z
Post-read-increment-address	1...1	00	10	PPPPP	EEEEEE	Z0	DDDDDDDDDDDDDDDD	Z

间接访问：即通过 MII 寄存器（mii_reg 0xd, mii_reg 0xe，其具体含义见下表）来访问 MMD 寄存器。

比如：

读 MMD 3 的 0x5 寄存器，其指令为：

write mii_reg 0xd 0x3; write mii_reg 0xe 0x5; write mii_reg 0xd 0x4003; read mii_reg 0xe

写 MMD 7 的 0x3c 寄存器值为 0x6，其指令为：

write mii_reg 0xd 0x7; write mii_reg 0xe 0x3c; write mii_reg 0xd 0x4007; write mii_reg 0xe 0x6

MII 0Dh: MMD access control register

Bit	Symbol	Access	Default	Description

15:14	Function	RW	2'b0	00 = Address 01 = Data, no post increment 10 = Data, post increment on reads and writes 11 = Data, post increment on writes only
13:5	Reserved	RO	9'b0	Always 0
4:0	DEVAD	RW	5'b0	MMD register device address. 00001 = MMD1 00011 = MMD3 00111 = MMD7

MII 0Eh: MMD access data register

Bit	Symbol	Access	Default	Description
15:0	Address data	RW	16'b0	If register 0xD bits [15:14] are 00, this register is used as MMD DEVAD address register. Otherwise, this register is used as MMD DEVAD data register as indicated by its address register.

5.2 通过 MII 寄存器访问其它类型寄存器

按上面所讲，无论 ext_reg 还是 mmd_reg 都可以通过 mii_reg 来访问。

假设函数 `read_mii_reg(phy_addr, reg_addr)` 与 `write_mii_reg(phy_addr, reg_addr, data)` 表示读写指定 phy 地址的指定 mii 寄存器地址。即：

MII 寄存器的读写：

函数名： `read_mii_reg(phy_addr, reg_addr)`

输入参数： `phy_addr, reg_addr`

返回值： 读取到的 mii reg_addr 寄存器的值

函数名： `write_mii_reg(phy_addr, reg_addr, data)`

输入参数： `phy_addr, reg_addr, data`

返回值： 无

那么：

Ext 寄存器的读写：

函数名： `read_ext_reg(phy_addr, reg_addr)`

输入参数： `phy_addr, reg_addr`

返回值： 读取到的 Ext 寄存器 reg_addr 内的值

```
def read_ext_reg(phy_addr, reg_addr):
```

```
    write_mii_reg(phy_addr, 0x1e, reg_addr)
```

```
    d = read_mii_reg(phy_addr, 0x1f)
```

```
    return d
```

函数名： `write_ext_reg(phy_addr, reg_addr, data)`

输入参数： `phy_addr, reg_addr, data`

返回值： 无

```
def write_ext_reg(phy_addr, reg_addr, data):
```

```
    write_mii_reg(phy_addr, 0x1e, reg_addr)
```

```
    write_mii_reg(phy_addr, 0x1f, data)
```

MMD 寄存器的读写：

函数名： `read_mmd_reg(phy_addr, device, reg_addr)`

输入参数： `phy_addr, device, reg_addr`

返回值：读取到的 MMD 寄存器的值

```
def read_mmd_reg(phy_addr, device, reg_addr):
    write_mii_reg(phy_addr, 0x0d, device)
    write_mii_reg(phy_addr, 0x0e, addr)
    write_mii_reg(phy_addr, 0x0d, 0x4000+device)
    d = read_mii_reg(phy_addr, 0x0e)
    return d
```

(说明，例如 MMD3 0x01, 其中 3 即为 MMD 寄存器的器件地址；0x01 为 MMD 的寄存器地址)

函数名： **write_mmd_reg(phy_addr, device, reg_addr, data)** 输入参数： phy_addr, device, reg_addr, data

返回值：无

```
def write_mmd_reg(phy_addr, device, reg_addr, data):
    write_mii_reg(phy_addr, 0x0d, device)
    write_mii_reg(phy_addr, 0x0e, reg_addr)
    write_mii_reg(phy_addr, 0x0d, 0x4000+device)
    write_mii_reg(phy_addr, 0x0e, data)
```

5.3 读写寄存器的某几位的表示方法

相对写寄存器的全部 bit 位，大部分功能只是写寄存器的某几个 bit，而其它位需要保持原值不变。实现的方法通常是先读出原值，将值的某几位修改后，再写回寄存器。或者，读出某个寄存器的值，只获取相关的某几位。方便起见，本文中我们用以下语法表示(以 mii_reg0x1, ext_reg0xa 为例)：

write_mii_reg0x1[15]: 1'b0:

将 mii 寄存器 0x1 的 bit15，共 1 比特，写 0。其余比特位保持不变。

read_mii_reg0x1[15]:

读 mii 寄存器 0x1 的值并获取 bit15，共 1 比特。

write_ext_reg0xa[15,14,13]: 3'b010:

将扩展寄存器 0xa 的 bit15, 14, 13，共 3 比特，分别写为 0, 1, 0。其余比特位保持不变。

read_ext_reg0xa[15,14,13]:

读扩展寄存器 0xa 的值并获取 bit 15, 14, 13。

5.4 phyaddr0 和 bdcst_addr 的使用说明

YT8512 有 phyaddr0 和 bdcst_addr 两种特色功能，用户可以方便、快捷地访问 YT8512 的寄存器。

当 phyaddr0 enable 时，YT8512 除了可以响应 phy address = PHYAD[1:0]的读/写命令，还可以响应 phy address = 0x0 的读/写命令。(Strapping PHYAD[1:0]的说明请参考 Datasheet。)

通过 write_ext_reg0x0[6]: 1'b1 来 enable phyaddr0，通过 write_ext_reg0x0[6]: 1'b0 disable phyaddr0。默认 phyaddr0 是 enable 的。

当 bdcst_addr enable 时，YT8512 可以响应 phy address = bdcst_addr 的写命令，达到一次配置 MDIO 总线上多颗 PHY 的目的。其中，bdcst_addr 可以配置成 0x0~0x1f 中的任意值。

通过 write_ext_reg0x0[5]: 1'b1 来 enable bdcst_addr，并 write_ext_reg0x0[4:0]来配置相应的 broadcast address。通过 write_ext_reg0x0[5]: 1'b0 来 disable bdcst_addr。默认 En_bdcst_addr 是 enable 的，并且 Bdcst_addr = 0x1f。

需要注意的是：

- 1、当 bdcst_addr enable 时，YT8512 不会响应 phy address = bdcst_addr 的读命令，只响应写命令；
- 2、当 phyaddr0 enable 时，YT8512 可以响应 phy address = 0x0 的读命令。但如果 MDIO 总线上挂了多颗 PHY 芯片，此时读回来的值是不可靠的。

EXT 0000h:				
Bit	Symbol	Access	default	Description
15	Reserved	RO	1'b0	Reserved
14	Res_calib_comp_dout	RO	1'b0	Resistor calibration's compare result.
13:08	Res_calib_config	RO	6'b0	Resistor calibration result
7	En_1588_addr_incr	RW	1'b0	Enable addr increase automatically for 1588 reg access to speed up.
				No use.
6	En_phyaddr0	RW	1'b1	Enable mdio phy address 0 access
5	En_bdcst_addr	RW	1'b1	Enable mdio broadcast address access
4:00	Bdcst_addr	RW	5'b11111	MDIO Broadcast address

6. 寄存器功能配置

YT8512 是一款单口百兆以太网 PHY。本文主要介绍它的一些用法，包括：硬件上某些特殊的应用如何实现，软件上特定的功能如何配置寄存器及硬件电路设计中常问到的问题。

6.1 初始化配置及不支持的功能

已知的 DUT 在某些应用下可能存在问题，需要通过初始化配置来避免。不做特别说明的话，这些配置在上电复位后进行一次即可，只有下电或硬复位后才需要重新配置。下面是需要在初始化阶段做的配置。

6.1.1 MII 模式下百兆半双工模式不通

在 MII 模式下，当连接状态为百兆半双工时，MII 接口的 TXC/RXC 会无输出，造成 MAC 与 PHY 之间无法通讯。为避免此问题，需要在初始化时（或硬复位后）做如下设置：

write_ext_reg0x200a[15]: 1'b0 (将扩展寄存器 0x200a 的 bit15 清零，其它位保持前值不变)。

注：

1. 在 RMII 模式下无此问题，不需要此初始化配置。
2. 此问题在 YT8512 errata notice 中也有详细介绍。

6.1.2 不插网线误连接到 10BT

YT8512 连接到 10BT，然后断开网线，YT8512 有概率仍显示连接到 10BT。为避免此问题，需要在初始化时（或硬复位后）做如下设置：

write_utp_ext_reg0x200a[10]: 1'b0. (将扩展寄存器 0x200a 的 bit10 清零，其它位保持前值不变)。

注：

1. 此问题在 YT8512 errata notice 中也有详细介绍。

6.2 UTP（电口）配置

6.2.1 10/100BT 的速度和双工

上电硬复位后，可以读取 phy mii_reg0x2,0x3 获取 PHY ID。默认 YT8512 不需做任何配置，它自协商能力是打开的，100/10BT 能力都是打开的。phy 会自主与对端建立连接，不需主 CPU（MAC）干涉。

如何改变 PHY 支持的速度和双工？包括：

1. 自协商模式下改变 PHY 支持的速度和双工能力：

write_mii_reg0x0[12]: 1'b1 #自协商打开，此为默认打开状态

write_mii_reg0x4[8:5] #相应 bit 置 1 来打开对应的本地自协商时的 100/10BT 的速度双工能力

write_mii_reg0x0[15]: 1'b1 #软复位，使上述配置生效

2. 强制模式下改变 PHY 支持的速度和双工能力：

write_mii_reg0x0[12]: 1'b0 #自协商能力关闭

write_mii_reg0x0[6,13] #相应 bit 置 1 来打开对应的本地强制的 100/10BT 的速度双工能力

write_mii_reg0x0[15]: 1'b1 #软复位，使上述配置生效

注：

1. 一般不建议配置成强制模式，因为这样会使得 PHY 或对方工作在半双工模式下，而半双工这种老旧的传输模式会出现包冲突（collision），严重影响速度或产生丢包。

2. 在自协商模式下，只是本地 UTP 速度和双工能力的打开或关闭，在与对方建立连接的过程中，双方通过线上传递自己支持的能力，最终按双方都支持的最高能力进行连接。

判断 PHY 当前的连接速度和双工？有两种方法：

通过标准的 PHY MII 寄存器来判断当前是否 link up (phy mii_reg0x1[2])，然后通过 phy mii_reg0x4, 0x5, 0x9 等标准寄存器和算法来得到当前 PHY 的连接速度和双工。

为了更方便地得到当前 PHY 的连接与否，速度与双工，YT PHY 将这些信息放在 phy mii_reg0x11(只读寄存器) 内，系统只需读取 0x11 寄存器，对应地配置 MAC，即可正常通信。

其中的几个 bit 含义如下（或参照芯片手册）：

Bit 15-Bit14 : speed mode, 11---系统保留; 01---100M; 00---10M

Bit 13 : Duplex, 1---Full duplex; 0---half duplex

Bit 10 : Link status, 1---link up; 0---link down

注：要先判断 bit 10 为 1，表示连接建立，再去判断 bit 15,14,13 得到速度和双工。

配置好 MAC 后，通过轮询上述的寄存器确认 PHY 已经连接，然后上层就可进行网络通信。此过程中有一点需注意，即轮询本地 PHY 已经连接后，可能对端的 PHY 连接还未建立好，上层需要稍微延迟一段时间（大概 0.5 秒以后）再发包，以确保此时对端连接建立，才不会导致丢包。

6.2.2 power down 设置

在不需要 UTP 连接的情况下，可以将 UTP 置为 power down 模式，以达到省电功能：

进入 utp power down: phy mii 0.11 置 1

退出 utp power down: phy mii 0.11 置 0, phy mii 0.15 置 1 进行 software reset 操作。

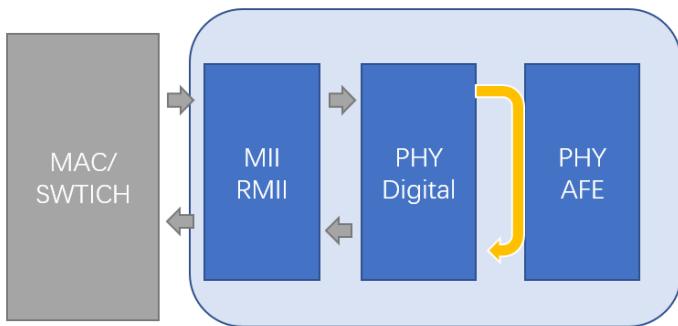
6.2.3 Loopback (回环) 模式

1、Internal loopback 情况下，utp 的 analog 电路被 bypass，直接将 mac 发送的数据从 utp phy 的数字电路回环回去，也可以称作 digital loopback，如下图所示。配置方法如下：

10M: utp phy mii register 0x0 = 0x4100

100M: utp phy mii register 0x0 = 0x6100

注意：一般的寄存器配置不会被软复位清掉，但 UTP mii register 0x0 里的 internal loopback 和 power down 2 个 bit 会被软复位清掉，因此在使能这两个功能后不要做软复位。

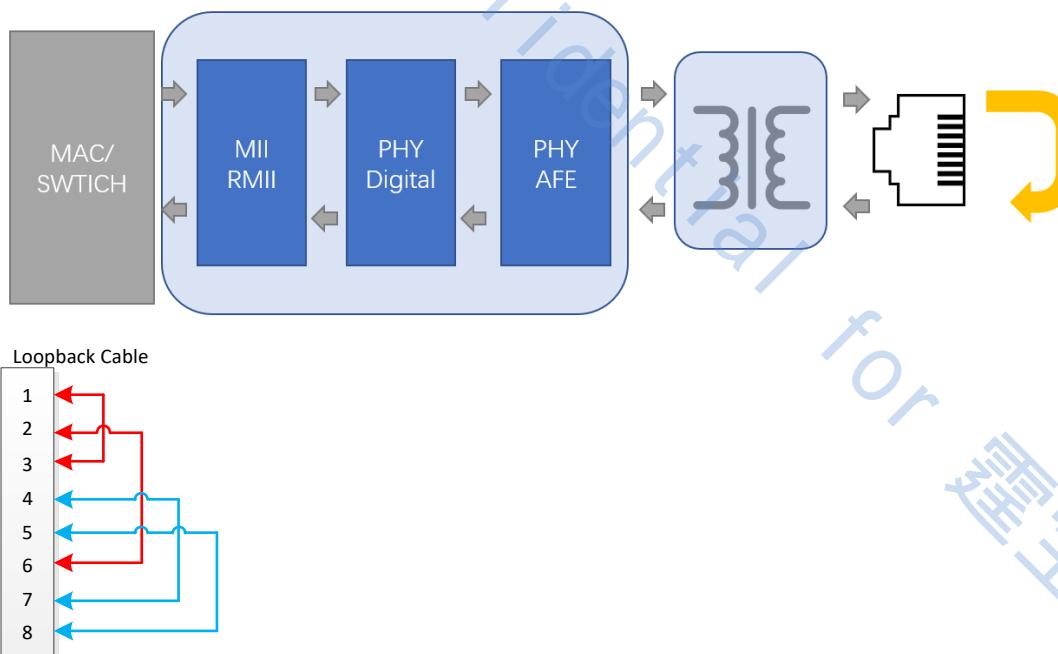


2、在 External Loopback 模式下，AFE 发送的信号由回环线直接回到 AFE 的接收电路。MAC 通过对比 MII/REMII/RMII 发送数据和接收数据，从而确保 MII/REMII/RMII 接口，PHY 数字和模拟功能的正确性。回环线将网线的 pair1&2 与 pair3&6 连接，pair4&5 与 pair7&8 连接，如下图所示。配置方法如下：

10M: 先 phy ext 0x2027 bit15 位清 0(关闭 sleep 模式); utp phy ext register 0x4000.12 = 1, mii register 0x0 = 0x8100;

100M: 先 phy ext 0x2027 bit15 位清 0(关闭 sleep 模式); utp phy ext register 0x4000.12 = 1, mii register 0x0 = 0xa100

注意：上述配置的最后一个 mii register 0x0，因为 bit15 置为 1，实现了软复位，使得前面的配置生效并发起 external loopback 机制，此时应保证 external loopback 线已经插上，否则会导致 external loopback 无法连接，这时需要再一次软复位才可以。



3、在 Remote Loopback 模式下，MII/REMII/RMII RX 接收的数据被直接注入 MII/REMII/RMII TX。远端的 MAC 通过对比发送数据和接收数据，从而确保自己和 link partner 的功能完好，如下图所示。配置方法如下：

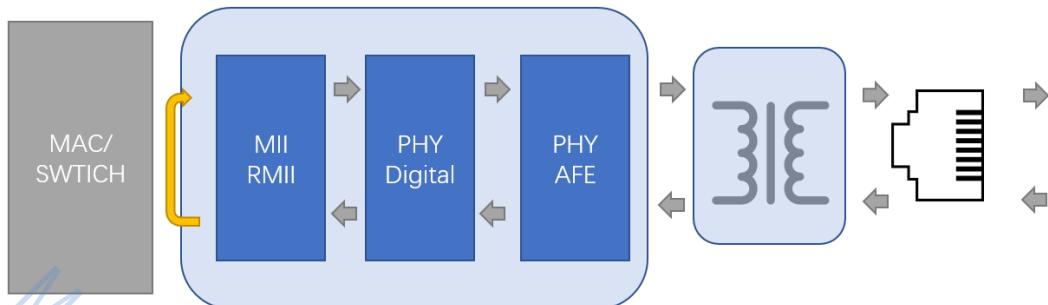
10M: utp ext register 0x4000.11 = 1

utp phy mii register 0x4 = 0xc41,

utp phy mii register 0x0 = 0x9140,

100M: utp extregister 0x4000.11 = 1

utp phy mii register 0x4 = 0xd01,
 utp phy mii register 0x0 = 0x9140,



6.2.4 Smart downgrade (自动降速) 配置

对于以太网 PHY，当网线插上时，默认情况下是从双方支持的最高速率开始尝试连接。有时由于外界干扰或连接线太长或质量太差，导致高速率连接不上。当反复尝试连接超过一定次数时，PHY 可以选择自动降速来完成连接。Smart speed，或者叫 auto-downgrade，就是指 PHY 的这个自动降低连接速度的功能。

比如双方都按 100BT 的能力发起自协商，但网线超过最大连接距离（比如到 200 米），双方经过 5 次尝试失败（每次 link up 维持时间小于 2 秒），仍无法连接到 100BT，此时具备自动降速功能的一方会发起 10BT 的自协商，尝试 10BT 的连接。

注：

- 如果对端口的快速插拔，导致短时间内的不断 link up/down，也会触发 smart downgrade 机制。
- 此功能只发生在连接建立时间内，若连接并维持住，PHY 不会根据丢包等自动降速。

以上是 YT8512 的默认行为。如果不需要此功能，则需要将此功能通过寄存器禁用。即，将寄存器 phy_mii_reg0x14 bit5 清 0，然后做 software reset (将 phy_mii_reg0x0 bit15 置 1)使其生效。描述如下：

Phy MII 14h: Speed Auto Downgrade Control Register				
Bit	Symbol	Access	Default	Description
15:12	Reserved	RO	0x0	Reserved
11	Reserved	RW	0x1	Reserved
10	Reserved	RW	0x0	Reserved
9	Reserved	RW	0x0	Reserved
8	Reserved	RW	0x0	Reserved
7:6	Reserved	RO	0x0	Reserved
5	En_speed_downgrade	RW	0x1	When this bit is set to 1, the PHY enables smart-speed function. Writing this bit requires a software reset to update. This bit will be set to 1'b0 in UTP_TO_FIBER_FORCE and UTP_TO_FIBER_AUTO mode; else set to 1'b1, only take effect after software reset
4:2	Reserved	RW	0x3	Reserved
1	Reserved	RW	0x0	Reserved
0	Reserved	RO	0x0	Reserved

注：若要禁用 smart speed，需要网线两端的 PHY 都禁用才可以。否则任一边自动降速都会导致连接降速。

6.2.5 Sleep (自动睡眠) 配置

UTP 处于不连接状态超过一定时间后（约 40 秒），PHY 会自动进入 sleep 状态。在 sleep 状态下，PHY 会自动关闭内部部分电路，以达到省电的功能。进入 sleep 后，PHY 会定期发送 Pulse 信号，并打开信号侦测的功能。一旦收到对方发过来的信号，超过信号侦测的门限，PHY 会立刻打开相关电路进入正常工作状态。

Sleep 功能默认是打开的，此功能不会影响 PHY 的正常连接。如果关闭 sleep 功能也不会有任何副作用。

关闭此功能的寄存器： write_ext_reg 0x2027[15]: 1'b0。

Phy EXT 2027h: Sleep Control1				
Bit	Symbol	Access	default	Description
15	En_sleep_sw	RW	0x1	1 = enable sleep mode: PHY will enter sleep mode and close AFE after unplug cable for a timer;
14	Pllon_in_slp	RO	0x0	1 = keep PLL on in sleep mode;
13	Slp_pulse_sw	RW	0x1	when PHY enter sleep,
12	En_upd_afe_sbs	RW	0x0	When AFE control is changed, no matter it's triggered by sleep control logic or normal work mode change,
11:6	Reserved	RO	0x0	Reserved
5	Sleeping	RO	0x0	1 = PHY is slept;
4	Gate_25m	RO	0x0	Not used.
3:0	Slp_state	RO	0x0	FSM state of internal sleep control logic.

6.2.6 包生成器和收发包统计

包生成器：packet generator；收发包统计：packet checker。

packet checker 应用说明：

1. 打开 packet checker：写 ext reg 0x40A0.[15:14]=2'b10
2. 关闭 packet checker：写 ext reg 0x40A0.[15:14]=2'b01
3. 查看 checker 统计结果：
4. Ext reg 0x40A3~0x40A8，代表从 MDI 收到的正确包的数量，寄存器为 read clear。
5. Ext reg 0x40A9~0x40AC，代表从 MDI 收到的包有 crc error，寄存器为 read clear。
6. Ext reg 0x40AD~0x40B2，代表从 MII/RMII 收到的正确包的数量，寄存器为 read clear。
7. Ext reg 0X40B3~0X40b6，代表从 MII/RMII 收到的包有 crc error，寄存器为 read clear。

packet generator 应用说明：

1. 设置 packet generator 的发送包的长度：写 ext reg 0x40A1[15:0]
2. 设置发送包的个数：写 ext reg 0x40A2[15:0]，值为 0 表示一直发包。
3. 设置 IPG 长度：写 ext reg 40A0[7:4]，不用时可以不设置。
4. 打开 packet generator：
5. 写 ext reg 0x40A0[15:13] = 3'b100，配置 packet checker 和 generator。
6. 写 ext reg 0x40A0[12] = 1'b1，打开 packet generator 开始发包。
7. Polling ext reg 0x40A0[12]，为 1'b0 时，表示发完了设置的包个数。包个数设置为 0 时，写 ext reg 0x40A0[12] : 1'b0，表示停止发包。
8. 关闭 packet generator：
9. 写 ext reg 0x40A0[14]: 1'b1，关闭 packet generator clock
10. 写 ext reg 0x40A0[15, 13, 12]: 3'b010，关闭 generator 和 checker

6.2.7 SNR 读取（当前连接质量指示）

YT8512 的 phy ext 扩展寄存器 0x205A 的 bit[14:0]值表征了一对 MDI 线的 SNR。

读取步骤如下：

步骤一，将 phy ext 扩展寄存器 0x2059 的 bit15 置 1

步骤二，分别读取 phy ext 扩展寄存器 0x205A 的 bit[14:0]

将其记为 MSE(i), i=0,1,2,3

将每个值读至少 100 次，并取平均值，得到每对 MDI 线上稳定 MSE

步骤四，将上述稳定 MSE 值，按照如下公式计算 SNR：

百兆： $10 * \log_{10}(32768/mse)$

要达到稳定连接，SNR 值应大于 19.13dB (百兆对应 MSE 约 400)；

6.2.8 Template (电口指标) 配置

YT8512 的 Template 测试指令如下：

初始化配置：

无

100BT:

```
Write_ext_reg0x2027: 0x2026
Write_mii_reg_0x10: 0x2
Write_mii_reg_0x0: 0xa100
```

10BTe:

```
Write_ext_reg0x2027: 0x2026
Write_mii_reg_0x10: 0x2
Write_mii_reg0x0: 0x8100
```

```
Write_ext_reg0x200a: 0xCA09      #packet with all ones, 10MHz sine wave, for harmonic test
Write_ext_reg0x200a: 0xCA0A      #pseudo random, for TP_idle/Jitter/Different voltage test
Write_ext_reg0x200a: 0xCA0B      #normal link pulse only
Write_ext_reg0x200a: 0xCA0C      #5MHz sine wave
Write_ext_reg0x200a: 0xCA0D      #Normal mode
```

上述设置能通过 template 测试。考虑到不同的板级设计，PHY 内部也有丰富的配置方便用户调整。其中常用的几项调整为：

A, 调整+/-Vout Differential Output Voltage:

100BT, 可调 ext_reg0x2057, 其中 bit[14:12]为粗调, [11:8]为细调

10BTe, 可调 ext_reg0x2057, 其中 bit[6:4]为粗调, [3:0]为细调

以上值增加则输出幅度增大，粗调每增加 1 则幅度增大约 25%，细调每增加 1 则幅度增大约 1/16。

B, 调整 rise/fall time:

```
ext_reg 0x1[9] set to 1
ext_reg 0x14 set to 0x1fe
ext_reg 0x15 set to 0x1fe
```

6.3 MII/RMII 配置

6.3.1 驱动能力的设置

UTP 的 MII/RMII 驱动能力，反映在上升/下降时间。驱动能力大，意味着能带更大的负载，但也可能引入 EMI 的问题。UTP 默认的驱动能力是权衡两者之后的设置，并非最强档位。若需要加强或减弱驱动能力，可以通过配置扩展寄存器来配置：

Write_ext_reg0x4001[5:4]: 默认为 2'b10。2'b11 为最强档，2'b00 为最弱档。

6.4 LED 灯配置

YT8512 支持两个 LED 灯：LED0, LED1。它们的行为（闪烁，亮，灭）由其各自对应的一个寄存器的配置来决定。

LED0 对应的寄存器为 ext Reg0x40C0; (上电默认值为 0x0311)

LED1 对应的寄存器为 ext Reg0x40C3; (上电默认值为 0x0320)

向这两个寄存器赋不同值，则 LED 做相应的状态显示，典型几种配置如下

ext Reg0x40C0/0X40C3 配置值	动作行为
0x0311	link在10M 时， LED 亮； link在10M 且收发包， LED 闪
0x0320	link在100M 时， LED 亮； link在100M 且收发包， LED 闪
0x30	link或收发包时， LED 亮； link down时灭
0x1300	收发包时闪烁； 不收发包即灭

6.5 Interrupt (中断) 配置

YT8512 的 11 管脚(INT_N/RXD2)为中断和 RXD2 的复用管脚。当芯片工作在 RMII 模式时，11 管脚作为 INT_N 使用；当工作在 mii 和 remii 模式时，作为 RXD2 使用。

作为中断使用时，有几点说明：

1. 中断管脚，它为低有效。
2. 配置 UTP mii reg0x12 寄存器来选择普通中断中何种事件可以触发中断信号。
3. 当相应中断事件产生后，中断管脚信号变低，CPU 通过读取中断寄存器 (mii reg0x13) 查看当前触发中断的具体事件。此寄存器读取后会自动清零，且中断信号自动变为高。。

相关的寄存器有：

PHY MII 12h: Interrupt Mask Register				
Bit	Symbol	Access	Default	Description
15	Auto-Negotiation Error INT mask	RW	0x0	1 = Interrupt enable
14	Speed Changed INT mask	RW	0x0	1 = Interrupt enable
13	Duplex changed INT mask	RW	0x0	1 = Interrupt enable
12	Page Received INT mask	RW	0x0	1 = Interrupt enable
11	Link Failed INT mask	RW	0x0	1 = Interrupt enable
10	Link Succeed INT mask	RW	0x0	1 = Interrupt enable
9:7	reserved	RW	0x0	No used.
6	WOL INT mask	RW	0x0	1 = Interrupt enable
5	Wirespeed downgraded INT mask	RW	0x0	1 = Interrupt enable
4:2	Reserved	RW	0x0	No used.
1	Polarity changed INT mask	RW	0x0	1 = Interrupt enable
0	Jabber Happened INT mask	RW	0x0	1 = Interrupt enable

PHY MII 13h: Interrupt Mask Register				
Bit	Symbol	Access	Default	Description
15	Link Down INT mask	RW	0x0	1 = Interrupt enable
14	Link Up INT mask	RW	0x0	1 = Interrupt enable
13	Link Speed Change INT mask	RW	0x0	1 = Interrupt enable
12	Link Duplex Change INT mask	RW	0x0	1 = Interrupt enable
11	Link Polarity Change INT mask	RW	0x0	1 = Interrupt enable
10	Link Page Received INT mask	RW	0x0	1 = Interrupt enable
9:7	reserved	RW	0x0	No used.
6	Link Failed INT mask	RW	0x0	1 = Interrupt enable
5	Link Succeed INT mask	RW	0x0	1 = Interrupt enable
4:2	Reserved	RW	0x0	No used.
1	WOL INT mask	RW	0x0	1 = Interrupt enable
0	Wirespeed Downgrade INT mask	RW	0x0	1 = Interrupt enable

Bit	Symbol	Access	Default	Description
15	Auto-Negotiation Error INT	RO	0x0	Error can take place when any of the following
14	Speed Changed INT	RO	0x0	1 = Speed changed
13	Duplex changed INT	RO	0x0	1 = duplex changed
12	Page Received INT	RO	0x0	1 = Page received
11	Link Failed INT	RO	0x0	1 = UTP link down takes place
10	Link Succeed INT	RO	0x0	1 = UTP link up takes place
9:7	Reserved	RO	0x0	No used
6	WOL INT	RO	0x0	1 = UTP received WOL magic frame.
5	Wirespeed downgraded INT	RO	0x0	1 = speed downgraded.
4:2	Reserved	RO	0x0	No used.
1	Polarity changed INT	RO	0x0	1 = UTP revered MDI polarity
0	Jabber Happened INT	RO	0x0	1 = 10BaseT TX jabber happened

以选择 Link up 作为产生中断的事件为例：

`Write_mii_reg0x12: 0x400 #bit10 置 1`

在完成以上设置，需要验证此设置是否能正确地触发中断，方法如下：

1. 对 YT8512 进行软复位，使已有的连接断开：

`Write_mii_reg0x0: 0x9140 #bit15 置 1`

2. 等待几秒钟，或 Polling UTP 状态寄存器直到 link up。

3. 测量 INT_N 管脚信号变低

4. 读中断状态寄存器，获取已触发的中断事件

`Read_mii_reg0x13 #得到的值的 bit10 就为 1`

`Read_mii_reg0x13 #得到的值应为全 0`

5. 测量 INT_N 管脚信号变高

6.6 其它维测寄存器

6.6.1 用户自配置寄存器

DUT 寄存器一般只在下电或硬复位后才会清掉，如果 DUT 工作中遇到异常，客户想确认发生异常时是否发生了掉电或硬复位，就需要一个指定的寄存器，在初始化写入值，在发生异常时读取此值。此寄存器为 ext_reg0x40B9，其默认值为 0x0。

6.6.2 PLL 锁定指示

DUT 内部的 PLL 在时钟信号出现很大频偏时，会发生失锁，从而导致内部工作异常。查看其指示的步骤如下：

1. ext_reg0x61[6]置 1， 默认是 enable 的，可以不配置；
2. 读取 ext_reg0xa0[1:0]，若这 2 位不同时为 0，则 PLL unlock。

备注：

对输入的 clk，一般偏差到 10MHz 以上才会出现 PLL unlock。

EXT 61h:				
Bit	Symbol	Access	default	Description
6	en_vco_check	RW	0x0	enable or disable VCO_fast or VCO_slow check circuit

EXT A0h:

Bit	Symbol	Access	default	Description
1	Pll_vco_h	RO	0x0	PLL fast indicator
0	Pll_vco_l	RO	0x0	PLL slow indicator

6.6.3 Clock drift

DUT 和 LP 之间频偏的计算方法:

- 1, 把 ext reg0x2058[14]置 ‘1’;
- 2, 把 ext reg0x2059[15]置 ‘1’;
- 3, 确认 DUT 和 Link parter 处于 link up status, 且 link 在 100BT;
- 4, 读取 ext reg0x2068 的值, 把读取值的补码*0.95 为 DUP 和 Link parter 之间的频差, 单位为 ppm

EXT 2068h: PHY Debug, Integrator

Bit	Symbol	Access	default	Description
15:00	Integrator	RO	16'h0	Channel 0's integrator. Frquence different between lp and dut. PPM = 0.95 * integrator; integrator is 2's complement value
				Valid when EXT 2058h bit14 en_snap_shot is set, or EXT 2059h bit15 prob_auto is set. Corresponding to these two setting, it is the:
				Integrator at the time snap shot condition is met, or
				Integrator at the time this EXT register is being read.