

Motorcomm

# YT8522 应用说明

单口 **10/100** 以太网物理层收发器

**Issue**    **V1.8**

**Date**    **2025-06-16**

## 版权声明

本文档版权归裕太微电子股份有限公司（以下简称“裕太微”）所有，并保留一切权利。未经裕太微书面许可，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。否则裕太微将保留追究其法律责任的权利。

## 免责声明

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，裕太微不承担任何责任。

更多信息，参见 <https://www.motor-comm.com>

# 前言

## 概述

本文档主要描述 YT8522 芯片硬件电路的注意事项和软件相关配置，内容涵盖寄存器类型及访问方法、各项功能配置、硬件电路相关要求等。

## 读者对象

本文档主要适用于 YT8522 相关研发人员。

## 缩略语

缩略语	英文全称	中文解释
PHY	Physical Layer	物理层
MAC	Media Access Control	数据链路层，本文代指与 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，本文代指 YT8522 芯片
LP	Link Partner	陪测对象，本文代指与 DUT 通过网线，光纤或 SMA 等线相连接的对端芯片
UTP	Unshielded Twisted Pair	非屏蔽双绞线，本文代指以太网电口（或称 RJ45），包括 100/10Mbps 两种速率
MDIO	Management Data Input Output	管理数据输入输出
SMI	Serial Management Interface	串行管理接口，本文指 MDC/MDIO
mii_reg	Standard Register	IEEE802.3 Clause22 规定的标准寄存器
ext_reg	Extended Register	扩展寄存器
mmd_reg	MDIO Manageable Device Register	IEEE802.3 Clause45 规定的 MDIO 接口进行管理的寄存器

## 变更信息

Revision	Release Date	Description
V1.8	2025-06-16	<ul style="list-style-type: none"><li>更新 Pin27 COL/BP_I2C 上下拉要求说明，参见 <a href="#">2.5 POS</a>。</li><li>更新使用外部时钟的相关说明，参见 <a href="#">2.2 上电时序</a>。</li><li>修正判断当前选择时钟源的寄存器，参见 <a href="#">2.3 时钟</a>。</li></ul>
V1.7	2024-12-04	<ul style="list-style-type: none"><li>初始化配置增加内部采样说明，参见 <a href="#">4.1 初始化配置</a>。</li><li>调整文档结构，修改文档描述及样式。</li></ul>
V1.6	2023-08-29	更新上电时序相关内容，参见 <a href="#">2.2 上电时序</a> 。
V1.5	2023-08-21	<ul style="list-style-type: none"><li>更新上电时序相关内容。</li><li>新增 MII Tx_delay 相关描述。</li><li>更新系统框图。</li></ul>
V1.4	2023-06-29	初始化配置中新增 RMII1/RMII2 模式初始化描述。
V1.3	2023-05-10	<ul style="list-style-type: none"><li>新增初始化配置。</li><li>修改 CSD 命名。</li></ul>
V1.2	2023-03-22	<ul style="list-style-type: none"><li>上电时序中加入 VDDIO。</li><li>更新时钟选择配置。</li><li>更新 LED 输出极性以及 LED 行为。</li><li>更新 MDIO 驱动能力配置。</li><li>新增 MII/RMII 电压配置。</li><li>更新数据传输通路延时。</li><li>新增初始化配置。</li><li>更新 CSD 配置。</li><li>更新 Template 配置。</li><li>新增 Unidirection 配置。</li><li>修改 LED 配置。</li><li>修改驱动能力配置。</li></ul>
V1.1	2023-03-17	<ul style="list-style-type: none"><li>更新 LED 配置相关描述。</li><li>增加 IO 口电源配置方式。</li></ul>
V1.0	2023-03-02	初稿。

# 目录

前言 .....	1
变更信息 .....	2
1 简介 .....	7
2 硬件电路 .....	8
2.1 电源 .....	8
2.2 上电时序 .....	8
2.3 时钟 .....	9
2.4 RBIAS .....	10
2.5 POS .....	10
2.6 LED .....	11
2.7 SMI (MDIO) 接口 .....	12
2.8 MII/REMII/RMII 接口 .....	13
2.8.1 MII 与 REMII 模式 .....	13
2.8.2 RMII1 与 RMII2 模式 .....	14
2.8.3 MII/RMII 电压配置 .....	17
2.9 MDI 接口 .....	17
2.10 Fiber 接口 .....	17
2.11 硬复位后延时进行 MDIO 操作 .....	18
2.12 外接参考时钟的抖动 .....	18
2.13 外接参考时钟源切换操作 .....	18
2.14 数据传输通路延时 .....	18
3 寄存器类型及访问方法 .....	20
3.1 寄存器类型 .....	20
3.1.1 MII 寄存器 .....	20
3.1.2 扩展寄存器 .....	20
3.1.3 MMD 寄存器 .....	20
3.2 寄存器软件访问方式 .....	22
3.2.1 MII 寄存器的读写 .....	22
3.2.2 扩展寄存器的读写 .....	23
3.2.3 MMD 寄存器的读写 .....	24
3.3 读写寄存器的表示方法 .....	25
3.4 PHYADDR0 和 BDCST_ADDR .....	26

4	寄存器功能配置.....	27
4.1	初始化配置.....	27
4.1.1	MII/RMII.....	28
4.1.2	100Base-TX/100FX.....	28
4.1.3	IO 电平 .....	28
4.2	UTP 配置 .....	29
4.2.1	10/100BT 的速度和双工.....	29
4.2.2	Power Down .....	30
4.2.3	Loopback 模式 .....	30
4.2.4	自动降速 .....	32
4.2.5	自动睡眠 .....	32
4.2.6	包生成器和收发包统计.....	33
4.2.7	SNR 读取.....	34
4.2.8	线缆状态检测.....	34
4.2.9	Template.....	35
4.3	100FX 配置 .....	36
4.3.1	Fiber .....	36
4.3.2	Unidirection .....	36
4.3.3	100FX 眼图测试 .....	36
4.4	MII/RMII 配置.....	38
4.4.1	驱动能力 .....	38
4.4.2	MII 模式下 Tx_delay .....	38
4.5	LED 灯配置.....	38
4.6	中断配置.....	39
4.7	其它维测寄存器 .....	40
4.7.1	用户自配置寄存器 .....	40
4.7.2	PLL 锁定指示 .....	40

## 表目录

Table 1	Power Sequence Timing Parameters.....	8
Table 2	PHY MII Register 0x1A.....	10
Table 3	MII/REMII/RMII 模式配置.....	13
Table 4	数据传输通路延时.....	19
Table 5	MMD Access Control Register (PHY mii_reg 0x0D).....	21
Table 6	MMD Access Data Register (PHY mii_reg 0x0E) .....	22
Table 7	EXT 0000 Register .....	26
Table 8	Ext_Reg 0x4210 .....	27
Table 9	Speed Auto Downgrade Control Register (Phy MII 14h).....	32
Table 10	Sleep Control1 (Phy EXT 2027h) .....	33
Table 11	LED 典型配置 .....	38
Table 12	Interrupt Mask Register (PHY MII 12h) .....	39
Table 13	Interrupt Mask Register (PHY MII 13h) .....	39
Table 14	EXT 61h Register .....	41
Table 15	EXT A0h Register .....	41

## 图目录

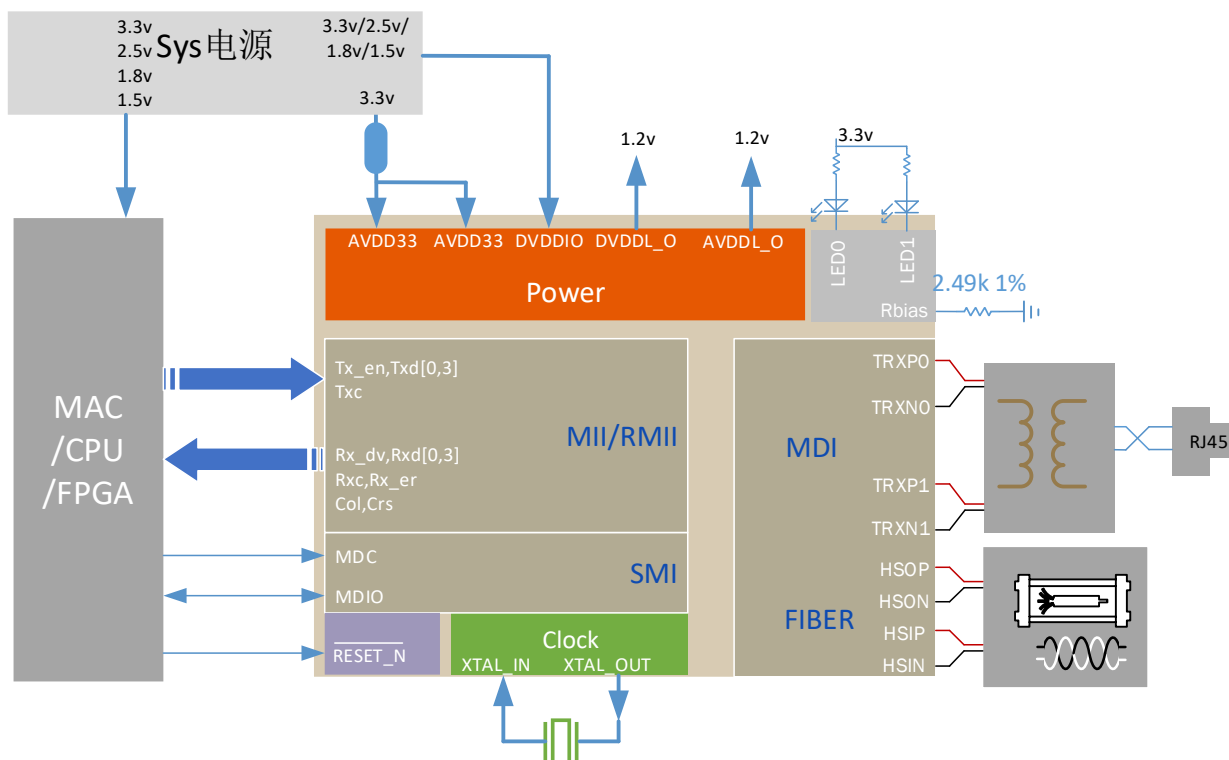
Figure 1	系统框图 .....	7
Figure 2	上电时序 .....	8
Figure 3	XTAL_IN 接 25M 时钟源 .....	9
Figure 4	XTAL_OUT 接 25M 时钟源 .....	9
Figure 5	采用 25M 晶振 .....	9
Figure 6	上拉-sink 模式 .....	11
Figure 7	下拉-source 模式 .....	12
Figure 8	YT8522 工作在 MII 状态 .....	14
Figure 9	YT8522 工作在 REMII 状态 .....	14
Figure 10	RMII1/RMII2 模式 ( 25MHz ) .....	15
Figure 11	RMII1 模式 ( 50M TXC ) .....	16
Figure 12	RMII1 模式 ( 外部 50M 时钟 ) .....	16
Figure 13	MDI 接口 .....	17
Figure 14	Fiber 接口 .....	18
Figure 15	Internal Loopback 框图 .....	30
Figure 16	External Loopback 框图 .....	30
Figure 17	Remote Loopback 框图 .....	31
Figure 18	Unidirection .....	36
Figure 19	测试拓扑 .....	37



# 1 简介

YT8522 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，内部包括 xMII、SMI、MDI、FIBER 四个模块。

**Figure 1** 系统框图



## 2 硬件电路

### 2.1 电源

DVDDIO (pin14) 与 AVDD33 (pin7) 外接 3.3V@80mA 电压，分别向 YT8522 的数字部分与模拟部分供电，两者之间建议接一个磁珠，且靠近管脚各自接 1 $\mu$ F 并 0.1 $\mu$ F 的电容。

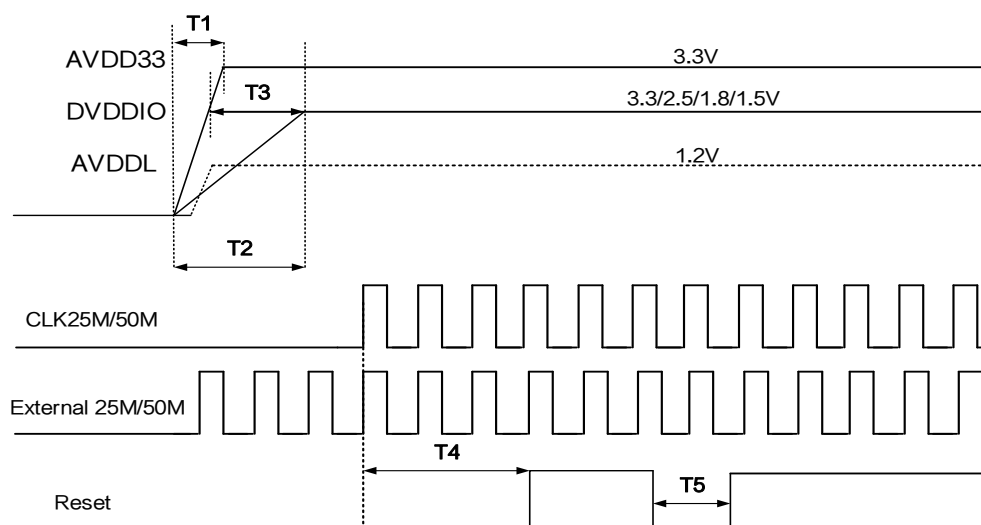
DVDDL\_OUT (pin29) 与 AVDDL\_O (pin2) 是 YT8522 内部产生的分别向数字与模拟部分供 1.2V 电压的输出管脚。此两管脚在使用中，建议各自靠近管脚外接 1 $\mu$ F 并 0.1 $\mu$ F 电容。

DVDDL\_OUT 与 AVDDL\_O 仅供 YT8522 内部电路供电，不可对外给其他器件供电。

### 2.2 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 低电平维持 10ms 即可实现可靠地复位。执行硬复位后 YT8522 内寄存器值均恢复为默认值。

**Figure 2 上电时序**



**Table 1 Power Sequence Timing Parameters**

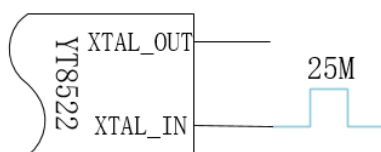
Symbol	Description	Min	Typ	Max	Unit
T1	AVDD33 power rising time.	0.1	-	-	ms
T2	DVDDIO power rising time.	0.1	-	-	ms
T3	DVDDIO delay time after AVDD33 over 2.0V.	-	-	5	ms
T4	PHY Reset delay time after CLK Stable.	10	-	-	ms
T5	PHY Reset active time.	10	-	-	ms

- 为了稳定的 POS，建议 DVDDIO 和 AVDD33 同时上电，最迟在 AVDD33（以到达 2V 的时间为参考）上电后的 5ms 内，DVDDIO 需要完成上电。
- 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。或者 CLK\_25MHz 在 3.3V 上电未稳定时即有输出也是正常现象。
- 若使用外部时钟，外部时钟需要在 AVDDH/AVDDL 到达 3.3V/1.2V 前稳定。如果外部时钟不稳定，芯片可能进入无时钟状态而无法读写大部分寄存器（除了 mii\_reg\_0x1A）。在此情形下，可以等到时钟稳定后，通过 mii\_reg\_0x1A 选择时钟源。
- 电源的爬坡时间至少需 >100μs，防止芯片内部产生过压。

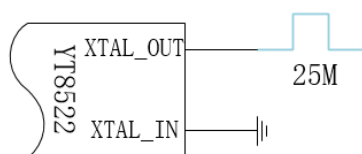
## 2.3 时钟

YT8522 可采用 25M 晶振，或外接 3.3V 的 25MHz 时钟信号。

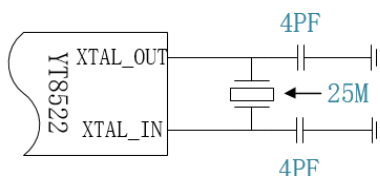
**Figure 3 XTAL\_IN 接 25M 时钟源**



**Figure 4 XTAL\_OUT 接 25M 时钟源**



**Figure 5 采用 25M 晶振**



YT8522 亦可接 50MHz 时钟信号（将上图中的时钟源或晶振替换为 50MHz）。

通过读取 mii\_reg\_0x1A bit0 的值来判断当前选择的是 50M 时钟还是 25M 时钟。

- 置 1，使用 50M 时钟
- 清 0，使用 25M 时钟

**Table 2 PHY MII Register 0x1A**

Bit	Symbol	Access	Default	Description
15:2	Reserved	NA	0x0	Reserved
1	txc_xtal_sel	RW	0x0	1: Use TXC as ext reference clock 0: Use XTAL as ext reference clock
0	xtal_freq_sel	RW	0x0	1: XTAL = 50MHz 0: XTAL = 25MHz

**Note:** 使用晶振提供时钟，支持 25M/50M 模式；使用外部 TXC 提供时钟，仅支持 50M 模式。

## 2.4 RBIAS

RBIAS (pin1) 管脚需外接 2.49kΩ (1%精度) 的电阻到地，YT8522 内部的部分电参数与此电阻相关。

## 2.5 POS

LED0、LED1、RX\_DV、RXD1、RXD3、COL 为复用管脚，除了正常状态下的功能外，在上电初始阶段，YT8522 检测此 6 位管脚的状态，以此对芯片做一些基本配置，这个过程称为 POS( Power on strapping)。

这 6 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，建议外部接上拉或下拉 4.7kΩ 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 YT8522 的 PHY 地址。

管脚名称_PHY 地址位		PHY 地址
Led1_add1	Led0_add0	
00		0
01		1
10		2
11		3

**Note:** 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。

- 0: LED0 作为 LED 输出管脚。
- 1: LED0 作为 WOL 中断输出 (open-drain 模式，只能 sink 电流)。

- RX\_DV (pin8)

- 0: MII 模式。
- 1: RMII 模式。

- RXD3 (pin12)
  - 0: TXC 为输出。
  - 1: TXC 为输入。
- COL (pin27)
  - 0: Enable I2C, 上电或者复位之后, MDIO/MDC 会发出 I2C 波形读取 E2PROM。
  - 1: Bypass I2C, 上电或者复位之后, 无 I2C 波形。

**Note:** 当不使用 I2C 功能时, 必须将此 pin 上拉。否则可能会影响主控或者相邻 PHY SMI 接口。

## 2.6 LED

YT8522 有两个 LED 管脚输出, 分别是 LED0 (pin24) 和 LED1 (pin25), 两管脚内部都有弱下拉电阻。除了作 LED 输出管脚外, 在 POS 阶段, 还作为 PHY 地址配置管脚, 所以这两个管脚外部会有强上拉或强下拉 (4.7k)。

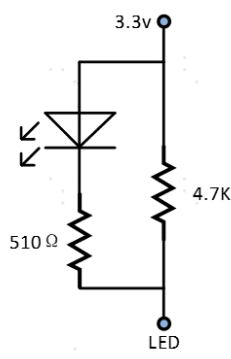
LED 管脚输出极性 (即高有效还是低有效) 与其管脚上接的上拉电阻还是下拉电阻有关。

- 有外部上下拉电阻, 则以外部为准。
  - 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)。
- 无外部上下拉电阻, 则依赖内部默认上下拉电阻

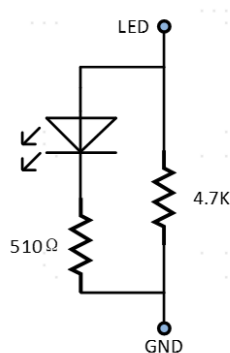
LED 的输出极性可以通过调节 ext\_reg0x4000 来实现, ext\_reg0x4000 bit14、bit13 分别代表 LED0 和 LED1 的极性。

- 0 = active high
- 1 = active low。

**Figure 6** 上拉—sink 模式



**Figure 7** 下拉—source 模式



- 在 power strapping 期间，LED0 与 LED1 上拉，则工作模式下，LED 低有效，sink 电流，默认的 LED 行为如下：
  - LED0
    - 10M 或者 100M 模式 link up 时，LED0 亮，输出为低。
    - 10M 或者 100M 模式 traffic 时，LED0 亮，输出为低。
    - 其余状态灯灭。
  - LED1
    - 10M 或者 100M 模式 link up 时，LED1 亮，输出为低。
    - 10M 或者 100M 模式 traffic 时，LED1 闪。
    - 其余状态灯灭。
- 在 power strapping 期间，LED0 与 LED1 下拉（芯片内部默认下拉），则工作模式下，LED 高有效，source 电流，默认的 LED 行为如下：
  - LED0
    - 10M 或者 100M 模式 link up 时，LED0 亮，输出为高。
    - 10M 或者 100M 模式 traffic 时，LED0 亮，输出为高。
    - 其余状态灯灭。
  - LED1
    - 10M 或者 100M 模式 link up 时，LED1 亮，输出为高。
    - 10M 或者 100M 模式 traffic 时，LED1 闪。
    - 其余状态灯灭。

如要改变 LED 的行为，则需配置内部寄存器，详情参见 [4.5 LED 灯配置](#)。

**Note:** SINK/SOURCE 电流模式下，设计时需要注意 LED 以及串阻的选型，以确保 LED 在关闭或者打开时都可以正常工作，避免出现 LED 漏光或者无法点亮。

## 2.7 SMI ( MDIO ) 接口

MDIO 接口包括 MDC ( pin22 ) 和 MDIO ( pin23 ) 两根信号线，YT8522 的 MDC 所支持数据时钟频率达到 12.5MHz，MDIO 支持双向数据交换。

**Note:** MDC 时钟与 TXC、RXC 时钟无关联性。尽管内部有上拉，MDIO 在使用中，外部仍需采用 4.7kΩ 电阻上拉至 DVDD33。

配置 ext\_reg0x18[8:4]，可以调节 MDIO 的 Delay。

- ext\_reg0x18[8:4]=5'b0 时，Delay 最小。
- ext\_reg0x18[8:4]=5'b11111 时，Delay 最大。

配置 ext\_reg0x4002[1:0]，可以调节 MDIO 的驱动能力。

- ext\_reg0x4002[1:0]=2'b00 时，驱动能力最小。
- ext\_reg0x4002[1:0]=2'b11 时，驱动能力最大。

## 2.8 MII/REMII/RMII 接口

YT8522 支持 MII/RMII 两种接口与 MAC 相连。YT8522 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。由于 RX\_DV 管脚内部为弱下拉，如果外部电路不做设置，则 YT8522 默认是 MII 接口。在实际应用中，建议在外部加上/下拉电阻，不依赖内部的弱上/下拉。

YT8522 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。

**Table 3 MII/REMII/RMII 模式配置**

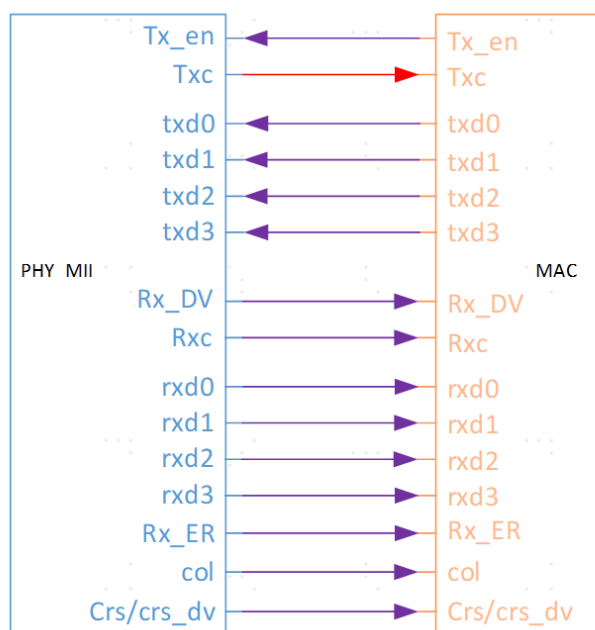
RXDV	RXD3	模式	备注	应用场景
0	0	MI I 模式	TXC 和 RXC 由 PHY 输出	与 MAC MII 接口相连
0	1	REMII 模式	TXC 和 RXC 由对端输出给 PHY	一般不与 MAC 相连，而是与另一颗 PHY
1	0	RMII2	TXC 由 PHY 输出给 MAC	与 MAC RMII (TXC 输入) 接口相连
1	1	RMII1	TXC 由 MAC 输出给 PHY	与 MAC RMII (TXC 输出) 接口相连

### 2.8.1 MII 与 REMII 模式

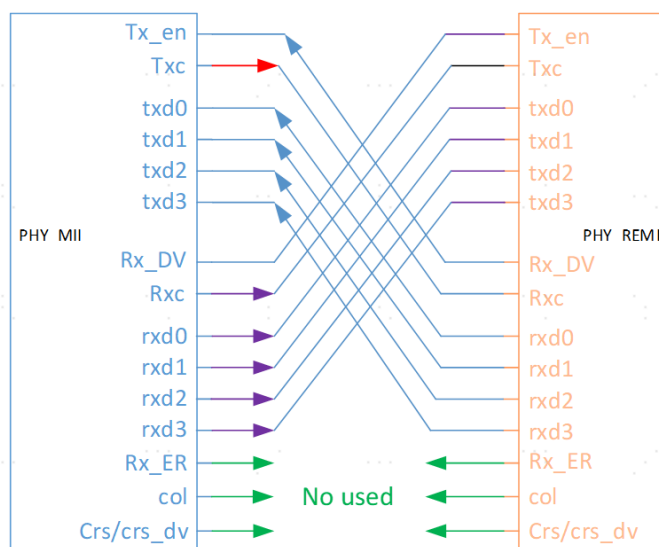
YT8522 工作在 MII 状态下，RX\_DV 和 RXD3 外接 4.7kΩ 电阻到地。

YT8522 工作在 REMII 状态下，RX\_DV 外接 4.7kΩ 电阻到地，RXD3 接 4.7kΩ 电阻到 3.3V。

**Figure 8** YT8522 工作在 MII 状态



**Figure 9** YT8522 工作在 REMII 状态



## 2.8.2 RMII1 与 RMII2 模式

PHY 的 RMII 模式分为 RMII1 和 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。



RMII 模式用到 50MHz 的 RMII TXC, 为节省 PHY 的晶体, 可能会把 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/RMII2 (25MHz 时钟) 和 RMII1 (外灌 50MHz 时钟)。

- PHY 工作在 RMII1 Mode 时, 需要配置如下寄存器。

```
write_ext_reg0x4000[4]: 1'b1
```

- PHY 工作在 RMII2 Mode 时, 需要配置如下三个寄存器。

```
write_ext_reg0x4000[4]: 1'b1;
```

```
write_ext_reg0x19 0x9f;
```

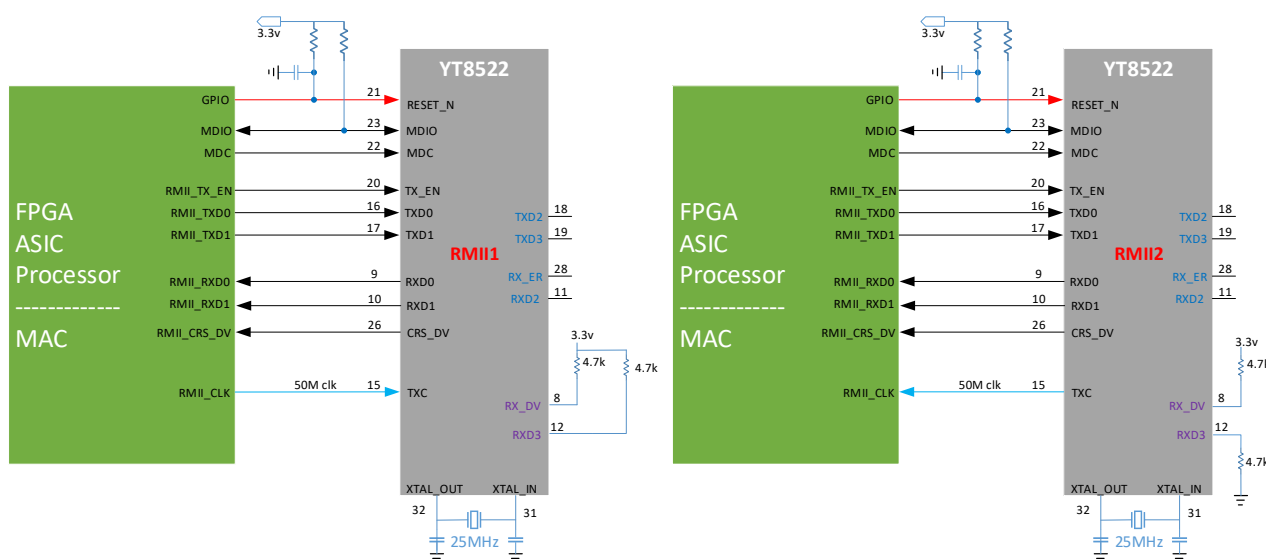
```
write_ext_reg0x4001 0x81d4;
```

### 2.8.2.1 RMII1/RMII2 (25MHz 时钟)

25MHz 参考时钟下, RMII1/RMII2 两种模式的参考时钟仅支持 XI XO 管脚输入。

以 25MHz 晶体作参考时钟情况下 (外灌 25MHz 时钟信号与此类似), PHY 与 MAC 的连接方式如下:

Figure 10 RMII1/RMII2 模式 (25MHz)



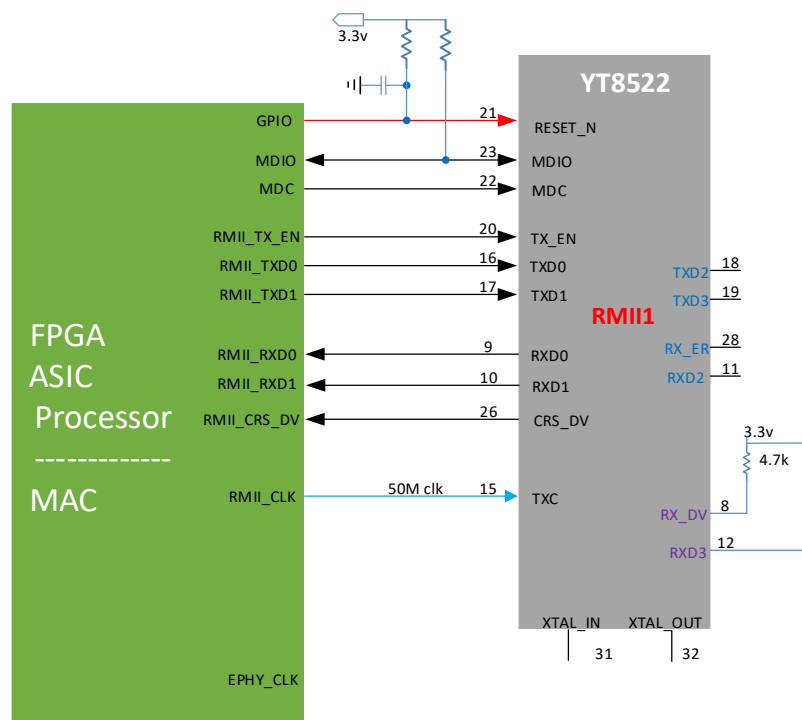
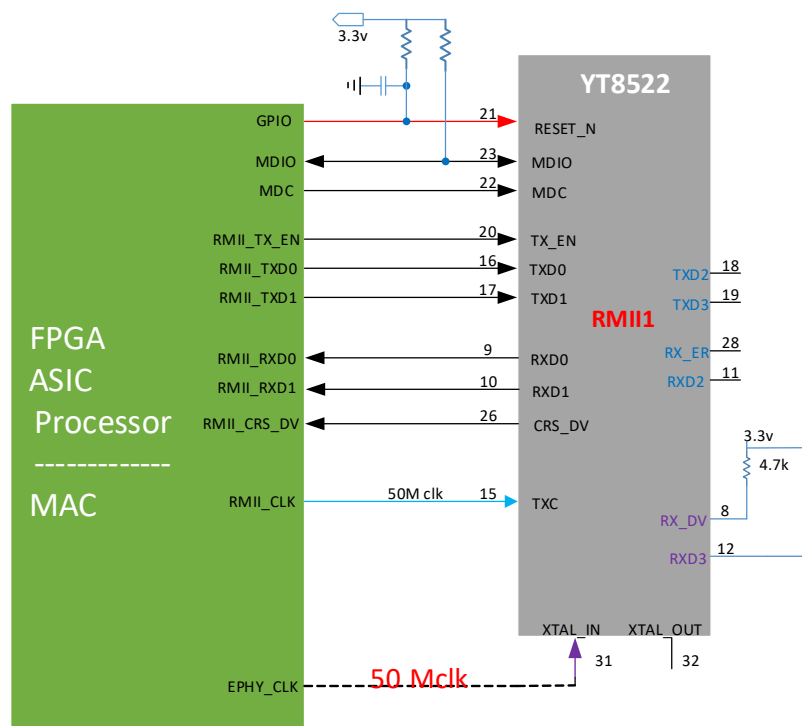
**Caution:** PHY 的 CRS\_DV 与 MAC 相连, 而非 RXDV。

### 2.8.2.2 RMII1 (50MHz 时钟)

50MHz 参考时钟下, YT8522 仅支持 RMII1。在 RMII1 模式下, 50M 内部参考时钟来源有如下两种:

- 来自于外部 50M
- 来自 MAC 给到 PHY 的 50M TXC

以 50M TXC 以及外部 50M 时钟为例 (用 50MHz 晶体与此类似), PHY 与 MAC 的两种连接方式如下:

**Figure 11 RMII1 模式 ( 50M TXC )**

**Figure 12 RMII1 模式 ( 外部 50M 时钟 )**


硬件 POS 设置：RXDV 和 RXD3 外接上拉

### Caution:

- PHY 的 CRS\_DV 与 MAC 相连，而非 RXDV。

- 上电时的电压、时钟、复位时序需达到 2.2 上电时序提出的要求。因为时钟由 CPU 提供，所以 PHY 的复位管脚也应该由 CPU 控制，方便调节先后以满足上电时序要求。
- 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。

### 2.8.3 MII/RMII 电压配置

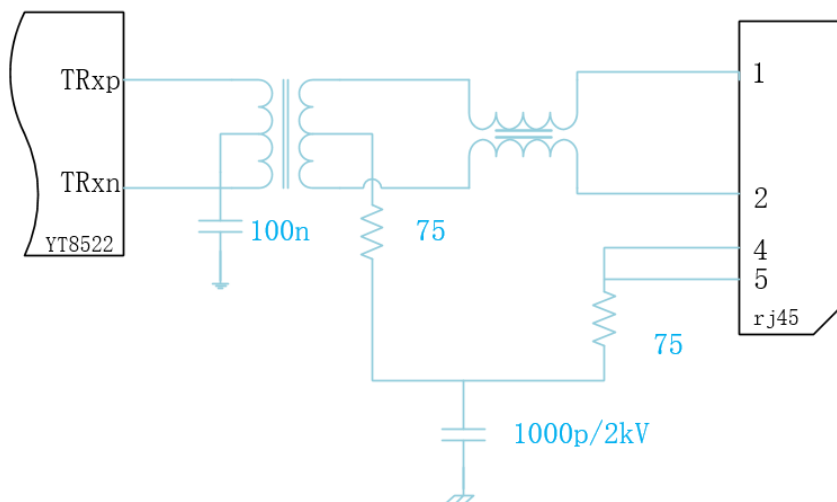
YT8522 的 MII/RMII 电压一共有 4 种配置，可以将总线电压配置为 1.5V、1.8V、2.5V、3.3V。具体的配置方式可以通过调节 ext\_reg0x4216 bit[1:0]实现。

- 11 = 3.3V
- 10 = 2.5V
- 01 = 1.8V
- 00 = 1.5V

## 2.9 MDI 接口

YT8522 内置端口自转功能，能自动识别直连线或交叉网线。YT8522 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 100nF 电容接地。

Figure 13 MDI 接口

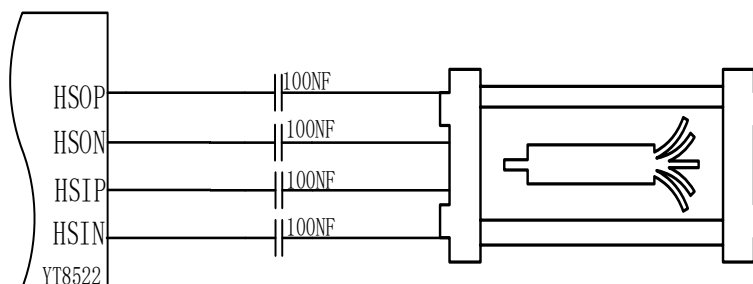


- 上图只列出一对 MDI 线的接法，另一对 MDI 线接法相同。
- 由于百兆只用两对线，对于 RJ45 上另外两对线，可按上图 RJ45 pin 4, 5 的接法。
- UTP 与 fiber 共用管脚，使用时需要通过 RXER/FXEN 管脚设置模式具体 UTP/FIBER 模式，此管脚内部存在弱下拉，但是推荐外部加 4.7kΩ 下拉电阻以确保进入 UTP 模式。

## 2.10 Fiber 接口

UTP 与 Fiber 共用管脚，使用时需要通过 RXER/FXEN 管脚设置模式。此管脚内部存在弱下拉，但是推荐外部加 4.7kΩ 上拉电阻以确保进入 Fiber 模式。

**Figure 14 Fiber 接口**



## 2.11 硬复位后延时进行 MDIO 操作

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，因此要求在硬复位释放至少 100ms 后再进行 MDIO 操作。

## 2.12 外接参考时钟的抖动

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。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，如幅度、占空比等参见 datasheet。

## 2.13 外接参考时钟源切换操作

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：

- 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。
- 硬复位的时序，参考复位时序要求，即 RESET 信号应该在时钟稳定后保持为低至少 10ms，再拉高/释放。
- 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）。

## 2.14 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，以及时间的抖动范围（Variation）。

**Table 4** 数据传输通路延时

Mode	Direction	Speed	Basic Delay (ns)	Variation (ns)
mii_mode	mii_tx->utp_tx	100M	54	0
	utp_rx ->mii_rx	100M	216	0
	mii_tx->utp_tx	10M	740	0
	utp_rx ->mii_rx	10M	6910	230
rmii1_mode	rmii1_tx->utp_tx	100M	298	18
	utp_rx ->rmii1_rx	100M	430	6
	rmii1_tx->utp_tx	10M	2490	90
	utp_rx ->rmii1_rx	10M	8940	200
rmii2_mode	rmii2_tx->utp_tx	100M	300	16
	utp_rx ->rmii2_rx	100M	440	14
	rmii2_tx->utp_tx	10M	2420	180
	utp_rx ->rmii2_rx	10M	8920	260
remii_mode	remii_tx->utp_tx	100M	342	16
	utp_rx ->remii_rx	100M	490	10
	remii_tx->utp_tx	10M	3150	130
	utp_rx ->remii_rx	10M	9250	270

**Note:** 上表中的数值有一定误差，具体数据以实测结果为准。

## 3 寄存器类型及访问方法

系统对 YT8522 的配置管理、状态获取均依赖 MDIO 接口通信读写 YT8522 内部寄存器实现。

### 3.1 寄存器类型

YT PHY 内部有三种类型的寄存器：

- MII 寄存器（MII Register，下文简称为 mii\_reg）
- 扩展寄存器（Extended register，下文简称为 ext\_reg）
- MMD 寄存器（下文简称为 mmd\_reg）。

#### 3.1.1 MII 寄存器

MI 寄存器地址为 0x0~0x1f，遵从 IEEE802.3 clause 22 的标准定义进行访问。MDC/MDIO 协议如下：

Frame	Management Frame Fields							IDLE
	PRE (32bit)	ST	OP	PHYAD	REGAD	TA	DATA	
Read	1.....1	01	10	PHYAD[4:0]	REGAD[4:0]	Z0	16bit	Z
Write	1.....1	01	01	PHYAD[4:0]	REGAD[4:0]	10	16bit	Z

#### 3.1.2 扩展寄存器

由于 MII 寄存器只有 32 个，为满足更多功能需求，PHY 芯片配置了更多的扩展寄存器。可通过 mii\_reg 0x1e, 0x1f 间接访问，具体访问方法如下：

- 1 将要访问的 ext\_reg 地址，写入 mii\_reg 0x1e。
- 2 读/写 mii\_reg 0x1f。

例如：

- 1 读取 ext\_reg 0x1000 的值

```
write mii_reg 0x1e = 0x1000;
read mii_reg 0x1f;
```

- 2 写 ext\_reg 0x1000 的值为 0x3456

```
write mii_reg 0x1e = 0x1000;
write mii_reg 0x1f = 0x3456;
```

#### 3.1.3 MMD 寄存器

为了扩展更多的寄存器，以适应更高速率的以太网，IEEE802.3 clause45 定义了 MMD 寄存器。mmd\_reg 有直接和间接两种访问方式。

#### 直接访问

遵从 IEEE802.3 clause 45 的标准定义进行访问，通过两条指令实现 MMD 寄存器的读或写，先发 address 指令，再发 Write 或 Read 指令。具体协议如下：

	Management Frame Fields							
Frame	PRE (32bit)	ST	OP	PRTAD	DEVAD	TA	DATA	IDLE
Address	1.....1	00	00	PRTAD[4:0]	DEVAD[4:0]	10	ADDR[15:0]	Z
Write	1.....1	00	01	PRTAD[4:0]	DEVAD[4:0]	10	DATA[15:0]	Z
Read	1.....1	00	11	PRTAD[4:0]	DEVAD[4:0]	Z0	DATA[15:0]	Z
Post-read increment address	1.....1	00	10	PRTAD[4:0]	DEVAD[4:0]	Z0	DATA[15:0]	Z

## 间接访问

通过 mii\_reg 0xd、0xe 间接访问 mmd\_reg，具体指令如下：

```
write mii_reg 0xd = device address;
write mii_reg 0xe = reg address;
write mii_reg 0xd = 0x4000 + device address;
write mii_reg 0xe = data; 或 read mii_reg 0xe;
```

例如：

- 读 MMD 3 的 0x5 寄存器

```
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

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

相关寄存器定义如下：

**Table 5 MMD Access Control Register (PHY mii\_reg 0x0D)**

Bit	Symbol	Access	Default	Description
15:14	Function	RW	0x0	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	0x0	Always 0
4:0	DEVAD	RW	0x0	MMD register device address. 00001 = MMD1 00011 = MMD3 00111 = MMD7

**Table 6 MMD Access Data Register (PHY mii\_reg 0x0E)**

Bit	Symbol	Access	Default	Description
15:0	Address data	RW	0x0	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.

## 3.2 寄存器软件访问方式

ext\_reg 和 mmd\_reg 都可以通过 mii\_reg 来间接访问，具体参考 3.1 寄存器类型，本节主要描述软件上寄存器访问的实现方法，供参考。

### 3.2.1 MII 寄存器的读写

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。

#### 3.2.1.1 read\_mii\_reg

##### 【定义】

```
read_mii_reg(phy_addr, reg_addr)
```

##### 【参数说明】

参数	说明
phy_addr	PHY 地址
reg_addr	寄存器地址

##### 【返回值】

读取到的 mii\_reg\_addr 寄存器的值。

#### 3.2.1.2 write\_mii\_reg

##### 【定义】

```
write_mii_reg(phy_addr, reg_addr, data)
```

##### 【参数说明】

参数	说明
phy_addr	PHY 地址
reg_addr	寄存器地址
data	寄存器值



### 【返回值】

无。

## 3.2.2 扩展寄存器的读写

### 3.2.2.1 read\_ext\_reg

#### 【定义】

```
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
```

#### 【参数说明】

参数	说明
phy_addr	PHY 地址
reg_addr	寄存器地址

### 【返回值】

读取到的扩展寄存器 reg\_addr 内的值。

### 3.2.2.2 write\_ext\_reg

#### 【定义】

```
def write_ext_reg(phy_addr, reg_addr, data)  
    write_mii_reg(phy_addr, 0x1e, reg_addr)  
    write_mii_reg(phy_addr, 0x1f, data)
```

#### 【参数说明】

参数	说明
phy_addr	PHY 地址
reg_addr	寄存器地址
data	寄存器值

### 【返回值】

无。

### 3.2.3 MMD 寄存器的读写

#### 3.2.3.1 read\_mmd\_reg

##### 【定义】

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

##### 【参数说明】

参数	说明
phy_addr	PHY 地址
device	MMD 寄存器编号
reg_addr	寄存器地址

##### 【返回值】

读取到的 MMD 寄存器的值。

**Note:** 例如 MMD 3 reg0x01，device 即为 3，reg\_addr 即为 0x01。

#### 3.2.3.2 write\_mmd\_reg

##### 【定义】

```
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)
```

##### 【参数说明】

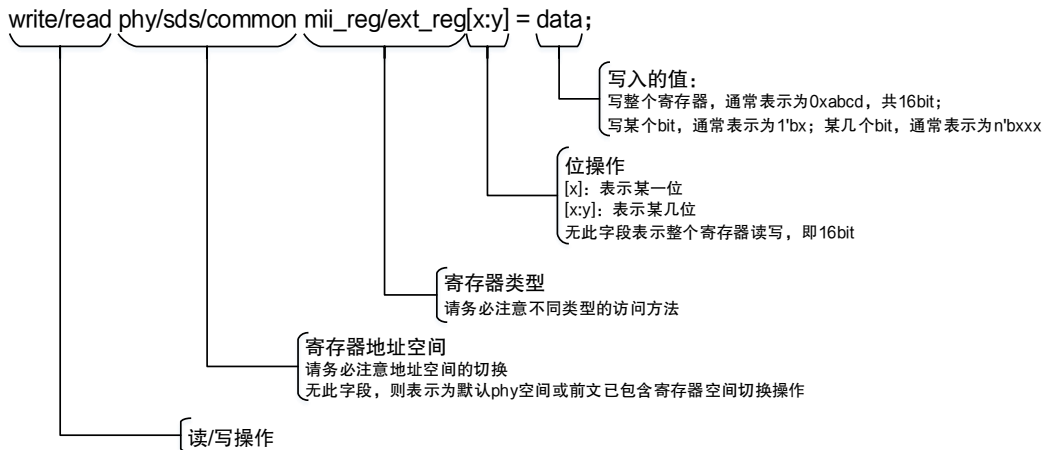
参数	说明
phy_addr	PHY 地址
device	MMD 寄存器编号
reg_addr	寄存器地址
data	寄存器值

## 【返回值】

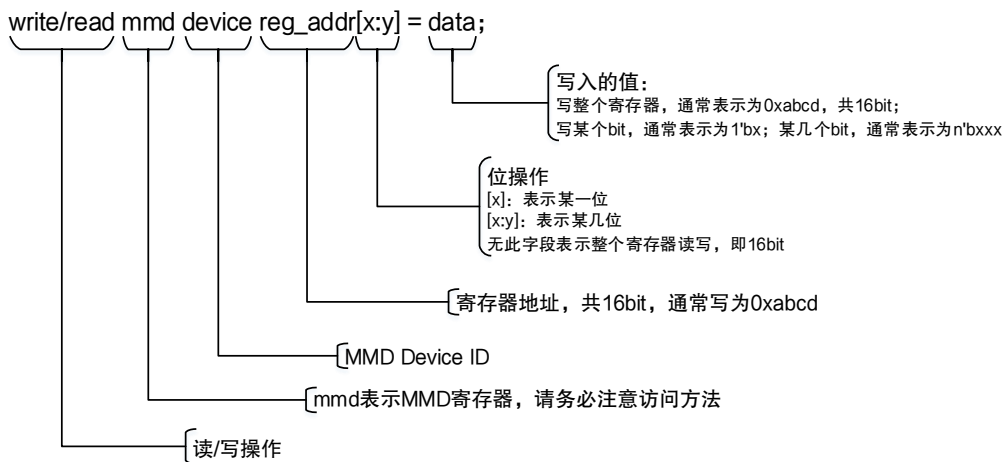
无。

## 3.3 读写寄存器的表示方法

### mii\_reg 和 ext\_reg 读写



### mmd\_reg 读写



**Note:** 针对某几个 bit 的读写而其它位需要保持原值不变，通常需要先读出原值，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。

以 mii\_reg0x1 和 ext\_reg 0xa 为例进行说明：

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

```
write mii_reg 0x1[15] = 1'b0
```

- 读 MII 寄存器 0x1 的值并获取 bit15（共 1 比特）。

```
read mii_reg 0x1[15]
```

- 将扩展寄存器 0xa 的 bit[15:13]（共 3 比特），分别写为 0，1，0，其余比特位保持不变。

```
write ext_reg 0xa[15:13] = 3'b010
```

- 读扩展寄存器 0xa 的值并获取 bit[15:13]。

```
read_ext_reg 0xa[15:13]
```

### 3.4 PHYADDR0 和 BDCST\_ADDR

YT8522 有 phyaddr0 和 bdcst\_addr 两种特色功能，可以方便、快捷地访问 YT8522 的寄存器。

当 phyaddr0 enable 时，YT8522 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]的说明参考 Datasheet。

- enable phyaddr0

```
write_ext_reg0x0[6]: 1'b1 #默认 enable
```

- disable phyaddr0

```
write_ext_reg0x0[6]: 1'b0
```

当 bdcst\_addr enable 时，YT8522 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。

- enable bdcst\_addr

```
write_ext_reg0x0[5]: 1'b1 #默认 enable
```

write\_ext\_reg0x0[4:0]配置相应的 broadcast address，默认 Bdcst\_addr = 0x1f。

- disable bdcst\_addr

```
write_ext_reg0x0[5]: 1'b0
```

#### Caution:

- 当 bdcst\_addr enable 时，YT8522 不会响应 phy address = bdcst\_addr 的读命令，只响应写命令。
- 当 phyaddr0 enable 时，YT8522 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，此时读回来的值不可靠。

**Table 7 EXT 0000 Register**

Bit	Symbol	Access	Default	Description
15:7	Reserved	RO	0	Reserved
6	En_phyaddr0	RW	1'b1	Enable MDIO PHY address 0 access
5	En_bdcst_addr	RW	1'b1	Enable MDIO broadcast address access
4:0	Bdcst_addr	RW	5'b11111	MDIO Broadcast address

## 4 寄存器功能配置

### 4.1 初始化配置

为了优化 DUT 在实际应用场景下的性能，需要进行初始化配置。若不做特别说明，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。

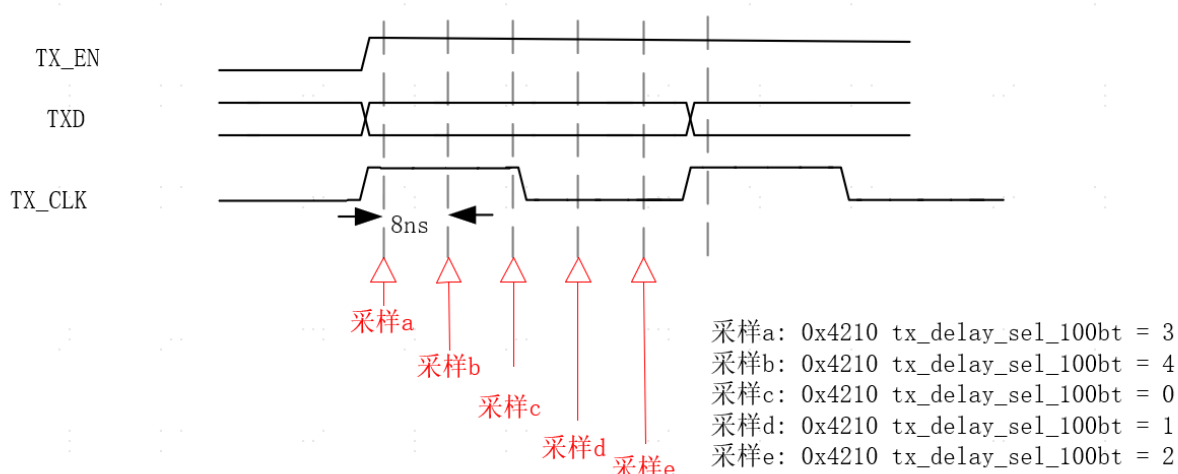
使用 RMII1/RMII2 模式时，参照 2.8.2 RMII1 与 RMII2 模式配置相关寄存器（如不使用 RMII 模式，无需配置）。上电复位后，YT8522 还需要做如下初始化配置：

```
write_ext_reg0x4210 0x0;           #内部 tx clk delay（MII mode 配置，RMII mode 无需配置）
write_ext_reg0x4008 0xbf2a;        #提升差分电压幅度
write_ext_reg0x2057 0x297f;        #调整差分电压幅度
write_ext_reg0x14 0x1fe;           #调整波形上升下降时间
write_ext_reg0x15 0x1fe            #调整波形上升下降时间
```

**Note:** reg0x4210 是 MII 模式特有的芯片内部采样（下降沿采样）控制 delay 的寄存器，外部 tx clk 无变化。

**Table 8 Ext\_Reg 0x4210**

Symbol	Description
txclk_delay_sel_100bt[2:0]	0x1: use 8ns delayed tx mii clock 0x2: use 16ns delayed tx mii clock 0x3: use 24ns delayed tx mii clock 0x4: use 32ns delayed tx mii clock others: use no delayed tx mii clock



### 4.1.1 MII/RMII

MIIR/MII 一般采用 POS 方式上电完成模式选择，配置管脚为 RX\_DV 和 RXD3。

POS 配置方法如下：

- {RX\_DV, RXD3}=2'b00: MII mode
- {RX\_DV, RXD3}=2'b01: REMII mode
- {RX\_DV, RXD3}=2'b10: RMII2 mode, TXC 50MHz reference clock is output.
- {RX\_DV, RXD3}=2'b11: RMII1 mode, TXC 50MHz reference clock is input.

除 POS 方式配置以外，还可以通过 ext\_reg0x4000[1:0]配置选择 MII/RMII：

- 2'b00: MII mode
- 2'b01: REMII mode
- 2'b10: RMII2 mode
- 2'b11: RMII1 mode

**Note:** 无论使用 POS 还是寄存器配置方式，使用 RMII1/2 模式时，都需要先将 ext\_reg0x4000 bit4 置 1。

### 4.1.2 100Base-TX/100FX

100Base-TX/100FX 支持 POS 方式上电完成模式选择，配置管脚为 RX\_ER/FXEN：

- RX\_ER/FXEN=1'b0: 100Base-TX
- RX\_ER/FXEN=1'b1: 100FX

除 POS 方式配置以外，还可以通过 ext\_reg0x4000[3]配置选择 100Base-TX/100FX：

- 1'b0: 100Base-TX
- 1'b1: 100FX

### 4.1.3 IO 电平

YT8522 的 MII/RMII 电压一共有 4 种配置，可以将总线电压配置为 1.5V、1.8V、2.5V、3.3V；具体的配置方式可以通过调节 ext\_reg 0x4216 bit1,0 实现。

寄存器配置 ext\_reg0x4216[1:0]与 4 种 io 电压对应关系

- 2'b00: 1.5V
- 2'b01: 1.8V
- 2'b10: 2.5V
- 2'b11: 3.3V

DVDDIO 为外部供电，电压需改为 1.5V、1.8V、2.5V、3.3V，同时调节 ext\_reg0x4216 才能实现不同电压配置。

## 4.2 UTP 配置

### 4.2.1 10/100BT 的速度和双工

YT8522 不需做任何配置，默认开启自协商，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。

- 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
```

- 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
```

#### Caution:

- 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。
- 在自协商模式下，只是本地 UTP 速度和双开能力的打开或关闭，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。
- 自协商模式下，切换百兆全双工（mii\_reg\_0x0 0x2100）或者百兆半双工（mii\_reg\_0x0 0x2000），需要切换 2 次才能生效。

获取 PHY 当前的连接速度和模式，有如下 2 种方法：

- 通过标准的 PHY MII 寄存器来判断当前是否 link up (phy mii\_reg 0x1[2])，通过 phy mii\_reg 0x4, 0x5, 0x9 等标准寄存器和算法得到当前 PHY 的连接速度和双工。
- 为了方便地判断当前 PHY 是否连接，获取速度与双工状态，YT PHY 将这些信息放在 phy mii\_reg 0x11（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。0x11 的部分 bit 含义如下：

Bit	Symbol	Description
15:14	Speed mode	11: Reserved 01: 100M 00: 10M
13	Duplex	1 = Full-duplex 0 = Half-duplex
10	Link status real-time	1 = Link up 0 = Link down

**Note:** 需要先判断 bit 10 为 1（表示连接建立），再通过 bit[15:13]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，则上层可进行网络通信。注意轮询本地 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。

## 4.2.2 Power Down

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：

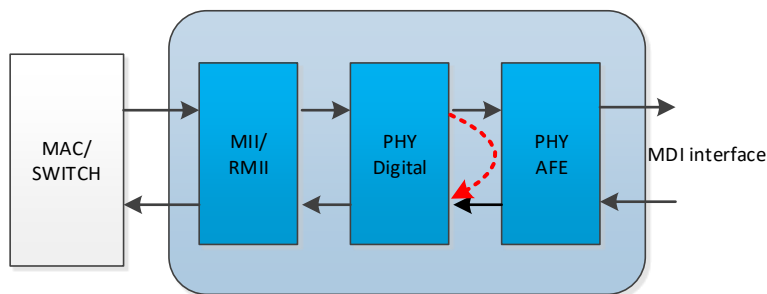
- 进入 UTP power down: phy mii\_reg0x0 bit11 置 1
- 退出 UTP power down: phy mii\_reg0x0 bit11 置 0, phy mii\_reg0x0 bit15 置 1 进行软复位操作。

## 4.2.3 Loopback 模式

### 4.2.3.1 Internal Loopback

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。

Figure 15 Internal Loopback 框图



配置方法如下：

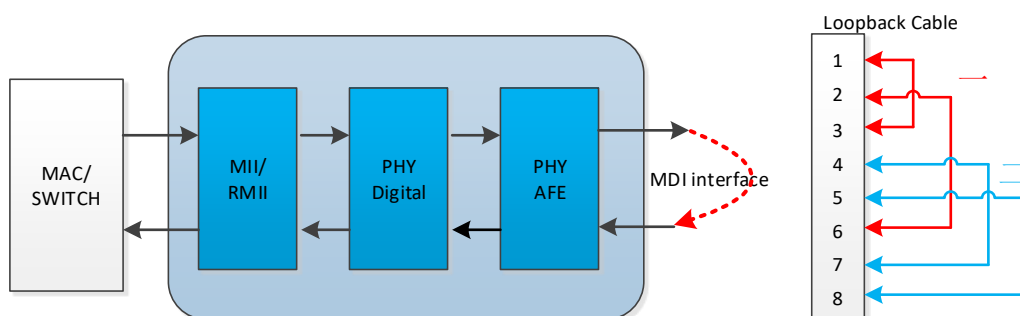
- 10M: utp phy mii register 0x0 = 0x4100
- 100M: utp phy mii register 0x0 = 0x6100

**Caution:** 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。

### 4.2.3.2 External Loopback

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 通过比对 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 Figure 16 所示。

Figure 16 External Loopback 框图





配置方法如下：

1 关闭 sleep 模式

```
write_bits phy_ext_0x2027[15] 1'b0
```

2 使能 External Loopback 模式

```
utp phy ext register 0x4000.12= 1
```

3 设置 External Loopback 连接速率

- 10M

```
mii register 0x0 = 0x8100
```

- 100M

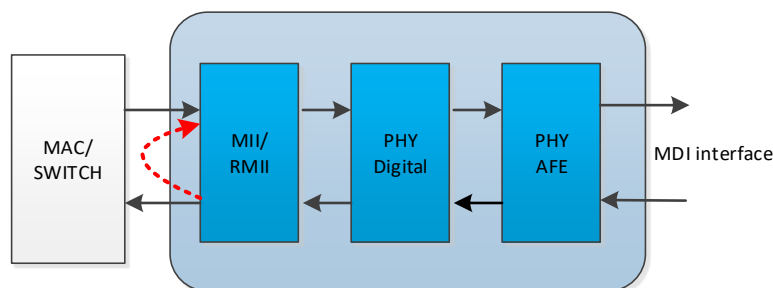
```
mii register 0x0 = 0xa100
```

**Note:** 配置 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。如果在第一次软复位后才插上 external loopback cable，需要再次执行软复位。

### 4.2.3.3 Remote Loopback

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。

**Figure 17 Remote Loopback 框图**



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
```

## 4.2.4 自动降速

对于以太网 PHY，当插上网线时，默认从双方支持的最高速率开始尝试连接。有时由于外界干扰、连接线太长、质量太差等原因，导致高速率连接不上。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 downgrade（或者 auto-downgrade）被称为 PHY 的自动降低连接速度的功能。

例如，双方都按 100BT 的能力发起自协商，但网线超过最大连接距离（例如 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。

### Note:

- 如果对端口快速插拔，导致短时间内不断 link up/down，也会触发 smart downgrade 机制。
- smart downgrade 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。
- 若要禁用 smart downgrade，需网线两端 PHY 都禁用才可以，否则任一边自动降速会导致连接降速。

如果不需要 smart downgrade，需要将此功能通过寄存器禁用。寄存器 phy\_mii\_0x14[5]=1'b0，然后做软复位（phy\_mii\_0x0[15]=1'b1）使其生效。

**Table 9 Speed Auto Downgrade Control Register (Phy MII 14h)**

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 takes effect after the software reset
4:2	Reserved	RW	0x3	Reserved
1	Reserved	RW	0x0	Reserved
0	Reserved	RO	0x0	Reserved

## 4.2.5 自动睡眠

PHY 处于不连接状态超过一定时间后（约 40 秒），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 功能默认打开，此功能不会影响 PHY 的正常连接。如果关闭 sleep 功能也不会有任何影响。

开启 sleep 功能：

```
write_ext_reg0x2027[15]: 1'b1
```

**Table 10 Sleep Control1 (Phy EXT 2027h)**

Bit	Symbol	Access	default	Description
15	En_sleep_sw	RW	0x1	1 = enable sleep mode 0 = disable sleep mode If in sleep mode, PHY will close AFE after unplugging the cable for a timer.
14	Pllon_in_slp	RO	0x0	1: keep PLL on in sleep mode 0: close PLL in sleep mode
13	Slp_pulse_sw	RW	0x1	when PHY enters sleep 1: enable PHY to send out one pulse periodically 0: disable PHY to send pulse
12	En_upd_afe_sbs	RW	0x0	When AFE control is changed, no matter if 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 0 = normal
4	Gate_25m	RO	0x0	Not used.
3:0	Slp_state	RO	0x0	FSM state of internal sleep control logic.

## 4.2.6 包生成器和收发包统计

包生成器（packet checker）：

- 打开/关闭 packet checker

```
write_bits ext reg 0x40A0.[15:14]=2'b10    #打开
write_bits ext reg 0x40A0.[15:14]=2'b01    #关闭
```

- 查看 checker 统计结果

- Ext reg 0x40A3~0x40A8：从 MDI 收到的正确包的数量，寄存器为 read clear。
- Ext reg 0x40A9~0x40AC：从 MDI 收到的包有 crc error，寄存器为 read clear。
- Ext reg 0x40AD~0x40B2：从 MII/RMII 收到的正确包的数量，寄存器为 read clear。
- Ext reg 0x40B3~0x40B6：从 MII/RMII 收到的包有 crc error，寄存器为 read clear。

收发包统计（packet generator）：

- 设置 packet generator 的发送包的长度。

```
write_ext_reg 0x40A1[15:0] value
```

- 设置发送包的个数。

```
write_ext_reg 0x40A2[15:0] value    #值为 0 表示一直发包。
```

- 设置 IPG 长度，不用时可以不设置。

```
write_ext_reg 40A0[7:4] value
```

#### 4 打开 packet generator。

- 配置 packet checker 和 generator

```
write ext reg 0x40A0[15:13] = 3'b100
```

- 使能 packet generator 开始发包

```
write ext reg 0x40A0[12] = 1'b1
```

Polling read ext reg 0x40A0[12], 为 1'b0 时, 当包个数设置为非 0 时, 表示发完了设置的包个数; 当包个数设置为 0 时, 表示停止发包。

#### 5 关闭 packet generator。

- 关闭 packet generator clock

```
write ext reg 0x40A0[14]: 1'b1
```

- 关闭 generator 和 checker

```
write ext reg 0x40A0[15, 13, 12]: 3'b010
```

### 4.2.7 SNR 读取

YT8522 的 phy ext 扩展寄存器 0x205A 的 bit[14:0] 值表征了一对 MDI 线的 SNR。读取方法如下:

#### 1 将 phy ext 扩展寄存器 0x2059 的 bit15 置 1。

#### 2 读取 MSE。

分别读取 phy ext 扩展寄存器 0x205A 的 bit[14:0], 记为 MSE(i), i=0。每个值读至少 100 次, 并取平均值, 得到 MDI 线上稳定 MSE。

#### 3 计算 SNR。

100M:  $10 \cdot \log_{10}(32768/\text{MSE})$

要达到稳定连接, SNR 值应大于 19.13dB (100M 对应 MSE 约 400)。

### 4.2.8 线缆状态检测

在进行 CSD 测试之前, 需要先确认具体检测方式, 线对检测方式分为 INTRA check 和 INTER check。

- INTRA check: 0 发 0 收, 1 发 1 收
- INTER check: 0 发 1 收, 1 发 0 收

具体测试方法如下:

#### 1 设置好连线 (open 状态或者 short 状态)。

#### 2 配置 Force 100M 模式。

设置 reg0x10 0x2

设置 reg0x0 0xa100

设置 reg0x0 0x2100

#### 3 设置 ext reg0x4108 (初始设置 0xc28), 配置 CSD 初始化阈值。

#### 4 设置 ext reg0x4102 0x9446

设置 ext reg0x4107 0x82b

#### 5 打开 CSD 测试。

设置 ext reg0x4100 0x8000

#### 6 读取 ext reg0x4100，当值为 0 时，CSD 检测结束。

#### 7 CSD 检测结束后，读取检测结果。

读取寄存器 ext reg0x410b，ext reg0x410c，ext reg0x410d，ext reg0x410e，410b/410c 为 intra check，410d/410e 为 inter check。

- 其中 bit13 为 1 表示 open
- bit12 为 1 表示 short
- bit[13:12]为 00 表示 normal
- bit[8:0]为故障点线长

#### 8 根据测距调整 near\_echo\_positon 的取值，重新设置 ext reg0x4108 以符合实际线长。重复 5~7，读取线长结果。

near\_echo\_positon：指的是信号在芯片内部的各种延迟量（包括输出延时与输入延时），如果该值设置的偏大（大于实际值），那么测距值将偏小，反之如果该值设置地偏小，那么测距值则偏大。实际中可以根据真实线长来对该值进行调整，以使得测距值符合真实线长。

## 4.2.9 Template

YT8522 的 Template 测试指令如下：

#### • 初始化配置

```
write_ext_reg0x14: 0x1fe
write_ext_reg0x15: 0x1fe
write_ext_reg0x2057: 0x297f
write_ext_reg0x4008: 0xbf2a
```

#### • 100BT

```
write_ext_reg0x2027: 0x2026
write_mii_reg_0x10: 0x2
write_mii_reg_0x0: 0xa100
write_mii_reg_0x0: 0x2100
```

#### • 10BTc

```
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
```

考虑到不同的板级设计，PHY 内部也有丰富的配置方便调整，其中常用的几项调整如下：

- 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。

- 调整 rise/fall time

可以调节 ext\_reg 0x11-15 优化 rise/fall time，其中 bit[15:8]为光口调节，bit[7:0]为电口调节。

## 4.3 100FX 配置

### 4.3.1 Fiber

YT8522 可以工作在 UTP/Fiber 两种模式，具体方式通过 RXER/FXEN（PIN28）配置。RXER/FXEN 上拉进入 Fiber 模式，下拉进入 UTP 模式。YT8522 上电默认 UTP 模式（RXER/FXEN 内部有弱下拉），如果需要进入 Fiber 模式，可以通过以下两种方式：

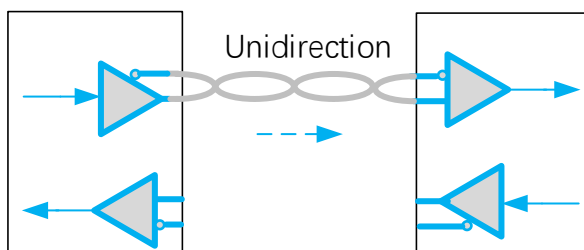
- 硬件方式：将 RXER/FXEN 管脚上拉（推荐 4.7K 电阻）
- 软件方式：配置 ext reg0x4000 bit3 1。

### 4.3.2 Unidirection

在 Fiber 应用中，有时需要单向传递数据，可通过如下配置实现 Unidirection 100M。

- 1 mii\_reg\_0x0 设置为 0x2120
- 2 对端配置为 Fiber 100M

Figure 18 Unidirection



### 4.3.3 100FX 眼图测试

YT8522 的眼图测试，大体分为模版测试（mask 测试）和抖动测试（jitter 测试），以此评估 serdes 发送信号的 BER 和抖动等。

对于 YT8522 的眼图测试结果，重点关注如下参数：

- 抖动 Tj@BER

一般 BER 要求  $10E-12$ 。通过浴盆曲线可看到在 BER 为  $10E-12$  的概率下眼宽  $> 70\%UI$ ，即  $T_j$  小于  $0.3UI$ 。Dj, Rj, DDj 等一般用于 debug，即  $T_j$  不满足时确定是哪部分抖动造成的。

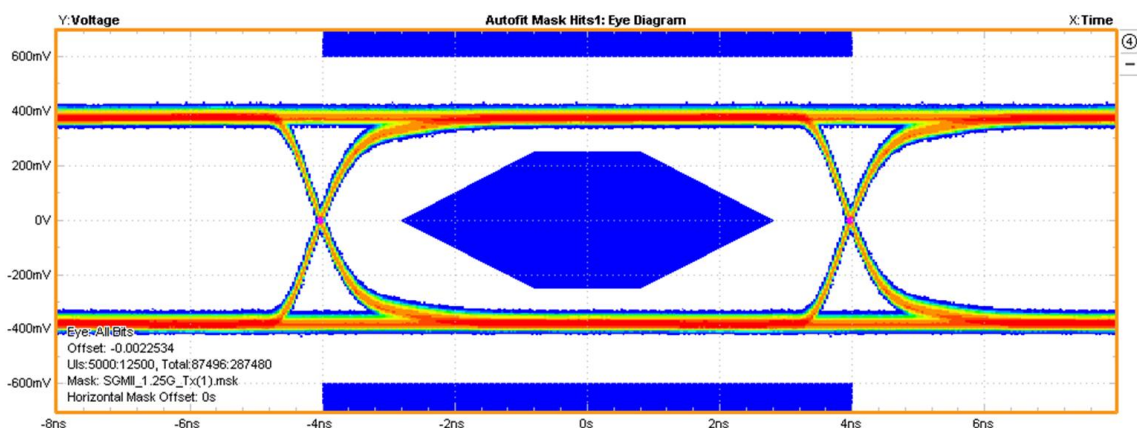
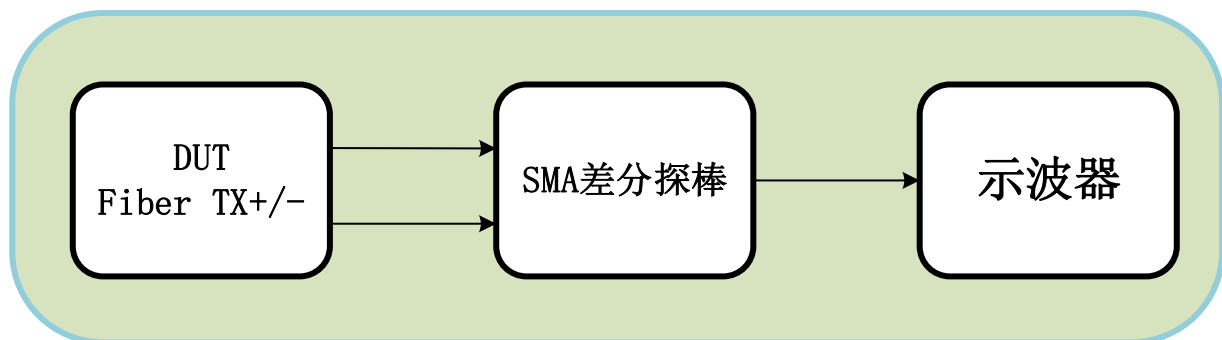
- 眼图 mask。

在时序波形上显示发送信号质量，要求发送眼图不能打到 mask 覆盖的区域。当无限时间测量累加时，眼图会闭合，所以要有定量的测量点数要求。一般要求  $> 1M$  sample 的情况下，能过模版测试即可。

测试环境要求如下：

- DUT：配置 YT8522 serdes 发送特定码型，如 prbs31 或其默认输出码型。
- SMA 线/差分探棒：要求其有效带宽远大于被测信号带宽，减少传输损耗。
- 示波器：
  - Mask 文件：示波器内置 mask 文件或自制 mask 文件。
  - 眼图和抖动测量分析工具：Jitter And Eye Analysis，分为 One Touch Jitter 和 Serial Data/Jitter Wizard。One Touch Jitter 方便用户测量，许多配置用默认设置，准确性不如 Serial Data/Jitter Wizard，因此一般用 Serial Data/Jitter Wizard 测量。

Figure 19 测试拓扑





## 4.4 MII/RMII 配置

### 4.4.1 驱动能力

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。TXC 配置 `write_ext_reg0x4002[12:10]`，默认值为 `3'b001`。

- RXC 配置: `write_ext_reg0x4002[9:7]`，默认值为 `3'b001`
- RXD 配置: `write_ext_reg0x4002[6:4]`，默认值为 `3'b001`

MIITXC/RXC/RXD 驱动强度分别由 3bit 控制，共分为 6 档，由强到弱的配置如下：

- 第 1 档: `0x100`
- 第 2 档: `0x101`
- 第 3 档: `0x110&0x001&0x000`
- 第 4 档: `0x111`
- 第 5 档: `0x011`
- 第 6 档: `0x010`

### 4.4.2 MII 模式下 Tx\_delay

MIIT 模式下，TXC/RXC 时钟都由 YT8522 提供。在通讯时，需要注意 TXC/TXD 之间的 Tsetup/Thold 是否满足通讯要求。YT8522 支持调节 Tx\_delay 来优化 Tsetup/Thold，通过 `ext_reg 0x19[4:0]` 配置。一共有 32 个档位可调，步进满足 150-300ps，跨度 4.7ns 左右。

## 4.5 LED 灯配置

YT8522 支持 LED0 和 LED1 两个 LED 灯，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 `0x031`。
- LED1: 对应的寄存器为 `ext Reg0x40C3`，上电默认值为 `0x1300`。

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

**Table 11 LED 典型配置**

extReg0x40C0 配置值	extReg0x40C3 配置值	动作行为
0x11	0x0311	link 在 10M 时，LED0/1 常亮； link 在 10M 且收发包，LED0 常亮，LED1 闪。
0x30	0x0320	link 在 100M 时，LED0/1 常亮； link 在 100M 且收发包，LED0 常亮，LED1 闪。
0x0311	0x0311	link 在 10M 时，LED 常亮；



extReg0x40C0 配置值	extReg0x40C3 配置值	动作行为
		link 在 10M 且收发包，LED 闪。
0x0320	0x0320	link 在 100M 时，LED 常亮； link 在 100M 且收发包，LED 闪。
0x30	0x30	link 或收发包时，LED 常亮； link down 时，LED 灭。
0x1300	0x1300	收发包时，LED 闪烁；不收发时，即灭。

## 4.6 中断配置

YT8522 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：

- 中断管脚，低有效。
- 配置 UTP mii reg0x12 寄存器，选择触发中断信号的事件。
- 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。

**Table 12 Interrupt Mask Register (PHY MII 12h)**

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	Not 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

**Table 13 Interrupt Mask Register (PHY MII 13h)**

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

Bit	Symbol	Access	Default	Description
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 reverved 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
```

在完成以上设置后，需要验证此设置是否能正确地触发中断，方法如下：

- 1 对 YT8522 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。

## 4.7 其它维测寄存器

### 4.7.1 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，默认值为 0x0。

### 4.7.2 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，导致内部工作异常。查看方法如下

- 1 ext\_reg0x61[6]置 1，开启 PLL Lock 检测。默认是 enable，可以不配置。
- 2 读取 ext\_reg0xa0[1:0]，若这 2 位不同时为 0，则 PLL unlock。

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

**Table 14 EXT 61h Register**

Bit	Symbol	Access	Default	Description
6	en_vco_check	RW	0x0	enable or disable VCO_fast or VCO_slow check circuit

**Table 15 EXT A0h Register**

Bit	Symbol	Access	Default	Description
1	PLL_vco_h	RO	0x0	PLL fast indicator
0	PLL_vco_l	RO	0x0	PLL slow indicator