

裕太微电子  
Motor Comm

MAY 6, 2023

# YT8521SH-CA YT8521SC-CA 应用说明

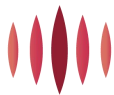
版本 1.5

苏州裕太微电子有限公司

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](http://www.motor-comm.com)



## 版权声明

### Copyright Statement

本文档版权归苏州裕太车通电子科技有限公司（以下简称“裕太车通”）所有，并保留一切权利。未经裕太车通书面许可，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。否则裕太车通将保留追究其法律责任的权利。

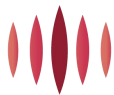
This document is copyright of Suzhou Motorcomm Electronic Technology Co., Ltd. ("Motorcomm"). All rights reserved. No company or individual may copy, disseminate, disclose or otherwise distribute any part of this document to any third party without the written consent of Motorcomm. If any company or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

## 免责声明

### Disclaimer

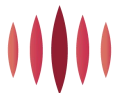
本文档仅提供阶段性信息，所含内容将/可根据产品的实际情况随时更新，恕不另行通知。如因文档使用不当造成直接或间接损失，裕太车通不承担任何责任。

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.



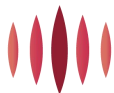
## Revision History

Revision	Release Date	Summary
1.0	2020.06.18	初始化配置：去掉“不支持 100fx”和“不插网线误连接到 10BT”的问题； 加入 Jumbo 功能说明；
1.1	2020.07.03	加入 sleep 功能介绍
1.2	2020.08.17	更新 LDS 和 4 对线百兆功能介绍
1.3	2020.08.20	1. 加入关于 phyaddr0 和 broadcast address 的说明； 2. fast link down 配置说明。 3. 增加 RGMII In-Band Status 的说明。 4. 修改 3.3 UTP MII/LDS MII 地址选择位默认值 5. 修改 5.2 默认高优先级为 UTP
1.4	2022.04.23	修改外接参考时钟的抖动要求
1.5	2023.05.06	CSD (cable status diagnostic ) 命名修改

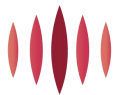


## 目录

<b>1. 简介</b>	<b>5</b>
<b>2. 缩略语</b>	<b>5</b>
<b>3. 寄存器类型及访问方法</b>	<b>5</b>
3.1 寄存器的类型	5
3.2 通过MII寄存器访问其它类型寄存器	7
3.3 寄存器地址空间	8
3.4 读写寄存器的某几位的表示方法	8
3.5 PHYADDR0和BDCST_ADDR的使用说明	9
<b>4. 初始化配置及不支持的功能</b>	<b>10</b>
4.1 LDS在某些情况下建立连接时间过长	10
<b>5. 支持的工作模式</b>	<b>10</b>
5.1 UTP-RGMII, FIBER-RGMII, UTP-SGMII	10
5.2 UTP/FIBER-RGMII	10
5.3 UTP-FIBER AUTO	10
5.4 UTP-FIBER FORCE	11
5.5 SGMII_MAC-RGMII_PHY	11
5.6 SGMII_PHY-RGMII_MAC	12
<b>6. UTP（电口）配置</b>	<b>13</b>
6.1 10/100/1000BT的速度和双工	13
6.2 只保留1000BT能力	14
6.3 POWER DOWN设置	15
6.4 LOOPBACK（回环）模式	15
6.5 SMART DOWNGRADE（自动降速）配置	17
6.6 SLEEP（自动睡眠）配置	18
6.7 包生成器和收发包统计	18
6.8 LDS和4对线百兆长距离	19
6.9 SNR读取（当前连接质量指示）	23
6.10 CSD（线长检测）配置	23
6.11 TEMPLATE（电口指标）配置	24
6.12 FAST LINK DOWN配置	25
<b>7. RGMII配置</b>	<b>26</b>
7.1 通过POS设置RX_DELAY	26
7.2 通过寄存器加长RX_DELAY	26
7.3 TX_DELAY的设置	26
7.4 驱动能力的设置	26
7.5 RGMII支持IN-BAND STATUS	26
<b>8. SERDES（FIBER/SGMII）配置</b>	<b>27</b>
8.1 SGMII的速率和双工	27



8.2	FIBER的速率和双工 .....	28
8.3	SGMII SYNC状态位 .....	29
8.4	SGMII PRBS配置 .....	29
8.5	SGMII输出抖动范围 .....	30
8.6	SGMII输出预加重配置 .....	31
8.7	SGMII允许接收的最大幅值 .....	31
8.8	SGMII输出极性反转配置 .....	31
8.9	LOOPBACK（回环）模式 .....	31
8.10	包生成器和收发包统计 .....	32
9.	LED灯配置 .....	33
10.	INTERRUPT（中断）配置 .....	34
11.	SYNC-E（时钟输出）配置 .....	36
12.	JUMBO FRAME（巨型帧）配置 .....	36
13.	其它维测寄存器 .....	37
13.1	用户自配置寄存器 .....	37
13.2	PLL锁定指示 .....	37
13.3	UTP BIG ERROR COUNTER .....	37
13.4	CLOCK DRIFT .....	39
14.	数据传输通路延时 .....	39
15.	硬件电路设计相关问题 .....	40
15.1	快速上下电的要求 .....	40
15.2	上电复位时序要求 .....	41
15.3	硬复位后延时进行MDIO操作要求 .....	43
15.4	外接参考时钟的抖动要求 .....	43
15.5	外接参考时钟源切换操作要求 .....	44



# 1. 简介

YT8521SH-CA/YT8521SC-CA 是一款单口千兆以太网 PHY。与相对应的 YT8521SH/YT8521SC 相比，主要增加了对 100FX 的支持。本文主要介绍它的一些用法，还包括客户使用过程中经常遇到的问题，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。为方便起见，下文中仍用 YT8521S 代指 YT8521SH-CA/YT8521SC-CA。

## 2. 缩略语

**YT:** 裕太车通，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 YT8521S 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 YT8521S 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 1000/100/10BT 三种速率。

**Fiber:** 光纤，用来与其它 PHY 芯片相连的接口。本文用来代指以太网光口，包括 1000BX 和 100FX 两种速率。

**RGMII:** Reduced Giga Media Independent Interface，是 PHY 与 MAC 之间的并行接口。

**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, 早期以太网 PHY 芯片与 MAC 相连的接口，只适用于百兆/十兆速率，在 YT8521S 中已经不存在。但因为 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。

**SGMII:** Serial giga media independent interface。一种用来与 MAC 相连的接口。

**SDS:** Serdes, 串行解串总线的统称，上述的 SGMII 和 Fiber 都属于 SDS 总线。相关的寄存器也称为 SDS 寄存器。

## 3. 寄存器类型及访问方法

### 3.1 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，以下简称为 mii\_reg），扩展寄存器（extended register，以下简称为 ext\_reg）和 MMD 寄存器（简称为 mmd\_reg）。

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：

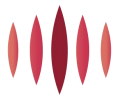


Table 22-12—Management frame format

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

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

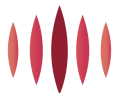
write mii\_reg 0xd 0x7; write mii\_reg 0xe 0x3c; write mii\_reg 0xd 0x4007; write mii\_reg 0xe 0x6

MI1 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

MI1 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.
------	--------------	----	-------	---

## 3.2 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, reg_addr)
```

```
    write_mii_reg(phy_addr, 0x0d, 0x4000+device)
```

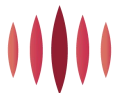
```
    d = read_mii_reg(phy_addr, 0x0e)
```

```
    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`

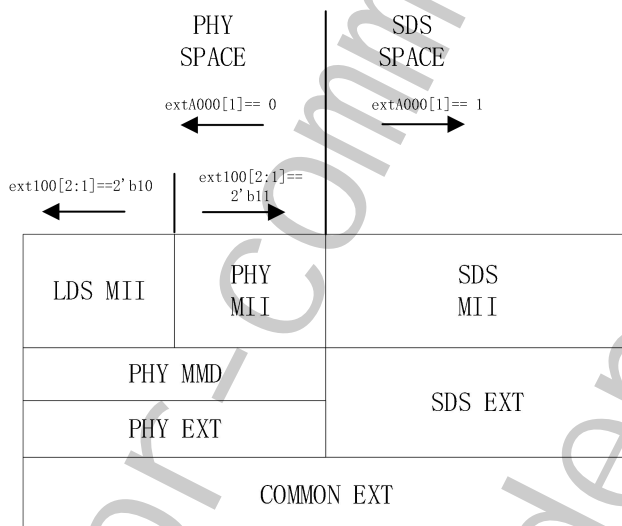




返回值：无

```
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)
```

### 3.3 寄存器地址空间



YT8521S 寄存器可以分为三块大的空间，UTP(1000/100/10BT 和百兆长距离相关的寄存器), SDS(Serdes 相关寄存器，包括 SGMII/Fiber 共用)和 COMMON(通用寄存器)。见上图。其中，

**COMMON 空间寄存器：**都属于 ext\_reg，且其起始地址为 0xa000。一般地用法是，

Write\_ext\_reg(phy\_addr, 0xa000, 0x2): 选择访问 SDS（包括 SGMII/Fiber）寄存器空间

Write\_ext\_reg(phy\_addr, 0xa000, 0x0): 选择访问 UTP 寄存器空间

**UTP 空间寄存器：**分 mii\_reg, ext\_reg(起始地址 0x0~0x2xx，不与 common 寄存器地址冲突), mmd\_reg。

UTP MII 寄存器: Ext\_reg0x100 bit1 为 1，这是默认值，选择 UTP MII 寄存器，IEEE1000/100/10BT 的速度，自协商能力等，判断 UTP 连接的速度与双工等寄存器，都在 UTP mii\_reg 中。

LDS MII 寄存器: Ext\_reg0x100 bit1 手动设为 0，选择跟 LDS 相关的 MII 寄存器。

UTP MMD 寄存器: mmd\_reg，寄存器定义见 802.3 定义和芯片手册

UTP EXT 寄存器: ext\_reg，寄存器定义见 802.3 定义和芯片手册

**SDS 空间寄存器：**分 mii\_reg, ext\_reg(起始地址 0x0~0x2xx，不与 common 寄存器地址冲突)，。主要用来配置 SGMII/Fiber 相关的功能。

SDS MII 寄存器: mii\_reg, SGMII/Fiber 的速度，自协商能力等，SGMII/Fiber 连接的状态寄存器等。

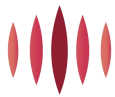
SDS EXT 寄存器: ext\_reg，主要是一些跟 SGMII/Fiber 相关的 debug 寄存器等。

### 3.4 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\_utp/sds\_mii\_reg0x1[15]:1'b0:**

1. 通过写 ext\_reg0xa000 为 0x0/0x2，选择 UTP 或 SDS 寄存器空间。



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

**read\_otp/sds\_mii\_reg0x1[15]:**

1. 通过写 ext\_reg0xa000 为 0x0/0x2, 选择 UTP 或 SDS 寄存器空间。
2. 读 utp/sds 的 mii 寄存器 0x1 的值并获取 bit15, 共 1 比特。

**write\_otp/sds\_ext\_reg0xa[15,14,13]:3'b010:**

1. 通过写 ext\_reg0xa000 为 0x0/0x2, 选择 UTP 或 SDS 寄存器空间。
2. 将 utp/sds 的扩展寄存器 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\_otp/sds\_ext\_reg0xa[15,14,13]:**

1. 通过写 ext\_reg0xa000 为 0x0/0x2, 选择 UTP 或 SDS 寄存器空间。
2. 读 utp/sds 的扩展寄存器 0xa 的值并获取 bit15, 14, 13, 共 3 比特。

## 3.5 phyaddr0 和 bdcst\_addr 的使用说明

YT8521S 有 phyaddr0 和 bdcst\_addr 两种特色功能, 用户可以方便、快捷地访问 YT8521S 的寄存器,。

当 phyaddr0 enable 时, YT8521S 除了响应 phy address = Strapping PHYAD[2:0] 的读/写命令, 还可以响应 phy address = 0x0 的读/写命令。(Strapping PHYAD[2:0]的说明请参考 Datasheet。)

通过 write\_ext\_reg0xa005[6]: 1'b1 来 enable phyaddr0, 通过 write\_ext\_reg0xa005[6]: 1'b0 disable phyaddr0。默认 phyaddr0 是 enable 的。

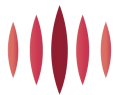
当 bdcst\_addr enable 时, YT8521S 可以响应 phy address = bdcst\_addr 的写命令, 可以达到一次配置 MDIO 总线上多颗 PHY 的目的。其中, bdcst\_addr 可以配置成 0x0~0x1f 中的任意值。

通过 write\_ext\_reg0xa005[5]: 1'b1 来 enable bdcst\_addr, 并 write\_ext\_reg0xa005[4:0]来配置相应的 broadcast address。通过 write\_ext\_reg0xa005[5]: 1'b0 来 disable bdcst\_addr。默认 bdcst\_addr 是 disable 的。

需要注意的是:

- 1、当 bdcst\_addr enable 时, YT8521S 不会响应 phy address = bdcst\_addr 的读命令, 可以响应写命令;
- 2、当 phyaddr0 enable 时, YT8521S 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。

Common EXT A005: mdio cfg and rgmii oob mon				
Bit	Symbol	Access	Default	Description
15:14	Speed_rgmac_ob	RO	0x0	speed information RGMII MAC decodes from the OOB
13	Duplex_rgmac_ob	RO	0x0	duplex information RGMII MAC decodes from the OOB
12	Link_up_rgmac_ob	RO	0x0	linkup information RGMII MAC decodes from the OOB
11	Reserved	RO	0x0	Reserved
10	Bypass_mdio_watchdog	RW	0x0	bypass mdio watch dog
9:8	Reserved	RO	0x0	Reserved
7	En_mdc_la	RW	0x1	enable mdc latch for read data
6	En_phyaddr0	RW	0x1	1: to always respond to MDIO command whose PHYAD field is 0; 0: to only respond to MDIO command whose PHYAD filed equals to PHY address strapping.
5	En_bdcst_addr	RW	0x0	enable broadcast address
4:0	Bdcst_addr	RW	0x0	broadcast address



## 4. 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。

### 4.1 LDS 在某些情况下建立连接时间过长

YT8521S 除了支持 IEEE 的 1000/100/10BT 外，还支持 LRE100-4 (百兆的长距离以太网)，它是在 4 对线上实现超过 400 米 (cat5e) 的百兆可靠传输。要连到这个模式，需要对连的 PHY 都支持此模式。类似于 IEEE 1000/100/10BT 的发送 TLP 实现双方能力协商，支持百兆长距离以太网的双方 PHY 是靠 LDS (long-distance signal) 实现双方连接能力的交互，并根据连接策略，最终连接到一个双方都支持的速率。

芯片内部的问题会导致本地 LDS 打开，对方 LDS 关闭并 force 10BT 情况下，建立连接的时间较长 (几秒钟)。默认情况下，YT8521S 的 LDS 功能是不开的，此问题不存在。

为避免此问题，在初始化时 (或硬复位后) 需软件对 YT8521S 寄存器配置如下：

write ext\_reg 0xa000 → 0x0 (选择 utp 寄存器页)

write ext\_reg 0x9e → 0x33 (修改 utp 扩展寄存器 0x9e)

## 5. 支持的工作模式

YT8521S 的 RGMII, SGMII/Fiber, RGMII 可以组合为多种模式：

UTP-RGMII

FIBER-RGMII

UTP/FIBER-RGMII

UTP-SGMII

UTP-FIBER\_AUTO

UTP-FIBER\_FORCE

SGMII\_MAC-RGMII\_PHY

SGMII\_PHY-RGMII\_MAC

上述各模式的选择是通过 power-on-strapping 实现的，具体详见 datasheet 或参考电路。

### 5.1 UTP-RGMII, FIBER-RGMII, UTP-SGMII

这几种模式使用上没有特别说明的地方。UTP, FIBER, RGMII, SGMII 每部分功能独立，相关配置详见后面的各部分配置说明。

### 5.2 UTP/FIBER-RGMII

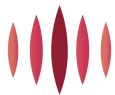
应用场景，详见 Datasheet。几点说明如下：

1. Fiber/UTP 默认 UTP 优先，其优先级可以通过 ext\_reg0xa006[8]来设置：1'b0: UTP 优先；1'b1: Fiber 优先。然后 write ext\_reg0xa001[15]: 1'b0，使配置生效。

2. 可以通过读取 UTP mii\_reg0x11[10]和 SDS mii\_reg0x11[10]来查询当前是 UTP 还是 Fiber 连接。

### 5.3 UTP-FIBER AUTO

应用场景，详见 Datasheet。



UTP\_TO\_FIBER\_AUTO 模式的“AUTO”是指：

UTP 的 10/00/1000 能力是随跟 Fiber 状态而自动变化，且 Fiber 不 link，UTP 是不会 link 的。具体如下：

- A. Fiber link down: UTP 内部自动进入 power down 模式，即使和 Link partner 对接，也不会 link 上；
- B. Fiber link 在 1000BX: UTP 自动配置 mii reg 4 = 0x1c01, reg 9 = 0x200，表示只有千兆的能力，并与对端 UTP 协商；
- C. Fiber link 在 100FX: UTP 自动配置 mii reg 4 = 0x1d81, reg 9 = 0x0，表示只有 100M full/half 的能力，并与对端 UTP 协商；

UTP 和 Fiber 的速度和双工的配置和状态，详细介绍请分别参见 6.1 和 8.2。

在 UTP\_TO\_FIBER\_AUTO mode 下，Fiber auto sensing 功能默认是打开的，即 Fiber 会自动检测插入的是百兆还是千兆光模块来适配到相应的连接速度。也可以关闭 auto sensing 功能，再强制 Fiber 为 1000BX 或者 100FX 速度：

- A. sds ext 0xA5 的 bit15 先清 0（关闭 auto sensing）；
- B. 设置 ext reg 0xA006 的 Bit0（0: 100FX；1: 1000BX）；
- C. 将 0xA001.Bit15 清 0；

## 5.4 UTP-FIBER FORCE

应用场景，详见 Datasheet。

在 UTP\_TO\_FIBER\_FORCE mode 下，Fiber auto sensing 功能默认是关闭的，fiber 默认为 1000BX 能力与对端连接。如果需要改为 100FX，设置如下：

- A. write\_ext\_reg0xA006[0]: 1'b0 或 1'b1 # (0: 100FX; 1: 1000BX);
- B. write\_ext\_reg0xA001[15]: 1'b0 #PHY 整个软复位；

根据 Fiber 的当前的连接状态，UTP 也要相应地手动配置能力，使 UTP 与 Fiber 状态相匹配：

- A. Fiber link down: UTP 也可配置为 power down 模式，即使和 Link partner 对接，也不会 link 上；
- B. Fiber link 在 1000BX: UTP 可配置 utp\_mii\_reg0x4 = 0x1c01, utp\_mii\_reg0x9 = 0x200，表示只有千兆的能力，并与对端 UTP 协商；
- C. Fiber link 在 100FX: UTP 可配置 utp\_mii\_reg0x4 = 0x1d81, utp\_mii\_reg0x9 = 0x0，表示只有 100M full/half 的能力，并与对端 UTP 协商；

## 5.5 SGMII\_MAC-RGMII\_PHY

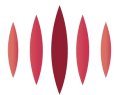
应用场景，详见 Datasheet。

SGMII\_MAC 的 link 状态、速率、双工模式参考 section 8.1。

RGMII\_PHY 的状态是跟随 SGMII\_MAC 的，可以通过 ext\_reg0xa004 的 bit[12] (link status), bit[15:14] (speed), bit[13] (duplex) 查询。

### Common EXT A004: rgmii cfg2

Bit	Symbol	Access	Default	Description
15:14	Speed_rgphy	RO	0x0	RGMII's speed information when it works as RGMII PHY. It's also the source of RGMII OOB.
13	Duplex_rgphy	RO	0x0	RGMII's duplex information when it works as RGMII PHY. It's also the source of RGMII OOB.
12	Link_up_rgphy	RO	0x0	RGMII's linkup information when it works as RGMII PHY. It's also the source of RGMII OOB.



11:10	Pause_rgphy	RO	0x0	RGMII's pause information when it works as RGMII PHY.
9	Eee_cap_rgphy	RO	0x0	RGMII's EEE capability information when it works as RGMII PHY.
8	Eee_clkstp_cap_rgphy	RO	0x0	RGMII's EEE clock stopable capability information when it works as RGMII PHY.
7:6	Speed_rgmact	RW	0x0	RGMII's speed configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.
5	Duplex_rgmact	RW	0x0	RGMII's duplex configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.
4	Link_up_rgmact	RW	0x0	RGMII's linkup configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.
3:2	Pause_rgmact	RW	0x0	RGMII's pause configuration when it works as RGMII MAC.
1	Eee_cap_rgmact	RW	0x0	RGMII's EEE capability configuration when it works as RGMII MAC.
0	Eee_clkstp_cap_rgmact	RW	0x0	RGMII's EEE clock stopable capability configuration when it works as RGMII MAC.

## 5.6 SGMII\_PHY-RGMII\_MAC

应用场景，详见 Datasheet。

当 YT8521S 通过 RGMII 和其他家 PHY 对接时（如下图所示），需要注意：

如果其他家 PHY 的 RGMII 支持 In-Band Status 传递 link、speed、duplex 信息，YT8521S 的 RGMII\_MAC 可以通过 In-Band Status 获取信息，并体现在 ext\_reg0xa005 的 bit[12] (link status), bit[15:14](speed), bit[13] (duplex)。YT8521S 的 SGMII\_PHY 会获取此信息，并与 CPU 的 SGMII\_MAC 协商。

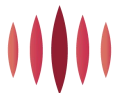
如果其他家 PHY 的 RGMII 不支持 In-Band Status。可以通过 write\_ext\_reg0xa003[15]:1'b1，设置 YT8521S 的 RGMII 为强制模式。然后再配置 ext\_reg0xa004 的 bit[4] (link status), bit[7:6](speed), bit[5] (duplex)。最后，将 ext\_reg0xa001 的 bit[15]清零，PHY 内部会整体软复位，使上述配置生效。同样地，YT8521S 的 SGMII\_PHY 会获取此信息，并与 CPU 的 SGMII\_MAC 协商。



Common EXT A004: rgmii cfg2

Bit	Symbol	Access	Default	Description
15:14	Speed_rgphy	RO	0x0	RGMII's speed information when it works as RGMII PHY. It's also the source of RGMII OOB.
13	Duplex_rgphy	RO	0x0	RGMII's duplex information when it works as RGMII PHY. It's also the source of RGMII OOB.
12	Link_up_rgphy	RO	0x0	RGMII's linkup information when it works as RGMII PHY. It's also the source of RGMII OOB.
11:10	Pause_rgphy	RO	0x0	RGMII's pause information when it works as RGMII PHY.
9	Eee_cap_rgphy	RO	0x0	RGMII's EEE capability information when it works as RGMII PHY.
8	Eee_clkstp_cap_rgphy	RO	0x0	RGMII's EEE clock stopable capability information when it works as RGMII PHY.
7:6	Speed_rgmact	RW	0x0	RGMII's speed configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.
5	Duplex_rgmact	RW	0x0	RGMII's duplex configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.





4	Link_up_rgmac	RW	0x0	RGMII's linkup configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.
3:2	Pause_rgmac	RW	0x0	RGMII's pause configuration when it works as RGMII MAC.
1	Eee_cap_rgmac	RW	0x0	RGMII's EEE capability configuration when it works as RGMII MAC.
0	Eee_clkstp_cap_mac	RW	0x0	RGMII's EEE clock stopable capability configuration when it works as RGMII MAC.

## 6. UTP（电口）配置

### 6.1 10/100/1000BT 的速度和双工

在硬件电路板上通过 POS 管脚设置 YT8521S 到不同的工作模式，比如 UTP-RGMII, UTP-Fiber 等。

以常用的 UTP-SGMII/UTP-RGMII 为例（此时，默认的 ext\_reg0xa000 值为 0，且 utp\_ext\_reg0x100 值为 0，即选择 PHY mii\_reg）：

上电硬复位后，可以读取 phy\_mii\_reg0x2,0x3 获取 PHY ID。默认 YT8521S 不需做任何配置，YT8521S 的自协商能力和 1000/100/10BT 能力都是打开的。

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\_otp\_mii\_reg0x0[12]: 1'b1 #自协商打开，此为默认打开状态

write\_otp\_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\_otp\_mii\_reg0x9[9] #相应 bit 置 1 来打开对应的本地自协商时的 1000BT 的速度双工能力

write\_otp\_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\_otp\_mii\_reg0x0[12]: 1'b0 #自协商能力关闭

write\_otp\_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\_otp\_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。

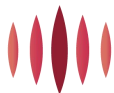
其中的几个 bit 含义如下（或参照芯片手册）：

Bit 15-Bit14 : speed mode, 11---系统保留；10---1000M；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。



## 6.2 只保留 1000BT 能力

YT8521S 电口不支持 Force 配置千兆，可通过配置自协商能力，来关闭百兆、十兆，只保留千兆能力。相关寄存器是 UTP MII 的 0x04、0x09 寄存器。

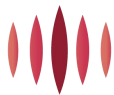
上电复位后，进行如下寄存器配置初始化动作：

write mii\_reg0x04: 0x1C01   #将 Bit8、7、6、5 位的值由默认的 1 改为 0，分别关闭了 100M\_Full、100M\_Half、10M\_Full、10M\_Half 的自协商能力。其余位的值不做处理。若其他位的值在其他功能中有改变，请按改变后的赋值。

Write\_mii\_reg0x00[15]: 1'b1   #置位 mii 寄存器 0x00 的 Bit15，进行软复位，使得上述寄存器设置生效。软复位后，YT8521S 会重新启动 link up 过程。(软复位不会清寄存器值，只会使其生效)

Phy MII 04h: Auto-Negotiation advertisement				
Bit	Symbol	Access	Default	Description
15	NEXT_Page	RW	0x0	If 1000BASE-T is advertised, the required next pages are automatically transmitted. This bit must be set to 0 if no additional next page is needed. 1 = Advertise 0 = Not advertised
14	Ack	RO	0x0	Always 0.
13	Remote_Fault	RW	0x0	1 = Set Remote Fault bit 0 = Do not set Remote Fault bit
12	Extended_NEXT_Page	RW	0x1	1 = Local device supports transmission of extended next pages 0 = Local device does not support transmission of extended next pages.
11	Asymmetric_Pause	RW	0x1	1 = Asymmetric Pause 0 = No asymmetric Pause
10	Pause	RW	0x1	1 = MAC PAUSE implemented 0 = MAC PAUSE not implemented
9	100BASE-T4	RO	0x0	1 = Able to perform 100BASE-T4 0 = Not able to perform 100BASE-T4 Always 0
8	100BASE-TX_Full_Duplex	RW	0x1	1 = Advertise 0 = Not advertised Needs Soft Reset action to take effect
7	100BASE-TX_Half_Duplex	RW	0x1	1 = Advertise 0 = Not advertised Needs Soft Reset action to take effect
6	10BASE-Tc_Full_Duplex	RW	0x1	1 = Advertise 0 = Not advertised Needs SoftReset action to take effect
5	10BASE-Tc_Half_Duplex	RW	0x1	1 = Advertise 0 = Not advertised Needs SoftReset action to take effect
4:0	Selector_Field	RW	0x1	Selector Field mode. 00001 = IEEE 802.3

Phy MII 09h: MASTER-SLAVE control register



Bit	Symbol	Access	Default	Description
15:13	Test mode	RW	0x0	The TX_TCLK signals from the RX_CLK pin is for jitter testing in test modes 2 and 3. Needs soft reset to take effect 000 = Normal Mode 001 = Test Mode 1 - Transmit Waveform Test 010 = Test Mode 2 - Transmit Jitter Test (MASTER mode) 011 = Test Mode 3 - Transmit Jitter Test (SLAVE mode) 100 = Test Mode 4 - Transmit Distortion Test 110, 111 = Reserved
12	Master/Slave Manual configuration Enable	RW	0x0	1 = Manual MASTER/SLAVE configuration 0 = Automatic MASTER/SLAVE configuration. Needs soft reset to take effect
11	Master/Slave configuration	RW	0x0	This bit is ignored if bit[12] is 0. 1 = Manual configuration as MASTER 0 = Manual configuration as SLAVE. Needs soft reset to take effect
10	Port Type	RW	0x0	This bit is ignored if bit[12] is 1. 1 = Prefer multi-port device (MASTER) 0 = Prefer single port device (SLAVE) Needs soft reset to take effect
9	1000BASE-T Full	RW	0x1	1 = Advertise 0 = Not advertised Needs soft reset to take effect
8	1000BASE-T Half-	RW	0x0	1 = Advertise 0 = Not advertised (default) Needs soft reset to take effect
7:0	Reserved	RW	0x0	Write as 0, ignore on read.

## 6.3 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。

## 6.4 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：

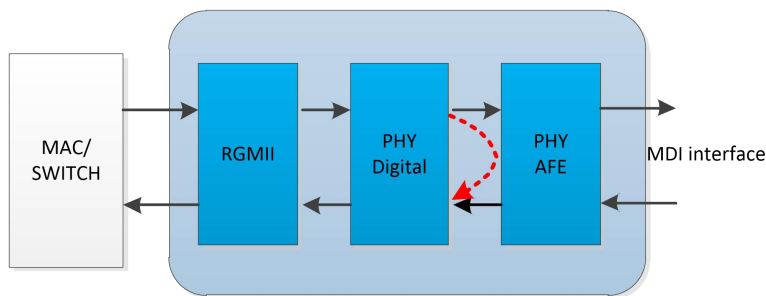
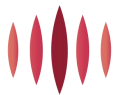
10M: utp phy mii register 0x0 = 0x4100

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

1000M: utp phy mii register 0x0 = 0x4140

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。





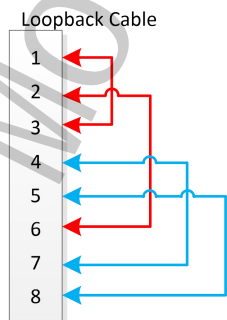
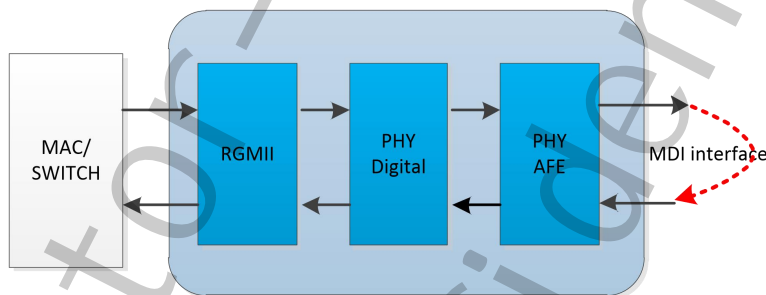
2、在 External Loopback 模式下，AFE 发送的信号由回环线直接回到 AFE 的接收电路。MAC 通过比对 RGMII 发送数据和接收数据，从而确保 RGMII 接口，PHY 数字和模拟功能的正确性。回环线将网线的 pair1&2 与 pair3&6 连接，pair4&5 与 pair7&8 连接，如下图所示。配置方法如下：

10M: 先 phy ext 0x27 bit15 位清 0(关闭 sleep 模式); utp phy ext register 0xa.4 = 1, mii register 0x0 = 0x8100;

100M: 先 phy ext 0x27 bit15 位清 0(关闭 sleep 模式); utp phy ext register 0xa.4 = 1, mii register 0x0 = 0xa100

1000M:先 phy ext 0x27 bit15 位清 0(关闭 sleep 模式); utp phy ext register 0xa.4 = 1, mii register 0x0 =0x8140

注意：上述配置的最后一步 mii register x0，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，RGMII RX 接收的数据被直接注入 RGMII 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: common register 0xa006.5 = 1

utp phy mii register 0x9=0x0,

utp phy mii register 0x4 = 0xc41,

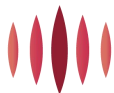
utp phy mii register 0x0 = 0x9140,

100M: common register 0xa006.5 = 1

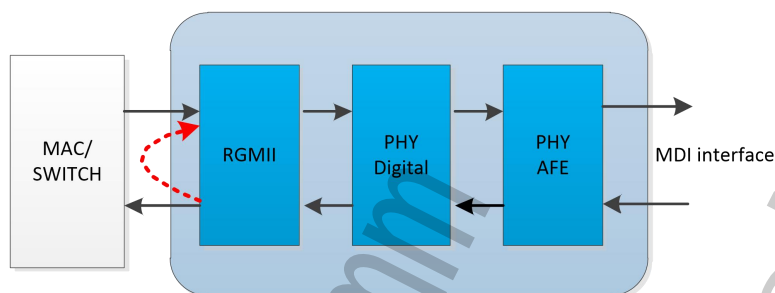
utp phy mii register 0x9=0x0,

utp phy mii register 0x4 = 0xd01,

utp phy mii register 0x0 = 0x9140,



1000M: common register 0xa006.5 = 1  
utp phy mii register 0x9=0x200,  
utp phy mii register 0x4 = 0xc01,  
utp phy mii register 0x0 = 0x9140



## 6.5 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 1000BT 的能力发起自协商，但网线超过最大连接距离（比如到 150 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 1000BT，此时具备自动降速功能的一方会发起 100BT 的自协商，尝试 100BT 的连接。当然若 100BT 经过 5 次尝试，仍不能连接，PHY 会降到 10BT 再尝试。

注：

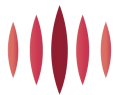
1. 如果对端口的快速插拔，导致短时间内的不断 link up/down，也会触发 smart downgrade 机制。
2. 此功能只发生在连接建立时间内，若连接并维持住，PHY 不会根据丢包等自动降速。

以上是 YT8521S 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。



## 6.6 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\_utp\_ext\_reg0x27[15]: 1'b0。

Phy EXT 27h: 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.

## 6.7 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，以下寄存器地址无特殊说明均指 utp phy 寄存器空间，：

1. 打开 packet checker: 写 ext reg 0xA0.[15:14]=2'b10
2. 关闭 packet checker: 写 ext reg 0xA0.[15:14]=2'b01
3. 查看 checker 统计结果:

Ext reg 0xA3~0xA8, 代表从 MDI 收到的正确包的数量，寄存器为 read clear。

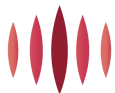
Ext reg 0xA9~0xAC, 代表从 MDI 收到的包有 crc error, 寄存器为 read clear。

Ext reg 0xAD~0xB2, 代表从 RGMII 收到的正确包的数量，寄存器为 read clear。

Ext reg 0XB3~0XB6, 代表从 RGMII 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 0xA1.[15:0]
2. 设置发送包的个数: 写 ext reg 0xA2.[15:0], 值为 0 表示一直发包。
3. 设置 IPG 长度: 写 ext reg A0.[7:4], 不用时可以不设置。
4. 还可以设置 packet DA/SA 地址，不用时可以不设置。  
使能该功能，写 ext reg 0x38.[12] = 1'b1, 关闭该功能，写 ext reg 0x38.[12] = 1'b0  
设置 DA 的低 8 位，写 ext reg 0x3A.[15:8], 则 DA 地址为 00 00 00 00 00 value  
设置 SA 的低 8 位，写 ext reg 0x3A.[7:0], 则 SA 地址为 00 00 00 00 00 value
5. 打开 packet generator:  
写 ext reg 0xA0[15:13] = 3'b100, 配置 packet checker 和 generator。  
写 ext reg 0xA0[12] = 1'b1, 打开 packet generator 开始发包。
6. Polling ext reg 0xA0[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 0xA0[12] = 1'b0, 表示停止发包。
7. 关闭 packet generator:



写 ext reg 0xA0[14] = 1'b1, 关闭 packet generator clock

写 ext reg 0xA0[15, 13, 12] = [0,1,0], 关闭 generator 和 checker

## 6.8 LDS 和 4 对线百兆长距离

YT8521S 支持在 4 对 UTP 线上实现长距离 (>400 米 cat5e) 百兆以太网的信号传输。当然, 前提是对方也必须支持这种模式 (通常也是 YT 系列型号, 目前包括 YT8511, YT8521S)。除去上面提到的区别外, 与传统百兆以太网 (100BT) 相比, 4 对线长距离以太网 (暂命名为 LRE100-4) 还有以下特点:

通过 LDS (Link Discovery Signaling) 协商的机制实现双方支持能力的交互 (10/100/100BT, LRE100-4)。

在 LDS 协商过程中检测到连接的线长, 根据双方能力和线长, 决定连接到哪种速率, 在 LDS enable, 并且所有能力都打开的情况下, 最终速率与线长的关系为:

线长	速率
< 200m	10/100/1000BT 自协商
>= 200m	LRE100-4

注:

1. 整个过程全部由 PHY 自动实现, MAC 不参与。MAC 通过 polling PHY 寄存器得到当前的状态。
2. 默认情况下, PHY 的所有能力 (1000BT, 100BT, 10BT, LRE100-4) 都打开, 但 LDS 不开, 所以跟传统的 GE 一样, 不会进 LRE100-4 模式。
3. 在 LDS 打开的情况下, 若对端是传统 10/100/1000 以太网, DUT 也能够与其对连到传统以太网模式。
4. 在长线条件下, 线长检测精度通常 <5%。
5. LDS/LRE100-4 模式下, 支持直连线与交叉线, 支持 MDI 自动极性翻转。
6. 与同样打开 LDS/LRE100-4 的 PHY 相连, 线长在 200 米以下按 IEEE 10/100/1000 机制连接, 超过 200 米, 自动连到 LRE100-4。
7. 与不支持 LDS/LRE100-4 的 PHY 相连, 自动检测对方发的 AN/Force 信号, 再依照 IEEE 10/100/1000BT 连接到全双工或半双工模式。

在 YT8521S 与 YT8521S 相连情况下, 若要进入 LRE100-4 模式, 只要:

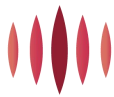
1. 双方 LDS 功能打开情况下, 连线超过 200m (极限线长超过 400m)
2. 读 LDS 或 PHY MII\_reg0x1[2], 判断 DUT 已连接。
3. 读 DUT LDS MII\_reg0xb[5], 判断当前连到 LRE100-4, 则配置 MAC 到百兆
4. 正常通信

相关寄存器配置:

LDS 的相关寄存器放在 LDS MII 寄存器中, 与 PHY MII 寄存器属于不同的地址空间, 其通过 UTP ext reg 0x100 来选择, 其含义为:

Phy EXT 100h: PHY/LDS sel				
Bit	Symbol	Access	Default	Description
15:3	Reserved	RO	0x0	reserved
2	Acc_ctrl_ovrd_en	RW	0x1	Access control override enable; 1'b1: override LRE register access; 1'b0: Normal operation
1	Acc_ctrl_ovrd_val	RW	0x1	1'b1: Access IEEE registers; 1'b0: Access LRE registers
0	Acc_ctrl_val	RO	0x1	1'b1: IEEE register is active; 1'b0: LRE register is active

LDS 打开/关闭的寄存器:



#### Lds mii 00h: LRE control

Bit	Symbol	Access	Default	Description
15	Reset	RW SC	0x0	PHY Software Reset. Writing 1 to this bit causes immediate PHY reset. Once the operation is done, this bit is cleared automatically.; 1'b0: Normal operation; 1'b1: PHY reset
14	Loopback	RW	0x0	Loopback control; 1'b0: disable loopback; 1'b1: enable loopback
13	Restart_LDS	RW SC	0x0	1'b1: restart LDS process
12	LDS_Enable	RW	0x0	1'b1: LDS enabled; 1'b0: LDS disabled
11	Power_down	RW	0x0	1 = Power down; 0 = Normal operation; When the port is switched from power down to; normal operation, software reset and Auto-; Negotiation are performed even bit[15] RESET and bit[9] RESTART_AUTO_NEGOTIATION are not set by the user.
10	Isolate	RW	0x0	Isolate phy from MII/GMII/RGMII: PHY will not respond to RGMII TXD/TX_CTL, and present high impedance on RXD/RX_CTL.; 1'b0: Normal mode; 1'b1: Isolate mode
9:6	Speed_selection	RW	0x0	4'b0000: 10Mbps; 4'b1000: 100Mbps; Others: reserved
5:4	Pair_selection	RW	0x0	2'b00: 1 pair connection; 2'b01: 2 pair connections; 2'b10: 4 pair connections; 2'b11: reserved
3	M/S_selection	RW	0x0	1'b1: manually force local device to master, when reg0.12 = 0; 1'b0: manually force local device to slave, when reg0.12 = 0
2	Force auto negotiation	RW	0x0	1'b1: manually force local device to auto negotiation state, when reg0.12 = 0
1:0	Reserved	RO	0x0	Reserved. Write as 0, ignore on read

线长检测的长度:

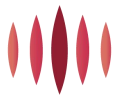
#### Lds mii 0Ah: LDS expansion

Bit	Symbol	Access	Default	Description
15	Downgrade_ability	RO	0x0	1'b1: LDS speed downgrade
14	Master/Slave	RO	0x0	1 = Local PHY configuration resolved to Master; 0 = Local PHY configuration resolved to Slave
13:12	Connections_pairs	RO	0x0	Number of pairs; 2'b00: 1 pair; 2'b01: 2 pairs; 2'b10: 4 pairs; 2'b11: reserved
11:0	Estimated_cable_length	RO	0x0	

判断 DUT 当前是否已连接:

#### Lds mii 01h: LRE status

Bit	Symbol	Access	Default	Description
15:14	Reserved	RO	0x0	Ignore on read
13	100Mbps_1-pair capable	RO	0x0	1'b1: 100Mbps 1-pair capable; 1'b0: Not 100Mbps 1-pair capable
12	100Mbps_4-pair capable	RO	0x1	1'b1: 100Mbps 4-pair capable; 1'b0: Not 100Mbps 4-pair capable
11	100Mbps_2-pair capable	RO	0x0	1'b1: 100Mbps 2-pair capable; 1'b0: Not 100Mbps 2-pair capable
10	10Mbps_2-pair capable	RO	0x0	1'b1: 10Mbps 2-pair capable; 1'b0: Not 10Mbps 2-pair capable
9	10Mbps_1-pair capable	RO	0x0	1'b1: 10Mbps 1-pair capable; 1'b0: Not 10Mbps 1-pair capable
8	Extended_Status	RO	0x1	Whether support EXTENDED status register in 0Fh; 0: Not supported; 1: Supported



7	Reserved	RO	0x1	
6	Mf_Preamble_Suppression	RO	0x1	1'b0: PHY will not accept management frames with preamble suppressed; 1'b1: PHY will accept management frames with preamble suppressed
5	LDS_Complete	RO	0x0	1'b1: LDS auto-negotiation complete; 1'b0: LDS auto-negotiation not complete
4	Support_IEEE_802.3_PHY	RO	0x1	1'b1: Support IEEE 802.3 PHY operation; 1'b0: Not Support IEEE 802.3 PHY operation
3	LDS_Ability	RO	0x1	1'b1: LDS auto-negotiation capable; 1'b0: Not LDS auto-negotiation capable
2	Link_Status	RO	0x0	Link status; 1'b0: Link is down; 1'b1: Link is up
1	Jabber_Detect	RO LH	0x0	10Baset jabber detected; 1'b0: no jabber condition detected; 1'b1: Jabber condition detected
0	Extended_Capability	RO LH	0x1	To indicate whether support EXTended registers, to access from address register 1Eh and data register 1Fh; 1'b0: Not supported; 1'b1: Supported

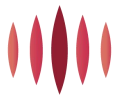
判断当前连接是 LRE100-4 还是 10/100/1000BT:

Lds mii 0Bh: LDS Results				
Bit	Symbol	Access	Default	Description
15:6	Reserved	RO	0x0	
5	4-pair_100M	RO	0x0	1'b1: local PHY configuration resolved to 4-pair 100M
4	Auto_negotiation	RO	0x0	1'b1: local PHY configuration resolved to AN
3	1-pair_100M	RO	0x0	1'b1: local PHY configuration resolved to 1-pair 100M
2	1-pair_10M	RO	0x0	1'b1: local PHY configuration resolved to 1-pair 10M
1	2-pair_100M	RO	0x0	1'b1: local PHY configuration resolved to BR 100M
0	2-pair_10M	RO	0x0	1'b1: local PHY configuration resolved to BR 10M

判断当前连接是 10BT/100BT/1000BT:

Phy MII 11h: PHY specific status register				
Bit	Symbol	Access	Default	Description
15:14	Speed_mode	RO	0x0	These status bits are valid only when bit11 is 1. Bit11 is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. 11 = Reserved 10 = 1000 Mbps 01 = 100 Mbps 00 = 10 Mbps
13	Duplex	RO	0x0	This status bit is valid only when bit11 is 1. Bit11 is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. 1 = Full-duplex 0 = Half-duplex
12	Page Received real-time	RO	0x0	1 = Page received 0 = Page not received
11	Speed and Duplex Resolved	RO	0x0	When Auto-Negotiation is disabled, this bit is set to 1 for force speed mode. 1 = Resolved 0 = Not resolved





10	Link status real-time	RO	0x0	1 = Link up 0 = Link down
9:7	Reserved	RO	0x0	Reserved
6	MDI Crossover Status	RO	0x0	This status bit is valid only when bit11 is 1. Bit11 is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. The bit value depends on register 0x10 "PHY specific function control register" bits6~bit5 configurations. Register 0x10 configurations take effect after software reset. 1 = MDIX 0 = MDI
5	Wirespeed downgrade	RO	0x0	1 = Downgrade 0 = No Downgrade
4	Reserved	RO	0x0	Reserved
3	Transmit Pause	RO	0x0	This status bit is valid only when bit11 is 1. Bit11 is set when Auto-Negotiation is completed. This bit indicates MAC pause resolution. This bit is for information purposes only and is not used by the device. When in force mode, this bit is set to be 0. 1 = Transmit pause enabled 0 = Transmit pause disabled
2	Receive Pause	RO	0x0	This status bit is valid only when bit[11] is 1. Bit[11] is set when Auto-Negotiation is completed. This bit indicates MAC pause resolution. This bit is for information purposes only and is not used by the device. When in force mode, this bit is set to be 0. 1 = Receive pause enabled 0 = Receive pause disabled
1	Polarity Real Time	RO	0x0	1 = Reverted polarity 0 = Normal polarity
0	Jabber Real Time	RO	0x0	1 = Jabber 0 = No jabber

详细的寄存器操作包括初始配置和判断当前连接速率，其中初始化配置：

1. Write ext\_reg0xa000: 0x0 #选择 PHY&LDS 地址空间
2. Write ext\_reg0x100: 0x4 #选择 LDS 地址空间
3. Write mii\_reg0x0: bit12 置 1 #打开 LDS 能力
4. Write mii\_reg0x0: bit15 置 1 #软复位

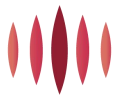
判断当前连接速率：

1. Write ext\_reg0xa000: 0x0 #选择 PHY&LDS 地址空间
2. Write ext\_reg0x100: 0x4 #选择 LDS 地址空间
3. polling mii\_reg0x1[2]变为 1 #表示已连接

Read mii\_reg0xb: bit5 若为 1: 则为 LRE100-4。将 MAC 配置为百兆全双工，结束。

bit5 若为 0，且 bit4 为 1，则连到 10/100/1000BT，然后：

- a) write ext\_reg0x100: 0x6 #选择 PHY&IEEE 地址空间
- b) read mii\_reg0x11: bit 15, 14, 13 #判断速度双工
- c) 配置 MAC 到相同速率，结束



## 6.9 SNR 读取（当前连接质量指示）

YT8521S 的 phy ext 扩展寄存器 0x5A, 0x5B, 0x5C, 0x5D 内的 bit[14:0] 值分别表征了 4 对 MDI 线的 SNR 读取步骤如下：

步骤一，将 phy ext 扩展寄存器 0x59 的 bit15 置 1

步骤二，将 phy ext 扩展寄存器 0xf8 的 bit10 清 0

步骤三，分别读取 phy ext 扩展寄存器 0x5A, 0x5B, 0x5C, 0x5D 的 bit[14:0]

将其记为 MSE(i), i=0,1,2,3

将每个值读至少 100 次，并取平值，得到每对 MDI 线上稳定 MSSE

步骤四，将上述稳定 MSE 值，按照如下公式计算 SNR：

千兆： $10 \cdot \log_{10}(29696/mse)$

百兆： $10 \cdot \log_{10}(32768/mse)$

要达到稳定连接，SNR 值应大于 24.73dB (千兆，对应 MSE 约为 100)；19.13dB (百兆，对应 MSE 约 400)；

附千兆情况下几个 MSE 与 SNR 的对应表：

MSE	SNR
118	24
187	22
297	20
374	19
470	18
592	17
746	16

## 6.10 CSD（线长检测）配置

YT8521S 进入含 copper 的一种模式，然后进行 CSD 测试：

步骤一，把 common ext reg0xa000 写 0x0；#访问 phy 地址空间

步骤二，把 phy ext reg0x98 写为 0xb0a6

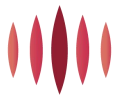
步骤三，把 phy ext reg0x27 的 bit15 清 0；#关掉 sleep 模式

步骤四，把 phy ext reg0x80 的 bit0 置 1；#进入 CSD 测试

步骤五，当 phy ext reg0x84 的 bit15 变为 0，CSD 测试结束。通过读 phy ext reg0x84 的 bit7~0，判断 pair 内 open 和 short 状态；

UTP EXT 84h: CSD_mon0				
7:6	Self_st_3	RO	2'b0	Intra pair status of channel 3. 00: normal, pair impedance matches. 01: error happened during last CSD test, the error may be RX channel is busy, no near-end echo was detected, or the location of far-end echo was not farther than that of near-end echo and the gap between near-end echo and far-end echo is not small; 10: pair is short; 11: pair is open.
5:4	Self_st_2	RO	2'b0	Intra pair status of channel 2. See bit7~6 for detail.
3:2	Self_st_1	RO	2'b0	Intra pair status of channel 1. See bit7~6 for detail.





1:0	Self_st_0	RO	2'b0	Intra pair status of channel 0. See bit7~6 for detail.
-----	-----------	----	------	---

步骤六，通过读取 phy ext reg0x87~8a 的值可以得到 pair0~3 出问题的点，单位为 cm

UTP EXT 87h: CSD_mon3				
Bit	Symbol	Access	default	Description
15:00	Self_dmg_loc_0	RO	0x0	The intra pair damage location of channel 0. In unit cm.
UTP EXT 88h: CSD_mon4				
Bit	Symbol	Access	default	Description
15:00	Self_dmg_loc_1	RO	0x0	The intra pair damage location of channel 1. In unit cm.
UTP EXT 89h: CSD_mon5				
Bit	Symbol	Access	default	Description
15:00	Self_dmg_loc_2	RO	0x0	The intra pair damage location of channel 2. In unit cm.
UTP EXT 8Ah: CSD_mon6				
Bit	Symbol	Access	default	Description
15:00	Self_dmg_loc_3	RO	0x0	The intra pair damage location of channel 3. In unit cm.

## 6.11 Template（电口指标）配置

YT8521S 的 Template 测试指令如下：

初始化：

```
Write_comm_ext_reg0xa000: 0x0 #选择 UTP 地址空间
write_otp_ext_reg0x41 to 0x4
write_otp_ext_reg0x42 to 0x8
write_otp_ext_reg0x43 to 0xc
write_otp_ext_reg0x44 to 0x10
write_otp_ext_reg0x45 to 0x10
write_otp_ext_reg0x46 to 0x10
write_otp_ext_reg0x47 to 0x10
write_otp_ext_reg0x4a to 0x1180
```

1000BT:

```
Write_comm_ext_reg0xa000: 0x0 #选择 UTP 地址空间
write_otp_ext_reg0x27: 0x2026
write_otp_mii_reg0x10: 0x2
write_otp_mii_reg0x0: 0x8140
```

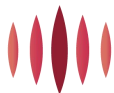
Test Mode 1, Transmit waveform test

```
write_otp_mii_reg0x9: 0x2200
write_otp_mii_reg0x0: 0x8140
```

Test Mode 2, Transmit Jitter test (master mode)

```
write_otp_mii_reg0x9: 0x5a00
write_otp_mii_reg0x0: 0x8140
```

Test Mode 3, Transmit Jitter test (slave mode)



```
write_otp_mii_reg0x9: 0x7200
write_otp_mii_reg0x0: 0x8140
Test Mode 4, Transmit distortion test
write_otp_mii_reg0x9: 0x8200
write_otp_mii_reg0x0: 0x8140
```

#### 100BT:

```
Write_comm_ext_reg0xa000: 0x0 #选择 UTP 地址空间
Write_otp_ext_reg0x27: 0x2026
Write_otp_mii_reg_0x10: 0x2
Write_otp_mii_reg_0x0: 0xa100
```

#### 10BTe:

```
Write_comm_ext_reg0xa000: 0x0 #选择 UTP 地址空间
Write_otp_mii_reg0x0: 0x8100
```

packet with all ones, 10MHz sine wave, for harmonic test

```
Write_otp_ext_reg0xa: 0x209
```

pseudo random, for TP\_idle/Jitter/Different voltage test

```
Write_otp_ext_reg0xa: 0x20A
```

normal link pulse only

```
Write_otp_ext_reg0xa: 0x20B
```

5MHz sine wave

```
Write_otp_ext_reg0xa: 0x20C
```

Normal mode

```
Write_otp_ext_reg0xa: 0x20D
```

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：

调整 +/-Vout Differential Output Voltage:

1000BT, 可调 UTP\_ext\_reg0x51, 其中 bit[11:9]为粗调, [3:0]为细调

100BT, 可调 UTP\_ext\_reg0x57, 其中 bit[14:12]为粗调, [11:8]为细调

10BTe, 可调 UTP\_ext\_reg0x57, 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。

## 6.12 Fast link down 配置

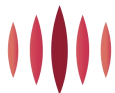
YT8521 支持 fast link down 功能, 但默认是关闭的, 打开步骤如下:

1、使能 link down 中断

2、打开 fast link down 功能, write\_otp\_ext\_reg(0x34[15],1'b0)

3、设定 fast link down 时间, write\_otp\_ext\_reg(0x37[15:13],3'b0), 时间选项如下:

0x37[15:13]	3'b000	3'b001	3'b010	3'b011	3'b100	3'b101	3'b110	3'b111
Time	0ms	5ms	10ms	20ms	40ms	80ms	160ms	320ms



## 7. RGMII 配置

### 7.1 通过 POS 设置 RX\_delay

根据 RGMII 标准，时钟的上升和下降沿采样，且时钟信号需要比数据信号 delay 1~2ns 来保证 setup/hold 时间，这个 delay 可以发送端加，也可以接收端加，甚至在早期的 RGMII 标准中通过时钟信号的板上走线延迟来达到此目的。目前的应用中一般都是发送端加，对 UTP 来说就是 RX\_CLK 的延迟，称为 rx\_delay。YT8521 是通过芯片的 power-on-strapping 管脚来实现的开/关 rx\_delay（具体见参考电路或 datasheet）。开启后，RGMII 在 125MHz（1000BT/1000BX）下的 rx\_delay 约为 1.x 纳秒，25MHz/2.5MHz（100BT/10BT）下的 rx\_delay 约为 7.x 纳秒。

### 7.2 通过寄存器加长 RX\_delay

实际使用中若需要加长 rx\_delay，以改善 setup 时间，也可以通过配置 common 扩展寄存器来配置：

Write\_ext\_reg0xa003[13:10]: 默认为 4'b0，每加 1，则 rx\_delay 延长约 150ps。（无论 RGMII 工作在 125MHz/25MHz/2.5MHz）

### 7.3 TX\_delay 的设置

UTP 的 RGMII tx\_delay 默认很少，也就是 MAC 发给 UTP 的时钟和数据信号要保证足够的 setup/hold 时间。假如 MAC 发送给 UTP 的 RGMII 的时钟和数据信号不满足要求，通常为 setup 时间不够，UTP 也可以通过配置 common 扩展寄存器来配置：

Write\_ext\_reg0xa003[3:0]: RGMII 125MHz 时的 tx\_delay，默认为 4'b1（150ps），每加 1，则 tx\_delay 延长约 150ps。

Write\_ext\_reg0xa003[7:4]: RGMII 25/2.5MHz 时的 tx\_delay，默认为 4'b1111（2.25ns），已经是最大延迟。

### 7.4 驱动能力的设置

UTP 的 RGMII 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 common 扩展寄存器来配置：

Write\_ext\_reg0xa010[5:4]: 默认为 2'b01。2'b11 为最强档，2'b00 为最弱档。

### 7.5 RGMII 支持 In-Band Status

RGMII 支持 In-Band Status，用来传递当前 RGMII 的 link status、speed 和 duplex。

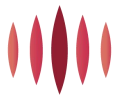
RGMII 分为 RGMII\_PHY 和 RGMII\_MAC。是接收方。

RGMII\_PHY 是 In-Band Status 的发起方。发送的状态可以通过 ext\_reg0xa004 的 bit[12] (link status), bit[15:14] (speed), bit[13] (duplex) 查询。

RGMII\_MAC 是 In-Band Status 的接收方。接收到的状态体现在 ext\_reg0xa005 的 bit[12] (link status), bit[15:14] (speed), bit[13] (duplex)。

Common EXT A004: rgmii cfg2

Bit	Symbol	Access	Default	Description
-----	--------	--------	---------	-------------



15:14	Speed_rgphy	RO	0x0	RGMII's speed information when it works as RGMII PHY. It's also the source of RGMII OOB.
13	Duplex_rgphy	RO	0x0	RGMII's duplex information when it works as RGMII PHY. It's also the source of RGMII OOB.
12	Link_up_rgphy	RO	0x0	RGMII's linkup information when it works as RGMII PHY. It's also the source of RGMII OOB.
11:10	Pause_rgphy	RO	0x0	RGMII's pause information when it works as RGMII PHY.
9	Eee_cap_rgphy	RO	0x0	RGMII's EEE capability information when it works as RGMII PHY.
8	Eee_clkstp_cap_rgphy	RO	0x0	RGMII's EEE clock stopable capability information when it works as RGMII PHY.
7:6	Speed_rgmact	RW	0x0	RGMII's speed configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.
5	Duplex_rgmact	RW	0x0	RGMII's duplex configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.
4	Link_up_rgmact	RW	0x0	RGMII's linkup configuration when it works as RGMII MAC and EXT A003 bit[15] is 1.
3:2	Pause_rgmact	RW	0x0	RGMII's pause configuration when it works as RGMII MAC.
1	Eee_cap_rgmact	RW	0x0	RGMII's EEE capability configuration when it works as RGMII MAC.
0	Eee_clkstp_cap_rgmact	RW	0x0	RGMII's EEE clock stopable capability configuration when it works as RGMII MAC.

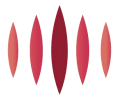
Common EXT A005: mdio cfg and rgmii oob mon				
Bit	Symbol	Access	Default	Description
15:14	Speed_rgmact_ob	RO	0x0	speed information RGMII MAC decodes from the OOB
13	Duplex_rgmact_ob	RO	0x0	duplex information RGMII MAC decodes from the OOB
12	Link_up_rgmact_ob	RO	0x0	linkup information RGMII MAC decodes from the OOB
11	Reserved	RO	0x0	Reserved
10	Bypass_mdio_watchdog	RW	0x0	bypass mdio watch dog
9:8	Reserved	RO	0x0	Reserved
7	En_mdc_la	RW	0x1	enable mdc latch for read data
6	En_phyaddr0	RW	0x1	1: to always respond to MDIO command whose PHYAD field is 0; 0: to only respond to MDIO command whose PHYAD field equals to PHY address strapping.
5	En_bdcst_addr	RW	0x0	enable broadcast address
4:0	Bdcst_addr	RW	0x0	broadcast address

Note: 在 UTP-RGMII, FIBER-RGMII, UTP-SGMII, UTP/FIBER-RGMII 和 SGMII\_MAC-RGMII\_PHY 模式中, RGMII 是 RGMII\_PHY 模式。在 SGMII\_PHY-RGMII\_MAC 模式中, 是 RGMII\_MAC 模式。

## 8. Serdes (Fiber/SGMII) 配置

### 8.1 SGMII 的速率和双工

SGMII 采用 Serdes 的传输方式, 代替 RGMII 接口, 用来连接 MAC 和 UTP。它分为 SGMII\_UTP 模式和 SGMII\_MAC 模式, 其区别是 SGMII\_UTP 需要把与之对应的网口的连接信息, 包括 link up/down, speed and duplex, 通过 SGMII 的 AN (自协商) 机制传递给 SGMII\_MAC, 最终实现 SGMII\_MAC, SGMII\_UTP 与网口速率保持一致。网口速率分为 10M, 100M, 1000M, 而 SGMII 速率是固定的 1.25G, 算上 8B/10B 造



成的损耗，它与 1000M 以太网的速率是匹配的。要匹配 10M, 100M 的传输率，SGMII 需要将同样的数据重复传 100 遍，10 遍。

通常的应用中 YT8521S 设置为 UTP-SGMII 模式与 CPU 的 SGMII\_MAC 搭配。当 YT8521S 侦测到 UTP(网口)发生连接或速率的变化时，就会在 SGMII 上发出 AN 信号，CPU MAC 收到后与之协商完成 AN 过程。此时 UTP 会显示 SGMII link up 状态，并且协商后的速度和双工也会在寄存器里体现出来。这个寄存器属于 SDS 空间寄存器，位置就 SDS MII reg 0x11，相应的位含义如下：

Sds MII 11h: Sds specific status register				
Bit	Symbol	Access	Default	Description
15:14	Speed_mode	RO	0x0	These status bits are valid only when bit10 is 1. Bit10 is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. 10---1000M; 01---100M; 00---10M
13	Duplex	RO	0x0	This status bit is valid only when bit10 is 1. Bit10 is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. 1: full duplex; 0: half duplex
12:11	Pause	RO	0x0	Pause to mac
10	Link status real-time	RO	0x0	1 = Link up
9	Rx_lpi_active	RO	0x0	rx lpi is active
8	Duplex_error	RO	0x0	realtime duplex error
7	En_flowctrl_rx	RO	0x0	realtime en_flowctrl_rx
6	En_flowctrl_tx	RO	0x0	realtime en_flowctrl_tx
5:4	Ser_mode_cfg	RO	0x0	realtime serdes working mode
3:1	Reserved	RO	0x0	Reserved
0	Syncstatus	RO	0x0	realtime syncstatus

先通过 bit10 确定目前是否 link up，若 link up，则通过 bit 15, 14, 13 获取速度和双工。

SGMII 也可以通过软件配置进入强制（force）模式，此时必须同时关闭两端的自协商。

对于 SGMII\_PHY，只需要 write\_sds\_mii\_reg0x0[12]:1'b0 关闭自协商。SGMII\_PHY 传递的速率和双工模式由对应的 UTP/Fiber 决定。

对于 SGMII\_MAC，除了关闭自协商，还需要设置速率和双工模式。具体操作如下：

Force SGMII transfer 1000FULL: write\_sds\_mii\_reg0x0: 0x8140

Force SGMII transfer 100FULL: write\_sds\_mii\_reg0x0: 0xa100

Force SGMII transfer 100HALF: write\_sds\_mii\_reg0x0: 0xa000

Force SGMII transfer 10FULL: write\_sds\_mii\_reg0x0: 0x8100

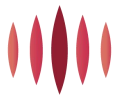
Force SGMII transfer 10HALF: write\_sds\_mii\_reg0x0: 0x8000

## 8.2 Fiber 的速率和双工

Fiber 当前状态，可以通过 sds mii reg 0x11 读到：

- Link state: bit 10. 1'b1 表示当前是 link up 的。
- speed: bit 15:14. 2b'10: 1000M, 2b'01: 100M.
- duplex: bit 13. 1'b1 表示 full duplex

Sds MII 11h: Sds specific status register



Bit	Symbol	Access	Default	Description
15:14	Speed_mode	RO	0x0	These status bits are valid only when bit10 is 1. Bit10 is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. 10---1000M; 01---100M;
13	Duplex	RO	0x0	This status bit is valid only when bit10 is 1. Bit10 is set when Auto-Negotiation is completed or Auto-Negotiation is disabled. 1: full duplex; 0: half duplex
12:11	Pause	RO	0x0	Pause to mac
10	Link status real-time	RO	0x0	1 = Link up
9	Rx_lpi_active	RO	0x0	rx lpi is active
8	Duplex_error	RO	0x0	realtime duplex error
7	En_flowctrl_rx	RO	0x0	realtime en_flowctrl_rx
6	En_flowctrl_tx	RO	0x0	realtime en_flowctrl_tx
5:4	Ser_mode_cfg	RO	0x0	realtime serdes working mode
3:1	Reserved	RO	0x0	Reserved
0	Syncstatus	RO	0x0	realtime syncstatus

可以关闭 fiber auto sensing 功能，可以强制为 1000BX 或者 100FX，配置如下：

- sds ext 0xA5 的 bit15 先清 0（关闭 autosensing）；
- 设置 ext reg 0xA006 的 Bit0（0：100FX；1：1000BX）；
- 将 0xA001.Bit15 清 0；

## 8.3 SGMII SYNC 状态位

步骤一，首先确认当前工作模式含 SGMII 模式

步骤二，read\_sds\_mii\_reg0x11[0]确认 SGMII sync 状态

## 8.4 SGMII PRBS 配置

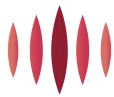
具体步骤为：

- 将 DUT 自身的 SGMII TX 与 RX 相连（或者 DUT 的 SGMII 与另一端 Link partner 的 SGMII 相连）。
- 配置 DUT（或 DUT 与 link partner）寄存器来设置要发送和接收的 PRBS 码型，支持的码型包括：PRBS7, PRBS10, PRBS31 及自定义。
- 配置 DUT（或 DUT 与 link partner）寄存器打开 PRBS 和 BERT 功能。
- 读取 DUT（或 DUT 与 link partner）寄存器以检查 SGMII 是否同步上，且无误码。

相关内部寄存器的各 bit 说明如下：

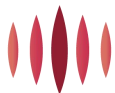
Sds EXT 05h: sds prbs cfg1		0x05		
Bit	Symbol	Access	Default	Description
15	En_prbs	RW	0x0	enable TX PRBS or self-defined pattern, the pattern type is determined by bit7:5 test_mode.
14	En_bert	RW	0x0	enable Bit Error Rate Test;
13	Prbs_invert	RW	0x0	TX error injection function, it has highest polarity than prbs_err_once and prbs_err_cont. 1: to send polarity inverted PRBS or self-defined pattern;





12	Prbs_err_cont	RW	0x0	TX error injection function, it has the lowest polarity.1: to inject error on TX pattern continuously and periodic, the period is controlled by prbs_err_rate.
11:10	Prbs_err_rate	RW	0x0	It's valid only when prbs_err_cont is 1. 2'b00: inject one error every 1024 cycles; 2'b01: inject one error every 2048 cycles; 2'b10: inject one error every 4096 cycles; 2'b11: inject one error every 8192 cycles;
9	Prbs_err_once	RW	0x0	TX error injection function.At the rising of this bit, to send polarity inverted PRBS or self-defined pattern for one cycle;
8	Test_mode_prbs31	RW	0x0	1: PRBS31, bit7:5 test_mode[2:0] has no effect.0: not PRBS31, bit7:5 test_mode[2:0] take effect then.
7:5	Test_mode	RW	0x0	Control the TX pattern in test mode: 3'h0, PRBS7; 3'h1, PRBS10; 3'h2, Fix pattern, the fix pattern is controlled by Ext.6;3'h3, 010101...; 3'h4, 00110011...; 3'h5, 00000_11111_00000_11111...; 3'h6, 0000_0000_00_1111_1111_11...; 3'h7, Increase pattern, 0->1023->0->1023...
4	Duration_check_en	RW	0x0	enable fixed number of bits check define in sds EXT BA/BB
3:0	Reserved	RO	0x0	Reserved
Sds EXT 06h: sds prbs cfg2		0x06		
Bit	Symbol	Access	Default	Description
15:10	Reserved	RO	0x0	Reserved
9:0	Fix_pattern_9_0	RW	0x0	fix pattern transmited in test_mode 2
Sds EXT 08h: sds prbs mon1		0x08		
Bit	Symbol	Access	Default	Description
15:9	Reserved	RO	0x0	Reserved
8	Err	RO	0x0	Err flag after bert has synchronized
7:1	Reserved	RO	0x0	Reserved
0	Bitsync	RO	0x0	real time synchronization status
Sds EXT 09h: sds prbs mon2		0x09		
Bit	Symbol	Access	Default	Description
15:0	Err_cnt_15_0	RO	0x0	real time lowest 16 bits received error bit counter
Sds EXT 0Ah: sds prbs mon3		0x0A		
Bit	Symbol	Access	Default	Description
15:0	Err_cnt_31_16	RO	0x0	real time highest 16 bits received error bit counter

## 8.5 SGMII 输出抖动范围



YT8521S SGMII 标称的输出抖动(Tj)为<0.3UI (如 datasheet 所列), 但根据各种极限情况下的测试结果来看, 真实的输出抖动<0.2UI, 可以满足对 PHY SGMII 抖动要求较为严格的应用 (比如对端 SGMII 接收对抖动的容忍度差)。

## 8.6 SGMII 输出预加重配置

PHY 有两个加重: main 和 post。可通过 SDS ext\_reg0xa1 设置:

Main cursor: 3 个 bit, write ext\_reg 0xa1[13, 15, 14]: 3'b111 最大, 3'b000 最小

Post cursor: 3 个 bit, write ext\_reg 0xa1[3,2,1]: 3'b111 最大, 3'b000 最小

注意: ext\_reg 0xa1 的其它位请保留原值不要变。

推荐的预加重有以下 3 档, 根据信道特性 (短, 中, 长线) 可选一种最优设置, 分别如下:

register	bit[13]: main_stage1	bit[15:14]: main_stage2	bit[3]: post_stage1	bit[2:1]: post_stage2	Note
sds ext_reg0xa1	1'b1	2'b11	1'b0	2'b00	无预加重
	1'b1	2'b10	1'b1	2'b01	中预加重
	1'b1	2'b01	1'b1	2'b10	强预加重

## 8.7 SGMII 允许接收的最大幅值

YT8521S SGMII 标称允许接收的单端最大幅值 (R\_Y2) 是 600mV, 差分最大幅值 (VRX\_DIFFp-p) 是 1200mV。从仿真的结果来看, YT8521S 允许接收更高的幅值, 单端最大幅值 (R\_Y2) 达到 800mV, 差分最大幅值 (VRX\_DIFFp-p) 达到 1600mV。

## 8.8 SGMII 输出极性反转配置

PHY 的 SGMII 支持 +/- 极性反转, 包括发送和接收两方向, 分别的配置如下:

HSIP、HSIN: sds mii\_reg0x02 Bit4, 默认为 0; 该 bit 为 1, 则 HSIP、HSIN 极性互换

HSOP、HSO: sds mii\_reg0x02 Bit5, 默认为 0; 该 bit 为 1, 则 HSOP、HSO 极性互换

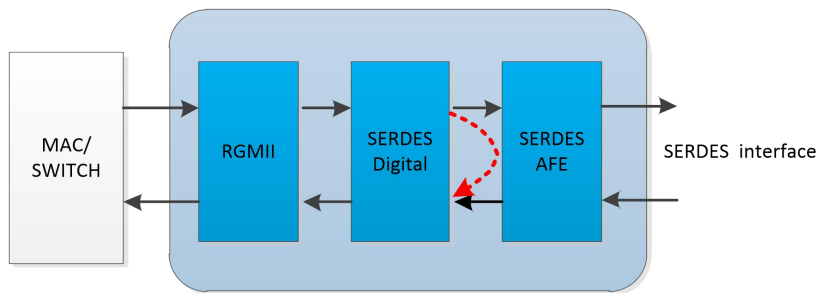
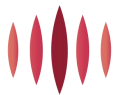


## 8.9 Loopback (回环) 模式

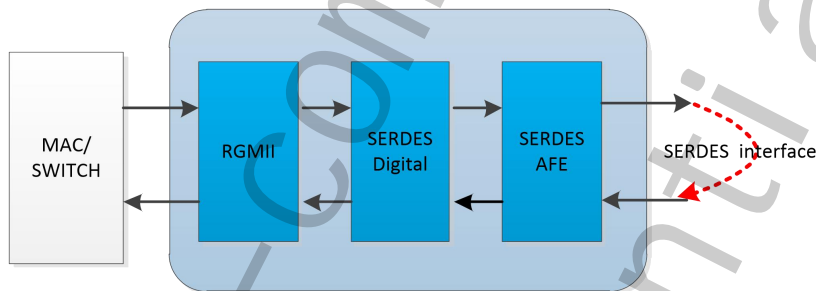
1、Internal loopback 情况下, SERDES 的 analog 电路会被 bypass, 直接将 mac 发送的数据从 SERDES 的数字电路回环回去, 也可以称作 digital loopback, 如下图所示。配置方法如下:

Write sds register mii 0.14 = 1





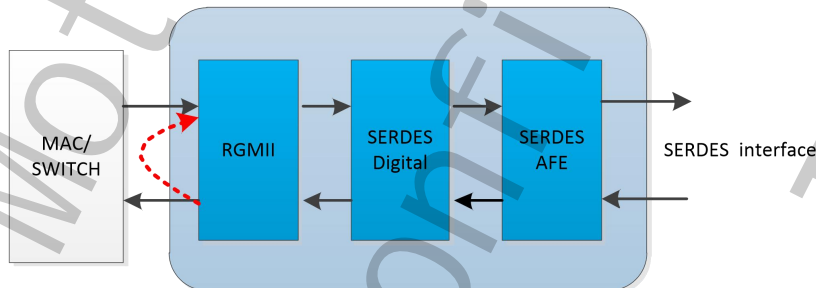
2、在 External Loopback 模式下，SERDES 发送的信号由回环线直接回到 SERDES 的接收电路。MAC 通过对比 RGMII 发送数据和接收数据，从而确保 RGMII 接口，UTP 数字和模拟功能的正确性，如下图所示。回环线将 SERDES 的发送和接收互连即可，不需要配寄存器。



3、在 Remote Loopback 模式下，RGMII RX 接收的数据被直接注入 RGMII 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：

Write common register 0xa006.6 = 1

Write sds register 0x0.15 = 1



## 8.10 包生成器和收发包统计

packet checker 应用说明，以下寄存器地址无特殊说明均指 sds UTP 寄存器空间，：

1. 打开 packet checker: 写 `ext_reg0x1A0.[15:14]=2'b10`
2. 关闭 packet checker: 写 `ext_reg0x1A0.[15:14]=2'b01`
3. 查看 checker 统计结果:

Ext\_reg0x1A3~0x1A8, 代表从 SERDES 收到的正确包的数量，寄存器为 read clear。

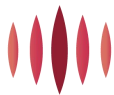
Ext\_reg0x1A9~0x1AC, 代表从 SERDES 收到的包有 crc error，寄存器为 read clear。

Ext\_reg0x1AD~0x1B2, 代表从 RGMII 收到的正确包的数量，寄存器为 read clear。

Ext\_reg0x1B3~0x1B6, 代表从 RGMII 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_reg0x1A1.[15:0]`
2. 设置发送包的个数: 写 `ext_reg0x1A2.[15:0]`，值为 0 表示一直发包。



3. 设置 IPG 长度: 写 `ext_reg1 A0.[7:4]`, 不用时可以不设置。
4. 还可以设置 `packet DA/SA` 地址, 不用时可以不设置。  
使能该功能, 写 `ext_reg0x38.[12] = 1'b1`, 关闭该功能, 写 `ext_reg0x38.[12] = 1'b0`  
设置 DA, 写 `ext_reg0x3A.[15:8]`, 则 DA 地址为 00 00 00 00 00 value  
设置 SA, 写 `ext_reg0x3A.[7:0]`, 则 SA 地址为 00 00 00 00 00 value
5. 打开 `packet generator`:  
写 `ext_reg0x1A0[15:13] = 3'b100`, 配置 `packet checker` 和 `generator`。  
写 `ext_reg0x1A0[12] = 1'b1`, 打开 `packet generator` 开始发包。
6. Polling `ext_reg0x1A0[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_reg0x1A0[12] = 1'b0`, 表示停止发包。
7. 关闭 `packet generator`:  
写 `ext_reg0x1A0[14] = 1'b1`, 关闭 `packet generator clock`  
写 `ext_reg0x1A0[15, 13, 12] = [0, 1, 0]`, 关闭 `generator` 和 `checker`

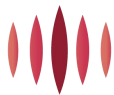
## 9. LED 灯配置

YT8521S 支持三个 LED 灯: LED0, LED1, LED2。它们对应的配置寄存器为 `utp_ext_reg0xa00c`, `utp_ext_reg0xa00d`, `utp_ext_reg0xa00e`。用来配置 LED 在不同状态下是闪、灭、亮。其中闪的频率由 `utp_ext_reg0xa00f` 定义闪的频率分为 `Blink mode2` 和 `Blink mode1` 两种。

注意: 除了 `collision` 状态和 `force LED` 状态可选 `Blink mode2` 或 `Blink mode1` (见 `utp_ext_reg0xa00b`) 外, 其它的状态若要闪只能是 `Blink mode2`。

为方便使用, 下表列出了一些常用 LED 行为对应的寄存器配置:

LED 状态分类	LED 详细状态	各连接下的 LED 配置寄存器值				备注
		UTP Only	Fiber Only	UTP 并且 Fiber	UTP 或者 Fiber	
连接: 亮 收发包: 闪	1000/100/10 Link: 亮 1000/100/10 Act: 闪 其余: 灭	0x1e00	0x5e00	0x9e00	0xde00	LED 状态中: Link 不区分全双工/半双工; Act 不区分 TX/RX ; 若想指定全/半双工或 TX/RX, 请修改如下 bit 位: bit[12]-->full duplex; bit[11]-->half duplex; bit[10]-->tx act; bit[9]-->rx act;
	1000M Link: 亮 1000M Act: 闪 其余: 灭	0x640	0x4640	0x8640	0xc640	
	100M Link: 亮 100M Act: 闪 其余: 灭	0x620	0x4620	0x8620	0xc620	
	10M Link: 亮 10M Act: 闪 其余: 灭	0x610	0x4610	0x8610	0xc610	
	100/10 Link: 亮 100/10 Act: 闪 其余: 灭	0x630	0x4630	0x8630	0xc630	
连接: 亮 收发包: 亮	1000/100/10 Link: 亮 1000/100/10 Act: 亮 其余: 灭	0x1800	0x5800	0x9800	0xd800	
	1000M Link: 亮 1000M Act: 亮 其余: 灭	0x40	0x4040	0x8040	0xc040	
	100M Link: 亮 100M Act: 亮 其余: 灭	0x20	0x4020	0x8020	0xc020	
	100M Link: 亮 100M Act: 亮 其余: 灭	0x10	0x4010	0x8010	0xc010	
	100/10M Link: 亮 100/10M Act: 亮 其余: 灭	0x30	0x4030	0x8030	0xc030	



连接: 灭 收发包: 闪	1000/100/10 Link: 灭 1000/100/10 Act: 闪 其余: 灭	0x2600	0x6600	0xa600	0xe600	
-----------------	--	--------	--------	--------	--------	--

另外, 闪烁的占空比和频率配置分别放在 `utp_ext_reg0xa00f` 中。除了连接和收发包外的一些 LED 行为配置放在 `utp_ext_reg0xa00b` 中。具体请参考 datasheet。

## 10. Interrupt (中断) 配置

YT8521S 的 34 管脚 (INT\_N/PME\_N) 为中断和 WOL 的复用管脚。需要配置 `common_ext_reg0xa00a` 来选择其一。

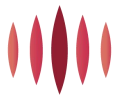
作为中断使用时, 有几点说明:

1. 中断管脚, 它为低有效。
2. 配置 `common_ext_reg0xa00a[6]` 来选择普通中断和 WOL 中断(1:WOL 中断; 0:普通中断)。当配置成普通中断时, wol 事件为普通中断中的一种特定事件; 当配置成 WOL 中断时, 34 管脚只响应 wol 事件。
3. 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。
4. 对于普通中断, 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 (UTP\_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。对于 WOL 中断, 可以通过配置 `common_ext_reg0xa00a[2]` 选择是 level trigger 还是 pulse trigger。通过配置 `common_ext_reg0xa00a[3]` 为 1'b0 来清零。如果继续监测 wol 事件, 可以将 `common_ext_reg0xa00a[3]` 再次配置为 1'b1。

相关的寄存器有:

Common EXT A00A: wol cfg				
Bit	Symbol	Access	Default	Description
15:7	Reserved	RO	0x0	Reserved
6	Pmeb_intb_sel	RW	0x0	1=wol interrupt is selected, else UTP interrupt is selected
5	Wol_src_manual	RW	0x0	enable wol src manual selection
4	Wol_src_sel	RW	0x0	1=sds is selected; 0=UTP is selected
3	Wol_en	RW	0x0	enable wol
2	Wol_int_type_sel	RW	0x0	1: PME_N is level triggerd and active LOW; When PME_N is LOW, EXT 0xA00A bit3 wol_en should be set to 0 to clear the PME_N. 0: PME_N is pulse triggered and active LOW, the pusel width is controlled by wol_lth_sel[2:1].
1:0	Wol_lth_sel	RW	0x2	wol threshlod setting 00: 84ms; 01: 168ms; 10: 336ms; 11: 672ms.

UTP 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	Reserved	RW	0x0	No used.
3	Serdes Link Failed INT mask	RW	0x0	1 = Interrupt enable
2	Serdes Link Success INT mask	RW	0x0	1 = Interrupt enable
1	Polarity changed INT mask	RW	0x0	1 = Interrupt enable
0	Jabber Happened INT mask	RW	0x0	1 = Interrupt enable

**UTP MII 13h: Interrupt Mask Register**

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

以选择 Link up 作为产生中断的事件为例：

1. 配置 common\_ext\_reg0xa00a，选择管脚 34 为中断功能：

```
write_ext_reg0xa002 0x42 #bit6 置 1
```

2. 选择 Link up 中断作为触发中断事件：

```
write_ext_reg0xa000 0x0 #选择 UTP 地址空间
```

```
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 YT8521S 进行软复位，使已有的连接断开：

```
write_ext_reg0xa000 0x0 #选择 UTP 地址空间
```

```
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

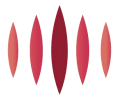
4. 读中断状态寄存器，获取已触发的中断事件

```
write_ext_reg0xa000 0x0 #选择 UTP 地址空间
```

```
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



## 11. SYNC-E（时钟输出）配置

YT8521S 的时钟输出管脚为 Pin44 CLK\_OUT，通过配置扩展寄存器 0xa012 的相关 bit，可以选择不同的源和输出不同的频率。具体设置如下：

通过 common ext reg0xa012[5] enable sync e clk output;

通过 common ext reg0xa012[3]选择输出频率为 25MHz 或 125MHz;

通过 common ext reg0xa012[2:1]选择输出时钟的来源;

注意：

1. 当选择 pll clk 作为源头时，需设置 UTP\_ext\_reg0x27[15]为 0（先 write ext reg0xa000 0 选择 UTP 寄存器空间，再改这个寄存器），让网口不要进入 sleep，否则不插线一段时间后，网口自动进入 sleep，此时 pll\_clk 就没有了。同时需要把 common ext reg0xa012[4]也置"1"。

2. 当选择 utp recovered rx clock 作为源头时，当 utp link 在 1000M slave mode/ 100M / 10M 时，源头来自 utp recovered rx clock；当 utp link 在 1000M master mode 时，源头来自本地 PLL。

Common ext reg0xa012 的各 bit 说明如下：

Common EXT A012: sync-e cfg				
Bit	Symbol	Access	default	Description
15:7	Reserved	RO	0x0	Reserved
6	UTP_do_fib	RW	0x1	1=UTP shut off until fiber link up
5	En_sync_e	RW	0x0	enable sync e clock output
4	En_sync_e_during_lnkdn	RW	0x0	always output sync e clock even when link is down
3	Clk_fre_sel	RW	0x1	1'b1: output 125m clock; 1'b0: output 25m clock
2:1	Clk_src_sel	RW	0x0	select clock source of sync e. 2'b00: pll clock; 2'b01: utp recovered rx clock; 2'b10: sds recovered rx clock; 2'b11: Reserved
0	Ptp_clk_to_sds_sel	RW	0x0	1=output transmit clock to sync e clock, used for template test; 1'b0=output ptpin clock to sync e clock

## 12. Jumbo Frame（巨型帧）配置

以太网标准包大小为 64~1518 Byte。超过 1518 Btye 的包被称为 Jumbo frame（巨型帧）。Jumbo frame 最大支持多少，取决于两个因素：FIFO 深度和频偏。其中，

FIFO 深度加大以支持更大的 Jumbo frame。所以打开 Jumbo frame 功能，无论收发包大小如何，相对于不打开此功能时，数据的传输延时都会增加。

频偏分为 DUT 与相连 PHY 的频偏和 DUT 与 MAC 的频偏。按照 IEEE 规定，前者要求是最大 200ppm。后者最大 100ppm。所以标称支持的 Jumbo frame 大小都应该满足最大频偏下的正常收发功能（实际使用中的频偏不会这么大，所以支持的最大包长比标称的 Jumbo frame 值要大）。

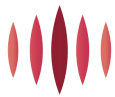
对于 YT8521S 而言，

默认 Jumbo frame 功能是关闭的。此时能支持传输的最大包长约为 9KB。

Jumbo frame 功能打开的情况下，1000/100BT 支持的最大包长为 18KB，10BT 支持的最大包长为 10KB。打开 Jumbo frame 功能后，数据在接收和发送方向的传输延时都会固定增加 16ns（1000BT/1000BX），80ns（100BT/100FX），800ns（10BT）。

寄存器配置：

Write\_ext\_reg0xa006[7]: 1'b1（其余位保持原值不变），打开 Jumbo frame 功能。



Write\_ext\_reg0xa006[7]: 1'b0 (其余位保持原值不变), 关闭 Jumbo frame 功能。

#### Common EXT A006: misc cfg

Bit	Symbol	Access	Default	Description
7	Jumbo_enable	RW	0x0	enable jumbo frame

## 13. 其它维测寄存器

### 13.1 用户自配置寄存器

因为 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, 在初始化时写入值, 在发生异常时读取此值。此寄存器为 common 扩展寄存器 ext\_reg0xa0ff, 其默认值为 0x0。

### 13.2 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:

步骤一, 把 common ext reg0xa000 写 0x0;

步骤二, 把 UTP\_ext\_reg0x53[7]置 1;

步骤三, check UTP\_ext\_reg0x56[1:0], 若这 2 位不同时为 0, 则 PLL unlock。

步骤四, 把 common ext reg0xa000 写 0x2;

步骤五, 设置 serdes ext reg0xa0[10]为 1;

步骤六, check serdes ext reg0xa0[1:0], 若这 2 为不同时为 0, 则 PLL unlock;

备注:

对 UTP 或 serdes 进行访问, 首先需要确认是否当前模式是 UTP 还是 serdes 模式;

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

#### UTP EXT 53h: analog cfg4

Bit	Symbol	Access	default	Description
7	en_vco_check	RW	0x0	enable or disable VCO_fast or VCO_slow check circuit

#### UTP EXT 56h: analog cfg7

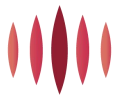
Bit	Symbol	Access	default	Description
1	Pll_vco_h	RO	0x0	PLL fast indicator
0	Pll_vco_l	RO	0x0	PLL slow indicator

#### Sds EXT A0h: analog cfg1

Bit	Symbol	Access	default	Description
10	Pll_en_vco_check	RW	0x0	enable or disable VCO_fast or VCO_slow check circuit
1	Pll_vco_h	RO	0x0	pll_vco_high flag
0	Pll_vco_l	RO	0x0	pll_vco_low flag

### 13.3 UTP big error counter





当 UTP 出现错包，丢包或断线的时候，可能是 UTP 连接通路质量不好（比如线太长或太差），或者出现了短时的大干扰。前者可以通过读 SNR 相关寄存器获取，后者通过 big error counter 或者 clipping 寄存器得到。前者说明出现了较大的 burst error，后者说明在 ADC 处出现了饱和的现象。出现该现象时，不一定会有丢包发生。其步骤为：

1. 把 common\_ext\_reg0xa000 写为 0x0
2. 把 UTP\_ext\_reg0xf8[10]清零
3. 把 UTP\_ext\_reg0x59[14:13]置 1
4. 读取 UTP\_ext\_reg0x69~0x70 的值，分 channel 0, 1, 2, 3 来判断是否发生 big error

**UTP EXT F8h: Clock gating ctrl**

Bit	Symbol	Access	default	Description
10	En_gate_UTPdbg	RW	0x1	gating UTPdbg module when it is diable

**UTP EXT 59h: debug cfg**

Bit	Symbol	Access	default	Description
14	Cnt_err_auto	RW	0x0	1 = monitor big slicer error after 1000BT training done;
13	Cnt_clp_auto	RW	0x0	1 = monitor big ADC output after 1000BT training done;

**UTP EXT 69h: debug mon16**

Bit	Symbol	Access	default	Description
15:0	Cnt_err0_big_pre	RO RC	0x0	It's fixed to 0 when EXT 59h bit14 cnt_err_auto is set, otherwise, it's the channel 0's big slicer error counter before the latest assertion of RX_DV.
7:0	Cnt_err0_big	RO RC	0x0	The counter of Channel 0's slicer error that larger than the EXT 59h bit7~0 err_big_th.

**UTP EXT 6Ah: debug mon17**

Bit	Symbol	Access	default	Description
15:0	Cnt_err1_big_pre	RO RC	0x0	Same as EXT 69h, except it's channel 1's big error counter.
7:0	Cnt_err1_big	RO RC	0x0	Same as EXT 69h, except it's channel 1's big error counter.

**UTP EXT 6Bh: debug mon18**

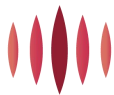
Bit	Symbol	Access	default	Description
15:0	Cnt_err2_big_pre	RO RC	0x0	Same as EXT 69h, except it's channel 2's big error counter.
7:0	Cnt_err2_big	RO RC	0x0	Same as EXT 69h, except it's channel 2's big error counter.

**UTP EXT 6Ch: debug mon19**

Bit	Symbol	Access	default	Description
15:0	Cnt_err3_big_pre	RO RC	0x0	Same as EXT 69h, except it's channel 3's big error counter.
7:0	Cnt_err3_big	RO RC	0x0	Same as EXT 69h, except it's channel 3's big error counter.

**UTP EXT 6Dh: debug mon20**

Bit	Symbol	Access	default	Description
-----	--------	--------	---------	-------------



15:0	Cnt_clp0_pre	RO RC	0x0	It's fixed to 0 when EXT 59h bit13 cnt_clp_auto is set, otherwise, it's the channel 0's big ADC output counter before the latest assertion of RX_DV.
7:0	Cnt_clp0	RO RC	0x0	The counter of Channel 0's ADC output that larger than the EXT 58h bit11~8 adc_clp_th.
<b>UTP EXT 6Eh: debug mon21</b>				
Bit	Symbol	Access	default	Description
15:0	Cnt_clp1_pre	RO RC	0x0	Same as EXT 69h, except it's channel 1's ADC output clipping counter.
7:0	Cnt_clp1	RO RC	0x0	Same as EXT 69h, except it's channel 1's ADC output clipping counter.
<b>UTP EXT 6Fh: debug mon22</b>				
Bit	Symbol	Access	default	Description
15:0	Cnt_clp2_pre	RO RC	0x0	Same as EXT 69h, except it's channel 2's ADC output clipping counter.
7:0	Cnt_clp2	RO RC	0x0	Same as EXT 69h, except it's channel 2's ADC output clipping counter.
<b>UTP EXT 70h: debug mon23</b>				
Bit	Symbol	Access	default	Description
15:0	Cnt_clp3_pre	RO RC	0x0	Same as EXT 69h, except it's channel 3's ADC output clipping counter.
7:0	Cnt_clp3	RO RC	0x0	Same as EXT 69h, except it's channel 3's ADC output clipping counter.

## 13.4 Clock drift

DUT 和 LP 之间频偏的计算方法:

把 ext reg0xf8[10]清零;

把 ext reg0x59[15]置 '1';

确认 DUT 和 Link parter 处于 link up status, 且 DUT link 在 1000BT slave mode or 100BT;

读取 ext reg0x68 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, 频差的极性为 Link parter 相对于 DUT

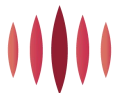
Phy EXT 68h: debug mon15				
Bit	Symbol	Access	default	Description
15:0	Integrator	RO	0x0	Channel 0's integrator.

## 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 RGMII/SGMII 接口, 或从 RGMII/SGMII 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:

模式	数据方向 (UTP 方向)	连接速度	延时绝对值(ns)	延时抖动(ns)
sgmii<->utp	sgmii_rx->utp_tx	1000M	296	16
		100M	696	104
		10M	4854	800





	utp_rx->sgmii_tx	1000M	360	32
		100M	688	80
		10M	11116	800
sgmii<->rgmii	sgmii_rx->rgmii_rx	1000M	120	8
		100M	240	72
		10M	1684	400
	rgmii_tx->sgmii_tx	1000M	136	16
		100M	536	48
		10M	4856	800
rgmii<->utp	rgmii_tx->utp_tx	1000M	208	8
		100M	616	72
		10M	4770	800
	utp_rx->rgmii_rx	1000M	256	16
		100M	312	72
		10M	7860	400
1000bx<->utp	1000bx_rx->utp_tx	1000M	296	16
	utp_rx->1000bx_tx	1000M	360	32
1000bx<->rgmii	1000bx_rx->rgmii_rx	1000M	120	8
	rgmii_tx->1000bx_tx	1000M	136	16

注：

该延时考虑所有 phy 引入的延时，指的是芯片入口端 phy 包的包头到芯片出口端 phy 包的包头的 delay，包括内部 fifo。由于 SGMII 协议的原因，tx 方向有可能会丢掉一拍前导头，这些延时也考虑在 variation 里面。

上表中绝对值在千兆模式下约为+/-10ns 误差，百兆和十兆误差会更大一些，具体数据以实测结果为准。

延时的范围为【延时绝对值，延时绝对值+延时抖动】

## 15. 硬件电路设计相关问题

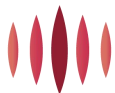
### 15.1 快速上下电的要求

YT8521S 快速上下电的要求和条件，根据实验和仿真，结论如下：

1. 当 3.3V 电压掉到 3.3~2.5V 或 0~1.4V，再上升回 3.3V，芯片会正常工作；
2. 当 3.3V 电压掉到 1.4~2.5V 的范围，再上升回 3.3V，则分为两种情况：

如果期间芯片的复位管脚为低（一般小于 1V），再升回 3.3V 后，此信号拉到高，则芯片经过复位，会正常工作。

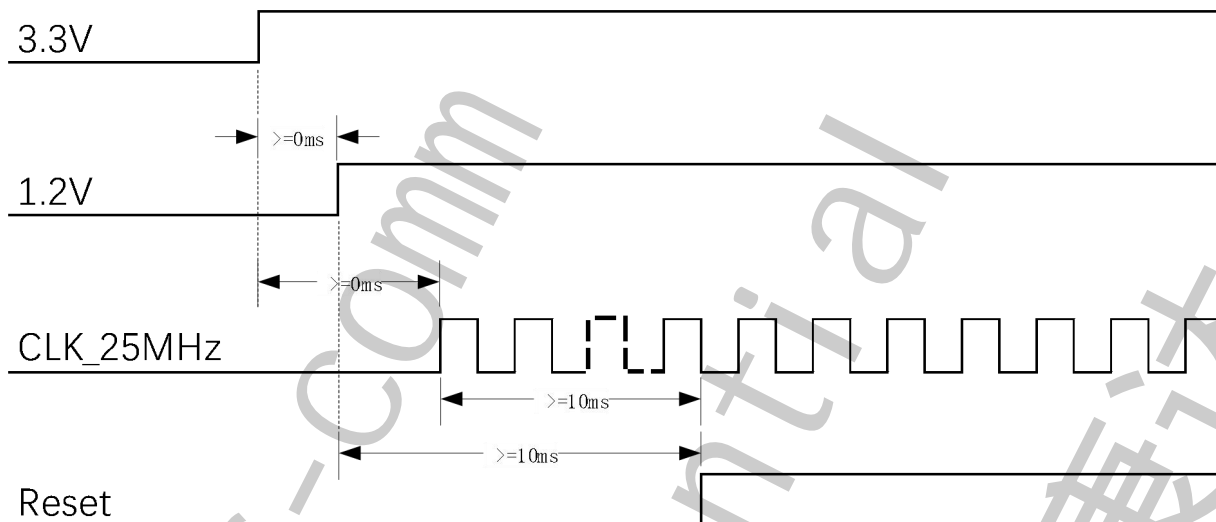
如果期间芯片的复位管脚为高（一般高于 1.5V），再升回 3.3V 后，则芯片未经过复位，可能工作不正常，但在下一次硬复位后即可正常工作。



## 15.2 上电复位时序要求

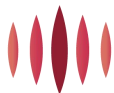
YT8521S 芯片支持多种模式。

一、对于不包含 RGMII 相关的模式，它的上电时序要求如下：



说明：

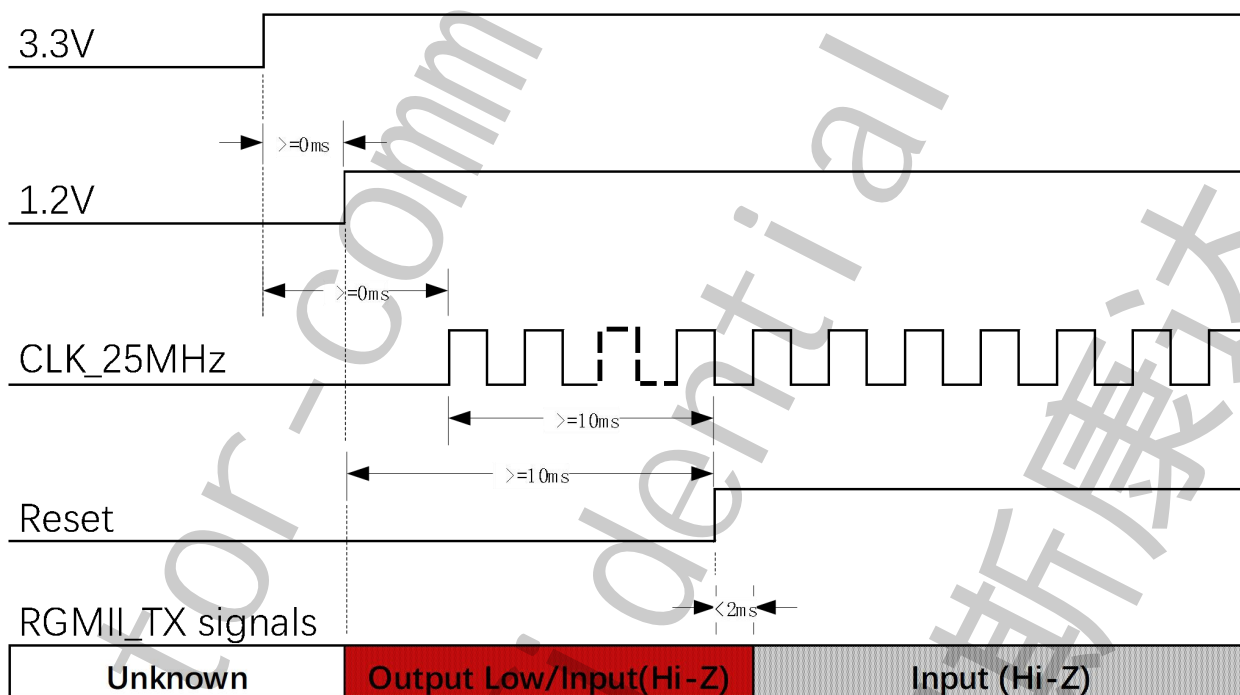
1. 若 1.2V 或 CLK\_25MHz 由芯片内部电源/晶体振荡电路产生，则它们与 3.3V 的时序由芯片内部控制，满足芯片内部时序要求。其中可能由于 3.3V 上升慢，而 1.2V 上升快，导致 1.2V 比 3.3V 先稳定，这是正常现象，无问题。或者 CLK\_25MHz 在 3.3 上电未稳定时即有输出也是正常现象。
2. 若 1.2V 或 CLK\_25MHz 由外灌，则需遵守上述时序。其中 1.2V 和 CLK\_25MHz 并不要求先后，但要求 1.2V 和 CLK\_25MHz 都稳定并保持至少 10ms 后再释放 Reset 信号。

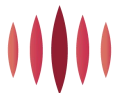


二、对于含 RGMII 的模式，要注意 YT8521S 的 RGMII TX 管脚在上电 reset 第一次释放前的状态。此时有概率使 RGMII 的 TX 管脚（包括 TXD0~3, TXC, TXCTL）为输出低状态，从而拉低与之相连的 CPU/MAC RGMII TX 相关管脚。若 CPU/MAC 的这些管脚要求在此刻为高（通常为 power-on-strapping），则有可能不满足其要求。

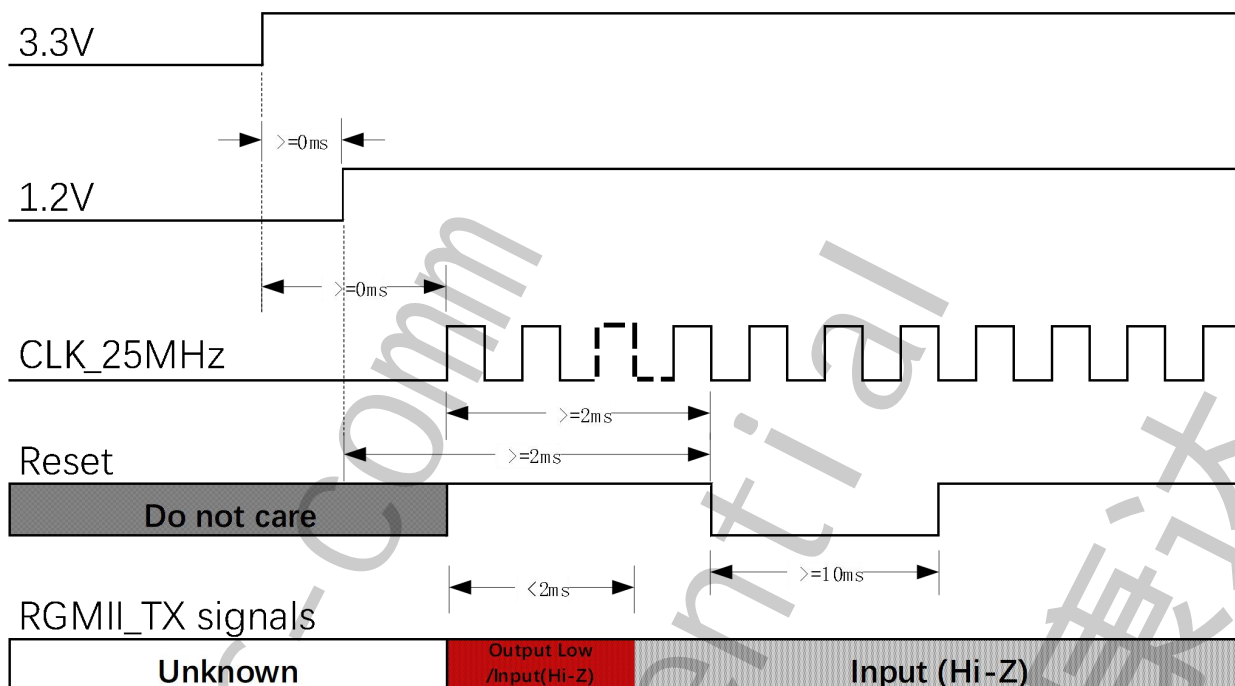
要避免上述问题，在使用带 RGMII 模式时，上电复位的时序可以有以下两种方式：

方式一：如正常时序要求相同，只是注意 RGMII\_TX signals 在上电 reset 第一次释放前及之后 2ms 内可能存在的不确定态。





方式二：



特别注意：

1. 上述两种时序中的 RGMII\_TX signals 包含 RGMII TXD[3:0], TXC, TXCTL 信号。
2. 上述两种时序中的 RGMII\_TX signals 中的 Output Low/Input(Hi-Z)状态就是上文提到的不确定态，与之相连的 CPU/MAC RGMII 信号在此阶段可能被拉为低。
3. CPU/MAC 在上电阶段的 RGMII TX 信号如果没有作为 power-on-strapping，或者需要在此阶段为低时，则这段不确定态不会对 CPU/MAC 造成任何影响。
4. 若 CPU/MAC 用 RGMII TX 管脚作为 power-on-strapping 并且恰巧在这一阶段 load power-on-strapping 信息（通常在 CPU/MAC reset 释放后的一小段时间），则可能导致 CPU/MAC power-on-strapping 出错。这时要通过调整 CPU/MAC 和 UTP 的上电和复位时序，避开两者重合时间。
5. 不确定态只存在于上电第一次 reset 释放前，只要释放过一次，后续再 reset 都不会出现不确定态。

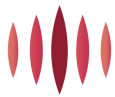
### 15.3 硬复位后延时进行 MDIO 操作要求

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。

### 15.4 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~20MHz rms jitter < 3ps;
- broadband rms jitter < 9ps



（其它的要求，比如幅度，占空比等参见 datasheet）

## 15.5 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）。