

Anagha Jagannath, Melissa Chang

摘要

串行千兆位媒体独立接口 (SGMII) 是一种通信接口类型，用于将以太网 PHY 连接到介质访问控制 (MAC)。SGMII 使用低电压差分信号 (LVDS) 以 10/100/1000/2500Mbps 速率接收和发送数据。SGMII 中有四个数据信号，两个用于 TX 路径，两个用于 RX 路径。也可以选择使用额外的两个信号，一个用于 RX CLK，一个用于 TX CLK。SGMII 相较于其他 MAC 接口的优势是 SGMII 支持千兆位通信，并且射频辐射更低。

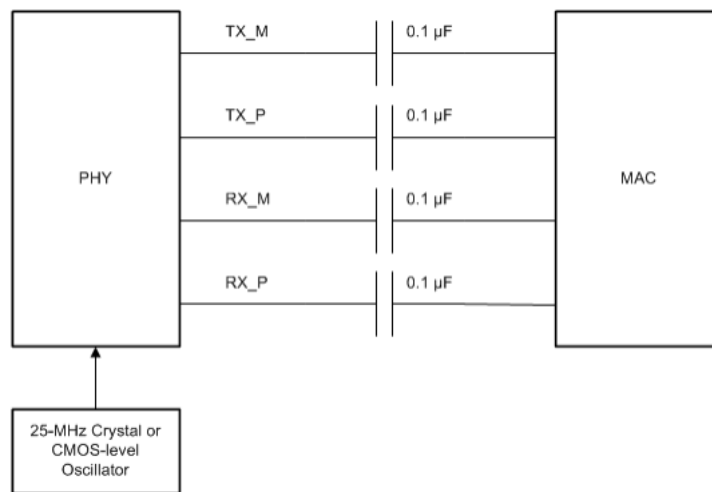


图 1-1. 四线 SGMII 实现

内容

1 MAC 接口疑难解答 - SGMII	2
1.1 验证自举配置.....	2
1.2 读取并检查寄存器值.....	3
1.3 自动协商.....	4
1.4 吞吐量和环回测试.....	4
1.5 检查时钟信号.....	11
1.6 测量 SGMII 眼图.....	11
1.7 SGMII 布局.....	13
2 总结	13
3 参考资料	13

商标

所有商标均为其各自所有者的财产。

1 MAC 接口疑难解答 - SGMII

本指南旨在解决常见问题，例如使用 TI 汽车和工业以太网 PHY 实现 SGMII MAC 接口时可能出现的链路断开和数据包错误。

1.1 验证自举配置

在继续进行任何进一步调试之前，请检查数据表的 *Strap 配置* 部分，以检查所使用的 PHY 是否已正确配置进入 SGMII 模式。

PHY 的 strap 状态可通过读取 CHIP_SOR 或 SOR_VECTOR 寄存器（具体取决于器件）进行确认。这些寄存器可以通过扩展寄存器访问进行访问。

图 1-1 是 DP83TG720S-Q1 汽车以太网 PHY 的 SGMII strap 配置示例。DP83TG720S-Q1 的默认 MAC 接口为 SGMII，因此引脚 RX_D0、RX_D1 和 RX_D2 可以保持悬空。

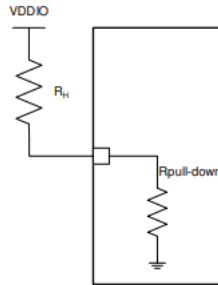


图 1-1.2 级自举电路

1.1.1 DP83TG720S-Q1 的 SGMII 自举配置

表 1-1.2 级自举

引脚名称	引脚编号	配置 (strap) 模式	配置 (strap) 功能
RX_D0	26	1 (默认值)	MAC[0] = 0
		2	MAC[0] = 1
RX_D1	25	1 (默认值)	MAC[1] = 0
		2	MAC[1] = 1
RX_D2	24	1 (默认值)	MAC[2] = 0
		2	MAC[2] = 1

表 1-2. MAC 接口选择自举

MAC[2]	MAC[1]	MAC[0]	说明
0	0	0	SGMII (4 线制)

1.2 读取并检查寄存器值

读取寄存器并验证 SGMII 链路是否接通，SGMII 自动协商是否完成。请注意，某些寄存器的初始值可能会因 strap 配置选项而异。

下面展示了 DP83TG720-Q1 和 DP83869 的寄存器转储示例。这些是 MDI 链路建立、PHY 处于 SGMII 模式、SGMII 链路建立时的可能值。

表 1-3. DP83TG720S-Q1 汽车 PHY - SGMII 寄存器值

寄存器地址	寄存器名称	示例值	说明
0x0000	BMCR	0x0140	位 [14] 可用于配置 MII 环回。
0x0001	BMSR	0x0145	位 [2] 指示 MDI 链路已建立。
0x0011	MII_REG_11	0x000B	位 [11] 可置为有效以启动 SGMII 软复位。
0x0428	A2D_REG_40	0x6002	位 [14:13] 指示输出电压摆幅为 720mV。可切换位以增大 SGMII 输出摆幅。
0x045D	SOR_VECTOR_1	0x0000	位 [13] 表示 SGMII 已启用，位 [8:6] 表示 PHY 已配置进入的 MAC 模式。
0x0608	SGMII_CTRL_1	0x027B	位 [0] 表示 SGMII 自动协商被启用，位 [2:1] 表示自协商计时器为 2us。可以切换这些位以调整计时器。
0x060A	SGMII_STATUS	0x0d46	位 [11] 表示 SGMII 链路已建立，位 [10] 表示 SGMII 自协商完成。
0x060C	SGMII_CTRL_2	0x001B	当没有 SGMII 链路时，位 [6] 可置为有效以重新启动 SGMII 自动协商。位 [5:3] 表示 TX，而位 [2:0] 表示 RX FIFO 半满阈值。
0x060D	SGMII_FIFO_STATUS	0x0000	位 [0] 指示数据包下溢。位 [1] 指示数据包溢出。
0x0639	PKT_STAT_1	0x0000	寄存器值指示 TX 数据包计数器。
0x063A	PKT_STAT_2	0x0000	寄存器值指示 TX 数据包计数器。
0x063B	PKT_STAT_3	0x0000	TX 数据包错误计数器。
0x063C	PKT_STAT_4	0x0000	寄存器值指示 RX 数据包计数器。
0x063D	PKT_STAT_5	0x0000	寄存器值指示 RX 数据包计数器。
0x063E	PKT_STAT_6	0x0000	RX 数据包错误计数器。

备注

按顺序读取寄存器 0x0639、0x063A、0x063B 时，这些寄存器被清零。同样，按顺序读取寄存器 0x063C、0x063D、0x063E 时，这些寄存器也被清零。

表 1-4. DP83869HM 工业 PHY - SGMII 寄存器值

寄存器地址	寄存器名称	示例值	说明
0x0000	BMCR	0x1140	位 [14] 可用于配置 MII 环回。
0x0001	BMSR	0x796D	位 [2] 指示 MDI 链路已建立
0x0014	GEN_CFG2	0x29C7	位 [7] 启用 SGMII 自动协商。
0x0031	GEN_CFG3	0x10B0	可切换位 [6:5] 以调整自动协商计时器。
0x0037	SGMII_AUTO_NEG_STATUS	0x0001	位 [0] 指示 SGMII 自动协商完成。

1.3 自动协商

以太网和 SGMII 都具有相互独立的自协商过程。以太网 PHY 和 PHY 链路伙伴之间通过 MDI 线路进行以太网自动协商。在此过程中，两个器件会交换有关速度、双工模式和流控制的信息，并以两个链路伙伴广播的最大能力建立链路。

SGMII 自动协商是 PHY 向 MAC 发送更新后的控制信息的过程。此控制信息在 [Cisco SGMII 标准](#) 中指定。当 MAC 收到此信息时，MAC 通过将确认位置为有效来确认收到更新后的控制信息。在 TI 以太网 PHY 中，此确认位与 *SGMII Page Received* 寄存器位相关联。在 SGMII 自动协商中，不存在功能最大化，只是 PHY 和 MAC 之间的信息交换。

如果发生 SGMII 链路建立问题，请检查在 PHY 和 MAC 中自动协商是否都已禁用或都已启用。在任一情况下，PHY 和 MAC 必须支持相同的通信速度。

如果 PHY 先于 MAC 启动，则可能需要重新启动 SGMII，MAC 才能接收控制信息，从而成功建立链路。

如果 SGMII 链路建立仍然失败，还可以调整某些 TI 以太网 PHY 的自动协商计时器，以使 PHY 端的整个自动协商周期更慢或更快。要调整计时器，请设置器件数据表中所述的 *sgmii_autoneg_timer* 位字段，然后重新启动 SGMII 自动协商或通过写入 $0x1F=0x4000$ 复位 PHY。

1.4 吞吐量和环回测试

为了验证通信误差是否与 SGMII 相关，我们结合使用了吞吐量和环回测试：首先是简单的测试设置，然后缓慢地向更复杂、更精确的测试设置转变。其中包括有关如何以 Windows 或 Linux 系统为例运行这些测试的建议，但每个系统都是独特的，这些方法并不总是适用于所有设计。有关如何对您的系统运行吞吐量测试的更多详细信息，请联系您的处理器或软件供应商。

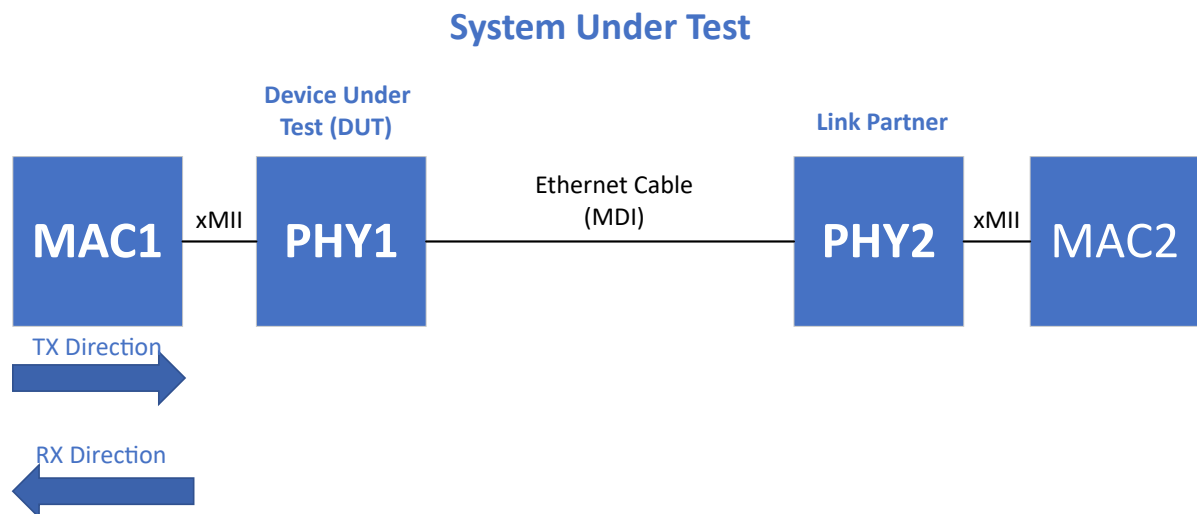


图 1-2. SGMII 设置示例

1. **双向吞吐量测试**：SGMII 链路建立后，通过执行吞吐量测试来验证是否存在通信错误。
2. **RX 和 TX 吞吐量测试**：如果存在数据包错误，请确定 RX 或 TX 路径中是否发生错误。可以通过仅在 TX 方向发送数据包并检查发送器是否接收到所有正确的数据来识别错误路径。如果有错误，则问题出在 TX 路径上。否则，错误来自 RX 路径。
3. **使用固定数量的数据包进行 RX 和 TX 吞吐量测试**：如果出现数据包丢失，请确定 RX 或 TX 路径中是否发生了丢失。要确定数据包丢失的位置，请尝试沿一个方向发送固定数量的数据包。
4. **环回测试**：使用环回来确认问题是否在 SGMII 块内。

1.4.1 双向吞吐量测试

测试吞吐量的一种简单方法是通过 ping。在 TX 和 RX 方向上进行 Ping 测试通信。以下是有关如何在 Windows 命令提示符下或 Linux 终端运行 ping 的示例：

```
ping -c 10 -i 0.5 <IP address> //ping <IP address> 10 times at an interval of 0.5 seconds
```

图 1-3 展示了成功 ping 结果的示例，图 1-4 展示了 Linux 终端中失败 ping 结果的示例。

```
root@j7-evm:~# ping 192.168.1.2 -c 10 -i 0.01
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: seq=0 ttl=64 time=0.257 ms
64 bytes from 192.168.1.2: seq=1 ttl=64 time=0.079 ms
64 bytes from 192.168.1.2: seq=2 ttl=64 time=0.078 ms
64 bytes from 192.168.1.2: seq=3 ttl=64 time=0.073 ms
64 bytes from 192.168.1.2: seq=4 ttl=64 time=0.131 ms
64 bytes from 192.168.1.2: seq=5 ttl=64 time=0.074 ms
64 bytes from 192.168.1.2: seq=6 ttl=64 time=0.071 ms
64 bytes from 192.168.1.2: seq=7 ttl=64 time=0.071 ms
64 bytes from 192.168.1.2: seq=8 ttl=64 time=0.069 ms
64 bytes from 192.168.1.2: seq=9 ttl=64 time=0.072 ms

--- 192.168.1.2 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.069/0.097/0.257 ms
root@j7-evm:~#
```

图 1-3. 成功的 Ping 结果

```
root@j7-evm:~# ping 192.168.1.3 -c 10 -i 0.01
PING 192.168.1.3 (192.168.1.3): 56 data bytes

--- 192.168.1.3 ping statistics ---
10 packets transmitted, 0 packets received, 100% packet loss
root@j7-evm:~#
```

图 1-4. 失败的 Ping 结果

1.4.2 RX 和 TX 吞吐量测试

要在一个或两个方向上以最大吞吐量运行测试，可以使用 iPerf3。IPerf3 可以安装在 MAC、Windows 和 Linux 操作系统上。安装后，只需在终端或 CMD 窗口中输入 iPerf3 的安装目录，即可访问 iPerf3。以下示例中列出了带有 iperf 的一些示例命令。

```

C:\Users\labuser\Downloads>cd iperf3.1.1_64

C:\Users\labuser\Downloads>cd iperf3.1.1_64 ipconfig
// ipconfig provides IPV4 address used for ethernet configuration on PC

iperf3 -s // runs iperf as server on one PC

iperf3 -c <server PC IP address> -p <server port number> //runs as client to sniff out packets

iperf3 -h // displays other useful shortcuts and commands
    
```

以下是在 Linux 终端中通过 Iperf3 以 100Mbps 速率成功传输数据的示例。图 1-5 展示了服务器端，而图 1-6 展示了客户端。

```

root@j7-evm:~# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 192.168.1.2, port 47326
[ 5] local 192.168.1.1 port 5201 connected to 192.168.1.2 port 47342
[ ID] Interval           Transfer             Bitrate
[ 5]  0.00-1.00      sec    10.8 MBytes    90.4 Mbits/sec
[ 5]  1.00-2.00      sec    11.2 MBytes    94.2 Mbits/sec
[ 5]  2.00-3.00      sec    11.2 MBytes    94.1 Mbits/sec
[ 5]  3.00-4.00      sec    11.2 MBytes    94.1 Mbits/sec
[ 5]  4.00-5.00      sec    11.2 MBytes    94.2 Mbits/sec
[ 5]  5.00-6.00      sec    11.2 MBytes    94.1 Mbits/sec
[ 5]  6.00-7.00      sec    11.2 MBytes    94.2 Mbits/sec
[ 5]  7.00-8.00      sec    11.2 MBytes    94.2 Mbits/sec
[ 5]  8.00-9.00      sec    11.2 MBytes    94.1 Mbits/sec
[ 5]  9.00-10.00     sec    11.2 MBytes    94.2 Mbits/sec
[ 5] 10.00-10.04     sec     488 KBytes    94.1 Mbits/sec
-----
[ ID] Interval           Transfer             Bitrate
[ 5]  0.00-10.04     sec    112 MBytes    93.8 Mbits/sec
-----
Server listening on 5201
    
```

图 1-5. 成功的 Iperf 服务器结果

```

root@j7-evm:~# iperf3 -c 192.168.1.1 -p 5201
Connecting to host 192.168.1.1, port 5201
[ 5] local 192.168.1.2 port 47342 connected to 192.168.1.1 port 5201
[ ID] Interval           Transfer             Bitrate          Retr          Cwnd
[ 5]  0.00-1.00      sec    11.8 MBytes    99.3 Mbits/sec     0           100 KBytes
[ 5]  1.00-2.00      sec    11.2 MBytes    93.6 Mbits/sec     0           112 KBytes
[ 5]  2.00-3.00      sec    11.2 MBytes    93.8 Mbits/sec     0           112 KBytes
[ 5]  3.00-4.00      sec    11.2 MBytes    93.8 Mbits/sec     0           112 KBytes
[ 5]  4.00-5.00      sec    11.2 MBytes    93.8 Mbits/sec     0           112 KBytes
[ 5]  5.00-6.00      sec    11.4 MBytes    95.9 Mbits/sec     0           112 KBytes
[ 5]  6.00-7.00      sec    11.2 MBytes    93.8 Mbits/sec     0           112 KBytes
[ 5]  7.00-8.00      sec    11.2 MBytes    93.8 Mbits/sec     0           112 KBytes
[ 5]  8.00-9.00      sec    11.2 MBytes    93.8 Mbits/sec     0           112 KBytes
[ 5]  9.00-10.00     sec    11.2 MBytes    93.8 Mbits/sec     0           112 KBytes
-----
[ ID] Interval           Transfer             Bitrate          Retr
[ 5]  0.00-10.00     sec    113 MBytes    94.6 Mbits/sec     0
[ 5]  0.00-10.04     sec    112 MBytes    93.8 Mbits/sec
-----
iperf Done.
root@j7-evm:~#
    
```

图 1-6. 成功的 Iperf 客户端结果

1.4.3 使用固定数量的数据包进行 RX 和 TX 吞吐量测试

对于 2018 年之后发布的单线对以太网器件，可以在数据表中找到 RX/TX 数据包计数器和错误计数器寄存器。例如，在 DP83TG720S-Q1 中，这些寄存器为 0x639-0x63E。按照以下步骤检查 RX 和 TX 方向的通信：

1. 打开待测系统电源并建立其链路。
2. 禁用系统中的所有后台数据包。
3. 读取 0x639-0x63E 以清除寄存器。
4. 在系统中沿两个方向发送 100 个数据包。
5. 检查 TX/RX 计数器是否正确递增，以及寄存器中是否没有 CRC 错误，寄存器需要具有以下值：
0x639=0x64、0x63A=0x0、0x63B=0x0、0x63C=0x64、0x63D=0x0、0x63E=0x0。

图 1-7 展示了使用 DP83TC812 的数据包计数器寄存器检查 RX 和 TX 通信的示例。脚本 ./packetcounters.sh 按顺序输出寄存器 0x639-0x63E。寄存器值表明 PHY 中发送了 10 个数据包，接收了 10 个数据包，且没有错误。因此，RX 和 TX 通信正常工作。

```

root@j7-evm:~# ./packetcounters.sh #read packet counter register 0x639-0x63E
0000
0000
0000
0000
0000
0000
root@j7-evm:~#
root@j7-evm:~#
root@j7-evm:~# ping 192.168.1.2 -i 0.001 -c 10 #send 10 packets
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: seq=0 ttl=64 time=0.272 ms
64 bytes from 192.168.1.2: seq=1 ttl=64 time=0.078 ms
64 bytes from 192.168.1.2: seq=2 ttl=64 time=0.089 ms
64 bytes from 192.168.1.2: seq=3 ttl=64 time=0.070 ms
64 bytes from 192.168.1.2: seq=4 ttl=64 time=0.070 ms
64 bytes from 192.168.1.2: seq=5 ttl=64 time=0.069 ms
64 bytes from 192.168.1.2: seq=6 ttl=64 time=0.068 ms
64 bytes from 192.168.1.2: seq=7 ttl=64 time=0.069 ms
64 bytes from 192.168.1.2: seq=8 ttl=64 time=0.072 ms
64 bytes from 192.168.1.2: seq=9 ttl=64 time=0.088 ms

--- 192.168.1.2 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.068/0.094/0.272 ms
root@j7-evm:~#
root@j7-evm:~#
root@j7-evm:~# ./packetcounters.sh #read packet counter register 0x639-0x63E
0x000a
0000
0000
0x000a
0000
0000

```

图 1-7. 使用 DP83TC812 检查吞吐量和数据包计数器

如果正在使用不带数据包计数器寄存器的以太网 PHY，或者无法在系统中禁用后台数据包，则可以使用 Wireshark 或 Tcpdump 来探查数据包。按照以下步骤检查 TX 方向的通信：

1. 打开待测系统电源并建立其链路。
2. 开始在链路伙伴的终端上运行 tcpdump 或 wireshark。
3. 从受测器件 (DUT) 向链路伙伴 Ping/发送 10 个数据包。请参阅图 1-8。
4. 如果接收到所有 10 个数据包，则可能会有 10 个请求显示在 wireshark/tcpdump 中。请参阅图 1-9。

```

root@j7-evm:~# ping 192.168.1.2 -c 10 -i 0.01
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: seq=0 ttl=64 time=0.296 ms
64 bytes from 192.168.1.2: seq=1 ttl=64 time=0.138 ms
64 bytes from 192.168.1.2: seq=2 ttl=64 time=0.084 ms
64 bytes from 192.168.1.2: seq=3 ttl=64 time=0.077 ms
64 bytes from 192.168.1.2: seq=4 ttl=64 time=0.080 ms
64 bytes from 192.168.1.2: seq=5 ttl=64 time=0.144 ms
64 bytes from 192.168.1.2: seq=6 ttl=64 time=0.082 ms
64 bytes from 192.168.1.2: seq=7 ttl=64 time=0.074 ms
64 bytes from 192.168.1.2: seq=8 ttl=64 time=0.076 ms
64 bytes from 192.168.1.2: seq=9 ttl=64 time=0.106 ms

--- 192.168.1.2 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.074/0.115/0.296 ms

```

图 1-8. 从 DUT 向链路伙伴发送 10 个数据包

```

root@j7-evm:~# tcpdump
[ 1303.740628] device eth4 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth4, link-type EN10MB (Ethernet), capture size 262144 bytes
08:18:21.008389 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 0, length 64
08:18:21.008490 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 0, length 64
08:18:21.018407 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 1, length 64
08:18:21.018465 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 1, length 64
08:18:21.028451 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 2, length 64
08:18:21.028467 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 2, length 64
08:18:21.038471 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 3, length 64
08:18:21.038485 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 3, length 64
08:18:21.048494 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 4, length 64
08:18:21.048510 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 4, length 64
08:18:21.058574 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 5, length 64
08:18:21.058587 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 5, length 64
08:18:21.068604 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 6, length 64
08:18:21.068617 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 6, length 64
08:18:21.078626 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 7, length 64
08:18:21.078638 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 7, length 64
08:18:21.088649 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 8, length 64
08:18:21.088663 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 8, length 64
08:18:21.098675 IP 192.168.1.1 > j7-evm: ICMP echo request, id 1286, seq 9, length 64
08:18:21.098709 IP j7-evm > 192.168.1.1: ICMP echo reply, id 1286, seq 9, length 64
^C
20 packets captured
20 packets received by filter

```

图 1-9. TCP 转储日志示例

除了链路伙伴和 DUT 的角色相反外，测试 RX 方向的步骤是类似的。

1.4.4 环回测试

环回测试可以确认任何错误都是 SGMII 连接导致的。如果在 MII 环回期间发生错误，则问题是由 MII 接口导致的。如果在反向环回期间发生错误，则问题是由 MDI 接口导致的。

1.4.4.1 MII 环回

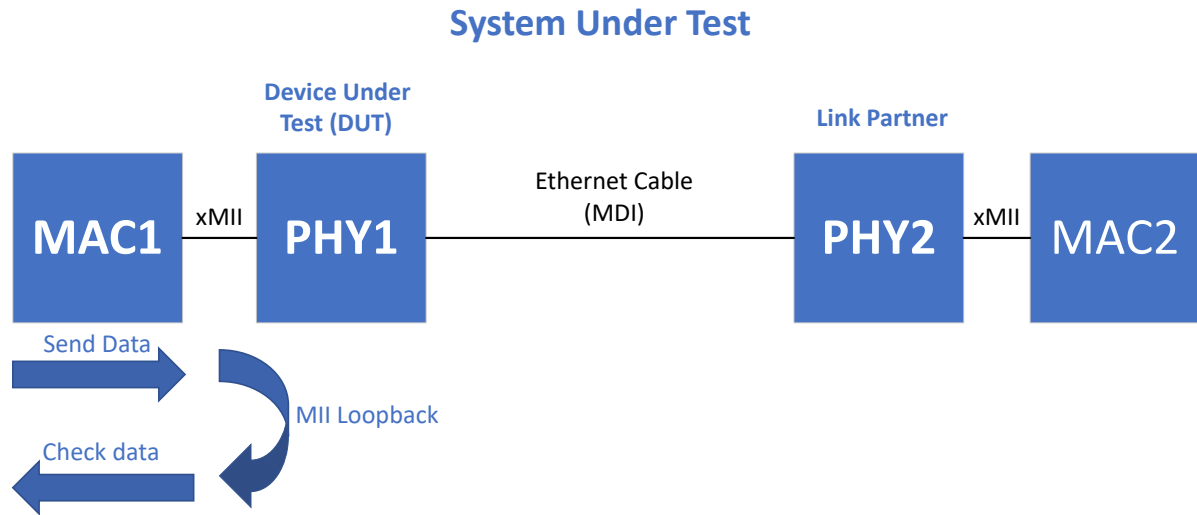


图 1-10. MII 环回图

1. 打开受测器件的电源
2. 在寄存器 0x0 中启用 MII 环回
3. 从 MAC 发送固定数量的数据包。如果使用 ping 发送数据包，则不要使用与源相同的 IP 地址，因为数据包绝不会传输到 PHY。使用随机 IP 地址。
4. 如果发送的所有数据包也都收到了，则 SGMII 连接正常工作。Tcpdump 或 Wireshark 可用于检查接收的数据包数量。

在图 1-11 中，启用 MII 环回后从 MAC 输出 10 个数据包。Ping 显示发送了 10 个数据包，收到 0 个数据包。这是预期结果，因为目标地址不存在。但是，TCPdump 日志会显示入站数据包中的所有出站数据。总共有 20 条消息。将传出 10 条消息，然后通过环回发送回主机。因此，PHY 内的 MII 块正常工作。

```

root@j7-evm:~#
root@j7-evm:~# phytool write eth4/0x8/0x0 0x6100 #set reigster 0x0=0x6100 at PHY Address 0x8, Ethernet Port 4
root@j7-evm:~#
root@j7-evm:~# ifconfig eth4 192.168.1.1 #config Ethernet port 4 IP address
root@j7-evm:~#
root@j7-evm:~#
root@j7-evm:~# tcpdump & #run tcpdump in background
[2] 1464
root@j7-evm:~# tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
root@j7-evm:~#
root@j7-evm:~# ping 192.168.1.2 -c 10 -i 0.001 #ping 10 packets
PING 192.168.1.2 (192.168.1.2): 56 data bytes
08:08:40.317882 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317903 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317909 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317914 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317919 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317923 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317928 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317934 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317938 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317943 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317875 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317902 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317907 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317913 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317918 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317922 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317927 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317932 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317937 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
08:08:40.317941 IP j7-evm > j7-evm: ICMP host 192.168.1.2 unreachable, length 92
--- 192.168.1.2 ping statistics ---
10 packets transmitted, 0 packets received, 100% packet loss
root@j7-evm:~#
root@j7-evm:~# █
  
```

图 1-11. 在 Linux 中测试 MII 环回

要仅显示传出或传入的数据包，请使用以下命令：

```

tcpdump dst hst <host IP> & #Packets coming to the host
tcpdump src hst <host IP> & #Packets sent out from the host
  
```

备注

对于 TI SPE 以太网 PHY 中的寄存器 0x639-0x63E，MII 环回不会使数据包计数器递增。请改用数字环回作为环回选项，以使这些计数器递增。

1.4.4.2 反向环回

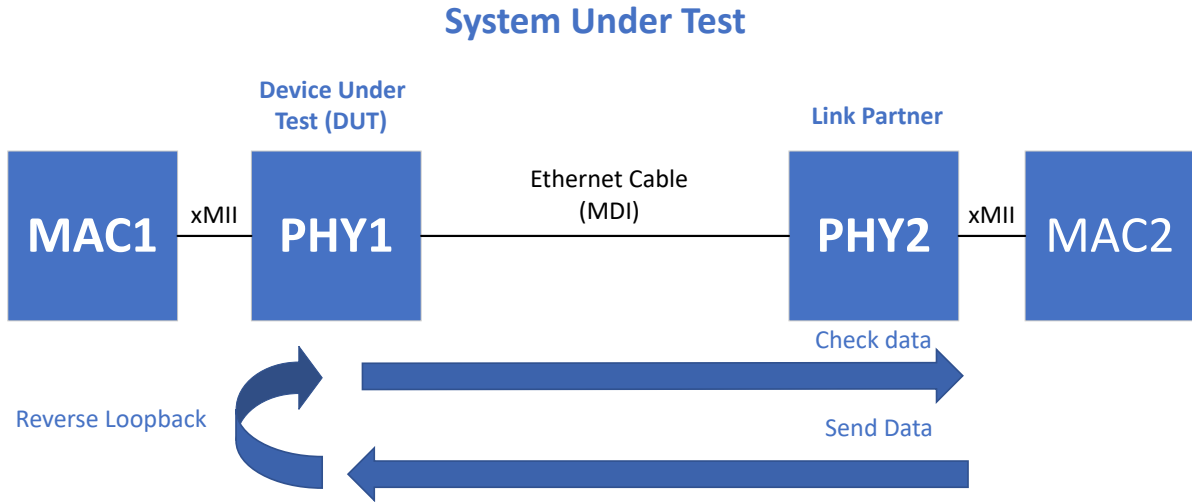


图 1-12. 反向环回图

1. 打开受测系统的电源。
2. 在寄存器 0x16 中启用反向环回。
3. 从 PHY 的 MDI 链路伙伴发送固定数量的数据包。如果使用 ping，请勿使用与源相同的 IP 地址。
4. 如果发送的所有数据包也都收到了，则 MDI 连接正常工作。

1.5 检查时钟信号

使用不符合数据表中提到的 PPM 规范的时钟源会导致寄存器访问、数据包丢失、链路中断或 PHY 完全无法运行等问题。

请联系时钟源供应商获取晶体报告并直接测量 PPM，以确认所使用的时钟源符合正确的 PPM 规范。

要测量时钟源的 PPM，请使用信号分析仪测量 PHY CLK_OUT 引脚的平均频率。

备注

请勿直接探测 XI/XO 引脚以测量 PPM，因为这些引脚更敏感，可能会受到探头电容的影响。

如果信号分析仪测得的平均频率为 f_{avg} ，则可以通过以下公式计算 PPM：

$$\left[\frac{(f_{expected} - f_{avg})}{f_{expected}} \right] \times 10^6 \quad (1)$$

例如，如果要求为 $25\text{MHz} \pm 100\text{ppm}$ ，则平均频率必须在 $24.997500\text{MHz} - 25.002500\text{MHz}$ 范围内。

1.6 测量 SGMII 眼图

以下部分详细介绍了 SGMII 的眼图模板要求。

1.6.1 SGMII 眼图模板要求

要测量 PHY 的 SGMII 输出，请在输入引脚附近使用示波器进行测量。

PHY 上 TX_M 和 TX_P 引脚的输入需要遵循图 1-13 中所示的最低要求。

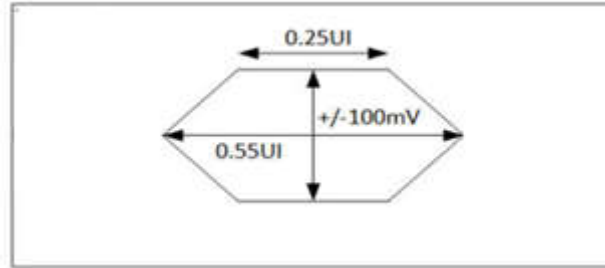


图 1-13. SGMII 眼图模板输入要求

节 1.6.1 是根据输入要求测得的样本眼图，其中 y 轴以毫伏为单位测量，x 轴以 ns 为单位测量。由于模板适合此眼图，因此模板满足 SGMII 输入要求。

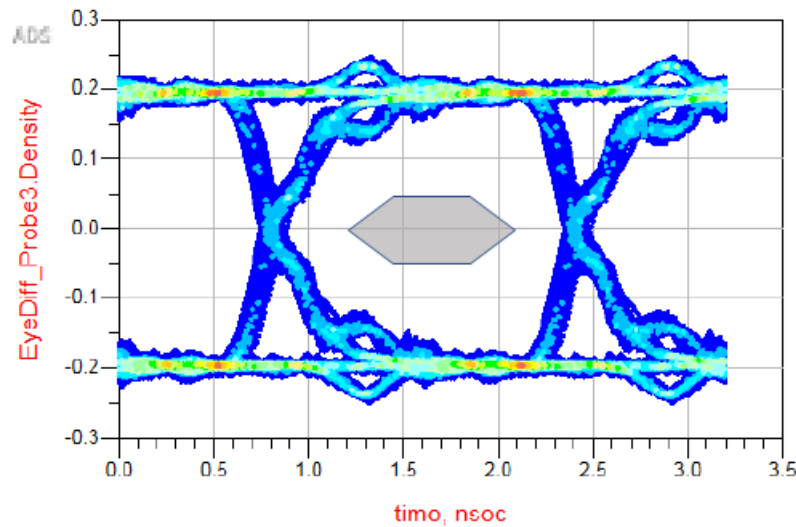


图 1-14. SGMII 示例仿真

如果 PHY SGMII 输入不满足上述要求，则布局可能会存在一些问题。有关更多详细信息，请参阅节 1.7。PHY SGMII 输出也可以通过 VOD 摆幅寄存器进行调整。

例如，可以通过切换 DP83TG720S-Q1 中寄存器 0x428 的位 13 和 14 来更改这些设置。

表 1-5. DP83TG720S-Q1 : 428h 输出摆幅设置

位 14-13	寄存器值	SGMII 输出摆幅	单位
00b	0x0002	1000	mV
01b	0x2002	1260	mV
10b	0x4002	900	mV
11b	0x6002	720	mV

1.7 SGMII 布局

数据包丢失、错误和链路断开的另一个原因是 SGMII 布线的长度不匹配和不连续性。布线需要具有尽可能少的弯曲，RX 布线需要与 RX 布线长度一致，TX 布线需要与 TX 布线长度一致，以确保每次系统上电时正确传输数据并成功建立链路。

此外，确保 100nF 交流耦合电容器已靠近发送器引脚放置。布线上还需要存在 100 Ω 的差分阻抗，并且布线需要小于 6 英寸。

可以在原理图检查清单（可在每个产品页面的“设计工具和仿真”部分找到）中找到完整的布局检查。

2 总结

本应用手册提供了对常见 SGMII 问题进行疑难解答的建议流程。如果在遵循前面列出的指示后 SGMII 问题仍然存在，请在 [TI E2E 设计支持论坛](#) 上联系 TI 代表，获取进一步指导。

3 参考资料

- 德州仪器 (TI)，[DP83TG720 产品文件夹](#)。
- 德州仪器 (TI)，[DP83869 产品文件夹](#)。
- [Cisco SGMII 标准](#)。

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2024，德州仪器 (TI) 公司