

DS280DF810/DS250DF810 在通信系统中的快速编程指南

陈鉴宇

South China Application Team

摘要

随着 5G 通信的到来，高达 25Gbps 的单口速率变得普遍，这对高速数据通过 PCB 走线和连接器传输后的信号完整性提出了挑战。在高速信号设计中，要求使用调理芯片来补偿链路损耗以保证信号质量。TI 的 Retimer 产品 DS280DF810/DS250DF810 是可靠的 25G 高速信号调理芯片。本文介绍这两款芯片在典型通信系统中的快速编程方法，达到提高项目调试效率的目的。

目录

1	引言	3
2	基本编程	3
2.1	Egress 基本编程	4
2.2	Ingress 基本编程.....	5
2.3	读取眼图	5
3	优化编程	6
3.1	Egress 方向 Retimer Rx 的优化(链路 A).....	6
3.2	Egress 方向 Retimer Tx 的优化(链路 B)	7
3.3	Ingress 方向 Retimer Rx 的优化(链路 C)	7
3.4	Ingress 方向 Retimer Tx 的优化(链路 D)	7
4	调试编程	8
4.1	Bypass CDR.....	8
4.2	PRBS 功能.....	8
4.3	交叉和环回.....	11
5	结论	11
6	参考资料	11

Figures

Figure 1.	通信系统	3
------------------	-------------------	----------

Tables

Table 1.	Example of Write Mask	4
Table 2.	Egress Configuration Sequence	4
Table 3.	Reset Channel Register to Default	5
Table 4.	Ingress Configuration Sequence	5
Table 5.	Read Eye Diagram	5
Table 6.	Enable Continuous DFE Adaption	6
Table 7.	Set CTLE Boost Value	6
Table 8.	Set Transmit FIR Filter and VOD	7
Table 9.	Bypass CDR	8

Table 10.	Enable PRBS Generator	9
Table 11.	Disable PRBS Generator	9
Table 12.	Enable PRBS Checker	9
Table 13.	Read PRBS Checker	10
Table 14.	Disable PRBS Checker	10
Table 15.	Configure Cross-Point.....	11

1 引言

典型的通信系统(Figure 1)由交叉盘/业务盘、背板和线卡组成，盘与板卡之间由高速连接器连接。光模块将光纤数据接入线卡，然后通过背板传给交叉盘或业务盘上的网络处理器。处理器处理后的数据再通过背板发给线卡，最后经过光模块发给光纤网络。

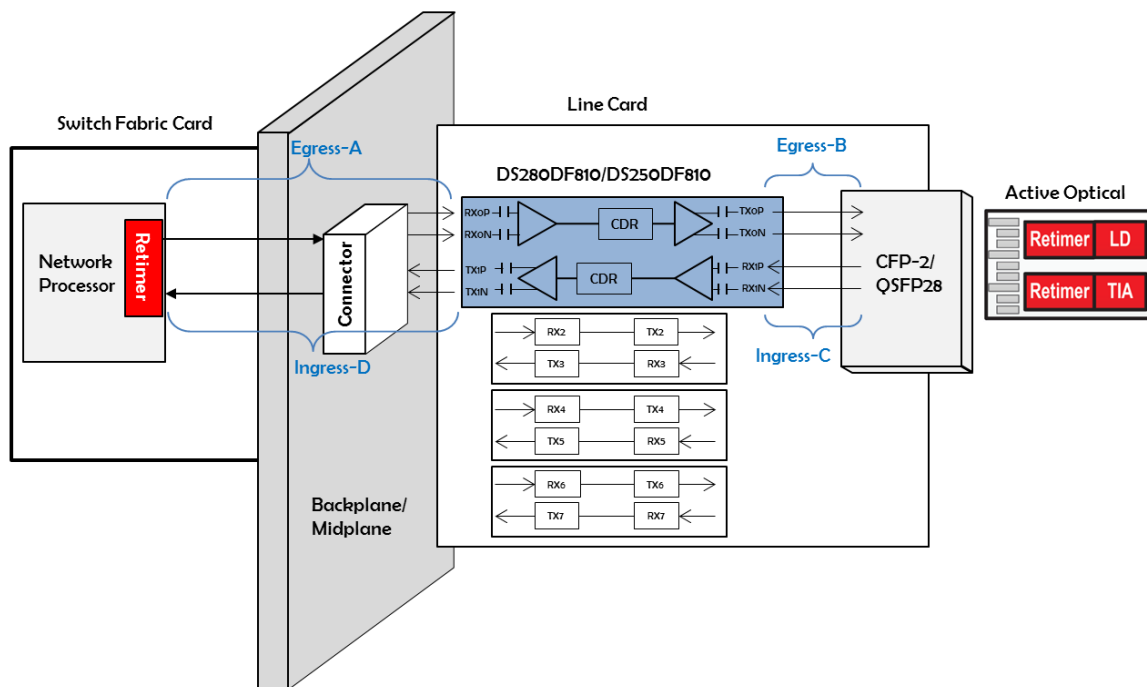


Figure 1. 通信系统

在这个通信系统中，信号走过背板的 PCB 线较长且要经过连接器，故信号的衰减大，需要在收发两侧都有信号调理功能。目前的网络处理器都集成了较强的信号调理功能，但是光模块由于体积和功耗的限制只具有微弱的信号调理能力，所以需要在靠近光模块的一侧加入调理能力出色的 DS280DF810/DF250DF810 来解决信号完整性问题。

2 基本编程

Figure 1 的系统中，从网络处理器经连接器发向光模块为 Egress 方向，反之为 Ingress 方向。从网络处理器到 DS2x0DF810 的链路标注为 A，从 DS2x0DF810 到光模块的链路标注为 B，另外两条链路依次标注为 C 和 D，其中，A 和 D 是最长、损耗最大的两条链路。我们先将 A、B 组成的 Egress 方向和 C、D 组成的 Ingress 方向用基本的编程方法调通，再对四个链路分别进行优化。

需要说明的是，我们使用掩码对 DS2x0DF810 寄存器的每个位(bit)进行操作。掩码为 1 的位表示需要操作，掩码为 0 的位则不需要操作。使用掩码的操作方法满足式 1^[1]。其中 W_{FV} 是操作后寄存器的最终结果值(result)， R_{OV} 是操作之前的初值(default)， W_V 是位写入的操作值(value)， W_M 是位操作的掩码(mask)。

$$W_{FV} = [R_{OV} | (W_V \& W_M)] \& [(W_V \& W_M) | \sim W_M] \quad (1)$$

例如 Table 1, 0x2F 通道寄存器的默认初始值为 0x54, Bit[6:4]是设置传输速率的位, 默认为 3'b101(25.78125Gbps)。如果要将速率配置为 25Gbps, 我们只需要更改 Bit[6:4]=3'b100, 那么这时候 Bit[6:4]的掩码为 3'b111, 而其它不需要操作的 bit 的掩码都为 0, 最终的掩码为 0x70。我们通过掩码 0x70 写入操作值 0x40, 寄存器值变为 0x44。Bit[3]初始值为 1, 由于它的掩码为 0 不需要操作, 故操作结果保持为 1。总之, 只要掩码是 0x70, 即使我们写入的是 0x4F, 低四位也不会改变, 寄存器结果值依然为 0x44。

Table 1. Example of Write Mask

寄存器(register)	初始值(default)	掩码(mask)	操作值(value)	结果(result)
0x2F	0x54	0x70	0x40	0x44

2.1 Egress 基本编程

Figure 1 中, DS2x0DF810 的 Channel 0/2/4/6 用作 Egress 通道。常用的 Egress 基本编程可以按照下面 9 步写寄存器来完成。

Table 2. Egress Configuration Sequence

步骤	寄存器	初始值	操作值	掩码	结果	说明
1	0xFC	0x00	0x55	0xFF	0x55	通过全局寄存器选择通道 0/2/4/6
2	0xFF	0x20	0x01	0xFF	0x01	设置只操作第 1 步选中通道的寄存器
3	0x0A	0x00	0x0C	0x0C	0x0C	将选中的通道置于复位, 通道中的数据传输会中断
4	0x2F	0x54	0x54	0x70	0x54	设置数据速率为 25.78125Gbps
5	0x31	0x20	0x40	0x60	0x40	选择通道 Rx 的接收自适应模式为 Mode 2
6	0x3D	0x1A	0x8A	0xDF	0x8A	设置通道 Tx 的 FIR Main, 输出幅度 VOD 为 637mV
7	0x3E	0x40	0x40	0x4F	0x40	设置 FIR Pre-cursor 为 0
8	0x3F	0x40	0x40	0x4F	0x40	设置 FIR Post-cursor 为 0
9	0x0A	0x00	0x00	0x0C	0x00	释放复位, 通道开始工作, 数据传输恢复

Table 2 中, 前两步都是操作的全局寄存器(global register), 从第 3 步开始操作的是 0xFC 中选中通道的通道寄存器(channel register)。在第 4 步, 我们选择自适应效果最好的 Mode 2。配置生效后, 如果我们想让通道重新自适应一次, 可以先第 3 步再第 9 步, 通过将通道重新复位后再释放来实现。如果寄存器 0xFF=0x03 则之后会忽略 0xFC 的通道选择而同时操作所有 8 个通道。

Table 3 中对 0x00 寄存器操作可以将所有通道寄存器恢复为初始值, 操作后 0x00 寄存器自动恢复为初始值 0x00。该操作一般用于对所有通道寄存器的复位再重写, 建议用在 2、3 步之间。

Table 3. Reset Channel Register to Default

寄存器	初始值	操作值	掩码	结果
0x00	0x00	0x04	0x04	0x00

2.2 Ingress 基本编程

Figure 1 中, DS2x0DF810 的 Channel 1/3/5/7 用作 Ingress 通道。常用的 Ingress 基本编程可以按照下面 9 步写寄存器来完成。

Table 4. Ingress Configuration Sequence

步骤	寄存器	初始值	操作值	掩码	结果	说明
1	0xFC	0x00	0xAA	0xFF	0xAA	通过全局寄存器选择通道 1/3/5/7
2	0xFF	0x20	0x01	0xFF	0x01	设置只操作第 1 步选中通道的寄存器
3	0x0A	0x00	0x0C	0x0C	0x0C	将选中的通道置于复位, 通道中的数据传输会中断
4	0x2F	0x54	0x54	0x70	0x54	设置数据速率为 25.78125Gbps
5	0x31	0x20	0x40	0x60	0x40	选择通道 Rx 的接收自适应模式为 Mode 2
6	0x3D	0x1A	0x8E	0xDF	0x8E	设置通道 Tx 的 FIR Main, 输出幅度 VOD 为 815mV
7	0x3E	0x40	0x41	0x4F	0x41	设置 FIR Pre-cursor 为-1, Pre-de-emphasis 为 2dB
8	0x3F	0x40	0x40	0x4F	0x40	设置 FIR Post-cursor 为 0
9	0x0A	0x0C	0x00	0x0C	0x00	释放复位, 通道开始工作, 数据传输恢复

和 Egress 相比, Ingress 的基本编程只有第 1、6、7 步不同, 即首先选择对应的通道, 然后配置合适的输出幅度。由于链路 D 比链路 B 的链路长、损耗大, 所以 Egress 输出的电压幅度需要更大, 同时还需要加入适当的去加重(de-emphasis)。

2.3 读取眼图

通过 2.1 和 2.2 章节的配置后, Retimer DS2x0DF810 通常能较好的工作, 数据传输能够正常进行。我们通过读取 Table 5 中的通道寄存器来检查通道的工作状态。需要注意的是, 每次只能读取一个通道的寄存器值, 所以必须先在 0xFC 全局寄存器中选择单个通道。

Table 5. Read Eye Diagram

寄存器	初始值	说明
0x78	0x00	检查 CDR 是否锁定

0x27	0x00	检查眼宽(HEO)
0x28	0x00	检查眼高(VEO)

0x78 寄存器的值为 0x30(有时候为 0x38)表示 CDR 已经锁定。0x27 和 0x28 寄存器分别为 DS2x0DF810 测量的眼宽和眼高值。可以通过下面公式^[2]计算出实际的眼图情况。

$$HEO = Reg_0x27 \div 32 [UI] \quad (2)$$

$$VEO = Reg_0x28 \times 3.125 [mV] \quad (3)$$

3 优化编程

我们对 A、B、C、D 四条链路分别进行进一步优化，使整个系统的信号完整性达到最佳。其中，DS2x0DF810 的眼图检测功能可以帮助 A 和 C 两条链路的优化。

3.1 Egress 方向 Retimer Rx 的优化(链路 A)

Rx 的均衡(Equalization)使用自适应模式 2 通常就能获得较好的眼图。一般要求 25Gbps 信号的眼图高于 0.4375UI 和 200mV，最好能够达到 0.5UI 和 250mV。需要注意的是网络处理器 Tx 发出来的信号幅度通常在 800mV 左右再加上一点 Pre-cursor 去加重即可。如果幅度过大会使眼图劣化而产生误码，这是链路 A 常见的问题。

如果链路条件会发生较大变化，例如温度变化，建议打开 DFE 连续适应，可以在 Table 2 的第 5 步后加入下面寄存器操作。其中设置 DFE “look beyond”为 1，表示本次 DFE 调整是根据在此之前的一个 DFE 权重来调试的；设置眼高变化为 15，表示眼高变化 $15 \times 3.125mV = 46.875mV$ 时，开始一次 DFE 自动适应。

Table 6. Enable Continuous DFE Adaption

寄存器	初始值	操作值	掩码	结果	说明
0x7F	0x2A	0x01	0x07	0x29	设置 DFE “look beyond”计数器为 1
0x7D	0x48	0x0F	0x0F	0x4F	设置眼高变化了 15 时，开始一次自适应
0x7F	0x29	0x10	0x10	0x39	打开 DFE 连续适应

将 Rx 的均衡(Equalization)的 CTLE 配置为固定值可以提高对噪声的抗干扰性能。这时候，可以设置 DFE 为自适应模式。配置方法是在 Table 2 的操作之后进行下面操作。

Table 7. Set CTLE Boost Value

步骤	寄存器	初始值	操作值	掩码	结果	说明
10	0x0A	0x00	0x0C	0x0C	0x0C	将选中的通道置于复位，通道中的数据传输会中断
11	0x31	0x40	0x40	0x60	0x40	选择通道 Rx 的接收自适应模式为 Mode 2

12	0x23	0x40	0x40	0x40	0x40	使能 DFE override
13	0x1E	0XE9	0x02	0x0A	0xE3	使能 DFE(DFE_PD=0), 打开 DFE taps 3-5
14	0x2D	0x30	0x08	0x08	0x38	使能 CTLE override
15	0x03	0x00	0x08	0xFF	0x08	设置 CTLE 的 boost value 为 0-0-2-0
16	0x0A	0x0C	0x00	0x0C	0x00	释放复位, 通道开始工作, 数据传输恢复

Table 7 中, 0x03 寄存器的值是按照 CTLE Boost Table^[1]来配置的。可以将 Table 6 的配置加在 Table 7 中第 11 步后以使能 DFE 连续自适应。

3.2 Egress 方向 Retimer Tx 的优化(链路 B)

链路 B 是由 DS2x0DF810 Tx 发给光模块 Rx, 由于光模块的均衡能力较弱, 所以要求 Retimer 能靠近光模块来减少损耗。根据 IEEE P802.3bm™/D3.3^[3], 光模块的接收幅度不能超过 900mV, 所以我们按照 Table 2 设置 0x3D、3E 和 3F 使 Tx 输出幅度为 637mV。如果将 0x3D 寄存器的值增大, 从 0x8A、0x8B 增加到 0x91, 输出幅度会逐渐增加到 879mV。

3.3 Ingress 方向 Retimer Rx 的优化(链路 C)

由于链路 C 中, 从光模块 Tx 到 Retimer Rx 的线路短、损耗小, Retimer 的均衡使用自适应模式 2 通常就能获得较好的眼图。同样的, 光模块输出 Tx 的信号幅度不能太大, 可以和 3.2 章节的 Tx 幅度一致。Retimer 的 Rx 均衡至少有约 9dB 的增益, 按照应用经验从光模块到 Retimer 的链路需要设计出至少 4dB 的损耗, 这样就不会出现过均衡的问题。

3.4 Ingress 方向 Retimer Tx 的优化(链路 D)

在调试中, 这一路往往是最难调试的, 如果接收器 Rx 能测试眼图则可以减小难度。Retimer Tx 的输出幅度和去加重是通过 0x3D、0x3E 和 0x3F 通道寄存器配置的, 相关位的定义如下。一定要 0x3D[7]=1, 这样 0x3E 和 0x3F 中配置的去加重才能有效。

Table 8. Set Transmit FIR Filter and VOD

寄存器	初始值	符号位	符号定义	数值位	数值定义	函数定义
0x3D	0x1A	0x3D[6]	Bit[6]=0:+	0x3D[4:0]	计算输出幅度和去加重, 需要把数值位转化为十进制并带上符号。	Main
0x3E	0x40	0x3E[6]	Bit[6]=1:-	0x3E[3:0]		Pre
0x3F	0x40	0x3F[6]		0x3F[3:0]		Pst

用 Main、Pre 和 Pst 可以通过下面算式 4 至 9 近似计算出输出幅度和去加重值。由于链路 D 通常和链路 A 长度和损耗一样, 所以在链路 A 调试好后, 可以将网络处理器 Tx 的输出幅度和去加重读取出来, 然后将读出来的值转换成寄存器 0x3D、0x3E 和 0x3F 中的值配置好。

$$Sum = |Main| + |Pre| + |Pst| \quad (4)$$

$$PreNorm = \frac{Pre}{Sum} \quad (5)$$

$$PstNorm = \frac{Pst}{Sum} \quad (6)$$

$$VOD = 0.2089 + 0.0479 \times Sum - 0.0005 \times Sum^2 [V] \quad (7)$$

$$Pst_cursor_deemphasis = 1.046 - 15.167 \times PstNorm - 24.336 \times PstNorm^2 - 79.79 \times PstNorm^3 [dB] \quad (8)$$

$$Pre_cursor_deemphasis = 1.046 - 15.167 \times PreNorm - 24.336 \times PreNorm^2 - 79.79 \times PreNorm^3 [dB] \quad (9)$$

4 调试编程

4.1 Bypass CDR

我们有时候需要 Bypass CDR 来检查没有 CDR 的时候，链路的工作情况。可以在 Table 2 和 3 的第 1、2 两步选择好通道后，加入下面几步配置。

Table 9. Bypass CDR

步骤	寄存器	初始值	操作值	掩码	结果	说明
3	0x00	0x00	0x04	0x04	0x00	复位选择的通道寄存器为初始值
4	0x31	0x20	0x00	0x60	0x00	选择通道 Rx 的接收自适应模式为 Mode 0
5	0x1E	0xE9	0x00	0xE0	0x09	选择 RAW Mode
6	0x2D	0x30	0x08	0x08	0x38	使能 CTLE override
7	0x03	0x00	0x00	0xFF	0x00	设置 CTLE 的 boost value 为 0-0-0-0
8	0x8E	0x00	0x00	0x01	0x00	设置 VGA low gain mode
9	0x13	0xB0	0x20	0x20	0xB0	设置 High EQ gain mode

在 0x03 寄存器中将 boost value 配置为最小的 0-0-0-0 时，Retimer 也有 9.2dB 的 high-frequency boost。Bypass CDR 之后，DS2x0DF810 能够通过其它 CDR 不支持的速率。

4.2 PRBS 功能

DS2x0DF810 每个通道都带 PRBS 发生器和接收器，但是由于功耗的限制，通道 0 至 3 和通道 1 至 7 能开启发生器和接收器的总数分别只能在 2 个以内，否则功耗高、温升大，芯片的性能将受到极大的影响。

发送 PRBS 码还需要 CDR 锁定，PRBS 发送速率与锁定速率一致，所以需要先通过 2 和 3 章节的配置将 Retimer 优化好。然后通过 Table 10 的操作可以打发生器，PRBS31 是最常用的码型。

Table 10. Enable PRBS Generator

步骤	寄存器	初始值	操作值	掩码	结果	说明
1	0x2E	0x00	0x04	0x04	0x04	选择 PRBS23/31/58 或 63
2	0x30	0x00	0x01	0x03	0x01	选择 PRBS31
3	0x1E	0xE9	0x10	0x10	0xF9	打开串行器
4	0x79	0x10	0x00	0x20	0x10	关闭 PRBS 发生器
5	0x79	0x10	0x20	0x20	0x30	打开 PRBS 发生器
6	0x30	0x01	0x00	0x08	0x01	关闭 PRBS 时钟
7	0x30	0x01	0x08	0x08	0x09	打开 PRBS 时钟
8	0xA5	0x20	0x80	0xE0	0x80	使能 Retimer 输出 PRBS

用下面操作可以关闭 PRBS 发生器。

Table 11. Disable PRBS Generator

步骤	寄存器	初始值	操作值	掩码	结果	说明
1	0x1E	0xF9	0x00	0x10	0xE9	关闭串行器
2	0x79	0x30	0x00	0x20	0x10	关闭 PRBS 发生器
3	0x30	0x09	0x00	0x08	0x01	关闭 PRBS 时钟
4	0xA5	0x80	0x20	0xE0	0x20	配置 retimer 输出 CDR 锁定的数据

CDR 锁定后，用下面操作可以打开 PRBS 接收器。其中需要严格遵循第 3 到 6 步对 0x30 寄存器的操作顺序。

Table 12. Enable PRBS Checker

步骤	寄存器	初始值	操作值	掩码	结果	说明
1	0x0D	0x80	0x00	0x80	0x00	打开解串器
2	0x79	0x10	0x40	0x40	0x50	打开 PRBS checker
3	0x30	0x00	0x00	0x08	0x00	关闭 PRBS 时钟

4	0x30	0x00	0x08	0x08	0x08	打开 PRBS 时钟
5	0x30	0x08	0x10	0x10	0x18	复位 PRBS 接收器
6	0x30	0x18	0x00	0x10	0x08	释放 PRBS 接收器
7	0x82	0x00	0x40	0x40	0x40	复位 PRBS 计数器
8	0x82	0x40	0x00	0x40	0x00	释放 PRBS 计数器

打开 PRBS 接收器后，读取下面寄存器可以得到该通道 CDR，眼图和 PRBS 接收的情况。Table 13 所列是能反映通道状态的所有寄存器。

Table 13. Read PRBS Checker

寄存器	初始值	说明
0x78	0x00	检查 CDR 是否锁定
0x27	0x00	检查眼宽(HEO)
0x28	0x00	检查眼高(VEO)
0x01	0x00	读取接收到的码型，0x9A 表示收到 PRBS31
0x83	0x00	0x83 和 0x84 通道寄存器记录误码数量， 0x83[2:0]是高 3 位，0x84[7:0]是低 8 位。
0x84	0x00	

配置寄存器 0x82[7]=1 可以将 PRBS 误码计数器暂停，将 0x82[7]恢复为 0 后，误码计数器继续工作。重复一次 Table 12 中的 7、8 两步可以将计数器清零。如果要使 PRBS 接收器先复位再重新工作，可以重复一遍 Table 12 中第 3 到 8 步的配置。

使用 Table 14 的操作可以关闭 PRBS 接收器。

Table 14. Disable PRBS Checker

步骤	寄存器	初始值	操作值	掩码	结果	说明
1	0x0D	0x00	0x80	0x80	0x80	关闭解串器
2	0x79	0x50	0x00	0x40	0x10	关闭 PRBS checker
3	0x30	0x08	0x00	0x08	0x00	关闭 PRBS 时钟

4.3 交叉和环回

DS2x0DF810 中通道分为 Channel 0 和 1、2 和 3、4 和 5、6 和 7 四组，每组的两个通道之间可以做交叉。如果按 Figure 1 中的系统结构，每组的一个通道为 Egress，另一个通道为 Ingress，则可以用交叉功能实现环回。Table 15 是实现通道 2 和 3 交叉的配置方法。

Table 15. Configure Cross-Point

步骤	寄存器	初始值	操作值	掩码	结果	说明
1	0xFC	0x00	0x04	0xFF	0x04	通过全局寄存器选择通道 2
2	0xFF	0x20	0x01	0xFF	0x01	设置只操作第 1 步选中通道的寄存器
3	0x95	0x08	0x08	0x08	0x08	打开 EQ
4	0x96	0x08	0x06	0x0F	0x06	关闭默认路径，打开交叉路径，使能交叉
5	0x0A	0x00	0x0C	0x0C	0x0C	将选中的通道置于复位
6	0x0A	0x00	0x00	0x0C	0x00	释放复位，通道开始工作。
7	0xFC	0x00	0x08	0xFF	0x08	通过全局寄存器选择通道 3
8	0x95	0x08	0x08	0x08	0x08	打开 EQ
9	0x96	0x08	0x06	0x0F	0x06	关闭默认路径，打开交叉路径，使能交叉
10	0x0A	0x00	0x0C	0x0C	0x0C	将选中的通道置于复位
11	0x0A	0x00	0x00	0x0C	0x00	释放复位，通道开始工作。

5 结论

DS280DF810 和 DS250DF810 不仅能解决高速数据传输中的信号完整性问题，其丰富的功能还能帮助检查和调试数据传输系统的问题。按照第 2 章节的基本编程能使数据链路较好的开始工作。按照第 3 章节的优化编程能提高链路的质量。第 4 章节的调试编程可以帮助检查和调试数据链路的问题。作为性能强大的 Retimer，DS280DF810 和 DS250DF810 已经广泛应用于通道速率高达 28Gbps 的通信系统中。

6 参考资料

1. Texas Instruments, "DS2x0DF810, DS250DFx10, DS250DF230 Programmer's Guide," SNLU182E application note, Dec. 2015[Revised Aug. 2018]
2. Texas Instruments, "DS280DF810 28 Gbps Multi-Rate 8-Channel Retimer," SNLS538 datasheet, Sept. 2016
3. IEEE Draft Standard for Ethernet Amendment: Physical Layer Specifications and Management Parameters for 40 Gb/s and 100 Gb/s Operation Over Fiber Optic Cables," in IEEE P802.3bm/D3.23.3, November 2014 (Amendment of IEEE Std 802.3-2012), vol., no., pp.1-16, 21 Nov. 2014

4. Texas Instruments, “DS250DF810 25 Gbps Multi-Rate 8-Channel Retimer,” SNLS495A datasheet, Sept. 2015[Revised Jan. 2016]

重要声明和免责声明

TI 均以“原样”提供技术性数据（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，不保证其中不含任何瑕疵，且不做任何明示或暗示的担保，包括但不限于对适销性、适合某特定用途或不侵犯任何第三方知识产权的暗示担保。

所述资源可供专业开发人员应用TI 产品进行设计使用。您将对以下行为独自承担全部责任：(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安全、安保或其他要求。所述资源如有变更，恕不另行通知。TI 对您使用所述资源的授权仅限于开发资源所涉及TI 产品的相关应用。除此之外不得复制或展示所述资源，也不提供其它TI 或任何第三方的知识产权授权许可。如因使用所述资源而产生任何索赔、赔偿、成本、损失及债务等，TI 对此概不负责，并且您须赔偿由此对TI 及其代表造成的损害。

TI 所提供产品均受TI 的销售条款 (<http://www.ti.com.cn/zh-cn/legal/termsofsale.html>) 以及ti.com.cn上或随附TI产品提供的其他可适用条款的约束。TI提供所述资源并不扩展或以其他方式更改TI 针对TI 产品所发布的可适用的担保范围或担保免责声明。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122
Copyright © 2019 德州仪器半导体技术（上海）有限公司