

AFE77xx DAC JESD204B 调试

余运涛 Yuntao Yu

China Auto

ABSTRACT

AFE77xx 是一款高性能的 4T4R2F (optional) 收发器。低功耗和高集成度使得 AFE77xx 适用于功率和尺寸受限的 4G 和 5G Massive MIMO 基站。AFE77xx 发射通道采用零中频架构，支持大带宽 600MHz。

AFE77xx Serdes 支持高达 29.49Gbps (8/10b 编码) 或者 24.33Gbps (64/66b 编码) 速率。在采用 AFE77xx 的客户单板，SPI 访问成功以后，即会开始 JESD204B/C 的建链过程。随着 Serdes 速率的提高，JESD204B/C 链路遇到的挑战也会更大。

本文主要介绍在客户 AAS 产品中遇到的发射通道 JESD204B 断链问题的调试过程。

Contents

1	问题介绍	2
2	调试步骤	3
	2.1 Eye margin 正常的情况	3
	2.1.1 Eye margin	3
	2.1.2 单步建链	4
	2.1.3 示波器&PRBS 测试	5
	2.1.4 AFE77xx 内部链路 debug	7
	2.2 Eye margin 很低的情况	8
3	结论	11
4	参考文献	11

Figures

Figure 1.	JESD204B 建链过程	4
Figure 2.	FIFO 示意图	8
Figure 3.	Override SyncB pin 为高电平	10
Figure 4.	SyncB 第二个上升沿	10

Tables

Table 1.	DAC JESD204B Alarm 寄存器读数	2
Table 2.	DAC Serdes RX Eye Margin 寄存器	3
Table 3.	Enable JESD RX PRBS Check	5
Table 4.	Get PRBS Error Count	6

1 问题介绍

在客户测试发射通道 JESD204B 稳定性时，DAC JESD204B 会随机地断链，fail rate 大约为 10%。

注：16pcs AFE77xx /Board，每次重启有任何一片 DAC JESD204B 断链，我们记为 fail 一次。

读 Alarm 寄存器如 table 1:

Table 1. DAC JESD204B Alarm 寄存器读数

Alarm Register	Page	
	0x15 0x02	0x15 0x20
0x0163	0x0	0x0
0x0162	0x0	0x0
0x0161	0x0	0x0
0x0160	0x0	0x0
0x0167	0x7	0x7
0x0166	0x7	0x7
0x0165	0x7	0x7
0x0164	0x7	0x7

根据寄存器 0x167~0x164 各 bit 的含义，可知是 8b/10b error 导致的断链。

Bit7= multiframe alignment error.

Bit6=frame alignment error.

Bit5=link configuration error.

Bit4=elastic buffer overflow (bad RBD value).

Bit3=elastic buffer match error. The first non-/K/ doesn't match 'match_ctrl' and 'match_data' programmed values.

Bit2=code synchronization error.

Bit1=8b/10b not-in-table code error.

Bit0=8b/10b disparity error.

2 调试步骤

2.1 Eye margin 正常的情况

2.1.1 Eye margin

AFE77xx 支持读取内部 eye margin，来判断 AFE Serdes RX 自适应是否正常。

绝大多数情况下，断链时，读 eye margin 也没有问题：4bit MSBs 即 bit [11:8] 为 0x5 或 0x6。

Note: ensure that it is more than 800 (0x320). The read value *0.5 is the approximate Serdes eye height in peak.

Table 2. DAC Serdes RX Eye Margin 寄存器

Lane 0-3	Lane 4-7
write 0015 0x4	write 0015 0x40
read 4661 //Lane 0	read 4261 //Lane 4
read 4661	read 4261
read 4660	read 4260
read 4660	read 4260
read 4461 //Lane 1	read 4061 //Lane 5
read 4461	read 4061
read 4460	read 4060
read 4460	read 4060
read 4061 //Lane 2	read 4461 //Lane 6
read 4061	read 4461
read 4060	read 4460
read 4060	read 4460
read 4261 //Lane 3	read 4661 //Lane 7
read 4261	read 4661
read 4260	read 4660
read 4260	read 4660
write 0015 0x0	write 0015 0x0

寄存器 0x4z61&0x4z60 描述如下：

Table 2-513. Register 4061 Field Descriptions

Bit	Field	Type	Reset	Description
3-0	EYE_MARGIN_VAL[11:8]	R	0h	Eye Margin read back value.

Table 2-512. Register 4060 Field Descriptions

Bit	Field	Type	Reset	Description
7-0	EYE_MARGIN_VAL[7:0]	R	0h	Eye Margin read back value.

下面我们先针对此种场景进行定位。

2.1.2 单步建链

正常情况下，JESD204B 的建链分为三步：Code group synchronization (CGS)，ILAS (Initial Lane Alignment Sequence)和 User data.

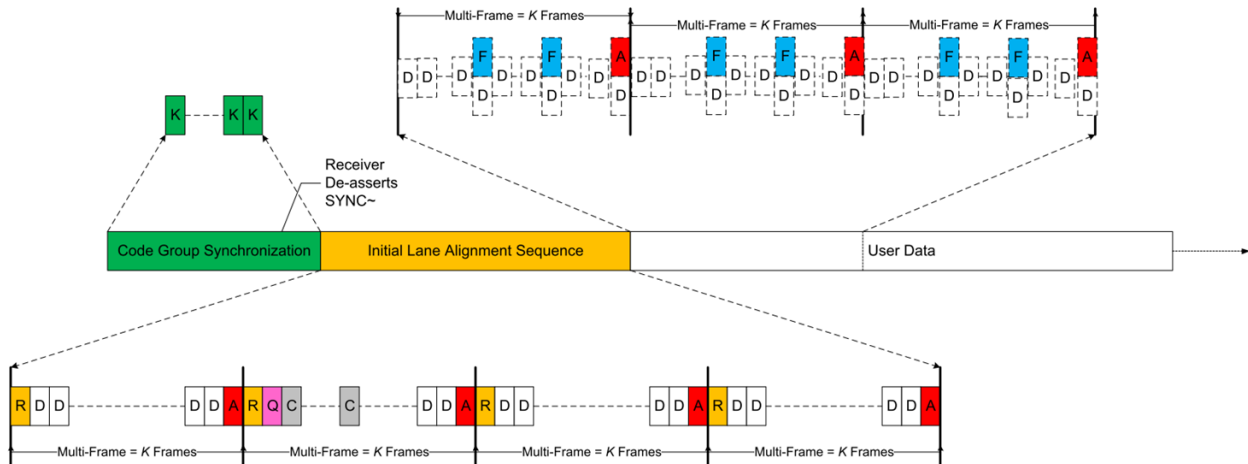


Figure 1. JESD204B 建链过程

我们可以单步执行建链的过程，来判断是哪一步出现了问题。

Override SYNCout pin (of AFE77xx) to low (Receiver de-asserts SYNC~):

寄存器命令如下：

```
// SyncB pin override. In 0x555 first 4 bits for AB side second 4 bits for CD side
```

```
## override sync pin to 0
```

```
write 0x15 0x80
```

```
write 0x555 0xAA
```

```
write 0x15 0x00
```

SyncB 置低以后，理论上，ASIC (JESD TX) 应连续地发送 K28.5，然后 AFE77xx (JESD RX) 进行码组同步。通过读 AFE77xx 寄存器可以判断码组同步是否已经完成。

```
write 0x15 0x02 or 0x20 // Open page: tx_jesd
```

```
read 0x12A // Read jesd_cs_state, expected value=0xAA
```

```
write 0x15 0x0
```

Bit	Field	Type	Reset	Description
7-0	JESD_CS_STATE_TX0	R	0h	JESD CS_STATE value bits(1:0) = lane0 bits(3:2) = lane1 bits(5:4) = lane2 bits(7:6) = lane3

0x12A 是 non-sticky 的，可以实时反应 CS (Code Synchronization) 状态。发现读出的 0x12A 值不断地在变化。

有两种可能原因导致 CS fail:

1. ASIC 没有按照期望连续地发 K28.5.
2. AFE77xx 没有正确地接收到 K28.5，或者状态不稳定。

2.1.3 示波器&PRBS 测试

针对 2.1.2 中的可能原因 1 (ASIC 没有按照期望连续地发 K28.5)，用高速示波器测试 Serdes lane，发现可以解出连续的 K28.5，因此可以排除此疑点。

此外，我们还可以进行 PRBS 15 测试 (AFE77xx 脚本如下)。没有发现误码，也即物理层是没有问题的，这也和示波器测试 Serdes lane 的结果一致。

Table 3. Enable JESD RX PRBS Check

Lane 0-3	Lane 4-7
write 0x15 0x04	write 0x15 0x40
//Lane 0	//Lane 4
write 0x4685 0x00	write 0x4285 0x00
write 0x4684 0x04	write 0x4284 0x04
write 0x4685 0x00	write 0x4285 0x00
write 0x4684 0x06	write 0x4284 0x06
write 0x4685 0x00	write 0x4285 0x00
write 0x4684 0x26	write 0x4284 0x26
write 0x4685 0x00	write 0x4285 0x00
write 0x4684 0x06	write 0x4284 0x06
//Lane 1	//Lane 5
write 0x4485 0x00	write 0x4085 0x00
write 0x4484 0x04	write 0x4084 0x04
write 0x4485 0x00	write 0x4085 0x00
write 0x4484 0x06	write 0x4084 0x06
write 0x4485 0x00	write 0x4085 0x00
write 0x4484 0x26	write 0x4084 0x26
write 0x4485 0x00	write 0x4085 0x00
write 0x4484 0x06	write 0x4084 0x06
//Lane 2	//Lane 6
write 0x4085 0x00	write 0x4485 0x00
write 0x4084 0x04	write 0x4484 0x04

write 0x4085 0x00	write 0x4485 0x00
write 0x4084 0x06	write 0x4484 0x06
write 0x4085 0x00	write 0x4485 0x00
write 0x4084 0x26	write 0x4484 0x26
write 0x4085 0x00	write 0x4485 0x00
write 0x4084 0x06	write 0x4484 0x06
//Lane 3	//Lane 7
write 0x4285 0x00	write 0x4685 0x00
write 0x4284 0x04	write 0x4684 0x04
write 0x4285 0x00	write 0x4685 0x00
write 0x4284 0x06	write 0x4684 0x06
write 0x4285 0x00	write 0x4685 0x00
write 0x4284 0x26	write 0x4684 0x26
write 0x4285 0x00	write 0x4685 0x00
write 0x4284 0x06	write 0x4684 0x06
write 0x15 0x00	write 0x15 0x00

Table 4. Get PRBS Error Count

Lane 0-3	Lane 4-7
write 0x0015 0x04	write 0x0015 0x40
//Lane 0	//Lane 4
write 0x4685 0x00	write 0x4285 0x00
write 0x4684 0x26	write 0x4284 0x26
write 0x4685 0x00	write 0x4285 0x00
write 0x4684 0x06	write 0x4284 0x06
read 0x4699	read 0x4299
read 0x4699	read 0x4299
read 0x4698	read 0x4298
read 0x4698	read 0x4298
read 0x469b	read 0x429b
read 0x469b	read 0x429b
read 0x469a	read 0x429a
read 0x469a	read 0x429a
//Lane 1	//Lane 5
write 0x4485 0x00	write 0x4085 0x00
write 0x4484 0x26	write 0x4084 0x26
write 0x4485 0x00	write 0x4085 0x00
write 0x4484 0x06	write 0x4084 0x06
read 0x4499	read 0x4099
read 0x4499	read 0x4099
read 0x4498	read 0x4098
read 0x4498	read 0x4098
read 0x449b	read 0x409b
read 0x449b	read 0x409b
read 0x449a	read 0x409a
read 0x449a	read 0x409a
//Lane 2	//Lane 6

write 0x4085 0x00	write 0x4485 0x00
write 0x4084 0x26	write 0x4484 0x26
write 0x4085 0x00	write 0x4485 0x00
write 0x4084 0x06	write 0x4484 0x06
read 0x4099	read 0x4499
read 0x4099	read 0x4499
read 0x4098	read 0x4498
read 0x4098	read 0x4498
read 0x409b	read 0x449b
read 0x409b	read 0x449b
read 0x409a	read 0x449a
read 0x409a	read 0x449a
//Lane 3	//Lane 7
write 0x4285 0x00	write 0x4685 0x00
write 0x4284 0x26	write 0x4684 0x26
write 0x4285 0x00	write 0x4685 0x00
write 0x4284 0x06	write 0x4684 0x06
read 0x4299	read 0x4699
read 0x4299	read 0x4699
read 0x4298	read 0x4698
read 0x4298	read 0x4698
read 0x429b	read 0x469b
read 0x429b	read 0x469b
read 0x429a	read 0x469a
read 0x429a	read 0x469a
write 0x0015 0x00	write 0x0015 0x00

后面我们针对可能原因 2 进行定位。

2.1.4 AFE77xx 内部链路 debug

Read FIFO: 针对 2.1.2 中的可能原因 2 (AFE77xx 没有正确地接收到 K28.5, 或者状态不稳定), 参考下面框图, 在 AFE77xx 芯片内部, 处于 Serdes 和 JESD 之间的是 FIFO, 我们可以顺着信号流向, 去检查 FIFO pointer 输入端的情况。

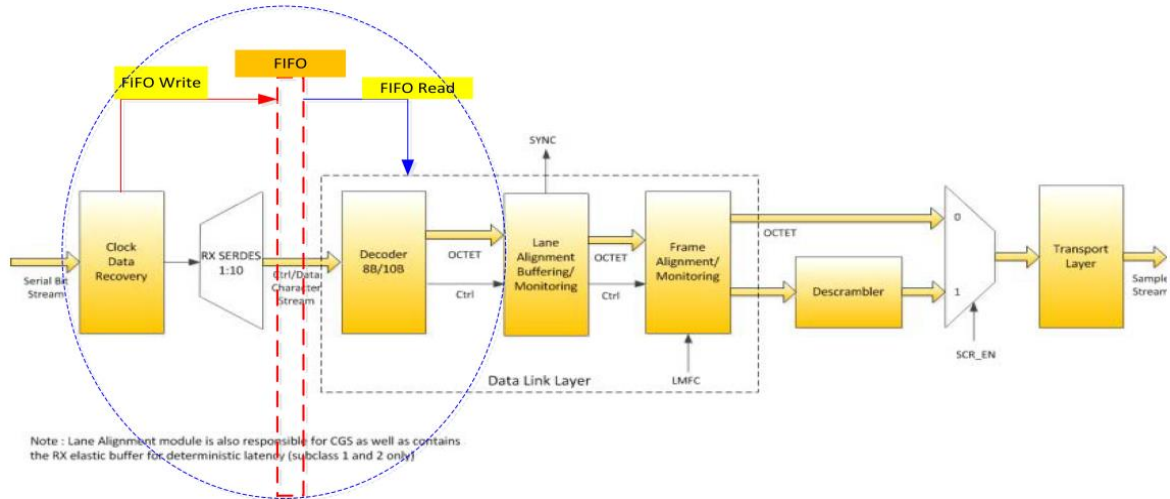


Figure 2. FIFO 示意图

通过采数，可以看到 FIFO 输入端确实是 K28.5. 虽然此测试有其局限性（不能实时地进行观测）。但仍然可以帮助我们缩小范围：AFE77xx 接收到了 K28.5，但是状态不稳定。

我们需要研究为什么 AFE77xx 状态不稳定。根据 JESD 标准，如果有低于一定门限的 error 被 JESD RX（即 AFE77xx）检测到，SyncB 信号上会有窄的脉冲产生。这些窄脉冲可能会使 JESD TX（ASIC）误以为断链（SyncB 为低）。我们可以在 AFE77xx 初始化序列中，配置 JESD 时关闭此功能。

```
write 0x15 0x22
```

```
write 0x8C 0x00
```

```
write 0x15 0x00
```

在 AFE77xx JESD IP 内部，有参数可以设置需要接收的最小数量 K28.5 门限，根据设计，它不需要很高的值。因此我们可以在 AFE77xx 初始化序列中减少该门限值。

```
write 0x15 0x22
```

```
write 0x123 0x08 //change from 0x7f to 0x08.
```

```
write 0x15 0x0
```

在初始化序列中修改了 0x8C 和 0x123 以后，重启 68 次仅有一次断链，断链概率已经大大降低。注意，这一次断链时对应的 eye margin 是不正常的（4bit MSBs 即 bit [11:8] 为 0x0）。至此，我们解决了 eye margin 为正常且 JESD204B 链路不稳定的问题。

下面我们针对 eye margin 不正常（很低）的场景进行分析。

2.2 Eye margin 很低的情况

如果用高速示波器测量 Serdes lane 发现眼图正常，而读取 AFE 内部 eye margin 非常低，首先要怀疑是不是 AFE Serdes RX 适应不成功。

首先，在断链的情况下，复位 AFE77xx Serdes 可以使链路恢复。

```
# Serdes Logic Reset
```

```
write 0x0015 0x44
```

```
write 0x701B 0x07
```

```
write 0x701A 0x77
```

```
write 0x701B 0x00
```

```
write 0x701A 0x00
```

```
write 0x0015 0x00
```

然后，针对复现问题的操作（单板重启，包括 ASIC 和 AFE77xx 的软件），我们现在只 reset AFE（从 HW reset 开始，执行 AFE77xx 初始化序列），reset 350 次，没有复现问题。

我们怀疑问题是由于 AFE77xx Serdes 适应失败导致，可能是因为在 Serdes 适应的过程中，输入数据不够稳定。

因此，我们检查了当前的 JESD204B 建链流程：在单板启动过程中，直到建链之前，没有在 Serdes lane 上面观测到数据。这是由于 ASIC 初始化完成之后，会等到 AFE77xx 的建链阶段才会 enable sysref，开始建链流程。而 AFE77xx 在初始化过程中，在 Serdes 配置之前，就期望可以开始收到随机的数据来做训练和适应。

针对此建链流程，我们做了优化，改变 ASIC 流程，使得 ASIC 初始化完成之后即 enable sysref 开始建链流程，也即可以响应 AFE77xx 的 SyncB out 信号（为低电平，则 ASIC 发 K28.5；为高电平，则 ASIC 发 data），这样在 AFE77xx Serdes 配置中，Serdes logic reset 之前，override SyncB pin 为高电平，以便使 ASIC 开始发送随机数据给 AFE77xx Serdes 做训练和适应。

下图是示波器抓取的该 SyncB 上升沿（绿色曲线），可见在上升沿后，Serdes lane 上开始有 data 出现（浅绿色部分），而在上升沿之前一直在发 K28.5（粉红色部分）。



Figure 3. Override SyncB pin 为高电平

在第二次出现 SyncB 上升沿以后，JESD204B 建链完成。



Figure 4. SyncB 第二个上升沿

按照最新的建链流程，没有再出现断链且 eye margin 很低的现象。

3 结论

本文中，我们从客户测试发射通道 JESD204B 稳定性时，遇到的 DAC JESD204B 随机断链问题出发，针对两个不同的场景：Eye margin 正常，和 Eye margin 很低，分别进行了分析，给出了对应的解决方案。

1. AFE77xx 内部如果读取 eye margin 正常，说明 AFE Serdes RX 适应是没有问题的，需要进一步沿着 AFE77xx 内部链路进行定位。

解决方案：关闭 SyncB 信号上的窄脉冲，并修改 JESD IP 内部参数，减小需要接收的最小数量 K28.5 门限。

2. AFE77xx 内部如果读取 eye margin 很低，要考虑是不是 AFE Serdes RX 适应不成功导致，并研究 JESD204B 建链流程是否可以优化。

解决方案：修改建链流程（包括 JESD TX 和 RX）。

4 参考文献

1. *AFE77xx datasheet (SLASES4A)*
2. *AFE77xx Programming User Guide (SBAU329)*

重要声明和免责声明

TI 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安全、安保或其他要求。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 (<https://www.ti.com.cn/zh-cn/legal/termsofsale.html>) 或 [ti.com.cn](https://www.ti.com.cn) 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。

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 © 2021 德州仪器半导体技术（上海）有限公司