

## TI AFE8092/AFE8030 JESD204C 配置及调试手册 Part B

Zhizhao Niu

North &amp; West China Team

### 摘要

AFE8092, AFE8030 是 TI 的一代高性能、大带宽, 多通道收发器。集成了 8 个射频采样发射通道、8 个射频采样接收通道和 2 个射频采样反馈通道 (AFE8092 中的两个反馈 ADC 是和接收通道复用的, AFE8030 中含有两个独立的反馈 ADC)。AFE80 发射通道在 8 通道模式下最多能够支持 800MHz 带宽, 使其非常适用于多频段的 4G 和 5G 基站。800MHz 大带宽需要 260Gbps 的串行数据总量在 ASIC/FPGA 和 AFE80 之间交互。为了能够支持如此高的数据总量, AFE80 集成了 8 对 Serdes 收发通道, 并且支持 JESD204C 协议, 每条 lane 的最大速率能够达到 32.5Gbps。JESD204B/C 接口作为系统中速率最高的总线之一, 往往在产品研发调试阶段需要花费大量的时间去进行配置和调试。本系列文章分为三个部分, Part A 将以 JESD204C 为例介绍 AFE80 JESD204C 模块的组成以及用户将如何进行系统参数配置。Part B 中将详细介绍 AFE80 JESD 建链流程, 常见的告警以及解决方法, Part C 将介绍两种优化 JESD 链路稳定性的方法。

### 目录

1	引言 .....	2
2	AFE80 JESD204C 建链流程 .....	2
	2.1 建链前准备工作 .....	2
	2.2 JESD204C 建链流程 .....	3
3	常见的 JESD204C 告警 .....	4
	3.1 检查告警类型 .....	4
	3.2 回读 JESD204C 告警类型 .....	4
	3.3 检查 JESD204C 状态机状态 .....	5
	3.4 JESD204C 告警解决方法 .....	6
	3.4.1 Serdes loss of signal(LOS) indicator .....	6
	3.4.2 SerDes PLL Unlock .....	6
	3.4.3 Serdes-FIFO Error .....	7
	3.4.4 Sync-header invalid error .....	8
	3.4.5 EoEMB/EoMB Alignment Error .....	9
	3.4.6 Sync-header invalid error .....	9
	3.4.7 Extended Multiblock Alignment Error .....	9
	3.4.8 Elastic buffer overflow (bad RBD value) .....	10
4	结论 .....	10
5	参考文献 .....	10
图	Figure 1. AFE80 Serdes FIFO 框图 .....	7

## 1 引言

在本系列文章“TI AFE8092/AFE8030 JESD204C 配置及调试手册 Part A”中已经详细的介绍了 JESD204 模块的组成以及配置方法。在本文中详细介绍 AFE80 JESD204C 的建链流程，常见 JESD204C 告警类型以及解决方法。根据不同的告警类型罗列出在这个过程中常用到的寄存器序列调试脚本和 CAPI 函数。

## 2 AFE80 JESD204C 建链流程

### 2.1 建链前准备工作

在用户对 AFE80 JESD204C 进行建链前，需要确保系统时钟和 JESD204C 配置设置正确：

- JESD204C 的接收和发送侧的配置：。包括硬件连接和协议配置两个方面。硬件连接包括 ASIC/FPGA 与 AFE80 之间的硬件连接和极性配置。协议配置包括 JESD204C 物理层，链路层和传输层的相关配置，详见应用笔记“TI AFE8092, AFE8030 JESD204 配置及调试手册 Part A”。
- Device clock: Device clock 从 REFCLK pin 脚输入到 AFE80。经过内部的整数 PLL 为 AFE80 提供 DAC 和 ADC 的采样钟。因此 Device clock 频率的选择不仅要考虑能否得到期望的 DAC 和 ADC 采样钟，同时也要考虑 Device clock 的高次谐波是否有可能落入信号带内。在 JESD204C 中，device clock 作为系统时钟需要和 SYSREF clock 保持相位固定。
- SYSREF clock: SYSREF clock 的选择有以下几点限制条件：
  - 1) SYSREF clock 频率最大值：需要能够满足 JESD204C 的要求，SYSREF 的最高频率是选择 JESD204C 的 LEMC 时钟周期。

$$F_{sysref} = \frac{F_{serdes\ lane\ rate}}{66 \times 32 \times E \times N}$$

其中，

$F_{sysref}$ : SYSREF clock 的频率；

$F_{serdes\ lane\ rate}$ : Serdes lane 的速率；

E: JESD204C 扩展多块中的多块数量；

N: 系数，可以取 1, 2, 3 等整数；

- 2) **SYSREF clock 频率最小值:** SYSREF 还用于内部不同模块之间的同步, 所以如果 SYSREF 设置的过低, SYSREF 的频率过低, 此时会增加内部模块的同步和唤醒延迟时间, 从而影响 TDD 切换时系统性能。另外, 如果 SYSREF 频率太低, 也会导致 JESD204C 的 RBD 发生溢出。当我们考虑到以下的场景: SYSREF 的频率非常低, 而 SYSREF 到 ASIC 侧的线路很短, 到 AFE80 侧的线路很长。由于 SYSREF 是用来对齐每一侧的边界时钟以对齐时钟的发出时间, 因此, 低频的 SYSREF 会在 ASIC 和 AFE80 之间产生非常大的数据起始点偏移, 从而导致接收侧 RBD 需要保存长时间的数据, 进而导致 RBD 溢出。
- 3) 通常常见的 Device clock 频率有 245.67MHz, 491.52MHz。常见的 SYSREF 频率有 960KHz, 1.92MHz, 3.84MHz, 7.68MHz。例如, 对于 E=1, serdes 速率为 16.22Gbps 的 JESD204C 链路, SYSREF 时钟可以选择 7.68MHz, 3.84MHz, 1.92MHz 等。

## 2.2 JESD204C 建链流程

建链流程发生在对器件进行静态初始化配置 (hex 文件或 txt 文件) 之后, 在初始化配置中会对 AFE80 进行软件 Reset, 不同模块进行初始化其中对 JESD 模块的初始化包括 ADC\_JESD, DAC\_JESD, Data mux, Lane mux 以及 sync mux 的配置。在配置完成后进行 JESD204C 的建链。建链中的步骤以及对应的 CAPI 函数如下表 1 所示。

**Table. 1 AFE80 JESD204C 建链流程**

Step	Description	Related CAPI
1	确认 Device clock 和 SYSREF clock 已经使能且频率, 相位符合要求。	需要用户自查。
2	ASIC/FPGA 初始化, 配置 ASIC/FPGA Serdes 模块。	需要用户自查。
3	ASIC/FPGA 和 AFE80 使能 Serdes STX 模块输出随机码序列用来为 AFE80 和 ASIC/FPGA 的 SRX 模块进行自适应。	ASIC/FPGA STX 发送随机序列 AFE80 发 PRBS: sendSerdesTxPrbs
4	当 serdes 上数据稳定后, 重置 AFE80 SRX lane, 重新进行参数自适应。	reAdaptSerDesAllLanes
5	AFE80 检查 Serdes lane 的自适应是否成功。	pollSerdesLinkStatusAllLanes
6	AFE80 检查 JESD 模块是否收到 SYSREF 信号。	Checksysref
7	AFE80 重置 JESD 状态机, AFE80 JESD 状态机和 SYSREF 对齐, 等待建链开始。	adcDacJesdLinkRelnit
8	AFE80 清除 JESD TX 路径的数据, 反转 Sync pin 的状态 (204B)。ASIC/FPGA 侧重置 JESD 状态机, 检查 SYSREF 的状态, 开始 ADC 侧建链。	adcDacLinksJesdTxStart ASIC 侧需要用户自查。
9	AFE80 清除 JESD RX 路径的数据, 清除 JESD 告警, 开始 DAC 侧建链。	adcDacLinksJesdRxStart
10	AFE80/ASIC 清除 JESD RX/TX 告警防止误告警。	clearJesdRxAlarms/clearJesdTxAlarms ASIC 侧需要用户自查。
11	AFE80/ASIC 侧检查 JESD RX/TX 告警。	getJesdRxAlarms/getJesdTxFifoErrors ASIC 侧需要用户自查。

### 3 常见的 JESD204C 告警

虽然 AFE80 JESD204C 建链步骤看起来比较简洁，但是由于 JESD204C 数据速率高，在建链过程中接收和发送侧的交互比较多，并且很多状态并不容易用示波器进行检测，因此 JESD204C 的调试过程常常在项目开发阶段占用大量资源。为了方便用户使用，AFE80 Serdes 和 JESD 状态机内部包含了丰富的告警机制和状态检测。用户可以采用 CAPI 或者寄存器的方式回读告警内容，加速调试过程。本章节将按照 JESD204C 的物理层，链路层和传输层，分别介绍不同的告警类型以及解决方法。

#### 3.1 检查告警类型

在 JESD204C 协议中，状态机和告警寄存器包含在 JESD204C 的接收侧，因此下面以下行方向的 JESD204C 为例进行讲解。

**Step1:** 检查告警是不是历史告警。可以首先清除历史 sticky 告警，如果清除告警后仍然存在告警，说明是当前状态下存在告警，否则是初始化过程或者某一时态下存在过历史告警。可以用以下寄存器或者 CAPI 函数清除 DAC\_JESD 告警。注意，因为 AFE80 的寄存器是按照页进行组织的，如果想要访问某一个特定的寄存器，需要先打开这个寄存器所在页。并且在对这个页操作完成后，需要将该页关闭并打开新的页进行其他的操作。因此，请用户在使用 SPI 的方式对寄存器进行读写时，确保软件系统中没有对 AFE80 的其他 SPI 操作。用户也可以使用 CAPI 的方式进行告警清除，CAPI 会在软件进程中加锁，防止被其他进程打断。

<pre>#Step1: clear Alarms by register device.writeReg(0x16,0xC) device.writeReg(0x224,0x05) device.writeReg(0x224,0x00) device.writeReg(0x16,0x00)</pre>	<pre>#Step1: clear Alarms by CAPI clearJesdRxAlarms</pre>
--	---

#### 3.2 回读 JESD204C 告警类型

**Step2:** 在清除告警之后，使用下列寄存器或者 CAPI 去回读告警。如果告警仍然存在，说明确实存在 DAC\_JESD 告警，此时需要进一步分析产生的告警的内容。其中 getJesdRxAlarms 已经包含了其他三个 CAPI。

<pre>#Step2: Read Alarms by register #DAC_JESD_ABCD device.writeReg(0x16,0x4) device.readReg(0x00000215) device.readReg(0x00000214) device.readReg(0x00000217) device.readReg(0x00000216) device.readReg(0x00000219) device.readReg(0x00000218) device.readReg(0x0000021B) device.readReg(0x0000021A) device.writeReg(0x16,0x00)  #DAC_JESD_EFGH device.writeReg(0x16,0x8)</pre>	<pre>#Step2: Read Alarms by CAPI getJesdRxAlarms getJesdRxLaneErrors getJesdRxLaneFifoErrors getJesdRxMiscSerdesErrors</pre>
--	--

device.readReg(0x00000215) device.readReg(0x00000214) device.readReg(0x00000217) device.readReg(0x00000216) device.readReg(0x00000219) device.readReg(0x00000218) device.readReg(0x0000021B) device.readReg(0x0000021A) device.writeReg(0x16,0x00)	
--	--

寄存器 0x214~0x21B 所对应的告警如下表所示。用户可以参考下表去查找具体的告警内容。同时如果系统已经集成了 AFE80 操作所需要的 CAPI，运行 CAPI 将会返回具体的告警内容。

Reg	Reg Name	JESD204C Alarm
0x214	LANE0_ALARMS_TO_PIN[7:0]	[13] = Serdes loss of signal(LOS) indicator [12] = Serdes-FIFO Error (read and write clock frequency mismatch)
0x215	LANE0_ALARMS_TO_PIN[15:8]	[11] = DeMapper Alignment to LMFC/LEMC Error Each of the bits mentioned below are lane errors which are mapped to :
0x216	LANE1_ALARMS_TO_PIN[7:0]	[10]= JESDC: UNUSED
0x217	LANE1_ALARMS_TO_PIN[15:8]	[9] = JESDC: UNUSED [8] = JESDC: Fixed Ones error in sync-header [7] = JESDC: EoEMB alignment error
0x218	LANE2_ALARMS_TO_PIN[7:0]	[6] = JESDC: EoMB alignment error [5] = JESDC: cmd-data in crc mode not matching with spi register bits
0x219	LANE2_ALARMS_TO_PIN[15:8]	[4] = JESDC: elastic buffer overflow (bad RBD value) [3] = JESDC: FEC syndrome error (Either 26-bit FEC or 32 blocks of data is corrupt).
0x21A	LANE3_ALARMS_TO_PIN[7:0]	[2] = JESDC: extended multiblock alignment error [1] = JESDC: sync-header invalid error ('11' or '00' received in expected sync header location)
0x21B	LANE3_ALARMS_TO_PIN[15:8]	[0] = JESDC: CRC error or FEC Uncorrectable error

由于 JESD204C 告警和多个因素有关，还需要结合 JESD204C 的状态机的状态进一步定位问题所在。

### 3.3 检查 JESD204C 状态机状态

JESD204C 建链可以分为三个阶段，Sync Header Alignment，Extended multiblock synchronization 以及 Extended multiblock alignment。AFE80 内部也含有回读 JESD204C 状态机的寄存器。寄存器和 JESD204C 状态机的状态对应关系如下表所示，其中 0x19C 和 0x19E 所对应的状态和 JESD204C 协议中的 7.6.1 和 7.6.2 是相同的。同时，AFE80 也支持用 CAPI “getJesdRxLinkStatus” 回读 JESD 状态机。需要注意的是 JESD 状态机通常存在于 JESD 接收侧，上行链路的状态需要到 ASIC 或 FPGA 中查看。

JESD204C State	Page	0x16 0x04/0x08	0x04 page for DAC_JESD_ABCD lane0~lane3 0x08 page for DAC_JESD_EFGH lane4~lane7
Sync Header Alignment	JESD_SH_STATE	0x19C	bits(1:0) = lane0/4 bits(3:2) = lane1/5 bits(5:4) = lane2/6 bits(7:6) = lane3/7. 2b'10: JESD_SH_STATE=SH_Lock 2b'01: JESD_SH_STATE=SH_HUNT 2b'00: JESD_SH_STATE=SH_INIT
Extended multiblock alignment	JESD_CS_STATE	0x19E	bits(1:0) = lane0/4 bits(3:2) = lane1/5 bits(5:4) = lane2/6 bits(7:6) = lane3/7. 2b'10: JESD_CS_STATE=EMB_Lock 2b'01: JESD_CS_STATE=EMB_HUNT 2b'00: JESD_CS_STATE=EMB_INIT
Current Elastic Buffer State	JESD_BUF_STATE	0x1A2	Elastic Buffer current state (non-sticky). This denotes if the elastic buffer is released and the state machine moved into data state. bits(1:0) = lane0/4 bits(3:2) = lane1/5 bits(5:4) = lane2/6 bits(7:6) = lane3/7. 2b'00: Waiting for the state machine to move into data state. 2b'11: State Machine in data state
CAPI	getJesdRxLinkStatus()		Return Value is 4 bits. 2 bits for top 4 lanes and 2 bits for bottom 4 lanes. =0: Idle state. No change in state. =1: Header Aligned but EoEMB lock yet to happen. =2: Link is up.

### 3.4 JESD204C 告警解决方法

经过前文的回读可以得知目前 JESD204C 状态机所处的状态以及告警的内容。下面可以结合两者提供的信息进行调试。

#### 3.4.1 Serdes loss of signal(LOS) indicator

这个告警是存在于每一个 SRX Lane 中的。当 SRX 端口发现输入信号幅度很小或者处于电气 Idle 的状态时会上报此告警。用户可以通过下面的 CAPI 函数回读 LOS 发生在哪条 Lane 中。

- getJesdRxMiscSerdesErrors()

#### 3.4.2 SerDes PLL Unlock

AFE80 内部除了 Data converter PLL 外，在 Serdes 模块中还有 Serdes PLL 为 Serdes 模块提供所需的时钟信号。如果 Serdes PLL 失锁会出现此告警。可以用下面的 CAPI 函数去检查哪个 Serdes 模块的 Serdes PLL 失锁：

```
getJesdRxMiscSerdesErrors()
```

而失锁的原因可能有：

- 1) Serdes Lane 没有输入信号或信号质量过差，导致 CDR 状态异常。

- 2) AFE80 REFCLK pin 上的时钟信号异常导致 PLL 失锁或者相位漂移。
- 3) ASIC 侧的时钟信号异常。

### 3.4.3 Serdes-FIFO Error

AFE80 JESD204C 支持 64/66B 和 64/80B 两种编码格式，不同的编码格式，Serdes 模块运行的速率不同。在下行链路方向，来自 ASIC 的数据以串行码流的方式输入 AFE80 SRX，之后在 Serdes 模块内部转换成并行码流传递给 JESD204C 模块。JESD204C 模块处理码流的速度与选择的编码格式相关。由于 Serdes 模块和 JESD204C 模块的时钟域不同，因此在两个模块之间含有 Gear box 和 FIFO 两个模块确保数据能够顺利交换。

Gear Box 模块将 Serdes 的数据速率转换成 JESD204C 模块的速率。随后 FIFO 模块对 Serdes 和 JESD204C 模块之间数据传递做缓冲，防止数据在两个模块之间传递时某一个模块没有准备好数据而导致数据传输失败。当出现数据在两个模块之间传递发生故障时会上报 Serdes FIFO 告警。FIFO 模块所处的位置如图 1 所示。

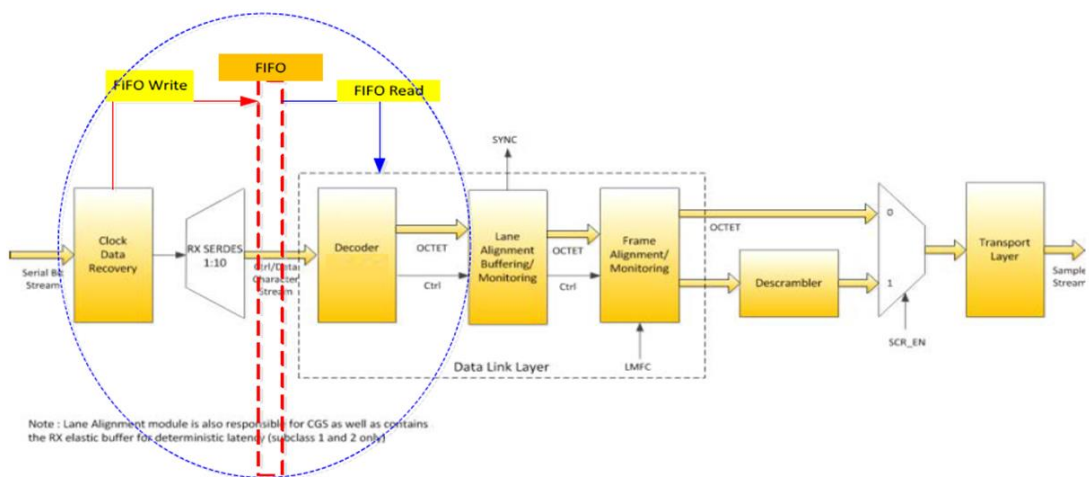


Figure 1. AFE80 Serdes FIFO 框图

Serdes FIFO 有两个关键的参数 FIFO\_WRITE\_CLK 和 FIFO\_READ\_CLK。FIFO\_WRITE\_CLK 表示 Serdes 模块将数据写入 FIFO 模块中的时钟速率，FIFO\_READ\_CLK 表示 JESD204C 模块从 FIFO 中读出数据时的时钟速率。为了 FIFO 的可靠运行需要保证 FIFO 的读写时钟速率相同，并且保证 Serdes 的读写指针（Pointer）相差一个固定的距离不会相撞。这个固定的距离就是数据在两个模块之间传递时的缓冲，用来防止 Serdes 没有准备好数据时，JESD 模块就要取走数据。AFE80 提供了以下寄存器的方式去检查 Serdes FIFO 的读写指针状态，由于 Serdes FIFO 一直在运行，所以建议连续回读 3 次以确保读写 FIFO 的指针之间的距离处于合理的范围内。同时提供了 CAPI 的方式可以修改 FIFO 的偏置值。

Read FIFO Read/Write Pointer: device.writeReg(0x16,0x4)	0x23C: FIFO 写指针，回读范围为 0~15 0x23D: FIFO 读指针，回读范围为 0~15
--	--

<pre>device.writeReg(0x238,0xFF) device.writeReg(0x238,0x0F) device.readReg(0x0000023C) device.readReg(0x0000023D) delay(0.1s) device.writeReg(0x238,0xFF) device.writeReg(0x238,0x0F) device.readReg(0x0000023C) device.readReg(0x0000023D) delay(0.1s) device.writeReg(0x238,0xFF) device.writeReg(0x238,0x0F) device.readReg(0x0000023C) device.readReg(0x0000023D) device.writeReg(0x16,0x0)</pre>	<p>当指针之间的差为：                  0: 读写指针相撞，Serdes FIFO 告警。                  1~5: 读写指针的差距较小需要用 <b>dacJesdSerdesFifoOffset</b> 修改读写指针之间的 offset。                  6~10: 读写指针差距处于合适的范围内。                  *若每次回读指针之间的差变化很大，说明 FIFO 指针不稳定。</p> <p>Read FIFO Read/Write Pointer by CAPI:  <b>getJesdRxLaneFifoErrors</b></p>
--	--

Serdes FIFO 告警通常和时钟频率或 Serdes Lane 速率不匹配相关，因此可以从以下几个方面进行排查：

- JESD204C 接收/发送侧时钟信号是否同源并且相位锁定。
- AFE80 内部的 Serdes PLL 是否有失锁告警。
- JESD204C 的接收/发送侧 Serdes Lane 速率是否匹配。
- AFE80 的 FIFO\_WRITE\_CLK 和 Serdes 模块内的 CDR 相关，而 CDR 是从 Serdes Lane 上的数据中恢复得到的。因此 Serdes lane 的信号质量不好也有可能造成 FIFO Error。可以使用 CAPI “reAdaptSerDesAllLanes” 对 Serdes Lane 进行重新自适应，并回读通过 CAPI “getSerdesRxLaneEyeMarginValue” 去回读 Eye margin 去确保信号质量。如何检查和优化眼高将在后面章节进行详细描述。

### 3.4.4 Sync-header invalid error

当出现 Sync-header error 时，状态机会停在初始阶段，用户也会发现 JESD\_SH\_State 寄存器的回读值不对，CAPI “getJesdRxLinkStatus” 返回 0，这样的情况表示状态机处于 Idle 状态，即没有通过 Sync-Header Alignment。

Sync-Header 由两个 bit 组成，有效的 Sync-header 为 2b'01 或者 2b'10。当两个 bit 为 2b'00 或者 2b'11 时，JESD204C 状态机会将当前的同步头认为无效。当该告警发生时，需要检查 Serdes 眼图是否张开，CTLE 的自适应是否完成以及 Serdes 的极性配置是否正确。可以用下面 CAPI 去检查 Serdes 的极性。

CAPI	系统参数
<b>get_serdesRxLanePolarity()</b> <b>get_serdesTxLanePolarity()</b>	sysParams.serdesRxLanePolarity/ sysParams.serdesTxLanePolarity =[1,1,1,1,1,1,1,1] 1 – 反转极性 0 – 默认极性



### 3.4.5 EoEMB/EoMB Alignment Error

此类告警通常发生在 JESD204C 中的扩展多块对齐阶段，当接收侧状态机发现 Sync Header 组成的 CMD 信号中 bit[22]/bit[27:31]的数据异常时，会发生该告警。当此类告警发生时，需要结合状态机的状态去进一步判断，通常有以下几种情况：

- 当 JESD\_SH\_STATE 没有通过，说明 Sync Header 同步就有问题，需要检查物理层的数据信号完整性。
- 当 JESD\_SH\_STATE 已经通过，说明 JESD204C 物理层没有问题，可能的错误是
  - 1) 没有收到 SYSREF 时钟，可以通过 CAPI checkSysref()进行检查。
  - 2) Sync Header 的极性相反。
  - 3) JESD204C 的参数 E 设置不同导致的。AFE80 的参数 E 可以用以下 CAPI 进行回读或者检查右侧系统参数的设置值。

CAPI	系统参数
<b>get_rxJesdTxK()</b>	sysParams.jesdRxK=[1,1,1,1] sysParams.rxJesdTxK=[1,1,1,1] sysParams.fbJesdTxK=[1,1]

### 3.4.6 Sync-header invalid error

当出现 Sync-header error 时，状态机会停在初始阶段，用户也会发现 JESD\_SH\_State 寄存器的回读值不对，CAPI “getJesdRxLinkStatus” 返回 0，这样的情况表示状态机处于 Idle 状态，即没有通过 Sync-Header Alignment。

Sync-Header 由两个 bit 组成，有效的 Sync-header 为 2b'01 或者 2b'10。当两个 bit 为 2b'00 或者 2b'11 时，JESD204C 状态机会将当前的同步头认为无效。当该告警发生时，需要检查 Serdes 眼图是否张开，CTLE 的自适应是否完成以及 Serdes 的极性配置是否正确。可以用下面 CAPI 去检查 Serdes 的极性。

CAPI	系统参数
<b>get_serdesRxLanePolarity()</b> <b>get_serdesTxLanePolarity()</b>	sysParams.serdesRxLanePolarity/ sysParams.serdesTxLanePolarity =[1,1,1,1,1,1,1,1] 1 – 反转极性 0 – 默认极性

### 3.4.7 Extended Multiblock Alignment Error

当发生该告警时，需要回读状态机状态判断第一阶段 Sync-Header Alignment 是否通过，因为在 JESD204C 中，如果 Serdes 极性相反，Sync Header 仍然可以同步。但是后续状态机在校验 Sync Header 传递的配置信息时会报错。如果第一阶段通过，用户需要去检查 JESD204C 接收发送双方的 E 以及 LMFS 设置是否相同，并且检查 Serdes 的极性是否正确。

### 3.4.8 Elastic buffer overflow (bad RBD value)

当发生该警告时，有以下几种可能：

- 如果多次建链中一直存在，此时说明 RBD 的值不合适，可以通过 Part C 中描述的方法选择最优的 RBD 值。
- 如果偶尔存在，可能的原因时 SYSREF 在多次建链过程中不是确定相位的。从而导致相同的 RBD 值偶尔会告警。或者是帧头差值靠近 RBD 边界，在帧头轻微扰动时，也会导致溢出
- 如果重建链可以解决该告警，可能的原因是在第一次建链时 JESD204C 的发送端没有发出数据让接收端进行自适应。需要用户检查建联流程中是否让 JESD204C 的发送端提前发送数据给接收侧进行自适应。

## 4 结论

本文首先对 AFE80 JESD204C 建链前需要做的准备工作进行了梳理，随后对建链流程进行了总结，并详细介绍了所需 AFE80 的 CAPI 函数。随后介绍了常见 JESD204C 调试的方法，并列出调试过程中需要用到的 CAPI 以及相应的寄存器序列方便用户直接调用。最后结合 CAPI 以及告警寄存器的回读信息，详细介绍了对于不同的 JESD204C 告警，应当如何解决。下一篇文章 Part C 中将详细介绍如何判断 Serdes 眼图的可靠性以及优化 RBD 的方法以提高 JESD204C 链路的稳定性。

## 5 参考文献

1. User Guide “AFE80xx Configuration Guide”
2. 数据手册 “AFE8030 Octal-Channel RF Transceiver with Feedback Paths”

## 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 © 2023，德州仪器 (TI) 公司