

Application Note

调试 AFE7950 的运行时和启动后故障



Dhruvil Solanki and Nikhil Jain

摘要

本应用手册介绍了用于识别、处理和解决 AFE 启动期间所遇错误的系统性流程，进而优化系统的整体效率和可靠性。在 AFE 启动过程中，通过在启动的各个阶段进行读取检查和寄存器轮询来进行全面验证，确认是否满足继续运行的先决条件。此外，轮询寄存器用作确认位，用于指示成功执行，并在启动流程中确保 AFE 功能的完整性。启动故障是由于读取检查失败或轮询失败所致。

此外，本应用手册还针对启动后出现的问题（如串行器/解串器链路或 JESD 链路不稳定、TX 音调或 RX 捕获问题）提供了调试策略。

内容

1 引言.....	2
2 启动期间 SPI 故障.....	2
2.1 有关芯片读数的详细信息.....	2
2.2 芯片读取检查的故障和修复.....	2
2.3 针对 PLL 页面的 SPI 访问进行轮询检查.....	3
2.4 关于 PLL 页面访问 SPI 轮询检查的故障和修复.....	3
2.5 指示熔丝组自动加载状态的读取检查.....	3
2.6 自动加载读取检查的故障和修复.....	3
3 中断启动流程的宏故障.....	4
3.1 针对宏错误的读取检查和针对宏完成的轮询检查.....	4
3.2 宏错误的故障和修复以及宏完成的轮询检查.....	4
4 AFE PLL 故障.....	5
4.1 PLL 锁定的读取检查.....	5
4.2 PLL 读取检查的故障和修复.....	5
5 AFE 内部 Sysref 标志故障.....	6
5.1 sysref 标志位的读取检查状态.....	6
5.2 Sysref 标志位的读取检查状态的故障和修复.....	6
6 JESD 链路检查失败.....	6
6.1 指示 JESD 链路建立状态的多项读取检查.....	6
6.2 JESD 错误的故障和修复.....	8
7 使用 CAPI 验证串行器/解串器和 JESD 链路.....	9
7.1 有用的串行器/解串器调试 CAPI.....	9
7.2 有用的 JESD 调试 CAPI.....	10
8 TX 链验证.....	11
9 RX 链验证.....	11
10 器件运行状况.....	11
11 总结.....	11
12 参考资料.....	11

商标

所有商标均为其各自所有者的财产。

1 引言

AFE 启动涉及一个从上到下的系统性配置过程。为便于在配置文件中拆分步骤，我们将启动文件分为多个步骤。前面配置的部分在后续的调试步骤中起着至关重要的作用。有关下面所述每个步骤的详细信息，请参阅 AFE79xx 配置指南中的启动流程和日志文件部分。

启动流程：

rstDevice、fuseChain、mcuWakeUp、pllEfuse、pllConfig、serdesConfig、topConfig、sysConfig、configTune、analogWrites、jesdConfig、agcConfig、miscConfig、gpioConfig、sysrefJesdLinkup、postLinkUp、dlJesdLinkupCheck

在 AFE 启动过程中，通过在启动的各个阶段进行读取检查和寄存器轮询来进行全面验证。以下是 AFE7950 配置文件中 SPI 命令的格式定义：

SPIWrite Addr, valueToWrite, LSB, MSB：此命令用于针对 AFE 中的 Addr 执行 SPI 写入，addr 最多为 15 位，valueToWrite 值是针对上述 LSB 到 MSB 位要写入的值。

SPIRead Addr, LSB, MSB：此命令用于从上述 addr 中读取所设置 LSB 到 MSB 位的值。

SPIBurstWrite 起始地址, [要写入增量地址的数组]：此命令针对 AFE 进行突发写入，提到了起始地址，并且数组指示要为每个增量地址写入的值。

SPIReadCheck Addr, LSB, MSB, Expectedvalue：读取检查命令验证寄存器的读数是否与预期值匹配。它是一次性检查。读取预期值时出错会导致故障。

SPIPoll Addr, LSB, MSB, Expectedvalue：轮询检查命令在设置的特定时间内反复验证寄存器的读数，直到寄存器读取到预期值、超时或失败（读数与预期不符）为止。

2 启动期间 SPI 故障

2.1 有关芯片读数的详细信息

AFE 有三种不同的芯片标识检查：chip_type、chip_id、chip_ver。在启动生成库时，会提供有关芯片读出结果的信息，读取检查命令会嵌入有关地址、位大小和读出结果的信息。

chip_type = 表示器件型号，# 0xa = AFE

chip_id = 表示芯片 ID，# 0x78 = AFE79xx

chip_ver = 表示芯片版本

SPIReadCheck 0003,0,7,0a //Read	chip_type=0xa;	Address(0x3[7:0])
SPIReadCheck 0004,0,7,78		
SPIReadCheck 0005,0,7,00 //Read	chip_id=0x78;	Address(0x4[7:0],0x5[7:0])
SPIReadCheck 0006,0,7,20 //Read	chip_ver=0x20;	Address(0x6[7:0],0x7[7:0])

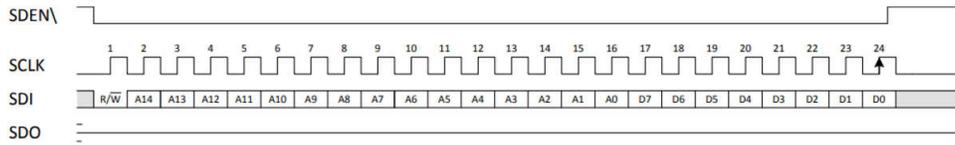
2.2 芯片读取检查的故障和修复

1. 芯片读数为 0x0 或 0xff：

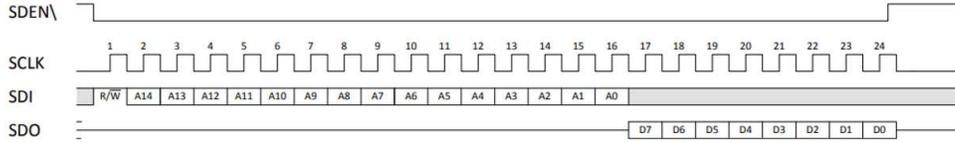
- 我们可以检查 SPI 是否正常运行。确保地址长度为 16，数据包长度为 24，数据包顺序为地址优先，数据包类型为 MSB 优先，启用状态为低电平有效，写入时在正边沿锁存数据，读取时在负边沿锁存数据，还会检查物理 SPI 驱动器连接、以及 AFE 的 SPI 和 GPIO 逻辑电路是否在 1.8V 电压下工作。
- 如果所有设置均符合预期，则下一步是探测 SEN、SCLK、SDI、SDO 并检查波形电平和时序数据，以确保 SPI 驱动器和 AFE 的功能正常。
- SEN、SCLK 和 SDI 的时序和电平是否符合预期，AFE 是否未响应，SDO 是否为低电平。根据系统需要，对 AFE 进行硬件复位，并写入 SPI 以配置 4 线或 3 线模式。

SPIWrite 0000,30,0,7 //位 4 (1 : 4 引脚控制 ; 0 : 3 引脚控制)

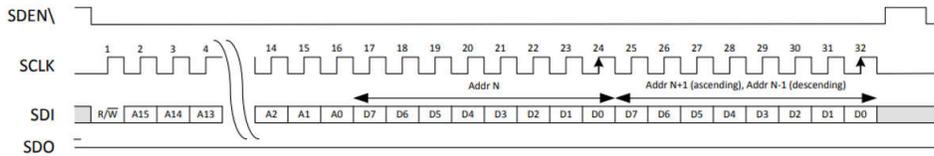
- 检查器件输入电压电平，并检查器件复位状态电流电平。



2. LSB 读数不正确：检查上述 SPI 设置，并检查 SDO、SCLK 和 SEN 的时序。为了进行调试，我们可以单独调用任何芯片读数并探测 SPI 引脚。SEN 需要在最后一个 SCLK 边沿再保持一个额外的时钟周期。



3. 继续执行 AFE 启动过程之前，先验证 SPI 突发写入的实现非常重要。这一点非常重要，因为 AFE 启动的各个方面都使用了 SPI 突发写入，并且实现过程中的任何问题都可能导致日后出现宏错误。某些宏操作涉及加载和验证突发写入操作是否成功。为了进行验证，我们可以使用一个固定总线写入序列，并按顺序读取寄存器。SPIBurstWrite 0010, [01,02,03,04,05,06,07,08,09,0A]，然后读取 0x10 至 0x19 的 SPI 地址，并确认读数是否符合预期。之后将所有地址设置回零。



2.3 对针对 PLL 页面的 SPI 访问进行轮询检查

在 AFE 启动时，如果我们尝试在 PLL 页面中进行写入或读取，则器件需要在内部请求通过 SPI 访问 PLL 页面，并且 SPIPoll 0171,0,0,01 是轮询检查，用于获取有关 SPI 是否已访问 PLL 页面的信息。

2.4 关于 PLL 页面访问 SPI 轮询检查的故障和修复

根据数据表建议，检查 AFE 电源网 (0.925V、1.2V 和 1.8V) 上的电压是否处于预期范围内。

2.5 指示熔丝组自动加载状态的读取检查

完成 EFuse 自动加载，并检查是否存在任何自动加载错误。

熔丝组自动加载检查用于验证熔丝已正确加载。

```
SPIReadCheck 0150,0,3,0f //Read    obs_func_spi_chain_autoload_done=0xf;
SPIReadCheck 0150,4,7,00 //Read    obs_func_spi_chain_autoload_error=0x0;
SPIReadCheck 0160,0,3,0f //Read    obs_func_spi_chain_autoload_done=0xf;
SPIReadCheck 0160,4,7,00 //Read    obs_func_spi_chain_autoload_error=0x0;
```

2.6 自动加载读取检查的故障和修复

自动加载读取检查失败的最常见原因是基准时钟未到达器件。自动加载操作使用的时钟是在 AFE 内部对基准时钟进行分频后得到的。因此，务必检查基准时钟是否在预期电压范围内到达 AFE 引脚，还应检查基准时钟 AFE 侧引脚是否存在 1.2V 共模 (在内部强制执行共模)。

根据数据表建议，检查 AFE 电源网 (0.925V、1.2V 和 1.8V) 上的电压是否处于预期范围内。

3 中断启动流程的宏故障

3.1 针对宏错误的读取检查和针对宏完成的轮询检查

1. 通过 SPI 对器件进行配置，采用直接寄存器读取/写入（对于简单配置）和寄存器写入（用于启动宏）相结合的方式。宏命令将内部器件配置序列抽象为一组简单的配置，简化了主机交互。这些命令将复杂的配置缩减为简单的写入，避免了主机端的计算复杂性，并在响应时提供简单的状态信息。宏命令在启动过程中多处使用。
2. 宏操作码：用于通知 AFE 将要执行操作的操作码。针对 AFE 的差分操作，定义了不同的宏操作码。
宏操作数：操作数是用于在 AFE 中执行操作的值或表达式。
3. 第一个 SPIPoll，对地址 0xf0 中的第 0 位进行轮询，指示宏/MCU 是否准备好进行下一步操作。如果轮询失败，MCU 仍然没有为新操作做好准备，此时如果我们运行新的宏操作，MCU 可能无法执行。
4. 稍后，如果宏就绪检查通过，则加载操作数和操作码。
5. 之后，对宏完成状态进行轮询 #SPIPoll 00f0,2,2,04。这是为了检查宏操作是否完成。
6. 然后读取 0xf0 第 3 位的读取检查，以检查宏操作是正确执行还是出错。
7. 如果任何一项检查失败，启动都会中断。下面将讨论如果发生宏错误和故障，如何找出错误并解决问题。

示例代码

```

SPIPoll 00f0,0,0,01           //MACRO_READY
SPIWrite 00a3,00,0,7         //MACRO_OPERAND_REG
SPIWrite 00a2,00,0,7         //MACRO_OPERAND_REG
SPIWrite 00a1,00,0,7         //MACRO_OPERAND_REG
SPIWrite 00a0,02,0,7         //MACRO_OPERAND_REG
SPIWrite 0193,01,0,7         //MACRO_OPCODE=0x1;
WAIT 0.001
SPIRead 00f0,2,2 //Read     MACRO_DONE=0x1;
SPIPoll 00f0,2,2,04
SPIReadCheck 00f0,3,3,00 //Read     MACRO_ERROR=0x0;
SPIRead 00f1,0,7 //Read     MACRO_ERROR_OPCODE=0x0;
SPIRead 00f0,4,4 //Read     MACRO_ERROR_IN_OPCODE=0x0;
SPIRead 00f0,5,5 //Read     MACRO_ERROR_OPCODE_NOT_ALLOWED=0x0;
SPIRead 00f0,6,6 //Read     MACRO_ERROR_IN_OPERAND=0x0;
SPIRead 00f0,7,7 //Read     MACRO_ERROR_IN_EXECUTION=0x0;
SPIRead 00f3,0,7 //Read     MACRO_ERROR_EXTENDED_CODE=0x0;
SPIRead 00f2,0,7 //Read     MACRO_ERROR_EXTENDED_CODE=0x0;
    
```

3.2 宏错误的故障和修复以及宏完成的轮询检查

1. 宏操作对 AFE 电源网（0.925V、1.2V 和 1.8V）敏感。首先确保所有电压都在相应范围内，并且每条电压轨的电流限制足够，最高可达 3A。结果表明，如果电流源不足，则可能会出现电压骤降，并可能导致宏故障。
2. 要使某项宏操作成功，Sysref 功能必须正常运行。务必要检查 Sysref 电平是否到达数据表规格范围和频率要求的器件引脚。
3. 当故障发生在 (SPIPoll 00f0,0,0,01) 中，即“宏就绪”检查失败时，意味着宏仍然繁忙，无法执行新操作。一旦找到问题所在，确定“宏就绪”的哪个部分出现故障，可以在发生故障的宏就绪命令之前添加一些延迟 (WAIT 1) 并进行检查。如果故障仍然存在，请检查第 1 点。
4. 当故障发生在 (SPIPoll 00f0,2,2,04) 中，即“宏完成”检查失败时，意味着宏仍在进行当前操作。一旦找到问题所在，确定“宏完成”的哪个部分出现故障，可以在发生故障的宏完成命令之前添加一些延迟 (WAIT 1) 并进行检查，如果故障仍然存在，请检查第 1 点。
5. 现在，如果故障发生在 (SPIReadCheck 00f0,3,3,00) 中，则意味着在当前宏操作期间发生了一个宏错误。我们有一个读出寄存器，可以更详细地说明发生了哪种类型的错误，我们可以阅读下面的文章，了解更多有关错误的洞察，如需了解更多细节，可以查看 TRM 文件，同时检查第 1 点。

```

SPIWrite 0018,20,0,7
SPIRead 00f1,0,7 //Read     MACRO_ERROR_OPCODE=0x0;
SPIRead 00f0,4,4 //Read     MACRO_ERROR_IN_OPCODE=0x0;
SPIRead 00f0,5,5 //Read     MACRO_ERROR_OPCODE_NOT_ALLOWED=0x0;
SPIRead 00f0,6,6 //Read     MACRO_ERROR_IN_OPERAND=0x0;
SPIRead 00f0,7,7 //Read     MACRO_ERROR_IN_EXECUTION=0x0;
SPIRead 00f3,0,7 //Read     MACRO_ERROR_EXTENDED_CODE=0x0;
SPIRead 00f2,0,7 //Read     MACRO_ERROR_EXTENDED_CODE=0x0;
SPIRead 00f4,0,7 //Read     MACRO_ERROR_EXTENDED_CODE=0x0;
SPIRead 00f5,0,7 //Read     MACRO_ERROR_EXTENDED_CODE=0x0;
SPIWrite 0018,00,0,7
    
```

6. 0x78 宏操作码的有效执行取决于 SPI 突发写入的精确实现。

4 AFE PLL 故障

4.1 PLL 锁定的读取检查

读取 PLL 锁定状态以检查器件主 PLL 是否已锁定且稳定。

SPIReadCheck 0066,4,4,10	//Lock
SPIReadCheck 0066,6,6,00	//Lock Lost Sticky

我们监控两个特定的位，Bit[4] 表示 PLL 的当前锁定状态，Bit[6] 表示 PLL 首次锁定后是否失去锁定。该位表示 PLL 首次锁定后的不稳定性。

表 4-1. 基准时钟电气特性

f_{PFD}	PFD 频率		100	500	MHz
F_{REF}	输入时钟频率		0.1	12	GHz
V_{SS}	输入时钟电平		0.6	1.8	V_{PPdiff}
耦合			仅交流耦合		
	REFCLK 输入阻抗	并联电阻	100		Ω
		并联电容	0.5		pF

4.2 PLL 读取检查的故障和修复

PLL 读取检查失败的主要错误原因是基准时钟不正确。检查基准时钟的频率是否正确，引脚的功率电平是否在预期范围内，同时检查从器件强制输出的共模电压是否在 1.2V 左右。此外，在位 6 为高电平时检查基准时钟的相位噪声，以及 PLL 1.8V 是否稳定。

5 AFE 内部 Sysref 标志故障

5.1 sysref 标志位的读取检查状态

我们有内部时钟和 sysref 标志状态，用于检查器件是否注册了引脚 sysref。

```
SPIReadCheck 012c,3,3,08 // jesd_clk_rx1
SPIReadCheck 0130,3,3,08 // monitor_jesd_sysref_rx1
```

5.2 Sysref 标志位的读取检查状态的故障和修复

读取检查 sysref 状态失败，说明引脚 sysref 不正确。确保 sysref 频率和电平在预期范围内。如果选择使用脉冲 sysref 模式，则必须通过外部驱动器提供共模电压以及摆幅限制。如果选择了连续的 sysref 模式，则可以使用交流耦合电容器，并且器件可在内部强制将 sysref 引脚的共模电压控制在 0.7V 左右。

表 5-1. Sysref 电气特性

差分输入：±模式 A				
F _{SYSREFMAX}	SYSREF 输入频率最大值		1	GHz
V _{SWINGSRMAX}	SYSREF 输入摆幅最大值		1.8	V _{PPdiff}
V _{SWINGSRMIN}	SYSREF 输入摆幅最小值	f _{REF} < 500MHz	0.3	V _{PPdiff}
		f _{REF} > 500MHz	0.6	V _{PPdiff}
V _{COMSRMAX}	SYSREF 输入共模电压最大值		0.8	V
V _{COMSRMIN}	SYSREF 输入共模电压最小值		0.6	V
Z _T	输入端接	差分	100	Ω
C _L	输入电容	每个引脚端接至 GD	500	pF

6 JESD 链路检查失败

6.1 指示 JESD 链路建立状态的多项读取检查

1. JESD 链路建立在 AFE 启动结束时完成。我们有多项检查点来检查 AFE 的 JESD 链路建立状态。在启动流程中，串行器/解串器链路建立已在 AFE 启动期间完成。请确保 FPGA/ASIC STX 传输一些数据，以便 AFE SRX 的 CDR 可以适应并成功建立 AFE 和 FPGA/ASIC 之间的串行器/解串器链路。

在 JESD 链路建立期间，我们会检查串行器/解串器和 JESD 链路建立状态以及错误状态。下面列出了启动期间完成的读取检查。

a. SPIReadCheck 0118,0,7,00 # 以下是为 0x118 指示的错误的定义

[3:0] = 连接至 0

[4] = JESD 短路测试警报

[5] = 连接至 0

[6] = serdesab_pll_loss_of_lock

[7] = serdescd_pll_loss_of_lock

b. SPIReadCheck 0119,0,7,00

[0] = SRX1 LOS 指示器

[1] = SRX2 LOS 指示器

[2] = SRX3 LOS 指示器

[3] = SRX4 LOS 指示器

[4] = SRX1 串行器/解串器 FIFO 错误

- [5] = SRX2 串行器/解串器 FIFO 错误
- [6] = SRX3 串行器/解串器 FIFO 错误
- [7] = SRX4 串行器/解串器 FIFO 错误
- c. SPIReadCheck 011a,0,7,00
连接至 0
- d. SPIReadCheck 011b,0,7,00
[3:0] = 连接至 0
- [4] = JESDB : Lane0 帧同步错误 (数据中间出现 Ctrl-K) JESDC : Lane0 修复了数字 “1” 错误
- [5] = JESDB : Lane1 帧同步错误 (数据中间出现 Ctrl-K) JESDC : Lane1 修复了数字 “1” 错误
- [6] = JESDB : Lane2 帧同步错误 (数据中间出现 Ctrl-K) JESDC : Lane2 修复了数字 “1” 错误
- [7] = JESDB : Lane3 帧同步错误 (数据中间出现 Ctrl-K) JESDC : Lane3 修复了数字 “1” 错误
- e. SPIReadCheck 011c,0,7,00

以下是通道 0 的 JESD 204B 协议通道错误：

- bit7 = JESDB : 多帧对齐错误
- bit6 = JESDB : 帧对齐错误
- bit5 = JESDB : 链路配置错误
- bit4 = JESDB : 弹性缓冲器溢出 (错误的 RBD 值)
- bit3 = JESDB : 弹性缓冲器匹配错误。第一个非 /K/ 与 “match_ctrl” 和 “match_data” 编程值不匹配
- bit2 = JESDB : 代码同步错误
- bit1 = JESDB : 8b/10b 非表内代码错误
- bit0 = JESDB : 8b/10b 视差错误

如果我们使用 JESD 204C 协议，则以下是通道 0 映射到的通道错误：

- bit7 = JESDC : EoEMB 对齐错误
- bit6 = JESDC : EoMB 对齐错误
- bit5 = JESDC : crc 模式下的 cmd-data 与 SPI 寄存器位不匹配
- bit4 = JESDC : 弹性缓冲器溢出 (错误的 RBD 值)
- bit3 = JESDC : 连接至 0。bit2 = JESDC : 扩展多块对齐错误
- bit1 = JESDC : 同步标头无效错误 (在预期的同步标头位置接收到 “11” 或 “00”)
- bit0 = JESDC : 同步标头 CRC 错误
- f. SPIReadCheck 011e,0,7,00
与上述 JESD 通道 1 的通道错误映射相同
- g. SPIReadCheck 011d,0,7,00
与上述 JESD 通道 2 的通道错误映射相同
- h. SPIReadCheck 011c,0,7,00
与上述 JESD 通道 2 的通道错误映射相同
- i. SPIReadCheck 00ee,0,3,0f
JESDB : comma_align_lock_lane[0:3]monitor_flag
JESDC : sync_header_align_lock_lane[0:3]monitor_flag

- j. SPIReadCheck 00a2,0,7,aa
 JESDB : CS_STATE 值
 JESDC : EMB_STATE 值
 bits(1:0) = Lane0
 bits(3:2) = Lane1
 bits(5:4) = Lane2
 bits(7:6) = Lane3
 对于稳定链路，每个通道启用的位可读取为“10”
- k. SPIReadCheck 00a4,0,7,55
 JESDB : FS_STATE 值
 bits(1:0) = Lane0
 bits(3:2) = Lane1
 bits(5:4) = Lane2
 bits(7:6) = Lane3
 对于稳定链路，每个通道启用的位可读取为“01”
- l. SPIReadCheck 00a6,0,7,FF
 JESDB/C : ELASTIC_BUFFER_STATE 值
 bits(1:0) = Lane0
 bits(3:2) = Lane1
 bits(5:4) = Lane2
 bits(7:6) = Lane3
 对于稳定链路，每个通道启用的位可读取为“11”。

对 JESD 的第二个实例，即 JESD Lane4、5、6、7，进行相同的寄存器读取检查

6.2 JESD 错误的故障和修复

- 如果 0x118 或 0x119 中存在错误，则这是与串行器/解串器相关的错误寄存器，因此检查串行器/解串器的信号完整性非常重要。我们有 CAPI 来检查 SRX 中的 PRBS 测试模式并读取错误计数器。

适用于 SRX PRBS 校验器的 CAPI :

enableSerdesRxPrbsCheck

clearSerdesRxPrbsErrorCounter

getSerdesRxPrbsError

此外，如果您要验证串行器/解串器 PHY 层 AFE 到 FPGA 连接的信号完整性，AFE 使用 CAPI 发送 PRBS 模式。

适用于 STX PRBS 使能的 CAPI :

sendserdesTxPrbs

- 其他寄存器 (0x11b、0x11c、0x11d、0x11e 和 0x11f) 是 JESD 通道错误指示器。错误位和说明不言自明，用以指示 JESD 链路中的问题。下面列出了一些常见错误以及错误设计。

JESD204B

- 使用子类 1，确保 AFE 和 FPGA/ASIC 以确定性方式正确确认 Sysref。必须与器件时钟和 FPGA 基准时钟同步，上升沿转换确定 LMFC 对齐。

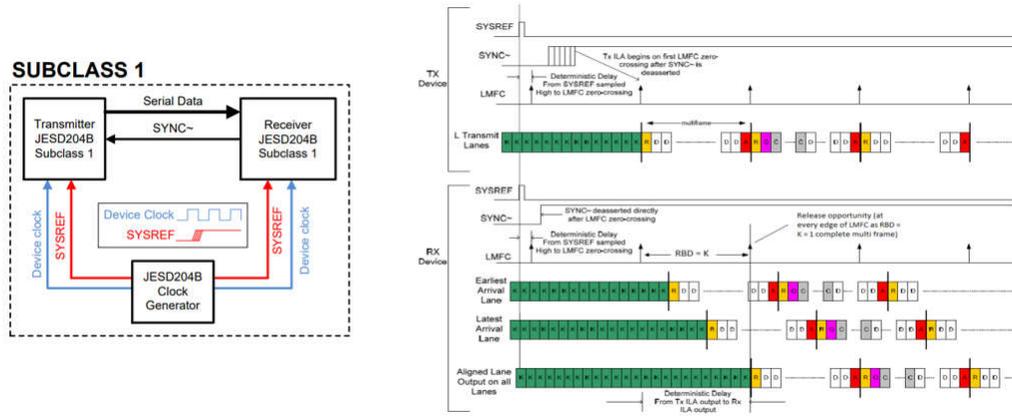


图 6-1. 确定性延迟

备注

有关更多详细信息，请参阅 [了解 JESD204B 子类](#) 和 [确定性延迟](#)。

2. 较低的串行器/解串器眼图裕度会导致问题，请尝试调整 FPGA/ASIC 的 FFE 抽头以提高 AFE 上 SRX 的摆幅。
3. 从 FPGA 发送 K28.5 模式，检查 AFE SYNC PIN 是否正在响应，并检查 CS 状态是否符合预期。
4. 如果我们看到任何与对齐相关的错误，则需要调整 RBD 值。RBD 是一个释放缓冲器空间，用于在调整通道的延迟变化的时间内缓冲数据。请参阅 [确定 JESD204B 和 JESD204C 接收器中的最佳接收缓冲器延迟](#) 应用手册。
5. 正确设置 RBD 后，FS 状态也变为正确，此时链路稳定。
6. 检查串行器/解串器极性。如果串行器/解串器极性反转，可能会进入 CS 状态，但不会进入 FS 和缓冲器状态。
7. 检查 204B 扰频器状态是否匹配 AFE 和 FPGA/ASIC。两者可同时启用，也可同时禁用。

JESD204C

1. 如前所述，可以针对 AFE 和 FPGA/ASIC，同步并以确定性方式应用 sysref。
2. 较低的串行器/解串器眼图裕度会导致问题，请尝试调整 FPGA/ASIC 的 FFE 抽头以提高 AFE 上 SRX 的摆幅。
3. 204C 中出现对齐错误的主要原因是 RBD 大小不正确。因此，请参阅关于如何设置 RBD 的应用手册。（[确定 JESD204B 和 JESD204C 接收器中的最佳接收缓冲器延迟](#)。）
4. 根据采样分辨率，如果分辨率为 16 位，则选择扩展多块 E = 1，如果分辨率为 12/24 位，则选择 E = 3。
5. 为 JESD 接收器和发送器选择相同的 CRC 模式。
6. 检查串行器/解串器极性。如果串行器/解串器极性反转，则不会进入 CS、缓冲器和 FS 状态。

7 使用 CAPI 验证串行器/解串器和 JESD 链路

7.1 有用的串行器/解串器调试 CAPI

以下 CAPI 与 PRBS CAPI 一起用于调试串行器/解串器链路建立。

getSerdesLinkStatus：这将提供有关 SRX 通道（CDR 已锁定状态）串行器/解串器链路状态的动态信息。

getSerdesRxLaneEyeMarginValue：该函数可在处理后获取接收串行器/解串器通道的眼图高度。

reAdaptSerDesAllLanes：这可对所有通道执行逻辑复位和重新调整。

pollSerdesLinkStatusAllLanes：每个通道的逐位链路状态。如果位为 1：通道调整成功。如果位为 0，则不会进行通道恢复。即使已关闭通道，也会返回 1。正常情况下，该值必须为 0xff。

SetSerdesTxCursor：该函数可用于设置 AFE STX 的 FFE 抽头。配置指南中提供了抽头的详细信息。

解决串行器/解串器 PHY 层信号完整性问题可以消除信号丢失和其他与串行器/解串器相关的错误。

7.2 有用的 JESD 调试 CAPI

以下 CAPI 可用于 JESD 调试。

getJesdRxDLaneErrors：用于检查是否存在通道错误。

getJesdRxAlarms：该函数可在完整的 JESD 和串行器/解串器链路中记录错误。

clearJesdRxAlarms：用于清除 JESD 警报，因为 JESD 警报寄存器具有粘滞性。要读取新的错误状态，建议清除后再读取。

getAllLaneReady：该函数读取所有通道就绪计数器，此计数器是内部 LMFC 边界与最后到达通道的多帧边界（在 JESD204B 中）或扩展多块边界（在 JESD204C 中）之间的偏移量。该值具有一定的偏移，在 RBD 中设置。

setManualRbd：用于设置 RBD # 有关如何设置 RBS 的详细信息，请查阅另一份应用手册（[确定 JESD204B 和 JESD204C 接收器中的最佳接收缓冲器延迟](#)）。

adcDacSync：这是重新同步 AFE JESD 块的一个重要函数，在该函数执行期间需要泄漏 `sysref`。它是用户定义函数的一部分。

前面的信息有助于解决启动过程中与 JESD 相关的故障。

在前面的章节中，我们讨论了 AFE 启动过程中可能出现的错误或故障，以及故障的可能原因和解决启动错误的指南。首次捕获数据时，或在启动后出现任何故障时，还有一些调试操作可在启动 AFE 后派上用场。AFE79xx 具有许多可在启动后动态编程的功能，为控制或启用这些功能，我们将它们整合到 CAPI 中。

8 TX 链验证

1. 我们的 `CAPI dacJesdConstantTestPatternValue` 可将 TX 链中输入端的固定 I 和 Q 值发送到 NCO。可以使用 I 和 Q 值控制 NCO 值处的单音调，并控制音调振幅。
2. 该测试有助于在 TX 频谱中出现随机或不正确数据时，检查 TX 链的配置是否正确。在数据不正确的情况下，如果我们启用 `dacJesdConstantTestPatternValue` 并在 NCO 上看到音调输出正确，就可以将问题范围从 TX 链缩小到 TX 链的 JESD 块、串行器/解串器链路或 FPGA/ASIC 的 JESD。
3. 我们可以使用该测试单音来检查 TX 输出功率或校准相当于信号发生器的 RX 链，还可以通过在一定范围内动态配置发 TX NCO 来编程频率。要动态配置 NCO，我们可以使用 `CAPI updateTxNco`。
4. 此外，如果要进行任何杂散调试，可通过使用 `CAPI setTxDsa` 更改 TX DSA，轻松确定耦合点是位于 AFE 内部还是外部。如果杂散电平随 DSA 而变化，则可能是在链内部的某个位置发生了耦合；如果杂散电平不随 TX DSA 变化或以非线性方式扩展，则既可能是外部耦合，也可能是内部和外部耦合两种，并且外部耦合占主导地位。
5. AFE CAPI 库还可使用 `CAPI readTxPower` 在 Tx 链内部读取 dbfs 功率。这在调试测试期间也很方便。

9 RX 链验证

在 RX 链中，我们可以使用测试模式从 JESD 发送斜坡数据以在 FPGA 上进行捕获，而无需使用 `CAPI adcRampTestPattern` 将外部信号发生器连接到 RX。此测试模式对于检查 JESD 捕获功能很有用。

我们使用 `getRxRmsPower` 来测量 RX 链中的 RMS 功率。它可在捕获功能不工作时用于调试，如果需要检查 ADC 是否工作正常，该函数将确认其基本功能。

10 器件运行状况

CAPI 用于检查器件运行状况，`checkDeviceHealth` 表示 PLL、DAC JESD、ADC JESD、SPI、MCU 和 PAP 的状态。

11 总结

本应用手册介绍了导致 AFE7950 启动过程中所遇问题的原因和故障解决分步方法，可作为指导书来解决该启动过程中的各种问题。如上所述，大多数问题都源自外部因素，也可能与电路板有关，因此最好从检查电路板入手。根据问题的性质，需要检查的关键因素可能包括工作电压、SPI 时序、基准时钟电平、`sysref` 共模和电平、串行器/解串器连接映射和极性。

此外，我们还提到了不同的 CAPI，这些 CAPI 可用于分析启动后 TX 和 RX 相关问题，并可动态完成。在 FPGA 或 ASIC 中实施 CAPI 需要板载处理器。我们正在编写有关简化 FPGA 平台上 CAPI 实施的应用手册。因为 CAPI 具有潜在优势，可为了解 AFE 运行提供更多便利。

12 参考资料

1. 德州仪器 (TI)，[JESD204B 概述](#) 高速数据转换器培训。
2. JEDEC 标准：[JESD204C.1](#)。
3. 德州仪器 (TI)，[确定 JESD204B 和 JESD204C 接收器中的最佳接收缓冲器延迟](#) 应用手册。
4. 德州仪器 (TI)，[了解 JESD204B 子类和确定性延迟](#)。

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