

AFE7686 RX DSA 校正简析

Joyce Li

摘 要

AFE7686 是一颗高性能射频采样 DAC 和 ADC 集成的 4T4R 器件。14bit DAC 采样率最高可以达到 9GSPS，14bit ADC 采样率可以达到 3GSPS。AFE7686 还包括 TX 和 RX 的射频前端 DSA 模块，可以给系统带来灵活的 AGC 方案。本文主要介绍 AFE7686 RX DSA 校正的目的、具体实现步骤以及校正前后的增益和相位的对比结果。

目 录

1	引言.....	2
2	AFE76xx 系列 RX DSA 增益和相位校正.....	2
	2.1 RX 链路解析.....	2
	2.2 RX DSA 校正目的以及校正前后增益、相位误差对比.....	3
	2.3 RX DSA 校正步骤.....	4
	2.4 使用校正完生成的 Packet	11
3	总结.....	12
4	参考资料.....	12

图

1	AFE76xx 内部架构图.....	3
2	接收链路 RX DSA	3
3	DSA 校正前后增益误差对比.....	4
4	DSA 校正前后相位误差对比.....	4
5	DSA 校正流程图.....	6
6	GP_INPUT_SIGNAL_REGISTER 寄存器说明.....	8
7	GP_CURR_CALIB_STATUS 报错回读寄存器说明.....	10
8	GP_CALIB_DATA_PACKET_MODE_STATUS 寄存器说明.....	11

1 引言

AFE76xx 是一系列高性能四通道/双通道 14 位集成射频采样模拟前端 (AFE)，配备 4 路 9GSPS DAC 和 4 路 3GSPS ADC，支持数字上变频和数字下变频带，可以直接输出或者输入高达 5.2GHz 的射频信号，相较传统的模拟上下变频链路，AFE76xx 可以为用户节省收发链路器件、布板面积。其高动态范围使得 AFE76xx 能够为无线基站生成和数字化 4G/5G 信号。在 TDD 模式下，ADC 通道经过配置可在接收器 (TDD RX) 状态和宽带反馈接收器 (TDD FB) 状态间动态切换以辅助发射路径上功率放大器 (PA) 的 DPD (数字预失真)。

AFE76xx 内部集成了低抖动 PLL/VCO，外部仅需提供低频率的参考时钟就可以生成 9GHz 采样时钟。AFE76xx 系列拥有 8 对支持 JESD204B 的 SerDes 收发器，运行速率高达 15Gbps。AFE76xx 系列在每个接收链路上集成了 2 个 DSA，每个接收器通道都有一个模拟射频峰值功率测器和多个数字功率测器，可在接收器通道上辅助进行自动增益控制 (AGC)，另有两个射频过载检测器可实现器件可靠性保护。AFE76xx 器件每 TX 通道最多有 2 个 DUC，每 RX 通道有 2 个 DDC，采用多种插值/抽取率以及具有频率灵活的独立 NCO 的数字正交调制器/解调器。这些器件在单频带模式下支持超过 1000MHz (4T4R 下为 800MHz) 射频信号带宽，在双频带模式下支持每频带高达 800MHz (4T4R 下为 300MHz) 射频信号带宽。

2 AFE76xx 系列 RX DSA 增益和相位校正

2.1 RX 链路解析

AFE76xx 内部组成架构图如下图 1 所示，红色框为接收链路架构，图 2 为接收链路 DSA 框图。每个接收链路输入端含两个 DSA，经过 MUX 选择开关以及采样保持 buffer 进入 ADC 采样，在任何时候，仅有一个 DSA 在工作。DSA 具有 28dB 的衰减范围，以 1dB 为步进。最高 3GSPS ADC 由两个 1.5GSPS ADC 交织组成，采样后，进入数字下变频 NCO 链路以及抽取链路，进行降速处理。每个通道最多可以支持两个 NCO，即可以支持双频段。

在 TDD 模式下，接收链路在在 RX、FB 状态间切换，AFE76xx 提供了两种切换方式：

1. RX、FB 链路分别用不同的 DSA。两个 DSA 衰减值独立配置。
2. RX、FB 链路使用相同的 DSA，在切换时，DSA 的值在 FB 和 RX 下动态切换。

从图 2 可知，在 DSA 之前有模拟峰值功率检测点，在 ADC 之后，有数字峰值、均值功率监测点，DSA 可以通过 GPIO 快速配置也可通过 SPI 配置。

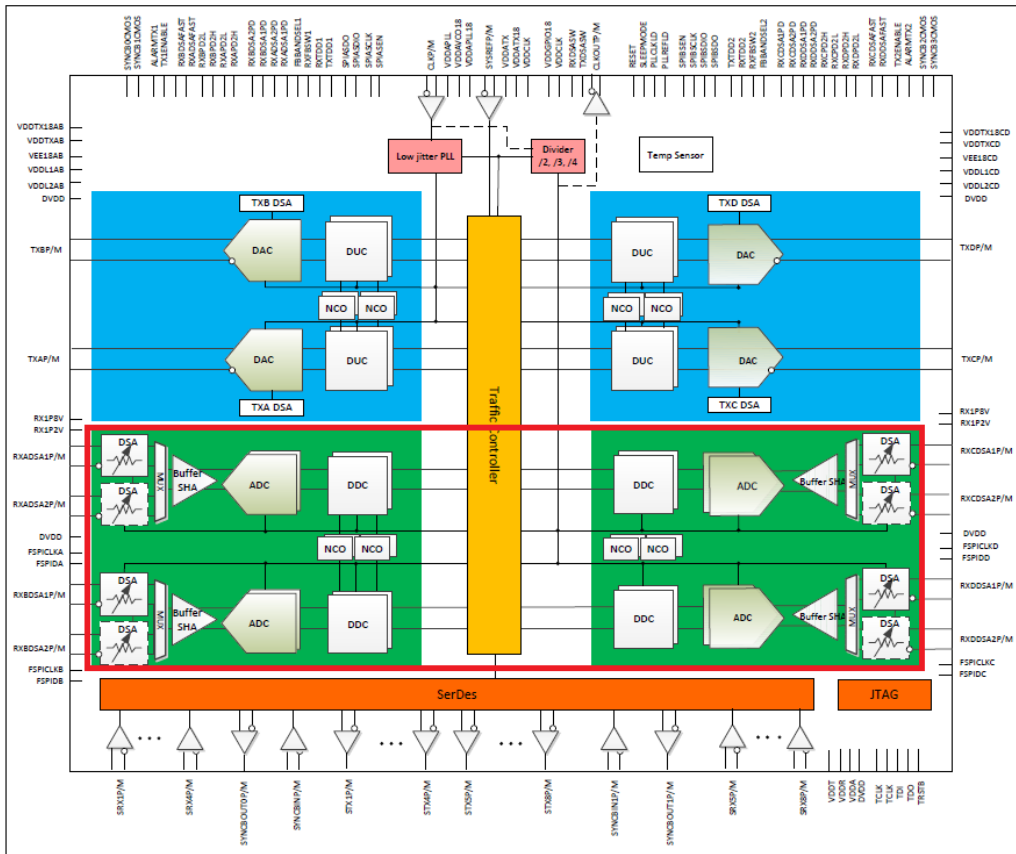


图 1 AFE76xx 内部架构图

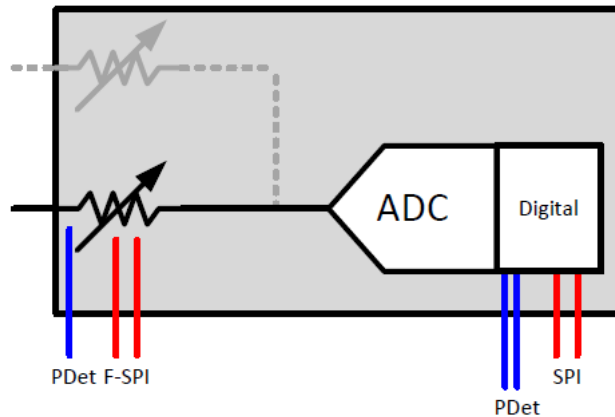


图 2 接收链路 RX DSA

2.2 RX DSA 校正目的以及校正前后增益、相位误差对比

AFE76xx RX DSA 总共有 28dB 的衰减范围，以 1dB 为步进。在未校正前，每个档位实际增益步进会有偏差，当 DSA 衰减值 > 10dB 时，增益误差会明显增大，当用户使用 AGC 功能需要准确调整模拟链路增

益时，需要准确知道每档 DSA 的衰减值，所以需要 DSA 增益校正。图 3 显示了在输入信号在 1800MHz 时，ADC 采样率为 2949.12MSPS，在 DSA 校正前后 DSA 增益误差对比。AFE76xx RX 数据端口是差分输入信号，P、M 信号相位差距 180 度，理想的 DSA 衰减时，不会引起信号的相位变化。未校正时，在 DSA 衰减大于 10dB 时，会明显看到相位变化超过 6 度，影响接收链路的相位，所以需要 DSA 相位校正。图 4 显示了在 DSA 校正前后 DSA 相位误差对比。

DSA- Gain error with and without DSA calibration

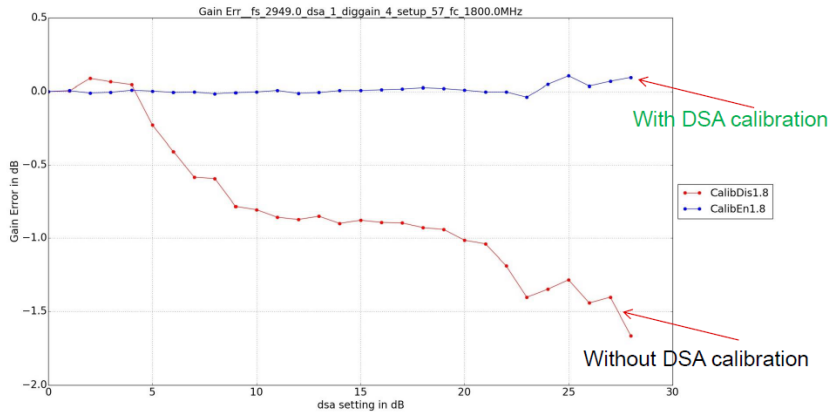


图 3 DSA 校正前后增益误差对比

DSA- Phase error with and without DSA calibration

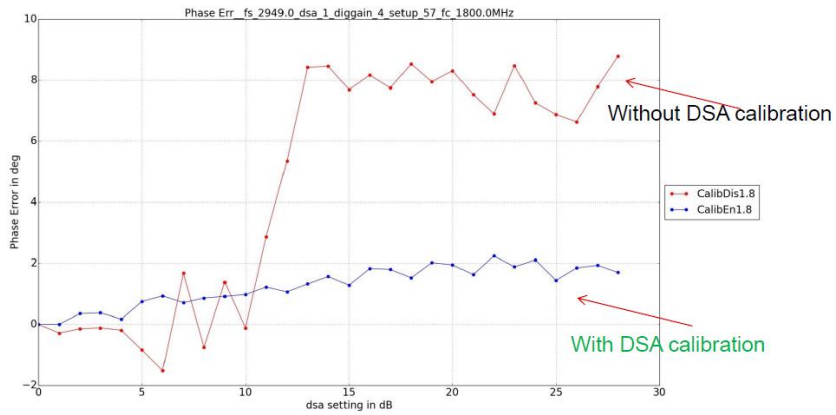


图 4 DSA 校正前后相位误差对比

2.3 RX DSA 校正步骤

由于最终需要校正的是整个 RX 链路上的增益和相位误差，校正也和 AFE76xx 前端射频链路 RF 放大器、PCB 等因素相关，DSA 校正需要用户在整机出厂时校正，校正整个链路上的误差。校正完成后，AFE76xx 会产生一组寄存器配置表，用户只需要在以后 AFE76xx 需要重新上电初始化配置后，将寄存器配置再次写入就可以了。

每个 DSA 都可以校正，需要用户在校正时给 RX 通道灌入射频信号，该射频信号的频率即为待校正频段，幅度要满足在输入信号在 AFE76xx 端口上为-10dBm~-15dBm，如果不能满足该要求，输入幅度更小的话，需要花费更多的校正时间。每个频段可以输入单音或者双音，单音一般位于频段的中心频点，双音则位于频段的两边，可以同时并行输入或者或者串行输入，即为一个频点校正完，再进行下一个频点校正。对于单频段来说，单音输入即可。

校正须知：

1. 外灌的校正信号 F_{in} 必须和 AFE76xx 参考钟同源。否则校正会失败。
2. 灌入的信号 F_{in} 和校正时设置的 NCO 频率 F_c ，以及 F_{bb} 要满足以下关系： $F_{in}=F_c+F_{bb}$ ， $F_{bb}/(\text{ADC output data rate}/1024)$ 必须为整数，这是由于 F_{bb} 需要满足数字 FFT coherent；并且 $F_{in}*1024/F_s$ 必须为整数。这是由于 DSA 校正时的 NCO 并不是正常工作的 NCO，校正 NCO 的位数只有 10 位。所以 $F_{in}*1024/F_s$ 需要为整数。所以 F_{in} 和 F_c 就固定下来了。
3. 在校正前，待校正的 DSA 必须要使能。
4. 每个频段可以校正 1 或者 2 个频点。如果是单音校正，需要将 F_{in} 设置为频段的中心频点；如果是双音校正，则将两个频段设置为频段的始末位置。一般每个频段校正一个音就足够了。
5. sysref 需要被内部屏蔽。
6. 输入信号 F_{in} 的幅度为芯片口-10dBm 到-15dBm

校正流程如下图 5 所示：

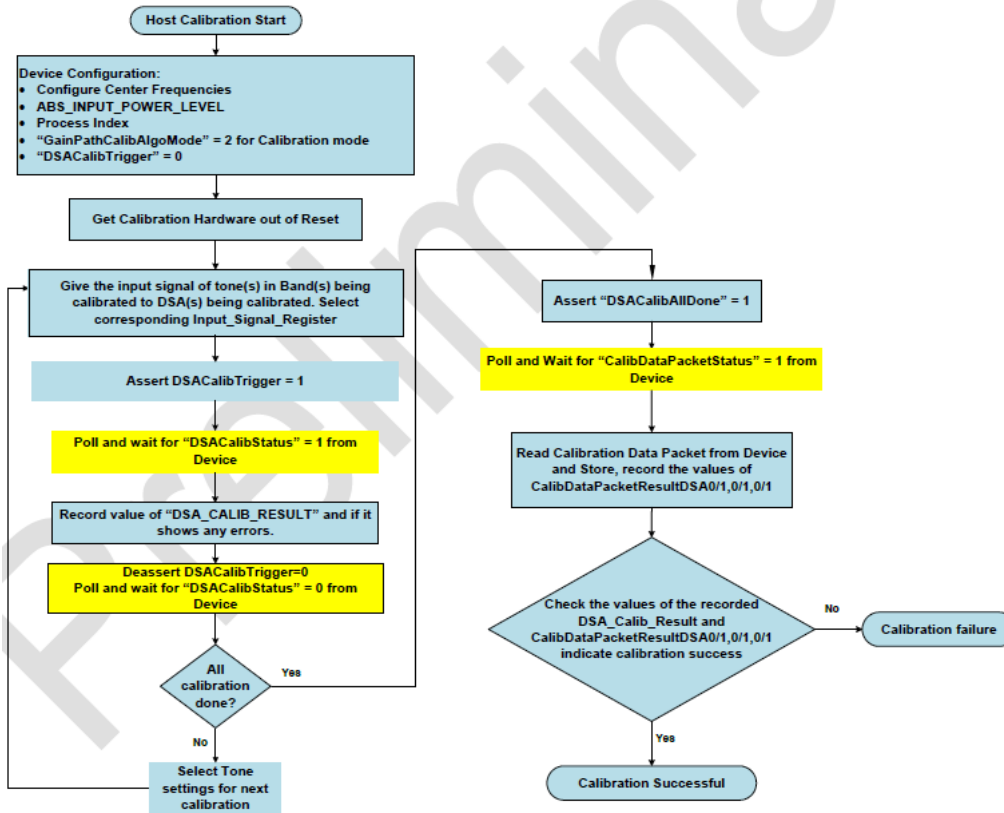


图 5 DSA 校正流程图

校正开始之前，需要确保 AFE76xx 芯片已经完成了初始化配置，校正时配置的 NCO 频率尽量靠近实际使用的 NCO 频率。

Step1: 配置相应寄存器开始校正

```

SPIWrite 0x17 0x0
SPIWrite 0x17 0x0
SPIWrite 0x11 0x80
SPIWrite 0xc4 0x12
SPIWrite 0x11 0x80
SPIWrite 0xc4 0x12
SPIWrite 0x11 0x80
SPIWrite 0xc4 0x2
SPIWrite 0x11 0x0
SPIWrite 0x11 0x80
SPIWrite 0x11 0x80
SPIWrite 0x50 0x0
SPIWrite 0x11 0x0
    
```

Step2: 配置校正时的 NCO Fc，校正时的 Fc 是 10bit NCO，和实际使用的 32bit NCO 不是同一个 NCO。

写入寄存器使用以下公式：Write Value=round((Fc/Fs)*2^10)*2^6

Step3: 设置 process index

```

SPIWrite 0x23 0x5
    
```

Step4: 设置 calibration trigger assert

```

SPIWrite 0x4c 0x0
SPIWrite 0x4c 0x0 # DSACalibTrigger Deassert 即DSACalibtrigger=0.
SPIWrite 0x22 0x2
SPIWrite 0x11 0x0
SPIWrite 0x11 0x0

SPIWrite 0x17 0x20
SPIWrite 0x17 0xa0
SPIWrite 0x20b 0x3
SPIWrite 0x20b 0x3
SPIWrite 0x20b 0x2
SPIWrite 0x20b 0x0
SPIWrite 0x20f 0x70
SPIWrite 0x22b 0x3
SPIWrite 0x22b 0x3
SPIWrite 0x22b 0x2
SPIWrite 0x22b 0x0
SPIWrite 0x22f 0x70
SPIWrite 0x24b 0x3
SPIWrite 0x24b 0x3
SPIWrite 0x24b 0x2
SPIWrite 0x24b 0x0
SPIWrite 0x24f 0x70
SPIWrite 0x26b 0x3
SPIWrite 0x26b 0x3
SPIWrite 0x26b 0x2
SPIWrite 0x26b 0x0
SPIWrite 0x26f 0x70
SPIWrite 0x17 0x80
SPIWrite 0x17 0x0
SPIWrite 0x17 0x0
SPIWrite 0x17 0x0
SPIWrite 0x11 0x80
SPIWrite 0xc4 0x0
SPIWrite 0x11 0x0

```

Step5: 选择 DSA 以及需要校正的频段。

注意可以逐次对每个 DSA 校正、每个 band 校正，也可以同时对多个 DSA，多个 band 校正。如果要同时对多个 DSA 校正，需要给所有要校正的 DSA 端口灌入校正信号 Fin。或者也可以串行校正，一个端口 DSA 校正完成后，再进行下一个端口校正。校正多个频段也是类似的。如果要同时对多个频段校正，需要给 DSA 端口灌入多个频段信号。或者串行方式，校正完 band0，再校正 band1。寄存器 GP_INPUT_SIGNAL_REGISTER 需要按照校正方式进行相应设置。

GP_INPUT_SIGNAL_REGISTER 寄存器说明如下图 6:

GP_INPUT_SIGNAL_REGISTER	0x4A	1:0	DSA000_Signal_Type	Input Signal Type for Unit2Rx0, RX0, DSA0 00 -> No Signal 01 -> Band0 Signal 10 -> Band1 Signal 11 -> Both Band0 and Band1 Signal
		3:2	DSA001_Signal_Type	Same as Above
		5:4	DSA010_Signal_Type	Same as Above
		7:6	DSA011_Signal_Type	Same as Above
		9:8	DSA100_Signal_Type	Same as Above
		11:10	DSA101_Signal_Type	Same as Above
		13:12	DSA110_Signal_Type	Same as Above
		15:14	DSA111_Signal_Type	Same as Above

图 6 GP_INPUT_SIGNAL_REGISTER 寄存器说明

DSAxyz 代表的含义为，x=0 为第一个 2T2R，即 channelA 和 B；x=1 为第二个 2T2R，即 channel C 和 D；y=0/1 代表每个 2T2R 里面的第几个 channel，y=0，则代表是 channel A 或者 C；y=1 代表 channelB 或者 D；z=0/1 代表每个 channel 的第几个 DSA，z=0 代表是第一个 DSA，z=1 代表是第二个 DSA。

例如：需要校正 channel A 第一个 DSA 和 channel D 第二个 DSA，待校正频段 band0 Fc=1899.36MHz，Fin1=1918.08MHz；band1 Fc=2599.52MHz Fin2=2617.92MHz。

Option1: 逐次对 channel 以及 band 校正，需要串行校正 4 次。

1. 校正 channel A 第一个 DSA，即 DSA000，band0，： GP_ISR =0x0001
2. 校正 channel A 第一个 DSA，即 DSA000，band1： GP_ISR =0x0002
3. 校正 channel D 第二个 DSA，即 DSA111，band0： GP_ISR =0x4000
4. 校正 channel D 第二个 DSA，即 DSA111band1： GP_ISR =0x8000

Option2: 同时对 channel 校正，band 逐次校正。需要串行校正 2 次。同时给 channelA 和 D 灌入信号。

1. 同时校正 channel A 第一个 DSA 和 channel D 第二个 DSA，band0： GP_ISR =0x4001
2. 同时校正 channel A 第一个 DSA 和 channel D 第二个 DSA，band1： GP_ISR =0x8002

Option3: 同时对 band 校正，channel 逐次校正。需要串行校正 2 次。同时灌入 band0 和 1 的信号。

1. 校正 channel A 第一个 DSA，同时灌入 band0+band1. GP_ISR =0x0003
2. 校正 channel D 第二个 DSA，同时灌入 band0+band1. GP_ISR =0xC000

Option4: 同时对 band 校正，同时对 channel 校正。需要校正 1 次。同时给 channelA 和 D 灌入 band0 和 1 的信号。

1. 同时校正 channel A 和 D，同时给 A 和 D 灌入 band0+band1。GP_ISR=0xC003

如果用户想对一个 band 信号校正两个频点，也可以逐次进行，也可以同时进行。如果同时进行，需要同时提供两个 tone 的信号；如果逐次进行，需要保证 GP_ISR 一致，提供第一个 tone，校正完成后，再提供第二个 tone，重复 step5.

Step 5.1: 给需要校正的DSA提供input信号

```
SPIWrite 0x11 0x80
SPIWrite 0x11 0x80
SPIWrite 0x4a 0x4 # Program GP_INPUT_SIGNAL_REGISTER LSB
SPIWrite 0x4b 0x0 # Program GP_INPUT_SIGNAL_REGISTER MSB
SPIWrite 0x4e 0x50 # GP_CALIB_INPUT_TONE_PWR_LEVEL
SPIWrite 0x11 0x0
SPIWrite 0x11 0x0
SPIWrite 0x13 0x3
SPIWrite 0xb8 0x2
SPIWrite 0xdb 0x2
SPIWrite 0x13
```

Step 5.2: 等待1s

```
SPIWrite 0x13 0x0
SPIWrite 0x12 0xff
SPIWrite 0x17b 0x1
SPIWrite 0x100 0x81
SPIWrite 0xac 0x21
SPIWrite 0x10b 0xe1
SPIWrite 0xb8 0x21
SPIWrite 0x116 0x81
SPIWrite 0xcc 0x1
SPIWrite 0xd8 0x2
SPIWrite 0x12 0x0
SPIWrite 0x12 0x0
SPIWrite 0x13 0x3
SPIWrite 0xbc 0x1
SPIWrite 0xdf 0x1
SPIWrite 0x13 0x0
SPIWrite 0x13 0x0
SPIWrite 0x11 0x80
SPIWrite 0x11 0x80
SPIWrite 0x4c 0x1 # DSACalibTrigger Assert即DSACalibTrigger=1
SPIWrite 0x11 0x0
```

Step 5.3: 等待校正完成

```
SPIWrite 0x11 0x0
SPIWrite 0x11 0x20
SPIRead 0xbe #轮询该寄存器直到回读为0x1.Poll till this is 1. 表示DSACalibstatus=1.校正完成。
SPIRead 0xbf #校正报错回读。如果校正完成并且回读0xbf=0，代表校正时没有发生错误。SPIWrite 0x11 0x0
SPIWrite 0x11 0x80
SPIWrite 0x11 0x80
SPIWrite 0x4c 0x0 # DSACalibTrigger Deassert
SPIWrite 0x11 0x0
```

报错回读寄存器的定义如图7:

GP_CURR_CALIB_STATUS	0xBF	0	NO_SIGNAL	1 => Signal did not pass the power threshold Signal not connected to right pin or DSA turned off because of improper setting or any OVR or protection.
		1	OUT_OF_BAND_SIGNAL	1 => The signal is not within the band; Incorrect input frequency or center frequency.
		2	UNEXPECTED_MISMATCH	1 => The mismatch is unexpected, This triggers if gain mismatch ≥ 12 dB w.r.t expected attenuation (or) phase mismatch ≥ 90 deg or ≤ -90 deg The mismatch was high than possible for correction. This is possible if the signal sources are asynchronous.
		3	SIGNAL_OVERFLOW	1 => The Adc/Dsa is overflown OVR flag could have triggered it.
		4	Misc Error 1	1 => Possibly too high signal; Reduce the amplitude and rerun.
		6:5	RESERVED	
		7	MISC_ERR	1 => Some misc error has occurred due to which calib has failed

图7 GP_CURR_CALIB_STATUS 报错回读寄存器说明

常见的报错原因有如下几种：

1. No signal. 出现的原因是没有给待校正的 DSA 灌入信号或者信号幅度太小。检查 GP_ISR 设置以及灌入信号的端口是否正确。
2. Out of band signal. 出现的原因是 Fin 和 Fbb 没有在带内。检查 Fin 和 Fnc0 设置是否正确，Fin 和 Fbb 是否满足校正须知第二点的要求，同时检查是否 Fin 和 Fs 同源。
3. Unexpected mismatch. 出现的原因是 Fin 和 Fs 没有同源，导致增益和相位误差较大。
4. Signal overflow. 出现的原因是输入信号幅度过大，减小输入幅度再重新校正。
5. Misc Error 1: 出现的原因是输入信号幅度过大，减小输入幅度再重新校正。

Step 5.4: 等待校正硬件为下一步做好准备

```
SPIWrite 0x11 0x0
SPIWrite 0x11 0x20
SPIRead 0xbe # 等待0xbe回读为0.
SPIWrite 0x11 0x0
```

如果所有的DSA已经校正完成，则继续Step6；否则继续做Step5，直到所有需要校正的DSA完成校正。

Step6: 发起DSACalibALL Done Assert

```
SPIWrite 0x11 0x0
SPIWrite 0x11 0x80
SPIWrite 0x4c 0x2 # DSACalibAllDone Assert
SPIWrite 0x11 0x0
```

Step 6.1: 等待产生校正packet

```
SPIWrite 0x11 0x20
SPIRead 0xbe # 如果回读为0，但是以上的步骤都完成了，等待该寄存器变为1.
SPIWrite 0x11 0x20
```

SPIWrite 0x11 0x0

Step7: 回读校正生成的Packet.

回读GP_CALIB_DATA_PACKET_MODE_STATUS 寄存器，检查是否正确。如果回读不正常，检查以上每一步是否有报错。

以下是实现代码：

Opening page:

SPIWrite 0x11 0x20

Read 0xC2 #回读GP_CALIB_DATA_PACKET_MODE_STATUS校正状态

Closing Page:

SPIWrite 0x11 0x00

GP_CALIB_DATA_PACKET_MODE_STATUS寄存器定义如图8:

GP_CALIB_DATA_PACKET_MODE_STATUS	0xc2	1:0	DSA000CalibDataPacket Result	Calibration Data Packet Status for Unit2Rx0, RX0, DSA0 00 -> Calibration data NOT available 01 -> Calibration data available within range 10 -> Calibration data available with high mismatch 11 -> Calibration data invalid
		3:2	DSA001CalibDataPacket Result	Same as Above
		5:4	DSA010CalibDataPacket Result	Same as Above
		7:6	DSA011CalibDataPacket Result	Same as Above
		9:8	DSA100CalibDataPacket Result	Same as Above
		11:10	DSA101CalibDataPacket Result	Same as Above
		13:12	DSA110CalibDataPacket Result	Same as Above
		15:14	DSA111CalibDataPacket Result	Same as Above

图 8 GP_CALIB_DATA_PACKET_MODE_STATUS 寄存器说明

如果回读正常，进行一下操作：

校正完成后的数据会存在以 0x20 开始的地址，一直存储到以 MAX_Address 地址作为最后一个地址。

完成 DSA 校正后，用户需要回读从 0x20 到 MAX_Address 寄存器的内容，存储到系统内存空间里。以后对 AFE76xx 上下电处理后，再完成初始化配置后，再运行 packet 就可以导入校正后的值。

Max_Address 是和校正的 DSA 数量相关。

Max_Address= 0x41+468*(number of DSAs calibrated for dual band) +260*(number of DSAs calibrated for single band)

以下是实现的代码：

SPIWrite 0x11 0x40

Read the values at addresses from 0x20 to "Max_Address" and store them.

SPIWrite 0x11 0x0

2.4 使用校正完生成的 Packet

AFE76xx 只需在出厂时要完成一次校正，用户需要把校正生成的 packets 保存下来，以后在每次上下电重新初始化配置 AFE76xx 时，将 packet 写入 AFE76xx 就完成了对 DSA 增益和相位的修正。具体操作如下：

在 AFE76xx 初始化配置完成时，写入以下寄存器：

```
SPIWrite 0x11 0x80
```

Write the values of the packet from 0x6000 to the end of the packet incrementing the address by 1 per byte.

```
SPIWrite 0x11 0x0
```

写完所有 packet 里面的寄存器后：将 DSA 配置为 normal mode，写入以下寄存器：

```
// configure DSA Gain/Phase Estimation in Normal Mode, IL engine reset
```

```
SPIWrite 0x11 0x80
```

```
SPIWrite 0x22 0x1
```

```
SPIWrite 0x54 0x44
```

```
SPIWrite 0x5A 0x2
```

```
SPIWrite 0xC4 0x2
```

```
SPIWrite 0xC4 0x0
```

```
SPIWrite 0x11 0x0
```

以上就完成了校正 packet 的导入，AFE76xx 的 DSA 增益和相位就是校正后的性能了。

3 总结

本文主要介绍了 AFE76xx 射频采样 ROC 系列 RX DSA 增益和相位校正的步骤，用户可以利用本文档实现出厂时 RX 链路的 DSA 校正，帮助改善整个 RX 链路的 DSA 增益和相位误差。

4 参考资料

1. AFE7686 datasheet
2. AFE76xx: RX DSA Gain-Phase Calibration

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