

AFE77xx 反馈通道采数

余运涛 Yuntao Yu, 刘先锋 Seasat Liu

NWC AA5

摘要

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 反馈通道 (FB) 采用射频采样架构, 支持大带宽 (600MHz) 以便于做数字预失真 (DPD)。

本文主要介绍 AFE77xx 反馈通道内部抓数的步骤及后处理方法, 以便帮助客户在项目初期可以更快地定位问题。

目录

1	引言.....	1
2	采数步骤.....	2
3	数据处理.....	3
4	总结.....	6
5	参考资料.....	6

图表

图 1.	AFE77xx 反馈通道框图	2
图 2.	原始数据曲线	5
图 3.	频谱图.....	6

1 引言

AFE77xx 可以支持最多两路 FB, FB 是基于直接射频采样架构, 主要是用来接受发射 (TX) 链路的信号, 并用于数字预失真 (DPD)。直接射频采样架构有着大带宽的优势, 而且简化了 TX 链路 QEC (quadrature error correction) 和本振泄露校准。下图是 AFE77xx 反馈通道框图。

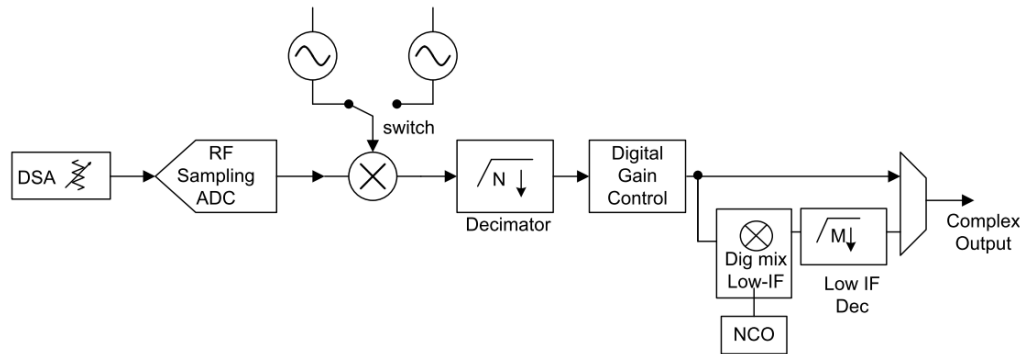


图 1. AFE77xx 反馈通道框图

在项目初期，JESD204B/C 还没有建链成功，或者 ASIC 内部抓数功能还不能用的情况下，射频工程师需要开始前期的反馈通道链路调试，例如验证 FB DSA，NCO 功能及异常 spur 等，就需要借助 AFE77xx 内部抓数功能。

AFE77xx 内部 FB 采数点位于 JESD204B 模块前端，第二章介绍了抓数的步骤。

2 采数步骤

以下为 SPI 读写操作步骤。

Step 1: Trigger the capture.

Override TDD 确保 FB 是打开状态，并可以在此设置 FB DSA 为期望的值。

Step 2: Clear log. 便于下一步保存采数结果。

Step 3: Capture FB data.

最大支持的采数点数为 512 (IQ)，假设 FB data rate=491.52M，Bin=491.52M/512=0.96MHz，即 FFT 最小分辨率 (RBW)=0.96MHz。

由此可以看出，此采数功能主要是用于前期验证链路通断，DSA 及 NCO 功能，而较少用于性能测试。

```
write 0x0013 0x40
read 0x20 to 0x11f    //read_array_i, maximum to 0x41f
read 0x4020 to 0x411f //read_array_i1, maximum to 0x441f
read 0x1020 to 0x111f //read_array_q, maximum to 0x141f
read 0x5020 to 0x511f //read_array_q1, maximum to 0x541f
write 0x0013 0x00
```

Step 4: Save the contents of the log window into a text file.

便于后续数据处理，如原始数据组合，FFT 等。

Step 5: Close the trigger.

Step 6: 针对 step 4 保存的 log，做 FFT，进行后处理。

3 数据处理

以客户环境读取的数据（读数范围从 0x20 to 0x11f (read_array_i), 0x4020 to 0x411f (read_array_i1), 0x1020 to 0x111f (read_array_q), 0x5020 to 0x511f (read_array_q1), 总共 $256 \times 4 = 1024$ 个点）为例，阐述如何组合原始数据。

首先，将 “read_array_i, read_array_i1, read_array_q, read_array_q1” 中，每 4 个组合成一个 32bit 数。

例如，对于 read_array_i, 0x0020 (LSB)-0x0023 (MSB) 组合为 0xffff00ea，作为 data_i[2*i], $i=0,1,2,\dots, 256/4$; 对于 read_array_i1, 0x4020 (LSB)-0x4023 (MSB) 组合为 0xffff013f，作为 data_i[2*i+1]。这样得到了 data_i，共 128 个点。

从 read_array_q& read_array_q1 得到 data_q 的方法类似。

read_array_i		read_array_i1		read_array_q		read_array_q1	
Address	Value	Address	Value	Address	Value	Address	Value
0x0020	0xea	0x4020	0x3f	0x1020	0xff	0x5020	0xf3
0x0021	0xe0	0x4021	0xe1	0x1021	0xfb	0x5021	0xf9
0x0022	0xff	0x4022	0xff	0x1022	0xff	0x5022	0xff
0x0023	0xff	0x4023	0xff	0x1023	0xff	0x5023	0xff
0x0024	0xb5	0x4024	0x53	0x1024	0xf8	0x5024	0x8
0x0025	0xe1	0x4025	0xe2	0x1025	0xf7	0x5025	0xf6
0x0026	0xff	0x4026	0xff	0x1026	0xff	0x5026	0xff
0x0027	0xff	0x4027	0xff	0x1027	0xff	0x5027	0xff
0x0028	0xf7	0x4028	0xb7	0x1028	0x3c	0x5028	0x69
0x0029	0xe2	0x4029	0xe3	0x1029	0xf4	0x5029	0xf2
0x002a	0xff	0x402a	0xff	0x102a	0xff	0x502a	0xff
0x002b	0xff	0x402b	0xff	0x102b	0xff	0x502b	0xff
0x002c	0xa6	0x402c	0xba	0x102c	0xae	0x502c	0x9
0x002d	0xe4	0x402d	0xe5	0x102d	0xf0	0x502d	0xef
0x002e	0xff	0x402e	0xff	0x102e	0xff	0x502e	0xff
0x002f	0xff	0x402f	0xff	0x102f	0xff	0x502f	0xff
0x0030	0xda	0x4030	0x2d	0x1030	0x3d	0x5030	0xba
0x0031	0xe6	0x4031	0xe8	0x1031	0xed	0x5031	0xeb
0x0032	0xff	0x4032	0xff	0x1032	0xff	0x5032	0xff
0x0033	0xff	0x4033	0xff	0x1033	0xff	0x5033	0xff
.....							

上面得到的 data_i& data_q 要转换成二进制补码格式。

下面是组合原始数据的 Python 脚本示例：

```
for i in range(0,int(len(read_array_i)/4)):

    data_0_i=read_array_i[(i)*4];
    data_1_i=read_array_i[1+(i)*4];
    data_2_i=read_array_i[2+(i)*4];
    data_3_i=read_array_i[3+(i)*4];

    data_0_i1=read_array_i1[(i)*4];
    data_1_i1=read_array_i1[1+(i)*4];
    data_2_i1=read_array_i1[2+(i)*4];
    data_3_i1=read_array_i1[3+(i)*4];

    data_i[2*i]=data_0_i+(data_1_i<<8)+(data_2_i<<16)+(data_3_i<<24)
    data_i[2*i]=twos_comp(int(data_i[2*i]),32)

    data_i[2*i+1]=data_0_i1+(data_1_i1<<8)+(data_2_i1<<16)+(data_3_i1<<24)

    data_i[2*i+1]=twos_comp(int(data_i[(2*i)+1]),32)

    data_0_q=read_array_q[(i)*4];
    data_1_q=read_array_q[1+(i)*4];
    data_2_q=read_array_q[2+(i)*4];
    data_3_q=read_array_q[3+(i)*4];

    data_0_q1=read_array_q1[(i)*4];
    data_1_q1=read_array_q1[1+(i)*4];
    data_2_q1=read_array_q1[2+(i)*4];
    data_3_q1=read_array_q1[3+(i)*4];

    data_q[i*2]=data_0_q+(data_1_q<<8)+(data_2_q<<16)+(data_3_q<<24)
    data_q[i*2]=twos_comp(int(data_q[i*2]),32)

    data_q[i*2+1]=data_0_q1+(data_1_q1<<8)+(data_2_q1<<16)+(data_3_q1<<24)

    data_q[i*2+1]=twos_comp(int(data_q[i*2+1]),32)

    i=i+1

print("data_i="+str(data_i))
print("data_q="+str(data_q))
```

下图为得到的原始数据曲线（时域）：

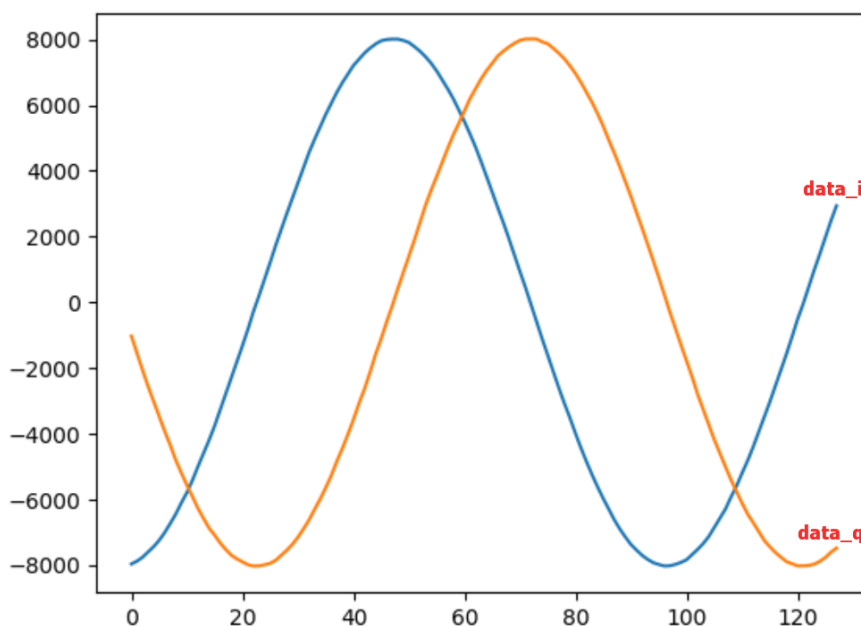


图 2. 原始数据曲线

将原始数据组合为复数格式: $\text{inputDataComplex} = \text{data_i} + j * \text{data_q}$, 然后做 FFT, 即可得到信号频谱。

```
N=len(data_i)
window=1
fftOutputComplex=(2**0)*np.fft.fft(window*inputDataComplex)/N

fftOutputComplexCopy = np.copy(fftOutputComplex)
fftOutputComplex[int(N/2)] = fftOutputComplexCopy[0]
fftOutputComplex[int(N/2)+1:N] = fftOutputComplexCopy[1:int(N/2)]
fftOutputComplex[1:int(N/2)] = fftOutputComplexCopy[int(N/2)+1:N]
fftOutputComplex[0] = fftOutputComplexCopy[int(N/2)]

magDataComplex=np.abs(fftOutputComplex)
magDataComplex[0]=magDataComplex[0]/2 #DC summed twice.
phaseDataComplex=np.angle(fftOutputComplex)

normComplexSpectrum=20*np.log10(magDataComplex/(2.**15)) #-20*np.log10(2**15)
plt.plot(range(-245760,245760,int(491520/(len(normComplexSpectrum))))),normComplexSpectrum)
plt.xlabel('Freq (KHz)')
plt.ylabel('Power(dBfs)')
plt.title("Power(Frequency domain)")
```

上面脚本中, FB data rate=491.52M.

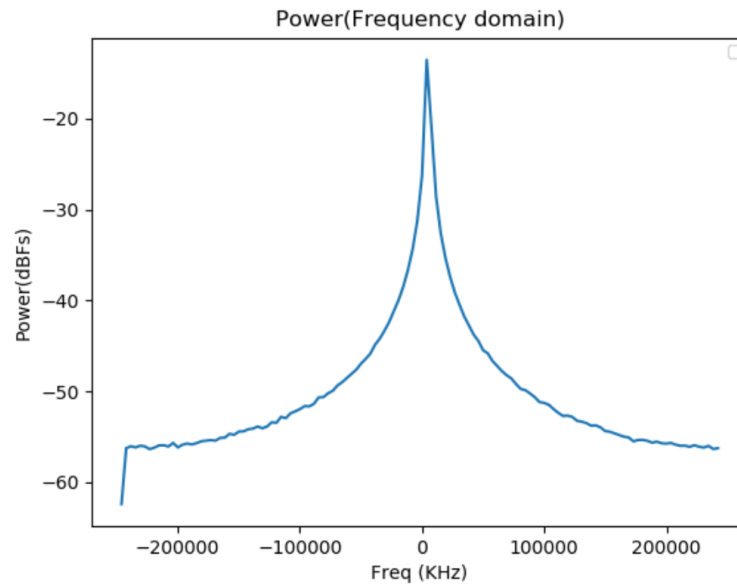


图 3. 频谱图

在频谱图中，我们看到 Skirt 效应是由于输入信号频点 ($=R_{Fin}-NCO$) 没有落在特定的 Bin 上（对于 $FB \text{ data rate}=491.52M$, $Bin=491.52M/128=3.84MHz$ ）。可以设置输入信号频点为 $3.84MHz$ 的整数倍（即做相干采样）或者加窗（例如设置 $window=blackman$ ）来解决此问题。

4 总结

本文介绍了 AFE77xx 内部 FB 抓数以及数据处理的详细步骤，用户可以通过这些步骤得到客户单板上的 FB 原始数据及 FFT 结果，帮助在项目早期快速定位问题。

5 参考资料

1. AFE77xx datasheet
2. AFE77xx Programming User Guide

重要声明和免责声明

TI 均以“原样”提供技术性 & 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](http://www.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