步骤1: 加载程序，烧入bit流文件

2：打开sdk工程，app2的build工程和run as，将ad关联，通过板卡上的灯进行判别。

3：ila1的数据提取出csv文件，并进行保存，存储到after里面，命名格式为J\*\_JIAOZHUNQIAN\_频率

4：将输入信号设置为113.147MHz，并进行采集，此时**int** NO2\_low\_NZ= 1 ; //0;//0--1717MHz(even); 1--113MHz(odd)

**int** NO2\_cap\_cal\_key = 0;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_key = 1;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_auto\_key = 1;//TI mismatch calibration key (auto mode), 1:on 0:off

**int** NO2\_TI\_cal\_manual\_key = 0;//TI mismatch calibration key (manual mode), 1:on 0:off

//int NO2\_pulse\_detect\_key = 0;

**int** NO2\_output\_sel\_key = 1;//0:raw data; 1:after cal (interstage gain / TI) data

这么表示，并存到after中，用matlab计算signalpower的数值，工程是在after里面需要通过调节信号源将signalpower的数值置为-1dB。

5：matlab当中需要进行修改，第一处为char\_data=char(s\_data(7));其中7的值需要按照笔记内来根据J\*进行确定，运行matlab，观察终端处共10组带有signalpower的结果，其中前四组代表一片ADC中一个通道的子AD1234，第五组表示该通道到整个ADC综合结果，6-9组的代表一片ADC中另一个通道的子AD1234，10表示该通道下的整个ADC综合结果。观察其Signalpower信号，通过多次调解信号源，实现signalpower的值为-1dB，并且该程序是adc\_data\_read\_new。

确定好信号源的db后，将**int** NO2\_output\_sel\_key = 1;//0:raw data; 1:after cal (interstage gain / TI) data

改为0，重新加载，提取ila\_1结果放入到before里面。

6：进行校准，打开sdk，找到AAD12D2000\_CONFIG\_NO\*，进入子模块中，\*根据笔记手册的J\*进行进入子模块，并重新配置ADC。

7：电容失配校准：

(1)裸数据模式

配置**int** NO2\_low\_NZ= 1 ; //0;//0--1717MHz(even); 1--113MHz(odd)，其中1表示一区，0表示2区，校准需要用一区校准。

**int** NO2\_cap\_cal\_key = 0;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_key = 1;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_auto\_key = 1;//TI mismatch calibration key (auto mode), 1:on 0:off

**int** NO2\_TI\_cal\_manual\_key = 0;//TI mismatch calibration key (manual mode), 1:on 0:off

这四个保持不变为0110

**int** NO2\_output\_sel\_key = 0;//0:raw data; 1:after cal (interstage gain / TI) data 整个0表示校准前数据，1表示校准后的数据结果，目前为校准前则设置为0

将修改的保存。

(2)：进行sdk的调试，点击蜘蛛箭头对应下的第一个，Debug处有显示，则表明程序加载进去，然后一直resume，直到ADC\_config\_adc4., spi\_send(spi\_base\_addr,533,0x02B0);

(3): 此时再采集数据将ila\_1的结果保存到before中，命名格式为J\*\_JIAOZHUNQIAN\_113.147\_-1（-1为113.147db数），进入matlab程序，capacitor\_mis\_cal\_20200728，将其作为主函数进行跑，需要修改参数

channel\_1\_key=0;%p6,ADC1

channel\_2\_key=1;%p3,ADC2;0:disable cap cal

其中P6和P3根据J\*决定，用哪个哪个就置为1。

char\_data=char(s\_data(7)); 与上述表达一直，数字7根据J\*进行确定修改。然后点击运行。观察其鱼叉曲线(17,18,19,20),拟合程度越高，效果越好，最高点不要超过4096，最低点不要拉倒最低端，终端会有8组数据，16个图，每四张图表示校准前后的时域和频域结果，对应终端的1组数据。

(4) 查看cap\_value.txt中的数据，将结果修改到对应的adc\_config\_ADC\*，同时将

//注释掉该部分

**int** NO2\_low\_NZ= 1 ; //0;//0--1717MHz(even); 1--113MHz(odd)

**int** NO2\_cap\_cal\_key = 1;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_key = 1;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_auto\_key = 1;//TI mismatch calibration key (auto mode), 1:on 0:off

**int** NO2\_TI\_cal\_manual\_key = 0;//TI mismatch calibration key (manual mode), 1:on 0:off

//int NO2\_pulse\_detect\_key = 0;

**int** NO2\_output\_sel\_key = 1;//0:raw data; 1:after cal (interstage gain / TI) data

其中NO2\_cap\_cal\_key 和NO2\_output\_sel\_key必须要改为1，NO2\_low\_NZ根据采集的时钟频率决定是0还是1.//————此步骤为电容失配校准，只在1区做，

8：交织校准：

更改ADC1和ADC2主要由J\*来决定更改哪一个

u16 NO2\_ADC1\_o\_mis1\_NZ1=5500; //to be refreshed

u16 NO2\_ADC1\_o\_mis2\_NZ1=1; //to be refreshed

u16 NO2\_ADC1\_o\_mis3\_NZ1=5600; //to be refreshed

u16 NO2\_ADC1\_o\_mis4\_NZ1=1; //to be refreshed

u16 NO2\_ADC1\_g\_mis2\_NZ1=5700; //to be refreshed

u16 NO2\_ADC1\_g\_mis3\_NZ1=1; //to be refreshed

u16 NO2\_ADC1\_g\_mis4\_NZ1=5800; //to be refreshed

u16 NO2\_ADC1\_t\_mis1\_NZ1=1; //to be refreshed

u16 NO2\_ADC1\_t\_mis2\_NZ1=0xbc98; //to be refreshed

u16 NO2\_ADC1\_t\_mis3\_NZ1=1; //to be refreshed

u16 NO2\_ADC1\_t\_mis4\_NZ1=0x5a00; //to be refreshed

**#endif**

u16 NO2\_ADC1\_o\_mis1\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_o\_mis2\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_o\_mis3\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_o\_mis4\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_g\_mis2\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_g\_mis3\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_g\_mis4\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_t\_mis1\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_t\_mis2\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_t\_mis3\_NZ2=0; //to be refreshed

u16 NO2\_ADC1\_t\_mis4\_NZ2=0; //to be refreshed

**#if** 1

/////////////////////////ADC2

u16 NO2\_ADC2\_o\_mis1\_NZ1=3409; //to be refreshed

u16 NO2\_ADC2\_o\_mis2\_NZ1=356; //to be refreshed

u16 NO2\_ADC2\_o\_mis3\_NZ1=172; //to be refreshed

u16 NO2\_ADC2\_o\_mis4\_NZ1=210; //to be refreshed

u16 NO2\_ADC2\_g\_mis2\_NZ1=49; //to be refreshed

u16 NO2\_ADC2\_g\_mis3\_NZ1=4088; //to be refreshed

u16 NO2\_ADC2\_g\_mis4\_NZ1=62; //to be refreshed

u16 NO2\_ADC2\_t\_mis1\_NZ1=496; //to be refreshed

u16 NO2\_ADC2\_t\_mis2\_NZ1=651; //to be refreshed

u16 NO2\_ADC2\_t\_mis3\_NZ1=186; //to be refreshed

u16 NO2\_ADC2\_t\_mis4\_NZ1=589; //to be refreshed

**#endif**

u16 NO2\_ADC2\_o\_mis1\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_o\_mis2\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_o\_mis3\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_o\_mis4\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_g\_mis2\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_g\_mis3\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_g\_mis4\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_t\_mis1\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_t\_mis2\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_t\_mis3\_NZ2=0; //to be refreshed

u16 NO2\_ADC2\_t\_mis4\_NZ2=0; //to be refreshed

这一部分进行参数的修改，如果使用1.5G信号则只修改2区，NZ2部分，先将信号源改成1313.147MHz，sdk配置为

**int** NO2\_low\_NZ= 0 ; //0;//0--1717MHz(even); 1--113MHz(odd)

**int** NO2\_cap\_cal\_key = 1;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_key = 1;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_auto\_key = 1;//TI mismatch calibration key (auto mode), 1:on 0:off

**int** NO2\_TI\_cal\_manual\_key = 0;//TI mismatch calibration key (manual mode), 1:on 0:off

//int NO2\_pulse\_detect\_key = 0;

**int** NO2\_output\_sel\_key = 1;//0:raw data; 1:after cal (interstage gain / TI) data

然后打断点，进行调试，不断观察expression内的添加的数据的值变化范围是否在4096以内，如果在则判断该芯片合格，将其中一组数据然后采用excel表格的形式将16进制数改为10进制数，再填回sdk对应的NO2\_ADC2\_o\_mis1\_NZ2=0部分(根据ADC1和ADC2来选择，都选用2区)。

然后将int NO2\_TI\_cal\_auto\_key = 0;//TI mismatch calibration key (auto mode), 1:on 0:off

int NO2\_TI\_cal\_manual\_key = 1;//TI mismatch calibration key (manual mode), 1:on 0:off

将模式修改为手动校准即对应的0和1。

最后打断点到spi\_send(spi\_base\_addr,533,0x02B0); (接近最后)，再debug，蜘蛛箭头+resume。直到spi\_send(spi\_base\_addr,533,0x02B0)再点击一次即可。

11：在进行数据采集，提取ila\_1的结果，保存到after中，命名格式为J\*\_JIAOZHUNHOU\_113.147\_\*，

12：进入after中的matlab程序，检查一下char\_data=char(s\_data(7)); %4´ú±íCSVÎÄ¼þÖÐµÚ4ÁÐÊý¾Ý£¬¶ÔÓ¦ÓÚADC1

的准确性，然后运行，观察SFDR结果，一般是第五个或者第十个为综合的结果。(目的是为了验证sdk中改的数的正确性)

13：将信号源改成1.500123G的信号，dbm和校准前保持一致就可以，dm根据自己要求随意，但需要通过signalpower来最终确认。

14：进入sdk工程，修改参数NO2\_low\_NZ= 0 ;(更改为2区)，

**int** NO2\_low\_NZ= 0 ; //0;//0--1717MHz(even); 1--113MHz(odd)

**int** NO2\_cap\_cal\_key = 1;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_key = 1;//capacitor mismatch calibration key, 1:on 0:off

**int** NO2\_TI\_cal\_auto\_key = 1;//TI mismatch calibration key (auto mode), 1:on 0:off

**int** NO2\_TI\_cal\_manual\_key = 0;//TI mismatch calibration key (manual mode), 1:on 0:off

//int NO2\_pulse\_detect\_key = 0;

**int** NO2\_output\_sel\_key = 1;//0:raw data; 1:after cal (interstage gain / TI) data