

DFSDM 时钟配置问题

关键字: DFSDM, 时钟

引言

DFSDM 全称为 Digital filter for sigma delta modulators。顾名思义，其作用主要是对外部 $\Sigma\Delta$ 调制的数字信号进行滤波。STM32L462xx 系列支持最多 4 个外部串行通道，2 个数字滤波器，最大可达到 24bit 的 ADC 分辨率。并且支持 SPI 接口和曼切斯特编码 1-wire 接口。

1. 问题描述

客户在使用 STM32L462xx 的 DFSDM 连接数字麦克风，将数字麦克风的 PDM 信号转化为 PCM 信号，并采集数据。使用的参数为 16KHz 采样，2M 左右的时钟驱动数字麦克风，能够正常读取麦克风数据。了解到他们所使用的数字麦克风可以使用 768K 时钟驱动，从而达到更低功耗的状态，但是客户无法配置出合适的时钟，并且勉强配置到了 768K 附近，通过 DFSDM 采集到的数据也是混乱的，完全无法解析。

2. 问题分析与定位

在了解到基本需求后，我们需要对 DFSDM 的时钟有一定了解。从 RM0394 参考手册中可以找到如下内容：

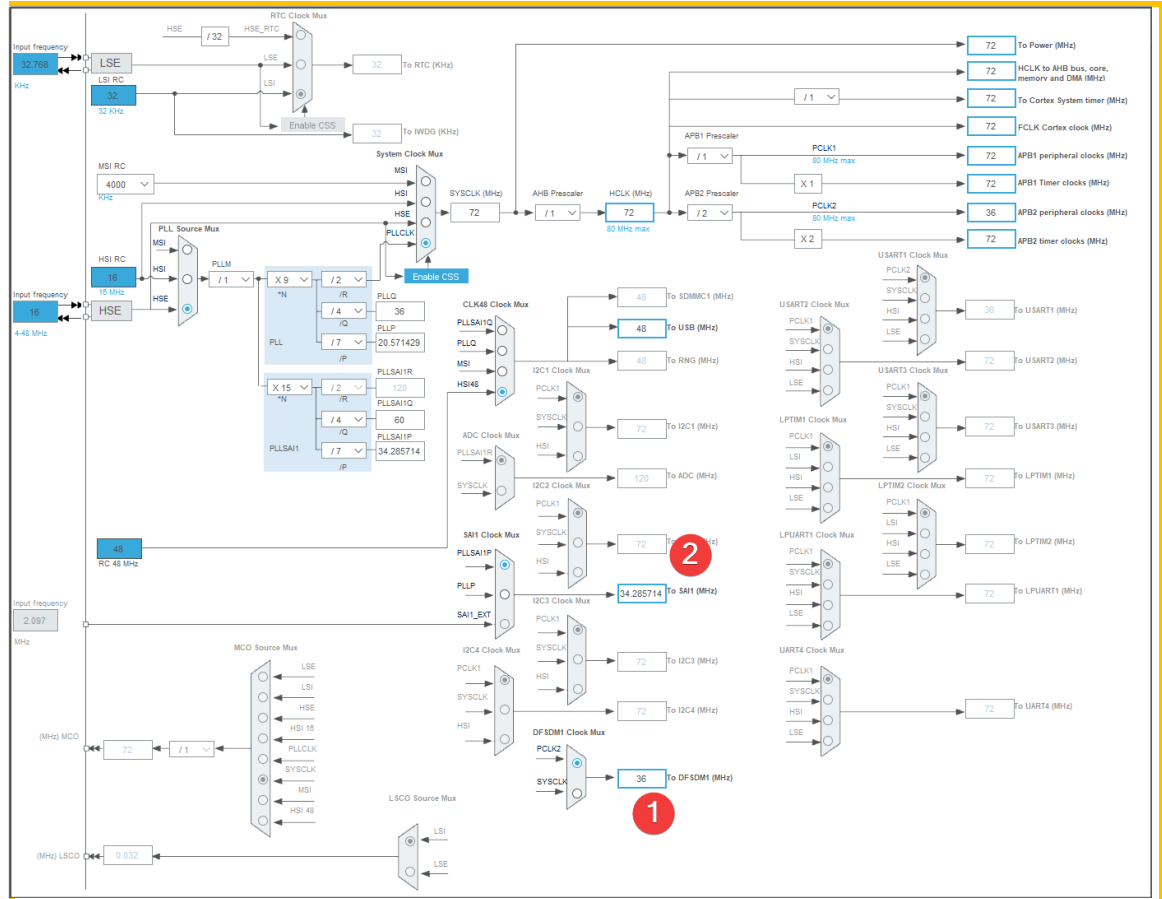
图1. DFSDM 时钟

DFSDM can provide one external output clock signal to drive external $\Sigma\Delta$ modulator(s) clock input(s). It is provided on CKOUT pin. This output clock signal must be in the range specified in given device datasheet and is derived from DFSDM clock or from audio clock (see CKOUTSRC bit in DFSDM_CH0CFGR1 register) by programmable divider in the range 2 - 256 (CKOUTDIV in DFSDM_CH0CFGR1 register). Audio clock source is SAI1 clock selected by SAI1SEL[1:0] field in RCC configuration (see).

DFSDM 可以提供一个时钟用于驱动外部 sigma delta 调制器，并且时钟来源可以是 DFSDM 时钟或者 Audio 时钟，其中 Audio 时钟就是 SAI1 的时钟。

在处理与时钟相关的配置问题时，强烈建议使用 CubeMX 的时钟配置界面进行配置。我们先来看下客户用于驱动外部数字麦克风的时钟配置。

图2. 时钟树



从图中可以看出，DFSDM 的时钟为 36MHz，来自 PCLK2。SAI1 的时钟约为 34.29MHz 来自 PLLSAI1P。

```
hdfsdm1_channel1.Init.OutputClock.Activation = ENABLE;
hdfsdm1_channel1.Init.OutputClock.Selection = DFSDM_CHANNEL_OUTPUT_CLOCK_AUDIO;
hdfsdm1_channel1.Init.OutputClock.Divider = 17;
```

这是关于 DFSDM 输出时钟的代码片段，可以看到使用了 AUDIO 的时钟作为输出，并且使用 17 分频，那么最终驱动数字麦克风的时钟为 $34.29/17 \approx 2.02\text{MHz}$ 。

再来看看滤波器部分参数的设计：

```
hdfsdm1_filter0.Init.FilterParam.SincOrder = DFSDM_FILTER_SINC4_ORDER;
hdfsdm1_filter0.Init.FilterParam.Oversampling = 128;
hdfsdm1_filter0.Init.FilterParam.IntOversampling = 1;
```

其中，使用了 4 阶 SINC 滤波，过采样参数为 128，那么 $2.02\text{MHz}/128 \approx 16\text{KHz}$ 。也就是说这种配置参数下，可以接近 16KHz 的采样率来对音频数据采样。

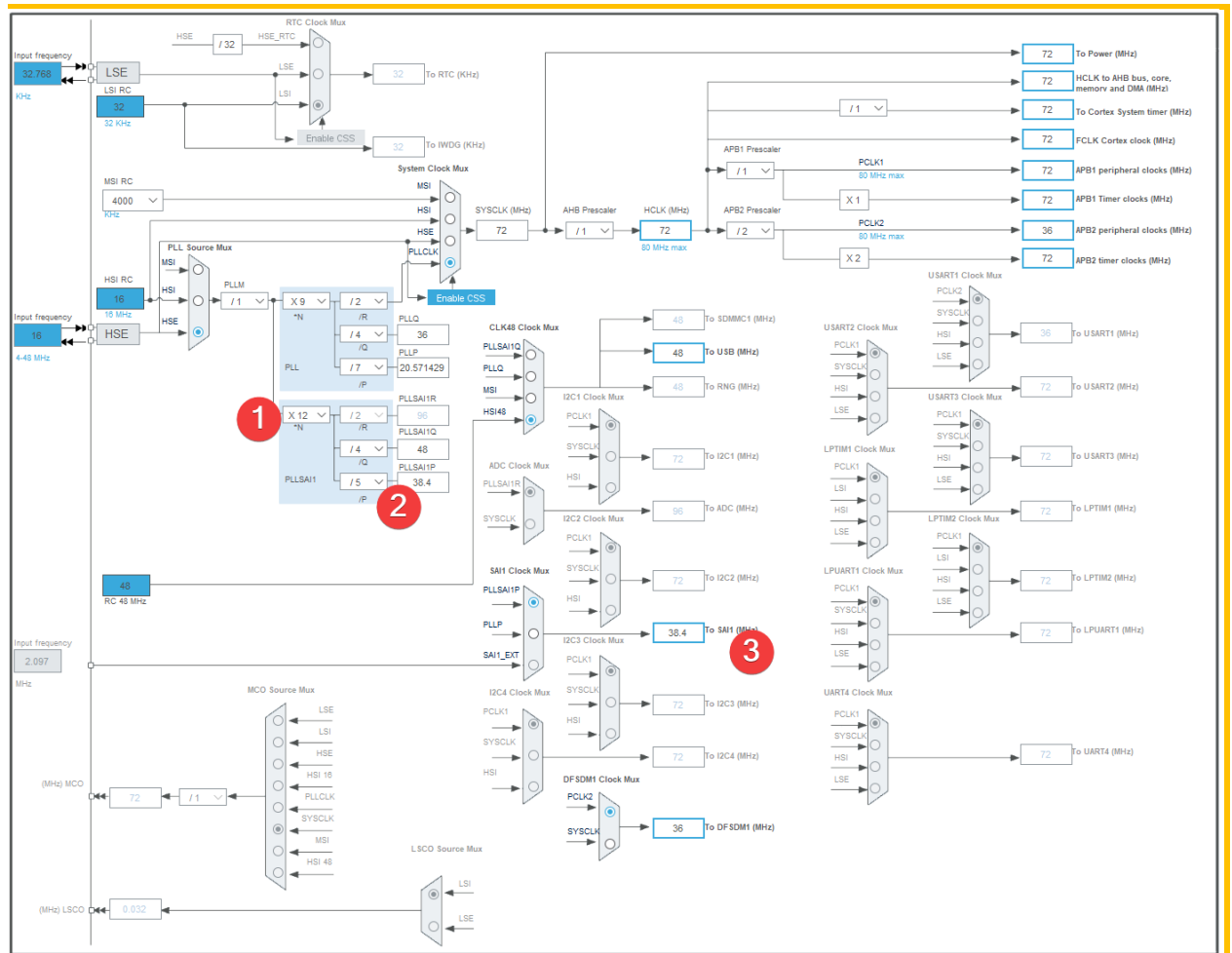
客户想要将麦克风驱动时钟重新配置为 768K，但只考虑修改 Divider 和 Oversampling 参数是配置不出来的。

3. 问题解决

我们可以反过来推，要想获得 16KHz 采样率，768KHz 的时钟，首先需要满足以下公式：768K/ Oversampling = 16K，那么 Oversampling=48。Oversampling 这个参数是比较好确定的，而 768K 来自于 CLK_SAI1/Divider，这两个参数目前无法确定，其中 CLK_SAI1 又来自于 PLLSAI1P，它由 PLLSAI1_N, PLLSAI1_P 两个系数决定。也就是说要想获得 768K 的时钟，需要把这些参数都配置为合适的值。到这一步，实际上没有直接的公式可以求下去了，只能自己测试并调整，建议可以先将 Divider 定好，然后通过 CLK_SAI1/Divider=768K 就可以求出 CLK_SAI1 的值，将这个值手动输入到 Cubemx 的时钟配置界面可以自动求出合适的 PLLSAI1_N, PLLSAI1_P 系数，这样可以大大减少调整时间。如果无法求出合适的值，CubeMX 也会进行提示，那么只能重新换 Divider 来测试了。

最终我调试下来的值如下图所示：

图3. 时钟树



我选择的 Divider=50，那么可以算出 CLK_SAI1=50*768K=38.4MHz，输入到上图界面中可以得到 PLLSAI1_N=12, PLLSAI1_P=5。

至此，已经可以正确得到 768KHz 的时钟驱动麦克风了，但是目前仍然无法获取到准确的音频数据。

由于前面修改了 Oversampling，该参数会影响采样值。在参考手册中可以找到下表：

图4. 滤波器最大输出精度表

Table 99. Filter maximum output resolution (peak data values from filter output) for some FOSR values						
FOSR	Sinc ¹	Sinc ²	FastSinc	Sinc ³	Sinc ⁴	Sinc ⁵
x	+/- x	+/- x ²	+/- 2x ²	+/- x ³	+/- x ⁴	+/- x ⁵
4	+/- 4	+/- 16	+/- 32	+/- 64	+/- 256	+/- 1024
8	+/- 8	+/- 64	+/- 128	+/- 512	+/- 4096	-
32	+/- 32	+/- 1024	+/- 2048	+/- 32768	+/- 1048576	+/- 33554432
64	+/- 64	+/- 4096	+/- 8192	+/- 262144	+/- 16777216	+/- 1073741824
128	+/- 128	+/- 16384	+/- 32768	+/- 2097152	+/- 268435456	Result can overflow on full scale input (> 32-bit signed integer)
256	+/- 256	+/- 65536	+/- 131072	+/- 16777216		
1024	+/- 1024	+/- 1048576	+/- 2097152	+/- 1073741824		

从表中可以看到，之前是 FOSR=128，Sinc4，对应的数值为+/-268435456，但是 DFSDM 的数据寄存器只有 24bit， $2^{23} < 268435456$ 的，也就是说需要进行移位，只能以损失精度为代价进行转换。客户代码中有以下配置：

```
hdfsdms1_channel11.Init.Offset = 0;
hdfsdms1_channel11.Init.RightBitShift = 0x05;
```

数据需要右移 5 位保存到数据寄存器，那么现在我们的 FOSR 修改为了 48，对应上表可以看到数值是比较小的，不需要移位处理，所以因为改为：

```
hdfsdms1_channel11.Init.Offset = 0;
hdfsdms1_channel11.Init.RightBitShift = 0x00;
```

到这里，我们需要修改的内容就全部结束了，将这些修改内容通知客户，让其进行测试，测试结果也比较满意。

4. 总结

DFSDM 在数字音频应用中比较常见，主要用于将数字麦克风的 PDM 信号转换为 PCM 信号，如果没有 DFSDM 外设，就只能使用软件库进行转化，效率比较低。当然，也可以用于一些模拟信号的采集，需要外部 Sigma Delta 调制器调制好之后再进行滤波，这样得到的 ADC 精度要比通用的 ADC 精度更高，在要求高精度 ADC 的场合，不失为一种较好的方案。

参考文献

STM32L462xx 参考手册 RM0394:

- https://www.st.com/content/ccc/resource/technical/document/reference_manual/group0/b0/ac/3e/8f/6d/21/47/af/DM00151940/files/DM00151940.pdf/jcr:content/translations/en.DM00151940.pdf

文档中所用到的工具及版本

测试工具版本信息:

- CubeMx: 6.3.0

版本历史（内部参考）

(仅供内部参考，评审完成后 Auditor 需要删除这部分内容)

日期	版本	变更	撰写/变更者
2021 年 11 月 5 日	0.1	原始版本	Wayne Xiong
2021 年 11 月 25 日	0.2	reviewed	Michael Liang
2022 年 1 月 14 日	0.3	个别字词调整、格式及版本维护	Miler Shao

版本历史

日期	版本	变更
YYYY 年 MM 月 DD 日	1.0	详细变更信息

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 www.st.com/trademarks。所有其他产品或服务名称均为其各自所有者的财产。

本文档是 ST 中国本地团队的技术性文章，旨在交流与分享，并期望借此给予客户产品应用上足够的帮助或提醒。若文中内容存有局限或与 ST 官网资料不一致，请以实际应用验证结果和 ST 官网最新发布的内容为准。您拥有完全自主权是否采纳本文档（包括代码，电路图等信息，我们也不承担因使用或采纳本文档内容而导致的任何风险。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2020 STMicroelectronics - 保留所有权利