

*. Author:Vactus

*Date:2019-6-4

*Mod: 本文件对于调用 CIC IP 核滤波器进行一些说明和个人见解

今天在使用 CIC IP 核滤波器的时候碰到一些问题是以前从来没有注意到的。也是学习一下。

一般 FIR IIR 都是固定不变的采样率。传统的速率转换可以实现采样速率的转换，假如一个正弦模拟信号，AD 的采样速率是 f_1 ，你需要采样率为 f_2 的信号，传统方法是信号经过 f_1 采样后进行 DA 转换，再将转换后的模拟信号经过 f_2 采样频率抽样，得到采样率为 f_2 的数字信号。这样做不仅麻烦，处理不好很容易使信号受到损伤，所以现在使用抽取和内插。

首先说一些基础概念(也是今天认真看的，之前学过，不过没有认真注意)

混叠：当采样率为 FS 对一个信号进行采样时，信号中大于 $FS/2$ 以上的频率不是消失了，而是对称映射到 $FS/2$ 以下的频带中，并且和原有 $FS/2$ 以下的频率成分叠加起来，这就是混叠。

频率混叠现象：是由于采样信号频谱发生变化，而出现高、低频成分发生混淆的一种现象。抽样时频率不够高，抽样出来的点既代表了信号中的低频信号的样本值，也同时代表高频信号样本值，在信号重建的时候，高频信号被低频信号代替，两种波形完全重叠在一起，形成严重失真。

消除混叠的方法有两种： 1. 提高采样频率 F , 即缩小采样时间间隔. 然而实际的信号处理系统不可能达到很大的采样频率, 处理不了很多的数据. 另外, 许多信号本身可能含有全频带的频率成分, 不可能将采样频率提高到无穷大. 所以, 通过采样频率避免混叠是有限制的.

2. 采用抗混叠滤波器. 在采样频率 F 一定的前提下, 通过低通滤波器滤掉高于 $F/2$ 的频率成分, 通过低通滤波器的信号则可避免出现频率混叠。

建议可以去看参考博客 1 和 2，因为版权问题不在多描述，里面图和代码非常详细，对理解很有帮助。

根据奈奎斯特定理，当采样率 $f_s > 2f_{\max}$ 时，就能够从采样后的数据中无失真地恢复原来的信号，信号在时域的采样等效于信号频谱在频域上的周期沿拓，满足奈奎斯特定律就是为了保证采样后的信号频谱不发生重叠。可是有时也会用到的欠采样和过采样。

欠采样： 欠采样允许 ADC 作为一个混频器，能够接收调制高频载波信号并产生较低频率的镜像。这种方式下，就像下变频器。另一种主要优点是允许 ADC 的采样率低于奈奎斯特频率，一般具有较明显的成本优势。

例如，假设调制载波为 10MHz，带宽为 100kHz（ $\pm 50\text{kHz}$ ，中心频率为 10MHz）。以 4MHz 进行欠采样，产生 1 阶和与差项 (f_1+f_2 和 f_1-f_2)，分别为 14MHz 和 6MHz；2 阶项 ($2f_1$ 、 $2f_2$ 、 $2f_1+f_2$ 、

$f_1/2f_2$ 、 $|2f_1-f_2|$ 、 $|f_1-2f_2|$)，分别为 8MHz、20MHz、18MHz、2MHz、24MHz 和 16MHz。出现在 2MHz 处的镜像信号为有用信号。注意，我们的原始信号在 10MHz，通过对其进行数字化在 2MHz 产生了镜像。现在，我们可以在数字域进行信号处理(滤波和混频)，恢复原始 50kHz 信号。

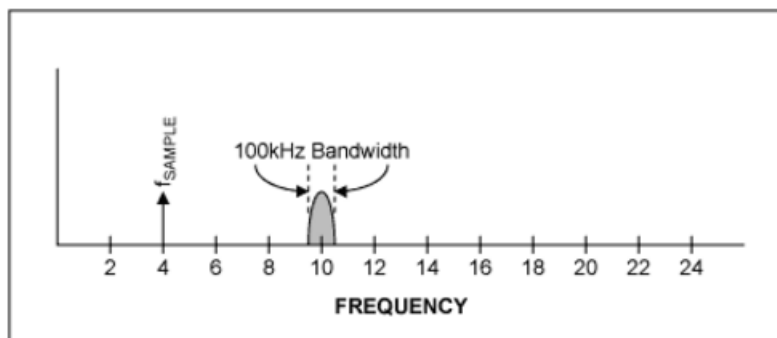


图3a. 欠采样示例。

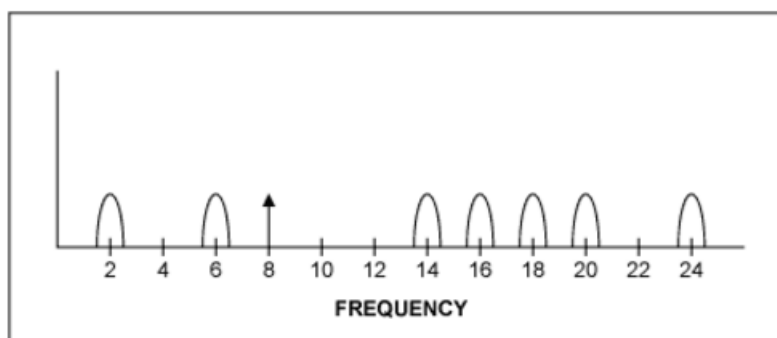


图3b. 欠采样数字化的镜像信号(1阶和2阶)。

该过程无需大幅的模拟处理，这是其主要优势之一。由于所有处理都在数字域完成，如果需要对电路的性能和特性进行更改，只需修改软件即可。相对而言，对于模拟设计，如果需要更改电路性能，需要改变电路硬件元件和布局，并且成本相当高。

过采样：过采样提供所谓的处理增益。在过采样时，以较高采样频率获得多出实际需要的采样数量，然后对数据滤波，从而有效降低系统的噪底(假设噪声为宽带白噪声)。这不同于平均，后者是获取很多采样，噪声被平均。可以这么理解过采样：如果输入信号来自于扫描频率的信号源，频谱则可以分为多个范围或“容器”，每个容器的带宽固定。宽带噪声分散在整个有用频率范围内，所以每个容器具有特定量的噪声。现在，**如果提高采样率，那么频率容器的数量也增多。在这种情况下，出现的噪声量仍相同，但我们有更多的容器可供容纳噪声。然后我们利用滤波器滤除超出有用频带的噪声。结果就是每个容器的噪声减少，所以就通过过采样有效降低了系统的噪底。**

具体需要做滤波器的可以参考里面的博客。

参考博客：

1. <https://blog.csdn.net/daduzimama/article/details/80182108>
2. <https://blog.csdn.net/daduzimama/article/details/80263296>
3. <https://blog.csdn.net/wzz4420381/article/details/50678197>