Freescale Semiconductor, Inc.
应用笔记

Document Number: AN4847
Rev. 2, 07/2015

在 Σ-Δ ADC上使用FFT

by: Ludek Slosarcik

1 简介

本应用笔记讨论数字信号处理与计量中的两个专题：快速傅里叶变换(FFT)和Σ-Δ模数转换器(SD ADC)。

首先， FFT是一种数学方法，用于将信号的表示方法由时域转换为频域。在频域中，信号会转换为一系列不同振幅，不同频率的信号叠加。 FFT在频率（频谱）分析领域极其重要。其次， SD ADC是高分辨率、高集成度且低成本的ADC，适用于计量、过程控制和监控等应用。

本应用笔记中深入阐述了这两个专题，感兴趣的用户可通过相关网页或参考资源获取更多信息。本应用笔记旨在介绍一些方法，以便开发者在应用中（特别是电力计量应用）合理地使用FFT结合SD ADC。

目录

2 基本问题描述

在电力 （电能）计量应用中使用FFT时具有一些特定的要求，*《面向计量应用的基于FFT的算法》*（文档AN4255）中对这些要求进行了说明。其基本要求是：在一个输入信号周期内的采样数需为2的幂。为了满足这一要求，必须了解输入信号的频率以计算相邻样本之间的正确时间。这样才能确保信号周期内的采样数为2的幂。这样做的前提是，每个ADC样本之间的时间能够通过可编程延时模块(PDB)或另一个特殊定时器进行微调，定时器应能为ADC使用的硬件触发器提供可控的间隔节拍。严格而言，我们需要使输入信号频率与ADC采样率保持同步。例如，该技术可用在基于逐次逼近寄存器(SAR)并且可由PDB硬件触发的ADC上。另外，也有某些其他类型的ADC及应用无法使用该技术。在Σ-Δ ADC上就存在此问题，其采样间隔无法微调。但某些用例可以避开这个问题。因此，本应用笔记是原来未解决此问题的AN4255应用笔记在逻辑上的延拓。

3 用例1 – 同步处理

为了确保SD ADC输出样本数为2的幂数，以便后续的FFT计算，最简单的方法是使信号频率和ADC采样率两者同步。在AN4255中对该过程进行了说明。这需要用到关于测量信号频率的知识，并且可在单次转换模式下运行SD ADC。可通过执行过零检测(ZCD)技术对信号频率 （周期）进行检测， AN4255的3.2.1节中对此进行了说明。由于SD ADC的启动时间过长，因而不太适合单次转换配置，但在某些情况下我们仍可以使用该AD转换模式。在此模式下使用SD ADC的主要限制因素是启动时间，该时间可能是普通AD转换所需时间的好几倍。这是由于采样滤波器的延迟所导致。例如， Freescale MKM34Z128 MCU （ARM® Cortex®-M0+内核） [2.]中使用的SD ADC的启动时间通常比其普通转换时间高出3倍。在配合后续的FFT计算时，该特性会限制这一模式在更高的输出采样率(OSR)中的使用。

4 用例2 – 异步处理

在SD ADC上使用FFT的另一种解决方法是使用过采样技术。实际上， SD ADC在信号周期内收集的样本远比实际需要的多。该过程的下一步是使用某些再计算技术，将个数并非2的幂数的原始ADC样本转换为FFT所需的2的幂数个样本。我们可以将这整个过程简单地称为输入信号插值法。图1以图形方式表示了该过程。为了简化图形，只使用了8个FFT点和12个测量样本 （比率为1.5）。虽然理论上可以使用欠采样方法结合相同的后续过程 （对2的幂数个样本进行插值），但实际上输入样本数高于FFT实际所需数往往会得到更佳的结果。这是由于插值信号存在最小计算误差。因此，输入样本数与2的幂样本数之比应大于1。该比率的上限受所用MCU的计算能力限制。通用规则：比率越高，插值误差越低，但MCU(ADC)负荷较大。

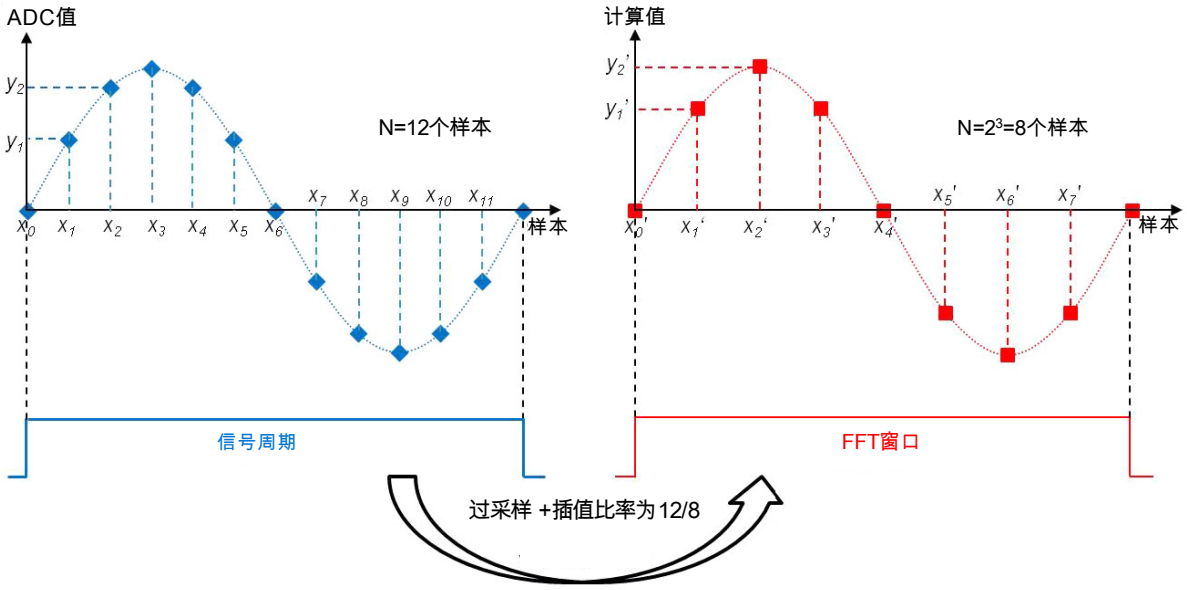


图1. FFT过采样用例

该计算方法通常假设这两种过程 （信号频率和ADC采样率）差不多完全异步。借助该计算方法，我们可以将SD ADC运行在连续转换模式下，这种模式比单次转换模式更适合于SD ADC（见节3，“用例1 – 同步处理”，位于第2页）。

如上所述，该用例的关键是插值。在数值分析的数学领域中，插值是在一组离散的已知数据点中构建新数据点的方法。对图1而言，新数据点在右侧 （红色数据集），而已知数据点在左侧 （蓝色数据集）。已知数据点由ADC测量而得，新数据点由MCU计算而得。因此，插值提供了一种方法来估算已知数据点 （例如测量点）中间处的函数。

大家熟知的插值方法有很多，以下对其中个别方法进行说明。这些方法在基于MKM34Z128 MCU的双相计量参考设计中经过了实际的测试。

4.1 线性插值

线性插值是在数据点之间的位置获取数值的最简方法。数据点之间由直线段简单地连接。每段 （以相邻两个数据点为边界）均可独立插值。在图1的示例中，我们有一组数据点(x_0,y_0), (x_1,y_1),...,(x_{11},y_{11})。在这组数据点上的线性插值定义为每对数据点之间线性插值体的级联。线性插值体是相邻数据点之间的直线。图2中显示了一段插值的细节。颜色表示与图1中一致，即蓝色点为测量值，红色点为计算值。

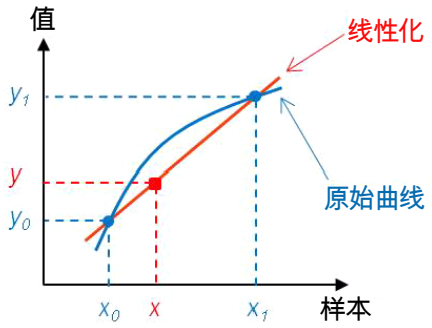


图2. 线性插值示例

对于区间(x_0,x_1)内的值x，其在直线上对应的值y的关系如以下公式所示：

$$\frac{y-y_0}{x-x_0}=\frac{y_1-y_0}{x_1-x_0} \qquad \textbf{公式 1}$$

根据此公式求解x处的y，得到：

$$y=(y_1-y_0)\frac{x-x_0}{x_1-x_0}+y_0=(ax+b) \qquad \textbf{公式 2}$$

其中a 和b为线性系数；a 为增益， b为偏移量。

线性插值是多项式插值在阶数等于1时的特例。线性插值快速简便，但是精度不高。当输入信号中存在高次谐波且输入样本与计算点之间的数量比率较低时，该问题尤为明显。如可行，我们可以通过扩大比率来解决该问题。

4.2 多项式插值

多项式插值是通过多项式对一组给定数据进行插值。假设我们有n+1个离散数据点(x_0,y_0), (x_1,y_1),....,(x_n,y_n)，这些数据点的x坐标均不同。主要目的是找到一个可以经过这n+1个数据点的n次多项式函数。该多项式称为插值多项式。有许多已知方法可用来解决该任务。其中最简单的解法是使用拉格朗日插值多项式。这是一种众所周知的数值分析方法。拉格朗日插值多项式的一般公式为：

$$y=L(x)=\sum_{i=0}^{n} y_i \cdot L_i(x) \qquad \textbf{公式 3}$$

其中， n为多项式的阶数， $L_i(x)$为拉格朗日基本多项式，其表达式为：

$$L_i(x)= \prod_{j=0, j\neq i}^{n} \frac{(x-x_j)}{(x_i-x_j)} \qquad \text{公式 4}$$

基本多项式$L_i(x)$具有属性：

$$L_i(x_j)= \begin{cases} 0 & j \neq i \\ 1 & j = i \end{cases} \qquad \text{公式 5}$$

多项式插值是线性插值的泛化形式，但其精度高于线性插值。其精度取决于多项式的阶数（公式3中的n的大小）。这种方法能为具有更多变量的函数提供更准确的结果。另一方面，插值多项式的计算比线性插值更为复杂，这主要针对实时应用而言 （如电力计量）。因此，以下内容仅针对最高阶数为3的多项式插值进行介绍。换言之，我们不会去计算一个n次插值多项式，而是将该任务分解为寻求多个简单的低次 （n=2或n=3最大）插值多项式。实际上，我们将通过3或4个相邻数据点计算y值 （x处的未知值）。利用多个新的相邻数据点(x_i,y_i)重复相同的过程以获取每对(x,y)值。以下将对该流程进行说明。

4.2.1 拉格朗日二次插值

求解基本拉格朗日插值多项式公式3在n=2时的一般值y （ x处的未知值），得到：

$$y= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2 = \qquad \text{公式 6}$$

$$= y_0L_0(x) + y_1L_1(x) + y_2L_2(x)$$

其中$L_i(x)$为基本拉格朗日二次多项式。

求解该公式需要三对 （通常为n+1）输入数据点(x_0,y_0)、 (x_1,y_1)和(x_2,y_2)。新的x值应位于x_0 - x_1或x_1 - x_2之间。实际上，新的x值为重新计算所得的值 （关于新的x_i值还可参见图1）。其为ADC值的个数与FFT点数 （2的幂）之比的倍数。

n=2的插值示例如图3所示。三个输入点 （测量值）显示为蓝色。经过这些输入点的插值函数为红色抛物线。通常，阶数为2的拉格朗日基本多项式$L_i(x)$为二次函数。其曲线为抛物线。

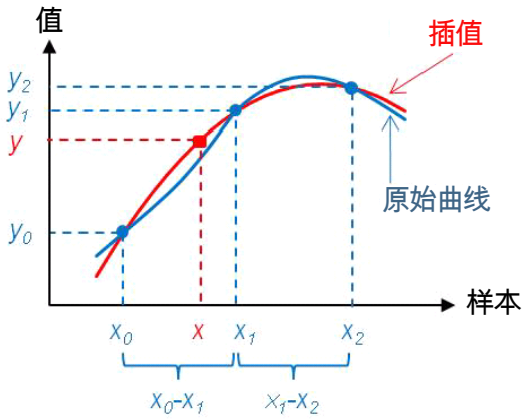


图3. 二次插值示例

通常，3个x值之间的间距不需要相等，(x_0 - x_1可以不等于x_1 - x_2)。不过，数据点均匀分布是ADC周期性采样间隔 （即用于计量应用）的典型用例。这样简化之后，我们可以重写公式6为以下形式：

$$y = \frac{(x-x_0-1)(x-x_0-2)}{2}y_0 - (x-x_0)(x-x_0-2)y_1 + \frac{(x-x_0)(x-x_0-1)}{2}y_2 \qquad \textbf{公式 7}$$

4.2.2 拉格朗日三次插值

求解基本拉格朗日插值多项式公式3在n=3时的一般值y （ x处的未知值），得到：

$$y = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}y_2 +$$

$$+ \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}y_3 = y_0L_0(x) + y_1L_1(x) + y_2L_2(x) + y_3L_3(x) \qquad \textbf{公式 8}$$

其中$L_i(x)$为基本拉格朗日三次多项式。

已知4个离散的输入数据点(x_0,y_0)、(x_1,y_1)、(x_2,y_2)和(x_3,y_3)，我们可以得到每个独立的三次多项式$L_i(x)$。新的x值应位于这三个区间中的某区间内：$x_0 - x_1$、$x_1 - x_2$或$x_2 - x_3$。实际上，新的x值为重新计算所得的值 （关于新的x_i值还可参见图1）。其为ADC值的个数与FFT点数 （2的幂）之比的倍数。

n=3的插值示例如图4所示。四个输入点 （测量值）显示为蓝色。经过这些输入点的插值函数为红色三次曲线。通常，阶数为3的拉格朗日基本多项式$L_i(x)$为三次函数。

类似前一示例，我们假设计量应用中的x点呈均匀分布。因此，公式8可改写为以下形式：

$$y = \frac{(x-x_0-1)(x-x_0-2)(x-x_0-3)}{-6}y_0 + \frac{(x-x_0)(x-x_0-2)(x-x_0-3)}{2}y_1 =$$

公式 9

$$= \frac{(x-x_0)(x-x_0-1)(x-x_0-3)}{-2}y_2 + \frac{(x-x_0)(x-x_0-1)(x-x_0-2)}{6}y_3$$

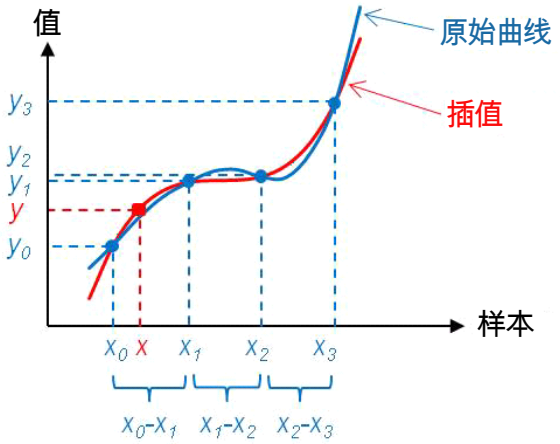


图4. 三次插值示例

5 实践操作

本章节将介绍FFT重新采样和插值技术的实际操作，采用C语言代码并在Excel®中对这些技术进行仿真。其中包含对于三个C语言函数 （用于大多数常用电表拓扑）的应用程序编程接口(API)的说明，这三个函数用于在最终FFT计算前对测量的AD样本进行插值。关于后续FFT计算所用的API，在另一个应用笔记AN4255[1.]中进行了说明。这些插值函数是基于FFT的计量库METERLIBFFT中的一部分，该库文件在某些Freescale电表参考设计 （[4.]和[5.]）中用于功率计算。

5.1 METERLIBFFT_Interpolation

这三个函数用于对无符号整型样本提供的原始输入曲线进行插值，形成2的幂数个样本 （FFT函数所需）构成的曲线。每种算法的C源代码包含在METERLIBFFT库的*fft.c*文件中。这些函数支持过采样和欠采样。附录A，“在单相用户代码中使用插值函数的示例”中给出了使用插值函数的C语言代码示例。该示例中还使用了其他函数。这些函数 （AN4255[1.]中对相关API进行了说明）也是计量库的一部分，用于基于FFT的功率计算。

5.1.1 语法

```
#include “meterlibfft.h”
long METERLIBFFT1PH_Interpolation (tMETERLIBFFT1PH_DATA *p, unsigned long u_ord, unsigned long i_ord, unsigned long samples_inp);
long METERLIBFFT2PH_Interpolation (tMETERLIBFFT2PH_DATA *p, unsigned long u_ord, unsigned long i_ord, unsigned long samples_inp);
long METERLIBFFT3PH_Interpolation (tMETERLIBFFT3PH_DATA *p, unsigned long u_ord, unsigned long i_ord, unsigned long samples_inp);
```

5.1.2 参数

表1. METERLIBFFT_Interpolation 函数参数

类型	名称	方向	说明
tMETERLIBFFT1PH_DATA	p	输入	指向单相计量库数据结构的指针
tMETERLIBFFT2PH_DATA	p	输入	指向双相计量库数据结构的指针
tMETERLIBFFT3PH_DATA	p	输入	指向三相计量库数据结构的指针
无符号长整型	u_ord	输入	电压插值阶 – 请参见表2
无符号长整型	i_ord	输入	电流插值阶 – 请参见表2
无符号长整型	samples_inp	输入	输入样本数可以高于或低于FFT所需样本数 （2的幂）

表2. 插值阶定义

定义名称	说明
ORD1	1阶 （线性）插值
ORD2	2阶 （二次）插值
ORD3	3阶 （三次）插值

5.1.3 返回

这些函数返回以下错误代码之一，仅对输入样本少于FFT样本的欠采样用例有效：

- FFT_ERROR (positive) – FFT样本多于输入样本，且FFT采样超过128个。
- FFT_OK (zero) – 欠采样率正确。

5.1.4 调用顺序

这些函数仅在需要插值处理时使用。在此情况下，系统将以既定的间隔周期调用这些函数，具体取决于线路频率。这些函数的调用时机应在即将进行主(FFT)计算处理之前。在调用这些函数之前必须执行单次强制参数初始化。除了其他任务之外，此参数初始化函数还可设置所需的FFT点数，并初始化插值函数所用的输入缓冲区的所有指针。

附注

输入缓冲区中的原始值 （ADC值）将在调用这些函数之后由新（已插值）的值改写。

5.1.5 性能

表3. METERLIBFFT_Interpolation 函数性能 （适用于CM0+内核）

函数名称	代码大小[B]			堆栈大小[B][1]	时钟周期数[2]		
	1阶	2阶	3阶		1阶	2阶	3阶
METERLIBFFT1PH_Interpolation	506	842	1654	512	12521	35260	76996
METERLIBFFT2PH_Interpolation	586	922	1734		24946	70519	153991
METERLIBFFT3PH_Interpolation	682	1018	1830		37418	106019	231227

1 由于示例是欠采样用例 （输入样本数 < FFT样本数）

2 输入样本数 = 120，所需FFT点数 = 64，两个通道的插值阶相同

表4. METERLIBFFT_Interpolation 函数性能 （适用于带MMAU的CM0+内核）

函数名称	代码大小[B]			堆栈大小[B][1]	时钟周期[2]		
	1阶	2阶	3阶		1阶	2阶	3阶
METERLIBFFT1PH_Interpolation	940	1116	1560	512	7388	15543	28304
METERLIBFFT2PH_Interpolation	1020	1196	1640		14728	30942	56607
METERLIBFFT3PH_Interpolation	1116	1292	1736		22067	46533	84911

1 由于示例是欠采样用例 （输入样本数 < FFT样本数）

2 输入样本数 = 120，所需FFT点数 = 64，两个通道的插值阶相同

5.2 仿真结果

四个示例的仿真结果分别记录在附录B、 C、 D和E中。这些都是典型的计量用例。所有示例均在Excel中进行了仿真。每个示例均具有参数表，用于说明仿真条件。随后是两张信号曲线图，用于实际演示输入信号与插值输出信号的图形。最后附上三张误差曲线图，分别针对每种特定的插值方法。这些误差曲线图表示一个周期内每个数据点的理想信号 （非插值）与插值信号之间的计算误差百分比。

6 总结

本应用笔记说明了多种插值方法，便于在难以修改ADC周期性采样间隔 （如Σ-Δ ADC）的特定计量应用中正常使用FFT算法。所述的插值方法是基于其在实时应用中的高效计算性能而选择的方法，并不是仅有的插值方法。比率在2-3之间的拉格朗日二次插值方法具有良好的计算性能和精度。本软件库支持的最佳插值方法 （相对于精度而言）是三次插值法，特别适用于对失真度较大的电流信号进行插值。当然，在数学中还有很多其他常用的插值方法可用。这些方法在计量应用中的限制因素是计算性能与最终精度。

7 参考文档

1. *《面向计量应用的基于FFT的算法》*（文档AN4255）
2. *《Kinetis M参考手册》*（文档MKMxxZxxACxx5RM）
3. 维基百科词条 “Interpolation”（插值）、“Linear interpolation”（线性插值）、“Polynomial interpolation”（多项式插值）和 “Lagrange polynomial”（拉格朗日多项式），可在en.wikipedia.org获取
4. *《Kinetis-M两相电表参考设计》*（文档DRM149）
5. *《MKM34Z256单相电表参考设计》*（文档DRM163）

8 修订历史记录

表5. 修订历史记录

修订版本号	日期	重大变更
0	12/2013	初始版本
1	11/2014	更新了章节5，“实践操作” 更新了章节6，“总结”
2	07/2015	更新了章节5.1，“METERLIBFFT_Interpolation”

附录A 在单相用户代码中使用插值函数的示例

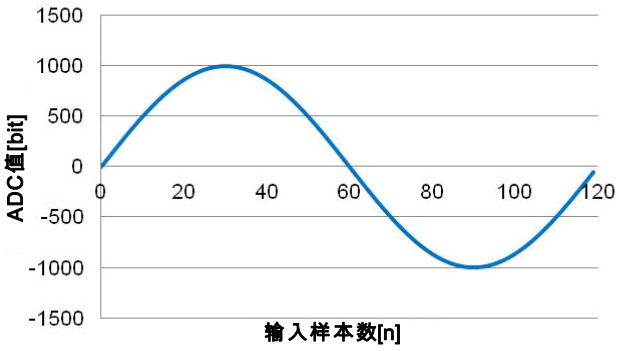
```
/*******************************************************************************************
 * (c) Copyright 2015, Freescale Semiconductor Inc.
 * ALL RIGHTS RESERVED.
*******************************************************************************************/
#include "fraclib.h"           /* fractional library header */
#include "meterlibfft.h"       /* metering library header */
#include "inputdata.h"         /* library of a typical periodic signals (LUT) */

/*******************************************************************************************
 * Buffers definitions                                                                     *
*******************************************************************************************/
/* multiplexed mandatory buffers (time domain / frequency domain in the Cartesian form) */
Frac24 u_re[120];     /* U-ADC output buffer/FFT real part output buffer */
Frac24 i_re[120];     /* I-ADC output buffer/FFT real part output buffer */

/* dedicated mandatory buffers (frequency domain in the Cartesian form) */
Frac24 u_im[SAMPLES64];     /* U-FFT imaginary part output buffer */
Frac24 i_im[SAMPLES64];     /* I-FFT imaginary part output buffer */

/*******************************************************************************************
 * Variables definitions                                                                   *
*******************************************************************************************/
tMETERLIBFFT1PH_DATA ui;              /* 1-PH main metering structure */
long fcn_out;                         /* metering function output or function error state */
long *pu,*pi;                         /* pointers to LUTs */

/*******************************************************************************************
 * Main                                                                                    *
*******************************************************************************************/
void main (void)
{
    /* Mandatory initialization section - for main FFT calculation */
    fcn_out = METERLIBFFT1PH_InitParam(&ui, SAMPLES64, SENS_PROP, IMP5000, IMP5000, EN_RES10);
    METERLIBFFT1PH_InitMainBuff(&ui, u_re, i_re, u_im, i_im, NULL);
    fcn_out = METERLIBFFT1PH_SetCalibCoeff(&ui, 325.27, 141.422, NULL, 0, 0);

    /* ADC sampling simulation (function fills-up both U and I buffers) */
    pu = sin_120s_6e6_5h_10p; /* set pointer to the beginning of U-LUT */
    pi = sin_120s_4e6_5h_40p; /* set pointer to the beginning of I-LUT */
    for (unsigned long cnt = 0; cnt < 120; cnt++)
    {
        u_re[cnt] = (*pu++);        /* copy U-binary values from LUT to the U-output buffer */
        i_re[cnt] = (*pi++);        /* copy I-binary values from LUT to the I-output buffer */
    }

    /* performs interpolation only for asynchronous processing */
    METERLIBFFT1PH_Interpolation(&ui, ORD2, ORD2, 120);

    /* main calculation (FFT, I-signal conditioning, scaling, averaging) */
    METERLIBFFT1PH_CalcMain(&ui);
    while (1);
}
/*******************************************************************************************
 * End of module                                                                           *
*******************************************************************************************/
```

附录B 仿真结果 – 示例1

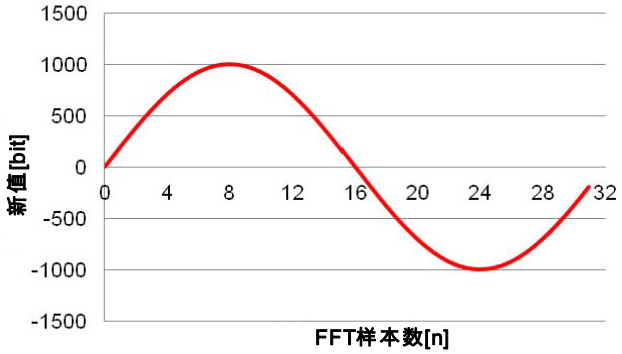
表6. 示例1的参数表

输入信号	仅正弦一次谐波
输入样本数	120
所需的FFT样本数	32
插值比	120/32 = 3.75
等效计量用例	fADC = 6.144 MHz， OSR = 1024，采样率 = 6144000/1024 = 6 KHz， fINP = 50 Hz

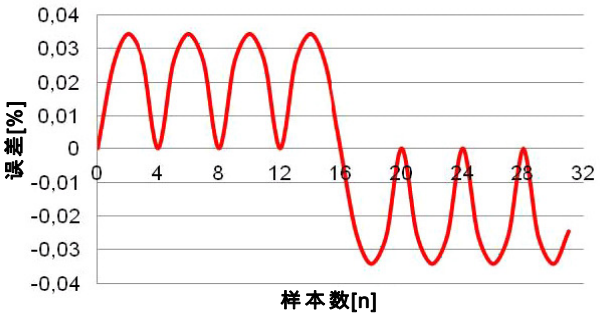
输入信号曲线



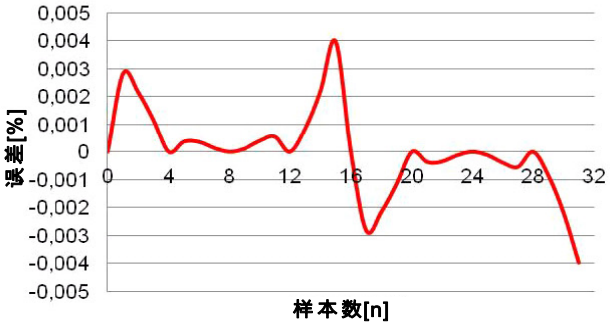
典型插值曲线



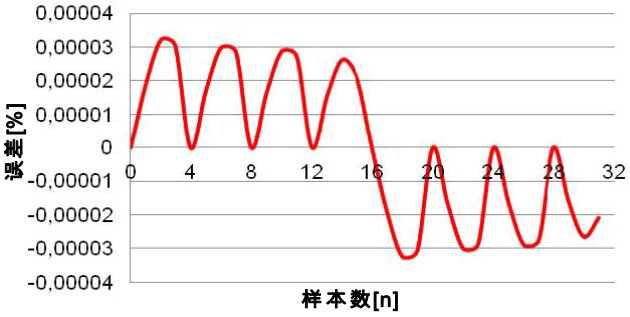
线性插值误差曲线



二次插值误差曲线



三次插值误差曲线

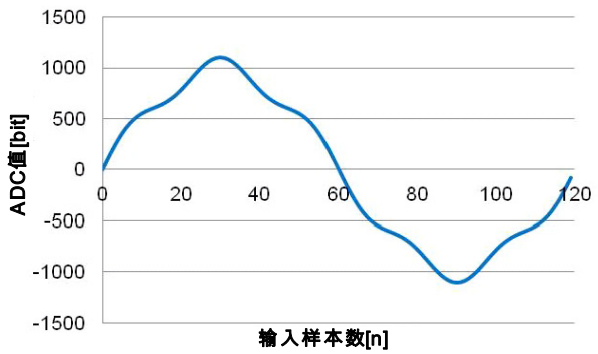


附录C 仿真结果 – 示例2

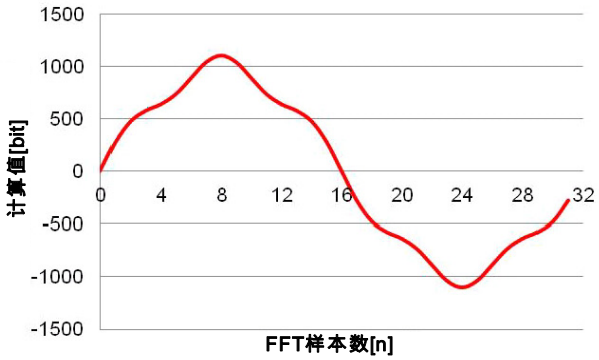
表7. 示例2的参数表

输入信号	正弦一次谐波 + 正弦五次谐波 ， 调制深度为10%
输入样本数	120
所需的FFT样本数	32
插值比	120/32 = 3.75
等效计量用例	fADC = 6.144 MHz， OSR = 1024，采样率 = 6144000/1024 = 6 KHz， fINP = 50 Hz

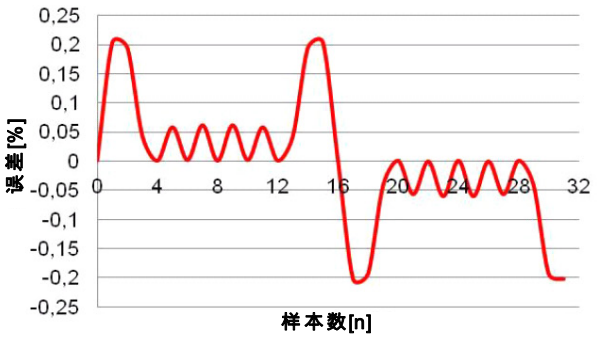
输入信号曲线



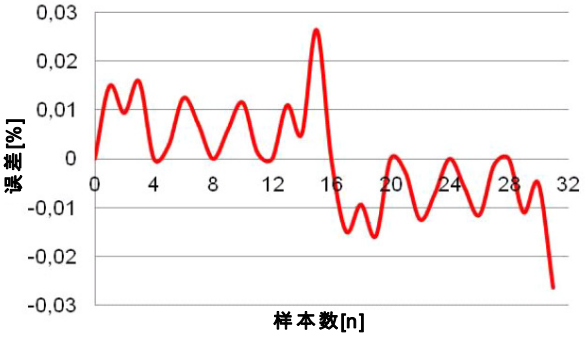
典型插值曲线



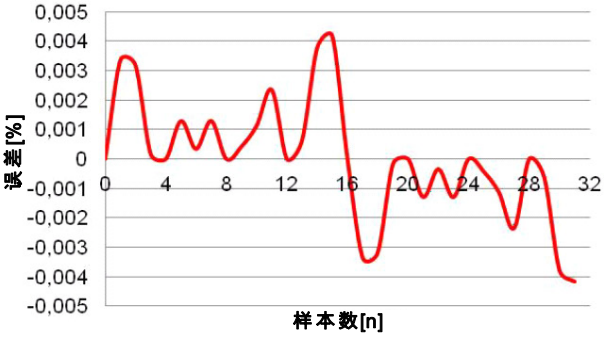
线性插值误差曲线



二次插值误差曲线



三次插值误差曲线

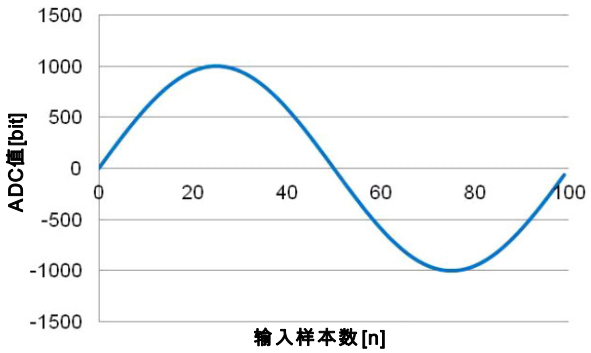


附录D 仿真结果 – 示例3

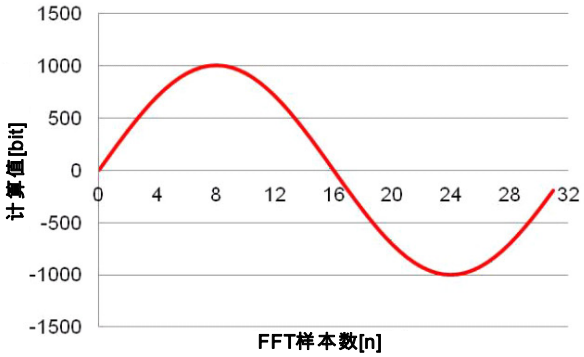
表8. 示例3的参数表

输入信号	仅正弦一次谐波
输入样本数	101
所需的FFT样本数	32
插值比	101/32 = 3.15625
等效计量用例	f_{ADC} = 6.144 MHz， OSR = 1024，采样率 = 6144000/1024 = 6 KHz， f_{INP} = 59.4 Hz

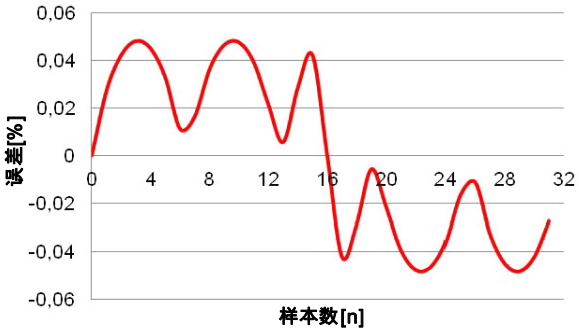
输入信号曲线



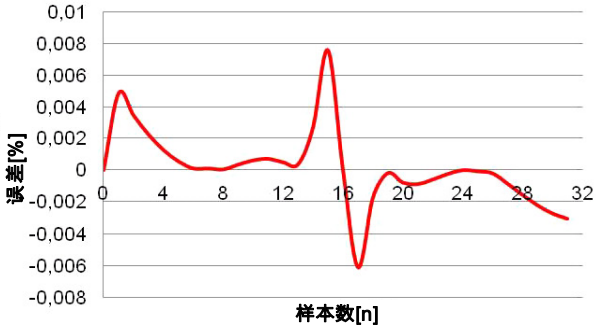
典型插值曲线



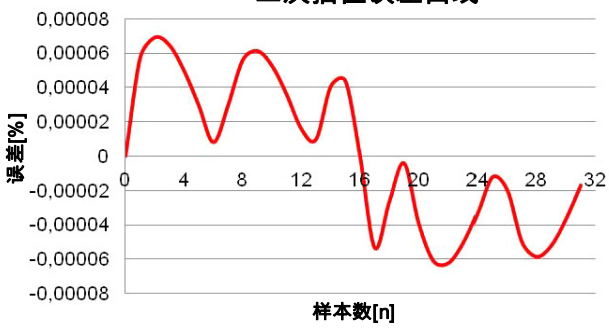
线性插值误差曲线



二次插值误差曲线



三次插值误差曲线

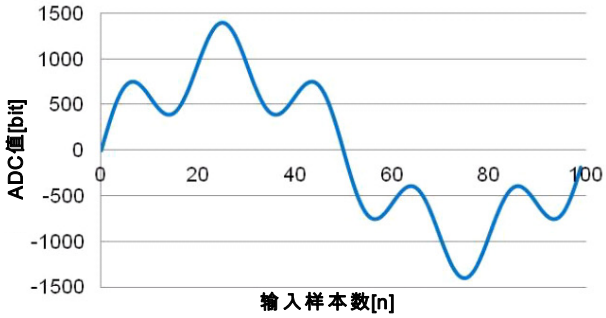


附录E 仿真结果 – 示例4

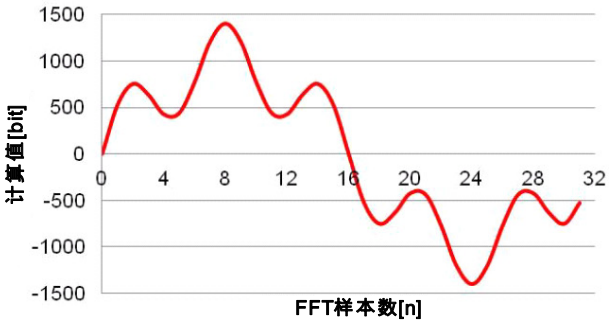
表9. 示例4的参数表

输入信号	正弦一次谐波 + 正弦五次谐波，调制深度为40%
输入样本数	100
所需的FFT样本数	32
插值比	100/32 = 3.125
等效计量用例	fADC = 6.144 MHz， OSR = 1024，采样率 = 6144000/1024 = 6 KHz， fINP = 60 Hz

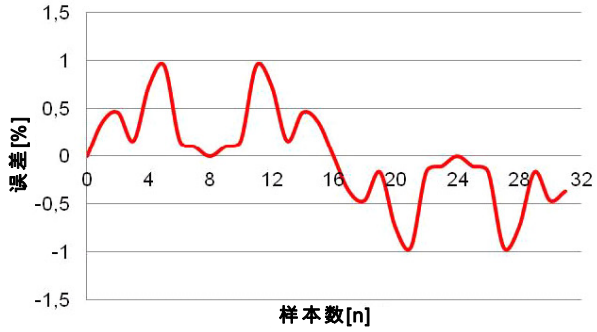
输入信号曲线



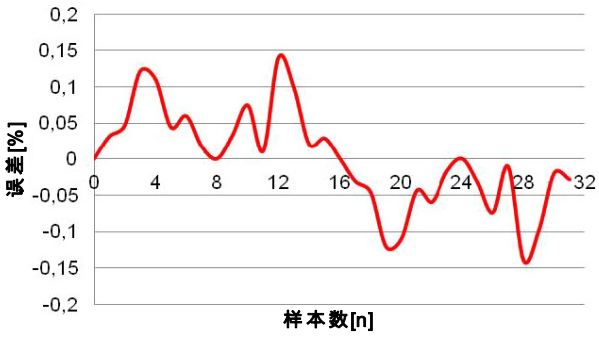
典型插值曲线



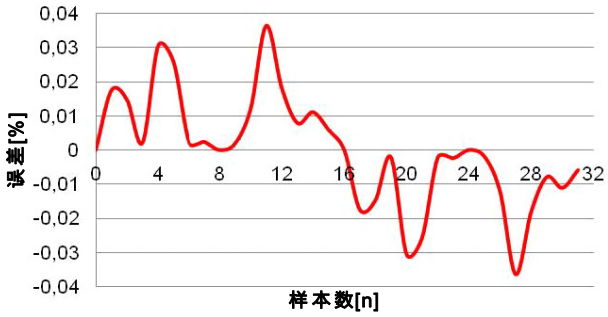
线性插值误差曲线



二次插值误差曲线



三次插值误差曲线



How to Reach Us:

Home Page:

Freescale.com

Web Support:

Freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不局限于后果性的或附带性的损害在内的所有责任。Freescale 的数据表和/或规格中所提供的"典型"参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括"经典值"在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件: freescale.com/SalesTermsandConditions.

Document Number: AN4847
Rev 2, 07/2015