

SPC1068

ADC 码值

校准指南

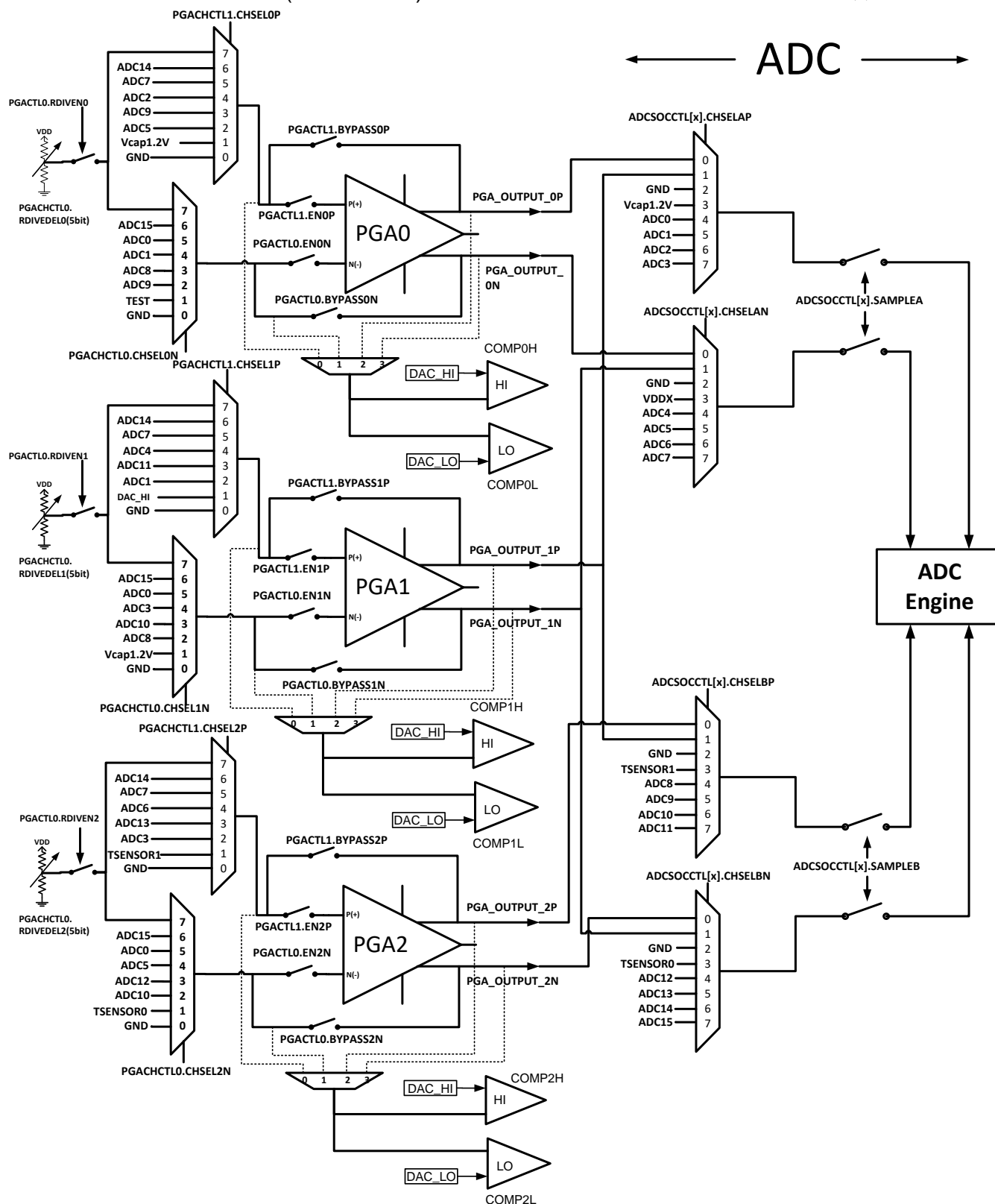
内容

1	INTRODUCTION	3
2	PGA 增益误差和失调	5
2.1	PGA 单元概述.....	5
2.2	PGA 的增益误差和失调	6
2.3	PGA 输出电压温漂	6
3	ADC 的增益误差和失调	6
3.1	ADC 单元概述.....	6
3.2	ADC 的增益误差和失调	6
3.3	ADC 输出码值温漂	7
4	增益误差和失调校准.....	8
4.1	ADC 输出码值校准	8
4.2	校准 ADC 函数.....	8
4.3	PGA 输出电压校准	9
4.4	校准 PGA 函数.....	9

1 Introduction

SPC1068 集成了 3 组增益可调放大器 (PGA) 和支持两路同步采样的 12 位模式转换器 (ADC)。

实际使用时, 输入信号先经由 PGA 放大, 然后送入 ADC 转化成数字码值。下图集成了 SPC1068 中三个重要的仿真器件之讯号流程图 (Signal Flow)。PGA 和 ADC 均存在增益误差 (Gain Error) 和失调误差 (Offset Error)。这些误差为线性误差, 可以通过校准消除。



实际应用时，随着温度变化，输入信号，PGA 和 ADC 的增益误差和失调误差均会变化，最终体现为 ADC 的输出码值温漂。在这里，我们仅讨论 PGA 和 ADC 的增益误差和失调误差温漂对输出结果的影响。输入信号的温漂与具体应用环境有变化，可以通过本文的分析方法做类似处理。

在阅读完本章后，您将可以了解到以下内容：

- (1) 信号经由 PGA 放大，然后送入 ADC 转化为数字码值。
- (2) PGA 和 ADC 存在失调误差和增益误差，这两个误差存在温漂，因此体现为 ADC 的输出结果会存在温漂。

2 PGA 增益误差和失调

2.1 PGA 单元概述

SPC1068 集成了 3 组可调增益的差分运放(Truly Differential PGA)，可抗拒同模噪声，亦可仿真为单端 PGA 使用。

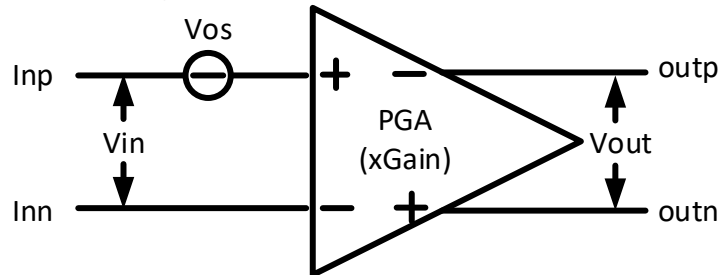
PGA 的基本特点如下：

- 差分模式时放大增益
4X 8X 16X 32X
- 单端模式时放大增益
2X 4X 8X 16X
(差分仿真为单端模式时，则与差分模式同)
- 弹性的信道选择
- 输出直接链接 ADC 采样通道
- 输入与输出均可输入比较器 COMP
- Slew Rate: 10V/us
- Unit Gain Band Width: 2MHz
- 建立时间 600ns (0.4V~VDDX-0.4V)
- PGA 的增益温漂 $\pm 57\text{ppm}/^{\circ}\text{C}$
- PGA 的失调温漂 $\pm 0.024\text{mV}/^{\circ}\text{C}$
在本章节中，将介绍
- 增益误差和失调误差对 PGA 输出结果的影响。
- 当温度变化时，PGA 输出变化。

2.2 PGA 的增益误差和失调

下图为 SPC1068 仿真电路构架，PGA 为下图套色部分

Figure 2-1 PGA Offset and Gain



假设 PGA 在差分模式，增益为 G_{PGA} ，增益误差为 ΔG_{PGA} ，失调为 OS_{PGA} ，输入信号为 V_{in} ，输出电压为 V_{out} 。所以存在误差的情况下，PGA 的输出 V_{out} 为：

$$V_{out} = G_{PGA} * (1 + \Delta G_{PGA}) * (V_{in} + OS_{PGA}) \quad (2-1)$$

假设差分增益为 4，增益误差 1%，失调为 1mV。输入差分电压 V_{in} 为 0.5V，则输出电压 V_{out} 为 2.02404V。这个结果相对于理想值 2V 存在 1.2% 的偏差。

2.3 PGA 输出电压温漂

实际环境中，PGA 的增益和失调都会温度变化。考虑温度变化的情况下，假设 PGA 的增益温漂为 TG_{PGA} 和失调温漂 TOS_{PGA} ，为公式 2-1 将变为：

$$V_{out} = G_{PGA} * (1 + \Delta G_{PGA}) * (1 + TG_{PGA}) * (V_{in} + OS_{PGA} + TOS_{PGA}) \quad (2-2)$$

假设增益温漂为 57ppm/C，失调温漂为 0.024mV/C。输入差分电压 V_{in} 为 0.5V，如果温度变化 50C，则输出电压 V_{out} 为 2.03467V。这个结果相对于常温结果 2.02404V 存在 0.53% 的偏差。

3 ADC 的增益误差和失调

3.1 ADC 单元概述

SPC1068 集成了 13-bit，4Msps 差分输入模数转换器（ADC）。该 ADC 也可以单端输入使用，此时为 12-bit。

ADC 的基本特点如下：

- 差分和单端模式的精度分别为 13-bit/12-bit
- 单次转换时间为 125ns
- 16 个独立的外部模拟输入信号通道（不包含 PGA 输入）
- ADC 的增益温漂 $\pm 50\text{ppm}/^\circ\text{C}$
- ADC 的失调温漂 $\pm 0.03\text{mV}/^\circ\text{C}$

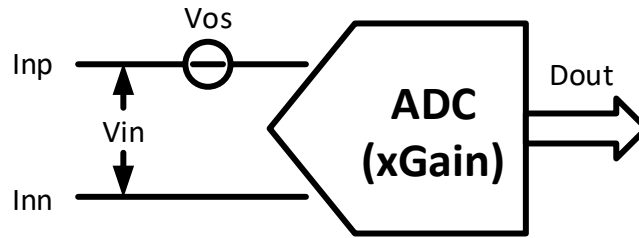
在本章节中，将介绍

- 增益误差和失调误差对 ADC 输出结果的影响。
- 当温度变化时，ADC 输出变化。

3.2 ADC 的增益误差和失调

下图为 SPC1068 仿真电路构架，ADC 为下图套色部分

Figure 3-1 ADC Offset and Gain



ADC 增益理想情况为 1，增益误差为 ΔG_{ADC} ，失调为 OS_{ADC} ，输入信号为 V_{in} ，输出码值为 D_{out} 。所以存在误差的情况下，ADC 的输出 D_{out} 为：

$$D_{out} = (1 + \Delta G_{ADC}) * (V_{in} + OS_{ADC} + 3.657) / 3.657 * 4096 \quad (3-1)$$

假设 ADC 的增益误差 1%，失调为 1mV。输入差分电压 V_{in} 为 1V，则输出码值 D_{out} 为 5228。这个结果相对于理想值 5216 存在 0.23% 的偏差。

3.3 ADC 输出码值温漂

实际环境中，ADC 的增益和失调都会温度变化。考虑温度变化的情况下，假设 ADC 的增益温漂为 TG_{ADC} 和失调温漂 TOS_{ADC} ，为公式 3-1 将变为：

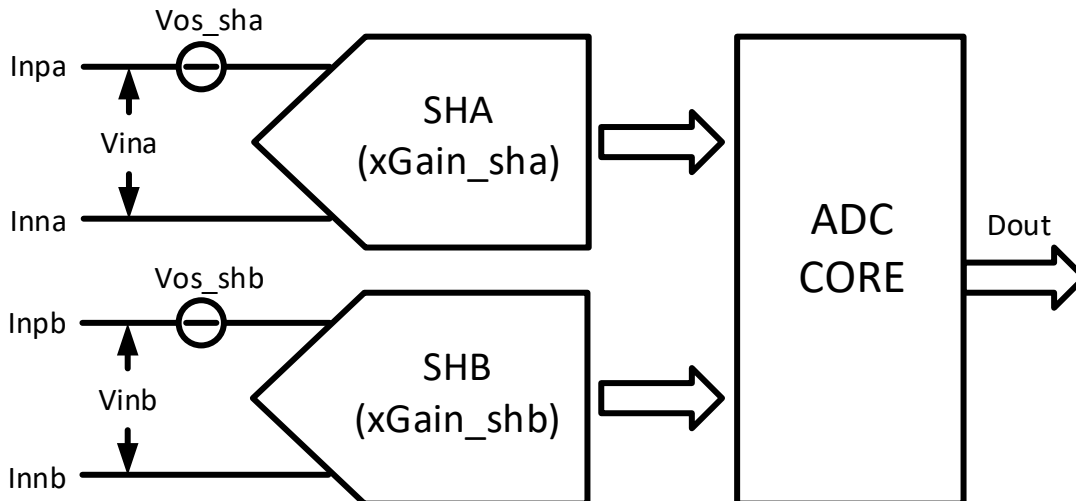
$$D_{out} = (1 + \Delta G_{ADC}) * (1 + TG_{ADC}) * (V_{in} + OS_{ADC} + TOS_{ADC} + 3.657) / 3.657 * 4096 \quad (3-2)$$

假设增益温漂为 50ppm/C，失调温漂为 0.03mV/C。输入差分电压 V_{in} 为 1V，如果温度变化 50C，则输出码值 D_{out} 为 5232。这个结果相对于常温结果 5228 存在 0.08% 的偏差。

4 增益误差和失调校准

4.1 ADC 输出码值校准

由 2.2 和 2.3 可知，由于增益误差，失调，以及温度变化，ADC 输出结果存在偏差，导致了信号链的准确度，影响客户应用。下面介绍校准这些误差的方法，客户可以根据需要在软件里面校准这个误差，或者直接使用 Spintrol 提供的校准函数库对该误差进行校准。



在 SPC1068 内部，有两个采样保持器 SHA 和 SHB，他们的增益误差和失调不同，需要分别校准。我们利用 ADCOFFSETA/ADCGAINA 存储 SHA 的失调和增益校准参数，利用 ADCOFFSETB/ADCGAINB 存储 SHB 的失调和增益校准参数。

为了校准 SHA 和 SHB，需要分别测量他们的增益误差和失调。

测量 SHA 增益误差和失调校准参数的过程如下：

第一步，SHA 正端输入和负端输入到 AGND。用 ADC 测到的输出码值为 D_{out1}。

第二步，SHA 正端输入 3.3V，负端输入 AGND。用 ADC 测到的输出码值为 D_{out2}。

$$D1(0V) = 4096 = (1 + \Delta G_{ADC_SHA}) * (D_{out1} + OS_{ADC_SHA}) \quad (3-1)$$

$$D2(3.3V) = 7792 = (1 + \Delta G_{ADC_SHA}) * (D_{out2} + OS_{ADC_SHA}) \quad (3-2)$$

$$\Delta G_{ADC_SHA} = 3696 / (D_{out2} - D_{out1}) - 1 \quad (3-3)$$

$$OS_{ADC_SHA} = 7792 / (1 + \Delta G_{ADC_SHA}) - D_{out2} \quad (3-4)$$

将 OS_{ADC_SHA} 存入 ADCOFFSETA。ΔG_{ADC_SHA} 是小数，将其左移 15 位，存入 ADCGAINA。

SHB 的测量过程和 SHA 类似，最终将 OS_{ADC_SHB} 存入 ADCOFFSETB，将 ΔG_{ADC_SHB} 左移 15 位存入 ADCGAINB。

得到校准参数以后，在对于任意的 D_{out}，只需要按照如下公式计算，即可得到正确结果。

对于 SHA 输入的信号有

$$D_{OUT_SHA_CAL} = (1 + \Delta G_{ADC_SHA}) * (D_{OUT_SHA} + OS_{ADC_SHA}) \quad (3-5)$$

对于 SHB 输入的信号有

$$D_{OUT_SHB_CAL} = (1 + \Delta G_{ADC_SHB}) * (D_{OUT_SHB} + OS_{ADC_SHB}) \quad (3-6)$$

4.2 校准 ADC 函数

芯片在出厂前，Spintrol 对每一颗 ADC 进行校准，将校准参数存储在 ADCOFFSETA/ADCGAINA 和 ADCOFFSETB/ADCGAINB 中。同时 Spintrol 也提供了 ADC 校准函数

库，客户只要根据要求配置 ADC 的输入通道，调用校准函数，即可得到校准参数。ADC 的通道配置如下：

Example2.2

```
void ADC_Calibration_Example(void)
{
    /* Select PGA input as analog pin */
    GPIO_SetPinAsAnalog(GPIO_1); /* ADC0 输入 3.3V */
    GPIO_SetPinAsAnalog(GPIO_2); /* ADC1 输入 0V */

    /* Run ADC calibration function */
    ADC_Cal(void);
}
```

4.3 PGA 输出电压校准

由 2.2 和 2.3 可知，由于增益误差，失调，以及温度变化，PGA 输出结果存在偏差，导致了信号链的准确度，影响客户应用。下面介绍校准这些误差的方法，客户可以根据需要在软件里面校准这个误差，或者直接使用 Spintrol 提供的校准函数库对该误差进行校准。

由于 PGA 需要使用 ADC 测量，所以首先要对 ADC 进行校准，得到 ADC 的校准参数。每颗芯片在出厂前会进行 ADC 的校准，所以下面的校准过程假设 ADC 的输出是已经校准后的结果。

PGA 在差分模式，共模输入端设为负输入端。增益为 G_{PGA} ，增益误差为 ΔG_{PGA} ，失调为 OS_{PGA} ，正端输入信号为 V_{in} ，负端输入为 V_{inn} ，正端输出电压为 V_{outp} ，负端输出电压为 V_{outn} 。测量 PGA 增益误差和失调校准参数的过程如下：

第一步，PGA 设为 bypass 模式，在 PGA 正端和负端同时输入 $VDD/2$ ，用 ADC 测到结果为 $Dout_byp1$ 。

第二步，PGA 设为放大模式，在 PGA 正端和负端同时输入 $VDD/2$ 。用 ADC 测到的输出码值为 $Dout_amp1$ 。

第三步，PGA 设为 bypass 模式，在 PGA 正端输入 $VDD/2+V_{in}$ ，负端输入 $VDD/2$ 。用 ADC 测到的输出码值为 $Dout_byp2$ 。

第四步，PGA 设为放大模式，在 PGA 正端输入 $VDD/2+V_{in}$ ，负端输入 $VDD/2$ 。用 ADC 测到的输出码值为 $Dout_amp2$ 。

得到 $Dout_byp1$ ， $Dout_amp1$ ， $Dout_byp2$ 和 $Dout_amp2$ 以后按照如下公式计算校准参数：

$$Dout_amp1 = Gain * (1 + \Delta G_{PGA_SHA}) * (Dout_byp1 + OS_{PGA_SHA}) \quad (3-1)$$

$$Dout_amp2 = Gain * (1 + \Delta G_{PGA_SHA}) * (Dout_byp2 + OS_{PGA_SHA}) \quad (3-2)$$

$$\Delta G_{PGA_SHA} = (Dout_amp2 - Dout_amp1) / (Dout_byp2 - Dout_byp1) / Gain - 1 \quad (3-3)$$

$$OS_{PGA_SHA} = Dout_amp2 / (1 + \Delta G_{PGA_SHA}) / Gain - Dout_byp2 \quad (3-4)$$

将 OS_{PGA_SHA} 和 ΔG_{PGA_SHA} 存为常量，在后续 ADC 输出结果中校准 PGA。

$$DOUT_PGA_CAL = (1 + \Delta G_{PGA_SHA}) * (DOUT_PGA + OS_{PGA_SHA}) \quad (3-5)$$

4.4 校准 PGA 函数

客户可以根据 4.3 提供的方法得到 PGA 的校准参数，Spintrol 也提供了 PGA 校准函数，客户只要根据自己的需求配置该函数，即可得到相应的校准参数。该函数配置实列如下：

Example2.2

```
void PGA_Caliration_Example(void)
{
    Int pga_gain_cal;
    Int pga_offset_cal;

    /* Run PGA calibration function */
    PGA_Cal(PGA_Diff,PGA_Gain,PGA_Com);
}
```