Application Note

SPC1068 PGA 和 COMP 使用指南

Revision 1 – September 2017

目录

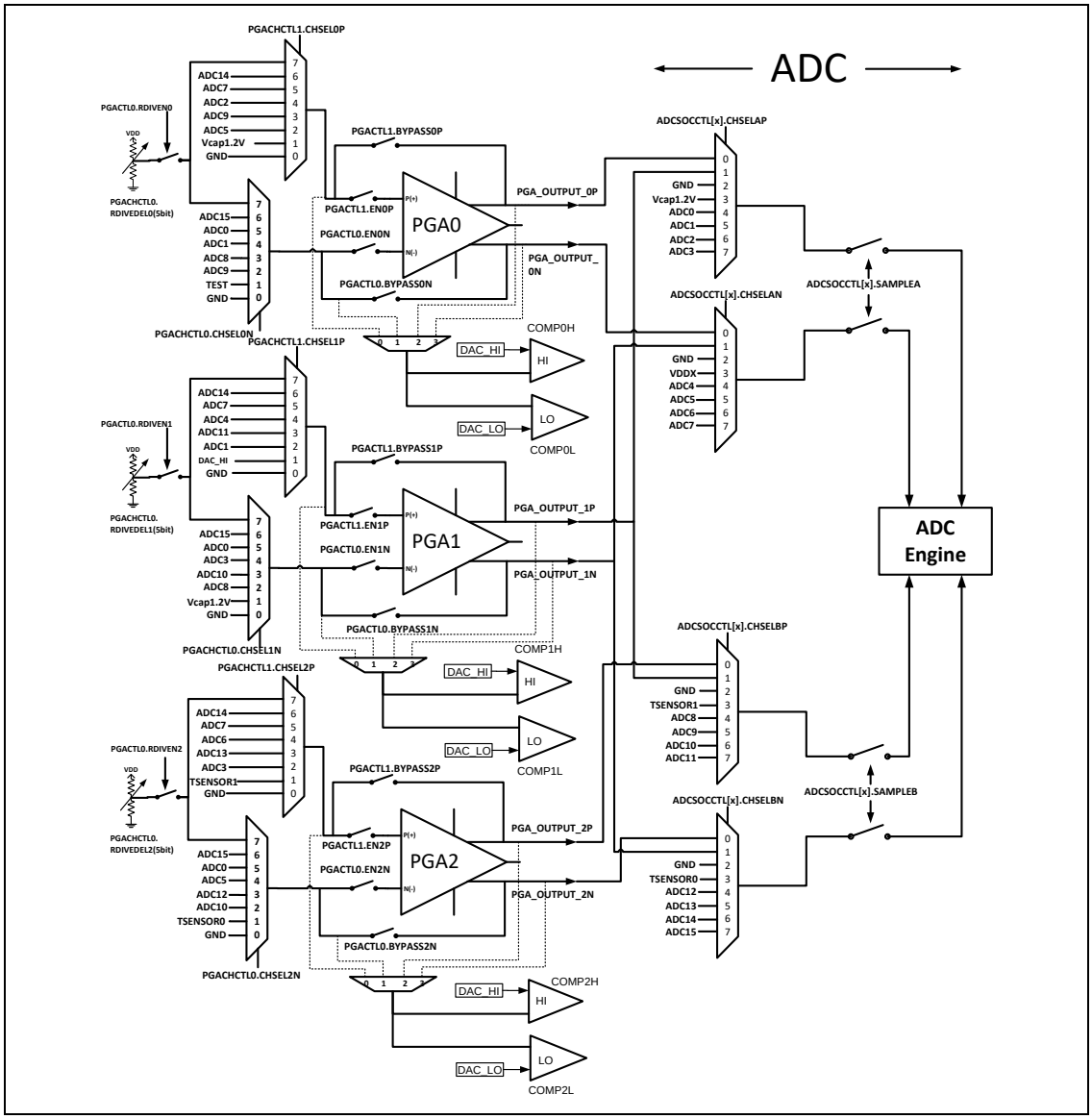
表格列表

图片列表

1 概述

SPC1068 内部集成了 3 组可编程增益的内部运放（PGA），每一组 PGA，配置两个比较器（Comparator），提供讯号 Too High 与 Too Low 时的两种保护，共六个比较器。每一个比较器又可以致使 PWM 输出停止（PWM trip zone），可轻易达成过电流防护功能，下图集成了 SPC1068 中三个重要的仿真器件之讯号流图（Signal Flow）。

图 1-1. 仿真器件信号流图



在阅读完本章后，您将可以了解：

（1） 如何利用差分 PGA 放大电流讯号；

（2） 如何使用 COMP 侦测过电流讯号，以及如何利用此过电流讯号停止 PWM。

2 PGA 使用教学

差分 ADC 可配置成常见的单端 ADC 使用，建议采用 SDK 内提供的 EasyInit1（）函数进行设定，SPC1068 集成了 3 组可调增益的差分运放（Truly Differential PGA ），可抗拒同模噪声干扰，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）

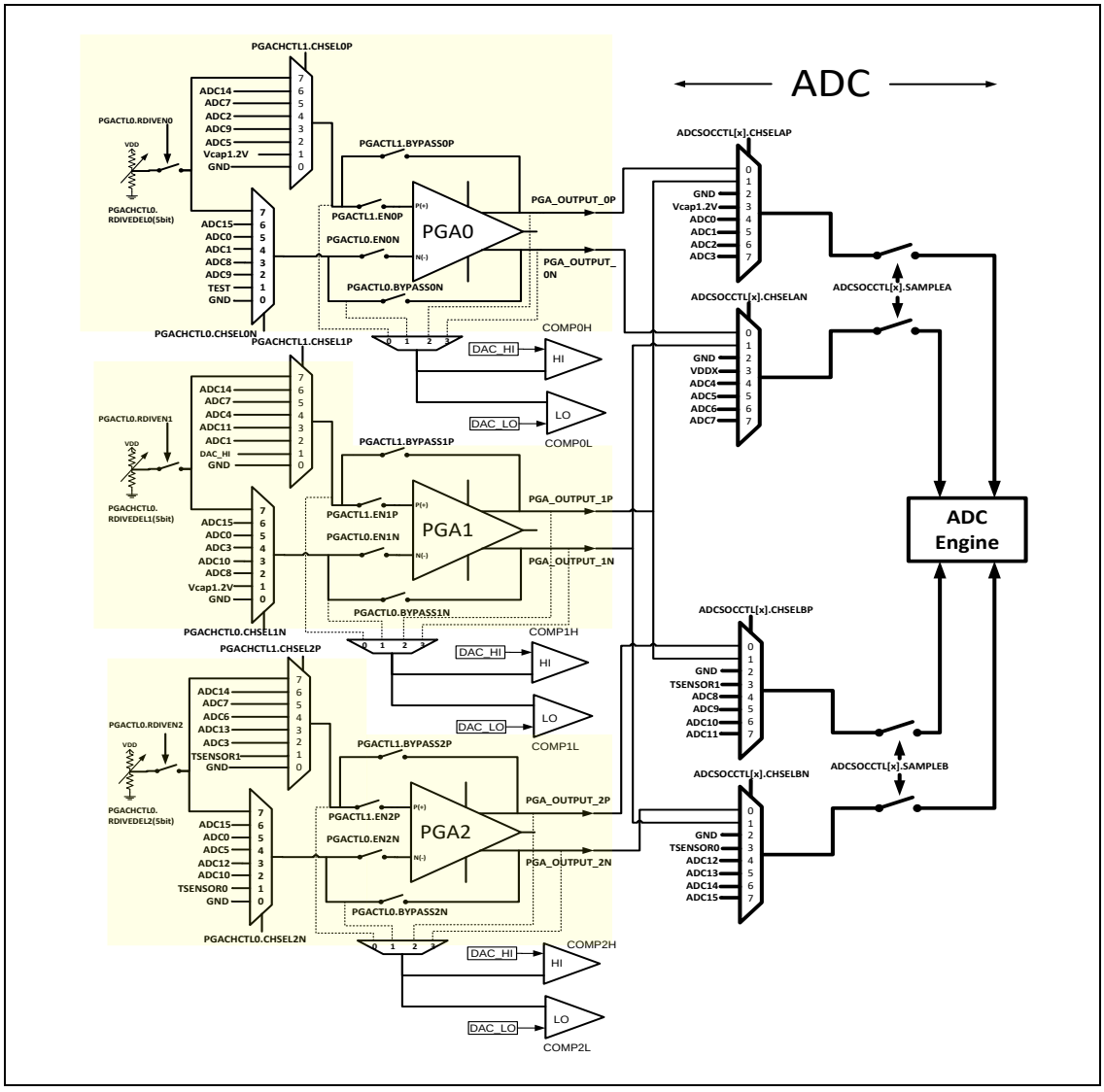
在本章节中，将介绍：

- 实际放大电流范例，利用差分模式放大电流
- MCU 建议的脚位排法
- PGA SDK API 参考表

2.1 PGA 单元概述

下图为 SPC1068 仿真电路构架，PGA 为下图套色部分。

图 2-1. PGA in Analog Signal Structure



2.2 利用差分 PGA 放大电流信号

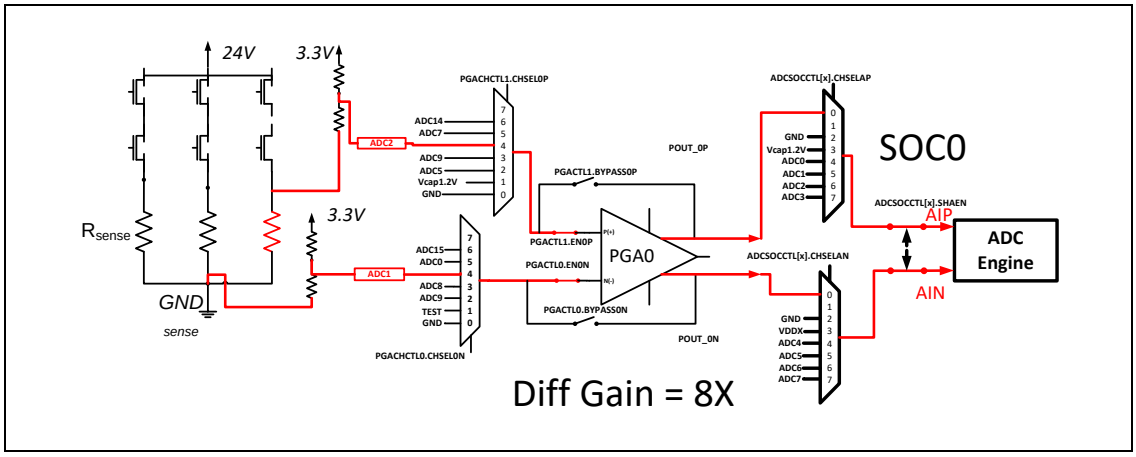
Spintrol 提供了 PGA 驱动函数，只需定义好通道选择就可以使用。下面是一个利用 PGA 放大 shunt resistor 上电阻的实际范例。

Example Code

```
void PGA_InitExample2_2(void)
{
  /* Select PGA input as analog pin */
  GPIO_SetPinAsAnalog(GPIO_2); /* ADC1 */
  GPIO_SetPinAsAnalog(GPIO_3); /* ADC2 */
  /* Init PGA0 */
  PGA_DiffInit( PGA0,
            PGA0_CH_P_ADC2 /* Positive CH   */,
            PGA0_CH_N_ADC1 /* Negative CH   */,
            PGA_SCALE_8X   /* PGA Diff Gain */);
  /* Init ADC in 2 channel mode */
  ADC_EasyInit2(ADC_SOC_0,ADCx_PGA0P,ADCx_PGA0N,ADCTRIG_Software);
}
```

请注意，使用 PGA 之前请先将管脚初始化为模拟管脚。上述代码对应于实际硬件配置如图 2-2 所示。必须将信号的正端与负端均偏值到 VDD/2 的地方，请注意以下的电路，信号在偏值时会变成 1/2 倍，请见后方信号流程细节。

图 2-2. PGA 放大信号示意图



以下以一个例子分析内部电压。请注意：

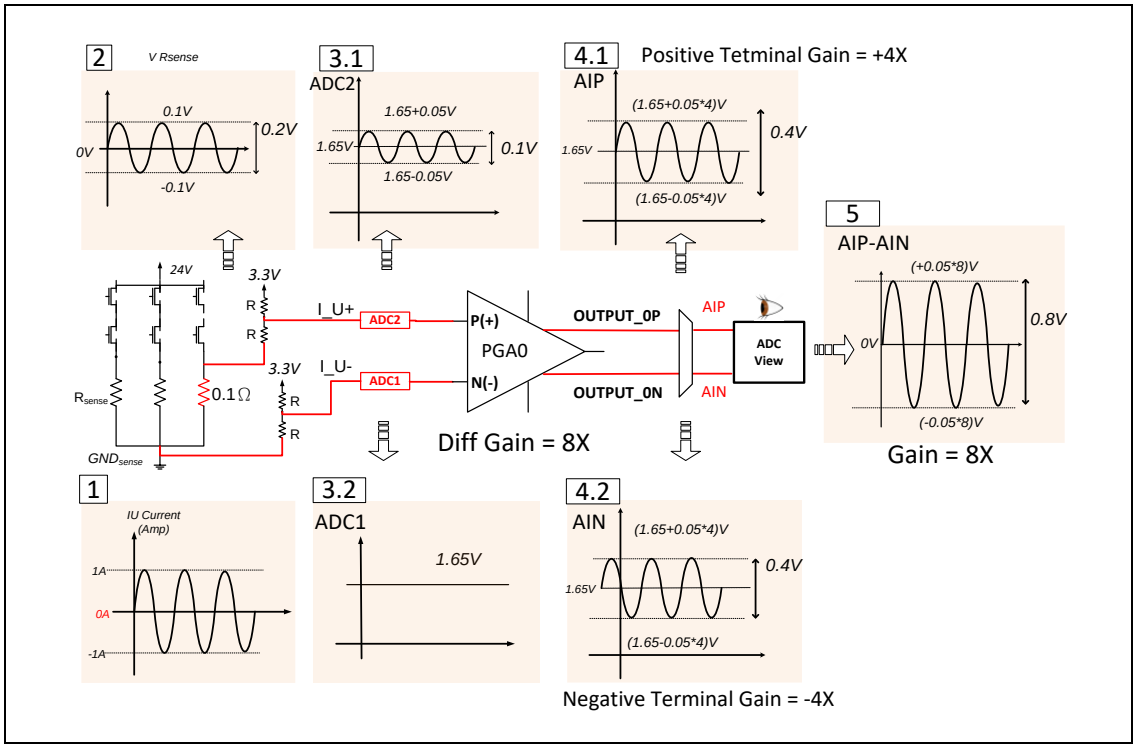
（1） 假设 shunt resistor 为 0.1 欧姆，电流振幅为 1A

（2） 设定的差分放大增益是 8X

（3） 请注意电压在不同节点的摆幅

（4） R 建议为 10K 欧姆

图 2-3. PGA 差分模式放大信号细节流程图



以下讲解讯号放大之细部流程，编号对应上图。

（1） 假设 U 相电流为 1A 振幅之 sine 波

（2） 此电流通过采样电阻（0.1Ω）后，可得到 0V 附近摆动，振幅为 0.1V 的电压 sine 波。电流讯号（V Rsense）与地（GND）在进入 MCU 前，务必抬升到 VDD/2=1.65V。

（3） 1. 在进入芯片前，请用两个电阻将电压抬升到 VDD/2，也就是说在进入芯片前，必须要将正端讯号拉升到 VDD/2 附近，否则 PGA 无法正常以差分模式进行放大，推荐的拉升电阻为 10K 奥姆。注意此时的电压摆幅为 0.05V。

2. 因为本范例是差分讯号放大，所以必须将 GND 讯号也做一个拉升，如上图所示。也就是说在进入芯片前，负端讯号也必须拉升到 VDD/2，以本例来说，即在 1.65V。

（4） 1. 本范例设定的 PGA 差分增益为 8X，实际上芯片内部是以两个 OP 实现之，内部两个 OP 的增益都是 4X，负责放大正端讯号的 OP 是 4X，负责放大负端讯号的 OP 是-4X。必须注意的是正端与负端的放大倍率会差一个负号。可以发现讯号在 PGA 正端的输出仍是在 1.65V 附近摆荡，但是幅值已经达到了 4 倍，也就是 0.05V*4=0.2V，峰峰值为 0.4V。

2. 注意负责放大负端讯号的 OP 是-4X。所以他会是一个输出在 1.65V 附近摆荡，振幅放大四倍的讯号，但是他的相位会跟正端讯号刚好相反。

（5） ADC 选择输入为 PGA 的正端与负端讯号，因此以 ADC 的角度来看，其实是看到

AIP-AIN = （1.65+0.05V*4） – （1.65 - 0.05V*4） = 0.05*（4+4） =0.05*8 =0.4V

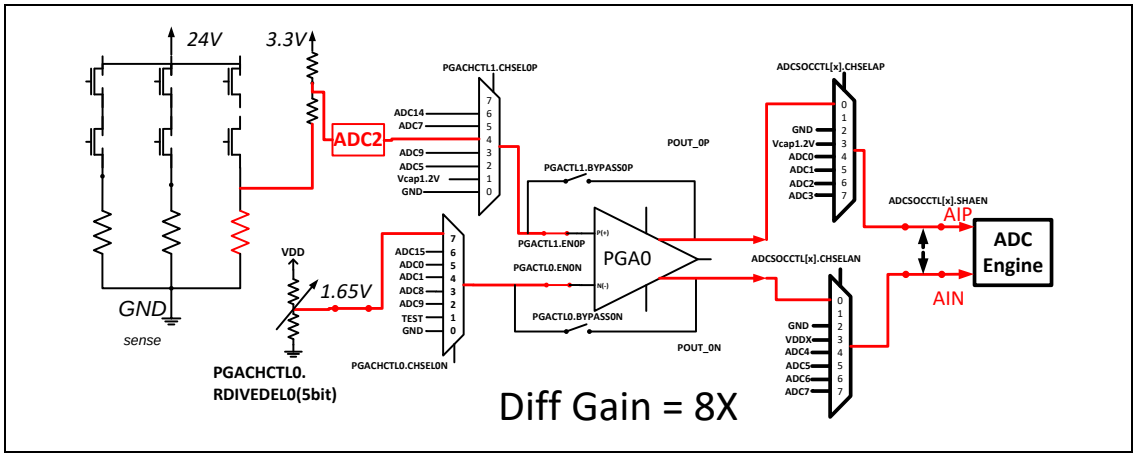
的讯号，也就是说在 ADC 端，看的到讯号是一个在 0V 附近摆荡，但是振幅已经来到了 8 倍。

（6） 从 Shunt resistor 的信号出发，到 ADC 所看到的讯号，实际上是放大了 4 倍，这是因为偏值到 VDD/2 之后，信号会变为 1/2。

2.3 差分 PGA 模拟为单端 PGA 使用

由上一章节可知，差分 PGA 的负端讯号其实是需要一个 1.65V 的讯号，但若是同模噪声在应用中不明显，使用者可以透过芯片内部的电阻分压器制造此 1.65V，可以节省 PIN 脚的使用。

图 2-4. PGA 单端模式示意图



Example Code

```
void PGA_InitExample2_3(void)
{
  /* Select PGA input as analog pin */
  GPIO_SetPinAsAnalog(GPIO_3); /* ADC2 */
  /* Init PGA0 */
  PGA_DiffInit( PGA0,
            PGA0_CH_P_ADC2     /* Positive CH          */,
            PGA0_CH_N_RESDIV0 /* Negative CH :
                                 Inner Resistor Divider */,
            PGA_SCALE_8X       /* PGA Diff Gain        */);
  /* Config resistor divider */
  PGA_ResistorDividerConifg(PGA0,16 /* 1.65V --> 5 bit : 0 is 0V and 31
is 3.3V*/ );
  /* Init ADC in 2 channel mode */
ADC_EasyInit2(ADC_SOC_0,ADCx_PGA0P,ADCx_PGA0N,ADCTRIG_Software);
}
```

请注意 PGA 的负端输入通道，以改为电阻分压之输出，电阻分压设定为 16（满幅为 5bit=31），可输出 1.65V。

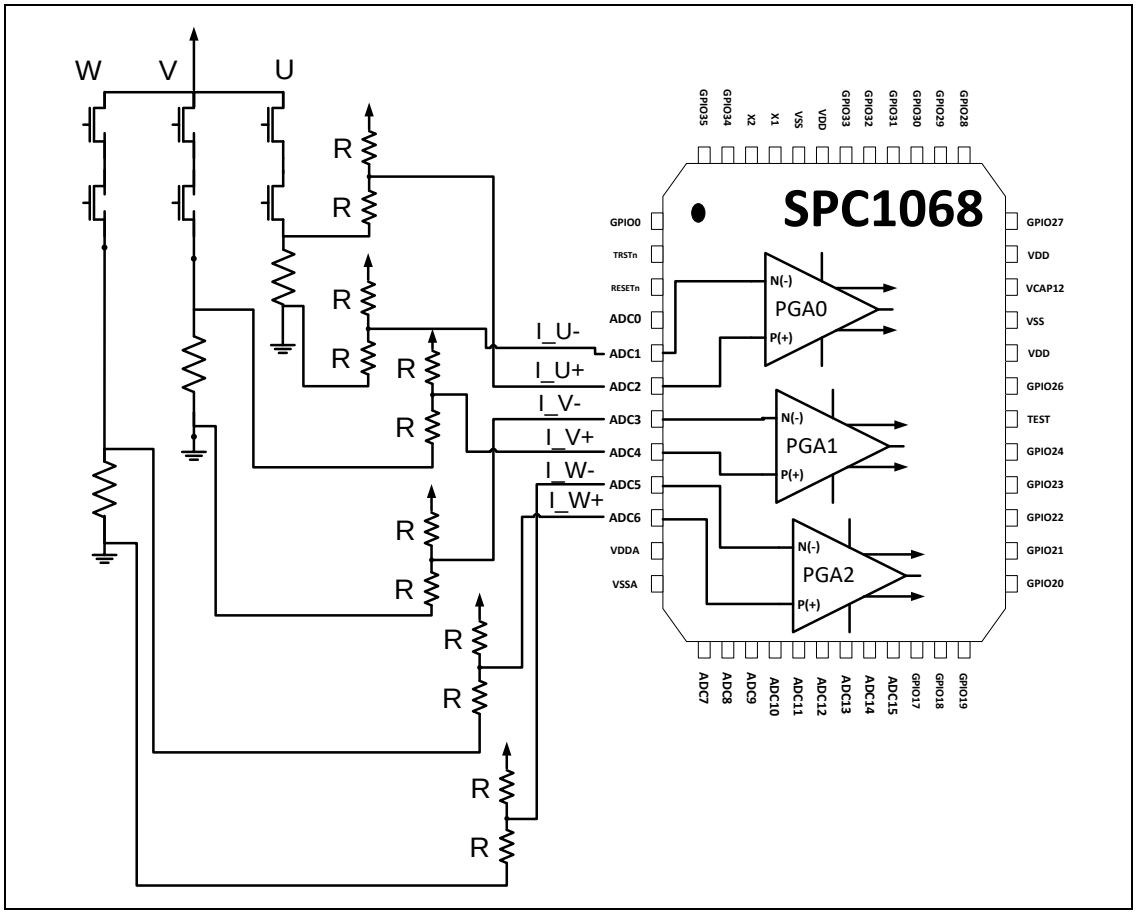
2.4 MCU 管脚建议排法

SPC1068 的 PGA 信道配置弹性，用户可根据需求自行调配，但建议可根据以下配置进行不同采样应用的搭配。

（1）3 shunt 电阻采样，真正差分讯号

在这个范例中，每一个相电流采样电阻的负端讯号（GND）也需要一个 Pin 进行放大。

图 2-4. 3 shunt 电阻差分采样示意图



若有滤波需求，可在进入 MCU 前加上一个电容。图 2-4 对应的范例代码如下：

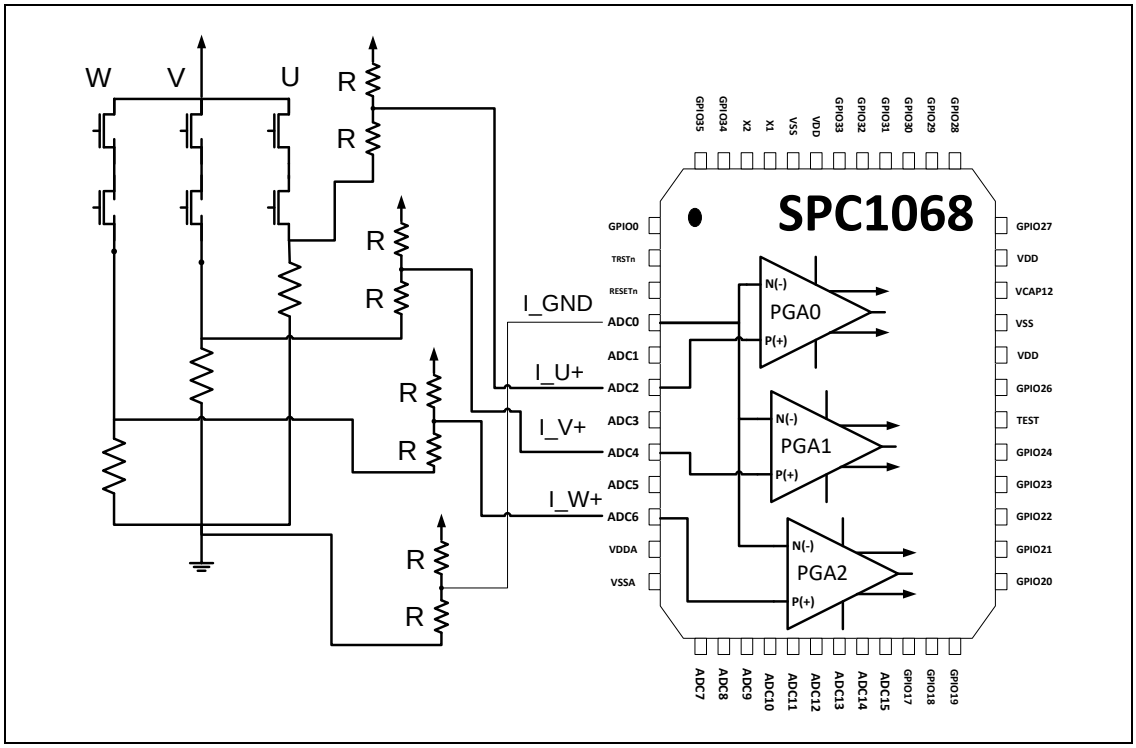
Example Code

```
void PGA_InitExample2_4_1(void)
{
  /* Select PGA input as analog pin */
  GPIO_SetPinAsAnalog(GPIO_2); /* ADC1 I_U- */
  GPIO_SetPinAsAnalog(GPIO_3); /* ADC2 I_U+ */
  GPIO_SetPinAsAnalog(GPIO_4); /* ADC3 I_V- */
  GPIO_SetPinAsAnalog(GPIO_5); /* ADC4 I_V+ */
  GPIO_SetPinAsAnalog(GPIO_6); /* ADC5 I_W- */
  GPIO_SetPinAsAnalog(GPIO_7); /* ADC6 I_W+ */
```

```
  /* Init PGA */
  PGA_DiffInit( PGA0,
          PGA0_CH_P_ADC2 /* Positive CH   */,
          PGA0_CH_N_ADC1 /* Negative CH   */,
          PGA_SCALE_8X   /* PGA Diff Gain */);
  PGA_DiffInit( PGA1,
          PGA1_CH_P_ADC4 /* Positive CH   */,
          PGA1_CH_N_ADC3 /* Negative CH   */,
          PGA_SCALE_8X   /* PGA Diff Gain */);
  PGA_DiffInit( PGA2,
          PGA2_CH_P_ADC6 /* Positive CH   */,
          PGA2_CH_N_ADC5 /* Negative CH   */,
          PGA_SCALE_8X   /* PGA Diff Gain */);
  /* Init ADC in 2 channel mode */
  ADC_EasyInit2(ADC_SOC_0,ADCx_PGA0P,ADCx_PGA0N,ADCTRIG_Software);
  ADC_EasyInit2(ADC_SOC_1,ADCx_PGA1P,ADCx_PGA1N,ADCTRIG_Software);
  ADC_EasyInit2(ADC_SOC_2,ADCx_PGA2P,ADCx_PGA2N,ADCTRIG_Software);
}
```

（2）3 shunt 电阻采样，共享 GND 讯号

图 2-5. 3 shunt 电阻共 GND 采样示意图



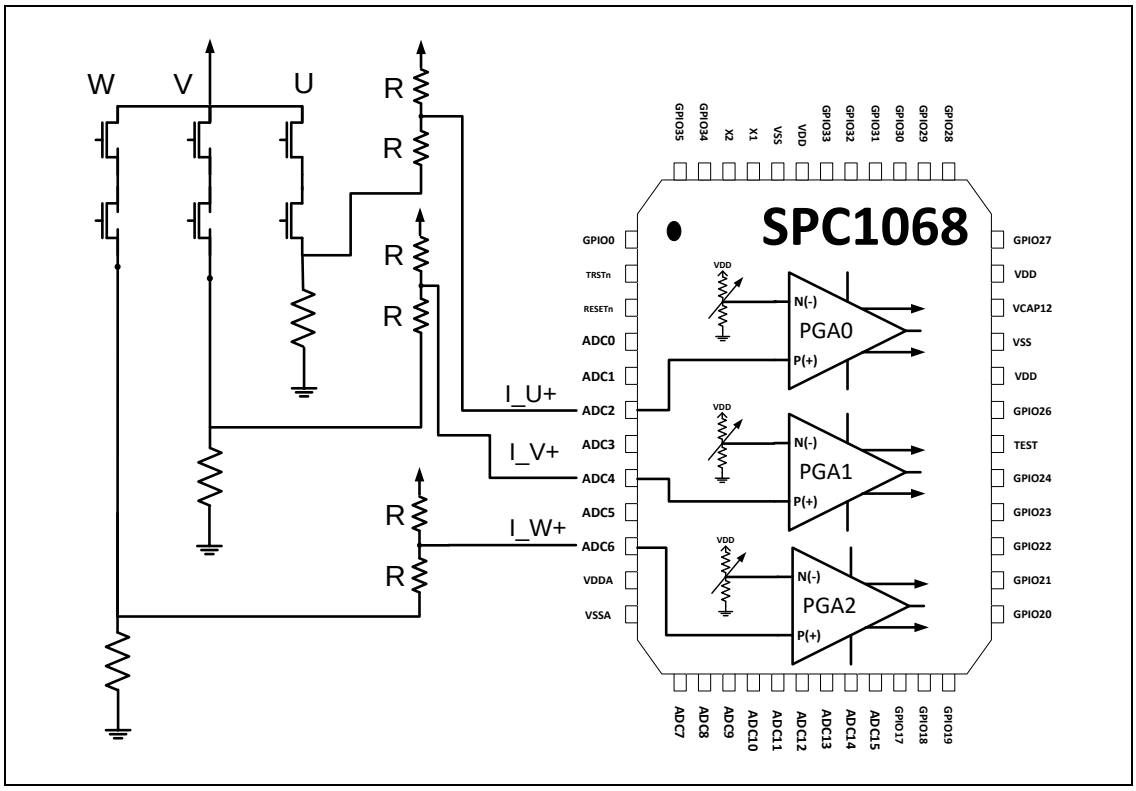
ADC0 是三个 PGA 负端输入都共有的端点，可作为三个讯号的共地使用。依照此接法可节省两个输入管脚与四个电阻，但是在 PCB Layout 上 V_GND 信号必须与三相电流讯号尽可能走相同路径，才能有较佳的共模噪声滤除效果。图 2-5 对应的范例代码如下：

Example Code

```
void PGA_InitExample2_4_2(void)
{
  /* Select PGA input as analog pin */
  GPIO_SetPinAsAnalog(GPIO_1); /* ADC0 Comman Negative */
  GPIO_SetPinAsAnalog(GPIO_3); /* ADC2 I_V+ */
  GPIO_SetPinAsAnalog(GPIO_5); /* ADC4 I_V+ */
  GPIO_SetPinAsAnalog(GPIO_7); /* ADC6 I_W+ */
  /* Init PGA */
  PGA_DiffInit( PGA0,
          PGA0_CH_P_ADC2 /* Positive CH   */,
          PGA0_CH_N_ADC0 /* Negative CH   */,
          PGA_SCALE_8X   /* PGA Diff Gain */);
  PGA_DiffInit( PGA1,
          PGA1_CH_P_ADC4 /* Positive CH   */,
          PGA1_CH_N_ADC0 /* Negative CH   */,
          PGA_SCALE_8X   /* PGA Diff Gain */);
  PGA_DiffInit( PGA2,
          PGA2_CH_P_ADC6 /* Positive CH   */,
          PGA2_CH_N_ADC0 /* Negative CH   */,
          PGA_SCALE_8X   /* PGA Diff Gain */);
  /* Init ADC in 2 channel mode */
  ADC_EasyInit2(ADC_SOC_0,ADCx_PGA0P,ADCx_PGA0N,ADCTRIG_Software);
  ADC_EasyInit2(ADC_SOC_1,ADCx_PGA1P,ADCx_PGA1N,ADCTRIG_Software);
  ADC_EasyInit2(ADC_SOC_2,ADCx_PGA2P,ADCx_PGA2N,ADCTRIG_Software);
}
```

（3）3 shunt 电阻采样，使用内部电压分压制造 VDD/2

图 2-6. 3 shunt 电阻内部电阻分压采样



使用芯片内部分压电阻作为负端讯号源，设定此分压电阻输出 1.65V，与上一章节有同样效果，依照此接法只需要三个管脚与六个讯号平移电阻，但是并无共模噪声滤除效果。以上配置请参考如下范例码：

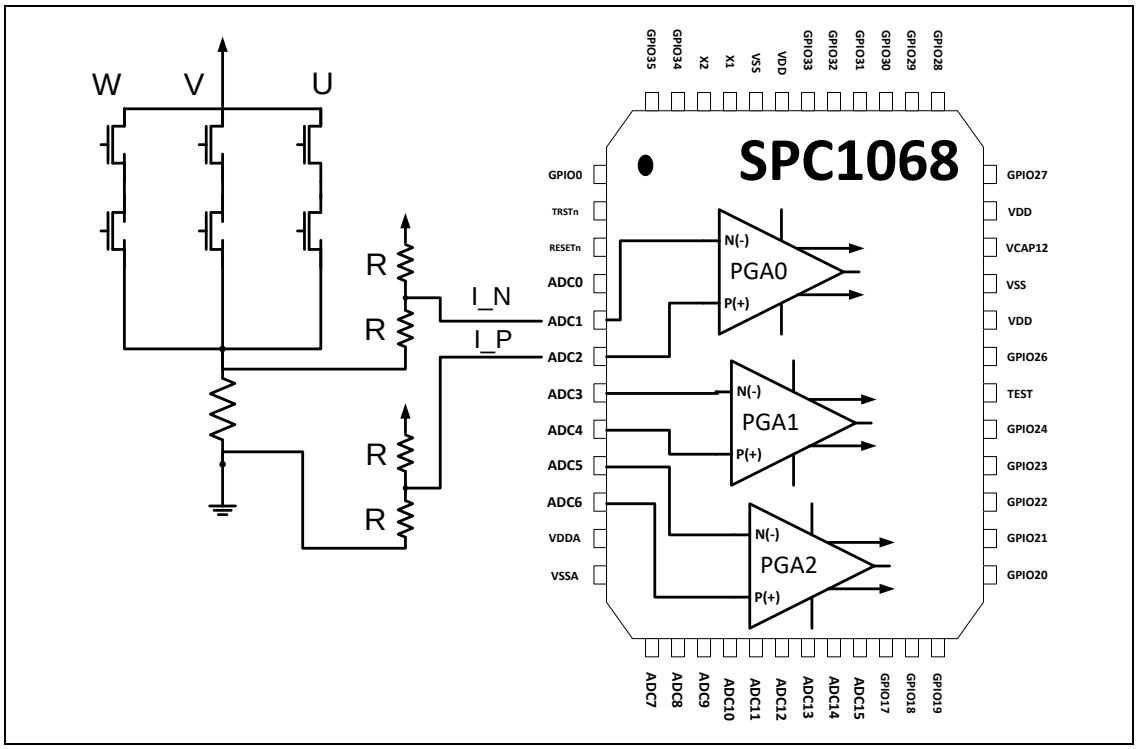
Example Code

```
void PGA_InitExample2_4_3(void)
{
  /* Select PGA input as analog pin */
  GPIO_SetPinAsAnalog(GPIO_3); /* ADC2 I_V+ */
  GPIO_SetPinAsAnalog(GPIO_5); /* ADC4 I_V+ */
  GPIO_SetPinAsAnalog(GPIO_7); /* ADC6 I_W+ */
  /* Init PGA */
  PGA_DiffInit( PGA0,
          PGA0_CH_P_ADC2    /* Positive CH   */,
          PGA0_CH_N_RESDIV0 /* Negative CH   */,
          PGA_SCALE_8X      /* PGA Diff Gain */);
  PGA_ResistorDividerConifg(PGA0,16 /* 1.65V --> 5 bit : 0 is 0V and 31
is 3.3V*/ );
  PGA_DiffInit( PGA1,
          PGA1_CH_P_ADC4    /* Positive CH   */,
          PGA1_CH_N_RESDIV1 /* Negative CH   */,
```

```
            PGA_SCALE_8X      /* PGA Diff Gain */);
  PGA_ResistorDividerConifg(PGA1,16 /* 1.65V --> 5 bit : 0 is 0V and 31
is 3.3V*/ );
  PGA_DiffInit( PGA2,
            PGA2_CH_P_ADC6    /* Positive CH   */,
            PGA2_CH_N_RESDIV2 /* Negative CH   */,
            PGA_SCALE_8X      /* PGA Diff Gain */);
  PGA_ResistorDividerConifg(PGA2,16 /* 1.65V --> 5 bit : 0 is 0V and 31
is 3.3V*/ );
  /* Init ADC in 2 channel mode */
  ADC_EasyInit2(ADC_SOC_0,ADCx_PGA0P,ADCx_PGA0N,ADCTRIG_Software);
  ADC_EasyInit2(ADC_SOC_1,ADCx_PGA1P,ADCx_PGA1N,ADCTRIG_Software);
  ADC_EasyInit2(ADC_SOC_2,ADCx_PGA2P,ADCx_PGA2N,ADCTRIG_Software);
}
```

2.5 单电阻采样配置

图 2-7. 单电阻采样电路图



上图对应的范例代码如下：

Example Code

```
void PGA_InitExample2_4_4(void)
{
  /* Select PGA input as analog pin */
  GPIO_SetPinAsAnalog(GPIO_1); /* ADC1 I- */
  GPIO_SetPinAsAnalog(GPIO_2); /* ADC2 I+ */

  /* Init PGA */
  PGA_DiffInit( PGA0,
          PGA0_CH_P_ADC2    /* Positive CH   */,
          PGA0_CH_N_ADC1    /* Negative CH   */,
          PGA_SCALE_8X      /* PGA Diff Gain */);

  /* Init ADC in 2 channel mode */
  ADC_EasyInit2(ADC_SOC_0,ADCx_PGA0P,ADCx_PGA0N,ADCTRIG_Software);
}
```

3 Comparator 使用教学

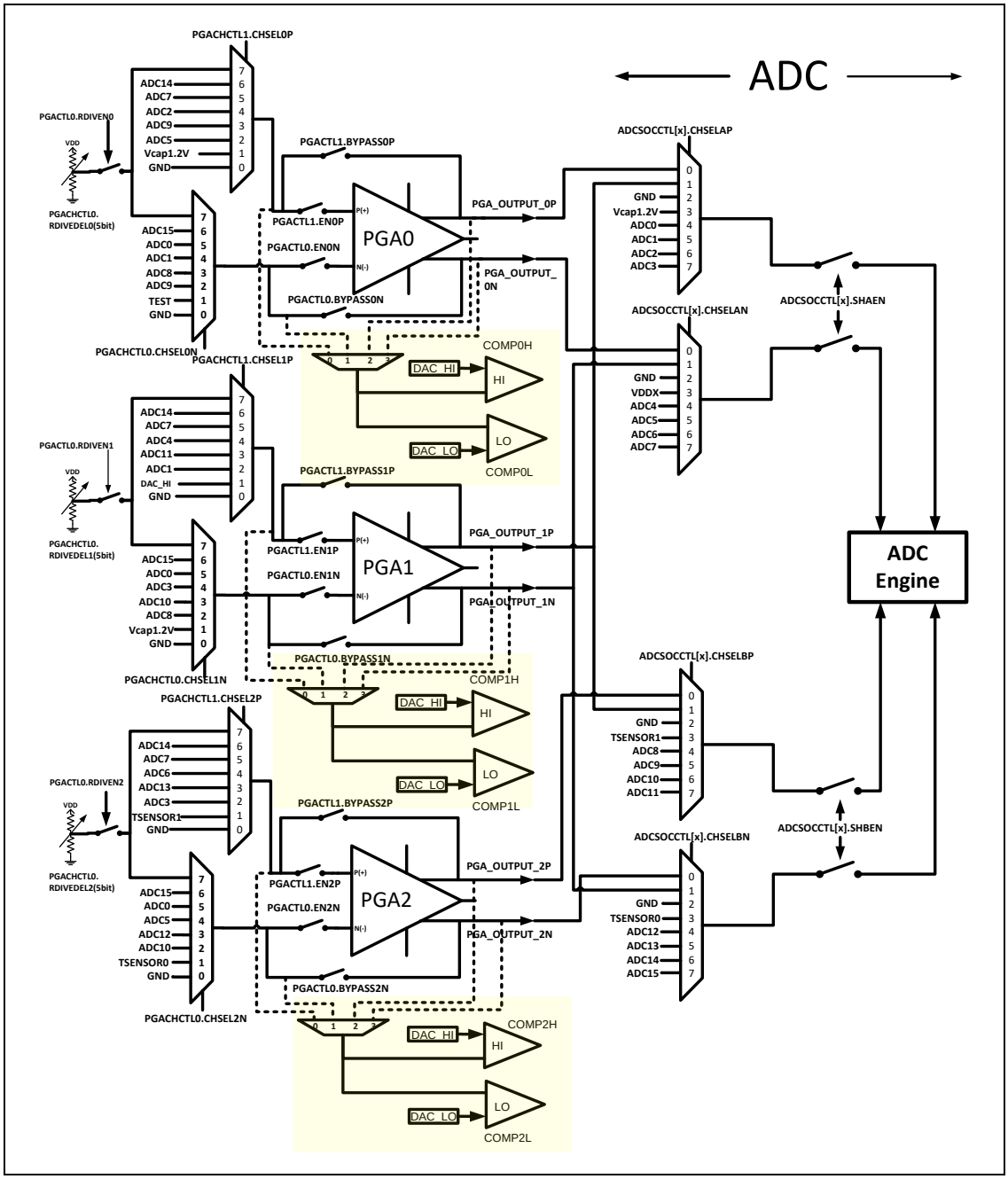
3.1 Comparator 单元概述

SPC1068 共有六个模拟比较器（COMP），特点如下：

- 与 PGA 绑定，每个 PGA 搭配两个 COMP
  一组 COMP 中，一个作为信号 Too High 检测，一个作为 Too Low 检测
- 可利用内部 DAC 调整 COMP 比较阀值
- COMP 输出具数字滤波功能（Deglitch or debounce）可避免 PWM 开关之噪声造成误动作。请注意最长滤波时间限制。
- 具磁滞功能，输出可反转
- 任意一个 COMP 输出可同步关断所有 PWM 输出信号

3.2 COMP 模拟架构

图 3-1. COMP 模拟架构



3.3 COMP 应用于过电流防护实例

1. 范例码展示

COMP 可选择 PGA 输出作为电流防护使用，以下为一个使用实例，以 PGA0 搭配 COMP0 为例，并且使 PWM 停止波形输出。

Example Code

```
void COMPandPGA_InitExample3_3(void)
{
  /* Select PGA input as analog pin */
  GPIO_SetPinAsAnalog(GPIO_2); /* ADC1 */
  GPIO_SetPinAsAnalog(GPIO_3); /* ADC2 */
  /* Init PGA0 */
  PGA_DiffInit(PGA0, PGA0_CH_P_ADC2,PGA0_CH_N_ADC1,PGA_SCALE_8X);
  /* Init ADC in 2 channel mode */
  ADC_EasyInit2(ADC_SOC_0,ADCx_PGA0P,ADCx_PGA0N,ADCTRIG_Software);
  /* Init COMP0 High and Low */
  COMP_Init(COMP_0_HI,
            FROM_PGAxP_OUT,  /* From PGA0P input */
            2450             /* DAC HI compare viltage mV */,
            300              /* Output deglitch time(ns)  */);
  COMP_Init(COMP_0_LO,
            FROM_PGAxN_OUT,  /* From PGA0N output */
            850              /* DAC LO compare viltage mV */,
            300              /* Output deglitch time(ns)  */);
/* When COMP0H or COMP0L equals 1, it can turn off 3 PWM pairs(6 PWM CH)
immediately */
  PWM_EnableOneShotTripFromComp(PWM1,COMP_0_HI);
  PWM_EnableOneShotTripFromComp(PWM1,COMP_0_LO);

  PWM_EnableOneShotTripFromComp(PWM2,COMP_0_HI);
  PWM_EnableOneShotTripFromComp(PWM2,COMP_0_LO);

  PWM_EnableOneShotTripFromComp(PWM3,COMP_0_HI);
  PWM_EnableOneShotTripFromComp(PWM3,COMP_0_LO);
}
```

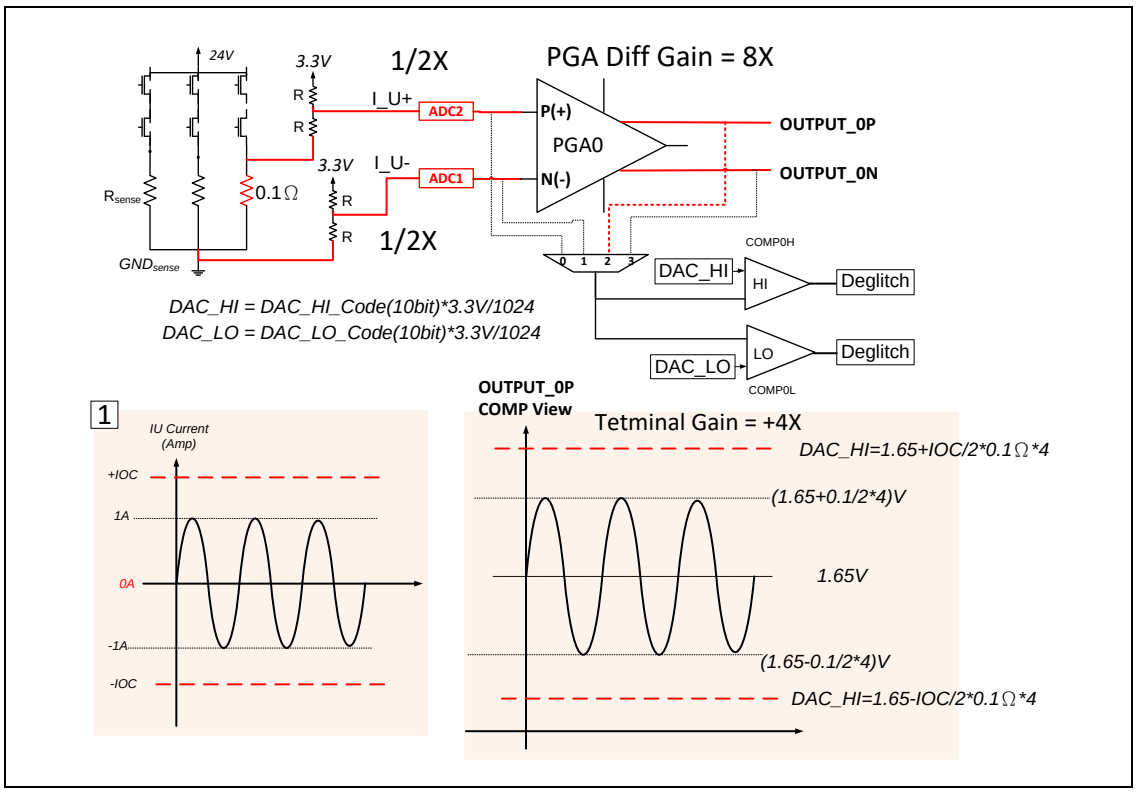
SPC1068 的 COMP 特点在于任何一个 COMP 讯号，均可触发任一 PWM 关断功能，上例为当 COMP0H 或是 COMP0L 产生讯号后，可立即马上关断三相六臂 PWM 讯号。

以上代码同时设定了 COMP 与 PGA，ADC，COMP 的 DAC_HI 阀值为 2.45V，DAC_LOW 为 0.850V，设计原因请参考以下讲解。

2. 图形化说明

请注意 COMP 只能挑选一个 PGA 通道进行比较，而 PGA 若是选择差动增益，实际上单端增益只有差动增益的一半，因此设定 COMP 的比较阀值（DAC_HI 与 DAC_LO）时必须特别小心。如下图所示，PGA 设定差动增益为 8X，但是单端对地的增益只有 4X。

图 3-2. COMP 范例（IOC=2A）



上图的解释如下：

（1） 假若输入是 1A 摆幅，采样电阻 0.1 欧姆，想设定的过电流防护 IOC 为 2A。

（2） COMP0 选择 PGA0 的正端输出作为比较器的输入。

注意三组 COMP，均共享相同的 DAC_HI 与 DAC_LO。

（3） 假若 PGA 差动增益选择为 8X，但这是输出的正端对负端之增益，事实上单端对地的增益只有 4X，而 COMP 看到的增益亦只有 4X。

（4） COMP 看到的电压摆幅只有 4 倍，因此必须设定 DAC 的范围如上图所示。

DAC_HI = 1.65+2A*0.1/2*4 = 2.05V

DAC_LO = 1.65-2A*0.1/2*4 = 1.25 V

（5） 根据以下公式，可以决定 DAC_HI 与 DAC_LOW 之设定数值

$$\text{DAC HI} = \frac{\text{DAC\_HI\_CODE(10bit)}}{1024} \times 3.3\text{V}$$

$$\text{DAC LO} = \frac{\text{DAC\_LO\_CODE(10bit)}}{1024} \times 3.3\text{V}$$

根据（4）的结果，可以计算出：

DAC_HI 必须在寄存器设定为 636，DAC_LO 必须在寄存器设定为 387。

为了使用方便，Spintrol 提供之 COMP_Init（）函数可直接输入 mV 值进行设定。

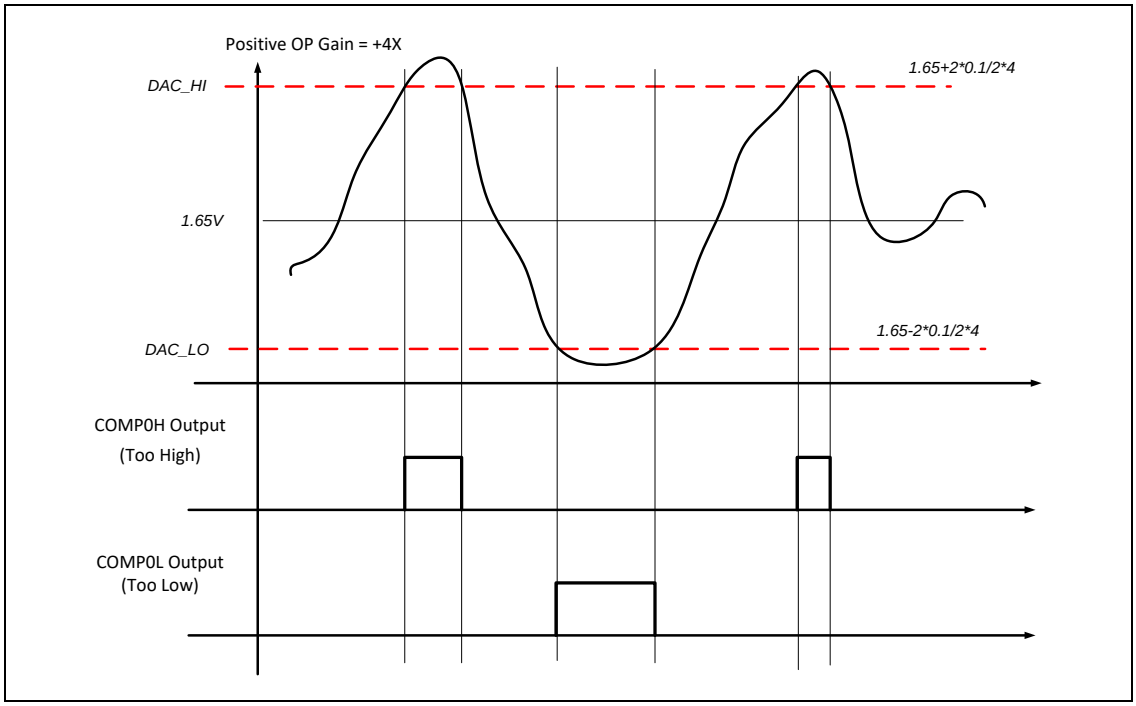
（6） COMP 的输出需要通过数字滤波器，可避免 PWM 开关之噪声误触 COMP。为了使用方便，Spintrol 提供之 COMP_Init（）函数可直接输入此滤波器之宽度，单位为 ns，此例

设定的滤波时间长度为 300ns，可滤除 300ns 长度以下的噪声。详细配置请见下一章节之 API 介绍。

3. COMP 输出演示

根据上一小节的设计，当输入电流过大或过小时，都会触发 COMP 输出。

图 3-3. COMP 输出演示

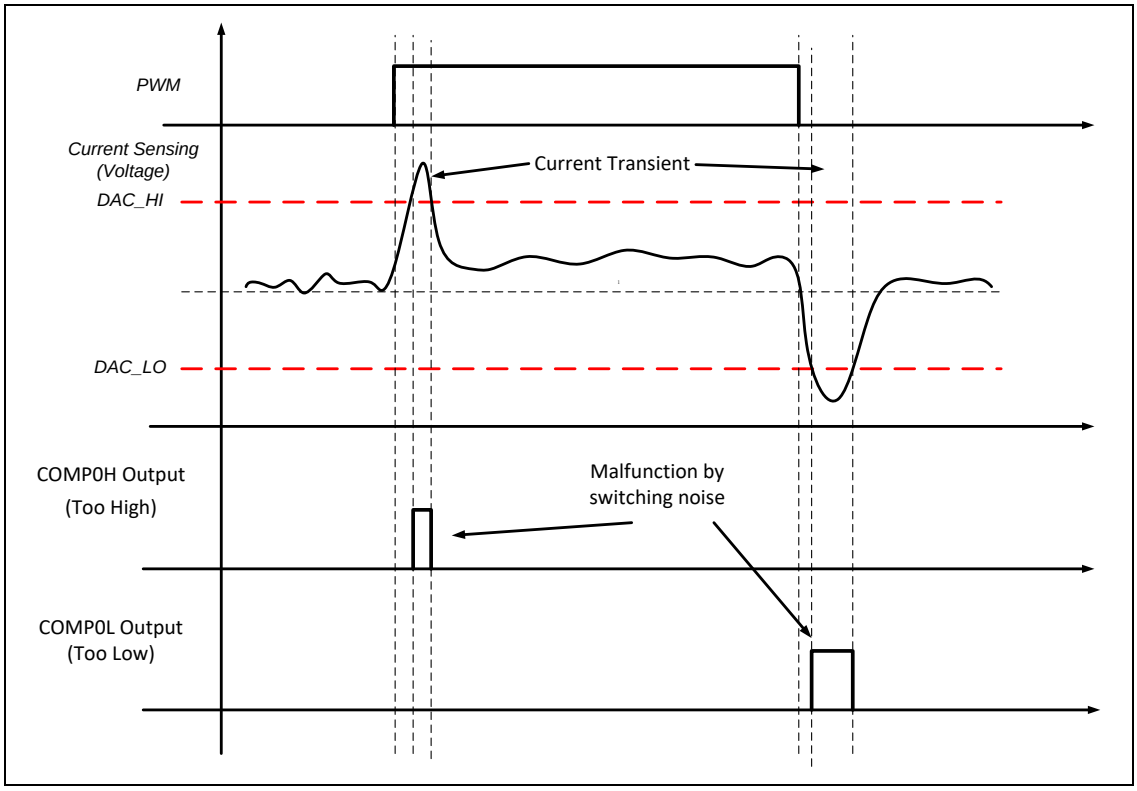


4. COMP 输出 Deglitch（Debounce）功能

PWM 控制 H 桥电路，在电机控制系统或是电源控制系统是非常常见的技术，但是在 PWM 开关的瞬间，往往会对电流回授电路有较大的干扰，或是开关的瞬间常有较大的电流瞬时（Current Transient），此干扰与瞬时是正常反应，如图 3-4 所示。

但是假若 COMP 的输出没有数字滤波电路（Deglitch）功能，此瞬时响应会误触发 COMP 的输出，并且将 PWM 关断，造成不必要的停机，因此在设计上，提供了输出数字滤波器，此滤波器的 clock 来源为 CLK0，长度为 7bit，常见的电机系统数字滤波的长度为 100~300ns。

图 3-4. PWM 切换噪声



需要注意的是滤波时间的限制：此滤波寄存器为 7bit，clock 来源为 CLK0，SPC1068 默认的 CLOCK 配置方式为利用 PLL 将频率提高至 CLK0（PLL 只保证 100M~200M 之频率精准），再利用除频器将需要的 CPU 频率降下，最长滤波时间如下表所示。实际应用上不建议设置过长滤波时间，否则系统过电流时，可能导致驱动电路毁损。

表 3-1. COMP 输出滤波时间

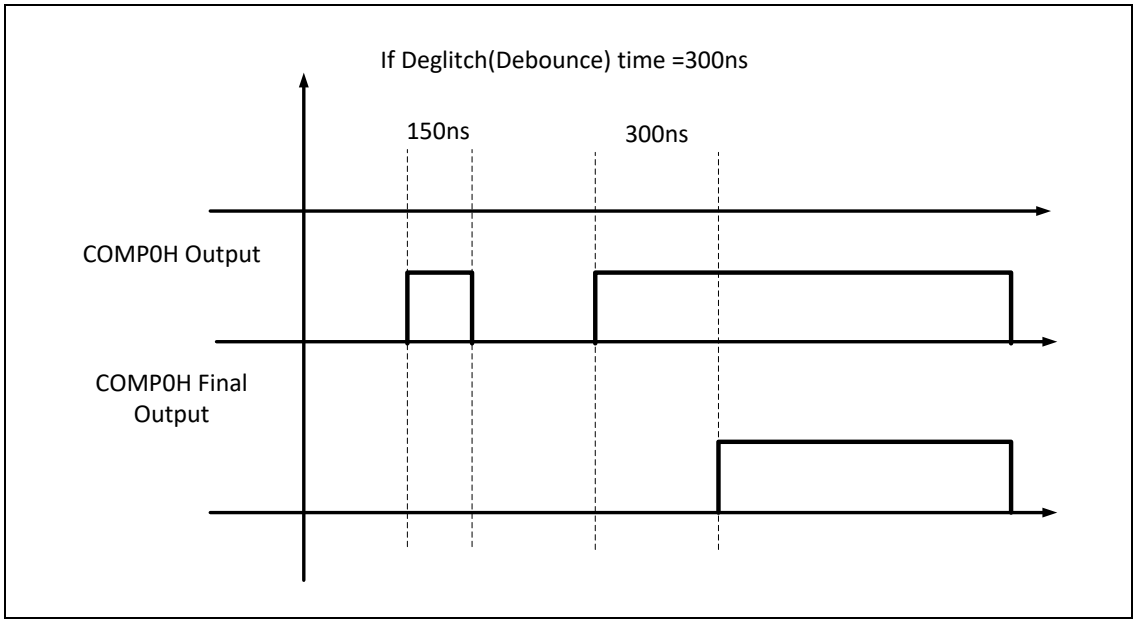
CLK0	HCLK（CPU）	Deglitch Time（7 bit）
120M	24M	1058ns
120M	40M	1058ns
150M	75M	846ns
100M	100M	1270ns
120M	120M	1058ns
150M	150M	846ns

重要 Note：

请执行 COMP 初始化前，务必执行 SPC1068 SDK 中的系统初始化与 Clock 的初始化函数，此两个初始化函数会更新 Clock 信息，如此才能作正确的配置。

加入数字滤波器之后，COMP 输出如下所示。

图 3-5. 带数字滤波器的 COMP 输出



注意 SPC1068 有六个 COMP，而此六个 COMP 的输出滤波电路共享同一个滤波长度，所有的 COMP 滤波长度都一样。

3.4 COMP 初始化 API 使用介绍

COMP 初始化之 API 一共有四个输入参数可以设定，如下面范例所示。

Example Code

```
void COMP_Init(COMP_ComparatorSelEnum eComp, COMP_CHSelEnum eCH,
uint32_t u32DACVoltageMV, uint32_t u32DeglicthTimeNs);

COMP_Init(COMP_0_HI,
          FROM_PGAxP_OUT,  /* From PGA0P input              */
          2450             /* DAC HI/LO compare viltage mV */,
          300              /* Output deglitch time(ns)     */);
```

COMP_Init（）函数的参数介绍如下表所示：

表 3-2. COMP_Init（）函数介绍

COMP_Init	
Parameter	**Description**
eCOMP	It can be COMP_0_LO COMP_1_LO COMP_2_LO COMP_0_HI COMP_1_HI COMP_2_HI 可选择配置以上六个 COMP
eCH	It can be FROM_PGAxN_OUT FROM_PGAxP_OUT FROM_PGAxN_IN FROM_PGAxP_IN 选择该 COMP 之输入通道
u32DACVoltageMV	0～3300（mV）
u32DeglicthTimeNs	输出数字滤波器之时间长度（7 bit），此滤波器来源为 CLK0 if CLK0=100MHZ Its range from 0~127*（1/100MHz） = 1270 ns = 1.27 us if CLK0=150MHZ Its range from 0~127*（1/150MHz） = 846 ns = 0.846 us

请注意以下两点：

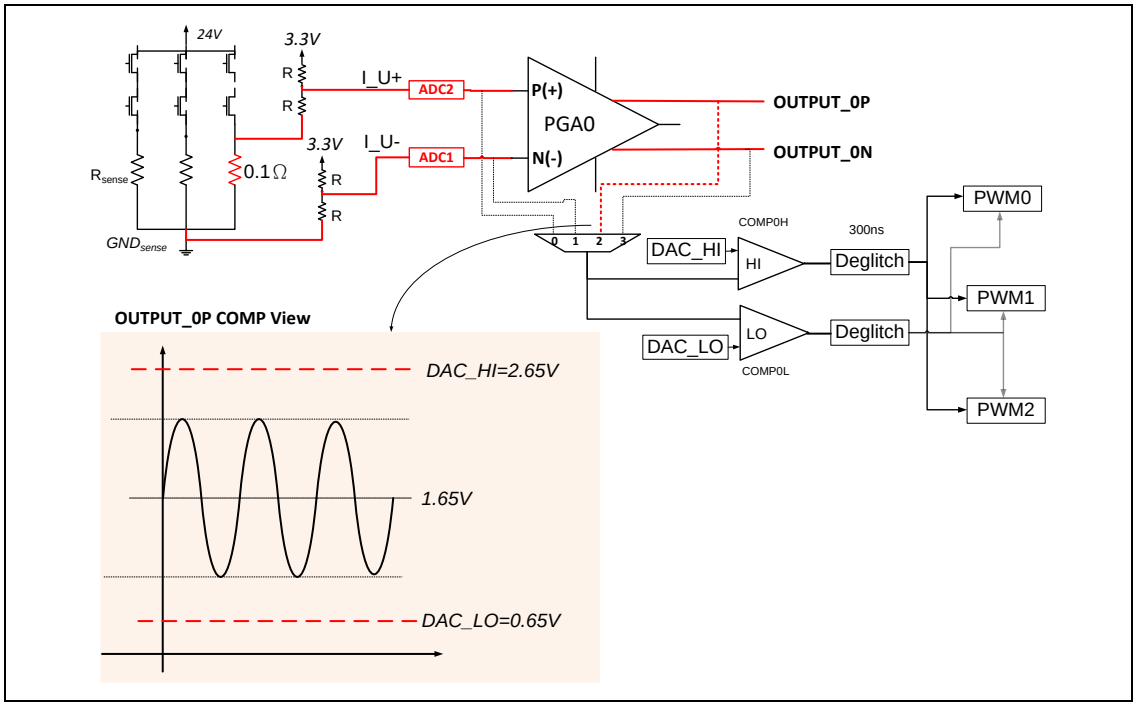
- 所有 Too High comparator 仅共享一组 DAC，并非每一个都有独立的 DAC。同理 Too Low comparator 亦同，因此用 API 设定时，最后设定值将覆盖之前设定的 DAC 数值；
- 所有的 COMP 输出数字滤波时间宽度均一样，因此用 API 设定时，最后设定值将覆盖之前设定的滤波时间数值，建议都设定为一样即可。

3.5 COMP 停止（Trip）PWM 波形功能介绍

SPC1068 的 COMP 专为相电流防护做特殊设计。当 PGA 作为三相电流采集的时候，SPC1068 的 COMP 提供了每一相独立的电流防护，较一般的总电流防护更加安全，以下代码以 U 相电流为例，当 PGA0 输出超过 3V 时，COMP0H 触动 PWM 输出停止，当 PGA0 的电流小于 1V，由 COMPL 触发 PWM 停止。Spintrol 的 COMP 触发 PWM 停止可达成任一个 COMP 同时停止所有 PWM 输出波形功能，详见 PWM trip zone 章节。

举例来说，COMP0H 与 COMP0L，只要任一种状况发生，均可以同时将三相 PWM 停止，同理其他四个 COMP 亦具有同样功能，三相电流一共有六个 COMP，只要其中一个发生触发，均可同时将 PWM 停止，对于相电流防护来说是很重要的一个功能。

图 3-5. COMP 停止 PWM 波形原理图



相比于大部分 IC 需要复杂的设定，Spintrol 亦提供了简单的 API 设定 COMP 触发 PWM trip 功能。当 eCOMP 代表的比较器发生输出由 L 转 H 的瞬间，关闭 PWMx 的 CHA 与 CHB 输出。可重复呼叫设定多种组合，请见范例码。

表 3-3. PWM_EnableOneShotTripFromComp（）函数介绍

PWM_EnableOneShotTripFromComp(PWMx,eCOMP)	
Parameter	**Description**
PWMx	It can be PWM0~PWM6
eCOMP	It can be COMP_0_LO COMP_1_LO COMP_2_LO

	COMP_0_HI COMP_1_HI COMP_2_HI 可选择配置以上六个 COMP

以下为范例码：

Example Code

```
void MotorPWM_InitTripZoneAction(void)
{
  /* Note: Please initial PGA pin before comparator initial */
  COMP_Init(COMP_0_HI,
            FROM_PGAxP_IN,      /* From PGA0P output        */
                 2650           /* DAC HI compare viltage mV */,
                 300            /* Output deglitch time(ns)  */);
  COMP_Init(COMP_0_LO,
            FROM_PGAxP_IN,      /* From PGA0P output        */
                 650            /* DAC LO compare viltage mV */,
                 300            /* Output deglitch time(ns)  */);
/* Step 3: PWM U,V,W will be triped when comparator0 output high.
    This code is prepared for total current protection*/
  PWM_EnableOneShotTripFromComp(PWM_U,COMP_0_HI);
  PWM_EnableOneShotTripFromComp(PWM_U,COMP_0_LO);

  PWM_EnableOneShotTripFromComp(PWM_V,COMP_0_HI);
  PWM_EnableOneShotTripFromComp(PWM_V,COMP_0_LO);

  PWM_EnableOneShotTripFromComp(PWM_W,COMP_0_HI);
  PWM_EnableOneShotTripFromComp(PWM_W,COMP_0_LO);
 /* Step 5: Set PWM reaction after one shot or cycle by cycle event */
  PWM_SetCHAOutputWhenTrip(PWM_U,TRIP_AT_LOW);
  PWM_SetCHBOutputWhenTrip(PWM_U,TRIP_AT_LOW);

  PWM_SetCHAOutputWhenTrip(PWM_V,TRIP_AT_LOW);
  PWM_SetCHBOutputWhenTrip(PWM_V,TRIP_AT_LOW);

  PWM_SetCHAOutputWhenTrip(PWM_W,TRIP_AT_LOW);
  PWM_SetCHBOutputWhenTrip(PWM_W,TRIP_AT_LOW);
  /* Step 6: Enable one shot interrupt in this application */
  PWM_U->TZIE.bit.OST=TZIE_BIT_OST_ENABLE;
  PWM_V->TZIE.bit.OST=TZIE_BIT_OST_ENABLE;
  PWM_W->TZIE.bit.OST=TZIE_BIT_OST_ENABLE;
}
```

4 修订记录

表 4-1. 文档修订记录

日期	版本	修改内容
2017-09-14	1	初始版本