

概述

本手册适用范围：

适用范围	
SPC1125 系列	SPC1125, SPC1128
SPC1168 系列	SPC1155, SPC1156, SPC1158, SPC1168, SPD1148, SPD1178, SPD1188, SPD1163, SPM1173
SPC2168 系列	SPC2168, SPC2165, SPC2166, SPC1198
SPC1169 系列	SPC1169, SPD1179, SPD1176
SPC2188 系列	SPC1185, SPC2188

目录

1	概述	6
2	功能描述	6
3	功能实例	11
3.1	COMP 结果输出到 IO	11
3.1.1	功能需求	11
3.1.2	功能实现	11

SPIN TROL

图片列表

图 2-1: COMPO 示意图	6
图 2-2: COMPO 示意图	7
图 2-3: COMPO 示意图	8
图 2-4: COMPO 示意图	8
图 2-5: COMPO 示意图	9
图 2-6: 数字滤波器框图	9
图 2-7: 数字滤波器模拟框图	10

SPIN
TROL

版本历史

版本	日期	作者	状态	变更
C/0	2024-08-29	苏杭	Released	1. 首次发布。

SPIN TROL

术语或缩写

术语或缩写	描述
MCU	Microcontroller Unit, 微控制器单元

SPIN
TROL

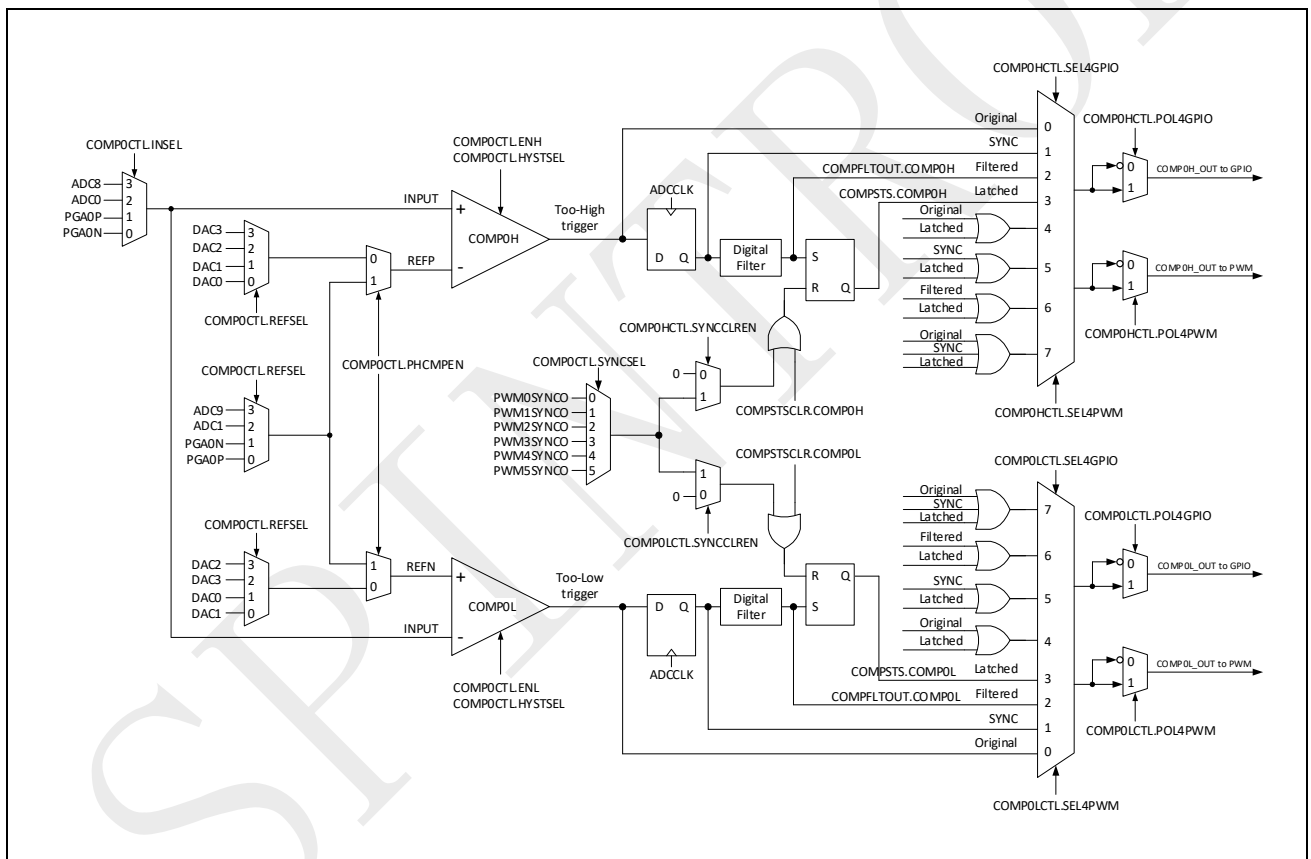
1 概述

COMP 最常规的用法，是将输入电压与 DAC 比较，从而产生 COMPH 或 COMPL 事件。

2 功能描述

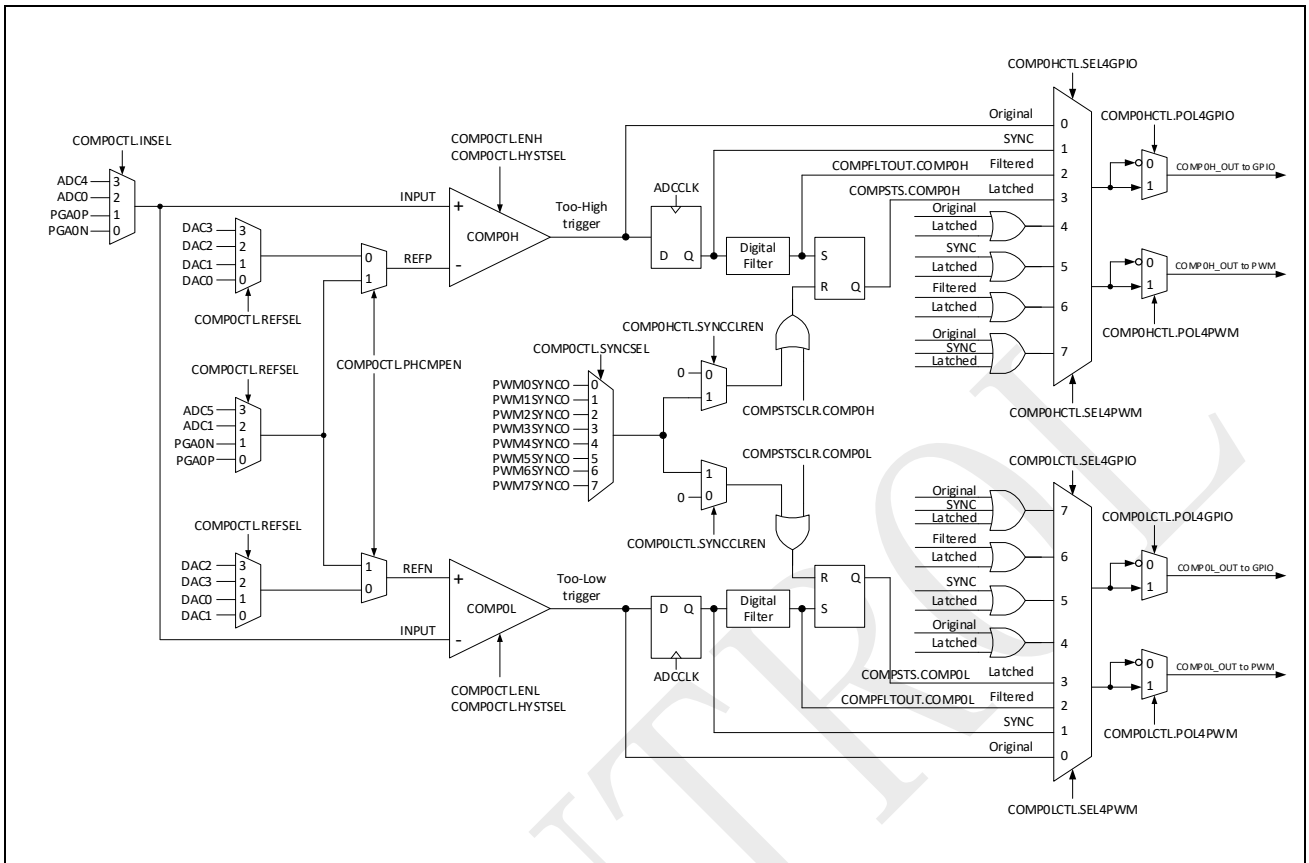
SPC1168/SPC2168/SPC1125/SPC1169/SPC2188 系列 COMP0 的模拟架构如图 2-1 到图 2-5 所示，其余 COMPx 输入端选择见《技术参考手册》相关章节描述。其输出事件作用于 PWM 模块（产生对应的同步、封锁、TZ 中断、ADC 采样、PWM 中断等事件）。

图 2-1: COMP0 示意图



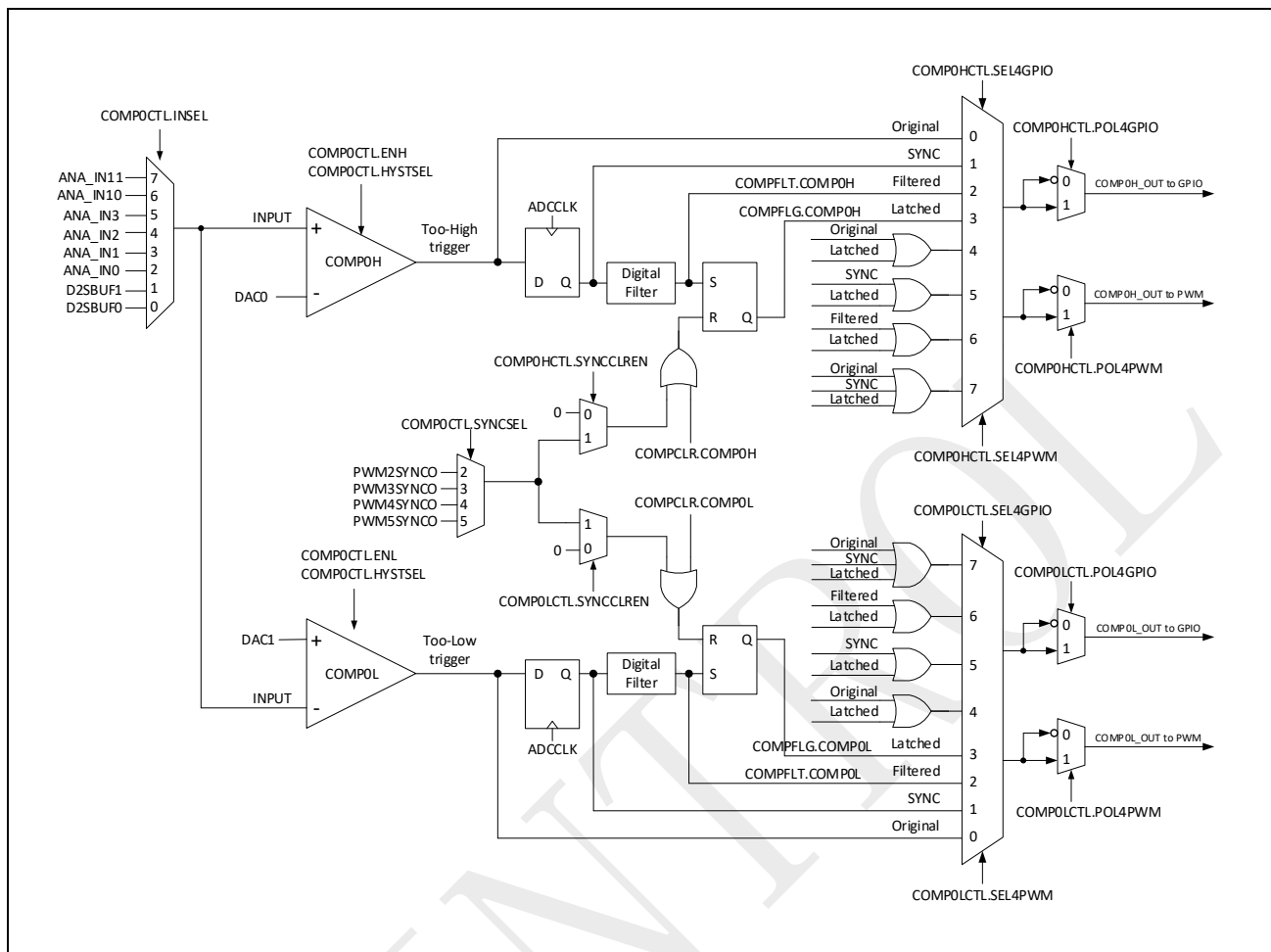
[1] SPC1168 COMP0 结构图。

图 2-2: COMPO 示意图



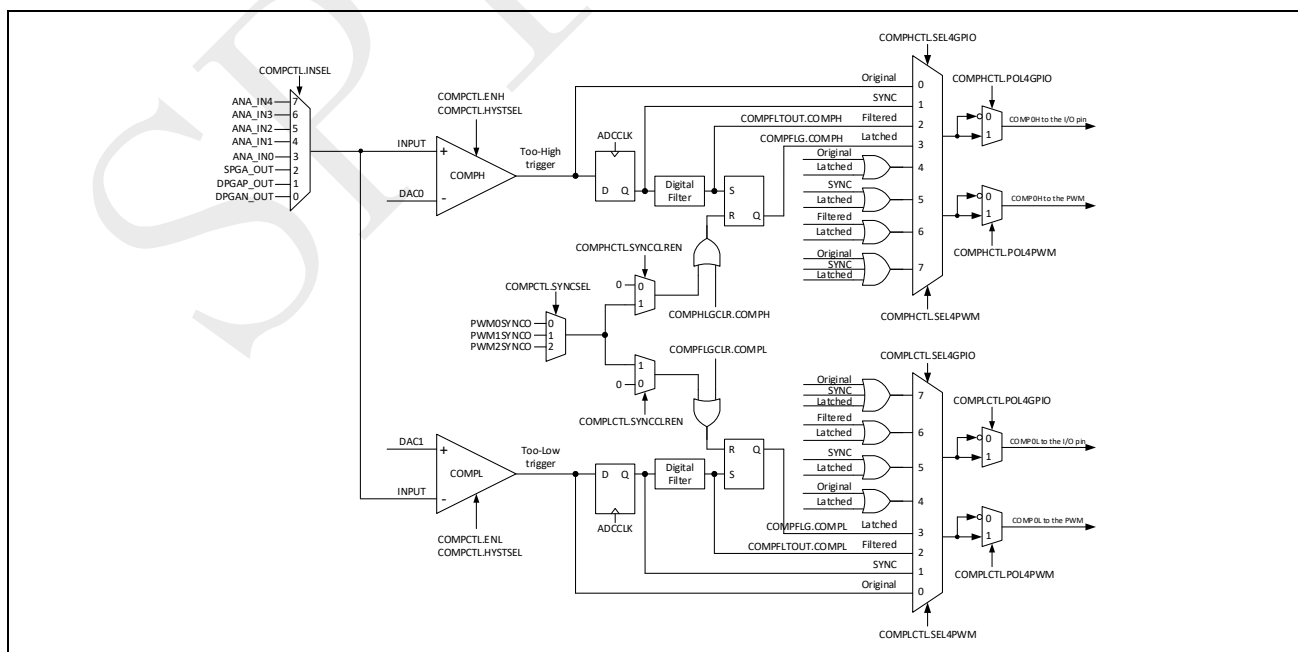
[1] SPC2168 COMPO 结构图。

图 2-3: COMPO 示意图



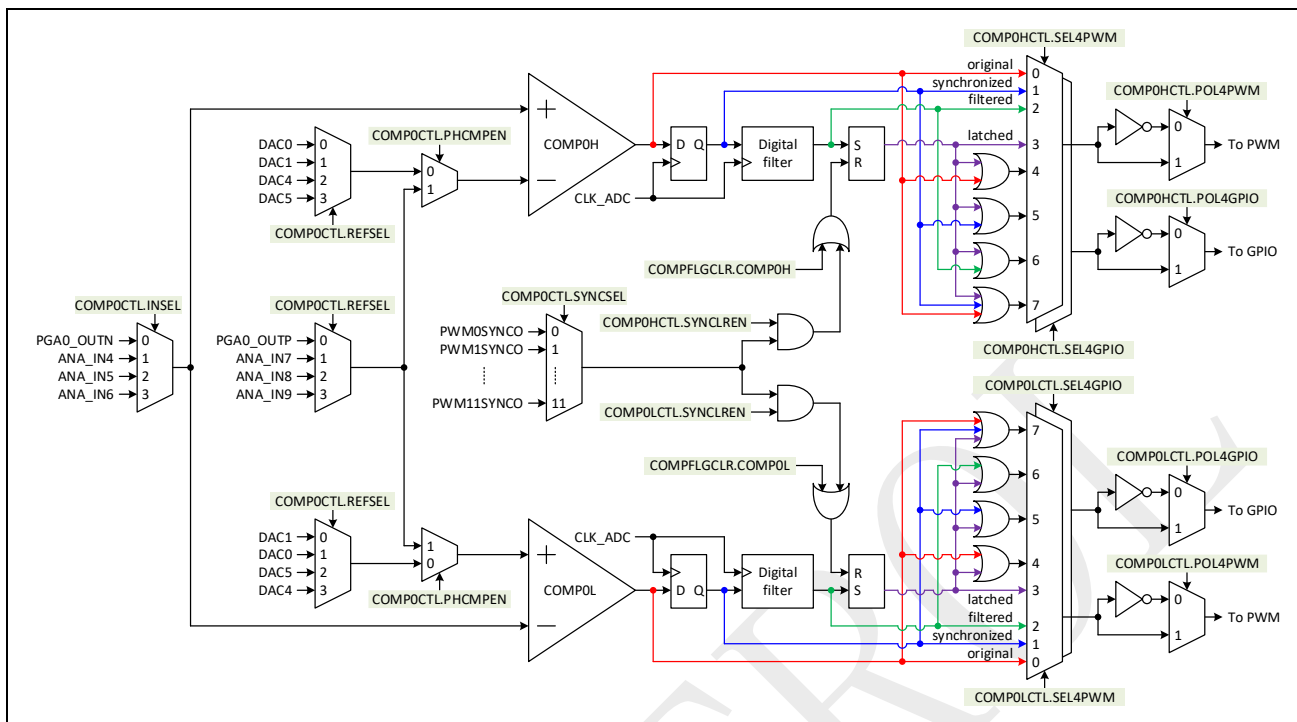
[1] SPC1128 COMPO 结构图。

图 2-4: COMPO 示意图



[1] SPC1169 COMPO 结构图。

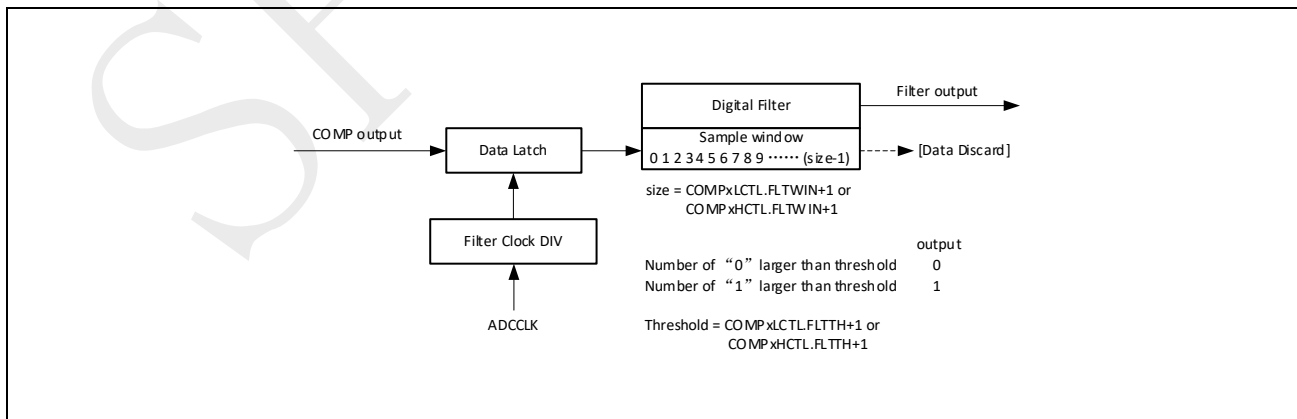
图 2-5: COMPO 示意图



[1] SPC2188 COMPO 结构图。

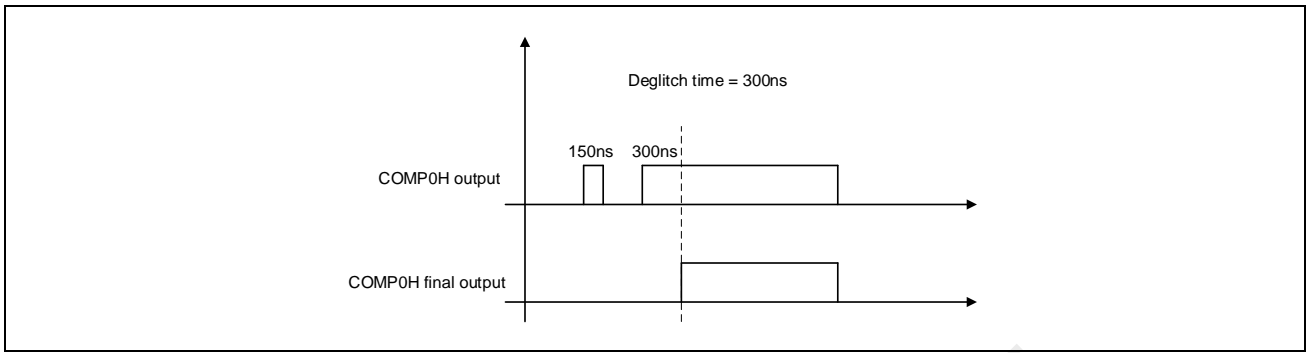
为避免瞬时响应误触发 COMP 对应 COMPH 或 COMPL 事件，设计上提供了输出数字滤波器。其基本原理如下图 2-6 所示，COMP 的输出首先送到锁存器中，时钟来源于 ADCCLK，并且最大可以分频到 1024 倍。然后设置采样窗口的大小和阈值的大小，范围可以从 0~31 中选择，并且要保证阈值要大于采样窗口的一半，如果采样窗口内采到的 COMP 输出 0 的个数大于阈值，则数字滤波器输出为 0，如果采样窗口内的 COMP 输出 1 的个数大于阈值，则数字滤波器输出为 1，如果采用窗口内的 COMP 输出 0 和 1 个数都没有超过阈值，则数字滤波器输出不变。

图 2-6: 数字滤波器框图



请执行 COMP 初始化前，务必执行 Clock 的初始化函数，如此才能作正确的配置相关信息。加入数字滤波器之后，COMP 输出如图 2-7 所示，假设设定阈值是 300ns，不足 300ns 的信号被过滤。

图 2-7: 数字滤波器模拟框图



SPIN TROL

3 功能实例

3.1 COMP 结果输出到 IO

3.1.1 功能需求

COMP 结果输出到 IO。

3.1.2 功能实现

通过 COMP 比较 GPIO8 和 DAC2, DAC3 的电压, 并将 COMP 信号输出到 GPIO0, GPIO1 的操作如下。

Example Code

```
int main()
{
    FLASH_WALLOW();

#ifdef SPC1158
    FLASH_SetTiming(100000000);
    /* Disable flash write access after flash operation had done */
    FLASH_WDIS();

    CLOCK_InitWithRCO(CLOCK_HCLK_100MHZ);
#else
    FLASH_SetTiming(200000000);
    /* Disable flash write access after flash operation had done */
    FLASH_WDIS();

    CLOCK_InitWithRCO(CLOCK_HCLK_200MHZ);
#endif

    Delay_Init();

    /*
     * Init the UART
     *
     * 1.Set the GPIO34/35 as UART FUNC
     *
     * 2.Enable the UART CLK
     *
     * 3.Set the rest para
     */
    GPIO_SetPinChannel(GPIO_34, GPIO34_UART_TXD);
    GPIO_SetPinChannel(GPIO_35, GPIO35_UART_RXD);
    CLOCK_EnableModule(UART_MODULE);
    UART_Init(UART, 38400);

    /*
     * Configure Comparator.
     *
     * 1.Set low detection.
     *
     * 2.Set ADC8 as the input.
     *
     * 3.Set DAC as 1000mV for low voltage detecting reference voltage.
     *
     * 4.Set the Digital Filter as 25ns, in order to filter the noise.
     */
}
```

Example Code

```

COMP_Init(COMP_0_LO, COMP0_FROM_ADC8, 1000, 25);

/* Set high detection for ADC8 and reference value is 2000mV */
COMP_Init(COMP_0_HI, COMP0_FROM_ADC8, 2000, 25);

COMP->COMP0HCTL.bit.SEL4GPIO = COMP0HCTL_BIT_SEL4GPIO_FILTRED;
COMP->COMP0HCTL.bit.POL4GPIO = COMP0HCTL_BIT_POL4GPIO_ACTIVE_HIGH;
GPIO_SetPinChannel(GPIO_0, GPIO0_COMP0H);

COMP->COMP0LCTL.bit.SEL4GPIO = COMP0LCTL_BIT_SEL4GPIO_FILTRED;
COMP->COMP0LCTL.bit.POL4GPIO = COMP0LCTL_BIT_POL4GPIO_ACTIVE_HIGH;
GPIO_SetPinChannel(GPIO_1, GPIO1_COMP0L);
while (1)
{
    /* Get the comparator status */
    u32status = COMP_GetFilterOutputStatus(COMP_0_HI);
    if (u32status & 0x1)
    {
        COMP_ClearAllFilterOutputStatus();
        printf("The target voltage is higher than high-boundary of
comparator\n");
    }

    u32status = 0;
    u32status = COMP_GetFilterOutputStatus(COMP_0_LO);
    if (u32status & 0x1)
    {
        COMP_ClearAllFilterOutputStatus();
        printf("The target voltage is lower than low-boundary of
comparator\n");
    }

    Delay_Ms(500);
}
}

```

[1] 示例代码适用于 SPC1168。

通过 COMP 比较 GPIO0 和 DAC0, DAC1 的电压，并将 COMP 信号输出到 GPIO6, GPIO7 的操作如下。

Example Code

```

int main(void)
{
    CLOCK_InitWithRCO(CLOCK_CPU_100MHZ);

    Delay_Init();

    /*
     * Init the UART
     */
    PIN_SetChannel(PIN_GPIO10, PIN_GPIO10_UART0_TXD);
    PIN_SetChannel(PIN_GPIO11, PIN_GPIO11_UART0_RXD);
    UART_Init(UART0, 38400);
    printf("Enter the test\n");

    PIN_SetChannel(PIN_GPIO0, PIN_GPIO0_ANA_IN0);
}

```

Example Code

```
/* Set ANA_IN0 as input to comparator, set Too High threshold is 2500mV,
filter window is 200ns */
COMP_Init(COMP_H, COMP_FROM_ANA_IN0, 2500, 200);

/* Set ANA_IN0 as input to comparator, set Too Low threshold is 1900mV,
filter window is 200ns */
COMP_Init(COMP_L, COMP_FROM_ANA_IN0, 1900, 200);
PIN_SetChannel(PIN_GPIO6, PIN_GPIO6_COMP_MON2);
PIN_SetComparatorMonitorSource(PIN_COMP_MONITOR_CH2, PIN_MONITOR_COMP_L);
COMP_SetOutputPolarityForGPIO(COMP_L, HIGH);

PIN_SetChannel(PIN_GPIO7, PIN_GPIO7_COMP_MON3);
PIN_SetComparatorMonitorSource(PIN_COMP_MONITOR_CH3, PIN_MONITOR_COMP_H);
COMP_SetOutputPolarityForGPIO(COMP_H, HIGH);

while (1)
{
    /* Get the comparator status */
    if (COMP_GetFilterOutputFlag(COMP_H) != 0)
    {
        printf("ANA_IN0 is higher than 2500mV\n");
    }
    else if (COMP_GetFilterOutputFlag(COMP_L) != 0)
    {
        printf("ANA_IN0 is lower than 1900mV\n");
    }

    /* Clear latched filter status */
    COMP_ClearFilterOutputFlag(COMP_L | COMP_H);

    Delay_Ms(500);
}
}
```

[1] 示例代码适用于 SPC1169，其它系列产品的示例代码会根据实际需求进行补充。