



CMS32F033 应用笔记

内置比较器应用笔记

Rev. 1.0.1

请注意以下有关CMS知识产权政策

* 中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 www.mcu.com.cn

目录

1. 模块概述	3
2. 模块特性	3
3. 模块功能	3
3.1 结构图	3
3.2 寄存器	4
4. 操作模式	5
4.1 模拟比较器调节模式	5
4.2 模拟比较器的迟滞模式	6
4.3 模拟比较器的负端电压选择	6
4.4 模拟比较器的中断/事件模式	6
4.5 模拟比较器的寄存器操作保护	6
5. 参考样例	7
6. 版本修订说明	9

1. 模块概述

芯片内部包含两个模拟比较器。可按照比较器的配置适用于不同的应用场合。当正端电压大于负端电压时，比较器输出逻辑 1，反之输出 0，也可以通过输出极性选择位进行改变。当比较器输出值发生改变时，每路比较器都可通过配置产生中断。

2. 模块特性

- ◆ 模拟输入电压范围：0~(VDD-1.5)
- ◆ 支持迟滞电压选择(10mV/20mV/60mV-典型值)
- ◆ 每个比较器正端可选择 4 路端口输入
- ◆ 每个比较器负端可选择端口输入与内部参考电压
- ◆ 内部参考电压可选择内部 Bandgap(1.2V)与 VDD 的分压输出
- ◆ 内部参考 VDD 分压范围：(2/20)*VDD~(17/20)*VDD 共 16 档位选择
- ◆ 输出可滤波时间可选择：0~512* T_{sys}
- ◆ 比较器事件输出可作为增强型 PWM 的刹车触发信号
- ◆ 输出改变可产生中断

3. 模块功能

3.1 结构图

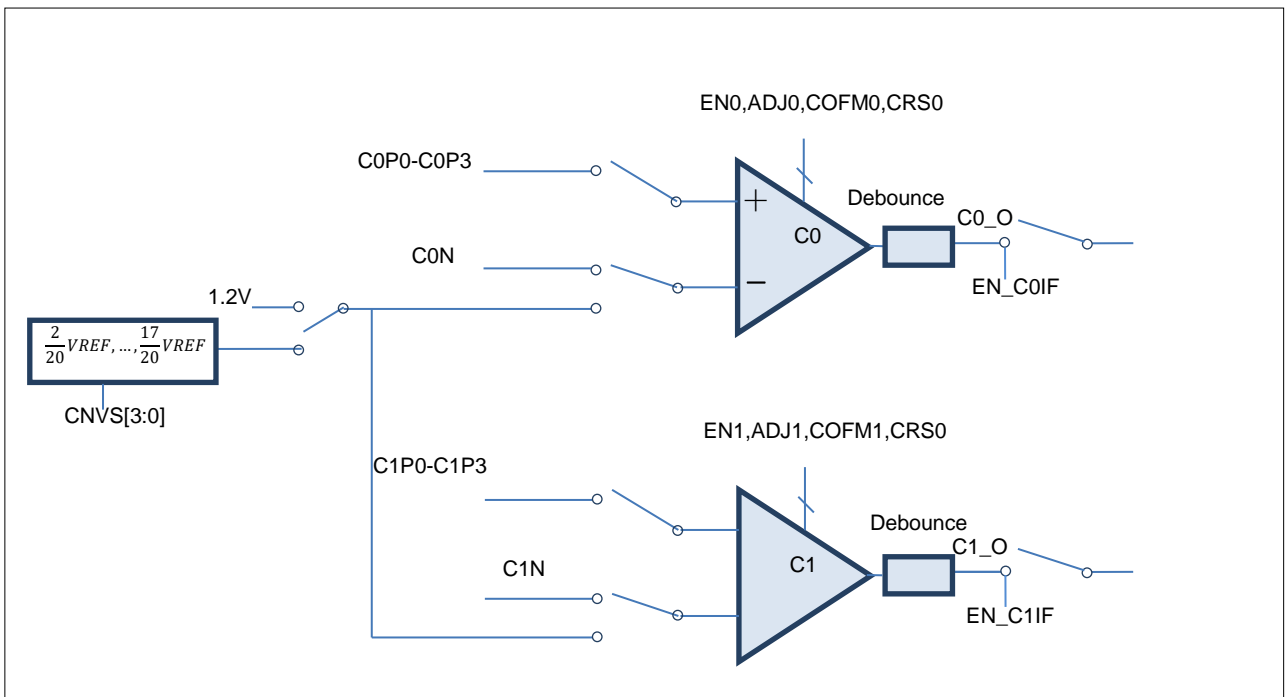


图 3-1：模块功能结构图

3.2 寄存器

参见 CMS32F033 用户手册的“模拟比较器（ACMP0/1）”章节了解关于寄存器设置的更多信息。

4. 操作模式

4.1 模拟比较器调节模式

1. 模拟比较器使用前都需要校调。可通过设置最佳的比较器校调值实现模拟比较器的校调。
2. 设置 CnADJE 的值来选择模拟比较器调节模式的调节值来源。
 - 1) 写入 0xAA, 模拟比较器调节值使用 CnCON1 中的 OPnADJ<4:0>的值。
 - 2) 写入其他值, 模拟比较器调节值使用相关 Config 的值, 此值由厂家校调写入芯片 Config。芯片复位时, CnADJE 值为 0x00, 模拟比较器调节模式默认使用 Config 的值。
3. 使用比较器调节模式获取最佳调节值 (无特殊说明, 以下均以 ACMP0 为例)。
 - 1) 使能 ACMP0 调节模式, C0CON0<C0CFM> = 1。
 - 2) 使能 ACMP0 调节模式负端接地, C0CON0<C0N2GND> = 1。
 - 3) 设置比较器调节模式输入端为正端, C0CON1<C0CRS> = 1。
 - 4) 设置 ACMP0 调节模式使用 C0CON1 中的 OP0ADJ<4:0>的值, C0ADJE = 0xAA。
 - 5) 开启 ACMP0 模块, C0CON0<C0EN> = 1。
 - 6) 获取 ACMP0 的输出值 (C0CON1<C0OUT>), 改变 C0CON1 中的 OP0ADJ<4:0>的值, 若 ACMP0 的输出值 (C0CON1<C0OUT>) 发生跳变, 则此值就是最佳调节值。

部分样例代码:

```
if( ACMP0 == ACMPx )
{
    ACMP_EnableAdjustMode(ACMP0,ACMP_ADJ_SEL_P);
    ACMP->LOCK = ACMP_WRITE_KEY;
    ACMP->C0CON1 &= ~(ACMP_C0CON1_ADJ_Msk);
    ACMP->C0ADJE = ACMP_ADJ_USE_CNADJE;
    ACMP->LOCK = 0x00;
ACMP_Start(ACMP0);
    for(AdjDelay=100; AdjDelay>0; AdjDelay--);

    AdjResltFlag =ACMP_GetResult(ACMP0);
    for(AdjValue =0; AdjValue<32;AdjValue++)
    {
        ACMP->LOCK = ACMP_WRITE_KEY;
        ACMP->C0CON1 |= AdjValue;
        ACMP->LOCK = 0x00;
        for(AdjDelay=100; AdjDelay>0; AdjDelay--);

        if(AdjResltFlag != ACMP_GetResult(ACMP0))
        {
            ACMP_DisableAdjustMode(ACMP0);
            ACMP_Stop(ACMP0);
            return AdjValue;
        }
    }
}
```

4.2 模拟比较器的迟滞模式

模拟比较器的迟滞模式可选择不同的迟滞电压：10mV、20mV、60mV。

模拟比较器可选择在不同的跳变点加设置迟滞电压：

设置跳变点	迟滞功能描述	附加描述
输出 0 -> 1 时迟滞	正端电压+迟滞电压>负端电压时，输出从 0 变化为 1	负端电压 > 正端电压时，输出从 1 变化为 0
输出 1 -> 0 时迟滞	负端电压+迟滞电压>正端端电压时，输出从 1 变化为 0	正端电压 > 正端电压时，输出从 1 变化为 0

注：

- 1) 应该在输出极性选择之前选择迟滞控制方式。
- 2) 输出极性：
 - 正常模式：正端电压大于负端电压，输出 1，反之输出 0。
 - 反相模式：正端电压大于负端电压，输出 0，反之输出 1。

4.3 模拟比较器的负端电压选择

模拟比较器的负端电压可以有以下选择：

- 1) CnN 端，可以通过 IO 外部给入负端电压。
- 2) 内部参考电压 1.2V。（厂家校调）
- 3) VDD 的分压，分压系数 K 可设置。 $K = (2/3/4/5/6\dots/16/17) / 20$ 。
- 4) 内部 1.2V 参考电压的分压，分压系数 K 可设置。 $K = (2/3/4/5/6\dots/16/17) / 20$ 。

4.4 模拟比较器的中断/事件模式

模拟比较器的事件输出模式可用来触发 EPWM 刹车。

- 1) 模拟比较器的中断/事件的触发模式可设置为 3 种模式：（中断、事件的触发共用一种模式）
- 2) 比较器 1 输出从 0->1 的跳变触发中断和产生事件。
- 3) 比较器 1 输出从 1->0 的跳变触发中断和产生事件。
- 4) 比较器 0 输出从 0->1 的跳变或从 1->0 的跳变触发中断和产生事件。

4.5 模拟比较器的寄存器操作保护

为了防止误操作模拟比较器寄存器，设置了对应的保护模式。

当寄存器 LOCK=0x55 时，使能操作 ACMP 相关寄存器（详见寄 ACMP 寄存器映射说明）；

当寄存器 LOCK=其他值时，禁止操作 ACMP 相关寄存器。

5. 参考样例

比较器模块的使用相对比较简单，请按照如下的方法进行配置：（以 ACMP0 为例）

- 1) 开启 ACMP0 的时钟。
- 2) 获取 ACMP0 的调节模式的最佳调节值，获取值后关闭 ACMP0。
- 3) 设置 ACMP0 正端输入。
- 4) 设置 ACMP0 负端输入。
- 5) 设置 ACMP0 调节方式与调节值。
- 6) 设置 ACMP0 的迟滞电压。
- 7) 设置 ACMP0 的输出滤波时间。
- 8) 设置 ACMP0 的输出极性。
- 9) 设置 ACMP0 的中断模式与中断优先级。
- 10) 设置 ACMP0 的 IO 口模拟模式，目的是为了关闭 IO 口的数字功能。
- 11) 开启 ACMP0。

样例代码：

```

/*****
** \brief   ACMP_ACMP0_Config
**          实现 ACMP0 的比较功能
**          (1) 正端电压 (IO) > 负端电压 (内部基准 1.2V) 输出 1 (高电平)
**          (2) 正端电压 (IO) < 负端电压 (内部基准 1.2V) 输出 0 (低电平)
**          (3) 输出从 0 跳变到 1，产生上升沿中断
** \param [in] none
** \return  none
** \note
*****/
void ACMP_ACMP0_Config(void)
{
    uint32_t AdjResult;

    /*
    (1) 开启 ACMP 模块时钟
    */
    SYS_EnablePeripheralClk(SYS_CLK_ACMP_MSK);
    /*
    (2) 获取 ACMP0 校调调节值
    */
    AdjResult = ACMP_EnableAutoAdjust(ACMP0);

    /*
    (3) 设置 ACMP0 正端输入
    */
    ACMP_ConfigPositive(ACMP0, ACMP_POSEL_P0);
    /*
    (4) 设置 ACMP0 负端输入
    */
    ACMP_ConfigNegative(ACMP0, ACMP_NEGSEL_VREF_BG, 0);          /*负端选择内部 1.2V
基准电压*/
    /*
    (5) 设置电压调节
    */
    ACMP_EnableCADJAdjust(ACMP0, AdjResult);

    /*
    (6) 开启滤波输出
    */
    ACMP_EnableFilter(ACMP0, ACMP_NFCLK_3_TSYS);
    
```

```
    /*
    (7) 关闭反相输出
    */
    ACMP_DisableReverseOutput (ACMP0);
    /*
    (8) 关闭事件输出
    */
    ACMP_DisableEventOut (ACMP0);

    /*
    (9) 设置中断
    */
    ACMP_ConfigEventAndIntMode (ACMP0,ACMP_EVENT_INT_RISING); /* 上升沿触发中断与
事件，中断触发方式与事件触发方式共用*/
    ACMP_EnableInt (ACMP0);

    NVIC_EnableIRQ (ACMP_IRQn);
    /*
    (10) 设置中断优先级
    */
    NVIC_SetPriority (ACMP_IRQn,2);

    /*
    (11) 设置迟滞
    */
    ACMP_ConfigHYS (ACMP0,ACMP_HYS_0_TO_1,ACMP_HYS_S_60);

    /*
    (12) 配置 ACMP IO 口
    */
    SYS_SET_IOCFG (IOP13CFG,SYS_IOCFG_P13_C0P0); /*设置 P13 为 ACMP0
正端输入*/
    SYS_SET_IOCFG (IOP26CFG,SYS_IOCFG_P26_C0O); /*设置 P26 为 ACMP0
输出*/
    /*
    (13) 开启 ACMP0
    */
    ACMP_Start (ACMP0);

    /*
    (14) P12 设置为推挽输出模式，指示 ACMP0 的中断
    */
    SYS_SET_IOCFG (IOP12CFG,SYS_IOCFG_P12_GPIO); /*
设置 P12 为 GPIO 模式*/
    GPIO_CONFIG_IO_MODE (GPIO1,GPIO_PIN_2,GPIO_MODE_OUTPUT); /* 设置 P12
为推挽输出模式*/
    GPIO1->DO_f.P2 = 0;
    /*P12 输出低电平*/
}
}
```


6. 版本修订说明

版本号	时间	修改内容
V1.0.0	2019 年 11 月	初始版本
V1.0.1	2020 年 12 月	更改为新格式