

OB6619 配置文档

使用说明

编写:

日期: **20211012**

Revision Record

Date	Version	Revision History	Author
20211012	V1.0.0	First Issue	JianWei Zhang

一、功能框图

OB6619 是一款高集成、高性能、低成本的直流无刷电机 (BLDC)、永磁同步电机 (PMSM) 控制芯片，只需要少量的外围器件即可搭建完整的 BLDC/PMSM 控制电路。功能方框图如图 1-1 所示。

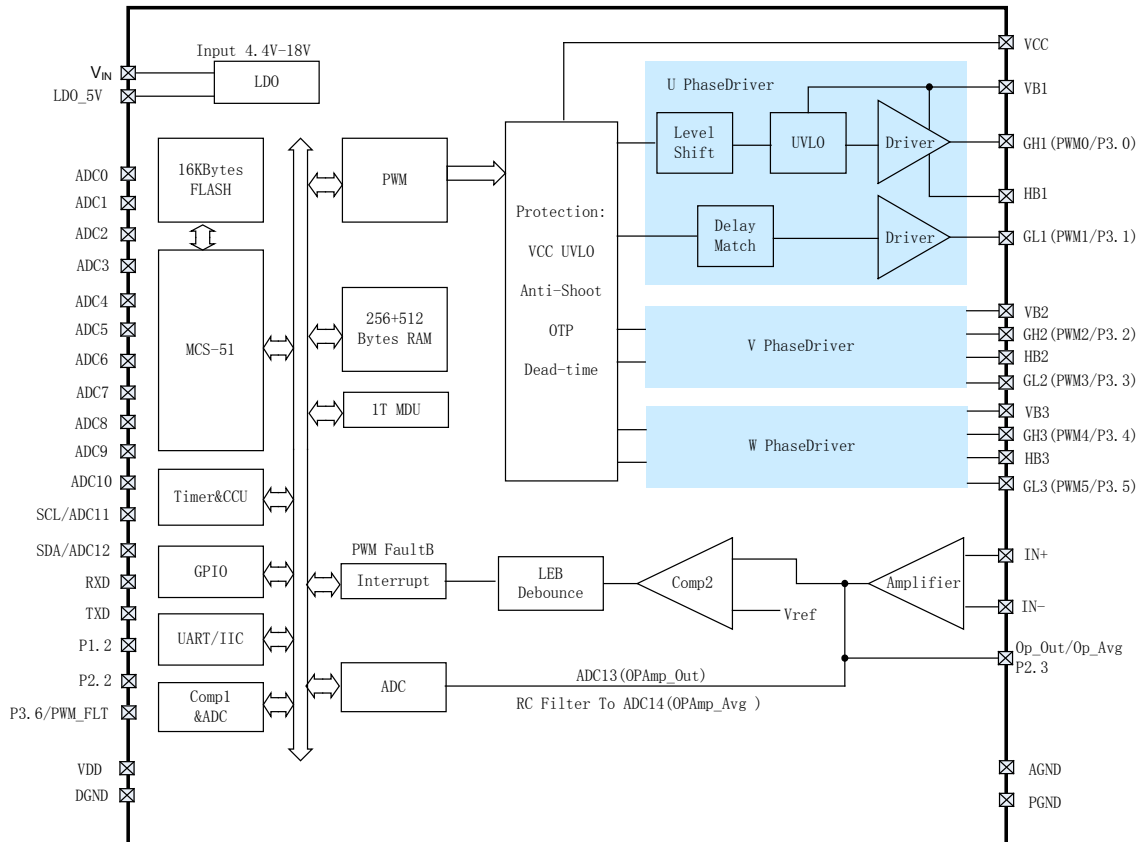


图 1-1 功能方框图

二、常用寄存器配置说明

1. 存储器结构

- a. OB6619 有 16KB 的嵌入式闪存，如以下图 2-1，可做为通用的程序存储或 EEPROM，其中有 4K 地址可选作为 ISP 程序服务空间；
- b. OB6619 具有 512 字节+ 256 字节片上 SRAM，如以下图 2-2，其中 256 字节和通用的 8052 内部存储器结构一样，同时扩大 512 字节片上 SRAM 可以访问外部内存寻址（由指令 MOVX）。

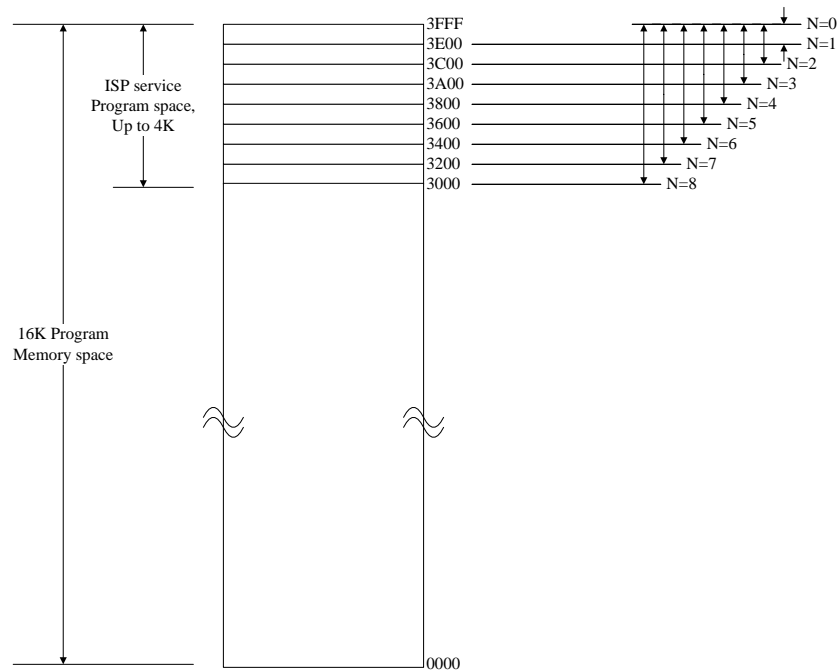


图 2-1: OB6619 programmable Flash

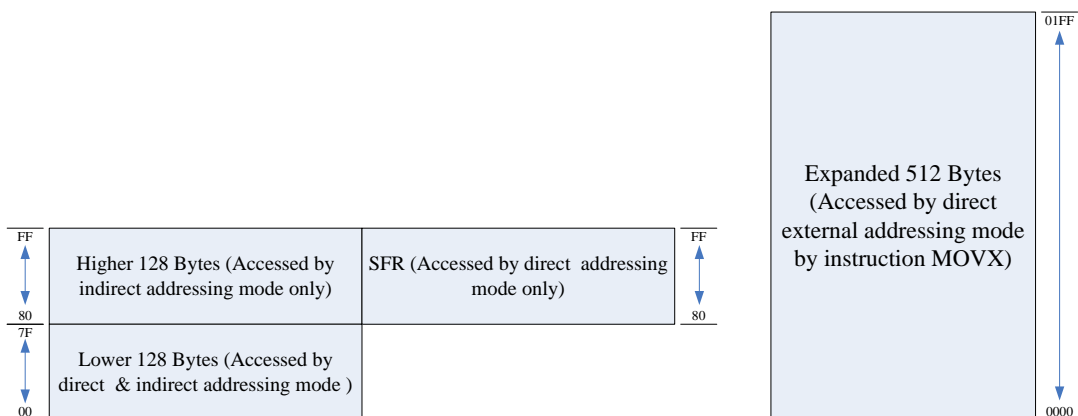


图 2-2: RAM architecture

2. 看门狗初始配置和复位

看门狗定时器是一个 8 位的计数器并会在计数器溢出时产生复位讯号。时钟由的片上 RC 振荡器 (约 20 KHz) 提供。WDT 有可选择的分频。分频设置由看门狗控制寄存器 (WDTC) 的位 3~位 0 (WDTM [3:0]) 配置。

$$WDTCLK = \frac{20\text{KHz}}{2^{WDTM}}$$

$$\text{Watchdog reset time} = \frac{256}{WDTCLK}$$

InitWDT:

软件需依序于寄存器 TAKEY 写入 55h、AAh 及 5Ah 才能对看门狗控制寄存器 (WDTC) 执行写入:

TAKEY = 0x55;

TAKEY = 0xAA;

TAKEY = 0x5A;

设置 WDT 使能和复位时间 51.2ms:

WDTC = 0x24;

KickWDT:

看门狗喂狗, 复位计数器:

WDTK = 0x55;

3. GPIO 配置

OB6619 有 4 个 I/O 口: Port 0、Port 1、Port 2 与 Port 3。它们是: 准双向口 (标准的 8051 端口输出)、推挽电路、开漏、与只输入。两种寄存器 (POM0、POM1)、(P1M0、P1M1)、(P2M0、P2M1)、(P3M0、P3M1) 的配置为每个端口的每个位选择输出方式, 如下表 1 所示:

PxM1.y	PxM0.y	Port output mode
0	0	准双向口 (标准的8051端口输出)
0	1	推挽电路
1	0	只输入 (high-impedance)
1	1	开漏

表 1 端口功能选择

注: IO 口初始态设置可透过烧录软件内 ” 芯片配置 ” 设置窗口中的 ” IO 输出模式 ” 选项, 设置 MCU 初始时 3 个 I/O 口组态: Port 1, Port 2 与 Port 3 为 ” 准双向口 (标准的 8051 端口输出) ” 或 ” 只输入 ” 模式。

4. Timer0/ Timer1 模块

OB6619 有二个 16bit 的定时/计数寄存器：定时器 0、定时器 1。所有这些都
可以被设置为定时或计数操作。以定时器的模式 1, 2ms 定时器为例，如图 2-3：
模式 1 -16 位定时器操作。

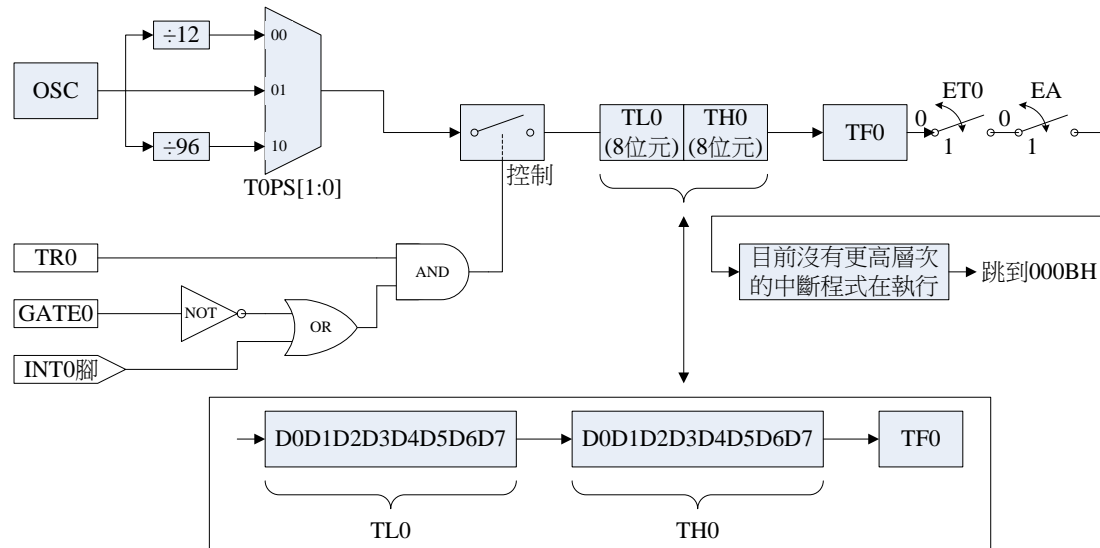


图 2-3：模式 1 -16 位定时器操作

InitTimer0:

```

TMOD = 0X01; // 16 bit timer mode
PFCON = 0X01; // timero freq Fosc/1 不分频
TL0=0x7F;TH0=0x44; //Set InitCounter
TR0 = 1; // Start Timer
TF0 = 0; // Clear Timer Interrupt Flag
IEN0 = 0X02; // Enable Timer0 Interrupt
EA = 1; // Enable All Interrupt
    
```

5. ADC 模块配置

OB6619 提供 13 通道的 12 位 ADC，数字输出采样的模拟信号放入 ADCRS0~
ADCRS3 中，ADC 的操作框图设置如图 2-4 所示。

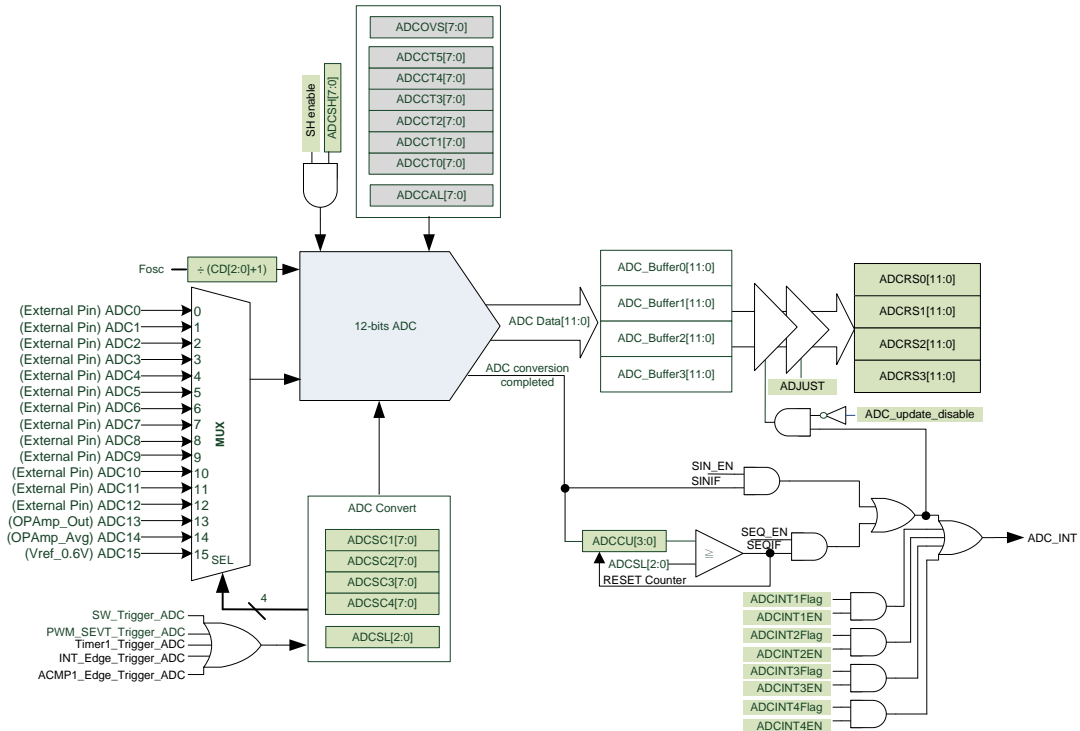


图 2-4: ADC 模拟到数字转换器的操作设置

以 PWM 硬件触发 4 channel ADC 为例说明其相应配置，如下图 2-5 ADC 为其时序图。

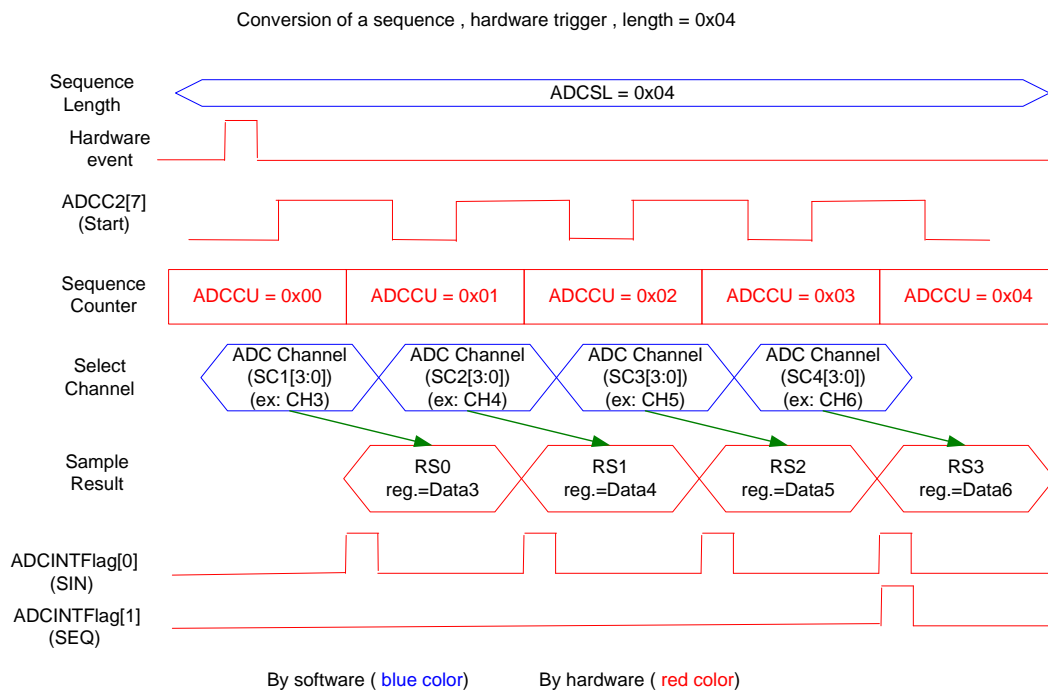


图 2-5: ADC 模拟到数字转换器时序图

InitADC:

```

ADCINTEN = 0x02;           // Enable ADC
ADCCD = 0x80;             // 数据存储方式和 ADC 时钟选择不分频
ADCC2 = 0x02;            // 采样保持 Enable
while(1)
    
```

```
{
    if(ADCCAL == 0x02)    //wait  ADC  Calibration
        break;
}

ADCINTFlag = 0x00;      // Clear ADC 完成 Flag
SEVTCMP = 0;           // PWM ZERO Trigger ADC
ADCSC1 = 0x00;         // ADC channel select
ADCSC2 = 0x01;         // ADC channel select
ADCSC3 = 0x02;        // ADC channel select
ADCSC4 = 0x03;        // ADC channel select
ADCSL = 4;             //ADC sequence length
```

GetADCValue:

```
while((ADCINTFlag & 0x02) == 0x02)    //completion of a conversion on sequence
    break;
TestData1.Buff_BYTE[1] = ADCRS0L;    //Read channel 1 Data
TestData1.Buff_BYTE[0] = ADCRS0H;

TestData2.Buff_BYTE[1] = ADCRS1L;    //Read channel 2 Data
TestData2.Buff_BYTE[0] = ADCRS1H;

TestData3.Buff_BYTE[1] = ADCRS2L;    //Read channel 3 Data
TestData3.Buff_BYTE[0] = ADCRS2H;

TestData4.Buff_BYTE[1] = ADCRS3L;    //Read channel 4 Data
TestData4.Buff_BYTE[0] = ADCRS3H;
```

注：上例程是以 PWM ZERO 触发 ADC 转换，如不要触发 ADC 需将 SEVTCMP 的值设为大于 PWM PERIOD。ADCC2 = 0x82，开始软件转换。

6. PWM 模块

PWM 模块的特征。

- 六通道（四对）PWM 输出引脚。
- 14-bit 解析。
- 中心和边缘对齐方式输出模式。
- 死计(dead time)时产生器。
- 脉宽调制和特殊事件中断触发器。
- 电机控制有输出改写功能。
- 故障保护（FLTA 和 FLTB）。

以下是 PWM 的工作模块图，如图2-6中所示。

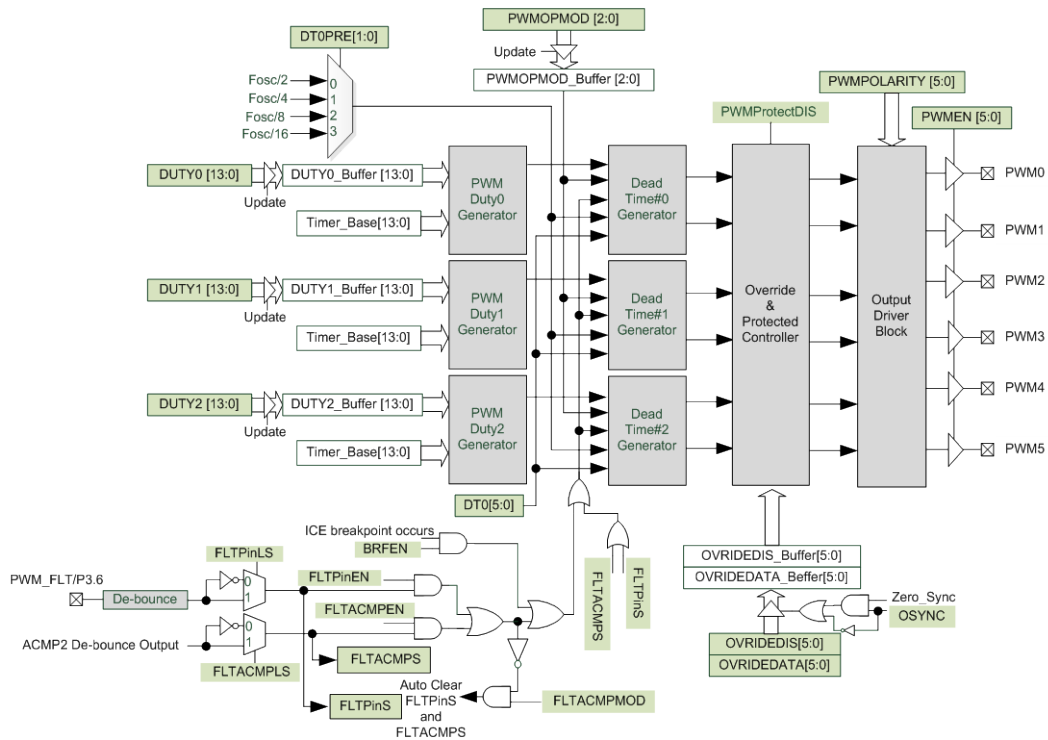


图 2-6 PWM 的工作模块图

InitPWM:

```

PWMEN = 0X00;           // Disable PWM;
PWMPOLARITY = 0X3F;    // PWM0~PWM5 Polarity Output High
PWMOPMOD= 0X07;        // PWM0~PWM5 Independent Mode
PWMTBC0= 0x02;         // PWM Counter UPDOWN Mode
PWMPERIOD = 1200;      // PWM PRDLOWFRQ
TBCOUNTERL = 0x00;     // Reset 14-bit Time base counter low 8 bit
TBCOUNTERH = 0x00;    // Reset 14-bit Time base counter high 6 bit

SEVTCMP = 0;           // PWM ZERO Trigger ADC
PWMSEV= 0;             // PWM Output No Buffer ; Evey PWM Cycly Trigger ADC;

PWMTBPOSTSCALE = 0x00; // PWM Counter 不分频

DEADTIME0= 0x00;      // Set DEAD_TIME;

OVRIDEDATA= 0x00;
OVRIDEDIS= 0x00;
PWMTBC1= 0X81; // Enable PWM Counter ; Enable PWM Intruput;

PWMINTF= 0x00; // Clear PWM Intruput Flag
FLTCONFIG = 0X90; // Enable Fault B
FLTNF = 0x20 ; // FaultB 高电平动作
    
```

```
FLTPINDBCNT = 20;//Debounce count for N*TCLK =N*41.6ns
```

StepSixAutoCommSet: 上桥 PWM 调制，下桥强制输出高低电平

```
uint8 pdata lib_FWDBEMFSeqTbl[6]={2,3,1,5,4,6};
```

```
uint8 pdata lib_BEMFFWDOvrdisTrapTbl[8]=
{
    0x00,    // Invalid !
    0x04,    // // H-PWM2
    0x10,    // // H-PWM4
    0x04,    // // H-PWM2
    0x01,    // // H-PWM0
    0x01,    // // H-PWM0
    0x10,    // // H-PWM4
    0x00    // Invalid !
};
```

```
uint8 pdata lib_BEMFFWDOvrdatTbl[8]=
{
    0x00,    // Invalid !
    0x20,    // L-PWM5
    0x02,    // L-PWM1
    0x02,    // L- PWM1
    0x08,    // L-PWM3
    0x20,    //L- PWM5
    0x08,    //L- PWM3
    0x00    // Invalid !
};
```

```
PWMDUTY0 = 300; // SetDuty
PWMDUTY1 = 300; // SetDuty
PWMDUTY2 = 300; // SetDuty
```

```
if(PosIdx>=5)
{
    PosIdx = 0;
}
else
{
    PosIdx++;
}
lib_pos = lib_FWDBEMFSeqTbl[PosIdx];
```

```
OVRIDEDIS = lib_BEMFFWDOvrdisTrapTbl[lib_pos]; // 上桥输出
OVRIDEDATA = lib_BEMFFWDOvrdatTbl[lib_pos]; // 下桥输出
```

7. 比较器

OB6619 已集成比较器在芯片上。當使用它作为比较器時，正输入大于负输入，比较器输出是一个有效电平，并可以与 PWM 输出配合产生锁存功能。下图 2-7 是比较器 1 的工作运作框图。

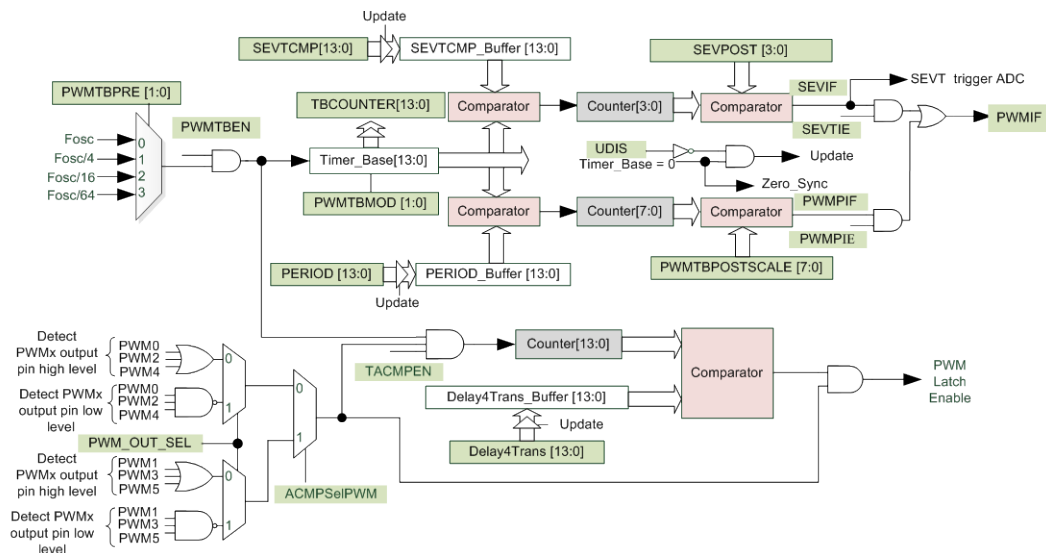


图 2-7 比较器 1 的工作运作框图

以比较器 1 与 PWM 输出配合产生锁存功能为例说明。

InitACMP1:

```
Delay4TransL = 0; // PWM Output rising edge 时启动计数器， Delay Start COMP Latch
Delay4TransH = 0;
```

```
ACMP1PSEL = 0x01; // 比较器反相输入通道选择 CMP_PIN1
```

```
ACMPHYS = 0x03; // 比较器磁滞 30mV
```

```
ACMP1DBCNT = 120; //Debounce count for ACMP1 output N*TCLK =N*41.6ns
```

```
ACMP1OUTPUT = 0x 09; // PWM Trigger 比较器， Enable 比较器 Latch 功能
```

注：如需要比较器延时锁存功能，需给寄存器 Delay4TransL、Delay4TransH 配置初始值，其寄存器默认值为 0xFF。

以比较器 2 结合运放为例说明硬件过流保护，如下图 2-8 硬件保护电路框架图。

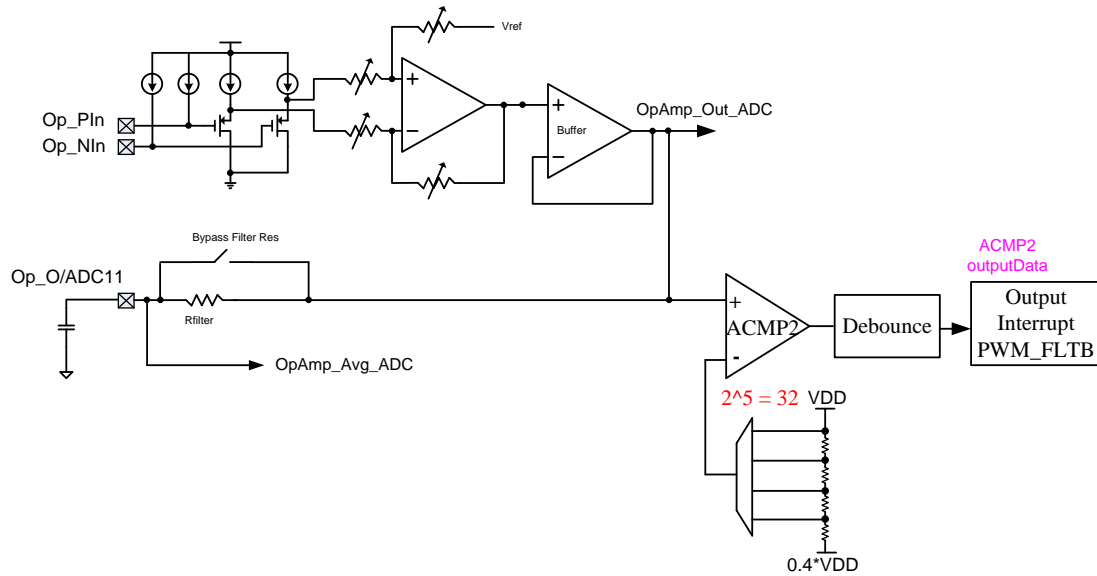


图 2-8 硬件保护电路框架图

InitACMP2:

```

ACMP2NSEL = 0x01;    //Enable Cmp2 ;反相输入 0.4V*VDD
ACMP2OUTPUT = 0x40; //to set interrupt enable (for ACMP2 rising edge)
ACMP2DBCNT = 20;    //Debounce count for ACMP2 output N*TCLK =N*41.6ns
    
```

Check ACMP2Output:

```

if((ACMP2OUTPUT&0x80)==0x80) // Check CMP2OUTPUT
{
    OVRIDEDIS = 0x00; // Close PWMOUT
    OVRIDEDATA = 0x00;
    DisablePWM();
    ClearACMP2IsrFlg(); // Clear CMP2 interrupt Flag
}
    
```

注：比较器 2 输出到 PWM 模块 FaultB。

8. 运算放大器 (OPA)

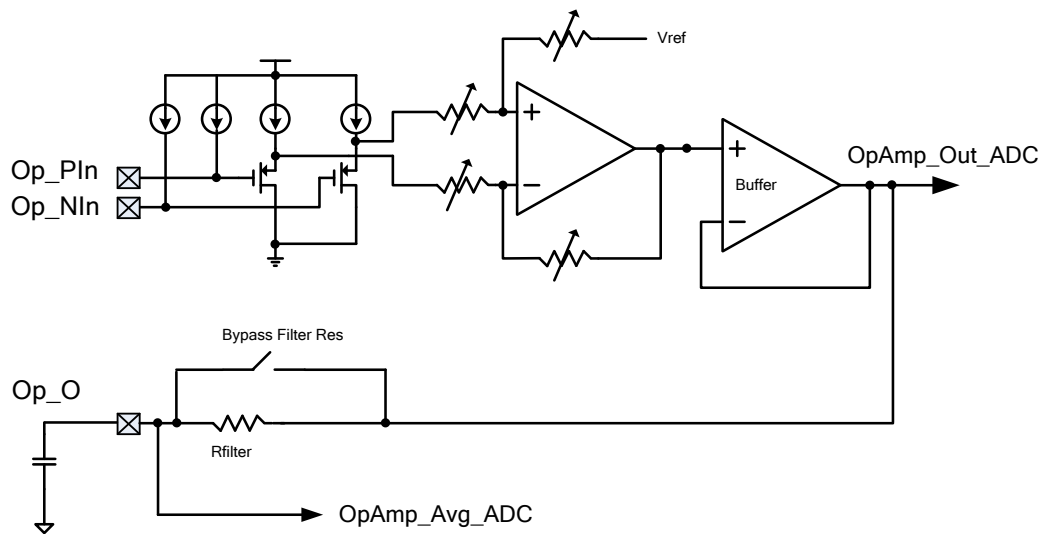


图 2-8 OPA 的工作运作框图

InitOPA:

OPACON = 0x41; // Enable LPF Offset 0.4V 放大倍数 16 Enable OPA

注：需测量运放输出 OpAmp_Out_ADC，可 Disable LPF。