

图64 输出模块框图

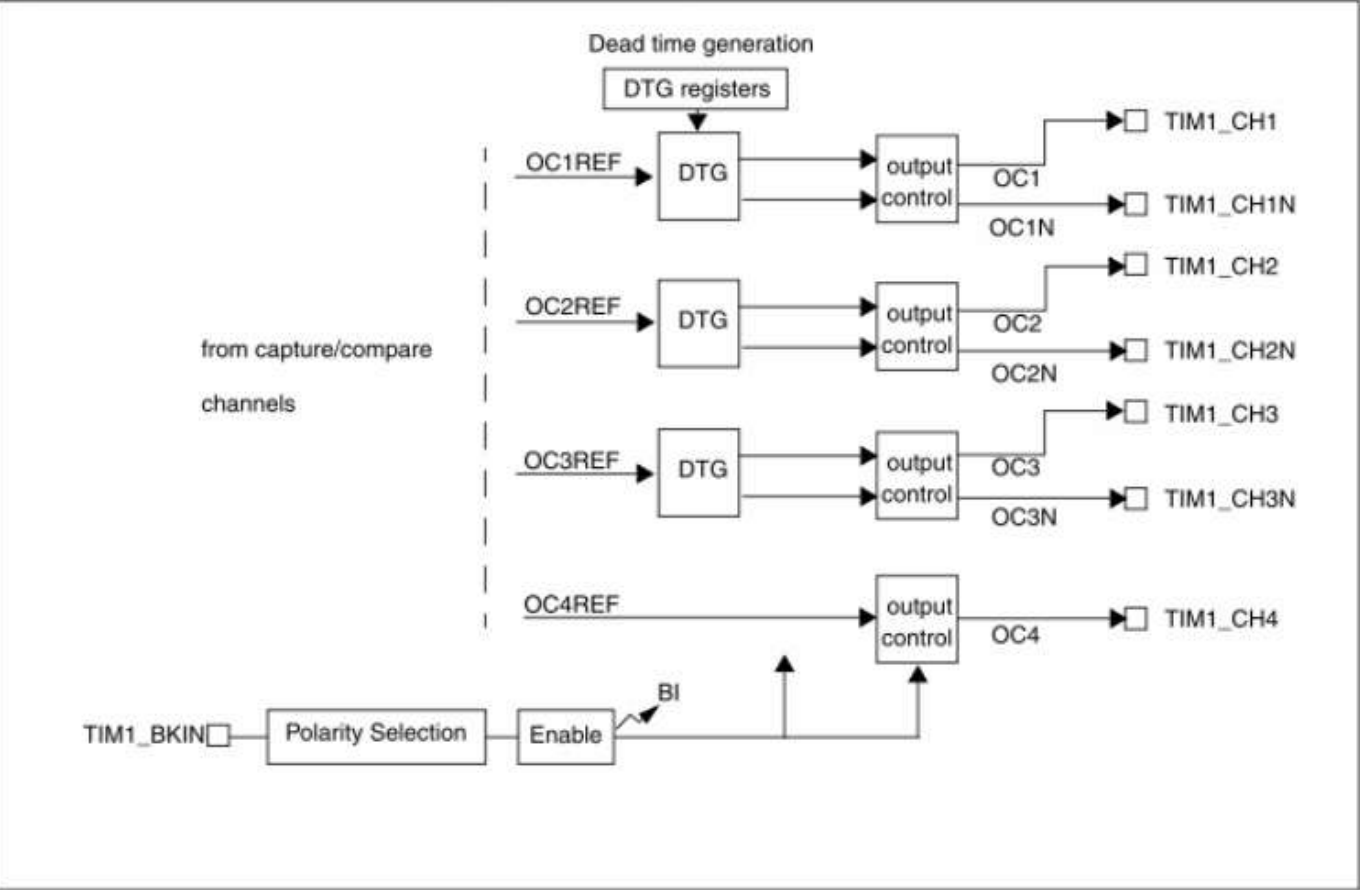
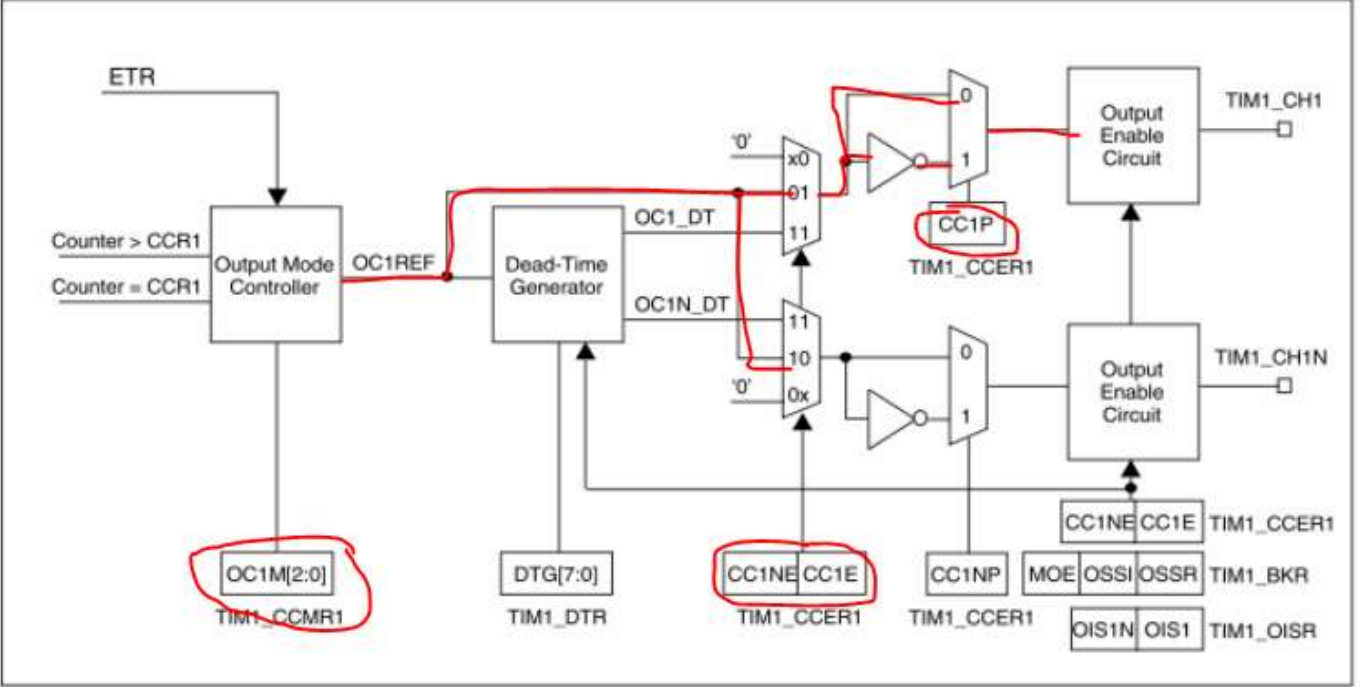


图65 详细的带互补输出的输出模块框图(通道1)



不考虑死区和互补输出，则如图所示，只需要配置：

1. CCMR_OC1M的输出模式
2. CCER_CC1NE:CCER_CC1E = 01;
3. CCER_CC1P = 高电平，低电平;
- 4.

CCMR1:

位7	<p>OC1CE: 输出比较1清零使能</p> <p>该位用于使能使用TIM1_TRIG引脚上的外部事件来清通道1的输出信号(OC1REF), 参考17.5.9在外部事件发生时清除OCREF信号</p> <p>0: OC1REF 不受ETRF输入(来自TIM1_TRIG引脚)的影响;</p> <p>1: 一旦检测到ETRF输入高电平, OC1REF=0。</p>
位6:4	<p>OC1M[2:0]: 输出比较1模式</p> <p>该3位定义了输出参考信号OC1REF的动作, 而OC1REF决定了OC1的值。OC1REF是高电平有效, 而OC1的有效电平取决于CC1P位。</p> <p>000: 冻结。输出比较寄存器TIM1_CCR1与计数器TIM1_CNT间的比较对OC1REF不起作用;</p> <p>001: 匹配时设置通道1的输出为有效电平。当计数器TIM1_CNT的值与捕获/比较寄存器1(TIM1_CCR1)相同时, 强制OC1REF为高。</p> <p>010: 匹配时设置通道1的输出为无效电平。当计数器TIM1_CNT的值与捕获/比较寄存器1(TIM1_CCR1)相同时, 强制OC1REF为低。</p> <p>011: 翻转。当TIM1_CCR1=TIM1_CNT时, 翻转OC1REF的电平。</p> <p>100: 强制为无效电平。强制OC1REF为低。</p> <p>101: 强制为有效电平。强制OC1REF为高。</p> <p>110: PWM模式1— 在向上计数时, 一旦TIM1_CNT<TIM1_CCR1时通道1为有效电平, 否则为无效电平; 在向下计数时, 一旦TIM1_CNT>TIM1_CCR1时通道1为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。</p> <p>111: PWM模式2— 在向上计数时, 一旦TIM1_CNT<TIM1_CCR1时通道1为无效电平, 否则为有效电平; 在向下计数时, 一旦TIM1_CNT>TIM1_CCR1时通道1为有效电平, 否则为无效电平。</p> <p>注1: 一旦LOCK级别设为3(TIM1_BKR寄存器中的LOCK位)并且CC1S=00(该通道配置成输出)则该位不能被修改。</p> <p>注2: 在PWM模式1或PWM模式2中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到PWM模式时, OC1REF电平才改变。(参考17.5.7PWM模式)</p> <p>注3: 在有互补输出的通道上, 这些位是预装载的。如果TIM1_CR2寄存器的CCPC=1, OCM位只有在COM事件发生时, 才从预装载位取新值。</p>
位3	<p>OC1PE: 输出比较1预装载使能</p> <p>0: 禁止TIM1_CCR1寄存器的预装载功能, 可随时写入TIM1_CCR1寄存器, 并且新写入的数值立即起作用。</p> <p>1: 开启TIM1_CCR1寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR1的预装载值在更新事件到来时被加载至当前寄存器中。</p> <p>注1: 一旦LOCK级别设为3(TIM1_BKR寄存器中的LOCK位)并且CC1S=00(该通道配置成输出)则该位不能被修改。</p> <p>注2: 为了操作正确, 在PWM模式下必须使能预装载功能。但在单脉冲模式下(TIM1_CR1寄存器的OPM=1), 它不是必须的。</p>
位2	<p>OC1FE: 输出比较1快速使能</p> <p>该位用于加快CC输出对触发输入事件的响应。</p> <p>0: 根据计数器与CCR1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。</p> <p>OCFE只在通道被配置成PWM1或PWM2模式时起作用。</p>
位1:0	<p>CC1S[1:0]: 捕获/比较1 选择。</p> <p>这2位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1FP1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2FP1上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIM1_SMCR寄存器的TS位选择)。</p> <p>注: CC1S仅在通道关闭时(TIM1_CCER1寄存器的CC1E=0)才是可写的。</p>

CCER1:

位7	CC2NP: 输入捕获/比较2互补输出极性。参考CC1NP的描述。
位6	CC2NE: 输入捕获/比较2互补输出使能。参考CC1NE的描述。
位5	CC2P: 输入捕获/比较2输出极性。参考CC1P的描述。
位4	CC2E: 输入捕获/比较2输出使能。参考CC1E的描述。
位3	<p>CC1NP: 输入捕获/比较1互补输出极性</p> <p>0: OC1N高电平有效;</p> <p>1: OC1N低电平有效。</p> <p>注1: 一旦LOCK级别(TIM1_BKR寄存器中的LCCK位)设为3或2且CC1S=00(通道配置为输出)则该位不能被修改。</p> <p>注2: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1 (TIM1_CR2寄存器), 只有在COM事件发生时, CC1NP位才从预装载位中取新值。</p>
位2	<p>CC1NE: 输入捕获/比较1互补输出使能</p> <p>0: 关闭— OC1N禁止输出, 因此OC1N的输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的值。</p> <p>1: 开启— OC1N信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的值。</p> <p>注: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1(TIM1_CR2寄存器), 只有在COM事件发生时, CC1NE位才从预装载位中取新值。</p>
位1	<p>CC1P: 输入捕获/比较1输出极性</p> <p>CC1通道配置为输出:</p> <p>0: OC1高电平有效;</p> <p>1: OC1低电平有效。</p> <p>CC1通道配置为触发(参考图61):</p> <p>0: 触发发生在TI1F的高电平或上升沿;</p> <p>1: 触发发生在TI1F的低电平或下降沿。</p> <p>CC1通道配置为输入(参考图61):</p> <p>0: 捕捉发生在TI1F的高电平或上升沿;</p> <p>1: 捕捉发生在TI1F的低电平或下降沿。</p> <p>注1: 一旦LOCK级别(TIM1_BKR寄存器中的LCCK位)设为3或2, 则该位不能被修改。</p> <p>注2: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1 (TIM1_CR2寄存器), 只有在COM事件发生时, CC1P位才从预装载位中取新值。</p>
位0	<p>CC1E: 输入捕获/比较1输出使能</p> <p>CC1通道配置为输出:</p> <p>0: 关闭— OC1禁止输出, 因此OC1的输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。</p> <p>1: 开启— OC1信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。</p> <p>CC1通道配置为输入:</p> <p>该位决定了计数器的值是否能捕获入TIM1_CCR1寄存器。</p> <p>0: 捕获禁止;</p> <p>0: 捕获使能。</p> <p>注: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1(TIM1_CR2寄存器), 只有在COM事件发生时, CC1E位才从预装载位中取新值。</p>

BKR:

位7	<p>MOE: 主输出使能</p> <p>一旦刹车输入有效, 该位被硬件异步清0。根据AOE位的设置值, 该位可以由软件置1或被自动置1。它仅对配置为输出的通道有效。</p> <p>0: 禁止OC和OCN输出或强制为空闲状态;</p> <p>1: 如果设置了相应的使能位(TIM1_CCERX寄存器的CC/E位), 则使能OC和OCN输出。</p> <p>有关OC/OCN使能的细节, 参见17.7.13。</p>
----	--

```

#include "type_def.h"
void main(void)
{
    #if 0
        TIM2_PSCR = 0X00;
        TIM2_ARRH = 0X00;
        TIM2_ARRL = 136;
        TIM2_CR1 &= 0XFE;
        CLK_CKDIVR&= (uint8_t) (~0x18); /*使能内部时钟*/
        CLK_CKDIVR|= (uint8_t) 0x00; /*设置时钟为内部16M高速时钟*/
        TIM2_CCER1 &= 0xFC;
        TIM2_CCER1 |= 0x01;
        TIM2_CCMR1 &= 0x8F;
        TIM2_CCMR1 |= 0x6C;
        TIM2_CCR1H = 0X00;
        TIM2_CCR1L = 54;
        TIM2_CR1 |= 0X01;
        while (1); //
    #else
        TIM1_PSCRH = 0X00;
        TIM1_PSCRL = 0X00;
        TIM1_ARRH = 0X00;
        TIM1_ARRL = 136;
        TIM1_CR1 &= 0XFE;

        TIM1_BKR = 0X80; //这个是TIM1和TIM2的最大的区别

        CLK_CKDIVR&= (uint8_t) (~0x18); /*使能内部时钟*/
        CLK_CKDIVR|= (uint8_t) 0x00; /*设置时钟为内部16M高速时钟*/
        TIM1_CCER1 &= 0xFC;
        TIM1_CCER1 |= 0x01;
        TIM1_CCMR1 &= 0x8F;
        TIM1_CCMR1 |= 0x6C;
        TIM1_CCR1H = 0X00;
        TIM1_CCR1L = 54;
        TIM1_CR1 |= 0X01;
        int i;
        while (1)
        {
            //TIM1_CCER1 &= 0xFE; //关闭TIM1输出
            for(i = 0;i<2000;i++); //证明可以不影响pwm输出
            //TIM1_CCER1 |= 0x01; //开启TIM1输出
            for(i = 0;i<2000;i++);

        }
    #endif
}

```