# ST MCSDK学习笔记

## hall\_speed\_pos\_fdbk

根据HALL1/2/3 跳变捕获时间，计算得到电机当前转速。

该工程使用STM32的TIM2定时器的CH1、CH2、CH3分别接HALL1、HALL2、HALL3，采用(Hall Sensor Mode)/XOR ON三个引脚异或来捕获跳变时间。

PseudoFreqConv 为转换系数

PseudoFreqConv = ((TIMClockFreq / 6)/hMeasurementFrequency) \* (DPPConvFactor);

SensorPeriod[SpeedBufferSize]为SpeedBufferSize个TIM2最近捕获的周期数

SensorPeriod[SpeedFIFOIdx] = (TIM2->PSC + 1)\*TIM2->CCR1;

ElPeriodSum 为最近SensorPeriod[SpeedBufferSize]累加值

hElSpeedDpp 为dpp（digit-per-PWM） 每个测量周期的电角度（65536表示360度）

hAvrMecSpeedUnit 为电机速度，单位0.1rps(0.1圈/秒)

AvrElSpeedDpp = PseudoFreqConv / ( ElPeriodSum / SpeedBufferSize );

hElSpeedDpp = AvrElSpeedDpp ;

令 kElSpeedDpp2MecSpeedUnit = (hMeasurementFrequency \* SPEED\_UNIT ) / ( DPPConvFactor \* bElToMecRatio );

hAvrMecSpeedUnit = AvrElSpeedDpp \* kElSpeedDpp2MecSpeedUnit ;

hMaxReliableMecSpeedUnit为速度报错上限值，单位0.1rps

hMinReliableMecSpeedUnit为速度报错下限值，单位0.1rps

## **speed\_torq\_ctrl**

目标速度接收，解析串口数据帧

**ClientFrameReceivedCallback**

**-->MCP\_ReceivedFrame**( pBaseHandle->ClientEntity,

pBaseHandle->RxFrame.Code,

pBaseHandle->RxFrame.Buffer,

pBaseHandle->RxFrame.Size );

**MCI\_ExecSpeedRamp**(pMCI,(int16\_t)((wValue\*SPEED\_UNIT)/\_RPM),0);

单位转换：rpm转换为0.1rps

pMCI->hTargetFinal = (wValue\*SPEED\_UNIT)/\_RPM;

SpeedRefUnitExt = ( int32\_t )pMCI->hTargetFinal \* 65536;

速度环PI控制，以及FOC控制

FOC\_CalcCurrRef( M1 );

**-->STC\_CalcTorqueReference**(pSTC[bMotor]);

wCurrentReference = SpeedRefUnitExt;

hTargetSpeed = ( int16\_t )( wCurrentReference / 65536 );

hMeasuredSpeed = hAvrMecSpeedUnit ;

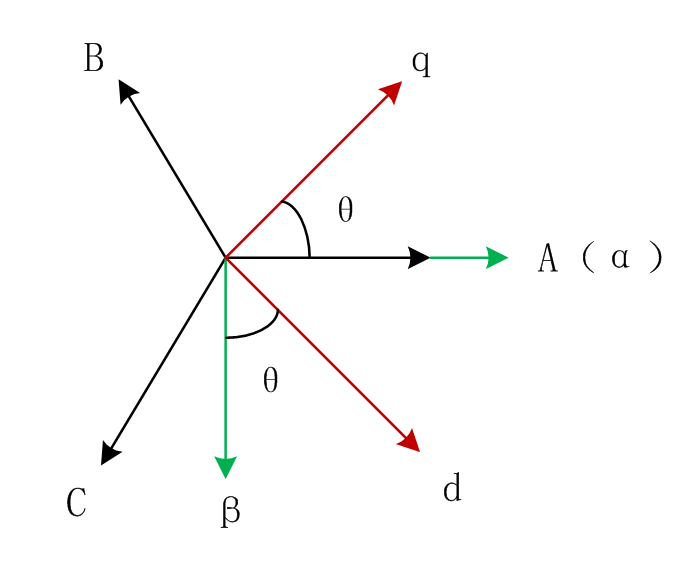
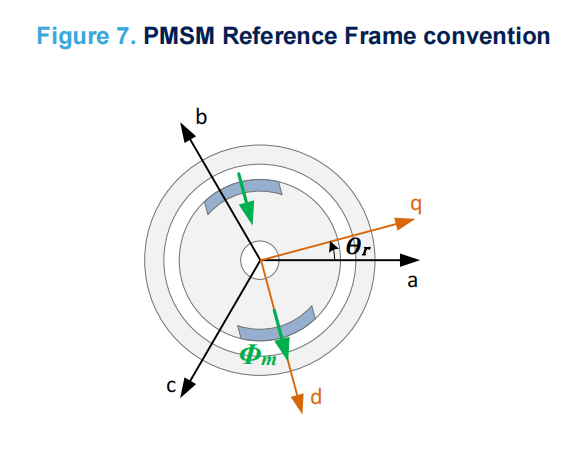
hError = hTargetSpeed - hMeasuredSpeed;

hTorqueReference = PI\_Controller( pHandle->PISpeed, ( int32\_t )hError );

SpeedRefUnitExt = wCurrentReference;

TorqueRef = ( int32\_t )hTorqueReference \* 65536;

## 3、ST-MCSDK坐标转换



**MCM\_Clarke变换**

**MCM\_Park变换**



**MCM\_Rev\_Park**

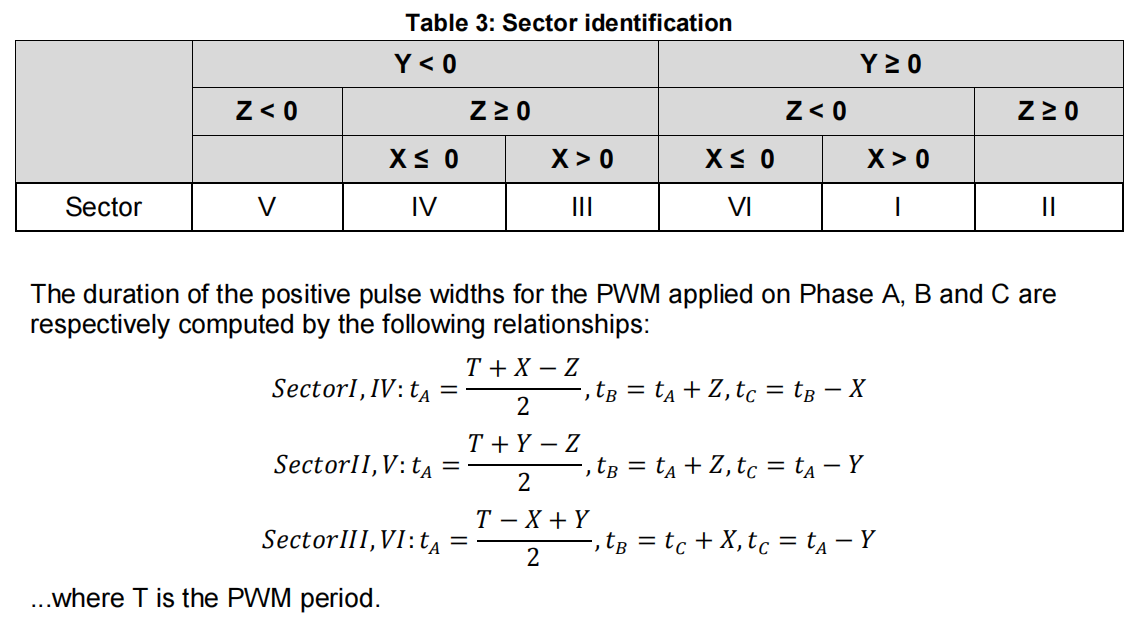
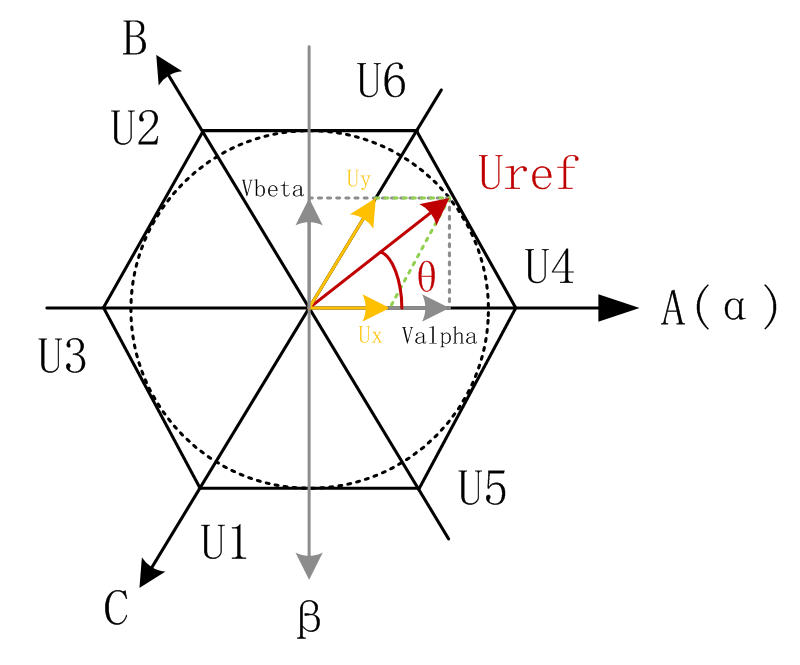


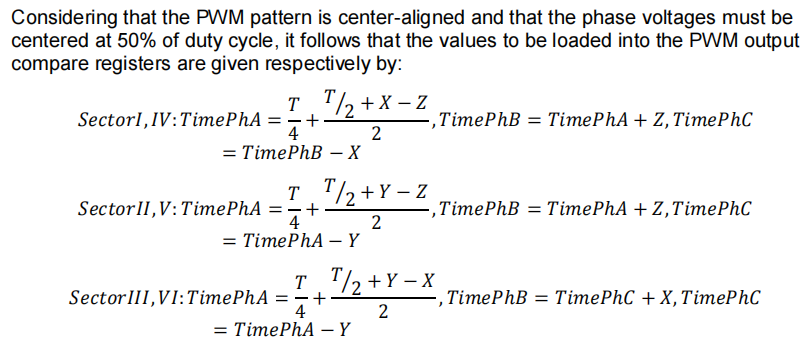
**P**WMC\_SetPhaseVoltag**e**



辅助变量







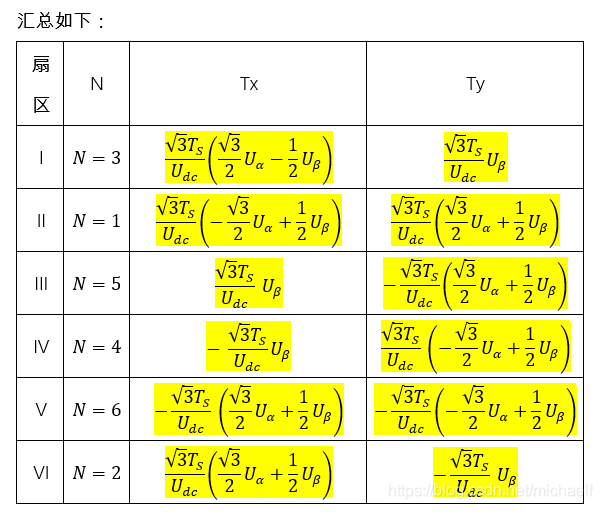
扇区I分析



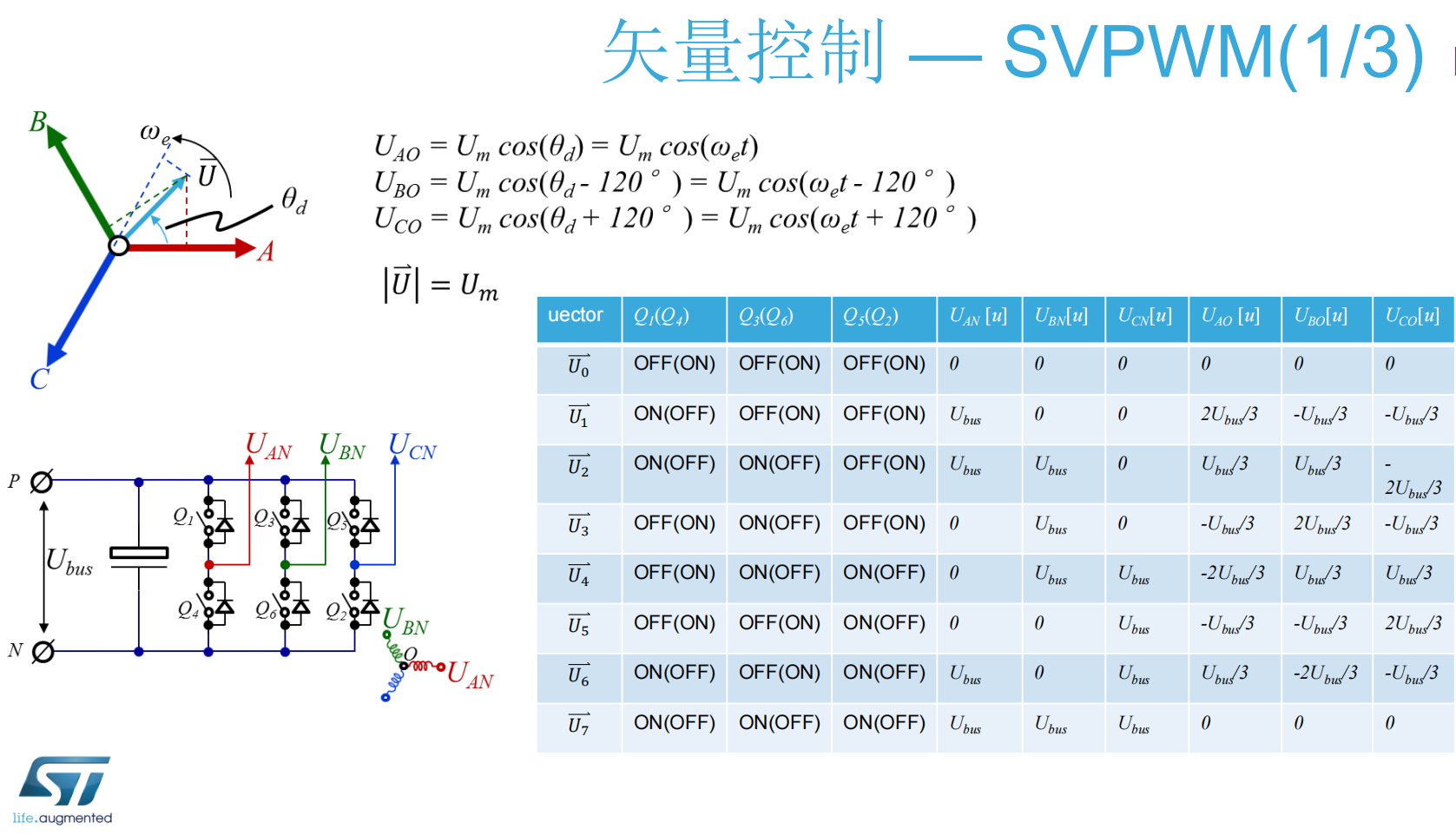
 

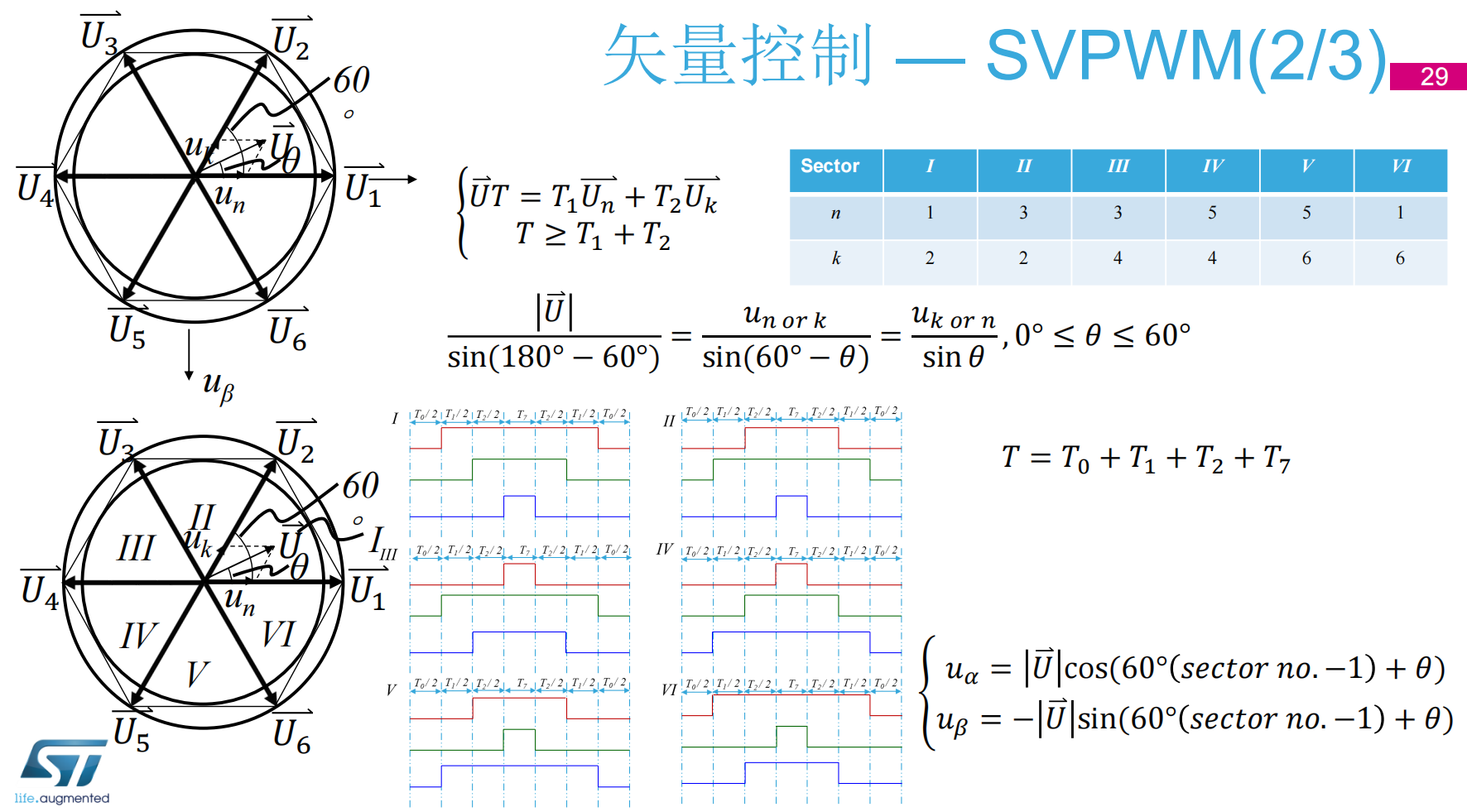
wUAlpha 和 wUBeta 都乘以系数2\*PWMperiod ，即wX、wY、wZ都包含2T的系数。但是因为STM32采用的是中央对齐模式，所以设置减小为原来的1/2。

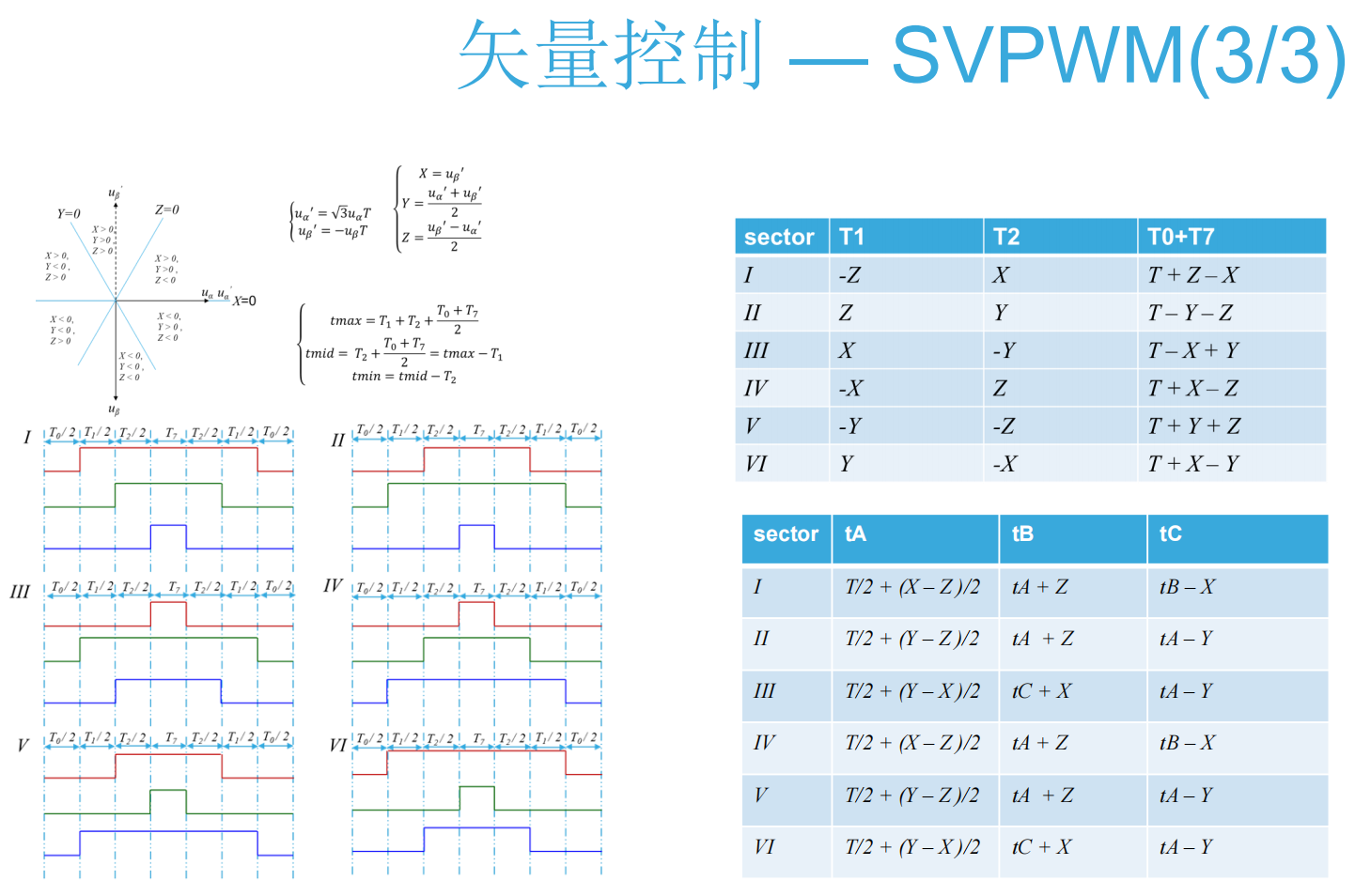




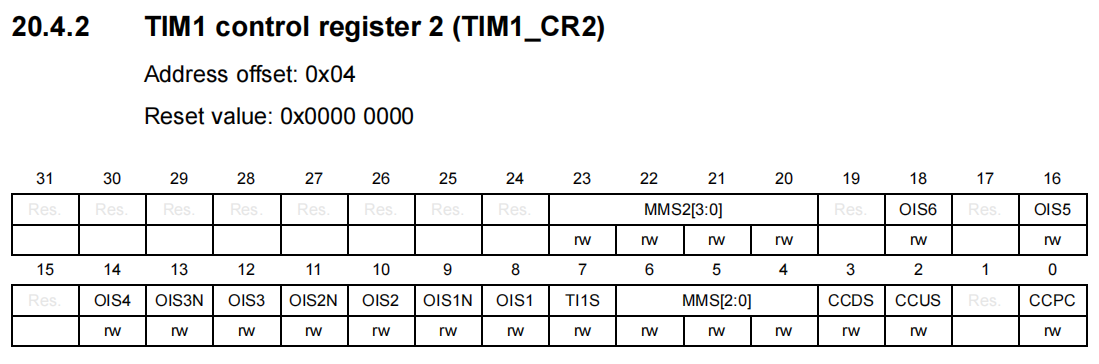
注意：该汇总为第一象限的计算公式，第四象限后续更新再补上。

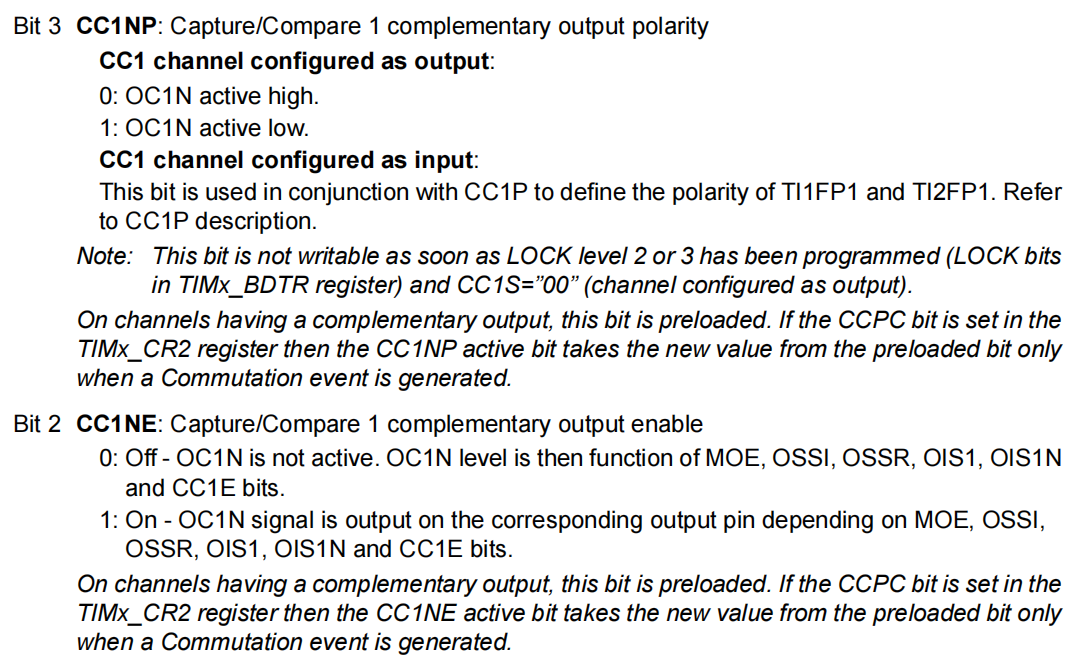


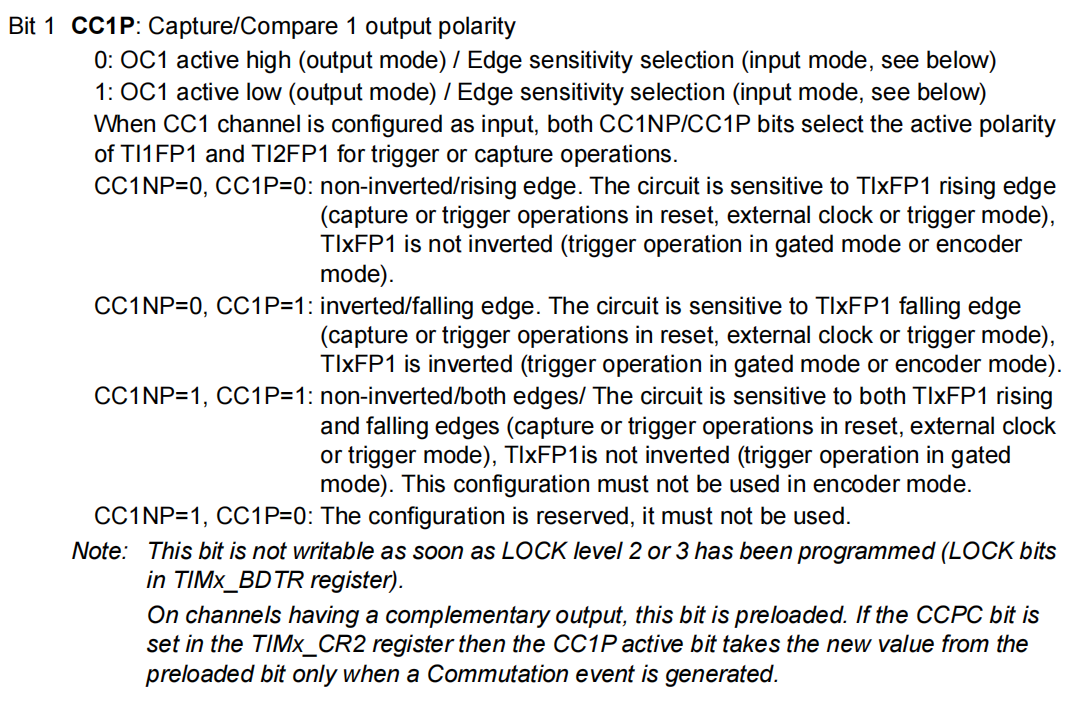


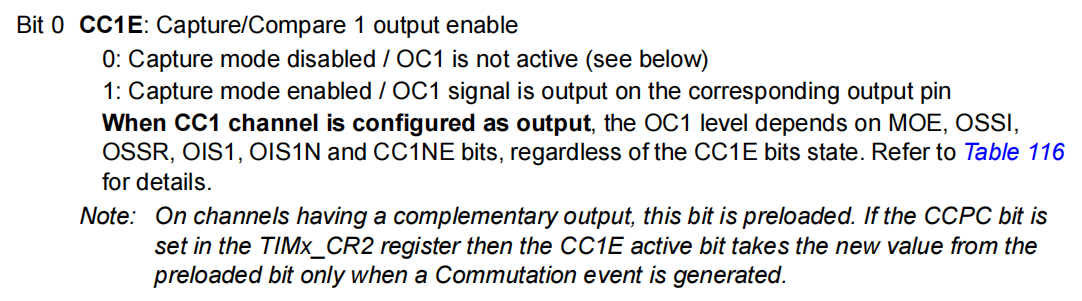


1. B、C、AN、BN、CN极性与使能配置(TIMx\_CR2 = 0x2A00)

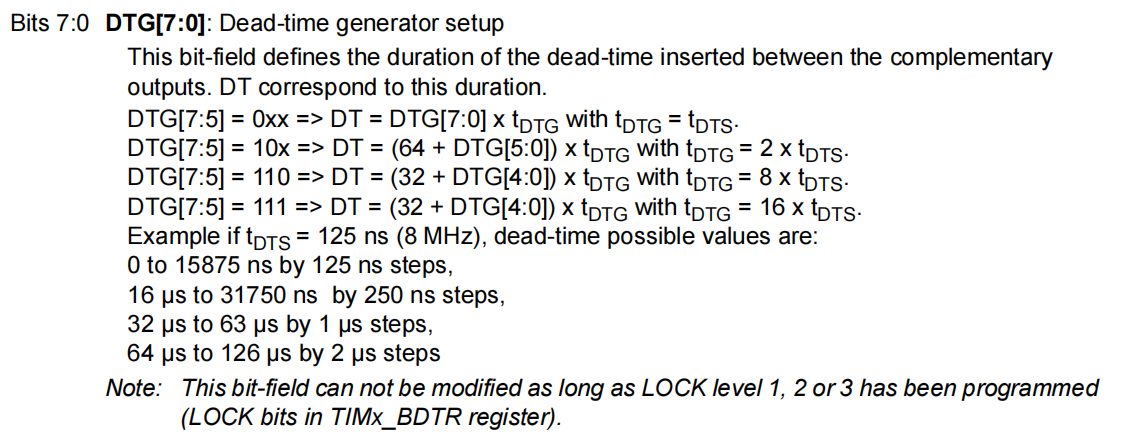


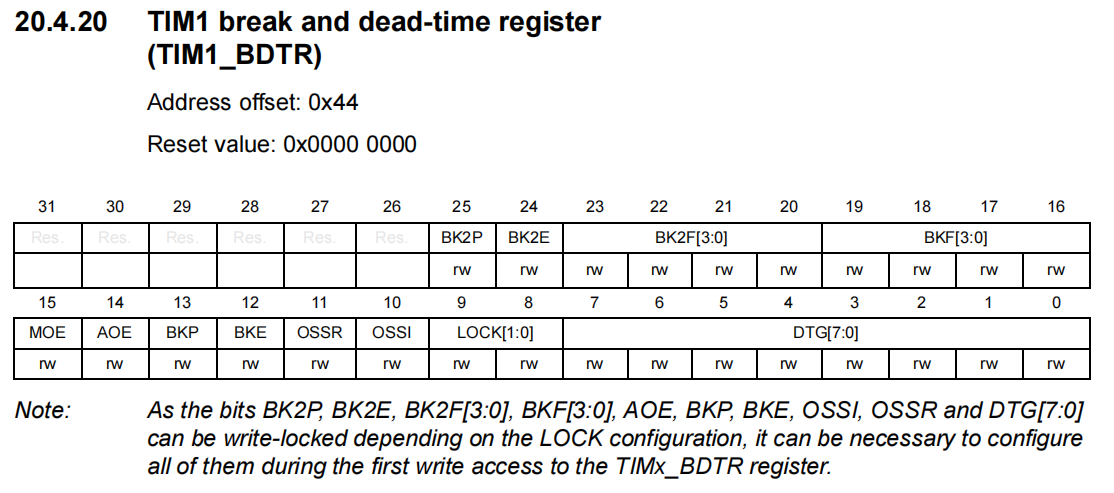


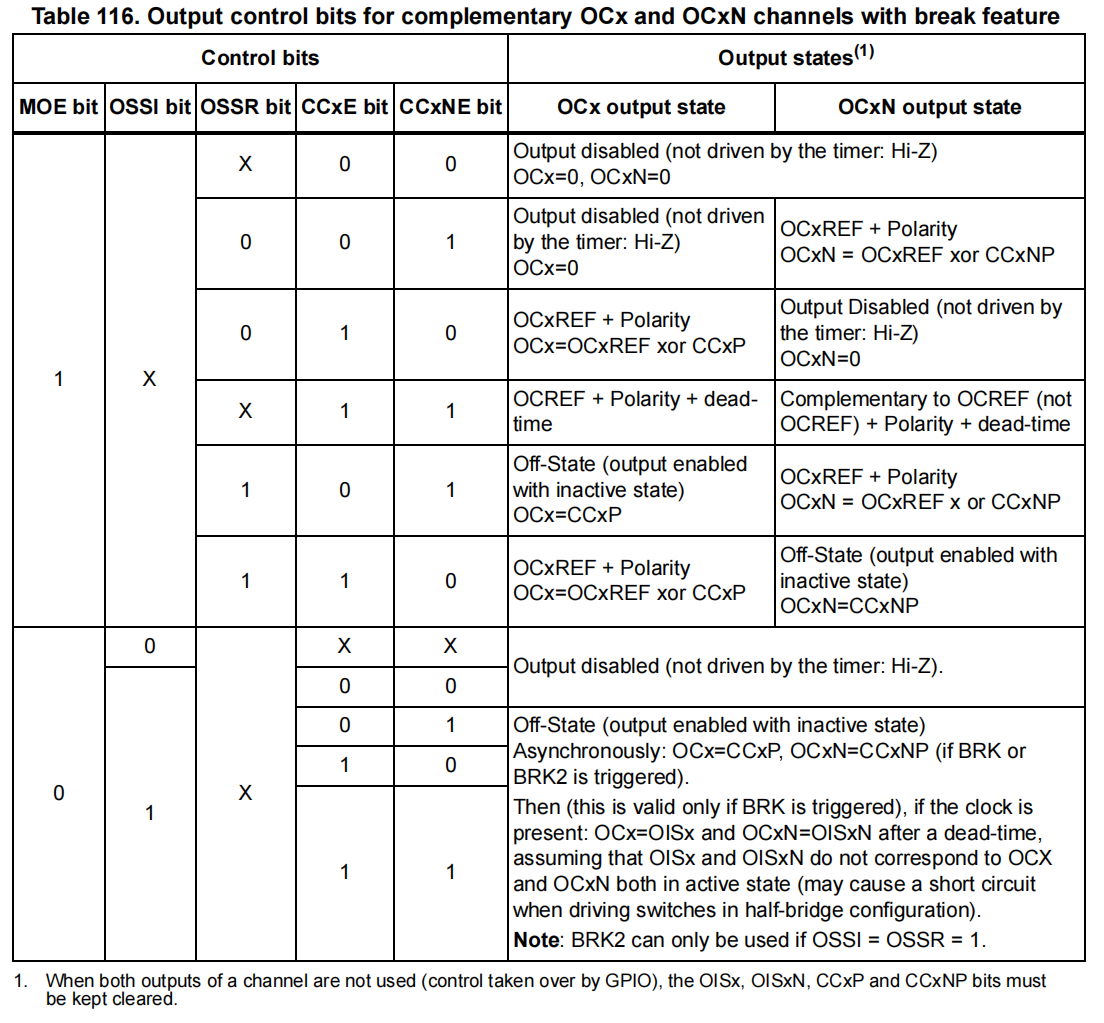


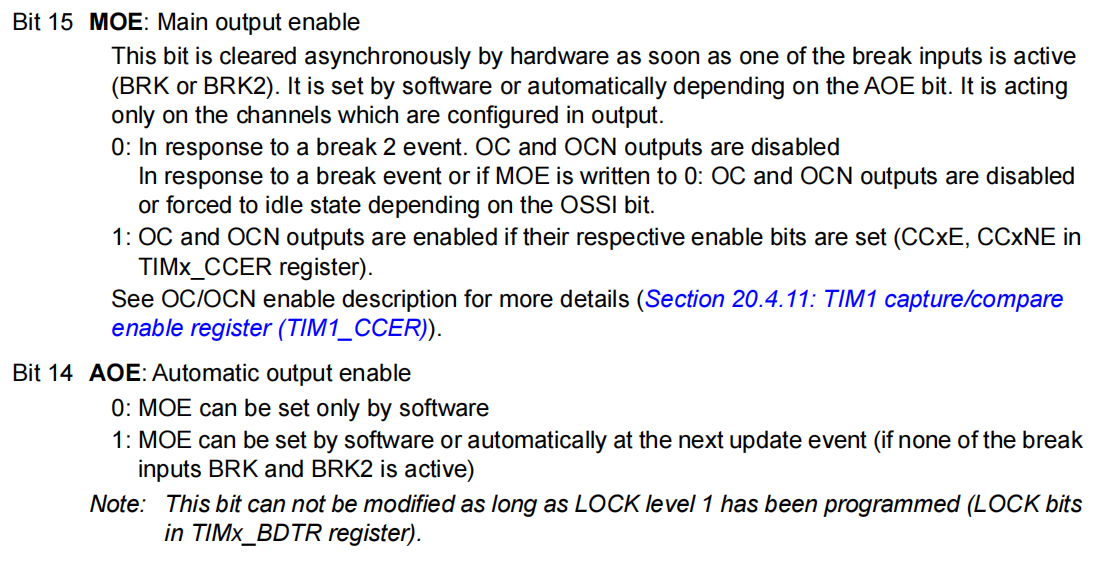


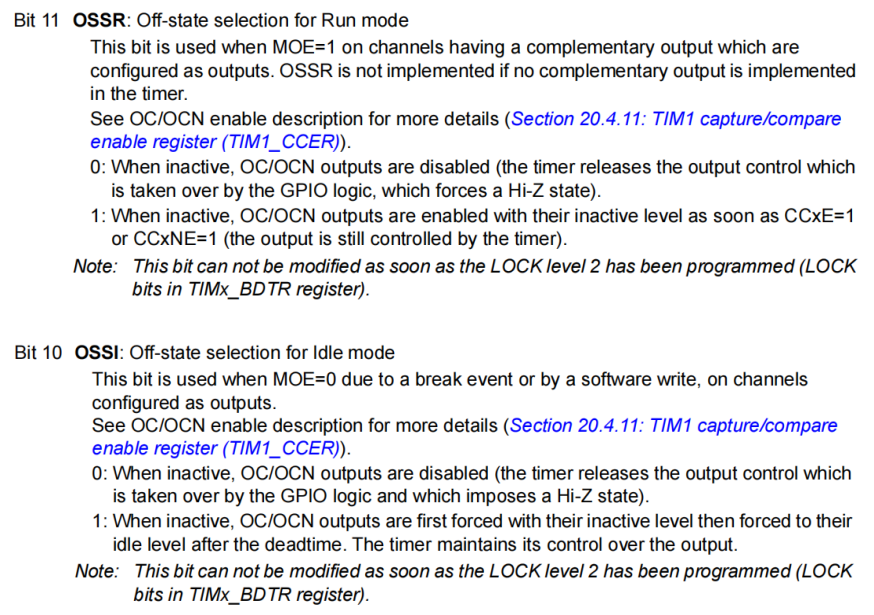
死区时间(TIMx\_DBTR = 0x130AD48)









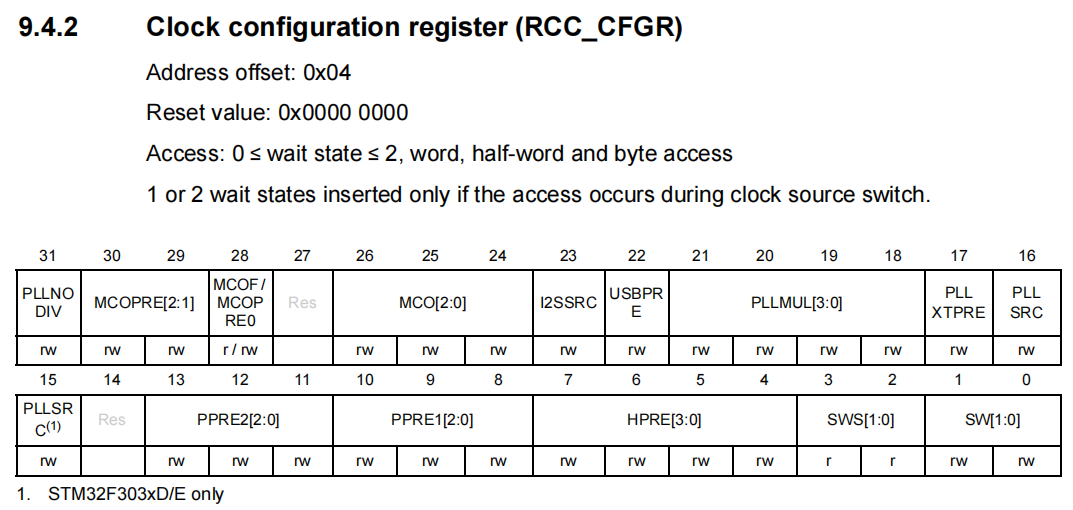


死区时间计算：

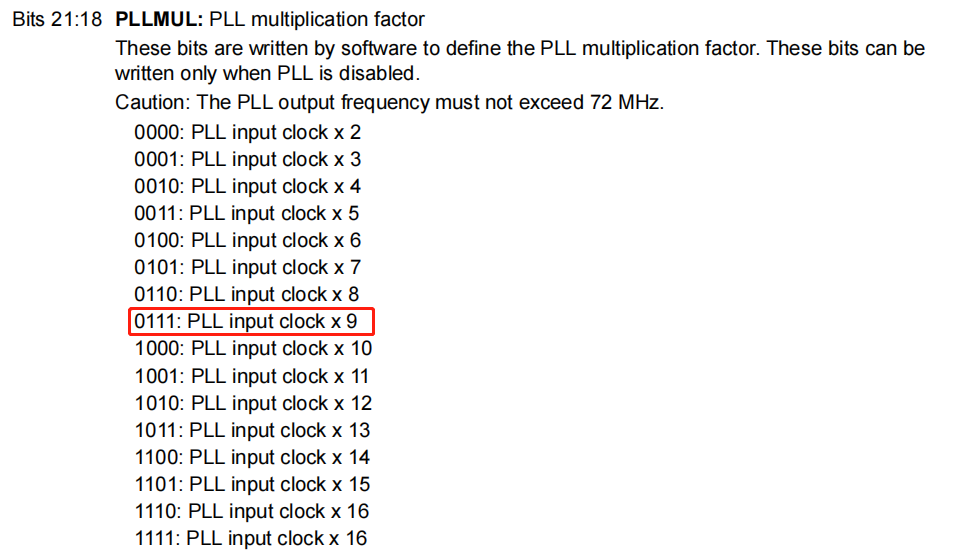
tDTS = 1/Fclk = 1/144000000 = 6.944444 ns

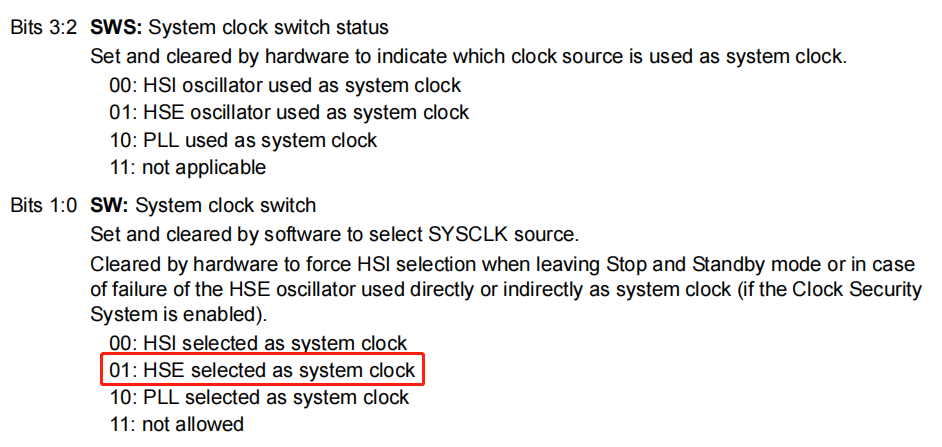
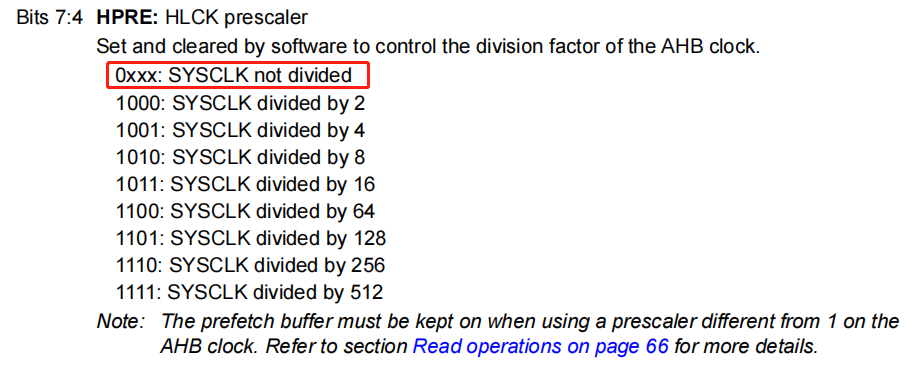
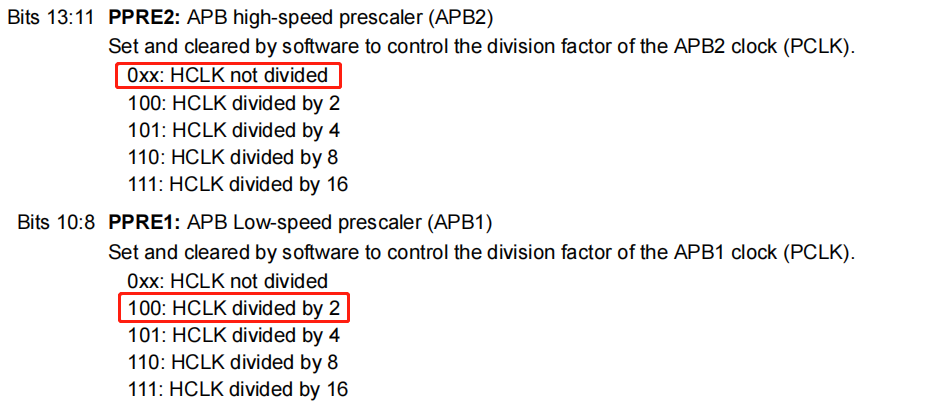
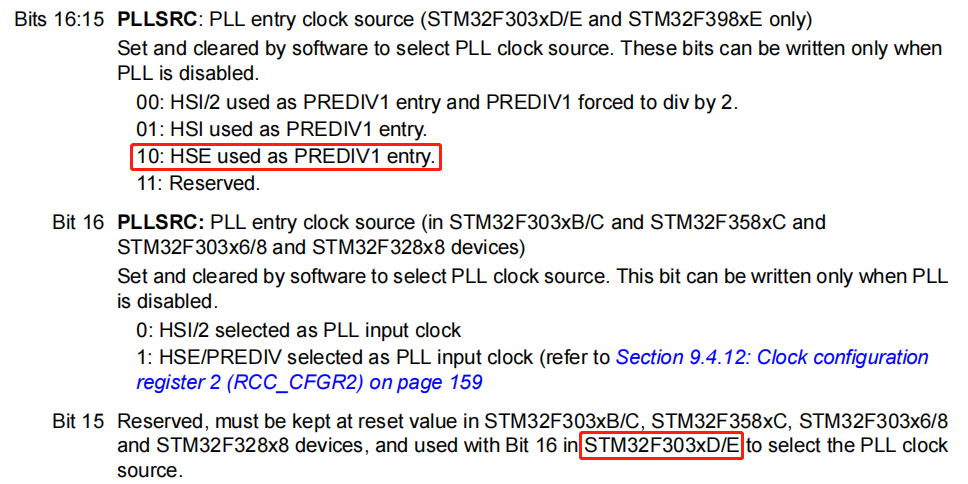
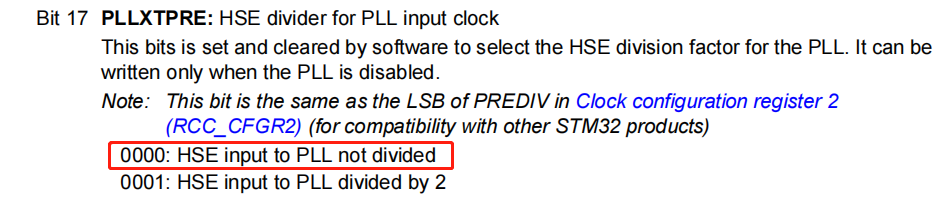
设置的死区时间为1000ns ，1000/6.944444 =144

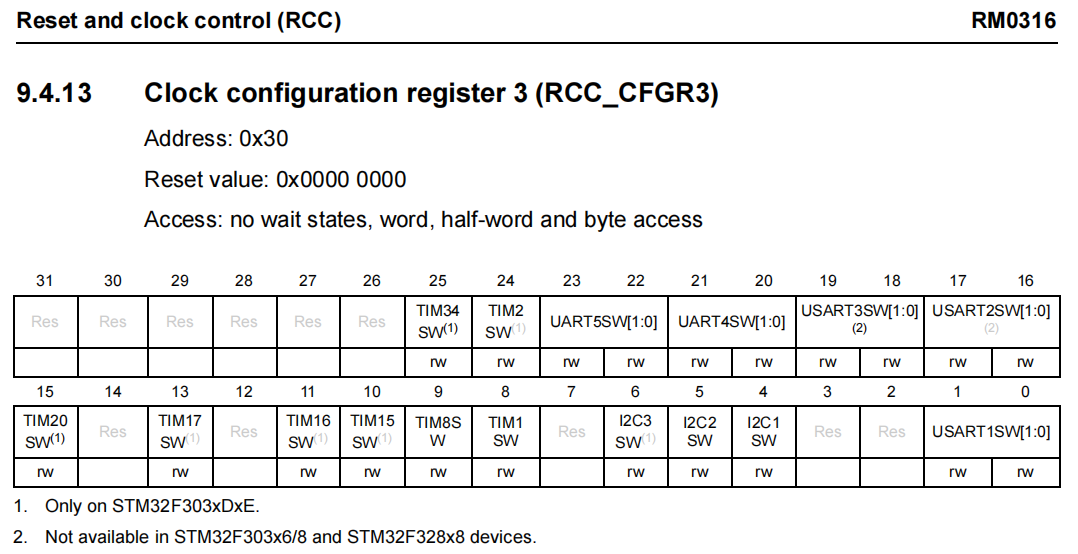
然而，CHx和CHxN 各自会插入死区时间，所以DTG[7:0]设置值只要计算值的一半，即144/2 = 72 = 0x48



设置值：0x1D040A (0001 1101 0000 0100 0000 1010)







设置值：0x100 (0001 0000 0000)

