

32 位 MCU
ES32F0541

参 考 手 册

- 产品简介
- 数据手册
- 参考手册

上海东软载波微电子有限公司

2020-07-16

目录

目录	2
前言	19
相关文件	20
文件约定	21
第 1 章 系统和内存概述	22
1.1 概述	22
1.2 结构图	24
1.3 功能描述	25
1.4 内存映射	28
第 2 章 引脚和信号	31
2.1 引脚	31
2.1.1 ES32F0541 LQFP – 64 引脚	31
2.1.2 ES32F0541 LQFP – 48 引脚	32
2.1.3 ES32F0541 LQFP – 32 引脚	33
2.1.4 ES32F0541 QFN – 32 引脚	34
2.2 ES32F0541 引脚定义	35
2.3 多路复用引脚功能选择	40
2.4 信号描述	45
2.5 电源引使用说明	48
2.5.1 ADC/DAC 参考电源引脚 VREF	48
第 3 章 ARM® Cortex™-M0 Core	50
3.1 概述	50
3.2 特性	50
3.3 功能描述	51
3.3.1 CPU 系统定时器控制寄存器(SYST)	51
3.3.2 嵌套向量中断控制器(NVIC)	52
3.3.3 CPU 系统控制	54
3.4 特殊功能寄存器	55
3.4.1 寄存器列表	55
3.4.2 寄存器描述	56
第 4 章 系统配置控制器 (SYSCFG)	69
4.1 概述	69
4.2 特性	69
4.3 功能描述	70
4.3.1 电源	70
4.3.2 低功耗模式 (Low Power Mode)	72
4.3.3 系统重映射	76
4.3.4 ADC/DAC 电源配置	77
4.3.5 温度传感器	77
4.3.6 NRST 复位请求	77
4.3.7 停止计数配置	77
4.3.8 红外线接口输出配置	78

4.4	特殊功能寄存器	79
4.4.1	寄存器列表	79
4.4.2	寄存器描述	80
第5章	复位和时钟控制 (RCU).....	88
5.1	概述	88
5.2	特性	88
5.3	功能描述	88
5.3.1	时钟	88
5.3.2	复位	93
5.3.3	微控制器时钟输出	94
5.3.4	睡眠模式中的外设时钟使能	94
5.3.5	时钟安全系统	95
5.4	特殊功能寄存器	96
5.4.1	寄存器列表	96
5.4.2	寄存器描述	97
第6章	闪存控制器 (FLASH).....	117
6.1	概述	117
6.2	特性	117
6.3	结构图	118
6.4	功能描述	119
6.4.1	用户配置字	120
6.4.2	闪存保护	122
6.4.3	预取	125
6.4.4	闪存重映射	125
6.4.5	闪存编程/擦除	126
6.4.6	闪存全擦除	126
6.4.7	更新保护	127
6.4.8	识别码	127
6.5	特殊功能寄存器	128
6.5.1	寄存器列表	128
6.5.2	寄存器描述	129
第7章	通用 I/Os (GPIO).....	141
7.1	概述	141
7.2	特性	141
7.3	结构图	141
7.4	功能描述	142
7.4.1	通用 I/O (GPIO)	143
7.4.2	I/O 端口控制寄存器	143
7.4.3	I/O 端口数据寄存器	143
7.4.4	I/O 数据位操作	143
7.4.5	GPIO 锁定机制	143
7.5	特殊功能寄存器	144
7.5.1	寄存器列表	144
7.5.2	寄存器描述	145

第 8 章	直接存储器访问控制器 (DMA)	157
8.1	概述	157
8.2	特性	157
8.3	结构图.....	158
8.4	功能描述	158
8.4.1	直接存储器访问控制器事务	158
8.4.2	外设通道选择.....	159
8.4.3	仲裁器.....	160
8.4.4	来源, 目的和转移模式	160
8.4.5	地址递增	160
8.4.6	循环模式	161
8.4.7	可编程数据宽度, 打包/解包, 字节顺序.....	161
8.4.8	单次和突发传输	163
8.4.9	直接模式	163
8.4.10	DMA 传输完成	164
8.4.11	流量控制器.....	165
8.4.12	支持配置的总结	165
8.4.13	信道配置程序.....	166
8.4.14	DMA 通道错误讯息	167
8.5	特殊功能寄存器	168
8.5.1	寄存器列表	168
8.5.2	寄存器描述	169
第 9 章	硬件除法器 (HDIV)	212
9.1	概述	212
9.2	功能描述	212
9.3	特殊功能寄存器	213
9.3.1	寄存器列表	213
9.3.2	寄存器描述	214
第 10 章	通用异步收发器 (UART)	217
10.1	概述	217
10.2	特性	217
10.3	结构图.....	219
10.4	功能描述	220
10.4.1	具体功能配置.....	221
10.4.2	功能描述	222
10.4.3	发送器.....	223
10.4.4	接收器.....	225
10.4.5	状态寄存器	230
10.4.6	波特率产生器.....	232
10.4.7	自动波特率侦测	234
10.4.8	自动流量控制.....	237
10.4.9	Modbus 通讯	239
10.4.10	校验控制	240
10.4.11	多处理器通讯.....	241

10.4.12	LIN 模式.....	242
10.4.13	单线半双工通讯.....	244
10.4.14	智能卡模式.....	245
10.4.15	IrDA SIR 模块.....	247
10.4.16	使用 DMA 连续通讯.....	249
10.4.17	中断配置.....	251
10.5	特殊功能寄存器.....	252
10.5.1	寄存器列表.....	252
10.5.2	寄存器描述.....	253
第 11 章	外部中断 (EXTI).....	285
11.1	概述.....	285
11.2	特性.....	285
11.3	结构图.....	285
11.4	功能描述.....	285
11.4.1	硬件中断选择.....	285
11.4.2	软件事件选择.....	286
11.4.3	外部和内部中断 / 事件线路映射.....	287
11.5	特殊功能寄存器.....	288
11.5.1	寄存器列表.....	288
11.5.2	寄存器描述.....	289
第 12 章	模数转换器 (ADC).....	302
12.1	概述.....	302
12.2	特性.....	302
12.3	结构图.....	303
12.4	功能描述.....	304
12.4.1	采样序列发生器.....	304
12.4.2	取样时钟.....	304
12.4.3	采样优先次序.....	304
12.4.4	采样事件/触发器.....	305
12.4.5	输入讯号放大.....	305
12.4.6	中断控制.....	305
12.4.7	DMA 请求.....	305
12.4.8	初始化及配置.....	306
12.5	特殊功能寄存器.....	307
12.5.1	寄存器列表.....	307
12.5.2	寄存器描述.....	309
第 13 章	模拟比较器 (CMP).....	343
13.1	概述.....	343
13.2	特性.....	343
13.3	结构图.....	343
13.4	功能描述.....	344
13.5	特殊功能寄存器.....	345
13.5.1	寄存器列表.....	345
13.5.2	寄存器描述.....	346

第 14 章	数模转换器 (DAC)	352
14.1	概述	352
14.2	特性	352
14.3	结构图.....	352
14.4	功能描述	353
14.4.1	输出电压	353
14.4.2	触发选择	353
14.4.3	数据转换	353
14.4.4	波形生成	354
14.4.5	数据输出反转	355
14.4.6	DMA 请求.....	355
14.5	特殊功能寄存器	356
14.5.1	寄存器列表	356
14.5.2	寄存器描述	357
第 15 章	通用定时器 16 位 2 通道 (GP16C2T)	361
15.1	概述	361
15.2	特性	361
15.3	结构图.....	362
15.4	功能描述	363
15.4.1	预分频器	363
15.4.2	重复计数器	364
15.4.3	时钟源.....	364
15.4.4	计数模式	366
15.4.5	捕获/比较通道.....	368
15.4.6	输入捕获模式.....	370
15.4.7	PWM 模式	372
15.4.8	输出比较模式.....	373
15.4.9	单脉冲模式	374
15.4.10	互补输出与死区时间.....	375
15.4.11	刹车功能	376
15.4.12	外部触发的同步	378
15.4.13	定时器同步	381
15.4.14	调试模式	384
15.5	特殊功能寄存器	385
15.5.1	寄存器列表	385
15.5.2	寄存器描述	386
第 16 章	通用定时器 16 位 4 通道 (GP16C4T)	417
16.1	概述	417
16.2	特性	417
16.3	结构图.....	418
16.4	功能描述	419
16.4.1	预分频器	419
16.4.2	时钟源.....	420
16.4.3	计数模式	424

16.4.4	捕获/比较通道.....	428
16.4.5	输入捕获模式.....	429
16.4.6	PWM 模式.....	430
16.4.7	输出比较模式.....	434
16.4.8	单脉冲模式.....	436
16.4.9	编码器接口模式.....	437
16.4.10	输入异或功能.....	439
16.4.11	霍尔传感器接口.....	439
16.4.12	外部触发的同步.....	441
16.4.13	定时器同步.....	445
16.4.14	调试模式.....	448
16.5	特殊功能寄存器.....	450
16.5.1	寄存器列表.....	450
16.5.2	寄存器描述.....	451
第 17 章	通用定时器 32 位 4 通道 (GP32C4T).....	487
17.1	概述.....	487
17.2	特性.....	487
17.3	结构图.....	488
17.4	功能描述.....	489
17.4.1	预分频器.....	489
17.4.2	时钟源.....	490
17.4.3	计数模式.....	494
17.4.4	捕获/比较通道.....	498
17.4.5	输入捕获模式.....	499
17.4.6	PWM 模式.....	501
17.4.7	输出比较模式.....	504
17.4.8	单脉冲模式.....	506
17.4.9	编码器接口模式.....	507
17.4.10	输入异或功能.....	509
17.4.11	霍尔传感器接口.....	509
17.4.12	外部触发的同步.....	511
17.4.13	定时器同步.....	515
17.4.14	调试模式.....	519
17.5	特殊功能寄存器.....	520
17.5.1	寄存器列表.....	520
17.5.2	寄存器描述.....	521
第 18 章	高级控制定时器 (AD16C4T).....	557
18.1	概述.....	557
18.2	特性.....	557
18.3	结构图.....	558
18.4	功能描述.....	559
18.4.1	预分频器.....	559
18.4.2	重复计数器.....	560
18.4.3	时钟源.....	561

18.4.4	计数模式	565
18.4.5	捕获/比较通道	569
18.4.6	输入捕获模式	571
18.4.7	PWM 模式	573
18.4.8	输出比较模式	576
18.4.9	单脉冲模式	578
18.4.10	互补输出与死区时间	579
18.4.11	刹车功能	580
18.4.12	编码器接口模式	582
18.4.13	输入异或功能	584
18.4.14	霍尔传感器接口	584
18.4.15	外部触发的同步	586
18.4.16	定时器同步	590
18.4.17	调试模式	594
18.5	特殊功能寄存器	595
18.5.1	寄存器列表	595
18.5.2	寄存器描述	596
第 19 章	基本定时器 (BS16T).....	638
19.1	概述	638
19.2	特性	638
19.3	结构图	638
19.4	功能描述	639
19.4.1	预分频器	639
19.4.2	时钟源	640
19.4.3	递增计数模式	640
19.4.4	调试模式	642
19.5	特殊功能寄存器	643
19.5.1	寄存器列表	643
19.5.2	寄存器描述	644
第 20 章	独立看门狗 (IWDG).....	653
20.1	概述	653
20.2	特性	653
20.3	结构图	653
20.4	功能描述	654
20.4.1	窗口选项	654
20.4.2	Sleep 模式下的行为	655
20.4.3	调试模式	655
20.5	特殊功能寄存器	656
20.5.1	寄存器列表	656
20.5.2	寄存器描述	657
第 21 章	窗口看门狗 (WWDG).....	662
21.1	概述	662
21.2	特性	662
21.3	结构图	662

21.4	功能描述	663
21.4.1	启用看门狗	663
21.4.2	控制递减计数器	663
21.4.3	高级看门狗中断功能	663
21.4.4	如何编程看门狗超时	664
21.4.5	调试模式	664
21.5	特殊功能寄存器	665
21.5.1	寄存器列表	665
21.5.2	寄存器描述	666
第 22 章	实时时钟 (RTC)	671
22.1	概述	671
22.2	特性	671
22.3	结构图	672
22.4	功能描述	672
22.4.1	设定并开启 RTC	672
22.4.2	读取 RTC 日期与时间	672
22.4.3	RTC 校正	673
22.4.4	RTC 睡眠计数器	674
22.4.5	RTC 闹铃	674
22.4.6	RTC 触发 ADC 或 DAC	674
22.5	特殊功能寄存器	675
22.5.1	寄存器列表	675
22.5.2	寄存器描述	676
第 23 章	串行总线 (I2C)	696
23.1	概述	696
23.2	特性	696
23.3	结构图	697
23.4	功能描述	698
23.4.1	I2C 总线协议	698
23.4.2	I2C 时钟要求	702
23.4.3	数据传输	703
23.4.4	I2C 从机模式	705
23.4.5	I2C 主机模式	709
23.4.6	I2C_TIMINGR 寄存器的配置的例子	713
23.4.7	SMBus 具体功能	715
23.4.8	SMBus 初始化	717
23.4.9	SMBus: I2C_TIMEOUTR 寄存器配置的例子	719
23.4.10	DMA 请求	720
23.4.11	错误情况	720
23.4.12	I2C 中断	721
23.5	特殊功能寄存器	722
23.5.1	寄存器列表	722
23.5.2	寄存器描述	723
第 24 章	串行外设接口 (SPI)	755

24.1	概述	755
24.2	特性	755
24.3	SPI 结构图	757
24.4	SPI 功能描述	758
24.4.1	通信格式	758
24.4.2	从机选择 (NSS) 引脚管理	759
24.4.3	单对单应用	759
24.4.4	数据发送和接收	760
24.4.5	DMA 请求	767
24.4.6	SPI 状态标志	768
24.4.7	SPI 错误标志	770
24.4.8	SPI TI 模式	771
24.4.9	CRC 计算	771
24.5	I2S 结构图	773
24.6	I2S 功能描述	774
24.6.1	音频协议	774
24.6.2	时钟产生器	781
24.6.3	I2S 主机模式	782
24.6.4	I2S 从机模式	783
24.6.5	I2S 状态标志	785
24.6.6	I2S 错误标志	786
24.7	特殊功能寄存器	787
24.7.1	寄存器列表	787
24.7.2	寄存器描述	788
第 25 章	无刷马达矢量控制器 (MCM)	813
25.1	概述	813
25.2	特性	813
25.3	结构图	814
25.4	功能描述	815
25.4.1	斜坡函数产生器	815
25.4.2	Clarke/ Park 逆变换	816
25.4.3	相电流 Clarke/Park 变换	818
25.4.4	比例积分控制器 (PI)	819
25.4.5	ADC 检测电流低通滤波器 (LPF)	822
25.4.6	滑模观测理论 (Sliding Mode Observer Theory)	824
25.4.7	虚拟电机模型	825
25.4.8	滑模控制器与动态 LPF	828
25.4.9	CORDIC-反正切角度估算	830
25.5	特殊功能寄存器	831
25.5.1	寄存器列表	831
25.5.2	寄存器描述	834
版本历史		916

图目录

图 1-1 ES32F0541 系统框图	24
图 1-2 内存映射	28
图 2-1 ES32F0541 64-Pin LQFP 封装引脚图	31
图 2-2 ES32F0541 48-Pin LQFP 封装引脚图	32
图 2-3 ES32F0541 32-Pin LQFP 封装引脚图	33
图 2-4 ES32F0541 32-Pin QFN 封装引脚图	34
图 2-5 VREF 的内部电源示意图	48
图 2-6 VREF 与 VDD5 电源对接示意图	48
图 2-7 VREF 接上外部参考电源示意图	49
图 2-8 VREF 接稳压电容示意图	49
图 4-1 电源域	70
图 4-2 POR/PDR 复位	71
图 4-3 PVD 旗标	72
图 5-1 时钟树	90
图 5-2 NRST 复位	93
图 5-3 微控制器时钟输出	94
图 6-1 闪存控制器架构	118
图 6-2 不连续区间保护示意图	123
图 6-3 闪存读取位置对照	125
图 7-1 I/O 端口位的基本结构图	141
图 8-1 直接存储器访问控制器框图	158
图 9-1 硬件除法器运作流程图	212
图 10-1 UART 框图	219
图 10-2 字长编程	222
图 10-3 配置停止位	223
图 10-4 防抖动波形	225
图 10-5 防抖动输出	225
图 10-6 起始位侦测	226
图 10-7 数据采样	227
图 10-8 自动波特率侦测模式 0	235
图 10-9 自动波特率侦测模式 1	235
图 10-10 自动波特率侦测模式 2	236
图 10-11 自动流量控制框图	237
图 10-12 自动 RTSn 控制	237
图 10-13 自动 CTSn 控制	238
图 10-14 驱动使能当 AADINV=0	238
图 10-15 使用地址标示侦测模式	241
图 10-16 LIN 模式下断开信号侦测（11 位断开长度 – LBDL 位为 1）	243
图 10-17 LIN 模式下的断开检测与帧错误的检测	244
图 10-18 ISO 7816-3 异步协定	245
图 10-19 用 1.5 位停止位时检测校验错误	246
图 10-20 红外收发框图	247

图 10-21 IrDA 数据调制 (3/16) –正常模式	248
图 11-1 外部中断 / 事件框图	285
图 11-2 外部中断 / 事件 GPIO 映射.....	287
图 12-1 ADC 模块图	303
图 13-1 比较器结构图.....	343
图 13-2 比较器框图.....	344
图 14-1 DAC 结构框图.....	352
图 14-2 数据传输的时序图, 当 DAC_CTRL.TRIEN = 0	353
图 14-3 数据传输的时序图, 当 DAC_CTRL.TRIEN = 1	353
图 14-4 DAC 产生锯齿波 (向上计数模式).....	354
图 14-5 DAC 产生锯齿波 (向下计数模式).....	354
图 14-6 DAC 产生三角波 (上/下计数).....	355
图 14-7 DAC 具有波形生成的 DAC 转换 (触发器启用)	355
图 15-1 GP16C2Tn 结构框图	362
图 15-2 预分频值计数时序图	363
图 15-3 重复计数器工作模式	364
图 15-4 采用内部时钟计数.....	364
图 15-5 外部时钟连接	365
图 15-6 ITn 外部时钟连接.....	366
图 15-7 计数器递增计数时序图	367
图 15-8 当 ARPEN=0 时计数器时序图	367
图 15-9 当 ARPEN=1 时计数器时序图	368
图 15-10 捕获/比较通道.....	368
图 15-11 捕获/比较通道 1 结构图	369
图 15-12 捕获/比较信道的输出部分.....	369
图 15-13 PWM 输入模式时序	371
图 15-14 边沿对齐递增计数 PWM 波形 (AR=8)	372
图 15-15 输出比较模式, 触发 CHn.....	373
图 15-16 单脉冲模式.....	375
图 15-17 互补输出含死区时间插入.....	376
图 15-18 刹车输出行为.....	377
图 15-19 复位模式控制电路.....	378
图 15-20 门控模式控制电路.....	379
图 15-21 触发模式控制电路.....	380
图 15-22 主/从定时器范例	381
图 15-23 门控从定时器使用主定时器 CH1REF.....	381
图 15-24 通过使能定时器 1 可以控制定时器 2	382
图 15-25 触发中从定时器使用主定时器更新事件.....	383
图 15-26 使用定时器 1 的 I1 输入触发定时器 1 和定时器 2	384
图 16-1 GP16C4Tn 结构框图	418
图 16-2 预分频值计数时序图	419
图 16-3 采用内部时钟计数.....	420
图 16-4 外部时钟连接	421
图 16-5 外部触发输入模块.....	422

图 16-6 ITn 外部时钟连接.....	423
图 16-7 计数器递增计数时序图	424
图 16-8 当 ARPEN=0 时计数器时序图	425
图 16-9 当 ARPEN=1 时计数器时序图.....	425
图 16-10 定时器递减计数时序图	426
图 16-11 增减计数器时序图.....	427
图 16-12 捕获/比较通道.....	428
图 16-13 捕获/比较通道 1 结构图.....	428
图 16-14 捕获/比较通道 1 结构图.....	429
图 16-15 PWM 输入模式时序.....	430
图 16-16 边沿对齐递增计数 PWM 波形 (AR=8)	431
图 16-17 边沿对齐递减计数 PWM 波形 (AR=8)	432
图 16-18 中央对齐 PWM 波形 (AR=0x3F)	432
图 16-19 输出比较模式, 触发 CHn.....	434
图 16-20 清除比较输出 CHn.....	435
图 16-21 单脉冲模式.....	436
图 16-22 编码器接口模式下的计数操作	438
图 16-23 滤波后极性反相时编码器接口例子	438
图 16-24 霍尔传感器接口范例	440
图 16-25 复位模式控制电路.....	441
图 16-26 门控模式控制电路.....	442
图 16-27 触发模式控制电路.....	443
图 16-28 外部时钟源 2+触发模式下的控制电路.....	444
图 16-29 主/从定时器范例	445
图 16-30 门控从定时器使用主定时器 CH1REF.....	445
图 16-31 通过使能定时器 1 可以控制定时器 2.....	446
图 16-32 触发中从定时器使用主定时器更新事件.....	447
图 16-33 使用定时器 1 的 I1 输入触发定时器 1 和定时器 2	448
图 17-1 GP32C4Tn 结构框图	488
图 17-2 预分频值计数时序图	489
图 17-3 采用内部时钟计数.....	490
图 17-4 外部时钟连接.....	491
图 17-5 外部触发输入模块.....	492
图 17-6 ITn 外部时钟连接.....	493
图 17-7 计数器递增计数时序图	494
图 17-8 当 ARPEN=0 时计数器时序图	495
图 17-9 当 ARPEN=1 时计数器时序图	495
图 17-10 定时器递减计数时序图	496
图 17-11 增减计数器时序图.....	497
图 17-12 捕获/比较通道.....	498
图 17-13 捕获/比较通道 1 结构图.....	498
图 17-14 捕获/比较通道 1 结构图.....	498
图 17-15 PWM 输入模式时序.....	500
图 17-16 边沿对齐递增计数 PWM 波形 (AR=8)	501

图 17-17 边沿对齐递减计数 PWM 波形 (AR=8)	502
图 17-18 中央对齐 PWM 波形 (AR=0x3F)	503
图 17-19 输出比较模式, 触发 CHn.....	504
图 17-20 清除比较输出 CHn.....	505
图 17-21 单脉冲模式.....	506
图 17-22 编码器接口模式下的计数操作	508
图 17-23 滤波后极性反相时编码器接口例子	508
图 17-24 霍尔传感器接口范例	510
图 17-25 复位模式控制电路.....	511
图 17-26 门控模式控制电路.....	512
图 17-27 触发模式控制电路.....	513
图 17-28 外部时钟源 2+触发模式下的控制电路.....	514
图 17-29 主/从定时器范例	515
图 17-30 门控从定时器使用主定时器 CH1REF.....	515
图 17-31 通过使能定时器 1 可以控制定时器 2.....	516
图 17-32 触发中从定时器使用主定时器更新事件.....	517
图 17-33 使用定时器 1 的 I1 输入触发定时器 1 和定时器 2	518
图 18-1 高级控制定时器结构框图.....	558
图 18-2 预分频值计数时序图	559
图 18-3 重复计数器工作模式	560
图 18-4 采用内部时钟计数.....	561
图 18-5 外部时钟连接	562
图 18-6 外部触发输入模块.....	563
图 18-7 ITn 外部时钟连接.....	564
图 18-8 计数器递增计数时序图	565
图 18-9 当 ARPEN=0 时计数器时序图	566
图 18-10 当 ARPEN=1 时计数器时序图	566
图 18-11 定时器递减计数时序图.....	567
图 18-12 增减计数器时序图.....	568
图 18-13 捕获/比较通道	569
图 18-14 捕获/比较通道 1 结构图	569
图 18-15 捕获/比较信道的输出部分.....	570
图 18-16 PWM 输入模式时序	572
图 18-17 边沿对齐递增计数 PWM 波形 (AR=8)	573
图 18-18 边沿对齐递减计数 PWM 波形 (AR=8)	574
图 18-19 中央对齐 PWM 波形 (AR=0x3F)	575
图 18-20 输出比较模式, 触发 CHn.....	576
图 18-21 清除比较输出 CHn.....	577
图 18-22 单脉冲模式.....	578
图 18-23 互补输出含死区时间插入.....	579
图 18-24 刹车输出行为.....	581
图 18-25 编码器接口模式下的计数操作	583
图 18-26 滤波后极性反相时编码器接口例子	583
图 18-27 霍尔传感器接口范例	585

图 18-28 复位模式控制电路.....	586
图 18-29 门控模式控制电路.....	587
图 18-30 触发模式控制电路.....	588
图 18-31 外部时钟源 2+触发模式下的控制电路.....	589
图 18-32 主/从定时器范例	590
图 18-33 门控从定时器使用主定时器 CH1REF.....	591
图 18-34 通过使能定时器 1 可以控制定时器 2.....	592
图 18-35 触发中从定时器使用主定时器更新事件.....	593
图 18-36 使用定时器 1 的 I1 输入触发定时器 1 和定时器 2	594
图 19-1 基本定时器结构框图	638
图 19-2 预分频值计数时序图	639
图 19-3 采用内部时钟计数.....	640
图 19-4 计数器递增计数时序图	641
图 19-5 当 ARPEN=0 时计数器时序图.....	641
图 19-6 当 ARPEN=1 时计数器时序图.....	642
图 20-1 独立看门狗框图	653
图 21-1 窗口看门狗框图	662
图 21-2 WWDT 中断示意图.....	663
图 21-3 窗口看门狗时序图.....	664
图 23-1 I2C 结构图	697
图 23-2 START 和 STOP 条件.....	698
图 23-3 I2C 总线上的应答	699
图 23-4 7 位地址格式.....	700
图 23-5 10 位地址格式.....	700
图 23-6 主机 - 发送协议	701
图 23-7 主机 - 接收协议	701
图 23-8 I2C 总线上的数据传输	703
图 23-9 从机初始化流程图.....	706
图 23-10 从机发送的传输序列图	707
图 23-11 从机接收的传输序列图.....	708
图 23-12 主时钟产生.....	709
图 23-13 SCL 主时钟同步	710
图 23-14 主机发送的传输序列图	711
图 23-15 主机接收的传输序列图	712
图 24-1 SPI 电路结构框图	757
图 24-2 SPI 格式.....	758
图 24-3 全双工通信.....	759
图 24-4 全双工通信(SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)的 TXE、RXE、BUSY 行为(直接存取操作模式在连续传输的情况下).....	762
图 24-5 全双工通信(SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)的 TXTH、RXTH、TXFLV、RXFLV、BUSY 行为(FIFO 缓存操作模式在连续传输的情况下).....	763
图 24-6 单工通信-只发送模式(SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)的 TXE、BUSY 行为(直接存取操作模式在连续传输的情况下).....	764
图 24-7 单工通信-只发送模式(SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)的 TXTH、TXFLV、	

BUSY 行为(FIFO 缓存操作模式在连续传输的情况下).....	765
图 24-8 单工通信-只接收模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1) 的 RXE 行为(直接存取操作模式在连续传输的情况下).....	766
图 24-9 单工通信-只接收模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1) 的 RXTH、RXFLV 行为(FIFO 缓存操作模式在连续传输的情况下).....	766
图 24-10 发送时 (SPI_CON1.BIDEN = 0 且 SPI_CON1.RXO=0) 的 TXE/BUSY 行为 (在间断传输的情况下)	767
图 24-11 使用 DMA 进行发送.....	768
图 24-12 使用 DMA 进行接收	768
图 24-13 TI 格式	771
图 24-14 I2S 框图.....	773
图 24-15 I2S 飞利浦标准波形 (16/32 位数据帧, CPOL = 0)	775
图 24-16 I2S 飞利浦标准波形 (24 位数据帧, CPOL = 0)	775
图 24-17 发送 0x123456.....	775
图 24-18 接收 0x123456.....	776
图 24-19 I2S 飞利浦标准波形 (16 位数据帧扩展到 32 位通道帧, CPOL = 0)	776
图 24-20 16 位数据帧扩展到 32 位通道帧的示例.....	776
图 24-21 MSB 对齐标准波形 (16/32 位数据帧, CPOL = 0)	777
图 24-22 MSB 对齐标准波形 (24 位数据帧, CPOL = 0)	777
图 24-23 MSB 对齐标准波形 (16 位数据帧扩展到 32 位通道帧, CPOL = 0)	777
图 24-24 LSB 对齐标准波形 (16/32 位数据帧, CPOL = 0)	778
图 24-25 LSB 对齐标准波形 (24 位数据帧, CPOL = 0)	778
图 24-26 传输 0x123456 所需的操作	778
图 24-27 接收 0x123456 所需的操作	779
图 24-28 LSB 对齐标准波形 (16 位数据帧扩展到 32 位通道帧, CPOL = 0)	779
图 24-29 16 位数据帧扩展到 32 位通道帧的示例.....	779
图 24-30 PCM 标准波形 (16 位)	780
图 24-31 PCM 标准波形 (16 位数据帧扩展到 32 位通道帧)	780
图 24-32 音频采样频率定义.....	781
图 25-1 FOC 系统方块图.....	814
图 25-2 MCM 斜波函数产生器.....	815
图 25-3 Clarke/Park 逆变换与 SVPWM 用以产生三相占空比.....	816
图 25-4 MCM Clarke/Park 逆变换与 SVM 转换模块	817
图 25-5 Clarke/ Park 变换.....	818
图 25-6 MCM Clarke/ Park 变换	818
图 25-7 PI 控制器方块图.....	819
图 25-8 PI 控制器的两种控制配置	820
图 25-9 MCM PI 控制器.....	821
图 25-10 一阶 IIR 数字低通滤波器.....	822
图 25-11 MCM 电流感测低通滤波器.....	823
图 25-12 滑模观测器.....	824
图 25-13 虚拟电机模型	826
图 25-14 SMC 流程图.....	828
图 25-15 MCM 滑模控制器	828

图 25-16 CORDIC Atan 方块图	830
图 25-17 MCM CORDIC.....	830

表目录

表 1-1 ES32F0541 设备摘要	23
表 1-2 ES32F0541 微控制器特性	27
表 1-3 外设寄存器边界地址	30
表 2-1 引脚定义	39
表 2-2 复用功能端口 A	41
表 2-3 复用功能端口 B	42
表 2-4 复用功能端口 C	43
表 2-5 复用功能端口 D	44
表 2-6 复用功能描述	47
表 4-1 低功耗模式	72
表 4-2 红外线控制讯号组合	78
表 5-1 外设分配表	92
表 6-1 闪存信息区规划	119
表 6-2 讯息区页 6 保护区设定	122
表 6-3 闪存保护区权限	124
表 8-1 DMA 请求对应表	159
表 8-2 转移模式	160
表 8-3 打包/解包和端序行为 (位 SINC = DINC = 1)	162
表 8-4 对 BCR 与 SDWSEL 和 DDWSEL 的限制	162
表 8-5 DMA 配置组合	165
表 10-1 UART/SUART 具体功能配置	221
表 10-2 采样后的噪声侦测值	229
表 10-3 时钟为 48MHz 下, 设置波特率时的误差计算	233
表 10-4 帧格式	240
表 10-5 中断配置表	251
表 11-1 EXTI 线路连接	287
表 12-1 序列发生器的采样和 FIFO 深度	304
表 15-1 GP16C2Tn 内部触发连接	391
表 16-1 计数方向与编码器信号的关系	437
表 16-2 GP16C4Tn 内部触发连接	458
表 17-1 计数方向与编码器信号的关系	507
表 17-2 GP32C4Tn 内部触发连接	528
表 18-1 计数方向与编码器信号的关系	582
表 18-2 AD16C4Tn 内部触发连接	603
表 23-1 第一个字节中位的定义	700
表 23-2 $F_{I2CCLK} = 8 \text{ MHz}$ 的时序设置示例	713
表 23-3 $F_{I2CCLK} = 16 \text{ MHz}$ 的时序设置示例	713
表 23-4 $F_{I2CCLK} = 48 \text{ MHz}$ 的时序设置示例	714
表 23-5 SMBus 超时规格	716
表 23-6 各种 I2CCLK 频率的 TIMEOUTA 设置示例 (最大值 $T_{TIMEOUT} = 25 \text{ ms}$)	719
表 23-7 各种 I2CCLK 频率的 TIMEOUTB 设置示例	719
表 23-8 各种 I2CCLK 频率的 TIMEOUTA 设置示例 (最大 $T_{IDLE} = 50 \mu\text{s}$)	719

前言

该数据表旨在为系统软件开发人员、硬件设计者和应用程序开发人员提供数据。它提供了完整的ES32F0541 微控制器的信息，包括如何使用设备、系统和总线结构、存储器组织及其外设指令。

客户通知

请注意，随着EASTSOFT工具和文档的不断发展以满足市场的需要，因此一些实际的对话框和工具描述可能与本文档中的不同。有关此产品的最新信息，请参阅东软在线支持 (www.essemi.com)

相关文件

- Cortex®-M0 技术参考手册, 可参考: <http://infocenter.arm.com>

文件约定

下表解释了文档中经常使用的缩写。

缩写词	说明	描述
R/W	读/写(_IO)	软件可以读写这些位
R	只读(_I)	软件只能读取这些位
W	只写(_O)	软件只能写入该位，读取该位时将返回复位值
W1	只写(写1)(_O)	软件只能写入该位，写1有效，写0无作用。
R/C_W1	读取/清零(写1)(_IO)	软件可以读取该位，也可以通过写入1将该位清零。写入“0”对该位的值无影响
R/C_W0	读取/清零(写0)(_IO)	软件可以读取该位，也可以通过写入0将该位清零。写入“1”对该位的值无影响
R/C_R	读取/清零(读取)(_IO)	软件可以读取该位。读取该位时，将自动清零。写入“0”对该位的值无影响
C_W1	清零(写1)(_O)	通过写入1将该位清零。写入“0”对该位的值无影响
S_W1	置位(写1)(_O)	通过写入1将该位置位。写入“0”对该位的值无影响
C_W0	清零(写0)(_O)	通过写入0将该位清零。写入“1”对该位的值无影响
T_W1	触发(写1)(_O)	通过写入1将触发硬件动作。写入“0”对该位的值无影响
Reserved	保留(无)	保留位，必须保持复位值。

第1章 系统和内存概述

1.1 概述

ES32F0541 微控制器是一系列低功耗微控制器，集成了高性能 ARM Cortex™-M0 32 位 RISC 内核。它的最高工作频率为 48 MHz，具有高达 64 KB 的闪存和高达 8 KB 的 SRAM。一整套节能模式使其可用于低功耗应用。提供广泛的增强型外设和 I/O，允许该器件在不同的应用中采用。一整套节能模式使其可用于低功耗应用。Flash 程序存储器可以通过 SWD 接口在系统内重新编程。

ES32F0541 使用独特的电机控制模块（MCM）提供不同类型的电机控制算法。MCM (Motor Control Module) 为针对电机应用所集成的控制模块，通过配置其中的独立模块可实现不同的电机控制算法如无感测速度积分、FOC (field orthogonal control) 滑模观测器等。MCM 包含了滑动模式控制器的虚拟电机模型与用于计算角度的反正切运算模块 CORDIC、三个独立的 PI (Proportional-Integral) 控制器，可用于调节电流、转速等电机控制参数。藉由 MCM 中的高速运算模块，使得 MCU 不须处理制式的数学运算，而有更为充裕的时间做系统管理，实现更为高效、强健的电机控制器。

ES32F0541 微控制器外设集还包括外部总线接口，3 路 UART，2 路 SUART，2 路 I2C，2 路 SPI/I2S，7 路通用 16 位定时器，1 路通用 32 位定时器，1 路高级 16 位定时器，1 路 RTC，1 路 ADC，1 路 12 位 DAC 和 4 路模拟比较器。

主要用于 BLDC/PSMS 电机应用，如电动工具、园林工具、吸尘器、风机和风扇等电机控制领域。本章介绍了 ES32F0541 的特点，其系统和内存结构：

- ◆ ES32F0541 系统体系结构
- ◆ ES32F0541 系统特点
- ◆ ES32F0541 内存映射

		ES32F0541LT	ES32F0541LQ	ES32F0541LK	ES32F0541NK
Flash 存储(KBytes)		64	64	64	64
SRAM(KBytes)		8	8	8	8
GPIO		55	39	28	29
12-bit ADC		1 (16 channels)	1 (10 channels)		
12-bit DAC		1	1	1	1
模拟比较器(CMP)		4	4	4	4
温度传感器(TEMP)		1	1	1	1
无刷马达矢量控制器(MCM)		1	1	1	1
定时器	通用 GP32C4T	1	1	1	1
	通用 GP16C4T	3	3	3	3
	通用 GP16C2T	4	4	4	4
	基本 BS16T	1	1	1	1
	高级控制 AD16C4T	1	1	1	1
通信	I2C	2	2	2	2
	SPI/I2S	2	2	2	2
	UART	3	2	2	2
	SUART	2	1	0	0
CPU 运行频率		48 MHz			
工作电压范围		2.2V~5.5V			
封装		LQFP64	LQFP48	LQFP32	QFN32

表 1-1 ES32F0541 设备摘要

1.2 结构图

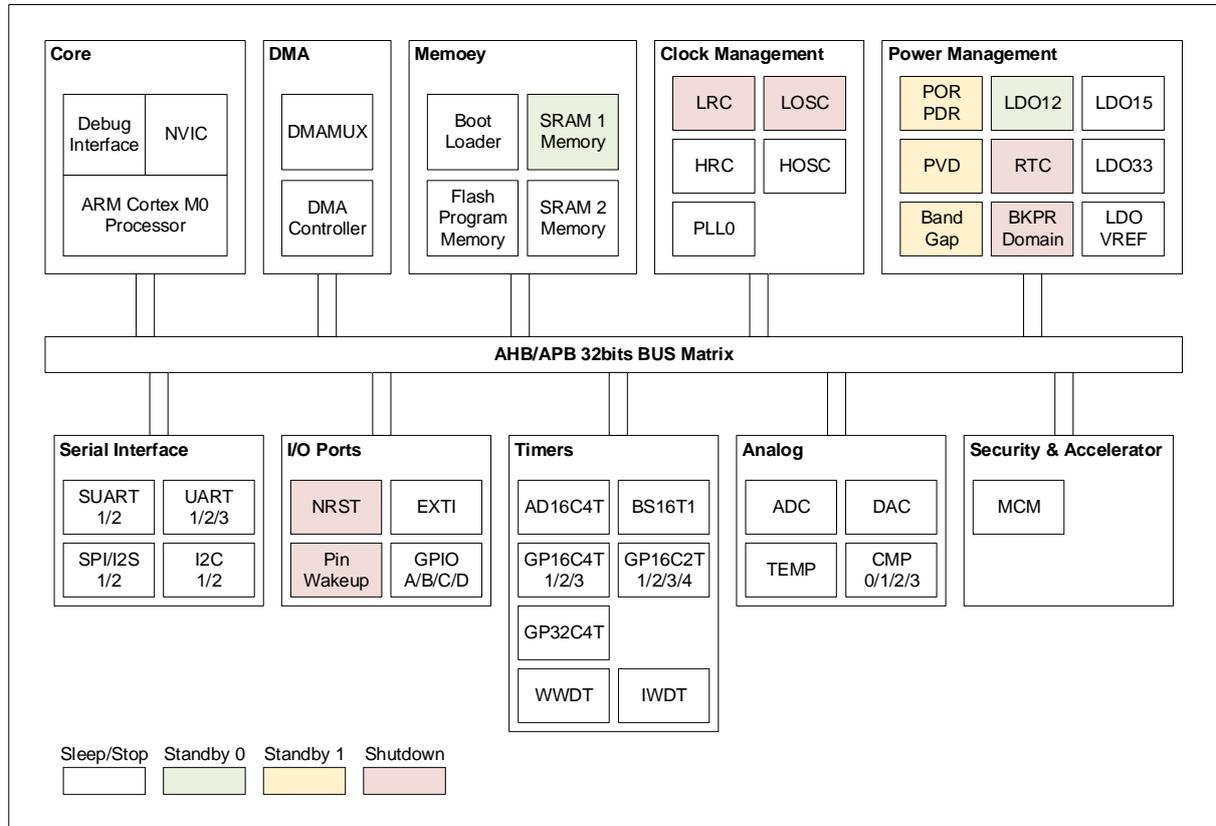


图 1-1 ES32F0541 系统框图

注意，仅在单线和智能卡模式下，UARTx_TX 用于发送和接收数据。

注意，SPI 模块可以配置为 I2S 模块，I2S_MCK 只能输出。

1.3 功能描述

本节概述 ES32F0541 微控制器特点。关于这些特征的进一步信息在它们各自的章节中描述。

特征	描述
处理器	ARM Cortex-M0处理器
性能	48 MHz操作；0.9 DMIPS性能
闪存	8 KB讯息区；64 KB程序区
系统静态随机存取存储器	8 KB的SRAM
系统	
功率控制	上电/掉电复位 (POR/PDR) 可编程电压检测器 (PVD) 电压调节器
低功耗模式	五种低功耗模式： 睡眠模式(Sleep) 停止模式(Stop) 待机模式0(Standby0) 待机模式1(Standby1) 停止运转模式(Shutdown)
时钟控制	一个4至48 MHz锁相环倍频时钟 (PLL0) 32 kHz内部低速RC振荡器 (LRC) 4 MHz内部高速RC振荡器 (HRC) 4-32 MHz外部高速晶体振荡器 (HOSC) 用于RTC和低功耗块 (LOSC) 的32.768 kHz外部低速晶体振荡器
计时器	
通用定时器类型A和B： GP16C4T1， GP16C4T2，GP16C4T3，GP32 C4T1 通用定时器定时器C型：GP16C2T1， GP16C2T 2，GP16C2T3，GP16C2T4 高级控制定时器：AD16C4T1 基本定时器：BS16T1	通用定时器类型A和B： 通过同步或事件链接的计时器链接功能一起工作或与高级控制计时器一起工作 可以在调试模式下冻结计数器。 类型A基于32位自动重载上/下计数器和16位预分频器。 B型是基于16位自动重载上/下计数器和16位预分频器。 PWM或单脉冲模式输出 4捕获/比较通道 通用定时器类型C： 16位自动重载上数计数器和16位预分频器 可编程插入死区的互补PWM输出 2捕获/比较通道和1互补输出 高级控制定时器： 可编程插入死区的互补PWM输出 4捕获/比较通道和3互补输出 基本定时器： 16位自动重载上数计数器和16位预分频器

特征	描述
窗口看门狗(WWDT)	一个WWDT 7位自由运行计数器,可编程超时间隔
独立看门狗(IWDT)	一个IWDT 12位自由运行下数计数器和8位预分频器
实时时钟(RTC)	睡眠模式下的支持计数 时间: 小时, 分钟, 秒 日历: 年、月、日、周、闰年 闹铃: 年、月、日、周、小时、分钟、秒。(单一次/周期)
输入输出端口	
通用输入输出 (GPIO)	四个物理GPIO块 外部中断能力(边沿或电平灵敏度) 去抖动能力
扩展中断和事件控制器 (EXTI)	生成多达22个事件/中断请求 每一个可以独立配置以选择触发事件(上升沿、下降沿, 两者皆有), 并且可以独立屏蔽 多达64个GPIOs可以连接到16个外部中断线
模拟	
模数转换器 (ADC)	一个多达16个通道的12位ADC模块 硬件序列发生器以减轻CPU负载 温度传感器 支持单差分转换
模拟比较器 (CMP)	四个模拟比较器 可编程去抖计数器
数模转换器 (DAC)	12位DAC 支持高达200 kHz的转换 提供锯齿波和三角波生成
DMA	
直接存储器存取 (DMA)	多达6个独立可配置的信道(请求) 外设到存储器、存储器到外设和存储器到存储器传输
串行接口	
通用异步接收机/发射机 (UART)	两个SUART: 可编程波特率发生器 自动波特率检测 自动硬件流量控制 可编程(CTS _n , RTS _n)触发电平 提供TX和RX FIFO, 深度为16 三个UART: 单线半双工通信 支持智能卡通信、IrDA SIR ENDEC、LIN主/从能力和Modbus通信 提供TX和RX FIFO, 深度为16
内部集成电路 (I2C)	两个I2C模块: 支持主从模式

特征	描述
	支持不同的通信速度（100k/400k/1 MHz） 7位/ 10位寻址的产生与检测 可编程I2C地址检测 SMBus能力 PMBUS规范Rev 1.1兼容性 提供TX和RX FIFO，深度为8
串行外设接口 (SPI)	三线全双工同步传输 双线（双向数据线）的半双工同步传输 双线（单向数据线）的单工同步传输 8位至16位传输帧格式选择 支持SPI TI模式 主从操作 提供TX和RX FIFO，深度为16
集成电路内置音频总线 (I2S)	两个SPI模块，可以配置为I2S模块。 主从模式 四个数据和数据包帧可用
加速器	
硬件除法器 (HDIV)	有符号（2的补码）整数计算 32位被除数与32位除数计算能力 32位商和32位余数输出 2~17个HCLK时钟的单周期计算
电机控制模块 (MCM)	适用于电流重构低滤波器的三个独立第一阶数字IIR滤波器 SVPWM（空间矢量PWM）发生器 可配置参数支持不同电机的虚拟电机模型及滑模控制
调试配置	
串行线调试 (SWD)	提供了一个ARM SWD接口，以允许串行线调试工具连接到MCU。

表 1-2 ES32F0541 微控制器特性

边界地址		大小 (Byte)	外设	外设寄存器映射	总线
起点	终点				
0xE000_0000	0xE00F_FFFF	1M	ARM® Cortex™-M0 System Timer (SysTick)	第 3 章	
0x4800_1800	0x5FFF_FFFF		Reserved		AHB2
0x4800_0C00	0x4800_0FFF	1K	GPIO D	第 7 章	
0x4800_0800	0x4800_0BFF	1K	GPIO C	第 7 章	
0x4800_0400	0x4800_07FF	1K	GPIO B	第 7 章	
0x4800_0000	0x4800_03FF	1K	GPIO A	第 7 章	
0x4002_4000	0x47FF_FFFF		Reserved		AHB1
0x4002_3C00	0x4002_3FFF	1K	HDIV	第 9 章	
0x4002_2400	0x4002_3BFF		Reserved		
0x4002_2000	0x4002_23FF	1K	FLASH interface	第 6 章	
0x4002_1C00	0x4002_1FFF		Reserved		
0x4002_1800	0x4002_1BFF	1K	RTC	第 22 章	
0x4002_1400	0x4002_17FF	1K	SYSCFG	第 4 章	
0x4002_1000	0x4002_13FF	1K	RCU	第 5 章	
0x4002_0400	0x4002_0FFF		Reserved		
0x4002_0000	0x4002_03FF	1K	DMA1	第 8 章	
0x4001_6000	0x4001_FFFF		Reserved		APB2
0x4001_5C00	0x4001_5FFF	1K	CMP	第 13 章	
0x4001_5800	0x4001_5BFF		Reserved		
0x4001_5400	0x4001_57FF	1K	MCM	第 25 章	
0x4001_5000	0x4001_53FF		Reserved		
0x4001_4C00	0x4001_4FFF	1K	GP16C2T4	第 15 章	
0x4001_4800	0x4001_4BFF	1K	GP16C2T3	第 15 章	
0x4001_4400	0x4001_47FF	1K	GP16C2T2	第 15 章	
0x4001_4000	0x4001_43FF	1K	GP16C2T1	第 15 章	
0x4001_3C00	0x4001_3FFF		Reserved		
0x4001_3800	0x4001_3BFF	1K	UART1	第 10 章	
0x4001_3400	0x4001_37FF		Reserved		
0x4001_3000	0x4001_33FF	1K	SPI1/I2S1	第 24 章	
0x4001_2C00	0x4001_2FFF	1K	AD16C4T1	第 18 章	
0x4001_2800	0x4001_2BFF		Reserved		
0x4001_2400	0x4001_27FF	1K	ADC	第 12 章	
0x4001_0800	0x4001_23FF		Reserved		
0x4001_0400	0x4001_07FF	1K	EXTI	第 11 章	
0x4001_0000	0x4001_03FF		Reserved		
0x4000_6000	0x4000_FFFF		Reserved		
0x4000_5C00	0x4000_5FFF	1K	DAC	第 14 章	
0x4000_5800	0x4000_5BFF	1K	I2C2	第 23 章	
0x4000_5400	0x4000_57FF	1K	I2C1	第 23 章	
0x4000_5000	0x4000_53FF	1K	SUART2	第 10 章	
0x4000_4C00	0x4000_4FFF	1K	SUART1	第 10 章	
0x4000_4800	0x4000_4BFF	1K	UART3	第 10 章	

边界地址		大小 (Byte)	外设	外设寄存器映射	总线	
起点	终点					
0x4000_4400	0x4000_47FF	1K	UART2	第 10 章		
0x4000_3C00	0x4000_43FF		Reserved			
0x4000_3800	0x4000_3BFF	1K	SPI2/I2S2	第 24 章		
0x4000_3400	0x4000_37FF		Reserved			
0x4000_3000	0x4000_33FF	1K	IWDT	第 20 章		
0x4000_2C00	0x4000_2FFF	1K	WWDT	第 21 章		
0x4000_1400	0x4000_2BFF		Reserved			
0x4000_1000	0x4000_13FF	1K	BS16T1	第 19 章		
0x4000_0C00	0x4000_0FFF	1K	GP16C4T3	第 16 章		
0x4000_0800	0x4000_0BFF	1K	GP16C4T2	第 16 章		
0x4000_0400	0x4000_07FF	1K	GP16C4T1	第 16 章		
0x4000_0000	0x4000_03FF	1K	GP32C4T1	第 17 章		
0x2000_2000	0x3FFF_FFFF		Reserved			SRAM
0x2000_0000	0x2000_1FFF	4K	SRAM2			
0x2000_0000	0x2000_0FFF	4K	SRAM1			
0x1FFF_FC00	0x1FFF_FFFF	1K	Option2 bytes (programmable)		FLASH INFO	
0x1FFF_F800	0x1FFF_FBFF	1K	Option1 bytes			
0x1FFF_E000	0x1FFF_F7FF	6K	Boot Loader memory			
0x1000_1000	0x1FFF_DFFF		Reserved		SRAM	
0x1000_0000	0x1000_0FFF	4K	SRAM1 (Library)			
0x0801_0000	0x0FFF_FFFF		Reserved		FLASH MAIN	
0x0800_0000	0x0800_FFFF	64K	Main Flash memory			
0x0001_0000	0x07FF_FFFF		Reserved		MAIN	
0x0000_0000	0x0000_FFFF	64K	Main Flash memory, system memory or SRAM depending on BOOT Configuration			

表 1-3 外设寄存器边界地址

第2章 引脚和信号

2.1 引脚

2.1.1 ES32F0541 LQFP – 64 引脚

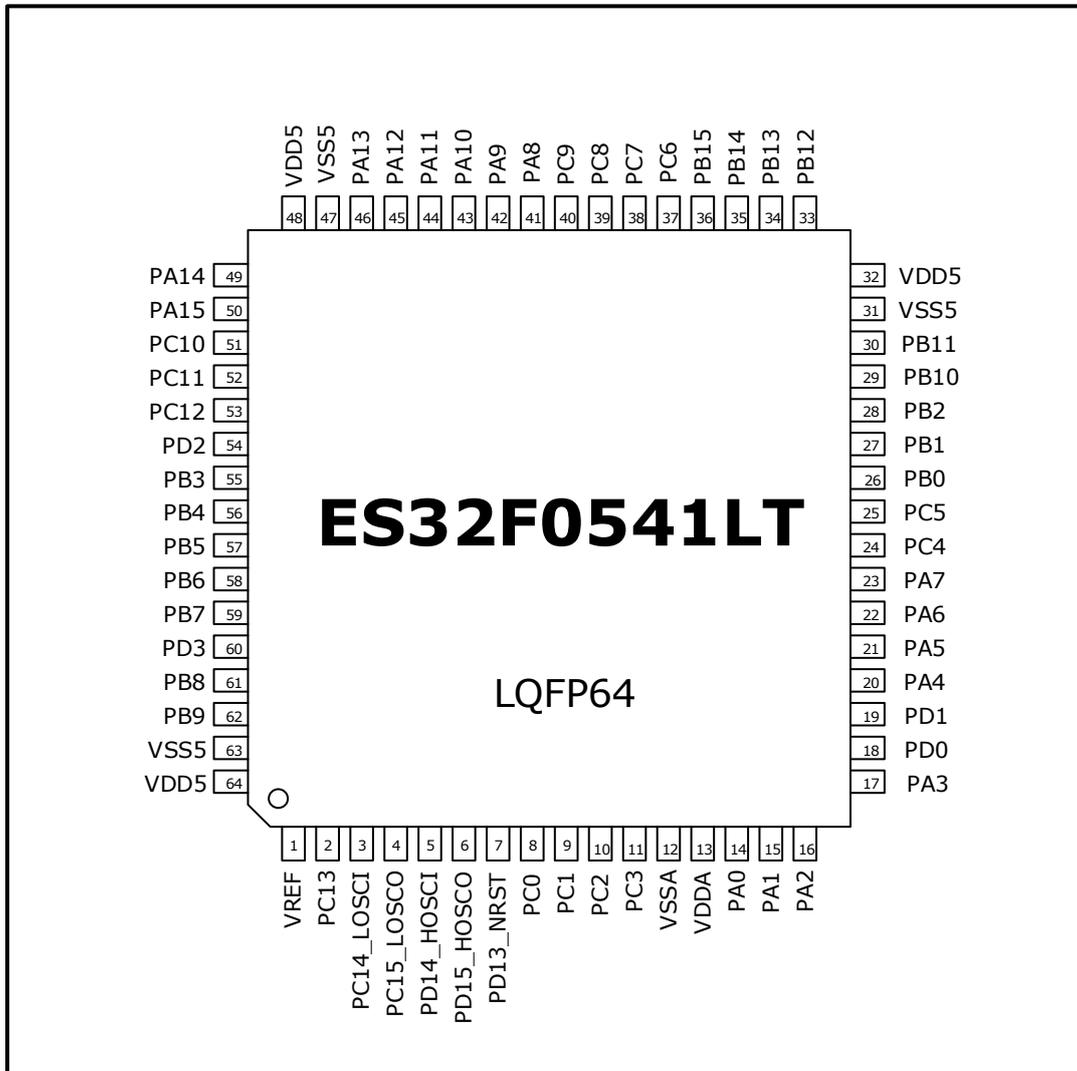


图 2-1 ES32F0541 64-Pin LQFP 封装引脚图

2.1.2 ES32F0541 LQFP – 48 引脚

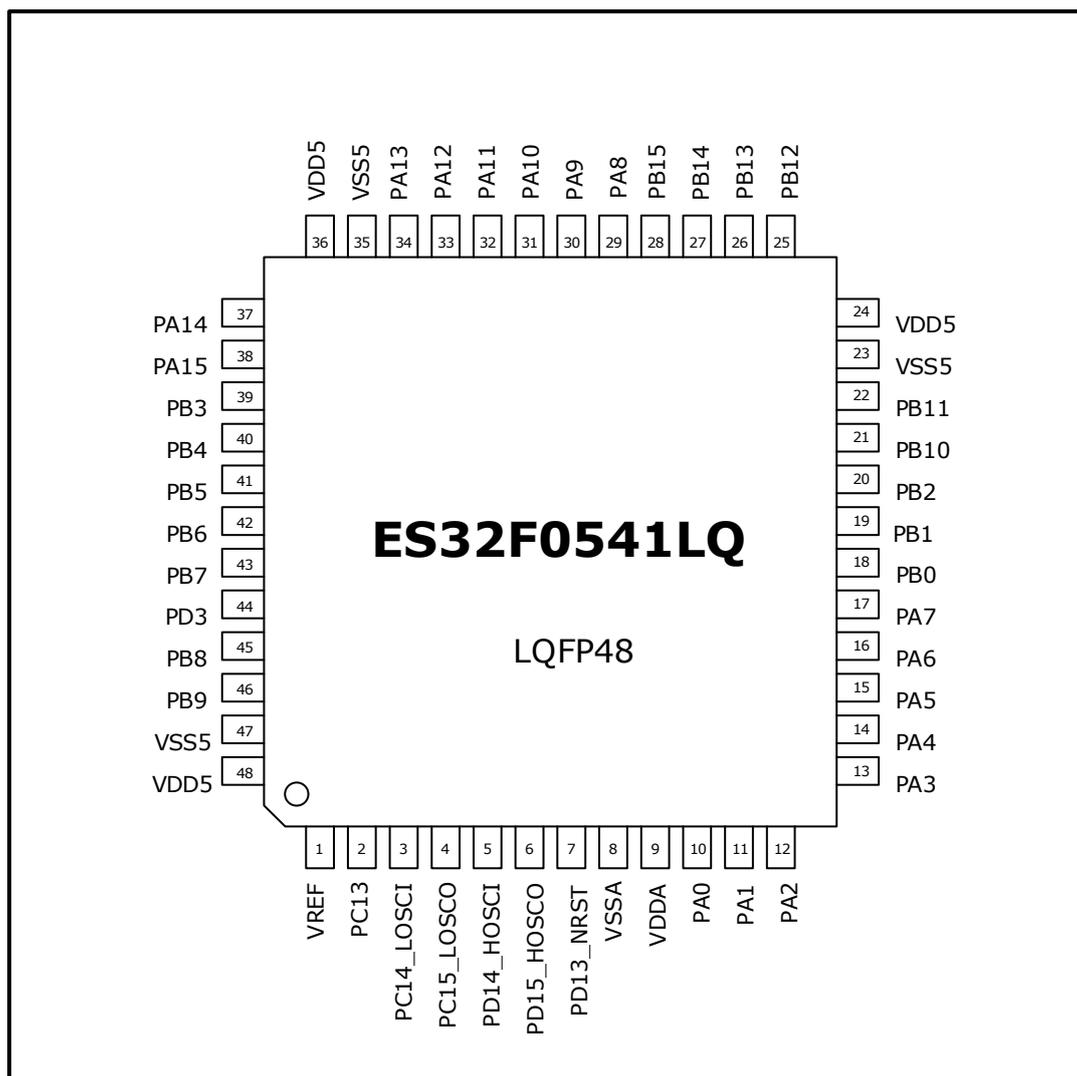


图 2-2 ES32F0541 48-Pin LQFP 封装引脚图

2.1.3 ES32F0541 LQFP – 32 引脚

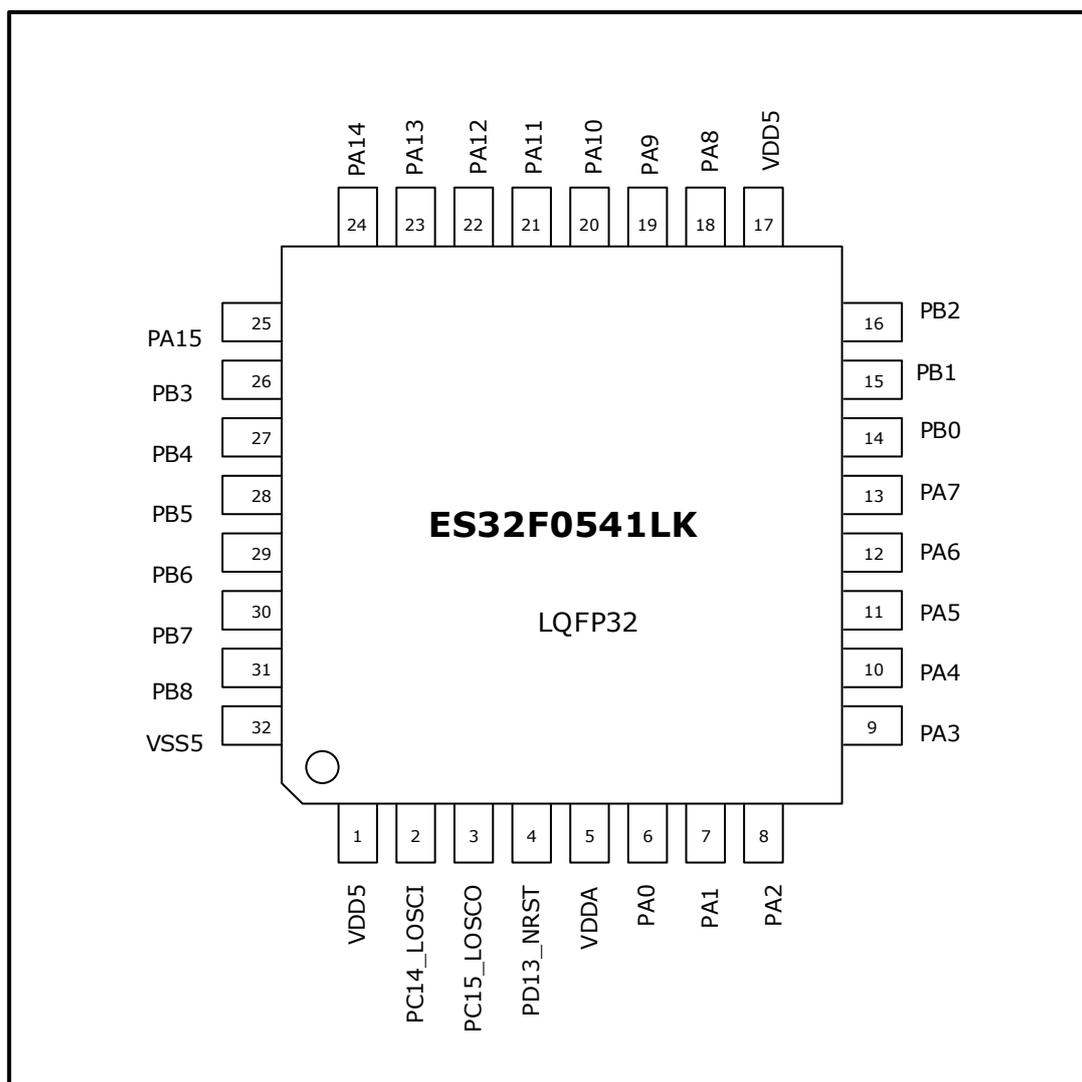


图 2-3 ES32F0541 32-Pin LQFP 封装引脚图

2.1.4 ES32F0541 QFN – 32 引脚

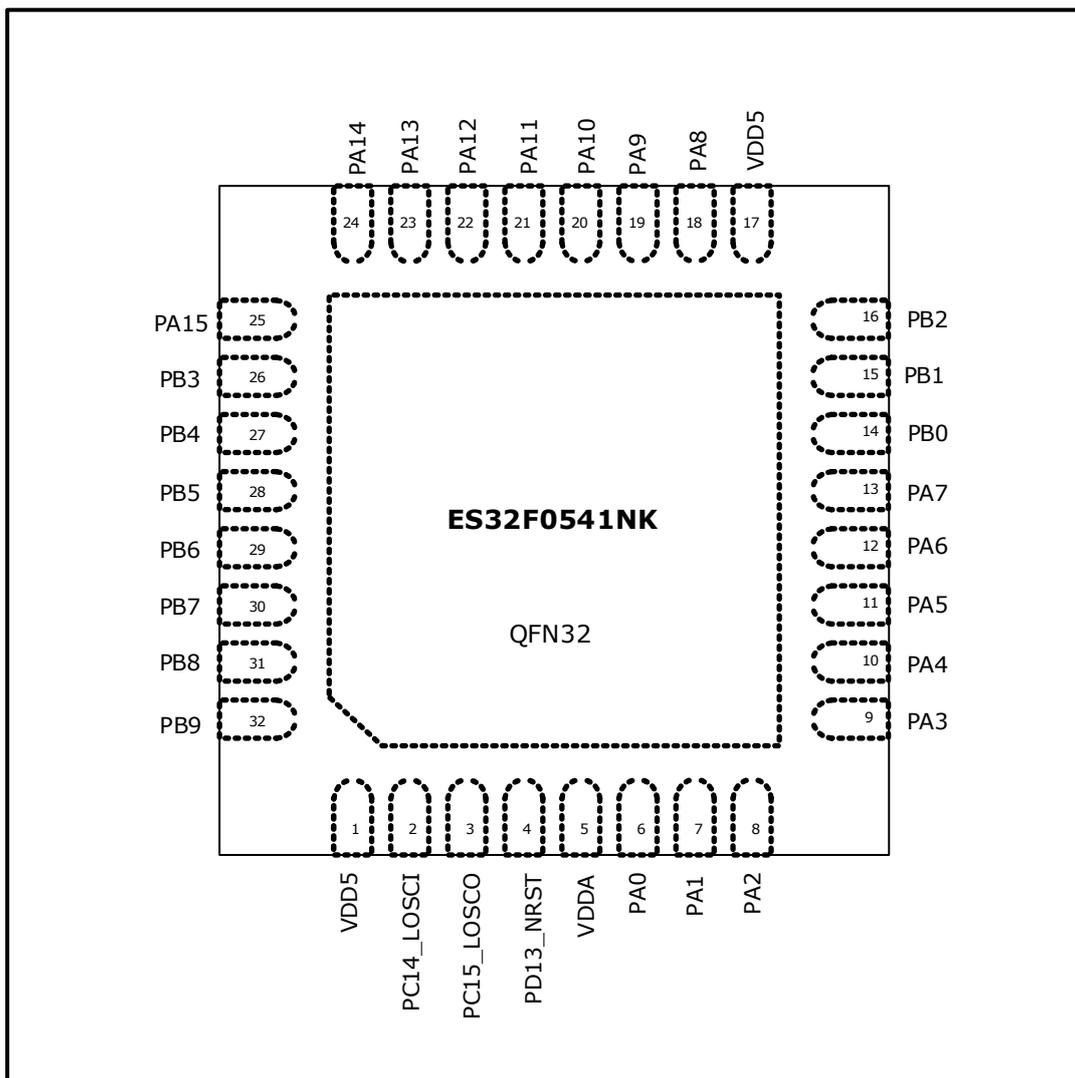


图 2-4 ES32F0541 32-Pin QFN 封装引脚图

2.2 ES32F0541 引脚定义

引脚类型:

P: 电源引脚

G: 接地引脚

I/O: 输入输出引脚

引脚				引脚名	引脚类型	备注	替代函数	模拟函数
LQFP64	LQFP48	LQFP32	QFN32					
1	1	-	-	VREF	P	-	-	-
2	2	-	-	PC13	I/O	-	I2S_CLK1	-
3	3	2	2	PC14_LOSCI	I/O	-	-	LOSCI
4	4	3	3	PC15_LOSCO	I/O	-	-	LOSCO
5	5	-	-	PD14_HOSCI	I/O	-	-	HOSCI
6	6	-	-	PD15_HOSCO	I/O	-	-	HOSCO
7	7	4	4	PD13_NRST	I/O	-	-	NRST
8	-	-	-	PC00	I/O	-	SPI2_SCK, SUART2_TX, GP16C4T3_CH1	ADIN10
9	-	-	-	PC01	I/O	-	SPI2_NSS, SUART2_RX, GP16C4T3_CH3	ADIN11
10	-	-	-	PC02	I/O	-	SPI2_MISO, SUART2_CTS, GP16C4T3_CH2	ADIN12
11	-	-	-	PC03	I/O	-	SPI2_MOSI, SUART2_RTS_DE, GP16C4T3_ETR, GP32C4T1_ETR	ADIN13
12	8	-	-	VSSA	G	-	-	-
13	9	5	5	VDDA	P	-	-	-
14	10	6	6	PA00	I/O	-	SPI2_SCK, UART2_CTS, GP32C4T1_CH1, GP32C4T1_ETR, GP16C2T1_BKIN, CMP2_OUT	ADIN0, CMPIP1
15	11	7	7	PA01	I/O	-	SPI2_NSS, I2C2_SMBA, UART2_RTS_DE, GP32C4T1_CH2, GP16C4T1_ETR, GP16C2T1_CH1N	ADIN1, CMPIN1
16	12	8	8	PA02	I/O	-	SPI2_MOSI, I2C2_SCL,	ADIN2, CMPIP2

引脚				引脚名	引脚类型	备注	替代函数	模拟函数
LQFP64	LQFP48	LQFP32	QFN32					
							UART2_TX, GP32C4T1_CH3, GP16C4T1_CH1, GP16C2T1_CH1, CMP2_OUT	
17	13	9	9	PA03	I/O	-	SPI2_MISO, I2C2_SDA, UART2_RX, GP32C4T1_CH4, GP16C4T1_CH2, GP16C2T1_CH2	ADIN3, CMPIN2
18	-	-	-	PD00	I/O	-	GP16C4T1_CH3	-
19	-	-	-	PD01	I/O	-	GP16C4T1_CH4	-
20	14	10	10	PA04	I/O	-	SPI1_NSS, UART2_CK, GP16C4T2_ETR, GP16C4T1_CH1, GP16C2T4_CH1,	ADIN4, CMPIP3, DAC_OUT
21	15	11	11	PA05	I/O	-	SPI1_SCK, GP32C4T1_ETR, GP16C4T2_CH3, GP16C4T1_CH2, GP16C2T2_CH1N, GP32C4T1_CH1	ADIN5, CMPIN3
22	16	12	12	PA06	I/O	-	SPI1_MISO, SUART1_CTS, GP16C4T2_CH1, GP16C4T1_CH3, GP16C2T2_CH1, AD16C4T1_BKIN, CMP1_OUT	ADIN6, CMPIP4
23	17	13	13	PA07	I/O	-	SPI1_MOSI, GP16C2T4_CH1, GP16C4T2_CH2, GP16C4T1_CH4, GP16C2T3_CH1, AD16C4T1_CH1N, CMP2_OUT	ADIN7, CMPIN4
24	-	-	-	PC04	I/O	-	SUART1_TX, GP16C4T1_ETR	ADIN14
25	-	-	-	PC05	I/O	-	SUART1_RX	ADIN15
26	18	14	14	PB00	I/O	-	UART1_TX, GP16C4T2_CH3, GP16C2T1_BKIN, AD16C4T1_CH2N	ADIN8, WKUP0
27	19	15	15	PB01	I/O	-	UART1_RX, SUART1_RTS_DE, GP16C4T2_CH4, GP16C2T4_CH1, GP16C2T1_CH2, AD16C4T1_CH3N	ADIN9, WKUP1

引脚				引脚名	引脚类型	备注	替代函数	模拟函数
LQFP64	LQFP48	LQFP32	QFN32					
28	20	16	16	PB02	I/O	-	I2C2_SMBA, GP16C4T3_CH3, GP16C2T1_CH1	WKUP2
29	21	-	-	PB10	I/O	-	SPI2_SCK, I2C2_SCL, SUART1_TX, GP32C4T1_CH3, GP16C4T3_CH4, AD16C4T1_CH4, CMP3_OUT	-
30	22	-	-	PB11	I/O	-	I2C2_SDA, SUART1_RX, GP32C4T1_CH4, GP16C4T3_CH3, AD16C4T1_CH3, CMP4_OUT	-
31	23	-	-	VSS5	G	-	-	-
32	24	17	17	VDD5	P	-	-	-
33	25	-	-	PB12	I/O	-	SPI2_NSS, I2C2_SMBA, SUART1_RTS_DE, GP16C2T3_CH1, GP16C2T4_CH1, AD16C4T1_BKIN, CMP3_OUT	-
34	26	-	-	PB13	I/O	-	SPI2_SCK, I2C2_SCL, SUART1_CTS, GP16C4T1_CH1, GP16C2T2_CH1, GP16C2T4_CH1N, AD16C4T1_CH1N, CMP4_OUT	-
35	27	-	-	PB14	I/O	-	SPI2_MISO, I2C2_SDA, SUART1_RTS_DE, GP16C4T1_CH2, GP16C2T1_CH1, GP16C2T4_CH2, AD16C4T1_CH2N, RTC_OUT	-
36	28	-	-	PB15	I/O	-	SPI2_MOSI, UART3_CK, GP16C2T1_CH2, GP16C2T1_CH1N, GP16C2T4_BKIN, AD16C4T1_CH3N	-
37	-	-	-	PC06	I/O	-	UART3_TX, GP16C4T2_CH1	-
38	-	-	-	PC07	I/O	-	UART3_RX, GP16C4T2_CH2	-

引脚				引脚名	引脚类型	备注	替代函数	模拟函数
LQFP64	LQFP48	LQFP32	QFN32					
39	-	-	-	PC08	I/O	-	UART3_CTS, GP16C4T2_ETR, GP16C4T2_CH3	-
40	-	-	-	PC09	I/O	-	UART3_RTS_DE, GP16C4T1_ETR, GP16C4T2_CH4	-
41	29	18	18	PA08	I/O	-	I2S_CLK2, UART1_CK, GP16C2T3_CH1, AD16C4T1_CH1, MUX_CLK_OUT,	-
42	30	19	19	PA09	I/O	-	UART1_TX, GP16C2T1_BKIN, GP16C2T3_CH1N, AD16C4T1_CH2, MUX_CLK_OUT,	-
43	31	20	20	PA10	I/O	-	UART1_RX, GP16C2T3_BKIN, AD16C4T1_CH3,	-
44	32	21	21	PA11	I/O	-	SPI1_MISO, UART1_CTS, AD16C4T1_CH4, CMP1_OUT	-
45	33	22	22	PA12	I/O	-	SPI1_MOSI, UART1_RTS_DE, AD16C4T1_ETR, CMP2_OUT	-
46	34	23	23	PA13	I/O	-	SWDIO, IR_OUT, GP16C2T3_CH2, AD16C4T1_CH1,	-
47	35	-	-	VSS5	G	-	-	-
48	36	-	-	VDD5	P	-	-	-
49	37	24	24	PA14	I/O	-	SWCLK, UART2_TX, GP16C4T2_CH4, AD16C4T1_BKIN, CMP1_OUT,	-
50	38	25	25	PA15	I/O	-	SPI1_NSS, UART2_RX, GP32C4T1_ETR, GP32C4T1_CH1, GP16C4T2_CH3, CMP2_OUT,	-
51	-	-	-	PC10	I/O	-	SUART1_TX	-
52	-	-	-	PC11	I/O	-	SUART1_RX	-
53	-	-	-	PC12	I/O	-	I2C1_SCL	-

引脚				引脚名	引脚类型	备注	替代函数	模拟函数
LQFP64	LQFP48	LQFP32	QFN32					
54	-	-	-	PD02	I/O	-	I2C1_SDA, SUART1_RTS_DE, GP16C4T2_ETR	-
55	39	26	26	PB03	I/O	-	SPI1_SCK, GP32C4T1_CH2, GP16C2T2_CH2, AD16C4T1_CH3N,	WKUP3
56	40	27	27	PB04	I/O	-	SPI1_MISO, GP16C4T2_CH1, AD16C4T1_CH2N,	WKUP4
57	41	28	28	PB05	I/O	-	SPI1_MOSI, I2C1_SMBA, GP16C4T2_CH2, GP16C4T3_CH1, GP16C2T2_BKIN, AD16C4T1_CH1N,	WKUP5
58	42	29	29	PB06	I/O	-	I2C1_SCL, UART1_TX, GP16C4T3_ETR, GP16C2T2_CH1N, AD16C4T1_CH3, CMP3_OUT,	WKUP6
59	43	30	30	PB07	I/O	-	I2C1_SDA, UART1_RX, GP16C4T3_CH2, GP16C2T3_CH1N, AD16C4T1_CH2, CMP4_OUT,	WKUP7
60	44	-	-	PD03	I/O	-	SPI1_MOSI, I2C1_SMBA, AD16C4T1_CH1,	-
61	45	31	31	PB08	I/O	-	I2C1_SCL, IR_OUT, GP16C4T3_CH1, GP16C2T2_CH1, AD16C4T1_BKIN	-
62	46	-	32	PB09	I/O	-	SPI2_NSS, I2C1_SDA, IR_OUT, GP16C4T3_CH2, GP16C2T3_CH1, AD16C4T1_ETR	-
63	47	32	-	VSS5	G	-	-	-
64	48	1	1	VDD5	P	-	-	-

表 2-1 引脚定义

2.3 多路复用引脚功能选择

下表描述了多路复用到 GPIO 线的外设信号。复用功能（AF）通过配置 **GPIOx_MOD** 启用和通过 **GPIOx_AFL** 和 **GPIOx_AFH** 寄存器选择复用功能。

GPIOx_MOD 控制:

MODy<1:0>	描述
00	GPIO 输入方式
01	GPIO 输出模式
10	复用功能模式
11	模拟模式（复位状态）

注 1: y = 0~15.

注 2: 默认情况下, 所有引脚都处于模拟模式。

注 3: I2S 与 SPI 共享引脚, 这些引脚可以被配置为 I2S 或 SPI 协议取决于用户的设置。

引脚名	复用功能								
	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8
PA00	SPI2_SCK	-	UART2_CTS	GP32C4T1_CH1	GP32C4T1_ETR	GP16C2T1_BKIN	-	CMP1_OUT	-
PA01	SPI2_NSS	I2C2_SMBA	UART2_RTS_DE	GP32C4T1_CH2	GP16C4T1_ETR	GP16C2T1_CH1N	-	-	-
PA02	SPI2_MOSI	I2C2_SCL	UART2_TX	GP32C4T1_CH3	GP16C4T1_CH1	GP16C2T1_CH1	-	CMP2_OUT	-
PA03	SPI2_MISO	I2C2_SDA	UART2_RX	GP32C4T1_CH4	GP16C4T1_CH2	GP16C2T1_CH2	-	-	-
PA04	SPI1_NSS	-	UART2_CK	GP16C4T2_ETR	GP16C4T1_CH1	GP16C2T4_CH1	-	-	-
PA05	SPI1_SCK	-	GP32C4T1_ETR	GP16C4T2_CH3	GP16C4T1_CH2	GP16C2T2_CH1N	GP32C4T1_CH1	-	-
PA06	SPI1_MISO	-	SUART1_CTS	GP16C4T2_CH1	GP16C4T1_CH3	GP16C2T2_CH1	AD16C4T1_BKIN	CMP1_OUT	-
PA07	SPI1_MOSI	-	GP16C2T4_CH1	GP16C4T2_CH2	GP16C4T1_CH4	GP16C2T3_CH1	AD16C4T1_CH1N	CMP2_OUT	-
PA08	I2S_CLK2	-	UART1_CK	-	-	GP16C2T3_CH1	AD16C4T1_CH1	MUX_CLK_OUT	-
PA09	-	-	UART1_TX	GP16C2T1_BKIN	-	GP16C2T3_CH1N	AD16C4T1_CH2	MUX_CLK_OUT	-
PA10	-	-	UART1_RX	-	-	GP16C2T3_BKIN	AD16C4T1_CH3		-
PA11	SPI1_MISO	-	UART1_CTS	-	-	-	AD16C4T1_CH4	CMP1_OUT	-
PA12	SPI1_MOSI	-	UART1_RTS_DE	-	-	-	AD16C4T1_ETR	CMP2_OUT	-
PA13	SWDIO	-	IR_OUT	-	-	GP16C2T3_CH2	AD16C4T1_CH1		-
PA14	SWCLK	-	UART2_TX	-	-	GP16C4T2_CH4	AD16C4T1_BKIN	CMP1_OUT	-
PA15	SPI1_NSS	-	UART2_RX	GP32C4T1_ETR	GP32C4T1_CH1	GP16C4T2_CH3	-	CMP2_OUT	-

表 2-2 复用功能端口 A

引脚名	复用功能								
	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8
PB00	UART1_TX	-	-	GP16C4T2_CH3	-	GP16C2T1_BKIN	AD16C4T1_CH2N	-	-
PB01	UART1_RX	-	SUART1_RTS_DE	GP16C4T2_CH4	GP16C2T4_CH1	GP16C2T1_CH2	AD16C4T1_CH3N	-	-
PB02	-	I2C2_SMBA	-	GP16C4T3_CH3	-	GP16C2T1_CH1	-	-	-
PB03	SPI1_SCK	-	-	-	GP32C4T1_CH2	GP16C2T2_CH2	AD16C4T1_CH3N	-	-
PB04	SPI1_MISO	-	-	GP16C4T2_CH1	-	-	AD16C4T1_CH2N	-	-
PB05	SPI1_MOSI	I2C1_SMBA	-	GP16C4T2_CH2	GP16C4T3_CH1	GP16C2T2_BKIN	AD16C4T1_CH1N	-	-
PB06	-	I2C1_SCL	UART1_TX	-	GP16C4T3_ETR	GP16C2T2_CH1N	AD16C4T1_CH3	CMP3_OUT	-
PB07	-	I2C1_SDA	UART1_RX	-	GP16C4T3_CH2	GP16C2T3_CH1N	AD16C4T1_CH2	CMP4_OUT	-
PB08	-	I2C1_SCL	IR_OUT	-	GP16C4T3_CH1	GP16C2T2_CH1	AD16C4T1_BKIN	-	-
PB09	SPI2_NSS	I2C1_SDA	IR_OUT	-	GP16C4T3_CH2	GP16C2T3_CH1	AD16C4T1_ETR	-	-
PB10	SPI2_SCK	I2C2_SCL	SUART1_TX	GP32C4T1_CH3	GP16C4T3_CH4	-	AD16C4T1_CH4	CMP3_OUT	-
PB11	-	I2C2_SDA	SUART1_RX	GP32C4T1_CH4	GP16C4T3_CH3	-	AD16C4T1_CH3	CMP4_OUT	-
PB12	SPI2_NSS	I2C2_SMBA	SUART1_RTS_DE	-	GP16C2T3_CH1	GP16C2T4_CH1	AD16C4T1_BKIN	CMP3_OUT	-
PB13	SPI2_SCK	I2C2_SCL	SUART1_CTS	GP16C4T1_CH1	GP16C2T2_CH1	GP16C2T4_CH1N	AD16C4T1_CH1N	CMP4_OUT	-
PB14	SPI2_MISO	I2C2_SDA	SUART1_RTS_DE	GP16C4T1_CH2	GP16C2T1_CH1	GP16C2T4_CH2	AD16C4T1_CH2N	RTC_OUT	-
PB15	SPI2_MOSI	-	UART3_CK	GP16C2T1_CH2	GP16C2T1_CH1N	GP16C2T4_BKIN	AD16C4T1_CH3N	-	-

表 2-3 复用功能端口 B

引脚名	复用功能								
	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8
PC00	SPI2_SCK	-	SUART2_TX	GP16C4T3_CH1	-	-	-	-	-
PC01	SPI2_NSS	-	SUART2_RX	GP16C4T3_CH3	-	-	-	-	-
PC02	SPI2_MISO	-	SUART2_CTS	GP16C4T3_CH2	-	-	-	-	-
PC03	SPI2_MOSI	-	SUART2_RTS_DE	GP16C4T3_ETR	GP32C4T1_ETR	-	-	-	-
PC04	-	-	SUART1_TX	-	GP16C4T1_ETR	-	-	-	-
PC05	-	-	SUART1_RX	-	-	-	-	-	-
PC06	-	-	UART3_TX	GP16C4T2_CH1	-	-	-	-	-
PC07	-	-	UART3_RX	GP16C4T2_CH2	-	-	-	-	-
PC08	-	-	UART3_CTS	GP16C4T2_ETR	GP16C4T2_CH3	-	-	-	-
PC09	-	-	UART3_RTS_DE	GP16C4T1_ETR	GP16C4T2_CH4	-	-	-	-
PC10	-	-	SUART1_TX	-	-	-	-	-	-
PC11	-	-	SUART1_RX	-	-	-	-	-	-
PC12	-	I2C1_SCL	-	-	-	-	-	-	-
PC13	I2S_CLK1	-	-	-	-	-	-	-	-
PC14_LOSCI	-	-	-	-	-	-	-	-	-
PC15_LOSCO	-	-	-	-	-	-	-	-	-

表 2-4 复用功能端口 C

引脚名	复用功能								
	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8
PD00	-	-	-	-	GP16C4T1_CH3	-	-	-	-
PD01	-	-	-	-	GP16C4T1_CH4	-	-	-	-
PD02	-	I2C1_SDA	SUART1_RTS_DE	-	-	GP16C4T2_ETR	-	-	-
PD03	SPI1_MOSI	I2C1_SMBA	-	-	-	-	AD16C4T1_CH1	-	-
PD13_NEST	-	-	-	-	-	-	-	-	-
PD14_HOSCI	-	-	-	-	-	-	-	-	-
PD15_HOSCO	-	-	-	-	-	-	-	-	-

表 2-5 复用功能端口 D

2.4 信号描述

下表是外围设备信号的详细定义。

信号名	I/O	功能描述
电源供应器		
VDDA	Supply	模拟域5.0 V供电
VREF	Supply	ADC/DAC参考电压
VDD5	Supply	数字域5.0V供电
VSSA	Ground	模拟接地
VSS5	Ground	数字接地
通用异步接收机/发射机 (UART1, UART2, UART3), x = 1, 2, 3		
UARTx_TX	I/O	UARTx 传输数据引脚
UARTx_RX	I/O	UARTx 接收数据引脚
UARTx_CTS	I	UARTx 清除发送引脚
UARTx_RTS_DE	O	UARTx 请求发送引脚
简易通用异步收发两用机 (SUART1, SUART2), x = 1, 2		
SUARTx_TX	O	SUARTx 传输数据引脚
SUARTx_RX	I	SUARTx 接收数据引脚
SUARTx_CTS	I	SUARTx 清除发送引脚
SUARTx_RTS_DE	O	SUARTx 请求发送引脚
串行线调试 (SWD)		
SWCLK	I	SWD 时钟
SWDIO	I/O	SWD 数据输入输出
内部集成电路 (I2C1, I2C2), x = 1, 2		
I2Cx_SCL	I/O	I2Cx 时钟
I2Cx_SDA	I/O	I2Cx 数据
I2Cx_SMBA	I/O	SMBus 警示引脚
串行外设接口 (SPI1, SPI2), x = 1, 2		
SPI_MISO	I/O	SPIx 主输入从属输出
SPI_MOSI	I/O	SPIx 主输出从输入
SPI_NSS	I/O	SPIx 芯片选择
SPI_SCK	I/O	SPIx 时钟
集成电路内置音频总线 (I2S1, I2S2), x = 1, 2		
I2Sx_SD	I/O	I2Sx 接收/发送数据
I2Sx_MCK	O	I2Sx 主时钟
I2Sx_WS	I/O	I2Sx 字选择
I2Sx_CK	I/O	I2Sx 位时钟
通用输入/输出 (GPIOA, GPIOB, GPIOC, GPIOD)		
PA15-PA00	I/O	GPIO 端口A
PB15-PB00	I/O	GPIO 端口B
PC15-PC00	I/O	GPIO 端口C
PD15-PD13	I/O	GPIO 端口D

信号名	I/O	功能描述
PD03-PD00		
通用定时器 (GP16C2Tx), x = 1, 2, 3, 4		
GP16C2Tx_CH1	I/O	GP16C2Tx 通道 1 输入/输出引脚
GP16C2Tx_CH2	I/O	GP16C2Tx 通道 2 输入/输出引脚
GP16C2Tx_CH1N	I/O	GP16C2Tx 通道 1 双相输出引脚
GP16C2Tx_BKIN	I	GP16C2Tx 中断输入
通用定时器 (GP16C4Tx), x = 1, 2, 3		
GP16C4Tx_CH1	I/O	GP16C4Tx 通道 1 输入/输出引脚
GP16C4Tx_CH2	I/O	GP16C4Tx 通道 2 输入/输出引脚
GP16C4Tx_CH3	I/O	GP16C4Tx 通道 3 输入/输出引脚
GP16C4Tx_CH4	I/O	GP16C4Tx 通道 4 输入/输出引脚
GP16C4Tx_ETR	I	GP16C4Tx 外部触发输入
通用定时器 (GP32C4T1)		
GP32C4T1_CH1	I/O	GP32C4T1 通道 1 输入/输出引脚
GP32C4T1_CH2	I/O	GP32C4T1 通道 2 输入/输出引脚
GP32C4T1_CH3	I/O	GP32C4T1 通道 3 输入/输出引脚
GP32C4T1_CH4	I/O	GP32C4T1 通道 4 输入/输出引脚
GP32C4T1_ETR	I	GP32C4T1 外部触发输入引脚
高级控制定时器 (AD16C4T1)		
AD16C4T1_CH1	I/O	AD16C4T1 通道 1 输入/输出引脚
AD16C4T1_CH1N	O	AD16C4T1 通道 1 双相输出引脚
AD16C4T1_CH2	I/O	AD16C4T1 通道 2 输入/输出引脚
AD16C4T1_CH2N	O	AD16C4T1 通道 2 双相输出引脚
AD16C4T1_CH3	I/O	AD16C4T1 通道 3 输入/输出引脚
AD16C4T1_CH3N	O	AD16C4T1 通道 3 双相输出引脚
AD16C4T1_CH4	I/O	AD16C4T1 通道 4 输入/输出引脚
AD16C4T1_BKIN	I	AD16C4T1 中断输入
AD16C4T1_ETR	I	AD16C4T1 外部触发输入
模数转换器 (ADC)		
AD[15:0]	I	ADC单端通道输入 ADC差分通道输入正负输入
模拟比较器 (CMP1, CMP2, CMP3, CMP4), x=1, 2, 3, 4		
CMP1Px	I	Comparator x 正输入
CMP1Px	I	Comparator x 负输入
CMPx_OUT	O	比较器输出
数模转换器 (DAC)		
DAC_OUT	O	DAC输出通道
系统控制 (SC)		
LOSCI	I	32.768 kHz RTC 时钟输入
LOSCO	O	32.768 kHz RTC 时钟输出
HOSCI	I	32 MHz 高速晶体时钟输入
HOSCO	O	32 MHz 高速晶体时钟输出

信号名	I/O	功能描述
NRST	I	系统复位
WKUPx	I	唤醒引脚
MCO	O	微控制器时钟输出
RTC_1Hz	O	RTC 1Hz 时钟脉冲输出

表 2-6 复用功能描述

2.5 电源引使用说明

芯片电源引脚，其中 VREF 是 ADC 与 DAC 的参考电压，内部相关配置如图 2-5 所示。

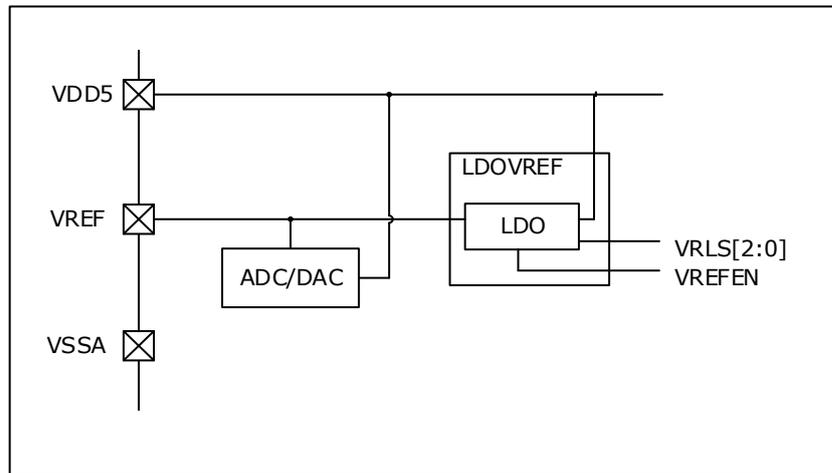


图 2-5 VREF 的内部电源示意图

2.5.1 ADC/DAC 参考电源引脚 VREF

对于 VREF 管脚，它仅在 LQFP64 与 LQFP48 封装上作为外部引脚。在 LQFP32 与 QFN32 封装上它在内部被固定到 5V，因此在这 2 个封装只能使用 5V 当作参考电压，其中 VREFEN 必须设置为 0；在 LQFP64 与 LQFP48 封装，其外部参考电路有以下几种接法：

1. 外部参考电源 VREF 与 VDD5 相等

此应用上，VREF 管脚直接接上电源 VDD5，因此 VREFEN 必须设置为 0，无法使用内部 LDOVREF 产生参考电压，外部参考电路可如图 2-6：

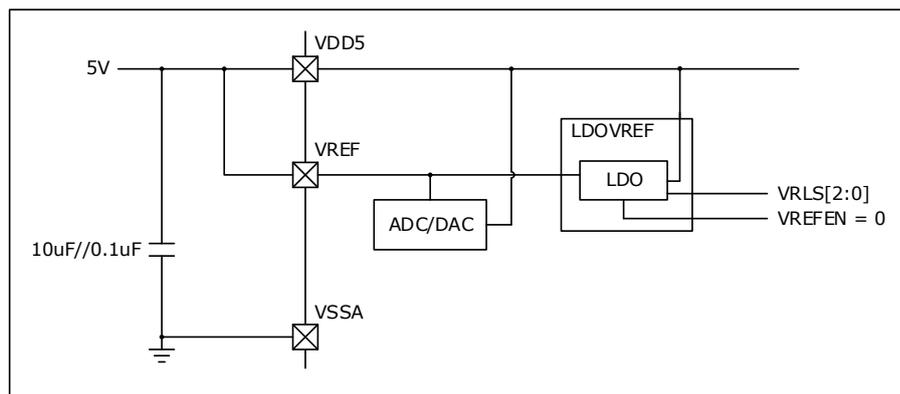


图 2-6 VREF 与 VDD5 电源对接示意图

2. 外部参考电源 VREF 与 VDD5 不相等

此应用上,由外部提供任意参考电源(1.5V~5V)至 VREF 引脚,因此 VREFEN 必须设置为 0,无法使用内部 LDOVREF 产生参考电压,外部参考电路可如图 2-7:

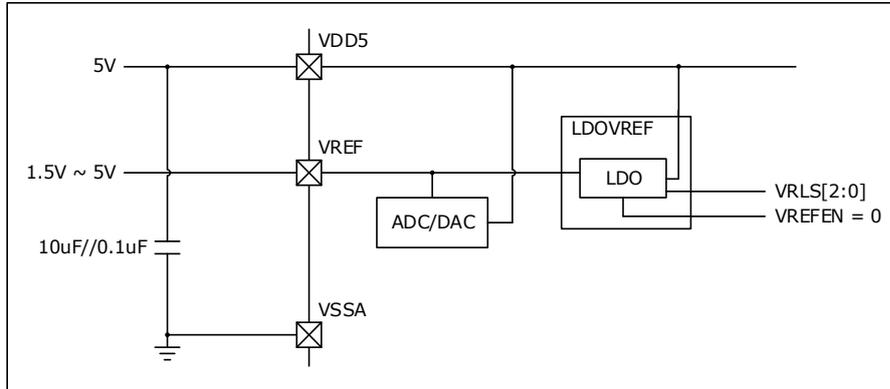


图 2-7 VREF 接上外部参考电源示意图

3. 内部参考电源 LDOVREF

使用内部 LDOVREF 产生参考电压时,VREFEN 需设置为 1,此时 VREF_LDO 会将 VRLS[2:0] 所选择的 LDOVREF 电压输出至 VREF 管脚,且提供内部 ADC/DAC 作为参考电源,为了避免因为电压抖动造成 ADC 与 DAC 的输出结果偏移,建议 VREF 管脚接上 1uF 的稳压电容以提高 LDO 电源稳定。参考电路可如与图 2-8 所示:

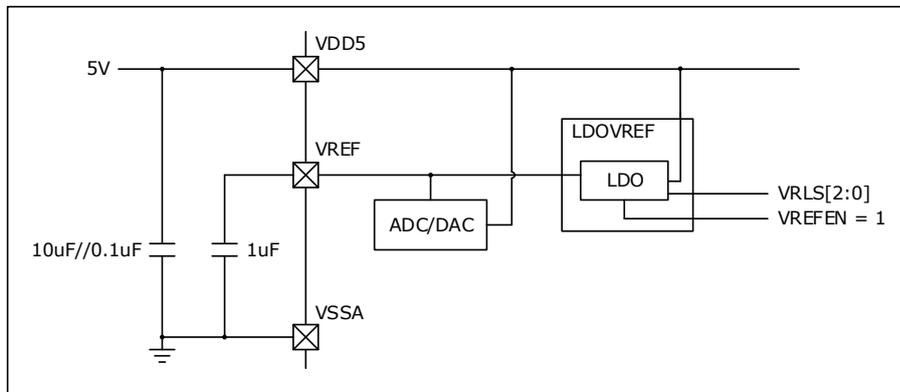


图 2-8 VREF 接稳压电容示意图

第3章 ARM® Cortex™-M0 Core

3.1 概述

ARM Cortex™-M0 处理器是最小型和最节能的 ARM 处理器。它满足了越来越低成本应用的需求，同时增加了连通性。M0 处理器是一个可配置的多级 32 位 RISC 处理器。

3.2 特性

在 ES32F0541 中，该处理器配置以下特征：

- ◆ 内置嵌套向量中断控制器 (NVIC)：32 外部中断
- ◆ 小端序
- ◆ 集成系统计时器 - SysTick
- ◆ 支持暂停调试
- ◆ 快速乘法器
- ◆ 支持串行线调试 (SWD) 连接

本章提供以下处理器外围设备的基本讯息

- ◆ CPU 系统定时器控制 (SysTick)
- ◆ CPU 嵌套向量中断控制器 (NVIC)
- ◆ CPU 系统控制

欲知详情，请参阅：

- ◆ ARM Cortex™-M0 技术参考手册
- ◆ ARM v6-M 结构参考手册

3.3 功能描述

3.3.1 CPU 系统定时器控制寄存器(SYST)

Cortex™-M0 包括一个集成的系统计时器 - **SysTick**，它提供了一个简单的、24 位的写入清零、递减、倒数到 0 重载计数器，并具有灵活的控制机制。计数器可作为实时操作系统(RTOS) 时钟计时器或简单计数器使用。

当系统定时器被启用时，它开始从 **SysTick** 当前值寄存器 (**SYST_CVR**) 中的值倒数到 0，并在下一个时钟周期中重新加载(包装) **SysTick** 重载值寄存器 (**SYST_RVR**) 中的值，然后在随后的时钟上递减。一旦计数器转换为 0，设置 **COUNTFLAG** 状态位。**COUNTFLAG** 位在读取时清除。

复位时，系统的 **SYST_CVR** 值未知。在启用该功能之前，软件应该写入寄存器以将其清除为零。这确保了计时器在启用时将从 **SYST_RVR** 值计数而不是任意值。

如果 **SYST_RVR** 为 0，则计时器将用该值重新加载后保持当前值为 0。该机制可用于独立于定时器允许位禁用该特征。

3.3.2 嵌套向量中断控制器(NVIC)

3.3.2.1 NVIC 主要特征

- ◆ 32 个可屏蔽中断通道 (不包括十六个 Cortex®-M0 的中断线)
- ◆ 可编程优先级 (使用了 2 位中断优先级)
- ◆ 低延迟异常和中断处理
- ◆ 电源管理控制
- ◆ 系统控制寄存器的实现

NVIC 与处理器内核接口紧密配合, 可以实现低延迟的中断处理和和高效地处理晚到的中断。包括内核异常在内的所有中断都由 NVIC 管理。

3.3.2.2 SysTick 校准值寄存器

SysTick 校准值被设置为 5000, 提供了 10 ms 的基准时间, SysTick 时钟被设置为 500 kHz (默认 $f_{HCLK}/8 = 4 \text{ MHz}/8$)。

3.3.2.3 中断和异常向量

位置	优先次序	优先级类型	名称	描述	地址
	-	-		保留	0x0000 0000
	-3	固定的	Reset	复位	0x0000 0004
	-2	固定的	NMI_Handler	不可屏蔽中断	0x0000 0008
	-1	固定的	HardFault_Handler	所有类型的故障	0x0000 000C
	-	-		保留	
	3	可设置的	SVC_Handler	通过 SWI 指令调用的系统服务	0x0000 002C
	-	-		保留	
	5	可设置的	PendSV_Handler	可挂起的系统服务	0x0000 0038
	6	可设置的	SysTick_Handler	系统嘀嗒(SysTick)定时器	0x0000 003C
0	7	可设置的	WWDT	WWDT 全局中断	0x0000 0040
1	8	可设置的	PVD	PVD 通过 EXTI 线 20 检测中断	0x0000 0044
2	9	可设置的	RTC	实时时钟(RTC)全局中断	0x0000 0048
3	10	可设置的	Low Power Wakeup	PVD 通过 EXTI 线 21 检测中断	0x0000 004C
4	11	可设置的	-	保留	0x0000 0050
5	12	可设置的	EXTI[1:0]	EXTI 线 0 至 1 中断	0x0000 0054
6	13	可设置的	EXTI[3:2]	EXTI 线 2 至 3 中断	0x0000 0058
7	14	可设置的	EXTI[15:4]	EXTI 线 4 至 15 中断	0x0000 005C
8	15	可设置的			0x0000 0060
9	16	可设置的	DMA1_CH[0]	DMA1 通道 1 全局中断	0x0000 0064
10	17	可设置的	DMA1_CH[2:1]	DMA1 通道 2 和 3 中断	0x0000 0068
11	18	可设置的	DMA1_CH[5:3]	DMA1 通道 4 至 6 中断	0x0000 006C
12	19	可设置的	ADC_CMP	ADC 和比较器通过 EXTI 线 16 至 19	0x0000 0070
13	20	可设置的	AD16C4T1	AD16C4T1 全局中断	0x0000 0074
14	21	可设置的	BS16T1	BS16T1 全局中断	0x0000 0078
15	22	可设置的	GP32C4T1	GP32C4T1 全局中断	0x0000 007C
16	23	可设置的	GP16C4T1	GP16C4T1 全局中断	0x0000 0080
17	24	可设置的	GP16C4T2	GP16C4T2 全局中断	0x0000 0084
18	25	可设置的	GP16C4T3	GP16C4T3 全局中断	0x0000 0088
19	26	可设置的	GP16C2T1	GP16C2T1 全局中断	0x0000 008C
20	27	可设置的	GP16C2T2	GP16C2T2 全局中断	0x0000 0090
21	28	可设置的	GP16C2T3	GP16C2T3 全局中断	0x0000 0094
22	29	可设置的	GP16C2T4	GP16C2T4 全局中断	0x0000 0098
23	30	可设置的	I2C1	I2C1 全局中断	0x0000 009C
24	31	可设置的	I2C2	I2C2 全局中断	0x0000 00A0
25	32	可设置的	SPI1	SPI1 全局中断	0x0000 00A4
26	33	可设置的	SPI2	SPI2 全局中断	0x0000 00A8
27	34	可设置的	UART1	UART1 全局中断	0x0000 00AC
28	35	可设置的	UART2	UART2 全局中断	0x0000 00B0
29	36	可设置的	UART3	UART3 中断	0x0000 00B4
30	37	可设置的	SUART1_SUART2	SUART1 和 SUART2 全局中断	0x0000 00B8
31	38	可设置的	-	保留	0x0000 00BC

3.3.3 CPU 系统控制

Cortex™-M0 的状态和操作模式控制由 CPU 系统控制寄存器管理，包括 CPUID 在内，可以通过这些系统控制寄存器来控制 Cortex™-M0 中断优先级和 Cortex™-M0 电源管理。

3.4 特殊功能寄存器

3.4.1 寄存器列表

SYST 寄存器列表			
名称	偏移地址	类型	描述
SYST_CSR	0010H	R/W	SysTick 控制和状态寄存器
SYST_RVR	0014H	R/W	SysTick 重载值寄存器
SYST_CVR	0018H	R/W	SysTick 当前值寄存器

NVIC 寄存器列表			
名称	偏移地址	类型	描述
NVIC_ISER	000H	R/W	NVIC IRQ 设置使能控制寄存器
NVIC_ICER	080H	R/W	NVIC IRQ 清除使能控制寄存器
NVIC_ISPR	100H	R/W	NVIC IRQ 设置挂起的控制寄存器
NVIC_ICPR	180H	R/W	NVIC IRQ 清除挂起的控制寄存器
NVIC_IPR0	300H	R/W	NVIC IRQ0 - IRQ3 优先级控制寄存器
NVIC_IPR1	304H	R/W	NVIC IRQ4 - IRQ7 优先级控制寄存器
NVIC_IPR2	308H	R/W	NVIC IRQ8 - IRQ11 优先级控制寄存器
NVIC_IPR3	30CH	R/W	NVIC IRQ12 - IRQ15 优先级控制寄存器
NVIC_IPR4	310H	R/W	NVIC IRQ16 - IRQ19 优先级控制寄存器
NVIC_IPR5	314H	R/W	NVIC IRQ20 - IRQ23 优先级控制寄存器
NVIC_IPR6	318H	R/W	NVIC IRQ24 - IRQ27 优先级控制寄存器
NVIC_IPR7	31CH	R/W	NVIC IRQ28 - IRQ31 优先级控制寄存器

SYS 寄存器列表			
名称	偏移地址	类型	描述
SYS_CPUID	000H	R/W	CPU ID 寄存器
SYS_ICSR	004H	R/W	中断控制与状态寄存器
SYS_AIRCR	00CH	R/W	应用中断和复位控制寄存器
SYS_SCR	010H	R/W	系统控制寄存器
SYS_SHPR2	01CH	R/W	系统处理程序优先级寄存器 2
SYS_SHPR3	020H	R/W	系统处理程序优先级寄存器 3

3.4.2 寄存器描述

3.4.2.1 SysTick 控制和状态寄存器(SYST_CSR)

SysTick 控制和状态寄存器 (SYST_CSR)																																
偏移地址: 10H																																
复位值: 00000000_00000000_00000000_00000000B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
															COUNT																	
																														CLKSRC	TICKIE	ENABLE

—	Bits 31-17	—	—
COUNT	Bit 16	R/W	计数标示 0: 自上次读取该位以来, SysTick 计时器没有计算到 0。 1: 自上次读取该位以来, SysTick 计时器已计算为 0。 COUNT 在读取或写入当前值寄存器时被清除
—	Bits 15-3	—	—
CLKSRC	Bit 2	R/W	System Tick 时钟源选择 0: 时钟源是 (任意的) 外部参考时钟。 1: 核心时钟用于 SysTick.
TICKIE	Bit 1	R/W	System Tick 中断使能 0: 倒数计时到 0 时产生 SysTick 异常请求. 软件可以使用 COUNTFLAG 来确定是否发生了 0 的计数。 1: 倒数计时到 0 时不会产生 SysTick 异常请求。 通过写入软件清除 SysTick 当前值寄存器将不会产生 SysTick 异常请求
ENABLE	Bit 0	R/W	System 计数器使能 0: 计数器已禁用 1: 计数器以多种方式执行

3.4.2.2 SysTick 重载值寄存器 (SYST_RVR)

SysTick 重载值寄存器 (SYST_RVR)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RELOAD <23:0>																							

—	Bits 31-24	—	—
RELOAD	Bits 23-0	R/W	重载值 当计数器达到 0 时，加载到 SysTick 当前值寄存器 (SYST_CVR) 寄存器中。

3.4.2.3 SysTick 当前值寄存器 (SYST_CVR)

SysTick 当前值寄存器 (SYST_CVR)																															
偏移地址: 18 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CURRENT <23:0>																							

—	Bits 31-24	—	—
CURRENT	Bits 23-0	R/W	System Tick 当前计数值 当前计数器值，是计数器在采样时的值。计数器不提供读修改写保护。寄存器是写清除的。任何值的软件写入都会将寄存器清除为 0。

3.4.2.4 NVIC IRQ 设置使能控制寄存器 (NVIC_ISER)

NVIC IRQ 设置使能控制寄存器 (NVIC_ISER)																															
偏移地址: 00H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETENA<31:0>																															

SETENA	Bits 31-0	R/W	<p>中断使能 启用一个或多个中断。每个位代表从 IRQ0 到 IRQ31 中断号码 0: 在读取时, 指示中断被禁用; 在写入时, 没有影响 1: 在读取时, 指示中断已启用; 在写入时, 启用中断</p>
--------	-----------	-----	--

3.4.2.5 NVIC IRQ 清除使能控制寄存器 (NVIC_ICER)

NVIC IRQ 清除使能控制寄存器 (NVIC_ICER)																															
偏移地址: 80H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRENA<31:0>																															

CLRENA	Bits 31-0	R/W	<p>中断禁用 禁用一个或多个中断。每个位代表从 IRQ0 到 IRQ31 中断号码 0: 在读取时, 指示中断被禁用; 在写入时, 没有影响 1: 在读取时, 指示中断已启用; 在写入时, 禁用中断</p>
--------	-----------	-----	--

3.4.2.6 NVIC IRQ 设置挂起控制寄存器 (NVIC_ISPR)

NVIC IRQ 设置挂起控制寄存器 (NVIC_ISPR)																															
偏移地址: 100 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETPEND<31:0>																															

SETPEND	Bits 31-0	R/W	<p>设置中断挂起 禁用一个或多个中断。每个位代表从 IRQ0 到 IRQ31 中断号码</p> <p>0: 在读取时, 指示中断不被停止; 在写入时, 没有影响。</p> <p>1: 在读取时, 指示中断被停止; 在写入时, 相应的中断被设置为停止, 即使它被禁用。</p>
---------	-----------	-----	---

3.4.2.7 NVIC IRQ 清除挂起控制寄存器 (NVIC_ICPR)

NVIC IRQ 清除挂起控制寄存器 (NVIC_ICPR)																															
偏移地址: 180 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRPEND<31:0>																															

CLRPEND	Bits 31-0	R/W	<p>清除中断挂起 禁用一个或多个中断。每个位代表从 IRQ0 到 IRQ31 中断号码</p> <p>0: 在读取时, 指示中断不被停止; 在写入时, 没有影响。</p> <p>1: 在读取时, 指示中断被停止; 在写入时, 写入 1 以清除停止状态, 以便相应的中断不再停止。</p>
---------	-----------	-----	---

3.4.2.8 NVIC IRQ0 - IRQ3 优先级控制寄存器 (NVIC_IPR0)

NVIC IRQ0 - IRQ3 优先级控制寄存器 (NVIC_IPR0)																																																																															
偏移地址: 300 _H																																																																															
复位值: 00000000_00000000_00000000_00000000 _b																																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																
PRI3<1:0>																				PRI2<1:0>																				PRI1<1:0>																				PRI0<1:0>																			

PRI3	Bits 31-30	R/W	IRQ3 优先级
—	Bits 29-24	—	—
PRI2	Bits 23-22	R/W	IRQ2 优先级
—	Bits 21-16	—	—
PRI1	Bits 15-14	R/W	IRQ1 优先级
—	Bits 13-8	—	—
PRI0	Bits 7-6	R/W	IRQ0 优先级
—	Bits 5-0	—	—

3.4.2.9 NVIC IRQ4 – IRQ7 优先级控制寄存器 (NVIC_IPR1)

NVIC IRQ4 – IRQ7 优先级控制寄存器 (NVIC_IPR1)																																																																															
偏移地址: 304 _H																																																																															
复位值: 00000000_00000000_00000000_00000000 _b																																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																
PRI7<1:0>																				PRI6<1:0>																				PRI5<1:0>																				PRI4<1:0>																			

PRI7	Bits 31-30	R/W	IRQ7 优先级
—	Bits 29-24	—	—
PRI6	Bits 23-22	R/W	IRQ6 优先级
—	Bits 21-16	—	—
PRI5	Bits 15-14	R/W	IRQ5 优先级
—	Bits 13-8	—	—
PRI4	Bits 7-6	R/W	IRQ4 优先级
—	Bits 5-0	—	—

3. 4. 2. 10 NVIC IRQ8 – IRQ11 优先级控制寄存器 (NVIC_IPR2)

NVIC IRQ8 – IRQ11 优先级控制寄存器 (NVIC_IPR2)																																																															
偏移地址: 308 _H																																																															
复位值: 00000000_00000000_00000000_00000000 _b																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
PRI11<1:0 ^Δ																PRI10<1:0 ^Δ																PRI9<1:0 ^Δ																PRI8<1:0 ^Δ															

PRI11	Bits 31-30	R/W	IRQ11 优先级
—	Bits 29-24	—	—
PRI10	Bits 23-22	R/W	IRQ10 优先级
—	Bits 21-16	—	—
PRI9	Bits 15-14	R/W	IRQ9 优先级
—	Bits 13-8	—	—
PRI8	Bits 7-6	R/W	IRQ8 优先级
—	Bits 5-0	—	—

3. 4. 2. 11 NVIC IRQ12 – IRQ15 优先级控制寄存器 (NVIC_IPR3)

NVIC IRQ12 – IRQ15 优先级控制寄存器 (NVIC_IPR3)																																																															
偏移地址: 30C _H																																																															
复位值: 00000000_00000000_00000000_00000000 _b																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
PRI15<1:0 ^Δ																PRI14<1:0 ^Δ																PRI13<1:0 ^Δ																PRI12<1:0 ^Δ															

PRI15	Bits 31-30	R/W	IRQ15 优先级
—	Bits 29-24	—	—
PRI14	Bits 23-22	R/W	IRQ14 优先级
—	Bits 21-16	—	—
PRI13	Bits 15-14	R/W	IRQ13 优先级
—	Bits 13-8	—	—
PRI12	Bits 7-6	R/W	IRQ12 优先级
—	Bits 5-0	—	—

3. 4. 2. 12 NVIC IRQ16 – IRQ19 优先级控制寄存器 (NVIC_IPR4)

NVIC IRQ16 – IRQ19 优先级控制寄存器 (NVIC_IPR4)																																																															
偏移地址: 310 _H																																																															
复位值: 00000000_00000000_00000000_00000000 _b																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
PRI19<1:0>																PRI18<1:0>																PRI7<1:0>																PRI6<1:0>															

PRI19	Bits 31-30	R/W	IRQ19 优先级
—	Bits 29-24	—	—
PRI18	Bits 23-22	R/W	IRQ18 优先级
—	Bits 21-16	—	—
PRI17	Bits 15-14	R/W	IRQ17 优先级
—	Bits 13-8	—	—
PRI16	Bits 7-6	R/W	IRQ16 优先级
—	Bits 5-0	—	—

3. 4. 2. 13 NVIC IRQ20 – IRQ23 优先级控制寄存器 (NVIC_IPR5)

NVIC IRQ20 – IRQ23 优先级控制寄存器 (NVIC_IPR3)																																																															
偏移地址: 314 _H																																																															
复位值: 00000000_00000000_00000000_00000000 _b																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
PRI23<1:0>																PRI22<1:0>																PRI21<1:0>																PRI20<1:0>															

PRI23	Bits 31-30	R/W	IRQ23 优先级
—	Bits 29-24	—	—
PRI22	Bits 23-22	R/W	IRQ22 优先级
—	Bits 21-16	—	—
PRI21	Bits 15-14	R/W	IRQ21 优先级
—	Bits 13-8	—	—
PRI20	Bits 7-6	R/W	IRQ20 优先级
—	Bits 5-0	—	—

3. 4. 2. 14 NVIC IRQ24 – IRQ27 优先级控制寄存器 (NVIC_IPR6)

NVIC IRQ24 – IRQ27 优先级控制寄存器 (NVIC_IPR6)																															
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

偏移地址: 318 _H																																																																															
复位值: 00000000_00000000_00000000_00000000 _b																																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																
PRI27<1:0>																				PRI26<1:0>																				PRI25<1:0>																				PRI24<1:0>																			

PRI27	Bits 31-30	R/W	IRQ27 优先级
—	Bits 29-24	—	—
PRI26	Bits 23-22	R/W	IRQ26 优先级
—	Bits 21-16	—	—
PRI25	Bits 15-14	R/W	IRQ25 优先级
—	Bits 13-8	—	—
PRI24	Bits 7-6	R/W	IRQ24 优先级
—	Bits 5-0	—	—

3.4.2.15 NVIC IRQ28 – IRQ31 优先级控制寄存器 (NVIC_IPR7)

NVIC IRQ20 – IRQ23 优先级控制寄存器(NVIC_IPR3)																																																																															
偏移地址: 31C _H																																																																															
复位值: 00000000_00000000_00000000_00000000 _b																																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																
PRI31<1:0>																				PRI30<1:0>																				PRI29<1:0>																				PRI28<1:0>																			

PRI31	Bits 31-30	R/W	IRQ31 优先级
—	Bits 29-24	—	—
PRI30	Bits 23-22	R/W	IRQ30 优先级
—	Bits 21-16	—	—
PRI29	Bits 15-14	R/W	IRQ29 优先级
—	Bits 13-8	—	—
PRI28	Bits 7-6	R/W	IRQ28 优先级
—	Bits 5-0	—	—

3.4.2.16 CPU ID 寄存器 (SYS_CPUID)

CPU ID 寄存器 (SYS_CPUID)																															
偏移地址: 00 _H																															
复位值: 01000001_00001100_11000010_00000000 _b																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMPC<7:0>												PART<3:0>				PARTNO<11:0>											REV<3:0>				

IMPC	Bits 31-24	R	ARM 分配的实现代码 ARM = 0x41
—	Bits 23-20	—	—
PART	Bits 19-16	R/W	处理器的结构
PARTNO	Bits 15-4	R	处理器的零件号
REV	Bits 3-0	R/W	修订号

3.4.2.17 中断控制与状态寄存器 (SYS_ICSR)

中断控制与状态寄存器 (SYS_ICSR)																																					
偏移地址: 04 _H																																					
复位值: 00000000_00000000_00000000_00000000 _B																																					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
NMISP	—	—	PENDSV	PENDSV	PENDST	PENDSTC	—	ISRPRE	SRPEND	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—						
																VTPEND<5:0>																VTACT<5:0>					

NMISP	Bit 31	R/W	<p>NMI (非可屏蔽中断) 挂起位</p> <p>0: 在读取时, 指示 NMI 异常未挂起。在写入时, 没有影响。</p> <p>1: 在读取时, 指示 NMI 异常挂起。在写入时, 将 NMI 异常状态更改为挂起状态。</p> <p>因为 NMI 是最高优先级的异常, 所以处理器通常在检测到 1 对这个位的写时立即进入 NMI 异常处理程序。进入处理程序然后将此位清除为 0。这意味着, 只有当处理器正在执行 NMI 异常处理程序时, NMI 信号被重新置入, NMI 异常处理程序才读取该位, 返回 1。</p>
—	Bits 30-29	—	—
PENDSV	Bit 28	R/W	<p>PendSV 挂起位</p> <p>0: 在读取时, 指示 PendSV 异常未挂起。在写入时, 没有影响。</p> <p>1: 在读取时, 指示 PendSV 异常挂起。在写入时, 将 PendSV 异常状态更改为挂起状态。</p> <p>只有通过写一个“1”到这个位, 才能将 PendSV 异常状态设置为挂起状态。</p>
PENDSV	Bit 27	W	<p>PendSV 清除挂起位</p> <p>0: 没有影响。</p> <p>1: 从 PendSV 异常中移除挂起状态。</p>
PENDST	Bit 26	R/W	<p>SysTick 异常挂起位</p> <p>0: 在读取时, 指示一个 SysTick 异常未挂起。在写入时, 没有效果。</p> <p>1: 在读取时, 指示一个 SysTick 异常挂起。在写入时, 将 PendSV 异常状态更改为挂起状态。</p>
PENDSTC	Bit 25	W	<p>SysTick 异常清除挂起位</p> <p>0: 没有影响。</p> <p>1: 从 SysTick 异常移除挂起状态。</p>
—	Bit 24	—	—
ISRPRE	Bit 23	R	<p>调试中断处理</p> <p>0: 停止退出不中断</p> <p>1: 在调试停止状态退出时, 挂起异常将会执行。</p>
SRPEND	Bit 22	R	<p>中断挂起标志, 不包括 NMI 和故障</p>

			0: 中断未挂起 1: 中断挂起
—	Bits 21-18	—	—
VTPEND	Bits 17-12	R/W	中断挂起向量数 0: 没有挂起的异常。 其它: 最高优先级挂起启用异常的异常数目。
—	Bits 11-6	—	—
VTACT	Bits 5-0	R/W	活动异常数 此字段包含活动异常数。 0: 线程模式。 其它: 当前活动异常的异常数目。

3.4.2.18 应用中断和复位控制寄存器 (SYS_AIRCR)

应用中断和复位控制寄存器 (SYS_AIRCR)																																																
偏移地址: 0C _H																																																
复位值: 11111010_00000101_00000000_00000000 _B																																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
VTKEY<15:0>																																																
																					</																											

3.4.2.20 系统处理程序优先级寄存器 2 (SYST_SHPR2)

系统处理程序优先级寄存器 2 (SYST_SHPR2)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRISH11<1:0>																															

PRISH11	Bits 31-30	R/W	系统处理程序 11 优先级 - SVCall “0”表示最高优先级,“3”表示最低优先级。
—	Bits 29-0	—	—

3.4.2.21 系统处理程序优先级寄存器 3 (SYST_SHPR3)

系统处理程序优先级寄存器 3 (SYST_SHPR3)																																
偏移地址: 20 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PRISH15<1:0>								PRISH14<1:0>																								

PRISH15	Bits 31-30	R/W	系统处理程序 15 优先级 - SVCall “0”表示最高优先级,“3”表示最低优先级。
—	Bits 29-24	—	—
PRISH14	Bits 23-22	R/W	系统处理程序 14 优先级 - SVCall “0”表示最高优先级,“3”表示最低优先级。
—	Bits 21-0	—	—

第4章 系统配置控制器 (SYSCFG)

4.1 概述

系统配置控制器主要针对系统级的应用与电源管理进行说明，用户可透过阅读此章节了解如何配置电源电压侦测，同时也可了解如何开启低功耗模式来降低芯片的功率消耗，让使用外部电源的用户能够延长装置使用的时间。

4.2 特性

- ◆ 支持可编程掉电压侦测触发中断
- ◆ 支持 5 种低功耗模式
- ◆ 支持 4 组备份寄存器
- ◆ 支持用户配置系统重映射
- ◆ 支持红外线(IR)接口输出配置
- ◆ 支持除错模式(Debug Mode)下让独立看门狗(IWDT)与窗口看门狗(WWDT)暂停止计数
- ◆ 支持除错模式(Debug Mode)下让计数器(Timer)暂停止计数
- ◆ 支持发生电源掉电、发生 HOSC 系统时钟故障或是 CPU 发生 Hard Fault 故障时让计数器(Timer)停止计数

4.3 功能描述

4.3.1 电源

此芯片的电源为外部电压源提供，支持 2.2 伏至 5.5 伏的电压操作范围。系统内部依据工作电压的不同共可分为 2 个电压域，分为使用外部供电的系统电源域(VDD5 Domain)与备份域 (Backup Domain)以及使用 1.5 伏电压调节器产生 1.5 伏电压的 1.5 伏电压域(1.5V Internal)。

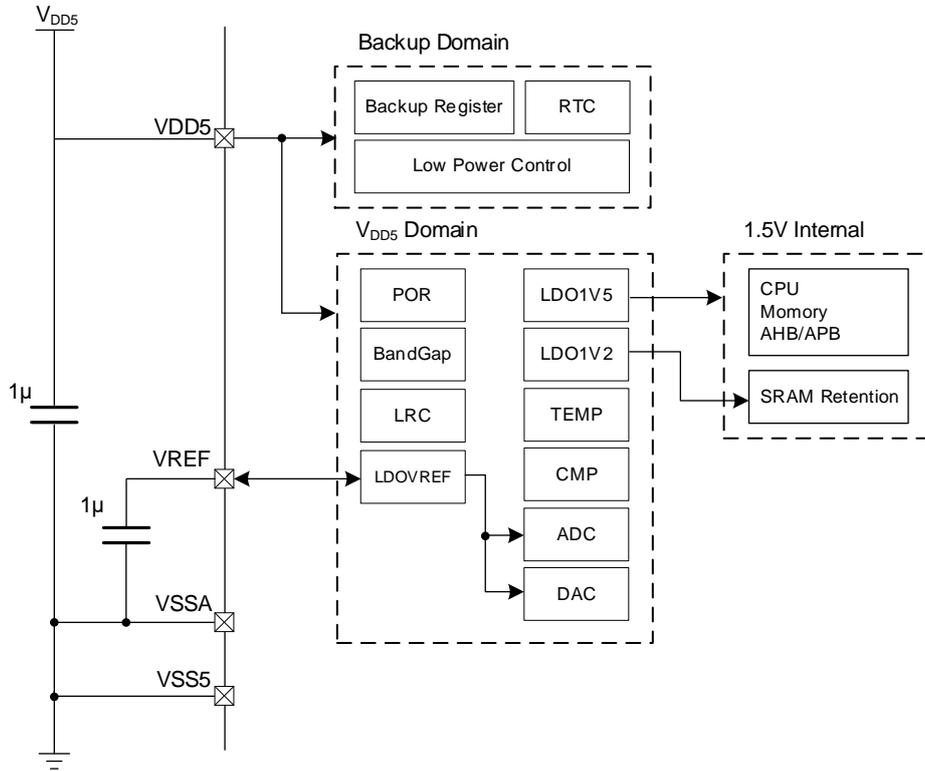


图 4-1 电源域

4.3.1.1 备份域 (Backup Domain)

备份域主要由操作在 VDD5 电源域的备份寄存器、低功耗控制单元以及实时时钟(RTC)组成，此区的电源仅受 VDD5 电压影响，因此当系统进入低功耗模式时仍会保留此区的电源。备份寄存器由 4 组 32 位的寄存器组成，这 4 组寄存器分别为 **SYSCFG_BKREG0(0x50)**、**SYSCFG_BKREG1(0x54)**、**SYSCFG_BKREG2(0x58)**与 **SYSCFG_BKREG3(0x5C)**，当系统进入低功耗模式前可将需要备份的讯息存放在备份寄存器内，当系统从低功耗模式唤醒后再从备份寄存器中读出暂存讯息。用户在使用备份寄存器时需注意 **SYSCFG_BKREG3(0x5C)**仅能存放 30 位的数据。低功耗控制单元共有 3 组寄存器，分别为 **SYSCFG_PWCON(0x40)**、**SYSCFG_WKCON(0x48)**与 **SYSCFG_WKSTAT (0x4C)**，这 3 组寄存器可控制系统电源侦测与低功耗模式。

备份域的寄存器仅支持以 32 位为单位进行读写操作，由于读写备份域的寄存器时牵涉到跨电源域的读写操作，因此在读写备份寄存器时会有一段等待时间，在完成读写备份域的寄存器前会让 CPU 进行等待，用户在操作此电源域的寄存器时须特别注意。

4.3.1.2 电压调节器 (Voltage Regulator)

此芯片共有 3 个电压调节器，说明如下。

- ◆ **1.5 伏电压调节器**：可提供稳定的 1.5 伏电压，确保 1.5 伏电压域内的 HRC 模块、CPU、内存、闪存、AHB 外设以及 APB 外设能够稳定运作。由于电压调节器使用的是 VDD5 电源，因此芯片刚上电时为确保电源稳定会让电压调节器为关闭状态，系统会在开机流程内等待电源稳定后才开启电压调节器产生 1.5 伏电压，避免影响到 1.5 伏电压域内的逻辑。由于电压调节器主要提供电源给 1.5 伏电压域，因此当用户需要让系统长时间待机时，可让系统进入低功耗模式的待机模式 0、待机模式 1 与停止运转模式。在进入这 3 种低功耗模式时皆会关闭 1.5 伏电压调节器，此时 1.5 伏电压域内的逻辑皆会被关闭，藉此延长电池使用时间。
- ◆ **1.2 伏电压调节器**：在待机模式 0 下被开启后可提供稳定的 1.2 伏电压给备份 SRAM，让备份 SRAM 能够继续维持数据不被清除，而在一般模式下此电压调节器会关闭并开启旁路模式输出 1.5 伏电压。当用户无保留 SRAM 内数据的需求时，可让系统进入低功耗模式的待机模式 1 与停止运转模式。
- ◆ **VREF 电压调节器**：此电压调节器可提供 ADC 模块与 DAC 模块参考电压。用户可配置 SYSCFG_CFG(0x3C)内的 VRLS(位 0 至位 2)来调整 VREF 电压调节器的输出电压值。配置完电压准位后，用户需再配置 SYSCFG_CFG(0x3C)内的 VREFEN(位 3)为 1 开启电压调节器。若用户需要从 VREF 管脚输入电压时，需先确认此电压调节器未被开启。若芯片封装没有 VREF 管脚时则 VREF 电压固定为 VDD5，且无法透过 VREF 电压调节器调整电压值。

4.3.1.3 电源侦测

POR 模块负责监控系统电源(VDD5)，并提供下列 2 种功能。

- ◆ **上电/掉电复位(Power On/Down Reset)**
当系统电源从 0 伏上升至超过 V_{POR} 时 POR 模块会再等待约 2.1 毫秒(2.1ms)后会拉高 POR 旗标，此时系统便会离开复位模式并开始执行开机流程。当系统电源从 VDD5 降至低于 V_{PDR} 时会拉低 POR 旗标，此时系统会进入复位状态。POR 检测上电/掉电复位发生时的电压为固定值，并不支持用户自行配置，此电压值仅受温度与芯片制程影响。

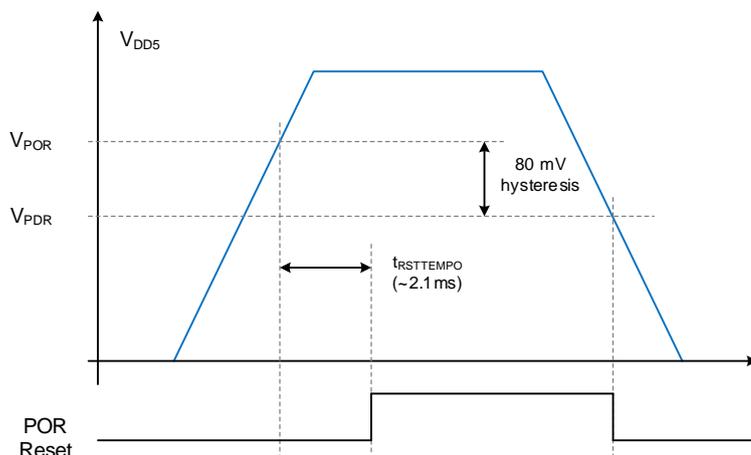


图 4-2 POR/PDR 复位

- ◆ 可编程电压检测器 (Programmable Voltage Detector)
 用户可配置 **SYSCFG_PWCON(0x40)**内的 PVLS(位 0 至位 3)决定要检测的电压值，再配置 **PVDEN** (位 4)为 1 开启电压检测此能。当系统电源 **VDD5** 降至低于用户配置的电压值时便会发出低电压旗标通知位于 **APB** 内的 **EXTI** 逻辑，若用户有开启 **EXTI** 的中断功能则会在低电压旗标产生时触发 **CPU** 中断，让用户能够提前在在掉电复位发生前将重要信息进行保留。由于 **EXTI** 逻辑支持用户检测低电压旗标的正缘发生点与负缘发生点，因此当低电压旗标因电源上升而被拉低时可再一次产生中断告知用户。此外电压检测旗标也可当作外部触发讯号唤醒处在睡眠模式、停止模式、待机模式 0 或待机模式 1 下的系统。有关 **EXTI** 逻辑的说明与配方式请参阅第 11 章外部中断(**EXTI**)内的描述。

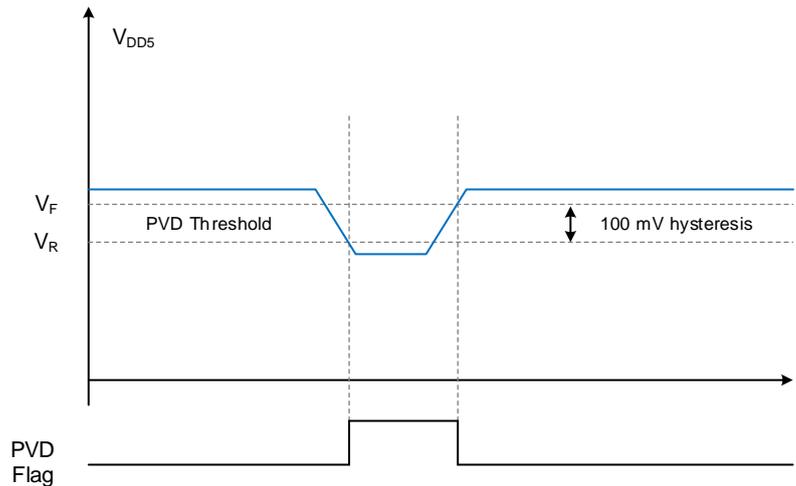


图 4-3 PVD 旗标

4.3.2 低功耗模式 (Low Power Mode)

当用户需要让程序长时间进行待机或是等待外部讯号触发时，可让系统进入低功耗模式进行等待，藉此降低系统功率消耗并延长电池使用时间。若用户希望在系统持续运行的情况下降低功率消耗，可藉由降低系统时钟频率以及关闭不使用的的外设时钟来达成。

模式	进入条件	唤醒	1.5 伏域 时钟	VDD5 域 时钟	LDO/ BandGap	唤醒 时间	电流 消耗		
睡眠模式 (Sleep)	SLEEP + WFI or WFE	外设中断 或 唤醒事件	CPU 时钟关闭 外设时钟关闭	LRC 开启	LDO1V5 开启 LDO1V2 关闭 BandGap 开启	<4.3us	<1mA		
停止模式 (Stop)									
待机模式 0 (Standby0)	SLEEPDEEP + WFI or WFE	唤醒事件	1.5V 时钟关闭	LRC 关闭 SRAM 保留	LDO1V5 关闭 LDO1V2 开启 BandGap 开启	~1.2ms	<750uA		
待机模式 1 (Standby1)				LRC 关闭	LDO1V5 关闭 LDO1V2 关闭 BandGap 开启			~1.2ms	<600nA
停止运转模式 (Shutdown)					LDO1V5 关闭 LDO1V2 关闭 BandGap 关闭				

表 4-1 低功耗模式

4.3.2.1 低速执行(Low Speed)

芯片支持用户将系统时钟切换到低速的 LRC 时钟，当程序需要长时间等待触发讯号时，可暂时将系统时钟切换至 LRC 时钟，并于系统时钟切换完毕以后再将没有用到的时钟关闭，同时也可暂时将不会使用到的外设时钟关闭，藉此降低系统的功率消耗。当系统收到触发讯号以后，重新进行系统时钟配置，恢复系统正常运行。

4.3.2.2 睡眠模式(Sleep)

进入睡眠模式后会暂时关闭 CPU 时钟与外设时钟但并不会关闭 1.5 伏域的电，因此外设的配置与 SRAM 内的数据依然会保存。系统进入睡眠模式时可额外配置 RCU 逻辑内位于 0x48 至 0x50 的睡眠模式外设时钟使能寄存器，让被开启的外设在系统进入睡眠模式后依然能够继续运行。

◆ 使用 CPU 睡眠模式(Sleep Mode)

当用户配置 **SYSCFG_WKCON**(0x48)内的 LPLS(位 28 至位 30)为 0，并配至 CPU 系统控制寄存器(System Control Register)内的 SLEEPDEEP(位 2)为 0 后即可搭配 WFI 让 CPU 进入睡眠模式。在此模式下默认关闭 CPU 时钟与外设时钟，用户若有需要让外设持续运行的需求时，可配置 RCU 逻辑内的 **RCU_AHBSL**(0x28)、**RCU_APB1SL**(0x2C)与 **RCU_APB2SL**(0x30)寄存器来开启外设时钟。

◆ 从睡眠模式唤醒

当系统进入睡眠模式时，可选择使用外设中断、唤醒脚位(Wakeup Pin)、RTC 睡眠计数器、比较器 0 唤醒事件、电压检测(PVD)或是拉低外部 NRST 脚位来唤醒 CPU。

- ◇ 外设中断唤醒：使用外设中断唤醒时需在系统进入睡眠模式前配置 RCU 逻辑内位于 0x48 至 0x50 的睡眠模式外设时钟使能寄存器开启外设时钟，让外设睡眠模式也能继续运行。使用中断唤醒 CPU 时并不会产生低功耗复位旗标，同时 CPU 会在唤醒后继续往下执行用户代码。
- ◇ 唤醒脚位唤醒：支持用户使用 8 个唤醒脚位搭配 EXTI 中断来唤醒 CPU，用户需配置 **SYSCFG_WKCON**(0x48)内 WKEN(位 0 至位 11)的位 0 至位 7 为 1 使能唤醒脚位，并配至 **WKEG**(位 16 至位 27)决定要在唤醒讯号的正缘或是负缘发生时产生唤醒旗标来触发 EXTI 逻辑产生中断，有关 EXTI 的配置方式请参阅第 11 章的描述。使用唤醒脚位产生中断唤醒 CPU 时会产生复位旗标，但并不会产生系统复位，CPU 会于唤醒后继续往下执行用户代码。
- ◇ RTC 睡眠计数器唤醒：支持用户配置 RTC 的睡眠计数器于计数完毕时发出唤醒讯号，并触发 EXTI 中断来唤醒 CPU，有关 RTC 的配置方式请参阅第 22 章的描述。若用户选择当作唤醒事件时，则需配置 **SYSCFG_WKCON** (0x48)内 WKEN(位 0 至位 11)的位 8 为 1 使能唤醒功能。
- ◇ 比较器 0 唤醒：支持用户配置比较器 0 的比较结果当作唤醒事件，当条件满足时可触发 EXTI 中断来唤醒 CPU，有关比较器的配置方式请参阅第 13 章的描述。若用户选择当作唤醒事件时，则需配置 **SYSCFG_WKCON** (0x48)内 WKEN(位 0 至位 11)的位 9 为 1 使能唤醒功能。

- ◇ 电压检测(PVD)唤醒：当用户开启电压检测功能时，会在系统电压降至用户选定的电压时产生 PVD 旗标，PVD 旗标发生时可触发中断唤醒 CPU 或是当作唤醒事件唤醒 CPU。当用户决定以中断的方式唤醒 CPU 时，需于系统进入低功耗模式前配置 EXTI 逻辑。若用户选择当作唤醒事件时，则需配置 **SYSCFG_WKCON** (0x48)内 WKEN(位 0 至位 11)的位 10 为 1 使能唤醒功能。
- ◇ 外部 NRST 脚位唤醒：当用户拉低外部 NRST 脚位时会重置系统并产生 NRST 复位旗标，用户可于系统重启后检查 RCU 逻辑内的 NRST 复位旗标。

当 CPU 被唤醒后皆会继续往下执行用户代码，CPU 可于被唤醒后检查 **SYSCFG_WKSTAT**(0x4C)内的 FG(位 0 至位 11)判断是由哪一个唤醒事件唤醒，若 CPU 为唤醒事件唤醒时，需配置 **SYSCFG_WKCON**(0x48)内的 WKCLR (位 31)为 1 来清除唤醒旗标，若未清除旗标则系统无法再次进入低功耗模式。

4.3.2.3 停止模式(Stop)

停止模式的功能接近睡眠模式，差别在于进入停止模式以后会再额外关闭系统配置逻辑与内存所使用的时钟，而从此模式下唤醒所需要的时间会略长于睡眠模式。

◆ 使用 CPU 深度睡眠模式(Deepsleep Mode)

当用户配置 **SYSCFG_WKCON**(0x48)内的 LPLS(位 28 至位 30)为 1，并配至 CPU 内系统控制寄存器(System Control Register)内的 SLEEPDEEP(位 2)为 1 后即可搭配 WFI 让 CPU 进入深度睡眠模式。在此模式下除了关闭 CPU 时钟与外设时钟外，也会关闭 RCC 逻辑、系统配置逻辑与内存所使用的时钟。用户若有需要让外设持续运行的需求时，可配置 RCU 逻辑内的 **RCU_AHBDSL**(0x34)、**RCU_APB1DSL**(0x38)与 **RCU_APB2DSL**(0x3C)寄存器来开启外设时钟。

◆ 从睡眠模式唤醒

从停止模式唤醒的方式与从睡眠模式唤醒的方式相同，请参阅睡眠模式内的描述。

4.3.2.4 待机模式 0(Standby0)

进入待机模式 0 后系统会关闭 1.5 伏电压调节器同时开启 1.2 伏电压调节器，此时 1.5 伏域内的逻辑电路皆会掉电，但因 1.2 伏电压调节被开启故仍会保留 SRAM 内特定位置的信息（保留位置为 0x20001000 ~ 0x20001FFF 区间的信息）。由于 1.5 伏域电源已被关闭，因此 GPIO 的相关配置都会被重置，若用户希望在待机模式 0 下保留唤醒脚位的 I/O 配置(如 I/O 上拉或是下拉)时，可藉由配置 **SYSCFG_PWCON**(0x40)内的 IOLOCK(位 24)为 1 让唤醒脚位的 I/O 配置能够继续保留，用户在系统从低功耗模式唤醒后需将 IOLOCK(位 24)清为 0 才可重新调整唤醒脚位的 I/O 配置。由于 **SYSCFG_PWCON**(0x40)寄存器仅支持 32 位的读写操作，用户在修改此寄存器时需避免修改到寄存器内其他栏位的数值。当系统从此低功耗模式唤醒时会重启系统并重新开始执行闪存内的用户程序，建议用户于程序内加入检查 **SYSCFG_WKSTAT**(0x4C)内的 FG(位 0 至位 11)判断是由哪一个唤醒事件唤醒，并配置 **SYSCFG_WKCON** (0x48)内的 WKCLR (位 31)为 1 来清除唤醒旗标，完成低功耗流程。用户须注意的是若未清除旗标则系统无法再次进入低功耗模式。

◆ 仅支持使用 CPU 深度睡眠模式(Deepsleep Mode)

待机模式 0 仅支持用户配置 CPU 进入深度睡眠模式，同时需配置 **SYSCFG_WKCON**(0x48)内的 LPLS(位 28 至位 30)为 2。由于在进入待机模式 0 后 1.5

伏电压调节器已被关闭，因此不支持用户在此模式下开启外设时钟或使用外设中断唤醒。

◆ 从睡眠模式唤醒

此模式仅支持用户使用唤醒事件唤醒系统，用户须于进入待机模式 0 前配置 **SYSCFG_WKCON**(0x48)内的 **WKEN**(位 0 至位 11)与 **WKEG**(位 16 至位 27)决定要使用哪些唤醒事件。当系统被唤醒后，可检查 **SYSCFG_WKSTAT**(0x4C)内的 **FG**(位 0 至位 11)判断是由哪一个唤醒事件唤醒，并配置 **SYSCFG_WKCON** (0x48)内的 **WKCLR** (位 31)为 1 来清除唤醒旗标。

4.3.2.5 待机模式 1(Standby1)

进入待机模式 1 后系统会关闭 1.5 伏电压调节器以及 1.2 伏电压调节器，此时 1.5 伏域内的逻辑电路皆会掉电且存放于 **SRAM** 内的数据也会被清除。由于 1.5 伏域电源已被关闭，因此 **GPIO** 的相关配置都会被重置，若用户希望在待机模式 1 下保留唤醒脚位的 **I/O** 配置(如 **I/O** 上拉或是下拉)时，可藉由配置 **SYSCFG_PWCON**(0x40)内的 **IOLOCK**(位 24)为 1 让唤醒脚位的 **I/O** 配置能够继续保留，用户在系统从低功耗模式唤醒后需将 **IOLOCK**(位 24)清为 0 才可重新调整唤醒脚位的 **I/O** 配置。由于 **SYSCFG_PWCON**(0x40)寄存器仅支持 32 位的读写操作，用户在修改此寄存器时需避免修改到寄存器内其他栏位的数值。当系统从此低功耗模式唤醒时会重启系统并重新开始执行闪存内的用户程序，建议用户于程序内加入检查 **SYSCFG_WKSTAT**(0x4C)内的唤醒事件旗标来检查唤醒事件，并配置 **SYSCFG_WKCON** (0x48)内的 **WKCLR** (位 31)为 1 来清除唤醒旗标，藉此完成低功耗流程。用户须注意的是若未清除旗标则系统无法再次进入低功耗模式。

◆ 仅支持使用 CPU 深度睡眠模式(Deepsleep Mode)

待机模式 1 仅支持用户配置 CPU 进入深度睡眠模式，同时需配置 **SYSCFG_WKCON**(0x48)内的 **LPLS**(位 28 至位 30)为 3。由于在进入待机模式 1 后 1.5 伏电压调节器已被关闭，因此不支持用户在此模式下开启外设时钟或使用外设中断唤醒，同时存放于 **SRAM** 内的数据也会被清除。

◆ 从睡眠模式唤醒

此模式仅支持用户使用唤醒事件唤醒系统，用户须于进入待机模式 1 前配置 **SYSCFG_WKCON**(0x48)内的 **WKEN**(位 0 至位 11)与 **WKEG**(位 16 至位 27)决定要使用哪些唤醒事件。当系统被唤醒后，可检查 **SYSCFG_WKSTAT**(0x4C)内的 **FG**(位 0 至位 11)判断是由哪一个唤醒事件唤醒，并配置 **SYSCFG_WKCON** (0x48)内的 **WKCLR** (位 31)为 1 来清除唤醒旗标。

4.3.2.6 停止运转模式(Shutdown)

进入此模式后系统会关闭 1.5 伏电压调节器、1.2 伏电压调节器与 **BandGap**，故此低功耗模式能提供最佳的省电效果。由于 1.5 伏域电源已被关闭，因此 **GPIO** 的相关配置都会被重置，若用户希望在停止运转模式下保留唤醒脚位的 **I/O** 配置(如 **I/O** 上拉或是下拉)时，可藉由配置 **SYSCFG_PWCON**(0x40)内的 **IOLOCK**(位 24)为 1 让唤醒脚位的 **I/O** 配置能够继续保留，用户在系统从低功耗模式唤醒后需将 **IOLOCK**(位 24)清为 0 才可重新调整唤醒脚位的 **I/O** 配置。由于 **SYSCFG_PWCON**(0x40)寄存器仅支持 32 位的读写操作，用户在修改此寄存器时需避免修改到寄存器内其他栏位的数值。在此低功耗模式下因 **BandGap** 逻辑被关闭，所以无法再使用

PVD 检测事件来唤醒系统，同时从此低功耗模式唤醒所需的时间也较其他低功耗模式长。当系统被唤醒后会重新执行闪存程序区内的程序。

◆ 仅支持使用 CPU 深度睡眠模式(Deepsleep Mode)

停止运转模式仅支持用户配置 CPU 进入深度睡眠模式，同时需配置 **SYSCFG_WKCON(0x48)**内的 LPLS(位 28 至位 30)为 4。由于在进入停止运转模式后 1.5 伏电压调节器已被关闭，因此不支持用户在此模式下开启外设时钟或使用外设中断唤醒，同时存放于 SRAM 内的数据也会被清除。

◆ 从睡眠模式唤醒

此模式仅支持用户使用唤醒脚位来唤醒系统，不再支持用户使用 PVD 事件来唤醒系统，用户须于进入停止运转模式前配置 **SYSCFG_WKCON(0x48)**内的 WKEN(位 0 至位 11)与 WKEG(位 16 至位 27)决定要使用那些唤醒脚位，用户须于唤醒后配置 **SYSCFG_WKCON(0x48)**内的 WKCLR(位 31)为 1 来清除唤醒旗标，藉此完成低功耗流程，若未清除旗标则系统无法再次进入低功耗模式。

4.3.3 系统重映射

此芯片的存储区共分为 3 个区块，分别为闪存(Flash)、系统闪存以及嵌入式只读存储器(Maskrom)。闪存可供用户存放代码，而系统闪存主要存放芯片下载程序(Bootloader)，此程序会在芯片出厂前更新至系统闪存内，其主要功能为协助用户将代码下载至闪存内。嵌入式只读存储器(Maskrom)内主要存放库可供用户调用，内容同样在芯片出厂前更新至 Maskrom 内。当系统开机完毕后默认会从系统闪存开始执行芯片下载程序，若用户未与芯片下载程序进行同步沟通，则下载程序在等待一段时间后会映射至闪存开始执行用户代码。

4.3.3.1 硬件重映射

硬件重映射的主要功能为当用户不再需要使用芯片下载程序(Bootloader)时，可于系统开机完毕后直接从闪存开始执行用户的代码，达到快速开机的效果。硬件映射可由用户决定是否开启，若决定开启此功能时用户需自行配置位于闪存讯息区页 7 内的用户配置字。有关用户配置字的说明请参阅闪存控制器章节内的描述。

4.3.3.2 软件重映射

软件重映射支持用户重新选择要执行闪存的程序或是执行位于系统闪存内的芯片下载程序，如当系统运行在闪存时可藉由软件重映射的功能重新映射回系统闪存执行芯片下载程序。当用户在使用软件重映射的功能时，需配置 **SYSCFG_REMAP(0x00)**内的 MEMMOD(位 2 至位 3)来决定后续程序要映射至哪一个存储区继续执行，并配置 REMAP(位 0)为 1 开启重映射流程。开启重映射流程以后系统并不会立即进行映射，用户需使用 `NVIC_SystemReset()`复位函数重置 CPU 后，系统才会真正完成重映射程序。

4.3.3.3 闪存重映射

当用户使用软件重映射将系统映射至闪存执行时，可进一步设定闪存的内部映射。内部映射是以 4 个页(4K Byte)为基准进行映射，用户可在配置 **SYSCFG_REMAP(0x00)**内的 MEMMOD(位 2 至位 3)时再额外配置 EFBASE(位 4 至位 7)来决定要映射至闪存的哪一个 4K Byte 区块开始执行。闪存的内部映射只有在读取闪存的数据时才会进行映射，当用户对闪存进行编程与擦除时并不会受到内部映射影响。此外若用户使用闪存的实体位置来读取数据时，同样不会受到内部映射的影响。

4.3.4 ADC/DAC 电源配置

用户在使用 ADC 模块与 DAC 模块时需配置 **SYSCFG_CFS(0x3C)**内的 **CURRGEN(位 5)**开启 ADC 与 DAC 的电源开关, 未配置 **CURRGEN** 为 1 会导致 ADC 模块与 DAC 模块无法启用。

ADC 与 DAC 支持用户自行配置参考电压, 可分为下列 2 种用法:

- ◆ **VREF** 电压调节器提供参考电压: 用户可配置 **SYSCFG_CFG(0x3C)**内的 **VRLS(位 0 至位 2)**来调整 **VREF** 电压调节器的输出电压值。配置完电压值后需再配置 **SYSCFG_CFG(0x3C)**内的 **VREFEN(位 3)**为 1 开启电压调节器。
- ◆ 外部输入参考电压: 在未开启 **VREF** 电压调节器时, 用户可直接从 **VREF** 管脚输入稳定电压。当用户选择使用此方式提供 ADC 与 DAC 参考电压时, 须确保外部提供的电压源为一个稳定的电压源。

4.3.5 温度传感器

芯片内建温度传感器, 用户可藉由配置 **SYSCFG_CFS(0x3C)**内的 **TEMPEN(位 4)**为 1 来开启温度传感器。当温度传感器开启后, 会输出参考电压供 ADC 进行取样, 用户可藉由取样后的数值推算出当前温度数值。有关 ADC 取样的配置方式请参阅第 12 章模数转换器 (ADC)内的描述。

4.3.6 NRST 复位请求

芯片支持用户配置 **SYSCFG_CFG(0x3C)**内的 **NRSTREQ(位 7)**为 1 来触发 **NRST** 复位, **NRSTREQ** 会在 **NRST** 复位发生的同时被重置回 0, 有关 **NRST** 复位请参阅 **RCU** 章节内的说明。

4.3.7 停止计数配置

独立看门狗与窗口看门狗可在计数完毕时触发系统复位, 为了避免用户在除错模式时因为没有重置看门狗而触发系统复位, 因此芯片支持用户配置 **SYSCFG_CFG(0x3C)**的 **DBGH_EN(位 16 至位 31)**让看门狗可以暂时停止计数。同理, 在除错模式下也可配置 **DBGH_EN(位 16 至位 31)**让计数器(Timer)暂时停止计数, 避免发生计数错误或是让芯片持续送出无法控制的 PWM 讯号。除了除错模式以外, 用户也可配置 **SYSCFG_CFG(0x3C)**内的 **PVD_LCK(位 14)**、**CSS_LCK(位 13)**与 **CPU_LCK(位 12)**让系统发生可编程电压检测器(PVD)旗标、系统时钟安全旗标或是 CPU 发生锁定时让计数器暂时停止计数。

4.3.8 红外线接口输出配置

芯片支持用户使用计数器(Timer)搭配串口(UART)输出红外线(IR)的控制讯号, 藉此实现远程遥控的功能。在使用上需配置 **SYSCFG_IRSEL** (0x0C)内的 **SEL1**(位 0 至位 3)与 **SEL2**(位 4 至位 7)决定要如何搭配出 **PWM** 控制讯号, 此芯片目前仅支持用户从 **GP16C2T2** 与 **GP16C2T3** 中挑选其中一个信道输出 **PWM** 讯号去与 **GP16C4T4**、**UART3 Tx** 与 **SUART2 Tx** 的其中一个讯号进行逻辑 **AND** 运算, 并配置 **PLR**(位 8)决定 **IR** 输出讯号的极性。当 **PLR** 数值为 1 时代表 **SEL1** 与 **SEL2** 选择出来的讯号做逻辑 **AND** 运算, 反之若 **PLR** 数值为 0 则代表 **SEL1** 与 **SEL2** 选择出来的讯号做逻辑 **NAND** 运算。有关计数器与串口的配置方式请参阅相关章节的说明。

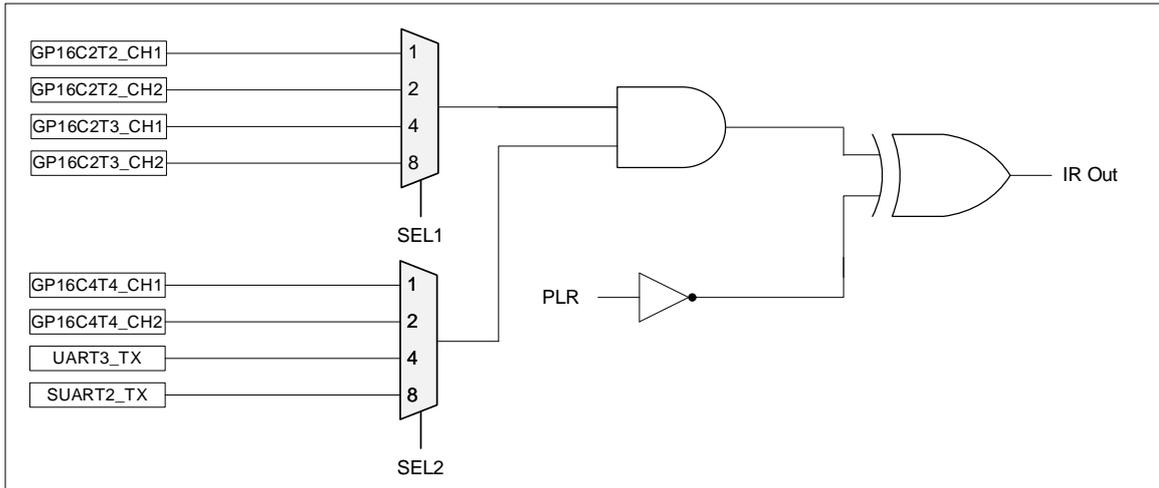


表 4-2 红外线控制讯号组合

4.4 特殊功能寄存器

4.4.1 寄存器列表

外设寄存器可支持半字（16 位）或字（32 位）访问。

SYSCFG 寄存器列表			
名称	偏移地址	类型	描述
SYSCFG_REMAP	0000 _H	R/W	系统启动内存重映射寄存器
SYSCFG_IRSEL	0008 _H	R/W	IR 输出选择寄存器
SYSCFG_CFG	003C _H	R/W	系统配置寄存器
SYSCFG_PWCON	0040 _H	R/W	系统电源控制与状态寄存器
SYSCFG_WKCON	0048 _H	R/W	低功耗唤醒控制寄存器
SYSCFG_WKSTAT	004C _H	R	唤醒状态寄存器
SYSCFG_BKREG0	0050 _H	R/W	备份 0 寄存器
SYSCFG_BKREG1	0054 _H	R/W	备份 1 寄存器
SYSCFG_BKREG2	0058 _H	R/W	备份 2 寄存器
SYSCFG_BKREG3	005C _H	R/W	备份 3 寄存器

4.4.2.2 IR 输出选择寄存器 (SYSCFG_IRSEL)

IR 输出选择寄存器 (SYSCFG_IRSEL)																																
偏移地址: 08H																																
复位值: 00000000_00000000_00000000_00000000B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																								PLR	SEL2<3:0>				SEL1<3:0>			

—	Bits 31-9	—	—
PLR	Bit 8	R/W	IR 极性 0: IR out = ~(SEL1 & SEL2) 1: IR out = SEL1 & SEL2
SEL2	Bits 7-4	R/W	IR 选择 2 0000: 关闭 0001: GP16C2T4_CH1 0010: GP16C2T4_CH2 0100: UART3_TX 1000: SUART2_TX
SEL1	Bits 3-0	R/W	IR 选择 1 0000: 关闭 0001: GP16C2T2_CH1 0010: GP16C2T2_CH2 0100: GP16C2T3_CH1 1000: GP16C2T3_CH2

4.4.2.3 系统配置寄存器 (SYSCFG_CFG)

系统配置寄存器 (SYSCFG_CFG)																																				
偏移地址: 3C _H																																				
复位值: 00000000_00000000_00000011_00000111 _B																																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
DBGH_EN<15:0>																—	PVD_LCK	CSS_LCK	CPU_LCK	—	—	—	—	NRSTREQ	—	CURRGEN	TEMPEN	VREFEN	VRLS<2:0>							

DBGH_EN	Bits 31-16	R/W	<p>DBG 停止使能位</p> <p>0: 即使内核停止, 计数器时钟仍然正常工作。 1: 当内核停止时计数器时钟停止。</p> <p>DBGH_EN<15>: Reserved DBGH_EN<14>: Reserved DBGH_EN<13>: IWDT DBGH_EN<12>: WWDT DBGH_EN<11>: Reserved DBGH_EN<10>: Reserved DBGH_EN<9>: BS16T1 DBGH_EN<8>: GP16C4T3 DBGH_EN<7>: GP16C4T2 DBGH_EN<6>: GP16C4T1 DBGH_EN<5>: GP32C4T1 DBGH_EN<4>: GP16C2T4 DBGH_EN<3>: GP16C2T3 DBGH_EN<2>: GP16C2T2 DBGH_EN<1>: GP16C2T1 DBGH_EN<0>: AD16C4T1</p>
—	Bit 15	—	—
PVD_LCK	Bit 14	R/W	PVD 锁定使能位
CSS_LCK	Bit 13	R/W	时钟安全系统锁定使能位
CPU_LCK	Bit 12	R/W	<p>Cortex-M0 LOCKUP 位使能位</p> <p>此位由软件设定, 并由系统复位清除。它可以用来启用和锁定 CORTEX-M0 LOCKUP (Hardfault) 输出到 AD16C4T/GP16C2T1-4 断路输入的连接。</p> <p>0: Cortex-M0 LOCKUP 输出从 AD16C4T /GP16C2T1-4 断路输入(Break input)断开 1: Cortex-M0 LOCKUP 输出连接到 AD16C4T/ GP16C2T1 /GP16C2T2/ GP16C2T3/ GP16C2T4 断路输入(Break input)端</p>
—	Bits 11-8	—	保留, 默认值为 3, 仅供测试使用不可更动。
NRSTREQ	Bit 7	R/W	<p>外部 NRST 引脚复位请求</p> <p>这个位是由 NRST 复位设置和清除。</p> <p>0: NRST 拉低请求禁用 1: NRST 拉低请求启用</p>
—	Bit 6	—	—

CURRGEN	Bit 5	R/W	电流发生器使能 该位由软件设置和清除。 0: ADC/DAC 电流生成器禁用 1: ADC/DAC 电流发生器启用
TEMPEN	Bit 4	R/W	温度传感器使能 该位由软件设置和清除。 0: 温度禁用 1: 温度启用
VREFEN	Bit 3	R/W	内部电压基准使能 该位由软件设置和清除。 0: 内部 VREF 禁用 1: 内部 VREF 启用
VRLS	Bits 2-0	R/W	这些位由软件编写, 通过 ADC/DAC 选择内部电压基准电平: 000: 1.5V 001: 2.0V 010: 2.5V 011: 3.0V 100: 3.5V 101: 4.0V 110: 4.5V 111: 5.0V(VDD5)

4.4.2.4 系统电源控制与状态寄存器 (SYSCFG_PWCON)

系统电源控制与状态寄存器 (SYSCFG_PWCON)																															
偏移地址: 40H																															
POR 复位值: 00000000_00001110_10000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								IOLOCK			BGTRIM<3:0>	HRCLOCK			HRCTRIM<7:0>									LOSCDRV<2:0>	PVDEN					PVLS<3:0>	

—	Bits 31-24	—	—
IOLOCK	Bit 23	R/W	WKUP0-7 IO 设定值锁定 0: 解锁 1: 锁定
—	Bits 22-21	—	—
BGTRIM	Bits 20-17	R/W	BandGap 校准设定值 BandGap 校准值不可改动。校准数值介于 0 至 15, 改变校准值会影响 Bandgap 电压值。
HRCLOCK	Bit 16	R/W	HRC 自动校准设定值锁定 0: 解锁 1: 锁定
HRCTRIM	Bits 15-8	R/W	HRC 自动校准设定值 当设置 SC_TRIM.TRIM 位时, 自动校准结果将更新到 SC_TRIM.HRCTRIM 寄存器, 必须重新加载到备份寄存器, 以确保系统一直保持低功耗。
LOSCDRV	Bits 7-5	R/W	LOSC 驱动程序选择 LOSC 驱动设定, 数值介于 0 至 7, 影响 LOSC 起振时间。
PVDEN	Bit 4	R/W	电源电压检测器 (PVD) 使能 该位由软件设置和清除。 0: PVD 禁用 1: PVD 启用
PVLS	Bits 3-0	R/W	这些位由软件写入, 以选择由功率电压检测器检测到的电压阈值: 0000: R-1.9V, F-2.0V 1000: R-3.5V, F-3.6V 0001: R-2.1V, F-2.2V 1001: R-3.7V, F-3.8V 0010: R-2.3V, F-2.4V 1010: R-3.9V, F-4.0V 0011: R-2.5V, F-2.6V 1011: R-4.1V, F-4.2V 0100: R-2.7V, F-2.8V 1100: R-4.3V, F-4.4V 0101: R-2.9V, F-3.0V 1101: R-4.5V, F-4.6V 0110: R-3.1V, F-3.2V 1110: R-4.7V, F-4.8V 0111: R-3.3V, F-3.4V 1111: R-4.9V, F-5.0V

4.4.2.5 低功耗唤醒控制寄存器 (SYSCFG_WKCON)

低功耗唤醒控制寄存器 (SYSCFG_WKCON)																															
偏移地址: 48H																															
POR 复位值: 00000000_01010000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKCLR		LPLS<2:0>			WKEG<11:0>												WKEN<11:0>														

WKCLR	Bit 31	C_W1	清除唤醒标志 当系统从低功耗流唤醒时 0: 不动作。 1: 清除 Wakeup 信号。
LPLS	Bits 30-28	R/W	LPLS: 低功率电平选择 0: Sleep Mode, 睡眠模式 (正常) 1: Stop Mode, 停止模式 (同 Sleep Mode) 2: Standby0 Mode, 待机模式 0 (SRAM 保留) 3: Standby1 Mode, 待机模式 1 (LDO 关闭) 4: Shutdown Mode, 停止运转模式 (LDO 关闭 /BandGap 关闭)
WKEG	Bits 27-16	R/W	WKEGx: WKUPx 引脚边沿有效类型选择 0: 下降沿有效 (默认) 1: 上升沿有效
—	Bits 15-12	—	—
WKEN	Bits 11-0	R/W	WKENx: 启用 WKUPx 引脚或唤醒事件 0: WKUPx 引脚用于通用 I/Os。WKUPx 引脚上的事件不会从待机模式唤醒设备。 1: WKUPx 引脚用于唤醒待机模式。 Bit 0-7: WKUPx 引脚 Bit 8: RTC 唤醒计数器 (默认上升) Bit 9: 比较器 0 Bit 10: 内部 PVD 标志 (默认上升) Bit 11: NRST 引脚

4.4.2.6 唤醒状态寄存器 (SYSCFG_WKSTAT)

唤醒状态寄存器 (SYSCFG_WKSTAT)																																
偏移地址: 4C _H																																
POR 复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
															FLAG																	
																FG<11:0>																

—	Bits 31-17	—	—
FLAG	Bit 16	R	唤醒标志 0: 无唤醒标志。 1: 由 Wakeup Pin 唤醒。
—	Bits 15-12	—	—
FG	Bits 11-0	R	FGx: WKUPx 引脚或唤醒事件标志 当该位设置为 1 时, 出现唤醒事件。 Bit 0-7: WKUPx 引脚 Bit 8: RTC 唤醒计数器 (默认上升) Bit 9: 比较器 0 Bit 10: 内部 PVD 标志 (默认上升) Bit 11: NRST 引脚

4.4.2.7 备份寄存器 0 (SYSCFG_BKREG0)

备份寄存器 0 (SYSCFG_BKREG0)																															
偏移地址: 50 _H																															
POR 复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																BKREG0<31:0>															

BKREG0	Bits 31-0	R/W	备份寄存器0 此字段保存在“常开”块中, 通过电源重置复位。
--------	-----------	-----	--

4.4.2.8 备份寄存器 1 (SYSCFG_BKREG1)

备份寄存器 1 (SYSCFG_BKREG1)																															
偏移地址: 54 _H																															
POR 复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKREG1<31:0>																															

BKREG1	Bits 31-0	R/W	备份寄存器1 此字段保存在“常开”块中，通过电源重置复位。
--------	-----------	-----	----------------------------------

4.4.2.9 备份寄存器 2 (SYSCFG_BKREG2)

备份寄存器 2 (SYSCFG_BKREG2)																															
偏移地址: 58 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKREG2<31:0>																															

BKREG2	Bits 31-0	R/W	备份寄存器 2 此字段保存在“常开”块中，通过电源重置复位。
--------	-----------	-----	-----------------------------------

4.4.2.10 备份寄存器 3 (SYSCFG_BKREG3)

备份寄存器 3 (SYSCFG_BKREG3)																															
偏移地址: 5C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKREG3<29:0>																															

—	Bits 31-30	—	—
BKREG3	Bits 29-0	R/W	备份寄存器3 此字段保存在“常开”块中，通过电源重置复位。

第5章 复位和时钟控制 (RCU)

5.1 概述

此章节内容包含系统时钟架构、时钟相关配置与外设复位设定，用户透过阅读此章节可了解如何配置系统时钟以及如何使用外设复位。

5.2 特性

- ◆ 支持 5 种时钟源可当作系统时钟。
- ◆ 支持 AHB 时钟分频，可以据系统时钟设定 2、4、8 与 16 的分频比。
- ◆ 支持 APB 时钟分频，可以据 AHB 时钟设定 2、4、8 与 16 的分频比。
- ◆ 支持无晶体振荡器下藉由外部输入时钟讯号当作 HOSC 时钟。
- ◆ 支持微控制器时钟输出。
- ◆ 支持微控制器时钟输出分频。
- ◆ 支持低功耗模式-睡眠模式(Sleep)与停止模式(Stop)下开启外设时钟。
- ◆ 支持时钟安全系统(CSS)。

5.3 功能描述

5.3.1 时钟

此芯片有 5 种时钟源可供用户选择当作系统时钟源，如下所示。

- ◆ HRC：内部高速 RC 振荡器，频率约 4 MHz，预设使用此时钟当作系统时钟。
- ◆ HOSC：外部高速晶体振荡器，支持用户使用频率 8 MHz 至 32MHz 的晶体振荡器。
- ◆ LRC：内部低速 RC 振荡器，频率约 50 kHz。
- ◆ LOSC：外部低速晶体振荡器，频率约 32.768 kHz。
- ◆ PLL0：内部锁相环倍频时钟，支持 4 MHz 至 48 MHz。

系统默认使用 HRC 为系统时钟源，若使用者想要选择其他时钟源当作系统时钟时，必须先确保该时钟源有被使能，同时须确保该时钟源已稳定后才可进行切换，用户可藉由读取 RCU_CON(0x00)的位 16 至位 21 的时钟稳定旗标来判断时钟源是否已经稳定，当时钟源被开启且稳定时，相对应的位会被设置为 1。

选定系统时钟后用户可依据需求调整 AHB 时钟与 APB 时钟的频率，而 AHB 时钟与 APB 时钟默认不分频，因此 AHB 与 APB 的外设默认在相同频率下进行操作。AHB 与 APB 时钟皆支持 2 分频、4 分频、8 分频以及 16 分频，用户可依据需求配置 RCU_CFG (0x04)内的 HPRE(位 4 至位 7)与 PPRE(位 8 至位 10)来决定 AHB 时钟与 APB 时钟的分频比。建议用户在配置系统时钟时一并配置 AHB 时钟与 APB 时钟的分频比，或是在程序运行时先暂时关闭外设时钟以后再行进行时钟分频配置，避免外设产生预期之外的行为。

综合上述内容描述，系统时钟切换的建议流程如下。

- ◆ 配置闪存控制器 **FC_CON**(0x10)内的 **WAIT**(位 0 至位 1)为 3，增加读取闪存所需要的等待时间，确保在任何系统频率下皆能顺利读取闪存。
- ◆ 配置 **RCU_CON**(0x00)开启要使用的时钟源。
 - ◇ 配置 **HRCON**(位 0)为 1 可开启 **HRC** 时钟源。
 - ◇ 配置 **HOSCON**(位 1)为 1 可开启 **HOSC** 时钟源。
 - ◇ 配置 **LRCON**(位 2)为 1 可开启 **LRC** 时钟源。
 - ◇ 配置 **LOSCON**(位 3)为 1 可开启 **LOSC** 时钟源。
 - ◇ 配置 **PLL0ON**(位 4)为 1 可开启 **PLL0** 时钟源。
- ◆ 读取 **RCU_CON**(0x00)内的时钟就绪标志确保该时钟源已稳定。
 - ◇ 若有开启 **HRC** 时钟，读取 **HRCRDY** (位 16)确认时钟就绪标志必须为 1。
 - ◇ 若有开启 **HOSC** 时钟，读取 **HOSCRDY** (位 17) 确认时钟就绪标志必须为 1。
 - ◇ 若有开启 **LRC** 时钟，读取 **LRCRDY** (位 18) 确认时钟就绪标志必须为 1。
 - ◇ 若有开启 **LOSC** 时钟，读取 **LOSCRDY** (位 19)确认时钟就绪标志必须为 1。
 - ◇ 若有开启 **PLL0** 时钟，读取 **PLL0RDY** (位 20)确认时钟就绪标志必须为 1。
- ◆ 若用户后续会使用 **PLL** 时，需额外进行配置 **RCU_CFG**(0x04)。
 - ◇ 配置 **PLLSRC**(位 16)切换 **PLL** 的参考时钟，默认选择 **HRC** 当作参考时钟源。
 - ◇ 当 **PLL** 参考时钟源选择 **HOSC** 时，需依据 **HOSC** 的频率调整 **HOSCDIV**(位 12 至位 14)让分频器产生 4 MHz 的参考时钟。
- ◆ 依据用户需求配置 **AHB** 时钟与 **APB** 时钟的分频
 - ◇ 配置 **RCU_CFG**(0x04)的 **HPRE** (位 4 至位 7)决定 **AHB** 时钟的分频比。
 - ◇ 配置 **RCU_CFG**(0x04)的 **PPRE** (位 8 至位 10) 决定 **APB** 时钟的分频比。
- ◆ 配置 **RCU_CFG**(0x04)的 **SW**(位 0 至位 2)来切换系统时钟。
 - ◇ 配置 **SW** 为 0 可将系统时钟切换至 **HRC** 时钟源。
 - ◇ 配置 **SW** 为 1 可将系统时钟切换至 **HOSC** 时钟源。
 - ◇ 配置 **SW** 为 2 可将系统时钟切换至 **LRC** 时钟源。
 - ◇ 配置 **SW** 为 3 可将系统时钟切换至 **LOSC** 时钟源。
 - ◇ 配置 **SW** 为 4 可将系统时钟切换至 **PLL0** 时钟源。
- ◆ 再次配置闪存控制器 **FC_CON**(0x10)内的 **WAIT**(位 0 至位 1)，此次设定主要依据 **AHB** 时钟的频率进行配置。当频率不高于 24 MHz 的情况下可配置 **WAIT** 的数值为 0；当频率高于 24 MHz 但未超过 48MHz 时则需配置 **WAIT** 的数值为 1。
- ◆ 将未使用到的时钟源关闭，须注意不可关闭已被配置为系统时钟的时钟源。

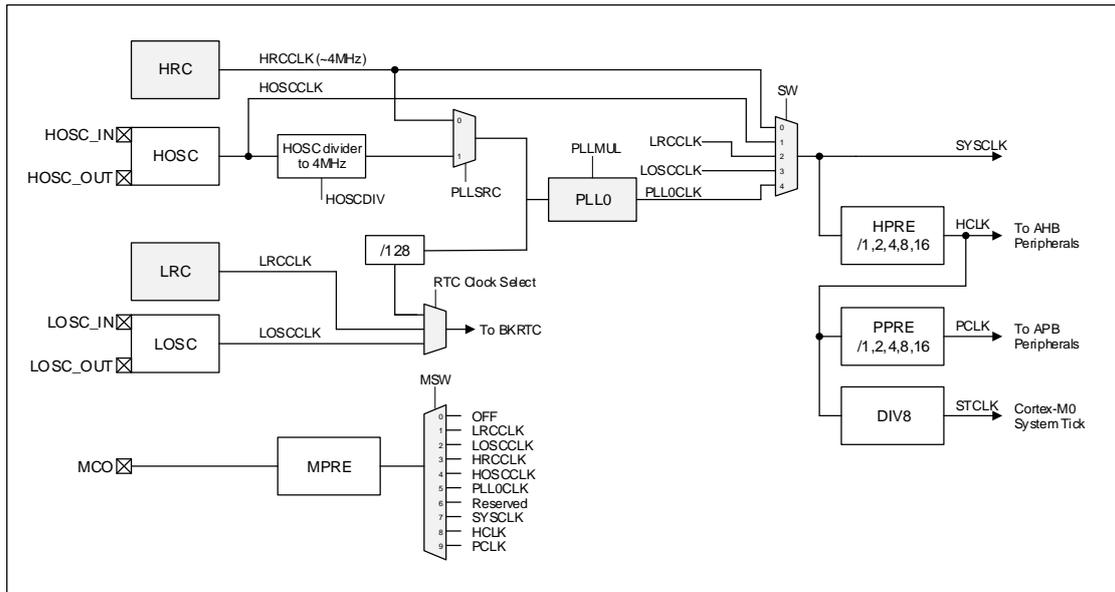


图 5-1 时钟树

5.3.1.1 内部高速 RC 振荡器(HRC)

内部高速 RC 振荡器(HRC)的频率固定为 4MHz，可直接当作系统时钟或是当作 PLL 的参考时钟使用。此时钟源预设开启，因此读取 RCU_CON(0x00)内 HRCON(位 0)的值为 1，同时读取 HRCRDY (位 16)的时钟就绪标志也会为 1。当用户不再需要使用 HRC 时钟时，可配置 RCU_CON(0x00)内 HRCON(位 0)的值为 0 来关闭此时钟源，并于关闭以后检查 RCU_CON(0x00)内 HRCRDY(位 16)的值为 0。由于 HRC 时钟为系统默认使用的时钟源，若用户有关闭的需求时请确保系统时钟已切换至 HRC 以外的时钟。

HRC 时钟在出厂前会经过校准，确保 HRC 的频率能够稳定的维持在 4MHz 左右。若用户有手动调整 HRC 频率的需求时，可藉由调整系统配置控制器内 SYSCFG_PWCON(0x40)的 HRCTRIM(位 8 至位 15)来调整 HRC 的频率，在调整的同时需确保 HRCLOCK(位 16)的数值保持在 1。由于此组寄存器为备份域的寄存器，因此在读操作与写操作上会有些许限制，有关此寄存器的操作说明请参阅系统配置控制器章节的描述。当用户在更改 HRC 的频率时，需避免使用 HRC 当作系统时钟，同时在调整校准数值时需避免一次调整过大的数值，以免因 HRC 不稳定而导致系统发生错误。

5.3.1.2 外部高速晶体振荡器(HOSC)

HOSC 时钟可直接当作系统时钟或是当作 PLL 的参考时钟使用，且 HOSC 时钟支持下列 2 种输入配置：

- ◆ 使用晶体振荡器当作时钟源输入：由外部晶体振荡器提供 8MHz 至 32MHz 的时钟输入，此时 HOSC 的精准度将会由外部晶体振荡器的规格决定。
- ◆ 使用外部时钟讯号当作时钟输入：当用户选择使用外部讯号源当作时钟时，需配置 RCU_CON (0x00)内的 HOSCON(位 1)为 1 并从 PD15_HOSCO 脚位输入时钟讯号。

开启时钟源后必须检查 HOSCRDY (位 17)的时钟就绪标志是否为 1 来确认 HOSC 时钟已经稳定，当时钟源稳定后才可使用此时钟。当用户不再需要使用 HOSC 时钟时，可配置

RCU_CON(0x00)内 **HOSCON(位 1)**的值为 0 来关闭此时钟源，并于关闭时钟后再次检查 **HOSCRDY(位 5)**的时钟就绪标志为 0 确认时钟源关闭。用户须注意的是若 **HOSC** 被选择当作系统时钟使用时不可将 **HOSC** 时钟源关闭。

由于 **HOSC** 时钟的输入与输出管脚与 **GPIO** 功能共享，因此当用户选择使用 **HOSC** 旁路模式时 **PD14_HOSCI** 是能够当作一般 **GPIO** 使用的，用户可依据需求调整 **PD14_HOSCI** 的功能，有关 **GPIO** 的配置方法请参阅 **GPIO** 章节内的描述。

5.3.1.3 内部低速 RC 振荡器(LRC)

内部低速 **RC** 振荡器(LRC)的频率接近 50KHz，主要提供系统开机流程与低功耗模式使用，或是当芯片长时间处于待机状态下也可以将系统时钟切换至 **LRC** 藉此降低芯片的功率消耗，此外此时钟也是独立看门狗用来计数的时钟，用户若有使用独立看门狗的需求时必须开启 **LRC** 时钟，有关独立看门狗的说明与配置方式请参阅第 22 章独立看门狗 (**IWDT**)章节内的描述。

此时钟源可藉由配置 **RCU_CON(0x00)**内的 **LRCON(位 2)**为 1 来开启，开启时钟源后必须检查 **LRCDY(位 18)**的时钟就绪标志是否为 1 来确认 **LRC** 时钟已经稳定，当时钟源稳定后才可使用此时钟。当用户不再需要使用 **LRC** 时钟时，可配置 **RCU_CON(0x00)**内 **LRCON(位 2)**的值为 0 来关闭此时钟源，用户须于关闭时钟后再次检查 **LRCDY(位 18)**的时钟就绪标志为 0 确认时钟源关闭。

5.3.1.4 外部低速晶体振荡器(LOSC)

LOSC 时钟源支持 32.768KHz 的外部时钟输入，可直接当作系统时钟或是当作实时时钟(RTC)的时钟源使用。此时钟支持用户使用晶体振荡器当作时钟源或是直接使用外部时钟讯号输入。当使用 **LOSC** 当作实时时钟(RTC)的时钟源时，晶体振荡器的精准度会直接反映在实时时钟(RTC)的计时误差上，用户若对实时时钟(RTC)计时有精准度上的要求时，可以先从 **LOSC** 的精准度上进行调整，再藉由实时时钟(RTC)的校准功能提升 **RTC** 的计时精度，有关 **RTC** 的说明与校准方式请参阅第 22 章实时时钟(RTC)章节内的描述。

5.3.1.5 内部倍频时钟(PLL0)

芯片内含一组倍频时钟可当作系统时钟使用，需提供频率为 4MHz 的参考时钟，并以 4MHz 为单位进行倍频，最高可支持至 48MHz (最大倍率为 12 倍)。倍频时钟 **PLL0** 默认使用 **HRC** 当作参考时钟，若用户希望更改为使用 **HOSC** 当作 **PLL0** 参考时钟的来源时，需依据使用的晶体振荡器频率设定 **RCU_CFG(0x04)**内的 **HOSCDIV(位 12 至位 14)**让分频器产生 4 MHz 的参考时钟，同时设定 **RCU_CFG(0x04)**内的 **PLLSRC(位 16)**切换 **PLL0** 的参考时钟。

用户在配置完参考时钟源以后可再依据需求变更 **RCU_CFG(0x04)**内的 **PLLMUL(位 17 至位 21)**来调整 **PLL0** 的频率，建议最高不可让 **PLLMUL** 的数值超过 11(**PLL0** 输出最高稳定频率为 48MHz)。

5.3.1.6 外设时钟

系统预设不开启任何外设时钟，当用户需要使用外设的功能时，需配置 **RCU_AHBEN(0x1C)**开启 **AHB** 区块的外设时钟、配置 **RCU_APB1EN(0x20)**开启位于 **APB1** 区块的外设时钟或配置 **RCU_APB2EN(0x24)**开启位于 **APB2** 区块的外设时钟。用户仅需配置寄存器内相对应外设的位为 1 即可开启该外设的时钟，如配置 **RCU_AHBEN(0x1C)**内的 **GPCEN(位 18)**为 1，即可开启 **GPIOC** 外设时钟。

	外设名称	寄存器变量名称
AHB	GPIO D	GPDEN
	GPIO C	GPCEN
	GPIO B	GPBEN
	GPIO A	GPAEN
	HDIV	HDIVEN
	RTC	RTCEN
	DMA	DMA1EN
APB1	DAC	DACEN
	I2C 2	I2C2EN
	I2C 1	I2C1EN
	Simple UART 2	SUART2EN
	Simple UART 1	SUART1EN
	UART 3	UART3EN
	UART 2	UART2EN
	SPI 2	SPI2EN
	Independent Watchdog	IWDTEN
	Window Watchdog	WWDTEN
	BS16T1	BS16T1EN
	GP16C4T 3	GP16C4T3EN
	GP16C4T 2	GP16C4T2EN
	GP16C4T 1	GP16C4T1EN
	GP32C4T1	GP32C4T1EN
APB2	Compartor	CMPEN
	MCM	MCMEN
	GP16C2T 4	GP16C2T4EN
	GP16C2T 3	GP16C2T3EN
	GP16C2T 2	GP16C2T2EN
	GP16C2T 1	GP16C2T1EN
	UART 1	UART1EN
	SPI 1	SPI1EN
	AD16C4T	AD16C4T1EN
	ADC	ADCEN

表 5-1 外设分配表

5.3.2 复位

此芯片提供多种复位功能，各种复位的描述分别如下所示。

5.3.2.1 上电/掉电复位

当系统电源从 0 伏上升至超过 V_{POR} 时 POR 模块会于等待约 2.1 毫秒(2.1ms)后释放 POR 复位，此时系统便会离开复位模式并开始执行开机流程。当系统电源从 V_{DD5} 降至低于 V_{PDR} 时会让 POR 模块重新拉低 POR 复位，此时系统会进入复位状态，以确保所有模块不会因电源不稳定而产生预期之外的行为，当此复位发生时用户存放于备份寄存器内的数据也会一并被清除。有关上电/掉电的描述请参阅系统配置控制器章节内的电源章节。

5.3.2.2 低功耗复位

当系统进入待机模式 0、待机模式 1 与停止运转模式时，会暂时关闭包含 CPU、SRAM 与 AHB/APB 外设的电源，因此当系统从这 3 种低功耗模式唤醒时会触发低功耗复位，让 CPU 重新开始执行程序。由于待机模式 0 会额外开启 1.2 伏稳压器，因此进入此低功耗模式下仍会保留 SRAM 内的信息（仅保留位置为 $0x20001000 \sim 0x20001FFF$ 区间的信息）。当系统被唤醒后，可检查 **SYSCFG_WKSTAT**($0x4C$)内的 FG(位 0 至位 11)判断是由哪一个唤醒事件唤醒，并配置 **SYSCFG_WKCON** ($0x48$)内的 WKCLR(位 31)为 1 来清除唤醒旗标。有关低功耗模式的说明与配置方式请参阅系统配置控制器章节内低功耗模式的内容描述。

5.3.2.3 NRST 复位

NRST 复位可清除 CPU 与 AHB/APB 外设的相关配置，而备份寄存器内所储存的信息依然会保留，此复位可由下列 2 种方式触发：

- ◆ 拉低外部 NRST 管脚：当用户将 NRST 管脚的讯号拉低一段时间(超过 180ns) 后可触发系统复位。
- ◆ 软件触发 NRST 复位：用户可藉由配置系统配置控制器内 **SYSCFG_CFG**($0x3C$)寄存器的 NRSTREQ(位 7)为 1 来触发外部 NRST 复位，而 NRSTREQ 会在 NRST 复位发生时被重置回 0。

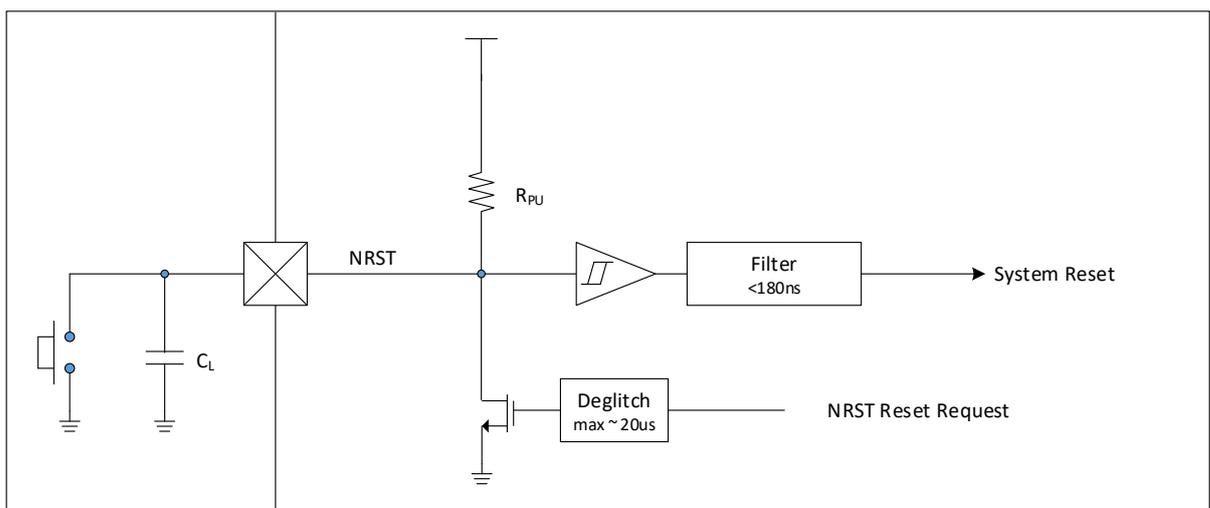


图 5-2 NRST 复位

5.3.2.4 软件复位

用户可以使用 `NVIC_SystemReset()` 复位函数来重启 CPU，此复位函数仅会重置 CPU，并不会将 AHB/APB 外设配置信息清除，因此用户存放于备份寄存器内所储存的信息与外设的相关配置依然会保留。

5.3.2.5 看门狗复位

看门狗可分为独立看门狗与窗口看门狗，这 2 种看门狗都可以定时触发 CPU 复位来重启 CPU，避免系统发生死机。有关看门狗的使用方法请参阅独立看门狗与窗口看门狗章节内的描述。

5.3.2.6 AHB/APB 外设复位

AHB/APB 外设默认都处在复位状态，当用户开启外设时钟后需再等待 3 个外设时钟才会离开复位状态。用户可藉由配置 `RCU_AHBRST(0x10)`、`RCU_APB1RST(0x14)` 与 `RCU_APB2RST(0x18)` 来重置外设。当这 3 组寄存器被设定为 1 时外设会处在复位状态，如配置 `RCU_AHBRST(0x10)` 内的 `GPCEN` (位 18) 为 1，便可让 GPIOC 外设进入复位状态，用户需自行再将这 3 组寄存器清除为 0 让外设离开复位状态。需特别注意当释放外设复位后，需等待 3 个外设时钟才会离开复位状态，在这段时间内是无法对外设的寄存器进行读写操作。由于 AHB 时钟与 APB 时钟的频率最多可能差至 16 倍，因此用户在使用此功能时需特别注意程序上的配置，不建议用户在对 `RCU_AHBRST(0x10)`、`RCU_APB1RST(0x14)` 与 `RCU_APB2RST(0x18)` 这 3 组寄存器填入 0 后随即访问外设的控制寄存器。

5.3.3 微控制器时钟输出

此芯片支持用户将时钟源输出，输出的时钟讯号除了可供用户检查以外也可当作其他装置的时钟源。用户可配置 `RCU_CFG(0x04)` 内的 `MSW` (位 24 至位 27) 选择要输出哪一个时钟，并依据需求配置 `MPRE` (位 28 至位 30) 将微控制器输出时钟进行分频，最多可分频 8 倍。用户须注意的是当分频比不为偶数倍时，则产生出的 MCO 占空比 (Duty Cycle) 不为 50%。由于此功能与 GPIO 共享管脚，因此用户需额外配置 GPIO 为复用功能 7 模式 (Alternate function mode 7)。

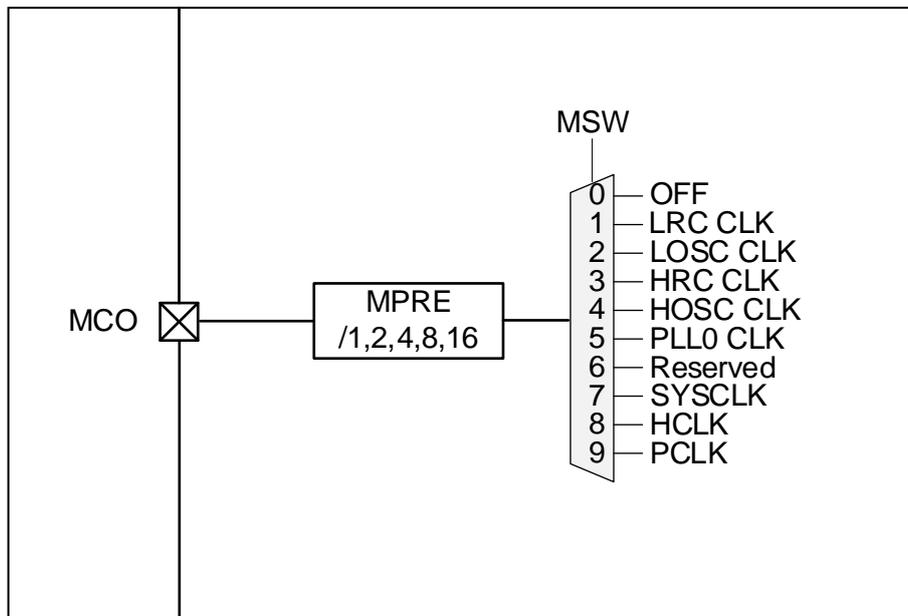


图 5-3 微控制器时钟输出

5.3.4 睡眠模式中的外设时钟使能

当系统进入低功耗模式-睡眠模式时，默认会关闭 CPU 与周边 IP 的时钟，借此达到降低功耗的

效果，但使用者仍可在 CPU 进入睡眠模式(Sleep Mode)前通过配置 **SC_AHBSL(0x28)**、**SC_APB1SL(0x2C)**与 **SC_APB2SL(0x30)**来开启 IP 的时钟，或是在 CPU 进入深度睡眠模式(Deep Sleep Mode) 前通过配置 **SC_AHBDL(0x34)**、**SC_APB1DSL(0x38)** 与 **SC_APB2DSL(0x3C)**来开启 IP 的时钟。被开启时钟的 IP 在系统进入低功耗模式-睡眠模式(Sleep)与停止模式(Stop)的期间皆会持续运作，因此可通过被开启的 IP 于特殊条件下使用中断唤醒系统。详细的低功耗模式简介以及设定方式请参阅系统配置控制器章节内的描述。

5.3.5 时钟安全系统

芯片支持侦测 HOSC 是否因故障导致停振的功能。当系统选择 HOSC 当作系统时钟，或是系统选择 PLL 当作系统时钟同时 PLL 的参考时钟选择 HOSC 时，当 HOSC 发生故障导致 HOSC 停振，若用户于配置系统时钟时有设定 **RCU_CON(0x00)**内的 **CSSON(位 7)**为 1 的话，系统会自动将系统时钟切换至 HRC，避免系统因为没有系统时钟而无法继续运行。若 HOSC 因特殊原因又恢复输出时钟时，则系统会在 HOSC 稳定后，依据原先设定将系统时钟更换回原先设定的时钟。若使用者有需要开启时钟安全系统的功能时，请确保 HRC 时钟源有被开启，否则会因 HRC 模块没有被使能而造成系统无法将系统时钟切换至 HRC。

5.4 特殊功能寄存器

5.4.1 寄存器列表

外设寄存器可支持半字（16 位）或字（32 位）访问。

RCU 寄存器列表			
名称	偏移地址	类型	描述
RCU_CON	0000 _H	R/W	时钟控制寄存器
RCU_CFG	0004 _H	R/W	时钟配置寄存器
RCU_STCALIB	0008 _H	R/W	System Tick 时间校准值寄存器
RCU_PSCK	000C _H	R/W	外设专用时钟控制寄存器
RCU_AHBRST	0010 _H	R/W	AHB 外设复位寄存器
RCU_APB1RST	0014 _H	R/W	APB1 外设复位寄存器
RCU_APB2RST	0018 _H	R/W	APB2 外设复位寄存器
RCU_AHBEN	001C _H	R/W	AHB 外设时钟使能寄存器
RCU_APB1EN	0020 _H	R/W	APB1 外设时钟使能寄存器
RCU_APB2EN	0024 _H	R/W	APB2 外设时钟使能寄存器
RCU_AHBSL	0028 _H	R/W	AHB 睡眠模式寄存器中的外设时钟使能
RCU_APB1SL	002C _H	R/W	APB1 睡眠模式中的外设时钟使能
RCU_APB2SL	0030 _H	R/W	APB2 睡眠模式中的外设时钟使能
RCU_AHBDL	0034 _H	R/W	AHB 深度睡眠模式寄存器中的外设时钟使能
RCU_APB1DSL	0038 _H	R/W	APB1 深度睡眠模式寄存器中的外设时钟使能
RCU_APB2DSL	003C _H	R/W	APB2 深度睡眠模式寄存器中的外设时钟使能

5.4.2 寄存器描述

5.4.2.1 时钟控制寄存器 (RCU_CON)

时钟控制寄存器 (RCU_CON)																															
偏移地址: 00H																															
复位值: 00000000_00000001_00000000_00000001b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SWRDY	CSSFLG			PLLORDY	LOSCRDY	LRCRDY	HOSCRDY	HRCRDY									CSSON			PLLOON	LOSCON	LRCON	HOSCON	HRCON

—	Bits 31-25	—	—
SWRDY	Bit 24	R	系统时钟转换就绪标志 该位由硬件设置,以指示系统时钟变化是稳定的。 0: 系统时钟未准备好 1: 系统时钟准备就绪
CSSFLG	Bit 23	R	时钟安全系统标志 CSSEN 设定为 0 可关闭提示,或是 HOSC 恢复时自动清除。 0: 没有 HOSC 时钟失效引起的时钟安全事件 1: 有 HOSC 时钟故障引起的时钟安全事件
—	Bits 22-21	—	—
PLLORDY	Bit 20	R	PLL0 时钟就绪标志 该比特由硬件设置,以指示 PLL0 振荡器是稳定的。 0: PLL0 未准备好 1: PLL0 准备就绪
LOSCRDY	Bit 19	R	LOSC 时钟就绪标志 该位由硬件设置,以指示 LOSC 振荡器是稳定的。 0: LOSC 振荡器未准备好 1: LOSC 振荡器准备就绪
LRCRDY	Bit 18	R	LRC 时钟就绪标志 该位由硬件设置,以指示 LRC 振荡器是稳定的。 0: LRC 振荡器未准备好 1: LRC 振荡器准备就绪
HOSCRDY	Bit 17	R	HOSC 时钟就绪标志 该位由硬件设置,以指示 HOSC 振荡器是稳定的。 0: HOSC 振荡器未准备好 1: HOSC 振荡器准备就绪
HRCRDY	Bit 16	R	HRC 时钟就绪标志 该位由硬件设置,以指示 HRC 振荡器是稳定的。

			0: HRC 振荡器未准备好 1: HRC 振荡器准备就绪
—	Bits 15-8	—	—
CSSON	Bit 7	R/W	时钟安全系统使能 (CSS) 由软件设置和清除，启用时钟安全系统。当 CSSON 被设置时，当 HOSC 振荡器准备好时，硬件启用时钟检测器，如果检测到 HOSC 时钟故障，则硬件禁用时钟检测器。 0: 时钟安全系统禁用(时钟检测器关闭)。 1: 启用时钟安全系统(如果 HOSC 已准备好，时钟检测器打开，如果没有就关闭)。
—	Bits 6-5	—	—
PLL0ON	Bit 4	R/W	PLL0 使能位 0: PLL0 关闭 1: PLL0 开启
LOSCON	Bit 3	R/W	外部低速振荡器时钟位 0: LOSC 关闭 1: LOSC 开启
LRCON	Bit 2	R/W	内置低速振荡器时钟位 0: LRC 关闭 1: LRC 开启
HOSCON	Bit 1	R/W	外部高速振荡器时钟位 0: HOSC 关闭 1: HOSC 开启
HRCON	Bit 0	R/W	内置高速振荡器时钟位 0: HRC 关闭 1: HRC 开启

5.4.2.2 时钟配置寄存器 (RCU_CFG)

时钟配置寄存器 (RCU_CFG)																																			
偏移地址: 04H																																			
复位值: 00000000_00000000_00000000_00000000B																																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
CKCFG		MPRE<2:0>			MSW<3:0>								PLLMUL<4:0>				PLLSRC				HOSCDIV<2:0>					PPRE<2:0>			HPRE<3:0>					SW<2:0>	

CKCFG	Bit 31	W	时钟配置设置位 这个位是只写入。对该位读取则读取到 0。 如果设置为高位，则开始配置时钟设置。当时钟配置过程完成时，该位由自身清除为零。
MPRE	Bits 30-28	R/W	微控制器时钟输出预分频器 输出频率 = MCO/(MPRE+1)
MSW	Bits 27-24	R/W	微控制器时钟输出 (MCO) 0000 : MCO 输出禁用, MCO 上没有时钟 0001 : 内部低速(LRC) 振荡器时钟选择 0010 : 外部低速(LOSC) 振荡器时钟选择 0011 : 内部高速(HRC) 振荡器时钟选择 0100 : 外部高速(HOSC) 振荡器时钟选择 0101 : PLL0 时钟选择 0110 : 保留 0111 : 系统时钟选择(SYSCLK) 1000 : AHB 时钟选择(HCLK) 1001 : APB 时钟选择(PCLK) 1xx1 : 保留, MCO 输出禁用
—	Bits 23-22	—	—
PLLMUL	Bits 21-17	R/W	PLL0 倍频系数 PLL0 输出时钟频率为 4X(PLL0MUL+1) MHz 输出最大频率不能超过 48 MHz, PLL0MUL = 11.
PLLSRC	Bit 16	R/W	PLL0 参考时钟源 0: HRC 被选作 PLL0 参考时钟。 1: HOSC/HOSCDIV 被选作 PLL 参考时钟。
—	Bit 15	—	—
HOSCDIV	Bit 14-12	R/W	PLL0 基准时钟 4 MHz 的 HOSC 时钟分频系数 这些位被软件设置和清除，选择 HOSCDIV 除法因子。 PLL_reference_clock (4 MHz) = HOSC / (HOSCDIV+1) 例: HOSC(20 MHz) / HOSCDIV(4)+1 = 4 MHz

—	Bit 11	—	—
PPRE	Bits 10-8	R/W	APB 时钟预分频器(PCLK) 0xx : HCLK 不分频 100 : HCLK 2 分频 101 : HCLK 4 分频 110 : HCLK 8 分频 111 : HCLK 16 分频
HPRE	Bits 7-4	R/W	AHB 时钟预分频器 (HCLK) 0xxx : SYSCLK 不分频 1000 : SYSCLK 2 分频 1001 : SYSCLK 4 分频 1010 : SYSCLK 8 分频 1011 : SYSCLK 16 分频 11XX : 保留
—	Bit 3	—	—
SW	Bits 2-0	R/W	系统时钟切换 (SYSCLK) 000 : HRC 作为系统时钟选择。 001 : HOSC 作为系统时钟选择。 010 : LRC 作为系统时钟选择。 011 : LOSC 作为系统时钟选择。 100 : PLL0 作为系统时钟选择。 1xx : 保留

5.4.2.3 系统时钟定时器校准值寄存器 (RCU_STCALIB)

系统时钟定时器校准值寄存器 (RCU_STCALIB)																															
偏移地址: 08H																															
复位值: 00000000_00000000_10011100_00111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SKEW			TENMS<23:0>																												

—	Bit 31	—	—
SKEW	Bit 30	R	TENMS 校准值是否准确的标志位 0: TENMS 校准值准确 1: TENMS 校准值不准确
—	Bits 29-24	—	—
TENMS	Bits 23-0	R/W	十毫秒校准值 其值取决于 MCU 设计。 默认值为 HCLK/8=4 MHz/8=500 kHz. (5000-1)

5.4.2.4 外设专用时钟控制寄存器 (RCU_PSCK)

外设专用时钟控制寄存器 (RCU_PSCK)																															
偏移地址: 0CH																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										I2S2CKSL<1:0>		I2S1CKSL<1:0>			

—	Bits 31-4	—	—
I2S2CKSL	Bits 3-2	R/W	SPI2/I2S2 外部时钟源选择 00: 禁用外部时钟(默认值) 01: I2SCLK1 10: I2SCLK2 11: 保留
I2S1CKSL	Bits 1-0	R/W	SPI1/I2S1 外部时钟源选择 00: 禁用外部时钟(默认值) 01: I2SCLK1 10: I2SCLK2 11: 保留

5.4.2.5 AHB 外设复位寄存器 (RCU_AHBRST)

AHB 外设复位寄存器 (RCU_AHBRST)																																	
偏移地址: 10H																																	
复位值: 00000000_00000000_00000000_00000000B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
												GPDEN	GPCEN	GPBEN	GPAEN	HDIVEN										RTCEN							DMA1EN

—	Bits 31-20	—	—
GPDEN	Bit 19	R/W	GPIO D 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GPIO D 端口
GPCEN	Bit 18	R/W	GPIO C 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GPIO C 端口
GPBEN	Bit 17	R/W	GPIO B 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GPIO B 端口
GPAEN	Bit 16	R/W	GPIO A 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GPIO A 端口
HDIVEN	Bit 15	R/W	HDIV 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 HDIV
—	Bits 14-7	—	—
RTCEN	Bit 6	R/W	RTC 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 RTC
—	Bits 5-1	R/W	—
DMA1EN	Bit 0	R/W	DMA1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 DMA1

5.4.2.6 APB1 外设复位寄存器 (RCU_APB1RST)

APB1 外设复位寄存器 (RCU_APB1RST)																																
偏移地址: 14H																																
复位值: 00000000_00000000_00000000_00000000B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								DACEN	I2C2EN	I2C1EN	SUART2EN	SUART1EN	UART3EN	UART2EN			SPI2EN		IWDTEN	WWDTEN								BS16T1EN	GP16C4T3EN	GP16C4T2EN	GP16C4T1EN	GP32C4T1EN

—	Bits 31-24	—	—
DACEN	Bit 23	R/W	DAC 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 DAC
I2C2EN	Bit 22	R/W	I2C2 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 I2C2
I2C1EN	Bit 21	R/W	I2C1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 I2C1
SUART2EN	Bit 20	R/W	SUART2 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 SUART2
SUART1EN	Bit 19	R/W	SUART1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 SUART1
UART3EN	Bit 18	R/W	UART3 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 UART3
UART2EN	Bit 17	R/W	UART2 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 UART2
—	Bits 16-15	—	—
SPI2EN	Bit 14	R/W	SPI2 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 SPI2
—	Bit 13	—	—
IWDTEN	Bit 12	R/W	IWDT 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 IWDT
WWDTEN	Bit 11	R/W	WWDT 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 WWDT
—	Bits 10-5	—	—
BS16T1EN	Bit 4	R/W	BS16T1 复位请求使能位 0: 设置后手动清除

			1: 置入复位信号到 BS16T1
GP16C4T3EN	Bit 3	R/W	GP16C4T3 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GP16C4T3
GP16C4T2EN	Bit 2	R/W	GP16C4T2 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GP16C4T2
GP16C4T1EN	Bit 1	R/W	GP16C4T1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GP16C4T1
GP32C4T1EN	Bit 0	R/W	GP32C4T1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GP32C4T1

5.4.2.7 APB2 外设复位寄存器 (RCU_APB2RST)

APB2 外设复位寄存器 (RCU_APB2RST)																																
偏移地址: 18H																																
复位值: 00000000_00000000_00000000_00000000B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								CMPEN		MCMEN		GP16C2T4EN	GP16C2T3EN	GP16C2T2EN	GP16C2T1EN		UART1EN		SPI1EN	AD16C4T1EN		ADCEN										

—	Bits 31-24	—	—
CMPEN	Bit 23	R/W	CMP 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 CMP
—	Bit 22	—	—
MCMEN	Bit 21	R/W	MCM 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 MCM
—	Bit 20	—	—
GP16C2T4EN	Bit 19	R/W	GP16C2T4 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GP16C2T4
GP16C2T3EN	Bit 18	R/W	GP16C2T3 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GP16C2T3
GP16C2T2EN	Bit 17	R/W	GP16C2T2 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GP16C2T2
GP16C2T1EN	Bit 16	R/W	GP16C2T1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 GP16C2T1
—	Bit 15	—	—
UART1EN	Bit 14	R/W	UART1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 UART1
—	Bit 13	R/W	—
SPI1EN	Bit 12	R/W	SPI1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 SPI1
AD16C4T1EN	Bit 11	R/W	AD16C4T1 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 AD16C4T1
—	Bit 10	—	—
ADCEN	Bit 9	R/W	ADC 复位请求使能位 0: 设置后手动清除 1: 置入复位信号到 ADC

5.4.2.9 APB1 外设时钟使能寄存器 (RCU_APB1EN)

APB1 外设时钟使能寄存器 (RCU_APB1EN)																															
偏移地址: 20H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DACEN	I2C2EN	I2C1EN	SUART2EN	SUART1EN	UART3EN	UART2EN			SPI2EN		IWDTEN	WWDTEN							BS16T1EN	GP16C4T3EN	GP16C4T2EN	GP16C4T1EN	GP32C4T1EN

—	Bits 31-24	—	—
DACEN	Bit 23	R/W	DAC 时钟使能位 0: 时钟禁用 1: 时钟使能
I2C2EN	Bit 22	R/W	I2C2 时钟使能位 0: 时钟禁用 1: 时钟使能
I2C1EN	Bit 21	R/W	I2C1 时钟使能位 0: 时钟禁用 1: 时钟使能
SUART2EN	Bit 20	R/W	SUART2 时钟使能位 0: 时钟禁用 1: 时钟使能
SUART1EN	Bit 19	R/W	SUART1 时钟使能位 0: 时钟禁用 1: 时钟使能
UART3EN	Bit 18	R/W	UART3 时钟使能位 0: 时钟禁用 1: 时钟使能
UART2EN	Bit 17	R/W	UART2 时钟使能位 0: 时钟禁用 1: 时钟使能
—	Bits 16-15	—	—
SPI2EN	Bit 14	R/W	SPI2 时钟使能位 0: 时钟禁用 1: 时钟使能
—	Bit 13	—	—
IWDTEN	Bit 12	R/W	IWDT 时钟使能位 0: 时钟禁用 1: 时钟使能
WWDTEN	Bit 11	R/W	WWDT 时钟使能位 0: 时钟禁用 1: 时钟使能
—	Bits 10-5	—	—
BS16T1EN	Bit 4	R/W	BS16T1 时钟使能位 0: 时钟禁用

			1: 时钟使能
GP16C4T3EN	Bit 3	R/W	GP16C4T3 时钟使能位 0: 时钟禁用 1: 时钟使能
GP16C4T2EN	Bit 2	R/W	GP16C4T2 时钟使能位 0: 时钟禁用 1: 时钟使能
GP16C4T1EN	Bit 1	R/W	GP16C4T1 时钟使能位 0: 时钟禁用 1: 时钟使能
GP32C4T1EN	Bit 0	R/W	GP32C4T1 时钟使能位 0: 时钟禁用 1: 时钟使能

5.4.2.10 APB2 外设时钟使能寄存器 (RCU_APB2EN)

APB2 外设时钟使能寄存器 (RCU_APB2EN)																																
偏移地址: 24H																																
复位值: 00000000_00000000_00000000_00000000B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								CMPEN		MCMEN		GP16C2T4EN	GP16C2T3EN	GP16C2T2EN	GP16C2T1EN		UART1EN		SPI1EN	AD16C4T1EN		ADCEN										

—	Bits 31-24	—	—
CMPEN	Bit 23	R/W	CMP 时钟使能位 0: 时钟禁用 1: 时钟使能
—	Bit 22	—	—
MCMEN	Bit 21	R/W	MCM 时钟使能位 0: 时钟禁用 1: 时钟使能
—	Bit 20	—	—
GP16C2T4EN	Bit 19	R/W	GP16C2T4 时钟使能位 0: 时钟禁用 1: 时钟使能
GP16C2T3EN	Bit 18	R/W	GP16C2T3 时钟使能位 0: 时钟禁用 1: 时钟使能
GP16C2T2EN	Bit 17	R/W	GP16C2T2 时钟使能位 0: 时钟禁用 1: 时钟使能
GP16C2T1EN	Bit 16	R/W	GP16C2T1 时钟使能位 0: 时钟禁用 1: 时钟使能
—	Bit 15	—	—
UART1EN	Bit 14	R/W	UART1 时钟使能位 0: 时钟禁用 1: 时钟使能
—	Bit 13	—	—
SPI1EN	Bit 12	R/W	SPI1 时钟使能位 0: 时钟禁用 1: 时钟使能
AD16C4T1EN	Bit 11	R/W	AD16C4T1 时钟使能位 0: 时钟禁用 1: 时钟使能
—	Bit 10	—	—
ADCEN	Bit 9	R/W	ADC 时钟使能位 0: 时钟禁用 1: 时钟使能

—	Bits 8-0	—	—
---	----------	---	---

5.4.2.11 AHB 睡眠模式中的外设时钟使能寄存器 (RCU_AHBSL)

AHB 睡眠模式中的外设时钟使能寄存器 (RCU_AHBSL)																																	
偏移地址: 28H																																	
复位值: 00000000_00000000_00000000_00000000b																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
													GPDEN	GPCEN	GPBEN	GPAEN	HDIVEN																

—	Bits 31-20	—	—
GPDEN	Bit 19	R/W	GPIOD 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GPCEN	Bit 18	R/W	GPIOC 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GPBEN	Bit 17	R/W	GPIOB 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GPAEN	Bit 16	R/W	GPIOA 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
HDIVEN	Bit 15	R/W	HDIV 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bit 14-7	—	—
RTCEN	Bit 6	R/W	RTC 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bits 5-1	—	—
DMA1EN	Bit 0	R/W	DMA1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟

5.4.2.12 APB1 休眠模式中的外设时钟使能寄存器 (RCU_APB1SL)

APB1 睡眠模式中的外设时钟使能寄存器 (RCU_APB1SL)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DACEN	I2C2EN	I2C1EN	SUART2EN	SUART1EN	UART3EN	UART2EN			SPI2EN		IWDTEN	WWDTEN							BS16T1EN	GP16C4T3EN	GP16C4T2EN	GP16C4T1EN	GP32C4T1EN

—	Bits 31-24	—	—
DACEN	Bit 23	R/W	DAC 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
I2C2EN	Bit 22	R/W	I2C2 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
I2C1EN	Bit 21	R/W	I2C1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
SUART2EN	Bit 20	R/W	SUART2 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
SUART1EN	Bit 19	R/W	SUART1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
UART3EN	Bit 18	R/W	UART3 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
UART2EN	Bit 17	R/W	UART2 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bits 16-15	—	—
SPI2EN	Bit 14	R/W	SPI2 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bit 13	—	—
IWDTEN	Bit 12	R/W	IWDT 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
WWDTEN	Bit 11	R/W	WWDT 睡眠模式中的时钟使能位. 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bits 10-5	—	—
BS16T1EN	Bit 4	R/W	BS16T1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟

			1: 在睡眠模式下启动时钟
GP16C4T3EN	Bit 3	R/W	GP16C4T3 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GP16C4T2EN	Bit 2	R/W	GP16C4T2 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GP16C4T1EN	Bit 1	R/W	GP16C4T1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GP32C4T1EN	Bit 0	R/W	GP32C4T1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟

5.4.2.13 APB2 睡眠模式中的外设时钟使能寄存器 (RCU_APB2SL)

APB2 睡眠模式中的外设时钟使能寄存器 (RCU_APB2SL)																															
偏移地址: 30 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CMPEN		MCMEN		GP16C2T4EN	GP16C2T3EN	GP16C2T2EN	GP16C2T1EN		UART1EN		SPI1EN	AD16C4T1EN		ADCEN									

—	Bits 31-24	—	—
CMPEN	Bit 23	R/W	CMP 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bit 22	—	—
MCMEN	Bit 21	R/W	MCM 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bit 20	—	—
GP16C2T4EN	Bit 19	R/W	GP16C2T4 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GP16C2T3EN	Bit 18	R/W	GP16C2T3 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GP16C2T2EN	Bit 17	R/W	GP16C2T2 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
GP16C2T1EN	Bit 16	R/W	GP16C2T1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bit 15	—	—

UART1EN	Bit 14	R/W	UART1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bit 13	—	—
SPI1EN	Bit 12	R/W	SPI1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
AD16C4T1EN	Bit 11	R/W	AD16C4T1 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bit 10	—	—
ADCEN	Bit 9	R/W	ADC 睡眠模式中的时钟使能位 0: 在睡眠模式下禁用时钟 1: 在睡眠模式下启动时钟
—	Bits 8-0	—	—

5.4.2.14 AHB 深度睡眠模式中的外设时钟使能寄存器 (RCU_AHBDSL)

AHB 深度睡眠模式中的外设时钟使能寄存器 (RCU_AHBDSL)																																	
偏移地址: 34H																																	
复位值: 00000000_00000000_00000000_00000000 _b																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
													GPDEN	GPCEN	GPBEN	GPAEN	HDIVEN																

—	Bits 31-20	—	—
GPDEN	Bit 19	R/W	GPIOD 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GPCEN	Bit 18	R/W	GPIOC 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GPBEN	Bit 17	R/W	GPIOB 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GPAEN	Bit 16	R/W	GPIOA 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
HDIVEN	Bit 15	R/W	HDIV 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
—	Bits 14-7	—	—
RTCEN	Bit 6	R/W	RTC 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟

			0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
—	Bits 10-5	—	—
BS16T1EN	Bit 4	R/W	BS16T1 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GP16C4T3EN	Bit 3	R/W	GP16C4T3 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GP16C4T2EN	Bit 2	R/W	GP16C4T2 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GP16C4T1EN	Bit 1	R/W	GP16C4T1 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GP32C4T1EN	Bit 0	R/W	GP32C4T1 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟

5.4.2.16 APB2 深度睡眠模式中的外设时钟使能寄存器 (RCU_APB2DSL)

APB2 深度睡眠模式中的外设时钟使能寄存器 (RCU_APB2DSL)																																
偏移地址: 3C _H																																
复位值: 00000000_00000000_00000000_00000000 _b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
									CMPEN		MCMEN		GP16C2T4EN	GP16C2T3EN	GP16C2T2EN	GP16C2T1EN		UART1EN		SPI1EN	AD16C4T1EN		ADCEN									

—	Bits 31-24	—	—
CMPEN	Bit 23	R/W	CMP 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
—	Bit 22	—	—
MCMEN	Bit 21	R/W	MCM 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
—	Bit 20	—	—
GP16C2T4EN	Bit 19	R/W	GP16C2T4 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GP16C2T3EN	Bit 18	R/W	GP16C2T3 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
GP16C2T2EN	Bit 17	R/W	GP16C2T2 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟

			1: 在深度睡眠模式下启动时钟
GP16C2T1EN	Bit 16	R/W	GP16C2T1 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
—	Bit 15	—	—
UART1EN	Bit 14	R/W	UART1 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
—	Bit 13	—	—
SPI1EN	Bit 12	R/W	SPI1 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
AD16C4T1EN	Bit 11	R/W	AD16C4T1 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
—	Bit 10	—	—
ADCEN	Bit 9	R/W	ADC 深度睡眠模式中的时钟使能位 0: 在深度睡眠模式下禁用时钟 1: 在深度睡眠模式下启动时钟
—	Bits 8-0	—	—

第6章 闪存控制器 (FLASH)

6.1 概述

嵌入式闪存分为 64 KB 用来存取应用程序(Application Code)的主存储器区块与 8 KB 用来储存周边设定的讯息区块。此内存允许用户利用在线系统编程(ISP)或芯片下载程序对已焊接于 PCB 版上的芯片进行编程或擦除数据。

6.2 特性

- ◆ 64 KByte 程序区(Main Block)
- ◆ 8 KByte 讯息区(Info. Block)
- ◆ 每一个页(Page)的大小以 1 KB 为单位
- ◆ 支持程序区与讯息区块的编程、擦除以及读取的保护设定
- ◆ 支持以页为单位进行擦除
- ◆ 支持未保护区的全区擦除
- ◆ 支持睡眠模式下降低控制器的电流消耗
- ◆ 支持预取(Prefetch)功能
- ◆ 支持用户自行配置保护区
- ◆ 支持用户以页为单位配置保护区
- ◆ 支持用户配置多组不连续区间的保护区
- ◆ 当系统运行在 24 MHz 时可设定 0 个等待周期以及当系统运行在 48 MHz 时仅需设定 1 个等待周期的连续访问
- ◆ 支持最大可设定 3 个等待周期
- ◆ 支持以 4 KB 为单位的内存内部映射(Remap)

6.3 结构图

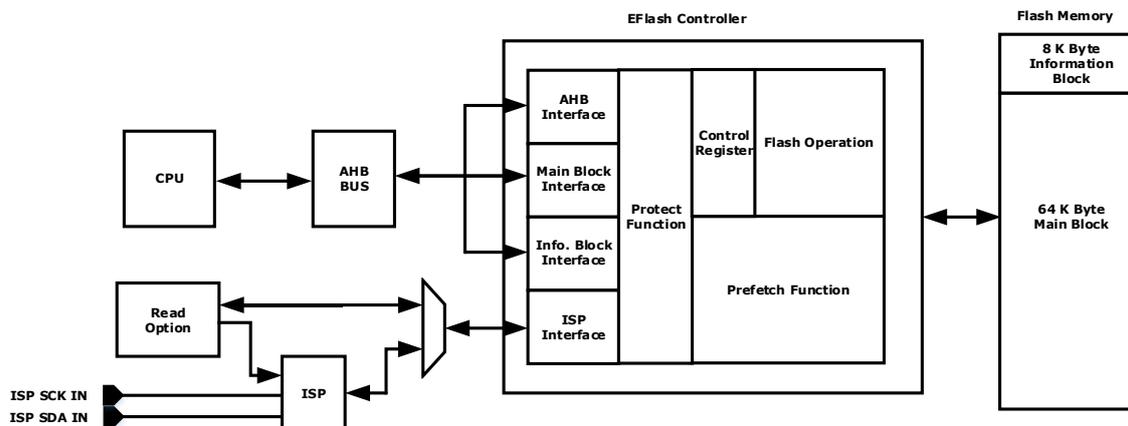


图 6-1 闪存控制器架构

闪存控制器的接口主要由 4 接口组成，分别为 AHB 接口、程序区接口、讯息区接口以及 ISP 接口。描述如下：

- ◆ AHB 接口：负责对闪存控制器控制缓存器进行写入与读取。
- ◆ 程序区接口：读取闪存程序区数据的接口。
- ◆ 讯息区接口：读取闪存讯息区数据的接口。
- ◆ ISP 接口：提供 ISP 与读取配置字节(Read Option Byte)逻辑存取闪存控制器与闪存的接口，此接口主要提供芯片出厂前的测试使用。

这 4 个接口的数据都会受到闪存保护逻辑的检测，以避免保护区内的数据被读出或是被修改。

6.4 功能描述

闪存由 2 个部分组成，分别为程序区 (Main Block) 以及讯息区 (Info. Block)。

- ◆ 程序区：大小为 64 KByte，内存位置为 0x0800_0000 至 0x0800_FFFF。共分为 64 个页(Page)，每个页的大小以 1 KByte 为单位。此区主要用来存取数据以及主程序。
- ◆ 信息区：大小为 8K Byte，内存位置为 0x1FFF_E000 至 0x1FFF_FFFF。共分为 8 个页(Page)，每个页的大小以 1K Byte 为单位。信息区共分为三个区块：
 - ◇ 系统闪存：位于信息区页 0 至页 5，主要存放芯片下载程序，此区无法藉由用户程序修改与读取，而芯片下载程序会在芯片出厂前下载至此区。系统开机完毕后默认会从此区的芯片下载程序开始执行，若用户希望跳过下载程序时，可藉由配置用户配置字内的硬件重映射选向来跳过下载程序。
 - ◇ 系统配置字：位于信息区页 6，要存放系统校准配置，此区会在 IC 出厂前针对每个 IC 的特性填入校准讯息与芯片讯息，故此区不开放给用户修改与读取。
 - ◇ 用户配置字：位于信息区页 7，主要存放程序区的保护设定与用户配置字，此区可由用户自行配置。

信息区页码		实体位置
系统闪存	页 0	0x1FFF_E000 ~ 0x1FFF_E3FF
	页 1	0x1FFF_E400 ~ 0x1FFF_E7FF
	页 2	0x1FFF_E800 ~ 0x1FFF_EBFF
	页 3	0x1FFF_EC00 ~ 0x1FFF_EFFF
	页 4	0x1FFF_F000 ~ 0x1FFF_F3FF
	页 5	0x1FFF_F400 ~ 0x1FFF_F7FF
系统配置字	页 6	0x1FFF_F800 ~ 0x1FFF_FBFF
用户配置字	页 7	0x1FFF_FC00 ~ 0x1FFF_FFFF

表 6-1 闪存信息区规划

6.4.1 用户配置字

信息区页 7 包含映射参考信息与程序区保护，用户可藉由程序修改此区的内容。

6.4.1.1 闪存保护字

闪存保护字主要存放于信息区页 7 的前 64 位(0x1C00 至 0x1C07)，每一位对应一个页，在设定保护时若将保护设定数值设定为 0，则代表相对应的页受到保护，反之则代表该页不受到保护。有关保护的配置方便与内容请参阅后续章节的描述。

配置字保护低位(PRTL)																															
偏移地址: 1C00H																															
复位值: 0xFFFF_FFFF																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTL<31:0>																															

PRTL	Bits 31-0	保护低 32 位 保护页面 0~第 31 页。当页面被保护时，相对位被设置为 0。
------	-----------	---

配置字保护高位(PRTH)																															
偏移地址: 1C04H																															
复位值: 0xFFFF_FFFF																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTH<31:0>																															

PRTH	Bits 31-0	保护高 32 位 保护页面 32~第 63 页。当页面被保护时，相对位被设置为 0。
------	-----------	--

6.4.2 闪存保护

闪存的保护逻辑可防止保护区域内的数据被读出或是被误修改。信息区依据不同区块有不同程度的保护，主要目的为防止系统校准信息被误擦除或是程序区的保护设定被修改，进而影响系统运行。

6.4.2.1 信息区保护

信息区总共 8 页，每一页分别受到不同的保护，如下所示：

- ◆ 页 0~页 5：存放芯片下载程序，此区不开放给用户修改与读取，芯片下载程序会在芯片出厂前刻录进此区。
- ◆ 页 6：存放系统校准信息，此区不开放给用户修改与读取。此区的校准信息会在出厂前针对每一颗芯片的特性进行配置，而这些校准信息会在系统开机时自动加载。
- ◆ 页 7：存放程序区保护设定与用户配置字，此页仅保护设定(0x1C00 ~ 0x1C07)在开启保护后无法重复编程，其余位置则不受读取/编程限制。若对此页进行擦除视为清除写保护设定，此时会同时擦除程序区与信息区页 7，因此页 7 内的用户选项字也会一并被擦除。

配置字节描述	描述	地址	数据
信息区页 0 ~ 5	芯片下载程序	0x0000 ~ 0x17FF	芯片出厂前配置
信息区页 6	系统配置字	0x1800 ~ 0x1BFF	芯片出厂前配置
信息区页 7	闪存保护低 32 位	0x1C00 ~ 0x1C03	用户定义
	闪存保护高 32 位	0x1C04 ~ 0x1C07	用户定义
	重映射配置 - SEFBASE	0x1C08(bit0 ~ bit3)	用户定义
	重映射配置 - SREMAP	0x1C08(bit4 ~ bit5)	用户定义
	硬件重映射 - HREMAP	0x1C09	用户定义
	保留	0x1C0A ~ 0x1FFF	保留

表 6-2 信息区页 6 保护区设定

6.4.2.2 程序区保护

保护的主要功能为防止保护区内的数据被 ISP、SWD、芯片下载程序或是非保护区的程序读出与修改，因此保护区内禁止保护区以外的程序以读取”数据”的方式进行读取但允许读取指令执行，同时为了防止保护区内的信息被覆盖或是被修改，因此保护区内禁止编程与抹除。当用户开启保护设定后系统会屏蔽 SWD，此时不再支持使用 SWD 进行除错与下载新的程序，若需要重新开启 SWD 的功能，此时仅能藉由用户代码内的流程或是芯片下载程序来清除保护设定，SWD 的功能会在保护设定被清除以后重新开启。

程序区的保护支持以页为单位进行配置，同时也支持不连续区间的保护，如下图所示。在设定保护时将保护设定数值设定为 0，则代表相对应的页受到保护，反之则代表该页不受到保护。以下图为例，保护设定为 0x000000FF_FFE00000 代表仅有页 21 至页 39 共 19 个页不受到保护，其余的 45 个页都受到保护

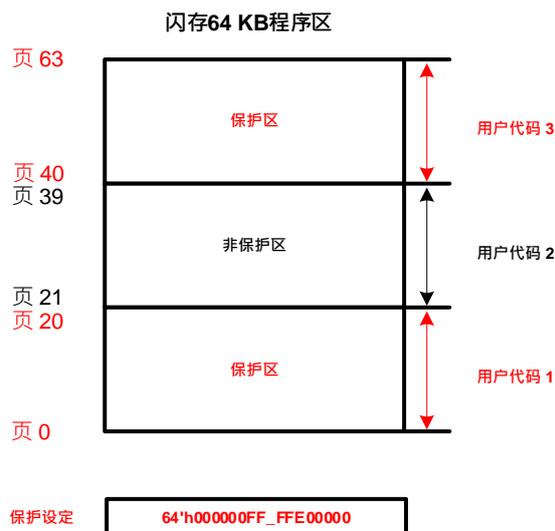


图 6-2 不连续区间保护示意图

开启保护流程如下，保护设定会在配置完毕后立即反映。

- ◆ 对 **FC_PA**(0x04)填入信息区页 7 的位置 0x0100_1C00。
- ◆ 对 **FC_PLD**(0x08)填入页 0 至页 31 的保护设定,相对应的位设定为 0 代表该页受到保护。
- ◆ 对 **FC_PHD**(0x0C)填入页 32 至页 63 的保护设定,相对应的位设定为 0 代表该页受到保护。
- ◆ 对 **FC_CMD**(0x00)写入 0xF0 配置程序区保护

开启保护以后，用户可通过下列 3 种方式检查是否成功开启保护功能

- ◆ 读取信息区页 7 的前 64 位(实体位置 0x1FFF_FC00 与 0x1FFF_FC04)。
- ◆ 读取 **FC_OP_PRTL** (0x58)与 **FC_OP_PPTH** (0x5C)，这 2 组寄存器分别纪录页 0 至页 31 与页 32 至页 63 的保护设定，寄存器中的数值为 0 代表相对应的页受到保护。
- ◆ 读取 **FC_STAT**(0x14)内的 MUNP1(位 0)，若数值为 0 代表已开启保护，若数值为 1 代表保护未成功开启。

当保护功能开启后是无法通过编程信息区页 7 修改保护的区域。若用户需要修改保护设定时可通过下列 3 种方式进行修改：

- ◆ **保护更新流程：**保护更新仅有保护区内的程序能够执行，在进行更新时会将程序区内未受到保护的区域以及所选择要清除的保护区完全清除，用户须于更新流程结束后通过重新上电或是通过执行 `NVIC_SystemReset()` 复位函数结束更新流程，更新流程结束后会立即反映新的保护设定。更新流程中被清除的区域会暂时解除保护设定，用户可于重启 CPU 前更新数据。有关更新保护的使用方法请参阅 6.4.7 更新保护章节内的描述。
- ◆ **清除保护流程：**在保护已被开启的状态下对信息区页 7 进行擦除视为清除保护设定，此时会将程序区与信息区页 7 完全擦除，用户需重新通过芯片下载程序或是 SWD 重新下载程序。
- ◆ **闪存全擦除：**用户可使用闪存全擦除指令将程序区与信息区页 7 一并擦除，擦除后用户需重新通过芯片下载程序或是 SWD 下载程序，有关全擦除的描述与使用方法请参阅 6.4.6 闪存全擦除。

综合上述，信息区与程序区的操作权限整理如下表所示。举例来说，当映射至程序区的未保护区时，无法对程序区内的保护区进行编程、页擦除与数据读取，但可读取指令；当映射至程序区的保护区时，无法对程序区内的保护区进行编程、页擦除，但可读取保护区内的数据与指令

			用户模式				调试模式
	闪存地址	闪存操作	映射于闪存		映射于 SRAM 执行	映射于系统闪存执行	SWD
			未保护区	保护区			
程序区	保护区	编程	X	X	X	X	X
		页擦除	X	X	X	X	X
		读指令	O	O	O	O	X
		读取数据	X	O	X	X	X
	未保护区	编程	O	O	O	O	O
		页擦除	O	O	O	O	O
		读指令	O	O	O	O	O
		读取数据	O	O	O	O	O
	闪存地址	闪存操作	映射于闪存		映射于 SRAM 执行	映射于系统闪存执行	SWD
			未保护区	保护区			
信息区	页 0~5	编程	X	X	X	X	X
		页擦除	X	X	X	X	X
		读	X	X	X	O	X
	页 6	编程	X	X	X	X	X
		页擦除	X	X	X	X	X
		读	X	X	X	X	X
	页 7	编程	保护设定以外区域可重复编程				
		页擦除	开启保护后视为清除保护				
		读	O	O	O	O	O

表 6-3 闪存保护区权限

6.4.3 预取

闪存控制器提供用户在闪存数据读取时的等待周期不为 0 时，可开启预取功能(Prefetch)。当开启预先读取功能后，闪存控制器会在适当时机提前抓取下一组数据，通过减少读取闪存的次数来提高系统执行效能。预取的功能默认是关闭的，用户须设定 **FC_CON(0x10)**的 PFEN(位 8) 为 1 来开启预取功能。须注意的是当 **FC_CON(0x10)**的 WAIT(位 0 至位 1)数值为 0 时，代表读取闪存数据时并不用进行等待而是单一读取周期即可完成读取闪存，此时开启预取功能并无法提升系统执行效能，因此预取的功能在此时是无效的。

6.4.4 闪存重映射

当用户选择从闪存的程序区开始执行程序时，可进一步设定闪存的内部映射。闪存内部映射是以 4 个页(4 KByte)为基准进行映射，可将程序区分为 16 个区块。用户可配置系统配置控制器内 **SYSCFG_REMAP(0x00)**寄存器的 EFBASE(位 4 至位 7)来决定要映射至哪一个区块开始执行。在进行配置时，用户需确保 **SYSCFG_REMAP(0x00)**寄存器的 MEMMOD (位 2 至位 3)的数值为 0。设定流程如下，以映射至程序区的第 3 个区块为例。

- ◆ 设定 **SYSCFG_REMAP(0x00)**寄存器的 MEMMOD (位 2 至位 3)的数值为 0。
- ◆ 设定 **SYSCFG_REMAP** 寄存器的 EFBASE(位 4 至位 7)为 0x2(代表映射至第 3 个 4 KByte)。
- ◆ 设定 **SYSCFG_REMAP(0x00)**寄存器的 REMAP (位 0)的数值为 1。
- ◆ 执行 NVIC_SystemReset()复位函数重置 CPU。
- ◆ 当 CPU 被重置以后，随即会从程序区位在 0x0800_2000 的位置开始读取(CPU 仍然是从 0x0000_0000 的位置开始读取)。

闪存程序区的内部映射只有在读取闪存的数据时才会进行映射，当进行闪存的编程以及擦除时不会受到内部映射影响。此外若使用闪存的实体位置来读取数据时，同样不会受到内部映射的影响。

Real ADDR	64K Byte Flash	After Remap(SC_REMAP.EFBASE=0x2)	
		Read With CPU ADDR	Read With Real ADDR
0x0800_F000	4K Byte(4 Page)	0x0000_D000	0x0800_F000
	⋮		
0x0800_4000	4K Byte(4 Page)	0x0000_2000	0x0800_4000
0x0800_3000	4K Byte(4 Page)	0x0000_1000	0x0800_3000
0x0800_2000	4K Byte(4 Page)	0x0000_0000	0x0800_2000
0x0800_1000	4K Byte(4 Page)	0x0000_F000	0x0800_1000
0x0800_0000	4K Byte(4 Page)	0x0000_E000	0x0800_0000

图 6-3 闪存读取位置对照

以上图为例，同样以映射至闪存的第 3 个区块(**SYSCFG_REMAP** 寄存器的 EFBASE 为 0x2)为例，当 CPU 读取位置为 0x0000_0000 的数据以及读取位置为 0x0800_2000 的资料时，会读取到同一个位置的资料。

6.4.5 闪存编程/擦除

闪存的编程以及擦除仅能通过闪存控制器进行。闪存的编程是以 64 位(Bit)为单位进行,而闪存擦除则是以页(1 K Byte)为单位进行。

闪存编程的流程如下,编程一组 64 位的数据共需 20 us。

- ◆ 对 **FC_PA(0x04)**内的 **PROG_ADDR**(位 0 至位 23)与 **PCNT**(位 25 至位 31)填入欲编程的位置以及编程的次数。总编程的次数为设定的编程次数加 1,每次编程完毕后编程位置会自动累加,最多可连续编程 128 次而不必重新填入新的编程位置。
- ◆ 若需要对信息区进行编程时需设定 **FC_PA(0x04)**内的 **IFREN**(位 24)为 1,否则需设定此位为 0。
- ◆ 对 **FC_PLD(0x08)**填入欲编程的低 32 位数据。
- ◆ 对 **FC_PHD(0x0C)**填入欲编程的高 32 位数据。
- ◆ 对 **FC_CMD(0x00)**填入编程指令 0x0000_00F0。
- ◆ 编程完成后会自动清除 **FC_CMD** 内的指令,但仍会保留 **FC_PLD** 与 **FC_PHD** 内的编程数据。若编程的次数已满足 **FC_PA(0x04)**内的 **PCNT**(位 25 至位 31)所设定的次数时会自动清除 **FC_PA**,否则会将 **FC_PA** 目前的位置加上 8。
- ◆ 重复对 **FC_PLD**、**FC_PHD** 与 **FC_CMD** 填入数据直到满足编程次数。

闪存擦除的流程如下,擦除一个页需耗时 2 ms。

- ◆ 对 **FC_PA(0x04)**填入欲擦除的位置,此位置需以页为单位。
- ◆ 若需要对信息区进行擦除时需设定 **FC_PA(0x04)**内的 **IFREN**(位 24)为 1,否则需设定此位为 0。
- ◆ 对 **FC_CMD(0x00)**填入页擦除指令 0x0000_00F1。

当擦除程序完成以后,会自动清除 **FC_PA** 内的位置以及 **FC_CMD** 内的指令。

用户通过闪存控制器所发出的编程或擦除指令仅能编程或擦除闪存内的未保护区,保护区内是禁止编程与擦除的。详细的闪存操作权限请参考 6.4.2 闪存保护。

6.4.6 闪存全擦除

闪存全擦除指令可以分为程序区全擦除以及未保护区全擦除。当进行程序区全擦除时,会将程序区与信息区页 7 完全擦除并重置保护设定。当进行未保护区全擦除时,则会将程序区未受到保护的区域完全擦除。

闪存全擦除的流程如下:

- ◆ 若要进行程序区全擦除则设定 **FC_PA(0x04)**内的 **IFREN**(位 24)为 1,若要进行为保护区全擦除则设定 **IFREN** 为 0。
- ◆ 对 **FC_CMD(0x00)**填入全擦除指令 0x0000_00F2。
- ◆ 当擦除程序完成以后,会自动清除 **FC_CMD(0x00)**内的指令。

当使用闪存全擦除时,擦除的时间须看有多少个未保护的页需要进行擦除,而每擦除一个页需耗时 2 ms,因此最多需耗时约 128 ms。

6.4.7 更新保护

闪存控制器提供用户在程序区开启保护时，通过保护区内的程序开启特定流程来更改保护设定以及保护区内的内容。在进行此流程前，用户必须先设定好要清除程序区哪些受到保护的页，以及在流程结束后新的保护设定。更新流程会先依照原先的保护设定，将程序区内未受到保护的区域以及所选择要清除的保护区完全清除，接着才会更新新的保护设定。在更新完保护设定后，会暂时移除被清除区域的保护设定，用户可在此期间更新数据。若要离开更新流程，可通过执行 `NVIC_SystemReset()` 复位函数重置 CPU、重新上电或是重新映射至程序区其他区域来结束更新流程，结束流程以后会加载新的保护设定。

详细更新流程如下：

- ◆ 对 `FC_PLD(0x08)` 与 `FC_PHD(0x0C)` 填入要清除哪些程序区受到保护的页，每一个位分别代表一个页，若写入 1 则会将相对应的页进行擦除。
- ◆ 对 `FC_UPL(0x20)` 与 `FC_UPH(0x24)` 填入新的保护设定。同样每一个位对应至一个页，若写入 1 则代表相对应的页不受保护。若用户未设定 `FC_UPL(0x20)` 与 `FC_UPH(0x24)` 即进行更新，则默认是将程序区未设定保护的区域全部开启保护。
- ◆ 对 `FC_CMD(0x00)` 写入 `0x0000_00F3` 开启更新流程。
- ◆ 更新流程结束后被清除的区域会暂时解除保护，用户可在离开更新程序前进行数据更新。

执行 `NVIC_SystemReset()` 复位函数重置 CPU、重新上电或是重新映射至程序区其他区域来结束更新流程，此后会加载新的保护设定。

6.4.8 识别码

闪存控制寄存器内存放 2 种识别码，分别为 3 组 32 位的芯片唯一码 (UID) 与一组 32 位的芯片产品识别码 (CHIPID)。

6.4.8.1 芯片唯一码

芯片唯一码共 96 位，用户可藉由读取闪存控制寄存器偏移地址 `0x34 ~ 0x3C` 的寄存器内容来获取芯片唯一码，每一颗芯片的识别码都是唯一的编码，用户可藉由此识别码实现下列功能：

- ◆ 终端产品序列号
- ◆ 通过特定的加密算法生成安全密钥

6.4.8.2 芯片产品识别码

芯片产品识别码共 32 位，用户可藉由读取闪存控制寄存器偏移地址 `0x40` 的寄存器内容来获取芯片产品识别码，此识别码可用于判断产品型号。

6.5 特殊功能寄存器

6.5.1 寄存器列表

FC 寄存器列表			
名称	偏移地址	类型	描述
FC_CMD	000 _H	RW	闪存控制命令寄存器
FC_PA	004 _H	RW	闪存控制编程地址寄存器
FC_PLD	008 _H	RW	闪存控制编程数据低位寄存器
FC_PHD	00C _H	RW	闪存控制编程数据高位寄存器
FC_CON	010 _H	RW	闪存控制寄存器
FC_STAT	014 _H	R	闪存控制状态寄存器
FC_UPL	020 _H	RW	闪存控制更新保护数据低位寄存器
FC_UPH	024 _H	RW	闪存控制更新保护数据高位寄存器
FC_OP_TRIM	030 _H	R	配置字节校准寄存器
FC_OP_UID0	034 _H	R	配置字节芯片唯一码 0
FC_OP_UID1	038 _H	R	配置字节芯片唯一码 1
FC_OP_UID2	03C _H	R	配置字节芯片唯一码 2
FC_OP_CHIPID	040 _H	R	配置字节芯片产品识别码
FC_OP_ADCTRIMV50	044 _H	R	配置字节 ADC 5V 校准值寄存器
FC_OP_ADCTRIMV33	048 _H	R	配置字节 ADC 3.3V 校准值寄存器
FC_OP_ADCTRIMV25	04C _H	R	配置字节 ADC 2.5V 校准值寄存器
FC_OP_PRTL	058 _H	R	配置字节保护低位寄存器
FC_OP_PRTH	05C _H	R	配置字节保护高位寄存器
FC_OP_REMAP	060 _H	R	配置字节重映射寄存器

6.5.2 寄存器描述

6.5.2.1 闪存控制命令寄存器(FC_CMD)

闪存控制命令寄存器 (FC_CMD)																																
偏移地址: 00 H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																												FC_CMD<7:0>				

—	Bits 31-8	—	—
FC_CMD	Bits 7-0	R/W	闪存命令 0xF0: 编程闪存 0xF1: 页擦除 0xF2: 全擦除 0xF3: 更新保护 其他: 保留 注: 程序完成后, 闪存命令将自行清除

6.5.2.2 闪存控制编程地址寄存器(FC_PA)

闪存控制编程地址寄存器 (FC_PA)																															
偏移地址: 04H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCNT<6:0>							IFREN	PROG_ADDR<23:0>																							

PCNT	Bits 31-25	R/W	程序计数器 提供最多128次连续程序。同时用户只需要填写FC_PLD, FC_PHD和FC_CMD
IFREN	Bit 24	R/W	讯息模块使能 0: 禁用讯息块访问 1: 启用讯息块访问
PROG_ADDR	Bits 23-0	R/W	编程/擦除地址 程序: PROG_ADDR [15:3]是双字地址 页擦除: PROG_ADDR [15:10]是页面地址, PROG_ADDR [9:0] 不需要在乎 全擦除: PROG_ADDR[15:0] 不需要在乎

6.5.2.3 闪存控制编程数据低位寄存器(FC_PLD)

闪存控制编程数据低位寄存器 (FC_PLD)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRGO_DATA_L<31:0>																															

PROG_DATA_L	Bits 31-0	R/W	编程数据低位 32 bits 闪存支持 64 位数据程序，该寄存器提供低 32 位数据
-------------	-----------	-----	---

6.5.2.4 闪存控制编程数据高位寄存器(FC_PHD)

闪存控制编程数据高位寄存器 (FC_PHD)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRGO_DATA_H<31:0>																															

PROG_DATA_H	Bits 31-0	R/W	编程数据高位 32 bits 支持 64 位数据程序，该寄存器提供高 32 位数据
-------------	-----------	-----	---

6.5.2.5 闪存控制寄存器 (FC_CON)

闪存控制寄存器 (FC_CON)																																
偏移地址: 10H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																							SLEEP	PFEN								WAIT<1:0>

—	Bits 31-10	—	—
SLEEP	Bit 9	R/W	<p>闪存睡眠控制</p> <p>当从CPU接收睡眠或深度睡眠时，如果选择此选项，控制器将屏蔽闪存控制信号</p> <p>0x0: 禁用闪存睡眠屏蔽功能</p> <p>0x1: 启用闪存睡眠屏蔽功能</p>
PFEN	Bit 8	R/W	<p>预取功能使能</p> <p>该功能允许用户在读取数据缓冲器中的最后32位数据时，从闪存中预取64位数据。</p> <p>0: 禁用预取功能</p> <p>1: 启用预取功能</p>
—	Bits 7-2	—	—
WAIT	Bits 1-0	R/W	<p>等待周期从嵌入式闪存读取数据</p> <p>该位的值指定是否需要等待状态来读取闪存，因为闪存控制器需要40 ns来获取有效数据。</p> <p>例如，如果我们使用48 MHz时钟(20.834ns)，那么我们必须将此寄存器设置为1（十进制）。</p> <p>如果我们的时钟周期超过40 ns，我们可以将该寄存器设置为0。</p>

6.5.2.7 闪存控制更新保护数据低位寄存器 (FC_UPL)

闪存控制更新保护数据低位寄存器(FC_UPL)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UPL<31:0>																															

UPL	Bits 31-0	R/W	更新保护低 32 位 更新程序区保护页面 0~第 31 页
-----	-----------	-----	----------------------------------

6.5.2.8 闪存控制更新保护数据高位寄存器 (FC_UPH)

闪存控制更新保护数据高位寄存器 (FC_UPH)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UPH<31:0>																															

UPH	Bits 31-0	R/W	更新保护高 32 位 更新程序区保护页面 32~第 63 页
-----	-----------	-----	-----------------------------------

6.5.2.9 选项字节校准寄存器 (FC_OP_TRIM)

选项字节校准寄存器 (FC_OP_TRIM)																															
偏移地址: 30H																															
复位值: 10100101_00000111_10100101_10000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BGTRIMVD<7:0>								BGTRIM<7:0>								HRCTRIMVD<7:0>								HRCTRIM<7:0>							

BGTRIMVD	Bits 31-24	R	BandGap 校准数值有效 0xA5: BandGap 校准数值是有效的 其他 : BandGap校准数值无效
BGTRIM	Bits 23-16	R	BandGap 校准数值 Bandgap校准参考值
HRCTRIMVD	Bits 15-8	R	HRC 校准值有效 0xA5: HRC 校准值是有效的 其他 : HRC 校准值无效
HRCTRIM	Bits 7-0	R	HRC 校准值 HRC校准参考值

6.5.2.10 选项字节芯片唯一码 0 (FC_OP_UID0)

选项字节芯片唯一码 0 (FC_OP_UID0)																															
偏移地址: 34H																															
复位值: -																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID0<31:0>																															

UID0	Bits 31-0	R	芯片唯一码 0
------	-----------	---	---------

6.5.2.11 选项字节芯片唯一码 1 (FC_OP_UID1)

选项字节芯片唯一码 1 (FC_OP_UID1)																															
偏移地址: 38H																															
复位值: -																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID1<31:0>																															

UID1	Bits 31-0	R	芯片唯一码 1
------	-----------	---	---------

6.5.2.12 选项字节芯片唯一码 2 (FC_OP_UID2)

选项字节芯片唯一码 2 (FC_OP_UID2)																															
偏移地址: 3CH																															
复位值: -																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID2<31:0>																															

UID2	Bits 31-0	R	芯片唯一码 2
------	-----------	---	---------

6.5.2.13 选项字节芯片产品识别码 (FC_OP_CHPID)

选项字节芯片产品识别码 (FC_OP_CHPID)																															
偏移地址: 40H																															
复位值: -																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHPID<31:0>																															

CHPID	Bits 31-0	R	芯片产品识别码 此识别码依据芯片型号而定
-------	-----------	---	-------------------------

6.5.2.14 配置字节 ADC 5V 校准值寄存器(FC_OP_ADCTRIMV50)

配置字节 ADC 5V 校准值寄存器(FC_OP_ADCTRIMV50)																															
偏移地址: 44 _H																															
复位值: 10100101_11111111_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCTRIMV50V D<7:0>								ADCTRIMV50P2 <11:0>														ADCTRIMV50P1 <11:0>									

ADCTRIMV50VD	Bits 31-24	R	ADC 5V 校准数值有效 0xA5: ADC 5V校准数值是有效的 其他: ADC 5V 校准数值无效
ADCTRIMV50P2	Bits 23-12	R	ADC 5V校准数值2 ADC 5V 校准数值第 2 个参数,使用条件为 VDD5 为 5V 的情况下 Vref 设定为 Bypass 模式(Vref 管脚连接至 VDD5 管脚,同时设定 SYSCFG_CFG 内的 VREFEN 位为 0)
ADCTRIMV50P1	Bits 11-0	R	ADC 5V校准数值1 ADC 5V 校准数值第 1 个参数,使用条件为 VDD5 为 5V 的情况下 Vref 设定为 Bypass 模式(Vref 管脚连接至 VDD5 管脚,同时设定 SYSCFG_CFG 内的 VREFEN 位为 0)

6.5.2.15 配置字节 ADC 3.3V 校准值寄存器 (FC_OP_ADCTRIMV33)

配置字节 ADC 3.3V 校准值寄存器 (FC_OP_ADCTRIMV33)																															
偏移地址: 48 _H																															
复位值: 10100101_11111111_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCTRIMV33VD <7:0>								ADCTRIMV33P2 <11:0>														ADCTRIMV33P1 <11:0>									

ADCTRIMV33VD	Bits 31-24	R	ADC 3.3V 校准数值有效 0xA5: ADC 3.3V校准数值是有效的 其他: ADC 3.3V 校准数值无效
ADCTRIMV33P2	Bits 23-12	R	ADC 3.3V校准数值2 ADC 3.3V 校准数值第 2 个参数, 使用条件为 VDD5 为 3.3V 的情况下 Vref 设定为 Bypass 模式 (Vref 管脚连接至 VDD5 管脚, 同时设定 SYSCFG_CFG 内的 VREFEN 位为 0)
ADCTRIMV33P1	Bits 11-0	R	ADC 3.3V校准数值1 ADC 3.3V 校准数值第 1 个参数, 使用条件为 VDD5 为 3.3V 的情况下 Vref 设定为 Bypass 模式 (Vref 管脚连接至 VDD5 管脚, 同时设定 SYSCFG_CFG 内的 VREFEN 位为 0)

6.5.2.16 配置字节 ADC 2.5V 校准值寄存器 (FC_OP_ADCTRIMV25)

配置字节 ADC 2.5V 校准值寄存器 (FC_OP_ADCTRIMV25)																															
偏移地址: 4C _H																															
复位值: 10100101_11111111_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCTRIMV25V D<7:0>								ADCTRIMV25V D<7:0>								ADCTRIMV25V D<7:0>															

ADCTRIMV25VD	Bits 31-24	R	ADC 2.5V 校准数值有效 0xA5: ADC 2.5V校准数值是有效的 其他: ADC 2.5V 校准数值无效
ADCTRIMV25P2	Bits 23-12	R	ADC 2.5V校准数值2 ADC 2.5V 校准数值第 2 个参数, 使用条件为 VDD5 为 5V 的情况下使用内部 Vref, 同时设定 Vref 电压值为 2.5V(Vref 管脚不可与 VDD5 管脚短接, 同时设定 SYSCFG_CFG 内的 VREFEN 位为 1, 设定 SYSCFG_CFG 内的 VRLS 位为 2)
ADCTRIMV25P1	Bits 11-0	R	ADC 2.5V校准数值1 ADC 2.5V 校准数值第 1 个参数, 使用条件为 VDD5 为 5V 的情况下使用内部 Vref, 同时设定 Vref 电压值为 2.5V(Vref 管脚不可与 VDD5 管脚短接, 同时设定 SYSCFG_CFG 内的 VREFEN 位为 1, 设定 SYSCFG_CFG 内的 VRLS 位为 2)

6.5.2.17 配置字节保护低位寄存器 (FC_OP_PRTL)

选项字节保护低位寄存器 (FC_OP_PRTL)																															
偏移地址: 58 _H																															
复位值: 11111111_11111111_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTL<31:0>																															

PRTL	Bits 31-0	R	保护低 32 位 保护页面 0~第 31 页。当页面被保护时, 相对位被设置为 0。
------	-----------	---	--

6.5.2.18 选项字节保护高位寄存器 (FC_OP_PRTH)

选项字节保护高位寄存器 (FC_OP_PRTH)																															
偏移地址: 5C _H																															
复位值: 11111111_11111111_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTH<31:0 ^Δ																															

PRTH	Bits 31-0	R	保护高32位 保护页面32~第63页。当页面被保护时，相对位被设置为0。
------	-----------	---	--

6.5.2.19 选项字节重映射寄存器 (FC_OP_REMAP)

选项字节重映射寄存器 (FC_OP_REMAP)																															
偏移地址: 60 _H																															
复位值: 00000000_00000000_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																HREMAP<7:0 ^Δ							SREMAP<1:0 ^Δ		SEFBASE<3:0 ^Δ						

—	Bits 31-16	—	—
HREMAP	Bits 15-8	R	ISP 硬件重映射选项 0xA5: 开机时跳过 Bootloader 并映射到主闪存 其他: 开机时映射到 Bootloader
—	Bits 7-6	—	—
SREMAP	Bits 5-4	R	软件重映射选项 0x0: 主闪存映射到0x0000 0000 0x1: 系统闪存映射到0x0000 0000 0x2: SRAM映射到0x0000 0000 0x3: 保留 注: 数值仅供参考, 不影响硬件
SEFBASE	Bits 3-0	R	重映射主闪存基地址选择 如果设置0x1, 则意味着第二个4 KByte, 如果设置为0x2, 则意味着第三个4 KByte, 依此类推。 注: 数值仅供参考, 不影响硬件

7.4 功能描述

根据数据表中列出的每个 I/O 端口的具体硬件特性，通用 I/O(GPIO)端口的每个端口位可以通过软件以几种模式单独配置：

- ◆ 浮空输入
- ◆ 上拉输入
- ◆ 下拉输入
- ◆ 模拟输入
- ◆ 具有上拉或下拉能力的开漏输出
- ◆ 具有上拉或下拉能力的推挽输出
- ◆ 复用功能且具有上拉或下拉能力的推挽输出
- ◆ 复用功能且具有上拉或下拉能力的开漏输出

每个 I/O 端口位可以自由编程，然而 I/O 端口寄存器可按 32 位字，半字或字节访问。**GPIOx_BSBR** 寄存器允许对任何 GPIO 寄存器进行位读 / 改写访问。这种情况下，在读和更改访问之间产生 IRQ 时也不会发生危险。

7.4.1 通用 I/O (GPIO)

复位期间和刚复位后，复用功能未开启且所有的 I/O 端口被配置为模拟功能。

当作为输出配置时，写到输出数据寄存器 (**GPIOx_ODATA**) 的值输出到相应的引脚上。可以以推挽模式或开漏模式（仅低电平被驱动，高电平表现为高阻）使用输出驱动器。

输入数据寄存器 (**GPIOx_IDATA**) 在每个 AHB 时钟周期捕捉 I/O 引脚上的数据。所有 GPIO 引脚都有一个内部弱上拉和弱下拉电阻，它们被激活或断开有赖 **GPIOx_PUD** 寄存器的值。

7.4.2 I/O 端口控制寄存器

每个 GPIO 口都有 3 个 32 位的控制寄存器 (**GPIOx_MOD**, **GPIOx_OT**, **GPIOx_PUD**) 用来配置多达 16 I/O 口线。**GPIOx_MOD** 寄存器用来选择 I/O 模式（如输入，输出、复用或模拟）。**GPIOx_OT** 寄存器用来选择输出类型（如推挽或开漏）。**GPIOx_PUD** 寄存器用来选择上拉 / 下拉方式。

7.4.3 I/O 端口数据寄存器

每个 GPIO 口有两个 16 位数据寄存器：输入和输出数据寄存器 (**GPIOx_IDATA** 和 **GPIOx_ODATA**)。 **GPIOx_ODATA** 用于存储输出数据，其可进行读 / 写访问。从 I/O 口线的输入数据存放在 (**GPIOx_IDATA**) 寄存器中，该寄存器为只读寄存器。

7.4.4 I/O 数据位操作

位置复位寄存器 (**GPIOx_BSR**) 是一个 32 位寄存器，其允许应用对输出数据寄存器 (**GPIOx_ODATA**) 的每个位进行置位和复位操作。位置位和复位寄存器的有效数据宽度是 **GPIOx_ODATA** 有效数据宽度的两倍。

对于 **GPIOx_ODATA** 中的每位，在 **GPIOx_BSR** 中有两位与之对应：BS(i) 和 BR(i)。当对位 BS(i) 写 1 时则设置相应的 OD(i) 位。当对 BR(i) 写 1 时，则复位相应的 OD(i) 位。对 **GPIOx_BSR** 中的任意位写 0 都不会影响 **GPIOx_ODATA** 寄存器的值。若对 **GPIOx_BSR** 的 BS(i) 和 BR(i) 同时置 1，那么其置位操作具有优先权（即对相应位做置位操作）。

7.4.5 GPIO 锁定机制

通过将特定的写序列应用到 **GPIOx_LCK** 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 **GPIOx_MOD**、**GPIOx_OT**、**GPIOx_PUD**、**GPIOx_AFL** 和 **GPIOx_AFH**。

为了写 **GPIOx_LCK** 寄存器，须发出一个特定的写 / 读序列。当正确的锁定序列作用于这个寄存器的位 16 时，LCK[15:0] 的值用来锁定 I/O 口的配置（在写序列期间要保持 LCK[15:0] 值不变）。当锁定序列已经作用于一个端口位，该端口位的值再也不能改变直到下一次复位。

7.5 特殊功能寄存器

7.5.1 寄存器列表

外设寄存器可支持半字（16 位）或字（32 位）访问。

GPIO 寄存器列表			
名称	偏移地址	类型	描述
GPIOx_IDATA	0000 _H	R	GPIOx 端口输入数据寄存器
GPIOx_ODATA	0004 _H	R/W	GPIOx 端口输出数据寄存器
GPIOx_BSBR	0008 _H	W1	GPIOx 端口置位和复位寄存器
GPIOx_LCK	000C _H	R/W	GPIOx 端口锁定寄存器
GPIOx_MOD	0010 _H	R/W	GPIOx 端口模式寄存器
GPIOx_PUD	0014 _H	R/W	GPIOx 端口上拉和下拉寄存器
GPIOx_OT	0018 _H	R/W	GPIOx 端口输出类型寄存器
GPIOx_DS	001C _H	R/W	GPIOx 端口输出驱动寄存器
GPIOx_FIR	0020 _H	R/W	GPIOx 端口滤波寄存器
GPIOx_IST	0024 _H	R/W	GPIOx 端口输入类型寄存器
GPIOx_AFL	0028 _H	R/W	GPIOx 复用功能低位寄存器
GPIOx_AFH	002C _H	R/W	GPIOx 复用功能高位寄存器

7.5.2 寄存器描述

7.5.2.1 GPIOx 端口输入数据寄存器 (GPIOx_IDATA)

GPIOx 端口输入数据寄存器 (GPIOx_IDATA)																															
偏移地址: 00H																															
复位值: 00000000_00000000_00000000_00000000B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

—	Bits 31-16	—	—
ID15	Bit 15	R	ID_y: 端口输入数据 (y = 0...15) 这些位只读。它们包含相应I/O口的输入值。
ID14	Bit 14	R	
ID13	Bit 13	R	
ID12	Bit 12	R	
ID11	Bit 11	R	
ID10	Bit 10	R	
ID9	Bit 9	R	
ID8	Bit 8	R	
ID7	Bit 7	R	
ID6	Bit 6	R	
ID5	Bit 5	R	
ID4	Bit 4	R	
ID3	Bit 3	R	
ID2	Bit 2	R	
ID1	Bit 1	R	
ID0	Bit 0	R	

7.5.2.2 GPIOx 端口输出数据寄存器(GPIOx_ODATA)

GPIOx 端口输出数据寄存器 (GPIOx_ODATA)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

—	Bits 31-16	—	—
OD15	Bit 15	R/W	<p>ODy: 端口输出数据 (y = 0..15) 这些位可由软件读写。 注：对于单独位的设置/清除，可单独对 GPIOx_BS或GPIOx_BR 寄存器操作来实现。</p>
OD14	Bit 14	R/W	
OD13	Bit 13	R/W	
OD12	Bit 12	R/W	
OD11	Bit 11	R/W	
OD10	Bit 10	R/W	
OD9	Bit 9	R/W	
OD8	Bit 8	R/W	
OD7	Bit 7	R/W	
OD6	Bit 6	R/W	
OD5	Bit 5	R/W	
OD4	Bit 4	R/W	
OD3	Bit 3	R/W	
OD2	Bit 2	R/W	
OD1	Bit 1	R/W	
OD0	Bit 0	R/W	

7.5.2.3 GPIOx 端口置位和复位寄存器 (GPIOx_BSBR)

GPIOx 端口置位和复位寄存器(GPIOx_BSBR)																															
偏移地址: 08H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0

BR15	Bit 31	W1	<p>BRy: 端口 x 复位位 y (y= 0..15)</p> <p>这些位只写。读这些位时返回0x0000数值。</p> <p>0: 对相应的 ODx 位无影响</p> <p>1: 复位相应的 ODx 位</p>
BR14	Bit 30	W1	
BR13	Bit 29	W1	
BR12	Bit 28	W1	
BR11	Bit 27	W1	
BR10	Bit 26	W1	
BR9	Bit 25	W1	
BR8	Bit 24	W1	
BR7	Bit 23	W1	
BR6	Bit 22	W1	
BR5	Bit 21	W1	
BR4	Bit 20	W1	
BR3	Bit 19	W1	
BR2	Bit 18	W1	
BR1	Bit 17	W1	
BR0	Bit 16	W1	
BS15	Bit 15	W1	<p>BSy: 端口 x 设置位 y (y= 0..15)</p> <p>这些位只写。读这些位时返回 0x0000 数值。</p> <p>0: 对相应的 ODx 位无影响</p> <p>1: 置位相应的 ODx 位</p>
BS14	Bit 14	W1	
BS13	Bit 13	W1	
BS12	Bit 12	W1	
BS11	Bit 11	W1	
BS10	Bit 10	W1	
BS9	Bit 9	W1	
BS8	Bit 8	W1	
BS7	Bit 7	W1	
BS6	Bit 6	W1	
BS5	Bit 5	W1	
BS4	Bit 4	W1	
BS3	Bit 3	W1	
BS2	Bit 2	W1	
BS1	Bit 1	W1	
BS0	Bit 0	W1	

7.5.2.4 GPIOx 端口锁定寄存器 (GPIOx_LCK)

GPIOx 端口锁定寄存器(GPIOx_LCK)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0

—	Bits 31-17	—	—
LCKK	Bit 16	R/W	<p>LCKK: 锁定键 该位可随时读取。它仅能由锁键写序列来改写。 0: 端口配置锁定键不激活。 1: 端口配置锁定键激活。GPIOx_LCK 寄存器锁定直到下一个 MCU 复位产生 锁定键写序列: WR GPIOx_LCK = (~LCK[15:0]<<16) + LCK[15:0] WR GPIOx_LCK = (~LCK[15:0]<<16) + LCK[15:0] RD GPIOx_LCK RD GPIOx_LCK[16] = '1' (这个读操作可选, 但其为确认锁定是否激活)</p>
LCK15	Bit 15	R/W	<p>LCKy: Port x lock bit y (y= 0..15) 这些位可读 / 写, 但仅 LCKK 为 '0' 时写。冻结的寄存器包括 GPIOx_MOD, GPIOx_PUD, GPIOx_OT, GPIOx_DS, GPIOx_FIR, GPIOx_IST, GPIOx_AFL and GPIOx_AFH. 0: 端口配置未锁定 1: 端口配置锁定</p>
LCK14	Bit 14	R/W	
LCK13	Bit 13	R/W	
LCK12	Bit 12	R/W	
LCK11	Bit 11	R/W	
LCK10	Bit 10	R/W	
LCK9	Bit 9	R/W	
LCK8	Bit 8	R/W	
LCK7	Bit 7	R/W	
LCK6	Bit 6	R/W	
LCK5	Bit 5	R/W	
LCK4	Bit 4	R/W	
LCK3	Bit 3	R/W	
LCK2	Bit 2	R/W	
LCK1	Bit 1	R/W	
LCK0	Bit 0	R/W	

7.5.2.5 GPIOx 端口模式寄存器 (GPIOx_MOD)

GPIOx 端口模式寄存器(GPIOx_MOD)																															
偏移地址: 10 _H																															
复位值: 11111111_11111111_11111111_11111111 _B (GPIOA: 11101011_11111111_11111111_11111111 _B)																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOD15<1:0>		MOD14<1:0>		MOD13<1:0>		MOD12<1:0>		MOD11<1:0>		MOD10<1:0>		MOD9<1:0>		MOD8<1:0>		MOD7<1:0>		MOD6<1:0>		MOD5<1:0>		MOD4<1:0>		MOD3<1:0>		MOD2<1:0>		MOD1<1:0>		MOD0<1:0>	

MOD15<1:0>	Bits 31-30	R/W	MODy[1:0]: 端口 x 配置位(y = 0..15) 这些位可由软件写来配置 I/O 口模式。 00: 输入模式 01: 通用输出模式 10: 复用功能模式 11: 模拟模式(复位状态)
MOD14<1:0>	Bits 29-28	R/W	
MOD13<1:0>	Bits 27-26	R/W	
MOD12<1:0>	Bits 25-24	R/W	
MOD11<1:0>	Bits 23-22	R/W	
MOD10<1:0>	Bits 21-20	R/W	
MOD9<1:0>	Bits 19-18	R/W	
MOD8<1:0>	Bits 17-16	R/W	
MOD7<1:0>	Bits 15-14	R/W	
MOD6<1:0>	Bits 13-12	R/W	
MOD5<1:0>	Bits 11-10	R/W	
MOD4<1:0>	Bits 9-8	R/W	
MOD3<1:0>	Bits 7-6	R/W	
MOD2<1:0>	Bits 5-4	R/W	
MOD1<1:0>	Bits 3-2	R/W	
MOD0<1:0>	Bits 1-0	R/W	

7.5.2.6 GPIOx 端口上拉和下拉寄存器 (GPIOx_PUD)

GPIOx 端口上拉和下拉寄存器(GPIOx_PUD)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B (GPIOA: 00100100_00000000_00000000 _B)																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD15<1:0>		PUD14<1:0>		PUD13<1:0>		PUD12<1:0>		PUD11<1:0>		PUD10<1:0>		PUD9<1:0>		PUD8<1:0>		PUD7<1:0>		PUD6<1:0>		PUD5<1:0>		PUD4<1:0>		PUD3<1:0>		PUD2<1:0>		PUD1<1:0>		PUD0<1:0>	

PUD15<1:0>	Bits 31-30	R/W	PUDy[1:0]: 端口 x 配置位(y = 0..15) 这些位由软件写来配置 I/O 口的上拉或下拉。 00: 无上拉和下拉 (复位状态) 01: 上拉 10: 下拉 11: 保留
PUD14<1:0>	Bits 29-28	R/W	
PUD13<1:0>	Bits 27-26	R/W	
PUD12<1:0>	Bits 25-24	R/W	
PUD11<1:0>	Bits 23-22	R/W	
PUD10<1:0>	Bits 21-20	R/W	
PUD9<1:0>	Bits 19-18	R/W	
PUD8<1:0>	Bits 17-16	R/W	
PUD7<1:0>	Bits 15-14	R/W	
PUD6<1:0>	Bits 13-12	R/W	
PUD5<1:0>	Bits 11-10	R/W	
PUD4<1:0>	Bits 9-8	R/W	
PUD3<1:0>	Bits 7-6	R/W	
PUD2<1:0>	Bits 5-4	R/W	
PUD1<1:0>	Bits 3-2	R/W	
PUD0<1:0>	Bits 1-0	R/W	

7.5.2.7 GPIOx 端口输出类型寄存器 (GPIOx_OT)

GPIOx 端口输出类型寄存器(GPIOx_OT)																																
偏移地址: 18H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0

—	Bits 31-16	—	—
OT15	Bit 15	R/W	<p>OTy: 端口 x 配置位(y = 0..15) 这些位可由软件写来配置 I/O 口的输出类型。 0: 推挽输出(复位状态) 1: 开漏输出</p>
OT14	Bit 14	R/W	
OT13	Bit 13	R/W	
OT12	Bit 12	R/W	
OT11	Bit 11	R/W	
OT10	Bit 10	R/W	
OT9	Bit 9	R/W	
OT8	Bit 8	R/W	
OT7	Bit 7	R/W	
OT6	Bit 6	R/W	
OT5	Bit 5	R/W	
OT4	Bit 4	R/W	
OT3	Bit 3	R/W	
OT2	Bit 2	R/W	
OT1	Bit 1	R/W	
OT0	Bit 0	R/W	

7.5.2.8 GPIOx 端口输出驱动寄存器 (GPIOx_DS)

GPIOx 端口输出驱动寄存器(GPIOx_DS)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DS15	DS14	DS13	DS12	DS11	DS10	DS9	DS8	DS7	DS6	DS5	DS4	DS3	DS2	DS1	DS0

—	Bits 31-16	—	—
DS15	Bit 15	R/W	<p>DSy: 端口 x 配置位(y = 0..15) 这些位可由软件写来配置 I/O 口的输出驱动电流。 0: 典型8 mA (复位状态) 1: 典型16 mA</p>
DS14	Bit 14	R/W	
DS13	Bit 13	R/W	
DS12	Bit 12	R/W	
DS11	Bit 11	R/W	
DS10	Bit 10	R/W	
DS9	Bit 9	R/W	
DS8	Bit 8	R/W	
DS7	Bit 7	R/W	
DS6	Bit 6	R/W	
DS5	Bit 5	R/W	
DS4	Bit 4	R/W	
DS3	Bit 3	R/W	
DS2	Bit 2	R/W	
DS1	Bit 1	R/W	
DS0	Bit 0	R/W	

7.5.2.9 GPIOx 端口滤波寄存器 (GPIOx_FIR)

GPIOx 端口滤波寄存器 (GPIOx_FIR)																															
偏移地址: 20H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																FIR15	FIR14	FIR13	FIR12	FIR11	FIR10	FIR9	FIR8	FIR7	FIR6	FIR5	FIR4	FIR3	FIR2	FIR1	FIR0

—	Bits 31-16	—	—
FIR15	Bit 15	R/W	FIRy: 端口 x 配置位(y = 0..15) 这些位可由软件写来配置输入是否要通过滤波 0: 旁路 (复位状态) 1: 消除一定脉冲宽度的毛刺 (20 ns)
FIR14	Bit 14	R/W	
FIR13	Bit 13	R/W	
FIR12	Bit 12	R/W	
FIR11	Bit 11	R/W	
FIR10	Bit 10	R/W	
FIR9	Bit 9	R/W	
FIR8	Bit 8	R/W	
FIR7	Bit 7	R/W	
FIR6	Bit 6	R/W	
FIR5	Bit 5	R/W	
FIR4	Bit 4	R/W	
FIR3	Bit 3	R/W	
FIR2	Bit 2	R/W	
FIR1	Bit 1	R/W	
FIR0	Bit 0	R/W	

7.5.2.10 GPIOx 端口输入类型寄存器 (GPIOx_IST)

GPIOx 端口输入类型寄存器(GPIOx_IST)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																IST15	IST14	IST13	IST12	IST11	IST10	IST9	IST8	IST7	IST6	IST5	IST4	IST3	IST2	IST1	IST0

—	Bits 31-16	—	—
IST15	Bit 15	R/W	ISTy: 端口 x 配置位 (y = 0..15) 这些位可由软件写来配置输入Schmitt触发器 0: TTL IO电平 ($V_{T+} = 2.0v, V_{T-} = 0.8v$) 1: CMOS IO电平 ($V_{T+} = 3.1v, V_{T-} = 1.5v$)
IST14	Bit 14	R/W	
IST13	Bit 13	R/W	
IST12	Bit 12	R/W	
IST11	Bit 11	R/W	
IST10	Bit 10	R/W	
IST9	Bit 9	R/W	
IST8	Bit 8	R/W	
IST7	Bit 7	R/W	
IST6	Bit 6	R/W	
IST5	Bit 5	R/W	
IST4	Bit 4	R/W	
IST3	Bit 3	R/W	
IST2	Bit 2	R/W	
IST1	Bit 1	R/W	
IST0	Bit 0	R/W	

7.5.2.11 GPIOx 复用功能低位寄存器 (GPIOx_AFL)

GPIOx 复用功能低位寄存器(GPIOx_AFL)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AF7<3:0>				AF6<3:0>				AF5<3:0>				AF4<3:0>				AF3<3:0>				AF2<3:0>				AF1<3:0>				AF0<3:0>			

AF7<3:0>	Bits 31-28	R/W	AFy[3:0]: 端口 x 位 y 的复用功能选择(y = 0..7) 这些位通过软件写入, 用于配置复用功能 I/O。 AFy 选择: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000:保留 1001:保留 1010:保留 1011:保留 1100:保留 1101:保留 1110:保留 1111:保留
AF6<3:0>	Bits 27-24	R/W	
AF5<3:0>	Bits 23-20	R/W	
AF4<3:0>	Bits 19-16	R/W	
AF3<3:0>	Bits 15-12	R/W	
AF2<3:0>	Bits 11-8	R/W	
AF1<3:0>	Bits 7-4	R/W	
AF0<3:0>	Bits 3-0	R/W	

7.5.2.12 GPIOx 复用功能高位寄存器 (GPIOx_AFH)

GPIOx 复用功能高位寄存器(GPIOx_AFH)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AF15<3:0>				AF14<3:0>				AF13<3:0>				AF12<3:0>				AF11<3:0>				AF10<3:0>				AF9<3:0>				AF8<3:0>			

AF15<3:0>	Bits 31-28	R/W	AFy[3:0]: 端口 x 位 y 的复用功能选择(y = 8..15) 这些位通过软件写入, 用于配置复用功能 I/O。 AFy 选择: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000:保留 1001:保留 1010:保留 1011:保留 1100:保留 1101:保留 1110:保留 1111:保留
AF14<3:0>	Bits 27-24	R/W	
AF13<3:0>	Bits 23-20	R/W	
AF12<3:0>	Bits 19-16	R/W	
AF11<3:0>	Bits 15-12	R/W	
AF10<3:0>	Bits 11-8	R/W	
AF9<3:0>	Bits 7-4	R/W	
AF8<3:0>	Bits 3-0	R/W	

第8章 直接存储器访问控制器 (DMA)

8.1 概述

直接存储器访问控制器包括一个外设直接存储器存取(DMA) 控制器。DMA 控制器可以通过它的六个通道来实现在存储器和外设之间的数据传输（外设到存储器，存储器到外设或者存储器到存储器）。

这里为每一个 DMA 通道-CH[5:0]，提供一个用作外围 APB 设备和内存之间通信的传输缓冲区。CPU 可以通过软件轮询或者中断方式来识别 DMA 操作的完成。

8.2 特性

- ◆ 支持六个 DMA 通道。每个通道支持单一方向的传输
- ◆ AHB 主/从接口兼容，用于数据传输和寄存器的读/写
- ◆ 每个通道可以使用四个字节深度的 FIFO 模式或直接模式。
 - ◇ 用于 FIFO 模式：可以设定突发模式
 - ◇ 用于直接模式：当配置在直接模式时(FIFO 禁用)
- ◆ 支持源地址和目的地址的增加模式或固定模式。
- ◆ 独立的源和目的数据宽度（字节，半字，字）：当源和目的地的数据宽度不相等，DMA 自动打包/解包所需的传输以优化带宽。此功能可用于 FIFO 模式
- ◆ 传输的数据量可以通过 DMA 流量控制器或者外设流量控制器控制：
 - ◇ DMA 流量控制器：要传输的数据量可编程(从 1 到 65535)
 - ◇ 外设流量控制器：要传输的数据量未知，通过源或目的的外设来控制传输数据量
- ◆ 硬件通道优先级。DMA 信道 0 具有最高优先级和通道 5 具有最低优先级。
- ◆ 软件通道优先级。设置软件通道优先级，当两个 DMA 通道设置为相同的优先级时，它将根据硬件通道优先顺序被依次传输。

8.3 结构图

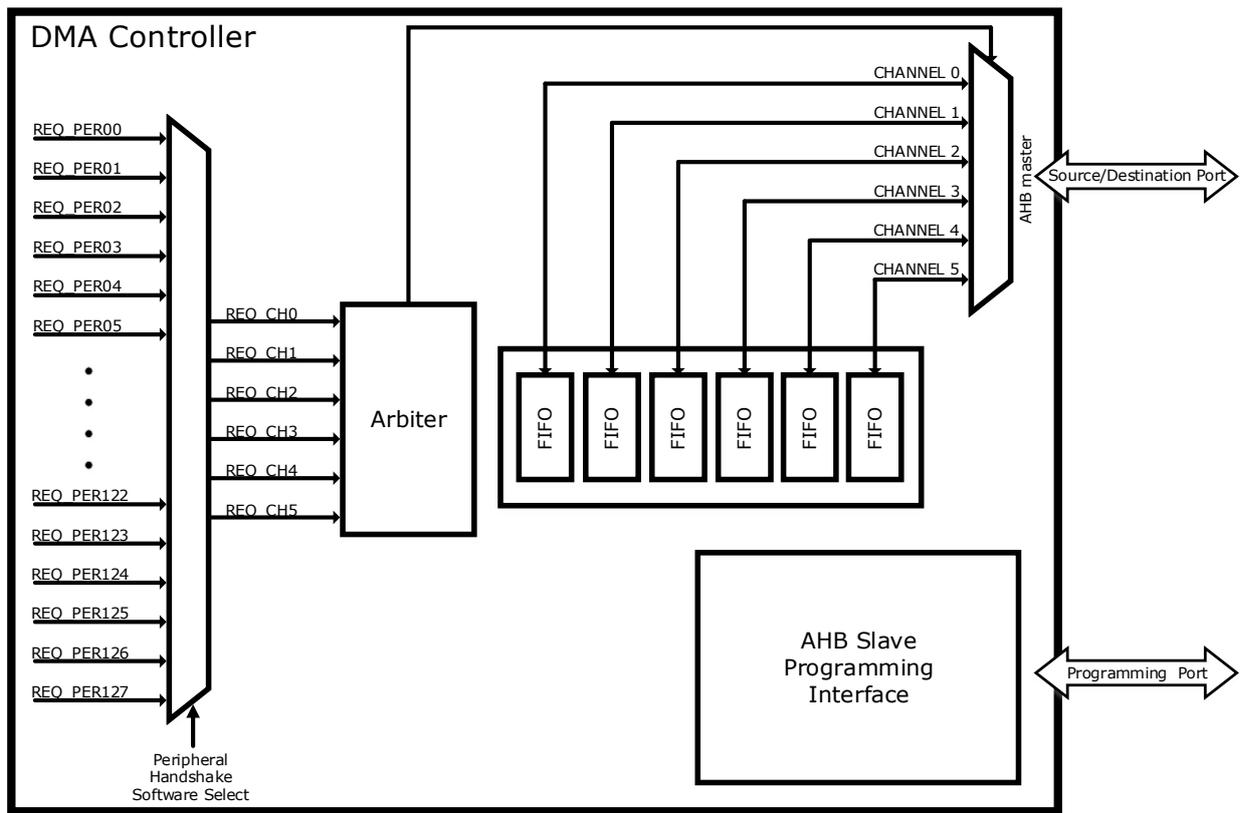


图 8-1 直接存储器访问控制器框图

8.4 功能描述

8.4.1 直接存储器访问控制器事务

DMA 事务由给定数量的数据传输的序列组成。要传输的数据项数及其宽度（8 位，16 位或 32 位）是可编程的。

每个 DMA 传输由三个操作组成：

- ◆ 通过 **DMA_SARx** 寄存器从外设数据寄存器或存储器中的位置加载
- ◆ 通过 **DMA_DARx** 寄存器从外设数据寄存器或存储器中的位置加载
- ◆ **DMA_BCRx** 寄存器的递减，其中包含仍需执行的事务数

事件发生后，外设向 DMA 控制器发送请求信号。DMA 控制器根据通道优先级响应请求。一旦 DMA 控制器访问外设，DMA 控制器就会向外设发送应答信号。外设从 DMA 控制器获取应答信号后，外设就会释放其请求。一旦该外围设备解除了该请求，DMA 控制器将释放应答信号。如果有更多请求，外设可以启动下一个事务。

8.4.2 外设通道选择

每个外设可以在 6 个通道请求中选择的 DMA 请求相关联。该选择由 **DMA_CSRx** 寄存器中的 PHSS [6:0]控制。

请求号	外围设备	请求号	外围设备	请求号	外围设备	请求号	外围设备
0	UART1_TX	32	BS16T1_UP	64	GP16C2T1_CH1	96	
1	UART2_TX	33	AD16C4T1_CH1	65	GP16C2T1_CH2	97	
2	UART3_TX	34	AD16C4T1_CH2	66	GP16C2T1_UP	98	
3	SUART1_TX	35	AD16C4T1_CH3	67	GP16C2T1_TRIG	99	
4	SUART2_TX	36	AD16C4T1_CH4	68	GP16C2T1_COM	100	
5	SPI1_TX	37	AD16C4T1_UP	69		101	
6	SPI2_TX	38	AD16C4T1_TRIG	70	GP16C2T2_CH1	102	
7	I2C1_TX	39	AD16C4T1_COM	71	GP16C2T2_CH2	103	
8	I2C2_TX	40	GP32C4T1_CH1	72	GP16C2T2_UP	104	
9		41	GP32C4T1_CH2	73	GP16C2T2_TRIG	105	
10	DAC	42	GP32C4T1_CH3	74	GP16C2T2_COM	106	
11		43	GP32C4T1_CH4	75		107	
12		44	GP32C4T1_UP	76	GP16C2T3_CH1	108	
13		45	GP32C4T1_TRIG	77	GP16C2T3_CH2	109	
14		46	GP16C4T1_CH1	78	GP16C2T3_UP	110	
15	UART1_RX	47	GP16C4T1_CH2	79	GP16C2T3_TRIG	111	
16	UART2_RX	48	GP16C4T1_CH3	80	GP16C2T3_COM	112	
17	UART3_RX	49	GP16C4T1_CH4	81		113	
18	SUART1_RX	50	GP16C4T1_UP	82	GP16C2T4_CH1	114	
19	SUART2_RX	51	GP16C4T1_TRIG	83	GP16C2T4_CH2	115	
20	SPI1_RTX	52	GP16C4T2_CH1	84	GP16C2T4_UP	116	
21	SPI2_RX	53	GP16C4T2_CH2	85	GP16C2T4_TRIG	117	
22	I2C1_RX	54	GP16C4T2_CH3	86	GP16C2T4_COM	118	
23	I2C2_RX	55	GP16C4T2_CH4	87		119	
24		56	GP16C4T2_UP	88		120	
25	ADCSS0	57	GP16C4T2_TRIG	89		121	
26	ADCSS1	58	GP16C4T3_CH1	90		122	
27	ADCSS2	59	GP16C4T3_CH2	91		123	
28	ADCSS3	60	GP16C4T3_CH3	92		124	
29		61	GP16C4T3_CH4	93		125	
30		62	GP16C4T3_UP	94		126	
31		63	GP16C4T3_TRIG	95		127	

表 8-1 DMA 请求对应表

8.4.3 仲裁器

仲裁器根据优先级来管理 6 个信道的请求，并启动外设/存储器访问序列

优先级分两个阶段进行管理：

- ◆ 软件：可以在 **DMA_CSRx** 寄存器中配置该信道优先级(设定数值越小，优先权越高)。
- ◆ 硬件：如果两个请求具有相同的软件优先级，则具有较低数字的通道优先于具有较高数字的通道。例如，信道 2 优先于信道 4。

8.4.4 来源，目的和转移模式

源和目的地传输都可以处理整个 4 GB 区域中的外设和存储器，地址包含在 0x0000 0000 到 0xFFFF FFFF 之间。

传输模式使用 **DMA_CSRx** 寄存器中的 MODESEL [1:0]进行配置，共有三种传输方式：存储器到外设，外设到存储器或存储器到存储器的传输。表 8-2 说明了转移模式的选择。

DMA_CSRx 寄存器的 MODESEL [1:0]配置	传输方式
00	存储器到存储器
01	外设到存储器
10	存储器到外设
11	保留

表 8-2 转移模式

当数据宽度（在 **DMA_CSRx** 寄存器中的 SDWSEL 或 DDWSEL 中编程）分别为半字或字时，写入 **DMA_SARx** 或 **DMA_DARx** 寄存器的外设或存储器地址必须对齐在一个字或半字地址边界。

8.4.5 地址递增

根据 **DMA_CSRx** 寄存器中的 SINC 和 DINC，每次传输后，外设和存储器指针可以选择自动递增或保持不变。

当目的或者源地址不递增时，可以禁用增量模式。

如果增量模式被使能，则下一个传输的地址将是前一个地址的地址增加 1(对应于字节传输)、2(对应于半字传输)、4(对应于字传输)，具体取决于在 **DMA_CSRx** 寄存器中的 SDWSEL 或 DDWSEL。

为了优化打包操作，可以固定外设地址的增量偏移大小。

DMA_CSRx 寄存器中的 SINCOS 或 DINCOS 位用于将增量偏移大小与外设 AHB 端口上的数据大小或 32 位地址对齐（该地址增加 4）。

如果 SINCOS 或 DINCOS 被设置，则接下来传输的地址是上一个地址增加 4（32 位地址自动对齐），无论 SDWSEL 或 DDWSEL 值是多少。

8.4.6 循环模式

循环模式可用于处理循环缓冲区和连续数据流(例如 ADC 扫描模式)。可以使用 **DMA_CSRx** 寄存器中的 **CIRC** 使能该功能。

当循环模式被激活时,在信道配置阶段,要在编程的初始值中自动重新载入要传输的数据项数,并且 **DMA** 请求继续被提供。

8.4.7 可编程数据宽度,打包/解包,字节顺序

要传输的数据项数必须在启用通道(如果外设流量控制器是外围设备,**DMA_CSRx** 中的 **PFCTRL** 需要被使能)之前被编程到 **DMA_BCRx** (传输字节数,BCR)。

使用内部 FIFO 时,源和目标数据的数据宽度可通过 **DMA_CSRx** 寄存器中的 **SDWSEL** 和 **DDWSEL** 来设置(可以是 8 位,16 位或 32 位)。

当 **SDWSEL** 和 **DDWSEL** 不相同:

在 **DMA_BCRx** 寄存器中配置的要传输的数据项数量的数据宽度取决于最大的源或目的宽度(比较 **DMA_CSRx** 寄存器中的 **SDWSEL** 和 **DDWSEL** 哪个设定较大)。例如,在外设到存储器,存储器到外设或存储器到存储器传输的情况下,如果 **SDWSEL** [1:0]被配置为 16 位,**DDWSEL**[1:0]被配置为 8 位,则要传送的字节数等于 $2 \times \text{BCR}$,**DMA** 控制器只处理源端和目标端的小端寻址。这在表 8-3 中描述:打包/解包和端序行为(位 **SINC** = **DINC** = 1)。

在数据完全打包/解包之前操作中中断时,此打包/拆包过程可能会导致数据损坏。因此,为了确保数据一致性,信道可以被配置为产生突发传输:在这种情况下,属于突发的每组传输是不可分割的(参见第 **錯誤! 找不到參照來源。**: 单个和突发传输)。

在直接模式下(**DMA_CSRx** 寄存器中的 **DIRMDEN**),数据的打包/解包是不可能的。在这种情况下,不允许具有不同的源和目标传输数据宽度:两者都相等并由 **DMA_CSRx** 中的 **SDWSEL** 定义,此时 **DDWSEL** 设定需与 **SDWSEL** 相同。

SDW SEL	DDW SEL	BCR	Source transfer number	Source port address / byte lane		Destination transfer number	Destination port address / byte lane	
				SINCOS=1	SINCOS=0		DINCOS=1	DINCOS=0
8	8	4	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
8	16	4	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1 2	0x0 / B1 B0[15:0] 0x4 / B3 B2[15:0]	0x0 / B1 B0[15:0] 0x2 / B3 B2[15:0]
8	32	4	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1	0x0 / B3 B2 B1 B0[31:0]	0x0 / B3 B2 B1 B0[31:0]
16	8	4	1 2	0x0 / B1 B0[15:0] 0x4 / B3 B2[15:0]	0x0 / B1 B0[15:0] 0x2 / B3 B2[15:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
16	16	4	1 2	0x0 / B1 B0[15:0] 0x4 / B3 B2[15:0]	0x0 / B1 B0[15:0] 0x2 / B3 B2[15:0]	1 2	0x0 / B1 B0[15:0] 0x4 / B3 B2[15:0]	0x0 / B1 B0[15:0] 0x2 / B3 B2[15:0]
16	32	4	1 2	0x0 / B1 B0[15:0] 0x4 / B3 B2[15:0]	0x0 / B1 B0[15:0] 0x2 / B3 B2[15:0]	1	0x0 / B3 B2 B1 B0[31:0]	0x0 / B3 B2 B1 B0[31:0]
32	8	4	1	0x0 / B3 B2 B1 B0[31:0]	0x0 / B3 B2 B1 B0[31:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
32	16	4	1	0x0 / B3 B2 B1 B0[31:0]	0x0 / B3 B2 B1 B0[31:0]	1 2	0x0 / B1 B0[15:0] 0x4 / B3 B2[15:0]	0x0 / B1 B0[15:0] 0x2 / B3 B2[15:0]
32	32	4	1	0x0 / B3 B2 B1 B0[31:0]	0x0 / B3 B2 B1 B0[31:0]	1	0x0 / B3 B2 B1 B0[31:0]	0x0 / B3 B2 B1 B0[31:0]

表 8-3 打包/解包和端序行为 (位 SINC = DINC = 1)

注意：外设端口可能是源或目标（在内存到内存传输的情况下也可能是内存来源）。必须配置 SDWSEL, DDWSEL 和 BCR [15:0], 以确保最后一次传输不会不完整。当外设端口 (DDWSEL) 的数据宽度低于存储器端口的数据宽度 (SDWSEL) 时, 可能会发生这种情况。这个约束总结在表 8-4 中。

SDWSEL[1:0] of DMA_CSRx	DDWSEL[1:0] of DMA_CSRx	BCR[15:0] of DMA_BCRx
00(8-bit)	01(16-bit)	必须是 2 的倍数
00(8-bit)	10(32-bit)	必须是 4 的倍数
01(16-bit)	00(8-bit)	必须是 2 的倍数
01(16-bit)	10(32-bit)	必须是 4 的倍数
10(32-bit)	00(8-bit)	必须是 4 的倍数
10(32-bit)	01(16-bit)	必须是 4 的倍数

表 8-4 对 BCR 与 SDWSEL 和 DDWSEL 的限制

8.4.8 单次和突发传输

DMA 控制器可以生成 4,8 或 16 个节拍的单次传输或增量突发传输。通过使用 **DMA_CSRx** 寄存器中的 **SBUSEL [2:0]**和 **DBUSEL [2:0]**, 为 AHB 端口配置脉冲串的大小。突发大小表示突发中的节拍数, 而不是传输的字节数。

为了确保数据一致性, 形成突发的每组传输是不可分割的: AHB 传输被锁定, 并且 AHB 总线矩阵的仲裁器在突发传送的序列期间不使 DMA 主设备失效。

根据单个或突发配置, 每个 DMA 请求在 AHB 外设端口上启动不同数量的传输:

- ◆ 当 AHB 外设端口配置为单次传输时, 每个 DMA 请求将根据 **DMA_CSRx** 寄存器中的 **DDWSEL [1:0]**产生字节, 半字或字的数据传输
- ◆ 当 AHB 外设端口配置为突发传输时, 根据 **DMA_CSRx** 中的 **DBUSEL [2:0]**和 **DDWSEL [1:0]**, 每个 DMA 请求产生 4,8 或 16 个字节, 半字或字传输次数寄存器。

AHB 来源的配置方式与上述相同, 差异在于考虑到 **SBUSEL [2:0]**和 **SDWSEL [1:0]**。

在直接模式下, 信道只能生成单个传输, 并且硬件强制禁用 **SBUSEL [2:0]**和 **DBUSEL [2:0]**。

必须选择地址指针 (**DMA_SADRx** 或 **DMA_DADRx** 寄存器), 以确保突发块中的所有传输在等于传输大小的地址边界上对齐。

选择突发配置必须遵守 AHB 协议, 其中突发不能跨越 1 KB 地址边界, 因为可以分配给单个从站的最小地址空间为 1 KB。这意味着 1 KB 的地址边界不应该被突发块传输超越, 否则会产生 AHB 错误, 这不是由 DMA 寄存器报告的。

8.4.9 直接模式

默认情况下, FIFO 以直接模式运行 (**DMA_CSRx** 中的 **DIRMDEN** 置位 1), 并且不使用 FIFO 阈值电平。当每个 DMA 请求之后系统需要立即单次传输或从存储器传输时, 此模式很有用。

当 DMA 被配置为直接模式 (FIFO 禁用) 时, 要以存储器到外设模式传输数据, DMA 将从存储器中将一个数据预加载到内部 FIFO, 以确保在 DMA 请求被触发时立即进行数据传输到外设。

为了避免 FIFO 饱和, 建议高优先级配置相应的信道。

此模式仅限于:

- ◆ 源和目标传输宽度相等, 并且都由 **DMA_CSRx** 中的 **SDWSEL [1:0]**定义 (**DDWSEL [1:0]** 会等于 **SDWSEL [1:0]**)。
- ◆ 不能进行突发传输 (禁用 **DMA_CSRx** 中的 **SBUSEL [2:0]**和 **DBUSEL [2:0]**)

8.4.10 DMA 传输完成

- ◆ 在 DMA 流量控制器模式下：
 - ◇ **DMA_CBCRx** 计数器下达到零
 - ◇ 通道在传输结束之前被禁用（通过清除传输中的 **DMA_CSRx** 寄存器中的 **CHEN**）和（当传输是外设到存储器或存储器到存储器）所有剩余的数据已从 FIFO 传入存储器
- ◆ 在外设流量控制器模式下：
 - ◇ 最后一个外部突发或单个请求是从外设产生的，（当 DMA 在外设到存储器模式下工作时）剩余的数据已经从 FIFO 传输到存储器
 - ◇ 通道被软件禁用，（当 DMA 在外设模式下运行时）剩余的数据已经从 FIFO 传输到存储器

注意：只有在外设到内存模式的情况下，传输完成取决于 FIFO 中剩余的数据才能传输到存储器中。这种情况不适用于存储器到外设模式。

如果信道被配置为非循环模式，则在传输结束（即要传输的数据数达到零）之后，DMA 被停止（**DMA_CSRx** 寄存器中的 **CHEN** 被硬件清零），此时外设请求 DMA 不会响应，除非软件重新编程并重新启用它（通过设置 **DMA_CSRx** 寄存器中的 **CHEN**）。

8.4.11 流量控制器

控制要传输的数据数量的实体称为流量控制器。该流量控制器使用 **DMA_CSRx** 寄存器中的 **PFCTRL** 为每个传输独立配置。

流量控制器可以是：

- ◆ **DMA 控制器**：在这种情况下，在 DMA 通道启用之前，要传输的数据项数由软件编程到 **DMA_BCRx** 寄存器中。
- ◆ **外设源或目的地**：当要传输的数据项数量未知时，在这种情况下当传输最后一个数据时，外设由硬件指示给 DMA 控制器。此功能仅支持能够通知传输结束的外设

当外部流量控制器用于给定传输时，写入 **DMA_BCRx** 的值对 DMA 传输没有影响。实际上，无论写入的值是什么，它的方案是：一旦通道启用，就被硬件强制为 **0xFFFF**，以便遵循以下方案：

- ◆ **预期传输中断**：**DMA_CSRx** 寄存器中的 **EN** 位由软件复位为 **0**，以在外设发送最后一个数据硬件信号（单个或突发）之前停止传输。要知道 DMA 传输期间传输的数据项数量，请读取 **DMA_BCRx** 寄存器并应用以下公式：
- ◆ **Number_of_data_transferred = 0xFFFF - DMA_CBCRx**
- ◆ **由于接收到最后一个数据硬件信号而导致正常传输中断**：当外设请求最后一次传输（单次或突发）以及此传输完成时，该传输自动中断。要知道传输的数据项数量，请读取 **DMA_BCRx** 寄存器并应用与上述相同的公式。
- ◆ **DMA_CBCRx 寄存器达到 0**：在中断寄存器中设置相应传输的 **TABORT** 标志，以指示强制 DMA 传输完成。即使最后一个数据硬件信号（单个或突发）尚未被置入，传输也自动关闭。已传输的数据将不会丢失。这意味着即使在外设传输控制模式下，DMA 也可以在单个事务中最多管理 **65535** 个数据项。

注意：在外设流量控制器模式下禁止循环模式。

8.4.12 支持配置的总结

DMA 支持配置的总结，表 8-5 总结了不同的可能的 DMA 配置。

DMA 传输模式	流量控制器	循环模式	转换类型	直接模式
Memory-to-memory	DMA only	支持	single	支持
			burst	不支持
Peripheral-to-memory	DMA	支持	single	支持
			burst	不支持
	Peripheral	不支持	single	支持
			burst	不支持
Memory-to-Peripheral	DMA	支持	single	支持
			burst	不支持
	Peripheral	不支持	single	支持
			burst	不支持

表 8-5 DMA 配置组合

8.4.13 信道配置程序

应遵循以下顺序来配置 DMA 信道 x （其中 x 是通道号）：

1. 如果通道启用，则通过复位 **DMA_CSR x** 寄存器中的 **CHEN** 来禁用它，然后读取该位，以确认没有正在进行的传输操作。将此位写入 0 不会立即生效，因为一旦所有当前传输完成，实际写入 0。当 **CHEN** 读为 0 时，这意味着传输已准备好配置。因此，在启动任何传输配置之前，需要等待 **CHEN** 清零，然后可以重新启用传输。
2. 将源端口寄存器地址设置在 **DMA_SADR x** 寄存器中。在周边事件之后，数据将从外部端口/从该地址移动到外设端口。
3. 将目的端口寄存器地址设置在 **DMA_DADR x** 寄存器中。在周边事件之后，数据将从外部端口/从该地址移动到外设端口。
4. 配置要在 **DMA_BCR x** 寄存器中传输的数据项的总数。在每个周边事件或突发的每个节拍之后，**DMA_BCR x** 寄存器中 **CBCR** 值会递减。
5. 在 **DMA_CSR x** 寄存器中使用 **PHSS [4:0]** 选择 DMA 通道（请求）。
6. 如果外围设备是流量控制器，并且它支持此功能，请将 **DMA_CSR x** 寄存器中的 **PFCTRL** 设置 1。
7. 使用 **DMA_CSR x** 寄存器中的 **CHPRI [2:0]** 来配置信道优先级。
8. 配置数据传输方向，外设和存储器递增/固定模式，单个或突发事件，外围设备和存储器数据宽度，循环模式，完全传输后的中断或错误传输的中断。
9. 通过设置 **DMA_CSR x** 寄存器中的 **CHEN** 位来激活通道。

一旦通道启用，它可以提供来自连接到通道的外设的任何 DMA 请求。

在传输结束时，如果设置了传输完成中断使能位（**BTD**），则设置传输完成标志（**BTD**）并产生中断。

8.4.14 DMA 通道错误讯息

DMA 控制器可以检测到以下错误并产生错误讯息:

- ◆ SETBCER: SDWSEL、DDWSEL 和传输数据之间设定错误, 可参考下表

BCR	SDWSEL	DDWSEL	SETBCER
0	Don't care	Don't care	1
0	8	8	1
不是 2 的倍数	8	16	1
不是 4 的倍数	8	32	1
不是 2 的倍数	16	8	1
不是 2 的倍数	16	16	1
不是 4 的倍数	16	32	1
不是 4 的倍数	32	8	1
不是 4 的倍数	32	16	1
不是 4 的倍数	32	32	1

- ◆ SETBUER : SBUSEL 或 DBUSEL 设定值为 1。
- ◆ PFOV :当流量控制器是选择外设, 且传输数据量超过 65535 时。
- ◆ PFSER :当流量控制器是选择外设, 且设定在外设到存储器传输模式, 由于 SDWSEL 与 DDWSEL 设定不同, 导致外设向 DMA 控制器请求结束传输时, 不满足目的数据宽度, 这时应体会补上剩余的所需的宽度数据(无效的值)。

8.5 特殊功能寄存器

8.5.1 寄存器列表

DMA 寄存器列表			
名称	偏移地址	类型	描述
DMA_IER	0000 _H	W1	DMA 中断允许寄存器
DMA_IDR	0004 _H	W1	DMA 中断禁用寄存器
DMA_IVS	0008 _H	R	DMA 中断有效状态寄存器
DMA_RIF	000C _H	R	DMA 原始中断标志状态寄存器
DMA_IFM	0010 _H	R	DMA 中断标志屏蔽状态寄存器
DMA_ICR	0014 _H	C_W1	DMA 中断清除寄存器
DMA_EMMSG	0018 _H	R	DMA 错误讯息寄存器
DMA_CSR0	0020 _H	R/W	DMA 通道 0 控制和状态寄存器
DMA_SAR0	0024 _H	R/W	DMA 信道 0 源地址寄存器
DMA_DAR0	0028 _H	R/W	DMA 通道 0 目的地址寄存器
DMA_BCR0	002C _H	R/W	DMA 通道 0 传输字节计数寄存器
DMA_CSR1	0030 _H	R/W	DMA 通道 1 控制和状态寄存器
DMA_SAR1	0034 _H	R/W	DMA 信道 1 源地址寄存器
DMA_DAR1	0038 _H	R/W	DMA 通道 1 目的地址寄存器
DMA_BCR1	003C _H	R/W	DMA 通道 1 传输字节计数寄存器
DMA_CSR2	0040 _H	R/W	DMA 通道 2 控制和状态寄存器
DMA_SAR2	0044 _H	R/W	DMA 信道 2 源地址寄存器
DMA_DAR2	0048 _H	R/W	DMA 通道 2 目的地址寄存器
DMA_BCR2	004C _H	R/W	DMA 通道 2 传输字节计数寄存器
DMA_CSR3	0050 _H	R/W	DMA 通道 3 控制和状态寄存器
DMA_SAR3	0054 _H	R/W	DMA 信道 3 源地址寄存器
DMA_DAR3	0058 _H	R/W	DMA 通道 3 目的地址寄存器
DMA_BCR3	005C _H	R/W	DMA 通道 3 传输字节计数寄存器
DMA_CSR4	0060 _H	R/W	DMA 通道 4 控制和状态寄存器
DMA_SAR4	0064 _H	R/W	DMA 信道 4 源地址寄存器
DMA_DAR4	0068 _H	R/W	DMA 通道 4 目的地址寄存器
DMA_BCR4	006C _H	R/W	DMA 通道 4 传输字节计数寄存器
DMA_CSR5	0070 _H	R/W	DMA 通道 5 控制和状态寄存器
DMA_SAR5	0074 _H	R/W	DMA 信道 5 源地址寄存器
DMA_DAR5	0078 _H	R/W	DMA 通道 5 目的地址寄存器
DMA_BCR5	007C _H	R/W	DMA 通道 5 传输字节计数寄存器

8.5.2 寄存器描述

8.5.2.1 DMA 中断允许寄存器 (DMA_IER)

DMA 中断允许寄存器 (DMA_IER)																																
偏移地址: 00H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					CH5TABIE	CH5BTDIE	CH4TABIE	CH4BTDIE	CH3TABIE	CH3BTDIE	CH2TABIE	CH2BTDIE	CH1TABIE	CH1BTDIE	CH0TABIE	CH0BTDIE

—	Bits 31-12	—	—
CH5TABIE	Bit 11	W1	DMA 通道5 TABORT中断使能 0: 无效 1: 启用中断
CH5BTDIE	Bit 10	W1	DMA 通道5 块传输完成中断使能 0: 无效 1: 启用中断
CH4TABIE	Bit 9	W1	DMA 通道4 TABORT中断使能 0: 无效 1: 启用中断
CH4BTDIE	Bit 8	W1	DMA 通道4 块传输完成中断使能 0: 无效 1: 启用中断
CH3TABIE	Bit 7	W1	DMA 通道3 TABORT中断使能 0: 无效 1: 启用中断
CH3BTDIE	Bit 6	W	DMA 通道3 块传输完成中断使能 0: 无效 1: 启用中断
CH2TABIE	Bit 5	W1	DMA 通道2 TABORT中断使能 0: 无效 1: 启用中断
CH2BTDIE	Bit 4	W1	DMA 通道2 块传输完成中断使能 0: 无效 1: 启用中断
CH1TABIE	Bit 3	W1	DMA 通道1 TABORT中断使能 0: 无效 1: 启用中断
CH1BTDIE	Bit 2	W1	DMA 通道1 块传输完成中断使能 0: 无效 1: 启用中断

CH0TABIE	Bit 1	W1	DMA 通道0 TABORT中断使能 0: 无效 1: 启用中断
CH0BTDIE	Bit 0	W1	DMA 通道0 块传输完成中断使能 0: 无效 1: 启用中断

			0: 无效 1: 禁用中断
CH0BTDID	Bit 0	W1	DMA 通道0 块传输完成中断禁用 0: 无效 1: 禁用中断

8.5.2.3 DMA 中断有效状态寄存器 (DMA_IVS)

DMA 中断有效状态寄存器 (DMA_IVS)																																
偏移地址: 08H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					CH5TABIVS	CH5BTDIVS	CH4TABIVS	CH4BTDIVS	CH3TABIVS	CH3BTDIVS	CH2TABIVS	CH2BTDIVS	CH1TABIVS	CH1BTDIVS	CH0TABIVS	CH0BTDIVS

—	Bits 31-12	—	—
CH5TABIVS	Bit 11	R	DMA 通道5 TABORT中断有效状态 0: 无效 1: 有效中断
CH5BTDIVS	Bit 10	R	DMA 通道5 块传输完成中断有效状态 0: 无效 1: 有效中断
CH4TABIVS	Bit 9	R	DMA 通道4 TABORT中断有效状态 0: 无效 1: 有效中断
CH4BTDIVS	Bit 8	R	DMA 通道4 块传输完成中断有效状态 0: 无效 1: 有效中断
CH3TABIVS	Bit 7	R	DMA 通道3 TABORT中断有效状态 0: 无效 1: 有效中断
CH3BTDIVS	Bit 6	R	DMA 通道3 块传输完成中断有效状态 0: 无效 1: 有效中断
CH2TABIVS	Bit 5	R	DMA 通道2 TABORT中断有效状态 0: 无效 1: 有效中断
CH2BTDIVS	Bit 4	R	DMA 通道2 块传输完成中断有效状态 0: 无效 1: 有效中断
CH1TABIVS	Bit 3	R	DMA 通道1 TABORT中断有效状态 0: 无效 1: 有效中断
CH1BTDIVS	Bit 2	R	DMA 通道1 块传输完成中断有效状态 0: 无效 1: 有效中断
CH0TABIVS	Bit 1	R	DMA 通道0 TABORT中断有效状态

			0: 无效 1: 有效中断
CH0BTDIVS	Bit 0	R	DMA 通道0 块传输完成中断有效状态 0: 无效 1: 有效中断

8.5.2.4 DMA 原始中断标志状态寄存器 (DMA_RIF)

DMA 原始中断标志状态寄存器 (DMA_RIF)																																
偏移地址: 0C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					CH5TABRIF	CH5BTDRIF	CH4TABRIF	CH4BTDRIF	CH3TABRIF	CH3BTDRIF	CH2TABRIF	CH2BTDRIF	CH1TABRIF	CH1BTDRIF	CH0TABRIF	CH0BTDRIF

—	Bits 31-12	—	—
CH5TABRIF	Bit 11	R	DMA 通道5 TABORT原始中断标志状态 0: 无效 1: 原始中断标志状态
CH5BTDRIF	Bit 10	R	DMA 通道5 块传输完成原始中断标志状态 0: 无效 1: 原始中断标志状态
CH4TABRIF	Bit 9	R	DMA 通道4 TABORT原始中断标志状态 0: 无效 1: 原始中断标志状态
CH4BTDRIF	Bit 8	R	DMA 通道4块传输完成原始中断标志状态 0: 无效 1: 原始中断标志状态
CH3TABRIF	Bit 7	R	DMA 通道3 TABORT原始中断标志状态 0: 无效 1: 原始中断标志状态
CH3BTDRIF	Bit 6	R	DMA 通道3 块传输完成原始中断标志状态 0: 无效 1: 原始中断标志状态
CH2TABRIF	Bit 5	R	DMA 通道2 TABORT原始中断标志状态 0: 无效 1: 原始中断标志状态
CH2BTDRIF	Bit 4	R	DMA 通道2 块传输完成原始中断标志状态 0: 无效 1: 原始中断标志状态
CH1TABRIF	Bit 3	R	DMA 通道1 TABORT原始中断标志状态 0: 无效 1: 原始中断标志状态
CH1BTDRIF	Bit 2	R	DMA 通道1 块传输完成原始中断标志状态 0: 无效 1: 原始中断标志状态
CH0TABRIF	Bit 1	R	DMA 通道0 TABORT原始中断标志状态

			0: 无效 1: 原始中断标志状态
CH0BTDRIF	Bit 0	R	DMA 通道0 块传输完成原始中断标志状态 0: 无效 1: 原始中断标志状态

8.5.2.5 DMA 中断标志屏蔽状态寄存器 (DMA_IFM)

DMA 中断标志屏蔽状态寄存器 (DMA_IFM)																																
偏移地址: 10H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					CH5TABIFM	CH5BTDIFM	CH4TABIFM	CH4BTDIFM	CH3TABIFM	CH3BTDIFM	CH2TABIFM	CH2BTDIFM	CH1TABIFM	CH1BTDIFM	CH0TABIFM	CH0BTDIFM

—	Bits 31-12	—	—
CH5TABIFM	Bit 11	R	DMA 通道5 TABORT 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH5BTDIFM	Bit 10	R	DMA 通道5 块传输完成 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH4TABIFM	Bit 9	R	DMA 通道4 TABORT 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH4BTDIFM	Bit 8	R	DMA 通道4 块传输完成 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH3TABIFM	Bit 7	R	DMA通道3 TABORT 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH3BTDIFM	Bit 6	R	DMA 通道3 块传输完成 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH2TABIFM	Bit 5	R	DMA 通道3 TABORT 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH2BTDIFM	Bit 4	R	DMA 通道3 块传输完成 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH1TABIFM	Bit 3	R	DMA 通道1 TABORT 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH1BTDIFM	Bit 2	R	DMA 通道1 块传输完成 中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
CH0TABIFM	Bit 1	R	DMA 通道0 TABORT 中断标志屏蔽状态

			0: 无效 1: 中断标志屏蔽状态
CH0BTDIFM	Bit 0	R	DMA 通道0 块传输完成中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态

8.5.2.6 DMA 中断清除寄存器 (DMA_ICR)

DMA 中断清除寄存器 (DMA_ICR)																																
偏移地址: 14 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					CH5TABICR	CH5BTDICR	CH4TABICR	CH4BTDICR	CH3TABICR	CH3BTDICR	CH2TABICR	CH2BTDICR	CH1TABICR	CH1BTDICR	CH0TABICR	CH0BTDICR

—	Bits 31-12	—	—
CH5TABICR	Bit 11	C_W1	DMA 通道5 TABORT中断清除 0: 无效 1: 中断清除
CH5BTDICR	Bit 10	C_W1	DMA 通道5 块传输完成中断清除 0: 无效 1: 中断清除
CH4TABICR	Bit 9	C_W1	DMA 通道4 TABORT中断清除 0: 无效 1: 中断清除
CH4BTDICR	Bit 8	C_W1	DMA 通道4 块传输完成中断清除 0: 无效 1: 中断清除
CH3TABICR	Bit 7	C_W1	DMA 通道3 TABORT中断清除 0: 无效 1: 中断清除
CH3BTDICR	Bit 6	C_W1	DMA 通道3 块传输完成中断清除 0: 无效 1: 中断清除
CH2TABICR	Bit 5	C_W1	DMA 通道2 TABORT中断清除 0: 无效 1: 中断清除
CH2BTDICR	Bit 4	C_W1	DMA 通道2 块传输完成中断清除 0: 无效 1: 中断清除
CH1TABICR	Bit 3	C_W1	DMA 通道1 TABORT中断清除 0: 无效 1: 中断清除
CH1BTDICR	Bit 2	C_W1	DMA 通道1 块传输完成中断清除 0: 无效 1: 中断清除
CH0TABICR	Bit 1	C_W1	DMA 通道0 TABORT中断清除

			0: 无效 1: 中断清除
CH0BTDICR	Bit 0	C_W1	DMA 通道0 块传输完成中断清除 0: 无效 1: 中断清除

8.5.2.8 DMA 通道 0 控制和状态寄存器 (DMA_CSR0)

DMA 通道 0 控制和状态寄存器 (DMA_CSR0)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINCOS		DBUSEL<2:0>			DDWSEL<1:0>			DINC		SINCOS		SBUSEL<2:0>			SDWSEL<1:0>		PHSS<6:0>						CHPRI<2:0>			MODESEL<1:0>		DIRMDEN	PFCTRL	CIRC	CHEN

—	Bit 31	—	—
DINCOS	Bit 30	R/W	<p>目的增量偏移尺寸</p> <p>0: 外设地址计算的偏移量与DWSEL连接</p> <p>1: 外设地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果比特DINC=0, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p> <p>如果选择直接模式, 或者如果DBUSEL与“00”不同, 则在启用DMA通道时(位EN='1'), 硬件会强制清零该位。</p>
DBUSEL<2:0>	Bits 29-27	R/W	<p>目的地传输突发选择</p> <p>000 : SINGLE</p> <p>001 : Reserved</p> <p>010 : WRAP4</p> <p>011 : INCR4</p> <p>100 : WRAP8</p> <p>101 : INCR8</p> <p>110 : WRAP16</p> <p>111 : INCR16</p>
DDWSEL<1:0>	Bits 26-25	R/W	<p>目的地传输数据宽度选择</p> <p>00: 为每个DMA操作传送一个字节 (8位)。</p> <p>01: 每一个DMA操作都传输一个半字 (16位)。</p> <p>10: 为每个DMA操作传送一个字 (32位)。</p> <p>11: 保留</p>
DINC	Bit 24	R/W	<p>目的地转移增量模式</p> <p>0: 禁用目的地址增量模式</p> <p>1: 启用目标地址增量模式</p>
—	Bit 23	—	—
SINCOS	Bit 22	R/W	<p>源增量偏移尺寸</p> <p>0: 源地址计算的偏移量与SWSEL链接</p> <p>1: 源地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果位SINC=“0”, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p>

			如果选择直接模式，或者如果SBUSEL与“00”不同，则当DMA通道被启用时（位EN='1'），硬件会强制降低该位。
SBUSEL<2:0>	Bits 21-19	R/W	源传输突发选择 000: SINGLE 001: 保留 010: WRAP4 011: INCR4 100: WRAP8 101: INCR 8 110: WRAP16 111: INCR 16
SDWSEL<1:0>	Bits 18-17	R/W	源传输数据宽度选择 00: 为每个DMA操作传送一个字节（8位）。 01: 每一个DMA操作都传输一个半字（16位）。 10: 为每个DMA操作传送一个字（32位）。 11: 保留
SINC	Bit 16	R/W	源转移增量模式 0: 源地址增量模式禁用 1: 启用源地址增量模式
PHSS	Bits 15-9	R/W	外设握手软件选择
CHPRI	Bits 8-6	R/W	DMA通道优先级 优先级的值是从5到0，设定数值越小，优先权越高。
MODESEL<1:0>	Bits 5-4	R/W	DMA模式选择 00: 内存到内存模式（内存到内存）。 01: 外设存储器模式（外设存储器）。 10: 内存到外设模式（内存到外设）。 11: 保留
DIRMDEN	Bit 3	R/W	直接模式使能 0: 禁用直接模式 1: 启用直接模式 这个比特是受保护的，只有当EN是“0”时才能写入。
PFCTRL	Bit 2	R/W	外设流量控制器 该位由软件设置和清除。 0: DMA是流量控制器 1: 外设设备是流量控制器 这个比特是受保护的，只有当EN是“0”时才能写入。 当选择存储器到存储器模式（位DIR [1:0]=10）时，该位通过硬件自动地被强制为0。
CIRC	Bit 1	R/W	循环模式 该位由软件设置和清除，并可通过硬件清除。

			<p>0: 禁用循环模式 1: 启用循环模式 当外设设备是流量控制器(位PFCTRL=1)并且DMA通道被启用(位EN=1)时, 该位被硬件自动强制为0。</p>
CHEN	Bit 0	R/W	<p>DMA通道启用 将此位设置为“1”使DMA的操作得以实现。如果该位被清除, DMA将忽略所有DMA请求, 并迫使总线主机进入空闲状态。</p>

8.5.2.9 DMA 通道 0 源地址寄存器 (DMA_SAR0)

DMA 通道 0 源地址寄存器 (DMA_SAR0)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR<31:0>																															

SAR<31:0>	Bits 31-0	R/W	DMA传输源地址寄存器 此字段指示DMA的32位源地址。 这些位是写保护的，并且只能在比特EN = “0” 时写入。
-----------	-----------	-----	--

8.5.2.10 DMA 通道 0 目的地址寄存器 (DMA_DAR0)

DMA 通道 0 目的地址寄存器 (DMA_DAR0)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR<31:0>																															

DAR<31:0>	Bits 31-0	R/W	传输目的地址寄存器 此字段指示DMA的32位目标地址。 这些位是写保护的，并且只能在比特EN = “0” 时写入。
-----------	-----------	-----	---

8.5.2.11 DMA 通道 0 传输字节计数寄存器 (DMA_BCR0)

DMA 通道 0 传输字节计数寄存器 (DMA_BCR0)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBCR<15:0>																BCR<15:0>															

CBCR<15:0>	Bits 31-16	R	DMA 当前传输字节数
BCR<15:0>	Bits 15-0	R/W	DMA传输字节计数寄存器 此字段指示DMA的16位传输字节计数。 这些位是写保护的，并且只能在比特EN = “0” 时写入。

8.5.2.12 DMA 通道 1 控制和状态寄存器 (DMA_CSR1)

DMA 通道 1 控制和状态寄存器 (DMA_CSR1)																															
偏移地址: 30 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINCOS		DBUSEL<2:0>			DDWSEL<1:0>		DINC		SINCOS		SBUSEL<2:0>			SDWSEL<1:0>		PHSS<6:0>						CHPRI<2:0>		MODESEL<1:0>		DIRMDEN	PFCTRL	CIRC	CHEN		

—	Bit 31	—	—
DINCOS	Bit 30	R/W	<p>目的增量偏移尺寸</p> <p>0: 外设地址计算的偏移量与DWSEL连接</p> <p>1: 外设地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果比特DINC=0, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p> <p>如果选择直接模式, 或者如果DBUSEL与“00”不同, 则在启用DMA通道时(位EN='1'), 硬件会强制降低该位。</p>
DBUSEL<2:0>	Bits 29-27	R/W	<p>目的地传输突发选择</p> <p>000 : SINGLE</p> <p>001 : Reserved</p> <p>010 : WRAP4</p> <p>011 : INCR4</p> <p>100 : WRAP8</p> <p>101 : INCR8</p> <p>110 : WRAP16</p> <p>111 : INCR16</p>
DDWSEL<1:0>	Bits 26-25	R/W	<p>目的地传输数据宽度选择</p> <p>00: 为每个DMA操作传送一个字节 (8位)。</p> <p>01: 每一个DMA操作都传输一个半字 (16位)。</p> <p>10: 为每个DMA操作传送一个字 (32位)。</p> <p>11: 保留</p>
DINC	Bit 24	R/W	<p>目的地转移增量模式</p> <p>0: 禁用目的地址增量模式</p> <p>1: 启用目标地址增量模式</p>
—	Bit 23	—	—
SINCOS	Bit 22	R/W	<p>源增量偏移尺寸</p> <p>0: 源地址计算的偏移量与SWSEL链接</p> <p>1: 源地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果位SINC=“0”, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p>

			如果选择直接模式，或者如果SBUSEL与“00”不同，则当DMA通道被启用时（位EN='1'），硬件会强制降低该位。
SBUSEL<2:0>	Bits 21-19	R/W	源传输突发选择 000: SINGLE 001: 保留 010: WRAP4 011: INCR4 100: WRAP8 101: INCR8 110: WRAP16 111: INCR16
SDWSEL<1:0>	Bits 18-17	R/W	源传输数据宽度选择 00: 为每个DMA操作传送一个字节（8位）。 01: 每一个DMA操作都传输一个半字（16位）。 10: 为每个DMA操作传送一个字（32位）。 11: 保留
SINC	Bit 16	R/W	源转移增量模式 0: 源地址增量模式禁用 1: 启用源地址增量模式
PHSS	Bits 15-9	R/W	外设握手软件选择
CHPRI	Bits 8-6	R/W	DMA通道优先级 优先级的值是从5到0，设定数值越小，优先级越高。
MODESEL<2:0>	Bits 5-4	R/W	DMA模式选择 00: 内存到内存模式（内存到内存）。 01: 外设存储器模式（外设存储器）。 10: 内存到外设模式（内存到外设）。 11: 保留
DIRMDEN	Bit 3	R/W	直接模式使能 0: 禁用直接模式 1: 启用直接模式 这个比特是受保护的，只有当EN是“0”时才能写入。
PFCTRL	Bit 2	R/W	外设流量控制器 该位由软件设置和清除。 0: DMA是流量控制器 1: 外围设备是流量控制器 这个比特是受保护的，只有当EN是“0”时才能写入。 当选择存储器到存储器模式（位DIR [1:0]=10）时，该位通过硬件自动地被强制为0。
CIRC	Bit 1	R/W	循环模式 该位由软件设置和清除，并可通过硬件清除。 0: 禁用循环模式

			<p>1: 启用循环模式</p> <p>当外围设备是流量控制器(位PFCTRL=1)并且DMA通道被启用(位EN=1)时, 该位被硬件自动强制为0。</p>
CHEN	Bit 0	R/W	<p>DMA通道启用</p> <p>将此位设置为“1”使DMA的操作得以实现。如果该位被清除, DMA将忽略所有DMA请求, 并迫使总线主机进入空闲状态。</p>

8.5.2.13 DMA 信道 1 源地址寄存器 (DMA_SAR1)

DMA 通道 1 源地址寄存器 (DMA_SAR1)																															
偏移地址: 34 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR<31:0>																															

SAR<31:0>	Bits 31-0	R/W	DMA传输源地址寄存器 此字段指示DMA的32位源地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----------	-----------	-----	--

8.5.2.14 DMA 通道 1 目的地址寄存器 (DMA_DAR1)

DMA 通道 1 目的地址寄存器 (DMA_DAR1)																															
偏移地址: 38 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR<31:0>																															

DAR<31:0>	Bits 31-0	R/W	传输目的地址寄存器 此字段指示DMA的32位目标地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----------	-----------	-----	---

8.5.2.15 DMA 通道 1 传输字节计数寄存器 (DMA_BCR1)

DMA 通道 1 传输字节计数寄存器 (DMA_BCR1)																															
偏移地址: 3C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBCR<15:0>																BCR<15:0>															

CBCR<15:0>	Bits 31-16	R	DMA 当前传输字节数
BCR<15:0>	Bits 15-0	R/W	DMA传输字节计数寄存器 此字段指示DMA的16位传输字节计数。 这些位是写保护的，并且只能在比特EN = “0” 时写入。

8.5.2.16 DMA 通道 2 控制和状态寄存器 (DMA_CSR2)

DMA 通道 2 控制和状态寄存器 (DMA_CSR2)																															
偏移地址: 40 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINCOS		DBUSEL<2:0>			DDWSEL<1:0>		DINC		SINCOS		SBUSEL<2:0>			SDWSEL<1:0>		PHSS<6:0>						CHPRI<2:0>		MODESEL<1:0>		DIRMDEN	PFCTRL	CIRC	CHEN		

—	Bit 31	—	—
DINCOS	Bit 30	R/W	<p>目的增量偏移尺寸</p> <p>0: 外设地址计算的偏移量与DWSEL连接</p> <p>1: 外设地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果比特DINC=0, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p> <p>如果选择直接模式, 或者如果DBUSEL与“00”不同, 则在启用DMA通道时(位EN='1'), 硬件会强制降低该位。</p>
DBUSEL<2:0>	Bits 29-27	R/W	<p>目的地传输突发选择</p> <p>000 : SINGLE</p> <p>001 : Reserved</p> <p>010 : WRAP4</p> <p>011 : INCR4</p> <p>100 : WRAP8</p> <p>101 : INCR8</p> <p>110 : WRAP16</p> <p>111 : INCR16</p>
DDWSEL<1:0>	Bits 26-25	R/W	<p>目的地传输数据宽度选择</p> <p>00: 为每个DMA操作传送一个字节 (8位)。</p> <p>01: 每一个DMA操作都传输一个半字 (16位)。</p> <p>10: 为每个DMA操作传送一个字 (32位)。</p> <p>11: 保留</p>
DINC	Bit 24	R/W	<p>目的地转移增量模式</p> <p>0: 禁用目的地址增量模式</p> <p>1: 启用目标地址增量模式</p>
—	Bit 23	—	—
SINCOS	Bit 22	R/W	<p>源增量偏移尺寸</p> <p>0: 源地址计算的偏移量与SWSEL链接</p> <p>1: 源地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果位SINC=“0”, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p>

			如果选择直接模式，或者如果SBUSEL与“00”不同，则当DMA通道被启用时（位EN='1'），硬件会强制降低该位。
SBUSEL<2:0>	Bits 21-19	R/W	源传输突发选择 000: SINGLE 001: 保留 010: WRAP4 011: INCR4 100: WRAP8 101: INCR8 110: WRAP16 111: INCR16
SDWSEL<1:0>	Bits 18-17	R/W	源传输数据宽度选择 00: 为每个DMA操作传送一个字节（8位）。 01: 每一个DMA操作都传输一个半字（16位）。 10: 为每个DMA操作传送一个字（32位）。 11: 保留
SINC	Bit 16	R/W	源转移增量模式 0: 源地址增量模式禁用 1: 启用源地址增量模式
PHSS<7:0>	Bits 15-9	R/W	外设握手软件选择
CHPRI	Bits 8-6	R/W	DMA通道优先级 优先级的值是从5到0，设定数值越小，优先权越高。
MODESEL<1:0>	Bits 5-4	R/W	DMA模式选择 00: 内存到内存模式（内存到内存）。 01: 外设存储器模式（外设存储器）。 10: 内存到外设模式（内存到外设）。 11: 保留
DIRMDEN	Bit 3	R/W	直接模式使能 0: 禁用直接模式 1: 启用直接模式 这个比特是受保护的，只有当EN是“0”时才能写入。
PFCTRL	Bit 2	R/W	外设流量控制器 该位由软件设置和清除。 0: DMA是流量控制器 1: 外围设备是流量控制器 这个比特是受保护的，只有当EN是“0”时才能写入。 当选择存储器到存储器模式（位DIR [1:0]=10）时，该位通过硬件自动地被强制为0。
CIRC	Bit 1	R/W	循环模式 该位由软件设置和清除，并可通过硬件清除。

			<p>0: 禁用循环模式 1: 启用循环模式 当外围设备是流量控制器(位PFCTRL=1)并且DMA通道被启用(位EN=1)时, 该位被硬件自动强制为0。</p>
CHEN	Bit 0	R/W	<p>DMA通道启用 将此位设置为“1”使DMA的操作得以实现。如果该位被清除, DMA将忽略所有DMA请求, 并迫使总线主机进入空闲状态。</p>

8.5.2.17 DMA 信道 2 源地址寄存器 (DMA_SAR2)

DMA 通道 2 源地址寄存器 (DMA_SAR2)																															
偏移地址: 44 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR<31:0>																															

SAR<31:0>	Bits 31-0	R/W	DMA传输源地址寄存器 此字段指示DMA的32位源地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----------	-----------	-----	--

8.5.2.18 DMA 通道 2 目的地址寄存器 (DMA_DAR2)

DMA 通道 2 目的地址寄存器 (DMA_DAR2)																															
偏移地址: 48 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR<31:0>																															

DAR<31:0>	Bits 31-0	R/W	传输目的地址寄存器 此字段指示DMA的32位目标地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----------	-----------	-----	---

8.5.2.19 DMA 通道 2 传输字节计数寄存器 (DMA_BCR2)

DMA 通道 2 传输字节计数寄存器 (DMA_BCR2)																															
偏移地址: 4C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBCR<15:0>																BCR<15:0>															

CBCR<15:0>	Bits 31-16	R	DMA 当前传输字节数
BCR<15:0>	Bits 15-0	R/W	DMA传输字节计数寄存器 此字段指示DMA的16位传输字节计数。 这些位是写保护的，并且只能在比特EN = “0” 时写入。

8.5.2.20 DMA 通道 3 控制和状态寄存器 (DMA_CSR3)

DMA 通道 3 控制和状态寄存器 (DMA_CSR3)																																				
偏移地址: 50H																																				
复位值: 00000000_00000000_00000000_00000000 _B																																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
DINCOS		DBUSEL<2:0>			DDWSEL<1:0>			DINC		SINCOS		SBUSEL<2:0>			SDWSEL<1:0>		SINC						PHSS<6:0>						CHPRI<2:0>		MODESEL<1:0>		DIRMDEN	PFCTRL	CIRC	CHEN

—	Bit 31	—	—
DINCOS	Bit 30	R/W	目的增量偏移尺寸 0: 外设地址计算的偏移量与DWSEL连接 1: 外设地址计算的偏移量固定为4 (32位对齐)。 如果比特DINC=0, 这个比特就没有意义了。 该位受到保护, 仅在EN=“0”时才可写入。 如果选择直接模式, 或者如果DBUSEL与“00”不同, 则在启用DMA通道时(位EN='1'), 硬件会强制降低该位。
DBUSEL<2:0>	Bits 29-27	R/W	目的地传输突发选择 000 : SINGLE 001 : Reserved 010 : WRAP4 011 : INCR4 100 : WRAP8 101 : INCR8 110 : WRAP16 111 : INCR16
DDWSEL<1:0>	Bits 26-25	R/W	目的地传输数据宽度选择 00: 为每个DMA操作传送一个字节 (8位)。 01: 每一个DMA操作都传输一个半字 (16位)。 10: 为每个DMA操作传送一个字 (32位)。 11: 保留
DINC	Bit 24	R/W	目的地转移增量模式 0: 禁用目的地址增量模式 1: 启用目标地址增量模式
—	Bit 23	—	—
SINCOS	Bit 22	R/W	源增量偏移尺寸 0: 源地址计算的偏移量与SWSEL链接 1: 源地址计算的偏移量固定为4 (32位对齐)。 如果位SINC=“0”, 这个比特就没有意义了。 该位受到保护, 仅在EN=“0”时才可写入。

			如果选择直接模式，或者如果SBUSEL与“00”不同，则当DMA通道被启用时（位EN='1'），硬件会强制降低该位。
SBUSEL<2:0>	Bits 21-19	R/W	源传输突发选择 000: SINGLE 001: 保留 010: WRAP4 011: INCR4 100: WRAP8 101: INCR8 110: WRAP16 111: INCR16
SDWSEL<1:0>	Bits 18-17	R/W	源传输数据宽度选择 00: 为每个DMA操作传送一个字节（8位）。 01: 每一个DMA操作都传输一个半字（16位）。 10: 为每个DMA操作传送一个字（32位）。 11: 保留
SINC	Bit 16	R/W	源转移增量模式 0: 源地址增量模式禁用 1: 启用源地址增量模式
PHSS	Bits 15-9	R/W	外设握手软件选择
CHPRI	Bits 8-6	R/W	DMA通道优先级 优先级的值是从5到0，设定数值越小，优先权越高。
MODESEL<1:0>	Bits 5-4	R/W	DMA模式选择 00: 内存到内存模式（内存到内存）。 01: 外设存储器模式（外设存储器）。 10: 内存到外设模式（内存到外设）。 11: 保留
DIRMDEN	Bit 3	R/W	直接模式使能 0: 禁用直接模式 1: 启用直接模式 这个比特是受保护的，只有当EN是“0”时才能写入。
PFCTRL	Bit 2	R/W	外设流量控制器 该位由软件设置和清除。 0: DMA是流量控制器 1: 外围设备是流量控制器 这个比特是受保护的，只有当EN是“0”时才能写入。 当选择存储器到存储器模式（位DIR [1:0]=10）时，该位通过硬件自动地被强制为0。
CIRC	Bit 1	R/W	循环模式 该位由软件设置和清除，并可通过硬件清除。

			<p>0: 禁用循环模式 1: 启用循环模式 当外围设备是流量控制器(位PFCTRL=1)并且DMA通道被启用(位EN=1)时, 该位被硬件自动强制为0。</p>
CHEN	Bit 0	R/W	<p>DMA通道启用 将此位设置为“1”使DMA的操作得以实现。如果该位被清除, DMA将忽略所有DMA请求, 并迫使总线主机进入空闲状态。</p>

8.5.2.21 DMA 信道 3 源地址寄存器 (DMA_SAR3)

DMA 通道 3 源地址寄存器 (DMA_SAR3)																															
偏移地址: 54 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR<31:0>																															

SAR<31:0>	Bits 31-0	R/W	DMA传输源地址寄存器 此字段指示DMA的32位源地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----------	-----------	-----	--

8.5.2.22 DMA 通道 3 目的地址寄存器 (DMA_DAR3)

DMA 通道 3 目的地址寄存器 (DMA_DAR3)																															
偏移地址: 58 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR<31:0>																															

DAR<31:0>	Bits 31-0	R/W	传输目的地址寄存器 此字段指示DMA的32位目标地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----------	-----------	-----	---

8.5.2.23 DMA 通道 3 传输字节计数寄存器 (DMA_BCR3)

DMA 通道 3 传输字节计数寄存器 (DMA_BCR3)																															
偏移地址: 5C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBCR<15:0>																BCR<15:0>															

CBCR<15:0>	Bits 31-16	R	DMA 当前传输字节数
BCR<15:0>	Bits 15-0	R/W	DMA传输字节计数寄存器 此字段指示DMA的16位传输字节计数。 这些位是写保护的，并且只能在比特EN = “0” 时写入。

8.5.2.24 DMA 通道 4 控制和状态寄存器 (DMA_CSR4)

DMA 通道 4 控制和状态寄存器 (DMA_CSR4)																															
偏移地址: 60H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINCOS		DBUSEL<2:0>			DDWSEL<1:0>		DINC	—	SINCOS	SBUSEL<2:0>			SDWSEL<1:0>		SINC	PHSS<6:0>						CHPRI<2:0>		MODESEL<1:0>		DIRMDEN	PFCTRL	CIRC	CHEN		

—	Bit 31	—	—
DINCOS	Bit 30	R/W	<p>目的增量偏移尺寸</p> <p>0: 外设地址计算的偏移量与DWSEL连接</p> <p>1: 外设地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果比特DINC=0, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p> <p>如果选择直接模式, 或者如果DBUSEL与“00”不同, 则在启用DMA通道时(位EN='1'), 硬件会强制降低该位。</p>
DBUSEL<2:0>	Bits 29-27	R/W	<p>目的地传输突发选择</p> <p>000 : SINGLE</p> <p>001 : Reserved</p> <p>010 : WRAP4</p> <p>011 : INCR4</p> <p>100 : WRAP8</p> <p>101 : INCR8</p> <p>110 : WRAP16</p> <p>111 : INCR16</p>
DDWSEL<1:0>	Bits 26-25	R/W	<p>目的地传输数据宽度选择</p> <p>00: 为每个DMA操作传送一个字节 (8位)。</p> <p>01: 每一个DMA操作都传输一个半字 (16位)。</p> <p>10: 为每个DMA操作传送一个字 (32位)。</p> <p>11: 保留</p>
DINC	Bit 24	R/W	<p>目的地转移增量模式</p> <p>0: 禁用目的地址增量模式</p> <p>1: 启用目标地址增量模式</p>
—	Bit 23	—	—
SINCOS	Bit 22	R/W	<p>源增量偏移尺寸</p> <p>0: 源地址计算的偏移量与SWSEL链接</p> <p>1: 源地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果位SINC=“0”, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p>

			如果选择直接模式，或者如果SBUSEL与“00”不同，则当DMA通道被启用时（位EN='1'），硬件会强制降低该位。
SBUSEL<2:0>	Bits 21-19	R/W	源传输突发选择 000: SINGLE 001: 保留 010: WRAP4 011: INCR4 100: WRAP8 101: INCR8 110: WRAP16 111: INCR16
SDWSEL<1:0>	Bits 18-17	R/W	源传输数据宽度选择 00: 为每个DMA操作传送一个字节（8位）。 01: 每一个DMA操作都传输一个半字（16位）。 10: 为每个DMA操作传送一个字（32位）。 11: 保留
SINC	Bit 16	R/W	源转移增量模式 0: 源地址增量模式禁用 1: 启用源地址增量模式
PHSS	Bits 15-9	R/W	外设握手软件选择
CHPRI	Bits 8-6	R/W	DMA通道优先级 优先级的值是从5到0，设定数值越小，优先权越高。
MODESEL<1:0>	Bits 5-4	R/W	DMA模式选择 00: 内存到内存模式（内存到内存）。 01: 外设存储器模式（外设存储器）。 10: 内存到外设模式（内存到外设）。 11: 保留
DIRMDEN	Bit 3	R/W	直接模式使能 0: 禁用直接模式 1: 启用直接模式 这个比特是受保护的，只有当EN是“0”时才能写入。
PFCTRL	Bit 2	R/W	外设流量控制器 该位由软件设置和清除。 0: DMA是流量控制器 1: 外围设备是流量控制器 这个比特是受保护的，只有当EN是“0”时才能写入。 当选择存储器到存储器模式（位DIR [1:0]=10）时，该位通过硬件自动地被强制为0。
CIRC	Bit 1	R/W	循环模式 该位由软件设置和清除，并可通过硬件清除。

			<p>0: 禁用循环模式 1: 启用循环模式 当外围设备是流量控制器(位PFCTRL=1)并且DMA通道被启用(位EN=1)时, 该位被硬件自动强制为0。</p>
CHEN	Bit 0	R/W	<p>DMA通道启用 将此位设置为“1”使DMA的操作得以实现。如果该位被清除, DMA将忽略所有DMA请求, 并迫使总线主机进入空闲状态。</p>

8.5.2.25 DMA 信道 4 源地址寄存器 (DMA_SAR4)

DMA 通道 4 源地址寄存器 (DMA_SAR4)																															
偏移地址: 64 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR<31:0>																															

SAR<31:0>	Bits 31-0	R/W	DMA传输源地址寄存器 此字段指示DMA的32位源地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----------	-----------	-----	--

8.5.2.26 DMA 通道 4 目的地址寄存器 (DMA_DAR4)

DMA 通道 4 目的地址寄存器 (DMA_DAR4)																															
偏移地址: 68 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR<31:0>																															

DAR<31:0>	Bits 31-0	R/W	传输目的地址寄存器 此字段指示DMA的32位目标地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----------	-----------	-----	---

8.5.2.27 DMA 通道 4 传输字节计数寄存器 (DMA_BCR4)

DMA 通道 4 传输字节计数寄存器 (DMA_BCR4)																															
偏移地址: 6C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBCR<15:0>																BCR<15:0>															

CBCR<15:0>	Bits 31-16	R	DMA 当前传输字节数
BCR<15:0>	Bits 15-0	R/W	DMA传输字节计数寄存器 此字段指示DMA的16位传输字节计数。 这些位是写保护的，并且只能在比特EN = “0” 时写入。

8.5.2.28 DMA 通道 5 控制和状态寄存器 (DMA_CSR5)

DMA 通道 5 控制和状态寄存器 (DMA_CSR5)																																				
偏移地址: 70 _H																																				
复位值: 00000000_00000000_00000000_00000000 _B																																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
DINCOS		DBUSEL<2:0>			DDWSEL<1:0>			DINC		SINCOS		SBUSEL<2:0>			SDWSEL<1:0>		SINC						PHSS<6:0>						CHPRI<2:0>		MODESEL<1:0>		DIRMDEN	PFCTRL	CIRC	CHEN

—	Bit 31	—	—
DINCOS	Bit 30	R/W	<p>目的增量偏移尺寸</p> <p>0: 外设地址计算的偏移量与DWSEL连接</p> <p>1: 外设地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果比特DINC=0, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p> <p>如果选择直接模式, 或者如果DBUSEL与“00”不同, 则在启用DMA通道时(位EN='1'), 硬件会强制降低该位。</p>
DBUSEL	Bits 29-27	R/W	<p>目的地传输突发选择</p> <p>000 : SINGLE</p> <p>001 : Reserved</p> <p>010 : WRAP4</p> <p>011 : INCR4</p> <p>100 : WRAP8</p> <p>101 : INCR8</p> <p>110 : WRAP16</p> <p>111 : INCR16</p>
DDWSEL	Bits 26-25	R/W	<p>目的地传输数据宽度选择</p> <p>00: 为每个DMA操作传送一个字节 (8位)。</p> <p>01: 每一个DMA操作都传输一个半字 (16位)。</p> <p>10: 为每个DMA操作传送一个字 (32位)。</p> <p>11: 保留</p>
DINC	Bit 24	R/W	<p>目的地转移增量模式</p> <p>0: 禁用目的地址增量模式</p> <p>1: 启用目标地址增量模式</p>
—	Bit 23	—	—
SINCOS	Bit 22	R/W	<p>源增量偏移尺寸</p> <p>0: 源地址计算的偏移量与SWSEL链接</p> <p>1: 源地址计算的偏移量固定为4 (32位对齐)。</p> <p>如果位SINC=“0”, 这个比特就没有意义了。</p> <p>该位受到保护, 仅在EN=“0”时才可写入。</p>

			如果选择直接模式，或者如果SBUSEL与“00”不同，则当DMA通道被启用时（位EN='1'），硬件会强制降低该位。
SBUSEL	Bits 21-19	R/W	源传输突发选择 000: SINGLE 001: 保留 010: WRAP4 011: INCR 100: WRAP8 101: INCR8 110: WRAP16 111: INCR16
SDWSEL	Bits 18-17	R/W	源传输数据宽度选择 00: 为每个DMA操作传送一个字节（8位）。 01: 每一个DMA操作都传输一个半字（16位）。 10: 为每个DMA操作传送一个字（32位）。 11: 保留
SINC	Bit 16	R/W	源转移增量模式 0: 源地址增量模式禁用 1: 启用源地址增量模式
PHSS	Bits 15-9	R/W	外设握手软件选择
CHPRI	Bits 8-6	R/W	DMA通道优先级 优先级的值是从5到0，设定数值越小，优先权越高。
MODESEL	Bits 5-4	R/W	DMA模式选择 00: 内存到内存模式（内存到内存）。 01: 外设存储器模式（外设存储器）。 10: 内存到外设模式（内存到外设）。 11: 保留
DIRMDEN	Bit 3	R/W	直接模式使能 0: 禁用直接模式 1: 启用直接模式 这个比特是受保护的，只有当EN是“0”时才能写入。
PFCTRL	Bit 2	R/W	外设流量控制器 该位由软件设置和清除。 0: DMA是流量控制器 1: 外围设备是流量控制器 这个比特是受保护的，只有当EN是“0”时才能写入。 当选择存储器到存储器模式（位DIR [1:0]=10）时，该位通过硬件自动地被强制为0。
CIRC	Bit 1	R/W	循环模式 该位由软件设置和清除，并可通过硬件清除。

			<p>0: 禁用循环模式 1: 启用循环模式 当外围设备是流量控制器(位PFCTRL=1)并且DMA通道被启用(位EN=1)时, 该位被硬件自动强制为0。</p>
CHEN	Bit 0	R/W	<p>DMA通道启用 将此位设置为“1”使DMA的操作得以实现。如果该位被清除, DMA将忽略所有DMA请求, 并迫使总线主机进入空闲状态。</p>

8.5.2.29 DMA 信道 5 源地址寄存器 (DMA_SAR5)

DMA 通道 5 源地址寄存器 (DMA_SAR5)																															
偏移地址: 74 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR<31:0>																															

SAR	Bits 31-0	R/W	DMA传输源地址寄存器 此字段指示DMA的32位源地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----	-----------	-----	--

8.5.2.30 DMA 通道 5 目的地址寄存器 (DMA_DAR5)

DMA 通道 5 目的地址寄存器 (DMA_DAR5)																															
偏移地址: 78 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR<31:0>																															

DAR	Bits 31-0	R/W	传输目的地址寄存器 此字段指示DMA的32位目标地址。 这些位是写保护的，并且只能在比特EN = “0”时写入。
-----	-----------	-----	---

8.5.2.31 DMA 通道 5 传输字节计数寄存器 (DMA_BCR5)

DMA 通道 5 传输字节计数寄存器 (DMA_BCR5)																															
偏移地址: 7C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBCR<15:0> [^]																BCR<15:0> [^]															

CBCR	Bits 31-16	R	DMA 当前传输字节数
BCR	Bits 15-0	R/W	DMA传输字节计数寄存器 此字段指示DMA的16位传输字节计数。 这些位是写保护的，并且只能在比特EN = “0” 时写入。

第9章 硬件除法器 (HDIV)

9.1 概述

硬件除法器(HDIV)对高效率应用很有帮助。硬件除法器为一有符号整数除法器同时含商数与余数输出。

HDIV 主要特点:

- ◆ 无符号与有符号(2 的补码) 整数计算
- ◆ 32-位被除数含 32-位除法器计算容量
- ◆ 32-位商数与 32-位余数输出
- ◆ 由零警示标志除
- ◆ 2~17 HCLK 时钟取自单一周期计算
- ◆ 写除数以触发计算
- ◆ 当读入商数与余数时自动等待计算预备完成

9.2 功能描述

欲使用硬件除法器，首先须设定被除数。接着设定除数与硬件除法器将在除数被写入后自动触发计算。通过读 HDIV_DIVQR 与 HDIV_DIVRR 寄存器可得到计算结果, 包含商数与余数。若 CPU 在硬件除法器计算完成前读取 HDIV_DIVQR 或 HDIV_DIVRR，CPU 将保留直到硬件除法器完成计算。因此，CPU 皆可在触发一硬件除法器计算后取得作用结果，而不会发生软件延迟。

若除数为 0，HDIV_DIVSTAT 之 DIV0 标志将被设定

被除数为 32 位有符号整数且除数为 32 位有符号整数。商数为 32 位有符号整数 而余数为 32 位有符号整数。

下图标示硬件除法器之运算流程。为了计算 X / Y ，CPU 必须写 X 至 HDIV_DIVDR 寄存器，接着写 Y 至 HDIV_DIVSR。CPU 可读 HDIV_DIVQR 与 HDIV_DIVRR 寄存器以取得 HDIV_DIVSR 被写之后的计算结果。

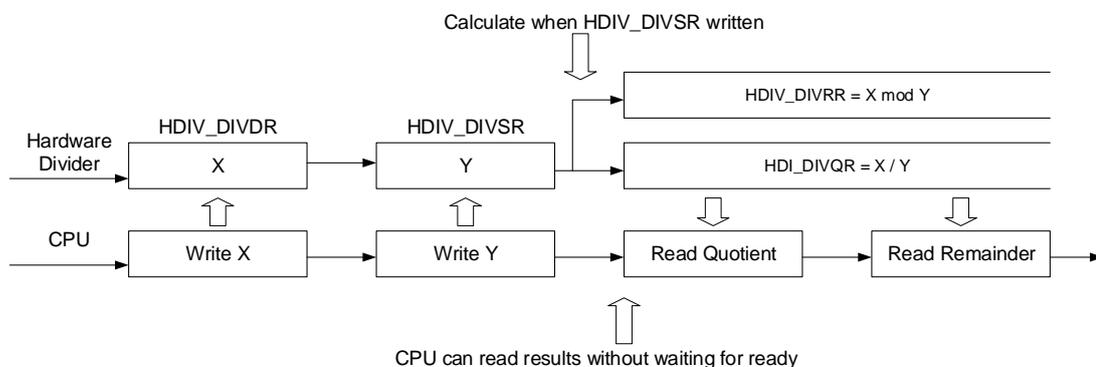


图 9-1 硬件除法器运作流程图

9.3 特殊功能寄存器

9.3.1 寄存器列表

HDIV 寄存器列表			
名称	偏移地址	类型	描述
HDIV_DIVDR	0000 _H	R/W	HDIV 被除数寄存器
HDIV_DIVSR	0004 _H	R/W	HDIV 除数寄存器
HDIV_DIVQR	0008 _H	R	HDIV 商结果寄存器
HDIV_DIVRR	000C _H	R	HDIV 余数结果寄存器
HDIV_DIVSTAT	0010 _H	R/W	HDIV 除法器状态寄存器

9.3.2 寄存器描述

9.3.2.1 HDIV 被除数寄存器 (HDIV_DIVDR)

HDIV 被除数寄存器 (HDIV_DIVDR)																															
偏移地址: 00 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVD<31:0>																															

DIVD	Bits 31-0	R/W	被除数 该寄存器在计算开始之前给出除法器的被除数。
------	-----------	-----	------------------------------

9.3.2.2 HDIV 除数寄存器 (HDIV_DIVSR)

HDIV 除数来源寄存器 (HDIV_DIVSR)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVS<31:0>																															

DIVS	Bits 31-0	R/W	除数 该寄存器在计算开始之前给出除法器的除数。 注意: 当写入此寄存器时, 硬件除法器将开始计算。
------	-----------	-----	---

9.3.2.3 HDIV 商结果寄存器 (HDIV_DIVQR)

HDIV 商结果寄存器 (HDIV_DIVQR)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVQ<31:0>																															

DIVQ	Bits 31-0	R	商结果 该寄存器在计算完成后保持除法器的商结果。
------	-----------	---	-----------------------------

9.3.2.4 HDIV 余数结果寄存器 (HDIV_DIVRR)

HDIV 余数结果寄存器 (HDIV_DIVRR)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVR<31:0>																															

DIVR	Bits 31-0	R	余数结果 该寄存器在计算完成后保存除法器的余数结果。
------	-----------	---	-------------------------------

第10章 通用异步收发器 (UART)

10.1 概述

通用异步收发器 (UART) 提供了一个灵活的方式, 使 MCU 可以与外部设备通过工业标准 NRZ 的形式实现全双工异步串行数据通讯。UART 可以使用小数波特率发生器, 提供了超宽的波特率设置范围。

UART 支持异步通讯模式和半双工单线通讯, 也支持 LIN (本地互连网络)、智能卡协议、IrDA (红外数据协会) SIRENDEC 规范和 modem 流控操作 (CTS_n/RTS_n), 同时还支持多机通讯方式。

可以使用 DMA 实现多缓冲区设置, 从而能够支持高速数据通讯。

10.2 特性

- ◆ 全双工异步通信
- ◆ 16-byte 接收和发送 FIFOs
- ◆ 兼容 16C550 标准
- ◆ 可软件控制接收 FIFO 触发点
- ◆ 通信波特率可设置
- ◆ 支持自动波特率检测
- ◆ 十七个中断源
- ◆ 可与 DMA 使用
 - ◇ 利用 DMA 功能将收/发字节缓冲到保留的 SRAM 空间
- ◆ 内置小数波特率发生器, 覆盖范围广, 不需要特定值的外部晶体
 - ◇ 在时钟频率为 48 MHz 下, 可编程收发波特率高达 3 MBps, 最低可达 732.4 Bps
 - ◇ 在时钟频率为 4 MHz 下, 可编程收发波特率高达 250 KBps, 最低可达 61 Bps
- ◆ 支持硬件自动流控制/流控制功能 (CTS_n、RTS_n), RTS_n 控制流触发点可程序设计
 - ◇ Modem 硬件自动控制
 - ◇ RS485 发送使能控制
- ◆ 支持 CTS_n 唤醒功能
- ◆ 支持 IrDA SIR 模式
 - ◇ 支持 3/16 位周期调制
- ◆ 支持 RS-485
 - ◇ 支持 9-位模式
 - ◇ 多处理器通信

- ◆ 完全可程序设计的串行接口特性
 - ◇ 可程序设计数据位个数，即 5, 6, 7, 8, 9 位，9 位用于 RS485 模式
 - ◇ 校验位，奇、偶、无校验
 - ◇ 停止位长度可程序设计：1, 2 位，在智能卡模式中支持 0.5, 1.5 位
 - ◇ 可设置高位在前或低位在前
- ◆ 单线半双工通讯
- ◆ 交换 Tx/Rx pin 配置
- ◆ LIN 主机的断开信号发送能力和 LIN 从机的断开信号检测能力
 - ◇ 将 UART 设置为 LIN 模式时，有 13 位的断开信号发生器与断开信号检测功能
- ◆ 智能卡模式
 - ◇ 支持 ISO/IEC7816-3 标准定义的 T=0 和 T=1 智能卡异步协议
 - ◇ 智能卡使用的 1.5 停止位长度
- ◆ 支持 ModBus 通讯
 - ◇ CR/LF 字符识别
 - ◇ 超时检测功能
- ◆ 噪声侦测

10.3 结构图

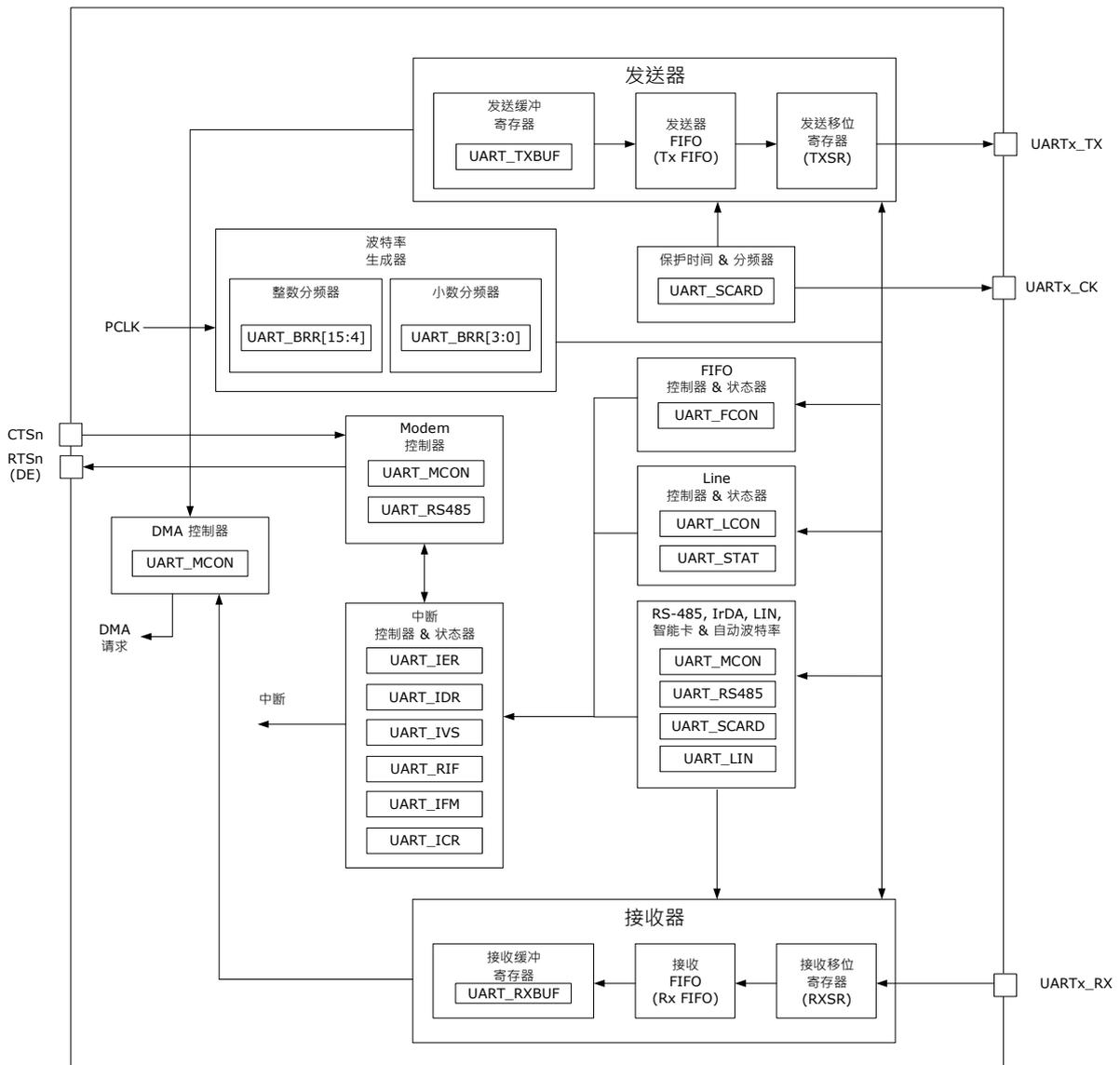


图 10-1 UART 框图

10.4 功能描述

接口与外部设备通过三个引脚相连。任何 UART 双向通讯要求最少有两个引脚：接收数据输入（RX）和发送数据输出（TX）

RX: 接收数据输入，是串行数据的输入口。使用过采样技术来完成数据恢复，以区别输入数据和噪声。

TX: 数据发送输出。当发送器被禁止，输出脚回到其 I/O 口配置状态。当发送器被使能，但不发送数据时，TX 脚为高电平输出。在单线和智能卡模式中，这个口线既用于发送数据也用于接收数据。

通过这些引脚，串行数据用数据帧的形式发送和接收：

- ◆ 在发送和接收之前为空闲状态
- ◆ 起始位
- ◆ 数据可通过 **UART_LCON** 寄存器的 MSB 位设定
- ◆ 1, 2 个停止位表明帧的结束(0.5, 1.5 个停止位用于智能卡模式)
- ◆ 采用小数波特率发生器，整数 12 位小数 4 位
- ◆ 一个状态寄存器（**UART_STAT**）
- ◆ 分开的接收和发送数据寄存器（**UART_RXBUF, UART_TXBUF**）
- ◆ 一个波特率寄存器（**UART_BRR**）12 位整数和 4 位小数
- ◆ 一个智能卡寄存器（**UART_SCARD**）用于智能卡模式
- ◆ 一个接收时间寄存器（**UART_RTOR**）侦测输入信号时间并产生中断

下面的引脚在智能卡模式中会用到：

CK: 时钟输出。智能卡模式中，CK 引脚会向智能卡提供时钟。

下列引脚用于支持硬件流控制模式：

CTS_n: 低电平发送，当高电平时作为发送阻塞信号。

RTS_n: 请求发送，表明 UART 已经准备好接收数据（低电平的时候）。

下列引脚在 RS485 驱动使能控制的时候会用到：

DE: 驱动使能将外部收发器的发送模式激活。

注：DE 和 RTS_n 共享同一个外部引脚。

10.4.1 具体功能配置

UART modes/feature	UART	SUART
Modem 的硬件控制	v	v
使用 DMA 实现连续通讯	v	v
多机通讯模式	v	v
智能卡模式	v	
单线半双工模式	v	
IrDA SIR 模块	v	
LIN 模式	v	
超时检测功能	v	v
Modbus 通讯	v	v
自动波特率检测模式	v	v
RS485 的驱动使能信号	v	v
UART 数据宽度	5, 6, 7, 8, 9bits	

表 10-1 UART/SUART 具体功能配置

1. V:支持
2. 支持 RS485 9bit 模式

10.4.2 功能描述

配置 LCON 寄存器中的 DLS 位可选择 8-5 位字长。

默认设置中，发送和接收的起始位都是低电平。而停止位都是高电平。

这个逻辑可以在 LCON 寄存器的 TXINV 与 RXINV 位设置为反向。

- ◆ 8 位字符宽度: DLS [1:0] = 00
- ◆ 7 位字符宽度: DLS [1:0] = 01
- ◆ 6 位字符宽度: DLS [1:0] = 10
- ◆ 5 位字符宽度: DLS [1:0] = 11

注：第 9 位使用于 RS485 多处理器模式。

空闲符号被视为完全由 1 组成的完整的数据帧，后面跟着包含了数据的下一帧的开始位（1 的位数也包括了停止位的位数）。

断开符号被视为在一个帧周期内全部收到 0（包括停止位期间，也是 0）。在断开帧结束时，发送器会再插入 2 个停止位。

发送和接收由一个共享的波特率发生器驱动，当发送器和接收器的使能位分别置 1 时，分别为其产生时钟。

下面是每个模块的详细说明。

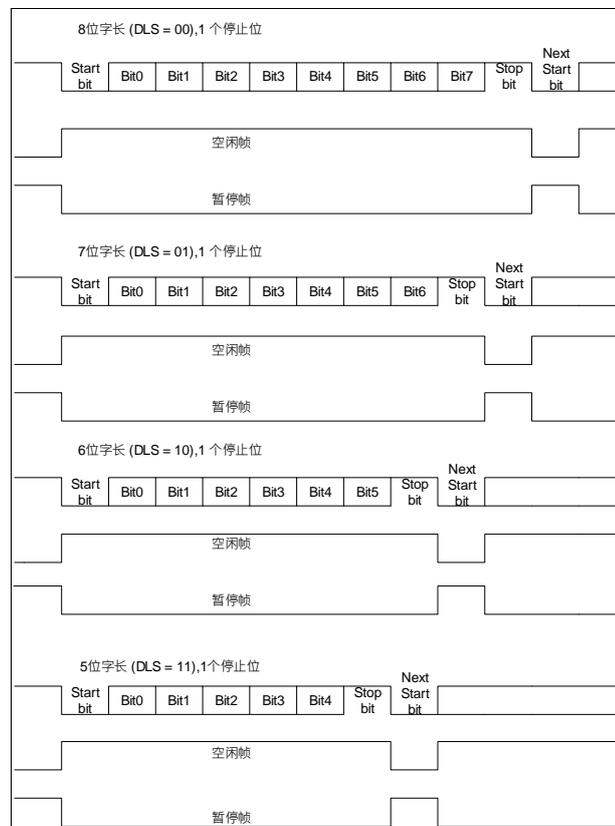


图 10-2 字长编程

10.4.3 发送器

发送器根据 LCON 寄存器中的 DLS 位的状态发送 8-5 位的数据字。

当写入 UART_TXBUF 后，发送移位寄存器中的数据在 TX 脚上输出。

在 UART 发送期间，在 TX 引脚上首先移出数据的最低有效位。在此模式里，TXBUF 寄存器充当了一个内部总线和发送移位寄存器之间的缓冲器(TXSR)。

每个字符之前都有一个低电平的起始位，字符结束有停止位，停止位的数目可配置。

UART 支持多种停止位的选择：0.5, 1, 1.5 和 2 个停止位。

注:

1. 在写入 TXBUF 寄存器数据前必须先令 STAT 寄存器中的 TFFULL 位为 0
2. 打开 TX 开关后，数据才能在 TX 脚上输出可配置的停止位随每个字符发送的停止位的位数可以通过 LCON 寄存器中的 STOP 位进行编程。

- ◆ 0.5 个停止位：在智能卡模式下发送和接收数据时使用。
- ◆ 1 个停止位：停止位的位数的默认值。
- ◆ 1.5 个停止位：在智能卡模式下发送和接收数据时使用。
- ◆ 2 个停止位：可用于常规 UART 模式、以及调制解调器模式。

空闲帧包括了停止位。

断开帧可通过 MCON 寄存器的 BKREQ 位产生 10 位低电平(当 DLS=00 时)，9 位低电平(当 DLS=01 时)，8 位低电平(当 DLS=10 时)或者 7 位低电平(当 DLS=11 时)，后跟 2 个停止位。

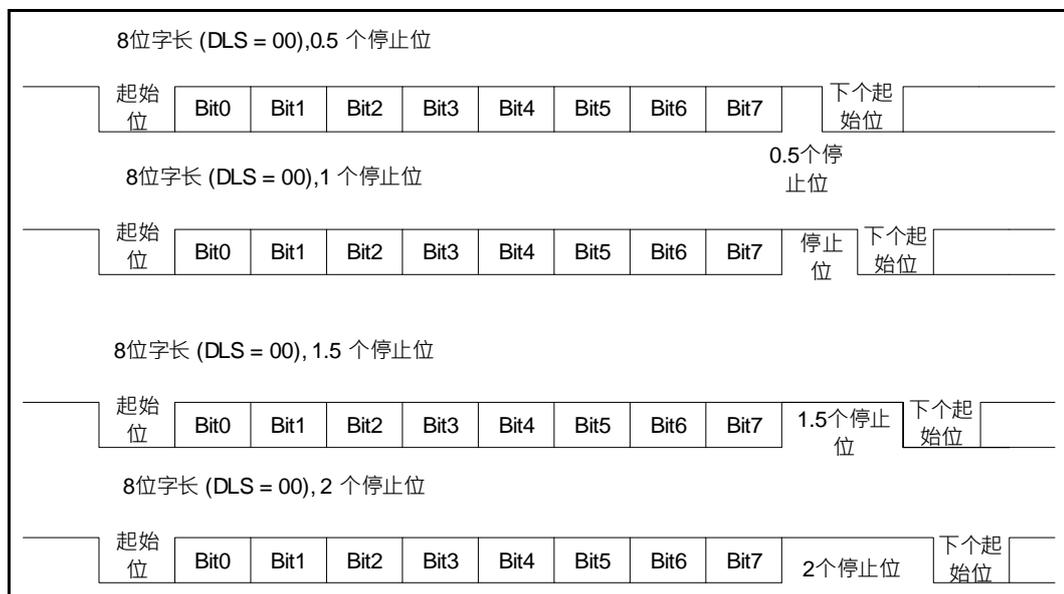


图 10-3 配置停止位

TX FIFO 门槛设置:

设置 FCON 寄存器的 TXTH 位, 可设定 FIFO 的门槛为 0,2,4,8, 当 FIFO 内的数据个数小于门槛会使得 STAT 寄存器的 TFTH 位为 1, 告知用户需要再填入数据, 以避免数据传送中断。

开启 IER 寄存器的 TFTH 位为 1, UART 会判断 FIFO 内的个数是否小于门槛, 使得 RIF 寄存器的 TFTH 位为 1, 产生中断。在产生中断的期间设置 ICR 寄存器的 TFTH 位为 1, 使得 RIF 寄存器的 TFTH 位清除, 若用户未写入新的数据至 FIFO 中, FIFO 内的数据个数仍然小于门槛则不会再次产生中断, 需要重新开启 IER 寄存器的 TFTH 位。

注: 一开始 RIF 寄存器的 TFTH 与 TFEMPTY 位为 0, 当产生 FIFO 内的数据个数小于门槛事件与 FIFO empty 事件时, 使得 TFTH 与 TFEMPTY 位为 1。STAT 寄存器的 TFTH 与 TFEMPTY 位则是反应 TX FIFO 状态。

配置步骤:

1. 设置 LCON 寄存器中的 DLS 位来定义字长。
2. 设置 LCON 寄存器中的 STOP 位设置停止位的位数。
3. 设置 LCON 寄存器中的 PE 与 PS 位设置校验控制开关与极性。
4. 设置 BRR 寄存器选择希望的波特率。
5. 如果采用多缓冲器通信, 配置 MCON 寄存器中的 TXDMAEN 位为 1。按多缓冲器通信中的描述配置 DMA 寄存器。
6. 设置 LCON 寄存器中的 TXEN 位, 使能发送器。
7. 把要发送的数据写进 TXBUF 寄存器(此动作将清除 STAT 寄存器中的 TFEMPTY 位)。
8. 在 TXBUF 寄存器中写入数据字时, 要等待 STAT 寄存器中的 TFFULL 位为 0, 它表示 FIFO 中未满足 16 byte。当需要关闭 UART, 需要确认传输结束 STAT 寄存器中的 TSBUSY 位为 0, 避免破坏最后一次传输。

注: 当 LCON 寄存器的 TXEN 与 RXEN 位为 1 时, 无法写入 LCON 寄存器与 BRR 寄存器

10.4.4 接收器

10.4.4.1 消抖电路

在 UART_RX 引脚上配置了一个防抖电路，设置 LCON 寄存器中的 DBCEN 位开启功能，输入信号须维持至少 8 个高位或低位，才能使得讯号反应至 UART 中，反之则被忽略，如下叙述。

在下图中，SYNC0 是指输入信号由系统时钟一次采样；SYNC1 是指 SYNC0 被系统时钟一次采样；SYNC2 表示 SYNC1 由系统时钟一次采样。SYNC0、SYNC1 和 SYNC2 可以表示成 $x[n] \times T_s$ ， $x[n+1] \times T_s$ 和 $x[n+2] \times T_s$ 。T_s 是系统时钟周期，n 是采样时间。如果关闭防抖模块，时间就可以表示为 $x[n+1] \times T_s$ 。

如果打开防抖模块，SYNC2 信号将进入去抖电路，那么它将被采样与计数(用户设定样本频率和计数时间值)。时间可表示为 $[(SPT+1) \times (FILTCNT+1)] \times T_s$ 。SPT 是采样频率值，FILTCNT 是计数次数。当 SYNC1 和 SYNC2 不相等时，计数值将被清零。如果计数值溢满 FILTCNT 寄存器，防抖模块将输出信号。

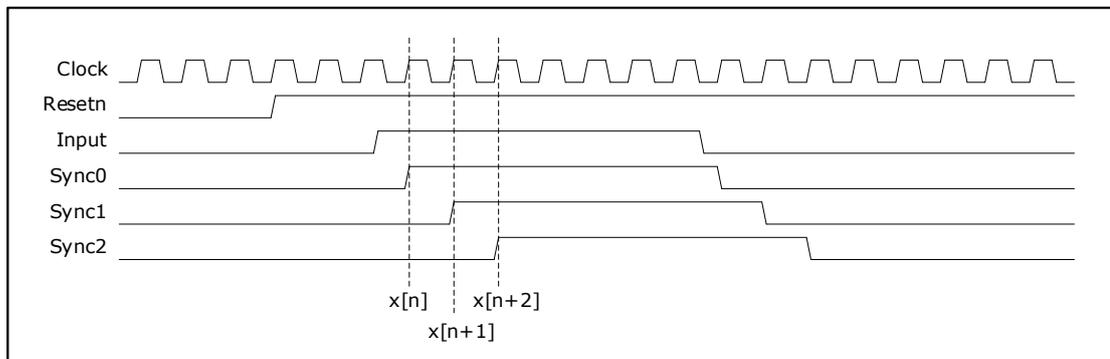


图 10-4 防抖动波形

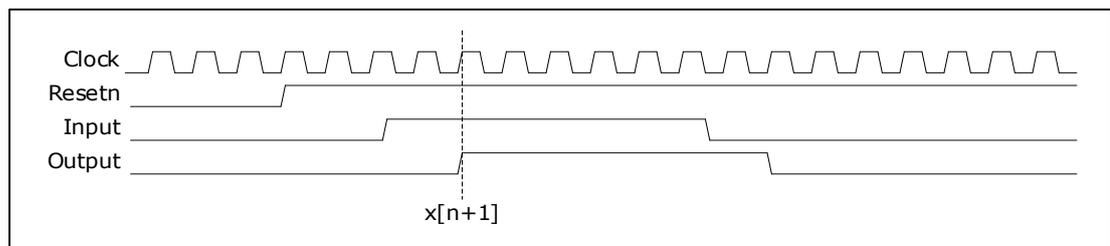


图 10-5 防抖动输出

10.4.4.2 起始位侦测

接收器根据 LCON 寄存器中 DLS 位的状态接收 8-5 位的数据字。

起始位侦测

在 UART 中，如果辨认出一个特殊的采样序列，那么就认为侦测到一个起始位。

该序列为：1 1 1 0 X 0 X 0 X 0 0 0 0 X X X X X X X

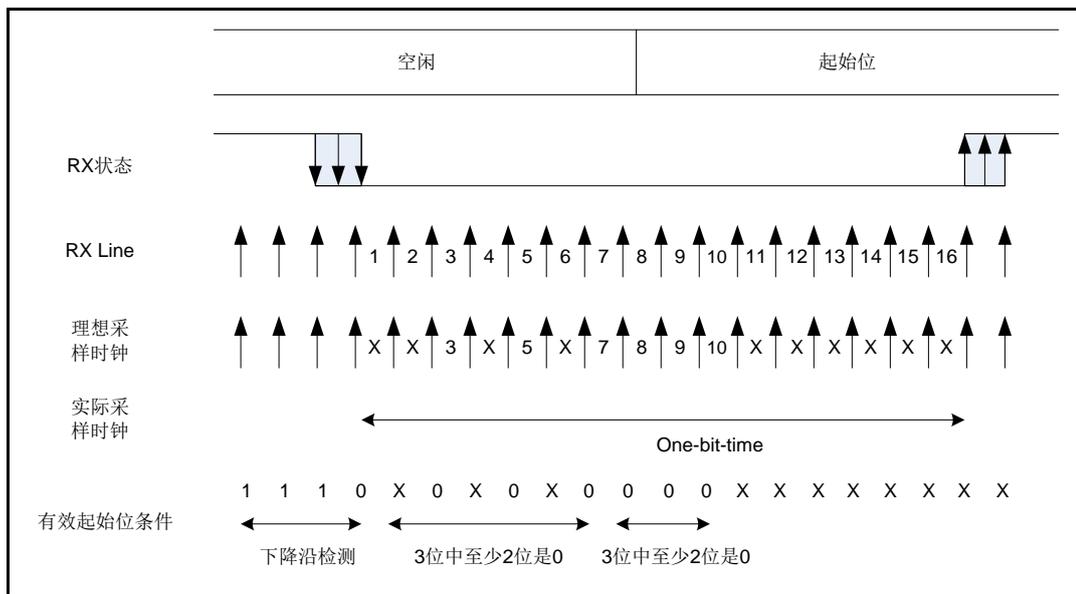


图 10-6 起始位侦测

注：如果该序列不完整，那么接收端将退出起始位侦测并回到空闲状态(不设置标志位)开始等待下降沿。

1. 如果 3 个采样点都为'0'(在第 3、5、7 位的第一次采样，和在第 8、9、10 的第二次采样都为'0')，则确认收到起始位。
2. 如果两次 3 个采样点上有 2 个是'0'(第 3、5、7 位的采样点和第 8、9、10 位的采样点)，那么起始位仍然是有效的。
3. 如果不能满足这个条件，则中止起始位的侦测过程，接收器会回到空闲状态。
4. 如果两次 3 个采样点上有 2 个是'1'(第 3、5、7 位的采样点和第 8、9、10 位的采样点)，那么起始位是无效的，将退出起始位侦测并回到空闲状态，并会设置噪声标志位。

RX FIFO 阈值设置:

设置 FCON 寄存器的 RXTH 位，可设定 FIFO 的阈值为 1,4,8,14，当 FIFO 内的数据个数大于阈值会使得 STAT 寄存器的 RFTH 位为 1，告知用户需要读取数据，以避免数据遗失。

开启 IER 寄存器的 RFTH 位为 1，UART 会判断 FIFO 内的个数是否大于阈值，使得 RIF 寄存器的 RFTH 位为 1，产生中断。在产生中断的期间设置 ICR 寄存器的 RFTH 位为 1，使得 RIF 寄存器的 RFTH 位清除，若用户未读取数据，FIFO 内的数据个数仍然大于阈值则不会再次产生中断，需要重新开启 IER 寄存器的 RFTH 位。

注：一开始 RIF 寄存器的 RFTH 位为 0，当产生 FIFO 内的数据个数大于阈值事件时，使得 RFTH 位为 1。SCON 寄存器的 RFTH 位则是反应 RX FIFO 状态。

配置步骤:

1. 设置 LCON 寄存器中的 DLS 位来定义字长。
2. 设置 LCON 寄存器中的 STOP 位设置停止位的位数。
3. 设置 LCON 寄存器中的 PE 与 PS 位设置校验控制开关与极性。
4. 设置 BRR 寄存器选择希望的波特率。
5. 如果采用多缓冲器通信，配置 MCON 寄存器中的 RXDMAEN 位为 1。按多缓冲器通信中的描述配置 DMA 寄存器。
6. 设置 LCON 寄存器中的 RXEN 位，这将启动接收器，使它开始寻找起始位。

当一个字符被接收到时，STAT 寄存器的 RFEMPTY 位被置 0。它表明移位寄存器的内容被转移到 RX FIFO 中。换句话说，数据已经被接收并且可以被读出(包括与之有关的错误标志)。

这时如果 IER 寄存器的 RFTH 位是 1，且 Rx FIFO 阈值为 1，将会引起中断请求。

在接收期间如果检测到帧错误，噪音或溢出错误，错误标志将被置起。RXBERR 标志也会和 RFTH 一起被置 1。

在多缓冲器通信时，RFEMPTY 在每个字节接收后被清零，并由 DMA 对数据寄存器的读操作而置起。

RFFULL 位必须在下一字符接收结束前被清零，以避免溢出错误。

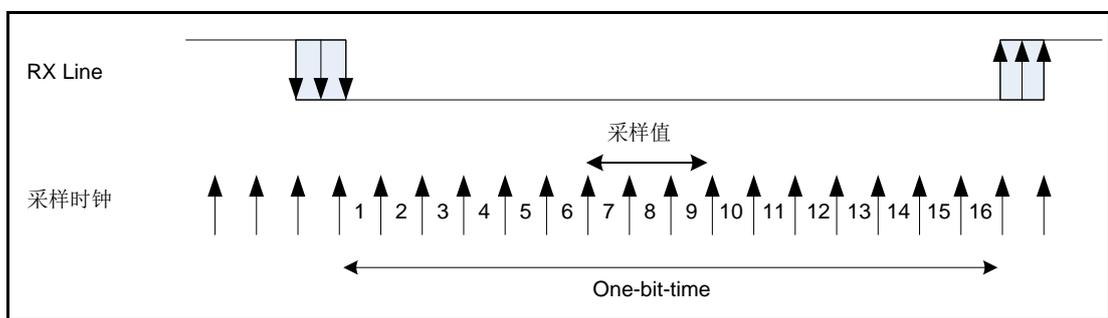


图 10-7 数据采样

帧错误

当以下情况发生时检测到帧错误：

由于没有同步上或大量噪音的原因，停止位没有在预期的时间上接收和识别出来。

当帧错误被检测到时：

1. FERR 位被硬件置 1
2. 此时读取的 RXBUF 寄存器数据可能有错。
3. 寄存器 STAT 中的 FERR 位显示当前从 FIFO 读取的 RXBUF 寄存器是否为帧错误。
4. 寄存器 RIF 中的 RXBERR 位则会在接收的过程中被置起，若 IER 中的 RXBERR 位为 1 则会产生中断。

奇偶位错误

当以下情况发生时检测到奇偶性错误：

由于没有同步上或大量噪音的原因，奇偶位没有在预期的时间上接收和识别出来。

当奇偶位错误被检测到时：

1. PERR 位被硬件置 1
2. 此时读取的 RXBUF 寄存器数据可能有错。
3. 寄存器 STAT 中的 PERR 位显示当前从 FIFO 读取的 RXBUF 寄存器是否为帧错误。
4. 寄存器 RIF 中的 RXBERR 位则会在接收的过程中被置起，若 IER 中的 RXBERR 位为 1 则会产生中断。

断开错误

当以下情况发生时检测到断开错误：

由于没有同步上或大量噪音的原因，资料与停止位为 0 且没有在预期的时间上接收和识别出来。

当断开错误被检测到时：

1. BKERR 位被硬件置 1
2. 此时读取的 RXBUF 寄存器数据可能有错。
3. 寄存器 STAT 中的 BKERR 位显示当前从 FIFO 读取的 RXBUF 寄存器是否为帧错误。
4. 这个错误并不会产生中断。

溢出错误

当以下情况发生时检测到溢出错误：

由于 FIFO 已满 16byte 还没有被读取，而又接收到一个字符，则发生溢出错误。

当溢出错误被检测到时：

1. RFOERR 位被硬件置 1
2. RXBUF 内容将不会丢失，读取 RXBUF 寄存器仍能得到先前的数据。
3. 移位寄存器中以前的内容将被覆盖，随后接收的数据将丢失。
4. 寄存器 RIF 中的 RFOERR 位则会在接收的过程中被置起，若 IER 中的 RFOERR 位为 1 则会产生中断。

采样值	接收到的值
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

表 10-2 采样后的噪声侦测值

10.4.5 状态寄存器

在 UART 中配置了 15 种 UART 状态提供用户使用，叙述如下

◆ PERR (Parity error):

当所接收的字符没有正确的校验字节，将生成一个奇偶校验错误。该错误与 FIFO 顶部的字符有关，即读取 RXBUF 寄存器的字符。

◆ FERR (Frame error):

当接收到的字符停止位为 0 时，产生帧错误。该错误与 FIFO 顶部的字符有关，即读取 RXBUF 寄存器的字符。

◆ BKERR (Break error):

当接收到的字符与字符停止位为 0 时，产生断开错误。该错误与 FIFO 顶部的字符有关，即读取 RXBUF 寄存器的字符。

◆ CTSSTA (CTS_n status error):

清除发送。此位为 CTS_n pin 上的状态。当 CTSSTA 位为 0，这是一个迹象表明调制解调器和数据设备已准备好与 UART 进行数据交换。

◆ RSBUSY (Rx shifter register busy):

当此位为 1 表示接收器正在接收字符，为 0 则为接收完成。

◆ RFTH(Rx FIFO 门槛):

当此位为 1 表示 Rx FIFO 内数据大于门槛(设置 FCR 寄存器的 RXTH，可设定门槛为 1,4,8,14)，告知用户需要读取 RXBUF 寄存器。

◆ RFEMPTY (Rx FIFO empty):

当此位为 1 表示 Rx FIFO 内无任何字符，为 0 则为已接收 1 个以上的字符。

◆ RFFULL (Rx FIFO full):

当此位为 1 表示 Rx FIFO 内已有 16 个字符，需要被读取。

◆ RFOERR (Rx FIFO overrun):

当此位为 1 表示 Rx FIFO 内已有 16 个字符，且又再接收 1 个字符，此时 FIFO 内字符不会丢失，接收的字符则会被丢失。

◆ RFUERR (Rx FIFO underrun):

当此位为 1 表示 Rx FIFO 内无任何字符，且又被读取。

◆ TSBUSY (Tx shifter register busy):

当此位为 1 表示发送器正在传送字符，为 0 则为传送完成，当写入第一个数据至 TXBUF 寄存器就会使得 TSBUSY 位为 1。

◆ TFTH(Tx FIFO 门槛):

当此位为 1 表示 Tx FIFO 内数据小于门槛(设置 FCR 寄存器的 TXTH，可设定门槛为 0,2,4,8)，告知用户需要写入 TXBUF 寄存器。

◆ TFEMPTY (Tx FIFO empty):

当此位为 1 表示 Tx FIFO 内无任何字符，为 0 则为准备发送 1 个以上的字符。

◆ TFFULL (Tx FIFO full):

当此位为 1 表示 Tx FIFO 内已有 16 个字符，准备发送。

◆ TFOERR (Tx FIFO overrun):

当此位为 1 表示 Tx FIFO 内已有 16 个字符，且又在写入 1 个字符，此时 FIFO 内字符不会丢失，写入的字符则会被丢失。

10.4.6 波特率产生器

接收器和发送器的波特率在 UARTDIV 的整数和小数寄存器中的值应设置成相同。

UARTDIV=UART.BRR.

TX/RX baud = PCLK / UARTDIV

注: 1. 当 LCON 寄存中的 RXEN 与 TXEN 为 1 时, BRR 寄存器无法被写入。2. 当 BRR[15:4]=0 时, 无法运行。

如何计算波特率寄存器的值

- ◆ 在 4 MHz 下, 为了得到 115200 波特
 - $UARTDIV = 4000000/115200 = 34.7$
 - $BRR[15:0] = UARTDIV = 35d = 23h$ (四舍五入)
- ◆ 在 8 MHz 下, 为了得到 9600 波特
 - $UARTDIV = 8000000/9600 = 833.33$
 - $BRR[15:0] = UARTDIV = 833d = 341h$ (四舍五入)
- ◆ 在 16 MHz 下, 为了得到 1200 波特
 - $UARTDIV = 16000000/1200 = 13333.33$
 - $BRR[15:0] = UARTDIV = 13333.33d = 3415h$ (四舍五入)
- ◆ 在 24 MHz 下, 为了得到 460800 波特
 - $UARTDIV = 24000000/460800 = 52.08$
 - $BRR[15:0] = UARTDIV = 52d = 34h$ (四舍五入)
- ◆ 在 48 MHz 下, 为了得到 115200 波特
 - $UARTDIV = 48000000/115200 = 416.66$
 - $BRR[15:0] = UARTDIV = 417d = 1A1h$ (四舍五入)
- ◆ 在 48 MHz 下, 为了得到 921600 波特
 - $UARTDIV = 48000000/921600 = 52.08$
 - $BRR[15:0] = UARTDIV = 52d = 34h$ (四舍五入)

波特率	16 倍过采样			
序号	预期值	实际值	BRR	%误差
1	734Bps	733KBps	0xFFCC	0
2	1.2KBps	1.2KBps	0x9C40	0
3	2.4KBps	2.4KBps	0x4E20	0
4	4.8KBps	4.8KBps	0x2710	0
5	9.6KBps	9.6KBps	0x1388	0
6	19.2KBps	19.2KBps	0x9C4	0
7	38.4KBps	38.4KBps	0x4E2	0
8	57.6KBps	57.62KBps	0x341	0.03
9	115.2KBps	115.11KBps	0x1A1	0.08
10	230.4KBps	230.77KBps	0xD0	0.16
11	460.8KBps	461.54KBps	0x68	0.16
12	921.6KBps	923.07KBps	0x34	0.16
13	1.5Mbps	1.5Mbps	0x20	0
14	2Mbps	2Mbps	0x18	0
15	3Mbps	3Mbps	0x10	0

表 10-3 时钟为 48MHz 下，设置波特率时的误差计算

10.4.7 自动波特率侦测

UART 可以根据接收到的一个字符来检测和自动设置 BRR 寄存器的值。自动波特率检测在两种情况下有用：

- ◆ 通讯速度不可知的情况下
- ◆ 使用低精度时钟源，需要在不测量时钟偏差的条件下纠正波特率的时候。

时钟源的频率必须和预期的波特率保持相对的稳定

（过采样率为 16，并且波特率处于 PCLK/65535 和 PCLK/16 之间）。

在打开自动波特率检测之前，字符的内容必须先确认。有两个可能的字符内容，能够通过 MCON 寄存器中的 ABRMOD 位进行选择。具体是：

- ◆ 模式 0 (0x00)：波特率在 UART 的 RX 引脚的两个连续的下降沿上测量（起始位的下降沿和最低数据位的下降沿（LSB））
- ◆ 模式 1 (0x01)：波特率在 UART 的 RX 引脚下降沿和后续上升沿之间（起始位的长度）测量。
- ◆ 模式 2 (0x10)：波特率在 UART 的 Rx 引脚下降沿和后续上升沿之间（起始位的长度加上 bit[0]的长度）测量（e.g. 0xFE）。

将 MCON 寄存器中的 ABREN 位置 1，开启自动波特率检测功能。UART 在 RX 线上等待第一个字符过来。当自动波特率操作结束后，RIF 寄存器中的 ABEND 标志会被硬件自动设置 1，若 IER 寄存器中的 ABEND 位为 1 则会产生中断。

如果线路噪声严重，不能保证得到的波特率是准确的。这时 BRR 值可能是错的或者 ABEND 错误标志会被置 1。在通讯速度超出自动波特率检测范围（位长度不在 0x10 到 0xFFFFF 个时钟周期之间）时也会发生这种情况。

在此模式中配置了两个中断，分别为侦测超时与侦测结束。

在软件并未清除 MCON 寄存器中的 ABREN 的情况下，UART 计数器会持续计数直到 0xFFFFF 后，硬件会自动将 ABREN 清除，并将 RIF 寄存器中的 ABTO 位置起，如果开启中断，则会产生 ABTO 中断。

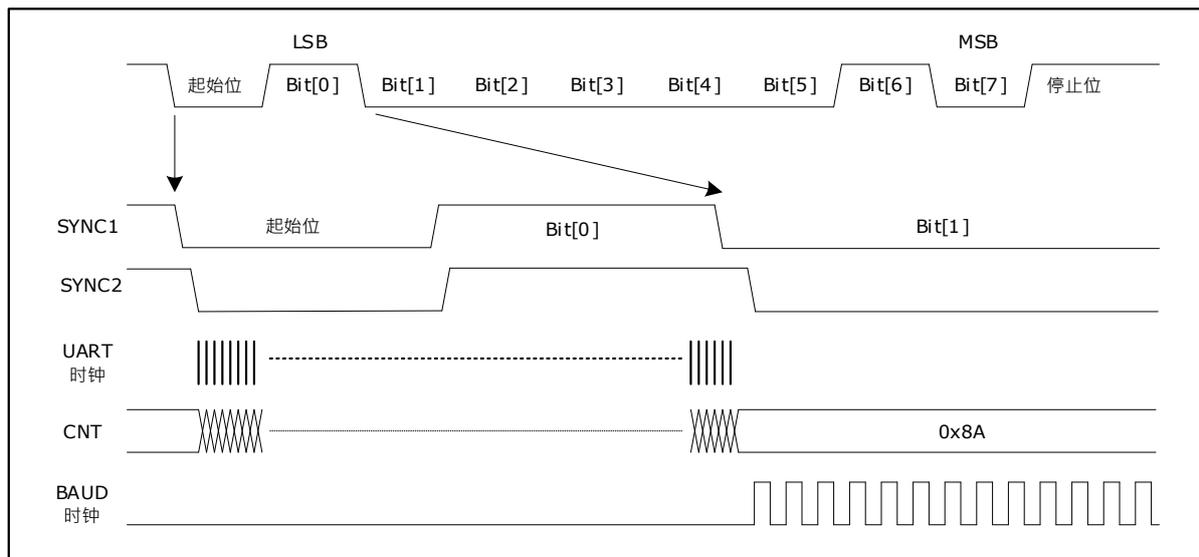


图 10-8 自动波特率侦测模式 0

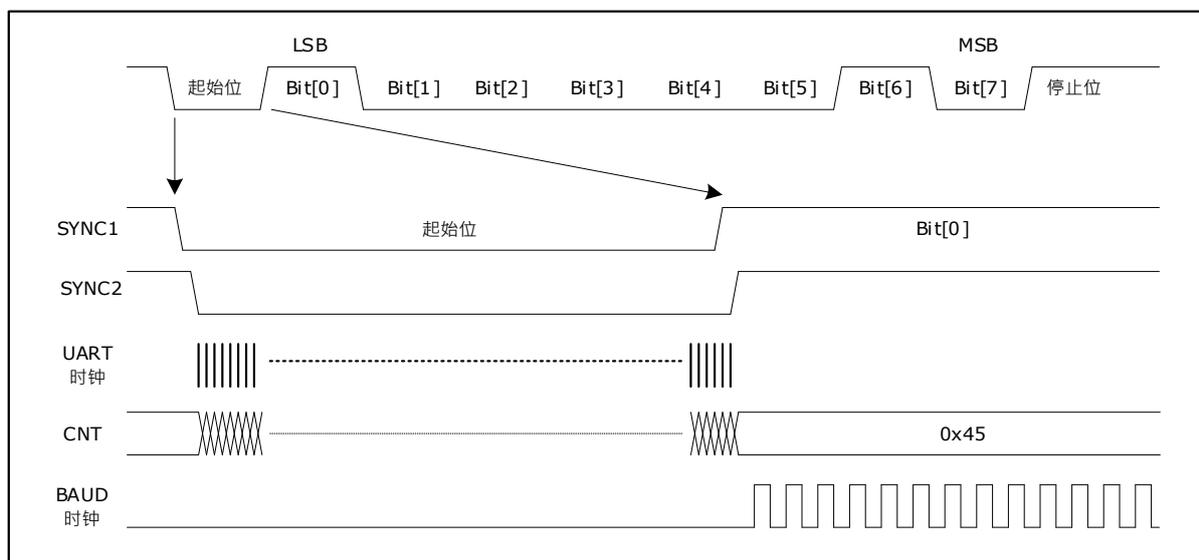


图 10-9 自动波特率侦测模式 1

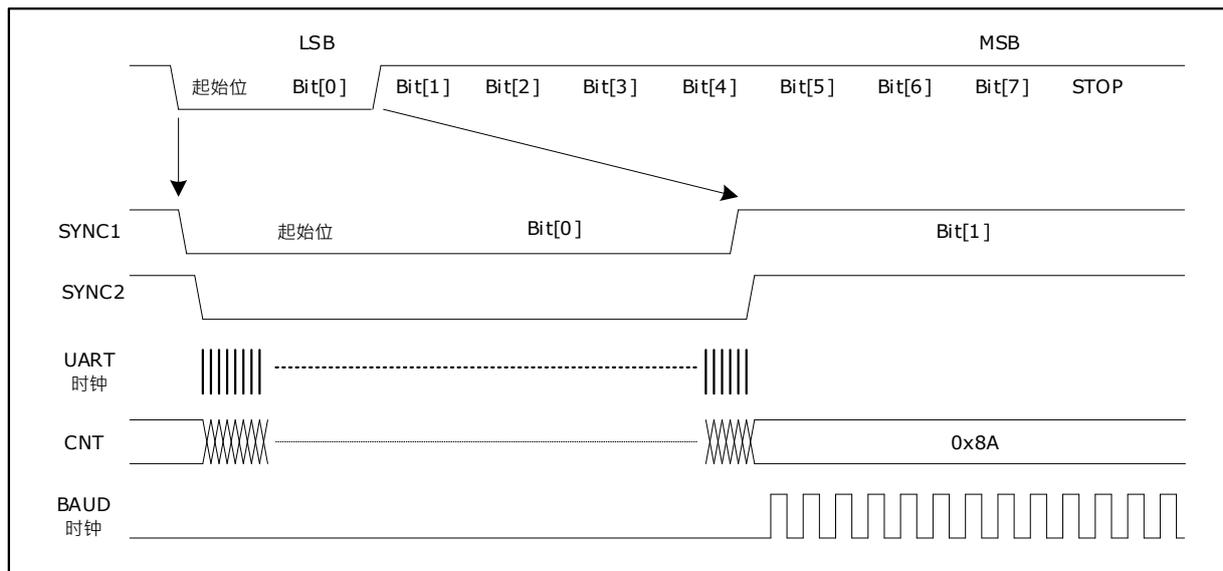


图 10-10 自动波特率侦测模式 2

10.4.8 自动流量控制

如果使能自动流控制，接收 FIFO 和发送 FIFO 会通过 **UARTx_RTSn** 和 **UARTx_CTSn** 引脚去控制 UART 的接收(RX)和发送(TX)。

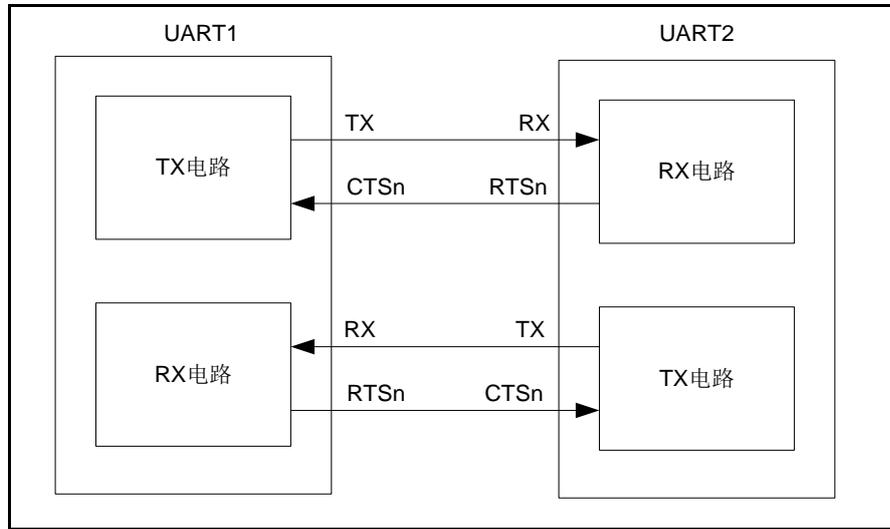


图 10-11 自动流量控制框图

10.4.8.1 RTSn 控制

当自动 RTS 被使能时，接收 FIFO 达到由 **UART_FCON** 寄存器设置的阈值，**UART_RTSn** 输出会拉为高电平。当 **UART_RTSn** 连接到另一个 UART 设备的 **UART_CTSn** 输入，另一个 UART 会停止发送串行数据，直到接收 FIFO 完全是空的。

可选择接收 FIFO 阈值的值是：1, 4, 8, 14 个字符。一个额外的字符有可能会在 **UART_RTSn** 成为无效后传送到 UART（由于 UART 进入发送器的数据尚未发送完成），阈值设置为 14 个字符能用一个字符的安全区最大限度的使用 FIFO。

硬件通过读取接收缓冲寄存器 **UART_RXBUF**，一旦得知接收 FIFO 完全为空时，**UART_RTSn** 变低电平，通知其他 UART 继续发送数据。

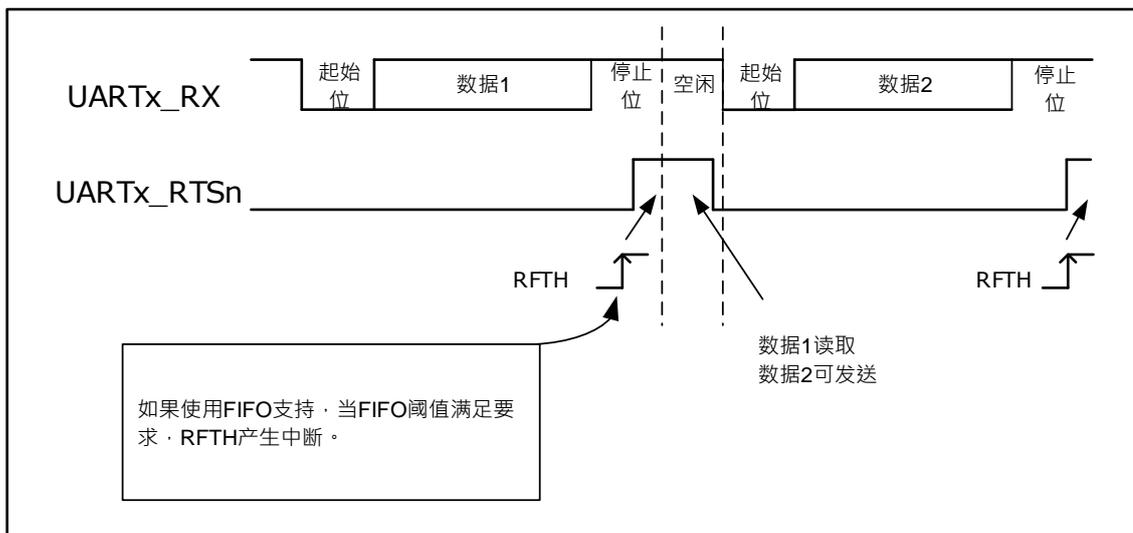


图 10-12 自动 RTSn 控制

10.4.8.2 CTSn 控制

当自动 CTS 启用，每当 UART_CTSn 输入变高时，UART 发送器被禁用。这可以防止接收端 UART 的 FIFO 溢出。在最后一个停止位发出时，假设 UART_CTSn 依然为低，则发送器会在禁用前继续传输一个字符。在发送器被禁用时，发送 FIFO 仍然可以被写入，甚至溢出。

只要 UART_CTSn 输入一变换状态，硬件就自动设置 RIF 寄存器中的 DCTS 位。它表明接收器是否准备好进行通信。如果开启中断，则会产生 DCTS 中断。

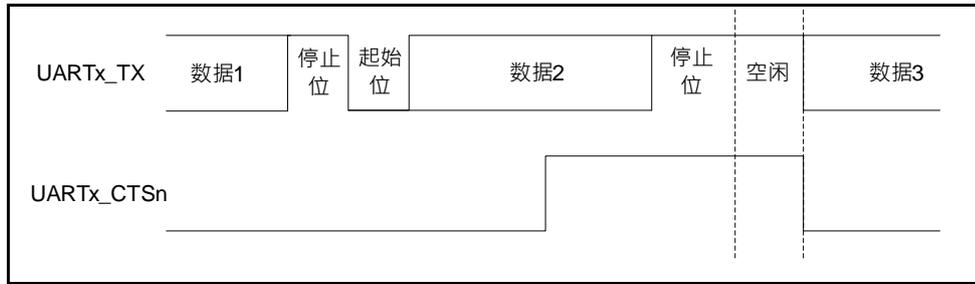


图 10-13 自动 CTSn 控制

自动流控制可以减少系统中断，当自动流控制使能，CTS_n 状态不会触发系统中断。因为是设备自动地控制其收发器。

10.4.8.3 RS485 驱动使能(DE)

当 RS485 驱动使能功能启用。它允许用户通过 DE（驱动使能）信号来激活外部收发器的控制端。滞后时间是一个发送消息的最后一个字节的停止位和释放 DE 信号之间的时间间隔，这个时间可以在 RS485 寄存器中的 DLY 位设置。DE 信号的极性则可以通过 RS485 寄存器中的 AADINV 位中设置并进行选择。

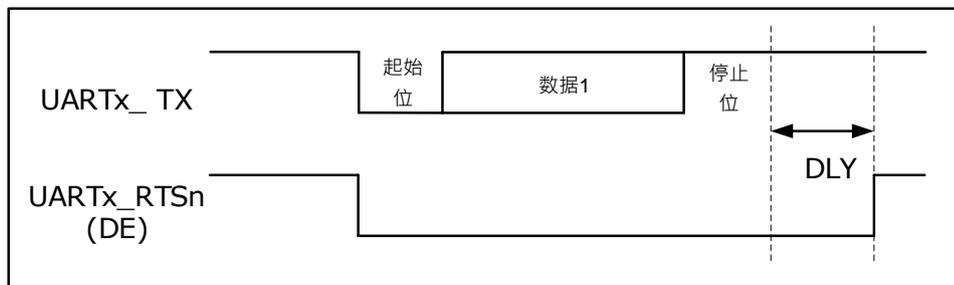


图 10-14 驱动使能当 AADINV=0

10.4.9 Modbus 通讯

UART 提供对 Modbus/RTU 和 Modbus/ASCII 协议实现的基本支持。Modbus/RTU 是一个半双工，块式传输协议。协议的控制部分（地址识别，块完整性控制和通信解释）必须由软件来完成。

UART 提供对块尾检测的基本支持，无需软件的经常性介入。

◆ Modbus/RTU

这个模式下，块尾一般公认为一个超过 2 个字符长度的静默阶段。通过一个可设置的超时长度功能来实现。

超时功能和相应的中断必须通过 RTOR 寄存器中的 RTOEN 位和 IER 寄存器中的 RXT0 位来打开。RTOR 寄存器中的 RTO 位要填入一个与超时长度相当的数字（例如 2 个字符长度为 22 个位长）。当接收线路保持空闲阶段达到这个长度时，在最后一个停止位被收到之后，会产生一个中断，表示当前的块接收已经完毕。

◆ Modbus/ASCII

在这个模式，块尾被公认为回车字符（CR/LF）串。UART 用字符匹配功能实现这个机制。

将 LF 的 ASCII 码写到 ADD[7:0]区域，IER 寄存器中的 ADDR0 位来打开，那么软件就会在收到 LF 字符后或者能够在 DMA 缓冲区中找到 CR/LF 字符时得到提示。

10.4.10 校验控制

设置 LCON 寄存器中的 PE 位, 可以使能校验控制(发送时生成一个校验位, 接收时进行校验检查)。根据 DLS 位定义的帧长度, 可能的 UART 帧格式栏在下表中。

DLS[1:0]	PE	UART 帧
00	0	起始位→8 位数据→停止位
00	1	起始位→8 位数据→校验位→停止位
01	0	起始位→7 位数据→停止位
01	1	起始位→7 位数据→校验位→停止位
10	0	起始位→6 位数据→停止位
10	1	起始位→6 位数据→校验位→停止位
11	0	起始位→5 位数据→停止位
11	1	起始位→5 位数据→校验位→停止位

表 10-4 帧格式

◆ 奇校验

校验位的内容使得一帧中的 8,7,6 或 5 个 LSB 数据以及校验位中'1'的个数为奇数。

例如: 数据=00110101, 有 4 个'1', 如果选择奇校验(在 LCON 寄存器中的 PS=0), 校验位将是'1'。

◆ 偶校验

校验位的内容使得一帧中的 8,7,6 或 5 个 LSB 数据以及校验位中'1'的个数为偶数。

例如: 数据=00110101, 有 4 个'1', 如果选择偶校验(在 LCON 寄存器中的 PS=1), 校验位将是'0'。

◆ 接收时的校验检查

如果校验检查失败, STAT 寄存器中的 PERR 标志会被置 1, 如果 IER 寄存器中的 RXBERR 为 1, 将引发相应中断。

◆ 发送时的校验生成

如果 LCON 寄存器的 PE 位被置 1, 写进数据寄存器的数据的 MSB 位被校验位替换后发送出去(如果选择奇校验奇数个'1', 如果选择偶校验偶数个'1')。

10.4.11 多处理器通讯

设置 DLS 位为 8 位字长(第 9bit 为判断地址或数据)

设置 RS485 寄存器的 AADEN 位为 1 以进入模式。

设置 RS485 寄存器的 ADDR 位配置匹配地址

可以将多个 UART 连接成一个网络来实现多机通讯。例如某个 UART 设备可以是主，它的 TX 输出和其他 UART 从设备的 RX 输入相连接；UART 从设备各自的 TXD 输出逻辑地与在一起，并且和主设备的 RX 输入相连接。

在多处理器配置中，我们通常希望只有被寻址的接收者才被激活，来接收随后的数据，这样就可以减少由未被寻址的接收器的参与带来的多余的 UART 服务开销。

未被寻址的设备可启用其静默功能进入静默模式。要使用静默模式功能，RS485 寄存器的 AADEN 位必须被置 1。

在这个模式里，如果 MSB 是 1，该字节被认为是地址，否则被认为是数据。在一个地址字节中，目标接收器的地址被放在 RS485 寄存器的 ADDR 位。

如果接收到的字节与它的编程地址不匹配时，UART 进入静默模式。当 UART 进到静默模式后，接收字节时既不会动 RIF 寄存器的 RFTH 位标志也不会产生中断或发出 DMA 请求。

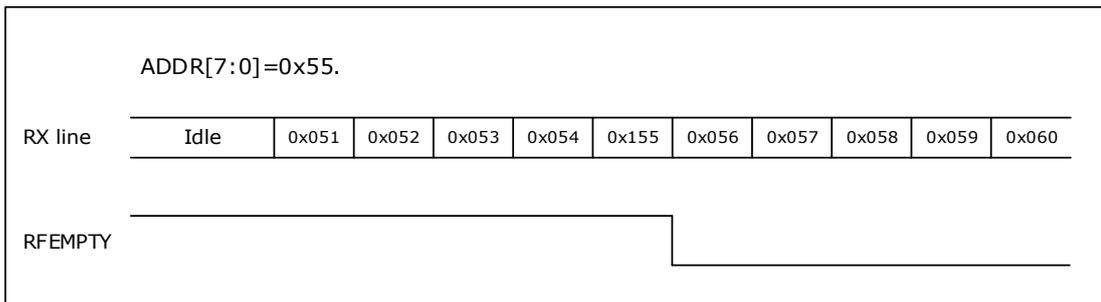


图 10-15 使用地址标示侦测模式

10.4.12 LIN 模式

◆ LIN 发送

和常规的 UART 发送相同，但包含下列区别：

设置 DLS 位为 8 位字长

设置 LIN 寄存器的 LINEN 位为 1 以进入 LIN 模式。这时，置 LIN 寄存器的 LINBKR 位为 1 将发送 13 位 0 作为断开符号。然后发两位 1，以允许对下一个开始位的检测。

◆ LIN 接收

当 LIN 模式被使能时(LIN 寄存器的 LINEN 位为 1)，断开符号检测电路被激活。该检测完全独立于 UART 接收器。不管是在总线空闲时还是在发送某数据帧期间，断开符号只要一出现就能检测到。

一旦接收器被激活(LCON 寄存器的 RXEN 位为 1)，电路就开始监测 RX 上的起始信号。监测起始位的方法同检测断开符号或数据是一样的。当起始位被检测到后，电路对每个接下来的位，在每个位的第 8, 9, 10 个过采样时钟点上进行采样，就像针对数据一样。如果 10 个(当 LIN 寄存器中的 LINBDL=0)或 11 个(当 MCON 寄存器中的 LINBDL=1)连续位都是'0'，并且又跟着一个定界符，RIF 寄存器的 LINBK 位标志就会被置 1。如果 IER 寄存器的 LINBK 位为 1，还会产生中断。在确认定界符前，要检查定界符，因为它表示 RX 线已经回到高电平。如果在第 10 或 11 个采样点之前采样到了'1'，检测电路取消当前检测并重新寻找起始位。如果 LIN 模式被禁止，接收器继续如正常 UART 那样工作，不再考虑检测断开符号。如果 LIN 模式被激活(LINEN=1)，只要一发生帧错误(例如：停止位检测到'0'，这种情况出现在断开符号被接收到的时候)，接收器就停止，直到断开符号检测电路接收到一个'1'(这种情况发生于断开符号没有完整的发出来)，或一个定界符(这种情况发生于已经检测到一个完整的断开符号)。

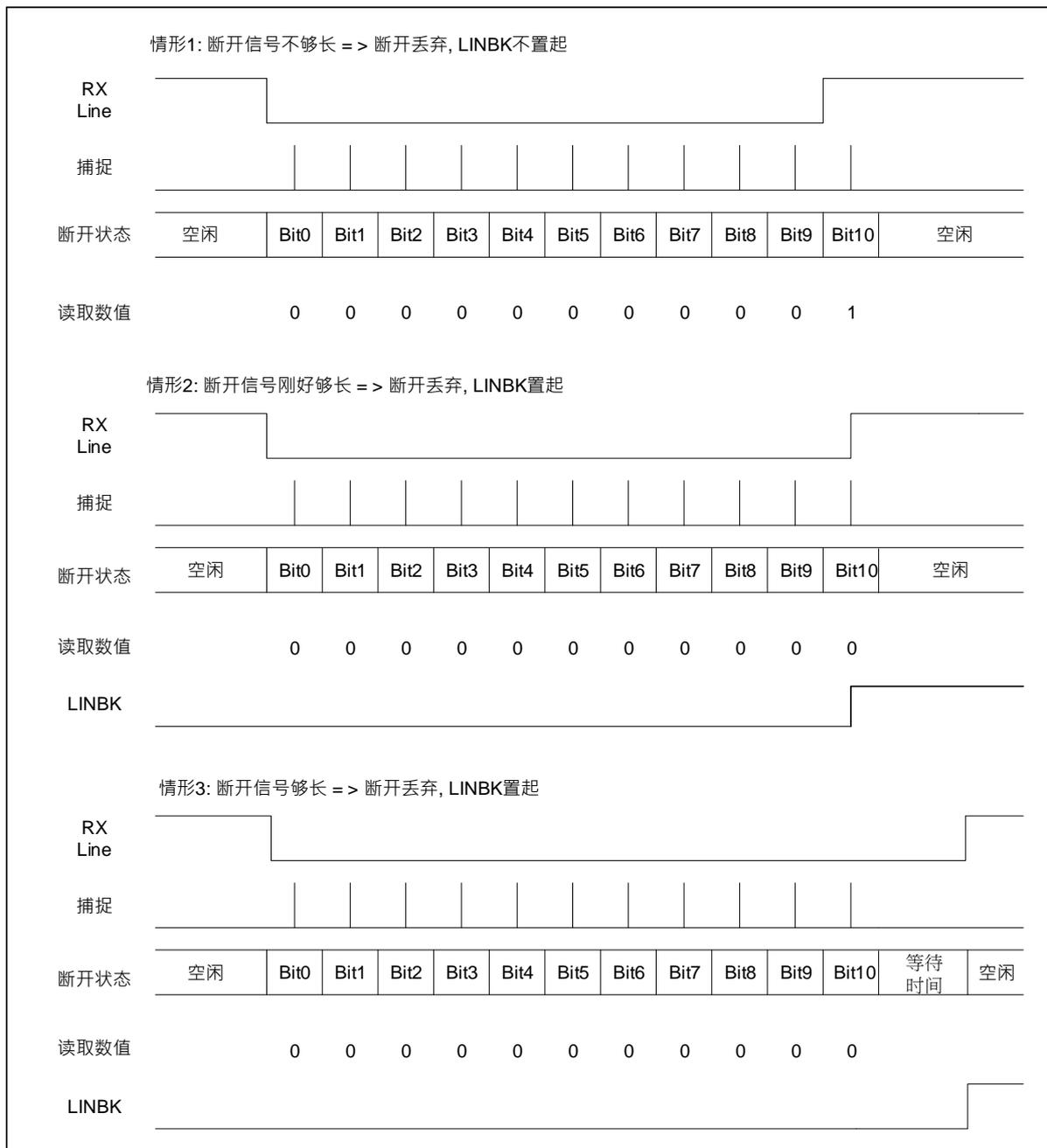


图 10-16 LIN 模式下断开信号侦测 (11 位断开长度 – LBDL 位为 1)

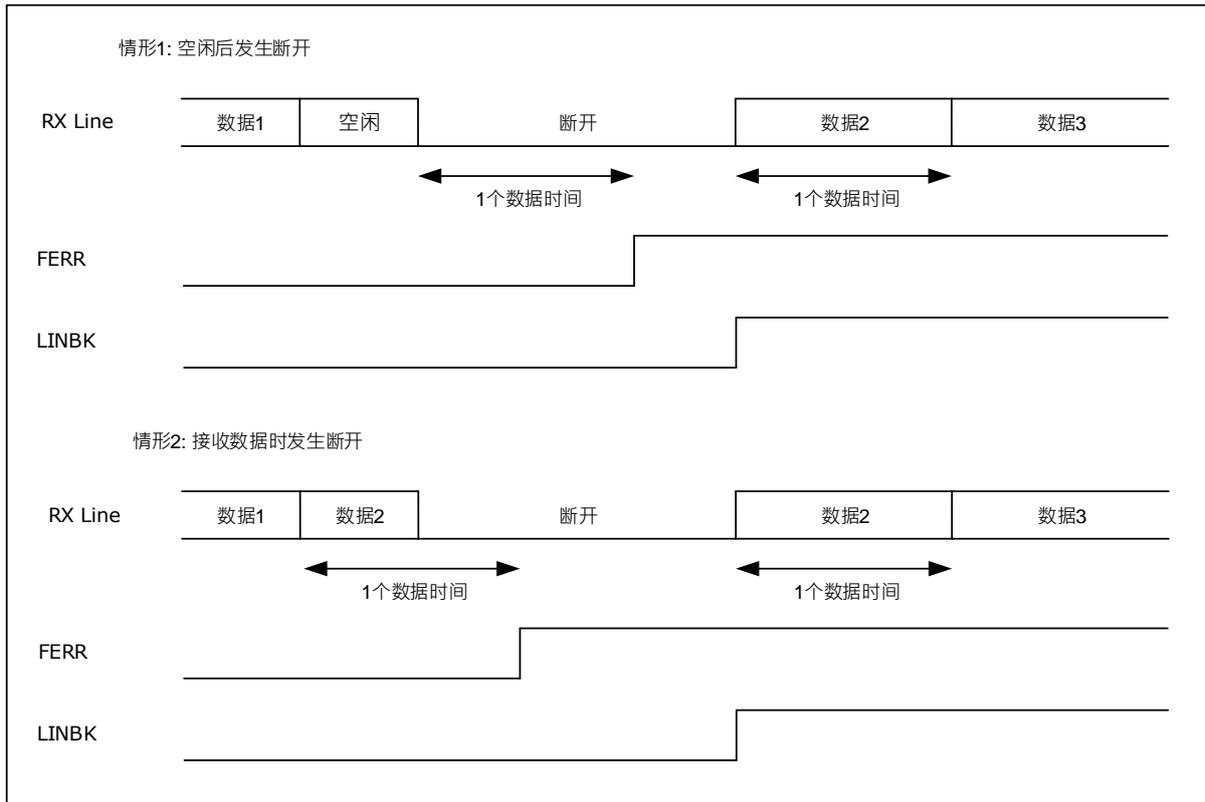


图 10-17 LIN 模式下的断开检测与帧错误的检测

10.4.13 单线半双工通讯

UART 可以配置成遵循单线半双工协议。在单线半双工模式下，TX 和 RX 引脚在芯片在内部是连在一起的。使用控制位(MCON 寄存器中的 HDEN 位)选择半双工或全双工通信。

当 HDEN 为 1 时：

TX 和 RX 引脚在芯片在内部是连在一起的

RX 不再被使用

当没有数据传输时，TX 总是被释放。因此，它在空闲状态或接收状态时表现为一个标准 I/O 口。这就意味该 I/O 在不被 UART 驱动时，必须配置成悬空输入(或开漏的输出高)。

除此以外，通信与正常 UART 模式类似。由软件来管理线上的冲突(例如通过使用一个中央仲裁器)。特别的是，发送从不会被硬件所阻碍。当 LCON 寄存器的 TXEN 为 1，只要数据一写到数据寄存器中，发送就会开始。

10.4.14 智能卡模式

设置 8 位数据位加校验位：即 LCON 寄存器中 DLS=00，PE=1

设置 0.5/1.5 个停止位：即 LCON 寄存器的 STOP=0/1

设置 SCARD 寄存器的 SCEN 为 1 以进入 Smart card 模式

在 T=0（字符）模式中，保护时间内，校验错误在字符发送完毕后被提出。

所示为在数据线上有校验错误和没有校验错误时的情形。

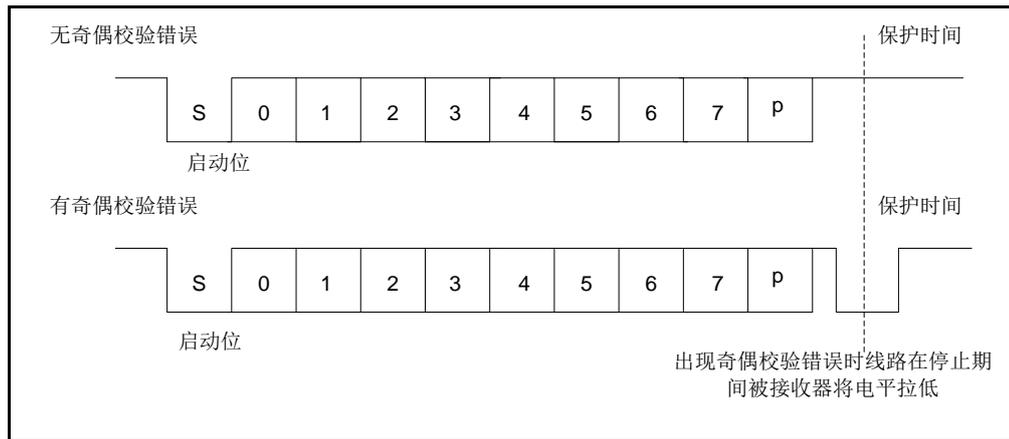


图 10-18 ISO 7816-3 异步协定

当连接到智能卡时，UART 的 TX 输出脚和智能卡通过同一根双向数据线进行通讯。所以 TX 引脚必须配置成开漏状态。

智能卡是一个单线半双工通信协议：

- ◆ 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个时钟边沿开始移位。在智能卡模式下，此发送过程还会进一步经过 1/2 波特时钟周期的延迟。
- ◆ 如果在接收一个使用 0.5 或 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 UART 的数据尚未正确接收。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发送器端（配置为 1.5 个停止位）出现帧错误。应用程序可根据协议重新发送数据。如果 NACK 控制位置 1，则接收器会发送“NACK”信号；否则不会发送 NACK 信号。
- ◆ 通过对保护时间寄存器进行编程，可以延迟 STAT 寄存器的 TBC 标志的置位。正常工作时，当发送移位寄存器为空时，会对 TBC 标志进行置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TBC 标志被强制为低电平。当保护时间计数器达到设置值时，TBC 置位为高电平。
- ◆ 对 TBC 标志的释放不受智能卡模式的影响。

- ◆ 如果在发送端检测到帧错误（由来自接收器的 NACK 信号引起），则发送端的接收器不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。
- ◆ 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

注：在智能卡模式下带有帧错误的 0x00 数据将被视为数据，而非中断。

下图详细介绍了 USART 如何对 NACK 信号采样。在本例中，USART 正在发送数据并配置了 1.5 个停止位。USART 的接收部分已被使能，以检查数据的完整性和 NACK 信号。

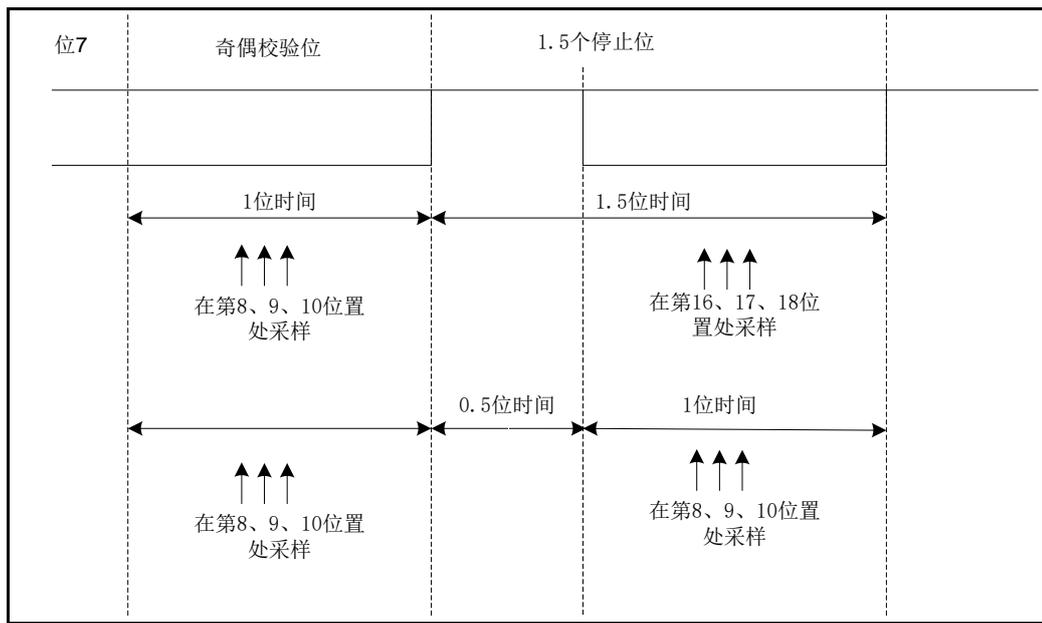


图 10-19 用 1.5 位停止位时检测校验错误

UART 可以通过 CK 脚向智能卡提供时钟。智能卡模式中，CK 和通讯没有关系，只是通过一个 5 位的预分频器从内部外设时钟源得到时钟信号。这个分频系数在 SCARD 寄存器的 PSC 位设置。CK 频率可以设置在 PCLK/2 到 PCLK/64 之间，PCLK 指外设输入时钟。

10.4.15 IrDA SIR 模块

设置 MCON 寄存器的 IREN 为 1 以进入 IrDA 模式。

IrDA SIR 物理层规定使用反相归零 (RZI) 调制方案, 它以红外光脉冲表示逻辑 0。

SIR 发送编码器用于调制 USART 发出的非归零 (NRZ) 位流。输出脉冲流会发送到外部输出驱动器和红外线 LED。UART 支持的 SIR 编码比特率最高为 115.2Kbps。在正常模式下, 所发送的脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器用于解调由红外探测器发出的归零位流, 并将接收到的 NRZ 串行位流输出到 UART。在空闲状态下, 解码器输入通常为高电平 (标记状态)。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时, 会检测到起始位。

- ◆ IrDA 是一个半双工通信协议。如果发送器忙, 例如 UART 正在向 IrDA 编码器发送数据, 则 IrDA 解码器会忽略 IrDA 接收线上的所有数据; 如果接收器忙, 例如 UART 正在接收来自 RX 引脚上的数据, 则 IrDA 不会对 UART 发送到 IrDA 的 TX 数据进行编码。在接收数据时, 应避免同时进行发送, 因为这样做可能会破坏要发送的数据。
- ◆ SIR 发送逻辑把 0 作为高脉冲发送, 把 1 作为低电平发送。脉冲的宽度规定为所选位周期的 3/16
- ◆ SIR 解码器用于将兼容 IrDA 的接收信号转换为 UART 的位流。
- ◆ SIR 接收逻辑把高电平状态解释为 1, 把低脉冲解释为 0。
- ◆ 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。
- ◆ 在 IrDA 模式里, LCON 寄存器中的 STOP 位必须配置成 1 个停止位。

接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10ms 的延迟 (IrDA 是一个半双工协议)。

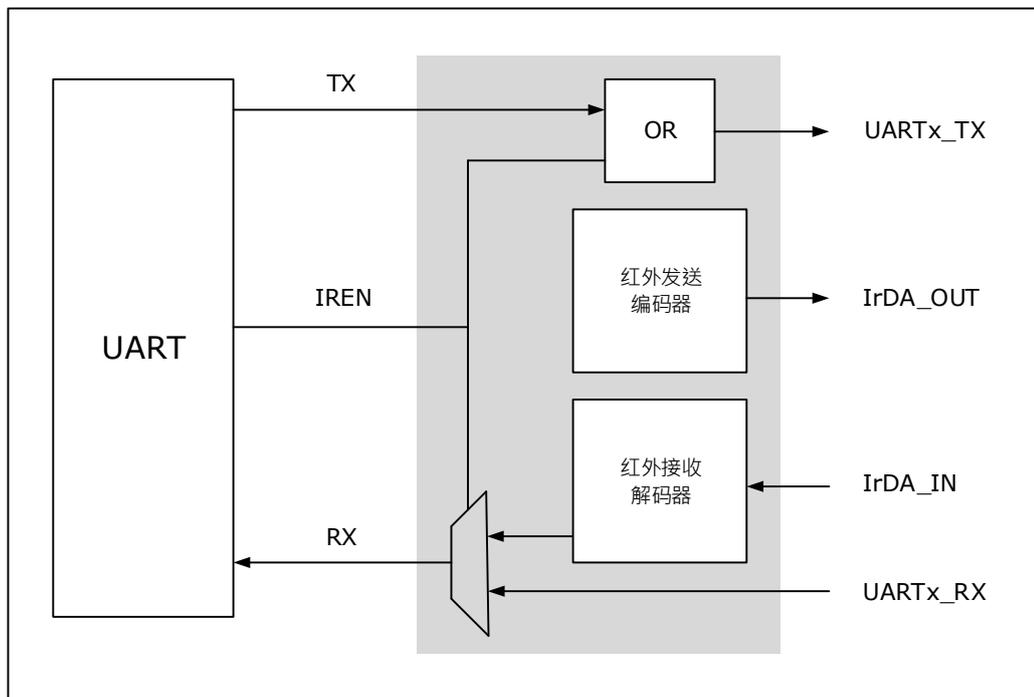


图 10-20 红外收发框图

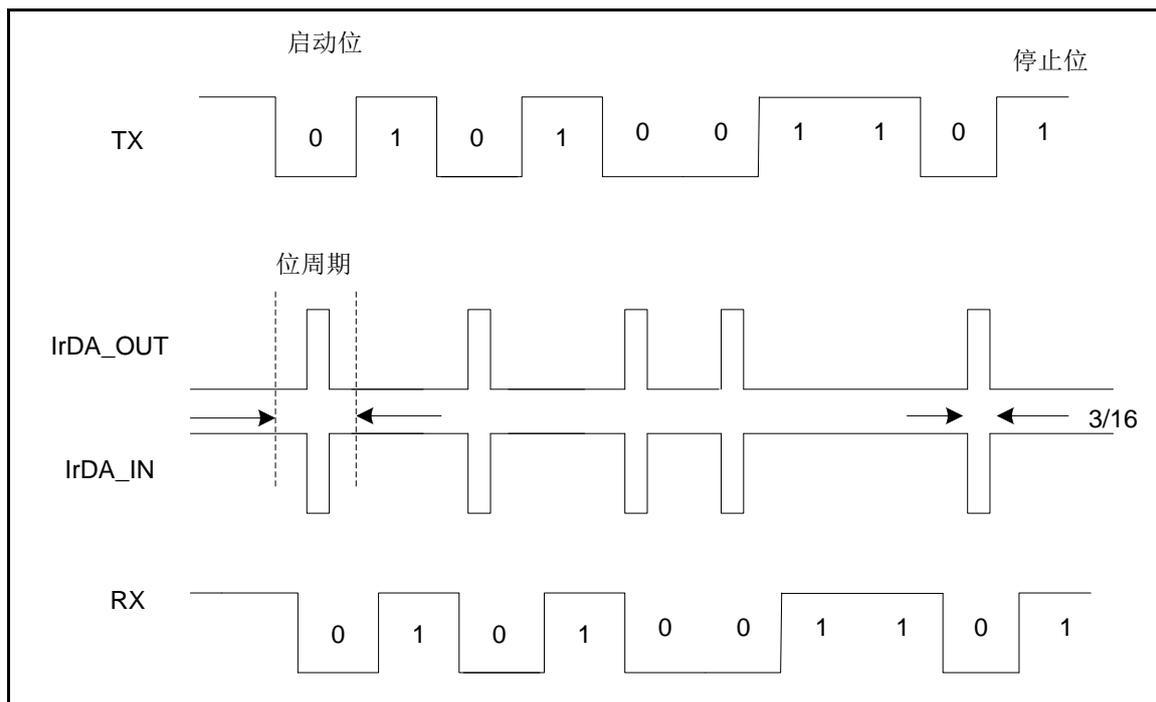


图 10-21 IrDA 数据调制 (3/16) - 正常模式

10.4.16 使用 DMA 连续通讯

设置 MCON 的 RXDMAEN 为 1 使能接收器 DMA 或 TXDMAEN 为 1 使能发送器 DMA。

UART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立的。

使用 DMA 进行发送

使用 DMA 进行发送，可以通过设置 MCON 寄存器中的 TXDMAEN 位使能。在数据被预先放到 DMA 外设所设定的 SRAM 区域设置一个 DMA 通道给 UART 发送，要使用下列步骤(x 指通道号)：

- ◆ 在 DMA 控制寄存器上将 TXBUF 寄存器地址配置成 DMA 传输的目的地址。在每个发送事件后，数据将被传送到这个地址。
- ◆ 在 DMA 控制寄存器上将内存地址配置成 DMA 传输的源地址。在每个 TTFH 事件后，将从此存储器区读出数据并传送到 TXBUF 寄存器中。
- ◆ 在 DMA 控制寄存器中配置要传输的总的字节数。
- ◆ 在 DMA 寄存器上配置信道优先级。
- ◆ 根据应用程序的要求，配置在传输完成一半还是全部完成时产生 DMA 中断。
- ◆ 将 ICR 寄存器的 TTFH 位置 1 以清除 RIF 寄存器的 TTFH 标志。
- ◆ 在 DMA 寄存器上激活该通道。

注：若使用 FIFO，可一次写入多个传输字节数，并设定 FCON 寄存器中的 TXTL 位，可设定 FIFO 深度为 0,2,4,8 个字符时，将新的数据将被传送到 TXBUF 寄存器中。

在发送模式下，当 DMA 传输完所有要发送的数据时，DMA 控制器设置 IFM 寄存器的 TFEMPTY 标志；监视 RIF 寄存器的 TFEMPTY 标志可以确认 UART 通信是否结束。这样可以在关闭 UART 或进入停机模式之前避免破坏最后一次传输的数据。软件必须等待 TBC 被置 1。TBC 标志在全部数据发送期间会是零，并且在最后一帧数据发送出去之后会由硬件置 1。

利用 DMA 接收

使用 DMA 进行接收，可以通过设置 MCON 寄存器中的 RXDMAEN 使能。当收到一个字节数据时，从 RXBUF 寄存器取出来的数据会被转移到 DMA 外设中指向的 SRAM 区域将一个 DMA 通道设置给 UART 接收，要按照下列步骤：

- ◆ 在 DMA 控制寄存器上将 RXBUF 寄存器地址配置成 DMA 传输的源地址。在每个 RFTH 事件后，数据将从这个地址取走。
- ◆ 在 DMA 控制寄存器上将内存地址配置成 DMA 传输的目标地址。在每个 RFTH 事件后，数据将从 RXBUF 寄存器取往这个目标地址。
- ◆ 在 DMA 控制寄存器中配置要传输的总的字节数。
- ◆ 在 DMA 寄存器上配置信道优先级。
- ◆ 根据应用程序的要求，配置在传输完成一半还是全部完成时产生 DMA 中断。
- ◆ 在 DMA 寄存器上激活该通道。

注：若使用 FIFO，可一次接收多个传输字节数，并设定 FCON 寄存器中的 RXTL，可设定 FIFO 深度为 1,4,8,14 个字符时，将新的数据将被传送到 RXBUF 寄存器中。当传输完成时，若使能了 DMA 中断，DMA 控制器会产生中断。

多缓冲器通信中的错误标志和中断产生

多缓冲器通信的情况中，通信期间如果发生任何错误，会在当前字节传输后将错误标志置 1。如果中断使能位被置 1，将产生中断。在单个字节接收的情况下，和 IFM 寄存器中的 RXBERR 和 RFOERR 一起被置起的帧错误和溢出错误，有单独的错误标志中断使能位；如果被置 1 了，会在当前字节传输结束后，产生中断。

10.4.17 中断配置

- ◆ **UART_IER Interrupt Enable Register**
中断使能寄存器，此位设定 1 时，表示开启中断功能，并且同时反映在 IVS 寄存器。此寄存器属于 Write Only，并且只允许写入 1，无法写 0 取消使能中断设定。
- ◆ **UART_IDR Interrupt Disable Register**
中断禁止寄存器，此位设定 1 时，表示关闭中断功能，并且同时反映在 IVS 寄存器。此寄存器属于 Write Only，并且只允许写入 1，无法写 0 取消禁止中断设定。
- ◆ **UART_IVS Interrupt Valid Status Register**
中断有效位状态寄存器，反映 IER 与 IDR 寄存器所设定的结果。0:中断禁止 1:中断使能
- ◆ **UART_RIF Raw Interrupt Flag Register**
无屏蔽中断标志寄存器，反映所有发生中断事件的状态，无论 IVS 是否有使能中断，皆会反映在此寄存器中，主要提供用户监控无屏蔽的中断位，使否有错误事件发生。
- ◆ **UART_IFM Interrupt Flag Masked Status**
屏蔽中断标志寄存器，记录中断使能位所发生中断事件。0:无中断事件 1:发生中断事件
- ◆ **UART_ICR Interrupt Clear Register**
中断清除寄存器，此位设定 1 时，清除中断标志 RIF 与 IFM，此寄存器属于 Write One Clear，并且只允许写入 1 清除，无法写 0 清除。

在 UART 中，配置了 17 种中断，分别为下表。

中断事件	中断标志
接收器字节格式错误	RXBERR
自动波特率侦测结束	ABEND
自动波特率侦测超时	ABTO
CTS _n 引脚电平改变	DCTS
接收超时	RXTO
地址匹配	ADDRM
LIN 断开侦测	LINBK
块结束	EOB
噪声位侦测	NOISE
接收器 FIFO 触发门槛	RFTH
接收器 FIFO 满	RFFULL
接收器 FIFO 溢出	RFOERR
接收器 FIFO 下溢	RFUERR
发送器字节完成	TBC
发送器 FIFO 触发门槛	TFTH
发送器 FIFO 空	TFEMPTY
发送器 FIFO 溢出	TFOVER

表 10-5 中断配置表

10.5 特殊功能寄存器

10.5.1 寄存器列表

UART 寄存器列表			
名称	偏移地址	类型	描述
UART_RXBUF	00 _H	R	接收缓冲寄存器
UART_TXBUF	04 _H	R/W	发送缓冲寄存器
UART_BRR	08 _H	R/W	波特率寄存器
UART_LCON	0C _H	R/W	格式控制寄存器
UART_MCON	10 _H	R/W	模式控制寄存器
UART_RS485	14 _H	R/W	RS485 控制寄存器
UART_SCARD	18 _H	R/W	智能卡控制寄存器
UART_LIN	1C _H	R/W	LIN 控制寄存器
UART_RTOR	20 _H	R/W	接收超时寄存器
UART_FCON	24 _H	R/W	FIFO 控制寄存器
UART_STAT	28 _H	R	状态寄存器
UART_IER	2C _H	W1	中断使能寄存器
UART_IDR	30 _H	W1	中断禁止寄存器
UART_IVS	34 _H	R	中断有效位状态寄存器
UART_RIF	38 _H	R	无屏蔽中断标志寄存器
UART_IFM	3C _H	R	屏蔽中断标志寄存器
UART_ICR	40 _H	C_W1	中断清除寄存器

10.5.2 寄存器描述

10.5.2.1 接收缓冲寄存器(UART_RXBUF)

接收缓冲寄存器 (UART_RXBUF)																															
偏移地址: 00H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								RXBUF<8:0>							

—	Bits 31-9	—	—
RXBUF	Bits 8-0	R	接收缓冲寄存器 包含接收到的字节。 RXBUF寄存器提供接收移位寄存器和内部总线间的并行接口。 注: 位8用于RS485寻址模式

10.5.2.2 发送缓冲寄存器 (UART_TXBUF)

发送缓冲寄存器 (UART_TXBUF)																															
偏移地址: 04H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								TXBUF<8:0>							

—	Bits 31-9	—	—
TXBUF	Bits 8-0	R/W	发送缓冲寄存器 用于写入要发送的数据字节。 TXBUF寄存器提供发送移位寄存器与内部总线间的并行接口。 注: 位8用于RS485地址模式

10.5.2.3 波特率寄存器 (UART_BRR)

波特率寄存器 (UART_BRR)																																													
偏移地址: 08H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																BRR<15:0>																													

—	Bits 31-16	—	—
BRR	Bits 15-0	R/W	波特率寄存器 整数部分 BRR[15:4] = DIVISOR[11:0] 小数部分 BRR[3:0] = FRACTION[3:0] 此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。 注:使用自动波特率功能时则可自动写入

10.5.2.4 格式控制寄存器 (UART_LCON)

格式控制寄存器 (UART_LCON)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TXEN	RXEN	DBCEN			BREAK	SWAP	TXINV	RXINV	DATAINV	MSB	PS	PE	STOP		DLS<1:0>

—	Bits 31-16	—	—
TXEN	Bit 15	R/W	发送器使能 使能发送器，此位由软件设置1和清除。 0: 发送器禁止 1: 发送器使能
RXEN	Bit 14	R/W	接收器使能 使能接收器，此位由软件设置1和清除。 0: 接收器禁止 1: 接收器使能
DBCEN	Bit 13	R/W	去抖动使能 使能去抖动功能，此位由软件设置1和清除。 0: 去抖动禁止 1: 去抖动使能，在RX线上须维持8个时钟的电平
—	Bits 12-11	—	—
BREAK	Bit 10	R/W	断开使能 使能断开功能，此位由软件设置1和清除。 此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。 0: 断开禁止 1: 断开使能，会使得TX输出为0
SWAP	Bit 9	R/W	交换TX/RX pins 使能交换功能，此位由软件设置1和清除。 此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。 0: 交换禁止，TX和RX引脚按照标准引脚配置使用 1: 交换使能，TX和RX引脚功能交换使用，此功能用与和其他UART接口进行交叉互联时
TXINV	Bit 8	R/W	TX引脚电平反向 TX引脚反向功能，此位由软件设置1和清除。 此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。

			<p>0: TX 引脚信号工作于标准逻辑电平 (VDD5=1/idle, Gnd=0/mark)</p> <p>1: TX 引脚信号反向 (VDD5=0/mark, Gnd=1/idle)。此功能可用于TX线上带有外部反向器时</p>
RXINV	Bit 7	R/W	<p>RX引脚电平反向</p> <p>RX引脚反向功能，此位由软件设置1和清除。此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。</p> <p>0: RX 引脚信号工作于标准逻辑电平 (VDD5=1/idle, Gnd=0/mark)</p> <p>1: RX 引脚信号反向 (VDD5=0/mark, Gnd=1/idle)。此功能可用于RX线上带有外部反向器时</p>
DATAINV	Bit 6	R/W	<p>二进制反向使能</p> <p>二进制反向功能，此位由软件设置1和清除。此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。</p> <p>0: 缓冲寄存器中的逻辑数据在接收的时候，采用正/直接逻辑。(1=H, 0=L)</p> <p>1: 缓冲寄存器中的逻辑数据在接收的时候，采用负/反向逻辑。(1=L, 0=H)</p>
MSB	Bit 5	R/W	<p>高位在前使能</p> <p>高位在前功能，此位由软件设置1和清除。此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。</p> <p>0: 数据在发送和接收的时候，采用起始位后面跟着第0位的顺序</p> <p>1: 数据在发送和接收的时候，采用起始位后面跟着的最高位的顺序</p>
PS	Bit 4	R/W	<p>校验位奇偶选择</p> <p>当使能校验功能时，选择校验位为奇校验或偶校验，此位由软件设置1和清除。此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。</p> <p>0: 奇校验</p> <p>1: 偶校验</p>
PE	Bit 3	R/W	<p>校验使能</p> <p>使能校验功能，计算好的校验位被插入到最高位，并检测接收数据的校验位(接收与发送功能)，此位由软件设置1和清除。此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。</p> <p>0: 校验位禁止</p>

			1: 校验位使能
STOP	Bit 2	R/W	<p>停止位选择 此位由软件设置1和清除。 此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。 普通模式下 0: 1个停止位 1: 2个停止位(在5字长模式为1.5个停止位) 智能卡模式 0: 0.5个停止位 1: 1.5个停止位</p>
DLS	Bits 1-0	R/W	<p>数据字长选择 此位由软件设置1和清除。 此位在LCON寄存器中的RXEN与TXEN位为0时才可以写入。 00: 8字长 01: 7字长 10: 6字长 11: 5字长</p>

10.5.2.5 模式控制寄存器 (UART_MCON)

模式控制寄存器 (UART_MCON)																																
偏移地址: 10H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																TXDMAEN	RXDMAEN			ABRREPT	ABRMOD<1:0>			ABREN			BKREQ	H DEN	I REN	A FCEN	R TSSET	L PBKEN

—	Bits 31-16	—	—
TXDMAEN	Bit 15	R/W	发送器DMA使能 此位由软件设置1和清除。 0: 发送器DMA通讯禁止 1: 发送器DMA通讯使能
RXDMAEN	Bit 14	R/W	接收器DMA使能 此位由软件设置1和清除。 0: 接收器DMA通讯禁止 1: 接收器DMA通讯使能
—	Bits 13-12	—	—
ABRREPT	Bit 11	R/W	重复自动波特率侦测 在使能自动波特率侦测时, 当产生波特率超时时并不会清除自动波特率开关, 并在下一个下降沿时重复自动波特率侦测, 此位由软件设置1和清除。 0: 重复自动波特率侦测禁止 1: 重复自动波特率侦测使能
ABRMOD	Bits 10-9	R/W	自动波特率模式选择 此位由软件设置1和清除。 00: 模式0, 侦测第一个下降沿到第二个下降沿时间(侦测2位) 01: 模式1, 侦测第一个下降沿到第一个上升沿时间(侦测1位) 10: 模式2, 侦测第一个下降沿到第一个上升沿时间(侦测2位) 11: 保留
ABREN	Bit 8	R/W	自动波特率使能 此位在使能后并完成自动波特率侦测后会自动清除, 也可由软件设置1和清除。 0: 自动波特率禁止 1: 自动波特率使能
—	Bits 7-6	—	—

BKREQ	Bit 5	W	<p>断开请求 此位在写入后的下一个时钟会自动清除。 0: 断开请求禁止 1: 断开请求使能, 根据设定的N位长(8/7/6/5)产生N个低脉冲信号</p>
HDEN	Bit 4	R/W	<p>单线半双工使能(Only UART) 此位由软件设置1和清除。 0: 单线半双工禁止 1: 单线半双工使能</p>
IREN	Bit 3	R/W	<p>IrDA红外线模式使能 (Only UART) 此位由软件设置1和清除。 0: IrDA红外线模式禁止 1: IrDA红外线模式使能</p>
AFCEN	Bit 2	R/W	<p>自动流量控制使能 此位由软件设置1和清除。 0: 自动流量控制禁止 1: 自动流量控制使能</p>
RTSSET	Bit 1	R/W	<p>RTSn设置控制 此位由软件设置1和清除。 0: 自动流量控制禁止时, RTSn引脚输出高电平 1: 自动流量控制禁止时, RTSn引脚输出低电平</p>
LPBKEN	Bit 0	R/W	<p>回送模式使能 此模式是用于UART测试项目的诊断模式, 在UART普通模式下运行, TX引脚输出为高电平, 串行的数据在内部回送至RX。在此模式下, 所有中断都是正常运行的, 此位由软件设置1和清除。 0: 回送模式禁止 1: 回送模式使能</p>

10.5.2.6 RS485 控制寄存器 (UART_RS485)

RS485 控制寄存器 (UART_RS485)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DLY<7:0>								ADDR<7:0>								AADINV AADACEN AADNEN AADEN							

—	Bits 31-24	—	—
DLY	Bits 23-16	R/W	延迟数值 用于设置延迟RTSn的输出时间，是由一个8位计数器计数，此位由软件设置和清除
ADDR	Bits 15-8	R/W	地址匹配数值 用于多机通讯时地址标记的检测。 接收器在RS485自动检测模式时，当接收数据的最高位为1且匹配ADDR，数据才允许接收至FIFO中，否则舍弃此数值，此位由软件设置和清除。
—	Bits 7-4	—	—
AADINV	Bit 3	R/W	驱动使能反向 在自动流量控制模式时，设置驱动使能引脚(RTSn/DE)的输出电平，此位由软件设置和清除。 0: 当发送器开始发送数据时，驱动使能引脚输出0，发送完成且FIFO内无数据时，驱动使能引脚输出1 1: 当发送器开始发送数据时，驱动使能引脚输出1，发送完成且FIFO内无数据时，驱动使能引脚输出0
AADACEN	Bit 2	R/W	自动流量控制模式使能 此位由软件设置1和清除。 0: 自动流量控制模式禁止 1: 自动流量控制模式使能
AADNEN	Bit 1	R/W	普通模式使能 此位由软件设置1和清除。 0: 普通模式禁止 1: 普通模式使能，接收地址位为第8位(UART_RXBUF)
AADEN	Bit 0	R/W	自动地址侦测模式使能 在普通模式时，设置此位无效，此位由软件设置1和清除。

			0: 自动地址侦测模式禁止 1: 自动地址侦测模式使能, 当接收数据的地址为1且匹配ADDR时, 才会将数据接收至FIFO中
--	--	--	---

10.5.2.7 智能卡控制寄存器 (UART_SCARD)

智能卡控制寄存器 (UART_SCARD)																																
偏移地址: 18H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
BLEN<7:0>								GT<7:0>								PSC<7:0>								—		—		SCCNT<2:0>		SCLKEN	SCNACK	SCEN

BLEN	Bits 31-24	R/W	<p>块长度 (Only UART) 设置了智能卡模式T=1的接收时的块长度，此位由软件设置1和清除。 例如 BLEN = 0 -> 0个信号字符 BLEN = 1 -> 1个信号字符 BLEN = 255 -> 255个信号字符 这个功能也可以在其他模式中使用，当LCON寄存器的RXEN位清除时，块长度计数器会重新寄数</p>
GT	Bits 23-16	R/W	<p>保护时间(Only UART) 设置保护时间长度，是使用波特时钟为单位。在智能卡模式中使用，完成标志(RIF寄存器TBC位)在保护时间过后置起，此位由软件设置1和清除</p>
PSC	Bits 15-8	R/W	<p>分频器数值(Only UART) 此位由软件设置1和清除。 智能卡模式: PSC[4:0]: 输出时钟分频数值 用于设定UART时钟的分频数，得到智能卡输出时钟，由五个有效位组成，乘以2得到的数值作为分频 00000: 保留 00001: 2分频 00010: 4分频 00011: 6分频</p>
—	Bits 7-6	—	—
SCCNT	Bits 5-3	R/W	<p>智能卡重试计数器 (Only UART) 设置智能卡模式中接收和发送的重试次数。此位由软件设置1和清除。 在发送模式下，在产生帧错误前重试发送的次数</p>

			<p>在接收模式下，在接收到NACK后重试接收的次数</p> <p>0x0: 重试功能关闭，在发送与接收模式下不进行自动重试</p> <p>0x1 至 0x7: 在产生错误前自动重试的次数</p>
SCLKEN	Bit 2	R/W	<p>智能卡时钟使能 (Only UART)</p> <p>此位由软件设置1和清除。</p> <p>0: CK引脚禁止</p> <p>1: CK引脚使能</p>
SCNACK	Bit 1	R/W	<p>智能卡NACK发送使能(Only UART)</p> <p>此位由软件设置1和清除。</p> <p>0: 出现校验错误时禁止发送NACK信号</p> <p>1: 出现校验错误时使能发送NACK信号</p>
SCEN	Bit 0	R/W	<p>智能卡模式使能 (Only UART)</p> <p>此位由软件设置1和清除。</p> <p>0: 智能卡模式禁止</p> <p>1: 智能卡模式使能</p>

10.5.2.9 接收器超时寄存器 (UART_RTOR)

超时接收寄存器 (UART_RTOR)																																
偏移地址: 20 _H																																
复位值: 00000000_00000000_00000000_11111111 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
							RTOEN																									RTO<23:0

—	Bits 31-25	—	—
RTOEN	Bit 24	R/W	接收器超时使能 此位由软件设置1和清除。 0: 接收器超时禁止 1: 接收器超时使能
RTO	Bits 23-0	R/W	接收器超时数值 2 (Only UART) RTO[7:0]: 接收器超时数值 1 设置接收超时时间，使用波特率时钟的字长为单 位。 在标准模式下，接收最后一个字节后，在超时时 间内未检测到新的起始位，将置起 RIF 寄存器的 RXTO 位，此位由软件设置和清除。 在智能卡模式下，这个数值是用来实现 CWT 和 BWT。

			1: 接收器FIFO重置使能
--	--	--	----------------

			<p>FIFO时, 将会置起此位。此位由硬件设置1, 在接收数据时清除或设置ICR寄存器的RFUERR位清除</p> <p>0: 接收器FIFO下溢错误未产生 1: 接收器FIFO下溢错误产生</p>
RFOERR	Bit 12	R	<p>接收器FIFO溢位错误 当接收器FIFO内已有16个数据时, 有新数据再次接收至FIFO中时, 将会置起此位并舍弃新数据。此位由硬件设置1, 在读取数据时清除或设置ICR寄存器的RFOERR位清除</p> <p>0: 接收器FIFO溢出错误未产生 1: 接收器FIFO溢出错误产生</p>
RFFULL	Bit 11	R	<p>接收器FIFO满 当接收器FIFO内有16个数据时, 此位由硬件设置1, 在FIFO内未满足16个数据时清除。</p> <p>0: 接收器FIFO未满足16个数据 1: 接收器FIFO满足16个数据</p>
RFEMPTY	Bit 10	R	<p>接收器FIFO空 当接收器FIFO内无任何数据时, 此位由硬件设置1, 接收数据至FIFO内时清除。</p> <p>0: 接收器FIFO有数据 1: 接收器FIFO无数据</p>
RFTH	Bit 9	R	<p>接收器FIFO触发阈值 当FCR寄存器的RXFL位到达FCR寄存器的RXTL设定的阈值, 此位由硬件设置1, 在未达到阈值清除</p> <p>0: 接收器FIFO未达到阈值水平 1: 接收器FIFO到达阈值水平</p>
RSBUSY	Bit 8	R	<p>接收移位寄存器忙碌 当接收数据时, 由硬件设置1在完成接收数据后清除</p> <p>0: 接收器未接收数据 1: 接收器正在接收数据</p>
—	Bits 7-4	—	—
CTSSTA	Bit 3	R	<p>CTS_n状态 此位显示CTS_n输入引脚状态, 由硬件设置1和清除。</p> <p>0: CTS_n输入引脚为0 1: CTS_n输入引脚为1</p>
BKERR	Bit 2	R	<p>断开错误 当接收数据与停止位皆为0时, 将由硬件设置1。此位为显示当前读取接收器FIFO的数值。</p> <p>0: 断开错误未产生 1: 断开错误产生</p>

FERR	Bit 1	R	<p>帧错误 当接收数据的停止位为0时，将由硬件设置1。此位为显示当前读取接收器FIFO的数值。 0: 帧错误未产生 1: 帧错误产生</p>
PERR	Bit 0	R	<p>校验错误 当接收数据的校验位接收错误时，将由硬件设置1。此位为显示当前读取接收器FIFO的数值。 0: 校验错误未产生 1: 校验错误产生</p>

			1: 块结束中断使能
LINBK	Bit 6	W1	LIN断开侦测中断使能 (Only UART) 0: 写入0无效 1: LIN断开侦测中断使能
ADDRM	Bit 5	W1	地址匹配中断使能 0: 写入0无效 1: 地址匹配中断使能
RXTO	Bit 4	W1	接收超时中断使能 0: 写入0无效 1: 接收超时中断使能
DCTS	Bit 3	W1	CTS_n引脚电平改变中断使能 0: 写入0无效 1: CTS _n 引脚电平改变中断使能
ABTO	Bit 2	W1	自动波特率侦测超时中断使能 0: 写入0无效 1: 自动波特率侦测超时中断使能
ABEND	Bit 1	W1	自动波特率侦测结束中断使能 0: 写入0无效 1: 自动波特率侦测结束中断使能
RXBERR	Bit 0	W1	接收器字节格式错误中断使能 0: 写入0无效 1: 接收器字节格式错误中断使能

10.5.2.13 中断禁止寄存器(UART_IDR)

中断禁止寄存器(UART_IDR)																																
偏移地址: 30 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
													TFOVER		TFEMPTY	TFTH	TBC	RFUERR	RFOERR	RFFULL			RFTH	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bits 31-19	—	—
TFOVER	Bit 18	W1	发送器FIFO溢出中断禁止 0: 写入0无效 1: 发送器FIFO溢出中断禁止
—	Bit 17	—	—
TFEMPTY	Bit 16	W1	发送器FIFO空中断禁止 0: 写入0无效 1: 发送器FIFO空中断禁止
TFTH	Bit 15	W1	发送器FIFO触发阈值中断禁止 0: 写入0无效 1: 发送器FIFO触发阈值中断禁止
TBC	Bit 14	W1	发送器字节完成中断禁止 0: 写入0无效 1: 发送器字节完成中断禁止
RFUERR	Bit 13	W1	接收器FIFO下溢中断禁止 0: 写入0无效 1: 接收器FIFO下溢中断禁止
RFOERR	Bit 12	W1	接收器FIFO溢出中断禁止 0: 写入0无效 1: 接收器FIFO溢出中断禁止
RFFULL	Bit 11	W1	接收器FIFO满中断禁止 0: 写入0无效 1: 接收器FIFO满中断禁止
—	Bit 10	—	—
RFTH	Bit 9	W1	接收器FIFO触发阈值中断禁止 0: 写入0无效 1: 接收器FIFO触发阈值中断禁止
NOISE	Bit 8	W1	噪声位侦测中断禁止 0: 写入0无效 1: 噪声位侦测中断禁止
EOB	Bit 7	W1	块结束中断禁止(Only UART) 0: 写入0无效

			1: 块结束中断禁止
LINBK	Bit 6	W1	LIN断开侦测中断禁止 (Only UART) 0: 写入0无效 1: LIN断开侦测中断禁止
ADDRM	Bit 5	W1	地址匹配中断禁止 0: 写入0无效 1: 地址匹配中断禁止
RXTO	Bit 4	W1	接收超时中断禁止 0: 写入0无效 1: 接收超时中断禁止
DCTS	Bit 3	W1	CTS_n引脚电平改变中断禁止 0: 写入0无效 1: CTS _n 引脚电平改变中断禁止
ABTO	Bit 2	W1	自动波特率侦测超时中断禁止 0: 写入0无效 1: 自动波特率侦测超时中断禁止
ABEND	Bit 1	W1	自动波特率侦测结束中断禁止 0: 写入0无效 1: 自动波特率侦测结束中断禁止
RXBERR	Bit 0	W1	接收器字节格式错误中断禁止 0: 写入0无效 1: 接收器字节格式错误中断禁止

10.5.2.14 中断有效位状态寄存器(UART_IVS)

中断有效位状态寄存器 (UART_IVS)																																
偏移地址: 34 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
													TFOVER		TFEMPTY	TFTH	TBC	RFUERR	RFOERR	RFFULL			RFTH	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bits 31-19	—	—
TFOVER	Bit 18	R	发送器FIFO溢出中断有效位 0: 发送器FIFO溢出中断禁止 1: 发送器FIFO溢出中断使能
—	Bit 17	—	—
TFEMPTY	Bit 16	R	发送器FIFO空中断有效位 0: 发送器FIFO空中断禁止 1: 发送器FIFO空中断使能
TFTH	Bit 15	R	发送器FIFO触发阈值中断有效位 0: 发送器FIFO触发阈值中断禁止 1: 发送器FIFO触发阈值中断使能
TBC	Bit 14	R	发送器字节完成中断有效位 0: 发送器字节完成中断禁止 1: 发送器字节完成中断使能
RFUERR	Bit 13	R	接收器FIFO下溢中断有效位 0: 接收器FIFO下溢中断禁止 1: 接收器FIFO下溢中断使能
RFOERR	Bit 12	R	接收器FIFO溢出中断有效位 0: 接收器FIFO溢出中断禁止 1: 接收器FIFO溢出中断使能
RFFULL	Bit 11	R	接收器FIFO满中断有效位 0: 接收器FIFO满中断禁止 1: 接收器FIFO满中断使能
—	Bit 10	—	—
RFTH	Bit 9	R	接收器FIFO触发阈值中断有效位 0: 接收器FIFO触发阈值中断禁止 1: 接收器FIFO触发阈值中断使能
NOISE	Bit 8	R	噪声位侦测中断有效位 0: 噪声位侦测中断禁止 1: 噪声位侦测中断使能
EOB	Bit 7	R	块结束中断有效位 (Only UART) 0: 块结束中断禁止

			1: 块结束中断使能
LINBK	Bit 6	R	LIN断开侦测中断有效位 (Only UART) 0: LIN断开侦测中断禁止 1: LIN断开侦测中断使能
ADDRM	Bit 5	R	地址匹配中断有效位 0: 地址匹配中断禁止 1: 地址匹配中断使能
RXTO	Bit 4	R	接收超时中断有效位 0: 接收超时中断禁止 1: 接收超时中断使能
DCTS	Bit 3	R	CTS_n引脚电平改变中断有效位 0: CTS _n 引脚电平改变中断禁止 1: CTS _n 引脚电平改变中断使能
ABTO	Bit 2	R	自动波特率侦测超时中断有效位 0: 自动波特率侦测超时中断禁止 1: 自动波特率侦测超时中断使能
ABEND	Bit 1	R	自动波特率侦测结束中断有效位 0: 自动波特率侦测结束中断禁止 1: 自动波特率侦测结束中断使能
RXBERR	Bit 0	R	接收器字节格式错误中断有效位 0: 接收器字节格式错误中断禁止 1: 接收器字节格式错误中断使能

10.5.2.15 无屏蔽中断标志寄存器 (UART_RIF)

无屏蔽中断标志寄存器(UART_RIF)																																
偏移地址: 38H																																
复位值: 00000000_00000000_00000000_00000000 _b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
													TFOVER		TFEMPTY	TFTH	TBC	RFUERR	RFOERR	RFFULL			RFTH	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bits 31-19	—	—
TFOVER	Bit 18	R	<p>发送器FIFO溢出中断标志 当发送器FIFO内已有16个数据时,有新数据再次写入FIFO中时,将会置起此位并舍弃新数据。此位由硬件设置1,设置ICR寄存器的TFOVER位清除 0: 无中断产生 1: 发送器FIFO溢出中断产生</p>
—	Bit 17	—	—
TFEMPTY	Bit 16	R	<p>发送器FIFO空中断标志 发送器FIFO由一个数值转为空时置起,或设置IER寄存器的TFEMPTY位时,UART会判断当前发送器FIFO是否为空将此位置起。此位由硬件设置1,设置ICR寄存器的TFEMPTY位清除 0: 无中断产生 1: 发送器FIFO空中断产生</p>
TFTH	Bit 15	R	<p>发送器FIFO触发阈值中断标志 发送器FIFO个数到达发送器设定的阈值时置起,或设置IER寄存器的TFTH位时,UART会判断当前发送器FIFO个数到达发送器设定的阈值将此位置起。此位由硬件设置1,设置ICR寄存器的TFTH位清除 0: 无中断产生 1: 发送器FIFO触发阈值中断产生</p>
TBC	Bit 14	R	<p>发送器字节完成中断标志 发送器完成单个字节时置起。此位由硬件设置1,设置ICR寄存器的TBC位清除 0: 无中断产生 1: 发送器字节完成中断产生</p>
RFUERR	Bit 13	R	<p>接收器FIFO下溢中断标志 当接收器FIFO内无数据时,又再次读取接收器FIFO时置起。此位由硬件设置1,设置ICR寄存器</p>

			的RFUERR位清除 0: 无中断产生 1: 接收器FIFO下溢中断产生
RFOERR	Bit 12	R	接收器FIFO溢出中断标志 当接收器FIFO内已有16个数据时,有新数据再次接收至FIFO中时置起并舍弃新数据。此位由硬件设置1, 设置ICR寄存器的RFOERR位清除 0: 无中断产生 1: 接收器FIFO溢出中断产生
RFFULL	Bit 11	R	接收器FIFO满中断标志 当接收器FIFO内由15个数据接收至16个数据时置起, 或设置IER寄存器的RFFULL位时, UART会判断当前发送器FIFO是否为16个数据而置起。此位由硬件设置1, 设置ICR寄存器的RFFULL位清除 0: 无中断产生 1: 接收器FIFO满中断产生
—	Bit 10	—	—
RFTH	Bit 9	R	接收器FIFO触发门槛中断标志 接收器FIFO个数到达发送器设定的门槛时置起, 或设置IER寄存器的RFTH位时, UART会判断当前发送器FIFO个数到达发送器设定的门槛将此位置起。此位由硬件设置1, 设置ICR寄存器的RFTH位清除 0: 无中断产生 1: 接收器FIFO触发门槛中断产生
NOISE	Bit 8	R	噪声位侦测中断标志 此位由硬件设置1, 设置ICR寄存器的NOISE位清除 0: 无中断产生 1: 噪声位侦测中断产生
EOB	Bit 7	R	块结束中断标志 (Only UART) 接收器接收个数等于SCARD寄存器的BLEN位时置起。此位由硬件设置1, 设置ICR寄存器的EOB位清除 0: 无中断产生 1: 块结束中断产生
LINBK	Bit 6	R	LIN断开侦测中断标志 (Only UART) 在LIN模式中, 当接收器侦测到断开字符时置起。此位由硬件设置1, 设置ICR寄存器的LINBK位清除 0: 无中断产生 1: LIN断开侦测中断产生
ADDRM	Bit 5	R	地址匹配中断标志

			<p>接收器接收到的字符匹配RS485寄存器的ADDR位时置起。此位由硬件设置1，设置ICR寄存器的ADDRM位清除</p> <p>0: 无中断产生</p> <p>1: 地址匹配中断产生</p>
RXTO	Bit 4	R	<p>接收超时中断标志</p> <p>接收最后一个字节后，在超时时间内未检测到新的起始位时置起。此位由硬件设置1，设置ICR寄存器的RXTO位清除</p> <p>0: 无中断产生</p> <p>1: 接收超时中断产生</p>
DCTS	Bit 3	R	<p>CTS_n引脚电平改变中断标志</p> <p>CTS_n引脚电平改变时置起。此位由硬件设置1，设置ICR寄存器的DCTS位清除</p> <p>0: 无中断产生</p> <p>1: CTS_n引脚电平改变中断产生</p>
ABTO	Bit 2	R	<p>自动波特率侦测超时中断标志</p> <p>自动波特率在侦测到下降沿后，在时间内未侦测到上升沿时置起。此位由硬件设置1，设置ICR寄存器的ABTO位清除</p> <p>0: 无中断产生</p> <p>1: 自动波特率侦测超时中断产生</p>
ABEND	Bit 1	R	<p>自动波特率侦测结束中断标志</p> <p>自动波特率完成时置起。此位由硬件设置1，设置ICR寄存器的ABEND位清除</p> <p>0: 无中断产生</p> <p>1: 自动波特率侦测结束中断产生</p>
RXBERR	Bit 0	R	<p>接收器字节格式错误中断标志</p> <p>接收器接收到的字符发生校验错误与帧错误时置起。此位由硬件设置1，设置ICR寄存器的RXBERR位清除</p> <p>0: 无中断产生</p> <p>1: 接收器字节格式错误中断产生</p>

10.5.2.16 屏蔽中断标志寄存器(UART_IFM)

无屏蔽中断标志寄存器 (UART_IFM)																																
偏移地址: 3C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
													TFOVER		TFEMPTY	TFTH	TBC	RFUERR	RFOERR	RFFULL			RFTH	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bits 31-19	—	—
TFOVER	Bit 18	R	发送器FIFO溢出中断屏蔽标志 当IVS寄存器的TFOVER位为1时,发送器FIFO内已有16个数据时,有新数据再次写入FIFO中时,将会置起此位并舍弃新数据。此位由硬件设置1,设置ICR寄存器的TFOVER位清除 0: 无中断产生 1: 发送器FIFO溢出中断产生
—	Bit 17	—	—
TFEMPTY	Bit 16	R	发送器FIFO空中断屏蔽标志 当IVS寄存器的TFEMPTY位为1时,发送器FIFO由一个数值转为空时置起,或设置IER寄存器的TFEMPTY位时,UART会判断当前发送器FIFO是否为空将此位置起。此位由硬件设置1,设置ICR寄存器的TFEMPTY位清除 0: 无中断产生 1: 发送器FIFO空中断产生
TFTH	Bit 15	R	发送器FIFO触发门槛中断屏蔽标志 当IVS寄存器的TFTH位为1时,发送器FIFO个数到达发送器设定的门槛时置起,或设置IER寄存器的TFTH位时,UART会判断当前发送器FIFO个数到达发送器设定的门槛将此位置起。此位由硬件设置1,设置ICR寄存器的TFTH位清除 0: 无中断产生 1: 发送器FIFO触发门槛中断产生
TBC	Bit 14	R	发送器字节完成中断屏蔽标志 当IVS寄存器的TBC位为1时,发送器完成单个字节时置起。此位由硬件设置1,设置ICR寄存器的TBC位清除 0: 无中断产生 1: 发送器字节完成中断产生
RFUERR	Bit 13	R	接收器FIFO下溢中断屏蔽标志

			<p>当IVS寄存器的RFUERR位为1时，接收器FIFO内无数据时，又再次读取接收器FIFO时置起。此位由硬件设置1，设置ICR寄存器的RFUERR位清除</p> <p>0: 无中断产生 1: 接收器FIFO下溢中断产生</p>
RFOERR	Bit 12	R	<p>接收器FIFO溢出中断屏蔽标志</p> <p>当IVS寄存器的RFOERR位为1时，接收器FIFO内已有16个数据时，有新数据再次接收至FIFO中时置起并舍弃新数据。此位由硬件设置1，设置ICR寄存器的RFOERR位清除</p> <p>0: 无中断产生 1: 接收器FIFO溢出中断产生</p>
RFFULL	Bit 11	R	<p>接收器FIFO满中断屏蔽标志</p> <p>当IVS寄存器的RFFULL位为1时，接收器FIFO内由15个数据接收至16个数据时置起，或设置IER寄存器的RFFULL位时，UART会判断当前发送器FIFO是否为16个数据而置起。此位由硬件设置1，设置ICR寄存器的RFFULL位清除</p> <p>0: 无中断产生 1: 接收器FIFO满中断产生</p>
—	Bit 10	—	—
RFTH	Bit 9	R	<p>接收器FIFO触发阈值中断屏蔽标志</p> <p>当IVS寄存器的RFTH位为1时，接收器FIFO个数到达发送器设定的阈值时置起，或设置IER寄存器的RFTH位时，UART会判断当前发送器FIFO个数到达发送器设定的阈值将此位置起。此位由硬件设置1，设置ICR寄存器的RFTH位清除</p> <p>0: 无中断产生 1: 接收器FIFO触发阈值中断产生</p>
NOISE	Bit 8	R	<p>噪声位侦测中断屏蔽标志</p> <p>当IVS寄存器的NOISE位为1时。此位由硬件设置1，设置ICR寄存器的NOISE位清除</p> <p>0: 无中断产生 1: 噪声位侦测中断产生</p>
EOB	Bit 7	R	<p>块结束中断屏蔽标志 (Only UART)</p> <p>当IVS寄存器的EOB位为1时，接收器接收个数等于SCARD寄存器的BLEN位时置起。此位由硬件设置1，设置ICR寄存器的EOB位清除</p> <p>0: 无中断产生 1: 块结束中断产生</p>
LINBK	Bit 6	R	<p>LIN断开侦测中断屏蔽标志 (Only UART)</p> <p>当IVS寄存器的LINBK位为1时，LIN模式中，当接收器侦测到断开字符时置起。此位由硬件设置1，</p>

			<p>设置ICR寄存器的LINBK位清除</p> <p>0: 无中断产生</p> <p>1: LIN断开侦测中断产生</p>
ADDRM	Bit 5	R	<p>地址匹配中断屏蔽标志</p> <p>当IVS寄存器的ADDRM位为1时，接收器接收到的字符匹配RS485寄存器的ADDR位时置起。此位由硬件设置1，设置ICR寄存器的ADDRM位清除</p> <p>0: 无中断产生</p> <p>1: 地址匹配中断产生</p>
RXTO	Bit 4	R	<p>接收超时中断屏蔽标志</p> <p>当IVS寄存器的RXTO位为1时，接收最后一个字节后，在超时时间内未检测到新的起始位时置起。此位由硬件设置1，设置ICR寄存器的RXTO位清除</p> <p>0: 无中断产生</p> <p>1: 接收超时中断产生</p>
DCTS	Bit 3	R	<p>CTS_n引脚电平改变中断屏蔽标志</p> <p>当IVS寄存器的DCTS位为1时，CTS_n引脚电平改变时置起。此位由硬件设置1，设置ICR寄存器的DCTS位清除</p> <p>0: 无中断产生</p> <p>1: CTS_n引脚电平改变中断产生</p>
ABTO	Bit 2	R	<p>自动波特率侦测超时中断屏蔽标志</p> <p>当IVS寄存器的ABTO为1时，自动波特率在侦测到下降沿后，在时间内未侦测到上升沿时置起。此位由硬件设置1，设置ICR寄存器的ABTO位清除</p> <p>0: 无中断产生</p> <p>1: 自动波特率侦测超时中断产生</p>
ABEND	Bit 1	R	<p>自动波特率侦测结束中断屏蔽标志</p> <p>当IVS寄存器的ABEND位为1时，自动波特率完成时置起。此位由硬件设置1，设置ICR寄存器的ABEND位清除</p> <p>0: 无中断产生</p> <p>1: 自动波特率侦测结束中断产生</p>
RXBERR	Bit 0	R	<p>接收器字节格式错误中断屏蔽标志</p> <p>当IVS寄存器的RXBERR位为1时，接收器接收到的字符发生校验错误与帧错误时置起。此位由硬件设置1，设置ICR寄存器的RXBERR位清除</p> <p>0: 无中断产生</p> <p>1: 接收器字节格式错误中断产生</p>

10.5.2.17 中断清除寄存器 (UART_ICR)

中断清除寄存器(UART_ICR)																																
偏移地址: 40 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
													TFOVER		TFEMPTY	TFTH	TBC	RFUERR	RFOERR	RFFULL			RFTH	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bits 31-19	—	—
TFOVER	Bit 18	C_W1	发送器FIFO溢出中断清除 0: 写入0无效 1: 发送器FIFO溢出中断清除
—	Bit 17	—	—
TFEMPTY	Bit 16	C_W1	发送器FIFO空中断清除 0: 写入0无效 1: 发送器FIFO空中断清除
TFTH	Bit 15	C_W1	发送器FIFO触发阈值中断清除 0: 写入0无效 1: 发送器FIFO触发阈值中断清除
TBC	Bit 14	C_W1	发送器字节完成中断清除 0: 写入0无效 1: 发送器字节完成中断清除
RFUERR	Bit 13	C_W1	接收器FIFO下溢中断清除 0: 写入0无效 1: 接收器FIFO下溢中断清除
RFOERR	Bit 12	C_W1	接收器FIFO溢出中断清除 0: 写入0无效 1: 接收器FIFO溢出中断清除
RFFULL	Bit 11	C_W1	接收器FIFO满中断清除 0: 写入0无效 1: 接收器FIFO满中断清除
—	Bit 10	—	—
RFTH	Bit 9	C_W1	接收器FIFO触发阈值中断清除 0: 写入0无效 1: 接收器FIFO触发阈值中断清除
NOISE	Bit 8	C_W1	噪声位侦测中断清除 0: 写入0无效 1: 噪声位侦测中断清除
EOB	Bit 7	C_W1	块结束中断清除 (Only UART) 0: 写入0无效

			1: 块结束中断清除
LINBK	Bit 6	C_W1	LIN断开侦测中断清除 (Only UART) 0: 写入0无效 1: LIN断开侦测中断清除
ADDRM	Bit 5	C_W1	地址匹配中断清除 0: 写入0无效 1: 地址匹配中断清除
RXTO	Bit 4	C_W1	接收超时中断清除 0: 写入0无效 1: 接收超时中断清除
DCTS	Bit 3	C_W1	CTS_n引脚电平改变中断清除 0: 写入0无效 1: CTS _n 引脚电平改变中断清除
ABTO	Bit 2	C_W1	自动波特率侦测超时中断清除 0: 写入0无效 1: 自动波特率侦测超时中断清除
ABEND	Bit 1	C_W1	自动波特率侦测结束中断清除 0: 写入0无效 1: 自动波特率侦测结束中断清除
RXBERR	Bit 0	C_W1	接收器字节格式错误中断清除 0: 写入0无效 1: 接收器字节格式错误中断清除

第11章 外部中断 (EXTI)

11.1 概述

外部中断和事件控制器 (EXTI) 管理外部和内部异步事件 / 中断，并生成相应的事件请求到 CPU/ 中断控制器和到电源管理的唤醒请求。

EXTI 允许管理多达 22 个外部 / 内部事件线，可以在停止模式唤醒设备。

其中一些行是可配置的：在这种情况下，可以独立地选择活动边沿，并且状态标志指示中断的来源。可配置的线路由 I/Os 外部中断和少数外围设备使用。有些线路是直接的：它们被一些外围设备用来从停止事件或中断产生唤醒。在这种情况下，状态标志由外围设备提供。

作为外部或内部事件请求的每一线都可独立配置。EXTI 控制器还允许通过编程到专用寄存器，通过与相应的硬件事件线复用的软件来仿真事件或中断。

11.2 特性

EXTI 主要特性如下：

- ◆ 支持产生多达 22 个事件 / 中断请求
- ◆ 每个事件 / 中断线都有独立的屏蔽
- ◆ 可配置上升沿触发或下降沿触发
- ◆ 每个外部中断线都有专用的状态位
- ◆ 仿真所有的外部事件请求

11.3 结构图

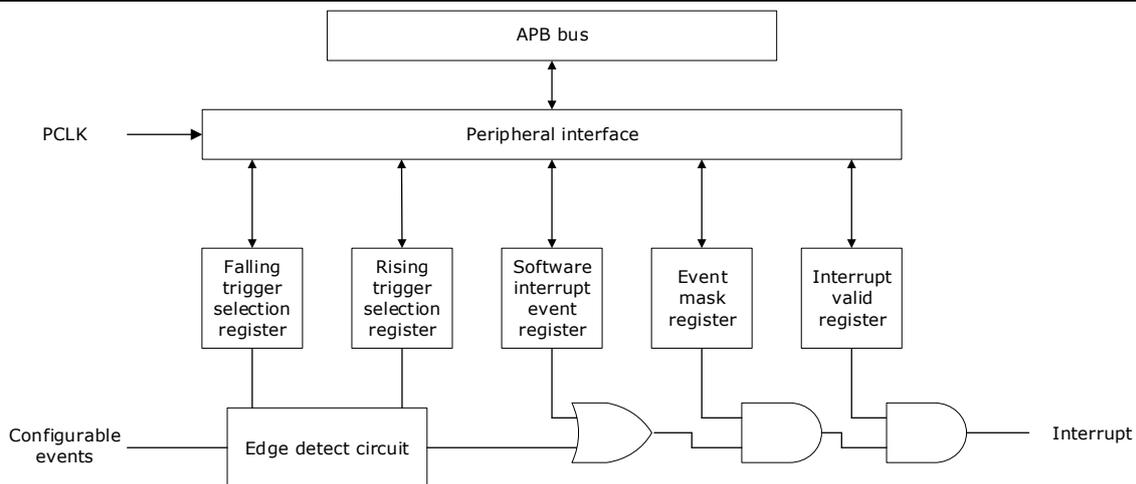


图 11-1 外部中断 / 事件框图

11.4 功能描述

11.4.1 硬件中断选择

硬件发生中断所需要的步骤为下：

1. 设定中断的 **EXTI_IER** 以及 **EXTI_IDR**，来开启或是关闭中断的来源。
2. 选择中断 Trigger 的触发方式，为 **EXTI_RTS** 或 **EXTI_FTS** 触发。
3. 用户可以由 **EXTI_IVS** 来判断开启中断的位置，中断发生后，可以由 **EXTI_IFM** 来观察中断是否发生。
4. 当中断产生后，设定 **EXTI_ICR**，用户即可清除中断。

11.4.2 软件事件选择

用户使用软件为了在某时间点发生中断，步骤为下：

1. 设定中断的 **EXTI_IER** 以及 **EXTI_IDR**，来开启或是关闭中断的来源。
2. 用户可以设定 **EXTI_SWI**，让在某些时间点可以发出中断。

11.4.3 外部和内部中断 / 事件线路映射

因为 GPIO 有 A,B,C,D, 每一组又各有[15:0]的信号, 四组 GPIO 共享 EXTI 的中断, 用户必须在先前就设定, 是要打开哪个 GPIO。依照 **EXTI_ICFG1/EXTI_ICFG2** 两个寄存器, 可以选择该中断是由哪组的 GPIO 产生。

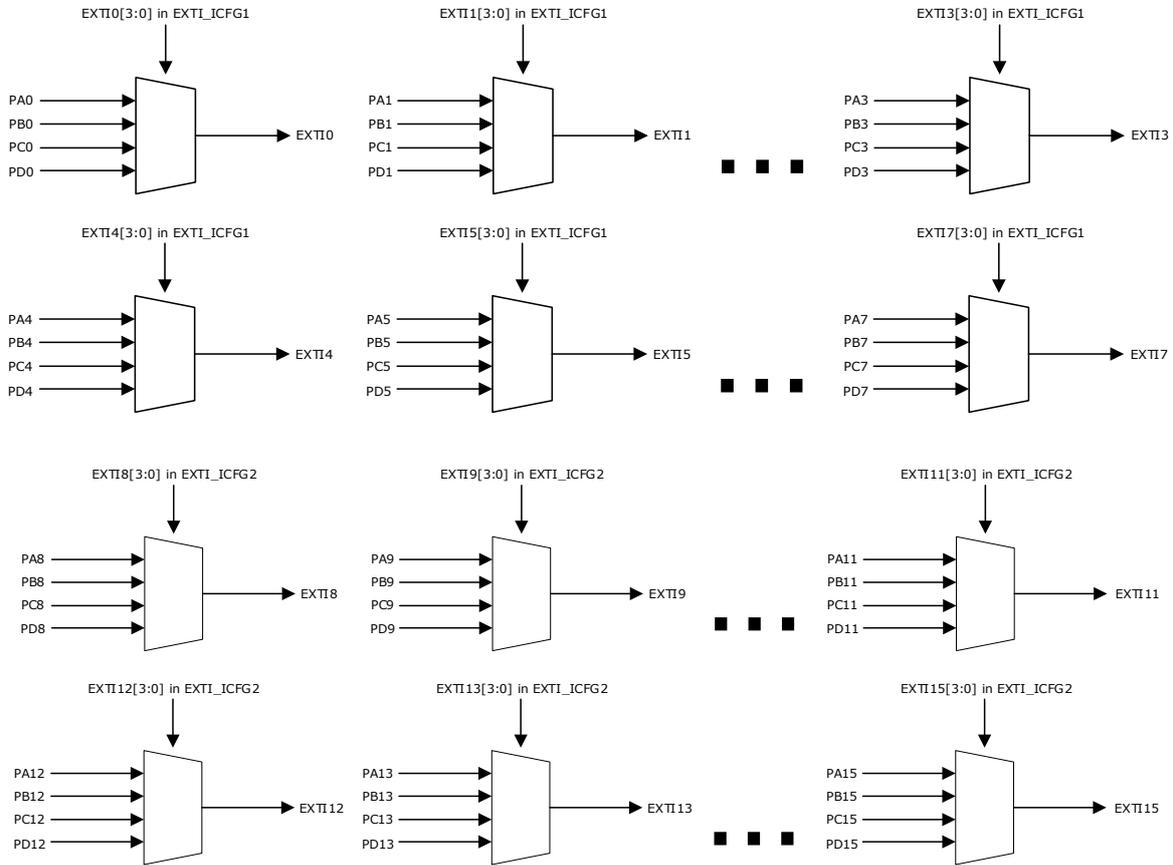


图 11-2 外部中断 / 事件 GPIO 映射

EXTI 线路	中断/事件	类型
0-15	GPIO	可配置
16	CMP1	可配置
17	CMP2	可配置
18	CMP3	可配置
19	CMP4	可配置
20	PVD	可配置
21	WAKEUP	可配置
22-23	保留	

表 11-1 EXTI 线路连接

11.5 特殊功能寄存器

11.5.1 寄存器列表

外设寄存器可支持半字（16 位）或字（32 位）访问。

EXTI 寄存器列表			
名称	偏移地址	类型	描述
EXTI_IER	0000 _H	W1	EXTI 中断使能寄存器
EXTI_IDR	0004 _H	W1	EXTI 中断禁止寄存器
EXTI_IVS	0008 _H	R	EXTI 中断有效位状态寄存器
EXTI_RIF	000C _H	R	EXTI 无屏蔽中断标志寄存器
EXTI_IFM	0010 _H	R	EXTI 屏蔽中断标志寄存器
EXTI_ICR	0014 _H	C_W1	EXTI 中断清除寄存器
EXTI_RTS	0018 _H	R/W	EXTI 上升沿触发选择寄存器
EXTI_FTS	001C _H	R/W	EXTI 下降沿触发选择寄存器
EXTI_SWI	0020 _H	R/W	EXTI 软件中断事件寄存器
EXTI_ADTE	0024 _H	R/W	EXTI ADC 触发使能寄存器
EXTI_DB	0028 _H	R/W	EXTI 输入去抖使能寄存器
EXTI_DBCON	002C _H	R/W	EXTI 输入去抖采样率控制寄存器
EXTI_ICFG1	0030 _H	R/W	EXTI 中断配置寄存器 1
EXTI_ICFG2	0034 _H	R/W	EXTI 中断配置寄存器 2

11.5.2 寄存器描述

11.5.2.1 EXTI 中断使能寄存器 (EXTI_IER)

EXTI 中断使能寄存器(EXTI_IER)																															
偏移地址: 00H																															
复位值: 00000000_00000000_00000000_00000000B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

名称	位	类型	描述
—	Bits 31-22	—	—
WAKEUP	Bit 21	W1	<p>IEy: 在EXTIx上启用中断 这些位只写。读这些位时返回0x0000数值。 0: 对相应的 IEx 位无影响 1: 将端口位配置为中断, 并对 IVx 状态位进行响应。</p>
PVD0	Bit 20	W1	
CMP4	Bit 19	W1	
CMP3	Bit 18	W1	
CMP2	Bit 17	W1	
CMP1	Bit 16	W1	
GPIO15	Bit 15	W1	
GPIO14	Bit 14	W1	
GPIO13	Bit 13	W1	
GPIO12	Bit 12	W1	
GPIO11	Bit 11	W1	
GPIO10	Bit 10	W1	
GPIO9	Bit 9	W1	
GPIO8	Bit 8	W1	
GPIO7	Bit 7	W1	
GPIO6	Bit 6	W1	
GPIO5	Bit 5	W1	
GPIO4	Bit 4	W1	
GPIO3	Bit 3	W1	
GPIO2	Bit 2	W1	
GPIO1	Bit 1	W1	
GPIO0	Bit 0	W1	

11.5.2.2 EXTI 中断禁止寄存器 (EXTI_IDR)

EXTI 中断禁止寄存器(EXTI_IDR)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

名称	位	类型	描述
—	Bits 31-22	—	—
WAKEUP	Bit 21	W1	<p>IDy: 禁用EXTIx中断 这些位只写。读这些位时返回0x0000数值。 0: 对相应的 IDx 位无影响 1: 禁用端口位作为中断, 并对 IVx 状态位进行响应</p>
PVD0	Bit 20	W1	
CMP4	Bit 19	W1	
CMP3	Bit 18	W1	
CMP2	Bit 17	W1	
CMP1	Bit 16	W1	
GPIO15	Bit 15	W1	
GPIO14	Bit 14	W1	
GPIO13	Bit 13	W1	
GPIO12	Bit 12	W1	
GPIO11	Bit 11	W1	
GPIO10	Bit 10	W1	
GPIO9	Bit 9	W1	
GPIO8	Bit 8	W1	
GPIO7	Bit 7	W1	
GPIO6	Bit 6	W1	
GPIO5	Bit 5	W1	
GPIO4	Bit 4	W1	
GPIO3	Bit 3	W1	
GPIO2	Bit 2	W1	
GPIO1	Bit 1	W1	
GPIO0	Bit 0	W1	

11.5.2.3 EXTI 中断有效位状态寄存器 (EXTI_IVS)

EXTI 中断有效位状态寄存器(EXTI_IVS)																															
偏移地址: 08H																															
复位值: 00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

名称	位	类型	描述
—	Bits 31-22	—	—
WAKEUP	Bit 21	R	Ivy: EXTIx中断有效位状态 这些位是只读的，并且根据IEy和IDy设置。 0: 无效的中断位 1: 有效中断位
PVD0	Bit 20	R	
CMP4	Bit 19	R	
CMP3	Bit 18	R	
CMP2	Bit 17	R	
CMP1	Bit 16	R	
GPIO15	Bit 15	R	
GPIO14	Bit 14	R	
GPIO13	Bit 13	R	
GPIO12	Bit 12	R	
GPIO11	Bit 11	R	
GPIO10	Bit 10	R	
GPIO9	Bit 9	R	
GPIO8	Bit 8	R	
GPIO7	Bit 7	R	
GPIO6	Bit 6	R	
GPIO5	Bit 5	R	
GPIO4	Bit 4	R	
GPIO3	Bit 3	R	
GPIO2	Bit 2	R	
GPIO1	Bit 1	R	
GPIO0	Bit 0	R	

11.5.2.4 EXTI 无屏蔽中断标志寄存器 (EXTI_RIF)

EXTI 无屏蔽中断标志寄存器(EXTI_RIF)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

名称	位	类型	描述
—	Bits 31-22	—	—
WAKEUP	Bit 21	R	IFy: EXTIx无屏蔽中断标志 这些位只读。 0: 无中断 1: 中断要求
PVD0	Bit 20	R	
CMP4	Bit 19	R	
CMP3	Bit 18	R	
CMP2	Bit 17	R	
CMP1	Bit 16	R	
GPIO15	Bit 15	R	
GPIO14	Bit 14	R	
GPIO13	Bit 13	R	
GPIO12	Bit 12	R	
GPIO11	Bit 11	R	
GPIO10	Bit 10	R	
GPIO9	Bit 9	R	
GPIO8	Bit 8	R	
GPIO7	Bit 7	R	
GPIO6	Bit 6	R	
GPIO5	Bit 5	R	
GPIO4	Bit 4	R	
GPIO3	Bit 3	R	
GPIO2	Bit 2	R	
GPIO1	Bit 1	R	
GPIO0	Bit 0	R	

11.5.2.5 EXTI 屏蔽中断标志寄存器 (EXTI_IFM)

EXTI 屏蔽中断标志寄存器(EXTI_IFM)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

名称	位	类型	描述
—	Bits 31-22	—	—
WAKEUP	Bit 21	R	IFMy: EXTIx屏蔽中断标志 这些位只读。 0: 无中断 1: 中断要求
PVD0	Bit 20	R	
CMP4	Bit 19	R	
CMP3	Bit 18	R	
CMP2	Bit 17	R	
CMP1	Bit 16	R	
GPIO15	Bit 15	R	
GPIO14	Bit 14	R	
GPIO13	Bit 13	R	
GPIO12	Bit 12	R	
GPIO11	Bit 11	R	
GPIO10	Bit 10	R	
GPIO9	Bit 9	R	
GPIO8	Bit 8	R	
GPIO7	Bit 7	R	
GPIO6	Bit 6	R	
GPIO5	Bit 5	R	
GPIO4	Bit 4	R	
GPIO3	Bit 3	R	
GPIO2	Bit 2	R	
GPIO1	Bit 1	R	
GPIO0	Bit 0	R	

11.5.2.6 EXTI 中断清除寄存器 (EXTI_ICR)

EXTI 中断清除寄存器(EXTI_ICR)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

名称	位	类型	描述
—	Bits 31-22	—	—
WAKEUP	Bit 21	C_W1	ICy: EXTIx中断清除 这些位只写。 0: 没有效果 1: 清除中断标志
PVD0	Bit 20	C_W1	
CMP4	Bit 19	C_W1	
CMP3	Bit 18	C_W1	
CMP2	Bit 17	C_W1	
CMP1	Bit 16	C_W1	
GPIO15	Bit 15	C_W1	
GPIO14	Bit 14	C_W1	
GPIO13	Bit 13	C_W1	
GPIO12	Bit 12	C_W1	
GPIO11	Bit 11	C_W1	
GPIO10	Bit 10	C_W1	
GPIO9	Bit 9	C_W1	
GPIO8	Bit 8	C_W1	
GPIO7	Bit 7	C_W1	
GPIO6	Bit 6	C_W1	
GPIO5	Bit 5	C_W1	
GPIO4	Bit 4	C_W1	
GPIO3	Bit 3	C_W1	
GPIO2	Bit 2	C_W1	
GPIO1	Bit 1	C_W1	
GPIO0	Bit 0	C_W1	

11.5.2.7 EXTI 上升沿触发选择寄存器 (EXTI_RTS)

EXTI 上升沿触发选择寄存器(EXTI_RTS)																															
偏移地址: 18H																															
复位值: 00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

名称	位	类型	描述
—	Bits 31-22	—	—
WAKEUP	Bit 21	R/W	RTSx: 上升沿触发事件配置(x = 21 to 0) 0: 上升沿触发禁止(事件和中断) 1: 上升沿触发使能 (事件和中断)
PVD0	Bit 20	R/W	
CMP4	Bit 19	R/W	
CMP3	Bit 18	R/W	
CMP2	Bit 17	R/W	
CMP1	Bit 16	R/W	
GPIO15	Bit 15	R/W	
GPIO14	Bit 14	R/W	
GPIO13	Bit 13	R/W	
GPIO12	Bit 12	R/W	
GPIO11	Bit 11	R/W	
GPIO10	Bit 10	R/W	
GPIO9	Bit 9	R/W	
GPIO8	Bit 8	R/W	
GPIO7	Bit 7	R/W	
GPIO6	Bit 6	R/W	
GPIO5	Bit 5	R/W	
GPIO4	Bit 4	R/W	
GPIO3	Bit 3	R/W	
GPIO2	Bit 2	R/W	
GPIO1	Bit 1	R/W	
GPIO0	Bit 0	R/W	

11.5.2.8 EXTI 下降沿触发选择寄存器 (EXTI_FTS)

EXTI 下降沿触发选择寄存器(EXTI_FTS)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

名称	位	类型	描述
—	Bits 31-22	—	—
WAKEUP	Bit 21	R/W	FTSx: 下降沿触发事件配置 (x = 21 to 0) 0: 下降沿触发禁止 (事件和中断) 1: 下降沿触发使能 (事件和中断)
PVD0	Bit 20	R/W	
CMP4	Bit 19	R/W	
CMP3	Bit 18	R/W	
CMP2	Bit 17	R/W	
CMP1	Bit 16	R/W	
GPIO15	Bit 15	R/W	
GPIO14	Bit 14	R/W	
GPIO13	Bit 13	R/W	
GPIO12	Bit 12	R/W	
GPIO11	Bit 11	R/W	
GPIO10	Bit 10	R/W	
GPIO9	Bit 9	R/W	
GPIO8	Bit 8	R/W	
GPIO7	Bit 7	R/W	
GPIO6	Bit 6	R/W	
GPIO5	Bit 5	R/W	
GPIO4	Bit 4	R/W	
GPIO3	Bit 3	R/W	
GPIO2	Bit 2	R/W	
GPIO1	Bit 1	R/W	
GPIO0	Bit 0	R/W	

11.5.2.9 EXTI 软件中断事件寄存器 (EXTI_SWI)

EXTI 软件中断事件寄存器(EXTI_SWI)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

—	Bits 31-22	—	—
WAKEUP	Bit 21	W1	<p>SWIx: 软件中断 (x = 21 to 0) 这些位可由软件写来配置, 用户可以让中断按时发生 0: 中断不发生 1: 按时发生中断</p>
PVD0	Bit 20	W1	
CMP4	Bit 19	W1	
CMP3	Bit 18	W1	
CMP2	Bit 17	W1	
CMP1	Bit 16	W1	
GPIO15	Bit 15	W1	
GPIO14	Bit 14	W1	
GPIO13	Bit 13	W1	
GPIO12	Bit 12	W1	
GPIO11	Bit 11	W1	
GPIO10	Bit 10	W1	
GPIO9	Bit 9	W1	
GPIO8	Bit 8	W1	
GPIO7	Bit 7	W1	
GPIO6	Bit 6	W1	
GPIO5	Bit 5	W1	
GPIO4	Bit 4	W1	
GPIO3	Bit 3	W1	
GPIO2	Bit 2	W1	
GPIO1	Bit 1	W1	
GPIO0	Bit 0	W1	

11.5.2.10 EXTI ADC 触发使能寄存器 (EXTI_ADTE)

EXTI ADC 触发使能寄存器(EXTI_ADTE)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

—	Bits 31-22	—	—
WAKEUP	Bit 21	R/W	ADTE_x: ADC 触发使能(x = 21 to 0) 0: ADC中断触发禁止 1: ADC中断触发使能
PVD0	Bit 20	R/W	
CMP4	Bit 19	R/W	
CMP3	Bit 18	R/W	
CMP2	Bit 17	R/W	
CMP1	Bit 16	R/W	
GPIO15	Bit 15	R/W	
GPIO14	Bit 14	R/W	
GPIO13	Bit 13	R/W	
GPIO12	Bit 12	R/W	
GPIO11	Bit 11	R/W	
GPIO10	Bit 10	R/W	
GPIO9	Bit 9	R/W	
GPIO8	Bit 8	R/W	
GPIO7	Bit 7	R/W	
GPIO6	Bit 6	R/W	
GPIO5	Bit 5	R/W	
GPIO4	Bit 4	R/W	
GPIO3	Bit 3	R/W	
GPIO2	Bit 2	R/W	
GPIO1	Bit 1	R/W	
GPIO0	Bit 0	R/W	

11.5.2.11 EXTI 输入去抖使能寄存器 (EXTI_DB)

EXTI 输入去抖使能寄存器(EXTI_DB)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WAKEUP	PVD0	CMP4	CMP3	CMP2	CMP1	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

—	Bits 31-22	—	—
WAKEUP	Bit 21	R/W	DBENx: 输入去抖使能 0:去抖禁止 1:去抖使能
PVD0	Bit 20	R/W	
CMP4	Bit 19	R/W	
CMP3	Bit 18	R/W	
CMP2	Bit 17	R/W	
CMP1	Bit 16	R/W	
GPIO15	Bit 15	R/W	
GPIO14	Bit 14	R/W	
GPIO13	Bit 13	R/W	
GPIO12	Bit 12	R/W	
GPIO11	Bit 11	R/W	
GPIO10	Bit 10	R/W	
GPIO9	Bit 9	R/W	
GPIO8	Bit 8	R/W	
GPIO7	Bit 7	R/W	
GPIO6	Bit 6	R/W	
GPIO5	Bit 5	R/W	
GPIO4	Bit 4	R/W	
GPIO3	Bit 3	R/W	
GPIO2	Bit 2	R/W	
GPIO1	Bit 1	R/W	
GPIO0	Bit 0	R/W	

11.5.2.12 EXTI 输入去抖采样率控制寄存器 (EXTI_DBCON)

EXTI 输入去抖采样率控制寄存器(EXTI_DBCON)																																	
偏移地址: 2C _H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																DBPRE<7:0>																DBCNT<2:0>	

—	Bits 31-16	—	—
DBPRE	Bits 15-8	R/W	DBPRE: 去抖预分频器
—	Bits 7-3	—	—
DBCNT	Bits 2-0	R/W	DBCNT: 去抖计数器

11.5.2.13 EXTI 中断配置寄存器 1 (EXTI_ICFG1)

EXTI 中断配置寄存器 1 (EXTI_ICFG1)																															
偏移地址: 30 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7<3:0>				GPIO6<3:0>				GPIO5<3:0>				GPIO4<3:0>				GPIO3<3:0>				GPIO2<3:0>				GPIO1<3:0>				GPIO0<3:0>			

GPIO7	Bits 31-28	R/W	EXTI 中断配置 (x = 0 to 7) 这些位由软件进行改写来选择EXTI的外部中断源。 0000: PA[x] 引脚 0001: PB[x] 引脚 0010: PC[x] 引脚 0011: PD[x] 引脚 其他配置保留。
GPIO6	Bits 27-24	R/W	
GPIO5	Bits 23-20	R/W	
GPIO4	Bits 19-16	R/W	
GPIO3	Bits 15-12	R/W	
GPIO2	Bits 11-8	R/W	
GPIO1	Bits 7-4	R/W	
GPIO0	Bits 3-0	R/W	

11.5.2.14 EXTI 中断配置寄存器 2 (EXTI_ICFG2)

EXTI 中断配置寄存器 2 (EXTI_ICFG2)																															
偏移地址: 34 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO15<3:0>				GPIO14<3:0>				GPIO13<3:0>				GPIO12<3:0>				GPIO11<3:0>				GPIO10<3:0>				GPIO9<3:0>				GPIO8<3:0>			

GPIO	Bits	R/W	EXTI 中断配置 (x = 8 to 15) 这些位由软件进行改写来选择EXTI的外部中断源。 0000: PA[x] 引脚 0001: PB[x] 引脚 0010: PC[x] 引脚 0011: PD[x] 引脚 其他配置保留。
GPIO15	Bits 31-28	R/W	
GPIO14	Bits 27-24	R/W	
GPIO13	Bits 23-20	R/W	
GPIO12	Bits 19-16	R/W	
GPIO11	Bits 15-12	R/W	
GPIO10	Bits 11-8	R/W	
GPIO9	Bits 7-4	R/W	
GPIO8	Bits 3-0	R/W	

第12章 模数转换器 (ADC)

12.1 概述

模数转换器 (ADC) 是将连续模拟电压转换为离散数字电压的外设。ADC 包含一个 12 位逐次逼近型模数转换器 (SARA/D 转换器), 支持 8 种倍率的 PGA 输入讯号放大, 并带有 16 个外部输入信道。ADC 模块支持四个可编程序列发生器, 无需控制器干预即可对多个模拟输入源进行采样。A/D 转换器支持三种工作模式: 单次转换模式, 单周扫描模式和连续扫描模式。每个采样序列都提供灵活的编程, 具有完全可配置的输入源, 触发事件和中断。

12.2 特性

ADC 模块提供以下特性:

- ◆ 模拟输入电压范围: 0.15V 至(VREF-0.15V)
- ◆ 多达 16 个单端模拟输入通道或 8 个差分模拟输入通道
- ◆ 支持可编程采样时间(ADC_CLK)
- ◆ 支持 8 种倍率的 PGA 输入讯号放大(可放大至 8 倍)
- ◆ 可于时钟频率为 48 MHz 下进行取样操作, 每次采样时间为 20 个周期
- ◆ 多种启动转换方式
 - ◇ 控制器 (软件)
 - ◇ 定时器 GP32C4T1
 - 定时器 GP16C4T1/ GP16C4T2/ GP16C4T3
 - 定时器 GP16C2T1/GP16C2T2/GP16C2T3/GP16C2T4
 - 定时器 AD16C4T1 比较器 1/2/3/4
 - ◇ GPIO
 - ◇ RTC
- ◆ A/D 转换可以在
 - ◇ 可以在指定信道上执行一次
 - ◇ 在所有指定信道上一个循环, 从最低编号信道到最高编号信道的序列
 - ◇ 单周期扫描直到软件停止 A/D 转换
- ◆ 使用 DMA 高效传输

12.3 结构图

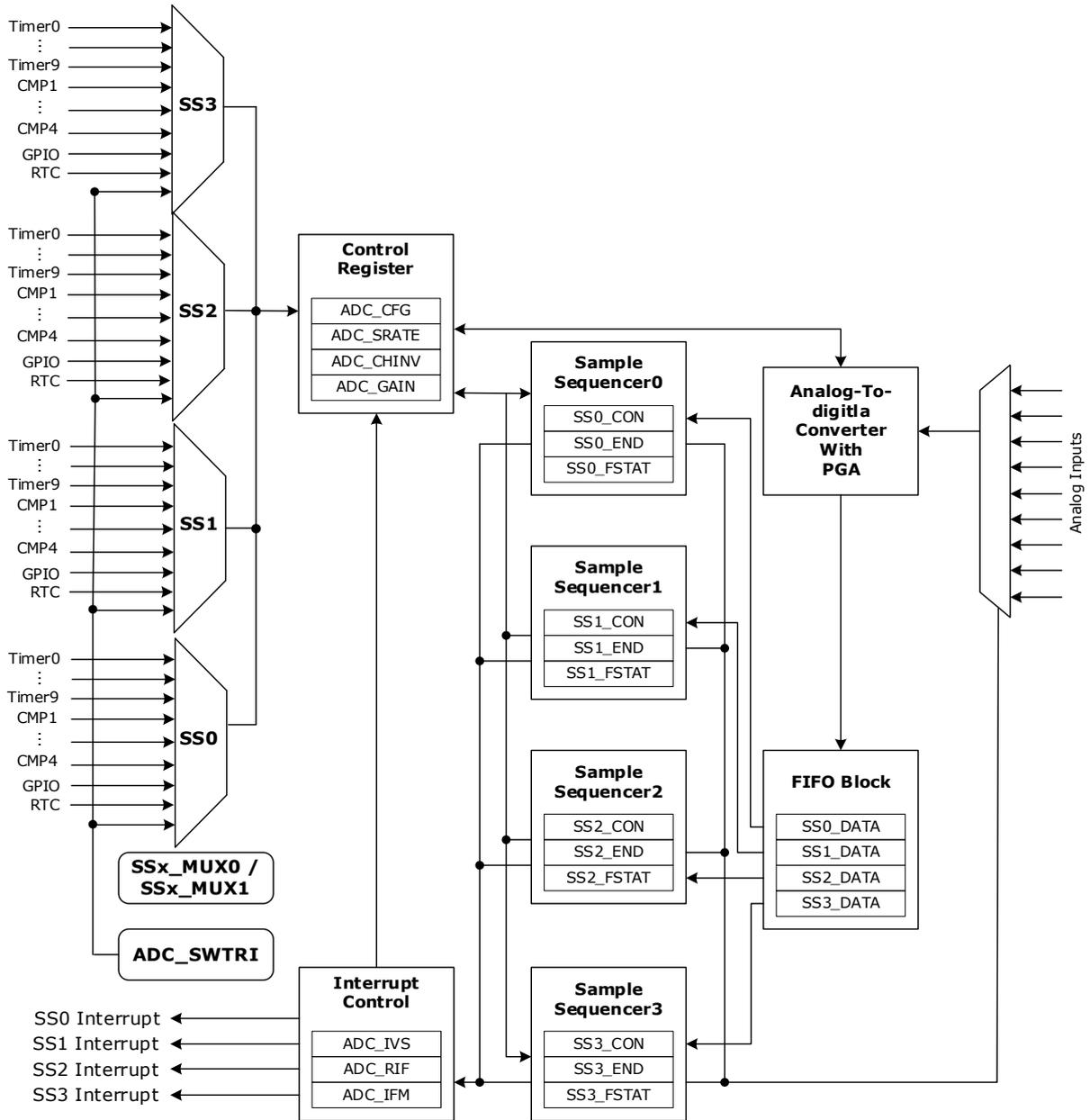


图 12-1 ADC 模块图

12.4 功能描述

12.4.1 采样序列发生器

采样控制和数据采集由采样序列发生器来处理，除了可捕获的样本数量和 FIFO 的深度之外，所有的序列发生器都是相同的。下表显示了每个序列发生器可以支持的最大采样数及其相应的 FIFO 深度。

序列发生器	样本数	FIFO 深度
SS0	16	16
SS1	8	8
SS2	4	4
SS3	1	1

表 12-1 序列发生器的采样和 FIFO 深度

对于给定的采样序列，每个采样由 ADC 样本序列输入多路复用器选择寄存器(SSx_MUX0 /SSx_MUX1)和 ADC 采样序列控制寄存器(SSx_CON) 中的位域定义。SSx_MUX0 / SSx_MUX1 字段选择输入引脚，而 SSx_CON 字段包含对应于诸如中断启用、序列结束和差分输入模式等参数的采样控制位。通过在 ADC 活动采样序列器(ADC_SSEN)寄存器中设置相应的 SSxEN 位来启用采样序列发生器，并且应该在启用采样之前进行配置，然后通过 ADC 软件触发器(ADC_PSSI)寄存器中设置 SSx 位来初始化启用采样。

在配置采样序列时，允许在同一序列内多次使用相同的输入引脚。在 SSx_END 寄存器中，IEx 位可以设置为采样的任何组合，如果必要，允许在序列中的每个采样之后生成中断。此外，可以在采样序列内的任意点设置 END 位。例如，如果使用序列发生器 0，则可以在与第五采样相关联的过程中设置结束位，允许序列发生器 0 完成第五采样后结束采样序列的执行。

在采样序列完成执行之后，可以从采样序列 x 数据 (SSx_DATA) 寄存器中检索结果数据。FIFOs 是采样的循环缓冲区，它读取单个地址到“pop”结果数据。为了软件调试的目的，FIFO 头部和尾部指针的位置在采样序列 FIFO 状态(SSx_FSTAT)寄存器中以及 FULL 和 EMPTY 状态标志中是可见的。如果在 FIFO 满时尝试写入，写入不会发生，并且发生溢出。溢出和下溢条件也使用采样序列 FIFO 状态寄存器 (SSx_FSTAT) 进行监控。

12.4.2 取样时钟

ADC 模块的时钟默认使用 APB 时钟，每一次的取样耗时 20 个 ADC 时钟周期，而 ADC 支持用户将 ADC 时钟配置在 48MHz 的频率下进行取样，因此理论上最高的取样频率为 2.4MHz。ADC 控制器支持用户配置 ADC_SRATE 寄存器来调整 ADC 时钟的频率，用户可在不降低 APB 时钟频率的条件下调整 ADC 的取样频率。

12.4.3 采样优先次序

具有相同优先级的多个有源采样序列发生器单元不能提供一致的结果，因此，用户必须确保所有启动的采样序列发生器单元具有唯一值。当采样事件（触发器）同时发生时，它们根据 ADC 采样序列发生器控制寄存器 (SSx_CON) 中的值决定处理的优先次序。有效优先级值在 0-3 的范围中，其中 0 是最高优先级，3 是最低值。

12.4.4 采样事件/触发器

每个采样序列器的事件/触发在 ADC 采样序列器复用器选择(SSx_MUX0 / SSx_MUX1)寄存器中定义。触发源包括处理器软件触发器(默认)、模拟比较器和 GPIO 指定的 GPIO 上的外部信号、定时器连续采样。处理器通过设置 ADC 软件触发控制寄存器(ADC_SWTRI)中的 SSX 位来触发采样。

用户使用连续采样触发器时必须小心。如果序列发生器的优先级太高,就有可能阻塞其他较低优先级的序列发生器。通常,使用连续采样的采样序列发生器应设置为最低优先级。连续采样可与数字比较器一起使用,以在输入上看到特定电压时引起中断。

12.4.5 输入讯号放大

ADC 支持用户藉由配置 PGA 将输入讯号进行放大,用户可配置 ADC_GAINL 与 ADC_GAINH 寄存器决定每一个输入讯号的放大倍率。用户在配置 PGA 倍率时,需注意 ADC 所使用的参考电压数值(VREF),避免因输入讯号被过度放大而导致取样失真。此外当不同输入讯号配置的 PGA 倍率差异太大时,建议用户可于切换输入讯号时增加等待时间,以增加 ADC 取样的可靠度。

12.4.6 中断控制

采样序列发生器的寄存器配置显示哪些事件产生无屏蔽中断,但无法控制中断是否实际发送到中断控制器。ADC 中的中断由一组六个寄存器控制。

◆ 中断控制 (IER, IDR, IVS)

ADC 中断使能寄存器(ADC_IER)通过写入“1”来启用中断请求。类似地,ADC 中断禁用寄存器(ADC_IDR)通过写入“1”来禁用中断请求。IER 和 IDR 是只写寄存器,它控制着中断的屏蔽。这两个寄存器的总结果可由中断有效状态寄存器(ADC_IVS)显示。IVS 是只读寄存器,使用“1”或“0”来显示中断请求线是启用还是禁用。

◆ 中断状态读取 (RIF, IFM)

无屏蔽中断标志状态(ADC_RIF)是只读寄存器,读取模块的所有中断状态。屏蔽中断标志寄存器(ADC_IFM)用于指示模块的非屏蔽中断状态,因为只有无屏蔽的中断才会被置给处理器。

◆ 中断清除状态 (ICR)

向该寄存器中的位写“1”可以清除相应的中断状态或通过向 IDR 写 1 来禁止中断。

12.4.7 DMA 请求

转换后的通道值将存储在相应的数据寄存器中。ADC 可以通过中断通知 MCU 存储新数据。在以下情况下生成 DMA 请求:

◆ A/D 采样序列发生器的 FIFO 已满

突发传输:整个数据块以一个连续的顺序传输。

◆ 完成单个 A/D 转换

单次传输:如果 ADC_DMA.DMAENx 位置 1,则在每次 A/D 转换结束时置位单个 DMA 请求。

12.4.8 初始化及配置

12.4.8.1 模块初始化

为了使用 ADC 模块，PLL 必须被启用和编程。使用不支持的频率会导致 ADC 模块的错误操作。ADC 模块的初始化是一个简单的过程，只需要很少的步骤：启用 ADC 的时钟，用模拟输入对 IO 进行多路复用，以及重新配置样本序列发生器优先级。

ADC 的初始化顺序如下：

1. 使用系统控制寄存器使能 ADC 时钟：使能 APB 外设门控时钟寄存器(**SC_GCLK_APB**) 中的 ADC 位。
2. 为 ADC 输入引脚设置 GPIO 的 **GPIOx_AFH** 和 **GPIOx_AFL** 寄存器。请参阅**錯誤! 找不到參照來源。錯誤! 找不到參照來源。和錯誤! 找不到參照來源。**以找出要启用的 GPIO 引脚。
3. 如果应用程序需要，请重新配置 **SSx_CON** 寄存器中的采样序列发生器优先级。默认配置采样序列发生器 0 具有最高优先级和采样序列发生器 3 具有最低优先级。
4. 用户可依序需求配置 **ADC_GAINL** 与 **ADC_GAINH** 开启 PGA 放大器放大输入讯号，最大支持将输入讯号放大 8 倍，且每一组输入讯号皆支持个别配置 PGA 放大倍率。

12.4.8.2 采样序列发生器配置

每个采样序列发生器的配置如下：

1. 通过清零 **ADC_SSEN** 寄存器中的相应 **SSxEN** 位，确保禁止采样序列器。在不启用它们的情况下，允许对采样序列发生器进行编程。如果在配置过程中发生触发事件，则在编程期间禁止采样序列器可防止错误执行。
2. 在 **SSx_CON** 寄存器中配置采样序列发生器的触发事件。
3. 对于采样序列中的每个采样，在 **SSx_MUX0 / SSx_MUX1** 寄存器中配置相应的输入源。
4. 对于采样序列中的每个采样，在 **SSx_END** 寄存器中配置相应的半字节的采样控制位。在编程最后一个字节时，确保设置 **END** 位。未能设置 **END** 位会导致不可预测的行为。
5. 如果要使用中断，则在 **ADC_IER / IDR** 寄存器中设置相应的 **MASK** 位。
6. 通过将 **ADC_SSEN** 寄存器中的相应 **SSxEN** 位置 1 来使能采样序列发生器逻辑。

12.5 特殊功能寄存器

12.5.1 寄存器列表

ADC 寄存器列表			
名称	偏移地址	类型	描述
ADC_CFG	0000 _H	R/W	ADC 配置寄存器
ADC_SRATE	0004 _H	R/W	ADC 采样速度控制寄存器
ADC_CHINV	0008 _H	R/W	ADC 信道反转控制寄存器
ADC_GAINL	000C _H	R/W	ADC 信道 0 ~ 7 PGA 增益控制寄存器
ADC_GAINH	0010 _H	R/W	ADC 信道 8 ~ 15 PGA 增益控制寄存器
ADC_FRF	0014 _H	R/W	ADC FIFO 刷新寄存器
ADC_SSEN	0018 _H	R/W	ADC 采样序列发生器启用寄存器
ADC_SWTRI	001C _H	R/W	ADC 软件触发控制寄存器
ADC_IER	0020 _H	W1	ADC 中断使能寄存器
ADC_IDR	0024 _H	W1	ADC 中断禁止寄存器
ADC_IVS	0028 _H	R	ADC 中断有效位状态寄存器
ADC_RIF	002C _H	R	ADC 无屏蔽中断标志寄存器
ADC_IFM	0030 _H	R	ADC 屏蔽中断标志寄存器
ADC_ICR	0034 _H	C_W1	ADC 中断清除寄存器
ADC_DMA	0038 _H	R/W	ADC DMA 控制寄存器
ADC_SS0_CON	0040 _H	R/W	ADC 采样序列发生器 0 控制寄存器
ADC_SS0_MUX0	0044 _H	R/W	ADC 采样序列发生器 0 输入多路复用器选择 0 寄存器
ADC_SS0_MUX1	0048 _H	R/W	ADC 采样序列发生器 0 输入多路复用器选择 1 寄存器
ADC_SS0_END	004C _H	R/W	ADC 采样序列发生器 0 结束控制寄存器
ADC_SS0_FSTAT	0050 _H	R/W	ADC 采样序列发生器 0 FIFO 状态寄存器
ADC_SS0_DATA	0054 _H	R/W	ADC 采样序列发生器 0 结果 FIFO 寄存器
ADC_SS1_CON	0060 _H	R/W	ADC 采样序列发生器 1 控制寄存器
ADC_SS1_MUX0	0064 _H	R/W	ADC 采样序列发生器 1 输入多路复用器选择 0 寄存器
ADC_SS1_END	0068 _H	R/W	ADC 采样序列发生器 1 结束控制寄存器
ADC_SS1_FSTAT	006C _H	R/W	ADC 采样序列发生器 1 FIFO 状态寄存器
ADC_SS1_DATA	0070 _H	R/W	ADC 采样序列发生器 1 结果 FIFO 寄存器
ADC_SS2_CON	0078 _H	R/W	ADC 采样序列发生器 2 控制寄存器
ADC_SS2_MUX0	007C _H	R/W	ADC 采样序列发生器 2 输入多路复用器选择 0 寄存器
ADC_SS2_END	0080 _H	R/W	ADC 采样序列发生器 2 结束控制寄存器
ADC_SS2_FSTAT	0084 _H	R/W	ADC 采样序列发生器 2 FIFO 状态寄存器
ADC_SS2_DATA	0088 _H	R/W	ADC 采样序列发生器 2 结果 FIFO 寄存器
ADC_SS3_CON	0090 _H	R/W	ADC 采样序列发生器 3 控制寄存器
ADC_SS3_MUX0	0094 _H	R/W	ADC 采样序列发生器 3 输入多路复用器

ADC 寄存器列表			
名称	偏移地址	类型	描述
			选择 0 寄存器
ADC_SS3_END	0098 _H	R/W	ADC 采样序列发生器 3 结束控制寄存器
ADC_SS3_FSTAT	009C _H	R/W	ADC 采样序列发生器 3 FIFO 状态寄存器
ADC_SS3_DATA	00A0 _H	R/W	ADC 采样序列发生器 3 结果 FIFO 寄存器

12.5.2.3 ADC 通道反转控制寄存器(ADC_CHINV)

ADC 通道反转控制寄存器(ADC_CHINV)																															
偏移地址: 08H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CH15INV	CH14INV	CH13INV	CH12INV	CH11INV	CH10INV	CH9INV	CH8INV	CH7INV	CH6INV	CH5INV	CH4INV	CH3INV	CH2INV	CH1INV	CH0INV

—	Bits 31-16	—	—
CH15INV	Bit 15	R/W	通道15反向控制 该寄存器用于反转信道15的数据 0: 不反转信道15的数据 1: 反转信道15的数据
CH14INV	Bit 14	R/W	通道14反向控制 该寄存器用于反转信道14的数据
CH13INV	Bit 13	R/W	通道13反向控制 该寄存器用于反转信道13的数据
CH12INV	Bit 12	R/W	通道 12 反向控制 该寄存器用于反转信道 12 的数据
CH11INV	Bit 11	R/W	通道 11 反向控制 该寄存器用于反转信道 11 的数据
CH10INV	Bit 10	R/W	通道 10 反向控制 该寄存器用于反转信道 10 的数据
CH9INV	Bit 9	R/W	通道 9 反向控制 该寄存器用于反转信道 9 的数据
CH8INV	Bit 8	R/W	通道8反向控制 该寄存器用于反转信道8的数据
CH7INV	Bit 7	R/W	通道 7 反向控制 该寄存器用于反转信道 7 的数据
CH6INV	Bit 6	R/W	通道6反向控制 该寄存器用于反转信道6的数据
CH5INV	Bit 5	R/W	通道5反向控制 该寄存器用于反转信道5的数据
CH4INV	Bit 4	R/W	通道 4 反向控制 该寄存器用于反转信道 4 的数据
CH3INV	Bit 3	R/W	通道 3 反向控制 该寄存器用于反转信道 3 的数据
CH2INV	Bit 2	R/W	通道2反向控制 该寄存器用于反转信道2的数据
CH1INV	Bit 1	R/W	通道1反向控制

12.5.2.16 ADC 采样序列发生器 0 控制寄存器(ADC_SS0_CON)

ADC 采样序列发生器 0 控制寄存器(ADC_SS0_CON)																															
偏移地址: 40H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SEL<4:0>						TYP			PRI<1:0>						ONE

—	Bits 31-16	—	—
SEL	Bits 15-11	R/W	SS0 触发选择 该字段选择用于采样序列发生器 0 的触发源 此字段的有效配置是: 0x0 SW 控制器 (默认) 0x1 BS16T1 0x2 Always (连续采样) 0x3 GP32C4T1 0x4 GP16C4T1 0x5 GP16C4T2 0x6 GP16C4T3 0x7 GP16C2T1 0x8 GP16C2T2 0x9 GP16C2T3 0xA GP16C2T4 0xB AD16C4T1 0xC GPIO 0xD RTC 0xE CMP1 0xF CMP2 0x10 CMP3 0x11 CMP4 0x12~0x1F —
—	Bits 10-9	—	—
TYP	Bit 8	R/W	ADC 采样序列发生器 0 (SS0) 触发器类型选择 用户可以设置此位来选择触发器的类型 0: 边沿触发 1: 电平触发
—	Bits 7-6	—	—
PRI	Bits 5-4	R/W	ADC 采样序列发生器 0 (SS0) 优先级 此字段包含指定 SS0 优先级编码的二进制编码值。分配给顺控程序的优先级必须是唯一映射的。 0x0: 最高优先级 0x1: 第二优先 0x2: 第三优先 0x3: 最低优先级
—	Bits 3-1	—	—
ONE	Bit 0	R/W	单次触发模式

			设置为 HIGH 时，每个触发器仅采样一次并在 FIFO 中使用一个插槽(slot)。
--	--	--	---

12.5.2.17 ADC 采样序列发生器 0 输入多路复用器选择 0 寄存器(ADC_SS0_MUX0)

ADC 采样序列发生器 0 输入多路复用器选择 0 寄存器(ADC_SS0_MUX0)																															
偏移地址: 44H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7<3:0>				MUX6<3:0>				MUX5<3:0>				MUX4<3:0>				MUX3<3:0>				MUX2<3:0>				MUX1<3:0>				MUX0<3:0>			

MUX7	Bits 31-28	R/W	第 8 采样输入选择
MUX6	Bits 27-24	R/W	第 7 采样输入选择
MUX5	Bits 23-20	R/W	第 6 采样输入选择
MUX4	Bits 19-16	R/W	第 5 采样输入选择
MUX3	Bits 15-12	R/W	第 4 采样输入选择
MUX2	Bits 11-8	R/W	第 3 采样输入选择
MUX1	Bits 7-4	R/W	第 2 采样输入选择
MUX0	Bits 3-0	R/W	第 1 采样输入选择 用户可以设置此寄存器来选择哪个信道将被采样。例如，值为 1 表示输入为通道 1。

12.5.2.18 ADC 采样序列发生器 0 输入多路复用器选择 1 寄存器(ADC_SS0_MUX1)

ADC 采样序列发生器 0 输入多路复用器选择 1 寄存器(ADC_SS0_MUX1)																															
偏移地址: 48 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX15<3:0> [△]				MUX14<3:0> [△]				MUX13<3:0> [△]				MUX12<3:0> [△]				MUX11<3:0> [△]				MUX10<3:0> [△]				MUX9<3:0> [△]				MUX8<3:0> [△]			

MUX15	Bits 31-28	R/W	第 16 采样输入选择
MUX14	Bits 27-24	R/W	第 15 采样输入选择
MUX13	Bits 23-20	R/W	第 14 采样输入选择
MUX12	Bits 19-16	R/W	第 13 采样输入选择
MUX11	Bits 15-12	R/W	第 12 采样输入选择
MUX10	Bits 11-8	R/W	第 11 采样输入选择
MUX9	Bits 7-4	R/W	第 10 采样输入选择
MUX8	Bits 3-0	R/W	第 9 采样输入选择 用户可以设置此寄存器来选择哪个信道将被采样。例如, 值为 1 表示输入为通道 1。

12.5.2.19 ADC 采样序列发生器 0 结束控制寄存器(ADC_SS0_END)

ADC 采样序列发生器 0 结束控制寄存器(ADC_SS0_END)																																			
偏移地址: 4C _H																																			
复位值: 00000000_00000000_00000000_00001111 _B																																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
—	—	—	—	—	—	—	—	—	—	—	—	IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0	—	—	—	—	END<3:0>			

—	Bits 31-24	—	—
IE15	Bit 23	R/W	第 16 个采样中断使能
IE14	Bit 22	R/W	第 15 个采样中断使能
IE13	Bit 21	R/W	第 14 个采样中断使能
IE12	Bit 20	R/W	第 13 次采样中断使能
IE11	Bit 19	R/W	第 12 个采样中断使能
IE10	Bit 18	R/W	第 11 个采样中断使能
IE9	Bit 17	R/W	第 10 个采样中断使能
IE8	Bit 16	R/W	第 9 个采样中断使能
IE7	Bit 15	R/W	第 8 个采样中断使能
IE6	Bit 14	R/W	第 7 个采样中断使能
IE5	Bit 13	R/W	第 6 个采样中断使能
IE4	Bit 12	R/W	第 5 个采样中断使能
IE3	Bit 11	R/W	第 4 个采样中断使能
IE2	Bit 10	R/W	第 3 个采样中断使能
IE1	Bit 9	R/W	第 2 个采样中断使能
IE0	Bit 8	R/W	第 1 个采样中断使能 如果 ADC_IER 寄存器置 1, 则中断将提升为控制器级中断。该位置 1 时, 原始中断置位。该位清零时, 原始中断未置位。在序列中生成多个采样生成中断是合法的。
—	Bits 7-4	—	—
END	Bits 3-0	R/W	结束采样 用户可以设置该寄存器以选择 ADC 采样数据的次数

12.5.2.20 ADC 采样序列发生器 0 FIFO 状态寄存器(ADC_SS0_FSTAT)

ADC 采样序列发生器 0 FIFO 状态寄存器(ADC_SS0_FSTAT)																																
偏移地址: 50H																																
复位值: 00000000_00000000_00000001_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					OV	FULL	UV	EMPTY	HPTR<3:0>				TPTR<3:0>			

—	Bits 31-12	—	—
OV	Bit 11	R/W	ADC 采样序列发生器 0 (SS0) FIFO 溢出状态 0: FIFO 没有溢出 1: SS0 FIFO 有位溢出条件, 其中在 FIFO 已满, 请求写入。当检测到溢出时, 最新的写入被删除。
FULL	Bit 10	R	FIFO Full 0: FIFO 当前未滿 1: FIFO 目前已滿
UV	Bit 9	R/W	ADC 采样序列发生器 0 (SS0) FIFO 下溢状态 0: FIFO 没有下溢 1: SS0 FIFO 有一个下溢的条件, 即在 FIFO 空请求读。有问题的读取不会移动 FIFO 指针, 并返回 0。
EMPTY	Bit 8	R	FIFO 空 0: FIFO 当前不是空的 1: FIFO 目前是空的
HPTR	Bits 7-4	R	FIFO 头指针 该字段包含当前“头”指针索引, 它是 FIFO 的下一个要写入的条目。 0 – 15: FIFO 0
TPTR	Bits 3-0	R	FIFO 尾指针 该字段包含当前“尾部”指针索引, 它是 FIFO 的下一个要读取的条目 0 – 15: FIFO 0

12.5.2.21 ADC 采样序列发生器 0 结果 FIFO 寄存器(ADC_SS0_DATA)

ADC 采样序列发生器 0 结果 FIFO 寄存器(ADC_SS0_DATA)																																											
偏移地址: 54H																																											
复位值: 00000000_00000000_00000000_00000000 _B																																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
																					DATA<11:0>																						

—	Bits 31-12	—	—
DATA	Bits 11-0	R/W	转换结果数据 用户可以通过读取该寄存器来读取转换结果数据

			设置为 HIGH 时，每个触发器仅采样一次并在 FIFO 中使用一个插槽(slot)。
--	--	--	---

12.5.2.23 ADC 采样序列发生器 1 输入多路复用器选择 0 寄存器(ADC_SS1_MUX0)

ADC 采样序列发生器 1 输入多路复用器选择 0 寄存器(ADC_SS1_MUX0)																															
偏移地址: 64H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7<3:0>				MUX6<3:0>				MUX5<3:0>				MUX4<3:0>				MUX3<3:0>				MUX2<3:0>				MUX1<3:0>				MUX0<3:0>			

MUX7	Bits 31-28	R/W	第 8 采样输入选择
MUX6	Bits 27-24	R/W	第 7 采样输入选择
MUX5	Bits 23-20	R/W	第 6 采样输入选择
MUX4	Bits 19-16	R/W	第 5 采样输入选择
MUX3	Bits 15-12	R/W	第 4 采样输入选择
MUX2	Bits 11-8	R/W	第 3 采样输入选择
MUX1	Bits 7-4	R/W	第 2 采样输入选择
MUX0	Bits 3-0	R/W	第 1 采样输入选择 用户可以设置此寄存器来选择哪个信道将被采样。例如，值为 1 表示输入为通道 1。

12.5.2.24 ADC 采样序列发生器 1 结束控制寄存器(ADC_SS1_END)

ADC 采样序列发生器 1 结束控制寄存器(ADC_SS1_END)																																
偏移地址: 68H																																
复位值: 00000000_00000000_00000000_00000111b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0									END<2:0>

—	Bits 31-16	—	—
IE7	Bit 15	R/W	第 8 个采样中断使能
IE6	Bit 14	R/W	第 7 个采样中断使能
IE5	Bit 13	R/W	第 6 个采样中断使能
IE4	Bit 12	R/W	第 5 个采样中断使能
IE3	Bit 11	R/W	第 4 个采样中断使能
IE2	Bit 10	R/W	第 3 个采样中断使能
IE1	Bit 9	R/W	第 2 个采样中断使能
IE0	Bit 8	R/W	第 1 个采样中断使能 如果 ADC_IER 寄存器置 1, 则中断将提升为控制器级中断。该位置 1 时, 原始中断置位。该位清零时, 原始中断未置位。在序列中生成多个采样生成中断是合法的。
—	Bits 7-3	—	—
END	Bits 2-0	R/W	结束采样 用户可以设置该寄存器以选择 ADC 采样数据的次数

12.5.2.25 ADC 采样序列发生器 1 FIFO 状态寄存器(ADC_SS1_FSTAT)

ADC 采样序列发生器 1 FIFO 状态寄存器(ADC_SS1_FSTAT)																																
偏移地址: 6C _H																																
复位值: 00000000_00000000_00000001_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					OV	FULL	UV	EMPTY	HPTR<3:0>				TPTR<3:0>			

—	Bits 31-12	—	—
OV	Bit 11	R/C_W1	ADC 采样序列发生器 1 (SS1) FIFO 溢出状态 0: FIFO 没有溢出 1: SS1 FIFO 有位溢出条件, 其中在 FIFO 已满, 请求写入。当检测到溢出时, 最新的写入被删除。
FULL	Bit 10	R	FIFO Full 0: FIFO 当前未满 1: FIFO 目前已满
UV	Bit 9	R/C_W1	ADC 采样序列发生器 1 (SS1) FIFO 下溢状态 0: FIFO 没有下溢 1: SS1 FIFO 有一个下溢的条件, 即在 FIFO 空请求读。有问题的读取不会移动 FIFO 指针, 并返回 0。
EMPTY	Bit 8	R	FIFO 空 0: FIFO 当前不是空的 1: FIFO 目前是空的
HPTR	Bits 7-4	R	FIFO 头指针 该字段包含当前“头”指针索引, 它是 FIFO 的下一个要写入的条目。 0 – 15: FIFO 1
TPTR	Bits 3-0	R	FIFO 尾指针 该字段包含当前“尾部”指针索引, 它是 FIFO 的下一个要读取的条目 0 – 15: FIFO 1

12.5.2.26 ADC 采样序列发生器 1 结果 FIFO 寄存器(ADC_SS1_DATA)

ADC 采样序列发生器 1 结果 FIFO 寄存器(ADC_SS1_DATA)																															
偏移地址: 70 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																					DATA<11:0>										

—	Bits 31-12	—	—
DATA	Bits 11-0	R/W	转换结果数据 用户可以通过读取该寄存器来读取转换结果数据

12.5.2.29 ADC 采样序列发生器 2 结束控制寄存器(ADC_SS2_END)

ADC 采样序列发生器 2 结束控制寄存器(ADC_SS2_END)																																
偏移地址: 80H																																
复位值: 00000000_00000000_00000000_00000011 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					IE3	IE2	IE1	IE0								END<1:0>

—	Bits 31-12	—	—
IE3	Bit 11	R/W	第 4 个采样中断使能
IE2	Bit 10	R/W	第 3 个采样中断使能
IE1	Bit 9	R/W	第 2 个采样中断使能
IE0	Bit 8	R/W	第 1 个采样中断使能 如果 ADC_IER 寄存器置 1, 则中断将提升为控制器级中断。该位置 1 时, 原始中断置位。该位清零时, 原始中断未置位。在序列中生成多个采样生成中断是合法的。
—	Bits 7-2	—	—
END	Bits 1-0	R/W	结束采样 用户可以设置该寄存器以选择 ADC 采样数据的次数

12.5.2.30 ADC 采样序列发生器 2 FIFO 状态寄存器(ADC_SS2_FSTAT)

ADC 采样序列发生器 2 FIFO 状态寄存器(ADC_SS2_FSTAT)																																
偏移地址: 84 _H																																
复位值: 00000000_00000000_00000001_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					OV	FULL	UV	EMPTY	HPTR<3:0>				TPTR<3:0>			

—	Bits 31-12	—	—
OV	Bit 11	R/C_W1	ADC 采样序列发生器 2 (SS2) FIFO 溢出状态 0: FIFO 没有溢出 1: SS2 FIFO 有位溢出条件, 其中在 FIFO 已满, 请求写入。当检测到溢出时, 最新的写入被删除。
FULL	Bit 10	R	FIFO Full 0: FIFO 当前未满 1: FIFO 目前已满
UV	Bit 9	R/C_W1	ADC 采样序列发生器 2 (SS2) FIFO 下溢状态 0: FIFO 没有下溢 1: SS2 FIFO 有一个下溢的条件, 即在 FIFO 空请求读。有问题的读取不会移动 FIFO 指针, 并返回 0。
EMPTY	Bit 8	R	FIFO 空 0: FIFO 当前不是空的 1: FIFO 目前是空的
HPTR	Bits 7-4	R	FIFO 头指针 该字段包含当前“头”指针索引, 它是 FIFO 的下一个要写入的条目。 0 – 15: FIFO 2
TPTR	Bits 3-0	R	FIFO 尾指针 该字段包含当前“尾部”指针索引, 它是 FIFO 的下一个要读取的条目 0 – 15: FIFO 2

12.5.2.31 ADC 采样序列发生器 2 结果 FIFO 寄存器(ADC_SS2_DATA)

ADC 采样序列发生器 2 结果 FIFO 寄存器(ADC_SS2_DATA)																															
偏移地址: 88H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																					DATA<11:0>										

—	Bits 31-12	—	—
DATA	Bits 11-0	R/W	转换结果数据 用户可以通过读取该寄存器来读取转换结果数据

			设置为 HIGH 时，每个触发器仅采样一次并在 FIFO 中使用一个插槽(slot)。
--	--	--	---

12.5.2.33 ADC 采样序列发生器 3 输入多路复用器选择 0 寄存器(ADC_SS3_MUX0)

ADC 采样序列发生器 3 输入多路复用器选择 0 寄存器(ADC_SS3_MUX0)																															
偏移地址: 94H																															
复位值: 00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												MUX0<3:0			

—	Bits 31-4	—	—
MUX0	Bits 3-0	R/W	第 1 采样输入选择 用户可以设置此寄存器来选择哪个信道将被采样。例如，值为 1 表示输入为通道 1。

12.5.2.34 ADC 采样序列发生器 3 结束控制寄存器(ADC_SS3_END)

ADC 采样序列发生器 3 结束控制寄存器(ADC_SS3_END)																																
偏移地址: 98H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																												IE0	END			

—	Bits 31-9	—	—
IE0	Bit 8	R/W	第 1 个采样中断使能 如果 ADC_IER 寄存器置 1，则中断将提升为控制器级中断。该位置 1 时，原始中断置位。该位清零时，原始中断未置位。在序列中生成多个采样生成中断是合法的。
—	Bits 7-1	—	—
END	Bit 0	R/W	结束采样 用户可以设置该寄存器以选择 ADC 采样数据的次数

12.5.2.35 ADC 采样序列发生器 3 FIFO 状态寄存器(ADC_SS3_FSTAT)

ADC 采样序列发生器 3 FIFO 状态寄存器(ADC_SS3_FSTAT)																																
偏移地址: 9C _H																																
复位值: 00000000_00000000_00000001_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					OV	FULL	UV	EMPTY	HPTR<3:0>				TPTR<3:0>			

—	Bits 31-12	—	—
OV	Bit 11	R/C_W1	ADC 采样序列发生器 3 (SS3) FIFO 溢出状态 0: FIFO 没有溢出 1: SS3 FIFO 有位溢出条件, 其中在 FIFO 已满, 请求写入。当检测到溢出时, 最新的写入被删除。
FULL	Bit 10	R	FIFO Full 0: FIFO 当前未滿 1: FIFO 目前已滿
UV	Bit 9	R/C_W1	ADC 采样序列发生器 3 (SS3) FIFO 下溢状态 0: FIFO 没有下溢 1: SS3 FIFO 有一个下溢的条件, 即在 FIFO 空请求读。有问题的读取不会移动 FIFO 指针, 并返回 0。
EMPTY	Bit 8	R	FIFO 空 0: FIFO 当前不是空的 1: FIFO 目前是空的
HPTR	Bits 7-4	R	FIFO 头指针 该字段包含当前“头”指针索引, 它是 FIFO 的下一个要写入的条目。 0 – 15: FIFO 3
TPTR	Bits 3-0	R	FIFO 尾指针 该字段包含当前“尾部”指针索引, 它是 FIFO 的下一个要读取的条目 0 – 15: FIFO 3

12.5.2.36 ADC 采样序列发生器 3 结果 FIFO 寄存器(ADC_SS3_DATA)

ADC 采样序列发生器 3 结果 FIFO 寄存器(ADC_SS3_DATA)																																											
偏移地址: A0 _H																																											
复位值: 00000000_00000000_00000000_00000000 _B																																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
																					DATA<11:0>																						

—	Bits 31-12	—	—
DATA	Bits 11-0	R/W	转换结果数据 用户可以通过读取该寄存器来读取转换结果数据

第13章 模拟比较器 (CMP)

13.1 概述

模拟比较器是一个外围设备，它可以比较两个模拟电压的值，并以逻辑输出的形式显示比较结果。模拟比较器支持四个单独的比较器（CMP1、CMP2、CMP3 和 CMP4），每个比较器可以向设备管脚提供输出并替换电路板上的模拟比较器。CMP 可以用来通过中断触发 ADC，或者通知应用程序开始捕获样本序列。独立的外部参考电压。

13.2 特性

- ◆ 四个单独的模拟比较器
- ◆ 共享内部参考电压
- ◆ 可编程去抖计数器

13.3 结构图

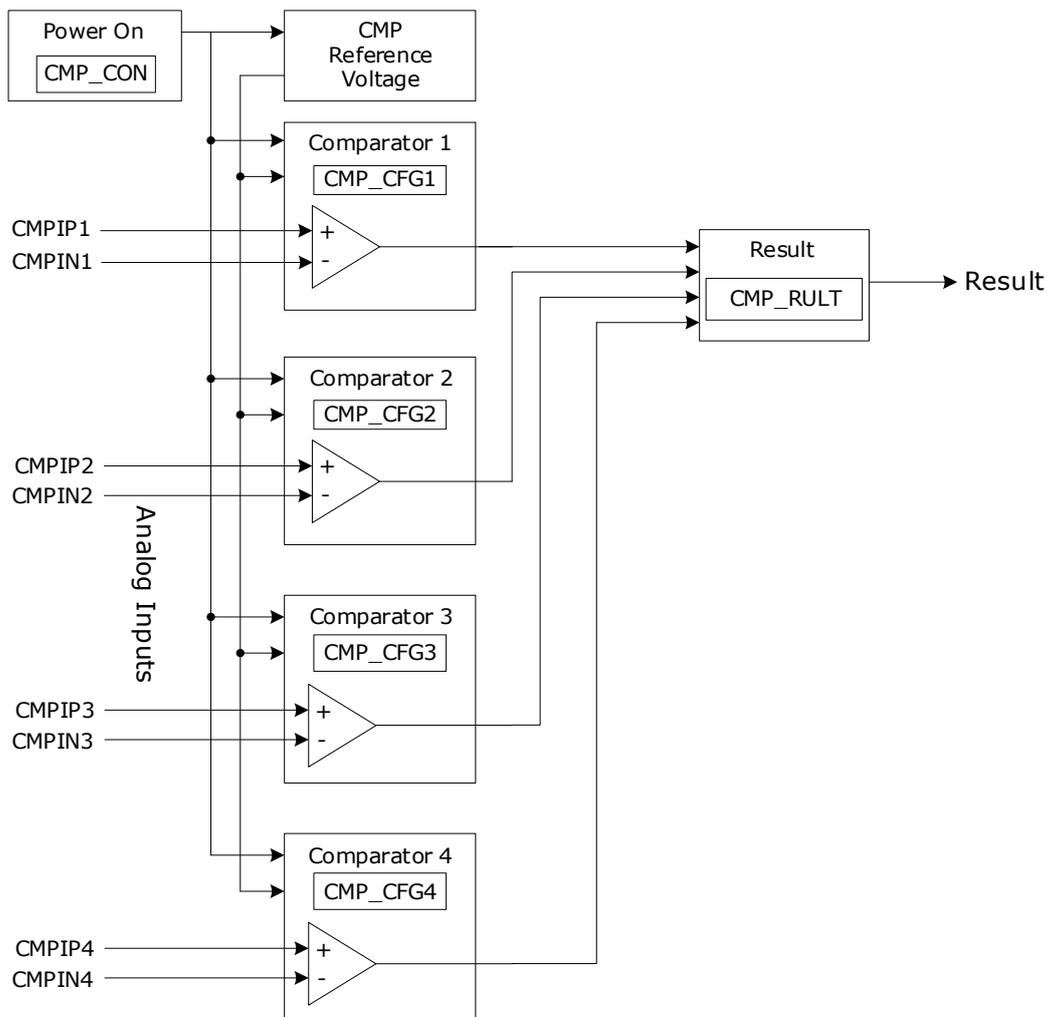


图 13-1 比较器结构图

13.4 功能描述

比较器比较 CMPIP 和 CMPIN 输入电压源以产生输出。下图显示了比较器单元的结构。

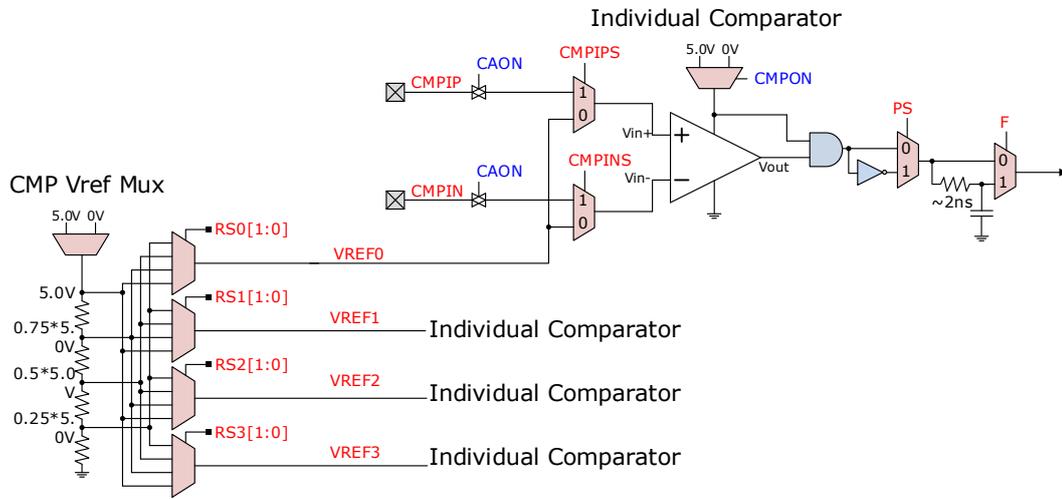


图 13-2 比较器框图

13.5 特殊功能寄存器

13.5.1 寄存器列表

CMP 寄存器列表			
名称	偏移地址	类型	描述
CMP_CON	0000 _H	R/W	控制寄存器
CMP_CFG1	0004 _H	R/W	配置寄存器 1
CMP_CFG2	0008 _H	R/W	配置寄存器 2
CMP_CFG3	000C _H	R/W	配置寄存器 3
CMP_CFG4	0010 _H	R/W	配置寄存器 4
CMP_RULT	0014 _H	R	结果寄存器

13.5.2.3 配置寄存器 2 (CMP_CFG2)

配置寄存器 2 (CMP_CFG2)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000011 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								RS<1:0>		FEN	PSEN	INSEL	IPSEL		

—	Bits 31-6	—	—
RS	Bits 5-4	R/W	比较器2参考电压选择 00: 0.25 * 5.0V 01: 0.5 * 5.0V 10: 0.75 * 5.0V 11: 5.0V
FEN	Bit 3	R/W	滤波电路启用 设置为1有效, 0是关闭
PSEN	Bit 2	R/W	反向电路使能 设置为1有效, 0是关闭
INSEL	Bit 1	R/W	比较器负输入选择 0: 参考电压 1: 输入电压
IPSEL	Bit 0	R/W	比较器正输入选择 0: 参考电压 1: 输入电压

13.5.2.5 配置寄存器 4 (CMP_CFG4)

配置寄存器 4 (CMP_CFG4)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000011 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								RS<1:0>		FEN	PSEN	INSEL	IPSEL		

—	Bits 31-6	—	—
RS	Bits 5-4	R/W	比较器4参考电压选择 00: 0.25 * 5.0V 01: 0.5 * 5.0V 10: 0.75 * 5.0V 11: 5.0V
FEN	Bit 3	R/W	滤波电路启用 设置为1有效, 0是关闭
PSEN	Bit 2	R/W	反向电路使能 设置为1有效, 0是关闭
INSEL	Bit 1	R/W	比较器负输入选择 0: 参考电压 1: 输入电压
IPSEL	Bit 0	R/W	比较器正输入选择 0: 参考电压 1: 输入电压

第14章 数模转换器 (DAC)

14.1 概述

数字 / 模拟转换模块 (DAC) 是 12 位电压输出的数字 / 模拟转换器，也可以与 DMA 控制器配合使用。DAC 有一个输出信道，信道有一个自己的转换器。

14.2 特性

DAC 模块提供以下特性:

- ◆ 12 位电压输出的数字 / 模拟转换器
- ◆ 支持外部，RTC，定时器，比较器和软件触发转换
- ◆ 支持高达 200 kHz 的转换
- ◆ 有 DMA 功能
- ◆ 提供锯齿波和三角波的产生
- ◆ 提供数据输出反转功能

14.3 结构图

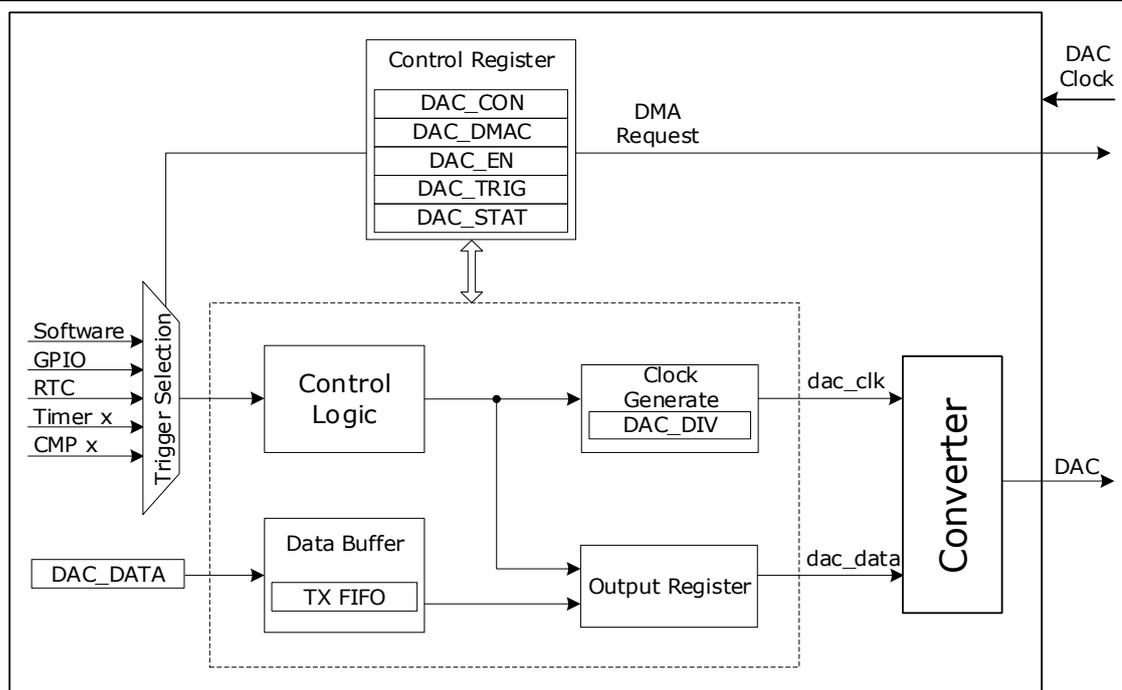


图 14-1 DAC 结构框图

14.4 功能描述

14.4.1 输出电压

经过线性转换后，数字输入会转换为 0.83V 到 4.13V 之间的输出电压，每个 DAC 通道引脚的模拟输出电压由以下方程决定：

$$DAC_{out} = [(5 * DATA) / 4095] + 0.83 (V)$$

14.4.2 触发选择

如果设置了 **DAC_CON.TRIEN** 控制位，则转换可以由外部事件（定时计数器、RTC 和比较器）触发。**DAC_CON.TRISEL [4:0]**控制位确定 17 个可能事件中的哪一个将触发转换。

每次 DAC 接口检测到所选触发器上的上升沿，就传输存储在 **DAC_DATA** 寄存器中的数据。在触发发生后，接口 **DAC_DATA** 数据被更新一个 APB 周期。

注：**DAC_CON** 寄存器在设置 **DAC_EN** 寄存器时不能被更改。

14.4.3 数据转换

DAC 在数据传输中集成了数据缓冲器。当没有设置 **DAC_CON.TRIEN** 时，存储在 **DAC_DATA** 寄存器中的数据在 1 pclk 周期后自动传输到数据输出逻辑。此时也可以写入寄存器 **DAC_DATA**，新数据将存储在数据缓冲区中，并在完成传输了先前的数据之后传输到数据输出逻辑。

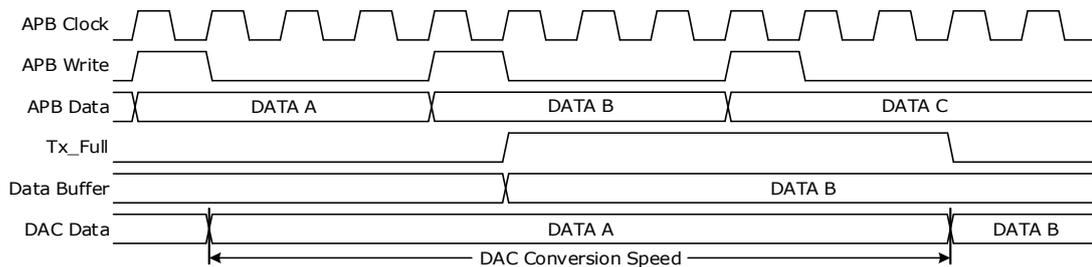


图 14-2 数据传输的时序图，当 **DAC_CTRL.TRIEN = 0**

如果设置了 **DAC_CON.TRIEN** 位，则当数据写入 **DAC_DATA** 寄存器时，数据将保存在数据缓冲区中并等待触发器发生，数据传输将在 1 pclk 周期之后开始。此时，对 **DAC_DATA** 数据寄存器的任何写入操作都是无用的。

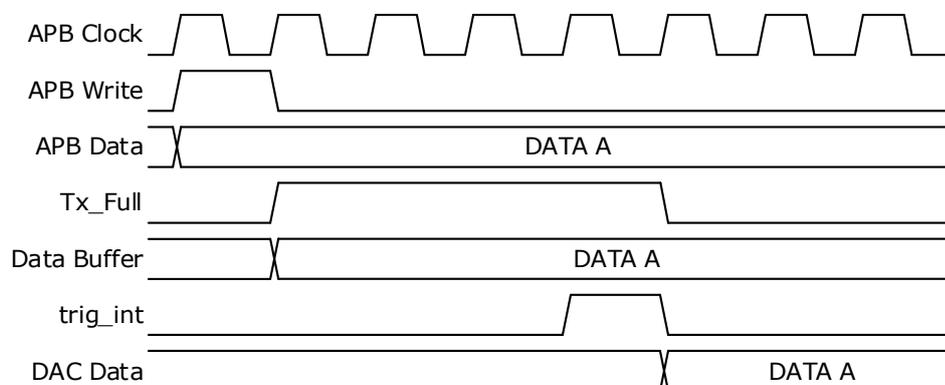


图 14-3 数据传输的时序图，当 **DAC_CTRL.TRIEN = 1**

14.4.4 波形生成

DAC 由 **DAC_CON.WAVE [1:0]** 配置，可以产生两种波形。

- ◆ 锯齿波
- ◆ 三角波

可以在直流或慢变信号上增加小振幅三角波形。振幅通过 **DAC_CON** 寄存器中的 **MAMP [3:0]** 位来配置。内部计数器递增、递减或先递增，取决于 **DAC_CON.WAVE [1:0]** 位的设置。然后将该计数器的值加到 **DAC_DATA** 中而不溢出，并将其存储到数据缓冲器中。

在向上计数模式中，只要计数器小于 **DAC_CON.MAMP [3:0]** 位所定义的最大幅度，计数器就递增。一旦达到配置的幅度，计数器返回到 0，然后再次递增，以此类推。

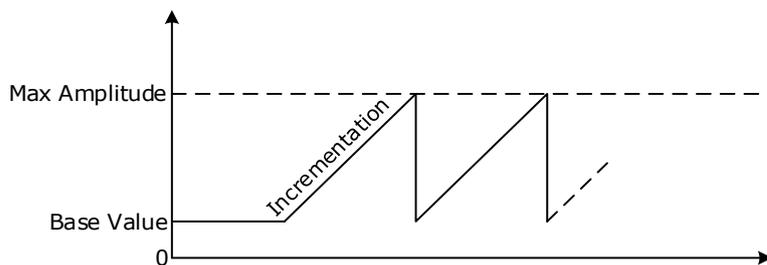


图 14-4 DAC 产生锯齿波 (向上计数模式)

在向下计数模式下，计数器递减到零。一旦达到零，计数器被重新加载到由 **DAC_CON.MAMP [3:0]** 位定义的最大幅度，然后再次减量等等。

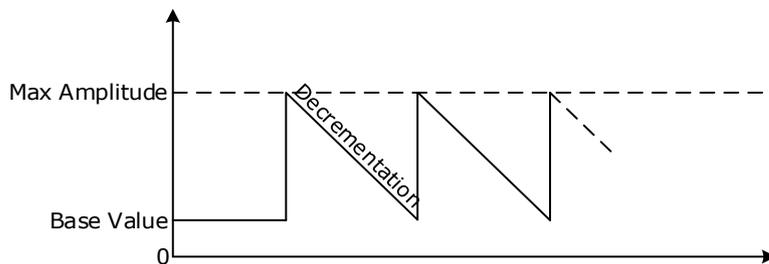


图 14-5 DAC 产生锯齿波 (向下计数模式)

在三角模式下，只要计数器小于 **DAC_CON.MAMP [3:0]**定义的最大振幅，计数器就递增。一旦达到配置的幅度，计数器递减到 0，然后再次递增，以此类推。

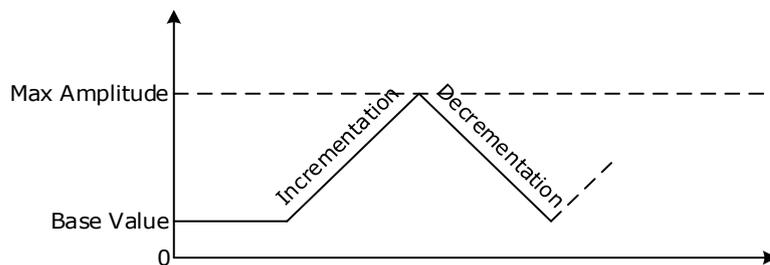


图 14-6 DAC 产生三角波 (上/下计数)

可以通过清除 **DAC_EN** 寄存器来重置三角波的产生。

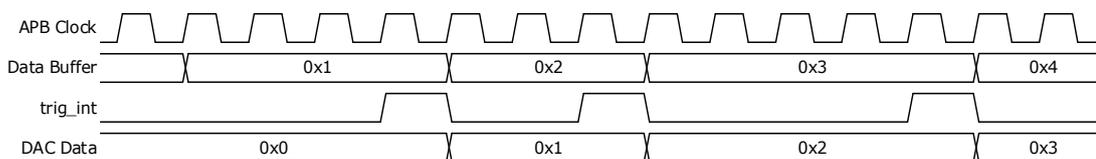


图 14-7 DAC 具有波形生成的 DAC 转换 (触发器启用)

14.4.5 数据输出反转

启用 **DAC_CON.INVREN** 位，则输出数据将被反转。例如，将 **DAC_DATA** 数据写入 0xA、0xB 和 0xFF3，然后输出数据将为 0xFF5、0xFF4 和 0xC。

注：当在波生成模式中，此位没有效果。

14.4.6 DMA 请求

使用 DMA 向 DAC 写入数据，在 **DAC_DMACH** 寄存器中设置 **TXDMAEN** 位。当数据缓冲区为空时，DAC 将请求发送到 DMA，等待 DMA 写入单个数据。

14.5 特殊功能寄存器

14.5.1 寄存器列表

DAC 寄存器列表			
名称	偏移地址	类型	描述
DAC_CON	000 _H	R/W	DAC 控制寄存器
DAC_DIV	004 _H	R/W	DAC 分频寄存器
DAC_STAT	008 _H	R	DAC 状态寄存器
DAC_EN	00C _H	R/W	DAC 使能寄存器
DAC_DMAC	010 _H	R/W	DAC 使能 DMA 寄存器
DAC_TRIG	014 _H	C_W1	DAC 触发寄存器
DAC_DATA	018 _H	R/W	DAC 数据寄存器

14.5.2 寄存器描述

14.5.2.1 DAC 控制寄存器 (DAC_CON)

DAC 控制寄存器 (DAC_CON)																															
偏移地址: 00H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																		T_SEL<4:0>				TRIEN			MAMP<3:0>			WAVE<1:0>		INVREN	

—	Bits 31-14	—	—
T_SEL	Bits 13-9	R/W	触发选择 Value Event 0x0 SW 控制器 (default) 0x1 BS16T1 0x2 Reserved 0x3 GP32C4T1 0x4 GP16C4T1 0x5 GP16C4T2 0x6 GP16C4T3 0x7 GP16C2T1 0x8 GP16C2T2 0x9 GP16C2T3 0xA GP16C2T4 0xB AD16C4T1 0xC GPIO 0xD RTC 0xE CMP1 0xF CMP2 0x10 CMP3 0x11 CMP4 0x12~0x1F 保留
TRIEN	Bit 8	R/W	触发使能 0: 关闭触发 1: 使能触发
—	Bit 7	—	—
MAMP	Bits 6-3	R/W	屏蔽振幅选择 0000: 不屏蔽 bit0 /振幅等于 1 0001: 不屏蔽 bits[1:0] /振幅等于 3 0010: 不屏蔽 bits[2:0] /振幅等于 7 0011: 不屏蔽 bits[3:0] /振幅等于 15

			0100: 不屏蔽 bits[4:0] /振幅等于 31 0101: 不屏蔽 bits[5:0] /振幅等于 63 0110: 不屏蔽 bits[6:0] /振幅等于 127 0111: 不屏蔽 bits[7:0] /振幅等于 255 1000: 不屏蔽 bits[8:0] /振幅等于 511 1001: 不屏蔽 bits[9:0] /振幅等于 1023 1010: 不屏蔽 bits[10:0] /振幅等于 2047 ≥1011: 不屏蔽 bits[11:0] /振幅等于 4095
WAVE	Bits 2-1	R/W	波形生成使能 00: 不产生波形 01: 生成递增波形 10: 生成递减波形 11: 生成三角波
INVREN	Bit 0	R/W	输出数据反转 0: 无反转 1: 输出数据反转

14.5.2.2 DAC 分频寄存器 (DAC_DIV)

DAC 分频寄存器(DAC_DIV)																																													
偏移地址: 04H																																													
复位值: 00000000_00000000_00000000_00000000b																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																DIV_VALUE<15:0>																													

—	Bits 31-16	—	—
DIV_VALUE	Bits 15-0	R/W	分频值 注: 不允许等于 0 的值

14.5.2.5 DAC 使能 DMA 寄存器 (DAC_DMACH)

DAC 使能 DMA 寄存器(DAC_DMACH)																																
偏移地址: 10H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																																TXDMAEN

—	Bits 31-1	—	—
TXDMAEN	Bit 0	R/W	DAC DMA Tx 使能 0: 禁止 1: 使能

14.5.2.6 DAC 触发寄存器(DAC_TRIG)

DAC 触发寄存器 (DAC_TRIG)																																
偏移地址: 14H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																																S_TRIG

—	Bits 31-1	—	—
S_TRIG	Bit 0	T_W1	软件触发

14.5.2.7 DAC 数据寄存器(DAC_DATA)

DAC 数据寄存器(DAC_DATA)																																
偏移地址: 18H																																
复位值: 00000000_00000000_00000111_11111111b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																																TXDATA<11:0>

—	Bits 31-12	—	—
TXDATA	Bits 11-0	R/W	数据寄存器

第15章 通用定时器 16 位 2 通道 (GP16C2T)

15.1 概述

通用定时器 16 位 2 通道 (GP16C2Tn) 包含一个 16 位自动重载计数器，该计数器由可配置的预分频器驱动。

通用定时器 16 位 2 通道 (GP16C2Tn) 的用途广泛，可测量信号脉冲长度（输入捕获）或输出脉冲波形（比较输出、PWM 及带死区时间插入的互补 PWM）。

15.2 特性

- ◆ 16 位递增自动加载计数器
- ◆ 16 位可编程预分频器，可在定时器运行中对计数器工作时钟进行 1 到 65536 间的任意分频
- ◆ 带有两个独立信道，每个信道支持以下功能
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ PWM 输出
 - ◇ 单脉冲输出
- ◆ 通道 1 支持互补输出，死区时间可配
- ◆ 同步电路用于外部信号控制定时器及内部互联多个定时器
- ◆ 在给定数目的计数周期之后更新重复计数寄存器
- ◆ 支持刹车功能，刹车后定时器输出状态可控
- ◆ 支持中断/DMA：
 - ◇ 更新事件：计数器上溢，计数器初始化（通过软件或内/外部触发）
 - ◇ 触发事件（计数器起始、停止、初始化或内/外触发计数）
 - ◇ 通信事件
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ 刹车输入

15.3 结构图

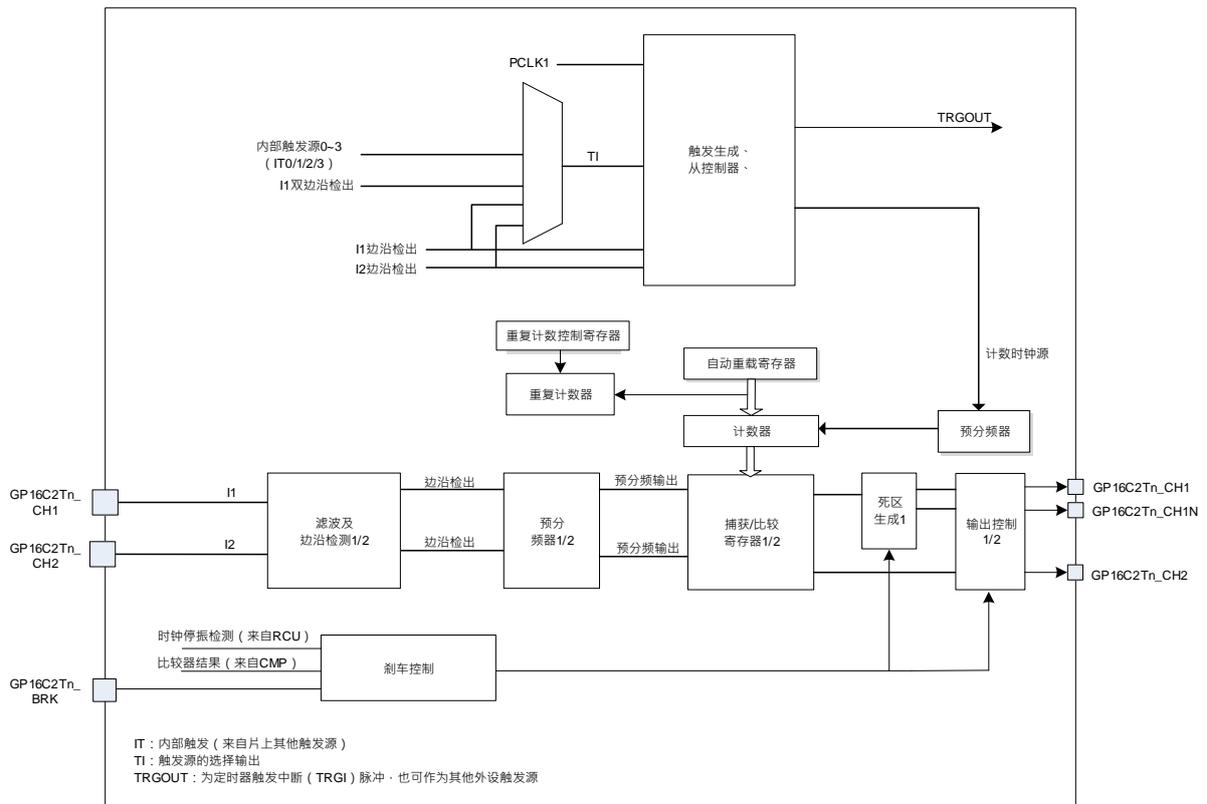


图 15-1 GP16C2Tn 结构框图

15.4 功能描述

15.4.1 预分频器

定时器包含一个 16-bit 的计数器（GP16C2Tn_COUNT），计数时钟由预分频寄存器（GP16C2Tn_PRES）进行分频。计数周期由自动重载计数器（GP16C2Tn_AR）设定。重复计数寄存器则可指定计数周期数目（GP16C2Tn_REPAR）。

自动重载寄存器（GP16C2Tn_AR）是一个可缓存的寄存器。当 GP16C2Tn_CON1 寄存器的 ARPEN 位复位时，GP16C2Tn_AR 寄存器重载功能失效，GP16C2Tn_AR 就是有效寄存器；ARPEN 置位时，GP16C2Tn_AR 寄存器具有重载功能，产生更新事件（UEV）时，加载值（GP16C2Tn_AR 寄存器值）更新到影子寄存器。

当 GP16C2Tn_CON1 寄存器中 DISUE 位为 0 时，计数器计数上溢时会产生更新事件（UEV）。同样，软件方式也可产生更新事件。GP16C2Tn_CON1 寄存器的 CNTEN 置位时，计数器开始计数。

注：计数器在 CNTEN 位置位 1 个时钟周期后开始计数。

预分频器可对定时器工作时钟进行 GP16C2Tn_PRES 寄存器值+1 次分频。由于 GP16C2Tn_PRES 是一个可重载寄存器，因此，定时器工作时可对该寄存器进行修改，修改值在下次更新事件（UEV）后有效。

下图给出了定时器运行过程中改变预分频值时计数器的计数情况。

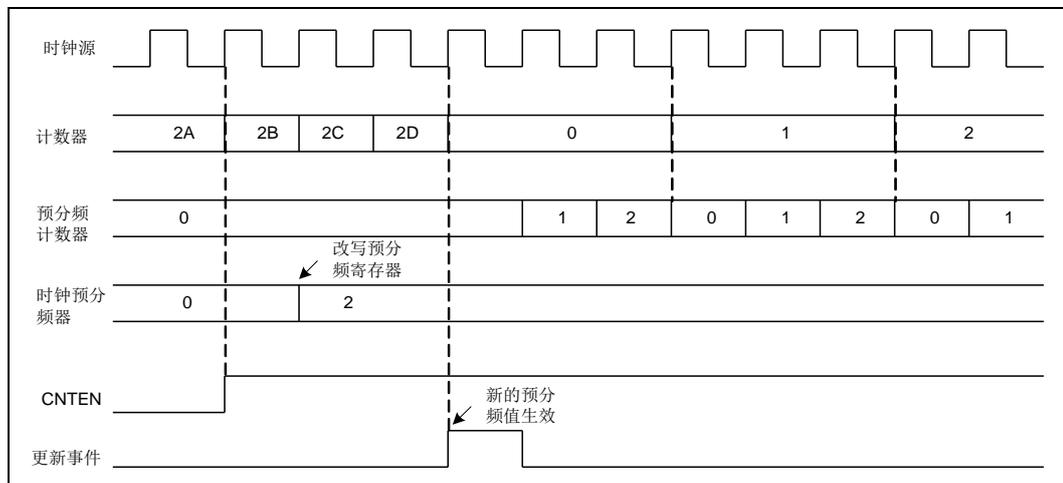


图 15-2 预分频值计数时序图

15.4.2 重复计数器

重复计数器用于控制发生多少次上溢出后产生更新事件。

重复计数器递减：

◇ 递增模式的每次上溢

GP16C2Tn_REPAR 寄存器是一个可缓存寄存器。软件（置位 **GP16C2Tn_SGE** 寄存器中的 **SGU** 位）或硬件从机模式控制方式产生更新事件时，无论重复计数器为何值，**GP16C2Tn_REPAR** 寄存器中值会立即更新到重复计数器的影子寄存器中。

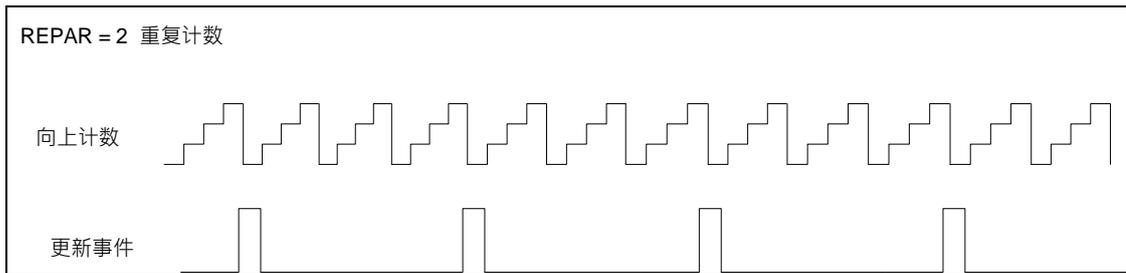


图 15-3 重复计数器工作模式

注意：置位 **GP16C2Tn_SGE** 寄存器中的 **SGU** 位也可以产生更新事件。

15.4.3 时钟源

计数器工作时钟可以选择内部时钟(**INT_CLK**)、外部时钟源 1 (**I1**、**I2**)，内部触发输入 (**IT0**、**IT1**、**IT2**、**IT3**)

15.4.3.1 内部时钟源 (**INT_CLK**)

若从模式控制器被关闭 (**GP16C2Tn_SMCON** 寄存器内，**SMODS**= "000")，则 **CNTEN**，**GP16C2Tn_SGE.SGU** 位为实际控制位，这些位只能软件修改 (**SGU** 位除外，仍硬件自动清除)。一旦 **CNTEN** 位被写为'1'，预分频器就由内部 **INT_CLK** 提供时钟。

下图给出了通常模式下控制电路和递增计数的情况，没有分频。

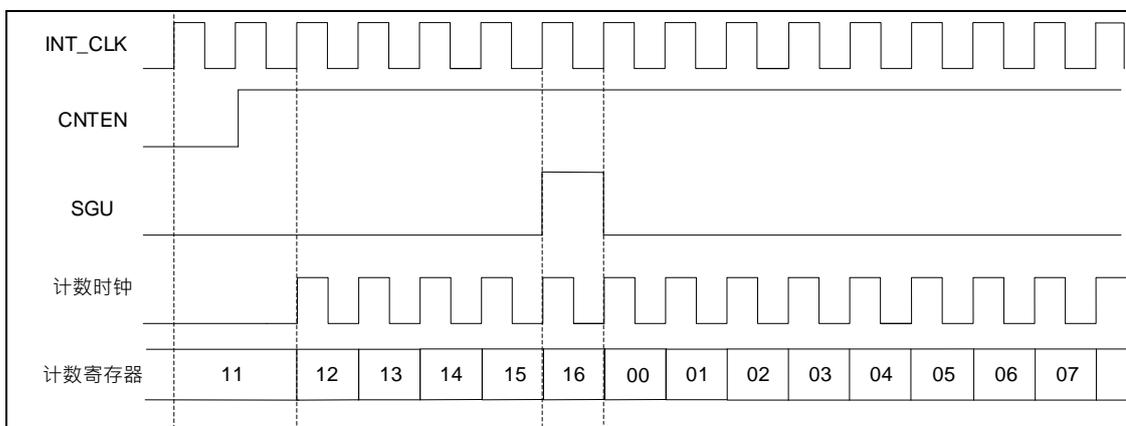


图 15-4 采用内部时钟计数

15.4.3.2 外部时钟源 1

GP16C2Tn_SMCON 寄存器的 **SMODS** = "111" 时, 可选择外部时钟源 1。计数器可根据选定的上升沿或下降沿计数。

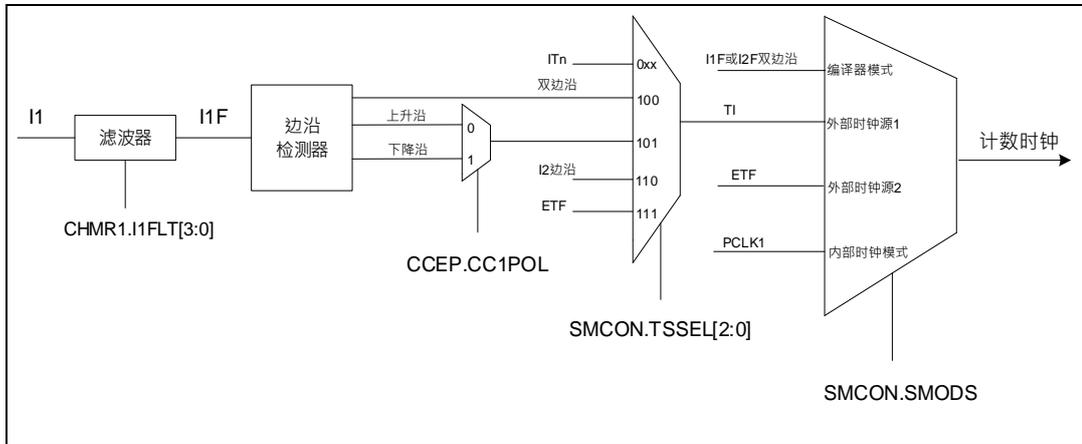


图 15-5 外部时钟连接

配置计数器为外部时钟源 1，步骤如下：

1. **GP16C2Tn_SMCON** 寄存器中 **SMODS** = "111"，配置定时器外部时钟模式 1。
2. 设置 **GP16C2Tn_SMCON** 寄存器中的 **TSSEL** 选择外部时钟源。
3. 如外部时钟源为 I1，可配置 **GP16C2Tn_CHMR1** 寄存器 **CC1SSEL** = "01"，配置信道 1 检测 I1 输入的上升沿；设置 **GP16C2Tn_CCEP** 寄存器中 **CC1POL** = '0'，选择极性为上升沿。
4. 写 **GP16C2Tn_CHMR1** 寄存器的 **I1FLT[3: 0]** 位，配置输入滤波器时间（若没有滤波器需求，维持 **I1FLT** = "0000"）。
5. **GP16C2Tn_CON1** 寄存器中 **CNTEN** = '1'，使能计数器。

当 I1 上出现一次上升沿时，计数器计数一次且 **TRGI** 标志位置位。

15.4.3.3 内部触发输入 (ITn)

当 **GP16C2Tn_SMCON** 寄存器的 **SMODS** = "111", 选定内部触发模式。计数器根据选定的内部输入端的上升或下降沿计数。

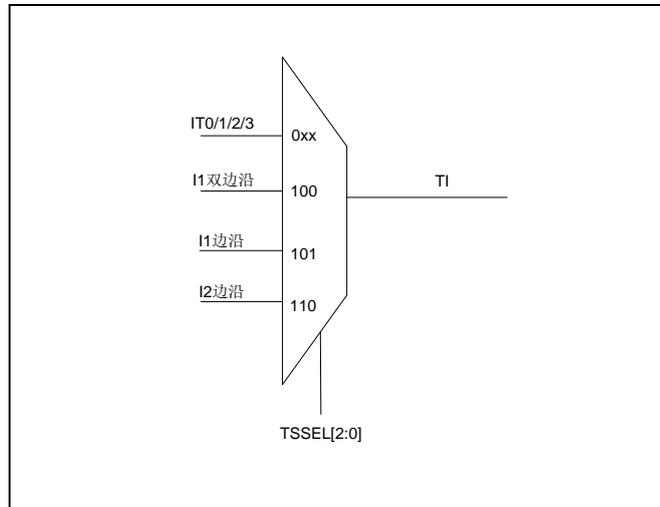


图 15-6 ITn 外部时钟连接

配置计数器在 ITn 输入端的上升沿递增计数，步骤如下：

1. **GP16C2Tn_SMCON** 寄存器中 **SMODS** = "111", 配置外部时钟模式 1。
2. **GP16C2Tn_SMCON** 寄存器的 **TSSEL** = "0xx", 选定 ITn 作为触发输入源。
3. **GP16C2Tn_CON1** 寄存器的 **CNTEN** = '1', 使能计数器。

ITn 产生上升沿时，计数器计数一次。ITn 上升沿与实际时钟间的延时，取决于 ITn 输入的再同步电路。

15.4.4 计数模式

15.4.4.1 递增计数模式

定时器配置为递增模式，计数器从 0 开始递增，直至 **GP16C2Tn_AR** 寄存器值；然后从 0 重新开始计数并产生一个更新事件 (UEV)。当 **GP16C2Tn_REPAR** 寄存器不为 0 时，则在 **GP16C2Tn_REPAR+1** 次计数后产生更新事件。

当有更新事件 (UEV) 产生时，预装载寄存器会更新到影子寄存器，更新标志位 (**GP16C2Tn_RIF** 寄存器中的 **UI** 位) 置位 (取决于 **UERSEL** 位)：

- ◇ 更新 **GP16C2Tn_REPAR** 寄存器的值到影子寄存器
- ◇ 更新 **GP16C2Tn_AR** 寄存器的值到影子寄存器
- ◇ 更新 **GP16C2Tn_PRES** 寄存器的值到影子寄存器

下图为 **GP16C2Tn_AR** = 0x16，预分频设为 2 分频时的计数器时序。

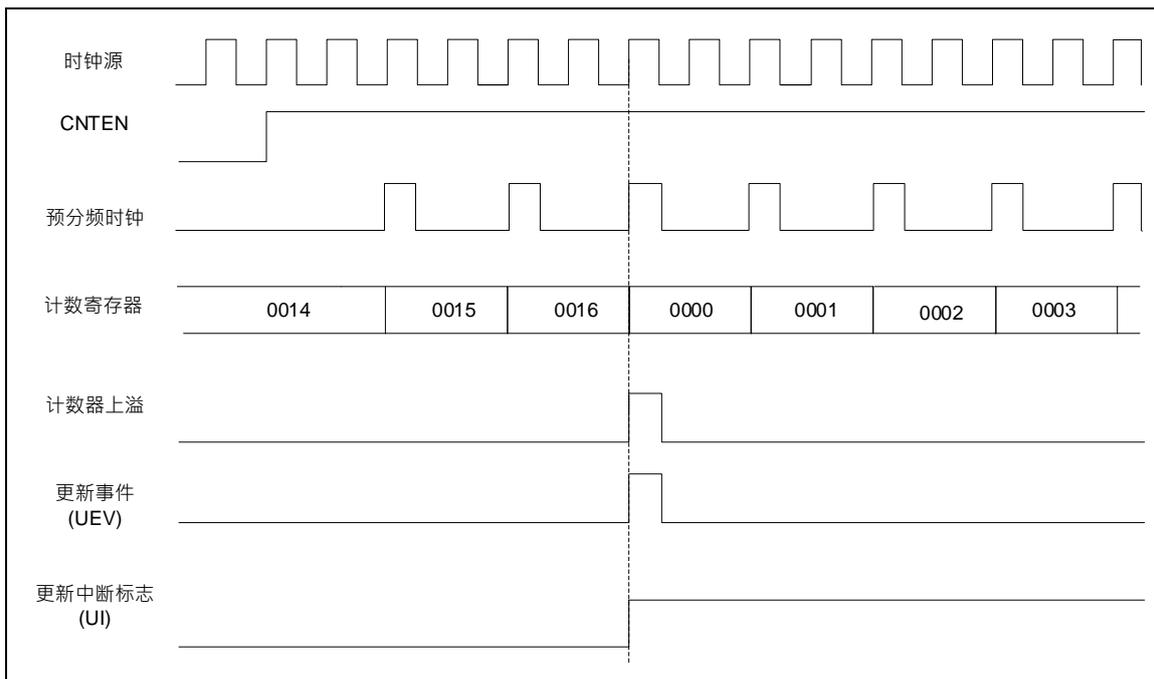


图 15-7 计数器递增计数时序图

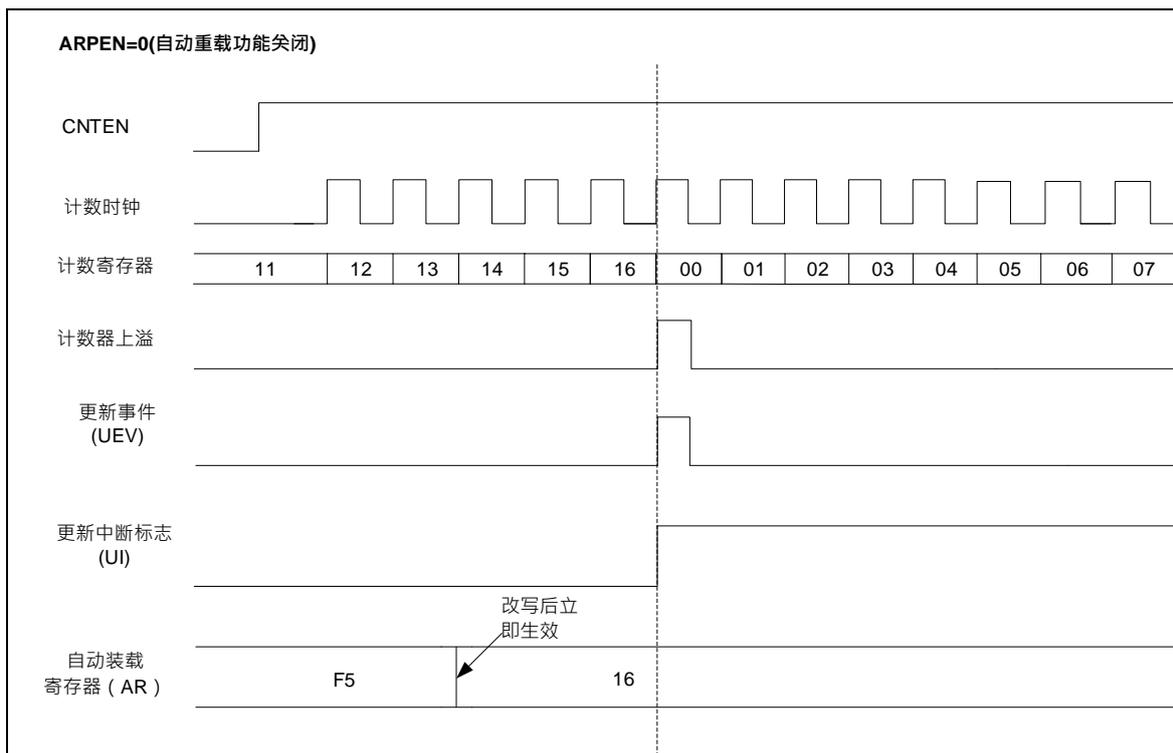


图 15-8 当 ARPEN=0 时计数器时序图

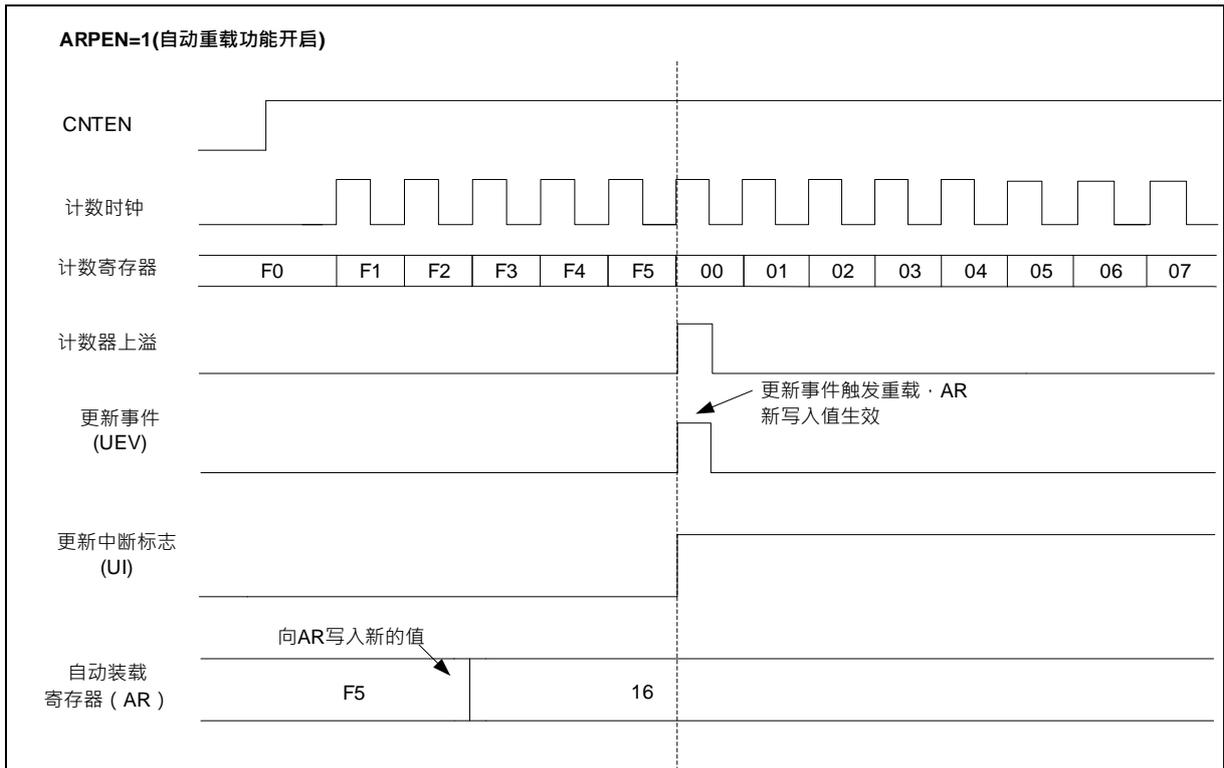


图 15-9 当 ARPEN=1 时计数器时序图

15.4.5 捕获/比较通道

输入电路对 In 输入端的信号进行采样，产生一个经过滤波的信号 InF。之后，一个可极性选择的边沿检测器产生 In 边沿检测信号，该信号可作为从模式控制器的触发输入或作为捕获控制命令，且信号经过分频后进入捕获寄存器。

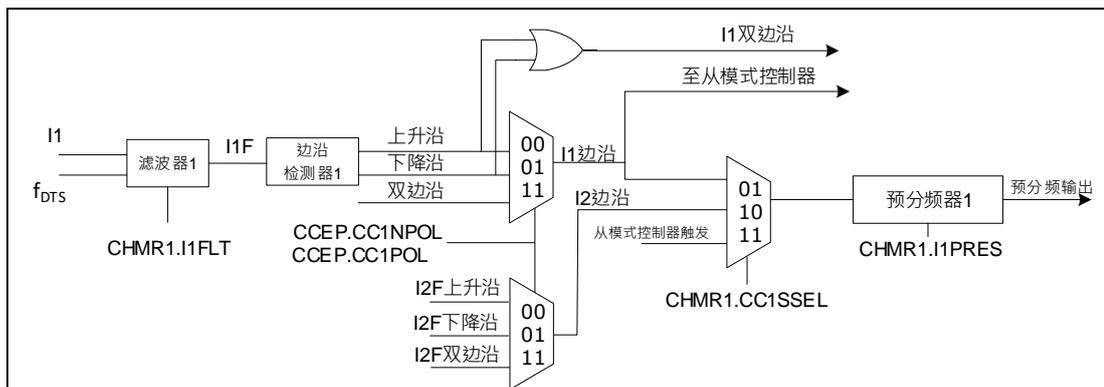


图 15-10 捕获/比较通道

输出部分产生一个中间波形（高有效）作为基准，在输出末端决定最终输出信号的极性。

15.4.6 输入捕获模式

在输入捕获模式下，当检测到 In 上相应信号变化时，计数器的值就会被锁存到捕获/比较寄存器(GP16C2Tn_CCVALn)寄存器中。当捕获发生时，相应的 CHnI 标志位(GP16C2Tn_RIF)会置位，同时会触发中断或 DMA (如果使能) 请求。若发生捕获时，CHnI 标志位已经置位，则过捕获 CHnOVI 标志位 (GP16C2Tn_RIF) 置位。软件于 GP16C2Tn_ICR 对 CHnI 与 CHnOVI 位写'1'可以复位 CHnI 标志位与 CHnOVI 标志位。

以下为以 I1 输入上升沿作为捕获输入时的流程：

1. 选择有效输入端：GP16C2Tn_CCVAL1 必须连接到 I1 输入端，因此需将 GP16C2Tn_CHMR1 寄存器中的 CC1SSEL 位写"01"。只要 CC1SSEL 不为"00"，信道被配置为输入且 GP16C2Tn_CCVAL1 寄存器为只读。
2. 根据定时器连接的输入信号，配置输入滤波器的持续时间。当输入信号翻转时，前 5 个内部时钟信号内信号是不稳定的，因此必须配置滤波器的时间大于 5 个时钟周期。当 I1 检测到新的电平，连续 8 次采样可确认电平变化有效。
3. 选择 I1 信道的有效边沿变换。GP16C2Tn_CCEP 寄存器中的 CC1POL 写'0'(上升沿)。
4. 配置输入预分频器。
5. 置位 GP16C2Tn_CCEP 寄存器中的 CC1EN 位，使能捕获计数器的值到捕获寄存器。
6. 如有需要，置位 GP16C2Tn_IER 寄存器中的 CH1I 位，使能中断请求。置位 GP16C2Tn_DMAEN 寄存器中的 CH1DE 位，使能 DMA 请求。

当发生输入捕获时：

1. 有效边沿产生，GP16C2Tn_CCVAL1 寄存器获取计数器的值。
2. CH1I 标志位置位 (中断标志)。若至少 2 个连续的捕获发生，但标志位没有及时清除，则 CH1OVI 也会置位。
3. 中断的产生取决于 GP16C2Tn_IER 寄存器中的 CH1I 位。
4. DMA 请求的产生取决于 GP16C2Tn_DMAEN 寄存器中的 CH1DE 位。

为了处理捕获溢出，建议在读取过捕获标志位前先读取捕获数据。这可以避免错过读过捕获标志位之后，读之前产生的捕获数据。

注：捕获中断请求可由软件设置 GP16C2Tn_SGE 寄存器中 SGCHn 位产生。

15.4.6.1 PWM 输入模式

测量 I1 上 PWM 信号的周期和占空比的过程如下：

1. 为 **GP16C2Tn_CCVAL1** 选择有效的输入： **GP16C2Tn_CHMR1** 寄存器中的 **CC1SSEL** 位写"01"（I1 被选择）。
2. 为 I1 边沿检测选择有效的极性（用于捕获数据到 **GP16C2Tn_CCVAL1** 寄存器和计数器清零）：**CC1POL** 位写'0'（上升沿有效）。
3. 为 **GP16C2Tn_CCVAL2** 选择有效输入： **GP16C2Tn_CHMR1** 寄存器的 **CC2SEL** 位写"10"（I1 被选择）。
4. 为 I1 边沿检测选择有效极性(用于捕获数据到 **GP16C2Tn_CCVAL2**):**CC2POL** 位写'1'。
5. 选择有效的触发输入： **GP16C2Tn_SMCON** 寄存器的 **TSSEL** 位写"101"（I1 边沿检测被选择）。
6. 配置从机模式控制器为复位模式： **GP16C2Tn_SMCON** 寄存器的 **SMODS** 位写"100"。
7. 使能捕获： **GP16C2Tn_CCEP** 寄存器的 **CC1EN** 位和 **CC2EN** 位写'1'。

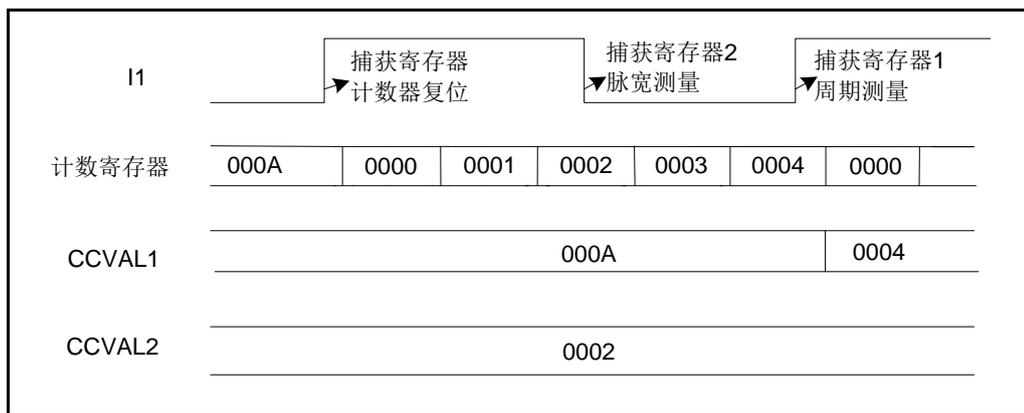


图 15-13 PWM 输入模式时序

15.4.7 PWM 模式

脉宽调制模式可以产生一个 GP16C2Tn_AR 寄存器值确定频率，GP16C2Tn_CCVALn 寄存器值确定占空比的信号。

每个信道的 PWM 模式是相互独立的（每个 CHn 输出一个 PWM），GP16C2Tn_CHMRn 寄存器的 CHnMOD 位写"110"（PWM 模式 1）或写"111"（PWM 模式 2）。必须通过置位 GP16C2Tn_CHMRn 寄存器的 CHnPEN 位来使能相应的预载寄存器，最后还需置位 GP16C2Tn_CON1 寄存器的 ARPEN 位来使能自动重装预载功能。

只有当更新事件发生时预载寄存器中的值才会传到影子寄存器，因此，在使能计数前，必须通过置位 GP16C2Tn_SGE 寄存器的 SGU 位来初始化所有的寄存器。

CHn 的极性可通过 GP16C2Tn_CCEP 寄存器的 CCnPOL 位配置，有效极性可配置为高或低。CHn 的输出使能由 CCnEN、CCnNEN、GOEN、OFFSSI 和 OFFSSR 位（GP16C2Tn_CCEP 和 GP16C2Tn_BDCFG 寄存器）组合控制。

在 PWM 模式（1 或 2）中，GP16C2Tn_COUNT 和 GP16C2Tn_CCVALn 寄存器的值会持续比较，确定 GP16C2Tn_CCVALn <= GP16C2Tn_COUNT 或 GP16C2Tn_CCVALn >= GP16C2Tn_COUNT。

15.4.7.1 PWM 边沿对齐模式

- ◇ GP16C2Tn_AR=8
- ◇ PWM 模式 1
- ◇ 递增计数配置

下图给出了 GP16C2Tn_AR = 8 时的边沿对齐 PWM 波形。

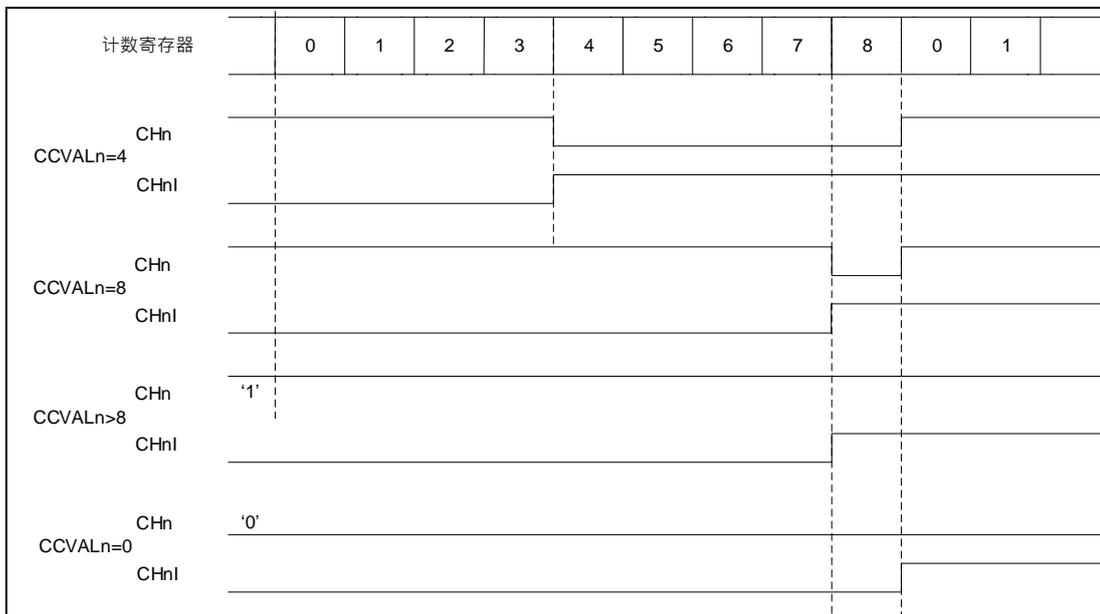


图 15-14 边沿对齐递增计数 PWM 波形 (AR=8)

15.4.8 输出比较模式

该功能用于控制输出波形或指示周期时间的结束。

当捕获/比较寄存器和计数器值匹配时，输出比较功能：

- ◇ 输出比较模式（GP16C2Tn_CHMRn 寄存器中的 CHnMOD 位）和输出极性（GP16C2Tn_CCEP 寄存器中的 CCnPOL 位）的配置值输出到对应的引脚上。
- ◇ 中断状态寄存器中的标志位置位（GP16C2Tn_RIF 寄存器的 CHnI 位）。
- ◇ 若相应的中断使能置位，则产生中断（GP16C2Tn_IER 寄存器的 CHnI 位）。
- ◇ 若相应的使能位置位（GP16C2Tn_DMAEN 寄存器的 CHnDE 位，GP16C2Tn_CON2 寄存器的 CCDMASEL 位用于 DMA 请求的选择），则发送 DMA 请求。

GP16C2Tn_CHMRn 寄存器中 CHnPEN 位的值可决定 GP16C2Tn_CCVALn 寄存器是否带有预装载寄存器。

在输出比较模式中，更新事件 UEV 对 CHn 的输出没有影响。计时分辨率为计数器的一次计数。输出比较模式同样可以用来输出单个脉冲（单脉冲模式）。

输出比较的配置过程：

1. 选定计数器时钟（内部，外部，预分频）。
2. GP16C2Tn_AR 与 GP16C2Tn_CCVALn 寄存器中写入预期值。
3. 若需要产生中断请求，置位 GP16C2Tn_IER 寄存器中的 CHnI 位。
4. 选择输出模式，例如：
 - CHnMOD = "011", 当 CNTV 与 CCRVALn 匹配时，CHn 输出翻转。
 - CHnPEN = '0', 关闭预载寄存器。
 - CCnPOL = '0', 选择有效极性为高。
 - CCnEN = '1', 使能输出。
5. GP16C2Tn_CON1 寄存器中的 CNTEN 位置位，使能计数器。

假设预载寄存器没有使能（CHnPEN = '0'，否则 GP16C2Tn_CCVALn 影子寄存器只有在下次更新事件发生时才更新）。通过软件方式，GP16C2Tn_CCVALn 寄存器的值可随时更新控制输出波形。

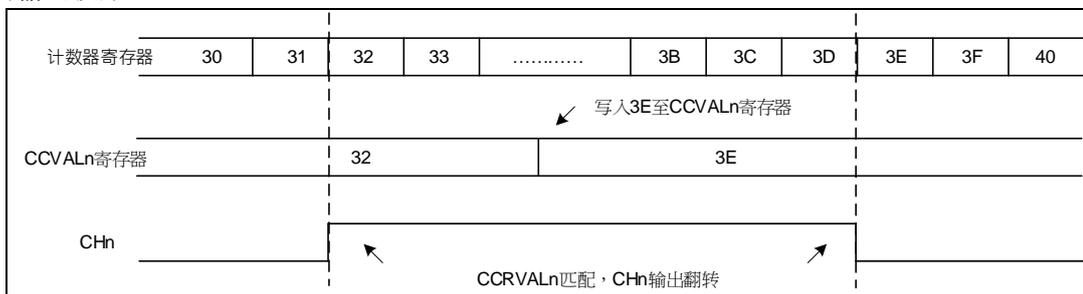


图 15-15 输出比较模式，触发 CHn

15.4.8.1 强制输出模式

在输出模式中（**GP16C2Tn_CHMRn** 寄存器中 $CCnSSEL = "00"$ ），软件可强制将每个输出比较信号（ $CHnN/CHnN$ ）改为有效或无效状态，这种修改独立于输出比较寄存器和计数器的比较结果。

为了将某输出比较信号（ CHn ）强制为有效状态，需将相应的 **GP16C2Tn_CHMRn** 寄存器中 $CHnMOD$ 位写"101"。因此，比较输出被强制为高（高时为有效状态）且 CHn 的值为 $CCnPOL$ 极性位的相反值。

例如： $CCnPOL = '0'$ （ CHn 高电平有效），则 CHn 被强制为高电平。

对 **GP16C2Tn_CHMRn** 寄存器的 $CHnMOD$ 位写"100"，比较输出可被置低。

无论怎样，**GP16C2Tn_CCVALn** 影子寄存器和计数器之间的比较仍然进行，相应的标志位仍可置位。

15.4.9 单脉冲模式

单脉冲模式（**SPMEN**）下，响应某个触发后，定时器的输出信道在可配置的延迟时间后产生一个脉冲，脉冲长度可配。从模式控制器可控制计数器的启动。脉冲波形可在输出比较模式和 **PWM** 模式下产生。置位 **GP16C2Tn_CON1** 寄存器的 **SPMEN** 位可选择单脉冲模式。计数器会在下次更新事件 **UEV** 产生时自动停止。

只有比较值不同于计数器初始值时，单脉冲才可以正确的产生。计数器开始计数前（定时器等待触发），必须如下配置：

- ◇ 递增计数： $CNTV < CCVALn \leq AR$ （特别地， $0 < CCVALn$ ）

基于 **PWM** 模式设置单脉冲输出波形的步骤如下：

- ◇ 设置 **GP16C2Tn_CHMRn** 寄存器的 $CHnMOD$ 位，选择 **PWM** 模式 1 或 2；
- ◇ 设置 **GP16C2Tn_CCEPn** 寄存器的 $CCnPOL$ 位，选择通道端口 CHn 的输出极性；
- ◇ 设置 **GP16C2Tn_CON1** 寄存器的 **SPMEN** 位，配置为递增计数，**PWM** 普通波形模式，单脉冲模式使能；
- ◇ 设置 **GP16C2Tn_CHMR** 寄存器的 $CH1PEN = 1$ ，**GP16C2Tn_CON1** 寄存器的 $ARPEN = 1$ ，使能比较寄存器和计数重载寄存器的缓冲功能（也可以根据实际情况不使能缓冲）；
- ◇ 设置 **GP16C2Tn_CCVALn** 寄存器和 **GP16C2Tn_AR** 寄存器，配置单脉冲输出延时和脉宽时间；
- ◇ 设置 **GP16C2Tn_SGE** 寄存器的 **SGU** 位来产生一个更新事件；
- ◇ 设置 **GP16C2Tn_CON1** 寄存器的 $CNTEN = 1$ 来启动计数器，也可以在触发模式下，通过外部触发输入信号来触发硬件自动设置 $CNTEN = 1$ 。

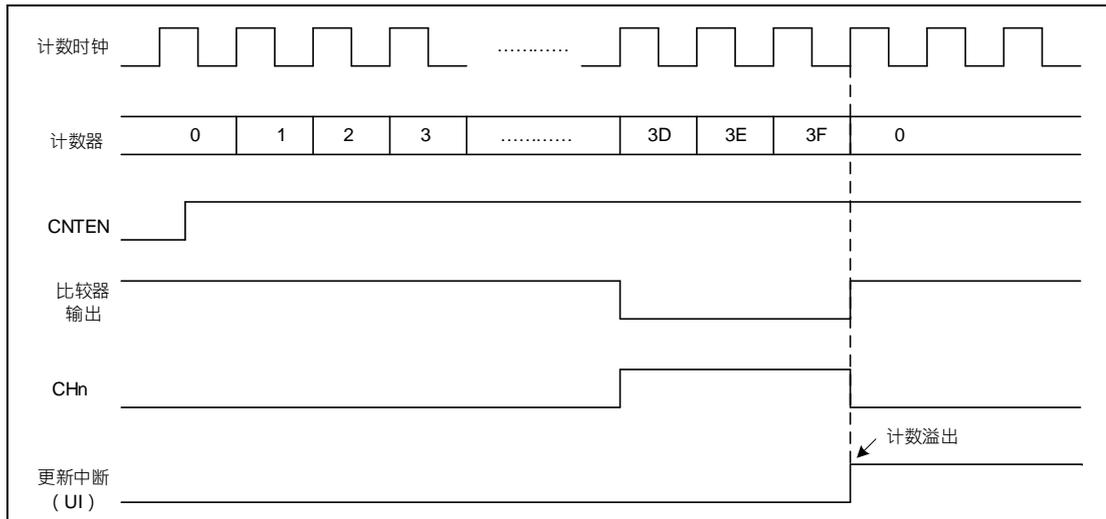


图 15-16 单脉冲模式

15.4.10 互补输出与死区时间

两个互补的通道输出信号，可以用来控制输出的瞬时开关，死区时间可配置。

每个输出可独立选择输出极性（主输出 CHn 或互补输出 CHnN），该操作可通过写 **GP16C2Tn_CCEP** 寄存器的 CCnPOL 和 CCnNPOL 位完成。

互补信号 CHn 和 CHnN 由几个控制位共同控制，分别是 **GP16C2Tn_CCEP** 寄存器中的 CCnEN 和 CCnNEN 位，**GP16C2Tn_BDCFG** 和 **GP16C2Tn_CON2** 寄存器中的 GOEN、OISSn、OISSnN、OFFSSI 及 OFFSSR 位，特别是死区时间使能后的空闲状态的切换（GOEN 变为 0）。

置位 CCnEN 和 CCnNEN 位，使能死区时间插入，若有刹车电路，同样需要置位 GOEN 位。**GP16C2Tn_BDCFG** 寄存器的 DT[7: 0]可以控制所有通道的死区时间的产生。根据比较输出波形，产生 CHn 和 CHnN 两路输出。若 CHn 和 CHnN 有效电平为高：

- ◇ CHn 的输出信号与参考信号一致。上升沿除外，相对参考信号的上升沿，CHn 输出会有延迟。
- ◇ CHnN 的输出信号与参考信号相反。上升沿除外，相对参考信号的下降沿，CHnN 输出会有延迟。

若延迟时间大于有效输出的宽度（CHn 或 CHnN），则相应的脉冲不会产生。

下图给出了死区时间输出信号和比较输出波形之间的关系。假设 CCnPOL = 0, CCnNP = 0, GOEN = 1, CCnEN = 1, 和 CCnNEN = 1

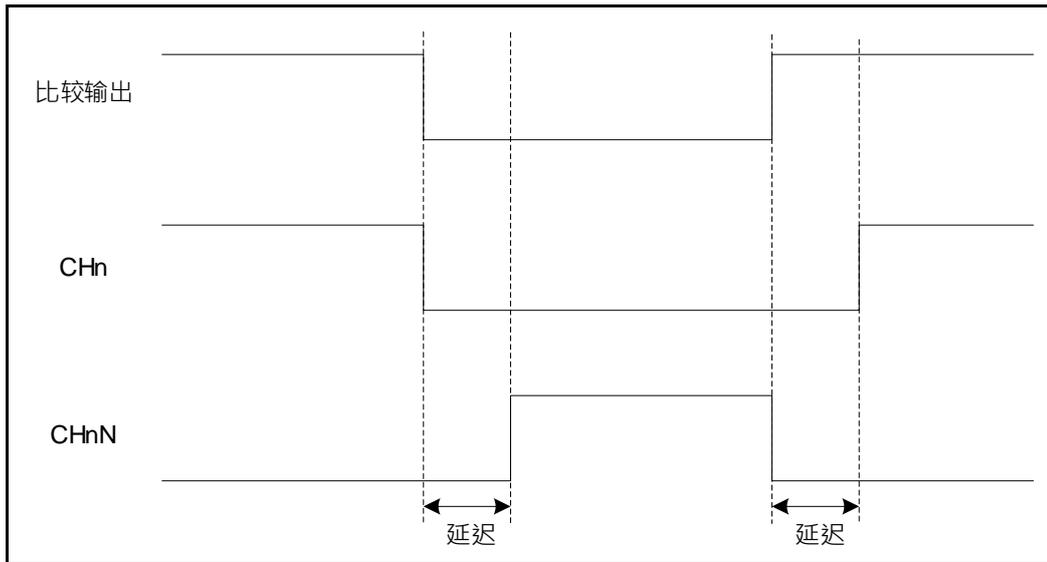


图 15-17 互补输出含死区时间插入

当 PWM 信道配置为互补输出时，如下寄存器控制位都会有缓冲：**CHnMOD**、**CCnEN** 和 **CCnNEN**。发生互补通道更新事件时，这些寄存器位才会真正生效，这样就可以预先设置好下一步的配置，并同时对所有互补信道的配置进行更新。互补通道更新事件可以通过设置 **GP16C2Tn_SGE** 寄存器的 **SGCOM=1** 产生，或由触发信号产生（由 **GP16C2Tn_SMCON** 寄存器的 **TSSEL** 位选择触发信号）。

15.4.11 刹车功能

刹车功能模式由以下几个控制位进行设置：**GP16C2Tn_BDCFG** 寄存器中的 **GOEN**、**OFFSSI** 和 **OFFSSR** 位，**GP16C2Tn_CON2** 寄存器中的 **OISSn** 和 **OISSnN** 位，输出使能信号和无效电平都会被修改。

刹车源可以是刹车输入引脚、时钟失败事件、比较器结果以及软件控制 **GP16C2Tn_SGE** 寄存器的 **SGBRK** 位。时钟失败事件由时钟控制器（**RCU**）中的时钟安全系统（**CSS**）产生。时钟安全系统（**CSS**）详细讯息可参考时钟安全系统章节。

系统复位后，刹车电路被禁止且 **GOEN** 位被复位。置位 **GP16C2Tn_BDCFG** 寄存器的 **BRKEN** 位可使能刹车功能，同样寄存器中，**BRKEN** 位可选择刹车输入信号的极性。**BRKEN** 和 **BRKP** 位可同时修改。对 **BRKEN** 和 **BRKP** 位写操作后，1 个 APB 时钟周期延时后写入值才会生效。因此，写操作后，需等待 1 个 APB 时钟周期后才能正确读回写入值。

由于 **GOEN** 的下降沿可以是异步的，在实际信号（作用在输出端）和同步控制位（**GP16C2Tn_BDCFG** 寄存器中）之间插入了一个同步电路。这也导致了异步和同步信号之间会产生一些延迟。特别是 **GOEN** 之前为低时对 **GOEN** 写 1 操作后，要读取正确值，必须先插入一个延时（空指令）。这是因为写入的是异步信号，而读取的是同步信号。

当发生刹车请求时（刹车输入端有刹车电平）：

- ◇ GOEN 位被异步清除，输出端进入无效状态，空闲状态或复位状态（OFFSSI 位选择）。即使 MCU 的振荡器关闭，该功能仍然有效。
- ◇ 一旦 GOEN=0，每个信道输出预先配置的电平。GP16C2Tn_CON2 寄存器中的 OISSn 位配置该电平。如果 OFFSSI=0，则定时器释放使能输出，否则使能输出一直为高。
- ◇ 当使用互补输出时：
 - 输出端首先被置于复位状态，无效状态（取决于极性）。这个过程是异步的，即使定时器没有时钟也有效。
 - 如果定时器时钟仍然存在，则死区时间生成器会重新生效，这样在死区时间后，OISSn 和 OISSnN 位的配置电平可驱动输出。这种情况下，CHn 和 CHnN 无法驱动输出端都为有效电平。
 - 由于对 GOEN 的重新同步，死区时间的周期会比通常情况下长一些（大约 2 个 TIMER 模块时钟周期）。
 - 如果 OFFSSI = 0，则定时器释放使能输出，否则使能输出保持或变高（一旦 CCnEN 和 CCnNEN 有一个变高时）。
- ◇ 当刹车状态标志位（GP16C2Tn_RIF 寄存器中的 BRKI 位）置位时，若 GP16C2Tn_IER 寄存器中的 BRKI 位置位，可触发中断。
- ◇ 当 GP16C2Tn_BDCFG 寄存器中的 AOEN 位置位时，在下次更新事件(UEV)发生时，GOEN 位会自动置位。例如，该功能可用来整形。否则，GOEN 位会保持为低，直到对其写'1'操作，该特性可用于安全方面的应用，可以将刹车输入端接到一个电源驱动的报警端、热敏传感器或其他安全器件上。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能置位（自动地或者通过软件）GOEN。同时，状态标志 BRKI 不能被清除。

除刹车输入和输出管理，为保证应用程序的安全，内部刹车电路具有写保护功能。用户可冻结几个配置参数（死区时间，CHn/CHnN 极性和失能时状态，CHnMOD 配置，刹车使能和极性）。通过 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位，可从三个保护等级中选择一种保护等级。MCU 复位后，LOCKLVL 位只能写一次。

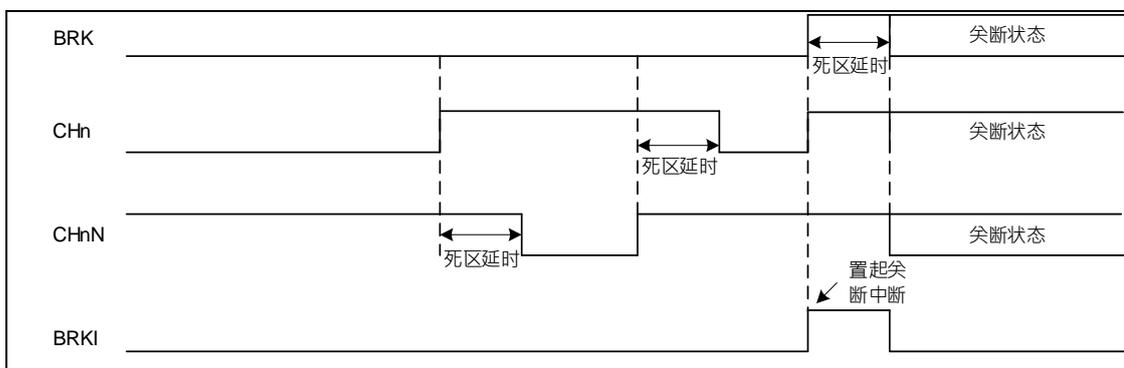


图 15-18 刹车输出行为

15.4.12 外部触发的同步

GP16C2Tn 定时器可在多种模式下与外部触发同步：复位模式、门控模式及触发模式。

15.4.12.1 复位模式

计数器及其预分频器可以在响应触发输入事件时重新初始化。此外，若 **GP16C2Tn_CON1** 寄存器的 **UERSEL** 位为低时会产生一次更新事件 **UEV**。所有预载寄存器 (**GP16C2Tn_AR**, **GP16C2Tn_CCVALn**) 都会因更新事件 **UEV** 而被更新。

在下面例子中，I1 输入端的上升沿让递增计数被清空：

- ◇ 配置信道 1 上检测 I1 上的上升沿。配置输入滤波周期（本例无需滤波器，故 **I1FLT** = "0000"）。触发捕获分频器没有使用，无需配置。**CC1SSEL** 位只选择输入捕获源，**GP16C2Tn_CHMR1** 寄存器中 **CC1SSEL** = "01"。**GP16C2Tn_CCEP** 寄存器中 **CC1POL** = 0 以确定极性（只检测上升沿）。
- ◇ 定时器配置为复位模式：**GP16C2Tn_SMCON** 寄存器中 **SMODS** = "100"。选择 I1 作为输入源：**GP16C2Tn_SMCON** 寄存器中 **TSSEL** = "101"。
- ◇ 启动计数器：**GP16C2Tn_CON1** 寄存器中 **CNTEN** = '1'。

计数器依据内部时钟开始计数，正常计数直到 I1 上出现上升沿。当 I1 上出现上升沿时，计数器会被清零且从 0 重新开始计数。同时，标志位置位（**GP16C2Tn_RIF** 寄存器中 **TRGI** 位），如果中断及 DMA 使能（取决于 **GP16C2Tn_IER** 寄存器中的 **TRGI** 和 **GP16C2Tn_DMAEN** 寄存器中的 **TRGIDE** 位），会发送中断及 DMA 请求。

下图给出了当自动重载寄存器 **GP16C2Tn_AR** = 0x36 时的信号变化。由于 I1 输入的再同步电路，I1 上的上升沿和计数器实际复位之间会存在延时。

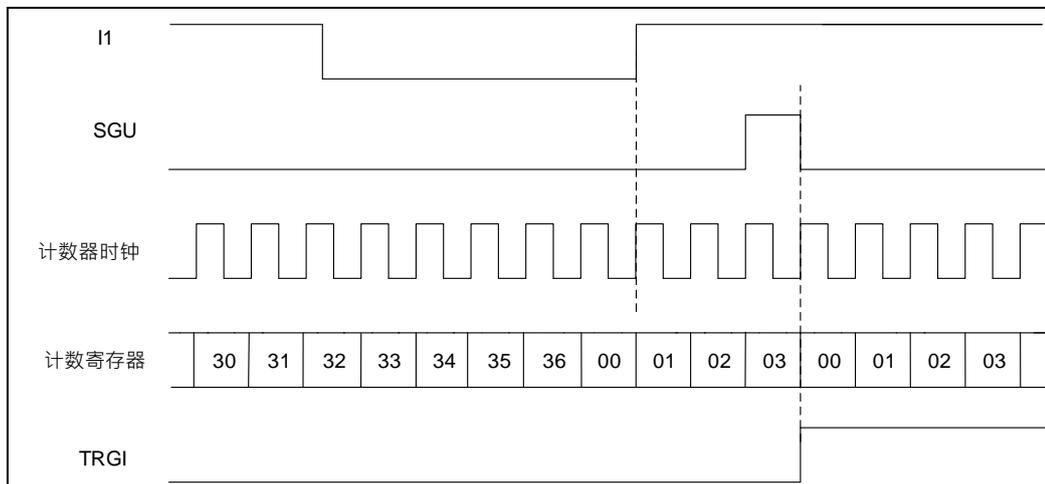


图 15-19 复位模式控制电路

15.4.12.2 门控模式

计数器根据选中的输入电平被使能。

下面的例子中，计数器只在 I1 输入为低电平时才递增计数：

- ◇ 配置信道 1 在 I1 上检测低电平。配置输入滤波周期（本例不需要滤波器，I1FLT = "0000"）。触发捕获分频器没有使用，无需配置。**GP16C2Tn_CHMR1** 寄存器中的 CC1SSEL = "01"，选择输入捕获源。**GP16C2Tn_CCEP** 寄存器中 CC1POL = '1'，确认极性（只检测低电平）。
- ◇ 配置定时器为门控模式：**GP16C2Tn_SMCON** 寄存器中 SMODS = "101"。选择 I1 作为输入源：**GP16C2Tn_SMCON** 寄存器中 TSSEL = "101"。
- ◇ 使能计数器：**GP16C2Tn_CON1** 寄存器中 CNTEN = '1'（门控模式中，如果 CNTEN = '0'，无论触发输入为何电平，计数器都不会启动）。

只要 I1 为低电平，计数器依据内部时钟开始计数，一旦 I1 为高则停止计数。由于 I1 输入端的再同步电路的原因，I1 上出现上升沿和计数器实际停止之间会有一定的延时。

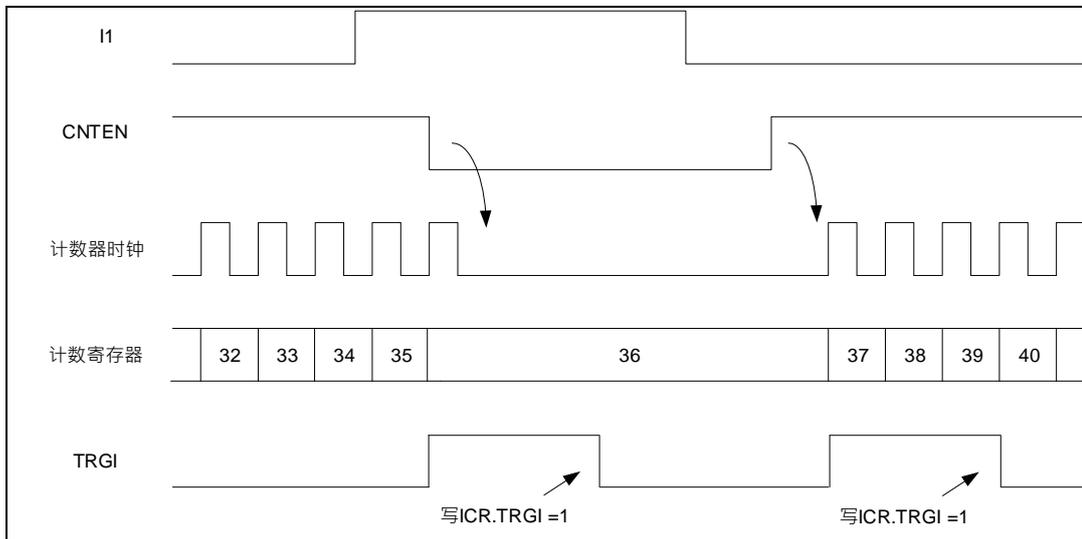


图 15-20 门控模式控制电路

15.4.12.3 触发模式

输入端选中的事件可以使能计数器。

下面的例子中，I2 输入端上的上升沿可以启动递增计数：

- ◇ 配置信道 2 可以检测 I2 上的上升沿。配置滤波时间（本例不需要滤波，I2FLT = "0000"）。触发捕获分频器没有使用，无需配置。GP16C2Tn_CHMR1 寄存器中 CC2SEL = "01", 用于选择捕获源。GP16C2Tn_CCEP 寄存器中 CC2POL = '0', 确认极性（只检测高电平）。
- ◇ 配置定时器为触发模式：GP16C2Tn_SMCON 寄存器中 SMODS = "110"。GP16C2Tn_SMCON 寄存器中 TSSEL = "110", 用于选择输入源。

I2 上出现上升沿时，计数器开始依据内部时钟计数并置位 TRGI 标志位。

由于 I2 输入的再同步原因，I2 上出现上升沿和计数器实际停止之间会有一定的延时。

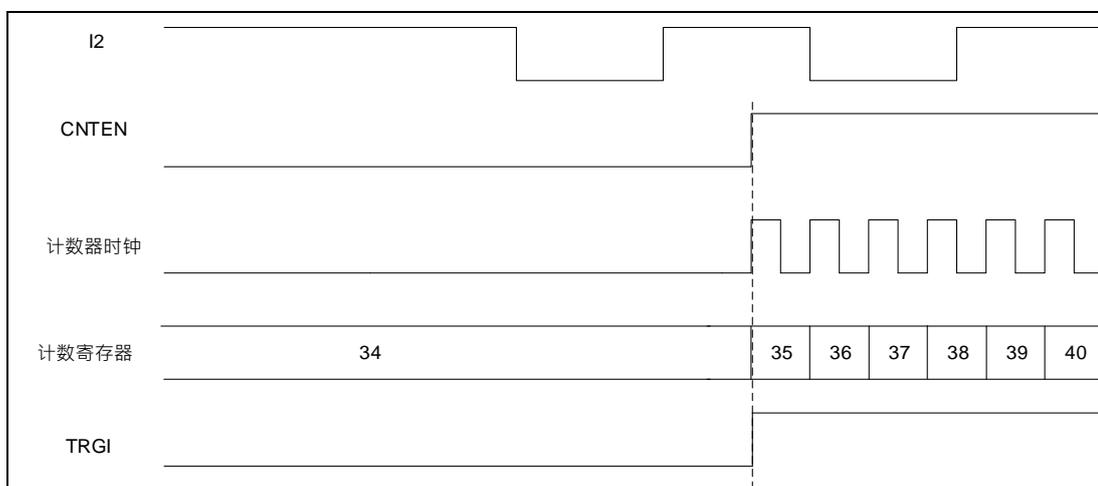


图 15-21 触发模式控制电路

15.4.13 定时器同步

所有定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况

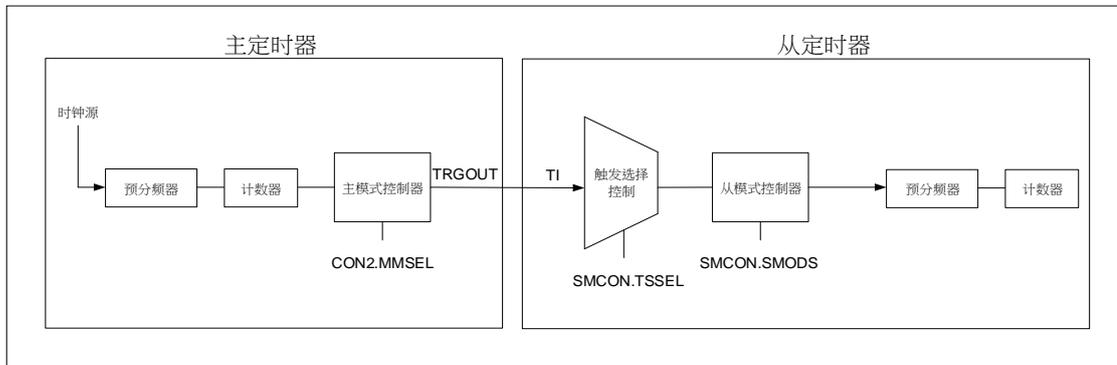


图 15-22 主/从定时器范例

15.4.13.1 使用一个定时器去使能其他定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考下图的连接。只当定时器 1 的 CH1REF 为高时，定时器 2 才对分频后的内部时钟计数。

配置定时器 1 为主模式，送出它的输出比较参考信号 (CH1REF) 为触发输出 (GP32C4T1_CON2 寄存器的 MMSEL=100)

- ◇ 配置定时器 1 的 CH1REF 波形(GP32C4T1_CHMR1 寄存器)
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP16C2T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为门控模式(GP16C2T1_SMCON 寄存器的 SMODS=101)
- ◇ 配置 GP16C2T1_CON1 寄存器的 CNTEN=1 以使能定时器 2
- ◇ 配置 GP32C4T1_CON1 寄存器的 CNTEN=1 以启动定时器 1

注：定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

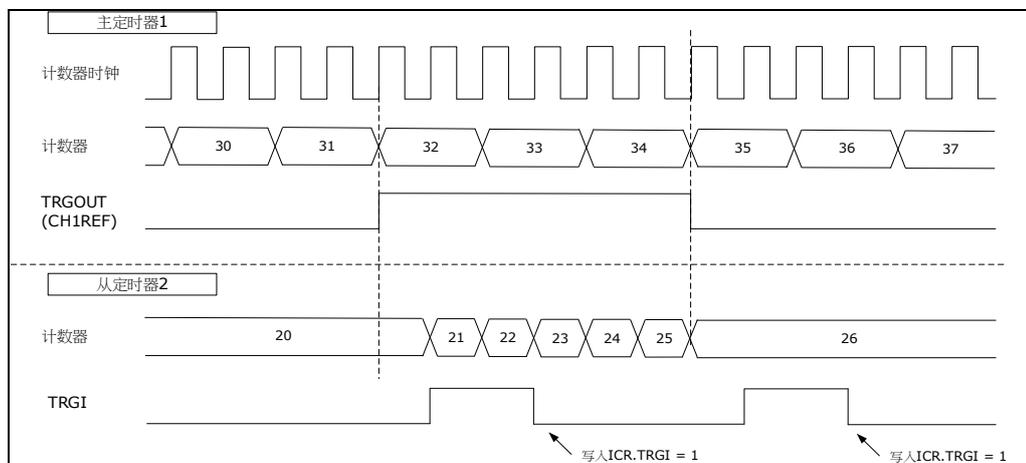


图 15-23 门控从定时器使用主定时器 CH1REF

在上图的例子中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 **GP32C4T1_SGE** 寄存器的 **SGU** 位即可复位定时器。

在下一个例子中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写'0'到 **GP32C4T1_CON1** 的 **CNTEN** 位将禁止定时器 1，定时器 2 随即停止。

- ◇ 配置定时器 1 为主模式，送出 **CNTEN** 位做为触发输出(**GP32C4T1_CON2** 寄存器的 **MMSEL=001**)。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(**GP16C2T1_SMCON** 寄存器的 **TSSEL=000**)
- ◇ 配置定时器 2 为门控模式(**GP16C2T1_SMCON** 寄存器的 **SMODS=101**)
- ◇ 配置 **GP32C4T1_SGE** 寄存器的 **SGU=1**，复位定时器 1。
- ◇ 配置 **GP16C2T1_SGE** 寄存器的 **SGU=1**，复位定时器 2。
- ◇ 写'0xE7'至定时器 2 的计数器(**GP16C2T1_COUNT**)，初始化它为 0xE7。
- ◇ 配置 **GP16C2T1_CON1** 寄存器的 **CNTEN=1** 以使能定时器 2。
- ◇ 配置 **GP32C4T1_CON1** 寄存器的 **CNTEN=1** 以启动定时器 1。
- ◇ 配置 **GP32C4T1_CON1** 寄存器的 **CNTEN=0** 以停止定时器 1。

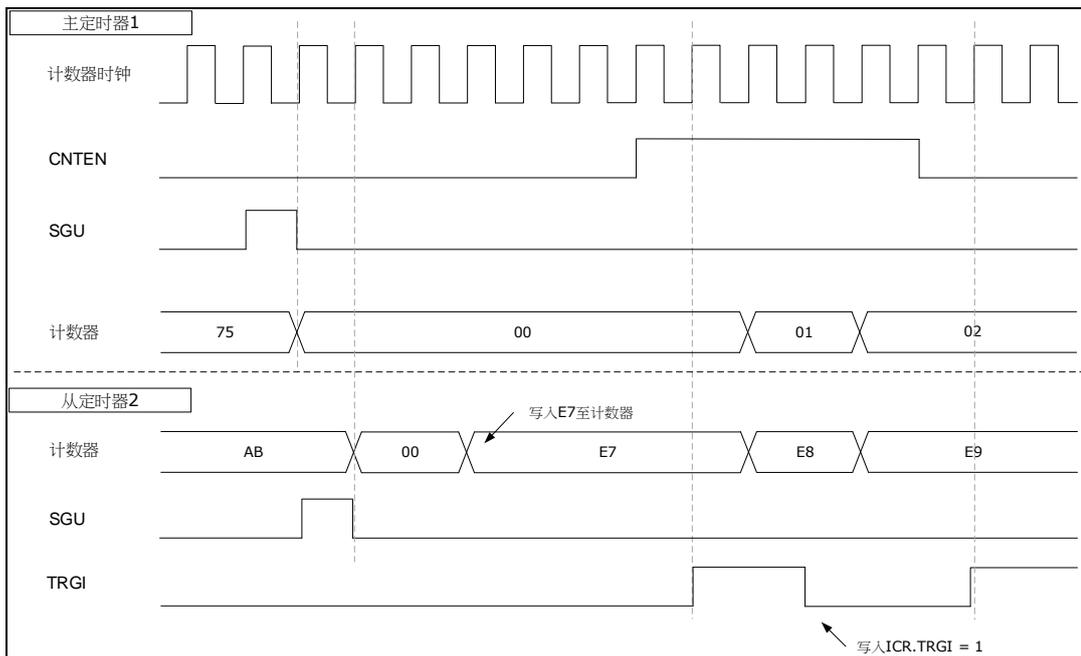


图 15-24 通过使能定时器 1 可以控制定时器 2

15.4.13.2 使用一个定时器去开启其他定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值(可以是非 0)按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CNTEN 位被自动地置'1'，同时计数器开始计数直到写'0'到 GP16C2T1_CON1 寄存器的 CNTEN 位。

- ◇ 配置定时器 1 的周期(GP32C4T1_AR 寄存器)。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP16C2T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为触发模式(GP16C2T1_SMCON 寄存器的 SMODS=110)
- ◇ 配置 GP32C4T1_CON1 寄存器的 CNTEN=1 以启动定时器 1。

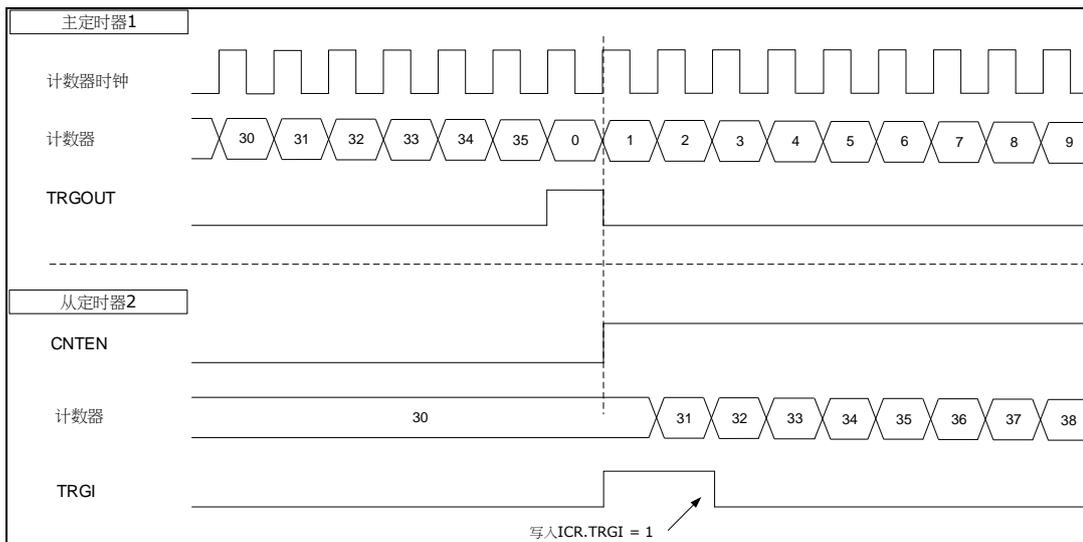


图 15-25 触发中从定时器使用主定时器更新事件

在上图例子中，可以在启动计数之前初始化两个计数器。上图显示在上上图相同配置情况下，使用触发模式而不是门控模式(GP16C2T1_SMCON 寄存器的 SMODS=110)的动作。

15.4.13.3 使用外部触发同步开始两个定时器

这个例子中当定时器 1 的 I1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见下图。为保证计数器的对齐，定时器 1 必须配置为主/从模式(对应 I1 为从，对应定时器 2 为主):

- ◇ 配置定时器 1 为主模式，送出它的使能做为触发输出(GP32C4T1_CON2 寄存器的 MMSEL=001)。
- ◇ 配置定时器 1 为从模式，从 TI1 获得输入触发(GP32C4T1_SMCON 寄存器的 TSSEL='101')。
- ◇ 配置定时器 1 为触发模式(GP32C4T1_SMCON 寄存器的 SMODS='110')。
- ◇ 配置定时器 1 为主/从模式，GP32C4T1_SMCON 寄存器的 MSCFG='1'。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP16C2T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为触发模式(GP16C2T1_SMCON 寄存器的 SMODS='110')。

当定时器 1 的 I1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TRGI 标志也同时被设置。

注：在这个例子中，在启动之前两个定时器都被初始化(设置相应的 SGU 位)，两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器(GP32C4T1_COUNT)在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNTEN 和计数器时钟之间有个延迟。

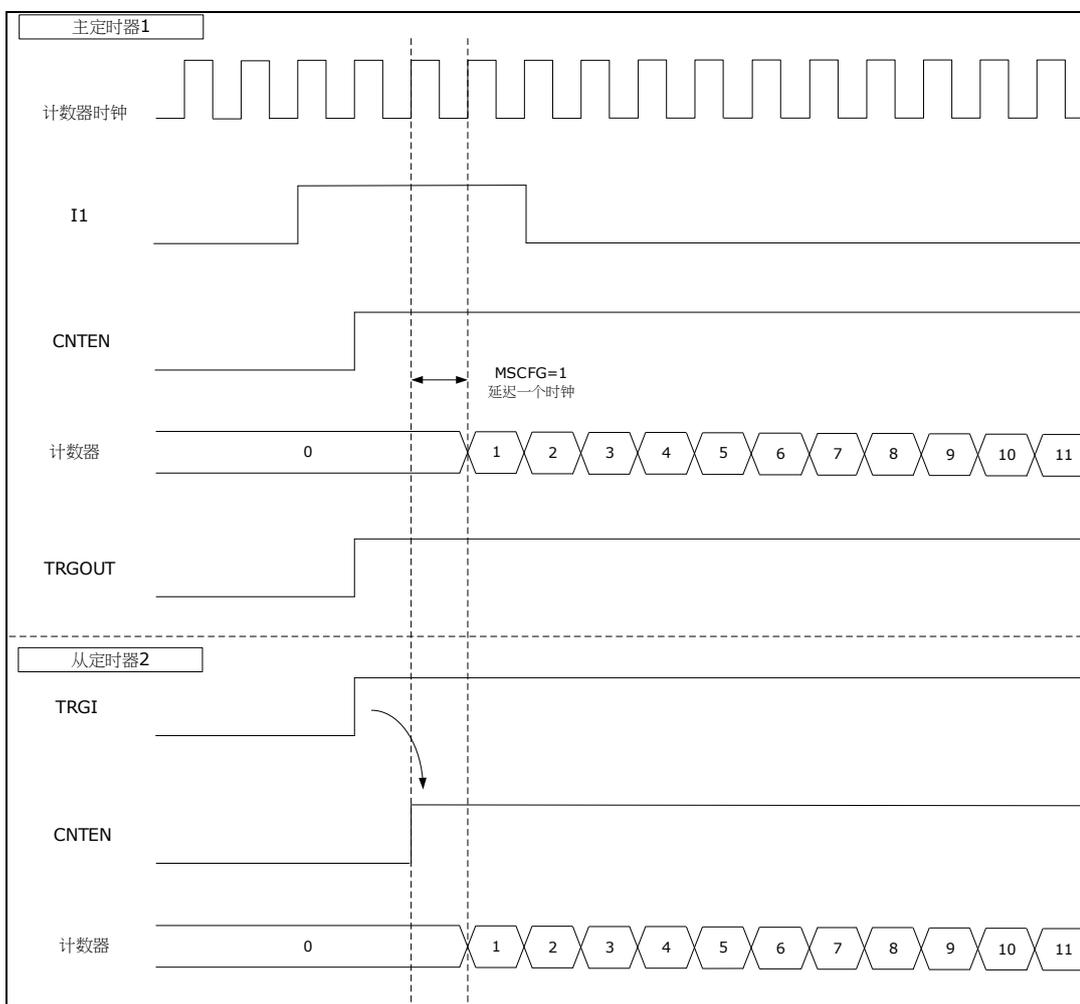


图 15-26 使用定时器 1 的 I1 输入触发定时器 1 和定时器 2

15.4.14 调试模式

当微控制器进入调试模式(Cortex™-M0 核停止运行)，计数器停止计数。

15.5 特殊功能寄存器

15.5.1 寄存器列表

GP16C2Tn 寄存器列表			
名称	偏移地址	类型	描述
GP16C2Tn_CON1	00 _H	R/W	控制寄存器 1
GP16C2Tn_CON2	04 _H	R/W	控制寄存器 2
GP16C2Tn_SMCON	08 _H	R/W	从模式控制寄存器
GP16C2Tn_IER	0C _H	W1	中断使能寄存器
GP16C2Tn_IDR	10 _H	W1	中断禁止寄存器
GP16C2Tn_IVS	14 _H	R	中断有效状态寄存器
GP16C2Tn_RIF	18 _H	R	原始中断标志寄存器
GP16C2Tn_IFM	1C _H	R	中断屏蔽标志寄存器
GP16C2Tn_ICR	20 _H	C_W1	中断标志清除寄存器
GP16C2Tn_SGE	24 _H	W1	软件生成事件寄存器
GP16C2Tn_CHMR1	28 _H	R/W	捕获/比较模式寄存器 1
GP16C2Tn_CCEP	30 _H	R/W	捕获/比较使能极性寄存器
GP16C2Tn_COUNT	34 _H	R/W	计数器
GP16C2Tn_PRES	38 _H	R/W	时钟预分频器
GP16C2Tn_AR	3C _H	R/W	自动重装载寄存器
GP16C2Tn_REPAR	40 _H	R/W	重复计数寄存器
GP16C2Tn_CCVAL1	44 _H	R/W	通道捕获/比较寄存器 1
GP16C2Tn_CCVAL2	48 _H	R/W	通道捕获/比较寄存器 2
GP16C2Tn_BDCFG	54 _H	R/W	刹车和死区配置寄存器
GP16C2Tn_DMAEN	58 _H	R/W	DMA 事件使能寄存器

			出时产生更新事件请求
DISUE	Bit 1	R/W	<p>更新事件禁止 设置更新事件(UEV)的产生</p> <p>0: 更新事件(UEV) 使能, 下列事件中产生更新事件请求</p> <ul style="list-style-type: none"> - 计数器溢出 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 <p>缓冲寄存器载入预装载值</p> <p>1: 更新事件(UEV)禁止, 不产生更新事件请求, AR、PRES、CCVALn 寄存器保持数值</p> <p>禁止更新事件时, 设置 SGE 寄存器的 SGU 位或从模式中产生的复位请求, 计数器和预分频器仍会被重新初始化</p>
CNTEN	Bit 0	R/W	<p>计数器使能 使能计数器后, 在外部时钟模式、门控模式和编码模式才能运作。触发模式则可以由硬件设置 CNTEN 位</p> <p>0: 计数器禁止</p> <p>1: 计数器使能</p>

15.5.2.2 控制寄存器 2 (GP16C2Tn_CON2)

控制寄存器 2 (GP16C2Tn_CON2)																																
偏移地址: 04 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																						OISS2	OISS1N	OISS1		MMSEL<2:0>			CCDMASEL	CCUSEL		CCPCEN

—	Bits 31-11	—	—
OISS2	Bit 10	R/W	通道 2 输出的空闲状态选择位 参考 OISS1 描述
OISS1N	Bit 9	R/W	通道 1 互补输出的空闲状态选择位 0: 当 GOEN=0, 在一段死区时间后, CH1N=0 1: 当 GOEN=0, 在一段死区时间后, CH1N=1 注意: 当 BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 1,2,或 3 后, OISS1N 不可更改。
OISS1	Bit 8	R/W	通道 1 输出的空闲状态选择位 0: 当 GOEN=0, 如果 CH1ON 已实现, 在一段死区时间后, CH1=0 1: 当 GOEN=0, 如果 CH1ON 已实现, 在一段死区时间后, CH1=1 注意: 当 BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 1,2,或 3 后, OISS1N 不可更改。
—	Bit 7	—	—
MMSEL	Bits 6-4	R/W	主模式选择 设置在主模式下发送到从定时器的同步信号 (TRGOUT)与 ADC 输入: 000: 复位 - 设置 SGE 寄存器信号用于同步触发输出(TRGOUT)。从模式的复位触发产生的复位信号(TRGOUT)则与实际信号相差一个时钟; 001: 使能 - 计数器的使能信号 CNTEN 用于同步触发输出(TRGOUT), 可用于同步使能数个定时器。门控模式下, 是使用 CON1 寄存器的 CNTEN 位与触发输入信号逻辑产生。当计数器使能信号受控于触发输入时, TRGOUT 上会有一个时钟延迟, 可设置 SMCON 寄存器的 MSCFG 位延迟一个时钟同步定时器计数器; 010: 更新事件 - 更新事件被用于同步触发输出(TRGOUT)。一个主定时器的更新事件可当作从定时器的预分频器时钟;

			<p>011: 比较脉冲 - 在发生一次捕获或一次比较成功时, 当要设置 CH1I 标志时, 触发输出送出一个正脉冲(TRGOUT) ;</p> <p>100: 比较信号 - CH1REF 信号用于触发输出 (TRGOUT) ;</p> <p>101: 比较信号 - CH2REF 信号用于触发输出 (TRGOUT) ;</p> <p>110: 保留;</p> <p>111: 保留。</p>
CCDMASEL	Bit 3	R/W	<p>捕获/比较事件的 DMA 选择</p> <p>0: 当发生 CHn 事件时, 设置 CHn DMA 请求</p> <p>1: 当发生更新事件时, 设置 CHn DMA 请求</p>
CCUSEL	Bit 2	R/W	<p>捕获/比较更新控制选择</p> <p>此功能只有在有互补输出通道作用</p> <p>0: 在捕获/比较预装载时(CCPCEN =1), 只能通过 SGR 寄存器的 SGCOM 位更新</p> <p>1: 在捕获/比较预装载时(CCPCEN =1), 可通过 SGR 寄存器的 SGCOM 与 TI 的上升沿时被更新</p>
—	Bit 1	R/W	—
CCPCEN	Bit 0	R/W	<p>捕获/比较预装载控制</p> <p>设置后只在通信事件(COM), 即 SGE 寄存器的 SGCOM 与 TI 的上升沿时更新</p> <p>0: CCnEN, CCnNEN 和 CHnMOD 位预装载禁止</p> <p>1: CCnEN, CCnNEN 和 CHnMOD 位预装载使能</p>

			<p>控</p> <p>110: 触发模式 - 计数器在触发输入 TI 的上升沿启动(但不复位), 只有计数器的启动是受控</p> <p>111: 外部时钟模式 1 - 选中的触发输入(TI) 的上升沿驱动计数器</p> <p>注: 如果 I1 双边沿检测被选为触发输入 (TSSEL='100'), 不能使用门控模式。I1 每一次转换, I1 双边沿检测就会输出 1 个脉冲, 而门控模式则是检查触发信号的电平</p>
--	--	--	--

从定时器	IT0(TSSEL =000)	IT1(TSSEL =001)	IT2(TSSEL =010)	IT3(TSSEL=011)
GP16C2T1	GP32C4T1	GP16C4T1	GP16C4T2	GP16C4T3
GP16C2T2	GP32C4T1	GP16C4T1	GP16C4T2	GP16C4T3
GP16C2T3	GP32C4T1	GP16C4T1	GP16C4T2	GP16C4T3
GP16C2T4	GP32C4T1	GP16C4T1	GP16C4T2	GP16C4T3

表 15-1 GP16C2Tn 内部触发连接

15.5.2.4 中断使能寄存器 (GP16C2Tn_IER)

中断使能寄存器 (GP16C2Tn_IER)																																
偏移地址: 0C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																						CH2OVI	CH1OVI		BRKI	TRGI	COMI			CH2I	CH1I	UI

—	Bits 31-11	—	—
CH2OVI	Bit 10	W1	捕获溢出 2 中断使能 0: 写入 0 无效 1: 捕获溢出 CH2 中断使能
CH1OVI	Bit 9	W1	捕获溢出 1 中断使能 0: 写入 0 无效 1: 捕获溢出 CH1 中断使能
—	Bit 8	—	—
BRKI	Bit 7	W1	刹车中断使能 0: 写入 0 无效 1: 刹车中断使能
TRGI	Bit 6	W1	触发中断使能 0: 写入 0 无效 1: 触发中断使能
COMI	Bit 5	W1	通信中断使能 0: 写入 0 无效 1: 通信中断使能
—	Bits 4-3	—	—
CH2I	Bit 2	W1	捕获/比较 2 捕获中断使能 0: 写入 0 无效 1: 捕获 CH2 中断使能
CH1I	Bit 1	W1	捕获/比较 1 捕获中断使能 0: 写入 0 无效 1: 捕获 CH1 中断使能
UI	Bit 0	W1	更新中断使能 0: 写入 0 无效 1: 更新中断使能

15.5.2.5 中断禁止寄存器 (GP16C2Tn_IDR)

中断禁止寄存器 (GP16C2Tn_IDR)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																					CH2OVI	CH1OVI		BRKI	TRGI	COMI			CH2I	CH1I	UI

—	Bits 31-11	—	—
CH2OVI	Bit 10	W1	捕获溢出 2 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH2 中断禁止
CH1OVI	Bit 9	W1	捕获溢出 1 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH1 中断禁止
—	Bit 8	—	—
BRKI	Bit 7	W1	刹车中断禁止 0: 写入 0 无效 1: 刹车中断禁止
TRGI	Bit 6	W1	触发中断禁止 0: 写入 0 无效 1: 触发中断禁止
COMI	Bit 5	W1	通信中断禁止 0: 写入 0 无效 1: 通信中断禁止
—	Bits 4-3	—	—
CH2I	Bit 2	W1	捕获/比较 2 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH2 中断禁止
CH1I	Bit 1	W1	捕获/比较 1 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH1 中断禁止
UI	Bit 0	W1	更新中断禁止 0: 写入 0 无效 1: 更新中断禁止

15.5.2.6 中断有效状态寄存器 (GP16C2Tn_IVS)

中断有效状态寄存器 (GP16C2Tn_IVS)																																
偏移地址: 14 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					CH2OVI	CH1OVI		BRKI	TRGI	COMI				CH2I	CH1I	UI

—	Bits 31-11	—	—
CH2OVI	Bit 10	R	捕获溢出 2 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH2 中断禁止 1: 捕获溢出 CH2 中断使能
CH1OVI	Bit 9	R	捕获溢出 1 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH1 中断禁止 1: 捕获溢出 CH1 中断使能
—	Bit 8	—	—
BRKI	Bit 7	R	煞车中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 煞车中断禁止 1: 煞车中断使能
TRGI	Bit 6	R	触发中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 触发中断禁止 1: 触发中断使能
COMI	Bit 5	R	通信中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 通信中断禁止 1: 通信中断使能
—	Bits 4-3	—	—
CH2I	Bit 2	R	捕获/比较 2 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH2 中断禁止 1: 捕获 CH2 中断使能
CH1I	Bit 1	R	捕获/比较 1 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH1 中断禁止 1: 捕获 CH1 中断使能
UI	Bit 0	R	更新中断有效位

			设置 IER 和 IDR 寄存器使能或禁止 0: 更新中断禁止 1: 更新中断使能
--	--	--	---

15.5.2.7 原始中断标志寄存器 (GP16C2Tn_RIF)

原始中断标志寄存器 (GP16C2Tn_RIF)																																
偏移地址: 18 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																						CH2OVI	CH1OVI		BRKI	TRGI	COMI			CH2I	CH1I	UI

—	Bits 31-11	—	—
CH2OVI	Bit 10	R	捕获溢出2原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH2 原始中断产生
CH1OVI	Bit 9	R	捕获溢出1原始中断标志 当CH1I已经被设置又再次收到捕获信号将计数器数值写入CCVAL1寄存器中。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 捕获溢出 CH1 原始中断产生
—	Bit 8	—	—
BRKI	Bit 7	R	刹车原始中断标志 产生刹车事件时产生。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 刹车原始中断产生
TRGI	Bit 6	R	触发原始中断标志 有触发事件时产生(在从模式-门控时, 在任一沿产生, 其他模式则在有效沿时产生)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 触发原始中断产生
COMI	Bit 5	R	通信原始中断标志 有通信事件时产生(在捕获/比较预装载控制使能时, 更新 CCnEN, CCnNEN, CHnMOD 位)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 通信原始中断产生
—	Bits 4-3	—	—
CH2I	Bit 2	R	捕获/比较 2 原始中断标志 参照 CH1I 描述

			<p>0: 无中断产生 1: 捕获/比较 2 原始中断产生</p>
CH1I	Bit 1	R	<p>捕获/比较 1 原始中断标志 通道 CH1 设置为输出: 计数器匹配 CCVAL1 寄存器时设置, 当 CCVAL1 寄存器大于 AR 寄存器时, 则在递增边界时置起。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 原始中断产生 通道 CH1 设置为输入: 发生捕获事件, 将计数器捕获至 CCVAL1 寄存器中, 于 I1 的有效沿产生。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 原始中断产生</p>
UI	Bit 0	R	<p>更新原始中断标志 当 CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断 - 当重复计数器数值上溢时 (重复计数器为 0 时产生更新事件)。 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器重新初始化时。 - 当计数器被触发事件重新初始化时 0: 无中断产生 1: 更新原始中断产生</p>

15.5.2.8 中断屏蔽标志寄存器 (GP16C2Tn_IFM)

中断屏蔽标志寄存器 (GP16C2Tn_IFM)																																
偏移地址: 1C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																						CH2OVI	CH1OVI		BRKI	TRGI	COMI			CH2I	CH1I	UI

—	Bits 31-11	—	—
CH2OVI	Bit 10	R	捕获溢出2中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH2 中断屏蔽标志产生
CH1OVI	Bit 9	R	捕获溢出1中断屏蔽标志 当IVS寄存器为1时, CH1I已经被设置又再次收到捕获信号将计数器数值写入CCVAL1寄存器中。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 捕获溢出 CH1 中断屏蔽标志产生
—	Bit 8	—	—
BRKI	Bit 7	R	刹车中断屏蔽标志 当 IVS 寄存器为 1 时, 有刹车事件时产生。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 刹车中断屏蔽标志产生
TRGI	Bit 6	R	触发中断屏蔽标志 当 IVS 寄存器为 1 时, 有触发事件时产生(在从模式-门控时, 在任一沿产生, 其他模式则在有效沿时产生)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 触发中断产生
COMI	Bit 5	R	通信中断屏蔽标志 当 IVS 寄存器为 1 时, 产生通信事件时产生(在捕获/比较预装载控制使能时, 更新 CCnEN, CCnNEN 和 CH1MOD 位)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 通信中断屏蔽标志产生
—	Bits 4-3	—	—

CH2I	Bit 2	R	<p>捕获/比较 2 中断屏蔽标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 2 中断屏蔽标志产生</p>
CH1I	Bit 1	R	<p>捕获/比较 1 中断屏蔽标志 当 IVS 寄存器为 1 时 通道 CH1 设置为输出: 计数器匹配 CCVAL1 寄存器时设置, 当 CCVAL1 寄存器大于 AR 寄存器时, 则在递增边界时置起。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 中断屏蔽标志产生 通道 CH1 设置为输入: 发生捕获事件, 将计数器捕获至 CCVAL1 寄存器中, 于 I1 的有效沿产生。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 中断屏蔽标志产生</p>
UI	Bit 0	R	<p>更新中断屏蔽标志 当 CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断 - 当重复计数器数值上溢时 (重复计数器为 0 时产生更新事件)。 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器重新初始化时。 - 当计数器被触发事件重新初始化时 0: 无中断产生 1: 更新中断屏蔽标志产生</p>

15.5.2.9 中断标志清除寄存器 (GP16C2Tn_ICR)

中断标志清除寄存器 (GP16C2Tn_ICR)																																
偏移地址: 20 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																						CH2OVI	CH1OVI		BRKI	TRGI	COMI			CH2I	CH1I	UI

—	Bits 31-11	—	—
CH2OVI	Bit 10	C_W1	捕获溢出 2 中断标志清除 0: 写入0无效 1: 捕获溢出 CH2 中断标志清除
CH1OVI	Bit 9	C_W1	捕获溢出 1 中断标志清除 0: 写入0无效 1: 捕获溢出 CH1 中断标志清除
—	Bit 8	—	—
BRKI	Bit 7	C_W1	刹车中断标志清除 0: 写入 0 无效 1: 刹车中断标志清除
TRGI	Bit 6	C_W1	触发中断标志清除 0: 写入 0 无效 1: 触发中断标志清除
COMI	Bit 5	C_W1	通信中断标志清除 0: 写入 0 无效 1: 通信中断标志清除
—	Bits 4-3	—	—
CH2I	Bit 2	C_W1	捕获/比较 2 中断标志清除 0: 写入 0 无效 1: 捕获 CH2 中断标志清除
CH1I	Bit 1	C_W1	捕获/比较 1 中断标志清除 0: 写入 0 无效 1: 捕获 CH1 中断标志清除
UI	Bit 0	C_W1	更新中断标志清除 0: 写入 0 无效 1: 更新中断标志清除

15.5.2.10 软件生成事件寄存器 (GP16C2Tn_SGE)

软件生成事件寄存器 (GP16C2Tn_SGE)																																	
偏移地址: 24H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																									SGBRK	SGTRG	SGCOM				SGCH2	SGCH1	SGU

—	Bits 31-8	—	—
SGBRK	Bit 7	W1	<p>软件生成刹车事件 该位由软件设置来生成刹车事件，可由硬件自动清零。 0: 无动作 1: 产生刹车事件。GOEN清零，BRKI标志位置起，产生相关中断或DMA传输。</p>
SGTRG	Bit 6	W1	<p>软件生成触发事件 该位由软件设置来生成触发事件，可由硬件自动清零。 0: 无动作 1: RIF 寄存器中的 TRGI 被置起，产生相关中断或 DMA 传输</p>
SGCOM	Bit 5	W1	<p>软件生成通信事件捕获 该位由软件设置来生成通信事件，可由硬件自动清零。。 0: 无动作 1: 当 CCPCEN 被置 1，则可更新 CCnEN, CCnNEN 和 CHnMOD 注意: 该位只有用作于通道时才有互补输出</p>
—	Bits 4-3	—	保留
SGCH2	Bit 2	W1	<p>软件生成通道 2 捕获/比较事件 参考 SGCH1 描述</p>
SGCH1	Bit 1	W1	<p>软件生成通道 1 捕获/比较事件 通道 CH1 设置为输出: 产生捕获/比较但不影响输出，若使能中断或 DMA，则产生中断与请求。由软件设置，于下一个时钟自动清除 0: 写入 0 无效 1: 捕获/比较 1 中断产生 通道 CH1 设置为输入: 发生捕获事件，将计数器捕获至 CCVAL1 寄存器中，于 I1 的有效沿产生，若使能中断或 DMA，则产生中断与请求。由软件设置，于下一个时钟</p>

			自动清除 0: 写入 0 无效 1: 捕获/比较 1 中断产生
SGU	Bit 0	W1	软件触发更新事件 该位由软件设置，可由硬件自动清零。 0: 无动作 1: 重新初始化计数器，更新寄存器。注意，预分频器也会被清零（但预分频比不会受到影响）。

			<p>寄存器时, CH1REF 设置翻转 (当前高/低电平翻转成低/高电平)</p> <p>100: 强制低电平 - CH1REF 强制设置低电平</p> <p>101: 强制高电平 - CH1REF 强制设置高电平</p> <p>110: PWM 模式 1 - 递增模式时, 当计数器小于 CCVAL1 寄存器时, 输出高电平, 其他则输出低电平。递减模式时, 当计数器大于 CCVAL1 寄存器时输出低电平, 其他则输出高电平</p> <p>111: PWM 模式 2 - 递增模式时, 当计数器小于 CCVAL1 寄存器时, 输出低电平, 其他则输出高电平。递减模式时, 当计数器大于 CCVAL1 寄存器时输出高电平, 其他则输出低电平</p>
CH1PEN	Bit 3	R/W	<p>输出比较通道 1 预装载使能</p> <p>设置后在更新事件时, 将设置的寄存器 CCVAL1 数值载入预装载 CCVAL1 寄存器中</p> <p>0: CCVAL1 寄存器预装载禁止</p> <p>1: CCVAL1 寄存器预装载使能</p>
CH1FEN	Bit 2	R/W	<p>输出比较通道 1 快速使能</p> <p>用于加速事件的产生</p> <p>0: CH1 的正常操作依赖于计数器与 CCVAL1 的值, 即使工作于触发器状态。当触发器的输入有一个有效沿时, 激活 CH1 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, CH1 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 CH1 输出间的延时被缩短为 3 个时钟周期。CH1FEN 只在信道被配置成 PWM1 或 PWM2 模式时起作用</p>
CC1SSEL	Bits 1-0	R/W	<p>捕获/比较通道 1 选择</p> <p>设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC1EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入, 捕获源为 I1</p> <p>10: 通道设置为输入, 捕获源为 I2</p> <p>11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测</p>

输入捕获模式

—	Bits 31-16	—	—
I2FLT	Bits 15-12	R/W	输入捕获通道2滤波器 参照I1FLT描述
I2PRES	Bits 11-10	R/W	输入捕获通道 2 预分频器 参照 IC1PRES 描述
CC2SSEL	Bits 9-8	R/W	捕获/比较通道 2 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC2EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I2 10: 通道设置为输入，捕获源为 I1 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测
I1FLT	Bits 7-4	R/W	输入捕获通道 1 滤波器 设置 I1 信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变 0000: 采样频率 fDTS，滤波器禁止 0001: 采样频率 fINT_CLK, N = 2 0010: 采样频率 fINT_CLK, N = 4 0011: 采样频率 fINT_CLK, N = 8 0100: 采样频率 fDTS / 2, N = 6 0101: 采样频率 fDTS / 2, N = 8 0110: 采样频率 fDTS / 4, N = 6 0111: 采样频率 fDTS / 4, N = 8 1000: 采样频率 fDTS / 8, N = 6 1001: 采样频率 fDTS / 8, N = 8 1010: 采样频率 fDTS / 16, N = 5 1011: 采样频率 fDTS / 16, N = 6 1100: 采样频率 fDTS / 16, N = 8 1101: 采样频率 fDTS / 32, N = 5 1110: 采样频率 fDTS / 32, N = 6 1111: 采样频率 fDTS / 32, N = 8
I1PRES	Bits 3-2	R/W	输入捕获通道 1 预分频器 设置 I1 的预分频计数器数值，当清除 CCEP 寄存器的 CC1EN 位，预分频计数器同时被清除 00: 预分频禁止，于每次事件时捕获 01: 每 2 次事件捕获 10: 每 4 次事件捕获 11: 每 8 次事件捕获
CC1SSEL	Bits 1-0	R/W	捕获/比较通道 1 选择 设置通道的输出方向与信号的选择，当 CCEP 寄

			<p>寄存器的 CC1EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入，捕获源为 I1</p> <p>10: 通道设置为输入，捕获源为 I2</p> <p>11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测</p>
--	--	--	---

15.5.2.12 捕获/比较使能极性寄存器 (GP16C2Tn_CCEP)

捕获/比较使能寄存器 (GP16C2Tn_CCEP)																																
偏移地址: 30 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																									CC2NPOL		CC2POL	CC2EN	CC1NPOL	CC1NE	CC1POL	CC1EN

—	Bits 31-8	—	—
CC2NPOL	Bit 7	R/W	捕获/比较通道 2 互补输出有效位极性 参照 CC1NPOL 描述
—	Bit 6	—	—
CC2POL	Bit 5	R/W	捕获/比较通道 2 输出有效位极性 参照 CC1POL 描述
CC2EN	Bit 4	R/W	捕获/比较通道 2 输出使能 参照 CC1EN 描述
CC1NPOL	Bit 3	R/W	捕获/比较通道 2 互补输出有效位极性 通道 CH1 设置为输出: 0: CH1N 高电平有效 1: CH1N 低电平有效 通道 CH1 设置为输入: 该位需和 CC1POL 一起使用来定义输入边沿的极性。参考 CC1POL 描述。 注: 对于有互补输出的通道, 该位设置为预载值。如果 GP16C2Tn_CON2 寄存器中的 CCPCEN 位设置为 1, 则只有当 COM 事件发生时, CC1NP 有效位才会设置为预载值中新的值。 注意: 当 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 2 或 3, 且 CC1SSEL=00 (信道为输出模式), 该位将不可写。
CC1NEN	Bit 2	R/W	捕获/比较通道 1 互补输出使能 0: 关闭 - CH1N 无效。CH1N 电平取决于 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1EN 的功能 1: 开启 - CH1N 为对应输出引脚上的输出信号, 由 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1EN 决定。 注: 对于有互补输出的通道, 该位设置为预载值。如果 GP16C2Tn_CON2 寄存器中的 CCPCEN

			位设置为 1, 则只有当 COM 事件发生时, CC1NE 有效位才会设置为预载值中新的值
CC1POL	Bit 1	R/W	<p>捕获/比较通道 1 输出有效位极性</p> <p>通道 CH1 设置为输出:</p> <p>0: CH1 高电平有效</p> <p>1: CH1 低电平有效</p> <p>通道 CC1 设置为输入:</p> <p>CC1NPOL/CC1POL 位选择触发边沿或捕获模式下 I1 和 I2 的极性</p> <p>00: 非反相/上升沿</p> <p>01: 反相/下降沿</p> <p>10: 保留</p> <p>11: 非反相/上升沿+下降沿</p> <p>注: 对于有互补输出的通道, 该位设置为预载值。如果 GP16C2Tn_CON2 寄存器中的 CCPCEN 位设置为 1, 则只有当 COM 事件发生时, CC1POL 有效位才会设置为预载值中新的值。</p> <p>注: 当 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 2 或 3, 且 CC1SSEL=00 (信道为输出模式), 该位将不可写。</p>
CC1EN	Bit 0	R/W	<p>捕获/比较通道 1 输出使能</p> <p>通道 CH1 设置为输出:</p> <p>0: 关闭 - CH1 无效。CH1 电平取决于 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1NEN 的功能</p> <p>1: 开启 - CH1 为对应输出引脚上的输出信号, 由 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1NEN 决定</p> <p>通道 CH1 设置为输入:</p> <p>0: 捕获禁止</p> <p>1: 捕获使能</p> <p>注: 对于有互补输出的通道, 该位设置为预载值。如果 GP16C2Tn_CON2 寄存器中的 CCPCEN 位设置为 1, 则只有当 COM 事件发生时, CC1EN 有效位才会设置为预载值中新的值。</p>

15.5.2.13 计数寄存器 (GP16C2Tn_COUNT)

计数寄存器 (GP16C2Tn_COUNT)																																														
偏移地址: 34 _H																																														
复位值: 00000000_00000000_00000000_00000000 _B																																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																CNTV<15:0>																														

—	Bits 31-16	—	—
CNTV	Bits 15-0	R/W	计数器数值

15.5.2.14 预分频寄存器 (GP16C2Tn_PRES)

时钟预分频寄存器 (GP16C2Tn_PRES)																																														
偏移地址: 38 _H																																														
复位值: 00000000_00000000_00000000_00000000 _B																																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																PSCV<15:0>																														

—	Bits 31-16	—	—
PSCV	Bits 15-0	R/W	预分频数值 当计数器时钟频率等于fINT_CLK/(PSCV<15:0> + 1)时计数器递增。在更新事件产生时，将PSCV数值被载入预装载寄存器中

15.5.2.15 自动重载寄存器 (GP16C2Tn_AR)

自动重载寄存器 (GP16C2Tn_AR)																																															
偏移地址: 3C _H																																															
复位值: 00000000_00000000_11111111_11111111 _B																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
																ARV<15:0>																															

—	Bits 31-16	—	—
ARV	Bits 15-0	R/W	自动装载数值 设置计数器的递增边界，设置数值为 0 时计数器停止计数

15.5.2.16 重复计数寄存器 (GP16C2Tn_REPAR)

重复计数寄存器 (GP16C2Tn_REPAR)

偏移地址: 40H																															
复位值: 00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												REPV <7:0>			

—	Bits 31-8	—	—
REPV	Bits 7-0	R/W	<p>重复计数数值</p> <p>当预载寄存器使能，该位允许用户设置比较寄存器的更新率（例如：预载到有效寄存器的周期性传输），同样也可以设置更新中断生成率。每次当 REPV_CNT 的相关递减计数器递减至 0，会产生更新事件，会从 REPV 值重新计数。因为只有当发生重复更新事件时，REPV_CNT 才会重新载入 REPV 值，所以只有在发生下一次重复更新事件时，写入 GP16C2Tn_REPAR 寄存器的值才会生效。</p> <p>即，在 PWM 模式下，(REPV+1) 相当于：</p> <ul style="list-style-type: none"> – 在边沿对齐模式下，(REPV+1) 对应的是 PWM 的周期数 – 在中央对齐模式下，(REPV+1) 对应的是 1/2 PWM 的周期数

15.5.2.17 通道捕获/比较寄存器 1 (GP16C2Tn_CCVAL1)

通道捕获/比较寄存器 1 (GP16C2Tn_CCVAL1)																															
偏移地址: 44 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CCRV1[15:0] [△]															

—	Bits 31-16	—	—
CCRV1	Bits 15-0	R/W	<p>捕获/比较数值 1 信道 CHn 配置为输出: CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。 如果在 GP16C2Tn_CHMRn 寄存器中的预载功能没有选中, CCRVn 中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与 GP16C2Tn_COUNT 中的值进行比较, 并在 CHn 上输出。</p> <p>信道 CHn 配置为输入: CCRVn 为由上一个输入捕获事件 (In) 传输的计数值。</p>

15.5.2.18 通道捕获/比较寄存器 2 (GP16C2Tn_CCVAL2)

通道捕获/比较寄存器 2 (GP16C2Tn_CCVAL2)																																													
偏移地址: 48H																																													
复位值: 00000000_00000000_00000000_00000000 _b																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																CCRV2<15:0>																													

—	Bits 31-16	—	—
CCRV2	Bits 15-0	R/W	<p>捕获/比较数值2</p> <p>CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。</p> <p>如果在 GP16C2Tn_CHMRn 寄存器中的预载功能没有选中, CCRVn 中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与 GP16C2Tn_COUNT 中的值进行比较, 并在 CHn 上输出。</p> <p>信道 CHn 配置为输入:</p> <p>CCRVn 为由上一个输入捕获事件 (In) 传输的计数值。</p>

15.5.2.19 刹车和死区配置寄存器 (GP16C2Tn_BDCFG)

刹车和死区配置寄存器 (GP16C2Tn_BDCFG)																																	
偏移地址: 54H																																	
复位值: 00000000_00000000_00000000_00000000 _b																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																GOEN	AOEN	BRKP	BRKEN	OFFSSR	OFFSSI	LOCKLVL <1:0>					DT <7:0>						

—	Bits 31-16	—	—
GOEN	Bit 15	R/W	<p>通道主要输出使能</p> <p>一旦刹车输入有效, 该位会由硬件异步清零。该位可由软件置1或自动置1, 取决于 AOEN 位。该位仅作用于配置为输出的信道。</p> <p>0: CHn 和 CHnN 输出禁止或强制为空闲状态。</p> <p>1: 如果 CHn 和 CHnN 各自的使能位都置 1</p>

			(GP16C2Tn_CCEP 寄存器中的 CCnEN, CCnNEN), 则CHn和CHnN输出使能。
AOEN	Bit14	R/W	<p>通道自动输出使能 在发生更新事件时, 将 GOEN 位置起 0: GOEN 仅可由软件置位 1: 在下一个更新事件发生时 (如果刹车输入无效), GOEN 可由软件或自动置位。 注意: 当 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 则该位不可更改。</p>
BRKP	Bit 13	R/W	<p>选择通道刹车极性 0: 刹车输入 BRKP 为低有效 1: 刹车输入 BRKP 为高有效 注意: 当 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 则该位不可更改 注意: 任何对该位的写操作都要延时 1APB 时钟周期后才变为有效。</p>
BRKEN	Bit 12	R/W	<p>使能刹车 0: 刹车输入 (BRKP 和 CCS 时钟失效事件) 禁止 1: 刹车输入 (BRKP 和 CCS 时钟失效事件) 使能 注意: 当 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 则该位不可更改 注意: 任何对该位的写操作都要延时 1APB 时钟周期后才变为有效。</p>
OFFSSR	Bit 11	R/W	<p>运行模式下关闭状态选择 当 GOEN 为 1 且通道设置为输出时。当定时器禁止时, CCnEN 为 1 或 CCnNNE 为 1 时, 则 CHn/CHnN 输出 CCxPOL/CCxNPPOL 0: 当定时器禁止时, 输出禁止 (CHn/CHnN 使能输出信号=0)。 1: 当定时器禁止时, 输出 CCxPOL/CCxNPPOL 注: 当 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 2, 则该位不可更改。</p>
OFFSSI	Bit 10	R/W	<p>空闲模式下关闭选择 当 GOEN 为 0 且通道设置为输出时。当定时器禁止时, CCnEN 为 1 或 CCnNNE 为 1 时, CHn/CHnN 输出空闲状态 0: 当定时器禁止时, 输出禁止 (CHn/CHnN 使能输出信号=0)。</p>

			<p>1: 当定时器禁止时, 输出空闲状态(OISSx/OISSxN)</p> <p>注: 当 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 2, 则该位不可更改。</p>
LOCKLVL	Bits 9-8	R/W	<p>锁定级别配置</p> <p>针对软件错误, 该位提供写保护。</p> <p>00: 锁定关闭—不提供写保护</p> <p>01: 锁定级别 1 = BDCFG 寄存器中的 DT, CON2 寄存器中的 OISSx 和 OISSxN, 和 BDCFG 寄存器中的 BRKEN/BRKP/AOEN 不再可写。</p> <p>10: 锁定级别 2 = 锁定级别 1 + CC 极性位 (CCEP 寄存器中的 CCnPOL/CCnNPOL, 只要相关信道由 CCnSSEL 配置为输出) 以及 OFFSSR 和 OFFSSI 都不再可写。</p> <p>11: 锁定级别 3 = 锁定级别 2 + CC 控制位 (CHMRn 寄存器中的 CHnMOD 和 CHnPEN, 只要相关信道由 CCnSSEL 配置为输出) 都不再可写。</p> <p>注: 锁定配置为仅在复位后可写。一旦 BDCFG 已写, 其设置内容在下一个复位前都处于冻结状态。</p>
DT	Bits 7-0	R/W	<p>死区延时</p> <p>设置值该位定义了互补输出之间插入的死区时间。DT 对应的就是该时间段。</p> <p>DT[7: 5]=0xx => DT=DT[7:0]x t_{dtg}, 式中 t_{dtg}=t_{DTS}。</p> <p>DT[7: 5]=10x => DT=(64+DT[5:0]) x t_{dtg}, 式中 t_{dtg}=2x t_{DTS}。</p> <p>DT[7: 5]=110=> DT=(32+DT[4:0]) x t_{dtg}, 式中 t_{dtg}=8x t_{DTS}。</p> <p>DT[7: 5]=111 => DT=(32+DT[4: 0]) x t_{dtg}, 式中 t_{dtg}=16x t_{DTS}。</p> <p>注意: 当 GP16C2Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 2 或 3, 则该位不可更改</p>

15.5.2.20 DMA 事件使能寄存器 (GP16C2Tn_DMAEN)

DMA 事件使能寄存器 (GP16C2Tn_DMAEN)																																
偏移地址: 58H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																										TRGIDE	COMDE			CH2DE	CH1DE	UDE

—	Bits 31-7	—	—
TRGIDE	Bit 6	R/W	触发DMA请求使能 0: 触发DMA请求禁止 1: 触发DMA请求使能
COMDE	Bit 5	R/W	通信 DMA 请求使能 0: 通信 DMA 请求禁止 1: 通信 DMA 请求使能
—	Bits 4-3	—	—
CH2DE	Bit 2	R/W	通道捕获/比较 2 DMA 请求使能 0: 捕获/比较 2 DMA 请求禁止 1: 捕获/比较 2 DMA 请求使能
CH1DE	Bit 1	R/W	通道捕获/比较 1 DMA 请求使能 0: 捕获/比较 1 DMA 请求禁止 1: 捕获/比较 1 DMA 请求使能
UDE	Bit 0	R/W	更新 DMA 请求使能 0: 更新 DMA 请求禁止 1: 更新 DMA 请求使能

第16章 通用定时器 16 位 4 通道 (GP16C4T)

16.1 概述

通用定时器 16 位 4 通道 (GP16C4Tn) 包含一个 16 位自动重载计数器，该计数器由可配置的预分频器驱动。

通用定时器 16 位 4 通道 (GP16C4Tn) 的用途广泛，可测量信号脉冲长度（输入捕获）或输出脉冲波形（比较输出、PWM）。

16.2 特性

- ◆ 16 位递增，递减，递增/递减自动加载计数器
- ◆ 16 位可编程预分频器，可在定时器运行中对计数器工作时钟进行 1 到 65536 间的任意分频
- ◆ 带有四个独立信道，每个信道支持以下功能
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ PWM 输出（边沿与中央对齐模式）
 - ◇ 单脉冲输出
- ◆ 同步电路用于外部信号控制定时器及内部互联多个定时器
- ◆ 以下事件中产生中断/DMA:
 - ◇ 更新事件：计数器上溢/下溢，计数器初始化（通过软件或内/外部触发）
 - ◇ 触发事件（计数器起始、停止、初始化或内/外触发计数）
 - ◇ 输入捕获
 - ◇ 输出比较
- ◆ 支持增量（正交）编码及霍尔电路进行定位
- ◆ 触发输入可对外部时钟或周期性电流管理

16.3 结构图

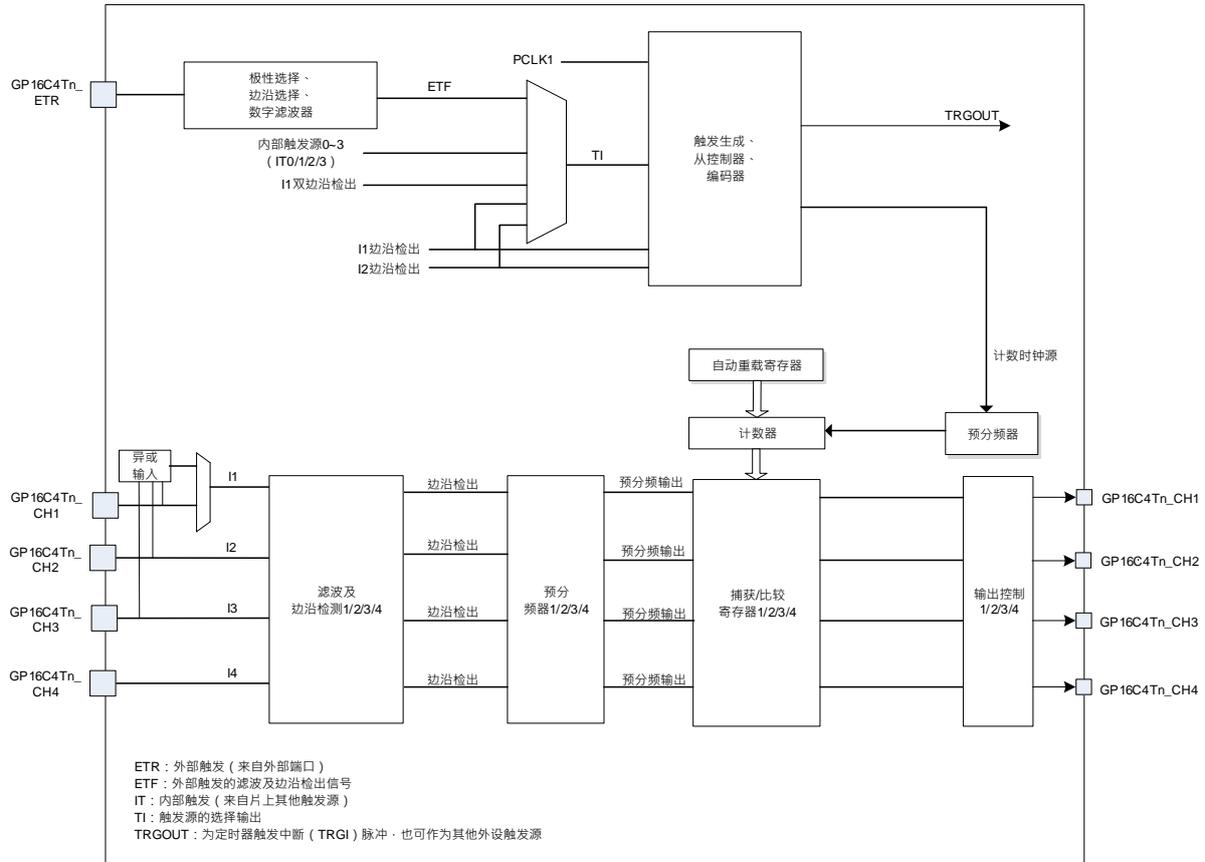


图 16-1 GP16C4Tn 结构框图

16.4 功能描述

16.4.1 预分频器

定时器包含一个 16-bit 的计数器（GP16C4Tn_COUNT），计数时钟由预分频寄存器（GP16C4Tn_PRES）进行分频。计数周期由自动重载计数器（GP16C4Tn_AR）设定。

自动重载寄存器（GP16C4Tn_AR）是一个可缓存的寄存器。当 GP16C4Tn_CON1 寄存器的 ARPEN 位复位时，GP16C4Tn_AR 寄存器重载功能失效，GP16C4Tn_AR 就是有效寄存器；ARPEN 置位时，GP16C4Tn_AR 寄存器具有重载功能，产生更新事件（UEV）时，加载值（GP16C4Tn_AR 寄存器值）更新到影子寄存器。

当 GP16C4Tn_CON1 寄存器中 DISUE 位为 0 时，计数器计数上溢（或递减下溢）时会产生更新事件（UEV）。同样，软件方式也可产生更新事件。GP16C4Tn_CON1 寄存器的 CNTEN 置位时，计数器开始计数。

注：计数器在 CNTEN 位置位 1 个时钟周期后开始计数。

预分频器可对定时器工作时钟进行 GP16C4Tn_PRES 寄存器值+1 次分频。由于 GP16C4Tn_PRES 是一个可重载寄存器，因此，定时器工作时可以对该寄存器进行修改，修改值在下次更新事件（UEV）后有效。

下图给出了定时器运行过程中改变预分频值时计数器的计数情况。

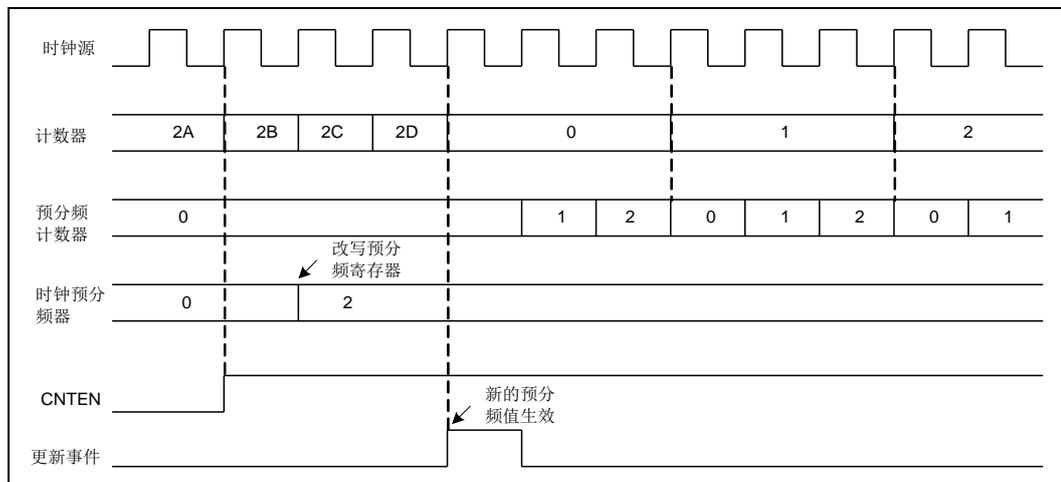


图 16-2 预分频值计数时序图

16.4.2 时钟源

计数器工作时钟可以选择内部时钟(INT_CLK)、外部时钟源 1(I1、I2)、外部时钟源 2(ETR)，内部触发输入 (IT0、IT1、IT2、IT3)

16.4.2.1 内部时钟源 (INT_CLK)

若从模式控制器被关闭 (GP16C4Tn_SMCON 寄存器内, SMODS= "000"), 则 CNTEN, GP16C4Tn_CON1.DIRSEL 与 GP16C4Tn_SGE.SGU 位为实际控制位, 这些位只能软件修改 (SGU 位除外, 仍硬件自动清除)。一旦 CNTEN 位被写为'1', 预分频器就由内部 INT_CLK 提供时钟。

下图给出了通常模式下控制电路和递增计数的情况, 没有分频。

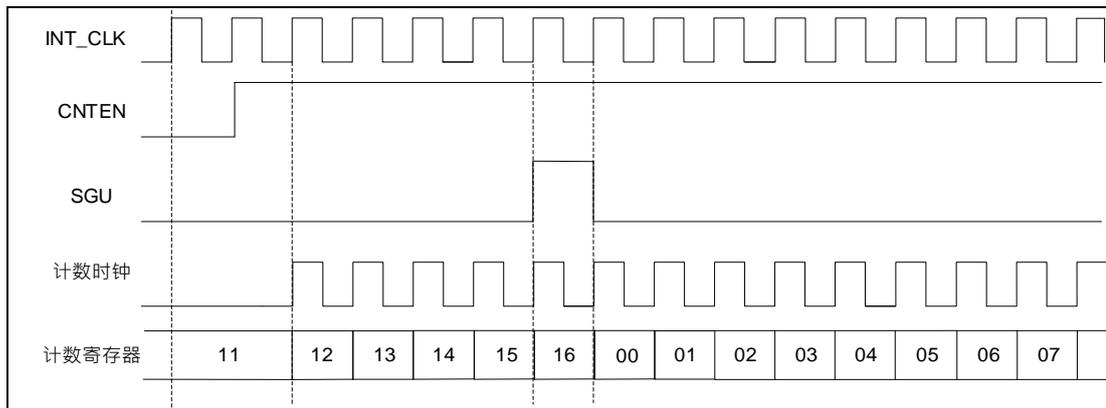


图 16-3 采用内部时钟计数

16.4.2.2 外部时钟源 1

GP16C4Tn_SMCON 寄存器的 **SMODS** = "111" 时, 可选择外部时钟源 1。计数器可根据选定的上升沿或下降沿计数。

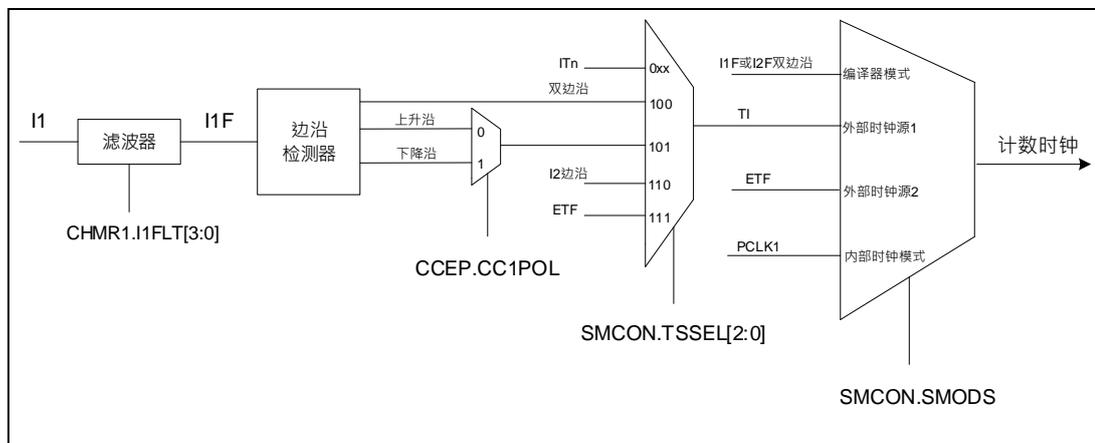


图 16-4 外部时钟连接

配置计数器为外部时钟源 1, 步骤如下:

1. **GP16C4Tn_SMCON** 寄存器中 **SMODS** = "111", 配置定时器外部时钟模式 1。
2. 设置 **GP16C4Tn_SMCON** 寄存器中的 **TSSEL** 选择外部时钟源。
3. 如外部时钟源为 I1, 可配置 **GP16C4Tn_CHMR1** 寄存器 **CC1SSEL** = "01", 配置信道 1 检测 I1 输入的上升沿; 设置 **GP16C4Tn_CCEP** 寄存器中 **CC1POL** = '0', 选择极性为上升沿。
4. 写 **GP16C4Tn_CHMR1** 寄存器的 **I1FLT[3: 0]** 位, 配置输入滤波器时间 (若没有滤波器需求, 维持 **I1FLT** = "0000")。
5. **GP16C4Tn_CON1** 寄存器中 **CNTEN** = '1', 使能计数器。

当 I1 上出现一次上升沿时, 计数器计数一次且 **TRGI** 标志位置位。

16.4.2.3 外部时钟源 2

置位 **GP16C4Tn_SMCON** 寄存器的 ECM2EN 位选定外部时钟源 2。

计数器可对外部触发输入 ETR 进行上升沿或下降沿计数。

下图给出了外部输入模块的概况。

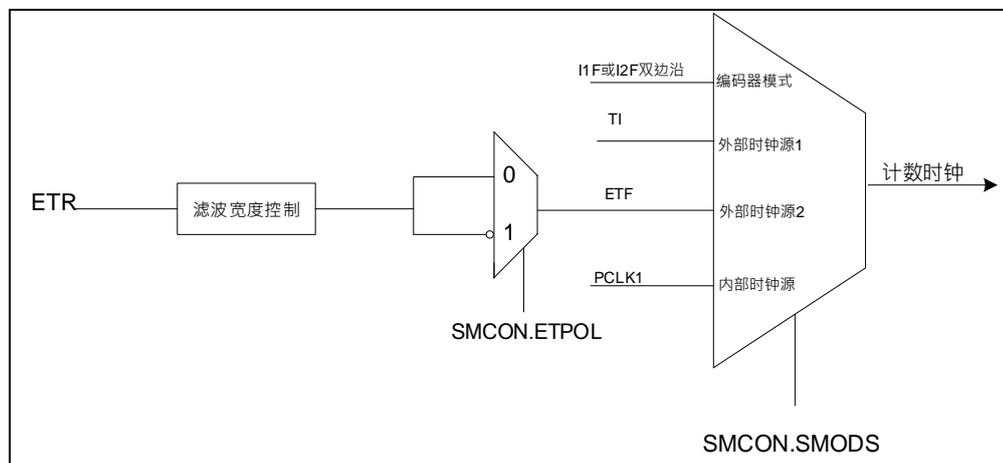


图 16-5 外部触发输入模块

配置计数器为外部时钟源 2，配置过程如下：

1. 设置 **GP16C4Tn_SMCON** 寄存器的 ETFLT[3: 0]，配置输入滤波时间。
2. 设置 **GP16C4Tn_SMCON** 寄存器中 ETPOL，检测 ETR 引脚上升沿或下降沿。
3. 设置 **GP16C4Tn_SMCON** 寄存器中 ECM2EN = '1'，使能外部时钟模式 2。
4. 设置 **GP16C4Tn_CON1** 寄存器的 CNTEN = '1'，使能计数器。

计数器每一个上升沿计一次数。

16.4.2.4 内部触发输入 (ITn)

当 **GP16C4Tn_SMCON** 寄存器的 **SMODS** = "111", 选定内部触发模式。计数器根据选定的内部输入端的上升或下降沿计数。

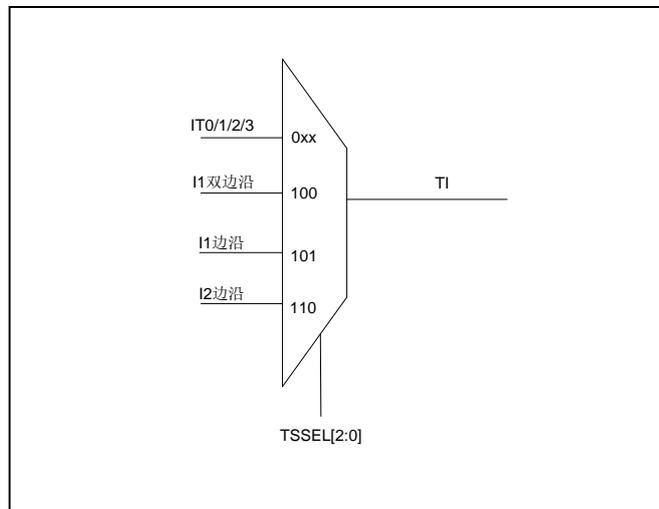


图 16-6 ITn 外部时钟连接

配置计数器在 ITn 输入端的上升沿递增计数，步骤如下：

4. **GP16C4Tn_SMCON** 寄存器中 **SMODS** = "111", 配置外部时钟模式 1。
5. **GP16C4Tn_SMCON** 寄存器的 **TSSEL** = "0xx", 选定 ITn 作为触发输入源。
6. **GP16C4Tn_CON1** 寄存器的 **CNTEN** = '1', 使能计数器。

ITn 产生上升沿时，计数器计数一次。ITn 上升沿与实际时钟间的延时，取决于 ITn 输入的再同步电路。

16.4.3 计数模式

16.4.3.1 递增计数模式

当 **GP16C4Tn_CON1** 寄存器的 **DIRSEL** 值为 0 时，定时器配置为递增模式，计数器从 0 开始递增，直至 **GP16C4Tn_AR** 寄存器值；然后从 0 重新开始计数并产生一个更新事件(UEV)。

当有更新事件（UEV）产生时，预装载寄存器会更新到影子寄存器，更新标志位（**GP16C4Tn_RIF** 寄存器中的 **UI** 位）置位（取决于 **UERSEL** 位）：

- ◇ 更新 **GP16C4Tn_AR** 寄存器的值到影子寄存器
- ◇ 更新 **GP16C4Tn_PRES** 寄存器的值到影子寄存器

下图为 **GP16C4Tn_AR = 0x16**，预分频设为 2 分频时的计数器时序。

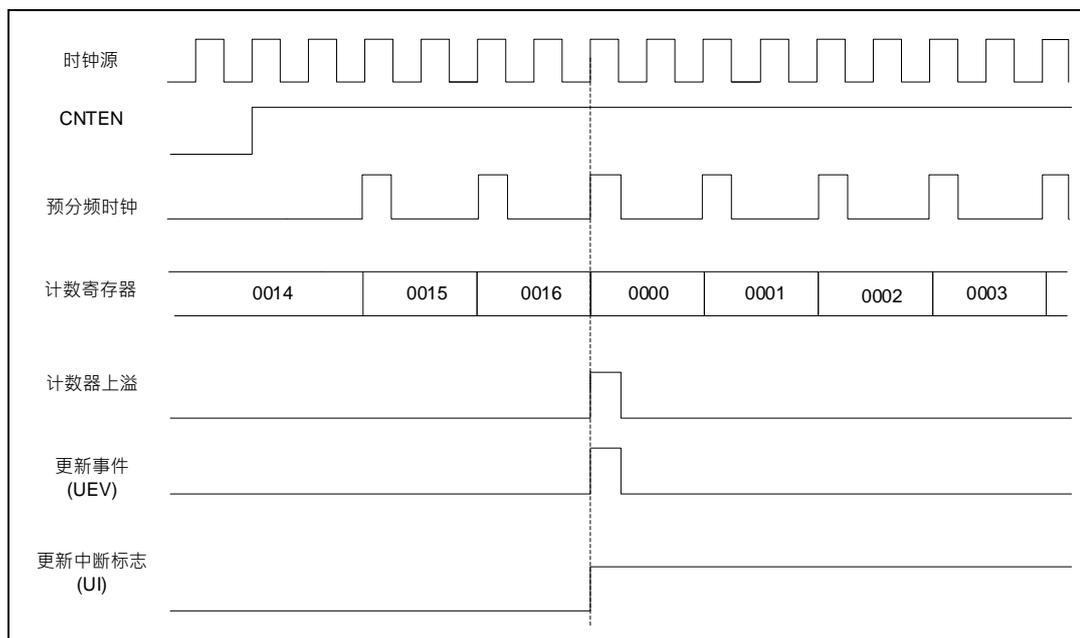


图 16-7 计数器递增计数时序图

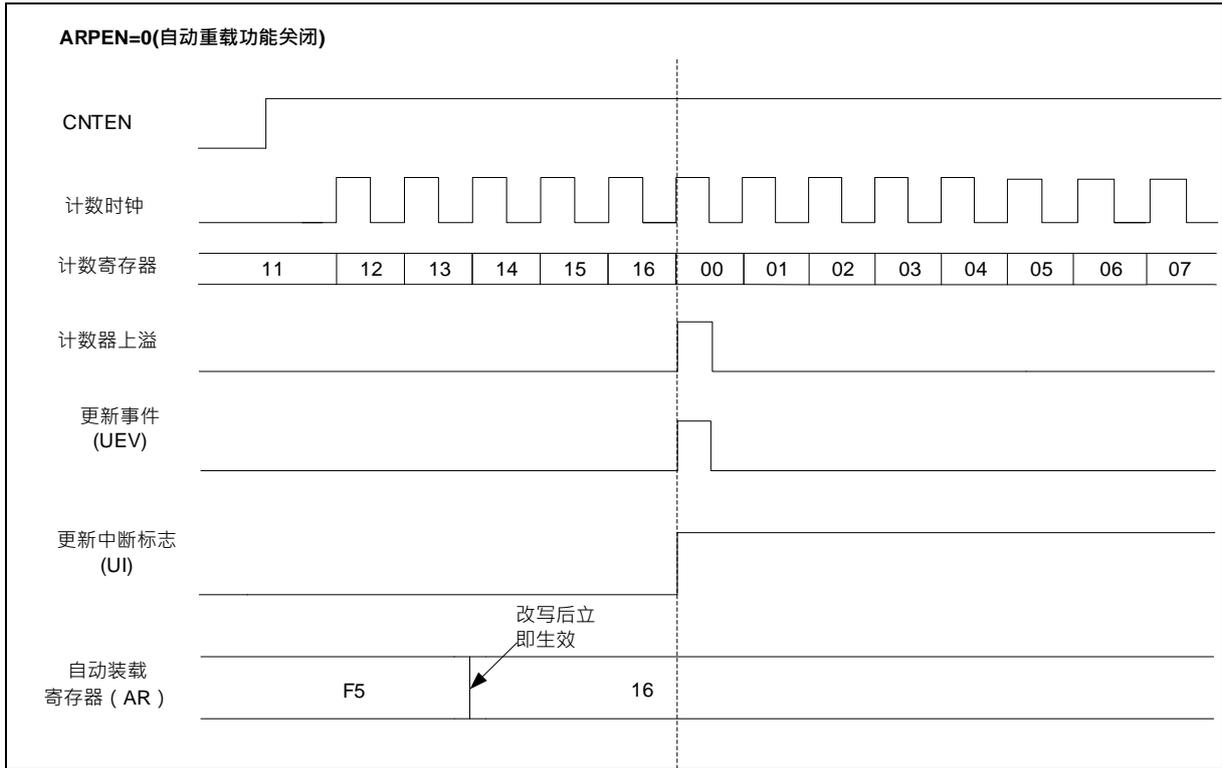


图 16-8 当 ARPEN=0 时计数器时序图

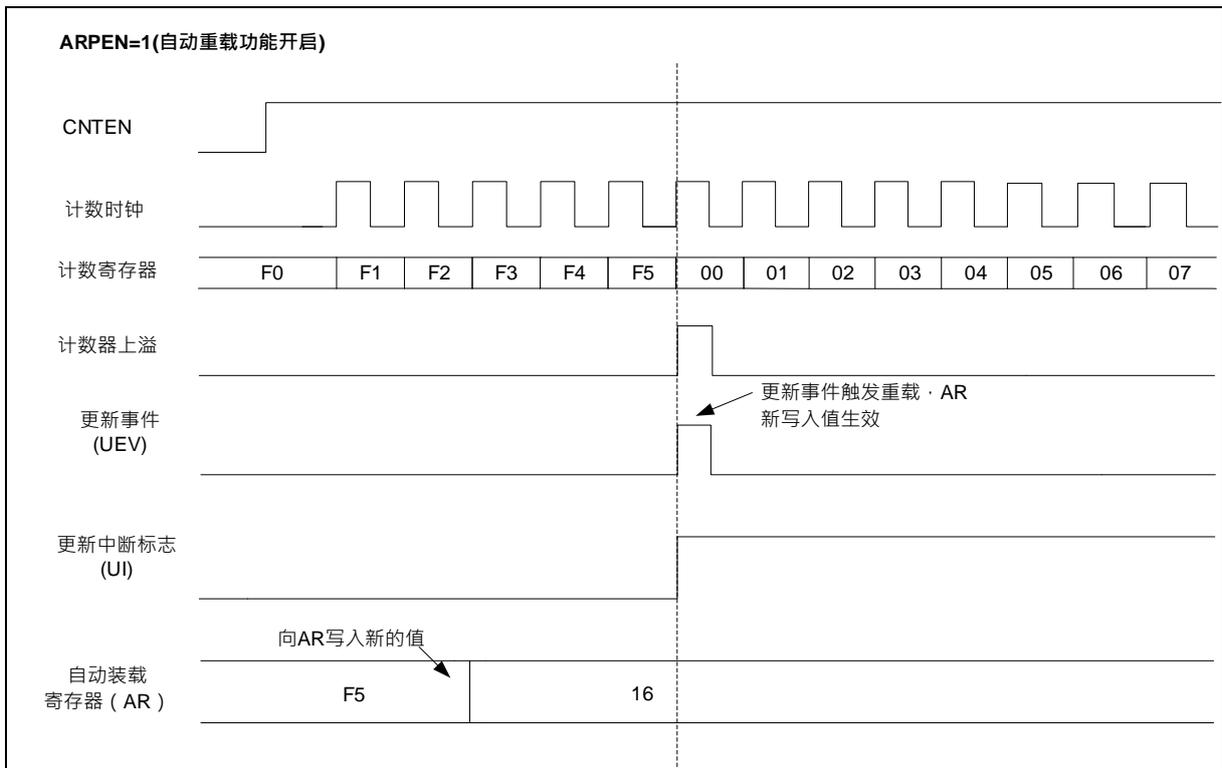


图 16-9 当 ARPEN=1 时计数器时序图

16.4.3.2 递减计数模式

当 **GP16C4Tn_CON1** 寄存器的 **DIRSEL** 值为 1 时，定时器配置为递减模式，计数器从 **GP16C4Tn_AR** 寄存器值开始递减至 0；然后重复递减并产生更新事件（UEV）。

置位 **GP16C4Tn_SGE** 寄存器中的 **SGU** 位（通过软件或使用从机模式控制器）同样会产生更新事件。

当有更新事件（UEV）产生时，预载寄存器值会更新到影子寄存器，更新标志位（**GP16C4Tn_RIF** 寄存器中的 **UI** 位）置位（取决于 **UERSEL** 位）。

下图为 **GP16C4Tn_AR = 0x27**，预分频设为 1 分频时的计数器时序。

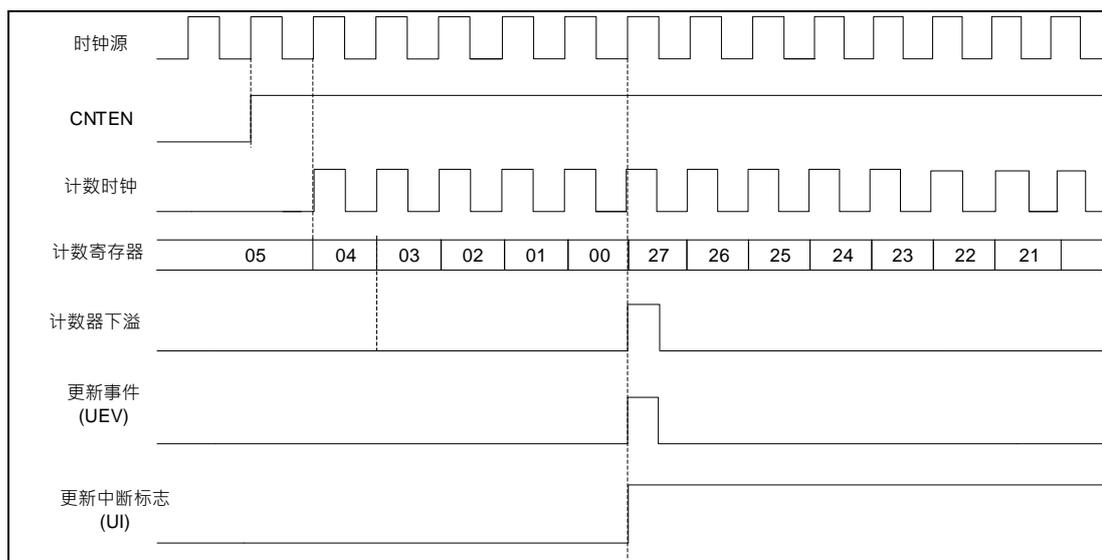


图 16-10 定时器递减计数时序图

16.4.3.3 中央对齐模式

当 **GP16C4Tn_CON1** 寄存器的 **CMSEL** 位的值不等于"00"时,定时器工作在中央对齐模式。定时器配置为中央对齐模式时,计数器先从 0 开始递增至 **GP16C4Tn_AR** 寄存器值减 1 后产生更新事件(UEV);接着计数器从 **GP16C4Tn_AR** 寄存器值递减至 1,并产生更新事件,如此循环计数。计数器递减计数(中央对称模式 1, **CMSEL**="01")、计数器递增计数(中央对称模式 2, **CMSEL**="10")、计数器递增和递减计数(中央对称模式 3, **CMSEL**="11"),每个通道的输出比较中断标志位都会置位。

在中央对齐模式下, **GP16C4Tn_CON1** 寄存器的 **DIRSEL** 位无法进行写操作,该位由硬件自动更新指示当前计数方向。

计数上溢、下溢或者置位 **GP16C4Tn_SGE** 寄存器的 **SGU** 位(通过软件或使用从模式控制器)都会产生更新事件。因此,计数器根据 **DIRSEL** 位由 0 或是 **GP16C4Tn_AR** 寄存器值开始递增或递减,而预分频器都从 0 开始计数。

软件置位 **GP16C4Tn_CON1** 寄存器中的 **DISUE** 位可关闭更新事件(UEV)的产生。更新事件(UEV)关闭时,可避免向预载寄存器写新值时更新影子寄存器。**DISUE** 复位之前都不会产生更新事件。而在正常产生更新事件时,计数器仍然从 0 开始,同样预分频计数也是从 0 开始(但预分频值没有改变)。此外,若置位 **GP16C4Tn_CON1** 寄存器中的 **UERSEL** 位(更新请求选择),置位 **SGU** 位时会产生一次更新事件(UEV),但 **UI** 标志位不会置位(因此,不会触发中断或 DMA 请求)。这就避免了在捕获事件时,清除计数器值时产生更新和捕获中断。

当有更新事件(UEV)产生时,预载寄存器值会更新到影子寄存器,更新标志位(**GP16C4Tn_RIF** 寄存器中的 **UI** 位)置位(取决于 **UERSEL** 位)。

注:若更新源为计数器上溢,自动重载会在计数器重载前更新。因此,下一周期即为预期值(计数器载入新值)。

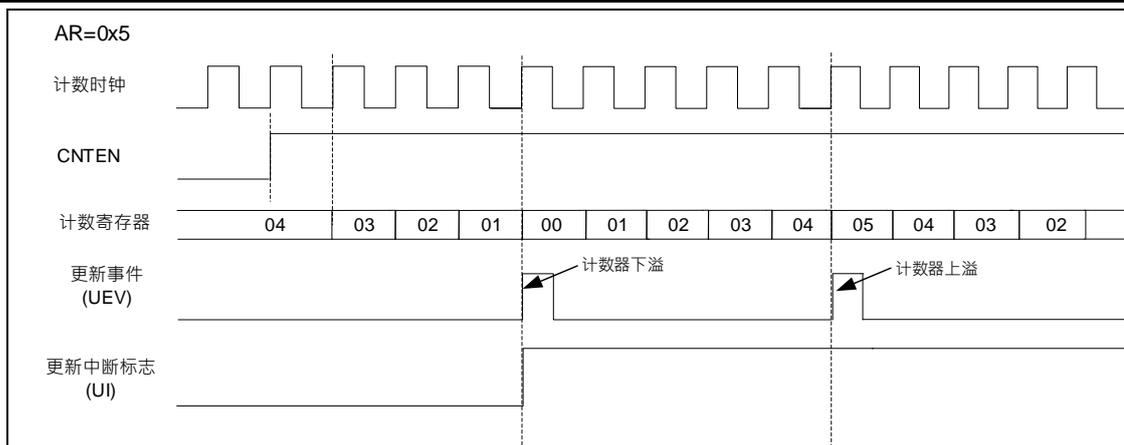


图 16-11 增减计数器时序图

16.4.4 捕获/比较通道

输入电路对 I_n 输入端的信号进行采样，产生一个经过滤波的信号 I_nF 。之后，一个可极性选择的边沿检测器产生 I_n 边沿检测信号，该信号可作为从模式控制器的触发输入或作为捕获控制命令，且信号经过分频后进入捕获寄存器。

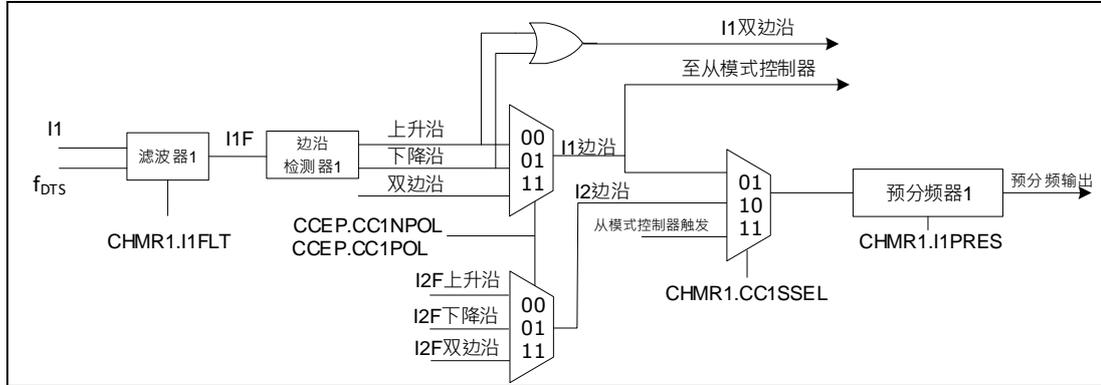


图 16-12 捕获/比较通道

输出部分产生一个中间波形（高有效）作为基准，在输出末端决定最终输出信号的极性。

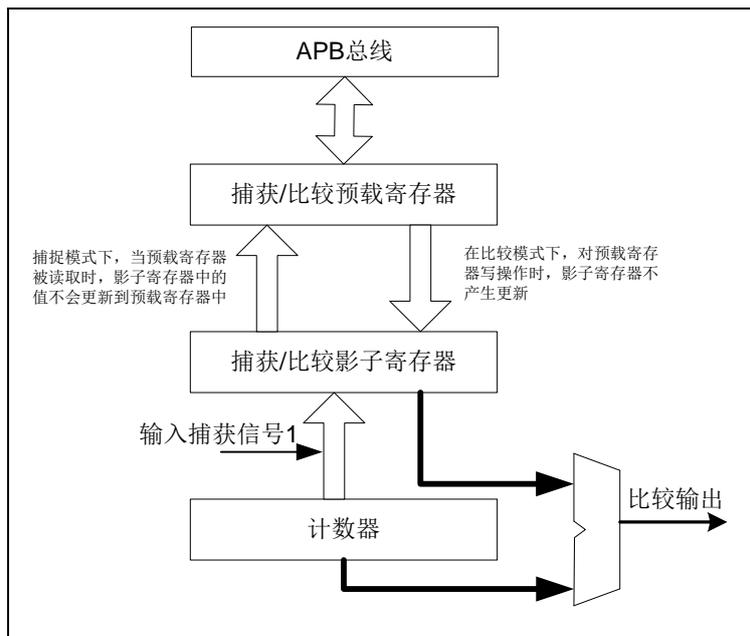


图 16-13 捕获/比较通道 1 结构图

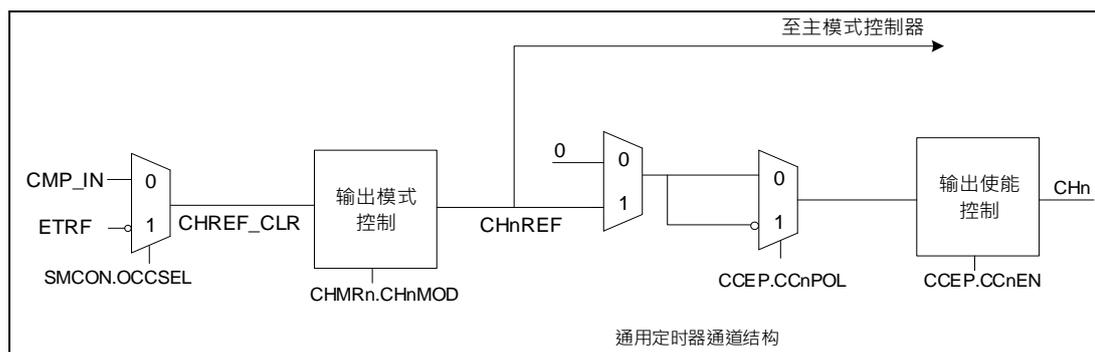


图 16-14 捕获/比较通道 1 结构图

16.4.5 输入捕获模式

在输入捕获模式下，当检测到 In 上相应信号变化时，计数器的值就会被锁存到捕获/比较寄存器(GP16C4Tn_CCVALn)寄存器中。当捕获发生时，相应的 CHnI 标志位(GP16C4Tn_RIF)会置位，同时会触发中断或 DMA（如果使能）请求。若发生捕获时，CHnI 标志位已经置位，则过捕获 CHnOVI 标志位(GP16C4Tn_RIF)置位。软件于 GP16C4Tn_ICR 对 CHnI 与 CHnOVI 位写'1'可以复位 CHnI 标志位与 CHnOVI 标志位。

以下为以 I1 输入上升沿作为捕获输入时的流程：

1. 选择有效输入端：GP16C4Tn_CCVAL1 必须连接到 I1 输入端，因此需将 GP16C4Tn_CHMR1 寄存器中的 CC1SSEL 位写"01"。只要 CC1SSEL 不为"00"，信道被配置为输入且 GP16C4Tn_CCVAL1 寄存器为只读。
2. 根据定时器连接的输入信号，配置输入滤波器的持续时间。当输入信号翻转时，前 5 个内部时钟信号内信号是不稳定的，因此必须配置滤波器的时间大于 5 个时钟周期。当 I1 检测到新的电平，连续 8 次采样可确认电平变化有效。
3. 选择 I1 信道的有效边沿变换。GP16C4Tn_CCEP 寄存器中的 CC1POL 写'0'(上升沿)。
4. 配置输入预分频器。
5. 置位 GP16C4Tn_CCEP 寄存器中的 CC1EN 位，使能捕获计数器的值到捕获寄存器。
6. 如有需要，置位 GP16C4Tn_IER 寄存器中的 CH1I 位，使能中断请求。置位 GP16C4Tn_DMAEN 寄存器中的 CH1DE 位，使能 DMA 请求。

当发生输入捕获时:

1. 有效边沿产生, **GP16C4Tn_CCVAL1** 寄存器获取计数器的值。
2. **CH1I** 标志位置位 (中断标志)。若至少 2 个连续的捕获发生, 但标志位没有及时清除, 则 **CH1OVI** 也会置位。
3. 中断的产生取决于 **GP16C4Tn_IER** 寄存器中的 **CH1I** 位。
4. **DMA** 请求的产生取决于 **GP16C4Tn_DMAEN** 寄存器中的 **CH1DE** 位。

为了处理捕获溢出, 建议在读取过捕获标志位前先读取捕获数据。这可以避免错过读过捕获标志位之后, 读之前产生的捕获数据。

注: 捕获中断请求可由软件设置 **GP16C4Tn_SGE** 寄存器中 **SGCHn** 位产生。

16.4.5.1 PWM 输入模式

测量 I1 上 PWM 信号的周期和占空比的过程如下:

1. 为 **GP16C4Tn_CCVAL1** 选择有效的输入: **GP16C4Tn_CHMR1** 寄存器中的 **CC1SSEL** 位写"01" (I1 被选择)。
2. 为 I1 边沿检测选择有效的极性 (用于捕获数据到 **GP16C4Tn_CCVAL1** 寄存器和计数器清零): **CC1POL** 位写'0' (上升沿有效)。
3. 为 **GP16C4Tn_CCVAL2** 选择有效输入: **GP16C4Tn_CHMR1** 寄存器的 **CC2SEL** 位写"10" (I1 被选择)。
4. 为 I1 边沿检测选择有效极性(用于捕获数据到 **GP16C4Tn_CCVAL2**): **CC2POL** 位写'1'。
5. 选择有效的触发输入: **GP16C4Tn_SMCON** 寄存器的 **TSSEL** 位写"101" (I1 边沿检测被选择)。
6. 配置从机模式控制器为复位模式: **GP16C4Tn_SMCON** 寄存器的 **SMODS** 位写"100"。
7. 使能捕获: **GP16C4Tn_CCEP** 寄存器的 **CC1EN** 位和 **CC2EN** 位写'1'。

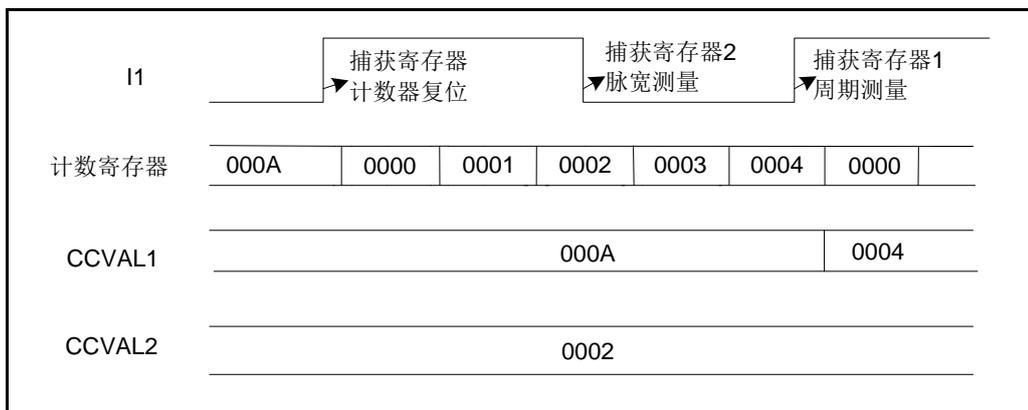


图 16-15 PWM 输入模式时序

16.4.6 PWM 模式

脉宽调制模式可以产生一个 **GP16C4Tn_AR** 寄存器值确定频率, **GP16C4Tn_CCVALn** 寄存器值确定占空比的信号。

每个信道的 PWM 模式是相互独立的（每个 CHn 输出一个 PWM），GP16C4Tn_CHMRn 寄存器的 CHnMOD 位写"110"（PWM 模式 1）或写"111"（PWM 模式 2）。必须通过置位 GP16C4Tn_CHMRn 寄存器的 CHnPEN 位来使能相应的预载寄存器，最后还需置位 GP16C4Tn_CON1 寄存器的 ARPEN 位来使能自动重装预载功能。

只有当更新事件发生时预载寄存器中的值才会传到影子寄存器，因此，在使能计数前，必须通过置位 GP16C4Tn_SGE 寄存器的 SGU 位来初始化所有的寄存器。

CHn 的极性可通过 GP16C4Tn_CCEP 寄存器的 CCnPOL 位配置，有效极性可配置为高或低。CHn 的输出使能由 CCnEN、CCnNEN 位（GP16C4Tn_CCEP 寄存器）组合控制。

在 PWM 模式（1 或 2）中，GP16C4Tn_COUNT 和 GP16C4Tn_CCVALn 寄存器的值会持续比较，确定 GP16C4Tn_CCVALn <= GP16C4Tn_COUNT 或 GP16C4Tn_CCVALn >= GP16C4Tn_COUNT（取决于计数器的计数方向）。

定时器产生 PWM 波形是边沿对齐或中央对齐，取决于 GP16C4Tn_CON1 寄存器的 CMSEL 位。

16.4.6.1 PWM 边沿对齐模式

- ◇ GP16C4Tn_AR=8
- ◇ PWM 模式 1
- ◇ 递增计数配置

当 GP16C4Tn_CON1 寄存器的 DIRSEL 位为低时，计数器递增计数。

下图给出了 GP16C4Tn_AR = 8 时的边沿对齐 PWM 波形。

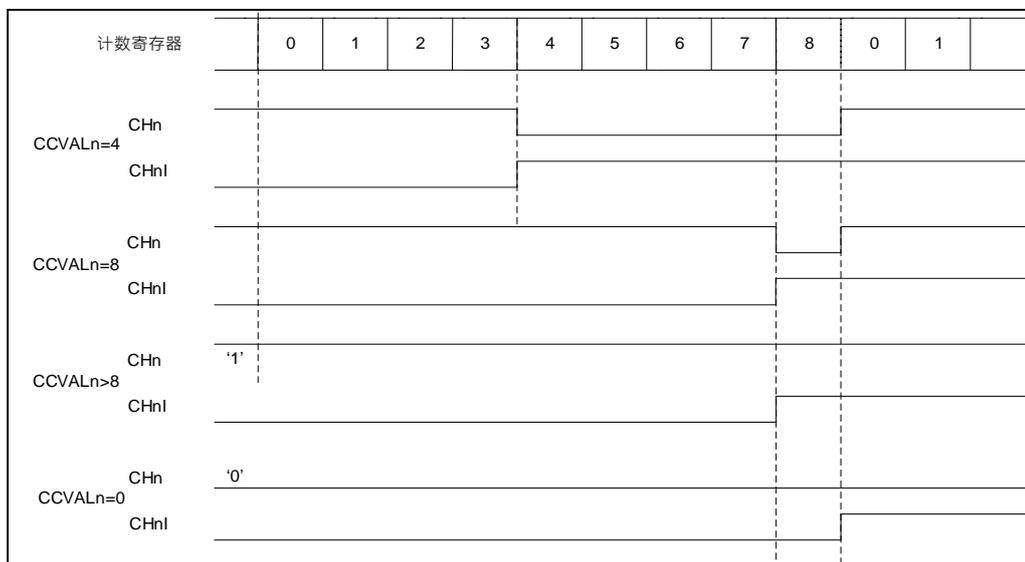


图 16-16 边沿对齐递增计数 PWM 波形 (AR=8)

- ◇ GP16C4Tn_AR=8
- ◇ PWM 模式 1
- ◇ 递减计数配置

当 GP16C4Tn_CON1 寄存器的 DIRSEL 位为高时，计数器递减计数。

下图给出了 GP16C4Tn_AR = 8 时的边沿对齐 PWM 波形。

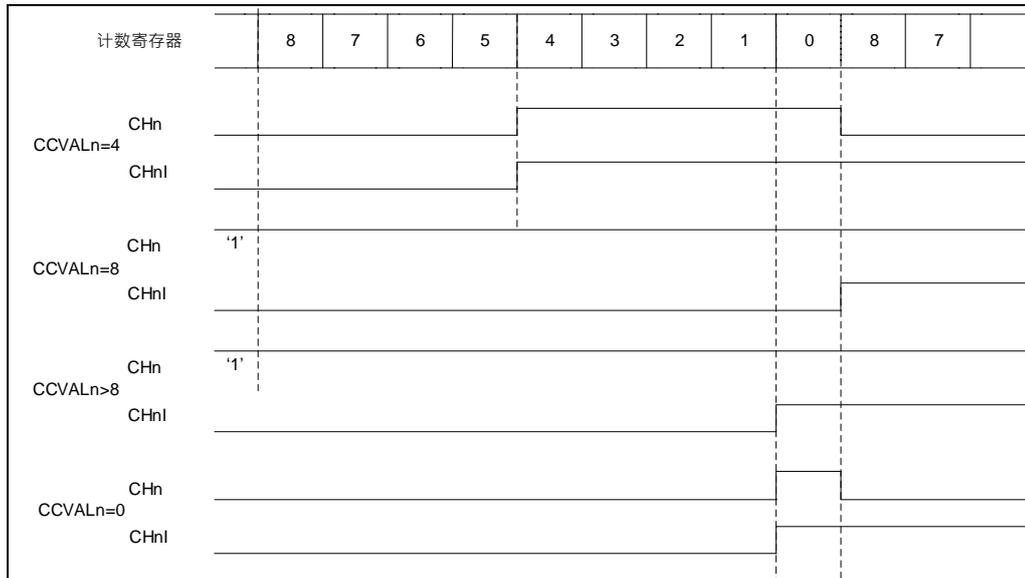


图 16-17 边沿对齐递减计数 PWM 波形 (AR=8)

16.4.6.2 PWM 中央对齐模式

当 GP16C4Tn_CON1 寄存器中的 CMSEL 位不为"00"时，中央对齐模式有效。计数器是递增、递减计数分别置比较标志位或递增递减都置比较标志位，取决于 CMSEL 位的配置。

GP16C4Tn_CON1 寄存器的方向位 (DIRSEL) 是由硬件更新的，软件无法修改。

下图为中央对齐方式产生的 PWM 波形的例子：

- ◇ GP16C4Tn_AR=0x3F
- ◇ PWM 模式 1
- ◇ GP16C4Tn_CON1 寄存器的 CMSEL="10"，在中央对齐模式 2 下，计数器向上计数时会置位比较标志位。

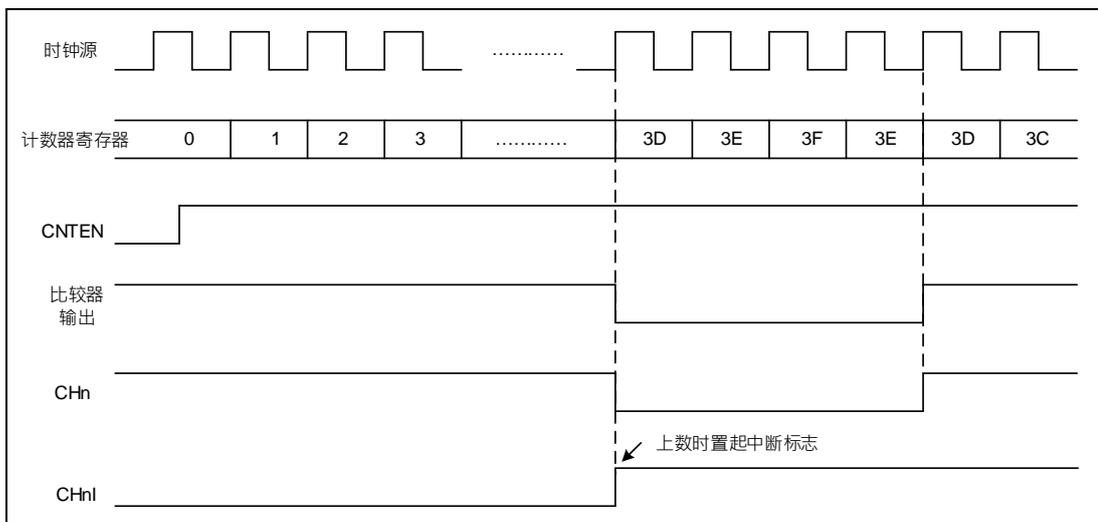


图 16-18 中央对齐 PWM 波形 (AR=0x3F)

中央对齐模式的使用技巧:

- ◇ 当进入中央对齐模式后, 当前递增或递减配置生效。计数器递增或递减计数取决于 **GP16C4Tn_CON1** 寄存器的 **DIRSEL** 位的值。此外, 软件无法对 **DIRSEL** 和 **CMSEL** 位同时进行修改。
- ◇ 计数器在中央对齐模式下运行时, 对计数器写操作可能导致不可预知的结果。特别是:
 - 若向计数器入的值大于自动重载值 ($GP16C4Tn_COUNT > GP16C4Tn_AR$), 计数方向不更新。例如, 如果计数器递增计数, 写入值后从 0 重新递增计数。
 - 若向计数器写 0 或 **GP16C4Tn_AR** 中的重载值, 则计数方向更新, 但并没有产生 **UEV**。
- ◇ 使用中央对齐模式最安全的方式是计数器开始计数前通过软件产生更新事件 (置位 **GP16C4Tn_SGE** 寄存器中的 **SGU** 位) 且在计数器运行过程中不对计数器写值。

16.4.7 输出比较模式

该功能用于控制输出波形或指示周期时间的结束。

当捕获/比较寄存器和计数器值匹配时，输出比较功能：

- ◇ 输出比较模式（**GP16C4Tn_CHMRn** 寄存器中的 **CHnMOD** 位）和输出极性（**GP16C4Tn_CCEP** 寄存器中的 **CCnPOL** 位）的配置值输出到对应的引脚上。
- ◇ 中断状态寄存器中的标志位置位（**GP16C4Tn_RIF** 寄存器的 **CHnI** 位）。
- ◇ 若相应的中断使能置位，则产生中断（**GP16C4Tn_IER** 寄存器的 **CHnI** 位）。
- ◇ 若相应的使能位置位（**GP16C4Tn_DMAEN** 寄存器的 **CHnDE** 位，**GP16C4Tn_CON2** 寄存器的 **CCDMASEL** 位用于 DMA 请求的选择），则发送 DMA 请求。

GP16C4Tn_CHMRn 寄存器中 **CHnPEN** 位的值可决定 **GP16C4Tn_CCVALn** 寄存器是否带有预装载寄存器。

在输出比较模式中，更新事件 **UEV** 对 **CHn** 的输出没有影响。计时分辨率为计数器的一次计数。输出比较模式同样可以用来输出单个脉冲（单脉冲模式）。

输出比较的配置过程：

1. 选定计数器时钟（内部，外部，预分频）。
2. **GP16C4Tn_AR** 与 **GP16C4Tn_CCVALn** 寄存器中写入预期值。
3. 若需要产生中断请求，置位 **GP16C4Tn_IER** 寄存器中的 **CHnI** 位。
4. 选择输出模式，例如：
 - **CHnMOD** = "011"，当 **CNTV** 与 **CCRVALn** 匹配时，**CHn** 输出翻转。
 - **CHnPEN** = '0'，关闭预载寄存器。
 - **CCnPOL** = '0'，选择有效极性为高。
 - **CCnEN** = '1'，使能输出。
5. **GP16C4Tn_CON1** 寄存器中的 **CNTEN** 位置位，使能计数器。

假设预载寄存器没有使能（**CHnPEN** = '0'，否则 **GP16C4Tn_CCVALn** 影子寄存器只有在下次更新事件发生时才更新）。通过软件方式，**GP16C4Tn_CCVALn** 寄存器的值可随时更新控制输出波形。

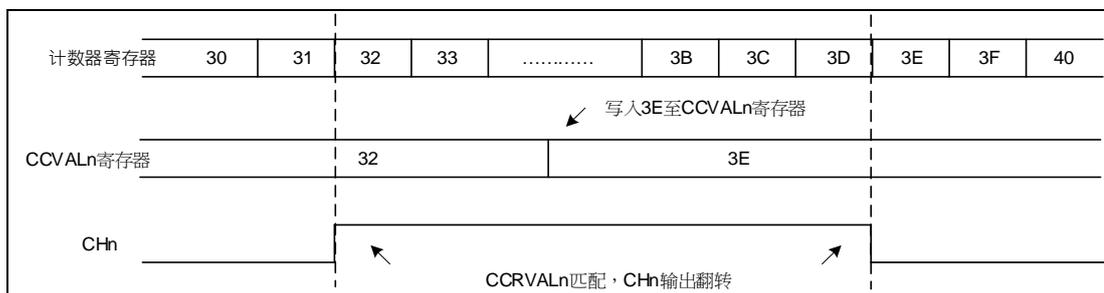


图 16-19 输出比较模式，触发 **CHn**

16.4.7.1 外部事件清除比较输出

ETF 输入端（**GP16C4Tn_CHMRn** 寄存器的 **CHnOCLREN** 位写'1'）上的高电平，可将给定通道的比较输出信号拉低。在下次更新事件（UEV）发生前，比较输出会一直保持为低。该功能只能应用在输出比较和 PWM 模式中，强制输出模式中不起作用。

ETR 信号可以接到电流控制比较器的输出端。该例中，ETR 须按如下流程配置：

1. 外部时钟源 2 关闭：**GP16C4Tn_SMCON** 寄存器的 **ECM2EN** 位写'0'
2. 外部触发极性（ETPOL）和外部触发滤波器（ETFLT）可根据用户需要配置

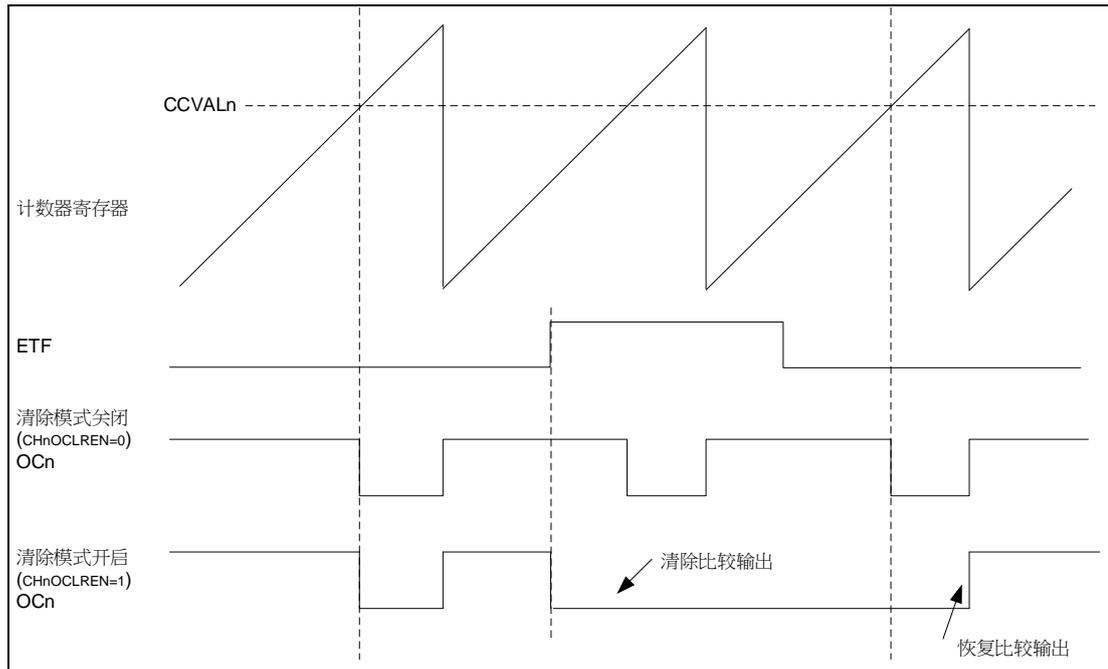


图 16-20 清除比较输出 CHn

16.4.7.2 强制输出模式

在输出模式中（**GP16C4Tn_CHMRn** 寄存器中 **CCnSSEL = "00"**），软件可强制将每个输出比较信号（**CHnN**）改为有效或无效状态，这种修改独立于输出比较寄存器和计数器的比较结果。

为了将某输出比较信号（**CHn**）强制为有效状态，需将相应的 **GP16C4Tn_CHMRn** 寄存器中 **CHnMOD** 位写"101"。因此，比较输出被强制为高（高时为有效状态）且 **CHn** 的值为 **CCnPOL** 极性位的相反值。

例如：**CCnPOL = '0'**（**CHn** 高电平有效），则 **CHn** 被强制为高电平。

对 **GP16C4Tn_CHMRn** 寄存器的 **CHnMOD** 位写"100"，比较输出可被置低。

无论怎样，**GP16C4Tn_CCVALn** 影子寄存器和计数器之间的比较仍然进行，相应的标志位仍可置位。

16.4.8 单脉冲模式

单脉冲模式 (SPMEN) 下, 响应某个触发后, 定时器的输出信道在可配置的延迟时间后产生一个脉冲, 脉冲长度可配。通过模式控制寄存器可控制计数器的启动。脉冲波形可在输出比较模式和 PWM 模式下产生。置位 **GP16C4Tn_CON1** 寄存器的 **SPMEN** 位可选择单脉冲模式。计数器会在下次更新事件 **UEV** 产生时自动停止。

只有比较值不同于计数器初始值时, 单脉冲才可以正确的产生。计数器开始计数前 (定时器等待触发), 必须如下配置:

- ◇ 递增计数: $CNTV < CCVALn \leq AR$ (特别地, $0 < CCVALn$)
- ◇ 递减计数: $CNTV > CCVALn$

基于 PWM 模式设置单脉冲输出波形的步骤如下:

- ◇ 设置 **GP16C4Tn_CHMRn** 寄存器的 **CHnMOD** 位, 选择 PWM 模式 1 或 2;
- ◇ 设置 **GP16C4Tn_CCEPn** 寄存器的 **CCnPOL** 位, 选择通道端口 **CHn** 的输出极性;
- ◇ 设置 **GP16C4Tn_CON1** 寄存器的 **DIRSEL**, **CMSEL**, **SPMEN** 位, 配置为递增或递减计数, PWM 普通波形模式, 单脉冲模式使能;
- ◇ 设置 **GP16C4Tn_CHMR** 寄存器的 **CH1PEN = 1**, **GP16C4Tn_CON1** 寄存器的 **ARPEN = 1**, 使能比较寄存器和计数重载寄存器的缓冲功能 (也可以根据实际情况不使能缓冲);
- ◇ 设置 **GP16C4Tn_CCVALn** 寄存器和 **GP16C4Tn_AR** 寄存器, 配置单脉冲输出延时和脉宽时间;
- ◇ 设置 **GP16C4Tn_SGE** 寄存器的 **SGU** 位来产生一个更新事件;
- ◇ 设置 **GP16C4Tn_CON1** 寄存器的 **CNTEN=1** 来启动计数器, 也可以在触发模式下, 通过外部触发输入信号来触发硬件自动设置 **CNTEN=1**。

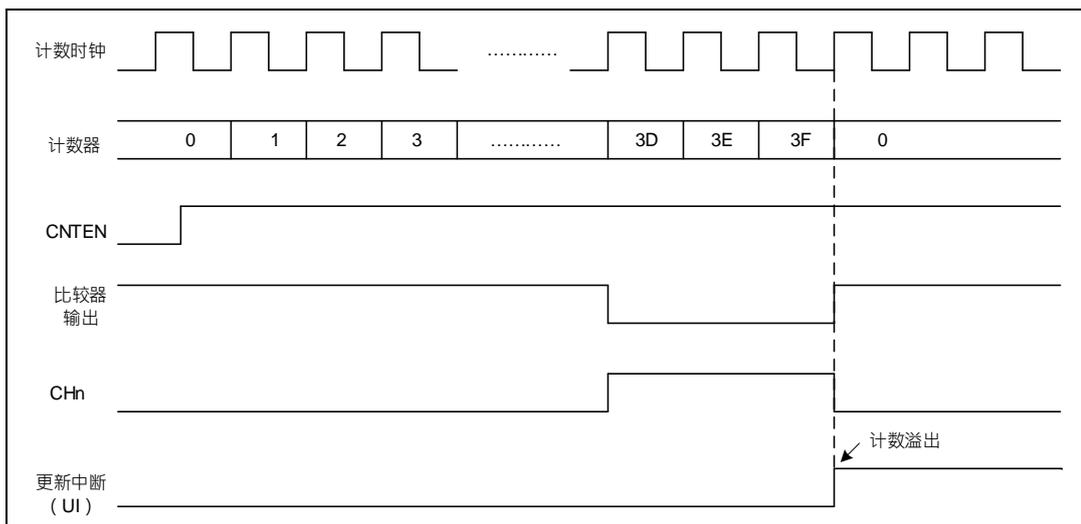


图 16-21 单脉冲模式

16.4.9 编码器接口模式

编码器接口模式的三种配置：若计数器只根据 I2 上的边沿计数，则 **GP16C4Tn_SMCON** 寄存器中的 **SMODS = "001"**；若计数器只根据 I1 上的边沿计数，则 **GP16C4Tn_SMCON** 寄存器中的 **SMODS = "010"**；若计数器同时根据 I1 和 I2 上的边沿计数，则 **GP16C4Tn_SMCON** 寄存器中的 **SMODS = "011"**。

配置 **GP16C4Tn_CCEP** 寄存器中的 **CC1POL** 和 **CC2POL** 位的值可选择 I1 和 I2 的极性。如果需要，也可以配置输入滤波器。

CH1 和 **CH2** 端口作为增量编码器的接口。当计数器使能时，计数器根据 I1 或 I2 上滤波后的有效电平变化时钟计数。I1 和 I2 滤波后的有效信号顺序会产生计数脉冲及方向信号。计数器是递增或递减计数由信号的跳变顺序决定，**GP16C4Tn_CON1** 寄存器中的 **DIRSEL** 计数方向位由硬件自动更新。

编码器接口模式的工作方式类似于一个带有方向选择的外部时钟。计数器在 0 到 **GP16C4Tn_AR** 寄存器中的自动重载值之间连续计数。因此，必须在开始计数前配置 **GP16C4Tn_AR** 寄存器。同样的，捕获器、预分频器、重复计数器、触发输出的特性正常工作。设定编码模式和选择外部时钟源 2 不兼容，不可以同时选择。

该模式下，计数器会根据增量式编码器的速度和方向自动修改，计数器的值反映的是编码器的位置。计数方向对应着连接传感器的旋转方向。

下表列出了所有的可能组合，假设 I1 和 I2 不同时变换。

有效边沿	有效边沿相对信号的电平 (I1 滤波信号对应 I2,I2 滤波信号对应 I1)	I1 滤波信号		I2 滤波信号	
		上升	下降	上升	下降
仅在 I1 计数	高	下降	上升	不计数	不计数
	低	上升	下降	不计数	不计数
仅在 I2 计数	高	不计数	不计数	上升	下降
	低	不计数	不计数	下降	上升
在 I1 和 I2 上计数	高	下降	上升	上升	下降
	低	上升	下降	下降	上升

表 16-1 计数方向与编码器信号的关系

外部增量编码器可直接与 MCU 连接，无需外部接口逻辑。而比较器通常用于将编码器的差分输出转换为数字信号，这极大地增加了抗噪声能力。编码器的第三个输出端用于指示机械零点，可以连接到外部中断输入引脚以触发一次计数复位。

下图给出了计数器操作的例子，给出了计数信号的产生和方向控制。同样给出了选择双边沿时，输入抖动如何被补偿。输入抖动可能发生在传感器靠近切换点处。

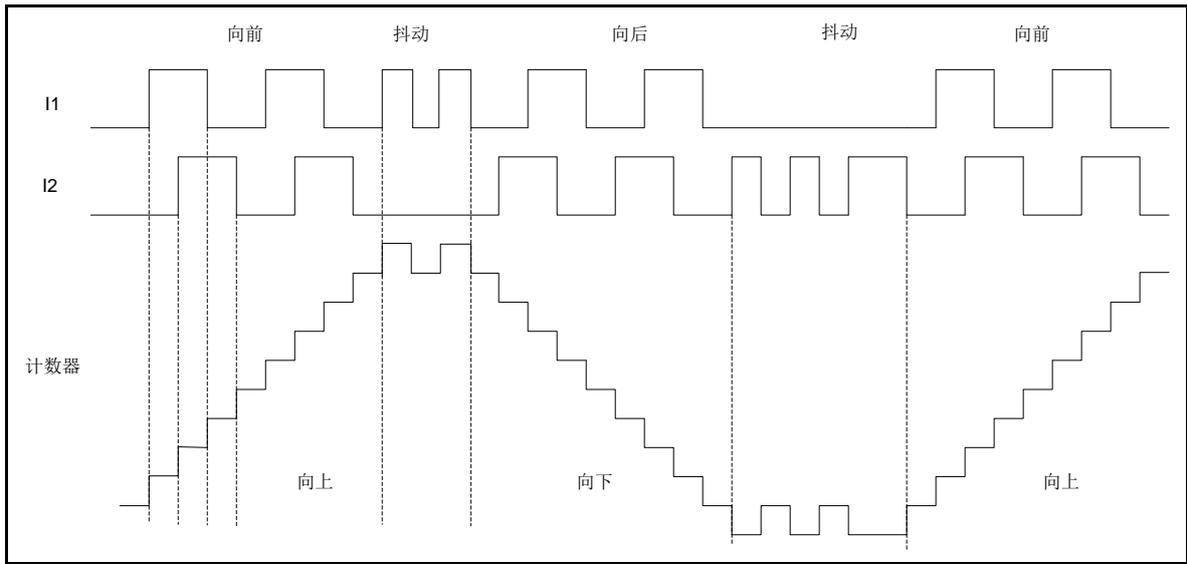


图 16-22 编码器接口模式下的计数操作

下图给出了计数器在 I1 滤波信号极性反相时的计数过程（除了 CC1POL = '1'，其他配置与上面一致）

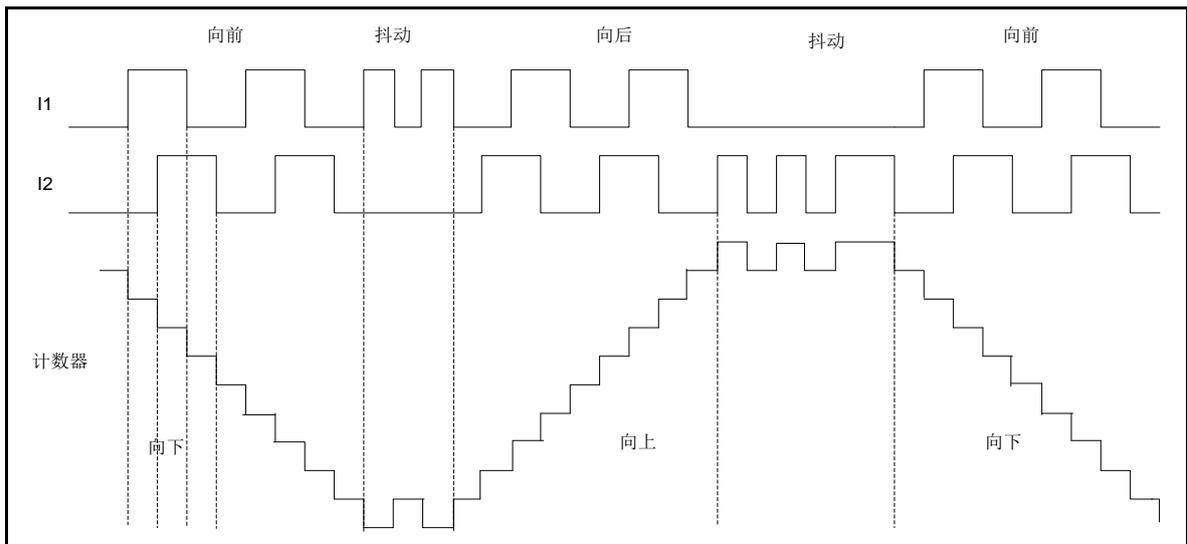


图 16-23 滤波后极性反相时编码器接口例子

当配置为编码器接口模式时，定时器可提供传感器的当前位置讯息。配置一个额外定时器为捕获模式，用于测量两个编码器事件的间隔，根据间隔时长获取动态讯息（速度、加速度、减速度）。编码器用于指示机械零点的输出就是此用处。根据编码器两个事件间隔，可以周期性的读取计数器的值。如果允许，可以将计数器值锁存到第三个输入捕获寄存器（捕获信号必须是周期性的且可由其它定时器产生）。条件允许时，可通过实时时钟产生 DMA 请求的方式读取计数器值。

16.4.10 输入异或功能

通过 **GP16C4Tn_CON2** 寄存器中 **I1SEL** 位，可将通道 1 的输入滤波器连接到 XOR 门的输出端，XOR 门联合了 CH1、CH2 和 CH3 三个输入引脚。

XOR 输出用于定时器的所有输入功能，如触发或输入捕获。该功能参见下节的霍尔传感器接口。

16.4.11 霍尔传感器接口

使用高级控制定时器产生 PWM 信号驱动马达，用另一个定时器作为“接口定时器”来连接霍尔传感器，请参见下图。3 个定时器输入脚(CH1、CH2、CH3)通过一个异或门连接到 I1 输入通道(通过设置 **GP16C4Tn_CON2** 寄存器中的 **I1SEL** 位来选择)，“接口定时器”捕获这个信号。

从模式控制器被配置为复位模式，从输入是 I1F 双边沿。这样每当 3 个输入之一变化时，计数器从 0 重新开始计数。由此产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”模式下，捕获/比较信道 1 被配置为捕获模式，捕获信号为 I1(捕获/比较通道(如：通道 1 输入部分))。捕获值反映了输入端两次变化之间的时间间隔，指示出了马达速度的讯息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级控制定时器 (AD16C4Tn) 各个通道的属性，AD16C4Tn 产生 PWM 信号驱动马达。因此，“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGOUT 输出被送到高级控制定时器 (AD16C4Tn)。

举例：霍尔输入连接到定时器，要求在每次霍尔输入发生变化之后的一个指定的时刻，改变高级控制定时器的 PWM 配置。

- ◇ 置 **GP16C4Tn_CON2** 寄存器的 **I1SEL** 位为'1'，配置三个定时器输入逻辑异或到 I1 输入，
- ◇ 时基编程：置 **GP16C4Tn_AR** 为其最大值(计数器必须通过 I1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- ◇ 设置信道 1 为捕获模式(选中 I1)：置 **GP16C4Tn_CHMR1** 寄存器中 **CC1SSEL=01**，如果需要，还可以设置数字滤波器。
- ◇ 设置信道 2 为 PWM 模式 2，带指定的延时：置 **GP16C4Tn_CHMR1** 寄存器中的 **CH2OMOD =111** 和 **CC2SEL =00**。
- ◇ 选择 **CH2REF** 作为 TRGOUT 上的触发输出：置 **GP16C4Tn_CON2** 寄存器中的 **MMSEL=101**。

在高级控制定时器 AD16C4Tn 中，正确的 IT 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(**AD16C4Tn_CON2** 寄存器中 **CCPCNTEN=1**)，由触发输入控制 COM 事件(**AD16C4Tn_CON2** 寄存器中 **CCUSEL=1**)。在一次 COM 事件后，下一步再写入 PWM 控制位(**CCnEN**、**CHnOCLREN**)，这可以在处理 **CH2REF** 上升沿的中断子程序里实现。

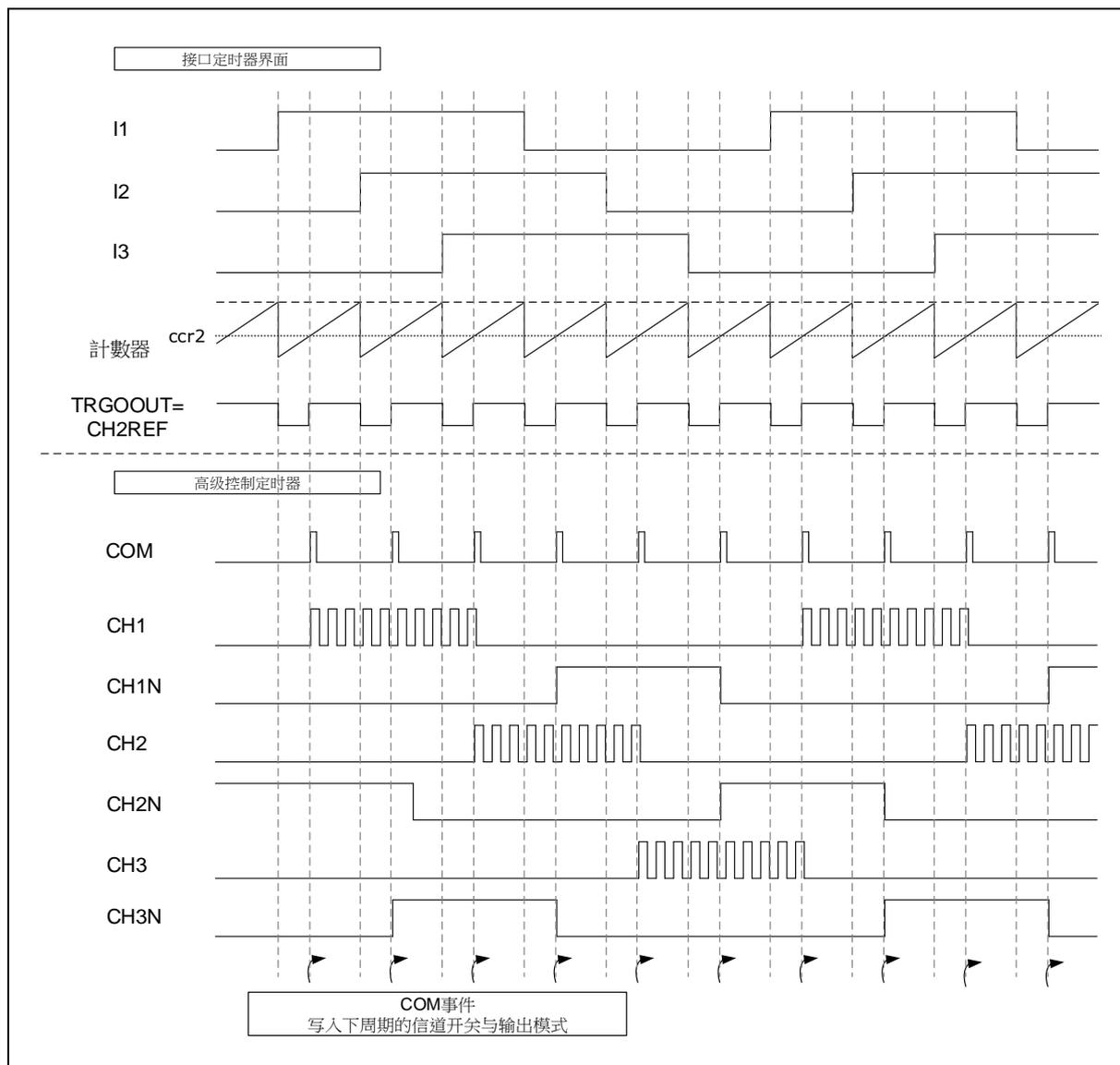


图 16-24 霍尔传感器接口范例

16.4.12 外部触发的同步

GP16C4Tn 定时器可在多种模式下与外部触发同步：复位模式、门控模式及触发模式。

16.4.12.1 复位模式

计数器及其预分频器可以在响应触发输入事件时重新初始化。此外，若 **GP16C4Tn_CON1** 寄存器的 **UERSEL** 位为低时会产生一次更新事件 **UEV**。所有预载寄存器 (**GP16C4Tn_AR**, **GP16C4Tn_CCVALn**) 都会因更新事件 **UEV** 而被更新。

在下面例子中，I1 输入端的上升沿让递增计数被清空：

- ◇ 配置信道 1 上检测 I1 上的上升沿。配置输入滤波周期（本例无需滤波器，故 **I1FLT** = "0000"）。触发捕获分频器没有使用，无需配置。**CC1SSEL** 位只选择输入捕获源，**GP16C4Tn_CHMR1** 寄存器中 **CC1SSEL** = "01"。**GP16C4Tn_CCEP** 寄存器中 **CC1POL** = 0 以确定极性（只检测上升沿）。
- ◇ 定时器配置为复位模式：**GP16C4Tn_SMCON** 寄存器中 **SMODS** = "100"。选择 I1 作为输入源：**GP16C4Tn_SMCON** 寄存器中 **TSSEL** = "101"。
- ◇ 启动计数器：**GP16C4Tn_CON1** 寄存器中 **CNTEN** = '1'。

计数器依据内部时钟开始计数，正常计数直到 I1 上出现上升沿。当 I1 上出现上升沿时，计数器会被清零且从 0 重新开始计数。同时，标志位置位（**GP16C4Tn_RIF** 寄存器中 **TRGI** 位），如果中断及 DMA 使能（取决于 **GP16C4Tn_IER** 寄存器中的 **TRGI** 和 **GP16C4Tn_DMAEN** 寄存器中的 **TRGIDE** 位），会发送中断及 DMA 请求。

下图给出了当自动重载寄存器 **GP16C4Tn_AR** = 0x36 时的信号变化。由于 I1 输入的再同步电路，I1 上的上升沿和计数器实际复位之间会存在延时。

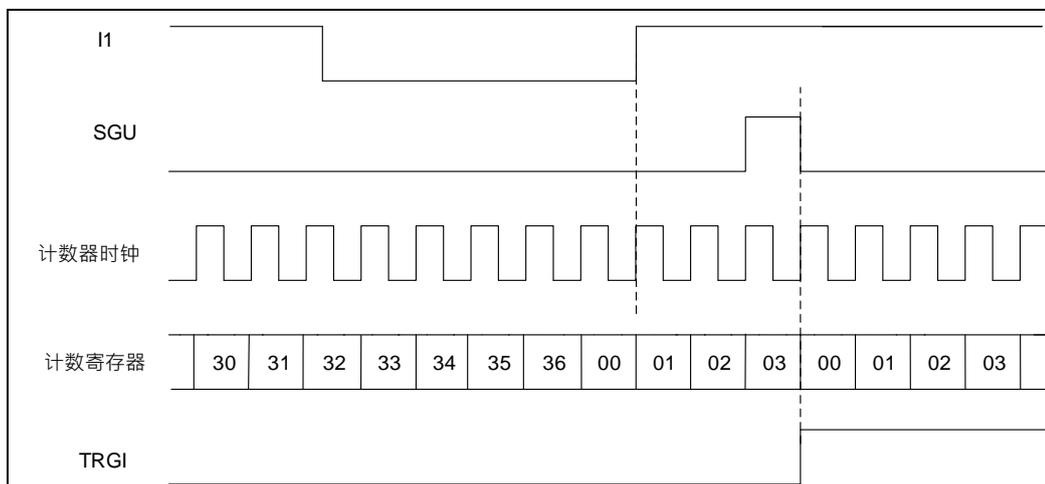


图 16-25 复位模式控制电路

16.4.12.2 门控模式

计数器根据选中的输入电平被使能。

下面的例子中，计数器只在 I1 输入为低电平时才递增计数：

- ◇ 配置信道 1 在 I1 上检测低电平。配置输入滤波周期(本例不需要滤波器, I1FLT = "0000")。触发捕获分频器没有使用, 无需配置。GP16C4Tn_CHMR1 寄存器中的 CC1SSEL = "01", 选择输入捕获源。GP16C4Tn_CCEP 寄存器中 CC1POL = '1', 确认极性(只检测低电平)。
- ◇ 配置定时器为门控模式: GP16C4Tn_SMCON 寄存器中 SMODS = "101"。选择 I1 作为输入源: GP16C4Tn_SMCON 寄存器中 TSSEL = "101"。
- ◇ 使能计数器: GP16C4Tn_CON1 寄存器中 CNTEN = '1' (门控模式中, 如果 CNTEN = '0', 无论触发输入为何电平, 计数器都不会启动)。

只要 I1 为低电平, 计数器依据内部时钟开始计数, 一旦 I1 为高则停止计数。由于 I1 输入端的再同步电路的原因, I1 上出现上升沿和计数器实际停止之间会有一定的延时。

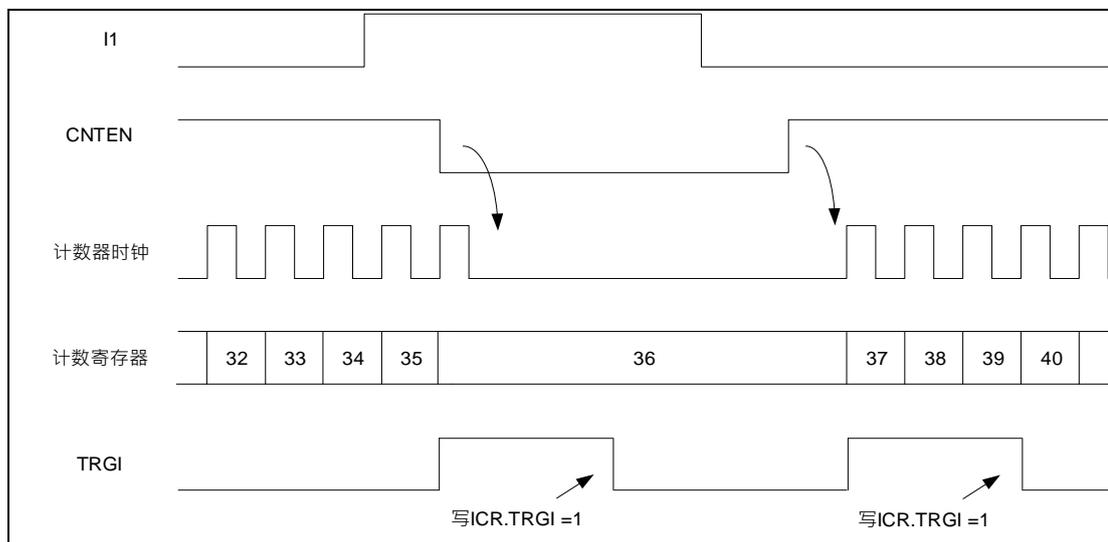


图 16-26 门控模式控制电路

16.4.12.3 触发模式

输入端选中的事件可以使能计数器。

下面的例子中，I2 输入端上的上升沿可以启动递增计数：

- ◇ 配置信道 2 可以检测 I2 上的上升沿。配置滤波时间（本例不需要滤波，I2FLT = "0000"）。触发捕获分频器没有使用，无需配置。GP16C4Tn_CHMR1 寄存器中 CC2SEL = "01", 用于选择捕获源。GP16C4Tn_CCEP 寄存器中 CC2POL = '0', 确认极性（只检测高电平）。
- ◇ 配置定时器为触发模式：GP16C4Tn_SMCON 寄存器中 SMODS = "110"。GP16C4Tn_SMCON 寄存器中 TSSEL = "110", 用于选择输入源。

I2 上出现上升沿时，计数器开始依据内部时钟计数并置位 TRGI 标志位。

由于 I2 输入的再同步原因，I2 上出现上升沿和计数器实际停止之间会有一定的延时。

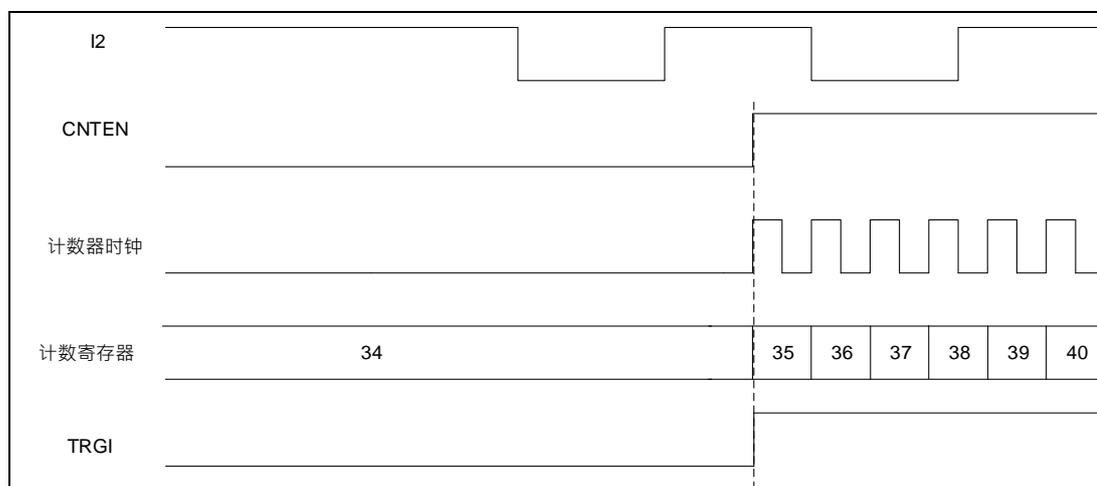


图 16-27 触发模式控制电路

16.4.12.4 选择外部时钟源 2 的触发模式

外部时钟源 2 可和其他模式一起使用（除编码模式）。ETR 信号可作为外部时钟输入，另一个输入可选择为触发输入（复位模式、门控模式或触发模式）。不推荐用 GP16C4Tn_SMCON 寄存器的 TSSEL 位选择 ETR 作为 TI。

下面的例子中，一旦 I1 上出现上升沿时，计数器会依据 ETR 信号的每个上升沿递增计数。

- ◇ 通过 GP16C4Tn_SMCON 寄存器，配置外部触发输入电路，过程如下：

ETFLT = "000": 无滤波

ETPOL = '0': 检测 ETR 的上升沿，ECM2EN = '1'使能外部时钟模式 2

- ◇ 配置信道 1 检测 I 的上升沿，过程如下：

I1FLT = "0000": 无滤波。

触发捕获分频器没有使用，无需配置。

GP16C4Tn_CHMR1 寄存器中 CC1SSEL = "01"选择输入捕获源，GP16C4Tn_CCEP 寄存器的 CC1POL = '0'确认极性（只检测上升沿）。

- ◇ 配置定时器为触发模式：GP16C4Tn_SMCON 寄存器中 SMODS = "110"。

GP16C4Tn_SMCON 寄存器中 TSSEL = "101"选择 I1 作为输入源。

I1 上出现上升沿时，计数器使能且 TRGI 标志位置位，然后计数器根据 ETR 上的上升沿开始计数。

由于 ETF 输入再同步电路的原因，ETR 信号的上升沿和实际计数器的复位会有延时。

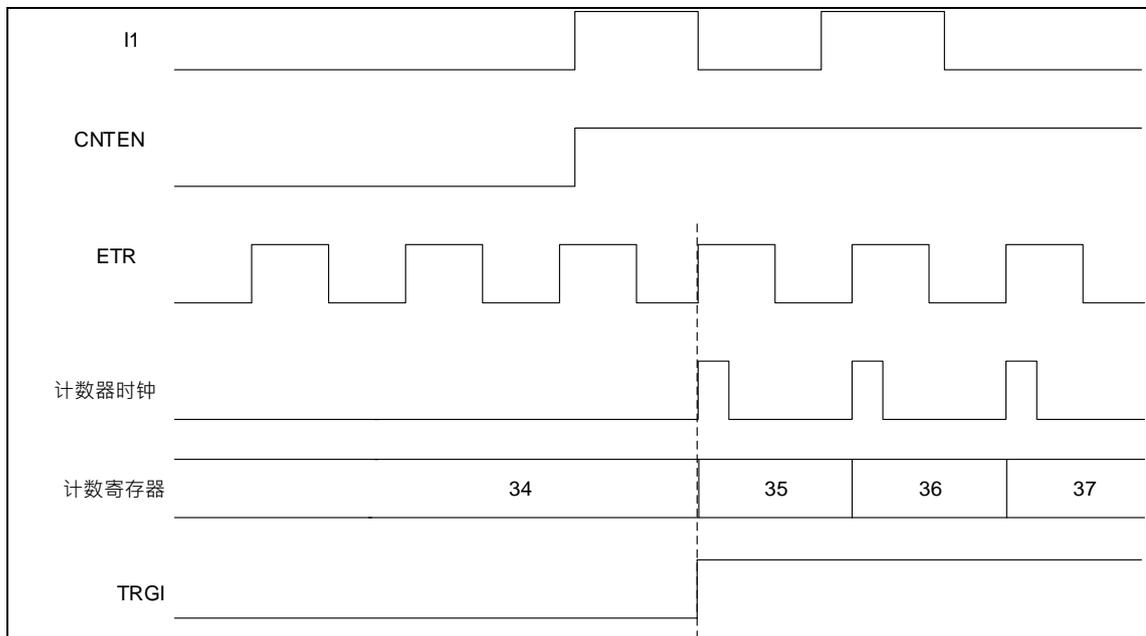


图 16-28 外部时钟源 2+触发模式下的控制电路

16.4.13 定时器同步

所有定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况

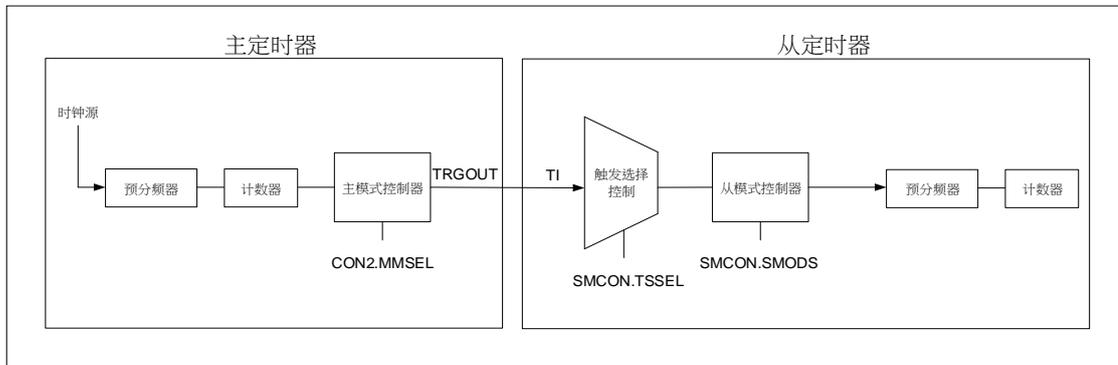


图 16-29 主/从定时器范例

16.4.13.1 使用一个定时器去使能其他定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考下图的连接。只当定时器 1 的 CH1REF 为高时，定时器 2 才对分频后的内部时钟计数。

配置定时器 1 为主模式，送出它的输出比较参考信号 (CH1REF) 为触发输出 (AD16C4T1_CON2 寄存器的 MMSEL=100)

- ◇ 配置定时器 1 的 CH1REF 波形 (AD16C4T1_CHMR1 寄存器)
- ◇ 配置定时器 2 从定时器 1 获得输入触发 (GP16C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为门控模式 (GP16C4T1_SMCON 寄存器的 SMODS=101)
- ◇ 配置 GP16C4T1_CON1 寄存器的 CNTEN=1 以使能定时器 2
- ◇ 配置 AD16C4T1_CON1 寄存器的 CNTEN=1 以启动定时器 1

注：定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

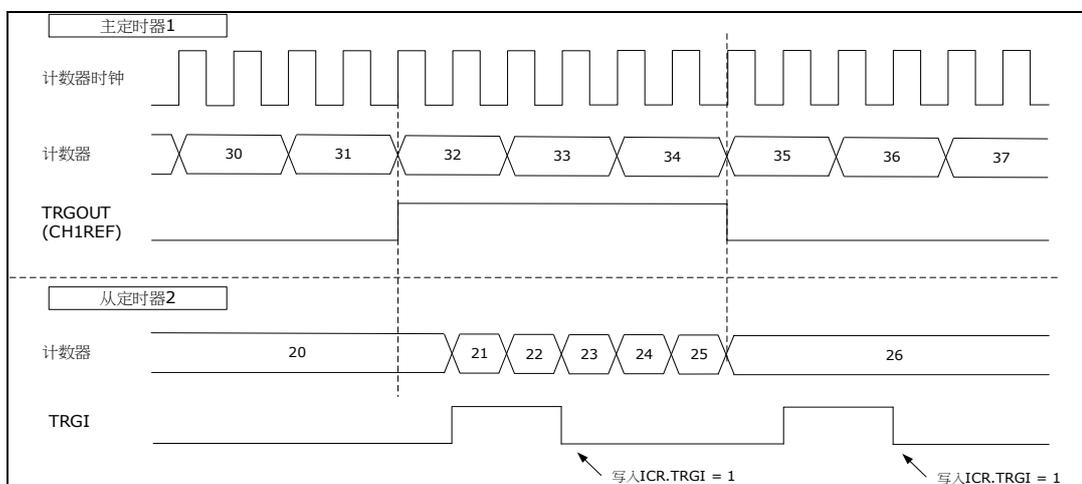


图 16-30 门控从定时器使用主定时器 CH1REF

在上图的例子中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 **AD16C4T1_SGE** 寄存器的 **SGU** 位即可复位定时器。

在下一个例子中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写'0'到 **AD16C4T1_CON1** 的 **CNTEN** 位将禁止定时器 1，定时器 2 随即停止。

配置定时器 1 为主模式，送出 **CNTEN** 位做为触发输出(**AD16C4T1_CON2** 寄存器的 **MMSEL=001**)。

- ◇ 配置定时器 2 从定时器 1 获得输入触发(**GP16C4T1_SMCON** 寄存器的 **TSSEL=000**)
- ◇ 配置定时器 2 为门控模式(**GP16C4T1_SMCON** 寄存器的 **SMODS=101**)
- ◇ 配置 **AD16C4T1_SGE** 寄存器的 **SGU=1**，复位定时器 1。
- ◇ 配置 **GP16C4T1_SGE** 寄存器的 **SGU=1**，复位定时器 2。
- ◇ 写'0xE7'至定时器 2 的计数器(**GP16C4T1_COUNT**)，初始化它为 0xE7。
- ◇ 配置 **GP16C4T1_CON1** 寄存器的 **CNTEN=1** 以使能定时器 2。
- ◇ 配置 **AD16C4T1_CON1** 寄存器的 **CNTEN=1** 以启动定时器 1。
- ◇ 配置 **AD16C4T1_CON1** 寄存器的 **CNTEN=0** 以停止定时器 1。

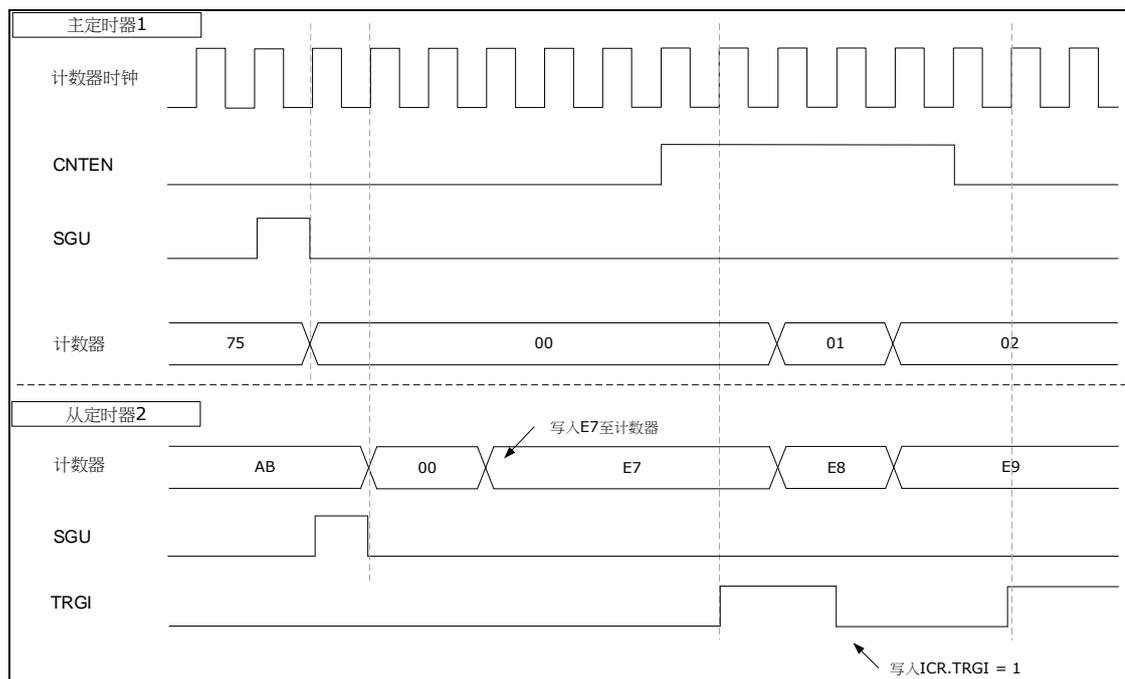


图 16-31 通过使能定时器 1 可以控制定时器 2

16.4.13.2 使用一个定时器去开启其他定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值(可以是非 0)按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CNTEN 位被自动地置'1'，同时计数器开始计数直到写'0'到 GP16C4T1_CON1 寄存器的 CNTEN 位。

- ◇ 配置定时器 1 的周期(AD16C4T1_AR 寄存器)。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP16C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为触发模式(GP16C4T1_SMCON 寄存器的 SMODS=110)
- ◇ 配置 AD16C4T1_CON1 寄存器的 CNTEN=1 以启动定时器 1。

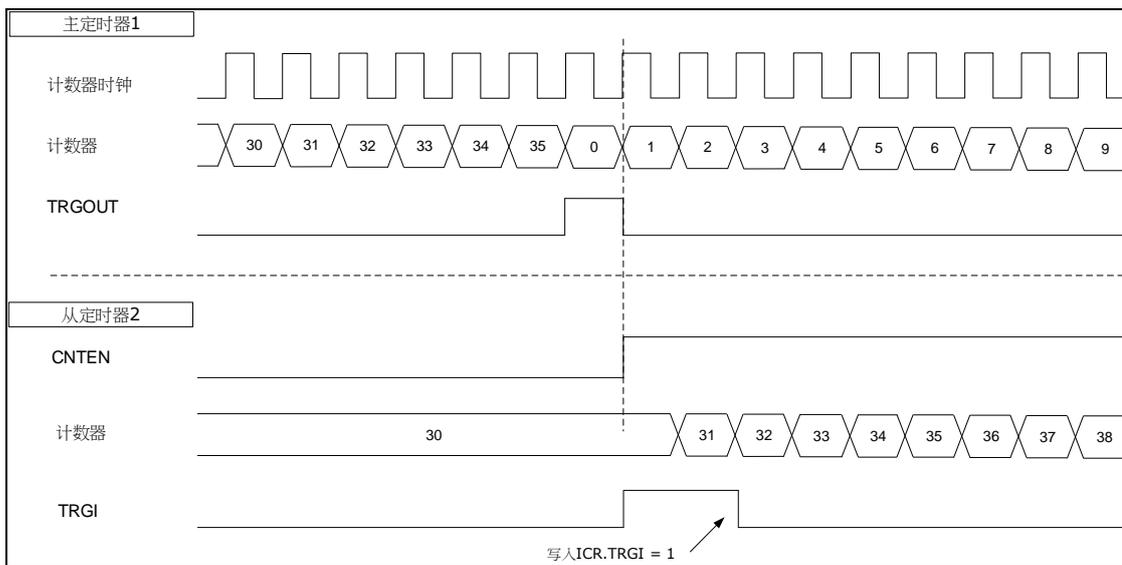


图 16-32 触发中从定时器使用主定时器更新事件

在上图例子中，可以在启动计数之前初始化两个计数器。上图显示在上上图相同配置情况下，使用触发模式而不是门控模式(GP16C4T1_SMCON 寄存器的 SMODS=110)的动作。

16.4.13.3 使用外部触发同步开始两个定时器

这个例子中当定时器 1 的 I1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见下图。为保证计数器的对齐，定时器 1 必须配置为主/从模式(对应 I1 为从，对应定时器 2 为主)：

- ◇ 配置定时器 1 为主模式，送出它的使能做为触发输出(AD16C4T1_CON2 寄存器的 MMSEL=001)。
- ◇ 配置定时器 1 为从模式，从 I1 获得输入触发(AD16C4T1_SMCON 寄存器的 TSSEL='101')。
- ◇ 配置定时器 1 为触发模式(AD16C4T1_SMCON 寄存器的 SMODS='110')。
- ◇ 配置定时器 1 为主/从模式，AD16C4T1_SMCON 寄存器的 MSCFG='1'。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP16C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为触发模式(GP16C4T1_SMCON 寄存器的 SMODS='110')。

当定时器 1 的 I1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TRGI 标志也同时被设置。

注：在这个例子中，在启动之前两个定时器都被初始化(设置相应的 SGU 位)，两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器(AD16C4T1_COUNT)在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNTEN 和计数器时钟之间有个延迟。

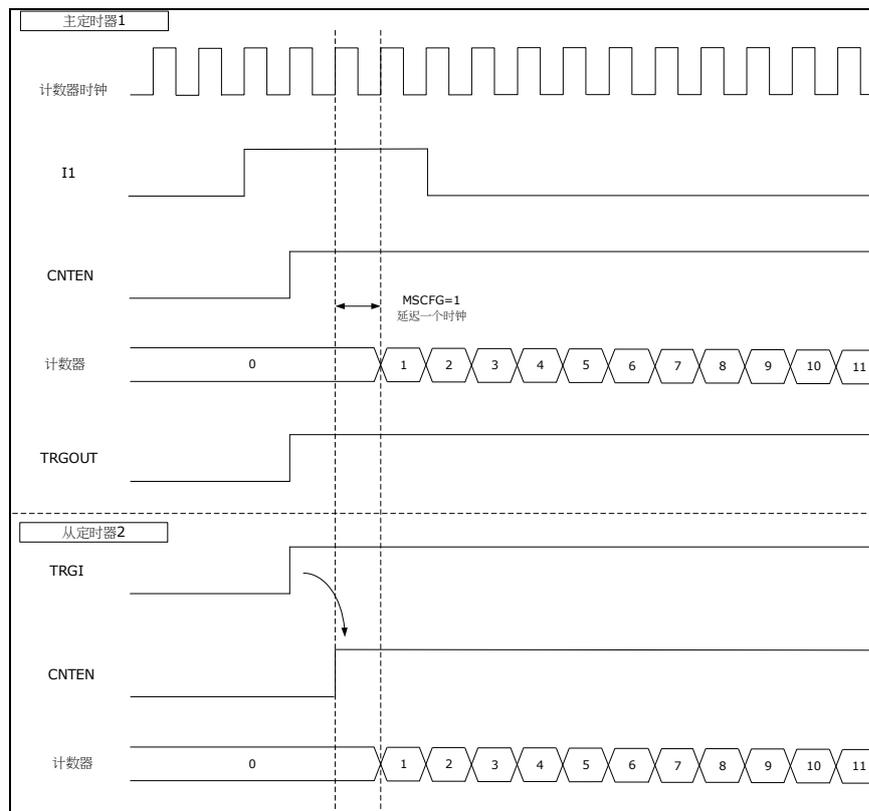


图 16-33 使用定时器 1 的 I1 输入触发定时器 1 和定时器 2

16.4.14 调试模式

当微控制器进入调试模式(Cortex™-M0 核停止运行)，计数器停止计数。

16.5 特殊功能寄存器

16.5.1 寄存器列表

GP16C4T 寄存器列表			
名称	偏移地址	类型	描述
GP16C4Tn_CON1	00 _H	R/W	控制寄存器 1
GP16C4Tn_CON2	04 _H	R/W	控制寄存器 2
GP16C4Tn_SMCON	08 _H	R/W	从模式控制寄存器
GP16C4Tn_IER	0C _H	W1	中断使能寄存器
GP16C4Tn_IDR	10 _H	W1	中断禁止寄存器
GP16C4Tn_IVS	14 _H	R	中断有效状态寄存器
GP16C4Tn_RIF	18 _H	R	原始中断标志寄存器
GP16C4Tn_IFM	1C _H	R	中断屏蔽标志寄存器
GP16C4Tn_ICR	20 _H	C_W1	中断标志清除寄存器
GP16C4Tn_SGE	24 _H	W1	软件生成事件寄存器
GP16C4Tn_CHMR1	28 _H	R/W	捕获/比较模式寄存器 1
GP16C4Tn_CHMR2	2C _H	R/W	捕获/比较模式寄存器 2
GP16C4Tn_CCEP	30 _H	R/W	捕获/比较使能极性寄存器
GP16C4Tn_COUNT	34 _H	R/W	计数器
GP16C4Tn_PRES	38 _H	R/W	时钟预分频器
GP16C4Tn_AR	3C _H	R/W	自动重载寄存器
GP16C4Tn_CCVAL1	44 _H	R/W	通道捕获/比较寄存器 1
GP16C4Tn_CCVAL2	48 _H	R/W	通道捕获/比较寄存器 2
GP16C4Tn_CCVAL3	4C _H	R/W	通道捕获/比较寄存器 3
GP16C4Tn_CCVAL4	50 _H	R/W	通道捕获/比较寄存器 4
GP16C4Tn_DMAEN	58 _H	R/W	DMA 事件使能寄存器

CMSEL	Bits 6-5	R/W	<p>中央对齐模式选择</p> <p>00: 边沿对齐模式，根据 CON1 寄存器的 DIRSEL 位，使得计数器递增或递减</p> <p>01: 中央对齐模式 1，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，只在计数器向下计数时被设置</p> <p>10: 中央对齐模式 2，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，只在计数器向上计数时被设置</p> <p>11: 中央对齐模式 3，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，在计数器向上和向下计数时皆会被设置</p>
DIRSEL	Bit 4	R/W	<p>计数方向选择</p> <p>当计数器配置为中央对齐模式时，此位只能读取计数器的计数方向</p> <p>0: 计数器向上计数</p> <p>1: 计数器向下计数</p>
SPMEN	Bit 3	R/W	<p>单脉冲模式</p> <p>0: 单脉冲模式禁止，计数器不停止</p> <p>1: 单脉冲模式使能，计数器在发生下一次更新事件时，清除 CNTEN 位，计数器停止</p>
UERSEL	Bit 2	R/W	<p>更新事件请求来源选择</p> <p>设置更新事件(UEV)的来源</p> <p>0: 若使能 UI 中断或 DMA 请求，下列事件中产生更新事件请求</p> <ul style="list-style-type: none"> - 计数器溢出/下溢 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 <p>1: 若使能 UI 中断或 DMA 请求，只在计数器溢出/下溢时产生更新事件请求</p>
DISUE	Bit 1	R/W	<p>更新事件禁止</p> <p>设置更新事件(UEV)的产生</p> <p>0: 更新事件(UEV) 使能，下列事件中产生更新事件请求</p> <ul style="list-style-type: none"> - 计数器溢出/下溢 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 <p>缓冲寄存器载入预装载值</p> <p>1: 更新事件(UEV)禁止，不产生更新事件请求，AR、PRES、CCVALn 寄存器保持数值</p> <p>禁止更新事件时，设置 SGE 寄存器的 SGU 位或</p>

			从模式中产生的复位请求，计数器和预分频器仍会被重新初始化
CNTEN	Bit 0	R/W	<p>计数器使能</p> <p>使能计数器后，在外部时钟模式、门控模式和编码模式才能运作。触发模式则可以由硬件设置 CNTEN 位</p> <p>0: 计数器禁止</p> <p>1: 计数器使能</p>

16.5.2.2 控制寄存器 2 (GP16C4Tn_CON2)

控制寄存器 2 (GP16C4Tn_CON2)																															
偏移地址: 04H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								I1SEL	MMSEL<2:0>			CCDMASEL			

—	Bits 31-8	—	—
I1SEL	Bit 7	R/W	<p>选择 I1 引脚功能</p> <p>0: GP16C4Tn_CH1 引脚与 I1 输入连接</p> <p>0: GP16C4Tn_CH1, CH2 和 CH3 引脚与 I1 输入 (XOR) 连接。</p>
MMSEL	Bits 6-4	R/W	<p>主模式选择</p> <p>设置在主模式下发送到从定时器的同步信号 (TRGOUT)与 ADC 输入</p> <p>000: 复位 - 设置 SGE 寄存器信号用于同步触发输出(TRGOUT)。从模式的复位触发产生的复位信号(TRGOUT)则与实际信号相差一个时钟</p> <p>001: 使能 - 计数器的使能信号 CNTEN 用于同步触发输出(TRGOUT), 可用于同步使能数个定时器。门控模式下, 是使用 CON1 寄存器的 CNTEN 位与触发输入信号逻辑产生。当计数器使能信号受控于触发输入时, TRGOUT 上会有一个时钟延迟, 可设置 SMCON 寄存器的 MSCFG 位延迟一个时钟同步定时器计数器</p> <p>010: 更新事件 - 更新事件被用于同步触发输出(TRGOUT)。一个主定时器的更新事件可当作从定时器的预分频器时钟</p> <p>011: 比较脉冲 - 在发生一次捕获或一次比较成功时, 当要设置 CH1I 标志时, 触发输出送出一个正脉冲(TRGOUT)</p> <p>100: 比较信号 - CH1REF 信号用于触发输出 (TRGOUT)</p> <p>101: 比较信号 - CH2REF 信号用于触发输出 (TRGOUT)</p> <p>110: 比较信号 - CH3REF 信号用于触发输出 (TRGOUT)</p> <p>111: 比较信号 - CH4REF 信号用于触发输出 (TRGOUT)</p>
CCDMASEL	Bit 3	R/W	<p>捕获/比较事件的 DMA 选择</p>

			0: 当发生 CHn 事件时, 设置 CHn DMA 请求 1: 当发生更新事件时, 设置 CHn DMA 请求
—	Bit 2-0	R/W	—

			<p>1001: 采样频率fDTS / 8, N = 8 1010: 采样频率fDTS / 16, N = 5 1011: 采样频率fDTS / 16, N = 6 1100: 采样频率fDTS / 16, N = 8 1101: 采样频率fDTS / 32, N = 5 1110: 采样频率fDTS / 32, N = 6 1111: 采样频率fDTS / 32, N = 8</p>
MSCFG	Bit 7	R/W	<p>主/从模式 0: 写入0无效 1: 延迟触发输入 (TI) 上的事件来允许当前计时器和其从器件之间的同步。该设置有效用于使用单个外部事件来同步多个计时器。</p>
TSSEL	Bits 6-4	R/W	<p>触发选择 设置触发选择, 用于同步寄存器 000: 内部触发 0 (IT0) 001: 内部触发 1 (IT1) 010: 内部触发 2 (IT2) 011: 内部触发 3 (IT3) 100: I1 边沿检测(I1F_ED) 101: I1 滤波后信号 110: I2 滤波后信号 111: 外部触发输入</p>
CHCSEL	Bit 3	R/W	<p>CHnREF 清除选择 0: CHnREF_CLR 连接至比较器输入 CMP_IN 1: CHnREF_CLR 连接至外部触发信号 ETF</p>
SMODS	Bits 2-0	R/W	<p>从模式选择 000: 从模式关闭 - 设置 CON1 寄存器 CNTEN 位, 计数器由内部时钟计数 001: 编码器模式 1 - 计数器向上/向下计数 I2 边沿, 取决于 I1 电平 010: 编码器模式 2 - 计数器向上/向下计数 I1 边沿检测边沿, 取决于 I2 边沿检测电平 011: 编码器模式 3 - 计数器向上/向下计数 I1 边沿检测和 I2 边沿检测边沿, 取决于另一个输入的电平 100: 复位模式 - 选中的触发输入(TI)的上升沿重新初始化计数器, 并且产生一次更新事件 101: 门控模式 - 当触发输入(TI)为高电平时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控 110: 触发模式 - 计数器在触发输入 TI 的上升沿启动(但不复位), 只有计数器的启动是受控 111: 外部时钟模式 1 - 选中的触发输入(TI) 的上升沿驱动计数器</p>

			注：如果 I1 双边沿检测被选为触发输入 (TSSEL='100')，不能使用门控模式。I1 每一次转换，I1 双边沿检测就会输出 1 个脉冲，而门控模式则是检查触发信号的电平
--	--	--	--

从定时器	IT0(TSSEL =000)	IT1(TSSEL =001)	IT2(TSSEL =010)	IT3(TSSEL=011)
GP16C4T1	AD16C4T	GP32C4T	GP16C4T2	GP16C4T3
GP16C4T2	AD16C4T	GP32C4T	GP16C4T1	GP16C4T3
GP16C4T3	AD16C4T	GP32C4T	GP16C4T1	GP16C4T2

表 16-2 GP16C4Tn 内部触发连接

16.5.2.4 中断使能寄存器 (GP16C4Tn_IER)

中断使能寄存器 (GP16C4Tn_IER)																																
偏移地址: 0C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																				CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	W1	捕获溢出 4 中断使能 0: 写入0无效 1: 捕获溢出CH4中断使能
CH3OVI	Bit 11	W1	捕获溢出 3 中断使能 0: 写入 0 无效 1: 捕获溢出 CH3 中断使能
CH2OVI	Bit 10	W1	捕获溢出 2 中断使能 0: 写入 0 无效 1: 捕获溢出 CH2 中断使能
CH1OVI	Bit 9	W1	捕获溢出 1 中断使能 0: 写入 0 无效 1: 捕获溢出 CH1 中断使能
—	Bit 8-7	—	—
TRGI	Bit 6	W1	触发中断使能 0: 写入 0 无效 1: 触发中断使能
—	Bit 5	—	—
CH4I	Bit 4	W1	捕获/比较 4 捕获中断使能 0: 写入 0 无效 1: 捕获 CH4 中断使能
CH3I	Bit 3	W1	捕获/比较 3 捕获中断使能 0: 写入 0 无效 1: 捕获 CH3 中断使能
CH2I	Bit 2	W1	捕获/比较 2 捕获中断使能 0: 写入 0 无效 1: 捕获 CH2 中断使能
CH1I	Bit 1	W1	捕获/比较 1 捕获中断使能 0: 写入 0 无效 1: 捕获 CH1 中断使能
UI	Bit 0	W1	更新中断使能 0: 写入 0 无效

			1: 更新中断使能
--	--	--	-----------

16.5.2.5 中断禁止寄存器 (GP16C4Tn_IDR)

中断禁止寄存器 (GP16C4Tn_IDR)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	W1	捕获溢出 4 中断禁止 0: 写入0无效 1: 捕获溢出CH4中断禁止
CH3OVI	Bit 11	W1	捕获溢出 3 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH3 中断禁止
CH2OVI	Bit 10	W	捕获溢出 2 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH2 中断禁止
CH1OVI	Bit 9	W	捕获溢出 1 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH1 中断禁止
—	Bit 8-7	—	—
TRGI	Bit 6	W	触发中断禁止 0: 写入 0 无效 1: 触发中断禁止
—	Bit 5	—	—
CH4I	Bit 4	W	捕获/比较 4 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH4 中断禁止
CH3I	Bit 3	W	捕获/比较 3 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH3 中断禁止
CH2I	Bit 2	W	捕获/比较 2 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH2 中断禁止
CH1I	Bit 1	W	捕获/比较 1 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH1 中断禁止
UI	Bit 0	W	更新中断禁止 0: 写入 0 无效

			1: 更新中断禁止
--	--	--	-----------

16.5.2.6 中断有效状态寄存器 (GP16C4Tn_IVS)

中断有效状态寄存器 (GP16C4Tn_IVS)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	捕获溢出 4 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH4 中断禁止 1: 捕获溢出 CH4 中断使能
CH3OVI	Bit 11	R	捕获溢出 3 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH3 中断禁止 1: 捕获溢出 CH3 中断使能
CH2OVI	Bit 10	R	捕获溢出 2 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH2 中断禁止 1: 捕获溢出 CH2 中断使能
CH1OVI	Bit 9	R	捕获溢出 1 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH1 中断禁止 1: 捕获溢出 CH1 中断使能
—	Bit 8-7	—	—
TRGI	Bit 6	R	触发中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 触发中断禁止 1: 触发中断使能
—	Bit 5	—	保留
CH4I	Bit 4	R	捕获/比较 4 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH4 中断禁止 1: 捕获 CH4 中断使能
CH3I	Bit 3	R	捕获/比较 3 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH3 中断禁止 1: 捕获 CH3 中断使能
CH2I	Bit 2	R	捕获/比较 2 捕获中断有效位

			设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH2 中断禁止 1: 捕获 CH2 中断使能
CH1I	Bit 1	R	捕获/比较 1 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH1 中断禁止 1: 捕获 CH1 中断使能
UI	Bit 0	R	更新中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 更新中断禁止 1: 更新中断使能

16.5.2.7 原始中断标志寄存器 (GP16C4Tn_RIF)

原始中断标志寄存器 (GP16C4Tn_RIF)																															
偏移地址: 18 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	捕获溢出4原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出CH4原始中断产生
CH3OVI	Bit 11	R	捕获溢出3原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH3 原始中断产生
CH2OVI	Bit 10	R	捕获溢出2原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH2 原始中断产生
CH1OVI	Bit 9	R	捕获溢出1原始中断标志 当CH1I已经被设置又再次收到捕获信号将计数器数值写入CCVAL1寄存器中。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 捕获溢出 CH1 原始中断产生
—	Bit 8-7	—	—
TRGI	Bit 6	R	触发原始中断标志 产生触发事件时产生(在从模式-门控时, 在任一沿产生, 其他模式则在有效沿时产生)。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 触发原始中断产生
—	Bit 5	—	—
CH4I	Bit 4	R	捕获/比较 4 原始中断标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 4 原始中断产生
CH3I	Bit 3	R	捕获/比较 3 原始中断标志

			<p>参照 CH1I 描述</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 3 原始中断产生</p>
CH2I	Bit 2	R	<p>捕获/比较 2 原始中断标志</p> <p>参照 CH1I 描述</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 2 原始中断产生</p>
CH1I	Bit 1	R	<p>捕获/比较 1 原始中断标志</p> <p>通道 CH1 设置为输出:</p> <p>计数器匹配 CCVAL1 寄存器时设置, 在中央对齐模式依据 CMSEL 位在递增、递减、递增/递减时产生。当 CCVAL1 寄存器大于 AR 寄存器时, 则在递增边界或递减边界时置起。此位由硬件设置 1, 设置 ICR 寄存器清除</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 1 原始中断产生</p> <p>通道 CH1 设置为输入:</p> <p>发生捕获事件, 将计数器捕获至 CCVAL1 寄存器中, 于 I1 的有效沿产生。此位由硬件设置 1, 设置 ICR 寄存器清除</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 1 原始中断产生</p>
UI	Bit 0	R	<p>更新原始中断标志</p> <p>当 CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断</p> <ul style="list-style-type: none"> - 当重复计数器数值上溢或下溢时 (重复计数器为 0 时时产生更新事件)。 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器重新初始化时。 - 当计数器被触发事件重新初始化时 <p>0: 无中断产生</p> <p>1: 更新原始中断产生</p>

16.5.2.8 中断屏蔽标志寄存器 (GP16C4Tn_IFM)

中断屏蔽标志寄存器 (GP16C4Tn_IFM)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	U

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	捕获溢出4中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出CH4中断屏蔽标志产生
CH3OVI	Bit 11	R	捕获溢出3中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH3 中断屏蔽标志产生
CH2OVI	Bit 10	R	捕获溢出2中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH2 中断屏蔽标志产生
CH1OVI	Bit 9	R	捕获溢出1中断屏蔽标志 当IVS寄存器为1时, CH1I已经被设置又再次收到捕获信号将计数器数值写入CCVAL1寄存器中。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 捕获溢出 CH1 中断屏蔽标志产生
—	Bit 8-7	—	—
TRGI	Bit 6	R	触发中断屏蔽标志 当 IVS 寄存器为 1 时, 产生触发事件时产生(在从模式-门控时, 在任一沿产生, 其他模式则在有效沿时产生)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 触发中断产生
—	Bit 5	—	—
CH4I	Bit 4	R	捕获/比较 4 中断屏蔽标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 4 中断屏蔽标志产生

CH3I	Bit 3	R	<p>捕获/比较 3 中断屏蔽标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 3 中断屏蔽标志产生</p>
CH2I	Bit 2	R	<p>捕获/比较 2 中断屏蔽标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 2 中断屏蔽标志产生</p>
CH1I	Bit 1	R	<p>捕获/比较 1 中断屏蔽标志 当 IVS 寄存器为 1 时 通道 CH1 设置为输出: 计数器匹配 CCVAL1 寄存器时设置, 在中央对齐模式依据 CMSEL 位在递增、递减、递增/递减时产生。当 CCVAL1 寄存器大于 AR 寄存器时, 则在递增边界或递减边界时置起。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 中断屏蔽标志产生 通道 CH1 设置为输入: 发生捕获事件, 将计数器捕获至 CCVAL1 寄存器中, 于 I1 的有效沿产生。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 中断屏蔽标志产生</p>
UI	Bit 0	R	<p>更新中断屏蔽标志 当 CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断 - 当重复计数器数值上溢或下溢时 (重复计数器为 0 时时产生更新事件)。 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器重新初始化时。 - 当计数器被触发事件重新初始化时 0: 无中断产生 1: 更新中断屏蔽标志产生</p>

16.5.2.9 中断标志清除寄存器 (GP16C4Tn_ICR)

中断标志清除寄存器 (GP16C4Tn_ICR)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	C_W1	捕获溢出 4 中断标志清除 0: 写入0无效 1: 捕获溢出CH4中断标志清除
CH3OVI	Bit 11	C_W1	捕获溢出 3 中断标志清除 0: 写入0无效 1: 捕获溢出 CH3 中断标志清除
CH2OVI	Bit 10	C_W1	捕获溢出 2 中断标志清除 0: 写入0无效 1: 捕获溢出 CH2 中断标志清除
CH1OVI	Bit 9	C_W1	捕获溢出 1 中断标志清除 0: 写入0无效 1: 捕获溢出 CH1 中断标志清除
—	Bit 8-7	—	—
TRGI	Bit 6	C_W1	触发中断标志清除 0: 写入 0 无效 1: 触发中断标志清除
—	Bit 5	—	—
CH4I	Bit 4	C_W1	捕获/比较 4 中断标志清除 0: 写入 0 无效 1: 捕获 CH4 中断标志清除
CH3I	Bit 3	C_W1	捕获/比较 3 中断标志清除 0: 写入 0 无效 1: 捕获 CH3 中断标志清除
CH2I	Bit 2	C_W1	捕获/比较 2 中断标志清除 0: 写入 0 无效 1: 捕获 CH2 中断标志清除
CH1I	Bit 1	C_W1	捕获/比较 1 中断标志清除 0: 写入 0 无效 1: 捕获 CH1 中断标志清除
UI	Bit 0	C_W1	更新中断标志清除 0: 写入 0 无效

			1: 更新中断标志清除
--	--	--	-------------

16.5.2.10 软件生成事件寄存器 (GP16C4Tn_SGE)

软件生成事件寄存器 (GP16C4Tn_SGE)																																
偏移地址: 24H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																										SGTRG		SGCH4	SGCH3	SGCH2	SGCH1	SGU

—	Bits 31-7	—	—
SGTRG	Bit 6	W1	<p>软件生成触发事件 该位由软件设置来生成触发事件，可由硬件自动清零。 0: 无动作 1: RIF 寄存器中的 TRGI 被置起，产生相关中断或 DMA 传输</p>
—	Bit 5	—	—
SGCH4	Bit 4	W1	<p>软件生成通道 4 捕获/比较事件 参考 SGCH1 描述</p>
SGCH3	Bit 3	W1	<p>软件生成通道 3 捕获/比较事件 参考 SGCH1 描述</p>
SGCH2	Bit 2	W1	<p>软件生成通道 2 捕获/比较事件 参考 SGCH1 描述</p>
SGCH1	Bit 1	W1	<p>软件生成通道 1 捕获/比较事件 通道 CH1 设置为输出: 产生捕获/比较但不影响输出，若使能中断或 DMA，则产生中断与请求。由软件设置，于下一个时钟自动清除 0: 写入 0 无效 1: 捕获/比较 1 中断产生 通道 CH1 设置为输入: 发生捕获事件，将计数器捕获至 CCVAL1 寄存器中，于 I1 的有效沿产生，若使能中断或 DMA，则产生中断与请求。由软件设置，于下一个时钟自动清除 0: 写入 0 无效 1: 捕获/比较 1 中断产生</p>
SGU	Bit 0	W1	<p>软件触发更新事件 该位由软件设置，可由硬件自动清零。 0: 无动作 1: 重新初始化计数器，更新寄存器。注意，预分频器也会被清零（但预分频比不会受到影响）。如果使用中央对齐模式或者 DIRSEL=0（递增），则计数器将清零；否则如果 DIRSEL=1（递减），则</p>

			将使用自动重载入值。
--	--	--	------------

16.5.2.11 捕获/比较模式寄存器 1 (GP16C4Tn_CHMR1)

捕获/比较模式寄存器 1 (GP16C4Tn_CHMR1)																																	
偏移地址: 28H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																CH2OCLREN	CH2MOD <2:0>			CH2PEN	CH2FEN	CC2SSEL <1:0>			CH1OCLREN	CH1MOD <2:0>			CH1PEN	CH1FEN	CC1SSEL <1:0>		
																I2FLT <3:0>			I2PRES <1:0>			CC2SSEL <1:0>			I1FLT <3:0>			I1PRES <1:0>			CC1SSEL <1:0>		

输出比较模式

—	Bits 31-16	—	—
CH2OCLREN	Bit 15	R/W	输出比较通道 2 清除使能 参考 CH1OCLREN 描述
CH2MOD	Bits 14-12	R/W	输出比较信道 2 模式 参考 CH1MOD 描述
CH2PEN	Bit 11	R/W	输出比较通道 2 预装载使能 参考 CH1PEN 描述
CH2FEN	Bit 10	R/W	输出比较通道 2 快速使能 参考 CH1FEN 描述
CC2SSEL	Bits 9-8	R/W	捕获/比较通道 2 选择 设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC2EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入, 捕获源为 I2 10: 通道设置为输入, 捕获源为 I1 11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测
CH1OCLREN	Bit 7	R/W	输出比较通道 1 清除使能 0: CH1REF 维持输出 1: CH1REF 根据 CHREF_CLR 的有效位清除
CH1MOD	Bits 6-4	R/W	输出比较信道 1 模式 设置 CH1REF 输出模式, CH1 与 CH1N 由 CH1REF 产生, 而有效位由 CCEP 寄存器的 CC1POL 与 CC1NPPOL 位设置电平 000: 禁止 - 无作用 001: 匹配时设置高电平 - 当计数器匹配

			<p>CCVAL1 寄存器时，CH1REF 设置为 1</p> <p>010: 匹配时设置低电平 - 当计数器匹配 CCVAL1 寄存器时，CH1REF 设置为 0</p> <p>011: 匹配时设置翻转电平 - 当计数器匹配 CCVAL1 寄存器时，CH1REF 设置翻转电平(当前高/低电平翻转成低/高电平)</p> <p>100: 强制低电平 - CH1REF 强制设置低电平</p> <p>101: 强制高电平 - CH1REF 强制设置高电平</p> <p>110: PWM 模式 1 - 上数时，当计数器小于 CCVAL1 寄存器时，输出高电平，其他则输出低电平。下数时，当计数器大于 CCVAL1 寄存器时输出低电平，其他则输出高电平</p> <p>111: PWM 模式 2 - 上数时，当计数器小于 CCVAL1 寄存器时，输出低电平，其他则输出高电平。下数时，当计数器大于 CCVAL1 寄存器时输出高准位，其他则输出低电平</p>
CH1PEN	Bit 3	R/W	<p>输出比较通道 1 预装载使能</p> <p>设置后在更新事件时，将设置的寄存器 CCVAL1 数值载入预装载 CCVAL1 寄存器中</p> <p>0: CCVAL1 寄存器预装载禁止</p> <p>1: CCVAL1 寄存器预装载使能</p>
CH1FEN	Bit 2	R/W	<p>输出比较通道 1 快速使能</p> <p>用于加速事件的产生</p> <p>0: CH1 的正常操作依赖于计数器与 CCVAL1 的值，即使工作于触发器状态。当触发器的输入有一个有效沿时，激活 CH1 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，CH1 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 CH1 输出间的延时被缩短为 3 个时钟周期。CH1FEN 只在信道被配置成 PWM1 或 PWM2 模式时起作用</p>
CC1SSEL	Bits 1-0	R/W	<p>捕获/比较通道 1 选择</p> <p>设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC1EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入，捕获源为 I1</p> <p>10: 通道设置为输入，捕获源为 I2</p> <p>11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测</p>

输入捕获模式

—	Bits 31-16	—	—
I2FLT	Bits 15-12	R/W	输入捕获通道2滤波器 参照I1FLT描述
I2PRES	Bits 11-10	R/W	输入捕获通道 2 预分频器 参照 IC1PRES 描述
CC2SSEL	Bits 9-8	R/W	捕获/比较通道 2 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC2EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I2 10: 通道设置为输入，捕获源为 I1 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测
I1FLT	Bits 7-4	R/W	输入捕获通道 1 滤波器 设置 I1 信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变 0000: 采样频率 fDTS，滤波器禁止 0001: 采样频率 fINT_CLK, N = 2 0010: 采样频率 fINT_CLK, N = 4 0011: 采样频率 fINT_CLK, N = 8 0100: 采样频率 fDTS / 2, N = 6 0101: 采样频率 fDTS / 2, N = 8 0110: 采样频率 fDTS / 4, N = 6 0111: 采样频率 fDTS / 4, N = 8 1000: 采样频率 fDTS / 8, N = 6 1001: 采样频率 fDTS / 8, N = 8 1010: 采样频率 fDTS / 16, N = 5 1011: 采样频率 fDTS / 16, N = 6 1100: 采样频率 fDTS / 16, N = 8 1101: 采样频率 fDTS / 32, N = 5 1110: 采样频率 fDTS / 32, N = 6 1111: 采样频率 fDTS / 32, N = 8
I1PRES	Bits 3-2	R/W	输入捕获通道 1 预分频器 设置 I1 的预分频计数器数值，当清除 CCEP 寄存器的 CC1EN 位，预分频计数器同时被清除 00: 预分频禁止，于每次事件时捕获 01: 每 2 次事件捕获 10: 每 4 次事件捕获 11: 每 8 次事件捕获
CC1SSEL	Bits 1-0	R/W	捕获/比较通道 1 选择 设置通道的输出方向与信号的选择，当 CCEP 寄

		<p>寄存器的 CC1EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入，捕获源为 I1</p> <p>10: 通道设置为输入，捕获源为 I2</p> <p>11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测</p>
--	--	--

16.5.2.12 捕获/比较模式寄存器 2 (GP16C4Tn_CHMR2)

捕获/比较模式寄存器 2 (GP16C4Tn_CHMR2)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CH4OCLREN	CH4MOD <2:0>		CH4PEN	CH4FEN	CC4SSEL <1:0>		CH3OCLREN	CH3MOD <2:0>		CH3PEN	CH3FEN	CC3SSEL <1:0>			
																I4FLT <3:0>		I4PRES <1:0>		CC4SSEL <1:0>		I3FLT <3:0>		I3PRES <1:0>		CC3SSEL <1:0>					

输出比较模式

—	Bits 31-16	—	—
CH4OCLREN	Bit 15	R/W	输出比较通道 4 清除使能 参考 CH1OCLREN 描述
CH4MOD	Bits 14-12	R/W	输出比较信道 4 模式 参考 CH1MOD 描述
CH4PEN	Bit 11	R/W	输出比较通道 4 预装载使能 参考 CH1PEN 描述
CH4FEN	Bit 10	R/W	输出比较通道 4 快速使能 参考 CH1FEN 描述
CC4SSEL	Bits 9-8	R/W	捕获/比较通道 4 选择 设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC4EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入, 捕获源为 I4 10: 通道设置为输入, 捕获源为 I3 11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测
CH3OCLREN	Bit 7	R/W	输出比较通道 3 清除使能 参考 CH1OCLREN 描述
CH3MOD	Bits 6-4	R/W	输出比较信道 3 模式 参考 CH1OMOD 描述
CH3PEN	Bit 3	R/W	输出比较通道 3 预装载使能 参考 CH1PEN 描述
CH3FEN	Bit 2	R/W	输出比较通道 3 快速使能 参考 CH1FEN 描述
CC3SSEL	Bits 1-0	R/W	捕获/比较通道 3 选择

		<p>设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC3EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入，捕获源为 I3</p> <p>10: 通道设置为输入，捕获源为 I4</p> <p>11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测</p>
--	--	--

输入捕获模式

—	Bits 31-16	—	—
I4FLT	Bits 15-12	R/W	输入捕获通道4滤波器 参照I1FLT描述
I4PRES	Bits 11-10	R/W	输入捕获通道 4 预分频器 参照 IC1PRES 描述
CC4SSEL	Bits 9-8	R/W	捕获/比较通道 4 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC4EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I4 10: 通道设置为输入，捕获源为 I3 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测
I3FLT	Bits 7-4	R/W	输入捕获通道 3 滤波器 参考 I1FLT 描述
I3PRES	Bits 3-2	R/W	输入捕获通道 3 预分频器 参考 IC1PRES 描述
CC3SSEL	Bits 1-0	R/W	捕获/比较通道 3 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC3EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I3 10: 通道设置为输入，捕获源为 I4 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测

16.5.2.13 捕获/比较使能极性寄存器 (GP16C4Tn_CCEP)

捕获/比较使能寄存器 (GP16C4Tn_CCEP)																																
偏移地址: 30 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																CC4NPOL		CC4POL	CC4EN	CC3NPOL			CC3POL	CC3EN	CC2NPOL		CC2POL	CC2EN	CC1NPOL		CC1POL	CC1EN

—	Bits 31-16	—	—
CC4NPOL	Bit 15	R/W	捕获/比较通道4互补输出有效位极性 参照CC1NPOL描述
—	Bit 14	R/W	—
CC4POL	Bit 13	R/W	捕获/比较通道4输出有效位极性 参照CC1POL描述
CC4EN	Bit 12	R/W	捕获/比较通道4输出使能 参照CC1EN描述
CC3NPOL	Bit 11	R/W	捕获/比较通道3互补输出有效位极性 参照CC1NPOL描述
—	Bit 10	R/W	—
CC3POL	Bit 9	R/W	捕获/比较通道3输出有效位极性 参照CC1POL描述
CC3EN	Bit 8	R/W	捕获/比较通道3输出使能 参照CC1EN描述
CC2NPOL	Bit 7	R/W	捕获/比较通道2互补输出有效位极性 参照CC1NPOL描述
—	Bit 14	R/W	—
CC2POL	Bit 5	R/W	捕获/比较通道2输出有效位极性 参照CC1POL描述
CC2EN	Bit 4	R/W	捕获/比较通道2输出使能 参照CC1EN描述
CC1NPOL	Bit 3	R/W	捕获/比较通道2互补输出有效位极性 通道CH1设置为输出: 0: CH1N 高电平有效 1: CH1N 低电平有效 通道CH1设置为输入: 该位需和CC1POL一起使用来定义输入边沿的极性。参考CC1POL描述。
—	Bit 2	R/W	—
CC1POL	Bit 1	R/W	捕获/比较通道1输出有效位极性 通道CH1设置为输出:

			<p>0: CH1 高电平有效 1: CH1 低电平有效 通道 CC1 设置为输入: CC1NPOL/CC1POL 位选择触发边沿或捕获模式下 I1 和 I2 的极性 00: 非反相/上升沿 01: 反相/下降沿 10: 保留 11: 非反相/上升沿+下降沿</p>
CC1EN	Bit 0	R/W	<p>捕获/比较通道 1 输出使能 通道 CH1 设置为输出: 0: 关闭 - CH1 无效 1: 开启 - CH1 为对应输出引脚上的输出信号 通道 CH1 设置为输入: 0: 捕获禁止 1: 捕获使能</p>

16.5.2.14 计数寄存器 (GP16C4Tn_COUNT)

计数寄存器 (GP16C4Tn_COUNT)																															
偏移地址: 34 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CNTV<15:0>															

—	Bits 31-16	—	—
CNTV	Bits 15-0	R/W	计数器数值

16.5.2.15 预分频寄存器 (GP16C4Tn_PRES)

时钟预分频寄存器 (GP16C4Tn_PRES)																															
偏移地址: 38 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PSCV<15:0>															

—	Bits 31-16	—	—
PSCV	Bits 15-0	R/W	预分频数值 当计数器时钟频率等于fINT_CLK/(PSCV<15:0> + 1)时计数器递增或递减。在更新事件产生时, 将PSCV数值被载入预装载寄存器中

16.5.2.16 自动重载寄存器 (GP16C4Tn_AR)

自动重载寄存器 (GP16C4Tn_AR)																															
偏移地址: 3C _H																															
复位值: 00000000_00000000_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																ARV<15:0>															

—	Bits 31-16	—	—
ARV	Bits 15-0	R/W	自动装载数值 设置计数器的递增边界，设置数值为 0 时计数器停止计数

16.5.2.17 通道捕获/比较寄存器 1 (GP16C4Tn_CCVAL1)

通道捕获/比较寄存器 1 (GP16C4Tn_CCVAL1)																															
偏移地址: 44 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CCRV1<15:0>															

—	Bits 31-16	—	—
CCRV1	Bits 15-0	R/W	捕获/比较数值 1 信道 CHn 配置为输出: CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。 如果在CHMRn寄存器中的预载功能没有选中, CCRVn中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与COUNT寄存器中的值进行比较, 并在CHn上输出。 信道CHn配置为输入: CCRVn为由上一个输入捕获事件 (In) 传输的计数值。

16.5.2.18 通道捕获/比较寄存器 2 (GP16C4Tn_CCVAL2)

通道捕获/比较寄存器 2 (GP16C4Tn_CCVAL2)																															
偏移地址: 48 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CCRV2<15:0>															

—	Bits 31-16	—	—
CCRV2	Bits 15-0	R/W	<p>捕获/比较数值2</p> <p>CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。</p> <p>如果在CHMRn寄存器中的预载功能没有选中, CCRVn中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与COUNT寄存器中的值进行比较, 并在CHn上输出。</p> <p>信道CHn配置为输入:</p> <p>CCRVn为由上一个输入捕获事件 (In) 传输的计数值。</p>

16.5.2.19 通道捕获/比较寄存器 3 (GP16C4Tn_CCVAL3)

通道捕获/比较寄存器 3 (GP16C4Tn_CCVAL3)																															
偏移地址: 4C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CCRV3<15:0>															

—	Bits 31-16	—	—
CCRV3	Bits 15-0	R/W	<p>捕获/比较数值3</p> <p>CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。</p> <p>如果在CHMRn寄存器中的预载功能没有选中, CCRVn中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存</p>

			<p>器中。有效捕获/比较寄存器中包含的值将会与COUNT寄存器中的值进行比较，并在CHn上输出。</p> <p>信道CHn配置为输入： CCRVn为由上一个输入捕获事件（In）传输的计数值。</p>
--	--	--	---

16.5.2.20 通道捕获/比较寄存器 4 (GP16C4Tn_CCVAL4)

通道捕获/比较寄存器 4 (GP16C4Tn_CCVAL4)																															
偏移地址：50H																															
复位值：00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CCRV4<15:0>															

—	Bits 31-16	—	—
CCRV4	Bits 15-0	R/W	<p>捕获/比较数值4 CCRVn 中的值将被载入实际的捕获/比较寄存器中（预载值）。 如果在CHMRn寄存器中的预载功能没有选中，CCRVn中的值将被永久载入；否则，每当发生更新事件，预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与COUNT寄存器中的值进行比较，并在CHn上输出。</p> <p>信道CHn配置为输入： CCRVn为由上一个输入捕获事件（In）传输的计数值。</p>

16.5.2.21 DMA 事件使能寄存器 (GP16C4Tn_DMAEN)

DMA 事件使能寄存器 (GP16C4Tn_DMAEN)																																
偏移地址: 58H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																										TRGIDE		CH4DE	CH3DE	CH2DE	CH1DE	UDE

—	Bits 31-7	—	—
TRGIDE	Bit 6	R/W	触发DMA请求使能 0: 触发DMA请求禁止 1: 触发DMA请求使能
—	Bits 5	—	—
CH4DE	Bit 4	R/W	通道捕获/比较 4 DMA 请求使能 0: 捕获/比较 4 DMA 请求禁止 1: 捕获/比较 4 DMA 请求使能
CH3DE	Bit 3	R/W	通道捕获/比较3 DMA请求使能 0: 捕获/比较3 DMA请求禁止 1: 捕获/比较3 DMA请求使能
CH2DE	Bit 2	R/W	通道捕获/比较 2 DMA 请求使能 0: 捕获/比较 2 DMA 请求禁止 1: 捕获/比较 2 DMA 请求使能
CH1DE	Bit 1	R/W	通道捕获/比较 1 DMA 请求使能 0: 捕获/比较 1 DMA 请求禁止 1: 捕获/比较 1 DMA 请求使能
UDE	Bit 0	R/W	更新 DMA 请求使能 0: 更新 DMA 请求禁止 1: 更新 DMA 请求使能

第17章 通用定时器 32 位 4 通道 (GP32C4T)

17.1 概述

通用定时器 32 位 4 通道 (GP32C4Tn) 包含一个 32 位自动重载计数器，该计数器由可配置的预分频器驱动。

通用定时器 32 位 4 通道 (GP32C4Tn) 的用途广泛，可测量信号脉冲长度 (输入捕获) 或输出脉冲波形 (比较输出、PWM)。

17.2 特性

- ◆ 32 位递增，递减，递增/递减自动加载计数器
- ◆ 16 位可编程预分频器，可在定时器运行中对计数器工作时钟进行 1 到 65536 间的任意分频
- ◆ 带有四个独立信道，每个信道支持以下功能
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ PWM 输出 (边沿与中央对齐模式)
 - ◇ 单脉冲输出
- ◆ 同步电路用于外部信号控制定时器及内部互联多个定时器
- ◆ 以下事件中产生中断/DMA:
 - ◇ 更新事件：计数器上溢/下溢，计数器初始化 (通过软件或内/外部触发)
 - ◇ 触发事件 (计数器起始、停止、初始化或内/外触发计数)
 - ◇ 输入捕获
 - ◇ 输出比较
- ◆ 支持增量 (正交) 编码及霍尔电路进行定位
- ◆ 触发输入可对外部时钟或周期性电流管理

17.3 结构图

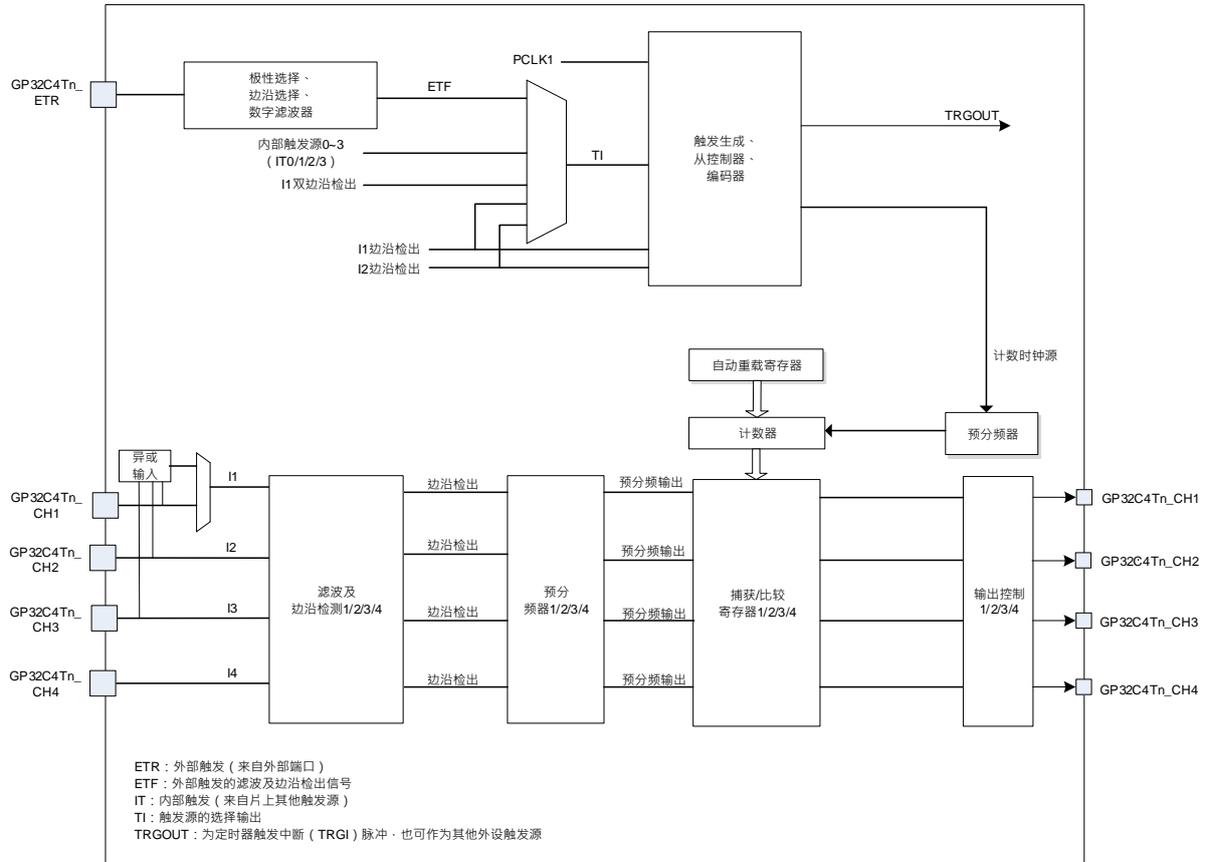


图 17-1 GP32C4Tn 结构框图

17.4 功能描述

17.4.1 预分频器

定时器包含一个 32-bit 的计数器（GP32C4Tn_COUNT），计数时钟由预分频寄存器（GP32C4Tn_PRES）进行分频。计数周期由自动重载计数器（GP32C4Tn_AR）设定。

自动重载寄存器（GP32C4Tn_AR）是一个可缓存的寄存器。当 GP32C4Tn_CON1 寄存器的 ARPEN 位复位时，GP32C4Tn_AR 寄存器重载功能失效，GP32C4Tn_AR 就是有效寄存器；ARPEN 置位时，GP32C4Tn_AR 寄存器具有重载功能，产生更新事件（UEV）时，加载值（GP32C4Tn_AR 寄存器值）更新到影子寄存器。

当 GP32C4Tn_CON1 寄存器中 DISUE 位为 0 时，计数器计数上溢（或递减下溢）时会产生更新事件（UEV）。同样，软件方式也可产生更新事件。GP32C4Tn_CON1 寄存器的 CNTEN 置位时，计数器开始计数。

注：计数器在 CNTEN 位置位 1 个时钟周期后开始计数。

预分频器可对定时器工作时钟进行 GP32C4Tn_PRES 寄存器值+1 次分频。由于 GP32C4Tn_PRES 是一个可重载寄存器，因此，定时器工作时可以对该寄存器进行修改，修改值在下次更新事件（UEV）后有效。

下图给出了定时器运行过程中改变预分频值时计数器的计数情况。

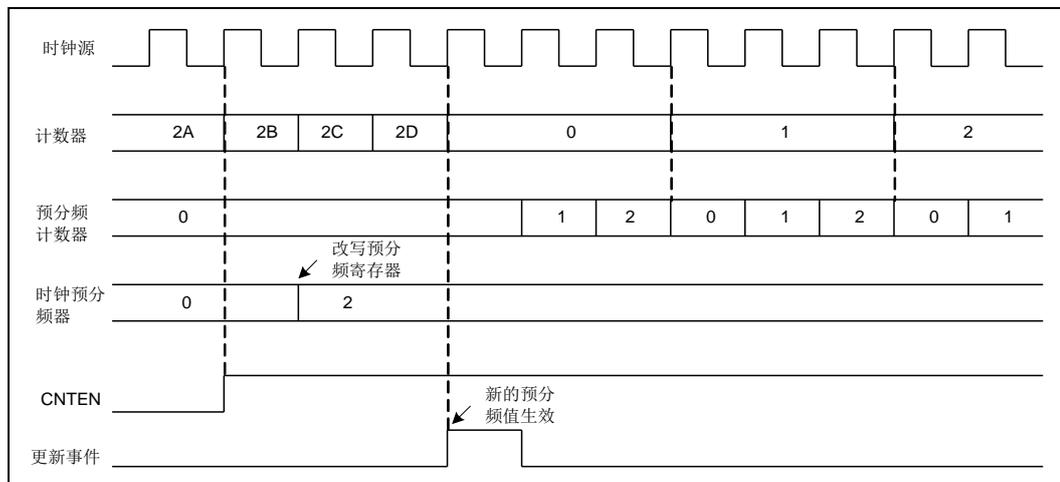


图 17-2 预分频值计数时序图

17.4.2 时钟源

计数器工作时钟可以选择内部时钟(INT_CLK)、外部时钟源 1(I1、I2)、外部时钟源 2(ETR)，内部触发输入 (IT0、IT1、IT2、IT3)

17.4.2.1 内部时钟源 (INT_CLK)

若从模式控制器被关闭 (GP32C4Tn_SMCON 寄存器内, SMODS= "000"), 则 CNTEN, GP32C4Tn_CON1.DIRSEL 与 GP32C4Tn_SGE.SGU 位为实际控制位, 这些位只能软件修改 (SGU 位除外, 仍硬件自动清除)。一旦 CNTEN 位被写为'1', 预分频器就由内部 INT_CLK 提供时钟。

下图给出了通常模式下控制电路和递增计数的情况, 没有分频。

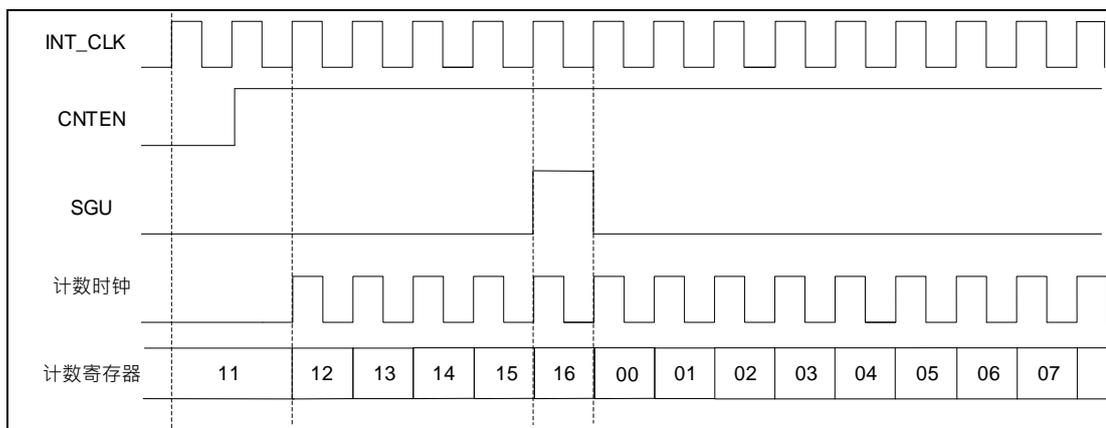


图 17-3 采用内部时钟计数

17.4.2.2 外部时钟源 1

GP32C4Tn_SMCON 寄存器的 **SMODS** = "111" 时, 可选择外部时钟源 1。计数器可根据选定的上升沿或下降沿计数。

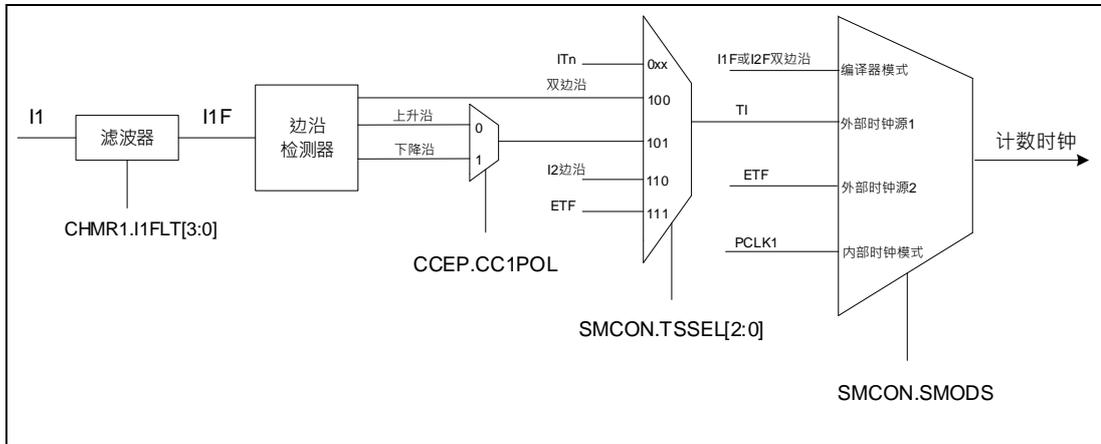


图 17-4 外部时钟连接

配置计数器为外部时钟源 1, 步骤如下:

1. **GP32C4Tn_SMCON** 寄存器中 **SMODS** = "111", 配置定时器外部时钟模式 1。
2. 设置 **GP32C4Tn_SMCON** 寄存器中的 **TSSEL** 选择外部时钟源。
3. 如外部时钟源为 I1, 可配置 **GP32C4Tn_CHMR1** 寄存器 **CC1SSEL** = "01", 配置信道 1 检测 I1 输入的上升沿; 设置 **GP32C4Tn_CCEP** 寄存器中 **CC1POL** = '0', 选择极性为上升沿。
4. 写 **GP32C4Tn_CHMR1** 寄存器的 **I1FLT[3: 0]** 位, 配置输入滤波器时间 (若没有滤波器需求, 维持 **I1FLT** = "0000")。
5. **GP32C4Tn_CON1** 寄存器中 **CNTEN** = '1', 使能计数器。

当 I1 上出现一次上升沿时, 计数器计数一次且 **TRGI** 标志位置位。

17.4.2.3 外部时钟源 2

置位 **GP32C4Tn_SMCON** 寄存器的 ECM2EN 位选定外部时钟源 2。

计数器可对外部触发输入 ETR 进行上升沿或下降沿计数。

下图给出了外部输入输入模块的概况。

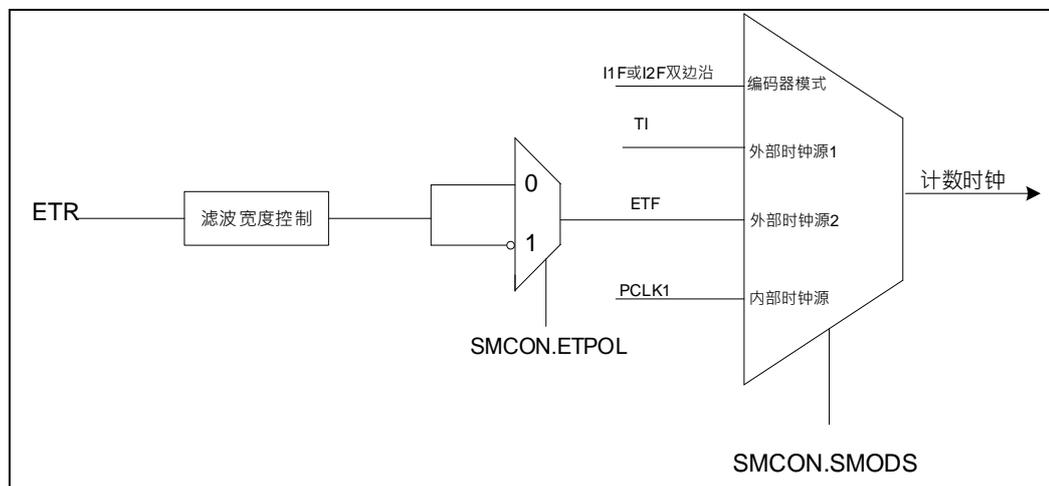


图 17-5 外部触发输入模块

配置计数器为外部时钟源 2，配置过程如下：

1. 设置 **GP32C4Tn_SMCON** 寄存器的 ETFLT[3: 0]，配置输入滤波时间。
2. 设置 **GP32C4Tn_SMCON** 寄存器中 ETPOL，检测 ETR 引脚上升沿或下降沿。
3. 设置 **GP32C4Tn_SMCON** 寄存器中 ECM2EN = '1'，使能外部时钟模式 2。
4. 设置 **GP32C4Tn_CON1** 寄存器的 CNTEN = '1'，使能计数器。

计数器每一个上升沿计一次数。

17.4.2.4 内部触发输入 (ITn)

当 **GP32C4Tn_SMCON** 寄存器的 **SMODS**= "111", 选定内部触发模式。计数器根据选定的内部输入端的上升或下降沿计数。

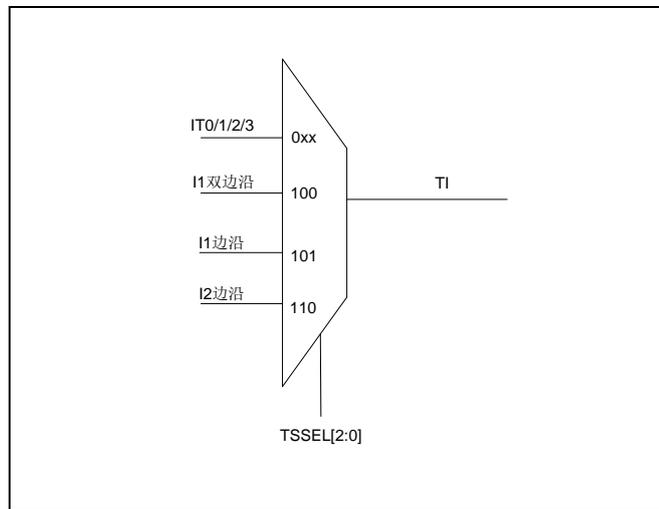


图 17-6 ITn 外部时钟连接

配置计数器在 ITn 输入端的上升沿递增计数，步骤如下：

1. **GP32C4Tn_SMCON** 寄存器中 **SMODS** = "111", 配置外部时钟模式 1。
2. **GP32C4Tn_SMCON** 寄存器的 **TSSEL** = "0xx", 选定 ITn 作为触发输入源。
3. **GP32C4Tn_CON1** 寄存器的 **CNTEN** = '1', 使能计数器。

ITn 产生上升沿时，计数器计数一次。ITn 上升沿与实际时钟间的延时，取决于 ITn 输入的再同步电路。

17.4.3 计数模式

17.4.3.1 递增计数模式

当 **GP32C4Tn_CON1** 寄存器的 **DIRSEL** 值为 0 时，定时器配置为递增模式，计数器从 0 开始递增，直至 **GP32C4Tn_AR** 寄存器值；然后从 0 重新开始计数并产生一个更新事件(UEV)。

当有更新事件（UEV）产生时，预装载寄存器会更新到影子寄存器，更新标志位（**GP32C4Tn_RIF** 寄存器中的 **UI** 位）置位（取决于 **UERSEL** 位）：

- ◇ 更新 **GP32C4Tn_AR** 寄存器的值到影子寄存器
- ◇ 更新 **GP32C4Tn_PRES** 寄存器的值到影子寄存器

下图为 **GP32C4Tn_AR = 0x16**，预分频设为 2 分频时的计数器时序。

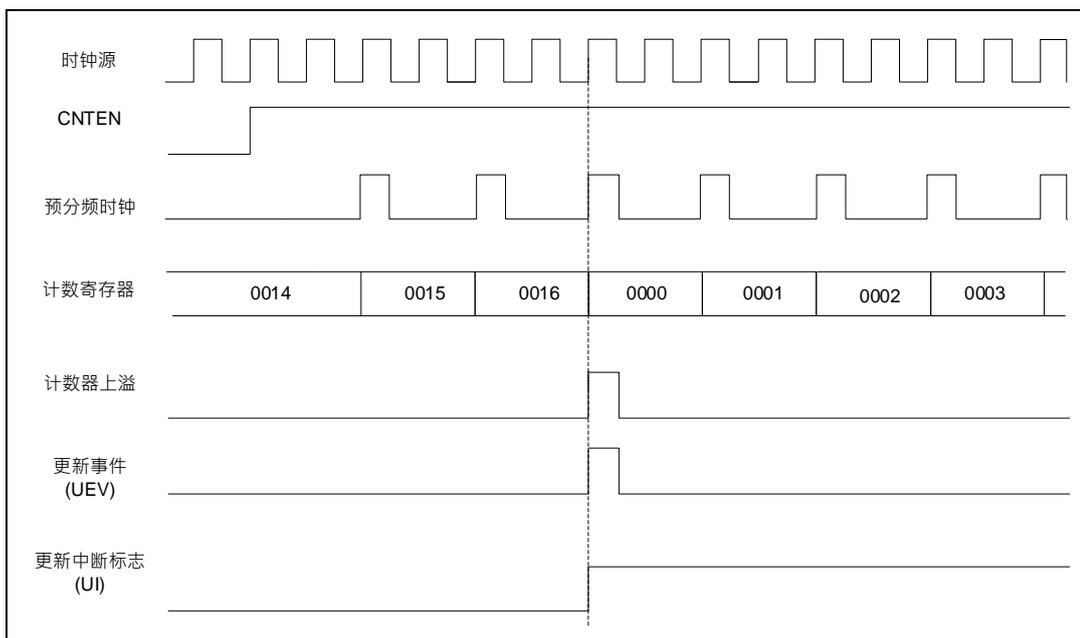


图 17-7 计数器递增计数时序图

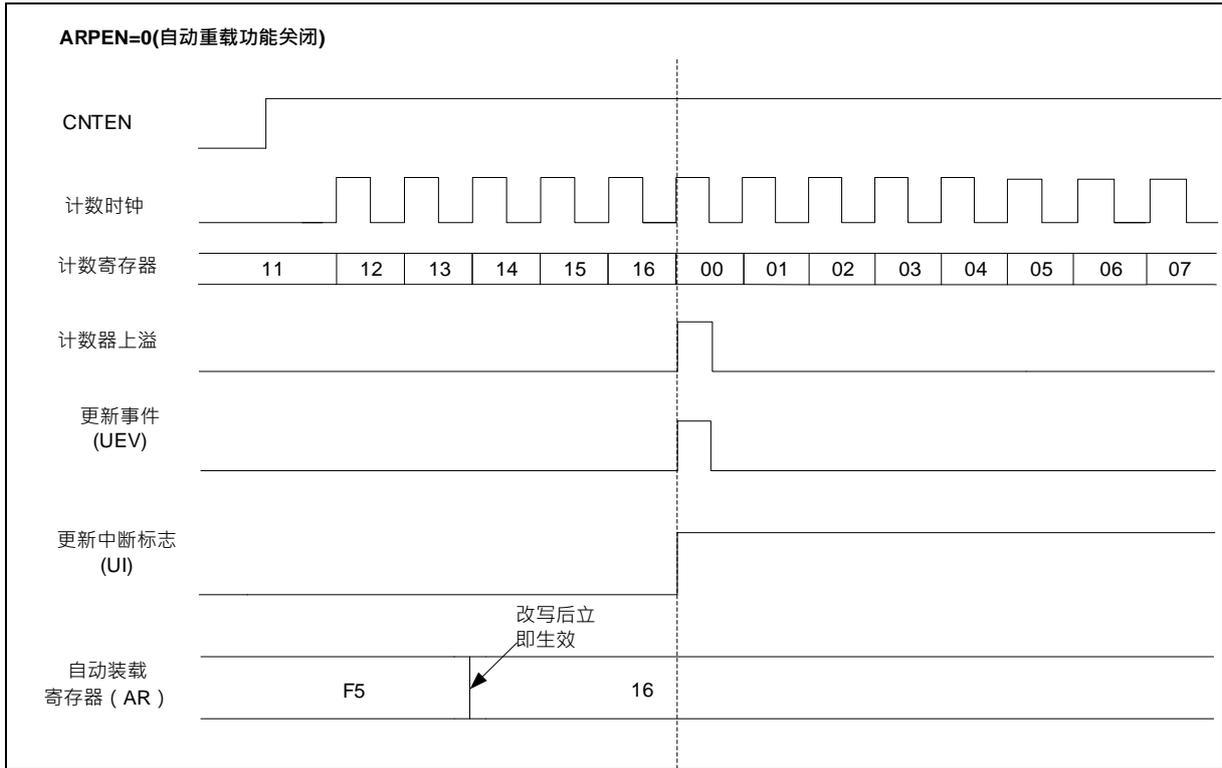


图 17-8 当 ARPEN=0 时计数器时序图

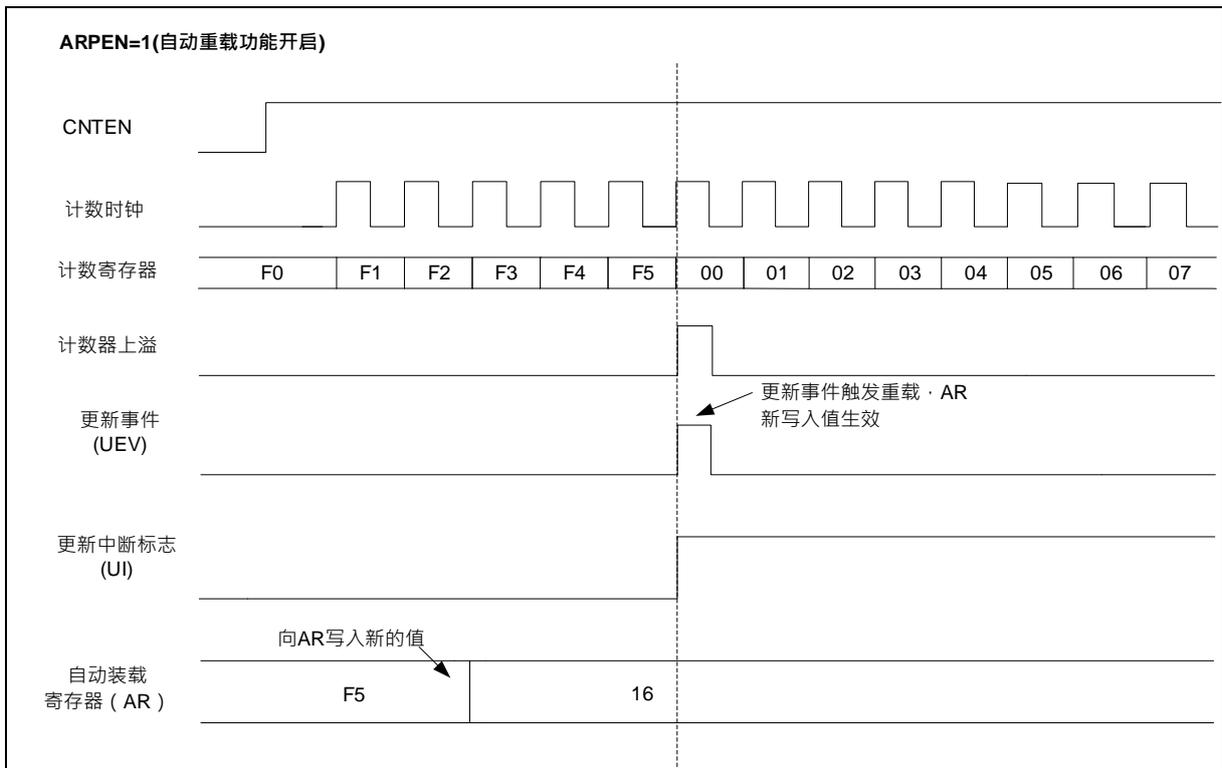


图 17-9 当 ARPEN=1 时计数器时序图

17.4.3.2 递减计数模式

当 GP32C4Tn_CON1 寄存器的 DIRSEL 值为 1 时，定时器配置为递减模式，计数器从 GP32C4Tn_AR 寄存器值开始递减至 0；然后重复递减并产生更新事件（UEV）。

置位 GP32C4Tn_SGE 寄存器中的 SGU 位（通过软件或使用从机模式控制器）同样会产生更新事件。

当有更新事件（UEV）产生时，预载寄存器值会更新到影子寄存器，更新标志位（GP32C4Tn_RIF 寄存器中的 UI 位）置位（取决于 UERSEL 位）。

下图为 GP32C4Tn_AR = 0x27，预分频设为 1 分频时的计数器时序。

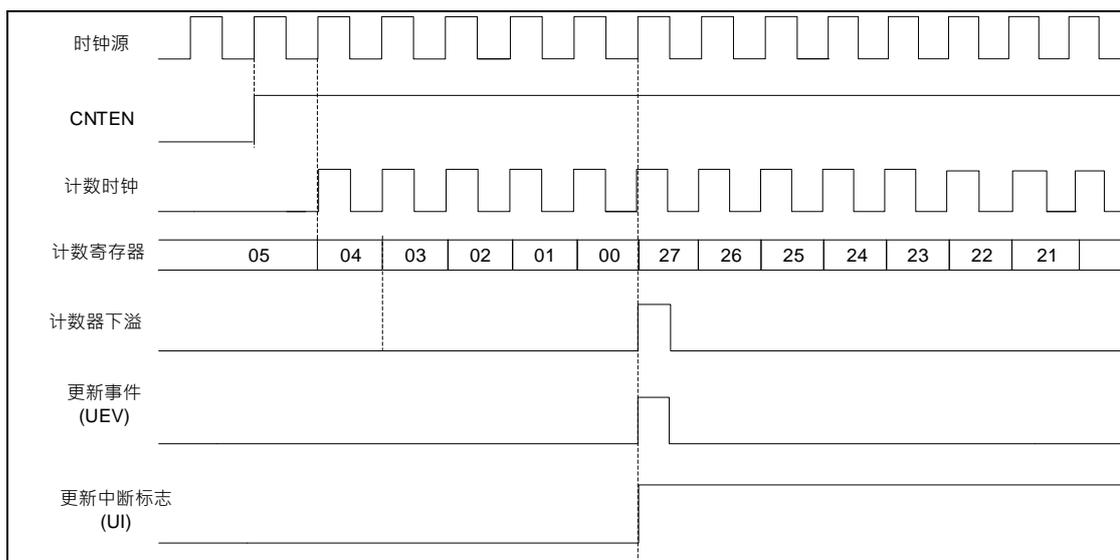


图 17-10 定时器递减计数时序图

17.4.3.3 中央对齐模式

当 **GP32C4Tn_CON1** 寄存器的 **CMSEL** 位的值不等于"00"时,定时器工作在中央对齐模式。定时器配置为中央对齐模式时,计数器先从 0 开始递增至 **GP32C4Tn_AR** 寄存器值减 1 后产生更新事件(UEV);接着计数器从 **GP32C4Tn_AR** 寄存器值递减至 1,并产生更新事件,如此循环计数。计数器递减计数(中央对称模式 1, **CMSEL**="01")、计数器递增计数(中央对称模式 2, **CMSEL**="10")、计数器递增和递减计数(中央对称模式 3, **CMSEL**="11"),每个通道的输出比较中断标志位都会置位。

在中央对齐模式下, **GP32C4Tn_CON1** 寄存器的 **DIRSEL** 位无法进行写操作,该位由硬件自动更新指示当前计数方向。

计数上溢、下溢或者置位 **GP32C4Tn_SGE** 寄存器的 **SGU** 位(通过软件或使用从模式控制器)都会产生更新事件。因此,计数器根据 **DIRSEL** 位由 0 或是 **GP32C4Tn_AR** 寄存器值开始递增或递减,而预分频器都从 0 开始计数。

软件置位 **GP32C4Tn_CON1** 寄存器中的 **DISUE** 位可关闭更新事件(UEV)的产生。更新事件(UEV)关闭时,可避免向预载寄存器写新值时更新影子寄存器。**DISUE** 复位之前都不会产生更新事件。而在正常产生更新事件时,计数器仍然从 0 开始,同样预分频计数也是从 0 开始(但预分频值没有改变)。此外,若置位 **GP32C4Tn_CON1** 寄存器中的 **UERSEL** 位(更新请求选择),置位 **SGU** 位时会产生一次更新事件(UEV),但 **UI** 标志位不会置位(因此,不会触发中断或 DMA 请求)。这就避免了在捕获事件时,清除计数器值时产生更新和捕获中断。

当有更新事件(UEV)产生时,预载寄存器值会更新到影子寄存器,更新标志位(**GP32C4Tn_RIF** 寄存器中的 **UI** 位)置位(取决于 **UERSEL** 位)。

注:若更新源为计数器上溢,自动重载会在计数器重载前更新。因此,下一周期即为预期值(计数器载入新值)。

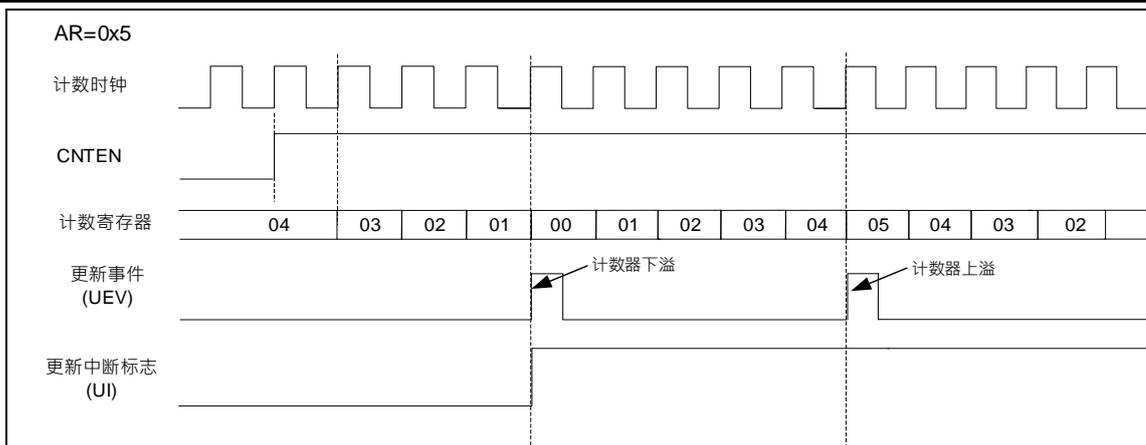


图 17-11 增减计数器时序图

17.4.4 捕获/比较通道

输入电路对 I_n 输入端的信号进行采样，产生一个经过滤波的信号 I_nF 。之后，一个可极性选择的边沿检测器产生 I_n 边沿检测信号，该信号可作为从模式控制器的触发输入或作为捕获控制命令，且信号经过分频后进入捕获寄存器。

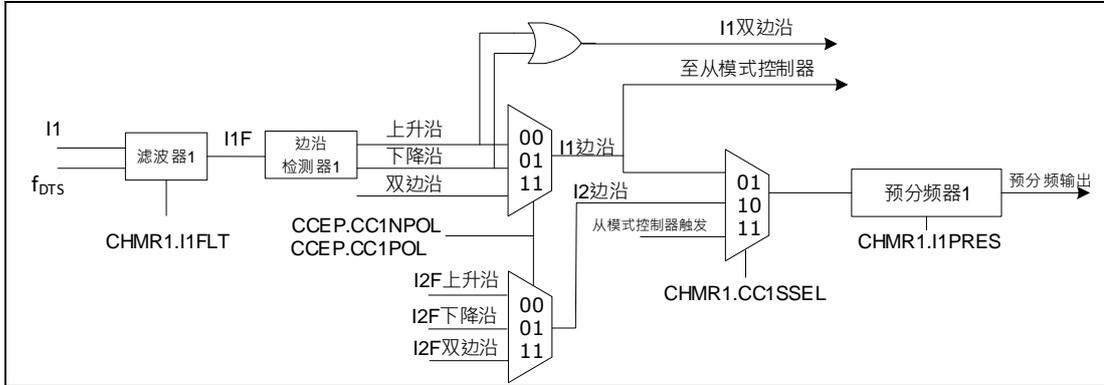


图 17-12 捕获/比较通道

输出部分产生一个中间波形（高有效）作为基准，在输出末端决定最终输出信号的极性。

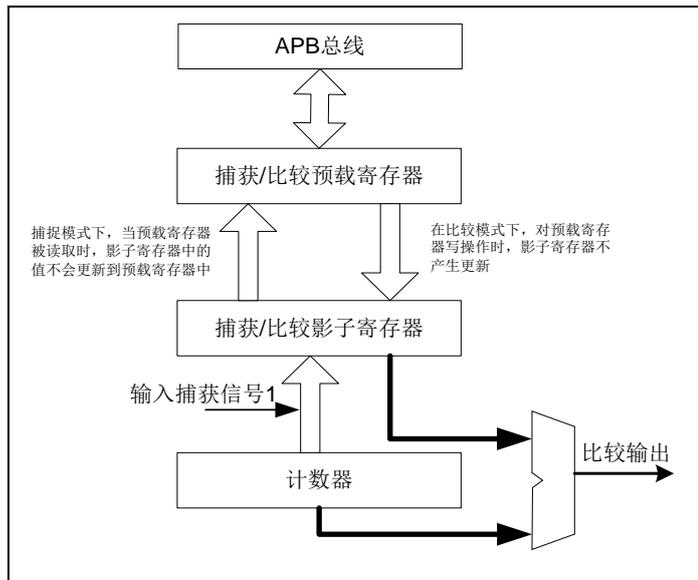


图 17-13 捕获/比较通道 1 结构图

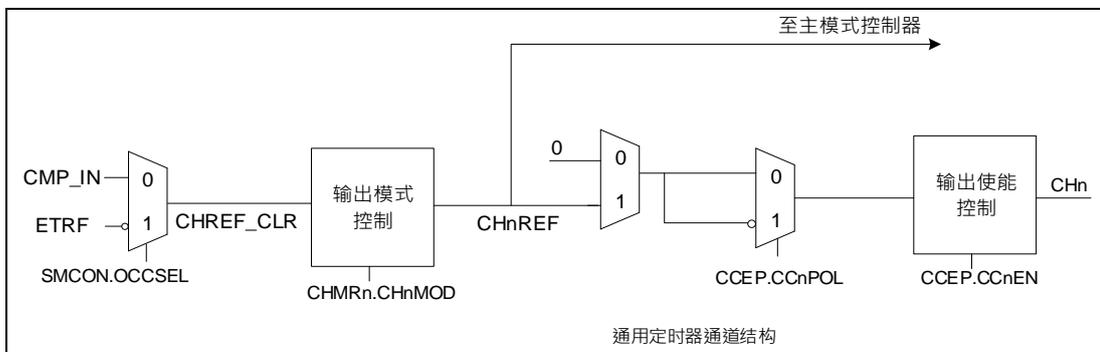


图 17-14 捕获/比较通道 1 结构图

17.4.5 输入捕获模式

在输入捕获模式下，当检测到 In 上相应信号变化时，计数器的值就会被锁存到捕获/比较寄存器(GP32C4Tn_CCVALn)寄存器中。当捕获发生时，相应的 CHnI 标志位(GP32C4Tn_RIF)会置位，同时会触发中断或 DMA (如果使能) 请求。若发生捕获时，CHnI 标志位已经置位，则过捕获 CHnOVI 标志位 (GP32C4Tn_RIF) 置位。软件于 GP32C4Tn_ICR 对 CHnI 与 CHnOVI 位写'1'可以复位 CHnI 标志位与 CHnOVI 标志位。

以下为以 I1 输入上升沿作为捕获输入时的流程：

1. 选择有效输入端：GP32C4Tn_CCVAL1 必须连接到 I1 输入端，因此需将 GP32C4Tn_CHMR1 寄存器中的 CC1SSEL 位写"01"。只要 CC1SSEL 不为"00"，信道被配置为输入且 GP32C4Tn_CCVAL1 寄存器为只读。
2. 根据定时器连接的输入信号，配置输入滤波器的持续时间。当输入信号翻转时，前 5 个内部时钟信号内信号是不稳定的，因此必须配置滤波器的时间大于 5 个时钟周期。当 I1 检测到新的电平，连续 8 次采样可确认电平变化有效。
3. 选择 I1 信道的有效边沿变换。GP32C4Tn_CCEP 寄存器中的 CC1POL 写'0'(上升沿)。
4. 配置输入预分频器。
5. 置位 GP32C4Tn_CCEP 寄存器中的 CC1EN 位，使能捕获计数器的值到捕获寄存器。
6. 如有需要，置位 GP32C4Tn_IER 寄存器中的 CH1I 位，使能中断请求。置位 GP32C4Tn_DMAEN 寄存器中的 CH1DE 位，使能 DMA 请求。

当发生输入捕获时：

1. 有效边沿产生，GP32C4Tn_CCVAL1 寄存器获取计数器的值。
2. CH1I 标志位置位 (中断标志)。若至少 2 个连续的捕获发生，但标志位没有及时清除，则 CH1OVI 也会置位。
3. 中断的产生取决于 GP32C4Tn_IER 寄存器中的 CH1I 位。
4. DMA 请求的产生取决于 GP32C4Tn_DMAEN 寄存器中的 CH1DE 位。

为了处理捕获溢出，建议在读取过捕获标志位前先读取捕获数据。这可以避免错过读过捕获标志位之后，读之前产生的捕获数据。

注：捕获中断请求可由软件设置 GP32C4Tn_SGE 寄存器中 SGCHn 位产生。

17.4.5.1 PWM 输入模式

测量 I1 上 PWM 信号的周期和占空比的过程如下：

1. 为 **GP32C4Tn_CCVAL1** 选择有效的输入： **GP32C4Tn_CHMR1** 寄存器中的 **CC1SSEL** 位写"01"（I1 被选择）。
2. 为 I1 边沿检测选择有效的极性（用于捕获数据到 **GP32C4Tn_CCVAL1** 寄存器和计数器清零）：**CC1POL** 位写'0'（上升沿有效）。
3. 为 **GP32C4Tn_CCVAL2** 选择有效输入： **GP32C4Tn_CHMR1** 寄存器的 **CC2SEL** 位写"10"（I1 被选择）。
4. 为 I1 边沿检测选择有效极性(用于捕获数据到 **GP32C4Tn_CCVAL2**):**CC2POL** 位写'1'。
5. 选择有效的触发输入： **GP32C4Tn_SMCON** 寄存器的 **TSSEL** 位写"101"（I1 边沿检测被选择）。
6. 配置从机模式控制器为复位模式： **GP32C4Tn_SMCON** 寄存器的 **SMODS** 位写"100"。
7. 使能捕获： **GP32C4Tn_CCEP** 寄存器的 **CC1EN** 位和 **CC2EN** 位写'1'。

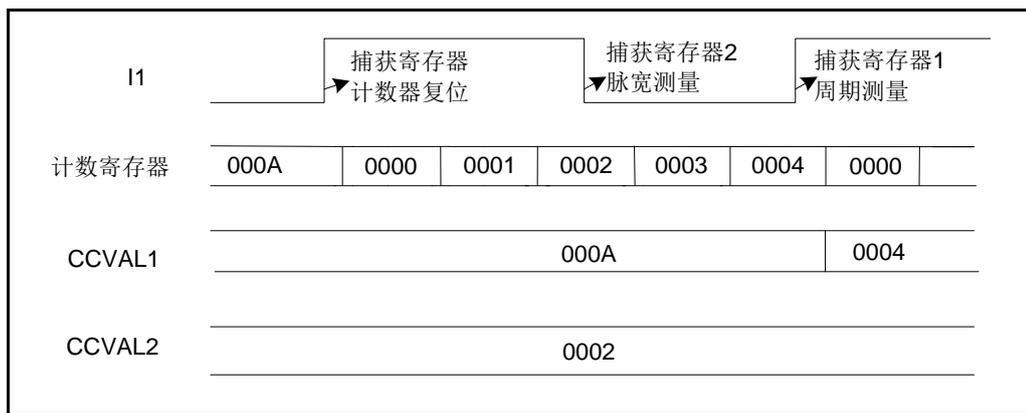


图 17-15 PWM 输入模式时序

17.4.6 PWM 模式

脉宽调制模式可以产生一个 GP32C4Tn_AR 寄存器值确定频率，GP32C4Tn_CCVALn 寄存器值确定占空比的信号。

每个信道的 PWM 模式是相互独立的（每个 CHn 输出一个 PWM），GP32C4Tn_CHMRn 寄存器的 CHnMOD 位写"110"（PWM 模式 1）或写"111"（PWM 模式 2）。必须通过置位 GP32C4Tn_CHMRn 寄存器的 CHnPEN 位来使能相应的预载寄存器，最后还需置位 GP32C4Tn_CON1 寄存器的 ARPEN 位来使能自动重装预载功能。

只有当更新事件发生时预载寄存器中的值才会传到影子寄存器，因此，在使能计数前，必须通过置位 GP32C4Tn_SGE 寄存器的 SGU 位来初始化所有的寄存器。

CHn 的极性可通过 GP32C4Tn_CCEP 寄存器的 CCnPOL 位配置，有效极性可配置为高或低。CHn 的输出使能由 CCnEN、CCnNEN 位（GP32C4Tn_CCEP 寄存器）组合控制。

在 PWM 模式（1 或 2）中，GP32C4Tn_COUNT 和 GP32C4Tn_CCVALn 寄存器的值会持续比较，确定 $GP32C4Tn_CCVALn \leq GP32C4Tn_COUNT$ 或 $GP32C4Tn_CCVALn > GP32C4Tn_COUNT$ （取决于计数器的计数方向）。

定时器产生 PWM 波形是边沿对齐或中央对齐，取决于 GP32C4Tn_CON1 寄存器的 CMSEL 位。

17.4.6.1 PWM 边沿对齐模式

- ◇ GP32C4Tn_AR=8
- ◇ PWM 模式 1
- ◇ 递增计数配置

当 GP32C4Tn_CON1 寄存器的 DIRSEL 位为低时，计数器递增计数。

下图给出了 GP32C4Tn_AR = 8 时的边沿对齐 PWM 波形。

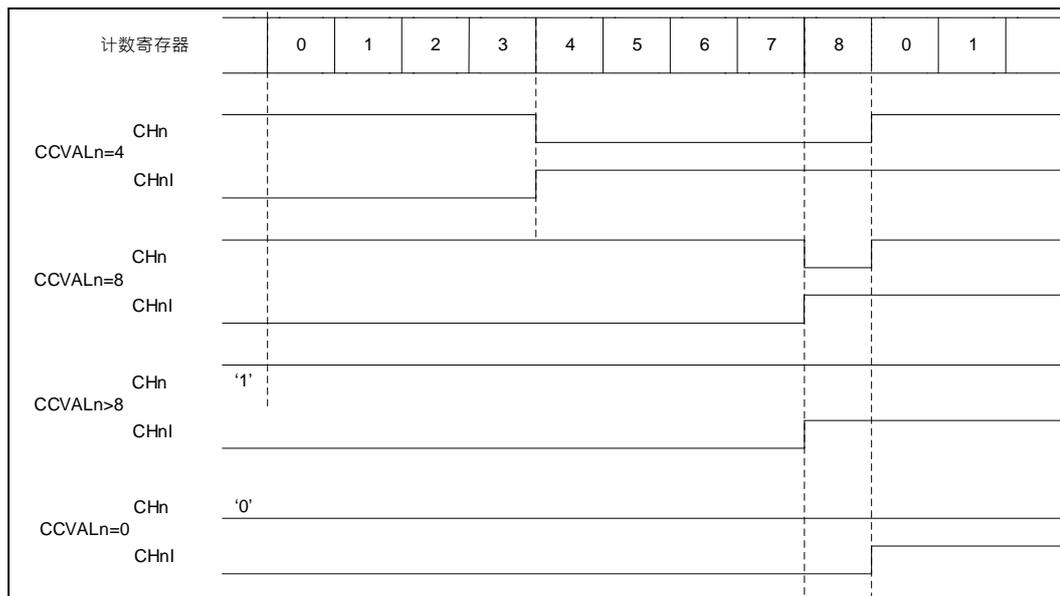


图 17-16 边沿对齐递增计数 PWM 波形 (AR=8)

- ◇ GP32C4Tn_AR=8
- ◇ PWM 模式 1
- ◇ 递减计数配置

当 GP32C4Tn_CON1 寄存器的 DIRSEL 位为高时，计数器递减计数。

下图给出了 GP32C4Tn_AR = 8 时的边沿对齐 PWM 波形。

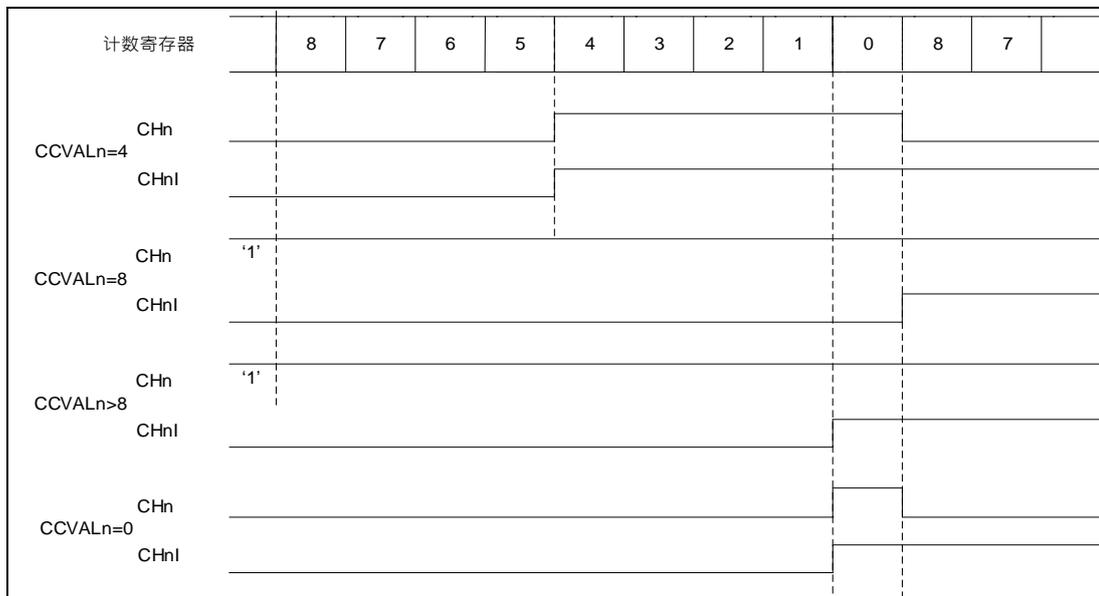


图 17-17 边沿对齐递减计数 PWM 波形 (AR=8)

17.4.6.2 PWM 中央对齐模式

当 **GP32C4Tn_CON1** 寄存器中的 **CMSEL** 位不为"00"时，中央对齐模式有效。计数器是递增、递减计数分别置比较标志位或递增递减都置比较标志位，取决于 **CMSEL** 位的配置。**GP32C4Tn_CON1** 寄存器的方向位 (**DIRSEL**) 是由硬件更新的，软件无法修改。

下图为中央对齐方式产生的 PWM 波形的例子：

- ◇ **GP32C4Tn_AR=0x3F**
- ◇ PWM 模式 1
- ◇ **GP32C4Tn_CON1** 寄存器的 **CMSEL="10"**，在中央对齐模式 2 下，计数器向上计数时会置位比较标志位。

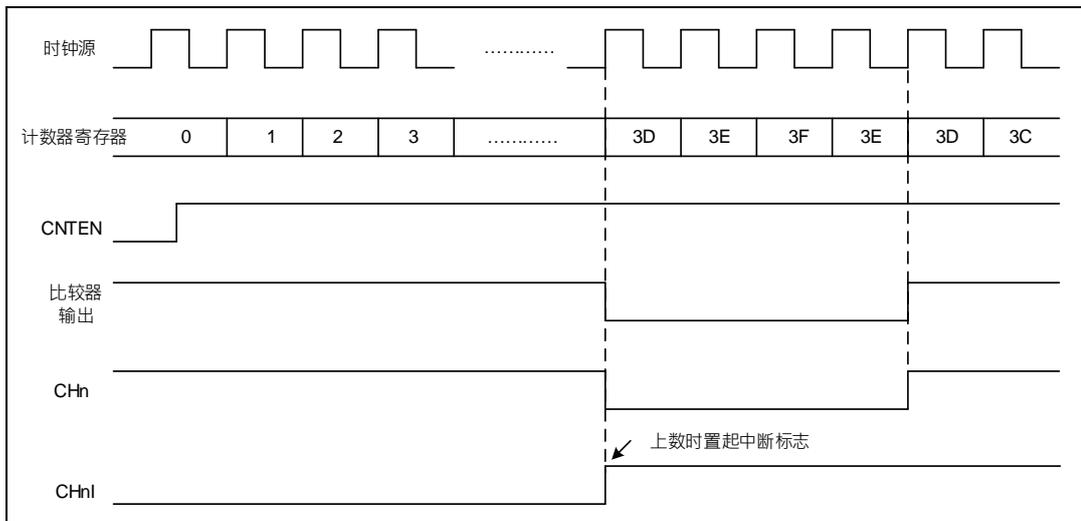


图 17-18 中央对齐 PWM 波形 (AR=0x3F)

中央对齐模式的使用技巧：

- ◇ 当进入中央对齐模式后，当前递增或递减配置生效。计数器递增或递减计数取决于 **GP32C4Tn_CON1** 寄存器的 **DIRSEL** 位的值。此外，软件无法对 **DIRSEL** 和 **CMSEL** 位同时进行修改。
- ◇ 计数器在中央对齐模式下运行时，对计数器写操作可能导致不可预知的结果。特别是：
 - 若向计数器入的值大于自动重载值 ($GP32C4Tn_COUNT > GP32C4Tn_AR$)，计数方向不更新。例如，如果计数器递增计数，写入值后从 0 重新递增计数。
 - 若向计数器写 0 或 **GP32C4Tn_AR** 中的重载值，则计数方向更新，但并没有产生 UEV。
- ◇ 使用中央对齐模式最安全的方式是计数器开始计数前通过软件产生更新事件（置位 **GP32C4Tn_SGE** 寄存器中的 **SGU** 位）且在计数器运行过程中不对计数器写值。

17.4.7 输出比较模式

该功能用于控制输出波形或指示周期时间的结束。

当捕获/比较寄存器和计数器值匹配时，输出比较功能：

- ◇ 输出比较模式（**GP32C4Tn_CHMRn** 寄存器中的 **CHnMOD** 位）和输出极性（**GP32C4Tn_CCEP** 寄存器中的 **CCnPOL** 位）的配置值输出到对应的引脚上。
- ◇ 中断状态寄存器中的标志位置位（**GP32C4Tn_RIF** 寄存器的 **CHnI** 位）。
- ◇ 若相应的中断使能置位，则产生中断（**GP32C4Tn_IER** 寄存器的 **CHnI** 位）。
- ◇ 若相应的使能位置位（**GP32C4Tn_DMAEN** 寄存器的 **CHnDE** 位，**GP32C4Tn_CON2** 寄存器的 **CCDMASEL** 位用于 DMA 请求的选择），则发送 DMA 请求。

GP32C4Tn_CHMRn 寄存器中 **CHnPEN** 位的值可决定 **GP32C4Tn_CCVALn** 寄存器是否带有预装载寄存器。

在输出比较模式中，更新事件 **UEV** 对 **CHn** 的输出没有影响。计时分辨率为计数器的一次计数。输出比较模式同样可以用来输出单个脉冲（单脉冲模式）。

输出比较的配置过程：

1. 选定计数器时钟（内部，外部，预分频）。
2. **GP32C4Tn_AR** 与 **GP32C4Tn_CCVALn** 寄存器中写入预期值。
3. 若需要产生中断请求，置位 **GP32C4Tn_IER** 寄存器中的 **CHnI** 位。
4. 选择输出模式，例如：
 - **CHnMOD** = "011"，当 **CNTV** 与 **CCRVALn** 匹配时，**CHn** 输出翻转。
 - **CHnPEN** = '0'，关闭预载寄存器。
 - **CCnPOL** = '0'，选择有效极性为高。
 - **CCnEN** = '1'，使能输出。
5. **GP32C4Tn_CON1** 寄存器中的 **CNTEN** 位置位，使能计数器。

假设预载寄存器没有使能（**CHnPEN** = '0'，否则 **GP32C4Tn_CCVALn** 影子寄存器只有在下次更新事件发生时才更新）。通过软件方式，**GP32C4Tn_CCVALn** 寄存器的值可随时更新控制输出波形。

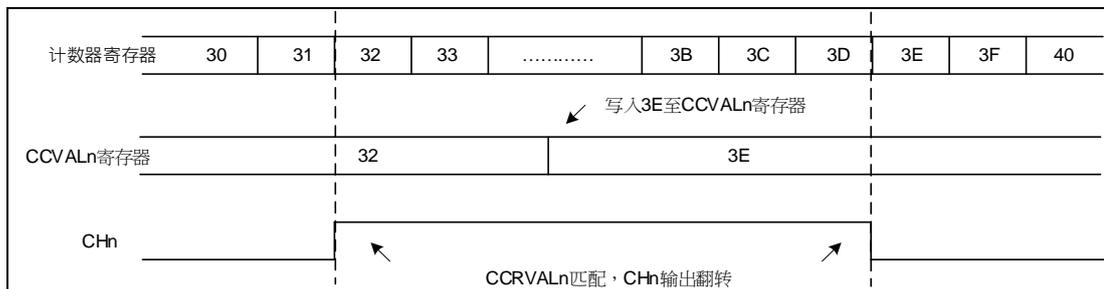


图 17-19 输出比较模式，触发 CHn

17.4.7.1 外部事件清除比较输出

ETF 输入端（**GP32C4Tn_CHMRn** 寄存器的 **CHnOCLREN** 位写'1'）上的高电平，可将给定通道的比较输出信号拉低。在下次更新事件（UEV）发生前，比较输出会一直保持为低。该功能只能应用在输出比较和 PWM 模式中，强制输出模式中不起作用。

ETR 信号可以接到电流控制比较器的输出端。该例中，ETR 须按如下流程配置：

1. 外部时钟源 2 关闭：**GP32C4Tn_SMCON** 寄存器的 **ECM2EN** 位写'0'
2. 外部触发极性（ETPOL）和外部触发滤波器（ETFLT）可根据用户需要配置

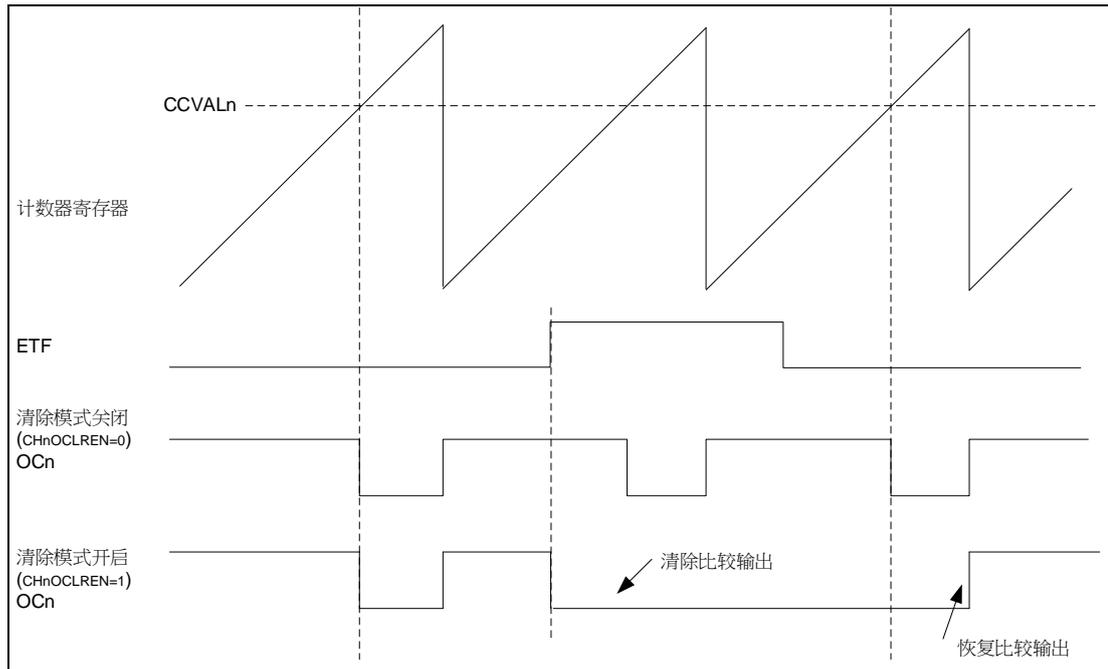


图 17-20 清除比较输出 CHn

17.4.7.2 强制输出模式

在输出模式中（**GP32C4Tn_CHMRn** 寄存器中 **CCnSSEL = "00"**），软件可强制将每个输出比较信号（**CHnN**）改为有效或无效状态，这种修改独立于输出比较寄存器和计数器的比较结果。

为了将某输出比较信号（**CHn**）强制为有效状态，需将相应的 **GP32C4Tn_CHMRn** 寄存器中 **CHnMOD** 位写"101"。因此，比较输出被强制为高（高时为有效状态）且 **CHn** 的值为 **CCnPOL** 极性位的相反值。

例如：**CCnPOL = '0'**（**CHn** 高电平有效），则 **CHn** 被强制为高电平。

对 **GP32C4Tn_CHMRn** 寄存器的 **CHnMOD** 位写"100"，比较输出可被置低。

无论怎样，**GP32C4Tn_CCVALn** 影子寄存器和计数器之间的比较仍然进行，相应的标志位仍可置位。

17.4.8 单脉冲模式

单脉冲模式 (SPMEN) 下, 响应某个触发后, 定时器的输出信道在可配置的延迟时间后产生一个脉冲, 脉冲长度可配。从模式控制器可控制计数器的启动。脉冲波形可在输出比较模式和 PWM 模式下产生。置位 **GP32C4Tn_CON1** 寄存器的 SPMEN 位可选择单脉冲模式。计数器会在下次更新事件 UEV 产生时自动停止。

只有比较值不同于计数器初始值时, 单脉冲才可以正确的产生。计数器开始计数前 (定时器等待触发), 必须如下配置:

- ◇ 递增计数: $CNTV < CCVALn \leq AR$ (特别地, $0 < CCVALn$)
- ◇ 递减计数: $CNTV > CCVALn$

基于 PWM 模式设置单脉冲输出波形的步骤如下:

- ◇ 设置 **GP32C4Tn_CHMRn** 寄存器的 CHnMOD 位, 选择 PWM 模式 1 或 2;
- ◇ 设置 **GP32C4Tn_CCEPn** 寄存器的 CCnPOL 位, 选择通道端口 CHn 的输出极性;
- ◇ 设置 **GP32C4Tn_CON1** 寄存器的 DIRSEL, CMSEL, SPMEN 位, 配置为递增或递减计数, PWM 普通波形模式, 单脉冲模式使能;
- ◇ 设置 **GP32C4Tn_CHMR** 寄存器的 CH1PEN = 1, **GP32C4Tn_CON1** 寄存器的 ARPEN = 1, 使能比较寄存器和计数重载寄存器的缓冲功能 (也可以根据实际情况不使能缓冲);
- ◇ 设置 **GP32C4Tn_CCVALn** 寄存器和 **GP32C4Tn_AR** 寄存器, 配置单脉冲输出延时和脉宽时间;
- ◇ 设置 **GP32C4Tn_SGE** 寄存器的 SGU 位来产生一个更新事件;
- ◇ 设置 **GP32C4Tn_CON1** 寄存器的 CNTEN=1 来启动计数器, 也可以在触发模式下, 通过外部触发输入信号来触发硬件自动设置 CNTEN=1。

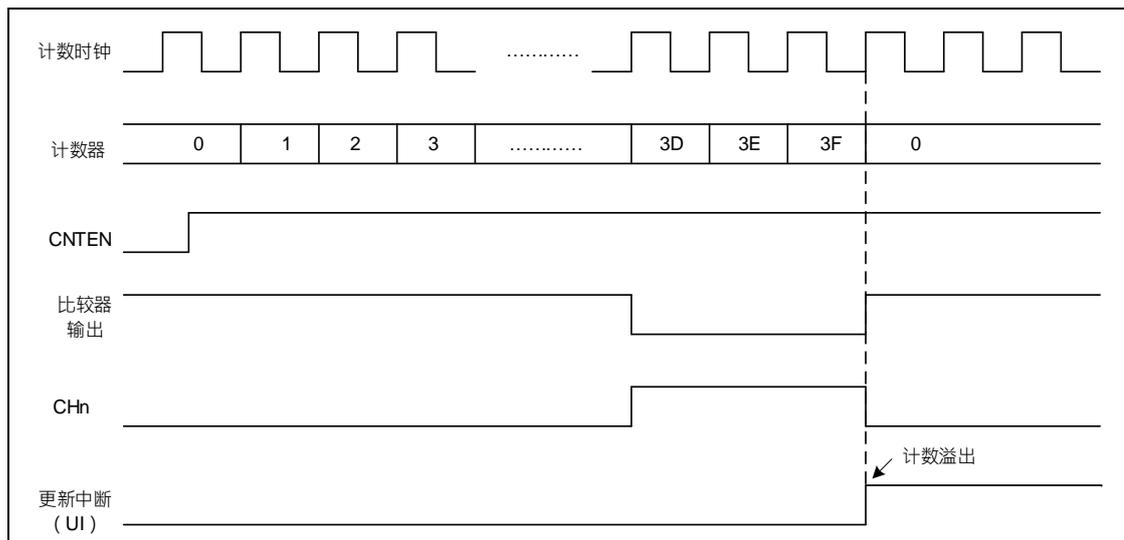


图 17-21 单脉冲模式

17.4.9 编码器接口模式

编码器接口模式的三种配置：若计数器只根据 I2 上的边沿计数，则 **GP32C4Tn_SMCON** 寄存器中的 **SMODS = "001"**；若计数器只根据 I1 上的边沿计数，则 **GP32C4Tn_SMCON** 寄存器中的 **SMODS = "010"**；若计数器同时根据 I1 和 I2 上的边沿计数，则 **GP32C4Tn_SMCON** 寄存器中的 **SMODS = "011"**。

配置 **GP32C4Tn_CCEP** 寄存器中的 **CC1POL** 和 **CC2POL** 位的值可选择 I1 和 I2 的极性。如果需要，也可以配置输入滤波器。

CH1 和 **CH2** 端口作为增量编码器的接口。当计数器使能时，计数器根据 I1 或 I2 上滤波后的有效电平变化时钟计数。I1 和 I2 滤波后的有效信号顺序会产生计数脉冲及方向信号。计数器是递增或递减计数由信号的跳变顺序决定，**GP32C4Tn_CON1** 寄存器中的 **DIRSEL** 计数方向位由硬件自动更新。

编码器接口模式的工作方式类似于一个带有方向选择的外部时钟。计数器在 0 到 **GP32C4Tn_AR** 寄存器中的自动重载值之间连续计数。因此，必须在开始计数前配置 **GP32C4Tn_AR** 寄存器。同样的，捕获器、预分频器、重复计数器、触发输出的特性正常工作。设定编码模式和选择外部时钟源 2 不兼容，不可以同时选择。

该模式下，计数器会根据增量式编码器的速度和方向自动修改，计数器的值反映的是编码器的位置。计数方向对应着连接传感器的旋转方向。

下表列出了所有的可能组合，假设 I1 和 I2 不同时变换。

有效边沿	有效边沿相对信号的电平 (I1 滤波信号对应 I2, I2 滤波信号对应 I1)	I1 滤波信号		I2 滤波信号	
		上升	下降	上升	下降
仅在 I1 计数	高	下降	上升	不计数	不计数
	低	上升	下降	不计数	不计数
仅在 I2 计数	高	不计数	不计数	上升	下降
	低	不计数	不计数	下降	上升
在 I1 和 I2 上计数	高	下降	上升	上升	下降
	低	上升	下降	下降	上升

表 17-1 计数方向与编码器信号的关系

外部增量编码器可直接与 MCU 连接，无需外部接口逻辑。而比较器通常用于将编码器的差分输出转换为数字信号，这极大地增加了抗噪声能力。编码器的第三个输出端用于指示机械零点，可以连接到外部中断输入引脚以触发一次计数复位。

下图给出了计数器操作的例子，给出了计数信号的产生和方向控制。同样给出了选择双边沿时，输入抖动如何被补偿。输入抖动可能发生在传感器靠近切换点处。

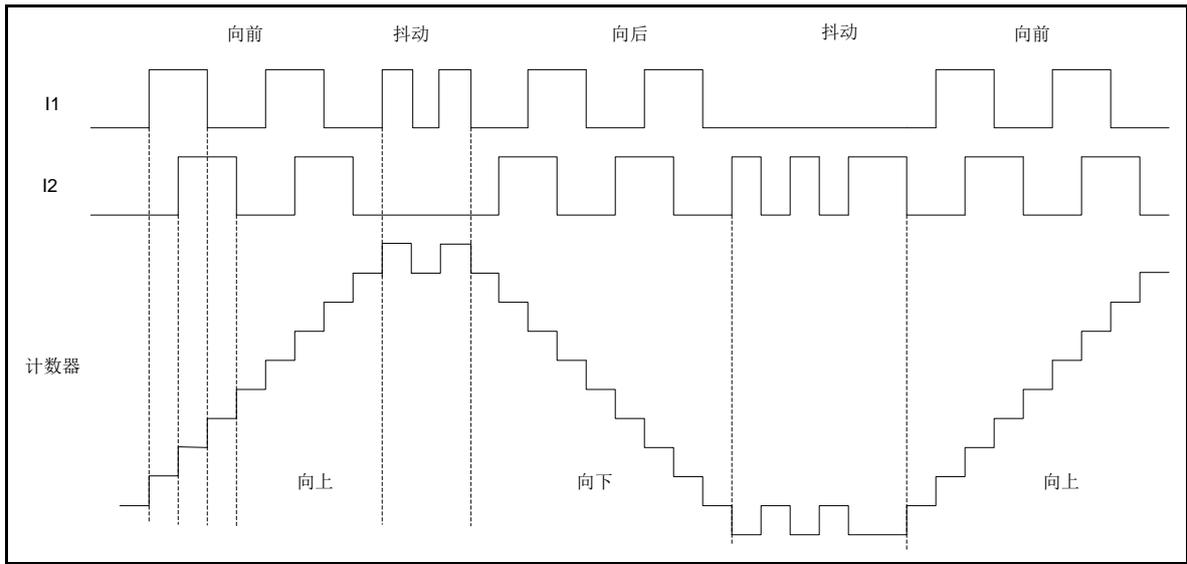


图 17-22 编码器接口模式下的计数操作

下图给出了计数器在 I1 滤波信号极性反相时的计数过程（除了 CC1POL = '1'，其他配置与上面一致）

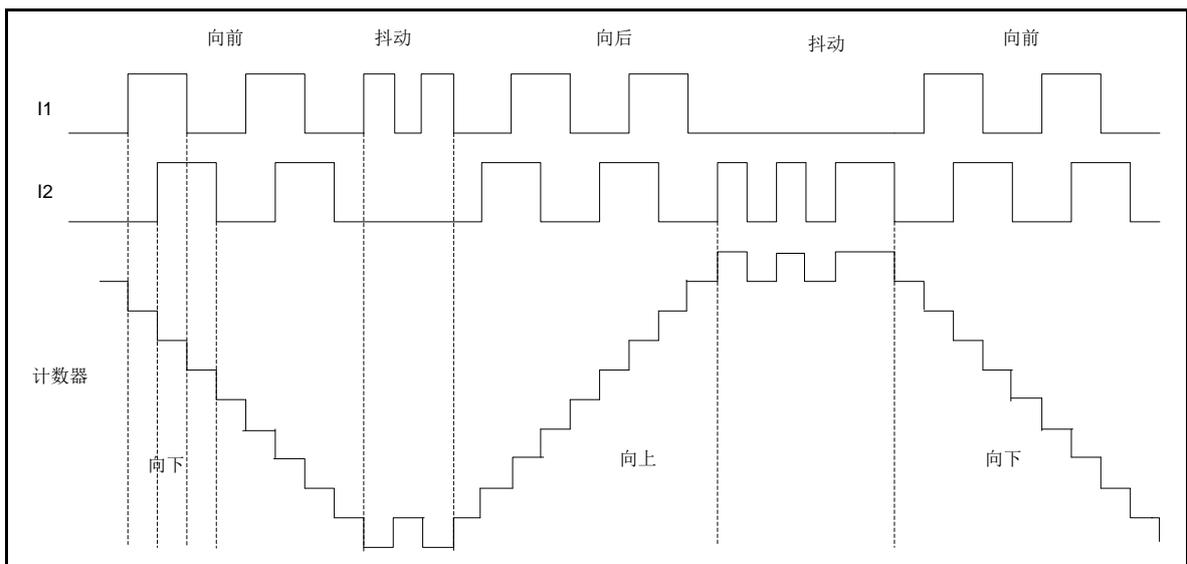


图 17-23 滤波后极性反相时编码器接口例子

当配置为编码器接口模式时，定时器可提供传感器的当前位置讯息。配置一个额外定时器为捕获模式，用于测量两个编码器事件的间隔，根据间隔时长获取动态讯息（速度、加速度、减速度）。编码器用于指示机械零点的输出就是此用处。根据编码器两个事件间隔，可以周期性的读取计数器的值。如果允许，可以将计数器值锁存到第三个输入捕获寄存器（捕获信号必须是周期性的且可由其它定时器产生）。条件允许时，可通过实时时钟产生 DMA 请求的方式读取计数器值。

17.4.10 输入异或功能

通过 **GP32C4Tn_CON2** 寄存器中 **I1SEL** 位，可将通道 1 的输入滤波器连接到 XOR 门的输出端，XOR 门联合了 CH1、CH2 和 CH3 三个输入引脚。

XOR 输出用于定时器的所有输入功能，如触发或输入捕获。该功能参见下节的霍尔传感器接口。

17.4.11 霍尔传感器接口

使用高级控制定时器产生 PWM 信号驱动马达，用另一个定时器作为“接口定时器”来连接霍尔传感器，请参见下图。3 个定时器输入脚(CH1、CH2、CH3)通过一个异或门连接到 I1 输入通道(通过设置 **GP32C4Tn_CON2** 寄存器中的 **I1SEL** 位来选择)，“接口定时器”捕获这个信号。

从模式控制器被配置为复位模式，从输入是 I1F 双边沿。这样每当 3 个输入之一变化时，计数器从 0 重新开始计数。由此产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”模式下，捕获/比较信道 1 被配置为捕获模式，捕获信号为 I1(捕获/比较通道(如：通道 1 输入部分))。捕获值反映了输入端两次变化之间的时间间隔，指示出了马达速度的讯息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级控制定时器(AD16C4Tn)各个通道的属性，AD16C4Tn 产生 PWM 信号驱动马达。因此，“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGOUT 输出被送到高级控制定时器(AD16C4Tn)。

举例：霍尔输入连接到定时器，要求在每次霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器的 PWM 配置。

- ◇ 置 **GP32C4Tn_CON2** 寄存器的 **I1SEL** 位为'1'，配置三个定时器输入逻辑异或到 I1 输入，
- ◇ 时基编程：置 **GP32C4Tn_AR** 为其最大值(计数器必须通过 I1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- ◇ 设置信道 1 为捕获模式(选中 I1)：置 **GP32C4Tn_CHMR1** 寄存器中 **CC1SSEL=01**，如果需要，还可以设置数字滤波器。
- ◇ 设置信道 2 为 PWM 模式 2，带指定的延时：置 **GP32C4Tn_CHMR1** 寄存器中的 **CH2OMOD =111** 和 **CC2SEL =00**。
- ◇ 选择 **CH2REF** 作为 TRGOUT 上的触发输出：置 **GP32C4Tn_CON2** 寄存器中的 **MMSEL=101**。

在高级控制定时器 AD16C4Tn 中，正确的 IT 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(**AD16C4Tn_CON2** 寄存器中 **CCPCNTEN=1**)，由触发输入控制 COM 事件(**AD16C4Tn_CON2** 寄存器中 **CCUSEL=1**)。在一次 COM 事件后，下一步再写入 PWM 控制位(**CCnEN**、**CHnOCLREN**)，这可以在处理 **CH2REF** 上升沿的中断子程序里实现。

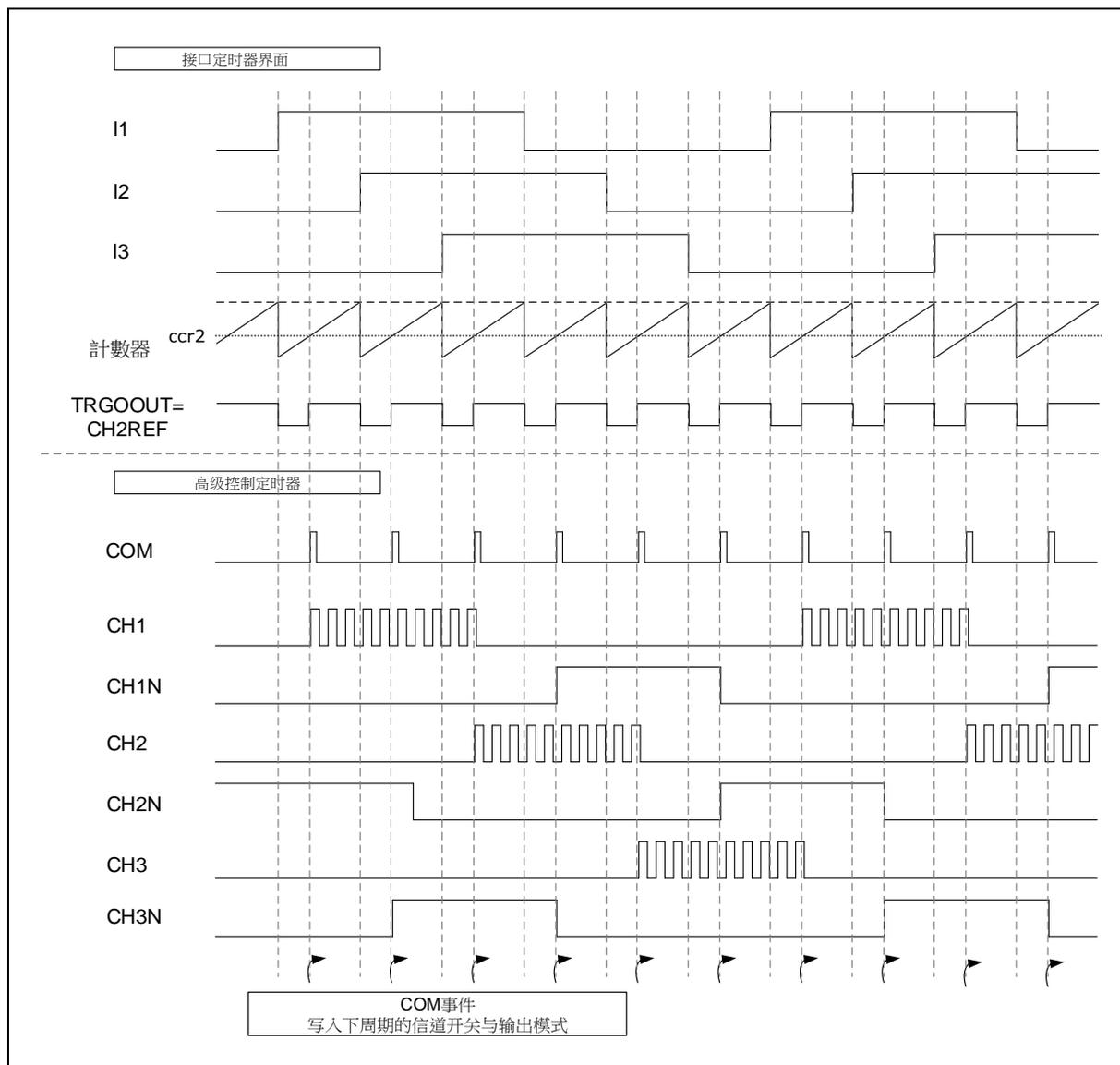


图 17-24 霍尔传感器接口范例

17.4.12 外部触发的同步

GP32C4Tn 定时器可在多种模式下与外部触发同步：复位模式、门控模式及触发模式。

17.4.12.1 复位模式

计数器及其预分频器可以在响应触发输入事件时重新初始化。此外，若 **GP32C4Tn_CON1** 寄存器的 **UERSEL** 位为低时会产生一次更新事件 **UEV**。所有预载寄存器 (**GP32C4Tn_AR**, **GP32C4Tn_CCVALn**) 都会因更新事件 **UEV** 而被更新。

在下面例子中，I1 输入端的上升沿让递增计数被清空：

- ◇ 配置信道 1 上检测 I1 上的上升沿。配置输入滤波周期（本例无需滤波器，故 **I1FLT** = "0000"）。触发捕获分频器没有使用，无需配置。**CC1SSEL** 位只选择输入捕获源，**GP32C4Tn_CHMR1** 寄存器中 **CC1SSEL** = "01"。**GP32C4Tn_CCEP** 寄存器中 **CC1POL** = 0 以确定极性（只检测上升沿）。
- ◇ 定时器配置为复位模式：**GP32C4Tn_SMCON** 寄存器中 **SMODS** = "100"。选择 I1 作为输入源：**GP32C4Tn_SMCON** 寄存器中 **TSSEL** = "101"。
- ◇ 启动计数器：**GP32C4Tn_CON1** 寄存器中 **CNTEN** = '1'。

计数器依据内部时钟开始计数，正常计数直到 I1 上出现上升沿。当 I1 上出现上升沿时，计数器会被清零且从 0 重新开始计数。同时，标志位置位（**GP32C4Tn_RIF** 寄存器中 **TRGI** 位），如果中断及 DMA 使能（取决于 **GP32C4Tn_IER** 寄存器中的 **TRGI** 和 **GP32C4Tn_DMAEN** 寄存器中的 **TRGIDE** 位），会发送中断及 DMA 请求。

下图给出了当自动重载寄存器 **GP32C4Tn_AR** = 0x36 时的信号变化。由于 I1 输入的再同步电路，I1 上的上升沿和计数器实际复位之间会存在延时。

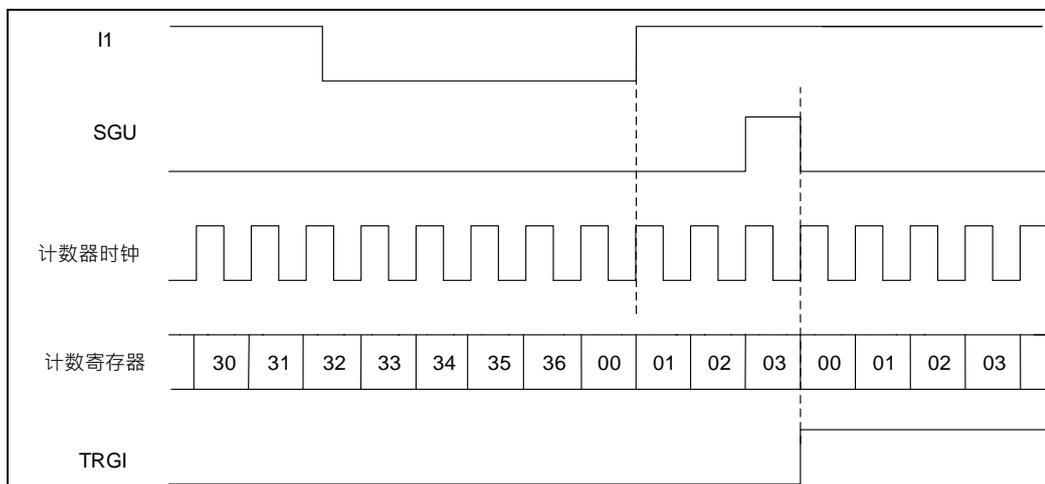


图 17-25 复位模式控制电路

17.4.12.2 门控模式

计数器根据选中的输入电平被使能。

下面的例子中，计数器只在 I1 输入为低电平时才递增计数：

- ◇ 配置信道 1 在 I1 上检测低电平。配置输入滤波周期(本例不需要滤波器, I1FLT = "0000")。触发捕获分频器没有使用, 无需配置。GP32C4Tn_CHMR1 寄存器中的 CC1SSEL = "01", 选择输入捕获源。GP32C4Tn_CCEP 寄存器中 CC1POL = '1', 确认极性(只检测低电平)。
- ◇ 配置定时器为门控模式: GP32C4Tn_SMCON 寄存器中 SMODS = "101"。选择 I1 作为输入源: GP32C4Tn_SMCON 寄存器中 TSSEL = "101"。
- ◇ 使能计数器: GP32C4Tn_CON1 寄存器中 CNTEN = '1' (门控模式中, 如果 CNTEN = '0', 无论触发输入为何电平, 计数器都不会启动)。

只要 I1 为低电平, 计数器依据内部时钟开始计数, 一旦 I1 为高则停止计数。由于 I1 输入端的再同步电路的原因, I1 上出现上升沿和计数器实际停止之间会有一定的延时。

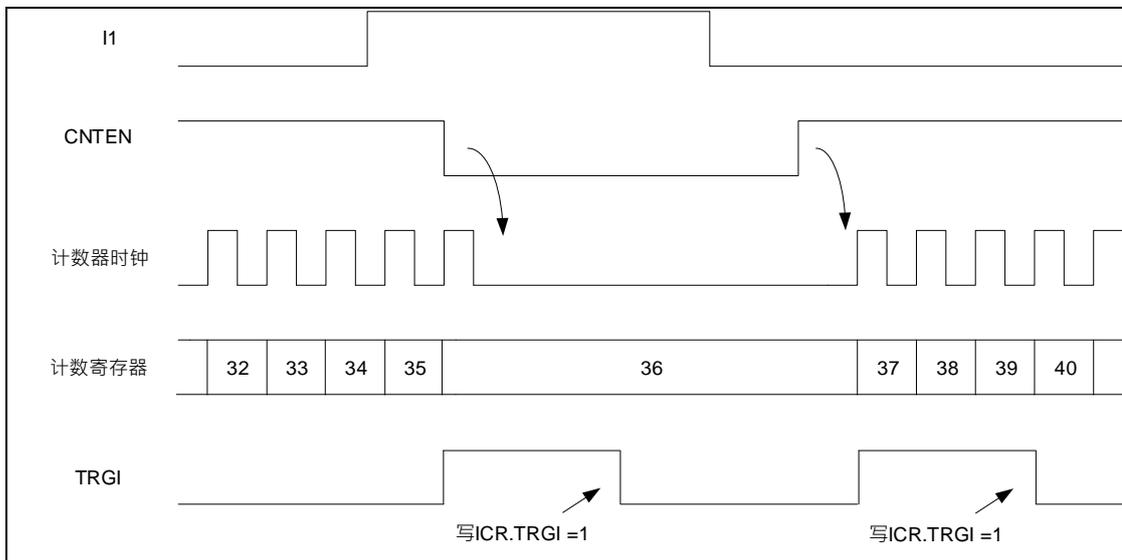


图 17-26 门控模式控制电路

17.4.12.3 触发模式

输入端选中的事件可以使能计数器。

下面的例子中，I2 输入端上的上升沿可以启动递增计数：

- ◇ 配置信道 2 可以检测 I2 上的上升沿。配置滤波时间（本例不需要滤波，I2FLT = "0000"）。触发捕获分频器没有使用，无需配置。GP32C4Tn_CHMR1 寄存器中 CC2SEL = "01",用于选择捕获源。GP32C4Tn_CCEP 寄存器中 CC2POL = '0', 确认极性（只检测高电平）。
- ◇ 配置定时器为触发模式：GP32C4Tn_SMCON 寄存器中 SMODS = "110"。GP32C4Tn_SMCON 寄存器中 TSSEL = "110", 用于选择输入源。

I2 上出现上升沿时，计数器开始依据内部时钟计数并置位 TRGI 标志位。

由于 I2 输入的再同步原因，I2 上出现上升沿和计数器实际停止之间会有一定的延时。

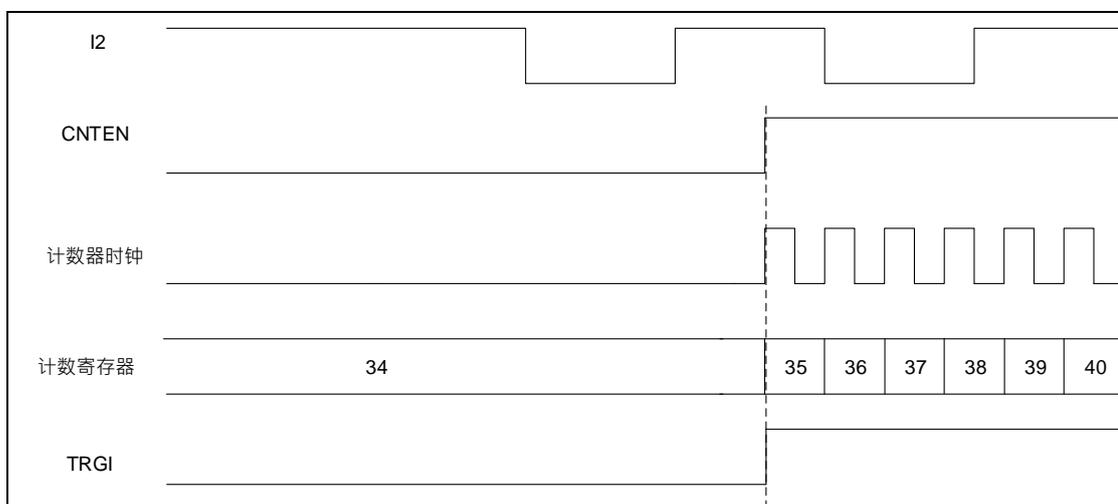


图 17-27 触发模式控制电路

17.4.12.4 选择外部时钟源 2 的触发模式

外部时钟源 2 可和其他模式一起使用（除编码模式）。ETR 信号可作为外部时钟输入，另一个输入可选择为触发输入（复位模式、门控模式或触发模式）。不推荐用 GP32C4Tn_SMCON 寄存器的 TSSEL 位选择 ETR 作为 TI。

下面的例子中，一旦 I1 上出现上升沿时，计数器会依据 ETR 信号的每个上升沿递增计数。

- ◇ 通过 GP32C4Tn_SMCON 寄存器，配置外部触发输入电路，过程如下：

ETFLT = "000": 无滤波

ETPOL = '0': 检测 ETR 的上升沿，ECM2EN = '1'使能外部时钟模式 2

- ◇ 配置信道 1 检测 I 的上升沿，过程如下：

I1FLT = "0000": 无滤波。

触发捕获分频器没有使用，无需配置。

GP32C4Tn_CHMR1 寄存器中 CC1SSEL = "01"选择输入捕获源，GP32C4Tn_CCEP 寄存器的 CC1POL = '0'确认极性（只检测上升沿）。

- ◇ 配置定时器为触发模式：GP32C4Tn_SMCON 寄存器中 SMODS = "110"。

GP32C4Tn_SMCON 寄存器中 TSSEL = "101"选择 I1 作为输入源。

I1 上出现上升沿时，计数器使能且 TRGI 标志位置位，然后计数器根据 ETR 上的上升沿开始计数。

由于 ETF 输入再同步电路的原因，ETR 信号的上升沿和实际计数器的复位会有延时。

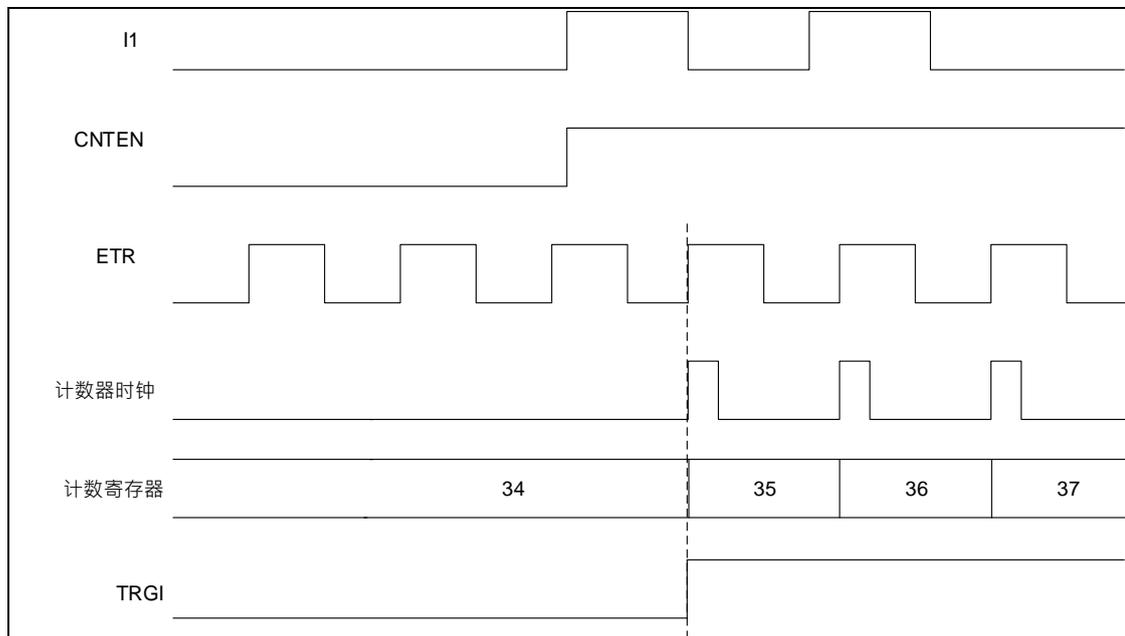


图 17-28 外部时钟源 2+触发模式下的控制电路

17.4.13 定时器同步

所有定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况

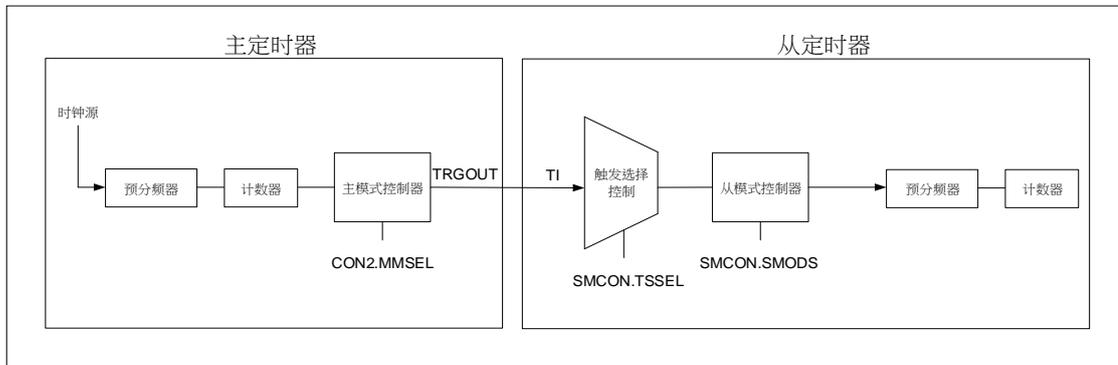


图 17-29 主/从定时器范例

17.4.13.1 使用一个定时器去使能其他定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考下图的连接。只当定时器 1 的 CH1REF 为高时，定时器 2 才对分频后的内部时钟计数。

配置定时器 1 为主模式，送出它的输出比较参考信号 (CH1REF) 为触发输出 (AD16C4T1_CON2 寄存器的 MMSEL=100)

- ◇ 配置定时器 1 的 CH1REF 波形 (AD16C4T1_CHMR1 寄存器)
- ◇ 配置定时器 2 从定时器 1 获得输入触发 (GP32C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为门控模式 (GP32C4T1_SMCON 寄存器的 SMODS=101)
- ◇ 配置 GP32C4T1_CON1 寄存器的 CNTEN=1 以使能定时器 2
- ◇ 配置 AD16C4T1_CON1 寄存器的 CNTEN=1 以启动定时器 1

注：定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

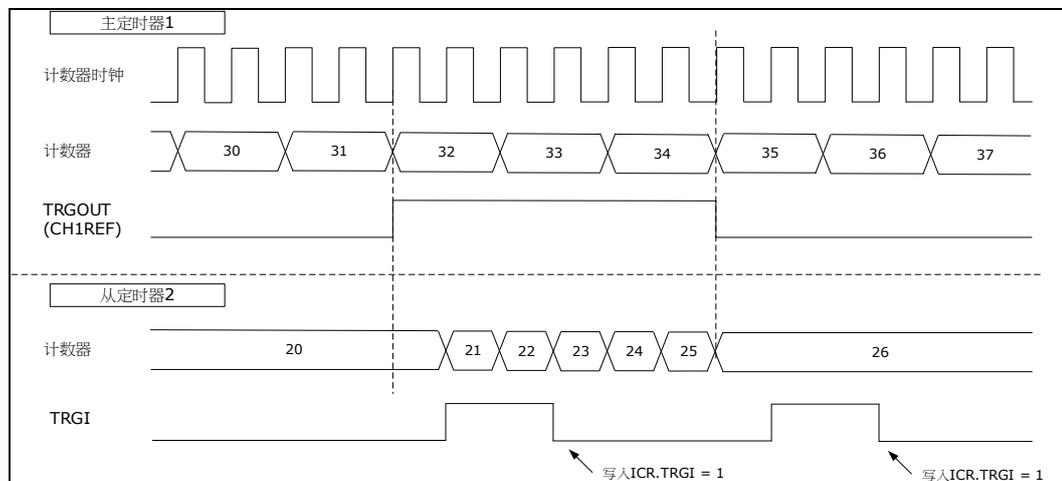


图 17-30 门控从定时器使用主定时器 CH1REF

在上图的例子中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 **AD16C4T1_SGE** 寄存器的 **SGU** 位即可复位定时器。

在下一个例子中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写'0'到 **AD16C4T1_CON1** 的 **CNTEN** 位将禁止定时器 1，定时器 2 随即停止。

- ◇ 配置定时器 1 为主模式，送出 **CNTEN** 位做为触发输出(**AD16C4T1_CON2** 寄存器的 **MMSEL=001**)。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(**GP32C4T1_SMCON** 寄存器的 **TSSEL=000**)
- ◇ 配置定时器 2 为门控模式(**GP32C4T1_SMCON** 寄存器的 **SMODS=101**)
- ◇ 配置 **AD16C4T1_SGE** 寄存器的 **SGU=1**，复位定时器 1。
- ◇ 配置 **GP32C4T1_SGE** 寄存器的 **SGU=1**，复位定时器 2。
- ◇ 写'0xE7'至定时器 2 的计数器(**GP32C4T1_COUNT**)，初始化它为 0xE7。
- ◇ 配置 **GP32C4T1_CON1** 寄存器的 **CNTEN=1** 以使能定时器 2。
- ◇ 配置 **AD16C4T1_CON1** 寄存器的 **CNTEN=1** 以启动定时器 1。
- ◇ 配置 **AD16C4T1_CON1** 寄存器的 **CNTEN=0** 以停止定时器 1。

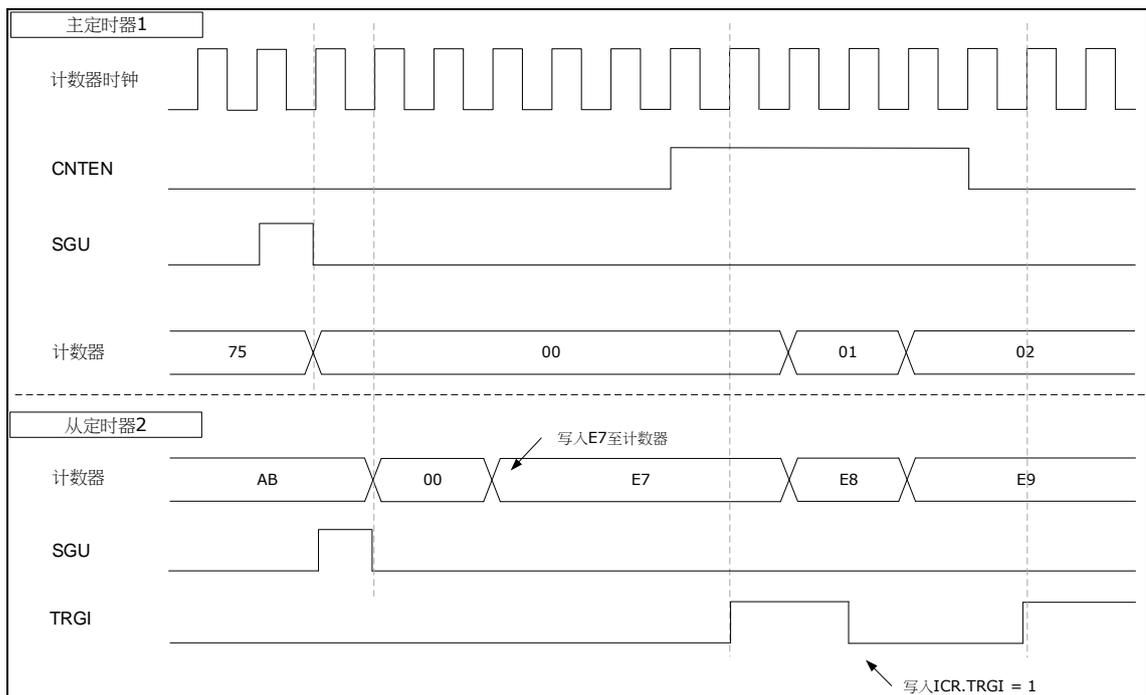


图 17-31 通过使能定时器 1 可以控制定时器 2

17.4.13.2 使用一个定时器去开启其他定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值(可以是非 0)按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CNTEN 位被自动地置'1'，同时计数器开始计数直到写'0'到 GP32C4T1_CON1 寄存器的 CNTEN 位。

- ◇ 配置定时器 1 的周期(AD16C4T1_AR 寄存器)。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP32C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为触发模式(GP32C4T1_SMCON 寄存器的 SMODS=110)
- ◇ 配置 AD16C4T1_CON1 寄存器的 CNTEN=1 以启动定时器 1。

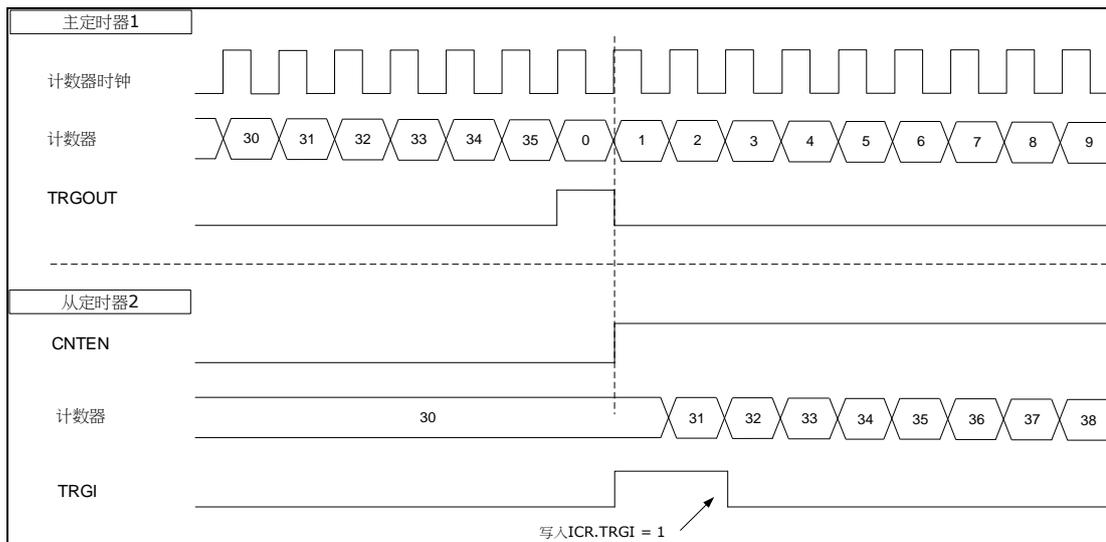


图 17-32 触发中从定时器使用主定时器更新事件

在上图例子中，可以在启动计数之前初始化两个计数器。上图显示在上上图相同配置情况下，使用触发模式而不是门控模式(GP32C4T1_SMCON 寄存器的 SMODS=110)的动作。

17.4.13.3 使用外部触发同步开始两个定时器

这个例子中当定时器 1 的 I1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见下图。为保证计数器的对齐，定时器 1 必须配置为主/从模式(对应 I1 为从，对应定时器 2 为主)：

- ◇ 配置定时器 1 为主模式，送出它的使能做为触发输出(AD16C4T1_CON2 寄存器的 MMSEL=001)。
- ◇ 配置定时器 1 为从模式，从 I1 获得输入触发(AD16C4T1_SMCON 寄存器的 TSSEL='101')。
- ◇ 配置定时器 1 为触发模式(AD16C4T1_SMCON 寄存器的 SMODS='110')。
- ◇ 配置定时器 1 为主/从模式，AD16C4T1_SMCON 寄存器的 MSCFG='1'。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP32C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为触发模式(GP32C4T1_SMCON 寄存器的 SMODS='110')。

当定时器 1 的 I1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TRGI 标志也同时被设置。

注：在这个例子中，在启动之前两个定时器都被初始化(设置相应的 SGU 位)，两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器(AD16C4T1_COUNT)在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNTEN 和计数器时钟之间有个延迟。

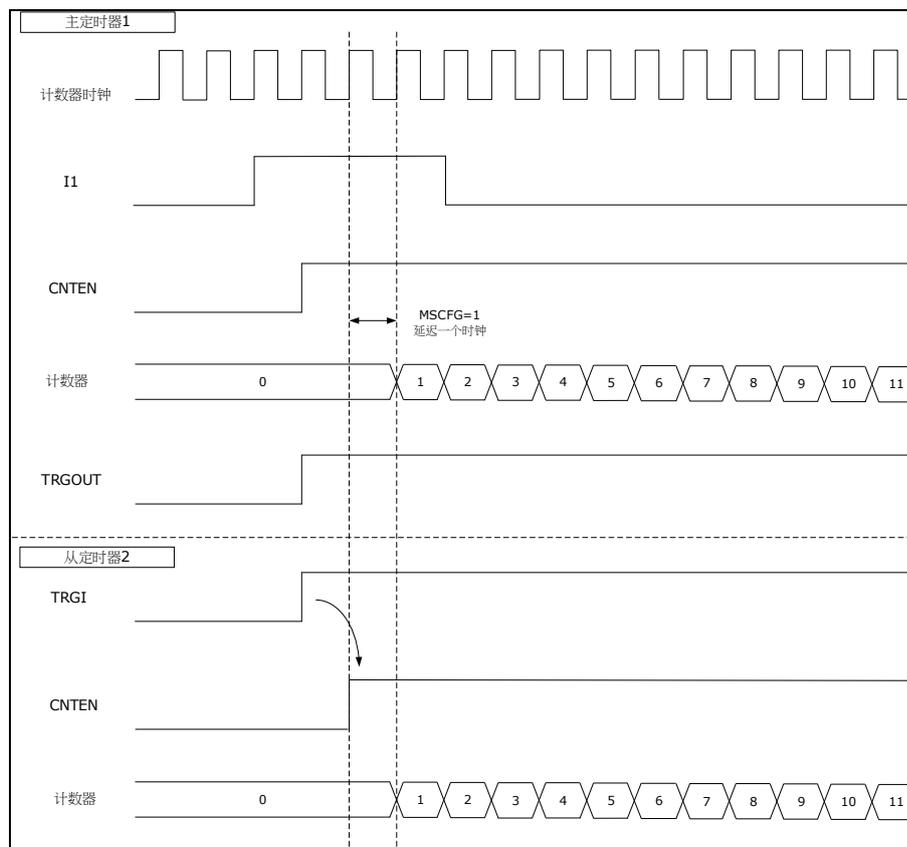


图 17-33 使用定时器 1 的 I1 输入触发定时器 1 和定时器 2

17.4.14 调试模式

当微控制器进入调试模式(Cortex™-M0 核停止运行)，计数器停止计数。

17.5 特殊功能寄存器

17.5.1 寄存器列表

GP32C4T 寄存器列表			
名称	偏移地址	类型	描述
GP32C4Tn_CON1	00 _H	R/W	控制寄存器 1
GP32C4Tn_CON2	04 _H	R/W	控制寄存器 2
GP32C4Tn_SMCON	08 _H	R/W	从模式控制寄存器
GP32C4Tn_IER	0C _H	W1	中断使能寄存器
GP32C4Tn_IDR	10 _H	W1	中断禁止寄存器
GP32C4Tn_IVS	14 _H	R	中断有效状态寄存器
GP32C4Tn_RIF	18 _H	R	原始中断标志寄存器
GP32C4Tn_IFM	1C _H	R	中断屏蔽标志寄存器
GP32C4Tn_ICR	20 _H	C_W1	中断标志清除寄存器
GP32C4Tn_SGE	24 _H	W1	软件生成事件寄存器
GP32C4Tn_CHMR1	28 _H	R/W	捕获/比较模式寄存器 1
GP32C4Tn_CHMR2	2C _H	R/W	捕获/比较模式寄存器 2
GP32C4Tn_CCEP	30 _H	R/W	捕获/比较使能极性寄存器
GP32C4Tn_COUNT	34 _H	R/W	计数器
GP32C4Tn_PRES	38 _H	R/W	时钟预分频器
GP32C4Tn_AR	3C _H	R/W	自动重载寄存器
GP32C4Tn_CCVAL1	44 _H	R/W	通道捕获/比较寄存器 1
GP32C4Tn_CCVAL2	48 _H	R/W	通道捕获/比较寄存器 2
GP32C4Tn_CCVAL3	4C _H	R/W	通道捕获/比较寄存器 3
GP32C4Tn_CCVAL4	50 _H	R/W	通道捕获/比较寄存器 4
GP32C4Tn_DMAEN	58 _H	R/W	DMA 事件使能寄存器

17.5.2 寄存器描述

17.5.2.1 控制寄存器 1 (GP32C4Tn_CON1)

控制寄存器 1 (GP32C4Tn_CON1)																																	
偏移地址: 00H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																DBGSEL																	

CMSEL	Bits 6-5	R/W	<p>中央对齐模式选择</p> <p>00: 边沿对齐模式，根据 CON1 寄存器的 DIRSEL 位，使得计数器递增或递减</p> <p>01: 中央对齐模式 1，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，只在计数器向下计数时被设置</p> <p>10: 中央对齐模式 2，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，只在计数器向上计数时被设置</p> <p>11: 中央对齐模式 3，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，在计数器向上和向下计数时皆会被设置</p>
DIRSEL	Bit 4	R/W	<p>计数方向选择</p> <p>当计数器配置为中央对齐模式时，此位只能读取计数器的计数方向</p> <p>0: 计数器向上计数</p> <p>1: 计数器向下计数</p>
SPMEN	Bit 3	R/W	<p>单脉冲模式</p> <p>0: 单脉冲模式禁止，计数器不停止</p> <p>1: 单脉冲模式使能，计数器在发生下一次更新事件时，清除 CNTEN 位，计数器停止</p>
UERSEL	Bit 2	R/W	<p>更新事件请求来源选择</p> <p>设置更新事件(UEV)的来源</p> <p>0: 若使能 UI 中断或 DMA 请求，下列事件中产生更新事件请求</p> <ul style="list-style-type: none"> - 计数器溢出/下溢 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 <p>1: 若使能 UI 中断或 DMA 请求，只在计数器溢出/下溢时产生更新事件请求</p>
DISUE	Bit 1	R/W	<p>更新事件禁止</p> <p>设置更新事件(UEV)的产生</p> <p>0: 更新事件(UEV) 使能，下列事件中产生更新事件请求</p> <ul style="list-style-type: none"> - 计数器溢出/下溢 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 <p>缓冲寄存器载入预装载值</p> <p>1: 更新事件(UEV) 禁止，不产生更新事件请求，AR、PRES、CCVALn 寄存器保持数值</p> <p>禁止更新事件时，设置 SGE 寄存器的 SGU 位或</p>

			从模式中产生的复位请求，计数器和预分频器仍会被重新初始化
CNTEN	Bit 0	R/W	<p>计数器使能</p> <p>使能计数器后，在外部时钟模式、门控模式和编码模式才能运作。触发模式则可以由硬件设置 CNTEN 位</p> <p>0: 计数器禁止</p> <p>1: 计数器使能</p>

17.5.2.2 控制寄存器 2 (GP32C4Tn_CON2)

控制寄存器 2 (GP32C4Tn_CON2)																															
偏移地址: 04H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								I1SEL	MMSEL<2:0>			CCDMASEL			

—	Bits 31-8	—	—
I1SEL	Bit 7	R/W	<p>选择 I1 引脚功能</p> <p>0: GP32C4Tn_CH1 引脚与 I1 输入连接</p> <p>0: GP32C4Tn_CH1, CH2 和 CH3 引脚与 I1 输入 (XOR) 连接。</p>
MMSEL	Bits 6-4	R/W	<p>主模式选择</p> <p>设置在主模式下发送到从定时器的同步信号 (TRGOUT)与 ADC 输入</p> <p>000: 复位 - 设置 SGE 寄存器信号用于同步触发输出(TRGOUT)。从模式的复位触发产生的复位信号(TRGOUT)则与实际信号相差一个时钟</p> <p>001: 使能 - 计数器的使能信号 CNTEN 用于同步触发输出(TRGOUT), 可用于同步使能数个定时器。门控模式下, 是使用 CON1 寄存器的 CNTEN 位与触发输入信号逻辑产生。当计数器使能信号受控于触发输入时, TRGOUT 上会有一个时钟延迟, 可设置 SMCON 寄存器的 MSCFG 位延迟一个时钟同步定时器计数器</p> <p>010: 更新事件 - 更新事件被用于同步触发输出(TRGOUT)。一个主定时器的更新事件可当作从定时器的预分频器时钟</p> <p>011: 比较脉冲 - 在发生一次捕获或一次比较成功时, 当要设置 CH1I 标志时, 触发输出送出一个正脉冲(TRGOUT)</p> <p>100: 比较信号 - CH1REF 信号用于触发输出 (TRGOUT)</p> <p>101: 比较信号 - CH2REF 信号用于触发输出 (TRGOUT)</p> <p>110: 比较信号 - CH3REF 信号用于触发输出 (TRGOUT)</p> <p>111: 比较信号 - CH4REF 信号用于触发输出 (TRGOUT)</p>
CCDMASEL	Bit 3	R/W	<p>捕获/比较事件的 DMA 选择</p>

			0: 当发生 CHn 事件时, 设置 CHn DMA 请求 1: 当发生更新事件时, 设置 CHn DMA 请求
—	Bit 2-0	R/W	—

			<p>1001: 采样频率$f_{DTS} / 8, N = 8$ 1010: 采样频率$f_{DTS} / 16, N = 5$ 1011: 采样频率$f_{DTS} / 16, N = 6$ 1100: 采样频率$f_{DTS} / 16, N = 8$ 1101: 采样频率$f_{DTS} / 32, N = 5$ 1110: 采样频率$f_{DTS} / 32, N = 6$ 1111: 采样频率$f_{DTS} / 32, N = 8$</p>
MSCFG	Bit 7	R/W	<p>主/从模式 0: 写入0无效 1: 延迟触发输入 (TI) 上的事件来允许当前计时器和其从器件之间的同步。该设置有效用于使用单个外部事件来同步多个计时器。</p>
TSSEL	Bits 6-4	R/W	<p>触发选择 设置触发选择，用于同步寄存器 000: 内部触发 0 (IT0) 001: 内部触发 1 (IT1) 010: 内部触发 2 (IT2) 011: 内部触发 3 (IT3) 100: I1 边沿检测(I1F_ED) 101: I1 滤波后信号 110: I2 滤波后信号 111: 外部触发输入</p>
CHCSEL	Bit 3	R/W	<p>CHnREF 清除选择 0: CHnREF_CLR 连接至比较器输入 CMP_IN 1: CHnREF_CLR 连接至外部触发信号 ETF</p>
SMODS	Bits 2-0	R/W	<p>从模式选择 000: 从模式关闭 - 设置 CON1 寄存器 CNTEN 位，计数器由内部时钟计数 001: 编码器模式 1 - 计数器向上/向下计数 I2 边沿，取决于 I1 电平 010: 编码器模式 2 - 计数器向上/向下计数 I1 边沿检测边沿，取决于 I2 边沿检测电平 011: 编码器模式 3 - 计数器向上/向下计数 I1 边沿检测和 I2 边沿检测边沿，取决于另一个输入的电平 100: 复位模式 - 选中的触发输入(TI)的上升沿重新初始化计数器，并且产生一次更新事件 101: 门控模式 - 当触发输入(TI)为高电平时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控 110: 触发模式 - 计数器在触发输入 TI 的上升沿启动(但不复位)，只有计数器的启动是受控 111: 外部时钟模式 1 - 选中的触发输入(TI) 的上</p>

			<p>升沿驱动计数器</p> <p>注：如果 I1 双边沿检测被选为触发输入 (TSSEL='100')，不能使用门控模式。I1 每一次转换，I1 双边沿检测就会输出 1 个脉冲，而门控模式则是检查触发信号的电平</p>
--	--	--	--

从定时器	IT0(TSSEL =000)	IT1(TSSEL =001)	IT2(TSSEL =010)	IT3(TSSEL=011)
GP32C4T1	AD16C4T1	GP16C4T1	GP16C4T2	GP16C4T3

表 17-2 GP32C4Tn 内部触发连接

17.5.2.4 中断使能寄存器 (GP32C4Tn_IER)

中断使能寄存器 (GP32C4Tn_IER)																																
偏移地址: 0C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																				CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	W1	捕获溢出 4 中断使能 0: 写入0无效 1: 捕获溢出CH4中断使能
CH3OVI	Bit 11	W1	捕获溢出 3 中断使能 0: 写入 0 无效 1: 捕获溢出 CH3 中断使能
CH2OVI	Bit 10	W1	捕获溢出 2 中断使能 0: 写入 0 无效 1: 捕获溢出 CH2 中断使能
CH1OVI	Bit 9	W1	捕获溢出 1 中断使能 0: 写入 0 无效 1: 捕获溢出 CH1 中断使能
—	Bit 8-7	—	—
TRGI	Bit 6	W1	触发中断使能 0: 写入 0 无效 1: 触发中断使能
—	Bit 5	—	—
CH4I	Bit 4	W1	捕获/比较 4 捕获中断使能 0: 写入 0 无效 1: 捕获 CH4 中断使能
CH3I	Bit 3	W1	捕获/比较 3 捕获中断使能 0: 写入 0 无效 1: 捕获 CH3 中断使能
CH2I	Bit 2	W1	捕获/比较 2 捕获中断使能 0: 写入 0 无效 1: 捕获 CH2 中断使能
CH1I	Bit 1	W1	捕获/比较 1 捕获中断使能 0: 写入 0 无效 1: 捕获 CH1 中断使能
UI	Bit 0	W1	更新中断使能 0: 写入 0 无效

			1: 更新中断使能
--	--	--	-----------

17.5.2.5 中断禁止寄存器 (GP32C4Tn_IDR)

中断禁止寄存器 (GP32C4Tn_IDR)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	W1	捕获溢出 4 中断禁止 0: 写入0无效 1: 捕获溢出CH4中断禁止
CH3OVI	Bit 11	W1	捕获溢出 3 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH3 中断禁止
CH2OVI	Bit 10	W1	捕获溢出 2 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH2 中断禁止
CH1OVI	Bit 9	W1	捕获溢出 1 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH1 中断禁止
—	Bit 8-7	—	—
TRGI	Bit 6	W1	触发中断禁止 0: 写入 0 无效 1: 触发中断禁止
—	Bit 5	—	—
CH4I	Bit 4	W1	捕获/比较 4 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH4 中断禁止
CH3I	Bit 3	W1	捕获/比较 3 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH3 中断禁止
CH2I	Bit 2	W1	捕获/比较 2 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH2 中断禁止
CH1I	Bit 1	W1	捕获/比较 1 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH1 中断禁止
UI	Bit 0	W1	更新中断禁止 0: 写入 0 无效

			1: 更新中断禁止
--	--	--	-----------

17.5.2.6 中断有效状态寄存器 (GP32C4Tn_IVS)

中断有效状态寄存器 (GP32C4Tn_IVS)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	捕获溢出 4 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH4 中断禁止 1: 捕获溢出 CH4 中断使能
CH3OVI	Bit 11	R	捕获溢出 3 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH3 中断禁止 1: 捕获溢出 CH3 中断使能
CH2OVI	Bit 10	R	捕获溢出 2 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH2 中断禁止 1: 捕获溢出 CH2 中断使能
CH1OVI	Bit 9	R	捕获溢出 1 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH1 中断禁止 1: 捕获溢出 CH1 中断使能
—	Bit 8-7	—	—
TRGI	Bit 6	R	触发中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 触发中断禁止 1: 触发中断使能
—	Bit 5	—	—
CH4I	Bit 4	R	捕获/比较 4 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH4 中断禁止 1: 捕获 CH4 中断使能
CH3I	Bit 3	R	捕获/比较 3 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH3 中断禁止 1: 捕获 CH3 中断使能
CH2I	Bit 2	R	捕获/比较 2 捕获中断有效位

			设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH2 中断禁止 1: 捕获 CH2 中断使能
CH1I	Bit 1	R	捕获/比较 1 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH1 中断禁止 1: 捕获 CH1 中断使能
UI	Bit 0	R	更新中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 更新中断禁止 1: 更新中断使能

17.5.2.7 原始中断标志寄存器 (GP32C4Tn_RIF)

原始中断标志寄存器 (GP32C4Tn_RIF)																															
偏移地址: 18H																															
复位值: 00000000_00000000_00000000_00000000B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	<p>捕获溢出4原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出CH4原始中断产生</p>
CH3OVI	Bit 11	R	<p>捕获溢出3原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH3 原始中断产生</p>
CH2OVI	Bit 10	R	<p>捕获溢出2原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH2 原始中断产生</p>
CH1OVI	Bit 9	R	<p>捕获溢出1原始中断标志 当CH1I已经被设置又再次收到捕获信号将计数器数值写入CCVAL1寄存器中。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 捕获溢出 CH1 原始中断产生</p>
—	Bit 8-7	—	—
TRGI	Bit 6	R	<p>触发原始中断标志 产生触发事件时产生(在从模式-门控时, 在任一沿产生, 其他模式则在有效沿时产生)。此位由硬件设置1, 设置 ICR 寄存器清除 0: 无中断产生 1: 触发原始中断产生</p>
—	Bit 5	—	—
CH4I	Bit 4	R	<p>捕获/比较 4 原始中断标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 4 原始中断产生</p>
CH3I	Bit 3	R	<p>捕获/比较 3 原始中断标志</p>

			<p>参照 CH1I 描述</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 3 原始中断产生</p>
CH2I	Bit 2	R	<p>捕获/比较 2 原始中断标志</p> <p>参照 CH1I 描述</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 2 原始中断产生</p>
CH1I	Bit 1	R	<p>捕获/比较 1 原始中断标志</p> <p>通道 CH1 设置为输出:</p> <p>计数器匹配 CCVAL1 寄存器时设置, 在中央对齐模式依据 CMSEL 位在递增、递减、递增/递减时产生。当 CCVAL1 寄存器大于 AR 寄存器时, 则在递增边界或递减边界时置起。此位由硬件设置 1, 设置 ICR 寄存器清除</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 1 原始中断产生</p> <p>通道 CH1 设置为输入:</p> <p>发生捕获事件, 将计数器捕获至 CCVAL1 寄存器中, 于 I1 的有效沿产生。此位由硬件设置 1, 设置 ICR 寄存器清除</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 1 原始中断产生</p>
UI	Bit 0	R	<p>更新原始中断标志</p> <p>当 CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断</p> <ul style="list-style-type: none"> - 当重复计数器数值上溢或下溢时 (重复计数器为 0 时时产生更新事件)。 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器重新初始化时。 - 当计数器被触发事件重新初始化时 <p>0: 无中断产生</p> <p>1: 更新原始中断产生</p>

17.5.2.8 中断屏蔽标志寄存器 (GP32C4Tn_IFM)

中断屏蔽标志寄存器 (GP32C4Tn_IFM)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	<p>捕获溢出4中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出CH4中断屏蔽标志产生</p>
CH3OVI	Bit 11	R	<p>捕获溢出3中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH3 中断屏蔽标志产生</p>
CH2OVI	Bit 10	R	<p>捕获溢出2中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH2 中断屏蔽标志产生</p>
CH1OVI	Bit 9	R	<p>捕获溢出1中断屏蔽标志 当IVS寄存器为1时, CH1I已经被设置又再次收到捕获信号将计数器数值写入CCVAL1寄存器中。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 捕获溢出 CH1 中断屏蔽标志产生</p>
—	Bit 8-7	—	—
TRGI	Bit 6	R	<p>触发中断屏蔽标志 当 IVS 寄存器为 1 时, 产生触发事件时产生(在从模式-门控时, 在任一沿产生, 其他模式则在有效沿时产生)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 触发中断产生</p>
—	Bit 5	—	—
CH4I	Bit 4	R	<p>捕获/比较 4 中断屏蔽标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 4 中断屏蔽标志产生</p>

CH3I	Bit 3	R	<p>捕获/比较 3 中断屏蔽标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 3 中断屏蔽标志产生</p>
CH2I	Bit 2	R	<p>捕获/比较 2 中断屏蔽标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 2 中断屏蔽标志产生</p>
CH1I	Bit 1	R	<p>捕获/比较 1 中断屏蔽标志 当 IVS 寄存器为 1 时 通道 CH1 设置为输出: 计数器匹配 CCVAL1 寄存器时设置, 在中央对齐模式依据 CMSEL 位在递增、递减、递增/递减时产生。当 CCVAL1 寄存器大于 AR 寄存器时, 则在递增边界或递减边界时置起。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 中断屏蔽标志产生 通道 CH1 设置为输入: 发生捕获事件, 将计数器捕获至 CCVAL1 寄存器中, 于 I1 的有效沿产生。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 中断屏蔽标志产生</p>
UI	Bit 0	R	<p>更新中断屏蔽标志 当 CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断 - 当重复计数器数值上溢或下溢时 (重复计数器为 0 时时产生更新事件)。 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器重新初始化时。 - 当计数器被触发事件重新初始化时 0: 无中断产生 1: 更新中断屏蔽标志产生</p>

17.5.2.9 中断标志清除寄存器 (GP32C4Tn_ICR)

中断标志清除寄存器 (GP32C4Tn_ICR)																															
偏移地址: 20H																															
复位值: 00000000_00000000_00000000_00000000B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI			TRGI		CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	C_W1	捕获溢出 4 中断标志清除 0: 写入0无效 1: 捕获溢出CH4中断标志清除
CH3OVI	Bit 11	C_W1	捕获溢出 3 中断标志清除 0: 写入0无效 1: 捕获溢出 CH3 中断标志清除
CH2OVI	Bit 10	C_W1	捕获溢出 2 中断标志清除 0: 写入0无效 1: 捕获溢出 CH2 中断标志清除
CH1OVI	Bit 9	C_W1	捕获溢出 1 中断标志清除 0: 写入0无效 1: 捕获溢出 CH1 中断标志清除
—	Bit 8-7	—	—
TRGI	Bit 6	C_W1	触发中断标志清除 0: 写入 0 无效 1: 触发中断标志清除
—	Bit 5	—	—
CH4I	Bit 4	C_W1	捕获/比较 4 中断标志清除 0: 写入 0 无效 1: 捕获 CH4 中断标志清除
CH3I	Bit 3	C_W1	捕获/比较 3 中断标志清除 0: 写入 0 无效 1: 捕获 CH3 中断标志清除
CH2I	Bit 2	C_W1	捕获/比较 2 中断标志清除 0: 写入 0 无效 1: 捕获 CH2 中断标志清除
CH1I	Bit 1	C_W1	捕获/比较 1 中断标志清除 0: 写入 0 无效 1: 捕获 CH1 中断标志清除
UI	Bit 0	C_W1	更新中断标志清除 0: 写入 0 无效

			1: 更新中断标志清除
--	--	--	-------------

17.5.2.10 软件生成事件寄存器 (GP32C4Tn_SGE)

软件生成事件寄存器 (GP32C4Tn_SGE)																																
偏移地址: 24H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																										SGTRG		SGCH4	SGCH3	SGCH2	SGCH1	SGU

—	Bits 31-7	—	—
SGTRG	Bit 6	W1	<p>软件生成触发事件 该位由软件设置来生成触发事件，可由硬件自动清零。 0: 无动作 1: RIF 寄存器中的 TRGI 被置起，产生相关中断或 DMA 传输</p>
—	Bit 5	—	—
SGCH4	Bit 4	W1	<p>软件生成通道 4 捕获/比较事件 参考 SGCH1 描述</p>
SGCH3	Bit 3	W1	<p>软件生成通道 3 捕获/比较事件 参考 SGCH1 描述</p>
SGCH2	Bit 2	W1	<p>软件生成通道 2 捕获/比较事件 参考 SGCH1 描述</p>
SGCH1	Bit 1	W1	<p>软件生成通道 1 捕获/比较事件 通道 CH1 设置为输出: 产生捕获/比较但不影响输出，若使能中断或 DMA，则产生中断与请求。由软件设置，于下一个时钟自动清除 0: 写入 0 无效 1: 捕获/比较 1 中断产生 通道 CH1 设置为输入: 发生捕获事件，将计数器捕获至 CCVAL1 寄存器中，于 I1 的有效沿产生，若使能中断或 DMA，则产生中断与请求。由软件设置，于下一个时钟自动清除 0: 写入 0 无效 1: 捕获/比较 1 中断产生</p>
SGU	Bit 0	W1	<p>软件触发更新事件 该位由软件设置，可由硬件自动清零。 0: 无动作 1: 重新初始化计数器，更新寄存器。注意，预分频器也会被清零（但预分频比不会受到影响）。如果使用中央对齐模式或者 DIRSEL=0（递增），则计数器将清零；否则如果 DIRSEL=1（递减），则</p>

			将使用自动重载入值。
--	--	--	------------

17.5.2.11 捕获/比较模式寄存器 1 (GP32C4Tn_CHMR1)

捕获/比较模式寄存器 1 (GP32C4Tn_CHMR1)																																	
偏移地址: 28H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																CH2OCLREN	CH2MOD <2:0>			CH2PEN	CH2FEN	CC2SSEL <1:0>			CH1OCLREN	CH1MOD <2:0>			CH1PEN	CH1FEN	CC1SSEL <1:0>		
																I2FLT <3:0>			I2PRES <1:0>			CC2SSEL <1:0>			I1FLT <3:0>			I1PRES <1:0>			CC1SSEL <1:0>		

输出比较模式

—	Bits 31-16	—	—
CH2OCLREN	Bit 15	R/W	输出比较通道 2 清除使能 参考 CH1OCLREN 描述
CH2MOD	Bits 14-12	R/W	输出比较信道 2 模式 参考 CH1MOD 描述
CH2PEN	Bit 11	R/W	输出比较通道 2 预装载使能 参考 CH1PEN 描述
CH2FEN	Bit 10	R/W	输出比较通道 2 快速使能 参考 CH1FEN 描述
CC2SSEL	Bits 9-8	R/W	捕获/比较通道 2 选择 设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC2EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入, 捕获源为 I2 10: 通道设置为输入, 捕获源为 I1 11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测
CH1OCLREN	Bit 7	R/W	输出比较通道 1 清除使能 0: CH1REF 维持输出 1: CH1REF 根据 CHREF_CLR 的有效位清除
CH1MOD	Bits 6-4	R/W	输出比较信道 1 模式 设置 CH1REF 输出模式, CH1 与 CH1N 由 CH1REF 产生, 而有效位由 CCEP 寄存器的 CC1POL 与 CC1NPPOL 位设置电平 000: 禁止 - 无作用 001: 匹配时设置高电平 - 当计数器匹配

			<p>CCVAL1 寄存器时, CH1REF 设置为 1</p> <p>010: 匹配时设置低电平 - 当计数器匹配 CCVAL1 寄存器时, CH1REF 设置为 0</p> <p>011: 匹配时设置翻转电平 - 当计数器匹配 CCVAL1 寄存器时, CH1REF 设置翻转电平(当前高/低电平翻转成低/高电平)</p> <p>100: 强制低电平 - CH1REF 强制设置低电平</p> <p>101: 强制高电平 - CH1REF 强制设置高电平</p> <p>110: PWM 模式 1 - 上数时, 当计数器小于 CCVAL1 寄存器时, 输出高电平, 其他则输出低电平。下数时, 当计数器大于 CCVAL1 寄存器时输出低电平, 其他则输出高电平</p> <p>111: PWM 模式 2 - 上数时, 当计数器小于 CCVAL1 寄存器时, 输出低电平, 其他则输出高电平。下数时, 当计数器大于 CCVAL1 寄存器时输出高电平, 其他则输出低电平</p>
CH1PEN	Bit 3	R/W	<p>输出比较通道 1 预装载使能</p> <p>设置后在更新事件时, 将设置的寄存器 CCVAL1 数值载入预装载 CCVAL1 寄存器中</p> <p>0: CCVAL1 寄存器预装载禁止</p> <p>1: CCVAL1 寄存器预装载使能</p>
CH1FEN	Bit 2	R/W	<p>输出比较通道 1 快速使能</p> <p>用于加速事件的产生</p> <p>0: CH1 的正常操作依赖于计数器与 CCVAL1 的值, 即使工作于触发器状态。当触发器的输入有一个有效沿时, 激活 CH1 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, CH1 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 CH1 输出间的延时被缩短为 3 个时钟周期。CH1FEN 只在信道被配置成 PWM1 或 PWM2 模式时起作用</p>
CC1SSEL	Bits 1-0	R/W	<p>捕获/比较通道 1 选择</p> <p>设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC1EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入, 捕获源为 I1</p> <p>10: 通道设置为输入, 捕获源为 I2</p> <p>11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测</p>

输入捕获模式

—	Bits 31-16	—	—
I2FLT	Bits 15-12	R/W	输入捕获通道2滤波器 参照I1FLT描述
I2PRES	Bits 11-10	R/W	输入捕获通道 2 预分频器 参照 IC1PRES 描述
CC2SSEL	Bits 9-8	R/W	捕获/比较通道 2 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC2EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I2 10: 通道设置为输入，捕获源为 I1 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测
I1FLT	Bits 7-4	R/W	输入捕获通道 1 滤波器 设置 I1 信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变 0000: 采样频率 fDTS，滤波器禁止 0001: 采样频率 fINT_CLK, N = 2 0010: 采样频率 fINT_CLK, N = 4 0011: 采样频率 fINT_CLK, N = 8 0100: 采样频率 fDTS / 2, N = 6 0101: 采样频率 fDTS / 2, N = 8 0110: 采样频率 fDTS / 4, N = 6 0111: 采样频率 fDTS / 4, N = 8 1000: 采样频率 fDTS / 8, N = 6 1001: 采样频率 fDTS / 8, N = 8 1010: 采样频率 fDTS / 16, N = 5 1011: 采样频率 fDTS / 16, N = 6 1100: 采样频率 fDTS / 16, N = 8 1101: 采样频率 fDTS / 32, N = 5 1110: 采样频率 fDTS / 32, N = 6 1111: 采样频率 fDTS / 32, N = 8
I1PRES	Bits 3-2	R/W	输入捕获通道 1 预分频器 设置 I1 的预分频计数器数值，当清除 CCEP 寄存器的 CC1EN 位，预分频计数器同时被清除 00: 预分频禁止，于每次事件时捕获 01: 每 2 次事件捕获 10: 每 4 次事件捕获 11: 每 8 次事件捕获
CC1SSEL	Bits 1-0	R/W	捕获/比较通道 1 选择 设置通道的输出方向与信号的选择，当 CCEP 寄

			<p>寄存器的 CC1EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入, 捕获源为 I1</p> <p>10: 通道设置为输入, 捕获源为 I2</p> <p>11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测</p>
--	--	--	---

17.5.2.12 捕获/比较模式寄存器 2 (GP32C4Tn_CHMR2)

捕获/比较模式寄存器 2 (GP32C4Tn_CHMR2)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CH4OCLREN	CH4MOD <2:0>		CH4PEN	CH4FEN	CC4SSEL <1:0>		CH3OCLREN	CH3MOD <2:0>		CH3PEN	CH3FEN	CC3SSEL <1:0>			
																I4FLT <3:0>		I4PRES <1:0>		CC4SSEL <1:0>		I3FLT <3:0>		I3PRES <1:0>		CC3SSEL <1:0>					

输出比较模式

—	Bits 31-16	—	—
CH4OCLREN	Bit 15	R/W	输出比较通道 4 清除使能 参考 CH1OCLREN 描述
CH4MOD	Bits 14-12	R/W	输出比较信道 4 模式 参考 CH1MOD 描述
CH4PEN	Bit 11	R/W	输出比较通道 4 预装载使能 参考 CH1PEN 描述
CH4FEN	Bit 10	R/W	输出比较通道 4 快速使能 参考 CH1FEN 描述
CC4SSEL	Bits 9-8	R/W	捕获/比较通道 4 选择 设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC4EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入, 捕获源为 I4 10: 通道设置为输入, 捕获源为 I3 11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测
CH3OCLREN	Bit 7	R/W	输出比较通道 3 清除使能 参考 CH1OCLREN 描述
CH3MOD	Bits 6-4	R/W	输出比较信道 3 模式 参考 CH1OMOD 描述
CH3PEN	Bit 3	R/W	输出比较通道 3 预装载使能 参考 CH1PEN 描述
CH3FEN	Bit 2	R/W	输出比较通道 3 快速使能 参考 CH1FEN 描述
CC3SSEL	Bits 1-0	R/W	捕获/比较通道 3 选择

		<p>设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC3EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入，捕获源为 I3</p> <p>10: 通道设置为输入，捕获源为 I4</p> <p>11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测</p>
--	--	--

输入捕获模式

—	Bits 31-16	—	—
I4FLT	Bits 15-12	R/W	输入捕获通道4滤波器 参照I1FLT描述
I4PRES	Bits 11-10	R/W	输入捕获通道 4 预分频器 参照 IC1PRES 描述
CC4SSEL	Bits 9-8	R/W	捕获/比较通道 4 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC4EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I4 10: 通道设置为输入，捕获源为 I3 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测
I3FLT	Bits 7-4	R/W	输入捕获通道 3 滤波器 参考 I1FLT 描述
I3PRES	Bits 3-2	R/W	输入捕获通道 3 预分频器 参考 IC1PRES 描述
CC3SSEL	Bits 1-0	R/W	捕获/比较通道 3 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC3EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I3 10: 通道设置为输入，捕获源为 I4 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测

17.5.2.13 捕获/比较使能极性寄存器 (GP32C4Tn_CCEP)

捕获/比较使能寄存器 (GP32C4Tn_CCEP)																																
偏移地址: 30 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																CC4NPOL		CC4POL	CC4EN	CC3NPOL			CC3POL	CC3EN	CC2NPOL		CC2POL	CC2EN	CC1NPOL		CC1POL	CC1EN

—	Bits 31-16	—	—
CC4NPOL	Bit 15	R/W	捕获/比较通道4互补输出有效位极性 参照CC1NPOL描述
—	Bit 14	R/W	—
CC4POL	Bit 13	R/W	捕获/比较通道4输出有效位极性 参照CC1POL描述
CC4EN	Bit 12	R/W	捕获/比较通道4输出使能 参照CC1EN描述
CC3NPOL	Bit 11	R/W	捕获/比较通道3互补输出有效位极性 参照CC1NPOL描述
—	Bit 10	R/W	—
CC3POL	Bit 9	R/W	捕获/比较通道3输出有效位极性 参照CC1POL描述
CC3EN	Bit 8	R/W	捕获/比较通道3输出使能 参照CC1EN描述
CC2NPOL	Bit 7	R/W	捕获/比较通道2互补输出有效位极性 参照CC1NPOL描述
—	Bit 14	R/W	—
CC2POL	Bit 5	R/W	捕获/比较通道2输出有效位极性 参照CC1POL描述
CC2EN	Bit 4	R/W	捕获/比较通道2输出使能 参照CC1EN描述
CC1NPOL	Bit 3	R/W	捕获/比较通道2互补输出有效位极性 通道CH1设置为输出: 0: CH1N 高电平有效 1: CH1N 低电平有效 通道CH1设置为输入: 该位需和CC1POL一起使用来定义输入边沿的极性。参考CC1POL描述。
—	Bit 2	R/W	保留
CC1POL	Bit 1	R/W	捕获/比较通道1输出有效位极性 通道CH1设置为输出:

			<p>0: CH1 高电平有效 1: CH1 低电平有效 通道 CC1 设置为输入: CC1NPOL/CC1POL 位选择触发边沿或捕获模式下 I1 和 I2 的极性 00: 非反相/上升沿 01: 反相/下降沿 10: 保留 11: 非反相/上升沿+下降沿</p>
CC1EN	Bit 0	R/W	<p>捕获/比较通道 1 输出使能 通道 CH1 设置为输出: 0: 关闭 - CH1 无效 1: 开启 - CH1 为对应输出引脚上的输出信号 通道 CH1 设置为输入: 0: 捕获禁止 1: 捕获使能</p>

17.5.2.14 计数寄存器 (GP32C4Tn_COUNT)

计数寄存器 (GP32C4Tn_COUNT)																															
偏移地址: 34 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CNTV<31:0>															

CNTV	Bits 31-0	R/W	计数器数值
------	-----------	-----	-------

17.5.2.15 预分频寄存器 (GP32C4Tn_PRES)

时钟预分频寄存器 (GP32C4Tn_PRES)																															
偏移地址: 38 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PSCV<15:0>															

—	Bits 31-16	—	—
PSCV	Bits 15-0	R/W	<p>预分频数值</p> <p>当计数器时钟频率等于fINT_CLK/(PSCV<15:0>+1)时计数器递增或递减。在更新事件产生时,将PSCV数值被载入预装载寄存器中</p>

17.5.2.16 自动重载寄存器 (GP32C4Tn_AR)

自动重载寄存器 (GP32C4Tn_AR)																															
偏移地址: 3C _H																															
复位值: 00000000_00000000_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARV<31:0>																															

ARV	Bits 31-0	R/W	自动装载数值 设置计数器的递增边界，设置数值为 0 时计数器停止计数
-----	-----------	-----	--

17.5.2.17 通道捕获/比较寄存器 1 (GP32C4Tn_CCVAL1)

通道捕获/比较寄存器 1 (GP32C4Tn_CCVAL1)																															
偏移地址: 44 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRV1<31:0>																															

CCRV1	Bits 31-0	R/W	捕获/比较数值 1 信道 CHn 配置为输出: CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。 如果在CHMRn寄存器中的预载功能没有选中，CCRVn中的值将被永久载入；否则，每当发生更新事件，预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与COUNT寄存器中的值进行比较，并在CHn上输出。 信道CHn配置为输入: CCRVn为由上一个输入捕获事件 (In) 传输的计数值。
-------	-----------	-----	--

17.5.2.18 通道捕获/比较寄存器 2 (GP32C4Tn_CCVAL2)

通道捕获/比较寄存器 2 (GP32C4Tn_CCVAL2)																															
偏移地址: 48 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRV2<31:0>																															

CCRV2	Bits 31-0	R/W	<p>捕获/比较数值2</p> <p>信道 CHn 配置为输出: CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。 如果在CHMRn寄存器中的预载功能没有选中, CCRVn中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与COUNT寄存器中的值进行比较, 并在CHn上输出。</p> <p>信道CHn配置为输入: CCRVn为由上一个输入捕获事件 (In) 传输的计数值。</p>
-------	-----------	-----	--

17.5.2.19 通道捕获/比较寄存器 3 (GP32C4Tn_CCVAL3)

通道捕获/比较寄存器 3 (GP32C4Tn_CCVAL3)																															
偏移地址: 4C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRV3<31:0>																															

CCRV3	Bits 31-0	R/W	<p>捕获/比较数值3</p> <p>信道 CHn 配置为输出: CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。 如果在CHMRn寄存器中的预载功能没有选中, CCRVn中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存</p>
-------	-----------	-----	---

			<p>器中。有效捕获/比较寄存器中包含的值将会与COUNT寄存器中的值进行比较，并在CHn上输出。</p> <p>信道CHn配置为输入： CCRVn为由上一个输入捕获事件（In）传输的计数值。</p>
--	--	--	---

17.5.2.20 通道捕获/比较寄存器 4 (GP32C4Tn_CCVAL4)

通道捕获/比较寄存器 4 (GP32C4Tn_CCVAL4)																															
偏移地址：50H																															
复位值：00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRV4<31:0>																															

CCRV4	Bits 15-0	R/W	<p>捕获/比较数值4</p> <p>信道 CHn 配置为输出： CCRVn 中的值将被载入实际的捕获/比较寄存器中（预载值）。</p> <p>如果在CHMRn寄存器中的预载功能没有选中，CCRVn中的值将被永久载入；否则，每当发生更新事件，预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与COUNT寄存器中的值进行比较，并在CHn上输出。</p> <p>信道CHn配置为输入： CCRVn为由上一个输入捕获事件（In）传输的计数值。</p>
-------	-----------	-----	---

17.5.2.21 DMA 事件使能寄存器 (GP32C4Tn_DMAEN)

DMA 事件使能寄存器 (GP32C4Tn_DMAEN)																																
偏移地址: 58H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																										TRGIDE		CH4DE	CH3DE	CH2DE	CH1DE	UDE

—	Bits 31-7	—	—
TRGIDE	Bit 6	R/W	触发DMA请求使能 0: 触发DMA请求禁止 1: 触发DMA请求使能
—	Bits 5	—	—
CH4DE	Bit 4	R/W	通道捕获/比较 4 DMA 请求使能 0: 捕获/比较 4 DMA 请求禁止 1: 捕获/比较 4 DMA 请求使能
CH3DE	Bit 3	R/W	通道捕获/比较3 DMA请求使能 0: 捕获/比较3 DMA请求禁止 1: 捕获/比较3 DMA请求使能
CH2DE	Bit 2	R/W	通道捕获/比较 2 DMA 请求使能 0: 捕获/比较 2 DMA 请求禁止 1: 捕获/比较 2 DMA 请求使能
CH1DE	Bit 1	R/W	通道捕获/比较 1 DMA 请求使能 0: 捕获/比较 1 DMA 请求禁止 1: 捕获/比较 1 DMA 请求使能
UDE	Bit 0	R/W	更新 DMA 请求使能 0: 更新 DMA 请求禁止 1: 更新 DMA 请求使能

第18章 高级控制定时器 (AD16C4T)

18.1 概述

高级控制定时器 (AD16C4Tn) 是一个功能强大、配置灵活的定时器模块，它包含一个 16-bit 定时器,具有定时、计数、脉冲输入信号测量 (输入捕获)、产生特定 PWM 波 (输出比较) 等功能。

18.2 特性

- ◆ 16 位递增, 递减, 递增/递减自动加载计数器
- ◆ 16 位可编程预分频器, 可在定时器运行中对计数器工作时钟进行 1 到 65536 间的任意分频
- ◆ 带有四个独立信道, 每个信道支持以下功能
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ PWM 输出 (边沿与中央对齐模式)
 - ◇ 单脉冲输出
- ◆ 通道 1~3 支持互补输出, 死区时间可配
- ◆ 同步电路用于外部信号控制定时器及内部互联多个定时器
- ◆ 在给定数目的计数周期之后更新重复计数寄存器
- ◆ 支持刹车功能, 刹车后定时器输出状态可控
- ◆ 支持中断/DMA:
 - ◇ 更新事件: 计数器上溢/下溢, 计数器初始化 (通过软件或内/外部触发)
 - ◇ 触发事件 (计数器起始、停止、初始化或内/外触发计数)
 - ◇ 通信事件
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ 刹车输入
- ◆ 支持增量 (正交) 编码及霍尔电路进行定位
- ◆ 触发输入可对外部时钟或周期性电流管理

18.3 结构图

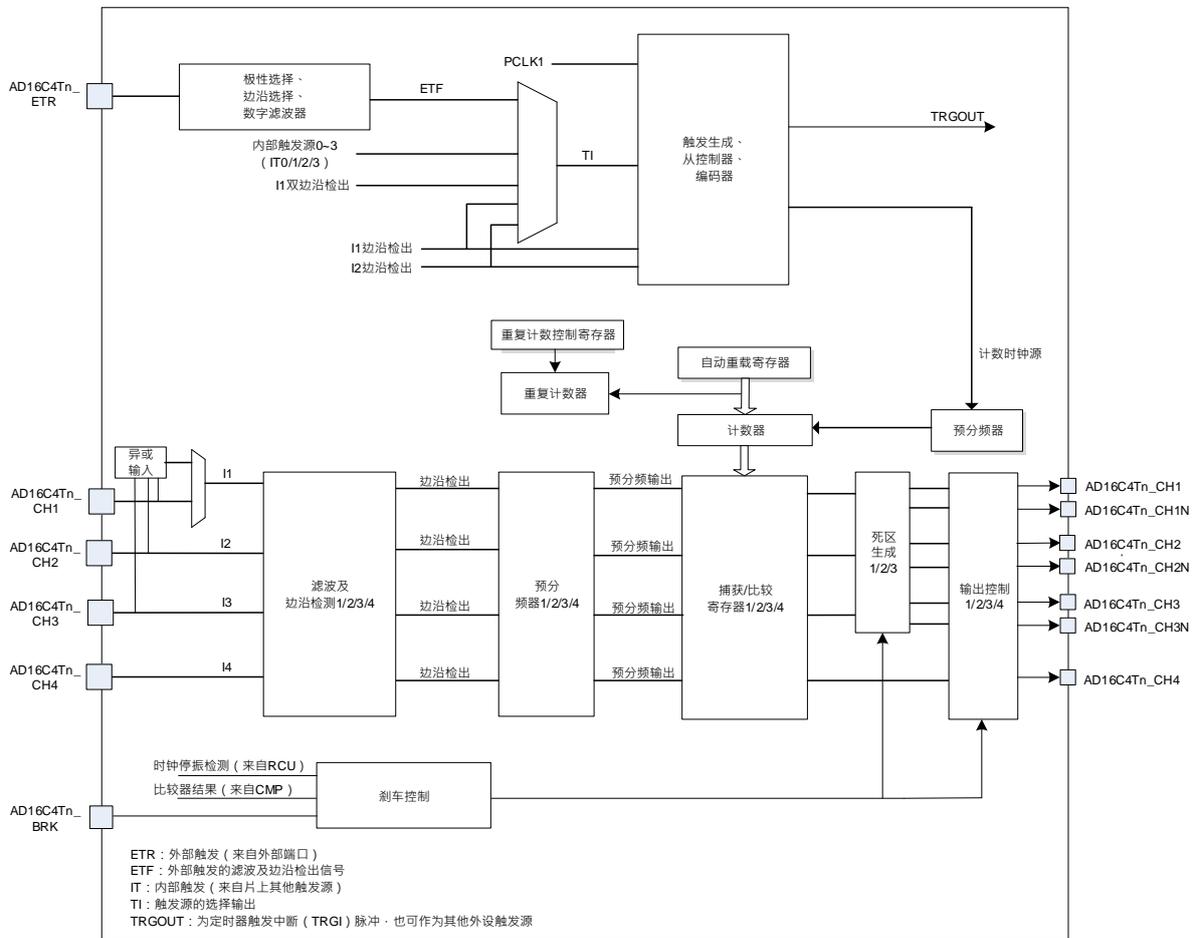


图 18-1 高级控制定时器结构框图

18.4 功能描述

18.4.1 预分频器

定时器包含一个 16-bit 的计数器（AD16C4Tn_COUNT），计数时钟由预分频寄存器（AD16C4Tn_PRES）进行分频。计数周期由自动重载计数器（AD16C4Tn_AR）设定。重复计数寄存器则可指定计数周期数目（AD16C4Tn_REPAR）。

自动重载寄存器（AD16C4Tn_AR）是一个可缓存的寄存器。当 AD16C4Tn_CON1 寄存器的 ARPEN 位复位时，AD16C4Tn_AR 寄存器重载功能失效，AD16C4Tn_AR 就是有效寄存器；ARPEN 置位时，AD16C4Tn_AR 寄存器具有重载功能，产生更新事件（UEV）时，加载值（AD16C4Tn_AR 寄存器值）更新到影子寄存器。

当 AD16C4Tn_CON1 寄存器中 DISUE 位为 0 时，计数器计数上溢（或递减下溢）时会产生更新事件（UEV）。同样，软件方式也可产生更新事件。AD16C4Tn_CON1 寄存器的 CNTEN 置位时，计数器开始计数。

注：计数器在 CNTEN 位置位 1 个时钟周期后开始计数。

预分频器可对定时器工作时钟进行 AD16C4Tn_PRES 寄存器值+1 次分频。由于 AD16C4Tn_PRES 是一个可重载寄存器，因此，定时器工作时可以对该寄存器进行修改，修改值在下次更新事件（UEV）后有效。

下图给出了定时器运行过程中改变预分频值时计数器的计数情况。

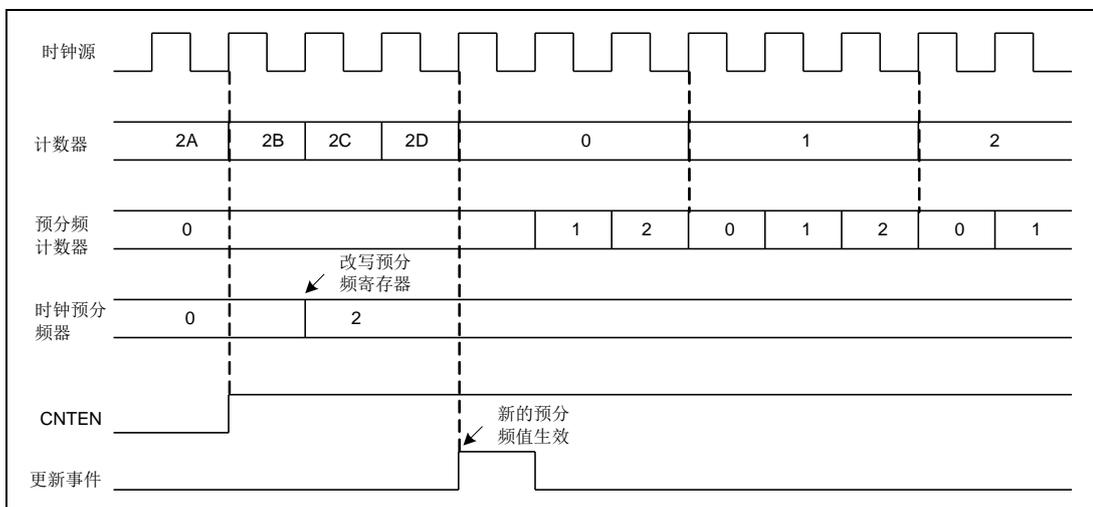


图 18-2 预分频值计数时序图

18.4.2 重复计数器

重复计数器用于控制发生多少次上溢或下溢出后产生更新事件。

重复计数器递减：

- ◇ 递增模式的每次上溢
- ◇ 递减模式的每次下溢
- ◇ 中央对齐模式时，计数器上溢与下溢。中央对齐模式限制了最大重复次数为 128 个 PWM 周期，每个 PWM 周期内可更新两次占空比。

AD16C4Tn_REPAR 寄存器是一个可缓存寄存器。软件（置位 **AD16C4Tn_SGE** 寄存器中的 **SGU** 位）或硬件从机模式控制方式产生更新事件时，无论重复计数器为何值，**AD16C4Tn_REPAR** 寄存器中值会立即更新到重复计数器的影子寄存器中。

中央对齐模式下，**REPAR** 中值为奇数时，更新事件是在上溢或下溢时产生，取决于何时写 **REPAR** 寄存器及何时开始计数。若在启动计数器前写 **REPAR**，则上溢时产生 **UEV**，反之则在下溢时产生 **UEV**。

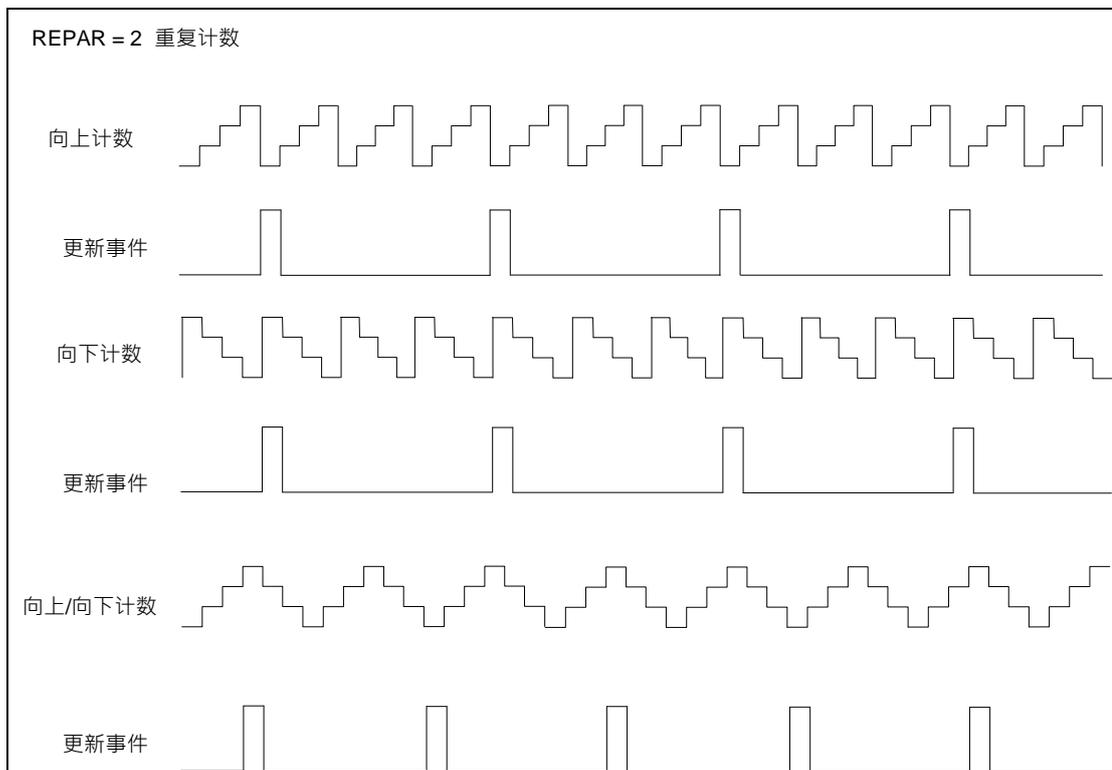


图 18-3 重复计数器工作模式

注意：置位 **AD16C4Tn_SGE** 寄存器中的 **SGU** 位也可以产生更新事件。

18.4.3 时钟源

计数器工作时钟可以选择内部时钟(INT_CLK)、外部时钟源 1 (I1、I2)、外部时钟源 2 (ETR)，内部触发输入 (IT0、IT1、IT2、IT3)

18.4.3.1 内部时钟源 (INT_CLK)

若从模式控制器被关闭 (AD16C4Tn_SMCON 寄存器内, SMODS= "000"), 则 CNTEN, AD16C4Tn_CON1.DIRSEL 与 AD16C4Tn_SGE.SGU 位为实际控制位, 这些位只能软件修改 (SGU 位除外, 仍硬件自动清除)。一旦 CNTEN 位被写为'1', 预分频器就由内部 INT_CLK 提供时钟。

下图给出了通常模式下控制电路和递增计数的情况, 没有分频。

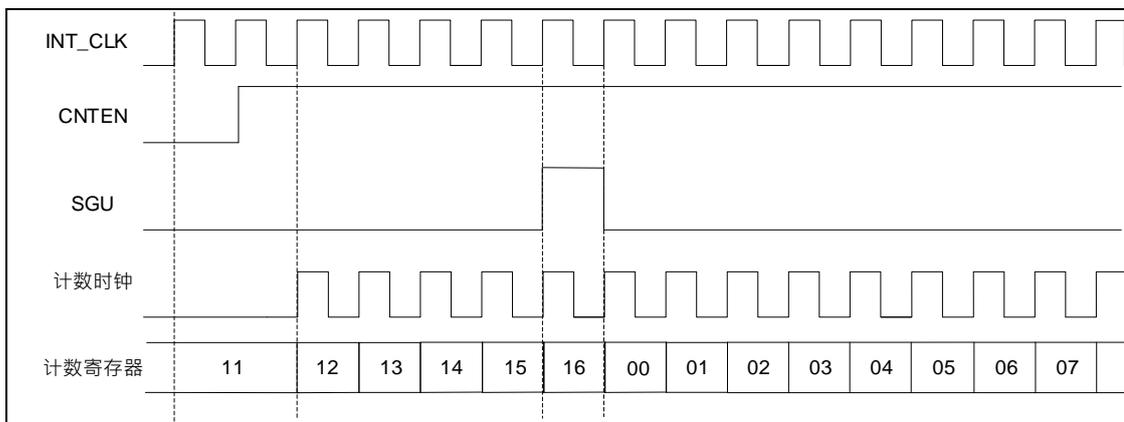


图 18-4 采用内部时钟计数

18.4.3.2 外部时钟源 1

AD16C4Tn_SMCON 寄存器的 **SMODS** = "111" 时, 可选择外部时钟源 1。计数器可根据选定的上升沿或下降沿计数。

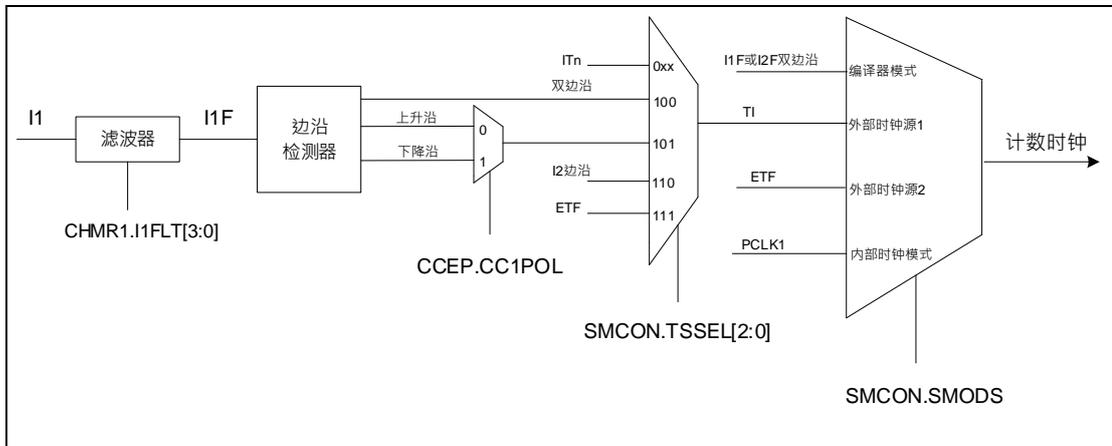


图 18-5 外部时钟连接

配置计数器为外部时钟源 1, 步骤如下:

1. **AD16C4Tn_SMCON** 寄存器中 **SMODS** = "111", 配置定时器外部时钟模式 1。
2. 设置 **AD16C4Tn_SMCON** 寄存器中的 **TSSEL** 选择外部时钟源。
3. 如外部时钟源为 I1, 可配置 **AD16C4Tn_CHMR1** 寄存器 **CC1SSEL** = "01", 配置信道 1 检测 I1 输入的上升沿; 设置 **AD16C4Tn_CCEP** 寄存器中 **CC1POL** = '0', 选择极性为上升沿。
4. 写 **AD16C4Tn_CHMR1** 寄存器的 **I1FLT[3: 0]** 位, 配置输入滤波器时间 (若没有滤波器需求, 维持 **I1FLT** = "0000")。
5. **AD16C4Tn_CON1** 寄存器中 **CNTEN** = '1', 使能计数器。

当 I1 上出现一次上升沿时, 计数器计数一次且 **TRGI** 标志位置位。

18.4.3.3 外部时钟源 2

置位 **AD16C4Tn_SMCON** 寄存器的 ECM2EN 位选定外部时钟源 2。

计数器可对外部触发输入 ETR 进行上升沿或下降沿计数。

下图给出了外部输入输入模块的概况。

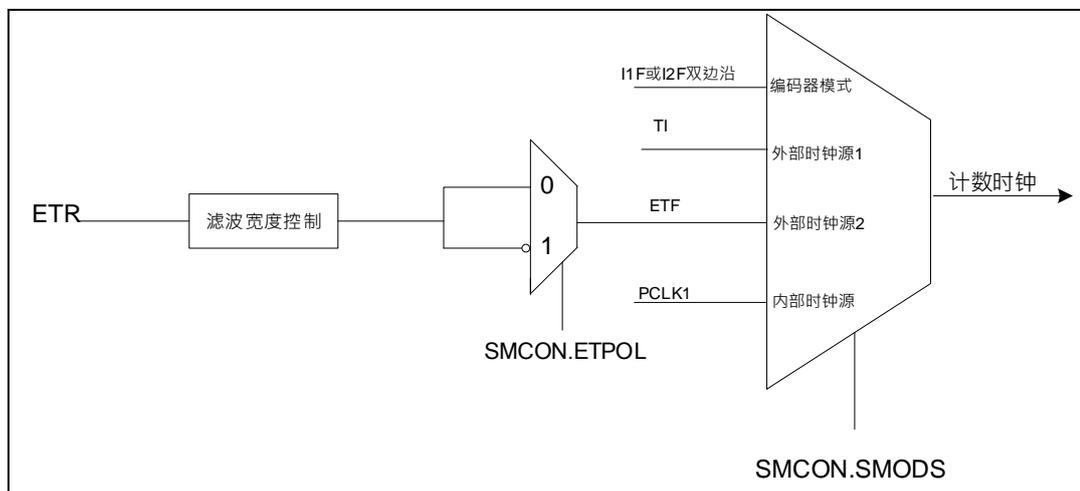


图 18-6 外部触发输入模块

配置计数器为外部时钟源 2，配置过程如下：

1. 设置 **AD16C4Tn_SMCON** 寄存器的 ETFLT[3: 0]，配置输入滤波时间。
2. 设置 **AD16C4Tn_SMCON** 寄存器中 ETPOL，检测 ETR 引脚上升沿或下降沿。
3. 设置 **AD16C4Tn_SMCON** 寄存器中 ECM2EN = '1'，使能外部时钟模式 2。
4. 设置 **AD16C4Tn_CON1** 寄存器的 CNTEN = '1'，使能计数器。

计数器每一个上升沿计一次数。

18.4.3.4 内部触发输入 (ITn)

当 **AD16C4Tn_SMCON** 寄存器的 **SMODS** = "111", 选定内部触发模式。计数器根据选定的内部输入端的上升或下降沿计数。

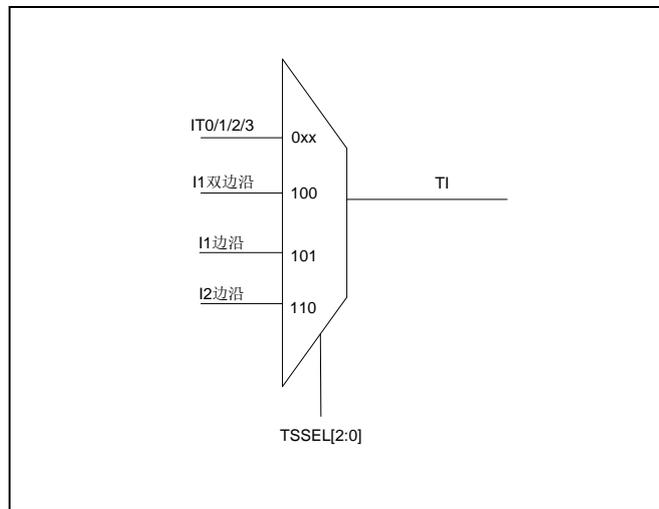


图 18-7 ITn 外部时钟连接

配置计数器在 ITn 输入端的上升沿递增计数，步骤如下：

1. **AD16C4Tn_SMCON** 寄存器中 **SMODS** = "111", 配置外部时钟模式 1。
2. **AD16C4Tn_SMCON** 寄存器的 **TSSEL** = "0xx", 选定 ITn 作为触发输入源。
3. **AD16C4Tn_CON1** 寄存器的 **CNTEN** = '1', 使能计数器。

ITn 产生上升沿时，计数器计数一次。ITn 上升沿与实际时钟间的延时，取决于 ITn 输入的再同步电路。

18.4.4 计数模式

18.4.4.1 递增计数模式

当 **AD16C4Tn_CON1** 寄存器的 **DIRSEL** 值为 0 时，定时器配置为递增模式，计数器从 0 开始递增，直至 **AD16C4Tn_AR** 寄存器值；然后从 0 重新开始计数并产生一个更新事件(UEV)。当 **AD16C4Tn_REPAR** 寄存器不为 0 时，则在 **AD16C4Tn_REPAR+1** 次计数后产生更新事件。

当有更新事件（UEV）产生时，预装载寄存器会更新到影子寄存器，更新标志位（**AD16C4Tn_RIF** 寄存器中的 **UI** 位）置位（取决于 **UERSEL** 位）：

- ◇ 更新 **AD16C4Tn_REPAR** 寄存器的值到影子寄存器
- ◇ 更新 **AD16C4Tn_AR** 寄存器的值到影子寄存器
- ◇ 更新 **AD16C4Tn_PRES** 寄存器的值到影子寄存器

下图为 **AD16C4Tn_AR = 0x16**，预分频设为 2 分频时的计数器时序。

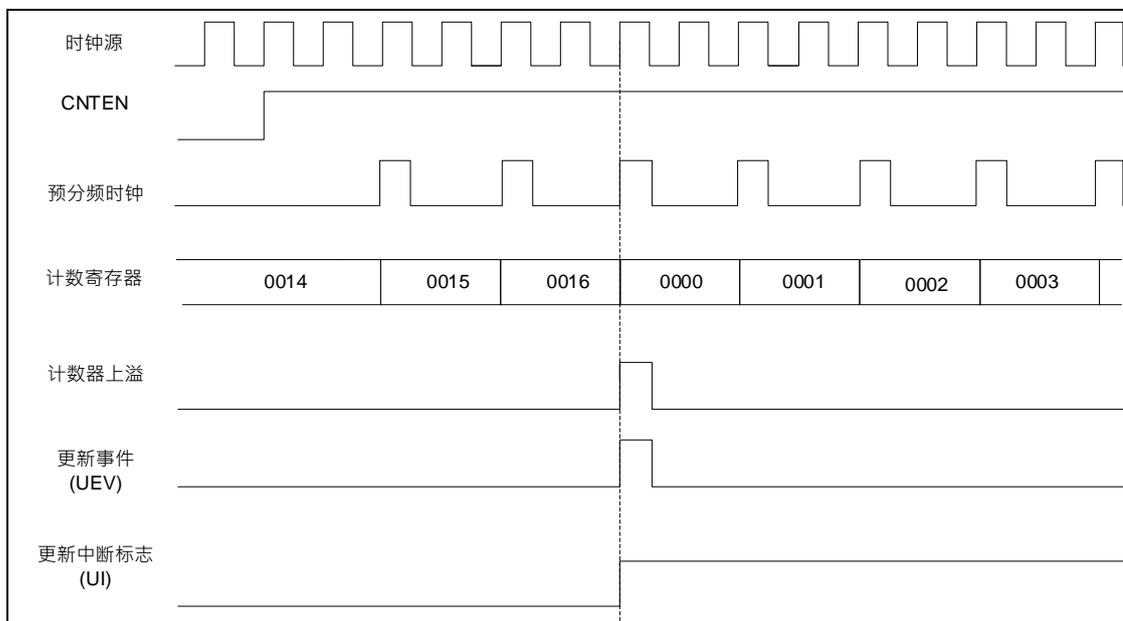


图 18-8 计数器递增计数时序图

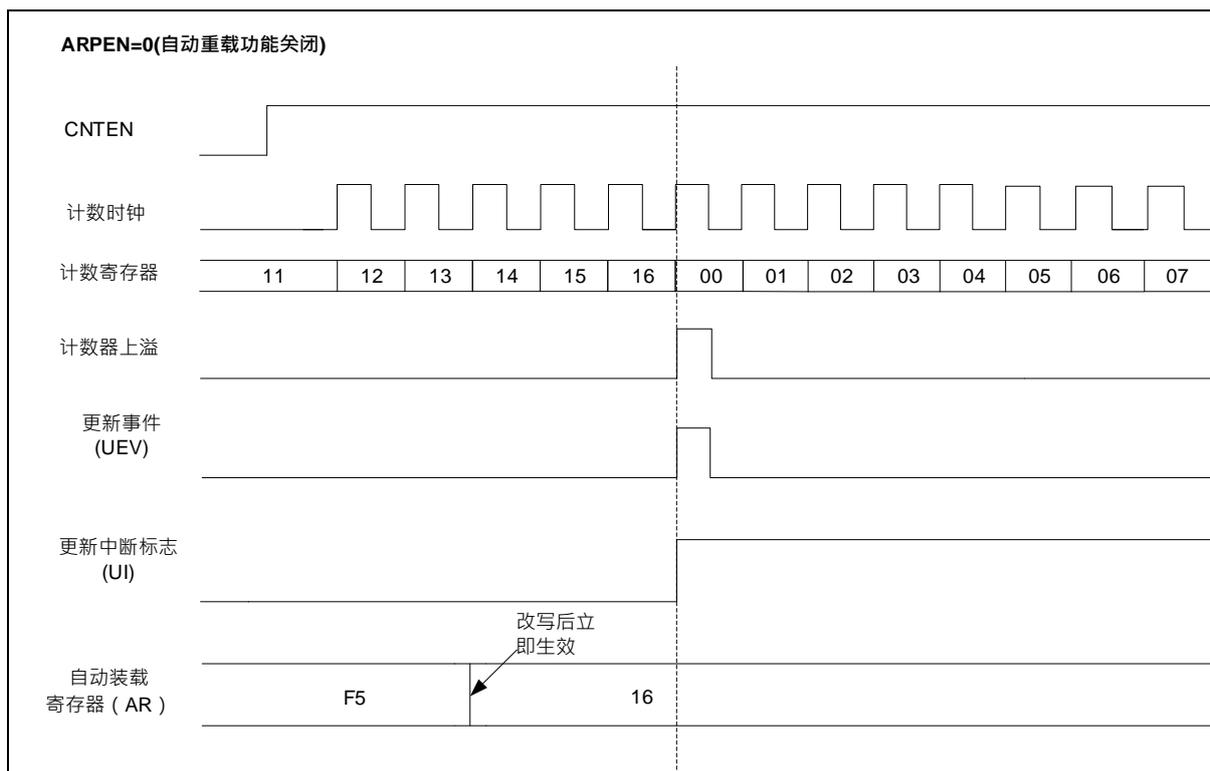


图 18-9 当 ARPEN=0 时计数器时序图

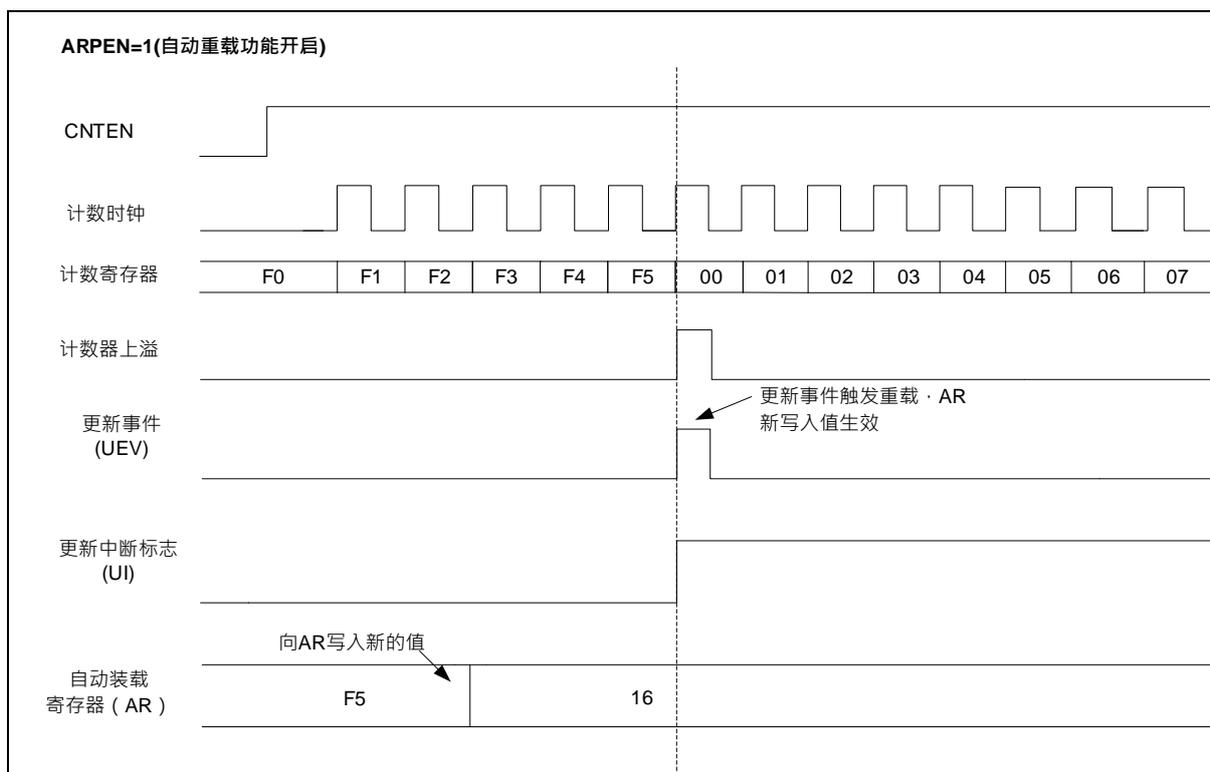


图 18-10 当 ARPEN=1 时计数器时序图

18.4.4.2 递减计数模式

当 **AD16C4Tn_CON1** 寄存器的 **DIRSEL** 值为 1 时，定时器配置为递减模式，计数器从 **AD16C4Tn_AR** 寄存器值开始递减至 0；然后重复递减并产生更新事件（UEV）。当 **AD16C4Tn_REPAR** 寄存器不为 0 时，则在 **AD16C4Tn_REPAR+1** 次后产生更新事件。

置位 **AD16C4Tn_SGE** 寄存器中的 **SGU** 位（通过软件或使用从机模式控制器）同样会产生更新事件。

当有更新事件（UEV）产生时，预载寄存器值会更新到影子寄存器，更新标志位（**AD16C4Tn_RIF** 寄存器中的 **UI** 位）置位（取决于 **UERSEL** 位）。

下图为 **AD16C4Tn_AR = 0x27**，预分频设为 1 分频时的计数器时序。

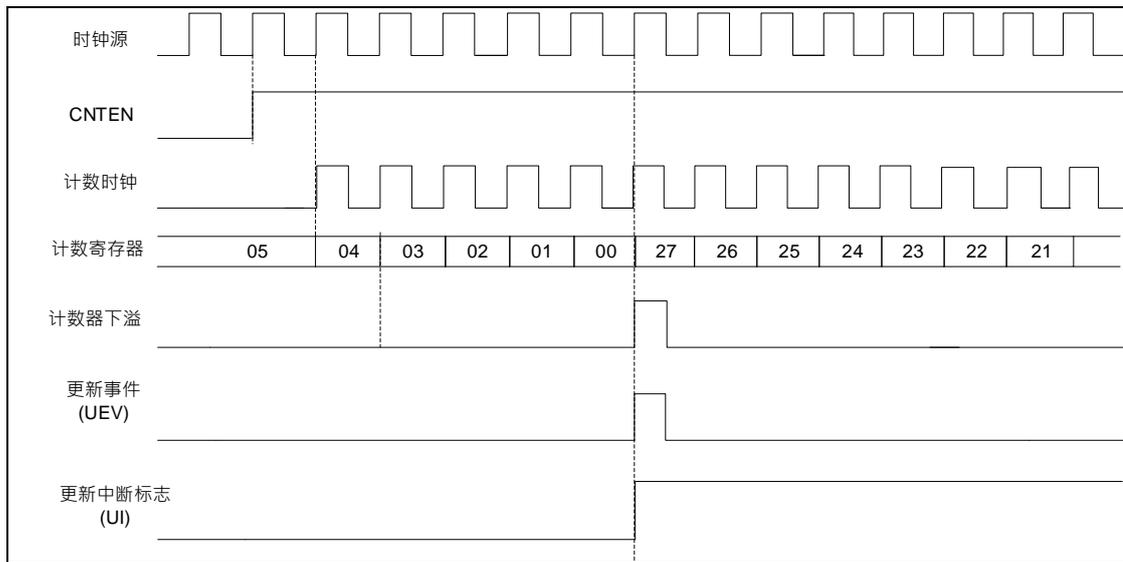


图 18-11 定时器递减计数时序图

18.4.4.3 中央对齐模式

当 **AD16C4Tn_CON1** 寄存器的 **CMSEL** 位的值不等于"00"时,定时器工作在中央对齐模式。定时器配置为中央对齐模式时,计数器先从 0 开始递增至 **AD16C4Tn_AR** 寄存器值减 1 后产生更新事件(UEV);接着计数器从 **AD16C4Tn_AR** 寄存器值递减至 1,并产生更新事件,如此循环计数。计数器递减计数(中央对称模式 1, **CMSEL**="01")、计数器递增计数(中央对称模式 2, **CMSEL**="10")、计数器递增和递减计数(中央对称模式 3, **CMSEL**="11"),每个通道的输出比较中断标志位都会置位。

在中央对齐模式下, **AD16C4Tn_CON1** 寄存器的 **DIRSEL** 位无法进行写操作,该位由硬件自动更新指示当前计数方向。

计数上溢、下溢或者置位 **AD16C4Tn_SGE** 寄存器的 **SGU** 位(通过软件或使用从模式控制器)都会产生更新事件。因此,计数器根据 **DIRSEL** 位由 0 或是 **AD16C4Tn_AR** 寄存器值开始递增或递减,而预分频器都从 0 开始计数。

软件置位 **AD16C4Tn_CON1** 寄存器中的 **DISUE** 位可关闭更新事件(UEV)的产生。更新事件(UEV)关闭时,可避免向预载寄存器写新值时更新影子寄存器。**DISUE** 复位之前都不会产生更新事件。而在正常产生更新事件时,计数器仍然从 0 开始,同样预分频计数也是从 0 开始(但预分频值没有改变)。此外,若置位 **AD16C4Tn_CON1** 寄存器中的 **UERSEL** 位(更新请求选择),置位 **SGU** 位时会产生一次更新事件(UEV),但 **UI** 标志位不会置位(因此,不会触发中断或 DMA 请求)。这就避免了在捕获事件时,清除计数器值时产生更新和捕获中断。

当有更新事件(UEV)产生时,预载寄存器值会更新到影子寄存器,更新标志位(**AD16C4Tn_RIF** 寄存器中的 **UI** 位)置位(取决于 **UERSEL** 位)。

注:若更新源为计数器上溢,自动重载会在计数器重载前更新。因此,下一周期即为预期值(计数器载入新值)。

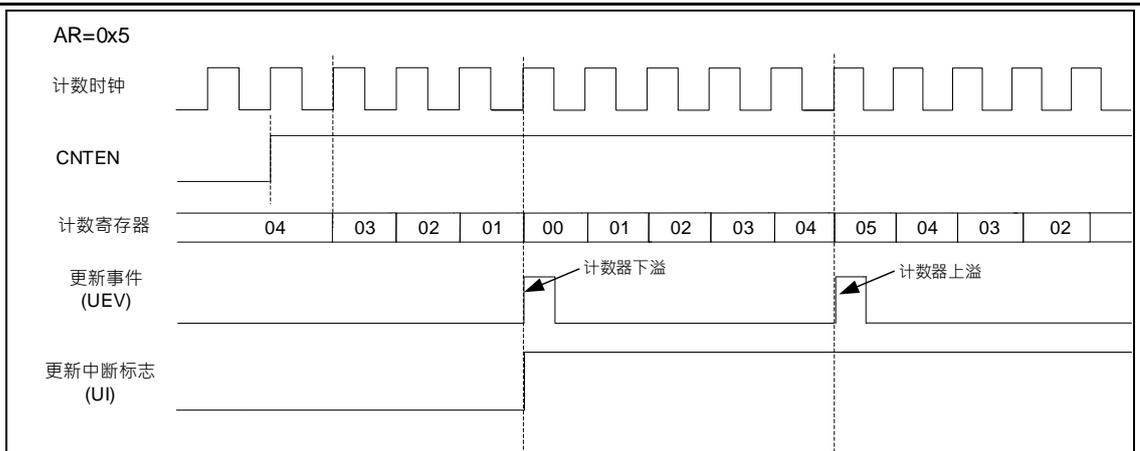


图 18-12 增减计数器时序图

18.4.5 捕获/比较通道

输入电路对 I_n 输入端的信号进行采样，产生一个经过滤波的信号 I_nF 。之后，一个可极性选择的边沿检测器产生 I_n 边沿检测信号，该信号可作为从模式控制器的触发输入或作为捕获控制命令，且信号经过分频后进入捕获寄存器。

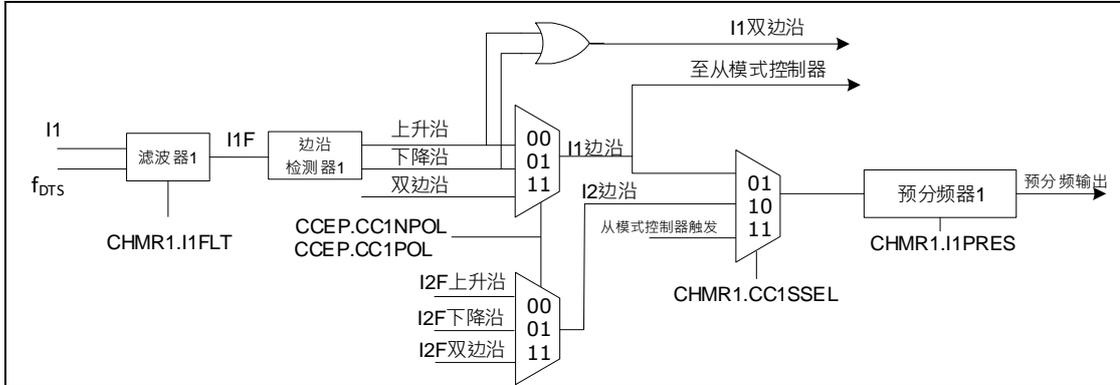


图 18-13 捕获/比较通道

输出部分产生一个中间波形（高有效）作为基准，在输出末端决定最终输出信号的极性。

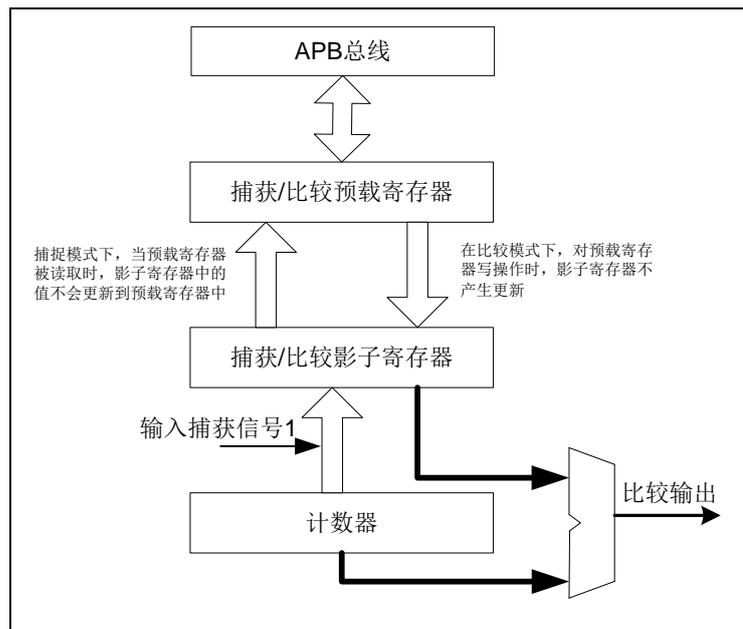


图 18-14 捕获/比较通道 1 结构图

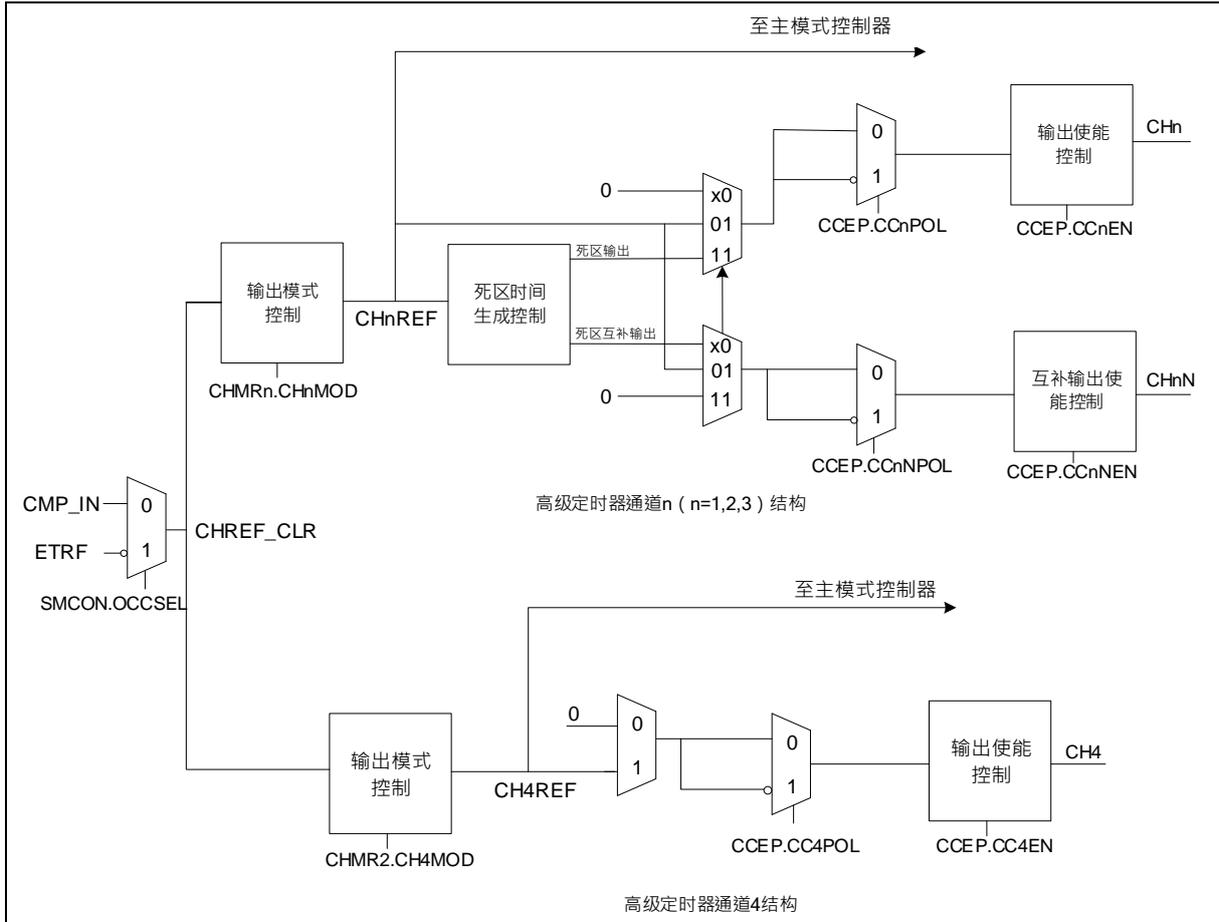


图 18-15 捕获/比较信道的输出部分

18.4.6 输入捕获模式

在输入捕获模式下，当检测到 In 上相应信号变化时，计数器的值就会被锁存到捕获/比较寄存器(AD16C4Tn_CCVALn)寄存器中。当捕获发生时，相应的 CHnI 标志位(AD16C4Tn_RIF)会置位，同时会触发中断或 DMA (如果使能) 请求。若发生捕获时，CHnI 标志位已经置位，则过捕获 CHnOVI 标志位 (AD16C4Tn_RIF) 置位。软件于 AD16C4Tn_ICR 对 CHnI 与 CHnOVI 位写'1'可以复位 CHnI 标志位与 CHnOVI 标志位。

以下为以 I1 输入上升沿作为捕获输入时的流程：

1. 选择有效输入端：**AD16C4Tn_CCVAL1** 必须连接到 I1 输入端，因此需将 **AD16C4Tn_CHMR1** 寄存器中的 CC1SSEL 位写"01"。只要 CC1SSEL 不为"00"，信道被配置为输入且 **AD16C4Tn_CCVAL1** 寄存器为只读。
2. 根据定时器连接的输入信号，配置输入滤波器的持续时间。当输入信号翻转时，前 5 个内部时钟信号内信号是不稳定的，因此必须配置滤波器的时间大于 5 个时钟周期。当 I1 检测到新的电平，连续 8 次采样可确认电平变化有效。
3. 选择 I1 信道的有效边沿变换。**AD16C4Tn_CCEP** 寄存器中的 CC1POL 写'0'(上升沿)。
4. 配置输入预分频器。
5. 置位 **AD16C4Tn_CCEP** 寄存器中的 CC1EN 位，使能捕获计数器的值到捕获寄存器。
6. 如有需要，置位 **AD16C4Tn_IER** 寄存器中的 CH1I 位，使能中断请求。置位 **AD16C4Tn_DMAEN** 寄存器中的 CH1DE 位，使能 DMA 请求。

当发生输入捕获时：

1. 有效边沿产生，**AD16C4Tn_CCVAL1** 寄存器获取计数器的值。
2. CH1I 标志位置位 (中断标志)。若至少 2 个连续的捕获发生，但标志位没有及时清除，则 CH1OVI 也会置位。
3. 中断的产生取决于 **AD16C4Tn_IER** 寄存器中的 CH1I 位。
4. DMA 请求的产生取决于 **AD16C4Tn_DMAEN** 寄存器中的 CH1DE 位。

为了处理捕获溢出，建议在读取过捕获标志位前先读取捕获数据。这可以避免错过读过捕获标志位之后，读之前产生的捕获数据。

注：捕获中断请求可由软件设置 AD16C4Tn_SGE 寄存器中 SGCHn 位产生。

18.4.6.1 PWM 输入模式

测量 I1 上 PWM 信号的周期和占空比的过程如下：

1. 为 **AD16C4Tn_CCVAL1** 选择有效的输入：**AD16C4Tn_CHMR1** 寄存器中的 **CC1SSEL** 位写"01"（I1 被选择）。
2. 为 I1 边沿检测选择有效的极性（用于捕获数据到 **AD16C4Tn_CCVAL1** 寄存器和计数器清零）：**CC1POL** 位写'0'（上升沿有效）。
3. 为 **AD16C4Tn_CCVAL2** 选择有效输入：**AD16C4Tn_CHMR1** 寄存器的 **CC2SEL** 位写"10"（I1 被选择）。
4. 为 I1 边沿检测选择有效极性（用于捕获数据到 **AD16C4Tn_CCVAL2**）：**CC2POL** 位写'1'。
5. 选择有效的触发输入：**AD16C4Tn_SMCON** 寄存器的 **TSSEL** 位写"101"（I1 边沿检测被选择）。
6. 配置从机模式控制器为复位模式：**AD16C4Tn_SMCON** 寄存器的 **SMODS** 位写"100"。
7. 使能捕获：**AD16C4Tn_CCEP** 寄存器的 **CC1EN** 位和 **CC2EN** 位写'1'。

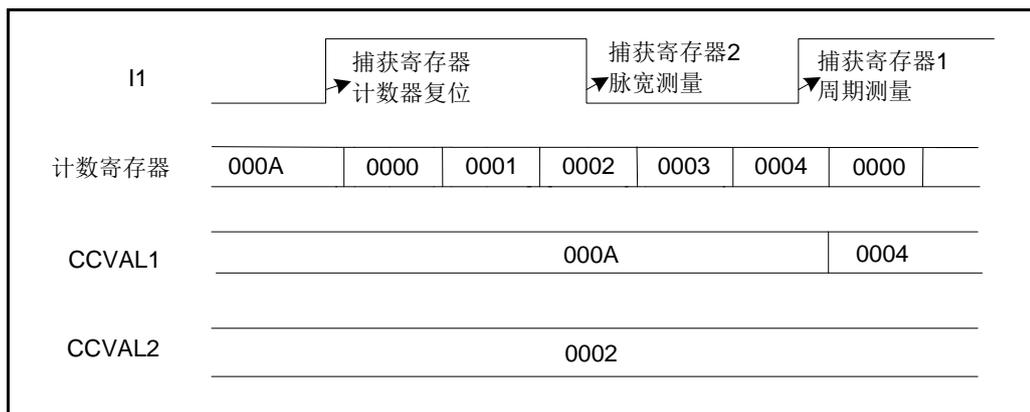


图 18-16 PWM 输入模式时序

18.4.7 PWM 模式

脉宽调制模式可以产生一个 **AD16C4Tn_AR** 寄存器值确定频率，**AD16C4Tn_CCVALn** 寄存器值确定占空比的信号。

每个信道的 PWM 模式是相互独立的（每个 CHn 输出一个 PWM），**AD16C4Tn_CHMRn** 寄存器的 CHnMOD 位写"110"（PWM 模式 1）或写"111"（PWM 模式 2）。必须通过置位 **AD16C4Tn_CHMRn** 寄存器的 CHnPEN 位来使能相应的预载寄存器，最后还需置位 **AD16C4Tn_CON1** 寄存器的 ARPEN 位来使能自动重装预载功能。

只有当更新事件发生时预载寄存器中的值才会传到影子寄存器，因此，在使能计数前，必须通过置位 **AD16C4Tn_SGE** 寄存器的 SGU 位来初始化所有的寄存器。

CHn 的极性可通过 **AD16C4Tn_CCEP** 寄存器的 CCnPOL 位配置，有效极性可配置为高或低。CHn 的输出使能由 CCnEN、CCnNEN、GOEN、OFFSSI 和 OFFSSR 位（**AD16C4Tn_CCEP** 和 **AD16C4Tn_BDCFG** 寄存器）组合控制。

在 PWM 模式（1 或 2）中，**AD16C4Tn_COUNT** 和 **AD16C4Tn_CCVALn** 寄存器的值会持续比较，确定 $AD16C4Tn_CCVALn \leq AD16C4Tn_COUNT$ 或 $AD16C4Tn_CCVALn > AD16C4Tn_COUNT$ （取决于计数器的计数方向）。

定时器产生 PWM 波形是边沿对齐或中央对齐，取决于 **AD16C4Tn_CON1** 寄存器的 CMSEL 位。

18.4.7.1 PWM 边沿对齐模式

- ◇ AD16C4Tn_AR=8
- ◇ PWM 模式 1
- ◇ 递增计数配置

当 **AD16C4Tn_CON1** 寄存器的 DIRSEL 位为低时，计数器递增计数。

下图给出了 AD16C4Tn_AR = 8 时的边沿对齐 PWM 波形。

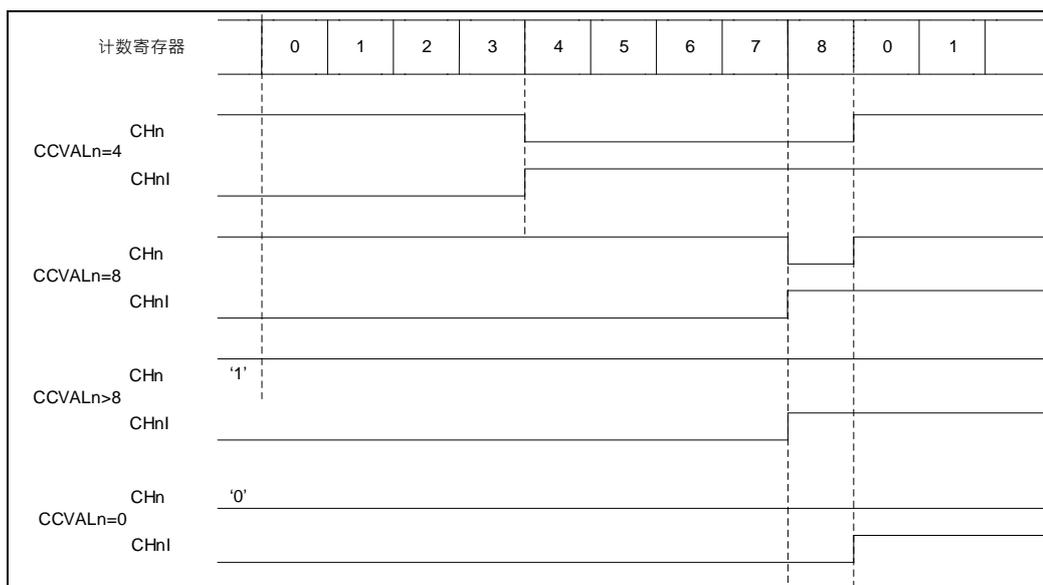


图 18-17 边沿对齐递增计数 PWM 波形 (AR=8)

- ◇ AD16C4Tn_AR=8
- ◇ PWM 模式 1
- ◇ 递减计数配置

当 AD16C4Tn_CON1 寄存器的 DIRSEL 位为高时，计数器递减计数。

下图给出了 AD16C4Tn_AR = 8 时的边沿对齐 PWM 波形。

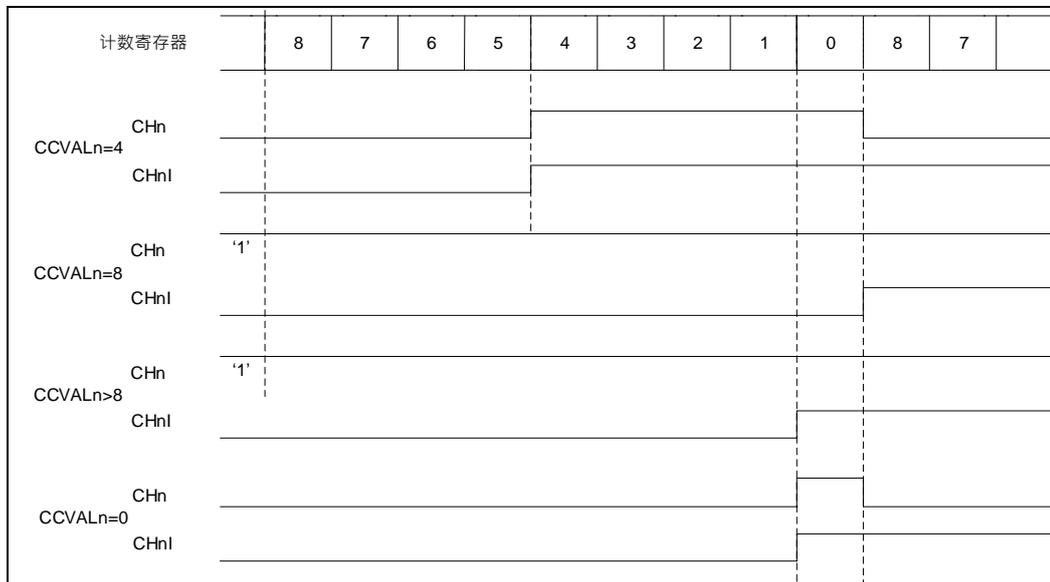


图 18-18 边沿对齐递减计数 PWM 波形 (AR=8)

18.4.7.2 PWM 中央对齐模式

当 **AD16C4Tn_CON1** 寄存器中的 **CMSEL** 位不为"00"时，中央对齐模式有效。计数器是递增、递减计数分别置比较标志位或递增递减都置比较标志位，取决于 **CMSEL** 位的配置。

AD16C4Tn_CON1 寄存器的方向位 (**DIRSEL**) 是由硬件更新的，软件无法修改。

下图为中央对齐方式产生的 PWM 波形的例子：

- ◇ **AD16C4Tn_AR=0x3F**
- ◇ PWM 模式 1
- ◇ **AD16C4Tn_CON1** 寄存器的 **CMSEL="10"**，在中央对齐模式 2 下，计数器向上计数时会置位比较标志位。

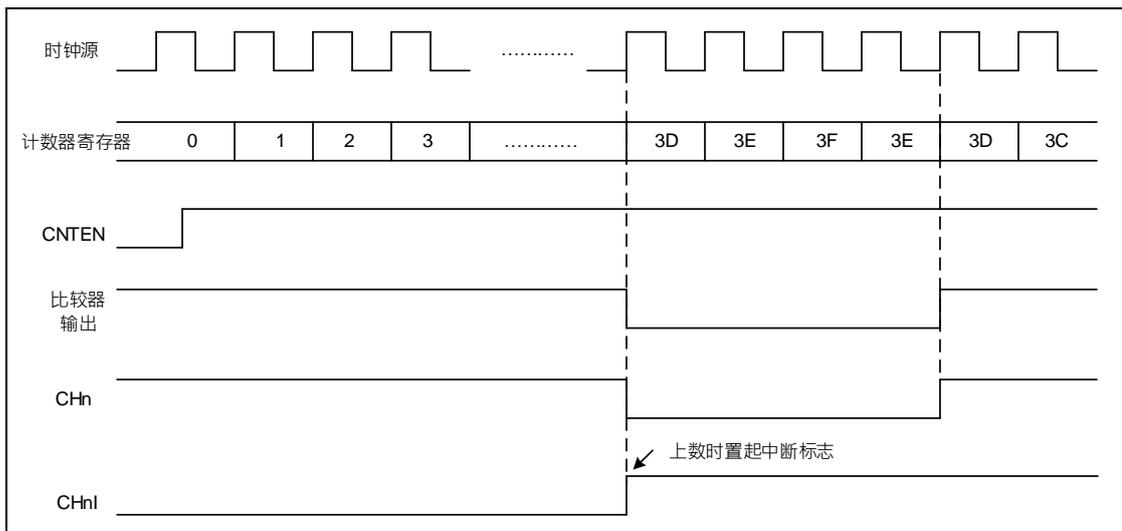


图 18-19 中央对齐 PWM 波形 (AR=0x3F)

中央对齐模式的使用技巧：

- ◇ 当进入中央对齐模式后，当前递增或递减配置生效。计数器递增或递减计数取决于 **AD16C4Tn_CON1** 寄存器的 **DIRSEL** 位的值。此外，软件无法对 **DIRSEL** 和 **CMSEL** 位同时进行修改。
- ◇ 计数器在中央对齐模式下运行时，对计数器写操作可能导致不可预知的结果。特别是：
 - 若向计数器入的值大于自动重载值 ($AD16C4Tn_COUNT > AD16C4Tn_AR$)，计数方向不更新。例如，如果计数器递增计数，写入值后从 0 重新递增计数。
 - 若向计数器写 0 或 **AD16C4Tn_AR** 中的重载值，则计数方向更新，但并没有产生 UEV。上数时
- ◇ 使用中央对齐模式最安全的方式是计数器开始计数前通过软件产生更新事件（置位 **AD16C4Tn_SGE** 寄存器中的 **SGU** 位）且在计数器运行过程中不对计数器写值。

18.4.8 输出比较模式

该功能用于控制输出波形或指示周期时间的结束。

当捕获/比较寄存器和计数器值匹配时，输出比较功能：

- ◇ 输出比较模式（**AD16C4Tn_CHMRn** 寄存器中的 **CHnMOD** 位）和输出极性（**AD16C4Tn_CCEP** 寄存器中的 **CCnPOL** 位）的配置值输出到对应的引脚上。
- ◇ 中断状态寄存器中的标志位置位（**AD16C4Tn_RIF** 寄存器的 **CHnI** 位）。
- ◇ 若相应的中断使能置位，则产生中断（**AD16C4Tn_IER** 寄存器的 **CHnI** 位）。
- ◇ 若相应的使能位置位（**AD16C4Tn_DMAEN** 寄存器的 **CHnDE** 位，**AD16C4Tn_CON2** 寄存器的 **CCDMASEL** 位用于 DMA 请求的选择），则发送 DMA 请求。

AD16C4Tn_CHMRn 寄存器中 **CHnPEN** 位的值可决定 **AD16C4Tn_CCVALn** 寄存器是否带有预装载寄存器。

在输出比较模式中，更新事件 **UEV** 对 **CHn** 的输出没有影响。计时分辨率为计数器的一次计数。输出比较模式同样可以用来输出单个脉冲（单脉冲模式）。

输出比较的配置过程：

1. 选定计数器时钟（内部，外部，预分频）。
2. **AD16C4Tn_AR** 与 **AD16C4Tn_CCVALn** 寄存器中写入预期值。
3. 若需要产生中断请求，置位 **AD16C4Tn_IER** 寄存器中的 **CHnI** 位。
4. 选择输出模式，例如：
 - **CHnMOD** = "011"，当 **CNTV** 与 **CCRVALn** 匹配时，**CHn** 输出翻转。
 - **CHnPEN** = '0'，关闭预载寄存器。
 - **CCnPOL** = '0'，选择有效极性为高。
 - **CCnEN** = '1'，使能输出。
5. **AD16C4Tn_CON1** 寄存器中的 **CNTEN** 位置位，使能计数器。

假设预载寄存器没有使能（**CHnPEN** = '0'，否则 **AD16C4Tn_CCVALn** 影子寄存器只有在下次更新事件发生时才更新）。通过软件方式，**AD16C4Tn_CCVALn** 寄存器的值可随时更新控制输出波形。

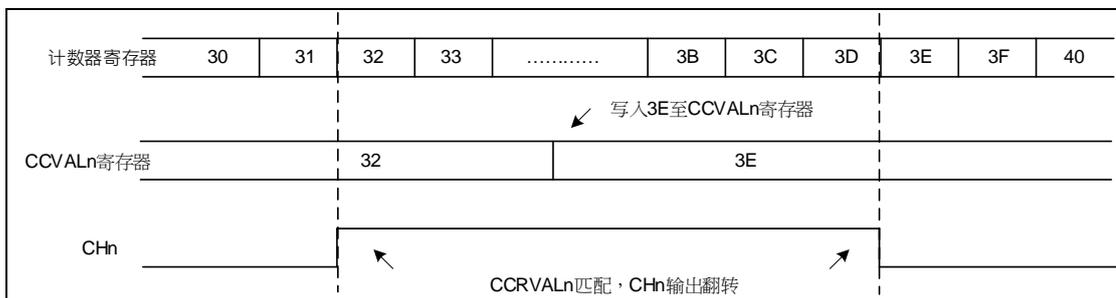


图 18-20 输出比较模式，触发 CHn

18.4.8.1 外部事件清除比较输出

ETF 输入端（AD16C4Tn_CHMRn 寄存器的 CHnOCLREN 位写'1'）上的高电平，可将给定通道的比较输出信号拉低。在下次更新事件（UEV）发生前，比较输出会一直保持为低。该功能只能应用在输出比较和 PWM 模式中，强制输出模式中不起作用。

ETR 信号可以接到电流控制比较器的输出端。该例中，ETR 须按如下流程配置：

1. 外部时钟源 2 关闭：AD16C4Tn_SMCON 寄存器的 ECM2EN 位写'0'
2. 外部触发极性（ETPOL）和外部触发滤波器（ETFLT）可根据用户需要配置

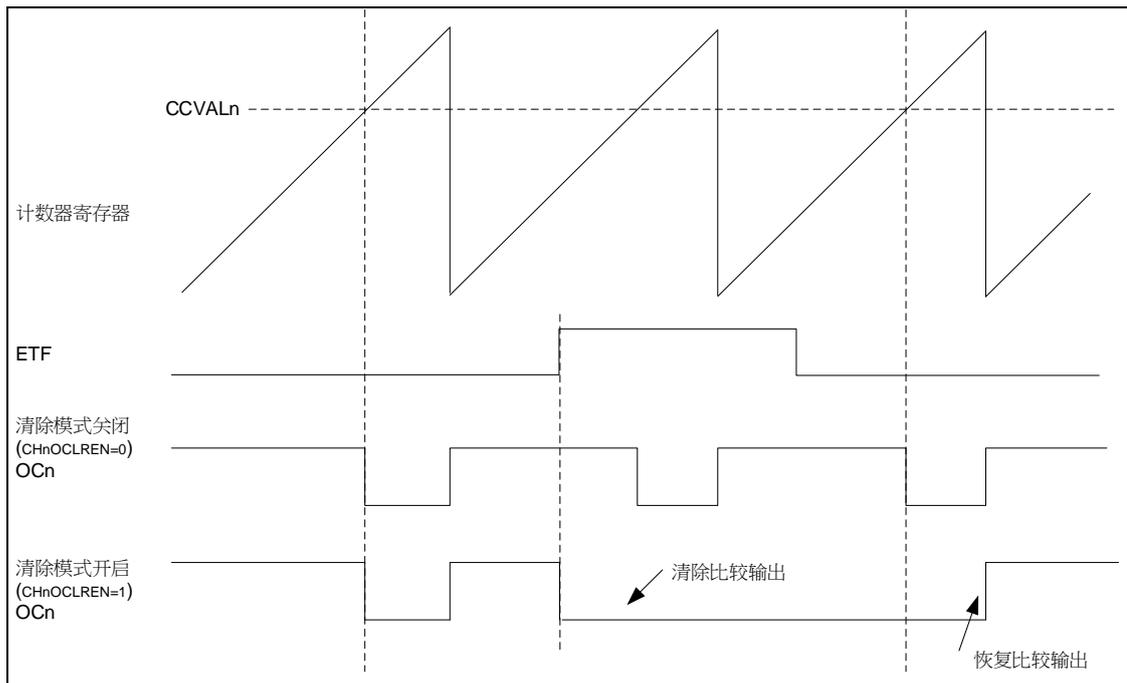


图 18-21 清除比较输出 CHn

18.4.8.2 强制输出模式

在输出模式中（AD16C4Tn_CHMRn 寄存器中 CCnSSEL = "00"），软件可强制将每个输出比较信号（CHn/CHnN）改为有效或无效状态，这种修改独立于输出比较寄存器和计数器的比较结果。

为了将某输出比较信号（CHn）强制为有效状态，需将相应的 AD16C4Tn_CHMRn 寄存器中 CHnMOD 位写"101"。因此，比较输出被强制为高（高时为有效状态）且 CHn 的值为 CCnPOL 极性位的相反值。

例如：CCnPOL= '0'（CHn 高电平有效），则 CHn 被强制为高电平。

对 AD16C4Tn_CHMRn 寄存器的 CHnMOD 位写"100"，比较输出可被置低。

无论怎样，AD16C4Tn_CCVALn 影子寄存器和计数器之间的比较仍然进行，相应的标志位仍可置位。

18.4.9 单脉冲模式

单脉冲模式 (SPMEN) 下, 响应某个触发后, 定时器的输出信道在可配置的延迟时间后产生一个脉冲, 脉冲长度可配。从模式控制器可控制计数器的启动。脉冲波形可在输出比较模式和 PWM 模式下产生。置位 **AD16C4Tn_CON1** 寄存器的 SPMEN 位可选择单脉冲模式。计数器会在下次更新事件 UEV 产生时自动停止。

只有比较值不同于计数器初始值时, 单脉冲才可以正确的产生。计数器开始计数前 (定时器等待触发), 必须如下配置:

- ◇ 递增计数: $CNTV < CCVALn \leq AR$ (特别地, $0 < CCVALn$)
- ◇ 递减计数: $CNTV > CCVALn$

基于 PWM 模式设置单脉冲输出波形的步骤如下:

- ◇ 设置 **AD16C4Tn_CHMRn** 寄存器的 CHnMOD 位, 选择 PWM 模式 1 或 2
- ◇ 设置 **AD16C4Tn_CCEPn** 寄存器的 CCnPOL 位, 选择通道端口 CHn 的输出极性
- ◇ 设置 **AD16C4Tn_CON1** 寄存器的 DIRSEL, CMSEL, SPMEN 位, 配置为递增或递减计数, PWM 普通波形模式, 单脉冲模式使能
- ◇ 设置 **AD16C4Tn_CHMR** 寄存器的 CH1PEN = 1, **AD16C4Tn_CON1** 寄存器的 ARPEN = 1, 使能比较寄存器和计数重载寄存器的缓冲功能 (也可以根据实际情况不使能缓冲)
- ◇ 设置 **AD16C4Tn_CCVALn** 寄存器和 **AD16C4Tn_AR** 寄存器, 配置单脉冲输出延时和脉宽时间
- ◇ 设置 **AD16C4Tn_SGE** 寄存器的 SGU 位来产生一个更新事件
- ◇ 设置 **AD16C4Tn_CON1** 寄存器的 CNTEN=1 来启动计数器, 也可以在触发模式下, 通过外部触发输入信号来触发硬件自动设置 CNTEN=1。

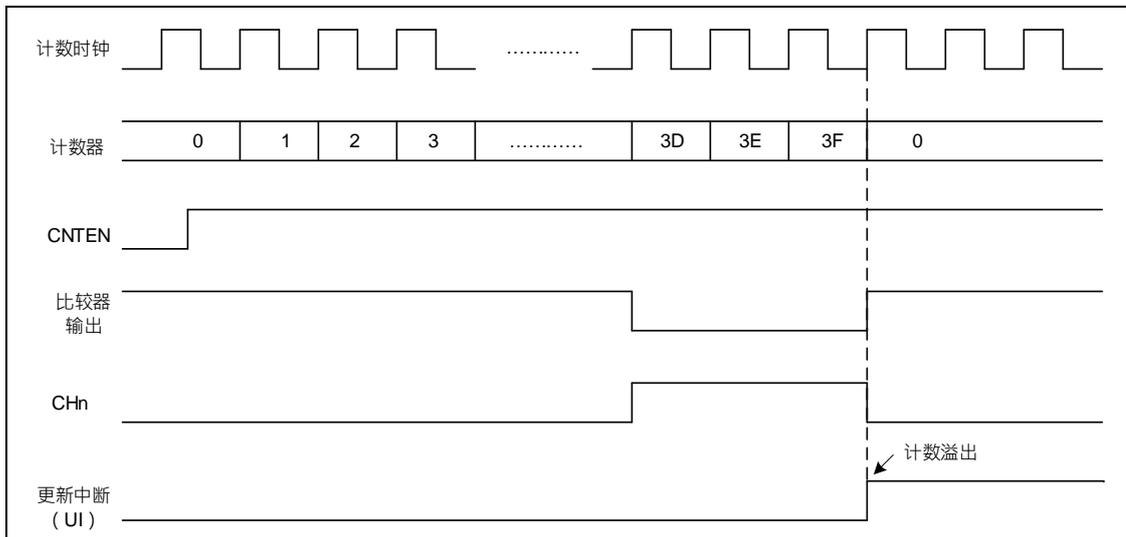


图 18-22 单脉冲模式

18.4.10 互补输出与死区时间

两个互补的通道输出信号，可以用来控制输出的瞬时开关。死区时间可配置。

每个输出可独立选择输出极性（主输出 CHn 或互补输出 CHnN），该操作可通过写 **AD16C4Tn_CCEP** 寄存器的 CCnPOL 和 CCnNPOL 位完成。

互补信号 CHn 和 CHnN 由几个控制位共同控制，分别是 **AD16C4Tn_CCEP** 寄存器中的 CCnEN 和 CCnNEN 位，**AD16C4Tn_BDCFG** 和 **AD16C4Tn_CON2** 寄存器中的 GOEN、OISSn、OISSnN、OFFSSI 及 OFFSSR 位。特别是死区时间使能后的空闲状态的切换（GOEN 变为 0）。

置位 CCnEN 和 CCnNEN 位，使能死区时间插入，若有刹车电路，同样需要置位 GOEN 位。**AD16C4Tn_BDCFG** 寄存器的 DT[7: 0]可以控制所有通道的死区时间的产生。根据比较输出波形，产生 CHn 和 CHnN 两路输出。若 CHn 和 CHnN 有效电平为高：

- ◇ CHn 的输出信号与参考信号一致。上升沿除外，相对参考信号的上升沿，CHn 输出会有延迟。
- ◇ CHnN 的输出信号与参考信号相反。上升沿除外，相对参考信号的下降沿，CHnN 输出会有延迟。

若延迟时间大于有效输出的宽度（CHn 或 CHnN），则相应的脉冲不会产生。

下图给出了死区时间输出信号和比较输出波形之间的关系。假设 CCnPOL = 0, CCnNP = 0, GOEN = 1, CCnEN = 1, 和 CCnNEN = 1

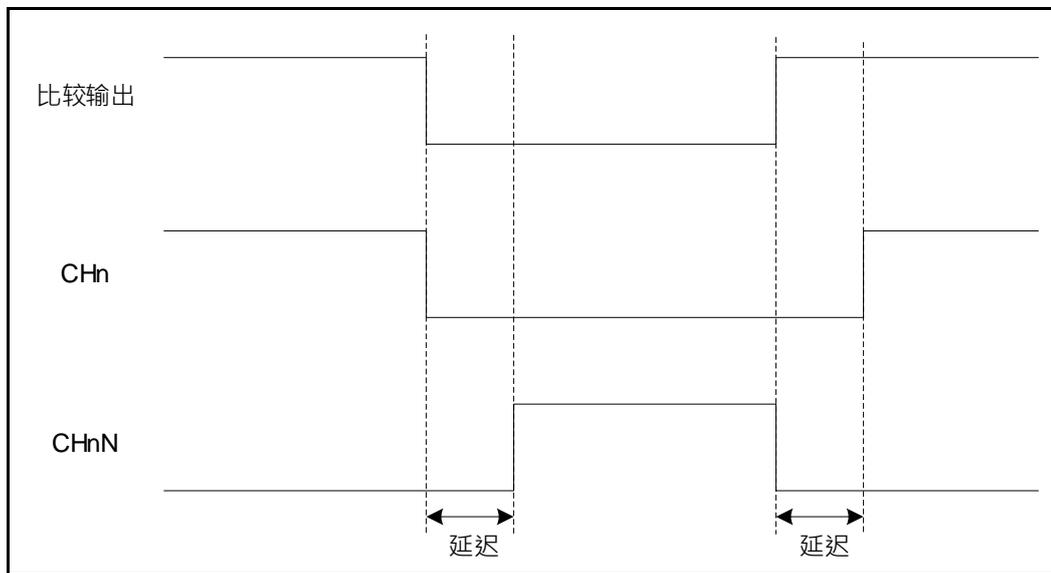


图 18-23 互补输出含死区时间插入

当 PWM 信道配置为互补输出时，如下寄存器控制位都会有缓冲：CHnMOD、CCnEN 和 CCnNEN。发生互补通道更新事件时，这些寄存器位才会真正生效，这样就可以预先设置好下一步的配置，并同时对所有互补信道的配置进行更新。互补通道更新事件可以通过设置 **AD16C4Tn_SGE** 寄存器的 SGCOM=1 产生，或由触发信号产生（由 **AD16C4Tn_SMCON** 寄存器的 TSSEL 位选择触发信号）。

18.4.11 刹车功能

刹车功能模式由以下几个控制位进行设置：**AD16C4Tn_BDCFG** 寄存器中的 **GOEN**、**OFFSSI** 和 **OFFSSR** 位，**AD16C4Tn_CON2** 寄存器中的 **OISSn** 和 **OISSnN** 位，输出使能信号和无效电平都会被修改。

刹车源可以是刹车输入引脚、时钟停振事件、比较器结果以及软件控制 **AD16C4Tn_SGE** 寄存器的 **SGBRK** 位。时钟停振事件由时钟控制器（RCU）中，时钟安全系统（CSS）产生。时钟安全系统（CSS）详细讯息可参考时钟安全系统章节。

系统复位后，刹车电路被禁止且 **GOEN** 位被复位。置位 **AD16C4Tn_BDCFG** 寄存器的 **BRKEN** 位可使能刹车功能，同样寄存器中，**BRKEN** 位可选择刹车输入信号的极性。**BRKEN** 和 **BRKP** 位可同时修改。对 **BRKEN** 和 **BRKP** 位写操作后，1 个 APB 时钟周期延时后写入值才会生效。因此，写操作后，需等待 1 个 APB 时钟周期后才能正确读回写入值。

由于 **GOEN** 的下降沿可以是异步的，在实际信号（作用在输出端）和同步控制位（**AD16C4Tn_BDCFG** 寄存器中）之间插入了一个同步电路。这也导致了异步和同步信号之间会产生一些延迟。特别是 **GOEN** 之前为低时对 **GOEN** 写 1 操作后，要读取正确值，必须先插入一个延时（空指令）。这是因为写入的是异步信号，而读取的是同步信号。

当发生刹车请求时（刹车输入端有刹车电平）：

- ◇ **GOEN** 位被异步清除，输出端进入无效状态，空闲状态或复位状态（**OFFSSI** 位选择）。即使 MCU 的振荡器关闭，该功能仍然有效。
- ◇ 一旦 **GOEN**=0，每个信道输出预先配置的电平。**AD16C4Tn_CON2** 寄存器中的 **OISSn** 位配置该电平。如果 **OFFSSI**=0，则定时器释放使能输出，否则使能输出一直为高。
- ◇ 当使用互补输出时：
 - 输出端首先被置于复位状态，无效状态（取决于极性）。这个过程是异步的，即使定时器没有时钟也有效。
 - 如果定时器时钟仍然存在，则死区时间生成器会重新生效，这样在死区时间后，**OISSn** 和 **OISSnN** 位的配置电平可驱动输出。这种情况下，**CHn** 和 **CHnN** 无法驱动输出端都为有效电平。
 - 由于对 **GOEN** 的重新同步，死区时间的周期会比通常情况下长一些（大约 2 个 **TIMER** 模块时钟周期）。
 - 如果 **OFFSSI** = 0，则定时器释放使能输出，否则使能输出保持或变高（一旦 **CCnEN** 和 **CCnNEN** 有一个变高时）。
- ◇ 当刹车状态标志位（**AD16C4Tn_RIF** 寄存器中的 **BRKI** 位）置位时，若 **AD16C4Tn_IER** 寄存器中的 **BRKI** 位置位，可触发中断。
- ◇ 当 **AD16C4Tn_BDCFG** 寄存器中的 **AOEN** 位置位时，在下次更新事件（**UEV**）发生时，**GOEN** 位会自动置位。例如，该功能可用于整形。否则，**GOEN** 位会保持为低，直到对其写'1'操作，该特性可用于安全方面的应用，可以将刹车输入端接到一个电源驱动的报警端、热敏传感器或其他安全器件上。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能置位（自动地或者通过软件）**GOEN**。同时，状态标志 **BRKI** 不能被清除。

除刹车输入和输出管理，为保证应用程序的安全，内部刹车电路具有写保护功能。用户可冻

结几个配置参数（死区时间，CHn/CHnN 极性和失能时状态，CHnMOD 配置，刹车使能和极性）。通过 **AD16C4Tn_BDCFG** 寄存器中的 LOCKLVL 位，可从三个保护等级中选择一种保护等级。MCU 复位后，LOCKLVL 位只能写一次。

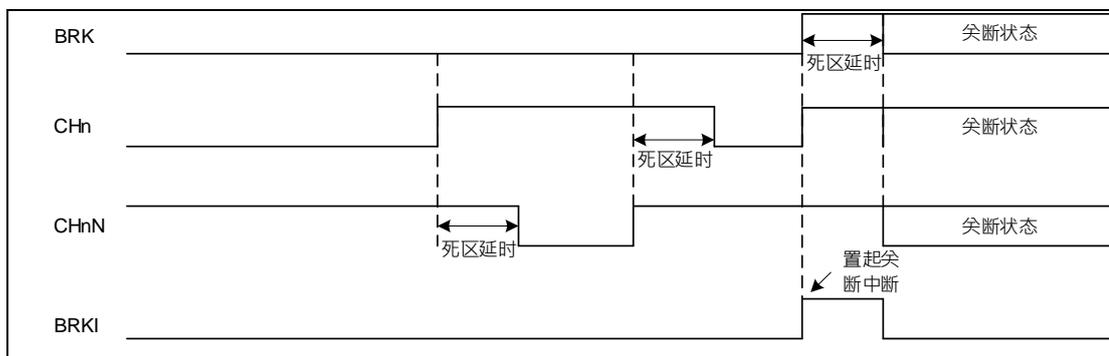


图 18-24 刹车输出行为

18.4.12 编码器接口模式

编码器接口模式的三种配置：若计数器只根据 I2 上的边沿计数，则 **AD16C4Tn_SMCON** 寄存器中的 **SMODS = "001"**；若计数器只根据 I1 上的边沿计数，则 **AD16C4Tn_SMCON** 寄存器中的 **SMODS = "010"**；若计数器同时根据 I1 和 I2 上的边沿计数，则 **AD16C4Tn_SMCON** 寄存器中的 **SMODS = "011"**。

配置 **AD16C4Tn_CCEP** 寄存器中的 **CC1POL** 和 **CC2POL** 位的值可选择 I1 和 I2 的极性。如果需要，也可以配置输入滤波器。

CH1 和 **CH2** 端口作为增量编码器的接口。当计数器使能时，计数器根据 I1 或 I2 上滤波后的有效电平变化时钟计数。I1 和 I2 滤波后的有效信号顺序会产生计数脉冲及方向信号。计数器是递增或递减计数由信号的跳变顺序决定，**AD16C4Tn_CON1** 寄存器中的 **DIRSEL** 计数方向位由硬件自动更新。

编码器接口模式的工作方式类似于一个带有方向选择的外部时钟。计数器在 0 到 **AD16C4Tn_AR** 寄存器中的自动重载值之间连续计数。因此，必须在开始计数前配置 **AD16C4Tn_AR** 寄存器。同样的，捕获器、预分频器、重复计数器、触发输出的特性正常工作。设定编码模式和选择外部时钟源 2 不兼容，不可以同时选择。

该模式下，计数器会根据增量式编码器的速度和方向自动修改，计数器的值反映的是编码器的位置。计数方向对应着连接传感器的旋转方向。

下表列出了所有的可能组合，假设 I1 和 I2 不同时变换。

有效边沿	有效边沿相对信号的 电平 (I1 滤波信号对应 I2, I2 滤波信号对应 I1)	I1 滤波信号		I2 滤波信号	
		上升	下降	上升	下降
仅在 I1 计数	高	下降	上升	不计数	不计数
	低	上升	下降	不计数	不计数
仅在 I2 计数	高	不计数	不计数	上升	下降
	低	不计数	不计数	下降	上升
在 I1 和 I2 上 计数	高	下降	上升	上升	下降
	低	上升	下降	下降	上升

表 18-1 计数方向与编码器信号的关系

外部增量编码器可直接与 MCU 连接，无需外部接口逻辑。而比较器通常用于将编码器的差分输出转换为数字信号，这极大地增加了抗噪声能力。编码器的第三个输出端用于指示机械零点，可以连接到外部中断输入引脚以触发一次计数复位。

下图给出了计数器操作的例子，给出了计数信号的产生和方向控制。同样给出了选择双边沿时，输入抖动如何被补偿。输入抖动可能发生在传感器靠近切换点处。

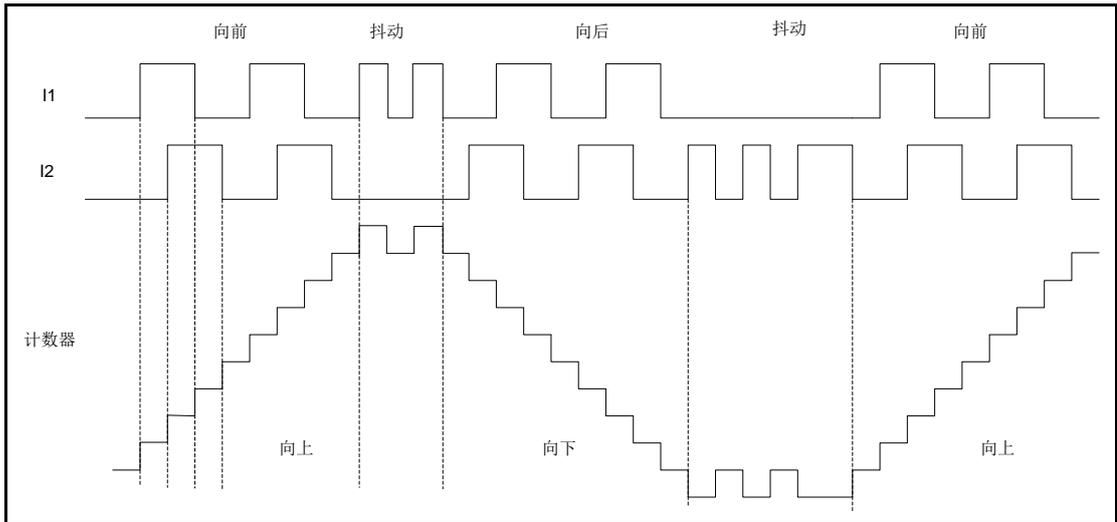


图 18-25 编码器接口模式下的计数操作

下图给出了计数器在 I1 滤波信号极性反相时的计数过程（除了 CC1POL = '1'，其他配置与上面一致）

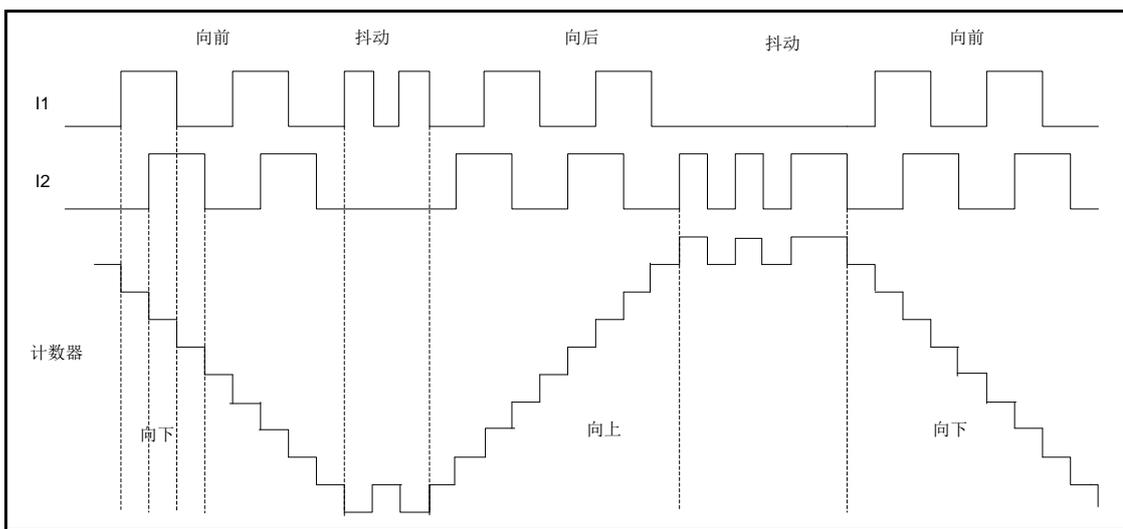


图 18-26 滤波后极性反相时编码器接口例子

当配置为编码器接口模式时，定时器可提供传感器的当前位置讯息。配置一个额外定时器为捕获模式，用于测量两个编码器事件的间隔，根据间隔时长获取动态讯息（速度、加速度、减速度）。编码器用于指示机械零点的输出就是此用处。根据编码器两个事件间隔，可以周期性的读取计数器的值。如果允许，可以将计数器值锁存到第三个输入捕获寄存器（捕获信号必须是周期性的且可由其它定时器产生）。条件允许时，可通过实时时钟产生 DMA 请求的方式读取计数器值。

18.4.13 输入异或功能

通过 **AD16C4Tn_CON2** 寄存器中 **I1SEL** 位，可将通道 1 的输入滤波器连接到 XOR 门的输出端，XOR 门联合了 CH1、CH2 和 CH3 三个输入引脚。

XOR 输出用于定时器的所有输入功能，如触发或输入捕获。该功能参见下节的霍尔传感器接口。

18.4.14 霍尔传感器接口

使用高级控制定时器产生 PWM 信号驱动马达，用另一个定时器作为“接口定时器”来连接霍尔传感器，请参见下图。3 个定时器输入脚(CH1、CH2、CH3)通过一个异或门连接到 I1 输入通道(通过设置 **GP16C4Tn_CON2** 寄存器中的 **I1SEL** 位来选择)，“接口定时器”捕获这个信号。

从模式控制器被配置为复位模式，从输入是 I1F 双边沿。这样每当 3 个输入之一变化时，计数器从 0 重新开始计数。由此产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”模式下，捕获/比较信道 1 被配置为捕获模式，捕获信号为 I1(捕获/比较通道(如：通道 1 输入部分))。捕获值反映了输入端两次变化之间的时间间隔，指示出了马达速度的讯息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级控制定时器 (**AD16C4Tn**) 各个通道的属性，**AD16C4Tn** 产生 PWM 信号驱动马达。因此，“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGOUT 输出被送到高级控制定时器 (**AD16C4Tn**)。

举例：霍尔输入连接到定时器，要求在每次霍尔输入发生变化之后的一个指定的时刻，改变高级控制定时器的 PWM 配置。

置 **GP16C4Tn_CON2** 寄存器的 **I1SEL** 位为'1'，配置三个定时器输入逻辑异或到 I1 输入，时基编程：置 **GP16C4Tn_AR** 为其最大值(计数器必须通过 I1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。

设置信道 1 为捕获模式(选中 I1)：置 **GP16C4Tn_CHMR1** 寄存器中 **CC1SSEL=01**，如果需要，还可以设置数字滤波器。

设置信道 2 为 PWM 模式 2，带指定的延时：置 **GP16C4Tn_CHMR1** 寄存器中的 **CH2OMOD=111** 和 **CC2SEL=00**。

选择 CH2REF 作为 TRGOUT 上的触发输出：置 **GP16C4Tn_CON2** 寄存器中的 **MMSEL=101**。在高级控制定时器 **AD16C4Tn** 中，正确的 IT 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(**AD16C4Tn_CON2** 寄存器中 **CCPCNTEN=1**)，由触发输入控制 COM 事件(**AD16C4Tn_CON2** 寄存器中 **CCUSEL=1**)。在一次 COM 事件后，下一步再写入 PWM 控制位(**CCnEN**、**CHnOCLREN**)，这可以在处理 CH2REF 上升沿的中断子程序里实现。

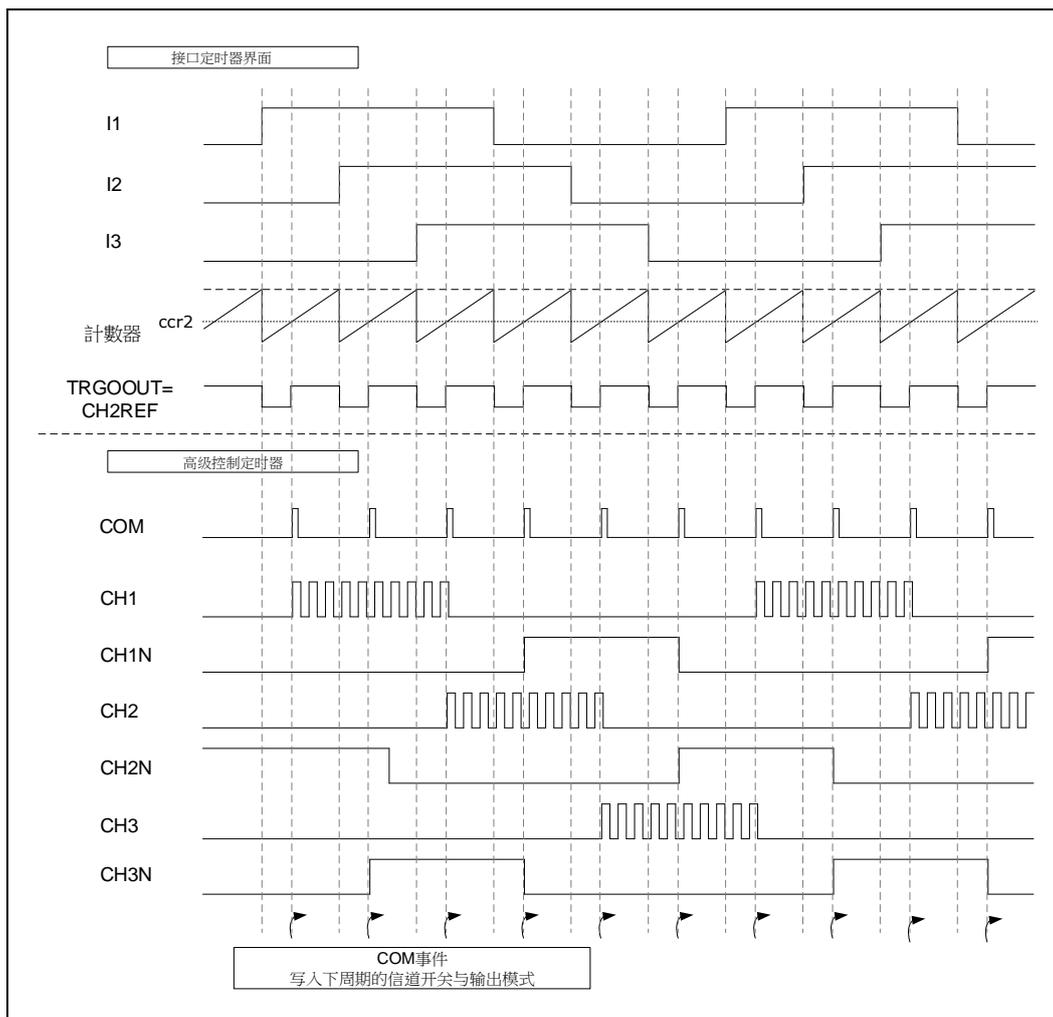


图 18-27 霍尔传感器接口范例

18.4.15 外部触发的同步

AD16C4Tn 定时器可在多种模式下与外部触发同步：复位模式、门控模式及触发模式。

18.4.15.1 复位模式

计数器及其预分频器可以在响应触发输入事件时重新初始化。此外，若 **AD16C4Tn_CON1** 寄存器的 **UERSEL** 位为低时会产生一次更新事件 **UEV**。所有预载寄存器 (**AD16C4Tn_AR**, **AD16C4Tn_CCVALn**) 都会因更新事件 **UEV** 而被更新。

在下面例子中，**I1** 输入端的上升沿让递增计数被清空：

- ◇ 配置信道 1 上检测 **I1** 上的上升沿。配置输入滤波周期（本例无需滤波器，故 **I1FLT** = "0000"）。触发捕获分频器没有使用，无需配置。**CC1SSEL** 位只选择输入捕获源，**AD16C4Tn_CHMR1** 寄存器中 **CC1SSEL** = "01"。**AD16C4Tn_CCEP** 寄存器中 **CC1POL** = 0 以确定极性（只检测上升沿）。
- ◇ 定时器配置为复位模式：**AD16C4Tn_SMCON** 寄存器中 **SMODS** = "100"。选择 **I1** 作为输入源：**AD16C4Tn_SMCON** 寄存器中 **TSSEL** = "101"。
- ◇ 启动计数器：**AD16C4Tn_CON1** 寄存器中 **CNTEN** = '1'。

计数器依据内部时钟开始计数，正常计数直到 **I1** 上出现上升沿。当 **I1** 上出现上升沿时，计数器会被清零且从 0 重新开始计数。同时，标志位置位（**AD16C4Tn_RIF** 寄存器中 **TRGI** 位），如果中断及 **DMA** 使能（取决于 **AD16C4Tn_IER** 寄存器中的 **TRGI** 和 **AD16C4Tn_DMAEN** 寄存器中的 **TRGIDE** 位），会发送中断及 **DMA** 请求。

下图给出了当自动重载寄存器 **AD16C4Tn_AR** = 0x36 时的信号变化。由于 **I1** 输入的再同步电路，**I1** 上的上升沿和计数器实际复位之间会存在延时。

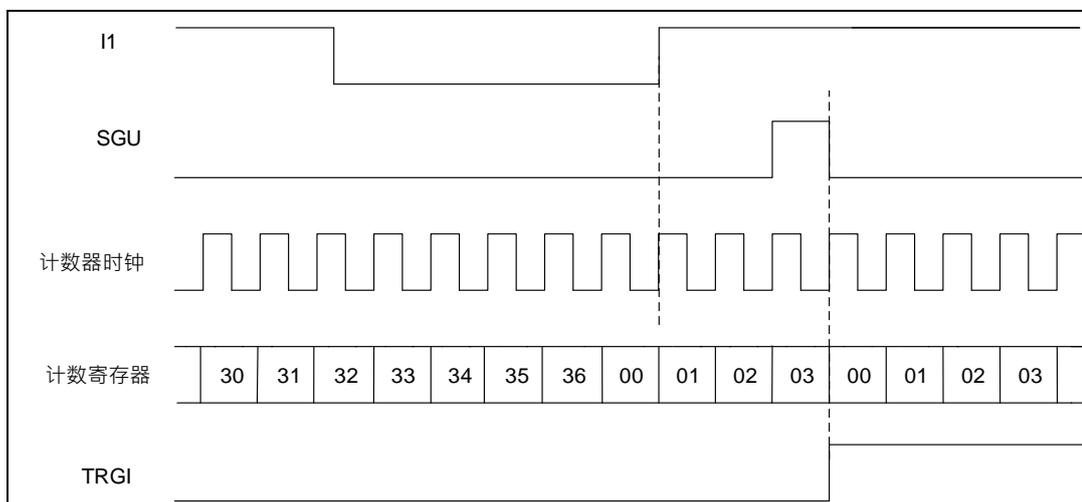


图 18-28 复位模式控制电路

18.4.15.2 门控模式

计数器根据选中的输入电平被使能。

下面的例子中，计数器只在 I1 输入为低电平时才递增计数：

- ◇ 配置信道 1 在 I1 上检测低电平。配置输入滤波周期(本例不需要滤波器, I1FLT = "0000")。触发捕获分频器没有使用, 无需配置。AD16C4Tn_CHMR1 寄存器中的 CC1SSEL = "01", 选择输入捕获源。AD16C4Tn_CCEP 寄存器中 CC1POL = '1', 确认极性(只检测低电平)。
- ◇ 配置定时器为门控模式: AD16C4Tn_SMCON 寄存器中 SMODS = "101"。选择 I1 作为输入源: AD16C4Tn_SMCON 寄存器中 TSSEL = "101"。
- ◇ 使能计数器: AD16C4Tn_CON1 寄存器中 CNTEN = '1' (门控模式中, 如果 CNTEN = '0', 无论触发输入为何电平, 计数器都不会启动)。

只要 I1 为低电平, 计数器依据内部时钟开始计数, 一旦 I1 为高则停止计数。由于 I1 输入端再同步电路的原因, I1 上出现上升沿和计数器实际停止之间会有一定的延时。

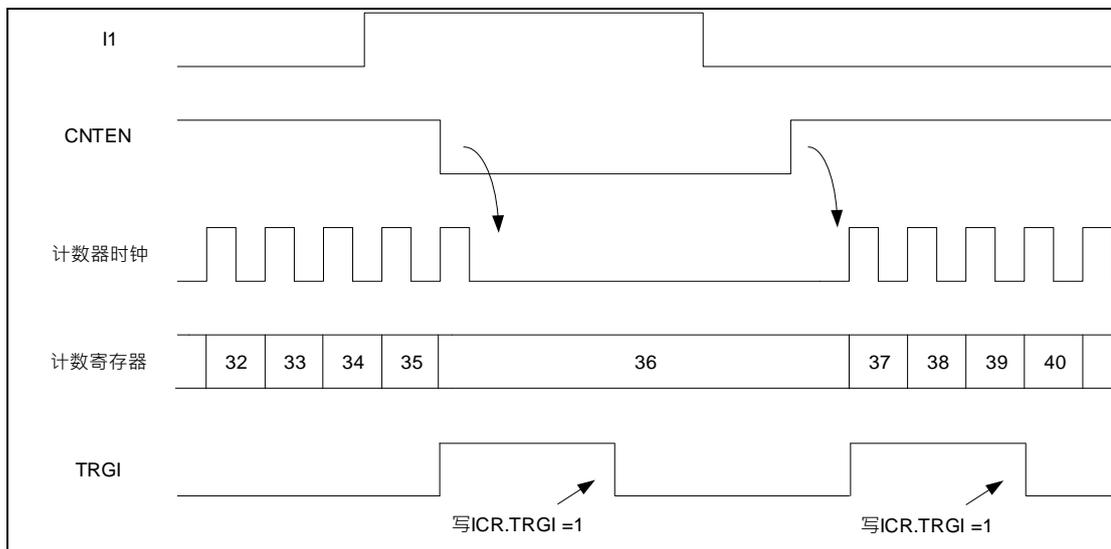


图 18-29 门控模式控制电路

18.4.15.3 触发模式

输入端选中的事件可以使能计数器。

下面的例子中，I2 输入端上的上升沿可以启动递增计数：

- ◇ 配置信道 2 可以检测 I2 上的上升沿。配置滤波时间（本例不需要滤波，I2FLT = "0000"）。触发捕获分频器没有使用，无需配置。AD16C4Tn_CHMR1 寄存器中 CC2SEL = "01"，用于选择捕获源。AD16C4Tn_CCEP 寄存器中 CC2POL = '0'，确认极性（只检测高电平）。
- ◇ 配置定时器为触发模式：AD16C4Tn_SMCON 寄存器中 SMODS = "110"。AD16C4Tn_SMCON 寄存器中 TSSEL = "110"，用于选择输入源。

I2 上出现上升沿时，计数器开始依据内部时钟计数并置位 TRGI 标志位。

由于 I2 输入的再同步原因，I2 上出现上升沿和计数器实际停止之间会有一定的延时。

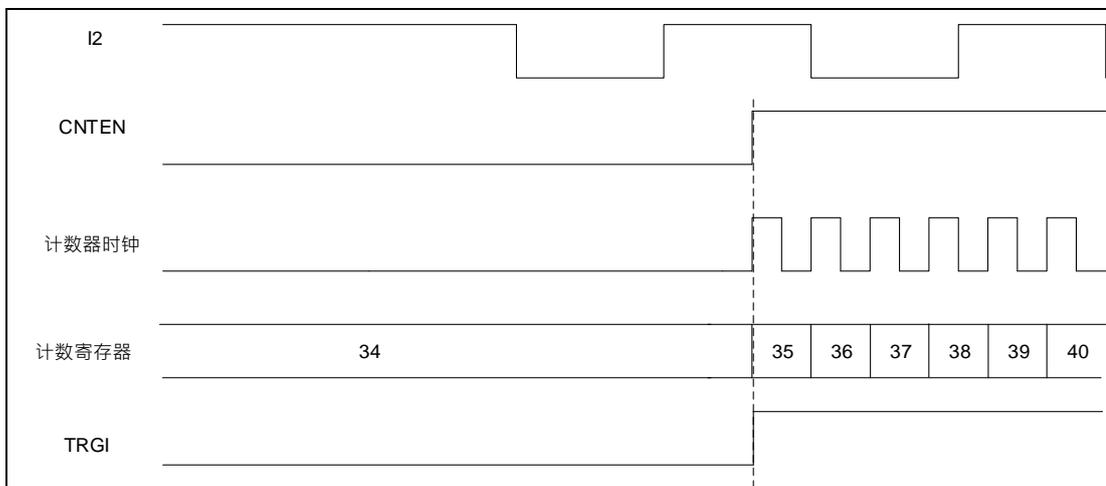


图 18-30 触发模式控制电路

18.4.15.4 选择外部时钟源 2 的触发模式

外部时钟源 2 可和其他模式一起使用（除编码模式）。ETR 信号可作为外部时钟输入，另一个输入可选择为触发输入（复位模式、门控模式或触发模式）。不推荐用 AD16C4Tn_SMCON 寄存器的 TSSEL 位选择 ETR 作为 TI。

下面的例子中，一旦 I1 上出现上升沿时，计数器会依据 ETR 信号的每个上升沿递增计数。

- ◇ 通过 AD16C4Tn_SMCON 寄存器，配置外部触发输入电路，过程如下：

ETFLT = "000": 无滤波

ETPOL = '0': 检测 ETR 的上升沿，ECM2EN = '1'使能外部时钟模式 2

- ◇ 配置信道 1 检测 I 的上升沿，过程如下：

I1FLT = "0000": 无滤波。

触发捕获分频器没有使用，无需配置。

AD16C4Tn_CHMR1 寄存器中 CC1SSEL = "01"选择输入捕获源，AD16C4Tn_CCEP 寄存器的 CC1POL = '0'确认极性（只检测上升沿）。

- ◇ 配置定时器为触发模式：AD16C4Tn_SMCON 寄存器中 SMODS = "110"。

AD16C4Tn_SMCON 寄存器中 TSSEL = "101"选择 I1 作为输入源。

I1 上出现上升沿时，计数器使能且 TRGI 标志位置位，然后计数器根据 ETR 上的上升沿开始计数。

由于 ETF 输入再同步电路的原因，ETR 信号的上升沿和实际计数器的复位会有延时。

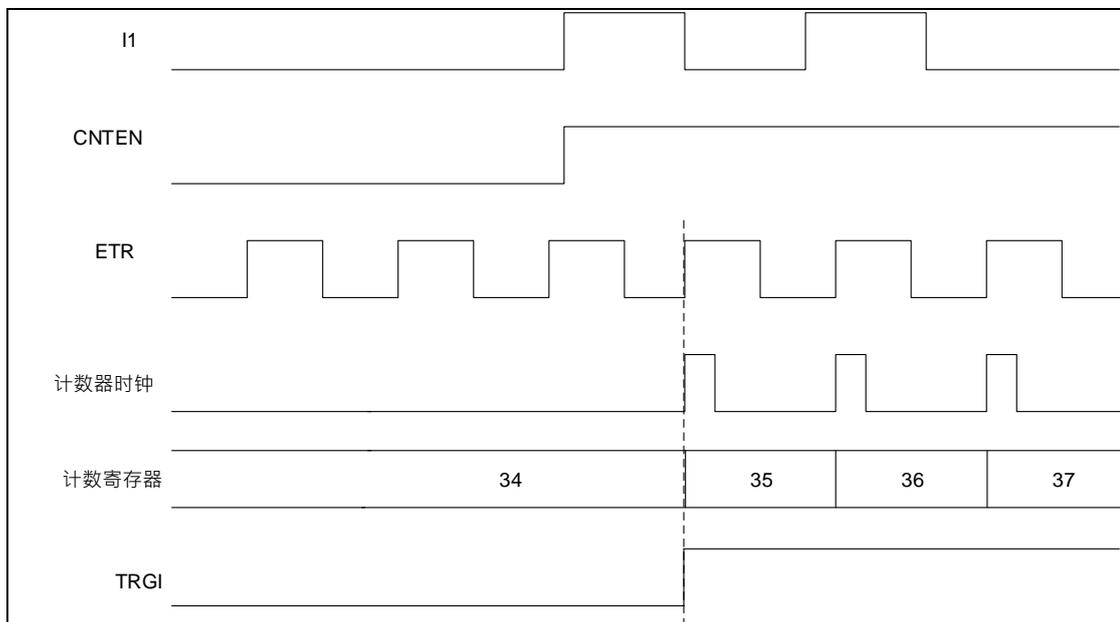


图 18-31 外部时钟源 2+触发模式下的控制电路

18.4.16 定时器同步

所有定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况

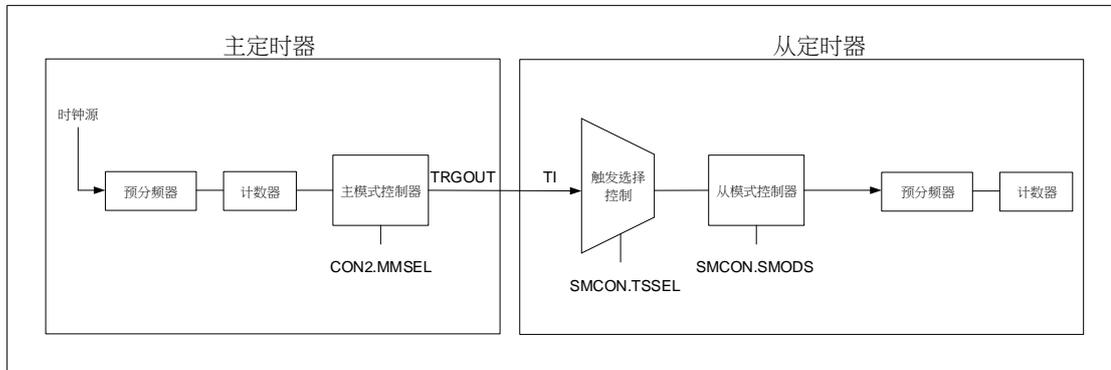


图 18-32 主/从定时器范例

18.4.16.1 使用一个定时器去使能其他定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考下图的连接。只当定时器 1 的 CH1REF 为高时，定时器 2 才对分频后的内部时钟计数。

配置定时器 1 为主模式，送出它的输出比较参考信号 (CH1REF) 为触发输出 (AD16C4T1_CON2 寄存器的 MMSEL=100)

- ◇ 配置定时器 1 的 CH1REF 波形 (AD16C4T1_CHMR1 寄存器)
- ◇ 配置定时器 2 从定时器 1 获得输入触发 (GP16C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为门控模式 (GP16C4T1_SMCON 寄存器的 SMODS=101)
- ◇ 配置 GP16C4T1_CON1 寄存器的 CNTEN=1 以使能定时器 2
- ◇ 配置 AD16C4T1_CON1 寄存器的 CNTEN=1 以启动定时器 1

注：定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

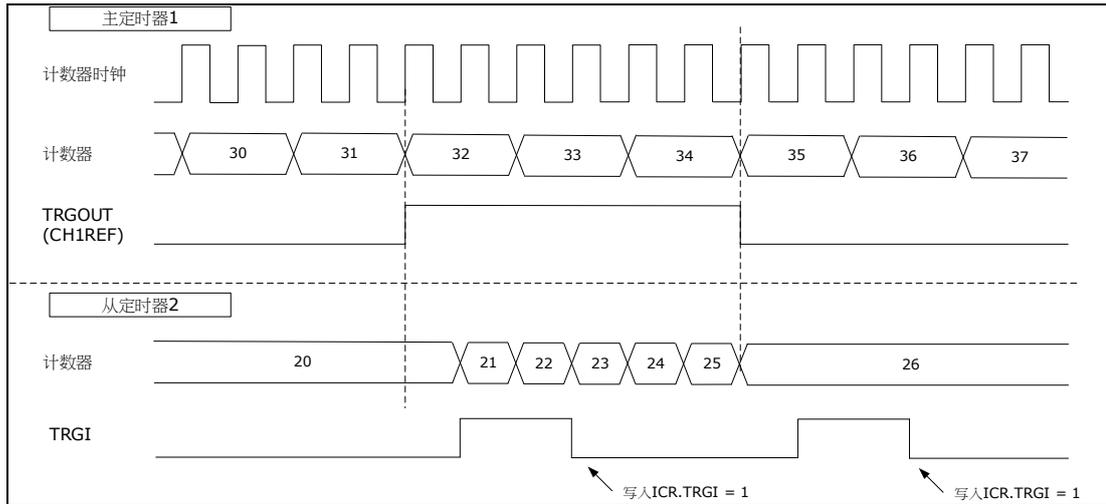


图 18-33 门控从定时器使用主定时器 CH1REF

在上图的例子中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 **AD16C4T1_SGE** 寄存器的 **SGU** 位即可复位定时器。

在下一个例子中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写 '0' 到 **AD16C4T1_CON1** 的 **CNTEN** 位将禁止定时器 1，定时器 2 随即停止。

- ◇ 配置定时器 1 为主模式，送出 **CNTEN** 位做为触发输出(**AD16C4T1_CON2** 寄存器的 **MMSEL=001**)。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(**GP16C4T1_SMCON** 寄存器的 **TSSEL=000**)
- ◇ 配置定时器 2 为门控模式(**GP16C4T1_SMCON** 寄存器的 **SMODS=101**)
- ◇ 配置 **AD16C4T1_SGE** 寄存器的 **SGU=1**，复位定时器 1。
- ◇ 配置 **GP16C4T1_SGE** 寄存器的 **SGU=1**，复位定时器 2。
- ◇ 写 '0xE7' 至定时器 2 的计数器(**GP16C4T1_COUNT**)，初始化它为 0xE7。
- ◇ 配置 **GP16C4T1_CON1** 寄存器的 **CNTEN=1** 以使能定时器 2。
- ◇ 配置 **AD16C4T1_CON1** 寄存器的 **CNTEN=1** 以启动定时器 1。
- ◇ 配置 **AD16C4T1_CON1** 寄存器的 **CNTEN=0** 以停止定时器 1。

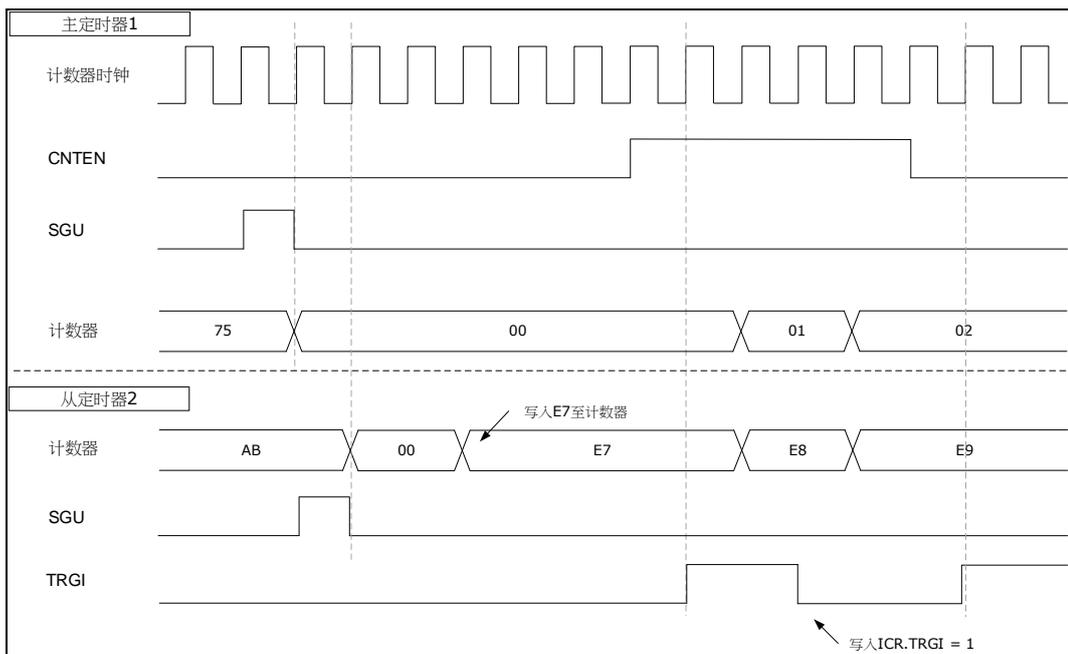


图 18-34 通过使能定时器 1 可以控制定时器 2

18.4.16.2 使用一个定时器去开启其他定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值(可以是非 0)按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CNTEN 位被自动地置'1'，同时计数器开始计数直到写'0'到 GP16C4T1_CON1 寄存器的 CNTEN 位。

- ◇ 配置定时器 1 的周期(AD16C4T1_AR 寄存器)。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP16C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为触发模式(GP16C4T1_SMCON 寄存器的 SMODS=110)
- ◇ 配置 AD16C4T1_CON1 寄存器的 CNTEN=1 以启动定时器 1。

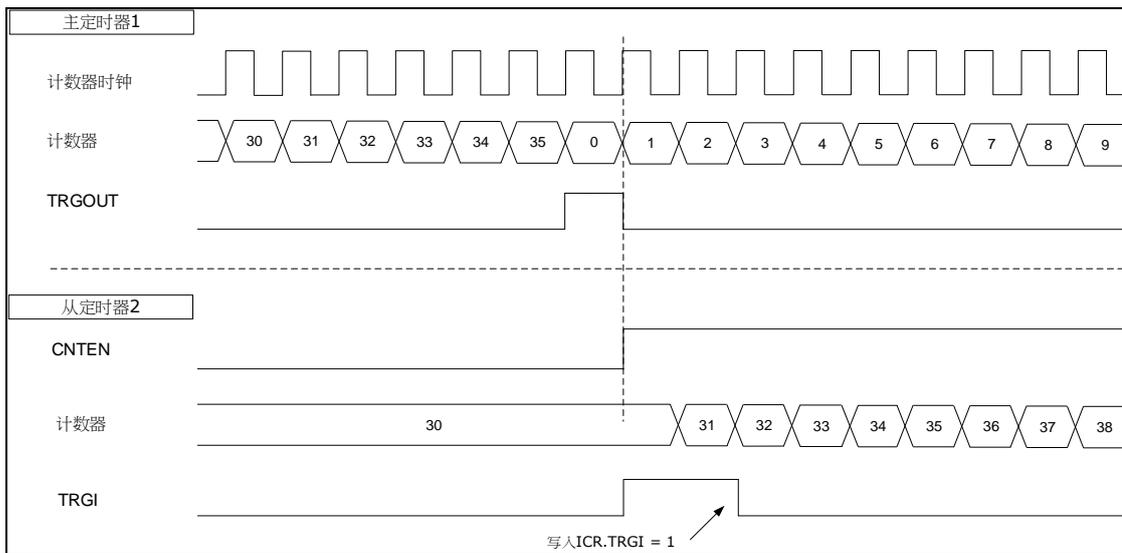


图 18-35 触发中从定时器使用主定时器更新事件

在上图例子中，可以在启动计数之前初始化两个计数器。上图显示在上上图相同配置情况下，使用触发模式而不是门控模式(GP16C4T1_SMCON 寄存器的 SMODS=110)的动作。

18.4.16.3 使用外部触发同步开始两个定时器

这个例子中当定时器 1 的 I1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见下图。为保证计数器的对齐，定时器 1 必须配置为主/从模式(对应 I1 为从，对应定时器 2 为主)：

- ◇ 配置定时器 1 为主模式，送出它的使能做为触发输出(AD16C4T1_CON2 寄存器的 MMSEL=001)。
- ◇ 配置定时器 1 为从模式，从 I1 获得输入触发(AD16C4T1_SMCON 寄存器的 TSSEL='101')。
- ◇ 配置定时器 1 为触发模式(AD16C4T1_SMCON 寄存器的 SMODS='110')。
- ◇ 配置定时器 1 为主/从模式，AD16C4T1_SMCON 寄存器的 MSCFG='1'。
- ◇ 配置定时器 2 从定时器 1 获得输入触发(GP16C4T1_SMCON 寄存器的 TSSEL=000)
- ◇ 配置定时器 2 为触发模式(GP16C4T1_SMCON 寄存器的 SMODS='110')。

当定时器 1 的 I1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TRGI 标志也同时被设置。

注：在这个例子中，在启动之前两个定时器都被初始化(设置相应的 SGU 位)，两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器(AD16C4T1_COUNT)在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNTEN 和计数器时钟之间有个延迟。

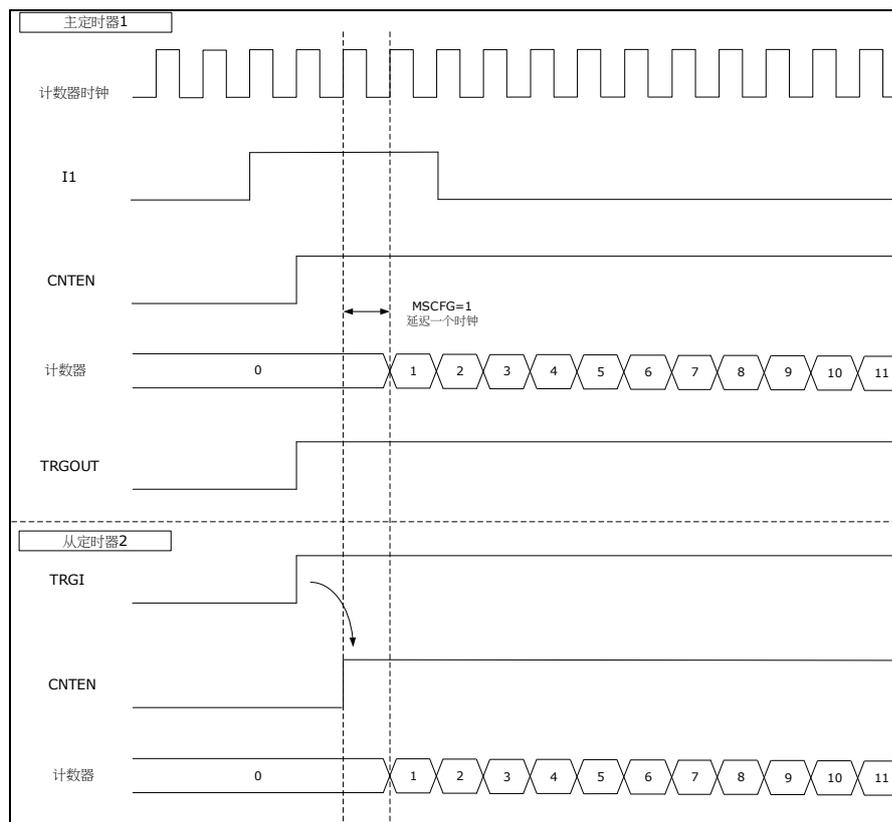


图 18-36 使用定时器 1 的 I1 输入触发定时器 1 和定时器 2

18.4.17 调试模式

当微控制器进入调试模式(Cortex™-M0 核停止运行)，计数器停止计数。

18.5 特殊功能寄存器

18.5.1 寄存器列表

AD16C4T 寄存器列表			
名称	偏移地址	类型	描述
AD16C4Tn_CON1	00 _H	R/W	控制寄存器 1
AD16C4Tn_CON2	04 _H	R/W	控制寄存器 2
AD16C4Tn_SMCON	08 _H	R/W	从模式控制寄存器
AD16C4Tn_IER	0C _H	W1	中断使能寄存器
AD16C4Tn_IDR	10 _H	W1	中断禁止寄存器
AD16C4Tn_IVS	14 _H	R	中断有效状态寄存器
AD16C4Tn_RIF	18 _H	R	原始中断标志寄存器
AD16C4Tn_IFM	1C _H	R	中断屏蔽标志寄存器
AD16C4Tn_ICR	20 _H	C_W1	中断标志清除寄存器
AD16C4Tn_SGE	24 _H	W1	软件生成事件寄存器
AD16C4Tn_CHMR1	28 _H	R/W	捕获/比较模式寄存器 1
AD16C4Tn_CHMR2	2C _H	R/W	捕获/比较模式寄存器 2
AD16C4Tn_CCEP	30 _H	R/W	捕获/比较使能极性寄存器
AD16C4Tn_COUNT	34 _H	R/W	计数器
AD16C4Tn_PRES	38 _H	R/W	时钟预分频器
AD16C4Tn_AR	3C _H	R/W	自动重载寄存器
AD16C4Tn_REPAR	40 _H	R/W	重复计数寄存器
AD16C4Tn_CCVAL1	44 _H	R/W	通道捕获/比较寄存器 1
AD16C4Tn_CCVAL2	48 _H	R/W	通道捕获/比较寄存器 2
AD16C4Tn_CCVAL3	4C _H	R/W	通道捕获/比较寄存器 3
AD16C4Tn_CCVAL4	50 _H	R/W	通道捕获/比较寄存器 4
AD16C4Tn_BDCFG	54 _H	R/W	刹车和死区配置寄存器
AD16C4Tn_DMAEN	58 _H	R/W	DMA 事件使能寄存器

CMSEL	Bits 6-5	R/W	<p>中央对齐模式选择</p> <p>00: 边沿对齐模式，根据 CON1 寄存器的 DIRSEL 位，使得计数器递增或递减</p> <p>01: 中央对齐模式 1，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，只在计数器向下计数时被设置</p> <p>10: 中央对齐模式 2，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，只在计数器向上计数时被设置</p> <p>11: 中央对齐模式 3，计数器交替地向上和向下计数。配置为输出的信道(CHMRn 寄存器中 CCnSSEL=00) 的输出比较中断标志位，在计数器向上和向下计数时皆会被设置</p>
DIRSEL	Bit 4	R/W	<p>计数方向选择</p> <p>当计数器配置为中央对齐模式时，此位只能读取计数器的计数方向</p> <p>0: 计数器向上计数</p> <p>1: 计数器向下计数</p>
SPMEN	Bit 3	R/W	<p>单脉冲模式</p> <p>0: 单脉冲模式禁止，计数器不停止</p> <p>1: 单脉冲模式使能，计数器在发生下一次更新事件时，清除 CNTEN 位，计数器停止</p>
UERSEL	Bit 2	R/W	<p>更新事件请求来源选择</p> <p>设置更新事件(UEV)的来源</p> <p>0: 若使能 UI 中断或 DMA 请求，下列事件中产生更新事件请求</p> <ul style="list-style-type: none"> - 计数器溢出/下溢 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 <p>1: 若使能 UI 中断或 DMA 请求，只在计数器溢出/下溢时产生更新事件请求</p>
DISUE	Bit 1	R/W	<p>更新事件禁止</p> <p>设置更新事件(UEV)的产生</p> <p>0: 更新事件(UEV) 使能，下列事件中产生更新事件请求</p> <ul style="list-style-type: none"> - 计数器溢出/下溢 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 <p>缓冲寄存器载入预装载值</p> <p>1: 更新事件(UEV)禁止，不产生更新事件请求，AR、PRES、CCVALn 寄存器保持数值</p> <p>禁止更新事件时，设置 SGE 寄存器的 SGU 位或</p>

			从模式中产生的复位请求，计数器和预分频器仍会被重新初始化
CNTEN	Bit 0	R/W	<p>计数器使能</p> <p>使能计数器后，在外部时钟模式、门控模式和编码模式才能运作。触发模式则可以由硬件设置 CNTEN 位</p> <p>0: 计数器禁止</p> <p>1: 计数器使能</p>

18.5.2.2 控制寄存器 2 (AD16C4Tn_CON2)

控制寄存器 2 (AD16C4Tn_CON2)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	OISS4	OISS3N	OISS3	OISS2N	OISS2	OISS1N	OISS1	I1SEL	MMSEL<2:0>			CCDMASEL	CCUSEL		CCPCEN

—	Bits 31-15	—	—
OISS4	Bit 14	R/W	通道 4 输出的空闲状态选择位 参考 OISS1 描述
OISS3N	Bit 13	R/W	通道 3 互补输出的空闲状态选择位 参考 OISS1N 描述
OISS3	Bit 12	R/W	通道 3 输出的空闲状态 3 选择位 参考 OISS1 描述
OISS2N	Bit 11	R/W	通道 2 互补输出的空闲状态选择位 参考 OISS1N 描述
OISS2	Bit 10	R/W	通道 2 输出的空闲状态选择位 参考 OISS1 描述
OISS1N	Bit 9	R/W	通道 1 互补输出的空闲状态选择位 0: 当 GOEN=0, 在一段死区时间后, CH1N=0 1: 当 GOEN=0, 在一段死区时间后, CH1N=1 注意: 当 BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 1,2,或 3 后, OISS1N 不可更改。
OISS1	Bit 8	R/W	通道 1 输出的空闲状态选择位 0: 当 GOEN=0, 如果 CH1ON 已实现, 在一段死区时间后, CH1=0 1: 当 GOEN=0, 如果 CH1ON 已实现, 在一段死区时间后, CH1=1 注意: 当 BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 1,2,或 3 后, OISS1N 不可更改。
I1SEL	Bit 7	R/W	选择 I1 引脚功能 0: AD16C4Tn_CH1 引脚与 I1 输入连接 1: AD16C4Tn_CH1, CH2 和 CH3 引脚与 I1 输入 (XOR) 连接。
MMSEL	Bits 6-4	R/W	主模式选择 设置在主模式下发送到从定时器的同步信号 (TRGOUT)与 ADC 输入 000: 复位 - 设置 SGE 寄存器信号用于同步触发输出(TRGOUT)。从模式的复位触发产生的复

			<p>位信号(TRGOUT)则与实际信号相差一个时钟</p> <p>001: 使能 - 计数器的使能信号 CNTEN 用于同步触发输出(TRGOUT), 可用于同步使能数个定时器。门控模式下, 是使用 CON1 寄存器的 CNTEN 位与触发输入信号逻辑产生。当计数器使能信号受控于触发输入时, TRGOUT 上会有一个时钟延迟, 可设置 SMCON 寄存器的 MSCFG 位延迟一个时钟同步定时器计数器</p> <p>010: 更新事件 - 更新事件被用于同步触发输出(TRGOUT)。一个主定时器的更新事件可当作从定时器的预分频器时钟</p> <p>011: 比较脉冲 - 在发生一次捕获或一次比较成功时, 当要设置 CH1I 标志时, 触发输出送出一个正脉冲(TRGOUT)</p> <p>100: 比较信号 - CH1REF 信号用于触发输出 (TRGOUT)</p> <p>101: 比较信号 - CH2REF 信号用于触发输出 (TRGOUT)</p> <p>110: 比较信号 - CH3REF 信号用于触发输出 (TRGOUT)</p> <p>111: 比较信号 - CH4REF 信号用于触发输出 (TRGOUT)</p>
CCDMASEL	Bit 3	R/W	<p>捕获/比较事件的 DMA 选择</p> <p>0: 当发生 CHn 事件时, 设置 CHn DMA 请求</p> <p>1: 当发生更新事件时, 设置 CHn DMA 请求</p>
CCUSEL	Bit 2	R/W	<p>捕获/比较更新控制选择</p> <p>此功能只有在有互补输出通道作用</p> <p>0: 在捕获/比较预装载时(CCPCEN =1), 只能通过 SGR 寄存器的 SGCOM 位更新</p> <p>1: 在捕获/比较预装载时(CCPCEN =1), 可通过 SGR 寄存器的 SGCOM 与 TI 的上升沿时被更新</p>
—	Bit 1	—	—
CCPCEN	Bit 0	R/W	<p>捕获/比较预装载控制</p> <p>设置后只在通信事件(COM), 即 SGE 寄存器的 SGCOM 与 TI 的上升沿时更新</p> <p>0: CCnEN, CCnNEN 和 CHnMOD 位预装载禁止</p> <p>1: CCnEN, CCnNEN 和 CHnMOD 位预装载使能</p>

			<p>1001: 采样频率$f_{DTS} / 8, N = 8$ 1010: 采样频率$f_{DTS} / 16, N = 5$ 1011: 采样频率$f_{DTS} / 16, N = 6$ 1100: 采样频率$f_{DTS} / 16, N = 8$ 1101: 采样频率$f_{DTS} / 32, N = 5$ 1110: 采样频率$f_{DTS} / 32, N = 6$ 1111: 采样频率$f_{DTS} / 32, N = 8$</p>
MSCFG	Bit 7	R/W	<p>主/从模式 0: 写入0无效 1: 延迟触发输入 (TI) 上的事件来允许当前计时器和其从器件之间的同步。该设置有效用于使用单个外部事件来同步多个计时器。</p>
TSSEL	Bits 6-4	R/W	<p>触发选择 设置触发选择, 用于同步寄存器 000: 内部触发 0 (IT0) 001: 内部触发 1 (IT1) 010: 内部触发 2 (IT2) 011: 内部触发 3 (IT3) 100: I1 边沿检测(I1F_ED) 101: I1 滤波后信号 110: I2 滤波后信号 111: 外部触发输入</p>
CHCSEL	Bit 3	R/W	<p>CHnREF 清除选择 0: CHnREF_CLR 连接至比较器输入 CMP_IN 1: CHnREF_CLR 连接至外部触发信号 ETF</p>
SMODS	Bits 2-0	R/W	<p>从模式选择 000: 从模式关闭 - 设置 CON1 寄存器 CNTEN 位, 计数器由内部时钟计数 001: 编码器模式 1 - 计数器向上/向下计数 I2 边沿, 取决于 I1 电平 010: 编码器模式 2 - 计数器向上/向下计数 I1 边沿检测边沿, 取决于 I2 边沿检测电平 011: 编码器模式 3 - 计数器向上/向下计数 I1 边沿检测和 I2 边沿检测边沿, 取决于另一个输入的电平 100: 复位模式 - 选中的触发输入(TI)的上升沿重新初始化计数器, 并且产生一次更新事件 101: 门控模式 - 当触发输入(TI)为高电平时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控 110: 触发模式 - 计数器在触发输入 TI 的上升沿启动(但不复位), 只有计数器的启动是受控 111: 外部时钟模式 1 - 选中的触发输入(TI) 的上</p>

			<p>升沿驱动计数器</p> <p>注：如果 I1 双边沿检测被选为触发输入 (TSSEL='100')，不能使用门控模式。I1 每一次转换，I1 双边沿检测就会输出 1 个脉冲，而门控模式则是检查触发信号的电平</p>
--	--	--	--

从定时器	IT0(TSSEL =000)	IT1(TSSEL =001)	IT2(TSSEL =010)	IT3(TSSEL=011)
AD16C4T1	GP32C4T1	GP16C4T1	GP16C4T2	GP16C4T3

表 18-2 AD16C4Tn 内部触发连接

18.5.2.4 中断使能寄存器 (AD16C4Tn_IER)

中断使能寄存器 (AD16C4Tn_IER)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI		BRKI	TRGI	COMI	CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	W1	捕获溢出 4 中断使能 0: 写入0无效 1: 捕获溢出CH4中断使能
CH3OVI	Bit 11	W1	捕获溢出 3 中断使能 0: 写入 0 无效 1: 捕获溢出 CH3 中断使能
CH2OVI	Bit 10	W1	捕获溢出 2 中断使能 0: 写入 0 无效 1: 捕获溢出 CH2 中断使能
CH1OVI	Bit 9	W1	捕获溢出 1 中断使能 0: 写入 0 无效 1: 捕获溢出 CH1 中断使能
—	Bit 8	—	—
BRKI	Bit 7	W1	刹车中断使能 0: 写入 0 无效 1: 刹车中断使能
TRGI	Bit 6	W1	触发中断使能 0: 写入 0 无效 1: 触发中断使能
COMI	Bit 5	W1	通信中断使能 0: 写入 0 无效 1: 通信中断使能
CH4I	Bit 4	W1	捕获/比较 4 捕获中断使能 0: 写入 0 无效 1: 捕获 CH4 中断使能
CH3I	Bit 3	W1	捕获/比较 3 捕获中断使能 0: 写入 0 无效 1: 捕获 CH3 中断使能
CH2I	Bit 2	W1	捕获/比较 2 捕获中断使能 0: 写入 0 无效 1: 捕获 CH2 中断使能

CH1I	Bit 1	W1	捕获/比较 1 捕获中断使能 0: 写入 0 无效 1: 捕获 CH1 中断使能
UI	Bit 0	W1	更新中断使能 0: 写入 0 无效 1: 更新中断使能

18.5.2.5 中断禁止寄存器 (AD16C4Tn_IDR)

中断禁止寄存器 (AD16C4Tn_IDR)																															
偏移地址: 10H																															
复位值: 00000000_00000000_00000000_00000000B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI		BRKI	TRGI	COMI	CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	W1	捕获溢出 4 中断禁止 0: 写入0无效 1: 捕获溢出CH4中断禁止
CH3OVI	Bit 11	W1	捕获溢出 3 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH3 中断禁止
CH2OVI	Bit 10	W1	捕获溢出 2 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH2 中断禁止
CH1OVI	Bit 9	W1	捕获溢出 1 中断禁止 0: 写入 0 无效 1: 捕获溢出 CH1 中断禁止
—	Bit 8	—	保留
BRKI	Bit 7	W1	刹车中断禁止 0: 写入 0 无效 1: 刹车中断禁止
TRGI	Bit 6	W1	触发中断禁止 0: 写入 0 无效 1: 触发中断禁止
COMI	Bit 5	W1	通信中断禁止 0: 写入 0 无效 1: 通信中断禁止
CH4I	Bit 4	W1	捕获/比较 4 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH4 中断禁止
CH3I	Bit 3	W1	捕获/比较 3 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH3 中断禁止
CH2I	Bit 2	W1	捕获/比较 2 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH2 中断禁止

CH1I	Bit 1	W1	捕获/比较 1 捕获中断禁止 0: 写入 0 无效 1: 捕获 CH1 中断禁止
UI	Bit 0	W1	更新中断禁止 0: 写入 0 无效 1: 更新中断禁止

18.5.2.6 中断有效状态寄存器 (AD16C4Tn_IVS)

中断有效状态寄存器 (AD16C4Tn_IVS)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI		BRKI	TRGI	COMI	CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	捕获溢出 4 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH4 中断禁止 1: 捕获溢出 CH4 中断使能
CH3OVI	Bit 11	R	捕获溢出 3 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH3 中断禁止 1: 捕获溢出 CH3 中断使能
CH2OVI	Bit 10	R	捕获溢出 2 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH2 中断禁止 1: 捕获溢出 CH2 中断使能
CH1OVI	Bit 9	R	捕获溢出 1 中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获溢出 CH1 中断禁止 1: 捕获溢出 CH1 中断使能
—	Bit 8	—	—
BRKI	Bit 7	R	刹车中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 刹车中断禁止 1: 刹车中断使能
TRGI	Bit 6	R	触发中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 触发中断禁止 1: 触发中断使能
COMI	Bit 5	R	通信中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 通信中断禁止 1: 通信中断使能
CH4I	Bit 4	R	捕获/比较 4 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止

			0: 捕获 CH4 中断禁止 1: 捕获 CH4 中断使能
CH3I	Bit 3	R	捕获/比较 3 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH3 中断禁止 1: 捕获 CH3 中断使能
CH2I	Bit 2	R	捕获/比较 2 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH2 中断禁止 1: 捕获 CH2 中断使能
CH1I	Bit 1	R	捕获/比较 1 捕获中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 捕获 CH1 中断禁止 1: 捕获 CH1 中断使能
UI	Bit 0	R	更新中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 更新中断禁止 1: 更新中断使能

18.5.2.7 原始中断标志寄存器 (AD16C4Tn_RIF)

原始中断标志寄存器 (AD16C4Tn_RIF)																															
偏移地址: 18 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI		BRKI	TRGI	COMI	CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	捕获溢出4原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出CH4原始中断产生
CH3OVI	Bit 11	R	捕获溢出3原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH3 原始中断产生
CH2OVI	Bit 10	R	捕获溢出2原始中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH2 原始中断产生
CH1OVI	Bit 9	R	捕获溢出1原始中断标志 当CH1I已经被设置又再次收到捕获信号将计数器数值写入CCVAL1寄存器中。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 捕获溢出 CH1 原始中断产生
—	Bit 8	—	—
BRKI	Bit 7	R	刹车原始中断标志 产生刹车事件时产生。此位由硬件设置 1, 设置ICR 寄存器清除 0: 无中断产生 1: 刹车原始中断产生
TRGI	Bit 6	R	触发原始中断标志 产生触发事件时产生(在从模式-门控时, 在任一沿产生, 其他模式则在有效沿时产生)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 触发原始中断产生
COMI	Bit 5	R	通信原始中断标志

			产生通信事件时产生(在捕获/比较预装载控制使能时, 更新 CCnEN, CCnNEN, CHnMOD 位)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 通信原始中断产生
CH4I	Bit 4	R	捕获/比较 4 原始中断标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 4 原始中断产生
CH3I	Bit 3	R	捕获/比较 3 原始中断标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 3 原始中断产生
CH2I	Bit 2	R	捕获/比较 2 原始中断标志 参照 CH1I 描述 0: 无中断产生 1: 捕获/比较 2 原始中断产生
CH1I	Bit 1	R	捕获/比较 1 原始中断标志 通道 CH1 设置为输出: 计数器匹配 CCVAL1 寄存器时设置, 在中央对齐模式依据 CMSEL 位在递增、递减、递增/递减时产生。当 CCVAL1 寄存器大于 AR 寄存器时, 则在递增边界或递减边界时置起。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 原始中断产生 通道 CH1 设置为输入: 发生捕获事件, 将计数器捕获至 CCVAL1 寄存器中, 于 I1 的有效沿产生。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 捕获/比较 1 原始中断产生
UI	Bit 0	R	更新原始中断标志 当 CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断 - 当计数器数值上溢或下溢时 (重复计数器为 0 时时产生更新事件)。 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器重新初始化时。 - 当计数器被触发事件重新初始化时 0: 无中断产生 1: 更新原始中断产生

18.5.2.8 中断屏蔽标志寄存器 (AD16C4Tn_IFM)

中断屏蔽标志寄存器 (AD16C4Tn_IFM)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI		BRKI	TRGI	COMI	CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	R	<p>捕获溢出4中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出CH4中断屏蔽标志产生</p>
CH3OVI	Bit 11	R	<p>捕获溢出3中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH3 中断屏蔽标志产生</p>
CH2OVI	Bit 10	R	<p>捕获溢出2中断标志 参照CH1OVI描述 0: 无中断产生 1: 捕获溢出 CH2 中断屏蔽标志产生</p>
CH1OVI	Bit 9	R	<p>捕获溢出1中断屏蔽标志 当IVS寄存器为1时, CH1I已经被设置又再次收到捕获信号将计数器数值写入CCVAL1寄存器中。此位由硬件设置1, 设置ICR寄存器清除 0: 无中断产生 1: 捕获溢出 CH1 中断屏蔽标志产生</p>
—	Bit 8	—	—
BRKI	Bit 7	R	<p>刹车中断屏蔽标志 当 IVS 寄存器为 1 时, 产生刹车事件时产生。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 刹车中断屏蔽标志产生</p>
TRGI	Bit 6	R	<p>触发中断屏蔽标志 当 IVS 寄存器为 1 时, 产生触发事件时产生(在从模式-门控时, 在任一沿产生, 其他模式则在有效沿时产生)。此位由硬件设置 1, 设置 ICR 寄存器清除 0: 无中断产生 1: 触发中断产生</p>
COMI	Bit 5	R	<p>通信中断屏蔽标志</p>

			<p>当 IVS 寄存器为 1 时,产生通信事件时产生(在捕获/比较预装载控制使能时,更新 CCnEN, CCnNEN 和 CHnMOD 位)。此位由硬件设置 1, 设置 ICR 寄存器清除</p> <p>0: 无中断产生</p> <p>1: 通信中断屏蔽标志产生</p>
CH4I	Bit 4	R	<p>捕获/比较 4 中断屏蔽标志</p> <p>参照 CH1I 描述</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 4 中断屏蔽标志产生</p>
CH3I	Bit 3	R	<p>捕获/比较 3 中断屏蔽标志</p> <p>参照 CH1I 描述</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 3 中断屏蔽标志产生</p>
CH2I	Bit 2	R	<p>捕获/比较 2 中断屏蔽标志</p> <p>参照 CH1I 描述</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 2 中断屏蔽标志产生</p>
CH1I	Bit 1	R	<p>捕获/比较 1 中断屏蔽标志</p> <p>当 IVS 寄存器为 1 时</p> <p>通道 CH1 设置为输出:</p> <p>计数器匹配 CCVAL1 寄存器时设置,在中央对齐模式依据 CMSEL 位在递增、递减、递增/递减时产生。当 CCVAL1 寄存器大于 AR 寄存器时,则在递增边界或递减边界时置起。此位由硬件设置 1, 设置 ICR 寄存器清除</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 1 中断屏蔽标志产生</p> <p>通道 CH1 设置为输入:</p> <p>发生捕获事件,将计数器捕获至 CCVAL1 寄存器中,于 I1 的有效沿产生。此位由硬件设置 1, 设置 ICR 寄存器清除</p> <p>0: 无中断产生</p> <p>1: 捕获/比较 1 中断屏蔽标志产生</p>
UI	Bit 0	R	<p>更新中断屏蔽标志</p> <p>当 CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断</p> <ul style="list-style-type: none"> - 当计数器数值上溢或下溢时 (重复计数器为 0 时时产生更新事件)。 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器重新初始化时。 - 当计数器被触发事件重新初始化时 <p>0: 无中断产生</p> <p>1: 更新中断屏蔽标志产生</p>

18.5.2.9 中断标志清除寄存器 (AD16C4Tn_ICR)

中断标志清除寄存器 (AD16C4Tn_ICR)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			CH4OVI	CH3OVI	CH2OVI	CH1OVI		BRKI	TRGI	COMI	CH4I	CH3I	CH2I	CH1I	UI

—	Bits 31-13	—	—
CH4OVI	Bit 12	C_W1	捕获溢出 4 中断标志清除 0: 写入0无效 1: 捕获溢出CH4中断标志清除
CH3OVI	Bit 11	C_W1	捕获溢出 3 中断标志清除 0: 写入0无效 1: 捕获溢出 CH3 中断标志清除
CH2OVI	Bit 10	C_W1	捕获溢出 2 中断标志清除 0: 写入0无效 1: 捕获溢出 CH2 中断标志清除
CH1OVI	Bit 9	C_W1	捕获溢出 1 中断标志清除 0: 写入0无效 1: 捕获溢出 CH1 中断标志清除
—	Bit 8	—	—
BRKI	Bit 7	C_W1	煞车中断标志清除 0: 写入 0 无效 1: 煞车中断标志清除
TRGI	Bit 6	C_W1	触发中断标志清除 0: 写入 0 无效 1: 触发中断标志清除
COMI	Bit 5	C_W1	通信中断标志清除 0: 写入 0 无效 1: 通信中断标志清除
CH4I	Bit 4	C_W1	捕获/比较 4 中断标志清除 0: 写入 0 无效 1: 捕获 CH4 中断标志清除
CH3I	Bit 3	C_W1	捕获/比较 3 中断标志清除 0: 写入 0 无效 1: 捕获 CH3 中断标志清除
CH2I	Bit 2	C_W1	捕获/比较 2 中断标志清除 0: 写入 0 无效 1: 捕获 CH2 中断标志清除

CH1I	Bit 1	C_W1	捕获/比较 1 中断标志清除 0: 写入 0 无效 1: 捕获 CH1 中断标志清除
UI	Bit 0	C_W1	更新中断标志清除 0: 写入 0 无效 1: 更新中断标志清除

			<p>通道 CH1 设置为输入: 发生捕获事件, 将计数器捕获至 CCVAL1 寄存器中, 于 I1 的有效沿产生, 若使能中断或 DMA, 则产生中断与请求。由软件设置, 于下一个时钟自动清除 0: 写入 0 无效 1: 捕获/比较 1 中断产生</p>
SGU	Bit 0	W1	<p>软件触发更新事件 该位由软件设置, 可由硬件自动清零。 0: 无动作 1: 重新初始化计数器, 更新寄存器。注意, 预分频器也会被清零 (但预分频比不会受到影响)。如果使用中央对齐模式或者 DIRSEL=0 (递增), 则计数器将清零; 否则如果 DIRSEL=1 (递减), 则将使用自动重载入值。</p>

18.5.2.11 捕获/比较模式寄存器 1 (AD16C4Tn_CHMR1)

捕获/比较模式寄存器 1 (AD16C4Tn_CHMR1)																																	
偏移地址: 28 _H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																CH2OCLREN	CH2MOD <2:0>			CH2PEN	CH2FEN	CC2SSEL <1:0>			CH1OCLREN	CH1MOD <2:0>			CH1PEN	CH1FEN	CC1SSEL <1:0>		
																I2FLT <3:0>			I2PRES <1:0>			CC2SSEL <1:0>			I1FLT <3:0>			I1PRES <1:0>			CC1SSEL <1:0>		

输出比较模式

—	Bits 31-16	—	—
CH2OCLREN	Bit 15	R/W	输出比较通道 2 清除使能 参考 CH1OCLREN 描述
CH2MOD	Bits 14-12	R/W	输出比较信道 2 模式 参考 CH1MOD 描述
CH2PEN	Bit 11	R/W	输出比较通道 2 预装载使能 参考 CH1PEN 描述
CH2FEN	Bit 10	R/W	输出比较通道 2 快速使能 参考 CH1FEN 描述
CC2SSEL	Bits 9-8	R/W	捕获/比较通道 2 选择 设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC2EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入, 捕获源为 I2 10: 通道设置为输入, 捕获源为 I1 11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测
CH1OCLREN	Bit 7	R/W	输出比较通道 1 清除使能 0: CH1REF 维持输出 1: CH1REF 根据 CHREF_CLR 的有效位清除
CH1MOD	Bits 6-4	R/W	输出比较信道 1 模式 设置 CH1REF 输出模式, CH1 与 CH1N 由 CH1REF 产生, 而有效位由 CCEP 寄存器的 CC1POL 与 CC1NPPOL 位设置电平 000: 禁止 - 无作用 001: 匹配时设置高电平 - 当计数器匹配

			<p>CCVAL1 寄存器时, CH1REF 设置为 1</p> <p>010: 匹配时设置低电平 - 当计数器匹配 CCVAL1 寄存器时, CH1REF 设置为 0</p> <p>011: 匹配时设置翻转电平 - 当计数器匹配 CCVAL1 寄存器时, CH1REF 设置翻转电平(当前高/低电平翻转成低/高电平)</p> <p>100: 强制低电平 - CH1REF 强制设置低电平</p> <p>101: 强制高电平 - CH1REF 强制设置高电平</p> <p>110: PWM 模式 1 - 上数时, 当计数器小于 CCVAL1 寄存器时, 输出高电平, 其他则输出低电平。下数时, 当计数器大于 CCVAL1 寄存器时输出低电平, 其他则输出高电平</p> <p>111: PWM 模式 2 - 上数时, 当计数器小于 CCVAL1 寄存器时, 输出低电平, 其他则输出高电平。下数时, 当计数器大于 CCVAL1 寄存器时输出高电平, 其他则输出低电平</p>
CH1PEN	Bit 3	R/W	<p>输出比较通道 1 预装载使能</p> <p>设置后在更新事件时, 将设置的寄存器 CCVAL1 数值载入预装载 CCVAL1 寄存器中</p> <p>0: CCVAL1 寄存器预装载禁止</p> <p>1: CCVAL1 寄存器预装载使能</p>
CH1FEN	Bit 2	R/W	<p>输出比较通道 1 快速使能</p> <p>用于加速事件的产生</p> <p>0: CH1 的正常操作依赖于计数器与 CCVAL1 的值, 即使工作于触发器状态。当触发器的输入有一个有效沿时, 激活 CH1 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, CH1 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 CH1 输出间的延时被缩短为 3 个时钟周期。CH1FEN 只在信道被配置成 PWM1 或 PWM2 模式时起作用</p>
CC1SSEL	Bits 1-0	R/W	<p>捕获/比较通道 1 选择</p> <p>设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC1EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入, 捕获源为 I1</p> <p>10: 通道设置为输入, 捕获源为 I2</p> <p>11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测</p>

输入捕获模式

—	Bits 31-16	—	—
I2FLT	Bits 15-12	R/W	输入捕获通道2滤波器 参照I1FLT描述
I2PRES	Bits 11-10	R/W	输入捕获通道 2 预分频器 参照 IC1PRES 描述
CC2SSEL	Bits 9-8	R/W	捕获/比较通道 2 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC2EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I2 10: 通道设置为输入，捕获源为 I1 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测
I1FLT	Bits 7-4	R/W	输入捕获通道 1 滤波器 设置 I1 信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变 0000: 采样频率 fDTS，滤波器禁止 0001: 采样频率 fINT_CLK, N = 2 0010: 采样频率 fINT_CLK, N = 4 0011: 采样频率 fINT_CLK, N = 8 0100: 采样频率 fDTS / 2, N = 6 0101: 采样频率 fDTS / 2, N = 8 0110: 采样频率 fDTS / 4, N = 6 0111: 采样频率 fDTS / 4, N = 8 1000: 采样频率 fDTS / 8, N = 6 1001: 采样频率 fDTS / 8, N = 8 1010: 采样频率 fDTS / 16, N = 5 1011: 采样频率 fDTS / 16, N = 6 1100: 采样频率 fDTS / 16, N = 8 1101: 采样频率 fDTS / 32, N = 5 1110: 采样频率 fDTS / 32, N = 6 1111: 采样频率 fDTS / 32, N = 8
I1PRES	Bits 3-2	R/W	输入捕获通道 1 预分频器 设置 I1 的预分频计数器数值，当清除 CCEP 寄存器的 CC1EN 位，预分频计数器同时被清除 00: 预分频禁止，于每次事件时捕获 01: 每 2 次事件捕获 10: 每 4 次事件捕获 11: 每 8 次事件捕获
CC1SSEL	Bits 1-0	R/W	捕获/比较通道 1 选择 设置通道的输出方向与信号的选择，当 CCEP 寄

			<p>寄存器的 CC1EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入, 捕获源为 I1</p> <p>10: 通道设置为输入, 捕获源为 I2</p> <p>11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测</p>
--	--	--	---

18.5.2.12 捕获/比较模式寄存器 2 (AD16C4Tn_CHMR2)

捕获/比较模式寄存器 2 (AD16C4Tn_CHMR2)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CH4OCLREN	CH4MOD <2:0>		CH4PEN	CH4FEN	CC4SSEL <1:0>		CH3OCLREN	CH3MOD <2:0>		CH3PEN	CH3FEN	CC3SSEL <1:0>			
																I4FLT <3:0>		I4PRES <1:0>		CC4SSEL <1:0>		I3FLT <3:0>		I3PRES <1:0>		CC3SSEL <1:0>					

输出比较模式

—	Bits 31-16	—	—
CH4OCLREN	Bit 15	R/W	输出比较通道 4 清除使能 参考 CH1OCLREN 描述
CH4MOD	Bits 14-12	R/W	输出比较信道 4 模式 参考 CH1MOD 描述
CH4PEN	Bit 11	R/W	输出比较通道 4 预装载使能 参考 CH1PEN 描述
CH4FEN	Bit 10	R/W	输出比较通道 4 快速使能 参考 CH1FEN 描述
CC4SSEL	Bits 9-8	R/W	捕获/比较通道 4 选择 设置通道的输出方向与信号的选择, 当 CCEP 寄存器的 CC4EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入, 捕获源为 I4 10: 通道设置为输入, 捕获源为 I3 11: 通道设置为输入, 捕获源为 ITn 或 I1 的双边沿检测
CH3OCLREN	Bit 7	R/W	输出比较通道 3 清除使能 参考 CH1OCLREN 描述
CH3MOD	Bits 6-4	R/W	输出比较信道 3 模式 参考 CH1OMOD 描述
CH3PEN	Bit 3	R/W	输出比较通道 3 预装载使能 参考 CH1PEN 描述
CH3FEN	Bit 2	R/W	输出比较通道 3 快速使能 参考 CH1FEN 描述
CC3SSEL	Bits 1-0	R/W	捕获/比较通道 3 选择

		<p>设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC3EN 为 0 才可写入。</p> <p>00: 通道设置为输出</p> <p>01: 通道设置为输入，捕获源为 I3</p> <p>10: 通道设置为输入，捕获源为 I4</p> <p>11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测</p>
--	--	--

输入捕获模式

—	Bits 31-16	—	—
I4FLT	Bits 15-12	R/W	输入捕获通道4滤波器 参照I1FLT描述
I4PRES	Bits 11-10	R/W	输入捕获通道 4 预分频器 参照 IC1PRES 描述
CC4SSEL	Bits 9-8	R/W	捕获/比较通道 4 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC4EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I4 10: 通道设置为输入，捕获源为 I3 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测
I3FLT	Bits 7-4	R/W	输入捕获通道 3 滤波器 参考 I1FLT 描述
I3PRES	Bits 3-2	R/W	输入捕获通道 3 预分频器 参考 IC1PRES 描述
CC3SSEL	Bits 1-0	R/W	捕获/比较通道 3 选择 设置通道的输出方向与信号的选择，当 CCEP 寄存器的 CC3EN 为 0 才可写入。 00: 通道设置为输出 01: 通道设置为输入，捕获源为 I3 10: 通道设置为输入，捕获源为 I4 11: 通道设置为输入，捕获源为 ITn 或 I1 的双边沿检测

18.5.2.13 捕获/比较使能极性寄存器 (AD16C4Tn_CCEP)

捕获/比较使能寄存器 (AD16C4Tn_CCEP)																															
偏移地址: 30 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CC4NPOL		CC4POL	CC4EN	CC3NPOL	CC3NEN	CC3POL	CC3EN	CC2NPOL	CC2NEN	CC2POL	CC2EN	CC1NPOL	CC1NEN	CC1POL	CC1EN

—	Bits 31-16	—	—
CC4NPOL	Bit 15	R/W	捕获/比较通道4互补输出有效位极性 参照CC1NPOL描述
—	Bit 14	—	—
CC4POL	Bit 13	R/W	捕获/比较通道4输出有效位极性 参照CC1POL描述
CC4EN	Bit 12	R/W	捕获/比较通道4输出使能 参照CC1EN描述
CC3NPOL	Bit 11	R/W	捕获/比较通道3互补输出有效位极性 参照CC1NPOL描述
CC3NEN	Bit 10	R/W	捕获/比较通道3互补输出使能 参照CC1NEN描述
CC3POL	Bit 9	R/W	捕获/比较通道3输出有效位极性 参照CC1POL描述
CC3EN	Bit 8	R/W	捕获/比较通道3输出使能 参照CC1EN描述
CC2NPOL	Bit 7	R/W	捕获/比较通道2互补输出有效位极性 参照CC1NPOL描述
CC2NEN	Bit 6	R/W	捕获/比较通道2互补输出使能 参照CC1NEN描述
CC2POL	Bit 5	R/W	捕获/比较通道2输出有效位极性 参照CC1POL描述
CC2EN	Bit 4	R/W	捕获/比较通道2输出使能 参照CC1EN描述
CC1NPOL	Bit 3	R/W	捕获/比较通道2互补输出有效位极性 通道CH1设置为输出: 0: CH1N 高电平有效 1: CH1N 低电平有效 通道CH1设置为输入: 该位需和CC1POL一起使用来定义输入边沿的极性。参考CC1POL描述。 注: 对于有互补输出的通道, 该位设置为预载值。

			<p>如果AD16C4Tn_CON2寄存器中的CCPCEN位设置为1，则只有当COM事件发生时，CC1NP有效位才会设置为预载值中新的值。</p> <p>注意：当AD16C4Tn_BDCFG寄存器中的LOCKLVL位被设置为锁定级别2或3，且CC1SSEL=00（信道为输出模式），该位将不可写。</p>
CC1NEN	Bit 2	R/W	<p>捕获/比较通道1互补输出使能</p> <p>0: 关闭 - CH1N 无效。CH1N 电平取决于 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1EN 的功能</p> <p>1: 开启 - CH1N 为对应输出引脚上的输出信号，由 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1EN 决定。</p> <p>注：对于有互补输出的通道，该位设置为预载值。如果AD16C4Tn_CON2寄存器中的CCPCEN位设置为1，则只有当COM事件发生时，CC1NE有效位才会设置为预载值中新的值</p>
CC1POL	Bit 1	R/W	<p>捕获/比较通道1输出有效位极性</p> <p>通道CH1设置为输出：</p> <p>0: CH1 高电平有效</p> <p>1: CH1 低电平有效</p> <p>通道CC1设置为输入：</p> <p>CC1NPOL/CC1POL 位选择触发边沿或捕获模式下I1和I2的极性</p> <p>00: 非反相/上升沿</p> <p>01: 反相/下降沿</p> <p>10: 保留</p> <p>11: 非反相/上升沿+下降沿</p> <p>注：对于有互补输出的通道，该位设置为预载值。如果AD16C4Tn_CON2寄存器中的CCPCEN位设置为1，则只有当COM事件发生时，CC1POL有效位才会设置为预载值中新的值。</p> <p>注：当AD16C4Tn_BDCFG寄存器中的LOCKLVL位被设置为锁定级别2或3，且CC1SSEL=00（信道为输出模式），该位将不可写。</p>
CC1EN	Bit 0	R/W	<p>捕获/比较通道1输出使能</p> <p>通道CH1设置为输出：</p> <p>0: 关闭 - CH1 无效。CH1 电平取决于 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1NEN 的功能</p> <p>1: 开启 - CH1 为对应输出引脚上的输出信号，由 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N</p>

		<p>和 CC1NEN 决定</p> <p>通道 CH1 设置为输入:</p> <p>0: 捕获禁止</p> <p>1: 捕获使能</p> <p>注: 对于有互补输出的通道, 该位设置为预载值。 如果 AD16C4Tn_CON2 寄存器中的 CCPCEN 位设置为 1, 则只有当 COM 事件发生时, CC1EN 有效位才会设置为预载值中新的值。</p>
--	--	---

18.5.2.14 计数寄存器 (AD16C4Tn_COUNT)

计数寄存器 (AD16C4Tn_COUNT)																																													
偏移地址: 34 _H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																CNTV<15:0>																													

—	Bits 31-16	—	—
CNTV	Bits 15-0	R/W	计数器数值

18.5.2.15 预分频寄存器 (AD16C4Tn_PRES)

时钟预分频寄存器 (AD16C4Tn_PRES)																																													
偏移地址: 38 _H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																PSCV<15:0>																													

—	Bits 31-16	—	—
PSCV	Bits 15-0	R/W	预分频数值 当计数器时钟频率等于fINT_CLK/(PSCV<15:0> + 1)时计数器递增或递减。在更新事件产生时, 将PSCV数值被载入预装载寄存器中

18.5.2.16 自动重载寄存器 (AD16C4Tn_AR)

自动重载寄存器 (AD16C4Tn_AR)																																														
偏移地址: 3C _H																																														
复位值: 00000000_00000000_11111111_11111111 _B																																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																ARV<15:0>																														

—	Bits 31-16	—	—
ARV	Bits 15-0	R/W	自动装载数值 设置计数器的递增边界，设置数值为 0 时计数器停止计数

18.5.2.17 重复计数寄存器 (AD16C4Tn_REPAR)

重复计数寄存器 (AD16C4Tn_REPAR)																															
偏移地址: 40H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												REPV <7:0>			

—	Bits 31-8	—	—
REPV	Bits 7-0	R/W	<p>重复计数数值</p> <p>当预载寄存器使能, 该位允许用户设置比较寄存器的更新率 (例如: 预载到有效寄存器的周期性传输), 同样也可以设置更新中断生成率。每次当 REPV_CNT 的相关递减计数器递减至 0, 会产生更新事件, 会从 REPV 值重新计数。因为只有当发生重复更新事件时, REPV_CNT 才会重新载入 REPV 值, 所以只有在发生下一次重复更新事件时, 写入 AD16C4Tn_REPAR 寄存器的值才会生效。</p> <p>即, 在 PWM 模式下, (REPV+1) 相当于:</p> <ul style="list-style-type: none"> - 在边沿对齐模式下, (REPV+1) 对应的是 PWM 的周期数 - 在中央对齐模式下, (REPV+1) 对应的是 1/2 PWM 的周期数

18.5.2.18 通道捕获/比较寄存器 1 (AD16C4Tn_CCVAL1)

通道捕获/比较寄存器 1 (AD16C4Tn_CCVAL1)																																													
偏移地址: 44H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																CCRV1<15:0> ^Δ																													

—	Bits 31-16	—	—
CCRV1	Bits 15-0	R/W	<p>捕获/比较数值 1 信道 CHn 配置为输出: CCRVn 中的值将被载入实际的捕获/比较寄存器中 (预载值)。 如果在AD16C4Tn_CHMRn寄存器中的预载功能没有选中, CCRVn中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与AD16C4Tn_COUNT中的值进行比较, 并在CHn上输出。</p> <p>信道CHn配置为输入: CCRVn为由上一个输入捕获事件 (In) 传输的计数值。</p>

18.5.2.19 通道捕获/比较寄存器 2 (AD16C4Tn_CCVAL2)

通道捕获/比较寄存器 2 (AD16C4Tn_CCVAL2)																																													
偏移地址: 48H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																CCRV2<15:0> ^Δ																													

—	Bits 31-16	—	—
CCRV2	Bits 15-0	R/W	<p>捕获/比较数值2 参照CCRV1描述</p>

18.5.2.20 通道捕获/比较寄存器 3 (AD16C4Tn_CCVAL3)

通道捕获/比较寄存器 3 (AD16C4Tn_CCVAL3)																															
偏移地址: 4C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CCRV3<15:0>															

—	Bits 31-16	—	—
CCRV3	Bits 15-0	R/W	捕获/比较数值3 参照CCRV1描述

18.5.2.21 通道捕获/比较寄存器 4 (AD16C4Tn_CCVAL4)

通道捕获/比较寄存器 4 (AD16C4Tn_CCVAL4)																															
偏移地址: 50 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CCRV4<15:0>															

—	Bits 31-16	—	—
CCRV4	Bits 15-0	R/W	捕获/比较数值4 参照CCRV1描述

18.5.2.22 刹车和死区配置寄存器 (AD16C4Tn_BDCFG)

刹车和死区配置寄存器 (AD16C4Tn_BDCFG)																																				
偏移地址: 54H																																				
复位值: 00000000_00000000_00000000_00000000 _B																																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
																GOEN	AOEN	BRKP	BRKEN	OFFSSR	OFFSSI	LOCKLVL<1:0>				DT<7:0>										

—	Bits 31-16	—	—
GOEN	Bit 15	R/W	<p>通道主要输出使能</p> <p>一旦刹车输入有效, 该位会由硬件异步清零。该位可由软件置1或自动置1, 取决于AOEN位。该位仅作用于配置为输出的信道。</p> <p>0: CHn和CHnN输出禁止或强制为空闲状态。</p> <p>1: 如果CHn和CHnN各自的使能位都置1 (AD16C4Tn_CCEP寄存器中的CCnEN, CCnNEN), 则CH和CHnN输出使能。</p>
AOEN	Bit14	R/W	<p>通道自动输出使能</p> <p>在发生更新事件时, 将GOEN位置起</p> <p>0: GOEN仅可由软件置位</p> <p>1: 在下一个更新事件发生时 (如果刹车输入无效), GOEN可由软件或自动置位。</p> <p>注意: 当AD16C4Tn_BDCFG寄存器中的LOCKLVL位已被设置为锁定级别1, 则该位不可更改。</p>
BRKP	Bit 13	R/W	<p>选择通道刹车极性</p> <p>0: 刹车输入BRKP为低有效</p> <p>1: 刹车输入BRKP为高有效</p> <p>注意: 当AD16C4Tn_BDCFG寄存器中的LOCKLVL位已被设置为锁定级别1, 则该位不可更改</p> <p>注意: 任何对该位的写操作都要延时1APB时钟周期后才变为有效。</p>
BRKEN	Bit 12	R/W	<p>使能刹车</p> <p>0: 刹车输入 (BRKP和CCS时钟失效事件) 禁止</p> <p>1: 刹车输入 (BRKP和CCS时钟失效事件) 使能</p> <p>注意: 当AD16C4Tn_BDCFG寄存器中的LOCKLVL位已被设置为锁定级别1, 则该位不可</p>

			更改 注意：任何对该位的写操作都要延时 1 APB 时钟周期后才变为有效。
OFFSSR	Bit 11	R/W	运行模式下关闭状态选择 当 GOEN 为 1 且通道设置为输出时。当定时器禁止时，CCnEN 为 1 或 CCnNNE 为 1 时，则 CHn/CHnN 输出 CCxPOL/CCxNPPOL 0: 当定时器禁止时，输出禁止 (CHn/CHnN 使能输出信号=0)。 1: 当定时器禁止时，输出 CCxPOL/CCxNPPOL 注：当 AD16C4Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 2，则该位不可更改。
OFFSSI	Bit 10	R/W	空闲模式下关闭选择 当 GOEN 为 0 且通道设置为输出时。当定时器禁止时，CCnEN 为 1 或 CCnNNE 为 1 时，CHn/CHnN 输出空闲状态 0: 当定时器禁止时，输出禁止 (CHn/CHnN 使能输出信号=0)。 1: 当定时器禁止时，输出空闲状态 (OISSx/OISSxN) 注：当 AD16C4Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 2，则该位不可更改。
LOCKLVL	Bits 9-8	R/W	锁定级别配置 针对软件错误，该位提供写保护。 00: 锁定关闭—不提供写保护 01: 锁定级别 1 = BDCFG 寄存器中的 DT，CON2 寄存器中的 OISSx 和 OISSxN，和 BDCFG 寄存器中的 BRKEN/BRKP/AOEN 不再可写。 10: 锁定级别 2 = 锁定级别 1 + CC 极性位 (CCEP 寄存器中的 CCnPOL/CCnNPOL，只要相关信道由 CCnSSEL 配置为输出) 以及 OFFSSR 和 OFFSSI 都不再可写。 11: 锁定级别 3 = 锁定级别 2 + CC 控制位 (CHMRn 寄存器中的 CHnMOD 和 CHnPEN，只要相关信道由 CCnSSEL 配置为输出) 都不再可写。 注：锁定配置为仅在复位后可写。一旦 BDCFG 已写，其设置内容在下一个复位前都处于冻结状态。
DT	Bits 7-0	R/W	死区延时 设置值该位定义了互补输出之间插入的死区时间。DT 对应的就是该时间段。

		<p>DT[7 : 5]=0xx => DT=DT[7:0]x t_{dtg} , 式中 t_{dtg}=t_{DTS}。</p> <p>DT[7 : 5]=10x => DT= (64+DT[5:0]) x t_{dtg} , 式中 t_{dtg}=2x t_{DTS}。</p> <p>DT[7 : 5]=110=> DT= (32+DT[4:0]) x t_{dtg} , 式中 t_{dtg}=8x t_{DTS}。</p> <p>DT[7 : 5]=111 => DT= (32+DT[4: 0]) x t_{dtg} , 式中 t_{dtg}=16x t_{DTS}。</p> <p>注意：当 AD16C4Tn_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 2 或 3, 则该位不可更改</p>
--	--	--

18.5.2.23 DMA 事件使能寄存器 (AD16C4Tn_DMAEN)

DMA 事件使能寄存器 (AD16C4Tn_DMAEN)																																
偏移地址: 58H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																										TRGIDE	COMDE	CH4DE	CH3DE	CH2DE	CH1DE	UDE

—	Bits 31-7	—	—
TRGIDE	Bit 6	R/W	触发DMA请求使能 0: 触发DMA请求禁止 1: 触发DMA请求使能
COMDE	Bit 5	R/W	通信 DMA 请求使能 0: 通信 DMA 请求禁止 1: 通信 DMA 请求使能
CH4DE	Bit 4	R/W	通道捕获/比较 4 DMA 请求使能 0: 捕获/比较 4 DMA 请求禁止 1: 捕获/比较 4 DMA 请求使能
CH3DE	Bit 3	R/W	通道捕获/比较3 DMA请求使能 0: 捕获/比较 3 DMA请求禁止 1: 捕获/比较 3 DMA请求使能
CH2DE	Bit 2	R/W	通道捕获/比较 2 DMA 请求使能 0: 捕获/比较 2 DMA 请求禁止 1: 捕获/比较 2 DMA 请求使能
CH1DE	Bit 1	R/W	通道捕获/比较 1 DMA 请求使能 0: 捕获/比较 1 DMA 请求禁止 1: 捕获/比较 1 DMA 请求使能
UDE	Bit 0	R/W	更新 DMA 请求使能 0: 更新 DMA 请求禁止 1: 更新 DMA 请求使能

第19章 基本定时器 (BS16T)

19.1 概述

基本定时器(BS16Tn)包含一个 16 位自动重载计数器,该计数器由可配置的预分频器驱动。

通过使用定时器的分频器和 APB 时钟控制器的预分频功能,可对脉冲长度和波形周期进行数微妙到几毫秒的调整。

19.2 特性

- ◆ 16 位自动加载递增计数器
- ◆ 16 位可编程预分频器,可对计数器工作时钟进行 1 到 65536 的任意分频(运行中也可以)
- ◆ 计数上溢更新事件产生中断/DMA 请求

19.3 结构图

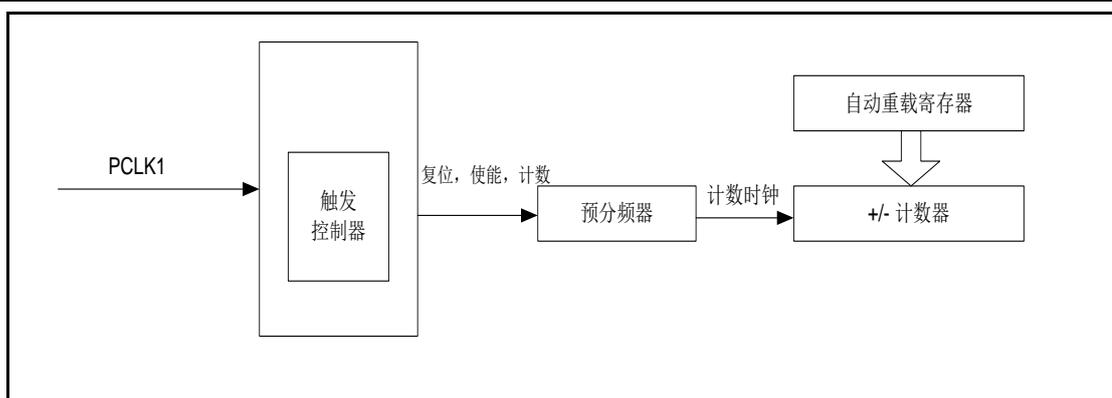


图 19-1 基本定时器结构框图

19.4 功能描述

19.4.1 预分频器

定时器包含一个 16-bit 的计数器（**BS16Tn_COUNT**），计数时钟由预分频寄存器（**BS16Tn_PRES**）进行分频。计数周期由自动重载计数器（**BS16Tn_AR**）设定。

自动重载寄存器（**BS16Tn_AR**）是一个可缓存的寄存器。当 **BS16Tn_CON1** 寄存器的 **ARPEN** 位复位时，**BS16Tn_AR** 寄存器重载功能失效，**BS16Tn_AR** 就是有效寄存器；**ARPEN** 置位时，**BS16Tn_AR** 寄存器具有重载功能，产生更新事件（**UEV**）时，加载值（**BS16Tn_AR** 寄存器值）更新到影子寄存器。

当 **BS16Tn_CON1** 寄存器中 **DISUE** 位为 0 时，计数器计数上溢（或递减下溢）时会产生更新事件（**UEV**）。同样，软件方式也可产生更新事件。**BS16Tn_CON1** 寄存器的 **CNTEN** 置位时，计数器开始计数。

注：计数器在 **CNTEN** 位置位 1 个时钟周期后开始计数。

预分频器可对定时器工作时钟进行 **BS16Tn_PRES** 寄存器值+1 次分频。由于 **BS16Tn_PRES** 是一个可重载寄存器，因此，定时器工作时可以对该寄存器进行修改，修改值在下次更新事件（**UEV**）后有效。

下图给出了定时器运行过程中改变预分频值时计数器的计数情况。

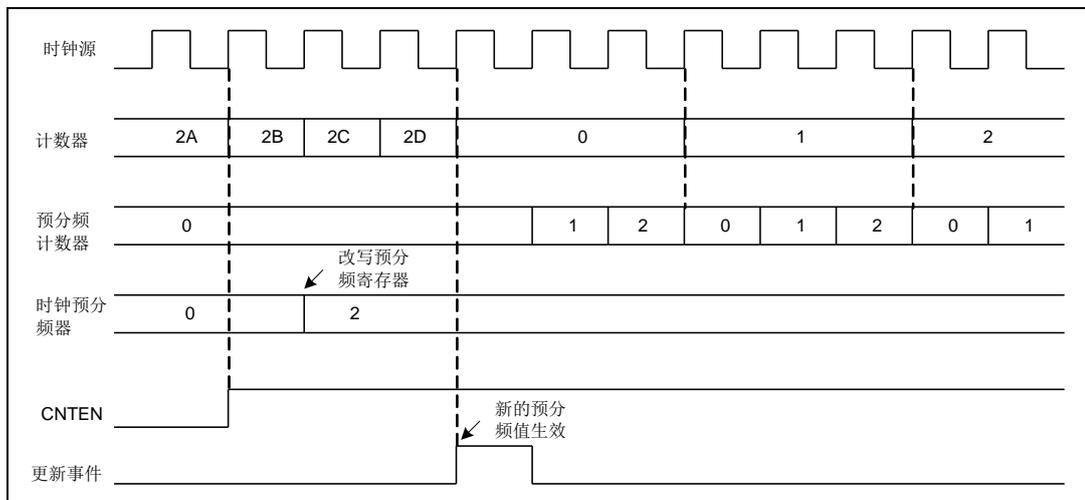


图 19-2 预分频值计数时序图

19.4.2 时钟源

计数时钟由内部时钟源（PCLK1）提供。

CNTEN 位（BS16Tn_CON1 寄存器）与 SGU 位（BS16Tn_SGE 寄存器）为实际控制位，这两个位只能软件修改（SGU 位除外，仍硬件自动清除）。一旦 CNTEN 位被写为'1'，预分频器就由内部 INT_CLK 提供时钟。

下图给出了通常模式下控制电路和递增计数的情况，没有分频。

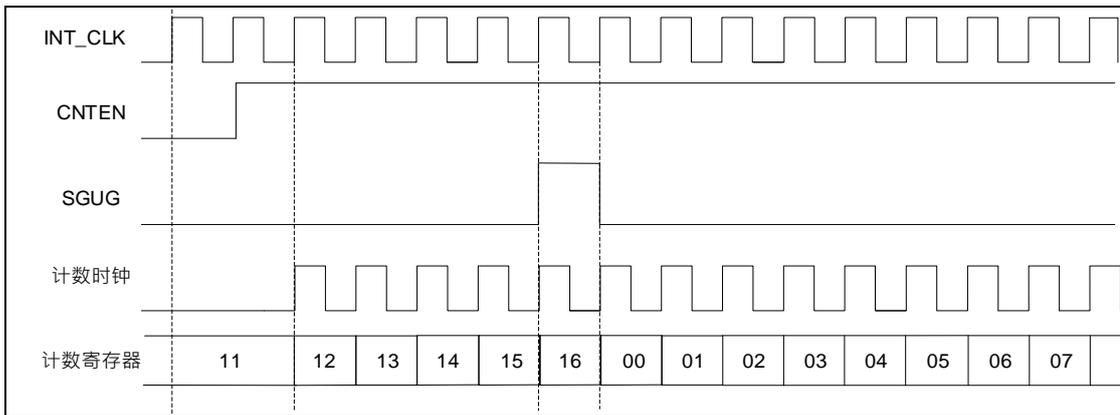


图 19-3 采用内部时钟计数

19.4.3 递增计数模式

在递增计数模式中，计数器由 0 开始计数至自动重载值（BS16Tn_AR 寄存器中的值），然后从 0 开始重新计数并产生一个计数溢出事件。

软件置位 BS16Tn_CON1 寄存器中的 DISUE 位可关闭更新事件（UEV）的产生。更新事件（UEV）关闭，可避免向预载寄存器写新值的过程中更新影子寄存器。这种情况下，DISUE 位在写'0'之前都不会产生更新事件。正常产生更新事件后，计数器和预载计数器都是从 0 重新开始（但预分频值没有改变）。此外，若置位 BS16Tn_CON1 寄存器中的 UERSEL 位（更新请求选择），置位 SGU 位时会产生一次更新事件（UEV），但 UI 标志位不会置位（因此，不会触发中断或 DMA 请求）。

当更新事件发生时，所有寄存器都会被更新且更新标志位（BS16Tn_RIF 寄存器中的 UI 位）置位（取决于 UERSEL 位）：

- ◇ 更新 BS16Tn_AR 寄存器的值到影子寄存器
- ◇ 更新 BS16Tn_PRES 寄存器的值到影子寄存器

下图为 BS16Tn_AR = 0x16，预分频设为 2 分频时的计数器时序。

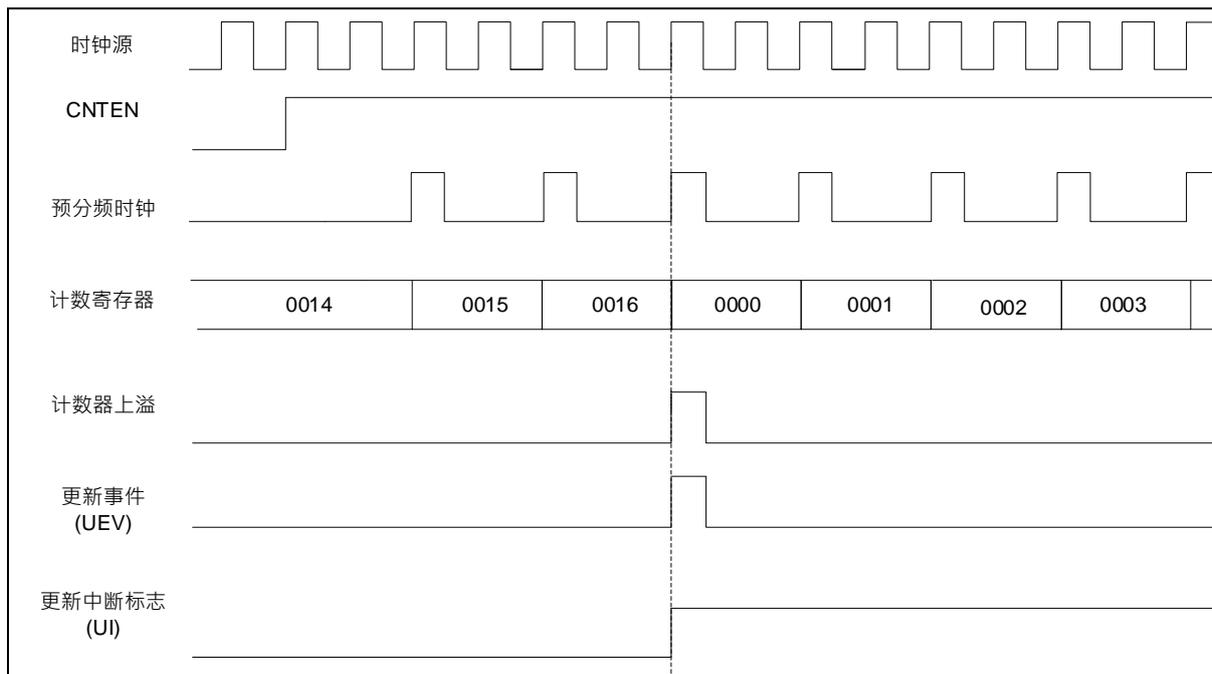


图 19-4 计数器递增计数时序图

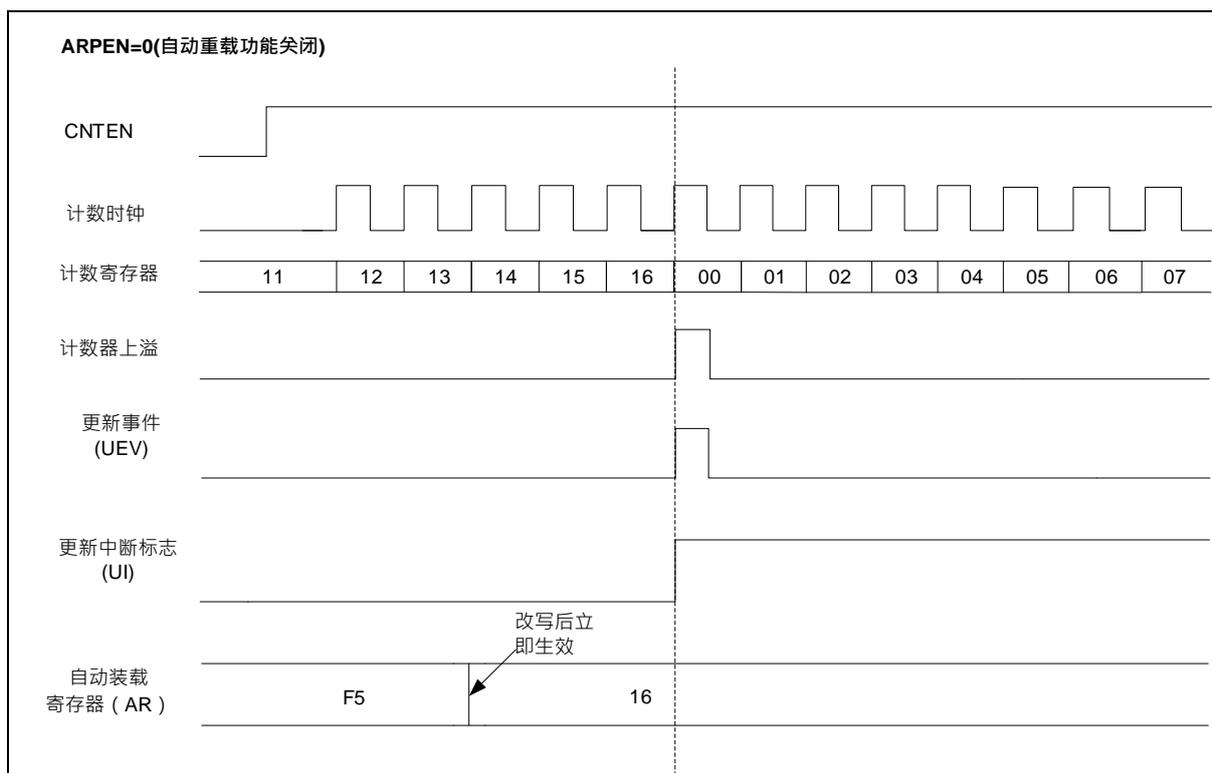


图 19-5 当 ARPEN=0 时计数器时序图

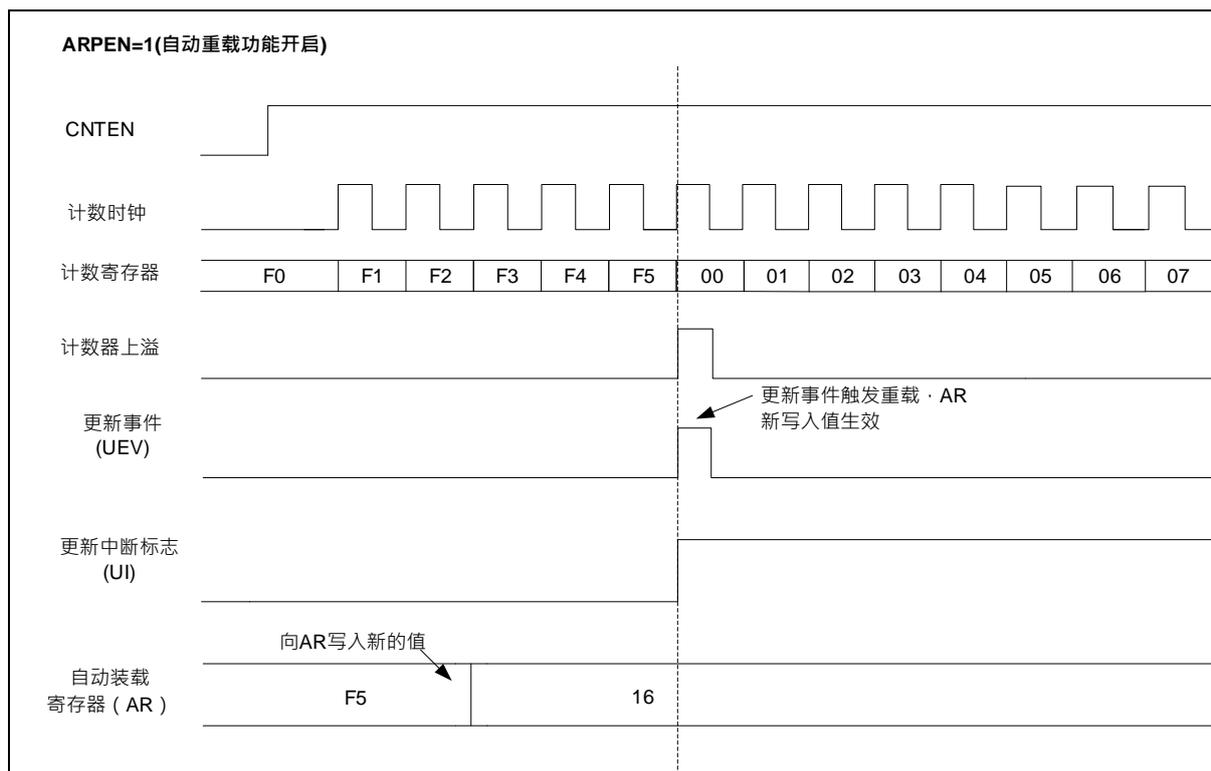


图 19-6 当 ARPEN=1 时计数器时序图

19.4.4 调试模式

当微控制器进入调试模式 (Cortex™-M0 内核终止), 计数器可被设定停止计数。

19.5 特殊功能寄存器

19.5.1 寄存器列表

寄存器列表			
名称	偏移地址	类型	描述
BS16Tn_CON1	00 _H	R/W	控制寄存器 1
BS16Tn_CON2	04 _H	R/W	控制寄存器 2
BS16Tn_IER	0C _H	W1	中断使能寄存器
BS16Tn_IDR	10 _H	W1	中断禁止寄存器
BS16Tn_IVS	14 _H	R	中断有效状态寄存器
BS16Tn_RIF	18 _H	R	原始中断标志寄存器
BS16Tn_IFM	1C _H	R	中断屏蔽标志寄存器
BS16Tn_ICR	20 _H	C_W1	中断标志清除寄存器
BS16Tn_SGE	24 _H	W1	软件生成事件寄存器
BS16Tn_COUNT	34 _H	R/W	计数器
BS16Tn_PRES	38 _H	R/W	时钟预分频器
BS16Tn_AR	3C _H	R/W	自动重装载寄存器
BS16Tn_DMAEN	58 _H	R/W	DMA 事件使能寄存器

19.5.2 寄存器描述

19.5.2.1 控制寄存器 1 (BS16Tn_CON1)

控制寄存器 1 (BS16Tn_CON1)																																
偏移地址: 00H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																								ARPEN					SPMEN	UERSEL	DISUE	CNTEN

—	Bits 31-8	—	—
ARPEN	Bit 7	R/W	<p>自动重载使能 发生更新事件时，将设定的值载入至缓冲寄存器中 0: BS16Tn_AR 寄存器未缓冲 1: BS16Tn_AR 寄存器被装入缓冲器</p>
—	Bits 6-4	—	—
SPMEN	Bit 3	R/W	<p>单脉冲模式 0: 单脉冲模式禁止，计数器不停止 1: 单脉冲模式使能，计数器在发生下一次更新事件时，清除 CNTEN 位，计数器停止</p>
UERSEL	Bit 2	R/W	<p>更新事件请求来源选择 设置更新事件(UEV)的来源 0: 若使能 UI 中断或 DMA 请求，下列事件中产生更新事件请求 - 计数器溢出 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 1: 若使能 UI 中断或 DMA 请求，只在计数器溢出/下溢时产生更新事件请求</p>
DISUE	Bit 1	R/W	<p>更新事件禁止 设置更新事件(UEV)的产生 0: 更新事件(UEV) 使能，下列事件中产生更新事件请求 - 计数器溢出 - 设置 SGE 寄存器的 SGU 位 - 从模式中产生的复位请求 缓冲寄存器载入预装载值 1: 更新事件(UEV) 禁止，不产生更新事件请求，AR、PRES 寄存器保持数值 禁止更新事件时，设置 SGE 寄存器的 SGU 位或</p>

			从模式中产生的复位请求，计数器和预分频器仍会被重新初始化
CNTEN	Bit 0	R/W	<p>计数器使能</p> <p>使能计数器后，在外部时钟模式、门控模式和编码模式才能运作。触发模式则可以由硬件设置 CNTEN 位</p> <p>0: 计数器禁止</p> <p>1: 计数器使能</p>

19.5.2.2 控制寄存器 2 (BS16Tn_CON2)

控制寄存器 2 (BS16Tn_CON2)																																
偏移地址: 04H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

—	Bits 31-7	—	—
MMSEL	Bits 6-4	R/W	<p>主模式选择 设置在主模式下发送到从定时器的同步信号 (TRGOUT)与 ADC 输入 000: 复位 - 设置 SGE 寄存器信号用于同步触发输出(TRGOUT)。从模式的复位触发产生的复位信号(TRGOUT)则与实际信号相差一个时钟 001: 使能 - 计数器的使能信号 CNTEN 用于同步触发输出(TRGOUT), 可用于同步使能数个定时器。门控模式下, 是使用 CON1 寄存器的 CNTEN 位与触发输入信号逻辑产生。 010: 更新事件 - 更新事件被用于同步触发输出 (TRGOUT)。一个主定时器的更新事件可当作从定时器的预分频器时钟 011: 保留 100: 保留 101: 保留 110: 保留 111: 保留</p>
—	Bits 3-0	—	—

19.5.2.3 中断使能寄存器 (BS16Tn_IER)

中断使能寄存器 (BS16Tn_IER)																																
偏移地址: 0C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																																⊔

—	Bits 31-1	—	—
UI	Bit 0	W1	更新中断使能 0: 写入 0 无效 1: 更新中断使能

19.5.2.4 中断禁止寄存器 (BS16Tn_IDR)

中断禁止寄存器 (BS16Tn_IDR)																																
偏移地址: 10 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																																⊔

—	Bits 31-1	—	—
UI	Bit 0	W1	更新中断禁止 0: 写入 0 无效 1: 更新中断禁止

19.5.2.5 中断有效状态寄存器 (BS16Tn_IVS)

中断有效状态寄存器 (BS16Tn_IVS)																																	
偏移地址: 14H																																	
复位值: 00000000_00000000_00000000_00000000b																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		

—	Bits 31-1	—	—
UI	Bit 0	R	更新中断有效位 设置 IER 和 IDR 寄存器使能或禁止 0: 更新中断禁止 1: 更新中断使能

19.5.2.6 原始中断标志寄存器 (BS16Tn_RIF)

原始中断标志寄存器 (BS16Tn_RIF)																																
偏移地址: 18H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

—	Bits 31-1	—	—
UI	Bit 0	R	更新原始中断标志 当 CON1 寄存器的 UERSEL、DISUE 为 0，以下三件事件触发中断 - 当计数器数值上溢 - 设置 SGE 寄存器的 SGU 位时产生更新事件，通过软件对计数器重新初始化时。 0: 无中断产生 1: 更新原始中断产生

19.5.2.7 中断屏蔽标志寄存器 (BS16Tn_IFM)

中断屏蔽标志寄存器 (BS16Tn_IFM)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

—	Bits 31-1	—	—
UI	Bit 0	R	<p>更新中断屏蔽标志 当 IVS 寄存器为 1 时, CON1 寄存器的 UERSEL、DISUE 为 0, 以下三件事件触发中断</p> <ul style="list-style-type: none"> - 当计数器数值上溢 - 设置 SGE 寄存器的 SGU 位时产生更新事件, 通过软件对计数器 CNT 重新初始化时。 <p>0: 无中断产生 1: 更新中断产生</p>

19.5.2.8 中断标志清除寄存器 (BS16Tn_ICR)

中断标志清除寄存器 (BS16Tn_ICR)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

—	Bits 31-1	—	—
UI	Bit 0	C_W1	<p>更新中断清除 0: 写入 0 无效 1: 更新中断清除</p>

19.5.2.9 软件生成事件生寄存器 (BS16Tn_SGE)

软件生成事件生寄存器 (BS16Tn_SGE)																																
偏移地址: 24H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																																SGU

—	Bits 31-1	—	—
SGU	Bit 0	W1	<p>软件触发更新事件 该位由软件设置, 可由硬件自动清零。 0: 无动作 1: 重新初始化计数器, 更新寄存器。注意, 预分频器也会被清零 (但预分频比不会受到影响)。</p>

19.5.2.10 计数寄存器 (BS16Tn_COUNT)

计数寄存器 (BS16Tn_COUNT)																																													
偏移地址: 34H																																													
复位值: 00000000_00000000_00000000_00000000b																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																CNTV<15:0>																													

—	Bits 31-16	—	—
CNTV	Bits 15-0	R/W	计数器数值

19.5.2.11 预分频寄存器 (BS16Tn_PRES)

时钟预分频寄存器 (BS16Tn_PRES)																															
------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

偏移地址: 38H																															
复位值: 00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PSCV<15:0>															

—	Bits 31-16	—	—
PSCV	Bits 15-0	R/W	<p>预分频数值</p> <p>当计数器时钟频率等于fINT_CLK/(PSCV<15:0> + 1)时计数器递增或递减。在更新事件产生时, 将PSCV数值被载入预装载寄存器中</p>

19.5.2.12 自动重载寄存器 (BS16Tn_AR)

自动重载寄存器 (BS16Tn_AR)																																														
偏移地址: 3C _H																																														
复位值: 00000000_00000000_11111111_11111111 _B																																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																ARV<15:0>																														

—	Bits 31-16	—	—
ARV	Bits 15-0	R/W	自动装载数值 设置计数器的递增边界，设置数值为 0 时计数器停止计数

19.5.2.13 DMA 事件使能寄存器 (BS16Tn_DMAEN)

DMA 事件使能寄存器 (BS16Tn_DMAEN)																																	
偏移地址: 58 _H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																																	UDE

—	Bits 31-1	—	—
UDE	Bit 0	R/W	更新 DMA 请求使能 0: 更新 DMA 请求禁止 1: 更新 DMA 请求使能

第20章 独立看门狗 (IWDT)

20.1 概述

独立看门狗 IWDT 可用于检测软件和硬件异常引起的故障，如主时钟停振、用户程序异常无法喂狗等；当计数器达到给定的超时值时，将触发系统复位。

独立看门狗 IWDT，当硬件使能时，IWDT 时钟强制为独立的 32.768 kHz LRC 时钟，且用户无法通过软件关闭 IWDT。

独立看门狗 IWDT 最适合于独立于主程序之外，并且对时间精度要求较低场合。

20.2 特性

- ◆ 自由运行的递减计数器
- ◆ 由一个独立的 RC 振荡器驱动(在 Sleep 状态下仍可操作)，复位条件如下：
 - ◇ 当向下递减计数器的值达到 0 时产生复位请求。
 - ◇ 当向下递减计数器的值处于窗口值之外时，被重新加载就会导致复位请求。

20.3 结构图

图 20-1 描述了独立看门狗模块的功能框图。

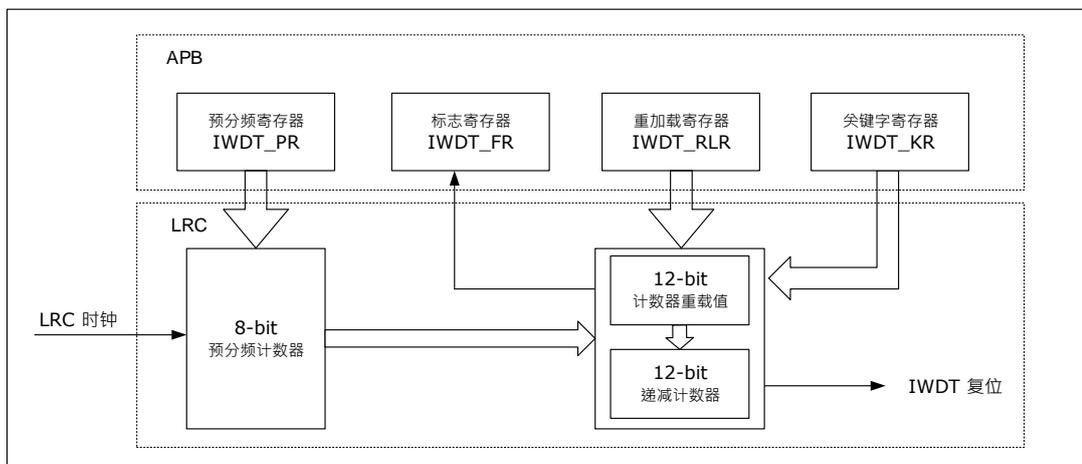


图 20-1 独立看门狗框图

当向独立看门狗的关键字寄存器写入启动指令 0x0000CCCC 的时候，看门狗计数器开始由复位值 0xFFFF 向下计数。

当计数值达到 0x000 的时候由独立看门狗发出复位信号。

启动后将关键字 0x0000AAAA 写到 IWDT_KR 寄存器中，都使得 IWDT_RLR 寄存器中的值被重新加载到看门狗计数器中，从而阻止即将发生的复位动作。

20.4 功能描述

20.4.1 窗口选项

IWDT 也能够工作在窗口看门狗模式下，只要在 **IWDT_WINR** 寄存器中设置适当的值即可。

如果重加载操作执行的同时，看门狗计数器的值超出了窗口寄存器(**IWDT_WINR**)中存储的值，也会引起复位操作。

IWDT_WINR 的默认值是 0x00000FFF,所以如果没有改写它，那么窗口选项默认是关闭的。

当窗口选项使能时配置 IWDT

- ◆ 将 0x0000CCCC 写到 **IWDT_KR** 寄存器，使能 IWDT。
- ◆ 等待状态寄存器 **IWDT_FR.BUSY** 的值更新为 0。
- ◆ 向 **IWDT_KR** 寄存器写 0x00005555 打开寄存器访问许可。
- ◆ 向 **IWDT_PR** 写 0~7 的值，以配置 IWDT 的预分频器。
- ◆ 等待状态寄存器 **IWDT_FR.BUSY** 的值更新为 0。
- ◆ 配置重加载寄存器(**IWDT_RLR**)。
- ◆ 等待状态寄存器 **IWDT_FR.BUSY** 的值更新为 0。
- ◆ 将 **IWDT_RLR** 的值刷新到看门狗定时器(**IWDT_KR=0x0000AAAA**)。
- ◆ 向 **IWDT_KR** 寄存器写 0x00005555 打开寄存器访问许可。
- ◆ 配置窗口寄存器 **IWDT_WINR**。

当窗口选项被禁止时配置 IWDT

- ◆ 将 0x0000CCCC 写到 **IWDT_KR** 寄存器，使能 IWWDWT。
- ◆ 等待状态寄存器 **IWDT_FR.BUSY** 的值更新为 0。
- ◆ 向 **IWDT_KR** 寄存器写 0x00005555 打开寄存器访问许可。
- ◆ 向 **IWDT_PR** 写 0~7 的值，以配置 IWDT 的预分频器。
- ◆ 等待状态寄存器 **IWDT_FR.BUSY** 的值更新为 0。
- ◆ 配置重加载寄存器(**IWDT_RLR**)。
- ◆ 等待状态寄存器 **IWDT_FR.BUSY** 的值更新为 0。
- ◆ 将 **IWDT_RLR** 的值刷新到看门狗定时器(**IWDT_KR=0x0000AAAA**)。

20.4.2 Sleep 模式下的行为

使能 IWDT 后，则无法停止 IWDT

20.4.2.1 寄存器访问保护

默认条件下，对 **IWDT_PR**、**IWDT_RLR** 和 **IWDT_WINR** 的写访问操作都是受保护的。想要改变这一点，必须先向 **IWDT_KR** 写入 0x00005555 解锁码。如果写入别的值，将会打破这个顺序，使得对寄存器的访问保护重新生效。这意味着在做重加载操作的时候（向该寄存器写入 0x0000AAAA）就属于这种情况。

可以通过一个状态寄存器观察预分频器的更新、看门狗计数器的重加载或窗口值的重加载。

20.4.3 调试模式

当微控制器进入调试模式时(内核被暂停)，看门狗计数器可以继续运行，也可以被停止。

20.5 特殊功能寄存器

20.5.1 寄存器列表

IWDT 寄存器列表			
名称	偏移地址	类型	描述
IWDT_KR	000 _H	W	IWDT 关键字寄存器
IWDT_PR	004 _H	R/W	IWDT 预分频寄存器
IWDT_RLR	008 _H	R/W	IWDT 重加载寄存器
IWDT_FR	00C _H	R	IWDT 标志寄存器
IWDT_WINR	010 _H	R/W	IWDT 窗口寄存器

20.5.2 寄存器描述

20.5.2.1 IWDT 关键字寄存器 (IWDT_KR)

IWDT 关键字寄存器(IWDT_KR)																																															
偏移地址: 00H																																															
复位值: 00000000_00000000_00000000_00000000B																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
																KEY<15:0>																															

—	Bits 31-16	—	—
KEY	Bits 15-0	W	<p>关键值（只写，读的话会是 0x0000）</p> <p>这些位必须周期性的由软件写入 0xAAAA，否则当计数器向下计数到 0 的时候会产生硬件复位请求。</p> <p>写入 0x5555 会使能对 IWDT_PR, IWDT_RLR 和 IWDT_WINR 三个寄存器的访问许可。</p> <p>写入 0xCCCC 启动看门狗</p> <p>注：关键值只能在 IWDT_FR 寄存器中的 BUSY 位为零时更新。</p>

20.5.2.2 IWDT 预分频寄存器 (IWDT_PR)

IWDT 预分频寄存器(IWDT_PR)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												PR<2:0>			

—	Bits 31-3	—	—
PR	Bits 2-0	R/W	<p>预分频器</p> <p>这些位平时处于写保护状态，由软件写入，用来选择对输入时钟的预分频系数。为了能够改变预分频器的分频系数，IWDT_FR 寄存器中的 BUSY 位必须先清零。</p> <p>000: 4分频 001: 8分频 010: 16分频 011: 32分频 100: 64分频 101: 128分频 110: 256分频 111: 256分频</p> <p>注:如果正在对该寄存器执行写操作，则读取的值可能不是最新的/ 有效的。因此，从这个地方读数据的时候要保证IWDT_FR寄存器中的BUSY位为 0 才行。</p>

20.5.2.3 IWDT 重加载寄存器 (IWDT_RLR)

IWDT 重加载寄存器(IWDT_RLR)																															
偏移地址: 08H																															
复位值: 00000000_00000000_00001111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												RL<11:0>			

—	Bits 31-12	—	—
RL	Bits 11-0	R/W	<p>看门狗计数器重载值</p> <p>这些位平时处于写保护状态，这个值是由软件来设置的，并且每次向 IWDT_KR 寄存器写入 0xAAAA 的时候，这个值会被更新到看门狗计数器中，如果想延时长一点，这个值就该大一些。因为看门狗计数器正是从这个值开始向下计数。定时的长度是由这个值和预分频器的设置值来共同决定的。为了能够改变看门狗计数器重载值，IWDT_FR 寄存器中的 BUSY 位必须先清零。</p> <p>Note: 如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，从这个地方读数据的时候要保证 IWDT_FR 寄存器中的 BUSY 位为 0 才行。</p>

20.5.2.4 IWDT 标志寄存器 (IWDT_FR)

IWDT 标志寄存器(IWDT_FR)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00001111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																BUSY				STR	CNT<11:0>										

—	Bits 31-16	—	—
BUSY	Bit 15	R	寄存器更新 该位由硬件设置，以指示关键值、预分频器、重新加载或窗口值正在进行更新。当完成更新操作后（需要多达 5 个 RC 周期），会通过硬件将该位复位。 必须等到 BUSY 位被清零后，才能更新关键值或预分频器或重载或窗口值。
—	Bits 14-13	—	—
STR	Bit 12	R	IWDT 正在进行 此位由硬件设置，以指示 IWDT 正在进行。
CNT	Bits 11-0	R	12-bit 计数器(MSB to LSB) 这些位包含看门狗计数器的值。

20.5.2.5 IWDT 窗口寄存器 (IWDT_WINR)

IWDT 窗口寄存器(IWDT_WINR)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00001111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												WIN<11:0 ^Δ			

—	Bits 31-12	—	—
WIN	Bits 11-0	R/W	<p>看门狗计数器窗口值</p> <p>这些位平时处于写保护状态，这些位包含的是窗口值和向下计数器的比较上限。为了阻止复位信号的产生，必须在向下计数器递减到窗口值和 0x0 之间的某个值时重加载它。</p> <p>要想改变这个窗口值，必须先保证IWDT_FR 中的BUSY 位为 0。</p> <p>Note: 如果正在对该寄存器执行写操作，则读取的值可能不是最新的/ 有效的。因此，从这个地方读数据的时候要保证 IWDT_FR 寄存器中的 BUSY位为 0 才行。</p>

第21章 窗口看门狗 (WWDT)

21.1 概述

窗口看门狗通常被用来监测，由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。窗口看门狗 WWDT 对于过早或过晚喂狗都将产生 WWDT 复位，可用于检测软件没有喂狗或在禁止喂狗区内喂狗行为，防止程序运行至不可控状态。看门狗电路在达到预置的时间周期时，如果 7 位的递减计数器数值(在控制寄存器中) 未被刷新，会产生一个 MCU 复位。在递减计数器达到窗口寄存器数值之前，如果 7 位的递减计数器数值被刷新，那么也将产生一个 MCU 复位。这表明递减计数器需要在一个有限的时间窗口中被刷新。

WWDT 时钟从 APB1 时钟预分频，适合要求看门狗在一个精确时间窗口内做出反应的应用程序。

21.2 特性

窗口看门狗 WWDT 功能概述：

- ◆ 可编程的递减计数器
- ◆ 看门狗被启动后，复位产生的条件
 - ◇ 当递减计数器的值从 0x3F 翻转时，则产生复位。
 - ◇ 当递减计数器在窗口外被重新装载，则产生复位。
- ◆ 提前唤醒中断：如果启用中断，当递减计数器等于 0x40 时产生提前唤醒中断。

21.3 结构图

图 21-1 描述了窗口看门狗模块的功能框图。

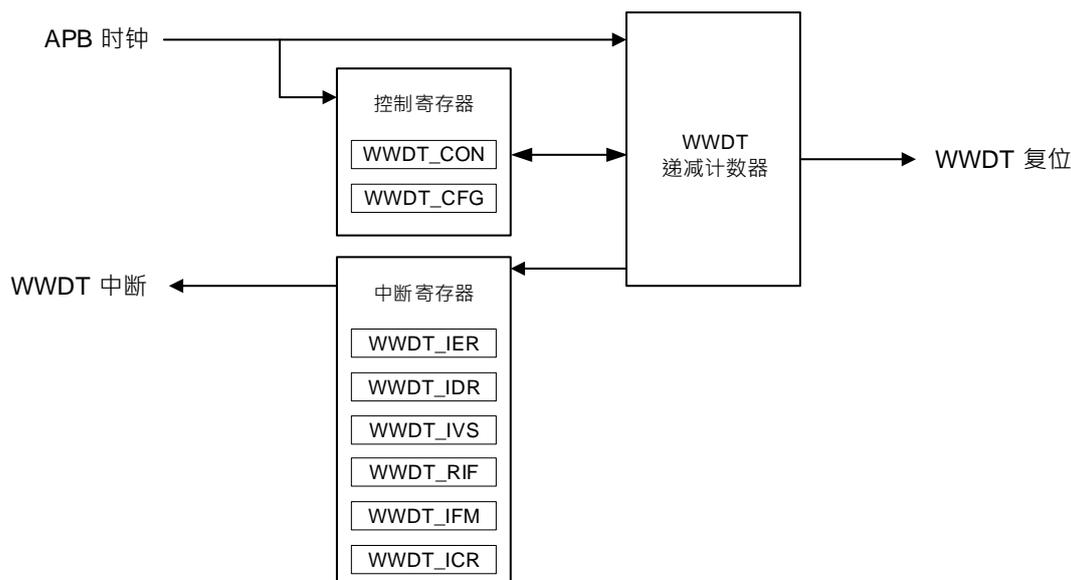


图 21-1 窗口看门狗框图

21.4 功能描述

上电复位后，窗口看门狗不启动，需通过软件配置使能窗口看门狗(WWDT_CON 寄存器中的 WDGA 位被置'1')。当 7 位(T[6:0])递减计数器从 0x3F 翻转时，则产生一个复位。如果软件在计数器值大于窗口寄存器中的数值时重新装载计数器，将产生一个复位。

应用程序在正常运行过程中必须定期地写入 WWDT_CON 寄存器以防止 MCU 发生复位。只有当计数器值小于窗口寄存器的值时，才能进行写操作。储存在 WWDT_CON 寄存器中的数值必须在 0xFF 和 0xC0 之间。

21.4.1 启用看门狗

设置 WWDT_CON 寄存器的 WDGA 位能够开启看门狗，随后它不能再被关闭，除非发生复位。

21.4.2 控制递减计数器

递减计数器处于自由运行状态：即使看门狗被禁止，递减计数器仍继续递减计数。

T[5:0]位包含了看门狗产生复位之前的计时数目，配置寄存器(WWDT_CFG)中包含窗口的上限值：要避免产生复位，递减计数器必须在其值小于窗口寄存器的数值并且大于 0x3F 时被重新装载，图 21-3 描述了窗口寄存器的工作过程。

21.4.3 高级看门狗中断功能

如果在实际复位产生之前必须进行特定的安全操作或数据记录，可以用提前唤醒中断。设置 WWDT_IER 寄存器中的 EWIE 位开启该中断。在复位之前当递减计数器到达 0x40 时，则产生此中断，同时可以用相应的中断服务程序(ISR)来触发特定的行为(例如通信或数据记录)，在某些应用中，提前唤醒中断可以用来管理软件系统检测和/或系统恢复/故障弱化，但不产生 WWDT 复位。在这种情况下，相应的中断服务程序 (ISR) 将重新加载 WWDT 计数器，以避免 WWDT 复位。

注：当提前唤醒中断无法启用时，例如由于系统锁定在更高优先级任务，最终将产生 WWDT 复位。

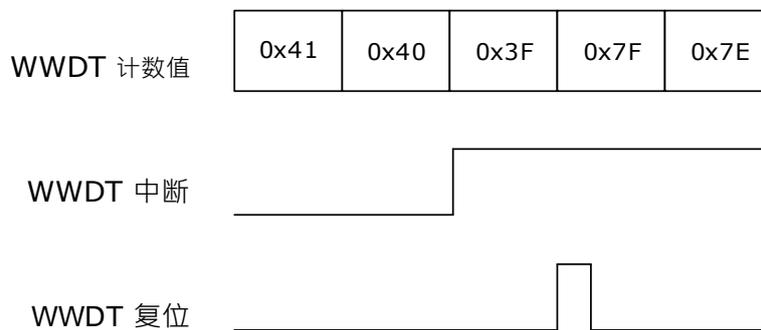


图 21-2 WWDT 中断示意图

21.4.4 如何编程看门狗超时

计算超时的公式如下：

$$t_{\text{WWDT}} = t_{\text{PCLK}} \times 4096 \times 2^{\text{WDGTB}} \times (t[5:0] + 1) \quad (\text{ms})$$

其中：

t_{WWDT} : WWDT 超时时间

t_{PCLK} : APB1 以 ms 为单位的时钟周期

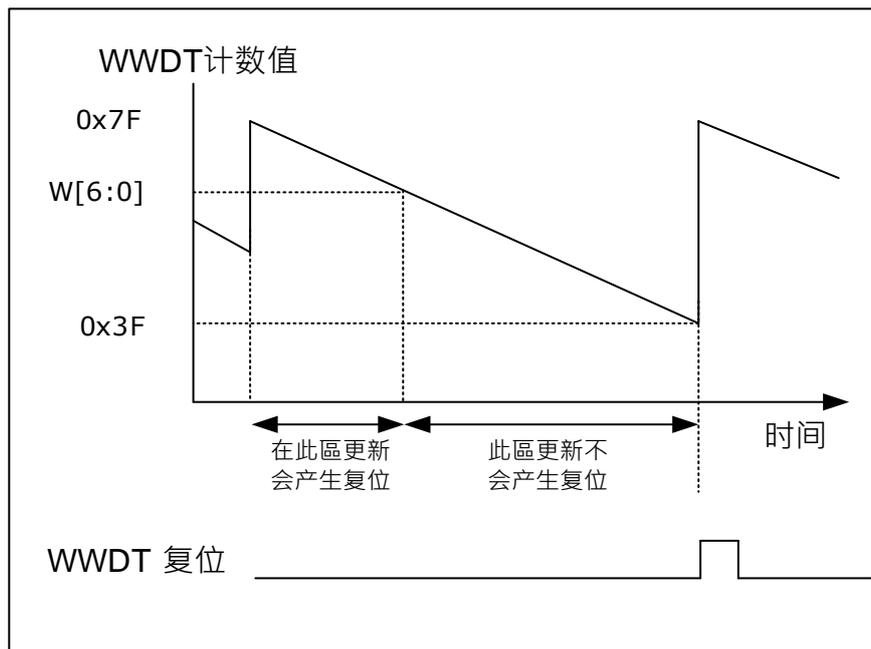


图 21-3 窗口看门狗时序图

21.4.5 调试模式

当微控制器进入调试模式时(内核被暂停)，看门狗计数器可以继续运行，也可以被停止。

21.5 特殊功能寄存器

21.5.1 寄存器列表

WWDT 寄存器列表			
名称	偏移地址	类型	描述
WWDT_CON	000 _H	R/W	WWDT 控制寄存器
WWDT_CFG	004 _H	R/W	WWDT 配置寄存器
WWDT_IER	008 _H	W1	WWDT 中断使能寄存器
WWDT_IDR	00C _H	W1	WWDT 中断禁止寄存器
WWDT_IVS	010 _H	R	WWDT 中断有效位状态寄存器
WWDT_RIF	014 _H	R	WWDT 无屏蔽中断标志寄存器
WWDT_IFM	018 _H	R	WWDT 屏蔽中断标志寄存器
WWDT_ICR	01C _H	C_W1	WWDT 中断清除寄存器

21.5.2.2 WWDT 配置寄存器 (WWDT_CFG)

WWDT 配置寄存器 (WWDT_CFG)																																									
偏移地址: 04H																																									
复位值: 00000000_00000000_00000000_01111111 _B																																									
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
																												WDGTB<1:0>							W<6:0>						

—	Bits 31-9	—	—
WDGTB	Bits 8-7	R/W	时基 预分频器的时基可以设置如下: 00: 计时器时钟 (PCLK 除以 4096) 分频器 1 01: 计时器时钟 (PCLK 除以 4096) 分频器 2 10: 计时器时钟 (PCLK 除以 4096) 分频器 4 11: 计时器时钟 (PCLK 除以 4096) 分频器 8
W	Bits 6-0	R/W	7-bit 窗口值 这些位包含了用来与递减计数器进行比较用的窗口值 (必须在 0x40~0x7F 之间)

21.5.2.3 WWDT 中断使能寄存器 (WWDT_IER)

WWDT 中断使能寄存器(WWDT_IER)																															
偏移地址: 08H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															EWIE

—	Bits 31-1	—	—
EWIE	Bit 0	W1	提前唤醒中断使能 0: 写入0无效 1: 启用中断

21.5.2.4 WWDT 中断禁止寄存器 (WWDT_IDR)

WWDT 中断禁止寄存器 (WWDT_IDR)																																
偏移地址: 0C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																															EWID	

—	Bits 31-1	—	—
EWID	Bit 0	W1	提前唤醒中断禁止 0: 写入0无效 1: 禁止中断

21.5.2.5 WWDT 中断有效位状态寄存器 (WWDT_IVS)

WWDT 中断有效位状态寄存器 (WWDT_IVS)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															EWIS

—	Bits 31-1	—	—
EWIS	Bit 0	R	提前唤醒中断有效位 0: 禁止中断 1: 启用中断

第22章 实时时钟 (RTC)

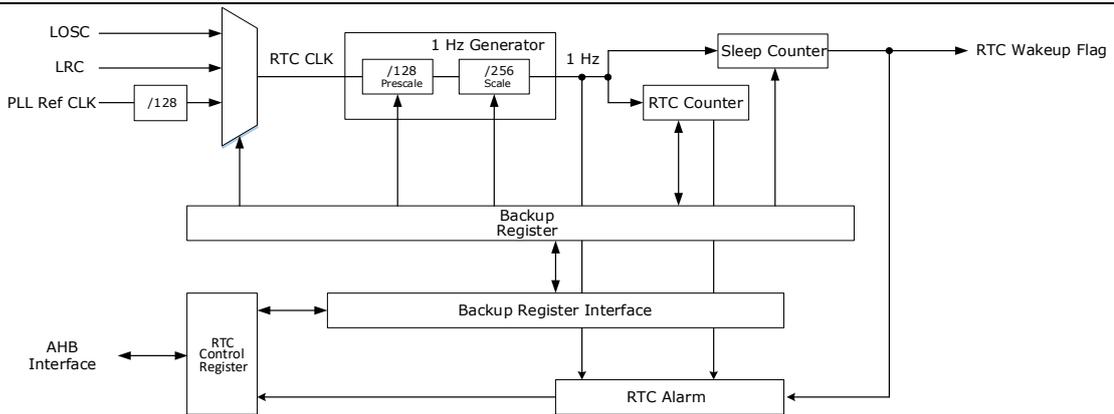
22.1 概述

Real Time Clock (RTC)可提供用户准确的时间以及日期讯息，这些讯息皆以 BCD 的格式储存在 RTC 控制缓存器内，同时提供用户自行设定闹铃功能。RTC 的 CLK 来源也分为 3 种供用户选择，这 3 种 CLK 分别为外部 32.768 kHz 的晶体振荡器、内部约 32 kHz 的 RC 振荡器以及 PLL 所使用的参考 CLK。当 MCU 处在低功耗模式时，RTC 仍可继续提供准确的时间讯息。

22.2 特性

- ◆ 支持日历显示：年、月、日。
- ◆ 支持时间显示：星期、小时、分钟、秒。
- ◆ 日历与时间皆以 BCD 格式储存。
- ◆ 支持闰年侦测功能。
- ◆ 支持用户自行设定 RTC 计数器校正功能。
- ◆ 支持用户自行设定闹铃中断功能。
- ◆ 支持年、月、日、星期、小时、分钟、秒的翻转(Rollover)中断。
- ◆ 支持 MCU 在睡眠模式下保持计数。
- ◆ 支持用户自行配置睡眠计数器。
- ◆ 支持最长 16777216 秒(194 天)的睡眠计数。
- ◆ 支持最短 1/16 秒的睡眠计数。
- ◆ 支持侦测日历、时间是否已被清除。

22.3 结构图



RTC 主要分为 2 个部分，第一个部分为 RTC 计数器，第二个部分为 RTC 时间存储与闹铃。RTC 计数器的部分主要负责计算日期以及时间，并于 MCU 处在睡眠模式时提供唤醒 CPU 功能。而另一个部分则是提供日期以及时间的读取，同时依据用户所设定的闹铃发出中断。

22.4 功能描述

22.4.1 设定并开启 RTC

在开启 RTC 计数之前，必须先对 **RTC_CON** 缓存器进行设定，并于 **RTC_CAL** 填入日期以及在 **RTC_TIME** 填入时间，日期以及时间皆须以 BCD 格式填入。用户可设定 **CKSEL** 来选定 RTC CLK 的来源为 **LOSC**、**LRC** 或是 **PLL** 参考 CLK，须注意的是当 MCU 在睡眠模式时，是无法选择 **PLL** 参考 CLK 当作 RTC CLK 的来源。紧接着必须依据所选的 RTC CLK 来源设定 **PSCALE** 与 **SCALE** 来产生不同的分频系数，由此产生准确的 1 秒(1 Hz) 信号，在默认情况下这 2 个缓存器是将 RTC CLK 除上 32768 来产生 1 秒信号。

当设定完成后，即可设定 **RTC_EN** 来开启 RTC 的功能。在 RTC 计数期间，为了确保日期以及时间不能随意更改，因此会暂时锁定 **RTC_CAL** 以及 **RTC_TIME** 的写入功能，若需要更改 RTC 的日期以及时间，则必须关闭 RTC 计数功能以后才能够进行修改。

22.4.2 读取 RTC 日期与时间

由于 RTC 日期与时间的计数是位在备份寄存器区，因此每一秒皆须将日期以及时间同步回 AHB 的缓存器，为了确保用户不会读取到错误的时间讯息，在读取 RTC 日期以及时间之前，可以先去检查 **RTC_STAT** 缓存器内的 **SYNDONE** 信号，当此信号被设定为 1 时代表已同步完成，若此信号为 0 则代表 RTC 正在同步备份寄存器区的时间讯息。

此外，用户也可以设定 **RTC_BKEN** 来开启直接读取备份寄存器内所保存的日期以及时间的功能。若由备份寄存器接口直接读取日期与时间的话，需等待 19 个 AHB CLK 的读取时间。读取完毕以后，需自行手动关闭 **RTC_BKEN**。

当 MCU 进入待机模式 0 以上的低功耗模式时，会关闭 AHB 区的电源，导致 RTC 控制缓存器内的时间讯息被清除，因此当 MCU 被唤醒时若要读取日期以及时间讯息的话，需要额外等待 1 秒的时间让 RTC 同步备份寄存器区的时间讯息，或是直接通过备份寄存器接口读取日期与时间。

22.4.3 RTC 校正

由于 RTC CLK 的来源并非准确的 32.768 kHz, 因此需要针对所使用的 CLK 进行频率补偿, 以确保能计算出准确的 1 秒。校正数值的计算方式如下:

1. 每秒校正数值 = 实际频率_{整数} - 预期频率_{整数}。若此数值为正值代表实际频率高于预期频率, 需要增加计数的次数来算出准确的 1 秒; 若此数值为负值代表实际频率低于预期频率, 需要减少计数的次数来算出准确的 1 秒。
2. 每分钟校正数值 = (实际频率_{小数} - 预期频率_{小数}) x 60。若此数值为正值, 则需要增加计数的次数, 反之则需要减少计数的次数。

当计算完校正数值以后, 则依据数值的正负号决定要增加校正数值或是减少校正数值。校正的周期可以选择每秒校正或是每分钟校正。下列为 RTC 校正范例, 校正周期设定为每秒校正。

22.4.3.1 RTC 校正示例 1(实际频率高于预期频率)

预期频率: 32768 Hz

实际频率: 32770.5 Hz

每秒校正数值: $32770 - 32768 = +2$

每分钟校正数值: $0.5 \times 60 = +30$

校正步骤如下:

1. 设定 **RTC_CON(0x0)** 内的 **CYCLE** 为 1, 开启每秒校正功能, 并设定 **CALVAL** 的数值为 2。
2. 设定 **MODE** 为 0, 开启每秒增加校正数值。
3. 设定 **CALIB** 开启 RTC 校正功能。
4. 当第 60 秒发生时, 更改 RTC 校正数值 **CALVAL** 为 $(2+30) = 32$, 并于下一秒将校正数值更改回 2。
5. 重复第 4 步校正流程。

22.4.3.2 RTC 校正示例 2(实际频率高于预期频率)

预期频率: 32768 Hz

实际频率: 32766.5 Hz

每秒校正数值: $32766 - 32768 = -2$

每分钟校正数值: $0.5 \times 60 = +30$

校正步骤如下:

1. 设定 **RTC_CON(0x0)** 内的 **CYCLE** 为 1, 并设定 **CALVAL** 的数值为 2。
2. 设定 **MODE** 为 1, 开启每秒减少校正数值。
3. 设定 **CALIB** 开启 RTC 每秒校正功能。
4. 当第 60 秒发生时, 更改 RTC 校正数值 **CALVAL** 为 $(-2+30) = 28$ 同时更改 **MODE** 为 0 开启每秒增加校正数值, 并于下一秒将校正数值更改回 2, 同时更改 **MODE** 为 1, 恢复每秒减少校正数值。
5. 重复第 4 步校正流程。

22.4.4 RTC 睡眠计数器

RTC 的睡眠计数器可由用户自行设定 **RTC_WKUP(0x04)** 内的 **WKSEL** 来决定开启计数的时间点, 当用户设定 **WKSEL** 为 0x0 时代表不使用 RTC 的睡眠计数器; 当设定 **WKSEL** 为 0x1 时, 此时睡眠计数器可当成一般计数器使用, 用户可由此设定 **WKCAL** 来决定要计数多少秒, 当计数完成时可发出中断信号告知。当设定 **WKSEL** 为 0x2 时, 睡眠计数器会在收到 CPU 发出的 **SLEEP** 信号以后开始计数, 并于计数完成时唤醒 MCU。计数器的最大值可设定至 16777216 秒, 即 194 天。此外, RTC 的睡眠计数器也可以设定计数周期。用户可以设定 **WKSCAL** 来决定睡眠计数器的计数周期, 分别为 1/2 秒、1/4 秒、1/8 秒或是 1/16 秒计数一次。

22.4.5 RTC 闹铃

RTC 的闹铃会在 RTC 的日期或时间符合设定时发出中断告知用户。用户可设定 **RTC_ALEN(0x18)** 开启 RTC 闹铃的种类, 并于 **RTC_ALTIME(0x10)** 以及 **RTC_ALCAL(0x14)** 设定产生闹铃的时间。闹铃的种类分为秒闹铃、分钟闹铃、小时闹铃、星期闹铃、日闹铃、月闹铃、年闹铃以及日期与时间完全相符的闹铃。

22.4.6 RTC 触发 ADC 或 DAC

RTC 允许在中断条件满足时送出 ADC 以及 DAC 的触发信号, 用户可设定 **RTC_TRIG(0x1C)** 寄存器来决定要送出触发信号的条件为何, 此触发信号仅会维持 32 个 AHB CLK 的宽度。

22.5 特殊功能寄存器

22.5.1 寄存器列表

RTC 寄存器列表			
名称	偏移地址	类型	描述
RTC_CON	000H	R/W	RTC 控制寄存器
RTC_WKUP	004H	R/W	RTC 唤醒计数器寄存器
RTC_TIME	008H	R/W	RTC 日历时间寄存器
RTC_CAL	00CH	R/W	RTC 日历日期寄存器
RTC_ALTIME	010H	R/W	RTC 闹钟定时器寄存器
RTC_ALCAL	014H	R/W	RTC 闹钟日历寄存器
RTC_ALEN	018H	R/W	RTC 闹钟使能寄存器
RTC_TRIG	01CH	R/W	RTC 触发源选择寄存器
RTC_IER	020H	W1	RTC 中断使能寄存器
RTC_IDR	024H	W1	RTC 中断禁止寄存器
RTC_IVS	028H	R	RTC 中断有效位状态寄存器
RTC_RIF	02CH	R	RTC 无屏蔽中断标志寄存器
RTC_IFM	030H	R	RTC 屏蔽中断标志寄存器
RTC_ICR	034H	C_W1	RTC 中断清除寄存器
RTC_STAT	040H	R	RTC 状态寄存器
RTC_BKEN	044H	R/W	RTC 日历读取备份控制寄存器

22.5.2 寄存器描述

22.5.2.1 RTC 控制寄存器(RTC_CON)

RTC 控制寄存器(RTC_CON)																															
偏移地址: 00H																															
复位值: 00000000_11111111_01111111_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE		CALVAL<5:0>					SCALE<7:0>							PSCALE<6:0>						CYCLE		CALIB		CKSEL<1:0>			RTCEN				

—	Bit 31	—	—
MODE	Bit 30	R/W	RTC 时钟校准模式 1: 匹配校准周期时减小校准值 0: 匹配校准周期时增加校准值
CALVAL	Bits 29-24	R/W	RTC 时钟校准值
SCALE	Bits 23-16	R/W	RTC 时钟计数器比率 此计数器的最大分频器为 (SCALE+1).
—	Bit 15	—	—
PSCALE	Bits 14-8	R/W	RTC 时钟计数器预分频器 此计数器的最大分频比为 (PSCALE+1).
—	Bits 7-6	—	—
CYCLE	Bit 5	R/W	RTC 校准周期选择 0: 启用校准功能时, 校准周期为 60 秒 1: 启用校准功能时, 校准周期为 1 秒
CALIB	Bit 4	R/W	RTC 校准功能启用 0: 禁用校准功能 1: 启用校准功能
CKSEL	Bits 3-2	R/W	RTC 时钟源选择 00: 没有时钟 01: LOSC 振荡器时钟用作 RTC 时钟 10: LRC 振荡器时钟用作 RTC 时钟 11: PLL 参考时钟源除以 128 用作 RTC 时钟 注: 在睡眠模式中, 待机模式 0 以上仅支持 LRC 和 LOSC 作为 RTC 时钟源。
—	Bit 1	—	—
RTCEN	Bit 0	R/W	RTC 计数器启用 0: 计数器未开启 1: 计数器开启

22.5.2.2 RTC 唤醒计数器寄存器(RTC_WKUP)

RTC 唤醒计数器寄存器(RTC_WKUP)																															
偏移地址: 04H																															
复位值: 00000000_11111111_11111111_11111111B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		WKSCAL<3:0>				WKSEL<1:0>		WKCAL<7:0>									WKCAL<15:0>														

—	Bits 31-30	—	—
WKSCAL	Bits 29-26	R/W	RTC 唤醒计数器比率值 0x0: 除 1, 唤醒计数器每秒计数 0x1: 除 2, 唤醒计数器每 1/2 秒计数一次 0x2: 除 4, 唤醒计数器每 1/4 秒计数一次 0x4: 除以 8, 唤醒计数器每 1/8 秒计数一次 0x8: 除以 16, 唤醒计数器将每 1/16 秒计数一次
WKSEL	Bits 25-24	R/W	RTC 唤醒计数器事件选择 00: 关闭 01: RTC 启动时启动计数器 10: 接收睡眠标志时启动计数器 11: 保留
WKCAL	Bits 23-0	R/W	RTC 唤醒计数器设定值 最大值为 WKCAL 秒

22.5.2.3 RTC 日历时间寄存器(RTC_TIME)

RTC 日历时间寄存器(RTC_TIME)																															
偏移地址: 08H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					WEEK<2:0>					HOUR_T<1:0>		HOUR_U<3:0>				MIN_T<2:0>		MIN_U<3:0>				SEC_T<2:0>		SEC_U<3:0>							

—	Bits 31-27	—	—
WEEK	Bits 26-24	R/W	星期值 星期值，以 BCD 格式存储，数值介于 1~7
—	Bits 23-22	—	—
HOUR_T	Bits 21-20	R/W	小时的十位 时，十位值以 BCD 格式存储
HOUR_U	Bits 19-16	R/W	小时的个位 时，个位值以 BCD 格式存储
—	Bit 15	—	—
MIN_T	Bits 14-12	R/W	分钟的十位 分，十位值以 BCD 格式存储
MIN_U	Bits 11-8	R/W	分钟的个位 分，个位值以 BCD 格式存储
—	Bit 7	—	—
SEC_T	Bits 6-4	R/W	秒的十位 秒，十位值以 BCD 格式存储
SEC_U	Bits 3-0	R/W	秒的个位 秒，个位值以 BCD 格式存储

22.5.2.4 RTC 日历日期寄存器(RTC_CAL)

RTC 日历日期寄存器(RTC_CAL)																																									
偏移地址: 0C _H																																									
复位值: 00000000_00000000_00000000_00000000 _B																																									
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
																YEAR_T<3:0>				YEAR_U<3:0>								MON_T		MON_U<3:0>								DATE_T<1:0>		DATE_U<3:0>	

—	Bits 31-24	—	—
YEAR_T	Bits 23-20	R/W	年份的十位 年, 十位值以 BCD 格式存储
YEAR_U	Bits 19-16	R/W	年份的个位 年, 个位值以 BCD 格式存储
—	Bits 15-13	—	—
MON_T	Bit 12	R/W	月份的十位 月, 十位值以 BCD 格式存储
MON_U	Bits 11-8	R/W	月份的个位 月, 个位值以 BCD 格式存储
—	Bits 7-6	—	—
DATE_T	Bits 5-4	R/W	日期的十位 日, 十位值以 BCD 格式存储
DATE_U	Bits 3-0	R/W	日期的个位 日, 个位值以 BCD 格式存储

22.5.2.5 RTC 闹钟定时器寄存器(RTC_ALTIME)

RTC 闹钟定时器寄存器(RTC_ALTIME)																																															
偏移地址: 10H																																															
复位值: 00000000_00000000_00000000_00000000 _B																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
				WEEK<2:0>												HOUR_T<1:0>				HOUR_U<3:0>								MIN_T<2:0>				MIN_U<3:0>								SEC_T<2:0>				SEC_U<3:0>			

—	Bits 31-27	—	—
WEEK	Bits 26-24	R/W	星期几闹钟 星期几闹钟, 采用 BCD 格式, 数值介于 1~7
—	Bits 23-22	—	—
HOUR_T	Bits 21-20	R/W	小时的十位闹钟 小时的十位闹钟, 采用 BCD 格式
HOUR_U	Bits 19-16	R/W	小时的个位闹钟 小时的个位闹钟, 采用 BCD 格式
—	Bit 15	—	—
MIN_T	Bits 14-12	R/W	分钟的十位闹钟 分钟的十位闹钟, 采用 BCD 格式
MIN_U	Bits 11-8	R/W	分钟的个位闹钟 分钟的个位闹钟, 采用 BCD 格式
—	Bit 7	—	—
SEC_T	Bits 6-4	R/W	秒的十位闹钟 秒的十位闹钟, 采用 BCD 格式
SEC_U	Bits 3-0	R/W	秒的个位闹钟 秒的个位闹钟, 采用 BCD 格式

22.5.2.6 RTC 闹钟日历寄存器(RTC_ALCAL)

RTC 闹钟日历寄存器 (RTC_ALCAL)																															
偏移地址: 14H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								YEAR_T<3:0>				YEAR_U<3:0>							MON_T	MON_U<3:0>						DATE_T<1:0>		DATE_U<3:0>			

—	Bits 31-24	—	—
YEAR_T	Bits 23-20	R/W	年份的十位闹钟 年份的十位闹钟, 采用 BCD 格式
YEAR_U	Bits 19-16	R/W	年份的个位闹钟 年份的个位闹钟, 采用 BCD 格式
—	Bits 15-13	—	—
MON_T	Bit 12	R/W	月份的十位闹钟 月份的十位闹钟, 采用 BCD 格式
MON_U	Bits 11-8	R/W	月份的个位闹钟 月份的个位闹钟, 采用 BCD 格式
—	Bits 7-6	—	—
DATE_T	Bits 5-4	R/W	日期的十位闹钟 日期的十位闹钟, 采用 BCD 格式
DATE_U	Bits 3-0	R/W	日期的个位闹钟 日期的个位闹钟, 采用 BCD 格式

22.5.2.7 RTC 闹钟使能寄存器(RTC_ALEN)

RTC 闹钟使能寄存器(RTC_ALEN)																																
偏移地址: 18H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																										YEAR	MONTH	DATE	WEEK	HOUR	MIN	SEC

—	Bits 31-7	—	—
YEAR	Bit 6	R/W	年份闹钟功能启用 0=禁止 1=使能
MONTH	Bit 5	R/W	月份闹钟功能启用 0=禁止 1=使能
DATE	Bit 4	R/W	日期闹钟功能启用 0=禁止 1=使能
WEEK	Bit 3	R/W	星期闹钟功能启用 0=禁止 1=使能
HOUR	Bit 2	R/W	小时闹钟功能启用 0=禁止 1=使能
MIN	Bit 1	R/W	分钟闹钟功能启用 0=禁止 1=使能
SEC	Bit 0	R/W	秒闹钟功能启用 0=禁止 1=使能

22.5.2.8 RTC 触发源选择寄存器(RTC_TRIG)

RTC 触发源选择寄存器(RTC_TRIG)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															WKTM	F1HZ	RYEAR	RMON	RDATE	RWEEK	RHOUR	RMIN	RSEC	AMALL	AYEAR	AMON	ADATE	AWEEK	AHOUR	AMIN	ASEC

—	Bits 31-17	—	—
WKTM	Bit 16	R/W	RTC 唤醒计数器匹配触发启用 0=禁止 1=使能
F1HZ	Bit 15	R/W	RTC 1 Hz 触发启用 0=禁止 1=使能
RYEAR	Bit 14	R/W	RTC 跨世纪闹钟触发启用 0=禁止 1=使能
RMON	Bit 13	R/W	RTC 跨年闹钟触发启用 0=禁止 1=使能
RDATE	Bit 12	R/W	RTC 跨月闹钟触发启用 0=禁止 1=使能
RWEEK	Bit 11	R/W	RTC 跨周闹钟触发启用 0=禁止 1=使能
RHOUR	Bit 10	R/W	RTC 跨天闹钟触发启用 0=禁止 1=使能
RMIN	Bit 9	R/W	RTC 跨小时闹钟触发启用 0=禁止 1=使能
RSEC	Bit 8	R/W	RTC 跨分闹钟触发启用 0=禁止 1=使能
AMALL	Bit 7	R/W	RTC 匹配所有闹钟触发启用 选择此触发源时, 用户应在 RTC_ALEN 中启用所有闹钟功能 0=禁止 1=使能
AYEAR	Bit 6	R/W	RTC 匹配年份闹钟触发启用 0=禁止 1=使能
AMON	Bit 5	R/W	RTC 匹配月闹钟触发启用

			0=禁止 1=使能
ADTAE	Bit 4	R/W	RTC 匹配日期闹钟触发启用 0=禁止 1=使能
AWEEK	Bit 3	R/W	RTC 匹配星期几闹钟触发启用 0=禁止 1=使能
AHOUR	Bit 2	R/W	RTC 匹配小时闹钟触发启用 0=禁止 1=使能
AMIN	Bit 1	R/W	RTC 匹配分钟闹钟触发启用 0=禁止 1=使能
ASEC	Bit 0	R/W	RTC 匹配秒闹钟触发启用 0=禁止 1=使能

22.5.2.9 RTC 中断使能寄存器(RTC_IER)

RTC 中断使能寄存器(RTC_IER)																															
偏移地址: 20H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															WKTM	F1HZ	RYEAR	RMON	RDATE	RWEEK	RHOUR	RMIN	RSEC	AMALL	AYEAR	AMON	ADATE	AWEEK	AHOUR	AMIN	ASEC

—	Bits 31-17	—	—
WKTM	Bit 16	W1	RTC 唤醒计数器匹配中断使能 用户可以设置此位以启用唤醒计数器匹配中断。当 RTC_WKUP 唤醒计数器匹配 WKCAL 时，将设置此中断。
F1HZ	Bit 15	W1	RTC 1 Hz 中断使能 用户可以设置此位以启用 1 Hz 中断。此中断将设置在 1 Hz 标志被设置之后。
RYEAR	Bit 14	W1	RTC 跨世纪闹钟中断使能 用户可以设置此位以启用跨世纪中断。这一中断将在进入新世纪后被设置。
RMON	Bit 13	W1	RTC 跨年闹钟中断使能 用户可以设置此位以启用跨年中断。这个中断将在新的一年后被设置。
RDATE	Bit 12	W1	RTC 跨月闹钟中断使能 用户可以设置此位以启用跨月中断。此中断将在新的月份后被设置。
RWEEK	Bit 11	W1	RTC 跨周闹钟中断使能 用户可以设置此位以启用跨周中断。此中断将在新的周被设置。
RHOUR	Bit 10	W1	RTC 跨天闹钟中断使能 用户可以设置此位以启用跨天中断。此中断将在新的天被设置。
RMIN	Bit 9	W1	RTC 跨小时闹钟中断使能 用户可以设置此位以启用跨小时中断。此中断将在新的小时被设置。
RSEC	Bit 8	W1	RTC 跨分闹钟中断使能 用户可以设置此位以启用跨分中断。此中断将在新的分钟被设置。
AMALL	Bit 7	W1	RTC 匹配所有闹钟中断使能 用户可以设置此位以匹配所有的闹钟中断。当 RTC 定时器与 RTC_ALCAL 和 RTC_ALTIME 中的值相匹配时，将设置此中断。
AYEAR	Bit 6	W1	RTC 匹配年闹钟中断使能 用户可以设置此位以启用匹配年闹钟中断。此中断将在 RTC 定时器的年数值匹配 RTC_ALTIME 中年的配置时被设置。

AMON	Bit 5	W1	RTC 匹配月闹钟中断使能 用户可以设置此位以启用匹配月闹钟中断。此中断将在 RTC 定时器的月份数值匹配 RTC_ALTIME 中月的配置时被设置。
ADTAE	Bit 4	W1	RTC 匹配日期闹钟中断使能 用户可以设置此位以启用匹配日期闹钟中断。此中断将在 RTC 定时器的日期数值匹配 RTC_ALTIME 中日期的配置时被设置。
AWEEK	Bit 3	W1	RTC 匹配星期几闹钟中断使能 用户可以设置此位以启用匹配星期几闹钟中断。此中断将在 RTC 定时器的星期几数值匹配 RTC_ALTIME 中星期几的配置时被设置。
AHOUR	Bit 2	W1	RTC 匹配小时闹钟中断使能 用户可以设置此位以启用匹配小时闹钟中断。此中断将在 RTC 定时器的小时数值匹配 RTC_ALTIME 中小时的配置时被设置。
AMIN	Bit 1	W1	RTC 匹配分钟闹钟中断使能 用户可以设置此位以启用匹配分钟闹钟中断。此中断将在 RTC 定时器的分钟数值匹配 RTC_ALTIME 中分钟的配置时被设置。
ASEC	Bit 0	W1	RTC 匹配秒闹钟中断使能 用户可以设置此位以启用匹配秒闹钟中断。此中断将在 RTC 定时器的秒数值匹配 RTC_ALTIME 中秒的配置时被设置。

22.5.2.10 RTC 中断禁止寄存器(RTC_IDR)

RTC 中断禁止寄存器(RTC_IDR)																															
偏移地址: 24H																															
复位值: 00000000_00000000_00000000_00000000B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															WKTM	F1HZ	RYEAR	RMON	RDATE	RWEEK	RHOUR	RMIN	RSEC	AMALL	AYEAR	AMON	ADATE	AWEEK	AHOUR	AMIN	ASEC

—	Bits 31-17	—	—
WKTM	Bit 16	W1	RTC 唤醒计数器匹配中断禁用 用户可以将该位置 1 以禁止唤醒计数器匹配中断
F1HZ	Bit 15	W1	RTC 1 Hz 中断禁用 用户可以将该位置 1 以禁止唤醒计数器匹配中断
RYEAR	Bit 14	W1	RTC 跨世纪闹钟中断禁用 用户可以设置此位禁用跨世纪闹钟中断
RMON	Bit 13	W1	RTC 跨年闹钟中断禁用 用户可以设置此位禁用跨年闹钟中断
RDATE	Bit 12	W1	RTC 跨月闹钟中断禁用 用户可以设置此位禁用跨月闹钟中断
RWEEK	Bit 11	W1	RTC 跨周闹钟中断禁用 用户可以设置此位禁用跨周闹钟中断
RHOUR	Bit 10	W1	RTC 跨天闹钟中断禁用 用户可以设置此位禁用跨天闹钟中断
RMIN	Bit 9	W1	RTC 跨小时闹钟中断禁用 用户可以设置此位禁用跨小时闹钟中断
RSEC	Bit 8	W1	RTC 跨分闹钟中断禁用 用户可以设置此位禁用跨分闹钟中断
AMALL	Bit 7	W1	RTC 匹配所有闹钟中断禁用 用户可以将此位设置为禁用匹配所有闹钟中断
AYEAR	Bit 6	W1	RTC 匹配年闹钟中断禁用 用户可以将此位设置为禁用匹配年份闹钟中断
AMON	Bit 5	W1	RTC 匹配月闹钟中断禁用 用户可以将此位设置为禁用匹配月闹钟中断
ADTAE	Bit 4	W1	RTC 匹配日期闹钟中断禁用 用户可以将此位设置为禁用匹配日期闹钟中断
AWEEK	Bit 3	W1	RTC 匹配星期几闹钟中断禁用 用户可以将此位设置为禁用匹配星期几闹钟中断
AHOUR	Bit 2	W1	RTC 匹配小时闹钟中断禁用 用户可以将此位设置为禁用匹配小时闹钟中断
AMIN	Bit 1	W1	RTC 匹配分钟闹钟中断禁用 用户可以将此位设置为禁用匹配分钟闹钟中断
ASEC	Bit 0	W1	RTC 匹配秒闹钟中断禁用 用户可以将此位设置为禁用匹配秒闹钟中断

22.5.2.11 RTC 中断有效位状态寄存器(RTC_IVS)

RTC 中断有效位状态寄存器(RTC_IVS)																															
偏移地址: 28H																															
复位值: 00000000_00000000_00000000_00000000B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															WKTM	F1HZ	RYEAR	RMON	RDATE	RWEEK	RHOUR	RMIN	RSEC	AMALL	AYEAR	AMON	ADATE	AWEEK	AHOUR	AMIN	ASEC

—	Bits 31-17	—	—
WKTM	Bit 16	R	RTC 唤醒计数器匹配中断有效位状态 0: 中断被禁用 1: 中断启用
F1HZ	Bit 15	R	RTC 1 Hz 中断有效位状态 0: 中断被禁用 1: 中断启用
RYEAR	Bit 14	R	RTC 跨世纪闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
RMON	Bit 13	R	RTC 跨年闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
RDATE	Bit 12	R	RTC 跨月闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
RWEEK	Bit 11	R	RTC 跨周闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
RHOUR	Bit 10	R	RTC 跨天闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
RMIN	Bit 9	R	RTC 跨小时闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
RSEC	Bit 8	R	RTC 跨分闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
AMALL	Bit 7	R	RTC 匹配所有闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
AYEAR	Bit 6	R	RTC 匹配年闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
AMON	Bit 5	R	RTC 匹配月闹钟中断有效位状态 0: 中断被禁用 1: 中断启用

ADTAE	Bit 4	R	RTC 匹配日期闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
AWEEK	Bit 3	R	RTC 匹配星期几闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
AHOUR	Bit 2	R	RTC 匹配小时闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
AMIN	Bit 1	R	RTC 匹配分钟闹钟中断有效位状态 0: 中断被禁用 1: 中断启用
ASEC	Bit 0	R	RTC 匹配秒闹钟中断有效位状态 0: 中断被禁用 1: 中断启用

22.5.2.12 RTC 无屏蔽中断标志寄存器(RTC_RIF)

RTC 无屏蔽中断标志寄存器(RTC_RIF)																															
偏移地址: 2CH																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															WKTM	F1HZ	RYEAR	RMON	RDATE	RWEEK	RHOUR	RMIN	RSEC	AMALL	AYEAR	AMON	ADATE	AWEEK	AHOUR	AMIN	ASEC

—	Bits 31-17	—	—
WKTM	Bit 16	R	RTC 唤醒计数器匹配无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
F1HZ	Bit 15	R	RTC 1 Hz 无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
RYEAR	Bit 14	R	RTC 跨世纪闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
RMON	Bit 13	R	RTC 跨年闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
RDATE	Bit 12	R	RTC 跨月闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
RWEEK	Bit 11	R	RTC 跨周闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
RHOUR	Bit 10	R	RTC 跨天闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
RMIN	Bit 9	R	RTC 跨小时闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
RSEC	Bit 8	R	RTC 跨分闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
AMALL	Bit 7	R	RTC 匹配所有闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
AYEAR	Bit 6	R	RTC 匹配年闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
AMON	Bit 5	R	RTC 匹配月闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位

ADTAE	Bit 4	R	RTC 匹配日期闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
AWEEK	Bit 3	R	RTC 匹配星期几闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
AHOUR	Bit 2	R	RTC 匹配小时闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
AMIN	Bit 1	R	RTC 匹配分钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位
ASEC	Bit 0	R	RTC 匹配秒闹钟无屏蔽中断标志状态 0: 没有中断 1: 中断正在置位

22.5.2.13 RTC 屏蔽中断标志寄存器(RTC_IFM)

RTC 屏蔽中断标志寄存器(RTC_IFM)																															
偏移地址: 30H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															WKTM	F1HZ	RYEAR	RMON	RDATE	RWEEK	RHOUR	RMIN	RSEC	AMALL	AYEAR	AMON	ADATE	AWEEK	AHOUR	AMIN	ASEC

—	Bits 31-17	—	—
WKTM	Bit 16	R	RTC 唤醒计数器匹配中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
F1HZ	Bit 15	R	RTC 1 Hz 中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
RYEAR	Bit 14	R	RTC 跨世纪闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
RMON	Bit 13	R	RTC 跨年闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
RDATE	Bit 12	R	RTC 跨月闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
RWEEK	Bit 11	R	RTC 跨周闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
RHOUR	Bit 10	R	RTC 跨天闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
RMIN	Bit 9	R	RTC 跨小时闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
RSEC	Bit 8	R	RTC 跨分闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
AMALL	Bit 7	R	RTC 匹配所有闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
AYEAR	Bit 6	R	RTC 匹配年闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
AMON	Bit 5	R	RTC 匹配月闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号

ADTAE	Bit 4	R	RTC 匹配日期闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
AWEEK	Bit 3	R	RTC 匹配星期几闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
AHOUR	Bit 2	R	RTC 匹配小时闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
AMIN	Bit 1	R	RTC 匹配分钟闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号
ASEC	Bit 0	R	RTC 匹配秒闹钟中断标志屏蔽状态 0: 没有中断或中断被屏蔽 1: 中断已发出信号

22.5.2.14 RTC 中断清除寄存器(RTC_ICR)

RTC 中断清除寄存器(RTC_ICR)																															
偏移地址: 34H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															WKTM	F1HZ	RYEAR	RMON	RDATE	RWEEK	RHOUR	RMIN	RSEC	AMALL	AYEAR	AMON	ADATE	AWEEK	AHOUR	AMIN	ASEC

—	Bits 31-17	—	—
WKTM	Bit 16	C_W1	RTC 唤醒计数器匹配中断清除 用户可以设置此位以清除中断状态
F1HZ	Bit 15	C_W1	RTC 1 Hz 中断清除 用户可以设置此位以清除中断状态
RYEAR	Bit 14	C_W1	RTC 跨世纪闹钟中断清除 用户可以设置此位以清除中断状态
RMON	Bit 13	C_W1	RTC 跨年闹钟中断清除 用户可以设置此位以清除中断状态
RDATE	Bit 12	C_W1	RTC 跨月闹钟中断清除 用户可以设置此位以清除中断状态
RWEEK	Bit 11	C_W1	RTC 跨周闹钟中断清除 用户可以设置此位以清除中断状态
RHOUR	Bit 10	C_W1	RTC 跨天闹钟中断清除 用户可以设置此位以清除中断状态
RMIN	Bit 9	C_W1	RTC 跨小时闹钟中断清除 用户可以设置此位以清除中断状态
RSEC	Bit 8	C_W1	RTC 跨分闹钟中断清除 用户可以设置此位以清除中断状态
AMALL	Bit 7	C_W1	RTC 匹配所有闹钟中断清除 用户可以设置此位以清除中断状态
AYEAR	Bit 6	C_W1	RTC 匹配年闹钟中断清除 用户可以设置此位以清除中断状态
AMON	Bit 5	C_W1	RTC 匹配月闹钟中断清除 用户可以设置此位以清除中断状态
ADTAE	Bit 4	C_W1	RTC 匹配日期闹钟中断清除 用户可以设置此位以清除中断状态
AWEEK	Bit 3	C_W1	RTC 匹配星期几闹钟中断清除 用户可以设置此位以清除中断状态
AHOUR	Bit 2	C_W1	RTC 匹配小时闹钟中断清除 用户可以设置此位以清除中断状态
AMIN	Bit 1	C_W1	RTC 匹配分钟闹钟中断清除 用户可以设置此位以清除中断状态
ASEC	Bit 0	C_W1	RTC 匹配秒闹钟中断清除 用户可以设置此位以清除中断状态

22.5.2.15 RTC 状态寄存器(RTC_STAT)

RTC 状态寄存器 (RTC_STAT)																															
偏移地址: 40H																															
复位值: 00000000_00000000_00000000_00000001B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										SYNDONE	EMPTY				

—	Bits 31-2	—	—
SYNDONE	Bit 1	R	RTC 日历同步状态 0: RTC 正在同步过程中 1: RTC 同步程序完成
EMPTY	Bit 0	R	RTC 日历状态 0 = RTC 日历不是空的 1 = RTC 日历为空

22.5.2.16 RTC 日历读取备份控制寄存器(RTC_BKEN)

RTC 日历读取备份控制寄存器(RTC_BKEN)																															
偏移地址: 44H																															
复位值: 00000000_00000000_00000000_00000000B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										BKEN					

—	Bits 31-1	—	—
BKEN	Bit 0	R/W	读取RTC备份寄存器使能 如果设置BKEN，需要较长的时间从备份寄存器来读取RTC_TIME和RTC_CAL寄存器（19 AHB CLK） 0:从控制寄存器读取RTC_TIME和RTC_CAL 1:从备份寄存器读取RTC_TIME和RTC_CAL

第23章 串行总线 (I2C)

23.1 概述

I2C 是两线双向的串行传输总线，提供了一种简单有效的方法来实现设备之间的数据交换。

I2C 标准是一个多主机总线包括冲突检测与仲裁，如果两个或两个以上的主机试图同时控制总线时，其仲裁可以防止数据损坏。在此提供了标准模式(Sm)、快速模式(Fm)与极快速模式(Fm+)供用户选择。并且也提供 SMBus(系统管理总线)与 PMBus(电源管理总线)。

23.2 特性

- ◆ 可配置为主机或从机
- ◆ 多主机模式(Multimaster capability)
- ◆ 标准模式(up to 100 kHz)
- ◆ 快速模式(up to 400 kHz)
- ◆ 极快速模式(up to 1 MHz)
- ◆ 7 位与 10 位地址模式
- ◆ 提供 2 组 7 位从机地址(2 个地址，其中一个包括屏蔽)
- ◆ 提供所有 7 位地址应答模式
- ◆ 提供广播模式(General call)
- ◆ 可编程的设置时间和保持时间
- ◆ 可选择时钟延长(Optional clock stretching)
- ◆ 可配置数字滤波器
- ◆ 提供深度为 16 字节的 TX / RX FIFOs
- ◆ 提供 DMA 传输
- ◆ 提供 SMBus 标准
- ◆ 硬件 PEC(封包错误检查)产生与 ACK 控制
- ◆ 命令与数据应答控制
- ◆ 提供地址解析协议(Address resolution protocol(ARP) support)
- ◆ 提供可选择为主控者或设备(Host and Device support)
- ◆ 提供 SMBus 警报
- ◆ 提供侦测超时与闲置功能
- ◆ 提供 PMBus rev 1.1 标准

23.3 结构图

关于 I2C 界面的结构如下图所示：

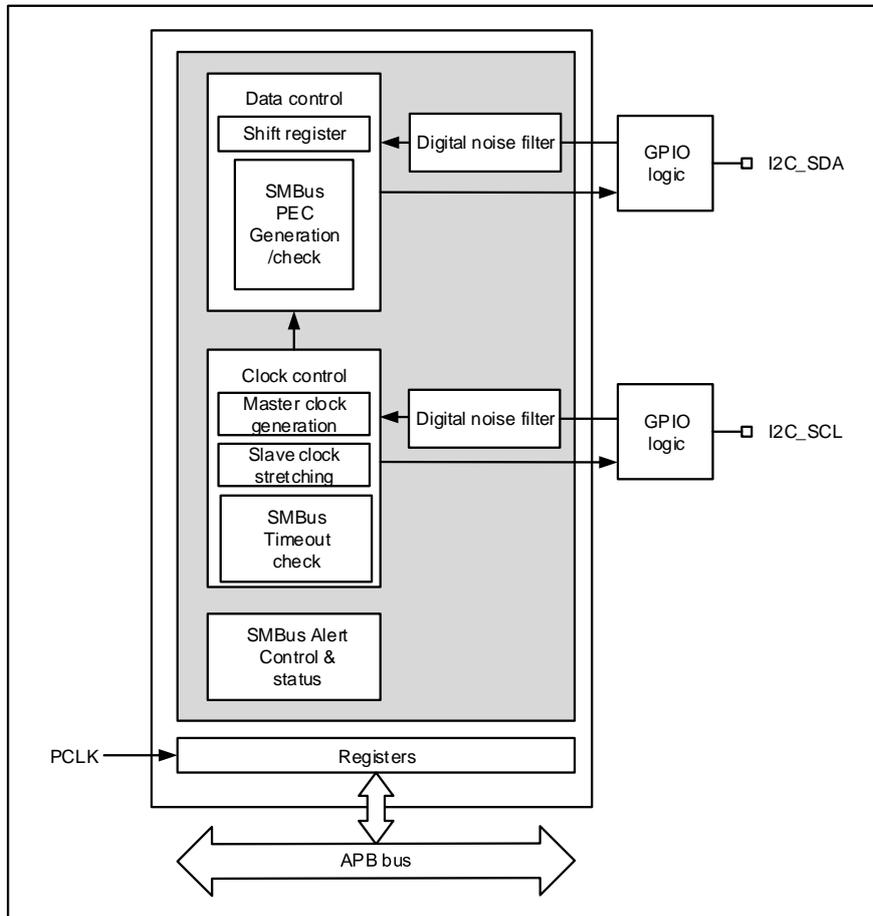


图 23-1 I2C 结构图

23.4 功能描述

除了接收和发送数据外，该接口还将其从串行格式转换为并行格式，反之亦然。中断由软件启用或禁用。该接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线。它可以连接标准模式（最高 100 kHz），快速模式（最高 400 kHz）或极快速模式（最高 1 MHz）的 I2C 总线。

该接口也可通过数据引脚（SDA）和时钟引脚（SCL）连接到 SMBus。

如果支持 SMBus 功能：还可以使用附加的可选 SMBus Alert 引脚（SMBA）。

23.4.1 I2C 总线协议

I2C 是一种双线双向串行总线，可在设备之间提供简单有效的数据交换方法。I2C 标准是真正的多主机总线，包括冲突检测和仲裁，如果两个或多个主机同时尝试控制总线，则可防止数据损坏。数据在主机和从机之间逐字节同步传输到串行数据（SDA）和串行时钟（SCL）线上，每个数据字节长度为 8 位。

23.4.1.1 START 和 STOP 条件协议

I2C 规范将启动条件定义为 SDA 线从高状态到低状态的转换，而 SCL 线为高电平。启动条件始终由主机生成，表示总线从空闲状态转换为活动状态。每个数据位有一个 SCL 时钟脉冲，首先发送 MSB。每个传送的字节后面都有一个应答位。当 SCL 为高电平时，SDA 线上的转换被解释为命令（START 或 STOP）。在 SCL 的高电平期间对每个位进行采样；因此，SDA 线可以仅在 SCL 的低周期期间改变，并且必须在 SCL 的高周期期间保持稳定。

当总线空闲时，SCL 和 SDA 信号都通过总线上的外部上拉电阻拉高。当主机想要在总线上开始传输时，主机发出 START 信号。这被定义为 SDA 信号从高到低的转换，而 SCL 为 1。当主机想要终止传输时，主机发出 STOP 条件。这被定义为 SDA 线从低到高的转换，而 SCL 为 1。下图显示了 START 和 STOP 条件的时序。当数据在总线上传输时，当 SCL 为 1 时，SDA 线必须稳定。

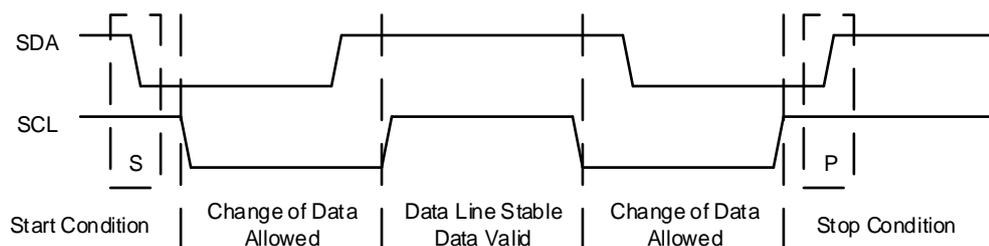


图 23-2 START 和 STOP 条件

23.4.1.2 应答位

带有应答位的数据传输是强制性的。应答位相关的时钟脉冲由主机产生。主机在应答时钟脉冲期间释放 SDA 线 (HIGH)。从机必须在应答时钟脉冲低电平期间下拉 SDA 线，以便在时钟脉冲的高电平期间保持稳定的低电平。当然，还必须考虑设置和保持时间。通常，已经被寻址的从机必须在接收到每个字节后生成应答，除非消息以 CBUS 地址开始。

当从机不应答从机地址时 (例如，由于它正在执行某些实时功能而无法接收或发送)，从机必须将数据线保持为高电平。然后，主机可以生成 STOP 条件以中止传输，或者重复 START 条件以启动新传输。

如果从机应答了从机地址，但是，传输后的某个时间不能再接收数据字节，则主机必须中止传输。这由从机在随后的第一个字节上产生非应答来指示。从机使数据线保持高电平，主机产生 STOP 或重复 START 条件。

如果主机涉及接收传输，它必须通过不在从时钟输出的最后一个字节上产生非应答来向从机发送数据结束信号。从机必须释放数据线，以允许主机产生 STOP 或重复 START 条件。

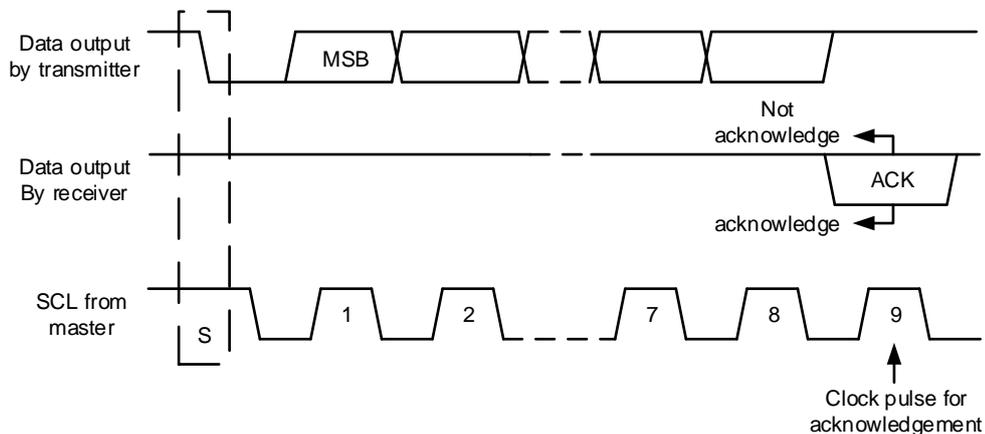
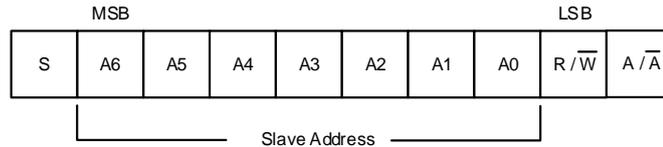


图 23-3 I2C 总线上的应答

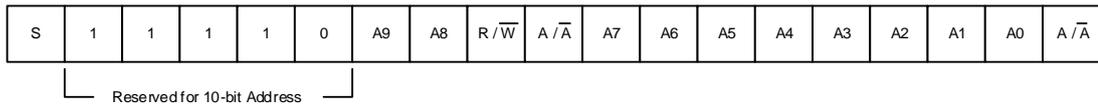
23.4.1.3 I2C 寻址从协议

有两种地址格式：7 位地址格式和 10 位地址格式。在 7 位地址格式期间，第一个字节的前 7 位（位 7：1）设置从地址，LSB 位（位 0）是 R / W 位。当位 8 设置为 0 时，主机写入从机。当位 8 设置为 1 时，主机从从机读取。数据首先传输最高有效位（MSB）。在 10 位寻址期间，传输两个字节以设置 10 位地址。第一个字节的传输包含以下位定义。前 5 位（位 7：3）通知从机这是一个 10 位传输，接着是后两位（位 2：1），它为从机地址位 9：8，LSB 位（位 8）是 R / W 位。传输的第二个字节设置从地址的位 7：0。



S = start condition
R / \bar{W} = Read / Write Pulse
A / \bar{A} = Acknowledge / not Acknowledge (Sent by slave)

图 23-4 7 位地址格式



S = start condition
R / \bar{W} = Read / Write Pulse
A / \bar{A} = Acknowledge / not Acknowledge (Sent by slave)

图 23-5 10 位地址格式

从地址	R/W位	描述
0000 000	0	广播地址
0000 000	1	START字节
0000 001	X	CBUS地址
1111 0XX	X	10位从机寻址

表 23-1 第一个字节中位的定义

23.4.2 I2C 时钟要求

I2C 内核由 I2CCLK 提供时钟。

I2CCLK 周期 T_{I2CCLK} 必须符合以下条件：

$$T_{I2CCLK} < (T_{LOW} - T_{Filter}) / 4 \text{ 与 } T_{I2CCLK} < T_{HIGH}$$

关于：

T_{LOW} : SCL 低电平时间。

T_{HIGH} : SCL 高电平时间。

T_{Filter} : 启用时，数字滤波器带来的延迟总和。数字滤波器延迟为 $DNF \times T_{I2CCLK}$ 。

PCLK 时钟周期 T_{PCLK} 必须符合以下条件：

$$T_{PCLK} < 4/3 T_{SCL}$$

T_{SCL} : SCL 周期。

23.4.3 数据传输

SDA 线上的每个字节必须为 8 位长。每次传输可以传输的字节数不受限制。每个字节后面都必须有一个应答位。首先使用最高有效位 (MSB) 传输数据。如果从机不能接收或发送另一个完整的数据字节, 则它可以时将时钟线 SCL 保持为低电平以强制主机进入等待状态, 直到它执行完了某些其他功能, 例如服务于内部中断。当从机准备好接收另一个数据字节并释放时钟线 SCL 时, 数据传输继续。

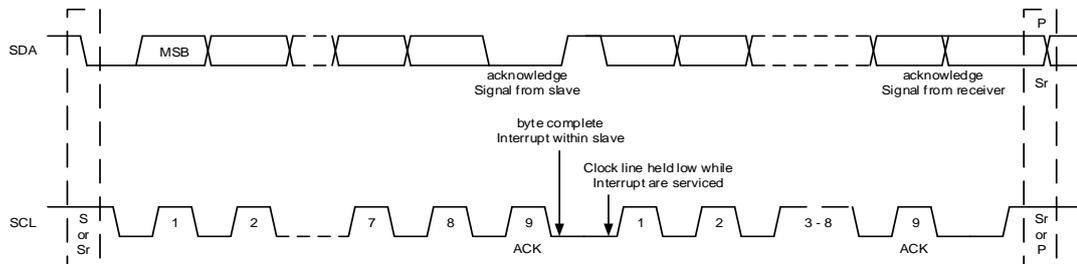


图 23-8 I2C 总线上的数据传输

接收

SDA 输入填充移位寄存器。在第 8 个 SCL 脉冲之后 (当接收到完整的数据字节时), 如果接收 FIFO 尚未满 (RXF = 0), 则将其复制到接收 FIFO 中。如果 RXF = 1, 意味着尚未读取先前接收的数据字节, 此时 SCL 线会在第 9 个 SCL 脉冲之后延长, 直到读取 I2C_RXDATA, 使得接收 FIFO 尚未满 (RXF = 0)。

传输

如果发送 FIFO 不为空 (TXE = 0), 则第 9 个 SCL 脉冲 (应答脉冲) 之后将其内容复制到移位寄存器中。然后移位寄存器的内容会在 SDA 线上发送。如果 TXE = 1, 意味着发送 FIFO 尚未写入数据, SCL 线将被拉低, 直到通过写入 I2C_TXDATA 到发送 FIFO。SCL 延长在第 9 个 SCL 脉冲之后完成。

硬件传输管理

I2C 具有嵌入在硬件中的字节计数器，以便管理字节传输并以各种模式关闭通信，例如：在主模式下生成 NACK, STOP 和 RESTART，从接收器模式下的 ACK 控制，支持 SMBus 功能时的 PEC 生成/检查。

字节计数器始终在主模式下使用。默认情况下，它在从机模式下禁用，但可以通过设置 I2C_CON1 寄存器中的 SBC(从机字节控制)位由软件启用。要传输的字节数在 I2C_CON2 寄存器的 NBYTES [7: 0]位字段中编程。如果要传输的字节数 (NBYTES) 大于 255，或者接收时想要控制接收数据字节的应答值，则必须通过将 I2C_CON2 寄存器中的 RELOAD 位置 1 来选择重载模式。在此模式下，当传送到 NBYTES 中所编程的字节数时，TCR 标志置位，如果 TCRIE 置位，则产生中断。只要设置了 TCR 标志，SCL 就会被延长。当软件对 NBYTES 写入非零值时，会清除 TCR。

当 NBYTES 计数器重新加载最后一个字节数时，必须清除 RELOAD 位。

当主模式下 RELOAD = 0 时，计数器可用于 2 种模式：

- ◆ **自动结束模式** (I2C_CON2 寄存器中的 AUTOEND = '1')。在此模式下，一旦传输了 NBYTES [7: 0]位字段中编程的字节数，主机就会自动发送 STOP 条件。
- ◆ **软件结束模式** (I2C_CON2 寄存器中的 AUTOEND = '0')。在此模式下，一旦传输了 NBYTES [7: 0]位字段中编程的字节数，就会产生软件操作；如果 TCIE 位置 1，则 TC 标志置 1，产生中断。只要 TC 标志置位，SCL 信号就会被拉长。当 I2C_CON2 寄存器中的 START 或 STOP 位置 1 时，TC 标志由软件清零。当主机要发送 RESTART 条件时，必须使用此模式。

23.4.4 I2C 从机模式

I2C 从机初始化

为了在从机模式下工作，用户必须至少启用一个从机地址。两个寄存器 **I2C_ADDR1** 和 **I2C_ADDR2** 可用于编程从机自身地址 **OA1** 和 **OA2**。

- ◆ 通过将 **I2C_ADDR1** 寄存器中的 **OA1MODE** 位置 1，可以在 7 位模式（默认情况下）或 10 位寻址模式下配置 **OA1**。通过将 **I2C_ADDR1** 寄存器中的 **OA1EN** 位置 1 启用 **OA1**。
- ◆ 如果需要额外的从地址，可以配置第二个从地址 **OA2**。通过配置 **I2C_ADDR2** 寄存器中的 **OA2MSK [2: 0]** 位，可以屏蔽最多 7 个 **OA2 LSB**。因此，对于配置为 1 至 6 的 **OA2MSK**，仅 **OA2 [7: 2]**，**OA2 [7: 3]**，**OA2 [7: 4]**，**OA2 [7: 5]**，**OA2 [7: 6]** 或 **OA2 [7]** 与收到的地址相比较。一旦 **OA2MSK** 不等于 0，**OA2** 的地址比较器就会排除未被应答的 I2C 保留地址（0000 XXX 和 1111 XXX）。如果 **OA2MSK = 7**，则应答所有接收的 7 位地址（保留地址除外）。**OA2** 始终是 7 位地址。

当 **OA2MSK = 0** 的情况下且在 **I2C_ADDR1** 或 **I2C_ADDR2** 寄存器中编程特定启用位置，则可应答这些保留地址。通过将 **I2C_ADDR2** 寄存器中的 **OA2EN** 位置 1 启用 **OA2**。

- ◆ 通过将 **I2C_CON1** 寄存器中的 **GCEN** 位置 1 启用广播地址。

当其中一个启用地址选择 I2C 时，地址匹配中断标志状态置 1，如果 **ADDRIE** 位置 1，则产生中断。

默认情况下，从机使用其时钟延长功能，这意味着它在需要时将 **SCL** 信号拉伸到低电平，以执行软件操作。如果主机不支持时钟延长，则必须在 **I2C_CON1** 寄存器中将 I2C 配置为 **NOSTRETCH = 1**。

收到地址匹配中断后，如果启用了多个地址，用户必须读取 **I2C_STAT** 寄存器中的 **ADDCODE [6: 0]** 位，以检查匹配的地址。还必须检查 **DIR** 标志以了解传输方向。

从机时钟延长（NOSTRETCH = 0）

在默认模式下，I2C 从机在以下情况下延长 **SCL** 时钟：

- ◆ 在传输过程中，如果先前的数据传输完成且没有在 **I2C_TXDATA** 寄存器中写入新数据。当数据写入 **I2C_TXDATA** 寄存器时，将释放此延长。
- ◆ 在接收时，**I2C_RXDATA** 寄存器尚未读取且接收 FIFO 已满。读取 **I2C_RXDATA** 且接收 FIFO 未滿时，将释放此延长。
- ◆ 从机字节控制模式下，当发生 **TCR = 1** 时，重载模式（**SBC = 1** 且 **RELOAD = 1**），表示数据字节已被传输。此时通过在 **NBYTES [7: 0]** 字段中写入非零值清除 **TCR** 时，释放该延长。
- ◆ 从机应答控制模式下，当收到地址或数据时，每个字节的第 8 和第 9 个 **SCL** 脉冲之间会将 **SCL** 延长。当设置应答更新时，将会释放该延长并回应应答脉冲。
- ◆ 在 **SCL** 下降沿检测后，I2C 拉低 **SCL** 延长单位为
$$[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times T_{I2CCLK}$$

从机没有时钟延长（NOSTRETCH = 1）

当 I2C_CON1 寄存器中的 NOSTRETCH = 1 时，I2C 从机不会延长 SCL 信号。

- ◆ 发送时，必须在与传输相对应的第一个 SCL 脉冲之前将数据写入 I2C_TXDATA 寄存器。如果不是，则发生欠载运算，在 I2C_STAT 寄存器中设置 TXUD 标志，如果 I2C_IER 寄存器中的 TXUDIE 位置 1，则会产生中断。
- ◆ 接收时，如果接收 FIFO 满时，必须在下一个数据字节的第 9 个 SCL 脉冲（应答脉冲）之前从 I2C_RXDATA 寄存器中读取数据。如果发生溢出，则在 I2C_STAT 寄存器中设置 RXOV 标志，如果 I2C_IER 寄存器中的 RXOVIE 位置 1，则会产生中断。

从机字节控制模式

为了在从机接收模式下允许字节 ACK 控制，必须通过将 I2C_CON1 寄存器中的 SBC 位置 1 来启用从机字节控制模式。这需要符合 SMBus 标准。

必须选择重载模式，以便在从机接收模式（RELOAD = 1）中允许字节控制。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每个接收到的字节后重新加载到 0x1。当接收到该字节时，TCR 位置 1，在第 9 个 SCL 脉冲之后将 SCL 信号拉低。用户可以从 I2C_RXDATA 寄存器读取数据，然后通过配置 I2C_CON2 寄存器中的 NACK 位决定下一个数据是否应答。通过将 NBYTES 编程为非零值来释放 SCL 延伸：发送应答或不应答。

NBYTES 可以加载大于 0x1 的值，在这种情况下，接收流程在 NBYTES 数据接收期间是连续的。

注释：当禁用 I2C 时，才可配置 SBC 位。当 TCR = 1 时，此时可以更改 RELOAD 位值。

警告：从机字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH = 1 时设置 SBC。

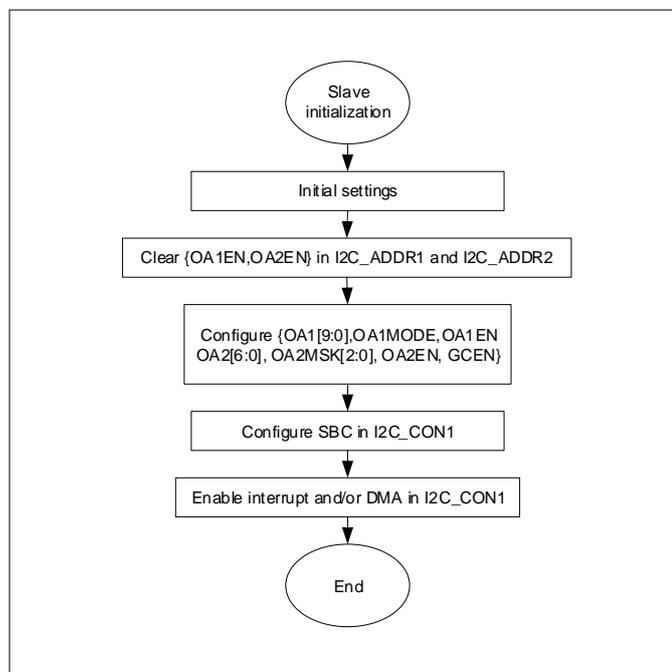


图 23-9 从机初始化流程图

从机传送

在接收到匹配的地址时，I2C_RIF 寄存器中 ADDRRI 位会置位，此时若 TX FIFO 中已准备好要发送的数据时，从机会通过内部移位寄存器将 TX FIFO 中的字节发送到 SDA 线。若 TX FIFO 此时为空，从机会延长 SCL 低电平时间，直到 TX FIFO 透过 I2C_TXDATA 寄存器写入发送数据为止。

当发送成功，主机会回答信号，当主机回答 NACK 时，此时 I2C_RIF 寄存器中 NACKRI 位会置位，此时从机会自动释放 SCL 与 SDA 总线让主机能够发送后续的 STOP 或 RESTART 命令。

当主机回答 STOP 时，此时 I2C_RIF 寄存器中 STOPRI 位会置位，并结束通信，等待下一次收到匹配的地址。然而，若 TX FIFO 内还有尚未传送的字节，使用者可以选择下一次地址匹配时，继续传送 FIFO 中的数据，或是对 I2C_FCON 寄存器中 TXFRST 位置位，来清除 FIFO 中的字节，下一次传送新的字节。

当从机字节控制启动时，需要传送多少字节需在 I2C_CON2 寄存器中 NBYTES 内配置。

警告：当 NOSTRETCH 模式启动时，由于 SCL 时钟无法被延长，因此需要传送的字节需要被提前写入 TX FIFO 内。

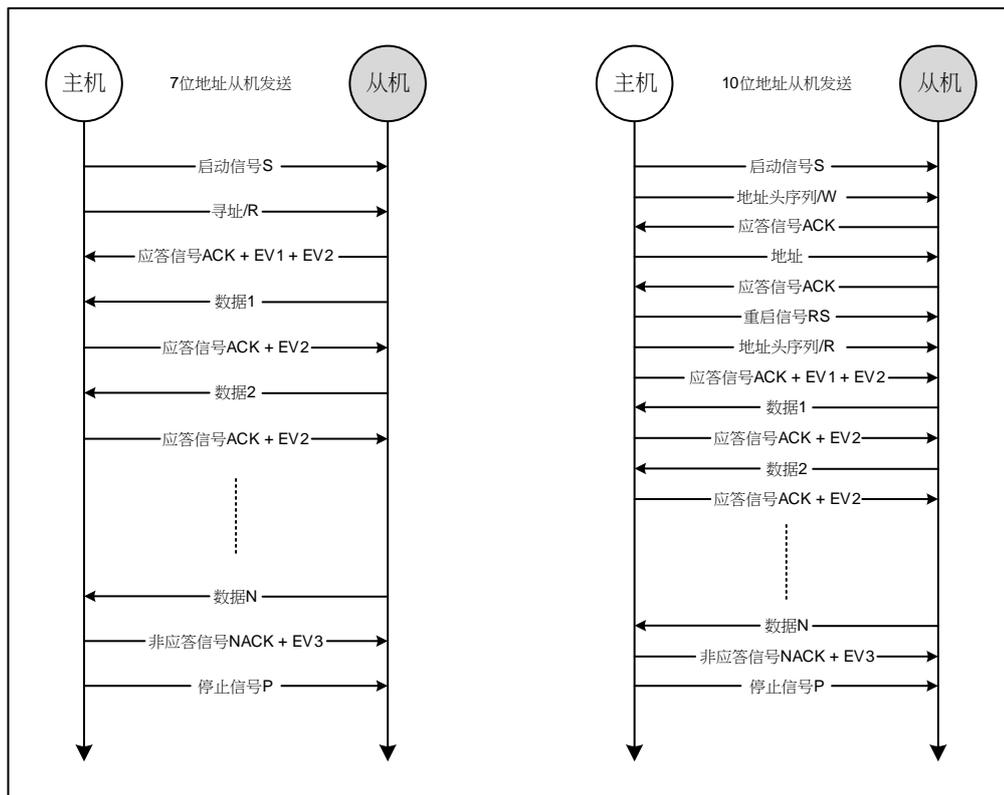


图 23-10 从机发送的传输序列图

注 1: S = 起始位, RS = 重复起始位, P = 停止位, ACK = 应答, NACK = 非应答
 注 2: EV1 = 当收到匹配地址时, I2C_RIF.ADDRRI = 1, 对 I2C_ICR.ADDRIC 置位来清除中断。
 注 3: EV2 = 判断 I2C_STAT.TXTH, 若低于水平则对 I2C_TXDATA 写入数据。
 注 4: EV3 = 当收到 NACK 时, I2C_RIF.NACKRI = 1, 对 I2C_ICR.NACKIC 置位来清除中断。此时还需

判断 TX FIFO 内是否还有尚未发送的字节，若有则软件需要额外处理。
注 5: 如果软件列在当前传送字节传输结束之前尚未写入下一个字节，导致 TX FIFO 为空时，EV2 事件将会延长 SCL 时钟低电平时间。

从机接收

在接收到匹配的地址时，I2C_RIF 寄存器中 ADDRRI 位会置位，从机此时会通过内部移位寄存器将 SDA 总线的字节保存到 RX FIFO 中，无论 RX FIFO 是否满，从机都会自动发送应答信号，差异仅在应答信号发送后，FIFO 满时会延长 SCL 时钟低电平时间，等待软件将 RX FIFO 中的字节取出后，才会释放 SCL 时钟。

当主机发送 STOP 时，此时 I2C_RIF 寄存器中 STOPRI 位会置位，并结束通信，等待下一次收到匹配的地址。

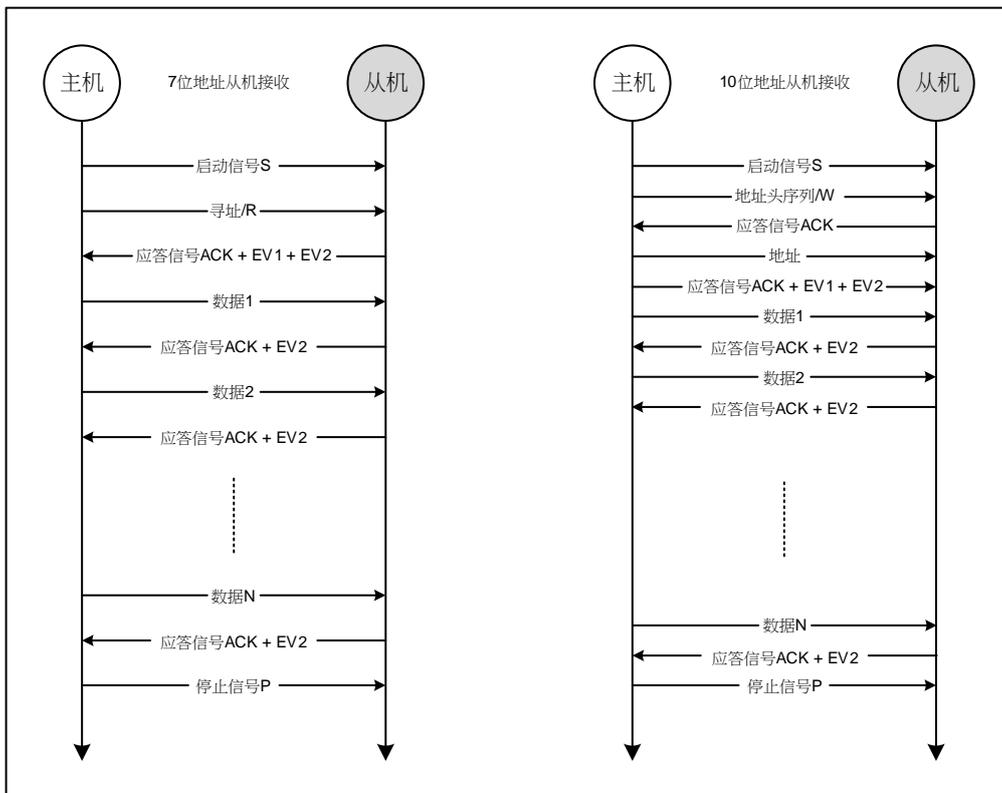


图 23-11 从机接收的传输序列图

注 1: S = 起始位，RS = 重复起始位，P = 停止位，ACK = 应答，NACK = 非应答
注 2: EV1 = 当收到匹配地址时，I2C_RIF.ADDRRI = 1，对 I2C_ICR.ADDRIC 置位来清除中断。
注 3: EV2 = 判断 I2C_STAT.RXTH，若高于水平则对 I2C_RXDATA 取出数据。
注 4: 如果软件列在当前传送字节传输结束时，导致 RX FIFO 满，EV2 事件将会延长 SCL 时钟低电平时间。

23.4.5 I2C 主机模式

I2C 主机初始化

在启用外设之前，必须通过将 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位置 1 来配置 I2C 主时钟。实现时钟同步机制以支持多主机环境和从机时钟延长。

为了允许时钟同步：

- ◆ 从 SCL 低电平内部检测开始，使用 SCLL 计数器计数低电平时钟。
- ◆ 从 SCL 高电平内部检测开始，使用 SCLH 计数器计数高电平时钟。

根据 SCL 下降沿，I2C 在 T_{SYNC} 延迟后检测到自己的 SCL 低电平。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器中 SCLL [7: 0]位中设置的值，I2C 就会将 SCL 释放为高电平，SCL 输入数字噪声滤波器和 SCL 与 I2CxCLK 时钟同步。

在 T_{SYNC} 产生延迟后，I2C 会检测自己的 SCL 高电平，具体取决于 SCL 上升沿，SCL 输入数字噪声滤波器和 SCL 与 I2CxCLK 时钟的同步。

一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器中的 SCLH [7: 0]位编程的值，I2C 就会将 SCL 置为低电平。

因此，主时钟周期为：

$$T_{SCL} = T_{SYNC1} + T_{SYNC2} + \{ [(SCLH+1) + (SCLL+1)] \times (PRESC+1) \times T_{I2CCLK} \}$$

T_{SYNC1} 的持续时间取决于以下参数：

- ◆ SCL 下降斜率
- ◆ 启用时，数字滤波器引起的输入延迟：DNF x T_{I2CCLK}
- ◆ 由于 SCL 与 I2CCLK 时钟同步（2 至 3 个 I2CCLK 周期）导致的延迟

T_{SYNC2} 的持续时间取决于以下参数：

- ◆ SCL 上升斜率
- ◆ 启用时，数字滤波器引起的输入延迟：DNF x T_{I2CCLK}
- ◆ 由于 SCL 与 I2CCLK 时钟同步（2 至 3 个 I2CCLK 周期）导致的延迟

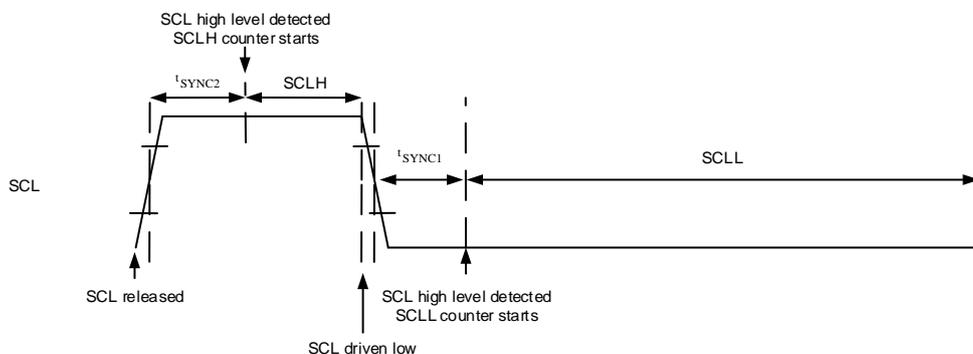


图 23-12 主时钟产生

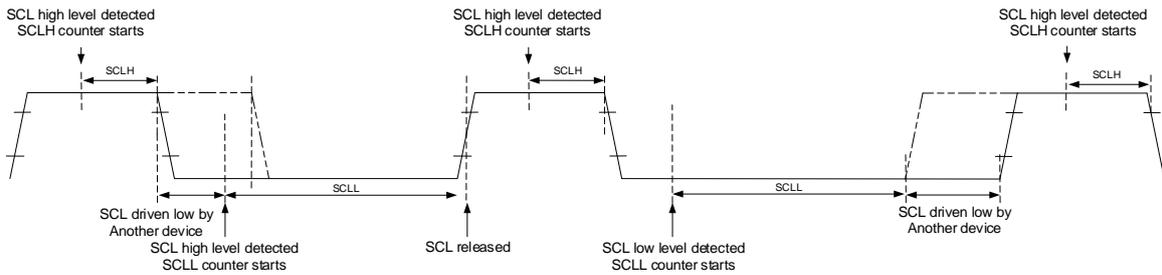


图 23-13 SCL 主时钟同步

主机通信初始化（地址阶段）

为了启动通信，用户必须为 **I2C_CON2** 寄存器中寻址的从机编程以下参数：

- ◆ 寻址模式（7 位或 10 位）：**ADD10**
- ◆ 要发送的从站地址：**SADD [9: 0]**
- ◆ 传输方向：**RD_WRN**
- ◆ 如果是 10 位地址读取：**HEAD10R** 位。必须配置 **HEAD10R** 以指示是否必须发送完整的地址序列，或者仅在方向改变时发送地址标头。
- ◆ 要传输的字节数：**NBYTES [7: 0]**。如果字节数等于或大于 255 个字节，则 **NBYTES [7: 0]** 最初必须填充 0xFF。

然后，用户必须将 **I2C_CON2** 寄存器中的 **START** 位置 1。当 **START** 位置 1 时，不允许更改所有上述位。一旦检测到总线空闲（**BUSY = 0**）后，主机就会自动发送 **START** 条件，再发送从机地址。

在仲裁丢失的情况下，主机自动切换回从机模式，如果它作为寻址从机，则可以应答自己的地址。

注：

无论接收到的应答值是什么，当总线上的从机地址发送时，**START** 位由硬件复位。如果发生仲裁丢失，则 **START** 位也由硬件复位。在 10 位寻址模式下，当从机地址前 7 位被从机 **NACK** 时，主机将自动重新启动从机地址传输，直到收到 **ACK**。如果在 **START** 位置 1 时 **I2C** 被寻址为从机（**ADDRRI = 1**），则 **I2C** 切换到从机模式，当 **ADDRRI** 位置 1 时，**START** 位清零。

注：

对重复启动条件应用相同的过程。在这种情况下，**BUSY = 1**。

初始化主机 10 位从地址寻址接收模式

- ◆ 如果从机地址采用 10 位格式，用户可以通过清零 **I2C_CON2** 寄存器中的 **HEAD10R** 位来选择发送完整的读序列。在这种情况下，主机在 **START** 位置 1 后自动发送以下完整序列：启动+从机地址 10 位标头写+从机地址第 2 字节+重新启动+从机地址 10 位标头读取；
- ◆ 如果主机寻址 10 位地址从机，将数据发送到该从机，然后从同一从机读取数据，则必须先完成主机传输流程，然后使用 **HEAD10R = 1** 配置的 10 位从地址设置重复启动。在这种情况下，主机发送此序列：重新启动+从机地址 10 位标头读取。

主机传送

在主机开始传送之前,需先配置 **I2C_CON2** 寄存器中 **NBYTES** 位,当传输字节大于 255 时,需要额外配置 **RELOAD** 位。在此配置下,当 **NBYTES** 配置的笔数传送完成后, **I2C_RIF** 寄存器中 **TCRRI** 位会置位,此时 **SCL** 时钟会保持低电平,直到 **NBYTES** 位重新写入新的数值以及对 **I2C_TXDATA** 寄存器写入发送数据后,才会继续进行传输。

当从机回应 **NACK**, **I2C_RIF** 寄存器中 **NACKRI** 位会置位,并且接下来主机会自动发送停止信号 **STOP**。

- ◆ 当从机响应 **ACK**, 且 **RELOAD = 0**、**NBYTES** 所配置的笔数都已经传完时,会有以下情况: 若 **AUTOEND = 1**, 此时会自动发送停止讯号 **STOP**。
- ◆ 若 **AUTOEND = 0**, 此时主机会将 **SCL** 时钟保持低电平, 等待软件控制后续操作:
RESTART: 对 **I2C_CON2** 寄存器中 **START** 置位, 此时会发送重启信号。
STOP: 对 **I2C_CON2** 寄存器中 **STOP** 置位, 此时会发送停止信号。

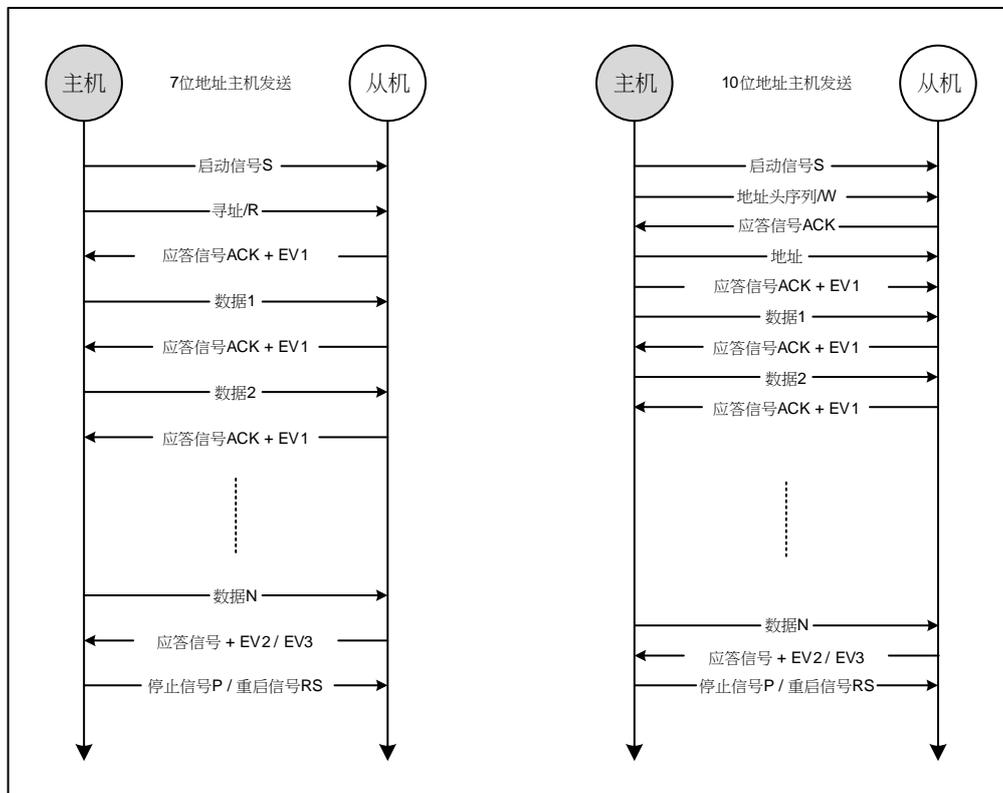


图 23-14 主机发送的传输序列图

- 注 1: S = 起始位, RS = 重复起始位, P = 停止位, ACK = 应答, NACK = 非应答
- 注 2: EV1 = 传输尚未完成事件, 判断 **I2C_STAT.TXTH**, 若低于水平则对 **I2C_TXDATA** 写入数据。
- 注 3: EV2 = 当应答信号为 **ACK** 时, 若传输已完成, 后续操作为停止信号 **STOP** 或重启信号 **RESTART**。
- 注 4: EV3 = 当应答信号为 **NACK** 时, 后续操作为停止信号 **STOP**。
- 注 5: 如果软件列在当前传送字节传输结束之前尚未写入下一个字节, 导致 **TX FIFO** 为空时, **EV1** 事件将会延长 **SCL** 时钟低电平时间。

主机接收

在主机开始接收之前，需先配置 **I2C_CON2** 寄存器中 **NBYTES** 位，当传输字节大于 255 时，需要额外配置 **RELOAD** 位。在此配置下，当 **NBYTES** 配置的笔数传送完成后，**I2C_RIF** 寄存器中 **TCRRI** 位会置位，此时 **SCL** 时钟会保持低电平，直到 **NBYTES** 位重新写入新的数值后，才会继续进行传输。

当 **RELOAD = 0** 并且 **NBYTES** 所配置的笔数都已经传完时，会有以下情况：

- ◆ 若 **AUTOEND = 1**，此时会自动发送非应答信号 **NACK** 与停止讯号 **STOP**。
- ◆ 若 **AUTOEND = 0**，此时会自动发送非应答信号 **NACK**，并将 **SCL** 时钟保持低电平，等待软件控制后续操作：

RESTART：对 **I2C_CON2** 寄存器中 **START** 置位，此时会发送重启信号。

STOP：对 **I2C_CON2** 寄存器中 **STOP** 置位，此时会发送停止信号。

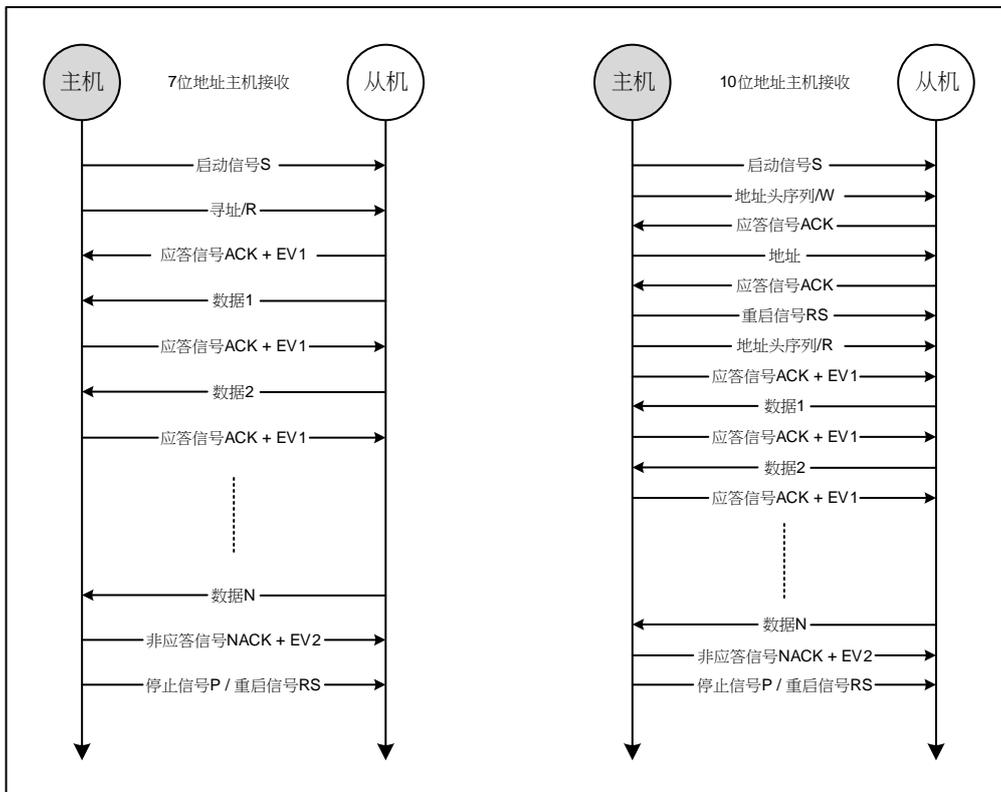


图 23-15 主机接收的传输序列图

- 注 1: S = 起始位, RS = 重复起始位, P = 停止位, ACK = 应答, NACK = 非应答
- 注 2: EV1 = 传输尚未完成事件, 判断 **I2C_STAT.RXTH**, 若高于水平则对 **I2C_RXDATA** 读出数据。
- 注 3: EV2 = 传输完成事件, 此时会自动发送非应答信号 **NACK**, 后续根据软件配置来决定是发送停止信号 **STOP** 还是重启信号 **RESTART**。
- 注 4: 如果软件列在当前传送字节传输结束之前尚读出数据, 导致 **RX FIFO** 满时, **EV1** 事件将会延长 **SCL** 时钟低电平时间。

23.4.6 I2C_TIMINGR 寄存器的配置的例子

下表提供了如何对 I2C_TIMINGR 进行编程以获得符合 I2C 规范的时序的示例。

参数	标准模式(Sm)		快速模式(Fm)	极快速模式(Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
T _{SCLL}	200x250 ns = 50 μs	20x250 ns = 5.0 μs	10x125 ns = 1250 ns	7x125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
T _{SCLH}	196x250 ns = 49 μs	16x250 ns = 4.0 μs	4x125ns = 500ns	4x125ns = 500ns
T _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽²⁾	~2500 ns ⁽³⁾	~2000 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x1	0x0
T _{SDADEL}	2x250 ns = 500 ns	2x250 ns = 500 ns	1x125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
T _{SCLDEL}	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

表 23-2 F_{I2CCLK} = 8 MHz 的时序设置示例

1. 由于 SCL 内部检测延迟，SCL 周期 T_{SCL} 大于 T_{SCLL} + T_{SCLH}。为 T_{SCL} 提供的值仅为示例。
2. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 500 ns。T_{SYNC1} + T_{SYNC2} = 1000 ns 的示例。
3. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 500 ns。T_{SYNC1} + T_{SYNC2} = 750 ns 的示例。
4. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 500 ns。T_{SYNC1} + T_{SYNC2} = 655 ns 的示例。

参数	标准模式(Sm)		快速模式(Fm)	极快速模式(Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
T _{SCLL}	200x250 ns = 50 μs	20x250 ns = 5.0 μs	10x125 ns = 1250 ns	6x62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
T _{SCLH}	196x250 ns = 49 μs	16x250 ns = 4.0 μs	4x125ns = 500ns	3x62.5ns = 187.5ns
T _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽²⁾	~2500 ns ⁽³⁾	~1000 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x2	0x0
T _{SDADEL}	2x250 ns = 500 ns	2x250 ns = 500 ns	2x125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
T _{SCLDEL}	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	3x62.5 ns = 187.5 ns

表 23-3 F_{I2CCLK} = 16 MHz 的时序设置示例

1. 由于 SCL 内部检测延迟，SCL 周期 T_{SCL} 大于 T_{SCLL} + T_{SCLH}。为 T_{SCL} 提供的值仅为示例。
2. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 250 ns。T_{SYNC1} + T_{SYNC2} = 1000 ns 的示例。
3. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 250 ns。T_{SYNC1} + T_{SYNC2} = 750 ns 的示例。
4. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 250 ns。T_{SYNC1} + T_{SYNC2} = 500 ns 的示例。

参数	标准模式(Sm)	快速模式(Fm)	极快速模式(Fm+)
----	----------	----------	------------

	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
T _{SCLL}	200x250 ns = 50 μs	20x250 ns = 5.0 μs	10x125 ns = 1250 ns	4x125 ns = 500 ns
SCLH	0xC3	0xF	0x3	0x1
T _{SCLH}	196x250 ns = 49 μs	16x250 ns = 4.0 μs	4x125 ns = 500 ns	2x125 ns = 250 ns
T _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽²⁾	~2500 ns ⁽³⁾	~875 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x3	0x0
T _{SDADEL}	2x250 ns = 500 ns	2x250 ns = 500 ns	3x125 ns = 375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
T _{SCLDEL}	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

表 23-4 F_{I2CCLK} = 48 MHz 的时序设置示例

1. 由于 SCL 内部检测延迟，SCL 周期 T_{SCL} 大于 T_{SCLL} + T_{SCLH}。为 T_{SCL} 提供的值仅为示例。
2. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 83.3 ns。T_{SYNC1} + T_{SYNC2} = 1000 ns 的示例。
3. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 83.3 ns。T_{SYNC1} + T_{SYNC2} = 750 ns 的示例。
4. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 83.3 ns。T_{SYNC1} + T_{SYNC2} = 250 ns 的示例。

23.4.7 SMBus 具体功能

介绍

系统管理总线（SMBus）是一个双线接口，各种设备可以通过该接口相互通信并与系统的其余部分通信。它基于 I2C 操作原理。SMBus 为系统和电源管理相关任务提供控制总线。

该外设与 SMBUS 规范 rev 2.0 (<http://smbus.org>) 兼容。

系统管理总线规范指的是三种类型的设备。

- ◆ 从机是接收或响应命令的设备。
- ◆ 主设备是发出命令，生成时钟并终止传输的设备。
- ◆ 主控者是专用主机，为系统的 CPU 提供主接口。主控者必须可当作是主机或从机，并且必须支持 SMBus 主机通信协议。

系统中只允许一个主控者。该外设可以配置为主设备或从设备，也可以配置为主控者。

SMBUS 基于 I2C 规范 rev 2.1。

总线协议

对于任何给定的设备，有 11 种可能的命令协议。设备可以使用 11 个协议中的任何一个或全部来进行通信。协议包括快速命令，发送字节，接收字节，写入字节，写入字，读取字节，读取字，进程调用，块读取，块写入和块写入块读取进程调用。这些协议应由用户软件实现。有关这些协议的更多详细讯息，请参阅 SMBus 规范 2.0 版 (<http://smbus.org>)。

地址解析协议（ARP）

可以通过为每个从设备动态分配新的唯一地址来解决 SMBus 从地址冲突。为了提供隔离每个设备以便进行地址分配的机制，每个设备必须实现唯一的设备标识符（UDID）。这个 128 位数字由软件实现。该外设支持地址解析协议（ARP）。通过将 I2C_CON1 寄存器中的 SMBDEN 位置 1 启用 SMBus 从设备默认地址（0b1100 001）。ARP 命令应由用户软件实现。仲裁也在从属模式下执行以支持 ARP。

有关 SMBus 地址解析协议的更多详细讯息，请参阅 SMBus 规范 2.0 版 (<http://smbus.org>)。

收到命令和数据应答控制

SMBus 接收器必须能够 NACK 每个接收到的命令或数据。为了在从机模式下允许 ACK 控制，必须通过将 I2C_CON1 寄存器中的 SBC 位置 1 来启用从机字节控制模式。

主控者通知协议

该外设通过将 I2C_CON1 寄存器中的 SMBHEN 位置 1 来支持主控者通知协议。在这种情况下，主控者将应答 SMBus 主控者地址（0b0001 000）。使用此协议时，设备充当主机，主控者充当从机。

SMBus 警报

支持 SMBus ALERT 可选信号。仅从设备可以通过它想要通话的 SMBALERT# 引脚向主控者发送信号。主控者处理中断并同时通过警报响应地址 (0b0001 100) 访问所有 SMBALERT# 设备。只有拉低 SMBALERT# 的设备才会应答警报响应地址。当配置为从设备 (SMBHEN = 0) 时, 通过将 I2C_CON1 寄存器中的 ALERTEN 位置 1, 可将 SMBA 引脚拉低。警报响应地址同时启用。当配置为主控者 (SMBHEN = 1) 时, 如果在 SMBA 引脚上检测到下降沿且 ALERTEN = 1, 则在 I2C_RIF 寄存器中设置 ALERT 标志。如果 I2C_IER 寄存器中的 ALERTIE 位置 1, 则会产生中断。当 ALERTEN = 0 时, 即使外部 SMBA 引脚为低电平, ALERT 线也会被视为高电平。如果不需要 SMBus ALERT 引脚, 如果 ALERTEN = 0, 则 SMBA 引脚可用作标准 GPIO。

数据包错误检查

SMBus 规范中引入了一种数据包错误检查机制, 以提高可靠性和通信稳健性。通过在每次消息传输结束时附加分组错误代码 (PEC) 来实现分组错误检查。通过在所有消息字节 (包括地址和读/写位) 上使用 $C(x) = X^8 + X^2 + X + 1$ CRC-8 多项式来计算 PEC。外设嵌入了硬件 PEC 计算器, 当接收到的字节与硬件计算的 PEC 不匹配时, 允许自动发送非应答。

超时

该外设嵌入了硬件定时器, 以符合 SMBus 规范 2.0 版中定义的 3 个超时。

标记	参数	范围		单位
		最小	最大	
T _{TIMEOUT}	检测时钟低超时	25	35	ms
T _{LOW : SEXT} ⁽¹⁾	累积时钟低延长时间 (从设备)	-	25	ms
T _{LOW : MEXT} ⁽²⁾	累积时钟低延长时间 (主设备)	-	10	ms

表 23-5 SMBus 超时规格

1. T_{LOW : SEXT} 是允许给定从设备在从初始 START 到 STOP 的一条消息中延长时钟周期的累积时间。另一个从设备或主设备也可能延长时钟, 导致组合时钟低延长时间大于 T_{LOW : SEXT}。因此, 该参数是在从设备作为全速主设备的唯一目标的情况下测量的。
2. T_{LOW : MEXT} 是允许主设备在从 START 到 ACK, ACK 到 ACK 或 ACK 到 STOP 定义的消息的每个字节内延长其时钟周期的累积时间。从设备或另一个主设备也可能延长时钟, 导致组合时钟低电平时间大于给定字节上的 T_{LOW : MEXT}。因此, 使用全速从设备作为主设备的唯一目标来测量该参数。

总线空闲检测

如果总线检测到时钟和数据信号已经持续高电平并且 T_{IDLE} 大于 T_{HIGH.MAX}, 则主设备可以假设总线是空闲的。

该时序参数涵盖了主机已动态添加到总线并且可能未检测到 SMBCLK 或 SMBDAT 线路上的状态转换的情况。在这种情况下, 主设备必须等待足够长的时间以确保当前没有进行传输。外设支持硬件总线空闲检测。

23.4.8 SMBus 初始化

除了 I2C 初始化之外，还必须进行一些其他特定的初始化以执行 SMBus 通信：

接收命令和数据应答控制（从机模式）

SMBus 接收器必须能够 NACK 每个接收到的命令或数据。为了在从机模式下允许 ACK 控制，必须通过将 I2C_CON1 寄存器中的 SBC 位置 1 来启用从机字节控制模式。

特定地址（从机模式）

如果需要，应启用特定的 SMBus 地址。

通过将 I2C_CON1 寄存器中的 SMBDEN 位置 1 启用 SMBus 设备从机地址(0b1100 001)。

通过将 I2C_CON1 寄存器中的 SMBHEN 位置 1 启用 SMBus 主控者从机地址(0b0001 000)。

通过将 I2C_CON1 寄存器中的 ALERTEN 位置 1 启用报警响应地址 (0b0001100)。

数据包错误检查

通过将 I2C_CON1 寄存器中的 PECEN 位置 1 启用 PEC 计算。然后在硬件字节计数器的帮助下管理 PEC 传输：I2C_CON2 寄存器中的 NBYTES [7: 0]。必须先配置 PECEN 位，然后再启用 I2C。

PEC 传输由硬件字节计数器管理，因此在从机模式下连接 SMBus 时必须设置 SBC 位。在 PECBYTE 位置 1 且 RELOAD 位清零后，在传输 NBYTES-1 数据后传输 PEC。如果设置了 RELOAD，则 PECBYTE 无效。

注意：启用 I2C 时，不允许更改 PECEN 配置。

超时检测

通过将 **I2C_TIMEOUTR** 寄存器中的 **TIMOUTEN** 和 **TEXTEN** 位置 1 来启用超时检测。定时器必须以这样的方式编程，即它们在 **SMBus** 规范版本 2.0 中给出的最大时间之前检测到超时。

◆ T_{TIMEOUT} 检查

为了启用 T_{TIMEOUT} 检查，必须对 12 位 **TIMEOUTA** 定时器进行编程，以检查 T_{TIMEOUT} 参数。必须将 **TIDLE** 位配置为“0”才能检测 **SCL** 低电平超时。然后通过设置 **I2C_TIMEOUTR** 寄存器中的 **TIMOUTEN** 来启用定时器。如果 **SCL** 在大于 $(\text{TIMEOUTA} + 1) \times 2048 \times T_{\text{I2CCCLK}}$ 的时间内被拉低，则在 **I2C_RIF** 寄存器中设置 **TOUTRI** 标志。

注意：当 **TIMEOUTEN** 位置 1 时，不允许更改 **TIMEOUTA** [11: 0]位和 **TIDLE** 位配置。

◆ $T_{\text{LOW:SEXT}}$ 和 $T_{\text{LOW:MEXT}}$ 检查

根据外设是配置为主机还是从机，必须配置 12 位 **TIMEOUTB** 定时器，以便检查从机的 $T_{\text{LOW:SEXT}}$ 和主机的 $T_{\text{LOW:MEXT}}$ 。由于标准仅指定最大值，因此用户可以为两者选择相同的值。然后，通过将 **I2C_TIMEOUTR** 寄存器中的 **TEXTEN** 位置 1 来启用定时器。

注意：设置 **TEXTEN** 位时，不允许更改 **TIMEOUTB** 配置。

总线空闲检测

为了启用 T_{IDLE} 检查，必须对 12 位 **TIMEOUTA** 定时器进行编程，以获得 T_{IDLE} 参数。必须将 **TIDLE** 位配置为 1 才能检测 **SCL** 和 **SDA** 高电平超时。

然后，通过将 **I2C_TIMEOUTR** 寄存器中的 **TIMOUTEN** 位置 1 来启用定时器。如果 **SCL** 和 **SDA** 线都保持高电平的时间大于 $(\text{TIMEOUTA} + 1) \times 4 \times T_{\text{I2CCCLK}}$ ，则在 **I2C_RIF** 寄存器中设置 **TOUTRI** 标志。

注意：设置 **TIMEOUTEN** 时，不允许更改 **TIMEOUTA** 和 **TIDLE** 配置。

23.4.9 SMBus: I2C_TIMEOUTR 寄存器配置的例子

- ◆ 将 T_{TIMEOUT} 的最大持续时间配置为 25 ms:

F_{I2CCCLK}	TIMEOUT[11:0] bits	TIDLE bit	TIMEOUTEN bit	T_{TIMEOUT}
8 MHz	0x61	0	1	$98 \times 2048 \times 125 \text{ ns} = 25 \text{ ms}$
16 MHz	0xC3	0	1	$196 \times 2048 \times 62.5 \text{ ns} = 25 \text{ ms}$
32 MHz	0x186	0	1	$391 \times 2048 \times 31.25 \text{ ns} = 25 \text{ ms}$

表 23-6 各种 I2CCCLK 频率的 TIMEOUTA 设置示例 (最大值 $T_{\text{TIMEOUT}} = 25 \text{ ms}$)

- ◆ 将 $T_{\text{LOW:SEXT}}$ 和 $T_{\text{LOW:MEXT}}$ 的最大持续时间配置为 8 ms

F_{I2CCCLK}	TIMEOUT[11:0] bits	TXMEOUTEN bit	$T_{\text{LOW:SEXT}}$
8 MHz	0x1F	1	$32 \times 2048 \times 125 \text{ ns} = 8 \text{ ms}$
16 MHz	0x3F	1	$64 \times 2048 \times 62.5 \text{ ns} = 8 \text{ ms}$
32 MHz	0x7C	1	$125 \times 2048 \times 31.25 \text{ ns} = 8 \text{ ms}$

表 23-7 各种 I2CCCLK 频率的 TIMEOUTB 设置示例

- ◆ 将 T_{IDLE} 的最大持续时间配置为 50 μs

F_{I2CCCLK}	TIMEOUT[11:0] bits	TIDLE bit	TIMEOUTEN bit	T_{IDLE}
8 MHz	0x63	1	1	$100 \times 4 \times 125 \text{ ns} = 50 \mu\text{s}$
16 MHz	0xC7	1	1	$200 \times 4 \times 62.5 \text{ ns} = 50 \mu\text{s}$
32 MHz	0x18F	1	1	$400 \times 4 \times 31.25 \text{ ns} = 50 \mu\text{s}$

表 23-8 各种 I2CCCLK 频率的 TIMEOUTA 设置示例 (最大 $T_{\text{IDLE}} = 50 \mu\text{s}$)

23.4.10 DMA 请求

使用 DMA 发送

通过将 **I2C_CON1** 寄存器中的 TXDMAEN 位置 1，可以启用 DMA 进行发送。只要 TXTH 位置 1，就会从使用 DMA 外设配置的 SRAM 区域加载数据到 **I2C_TXDATA** 寄存器。只有数据通过 DMA 发送。

使用 DMA 接收

通过将 **I2C_CON1** 寄存器中的 RXDMAEN 位置 1，可以启用 DMA 以进行接收。只要 RXTH 位置 1，就会将数据从 **I2C_RXDATA** 寄存器加载到使用 DMA 外设配置的 SRAM 区域。仅使用 DMA 传输数据（包括 PEC）。

23.4.11 错误情况

以下是可能导致通信失败的错误情况。

总线错误（BERR）

总线错误是指在总线上不在 9 个 SCL 时钟脉冲的倍数之后检测到 START 或 STOP 条件。仅当 I2C 作为主机或寻址从机进行传输时（不在从机模式下的地址阶段），才会设置总线错误标志。

如果在从机模式下检测到错误的 START 或 RESTART，则 I2C 会进入地址识别状态，就像正确的 START 条件一样。

检测到总线错误时，BERR 标志位在 **I2C_RIF** 寄存器中，如果 **I2C_IER** 寄存器中的 BERRIE 位置 1，则产生中断。

仲裁丢失（ARLO）

仲裁丢失为当在 SDA 线上发送高电平时，但在 SCL 上升沿上采样到低电平。

- ◆ 在主机模式下，会在地址阶段、数据阶段和数据应答阶段检测仲裁丢失。在这种情况下，SDA 和 SCL 线被释放，START 控制位由硬件清零，主机自动切换到从机模式。
- ◆ 在从机模式下，会在数据阶段和数据应答阶段检测仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。当检测到仲裁丢失时，ARLORI 标志位在 **I2C_RIF** 寄存器中，并且如果在 **I2C_IER** 寄存器中设置了 ARLOIE 位，则会产生中断。

接收溢出 / 发送欠载错误（RXOV / TXUD）

当 NOSTRETCH = 1 时，在从机模式下检测到溢出或欠载错误：

- ◆ 当接收到新字节且接收 FIFO 满时。新接收的字节将会丢失，并且自动发送 NACK 作为对新字节的应答。
- ◆ 在传输中：
 - 当应发送新字节且尚未写入发送 FIFO 时，将发送 0xFF。

当检测到接收溢出或发送欠载错误时，TXUD 与 RXOV 标志位在 **I2C_STAT** 寄存器中，如果 **I2C_IER** 寄存器中的 TXUDIE 或 RXOVIE 位置 1，则会产生中断。

23. 4. 12 I2C 中断

I2C 中的中断由一组六个寄存器控制。

◆ 中断控制 (IER, IDR, IVS)

I2C 中断启用寄存器 (I2C_IER) 通过写 1 来启用中断请求线。同样, I2C 中断禁用寄存器 (I2C_IDR) 通过写'1'来禁用中断。IER 和 IDR 是只写寄存器, 上述寄存器的总结果可由中断有效状态寄存器 (I2C_IVS) 显示。IVS 是一个只读寄存器, 使用“1”或“0”来指示中断请求线是否有效。

◆ 原始中断标志状态读取 (RIF)

I2C 原始中断状态 (I2C_RIF) 是一个只读寄存器, 用于读取模块的中断状态。该寄存器中的位显示 I2C 中断的真实状态。当观察到以下条件时, I2C 可以产生中断:

- ◇ SMBus 警报
- ◇ 发生超时
- ◇ PEC 错误
- ◇ 仲裁丢失
- ◇ 总线错误
- ◇ 传送完成并重新加载
- ◇ 传送完成
- ◇ 检测检测 STOP
- ◇ 收到非应答
- ◇ 地址匹配
- ◇ 接收 FIFO 不为空
- ◇ 发送 FIFO 为空
- ◇ 发生接收溢出或发送欠载

◆ 中断标志屏蔽状态 (IFM)

I2C 中断标志屏蔽状态寄存器 (I2C_IFM) 用于读取模块的屏蔽中断状态, 显示屏蔽了哪个中断。IFM 中的每个位是 IVS 和 RIF 中各个位的逻辑 AND。

◆ 中断清除 (ICR)

向该寄存器的位写“1”可以清除相应的中断。

23.5 特殊功能寄存器

23.5.1 寄存器列表

I2C 寄存器列表			
名称	偏移地址	类型	描述
I2Cx_CON1	0000 _H	R/W	I2C 控制寄存器 1
I2Cx_CON2	0004 _H	R/W	I2C 控制寄存器 2
I2Cx_ADDR1	0008 _H	R/W	I2C 本机地址寄存器 1
I2Cx_ADDR2	000C _H	R/W	I2C 本机地址寄存器 2
I2Cx_TIMINGR	0010 _H	R/W	I2C 时序寄存器
I2Cx_TIMEOUTR	0014 _H	R/W	I2C 超时寄存器
I2Cx_STAT	0018 _H	R	I2C 状态寄存器
I2Cx_FCON	001C _H	R/W	I2C FIFO 控制寄存器
I2Cx_PECR	0020 _H	R	I2C PEC 寄存器
I2Cx_RXDATA	0024 _H	R	I2C 接收数据寄存器
I2Cx_TXDATA	0028 _H	W	I2C 发送数据寄存器
I2Cx_IER	002C _H	W1	I2C 中断启用寄存器
I2Cx_IDR	0030 _H	W1	I2C 中断禁用寄存器
I2Cx_IVS	0034 _H	R	I2C 中断有效状态寄存器
I2Cx_RIF	0038 _H	R	I2C 原始中断标志状态寄存器
I2Cx_IFM	003C _H	R	I2C 中断标志屏蔽状态寄存器
I2Cx_ICR	0040 _H	C_W1	I2C 中断清除寄存器

			<p>0: 禁用主控者地址。地址 0b0001000x 为 NACKed。</p> <p>1: 启用主控者地址。地址 0b0001000x 被 ACKed。</p> <p>注意: 如果不支持 SMBus 功能, 则该位保留并由硬件强制为“0”。</p>
GCEN	Bit 19	R/W	<p>广播呼叫启用</p> <p>0: 禁用广播呼叫。地址 0b00000000 是 NACKed。</p> <p>1: 启用广播呼叫。地址 0b00000000 被 ACKed。</p>
—	Bit 18	—	—
NOSTRETCH	Bit 17	R/W	<p>时钟延长禁用</p> <p>该位用于禁用从机模式下的时钟延长。必须在主机模式下保持清除状态。</p> <p>0: 启用时钟延长</p> <p>1: 禁用时钟延长</p> <p>注: 只有在 I2C 被禁止 (PE = 0) 时才能对该位进行编程。</p>
SBC	Bit 16	R/W	<p>从机字节控制</p> <p>该位用于在从机模式下启用硬件字节控制。</p> <p>0: 禁用从机字节控制</p> <p>1: 启用从机字节控制</p>
RXDMAEN	Bit 15	R/W	<p>启用 DMA 接收请求</p> <p>0: 禁用 DMA 模式进行接收</p> <p>1: 启用 DMA 模式进行接收</p>
TXDMAEN	Bit 14	R/W	<p>启用 DMA 发送请求</p> <p>0: 禁止 DMA 模式进行发送</p> <p>1: 启用 DMA 模式进行发送</p>
—	Bits 13-12	—	—
DNF	Bits 11-8	R/W	<p>数字噪声滤波器</p> <p>这些位用于配置 SDA 和 SCL 输入上的数字噪声滤波器。数字滤波器将过滤长度高达 $DNF[3:0] * T_{I2CCLK}$ 的尖峰</p> <p>0000: 禁用数字滤波器</p> <p>0001: 启用数字滤波器, 滤波能力高达 $1 T_{I2CCLK}$</p> <p>...</p> <p>1111: 启用数字滤波器, 滤波能力高达 $15 T_{I2CCLK}$</p> <p>注意: 只有在禁用 I2C (PE = 0) 时才能对该滤波器进行编程。</p>
—	Bits 7-1	—	—

PE	Bit 0	R/W	<p>I2C 启用</p> <p>0: I2C 禁用</p> <p>1: I2C 启用</p> <p>注: 当 PE = 0 时, I2C SCL 和 SDA 线被释放。内部状态机和状态位将恢复为其复位值。清零后, PE 必须保持低电平至少 3 个 APB 时钟周期。</p>
----	-------	-----	---

23.5.2.2 I2C 控制寄存器 2 (I2Cx_CON2)

I2C 控制寄存器 2 (I2Cx_CON2)																																	
偏移地址: 04 _H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
—		ACK_UPD	HOLDACK	—	PEC_BYTE	AUTO_END	RELOAD	NBYTES<7:0>								NACK	STOP	START	HEAD_10R	ADD10	RD_WRN	SADD<9:0>											

—	Bits 31-30	—	—
ACK_UPD	Bit 29	R/C_W1	ACK 更新 0: 不更新 ACK 状态 1: 更新 ACK 状态并释放时钟
HOLDACK	Bit 28	R/W	等待回应 ACK 启用时, 第 8 和第 9 个 SCL 脉冲之间会将 SCL 延长, 并等待 USER 设置 ACK/NACK。在设置 ACK/NACK 之后, 设置 ACK_UPD 以发出 ACK/NACK 并释放 SCL。 0: 自动应答 ACK 1: 启用手动回应 ACK
—	Bit 27	—	—
PECBYTE	Bit 26	R/W	数据包错误检查字节 该位由软件置 1, 当传输 PEC 时, 或者当接收到 STOP 条件或匹配的地址时, 由硬件清零, 当 PE = 0 时也是如此。 0: 没有 PEC 传输。 1: 请求 PEC 发送/接收 注意: 向该位写“0”无效。 设置 RELOAD 时, 该位无效。 当 SBC = 0 时, 该位对从机模式无效。 如果不支持 SMBus 功能, 则该位保留并由硬件强制为“0”。
AUTOEND	Bit 25	R/W	自动结束模式 (主机模式) 该位由软件置位和清除。 0: 软件结束模式: 发送完 NBYTES 数据时设置 TC 标志, 将 SCL 拉低。 1: 自动结束模式: 发送完 NBYTES 数据时自动发送 STOP 条件。

			注：该位在从机模式或 RELOAD 位置 1 时无效。
RELOAD	Bit 24	R/W	<p>NBYTES 重新加载模式</p> <p>该位由软件置位和清除。</p> <p>0：在 NBYTES 数据传输之后完成传输（接下来是 STOP 或 RESTART）。</p> <p>1：NBYTES 数据传输后将继续传输（NBYTES 将重新加载）。发送完 NBYTES 数据后设置 TCR 标志，将 SCL 拉低。</p>
NBYTES	Bits 23-16	R/W	<p>字节数</p> <p>在此编程要发送/接收的字节数。</p> <p>在 SBC = 0 的从机模式下，该字段无关紧要。</p> <p>注：不允许在设置 START 位时更改这些位。</p>
NACK	Bit 15	R/S_W1	<p>NACK 生成（从机模式）</p> <p>该位由软件置位，在发送 NACK 时由硬件清零，或者在接收到 STOP 条件或地址匹配时，或者当 PE = 0 时由硬件清零。</p> <p>0：在下一个接收的字节发送 ACK。</p> <p>1：在下一个接收的字节发送 NACK。</p> <p>注意：向该位写“0”无效。</p> <p>该位仅用于从机模式：在主机接收模式下，无论 NACK 位值如何，在 STOP 或 RESTART 条件之前的最后一个字节之后自动生成 NACK。当从机接收 NOSTRETCH 模式发生溢出时，无论 NACK 位值如何，都会自动生成 NACK。当启用硬件 PEC 检查（PECBYTE = 1）时，PEC 确认值不依赖于 NACK 值。</p>
STOP	Bit 14	R/S_W1	<p>STOP 生成（主机模式）</p> <p>该位由软件置位，当检测到停止条件时，或者当 PE = 0 时由硬件清零。</p> <p>在主模式下：</p> <p>0：无 STOP 生成。</p> <p>1：当前字节传输后 STOP 生成。</p> <p>注意：向该位写“0”无效。</p>
START	Bit 13	R/S_W1	<p>START 生成</p> <p>该位由软件置位，并在启动后跟随地址序列发送，仲裁丢失，超时错误检测或 PE = 0 时由硬件清零。</p> <p>0：无 START 生成。</p> <p>1：RESTART / START 生成：</p>

			<p>- 如果 I2C 已经处于主模式且 AUTOEND = 0, 则在 NBYTES 传输结束后, 当 RELOAD = 0 时, 将该位置 1 会产生重复启动条件。</p> <p>- 否则, 一旦总线空闲, 将该位置 1 产生 START 条件。</p> <p>注意: 向该位写“0”无效。</p> <p>即使总线忙, 也可以设置 START 位。</p> <p>在 10 位寻址模式下, 如果在地址的第一部分接收到 NACK, 则 START 位不会被硬件清零, 主机将重新发送地址序列, 除非 START 位被软件清零</p>
HEAD10R	Bit 12	R/W	<p>10 位地址标头仅读取方向 (主机接收模式)</p> <p>0: 主机发送完整的 10 位从机地址读取序列: 在写入方向上启动 + 2 字节 10 位地址 + 在读取方向上重新启动 + 10 位地址的第 7 位。</p> <p>1: 主机仅发送 10 位地址的前 7 位, 然后发送读取方向。</p> <p>注: 不允许在设置 START 位时更改该位。</p>
ADD10	Bit 11	R/W	<p>10 位寻址模式 (主机模式)</p> <p>0: 主机工作在 7 位寻址模式</p> <p>1: 主机工作在 10 位寻址模式</p> <p>注: 不允许在设置 START 位时更改该位。</p>
RD_WRN	Bit 10	R/W	<p>传输方向 (主机模式)</p> <p>0: 主机请求写入传输。</p> <p>1: 主机请求读取传输。</p> <p>注: 不允许在设置 START 位时更改该位。</p>
SADD	Bits 9-0	R/W	<p>从机地址位 0 (主机模式)</p> <p>在 7 位寻址模式下 (ADD10 = 0): 此位不在乎</p> <p>在 10 位寻址模式下 (ADD10 = 1): 该位应写入要发送的从地址的第 0 位</p> <p>从机地址位 7: 1 (主机模式)</p> <p>在 7 位寻址模式下 (ADD10 = 0): 应使用要发送的 7 位从地址写入这些位</p> <p>在 10 位寻址模式下 (ADD10 = 1): 应使用要发送的从地址的位 7: 1 写入这些位。</p> <p>从机地址位 9: 8 (主机模式)</p> <p>在 7 位寻址模式下 (ADD10 = 0): 这些位都不在乎</p>

			在 10 位寻址模式下 (ADD10 = 1): 应使用要发送的从地址的第 9: 8 位写入这些位 注: 不允许在设置 START 位时更改这些位。
--	--	--	--

23.5.2.3 I2C 本机地址寄存器 1 (I2Cx_ADDR1)

I2C 本机地址寄存器 1 (I2Cx_ADDR1)																																		
偏移地址: 08H																																		
复位值: 00000000_00000000_00000000_00000000 _B																																		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
																OA1EN						OA1MODE	OA1<9:8>										OA1<7:1>	OA1<0>

—	Bits 31-16	—	—
OA1EN	Bit 15	R/W	<p>本机地址 1 启用</p> <p>0: 禁用本机地址 1。接收到的从机地址 OA1 是 NACKed。</p> <p>1: 启用本机地址 1。接收到的从机地址 OA1 是 ACKed。</p>
—	Bits 14-11	—	—
OA1MODE	Bit 10	R/W	<p>本机地址 1, 10 位地址模式启用</p> <p>0: 本机地址 1 是 7 位地址。</p> <p>1: 本机地址 1 是 10 位地址。</p> <p>注: 仅当 OA1EN = 0 时才能写入该位。</p>
OA1	Bits 9-0	R/W	<p>OA1 [0]: 本机地址 1</p> <p>7 位寻址模式: 不在乎</p> <p>10 位寻址模式: 地址的第 0 位</p> <p>OA1 [7: 1]: 本机地址 1</p> <p>7 位寻址模式: 7 位地址</p> <p>10 位寻址模式: 10 位地址的 7: 1 位</p> <p>OA1 [9: 8]: 本机地址 1</p> <p>7 位寻址模式: 不在乎</p> <p>10 位寻址模式: 地址的 9: 8 位</p> <p>注: 仅当 OA1EN = 0 时才能写入该位。</p>

23.5.2.4 I2C 本机地址寄存器 2 (I2Cx_ADDR2)

I2C 本机地址寄存器 2 (I2Cx_ADDR2)																																	
偏移地址: 0C _H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																OA2EN																	

—	Bits 31-16	—	—
OA2EN	Bit 15	R/W	<p>本机地址 2 启用</p> <p>0: 禁用本机地址 2。接收到的从机地址 OA2 是 NACKed。</p> <p>1: 启用本机地址 2。接收到的从机地址 OA2 是 ACKed。</p>
—	Bits 14-11	—	—
OA2MSK	Bits 10-8	R/W	<p>本机地址 2 屏蔽</p> <p>000: 没有屏蔽</p> <p>001: OA2 [1]被屏蔽并且不在乎。仅比较 OA2 [7: 2]。</p> <p>010: OA2 [2: 1]被屏蔽并且不在乎。仅比较 OA2 [7: 3]。</p> <p>011: OA2 [3: 1]被屏蔽并且不在乎。仅比较 OA2 [7: 4]。</p> <p>100: OA2 [4: 1]被屏蔽并且不在乎。仅比较 OA2 [7: 5]。</p> <p>101: OA2 [5: 1]被屏蔽并且不在乎。仅比较 OA2 [7: 6]。</p> <p>110: OA2 [6: 1]被屏蔽并且不在乎。仅比较 OA2 [7]。</p> <p>111: OA2 [7: 1]被屏蔽并且不在乎。不进行比较, 并且应答所有 (除了保留的) 7 位接收地址。</p> <p>注意: 仅当 OA2EN = 0 时才能写入这些位。</p> <p>一旦 OA2MSK 不等于 0, 则 I2C 即使比较匹配, 也不会应答保留地址 (0b0000xxx 和 0b1111xxx)。</p>
OA2	Bits 7-1	R/W	<p>本机地址 2</p> <p>7 位寻址模式: 7 位地址</p>

			注意：仅当 OA2EN = 0 时才能写入这些位。
—	Bit 0	—	—

23.5.2.5 I2C 时钟寄存器 (I2Cx_TIMINGR)

I2C 时钟寄存器 (I2Cx_TIMINGR)																															
偏移地址: 10H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRESC<3:0>				SCLDEL<3:0>				SDADEL<3:0>				SCLH<7:0>								SCLL<7:0>											

PRESC	Bits 31-28	R/W	<p>时钟预分频器</p> <p>该字段用于预分频 I2CCLK，以产生用于数据设置和保持计数器以及 SCL 高电平和低电平计数器的时钟周期 T_{PRESC}。</p> $T_{PRESC} = (PRESC + 1) \times T_{I2CCLK}$ <p>禁用 I2C (PE = 0) 时，必须配置该寄存器。</p>
—	Bits 27-24	—	—
SCLDEL	Bits 23-20	R/W	<p>数据设置时间</p> <p>该字段用于在 SDA 边沿和 SCL 上升沿之间生成延迟 T_{SCLDEL}。在主机和从机模式下，NOSTRETCH = 0 时，SCL 线在 T_{SCLDEL} 期间拉低。</p> $T_{SCLDEL} = (SCLDEL + 1) \times T_{PRESC}$ <p>注意：T_{SCLDEL} 用于生成 T_{SU: DAT} 时序。</p> <p>禁用 I2C (PE = 0) 时，必须配置该寄存器。</p>
SDADEL	Bits 19-16	R/W	<p>数据保持时间</p> <p>该字段用于在 SCL 下降沿和 SDA 边沿之间生成延迟 T_{SDADEL}。在主机和从机模式下，NOSTRETCH = 0 时，SCL 线在 T_{SDADEL} 期间拉低。</p> $T_{SDADEL} = SDADEL \times T_{PRESC}$ <p>注意：SDADEL 用于生成 T_{HD: DAT} 时序。</p> <p>禁用 I2C (PE = 0) 时，必须配置该寄存器。</p>
SCLH	Bits 15-8	R/W	<p>SCL 高电平期间 (主机模式)</p> <p>该字段用于在主机模式下生成 SCL 高电平周期。</p> $T_{SCLH} = (SCLH + 1) \times T_{PRESC}$ <p>注意：SCLH 还用于生成 T_{SU: STO} 和 T_{HD: STA} 时序。</p> <p>禁用 I2C (PE = 0) 时，必须配置该寄存器。</p>

SCLL	Bits 7-0	R/W	<p>SCL 低电平周期（主机模式）</p> <p>该字段用于在主机模式下生成 SCL 低电平周期。</p> <p>$T_{SCLL} = (SCLL + 1) \times T_{PRESC}$</p> <p>注意：SCLL 还用于生成 T_{BUF} 和 $T_{SU: STA}$ 时序。</p> <p>禁用 I2C（PE = 0）时，必须配置该寄存器。</p>
------	----------	-----	--

23.5.2.6 I2C 超时寄存器 (I2Cx_TIMEOUTR)

I2C 超时寄存器 (I2Cx_TIMEOUTR)																																	
偏移地址: 14H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
TEXTEN				TIMEOUTB<11:0>												TIMEOUTEN			TIDLE			TIMEOUTA<11:0>											

TEXTEN	Bit 31	R/W	<p>累积时钟延长超时启用</p> <p>0: 禁用累积时钟延长超时检测</p> <p>1: 启用累积时钟延长超时检测。</p> <p>当 I2C 接口完成累积 SCL 延长超过 $T_{LOW:EXT}$ 时, 会检测到超时错误 (TOUTRI = 1)。</p>
—	Bits 30-28	—	—
TIMEOUTB	Bits 27-16	R/W	<p>总线超时 B</p> <p>该字段用于配置累积时钟延长超时:</p> <p>在主机模式下, 检测到主机累积时钟低延长时间 ($T_{LOW:MEXT}$)。</p> <p>在从机模式下, 检测到从机累积时钟低延长时间 ($T_{LOW:SEXT}$)。</p> <p>$T_{LOW:EXT} = (TIMEOUTB + 1) \times 2048 \times T_{I2CCCLK}$</p> <p>注意: 仅当 TEXTEN = 0 时才能写入这些位。</p>
TIMEOUTEN	Bit 15	R/W	<p>时钟超时启用</p> <p>0: 禁用 SCL 超时检测</p> <p>1: 启用 SCL 超时检测: 当 SCL 为低电平超过 $T_{TIMEOUT}$ ($TIDLE = 0$) 或高电平超过 T_{IDLE} ($TIDLE = 1$) 时, 检测到超时错误 (TOUTRI = 1)。</p>
—	Bits 14-13	—	—
TIDLE	Bit 12	R/W	<p>空闲时钟超时检测</p> <p>0: TIMEOUTA 用于检测 SCL 低超时</p> <p>1: TIMEOUTA 用于检测 SCL 和 SDA 高超时 (总线空闲状态)</p> <p>注: 仅当 TIMEOUTEN = 0 时才能写入该位。</p>
TIMEOUTA	Bits 11-0	R/W	<p>总线超时 A</p> <p>该字段用于配置:</p> <ul style="list-style-type: none"> - 当 TIDLE = 0 时, SCL 低超时条件 $T_{TIMEOUT}$

		<p> $T_{\text{TIMEOUT}} = (\text{TIMEOUTA} + 1) \times 2048 \times T_{\text{I2CCCLK}}$ - 当 TIDLE = 1 时，总线空闲状态（SCL 和 SDA 均为高电平） $T_{\text{IDLE}} = (\text{TIMEOUTA} + 1) \times 4 \times T_{\text{I2CCCLK}}$ 注意：仅当 TIMOUTEN = 0 时才能写入这些位。 </p>
--	--	--

23.5.2.7 I2C 状态寄存器 (I2Cx_STAT)

I2C 状态寄存器 (I2Cx_STAT)																																
偏移地址: 18 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								ADDCODE<6:0>							DIR	BUSY					TCR	TC	RXTH	RXUD	RXOV	RXF	RXE	TXTH	TXUD	TXOV	TXF	TXE

—	Bits 31-24	—	—
ADDCODE	Bits 23-17	R	地址匹配代码 (从机模式) 当发生地址匹配事件 (ADDR = 1) 时, 使用接收的地址更新这些位。在 10 位地址的情况下, ADDCODE 提供 10 位标头与地址的 2 个 MSB。
DIR	Bit 16	R	传输方向 (从机模式) 发生地址匹配事件 (ADDRRI = 1) 时更新此标志。 0: 写入传输, 从机进入接收模式。 1: 读取传输, 从机进入发送模式。
BUSY	Bit 15	R	总线忙 该标志表示总线上正在进行通信。检测到 START 条件时由硬件设置。当检测到停止条件或 PE = 0 时, 它由硬件清零。
—	Bits 14-12	—	—
TCR	Bit 11	R	传输完成并重新加载 当 RELOAD = 1 且已将 NBYTES 数据传输完成时, 此标志由硬件设置。当 NBYTES 写入非零值时, 它由硬件清零。 注: 当 PE = 0 时, 该位由硬件清零。 该标志仅用于主模式, 或者用于从模式 SBC 位置 1 时。
TC	Bit 10	R	传输完成 (主机模式) 当 RELOAD = 0, AUTOEND = 0 且已将 NBYTES 数据传输完成时, 该标志由硬件设置。当 START 位或 STOP 位置 1 时, 由硬件清零。 注: 当 PE = 0 时, 该位由硬件清零。
RXTH	Bit 9	R	接收 FIFO 水平超出阈值 该位提供接收 FIFO 水平超过阈值。 0: 接收 FIFO 水平没有超过阈值

			1: 接收 FIFO 水平超过阈值
RXUD	Bit 8	R	接收 FIFO 下溢 0: 接收 FIFO 没有下溢 1: 接收 FIFO 下溢
RXOV	Bit 7	R	接收 FIFO 溢出 0: 接收 FIFO 没有溢出 1: 接收 FIFO 溢出
RXF	Bit 6	R	接收 FIFO 满 0: 接收 FIFO 没有满 1: 接收 FIFO 满
RXE	Bit 5	R	接收 FIFO 空 0: 接收 FIFO 没有空 1: 接收 FIFO 空
TXTH	Bit 4	R	发送 FIFO 水平低于阈值 该位提供发送 FIFO 水平低于阈值。 0: 发送 FIFO 水平没有超过阈值 1: 发送 FIFO 水平超过阈值
TXUD	Bit 3	R	发送 FIFO 下溢 0: 发送 FIFO 没有下溢 1: 发送 FIFO 下溢
TXOV	Bit 2	R	发送 FIFO 溢出 0: 发送 FIFO 没有溢出 1: 发送 FIFO 溢出
TXF	Bit 1	R	发送 FIFO 满 0: 发送 FIFO 没有满 1: 发送 FIFO 满
TXE	Bit 0	R	发送 FIFO 空 0: 发送 FIFO 没有空 1: 发送 FIFO 空

			1: 重置发送指针 该位由硬件自动清除。
TXFLV	Bits 4-0	R	发送 FIFO 电平 该位由硬件设置。这用于指示发送 FIFO 中的数据条目数。 它从 0 到 16。

23.5.2.9 I2C PEC 寄存器 (I2Cx_PECR)

I2C PEC 寄存器 (I2Cx_PECR)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								PEC<7:0>							

—	Bits 31-8	—	—
PEC	Bits 7-0	R	数据包错误检查寄存器 当 PECEN = 1 时, 该字段包含内部 PEC。 当 PECEN = 0 时, PEC 由硬件清零。

23.5.2.10 I2C 接收数据寄存器 (I2Cx_RXDATA)

I2C 接收数据寄存器 (I2Cx_RXDATA)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								RXDATA<7:0>							

—	Bits 31-8	—	—
RXDATA	Bits 7-0	R	8 位接收数据 从 I2C 总线接收的数据字节。

23.5.2.11 I2C 发送数据寄存器 (I2Cx_TXDATA)

I2C 发送数据寄存器 (I2Cx_TXDATA)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								TXDATA<7:0>							

—	Bits 31-8	—	—
TXDATA	Bits 7-0	W	8 位发送数据 要传输到 I2C 总线的数据字节。

23.5.2.12 I2C 中断启用寄存器 (I2Cx_IER)

I2C 中断启用寄存器 (I2Cx_IER)																																
偏移地址: 2C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
											ALERTIE	TOUTIE	PECEIE	ARLOIE	BERRIE		STOPIE	NACKIE	ADDRIE	TCRIE	TCIE	RXTHIE	RXUDIE	RXOVIE	RXFIE			TXTHIE	TXUDIE	TXOVIE		TXEIE

—	Bits 31-21	—	—
ALERTIE	Bit 20	W1	SMBus 报警中断启用 0: 无效 1: 启用中断
TOUTIE	Bit 19	W1	超时中断启用 0: 无效 1: 启用中断
PECEIE	Bit 18	W1	PEC 错误中断启用 0: 无效 1: 启用中断
ARLOIE	Bit 17	W1	仲裁丢失中断启用 0: 无效 1: 启用中断
BERRIE	Bit 16	W1	总线错误中断启用 0: 无效 1: 启用中断
—	Bit 15	—	—
STOPIE	Bit 14	W1	停止检测中断启用 0: 无效 1: 启用中断
NACKIE	Bit 13	W1	NACK 接收中断启用 0: 无效 1: 启用中断
ADDRIE	Bit 12	W1	地址匹配中断启用 0: 无效 1: 启用中断
TCRIE	Bit 11	W1	传输完成并重新加载中断启用 0: 无效 1: 启用中断
TCIE	Bit 10	W1	传输完成中断启用 0: 无效 1: 启用中断

RXTHIE	Bit 9	W1	接收 FIFO 超过阈值中断启用 0: 无效 1: 启用中断
RXUDIE	Bit 8	W1	接收 FIFO 欠载中断启用 0: 无效 1: 启用中断
RXOVIE	Bit 7	W1	接收 FIFO 溢出中断启用 0: 无效 1: 启用中断
RXFIE	Bit 6	W1	接收 FIFO 满中断启用 0: 无效 1: 启用中断
—	Bit 5	—	—
TXTHIE	Bit 4	W1	发送 FIFO 低于阈值中断启用 0: 无效 1: 启用中断
TXUDIE	Bit 3	W1	发送 FIFO 欠载中断启用 0: 无效 1: 启用中断
TXOVIE	Bit 2	W1	发送 FIFO 溢出中断启用 0: 无效 1: 启用中断
—	Bit 1	—	—
TXEIE	Bit 0	W1	发送 FIFO 空中断启用 0: 无效 1: 启用中断

23.5.2.13 I2C 中断禁用寄存器 (I2Cx_IDR)

I2C 中断禁用寄存器(I2Cx_IDR)																															
偏移地址: 30H																															
复位值: 00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ALERTID	TOUTID	PECEID	ARLOID	BERRID		STOPID	NACKID	ADDRID	TCRID	TCID	RXTHID	RXUDID	RXOVID	RXFID		TXTHID	TXUDID	TXOVID		TXEID

—	Bits 31-21	—	—
ALERTID	Bit 20	W1	SMBus 警报中断禁用 0: 无效 1: 禁用中断
TOUTID	Bit 19	W1	超时中断禁用 0: 无效 1: 禁用中断
PECEID	Bit 18	W1	PEC 错误中断禁用 0: 无效 1: 禁用中断
ARLOID	Bit 17	W1	仲裁丢失中断禁用 0: 无效 1: 禁用中断
BERRID	Bit 16	W1	总线错误中断禁用 0: 无效 1: 禁用中断
—	Bit 15	—	—
STOPID	Bit 14	W1	停止检测中断禁用 0: 无效 1: 禁用中断
NACKID	Bit 13	W1	NACK 接收中断禁用 0: 无效 1: 禁用中断
ADDRID	Bit 12	W1	地址匹配中断禁用 0: 无效 1: 禁用中断
TCRID	Bit 11	W1	传输完成并重新加载中断禁用 0: 无效 1: 禁用中断
TCID	Bit 10	W1	传输完成中断禁用 0: 无效 1: 禁用中断
RXTHID	Bit 9	W1	接收 FIFO 超过阈值中断禁用

			0: 无效 1: 禁用中断
RXUDID	Bit 8	W1	接收 FIFO 欠载中断禁用 0: 无效 1: 禁用中断
RXOVID	Bit 7	W1	接收 FIFO 溢出中断禁用 0: 无效 1: 禁用中断
RXFID	Bit 6	W1	接收 FIFO 满中断禁用 0: 无效 1: 禁用中断
—	Bit 5	—	—
TXTHID	Bit 4	W1	发送 FIFO 低于阈值中断禁用 0: 无效 1: 禁用中断
TXUDID	Bit 3	W1	发送 FIFO 欠载中断禁用 0: 无效 1: 禁用中断
TXOVID	Bit 2	W1	发送 FIFO 溢出中断禁用 0: 无效 1: 禁用中断
—	Bit 1	—	—
TXEID	Bit 0	W1	发送 FIFO 空中断禁用 0: 无效 1: 禁用中断

23.5.2.14 I2C 中断有效状态寄存器 (I2Cx_IVS)

I2C 中断有效状态寄存器 (I2Cx_IVS)																																
偏移地址: 34 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
											ALERTIV	TOUTIV	PECEIV	ARLOIV	BERRIV		STOPIV	NACKIV	ADDRIV	TCRIV	TCIV	RXTHIV	RXUDIV	RXOVIV	RXFIV			TXTHIV	TXUDIV	TXOVIV		TXEIV

—	Bits 31-21	—	—
ALERTIV	Bit 20	R	SMBus 警报中断有效 0: 无效 1: 中断有效
TOUTIV	Bit 19	R	超时中断有效 0: 无效 1: 中断有效
PECEIV	Bit 18	R	PEC 错误中断有效 0: 无效 1: 中断有效
ARLOIV	Bit 17	R	仲裁丢失中断有效 0: 无效 1: 中断有效
BERRIV	Bit 16	R	总线错误中断有效 0: 无效 1: 中断有效
—	Bit 15	—	—
STOPIV	Bit 14	R	停止检测中断有效 0: 无效 1: 中断有效
NACKIV	Bit 13	R	NACK 接收中断有效 0: 无效 1: 中断有效
ADDRIV	Bit 12	R	地址匹配中断有效 0: 无效 1: 中断有效
TCRIV	Bit 11	R	传输完成并重新加载中断有效 0: 无效 1: 中断有效
TCIV	Bit 10	R	传输完成中断有效 0: 无效 1: 中断有效

RXTHIV	Bit 9	R	接收 FIFO 超过阈值中断有效 0: 无效 1: 中断有效
RXUDIV	Bit 8	R	接收 FIFO 欠载中断有效 0: 无效 1: 中断有效
RXOVIV	Bit 7	R	接收 FIFO 溢出中断有效 0: 无效 1: 中断有效
RXFIV	Bit 6	R	接收 FIFO 满中断有效 0: 无效 1: 中断有效
—	Bit 5	—	—
TXTHIV	Bit 4	R	发送 FIFO 低于阈值中断有效 0: 无效 1: 中断有效
TXUDIV	Bit 3	R	发送 FIFO 欠载中断有效 0: 无效 1: 中断有效
TXOVIV	Bit 2	R	发送 FIFO 溢出中断有效 0: 无效 1: 中断有效
—	Bit 1	—	—
TXEIV	Bit 0	R	发送 FIFO 空中断有效 0: 无效 1: 中断有效

23. 5. 2. 15 I2C 原始中断标志状态寄存器 (I2Cx_RIF)

I2C 原始中断标志状态寄存器 (I2Cx_RIF)																															
偏移地址: 38H																															
复位值: 00000000_00000000_00000000_00000000b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ALERTRI	TOUTRI	PECERI	ARLORI	BERRRI		STOPRI	NACKRI	ADDRRI	TCRRI	TCRI	RXTHRI	RXUDRI	RXOVRI	RXFRI		TXTHRI	TXUDRI	TXOVRI		TXERI

—	Bits 31-21	—	—
ALERTRI	Bit 20	R	SMBus 警报中断标志状态 0: 无效 1: 原始中断标志状态
TOUTRI	Bit 19	R	超时中断标志状态 0: 无效 1: 原始中断标志状态
PECERI	Bit 18	R	PEC 错误中断标志状态 0: 无效 1: 原始中断标志状态
ARLORI	Bit 17	R	仲裁丢失中断标志状态 0: 无效 1: 原始中断标志状态
BERRRI	Bit 16	R	总线错误中断标志状态 0: 无效 1: 原始中断标志状态
—	Bit 15	—	—
STOPRI	Bit 14	R	停止检测中断标志状态 0: 无效 1: 原始中断标志状态
NACKRI	Bit 13	R	NACK 接收中断标志状态 0: 无效 1: 原始中断标志状态
ADDRRI	Bit 12	R	地址匹配中断标志状态 0: 无效 1: 原始中断标志状态
TCRRI	Bit 11	R	传输完成并重新加载中断标志状态 0: 无效 1: 原始中断标志状态
TCRI	Bit 10	R	传输完成中断标志状态 0: 无效 1: 原始中断标志状态

RXTHRI	Bit 9	R	接收 FIFO 超过阈值中断标志状态 0: 无效 1: 原始中断标志状态
RXUDRI	Bit 8	R	接收 FIFO 欠载中断标志状态 0: 无效 1: 原始中断标志状态
RXOVRI	Bit 7	R	接收 FIFO 溢出中断标志状态 0: 无效 1: 原始中断标志状态
RXFRI	Bit 6	R	接收 FIFO 满中断标志状态 0: 无效 1: 原始中断标志状态
—	Bit 5	—	—
TXTHRI	Bit 4	R	发送 FIFO 低于阈值中断标志状态 0: 无效 1: 原始中断标志状态
TXUDRI	Bit 3	R	发送 FIFO 欠载中断标志状态 0: 无效 1: 原始中断标志状态
TXOVRI	Bit 2	R	发送 FIFO 溢出中断标志状态 0: 无效 1: 原始中断标志状态
—	Bit 1	—	—
TXERI	Bit 0	R	发送 FIFO 空中断标志状态 0: 无效 1: 原始中断标志状态

23.5.2.16 I2C 中断标志屏蔽状态寄存器 (I2Cx_IFM)

I2C 中断标志屏蔽状态寄存器 (I2Cx_IFM)																																
偏移地址: 3C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
											ALERTFM	TOUTFM	PECEFM	ARLOFM	BERRFM		STOPFM	NACKFM	ADDRFM	TCRFM	TCFM	RXTHFM	RXUDFM	RXOVFM	RXFFM			TXTHFM	TXUDFM	TXOVFM		TXEFM

—	Bits 31-21	—	—
ALERTFM	Bit 20	R	SMBus 报警中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
TOUTFM	Bit 19	R	超时中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
PECEFM	Bit 18	R	PEC 错误中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
ARLOFM	Bit 17	R	仲裁丢失中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
BERRFM	Bit 16	R	总线错误中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
—	Bit 15	—	—
STOPFM	Bit 14	R	停止检测中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
NACKFM	Bit 13	R	NACK 接收中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
ADDRFM	Bit 12	R	地址匹配中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
TCRFM	Bit 11	R	传输完成并重新加载中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
TCFM	Bit 10	R	传输完成中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态

RXTHFM	Bit 9	R	接收 FIFO 超过阈值中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
RXUDFM	Bit 8	R	接收 FIFO 欠载中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
RXOVFM	Bit 7	R	接收 FIFO 溢出中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
RXFFM	Bit 6	R	接收 FIFO 满中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
—	Bit 5	—	—
TXTHFM	Bit 4	R	发送 FIFO 低于阈值中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
TXUDFM	Bit 3	R	发送 FIFO 欠载中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
TXOVFM	Bit 2	R	发送 FIFO 溢出中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
—	Bit 1	—	—
TXEFM	Bit 0	R	发送 FIFO 空中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态

23.5.2.17 I2C 中断清除寄存器 (I2Cx_ICR)

I2C 中断清除寄存器 (I2Cx_ICR)																															
偏移地址: 40 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ALERTIC	TOUTIC	PECEIC	ARLOIC	BERRIC		STOPIC	NACKIC	ADDRIC	TCRIC	TCIC	RXTHIC	RXUDIC	RXOVIC	RXFIC		TXTHIC	TXUDIC	TXOVIC		TXEIC

—	Bits 31-21	—	—
ALERTIC	Bit 20	C_W1	SMBus 警报中断清除 0: 无效 1: 中断清除
TOUTIC	Bit 19	C_W1	超时中断清除 0: 无效 1: 中断清除
PECEIC	Bit 18	C_W1	PEC 错误中断清除 0: 无效 1: 中断清除
ARLOIC	Bit 17	C_W1	仲裁丢失中断清除 0: 无效 1: 中断清除
BERRIC	Bit 16	C_W1	总线错误中断清除 0: 无效 1: 中断清除
—	Bit 15	—	—
STOPIC	Bit 14	C_W1	停止检测中断清除 0: 无效 1: 中断清除
NACKIC	Bit 13	C_W1	NACK 接收中断清除 0: 无效 1: 中断清除
ADDRIC	Bit 12	C_W1	地址匹配中断清除 0: 无效 1: 中断清除
TCRIC	Bit 11	C_W1	传输完成并重新加载中断清除 0: 无效 1: 中断清除
TCIC	Bit 10	C_W1	传输完成中断清除 0: 无效 1: 中断清除

RXTHIC	Bit 9	C_W1	接收 FIFO 超过阈值中断清除 0: 无效 1: 中断清除
RXUDIC	Bit 8	C_W1	接收 FIFO 欠载中断清除 0: 无效 1: 中断清除
RXOVIC	Bit 7	C_W1	接收 FIFO 溢出中断清除 0: 无效 1: 中断清除
RXFIC	Bit 6	C_W1	接收 FIFO 满中断清除 0: 无效 1: 中断清除
—	Bit 5	—	—
TXTHIC	Bit 4	C_W1	发送 FIFO 低于阈值中断清除 0: 无效 1: 中断清除
TXUDIC	Bit 3	C_W1	发送 FIFO 欠载中断清除 0: 无效 1: 中断清除
TXOVIC	Bit 2	C_W1	发送 FIFO 溢出中断清除 0: 无效 1: 中断清除
—	Bit 1	—	—
TXEIC	Bit 0	C_W1	发送 FIFO 空中断清除 0: 无效 1: 中断清除

第24章 串行外设接口 (SPI)

24.1 概述

SPI /I2S 接口可用于使用 SPI 协议或 I2S 音频协议来与外部设备通信。SPI 或 I2S 模式可通过软件选择。IP 复位后，默认选择 SPI 模式。

串行外设接口 (SPI) 可与外部器件进行半双工/全双工的同步串行通信。该接口可配置为主模式，在这种情况下，它可为外部从机提供通信时钟 (SCK)。该接口还能够得多主模式配置下工作。

此接口能运行多种模式，如基于双线的单工同步传输，其中一条线用于双向数据传输，同时可用 CRC 校验提高通信可靠性。

Inter-IC sound (I2S) 协议也是同步串行通信接口。它可以通过半双工通信在从机或主机模式下运行。通过组合两个 I2S 块可以实现全双工操作。它可以满足四种不同的音频标准，包括 Philips I2S 标准，MSB 和 LSB 对齐标准以及 PCM 标准。

24.2 特性

SPI 的主要特点

- ◆ 主机或从机操作
- ◆ 基于三条线的全双工同步传输
- ◆ 基于双线的半双工同步传输 (使用双向数据线)
- ◆ 基于双线的单工同步传输 (使用单向数据线)
- ◆ 8 位或 16 位传输帧格式选择
- ◆ 支持多主机模式功能
- ◆ 8 个主机模式波特率预分频器，最高可达 $F_{PCLK} / 2$
- ◆ 对于主机模式和从机模式都可通过硬件或软件进行 NSS 控制：动态切换主机/从机操作
- ◆ 时钟极性和相位可编程
- ◆ 数据顺序可编程，如最先移位 MSB 或 LSB
- ◆ 具有专用的发送和接收中断标志功能
- ◆ 支持 SPI 总线忙状态标志
- ◆ 支持摩托罗拉 SPI 协议
- ◆ 支持 TI SPI 协议
- ◆ 用于确保可靠通信的硬件 CRC 功能
- ◆ 提供发送和接收 FIFO，深度为 16
- ◆ 支持 DMA 传输

I2S 的主要特点

- ◆ 半双工通信（仅发送器或接收器）
- ◆ 主机或从机操作
- ◆ 8 位可编程线性预分频器，可达到精确的音频采样频率（从 8 kHz 到 96 kHz）
- ◆ 数据格式可以是 16 位，24 位或 32 位
- ◆ 通过音频信道将信道帧固定为 16 位（16 位数据帧）或 32 位（16 位，24 位，32 位数据帧）
- ◆ 可编程时钟极性（稳态）
- ◆ 提供主机和从机模式下接收与发送的下溢与溢出标志，从机在接收与发送模式下的帧错误标志
- ◆ 16 位寄存器，用于发送和接收
- ◆ 支持的 I2S 协议：
 - ◇ I2S 飞利浦标准
 - ◇ MSB 对齐标准（左对齐）
 - ◇ LSB 对齐标准（右对齐）
 - ◇ PCM 标准（16 位通道帧上的短帧和长帧同步或扩展到 32 位通道帧的 16 位数据帧）
- ◆ 数据方向始终为 MSB 优先
- ◆ 发送和接收的 DMA 功能（16 位宽）
- ◆ 可输出主时钟以驱动外部音频组件。比率固定为 $256 \times F_s$ （ F_s 为音频采样频率）
- ◆ 提供发送和接收 FIFO，深度为 16
- ◆ 提供两个可选的外部时钟输入引脚

24.3 SPI 结构图

SPI 允许 MCU 与外部设备之间进行同步串行通信。应用软件可以通过轮询状态标志或使用专用 SPI 中断来管理通信。SPI 的主要模块及其相互关系如下面的框图所示。

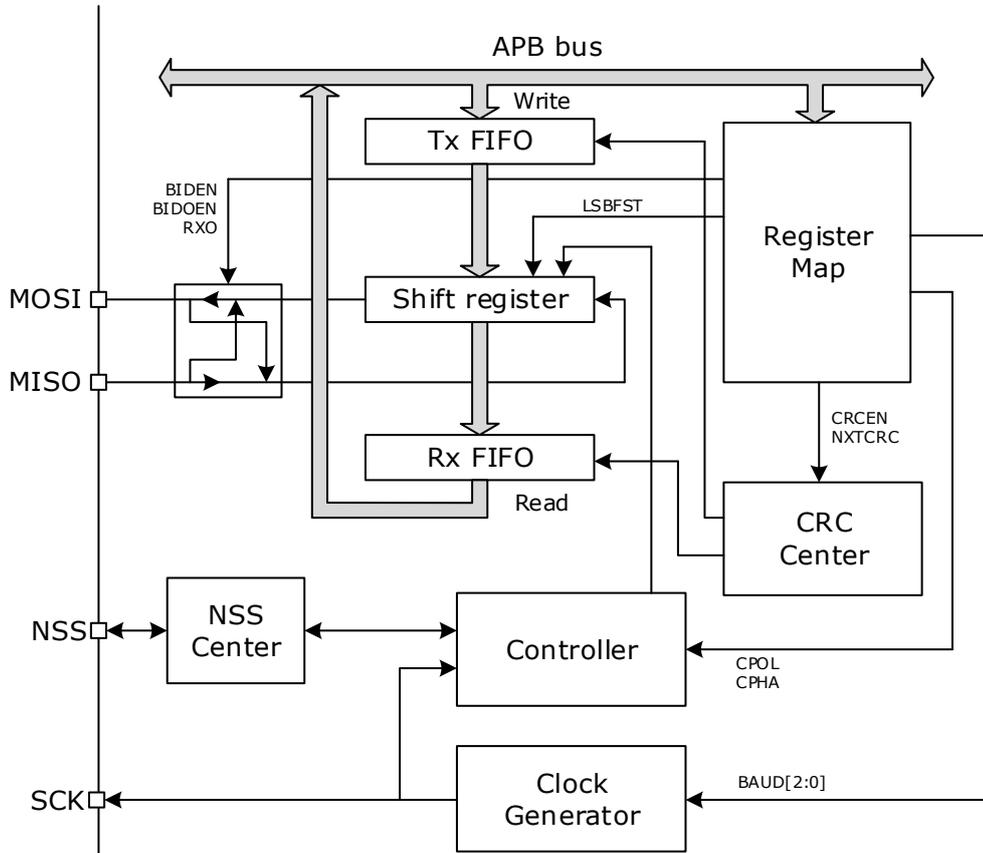


图 24-1 SPI 电路结构框图

通常，SPI 使用四个 I/O 引脚来与外部设备进行通信。

- ◆ **MISO:** 主机输入/从机输出数据。
- ◆ **MOSI:** 主机输出/从机输入数据。
- ◆ **SCK:** 用于 SPI 主机的串行时钟输出以及 SPI 从机的串行时钟输入。
- ◆ **NSS:** 从机选择引脚。这是用于选择从机的可选引脚。此引脚用作“片选”，可让 SPI 主机与从机进行单独通信，从而避免数据线上的竞争。从机的 NSS 输入可由主机上的标准 IO 端口驱动。NSS 引脚在使能（SPI_CON2.NSSOE 位）时还可用作输出，并可在 SPI 处于主模式配置时驱动为低电平。通过这种方式，只要器件配置成 NSS 硬件控制模式，所有连接到该主机 NSS 引脚的其它器件 NSS 引脚都将呈现低电平，并因此而作为从机。当配置为主模式，且 NSS 配置为输入（SPI_CON1.MSTREN=1 且 SPI_CON2.NSSOE=0）时，如果 NSS 拉至低电平，SPI 将进入模式错误状态：SPI_CON1.MSTREN 位自动清零，并且器件配置为从模式（参见章节 **错误! 找不到参照来源。**：SPI 错误标志）

24.4 SPI 功能描述

24.4.1 通信格式

在 SPI 通信期间，接收和发送操作同时执行。串行时钟（SCK）同步数据线上讯息的移位和采样。通信格式取决于时钟相位，时钟极性和数据帧格式。为了能够一起通信，主机和从机必须遵循相同的通信格式。

注意：在通过从机通讯时，从机所使用的系统频率需大于总线上的传输频率 8 倍以上才可正常工作。

24.4.1.1 时钟相位和极性控制

通过配置 **SPI_CON1.CPOL** 和 **SPI_CON1.CPHA** 位，可以用软件选择四种可能的时序关系。**SPI_CON1.CPOL**（时钟极性）位控制空闲时时钟线上的电平状态，此位对主机和从机都有作用。如果复位 **SPI_CON1.CPOL** 位，**SCK** 引脚在空闲状态时处于低电平。如果将 **SPI_CON1.CPOL** 位置 1，**SCK** 引脚在空闲状态时处于高电平。

如果将 **SPI_CON1.CPHA** 位置 1，则 **SCK** 引脚上的第二个边沿（如果 **SPI_CON1.CPOL** 位配置为 0，则为下降沿；如果 **SPI_CON1.CPOL** 位配置为 1，则为上升沿）对 **MSB** 采样。即在第二个时钟边沿锁存数据。如果复位 **CPHA** 位，则 **SCK** 引脚上的第一个边沿（如果 **SPI_CON1.CPOL** 位配置为 0，则为上升沿；如果 **SPI_CON1.CPOL** 位配置为 1，则为下降沿）对 **MSB** 采样。即在第一个时钟边沿锁存数据。

用户通过组合 **SPI_CON1.CPOL** 和 **SPI_CON1.CPHA** 位来选择数据捕获的时钟边沿。

下图显示了在 **SPI_CON1.CPHA** 和 **SPI_CON1.CPOL** 位的四种组合下的 SPI 传输。可以将该图解释为主机或从机时序图，其中 **SCK** 引脚、**MISO** 引脚、**MOSI** 引脚直接连接在主机和从机之间。

注意：在切换 **SPI_CON1.CPOL/SPI_CON1.CPHA** 位之前，必须通过复位 **SPI_CON1.SPIEN** 位来关闭 SPI。

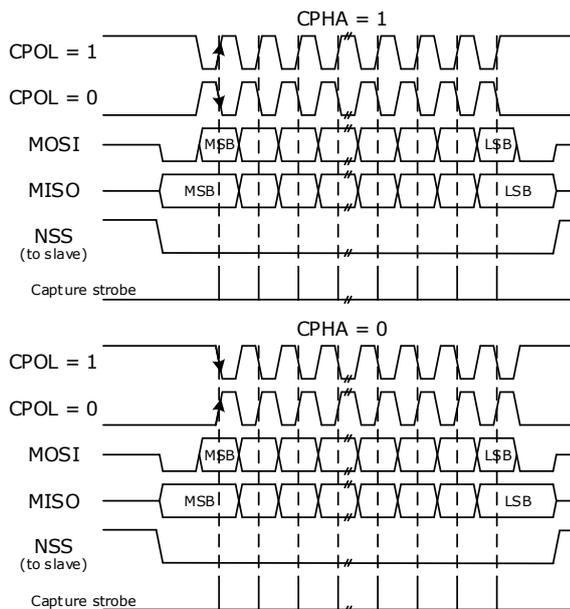


图 24-2 SPI 格式

注意：数据位的顺序取决于 **LSBFST** 位的设置。

24.4.1.2 数据帧格式

SPI 移位寄存器可以设置为移出 MSB 优先或 LSB 优先, 具体取决于 **SPI_CON1.LSBFST** 位的值。每个数据帧为 8 或 16 位长, 具体取决于 **SPI_CON1.FLEN** 位的配置。所选的数据帧长度适用于发送和/或接收。

24.4.2 从机选择 (NSS) 引脚管理

可以使用 **SPI_CON1.SSEN** 位设置硬件或软件控制从机选择。

- ◇ 软件控制 NSS (**SPI_CON1.SSEN = 1**) 从机选择讯息在内部由 **SPI_CON1.SSOUT** 位的值驱动。
- ◇ 硬件管理 NSS (**SPI_CON1.SSEN = 0**) 根据 NSS 输出配置 (**SPI_CON2.NSSOE** 位), 硬件管理 NSS 有两种模式。
 - NSS 输出使能 (**SPI_CON1.SSEN = 0, SPI_CON2.NSSOE = 1**) 仅当器件在主模式工作时才使用此配置。当主机开始传输数据时, NSS 信号驱动为低电平, 并保持到数据传输结束为止。
 - NSS 输出禁止 (**SPI_CON1.SSEN = 0, SPI_CON2.NSSOE = 0**) 对于在主模式下工作的器件, 此配置允许多主模式功能。对于设置为从机模式的器件, NSS 引脚用作传统 NSS 输入: 在 NSS 为低电平时片选该从机, 在 NSS 为高电平时取消对它的片选。

24.4.3 单对单应用

SPI 允许 MCU 使用不同的配置进行通信, 具体取决于所针对的设备和应用要求。这些配置使用 2 或 3 线 (使用软件 NSS 管理) 或 3 或 4 线 (使用硬件 NSS 管理)。通信始终由主机启动。

24.4.3.1 全双工通信

默认情况下, SPI 配置为全双工通信。在此配置中, MOSI 引脚连接在一起, MISO 引脚连接在一起。通过这种方式, 主机和从机之间以串行方式传输数据 (最高有效位在前)。

通信始终由主机发起。当主机通过 MOSI 引脚向从机发送数据时, 从机同时通过 MISO 引脚发出准备好的数据。这是一个数据输出和数据输入都由同一时钟进行同步的全双工通信过程, 时钟信号由主机的 SCK 引脚发出提供给从机。

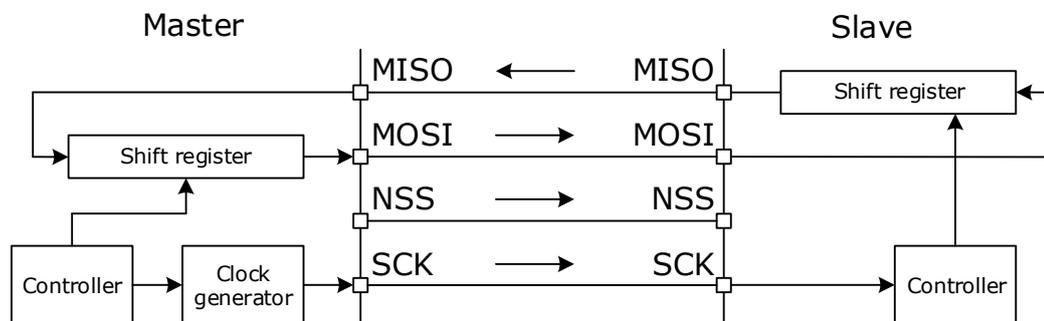


图 24-3 全双工通信

24.4.3.2 半双工通信

SPI 能够在以下两种配置中以半双工模式工作。

1 个时钟和 1 条双向数据线 (SPI_CON1.BIDEN=1)

可将 SPI_CON1.BIDEN 位置 1 来使能此模式。在此模式下, SCK 作为时钟信号输出引脚, MOSI (主模式下) 或 MISO (从模式下) 作为数据通信引脚。通过 SPI_CON1.BIDOEN 位来选择传输方向 (输入/输出)。当该位置 1 时, 数据线为输出, 否则为输入。

1 个时钟和 1 条单向数据线 (SPI_CON1.BIDEN=0)

在此模式下, 应用程序可使用 SPI 的只发送或只接收功能。

- ◇ 只发送模式类似于全双工模式: 在发送引脚 (主模式下的 MOSI 或从模式下的 MISO) 上发送数据, 不再接收数据。
- ◇ 只接收模式下, 应用程序可将 SPI_CON2.RXO 位置 1 来关闭 SPI 输出功能。

当 SPI 进入只读模式后:

- ◇ 一旦在主模式下使能 SPI 后, 主机会等待用户写入数据之后开始从 SCK 引脚发送时钟, 需要读回多少个数据由写入无效数据的个数来决定, 当 Tx FIFO 中的无效的数据发送完毕后, 通信也立即停止。
- ◇ 在从模式下, 只要 NSS 引脚被拉低 (或在 NSS 软件模式下将 SPI_CON1.SSOUT 位清零), 意味着从机被选中, 同时一直有来自主机的 SCK 输入, SPI 就会继续接收。

24.4.4 数据发送和接收

接收和发送 FIFO 缓存

所有 SPI 数据传输都通过嵌入式 16 级深度的 FIFO 缓存。使 SPI 能够连续传输工作, 并在数据帧长度较短时防止接收溢出。发送和接收都有自己的 FIFO 缓存。

对 SPI_DATA 寄存器的读访问将返回存储在接收 FIFO 缓存中但尚未读取的最旧的值。对 SPI_DATA 的写访问将已写数据存储在发送队列末尾的发送 FIFO 缓存中。SPI_STAT 寄存器中 RXFLV [4:0]和 TXFLV [4:0]位域指示两个 FIFO 缓存的有效数据个数。

对 SPI_DATA 寄存器的读访问必须由 RXTH 事件管理。当数据存储在接收 FIFO 缓存中并且达到阈值 (由 SPI_CON2 寄存器中 RXFTH 位域定义) 时, 触发此事件。当 RXTH 被清除时, 表示接收 FIFO 缓存中的有效数据个数小于阈值。以类似的方式, 要发送的数据帧的写访问由 TXTH 事件管理。当发送 FIFO 缓存有效数据个数小于或等于阈值 (由 SPI_CON2 寄存器中 TXFTH 位域定义) 时将触发此事件。

在主机模式下启动通信序列

- ◇ 在全双工通信 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)
 - 将数据写入到 SPI_DATA 寄存器 (发送 FIFO 缓存) 后, 启动通信序列。
 - 随后在第一个位的发送期间, 将数据从发送 FIFO 缓存并行加载到 8 位移位寄存器中, 然后以串行方式将其移出到 MOSI 引脚。
 - 同时, 将 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器, 然后并行加载到 SPI_DATA 寄存器 (接收 FIFO 缓存) 中。
- ◇ 在单工通信-只接收模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1)
 - 将无效数据写入到 SPI_DATA 寄存器 (发送 FIFO 缓存) 后, 启动通信序列。
 - 随后在第一个位的发送期间, 将数据从发送 FIFO 缓存并行加载到 8 位移位寄存器中, 然后以串行方式将其移出到 MOSI 引脚。

- 同时，将 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DATA 寄存器（接收 FIFO 缓存）中。
- ◇ 在半双工通信-发送模式（SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=1）
 - 将数据写入到 SPI_DATA 寄存器（发送 FIFO 缓存）时，通信序列启动。
 - 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
 - 不接收任何数据。
- ◇ 在半双工通信-接收模式（SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=0）
 - 只要 SPI_CON1.SPIEN=1 且 SPI_CON1.BIDOEN=0。
 - 将无效数据写入到 SPI_DATA 寄存器（发送 FIFO 缓存）后，启动通信序列。
 - 随后在第一个位的发送期间，将数据从发送 FIFO 缓存并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
 - 同时，将 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DATA 寄存器（接收 FIFO 缓存）中。

在从机模式下启动通信序列

- ◇ 在全双工模式（SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0）
 - 当从机在其 MOSI 引脚上收到时钟信号和数据的第一个位时，通信序列开始。其余 7 个位将加载到移位寄存器中。
 - 同时，在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MISO 引脚。在 SPI 主机启动传输前，软件必须已把要从机发送的数据写入发送 FIFO 缓存。
- ◇ 在单工通信-只接收模式（SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1）
 - 当从机在其 MOSI 引脚上收到时钟信号和数据的第一个位时，通信序列开始。其余 7 个位将加载到移位寄存器中。
 - 由于发送器没有激活，因此不会有数据以串行方式移出 MISO 引脚。
- ◇ 在半双工通信-发送模式（SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=1）
 - 当从机收到时钟信号，并且 MISO 引脚上发出发送 FIFO 缓存中的第一位数据时，通信序列开始。
 - 随后在第一个位的发送期间，将数据从发送 FIFO 缓存并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MISO 引脚。在 SPI 主机启动传输前，软件必须已把要从机发送的数据写入发送 FIFO 缓存。
 - 不接收任何数据。
- ◇ 在半双工通信-接收模式（SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=0）
 - 当从机在其 MISO 引脚上收到时钟信号和数据的第一个位时，通信序列开始。
 - 在 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到

SPI_DATA 寄存器（接收 FIFO 缓存）中。

- 由于发送器没有被激活，因此不会有数据以串行方式移出 MISO 引脚。

处理数据发送与接收

全双工通信（**SPI_CON1.BIDEN=0** 且 **SPI_CON1.RXO=0**），发送和接收数据的处理过程直接存取操作模式（参见图 24-4）：

1. 通过将 **SPI_CON1.SPIEN** 位置 1 来使能 SPI，将第一个要发送的数据项写入 **SPI_DATA** 寄存器（此操作会将 **SPI_STAT.TXE** 位清零）。
2. 等待 **SPI_STAT.TXE=1**，然后写入要发送的第二个数据项。然后等待 **SPI_STAT.RXE=0**，读取 **SPI_DATA** 以获取第一个接收到的数据（此操作会将 **SPI_STAT.RXE** 位置 1）。对每个要发送和接收的数据项重复此操作，直到发送并接收完最后的数据。
3. 检查 **SPI_STAT.TXE=1**，然后等待至 **SPI_STAT.BUSY=0**，再关闭 SPI。
4. 此外，还可以使用 TXE 或 RXTH(将 **SPI_CON2.RXFTH** 位域置 1)中断事件对应的各个中断子程序来实现该过程。

FIFO 缓存操作模式（参见图 24-5）：

1. 通过将 **SPI_CON1.SPIEN** 位置 1 来使能 SPI。
2. 配置 **SPI_CON2.TXFTH** 與 **SPI_CON2.RXFTH**。
3. 当 **SPI_STAT.TXTH=1**，将要发送的数据写入 **SPI_DATA** 寄存器（写入的数据个数必须大于 **SPI_CON2.TXFTH** 设定的阈值），当 **SPI_STAT.RXTH=1**，读取 **SPI_DATA** 寄存器以获取接收到的数据（读取的数据个数必须为 **SPI_CON2.RXFTH** 设定的阈值），重复此操作直到写入最后要发送的数据。
4. 等待至 **SPI_STAT.BUSY=0**，读取 **SPI_DATA** 寄存器以获取接收到的数据直到 **SPI_STAT.RXFLV** 位置 0，再关闭 SPI。
5. 此外，还可以使用 TXTH 或 RXTH 中断事件对应的各个中断子程序来实现该过程。

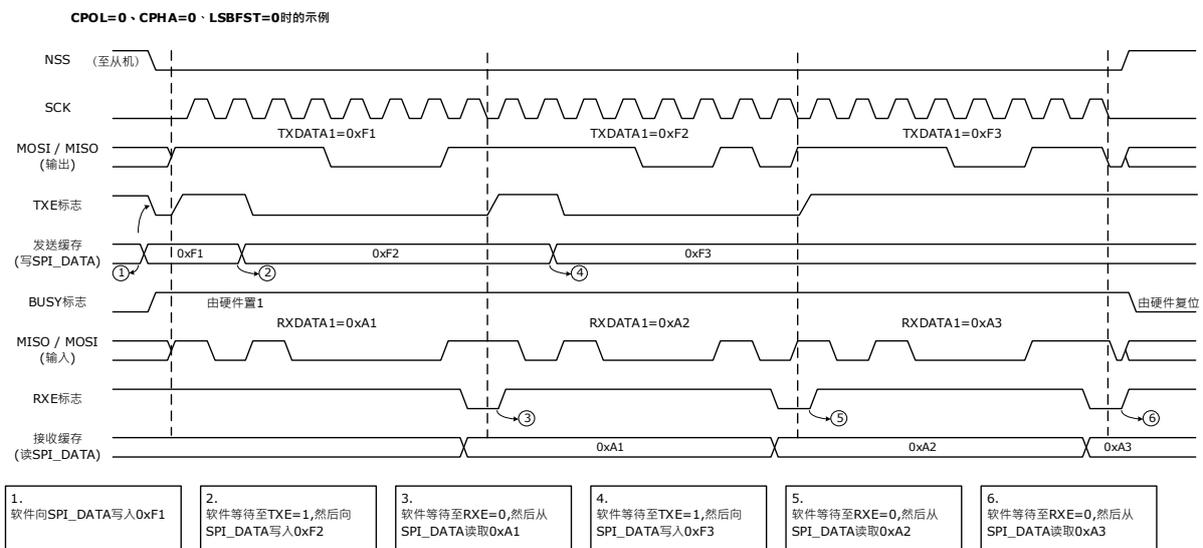


图 24-4 全双工通信(**SPI_CON1.BIDEN=0** 且 **SPI_CON1.RXO=0**)的 TXE、RXE、BUSY 行为(直接存取操作模式)

取操作模式在连续传输的情况下)

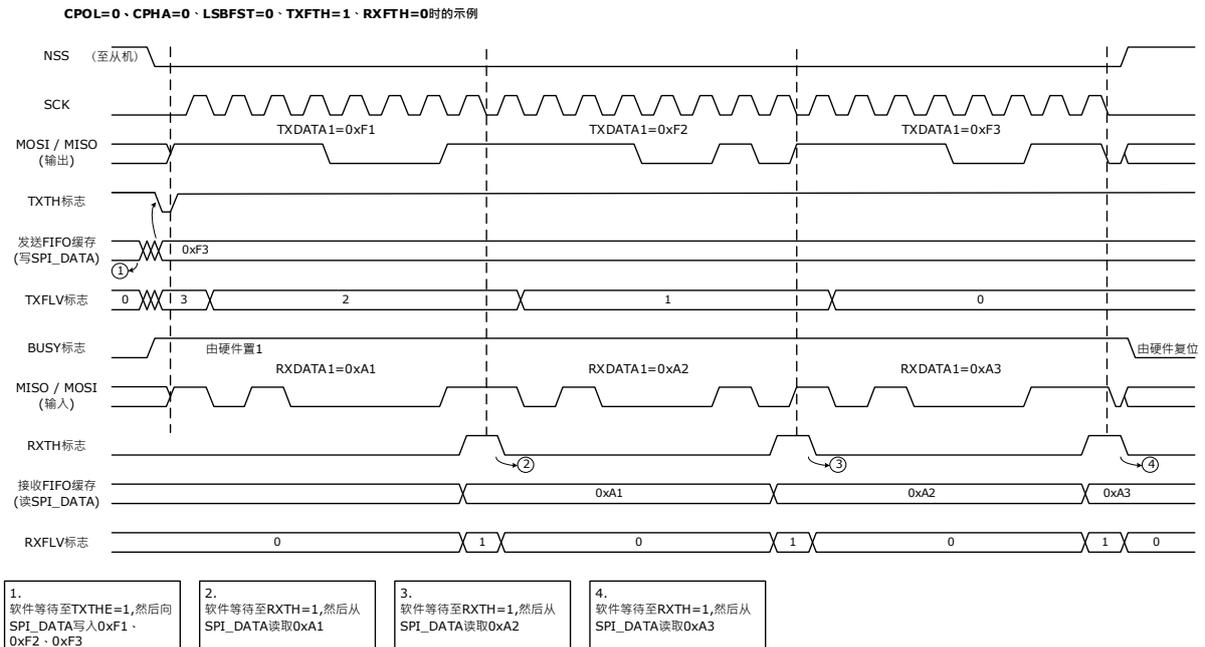


图 24-5 全双工通信(SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)的 TXTH、RXTH、TXFLV、RXFLV、BUSY 行为(FIFO 缓存操作模式在连续传输的情况下)

单工通信-只发送模式 (SPI_CON1.BIDEN=0、SPI_CON1.RXO=0)，发送数据的处理过程直接存取操作模式 (参见图 24-6)：

1. 通过将 SPI_CON1.SPIEN 位置 1 来使能 SPI。
2. 等待 SPI_STAT.TXE=1 然后写入要发送的数据。对每个要发送的数据项重复此步骤。
3. 将最后一个数据写入 SPI_DATA 寄存器后，等待至 SPI_STAT.TXE=1，然后等待至 SPI_STAT.BUSY=0 再关闭 SPI，这表示最后的数据发送完成。
4. 此外，还可以使用在 TXE 中断事件对应的中断子程序来实现该过程。

FIFO 缓存操作模式 (参见图 24-7)：

1. 通过将 SPI_CON1.SPIEN 位置 1 来使能 SPI。
2. 配置 SPI_CON2.TXFTH。
3. 当 SPI_STAT.TXTH=1，将要发送的数据写入 SPI_DATA 寄存器 (写入的数据个数必须大于 SPI_CON2.TXFTH 设定的阈值)，重复此操作直到写入最后要发送的数据。
4. 等待至 SPI_STAT.BUSY=0 再关闭 SPI。
5. 此外，还可以使用 TXTH 中断事件对应的中断子程序来实现该过程。

注意：

注 1：在不连续通信期间，在对 SPI_DATA 寄存器执行写操作与 SPI_STAT.BUSY 位置 1 之间有延迟。因此在只发送模式下，写入最后的数据后，必须先等待 SPI_STAT.TXE 位置 1，然后等待 SPI_STAT.BUSY 位清零。

注 2：在只发送模式下，发送 17 个数据项后，SPI_STAT.RXOV 标志将置 1，因为始终不会读取接收的数据。

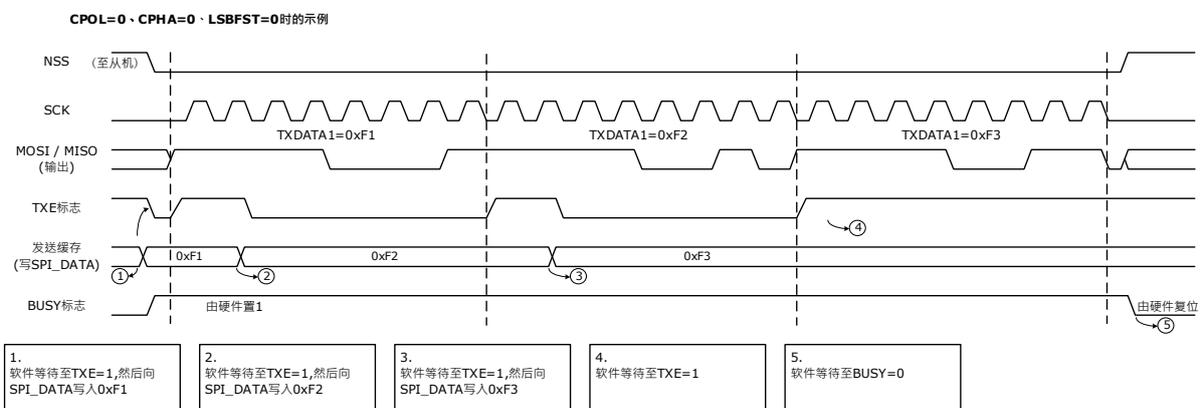


图 24-6 单工通信-只发送模式(SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)的 TXE、BUSY 行为(直接存取操作模式在连续传输的情况下)

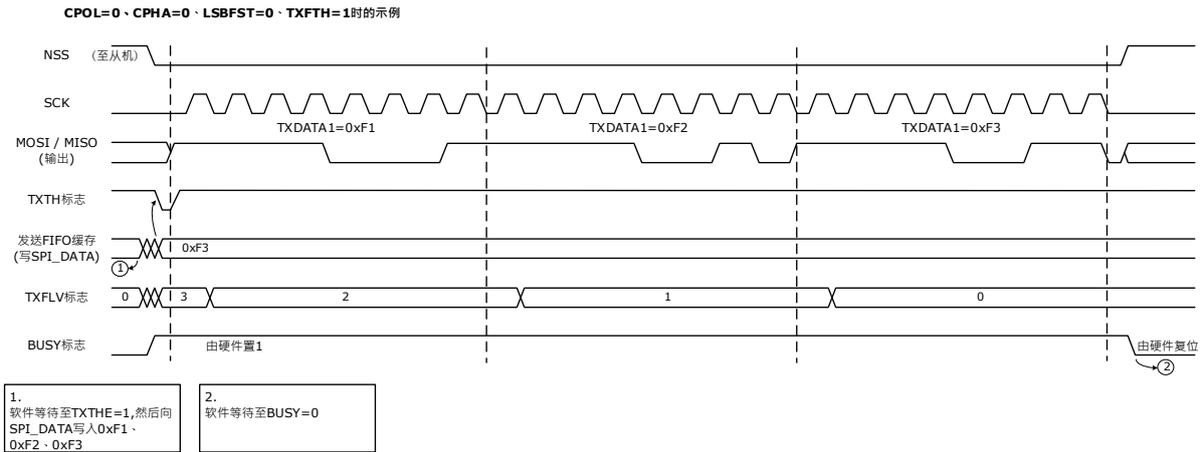


图 24-7 单工通信-只发送模式(SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)的 TXTH、TXFLV、BUSY 行为(FIFO 缓存操作模式在连续传输的情况下)

半双工通信-发送模式 (SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=1), 发送数据的处理过程

此模式与单工通信-只发送模式数据的处理过相似, 但是在 SPI 模块使能前, 必须将 SPI_CON1.BIDEN 位和 SPI_CON1.BIDOEN 位置 1。

单工通信-只接收模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1), 接收数据的处理过程直接存取操作模式 (参见图 24-8) :

1. 将 SPI_CON2.RXO 位置 1。
2. 通过将 SPI_CON1.SPIEN 位置 1 使能 SPI。
3. 在主机模式下。等待 SPI_STAT.TXE=1, 然后写入无效数据写入到 SPI_DATA 寄存器。等待 SPI_STAT.RXE=0, 然后读取 SPI_DATA 寄存器以获取接收的数据 (此操作会将 SPI_STAT.RXE 位置 1)。对每个要接收的数据项重复此操作。
4. 在从机模式下。等待 SPI_STAT.RXE=0, 然后读取 SPI_DATA 寄存器以获取接收的数据 (此操作会将 SPI_STAT.RXE 位置 1)。对每个要接收的数据项重复此操作。
5. 此外, 还可以使用 RXTH(将 SPI_CON2.RXFTH 位域置 1)中断事件对应的中断子程序来实现该过程。

FIFO 缓存操作模式 (参见图 24-9) :

1. 将 SPI_CON2.RXO 位置 1。
2. 配置 SPI_CON2.RXFTH。
3. 通过将 SPI_CON1.SPIEN 位置 1 使能 SPI。
4. 在主机模式下。当 SPI_STAT.TXTH=1, 将无效数据写入 SPI_DATA 寄存器 (写入的数据个数必须大于 SPI_CON2.TXFTH 设定的阈值), 当 SPI_STAT.RXTH=1, 读取 SPI_DATA 寄存器以获取接收到的数据 (读取的数据个数必须为 SPI_CON2.RXFTH 设定的阈值), 重复此操作直到获取最后要接收的数据。
5. 在从机模式下。当 SPI_STAT.RXTH=1, 读取 SPI_DATA 寄存器以获取接收到的数据 (读取的数据个数必须为 PI_CON2.RXFTH 设定的阈值), 重复此操作直到获取最后要接收的数据。

6. 此外，还可以使用 RXTH 中断事件对应的中断子程序来实现该过程。

注意：
 注 1：在主机模式下，一旦 SPI 使能后 SCK 会立即发送时钟，从机接收到时钟后会发送数据，直到主机关闭 SPI 功能结束通信。
 注 2：在从机模式下，当 NSS 被拉低并且接收到 SCK 时钟后开始接收数据。

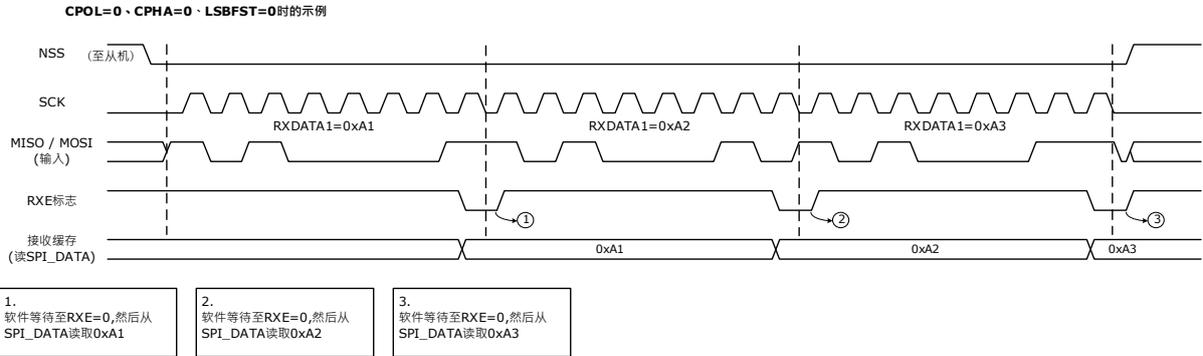


图 24-8 单工通信-只接收模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1) 的 RXE 行为(直接存取操作模式在连续传输的情况下)

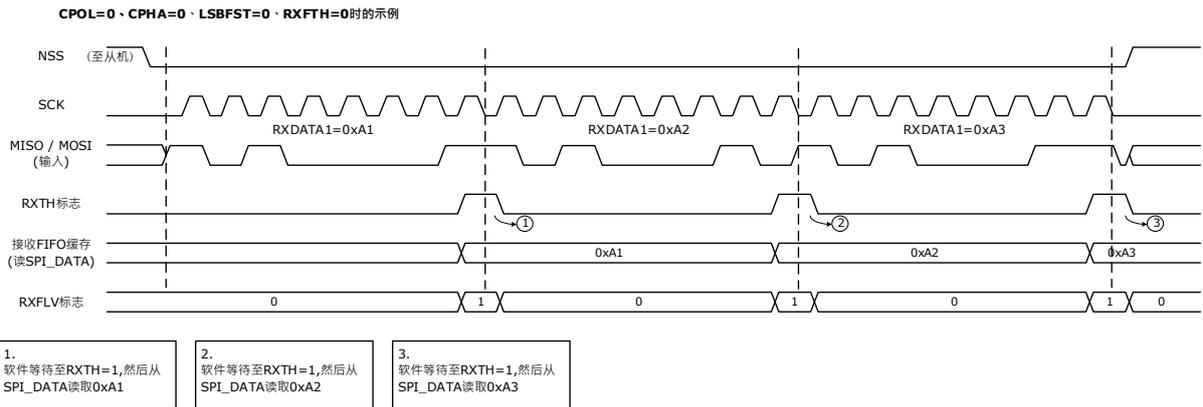


图 24-9 单工通信-只接收模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1) 的 RXTH、RXFLV 行为 (FIFO 缓存操作模式在连续传输的情况下)

半双工通信-接收模式 (SPI_CON1.BIDEN=1 和 SPI_CON1.BIDOEN=0)，接收数据的处理过程

此模式与单工通信-只接收模式数据的处理过程相似，但是在 SPI 模块使能之前，需要将 SPI_CON1.BIDEN 位置 1,并将 SPI_CON1.BIDOEN 与 SPI_CON1.RXO 位清 0。

连续传输和间断传输

在主机模式下发送数据时，如果软件处理速度足够快，可以在检测到 SPI_STAT.TXE=1（或发生 TXE 中断事件），并且当前数据传输未结束，立即将下一次的数据写入 SPI_DATA 寄存器，则能实现连续的通信。或者配置 SPI_CON2.TXFTH 检测当 SPI_STAT.TXTH=1（或发生 TXTH 中断事件），将要发送的数据写入 SPI_DATA 寄存器（写入的数据个数必须大于 SPI_CON2.TXFTH 设定的阈值），实现连续的通信。观察到的现象是 SPI_STAT.BUSY 位一直为 1 不被清除，并且每个数据的 SPI 时钟保持连续。

相反，如果软件速度不够快，则可能导致通信中断。在这种情况下，各数据传输之间会清零 SPI_STAT.BUSY 位。

在主机或从机模式下的单工通信-只接收模式 (SPI_CON1.RXO=1)，通信始终是连续的，且 SPI_STAT.BUSY 位始终为 1。

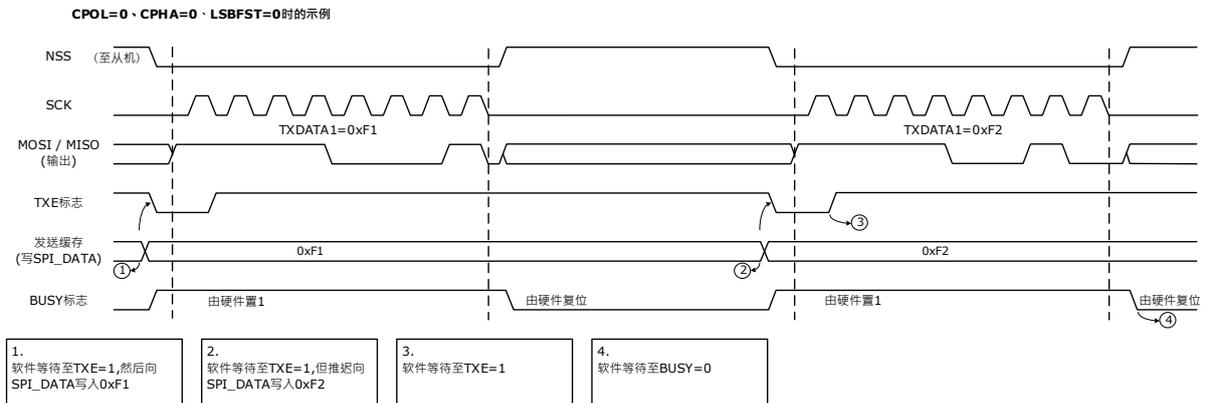


图 24-10 发送时 (SPI_CON1.BIDEN = 0 且 SPI_CON1.RXO=0) 的 TXE/BUSY 行为 (在间断传输的情况下)

24.4.5 DMA 请求

为了更方便的实现高速通信，SPI 提供了 DMA 功能。DMA 请求条件是根据 SPI_CON2 寄存器中的 TXFTH 与 RXFTH 位域配置，当使能 SPI_CON2 寄存器中相应的 DMA 使能位时，将请求 DMA 访问。发送 FIFO 缓存和接收 FIFO 缓存会发出各自的 DMA 请求（参见图 26-11 和图 24-12）：

- ◇ 在发送过程中，当 SPI_STAT.TXTH 位置 1 时会发出 DMA 请求。DMA 随后对 SPI_DATA 寄存器执行写操作（此操作会将 SPI_STAT.TXTH 位清零）。
- ◇ 在接收过程中，当 SPI_STAT.RXTH 位置 1 时会发出 DMA 请求。DMA 随后对 SPI_DATA 寄存器执行读操作（此操作会将 SPI_STAT.RXTH 位清零）。

当 SPI 仅用于只发送数据时，可以只使能 SPI Tx DMA 通道。在这种情况下，SPI_STAT.RXOV 位会置 1，因为未读取接收的数据。

在发送模式下，DMA 完成了所有要发送数据的传输后，DMA_RIF 会产生相对应通道的传输完成标志，使用者可以对 BUSY 标志进行监视，以确保 SPI 通信已完成。在关闭 SPI 或进入停止模式前必须等待 SPI_STAT.BUSY=0。

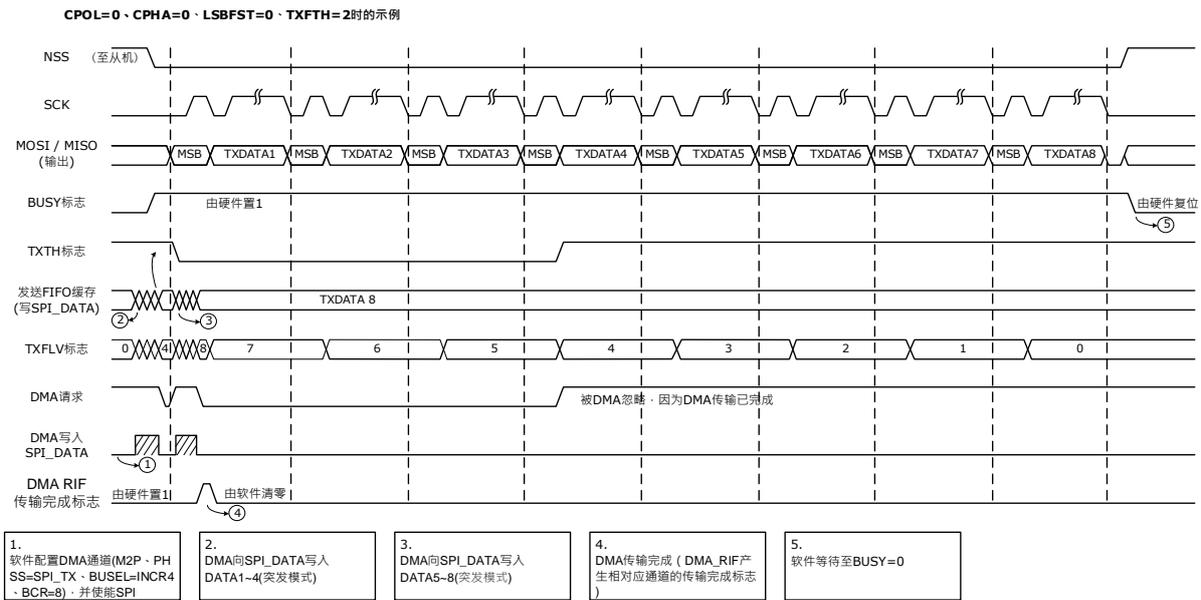


图 24-11 使用 DMA 进行发送

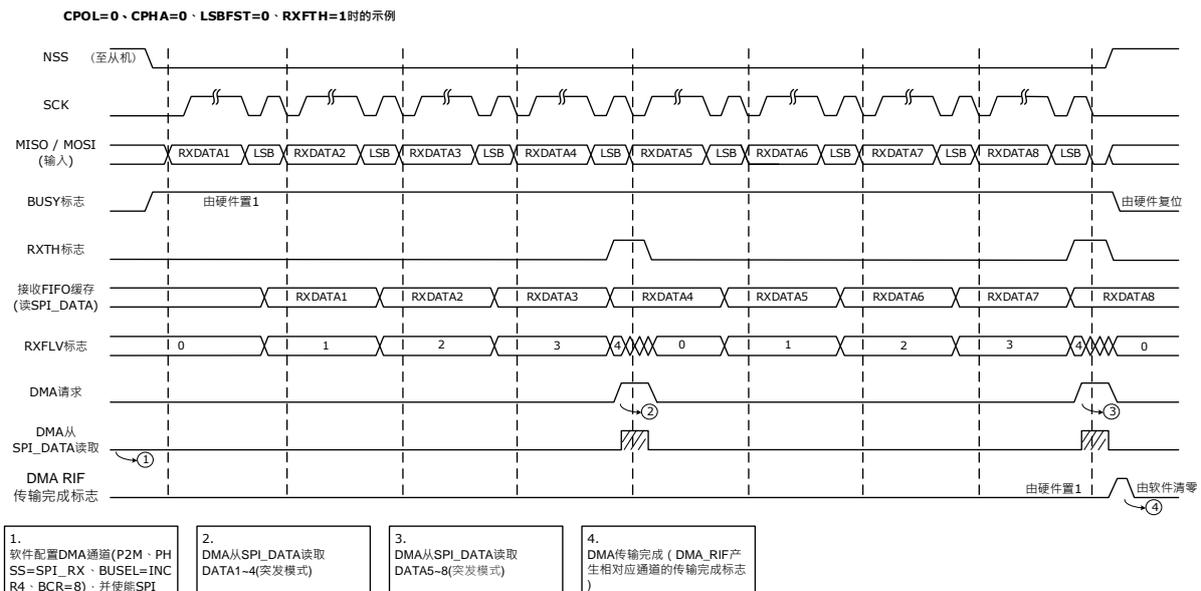


图 24-12 使用 DMA 进行接收

24.4.6 SPI 状态标志

为应用程序提供了五个状态标志，以完全监视 SPI 总线的状态。

24.4.6.1 发送 FIFO 空标志 (TXE)

设置时，TXE 标志指示发送 FIFO 为空，并且下一个要传输的数据可以加载到 FIFO 中。通过写 SPI_DATA 寄存器清零 TXE 标志。

24.4.6.2 发送 FIFO 满标志 (TXF)

设置时，TXF 标志指示发送 FIFO 已满，并且下一个要传输的数据无法加载到 FIFO 中。TXF 标志由要发送的下一个数据清除。

24.4.6.3 接收 FIFO 空标志 (RXE)

设置时, RXE 标志指示接收 FIFO 为空, 并且下一个要传输的数据可以加载到 FIFO 中。RXE 标志由下一个数据接收到 FIFO 中清除。

24.4.6.4 接收 FIFO 满标志 (RXF)

设置后, RXF 标志指示接收 FIFO 已满, 并且下一个要接收的数据无法加载到 FIFO 中。通过对 **SPI_DATA** 寄存器的读访问来清除 RXF 标志。

24.4.6.5 通信忙 (BUSY)

BUSY 标志用于指示 SPI 通信的状态。此标志由硬件置 1 和清零。

SPI_STAT.BUSY 位置 1 时, 表示 SPI 正在通信中。在通信结束前, 用户可检测 **SPI_STAT.BUSY** 位是否为 0, 表示通信已结束, 此时关闭 SPI 模块, 停止通信。

BUSY 标志还可用于避免在多主模式系统中发生写冲突。

在以下情况硬件将清零该标志:

- ◇ 传输完成时 (主模式下的连续通信除外)
- ◇ 关闭 SPI 时
- ◇ 发生模式错误时 (**SPI_STAT.MODERR=1**)

当通信不连续时, BUSY 标志在各通信之间处于低电平。

当通信连续时:

- ◇ 在主模式下, BUSY 标志在所有传输期间均保持高电平
- ◇ 在从模式下, BUSY 标志在各传输之间的一个 SPI 时钟周期内变为低电平

注意: 建议始终使用 TXTH 和 RXTH 标志 (而不是 BUSY 标志) 来处理数据传输或接收操作。

24.4.7 SPI 错误标志

如果设置了以下错误标志之一并通过将该位置 1 启用中断，则会产生 SPI 中断。

24.4.7.1 发送 FIFO 溢出标志 (TXOV)

发生发送溢出的条件是发送 FIFO 已满，但用户正在执行写操作。在这种情况下，新写入的数据不会加载到发送 FIFO 中。通过对 SPI_ICR 寄存器的写访问来清除 TXOV 位，或者在新传输开始时它将自动清零。

24.4.7.2 接收 FIFO 溢出标志 (RXOV)

当主机或从机完成下一个数据帧的接收而接收 FIFO 的前 16 帧的读操作尚未完成时（设置了 RXF 标志的情况），发生接收溢出条件。

在这种情况下，接收 FIFO 的内容不会将接收的新数据存入。SPI_DATA 寄存器的读操作返回先前接收的帧。所有其他在 RXF 标志设置的情况下，接收的新数据都将丢失。

通过对 SPI_ICR 寄存器的写访问来清除 RXOV 位，或者在新的传输开始时它将自动清零。

24.4.7.3 接收 FIFO 下溢标志 (RXUD)

当接收 FIFO 为空时，但用户正在执行读操作，会发生接收下溢条件。在这种情况下，读取操作不会从接收 FIFO 中读到有效的数据。

通过对 SPI_ICR 寄存器的写访问来清除 RXUD 位，或者在新的传输开始时它将自动清零。

24.4.7.4 模式故障 (MODF)

当主机的内部 NSS 信号（NSS 硬件模式下的 NSS 引脚或 NSS 软件模式下的 SSOUT 位）拉低时，发生模式故障。这会设置 MODF 位。主机模式故障通过以下方式影响 SPI 接口：

- ◆ 如果 MODFIE 位置 1，则 MODF 位置 1，产生 SPI 中断。
- ◆ SPIEN 位清零。这将阻止来自 SPI 的所有输出并禁用 SPI 接口。
- ◆ MSTREN 位清零，从而强制 SPI 进入从机模式。

通过对 SPI_ICR 寄存器的写访问来清除 MODF 位

为避免在包含多个 MCU 的系统中发生任何多个从机冲突，必须在 MODF 位清零序列期间将 NSS 引脚拉高。在此清除序列之后，SPIEN 和 MSTREN 位可以恢复到其原始状态。在从机中，除了由于先前的多主机冲突之外，不能设置 MODF 位。

24.4.7.5 TI 模式帧格式错误 (FRE)

当 SPI 在从机模式下工作并配置为符合 TI 模式协议时，在正在进行的通信期间发生 NSS 脉冲时，会检测到 TI 模式帧格式错误。发生此错误时，将在 SPI_RIF 寄存器中设置 FRERI 标志。发生错误时，SPI 不会被禁用，NSS 脉冲被忽略，SPI 在开始新的传输之前等待下一个 NSS 脉冲。数据可能被破坏，因为错误检测可能导致两个数据字节的丢失。

如果 FREIE 位置 1，则在 NSS 错误检测时产生中断。在这种情况下，应禁用 SPI，因为不再保证数据一致性，并且当再次启用从 SPI 时，主机应重新启动通信。

24.4.8 SPI TI 模式

SPI 接口与 TI 协议兼容。SPI_CON2 寄存器的 FRF 位可用于配置 SPI 以符合此协议。

无论 SPI_CON1 寄存器中设置的值如何，时钟极性和相位都必须符合 TI 协议要求。NSS 管理也特定于 TI 协议，在这种情况下，通过 SPI_CON1 和 SPI_CON2 寄存器 (SSEN, SSOUT, NSSOE) 无法配置 NSS 管理。

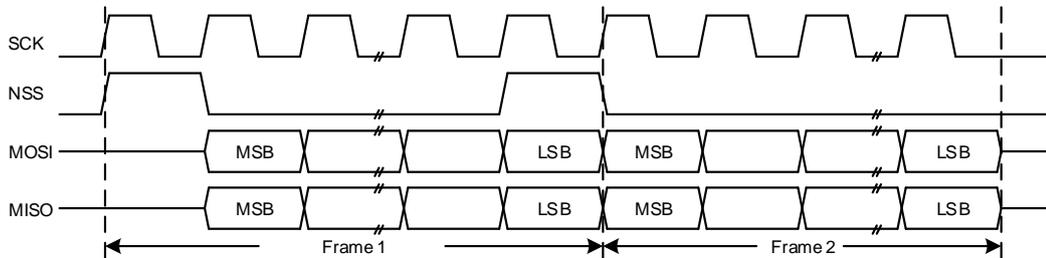


图 24-13 TI 格式

24.4.9 CRC 计算

为确保通信的可靠性，SPI 模块实现了硬件 CRC 功能。

针对发送或接收的数据帧宽度有 8 位和 16 位的选择，硬件 CRC 计算也提供了两种计算标准，分别为 8 位数据的 CRC8 和 16 位数据的 CRC16。CRC 是使用 SPI_CRCPOLY 寄存器中编程的多项式串行计算的。

将 SPI_CON1.CRCEN 位置 1 来使能 CRC 计算功能，此操作会复位 CRC 寄存器 (SPI_RXCRC 和 SPI_TXCRC)。仅支持由软件 (CPU 模式) 管理传输。

在全双工或只发送模式下，在连续传输的情况下，可以在最后一笔数据写入前任意时间点将 SPI_CON1.NXTCRC 位置 1，当最后一次数据传输结束时，将发送 SPI_TXCRC 寄存器内的值。在间断传输的情况下，必须在最后传输的数据写入 SPI_DATA 后，立即对 SPI_CON1.NXTCRC 位执行写操作，当最后一次数据传输结束时，将发送 SPI_TXCRC 寄存器内的值。

在只接收模式下，在连续传输的情况下，可以在接收到最后一个数据前将 SPI_CON1.NXTCRC 位置 1，在收到最后一个数据后会收到 CRC，透过软件读取 SPI_RXCRC 与接收的 CRC 数据做校验。在间断传输的情况下，则在接收到倒数第二个数据后，必须对 SPI_CON1.NXTCRC 位执行写操作，在收到最后一个数据后会收到 CRC，透过软件读取 SPI_RXCRC 与接收的 CRC 数据做校验。

如果发送 FIFO 缓存中存在数据，则只有在发送数据字节后才会发送 CRC 值。在 CRC 发送期间，CRC 计算器处于关闭状态且寄存器值保持不变。

可通过以下步骤使用 CRC 进行 SPI 通信：

1. 对 SPI_CON1.BAUD、SPI_CON1.CPOL、SPI_CON1.CPHA、SPI_CON1.LSBFST、SPI_CON1.SSEN、SPI_CON1.SSOUT 和 SPI_CON1.MSTREN 值进行设置。
2. 向 SPI_CRCPOLY 寄存器中写入计算 CRC 的多项式。
3. 通过将 SPI_CON1.CRCEN 位置 1 来使能 CRC 计算。此操作还会将 SPI_RXCRC 和 SPI_TXCRC 寄存器清零。

4. 通过将 **SPI_CON1.SPIEN** 位置 1 使能 SPI。
5. 启动并保持通信，直到只剩下一个字节或半字未发送或接收。
 - 在全双工或只发送模式下，在连续传输的情况下，可以在最后一笔数据写入前任意时间点将 **SPI_CON1.NXTCRC** 位置 1，以表示在发送完最后一个字节后将发送 CRC。在间断传输的情况下，必须在最后传输的数据写入 **SPI_DATA** 后，立即对 **SPI_CON1.NXTCRC** 位执行写操作，以表示在发送完最后一个字节后将发送 CRC。
 - 在只接收模式下，在连续传输的情况下，可以在接收到最后一个数据前将 **SPI_CON1.NXTCRC** 位置 1，以便使 SPI 准备好在接收完最后一个数据后进入 CRC 阶段。在间断传输的情况下，则在接收到倒数第二个数据后，必须对 **SPI_CON1.NXTCRC** 位执行写操作，以便使 SPI 准备好在接收完最后一个数据后进入 CRC 阶段。在 CRC 传输期间，CRC 计算将冻结。
6. 传输完最后一个字节或半字后，SPI 进入 CRC 传输阶段。在全双工模式或只接收模式下，透过软件将接收的 CRC 与 **SPI_RXCRC** 值进行比较。

当 SPI 处于从机模式时，注意只能在时钟稳定（时钟处于空闲电平）时使能 CRC 计算。否则，可能导致 CRC 计算错误。

在 SPI 通信时钟频率较高的情况下，发送 CRC 时务必小心。应于在 CRC 传输阶段 CPU 尽可能保持空闲，因此禁止在 CRC 发送阶段调用函数，以便避免最后的数据和 CRC 接收出错。实际上在发送或接收最后的数据之前必须对 **SPI_CON1.NXTCRC** 位执行写操作。

SPI 通信时钟频率较高时，建议使用 DMA 模式来避免由于 CPU 访问影响 SPI 带宽而导致 SPI 速度性能下降。

如果将 SPI 配置为从机，并且使用 NSS 硬件模式，则需要在数据阶段和 CRC 阶段之间将 NSS 引脚保持为低电平。

在对从机片选的切换期间内，应在主机和从机两端同时将 CRC 值清零，以重新同步主机和机双方的 CRC 计算。

要将 CRC 值清零，请按以下步骤操作：

1. 将 **SPI_CON1.SPIEN** 位清零
2. 将 **SPI_CON1.CRCEN** 位清零
3. 将 **SPI_CON1.CRCEN** 位置 1
4. 将 **SPI_CON1.SPIEN** 位置 1

24.5 I2S 结构图

I2S 的框图如下所示：

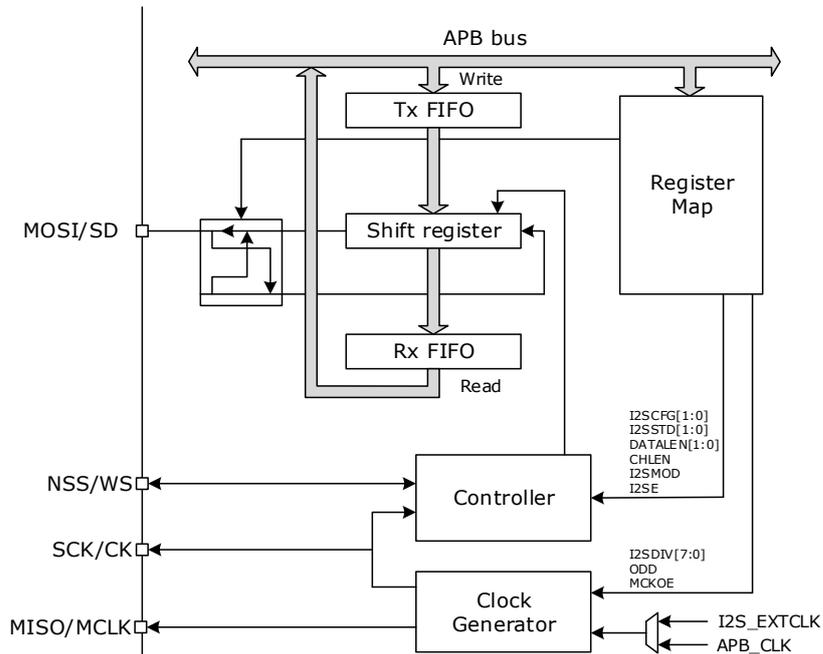


图 24-14 I2S 框图

当 I2S 功能使能时（通过设置 **SPI_I2SCFG.I2SMOD** 位），SPI 可用作音频 I2S 接口。该接口主要使用与 SPI 相同的引脚，标志和中断。

I2S 与 SPI 共享四个公共引脚：

SD: 串行数据（映射在 MOSI 引脚上），用于发送或接收两个时间复用数据通道。

WS: 声道选择（映射在 NSS 引脚上）是在主机模式下输出的数据控制信号，在从机模式下输入的数据控制信号。

CK: 串行时钟（映射在 SCK 引脚上）是主机模式下的串行时钟输出和从机模式下的串行时钟输入。

MCLK: 当 I2S 配置为主机模式时（以及当 **SPI_I2SPR.MCKOE** 位置 1 时），使用主时钟（映射在 MISO 引脚上），输出以预先配置的频率等于 $256 \times F_S$ 生成的附加时钟，其中 F_S 是音频采样频率。

当 I2S 设置为主机模式时，它使用自己的时钟发生器产生通信时钟。该时钟发生器也是主时钟输出的源。在 I2S 模式下有两个额外的寄存器。是钟发生器配置 **SPI_I2SPR**，另一个是通用 I2S 配置寄存器 **SPI_I2SCFG**（音频标准，从/主机模式，数据帧，通道帧，时钟极性）。

在 I2S 模式下不使用 **SPI_CON1** 寄存器和所有 CRC 寄存器。同样，不使用 **SPI_CON2** 寄存器中的 NSSOE 位和 **SPI_STAT** 中的 MODF 位。

I2S 在 16/32 位宽模式下使用相同的 SPI 寄存器进行数据传输（**SPI_DATA**）。

24.6 I2S 功能描述

24.6.1 音频协议

三线总线通常在处理两个通道上进行时间复用的音频数据：右声道和左声道。但是，只有一个 16 位寄存器用于发送或接收。因此，由软件向数据寄存器写入与每个通道侧相对应的适当值，或者从数据寄存器读取数据。始终先发送左声道，然后右声道（CHSIDE 对 PCM 协议没有意义）。

有四个数据和通道帧可供使用。数据可以采用以下格式发送：

- ◆ 16 位数据帧装在一个 16 位通道帧
- ◆ 16 位数据帧装在一个 32 位通道帧
- ◆ 24 位数据帧装在一个 32 位通道帧
- ◆ 32 位数据帧装在一个 32 位通道帧

当使用 16 位数据帧装在一个 32 位通道帧上时，前 16 位（MSB）是有效位，16 位 LSB 被强制为 0，无需任何软件操作或 DMA 请求（只有一个读/写操作）。

如果 DMA 是应用程序的首选，则 24 位和 32 位数据帧需要对 SPI_DATA 寄存器进行两次 CPU 读或写操作，或者需要两次 DMA 操作。具体而言，对于 24 位数据帧，8 个非有效位通过 0 位扩展为 32 位（通过硬件）。

对于所有数据格式和通信标准，始终首先发送最高有效位（MSB 优先）。

I2S 接口支持四种音频标准，可使用 SPI_I2SCFG.I2SSTD [1:0]进行配置。

24.6.1.1 I2S 飞利浦标准

对于该标准，WS 信号用于指示正在传输哪个信道。它在第一位 (MSB) 可用之前的一个 CK 时钟周期被激活。

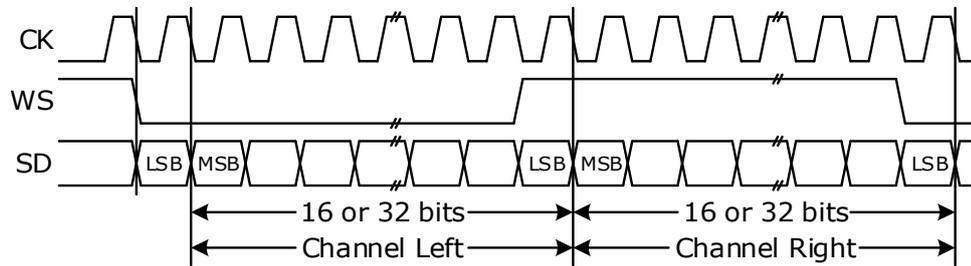


图 24-15 I2S 飞利浦标准波形 (16/32 位数据帧, CPOL = 0)

数据在 CK 的下降沿 (对于发送器) 被锁存, 并在上升沿 (对于接收器) 被读取。WS 信号也在 CK 的下降沿锁存。

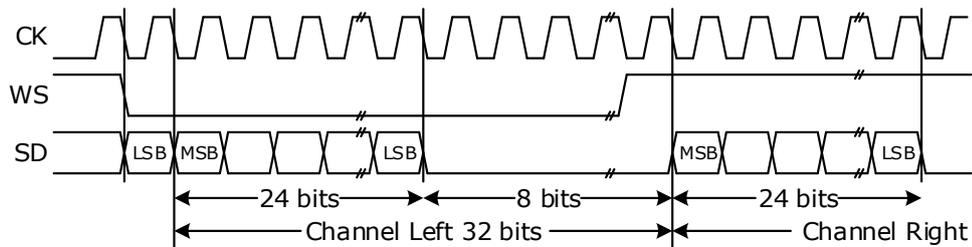


图 24-16 I2S 飞利浦标准波形 (24 位数据帧, CPOL = 0)

该模式需要对 SPI_DATA 寄存器进行两次写或读操作。

◆ 在传输模式下:

如果必须发送 0x123456 (24 位):

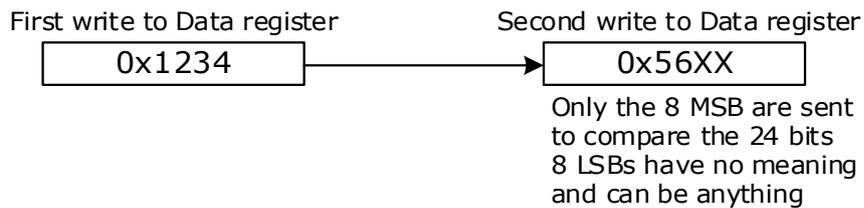


图 24-17 发送 0x123456

◆ 在接收模式下:

如果收到数据 0x123456:

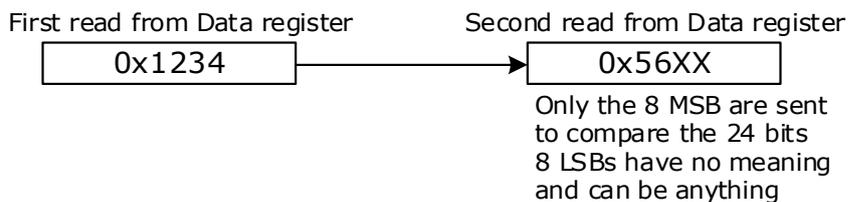


图 24-18 接收 0x123456

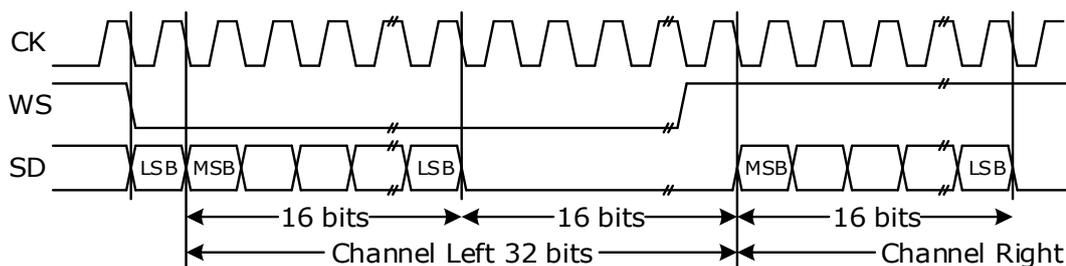


图 24-19 I2S 飞利浦标准波形 (16 位数据帧扩展到 32 位通道帧, CPOL = 0)

当在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧时, 只需要访问一次 **SPI_DATA** 寄存器。其余 16 位由硬件强制为 0x0000, 以将数据扩展为 32 位格式。

如果要传输的数据或接收的数据是 0x4567 (0x4567 0000 扩展到 32 位), 则需要执行下图所示的操作。

Only one access to SPIx_DR

0x4567

图 24-20 16 位数据帧扩展到 32 位通道帧的示例

24.6.1.2 MSB 对齐标准

对于该标准，WS 信号与第一个数据位（MSBit）同时生成。

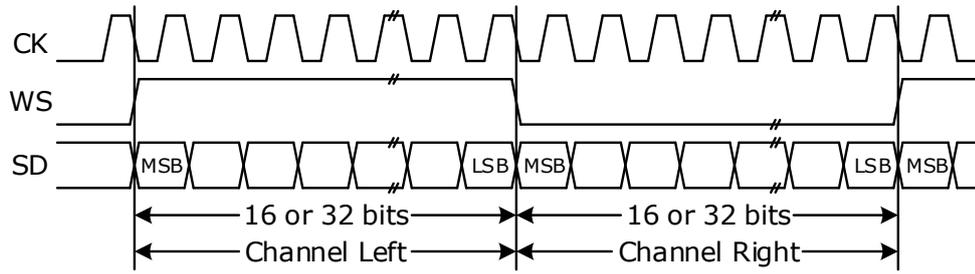


图 24-21 MSB 对齐标准波形（16/32 位数据帧，CPOL = 0）

数据在 CK 的下降沿（发送器）被锁存，并在上升沿（对于接收器）读取。

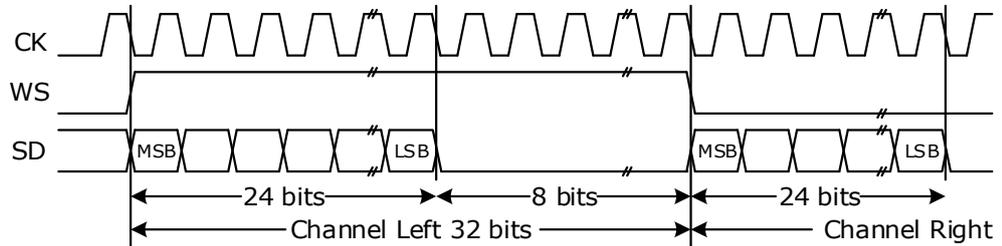


图 24-22 MSB 对齐标准波形（24 位数据帧，CPOL = 0）

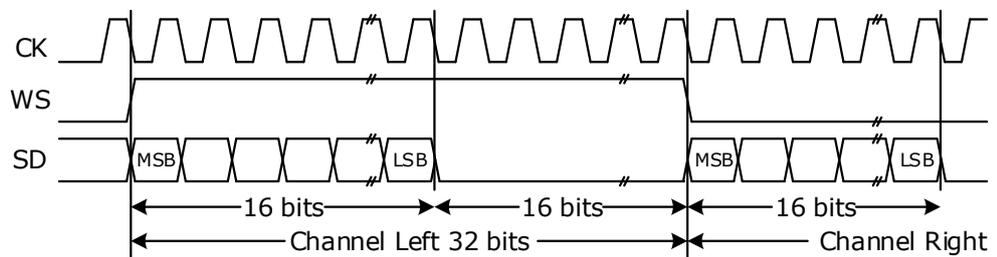


图 24-23 MSB 对齐标准波形（16 位数据帧扩展到 32 位通道帧，CPOL = 0）

24.6.1.3 LSB 对齐标准

该标准类似于 MSB 对齐标准（16 位和 32 位通道帧格式没有区别）。

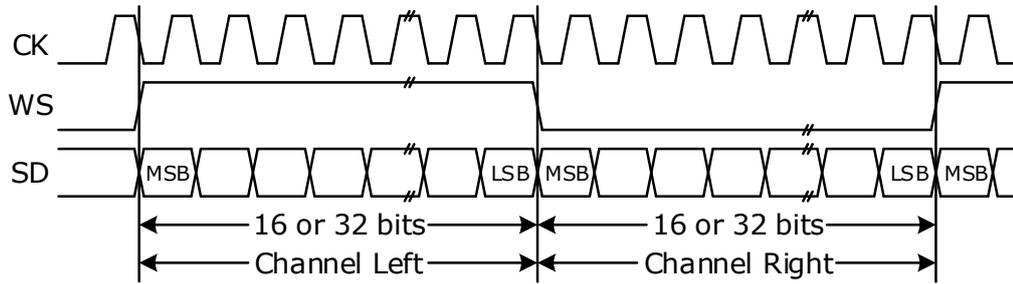


图 24-24 LSB 对齐标准波形（16/32 位数据帧，CPOL = 0）

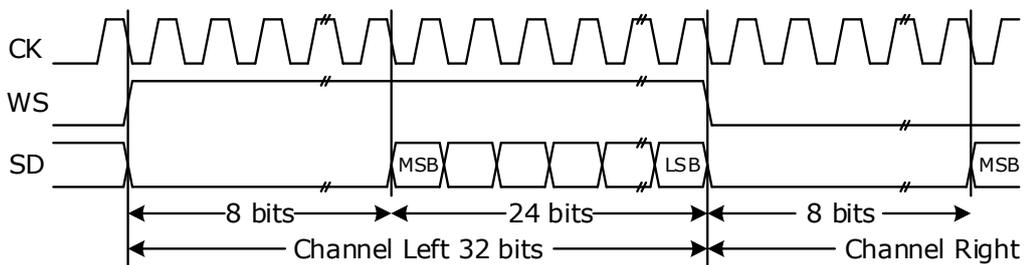


图 24-25 LSB 对齐标准波形（24 位数据帧，CPOL = 0）

◆ 在传输模式下：

如果必须发送数据 0x123456，则软件或 DMA 需要对 **SPI_DATA** 寄存器进行两次写操作。操作如下所示。

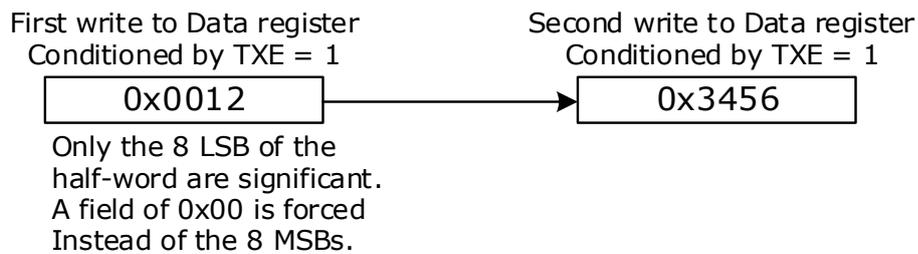


图 24-26 传输 0x123456 所需的操作

◆ 在接收模式下:

如果接收到数据 0x123456, 则每个 RXE 事件都需要从 **SPI_DATA** 寄存器执行两次连续的读操作。

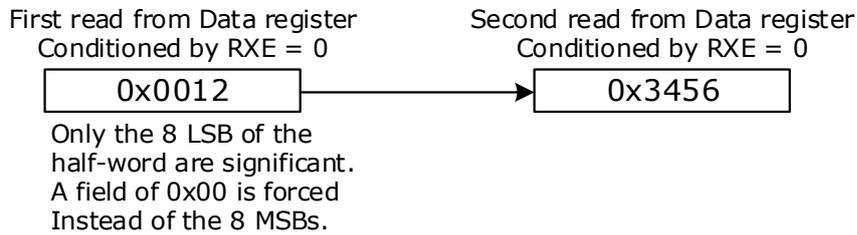


图 24-27 接收 0x123456 所需的操作

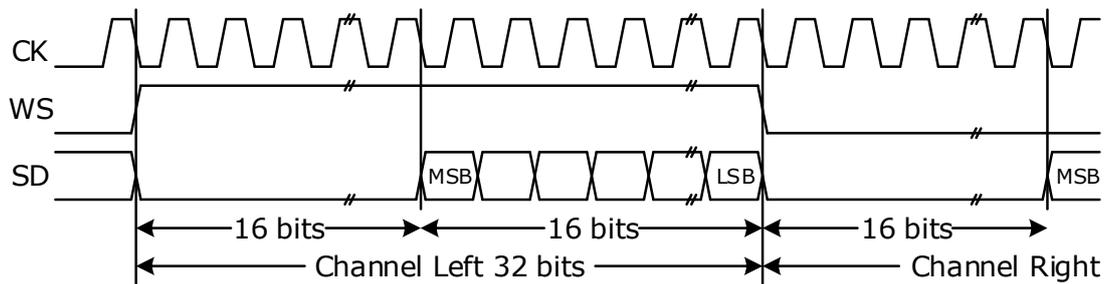


图 24-28 LSB 对齐标准波形 (16 位数据帧扩展到 32 位通道帧, CPOL = 0)

在 I2S 配置阶段选择扩展到 32 位通道帧的 16 位数据帧时, 只需要访问一次 **SPI_DATA** 寄存器。其余 16 位由硬件强制为 0x0000, 以将数据扩展为 32 位格式。在这种情况下, 它对应于半字 MSB。

如果要发送的数据或接收的数据是 0x4567 (0x0000 4567 扩展到 32 位), 则需要执行下图所示的操作。

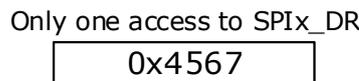


图 24-29 16 位数据帧扩展到 32 位通道帧的示例

在传输模式下, 当发生 TXE 事件时, 应用程序必须写入要传输的数据(在本例中为 0x4567)。首先发送 0x0000 字段 (32 位扩展)。

在接收模式下, 一旦接收到有效的半字 (而不是 0x0000 字段), RXE 就会被清零。

以这种方式, 在两次写入或读取操作之间提供更多时间以防止下溢或溢出条件。

24.6.1.4 PCM 标准

对于 PCM 标准，不需要使用信道侧讯息。两种 PCM 模式（短帧和长帧）可用，并可使用 **SPI_I2SCFG.PCMSYNC** 位进行配置。

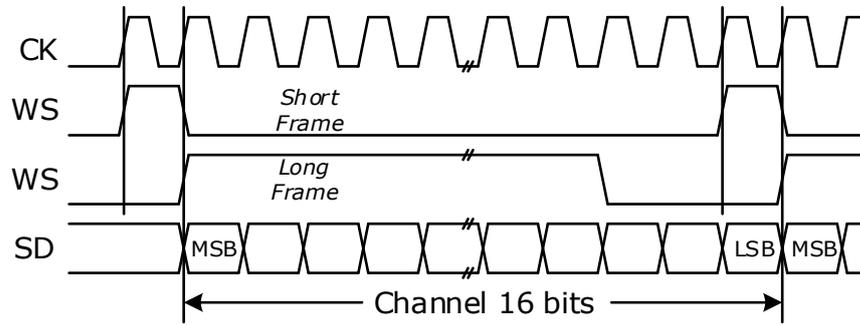


图 24-30 PCM 标准波形（16 位）

对于长帧同步，WS 信号置入时间在主模式下固定为 13 位。

对于短帧同步，WS 同步信号只有一个周期。

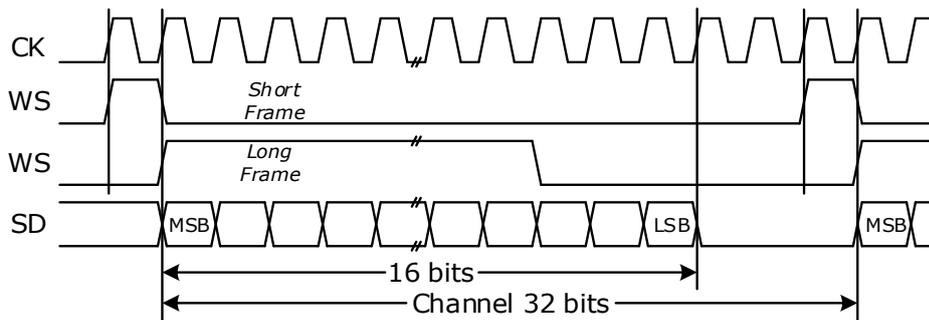


图 24-31 PCM 标准波形（16 位数据帧扩展到 32 位通道帧）

注意：对于两种模式（主机和从机）以及两种同步（短帧和长帧），需要指定两个连续数据（以及两个同步信号）之间的位数（**SPI_I2SCFG.DATLEN** 和 **SPI_I2SCFG.CHLEN** 位）即使在从机模式。

24.6.2 时钟产生器

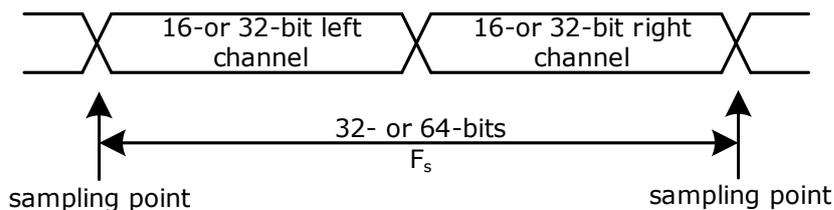
I2S 比特率决定 I2S 数据线上的数据流和 I2S 时钟信号频率。

I2S 比特率=每通道的比特数×通道数×采样音频

对于 16 位音频，左右声道，I2S 比特率计算如下：

I2S 比特率= 16×2×FS

它将是：如果数据包长度为 32 位宽，则 I2S 比特率= 32 x 2 x FS。



F_s : audio sampling frequency

图 24-32 音频采样频率定义

音频采样频率可以是 192 kHz, 96 kHz, 48 kHz, 44.1 kHz, 32 kHz, 22.05 kHz, 16 kHz, 11.025 kHz 或 8 kHz（或该范围内的任何其他值）。为了达到所需频率，需要根据以下公式对线性分频器进行编程：

在 I2S, MSB, LSB 模式中，CH 为 2。

在 PCM 模式下，CH 为 1。

当产生主时钟时（SPI_I2SPR 寄存器中的 MCKOE 位置 1）：

当通道帧为 16 位宽时， $F_s = I2S_CLK / [(16 * 2) * ((CH * I2SDIV) + ODD) * 8]$

当通道帧为 32 位宽时， $F_s = I2S_CLK / [(32 * 2) * ((CH * I2SDIV) + ODD) * 4]$

禁用主时钟时（SPI_I2SPR 寄存器中的 MCKOE 位清零）：

当通道帧为 16 位宽时， $F_s = I2S_CLK / [(16 * 2) * ((CH * I2SDIV) + ODD)]$

当通道帧为 32 位宽时， $F_s = I2S_CLK / [(32 * 2) * ((CH * I2SDIV) + ODD)]$

下图提供了不同时钟配置的示例精度值。

注意：其他配置可能允许最佳时钟精度。

APB_CLK (MHz)	Data length	I2SDIV	I2SODD	MCLK	Target fs(Hz)	Real fs (Hz)	Error
32	16	5	0	No	96000	100000	4.1667%
32	32	2	0	No	96000	100000	4.1667%
32	16	10	1	No	48000	47619	0.7937%
32	32	5	0	No	48000	50000	4.1667%
32	16	11	1	No	44100	43478	1.4098%
32	32	5	1	No	44100	45454	3.0715%
32	16	15	1	No	32000	32258	0.8065%
32	32	8	0	No	32000	31250	2.3430%
32	16	22	1	No	22050	22222	0.7811%
32	32	11	1	No	22050	21739	1.4098%
32	16	31	1	No	16000	15873	0.7937%
32	32	15	1	No	16000	16129	0.8065%
32	16	45	1	No	11025	10989	0.3264%
32	32	22	1	No	11025	11111	0.7811%
32	16	62	1	No	8000	8000	0.0000%
32	32	31	1	No	8000	7936	0.7937%
32	16	2	0	Yes	32000	31250	2.3430%
32	32	2	0	Yes	32000	31250	2.3430%
32	16	3	0	Yes	22050	20833	5.5170%
32	32	3	0	Yes	22050	20833	5.5170%
32	16	4	0	Yes	16000	15625	2.3428%
32	32	4	0	Yes	16000	15625	2.3428%
32	16	5	1	Yes	11025	11363	3.0715%
32	32	5	1	Yes	11025	11363	3.0715%
32	16	8	0	Yes	8000	7812	2.3428%
32	32	8	0	Yes	8000	7812	2.3428%

24.6.3 I2S 主机模式

I2S 可以配置为主机模式。这意味着在 CK 引脚上生成串行时钟以及字选择信号 WS。主时钟 (MCK) 可以输出与否, 由 **SPI_I2SPR.MCKOE** 位控制。

24.6.3.1 设置流程

1. 选择 **SPI_I2SPR.I2SDIV [7: 0]**位, 以定义串行时钟波特率, 以达到正确的音频采样频率。还必须定义 **SPI_I2SPR.ODD** 位。
2. 选择 **CKPOL** 位以定义通信时钟的稳定电平。如果需要将主时钟 MCK 提供给外部 DAC /ADC 音频组件, 则将 **SPI_I2SPR.MCKOE** 位置 1 (I2SDIV 和 ODD 值应根据 MCK 输出的状态进行计算, 更多详细讯息, 请参见第**錯誤! 找不到参照来源**。节: 时钟产生器)。
3. 将 **SPI_I2SCFG.I2SMOD** 位置 1 以激活 I2S 功能, 并通过 **SPI_I2SCFG.I2SSTD [1: 0]**和 **SPI_I2SCFG.PCMSYNC** 位选择 I2S 标准, 通过 **SPI_I2SCFG.DATLEN [1: 0]**位选择数据长度, 通过配置 **SPI_I2SCFG.CHLEN** 位来择数通道长度。通过 **SPI_I2SCFG.I2SCFG [1: 0]**位选择 I2S 主机模式和方向 (发送器或接收器)。
4. 如果需要, 通过写 **SPI_CON2** 寄存器选择 DMA 功能。
5. 如果需要, 通过写 **SPI_IER** 寄存器选择所有可能的中断源。
6. 必须设置 **SPI_I2SCFG.I2SE** 位。WS 和 CK 配置为输出模式。如果 **SPI_I2SPR.MCKOE** 位置 1, MCK 也是输出。

24.6.3.2 传输序列

当半字写入 Tx FIFO 时，传输序列开始。

写入 Tx FIFO 的第一个数据应对应于左声道数据。当左声道的数据写入 Tx FIFO 后必须紧跟着写入对应于右声道数据。CHSIDE 标志指示当前发送的声道。

必须将全帧视为左右声道数据传输。不会有仅发送左声道的部分帧。

有关写入操作的更多详细讯息，具体取决于所选的 I2S 标准模式，请参见第**錯誤! 找不到参照来源**。节：音频协议)。

为了确保连续的音频数据传输，必须在当前传输结束之前将对 SPI_DATA 寄存器写入下一个要传输的数据。若要禁用 I2S，必须等待 TXE = 1 且 BUSY = 0 时，通过清除 SPI_I2SCFG.I2SE 来关闭 I2S。

24.6.3.3 接收序列

操作模式与传输模式相同，除了第 3 点（参见第**錯誤! 找不到参照来源**。节：I2S 主机模式中描述的过程）之外，其中配置应通过 SPI_I2SCFG.I2SCFG [1:0]位设置主机接收模式。

Rx FIFO 满时，RXF 标志置 1，如果 SPI_IER.RXFIE 位置 1，则产生中断。根据数据和信道长度配置，右声道或左声道接收的音频值可能是由于进入 Rx FIFO 的一次或两次接收所致。

通过读 SPI_DATA 寄存器来执行清零 RXF 位。

有关根据所选 I2S 标准模式进行读取操作的更多详细讯息，请参见第**錯誤! 找不到参照来源**。节：音频协议。

如果在尚未读取先前接收的数据的同时接收到新的数据，则生成溢出并设置 RXOV 标志。如果 SPI_IER 寄存器中的 RXOVIE 位置 1，则会产生中断以指示错误。

24.6.4 I2S 从机模式

对于从机配置，I2S 可配置为发送或接收模式。

操作模式主要遵循与 I2S 主配置相同的规则。在从机模式下，I2S 接口不会生成时钟。时钟和 WS 信号从连接到 I2S 接口的外部主设备输入。因此，用户无需配置时钟。

下面列出了配置步骤：

1. 将 SPI_I2SCFG.I2SMOD 位置 1 选择 I2S 模式，并通过 SPI_I2SCFG.I2SSTD [1: 0] 位选择 I2S 标准，通过 SPI_I2SCFG.DATLEN [1: 0]位选择数据长度，并通过 SPI_I2SCFG.CHLEN 位选择通道长度。通过 SPI_I2SCFG.I2SCFG [1: 0]位选择从机的模式（发送或接收）。
2. 如果需要，通过写 SPI_CON2 寄存器选择 DMA 功能。
3. 如果需要，通过写 SPI_IER 寄存器选择所有可能的中断源
4. 必须设置 SPI_I2SCFG.I2SE 位。

24.6.4.1 传输序列

当外部主机发送时钟并且 WS 信号请求数据传输时，传输序列开始。当从机启动后，会等待用户将要传输的数据写入传送 FIFO。

对于 I2S，MSB 对齐和 LSB 对齐模式，要写入数据寄存器的第一个数据项对应于左通道的数据。通信开始时，数据从 Tx FIFO 传输到移位寄存器。

CHSIDE 标志指示当前发送的声道。与主机传输模式相比，在从机模式下，CHSIDE 对应来自外部主机设备的 WS 信号。

在第一位传输期间，数据半字并行加载到 16 位移位寄存器（来自内部总线），然后首先串行移出到 MOSI / SD 引脚 MSB。

注意，在尝试写入 Tx FIFO 之前，应检查 TXF 标志为 0。

有关写入操作的更多详细讯息，具体取决于所选的 I2S 标准模式，请参见第**錯誤! 找不到參照來源**。节：音频协议。

为了确保连续的音频数据传输，必须在当前传输结束之前将 SPI_DATA 寄存器写入下一个要传输的数据。如果在下一次数据通信的第一个时钟沿之前没有将数据写入 SPI_DATA 寄存器，则会产生下溢标志并产生中断。这向软件表明传输的数据是错误的。当发生下溢中断时，若要重新开始传输，此时需要先将 I2S 禁用后，重新启动，使其重新侦测并对齐左声道输出数据。

若要禁用 I2S，必须等待 TXE = 1 时，通过清除 I2SE 来关闭 I2S。

24.6.4.2 接收序列

操作模式与传输模式相同，除了第 1 点（参见第**錯誤! 找不到參照來源**。节：I2S 从机模式中描述的步骤）之外，其中配置应使用 SPI_I2SCFG.I2SCFG [1:0]位设置从机接收模式，并且接收的数据会从左声道开始接收。

无论数据长度或通道长度为何，音频数据都由 16 比特分组接收。根据数据长度和信道长度配置，右声道或左声道接收的音频值可能是由于接收到 Rx FIFO 的一次或两次接收所致。

有关根据所选 I2S 标准模式进行读取操作的更多详细讯息，请参见第**錯誤! 找不到參照來源**。节：音频协议。

如果在尚未读取先前接收的数据的同时接收到新的数据，则生成溢出并设置 RXOV 标志。如果 SPI_IER.RXOVIE 位置 1，则会产生中断以指示错误。

注意：外部主机应具有通过音频通道以 16 位或 32 位通道发送/接收数据帧的能力。

24.6.5 I2S 状态标志

为应用程序提供了六个状态标志，以完全监视 SPI 总线的状态。

24.6.5.1 发送 FIFO 空标志 (TXE)

设置时，TXE 标志指示发送 FIFO 为空，并且下一个要传输的数据可以加载到 FIFO 中。通过写 SPI_DATA 寄存器清零 TXE 标志。

24.6.5.2 发送 FIFO 满标志 (TXF)

设置时，TXF 标志指示发送 FIFO 已满，并且下一个要传输的数据无法加载到 FIFO 中。TXF 标志由要发送的下一个数据清除。

24.6.5.3 接收 FIFO 空标志 (RXE)

设置时，RXE 标志指示接收 FIFO 为空，并且下一个要传输的数据可以加载到 FIFO 中。RXE 标志由下一个数据接收到 FIFO 中清除。

24.6.5.4 接收 FIFO 满标志 (RXF)

设置后，RXF 标志指示接收 FIFO 已满，并且下一个要接收的数据无法加载到 FIFO 中。通过对 SPI_DATA 寄存器的读访问来清除 RXF 标志。

24.6.5.5 BUSY 标志 (BUSY)

BUSY 标志由硬件置位和清除（写入此标志无效）。当 BUSY 标志置 1 时，表示 I2S 上正在进行数据传输（I2S 总线忙）。BUSY 标志可以用于检测传输的结束，从而防止最后一次传输损坏。

在以下任一条件下清除 BUSY 标志：

- ◆ 当 I2S 正确地禁用
- ◆ 传输完成时
- ◆ 在主发送模式下，BUSY 标志在所有传输期间保持高电平
- ◆ 在从机模式下，BUSY 标志在每次传输之间变为低电平，持续一个 I2S 时钟周期

注意：建议始终使用 TXTH 和 RXTH 标志（而不是 BUSY 标志）来处理数据传输或接收操作。

24.6.5.6 声道标志 (CHSIDE)

此标志是提供用户判断此时总线上正在传输的声道为左声道还右声道使用，只有在 BUSY 标志被设置时有效。该标志在 PCM 标准中没有意义（短帧和长帧模式）。

24.6.6 I2S 错误标志

I2S 单元有五个错误标志。

24.6.6.1 发送 FIFO 溢出标志 (TXOV)

发生发送溢出条件的情况是发送 FIFO 已满，但用户正在执行写操作。在这种情况下，新写入的数据不会加载到发送 FIFO 中。通过对 **SPI_ICR** 寄存器的写访问来清除 TXOV 位，或者在新传输开始时它将自动清零。

24.6.6.2 发送 FIFO 下溢标志 (TXUD)

发生发送下溢运行条件是当发送 FIFO 为空时，但主机想要从发送 FIFO 拿取数据。在这种情况下，发送 FIFO 的内容不会被发送至总线上。通过对 **SPI_ICR** 寄存器的写访问来清除 TXUD 位，或者在新的传输开始时它将自动清零。

24.6.6.3 接收 FIFO 溢出标志 (RXOV)

当主机或从机完成下一个数据帧的接收而接收 FIFO 的前 16 帧的读操作尚未完成时（设置了 RXF 标志的情况），发生接收溢出条件。

在这种情况下，接收 FIFO 的内容不会将接收的新数据存入。**SPI_DATA** 寄存器的读操作返回先前接收的帧。所有其他在 RXF 标志设置的情况下，接收的新数据都将丢失。

通过对 **SPI_ICR** 寄存器的写访问来清除 RXOV 位，或者在新的传输开始时它将自动清零。

24.6.6.4 接收 FIFO 下溢标志 (RXUD)

当接收 FIFO 为空时，但用户正在执行读操作，会发生接收下溢条件。在这种情况下，读取操作不会从接收 FIFO 中读到有效的数据。

通过对 **SPI_ICR** 寄存器的写访问来清除 RXUD 位，或者在新的传输开始时它将自动清零。

24.6.6.5 帧格式错误 (FRE)

仅当 I2S 配置为从模式时，才能通过硬件设置该标志。如果外部主设备正在更改 WS 线路，而从设备不期望此更改，则设置它。如果同步丢失，则需要执行以下步骤以从此状态恢复并使外部主设备与 I2S 从设备重新同步：

1. 禁用 I2S。
2. 当在 WS 线上检测到正确的电平时再次启用它 (WS 线在 I2S 模式下为高电平或在 MSB 或 LSB 校验或 PCM 模式下为低电平)。

主设备和从设备之间的去同步可能是由于 SCK 通信时钟或 WS 帧同步线上的噪声环境造成的。如果 FREIE 位置 1，则可以产生帧格式错误中断。通过对 **SPI_ICR** 寄存器的写访问来清除 FRERI 位。

24.7 特殊功能寄存器

24.7.1 寄存器列表

外设寄存器可支持半字（16 位）或字（32 位）访问。

SPI 寄存器列表			
名称	偏移地址	类型	描述
SPI_CON1	0000 _H	R/W	SPI 控制寄存器 1
SPI_CON2	0004 _H	R/W	SPI 控制寄存器 2
SPI_STAT	0008 _H	R	SPI 状态寄存器
SPI_DATA	000C _H	R/W	SPI 数据寄存器
SPI_CRCPOLY	0010 _H	R/W	SPI CRC 多项式寄存器
SPI_RXCRC	0014 _H	R	SPI RX CRC 寄存器
SPI_TXCRC	0018 _H	R	SPI TX CRC 寄存器
SPI_I2SCFG	001C _H	R/W	SPI I2S 配置寄存器
SPI_I2SPR	0020 _H	R/W	SPI I2S 预分频寄存器
SPI_IER	0024 _H	W1	SPI 中断启用寄存器
SPI_IDR	0028 _H	W1	SPI 中断禁用寄存器
SPI_IVS	002C _H	R	SPI 中断有效状态寄存器
SPI_RIF	0030 _H	R	SPI 原始中断标志状态寄存器
SPI_IFM	0034 _H	R	SPI 中断标志屏蔽状态寄存器
SPI_ICR	0038 _H	C_W1	SPI 中断清除寄存器

24.7.2 寄存器描述

24.7.2.1 SPI 控制寄存器 1 (SPI_CON1)

SPI 控制寄存器 1 (SPI_CON1)																															
偏移地址: 00H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																BIDEN	BIDOEN	CRCEN	NXTCRC	FLEN	RXO	SSEN	SSOUT	LSBFST	SPIEN	BAUD<2:0>			MSTREN	CPOL	CPHA

—	Bits 31-16	—	—
BIDEN	Bit 15	R/W	<p>启用双向数据模式</p> <p>该位使用通用单双向数据线实现半双工通信。双向模式激活时，保持 RXO 位清零。</p> <p>0: 选择双线单向通信数据模式</p> <p>1: 选择单线双向通信数据模式</p> <p>注意：该位不用于 I2S 模式。</p>
BIDOEN	Bit 14	R/W	<p>双向模式下的输出使能</p> <p>该位与 BIDEN 位组合在一起选择双向模式下的传输方向</p> <p>0: 禁用输出（仅接收模式）</p> <p>1: 启用输出（仅发送模式）</p> <p>注意：在主机模式下，使用 MOSI 引脚，在从机模式，使用 MISO 引脚。在 I2S 模式下不使用该位。</p>
CRCEN	Bit 13	R/W	<p>硬件 CRC 计算使能</p> <p>0: 禁用 CRC 计算</p> <p>1: 启用 CRC 计算</p> <p>注意：为确保正确操作，只应在禁止 SPI (SPIEN = "0") 时对此位执行写操作。I2S 模式中不使用该位。</p>
NXTCRC	Bit 12	R/S_W1	<p>接下来传输 CRC</p> <p>0: 数据传输结束时不传输 CRC</p> <p>1: 数据传输结束时传输 CRC</p> <p>注意：若当前传输需要传送 CRC 时，在写入数据后即可将该位置 1。若为接收时，则在写入无效数据后将该位置 1。I2S 模式中不使用该位。</p>
FLEN	Bit 11	R/W	<p>数据帧长度</p> <p>0: 选择 8 位数据帧格式进行发送/接收</p> <p>1: 选择 16 位数据帧格式进行发送/接收</p> <p>注意：为确保正确操作，只应在禁止 SPI (SPIEN</p>

			=“0”) 时对此位执行写操作。I2S 模式中不使用该位。
RXO	Bit 10	R/W	<p>启用仅接收模式</p> <p>该位允许使用单个单向线进行单工通信, 以专门接收数据。当仅接收模式处于活动状态时, 保持 BIDEN 位清零。该位在多区域系统中也很有用, 在该系统中不访问此特定从机, 来自被访问从机的输出不会被破坏。</p> <p>0: 全双工 (发送和接收)</p> <p>1: 禁用输出 (仅接收模式)</p> <p>注意: 该位不用于 I2S 模式。</p>
SSEN	Bit 9	R/W	<p>软件从机选择管理</p> <p>当 SSEN 位置 1 时, NSS 引脚输入将被 SSOUT 位的值替换。</p> <p>0: 禁用软件从机选择管理</p> <p>1: 启用软件从机选择管理</p> <p>注: 该位不用于 I2S 模式和 SPI TI 模式。</p>
SSOUT	Bit 8	R/W	<p>内部从机选择</p> <p>0: NSS 引脚输入为 0。</p> <p>1: NSS 引脚输入为 1。</p> <p>仅当 SSEN 位置 1 时, 该位才有效。此位的值将作用到 NSS 引脚上, 并忽略 NSS 引脚的 IO 值。</p> <p>注: 该位不用于 I2S 模式和 SPI TI 模式。</p>
LSBFST	Bit 7	R/W	<p>先发送 LSB</p> <p>0: 首先使用 MSB 发送/接收数据</p> <p>1: 首先使用 LSB 发送/接收数据</p> <p>注意: 1.通信正在进行时, 不应更改此位。 2.该位不用于 I2S 模式和 SPI TI 模式。</p>
SPIEN	Bit 6	R/W	<p>SPI 启用</p> <p>0: 禁用 SPI 外设</p> <p>1: 启用 SPI 外设</p> <p>注意: 该位不用于 I2S 模式。</p>
BAUD	Bits 5-3	R/W	<p>波特率控制</p> <p>000: $F_{PCLK} / 2$</p> <p>001: $F_{PCLK} / 4$</p> <p>010: $F_{PCLK} / 8$</p> <p>011: $F_{PCLK} / 16$</p> <p>100: $F_{PCLK} / 32$</p> <p>101: $F_{PCLK} / 64$</p> <p>110: $F_{PCLK} / 128$</p> <p>111: $F_{PCLK} / 256$</p> <p>注意: 通信正在进行时, 不应更改这些位。在 I2S 模式下不使用该位。</p>

MSTREN	Bit 2	R/W	<p>主机选择</p> <p>0: 从机配置 1: 主机配置</p> <p>注意: 通信正在进行时, 不应更改此位。在 I2S 模式下不使用该位。</p>
CPOL	Bit 1	R/W	<p>时钟极性</p> <p>0: 在空闲的状态下, SCK 引脚保持低电平输出 1: 在空闲的状态下, SCK 引脚保持高电平输出</p> <p>注意: 通信正在进行时, 不应更改此位。该位不用于 I2S 模式和 SPI TI 模式。</p>
CPHA	Bit 0	R/W	<p>时钟相位</p> <p>0: 从第一个时钟边沿开始采样数据 1: 从第二个时钟边沿开始采样数据</p> <p>注意: 通信正在进行时, 不应更改此位。该位不用于 I2S 模式和 SPI TI 模式。</p>

			<p>1: 产生 NSS 脉冲</p> <p>注: 1. 只有在禁止 SPI (SPIEN = 0) 时才能写入该位。</p> <p>2. 该位不用于 I2S 模式和 SPI TI 模式。</p>
NSSOE	Bit 2	R/W	<p>NSS 引脚输出使能</p> <p>0: 在主模式下禁止 NSS 输出, 可在多主模式配置下工作</p> <p>1: 在主模式下使能 NSS 输出, 不能在多主模式环境下工作</p> <p>注: 该位不用于 I2S 模式和 SPI TI 模式。</p>
TXDMA	Bit 1	R/W	<p>Tx DMA 使能</p> <p>该位置 1 时, 只要 TXTH 标志置 1, 就会产生 DMA 请求。</p> <p>0: 禁用 Tx DMA</p> <p>1: 启用 Tx DMA</p>
RxDMA	Bit 0	R/W	<p>Rx DMA 使能</p> <p>该位置 1 时, 只要 RXTH 标志置 1, 就会产生 DMA 请求。</p> <p>0: 禁用 Rx DMA</p> <p>1: 启用 Rx DMA</p>

24.7.2.3 SPI 状态寄存器 (SPI_STAT)

SPI 状态寄存器(SPI_STAT)																															
偏移地址: 08H																															
复位值: 00000000_00000000_00000001_00010001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			RXFLV<4:0>								TXFLV<4:0>				BUSY	CHSIDE		RXTH	RXUD	RXOV	RXF	RXE					TXTH	TXUD	TXOV	TXF	TXE

—	Bits 31-29	—	—
RXFLV	Bits 28-24	R	Rx FIFO 的水平 该位提供 Rx FIFO 的水平值。
—	Bits 23-21	—	—
TXFLV	Bits 20-16	R	Tx FIFO 水平 该位提供 Tx FIFO 的水平值。
BUSY	Bit 15	R	忙标志 0: SPI (或 I2S) 不繁忙 1: SPI (或 I2S) 忙于通信 该标志由硬件设置和清除。
CHSIDE	Bit 14	R	声道侧 0: 当前正在发送或接收左声道 1: 当前正在传输或接收右声道 注: 该位不用于 SPI 模式。它在 PCM 模式下没有意义。
—	Bit 13	—	—
RXTH	Bit 12	R	接收 FIFO 水平超出阈值 该位提供接收FIFO水平超过阈值。 0: 接收FIFO水平没有超过阈值 1: 接收FIFO水平超过阈值
RXUD	Bit 11	R	接收 FIFO 下溢 0: 接收 FIFO 没有下溢 1: 接收FIFO下溢
RXOV	Bit 10	R	接收 FIFO 溢出 0: 接收 FIFO 没有溢出 1: 接收FIFO溢出
RXF	Bit 9	R	接收 FIFO 满 0: 接收 FIFO 没有满 1: 接收FIFO满
RXE	Bit 8	R	接收 FIFO 空 0: 接收 FIFO 没有空

			1: 接收FIFO空
—	Bits 7-5	—	—
TXTH	Bit 4	R	发送 FIFO 水平低于阈值 该位提供发送FIFO水平低于阈值。 0: 发送 FIFO 水平高于阈值 1: 发送FIFO水平低于阈值
TXUD	Bit 3	R	发送 FIFO 下溢 0: 发送 FIFO 没有下溢 1: 发送FIFO下溢
TXOV	Bit 2	R	发送 FIFO 溢出 0: 发送 FIFO 没有溢出 1: 发送FIFO溢出
TXF	Bit 1	R	发送 FIFO 满 0: 发送 FIFO 没有满 1: 发送FIFO满
TXE	Bit 0	R	发送 FIFO 空 0: 发送 FIFO 没有空 1: 发送FIFO空

24.7.2.4 SPI 数据寄存器 (SPI_DATA)

SPI 数据寄存器 (SPI_DATA)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DATA<15:0>															

—	Bits 31-16	—	—
DATA	Bits 15-0	R/W	<p>数据寄存器</p> <p>收到或将要传输的数据</p> <p>数据寄存器用作Rx和Tx FIFO之间的接口。读取数据寄存器时，访问Rx FIFO，而写入数据寄存器访问Tx FIFO。</p> <p>注意：数据始终是右对齐的。写入寄存器时忽略未使用的位，读取寄存器时读取为0。Rx阈值设置必须始终与当前使用的读取访问权限相对应。</p>

24.7.2.5 SPI CRC 多项式寄存器 (SPI_CRCPOLY)

SPI CRC 多项式寄存器 (SPI_CRCPOLY)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000111																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CRCPOLY<15:0>															

—	Bits 31-16	—	—
CRCPOLY	Bits 15-0	R/W	<p>CRC多项式寄存器</p> <p>该寄存器包含CRC计算的多项式。</p> <p>CRC多项式 (0007h) 是该寄存器的复位值。可以根据需要配置另一个多项式。</p> <p>注意：多项式值应仅为奇数。没有支持偶数。</p>

24.7.2.6 SPI RX CRC 寄存器 (SPI_RXCRC)

SPI RX CRC 寄存器 (SPI_RXCRC)																																															
偏移地址: 14 _H																																															
复位值: 00000000_00000000_00000000_00000000 _B																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
																RXCRC<15:0>																															

—	Bits 31-16	—	—
RXCRC	Bits 15-0	R	<p>接收 CRC 值</p> <p>使能 CRC 计算后, RXCRC[15:0]位将包含后续接收字节在计算后所得到的 CRC 值。当 SPI_CON1 寄存器中的 CRCEN 位写入 1 时, 此寄存器复位。CRC 通过 SPI_CRCPOLY 寄存器中编程的多项式连续计算。数据帧长度设置为 8 位数据 (SPI_CON1 的 FLEN 位清零) 时, 仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。选择 16 位数据帧长度 (SPI_CON1 寄存器的 FLEN 位置 1) 时, 考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。注意: 当 BUSY 标志置 1 时, 读取此寄存器可能返回一个不正确的值。</p>

24.7.2.7 SPI TX CRC 寄存器 (SPI_TXCRC)

SPI TX CRC 寄存器 (SPI_TXCRC)																															
偏移地址: 18H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TXCRC<15:0>															

—	Bits 31-16	—	—
TXCRC	Bits 15-0	R	<p>发送 CRC 值</p> <p>使能 CRC 计算后, TXCRC[15:0] 位将包含后续发送字节在计算后所得到的 CRC 值。当 SPI_CON1 寄存器中的 CRCEN 位写入 1 时, 此寄存器复位。CRC 通过 SPI_CRCPOLY 寄存器中编程的多项式连续计算。数据帧长度设置为 8 位数据 (SPI_CON1 的 FLEN 位清零) 时, 仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。选择 16 位数据帧长度 (SPI_CON1 寄存器的 FLEN 位置 1) 时, 考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC1 标准进行。注意: 当 BUSY 标志置 1 时, 读取此寄存器可能返回一个不正确的值。</p>

24. 7. 2. 8 SPI I2S 配置寄存器 (SPI_I2SCFG)

SPI I2S 配置寄存器 (SPI_I2SCFG)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				I2SMOD	I2SE	I2SCFG<1:0>	PCMSYNC		I2SSTD<1:0>	CKPOL	DATLEN<1:0>				CHLEN

—	Bits 31-12	—	—
I2SMOD	Bit 11	R/W	I2S 模式选择 0: 选择 SPI 模式 1: 选择 I2S 模式 注: 当 SPI 和 I2S 禁用时, 应配置该位。
I2SE	Bit 10	R/W	I2S 启用 0: 禁用 I2S 外设 1: 启用 I2S 外设 注: 该位不用于 SPI 模式。
I2SCFG	Bits 9-8	R/W	I2S 模式配置 00: 从机 - 发送 01: 从机 - 接收 10: 主机 - 发送 11: 主机 - 接收 注意: 禁用 I2S 时应配置这些位。它们不用于 SPI 模式。
PCMSYNC	Bit 7	R/W	PCM 帧同步 0: 短帧同步 1: 长帧同步 注意: 仅当 I2SSTD = 11 (使用 PCM 标准) 时, 该位才有意义。它不用于 SPI 模式。
—	Bit 6	—	—
I2SSTD	Bits 5-4	R/W	I2S 标准选择 00: I2S 飞利浦标准 01: MSB 对齐标准 (左对齐) 10: LSB 对齐标准 (右对齐) 11: PCM 标准 注意: 为了正确操作, 应在禁用 I2S 时配置这些位。它们不用于 SPI 模式。
CKPOL	Bit 3	R/W	非活动状态时钟极性 0: I2S 时钟非活动状态为低电平 1: I2S 时钟非活动状态是高电平

			<p>注意：为了正确操作，应在禁用 I2S 时配置该位。它不用于 SPI 模式。CKPOL 位不会影响用于接收或发送 SD 和 WS 信号的 CK 边沿灵敏度。</p>
DATLEN	Bits 2-1	R/W	<p>要传输的数据长度 00: 16 位数据长度 01: 24 位数据长度 10: 32 位数据长度 11: 不允许</p> <p>注意：为了正确操作，应在禁用 I2S 时配置这些位。它们不用于 SPI 模式。</p>
CHLEN	Bit 0	R/W	<p>信道长度（每个音频通道的位数） 0: 16 位宽 1: 32 位宽</p> <p>当 DATLEN 大于 00 时，通道长度需固定为 32 位。</p> <p>注意：为了正确操作，应在禁用 I2S 时配置该位。它不用于 SPI 模式。</p>

24.7.2.9 SPI I2S 预分频寄存器 (SPI_I2SPR)

SPI I2S 预分频寄存器 (SPI_I2SPR)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000010 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																						EXTCKEN	MCKOE	ODD	I2SDIV<7:0>						

—	Bits 31-11	—	—
EXTCKEN	Bit 10	R/W	外部 I2S 时钟使能 0: 选择 APB 时钟 1: 选择外部时钟 注意: 禁用 I2S 时应配置此位。该位不用于 SPI 模式。
MCKOE	Bit 9	R/W	主时钟输出使能 0: 禁用主时钟输出 1: 启用主时钟输出 注意: 禁用 I2S 时应配置此位。仅在 I2S 处于主模式时使用。它不用于 SPI 模式。
ODD	Bit 8	R/W	预分频器的奇数系数 0: 实分频器值= I2SDIV * 2 1: 实分频器值= (I2SDIV * 2) + 1 注意: 禁用 I2S 时应配置此位。仅在 I2S 处于主模式时使用。它不用于 SPI 模式。
I2SDIV	Bits 7-0	R/W	I2S 线性预分频器 I2SDIV [7: 0] = 0 是禁用值。 注意: 禁用 I2S 时应配置这些位。它们仅在 I2S 处于主模式时使用。它们不用于 SPI 模式。

			0: 无效 1: 启用中断
--	--	--	------------------

			0: 无效 1: 禁用中断
--	--	--	------------------

			0: 无效 1: 中断有效
--	--	--	------------------

			0: 无效 1: 原始中断标志状态
--	--	--	----------------------

24. 7. 2. 14 SPI 中断标志屏蔽状态寄存器 (SPI_IFM)

SPI 中断标志屏蔽状态寄存器 (SPI_IFM)																																
偏移地址: 34 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
													FREFM	MODFFM						RXTHFM	RXUDFM	RXOVFM	RXFFM					TXTHFM	TXUDFM	TXOVFM		TXEFM

—	Bits 31-19	—	—
FREFM	Bit 18	R	帧格式错误中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
MODFFM	Bit 17	R	模式故障中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
—	Bits 16-13	—	—
RXTHFM	Bit 12	R	接收 FIFO 超过阈值中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
RXUDFM	Bit 11	R	接收 FIFO 欠载中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
RXOVFM	Bit 10	R	接收 FIFO 溢出中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
RXFFM	Bit 9	R	接收 FIFO 满中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
—	Bits 8-5	—	—
TXTHFM	Bit 4	R	发送 FIFO 低于阈值中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
TXUDFM	Bit 3	R	发送 FIFO 欠载中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
TXOVFM	Bit 2	R	发送 FIFO 溢出中断标志屏蔽状态 0: 无效 1: 中断标志屏蔽状态
—	Bit 1	—	—
TXEFM	Bit 0	R	发送 FIFO 空中断标志屏蔽状态

			0: 无效 1: 中断标志屏蔽状态
--	--	--	----------------------

			0: 无效 1: 中断清除
--	--	--	------------------

第25章 无刷马达矢量控制器 (MCM)

25.1 概述

MCM (Motor Control Module)为针对电机应用所集成的控制模块，通过配置其中的独立模块可实现不同的电机控制算法如无感测速度积分、FOC (field orthogonal control)滑模观测器等。MCM 包含了滑动模式控制器的虚拟电机模型与用于计算角度的反正切运算模块 CORDIC、三个独立的 PI (Proportional-Integral)控制器，可用于调节电流、转速等电机控制参数。借由 MCM 中的高速运算模块，使得 MCU 不须处理制式的数学运算，而有更为充裕的时间做系统管理，实现更为高效、强健的电机控制器。MCU 通过 APB 总线将结果自 MCM 模块读取。

电机启动与电流重建需用户在软件当中实现，这样做的好处是避免 MCM 只能应用于特定的算法，使用上会更具弹性。MCM 模块额外提供三个独立的一阶 IIR 数字低通滤波器，可用于电流重建运算。

25.2 特性

- ◆ 用于电流重建的三个独立一阶 IIR 数字低通滤波器
- ◆ 提供速度或参考电压逐步改变的斜波产生器
- ◆ Clarke/Park 变换，将三相正弦电流映射至两相旋转坐标系
- ◆ Clarke/Park 逆变换，将电压矢量由两相旋转坐标系变换至三相定子坐标系，方便输出 PWM 讯号驱动功率组件
- ◆ 三个独立的 PI 控制器，可用于调节电流、转速等电机参数
- ◆ SVPWM (空间矢量脉宽调制)产生器
- ◆ 可配置参数的虚拟电机模型与滑模控制器，用以支持不同类型的电机
- ◆ 反正切函数 CORDIC，根据估算的 BEMF 计算角度

25.4 功能描述

25.4.1 斜波函数产生器

控制上一般通过调整 PWM 的占空比来改变电机的转速，但电机对占空比的变化反应速度较慢，因此为了给电机足够的时间来改变转速，需要逐渐增加/减少 PWM 占空比。斜波函数被用来避免参考值的剧烈变化，函数输出会逐渐地递增/递减到输出与设定值相等为止，如此一来，函数的设定(如占空比)产生变化时，输出的变化会较为平缓，给予电机充足的响应时间。

斜波函数可配置为独立模式来调整数据输出曲线，或是将其输出连接至 PI 控制器的输入端，以进行速度或转矩的控制。

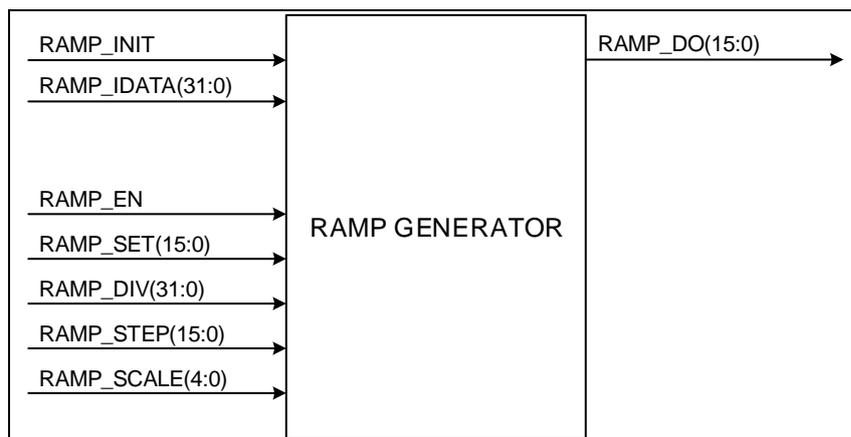


图 25-2 MCM 斜波函数产生器

斜波函数内部有一 32 位累加器，复位后默认值为零。初始化内部累加器时，需先将数值写入至 **RAMP_IDATA** 寄存器中，再将 **RAMP_INIT** 寄存器设为 1，如此便可将 **RAMP_IDATA** 中的数值写入内部累加器当中。**RAMP_INIT** 会被硬件自动清零。

斜波函数的更新频率为系统频率除以(**RAMP_DIV**+1)，举例来说，在 1MHz 系统频率下，将寄存器 **RAMP_DIV** 写为 999，则更新频率为 1000(1M / (999+1))。

通过对 **RAMP_EN** 寄存器设 1 启动斜波函数，在函数启动期间，累加器不断的与 **RAMP_SET** 寄存器值进行比较，当累加器值小于 **RAMP_SET** 寄存器时，则累加器递增，反之则累加器递减。累加器重复更新直到两者相等为止。每次的增量/减量则由 **RAMP_STEP** 控制。

RAMP_SCALE 用以缩减 **RAMP_STEP** 数值，两者间的转换关系为：

$$step\ value = RAMP_STEP \cdot 2^{-RAMP_SCALE}$$

其中: *step value* 为累加器每次的增量/减量。

RAMP_SCALE 有效值为 0~16。

举例来说，若 RAMP_STEP = 50，RAMP_SCALE = 5，则结果为：

$$step\ value = 50 \cdot 2^{-5} = 1.5625$$

斜波函数输出寄存器 RAMP_DO 为 16 位无号整数，其输出数值为内部累加器算术右移 RAMP_SCALE 位后所得的 16 位结果，所以实际上 RAMP_SCALE 寄存器即为内部累加器的小数点位置，例如，若 RAMP_SCALE 为 1 且当前的累加器数值为 2000，则出寄存器 RAMP_DO 的数值则为：

$$RAMP_DO = 2000 \cdot 2^{-RAMP_SCALE} = 2000 \cdot 2^{-1} = 1000$$

25.4.2 Clarke/ Park 逆变换

Clarke/Park 逆变换用以产生相互间 120° 的三相正弦波，空间矢量脉宽调制(SVPWM)则将三相正弦波进一步转换为空间矢量输出给逆变器。下图为 Clarke/Park 逆变换的方块图。

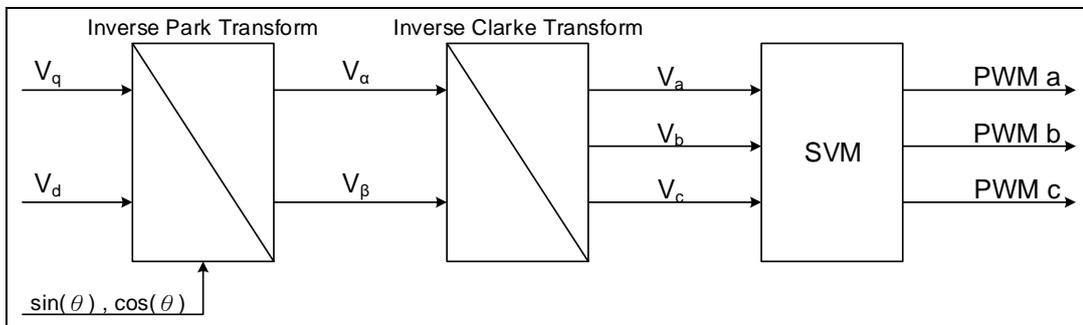


图 25-3 Clarke/Park 逆变换与 SVPWM 用以产生三相占空比

Clarke/Park 逆变换的输入可由端口寄存器或 PI 控制器的输出提供，其转换公式如下所示：

$$\begin{aligned} v_{\alpha} &= v_d * \cos(\theta) - v_q * \sin(\theta) \\ v_{\beta} &= v_q * \cos(\theta) + v_d * \sin(\theta) \\ v_a &= v_{\alpha} \\ v_b &= \frac{-v_{\alpha} + \sqrt{3} * v_{\beta}}{2} \\ v_c &= \frac{-v_{\alpha} - \sqrt{3} * v_{\beta}}{2} \end{aligned}$$

其中：

v_{α}, v_{β} 为两相正交坐标转换结果

v_a, v_b, v_c 为三相正弦波

$\cos(\theta), \sin(\theta)$ 为转子角度的正弦与余弦值

SVM 模块再将三相正弦波转换为空间矢量产生三相占空比用以驱动电机。下图为 MCM Clarke/Park 逆变换与 SVM 转换模块方块图。

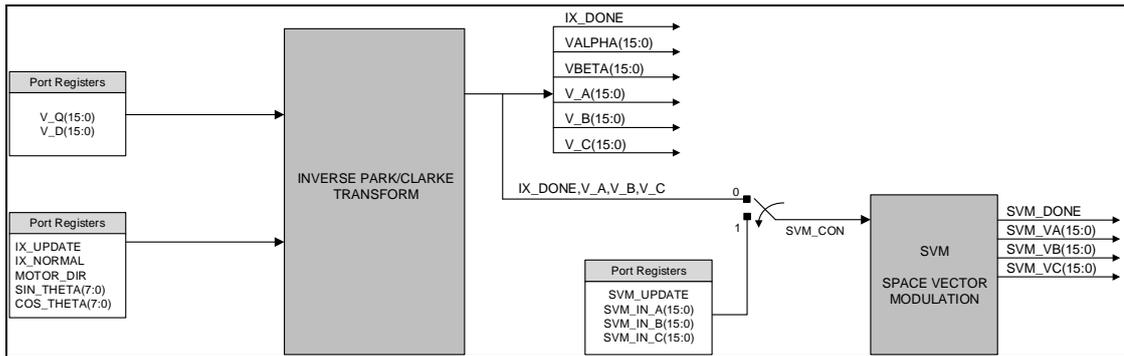


图 25-4 MCM Clarke/Park 逆变换与 SVM 转换模块

如上图所示，Clarke/Park 逆变换的输入来源可选择为寄存器 **V_Q**、**V_D**，SVM 模块的输入来源也可选择为寄存器 **SVM_IN_A**、**SVM_IN_B**、**SVM_IN_C**，或是使用逆变换的输出结果。默认情况下 SVM 模块的输入来源为逆变换输出结果，可通过修改 **SVM_CON** 寄存器改变输入来源。

SIN_THETA 与 **COS_THETA** 为当前转子角度的正弦与余弦值，**SIN_THETA** 与 **COS_THETA** 的数值分别代表 1 位的符号与 7 位的小数，例如，若当前转子角度为 15 度，则 **SIN_THETA** 为：

$$SIN\ THETA = sine(15) \cdot 2^7 = 0.2588 \cdot 128 = 33$$

若将 **MOTOR_DIR** 寄存器设定为 1，则 v_b 与 v_c 的计算结果将被互换，反应在电机控制上即会造成电机反转，默认情形下 **MOTOR_DIR** 值为 0。

可通过设定 **IX_NORMAL** 寄存器选择 Clarke 逆变换的计算方式，这种做法是为了将逆变换模块与 SVM 模块结合使用，所以，若有使用到 SVM 模块计算空间矢量 PWM 功能，则 **IX_NORMAL** 必须设置为 0，其 Clarke 逆变换的计算方式如下所示：

$$\begin{aligned} v_a &= v_\beta \\ v_b &= \frac{-v_\beta + \sqrt{3} * v_\alpha}{2} \\ v_c &= \frac{-v_\beta - \sqrt{3} * v_\alpha}{2} \end{aligned}$$

与标准的 Clarke 逆变换相比，在计算时将 v_α 与 v_β 对调。反之，若 **IX_NORMAL** 设置为 1，则是使用标准的 Clarke 计算。

对 **IX_UPDATE** 寄存器写入 1 启动 Clarke/Park 逆变换运算，硬件会自动将 **IX_UPDATE** 寄存器清零，运算完成后 **IX_DONE** 寄存器被置为 1。**VALPHA** 与 **VBETA** 为 Park 逆变换运算结果，**V_A**、**V_B** 与 **V_C** 为 Clarke 逆变换运算结果。**V_A**、**V_B** 与 **V_C** 为 16 位有号整数，其结果为相互间隔 120°的正弦值，可将数值做适当的缩放后做为 PWM 占空比输入。若有将 **V_A**、**V_B** 与 **V_C** 连接至 SVM 模块，则 SVM 模块输出 **SVM_VA**、**SVM_VB** 与 **SVM_VC** 亦可作为 PWM 占空比数值。

25.4.3 相电流 Clarke/Park 变换

Clarke/Park 变换将三相电流转换为与转子磁通同步旋转的两轴坐标系，转换后的坐标称为 d-q 轴，当中 θ 代表转子角度，如下图所示：

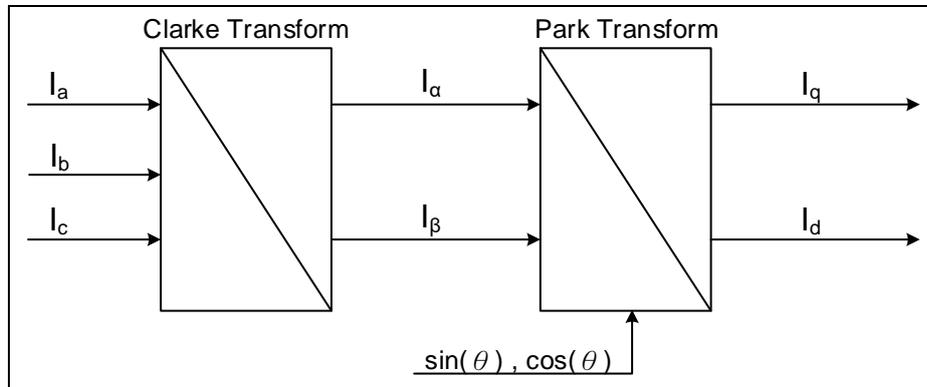


图 25-5 Clarke/ Park 变换

Clarke 变换将三相电流转换为两相固定坐标 α - β ，之后再通过 Park 变换将其转换成两相旋转 d-q 轴，转换的目的是把复杂的三相电流变换成简单的两相旋转坐标轴。当角度已知时，d-q 更易于控制，因为在稳态时可将 d、q 视为直流分量并分别控制。

Clarke/Park 变换的输入可由端口寄存器或低通滤波器的输出提供，其转换公式如下所示：

$$i_\alpha = i_a$$

$$i_\beta = \frac{(i_a + 2i_b)}{\sqrt{3}}$$

$$i_d = i_\alpha * \cos(\theta) + i_\beta * \sin(\theta)$$

$$i_q = i_\beta * \cos(\theta) - i_\alpha * \sin(\theta)$$

其中：

i_d, i_q 为两相旋转参考坐标

i_α, i_β 相互正交的静止坐标系

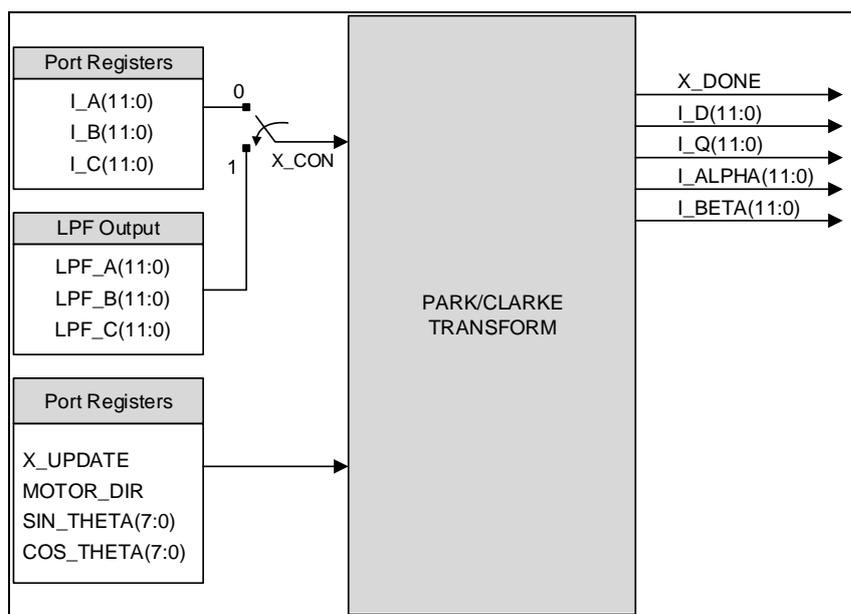


图 25-6 MCM Clarke/ Park 变换

配置 **X_CON** 寄存器选择 Clarke/Park 变换输入来源，写 **X_CON** 寄存器为 1 使用 LPF 模块的输出为输入来源。**MOTOR_DIR** 为电机正反转选择，**SIN_THETA** 与 **COS_THETA** 为上一章节提过的转子角度正、余弦值。

对 **X_UPDATE** 寄存器写入 1 启动 Clarke/Park 变换运算，硬件会自动将 **X_UPDATE** 寄存器清零，运算完成后 **X_DONE** 寄存器被置为 1。

I_ALPHA 与 **I_BETA** 为 Clarke 变换结果，**I_D** 与 **I_Q** 为 Park 变换结果。Clarke/Park 变换模块输入(**I_A**、**I_B**、**I_C** 或 **LPF_A**、**LPF_B**、**LPF_C**)与输出(**I_ALPHA**、**IBETA**、**I_D**、**I_Q**)皆为 12 位有号整数。

25.4.4 比例积分控制器(PI)

PI 控制器被广泛应用在工业控制当中，主要原理是通过比较量测参数与设定值之间的差异，由两者间的误差来决定控制器输出大小，使有效误差降至为零。下图为 PI 控制器的方块图。

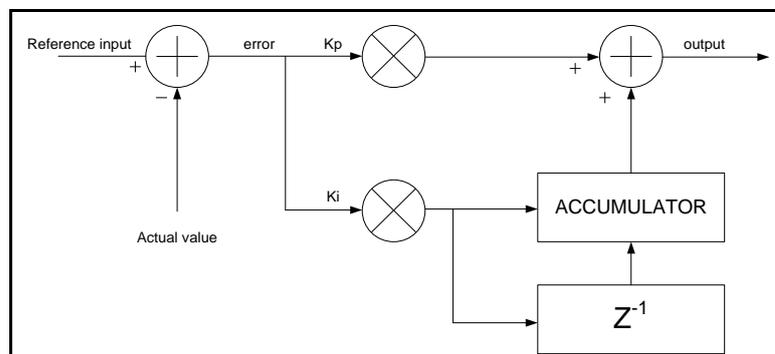


图 25-7 PI 控制器方块图

如上图所示，控制器当中有两个系数用来修正量测输入与设定值之间的误差，这两个系数需要仔细调整以达到最佳的控制响应。应用上可以使用 PI 控制器来控制电机的转速，设定值为目标转速，实际量测输入为当前电机转速，输出则为 PWM 占空比。当实际转速小于目标转速时，此时误差为正，会使得 PWM 占空比增加，连带使电机转速跟着上升，让实际转速向目标转速接近，直到两者相等时，占空比停止变化，电机转速停在目标转速。每当侦测到电机转速产生变化时，控制器便会根据产生的误差来调整占空比，直到误差被修正到零为止。简单的回馈机制让 PI 控制器成为工业控制的标准架构。PI 控制器的通用公式如下：

$$output = error * k_p + \int error * k_i dt$$

其中：

error 为设定值与实际量测值的误差

k_p 为比例常数

k_i 为积分常数

MCM PI 控制器设有上/下限输出限制器，防止输出超出系统可控制范围，此外，也可在启动 PI 控制器前将控制器初始化，以避免输出产生过于剧烈的变化。

MCM 中有三个独立的 PI 控制器，每个控制器都可以单独操作，控制不同的应用参数。例如：将三个 PI 控制器分别使用于电机转速、 I_d 与 I_q 控制。在默认情形下，控制器可独立操作，但也可以根据不同的控制应用进行配置，MCM 内建两种不同的控制类型，配置如下图所示：

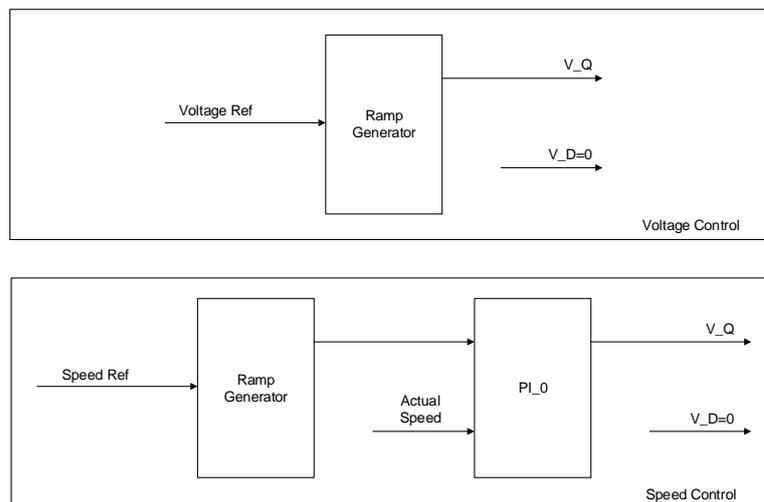


图 25-8 PI 控制器的两种控制配置

在电压控制(voltage control)中，三个 PI 控制器为独立操作模式。而在速度控制(speed control)当中，PI_0 被配置为速度控制器，控制器设定值与斜坡函数产生器的输出相连，速度设定由斜坡函数产生器控制，与电机实际转速进行比较，并通过修正误差来调整转速。

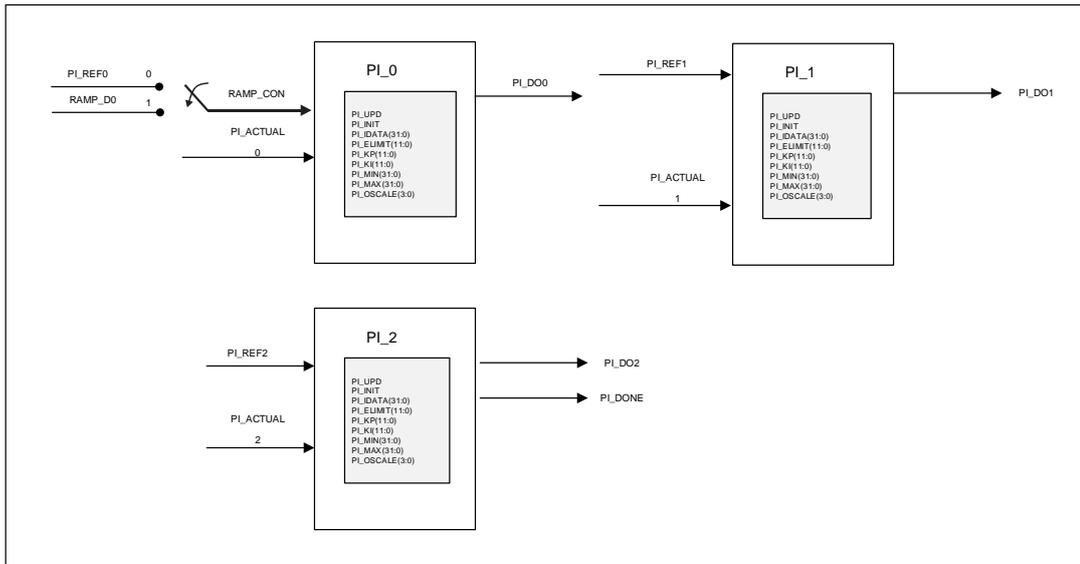


图 25-9 MCM PI 控制器

初始化 PI 控制器时，先将数据写入 **PI_IDATA** 寄存器中，再把 **PI_INIT** 寄存器置 1，即可将 **PI_IDATA** 中的数据加载至内部累加器中，**PI_INIT** 会被自动清零。**PI_KP** 与 **PI_KI** 分别为比例与积分常数，格式为 12 位无号小数，与实际数字间的转换公式如下所示：

$$coefficient = decimal \cdot 4096$$

例如，若比例常数 k_p 为 0.25，则 **PI_KP** 应该表示为：

$$PI_KP = 0.25 \cdot 4096 = 1024$$

PI 控制器中，**PI_REF** 与 **PI_ACTUAL** 分别代表控制器的参考值与实际输入值，两者皆为 16 位无号整数。而 **PI_ELIMIT** 寄存器代表控制器误差 *error* 的绝对值上限，最大可设置为 $2^{12} - 1 = 4095$ ，当 $error > PI_ELIMIT$ (或 $error < -PI_ELIMIT$) 时，防止过大的误差造成系统不稳定。

PI_SCALE 为 4 位寄存器 (有效值 0~15)，用以调整控制器输出的缩放比例，内部累加器在输出之前会根据 **PI_SCALE** 寄存器数值做算术右移。例如，内部累加器的数值为 2048，**PI_SCALE** 为 2，则控制器的输出会是：

$$PI_DO = 2048 \gg 2 = 512$$

控制器的输出上/下限分别由 **PI_MAX** 与 **PI_MIN** 寄存器控制，如此可使控制器输出维持在一定的范围之内，避免受控参数超出控制的有效值。例如，若受控参数为 PWM 占空比，通过设置 **PI_MAX** 与 **PI_MIN** 来限制输出占空比的范围，防止过大/过小的占空比导致电机不稳定，输出超出 **PI_MAX** 则会被限制在 **PI_MAX**，反之输出小于 **PI_MIN** 会被限制在 **PI_MIN**。

25.4.5 ADC 检测电流低通滤波器(LPF)

供重建电流使用之低通滤波器，将分流电阻上量测的原始 ADC 讯号做处理，用以消除 PWM 开噪声等讯号干扰。MCM 提供三个一阶 IIR 数字低通滤波器，在不同的截止频率下可通过配置系数加以处理。下图说明一阶 IIR 数字低通滤波器的架构与对应的转移函数：

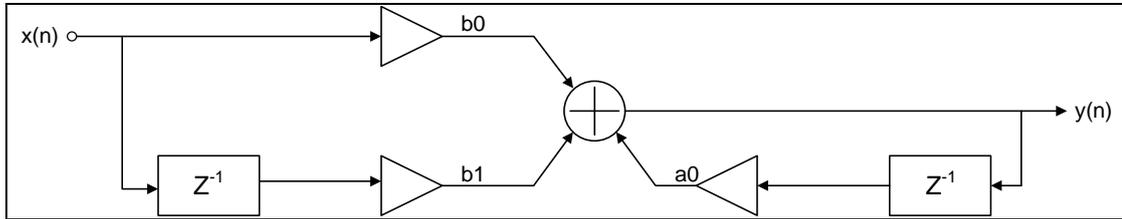


图 25-10 一阶 IIR 数字低通滤波器

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a_0 z^{-1}}$$

若将 b_1 设置为零，上图即为单极点低通递归滤波器(single pole low pass recursive filter)，其频率响应则由 a_0 与 b_0 决定，公式如下：

$$y[n] = b_0 \times x[n] + a_0 \times y[n - 1]$$

当中系数 a_0 为滤波器频率响应中的衰减因子，为了维持滤波器的稳定性与单位增益，一般将系数配置为：

$$b_0 = 1 - a_0$$

衰减因子 a_0 与截止频率的关系对应如下：

$$a_0 = e^{-2\pi f_c}$$

其中 f_c 为正规化(normalized)后的截止频率。例如，若 f_{pwm} 为 20kHz，所需截止频率为 100Hz，则：

$$f_c = \frac{100}{20000} = 0.005$$

$$a_0 = e^{-2\pi \times 0.005} = 0.969$$

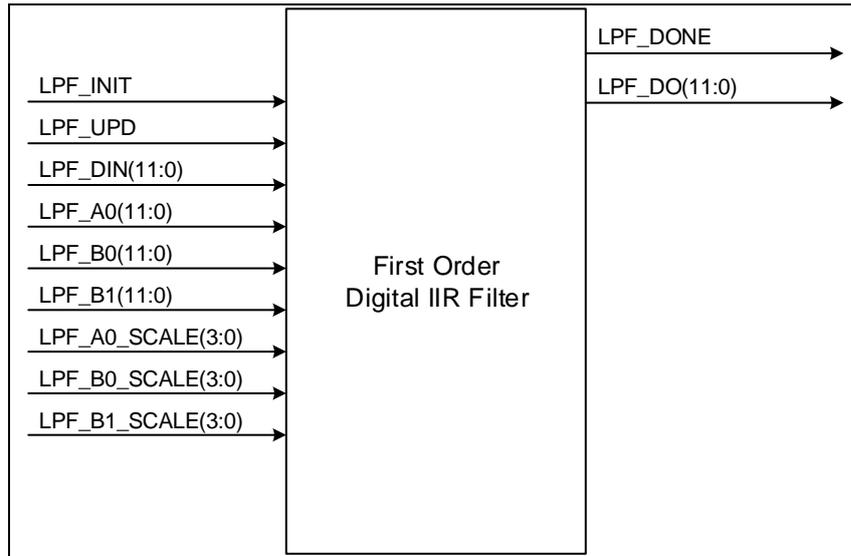


图 25-11 MCM 电流感测低通滤波器

初始化低通滤波器时，将 **LPF_INIT** 寄存器置 1 以清除输出与内部寄存器，**LPF_INIT** 被硬件自动清零。对 **LPF_UPD** 寄存器写 1 启动 LPF 计算，**LPF_UPD** 被硬件自动清零，计算完成后 **LPF_DONE** 被硬件置 1。

LPF_DIN 寄存器为滤波器输入，**LPF_DO** 滤波器输出，两者皆为 12 位有号数(有效范围：-2048~2047)。MCU 存取时会自动扩展为 32 位，不需要做额外的转换。

LPF_A0、**LPF_B0** 与 **LPF_B1** 为滤波器系数，格式为 1 位符号数与 11 位小数，与实际数字间的转换公式如下所示：

$$coefficient = decimal \cdot 2048$$

举例来说，若 $a_0 = 0.24$ ，则 **LPF_A0** 应该表示为：

$$LPF_A0 = 0.24 \cdot 2048 = 491$$

若 $b_0 = -0.35$ ，**LPF_B0** 则表示为：

$$LPF_B0 = -0.35 \cdot 2048 = -716$$

取 2 补码以表示负数，所以：-716 → 0xD34

通过寄存器的设定，可以进一步增加滤波器系数的分辨率，缩放系数的计算关系如下：

$$scaled\ coefficient = coefficient \cdot 2^{-scale}$$

举例来说，若系数 $b_1 = 0.24$ 且 b_1 的缩放系数(scale)为 1，则可计算出：

$$scaled\ coefficient = 0.24 \cdot 2^{-1} = 0.12$$

25. 4. 6 滑模观测理论(Sliding Mode Observer Theory)

在滑模观测器中(SMO)建构了一个虚拟电机模型，虚拟模型与实际电机有着相同的电压矢量输入，主要比较虚拟模型与实际电机间的电流差异，并将两者间的误差做为滑模控制器(slide mode controller, SMC)的输入。SMC 的主要目的是输出校正因子 z ，通过校正因子 z 来补偿虚拟模型与实际电机间的电流误差。SMC 中的线性增益系数 K_{slide} 被用来放大误差以产生校正因子 z ，但如果误差超出最大限制，校正因子 z 也会跟着受限。

校正因子 z 经过低通滤波器滤波后即为实际的 BEMF，此处的滤波器为自适应滤波器，截止频率将自动匹配电机的驱动电流频率。第二级的 LPF 被用来得到更为平滑的 BEMF 讯号，并将此讯号运用于反正切计算以获得估算的转子角度，MCM 中的反正切计算由 CORDIC 函数实现。

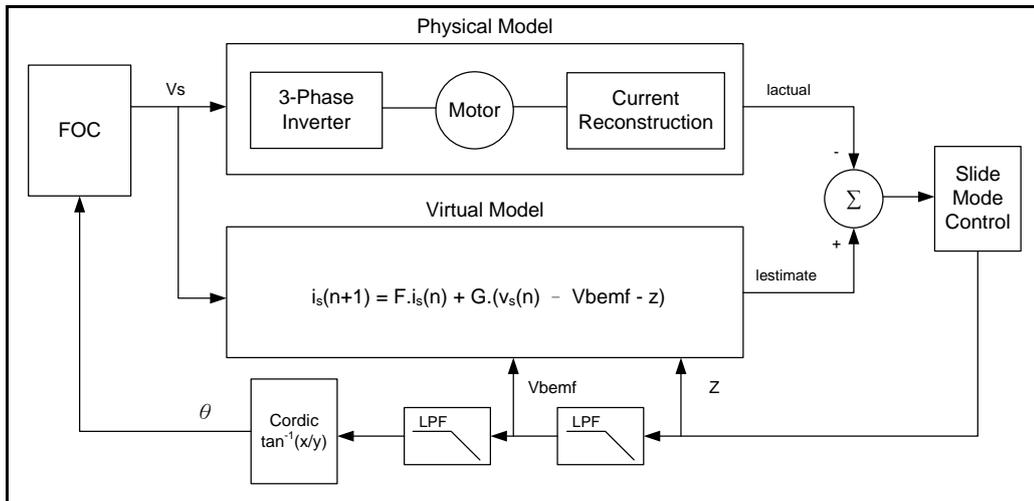


图 25-12 滑模观测器

25.4.7 虚拟电机模型

直流电机的模型如下所示：

$$v_s = Ri_s + L \frac{d}{dt} i_s + e_s$$

其中：

v_s 为输入电压矢量(V_α, V_β)

i_s 为电机电流矢量

e_s 为 BEMF 矢量

R 为电机电阻值

L 为电机电感值

T_s 为控制周期

通过求解 i_s 可得到电机电流：

$$\frac{d}{dt} i_s = -\left(\frac{R}{L}\right) i_s + \frac{1}{L}(v_s - e_s)$$

以数字的观点来看，上式可改写为：

$$\frac{i_s(n+1) - i_s(n)}{T_s} = \left(-\frac{R}{L}\right) i_s(n) + \frac{1}{L}(v_s(n) - e_s(n))$$

求解 i_s 可得：

$$i_s(n+1) = \left(1 - T_s \frac{R}{L}\right) i_s(n) + \frac{T_s}{L}(v_s(n) - e_s(n))$$

其中，令：

$$F = 1 - T_s \frac{R}{L}$$

$$G = \frac{T_s}{L}$$

因此，公式最终可写为：

$$i_s(n+1) = F \times i_s(n) + G \times (v_s(n) - e_s(n))$$

估算电流 i_s 为一矢量，代表 $i_s(\alpha)$ and $i_s(\beta)$ 两个分量，同理， v_s 与 e_s 也分别代表 $v_s(\alpha)$, $v_s(\beta)$, $e_s(\alpha)$, $e_s(\beta)$ 。

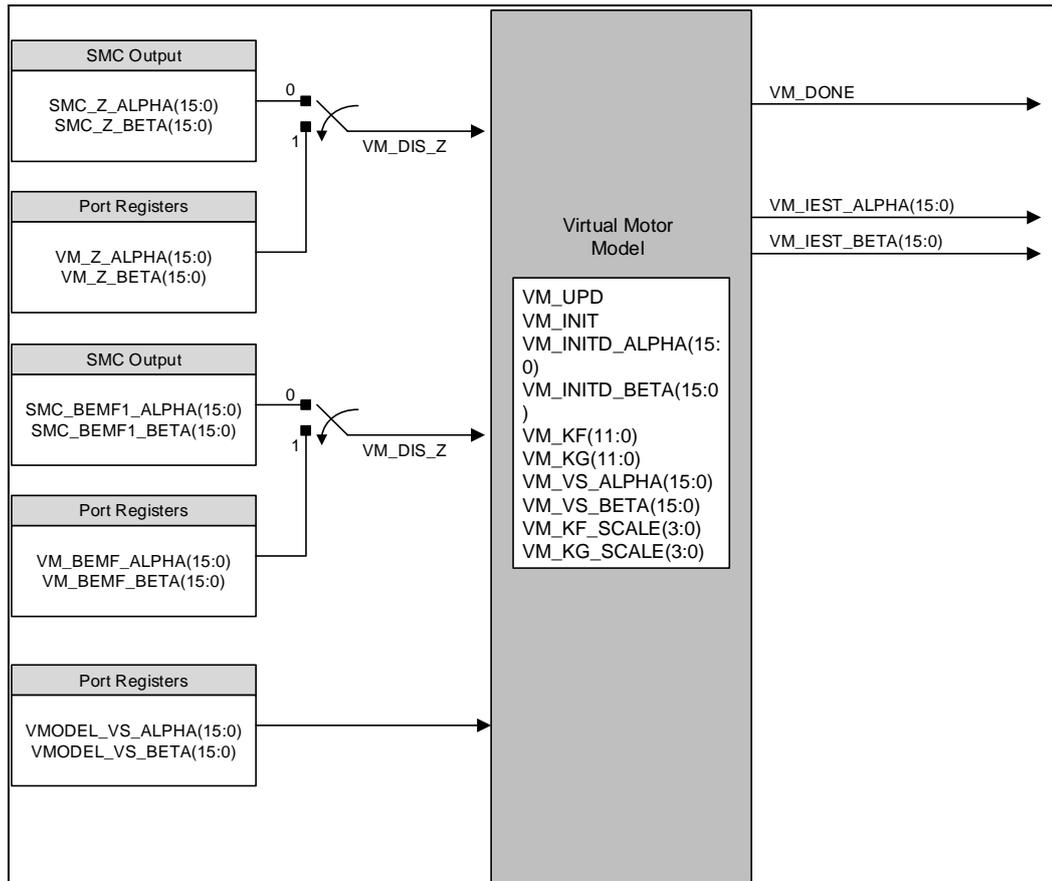


图 25-13 虚拟电机模型

虚拟电机模型被用来观测电流，根据实际电流与观测电流之间的误差来估算 BEMF。虚拟电机模型的输入可以配置为直接由滑模控制器的输出提供，或做为独立使用，如此一来输入由端口寄存器提供。默认情况下，输入由滑模控制器提供，写 VM_DIS_Z 寄存器为 1 来选择使用端口寄存器。

VM_IEST_ALPHA 与 VM_IEST_BETA, VM_INIT 被硬件自动清零。写 VM_UPD 为 1 来启动虚拟电机模型计算，VM_UPD 同样被硬件自动清零。VM_VS_ALPHA 代表电压矢量 VM_VS 的 alpha 方向分量，同理，VM_VS_BETA 为电压矢量 beta 方向分量。

VM_KF 与 VM_KG 为用于电流估算的常数，格式为 12 位无号小数，计算公式如下所示：

$$VM_KF = (1 - (T_s \cdot R/L)) \cdot 4095$$

$$VM_KG = \frac{T_s}{L} \cdot 4095$$

其中：

R 为电机电阻值

L 为电机电感值

T_s 为控制周期

例如， $R = 3.5\Omega$ 、 $L = 4.15\text{mH}$ 、 $T_s = 1/20\text{kHz}$ ，则：

$$VM_KF = (1 - ((0.00005 \times 3.5)/0.00415)) \times 4095 = 3922$$

$$VM_KG = \frac{0.00005}{0.00415} \times 4095 = 49$$

另外可通过 **KF_SCALE** 与 **KG_SCALE** 来增加 KF 与 KG 的分辨率，其关系为：

$$scaled\ VM_KF = VM_KF \cdot 2^{-kf_scale}$$

$$scaled\ VM_KG = VM_KG \cdot 2^{-kg_scale}$$

计算完成后，**VM_DONE** 寄存器自动置 1，估算的电流矢量分别为 **VM_IEST_ALPHA** 与 **VM_IEST_BETA**，格式为 16 位有号整数。

25.4.8 滑模控制器与动态 LPF

SMC 的输入为估算电流与实际流测电流的误差，控制器将输出校正因子 z 来补偿虚拟电机，使误差缩小至零。补偿虚拟电机时，可将校正因子 z 通过一低通滤波器来得到 BEMF 估计值，而 LPF 的截止频率需匹配当前矢量的频率，最后再根据 BEMF 的反正切值计算转子角度。SMC 流程图如下所示：

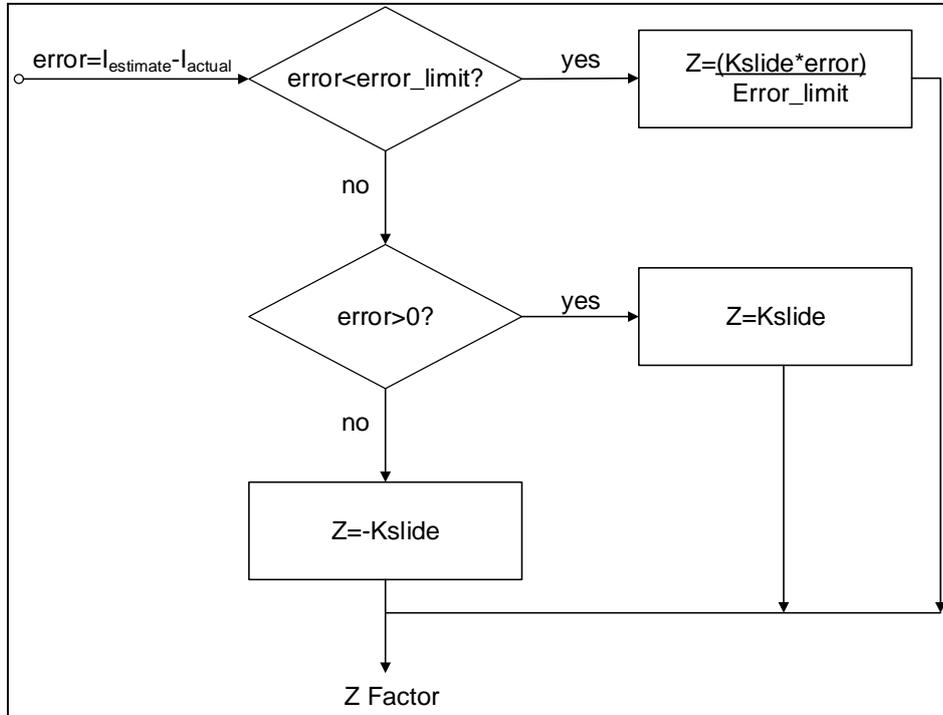


图 25-14 SMC 流程图

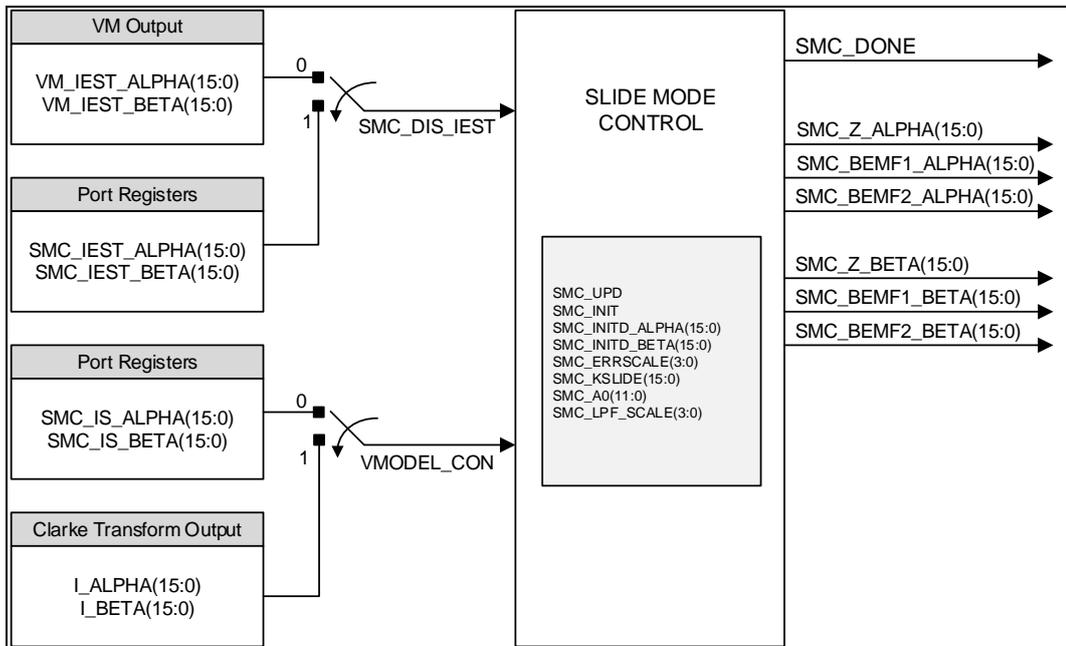


图 25-15 MCM 滑模控制器

配置 **SMC_DIS_IEST** 寄存器选择估测电流输入来源，写 1 选择使用端口寄存器，反之，使用虚拟电机输出做为输入来源。配置 **VMODEL_CON** 寄存器选择实际电流输入来源，写 0 选择使用端口寄存器，写 1 选择使用 Clarke 变换结果做为输入来源，内部会自动将 Clarke 变换输出(12 位)扩展为 16 位运算。

对 **SMC_INIT** 寄存器设置为 1 来初始化滑模控制模块，将 **SMC_INITD** 中的数据写入 **SMC_Z**、**SMC_BEMF1** 与 **SMC_BEMF2**，**SMC_INIT** 寄存器被硬件自动清零。

设置 **SMC_ERRSCALE** 寄存器决定误差上限，误差上限等同于 $2^{\text{SMC_ERRSCALE}}$ 。**SMC_KSLIDE** 寄存器为误差放大倍率，如图 25-15 所示，**SMC_KSLIDE** 格式为 16 位无号整数，最大有效值为 32768。

SMC_A0 为低通滤波器增益，目的是使 **SMC_Z** 结果更为平滑让 BEMF 估算更为准确，在设计上串联两个低通滤波器并使用相同的系数。**SMC_A0** 为 12 位无号小数，并有寄存器 **SMC_LPF_SCALE** 可增加 **SMC_A0** 的分辨率，计算方式如下：

$$\text{coefficient} = \frac{\text{SMC_A0}}{4095} \cdot 2^{-\text{SMC_LPF_SCALE}}$$

举例来说，若 **SMC_A0** = 250、**SMC_LPF_SCALE** = 2，则：

$$\text{coefficient} = \frac{250}{4095} \cdot 2^{-2} = 0.0153$$

将 **SMC_UPD** 寄存器置 1 启动滑模控制器计算，**SMC_UPD** 被硬件自动清零。计算完成后，**SMC_DONE** 被自动置 1。计算结果 **SMC_Z** 为修正因子，**SMC_BEMF1** 为 **SMC_Z** 经过低通滤波器的输出结果，用以估算 BEMF；**SMC_BEMF2** 为 **SMC_BEMF1** 经过第二个低通滤波器的输出结果，**SMC_BEMF2** 被做为 CORDIC 函数的输入用来估算转子角度。

25.4.9 CORDIC-反正切角度估算

MCM 中包含一 CORDIC(Coordinate Rotation by Digital Computer)硬件，其作用为计算 BEMF 的反正切数值以获得转子角度，该功能为 FOC 滑模观测器中的重要部分之一。CORDIC 利用加减法与位移运算即可计算常用的三角函数值，此算法具有高速与内存需求低的特点，适合于硬件上实现。下图为 CORDIC 方块图。

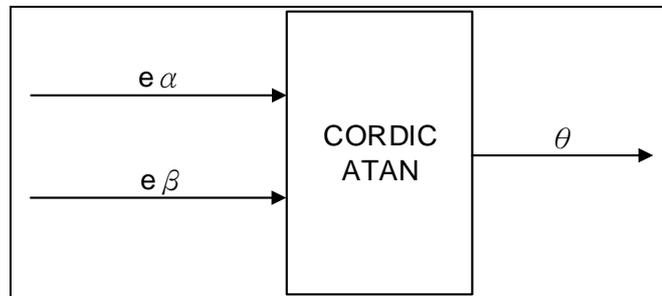


图 25-16 CORDIC Atan 方块图

CORDIC 的输出是经过正规化的 10-bit 无号整数，通过以下的公式可将输出结果转换为角度(degrees)或弧度(radians)。

$$degrees = (cordic_result * 360)/1024$$

$$radians = (cordic_result * 2\pi)/1024$$

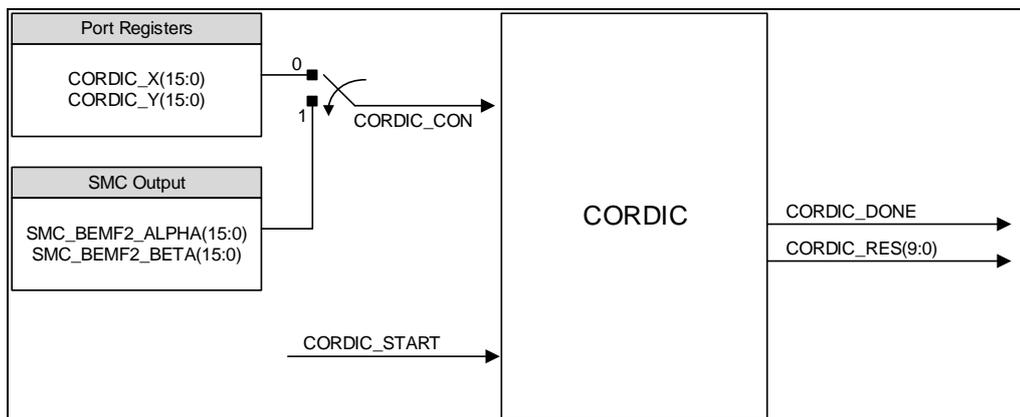


图 25-17 MCM CORDIC

配置 **CORDIC_CON** 寄存器选择 CORDIC 函数的输入来源，写 0 使用端口寄存器做为输入来源，写 1 将滑模控制输出 BEMF2 连接至 CORDIC 做为输入来源。输入格式为 16 位无号整数。

设置 **CORDIC_START** 寄存器为 1 启动 CORDIC 计算，**CORDIC_START** 被硬件自动清零。计算完成后 **CORDIC_DONE** 寄存器自动置 1，读取 **CORDIC_RES** 寄存器得知运算结果，**CORDIC_RES** 格式为 10 位无号整数，代表将圆 360°平分为 1024 等分，计算结果与角度间的转换如上面所描述。

25.5 特殊功能寄存器

25.5.1 寄存器列表

MCM 寄存器列表			
名称	偏移地址	类型	描述
MCM_RAMP_INIT	000 _H	R/W	斜波函数产生器初始化寄存器
MCM_RAMP_IDATA	004 _H	R/W	斜波函数产生器初始值寄存器
MCM_RAMP_DIV	008 _H	R/W	斜波函数分频器寄存器
MCM_RAMP_SCALE	00C _H	R/W	斜波函数输出比例寄存器
MCM_RAMP_STEP	010 _H	R/W	斜波函数输出增量寄存器
MCM_RAMP_SET	014 _H	R/W	斜波函数参考值寄存器
MCM_RAMP_EN	018 _H	R/W	斜波函数使能寄存器
MCM_RAMP_DO	01C _H	R/W	斜波函数输出寄存器
MCM_RMAP_CON	020 _H	R/W	斜波函数内部连接选择寄存器
MCM_IX_UPD	024 _H	R/W	逆变换更新寄存器
MCM_IX_VQ	028 _H	R/W	Park 逆变换输入 Vq 值寄存器
MCM_IX_VD	02C _H	R/W	Park 逆变换输入 Vd 值寄存器
MCM_IX_VA	030 _H	R	Clarke 逆变换输出 Va 值寄存器
MCM_IX_VB	034 _H	R	Clarke 逆变换输出 Vb 值寄存器
MCM_IX_VC	038 _H	R	Clarke 逆变换输出 Vc 值寄存器
MCM_IX_VALPHA	03C _H	R	Clarke 逆变换输出 V α 值寄存器
MCM_IX_VBETA	040 _H	R	Clarke 逆变换输出 V β 值寄存器
MCM_IX_SVA	044 _H	R	SVPWM Va 输出寄存器
MCM_IX_SVB	048 _H	R	SVPWM Vb 输出寄存器
MCM_IX_SVC	04C _H	R	SVPWM Vc 输出寄存器
MCM_IX_DONE	050 _H	R	逆变换处理状态寄存器
MCM_IX_SDONE	054 _H	R	SVPWM 处理状态寄存器
MCM_IX_NORMAL	05C _H	R/W	Clarke 逆变换计算方式选择寄存器
MCM_X_UPD	060 _H	R/W	Clarke/Park 变换更新寄存器
MCM_X_IA	064 _H	R/W	Clarke 变换输入 Ia 值寄存器
MCM_X_IB	068 _H	R/W	Clarke 变换输入 Ib 值寄存器
MCM_X_IC	06C _H	R/W	Clarke 变换输入 Ic 值寄存器
MCM_X_IALPHA	070 _H	R	Clarke 变换输出 I α 值寄存器
MCM_X_IBETA	074 _H	R	Clarke 变换输出 I β 值寄存器
MCM_X_ID	078 _H	R	Park 变换输出 Id 值寄存器
MCM_X_IQ	07C _H	R	Park 变换输出 Iq 值寄存器
MCM_X_DONE	080 _H	R	变换处理状态寄存器
MCM_X_CON	084 _H	R/W	Clarke 变换输入来源选择寄存器
MCM_PI0_UPD	08C _H	R/W	PI0 控制器更新寄存器
MCM_PI0_INI	090 _H	R/W	PI0 控制器初始化设定寄存器
MCM_PI0_IDATA	094 _H	R/W	PI0 控制器初始值寄存器
MCM_PI0_ELIMIT	098 _H	R/W	PI0 控制器误差上限寄存器

MCM_PI0_KP	09C _H	R/W	PI0 控制器 Kp 参数寄存器
MCM_PI0_KI	0A0 _H	R/W	PI0 控制器 Ki 参数寄存器
MCM_PI0_REF	0A4 _H	R/W	PI0 控制器参考值寄存器
MCM_PI0_ACTUAL	0A8 _H	R/W	PI0 控制器实际值寄存器
MCM_PI0_MIN	0AC _H	R/W	PI0 控制器输出下限寄存器
MCM_PI0_MAX	0B0 _H	R/W	PI0 控制器输出上限寄存器
MCM_PI0_DO	0B4 _H	R	PI0 控制器输出寄存器
MCM_PI0_OSCALE	0B8 _H	R/W	PI0 控制器输出比例寄存器
MCM_PI_DONE	0BC _H	R	PI 控制器处理状态寄存器
MCM_PI1_UPD	0C0 _H	R/W	PI1 控制器更新寄存器
MCM_PI1_INI	0C4 _H	R/W	PI1 控制器初始化设定寄存器
MCM_PI1_IDATA	0C8 _H	R/W	PI1 控制器初始值寄存器
MCM_PI1_ELIMIT	0CC _H	R/W	PI1 控制器误差上限寄存器
MCM_PI1_KP	0D0 _H	R/W	PI1 控制器 Kp 参数寄存器
MCM_PI1_KI	0D4 _H	R/W	PI1 控制器 Ki 参数寄存器
MCM_PI1_REF	0D8 _H	R/W	PI1 控制器参考值寄存器
MCM_PI1_ACTUAL	0DC _H	R/W	PI1 控制器实际值寄存器
MCM_PI1_MIN	0E0 _H	R/W	PI1 输出下限寄存器
MCM_PI1_MAX	0E4 _H	R/W	PI1 输出上限寄存器
MCM_PI1_DO	0E8 _H	R	PI1 控制器输出寄存器
MCM_PI1_OSCALE	0EC _H	R/W	PI1 控制器输出比例寄存器
MCM_PI2_UPD	0F0 _H	R/W	PI2 控制器更新寄存器
MCM_PI2_INI	0F4 _H	R/W	PI2 控制器初始化设定寄存器
MCM_PI2_IDATA	0F8 _H	R/W	PI2 控制器初始值寄存器
MCM_PI2_ELIMIT	0FC _H	R/W	PI2 控制器误差上限寄存器
MCM_PI2_KP	100 _H	R/W	PI2 控制器 Kp 参数寄存器
MCM_PI2_KI	104 _H	R/W	PI2 控制器 Ki 参数寄存器
MCM_PI2_REF	108 _H	R/W	PI2 控制器参考值寄存器
MCM_PI2_ACTUAL	10C _H	R/W	PI2 控制器实际值寄存器
MCM_PI2_MIN	110 _H	R/W	PI2 输出下限寄存器
MCM_PI2_MAX	114 _H	R/W	PI2 输出上限寄存器
MCM_PI2_DO	118 _H	R/W	PI2 控制器输出寄存器
MCM_PI2_OSCALE	11C _H	R/W	PI2 控制器输出比例寄存器
MCM_COR_START	120 _H	R/W	CORDIC 计算启动寄存器
MCM_COR_X	124 _H	R/W	CORDIC 函数 X 分量输入寄存器
MCM_COR_Y	128 _H	R/W	CORDIC 函数 Y 分量输入寄存器
MCM_COR_RES	12C _H	R	CORDIC 计算结果寄存器
MCM_COR_DONE	130 _H	R	CORDIC 处理状态寄存器
MCM_COR_CON	134 _H	R/W	CORDIC 输入来源选择寄存器
MCM_LPFA_INI	140 _H	R/W	LPF A 初始化设定寄存器
MCM_LPFA_UPD	144 _H	R/W	LPF A 更新控制寄存器
MCM_LPFA_DIN	148 _H	R/W	LPF A 输入讯号寄存器
MCM_LPFA_A0	14C _H	R/W	LPF A a0 系数寄存器

MCM_LPFA_B0	150 _H	R/W	LPF A b0 系数寄存器
MCM_LPFA_B1	154 _H	R/W	LPF A b1 系数寄存器
MCM_LPFA_A0SCALE	158 _H	R/W	LPF A a0 系数输出比例寄存器
MCM_LPFA_B0SCALE	15C _H	R/W	LPF A b0 系数输出比例寄存器
MCM_LPFA_B1SCALE	160 _H	R/W	LPF A b1 系数输出比例寄存器
MCM_LPFA_DO	164 _H	R	LPF A 输出结果寄存器
MCM_LPFA_DONE	168 _H	R	LPF A 处理状态寄存器
MCM_LPFB_INI	16C _H	R/W	LPF B 初始化设定寄存器
MCM_LPFB_UPD	170 _H	R/W	LPF B 更新控制寄存器
MCM_LPFB_DIN	174 _H	R/W	LPF B 输入讯号寄存器
MCM_LPFB_A0	178 _H	R/W	LPF B a0 系数寄存器
MCM_LPFB_B0	17C _H	R/W	LPF B b0 系数寄存器
MCM_LPFB_B1	180 _H	R/W	LPF B b1 系数寄存器
MCM_LPFB_A0SCALE	184 _H	R/W	LPF B a0 系数输出比例寄存器
MCM_LPFB_B0SCALE	188 _H	R/W	LPF B b0 系数输出比例寄存器
MCM_LPFB_B1SCALE	18C _H	R/W	LPF B b1 系数输出比例寄存器
MCM_LPFB_DO	190 _H	R	LPF B 输出结果寄存器
MCM_LPFB_DONE	194 _H	R	LPF B 处理状态寄存器
MCM_LPFC_INI	198 _H	R/W	LPF C 初始化设定寄存器
MCM_LPFC_UPD	19C _H	R/W	LPF C 更新控制寄存器
MCM_LPFC_DIN	1A0 _H	R/W	LPF C 输入讯号寄存器
MCM_LPFC_A0	1A4 _H	R/W	LPF C a0 系数寄存器
MCM_LPFC_B0	1A8 _H	R/W	LPF C b0 系数寄存器
MCM_LPFC_B1	1AC _H	R/W	LPF C b1 系数寄存器
MCM_LPFC_A0SCALE	1B0 _H	R/W	LPF C a0 系数输出比例寄存器
MCM_LPFC_B0SCALE	1B4 _H	R/W	LPF C b0 系数输出比例寄存器
MCM_LPFC_B1SCALE	1B8 _H	R/W	LPF C b1 系数输出比例寄存器
MCM_LPFC_DO	1BC _H	R	LPF C 输出结果寄存器
MCM_LPFC_DONE	1C0 _H	R	LPF C 处理状态寄存器
MCM_GL_UPD	1C4 _H	R/W	Global 更新寄存器
MCM_GL_STAT	1C8 _H	R	Global 状态寄存器
MCM_THETA_SIN	1CC _H	R/W	正弦数值寄存器
MCM_THETA_COS	1D0 _H	R/W	余弦数值寄存器
MCM_PI_TYPE	1D4 _H	R/W	PI 控制器类型设定寄存器
MCM_MOTOR_DIR	1D8 _H	R/W	电机旋转方向选择寄存器
MCM_VER	1DC _H	R	MCM 版本号寄存器
MCM_SMC_UPD	1E0 _H	R/W	SMC 更新控制寄存器
MCM_SMC_INI	1E4 _H	R/W	SMC 初始化设定寄存器
MCM_SMC_ALPHAINI	1E8 _H	R/W	SMC Alpha 分量初始值寄存器
MCM_SMC_BETAINI	1EC _H	R/W	SMC Beta 分量初始值寄存器
MCM_SMC_A0	1F0 _H	R/W	SMC 动态 LPF 系数寄存器
MCM_SMC_ELIMIT	1F4 _H	R/W	SMC 误差上限寄存器
MCM_SMC_KSLIDE	1F8 _H	R/W	SMC Z 因子放大倍率寄存器

MCM_SMC_DONE	1FC _H	R	SMC 处理状态寄存器
MCM_SMC_IEALPHA	200 _H	R/W	SMC 估测电流 Alpha 分量寄存器
MCM_SMC_IEBETA	204 _H	R/W	SMC 估测电流 Beta 分量寄存器
MCM_SMC_IALPHA	208 _H	R/W	SMC 量测电流 Alpha 分量寄存器
MCM_SMC_IBETA	20C _H	R/W	SMC 量测电流 Beta 分量寄存器
MCM_SMC_DISI	210 _H	R/W	SMC 估测电流输入来源选择寄存器
MCM_SMC_ZALPHA	214 _H	R	SMC Z 因子 Alpha 分量寄存器
MCM_SMC_ZBETA	218 _H	R	SMC Z 因子 Beta 分量寄存器
MCM_SMC_BEMF1A	21C _H	R	SMC 估测 BEMF Alpha 分量寄存器
MCM_SMC_BEMF1B	220 _H	R	SMC 估测 BEMF Beta 分量寄存器
MCM_SMC_BEMF2A	224 _H	R	SMC 滤波后 BEMF Alpha 分量寄存器
MCM_SMC_BEMF2B	228 _H	R	SMC 滤波后 BEMF Beta 分量寄存器
MCM_SMC_LPFSCALE	22C _H	R/W	SMC LPF 系数比例因子寄存器
MCM_VM_UPD	234 _H	R/W	虚拟电机更新控制寄存器
MCM_VM_INI	238 _H	R/W	虚拟电机初始化设定寄存器
MCM_VM_ALPHAINI	23C _H	R/W	虚拟电机 Alpha 分量初值寄存器
MCM_VM_BETAINI	240 _H	R/W	虚拟电机 Beta 分量初值寄存器
MCM_VM_KF	244 _H	R/W	虚拟电机 F 系数寄存器
MCM_VM_KG	248 _H	R/W	虚拟电机 G 系数寄存器
MCM_VM_VALPHA	24C _H	R/W	虚拟电机电压 Alpha 分量寄存器
MCM_VM_VBETA	250 _H	R/W	虚拟电机电压 Beta 分量寄存器
MCM_VM_BEMFA	254 _H	R/W	虚拟电机 BEMF Alpha 分量寄存器
MCM_VM_BEMFB	258 _H	R/W	虚拟电机 BEMF Beta 分量寄存器
MCM_VM_ZALPHA	25C _H	R/W	虚拟电机 Z 因子 Alpha 分量寄存器
MCM_VM_ZBETA	260 _H	R/W	虚拟电机 Z 因子 Beta 分量寄存器
MCM_VM_DISZ	264 _H	R/W	虚拟电机 Z 因子输入来源选择寄存器
MCM_VM_DONE	268 _H	R	虚拟电机处理状态寄存器
MCM_VM_IEALPHA	26C _H	R	虚拟电机估测结果 Alpha 寄存器
MCM_VM_IEBETA	270 _H	R	虚拟电机估测结果 Beta 寄存器
MCM_VM_CON	274 _H	R/W	虚拟电机实际电流输入来源选择寄存器
MCM_VM_KFSCALE	278 _H	R/W	虚拟电机 F 系数输出比例寄存器
MCM_VM_KGSCALE	27C _H	R/W	虚拟电机 G 系数输出比例寄存器
MCM_SVM_UPD	288 _H	R/W	SVPWM 更新控制寄存器
MCM_SVM_CON	28C _H	R/W	SVPWM 输入来源选择寄存器
MCM_SVM_DINA	290 _H	R/W	SVPWM A 相输入寄存器
MCM_SVM_DINB	294 _H	R/W	SVPWM B 相输入寄存器
MCM_SVM_DINC	298 _H	R/W	SVPWM C 相输入寄存器

25.5.2 寄存器描述

25.5.2.1 斜坡函数产生器初始化寄存器(MCM_RAMP_INIT)

斜坡函数产生器初始化寄存器(MCM_RAMP_INIT)
偏移地址: 00 _H

复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															RAMP_INIT

—	Bits 31-1	—	—
RAMP_INIT	Bit 0	W1	斜波函数产生器初始化设定位 写1以初始化内部累加器，此位被硬件自动清零

25.5.2.2 斜波函数产生器初始值寄存器(MCM_RAMP_IDATA)

斜波函数产生器初始值寄存器(MCM_RAMP_IDATA)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAMP_IDATA<31:0>																															

RAMP_IDATA	Bits 31-0	R/W	斜波函数产生器初始值 当斜波函数产生器初始化时将此值加载累加器当中
------------	-----------	-----	--------------------------------------

25.5.2.3 斜波函数分频器寄存器(MCM_RAMP_DIV)

斜波函数分频器寄存器(MCM_RAMP_DIV)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAMP_DIV<31:0>																															

RAMP_DIV	Bits 31-0	R/W	斜波函数分频器 系统频率分频器以控制斜波函数变动速率，其变化率为: PCLK / (CLKDIV + 1)
----------	-----------	-----	---

25.5.2.4 斜波函数输出比例寄存器(MCM_RAMP_SCALE)

斜波函数输出比例寄存器(MCM_RAMP_SCALE)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAMP_SCALE<4:0>																															

—	Bits 31-5	—	—
RAMP_SCALE	Bits 4-0	R/W	斜波函数输出比例 设定斜波函数步进值比例，对应关系为： 步进值 = RAMP_STEP / (1 << RAMP_SCALE) 举例来说： RAMP_STEP = 100; RAMP_SCALE = 2; 步进值 = 100 / (1 << 2) = 25

25.5.2.5 斜波函数输出增量寄存器(MCM_RAMP_STEP)

斜波函数输出增量寄存器(MCM_RAMP_STEP)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RAMP_STEP<15:0>															

—	Bits 31-16	—	—
RAMP_STEP	Bits 15-0	R/W	斜波函数输出增量 斜波函数的增加/递减量设定

25.5.2.6 斜波函数参考值寄存器(MCM_RAMP_SET)

斜波函数参考值寄存器(MCM_RAMP_SET)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RAMP_SET<15:0>															

—	Bits 31-16	—	—
RAMP_SET	Bits 15-0	R/W	斜波函数参考值 当斜波函数输出小于此参考值时，输出将逐渐增加直到两者相等，反之亦然。为了避免输出在参考值上下间振荡，建议将参考值RAMP_SET 设定为输出增量RAMP_STEP的倍数。

25.5.2.7 斜波函数使能寄存器(MCM_RAMP_EN)

斜波函数使能寄存器(MCM_RAMP_EN)																															
偏移地址: 18H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															RAMP_EN

—	Bits 31-1	—	—
RAMP_EN	Bit 0	R/W	斜波函数使能位 斜波函数的使能/禁止。禁止时函数输出会维持在最后的数值，使能后亦从最后数值开始增/减。 0: 禁止斜波函数 1: 使能斜波函数

25.5.2.8 斜波函数输出寄存器(MCM_RAMP_DO)

斜波函数输出寄存器(MCM_RAMP_DO)																															
偏移地址: 1CH																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RAMP_DO<15:0>															

—	Bits 31-16	—	—
RAMP_DO	Bits 15-0	R	斜波函数输出值

25.5.2.9 斜波函数内部连接选择寄存器(MCM_RAMP_CON)

斜波函数内部连接选择寄存器(MCM_RAMP_CON)																																	
偏移地址: 20H																																	
复位值: 00000000_00000000_00000000_00000000 _b																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RAMP_CON	

—	Bits 31-1	—	—
RAMP_CON	Bit 0	R/W	斜波函数内部连接选择 0: 禁止将斜波函数输出连接至PI0控制器。 1: 将斜波函数输出连接至PI0控制器的参考值。

25.5.2.10 逆变换更新寄存器(MCM_IX_UPD)

逆变换更新寄存器(MCM_IX_UPD)																																
偏移地址: 24H																																
复位值: 00000000_00000000_00000000_00000000 _b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IX_UPD

—	Bits 31-1	—	—
IX_UPD	Bit 0	W1	Clarke/Park逆变换更新位 写1以更新Clarke/Park逆变换结果。此位被硬件自动清零

25.5.2.11 Park 逆变换输入 Vq 值寄存器(MCM_IX_VQ)

Park 逆变换输入 Vq 值寄存器(MCM_IX_VQ)																																														
偏移地址: 28 _H																																														
复位值: 00000000_00000000_00000000_00000000 _B																																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																IX_VQ<15:0>																														

—	Bits 31-16	—	—
IX_VQ	Bits 15-0	R/W	Park 逆变换中的Vq值

25.5.2.12 Park 逆变换输入 Vd 值寄存器(MCM_IX_VD)

Park 逆变换输入 Vd 值寄存器(MCM_IX_VD)																																														
偏移地址: 2C _H																																														
复位值: 00000000_00000000_00000000_00000000 _B																																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																IX_VD<15:0>																														

—	Bits 31-16	—	—
IX_VD	Bits 15-0	R/W	Park 逆变换中的Vd值

25.5.2.13 Clarke 逆变换输出 Va 值寄存器(MCM_IX_VA)

Clarke 逆变换输出 Va 值寄存器(MCM_IX_VA)																																															
偏移地址: 30 _H																																															
复位值: 00000000_00000000_00000000_00000000 _b																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
																IX_VA<15:0>																															

—	Bits 31-16	—	—
IX_VA	Bits 15-0	R	Clarke 逆变换中的Va值 16位有号整数

25.5.2.14 Clarke 逆变换输出 Vb 值寄存器(MCM_IX_VB)

Clarke 逆变换输出 Vb 值寄存器(MCM_IX_VB)																																															
偏移地址: 34 _H																																															
复位值: 00000000_00000000_00000000_00000000 _b																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
																IX_VB<15:0>																															

—	Bits 31-16	—	—
IX_VB	Bits 15-0	R	Clarke 逆变换中的 Vb 值 16位有号整数

25.5.2.15 Clarke 逆变换输出 Vc 值寄存器(MCM_IX_VC)

Clarke 逆变换输出 Vc 值寄存器(MCM_IX_VC)																																													
偏移地址: 38H																																													
复位值: 00000000_00000000_00000000_00000000 _b																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																IX_VC<15:0>																													

—	Bits 31-16	—	—
IX_VC	Bits 15-0	R	Clarke 逆变换中的 Vc 值 16位有号整数

25.5.2.16 Clarke 逆变换输出 Va 值寄存器(MCM_IX_VALPHA)

Clarke 逆变换输出 Va 值寄存器(MCM_IX_VALPHA)																																													
偏移地址: 3CH																																													
复位值: 00000000_00000000_00000000_00000000 _b																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																IX_VALPHA<15:0>																													

—	Bits 31-16	—	—
IX_VALPHA	Bits 15-0	R	Clarke 逆变换中的 Va 值 16位有号整数

25.5.2.17 Clarke 逆变换输出 $V\beta$ 值寄存器(MCM_IX_VBETA)

Clarke 逆变换输出 $V\beta$ 值寄存器(MCM_IX_VBETA)																																													
偏移地址: 40 _H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																$IX_VBETA<15:0>$																													

—	Bits 31-16	—	—
IX_VBETA	Bits 15-0	R	Clarke 逆变换中的 $V\beta$ 值 16位有号整数

25.5.2.18 SVPWM Va 输出寄存器(MCM_IX_SVA)

SVPWM Va 输出寄存器(MCM_IX_SVA)																																													
偏移地址: 44 _H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																$IX_SVA<15:0>$																													

—	Bits 31-16	—	—
IX_SVA	Bits 15-0	R	SVPWM 中的 Va 输出

25.5.2.19 SVPWM Vb 输出寄存器(MCM_IX_SVB)

SVPWM Vb 输出寄存器(MCM_IX_SVB)																																													
偏移地址: 48H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																IX_SVB<15:0>																													

—	Bits 31-16	—	—
IX_SVB	Bits 15-0	R	SVPWM 中的 Vb 输出

25.5.2.20 SVPWM Vc 输出寄存器(MCM_IX_SVC)

SVPWM Vc 输出寄存器(MCM_IX_SVC)																																													
偏移地址: 4CH																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																IX_SVC<15:0>																													

—	Bits 31-16	—	—
IX_SVC	Bits 15-0	R	SVPWM 中的 Vc 输出

25.5.2.21 逆变换处理状态寄存器(MCM_IX_DONE)

逆变换处理状态寄存器(MCM_IX_DONE)																																
偏移地址: 50H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IX_DONE

—	Bits 31-1	—	—
IX_DONE	Bit 0	R	逆变换处理状态 当逆变换计算完毕, 此位会被硬件置1

25.5.2.22 SVPWM 处理状态寄存器(MCM_IX_SDONE)

SVPWM 处理状态寄存器(MCM_IX_SDONE)																																
偏移地址: 54H																																
复位值: 00000000_00000000_00000000_00000000b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IX_SDONE

—	Bits 31-1	—	—
IX_SDONE	Bit 0	R	SVPWM处理状态 当SVPWM计算完毕, 此位会被硬件置1

25. 5. 2. 23 Clarke 逆变换计算方式选择寄存器(MCM_IX_NORMAL)

Clarke 逆变换计算方式选择寄存器(MCM_IX_NORMAL)																																
偏移地址: 5C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IX_NORMAL

—	Bits 31-1	—	—
IX_NORMAL	Bit 0	R/W	Clarke逆变换计算方式选择 0: Clarke逆变换计算方式为: $V_a = V_{beta}$. 1: Clarke逆变换计算方式为: $V_a = V_{alpha}$

25. 5. 2. 24 Clarke/Park 变换更新寄存器(MCM_X_UPD)

Clarke/Park 变换更新寄存器(MCM_X_UPD)																																
偏移地址: 60 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	X_UPD

—	Bits 31-1	—	—
X_UPD	Bit 0	W1	Clarke/Park变换更新位 写1以更新Clarke/Park变换结果。此位被硬件自动清零

25. 5. 2. 25 Clarke 变换输入 Ia 值寄存器(MCM_X_IA)

Clarke 变换输入 Ia 值寄存器(MCM_X_IA)																															
偏移地址: 64 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				X_IA<11:0>											

—	Bits 31-12	—	—
X_IA	Bits 11-0	R/W	Clarke变换中的Ia值 12位有号整数

25. 5. 2. 26 Clarke 变换输入 Ib 值寄存器(MCM_X_IB)

Clarke 变换输入 Ib 值寄存器(MCM_X_IB)																															
偏移地址: 68 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				X_IB<11:0>											

—	Bits 31-12	—	—
X_IB	Bits 11-0	R/W	Clarke变换中的Ib值 12位有号整数

25.5.2.27 Clarke 变换输入 Ic 值寄存器(MCM_X_IC)

Clarke 变换输入 Ic 值寄存器(MCM_X_IC)																																				
偏移地址: 6C _H																																				
复位值: 00000000_00000000_00000000_00000000 _B																																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
																					X_IC<11:0>															

—	Bits 31-12	—	—
X_IC	Bits 11-0	R/W	Clarke变换中的Ic值 12位有号整数

25.5.2.28 Clarke 变换输出 Ia 值寄存器(MCM_X_IALPHA)

Clarke 变换输出 Ia 值寄存器(MCM_X_IALPHA)																																				
偏移地址: 70 _H																																				
复位值: 00000000_00000000_00000000_00000000 _B																																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
																					X_IALPHA<11:0>															

—	Bits 31-12	—	—
X_IALPHA	Bits 11-0	R	Clarke变换中的Ia值 12位有号整数

25.5.2.29 Clarke 变换输出 β 值寄存器(MCM_X_IBETA)

Clarke 变换输出 β 值寄存器(MCM_X_IBETA)																															
偏移地址: 74H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				X_IBETA<11:0>											

—	Bits 31-12	—	—
X_IBETA	Bits 11-0	R	Clarke变换中的β值 12位有号整数

25.5.2.30 Park 变换输出 Id 值寄存器(MCM_X_ID)

Park 变换输出 Id 值寄存器(MCM_X_ID)																															
偏移地址: 78H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				X_ID<11:0>											

—	Bits 31-12	—	—
X_ID	Bits 11-0	R	Park变换中的Id值 12位有号整数

25.5.2.31 Park 变换输出 Iq 值寄存器(MCM_X_IQ)

Park 变换输出 Iq 值寄存器(MCM_X_IQ)																																
偏移地址: 7C _H																																
复位值: 00000000_00000000_00000000_00000000 _b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																				X_IQ<11:0>												

—	Bits 31-12	—	—
X_IQ	Bits 11-0	R	Park变换中的Iq值 12位有号整数

25.5.2.32 变换处理状态寄存器(MCM_X_DONE)

变换处理状态寄存器(MCM_X_DONE)																															
偏移地址: 80 _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																														X_DONE	

—	Bits 31-1	—	—
X_DONE	Bit 0	R	Clarke/Park变换处理状态 当Clarke/Park变换计算完毕，此位会被硬件置1

25.5.2.33 Clarke 变换输入来源选择寄存器(MCM_X_CON)

Clarke 变换输入来源选择寄存器(MCM_X_CON)																																	
偏移地址: 84 _H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																															X_CON		

—	Bits 31-1	—	—
X_CON	Bit 0	R/W	Clarke 变换输入来源选择. 0: Clarke 变换输入为 X_IA、X_IB、X_IC 寄存器 1: Clarke 变换输入为 LPFA_DO、LPFB_DO、LPFC_DO 寄存器

25.5.2.34 PIO 控制器更新寄存器(MCM_PIO_UPD)

PIO 控制器更新寄存器(MCM_PIO_UPD)																																	
偏移地址: 8C _H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																															PIO_UPD		

—	Bits 31-1	—	—
PIO_UPD	Bit 0	W1	PIO 控制器更新位 写1以更新PIO控制器输出，此位被硬件自动清零

25.5.2.37 PI0 控制器误差上限寄存器(MCM_PI0_ELIMIT)

PI0 控制器误差上限寄存器(MCM_PI0_ELIMIT)																															
偏移地址: 98 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																					PI0_ELIMIT<11:0>										

—	Bits 31-12	—	—
PI0_ELIMIT	Bits 11-0	R/W	PI0控制器误差上限 控制器的参考值与实际值之间的绝对误差上限

25.5.2.38 PI0 控制器 Kp 参数寄存器(MCM_PI0_KP)

PI0 控制器 Kp 参数寄存器(MCM_PI0_KP)																															
偏移地址: 9C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																					PI0_KP<11:0>										

—	Bits 31-12	—	—
PI0_KP	Bits 11-0	R/W	PI0 控制器 Kp 参数 12 位无号小数。举例来说： PI0_KP = 512，其代表的数值为： $512/2^{12} = 0.125$

25.5.2.41 PI0 控制器实际值寄存器(MCM_PI0_ACTUAL)

PI0 控制器实际值寄存器(MCM_PI0_ACTUAL)																															
偏移地址: A8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PI0_ACTUAL<15:0>															

—	Bits 31-16	—	—
PI0_ACTUAL	Bits 15-0	R/W	PI0控制器实际值

25.5.2.42 PI0 控制器输出下限寄存器(MCM_PI0_MIN)

PI0 控制器输出下限寄存器(MCM_PI0_MIN)																															
偏移地址: AC _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PI0_MIN<31:0>															

PI0_MIN	Bits 31-0	R/W	PI0控制器输出下限
---------	-----------	-----	------------

25.5.2.43 PI0 控制器输出上限寄存器(MCM_PI0_MAX)

PI0 控制器输出上限寄存器(MCM_PI0_MAX)																															
偏移地址: B0 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PI0_MAX<31:0>															

PI0_MAX	Bits 31-0	R/W	PI0控制器输出上限
---------	-----------	-----	------------

25.5.2.44 PI0 控制器输出寄存器(MCM_PI0_DO)

PI0 控制器输出寄存器(MCM_PI0_DO)																															
偏移地址: B4 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PI0_DO<31:0>																															

PI0_DO	Bits 31-0	R	PI0控制器输出
--------	-----------	---	----------

25.5.2.45 PI0 控制器输出比例寄存器(MCM_PI0_OSCALE)

PI0 控制器输出比例寄存器(MCM_PI0_OSCALE)																															
偏移地址: B8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															PI0_OSCALE<3:0>

—	Bits 31-4	—	—
PI0_OSCALE	Bits 3-0	R/W	PI0 控制器输出比例 控制器输出: 计算结果 << PI0_OSCALE

25.5.2.46 PI 控制器处理状态寄存器(MCM_PI_DONE)

PI 控制器处理状态寄存器(MCM_PI_DONE)																																
偏移地址: BC _H																																
复位值: 00000000_00000000_00000000_00000001 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	PI_DONE

—	Bits 31-1	—	—
PI_DONE	Bit 0	R	PI 控制器处理状态 三个控制器共享同一个状态位。 0: PI控制器计算中 1: PI控制器计算完成

25.5.2.47 PI1 控制器更新寄存器(MCM_PI1_UPD)

PI1 控制器更新寄存器(MCM_PI1_UPD)																																
偏移地址: C0 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	PI1_UPD

—	Bits 31-1	—	—
PI1_UPD	Bit 0	W1	PI1 控制器更新位 写1以更新PI1控制器输出，此位被硬件自动清零

25.5.2.48 PI1 控制器初始化设定寄存器(MCM_PI1_INI)

PI1 控制器初始化设定寄存器(MCM_PI1_INI)																																
偏移地址: C4 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

—	Bits 31-1	—	—
PI1_INI	Bit 0	W1	PI1 控制器初始化设定位 写1将控制器初始化并将初始值加载内部累加器中，此位被硬件自动清零

25.5.2.49 PI1 控制器初始值寄存器(MCM_PI1_IDATA)

PI1 控制器初始值寄存器(MCM_PI1_IDATA)																															
偏移地址: C8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PI1_IDATA<31:0>																															

PI1_IDATA	Bits 31-0	R/W	PI1控制器初始值 控制器初始化时加载累加器中的初始值
-----------	-----------	-----	---------------------------------------

25.5.2.50 PI1 控制器误差上限寄存器(MCM_PI1_ELIMIT)

PI1 控制器误差上限寄存器(MCM_PI1_ELIMIT)																															
偏移地址: CC _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												PI1_ELIMIT<11:0>																			

—	Bits 31-12	—	—
PI1_ELIMIT	Bits 11-0	R/W	PI1控制器误差上限 控制器的参考值与实际值之间的绝对误差上限

25.5.2.51 PI1 控制器 Kp 参数寄存器(MCM_PI1_KP)

PI1 控制器 Kp 参数寄存器(MCM_PI1_KP)																															
偏移地址: D0 _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												PI1_KP<11:0>																			

—	Bits 31-12	—	—
PI1_KP	Bits 11-0	R/W	PI1 控制器 Kp 参数 12 位无号小数。举例来说： PI1_KP = 512，其代表的数值为： $512/2^{12} = 0.125$

25. 5. 2. 52 PI1 控制器 Ki 参数寄存器(MCM_PI1_KI)

PI1 控制器 Ki 参数寄存器(MCM_PI1_KI)																															
偏移地址: D4 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				PI1_KI<11:0>											

—	Bits 31-12	—	—
PI1_KI	Bits 11-0	R/W	PI1控制器Ki参数 12位无号小数。计算方式同PI1_KP

25. 5. 2. 53 PI1 控制器参考值寄存器(MCM_PI1_REF)

PI1 控制器参考值寄存器(MCM_PI1_REF)																															
偏移地址: D8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				PI1_REF<15:0>											

—	Bits 31-16	—	—
PI1_REF	Bits 15-0	R/W	PI1控制器参考值

25.5.2.54 PI1 控制器实际值寄存器(MCM_PI1_ACTUAL)

PI1 控制器实际值寄存器(MCM_PI1_ACTUAL)																															
偏移地址: DC _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PI1_ACTUAL<15:0>															

—	Bits 31-16	—	—
PI1_ACTUAL	Bits 15-0	R/W	PI1控制器实际值

25.5.2.55 PI1 输出下限寄存器(MCM_PI1_MIN)

PI1 输出下限寄存器(MCM_PI1_MIN)																															
偏移地址: E0 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PI1_MIN<31:0>															

PI1_MIN	Bits 31-0	R/W	PI1控制器输出下限
---------	-----------	-----	------------

25.5.2.56 PI1 输出上限寄存器(MCM_PI1_MAX)

PI1 输出上限寄存器(MCM_PI1_MAX)																															
偏移地址: E4 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PI1_MAX<31:0>																															

PI1_MAX	Bits 31-0	R/W	PI1控制器输出上限
---------	-----------	-----	------------

25.5.2.57 PI1 控制器输出寄存器(MCM_PI1_DO)

PI1 控制器输出寄存器(MCM_PI1_DO)																															
偏移地址: E8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PI1_DO<31:0>																															

PI1_DO	Bits 31-0	R	PI1控制器输出
--------	-----------	---	----------

25.5.2.60 PI2 控制器初始化设定寄存器(MCM_PI2_INI)

PI2 控制器初始化设定寄存器(MCM_PI2_INI)																																
偏移地址: F4 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

—	Bits 31-1	—	—
PI2_INI	Bit 0	W1	PI2 控制器初始化设定位 写1将控制器初始化并将初始值加载内部累加器中，此位被硬件自动清零

25.5.2.61 PI2 控制器初始值寄存器(MCM_PI2_IDATA)

PI2 控制器初始值寄存器(MCM_PI2_IDATA)																															
偏移地址: F8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PI2_IDATA<31:0>																															

PI2_IDATA	Bits 31-0	R/W	PI2控制器初始值 控制器初始化时加载累加器中的初始值

25.5.2.62 PI2 控制器误差上限寄存器(MCM_PI2_ELIMIT)

PI2 控制器误差上限寄存器(MCM_PI2_ELIMIT)																															
偏移地址: FC _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											PI2_ELIMIT<11:0>																				

—	Bits 31-12	—	—
PI2_ELIMIT	Bits 11-0	R/W	PI2控制器误差上限 控制器的参考值与实际值之间的绝对误差上限

25.5.2.63 PI2 控制器 Kp 参数寄存器(MCM_PI2_KP)

PI2 控制器 Kp 参数寄存器(MCM_PI2_KP)																															
偏移地址: 100 _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											PI2_KP<11:0>																				

—	Bits 31-12	—	—
PI2_KP	Bits 11-0	R/W	PI2 控制器 Kp 参数 12 位无号小数。举例来说： PI2_KP = 512，其代表的数值为： $512/2^{12} = 0.125$

25. 5. 2. 64 PI2 控制器 Ki 参数寄存器(MCM_PI2_KI)

PI2 控制器 Ki 参数寄存器(MCM_PI2_KI)																															
偏移地址: 104 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											PI2_KI<11:0>																				

—	Bits 31-12	—	—
PI2_KI	Bits 11-0	R/W	PI2控制器Ki参数 12位无号小数。计算方式同PI2_KP

25. 5. 2. 65 PI2 控制器参考值寄存器(MCM_PI2_REF)

PI2 控制器参考值寄存器(MCM_PI2_REF)																															
偏移地址: 108 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PI2_REF<15:0>															

—	Bits 31-16	—	—
PI2_REF	Bits 15-0	R/W	PI2控制器参考值

25.5.2.66 PI2 控制器实际值寄存器(MCM_PI2_ACTUAL)

PI2 控制器实际值寄存器(MCM_PI2_ACTUAL)																															
偏移地址: 10C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PI2_ACTUAL<15:0>															

—	Bits 31-16	—	—
PI2_ACTUAL	Bits 15-0	R/W	PI2控制器实际值

25.5.2.67 PI2 输出下限寄存器(MCM_PI2_MIN)

PI2 输出下限寄存器(MCM_PI2_MIN)																															
偏移地址: 110 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PI2_MIN<31:0>															

PI2_MIN	Bits 31-0	R/W	PI2控制器输出下限
---------	-----------	-----	------------

25.5.2.68 PI2 输出上限寄存器(MCM_PI2_MAX)

PI2 输出上限寄存器(MCM_PI2_MAX)																															
偏移地址: 114 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PI2_MAX<31:0>																															

PI2_MAX	Bits 31-0	R/W	PI2控制器输出上限
---------	-----------	-----	------------

25.5.2.69 PI2 控制器输出寄存器(MCM_PI2_DO)

PI2 控制器输出寄存器(MCM_PI2_DO)																															
偏移地址: 118 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PI2_DO<31:0>																															

PI2_DO	Bits 31-0	R	PI2控制器输出
--------	-----------	---	----------

25.5.2.72 CORDIC 函数 X 分量输入寄存器(MCM_COR_X)

CORDIC 函数 X 分量输入寄存器(MCM_COR_X)																															
偏移地址: 124 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																COR_X<15:0>															

—	Bits 31-16	—	—
COR_X	Bits 15-0	R/W	CORDIC 函数 X 分量输入寄存器 16位有号整数

25.5.2.73 CORDIC 函数 Y 分量输入寄存器(MCM_COR_Y)

CORDIC 函数 Y 分量输入寄存器(MCM_COR_Y)																															
偏移地址: 128 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																COR_Y<15:0>															

—	Bits 31-16	—	—
COR_Y	Bits 15-0	R/W	CORDIC 函数 Y 分量输入寄存器 16位有号整数

25.5.2.74 CORDIC 计算结果寄存器(MCM_COR_RES)

CORDIC 计算结果寄存器(MCM_COR_RES)																																
偏移地址: 12C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																						COR_RES<9:0>										

—	Bits 31-10	—	—
COR_RES	Bits 9-0	R	CORDIC 计算结果 10 位无号定点数。输出与角度关系为： 角度 = (COR_RES * 360)/1024

25.5.2.75 CORDIC 处理状态寄存器(MCM_COR_DONE)

CORDIC 处理状态寄存器(MCM_COR_DONE)																															
偏移地址: 130 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															COR_DONE

—	Bits 31-1	—	—
COR_DONE	Bit 0	R	CORDIC处理状态 当CORDIC计算完毕，此位会被硬件置1

25.5.2.76 CORDIC 输入来源选择寄存器(MCM_COR_CON)

CORDIC 输入来源选择寄存器(MCM_COR_CON)																																
偏移地址: 134 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

—	Bits 31-1	—	—
COR_CON	Bit 0	R/W	CORDIC 输入来源选择 0: CORDIC输入为COR_X、COR_Y寄存器 1: CORDIC输入为SMC的输出SMC_BEMF2A、SMC_BEMF2B

25.5.2.77 LPF A 初始化设定寄存器(MCM_LPFA_INI)

LPF A 初始化设定寄存器(MCM_LPFA_INI)																																
偏移地址: 140 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

—	Bits 31-1	—	—
LPFA_INI	Bit 0	W1	LPF A初始化设定位 写1以初始化LPF，此位被硬件自动清零

25.5.2.78 LPF A 更新控制寄存器(MCM_LPFA_UPD)

LPF A 更新控制寄存器(MCM_LPFA_UPD)																															
偏移地址: 144 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															LPFA_UPD

—	Bits 31-1	—	—
LPFA_UPD	Bit 0	W1	LPF A更新控制位 写1以更新LPF A输出, 此位被硬件自动清零

25.5.2.79 LPF A 输入讯号寄存器(MCM_LPFA_DIN)

LPF A 输入讯号寄存器(MCM_LPFA_DIN)																															
偏移地址: 148 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFA_DIN<11:0>																			

—	Bits 31-12	—	—
LPFA_DIN	Bits 11-0	R/W	LPF A 输入讯号 12位有号整数

25.5.2.80 LPF A a0 系数寄存器(MCM_LPFA_A0)

LPF A a0 系数寄存器(MCM_LPFA_A0)																															
偏移地址: 14C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFA_A0<11:0>																			

—	Bits 31-12	—	—
LPFA_A0	Bits 11-0	R/W	LPF A a0 系数 12 位有号小数。 举例来说, 若 LPF _x _A0 = 0xC00, 则与实际系数对应关系为: $0xC00 / 2^{11} = -1024 / 2048 = -0.5$

25.5.2.81 LPF A b0 系数寄存器(MCM_LPFA_B0)

LPF A b0 系数寄存器(MCM_LPFA_B0)																															
偏移地址: 150 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFA_B0<11:0>																			

—	Bits 31-12	—	—
LPFA_B0	Bits 11-0	R/W	LPF A b0 系数 同LPFA_A0

25.5.2.82 LPF A b1 系数寄存器(MCM_LPFA_B1)

LPF A b1 系数寄存器(MCM_LPFA_B1)																															
偏移地址: 154 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												LPFA_B1<11:0>			

—	Bits 31-12	—	—
LPFA_B1	Bits 11-0	R/W	LPF A b1系数 同LPFA_A0

25.5.2.83 LPF A a0 系数输出比例寄存器(MCM_LPFA_A0SCALE)

LPF A a0 系数输出比例寄存器(MCM_LPFA_A0SCALE)																															
偏移地址: 158 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												LPFA_A0SCALE<3:0>			

—	Bits 31-4	—	—
LPFA_A0SCALE	Bits 3-0	R/W	LPF A a0 系数输出比例

25.5.2.84 LPF A b0 系数输出比例寄存器(MCM_LPFA_B0SCALE)

LPF A b0 系数输出比例寄存器(MCM_LPFA_B0SCALE)																																
偏移地址: 15C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																															LPFA_B0SCALE<3:0>	

—	Bits 31-4	—	—
LPFA_B0SCALE	Bits 3-0	R/W	LPF A b0 系数输出比例

25.5.2.85 LPF A b1 系数输出比例寄存器(MCM_LPFA_B1SCALE)

LPF A b1 系数输出比例寄存器(MCM_LPFA_B1SCALE)																																
偏移地址: 160 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																															LPFA_B1SCALE<3:0>	

—	Bits 31-4	—	—
LPFA_B1SCALE	Bits 3-0	R/W	LPF A b1 系数输出比例

25.5.2.86 LPF A 输出结果寄存器(MCM_LPFA_DO)

LPF A 输出结果寄存器(MCM_LPFA_DO)																															
偏移地址: 164 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								LPFA_DO<11:0>							

—	Bits 31-12	—	—
LPFA_DO	Bits 11-0	R	LPF A 输出结果 12位有号整数

25.5.2.87 LPF A 处理状态寄存器(MCM_LPFA_DONE)

LPF A 处理状态寄存器(MCM_LPFA_DONE)																															
偏移地址: 168 _H																															
复位值: 00000000_00000000_00000000_00000001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															LPFA_DONE

—	Bits 31-1	—	—
LPFA_DONE	Bit 0	R	LPF A 处理状态 LPF A计算完毕, 此位会被硬件置1

25.5.2.88 LPF B 初始化设定寄存器(MCM_LPFB_INI)

LPF B 初始化设定寄存器(MCM_LPFB_INI)																																
偏移地址: 16C _H																																
复位值: 00000000_00000000_00000000_00000000 _b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LPFB_INI

—	Bits 31-1	—	—
LPFB_INI	Bit 0	W1	LPF B 初始化设定位 写1以初始化LPF，此位被硬件自动清零

25.5.2.89 LPF B 更新控制寄存器(MCM_LPFB_UPD)

LPF B 更新控制寄存器(MCM_LPFB_UPD)																																
偏移地址: 170 _H																																
复位值: 00000000_00000000_00000000_00000000 _b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LPFB_UPD

—	Bits 31-1	—	—
LPFB_UPD	Bit 0	W1	LPF B 更新控制位 写1以更新LPF B输出，此位被硬件自动清零

25.5.2.90 LPF B 输入讯号寄存器(MCM_LPFB_DIN)

LPF B 输入讯号寄存器(MCM_LPFB_DIN)																															
偏移地址: 174 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFB_DIN<11:0>																			

—	Bits 31-12	—	—
LPFB_DIN	Bits 11-0	R/W	LPF B 输入讯号 12位有号整数

25.5.2.91 LPF B a0 系数寄存器(MCM_LPFB_A0)

LPF B a0 系数寄存器(MCM_LPFB_A0)																															
偏移地址: 178 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFB_A0<11:0>																			

—	Bits 31-12	—	—
LPFB_A0	Bits 11-0	R/W	LPF B a0 系数 12位有号小数。 举例来说, 若 LPFB_A0 = 0xC00, 则与实际系数 对应关系为: $0xC00 / 2^{11} = -1024 / 2048 = -0.5$

25.5.2.94 LPF B a0 系数输出比例寄存器(MCM_LPFB_A0SCALE)

LPF B a0 系数输出比例寄存器(MCM_LPFB_A0SCALE)																															
偏移地址: 184 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															LPFB_A0SCALE<3:0>

—	Bits 31-4	—	—
LPFB_A0SCALE	Bits 3-0	R/W	LPF B a0 系数输出比例

25.5.2.95 LPF B b0 系数输出比例寄存器(MCM_LPFB_B0SCALE)

LPF B b0 系数输出比例寄存器(MCM_LPFB_B0SCALE)																															
偏移地址: 188 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															LPFB_B0SCALE<3:0>

—	Bits 31-4	—	—
LPFB_B0SCALE	Bits 3-0	R/W	LPF B b0 系数输出比例

25.5.2.96 LPF B b1 系数输出比例寄存器(MCM_LPFB_B1SCALE)

LPF B b1 系数输出比例寄存器(MCM_LPFB_B1SCALE)																																
偏移地址: 18C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																															LPFB_B1SCALE<3:0>	

—	Bits 31-4	—	—
LPFB_B1SCALE	Bits 3-0	R/W	LPF B b1 系数输出比例

25.5.2.97 LPF B 输出结果寄存器(MCM_LPFB_DO)

LPF B 输出结果寄存器(MCM_LPFB_DO)																																
偏移地址: 190 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
												LPFB_DO<11:0>																				

—	Bits 31-12	—	—
LPFB_DO	Bits 11-0	R	LPF B输出结果 12位有号整数

25.5.2.98 LPF B 处理状态寄存器(MCM_LPFB_DONE)

LPF B 处理状态寄存器(MCM_LPFB_DONE)																															
偏移地址: 194 _H																															
复位值: 00000000_00000000_00000000_00000001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															LPFB_DONE

—	Bits 31-1	—	—
LPFB_DONE	Bit 0	R	LPF B 处理状态 LPF B计算完毕, 此位会被硬件置1

25.5.2.99 LPF C 初始化设定寄存器(MCM_LPFC_INI)

LPF C 初始化设定寄存器(MCM_LPFC_INI)																															
偏移地址: 198 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															LPFC_INI

—	Bits 31-1	—	—
LPFC_INI	Bit 0	W1	LPF C 初始化设定位 写1以初始化LPF, 此位被硬件自动清零

25.5.2.100 LPF C 更新控制寄存器(MCM_LPFC_UPD)

LPF C 更新控制寄存器(MCM_LPFC_UPD)																															
偏移地址: 19C _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															LPFC_UPD

—	Bits 31-1	—	—
LPFC_UPD	Bit 0	W1	LPF C 更新控制位 写1以更新LPF C输出, 此位被硬件自动清零

25.5.2.101 LPF C 输入讯号寄存器(MCM_LPFC_DIN)

LPF C 输入讯号寄存器(MCM_LPFC_DIN)																															
偏移地址: 1A0 _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFC_DIN<11:0>																			

—	Bits 31-12	—	—
LPFC_DIN	Bits 11-0	R/W	LPF C输入讯号 12位有号整数

25.5.2.102 LPF C a0 系数寄存器(MCM_LPFC_A0)

LPF C a0 系数寄存器(MCM_LPFC_A0)																															
偏移地址: 1A4 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFC_A0<11:0>																			

—	Bits 31-12	—	—
LPFC_A0	Bits 11-0	R/W	LPF C a0 系数 12 位有号小数。 举例来说, 若 LPFC_A0 = 0xC00, 则与实际系数对应关系为: $0xC00 / 2^{11} = -1024 / 2048 = -0.5$

25.5.2.103 LPF C b0 系数寄存器(MCM_LPFC_B0)

LPF C b0 系数寄存器(MCM_LPFC_B0)																															
偏移地址: 1A8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFC_B0<11:0>																			

—	Bits 31-12	—	—
LPFC_B0	Bits 11-0	R/W	LPF C b0系数 同LPFC_A0

25.5.2.104 LPF C b1 系数寄存器(MCM_LPFC_B1)

LPF C b1 系数寄存器(MCM_LPFC_B1)																															
偏移地址: 1AC _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											LPFC_B1<11:0>																				

—	Bits 31-12	—	—
LPFC_B1	Bits 11-0	R/W	LPF C b1系数 同LPFC_A0

25.5.2.105 LPF C a0 系数输出比例寄存器(MCM_LPFC_A0SCALE)

LPF C a0 系数输出比例寄存器(MCM_LPFC_A0SCALE)																															
偏移地址: 1B0 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												LPFC_A0SCALE<3:0>			

—	Bits 31-4	—	—
LPFC_A0SCALE	Bits 3-0	R/W	LPF C a0 系数输出比例

25.5.2.108 LPF C 输出结果寄存器(MCM_LPFC_DO)

LPF C 输出结果寄存器(MCM_LPFC_DO)																															
偏移地址: 1BC _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												LPFC_DO<11:0>																			

—	Bits 31-12	—	—
LPFC_DO	Bits 11-0	R	LPF C 输出结果 12位有号整数

25.5.2.109 LPF C 处理状态寄存器(MCM_LPFC_DONE)

LPF C 处理状态寄存器(MCM_LPFC_DONE)																															
偏移地址: 1C0 _H																															
复位值: 00000000_00000000_00000000_00000001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															LPFC_DONE

—	Bits 31-1	—	—
LPFC_DONE	Bit 0	R	LPF C 处理状态 LPF C 计算完毕, 此位会被硬件置 1

25.5.2.110 Global 更新寄存器(MCM_GL_UPD)

Global 更新寄存器(MCM_GL_UPD)																																
偏移地址: 1C4 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																					SVM	VM	SMC	COR	LPFC	LPFB	LPFA	PI2	PI1	PI0	IX	X

—	Bits 31-12	—	—
SVM	Bit 11	W1	SVPWM更新控制位
VM	Bit 10	W1	VMODEL 更新控制位
SMC	Bit 9	W1	SMC 更新控制位
COR	Bit 8	W1	CORDIC 启动控制位
LPFC	Bit 7	W1	LPF C 更新控制位
LPFB	Bit 6	W1	LPF B 更新控制位
LPFA	Bit 5	W1	LPF A 更新控制位
PI2	Bit 4	W1	PI2 控制器更新控制位
PI1	Bit 3	W1	PI1 控制器更新控制位
PI0	Bit 2	W1	PI0 控制器更新控制位
IX	Bit 1	W1	Clarke/park 逆变换更新控制位
X	Bit 0	W1	Clarke/park 变换更新控制位

25. 5. 2. 111 Global 状态寄存器(MCM_GL_STAT)

Global 状态寄存器(MCM_GL_STAT)																																
偏移地址: 1C8 _H																																
复位值: 00000000_00000000_00000011_11101000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																							VM	SMC	LPFC	LPFB	LPFA	COR	PI	X	SVM	IX

—	Bits 31-10	—	—
VM	Bit 9	R	VMODEL 处理状态, 同 VMODEL_DONE
SMC	Bit 8	R	SMC 处理状态, 同 SMC_DONE
LPFC	Bit 7	R	LPFC 处理状态, 同 LPFC_DONE
LPFB	Bit 6	R	LPFB 处理状态, 同 LPFB_DONE
LPFA	Bit 5	R	LPFA 处理状态, 同 LPFA_DONE
COR	Bit 4	R	CORDIC 处理状态, 同 CORDIC_DONE
PI	Bit 3	R	PI 控制器处理状态, 同 PI_DONE
X	Bit 2	R	Clarke/Park 变换处理状态, 同 X_DONE
SVM	Bit 1	R	SVPWM 处理状态, 同 SVM_DONE
IX	Bit 0	R	Clarke/Park 逆变换处理状态, 同 IX_DONE

25.5.2.112 正弦数值寄存器(MCM_THETA_SIN)

正弦数值寄存器(MCM_THETA_SIN)																															
偏移地址: 1CC _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								THETA_SIN<7:0 ^Δ							

—	Bits 31-8	—	—
THETA_SIN	Bits 7-0	R/W	正弦数值 用于 Park 变换的正弦值，8 位有号小数。 举例来说，若 THETA_SIN = 0xC0，则： $\text{Sin}(\theta) = 0xC0 / 2^7 = -64 / 128 = -0.5$

25.5.2.113 余弦数值寄存器(MCM_THETA_COS)

余弦数值寄存器(MCM_THETA_COS)																															
偏移地址: 1D0 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								THETA_COS<7:0 ^Δ							

—	Bits 31-8	—	—
THETA_COS	Bits 7-0	R/W	余弦数值 用于 Park 变换的余弦值，8 位有号小数。 计算方式同THETA_SIN

25.5.2.114 PI 控制器类型设定寄存器(MCM_PI_TYPE)

PI 控制器类型设定寄存器(MCM_PI_TYPE)																																
偏移地址: 1D4 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																															PI_TYPE<1:0>	

—	Bits 31-2	—	—
PI_TYPE	Bits 1-0	R/W	PI 控制器类型设定 0:电压控制(Voltage Control) 1:速度控制(Speed Control)

25.5.2.115 电机旋转方向选择寄存器(MCM_MOTOR_DIR)

电机旋转方向选择寄存器(MCM_MOTOR_DIR)																															
偏移地址: 1D8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															MOTOR_DIR

—	Bits 31-1	—	—
MOTOR_DIR	Bit 0	R/W	电机旋转方向选择 0:正向旋转 1:反向旋转

25.5.2.116 MCM 版本号寄存器(MCM_VER)

MCM 版本号寄存器(MCM_VER)																															
偏移地址: 1DC _H																															
复位值: 01001101_01000011_01001101_00000001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCM_VER<31:0>																															

MCM_VER	Bits 31-0	R	MCM 版本号
---------	-----------	---	---------

25.5.2.117 SMC 更新控制寄存器(MCM_SMC_UPD)

SMC 更新控制寄存器(MCM_SMC_UPD)																															
偏移地址: 1E0 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															SMC_UPD

—	Bits 31-1	—	—
SMC_UPD	Bit 0	W1	SMC 更新控制位 写 1 用以更新 SMC 计算结果。此位被硬件自动清零

25.5.2.118 SMC 初始化设定寄存器(MCM_SMC_INI)

SMC 初始化设定寄存器(MCM_SMC_INI)																															
偏移地址: 1E4 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															SMC_INI

—	Bits 31-1	—	—
SMC_INI	Bit 0	W1	SMC 初始化设定位 写1将控制器初始化并将初始值加载控制器中, 此位被硬件自动清零

25.5.2.119 SMC Alpha 分量初始值寄存器(MCM_SMC_ALPHAINI)

SMC Alpha 分量初始值寄存器(MCM_SMC_ALPHAINI)																																													
偏移地址: 1E8 _H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																SMC_ALPHAINI<15:0>																													

—	Bits 31-16	—	—
SMC_ALPHAINI	Bits 15-0	R/W	SMC α 分量初始值 当 SMC 初始化时, 将此寄存器中的数值加载 SMC_ZALPHA 、 SMC_BEMF1A 、 SMC_BEMF2A 当中。 16位有号整数。

25.5.2.120 SMC Beta 分量初始值寄存器(MCM_SMC_BETAINI)

SMC Beta 分量初始值寄存器(MCM_SMC_BETAINI)																															
偏移地址: 1EC _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_BETAINI<15:0>															

—	Bits 31-16	—	—
SMC_BETAINI	Bits 15-0	R/W	<p>SMC β 分量初始值 当 SMC 初始化时, 将此寄存器中的数值加载 SMC_ZBETA、SMC_BEMF1B、SMC_BEMF2B 当中。 16位有号整数。</p>

25.5.2.121 SMC 动态 LPF 系数寄存器(MCM_SMC_A0)

SMC 动态 LPF 系数寄存器(MCM_SMC_A0)																															
偏移地址: 1F0 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SMC_A0<11:0>																			

—	Bits 31-12	—	—
SMC_A0	Bits 11-0	R/W	<p>SMC 动态 LPF 系数 用以滤除因子 Z 与估算 BEMF 的低通滤波器动态系数。12 位无号小数。 举例来说, 若 SMC_A0 = 0x100, 则动态系数实际值为: $0x100 / 2^{12} = 0.0625$</p>

25.5.2.122 SMC 误差上限寄存器(MCM_SMC_ELIMIT)

SMC 误差上限寄存器(MCM_SMC_ELIMIT)																															
偏移地址: 1F4 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															SMC_ELIMIT<3:0>

—	Bits 31-4	—	—
SMC_ELIMIT	Bits 3-0	R/W	SMC误差上限 最大误差= $2^{\text{SMC_ELIMIT}}$

25.5.2.123 SMC Z 因子放大倍率寄存器(MCM_SMC_KSLIDE)

SMC Z 因子放大倍率寄存器(MCM_SMC_KSLIDE)																															
偏移地址: 1F8 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_KSLIDE<15:0>															

—	Bits 31-16	—	—
SMC_KSLIDE	Bits 15-0	R/W	SMC Z因子放大倍率

25.5.2.124 SMC 处理状态寄存器(MCM_SMC_DONE)

SMC 处理状态寄存器(MCM_SMC_DONE)																																
偏移地址: 1FC _H																																
复位值: 00000000_00000000_00000000_00000001 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																																SMC_DONE

—	Bits 31-1	—	—
SMC_DONE	Bit 0	R	SMC 处理状态 当SMC计算完毕, 此位会被硬件置1

25.5.2.125 SMC 估测电流 Alpha 分量寄存器(MCM_SMC_IEALPHA)

SMC 估测电流 Alpha 分量寄存器(MCM_SMC_IEALPHA)																															
偏移地址: 200 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_IEALPHA<15:0>															

—	Bits 31-16	—	—
SMC_IEALPHA	Bits 15-0	R/W	SMC估测电流α分量. 16位有号整数

25.5.2.126 SMC 估测电流 Beta 分量寄存器(MCM_SMC_IEBETA)

SMC 估测电流 Beta 分量寄存器(MCM_SMC_IEBETA)																															
偏移地址: 204 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_IEBETA<15:0>															

—	Bits 31-16	—	—
SMC_IEBETA	Bits 15-0	R/W	SMC估测电流β分量. 16位有号整数

25.5.2.127 SMC 量测电流 Alpha 分量寄存器(MCM_SMC_IALPHA)

SMC 量测电流 Alpha 分量寄存器(MCM_SMC_IALPHA)																															
偏移地址: 208 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_IALPHA<15:0>															

—	Bits 31-16	—	—
SMC_IALPHA	Bits 15-0	R/W	SMC量测电流α分量 16位有号整数

25.5.2.128 SMC 量测电流 Beta 分量寄存器(MCM_SMC_IBETA)

SMC 量测电流 Beta 分量寄存器(MCM_SMC_IBETA)																															
偏移地址: 20C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_IBETA<15:0>															

—	Bits 31-16	—	—
SMC_IBETA	Bits 15-0	R/W	SMC 量测电流 β 分量 16位有号整数

25.5.2.129 SMC 估测电流输入来源选择寄存器(MCM_SMC_DISI)

SMC 估测电流输入来源选择寄存器(MCM_SMC_DISI)																															
偏移地址: 210 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															SMC_DISI

—	Bits 31-1	—	—
SMC_DISI	Bit 0	R/W	SMC 估测电流输入来源选择 0: SMC 估测电流输入为虚拟电机输出 VM_IEALPHA、VM_IEBETA 1: SMC 估测电流输入为 SMC_IEALPHA、SMC_IEBETA寄存器

25.5.2.130 SMC Z 因子 Alpha 分量寄存器(MCM_SMC_ZALPHA)

SMC Z 因子 Alpha 分量寄存器(MCM_SMC_ZALPHA)																															
偏移地址: 214 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_ZALPHA<15:0>															

—	Bits 31-16	—	—
SMC_ZALPHA	Bits 15-0	R	SMC Z 因子 α 分量输出 16位有号整数

25.5.2.131 SMC Z 因子 Beta 分量寄存器(MCM_SMC_ZBETA)

SMC Z 因子 Beta 分量寄存器(MCM_SMC_ZBETA)																															
偏移地址: 218 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_ZBETA<15:0>															

—	Bits 31-16	—	—
SMC_ZBETA	Bits 15-0	R	SMC Z 因子 β 分量输出 16位有号整数

25.5.2.132 SMC 估测 BEMF Alpha 分量寄存器(MCM_SMC_BEMF1A)

SMC 估测 BEMF Alpha 分量寄存器(MCM_SMC_BEMF1A)																															
偏移地址: 21C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_BEMF1A<15:0>															

—	Bits 31-16	—	—
SMC_BEMF1A	Bits 15-0	R	SMC 估测 BEMF α 分量输出 16位有号整数

25.5.2.133 SMC 估测 BEMF Beta 分量寄存器(MCM_SMC_BEMF1B)

SMC 估测 BEMF Beta 分量寄存器(MCM_SMC_BEMF1B)																															
偏移地址: 220 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_BEMF1B<15:0>															

—	Bits 31-16	—	—
SMC_BEMF1B	Bits 15-0	R	SMC 估测 BEMF β 分量输出 16位有号整数

25. 5. 2. 134 SMC 滤波后 BEMF Alpha 分量寄存器(MCM_SMC_BEMF2A)

SMC 滤波后 BEMF Alpha 分量寄存器(MCM_SMC_BEMF2A)																															
偏移地址: 224 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_BEMF2A<15:0>															

—	Bits 31-16	—	—
SMC_BEMF2A	Bits 15-0	R	SMC 滤波后 BEMF α 分量 通过低通滤波器后的估测BEMF，用于计算转子角度。16位有号整数。

25. 5. 2. 135 SMC 滤波后 BEMF Beta 分量寄存器(MCM_SMC_BEMF2B)

SMC 滤波后 BEMF Beta 分量寄存器(MCM_SMC_BEMF2B)																															
偏移地址: 228 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMC_BEMF2B<15:0>															

—	Bits 31-16	—	—
SMC_BEMF2B	Bits 15-0	R	SMC 滤波后 BEMF β 分量 通过低通滤波器后的估测BEMF，用于计算转子角度。16位有号整数。

25.5.2.138 虚拟电机初始化设定寄存器(MCM_VM_INI)

虚拟电机初始化设定寄存器(MCM_VM_INI)																															
偏移地址: 238 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

—	Bits 31-1	—	—
VM_INI	Bit 0	W1	虚拟电机初始化设定位 写1将虚拟电机初始化并将初始值加载控制器中，此位被硬件自动清零

25.5.2.139 虚拟电机 Alpha 分量初值寄存器(MCM_VM_ALPHAINI)

虚拟电机 Alpha 分量初值寄存器(MCM_VM_ALPHAINI)																																													
偏移地址: 23C _H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																VM_ALPHAINI<15:0>																													

—	Bits 31-16	—	—
VM_ALPHAINI	Bits 15-0	R/W	虚拟电机 α 分量初值 当虚拟电机初始化时，将此寄存器中的数值加载至VM_IEALPHA当中。16位有号整数

25.5.2.140 虚拟电机 Beta 分量初值寄存器(MCM_VM_BETAINI)

虚拟电机 Beta 分量初值寄存器(MCM_VM_BETAINI)																															
偏移地址: 240 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VM_BETAINI<15:0>															

—	Bits 31-16	—	—
VM_BETAINI	Bits 15-0	R/W	虚拟电机 β 分量初值 当虚拟电机初始化时，将此寄存器中的数值加载 VM_IEBETA 当中。16位有号整数。

25.5.2.141 虚拟电机 F 系数寄存器(MCM_VM_KF)

虚拟电机 F 系数寄存器(MCM_VM_KF)																															
偏移地址: 244 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VM_KF<11:0>															

—	Bits 31-12	—	—
VM_KF	Bits 11-0	R/W	虚拟电机 F 系数 12 位无号小数。 举例来说，若 VM_KF = 0x100，则动态系数实际值为： $0x100 / 2^{12} = 0.0625$

25.5.2.142 虚拟电机 G 系数寄存器(MCM_VM_KG)

虚拟电机 G 系数寄存器(MCM_VM_KG)																															
偏移地址: 248 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												VM_KG<11:0>																			

—	Bits 31-12	—	—
VM_KG	Bits 11-0	R/W	虚拟电机 G 系数 12 位无号小数。 计算方式同VM_KG。

25.5.2.143 虚拟电机电压 Alpha 分量寄存器(MCM_VM_VALPHA)

虚拟电机电压 Alpha 分量寄存器(MCM_VM_VALPHA)																															
偏移地址: 24C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VM_VALPHA<15:0>															

—	Bits 31-16	—	—
VM_VALPHA	Bits 15-0	R/W	虚拟电机电压α分量 16位有号整数

25.5.2.144 虚拟电机电压 Beta 分量寄存器(MCM_VM_VBETA)

虚拟电机电压 Beta 分量寄存器(MCM_VM_VBETA)																															
偏移地址: 250 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VM_VBETA<15:0>															

—	Bits 31-16	—	—
VM_VBETA	Bits 15-0	R/W	虚拟电机电压β分量 16位有号整数

25.5.2.145 虚拟电机 BEMF Alpha 分量寄存器(MCM_VM_BEMFA)

虚拟电机 BEMF Alpha 分量寄存器(MCM_VM_BEMFA)																															
偏移地址: 254 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VM_BEMFA<15:0>															

—	Bits 31-16	—	—
VM_BEMFA	Bits 15-0	R/W	虚拟电机 BEMF α分量 16位有号整数

25.5.2.146 虚拟电机 BEMF Beta 分量寄存器(MCM_VM_BEMFB)

虚拟电机 BEMF Beta 分量寄存器(MCM_VM_BEMFB)																															
偏移地址: 258 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VM_BEMFB<15:0>															

—	Bits 31-16	—	—
VM_BEMFB	Bits 15-0	R/W	虚拟电机 BEMF β分量 16位有号整数

25.5.2.147 虚拟电机 Z 因子 Alpha 分量寄存器(MCM_VM_ZALPHA)

虚拟电机 Z 因子 Alpha 分量寄存器(MCM_VM_ZALPHA)																															
偏移地址: 25C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VM_ZALPHA<15:0>															

—	Bits 31-16	—	—
VM_ZALPHA	Bits 15-0	R/W	虚拟电机 Z因子α分量 16位有号整数

25. 5. 2. 148 虚拟电机 Z 因子 Beta 分量寄存器(MCM_VM_ZBETA)

虚拟电机 Z 因子 Beta 分量寄存器(MCM_VM_ZBETA)																																															
偏移地址: 260H																																															
复位值: 00000000_00000000_00000000_00000000 _b																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
																VM_ZBETA<15:0>																															

—	Bits 31-16	—	—
VM_ZBETA	Bits 15-0	R/W	虚拟电机 Z因子β分量 16位有号整数

25. 5. 2. 149 虚拟电机 Z 因子输入来源选择寄存器(MCM_VM_DISZ)

虚拟电机 Z 因子输入来源选择寄存器(MCM_VM_DISZ)																																		
偏移地址: 264H																																		
复位值: 00000000_00000000_00000000_00000000 _b																																		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
																																		VM_DISZ

—	Bits 31-1	—	—
VM_DISZ	Bit 0	R/W	虚拟电机 Z 因子与 BEMF 输入来源选择 0: 虚拟电机输入来源为 SMC_ZALPHA、SMC_ZBETA、SMC_BEMF1A 与 SMC_BEMF1B。 1: 虚拟电机输入来源为 VM_ZALPHA、VM_ZBETA、VM_BEMFA与VM_BEMFB。

25.5.2.150 虚拟电机处理状态寄存器(MCM_VM_DONE)

虚拟电机处理状态寄存器(MCM_VM_DONE)																															
偏移地址: 268 _H																															
复位值: 00000000_00000000_00000000_00000001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															VM_DONE

—	Bits 31-1	—	—
VM_DONE	Bit 0	R	虚拟电机处理状态 当虚拟电机计算完毕, 此位会被硬件置1。

25.5.2.151 虚拟电机估测结果 Alpha 寄存器(MCM_VM_IEALPHA)

虚拟电机估测结果 Alpha 寄存器(MCM_VM_IEALPHA)																																												
偏移地址: 26C _H																																												
复位值: 00000000_00000000_00000000_00000000 _B																																												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
																VM_IEALPHA<15:0>																												

—	Bits 31-16	—	—
VM_IEALPHA	Bits 15-0	R	虚拟电机估测电流 α 分量 16位有号整数

25.5.2.152 虚拟电机估测结果 Beta 寄存器(MCM_VM_IEBETA)

虚拟电机估测结果 Beta 寄存器(MCM_VM_IEBETA)																																														
偏移地址: 270 _H																																														
复位值: 00000000_00000000_00000000_00000000 _B																																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																VM_IEBETA<15:0>																														

—	Bits 31-16	—	—
VM_IEBETA	Bits 15-0	R	虚拟电机估测电流β分量 16位有号整数

25.5.2.153 虚拟电机实际电流输入来源选择寄存器(MCM_VM_CON)

虚拟电机实际电流输入来源选择寄存器(MCM_VM_CON)																																	
偏移地址: 274 _H																																	
复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
																																	VM_CON

—	Bits 31-1	—	—
VM_CON	Bit 0	R/W	虚拟电机实际电流输入来源选择 0:实际电流输入为 SMC_IALPHA、SMC_IBETA。 1:实际电流输入为Clarke变换结果X_IALPHA、X_IBETA。

25.5.2.154 虚拟电机 F 系数输出比例寄存器(MCM_VM_KFSCALE)

虚拟电机 F 系数输出比例寄存器(MCM_VM_KFSCALE)																															
偏移地址: 278 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															VM_KFSCALE<3:0>

—	Bits 31-4	—	—
VM_KFSCALE	Bits 3-0	R/W	虚拟电机F系数输出比例

25.5.2.155 虚拟电机 G 系数输出比例寄存器(MCM_VM_KGSCALE)

虚拟电机 G 系数输出比例寄存器(MCM_VM_KGSCALE)																															
偏移地址: 27C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															VM_KGSCALE<3:0>

—	Bits 31-4	—	—
VM_KGSCALE	Bits 3-0	R/W	虚拟电机G系数输出比例

25.5.2.156 SVPWM 更新控制寄存器(MCM_SVM_UPD)

SVPWM 更新控制寄存器(MCM_SVM_UPD)																																
偏移地址: 288 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SVM_UPD

—	Bits 31-1	—	—
SVM_UPD	Bit 0	W1	SVPWM 更新控制位 写1用以更新SVPWM计算结果。此位被硬件自动清零。

25.5.2.157 SVPWM 输入来源选择寄存器(MCM_SVM_CON)

SVPWM 输入来源选择寄存器(MCM_SVM_CON)																																
偏移地址: 28C _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SVM_CON

—	Bits 31-1	—	—
SVM_CON	Bit 0	R/W	SVPWM 输入来源选择 1: SVPWM 输入为 SVM_DINA、SVM_DINB 与 SVM_DINC。 0: SVPWM 输入为 Clarke 逆变换结果 IX_VA、IX_VB 与 IX_VC。

25.5.2.158 SVPWM A 相输入寄存器(MCM_SVM_DINA)

SVPWM A 相输入寄存器(MCM_SVM_DINA)																																													
偏移地址: 290 _H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																SVM_DINA<15:0>																													

—	Bits 31-16	—	—
SVM_DINA	Bits 15-0	R/W	SVPWM A相输入

25.5.2.159 SVPWM B 相输入寄存器(MCM_SVM_DINB)

SVPWM B 相输入寄存器(MCM_SVM_DINB)																																													
偏移地址: 294 _H																																													
复位值: 00000000_00000000_00000000_00000000 _B																																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
																SVM_DINB<15:0>																													

—	Bits 31-16	—	—
SVM_DINB	Bits 15-0	R/W	SVPWM B相输入

25. 5. 2. 160 SVPWM C 相输入寄存器(MCM_SVM_DINC)

SVPWM C 相输入寄存器(MCM_SVM_DINC)																																														
偏移地址: 298 _H																																														
复位值: 00000000_00000000_00000000_00000000 _B																																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																SVM_DINC<15:0>																														

—	Bits 31-16	—	—
SVM_DINC	Bits 15-0	R/W	SVPWM C相输入

版本历史

版本	修改日期	更改概要
V1.0	2020-04-30	初版
V1.1	2020-05-29	1. 修改 SYSCFG 及 RCU 章节顺序 2. Flash、SYSCFG、RCU 内容更新
V1.2	2020-07-16	1. 更新 SPI(新增主/从机全双工收发图文叙述)、I2C(新增主/从发送器、接收机传输序列图及文字叙述) 2. Flash 章节新增 6.4.8 标识符叙述, 新增 6.5.2.10-6.5.2.13 章节寄存器内容