



Holtek 32-Bit 带 Arm® Cortex®-M3 内核单片机

# HT32F12365/HT32F12366/HT32F22366 开发使用手册

Revision: V1.20    Date: September 14, 2020

[www.holtek.com](http://www.holtek.com)

## 目录

<b>1 简介 .....</b>	<b>28</b>
概述 .....	28
特性 .....	29
单片机信息 .....	34
方框图 .....	35
<b>2 文档协议 .....</b>	<b>36</b>
<b>3 系统结构 .....</b>	<b>37</b>
Arm® Cortex®-M3 处理器 .....	37
总线结构 .....	38
存储器体系 .....	39
存储器映射 .....	40
嵌入式 Flash 存储器 .....	43
嵌入式 SRAM 存储器 .....	43
AHB 外设 .....	43
APB 外设 .....	43
<b>4 Flash 存储器控制器 (FMC) .....</b>	<b>44</b>
简介 .....	44
特性 .....	44
功能描述 .....	45
Flash 存储器映射 .....	45
Flash 存储器结构 .....	46
等待状态设置 .....	46
启动配置 .....	47
页擦除 .....	48
整片擦除 .....	49
字编程 .....	50
选项字节描述 .....	51
页擦除 / 编程保护 .....	51
安全保护 .....	53
寄存器列表 .....	54
寄存器描述 .....	55
Flash 目标地址寄存器 – TADR .....	55
Flash 写数据寄存器 – WRDR .....	56
Flash 操作命令寄存器 – OCMR .....	57
Flash 操作控制寄存器 – OPCR .....	58
Flash 操作中断使能寄存器 – OIER .....	59
Flash 操作中断状态寄存器 – OISR .....	60
Flash 页擦除 / 编程保护状态寄存器 – PPSR .....	61
Flash 安全保护状态寄存器 – CPSR .....	62

Flash 向量映射控制寄存器 – VMCR.....	63
Flash 制造商和单片机 ID 寄存器 – MDID.....	64
Flash 页数量状态寄存器 – PNSR .....	65
Flash 页大小状态寄存器 – PSSR .....	66
设备 ID 寄存器 – DID.....	67
Flash 缓存和预取控制寄存器 – CFCR .....	68
SRAM 启动向量寄存器 n – SBVTn, n = 0 ~ 3 .....	69
自定义 ID 寄存器 n – CIDRn, n = 0 ~ 3 .....	70
<b>5 电源控制单元 (PWRCU).....</b>	<b>71</b>
简介 .....	71
特性 .....	72
功能描述 .....	72
备份域 .....	72
V <sub>DD</sub> 电源域 .....	73
1.5 V 电源域 .....	75
工作模式 .....	75
寄存器列表 .....	77
寄存器描述 .....	78
备份域状态寄存器 – BAKSR .....	78
备份域控制寄存器 – BAKCR .....	79
备份域测试寄存器 – BAKTEST .....	80
低电压 / 欠压检测控制和状态寄存器 – LVDCSR.....	81
备份寄存器 n – BAKREGn, n = 0 ~ 9 .....	82
<b>6 时钟控制单元 (CKCU).....</b>	<b>83</b>
简介 .....	83
特性 .....	85
功能描述 .....	85
外部高速晶振 – HSE.....	85
内部高速 RC 振荡器 – HSI .....	86
自动微调内部高速 RC 振荡器 – HSI .....	86
锁相环 – PLL .....	88
USB 锁相环 – USB PLL.....	90
外部低速晶振 – LSE .....	92
内部低速 RC 振荡器 – LSI.....	92
时钟就绪标志位 .....	92
系统时钟选择 – CK_SYS .....	93
HSE 时钟监测器.....	94
时钟输出能力 .....	94
寄存器列表 .....	95
寄存器描述 .....	96
全局时钟配置寄存器 – GCFGR.....	96
全局时钟控制寄存器 – GCCR .....	98

全局时钟状态寄存器 – GCSR.....	99
全局时钟中断寄存器 – GCIR.....	100
PLL 配置寄存器 – PLLCFGR.....	102
PLL 控制寄存器 – PLLCR.....	103
AHB 配置寄存器 – AHBCFGR.....	104
AHB 时钟控制寄存器 – AHBCCR.....	105
APB 配置寄存器 – APBCFGR.....	107
APB 时钟控制寄存器 0 – APBCCR0.....	108
APB 时钟控制寄存器 1 – APBCCR1.....	110
时钟源状态寄存器 – CKST.....	111
APB 外设时钟选择寄存器 0 – APBPCR0.....	112
APB 外设时钟选择寄存器 1 – APBPCR1.....	114
HSI 控制寄存器 – HSICR.....	116
HSI 自动微调计数器寄存器 – HSIATCR.....	117
低功耗控制寄存器 – LPCR.....	118
单片机调试控制寄存器 – MCUDBGCR.....	119
<b>7 复位控制单元 (RSTCU).....</b>	<b>122</b>
简介.....	122
功能描述.....	123
上电复位.....	123
系统复位.....	123
AHB 和 APB 单元复位.....	123
寄存器列表.....	124
寄存器描述.....	124
全局复位状态寄存器 – GRSR.....	124
AHB 外设复位寄存器 – AHBPRSTR.....	125
APB 外设复位寄存器 0 – APBPRSTR0.....	127
APB 外设复位寄存器 1 – APBPRSTR1.....	129
<b>8 通用 I/O (GPIO).....</b>	<b>131</b>
简介.....	131
特性.....	132
功能描述.....	132
默认的 GPIO 引脚配置.....	132
通用 I/O – GPIO.....	132
GPIO 锁定机制.....	134
寄存器列表.....	134
寄存器描述.....	136
端口 A 数据方向控制寄存器 – PADIRCR.....	136
端口 A 输入功能使能控制寄存器 – PAINER.....	137
端口 A 上拉选择寄存器 – PAPUR.....	138
端口 A 下拉选择寄存器 – PAPDR.....	139
端口 A 漏极开路选择寄存器 – PAODR.....	140

端口 A 输出电流驱动选择寄存器 – PADRVR .....	141
端口 A 锁定寄存器 – PALOCKR .....	142
端口 A 数据输入寄存器 – PADINR .....	143
端口 A 输出数据寄存器 – PADOUTR .....	144
端口 A 输出置位 / 复位控制寄存器 – PASRR .....	145
端口 A 输出复位寄存器 – PARR .....	146
端口 B 数据方向控制寄存器 – PBDIRCR .....	147
端口 B 输入功能使能控制寄存器 – PBINER .....	148
端口 B 上拉选择寄存器 – PBPUR .....	149
端口 B 下拉选择寄存器 – PBPDR .....	150
端口 B 漏极开路选择寄存器 – PBODR .....	151
端口 B 输出电流驱动选择寄存器 – PBDRVR .....	152
端口 B 锁定寄存器 – PBLOCKR .....	153
端口 B 数据输入寄存器 – PBDINR .....	154
端口 B 输出数据寄存器 – PBDOUTR .....	155
端口 B 输出置位 / 复位控制寄存器 – PBSRR .....	156
端口 B 输出复位寄存器 – PBRR .....	157
端口 C 数据方向控制寄存器 – PCDIRCR .....	158
端口 C 输入功能使能控制寄存器 – PCINER .....	159
端口 C 上拉选择寄存器 – PCPUR .....	160
端口 C 下拉选择寄存器 – PCPDR .....	161
端口 C 漏极开路选择寄存器 – PCODR .....	162
端口 C 输出电流驱动选择寄存器 – PCDRVR .....	163
端口 C 锁定寄存器 – PCLOCKR .....	164
端口 C 数据输入寄存器 – PCDINR .....	165
端口 C 输出数据寄存器 – PCDOUTR .....	166
端口 C 输出置位 / 复位控制寄存器 – PCSRR .....	167
端口 C 输出复位寄存器 – PCRR .....	168
端口 D 数据方向控制寄存器 – PDDIRCR .....	169
端口 D 输入功能使能控制寄存器 – PDINER .....	170
端口 D 上拉选择寄存器 – PDPUR .....	171
端口 D 下拉选择寄存器 – PDPDR .....	172
端口 D 漏极开路选择寄存器 – PDODR .....	173
端口 D 输出电流驱动选择寄存器 – PDDRVR .....	174
端口 D 锁定寄存器 – PDLOCKR .....	175
端口 D 数据输入寄存器 – PDDINR .....	176
端口 D 输出数据寄存器 – PDDOUTR .....	177
端口 D 输出置位 / 复位控制寄存器 – PDSRR .....	178
端口 D 输出复位寄存器 – PDRR .....	179
端口 E 数据方向控制寄存器 – PEDIRCR .....	180
端口 E 输入功能使能控制寄存器 – PEINER .....	181
端口 E 上拉选择寄存器 – PEPUR .....	182
端口 E 下拉选择寄存器 – PEPDR .....	183
端口 E 漏极开路选择寄存器 – PEODR .....	184
端口 E 输出电流驱动选择寄存器 – PEDRVR .....	185

端口 E 锁定寄存器 – PELOCKR .....	186
端口 E 数据输入寄存器 – PEDINR .....	187
端口 E 输出数据寄存器 – PEDOUTR .....	188
端口 E 输出置位 / 复位控制寄存器 – PESRR .....	189
端口 E 输出复位寄存器 – PERR.....	190
<b>9 复用功能输入 / 输出控制单元 (AFIO).....</b>	<b>191</b>
简介 .....	191
特性 .....	192
功能描述 .....	192
外部中断引脚选择 .....	192
复用功能 .....	193
锁定机制 .....	193
寄存器列表 .....	193
寄存器描述 .....	194
EXTI 来源选择寄存器 0 – ESSR0 .....	194
EXTI 来源选择寄存器 1 – ESSR1 .....	195
GPIO 端口 x 配置低寄存器 – GPxCFGLR, x = A, B, C, D, E .....	196
GPIO 端口 x 配置高寄存器 – GPxCFGHR, x = A, B, C, D, E.....	197
<b>10 嵌套向量中断控制器 (NVIC).....</b>	<b>198</b>
简介 .....	198
特性 .....	200
功能描述 .....	201
SysTick 校准 .....	201
寄存器列表 .....	201
<b>11 外部中断 / 事件控制器 (EXTI).....</b>	<b>203</b>
简介 .....	203
特性 .....	203
功能描述 .....	204
唤醒事件管理 .....	204
外部中断 / 事件线配置 .....	205
中断和去抖 .....	205
寄存器列表 .....	206
寄存器描述 .....	207
EXTI 中断 n 配置寄存器 – EXTICFGRn, n = 0 ~ 15 .....	207
EXTI 中断控制寄存器 – EXTICR.....	208
EXTI 中断边沿标志寄存器 – EXTIEDGEFLGR .....	209
EXTI 中断边沿状态寄存器 – EXTIEDGESR .....	210
EXTI 中断软件设置命令寄存器 – EXTISSCR .....	211
EXTI 中断唤醒控制寄存器 – EXTIWAKUPCR .....	212
EXTI 中断唤醒极性寄存器 – EXTIWAKUPPOLR.....	213
EXTI 中断唤醒标志寄存器 – EXTIWAKUPFLG .....	214

<b>12 模数转换器 (ADC)</b>	<b>215</b>
模数转换器	215
特性	216
功能描述	217
ADC 时钟设置	217
常规和高优先级通道选择	217
转换模式	217
外部事件启动转换	222
高优先级组管理	222
采样时间设定	224
数据格式和对齐	224
模拟看门狗	224
中断	225
PDMA 请求	225
寄存器列表	226
寄存器描述	228
ADC 复位寄存器 – ADCRST	228
ADC 常规转换模式寄存器 – ADCCONV	229
ADC 高优先级转换模式寄存器 – ADCHCONV	230
ADC 常规转换列表寄存器 0 – ADCLST0	231
ADC 常规转换列表寄存器 1 – ADCLST1	232
ADC 常规转换列表寄存器 2 – ADCLST2	233
ADC 常规转换列表寄存器 3 – ADCLST3	234
ADC 高优先级转换列表寄存器 – ADCHLST	235
ADC 输入偏移量寄存器 n – ADCOFRn, n = 0 ~ 15	236
ADC 输入采样时间寄存器 n – ADCSTRn, n = 0 ~ 15	237
ADC 常规转换数据寄存器 y – ADCDRy, y = 0 ~ 15	238
ADC 高优先级转换数据寄存器 y – ADCHDRy, y = 0 ~ 3	239
ADC 常规触发控制寄存器 – ADCTCR	240
ADC 常规触发源寄存器 – ADCTSR	241
ADC 高优先级触发控制寄存器 – ADCHTCR	242
ADC 高优先级触发源寄存器 – ADCHTSR	243
ADC 看门狗控制寄存器 – ADCWCR	244
ADC 看门狗下限阈值寄存器 – ADCLTR	245
ADC 看门狗上限阈值寄存器 – ADCUTR	246
ADC 中断使能寄存器 – ADCIER	247
ADC 中断原始状态寄存器 – ADCIRAW	248
ADC 中断状态寄存器 – ADCISR	249
ADC 中断清除寄存器 – ADCICLR	250
ADC DMA 请求寄存器 – ADCDMAR	251
<b>13 比较器 (CMP)</b>	<b>252</b>
简介	252
特性	252

功能描述 .....	253
比较器输入和输出 .....	253
比较器电压参考 .....	253
中断和唤醒 .....	254
功耗模式和迟滞 .....	255
比较器写保护机制 .....	255
寄存器列表 .....	255
寄存器描述 .....	256
比较器控制寄存器 n – CMPCRn, n = 0 或 1 .....	256
比较器电压参考值寄存器 n – CVRVALRn, n = 0 或 1 .....	258
比较器中断使能寄存器 n – CMPIERn, n = 0 或 1 .....	259
比较器转换标志寄存器 n – CMPTFRn, n = 0 或 1 .....	260
<b>14 通用定时器 (GPTM).....</b>	<b>261</b>
简介 .....	261
特性 .....	262
功能描述 .....	262
计数器模式 .....	262
时钟控制器 .....	265
触发控制器 .....	266
从机控制器 .....	268
主机控制器 .....	270
通道控制器 .....	271
输入级 .....	274
输出级 .....	275
更新管理 .....	279
正交解码器 .....	280
数字滤波器 .....	282
ETIF 为高电平时清除 CHxOREF .....	282
单脉冲模式 .....	283
非对称 PWM 模式.....	285
定时器互连 .....	286
触发 ADC 开启.....	289
PDMA 请求.....	289
寄存器列表 .....	290
寄存器描述 .....	291
定时器计数器配置寄存器 – CNTCFR.....	291
定时器模式配置寄存器 – MDCFR .....	292
定时器触发配置寄存器 – TRCFR.....	294
定时器控制寄存器 – CTR .....	296
通道 0 输入配置寄存器 – CH0ICFR.....	297
通道 1 输入配置寄存器 – CH1ICFR.....	299
通道 2 输入配置寄存器 – CH2ICFR.....	300
通道 3 输入配置寄存器 – CH3ICFR.....	301



通道 0 输出配置寄存器 – CH0OCFR .....	302
通道 1 输出配置寄存器 – CH1OCFR .....	304
通道 2 输出配置寄存器 – CH2OCFR .....	306
通道 3 输出配置寄存器 – CH3OCFR .....	308
通道控制寄存器 – CHCTR .....	310
通道极性配置寄存器 – CHPOLR .....	311
定时器 PDMA / 中断控制寄存器 – DICTR .....	312
定时器事件发生器寄存器 – EVGR .....	313
定时器中断状态寄存器 – INTSR .....	315
定时器计数器寄存器 – CNTR .....	317
定时器预分频器寄存器 – PSCR .....	318
定时器计数器重载寄存器 – CRR .....	319
通道 0 捕捉 / 比较寄存器 – CH0CCR .....	320
通道 1 捕捉 / 比较寄存器 – CH1CCR .....	321
通道 2 捕捉 / 比较寄存器 – CH2CCR .....	322
通道 3 捕捉 / 比较寄存器 – CH3CCR .....	323
通道 0 非对称比较寄存器 – CH0ACR .....	324
通道 1 非对称比较寄存器 – CH1ACR .....	325
通道 2 非对称比较寄存器 – CH2ACR .....	326
通道 3 非对称比较寄存器 – CH3ACR .....	327
<b>15 基本功能定时器 (BFTM) .....</b>	<b>328</b>
简介 .....	328
特性 .....	328
功能描述 .....	329
重复模式 .....	329
单次模式 .....	330
寄存器列表 .....	331
寄存器描述 .....	331
BFTM 控制寄存器 – BFTMCR .....	331
BFTM 状态寄存器 – BFTMSR .....	332
BFTM 计数器值寄存器 – BFTMCNTR .....	333
BFTM 比较值寄存器 – BFTMCMPR .....	334
<b>16 马达控制定时器 (MCTM) .....</b>	<b>335</b>
简介 .....	335
特性 .....	336
功能描述 .....	337
计时器模式 .....	337
时钟控制器 .....	341
触发控制器 .....	342
从机控制器 .....	343
主机控制器 .....	345
通道控制器 .....	346

输入级 .....	348
输出级 .....	349
更新管理 .....	361
正交解码器 .....	363
数字滤波器 .....	364
ETIF 为高电平时清除 CHxOREF .....	365
单脉冲模式 .....	366
非对称 PWM 模式 .....	368
定时器互连 .....	369
触发 ADC 开启 .....	373
锁定级别表 .....	373
PDMA 请求 .....	374
寄存器列表 .....	375
寄存器描述 .....	376
定时器计数器配置寄存器 – CNTCFR .....	376
定时器模式配置寄存器 – MDCFR .....	377
定时器触发配置寄存器 – TRCFR .....	379
定时器计数器寄存器 – CTR .....	381
通道 0 输入配置寄存器 – CH0ICFR .....	382
通道 1 输入配置寄存器 – CH1ICFR .....	384
通道 2 输入配置寄存器 – CH2ICFR .....	385
通道 3 输入配置寄存器 – CH3ICFR .....	386
通道 0 输出配置寄存器 – CH0OCFR .....	387
通道 1 输出配置寄存器 – CH1OCFR .....	389
通道 2 输出配置寄存器 – CH2OCFR .....	391
通道 3 输出配置寄存器 – CH3OCFR .....	393
通道控制寄存器 – CHCTR .....	395
通道极性配置寄存器 – CHPOLR .....	397
通道刹车配置寄存器 – CHBRKCFR .....	398
通道刹车控制寄存器 – CHBRKCTR .....	399
定时器 PDMA / 中断控制寄存器 – DICTR .....	402
定时器事件发生器寄存器 – EVGR .....	404
定时器中断状态寄存器 – INTSR .....	405
定时器计数器寄存器 – CNTR .....	407
定时器预分频器寄存器 – PSCR .....	408
定时器计数器重载寄存器 – CRR .....	409
定时器重复寄存器 – REPR .....	410
通道 0 捕捉 / 比较寄存器 – CH0CCR .....	411
通道 1 捕捉 / 比较寄存器 – CH1CCR .....	412
通道 2 捕捉 / 比较寄存器 – CH2CCR .....	413
通道 3 捕捉 / 比较寄存器 – CH3CCR .....	414
通道 0 非对称比较寄存器 – CH0ACR .....	415
通道 1 非对称比较寄存器 – CH1ACR .....	416
通道 2 非对称比较寄存器 – CH2ACR .....	417
通道 3 非对称比较寄存器 – CH3ACR .....	418

<b>17 实时时钟 (RTC)</b>	<b>419</b>
简介	419
特性	419
功能描述	420
RTC 相关寄存器复位	420
读取 RTC 寄存器	420
低速时钟配置	420
RTC 计数器操作	420
中断和唤醒控制	421
RTCOUT 输出引脚配置	421
寄存器列表	422
寄存器描述	423
RTC 计数器寄存器 – RTCCNT	423
RTC 比较寄存器 – RTCCMP	424
RTC 控制寄存器 – RTCCR	425
RTC 状态寄存器 – RTCSR	427
RTC 中断和唤醒使能寄存器 – RTCIWEN	428
<b>18 看门狗定时器 (WDT)</b>	<b>429</b>
简介	429
特性	429
功能描述	430
寄存器列表	432
寄存器描述	432
看门狗定时器控制寄存器 – WDTCR	432
看门狗定时器模式寄存器 0 – WDTMR0	433
看门狗定时器模式寄存器 1 – WDTMR1	434
看门狗定时器状态寄存器 – WDTSR	435
看门狗定时器保护寄存器 – WDTPR	436
看门狗定时器时钟选择寄存器 – WDTCSR	437
<b>19 内部集成电路 (I<sup>2</sup>C)</b>	<b>438</b>
简介	438
特性	439
功能描述	439
双线串行接口	439
START 和 STOP 条件	439
数据有效性	440
寻址格式	441
数据传输和确认	443
时钟同步	444
仲裁	444
广播呼叫寻址	445

总线错误 .....	445
地址屏蔽使能 .....	445
地址捕获 .....	445
工作模式 .....	445
保持 SCL 线状态的条件 .....	451
I <sup>2</sup> C 超时功能 .....	451
PDMA 接口 .....	452
寄存器列表 .....	452
寄存器描述 .....	453
I <sup>2</sup> C 控制寄存器 – I2CCR .....	453
I <sup>2</sup> C 中断使能寄存器 – I2CIER .....	455
I <sup>2</sup> C 地址寄存器 – I2CADDR .....	456
I <sup>2</sup> C 状态寄存器 – I2CSR .....	457
I <sup>2</sup> C SCL 高电平周期发生寄存器 – I2CSHPGR .....	460
I <sup>2</sup> C SCL 低电平周期产生寄存器 – I2CSLPGR .....	461
I <sup>2</sup> C 数据寄存器 – I2CDR .....	462
I <sup>2</sup> C 目标寄存器 – I2CTAR .....	463
I <sup>2</sup> C 地址屏蔽寄存器 – I2CADDRMR .....	464
I <sup>2</sup> C 地址捕获寄存器 – I2CADDRSR .....	465
I <sup>2</sup> C 超时寄存器 – I2CTOUT .....	466
<b>20 串行外设接口 (SPI).....</b>	<b>467</b>
简介 .....	467
特性 .....	468
功能描述 .....	468
主机模式 .....	468
从机模式 .....	468
SPI 串行帧格式 .....	469
状态标志 .....	473
寄存器列表 .....	476
寄存器描述 .....	476
SPI 控制寄存器 0 – SPICR0 .....	476
SPI 控制寄存器 1 – SPICR1 .....	478
SPI 中断使能寄存器 – SPIIER .....	480
SPI 时钟预分频器寄存器 – SPICPR .....	481
SPI 数据寄存器 – SPIDR .....	482
SPI 状态寄存器 – SPISR .....	483
SPI FIFO 控制寄存器 – SPIFCR .....	485
SPI FIFO 状态寄存器 – SPIFSR .....	486
SPI FIFO 超时计数器寄存器 – SPIFTOCR .....	487
<b>21 通用同步异步收发器 (USART) .....</b>	<b>488</b>
简介 .....	488
特性 .....	489

功能描述 .....	490
串行数据格式 .....	490
波特率发生器 .....	491
硬件流控制 .....	492
IrDA 模式 .....	494
RS485 模式 .....	496
同步主机模式 .....	498
中断和状态 .....	500
PDMA 接口 .....	500
寄存器列表 .....	500
寄存器描述 .....	501
USART 数据寄存器 – USRDR .....	501
USART 控制寄存器 – USRCR .....	502
USART FIFO 控制寄存器 – USRFCR .....	504
USART 中断使能寄存器 – USRIER .....	505
USART 状态 & 中断标志位寄存器 – USRSIFR .....	507
USART 时序参数寄存器 – USRTPR .....	509
USART IrDA 控制寄存器 – IrDACR .....	510
USART RS485 控制寄存器 – RS485CR .....	511
USART 同步控制寄存器 – SYNCR .....	512
USART 分频器锁存寄存器 – USRDLR .....	513
USART 测试寄存器 – USRTSTR .....	514
<b>22 通用异步收发器 (UART) .....</b>	<b>515</b>
简介 .....	515
特性 .....	516
功能描述 .....	516
串行数据格式 .....	516
波特率发生器 .....	517
中断和状态 .....	518
PDMA 接口 .....	518
寄存器列表 .....	519
寄存器描述 .....	519
UART 数据寄存器 – URDR .....	519
UART 控制寄存器 – URCCR .....	520
UART 中断使能寄存器 – URIER .....	522
UART 状态 & 中断标志位寄存器 – URSIFR .....	523
UART 分频器锁存寄存器 – URDLR .....	524
UART 测试寄存器 – URTSTR .....	525
<b>23 智能卡接口 (SCI) .....</b>	<b>526</b>
简介 .....	526
特性 .....	527
功能描述 .....	527

基本时间单元计数器 .....	527
保护时间计数器 .....	529
等待时间计数器 .....	530
智能卡时钟和数据选择 .....	531
智能卡检测 .....	531
SCI 数据传输模式 .....	532
中断发生器 .....	534
PDMA 接口 .....	534
寄存器列表 .....	535
寄存器描述 .....	536
SCI 控制寄存器 – CR .....	536
SCI 状态寄存器 – SR .....	538
SCI 通信控制寄存器 – CCR .....	539
SCI 基本时间单元寄存器 – ETUR .....	540
SCI 保护时间寄存器 – GTR .....	541
SCI 等待时间寄存器 – WTR .....	542
SCI 中断使能寄存器 – IER .....	543
SCI 中断挂起寄存器 – IPR .....	544
SCI 发送缓冲器 – TXB .....	545
SCI 接收缓冲器 – RXB .....	546
SCI 预分频器寄存器 – PSCR .....	547
<b>24 USB 设备控制器 (USB) .....</b>	<b>548</b>
简介 .....	548
特性 .....	548
功能描述 .....	549
端点 .....	549
EP_SRAM .....	549
串行接口引擎 – SIE .....	550
双缓冲 .....	550
挂起模式和唤醒 .....	552
远程唤醒 .....	552
寄存器列表 .....	552
寄存器描述 .....	554
USB 控制和状态寄存器 – USBCSR .....	554
USB 中断使能寄存器 – USBIER .....	556
USB 中断状态寄存器 – USBISR .....	557
USB 帧计数寄存器 – USBFCR .....	558
USB 设备地址寄存器 – USBDEVAR .....	559
USB 端点 0 控制和状态寄存器 – USBEP0CSR .....	560
USB 端点 0 中断使能控制寄存器 – USBEP0IER .....	561
USB 端点 0 中断状态寄存器 – USBEP0ISR .....	563
USB 端点 0 传输计数寄存器 – USBEP0TCR .....	564
USB 端点 0 配置寄存器 – USBEP0CFGR .....	565

USB 端点 1~3 控制和状态寄存器 – USBEPnCSR, n = 1~3 .....	566
USB 端点 1~3 中断使能寄存器 – USBEPnIER, n = 1~3 .....	567
USB 端点 1~3 中断状态寄存器 – USBEPnISR, n = 1~3 .....	568
USB 端点 1~3 传输计数寄存器 – USBEPnTCR, n = 1~3 .....	569
USB 端点 1~3 配置寄存器 – USBEPnCFGR, n = 1~3 .....	570
USB 端点 4~7 控制和状态寄存器 – USBEPnCSR, n = 4~7 .....	571
USB 端点 4~7 中断使能寄存器 – USBEPnIER, n = 4~7 .....	573
USB 端点 4~7 中断状态寄存器 – USBEPnISR, n = 4~7 .....	574
USB 端点 4~7 传输计数寄存器 – USBEPnTCR, n = 4~7 .....	575
USB 端点 4~7 配置寄存器 – USBEPnCFGR, n = 4~7 .....	576
<b>25 外设直接存储器访问 (PDMA).....</b>	<b>577</b>
简介 .....	577
特性 .....	577
功能描述 .....	578
AHB 主机 .....	578
PDMA 通道 .....	578
PDMA 请求映射 .....	578
通道传输 .....	579
通道优先级 .....	579
传输请求 .....	580
地址模式 .....	580
自动重载 .....	581
传输中断 .....	581
寄存器列表 .....	581
寄存器描述 .....	583
PDMA 通道 n 控制寄存器 – PDMACHnCR, n = 0~11 .....	583
PDMA 通道 n 源地址寄存器 – PDMACHnSADR, n = 0~11 .....	585
PDMA 通道 n 目标地址寄存器 – PDMACHnDADR, n = 0~11 .....	586
PDMA 通道 n 传输大小寄存器 – PDMACHnTSR, n = 0~11 .....	587
PDMA 通道 n 当前传输大小寄存器 – PDMACHnCTSR, n = 0~11 .....	588
PDMA 中断状态寄存器 0 – PDMAISR0 .....	589
PDMA 中断状态寄存器 1 – PDMAISR1 .....	590
PDMA 中断状态清除寄存器 0 – PDMAISCR0 .....	591
PDMA 中断状态清除寄存器 1 – PDMAISCR1 .....	592
PDMA 中断使能寄存器 0 – PDMAIER0 .....	593
PDMA 中断使能寄存器 1 – PDMAIER1 .....	594
<b>26 外部总线接口 (EBI).....</b>	<b>595</b>
简介 .....	595
特性 .....	595
功能描述 .....	596
非复用 8-bit 数据 8-bit 地址模式 .....	597
非复用 16-bit 数据 N-bit 地址模式 .....	598

复用 16-bit 数据 16-bit 地址模式 .....	599
复用 8-bit 数据 24-bit 地址模式 .....	600
页读取操作 .....	601
写缓冲器和 EBI 状态 .....	603
总线周转和空闲周期 .....	603
AHB 事务宽度转换 .....	604
EBI Bank 访问 .....	606
EBI 就绪 .....	607
PDMA 请求 .....	607
寄存器列表 .....	607
寄存器描述 .....	608
EBI 控制寄存器 – EBICR .....	608
EBI 页控制寄存器 – EBIPCR .....	611
EBI 状态寄存器 – EBISR .....	612
EBI 地址时序寄存器 n – EBIATRn, n = 0 ~ 3 .....	613
EBI 读时序寄存器 n – EBIRTRn, n = 0 ~ 3 .....	614
EBI 写时序寄存器 n – EBIWTRn, n = 0 ~ 3 .....	615
EBI 极性寄存器 n – EBIPRn, n = 0 ~ 3 .....	616
EBI 中断使能寄存器 – EBIIENR .....	617
EBI 中断标志寄存器 – EBIIFR .....	618
EBI 中断清除寄存器 – EBIIFCR .....	619
<b>27 内置音频接口 (I²S) .....</b>	<b>620</b>
简介 .....	620
特性 .....	620
功能描述 .....	621
I²S 主机和从机模式 .....	621
I²S 时钟频率发生器 .....	622
I²S 接口格式 .....	624
FIFO 控制和分配 .....	631
PDMA 和中断 .....	632
寄存器列表 .....	632
寄存器描述 .....	633
I²S 控制寄存器 – I2SCR .....	633
I²S 中断使能寄存器 – I2SIER .....	635
I²S 时钟分频寄存器 – I2SCDR .....	636
I²S TX 数据寄存器 – I2STXDR .....	637
I²S RX 数据寄存器 – I2SRXDR .....	637
I²S FIFO 控制寄存器 – I2SFCR .....	638
I²S 状态寄存器 – I2SSR .....	639
I²S 频率计数值寄存器 – I2SRCNTR .....	641
<b>28 循环冗余校验 (CRC) .....</b>	<b>642</b>
简介 .....	642



特性 .....	643
功能描述 .....	643
CRC 计算 .....	643
CRC 计算按字节反序与位反序 .....	643
与 PDMA 搭配工作 .....	644
寄存器列表 .....	644
寄存器描述 .....	645
CRC 控制寄存器 – CRCCR .....	645
CRC 种子值寄存器 – CRCSDR .....	646
CRC 校验和寄存器 – CRCCSR .....	647
CRC 数据寄存器 – CRCDR .....	648
<b>29 AES 加密 / 解密接口 (AES) .....</b>	<b>649</b>
简介 .....	649
特性 .....	649
功能描述 .....	650
AES 模式 .....	650
AES 状态 .....	653
AES PDMA 接口 .....	653
AES 中断 .....	653
AES 初始矢量 .....	653
AES 字交换 .....	654
寄存器列表 .....	654
寄存器描述 .....	655
AES 控制寄存器 – AESCR .....	655
AES 状态寄存器 – AESSR .....	656
AES DMA 寄存器 – AESDMAR .....	657
AES 中断状态寄存器 – AESISR .....	658
AES 中断使能寄存器 – AESIER .....	659
AES 数据输入寄存器 – AESDINR .....	660
AES 数据输出寄存器 – AESDOUTR .....	660
AES 密钥寄存器 n – AESKEYRn, n = 0 ~ 7 .....	661
AES 初始矢量寄存器 n – AESIVRn, n = 0 ~ 3 .....	661
<b>30 SDIO 主机控制器 (SDIO) .....</b>	<b>662</b>
简介 .....	662
特性 .....	662
功能描述 .....	662
SD 时钟 .....	663
SD 协议 .....	664
命令 .....	665
响应 .....	665
数据 .....	667
缓冲器状态 .....	668

中断 .....	668
DMA 请求 .....	668
寄存器列表 .....	670
寄存器描述 .....	671
块大小寄存器 – BLSIZE .....	671
块计数寄存器 – BLCNT .....	672
参数寄存器 – ARG .....	673
传输模式寄存器 – TMR .....	674
命令寄存器 – CMD .....	675
响应寄存器 n – RESPn, n = 0 ~ 3 .....	676
数据端口寄存器 – DR .....	677
当前状态寄存器 – PSR .....	678
控制寄存器 – CR .....	679
时钟控制寄存器 – CLKCR .....	680
超时控制寄存器 – TMOCR .....	681
软件复位寄存器 – SWRST .....	682
状态寄存器 – SR .....	683
状态使能寄存器 – SER .....	685
中断使能寄存器 – IER .....	687
<b>31 CMOS 传感器接口 (CSIF, 仅适用于 HT32F22366) .....</b>	<b>689</b>
简介 .....	689
特性 .....	689
功能描述 .....	690
CSIF 信号 .....	690
CSIF 帧时序 .....	690
像素数据格式 .....	692
窗口捕捉 .....	693
行和列子采样 .....	694
PDMA 数据发送 – Rx 触发 PDMA .....	696
中断和状态 .....	696
寄存器列表 .....	697
寄存器描述 .....	698
CSIF 使能寄存器 – CSIFENR .....	698
CSIF 控制寄存器 – CSIFCR .....	699
CSIF 图像宽度和高度寄存器 – CSIFIMGWH .....	700
CSIF 窗口捕捉寄存器 0 – CSIFWCR0 .....	701
CSIF 窗口捕捉寄存器 1 – CSIFWCR1 .....	702
CSIF 子采样寄存器 – CSIFSMP .....	703
CSIF 列子采样寄存器 – CSIFSMPCOL .....	704
CSIF 行子采样寄存器 – CSIFSMPROW .....	705
CSIF FIFO 寄存器 n – CSIFFIFO <sub>n</sub> , n=0~7 .....	706
CSIF 中断使能寄存器 – CSIFIER .....	707
CSIF 状态寄存器 – CSIFSR .....	708

## 表列表

表 1. 单片机的特性及外设列表 .....	34
表 2. 文档协议 .....	36
表 3. 寄存器列表 .....	41
表 4. Flash 存储器和选项字节 .....	46
表 5. 等待状态周期与 HCLK 之间的关系 .....	46
表 6. 启动模式 .....	47
表 7. 选项字节的存储器映射 .....	51
表 8. 受保护主 Flash 页的访问权限 .....	52
表 9. 安全保护使能时的访问权限 .....	53
表 10. FMC 的寄存器列表 .....	54
表 11. 工作模式定义 .....	75
表 12. 进入 / 退出省电模式 .....	76
表 13. 系统复位后的电源状态 .....	77
表 14. PWRCU 寄存器列表 .....	77
表 15. 输出分频器 2 值映射 .....	89
表 16. 反馈分频器 2 值映射 .....	89
表 17. USB PLL 输出分频器 2 值映射 .....	91
表 18. USB PLL 反馈分频器 2 值映射 .....	91
表 19. CKOUT 的时钟源 .....	94
表 20. CKCU 寄存器列表 .....	95
表 21. RSTCU 寄存器列表 .....	124
表 22. AFIO、GPIO 和 I/O 引脚控制信号真值表 .....	133
表 23. GPIO 寄存器列表 .....	134
表 24. 外设 AFIO 分配列表范例 .....	193
表 25. AFIO 寄存器列表 .....	193
表 26. 异常类型 .....	198
表 27. NVIC 寄存器列表 .....	201
表 28. EXTI 寄存器列表 .....	206
表 29. ADCDRy[15:0] (y = 0 ~ 15) 和 ADCHDRy[15:0] (y = 0 ~ 3) 数据格式 .....	224
表 30. A/D 转换器寄存器列表 .....	226
表 31. CMP 寄存器列表 .....	255
表 32. 计数方向和编码信号 .....	281
表 33. GPTM 寄存器列表 .....	290
表 34. GPTM 内部触发器连接 .....	295
表 35. BFTM 寄存器列表 .....	331
表 36. 比较匹配输出设置 .....	351
表 37. 刹车事件发生时的互补输出控制位 .....	360
表 38. 计数方向和编码信号 .....	364
表 39. 锁定级别表 .....	373

表 40. MCTM 寄存器列表 .....	375
表 41. MCTM 内部触发器连接 .....	380
表 42. LSE 在不同启动模式下的工作电流和启动时间 .....	420
表 43. RTCOUT 输出模式和有效电平设置 .....	421
表 44. RTC 寄存器列表 .....	422
表 45. 看门狗定时器寄存器列表 .....	432
表 46. 保持 SCL 线状态的条件 .....	451
表 47. I <sup>2</sup> C 寄存器列表 .....	452
表 48. I <sup>2</sup> C 时钟设置范例 .....	461
表 49. SPI 接口格式设置 .....	469
表 50. SPI 模式故障触发条件 .....	474
表 51. SPI 主机模式 SEL 引脚状态 .....	474
表 52. SPI 寄存器列表 .....	476
表 53. 波特率误差计算 – CK_USART = 48 MHz .....	491
表 54. 波特率误差计算 – CK_USART = 96 MHz .....	492
表 55. USART 寄存器列表 .....	500
表 56. 波特率误差计算 – CK_UART = 48 MHz .....	517
表 57. 波特率误差计算 – CK_UART = 96 MHz .....	518
表 58. UART 寄存器列表 .....	519
表 59. 基于 DI 字段编码的 Di 十进制值 .....	528
表 60. 基于 FI 字段编码的 Fi 十进制值 .....	528
表 61. 由 Fi/Di 比例计算得到的 ETU 可能值 .....	528
表 62. SCI 寄存器列表 .....	535
表 63. 端点特性 .....	549
表 64. USB 数据类型和缓冲器容量 .....	549
表 65. USB 寄存器列表 .....	552
表 66. 恢复事件检测 .....	555
表 67. PDMA 通道配置 .....	579
表 68. PDMA 地址模式 .....	580
表 69. PDMA 寄存器列表 .....	581
表 70. AHB 事务宽度与外部设备事务的 EBI 映射 .....	605
表 71. AHB 事务宽度与外部设备事务宽度的 EBI 映射 ( 使用字节通道 EBI_BL[1:0] ) .....	605
表 72. EBI 寄存器列表 .....	607
表 73. F <sub>s</sub> 建议值列表 @ 8 MHz PCLK .....	623
表 74. F <sub>s</sub> 建议值列表 @ 48 MHz PCLK .....	623
表 75. F <sub>s</sub> 建议值列表 @ 72 MHz PCLK .....	623
表 76. F <sub>s</sub> 建议值列表 @ 96 MHz PCLK .....	624
表 77. I <sup>2</sup> S 寄存器列表 .....	632
表 78. CRC 寄存器列表 .....	644
表 79. AES 寄存器列表 .....	654
表 80. 命令格式 .....	665

表 81. R1 响应格式 ..... 666

表 82. R2 响应格式 ..... 666

表 83. R3 响应格式..... 666

表 84. R6 响应格式..... 666

表 85. R7 响应格式 ..... 666

表 86. SDIO 命令寄存器字段和值 ..... 669

表 87. SDIO 寄存器列表..... 670

表 88. CSIF 信号..... 690

表 89. CSIF\_MCK 输出设置 – 请参考 CKCU 章节..... 690

表 90. 像素数据格式 – 无窗口捕捉和子采样 ..... 692

表 91. 窗口捕捉设置 ..... 693

表 92. 行和列子采样设置 ..... 695

表 93. 中断和状态 ..... 696

表 94. 中断状态 ..... 696

表 95. CSIF 寄存器列表..... 697

## 图列表

图 1. 方框图 .....	35
图 2. Cortex®-M3 方框图 .....	38
图 3. 总线结构 .....	39
图 4. 存储器映射 .....	40
图 5. Flash 存储器控制器方框图 .....	44
图 6. Flash 存储器映射 .....	45
图 7. 向量重映射 .....	47
图 8. 页擦除操作流程 .....	48
图 9. 整片擦除操作流程 .....	49
图 10. 字编程操作流程 .....	50
图 11. PWRCU 方框图 .....	71
图 12. 上电复位 / 掉电复位波形 .....	74
图 13. CKCU 方框图 .....	84
图 14. HSE 外部晶体、陶瓷和谐振器 .....	85
图 15. HSI 自动微调方框图 .....	87
图 16. PLL 方框图 .....	88
图 17. USB PLL 方框图 .....	90
图 18. LSE 外部晶体、陶瓷和谐振器 .....	92
图 19. RSTCU 方框图 .....	122
图 20. 上电复位时序图 .....	123
图 21. GPIO 方框图 .....	131
图 22. AFIO/GPIO 控制信号 .....	133
图 23. AFIO 方框图 .....	191
图 24. EXTI 输入通道选择 .....	192
图 25. EXTI 方框图 .....	203
图 26. EXTI 唤醒事件管理 .....	204
图 27. EXTI 中断去抖功能 .....	205
图 28. ADC 方框图 .....	215
图 29. 单次转换模式 .....	218
图 30. 连续转换模式 .....	219
图 31. 常规组非连续转换模式 .....	220
图 32. 高优先级组非连续转换模式 .....	221
图 33. 高优先级组管理 .....	223
图 34. 比较器方框图 .....	252
图 35. 6-Bit Scaler 配置比较器参考电压输出方框图 .....	253
图 36. 比较器输出中断信号 .....	254
图 37. 比较器输出唤醒结构 .....	254
图 38. GPTM 方框图 .....	261
图 39. 向上计数范例 .....	262

图 40. 向下计数范例 .....	263
图 41. 中心对齐计数范例 .....	264
图 42. GPTM 时钟源选择 .....	266
图 43. 触发控制器方框图 .....	267
图 44. 从机控制器方框图 .....	268
图 45. 重启模式下的 GPTM .....	268
图 46. 暂停模式下的 GPTM .....	269
图 47. 触发模式下的 GPTM .....	269
图 48. 主机 GPTMn 和从机 GPTMm/MCTMm 相连接 .....	270
图 49. MTO 选项 .....	270
图 50. 捕捉 / 比较方框图 .....	271
图 51. 输入捕捉模式 .....	272
图 52. PWM 脉冲宽度测量范例 .....	273
图 53. 通道 0 和通道 1 输入级 .....	274
图 54. 通道 2 和通道 3 输入级 .....	274
图 55. 输出级方框图 .....	275
图 56. 翻转模式通道输出参考信号 – CHxPRE = 0 .....	276
图 57. 翻转模式通道输出参考信号 – CHxPRE = 1 .....	276
图 58. PWM 模式通道输出参考信号和向上计数模式下的计数器 .....	277
图 59. PWM 模式通道输出参考信号和向下计数模式下的计数器 .....	277
图 60. 模式通道输出参考信号和中心对齐计数模式下的计数器 .....	278
图 61. 更新事件设置方框图 .....	279
图 62. 输入级和正交解码器方框图 .....	280
图 63. TI0 和 TI1 正交解码器计数 .....	281
图 64. N=2 的 GTn_ETI 引脚数字滤波器方框图 .....	282
图 65. 通过 ETIF 清除 CHxOREF .....	282
图 66. 单脉冲模式 .....	283
图 67. 立即有效模式的延迟 .....	284
图 68. 中心对齐计数模式下的非对称 PWM 模式 .....	285
图 69. 用 GPTM0 CH0OREF 信号暂停 GPTM1 .....	286
图 70. 用 GPTM0 更新事件触发 GPTM1 .....	287
图 71. 用 GPTM0 CH0 输入信号触发 GPTM0 和 GPTM1 .....	288
图 72. GPTM PDMA 映射图 .....	289
图 73. BFTM 方框图 .....	328
图 74. BFTM – 重复模式 .....	329
图 75. BFTM – 单次模式 .....	330
图 76. BFTM – 单次模式计数器更新 .....	330
图 77. MCTM 方框图 .....	335
图 78. 向上计数范例 .....	337
图 79. 向下计数范例 .....	338
图 80. 中心对齐计数范例 .....	339

图 81. 更新事件 1 相关重复机制范例 .....	340
图 82. MCTM 时钟源选择 .....	341
图 83. 触发控制器方框图 .....	342
图 84. 从机控制器方框图 .....	343
图 85. 重启模式下的 MCTM .....	343
图 86. 暂停模式下的 MCTM .....	344
图 87. 触发模式下的 MCTM .....	344
图 88. 主机 MCTMn 和从机 GPTMm 相连接 .....	345
图 89. MTO 选项 .....	345
图 90. 捕捉 / 比较方框图 .....	346
图 91. 输入捕捉模式 .....	347
图 92. PWM Pulse 脉冲宽度测量范例 .....	348
图 93. 通道 0 和通道 1 输入级 .....	348
图 94. 通道 2 和通道 3 输入级 .....	349
图 95. 输出级方框图 .....	350
图 96. 翻转模式通道输出参考信号 – CHxPRE = 0 .....	351
图 97. 翻转模式通道输出参考信号 – CHxPRE = 1 .....	352
图 98. PWM 模式通道输出参考信号和向上计数模式下的计数器 .....	352
图 99. PWM 模式通道输出参考信号和向下计数模式下的计数器 .....	353
图 100. PWM 模式 1 通道输出参考信号和中心对齐计数模式下的计数器 .....	353
图 101. 带死区时间插入的互补输出 .....	354
图 102. MCTM 刹车信号方框图 .....	355
图 103. N = 2 的 MTn_BRK 引脚数字滤波器方框图 .....	355
图 104. 刹车事件发生时的通道 3 输出 .....	356
图 105. 刹车事件发生时的通道 0 ~ 2 互补输出 .....	357
图 106. 刹车事件发生时通道 0 ~ 2 仅一个输出使能 .....	358
图 107. CHxO 和 CHxNO 都处于有效状态时的硬件保护 .....	359
图 108. 更新事件 1 设置方框图 .....	361
图 109. 更新事件 2 更新 CHxE、CHxNE 和 CHxOM .....	362
图 110. 更新事件 2 设置方框图 .....	362
图 111. 输入级和正交解码器方框图 .....	363
图 112. TI0 和 TI1 正交解码器计数 .....	364
图 113. N = 2 的 MTn_ETI 引脚数字滤波器方框图 .....	364
图 114. 通过 ETIF 清除 CHxOREF .....	365
图 115. 单脉冲模式 .....	366
图 116. 立即有效模式的延迟 .....	367
图 117. 中心对齐计数模式下的非对称 PWM 模式 .....	368
图 118. 用 MCTM0 CH0OREF 信号暂停 GPTM0 .....	369
图 119. 用 MCTM0 更新事件 1 触发 GPTM0 .....	370
图 120. 用 MCTM0 CH0 输入信号触发 MCTM0 和 GPTM0 .....	371
图 121. CH1XOR 输入作为霍尔传感器接口 .....	372



图 122. MCTM PDMA 映射图.....	374
图 123. RTC 方框图.....	419
图 124. 看门狗定时器方框图 .....	429
图 125. 看门狗定时器的运转 .....	431
图 126. I <sup>2</sup> C 模块方框图.....	438
图 127. START 和 STOP 条件 .....	440
图 128. 数据有效性 .....	440
图 129. 7-bit 寻址模式.....	441
图 130. 10-bit 寻址写发送模式.....	442
图 131. 10-bit 寻址读接收模式.....	442
图 132. I <sup>2</sup> C 总线确认 .....	443
图 133. 仲裁期间时钟同步 .....	444
图 134. 两个主机仲裁程序 .....	444
图 135. 主机发送时序图 .....	446
图 136. 主机接收时序图 .....	448
图 137. 从机发送时序图 .....	449
图 138. 从机接收时序图 .....	450
图 139. SCL 时序图 .....	461
图 140. SPI 方框图 .....	467
图 141. SPI 单字节数据传输时序图 – CPOL = 0, CPHA = 0 .....	469
图 142. SPI 连续数据传输时序图 – CPOL = 0, CPHA = 0 .....	470
图 143. SPI 单个字节传输时序图 – CPOL = 0, CPHA = 1 .....	470
图 144. SPI 连续传输时序图 – CPOL = 0, CPHA = 1 .....	471
图 145. SPI 单个字节传输时序图 – CPOL = 1, CPHA = 0 .....	471
图 146. SPI 连续数据传输时序图 – CPOL = 1, CPHA = 0 .....	472
图 147. SPI 单个字节传输时序图 – CPOL = 1, CPHA = 1 .....	472
图 148. SPI 连续数据传输时序图 – CPOL = 1, CPHA = 1 .....	472
图 149. SPI 多主机从机环境 .....	474
图 150. USART 方框图 .....	488
图 151. USART 串行数据格式 .....	490
图 152. USART 时钟 CK_USART 和数据帧时序 .....	491
图 153. 两个 USART 之间的硬件流控制 .....	492
图 154. USART RTS 流控制 .....	493
图 155. USART CTS 流控制 .....	493
图 156. IrDA 调制与解调 .....	494
图 157. USART I/O 和 IrDA 方框图.....	496
图 158. RS485 接口和波形 .....	497
图 159. USART 同步传输范例 .....	498
图 160. 8-bit 格式 USART 同步波形.....	499
图 161. UART 方框图.....	515
图 162. UART 串行数据格式.....	516

图 163. UART 时钟 CK_UART 和数据帧时序 .....	517
图 164. SCI 方框图 .....	526
图 165. 字符帧和补偿模式 .....	529
图 166. 保护时间 .....	530
图 167. 字符和块等待时间 – CWT 和 BWT.....	531
图 168. SCI 卡检测方框图 .....	532
图 169. SCI 中断结构 .....	534
图 170. USB 方框图 .....	548
图 171. 端点缓冲器分配范例 .....	550
图 172. 双缓冲工作范例 .....	551
图 173. PDMA 方框图 .....	577
图 174. PDMA 请求映射架构 .....	578
图 175. PDMA 通道仲裁和安排范例 .....	580
图 176. EBI 方框图 .....	596
图 177. EBI 非复用 8-bit 数据 8-bit 地址读操作.....	597
图 178. EBI 非复用 8-bit 数据 8-bit 地址写操作.....	597
图 179. EBI 非复用 16-bit 数据 N-bit 地址读操作.....	598
图 180. EBI 非复用 16-bit 数据 N-bit 地址写操作.....	598
图 181. EBI 地址锁存器设置方框图.....	599
图 182. EBI 复用 16-bit 数据 16-bit 地址读操作.....	599
图 183. EBI 复用 16-bit 数据 16-bit 地址写操作.....	600
图 184. EBI 复用 8-bit 数据 24-bit 地址读操作.....	600
图 185. EBI 复用 8-bit 数据 24-bit 地址写操作.....	601
图 186. EBI 非复用 8-bit 数据 8-bit 地址模式页读取操作.....	601
图 187. EBI 非复用 16-bit 数据 N-bit 地址模式页读取操作.....	602
图 188. EBI 复用 16-bit 数据 16-bit 地址模式页读取操作.....	602
图 189. EBI 复用 8-bit 数据 24-bit 地址模式页读取操作.....	602
图 190. EBI 页面关闭范例 .....	603
图 191. EBI 在同一个 Bank 的各个事务之间插入一个 IDLE 周期 (NOIDLE = 0) .....	604
图 192. EBI 在同一个 Bank 的各个事务之间不插入 IDLE 周期 (NOIDLE = 1) .....	604
图 193. EBI Bank 存储器映射 .....	606
图 194. I <sup>2</sup> S 方框图 .....	620
图 195. I <sup>2</sup> S 主 / 从机配置简化图 .....	621
图 196. I <sup>2</sup> S 时钟发生器方框图 .....	622
图 197. I <sup>2</sup> S 对齐立体声模式波形 .....	624
图 198. I <sup>2</sup> S 对齐立体声模式波形 ( 32-bit 通道扩展 ) .....	624
图 199. 左对齐立体声模式波形 .....	625
图 200. 左对齐立体声模式波形 ( 32-bit 通道扩展 ) .....	625
图 201. 右对齐立体声模式波形 .....	626
图 202. 右对齐立体声模式波形 ( 32-bit 通道扩展 ) .....	626
图 203. I <sup>2</sup> S 对齐单声道模式波形 .....	627

图 204. I <sup>2</sup> S 对齐单声道模式波形 ( 32-bit 通道扩展 ) .....	627
图 205. 左对齐单声道模式波形 .....	628
图 206. 左对齐单声道模式波形 ( 32-bit 通道扩展 ) .....	628
图 207. 右对齐单声道模式波形 .....	629
图 208. 右对齐单声道模式波形 ( 32-bit 通道扩展 ) .....	629
图 209. I <sup>2</sup> S 对齐重复模式波形 .....	630
图 210. I <sup>2</sup> S 对齐重复模式波形 ( 32-bit 通道扩展 ) .....	630
图 211. 各个模式下 FIFO 数据内容分配 .....	631
图 212. CRC 方框图 .....	642
图 213. CRC 数据按字节反序和按位反序范例 .....	644
图 214. AES 方框图 .....	649
图 215. AES-ECB 模式 .....	650
图 216. AES-CBC 模式 .....	651
图 217. AES-CTR 模式 .....	652
图 218. AES 中断 .....	653
图 219. CTR 模式初始矢量 .....	653
图 220. AES 字交换功能 .....	654
图 221. SDIO 总线拓扑 .....	662
图 222. SDIO 方框图 .....	662
图 223. 正常速率时序 .....	663
图 224. 高速时序 .....	663
图 225. SD_CLK 占空比周期 .....	663
图 226. “无响应”和“无数据”操作 .....	664
图 227. 多数据块读操作 .....	664
图 228. 多数据块写操作 .....	664
图 229. 命令格式 .....	665
图 230. 响应格式 .....	665
图 231. 标准总线的通用数据格式 – 仅使用 DAT0 .....	667
图 232. 宽总线的通用数据格式 – 使用 DAT0 ~ DAT3 .....	667
图 233. 标准总线的宽数据格式 – 仅使用 DAT0 .....	667
图 234. 宽总线的宽数据格式 – 使用 DAT0 ~ DAT3 .....	668
图 235. CSIF 方框图 .....	689
图 236. VSYNC 和 HSYNC 时序图 .....	691
图 237. CSIF 帧时序图 .....	691
图 238. 图像结构 .....	692
图 239. 正确窗口捕捉 .....	693
图 240. 错误的窗口捕捉 .....	694
图 241. 行和列子采样范例 .....	695

# 1 简介

## 概述

本用户手册提供的详细资料，包括如何使用该系列单片机、系统和总线结构、存储器结构和外设说明。本文档的目标读者是软件开发人员、应用开发人员和硬件开发人员。欲了解更多有关引脚配置、封装和电气特性的信息，请参考该系列单片机规格书。

该系列单片机是一款基于 Arm® Cortex®-M3 处理器内核的 32 位高性能低功耗单片机。Cortex®-M3 是把嵌套向量中断控制器 (NVIC)，SysTick 定时器和先进的调试支持紧紧结合在一起的新一代处理器内核。

该系列单片机可工作在高达 96 MHz 频率下，通过借助 Flash 加速器以获得较大的效能。它提供多达 256 KB 的嵌入式 Flash 存储器用作代码 / 数据存储，128 KB 的嵌入式 SRAM 存储器用作系统操作和应用程序运用。此单片机具有多种外设，如 ADC、I<sup>2</sup>C、USART、UART、SPI、I<sup>2</sup>S、PDMA、GPTM、MCTM、SCI、EBI、CRC-16/32、AES-128/256、USB2.0 FS、SDIO、CSIF 和 SWJ-DP (串行线 JTAG 调试端口) 等。提供了几种省电模式，在唤醒延迟和功耗方面具有较优化的灵活性，这是低功耗应用方面的考虑要点。

以上这些特性使该系列单片机可以广泛地适用于各种应用，如白色家电应用控制、电源监控和报警系统、消费类产品、手持式设备、数据记录应用、马达控制、指纹识别等。

## 特性

- 内核
  - 32-bit Arm® Cortex®-M3 处理器内核
  - 高达 96 MHz 的工作频率
  - 单周期乘法和硬件除法器
  - 内建嵌套向量中断控制器 (NVIC)
  - 24-bit SysTick 定时器
- 片上存储器
  - 高达 256 KB 片上 Flash 存储器用作指令 / 数据和选项存储
  - 128 KB 片上 SRAM
  - 支持多种启动模式
- Flash 存储器控制器 – FMC
  - Flash 加速器用以提升效能
  - 带有在线系统编程接口 (ISP) 和在线应用编程 (IAP) 的 32-bit 字编程功能
  - Flash 保护能力, 防止非法访问
- 复位控制单元 – RSTCU
  - 电源监控: 上电复位 (POR) / 掉电复位 (PDR)、欠压检测器 (BOD) 和可编程低压检测器 (LVD)
- 时钟控制单元 – CKCU
  - 外部 4 ~ 16 MHz 晶振
  - 外部 32.768 kHz 晶振
  - 在工作电压为 3.3 V, 工作温度为 25 °C 下, 内部 8 MHz RC 振荡器精度可调整为 ±2%
  - 内部 32 kHz RC 振荡器
  - 集成的系统时钟 PLL 与 USB PLL
  - 用作外设时钟源的独立的时钟分频和门控位
- 电源管理 – PWRCU
  - 单个 V<sub>DD</sub> 电源: 2.0 V ~ 3.6 V
  - 集成的 1.5 V LDO 稳压器用作 CPU 内核、外设和存储器电源
  - V<sub>BAT</sub> 电源供电给 RTC 和备份寄存器
  - 三个电源域: V<sub>DD</sub>、1.5V 和备份域
  - 四个省电模式: 休眠模式、深度休眠模式 1、深度休眠模式 2、暂停模式
- 外部中断 / 事件控制器 – EXTI
  - 高达 16 个可配置触发源和触发类型的 EXTI 输入线
  - 所有 GPIO 引脚都可选作 EXTI 触发源
  - 触发源类型包括: 高电平、低电平、下降沿、上升沿或者双沿
  - 每个 EXTI 输入线都可单独进行中断使能、唤醒和状态位设置
  - 每个 EXTI 输入线都有软件中断触发模式
  - 集成抗干扰滤波器, 用于短脉冲封锁

- 模拟数字转换器 – ADC
  - 12-bit SAR ADC 内核
  - 最大 1 Msps 转换速率
  - 最多 16 个外部模拟输入通道
  - 可选择外部参考电压输入
  - 转换范围:  $V_{REF+} \sim V_{REF-}$
- 模拟比较器 – CMP
  - Rail-to-rail 比较器
  - 每个比较器带有可配置的负端输入, 可灵活的选择电压
    - ◆ 专用 I/O 引脚
    - ◆ 内部 6-bit 计数器用于提供参考电压
  - 可编程迟滞
  - 可编程速率和功耗
  - 比较器可输出至 I/O 引脚、定时器或 ADC 触发输入端
  - 由 6-bit 计数器提供的可编程内部电压参考
  - 比较器有产生中断的能力, 通过 EXTI 控制器, 有从休眠或深度休眠模式中唤醒的功能
- 输入 / 输出 – GPIO
  - 多达 80 个通用输入 / 输出
  - 端口 A~E 映射为 16 个外部中断 – EXTI
  - 几乎所有的 I/O 引脚可配置输出驱动电流
- 马达控制定时器 – MCTM
  - 16-bit 向上、向下、向上 / 向下自动重载计数器
  - 多达 4 个独立通道
  - 16-bit 可编程预分频器, 可以对计数器时钟进行 1 ~ 65536 之间的任何数值的分频
  - 输入捕捉功能
  - 比较匹配输出
  - 带有边沿对齐模式和中心对齐计数模式的 PWM 波形发生器
  - 单脉冲输出模式
  - 带有可编程死区时间插入的互补输出
  - 支持三相马达控制和霍尔传感器接口
  - 刹车信号输入可迫使定时器的输出信号进入复位或固定状态
- 通用定时器 – GPTM
  - 16-bit 向上、向下、向上 / 向下自动重载计数器
  - 多达 4 个独立通道用于每个定时器
  - 16-bit 可编程预分频器, 可以对计数器时钟进行 1 ~ 65536 之间的任何数值的分频
  - 捕捉输入功能
  - 比较匹配输出
  - 带有边沿对齐模式和中心对齐计数模式的 PWM 波形发生器
  - 单脉冲模式输出
  - 内建正交解码器带两个输入接口

- 基础功能定时器 – BFTM
  - 32-bit 比较 / 匹配向上计数器 – 无输入 / 输出引脚
  - 单次模式 – 匹配后停止计数
  - 重复模式 – 匹配后重新开始计数
- 看门狗定时器 – WDT
  - 带 3-bit 预分频器的 12-bit 向下计数器
  - 用于系统的中断或复位事件
  - 可编程看门狗定时器窗口功能
  - 寄存器写保护功能
- 实时时钟 – RTC
  - 带可编程预分频器的 32-bit 向上计数器
  - 闹钟功能
  - 中断和唤醒事件
- 内部集成电路 – I<sup>2</sup>C
  - 支持高达 1 MHz 频率的主从模式
  - 提供仲裁功能和时钟同步
  - 支持 7-bit 和 10-bit 寻址模式和扩播呼叫寻址
  - 支持带有屏蔽地址的从机多寻址模式
- 串行外设接口 – SPI
  - 支持主机和从机模式
  - 主机模式频率高达  $f_{\text{PCLK}}/2$  MHz, 从机模式频率高达  $f_{\text{PCLK}}/3$  MHz
  - FIFO 深度: 8 级
  - 多个主机和多个从机操作
- 通用同步异步收发器 – USART
  - 同时支持异步和时钟同步串行通信模式
  - 异步工作波特率时钟频率高达  $f_{\text{PCLK}}/16$  MHz, 同步工作速率时钟频率高达  $f_{\text{PCLK}}/8$  MHz
  - 全双工通信能力
  - 完全可编程串行通信功能包括: 字长、校验位、停止位和位顺序
  - 错误检测: 奇偶校验、溢出和帧错误
  - 支持自动硬件流控制模式 – RTS、CTS
  - IrDA SIR 编码器和解码器
  - 带有输出使能控制的 RS485 模式
  - FIFO 深度: 8 级, 用于发送和接收
- 通用异步收发器 – UART
  - 异步串行通信工作波特率时钟频率高达  $f_{\text{PCLK}}/16$  MHz
  - 全双工通信能力
  - 完全可编程串行通信功能包括: 字长、校验位、停止位和位顺序
  - 错误检测: 奇偶检验、溢出和帧错误

- 智能卡接口 – SCI
  - 支持 ISO 7816-3 标准
  - 字符传输模式
  - 一个发送缓冲器和一个接收缓冲器
  - 11-bit 基本时间单元 (ETU) 计数器
  - 9-bit 保护时间计数器
  - 24-bit 通用等待时间计数器
  - 奇偶校验位生成和检测
  - 发送和接收模式下检测到奇偶校验错误时字符自动重发
- 内置音频接口 – I²S
  - 主机或从机模式
  - 单声道和立体声
  - I²S 对齐、左对齐和右对齐模式
  - 使用 32-bit 通道扩展支持 8/16/24/32-bit 采样容量
  - 8 × 32-bit TX & RX FIFO 并支持 PDMA
  - 带速度控制的 8-bit 小数时钟分频器
- 循环冗余校验 – CRC
  - 支持 CRC16 多项式:  $0x8005$ ,  $X^{16}+X^{15}+X^2+1$
  - 支持 CCITT CRC16 多项式:  $0x1021$ ,  $X^{16}+X^{12}+X^5+1$
  - 支持 IEEE-802.3 CRC32 多项式:  $0x04C11DB7$ ,  $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$
  - 支持对数据和校验和进行反码、字节反序 & 位反序操作
  - 支持字节、半字和字数据大小
  - 可编程 CRC 初始种子值
  - 对 8-bit 数据执行 CRC 计算需要 1 个 AHB 时钟周期, 32-bit 数据需要 4 个 AHB 时钟周期
  - 支持 PDMA 完成指定存储器区块的 CRC 计算
- 外设直接存储器访问 – PDMA
  - 12 通道含触发源组
  - 数据传输宽度: 8/16/32-bit
  - 支持地址递减、递增或固定模式
  - 4 级可编程通道优先级
  - 自动重载模式
  - 支持触发源:  
ADC、SPI、EBI、CRC、USART、UART、I²C、I²S、GPTM、MCTM、SCI、AES、SDIO 和软件请求
- 外部总线接口 – EBI
  - 可编程接口适用于各种存储器类型
  - 将 AHB 传输协议转换成适当的外部设备协议
  - 4 个存储器 Bank, 每一个存储器 Bank 有独立的片选控制
  - 每一个存储器 Bank 有精准的设置控制、滤波、保持和周转时序
  - 支持页模式读取
  - 当 AHB 事务宽度和外部存储器接口宽度不同时, 可自动转换
  - 写缓存器可减少因 AHB 写突发事件而停滞的状况
  - 支持复用和非复用的地址和数据线配置



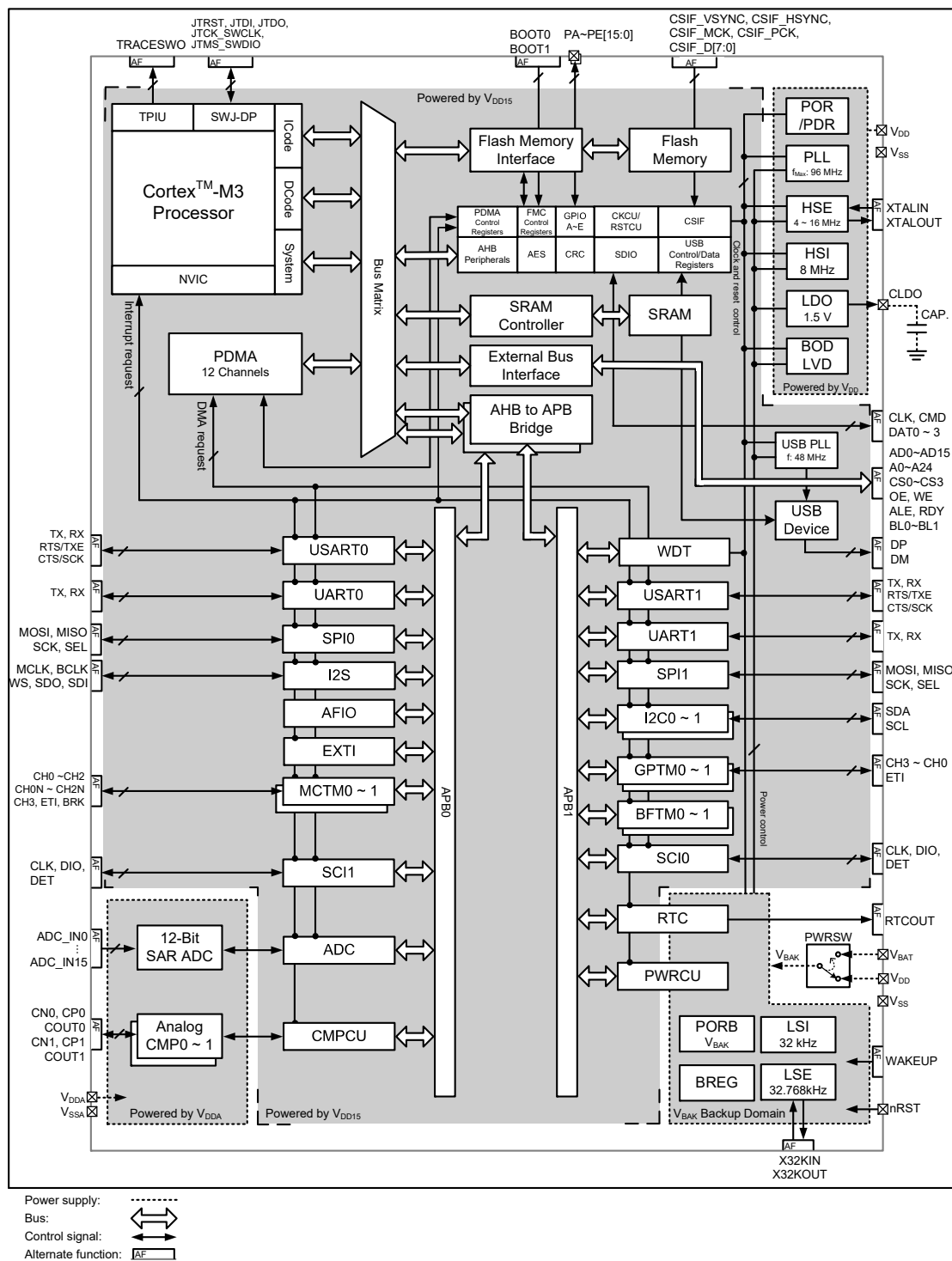
- 复用地址和数据线配置
  - ◆ 多达 25 条地址线
  - ◆ 多达 16-bit 数据总线宽度
- 通用串行总线设备控制器 – USB
  - 兼容 USB 2.0 全速模式 (12 Mbps) 设备的技术规范
  - 片上 USB 全速收发器
  - 1 个用于控制传输的控制端点 (EP0)
  - 3 个用于批量和中断传输的单缓冲端点 (EP1 ~ EP3)
  - 4 个用于批量、中断和等时传输的双缓冲端点 (EP4 ~ EP7)
  - 1 KB EP\_SRAM 作为端点数据缓冲器
- 高级加密标准 – AES
  - 支持 AES 加密 / 解密功能
  - 支持 AES ECB / CBC / CTR 三种模式
  - 支持 128 位、192 位和 256 位密钥长度
  - 支持 4 字长度的初始矢量用于 CBC 和 CTR 模式
  - 8 × 32 位 (输入 / 输出 FIFO) 用于 2 个 AES 数据块
  - 支持按字的数据交换功能
  - 支持 PDMA 接口
- 安全数字输入 / 输出 – SDIO
  - 支持两种不同的数据总线模式：1-bit (默认模式) 和 4-bit
  - 支持两种不同的速度模式：正常速度模式 (默认模式) 和高速模式
  - 高达系统频率的 SD 时钟频率
  - 不支持 SPI 模式和 MMC 串流模式
- CMOS 传感器接口 – CSIF, 仅 HT32F22366
  - 高达 2048 × 2048 输入解析度
  - 支持 8-bit YUV422 和 Raw RGB 格式
  - 高达  $f_{HCLK}/3$  输入像素时钟频率
  - 多种 VSYNC 和 HSYNC 设置用于图像捕捉
  - 硬件窗口捕捉功能
  - 分级硬件子采样功能
  - 两个 8 × 32 位 FIFO, 可通过 PDMA 或 CPU 读取
- 调试支持
  - 串行线 JTAG 调试端口 – SWJ-DP
  - 6 个指令比较器和 2 个内部比较器用于硬件断点或程序 / 文字补丁
  - 1-bit 异步追踪用于串行线调试模式 – TRACESWO
- 封装和工作温度
  - 46-pin QFN, 48/64/100-pin LQFP 封装
  - 工作温度范围: -40 °C ~ +85 °C

## 单片机信息

表 1. 单片机的特性及外设列表

外设		HT32F12365	HT32F12366	HT32F22366
主 Flash (KB)		255	255	255
选项字节 Flash (KB)		1	1	1
SRAM (KB)		64	128	128
定时器	MCTM	2		
	GPTM	2		
	BFTM	2		
	RTC	1		
	WDT	1		
通信	CSIF	—	—	1
	USB	1		
	SCI	2		
	SPI	2		
	USART	2		
	UART	2		
	I <sup>2</sup> C	2		
	I <sup>2</sup> S	1		
PDMA		12 通道		
AES		1		
SDIO		1		
EBI		1		
CRC		1		
GPIO		80 (Max.)		
EXTI		16		
12-bit ADC		1		
通道总数		16 通道 (Max.)		
比较器		2		
CPU 频率		高达 96 MHz		
工作电压		2.0 V ~ 3.6 V		
工作温度		-40 °C ~ +85 °C		
封装		46-pin QFN, 48/64/100-pin LQFP		

## 方框图



注：CSIF 为 AHB 外设功能，仅 HT32F22366 支持。

图 1. 方框图

## 2 文档协议

本文档中使用的协议列于下表。

表 2. 文档协议

符号	范例	描述
0x	0x5a05	带 0x 前缀的数字字符串表示十六进制数。
0xnnnn_nnnn	0x2000_0100	32-bit 十六进制地址或数据。
b	b0101	带小写字母 b 前缀的数字字符串表示二进制数。
NAME [n]	ADDR [5]	NAME 的特定位。NAME 可以是寄存器或寄存器字段。例如，ADDR [5] 指的是 ADDR 寄存器 ( 字段 ) 的第 5 位。
NAME [m:n]	ADDR [11:5]	NAME 的特定位。NAME 可以是寄存器或寄存器字段。例如，ADDR [11:5] 指的是 ADDR 寄存器 ( 字段 ) 的第 11 位 ~ 第 5 位。
X	b10X1	不用在意的符号，它表示可以允许的任何值。
RW	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">             19  <div style="border: 1px solid black; padding: 2px;">SERDYIE</div>             RW 0           </div> <div style="text-align: center;">             18  <div style="border: 1px solid black; padding: 2px;">PLLRDYIE</div>             RW 0           </div> </div>	软件可以读取或写入该位。
RO	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">             3  <div style="border: 1px solid black; padding: 2px;">HSIRDY</div>             RO 1           </div> <div style="text-align: center;">             2  <div style="border: 1px solid black; padding: 2px;">HSERDY</div>             RO 0           </div> </div>	软件只能读该位。写操作将不会有任何效果。
RC	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">             1  <div style="border: 1px solid black; padding: 2px;">PDF</div>             RC 0           </div> <div style="text-align: center;">             0  <div style="border: 1px solid black; padding: 2px;">BAK_PORF</div>             RC 1           </div> </div>	软件只能读该位。读操作将自动清零该位。
WC	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">             3  <div style="border: 1px solid black; padding: 2px;">SERDYF</div>             WC 0           </div> <div style="text-align: center;">             2  <div style="border: 1px solid black; padding: 2px;">PLLRDYF</div>             WC 0           </div> </div>	软件可以读该位或通过写 1 来清除它。写 0 将不会有任何效果。
W0C	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">             1  <div style="border: 1px solid black; padding: 2px;">RXCF</div>             RO 0           </div> <div style="text-align: center;">             0  <div style="border: 1px solid black; padding: 2px;">PARF</div>             W0C 1           </div> </div>	软件可以读该位或通过写 0 来清除它。写 1 将不会有任何效果。
WO	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">             31  <div style="border: 1px solid black; padding: 2px;">DB_CKSRC</div>             WO 0           </div> <div style="text-align: center;">             30  <div style="border: 1px solid black; padding: 2px;">DB_CKSRC</div>             WO 0           </div> </div>	软件只能写该位。读操作将总是返回 0。
保留位	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">             1  <div style="border: 1px solid black; padding: 2px;">LLRDY</div>             RO 0           </div> <div style="text-align: center;">             0  <div style="border: 1px solid black; padding: 2px;">Reserved</div> </div> </div>	保留位供将来使用。从这些位中读取的数据没有很好地被定义，应被视为随机数据处理。通常情况下，这些保留位应设置为 0。需要注意的是保留位必须保持在复位值。
Word		一个字的数据长度为 32-bit。
Half-word		一个半字的数据长度为 16-bit。
Byte		一个字节的数据长度为 8-bit。

## 3 系统结构

该系列单片机的系统结构包括 Arm® Cortex®-M3 处理器、总线结构和存储器体系，将在以下各节中描述。Cortex®-M3 处理器是新一代的处理器内核，提供了许多新的特性。集成和先进的特性使得 Cortex®-M3 处理器适合于需要高性能和低功耗单片机的市场产品。简而言之，Cortex®-M3 处理器包含三条 AHB-Lite 总线，即 ICode 总线、DCode 总线和系统总线。根据不同的目的和目标存储空间，Cortex®-M3 处理器访问所有存储器都是在这三条总线上执行的。存储器体系采用了哈佛结构，预先定义的存储器映射和高达 4 GB 的存储空间，使系统灵活和可扩展。

### Arm® Cortex®-M3 处理器

Cortex®-M3 处理器是一种通用的 32-bit 处理器内核，特别适合于要求高性能和低功耗单片机的产品。它提供了许多新的特性，如 Thumb-2 指令集、硬件除法器、低延迟中断响应时间、原子位带访问和适用于同时访问的多条总线。Cortex®-M3 处理器基于 ARMv7 结构，同时支持 Thumb 和 Thumb-2 指令集。下面列出了一些也是由 Cortex®-M3 提供的系统外设：

- 内部总线矩阵连接的 ICode 总线、DCode 总线、系统总线、专用外设总线 (PPB) 和调试访问 (AHB-AP)
- 嵌套向量中断控制器 (NVIC)
- Flash 补丁和断点 (FPB)
- 数据观察点及追踪 (DWT)
- 仪器跟踪宏单元 (ITM)
- 存储器保护单元 (MPU)
- 串行线 JTAG 调试端口 (SWJ-DP)
- 嵌入式跟踪宏单元 (ETM)
- 跟踪端口接口单元 (TPIU)

下图为 Cortex®-M3 处理器的方框图。欲了解更多信息，请参考 Arm® Cortex®-M3 技术参考手册。

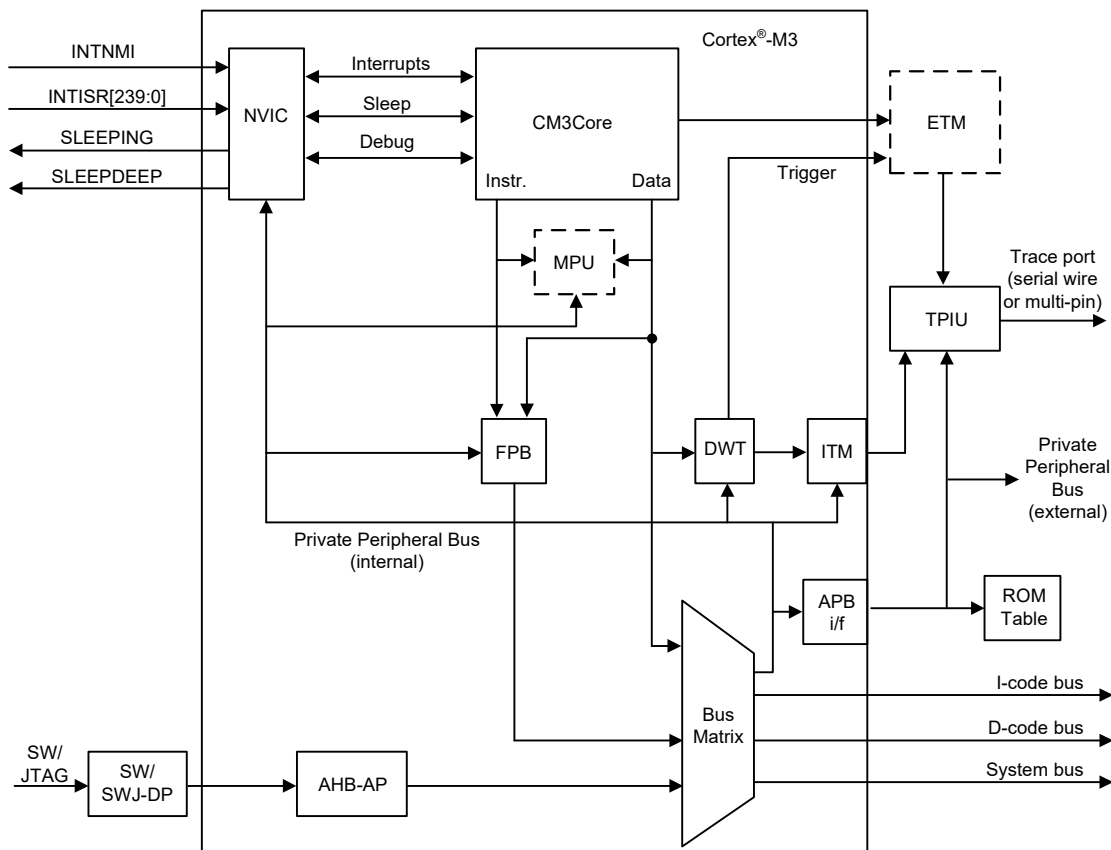


图 2. Cortex®-M3 方框图

## 总线结构

该系列单片机的总线结构包括四个主机总线和六个从机总线。Cortex®-M3 ICode、DCode、系统总线和外设直接存储器访问 (PDMA) 为主机总线，而内部 SRAM 访问总线、内部 Flash 存储器访问总线、AHB 外设访问总线、外部总线接口和两个 AHB 到 APB 总线桥为从机总线。ICode 总线用于从代码区域 (0x0000\_0000 ~ 0x1FFF\_FFFF) 取指令和向量到 Cortex®-M3 内核。DCode 总线用于对代码区域进行数据加载 / 存储以及调试访问。同样地，系统总线用于对系统区域执行取指令和向量、数据加载 / 存储以及调试访问。系统区域包括内部 SRAM 区域和外设区域。主机总线都是基于 32-bit 先进的高性能总线精简 (AHB-Lite) 协议。下图显示了该系列单片机的总线结构。

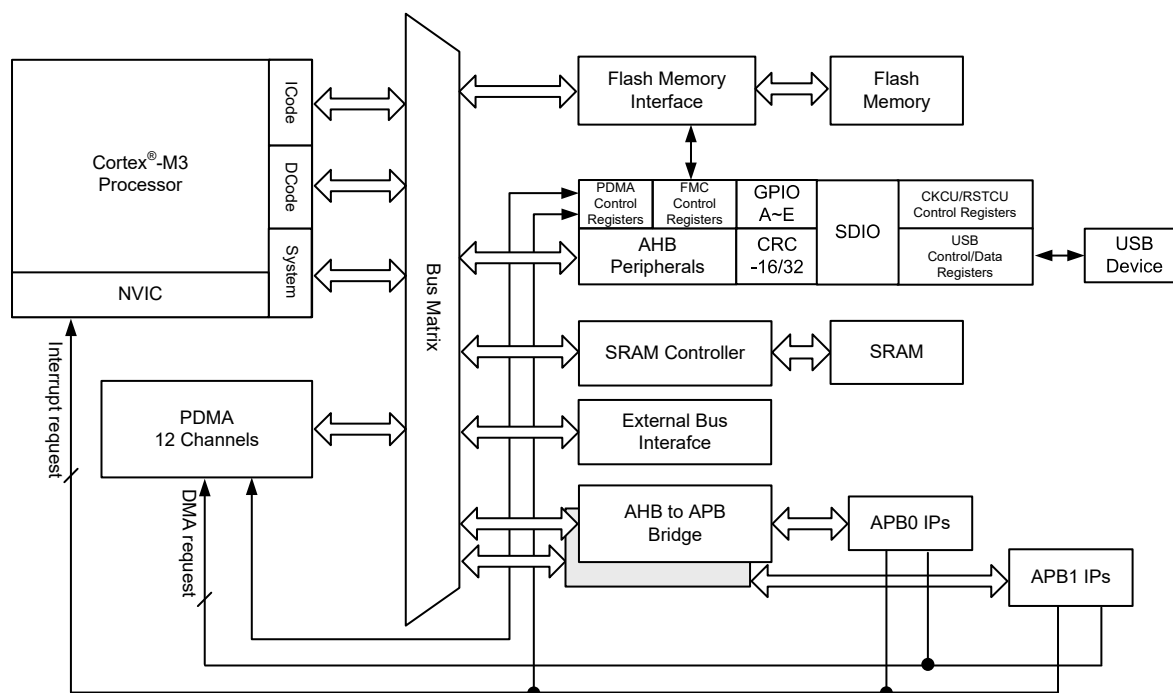


图 3. 总线结构

## 存储器体系

Arm® Cortex®-M3 处理器的结构是哈佛结构，可以使用不同的总线进行取指令和加载 / 存储数据。指令代码和数据都位于相同的存储地址空间，但在不同的地址范围。Cortex®-M3 的最大地址范围是 4 GB，因为它具有 32-bit 总线地址宽度。此外，预先定义的存储器映射由 Cortex®-M3 处理器提供，以减少软件被不同的单片机供应商重复实施的复杂性。但是，有些区域由 Arm® Cortex®-M3 系统外设使用。更多信息请参考 Arm® Cortex®-M3 技术参考手册。下图显示了该系列单片机的存储器映射，包括代码、SRAM、外设和其它预定义区域。

## 存储器映射

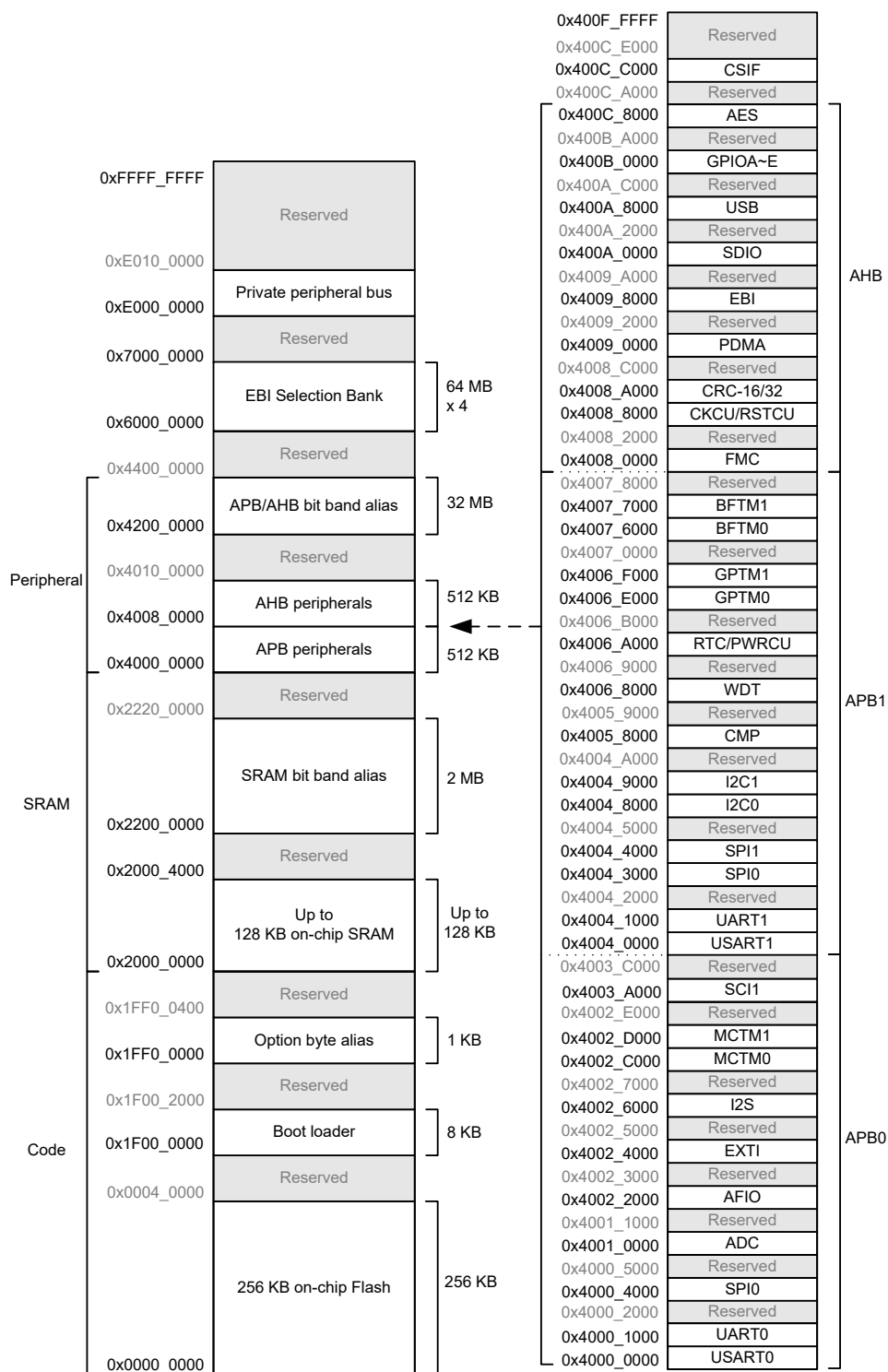


图 4. 存储器映射



表 3. 寄存器列表

起始地址	结束地址	外设	总线
0x4000_0000	0x4000_0FFF	USART0	APB0
0x4000_1000	0x4000_1FFF	UART0	
0x4000_2000	0x4000_3FFF	保留	
0x4000_4000	0x4000_4FFF	SPI0	
0x4000_5000	0x4000_5FFF	保留	
0x4001_0000	0x4001_0FFF	ADC	
0x4001_1000	0x4002_1FFF	保留	
0x4002_2000	0x4002_2FFF	AFIO	
0x4002_3000	0x4002_3FFF	保留	
0x4002_4000	0x4002_4FFF	EXTI	
0x4002_5000	0x4002_5FFF	保留	
0x4002_6000	0x4002_6FFF	I <sup>2</sup> S	
0x4002_7000	0x4002_7FFF	保留	
0x4002_C000	0x4002_CFFF	MCTM0	
0x4002_D000	0x4002_DFFF	MCTM1	
0x4002_E000	0x4003_9FFF	保留	
0x4003_A000	0x4003_BFFF	SCI1	
0x4003_C000	0x4003_FFFF	保留	
0x4004_0000	0x4004_0FFF	USART1	APB1
0x4004_1000	0x4004_1FFF	UART1	
0x4004_2000	0x4004_2FFF	保留	
0x4004_3000	0x4004_3FFF	SCI0	
0x4004_4000	0x4004_4FFF	SPI1	
0x4004_5000	0x4004_7FFF	保留	
0x4004_8000	0x4004_8FFF	I <sup>2</sup> C0	
0x4004_9000	0x4004_9FFF	I <sup>2</sup> C1	
0x4004_A000	0x4005_7FFF	保留	
0x4005_8000	0x4005_8FFF	CMP	
0x4005_9000	0x4006_7FFF	保留	
0x4006_8000	0x4006_8FFF	WDT	
0x4006_9000	0x4006_9FFF	保留	
0x4006_A000	0x4006_AFFF	RTC/PWRCU	
0x4006_B000	0x4006_DFFF	保留	
0x4006_E000	0x4006_EFFF	GPTM0	
0x4006_F000	0x4006_FFFF	GPTM1	
0x4007_0000	0x4007_5FFF	保留	
0x4007_6000	0x4007_6FFF	BFTM0	
0x4007_7000	0x4007_7FFF	BFTM1	
0x4007_8000	0x4007_FFFF	保留	

起始地址	结束地址	外设	总线
0x4008_0000	0x4008_1FFF	FMC	AHB
0x4008_2000	0x4008_7FFF	保留	
0x4008_8000	0x4008_9FFF	CKCU/RSTCU	
0x4008_A000	0x4008_BFFF	CRC	
0x4008_C000	0x4008_FFFF	保留	
0x4009_0000	0x4009_1FFF	PDMA 控制寄存器	
0x4009_2000	0x4009_7FFF	保留	
0x4009_8000	0x4009_9FFF	EBI 控制寄存器	
0x4009_A000	0x4009_FFFF	保留	
0x400A_0000	0x400A_1FFF	SDIO	
0x400A_2000	0x400A_7FFF	保留	
0x400A_8000	0x400A_9FFF	USB 控制寄存器	
0x400A_A000	0x400A_BFFF	EP_SRAM	
0x400A_C000	0x400A_FFFF	保留	
0x400B_0000	0x400B_1FFF	GPIOA	
0x400B_2000	0x400B_3FFF	GPIOB	
0x400B_4000	0x400B_5FFF	GPIOC	
0x400B_6000	0x400B_7FFF	GPIOD	
0x400B_8000	0x400B_9FFF	GPIOE	
0x400B_A000	0x400C_7FFF	保留	
0x400C_8000	0x400C_9FFF	AES	
0x400C_A000	0x400C_BFFF	保留	
0x400C_C000	0x400C_DFFF	CSIF	
0x400C_E000	0x400F_FFFF	保留	

## 嵌入式 Flash 存储器

该系列单片机提供高达 256 KB 片上 Flash 存储器，位于地址 0x0000\_0000。它支持字节、半字和字访问操作。注意，对于 Cortex®-M3 ICode 或 DCode 总线访问，Flash 存储器仅支持读操作。任何对 Flash 存储器的写操作（通过 DCode 总线）将导致总线故障发生。该 Flash 存储器容量高达 256 页，每一页都有 1 KB 的存储容量，并且可以单独擦除。32-bit 可编程接口拥有使位从 1 到 0 改变的能力。数据存储或固件升级可以通过使用在线系统编程 (ISP)、在线应用编程 (IAP) 或在线烧录 (ICP) 来实现。欲了解更多信息，请参考 Flash 存储器控制器章节。

## 嵌入式 SRAM 存储器

该系列单片机包含 128 KB 片上 SRAM 存储器，位于地址 0x2000\_0000。它支持字节、半字和字访问操作。为了减少读 - 修改 - 写操作的时间，Cortex®-M3 处理器提供一个位带功能来执行单个原子位操作。用户可以通过访问相应的位带别名来修改在 SRAM 位带区的单个位。有关位带的更多信息，请参考 Arm® Cortex®-M3 技术参考手册。下面的公式和例子说明了如何通过计算位带别名来访问位带区中的一个位。

$$\text{位带别名} = \text{位带基址} + (\text{字节偏移量} \times 32) + (\text{位序号} \times 4)$$

例如，要访问地址 0x2000\_0200 的第 7 位，位带别名为：

$$\text{位带别名} = 0x2200\_0000 + (0x200 \times 32) + (7 \times 4) = 0x2200\_401C$$

写数据到地址 0x2200\_401C 将导致地址 0x2000\_0200 的第 7 位改变，而读取地址 0x2200\_401C 的数据，根据 SRAM 地址 0x2000\_0200 第 7 位的值将返回 0x01 或 0x00。

## AHB 外设

AHB 外设的地址范围从 0x4008\_0000 到 0x400F\_FFFF。一些外设如时钟控制单元、复位控制单元和 Flash 存储器控制器直接连接到 AHB 总线。系统复位后，AHB 外设时钟总是开启的。可以通过 AHB 总线直接访问这些外设寄存器。注意，所有 AHB 总线上的外设寄存器只支持字访问。

## APB 外设

APB 外设的地址范围从 0x4000\_0000 到 0x4007\_FFFF。APB-AHB 总线桥在 Cortex®-M3 和 APB 外设之间有进行访问的能力。此外，系统复位后，APB 外设时钟关闭。在访问相应的外设寄存器之前，软件必须通过设置时钟控制单元中的 APBCCRn 寄存器打开外设时钟。需要注意的是，当对 APB 外设寄存器进行半字或字节访问时，APB-AHB 总线桥会将半字或字节数据增加到一个字的宽度。换言之，对 APB 外设寄存器进行半字或字节访问的结果会因访问的外设寄存器数据位宽度而有所不同。

## 4 Flash 存储器控制器 (FMC)

### 简介

Flash 存储器控制器 FMC 为嵌入式片上 Flash 存储器提供所有 Flash 操作和预取缓存器功能。下图显示了 FMC 的方框图, 包括烧录接口、控制寄存器、预取缓存器和访问接口。由于 Flash 存储器访问速度比 CPU 慢, 故提供一个带有预取缓存器的宽访问接口以减少 CPU 的等待时间, 从而避免引起指令执行延迟。Flash 存储器还提供字编程 / 页擦除功能来实现指令与数据的储存。

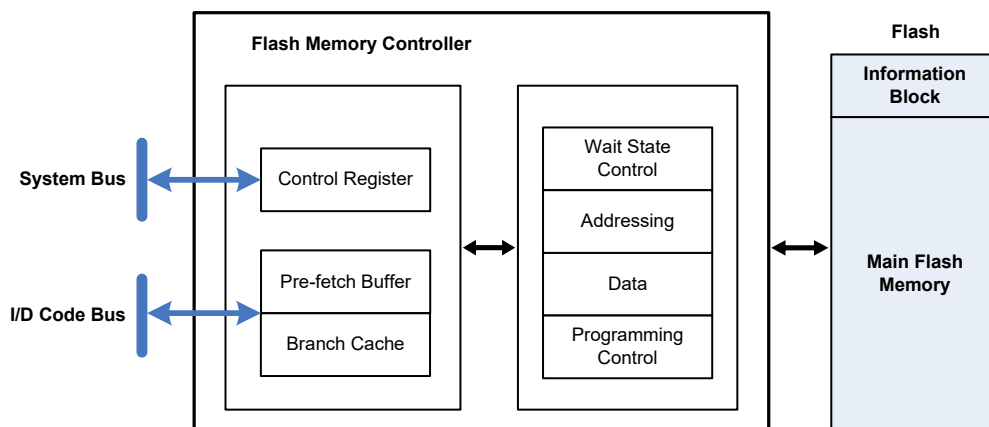


图 5. Flash 存储器控制器方框图

### 特性

- 多达 256 KB 片上 Flash 存储器用于存储指令 / 数据和选项字节
  - 256 KB: 255 KB + 1 KB (指令 / 数据 + 选项字节)
- 页大小为 1 KB, 总计多达 255 页
- 带有预取缓存器和分支缓存访问接口, 以减少指令间隙
- 页擦除和整片擦除
- 32-bit 字编程
- 中断功能用来指示 Flash 存储器运行结束或错误发生
- Flash 读保护, 防止非法代码 / 数据访问
- 页擦除 / 编程保护, 防止操作不慎毁坏数据

功能描述

Flash 存储器映射

下图是单片机地址从 0x0000\_0000 到 0x1FFF\_FFFF (0.5 GB) 的 Flash 存储器映射。0x1F00\_0000 ~ 0x1F00\_1FFF 的地址被映射到启动加载块 (8 KB)。此外，从 0x1FF0\_0000 到 0x1FF0\_03FF 的地址区域是选项字节块 (1 KB)，位于主 Flash 物理地址的最后一页。在系统视图下的存储器映射如下图所示。

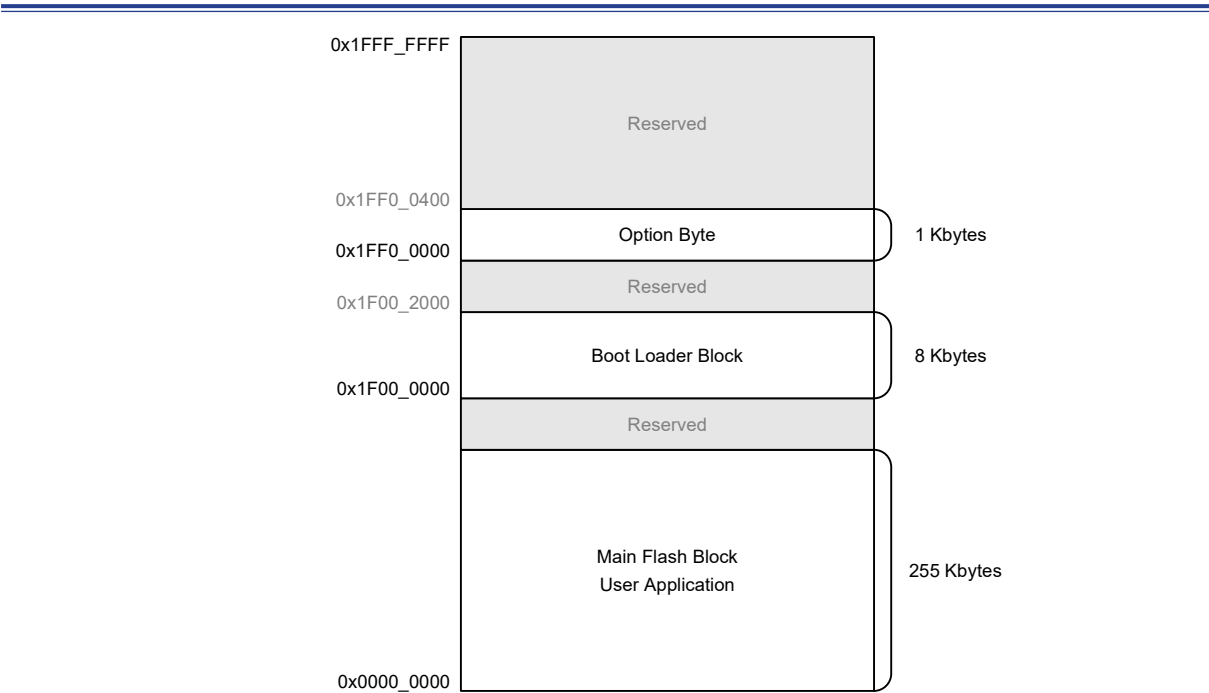


图 6. Flash 存储器映射

## Flash 存储器结构

Flash 存储器由高达 256 KB 主 Flash 存储块 (每页 1 KB) 和用于启动加载器的 8 KB 信息块组成。主 Flash 存储块总共包含 256 可单独擦除的页。下表显示了每页的基址、大小和保护设置。

表 4. Flash 存储器和选项字节

区块	名称	地址	页保护位	大小
主 Flash 块	Page 0	0x0000_0000 ~ 0x0000_03FF	OB_PP [0]	1 KB
	Page 1	0x0000_0400 ~ 0x0000_07FF		1 KB
	Page 2	0x0000_0800 ~ 0x0000_0BFF	OB_PP [1]	1 KB
	Page 3	0x0000_0C00 ~ 0x0000_0FFF		1 KB
	⋮	⋮	⋮	⋮
	Page 252	0x0003_F000 ~ 0x0003_F3FF	OB_PP [126]	1 KB
	Page 253	0x0003_F400 ~ 0x0003_F7FF		1 KB
	Page 254	0x0003_F800 ~ 0x0003_FBFF	OB_PP [127]	1 KB
	Page 255 (选项字节)	物理地址: 0x0003_FC00 ~ 0x0003_FFFF 别名地址: 0x1FF0_0000 ~ 0x1FF0_03FF	OB_CP [1]	1 KB
信息块	Boot Loader	0x1F00_0000 ~ 0x1F00_1FFF	NA	8 KB

注: 1. 该信息块存储启动加载器 – 此块不能由用户编程或擦除。

2. 选项字节始终位于主 Flash 块的最后一页。

## 等待状态设置

当 CPU 时钟 HCLK 比 Flash 存储器的访问速度快时, 必须在 CPU 取指令或从 Flash 存储器加载数据时插入等待状态周期。通过设置 Flash 缓存和预取控制寄存器 CFCR 中的 WAIT[2:0], 可以改变等待状态。为了满足等待状态的需要, 以下两条规则应该予以考虑。

- 从低频到高频切换 HCLK 时钟:  
更改等待状态设置, 然后再切换 HCLK 时钟。
- 从高频到低频切换 HCLK 时钟:  
切换 HCLK 时钟, 然后再更改等待状态设置。

下表显示了等待状态周期和 CPU 时钟 HCLK 之间的关系。默认的等待状态为 0, 因为系统复位后 HSI (8 MHz) 被选择作为 HCLK 的时钟源。

表 5. 等待状态周期与 HCLK 之间的关系

等待状态周期	HCLK
0	0 MHz < HCLK ≤ 24 MHz
1	24 MHz < HCLK ≤ 48 MHz
2	48 MHz < HCLK ≤ 72 MHz
3	72 MHz < HCLK ≤ 96 MHz

启动配置

通过使用 BOOT0 ~ BOOT1 引脚，该系列单片机可以选择三种启动模式。BOOT0 ~ BOOT1 引脚在上电复位或系统复位期间采样。一旦这些引脚上的逻辑值确定，根据启动模式，向量的前 4 个字将被重映射到相应的启动区域。下表列出了启动模式。

表 6. 启动模式

启动模式选择引脚		模式	描述
BOOT1	BOOT0		
0	0	SRAM	SBVT0 ~ SBVT3 被选为启动区域
0	1	启动加载器	启动加载器被选为启动区域
1	X	主 Flash	主 Flash 存储器被选为启动区域

向量映射控制寄存器 VMCR 用来在芯片复位后临时改变向量重映射设置。VMCR 的初始复位值是由将在复位时被采样的 BOOT0 ~ BOOT1 引脚定义的。

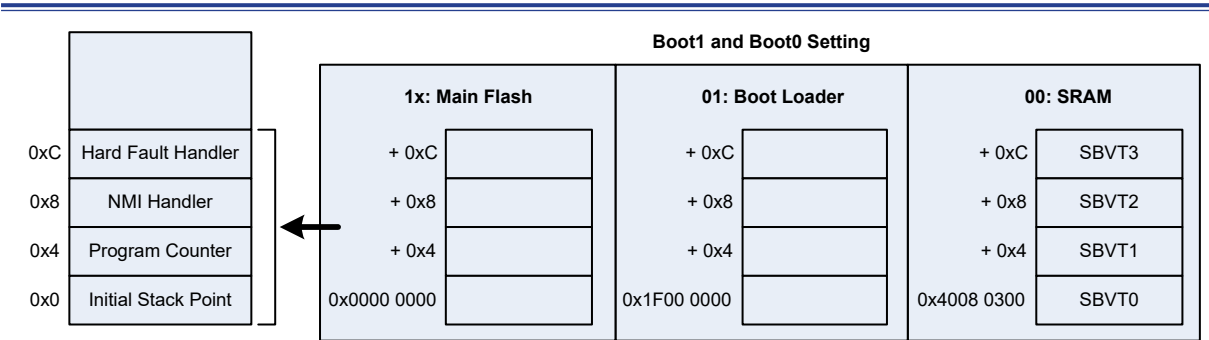


图 7. 向量重映射

## 页擦除

FMC 提供了页擦除功能，用来复位 Flash 存储器的部分内容。每一页可以单独擦除而不影响其它页的内容。下列步骤显示了用于页擦除的寄存器的访问顺序。

- 检查 OPCR 寄存器，以确认没有正在执行 Flash 存储器操作 (OPM [3:0] = 0xE 或 0x6)。否则需一直等待直到前一操作已经完成为止。
- 写入页地址到 TADR 寄存器。
- 写入页擦除命令到 OCMR 寄存器 (CMD [3:0] = 0x8)。
- 通过设置 OPCR 寄存器，发送页擦除命令到 FMC (OPM [3:0] = 0xA)。
- 等待，直到通过检查 OPCR 寄存器的值确认所有操作已经完成 (OPM [3:0] = 0xE)。
- 依据需要，通过访问 DCODE 来读取并验证 Flash 存储器的内容。

需要注意的是，必须确定正确的目标页地址。如果目标擦除页正进行程序代码读取或数据读取，软件可能会运行失控。如果发生这种情况，FMC 将不提供任何通知。此外，对于受保护的页，页擦除操作将被忽略。如果 OIER 寄存器中的 OREIEN 位被置位，FMC 将触发 Flash 操作错误中断。软件可以通过在中断处理程序中检查 OISR 寄存器中的 PPEF 位来检测此情况。下图显示了页擦除操作流程。

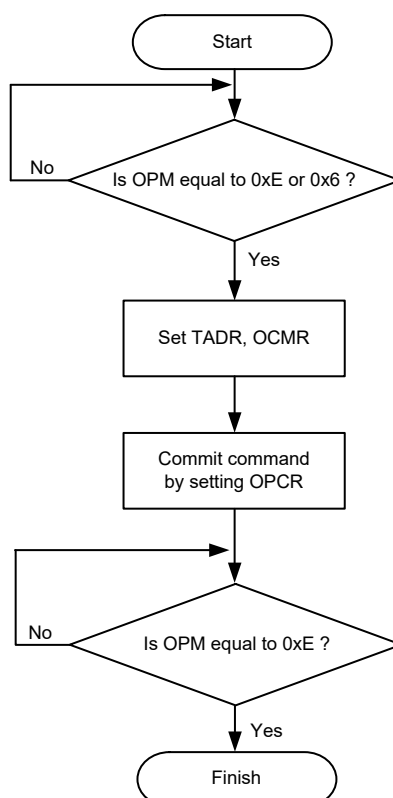


图 8. 页擦除操作流程图



## 整片擦除

FMC 提供了一个整片擦除功能, 用来重设整个 Flash 存储器的内容为高状态。下列步骤显示了整片擦除寄存器的访问顺序。

- 检查 OPCR 寄存器, 以确认没有正在执行 Flash 存储器操作 (OPM [3:0] = 0xE 或 0x6)。否则需一直等待直到 Flash 存储器操作已经完成为止。
- 写入整片擦除命令到 OCMR 寄存器 (CMD [3:0] = 0xA)。
- 通过设置 OPCR 寄存器, 发送整片擦除命令到 FMC (OPM [3:0] = 0xA)。
- 等待, 直到通过检查 OPCR 寄存器的值确认所有操作已经完成 (OPM [3:0] = 0xE)。
- 依据需要, 通过访问 DCODE 来读取和验证 Flash 存储器的内容。

由于所有的 Flash 数据将被复位到 0xFFFF\_FFFF, 使用在 SRAM 上运行的程序或使用直接访问 FMC 寄存器的调试工具, 可以实现整片擦除操作。在 Flash 存储器上所执行的软件功能将无法达成整片擦除操作。下图显示了整片擦除操作流程。

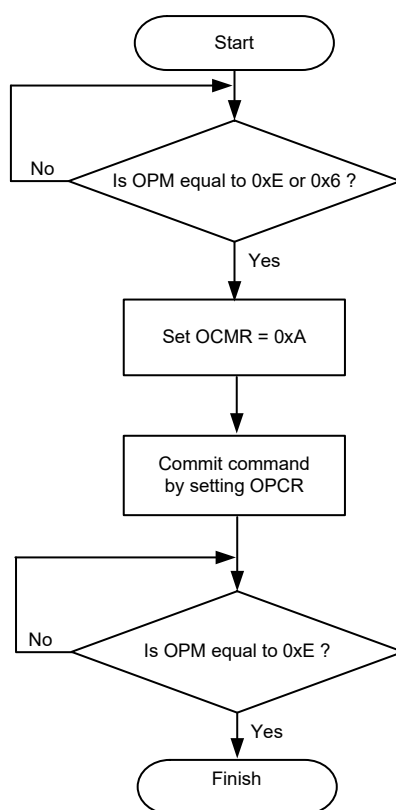


图 9. 整片擦除操作流程

## 字编程

FMC 提供了 32-bit 字编程功能，用于修改 Flash 存储器的内容。下列步骤显示了字编程操作寄存器的访问顺序。

- 检查 OPCR 寄存器，以确认没有正在执行 Flash 存储器操作 (OPM [3:0] = 0xE 或 0x6)。否则需一直等待直到前面的操作已经完成为止。
- 写入字地址到 TADR 寄存器。写入数据到 WRDR 寄存器。
- 写入字编程命令到 OCMR 寄存器 (CMD [3:0] = 0x4)。
- 通过设置 OPCR 寄存器，发送字编程命令到 FMC (OPM [3:0] = 0xA)。
- 等待，直到通过检查 OPCR 寄存器的值确认所有操作已经完成 (OPM [3:0] = 0xE)。
- 依据需要，通过访问 DCODE 来读取和验证 Flash 存储器的内容。

需要注意的是，不能对相同的地址连续两次进行字编程操作。对相同地址连续进行的字编程操作必须通过页擦除操作分开。此外，对于受保护的页，字编程操作将被忽略。如果 OIER 寄存器中的 OREIEN 位被置位，FMC 将触发 Flash 操作错误中断。软件可以通过在中断处理程序中检查 OISR 寄存器中的 PPEF 位来检测此情况。下图显示了字编程操作流程。

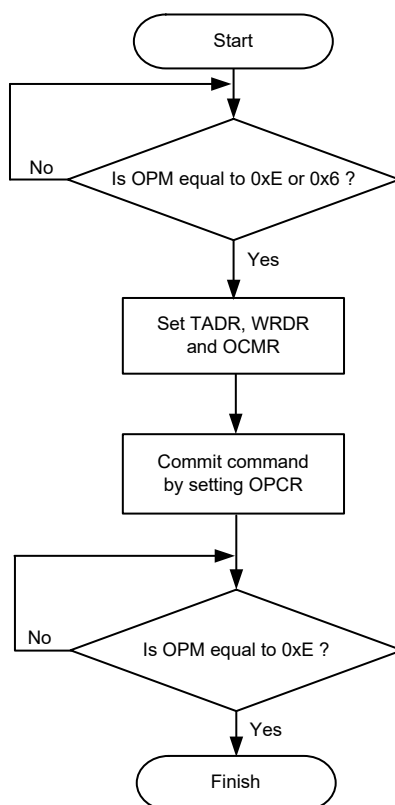


图 10. 字编程操作流程图

选项字节描述

选项字节可视为一个基址为 0x1FF0\_0000 的独立的 Flash 存储器。下表显示了功能描述和选项字节的存储器映射。

表 7. 选项字节的存储器映射

选项字节	偏移量	描述	复位值
选项字节基址 = 0x1FF0_0000			
OB_PP	0x000	OB_PP [n]: 主 Flash 页擦除 / 编程保护 (n = 0 ~ 127: 页 254 ~ 页 0) 0: Flash 页 2n 和 2n+1 擦除 / 编程保护使能 1: Flash 页 2n 和 2n+1 擦除 / 编程保护除能	0xFFFF_FFFF
	0x004		0xFFFF_FFFF
	0x008		0xFFFF_FFFF
	0x00C		0xFFFF_FFFF
OB_CP	0x010	OB_CP [0]: Flash 安全保护 0: Flash 安全保护使能 1: Flash 安全保护除能	0xFFFF_FFFF
		OB_CP [1]: 选项字节保护 0: 选项字节保护使能 1: 选项字节保护除能 OB_CP [31:2]: 保留位	
OB_CK	0x020	OB_CK [31:0]: 选项字节校验和 当 OB_PP 或 OB_CP 寄存器的内容不等于 0xFFFF_FFFF 时, OB_CK 应设置为 OB_PP 及 OB_CP 选项字节 5 个寄存器 内容的总和, 其地址偏移量范围在选项字节中从 0x000 到 0x010 (0x000 + 0x004 + 0x008 + 0x00C + 0x010)。否则, 页 擦除 / 程序保护和安全保护将使能	0xFFFF_FFFF

页擦除 / 编程保护

FMC 提供了页擦除 / 编程保护, 以防止 Flash 存储器上的误操作。FMC 受保护页不接受页擦除命令或字编程命令, 当向 FMC 发送对受保护页的页擦除或字编程命令时 OISR 寄存器中的 PPEF 位将被 FMC 置位。如果 OIER 寄存器的 OREIEN 位也被置位, 那么 FMC 将触发 Flash 操作错误中断。通过配置选项字节中的 OB\_PP [127:0] 位段, 可以单独地为每一页使能保护功能。下表显示了页保护使能时主 Flash 页的访问权限。

表 8. 受保护主 Flash 页的访问权限

模式 操作	ISP / IAP	ICP / 调试模式	从 SRAM 启动
DCode 读操作	O	O	O
编程	X	X	X
页擦除	X	X	X
整片擦除	O	O	O

- 注：1. 注意写保护设置是建立在页的基础上。上述访问只对使能保护功能的页有效。其它页无效。  
2. 主 flash 页保护由 OB\_PP[127:0] 配置。选项字节位于主 Flash 的最后一页。选项字节页保护由 OB\_CP[1] 位设置。  
3. 选项字节区域中的页擦除可除能主 Flash 的页保护。  
4. 选项字节的页保护只能由整片擦除操作除能。

下列步骤显示了页擦除 / 编程保护程序寄存器的访问顺序：

- 检查 OPCR 寄存器，以确认没有正在执行 Flash 存储器操作 (OPM [3:0] = 0xE 或 0x6)。否则，等待直到前面的操作已经完成。
- 将 OB\_PP 地址写到 TADR 寄存器 (TADR = 0x1FF0\_0000)。
- 将数据写到 WRDR 寄存器，定义相应的页保护功能是使能还是除能 (0: 使能, 1: 除能)。
- 将字编程命令写到 OCMR 寄存器 (CMD [3:0] = 0x4)。
- 通过设置 OPCR 寄存器，将字编程命令发送到 FMC (OPM [3:0] = 0xA)。
- 等待，直到通过检查 OPCR 寄存器的值确认所有操作已经完成 (OPM [3:0] = 0xE)。
- 依据需要，通过访问 DCode 来读取和验证操作字节。
- 根据选项字节校验和规则，编程 OB\_CK 使选项字节为 5 个字 0x000 ~ 0x010 的总和。
- 使系统复位以激活新的 OB\_PP 设置。

## 安全保护

FMC 提供了安全保护功能，防止 Flash 存储器上的非法代码 / 数据访问。该功能对保护软件 / 固件不受非法使用者侵害非常有用。通过设置选项字节 OB\_CP [0] 来激活此功能。一旦该功能被使能，除了用户的应用程序，所有主 Flash 的 DCODE 访问，编程和页擦除的数据访问将不会被允许。但是，为了能解除安全保护功能，整片擦除操作仍将被 FMC 接受。下表显示了安全保护使能时 Flash 存储器的访问权限。

表 9. 安全保护使能时的访问权限

操作 \ 模式	用户应用程序 <sup>(1)</sup>	ICP / 调试模式	从 SRAM 启动
读操作	O	X (读为 0)	X (读为 0)
编程	O <sup>(1)</sup>	X	X
页擦除	O <sup>(1)</sup>	X	X
整片擦除	O	O	O

注：1. 用户应用程序是指软件被执行或从断开 JTAG / SW 调试器的主 Flash 存储器启动。然而，选项字节块和第 0 页仍然受保护，不能编程 / 页擦除。  
2. 整片擦除操作可以擦除选项字节块和除能安全保护。

下列步骤显示了安全保护程序寄存器的访问顺序：

- 检查 OPCR 寄存器，以确认没有正在执行 Flash 存储器操作 (OPM [3:0] = 0xE 或 0x6)。否则，需一直等待到前面的操作已经完成。
- 将 OB\_CP 地址写入到 TADR 寄存器 (TADR = 0x1FF0\_0010)。
- 写 WRDR 寄存器设置 OB\_CP [0] 为 0。
- 将字编程命令写入到 OCMR 寄存器 (CMD [3:0] = 0x4)。
- 通过设置 OPCR 寄存器，将字编程命令发送到 FMC (OPM [3:0] = 0xA)。
- 等待，直到通过检查 OPCR 寄存器的值确认所有操作已经完成 (OPM [3:0] = 0xE)。
- 依据需要，通过访问 DCODE 读取和验证选项字节的内容。
- 根据选项字节校验和规则，编程 OB\_CK 使选项字节为 5 个字 0x000 ~ 0x010 的总和。
- 使系统复位以激活新的安全保护设置。

## 寄存器列表

下表显示了 FMC 寄存器及其复位值。

**表 10. FMC 的寄存器列表**

寄存器	偏移量	描述	复位值
TADR	0x000	Flash 目标地址寄存器	0x0000_0000
WRDR	0x004	Flash 写数据寄存器	0x0000_0000
OCMR	0x00C	Flash 操作命令寄存器	0x0000_0000
OPCR	0x010	Flash 操作控制寄存器	0x0000_000C
OIER	0x014	Flash 操作中中断使能寄存器	0x0000_0000
OISR	0x018	Flash 操作中中断状态寄存器	0x0001_0000
PPSR	0x020 0x024 0x028 0x02C	Flash 页擦除 / 编程保护状态寄存器	0xFFFF_XXXX 0xFFFF_XXXX 0xFFFF_XXXX 0xFFFF_XXXX
CPSR	0x030	Flash 安全保护状态寄存器	0x0000_000X
VMCR	0x100	Flash 向量映射控制寄存器	0x0000_000X
MDID	0x180	Flash 制造商和单片机 ID 寄存器	0x0376_XXXX
PNSR	0x184	Flash 页数量状态寄存器	0x0000_00XX
PSSR	0x188	Flash 页大小状态	0x0000_0400
DID	0x18C	设备 ID 寄存器	0x000X_XXXX
CFCR	0x200	Flash 缓存和预取控制寄存器	0x0000_1091
SBVT0	0x300	SRAM 启动向量 0 (堆栈指针)	0x200X_0000
SBVT1	0x304	SRAM 启动向量 1 (程序计数器)	0x2000_0159
SBVT2	0x308	SRAM 启动向量 2 (NMI 处理器)	0x0000_0000
SBVT3	0x30C	SRAM 启动向量 3 (硬故障处理器)	0x0000_0000
CIDR0	0x310	自定义 ID 寄存器 0	0xFFFF_XXXX
CIDR1	0x314	自定义 ID 寄存器 1	0xFFFF_XXXX
CIDR2	0x318	自定义 ID 寄存器 2	0xFFFF_XXXX
CIDR3	0x31C	自定义 ID 寄存器 3	0xFFFF_XXXX

注：“X”表示不同的复位值，取决于单片机、Flash 值、选项字节值或上电复位设置。

寄存器描述

Flash 目标地址寄存器 – TADR

该寄存器定义了页擦除和字编程操作的目标地址。

偏移量: 0x000  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	TADB								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	TADB								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	TADB								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	TADB								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	TADB	Flash 目标地址位 对于编程操作，TADR 寄存器指定了数据被写入的地址。由于编程长度是 32 位，TADR 应被设置为字对齐 ( 4 字节 )。TADB [1:0] 在编程操作期间将被忽略。对于页擦除操作，TADR 寄存器包含会被擦除的页地址。由于页大小为 1 KB，TADB [9:0] 将被忽略，以限制目标地址为 1 KB 对齐。对于 256 KB，TADB [31:18] 应设置为 0。从地址 0x1FF0_0000 到 0x1FF0_03FF 为 1 KB 选项字节。对于这个区域的可用 Flash 地址，必须在 0x0000_0000 ~ 0x1FFF_FFFF。否则，若相应中断使能，将产生无效目标地址中断。

## Flash 写数据寄存器 – WRDR

该寄存器用于存储编程操作中将写入 TADR 的数据。

偏移量: 0x004

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	WRDB							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	WRDB							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	WRDB							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	WRDB							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

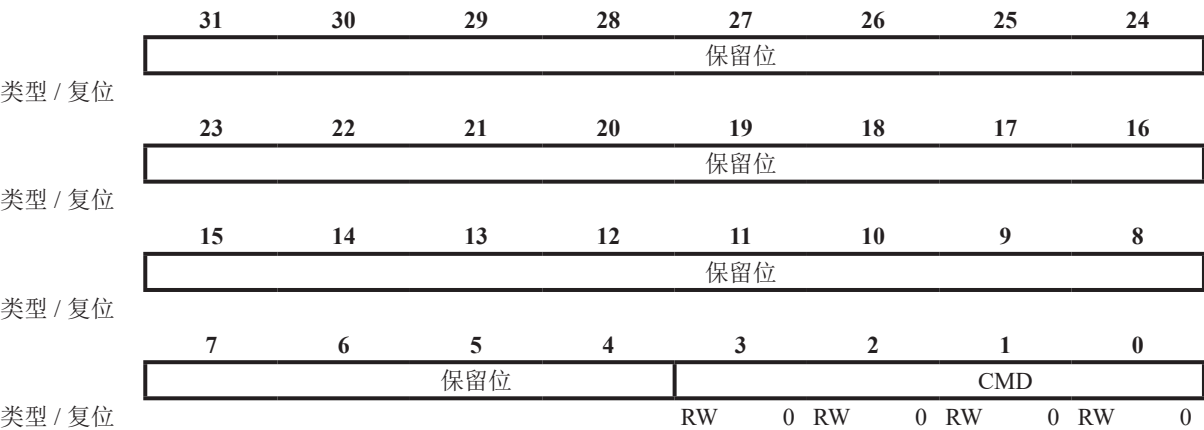
位	字段	描述
[31:0]	WRDB	Flash 写数据位 用于编程操作的数据值。



Flash 操作命令寄存器 – OCMR

该寄存器定义了 Flash 操作命令，包括字编程、页擦除和整片擦除。

偏移量： 0x00C  
复位值： 0x0000\_0000



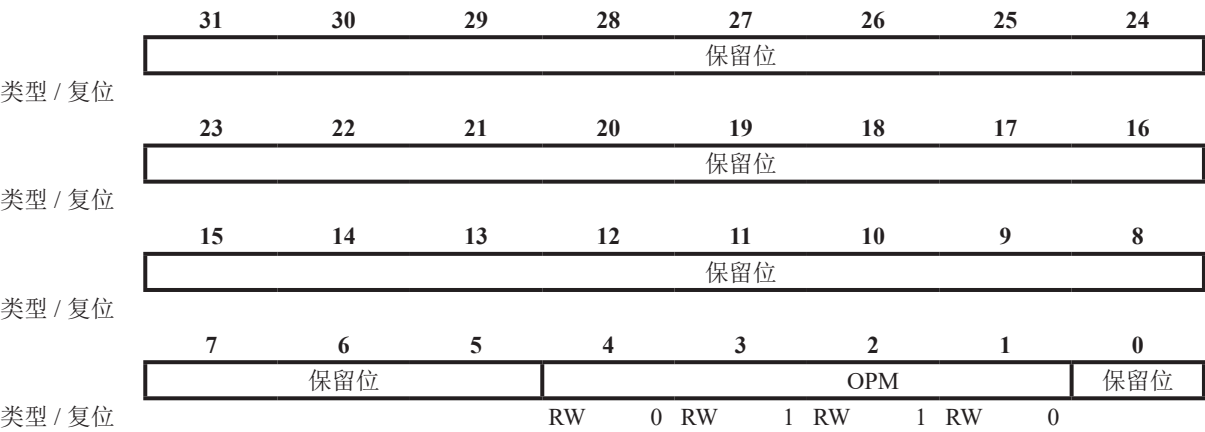
位	字段	描述
[3:0]	CMD	Flash 操作命令位 下表显示了 CMD [3:0] 的定义，显示了 Flash 操作。如果设置了一个无效的命令，且 IOCMEN 位等于 1，将产生一个无效操作命令中断。

CMD [3:0]	描述
0x0	空闲 (默认)
0x4	字编程
0x8	页擦除
0xA	整片擦除
其它值	保留

Flash 操作控制寄存器 – OPCR

该寄存器用于控制命令提交和检查 FMC 操作的状态。

偏移量： 0x010  
复位值： 0x0000\_000C



位	字段	描述
[4:1]	OPM	操作模式位 下表列出了 FMC 的操作模式。根据 TADR 寄存器的地址别名设置，用户可以提交由 OCMR 寄存器对主 Flash 设置的命令。TADR、WRDR 和 OCMR 寄存器的内容应该在设置该寄存器之前准备好。在所有操作都完成后，OPM 字段将由 FMC 硬件设置为 0xE。当所有操作已完成，设置进入空闲模式以降低功耗。注意，FMC 执行下一步动作之前，应检查操作状态。TADR、WRDR、OCMR 和 OPCR 寄存器的内容不应该被改变直到前面的操作完成。

OPM [3:0]	描述
0x6	空闲 (默认)
0xA	提交命令给主 Flash
0xE	所有操作在主 Flash 中完成
其它值	保留

Flash 操作中断使能寄存器 – OIER

该寄存器用来使能或除能 FMC 中断功能。当相应的中断使能位被置位，且相对应的事件发生时 FMC 会产生对控制器的中断。

偏移量:	0x014
复位值:	0x0000_0000
	3130292827262524
类型 / 复位	保留位
	2322212019181716
类型 / 复位	保留位
	15141312111098
类型 / 复位	保留位
	76543210
类型 / 复位	保留位OREIENIOCMIENOBEIENITADIENORFIEN
	RW0RWOBEIENITADIENORFIEN

位	字段	描述
[4]	OREIEN	操作错误中断使能位 0: 操作错误不产生中断 1: 操作错误产生中断
[3]	IOCMIEN	无效操作命令中断使能位 0: 无效操作命令不产生中断 1: 无效操作命令产生中断
[2]	OBEIEN	选项字节校验和错误中断使能位 0: 选项字节校验和错误不产生中断 1: 选项字节校验和错误产生中断
[1]	ITADIEN	无效目标地址中断使能位 0: 无效目标地址不产生中断 1: 无效目标地址产生中断
[0]	ORFIEN	操作完成中断使能位 0: 操作完成不产生中断 1: 操作完成产生中断

## Flash 操作中断状态寄存器 – OISR

该寄存器定义了 FMC 的中断状态，它用来检查一个 Flash 操作是否完成或者是否有错误发生。当 OIER 寄存器中相应的位被置位时，该状态位（位 [4:0]）可以使用。

偏移量： 0x018

复位值： 0x0001\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位						PPEF	RORFF
类型 / 复位						RO 0	RO 1
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位			OREF	IOCMF	OBEF	ITADF	ORFF
类型 / 复位			WC 0	WC 0	WC 0	WC 0	WC 0

位	字段	描述
[17]	PPEF	页擦除 / 编程保护错误标志位 0: 页擦除 / 编程保护错误未发生 1: 操作错误发生，由于无效的页擦除 / 编程操作被应用到受保护页一旦新的 Flash 操作命令被提交，该位由硬件复位。
[16]	RORFF	原始操作完成标志位 0: 上一个 Flash 操作命令未完成 1: 上一个 Flash 操作命令完成 该位直接连接到 Flash 存储器用于调试。
[4]	OREF	操作错误标志位 0: Flash 操作错误未发生 1: 上一个 Flash 操作失败 当任何一个 Flash 操作错误发生时，例如无效的命令、编程错误和擦除错误等，该位将被置位。如果 OIER 寄存器中的 OREIEN 位被置位，ORE 中断将发生。通过写 1 来复位该位。
[3]	IOCMF	无效操作命令标志位 0: 未设置无效的 Flash 操作命令 1: 无效的 Flash 操作命令被写入到 OCMR 寄存器 如果 OIER 寄存器中的 IOCMIEIEN 位被置位，IOCM 中断将发生。通过写 1 来复位该位。
[2]	OBEF	选项字节校验和错误标志位 0: 选项字节校验和正确 1: 选项字节校验和错误 如果 OIER 寄存器中的 OBEIEN 位被置位，OBE 中断将发生。但直到中断条件清除后 OBEF 才可通过软件写 1 来复位，即选项字节校验和的值需要正确调整或相应的中断控制需除能，否则将继续产生中断。

位	字段	描述
[1]	ITADF	无效目标地址标志位 0: 目标地址 TADR 有效 1: 目标地址 TADR 无效 TADR 字段必须小于 1FFF_FFFF。如果 OIER 寄存器中的 ITADIEN 位被置位, ITAD 中断将发生。通过写 1 来复位该位。
[0]	ORFF	操作完成标志位 0: Flash 操作完成未发生 1: Flash 操作完成发生 如果 OIER 寄存器中的 ORFIEN 位被置位, ORF 中断将发生。通过写 1 来复位该位。

Flash 页擦除 / 编程保护状态寄存器 – PPSR

该寄存器定义了 Flash 存储器页擦除 / 程序保护的状态。

偏移量: 0x020 (0) ~ 0x02C (3)  
复位值: 0xFFFF\_FFFF

	31	30	29	28	27	26	25	24	
	PPSBn								
类型 / 复位	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	23	22	21	20	19	18	17	16	
	PPSBn								
类型 / 复位	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	15	14	13	12	11	10	9	8	
	PPSBn								
类型 / 复位	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	7	6	5	4	3	2	1	0	
	PPSBn								
类型 / 复位	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X

位	字段	描述
[127:0]	PPSBn	第 n 页擦除 / 编程保护状态位 (n = 0 ~ 127) PPSB[n] = OB_PP[n] 0: 相应的页受保护 1: 相应的页不受保护 该寄存器的内容不是动态更新的, 当发生任何一种复位, 只能通过选项字节重新加载。当 PPSR 寄存器中的相应位被复位时, 特定页的擦除或编程功能不被允许。 PPSR [127:0] 的复位值由选项字节 OB_PP [127:0] 决定。因该系列主 Flash 最大页数量为 127, 因此第 n 个页擦除 / 编程保护状态位可对应 2 页。

Flash 安全保护状态寄存器 – CPSR

该寄存器定义了 Flash 存储器安全保护状态。它的内容不是动态更新的，只能在任何一种复位发生后通过选项字节加载器重新加载。

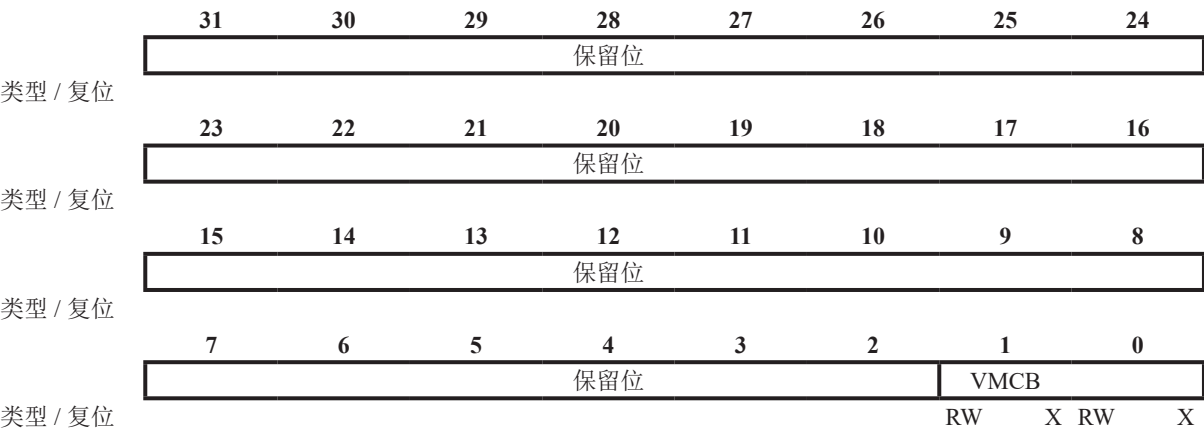
偏移量:	0x030
复位值:	0x0000_000X
	3130292827262524
类型 / 复位	保留位
	2322212019181716
类型 / 复位	保留位
	15141312111098
类型 / 复位	保留位
	76543210
类型 / 复位	保留位OBPSBCPSB
	ROXROX

位	字段	描述
[1]	OBPSB	选项字节页擦除 / 编程保护状态位 0: 选项字节页受保护 1: 选项字节页不受保护 OBPSB 位的复位值是由选项字节中的 OB_CP [1] 位决定。
[0]	CPSB	Flash 存储器安全保护状态位 0: Flash 存储器安全保护使能 1: Flash 存储器安全保护除能 CPSB 位的复位值是由选项字节中的 OB_CP [0] 位决定。

Flash 向量映射控制寄存器 – VMCR

该寄存器用来控制向量映射。VMCR 寄存器的复位值是由上电复位期间设置引脚 BOOT0 ~ BOOT1 决定。

偏移量: 0x100  
复位值: 0x0000\_000X



位	字段	描述																				
[1:0]	VMCB	向量映射控制位 VMCB 位用来控制地址从 0x0 到 0xC 的前 4 个字向量的映射源。下表显示了向量映射的设置。																				
		<table><tr><th>BOOT1</th><th>BOOT0</th><th>VMCB [1:0]</th><th>描述</th></tr><tr><td>低</td><td>低</td><td>00</td><td>SRAM 启动模式 向量映射源是 SBVT0 ~ 3。</td></tr><tr><td>低</td><td>高</td><td>01</td><td>启动加载器模式 向量映射源是启动加载器区域</td></tr><tr><td>高</td><td>低</td><td>10</td><td>主 Flash 模式</td></tr><tr><td>高</td><td>高</td><td>11</td><td>模式向量映射源是主 Flash 存储器区域</td></tr></table>	BOOT1	BOOT0	VMCB [1:0]	描述	低	低	00	SRAM 启动模式 向量映射源是 SBVT0 ~ 3。	低	高	01	启动加载器模式 向量映射源是启动加载器区域	高	低	10	主 Flash 模式	高	高	11	模式向量映射源是主 Flash 存储器区域
BOOT1	BOOT0	VMCB [1:0]	描述																			
低	低	00	SRAM 启动模式 向量映射源是 SBVT0 ~ 3。																			
低	高	01	启动加载器模式 向量映射源是启动加载器区域																			
高	低	10	主 Flash 模式																			
高	高	11	模式向量映射源是主 Flash 存储器区域																			
		VMCR 寄存器的复位值是由上电复位和系统复位期间外部启动引脚 BOOT1 和 BOOT0 的状态决定。当应用程序被执行时，通过配置 VMCB 位以改变向量映射设置。																				

Flash 制造商和单片机 ID 寄存器 – MDID

该寄存器定义了制造 ID 和单片机型号信息，可用于产品识别。

偏移量： 0x180  
复位值： 0x0376\_XXXX

	31	30	29	28	27	26	25	24
	MFID							
类型 / 复位	RO	0	RO	0	RO	0	RO	1
	23	22	21	20	19	18	17	16
	MFID							
类型 / 复位	RO	0	RO	1	RO	1	RO	0
	15	14	13	12	11	10	9	8
	ChipID							
类型 / 复位	RO	X	RO	X	RO	X	RO	X
	7	6	5	4	3	2	1	0
	ChipID							
类型 / 复位	RO	X	RO	X	RO	X	RO	X

位	字段	描述
[31:16]	MFID	制造商 ID 读为 0x0376。
[15:0]	ChipID	芯片 ID 读取 MCU 型号最后四位数字值。



## Flash 页数量状态寄存器 – PNSR

该寄存器定义了 Flash 存储器的页数。

偏移量: 0x184

复位值: 0x0000\_00XX

	31	30	29	28	27	26	25	24	
	PNSB								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	PNSB								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	PNSB								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	PNSB								
类型 / 复位	RO	X	RO	X	RO	X	RO	X	RO

位	字段	描述
[31:0]	PNSB	Flash 页数状态位 0x0000_0020: 片上 Flash 存储器总页数为 32 0x0000_0040: 片上 Flash 存储器总页数为 64 0x0000_0080: 片上 Flash 存储器总页数为 128 0x0000_00FF: 片上 Flash 存储器总页数为 255 这些位表示片上 Flash 存储器总页数。

**Flash 页大小状态寄存器 – PSSR**

该寄存器定义了以字节为单位的页大小。

偏移量: 0x188  
复位值: 0x0000\_0400

	31	30	29	28	27	26	25	24	
	PSSB								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	PSSB								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	PSSB								
类型 / 复位	RO	0	RO	0	RO	0	RO	1	RO
	7	6	5	4	3	2	1	0	
	PSSB								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[31:0]	PSSB	Flash 页大小状态位 0x200: 每页 512 字节 0x400: 每页 1 KB 0x800: 每页 2 KB

设备 ID 寄存器 – DID

该寄存器用于存储设备型号信息，可用于产品识别。

偏移量： 0x18C  
复位值： 0x000X\_XXXX

	31	30	29	28	27	26	25	24				
	保留位											
类型 / 复位												
	23	22	21	20	19	18	17	16				
	保留位				ChipID							
类型 / 复位					RO	X	RO	X	RO	X	RO	X
	15	14	13	12	11	10	9	8				
	ChipID											
类型 / 复位												
	RO	X	RO	X	RO	X	RO	X	RO	X	RO	X
	7	6	5	4	3	2	1	0				
	ChipID											
类型 / 复位												
	RO	X	RO	X	RO	X	RO	X	RO	X	RO	X

位	字段	描述
[19:0]	ChipID	芯片 ID MCU 设备型号。

Flash 缓存和预取控制寄存器 – CFCR

该寄存器用于控制 FMC 缓存和预取模块。

偏移量：0x200

复位值：0x0000\_1091

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位			CE	保留位			
类型 / 复位			RW	1			
7	6	5	4	3	2	1	0
DCDB	保留位		PFBE	保留位	WAIT		
RW	1		RW	1	RW	0	RW 0 RW 0 RW 1
类型 / 复位							

位	字段	描述
[12]	CE	分支缓存使能位 0: 分支缓存除能 1: 分支缓存使能
[7]	DCDB	DCODE 数据缓存使能位 0: DCODE 数据缓存使能 1: DCODE 数据缓存除能
[4]	PFBE	预取缓冲器使能位 0: 预取缓冲器除能 1: 预取缓冲器使能 (默认) 在默认状态下, 预取缓冲器使能。当预取缓冲器除能, 指令和数据直接由 Flash 存储器提供。
[2:0]	WAIT	Flash 等待周期设置位 WAIT [2:0] 用于设置在非连续 Flash 访问期间 HCLK 等待时钟。等待时钟的实际值是 (WAIT [2:0]-1)。由于提供了一个带预取缓存器和分支缓存的宽访问接口, 连续 Flash 访问的等待状态非常接近于零。

WAIT [2:0]	等待状态	HCLK 允许的范围
001	0	0 MHz < HCLK ≤ 24 MHz
010	1	24 MHz < HCLK ≤ 48 MHz
011	2	48 MHz < HCLK ≤ 72 MHz
其它值	3	72 MHz < HCLK ≤ 96 MHz

SRAM 启动向量寄存器 n – SBVTn, n = 0 ~ 3

这些寄存器定义了用于 SRAM 启动模式的堆栈指针、程序计数器、NMI 处理器地址和硬故障处理器地址。

偏移量: 0x300 (0) ~ 0x30C (3)  
复位值: 取决于地址偏移量的变化

	31	30	29	28	27	26	25	24	
	SBVTn								
类型 / 复位	RW	X	RW	X	RW	X	RW	X	RW
	23	22	21	20	19	18	17	16	
	SBVTn								
类型 / 复位	RW	X	RW	X	RW	X	RW	X	RW
	15	14	13	12	11	10	9	8	
	SBVTn								
类型 / 复位	RW	X	RW	X	RW	X	RW	X	RW
	7	6	5	4	3	2	1	0	
	SBVTn								
类型 / 复位	RW	X	RW	X	RW	X	RW	X	RW

位	字段	描述
[31:0]	SBVTn	SRAM 启动向量 n (n = 0 ~ 3)

SRAM 启动向量 0 ~ 3 提供了一个用于应用调试的 SRAM 启动能力。在 SRAM 启动模式下，SBVTn 寄存器的内容被重映射到地址 0x0 到 0xC 的 Flash 存储器的代码区。请参考 VMCR 寄存器和 BOOT1 / BOOT0 启动引脚的描述。下表列出了 SBVTn 寄存器的用途和复位值。复位值提供了在 SRAM 启动模式下程序执行的固定设置。这些寄存器可以由调试工具修改，以更改程序执行的设置。SBVTn 寄存器的复位值只能由上电复位重新加载，其它复位源将没有任何效果。

名称	地址偏移量	用途描述	复位值
SBVT0	0x300	堆栈指针	128 KB SRAM: 0x2002_0000 64 KB SRAM: 0x2001_0000
SBVT1	0x304	程序计数器	0x2000_0159
SBVT2	0x308	NMI 处理器地址	0x0000_0000
SBVT3	0x30C	硬故障处理器地址	0x0000_0000

SBVT0 ~ SBVT3 寄存器的访问宽度必须是 32-bit 访问 (字访问)。8 或 16 位访问 (字节或半字访问) 是不允许的。

自定义 ID 寄存器 n – CIDRn, n = 0 ~ 3

该寄存器定义了自定义 ID 信息，用来作为自定义标识。

- 偏移量： 0x310 (0) ~ 0x31C (3)  
复位值： 有多种复位值，取决于 Flash 制造权限信息块。

	31	30	29	28	27	26	25	24	
	CID								
类型 / 复位	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	23	22	21	20	19	18	17	16	
	CID								
类型 / 复位	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	15	14	13	12	11	10	9	8	
	CID								
类型 / 复位	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	7	6	5	4	3	2	1	0	
	CID								
类型 / 复位	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X

位	字段	描述
[31:0]	CIDn	自定义 ID Flash 制造权限信息块中自定义 ID 寄存器的 CIDn [31:0] (n = 0 ~ 3) 字段。

## 5 电源控制单元 (PWRCU)

### 简介

功耗被视为许多嵌入式系统应用中很重要的问题之一。因此，在这些单片机中，电源控制单元 PWRCU 提供多种省电模式如休眠模式、深度休眠模式 1、深度休眠模式 2 和暂停模式。这些工作模式可以降低功耗，并允许应用程序在 CPU 运行时间、速度和功耗相互冲突的需求中达到较佳平衡。图 11 中的虚线指出了三个数字电源域的供电来源。

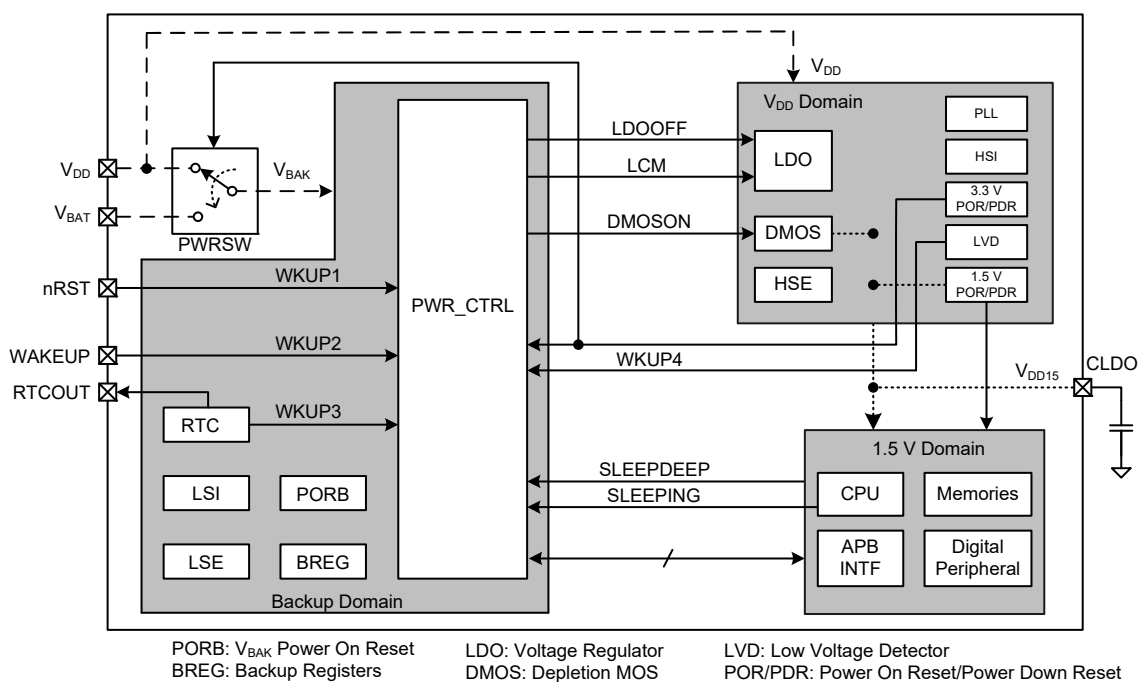


图 11. PWRCU 方框图

## 特性

- 三个电源域：备份域、 $V_{DD}$  和 1.5 V 电源域
- 四种省电模式：休眠模式、深度休眠模式 1、深度休眠模式 2 和暂停模式
- 内部稳压器提供 1.5 V 电压源
- 额外耗尽型 MOS 提供 1.5 V 具有小漏电流和小工作电流的电压源
- 下列条件之一成立时，将发生电源复位
  - 上电 / 暂停 (POR / PDR 复位)
  - 当退出暂停模式
  - 控制位  $BODEN = 1$ ,  $BODRIS = 0$  且电源电压  $V_{DD} \leq V_{BOD}$
- 当  $V_{DD}$  电源低于欠压检测器电压  $V_{BOD}$  时，欠压检测器可以产生系统复位或中断
- 当  $V_{DD}$  低于可编程阈值电压  $V_{LVD}$  时，低电压检测器可以产生中断或唤醒事件
- 当  $V_{DD}$  低于  $V_{PDR}$  电压时，电池  $V_{BAT}$  供电给备份域
- 在暂停模式下，备份寄存器的 40 字节由  $V_{BAK}$  供电用于用户应用数据的存储

## 功能描述

### 备份域

#### 电源开关

备份域由电源开关 **PWRSW** 选择是由  $V_{DD}$  电源还是电池电源  $V_{BAT}$  供电。备份域工作电压范围从 2.0 V 至 3.6 V。如果  $V_{DD}$  低于  $V_{PDR}$ ，那么备份域电源将从  $V_{DD}$  切换到  $V_{BAT}$ 。因此，即使  $V_{DD}$  暂停，所有备份域电路都可以正常运行。这意味着，备份寄存器的内容将被保留，RTC 电路将正常工作，低速振荡器可以继续运行。

#### 备份域复位

备份域复位源包括备份域上电复位 **PORB** 和备份域软件复位，备份域软件复位通过设置 **BAKCR** 寄存器中的 **BAKRST** 位来激活。**PORB** 信号迫使单片机停留在复位模式，直到  $V_{BAK}$  大于  $V_{PORB}$ 。应用软件可以设置 **BAKCR** 寄存器中的 **PORBDN** 位来除能 **PORB** 电路以节省备份域的耗电。通过设置 **BAKCR** 寄存器中的 **BAKRST** 位，应用软件还可以触发备份域软件复位来复位备份域。**PWRCU** 和 **RTC** 中的所有寄存器只能通过备份域复位来复位。



## LSE, LSI 和 RTC

实时时钟电路的时钟源可以来自内部低速 RC 振荡器 LSI 或外部低速晶体振荡器 LSE。通过执行 WFI / WFE 指令进入暂停模式之前, MCU 需要为比较寄存器设置一个预期唤醒时间, 并使能唤醒功能, 以实现 RTC 定时器唤醒事件。在进入暂停模式一定时间以后, 当比较匹配事件发生, 比较匹配标志位 CMFLAG 将会被置位以唤醒单片机。用于唤醒定时器的 RTC 配置的细节将在 RTC 章节描述。

## 备份寄存器和隔离单元

10 个位于备份域的 32-bit 寄存器, 高达 40 个字节, 用于用户应用数据的存储。当 1.5 V 内核电源被切断时, 这些寄存器由  $V_{BAK}$  持续供电。备份寄存器只能通过备份域上电复位 PORB 或备份域软件复位 BAKRST 复位。当单片机从 1.5 V 电源恢复运作, 无论是通过硬件还是软件, 由隔离单元禁止访问备份寄存器和 RTC 寄存器, 从而保护这些寄存器, 防止可能的寄生写访问。要恢复访问操作, 用户必须通过设置时钟控制单元 LPCR 寄存器中的 BKISO 位为 1 来除能这些隔离单元。

## LDO 电源控制

当有下列情况之一发生, LDO 将自动关闭:

- 进入暂停模式或深度休眠模式 2
- 控制位 BODEN = 1, BODRIS = 0 且电源  $V_{DD} \leq V_{BOD}$
- 电源电压  $V_{DD} \leq V_{PDR}$

电源电压  $V_{DD} > V_{POR}$  时, 如果下列任一情况发生, LDO 会自动由硬件开启:

- 从省电模式恢复运作 – RTC 唤醒, LVD 唤醒和 WAKEUP 引脚的上升沿
- 检测到外部复位引脚 ( $nRST$ ) 的下降沿
- 控制位 BODEN = 1, 电源  $V_{DD} > V_{BOD}$

要进入深度休眠模式 1, PWRCU 会要求 LDO 工作在低电流模式 LCM 下。要进入深度休眠模式 2, PWRCU 将关闭 LDO, 然后打开 DMOS 提供另一种 1.5 V 电源。

## $V_{DD}$ 电源域

### 稳压器

稳压器 LDO、耗尽型 MOS 管 DMOS、低电压检测器 LVD 和内部高速振荡器 HSI, 都在  $V_{DD}$  电源域下工作。LDO 可配置为工作在正常模式 ( $LDOOFF = 0$ ,  $SLEEPDEEP = 0$ ,  $I_{OUT}$  = 大电流模式) 或小电流模式 ( $LDOOFF = 0$ ,  $SLEEPDEEP = 1$ ,  $I_{OUT}$  = 小电流模式), 以提供 1.5 V 电源。另一种 1.5 V 电源来自具有小静态电流和小驱动电流特性的 DMOS 输出。它通过使用 BAKCR 寄存器中的 DMOSON 位被控制。DMOS 输出具有弱输出电流和稳压能力, 在  $V_{DD15}$  电源域只能工作在深度休眠模式 2 来实现数据保存。

### 上电复位 (POR) / 掉电复位 (PDR)

该系列单片机内置 POR / PDR 电路, 允许单片机在从 2.0 V 开始升压或者降压至 2.0 V 的过程中都可以适当工作。当单片机进入暂停模式时若是发生  $V_{DD}$  低于特定的阈值电压  $V_{PDR}$  时单片机将维持在暂停模式, 无需外部复位电路。欲知详细上电 / 掉电复位的阈值电压, 请参考相关规格书中的电气特性部分。

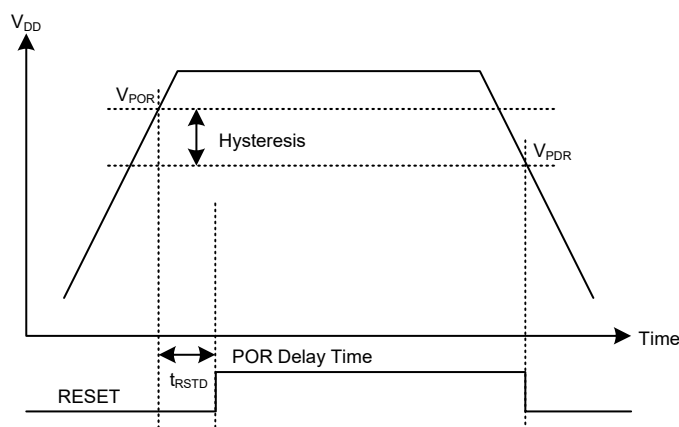


图 12. 上电复位 / 掉电复位波形

#### 低电压检测器 / 欠压检测器

低电压检测 LVD 也可以检测电源电压  $V_{DD}$  是否低于可编程阈值电压  $V_{LVD}$ 。阈值电压由 LVDCSR 寄存器中的 LVDS 字段选择。当在  $V_{DD}$  电源引脚上检测到一个低电压, LVDF 标志将有效, 如果 LVDCSR 寄存器中的 LVDEN 和 LVDIWEN 位被置位, 将产生一个中断并发送给 MCU 内核。有关 LVD 可编程阈值电压  $V_{LVD}$  的详细信息, 可参考规格书中的电气特性部分。

欠压检测器 BOD 用来检测  $V_{DD}$  电源电压是否等于或低于  $V_{BOD}$ 。当 LVDCSR 寄存器中的 BODEN 位被设置为 1,  $V_{DD}$  电源电压低于  $V_{BOD}$  时, BODF 标志将有效。PWRCU 就会把这个作为一个暂停复位的情况, 然后立即除能内部 LDO 稳压器 (当 BODRIS = 0) 或发出中断来通知 CPU 执行进入暂停模式的程序 (当 BODRIS = 1)。有关欠压检测器 BOD 电压  $V_{BOD}$  的详细信息, 可参考规格书中的电气特性部分。

#### 内部高速振荡器

内部高速振荡器 HSI 位于  $V_{DD}$  电源域。当从深度休眠模式退出时, 通过设置 PSRCEN 位为 1, HSI 时钟可以在一段时间内被配置作为系统时钟, 此位位于时钟控制单元 CKCU 中的全局时钟控制寄存器 GCCR。系统时钟不会切换回进入深度休眠模式之前使用的初始时钟源, 直到这个可能来自 PLL 或 HSE 的初始时钟源稳定。此外, 系统从暂停模式唤醒后, 即将发生 1.5 V 上电复位, 此时会迫使 HSI 振荡器作为系统时钟。

#### 外部高速振荡器

外部高速振荡器 HSE 位于  $V_{DD}$  电源域。当从深度休眠模式退出时, 通过设置 HSEEN 位可以关闭和开启该时钟, 此位位于时钟控制单元 CKCU 中的全局时钟控制寄存器 GCCR。HSE 时钟可直接用作系统时钟源或 PLL 时钟源。

1.5 V 电源域

主要功能包括用于备份域的 APB 接口、CPU 内核逻辑、AHB / APB 外设和存储器等，都位于此电源域。一旦 1.5 V 电源上电，POR 将产生一个在 1.5 V 电源域的复位序列 (参考 PORB)。随后，要进入预期的省电模式，相关的控制位包括 LDOOFF、DMOSON 和 SLEEPDEEP 位必须被配置。然后，一旦 WFI 或 WFE 指令被执行，单片机将进入预期的省电模式，这个将在下一章节讨论。

工作模式

运行模式

在运行模式下，系统可运行所有功能且所有电源域都可用。此模式下，有两种方法减少功耗。第一种是通过设置 CKCU AHBCFGR 寄存器中的 AHBPRE 字段以减慢系统时钟，第二种是通过设置 APBCCR0 和 APBCCR1 寄存器以关闭未使用的外设时钟或通过设置 APBPCSR0 和 APBPCSR1 寄存器减慢外围时钟以满足应用需求。在进入休眠模式之前减小系统时钟速度也有助于降低功耗。

此外，有几种省电模式可以提供单片机性能和功耗之间的较大优化。

表 11. 工作模式定义

模式名称	硬件动作
运行	系统复位后，CPU 获取指令执行。
休眠	1. CPU 内核时钟将停止。 2. 外设、Flash 和 SRAM 的时钟通过设置可以停止。
深度休眠 1 ~ 2	1. 停止 1.5 V 电源域的所有时钟。 2. 除能 HSI、HSE 和 PLL。 3. 通过打开 LDO 小电流模式或 DMOS 来减少 1.5 V 电源域的电流。
暂停	关闭 1.5 V 电源域。

休眠模式

系统进入休眠模式，只有 CPU 时钟会停止。清除 CKCU AHBCCR 寄存器中的 FMCEN 或 SRAMEN 位为 0，在系统进入休眠模式后将会停止 Flash 时钟或 SRAM 时钟。如果在休眠模式 CPU 没有必要访问 Flash 存储器和 SRAM，建议清除 AHBCCR 寄存器中的 FMCEN 和 SRAMEN 位来降低功耗。要进入休眠模式，只需要清除 SLEEPDEEP 位为 0，执行 WFI 或 WFE 指令。通过任何中断或事件触发，系统将退出休眠模式。下表提供了有关省电模式的更多信息。

表 12. 进入 / 退出省电模式

模式	模式进入				模式退出
	CPU 指令	CPU SLEEPDEEP	LDOOFF	DMOSON	
休眠	WFI 或 WFE (生效)	0	X	X	WFI: 任何中断 WFE: 任何唤醒事件 <sup>(1)</sup> 或 任何中断 (NVIC on) 或 SEVONPEND = 1 的任何中断 (NVIC off)
深度休眠 1		1	0	0	在事件模式的任何 EXTI 或 CMP 唤醒或 RTC 唤醒或 LVD 唤醒 <sup>(2)</sup> 或 WAKEUP 引脚的上升沿或 USB 恢复
深度休眠 2		1	X	1	RTC 唤醒或 LVD 唤醒 <sup>(2)</sup> 或 WAKEUP 引脚的上升沿
暂停		1	1	0	RTC 唤醒或 LVD 唤醒 <sup>(2)</sup> 或 WAKEUP 引脚的上升沿或 外部复位 (nRST)

注: 1. 唤醒事件是指在事件模式的 EXTI 引脚、RTC、LVD 和 WAKEUP 引脚的上升沿。

2. 系统进入省电模式之后, 如果允许 LVD 来唤醒, LVDCSR 寄存器中的 LVDEWEN 和 LVDEN 位需设为 1, 以确保系统可以由一个 LVD 事件唤醒, 从深度休眠 2 和暂停模式中唤醒时, LDO 稳压器可以开启。

### 深度休眠模式

要进入深度休眠模式, 配置如上表所示的寄存器, 并执行 WFI 或 WFE 指令。在深度休眠模式下, 所有时钟包括 PLL 和高速振荡器, 即 HSI 和 HSE 将被停止。此外, 深度休眠模式 1 使 LDO 变成小电流模式, 深度休眠模式 2 关闭 LDO, 并使用 DMOS 保持 1.5 V 电源。一旦 PWRCU 接收到一个如上表模式退出栏所述的唤醒事件或中断, LDO 将工作在正常模式, 高速振荡器将使能。最后, CPU 将返回到运行模式, 并依需要来处理唤醒中断。如果 LVDCSR 寄存器中相应的唤醒控制位 LVDEWEN 使能, 低电压检测也可以被看作是唤醒事件。最后一个唤醒事件是外部 WAKEUP 引脚由低到高电平的转换, 发送到 PWRCU 以从深度休眠模式中恢复。在深度休眠模式, 保持寄存器和存储器的内容会缩短唤醒延迟。

### 暂停模式

暂停模式衍生于 CPU 深度休眠模式外加控制位 LDOOFF 和 DMOSON。要进入暂停模式, 用户可以配置如前面的模式进入表所示的寄存器, 并执行 WFI 或 WFE 指令。RTC 唤醒触发事件、LVD 唤醒, 外部 WAKEUP 引脚从低到高电平的转换或外部复位引脚 (nRST) 信号将迫使 CPU 退出暂停模式。在暂停模式下, 1.5 V 电源将被关闭。其余的有效电源是 3.3 V 电源 ( $V_{DD}$  /  $V_{DDA}$ ) 和备份域电源 ( $V_{BAK}$ )。

系统复位后, RSTCU GRSR 寄存器中的 PORSTF 位, BAKSR 寄存器中的 PDF 和 BAKPORF 位应该由软件进行检查, 以确定单片机是否通过备份域上电复位或 1.5 V 电源意外掉电或其它复位 (nRST, WDT 等) 从暂停模式中恢复。如果单片机有正确的进入暂停模式, 那么 PDF 位将被置位。系统信息可以被保存在备份寄存器中, 当 1.5 V 电源域再次上电时可以被重新取回。有关 BAKSR 寄存器中 PDF 位和 BAKPORF 位以及 RSTCU GRSR 寄存器中的 PORSTF 位的更多信息如下表所示。

表 13. 系统复位后的电源状态

BAKPORF	PDF	PORSTF	描述
1	0	1	备份域复位后首次上电： 当 $V_{BAK}$ 首次使用或对备份域执行软件复位
0	0	1	从 1.5 V 电源意外掉电重新启动或其它复位 (nRST, WDT 等)
0	1	1	从暂停模式重新启动
1	1	X	保留

## 寄存器列表

下表显示了 PWRCU 寄存器及其复位值。注意，在本单元的所有寄存器位于  $V_{BAK}$  备份电源域。

表 14. PWRCU 寄存器列表

寄存器	偏移量	描述	复位值
BAKSR	0x100	备份域状态寄存器	0x0000_0001
BAKCR	0x104	备份域控制寄存器	0x0000_0000
BAKTEST	0x108	备份域测试寄存器	0x0000_0027
LVDCSR	0x110	低电压 / 欠压检测控制和状态寄存器	0x0000_0000
BAKREG0	0x200	备份寄存器 0	0x0000_0000
BAKREG1	0x204	备份寄存器 1	0x0000_0000
BAKREG2	0x208	备份寄存器 2	0x0000_0000
BAKREG3	0x20C	备份寄存器 3	0x0000_0000
BAKREG4	0x210	备份寄存器 4	0x0000_0000
BAKREG5	0x214	备份寄存器 5	0x0000_0000
BAKREG6	0x218	备份寄存器 6	0x0000_0000
BAKREG7	0x21C	备份寄存器 7	0x0000_0000
BAKREG8	0x220	备份寄存器 8	0x0000_0000
BAKREG9	0x224	备份寄存器 9	0x0000_0000

## 寄存器描述

### 备份域状态寄存器 – BAKSR

该寄存器定义了备份域状态。

偏移量: 0x100

复位值: 0x0000\_0001 (仅由备份域复位来复位)

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							WUPF
类型 / 复位							RC 0
7	6	5	4	3	2	1	0
保留位						PDF	BAKPORF
类型 / 复位						RC 0 RC 1	

位	字段	描述
[8]	WUPF	外部 WAKEUP 引脚标志位 0: WAKEUP 引脚无效 1: WAKEUP 引脚有效 当 WAKEUP 引脚有效时, 该位由硬件置位, 当软件读取时, 该位清零。系统从省电模式中唤醒后, 软件应读取该位使其清零。
[1]	PDF	暂停标志位 0: 从异常 V <sub>DD15</sub> 关机中唤醒 (V <sub>DD15</sub> 掉电是意料之外的) 1: 从暂停模式中唤醒 (V <sub>DD15</sub> 掉电是预期的) 当系统成功进入暂停模式, 该位由硬件置位。该位由软件读取清零。
[0]	BAKPORF	备份域复位标志位 0: 备份域复位未发生 1: 备份域复位发生 当备份域复位发生时, 无论是备份域上电复位还是备份域软件复位, 该位由硬件置位, 由软件读取清零。系统首次启动后, 该位必须被清零, 否则备份域复位触发时将无法检测到。当该位读为 1 时, 必须执行连续软件读取, 直到此位再次回到 0 为止。软件连续读取是为了确认备份域是否访问就绪。在备份域首次上电后必须执行连续软件读取。



## 备份域控制寄存器 – BAKCR

该寄存器定义了深度休眠和暂停模式下的电源控制位。

偏移量: 0x104

复位值: 0x0000\_0000 (仅由备份域复位来复位)

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位							
15	14	13	12	11	10	9	8
DMOSSTS	保留位		V15RDYSC	保留位		WUPIEN	WUPEN
RO 0			RW 0			RW 0	RW 0
7	6	5	4	3	2	1	0
DMOSON	保留位	LDOFTRM		LDOOFF	LDOLCM	保留位	BAKRST
RW 0		RW 0	RW 0	RW 0	RW 0		WO 0

位	字段	描述
[15]	DMOSSTS	DMOS 状态位 如果该寄存器中的 DMOSON 位被设置为 1，该位被置 1。 如果 DMOSON 位被清除为 0 或 POR / PDR 复位发生，该位被清零。
[12]	V15RDYSC	V <sub>DD15</sub> 就绪源选择位 0: 位于 CKCU 的 LPCR 寄存器中的 BKISO 位 1: V <sub>DD15</sub> POR 设置该位，以确定哪些隔离单元的控制信号是用来除能 V <sub>DD15</sub> 到 V <sub>DD</sub> 电源域电平转换器的隔离功能。
[9]	WUPIEN	外部 WAKEUP 引脚中断使能位 0: 除能外部 WAKEUP 引脚中断功能 1: 使能外部 WAKEUP 引脚中断功能 当 WUPEN 和 WUPF 位都被设置为 1 时，软件可以设置 WUPIEN 位为 1 来触发 NVIC 单元的 LPWUP 中断。
[8]	WUPEN	外部 WAKEUP 引脚使能位 0: 除能外部 WAKEUP 引脚功能 1: 使能外部 WAKEUP 引脚功能 在进入省电模式之前，软件可以设置 WUPEN 位为 1 以使能 WAKEUP 引脚功能。由于 WAKEUP 引脚是高电平有效，该引脚应设置为输入下拉模式。 注：该位由系统复位或备份域复位来复位。因为该位位于备份域，复位启动后，会有一定的延迟，直到该位有效。该位将一直无效，直到系统复位完成，备份域 ISO 信号除能。这意味着在系统复位完成，备份域 ISO 信号除能后，该位不能立即由软件置位。至少需延迟三个 32 kHz 时钟周期，直到该位的复位活动结束。
[7]	DMOSON	DMOS 控制位 0: DMOS 关闭 1: DMOS 开启 当 CPU 进入深度休眠模式 (SLEEPDEEP = 1)，DMOS 运行作为 1.5 V 电源域的另一个电压源。控制位 DMOSON 由软件置位，由软件或 PORB 清零。如果 DMOSON 位设置为 1，当 CPU 进入深度休眠模式，LDO 将自动关闭。

位	字段	描述
[5:4]	LDOFTRM	LDO 输出电压微调 00: LDO 默认输出电压 01: LDO 默认输出电压偏移量 -5 % 10: LDO 默认输出电压偏移量 +3 % 11: LDO 默认输出电压偏移量 +7 % 当 LDO 进入暂停模式或 V <sub>DD</sub> 电源域复位, 该位段将被清零。
[3]	LDOOFF	LDO 工作模式控制位 0: 当 CPU 进入深度休眠模式 (SLEEPDEEP = 1), LDO 工作在小电流模式。V <sub>DD15</sub> 电源可用。 1: 当 CPU 进入深度休眠模式 (SLEEPDEEP = 1), LDO 关闭。V <sub>DD15</sub> 电源不可用。 注: 该位仅在 DMOSON 位清除为 0 时有效。
[2]	LDOLCM	LDO 小电流模式 0: LDO 工作在正常电流模式 1: LDO 工作在小电流模式 注: 该位仅在 CPU 运行时可用。当 LDOLCM 置位时, LDO 输出电流的能力将限制在 10 mA 以下, 有较小的静态电流。适用于 CPU 工作在低速系统时钟时, 可具有较小功耗。该位当 LDO 暂停或 V <sub>DD</sub> 电源域复位时清零。
[0]	BAKRST	备份域软件复位 0: 无动作 1: 备份域软件复位启动 – 包括相关的 RTC 和 PWRCU 寄存器

备份域测试寄存器 – BAKTEST

该寄存器为软件定义了一个只读值来确认备份域是否准备访问。

偏移量: 0x108  
复位值: 0x0000\_0027

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	保留位							
	7	6	5	4	3	2	1	0
类型 / 复位	BAKTEST							
	RO	0	RO	0	RO	1	RO	0
	RO	0	RO	0	RO	1	RO	1
	RO	0	RO	0	RO	1	RO	1

位	字段	描述
[7:0]	BAKTEST	备份域测试位 当备份域为 CPU 的访问做好准备, 常量 0x27 将被读取。



低电压 / 欠压检测控制和状态寄存器 – LVDCSR

该寄存器定义了低电压检测器和欠压检测器的标志位、使能位和选项位。

偏移量: 0x110  
复位值: 0x0000\_0000 ( 仅由备份域复位而复位 )

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位	LVDS [2]	LVDEWEN	LVDIWEN	LVDF	LVDS [1:0]		LVDEN
		RW	0	RW	0	RW	0	RW
类型 / 复位	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位	7	6	5	4	3	2	1	0
	保留位				BODF	保留位	BODRIS	BODEN
					RO	0	RW	0

位	字段	描述
[21]	LVDEWEN	LVD 事件唤醒使能位 0: LVD 事件唤醒除能 1: LVD 事件唤醒使能 将该位设置为 1，将使能 LVD 事件唤醒功能，当低电压事件发生时，会唤醒系统并导致 LVDF 置位。如果系统需要通过 LVD 条件从深度休眠模式或暂停模式中唤醒，该位必须设为 1。
[20]	LVDIWEN	LVD 中断唤醒使能位 0: LVD 中断唤醒除能 1: LVD 中断唤醒使能 将该位设置为 1，将使能 LVD 中断功能。当 LVD 条件发生且 LVDIWEN 位设为 1，将发生 LVD 中断并发送到 CPU NVIC 单元。
[19]	LVDF	低电压检测状态标志位 0: V <sub>DD</sub> 高于指定的电压 1: V <sub>DD</sub> 等于或低于指定的电压 当 LVD 条件产生，LVDF 标志位将有效。当 LVDF 标志位被置位，如果 LVDIWEN 位设为 1，将发生 LVD 中断并发送到 CPU。但是，如果 LVDEWEN 位设为 1 且 LVDIWEN 清为 0，当 LVDF 标志位被置位时，将只发生 LVD 事件不产生 LVD 中断。
[22], [18:17]	LVDS [2:0]	低电压检测电平选择位 有关 LVD 可编程阈值电压的详细信息，请参考相关规格书中的电气特性部分。
[16]	LVDEN	低电压检测使能位 0: 除能低电压检测 1: 使能低电压检测 将该位设置为 1，当 V <sub>DD</sub> 电源低于由 LVDS 位设置的电压时，将发生 LVD 事件。因此，当该位在系统进入深度休眠模式 2 ( DMOS 打开, LDO 关闭 ) 或暂停模式 ( DMOS 和 LDO 都关闭 ) 之前被使能，则 LVDEWEN 位必须被使能，以避免在 CPU 由低电压检测功能唤醒期间 LDO 无法启动。
[3]	BODF	欠压检测标志位 0: V <sub>DD</sub> > V <sub>BOD</sub> 1: V <sub>DD</sub> ≤ V <sub>BOD</sub>

位	字段	描述
[1]	BODRIS	BOD 复位或中断选择位 0: 复位整个单片机 1: 产生中断
[0]	BODEN	欠压检测器使能位 0: 除能欠压检测器 1: 使能欠压检测器

备份寄存器 n – BAKREGn, n = 0 ~ 9

该寄存器定义了备份寄存器 n 用于在 VDD15 断电期间数据存储。

偏移量: 0x200 ~ 0x224  
复位值: 0x0000\_0000 ( 仅由备份域复位来复位 )

	31	30	29	28	27	26	25	24
	BAKREGn							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	BAKREGn							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	BAKREGn							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	BAKREGn							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:0]	BAKREGn	备份寄存器 n (n = 0 ~ 9) 这些寄存器用于通用数据存储。即使 VDD15 电源中断，BAKREGn 寄存器的内容还是会保存。

## 6 时钟控制单元 (CKCU)

### 简介

时钟控制单元 CKCU 提供了多种频率和时钟功能。包括内部高速 RC 振荡器 (HSI)、外部高速晶体振荡器 (HSE)、内部低速 RC 振荡器 (LSI)、外部低速晶体振荡器 (LSE)、锁相环 (PLL)、HSE 时钟监控、时钟分频器、时钟多路复用器和时钟门控电路。AHB、APB 和 CPU 的时钟来自系统时钟 CK\_SYS，而系统时钟可以来源于 HSI、HSE、LSI、LSE 或 PLL。看门狗定时器和实时时钟 RTC 使用 LSI 或 LSE 作为其时钟源。系统时钟  $f_{CK\_AHB}$  的最大工作频率可达 96 MHz。

一些内部时钟也可以通过 CKOUT 引脚引出用于调试。时钟监控电路可用于检测 HSE 时钟故障。一旦 HSE 时钟停止运行，不管什么原因，CKCU 将迫使系统时钟源切换到 HSI 时钟，以防止发生系统暂停。

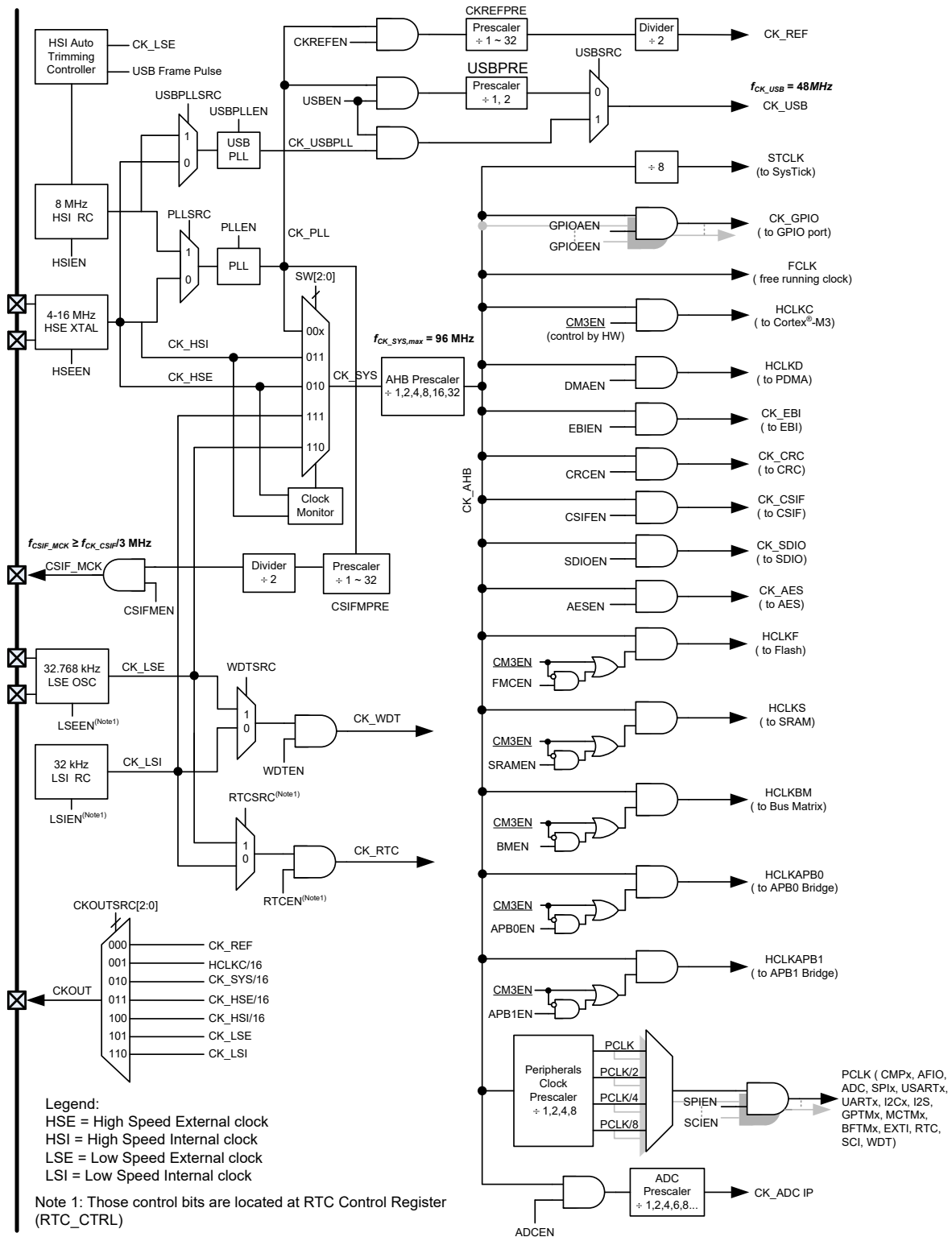


图 13. CKCU 方框图

## 特性

- 4 ~ 16 MHz 外部高速晶体振荡器 – HSE
- 8 MHz 内部高速 RC 振荡器 – HSI，具有配置选项校准和自定义微调功能。
- PLL 可选时钟源 (来自于 HSE 或 HSI) 用于系统时钟
- 32,768 Hz 外部低速晶体振荡器 – LSE，用于看门狗定时器、RTC 或系统时钟
- 32 kHz 内部低速 RC 振荡器 – LSI，用于看门狗定时器、RTC 或系统时钟
- HSE 时钟监控

## 功能描述

### 外部高速晶振 – HSE

外部高速晶体振荡器 HSE 的频率为 4 ~ 16 MHz，可以产生高精度度的时钟源作为系统时钟使用。相关硬件配置详见下图。一个特定频率的晶体需连接到两个 HSE 引脚 XTALIN / XTALOUT，并尽可能接近单片机。为了适当的振荡，外部电阻和电容元件连接到晶体上是必要的。

下面内容介绍了提高晶体电路布局的稳定性。

- 晶体振荡器应尽可能接近单片机，布线长度尽量缩短以减少寄生电容。
- 通过铺地接线来屏蔽晶体附近的任何线路以隔离信号和减小噪声。
- 保持频繁切换信号线离开晶体区，以防止串扰。

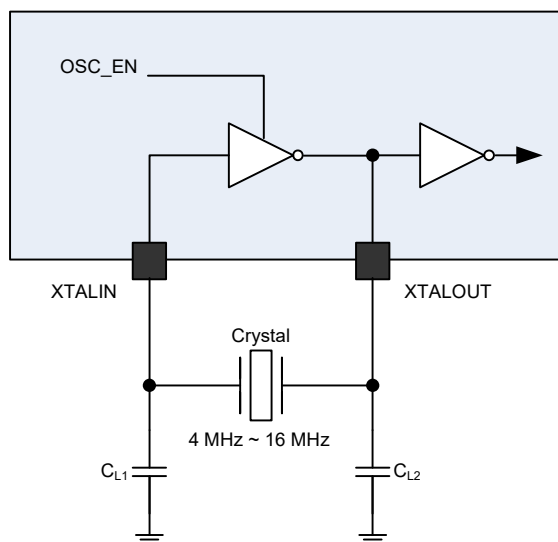


图 14. HSE 外部晶体、陶瓷和谐振器

HSE 晶体振荡器可以使用全局时钟控制寄存器 GCCR 中的 HSEEN 位开启或关闭。全局时钟状态寄存器 GCSR 中的 HSERDY 标志位表示外部高速晶体振荡器是否稳定。当开启 HSE 时, HSE 时钟不会被释放来使用, 直到 HSERDY 位由硬件置位。这个特定的延迟时间被称为振荡器的“启动时间”。当 HSE 变稳定, 如果全局时钟中断寄存器 GCIR 中相关的中断使能位 HSERDYIE 被置位, 将产生一个中断。此时, HSE 时钟可以直接用作系统时钟源或 PLL 输入时钟。

### 内部高速 RC 振荡器 – HSI

内部高速 8 MHz RC 振荡器 HSI 是单片机上电时 CPU 的默认时钟源。HSI RC 振荡器提供了一个较低成本类型的时钟源, 无需外部元件。HSI RC 振荡器可以使用全局时钟控制寄存器 GCCR 中的 HSIEN 位开启或关闭。全局时钟状态寄存器 GCSR 中的 HSIRDY 标志位用来指出内部 RC 振荡器是否稳定。HSI 振荡器的启动时间比 HSE 晶体振荡器短。当 HSI 变得稳定, 如果全局时钟中断寄存器 GCIR 中相关的中断使能位 HSIRDYIE 被置位, 将产生一个中断。HSI 时钟还可以用作 PLL 的输入时钟。

HSI 的频率精度可由配置选项校准, 但其工作频率仍不如 HSE 准确。由应用需求、环境和成本决定选择哪种振荡器类型。

当从深度休眠或暂停模式唤醒时, 软件可以设置 PSRCEN 位 (省电唤醒 RC 时钟使能位) 为 1, 以迫使 HSI 时钟作为系统时钟。随后, 当初时钟源 HSE 或 PLL 就绪标志位被置位, 系统时钟会自动切换回初始时钟源 HSE 或 PLL。当使用 HSE 或 PLL 作为系统时钟, 此功能会减少唤醒时间。

### 自动微调内部高速 RC 振荡器 – HSI

内部高速 RC 振荡器 HSI 的频率精确度可随着单片机制造过程的变化而不同。也就是说, 工厂校准后的单片机精度在  $V_{DD} = 3.3V$  和  $T_A = 25^\circ C$  条件下是  $\pm 2\%$ , 但这个精度不一定会满足一些应用领域和环境需求。因此, 单片机提供了 HSI 频率校准的微调机制, 使用更加精准的外部参考时钟。详细的方框图请参考下图。

复位后, 工厂微调值加载到 HSI 控制寄存器 HSICR 中的 HSICOARSE[4:0] 位和 HSIFINE[7:0] 位。HSI 的频率精确度会受到电压和温度改变而影响。如果应用需要更精准的 HSI 频率, 用户可以使用 HSI 控制寄存器 HSICR 中的 HSIFINE[7:0] 位手动微调, 或通过外部参考时钟使用自动微调控制器 ATC 来自动调整。参考时钟可以来自于以下时钟源:

- 32,768 Hz 的外部低速晶振或陶瓷谐振器振荡器 LSE 输出时钟
- 1 kHz USB SOF 封包接收
- 外部引脚 CKIN 1 kHz 脉冲

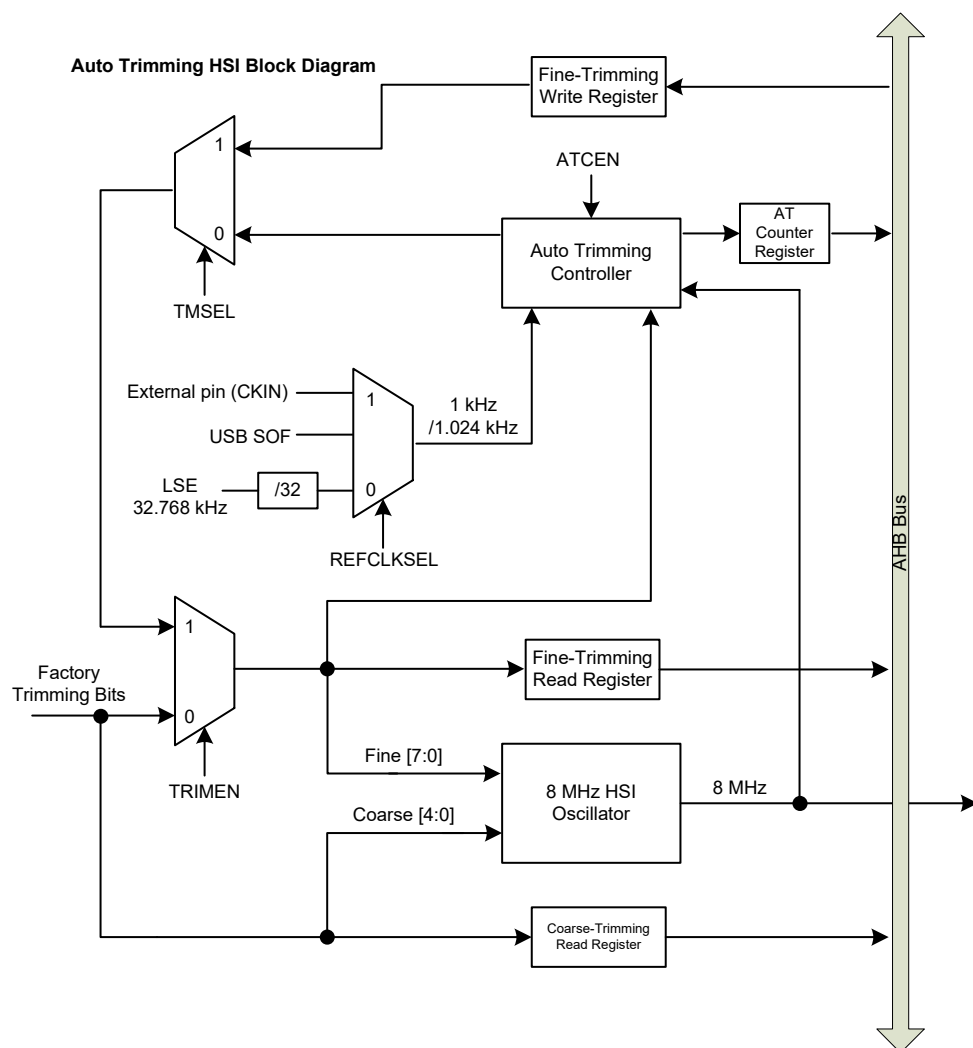


图 15. HSI 自动微调方框图

## 锁相环 – PLL

内部锁相环 PLL 可以提供 8 ~ 96 MHz 的时钟输出，是 4 ~ 16 MHz 基本参考频率的 2 ~ 24 倍。时钟合成器的原理依赖于 PLL，包括参考分频器、反馈分频器、数字相位频率检测器 PFD、电流控制的充电泵 CP、内部环路滤波器和电压控制的振荡器 VCO，以实现稳定的锁相状态。

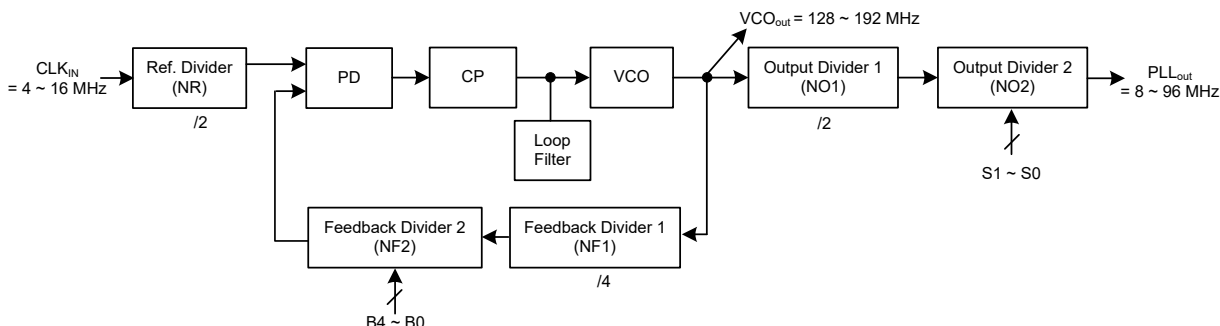


图 16. PLL 方框图

PLL 输出时钟频率可由以下公式决定：

$$PLL_{OUT} = CLK_{IN} \times \frac{NF1 \times NF2}{NR \times NO1 \times NO2} = CLK_{IN} \times \frac{4 \times NF2}{2 \times 2 \times NO2} = CLK_{IN} \times \frac{NF2}{NO2}$$

这里，NR = 参考分频器 = 2，NF1 = 反馈分频器 1 = 4，NF2 = 反馈分频器 2 = 1 ~ 32，NO1 = 输出分频器 1 = 2，NO2 = 输出分频器 2 = 1、2、4 或 8。

考虑到占空比为 50 % 时输入和输出频率都除以 2。如果一个给定的 PLL 输入时钟源频率  $CLK_{IN}$ ，将产生一个特定的 PLL 输出频率，那么建议使用一个更大的 NF2 值，以增加 PLL 稳定性并且利用调整时间来减少抖动。对输出和反馈分频器 2 设置位的描述如下两表。下表中所有的设置位 S1 ~ S0 和 B4 ~ B0，定义在寄存器定义章节的 PLL 配置寄存器 PLLCFGR 和 PLL 控制寄存器 PLLCR 中。注意， $VCO_{OUT}$  频率必须在 128 MHz ~ 192 MHz 范围内。如果所选的配置超出这个范围，则 PLL 输出频率不能保证符合上述  $PLL_{OUT}$  公式。

PLL 可以通过使用全局时钟控制寄存器 GCCR 中的 PLEN 位开启或关闭。全局时钟状态寄存器 GCSR 中的 PLLRDY 标志位表示 PLL 时钟是否稳定。当 PLL 变得稳定，如果全局时钟中断寄存器 GCIR 中相关的中断使能位 PLLRDYIE 被置位，将产生一个中断。



表 15. 输出分频器 2 值映射

输出分频器 2 设置位 S1 ~ S0 ( POTD 位 )	NO2 ( 输出分频器 2 的值 )
00	1
01	2
10	4
11	8

表 16. 反馈分频器 2 值映射

反馈分频器 2 设置位 B4 ~ B0 ( PFBD 位 )	NF2 ( 反馈分频器 2 的值 )
00000	32
00001	1
00010	2
00011	3
00100	4
00101	5
00110	6
00111	7
01000	8
01001	9
01010	10
01011	11
:	:
:	:
11110	30
11111	31

## USB 锁相环 – USB PLL

USB PLL 可以提供 4 ~ 48 MHz 的 USB 外设时钟, 是 4 ~ 16 MHz 基本参考频率的 2 ~ 24 倍。时钟合成器的原理依赖于数字 PLL, 包括参考分频器、反馈分频器、数字相位频率检测器 PFD、电流控制的充电泵 CP、内部环路滤波器和电压控制的振荡器 VCO, 以实现稳定的锁相状态。

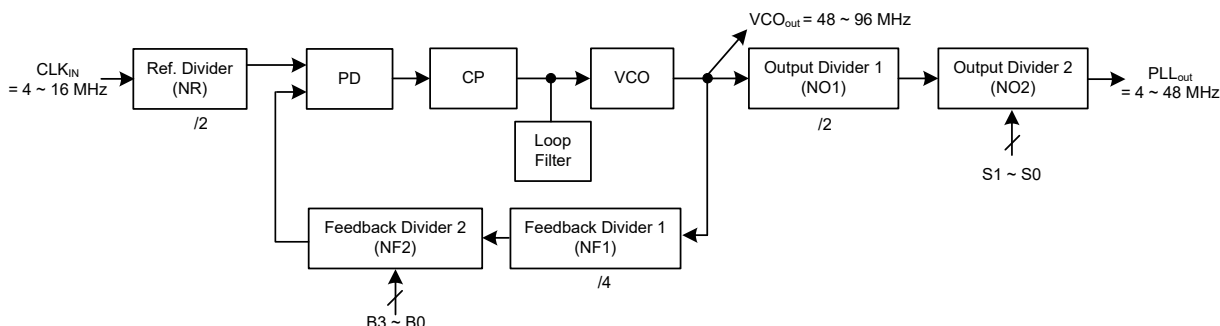


图 17. USB PLL 方框图

PLL 输出时钟频率可由以下公式决定:

$$PLL_{OUT} = CLK_{IN} \times \frac{NF1 \times NF2}{NR \times NO1 \times NO2} = CLK_{IN} \times \frac{4 \times NF2}{2 \times 2 \times NO2} = CLK_{IN} \times \frac{NF2}{NO2}$$

这里, NR = 参考分频器 = 2, NF1 = 反馈分频器 1 = 4, NF2 = 反馈分频器 2 = 1 ~ 16, NO1 = 输出分频器 1 = 2, NO2 = 输出分频器 2 = 1、2、4 或 8。

考虑到占空比为 50 % 时输入和输出频率都除以 2。如果一个给定的 USB PLL 输入时钟源频率  $CLK_{IN}$ , 将产生一个特定的 USB PLL 输出频率, 那么建议使用一个更大的 NF2 值, 以增加 PLL 稳定性并且利用调整时间来减少抖动。对输出和反馈分频器 2 设置位的描述如下两表。下表中所有的设置位 S1 ~ S0 和 B3 ~ B0, 定义在寄存器定义章节的 PLL 配置寄存器 PLLCFGR 和 PLL 控制寄存器 PLLCR 中。注意,  $VCO_{OUT}$  频率必须在 48 MHz ~ 96 MHz 范围内。如果所选的配置超出这个范围, 则 PLL 输出频率不能保证符合上述  $PLL_{OUT}$  公式。

USB PLL 可以通过使用全局时钟控制寄存器 GCCR 中的 USBPllen 位开启或关闭。全局时钟状态寄存器 GCSR 中的 USBPLLrdy 标志位表示 USB PLL 时钟是否稳定。当 USB PLL 变得稳定, 如果全局时钟中断寄存器 GCIR 中相关的中断使能位 USBPLLrdyie 被置位, 将产生一个中断。

表 17. USB PLL 输出分频器 2 值映射

输出分频器 2 设置位 S1 ~ S0 ( USBPOTD 位 )	NO2 ( 输出分频器 2 的值 )
00	1
01	2
10	4
11	8

表 18. USB PLL 反馈分频器 2 值映射

反馈分频器 2 设置位 B3 ~ B0 ( USBPFBD 位 )	NF2 ( 反馈分频器 2 的值 )
0000	16
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
:	:
:	:
1111	15

## 外部低速晶振 – LSE

外部低速晶体振荡器或陶瓷谐振器频率为 32768 Hz, 可以产生一个低功率但高精度的时钟源用于实时时钟电路、看门狗定时器或系统时钟。相关的硬件配置见下图。晶体或陶瓷谐振器必须靠近两个 LSE 引脚, X32KIN 和 X32KOUT。为达到适当的振荡, 其外部电阻和电容元件是必要的。LSE 振荡器可以使用 RTC 控制寄存器 RTCCR 中的 LSEEN 位开启或关闭。全局时钟状态寄存器 GCSR 中的 LSERDY 标志位将指出 LSE 时钟是否稳定。当 LSE 变得稳定, 如果全局时钟中断寄存器 GCIR 中相关的中断使能位 LSERDYIE 被置位, 将产生一个中断。

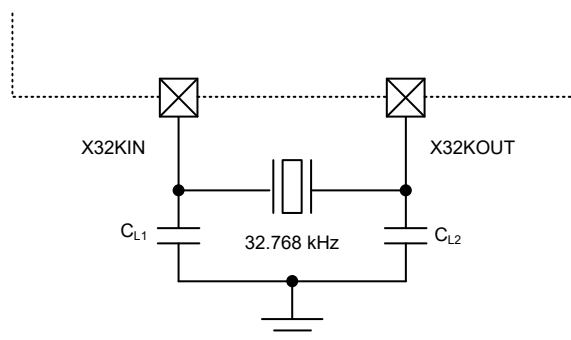


图 18. LSE 外部晶体、陶瓷和谐振器

## 内部低速 RC 振荡器 – LSI

内部低速 RC 振荡器的频率约为 32 kHz, 是一个低功耗的时钟源用于实时时钟电路或看门狗定时器。LSI 提供了一个低成本的时钟源, 无需外部元件。LSI RC 振荡器可以通过使用 RTC 控制寄存器 RTCCR 中的 LSIEN 位开启或关闭。LSI 频率精度可在规格书中查看。全局时钟状态寄存器 GCSR 中的 LSIRDY 标志位表示 LSI 时钟是否稳定。当 LSI 变得稳定, 如果全局时钟中断寄存器 GCIR 中相关的中断使能位 LSIRDYIE 被置位, 将产生一个中断。

## 时钟就绪标志位

CKCU 为 HSI、HSE、PLL、LSI 和 LSE 提供相关时钟就绪标志位, 用以表示这些时钟是否稳定。在使用这些时钟作为系统时钟或其它用途之前, 必须保证指定的时钟标志位已置位。可以通过轮询 GCSR 寄存器内的各自的时钟就绪状态位来检测指定的时钟是否就绪。此外, 如果 GCIR 寄存器的相关中断使能位置位, 则 CKCU 产生中断用以表示指定的时钟已就绪。可通过中断服务子程序来软件清零 GCIR 寄存器中的中断状态位。

## 系统时钟选择 – CK\_SYS

系统复位后，默认的系统时钟源 CK\_SYS 为 HSI，通过改变全局时钟控制寄存器 GCCR 中的系统时钟切换位 SW 值可以切换系统时钟为 HSI、HSE、LSE、LSI 或 PLL 输出时钟。当 SW 值发生改变，CK\_SYS 将使用初始时钟源继续运行，直到目标时钟源稳定。相应的全局时钟状态寄存器 GCSR 中的相应的时钟就绪状态位将显示所选的时钟是否已就绪，时钟源状态寄存器 CKST 则用于显示当前作为系统时钟源的时钟。当一个时钟源或 PLL 时钟被用作系统时钟源，停止它是不可能的。关于时钟使能功能的描述如下。

如果下面的任一动作发生时，HSI 将被使能。

- 使能 PLL 并配置 HSI 为其时钟源 (PLLEN, PLLSRC)
- 使能时钟监控器 (CKMEN)
- 配置时钟切换寄存器切换到 HSI (SW)
- 配置 HSI 使能寄存器为 1 (HSIEN)

如果下面的任一动作发生时，HSE 将被使能。

- 使能 PLL 并配置 HSE 为其时钟源 (PLLEN, PLLSRC)
- 配置时钟切换寄存器到 HSE (SW)
- 配置 HSE 使能寄存器为 1 (HSEEN)

如果下面的任一动作发生时，PLL 将被使能。

- 使能 USB 使能寄存器 (USBEN)
- 配置时钟切换寄存器到 PLL (SW)
- 配置 PLL 使能寄存器为 1 (PLLEN)

系统时钟选择的编程指南如下所示。

- 使能将作为系统时钟或 PLL 输入时钟的任意一个时钟源。
- 在 HSI 和 HSE 的就绪标志位都有效之后，配置 PLLSRC 寄存器。
- 在时钟源就绪标志位置起后配置 SW 寄存器以改变系统时钟源。注意，如果时钟监控器使能，且被配置为系统时钟的 PLL 输出或 HSE 时钟卡在了 0 或 1 时，将强制 HSI 作为系统时钟。

## HSE 时钟监测器

HSE 时钟监控功能由全局时钟控制寄存器 GCCR 中的 HSE 时钟监控使能位 CKMEN 开启。此功能应该在 HSE 振荡器启动延迟后开启，在 HSE 停止时关闭。一旦检测到 HSE 故障，HSE 将自动除能。如果全局时钟中断寄存器 GCIR 中的相关中断使能位 CKSIE 被置位，全局时钟中断寄存器 GCIR 中的 HSE 时钟故障标志位 CKSF 将被置位，并产生 HSE 故障事件。这种故障中断被连接到 CPU 的非可屏蔽中断 NMI 中。当 HSE 振荡器故障发生，HSE 将关闭且系统时钟将由硬件自动切换到 HSI。当故障发生时，如果选择 HSE 作为 PLL 电路的时钟输入，PLL 时钟输出作为系统时钟，PLL 电路将关闭，同时，HSE 也将关闭。

## 时钟输出能力

单片机有时钟输出能力，允许时钟输出在特定的外部输出引脚 CKOUT 上。相应的 GPIO 引脚应该被配置在适当的复用功能 I/O (AFIO) 模式以输出所选的时钟信号。有几种时钟信号可通过全局时钟配置寄存器 GCFGR 中的 CKOUT 时钟源选择位 CKOUTSRC 来选择。

表 19. CKOUT 的时钟源

CKOUTSRC[2:0]	时钟源
000	$CK\_REF = CK\_PLL / (CKREFPRE + 1) / 2$
001	CK_AHB / 16
010	CK_SYS / 16
011	CK_HSE / 16
100	CK_HSI / 16
101	CK_LSE
110	CK_LSI

## 寄存器列表

下表显示了CKCU 寄存器及其复位值。

表 20. CKCU 寄存器列表

寄存器	偏移量	描述	复位值
GCFGR	0x000	全局时钟配置寄存器	0x0000_0302
GCCR	0x004	全局时钟控制寄存器	0x0000_0803
GCSR	0x008	全局时钟状态寄存器	0x0000_0028
GCIR	0x00C	全局时钟中断寄存器	0x0000_0000
PLLCFGR	0x018	PLL 配置寄存器	0x0000_0000
PLLCR	0x01C	PLL 控制寄存器	0x0000_0000
AHBCFGR	0x020	AHB 配置寄存器	0x0000_0000
AHBCCR	0x024	AHB 时钟控制寄存器	0x0000_00E5
APBCFGR	0x028	APB 配置寄存器	0x0001_0000
APBCCR0	0x02C	APB 时钟控制寄存器 0	0x0000_0000
APBCCR1	0x030	APB 时钟控制寄存器 1	0x0000_0000
CKST	0x034	时钟源状态寄存器	0x0100_0003
APBPCSR0	0x038	APB 外设时钟选择寄存器 0	0x0000_0000
APBPCSR1	0x03C	APB 外设时钟选择寄存器 1	0x0000_0000
HSICR	0x040	HSI 控制寄存器	0xXXXX_0000 (X 表示未定义)
HSIATCR	0x044	HSI 自动微调计数寄存器	0x0000_0000
LPCR	0x300	低功耗控制寄存器	0x0000_0000
MCUDBGCR	0x304	MCU 调试控制寄存器	0x0000_0000

## 寄存器描述

### 全局时钟配置寄存器 – GCFGR

该寄存器用于为 PLL / USB / CKOUT 电路指定时钟源。

偏移量: 0x000

复位值: 0x0000\_0302

	31	30	29	28	27	26	25	24	
	LPMOD				CSIFMPRE				
类型 / 复位	RO	0	RO	0	RO	0	RO	0	
	23	22	21	20	19	18	17	16	
	USBPRE		保留位						
类型 / 复位	RW	0	RW	0					
	15	14	13	12	11	10	9	8	
	CKREFPRE					USBSRC	USBPLLSRC	PLLSRC	
类型 / 复位	RW	0	RW	0	RW	0	RW	1	
	7	6	5	4	3	2	1	0	
	保留位					CKOUTSRC			
类型 / 复位						RW	0	RW	
							1	RW	
								0	

位	字段	描述
[31:29]	LPMOD	低功耗模式状态 000: 单片机处于运行状态 001: 单片机要进入休眠模式 010: 单片机要进入深度休眠模式 1 011: 单片机要进入深度休眠模式 2 100: 单片机要进入暂停模式 其它值: 保留 由硬件置位和复位。
[28:24]	CSIFMPRE	CSIF_MCK 时钟预分频选择 $CK\_MCK = CK\_PLL / (CSIFMPRE + 1) / 2$ 00000: $CK\_MCK = CK\_PLL / 2$ 00001: $CK\_MCK = CK\_PLL / 4$ ... 11111: $CK\_MCK = CK\_PLL / 64$ 由软件置位和复位来控制 CSIF_MCK 时钟预分频值。
[23:22]	USBPRE	USB 时钟预分频选择 00: $CK\_USB = CK\_PLL$ 01: $CK\_USB = CK\_PLL / 2$ 其它值: 保留 由软件置位和复位来控制 USB 时钟的预分频值。
[15:11]	CKREFPRE	CK_REF 时钟预分频选择 $CK\_REF = CK\_PLL / (CKREFPRE + 1) / 2$ 00000: $CK\_REF = CK\_PLL / 2$ 00001: $CK\_REF = CK\_PLL / 4$ ... 11111: $CK\_REF = CK\_PLL / 64$ 由软件置位和复位来控制 CK_REF 时钟预分频值。



位	字段	描述
[10]	USBSRC	USB 时钟源选择位 0: CK_PLL 时钟 1: CK_USBPLL 时钟 由软件置位和复位来控制 USB 的时钟源。
[9]	USBPLLSRC	USB PLL 时钟源选择位 0: 外部 4 ~ 16 MHz 晶振 (HSE) 1: 内部 8 MHz RC 振荡器 (HSI) 由软件置位和复位来控制 USB PLL 的时钟源。
[8]	PLLSRC	PLL 时钟源选择位 0: 外部 4 ~ 16 MHz 晶振 (HSE) 1: 内部 8 MHz RC 振荡器 (HSI) 由软件置位和复位来控制 PLL 的时钟源。
[2:0]	CKOUTSRC	CKOUT 时钟源选择位 000: CK_REF, $CK\_REF = CK\_PLL / (CKREFPRE + 1) / 2$ 001: HCLKC / 16 010: CK_SYS / 16 011: CK_HSE / 16 100: CK_HSI / 16 101: CK_LSE 110: CK_LSI 111: 保留位 由软件置位和复位来控制 CKOUT 的时钟源。

## 全局时钟控制寄存器 – GCCR

该寄存器定义了时钟使能位。

偏移量: 0x004

复位值: 0x0000\_0803

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位						PSRCEN	CKMEN
类型 / 复位						RW 0	RW 0
15	14	13	12	11	10	9	8
保留位				HSIEN	HSEEN	PLLEN	HSEGAIN
类型 / 复位				RW 1	RW 0	RW 0	RW 0
7	6	5	4	3	2	1	0
保留位				USBPLEN	SW		
类型 / 复位				RW 0	RW 0	RW 1	RW 1

位	字段	描述
[17]	PSRCEN	省电唤醒 RC 时钟使能位 0: 无动作 1: 从深度休眠 1 / 深度休眠 2 唤醒后, 使用内部 8 MHz RC 时钟 (HSI) 作为系统时钟。在进入深度休眠 1 / 深度休眠 2 模式之前, 软件可以设置 PSRCEN 位来减少唤醒等待时间。当 PSRCEN = 1, 在系统从深度休眠 1 / 深度休眠 2 模式唤醒后将选择 HSI 作为系统时钟源。因为 HSI 提供时钟给 CPU, 可以在初始时钟 CK_SYS 源稳定前执行指令。在初始时钟源 CK_SYS 就绪后, CK_SYS 将自动从 HSI 时钟切换到初始时钟源。
[16]	CKMEN	HSE 时钟监控使能位 0: 外部 4 ~ 16 MHz 晶振 HSE 时钟监控除能 1: 外部 4 ~ 16 MHz 晶振 HSE 时钟监控使能 当硬件检测出 HSE 时钟停留在低电平或高电平状态, 内部硬件将切换内部高速 RC 时钟 HSI 为系统时钟。
[11]	HSIEN	内部高速振荡器使能位 0: 内部 8 MHz RC 振荡器除能 1: 内部 8 MHz RC 振荡器使能 由软件置位和复位。如果使用 HSI 作为系统时钟, 该位不能被复位。
[10]	HSEEN	外部高速振荡器使能位 0: 外部 4 ~ 16 MHz 晶振除能 1: 外部 4 ~ 16 MHz 晶振使能 由软件置位和复位。如果使用 HSE 作为系统时钟或 PLL 的输入时钟, 该位不能被复位。
[9]	PLLEN	PLL 使能 0: PLL 关闭 1: PLL 开启 由软件置位和复位。如果 PLL 时钟被用作系统时钟, 此位不会被复位。
[8]	HSEGAIN	外部高速振荡器增益选择 0: HSE 在低增益模式 1: HSE 在高增益模式

位	字段	描述
[3]	USBPLEN	USB PLL 使能 0: USB PLL 关闭 1: USB PLL 开启 由软件置位和复位 USB PLL。如果 PLL 时钟被用作系统时钟，此位不会被复位。
[2:0]	SW	系统时钟切换 00x: 选择 CK_PLL 作为系统时钟 010: 选择 CK_HSE 作为系统时钟 011: 选择 CK_HSI 作为系统时钟 110: 选择 CK_LSE 作为系统时钟 111: 选择 CK_LSI 作为系统时钟 其它值: CK_HSI 作为系统时钟 通过软件置位和复位该位段来选择 CK_SYS 时钟源。当 HSE 时钟监控使能，如果 HSE 时钟故障被检测到，则 HSI 振荡器将强制作作为系统时钟。 注：当使用 SW 切换系统时钟时，系统时钟不会立即被切换，会有一定的延迟时间。 系统时钟的选择由 SW 位段控制，由时钟源状态寄存器 CKST 的 CKSWST 位段来显示当前作为系统时钟源的时钟。

全局时钟状态寄存器 – GCSR

该寄存器定义了时钟就绪状态。

偏移量: 0x008  
复位值: 0x0000\_0028

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位		LSIRDY	LSERDY	HSIRDY	HSERDY	PLLRDY	USBPLLRDY
RO		1 RO	0 RO	1 RO	0 RO	0 RO	0
类型 / 复位							

位	字段	描述
[5]	LSIRDY	内部低速振荡器就绪标志位 0: 内部 32 kHz RC 振荡器未就绪 1: 内部 32 kHz RC 振荡器就绪 由硬件置位以表示 LSI 振荡器是否稳定。
[4]	LSERDY	外部低速振荡器 LSE 就绪标志位 0: LSE 振荡器未就绪 1: LSE 振荡器就绪 由硬件置位以表示 LSE 振荡器是否稳定且准备就绪。

位	字段	描述
[3]	HSIRDY	内部高速振荡器就绪标志位 0: 内部 8 MHz RC 振荡器未就绪 1: 内部 8 MHz RC 振荡器就绪 由硬件置位以表示 HSI 振荡器是否稳定。
[2]	HSERDY	外部高速振荡器就绪标志位 0: 外部 4 ~ 16 MHz 振荡器未就绪 1: 外部 4 ~ 16 MHz 振荡器就绪 由硬件置位以表示 HSE 振荡器是否稳定。
[1]	PLLRDY	PLL 时钟就绪标志位 0: PLL 未就绪 1: PLL 就绪 由硬件置位以表示 PLL 时钟是否稳定。
[0]	USBPLLRDY	USB PLL 时钟就绪标志位 0: USB PLL 未就绪 1: USB PLL 就绪 由硬件置位以表示 USB PLL 时钟是否稳定。

## 全局时钟中断寄存器 – GCIR

该寄存器定义了中断使能位和标志位。

偏移量: 0x00C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位	LSIRDYIE	LSERDYIE	HSIRDYIE	HSERDYIE	PLLRDYIE	USBPLLRDYIE	CKSIE	
类型 / 复位		RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8	
	保留位								
类型 / 复位									
	7	6	5	4	3	2	1	0	
	保留位	LSIRDYF	LSERDYF	HSIRDYF	HSERDYF	PLLRDYF	USBPLLRDYF	CKSF	
类型 / 复位		WC	0	WC	0	WC	0	WC	0

位	字段	描述
[22]	LSIRDYIE	LSI 就绪中断使能位 0: LSI 稳定中断除能 1: LSI 稳定中断使能 该位由软件置位和复位来使能或除能 LSI 稳定中断。
[21]	LSERDYIE	LSE 就绪中断使能位 0: LSE 稳定中断除能 1: LSE 稳定中断使能 该位由软件置位和复位来使能或除能 LSE 稳定中断。

位	字段	描述
[20]	HSIRDYIE	HSI 就绪中断使能位 0: HSI 稳定中断除能 1: HSI 稳定中断使能 该位由软件置位和复位来使能或除能 HSI 稳定中断。
[19]	HSERDYIE	HSE 就绪中断使能位 0: HSE 稳定中断除能 1: HSE 稳定中断使能 该位由软件置位和复位来使能或除能 HSE 稳定中断。
[18]	PLLRDYIE	PLL 就绪中断使能位 0: PLL 稳定中断除能 1: PLL 稳定中断使能 该位由软件置位和复位来使能或除能 PLL 稳定中断。
[17]	USBPLLRDYIE	USB PLL 就绪中断使能位 0: USB PLL 就绪中断除能 1: USB PLL 就绪中断使能 该位由软件置位和复位来使能或除能 USB PLL 就绪中断。
[16]	CKSIE	时钟故障中断使能位 0: 时钟故障中断除能 1: 时钟故障中断使能 该位由软件置位和复位来使能或除能监测到的故障中断。
[6]	LSIRDYF	LSI 就绪中断标志位 0: 不产生 LSI 就绪中断 1: LSI 时钟稳定, 产生 LSI 就绪中断 通过写 1 来清零。当内部 32 kHz RC 振荡器时钟稳定且 LSIRDYIE 位被置位时, 可通过硬件置位。
[5]	LSERDYF	LSE 就绪中断标志位 0: 不产生 LSE 就绪中断 1: LSE 时钟稳定, 产生 LSE 就绪中断 通过写 1 来清零。当外部 32,768 Hz 晶振时钟稳定且 LSERDYIE 位被置位时, 可通过硬件置位。
[4]	HSIRDYF	HSI 就绪中断标志位 0: 不产生 HSI 就绪中断 1: HSI 时钟稳定, 产生 HSI 就绪中断 通过写 1 来清零。当内部 8 MHz RC 振荡器时钟稳定且 HSIRDYIE 位被置位时, 可通过硬件置位。
[3]	HSERDYF	HSE 就绪中断标志位 0: 不产生 HSE 就绪中断 1: HSE 时钟稳定, 产生 HSE 就绪中断 通过写 1 来清零。当外部 4 ~ 16 MHz 晶振时钟稳定且 HSERDYIE 位被置位时, 可通过硬件置位。
[2]	PLLRDYF	PLL 就绪中断标志位 0: 不产生 PLL 就绪中断 1: PLL 时钟稳定, 产生 PLL 就绪中断 通过写 1 来清零。当 PLL 稳定且 PLLRDYIE 位被置位时, 可通过硬件置位。
[1]	USBPLLRDYF	USB PLL 就绪中断标志位 0: 不产生 USB PLL 就绪中断 1: USB PLL 时钟稳定, 产生 USB PLL 就绪中断 通过写 1 来清零。当 USB PLL 稳定且 USBPLLRDYIE 位被置位时, 可通过硬件置位。

位	字段	描述
[0]	CKSF	HSE 时钟故障中断标志位 0: 时钟正常运行 1: HSE 时钟故障 通过写 1 来清零。当 HSE 时钟故障且 CKMEN 位被置位时，可通过硬件置位。

PLL 配置寄存器 – PLLCFGR

该寄存器定义了 PLL 的配置。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位				PFBD			
类型 / 复位					RW	0	RW	0
	23	22	21	20	19	18	17	16
	PFBD	POTD		保留位				
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	保留位					USBPFBD		
类型 / 复位						RW	0	RW
	7	6	5	4	3	2	1	0
	USBPFBD	USBPOTD		保留位				
类型 / 复位	RW	0	RW	0	RW	0		

位	字段	描述
[27:23]	PFBD	PLL VCO 输出时钟反馈分频器 (B4 ~ B0) PLL VCO 输出时钟经反馈分频器分频。
[22:21]	POTD	PLL 输出时钟分频器 (S1 ~ S0)
[10:7]	USBPFBD	USB PLL VCO 输出时钟反馈分频器 (B3 ~ B0) USB PLL VCO 输出时钟经反馈分频器分频。
[6:5]	USBPOTD	USB PLL 输出时钟分频器 (S1 ~ S0)

## PLL 控制寄存器 – PLLCR

该寄存器定义了 PLL 旁路模式。

偏移量: 0x01C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	PLLBPS	保留位						
类型 / 复位	RW	0						
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位							
类型 / 复位								

位	字段	描述
[31]	PLLBPS	PLL 旁路模式使能位 0: PLL 旁路模式除能 1: PLL 旁路模式使能, $F_{OUT} = F_{IN}$

**AHB 配置寄存器 – AHBCFGR**

该寄存器定义了系统时钟频率。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位				AHBPRE			
类型 / 复位					RW	0	RW	0

位	字段	描述
[2:0]	AHBPRE	AHB 预分频 000: CK_AHB = CK_SYS 001: CK_AHB = CK_SYS / 2 010: CK_AHB = CK_SYS / 4 011: CK_AHB = CK_SYS / 8 100: CK_AHB = CK_SYS / 16 101: CK_AHB = CK_SYS / 32 110: CK_AHB = CK_SYS / 32 111: CK_AHB = CK_SYS / 32 由软件置位和复位来控制 AHB 时钟的分频比。



## AHB 时钟控制寄存器 – AHBCCR

该寄存器定义了 AHB 时钟使能位。

偏移量： 0x024

复位值： 0x0000\_00E5

	31	30	29	28	27	26	25	24				
	保留位											
类型 / 复位												
	23	22	21	20	19	18	17	16				
	保留位			PEEN	PDEN	PCEN	PBEN	PAEN				
类型 / 复位				RW	0	RW	0	RW	0			
	15	14	13	12	11	10	9	8				
	AESSEN	SDIOEN	CRCEN	EBIEN	CKREFEN	USBEN	CSIFMEN	CSIFEN				
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0		
	7	6	5	4	3	2	1	0				
	APB1EN	APB0EN	BMEN	PDMAEN	保留位	SRAMEN	保留位	FMCEN				
类型 / 复位	RW	1	RW	1	RW	0		RW	1		RW	1

位	字段	描述
[20]	PEEN	通用 I/O 口 PE 时钟使能位 0: PE 时钟除能 1: PE 时钟使能 该位由软件置位和复位。
[19]	PDEN	通用 I/O 口 PD 时钟使能位 0: PD 时钟除能 1: PD 时钟使能 该位由软件置位和复位。
[18]	PCEN	通用 I/O 口 PC 时钟使能位 0: PC 时钟除能 1: PC 时钟使能 该位由软件置位和复位。
[17]	PBEN	通用 I/O 口 PB 时钟使能位 0: PB 时钟除能 1: PB 时钟使能 该位由软件置位和复位。
[16]	PAEN	通用 I/O 口 PA 时钟使能位 0: PA 时钟除能 1: PA 时钟使能 该位由软件置位和复位。
[15]	AESSEN	AES 模块时钟使能位 0: AES 时钟除能 1: AES 时钟使能 该位由软件置位和复位。
[14]	SDIOEN	SDIO 模块时钟使能位 0: SDIO 时钟除能 1: SDIO 时钟使能 该位由软件置位和复位。

位	字段	描述
[13]	CRCEN	CRC 模块时钟使能位 0: CRC 时钟除能 1: CRC 时钟使能 该位由软件置位和复位。
[12]	EBIEN	EBI 时钟使能位 0: EBI 时钟除能 1: EBI 时钟使能 该位由软件置位和复位。
[11]	CKREFEN	CK_REF 时钟使能位 0: CK_REF 时钟除能 1: CK_REF 时钟使能 该位由软件置位和复位。
[10]	USBEN	USB 时钟使能位 0: USB 时钟除能 1: USB 时钟使能 该位由软件置位和复位。
[9]	CSIFMEN	CSIF_MCK 时钟使能位 0: CSIF_MCK 时钟除能 1: CSIF_MCK 时钟使能 该位由软件置位和复位。
[8]	CSIFEN	CSIF 时钟使能位 0: CSIF 时钟除能 1: CSIF 时钟使能 该位由软件置位和复位。
[7]	APB1EN	APB1 桥时钟使能位 0: 在休眠模式下 APB1 桥时钟由硬件自动除能 1: 在休眠模式下 APB1 桥时钟始终使能 由软件置位和复位。如果 APB1 桥在休眠模式下未使用，用户可将 APB1EN 位清零以降低功耗。
[6]	APB0EN	APB0 桥时钟使能位 0: 在休眠模式下 APB0 桥时钟由硬件自动除能 1: 在休眠模式下 APB0 桥时钟始终使能 由软件置位和复位。如果 APB0 桥在休眠模式下未使用，用户可将 APB0EN 位清零以降低功耗。
[5]	BMEN	总线矩阵时钟使能位 0: 在休眠模式下总线矩阵时钟由硬件自动除能 1: 在休眠模式下总线矩阵时钟始终使能 由软件置位和复位。如果总线矩阵桥在休眠模式下未使用，用户可将 BMEN 位清零以降低功耗。
[4]	PDMAEN	外设 DMA 时钟使能位 0: PDMA 时钟除能 1: PDMA 时钟使能 由软件置位和复位。注意，当处理器处于休眠模式时，PDMA 可独立工作。但是，相关的 AHB 总线从机时钟或外设时钟必须使能。
[2]	SRAMEN	SRAM 时钟使能位 0: 在休眠模式下 SRAM 时钟由硬件自动除能 1: 在休眠模式下 SRAM 时钟始终使能 由软件置位和复位。如果 SRAM 在休眠模式下未使用，用户可将 SRAMEN 位清零以降低功耗。

位	字段	描述
[0]	FMCEN	Flash 存储控制器时钟使能位 0: FMC 时钟在休眠模式下由硬件自动除能 1: FMC 时钟在休眠模式下始终使能 由软件置位和复位。如果 Flash 存储器在休眠模式下未使用，用户可将 FMCEN 位清零以降低功耗。

APB 配置寄存器 – APBCFGR

该寄存器定义了 A/D 转换器时钟频率。

偏移量: 0x028  
复位值: 0x0001\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位					ADCDIV		
类型 / 复位							
				RW	0	RW	0
				RW	0	RW	1
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位							
类型 / 复位							

位	字段	描述
[18:16]	ADCDIV	A/D 转换器时钟分频选择位 000: CK_ADC = CK_AHB / 1 001: CK_ADC = CK_AHB / 2 010: CK_ADC = CK_AHB / 4 011: CK_ADC = CK_AHB / 8 100: CK_ADC = CK_AHB / 16 101: CK_ADC = CK_AHB / 32 110: CK_ADC = CK_AHB / 64 111: CK_ADC = CK_AHB / 6 由软件置位和复位来选择 A/D 转换器时钟分频比。

## APB 时钟控制寄存器 0 – APBCCR0

该寄存器具体定义了几种 APB 外设时钟使能位。

偏移量： 0x02C

复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位				SCI1EN	保留位	I2SEN	SCI0EN
					RW 0		RW 0	RW 0
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	EXTIEN	AFIOEN	保留位		UR1EN	UR0EN	USR1EN	USR0EN
	RW 0	RW 0			RW 0	RW 0	RW 0	RW 0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位		SPI1EN	SPI0EN	保留位		I2C1EN	I2C0EN
			RW 0	RW 0			RW 0	RW 0

位	字段	描述
[27]	SCI1EN	智能卡接口 1 时钟使能位 0: SCI1 时钟除能 1: SCI1 时钟使能 由软件置位和复位。
[25]	I2SEN	I <sup>2</sup> S 接口使能位 0: I <sup>2</sup> S 时钟除能 1: I <sup>2</sup> S 时钟使能 由软件置位和复位。
[24]	SCI0EN	智能卡接口 0 时钟使能位 0: SCI0 时钟除能 1: SCI0 时钟使能 由软件置位和复位。
[15]	EXTIEN	外部中断时钟使能位 0: EXTI 时钟除能 1: EXTI 时钟使能 由软件置位和复位。
[14]	AFIOEN	复用功能 I/O 时钟使能位 0: AFIO 时钟除能 1: AFIO 时钟使能 由软件置位和复位。
[11]	UR1EN	UART1 时钟使能位 0: UART1 时钟除能 1: UART1 时钟使能 由软件置位和复位。
[10]	UR0EN	UART0 时钟使能位 0: UART0 时钟除能 1: UART0 时钟使能 由软件置位和复位。

位	字段	描述
[9]	USR1EN	USART1 时钟使能位 0: USART1 时钟除能 1: USART1 时钟使能 由软件置位和复位。
[8]	USR0EN	USART0 时钟使能位 0: USART0 时钟除能 1: USART0 时钟使能 由软件置位和复位。
[5]	SPI1EN	SPI1 时钟使能位 0: SPI1 时钟除能 1: SPI1 时钟使能 由软件置位和复位
[4]	SPI0EN	SPI0 时钟使能位 0: SPI0 时钟除能 1: SPI0 时钟使能 由软件置位和复位。
[1]	I2C1EN	I <sup>2</sup> C1 时钟使能位 0: I <sup>2</sup> C1 时钟除能 1: I <sup>2</sup> C1 时钟使能 由软件置位和复位。
[0]	I2C0EN	I <sup>2</sup> C0 时钟使能位 0: I <sup>2</sup> C0 时钟除能 1: I <sup>2</sup> C0 时钟使能 由软件置位和复位。

## APB 时钟控制寄存器 1 – APBCCR1

该寄存器定义了各种 APB 外设时钟使能位。

偏移量: 0x030

复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							ADCCEN
								RW 0
类型 / 复位	23	22	21	20	19	18	17	16
	保留位	CMPEN	保留位				BFTM1EN	BFTM0EN
		RW 0					RW 0	RW 0
类型 / 复位	15	14	13	12	11	10	9	8
	保留位						GPTM1EN	GPTM0EN
							RW 0	RW 0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位	BKPREN	保留位	WDTREN	保留位		MCTM1EN	MCTM0EN
		RW 0		RW 0			RW 0	RW 0

位	字段	描述
[24]	ADCCEN	ADC 时钟使能位 0: ADC 时钟除能 1: ADC 时钟使能 由软件置位和复位。
[22]	CMPEN	CMP 时钟使能位 0: CMP 时钟除能 1: CMP 时钟使能 由软件置位和复位。
[17]	BFTM1EN	BFTM1 时钟使能位 0: BFTM1 时钟除能 1: BFTM1 时钟使能 由软件置位和复位。
[16]	BFTM0EN	BFTM0 时钟使能位 0: BFTM0 时钟除能 1: BFTM0 时钟使能 由软件置位和复位。
[9]	GPTM1EN	GPTM1 时钟使能位 0: GPTM1 时钟除能 1: GPTM1 时钟使能 由软件置位和复位。
[8]	GPTM0EN	GPTM0 时钟使能位 0: GPTM0 时钟除能 1: GPTM0 时钟使能 由软件置位和复位。
[6]	BKPREN	备份域时钟使能位用于寄存器访问 0: RTC 时钟除能 1: RTC 时钟使能 由软件置位和复位。

位	字段	描述
[4]	WDTREN	看门狗定时器时钟使能位用于寄存器访问 0: 看门狗定时器时钟除能 1: 看门狗定时器时钟使能 由软件置位和复位。
[1]	MCTM1EN	MCTM1 时钟使能位 0: MCTM1 时钟除能 1: MCTM1 时钟使能 由软件置位和复位。
[0]	MCTM0EN	MCTM0 时钟使能位 0: MCTM0 时钟除能 1: MCTM0 时钟使能 由软件置位和复位。

### 时钟源状态寄存器 – CKST

该寄存器定义了时钟源状态。

偏移量: 0x034

复位值: 0x0100\_0003

	31	30	29	28	27	26	25	24
	保留位				HSIST			
类型 / 复位					RO	0	RO	0
	23	22	21	20	19	18	17	16
	保留位				HSEST			
类型 / 复位					RO	0	RO	0
	15	14	13	12	11	10	9	8
	保留位				PLLST			
类型 / 复位					RO	0	RO	0
	7	6	5	4	3	2	1	0
	保留位				CKSWST			
类型 / 复位					RO	0	RO	1

位	字段	描述
[27:24]	HSIST	内部高速时钟 CK_HSI 占用状态位 xxx1: HSI 用于系统时钟 CK_SYS (SW = 0x03) xx1x: HSI 用于 PLL x1xx: HSI 用于时钟监控 1xxx: HIS 用于 USB PLL
[18:16]	HSEST	外部高速时钟 CK_HSE 占用状态位 xx1: HSE 用于系统时钟 CK_SYS (SW = 0x02) x1x: HSE 用于 PLL 1xx: HSE 用于 USB PLL
[11:8]	PLLST	PLL 时钟占用状态位 xxx1: PLL 用于系统时钟 CK_SYS x1xx: PLL 用于 USB 1xxx: PLL 用于 CK_REF

位	字段	描述
[2:0]	CKSWST	时钟切换状态位 00x: CK_PLL 用于系统时钟 010: CK_HSE 用于系统时钟 011: CK_HSI 用于系统时钟 110: CK_LSE 用于系统时钟 111: CK_LSI 用于系统时钟 该位段为状态位用来表示哪一个时钟源作为当前的系统时钟。

## APB 外设时钟选择寄存器 0 – APBPCSR0

该寄存器定义了 APB 外设时钟分频选择。

偏移量: 0x038  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	UR1PCLK		UR0PCLK		USR1PCLK		USR0PCLK	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	GPTM1PCLK		GPTM0PCLK		MCTM1PCLK		MCTM0PCLK	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	BFTM1PCLK		BFTM0PCLK		保留位			
类型 / 复位	RW	0	RW	0	RW	0		
	7	6	5	4	3	2	1	0
	SPI1PCLK		SPI0PCLK		I2C1PCLK		I2C0PCLK	
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:30]	UR1PCLK	UART1 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[29:28]	UR0PCLK	UART0 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[27:26]	USR1PCLK	USART1 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟



位	字段	描述
[25:24]	USR0PCLK	USART0 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[23:22]	GPTM1PCLK	GPTM1 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[21:20]	GPTM0PCLK	GPTM0 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[19:18]	MCTM1PCLK	MCTM1 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[17:16]	MCTM0PCLK	MCTM0 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[15:14]	BFTM1PCLK	BFTM1 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[13:12]	BFTM0PCLK	BFTM0 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[7:6]	SPI1PCLK	SPI1 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟

位	字段	描述
[5:4]	SPI0PCLK	SPI0 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[3:2]	I2C1PCLK	I <sup>2</sup> C1 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[1:0]	I2C0PCLK	I <sup>2</sup> C0 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟

## APB 外设时钟选择寄存器 1 – APBPCSR1

该寄存器定义了 APB 外设时钟分频选择。

偏移量: 0x03C  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位		I2SPCLK		SCI1PCLK		SCI0PCLK	
类型 / 复位			RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	BKPRCLK		WDTRCLK		保留位		CMPPCLK	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	保留位		ADCCPCLK		EXTIPCLK		AFIOPCLK	
类型 / 复位			RW	0	RW	0	RW	0

位	字段	描述
[21:20]	I2SPCLK	I <sup>2</sup> S 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟

位	字段	描述
[19:18]	SCI1PCLK	SCI1 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[17:16]	SCI0PCLK	SCI0 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[15:14]	BKPRCLK	备份域寄存器访问时钟选择 00: PCLK = CK_AHB / 4 01: PCLK = CK_AHB / 8 10: PCLK = CK_AHB / 16 11: PCLK = CK_AHB / 32 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[13:12]	WDT RPCLK	WDT 寄存器访问时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[9:8]	CMPPCLK	CMP 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[5:4]	ADCCPCLK	ADC 控制器外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[3:2]	EXTIPCLK	EXTI 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟
[1:0]	AFIOPCLK	AFIO 外设时钟选择 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = 外设时钟; CK_AHB = AHB 和 CPU 时钟

## HSI 控制寄存器 – HSICR

该寄存器用来控制 HSI RC 振荡器频率微调。

偏移量: 0x040

复位值: 0xFFFF\_0000, X 表示未定义

	31	30	29	28	27	26	25	24		
	保留位			HSICOARSE						
类型 / 复位				RO	X RO	X RO	X RO	X RO	X	
	23	22	21	20	19	18	17	16		
	HSIFINE									
类型 / 复位	RW	RW	RW	X RW	X RW	X RW	X RW	X RW	X	
	X	X								
	15	14	13	12	11	10	9	8		
	保留位									
类型 / 复位										
	7	6	5	4	3	2	1	0		
	FLOCK	REFCLKSEL			TMSEL	保留位		ATCEN	TRIMEN	
类型 / 复位	RO	0 RW	0 RW	0 RW	0			RW	0 RW	0

位	字段	描述
[28:24]	HSICOARSE	HSI 时钟粗调值 开机时, 这些位自动初始化。只能出厂时调整, 不能通过程序调整。
[23:16]	HSIFINE	HSI 时钟细调值 开机时, 这些位自动初始化。出厂时微调。但这些位也提供了额外的用户程序微调值将增加到 HSICOARSE[4:0] 位, 在 HSI 频率受电压或温度影响时, 获取更加高的精度或补偿的变量。它可由软件或带外部参考时钟的自动微调控制器 (ATC) 编程。
[7]	FLOCK	频率锁定 0: HSI 频率没有微调到目标范围 1: HSI 频率微调到目标范围
[6:5]	REFCLKSEL	参考时钟选择 00: 选择 32.768 kHz 外部低速时钟源 (LSE) 01: 选择 1 kHz USB 帧脉冲 1x: 选择外部引脚 CKIN 1 kHz 脉冲 该位为 HSI 自动微调控制器选择参考时钟源
[4]	TMSEL	微调模式选择 0: 由自动微调控制器微调 1: 由用户手动微调 该位用来选择 HSI RC 振荡器微调功能, 通过 ATC 硬件或用户使用 HSI 控制寄存器中的 HSIFINE[7:0] 位编程来微调。
[1]	ATCEN	ATC 使能 0: 除能自动微调控制器 1: 使能自动微调控制器
[0]	TRIMEN	微调使能 0: HSI 微调除能 1: HSI 微调使能 该位使能 HSI RC 振荡器微调功能, 可由 ATC 硬件或用户编程微调。

HSI 自动微调计数器寄存器 – HSIATCR

该寄存器包含了 HSI 自动微调控制器的计数值。

偏移量： 0x044  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位									
类型 / 复位										
	15	14	13	12	11	10	9	8		
	保留位		ATCNT							
类型 / 复位			RO	0	RO	0	RO	0	RO	0
	7	6	5	4	3	2	1	0		
	ATCNT									
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO	0

位	字段	描述
[13:0]	ATCNT	自动微调计数器 这些位包含了 HSI 自动微调控制器的计数器的值。

低功耗控制寄存器 – LPCR

该寄存器定义了低功耗控制位。

偏移量: 0x300  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	保留位							USBSLEEP
	7	6	5	4	3	2	1	0
类型 / 复位	保留位							

位	字段	描述
[8]	USBSLEEP	USB 休眠软件控制使能位 0: 除能 1: 使能 USB 软件休眠 由软件置位和复位。详细信息请参考电源控制单元章节。

## 单片机调试控制寄存器 – MCUDBGCR

该寄存器定义了单片机的调试控制。

偏移量: 0x304

复位值: 0x0000\_0000 (仅由 V<sub>DD</sub> 域上电复位来复位)

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位		DBSCI1	DBTRACE	DBUR1	DBUR0	DBBFTM1	DBBFTM0		
类型 / 复位			RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8		
	DBSCI0	DBDSLP2	DBI2C1	DBI2C0	DBSPI1	DBSPI0	DBUSR1	DBUSR0		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0		
	DBGPTM1	DBGPTM0	DBMCTM1	DBMCTM0	DBWDT	DBPD	DBDSLP1	DBSLP		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[21]	DBSCI1	SCI1 调试模式使能位 0: 与正常模式相同 1: 当内核暂停, SCI1 溢出停止 由软件置位和复位。
[20]	DBTRACE	TRACESWO 调试模式使能位 0: 除能 TRACESWO 输出 1: 使能 TRACESWO 输出 由软件置位和复位。
[19]	DBUR1	UART1 调试模式使能位 0: 与正常模式相同 1: 当内核暂停, UART1 FIFO 溢出停止 由软件置位和复位。
[18]	DBUR0	UART0 调试模式使能位 0: 与正常模式相同 1: 当内核暂停, UART0 FIFO 溢出停止 由软件置位和复位。
[17]	DBBFTM1	BFTM1 调试模式使能位 0: 即使内核暂停, BFTM1 计数器继续计数 1: 当内核暂停, BFTM1 计时器停止计数 由软件置位和复位。
[16]	DBBFTM0	BFTM0 调试模式使能位 0: 即使内核暂停, BFTM0 计数器继续计数 1: 当内核暂停, BFTM0 计时器停止计数 由软件置位和复位。
[15]	DBSCI0	SCI0 调试模式使能位 0: 与正常模式相同 1: 当内核暂停, SCI0 溢出停止 由软件置位和复位。

位	字段	描述
[14]	DBDSLP2	调试深度休眠模式 2 0: 在深度休眠模式 2 下, LDO = Off, DMOS = On, FCLK = Off, CM3EN = 0 1: 在深度休眠模式 2 下, LDO = On, FCLK = On, CM3EN = 1 由软件置位和复位。
[13]	DBI2C1	I <sup>2</sup> C1 调试模式使能位 0: 超时计数与正常模式相同 1: 当内核暂停时, I <sup>2</sup> C1 超时计数停止 由软件置位和复位。
[12]	DBI2C0	I <sup>2</sup> C0 调试模式使能位 0: 超时计数与正常模式相同 1: 当内核暂停时, I <sup>2</sup> C0 超时计数停止 由软件置位和复位。
[11]	DBSPI1	SPI1 调试模式使能位 0: 与正常模式相同 1: 当内核暂停时, SPI1 FIFO 溢出停止 由软件置位和复位。
[10]	DBSPI0	SPI0 调试模式使能位 0: 与正常模式相同 1: 当内核暂停时, SPI0 FIFO 溢出停止 由软件置位和复位。
[9]	DBUSR1	USART1 调试模式使能位 0: 与正常模式相同 1: 当内核暂停时, USART1 FIFO 溢出停止 由软件置位和复位。
[8]	DBUSR0	USART0 调试模式使能位 0: 与正常模式相同 1: 当内核暂停时, USART0 FIFO 溢出停止 由软件置位和复位。
[7]	DBGPTM1	GPTM1 调试模式使能位 0: 即使内核暂停, GPTM1 计数器继续计数 1: 当内核暂停, GPTM1 计数器停止计数 由软件置位和复位。
[6]	DBGPTM0	GPTM0 调试模式使能位 0: 即使内核暂停, GPTM0 计数器继续计数 1: 当内核暂停, GPTM0 计数器停止计数 由软件置位和复位。
[5]	DBMCTM1	MCTM1 调试模式使能位 0: 即使内核暂停, MCTM1 计数器继续计数 1: 当内核暂停, MCTM1 计数器停止计数 由软件置位和复位。
[4]	DBMCTM0	MCTM0 调试模式使能位 0: 即使内核暂停, MCTM0 计数器继续计数 1: 当内核暂停, MCTM0 计数器停止计数 由软件置位和复位。
[3]	DBWDT	看门狗定时器调试模式使能位 0: 即使内核暂停, 看门狗定时器继续计数 1: 当内核暂停, 看门狗定时器停止计数 由软件置位和复位。
[2]	DBPD	调试暂停模式 0: 在暂停模式下, LDO = Off, FCLK = Off, CM3EN = 0 1: 在暂停模式下, LDO = On, FCLK = On, CM3EN = 1 由软件置位和复位。



位	字段	描述
[1]	DBDSLP1	调试深度休眠模式 1 0: 在深度休眠模式 1 下, LDO = 低功耗模式, FCLK = Off, CM3EN = 0 1: 在深度休眠模式 1 下, LDO = On, FCLK = On, CM3EN = 1 由软件置位和复位。
[0]	DBSLP	调试休眠模式 0: 在休眠模式下, LDO = On, FCLK = On, CM3EN = 0 1: 在休眠模式下, LDO = On, FCLK = On, CM3EN = 1 由软件置位和复位。

## 7 复位控制单元 (RSTCU)

### 简介

复位控制单元 RSTCU 有三种复位方式，上电复位、系统复位和 APB 单元复位。上电复位，即冷复位，在上电时复位了整个系统。系统复位复位了处理器内核和除调试端口控制器以外的外设 IP 元件。这些复位可以通过外部信号、内部事件和复位发生器触发。有关这些复位的更多信息将在下一章节描述。

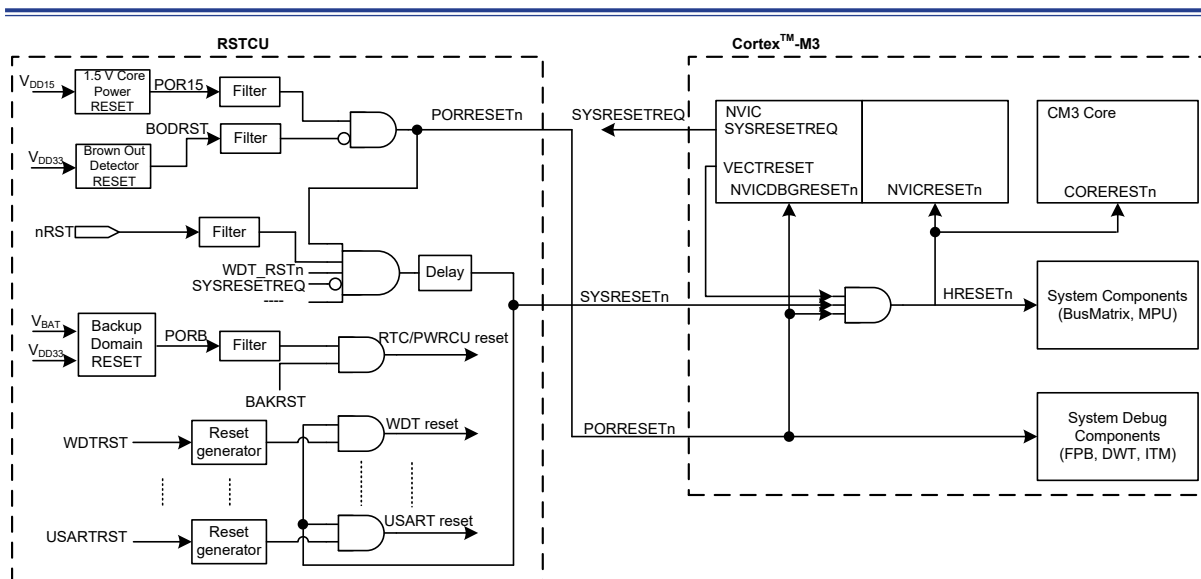


图 19. RSTCU 方框图

## 功能描述

### 上电复位

上电复位 POR 是由外部复位或内部复位发生器产生。这两种类型都有一个内部滤波器，以防止因噪声而导致错误的复位操作。参照图 20，POR15 信号会一直维持在低电平直到内部 LDO 稳压器准备好提供 1.5 V 电源时。除了 POR15 信号，当 LVDCSR 寄存器中的 BODEN 位被置位，欠压事件发生时，电源控制单元 PWRCU 将使 BODF 信号有效作为暂停复位 PDR。有关 PWRCU 功能的更多细节，请参考 PWRCU 章节。

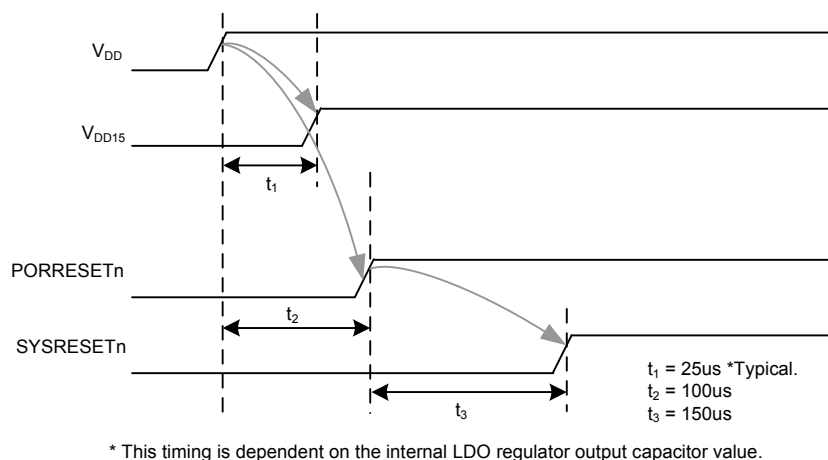


图 20. 上电复位时序图

### 系统复位

系统复位是由上电复位 PORRESETn、看门狗定时器复位 WDT\_RSTn、nRST 引脚或软件复位 SYSRESETREQ 事件产生。有关 SYSRESETREQ 和 VECTRESET 事件的更多信息，请参考 Cortex®-M3 参考手册的相关章节。

### AHB 和 APB 单元复位

AHB 和 APB 单元复位可分为硬件复位和软件复位。硬件复位可以由上电复位或系统复位对所有 AHB 和 APB 单元进行复位。连接到 AHB 和 APB 总线上的每个功能 IP 都可以通过 RSTCU 中相关的软件复位位单独复位。例如，应用软件可以通过 APBPRSTR0 寄存器中的 USR0RST 位产生一个 USART0 复位。

寄存器列表

下表显示了 RSTCU 寄存器及其复位值。

表 21. RSTCU 寄存器列表

寄存器	偏移量	描述	复位值
GRSR	0x100	全局复位状态寄存器	0x0000_0008
AHBPRSTR	0x104	AHB 外设复位寄存器	0x0000_0000
APBPRSTR0	0x108	APB 外设复位寄存器 0	0x0000_0000
APBPRSTR1	0x10C	APB 外设复位寄存器 1	0x0000_0000

寄存器描述

全局复位状态寄存器 – GRSR

该寄存器定义了复位状态的各种状况。

偏移量:	0x100
复位值:	0x0000_0008
类型 / 复位	
	31 30 29 28 27 26 25 24
	保留位
类型 / 复位	
	23 22 21 20 19 18 17 16
	保留位
类型 / 复位	
	15 14 13 12 11 10 9 8
	保留位
类型 / 复位	
	7 6 5 4 3 2 1 0
	保留位 PORSTF WDTRSTF EXTRSTF SYSRSTF
类型 / 复位	WC 1 WC 0 WC 0 WC 0

位	字段	描述
[3]	PORSTF	内核 1.5 V 上电复位标志位 0: 上电复位未发生 1: 上电复位发生 当上电复位发生时, 由硬件置位。通过写 1 来复位。
[2]	WDTRSTF	看门狗定时器复位标志位 0: 看门狗定时器复位未发生 1: 看门狗定时器复位发生 当看门狗定时器复位发生时, 由硬件置位。通过写 1 来复位, 或当上电复位发生时, 由硬件复位。
[1]	EXTRSTF	外部引脚复位标志位 0: 引脚复位未发生 1: 引脚复位发生 当外部引脚复位发生时, 由硬件置位。通过写 1 来复位, 或当上电复位发生时, 由硬件复位。

位	字段	描述
[0]	SYSRSTF	系统复位标志位 0: 通过 NVIC 建立的系统复位未发生 1: 通过 NVIC 建立的系统复位发生 当系统复位发生时, 由硬件置位。通过写 1 来复位, 或当上电复位发生时, 由硬件复位。

## AHB 外设复位寄存器 – AHBPRSTR

该寄存器定义了 AHB 外设的软件复位控制位。

偏移量: 0x104

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	AESRST	SDIORST	保留位	PERST	PDRST	PCRST	PBRST	PARST
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	CRCRST	EBIRST	USBRST	CSIFRST	保留位			DMARST
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15]	AESRST	AES 复位控制位 0: 无复位 1: AES 复位 该位由软件置位, 由硬件自动清零。
[14]	SDIORST	SDIO 复位控制位 0: 无复位 1: SDIO 复位 该位由软件置位, 由硬件自动清零。
[12]	PERST	通用 I/O 口 PE 复位控制位 0: 无复位 1: PE 复位 该位由软件置位, 由硬件自动清零。
[11]	PDRST	通用 I/O 口 PD 复位控制位 0: 无复位 1: PD 复位 该位由软件置位, 由硬件自动清零。
[10]	PCRST	通用 I/O 口 PC 复位控制位 0: 无复位 1: PC 复位 该位由软件置位, 由硬件自动清零。

位	字段	描述
[9]	PBRST	通用 I/O 口 PB 复位控制位 0: 无复位 1: PB 复位 该位由软件置位，由硬件自动清零。
[8]	PARST	通用 I/O 口 PA 复位控制位 0: 无复位 1: PA 复位 该位由软件置位，由硬件自动清零。
[7]	CRCRST	CRC 复位控制位 0: 无复位 1: CRC 复位 该位由软件置位，由硬件自动清零。
[6]	EBIRST	EBI 复位控制位 0: 无复位 1: EBI 复位 该位由软件置位，由硬件自动清零。
[5]	USBRST	USB 复位控制位 0: 无复位 1: USB 复位 该位由软件置位，由硬件自动清零。
[4]	CSIFRST	CSIF 复位控制位 0: 无复位 1: CSIF 复位 该位由软件置位，由硬件自动清零。
[0]	DMARST	外设 DMA (PDMA) 复位控制位 0: 无复位 1: 复位外设 DMA (PDMA) 该位由软件置位，由硬件自动清零。

## APB 外设复位寄存器 0 – APBPRSTR0

该寄存器定义了 APB 外设的软件复位控制位。

偏移量： 0x108

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
				保留位	SCI1RST	保留位	I2SRST	SCI0RST
类型 / 复位					RW 0		RW 0	RW 0
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	EXTIRST	AFIORST	保留位		UR1RST	UR0RST	USR1RST	USR0RST
类型 / 复位	RW 0	RW 0			RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	保留位		SPI1RST	SPI0RST	保留位		I2C1RST	I2C0RST
类型 / 复位			RW 0	RW 0			RW 0	RW 0

位	字段	描述
[27]	SCI1RST	SCI1 复位控制位 0: 无复位 1: 复位 SCI1 该位由软件置位，由硬件自动清零。
[25]	I2SRST	I <sup>2</sup> S 复位控制位 0: 无复位 1: 复位 I <sup>2</sup> S 该位由软件置位，由硬件自动清零。
[24]	SCI0RST	SCI0 复位控制位 0: 无复位 1: 复位 SCI0 该位由软件置位，由硬件自动清零。
[15]	EXTIRST	外部中断控制器复位控制位 0: 无复位 1: 复位 EXTI 该位由软件置位，由硬件自动清零。
[14]	AFIORST	复用功能 I/O 复位控制位 0: 无复位 1: 复位复用功能 I/O 该位由软件置位，由硬件自动清零。
[11]	UR1RST	UART1 复位控制位 0: 无复位 1: 复位 UART1 该位由软件置位，由硬件自动清零。
[10]	UR0RST	UART0 复位控制位 0: 无复位 1: 复位 UART0 该位由软件置位，由硬件自动清零。

位	字段	描述
[9]	USR1RST	USART1 复位控制位 0: 无复位 1: 复位 USART1 该位由软件置位, 由硬件自动清零。
[8]	USR0RST	USART0 复位控制位 0: 无复位 1: 复位 USART0 该位由软件置位, 由硬件自动清零。
[5]	SPI1RST	SPI1 复位控制位 0: 无复位 1: 复位 SPI1 该位由软件置位, 由硬件自动清零。
[4]	SPI0RST	SPI0 复位控制位 0: 无复位 1: 复位 SPI0 该位由软件置位, 由硬件自动清零。
[1]	I2C1RST	I <sup>2</sup> C1 复位控制位 0: 无复位 1: 复位 I <sup>2</sup> C1 该位由软件置位, 由硬件自动清零。
[0]	I2C0RST	I <sup>2</sup> C0 复位控制位 0: 无复位 1: 复位 I <sup>2</sup> C0 该位由软件置位, 由硬件自动清零。



## APB 外设复位寄存器 1 – APBPRSTR1

该寄存器定义了几种 APB 外设软件复位控制位。

偏移量: 0x10C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							ADCRST
类型 / 复位								RW 0
	23	22	21	20	19	18	17	16
	保留位	CMPRST	保留位				BFTM1RST	BFTM0RST
类型 / 复位		RW 0					RW 0	RW 0
	15	14	13	12	11	10	9	8
	保留位						GPTM1RST	GPTM0RST
类型 / 复位							RW 0	RW 0
	7	6	5	4	3	2	1	0
	保留位			WDTRST	保留位		MCTM1RST	MCTM0RST
类型 / 复位				RW 0			RW 0	RW 0

位	字段	描述
[24]	ADCRST	A/D 转换器复位控制位 0: 无复位 1: 复位 A/D 转换器 该位由软件置位, 由硬件自动清零。
[22]	CMPRST	比较器控制器复位控制位 0: 无复位 1: 复位 CMP 该位由软件置位, 由硬件自动清零。
[17]	BFTM1RST	BFTM1 复位控制位 0: 无复位 1: 复位 BFTM1 该位由软件置位, 由硬件自动清零。
[16]	BFTM0RST	BFTM0 复位控制位 0: 无复位 1: 复位 BFTM0 该位由软件置位, 由硬件自动清零。
[9]	GPTM1RST	GPTM1 复位控制位 0: 无复位 1: 复位 GPTM1 该位由软件置位, 由硬件自动清零。
[8]	GPTM0RST	GPTM0 复位控制位 0: 无复位 1: 复位 GPTM0 该位由软件置位, 由硬件自动清零。
[4]	WDTRST	看门狗定时器复位控制位 0: 无复位 1: 复位看门狗定时器 该位由软件置位, 由硬件自动清零。

位	字段	描述
[1]	MCTM1RST	MCTM1 复位控制位 0: 无复位 1: 复位 MCTM1 该位由软件置位，由硬件自动清零。
[0]	MCTM0RST	MCTM0 复位控制位 0: 无复位 1: 复位 MCTM0 该位由软件置位，由硬件自动清零。

## 8 通用 I/O (GPIO)

### 简介

单片机有多达 80 个通用 I/O 引脚 (GPIO), 即 PA0 ~ PA15、PB0 ~ PB15、PC0 ~ PC15、PD0 ~ PD15 和 PE0 ~ PE15, 可以实现逻辑输入 / 输出功能。每个 GPIO 端口都有相关的控制和配置寄存器, 以满足特定应用的需求。实际可用的通用 I/O 端口数量取决于单片机特性和封装类型。请参考单片机的规格书获得更多信息。

在封装上 GPIO 引脚与其它复用功能引脚 (AFs) 共用, 以获得较大的灵活性。通过配置相应的寄存器, GPIO 口可以被用作复用功能 AF 的输入或输出脚。

单片机 GPIO 引脚的外部中断在外部中断控制单元 (EXTI) 都有相关的控制和配置寄存器。

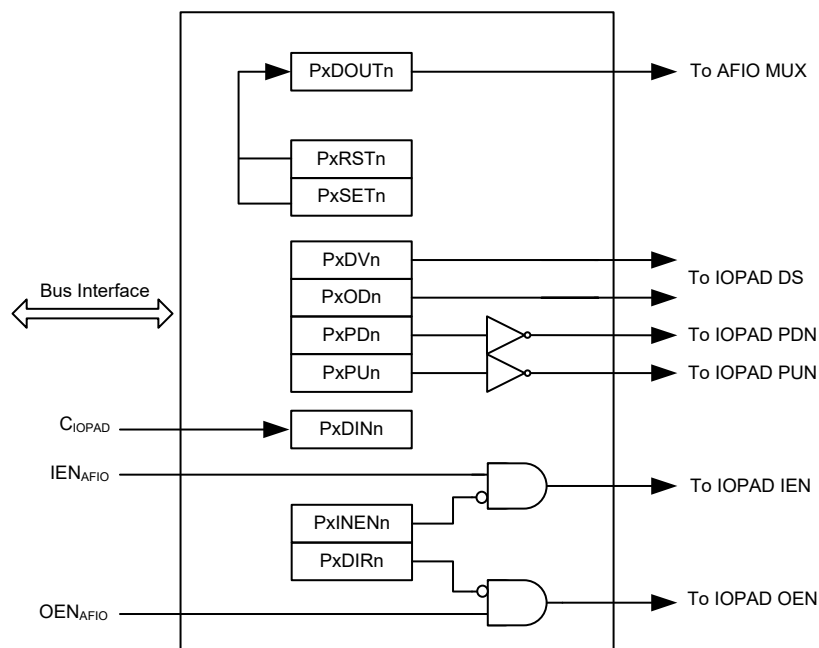


图 21. GPIO 方框图

## 特性

- 输入 / 输出方向控制
- 输入引脚弱上拉 / 下拉控制
- 输出推挽 / 漏极开路使能控制
- 输出置位 / 复位控制
- 输出驱动电流选择
- 可编程边沿触发的外部中断 – 通过 EXTI 配置寄存器
- 模拟输入 / 输出配置 – 通过 AFIO 配置寄存器
- 复用功能的输入 / 输出配置 – 通过 AFIO 配置寄存器
- 端口配置锁定

## 功能描述

### 默认的 GPIO 引脚配置

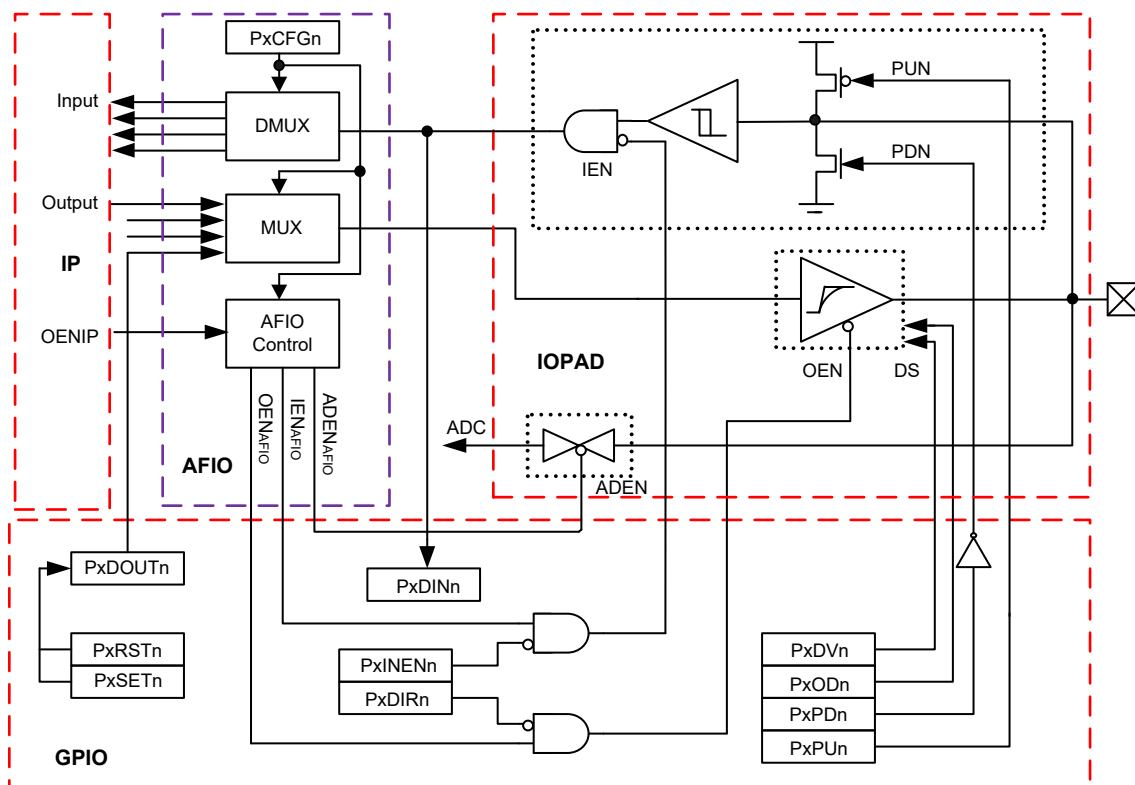
在复位期间或之后，复用功能都无效，GPIO 端口被配置为输入除能浮空模式，即不带有上拉 / 下拉电阻的输入除能。单片机复位后，只有启动和串行线调试引脚与 I/O 引脚共用是有效的。

- BOOT0：带有内部上拉的输入使能
- BOOT1：带有内部上拉的输入使能
- JTCK/SWCLK：带有内部上拉的输入使能
- JTMS/SWDIO：带有内部上拉的输入使能
- JTDI：带有内部上拉的输入使能
- JTRST：带有内部上拉的输入使能

### 通用 I/O – GPIO

GPIO 引脚可以通过数据方向控制寄存器 PxDIRCR ( 其中 x = A ~ E ) 配置为输入或输出。当 GPIO 引脚配置为输入引脚时，如果输入使能功能寄存器 PxINER 的使能位被置位，则外部引脚上的数据可读。GPIO 上拉 / 下拉寄存器 PxPUR / PxPDR 可配置以满足特定应用。当上拉和下拉功能都使能，上拉功能具有较高的优先级，上拉功能被除能时才能使能下拉功能。

GPIO 引脚可以配置为输出引脚，输出数据锁存到数据寄存器 PxDOCTR。通过漏极开路选择寄存器 PxODR，输出类型可设置为推挽式或漏极开路式。通过配置端口输出置位和复位控制寄存器 PxSRR 或端口输出复位控制寄存器 PxRR，在不影响未被选中的位的情况下，对选中的一个或几个特定位进行输出数据的置位 / 复位。端口输出置位和复位功能都使能时，端口输出置位功能具有更高的优先级，端口输出复位功能将被封锁。通过配置驱动电流选择寄存器 PxDRVR，选择 GPIO 引脚输出的驱动电流。



---

PxCFGn (x = A ~ E): AFIO 配置

类型	AFIO			GPIO		PAD		
	ADEN <sub>AFIO</sub>	OEN <sub>AFIO</sub>	IEN <sub>AFIO</sub>	PxDIRn	PxINENn	ADEN	OEN	IEN
GPIO 输入 <sup>(注)</sup>	1	1	1	0	1	1	1	0
GPIO 输出 <sup>(注)</sup>	1	1	1	1	0 (1 如需要)	1	0	1 (0)
AFIO 输入	1	1	0	0	X	1	1	0
AFIO 输出	1	0	1	X	0 (1 如需要)	1	0	1 (0)
ADC 输入	0	1	1	0	0 (1 如需要)	0	1	1 (0)
OSC 输出	0	1	1	0	0 (1 如需要)	0	1	1 (0)

September 14, 2020

## GPIO 锁定机制

GPIO 还提供了锁定功能用来锁定端口，直到复位事件发生。PxLOCKR (x = A ~ E) 寄存器是用来锁定端口 x 和锁定控制选项。在 PxLOCKR 寄存器 PxLKEY 字段写入值 0x5FA0 冻结 PxDIRCR、PxINER、PxPUR、PxPDR、PxODR、PxDRVR 控制和 AFIO 的模式配置 (GPxCFGHR 或 GPxCFGLR，其中 x = A ~ E)。如果 PxLOCKR 写入值为 0x5FA0\_0001，这意味着端口 x 锁定功能使能，端口 x 引脚 0 被冻结。

## 寄存器列表

下表显示 GPIO 寄存器和端口 A ~ E 的复位值。

表 23. GPIO 寄存器列表

寄存器	偏移量	描述	复位值
<b>GPIO A 基址 = 0x400B_0000</b>			
PADIRCR	0x000	端口 A 数据方向控制寄存器	0x0000_0000
PAINER	0x004	端口 A 输入功能使能控制寄存器	0x0000_0300
PAPUR	0x008	端口 A 上拉选择寄存器	0x0000_F300
PAPDR	0x00C	端口 A 下拉选择寄存器	0x0000_0000
PAODR	0x010	端口 A 漏极开路选择寄存器	0x0000_0000
PADRVr	0x014	端口 A 驱动电流选择寄存器	0x0000_0000
PALOCKR	0x018	端口 A 锁定寄存器	0x0000_0000
PADINR	0x01C	端口 A 数据输入寄存器	0x0000_F300
PADOUTR	0x020	端口 A 数据输出寄存器	0x0000_0000
PASRR	0x024	端口 A 输出置位和复位控制寄存器	0x0000_0000
PARR	0x028	端口 A 输出复位控制寄存器	0x0000_0000
<b>GPIO B 基址 = 0x400B_2000</b>			
PBDIRCR	0x000	端口 B 数据方向控制寄存器	0x0000_0000
PBINER	0x004	端口 B 输入功能使能控制寄存器	0x0000_0000
PBPUR	0x008	端口 B 上拉选择寄存器	0x0000_0000
PBPDR	0x00C	端口 B 下拉选择寄存器	0x0000_0000
PBODR	0x010	端口 B 漏极开路选择寄存器	0x0000_0000
PBDRVr	0x014	端口 B 驱动电流选择寄存器	0x0000_0000
PBLOCKR	0x018	端口 B 锁定寄存器	0x0000_0000
PBDINR	0x01C	端口 B 数据输入寄存器	0x0000_0000
PBDOUTR	0x020	端口 B 数据输出寄存器	0x0000_0000
PBSRR	0x024	端口 B 输出置位和复位控制寄存器	0x0000_0000
PBRR	0x028	端口 B 输出复位控制寄存器	0x0000_0000
<b>GPIO C 基址 = 0x400B_4000</b>			
PCDIRCR	0x000	端口 C 数据方向控制寄存器	0x0000_0000
PCINER	0x004	端口 C 输入功能使能控制寄存器	0x0000_0000
PCPUR	0x008	端口 C 上拉选择寄存器	0x0000_0000
PCPDR	0x00C	端口 C 下拉选择寄存器	0x0000_0000
PCODR	0x010	端口 C 漏极开路选择寄存器	0x0000_0000
PCDRVr	0x014	端口 C 驱动电流选择寄存器	0x0000_0000
PCLOCKR	0x018	端口 C 锁定寄存器	0x0000_0000
PCDINR	0x01C	端口 C 数据输入寄存器	0x0000_0000
PCDOUTR	0x020	端口 C 数据输出寄存器	0x0000_0000

寄存器	偏移量	描述	复位值
PCSRR	0x024	端口 C 输出置位和复位控制寄存器	0x0000_0000
PCRR	0x028	端口 C 输出复位控制寄存器	0x0000_0000
<b>GPIO D 基址 = 0x400B_6000</b>			
PDDIRCR	0x000	端口 D 数据方向控制寄存器	0x0000_0000
PDINER	0x004	端口 D 输入功能使能控制寄存器	0x0000_0000
PDPUR	0x008	端口 D 上拉选择寄存器	0x0000_0000
PDPDR	0x00C	端口 D 下拉选择寄存器	0x0000_0000
PDODR	0x010	端口 D 漏极开路选择寄存器	0x0000_0000
PDDRVCR	0x014	端口 D 驱动电流选择寄存器	0x0000_0000
PDLOCKR	0x018	端口 D 锁定寄存器	0x0000_0000
PDDINR	0x01C	端口 D 数据输入寄存器	0x0000_0000
PDDOUTR	0x020	端口 D 数据输出寄存器	0x0000_0000
PDSRR	0x024	端口 D 输出置位和复位控制寄存器	0x0000_0000
PDRR	0x028	端口 D 输出复位控制寄存器	0x0000_0000
<b>GPIO E 基址 = 0x400B_8000</b>			
PEDIRCR	0x000	端口 E 数据方向控制寄存器	0x0000_0000
PEINER	0x004	端口 E 输入功能使能控制寄存器	0x0000_0000
PEPUR	0x008	端口 E 上拉选择寄存器	0x0000_0000
PEPDR	0x00C	端口 E 下拉选择寄存器	0x0000_0000
PEODR	0x010	端口 E 漏极开路选择寄存器	0x0000_0000
PEDRVCR	0x014	端口 E 驱动电流选择寄存器	0x0000_0000
PELOCKR	0x018	端口 E 锁定寄存器	0x0000_0000
PEDINR	0x01C	端口 E 数据输入寄存器	0x0000_0000
PEDOUTR	0x020	端口 E 数据输出寄存器	0x0000_0000
PESRR	0x024	端口 E 输出置位和复位控制寄存器	0x0000_0000
PERR	0x028	端口 E 输出复位控制寄存器	0x0000_0000

寄存器描述

端口 A 数据方向控制寄存器 – PADIRCR

该寄存器用来控制 GPIO 端口 A 引脚方向为输入或输出。

偏移量: 0x000  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PADIR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PADIR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PADIRn	GPIO 端口 A 引脚 n 的方向控制位 (n = 0 ~ 15) 0: 引脚 n 处于输入模式 1: 引脚 n 处于输出模式



端口 A 输入功能使能控制寄存器 – PAINER

该寄存器用来使能或除能 GPIO 端口 A 的输入功能。

偏移量：0x004

复位值：0x0000\_0300

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	PAINEN							
	RW	0	RW	0	RW	1	RW	1
	7	6	5	4	3	2	1	0
类型 / 复位	PAINEN							
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PAINEN <sub>n</sub>	GPIO 端口 A 引脚 n 的输入使能控制位 (n = 0 ~ 15) 0: 除能引脚 n 的输入功能 1: 使能引脚 n 的输入功能 当除能引脚 n 的输入功能时，输入施密特触发器将关闭，施密特触发器输出将保持在零状态。

端口 A 上拉选择寄存器 – PAPUR

该寄存器用来使能或除能 GPIO 端口 A 的上拉功能。

偏移量： 0x008  
复位值： 0x0000\_F300

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	PAPU							
类型 / 复位	RW	1	RW	1	RW	1	RW	1
	7	6	5	4	3	2	1	0
	PAPU							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PAPUn	GPIO 端口 A 引脚 n 的上拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的上拉功能 1: 使能引脚 n 的上拉功能 注: 当上拉和下拉功能都使能, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 A 下拉选择寄存器 – PAPDR

该寄存器用来使能或除能 GPIO 端口 A 的下拉功能。

偏移量： 0x00C

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PAPD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
类型 / 复位	PAPD								
	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PAPDn	GPIO 端口 A 引脚 n 的下拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的下拉功能 1: 使能引脚 n 的下拉功能 注: 当上拉和下拉功能都使能, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 A 漏极开路选择寄存器 – PAODR

该寄存器用来使能或除能 GPIO 端口 A 的漏极开路功能。

偏移量： 0x010  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PAOD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PAOD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PAODn	GPIO 端口 A 引脚 n 的漏极开路选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的漏极开路输出 (输出类型为 CMOS 输出) 1: 使能引脚 n 的漏极开路输出 (输出类型为漏极开路) 注: 当漏极开路功能使能时, 引脚 n 内部上拉或下拉设置将无效。

端口 A 输出电流驱动选择寄存器 – PADRVR

该寄存器用来定义 GPIO 端口 A 的输出驱动电流。

偏移量： 0x014  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PADV15		PADV14		PADV13		PADV12		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PADV11		PADV10		PADV9		PADV8		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PADV7		PADV6		PADV5		PADV4		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PADV3		PADV2		PADV1		PADV0		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	PADVn[1:0]	GPIO 端口 A 引脚 n 的输出电流驱动选择控制位 (n = 0 ~ 15) 00: 4 mA 源 / 灌电流 01: 8 mA 源 / 灌电流 10: 12 mA 源 / 灌电流 11: 16 mA 源 / 灌电流

端口 A 锁定寄存器 – PALOCKR

该寄存器定义了 GPIO 端口 A 的锁定配置。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PALKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PALKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PALOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PALOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:16]	PALKEY	GPIO 端口 A 锁定键 0x5FA0: 使能端口 A 锁定功能 其它: 除能端口 A 锁定功能 为了锁定端口 A 功能, 在该寄存器 PALKEY 字段写入值 0x5FA0。为了在这个锁定寄存器执行一个成功的写操作, 写入 PALKEY 字段的值必须为 0x5FA0。如果写入这个字段值不等于 0x5FA0, PALOCKR 寄存器上的任何写操作将被中止。对 PALKEY 字段的一个读操作的结果将返回 GPIO 端口 A 锁定状态, 表示 GPIO 端口 A 是否被锁定。如果读值, PALKEY 字段为 0, 表明 GPIO 端口 A 锁定功能除能。否则, 它表明 GPIO 端口 A 锁定功能使能, 读值等于 1。
[15:0]	PALOCKn	GPIO 端口 A 引脚 n 的锁定控制位 (n = 0 ~ 15) 0: 端口 A 引脚 n 未锁定 1: 端口 A 引脚 n 锁定 当正确的锁定键应用于 PALKEY 字段, PALOCKn 位用来锁定相应 GPIO 引脚的配置。被锁定的配置包括 PADIRn、PAINENn、PAPUn、PAPDn、PAODn 和 PADVn 设置相关的 GPIO 寄存器。另外, 用来配置为相关 GPIO 引脚的复用功能的 AFIO 寄存器 GPACFGHR 或 GPACFGLR 字段也将被锁定。请注意, PALOCKR 寄存器只能写入一次, 这意味着 PALKEY 和 PALOCKn (锁定控制位) 应一起写入, 不能改变直到系统复位或 GPIO 端口 A 发生复位。

端口 A 数据输入寄存器 – PADINR

此寄存器定义了 GPIO 端口 A 的输入数据。

偏移量: 0x01C  
复位值: 0x0000\_F300

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	PADIN							
类型 / 复位	RO	1	RO	1	RO	1	RO	1
	7	6	5	4	3	2	1	0
	PADIN							
类型 / 复位	RO	0	RO	0	RO	0	RO	0

位	字段	描述
[15:0]	PADIN <sub>n</sub>	GPIO 端口 A 引脚 n 的数据输入位 (n = 0 ~ 15) 0: 引脚 n 的输入数据为 0 1: 引脚 n 的输入数据为 1

端口 A 输出数据寄存器 – PADOUTR

该寄存器定义了 GPIO 端口 A 的输出数据。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PADOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PADOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PADOUTn	GPIO 端口 A 引脚 n 的数据输出位 (n = 0 ~ 15) 0: 引脚 n 的输出数据为 0 1: 引脚 n 的输出数据为 1



端口 A 输出置位 / 复位控制寄存器 – PASRR

该寄存器用来置位或复位 GPIO 端口 A 输出数据相应的位。

偏移量: 0x024  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PARST								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	PARST								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	PASET								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	PASET								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO

位	字段	描述
[31:16]	PARSTn	GPIO 端口 A 引脚 n 的输出复位控制位 (n = 0 ~ 15) 0: 不影响 PADOUTn 位 1: 复位 PADOUTn 位
[15:0]	PASETn	GPIO 端口 A 引脚 n 的输出置位控制位 (n = 0 ~ 15) 0: 不影响 PADOUTn 位 1: 置位 PADOUTn 位 注意, 如果 PASETn 和 PARSTn 位同时被置位, PASETn 位使能的功能有比较高的优先级。

端口 A 输出复位寄存器 – PARR

该寄存器用来复位 GPIO 端口 A 输出数据相应的位。

偏移量： 0x028  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PARST								
	7	6	5	4	3	2	1	0	
类型 / 复位	PARST								
	WO	0	WO	0	WO	0	WO	0	WO
	0	WO	0	WO	0	WO	0	WO	0

位	字段	描述
[15:0]	PARSTn	GPIO 端口 A 引脚 n 的输出复位位 (n = 0 ~ 15) 0: 不影响 PADOUTn 位 1: 复位 PADOUTn 位

端口 B 数据方向控制寄存器 – PBDIRCR

该寄存器用来控制 GPIO 端口 B 引脚的方向作为输入或输出。

偏移量： 0x000  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PBDIR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PBDIR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PBDIRn	GPIO 端口 B 引脚 n 的方向控制位 (n = 0 ~ 15) 0: 引脚 n 处于输入模式 1: 引脚 n 处于输出模式

端口 B 输入功能使能控制寄存器 – PBINER

此寄存器用来使能或除能 GPIO 端口 B 的输入功能。

偏移量： 0x004  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	PBINEN							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	PBINEN							
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PBINEN <sub>n</sub>	GPIO 端口 B 引脚 n 的输入使能控制位 (n = 0 ~ 15) 0: 除能引脚 n 的输入功能 1: 使能引脚 n 的输入功能 当除能引脚 n 的输入功能时，输入施密特触发器将关闭，施密特触发器输出将保持在零状态。

端口 B 上拉选择寄存器 – PBPUR

该寄存器用来使能或除能 GPIO 端口 B 的上拉功能。

偏移量： 0x008  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	PBPUn							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	PBPUn							
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PBPUn	GPIO 端口 B 引脚 n 的上拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的上拉功能 1: 使能引脚 n 的上拉功能 注: 当上拉和下拉功能都使能, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 B 下拉选择寄存器 – PBPDR

该寄存器用来使能或除能 GPIO 端口 B 的下拉功能。

偏移量： 0x00C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PBPD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PBPD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PBPDn	GPIO 端口 B 引脚 n 的下拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的下拉功能 1: 使能引脚 n 的下拉功能 注: 当上拉和下拉功能都使能时, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 B 漏极开路选择寄存器 – PBODR

该寄存器用来使能或除能 GPIO 端口 B 的漏极开路功能。

偏移量： 0x010  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PBOD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PBOD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PBODn	GPIO 端口 B 引脚 n 的漏极开路选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的漏极开路输出 (输出类型为 CMOS 输出) 1: 使能引脚 n 的漏极开路输出 (输出类型为漏极开路) 注: 当漏极开路功能使能时, 引脚 n 内部上拉或下拉设置将无效。

端口 B 输出电流驱动选择寄存器 – PBDRVR

该寄存器用来定义 GPIO 端口 B 的输出驱动电流。

偏移量： 0x014  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PBDV15		PBDV14		PBDV13		PBDV12		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PBDV11		PBDV10		PBDV9		PBDV8		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PBDV7		PBDV6		PBDV5		PBDV4		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PBDV3		PBDV2		PBDV1		PBDV0		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	PBDVn[1:0]	GPIO 端口 B 引脚 n 的输出电流驱动选择控制位 (n = 0 ~ 15) 00: 4 mA 源 / 灌电流 01: 8 mA 源 / 灌电流 10: 12 mA 源 / 灌电流 11: 16 mA 源 / 灌电流



端口 B 锁定寄存器 – PBLOCKR

该寄存器定义了 GPIO 端口 B 的锁定配置。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PBLKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PBLKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PBLOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PBLOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:16]	PBLKEY	GPIO 端口 B 锁定键 0x5FA0: 使能端口 B 锁定功能 其它: 除能端口 B 锁定功能 为了锁定端口 B 功能, 在该寄存器 PBLKEY 字段写入值 0x5FA0。为了在这个锁定寄存器执行一个成功的写操作, 写入 PBLKEY 字段的值必须为 0x5FA0。如果写入这个字段值不等于 0x5FA0, PBLOCKR 寄存器上的任何写操作将被中止。对 PBLKEY 字段的一个读操作的结果将返回 GPIO 端口 B 锁定状态, 表示 GPIO 端口 B 是否被锁定。如果读值, PBLKEY 字段为 0, 表明 GPIO 端口 B 锁定功能除能。否则, 它表明 GPIO 端口 B 锁定功能使能, 读值等于 1。
[15:0]	PBLOCKn	GPIO 端口 B 引脚 n 的锁定控制位 (n = 0 ~ 15) 0: 端口 B 引脚 n 未锁定 1: 端口 B 引脚 n 锁定 当正确的锁定键应用于 PBLKEY 字段, PBLOCKn 位用来锁定相应 GPIO 引脚的配置。被锁定的配置包括 PBDIRn、PBINENn、PBPUn、PBPDn 和 PBODn 设置相关的 GPIO 寄存器。另外, 用来配置为相关 GPIO 引脚的复用功能的 AFIO 寄存器 GPBCFGHR 或 GPBCFGRLR 字段也将被锁定。请注意, PBLOCKR 寄存器只能写入一次, 这意味着 PBLKEY 和 PBLOCKn (锁定控制位) 应一起写入, 不能改变直到系统复位或 GPIO 端口 B 发生复位。

端口 B 数据输入寄存器 – PBDINR

该寄存器指定 GPIO 端口 B 的输入数据。

偏移量: 0x01C  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PBDIN								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	PBDIN								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[15:0]	PBDINn	GPIO 端口 B 引脚 n 的数据输入位 (n = 0 ~ 15) 0: 引脚 n 的输入数据为 0 1: 引脚 n 的输入数据为 1

端口 B 输出数据寄存器 – PBDOUTR

该寄存器指定 GPIO 端口 B 的输出数据。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PBDOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PBDOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PBDOUTn	GPIO 端口 B 引脚 n 的数据输出位 (n = 0 ~ 15) 0: 引脚 n 的输出数据为 0 1: 引脚 n 的输出数据为 1

## 端口 B 输出置位 / 复位控制寄存器 – PBSRR

该寄存器用来置位或复位 GPIO 端口 B 输出数据相应的位。

偏移量: 0x024

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PBRST								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	PBRST								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	PBSET								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	PBSET								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO

位	字段	描述
[31:16]	PBRSTn	GPIO 端口 B 引脚 n 的输出复位控制位 (n = 0 ~ 15) 0: 不影响 PBDOUTn 位 1: 复位 PBDOUTn 位
[15:0]	PBSETn	GPIO 端口 B 引脚 n 的输出置位控制位 (n = 0 ~ 15) 0: 不影响 PBDOUTn 位 1: 置位 PBDOUTn 位 注意, 如果 PBSETn 和 PBRSTn 位同时被置位, PBSETn 位使能的功能有比较高的优先级。

端口 B 输出复位寄存器 – PBRR

该寄存器用来复位 GPIO 端口 B 输出数据相应的位。

偏移量: 0x028  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PBRST								
	7	6	5	4	3	2	1	0	
类型 / 复位	PBRST								
	WO	0	WO	0	WO	0	WO	0	WO
	0		0		0		0		0

位	字段	描述
[15:0]	PBRSTn	GPIO 端口 B 引脚 n 的输出复位位 (n = 0 ~ 15) 0: 不影响 PBDOUTn 位 1: 复位 PBDOUTn 位

端口 C 数据方向控制寄存器 – PCDIRCR

该寄存器用来控制 GPIO 端口 C 引脚的方向作为输入或输出。

偏移量： 0x000  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PCDIR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PCDIR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PCDIRn	GPIO 端口 C 引脚 n 的方向控制位 (n = 0 ~ 15) 0: 引脚 n 处于输入模式 1: 引脚 n 处于输出模式

端口 C 输入功能使能控制寄存器 – PCINER

此寄存器用来使能或除能 GPIO 端口 C 的输入功能。

偏移量： 0x004  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PCINEN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PCINEN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PCINEN <sub>n</sub>	GPIO 端口 C 引脚 n 的输入使能控制位 (n = 0 ~ 15) 0: 除能引脚 n 的输入功能 1: 使能引脚 n 的输入功能 当除能引脚 n 的输入功能时，输入施密特触发器将关闭，施密特触发器输出将保持在零状态。

端口 C 上拉选择寄存器 – PCPUR

该寄存器用来使能或除能 GPIO 端口 C 的上拉功能。

偏移量： 0x008  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	PCPU							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	PCPU							
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PCPUn	GPIO 端口 C 引脚 n 的上拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的上拉功能 1: 使能引脚 n 的上拉功能 注: 当上拉和下拉功能都使能, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。



端口 C 下拉选择寄存器 – PCPDR

该寄存器用来使能或除能 GPIO 端口 C 的下拉功能。

偏移量： 0x00C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PCPD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
类型 / 复位	PCPD								
	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PCPDn	GPIO 端口 C 引脚 n 的下拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的下拉功能 1: 使能引脚 n 的下拉功能 注: 当上拉和下拉功能都使能时, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 C 漏极开路选择寄存器 – PCODR

该寄存器用来使能或除能 GPIO 端口 C 的漏极开路功能。

偏移量： 0x010  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PCOD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PCOD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PCODn	GPIO 端口 C 引脚 n 的漏极开路选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的漏极开路输出 (输出类型为 CMOS 输出) 1: 使能引脚 n 的漏极开路输出 (输出类型为漏极开路) 注: 当漏极开路功能使能时, 引脚 n 内部上拉或下拉设置将无效。

端口 C 输出电流驱动选择寄存器 – PCDRV

该寄存器用来定义 GPIO 端口 C 的输出驱动电流。

偏移量： 0x014  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	PCDV15		PCDV14		PCDV13		PCDV12	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	PCDV11		PCDV10		PCDV9		PCDV8	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	PCDV7		PCDV6		PCDV5		PCDV4	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PCDV3		PCDV2		PCDV1		PCDV0	
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:0]	PCDVn[1:0]	GPIO 端口 C 引脚 n 的输出电流驱动选择控制位 (n = 0 ~ 15) 00: 4 mA 源 / 灌电流 01: 8 mA 源 / 灌电流 10: 12 mA 源 / 灌电流 11: 16 mA 源 / 灌电流 由于 PC13 ~ 15 位于备份域内，因此只有灌电流能力可以由 PCDV[0] 位设置而与 PCDV[1] 位无关。 x0: 4 mA 灌电流 x1: 8 mA 灌电流 PC13 ~15 引脚的源电流始终限制在 1 mA。

端口 C 锁定寄存器 – PCLOCKR

该寄存器定义了 GPIO 端口 C 的锁定配置。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PCLKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PCLKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PCLOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PCLOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:16]	PCLKEY	GPIO 端口 C 锁定键 0x5FA0: 使能端口 C 锁定功能 其它: 除能端口 C 锁定功能 为了锁定端口 C 功能, 在该寄存器 PCLKEY 字段写入值 0x5FA0。为了在这个锁定寄存器执行一个成功的写操作, 写入 PCLKEY 字段的值必须为 0x5FA0。如果写入这个字段值不等于 0x5FA0, PCLOCKR 寄存器上的任何写操作将被中止。对 PCLKEY 字段的一个读操作的结果将返回 GPIO 端口 C 锁定状态, 表示 GPIO 端口 C 是否被锁定。如果读值, PCLKEY 字段为 0, 表明 GPIO 端口 C 锁定功能除能。否则, 它表明 GPIO 端口 C 锁定功能使能, 读值等于 1。
[15:0]	PCLOCKn	GPIO 端口 C 引脚 n 锁定控制位 (n = 0 ~ 15) 0: 端口 C 引脚 n 未锁定 1: 端口 C 引脚 n 锁定 当正确的锁定键应用于 PCLKEY 字段, PCLOCKn 位用来锁定相应 GPIO 引脚的配置。被锁定的配置包括 PCDIRn、PCINENn、PCPUn、PCPDn 和 PCODn 设置相关的 GPIO 寄存器。另外, 用来配置为相关 GPIO 引脚的复用功能的 AFIO 寄存器 GPCCFGHR 或 GPCCFGLR 字段也将被锁定。请注意, PCLOCKR 寄存器只能写入一次, 这意味着 PCLKEY 和 PCLOCKn (锁定控制位) 应一起写入, 不能改变直到系统复位或 GPIO 端口 C 发生复位。

端口 C 数据输入寄存器 – PCDINR

该寄存器指定 GPIO 端口 C 的输入数据。

偏移量: 0x01C  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PCDIN								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	PCDIN								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[15:0]	PCDINn	GPIO 端口 C 引脚 n 的数据输入位 (n = 0 ~ 15) 0: 引脚 n 的输入数据为 0 1: 引脚 n 的输入数据为 1

端口 C 输出数据寄存器 – PCDOUTR

该寄存器指定 GPIO 端口 C 的输出数据。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PCDOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PCDOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PCDOUTn	GPIO 端口 C 引脚 n 的数据输出位 (n = 0 ~ 15) 0: 引脚 n 的输出数据为 0 1: 引脚 n 的输出数据为 1

## 端口 C 输出置位 / 复位控制寄存器 – PCSRR

该寄存器用来置位或复位 GPIO 端口 C 输出数据相应的位。

偏移量: 0x024

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PCRST								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	PCRST								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	PCSET								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	PCSET								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO

位	字段	描述
[31:16]	PCRSTn	GPIO 端口 C 引脚 n 的输出复位控制位 (n = 0 ~ 15) 0: 不影响 PCDOUn 位 1: 复位 PCDOUn 位
[15:0]	PCSETn	GPIO 端口 C 引脚 n 的输出置位控制位 (n = 0 ~ 15) 0: 不影响 PCDOUn 位 1: 置位 PCDOUn 位 注意, 如果 PCSETn 和 PCRSTn 位同时被置位, PCSETn 位使能的功能有比较高的优先级。

端口 C 输出复位寄存器 – PCRR

该寄存器用来复位 GPIO 端口 C 输出数据相应的位。

偏移量: 0x028  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PCRST								
	7	6	5	4	3	2	1	0	
类型 / 复位	PCRST								
	WO	0	WO	0	WO	0	WO	0	WO
	0		0		0		0		0

位	字段	描述
[15:0]	PCRSTn	GPIO 端口 C 引脚 n 的输出复位位 (n = 0 ~ 15) 0: 不影响 PCDOUn 位 1: 复位 PCDOUn 位



端口 D 数据方向控制寄存器 – PDDIRCR

该寄存器用来控制 GPIO 端口 D 引脚的方向作为输入或输出。

偏移量： 0x000  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	PDDIR							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PDDIR							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PDDIRn	GPIO 端口 D 引脚 n 的方向控制位 (n = 0 ~ 15) 0: 引脚 n 处于输入模式 1: 引脚 n 处于输出模式

端口 D 输入功能使能控制寄存器 – PDINER

此寄存器用来使能或除能 GPIO 端口 D 的输入功能。

偏移量： 0x004  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PDINEN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PDINEN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PDINENn	GPIO 端口 D 引脚 n 的输入使能控制位 (n = 0 ~ 15) 0: 除能引脚 n 的输入功能 1: 使能引脚 n 的输入功能 当除能引脚 n 的输入功能时，输入施密特触发器将关闭，施密特触发器输出将保持在零状态。

端口 D 上拉选择寄存器 – PDPUR

该寄存器用来使能或除能 GPIO 端口 D 的上拉功能。

偏移量： 0x008  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	PDPUn							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	PDPUn							
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PDPUn	GPIO 端口 D 引脚 n 的上拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的上拉功能 1: 使能引脚 n 的上拉功能 注: 当上拉和下拉功能都使能, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 D 下拉选择寄存器 – PDPDR

该寄存器用来使能或除能 GPIO 端口 D 的下拉功能。

偏移量： 0x00C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PDPD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PDPD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PDPDn	GPIO 端口 D 引脚 n 的下拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的下拉功能 1: 使能引脚 n 的下拉功能 注: 当上拉和下拉功能都使能时, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 D 漏极开路选择寄存器 – PDODR

该寄存器用来使能或除能 GPIO 端口 D 的漏极开路功能。

偏移量： 0x010  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PDOD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PDOD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PDODn	GPIO 端口 D 引脚 n 的漏极开路选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的漏极开路输出 (输出类型为 CMOS 输出) 1: 使能引脚 n 的漏极开路输出 (输出类型为漏极开路) 注: 当漏极开路功能使能时, 引脚 n 内部上拉或下拉设置将无效。

## 端口 D 输出电流驱动选择寄存器 – PDDVR

该寄存器用来定义 GPIO 端口 D 的输出驱动电流。

偏移量: 0x014

复位值: 0x0000\_0000

	31				30				29				28				27				26				25				24			
	PDDV15								PDDV14								PDDV13								PDDV12							
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0				
	23				22				21				20				19				18				17				16			
	PDDV11								PDDV10								PDDV9								PDDV8							
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0				
	15				14				13				12				11				10				9				8			
	PDDV7								PDDV6								PDDV5								PDDV4							
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0				
	7				6				5				4				3				2				1				0			
	PDDV3								PDDV2								PDDV1								PDDV0							
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0				

位	字段	描述
[31:0]	PDDVn[1:0]	GPIO 端口 D 引脚 n 的输出电流驱动选择控制位 (n = 0 ~ 15) 00: 4 mA 源 / 灌电流 01: 8 mA 源 / 灌电流 10: 12 mA 源 / 灌电流 11: 16 mA 源 / 灌电流

端口 D 锁定寄存器 – PDLOCKR

该寄存器定义了 GPIO 端口 D 的锁定配置。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PDLKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PDLKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PDLOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PDLOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:16]	PDLKEY	GPIO 端口 D 锁定键 0x5FA0: 使能端口 D 锁定功能 其它: 除能端口 D 锁定功能 为了锁定端口 D 功能, 在该寄存器 PDLKEY 字段写入值 0x5FA0。为了在这个锁定寄存器执行一个成功的写操作, 写入 PDLKEY 字段的值必须为 0x5FA0。如果写入这个字段值不等于 0x5FA0, PDLOCKR 寄存器上的任何写操作将被中止。对 PDLKEY 字段的一个读操作的结果将返回 GPIO 端口 D 锁定状态, 表示 GPIO 端口 D 是否被锁定。如果读值, PDLKEY 字段为 0, 表明 GPIO 端口 D 锁定功能除能。否则, 它表明 GPIO 端口 D 锁定功能使能, 读值等于 1。
[15:0]	PDLOCKn	GPIO 端口 D 引脚 n 的锁定控制位 (n = 0 ~ 15) 0: 端口 D 引脚 n 未锁定 1: 端口 D 引脚 n 锁定 当正确的锁定键应用于 PDLKEY 字段, PDLOCKn 位用来锁定相应 GPIO 引脚的配置。被锁定的配置包括 PDDIRn、PDINENn、PDPUn、PDPDn 和 PDODn 设置相关的 GPIO 寄存器。另外, 用来配置为相关 GPIO 引脚的复用功能的 AFIO 寄存器 GPDCFGHR 或 GPD CFGLR 字段也将被锁定。请注意, PDLOCKR 寄存器只能写入一次, 这意味着 PDLKEY 和 PDLOCKn (锁定控制位) 应一起写入, 不能改变直到系统复位或 GPIO 端口 D 发生复位。

端口 D 数据输入寄存器 – PDDINR

该寄存器指定 GPIO 端口 D 的输入数据。

偏移量: 0x01C  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PDDIN								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	PDDIN								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[15:0]	PDDINn	GPIO 端口 D 引脚 n 的数据输入位 (n = 0 ~ 15) 0: 引脚 n 的输入数据为 0 1: 引脚 n 的输入数据为 1



端口 D 输出数据寄存器 – PDDOUTR

该寄存器指定 GPIO 端口 D 的输出数据。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PDDOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PDDOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PDDOUTn	GPIO 端口 D 引脚 n 的数据输出位 (n = 0 ~ 15) 0: 引脚 n 的输出数据为 0 1: 引脚 n 的输出数据为 1

端口 D 输出置位 / 复位控制寄存器 – PDSRR

该寄存器用来置位或复位 GPIO 端口 D 输出数据相应的位。

偏移量: 0x024  
复位值: 0x0000\_0000

	31		30		29		28		27		26		25		24	
	PDRST															
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	23		22		21		20		19		18		17		16	
	PDRST															
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	15		14		13		12		11		10		9		8	
	PDSET															
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	7		6		5		4		3		2		1		0	
	PDSET															
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0

位	字段	描述
[31:16]	PDRSTn	GPIO 端口 D 引脚 n 的输出复位控制位 (n = 0 ~ 15) 0: 不影响 PDDOUTn 位 1: 复位 PDDOUTn 位
[15:0]	PDSETn	GPIO 端口 D 引脚 n 的输出置位控制位 (n = 0 ~ 15) 0: 不影响 PDDOUTn 位 1: 置位 PDDOUTn 位 注意, 如果 PDSETn 和 PDRSTn 位同时被置位, PDSETn 位使能的功能有比较高的优先级。

端口 D 输出复位寄存器 – PDRR

该寄存器用来复位 GPIO 端口 D 输出数据相应的位。

偏移量： 0x028  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PDRST								
	7	6	5	4	3	2	1	0	
类型 / 复位	PDRST								
	WO	0	WO	0	WO	0	WO	0	WO
	0		0		0		0		0

位	字段	描述
[15:0]	PDRSTn	GPIO 端口 D 引脚 n 的输出复位位 (n = 0 ~ 15) 0: 不影响 PDDOUTn 位 1: 复位 PDDOUTn 位

端口 E 数据方向控制寄存器 – PEDIRCR

该寄存器用来控制 GPIO 端口 E 引脚的方向作为输入或输出。

偏移量：0x000

复位值：0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	PEDIR							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PEDIR							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PEDIRn	GPIO 端口 E 引脚 n 的方向控制位 (n = 0 ~ 15) 0: 引脚 n 处于输入模式 1: 引脚 n 处于输出模式

端口 E 输入功能使能控制寄存器 – PEINER

此寄存器用来使能或除能 GPIO 端口 E 的输入功能。

偏移量： 0x004  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PEINEN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PEINEN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PEINENn	GPIO 端口 E 引脚 n 的输入使能控制位 (n = 0 ~ 15) 0: 除能引脚 n 的输入功能 1: 使能引脚 n 的输入功能 当除能引脚 n 的输入功能时，输入施密特触发器将关闭，施密特触发器输出将保持在零状态。

端口 E 上拉选择寄存器 – PEPUR

该寄存器用来使能或除能 GPIO 端口 E 的上拉功能。

偏移量： 0x008

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PEPU								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PEPU								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PEPUn	GPIO 端口 E 引脚 n 的上拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的上拉功能 1: 使能引脚 n 的上拉功能 注: 当上拉和下拉功能都使能, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 E 下拉选择寄存器 – PEPDR

该寄存器用来使能或除能 GPIO 端口 E 的下拉功能。

偏移量： 0x00C

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PEPD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
类型 / 复位	PEPD								
	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PEPDn	GPIO 端口 E 引脚 n 的下拉选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的下拉功能 1: 使能引脚 n 的下拉功能 注: 当上拉和下拉功能都使能时, 上拉功能将有更高的优先级, 因此下拉功能将被封锁和除能。

端口 E 漏极开路选择寄存器 – PEODR

该寄存器用来使能或除能 GPIO 端口 E 的漏极开路功能。

偏移量： 0x010

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PEOD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
类型 / 复位	PEOD								
	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PEODn	GPIO 端口 E 引脚 n 的漏极开路选择控制位 (n = 0 ~ 15) 0: 除能引脚 n 的漏极开路输出 (输出类型为 CMOS 输出) 1: 使能引脚 n 的漏极开路输出 (输出类型为漏极开路) 注: 当漏极开路功能使能时, 引脚 n 内部上拉或下拉设置将无效。



端口 E 输出电流驱动选择寄存器 – PEDRVR

该寄存器用来定义 GPIO 端口 E 的输出驱动电流。

偏移量： 0x014  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	PEDV15		PEDV14		PEDV13		PEDV12	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	PEDV11		PEDV10		PEDV9		PEDV8	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	PEDV7		PEDV6		PEDV5		PEDV4	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PEDV3		PEDV2		PEDV1		PEDV0	
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:0]	PEDVn[1:0]	GPIO 端口 E 引脚 n 的输出电流驱动选择控制位 (n = 0 ~ 15) 00: 4 mA 源 / 灌电流 01: 8 mA 源 / 灌电流 10: 12 mA 源 / 灌电流 11: 16 mA 源 / 灌电流

## 端口 E 锁定寄存器 – PELOCKR

该寄存器定义了 GPIO 端口 E 的锁定配置。

偏移量: 0x018

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PELKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PELKEY								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PELOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PELOCK								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:16]	PELKEY	GPIO 端口 E 锁定键 0x5FA0: 使能端口 E 锁定功能 其它: 除能端口 E 锁定功能 为了锁定端口 E 功能, 在该寄存器 PELKEY 字段写入值 0x5FA0。为了在这个锁定寄存器执行一个成功的写操作, 写入 PELKEY 字段的值必须为 0x5FA0。如果写入这个字段值不等于 0x5FA0, PELOCKR 寄存器上的任何写操作将被中止。对 PELKEY 字段的一个读操作的结果将返回 GPIO 端口 E 锁定状态, 表示 GPIO 端口 E 是否被锁定。如果读值, PELKEY 字段为 0, 表明 GPIO 端口 E 锁定功能除能。否则, 它表明 GPIO 端口 E 锁定功能使能, 读值等于 1。
[15:0]	PELOCKn	GPIO 端口 E 引脚 n 的锁定控制位 (n = 0 ~ 15) 0: 端口 E 引脚 n 未锁定 1: 端口 E 引脚 n 锁定 当正确的锁定键应用于 PELKEY 字段, PELOCKn 位用来锁定相应 GPIO 引脚的配置。被锁定的配置包括 PEDIRn、PEINENn、PEPUn、PEPDn 和 PEODn 设置相关的 GPIO 寄存器。另外, 用来配置为相关 GPIO 引脚的复用功能的 AFIO 寄存器 GPECFGHR 或 GPECFGLR 字段也将被锁定。请注意, PELOCKR 寄存器只能写入一次, 这意味着 PELKEY 和 PELOCKn (锁定控制位) 应一起写入, 不能改变直到系统复位或 GPIO 端口 E 发生复位。

端口 E 数据输入寄存器 – PEDINR

该寄存器指定 GPIO 端口 E 的输入数据。

偏移量: 0x01C  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PEDIN								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	PEDIN								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[15:0]	PEDIN <sub>n</sub>	GPIO 端口 E 引脚 n 的数据输入位 (n = 0 ~ 15) 0: 引脚 n 的输入数据为 0 1: 引脚 n 的输入数据为 1

端口 E 输出数据寄存器 – PEDOUTR

该寄存器指定 GPIO 端口 E 的输出数据。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PEDOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PEDOUT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PEDOUTn	GPIO 端口 E 引脚 n 的数据输出位 (n = 0 ~ 15) 0: 引脚 n 的输出数据为 0 1: 引脚 n 的输出数据为 1

## 端口 E 输出置位 / 复位控制寄存器 – PESRR

该寄存器用来置位或复位 GPIO 端口 E 输出数据相应的位。

偏移量: 0x024

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PERST								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	PERST								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	PESET								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	PESET								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO

位	字段	描述
[31:16]	PERSTn	GPIO 端口 E 引脚 n 的输出复位控制位 (n = 0 ~ 15) 0: 不影响 PEDOUTn 位 1: 复位 PEDOUTn 位
[15:0]	PESETn	GPIO 端口 E 引脚 n 的输出置位控制位 (n = 0 ~ 15) 0: 不影响 PEDOUTn 位 1: 置位 PEDOUTn 位 注意, 如果 PESETn 和 PERSTn 位同时被置位, PESETn 位使能的功能有比较高的优先级。

端口 E 输出复位寄存器 – PERR

该寄存器用来复位 GPIO 端口 E 输出数据相应的位。

偏移量： 0x028  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	PERST								
	7	6	5	4	3	2	1	0	
类型 / 复位	PERST								
	WO	0	WO	0	WO	0	WO	0	WO
	0		0		0		0		0

位	字段	描述
[15:0]	PERSTn	GPIO 端口 E 引脚 n 的输出复位位 (n = 0 ~ 15) 0: 不影响 PEDOUTn 位 1: 复位 PEDOUTn 位

## 9 复用功能输入 / 输出控制单元 (AFIO)

### 简介

为了扩大 GPIO 或外设功能使用的灵活性, 每个 I/O 引脚可通过设置 GPxCFGLR 或 GPxCFGHR 寄存器 (其中 x 为不同引脚的名称) 配置为多达 16 个不同的功能, 如 GPIO 或 IP 功能。根据 IP 资源的使用情况和应用需求, 可以使用外设 I/O 重映射机制来选择各功能从哪个引脚引出。此外, 通过设置 ESSRn 寄存器中 EXTInPIN[3:0] 字段, 可以选择任意的 GPIO 引脚作为 EXTI 中断引脚来触发一个中断或事件。更多详细信息请参阅 EXTI 部分。

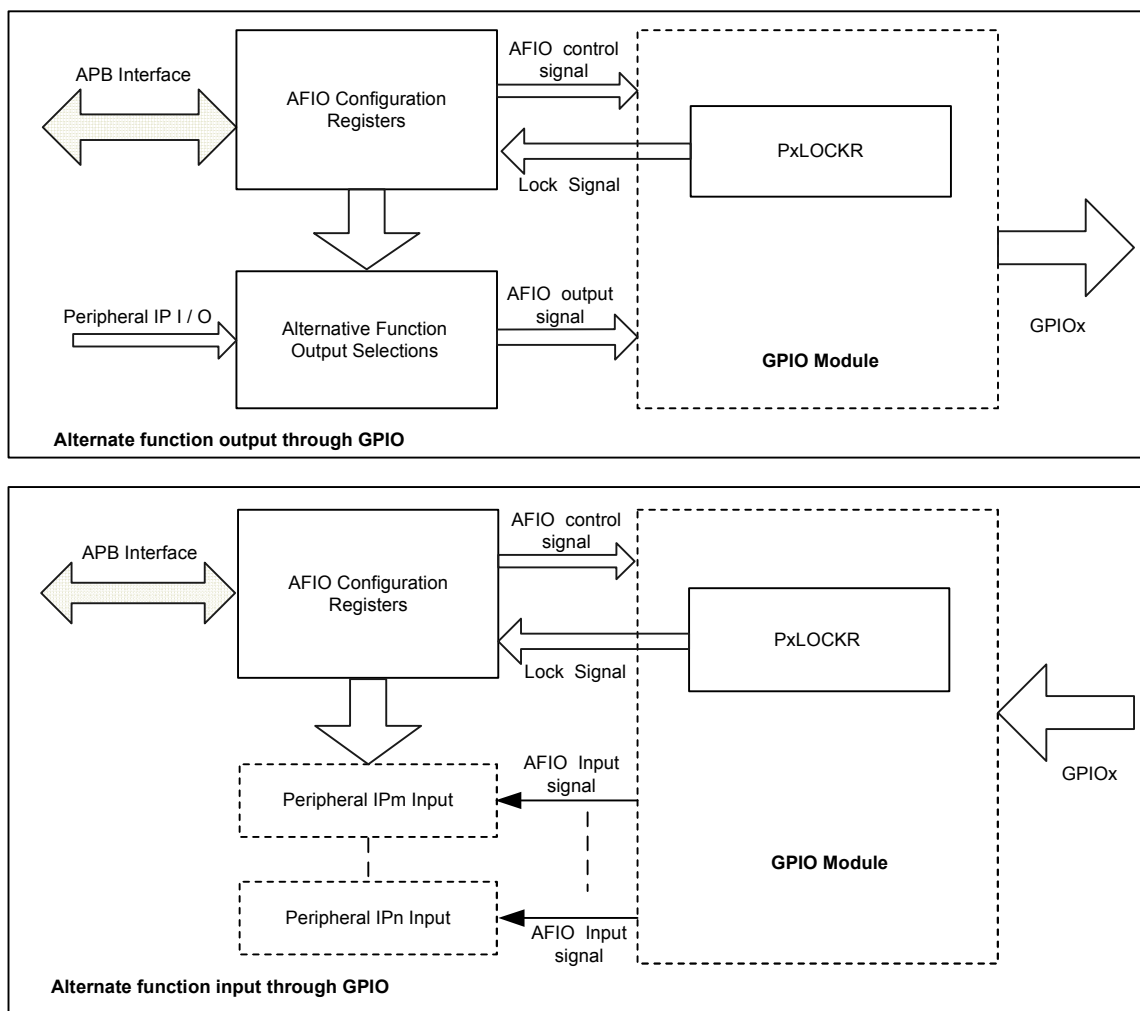


图 23. AFIO 方框图

## 特性

- APB 从机接口用于寄存器访问
- EXTI 来源选择
- 每个 GPIO 引脚的可配置功能，每个引脚上多达 16 个复用功能
- AFIO 锁定机制

## 功能描述

### 外部中断引脚选择

GPIO 引脚连接到 16 个 EXTI 引脚如附图所示。例如，用户可以设置 ESSR0 寄存器中 EXTI0PIN[3:0] 字段为 b0000 来选择 GPIO PA0 引脚作为 EXTI 0 输入。由于并非所有端口 A ~ E 的引脚在所有封装类型都是可用的，详细的引脚信息请参阅引脚图。当相应的引脚不可用时，EXTInPIN[3:0] 字段的设置是无效的。

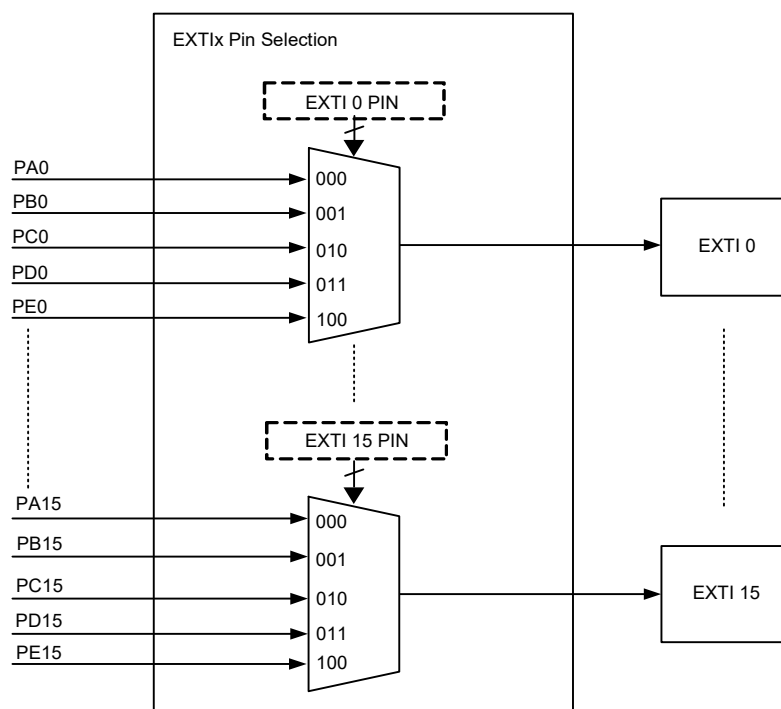


图 24. EXTI 输入通道选择



## 复用功能

每个 I/O 引脚有多达 16 种可供选择的功能, 可以通过设置 GPxCFGLR 或 GPxCFGHR (n = 0 ~ 15, x = A ~ E) 寄存器中 PxCFGn[3:0] 字段选择。可选功能 I/O 引脚的详细分配请参阅数据手册 (Datasheet) 中复用功能列表说明部分。除了这种机制外, 每个外设设有可选功能分别分配到不同的引脚, 在小封装单片机中优化了外设的可用数量。下面的说明显示 PxCFGn[3:0] 字段的设置。应注意, 如果比较器有效, PB[8:6] 或 PB[11:9] 不能同时设置为其它 AFIO 功能。

- PxCFGn[3:0] = 0000: 默认复用功能 (复位后, AF0)
- PxCFGn[3:0] = 0001: 复用功能 1 (AF1)
- PxCFGn[3:0] = 0010: 复用功能 2 (AF2)
- .....
- PxCFGn[3:0] = 1110: 复用功能 14 (AF14)
- PxCFGn[3:0] = 1111: 复用功能 15 (AF15)

表 24. 外设 AFIO 分配列表范例

AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
系统默认	GPIO	ADC	CMP	MCTM/GPTM	SPI	USART/UART	I <sup>2</sup> C	SCI	EBI	I <sup>2</sup> S	SDIO	CSIF	N/A	N/A	系统其它

## 锁定机制

该系列单片机具有锁定功能, 使用 GPIO 锁定寄存器 PxLOCKR (x = A ~ E), 可以通过锁定键 0x5FA0 来锁定 AFIO 配置, 直到复位发生。

## 寄存器列表

下表显示 AFIO 寄存器和复位值。

表 25. AFIO 寄存器列表

寄存器	偏移量	描述	复位值
ESSR0	0x000	EXTI 来源选择寄存器 0	0x0000_0000
ESSR1	0x004	EXTI 来源选择寄存器 1	0x0000_0000
GPACFGLR	0x020	GPIO 端口 A AFIO 配置低寄存器	0x0000_0000
GPACFGHR	0x024	GPIO 端口 A AFIO 配置高寄存器	0x0000_0000
GPBCFGLR	0x028	GPIO 端口 B AFIO 配置低寄存器	0x0000_0000
GPBCFGHR	0x02C	GPIO 端口 B AFIO 配置高寄存器	0x0000_0000
GPCCFGLR	0x030	GPIO 端口 C AFIO 配置低寄存器	0x0000_0000
GPCCFGHR	0x034	GPIO 端口 C AFIO 配置高寄存器	0x0000_0000
GPDCFGLR	0x038	GPIO 端口 D AFIO 配置低寄存器	0x0000_0000
GPDCFGHR	0x03C	GPIO 端口 D AFIO 配置高寄存器	0x0000_0000
GPECFGLR	0x040	GPIO 端口 E AFIO 配置低寄存器	0x0000_0000
GPECFGHR	0x044	GPIO 端口 E AFIO 配置高寄存器	0x0000_0000

寄存器描述

EXTI 来源选择寄存器 0 – ESSR0

该寄存器定义了 EXTI0 ~ EXTI7 的 I/O 选项。

偏移量: 0x000  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	EXTI7PIN				EXTI6PIN			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	EXTI5PIN				EXTI4PIN			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	EXTI3PIN				EXTI2PIN			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	EXTI1PIN				EXTI0PIN			
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:0]	EXTInPIN[3:0]	EXTIn 引脚选择 (n = 0 ~ 7) 0000: PA 位 n 被选作 EXTIn 来源信号 0001: PB 位 n 被选作 EXTIn 来源信号 0010: PC 位 n 被选作 EXTIn 来源信号 0011: PD 位 n 被选作 EXTIn 来源信号 0100: PE 位 n 被选作 EXTIn 来源信号 其它: 保留位 注: 由于并非所有端口的引脚在所有产品和封装类型都是可用的, 详细的引脚信息请参阅引脚图。当相应的引脚不可用时, EXTInPIN[3:0] 字段的设置是无效的。

EXTI 来源选择寄存器 1 – ESSR1

该寄存器定义了 EXTI8 ~ EXTI15 的 I/O 选项。

偏移量: 0x004  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	EXTI15PIN				EXTI14PIN			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	EXTI13PIN				EXTI12PIN			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	EXTI11PIN				EXTI10PIN			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	EXTI9PIN				EXTI8PIN			
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:0]	EXTInPIN[3:0]	EXTIn 引脚选择 (n = 8 ~ 15) 0000: PA 位 n 被选作 EXTIn 来源信号 0001: PB 位 n 被选作 EXTIn 来源信号 0010: PC 位 n 被选作 EXTIn 来源信号 0011: PD 位 n 被选作 EXTIn 来源信号 0100: PE 位 n 被选作 EXTIn 来源信号 其它: 保留位 注: 由于并非所有端口的引脚在所有产品和封装类型都是可用的, 详细的引脚信息 请参阅引脚图。当相应的引脚不可用时, EXTInPIN[3:0] 字段的设置是无效的。

**GPIO 端口 x 配置低寄存器 – GPxCFGnLR, x = A, B, C, D, E**

该寄存器定义了 GPIO 端口 x (x = A, B, C, D, E) 的复用功能。

偏移量: 0x020, 0x028, 0x030, 0x038, 0x040  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	PxCFG7				PxCFG6			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	PxCFG5				PxCFG4			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	PxCFG3				PxCFG2			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PxCFG1				PxCFG0			
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:0]	PxCFGn[3:0]	端口 x 引脚 n 复用功能选择 (n = 0 ~ 7) 0000: 端口 x 引脚 n 选择作为 AF0 0001: 端口 x 引脚 n 选择作为 AF1 ... 1110: 端口 x 引脚 n 选择作为 AF14 1111: 端口 x 引脚 n 选择作为 AF15 请参考单片机规格书中的复用功能列表获取复用功能的更多信息。

**GPIO 端口 x 配置高寄存器 – GPxCFGHR, x = A, B, C, D, E**

该寄存器定义了 GPIO 端口 x (x = A, B, C, D, E) 的复用功能。

偏移量: 0x024, 0x02C, 0x034, 0x03C, 0x044  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	PxCFG15							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	PxCFG13							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	PxCFG11							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PxCFG9							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:0]	PxCFGn[3:0]	端口 x 引脚 n 复用功能选择 (n = 8 ~ 15) 0000: 端口 x 引脚 n 选择作为 AF0 0001: 端口 x 引脚 n 选择作为 AF1 ... 1110: 端口 x 引脚 n 选择作为 AF14 1111: 端口 x 引脚 n 选择作为 AF15 请参考单片机规格书中的复用功能列表获取复用功能的更多信息。

# 10 嵌套向量中断控制器 (NVIC)

## 简介

为了减少延迟并提高中断处理效率，Cortex®-M3 提供了紧密耦合集成部分，即嵌套向量中断控制器 (NVIC)。NVIC 控制系统异常及包括如使能 / 除能控制、优先级、清除 - 挂起、有效状态报告、软件触发和向量表重置功能的外设中断。更多信息请参考“Cortex®-M3 技术参考手册”。

此外，Cortex®-M3 内置简单的 24-bit 向下计数定时器 (SysTick) 用来作为实时定时器操作系统 (RTOS) 的节拍定时器，或作为一个简单的计数器。SysTick 从预设值向下计数，当它到达零时，产生一个系统中断。

附表中列出 16 个系统异常类型和多个外设中断。

表 26. 异常类型

异常类型	优先级	中断号	异常号	向量地址	描述
—	—	—	0	0x000	初始堆栈指针值
复位	-3 (最高)	—	1	0x004	复位
NMI	-2	—	2	0x008	非屏蔽中断。时钟故障中断信号 (时钟控制单元提供时钟监控功能) 连接到 NMI 输入
硬件故障	-1	—	3	0x00C	所有故障类型
存储器管理	可配置 <sup>(1)</sup>	—	4	0x010	存储器保护单元 (MPU) 不匹配，包括访问冲突和不匹配
总线故障	可配置 <sup>(1)</sup>	—	5	0x014	预抓取故障，存储器访问故障和其它地址 / 存储器相关的故障
使用故障	可配置 <sup>(1)</sup>	—	6	0x018	使用故障，例如执行未定义的指令或试图进行非法的状态转变
—	—	—	7	0x01C	保留
—	—	—	8	0x020	保留
—	—	—	9	0x024	保留
—	—	—	10	0x028	保留
SVCCall	可配置 <sup>(1)</sup>	—	11	0x02C	SVC 指令的系统服务调用
调试监控	可配置 <sup>(1)</sup>	—	12	0x030	未处于 halt 时，为调试监控器
—	—	—	13	0x034	保留
PendSV	可配置 <sup>(1)</sup>	—	14	0x038	系统服务的可挂起请求
SysTick	可配置 <sup>(1)</sup>	—	15	0x03C	SysTick 定时器递减为零
CKRDY	可配置 <sup>(2)</sup>	0	16	0x040	时钟就绪中断 (HSE、HIS、LSE、LSI 或 PLL)
LVD	可配置 <sup>(2)</sup>	1	17	0x044	低压检测中断
BOD	可配置 <sup>(2)</sup>	2	18	0x048	掉电检测中断
—	—	3	19	0x04C	保留
RTC	可配置 <sup>(2)</sup>	4	20	0x050	RTC 总中断
FMC	可配置 <sup>(2)</sup>	5	21	0x054	FMC 总中断
EVWUP	可配置 <sup>(2)</sup>	6	22	0x058	EXTI 事件唤醒中断
LPWUP	可配置 <sup>(2)</sup>	7	23	0x05C	WAKEUP 引脚中断
EXTI0	可配置 <sup>(2)</sup>	8	24	0x060	EXTI 引脚 0 中断
EXTI1	可配置 <sup>(2)</sup>	9	25	0x064	EXTI 引脚 1 中断

异常类型	优先级	中断号	异常号	向量地址	描述
EXTI2	可配置 <sup>(2)</sup>	10	26	0x068	EXTI 引脚 2 中断
EXTI3	可配置 <sup>(2)</sup>	11	27	0x06C	EXTI 引脚 3 中断
EXTI4	可配置 <sup>(2)</sup>	12	28	0x070	EXTI 引脚 4 中断
EXTI5	可配置 <sup>(2)</sup>	13	29	0x074	EXTI 引脚 5 中断
EXTI6	可配置 <sup>(2)</sup>	14	30	0x078	EXTI 引脚 6 中断
EXTI7	可配置 <sup>(2)</sup>	15	31	0x07C	EXTI 引脚 7 中断
EXTI8	可配置 <sup>(2)</sup>	16	32	0x080	EXTI 引脚 8 中断
EXTI9	可配置 <sup>(2)</sup>	17	33	0x084	EXTI 引脚 9 中断
EXTI10	可配置 <sup>(2)</sup>	18	34	0x088	EXTI 引脚 10 中断
EXTI11	可配置 <sup>(2)</sup>	19	35	0x08C	EXTI 引脚 11 中断
EXTI12	可配置 <sup>(2)</sup>	20	36	0x090	EXTI 引脚 12 中断
EXTI13	可配置 <sup>(2)</sup>	21	37	0x094	EXTI 引脚 13 中断
EXTI14	可配置 <sup>(2)</sup>	22	38	0x098	EXTI 引脚 14 中断
EXTI15	可配置 <sup>(2)</sup>	23	39	0x09C	EXTI 引脚 15 中断
COMP	可配置 <sup>(2)</sup>	24	40	0x0A0	比较器总中断
ADC	可配置 <sup>(2)</sup>	25	41	0x0A4	A/D 转换器总中断
—	—	26	42	0x0A8	保留
MCTM0_BRK	可配置 <sup>(2)</sup>	27	43	0x0AC	MCTM0 暂停中断
MCTM0_UP	可配置 <sup>(2)</sup>	28	44	0x0B0	MCTM0 更新中断
MCTM0_TR_UP2	可配置 <sup>(2)</sup>	29	45	0x0B4	MCTM0 触发 / 更新事件 2 中断
MCTM0_CC	可配置 <sup>(2)</sup>	30	46	0x0B8	MCTM0 捕捉 / 比较中断
MCTM1_BRK	可配置 <sup>(2)</sup>	31	47	0x0BC	MCTM1 暂停中断
MCTM1_UP	可配置 <sup>(2)</sup>	32	48	0x0C0	MCTM1 更新中断
MCTM1_TR_UP2	可配置 <sup>(2)</sup>	33	49	0x0C4	MCTM1 触发 / 更新事件 2 中断
MCTM1_CC	可配置 <sup>(2)</sup>	34	50	0x0C8	MCTM1 捕捉 / 比较中断
GPTM0	可配置 <sup>(2)</sup>	35	51	0x0CC	GPTM0 总中断
GPTM1	可配置 <sup>(2)</sup>	36	52	0x0D0	GPTM1 总中断
—	—	37	53	0x0D4	保留
—	—	38	54	0x0D8	保留
—	—	39	55	0x0DC	保留
—	—	40	56	0x0E0	保留
BFTM0	可配置 <sup>(2)</sup>	41	57	0x0E4	BFTM0 总中断
BFTM1	可配置 <sup>(2)</sup>	42	58	0x0E8	BFTM1 总中断
I <sup>2</sup> C0	可配置 <sup>(2)</sup>	43	59	0x0EC	I <sup>2</sup> C0 总中断
I <sup>2</sup> C1	可配置 <sup>(2)</sup>	44	60	0x0F0	I <sup>2</sup> C1 总中断
SPI0	可配置 <sup>(2)</sup>	45	61	0x0F4	SPI0 总中断
SPI1	可配置 <sup>(2)</sup>	46	62	0x0F8	SPI1 总中断
USART0	可配置 <sup>(2)</sup>	47	63	0x0FC	USART0 总中断
USART1	可配置 <sup>(2)</sup>	48	64	0x100	USART1 总中断
UART0	可配置 <sup>(2)</sup>	49	65	0x104	UART0 总中断
UART1	可配置 <sup>(2)</sup>	50	66	0x108	UART1 总中断
SCI	可配置 <sup>(2)</sup>	51	67	0x10C	SCI0 & SCI1 总中断
I <sup>2</sup> S	可配置 <sup>(2)</sup>	52	68	0x110	I <sup>2</sup> S 总中断
USB	可配置 <sup>(2)</sup>	53	69	0x114	USB 总中断
SDIO	可配置 <sup>(2)</sup>	54	70	0x118	SDIO 总中断

异常类型	优先级	中断号	异常号	向量地址	描述
PDMA_CH0	可配置 <sup>(2)</sup>	55	71	0x11C	PDMA 通道 0 总中断
PDMA_CH1	可配置 <sup>(2)</sup>	56	72	0x120	PDMA 通道 1 总中断
PDMA_CH2	可配置 <sup>(2)</sup>	57	73	0x124	PDMA 通道 2 总中断
PDMA_CH3	可配置 <sup>(2)</sup>	58	74	0x128	PDMA 通道 3 总中断
PDMA_CH4	可配置 <sup>(2)</sup>	59	75	0x12C	PDMA 通道 4 总中断
PDMA_CH5	可配置 <sup>(2)</sup>	60	76	0x130	PDMA 通道 5 总中断
PDMA_CH6	可配置 <sup>(2)</sup>	61	77	0x134	PDMA 通道 6 总中断
PDMA_CH7	可配置 <sup>(2)</sup>	62	78	0x138	PDMA 通道 7 总中断
PDMA_CH8	可配置 <sup>(2)</sup>	63	79	0x13C	PDMA 通道 8 总中断
PDMA_CH9	可配置 <sup>(2)</sup>	64	80	0x140	PDMA 通道 9 总中断
PDMA_CH10	可配置 <sup>(2)</sup>	65	81	0x144	PDMA 通道 10 总中断
PDMA_CH11	可配置 <sup>(2)</sup>	66	82	0x148	PDMA 通道 11 总中断
CSIF	可配置 <sup>(2)</sup>	67	83	0x14C	CSIF 总中断
EBI	可配置 <sup>(2)</sup>	68	84	0x150	EBI 总中断
AES	可配置 <sup>(2)</sup>	69	85	0x154	AES 总中断

注：1. 使用 NVIC 系统处理程序优先级寄存器，异常类型的优先级可改变。更多信息请查阅 Arm “Cortex®-M3 技术参考手册” 文件。

2. 使用 NVIC 中断优先级寄存器，中断优先级可改变。更多信息请查阅 Arm “Cortex®-M3 技术参考手册” 文件。

## 特性

- 16 个 Cortex®-M3 系统异常
- 多达 64 个可屏蔽的外设中断
- 16 个可编程优先级 (4 位用于中断优先级设置)
- 非屏蔽中断
- 低延时异常和中断处理
- 向量表重置功能
- 内置简单，24-bit 系统定时器，SysTick
  - 24-bit 向下计数器
  - 自动重载功能
  - 当计数器递减到 0，产生可屏蔽的系统中断
  - HCLK 或 AHB 时钟除以 8 得到 SysTick 的参考时钟源



## 功能描述

### SysTick 校准

NVIC 提供 SysTick 校准值寄存器 SCALIB, 给 RTOS 节拍定时器一个 1 ms 参考时基或用作其它用途。当时钟源来自带有一个频率为 12 MHz (96 MHz 除以 8) 的 SysTick 参考输入时钟 STCLK, 在 SCALIB 寄存器 TENMS 字段有一个固定值 12000, 这是计数器重载值用来表示 1 ms。

## 寄存器列表

下表显示 NVIC 寄存器和复位值。

表 27. NVIC 寄存器列表

寄存器	偏移量	描述	复位值
NVIC 基址 = 0xE000_E000			
ICTR	0x004	中断控制类型寄存器	0x0000_0001
SCTRL	0x010	SysTick 控制和状态寄存器	0x0000_0000
SLOAD	0x014	SysTick 重载值寄存器	不可预知
SVAL	0x018	SysTick 电流值寄存器	不可预知
SCALIB	0x01C	SysTick 校准值寄存器	0x4000_2EE0
ISER0_31	0x100	Irq 0 ~ 31 置位使能寄存器	0x0000_0000
ISER32_63	0x104	Irq 32 ~ 63 置位使能寄存器	0x0000_0000
ISER64_95	0x108	Irq 64 ~ 95 置位使能寄存器	0x0000_0000
ICER0_31	0x180	Irq 0 ~ 31 清除使能寄存器	0x0000_0000
ICER32_63	0x184	Irq 32 ~ 63 清除使能寄存器	0x0000_0000
ICER64_95	0x188	Irq 64 ~ 95 清除使能寄存器	0x0000_0000
ISPR0_31	0x200	Irq 0 ~ 31 置位挂起寄存器	0x0000_0000
ISPR32_63	0x204	Irq 32 ~ 63 置位挂起寄存器	0x0000_0000
ISPR64_95	0x208	Irq 64 ~ 95 置位挂起寄存器	0x0000_0000
ICPR0_31	0x280	Irq 0 ~ 31 清除挂起寄存器	0x0000_0000
ICPR32_63	0x284	Irq 32 ~ 63 清除挂起寄存器	0x0000_0000
ICPR64_95	0x288	Irq 64 ~ 95 清除挂起寄存器	0x0000_0000
IABR0_31	0x300	Irq 0 ~ 31 有效位寄存器	0x0000_0000
IABR32_63	0x304	Irq 32 ~ 63 有效位寄存器	0x0000_0000
IABR64_95	0x308	Irq 64 ~ 95 有效位寄存器	0x0000_0000
IRQ0_3	0x400	Irq 0 ~ 3 优先级寄存器	0x0000_0000
IRQ4_7	0x404	Irq 4 ~ 7 优先级寄存器	0x0000_0000
IRQ8_11	0x408	Irq 8 ~ 11 优先级寄存器	0x0000_0000
IRQ12_15	0x40C	Irq 12 ~ 15 优先级寄存器	0x0000_0000
IRQ16_19	0x410	Irq 16 ~ 19 优先级寄存器	0x0000_0000
IRQ20_23	0x414	Irq 20 ~ 23 优先级寄存器	0x0000_0000
IRQ24_27	0x418	Irq 24 ~ 27 优先级寄存器	0x0000_0000
IRQ28_31	0x41C	Irq 28 ~ 31 优先级寄存器	0x0000_0000
IRQ32_35	0x420	Irq 32 ~ 35 优先级寄存器	0x0000_0000
IRQ36_39	0x424	Irq 36 ~ 39 优先级寄存器	0x0000_0000
IRQ40_43	0x428	Irq 40 ~ 43 优先级寄存器	0x0000_0000
IRQ44_47	0x42C	Irq 44 ~ 47 优先级寄存器	0x0000_0000

寄存器	偏移量	描述	复位值
IRQ48_51	0x430	Irq 48 ~ 51 优先级寄存器	0x0000_0000
IRQ52_55	0x434	Irq 52 ~ 55 优先级寄存器	0x0000_0000
IRQ56_59	0x438	Irq 56 ~ 59 优先级寄存器	0x0000_0000
IRQ60_63	0x43C	Irq 60 ~ 63 优先级寄存器	0x0000_0000
IRQ64_67	0x440	Irq 64 ~ 67 优先级寄存器	0x0000_0000
ICSR	0xD04	中断控制状态寄存	0x0000_0000
VTOR	0xD08	向量表偏移寄存器	0x0000_0000
AIRCR	0xD0C	应用中断 / 复位控制寄存器	0xFA05_0000
SCR	0xD10	系统控制寄存器	0x0000_0000
CCR	0xD14	配置控制寄存器	0x0000_0200
SHPR4_7	0xD18	系统处理程序 4 ~ 7 优先级寄存器	0x0000_0000
SHPR8_11	0xD1C	系统处理程序 8 ~ 11 优先级寄存器	0x0000_0000
SHPR12_15	0xD20	系统处理程序 12 ~ 15 优先级寄存器	0x0000_0000
SHCSR	0xD24	系统处理程序的控制和状态寄存器	0x0000_0000
CFSR	0xD28	可配置的故障状态寄存器	0x0000_0000
HFSR	0xD2C	硬件故障状态寄存器	0x0000_0000
DFSR	0xD30	调试故障状态寄存器	0x0000_0000
MMFAR	0xD34	分配信息管理地址寄存器	不可预知
BFAR	0xD38	总线故障地址寄存器	不可预知
AFSR	0xD3C	辅助异常状态寄存器	0x0000_0000
STIR	0xF00	软件触发中断寄存器	0x0000_0000

注：更多以上寄存器的详细描述，请参阅 Arm 文档“Cortex®-M3 技术参考手册”。

# 11 外部中断 / 事件控制器 (EXTI)

## 简介

外部中断 / 事件控制器 (EXTI) 包括 16 个边沿检测器, 都可以分别产生唤醒事件或中断请求。在中断模式中, 外部中断有 5 种触发类型, 分别为低电平、高电平、下降沿、上升沿和双沿, 通过 EXTICFGRn (n = 0 ~ 15) 寄存器中 SRCnTYPE 字段选择。在唤醒事件模式, 唤醒事件的极性可以通过 EXTIWAKUPPOLR 寄存器中 EXTInWPOL (n = 0 ~ 15) 字段设置。如果 EXTIWAKUPCR 寄存器中 EVWUPIEN 位置位, 当相关的唤醒事件发生时及相应的 EXTI 唤醒使能位置位时, 将产生 EVWUP 中断。每个 EXTI 输入线也可以个别地被屏蔽。

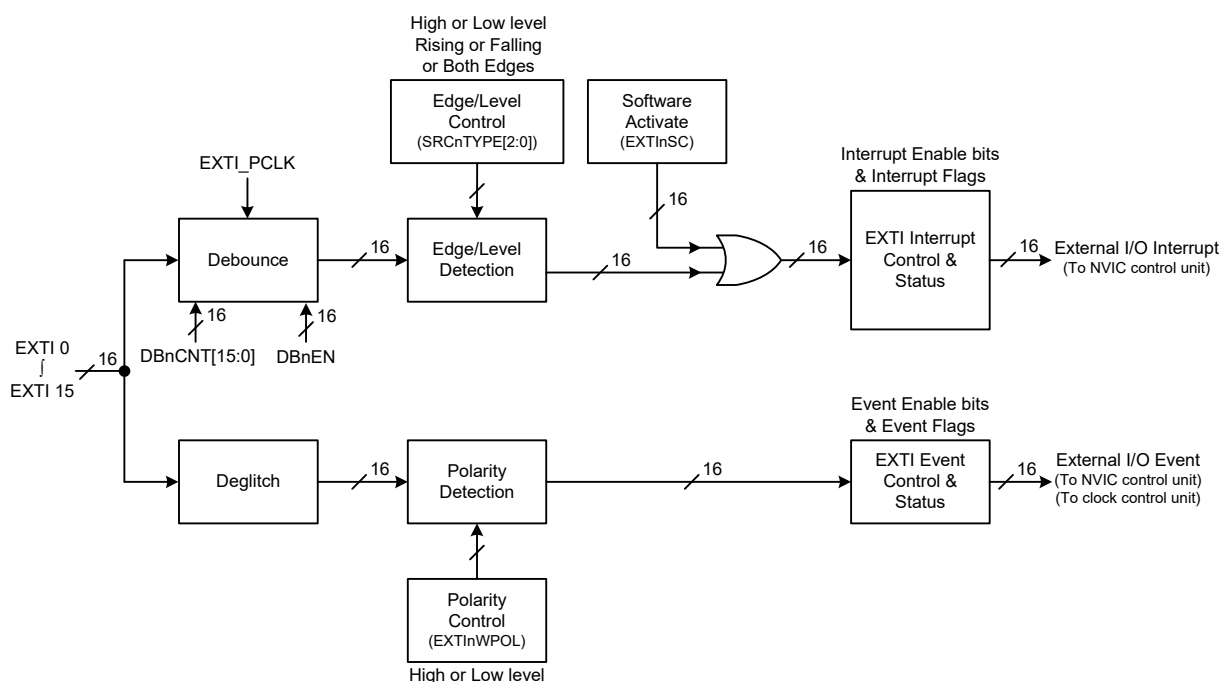


图 25. EXTI 方框图

## 特性

- 多达 16 个可配置触发源和类型的 EXTI 输入线
  - 所有 GPIO 引脚可以选择作为 EXTI 触发源
  - 触发源类型包括高电平、低电平、下降沿、上升沿或双沿
- 每个 EXTI 输入线有独立的中断使能, 唤醒使能及状态位
- 每个 EXTI 输入线有软件中断触发模式
- 内置去干扰滤波器用于封锁短脉冲

## 功能描述

### 唤醒事件管理

为了从省电模式中唤醒系统，EXTI 控制器提供可以监控外部事件的功能，并把它们送到 CPU 内核和时钟控制单元，CKCU。这些外部事件包括 EXTI 事件、低电压检测、WAKEUP 输入引脚、比较器、USB 和 RTC 唤醒功能。通过在相应的外设内配置唤醒事件的使能位，当相应的唤醒事件发生时，通过 EXTI 控制器，唤醒信号将被发送到 CPU 和 CKCU。此外，可以通过软件设置 EXTIWAKUPCR 寄存器 EVWUPIEN 位使能事件唤醒中断功能，当唤醒事件发生时，EXTI 控制器将确立中断。

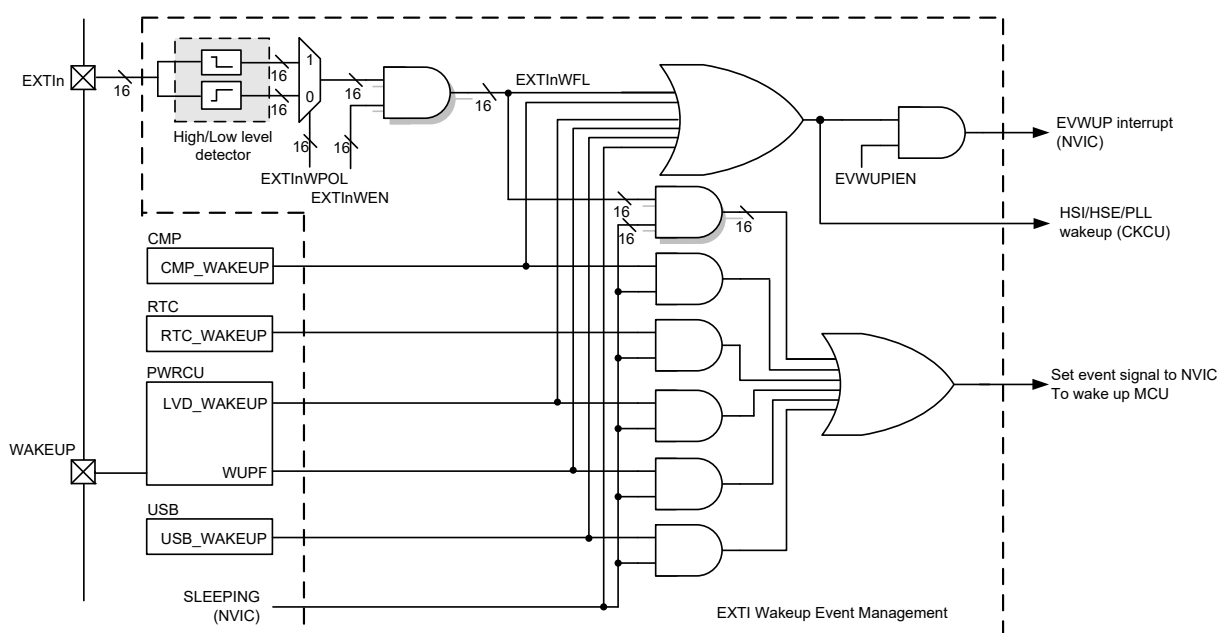


图 26. EXTI 唤醒事件管理

## 外部中断 / 事件线配置

所有 GPIO 引脚通过配置 AFIO ESSRn (n = 0 ~ 1) 寄存器 EXTInPIN[3:0] 字段选择哪个引脚作为 EXTI 触发源来触发中断或事件。更多信息请参考 AFIO 部分。

## 中断和去抖

应用软件可以设置 EXTIn 中断配置寄存器 EXTICFGRn (n = 0 ~ 15) 的 DBnEN 位, 使能相应引脚的去抖功能, 配置 EXTICFGRn 寄存器中 DBnCNT 字段, 以便为特定应用选择一个合适的去抖时间。但中断信号因去抖功能被延迟。当单片机通过外部中断从省电模式中唤醒时, EXTI 唤醒标志位产生中断请求。单片机被唤醒且时钟已经恢复后, 被 EXTI 引脚触发的 EXTI 唤醒标志位必须读出, 然后用应用程序清零。所附图表显示 EXTI 输入信号与 EXTI 中断 / 事件请求信号之间的关系。

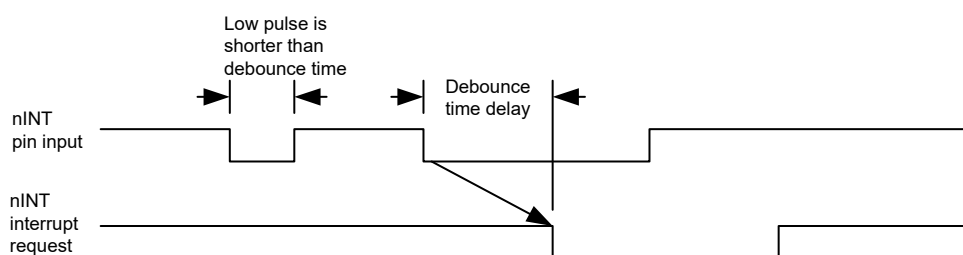


图 27. EXTI 中断去抖功能

## 寄存器列表

下表显示 EXTI 寄存器和复位值。

表 28. EXTI 寄存器列表

寄存器	偏移量	描述	复位值
EXTICFGR0	0x000	EXTI 中断 0 配置寄存器	0x0000_0000
EXTICFGR1	0x004	EXTI 中断 1 配置寄存器	0x0000_0000
EXTICFGR2	0x008	EXTI 中断 2 配置寄存器	0x0000_0000
EXTICFGR3	0x00C	EXTI 中断 3 配置寄存器	0x0000_0000
EXTICFGR4	0x010	EXTI 中断 4 配置寄存器	0x0000_0000
EXTICFGR5	0x014	EXTI 中断 5 配置寄存器	0x0000_0000
EXTICFGR6	0x018	EXTI 中断 6 配置寄存器	0x0000_0000
EXTICFGR7	0x01C	EXTI 中断 7 配置寄存器	0x0000_0000
EXTICFGR8	0x020	EXTI 中断 8 配置寄存器	0x0000_0000
EXTICFGR9	0x024	EXTI 中断 9 配置寄存器	0x0000_0000
EXTICFGR10	0x028	EXTI 中断 10 配置寄存器	0x0000_0000
EXTICFGR11	0x02C	EXTI 中断 11 配置寄存器	0x0000_0000
EXTICFGR12	0x030	EXTI 中断 12 配置寄存器	0x0000_0000
EXTICFGR13	0x034	EXTI 中断 13 配置寄存器	0x0000_0000
EXTICFGR14	0x038	EXTI 中断 14 配置寄存器	0x0000_0000
EXTICFGR15	0x03C	EXTI 中断 15 配置寄存器	0x0000_0000
EXTICR	0x040	EXTI 中断控制寄存器	0x0000_0000
EXTIEDGEFLGR	0x044	EXTI 中断边沿标志寄存器	0x0000_0000
EXTIEDGESR	0x048	EXTI 中断边沿状态寄存器	0x0000_0000
EXTISSCR	0x04C	EXTI 中断软件设置命令寄存器	0x0000_0000
EXTIWAKUPCR	0x050	EXTI 中断唤醒控制寄存器	0x0000_0000
EXTIWAKUPPOLR	0x054	EXTI 中断唤醒极性寄存器	0x0000_0000
EXTIWAKUPFLG	0x058	EXTI 中断唤醒标志寄存器	0x0000_0000

寄存器描述

EXTI 中断 n 配置寄存器 – EXTICFGRn, n = 0 ~ 15

该寄存器用来定义去抖功能和选择触发类型。

偏移量: 0x000 (0) ~ 0x03C (15)

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	DBnEN		SRCnTYPE				保留位	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	DBnCNT							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	DBnCNT							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述																								
[31]	DBnEN	EXTIn 去抖电路使能位 (n = 0 ~ 15) 0: 除能去抖电路 1: 使能去抖电路																								
[30:28]	SRCnTYPE	EXTIn 中断源触发类型 (n = 0 ~ 15) <table><tr><th colspan="3">SRCnTYPE[2:0]</th><th>中断源类型</th></tr><tr><td>0</td><td>0</td><td>0</td><td>低电平有效</td></tr><tr><td>0</td><td>0</td><td>1</td><td>高电平有效</td></tr><tr><td>0</td><td>1</td><td>0</td><td>下降沿触发</td></tr><tr><td>0</td><td>1</td><td>1</td><td>上升沿触发</td></tr><tr><td>1</td><td>X</td><td>X</td><td>双沿触发</td></tr></table>	SRCnTYPE[2:0]			中断源类型	0	0	0	低电平有效	0	0	1	高电平有效	0	1	0	下降沿触发	0	1	1	上升沿触发	1	X	X	双沿触发
SRCnTYPE[2:0]			中断源类型																							
0	0	0	低电平有效																							
0	0	1	高电平有效																							
0	1	0	下降沿触发																							
0	1	1	上升沿触发																							
1	X	X	双沿触发																							
[15:0]	DBnCNT	EXTIn 去抖计数器 (n = 0 ~ 15) 去抖时间根据 DBnCNT × APB 时钟 (EXTI_PCLK) 周期计算，并应足够长来对输入信号产生影响。																								

**EXTI 中断控制寄存器 – EXTICR**

该寄存器用来控制 EXTI 中断。

偏移量： 0x040  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	EXTI15EN	EXTI14EN	EXTI13EN	EXTI12EN	EXTI11EN	EXTI10EN	EXTI9EN	EXTI8EN
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	EXTI7EN	EXTI6EN	EXTI5EN	EXTI4EN	EXTI3EN	EXTI2EN	EXTI1EN	EXTI0EN
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	EXTInEN	EXTIn 中断使能位 (n = 0 ~ 15) 0: 除能 EXTI 输入线 n 中断 1: 使能 EXTI 输入线 n 中断



**EXTI 中断边沿标志寄存器 – EXTIEDGEFLGR**

该寄存器用来说明是否已检测到 EXTI 边沿。

偏移量： 0x044  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	EXTI15EDF	EXTI14EDF	EXTI13EDF	EXTI12EDF	EXTI11EDF	EXTI10EDF	EXTI9EDF	EXTI8EDF
类型 / 复位	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0
	7	6	5	4	3	2	1	0
	EXTI7EDF	EXTI6EDF	EXTI5EDF	EXTI4EDF	EXTI3EDF	EXTI2EDF	EXTI1EDF	EXTI0EDF
类型 / 复位	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

位	字段	描述
[15:0]	EXTInEDF	EXTIn 边沿检测标志位 (n = 0 ~ 15) 0: 没有检测到边沿 1: 检测到上升沿或下降沿 当相应的 EXTI 引脚检测到上升沿或下降沿，此位被硬件置位。通过写 1 来清零。

**EXTI 中断边沿状态寄存器 – EXTIEDGESR**

该寄存器用来说明检测到的 EXTI 边沿的极性。

偏移量： 0x048  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	EXTI15EDS	EXTI14EDS	EXTI13EDS	EXTI12EDS	EXTI11EDS	EXTI10EDS	EXTI9EDS	EXTI8EDS
类型 / 复位	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0
	7	6	5	4	3	2	1	0
	EXTI7EDS	EXTI6EDS	EXTI5EDS	EXTI4EDS	EXTI3EDS	EXTI2EDS	EXTI1EDS	EXTI0EDS
类型 / 复位	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

位	字段	描述
[15:0]	EXTInEDS	EXTIn 边沿检测状态位 (n = 0 ~ 15) 0: 检测到下降沿 1: 检测到上升沿 通过写 1 来清零。

**EXTI 中断软件设置命令寄存器 – EXTISSCR**

该寄存器用来激活 EXTI 中断。

偏移量： 0x04C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	EXTI15SC	EXTI14SC	EXTI13SC	EXTI12SC	EXTI11SC	EXTI10SC	EXTI9SC	EXTI8SC
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	EXTI7SC	EXTI6SC	EXTI5SC	EXTI4SC	EXTI3SC	EXTI2SC	EXTI1SC	EXTI0SC
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	EXTInSC	EXTIn 软件设置命令 (n = 0 ~ 15) 0: 停用相关 EXTI 中断 1: 激活相关 EXTI 中断

**EXTI 中断唤醒控制寄存器 – EXTIWAKUPCR**

该寄存器用来控制 EXTI 中断和唤醒功能。

偏移量： 0x050  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	EVWUPIEN	保留位						
类型 / 复位	RW	0						
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	EXTI15WEN	EXTI14WEN	EXTI13WEN	EXTI12WEN	EXTI11WEN	EXTI10WEN	EXTI9WEN	EXTI8WEN
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	EXTI7WEN	EXTI6WEN	EXTI5WEN	EXTI4WEN	EXTI3WEN	EXTI2WEN	EXTI1WEN	EXTI0WEN
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31]	EVWUPIEN	EXTI 事件唤醒中断使能位 0: 除能 EVWUP 中断 1: 使能 EVWUP 中断
[15:0]	EXTInWEN	EXTIn 唤醒使能位 (n = 0 ~ 15) 0: 除能省电模式唤醒 1: 使能省电模式唤醒

**EXTI 中断唤醒极性寄存器 – EXTIWAKUPPOLR**

该寄存器用来选择 EXTI 引脚中断唤醒极性。

偏移量： 0x054  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	EXTI15WPOL	EXTI14WPOL	EXTI13WPOL	EXTI12WPOL	EXTI11WPOL	EXTI10WPOL	EXTI9WPOL	EXTI8WPOL
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	EXTI7WPOL	EXTI6WPOL	EXTI5WPOL	EXTI4WPOL	EXTI3WPOL	EXTI2WPOL	EXTI1WPOL	EXTI0WPOL
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	EXTInWPOL	EXTIn 唤醒极性 (n = 0 ~ 15) 0: EXTIn 高电平时唤醒 1: EXTIn 低电平时唤醒

**EXTI 中断唤醒标志寄存器 – EXTIWAKUPFLG**

该寄存器是 EXTI 中断唤醒标志寄存器。

偏移量： 0x058  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	EXTI15WFL	EXTI14WFL	EXTI13WFL	EXTI12WFL	EXTI11WFL	EXTI10WFL	EXTI9WFL	EXTI8WFL
类型 / 复位	WC	0	WC	0	WC	0	WC	0
	7	6	5	4	3	2	1	0
	EXTI7WFL	EXTI6WFL	EXTI5WFL	EXTI4WFL	EXTI3WFL	EXTI2WFL	EXTI1WFL	EXTI0WFL
类型 / 复位	WC	0	WC	0	WC	0	WC	0

位	字段	描述
[15:0]	EXTInWFL	EXTIn 唤醒标志位 (n = 0 ~ 15) 0: 没发生唤醒 1: 系统被 EXTIn 唤醒 通过写 1 来清零。

# 12 模数转换器 (ADC)

## 模数转换器

单片机内建一个 12-bit 多通道 A/D 转换器，总共有 18 个通道，包括 16 个提供模拟信号的外部通道和 2 个内部通道。模拟看门狗功能用来确保 A/D 转换器的输入电压保持在一个特定的阈值。当输入电压高于或低于设定的阈值，将产生中断。有三种转换模式用来把模拟信号转换成数字数据。A/D 转换器可以工作在单次转换，连续和非连续转换模式。提供左对齐或右对齐的 16-bit 数据寄存器来储存转换后的数据。

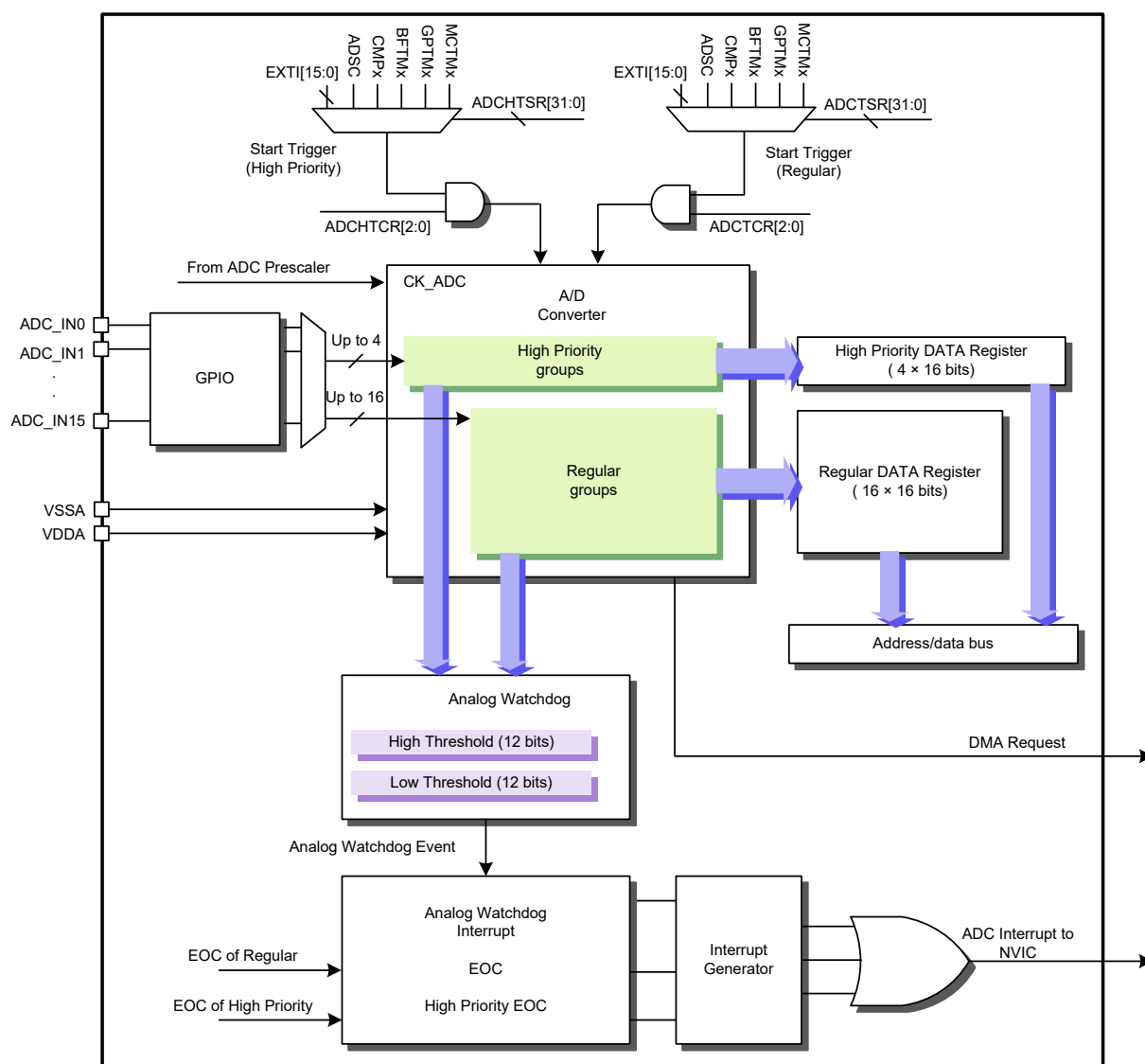


图 28. ADC 方框图

## 特性

- 12-bit SAR ADC 装置
- 高达 1 MSPS 转换速率
  - 84 MHz 下 1  $\mu$ s; 96 MHz 下 1.17  $\mu$ s
- 16 个外部模拟输入通道
- 2 个内部模拟输入通道用于参考电压检测
- 每个通道单独可编程的采样时间
- 3 种转换模式
  - 单次转换模式
  - 连续转换模式
  - 非连续转换模式
- 两级转换优先级
  - 常规 – 可被高优先级转换打断
  - 高优先级
- 多达 16 个专用的序列发生器和数据寄存器用于常规转换
- 多达 4 个专用的序列发生器和数据寄存器用于高优先级转换
- 数据对齐调节和偏移量消除功能
  - 右 / 左对齐
  - 有符号 / 无符号
  - 每个通道有 16 个偏移量寄存器
- 用于预定义的电压范围监控的模拟看门狗
  - 下限 / 上限阈值寄存器
  - 中断产生器
- 多种触发源用于常规或高优先级的转换模式
  - 软件触发
  - EXTI – 外部中断输入引脚
  - GPTM0 / GPTM1 触发
  - MCTM0 / MCTM1 触发
  - BFTM0 / BFTM1 触发
  - CMP0 / CMP1 触发
- 多种中断产生
  - 单次转换结束
  - 子组转换结束
  - 周期转换结束
  - 模拟看门狗
  - 数据寄存器覆盖
- 转换结束发生 PDMA 请求



## 功能描述

### ADC 时钟设置

与 AHB 时钟 HCLK 同步的时钟控制器提供 ADC 时钟 (CK\_ADC)。更多信息请参考时钟控制单元的章节。注意, ADC 需要至少两个 ADC 时钟周期在上电和掉电情况下 (ADCEN bit = “0”) 相互切换。

### 常规和高优先级通道选择

A/D 转换器支持 16 个多路复用通道, 在转换后把转换结果分为常规组和高优先级组。一个常规转换组可以安排一个转换序列, 此序列安排在一个特定的序列长度从 1 到 16 的通道。例如, 可以在通道顺序为 CH2、CH4、CH7、CH5、CH6、CH3、CH1 和 CH0 的序列里逐一进行转换。

一个常规组可包含多达 16 次转换。常规组转换通道可通过 ADCLST0 ~ ADCLST2 寄存器选择。转换序列的总长度通过 ADCCONV 寄存器 ADSEQL[2:0] 位设置。

高优先级组可包含最多 4 次转换。序列长度和所选的高优先级组通道可由 ADCHLST 寄存器设置。高优先级组转换序列的总长度通过 ADCHCONV 寄存器 ADHSEQL[1:0] 位设置。

在转换过程中, 修改 ADCCONV 或 ADCHCONV 寄存器将复位当前的转换。之后, 需要一个新的启动脉冲用于重新启动一个新的转换。

### 转换模式

A/D 转换器有三种工作转换模式。这三种转换模式分别是单次转换模式、连续转换模式和非连续转换模式。详细信息稍后描述。

#### 单次转换模式

在单次转换模式中, 当 A/D 转换事件发生时, A/D 转换器在 A/D 转换列表寄存器 ADCLSTn 或 ADCHLST 寄存器指定的通道中按照特定的顺序执行转换周期。当 ADCCONV 寄存器中的 A/D 转换模式字段的 ADMODE[1:0] 设置为 0x0 或 ADHMODE[1:0] 设置为 0x0, A/D 转换器将运行在单次转换模式。软件触发、比较器转变事件、外部 EXTI 事件或 TM 事件都可启动单次转换, 由触发控制寄存器 ADCTCR 或 ADCHTCR 和触发源寄存器 ADCTSR 或 ADCHTSR 设置。

#### 常规转换后:

- 转换后的数据将被储存在 16-bit 寄存器 ADCDRy (y = 0 ~ 15)。
- 单次采样转换完成时, ADCIRAW 寄存器中的 ADC 单次采样转换结束事件原始状态标志位 ADIRAWS 将被置位。
- 单次采样转换完成后, 如果 ADCIER 寄存器中的 ADIES 位使能, 将产生中断。
- 一个常规组的转换周期结束后, 如果 ADCIER 寄存器中 ADIEC 位使能, 将产生中断。

#### 高优先级转换后:

- 转换后的数据将被储存在 16-bit 寄存器 ADCHDRy (y = 0 ~ 3)。
- 当转换完成, ADCIRAW 寄存器中的 ADC 高优先级单次采样转换事件结束原始状态标志位 ADIRAWHS 将被置位。
- 一个高优先级单次采样转换结束, 如果 ADCIER 寄存器中 ADIEHS 位使能, 将产生中断。

- 一个高优先级组的转换周期结束后，如果 ADCIER 寄存器中 ADIEHC 位使能，将产生中断

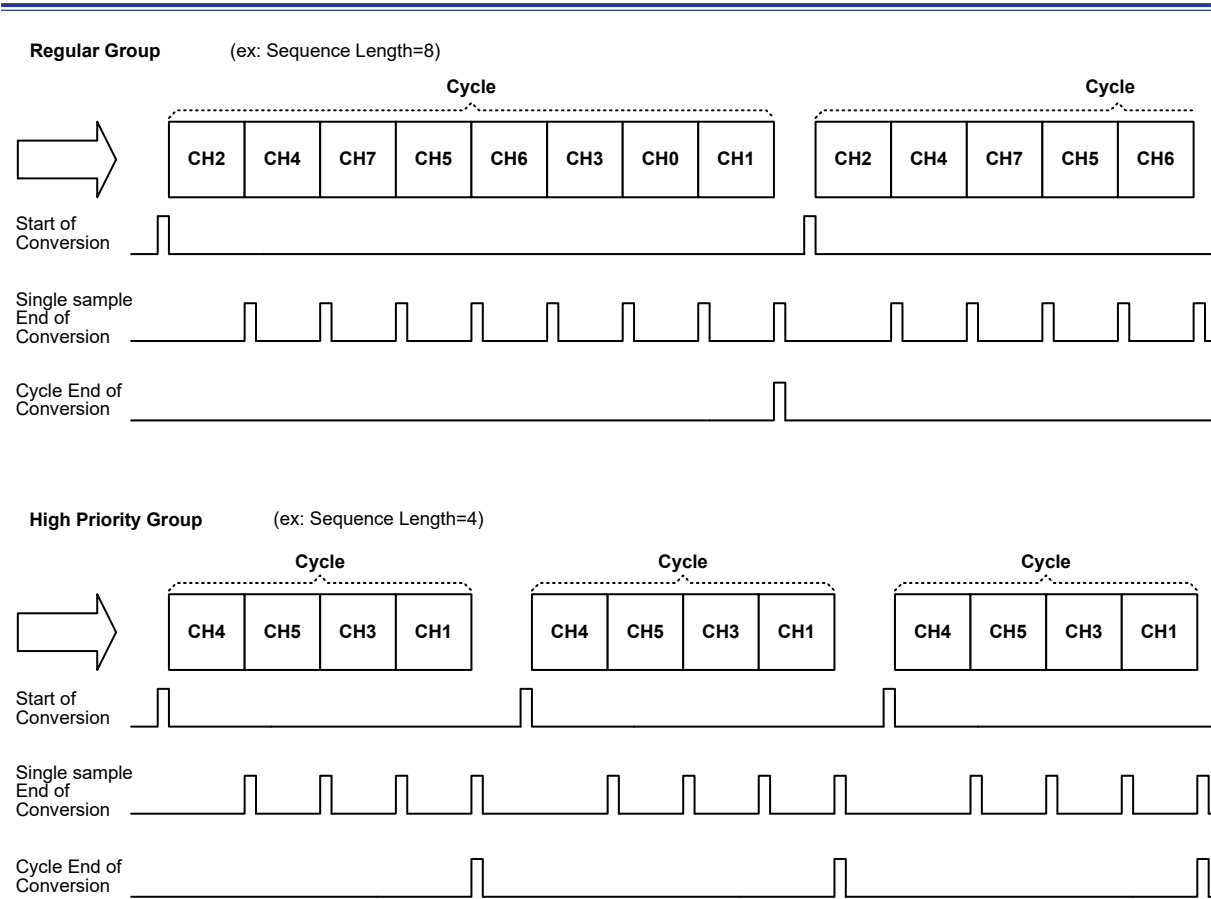


图 29. 单次转换模式

### 连续转换模式

在连续转换模式下，一个通道组转换完成后，无需额外的 A/D 转换启动触发信号，重复转换周期将自动重新启动。当 A/D 转换模式字段的 ADMODE[1:0] 设置为 0x2 或 ADHMODE[1:0] 设置为 0x2，A/D 转换器将运行在连续转换模式。软件触发，比较器转变事件、外部 EXTI 事件或 TM 事件都可启动连续转换，由触发控制寄存器 ADCTCR 或 ADCHTCR 和触发源寄存器 ADCTSR 或 ADCHTSR 决定。

#### 转换后：

- 转换后的数据将被储存在 16-bit ADCDRy (y = 0 ~ 15) 或 ADCHDRy (y = 0 ~ 3) 寄存器。
- 当转换周期结束时，ADCIRAW 寄存器中的 ADC 组转换事件周期结束原始状态标志位 ADIRAWC 将被置位。
- 组转换周期结束后，如果 ADCIER 寄存器中的 ADIEC 或 ADIHEC 位使能，将产生中断。

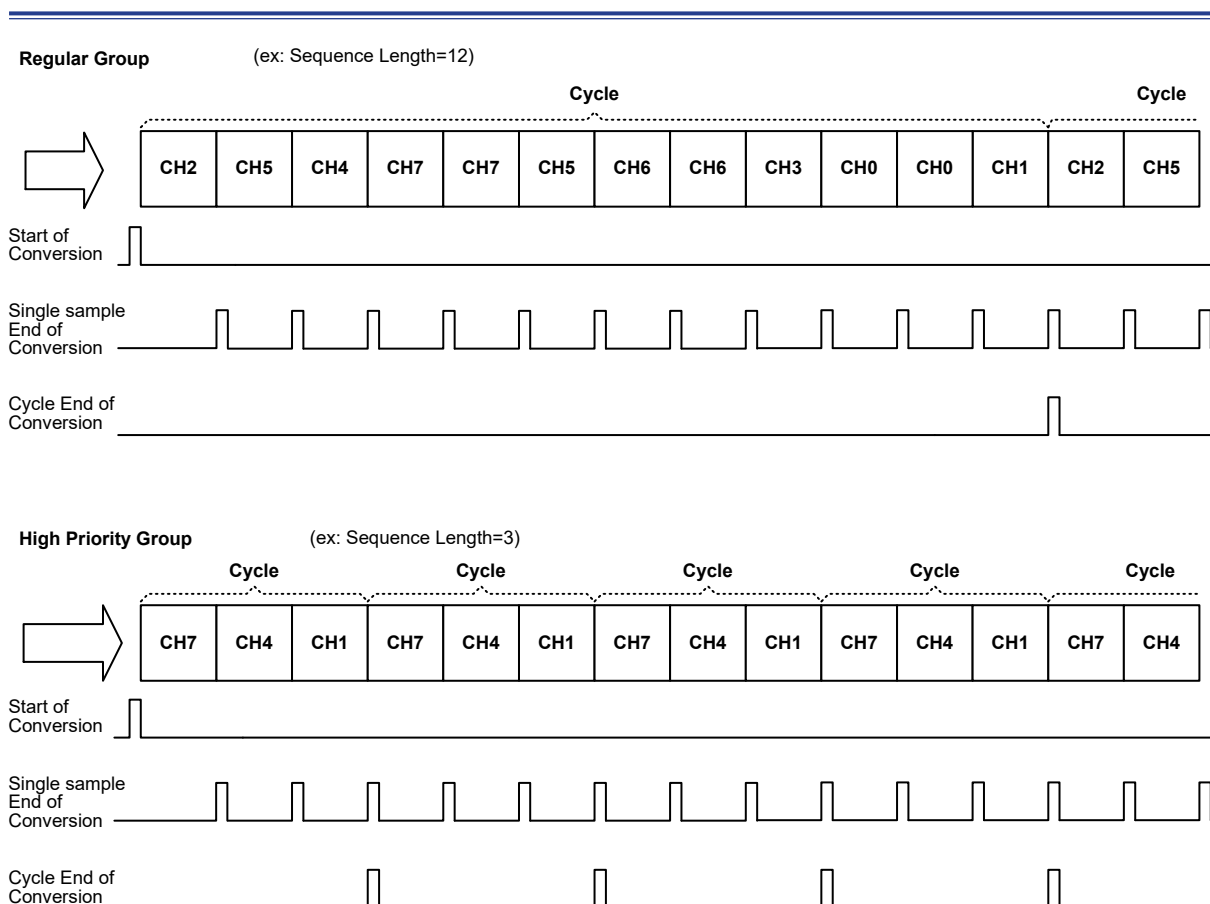


图 30. 连续转换模式

## 非连续转换模式

### 常规组

ADCCONV 寄存器中 A/D 转换模式位字段 ADMODE[1:0] 设置为 0x3, A/D 转换器将运行在常规组的非连续转换模式。要转换的组最多可以有 16 个通道, 可以以一个特定的顺序排列, 通过 ADCLSTn 寄存器配置, 其中 n 的范围从 0 到 3。在这种模式下, 每次触发事件发生时, 以短序列形式转换常规组数据, 称为 A/D 常规转换子组。子组长度由 ADSUBL[2:0] 字段定义。软件触发、比较器转变事件、外部 EXTI 事件或 TM 事件都可启动非连续转换, 由触发控制寄存器 ADCTCR 和触发源寄存器 ADCTSR 决定。

在非连续转换模式, A/D 转换器开始下一组的 n 个转换, 数字 n 是 ADSUBL 字段定义的子组长度。当触发事件发生时, 以一个特定序列转换的通道由 ADCLSTn 寄存器定义。经过 n 次转换完成, ADCIRAW 寄存器中子组 EOC 中断原始标志位 ADIRAWG 将被置位。A/D 转换器现在将无法继续执行后面的 n 次转换, 直到下一个触发事件发生。所有的常规组通道 (总数由 ADCCONV 寄存器 ADSEQL [2:0] 位定义) 完成它们的转换后, 转换周期结束, 此时, ADCIRAW 寄存器中周期 EOC 中断的原始标志位 ADIRAWC 将被置位。如果所有子组通道都已被转换后, 一个完整的转换周期已完成, 即一个新触发事件发生, 转换将从第一子组重新开始。

例如:

A/D 转换子组的长度 = 3 (ADSUBL = 2) 和序列的长度 = 8 (ADSEQL = 7), 被转换的通道 = 2、4、7、5、6、3、0 和 1 – ADCLSTn 寄存器中定义的转换顺序。

- 触发 1: 被转换的子组通道是 CH2、CH4 和 CH7, 子组 EOC 后 ADIRAWG 标志位被置位。
- 触发 2: 被转换的子组通道是 CH5、CH6 和 CH3, 子组 EOC 后 ADIRAWG 标志位被置位。
- 触发 3: 被转换的子组通道是 CH0 和 CH1, 子组 EOC 后 ADIRAWG 标志位被置位。一个转换周期结束时 (EOC) 中断原标志位 ADIRAWC 也被置位。
- 触发 4: 被转换的子组通道是 CH2、CH4 和 CH7, ADIRAWG 标志位被置位 – 从头开始重新启动转换序列。

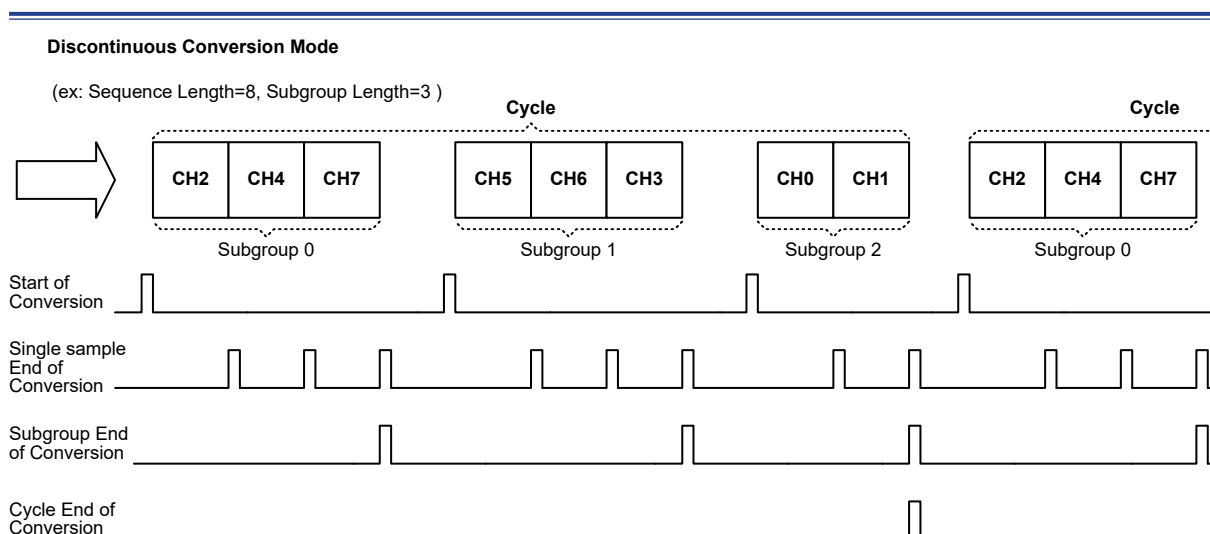


图 31. 常规组非连续转换模式

高优先级组

ADCHCONV 寄存器中 A/D 高优先级转换模式位字段 ADHMODE[1:0] 设置为 0x3, A/D 转换器将运行在高优先级组的非连续转换模式。要转换的高优先级组最多可以有 4 个通道, 可以以一个特定的顺序排列, 通过 ADCHLST 寄存器配置。在这种模式下, 每次触发事件发生时, 以短序列形式转换高优先级组数据, 称为 A/D 高优先级转换子组。ADHSUBL[1:0] 字段定义高优先级子组的长度。在非连续转换模式下, 软件触发, 比较器转变事件、外部 EXTI 事件或 TM 事件都可启动高优先级组 A/D 转换, 由高优先级触发控制寄存器 ADCHTCR 和触发源寄存器 ADCHTSR 决定。

在非连续转换模式, 当触发事件发生时, A/D 转换器开始下一组的 n 个转换, 数字 n 是 ADHSUBL 字段定义的子组长度。以一个特定顺序转换的通道由 ADCHLST 寄存器指定。经过 n 次转换完成, ADCIRAW 寄存器中高优先级子组 EOC 中断的原始标志位 ADIRAWHG 将被置位。A/D 转换器将不会继续执行后面的 n 次转换, 直到下一个触发事件发生。所有的高优先级组通道 (总数由 ADCHCONV 寄存器 ADHSEQL[1:0] 位定义) 完成它们的转换后, 转换周期结束, 此时, ADCIRAW 寄存器中高优先级周期 EOC 中断的原始标志位 ADIRAWHC 将被置位。如果所有子组通道都已被转换后, 一个完整的转换周期已完成, 即一个新触发事件发生, 转换将从第一子组重新开始。

例如:

A/D 转换子组的长度 = 2 (ADHSUBL = 1) 和序列的长度 = 3 (ADHSEQL = 2), 被转换的通道 = 4、7 和 1 – 特定的转换顺序定义在 ADCHLST 寄存器。

- 触发 1: 被转换的子组通道是 CH4 和 CH7, 子组 EOC 后, ADIRAWHG 标志位被置位。
- 触发 2: 被转换的子组通道是 CH1, 子组 EOC 后, ADIRAWHG 标志位被置位。一个转换周期结束时 (EOC) 中断原标志位 ADIRAWHC 也被置位。
- 触发 5: 被转换的子组通道是 CH4 和 CH7, ADIRAWHG 标志位在子组 ECO 后被置位 – 从头开始重新启动转换序列

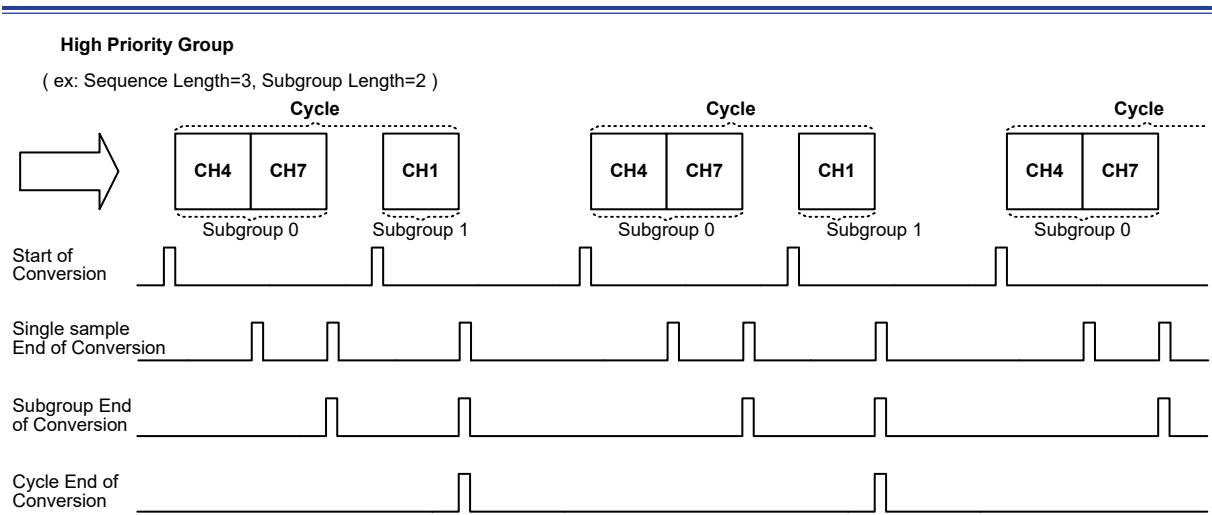


图 32. 高优先级组非连续转换模式

## 外部事件启动转换

A/D 转换可以通过一个软件触发、比较器转变事件、通用定时器模块 (GPTM) 事件, 马达控制定时器模块 (MCTM) 事件, 基本功能定时器模块 (BFTM) 事件或外部触发启动。每个触发源可以通过设置 ADCTCR 或 ADCHTCR 寄存器中相应的使能控制位启用, 然后通过配置 ADCTSR 和 ADCHTSR 寄存器相关的选择位选择从而开始一个组通道转换。

当 ADCTCR 或 ADCHTCR 寄存器中软件触发使能位 ADSW 或 ADHSW 置为 1 时, 可通过设置用于常规组通道的 ADCTSR 寄存器或高优先级组的 ADCHTSR 的软件触发位 ADSC 或 ADHSC 来启动 A/D 转换器的转换。A/D 转换开始转换模拟数据后, 相应的触发位 ADSC 或 ADHSC 将自动清零。

TM 事件也可以触发常规组或高优先级组通道转换开始 A/D 转换。TM 事件包括一个 GPTM 或 MCTM 的主触发输出 MTO, 四个 GPTM 或 MCTM 通道输出 CH0 ~ CH3 和一个 BFTM 触发输出。如果相应的 TM 触发使能位置为 1, 触发输出或 TM 通道事件已通过相应的 TM 事件选择位选择, 当一个选定的触发事件的上升沿发生时, A/D 转换器启动转换。

除了内部触发源, 外部触发事件也可以触发 A/D 开始转换。外部触发事件是来自外部引脚 EXTIn。如果外部触发使能位 ADEXTI 或 ADHEXTI 设置为 1, 并通过配置 ADCTSR 寄存器中的 ADEXTIS 或 ADHEXTIS 字段选择相应 EXTI 输入线, 定义的 EXTI 引脚的有效边沿发生时, A/D 转换器将启动一次转换。

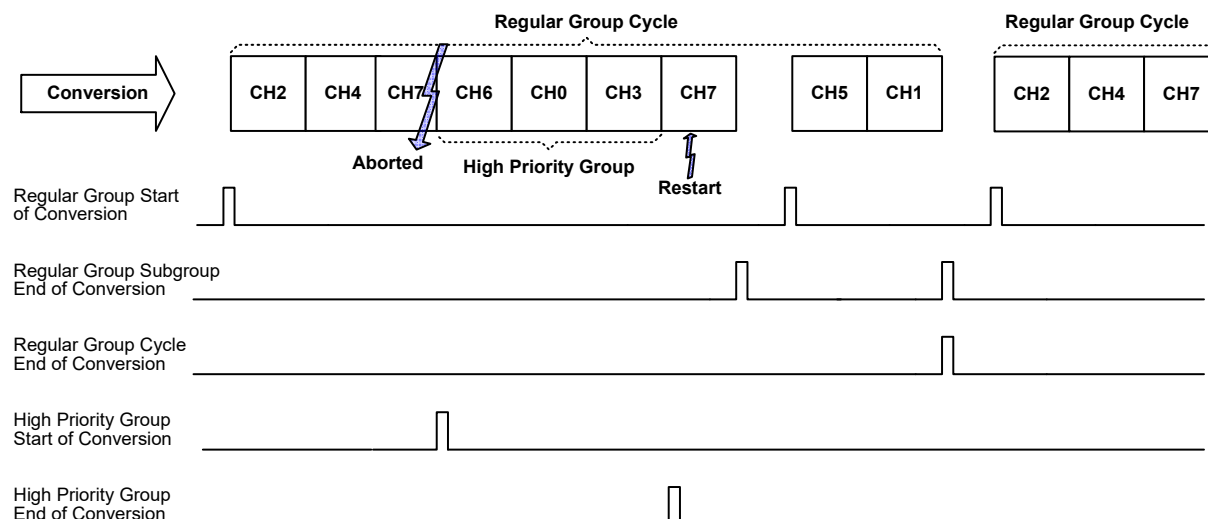
## 高优先级组管理

高优先级通道比常规 A/D 转换通道具有更高优先级。如果在常规转换进行中, 一个高优先级通道触发事件发生, 那么当前常规通道转换将中止, 将开始高优先级通道转换。

高优先级启动触发事件发生后, 要转换的高优先级通道长度取决于高优先级组的转换模式。当高优先级启动触发事件发生, 单次转换模式下, 要转换的高优先级通道长度由 ADHSEQL 字段定义。非连续转换模式下, 要转换的长度是由 ADHSUBL 字段定义的高优先级子组长度。如果高优先级组工作在连续转换模式, 在高优先级启动触发事件后, 高优先级转换将一直会重复转换在高优先级组中的每个通道, 直到有高优先级转换模式发生改变。

当高优先级转换完成, 常规组转换将重新从中止的通道地址开始。注意, 当高优先级启动触发事件发生时, 无论常规组工作在什么转换模式, 高优先级转换将始终会中断当前的常规组转换。

- ◆ **Regular Group** : Sequence Length=5, Subgroup Length=3, discontinuous conversion mode
- ◆ **High Priority Group** : Sequence Length=3, one shot conversion mode



- ◆ **Regular Group** : Sequence Length=5, Subgroup Length=3, discontinuous conversion mode
- ◆ **High Priority Group** : Sequence Length=3, Subgroup Length=2, discontinuous conversion mode

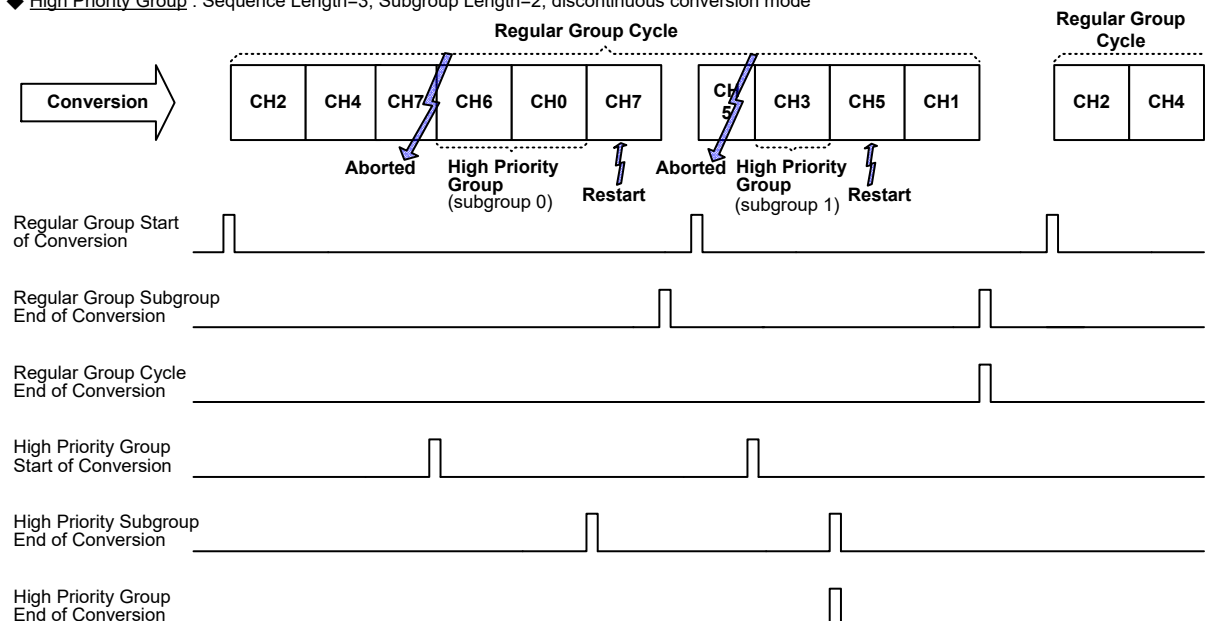


图 33. 高优先级组管理



采样时间设定

每个转换通道可通过输入电压源的阻值变化来编程不同的采样时间。采样时间需足够长可以使输入电压源将内部采样保持电容充电至输入电压电平。每个转换通道都可以用不同的采样时间采样，通过修改 ADCSTRn (n = 0 ~ 15) 寄存器中 ADSTn[7:0] 位，可以设置模拟输入信号的采样时间。

总转换时间 (T<sub>conv</sub>) 使用下列公式计算：

$$T_{conv} = T_{Sampling} + T_{Latency}$$

最小采样时间 T<sub>Sampling</sub> = 1.5 个周期 (ADST[7:0] = 0)，通道转换的最小延迟 T<sub>Latency</sub> = 12.5 个周期。

例如：

A/D 转换时钟 CK\_ADC = 14 MHz，采样时间 = 1.5 个周期：

$$T_{conv} = 1.5 + 12.5 = 14 \text{ 周期} = 1 \mu s$$

数据格式和对齐

A/D 转换结果有不同的输出数据格式，通过 ADCOFRn (n = 0 ~ 15) 寄存器中的 ADOFE 和 ADAL 位设置。参照表 29。

每个通道都有一个专用的偏移消除功能，偏移量是由用户定义的，并写入到 ADCOFRn (n = 0 ~ 15) 寄存器。被写入到数据寄存器的原始 A/D 转换数据是在 0x0FFF 和 0x0000 之间的无符号数字，仅 12 位有意义。如果 ADCOFRn (n = 0 ~ 15) 寄存器中的 ADOFEn 位设为 1，偏移消除功能使能。ADCDRy (y = 0 ~ 15) 或 ADCHDRy (y = 0 ~ 3) 寄存器中有意义的转换数据是 13 位格式，数据列的最高位是符号位。

表 29. ADCDRy[15:0] (y = 0 ~ 15) 和 ADCHDRy[15:0] (y = 0 ~ 3) 数据格式

ADOFE	ADAL	描述	数据格式
0	0	无符号右对齐	“0_0_0_0_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0”
0	1	无符号左对齐	“d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0_0_0_0_0”
1	0	偏移消除后有符号右对齐	“0_0_0_Sign bit_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0”
1	1	偏移消除后有符号左对齐	“Sign bit_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0_0_0_0_0”

模拟看门狗

A/D 转换器，包括一个看门狗功能用来监视转换后的数据。看门狗监视功能有两种阈值，即看门狗上限阈值和看门狗下限阈值，它们分别由看门狗上限和下限阈值寄存器定义。看门狗监视功能通过看门狗控制寄存器 ADCWCR 中看门狗上限阈值和下限阈值监视功能使能位 ADWUE 和 ADWLE 设置使能。通过配置 ADWCH 和 ADWALL 位可以指定要监控的通道。当转换后的数据小于或大于由 ADCLTR 或 ADCUTR 寄存器相应的位定义的下限阈值或上限阈值时，若看门狗下限或上限阈值监控功能使能，则置位 ADCIRAW 寄存器中的下限阈值中断标志 ADIRAWL 或上限阈值中断标志 ADIRAWU。如果下限阈值或上限阈值中断原始位被置位，且通过设置 ADCIER 寄存器中 ADIEL 或 ADIEU 使相应的中断使能，将产生 A/D 看门狗下限阈值或上限阈值中断。



## 中断

当 A/D 转换完成, 一个转换结束 EOC 事件将发生。对于 A/D 转换, 有三种 EOC 事件, 分别为单次采样 EOC、子组 EOC 和周期 EOC。当单通道转换已经完成, 单次采样 EOC 事件发生, 且 ADCIRAW 寄存器中单次采样 EOC 中断原始标志位 ADIRAWS 或 ADIRAWHS 位被置位。当子组转换已经完成, 子组 EOC 事件发生, 且 ADCIRAW 寄存器中子组 EOC 中断原始标志位 ADIRAWG 或 ADIRAWHG 位被置位。当周期转换已经完成, 周期 EOC 事件发生, 且 ADCIRAW 寄存器中周期 EOC 中断原始标志位 ADIRAWC 或 ADIRAWHC 位被置位。单次采样 EOC、子组 EOC 和周期 EOC 原始标志位被置位, ADCIER 寄存器中相应的中断使能位 ADIEHC、ADIEHG、ADIEHS、ADIEC、ADIEG 或 ADIES 位置 1, 产生相关中断。

转换完成后, 将 12 位数字数据存储在相关的 ADCDRy 或 ADCHDRy 寄存器中, 数据有效标志位的值, 即 ADVLDy 或 ADHVLDy 将从低转变为高。转换后的数据应由应用程序读取, 数据有效标志位 ADVLDy 或 ADHVLDy 将自动从高转变为低。否则, 数据覆盖事件发生, ADCIRAW 寄存器中数据覆盖中断原始标志位 ADIRAWO 或 ADIRAWHO 位将被置位。如果 ADCIER 寄存器 ADIEO 或 ADIEHO 中断使能位置为 1, 当相关的数据覆盖原标志位被置位, 将产生数据覆盖中断。

如果 A/D 看门狗监视功能使能, 通道转换后的数据是低于下限阈值或高于上限阈值, ADCIRAW 寄存器中看门狗下限或上限阈值中断原始标志位 ADIRAWL 或 ADIRAWU 被置位。当 ADIRAWL 或 ADIRAWU 位被置位, ADCIER 寄存器中相应的中断使能位 ADIEL 或 ADIEU 置 1, 将产生看门狗下限或上限阈值中断。

A/D 转换中断清除位用来清除相关的 A/D 转换中断原始位和中断状态位。在 A/D 转换中断清除寄存器 ADCICLR 中特定的 A/D 转换中断清除位写入 1, 将清除相应的 A/D 转换中断原始位和中断状态位。这些位被设置为 1 后, 由硬件自动清零。

## PDMA 请求

转换通道的值将存储在相应的数据寄存器。如果一个新的转换数据已存储 ADCDRy 或 ADCHDRy 寄存器中, A/D 转换器可以通过 A/D 转换器 EOC 中断告知单片机。用户也可以通过设置 ADCDMAR 寄存器中的位 ADDMAC、ADDMAG、ADDMAS、ADDMAHC、ADDMAHG 或 ADDMAHS 决定 PDMA 请求是否有效。当每次 A/D 转换结束时, PDMA 请求将自动产生。更多信息请参考 ADCDMAR 寄存器描述。

## 寄存器列表

下表显示 A/D 转换器寄存器和复位值。

表 30. A/D 转换器寄存器列表

寄存器	偏移量	描述	复位值
ADCRST	0x004	ADC 复位寄存器	0x0000_0000
ADCCONV	0x008	ADC 常规转换模式寄存器	0x0000_0000
ADCHCONV	0x00C	ADC 高优先级转换模式寄存器	0x0000_0000
ADCLST0	0x010	ADC 常规转换列表寄存器 0	0x0000_0000
ADCLST1	0x014	ADC 常规转换列表寄存器 1	0x0000_0000
ADCLST2	0x018	ADC 常规转换列表寄存器 2	0x0000_0000
ADCLST3	0x01C	ADC 常规转换列表寄存器 3	0x0000_0000
ADCHLST	0x020	ADC 高优先级转换列表寄存器	0x0000_0000
ADCOFR0	0x030	ADC 输入 0 偏移量寄存器	0x0000_0000
ADCOFR1	0x034	ADC 输入 1 偏移量寄存器	0x0000_0000
ADCOFR2	0x038	ADC 输入 2 偏移量寄存器	0x0000_0000
ADCOFR3	0x03C	ADC 输入 3 偏移量寄存器	0x0000_0000
ADCOFR4	0x040	ADC 输入 4 偏移量寄存器	0x0000_0000
ADCOFR5	0x044	ADC 输入 5 偏移量寄存器	0x0000_0000
ADCOFR6	0x048	ADC 输入 6 偏移量寄存器	0x0000_0000
ADCOFR7	0x04C	ADC 输入 7 偏移量寄存器	0x0000_0000
ADCOFR8	0x050	ADC 输入 8 偏移量寄存器	0x0000_0000
ADCOFR9	0x054	ADC 输入 9 偏移量寄存器	0x0000_0000
ADCOFR10	0x058	ADC 输入 10 偏移量寄存器	0x0000_0000
ADCOFR11	0x05C	ADC 输入 11 偏移量寄存器	0x0000_0000
ADCOFR12	0x060	ADC 输入 12 偏移量寄存器	0x0000_0000
ADCOFR13	0x064	ADC 输入 13 偏移量寄存器	0x0000_0000
ADCOFR14	0x068	ADC 输入 14 偏移量寄存器	0x0000_0000
ADCOFR15	0x06C	ADC 输入 15 偏移量寄存器	0x0000_0000
ADCSTR0	0x070	ADC 输入 0 采样时间寄存器	0x0000_0000
ADCSTR1	0x074	ADC 输入 1 采样时间寄存器	0x0000_0000
ADCSTR2	0x078	ADC 输入 2 采样时间寄存器	0x0000_0000
ADCSTR3	0x07C	ADC 输入 3 采样时间寄存器	0x0000_0000
ADCSTR4	0x080	ADC 输入 4 采样时间寄存器	0x0000_0000
ADCSTR5	0x084	ADC 输入 5 采样时间寄存器	0x0000_0000
ADCSTR6	0x088	ADC 输入 6 采样时间寄存器	0x0000_0000
ADCSTR7	0x08C	ADC 输入 7 采样时间寄存器	0x0000_0000
ADCSTR8	0x090	ADC 输入 8 采样时间寄存器	0x0000_0000
ADCSTR9	0x094	ADC 输入 9 采样时间寄存器	0x0000_0000
ADCSTR10	0x098	ADC 输入 10 采样时间寄存器	0x0000_0000
ADCSTR11	0x09C	ADC 输入 11 采样时间寄存器	0x0000_0000
ADCSTR12	0x0A0	ADC 输入 12 采样时间寄存器	0x0000_0000
ADCSTR13	0x0A4	ADC 输入 13 采样时间寄存器	0x0000_0000
ADCSTR14	0x0A8	ADC 输入 14 采样时间寄存器	0x0000_0000
ADCSTR15	0x0AC	ADC 输入 15 采样时间寄存器	0x0000_0000
ADCDR0	0x0B0	ADC 常规转换数据寄存器 0	0x0000_0000

寄存器	偏移量	描述	复位值
ADCDR1	0x0B4	ADC 常规转换数据寄存器 1	0x0000_0000
ADCDR2	0x0B8	ADC 常规转换数据寄存器 2	0x0000_0000
ADCDR3	0x0BC	ADC 常规转换数据寄存器 3	0x0000_0000
ADCDR4	0x0C0	ADC 常规转换数据寄存器 4	0x0000_0000
ADCDR5	0x0C4	ADC 常规转换数据寄存器 5	0x0000_0000
ADCDR6	0x0C8	ADC 常规转换数据寄存器 6	0x0000_0000
ADCDR7	0x0CC	ADC 常规转换数据寄存器 7	0x0000_0000
ADCDR8	0x0D0	ADC 常规转换数据寄存器 8	0x0000_0000
ADCDR9	0x0D4	ADC 常规转换数据寄存器 9	0x0000_0000
ADCDR10	0x0D8	ADC 常规转换数据寄存器 10	0x0000_0000
ADCDR11	0x0DC	ADC 常规转换数据寄存器 11	0x0000_0000
ADCDR12	0x0E0	ADC 常规转换数据寄存器 12	0x0000_0000
ADCDR13	0x0E4	ADC 常规转换数据寄存器 13	0x0000_0000
ADCDR14	0x0E8	ADC 常规转换数据寄存器 14	0x0000_0000
ADCDR15	0x0EC	ADC 常规转换数据寄存器 15	0x0000_0000
ADCHDR0	0x0F0	ADC 高优先级转换数据寄存器 0	0x0000_0000
ADCHDR1	0x0F4	ADC 高优先级转换数据寄存器 1	0x0000_0000
ADCHDR2	0x0F8	ADC 高优先级转换数据寄存器 2	0x0000_0000
ADCHDR3	0x0FC	ADC 高优先级转换数据寄存器 3	0x0000_0000
ADCTCR	0x100	ADC 常规触发控制寄存器	0x0000_0000
ADCTSR	0x104	ADC 常规触发源寄存器	0x0000_0000
ADCHTCR	0x110	ADC 高优先级触发控制寄存器	0x0000_0000
ADCHTSR	0x114	ADC 高优先级触发源寄存器	0x0000_0000
ADCWCR	0x120	ADC 看门狗控制寄存器	0x0000_0000
ADCLTR	0x124	ADC 看门狗下限阈值寄存器	0x0000_0000
ADCUTR	0x128	ADC 看门狗上限阈值寄存器	0x0000_0000
ADCIMR	0x130	ADC 中断使能寄存器	0x0000_0000
ADCIRAW	0x134	ADC 中断原始状态寄存器	0x0000_0000
ADCISR	0x138	ADC 中断状态寄存器	0x0000_0000
ADCICLR	0x13C	ADC 中断清除寄存器	0x0000_0000
ADCDMAR	0x140	ADC DMA 请求寄存器	0x0000_0000

寄存器描述

ADC 复位寄存器 – ADRST

ADC 软件复位寄存器

偏移量: 0x004

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位							ADRST
类型 / 复位								RW 0

位	字段	描述
[0]	ADRST	ADC 软件复位 0: 未复位 1: 复位 A/D 转换器，除 A/D 转换器控制器外

ADC 常规转换模式寄存器 – ADCCONV

该寄存器定义了 ADC 常规转换模式的设置，序列长度和 ADC 常规转换模式的子组长度。请注意，一旦 ADCCONV 寄存器的内容发生了变化，任何目前正在进行中的常规组转换将被退出，ADC 将被复位。应用程序发出下一个命令前等待至少有一个 A/D 转换时钟 ADCLK。

偏移量： 0x008  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位				ADSUBL			
类型 / 复位					RW	0	RW	0
	15	14	13	12	11	10	9	8
	保留位				ADSEQL			
类型 / 复位					RW	0	RW	0
	7	6	5	4	3	2	1	0
	ADCEN	保留位					ADMODE	
类型 / 复位	RW						RW	0

位	字段	描述															
[19:16]	ADSUBL	ADC 常规转换子组长度 非连续模式下的常规组每个子组的转换通道长度由 ADSUBL 字段定义。每个子组长度 = ADSUBL[3:0] + 1。如果 (ADSEQL[3:0] + 1) 不是 (ADSUBL[3:0] + 1) 的倍数，最后一个子组将比其它的短。															
[11:8]	ADSEQL	ADC 常规转换长度 0x00: 仅在指定的通道进行转换 ( 由 ADCLST0 寄存器中的 ADSEQ0 位指定 ) 其它: 序列长度 = ADSEQL[3:0] + 1															
[7]	ADCEN	ADC 使能 0: ADC 除能 1: ADC 使能 当该位清零，ADC 将除能，且 CK_ADC 时钟也将关闭。															
[1:0]	ADMODE	ADC 常规转换模式 <table><tr><th>ADMODE[1:0]</th><th>模式</th><th>描述</th></tr><tr><td>00</td><td>单次模式</td><td>启动触发后，在特定通道整体循环转换一次。</td></tr><tr><td>01</td><td>保留</td><td>—</td></tr><tr><td>10</td><td>连续模式</td><td>启动触发后，在特定通道全序列将进行连续转换直到转换模式改变。</td></tr><tr><td>11</td><td>非连续模式</td><td>启动触发后，当前的子组将进行转换。当最后一个子组完成转换，若其它开始触发生，从第一子组重新启动转换。</td></tr></table>	ADMODE[1:0]	模式	描述	00	单次模式	启动触发后，在特定通道整体循环转换一次。	01	保留	—	10	连续模式	启动触发后，在特定通道全序列将进行连续转换直到转换模式改变。	11	非连续模式	启动触发后，当前的子组将进行转换。当最后一个子组完成转换，若其它开始触发生，从第一子组重新启动转换。
ADMODE[1:0]	模式	描述															
00	单次模式	启动触发后，在特定通道整体循环转换一次。															
01	保留	—															
10	连续模式	启动触发后，在特定通道全序列将进行连续转换直到转换模式改变。															
11	非连续模式	启动触发后，当前的子组将进行转换。当最后一个子组完成转换，若其它开始触发生，从第一子组重新启动转换。															

## ADC 高优先级转换模式寄存器 – ADCHCONV

该寄存器定义了 ADC 高优先级转换模式的设置，序列长度和 ADC 高优先级转换模式的子组长度。请注意，一旦 ADCHCONV 寄存器的内容发生了变化，任何目前正在进行中的高优先级组转换将被退出，ADC 将被复位。应用程序发出下一个命令前等待至少有一个 A/D 转换时钟 ADCLK。

偏移量: 0x00C

复位值: 0x0000 0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位						ADHSUBL	
							RW	0
							RW	0
	15	14	13	12	11	10	9	8
类型 / 复位	保留位						ADHSEQL	
							RW	0
							RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	保留位						ADHMODE	
							RW	0
							RW	0

位	字段	描述	
[17:16]	ADHSUBL	ADC 高优先级转换子组长度 高优先级组非连续模式下，ADHSUBL 字段指定每个子组的转换通道长度。每个子组长度 = ADHSUBL[1:0] + 1。如果 (ADHSEQL[1:0] + 1) 不是 (ADHSUBL[1:0] + 1) 的倍数，最后一个子组将比其它的短。	
[9:8]	ADHSEQL	ADC 高优先级转换长度 0x00: 仅在指定的通道进行转换 ( 由 ADCHLST 寄存器中的 ADHSEQ0 位指定 ) 其它: 序列长度 = ADHSEQL[1:0] + 1	
[1:0]	ADHMODE	ADC 高优先级转换模式	
	ADHMODE[1:0]	模式	描述
	00	单次模式	启动触发后，在特定通道整体循环转换一次。
	01	保留	—
	10	连续模式	启动触发后，在特定通道全序列将进行连续转换直到转换模式改变。
	11	非连续模式	启动触发后，当前的子组将进行转换。当最后一个子组完成转换，若其它开始触发发生，从第一子组重新启动转换。

ADC 常规转换列表寄存器 0 – ADCLST0

该寄存器定义了 ADC 常规组的转换序列顺序 No.0 ~ No.3。

偏移量： 0x010  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位			ADSEQ3					
类型 / 复位				RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16	
	保留位			ADSEQ2					
类型 / 复位				RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8	
	保留位			ADSEQ1					
类型 / 复位				RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0	
	保留位			ADSEQ0					
类型 / 复位				RW	0	RW	0	RW	0

位	字段	描述
[28:24]	ADSEQ3	ADC 常规转换序列选择 3 定义 ADC 常规转换第 3 序列的 ADC 输入通道 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC: ADC_IN12 0xD: ADC_IN13 0xE: ADC_IN14 0xF: ADC_IN15 0x10: 模拟地, AV <sub>SS</sub> (V <sub>REF-</sub> ) 0x11: 模拟电源, AV <sub>DD</sub> (V <sub>REF+</sub> ) 0x12 ~ 0x1F: 设置无效。这些值不能被选择, 因为它们可能导致 ADC 不稳定。
[20:16]	ADSEQ2	ADC 常规转换序列选择 2
[12:8]	ADSEQ1	ADC 常规转换序列选择 1
[4:0]	ADSEQ0	ADC 常规转换序列选择 0

ADC 常规转换列表寄存器 1 – ADCLST1

该寄存器定义了 ADC 常规组的转换序列顺序 No.4 ~ No.7。

偏移量： 0x014  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24						
	保留位			ADSEQ7										
类型 / 复位				RW	0	RW	0	RW	0	0	RW	0	RW	0
	23	22	21	20	19	18	17	16						
	保留位			ADSEQ6										
类型 / 复位				RW	0	RW	0	RW	0	RW	0	RW	0	
	15	14	13	12	11	10	9	8						
	保留位			ADSEQ5										
类型 / 复位				RW	0	RW	0	RW	0	RW	0	RW	0	
	7	6	5	4	3	2	1	0						
	保留位			ADSEQ4										
类型 / 复位				RW	0	RW	0	RW	0	RW	0	RW	0	

位	字段	描述
[28:24]	ADSEQ7	ADC 常规转换序列选择 7 定义 ADC 常规转换第 7 序列的 ADC 输入通道 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC: ADC_IN12 0xD: ADC_IN13 0xE: ADC_IN14 0xF: ADC_IN15 0x10: 模拟地, AV <sub>SS</sub> (V <sub>REF-</sub> ) 0x11: 模拟电源, AV <sub>DD</sub> (V <sub>REF+</sub> ) 0x12 ~ 0x1F: 设置无效。这些值不能被选择, 因为它们可能导致 ADC 不稳定。
[20:16]	ADSEQ6	ADC 常规转换序列选择 6
[12:8]	ADSEQ5	ADC 常规转换序列选择 5
[4:0]	ADSEQ4	ADC 常规转换序列选择 4



ADC 常规转换列表寄存器 2 – ADCLST2

该寄存器定义了 ADC 常规组的转换序列顺序 No.8 ~ No.11。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24			
	保留位			ADSEQ11							
类型 / 复位				RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16			
	保留位			ADSEQ10							
类型 / 复位				RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8			
	保留位			ADSEQ9							
类型 / 复位				RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0			
	保留位			ADSEQ8							
类型 / 复位				RW	0	RW	0	RW	0	RW	0

位	字段	描述
[28:24]	ADSEQ11	ADC 常规转换序列选择 11 定义 ADC 常规转换第 11 序列的 ADC 输入通道 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC: ADC_IN12 0xD: ADC_IN13 0xE: ADC_IN14 0xF: ADC_IN15 0x10: 模拟地, AV <sub>SS</sub> (V <sub>REF-</sub> ) 0x11: 模拟电源, AV <sub>DD</sub> (V <sub>REF+</sub> ) 0x12 ~ 0x1F: 设置无效。这些值不能被选择, 因为它们可能导致 ADC 不稳定。
[20:16]	ADSEQ10	ADC 常规转换序列选择 10
[12:8]	ADSEQ9	ADC 常规转换序列选择 9
[4:0]	ADSEQ8	ADC 常规转换序列选择 8

ADC 常规转换列表寄存器 3 – ADCLST3

该寄存器定义了 ADC 常规组的转换序列顺序 No.12 ~ No.15。

偏移量： 0x01C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位			ADSEQ15					
类型 / 复位				RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16	
	保留位			ADSEQ14					
类型 / 复位				RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8	
	保留位			ADSEQ13					
类型 / 复位				RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0	
	保留位			ADSEQ12					
类型 / 复位				RW	0	RW	0	RW	0

位	字段	描述
[28:24]	ADSEQ15	ADC 常规转换序列选择 15 定义 ADC 常规转换第 15 序列的 ADC 输入通道 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC: ADC_IN12 0xD: ADC_IN13 0xE: ADC_IN14 0xF: ADC_IN15 0x10: 模拟地, AV <sub>SS</sub> (V <sub>REF-</sub> ) 0x11: 模拟电源, AV <sub>DD</sub> (V <sub>REF+</sub> ) 0x12 ~ 0x1F: 设置无效。这些值不能被选择, 因为它们可能导致 ADC 不稳定。
[20:16]	ADSEQ14	ADC 常规转换序列选择 14
[12:8]	ADSEQ13	ADC 常规转换序列选择 13
[4:0]	ADSEQ12	ADC 常规转换序列选择 12

## ADC 高优先级转换列表寄存器 – ADCHLST

该寄存器定义了 ADC 高优先级组的转换序列顺序 No.0 ~ No.3。

偏移量: 0x020

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24			
	保留位			ADHSEQ3							
类型 / 复位				RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16			
	保留位			ADHSEQ2							
类型 / 复位				RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8			
	保留位			ADHSEQ1							
类型 / 复位				RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0			
	保留位			ADHSEQ0							
类型 / 复位				RW	0	RW	0	RW	0	RW	0

位	字段	描述
[28:24]	ADHSEQ3	ADC 高优先级转换序列选择 3 定义 ADC 高优先级转换第 3 序列的 ADC 输入通道 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC: ADC_IN12 0xD: ADC_IN13 0xE: ADC_IN14 0xF: ADC_IN15 0x10: 模拟地, $AV_{SS}$ ( $V_{REF-}$ ) 0x11: 模拟电源, $AV_{DD}$ ( $V_{REF+}$ ) 0x12 ~ 0x1F: 设置无效。这些值不能被选择, 因为它们可能导致 ADC 不稳定。
[20:16]	ADHSEQ2	ADC 高优先级转换序列选择 2
[12:8]	ADHSEQ1	ADC 高优先级转换序列选择 1
[4:0]	ADHSEQ0	ADC 高优先级转换序列选择 0

## ADC 输入偏移量寄存器 n – ADCOFRn, n = 0 ~ 15

该寄存器定义了 ADC 输入通道 n 的偏移量以及偏移消除功能使能控制。

偏移量: 0x030 ~ 0x06C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	ADOFEn	ADALn	保留位		ADOFn			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	ADOFn							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15]	ADOFEn	ADC 输入通道 n 偏移消除使能位 (n = 0 ~ 15) 0: 除能 ADC_INn 偏移消除功能 1: 使能 ADC_INn 偏移消除功能
[14]	ADALn	ADC 输入通道 n 数据对齐 (n = 0 ~ 15) 0: 右对齐 1: 左对齐
[11:0]	ADOFn	ADC 输入通道 n 偏移量 (n = 0 ~ 15) 从 ADC 数据寄存器 ADCDR 读取的值是格式转换之后的 ADC 转换器的原始数据减去通道 n (ADC_INn) 上的偏移量。

**ADC 输入采样时间寄存器 n – ADCSTRn, n = 0 ~ 15**

该寄存器定义了 ADC 通道的采样时间。

偏移量： 0x070 ~ 0x0AC

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	ADSTn							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[7:0]	ADSTn	ADC 输入通道 n 采样时间 (n = 0 ~ 15) 采样时间 = (STn[7:0] + 1.5) ADC 时钟。

## ADC 常规转换数据寄存器 y – ADCDRy, y = 0 ~ 15

该寄存器定义了 ADCLSTn (n = 0 ~ 3) 寄存器中的 ADSEQy 常规转换数据。

偏移量: 0x0B0 ~ 0x0EC

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	ADVLDy	保留位						
类型 / 复位	RC	0						
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	ADDy							
类型 / 复位	RO	0	RO	0	RO	0	RO	0
	7	6	5	4	3	2	1	0
	ADDy							
类型 / 复位	RO	0	RO	0	RO	0	RO	0

位	字段	描述
[31]	ADVLDy	ADC 序列 y 常规转换数据有效位 (y = 0 ~ 15) 0: 数据无效或已经被读出 1: 新数据有效
[15:0]	ADDy	ADC 序列 y 转换数据 (y = 0 ~ 15) ADCLSTn 寄存器中 SEQy 字段的序列转换结果。(n = 0 ~ 3)

## ADC 高优先级转换数据寄存器 y – ADCHDRy, y = 0 ~ 3

该寄存器定义了 ADCHLST 寄存器中的 ADHSEQy 高优先级转换数据。

偏移量: 0x0F0 ~ 0x0FC

复位值: 0x0000\_0000

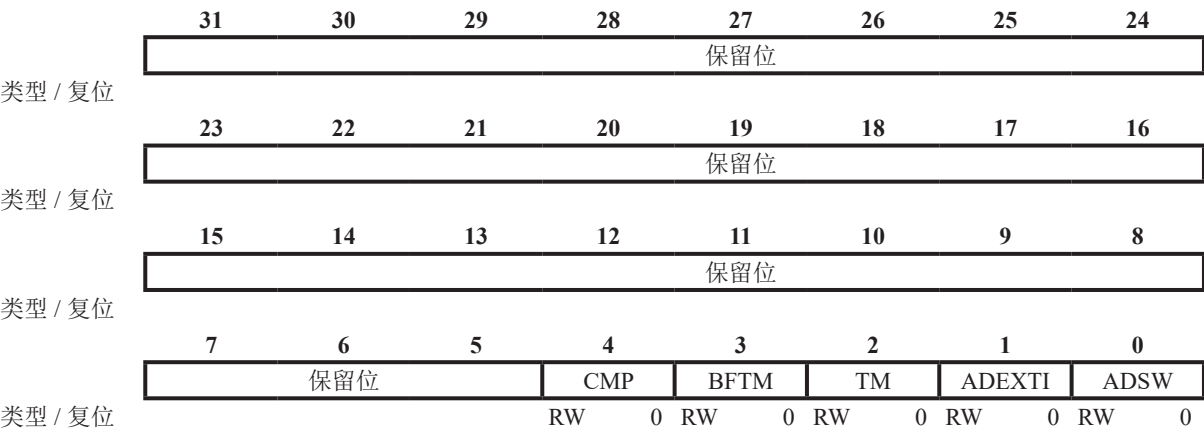
	31	30	29	28	27	26	25	24
	ADHVLDy	保留位						
类型 / 复位	RC	0						
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	ADHDy							
类型 / 复位	RO	0	RO	0	RO	0	RO	0
	7	6	5	4	3	2	1	0
	ADHDy							
类型 / 复位	RO	0	RO	0	RO	0	RO	0

位	字段	描述
[31]	ADHVLDy	ADC 序列 y 高优先级转换数据有效位 (y = 0 ~ 3) 0: 数据无效或已经被读出 1: 新数据有效
[15:0]	ADHDy	ADC 序列 y 高优先级转换数据 (y = 0 ~ 3) ADCHLST 寄存器中 HSEQy 高优先级转换结果。

ADC 常规触发控制寄存器 – ADCTCR

该寄存器包含了 ADC 常规转换启动转换的触发使能位。

偏移量： 0x100  
复位值： 0x0000\_0000



位	字段	描述
[4]	CMP	CMP 事件触发的 ADC 常规转换 0: 除能 CMP 功能触发的常规转换 1: 使能 CMP 功能触发的常规转换
[3]	BFTM	BFTM 事件触发的 ADC 常规转换 0: 除能 BFTM 功能触发的常规转换 1: 使能 BFTM 功能触发的常规转换
[2]	TM	GPTM 或 MCTM 事件触发的 ADC 常规转换 0: 除能 GPTM 或 MCTM 功能触发的常规转换 1: 使能 GPTM 或 MCTM 功能触发的常规转换
[1]	ADEXTI	EXTI 事件触发的 ADC 常规转换 0: 除能 EXTI 功能触发的常规转换 1: 使能 EXTI 功能触发的常规转换
[0]	ADSW	软件触发的 ADC 常规转换 0: 除能软件功能触发的常规转换 1: 使能软件功能触发的常规转换



## ADC 常规触发源寄存器 – ADCTSR

该寄存器包含了常规转换的触发源的选择和软件触发位。

偏移量: 0x104

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位					TME		
类型 / 复位						RW	0	RW
	23	22	21	20	19	18	17	16
	保留位			CMPS	BFTMS	TMS		
类型 / 复位				RW	0	RW	0	RW
	15	14	13	12	11	10	9	8
	保留位					ADEXTIS		
类型 / 复位						RW	0	RW
	7	6	5	4	3	2	1	0
	保留位							ADSC
类型 / 复位								RW

位	字段	描述
[26:24]	TME	ADC 常规转换的 GPTM 或 MCTM 触发事件的选择 000: GPTM 或 MCTM MTO 事件 001: GPTM 或 MCTM CH00 事件 010: GPTM 或 MCTM CH10 事件 011: GPTM 或 MCTM CH20 事件 100: GPTM 或 MCTM CH30 事件 其它: 保留位 – 不使用, 否则, 转换结果将是不可预知的。
[20]	CMPS	ADC 常规转换的 CMP 触发定时器的选择 0: CMP0 1: CMP1
[19]	BFTMS	ADC 常规转换的 BFTM 触发定时器的选择 0: BFTM0 1: BFTM1
[18:16]	TMS	ADC 常规转换的 GPTM 或 MCTM 触发定时器的选择 000: MCTM0 001: MCTM1 010: GPTM0 011: GPTM1 其它: 保留位 – 不使用, 否则, 转换结果将是不可预知的。
[11:8]	ADEXTIS	ADC 常规转换的 EXTI 触发源的选择 0x00: EXTI 引脚 0 0x01: EXTI 引脚 1 ... 0x0F: EXTI 引脚 15
[0]	ADSC	ADC 常规转换的软件触发位 0: 无操作 1: 立即启动常规转换 通过软件设置手动启动常规转换。转换启动后由硬件清零。

ADC 高优先级触发控制寄存器 – ADCHTCR

该寄存器包含了 ADC 高优先级转换启动转换的触发使能位。

偏移量: 0x110  
复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位	7	6	5	4	3	2	1	0
	保留位			HCMP	HBFTM	HTM	ADHEXTI	ADHSW
				RW	0	RW	0	RW
					0	RW	0	RW
						0	RW	0

位	字段	描述
[4]	HCMP	CMP 事件触发的 ADC 高优先级转换 0: 除能 CMP 功能触发的高优先级转换 1: 使能 CMP 功能触发的高优先级转换
[3]	HBFTM	BFTM 事件触发的 ADC 高优先级转换 0: 除能 BFTM 功能触发的高优先级转换 1: 使能 BFTM 功能触发的高优先级转换
[2]	HTM	GPTM 或 MCTM 事件触发的 ADC 高优先级转换 0: 除能 GPTM 或 MCTM 功能触发的高优先级转换 1: 使能 GPTM 或 MCTM 功能触发的高优先级转换
[1]	ADHEXTI	EXTI 事件触发的 ADC 高优先级转换 0: 除能 EXTI 功能触发的高优先级转换 1: 使能 EXTI 功能触发的高优先级转换
[0]	ADHSW	软件触发的 ADC 高优先级转换 0: 除能软件功能触发的高优先级转换 1: 使能软件功能触发的高优先级转换

## ADC 高优先级触发源寄存器 – ADCHTSR

该寄存器包含了高优先级转换的触发源的选择和软件触发位。

偏移量: 0x114

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位					HTME		
类型 / 复位						RW	0	RW
	23	22	21	20	19	18	17	16
	保留位			HCMP5	HBFTMS	HTMS		
类型 / 复位				RW	0	RW	0	RW
	15	14	13	12	11	10	9	8
	保留位				ADHEXTIS			
类型 / 复位					RW	0	RW	0
	7	6	5	4	3	2	1	0
	保留位							ADHSC
类型 / 复位								RW

位	字段	描述
[26:24]	HTME	ADC 高优先级转换的 GPTM 或 MCTM 触发事件的选择 000: GPTM 或 MCTM MTO 事件 001: GPTM 或 MCTM CH00 事件 010: GPTM 或 MCTM CH10 事件 011: GPTM 或 MCTM CH20 事件 100: GPTM 或 MCTM CH30 事件 其它: 保留位 – 不使用, 否则, 转换结果将是不可预知的。
[20]	HCMP5	ADC 高优先级转换的 CMP 触发定时器的选择 0: CMP0 1: CMP1
[19]	HBFTMS	ADC 高优先级转换的 BFTM 触发定时器的选择 0: BFTM0 1: BFTM1
[18:16]	HTMS	ADC 高优先级转换的 GPTM 或 MCTM 触发定时器的选择 000: MCTM0 001: MCTM1 010: GPTM0 011: GPTM1 其它: 保留位 – 不使用, 否则, 转换结果将是不可预知的。
[11:8]	ADHEXTIS	ADC 高优先级转换的 EXTI 触发源的选择 0x00: EXTI 引脚 0 0x01: EXTI 引脚 1 ... 0x0F: EXTI 引脚 15
[0]	ADHSC	ADC 高优先级转换的软件触发位 0: 无操作 1: 立即启动高优先级转换 通过软件设置手动启动高优先级转换。转换启动后由硬件清零。

## ADC 看门狗控制寄存器 – ADCWCR

该寄存器定义了 ADC 看门狗功能的控制位和状态。

偏移量: 0x120

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位				ADUCH			
类型 / 复位					RO	0	RO	0
	23	22	21	20	19	18	17	16
	保留位				ADLCH			
类型 / 复位					RO	0	RO	0
	15	14	13	12	11	10	9	8
	保留位				ADWCH			
类型 / 复位					RW	0	RW	0
	7	6	5	4	3	2	1	0
	保留位				ADWALL		ADWUE	ADWLE
类型 / 复位					RW	0	RW	0

位	字段	描述
[27:24]	ADUCH	上限阈值通道状态 0000: ADC_IN0 高于上限阈值 0001: ADC_IN1 高于上限阈值 ... 1111: ADC_IN15 高于上限阈值 其它: 保留位
[19:16]	ADLCH	下限阈值通道状态 0000: ADC_IN0 低于下限阈值 0001: ADC_IN1 低于下限阈值 ... 1111: ADC_IN15 低于下限阈值 其它: 保留位
[11:8]	ADWCH	ADC 看门狗指定通道选择 0000: ADC_IN0 被监控 0001: ADC_IN1 被监控 ... 1111: ADC_IN15 被监控 其它: 保留位
[2]	ADWALL	ADC 看门狗功能指定的通道或所有通道设置 0: 仅 ADWCH 字段指定的通道被监控 1: 所有通道被监控
[1]	ADWUE	ADC 看门狗上限阈值使能位 0: 除能上限阈值功能 1: 使能上限阈值功能
[0]	ADWLE	ADC 看门狗下限阈值使能位 0: 除能下限阈值功能 1: 使能下限阈值功能

ADC 看门狗下限阈值寄存器 – ADCLTR

该寄存器定义了 ADC 看门狗功能的下限阈值。

偏移量: 0x124  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位									
类型 / 复位										
	15	14	13	12	11	10	9	8		
	保留位				ADLT					
类型 / 复位					RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0		
	ADLT									
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[11:0]	ADLT	ADC 看门狗的下限阈值 指定 ADC 看门狗功能的下限阈值。

ADC 看门狗上限阈值寄存器 – ADCUTR

该寄存器定义了 ADC 看门狗功能的上限阈值。

偏移量： 0x128  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24				
	保留位											
类型 / 复位												
	23	22	21	20	19	18	17	16				
	保留位											
类型 / 复位												
	15	14	13	12	11	10	9	8				
	保留位				ADUT							
类型 / 复位					RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0				
	ADUT											
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[11:0]	ADUT	ADC 看门狗的上限阈值 指定 ADC 看门狗功能的上限阈值。

## ADC 中断使能寄存器 – ADCIER

该寄存器包含了 ADC 的中断使能位。

偏移量： 0x130

复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位						ADIEHO	ADIEO
							RW 0	RW 0
类型 / 复位	23	22	21	20	19	18	17	16
	保留位						ADIEU	ADIEL
							RW 0	RW 0
类型 / 复位	15	14	13	12	11	10	9	8
	保留位						ADIEHC	ADIEHG
							RW 0	RW 0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位						ADIEC	ADIEG
							RW 0	RW 0

位	字段	描述
[25]	ADIEHO	ADC 高优先级数据寄存器覆盖中断使能位 0: ADC 高优先级数据寄存器覆盖中断被除能 1: ADC 高优先级数据寄存器覆盖中断被使能
[24]	ADIEO	ADC 常规数据寄存器覆盖中断使能位 0: ADC 常规数据寄存器覆盖中断被除能 1: ADC 常规数据寄存器覆盖中断被使能
[17]	ADIEU	ADC 看门狗上限阈值中断使能位 0: ADC 看门狗上限阈值中断被除能 1: ADC 看门狗上限阈值中断被使能
[16]	ADIEL	ADC 看门狗下限阈值中断使能位 0: ADC 看门狗下限阈值中断被除能 1: ADC 看门狗下限阈值中断被使能
[10]	ADIEHC	ADC 高优先级周期 EOC 中断使能位 0: ADC 高优先级周期转换结束中断被除能 1: ADC 高优先级周期转换结束中断被使能
[9]	ADIEHG	ADC 高优先级子组 EOC 中断使能位 0: ADC 高优先级子组转换结束中断被除能 1: ADC 高优先级子组转换结束中断被使能
[8]	ADIEHS	ADC 高优先级单次 EOC 中断使能位 0: ADC 高优先级单次转换结束中断被除能 1: ADC 高优先级单次转换结束中断被使能
[2]	ADIEC	ADC 常规周期 EOC 中断使能位 0: ADC 常规周期转换结束中断被除能 1: ADC 常规周期转换结束中断被使能
[1]	ADIEG	ADC 常规子组 EOC 中断使能位 0: ADC 常规子组转换结束中断被除能 1: ADC 常规子组转换结束中断被使能
[0]	ADIES	ADC 常规单次 EOC 中断使能位 0: ADC 常规单次转换结束中断被除能 1: ADC 常规单次转换结束中断被使能

## ADC 中断原始状态寄存器 – ADCIRAW

该寄存器包含了 ADC 的中断原始状态位。

偏移量: 0x134

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位						ADIRAWHO	ADIRAWO
类型 / 复位							RO 0	RO 0
	23	22	21	20	19	18	17	16
	保留位						ADIRAWU	ADIRAWL
类型 / 复位							RO 0	RO 0
	15	14	13	12	11	10	9	8
	保留位						ADIRAWHC	ADIRAWHG
类型 / 复位							RO 0	RO 0
	7	6	5	4	3	2	1	0
	保留位						ADIRAWC	ADIRAWG
类型 / 复位							RO 0	RO 0

位	字段	描述
[25]	ADIRAWHO	ADC 高优先级数据寄存器覆盖中断原始状态位 0: 未发生 ADC 高优先级数据寄存器覆盖中断 1: 发生 ADC 高优先级数据寄存器覆盖中断
[24]	ADIRAWO	ADC 常规数据寄存器覆盖中断原始状态位 0: 未发生 ADC 常规数据寄存器覆盖中断 1: 发生 ADC 常规数据寄存器覆盖中断
[17]	ADIRAWU	ADC 看门狗上限阈值中断原始状态位 0: 未发生 ADC 看门狗上限阈值中断 1: 发生 ADC 看门狗上限阈值中断
[16]	ADIRAWL	ADC 看门狗下限阈值中断原始状态位 0: 未发生 ADC 看门狗下限阈值中断 1: 发生 ADC 看门狗下限阈值中断
[10]	ADIRAWHC	ADC 高优先级周期 EOC 中断原始状态位 0: 未发生 ADC 高优先级周期转换结束中断 1: 发生 ADC 高优先级周期转换结束中断
[9]	ADIRAWHG	ADC 高优先级子组 EOC 中断原始状态位 0: 未发生 ADC 高优先级子组转换结束中断 1: 发生 ADC 高优先级子组转换结束中断
[8]	ADIRAWHS	ADC 高优先级单次 EOC 中断原始状态位 0: 未发生 ADC 高优先级单次转换结束中断 1: 发生 ADC 高优先级单次转换结束中断
[2]	ADIRAWC	ADC 常规周期 EOC 中断原始状态位 0: 未发生 ADC 常规周期转换结束中断 1: 发生 ADC 常规周期转换结束中断
[1]	ADIRAWG	ADC 常规子组 EOC 中断原始状态位 0: 未发生 ADC 常规子组转换结束中断 1: 发生 ADC 常规子组转换结束中断
[0]	ADIRAWS	ADC 常规单次 EOC 中断原始状态位 0: 未发生 ADC 常规单次转换结束中断 1: 发生 ADC 常规单次转换结束中断



## ADC 中断状态寄存器 – ADCISR

该寄存器包含了 ADC 的中断状态位。如果相关的中断事件发生和相关的使能位置 1，相应的状态位将被置 1。

偏移量： 0x138

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位						ADISRHO	ADISRO
类型 / 复位							RO 0	RO 0
	23	22	21	20	19	18	17	16
	保留位						ADISRU	ADISRL
类型 / 复位							RO 0	RO 0
	15	14	13	12	11	10	9	8
	保留位						ADISRHC	ADISRHG
类型 / 复位							RO 0	RO 0
	7	6	5	4	3	2	1	0
	保留位						ADISRC	ADISRG
类型 / 复位							RO 0	RO 0

位	字段	描述
[25]	ADISRHO	ADC 高优先级数据寄存器覆盖中断状态位 0: 未发生 ADC 高优先级数据寄存器覆盖中断或相关中断控制除能 1: 发生 ADC 高优先级数据寄存器覆盖中断且相关中断控制使能
[24]	ADISRO	ADC 常规数据寄存器覆盖中断状态位 0: 未发生 ADC 常规数据寄存器覆盖中断或相关中断控制除能 1: 发生 ADC 常规数据寄存器覆盖中断且相关中断控制使能
[17]	ADISRU	ADC 看门狗上限阈值中断状态位 0: 未发生 ADC 看门狗上限阈值中断或相关中断控制除能 1: 发生 ADC 看门狗上限阈值中断且相关中断控制使能
[16]	ADISRL	ADC 看门狗下限阈值中断状态位 0: 未发生 ADC 看门狗下限阈值中断或相关中断控制除能 1: 发生 ADC 看门狗下限阈值中断且相关中断控制使能
[10]	ADISRHC	ADC 高优先级周期 EOC 中断状态位 0: 未发生 ADC 高优先级周期转换结束中断 1: 发生 ADC 高优先级周期转换结束中断且相关中断控制使能
[9]	ADISRHG	ADC 高优先级子组 EOC 中断状态位 0: 未发生 ADC 高优先级子组转换结束中断 1: 发生 ADC 高优先级子组转换结束中断且相关中断控制使能
[8]	ADISRHS	ADC 高优先级单次 EOC 中断状态位 0: 未发生 ADC 高优先级单次转换结束中断 1: 发生 ADC 高优先级单次转换结束中断且相关中断控制使能
[2]	ADISRC	ADC 常规周期 EOC 中断状态位 0: 未发生 ADC 常规周期转换结束中断 1: 发生 ADC 常规周期转换结束中断且相关中断控制使能
[1]	ADISRG	ADC 常规子组 EOC 中断状态位 0: 未发生 ADC 常规子组转换结束中断 1: 发生 ADC 常规子组转换结束中断且相关中断控制使能
[0]	ADISRS	ADC 常规单次 EOC 中断状态位 0: 未发生 ADC 常规单次转换结束中断 1: 发生 ADC 常规单次转换结束中断且相关中断控制使能

## ADC 中断清除寄存器 – ADCICLR

该寄存器用来清除 ADC 的中断原始位和中断状态位。这些位由软件置位来清除中断状态位，被置 1 后，由硬件自动清零。

偏移量： 0x13C

复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位						ADICLRHO	ADICLRO
							WO 0	WO 0
类型 / 复位	23	22	21	20	19	18	17	16
	保留位						ADICLRU	ADICLRL
							WO 0	WO 0
类型 / 复位	15	14	13	12	11	10	9	8
	保留位						ADICLRHC	ADICLRHG
							WO 0	WO 0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位						ADICLRC	ADICLRG
							WO 0	WO 0

位	字段	描述
[25]	ADICLRHO	ADC 高优先级数据寄存器覆盖中断状态清除位 0: 不影响 1: 清除 ADISRHO 和 ADIRAWHO 位
[24]	ADICLRO	ADC 常规数据寄存器覆盖中断状态清除位 0: 不影响 1: 清除 ADISRO 和 ADIRAWO 位
[17]	ADICLRU	ADC 看门狗上限阈值中断状态清除位 0: 不影响 1: 清除 ADISRU 和 ADIRAWU 位
[16]	ADICLRL	ADC 看门狗下限阈值中断状态清除位 0: 不影响 1: 清除 ADISRL 和 ADIRAWL 位
[10]	ADICLRHC	ADC 高优先级周期 EOC 中断状态清除位 0: 不影响 1: 清除 ADISRHC 和 ADIRAWHC 位
[9]	ADICLRHG	ADC 高优先级子组 EOC 中断状态清除位 0: 不影响 1: 清除 ADISRHG 和 ADIRAWHG 位
[8]	ADICLRHS	ADC 高优先级单次 EOC 中断状态清除位 0: 不影响 1: 清除 ADISRHS 和 ADIRAWHS 位
[2]	ADICLRC	ADC 常规周期 EOC 中断状态清除位 0: 不影响 1: 清除 ADISRC 和 ADIRAWC 位
[1]	ADICLRG	ADC 常规子组 EOC 中断状态清除位 0: 不影响 1: 清除 ADISRG 和 ADIRAWG 位

位	字段	描述
[0]	ADICLRS	ADC 常规单次 EOC 中断状态清除位 0: 不影响 1: 清除 ADISRS 和 ADIRAWS 位

### ADC DMA 请求寄存器 – ADCDMAR

该寄存器包含了 ADC DMA 请求使能位。

偏移量: 0x140

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位					ADDMAHC	ADDMAHG	ADDMAHS
类型 / 复位						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	保留位					ADDMAC	ADDMAG	ADDMAS
类型 / 复位						RW 0	RW 0	RW 0

位	字段	描述
[10]	ADDMAHC	ADC 高优先级周期 EOC DMA 请求使能位 0: ADC 高优先级周期转换结束 DMA 请求除能 1: ADC 高优先级周期转换结束 DMA 请求使能
[9]	ADDMAHG	ADC 高优先级子组 EOC DMA 请求使能位 0: ADC 高优先级子组转换结束 DMA 请求除能 1: ADC 高优先级子组转换结束 DMA 请求使能
[8]	ADDMAHS	ADC 高优先级单次 EOC DMA 请求使能位 0: ADC 高优先级单次转换结束 DMA 请求除能 1: ADC 高优先级单次转换结束 DMA 请求使能
[2]	ADDMAC	ADC 常规周期 EOC DMA 请求使能位 0: ADC 常规周期转换结束 DMA 请求除能 1: ADC 常规周期转换结束 DMA 请求使能
[1]	ADDMAG	ADC 常规子组 EOC DMA 请求使能位 0: ADC 常规子组转换结束 DMA 请求除能 1: ADC 常规子组转换结束 DMA 请求使能
[0]	ADDMAS	ADC 常规单次 EOC DMA 请求使能位 0: ADC 常规单次转换结束 DMA 请求除能 1: ADC 常规单次转换结束 DMA 请求使能

## 13 比较器 (CMP)

## 简介

该系列单片机具有两个通用比较器 (CMP)。他们可配置作为单独的比较器使用,也可以与其它不同的外设 IP 一起使用。每个比较器有置起 NVIC 中断的能力,或通过 EXTI 唤醒事件管理单元从休眠或深度休眠模式唤醒 CPU。

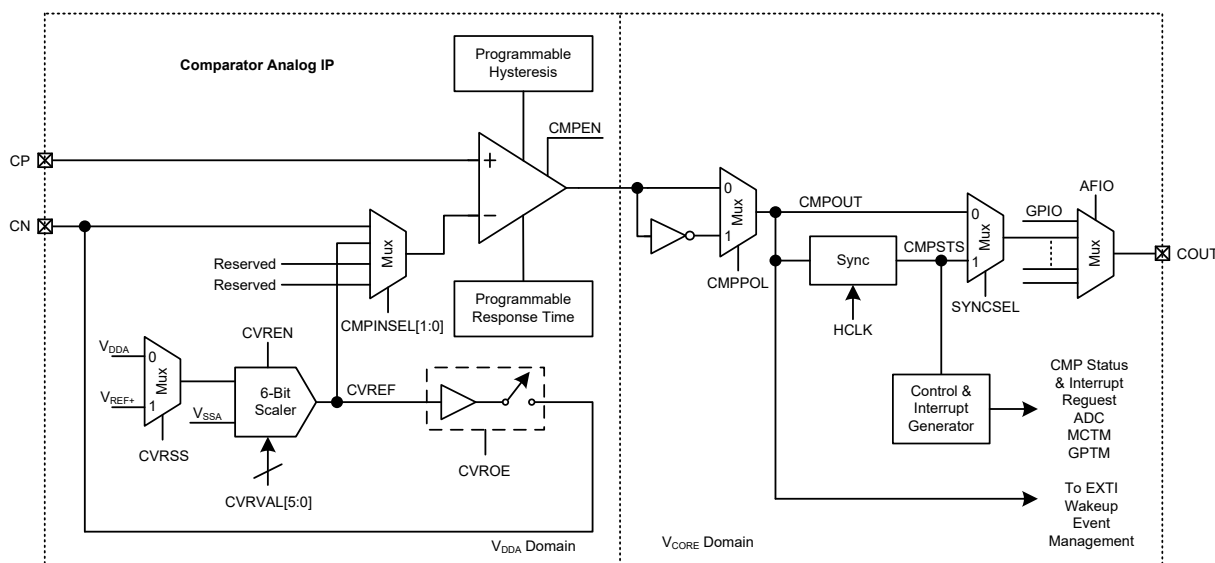


图 34. 比较器方框图

## 特性

- Rail-to-rail 比较器
- 每个比较器带有可配置负端输入灵活的电压选项
  - 专用 I/O 引脚
  - 内部 6-bit 计数器提供的参考电压
- 可编程迟滞
- 可编程速率和功耗
- 比较器输出可通过 I/O 输出或作为定时器, A/D 转换器输入
- 由 6-bit 计数器提供的可编程内部电压参考
- 比较器可产生中断并支持通过 EXTI 控制器从休眠或深度休眠模式唤醒

## 功能描述

### 比较器输入和输出

首先需配置 AFIO 控制寄存器选择 I/O 引脚作为比较器的输入或输出引脚。详细的比较器输入或输出功能请参考规格书中的引脚配置表格。输出可内部连接到各种定时器或用于触发 A/D 转换器功能。比较器输出可同时用于内部和外部功能。

### 比较器电压参考

比较器参考电压来自于一个 64-tap 电阻梯形网，提供可选的参考电压。当参考未使用时，可将其暂停以节省功耗。比较器参考电压提供 64 种不同的电压值。下面的等式用来计算比较器的参考电压：

$$CVREF = CVRVAL \times (V_{DDA} - V_{SSA}) / 63, \text{ 如果 CVR 参考电压源选择来自于 } V_{DDA}。$$

6-bit 计数器正端参考电压可以来自于  $V_{DDA}$  或外部  $VREF+$  引脚。电压源通过比较器控制寄存器 CMPCR 中的 CVRSS 位选择。CVREF 输出用来给模拟比较器提供一个参考电压。可内部使用或由 CMPCR 控制寄存器中的 CVROE 位设置连接到 CN 引脚。当 CVREF 输出电压变化时，必须考虑到比较器参考电压的设定时间。

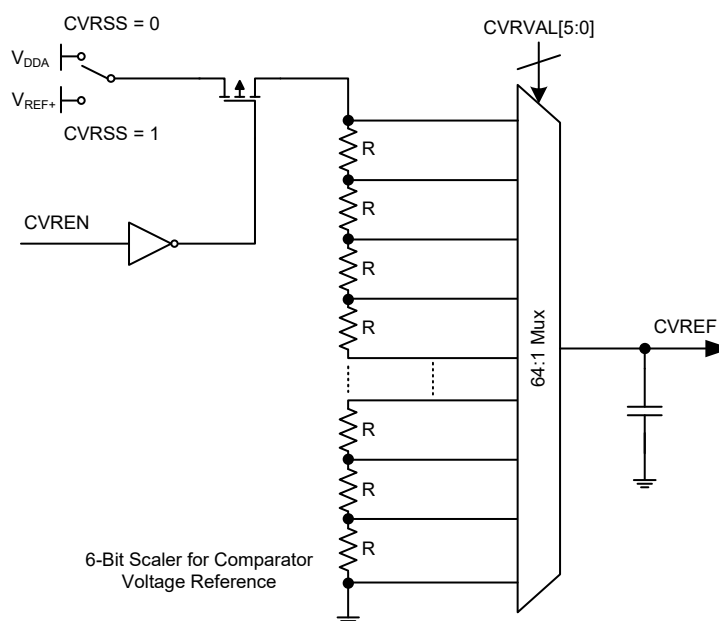


图 35. 6-Bit Scaler 配置比较器参考电压输出方框图

## 中断和唤醒

比较器的输出波形产生一个上升沿或下降沿且其相应的中断使能控制位也被置 1 时，模拟比较器产生一个中断。

例如，当一个比较器输出的上升沿发生，比较器转换标志寄存器 CMPTFR 中比较器的上升沿标志位 CMPRF 将被置位。如果比较器中断使能寄存器 CMPIER 中比较器输出上升沿中断使能控制位 CMPRIEN 使能，将产生一个中断并发送到 NVIC 单元。对比较器转换标志寄存器 CMPTFR 中比较器上升沿标志位 CMPRF 写 1 时，将会清零 CMPRF 位。比较器输出下降沿中断也是相同的相应中断设置。比较器中断信号方框图如图 36 所示。

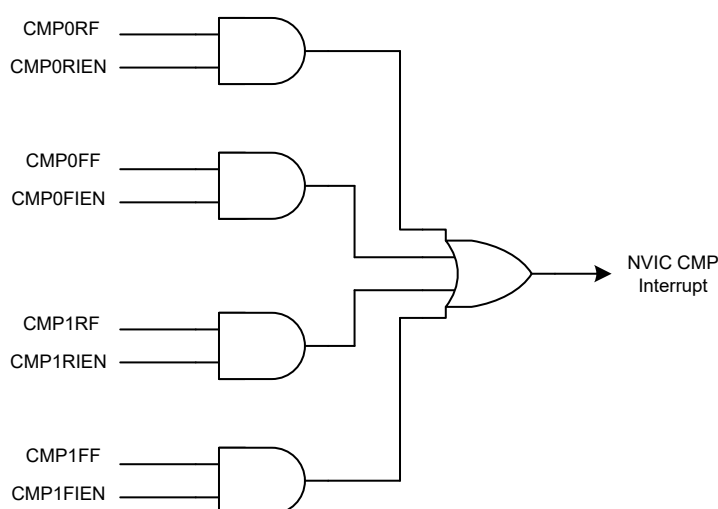


图 36. 比较器输出中断信号

比较器输出也可以内部连接到 EXTI 唤醒事件管理单元。当比较器控制寄存器 CMPCR 中的比较器唤醒使能位 CMPWPEN 置 1 时，比较器输出上升沿转换将 MCU 从休眠或深度休眠模式中唤醒。比较器输出唤醒信号方框图如图 37 所示。

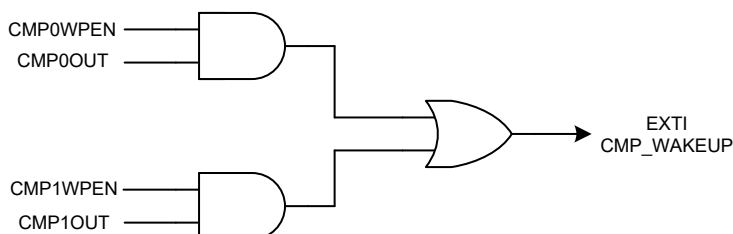


图 37. 比较器输出唤醒结构

## 功耗模式和迟滞

比较器响应时间可以被编程以满足电源功耗和应用速度的折中需求。CMPCR 寄存器中的 CMPSM 位置为 0 来使比较器工作在具有低电源功耗的低速模式。

比较器有四个迟滞电平来避免由于噪声信号而产生的伪输出边沿转换。CMPCR 寄存器中的 CMPHM[1:0] 位可选择不同的迟滞电平。

## 比较器写保护机制

由于比较器用于安全目的, 所以有必要确保比较器配置不会由于假的比较器访问或程序计数器改变而被更改。出于此目的, 通过写入一个特定的值到比较器控制寄存器 CMPCR 中的 PROTECT 字段可进行写保护。在配置 CMPCR 寄存器中的 [15:0] 位之前, CMPCR 寄存器中的保护位 [31:16] 必须写入 0x9C3A, 这样写保护功能将被除能, 位 [15:0] 可由应用程序配置。出于同一个原因, 比较器输入和输出也可由 GPIO 单元的端口 n 锁定寄存器 PnLOCKR (n = A ~ E) 使用相关锁定位来锁定。

## 寄存器列表

下表显示 CMP 寄存器和复位值。

表 31. CMP 寄存器列表

寄存器	偏移量	描述	复位值
CMPCR0	0x000	比较器控制寄存器 0	0x0001_0000
CVRVALR0	0x004	比较器电压参考值寄存器 0	0x0000_0000
CMPIER0	0x008	比较器中断使能寄存器 0	0x0000_0000
CMPTFR0	0x00C	比较器转换标志寄存器 0	0x0000_0000
CMPCR1	0x100	比较器控制寄存器 1	0x0001_0000
CVRVALR1	0x104	比较器电压参考值寄存器 1	0x0000_0000
CMPIER1	0x108	比较器中断使能寄存器 1	0x0000_0000
CMPTFR1	0x10C	比较器转换标志寄存器 1	0x0000_0000

寄存器描述

比较器控制寄存器 n – CMPCRn, n = 0 或 1

该寄存器包含了比较器和比较器电压参考控制位。

偏移量: 0x000 (n = 0), 0x100 (n = 1)  
复位值: 0x0001\_0000

	31	30	29	28	27	26	25	24
	PROTECT							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	PROTECT							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	CMPSTS	CMPWPEN	CMPOSEL			CVRSS	CVROE	CVREN
类型 / 复位	R	0	RW/R	0	RW/R	0	RW/R	0
	7	6	5	4	3	2	1	0
	SYNCSEL	CMPPOL	CMPINSEL		CMPHM		CMPSM	CMPEN
类型 / 复位	RW/R	0	RW/R	0	RW/R	0	RW/R	0

位	字段	描述
[31:16]	PROTECT	寄存器保护 写操作: 0x9C3A: 除能 CMPCR 寄存器写保护 其它值: 使能 CMPCR 寄存器写保护 读操作: 0x0000: CMPCR 寄存器写保护除能 0x0001: CMPCR 寄存器写保护使能 这些位用来使能或者除能 CMPCR 寄存器中 [14:0] 字段的写保护。使能写保护将使 CMPCR 寄存器中的 [14:0] 字段变为只读来防止不可预期的写操作。通过读取该位可以了解写保护功能是否使能。
[15]	CMPSTS	比较器输出状态 0: 输出低电平 1: 输出高电平 该只读位是比较器在极性选择输出状态之后的复制位。
[14]	CMPWPEN	比较器唤醒使能 0: 除能比较器唤醒功能 1: 使能比较器唤醒功能 当比较器的极性选择输出发生上升沿跳变时, 该位将 MCU 从休眠或深度休眠模式中唤醒。



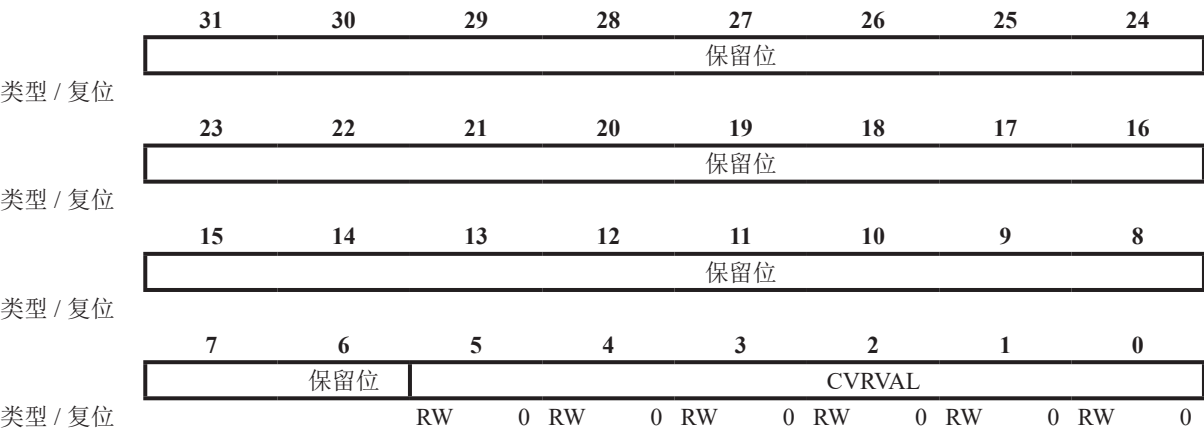
位	字段	描述
[13:11]	CMPOSEL	比较器 0 输出选择 000: 无选择 001: GPTM0 捕捉通道 3 010: MCTM0 捕捉通道 3 011: MCTM0 刹车输入 1 100: ADC 触发输入 其它: 保留位 比较器 1 输出选择 000: 无选择 001: GPTM1 捕捉通道 3 010: MCTM1 捕捉通道 3 011: MCTM1 刹车输入 1 100: ADC 触发输入 其它: 保留位 在比较器输出极性选择后, 这些位选择输出目标。
[10]	CVRSS	比较器电压参考源选择 0: 6-bit 计数器的参考电压源来自 $V_{DDA}$ 1: 6-bit 计数器的参考电压源来自 $V_{REF+}$
[9]	CVROE	比较器电压参考输出使能 0: 除能 6-bit 计数器输出到 CN 引脚 1: 使能 6-bit 计数器输出到 CN 引脚
[8]	CVREN	比较器电压参考使能 0: 除能用于比较器电压参考的 6-bit 计数器 1: 使能用于比较器电压参考的 6-bit 计数器
[7]	SYNCSEL	同步选择 0: 比较器输出异步信号已经选择 1: 比较器输出同步信号已经选择 在传输到 AFIO 单元之前比较器输出同步信号选择。
[6]	CMPPOL	比较器输出极性选择 0: 比较器输出同相 1: 比较器输出反相
[5:4]	CMPINSEL	比较器反相端输入选择 00: 比较器外部 CN 引脚 01: 比较器内部 6-bit 参考电压计数器输出 1x: 保留位 这些位选择比较器反相输入端的来源。
[3:2]	CMPHM	比较器迟滞模式选择 00: 无迟滞模式 01: 低迟滞模式 10: 中迟滞模式 11: 高迟滞模式
[1]	CMPSM	比较器速度模式选择 0: 低速模式 1: 高速模式
[0]	CMPEN	比较器使能 0: 除能比较器 (进入暂停模式) 1: 使能比较器

比较器电压参考值寄存器 n – CVRVALRn, n = 0 或 1

该寄存器用来设置比较器参考电压电平。

偏移量: 0x004 (n = 0), 0x104 (n = 1)

复位值: 0x0000\_0000



位	字段	描述
[5:0]	CVRVAL	比较器电压参考值 有 64 个比较器电压参考电平值，用 CVRVAL 位设置，CVRVAL 寄存器的值和比较器电压参考 CVREF 的关系由下列等式取得： $CVREF = CVRVAL \times (V_{DDA} - V_{SSA}) / 63$ ，如果 CVR 参考电压源选择来自于 $V_{DDA}$

比较器中断使能寄存器 n – CMPIERn, n = 0 或 1

该寄存器用来使能比较器 n 输出转换事件时发生中断。

偏移量: 0x008 (n = 0), 0x108 (n = 1)

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位						CMPRIEN	CMPFIEN
类型 / 复位							RW	0 RW 0

位	字段	描述
[1]	CMPRIEN	比较器输出上升沿中断使能 0: 比较器输出上升沿中断除能 1: 比较器输出上升沿中断使能
[0]	CMPFIEN	比较器输出下降沿中断使能 0: 比较器输出下降沿中断除能 1: 比较器输出下降沿中断使能

比较器转换标志寄存器 n – CMPTFRn, n = 0 或 1

该寄存器包含比较器 n 转换检测使能位和标志位。

偏移量: 0x00C (n = 0), 0x10C (n = 1)

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位						CMPRDEN	CMPFDEN
类型 / 复位						R/W 0	R/W 0
7	6	5	4	3	2	1	0
保留位						CMPRF	CMPFF
类型 / 复位						WC 0	WC 0

位	字段	描述
[9]	CMPRDEN	比较器输出上升沿检测使能 0: 除能比较器输出上升沿检测 1: 使能比较器输出上升沿检测 注: 检测到的比较器边沿转换信号与在极性选择后和由 HCLK 时钟同步的比较器输出状态相同。
[8]	CMPFDEN	比较器输出下降沿检测使能 0: 除能比较器输出下降沿检测 1: 使能比较器输出下降沿检测 注: 检测到的比较器边沿转换信号与在极性选择后和由 HCLK 时钟同步的比较器输出状态相同。
[1]	CMPRF	比较器输出上升沿标志 0: 无比较器输出上升沿发生 1: 比较器输出上升沿发生 当比较器输出上升沿检测使能, 该标志位才有效。 当比较器输出上升沿检测使能时, 该标志位由硬件置高。该位写 1 清零。
[0]	CMPFF	比较器输出下降沿标志 0: 无比较器输出下降沿发生 1: 比较器输出下降沿发生 当比较器输出下降沿检测使能, 该标志位才有效。 当比较器输出下降沿检测使能时, 该标志位由硬件置高。该位写 1 清零。

# 14 通用定时器 (GPTM)

## 简介

通用定时器由一个 16-bit 向上 / 向下计数器、四个 16-bit 捕捉 / 比较寄存器 (CCR)、一个 16-bit 计数器重载寄存器 (CRR) 和几个控制 / 状态寄存器组成。可用于多种用途, 包括通用计时、测量输入信号脉冲宽度或产生输出波形, 如单脉冲或 PWM 输出。GPTM 还内建可处理编码器接口信号的带两个输入口的正交解码器。

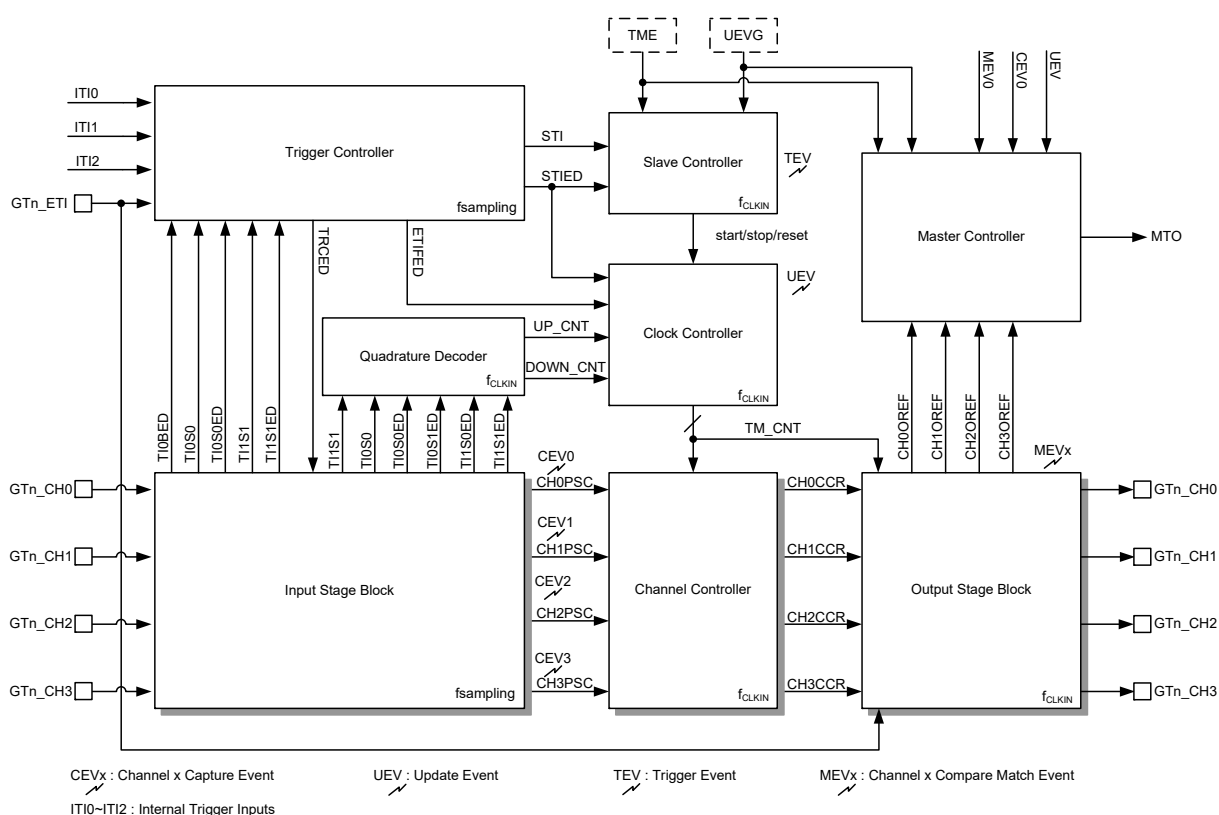


图 38. GPTM 方框图

特性

- 16-bit 向上 / 向下自动重载计数器
- 16-bit 可编程预分频器，可以对计数器时钟按 1 ~ 65536 之间的任意值分频
- 多达 4 个独立通道用于：
  - 输入捕捉功能
  - 比较匹配输出
  - PWM 波形生成 – 边沿对齐计数和中心对齐计数模式
  - 单脉冲模式输出
- 内建可处理编码器接口信号的带两个输入口的正交解码器
- 同步电路使用外部信号控制定时器并使其与其它定时器互相连接
- 下列事件发生时将产生中断 / PDMA：
  - 更新事件
  - 触发事件
  - 输入捕捉事件
  - 输出比较匹配事件
- GPTM 主机 / 从机模式控制器

功能描述

计数器模式

向上计数

在此模式下，计数器从 0 连续向上计数，一直计数到 CRR 寄存器定义的计数器的重载值。一旦计数器的值达到了计数器重载值，定时器模块将发生溢出事件，并从 0 开始重新计数。这一动作会反复执行。向上计数模式时，CNTCFR 寄存器中的计数方向位 DIR 应清零。

当通过设置 EVGR 寄存器中的 UEVG 位为 1 发生更新事件时，计数器的值将被初始化为 0。

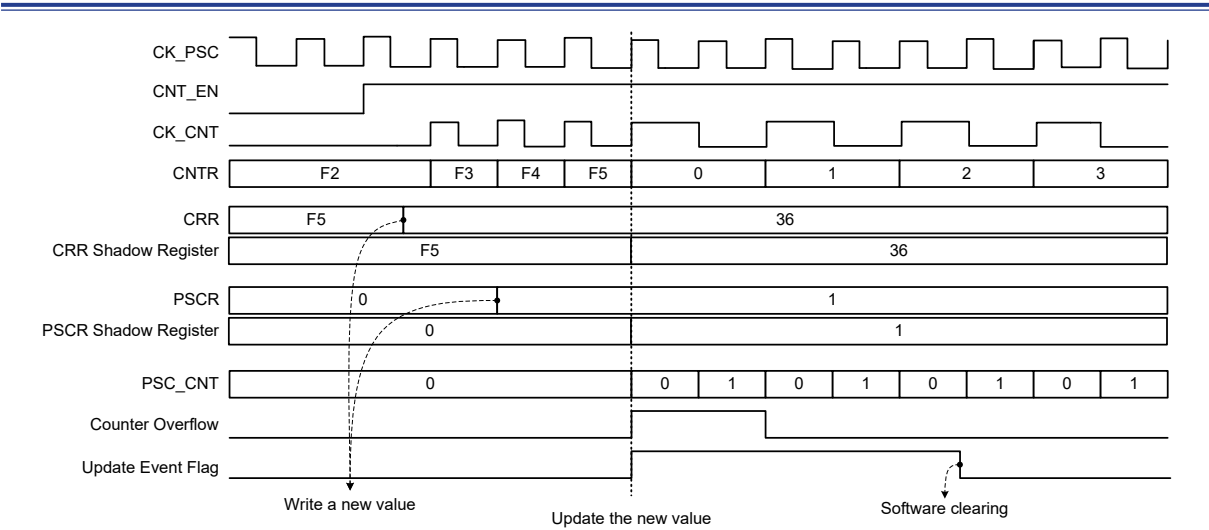


图 39. 向上计数范例

### 向下计数

在此模式下，计数器将从 CRR 寄存器定义的计数器重载值开始向下计数，一直计数到 0。一旦计数器的值达到 0，定时器模块将产生一个下溢事件，并从计数器重载值开始重新计数。这一动作会反复执行。向下计数模式时，CNTCFR 寄存器中的计数方向位 DIR 应置为 1。

当通过设置 EVGR 寄存器中的 UEVG 位为 1 发生更新事件时，计数器的值将被初始化为计数器重载值。

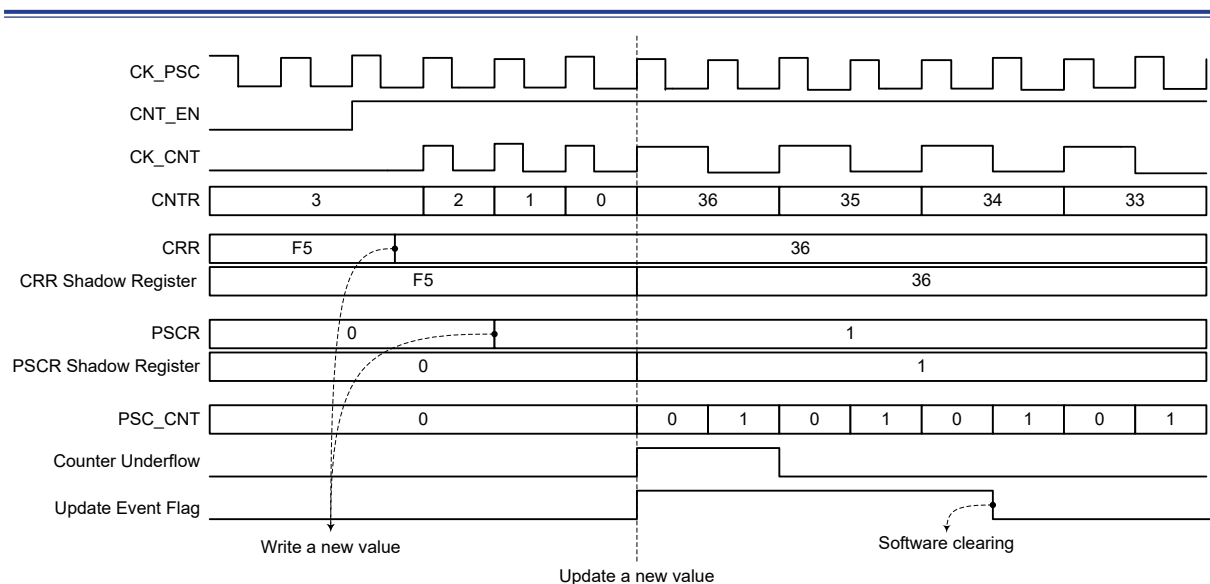


图 40. 向下计数范例

中心对齐计数

在中心对齐计数模式下，计数器将交替地从 0 计数到重载值而后从重载值计数到 0。在向上计数模式下，当计数器计数到计数器重载值时，计数器模块会产生一个上溢事件；而在向下计数模式下，当计数器计数到 0 时，计数器模块会产生一个下溢事件。CNTCFR 寄存器中的 DIR 位是只读位，其值表明了中心对齐计数模式下的计数方向。计数方向由硬件自动更新。

在中心对齐计数模式下，如果将 EVGR 寄存器中的 UEVG 位置位，那么不管计数器是向上计数还是向下计数，计数器的值都将初始化为 0。

当由 CNTCFR 寄存器中的 CMSEL 字段控制的上溢事件、下溢事件或两个事件都发生时，INTSR 寄存器中的 UEVIF 位将被置为 1。

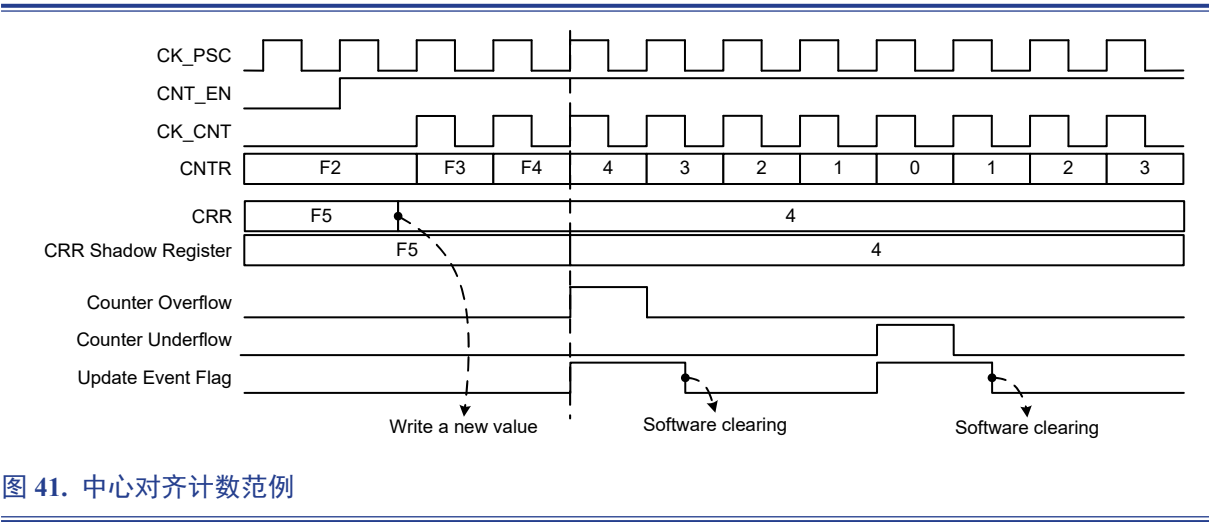


图 41. 中心对齐计数范例



## 时钟控制器

以下描述了定时器模块的时钟控制器，它用来选择内部预分频计数器的时钟源。

### ■ 内部 APB 时钟 $f_{CLKIN}$ :

当从机模式除能时，默认内部时钟源是 APB 时钟  $f_{CLKIN}$ ，用来驱动计数器预分频器。如果通过设置 MDCFR 寄存器中的 SMSEL 字段为 0x1、0x2、0x3 和 0x7 中的一个值来使能从机模式控制器，那么预分频器的时钟将由 TRCFR 寄存器中的 TRSEL 字段设置为其它时钟源，具体描述如下。当从机模式选择位 SMSEL 被置为 0x4、0x5 或 0x6 时，内部 APB 时钟  $f_{CLKIN}$  将作为驱动计数器预分频器的时钟源。

### ■ 正交解码器:

要选择正交解码器模式，应把 MDCFR 寄存器中的 SMSEL 字段设为 0x1、0x2 或 0x3。正交解码器功能通过 GTn\_CH0 和 GTn\_CH1 引脚的输入状态来产生时钟脉冲以驱动计数器预分频器。在每一个输入源信号发生转换时，计数器方向位 DIR 的值都会被硬件自动修改。输入源信号可以只来自于 GTn\_CH0 引脚或 GTn\_CH1 引脚，也可以来自于两个引脚。

### ■ STIED:

计数器预分频器在每一个 STI 信号的上升沿期间计数。此模式可通过将 MDCFR 寄存器中的 SMSEL 字段设为 0x7 来选择。这时计数器将作为一个事件计数器使用。输入事件，即 STI，可通过把 TRSEL 字段设成除 0x0 以外的可用值来选择。当 STI 信号被选择作为时钟源使用时，在每一个 STI 信号上升沿作用期间，内部边沿检测电路将会产生一个时钟脉冲来驱动计数器预分频器。值得注意的是，如果 TRSEL 字段设成 0x0 来选择软件 UEVG 位作为触发源，那么当 SMSEL 字段设成 0x7 时，计数器将会被更新而非计数。

### ■ ETIFED:

在外部引脚 GTn\_ET 的上升沿期间，计数器预分频器将被驱动用来计数。此模式可通过将 TRCFR 寄存器中的 ECME 位设为 1 来选择。将 ETIF 信号选择作为时钟源的方法是通过分别把 SMSEL 字段设成 0x7 和 TRSEL 字段设成 0x3。注意，ETIF 信号来自 GTn\_ETI 引脚的数字滤波器采样。当选择时钟源来自于 GTn\_ETI 信号时，含有边沿检测电路的触发控制器将会在每个 EFTI 信号上升沿作用期间产生一个时钟脉冲来作为计数分频器的时钟源。

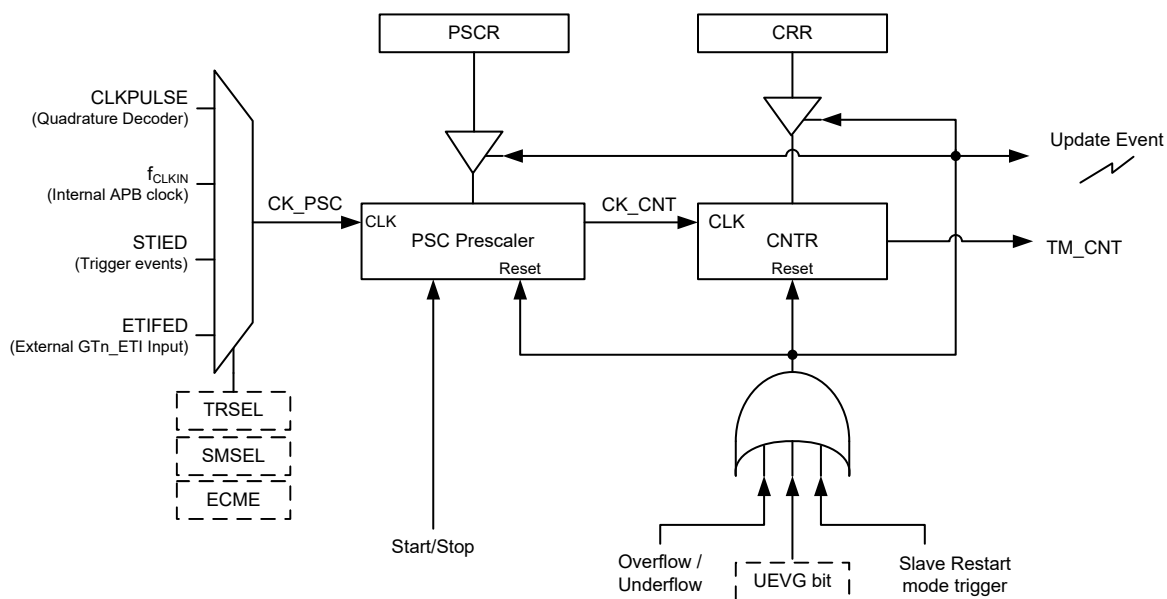
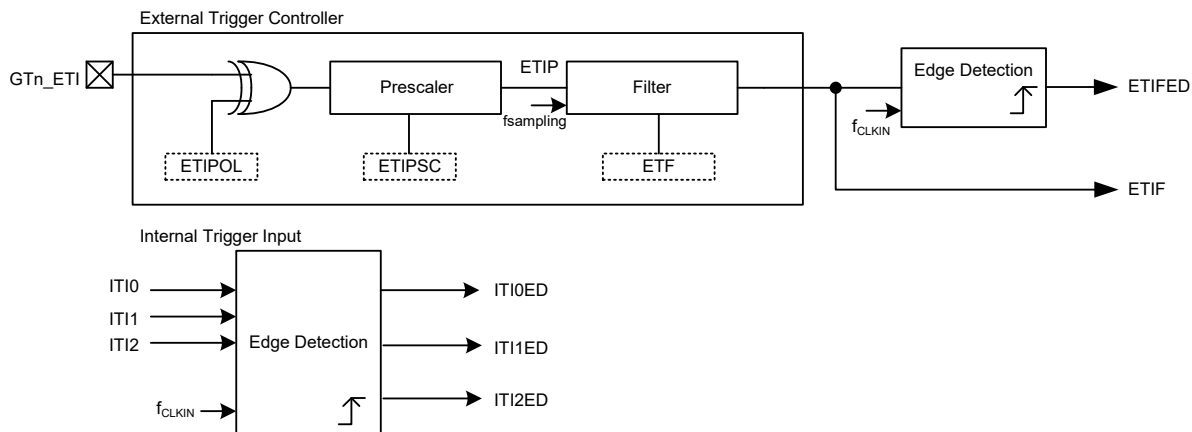


图 42. GPTM 时钟源选择

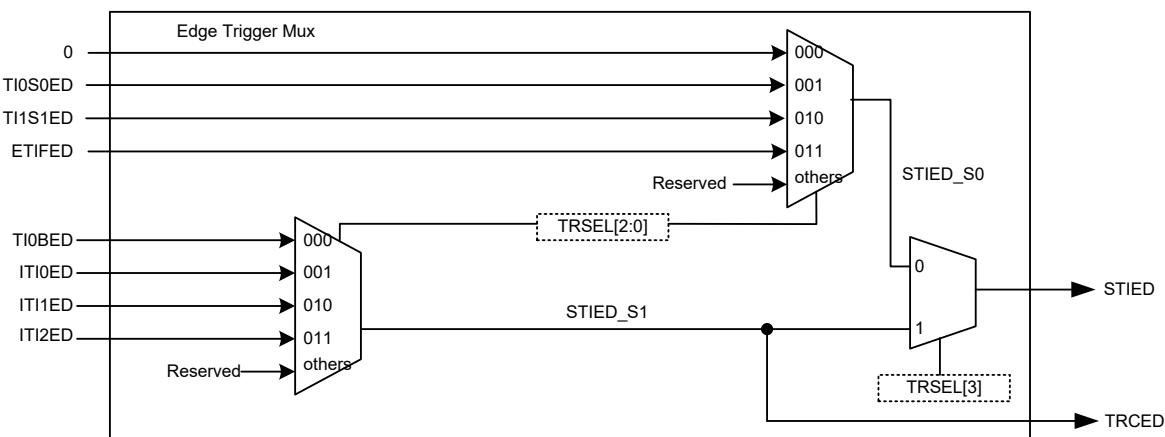
## 触发控制器

触发控制器用来选择触发源以及设置触发电平和边沿触发条件。外部触发输入信号 GTn\_ETI 的有效极性可由 GPTM 触发配置寄存器 TRCFR 中的外部触发极性控制位 ETIPOL 来进行配置。外部触发输入的频率可通过 TRCFR 寄存器中的外部触发预分频位 ETIPSC 来控制分频。对于特殊应用, 如果需要滤波信号时, 外部触发信号也可通过配置 TRCFR 寄存器中的外部触发滤波器 ETF 选择位进行滤波。内部触发输入, 可通过 TRCFR 寄存器中的触发选择位 TRSEL 进行选择。除了 UEVG 位软件触发之外的所有触发源, 内部边沿检测电路将会在每个触发信号上升沿作用期间产生一个时钟脉冲, 以激活某些因触发信号上升沿而触发的 GPTM 功能。

**Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux**



**Edge Trigger = External (ETI)+ Internal (ITIx) + Channel input (CHx) + XOR function**



**Level Trigger Source = External (ETI)+ Internal (ITIx) + Channel input (CHx) + Software UEVG bit**

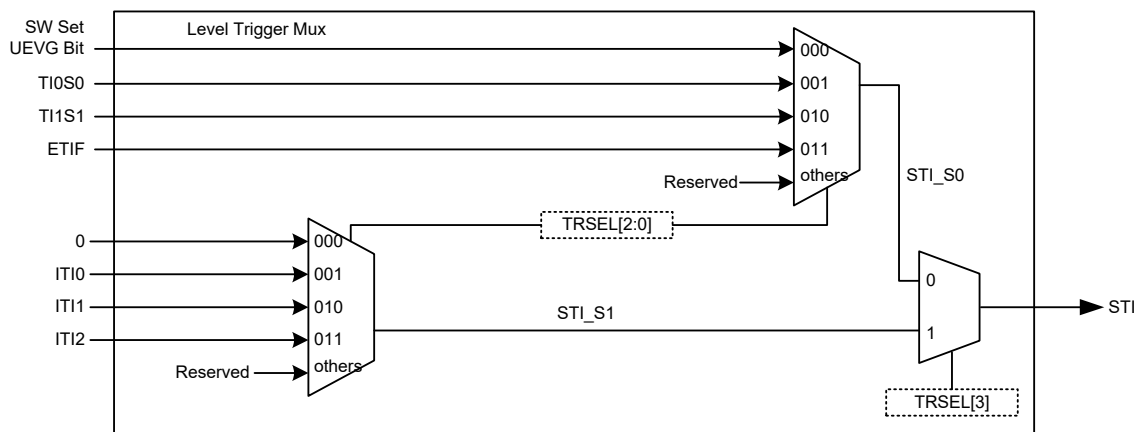


图 43. 触发控制器方框图

从机控制器

在几种模式下，GPTM 可以与一个外部触发器进行同步。这些模式包括重启模式、暂停模式和触发模式，是通过 MDCFR 寄存器中的 SMSEL 字段选择的。这些模式的触发输入来自于 STI 信号，通过 TRCFR 寄存器中的 TRSEL 字段选择。从机控制器中的工作模式在相关章节中有所描述。

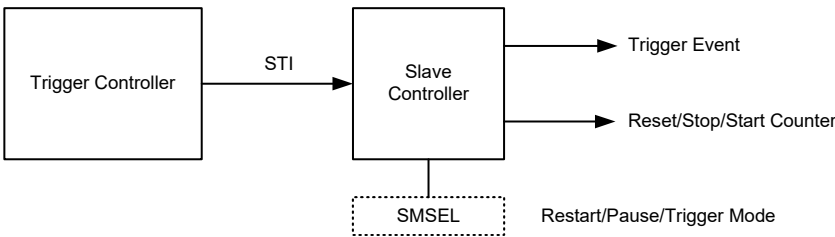


图 44. 从机控制器方框图

重启模式

作为对 STI 信号上升沿的响应，计数器及其预分频器会被重新初始化。当一个 STI 上升沿到来时，更新事件软件产生位 UEVG 将被硬件自动置位，触发事件标志位也将被置位，计数器和预分频器将被重新初始化。虽然 UEVG 位被硬件置 1，但是更新事件还没有真正发生，它取决于更新事件除能控制位 UEVDIS 是否被置 1。如果 UEVDIS 位被置 1 来除能更新事件，那么更新事件将不会发生，然而当 STI 上升沿到来时，计数器和预分频器仍会被重新初始化。如果 CNTCFR 寄存器中的 UEVDIS 位被清零来使能更新事件，则更新事件将伴随 STI 上升沿一起发生，所有预加载的寄存器将被更新。

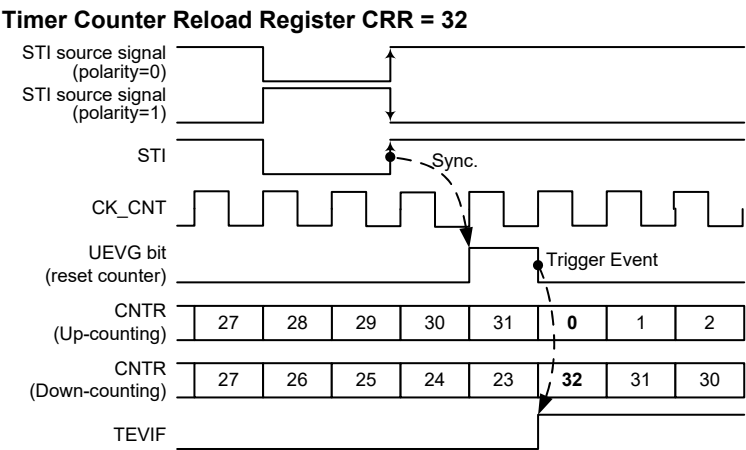


图 45. 重启模式下的 GPTM

### 暂停模式

在暂停模式下, 所选择的 STI 输入信号电平用来控制计数器的开始 / 停止操作。当 STI 信号处于高电平时, 计数器开始计数; 当 STI 信号转换为低电平时, 计数器停止计数并保持当前值不变, 且不会被复位。因为由 STI 电平决定的暂停功能可以控制计数器的开始 / 停止操作, 所以所选择的 STI 信号不能来自于 TI0BED。

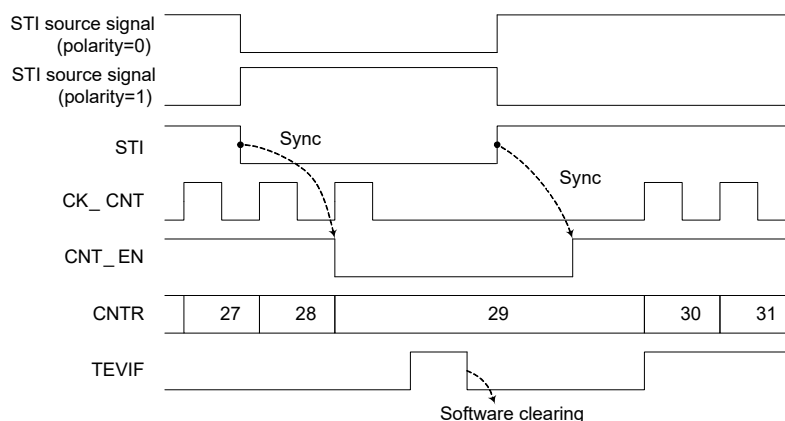


图 46. 暂停模式下的 GPTM

### 触发模式

在计数器停止计数后, 当一个 STI 上升沿信号到来时, 计数器将从当前值重新开始计数。注意, 如果 STI 信号来自于 UEVG 位软件触发, 计数器不会重新计数。当 STI 源信号通过 UEVG 位选择为软件触发时, 不会产生使计数器重新计数的时钟脉冲。还要注意, STI 信号只是用来使计数器重新计数, 而没有使计数器停止计数的作用。

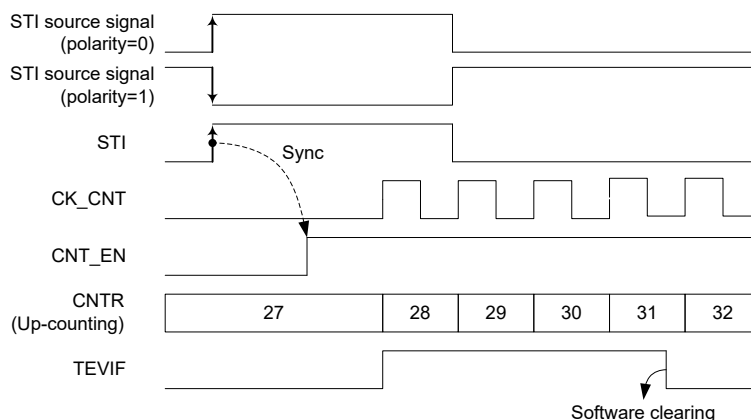


图 47. 触发模式下的 GPTM

## 主机控制器

GPTM 和 MCTM 可在内部连接在一起用于定时器同步或链接。当一个 GPTM 被配置在主机模式下时, GPTM 主机控制器将会产生一个主机触发输出 (MTO) 信号, 通过 MDCFR 寄存器中的 MMSEL 字段选择可以触发或驱动从机模式下的另一个 GPTM 或 MCTM 的定时器复位、启动、停止或为其提供时钟源。

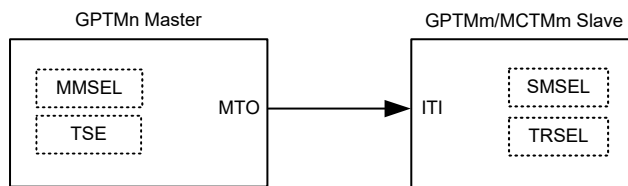


图 48. 主机 GPTMn 和从机 GPTMm/MCTMm 相连接

MDCFR 寄存器中的主机模式选择位 (MMSEL) 用来选择同步另外一个从机 GPTM 或 MCTM 的 MTO 源。

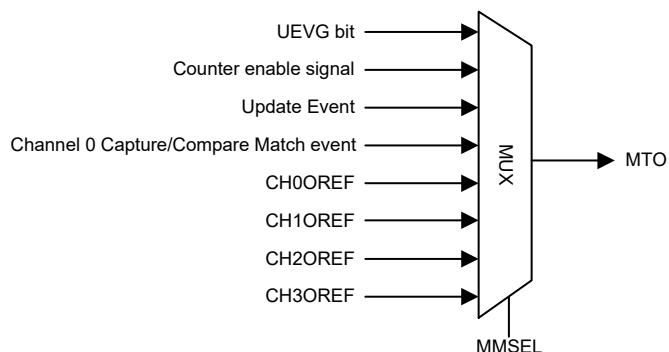


图 49. MTO 选项

例如, 把 MMSEL 字段设为 0x5, 以选择 CH1OREF 信号作为 MTO 信号来同步另外一个从机 GPTM 或 MCTM。欲知更多详细描述, 请参考相关的 MDCFR 寄存器中的 MMSEL 字段定义。

## 通道控制器

GPTM 有四个独立的通道，用来选择作为捕捉输入或比较匹配输出。每个捕捉输入或比较匹配输出通道都由一个预载寄存器和一个影子寄存器组成。APB 总线是通过读 / 写预载寄存器来进行数据访问的。

在输入捕捉模式下使用时，计数器的值会首先被捕捉到 CHxCCR 影子寄存器中，捕捉事件发生时，其值会被传送到 CHxCCR 预载寄存器中。

在比较匹配输出模式下使用时，CHxCCR 预载寄存器的内容会被复制到相应的影子寄存器中，然后计数器的值会与寄存器的值进行比较。

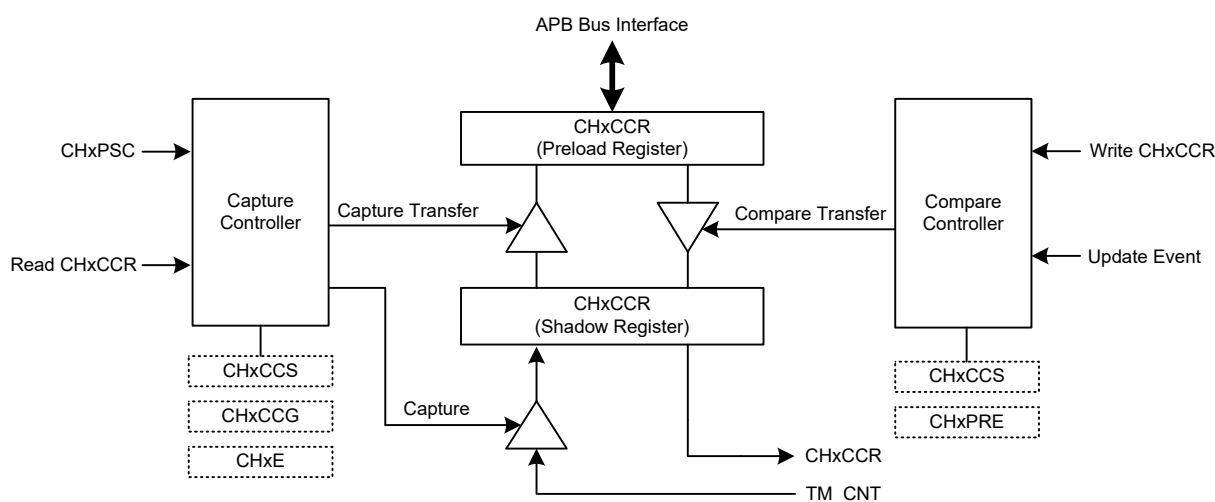


图 50. 捕捉 / 比较方框图

捕捉计数器值传送到 CHxCCR

当通道用作捕捉输入时，计数器的值会在有效输入信号传送发生时，被捕捉到通道捕捉 / 比较寄存器 (CHxCCR) 中。一旦捕捉事件发生，INTSR 寄存器中的 CHxCCIF 标志位会相应的被置位。如果 CHxCCIF 位已经被置位，即标志位还未被软件清零，而此通道的另外一个捕捉事件发生，则相应的通道过度捕捉标志位 CHxOCF 将被置位。

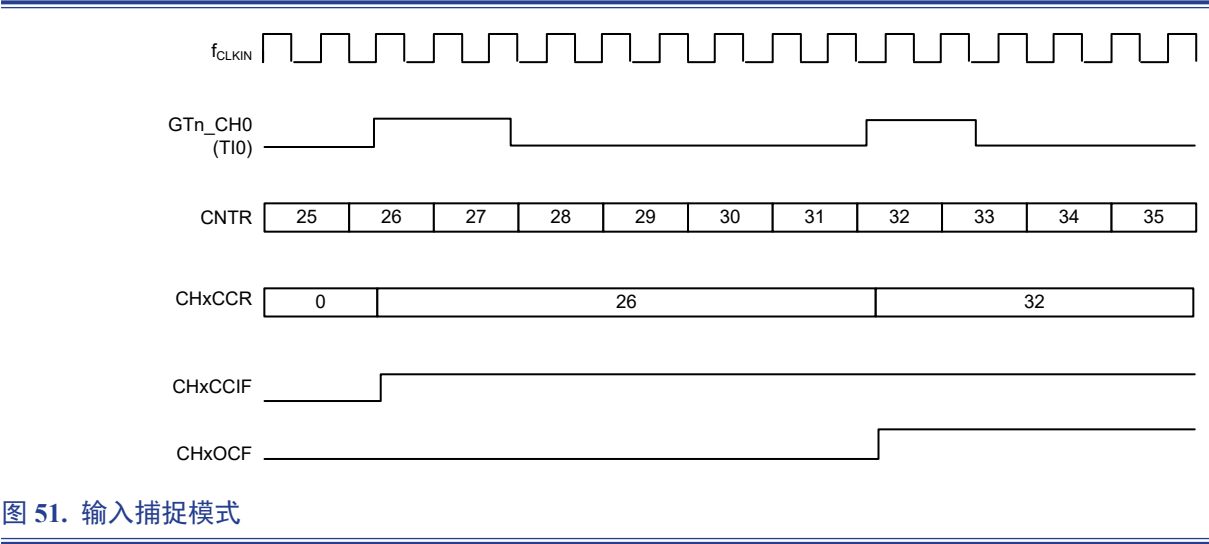


图 51. 输入捕捉模式



### 脉冲宽度测量

输入捕捉模式也可以用于 GTn\_CHx 引脚上的信号 (TIx) 脉冲宽度测量。下面的例子说明了在输入捕捉模式下, 如何配置 GPTMn 使用通道 0 和通道 1 测量在 GTn\_CH0 引脚上的高脉冲宽度和输入周期。基本步骤如下:

- 配置捕捉通道 0 (CH0CCS = 0x1), 选择 TI0 信号作为捕捉输入
- 设置 CH0P 位为 0, 选择 TI0 输入的上升沿作为有效极性
- 配置捕捉通道 1 (CH1CCS = 0x2), 选择 TI0 信号作为捕捉输入
- 设置 CH1P 位为 1, 选择 TI0 输入的下降沿作为有效极性
- 设置 TRSEL 位为 0x1, 选择 TI0S0 作为触发输入
- 把 MDCFR 寄存器的 SMSEL 字段置为 0x4, 使从机控制器工作在重启模式下
- 把 CHCTR 寄存器中的 CH0E 和 CH1E 位置为 1 来开启输入捕捉模式

如下图所示, 输入捕捉操作之后, GTn\_CH0 引脚上的高脉冲宽度将被捕捉到 CH1CCR 寄存器中, 输入周期将被捕捉到 CH0CCR 寄存器中。

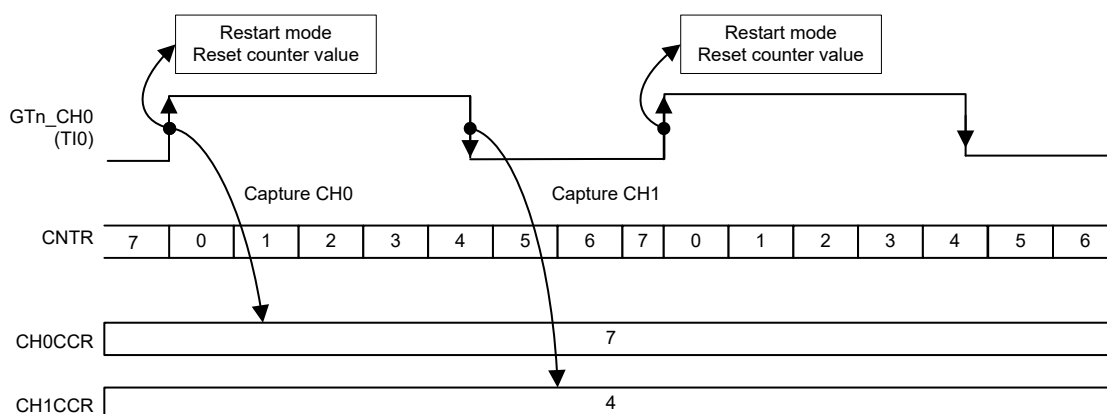


图 52. PWM 脉冲宽度测量范例

## 输入级

输入级由一个数字滤波器、一个通道极性选择器、边沿检测电路和一个通道预分频器组成。通道 0 输入信号  $TI0$  可以来自于  $GTn\_CH0$  信号或  $GTn\_CH0$ 、 $GTn\_CH1$  和  $GTn\_CH2$  信号的异或。通道输入信号  $TIx$  被一个数字滤波器采样，产生一个滤波输入信号  $TIxF$ 。通道极性和边沿检测模块可以产生一个  $TIxS0ED$  或  $TIxS1ED$  信号提供给输入捕捉功能。有效输入事件数量可通过通道捕捉输入源预分频器设置字段  $CHxPSC$  设置。

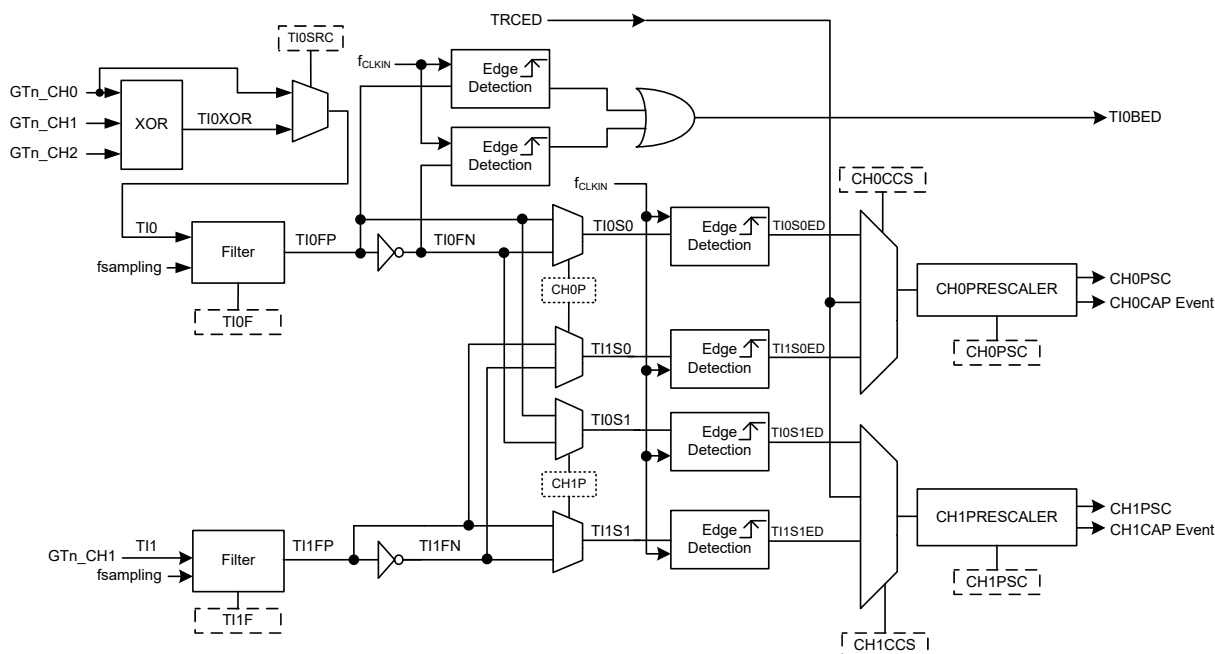


图 53. 通道 0 和通道 1 输入级

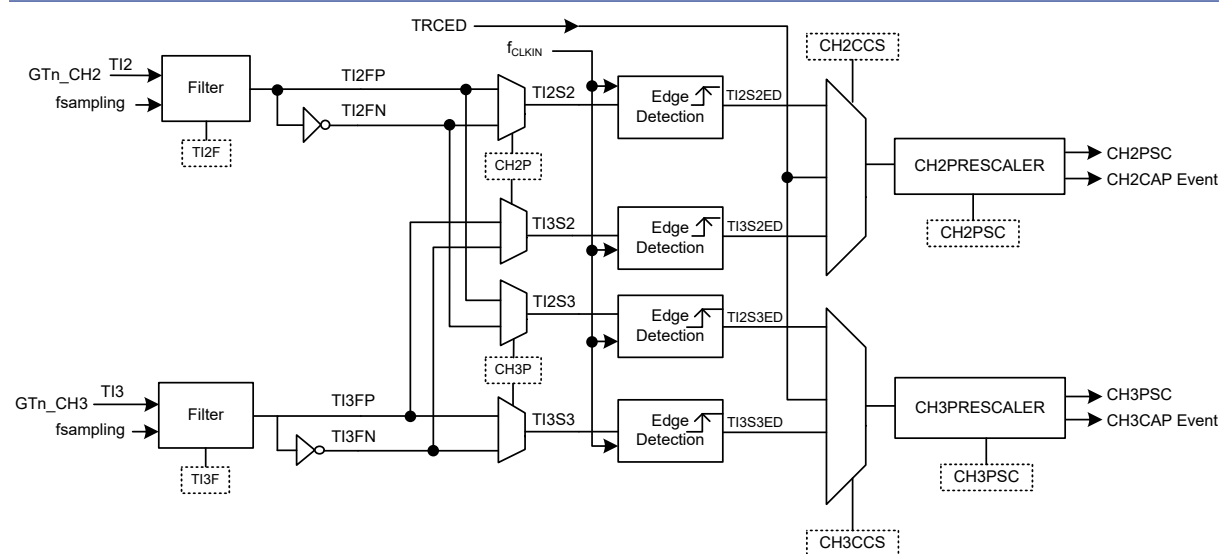


图 54. 通道 2 和通道 3 输入级

## 输出级

GPTM 有四个通道, 用于比较匹配、单脉冲或 PWM 输出功能。通道输出 GTn\_CHx 由 CHxOCFR、CHPOLR 和 CHCTR 寄存器分别对应的 REFxCE、CHxOM、CHxP 和 CHxE 位控制。

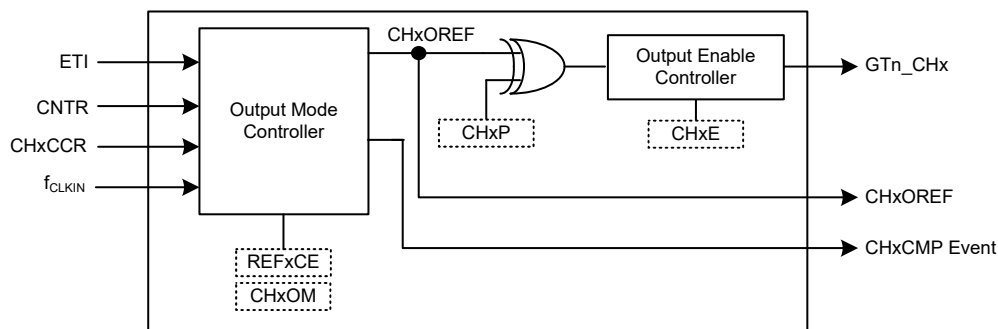


图 55. 输出级方框图

### 通道输出参考信号

当 GPTM 用在比较匹配输出模式时, CHxOREF 信号 (通道 x 输出参考信号) 通过设置 CHxOM 位来定义。CHxOREF 信号有几种输出功能类型, 这些类型包括当计数器的值与 CHxCCR 寄存器的内容匹配时, 通过设置 CHxOM 为 0x00 来保持原始电平, 通过设置 CHxOM 为 0x01 使其清为 0, 通过设置 CHxOM 为 0x02 使其置为 1, 或设置 CHxOM 为 0x03 使信号翻转。

PWM 模式 1 和 PWM 模式 2 输出也是 CHxOREF 输出的一种, 是通过把 CHxOM 字段置为 0x06 / 0x07 来设置的。在这些模式中, CHxOREF 信号电平根据计数方向和计数器的值与 CHxCCR 内容的关系而改变。至于更详细的说明请参考相应位的定义。

CHxOREF 信号另一种特殊的功能是可通过设置 CHxOM 字段为 0x04 / 0x05 使其作为一个强制输出信号。这里, 不论计数器和 CHxCCR 值的比较条件是什么, 输出都可强制为一个无效或有效的电平。

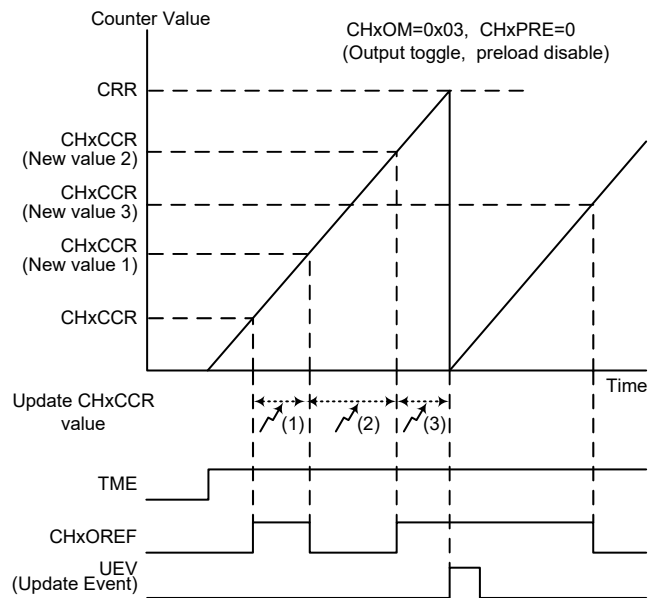


图 56. 翻转模式通道输出参考信号 – CHxPRE = 0

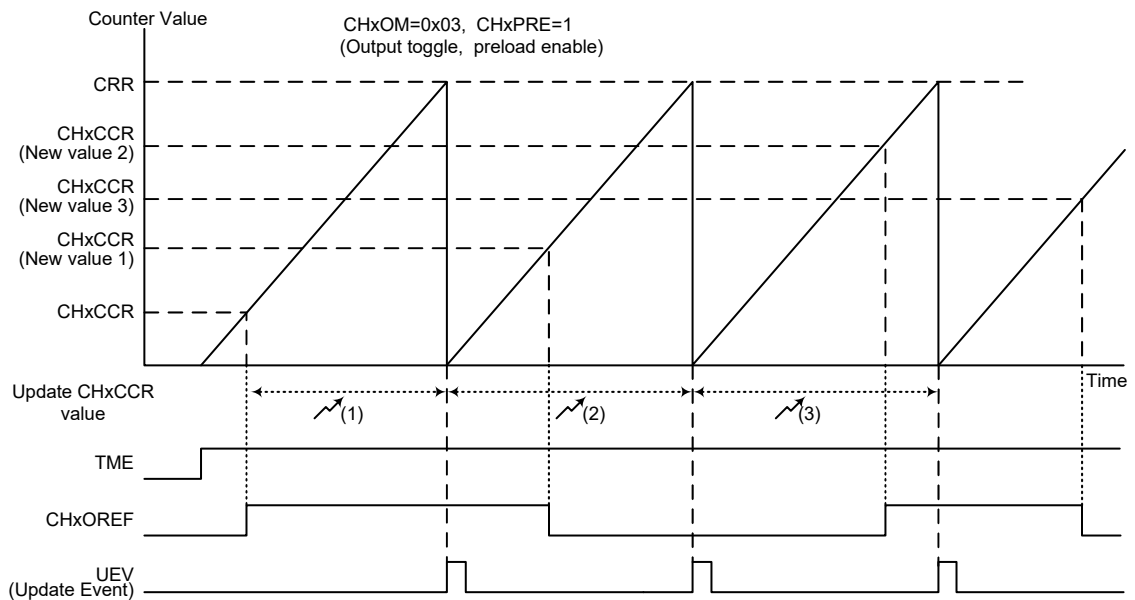


图 57. 翻转模式通道输出参考信号 – CHxPRE = 1

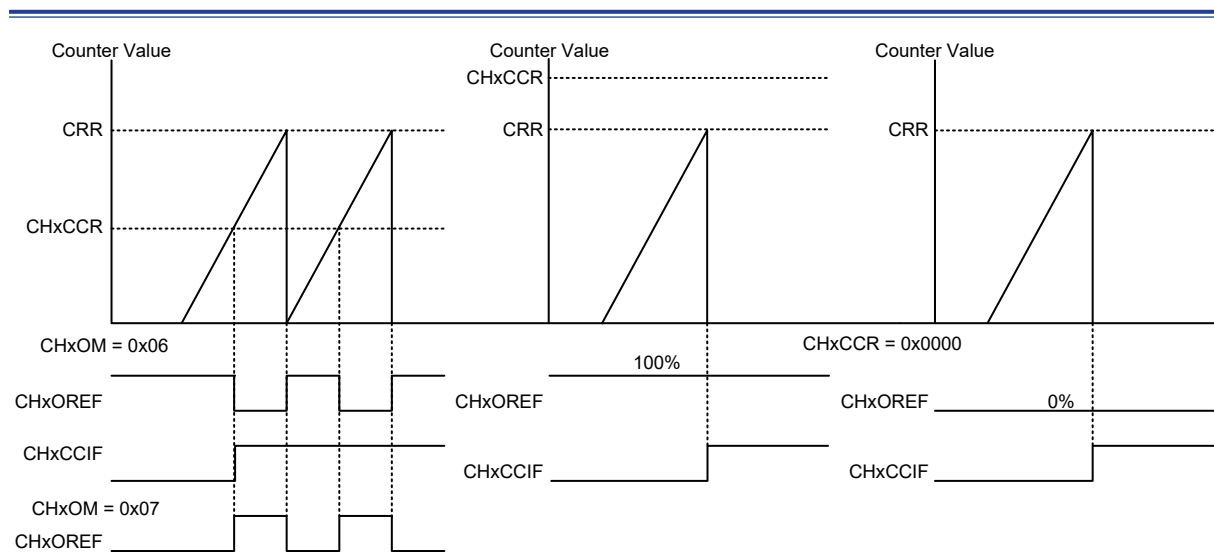


图 58. PWM 模式通道输出参考信号和向上计数模式下的计数器

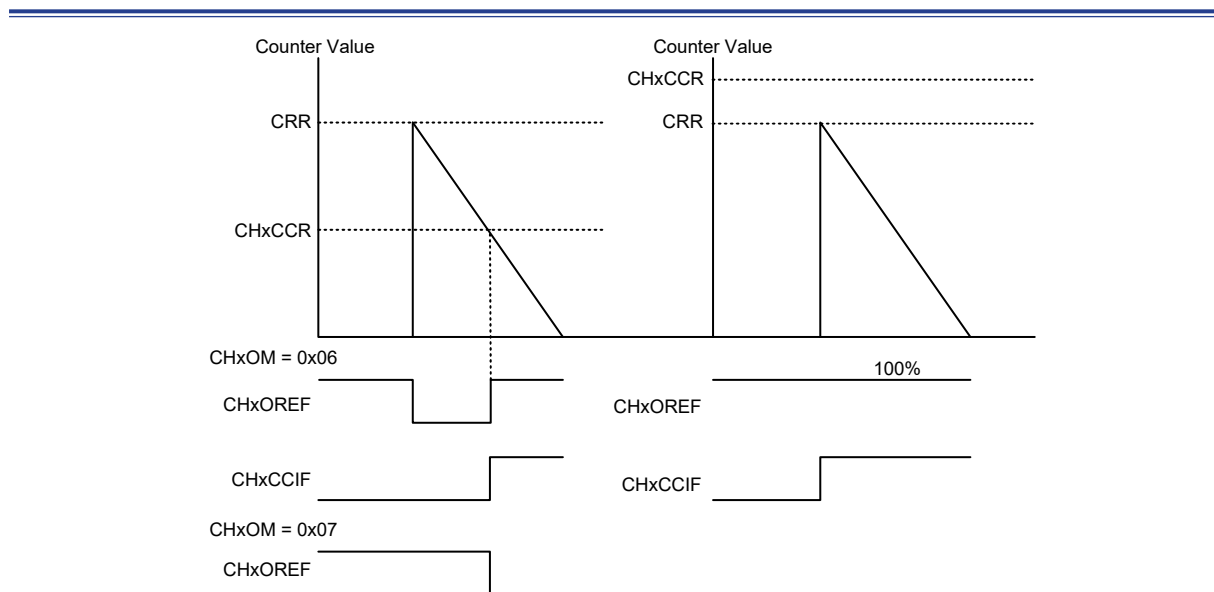


图 59. PWM 模式通道输出参考信号和向下计数模式下的计数器

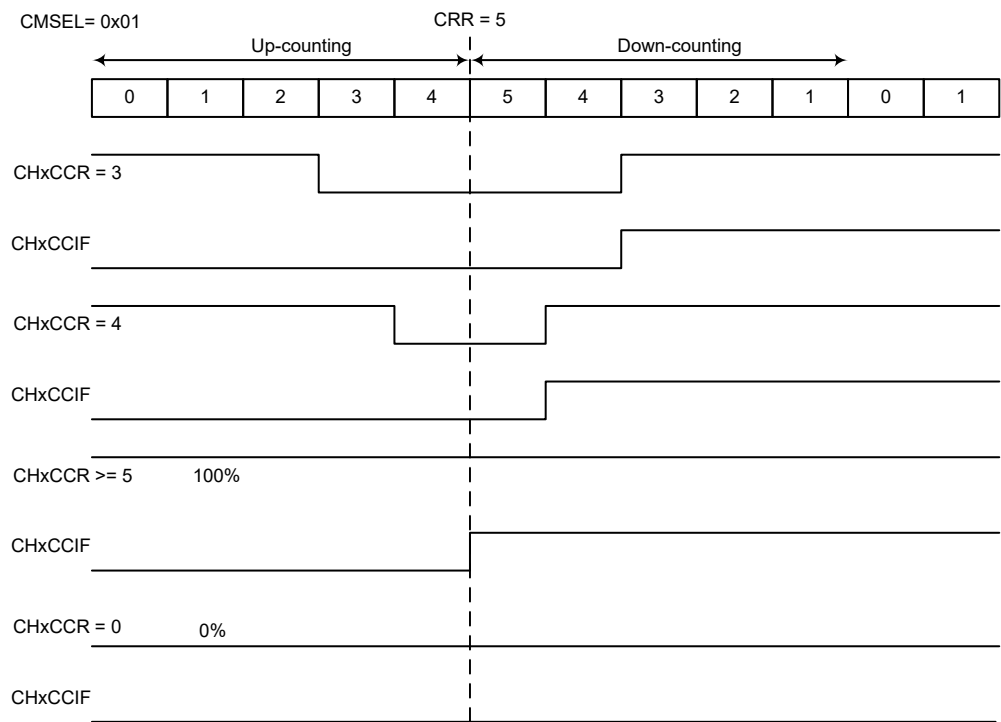


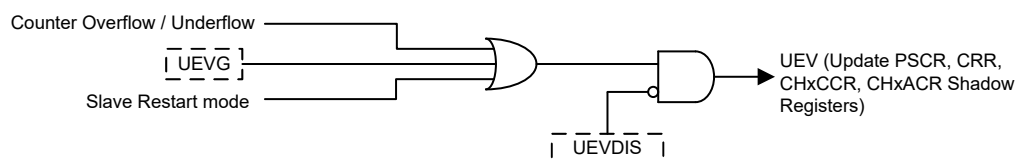
图 60. 模式通道输出参考信号和中心对齐计数模式下的计数器

## 更新管理

更新事件用来把实际寄存器中 CRR、PSCR、CHxACR 和 CHxCCR 的值更新到相应的影子寄存器中。更新事件可在计数器上溢 / 下溢、软件更新控制位被触发或从机控制器的更新事件产生时发生。

CNTCFR 寄存器的 UEVDIS 位控制更新事件是否发生。当更新事件发生，可能会产生相应的更新事件中断，但这取决于更新事件中断功能是否已经通过设置 CNTCFR 寄存器中的 UGDIS 位开启。欲知更多详细信息，请参考 CNTCFR 寄存器中的 UEVDIS 位和 UGDIS 位的相关定义。

### Update Event Management



### Update Event Interrupt Management

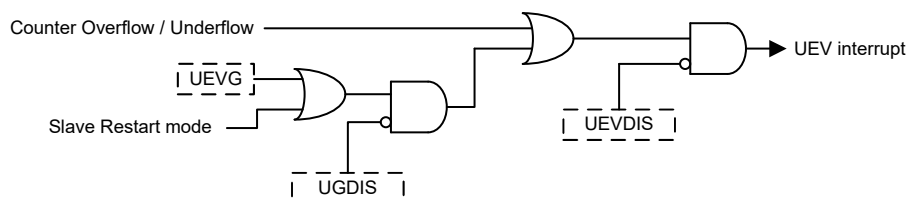


图 61. 更新事件设置方框图

## 正交解码器

正交解码器功能使用两个来自于 GTn\_CH0 和 GTn\_CH1 引脚的正交输入 TI0 和 TI1 进行交互以生成计数器的值。在每个输入源信号转换时，DIR 位会被硬件自动更改。输入源可以是只来自于 TI0 或 TI1，也可来自于 TI0 和 TI1，通过设置 SMSEL 字段为 0x01、0x02 或 0x03 来选择模式。下表列出了改变计数器方向的几种情况。正交解码器可视为一个带有方向性选择的外部时钟。这意味着，计数器将在 0 和计数器重载值之间连续计数。因此，用户必须在计数器开始计数之前设置 CRR 寄存器。

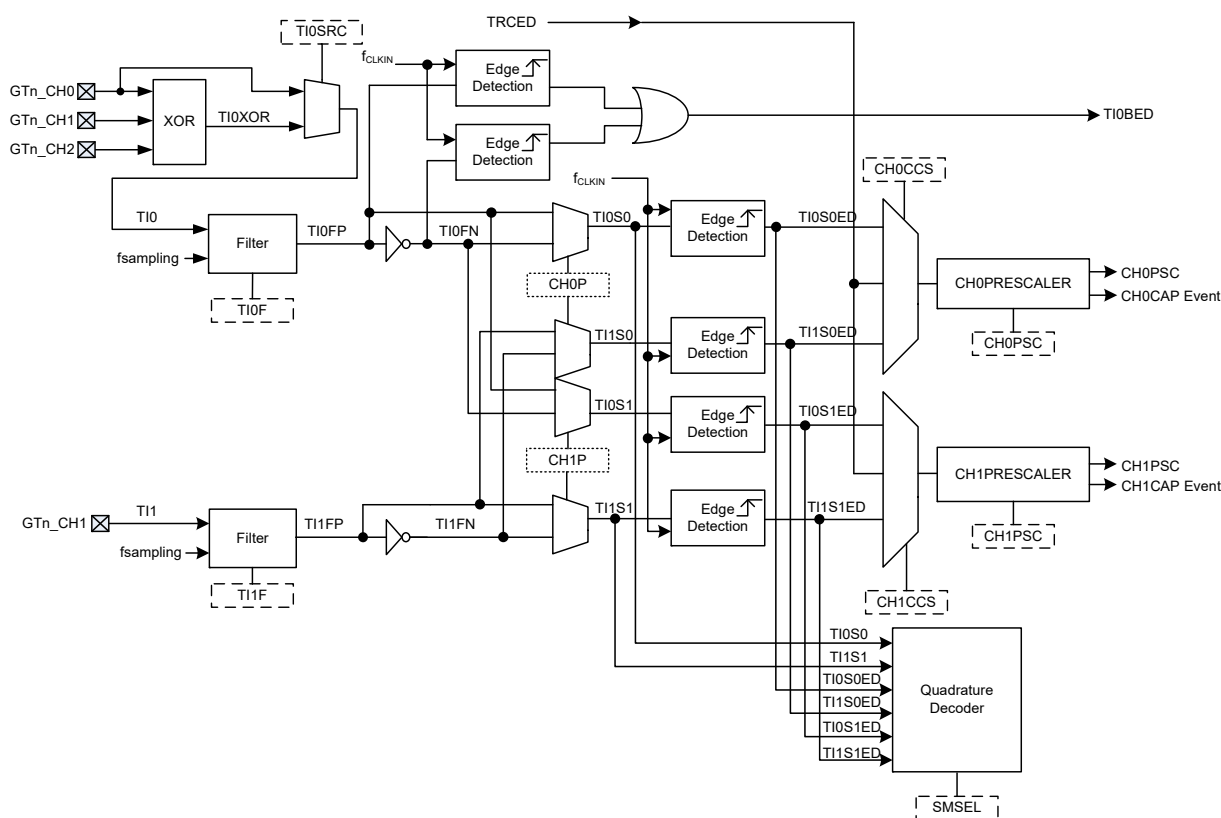


图 62. 输入级和正交解码器方框图



表 32. 计数方向和编码信号

计数模式	电平	TI0S0		TI1S1	
		上升沿	下降沿	上升沿	下降沿
仅 TI0 计数 (SMSEL = 0x01)	TI1S1 = 高	向下	向上	—	—
	TI1S1 = 低	向上	向下	—	—
仅 TI1 计数 (SMSEL = 0x02)	TI0S0 = 高	—	—	向上	向下
	TI0S0 = 低	—	—	向下	向上
TI0 和 TI1 计数 (SMSEL = 0x03)	TI1S1 = 高	向下	向上	X	X
	TI1S1 = 低	向上	向下	X	X
	TI0S0 = 高	X	X	向上	向下
	TI0S0 = 低	X	X	向下	向上

“—”：没有计数；“X”：不可能

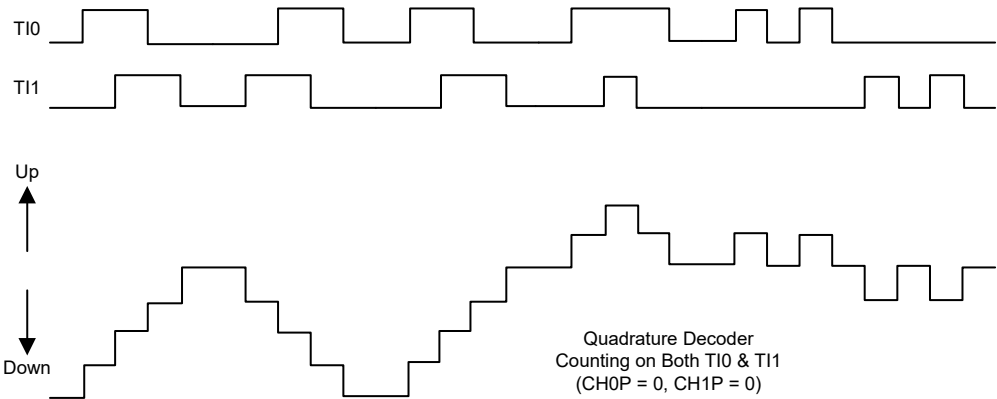


图 63. TI0 和 TI1 正交解码器计数

## 数字滤波器

数字滤波器嵌入在输入级和时钟控制器，分别用于 GTn\_CH0 ~ GTn\_CH3 引脚以及 GTn\_ETI 引脚。GPTM 中的数字滤波器是一个 N 事件计数器，N 指的是能够输出一个滤波信号所需的有效转换次数。N 的值可以是 0、2、4、5、6 或 8，是由用户针对每个滤波器选择的。

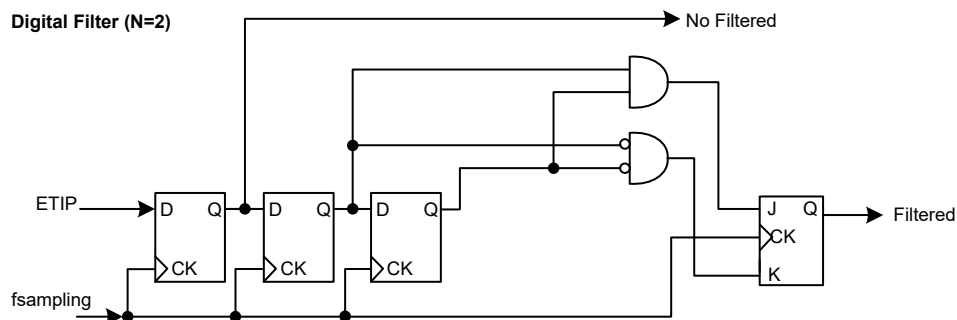


图 64. N=2 的 GTn\_ETI 引脚数字滤波器方框图

## ETIF 为高电平时清除 CHxOREF

当 ETIF 信号来自外部 GTn\_ETI 引脚且把 CHxOCFR 寄存器中的 REFxCE 位置 1，使 ETIF 信号为高电平时，CHxOREF 信号会被强制置为 0。直到下一个更新事件发生时，CHxOREF 信号才会回到其有效电平。

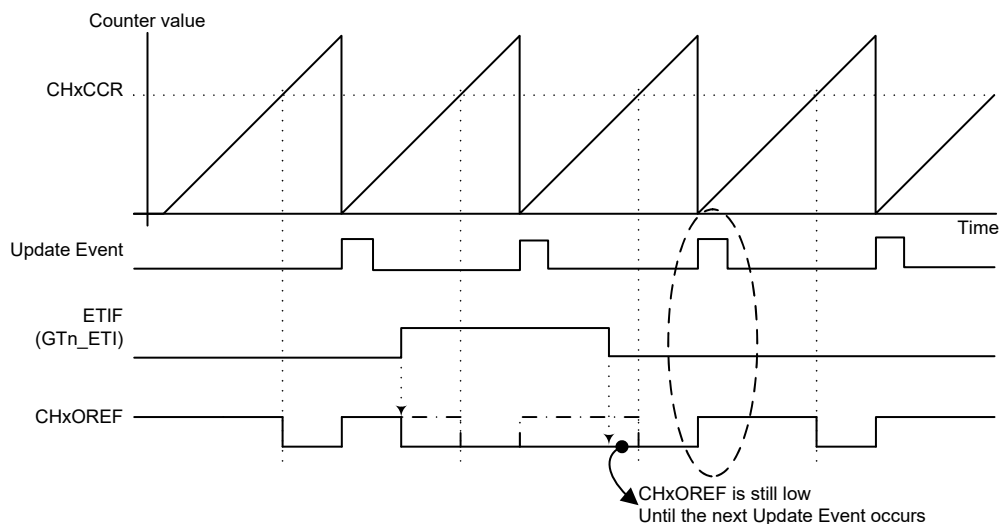


图 65. 通过 ETIF 清除 CHxOREF

## 单脉冲模式

一旦定时器被设置工作在单脉冲模式下，则无需把 CTR 寄存器中的定时器使能位 TME 置为 1 来使能定时器。当 STI 信号上升沿发生时或通过软件把 TME 位设置为 1 时，触发器将会产生一个脉冲。然后一直保持 TME 位为高电平直到更新事件发生或使用软件将 TME 清零。如果使用软件使 TME 位清零，计数器将停止且保持当前值不变。如果 TME 位是由硬件的更新事件自动清零，计数器将被重新初始化。

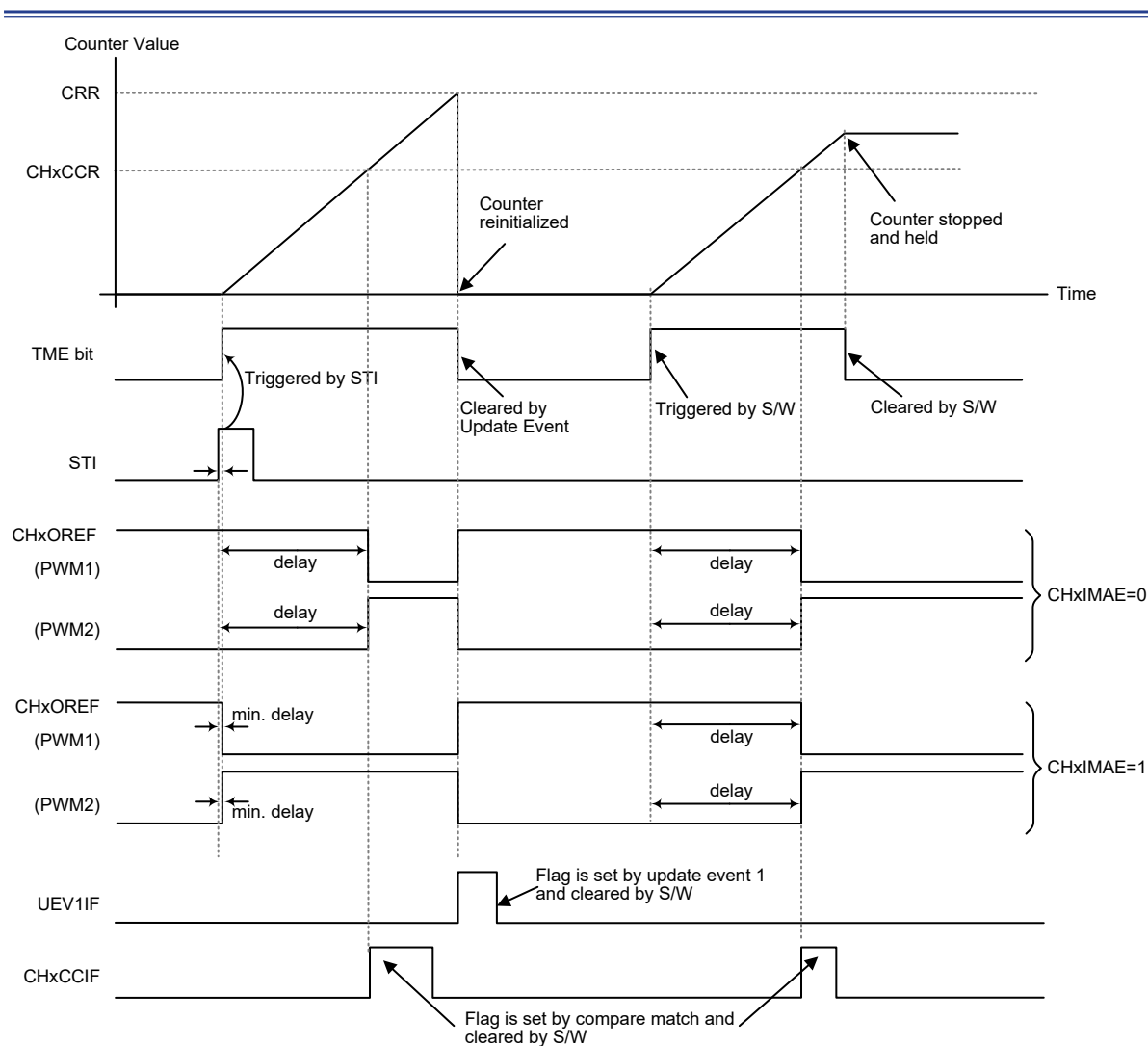


图 66. 单脉冲模式

在单脉冲模式下, STI 有效边沿使 TME 置为 1 时, 将使能计数器。然而, 由于要执行计数器值和 CHxCCR 值的比较结果, 会存在几个时钟延迟。用户可以通过设置 CHxOCFR 寄存器中的 CHxIMAE 位来使延迟时间减小。单脉冲模式下, STI 上升沿触发发生后, CHxOREF 信号将立即强制转变为使比较匹配事件发生的电平, 而无需考虑比较结果为何。只有当输出通道被配置工作在 PWM 模式 1 或 PWM 模式 2 下且触发源来自于 STI 信号时, CHxIMAE 位才可用。

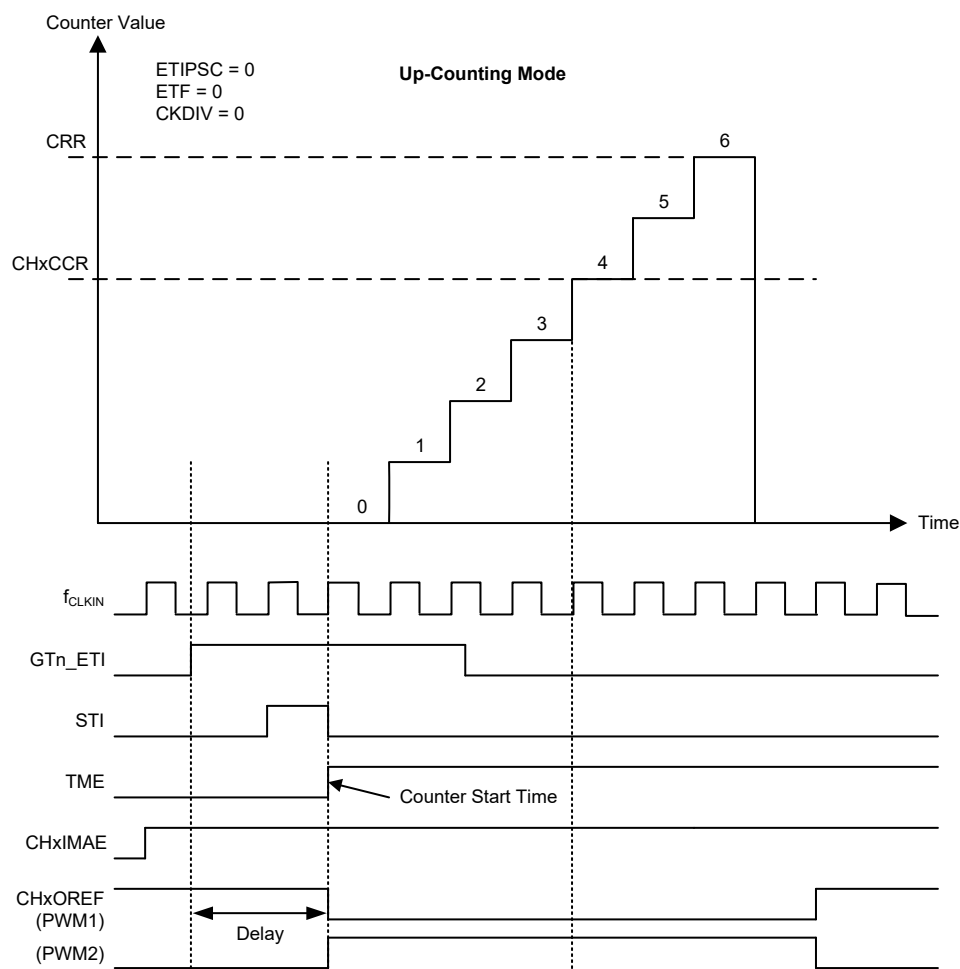


图 67. 立即有效模式的延迟

## 非对称 PWM 模式

非对称 PWM 模式允许两个中心对齐的 PWM 信号产生一个可编程的相位偏移。PWM 频率由 CRR 寄存器的值决定，占空比和相移由 CHxCCR 和 CHxACR 寄存器决定。当计数器向上计数时，PWM 使用 CHxCCR 的值作为向上计数的比较值。当计数器进入向下计数阶段，PWM 使用 CHxACR 的值作为向下计数的比较值。图 68 是在中心对齐计数模式下的非对称 PWM 模式的一个例子。

注：非对称 PWM 模式只能在中心对齐计数模式下运行。

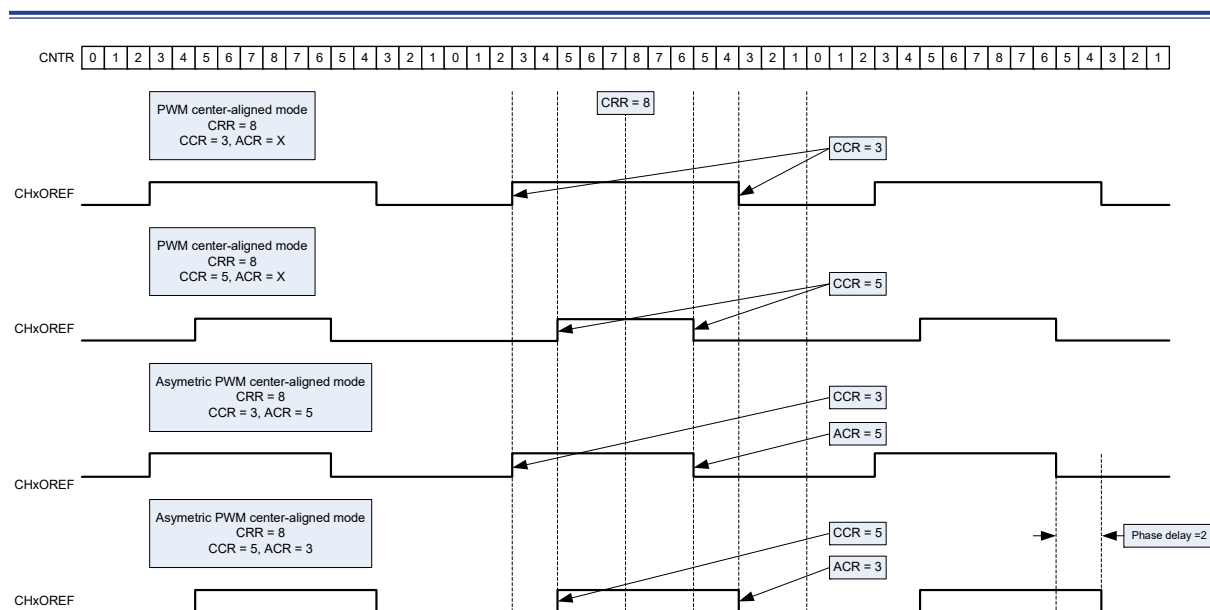


图 68. 中心对齐计数模式下的非对称 PWM 模式

定时器互连

定时器可以在内部互相连接用于使定时器链接和同步。它可通过配置一个定时器工作在主机模式，而配置另一个定时器工作在从机模式来实现。下面是主 / 从模式触发器选择的几个示例。

使用一个定时器使能或除能另外一个定时器使其开始或停止计数

- 使 GPTM0 工作在主机模式，发送通道 0 输出参考信号 CH0OREF 作为触发器输出 (MMSEL = 0x04)
- 配置 GPTM0 CH0OREF 波形
- 使 GPTM1 接收的输入触发器源来自于 GPTM0 触发器输出信号 (TRSEL = 0x09)
- 使 GPTM1 工作在暂停模式下 (SMSEL = 0x05)
- 通过向 TME 位写入 1 来使能 GPTM1
- 通过向 TME 位写入 1 来使能 GPTM0

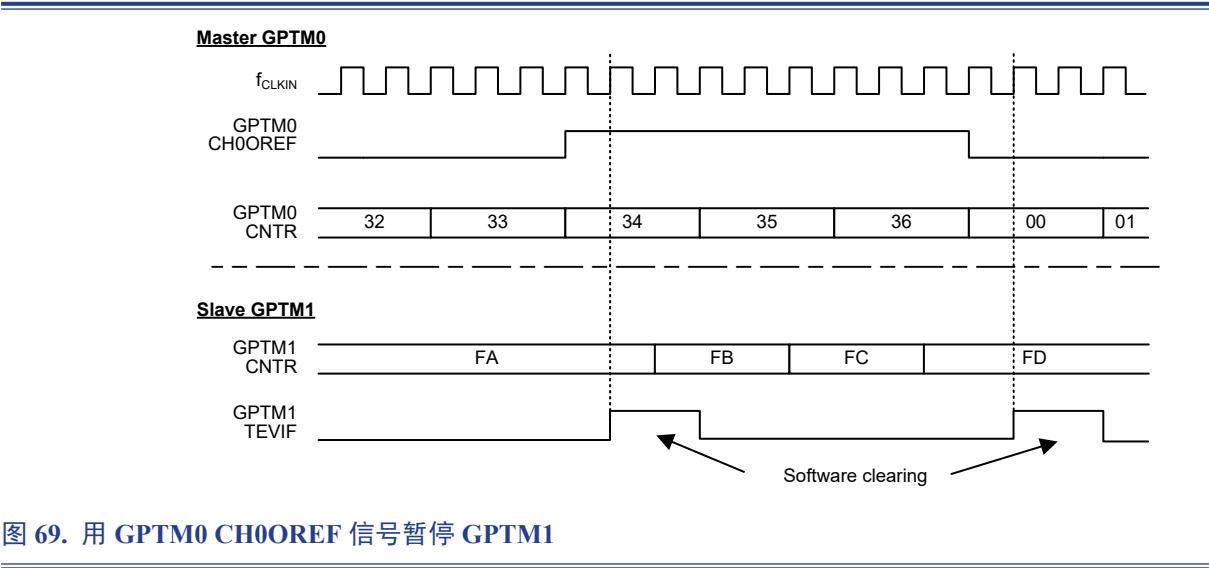


图 69. 用 GPTM0 CH0OREF 信号暂停 GPTM1

#### 使用一个定时器触发另一个定时器开始计数

- 使 GPTM0 工作在主机模式，发送更新事件 UEV 作为触发输出 (MMSEL = 0x02)
- 通过设置 CRR 寄存器配置 GPTM0 周期
- 使 GPTM1 从 GPTM0 触发输出信号获得输入触发源 (TRSEL = 0x09)
- 使 GPTM1 工作在从机触发模式 (SMSEL = 0x06)
- 向 TME 位写入 1 启动 GPTM0

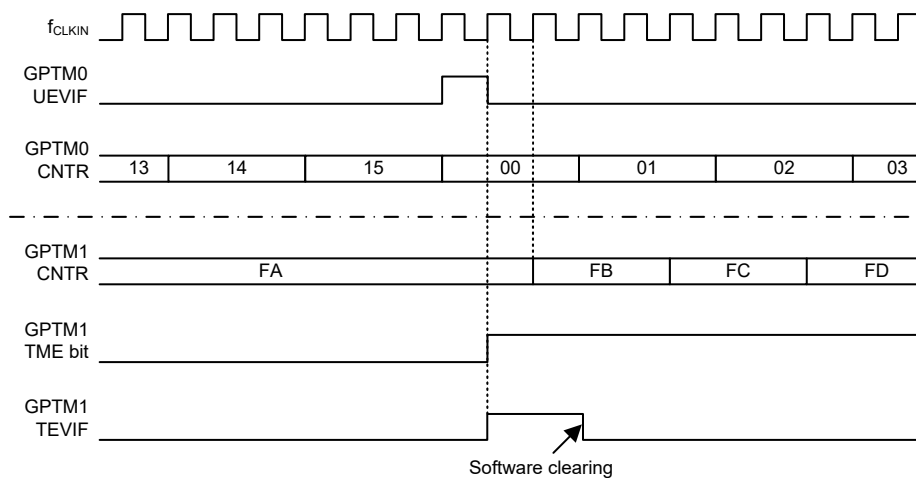


图 70. 用 GPTM0 更新事件触发 GPTM1

#### 启动两个定时器同步响应外部触发

- 使 GPTM0 工作在主机模式下，发送使能信号作为一个触发输出信号 (MMSEL = 0x01)
- 使 GPTM0 工作在从机模式下，从 GTn\_CH0 引脚获得其输入触发源 (TRSEL = 0x01)
- 使 GPTM0 工作在从机触发模式 (SMSEL = 0x06)
- 设置 MDCFR 寄存器中的 TSE 位为 1，使能 GPTM0 主机定时器同步功能使其与从机定时器同步
- 使 GPTM1 接收来自于 GPTM0 的触发输出信号作为其输入触发源 (TRSEL = 0x09)
- 使 GPTM1 工作在从机触发模式下 (SMSEL = 0x06)

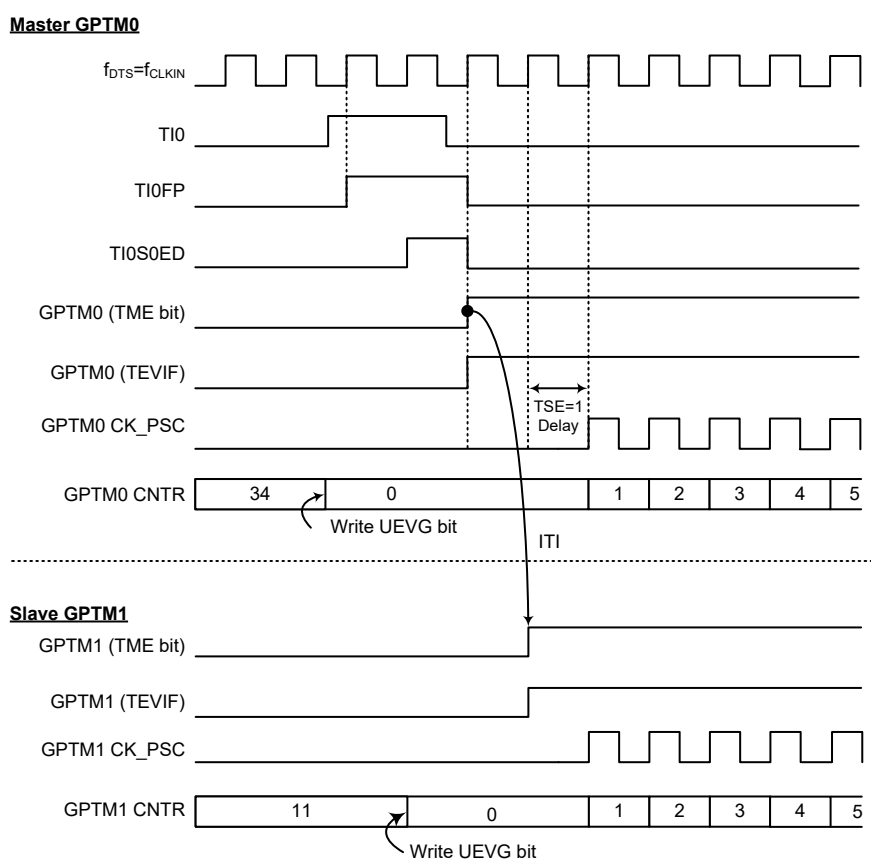


图 71. 用 GPTM0 CH0 输入信号触发 GPTM0 和 GPTM1



## 触发 ADC 开启

为了与模拟 / 数字转换器相连接, GPTM 可以输出 MTO 信号或通道比较匹配输出信号 CHxOREF (x = 0 ~ 3) 作为模拟 / 数字转换器的输入触发信号。

## PDMA 请求

GPTM 支持 PDMA 数据传送的接口。如果相应的使能控制位置 1 使能 PDMA 访问, 那么有一些事件能产生 PDMA 请求。这些事件可以是 GPTM 更新事件、触发事件和通道捕捉 / 比较匹配事件。当 GPTM 通道产生 PDMA 请求时, 对于所有通道都可通过通道 PDMA 选择位 CHCCDS 选择是来自通道捕捉 / 比较匹配事件或 GPTM 更新事件。更多 PDMA 详细配置信息, 请参考 PDMA 的相关章节。

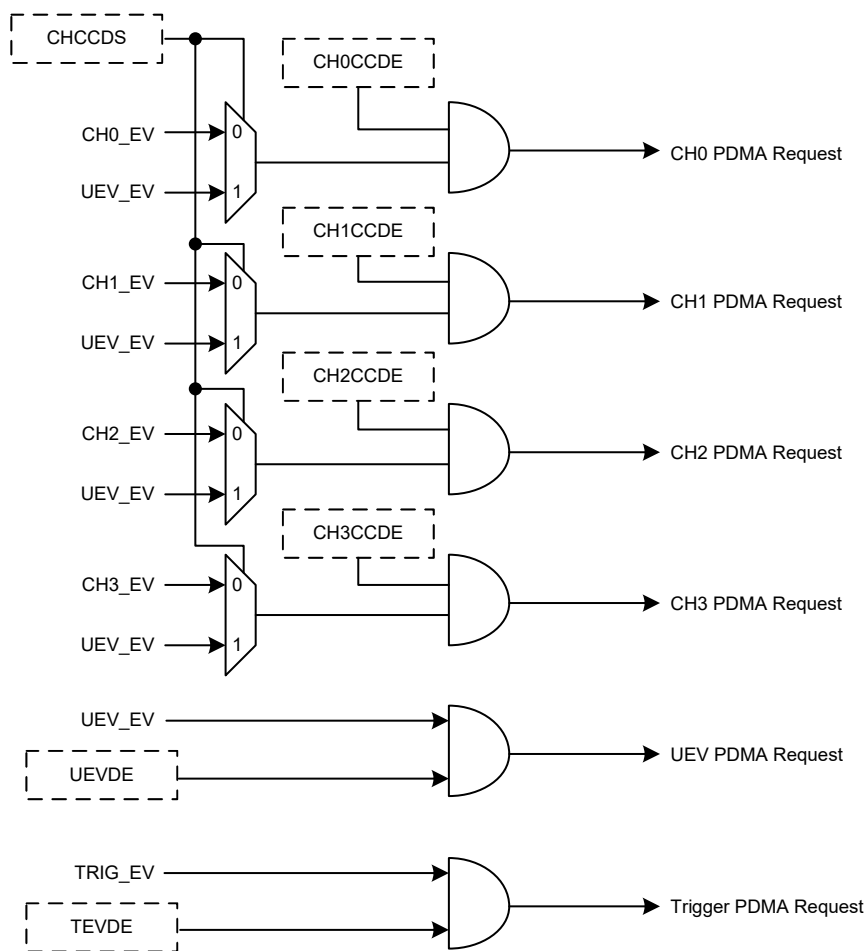


图 72. GPTM PDMA 映射图

## 寄存器列表

下表显示了 GPTM 寄存器及其复位值。

表 33. GPTM 寄存器列表

寄存器	偏移量	描述	复位值
CNTCFR	0x000	定时器计数器配置寄存器	0x0000_0000
MDCFR	0x004	定时器模式配置寄存器	0x0000_0000
TRCFR	0x008	定时器触发配置寄存器	0x0000_0000
CTR	0x010	定时器控制寄存器	0x0000_0000
CH0ICFR	0x020	通道 0 输入配置寄存器	0x0000_0000
CH1ICFR	0x024	通道 1 输入配置寄存器	0x0000_0000
CH2ICFR	0x028	通道 2 输入配置寄存器	0x0000_0000
CH3ICFR	0x02C	通道 3 输入配置寄存器	0x0000_0000
CH0OCFR	0x040	通道 0 输出配置寄存器	0x0000_0000
CH1OCFR	0x044	通道 1 输出配置寄存器	0x0000_0000
CH2OCFR	0x048	通道 2 输出配置寄存器	0x0000_0000
CH3OCFR	0x04C	通道 3 输出配置寄存器	0x0000_0000
CHCTR	0x050	通道控制寄存器	0x0000_0000
CHPOLR	0x054	通道极性配置寄存器	0x0000_0000
DICTR	0x074	定时器 PDMA/ 中断控制寄存器	0x0000_0000
EVGR	0x078	定时器事件发生器寄存器	0x0000_0000
INTSR	0x07C	定时器中断状态寄存器	0x0000_0000
CNTR	0x080	定时器计数器寄存器	0x0000_0000
PSCR	0x084	定时器预分频器寄存器	0x0000_0000
CRR	0x088	定时器计数器重载寄存器	0x0000_FFFF
CH0CCR	0x090	通道 0 捕捉 / 比较寄存器	0x0000_0000
CH1CCR	0x094	通道 1 捕捉 / 比较寄存器	0x0000_0000
CH2CCR	0x098	通道 2 捕捉 / 比较寄存器	0x0000_0000
CH3CCR	0x09C	通道 3 捕捉 / 比较寄存器	0x0000_0000
CH0ACR	0x0A0	通道 0 非对称比较寄存器	0x0000_0000
CH1ACR	0x0A4	通道 1 非对称比较寄存器	0x0000_0000
CH2ACR	0x0A8	通道 2 非对称比较寄存器	0x0000_0000
CH3ACR	0x0AC	通道 3 非对称比较寄存器	0x0000_0000

## 寄存器描述

### 定时器计数器配置寄存器 – CNTCFR

该寄存器定义了 GPTM 计数器配置。

偏移量: 0x000

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							DIR
							RW 0
23	22	21	20	19	18	17	16
保留位						CMSEL	
						RW 0	RW 0
15	14	13	12	11	10	9	8
保留位						CKDIV	
						RW 0	RW 0
7	6	5	4	3	2	1	0
保留位						UGDIS	UEVDIS
						RW 0	RW 0

位	字段	描述
[24]	DIR	计数方向 0: 向上计数 1: 向下计数 注: 当定时器工作在中心对齐计数模式下或作为正交解码器使用时, 此位为只读位。
[17:16]	CMSEL	计数器模式选择 00: 边沿对齐计数模式。此模式下可正常向上和向下计数。计数方向由 DIR 位定义。 01: 中心对齐计数模式 1。计数器向上向下交替计数。比较匹配中断标志位在向下计数期间被置位。 10: 中心对齐计数模式 2。计数器向上向下交替计数。比较匹配中断标志位在向上计数期间被置位。 11: 中心对齐计数模式 3。计数器向上向下交替计数。比较匹配中断标志位在向上和向下计数期间被置位。
[9:8]	CKDIV	时钟分频 00: $f_{DTS} = f_{CLKIN}$ 01: $f_{DTS} = f_{CLKIN} / 2$ 10: $f_{DTS} = f_{CLKIN} / 4$ 11: 保留位 这两位定义了定时器时钟 ( $f_{CLKIN}$ ) 和死区时钟 ( $f_{DTS}$ ) 之间的频率比值。死区时钟也可作为数字滤波器采样时钟使用。
[1]	UGDIS	更新事件中断产生除能控制 0: 以下任何一个事件都可产生一个更新 PDMA 请求或中断 – 计数器上溢 / 下溢 – 设置 UEVG 位 – 通过从机模式产生更新 1: 只在计数器上溢 / 下溢时产生一个更新 PDMA 请求或中断

位	字段	描述
[0]	UEVDIS	更新事件除能控制 0: 以下任何一个事件都可使能更新事件请求 – 计数器上溢 / 下溢 – 设置 UEVG 位 – 通过从机模式产生更新 1: 除能更新事件 ( 如果 UEVG 位被置位或在从机模式接收到硬件重启, 那么计数器和预分频器将被初始化 )

定时器模式配置寄存器 – MDCFR

该寄存器定义了 GPTM 主机和从机模式选项以及单脉冲模式。

偏移量:	0x004																																		
复位值:	0x0000_0000																																		
	<table><tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr><tr><td colspan="7">保留位</td><td>SPMSET</td></tr><tr><td colspan="7"></td><td>RW</td><td>0</td></tr></table>	31	30	29	28	27	26	25	24	保留位							SPMSET								RW	0									
31	30	29	28	27	26	25	24																												
保留位							SPMSET																												
							RW	0																											
类型 / 复位																																			
	<table><tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr><tr><td colspan="5">保留位</td><td colspan="3">MMSEL</td></tr><tr><td colspan="5"></td><td>RW</td><td>0</td><td>RW</td><td>0</td><td>RW</td><td>0</td></tr></table>	23	22	21	20	19	18	17	16	保留位					MMSEL								RW	0	RW	0	RW	0							
23	22	21	20	19	18	17	16																												
保留位					MMSEL																														
					RW	0	RW	0	RW	0																									
类型 / 复位																																			
	<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td colspan="5">保留位</td><td colspan="3">SMSEL</td></tr><tr><td colspan="5"></td><td>RW</td><td>0</td><td>RW</td><td>0</td><td>RW</td><td>0</td></tr></table>	15	14	13	12	11	10	9	8	保留位					SMSEL								RW	0	RW	0	RW	0							
15	14	13	12	11	10	9	8																												
保留位					SMSEL																														
					RW	0	RW	0	RW	0																									
类型 / 复位																																			
	<table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="7">保留位</td><td>TSE</td></tr><tr><td colspan="7"></td><td>RW</td><td>0</td></tr></table>	7	6	5	4	3	2	1	0	保留位							TSE								RW	0									
7	6	5	4	3	2	1	0																												
保留位							TSE																												
							RW	0																											
类型 / 复位																																			

位	字段	描述
[24]	SPMSET	单脉冲模式设置 0: 无论更新事件是否发生, 计数器正常计数 1: 下一个更新事件到来时, 计数器停止计数, 接着 TME 位被硬件清零

位	字段	描述																											
[18:16]	MMSEL	<p>主机模式选项</p> <p>主机模式选项用来选择同步其它从机定时器的 MTO 信号源。</p> <table> <tr> <th>MMSEL[2:0]</th><th>模式</th><th>描述</th></tr> <tr> <td>000</td><td>复位模式</td><td>复位模式下的 MTO 输出信号由以下条件之一产生： 1. 软件设置 UEVG 位 2. 当定时器用于从机重启模式，STI 触发输入信号被输出到 MTO 信号线</td></tr> <tr> <td>001</td><td>使能模式</td><td>计数器使能信号作为触发输出</td></tr> <tr> <td>010</td><td>更新模式</td><td>当 UEVDIS 位被清零时，更新事件用作触发输出且取决于下列条件之一： 1. 计数器上溢 / 下溢 2. 软件设置 UEVG 位 3. 从机重启模式下从机触发输入</td></tr> <tr> <td>011</td><td>捕捉 / 比较模式</td><td>当通道 0 捕捉或比较匹配事件发生，将产生一个正脉冲作为主机触发输出。</td></tr> <tr> <td>100</td><td>比较模式 0</td><td>通道 0 输出参考信号 CH0OREF 作为触发输出。</td></tr> <tr> <td>101</td><td>比较模式 1</td><td>通道 1 输出参考信号 CH1OREF 作为触发输出。</td></tr> <tr> <td>110</td><td>比较模式 2</td><td>通道 2 输出参考信号 CH2OREF 作为触发输出。</td></tr> <tr> <td>111</td><td>比较模式 3</td><td>通道 3 输出参考信号 CH3OREF 作为触发输出。</td></tr> </table>	MMSEL[2:0]	模式	描述	000	复位模式	复位模式下的 MTO 输出信号由以下条件之一产生： 1. 软件设置 UEVG 位 2. 当定时器用于从机重启模式，STI 触发输入信号被输出到 MTO 信号线	001	使能模式	计数器使能信号作为触发输出	010	更新模式	当 UEVDIS 位被清零时，更新事件用作触发输出且取决于下列条件之一： 1. 计数器上溢 / 下溢 2. 软件设置 UEVG 位 3. 从机重启模式下从机触发输入	011	捕捉 / 比较模式	当通道 0 捕捉或比较匹配事件发生，将产生一个正脉冲作为主机触发输出。	100	比较模式 0	通道 0 输出参考信号 CH0OREF 作为触发输出。	101	比较模式 1	通道 1 输出参考信号 CH1OREF 作为触发输出。	110	比较模式 2	通道 2 输出参考信号 CH2OREF 作为触发输出。	111	比较模式 3	通道 3 输出参考信号 CH3OREF 作为触发输出。
MMSEL[2:0]	模式	描述																											
000	复位模式	复位模式下的 MTO 输出信号由以下条件之一产生： 1. 软件设置 UEVG 位 2. 当定时器用于从机重启模式，STI 触发输入信号被输出到 MTO 信号线																											
001	使能模式	计数器使能信号作为触发输出																											
010	更新模式	当 UEVDIS 位被清零时，更新事件用作触发输出且取决于下列条件之一： 1. 计数器上溢 / 下溢 2. 软件设置 UEVG 位 3. 从机重启模式下从机触发输入																											
011	捕捉 / 比较模式	当通道 0 捕捉或比较匹配事件发生，将产生一个正脉冲作为主机触发输出。																											
100	比较模式 0	通道 0 输出参考信号 CH0OREF 作为触发输出。																											
101	比较模式 1	通道 1 输出参考信号 CH1OREF 作为触发输出。																											
110	比较模式 2	通道 2 输出参考信号 CH2OREF 作为触发输出。																											
111	比较模式 3	通道 3 输出参考信号 CH3OREF 作为触发输出。																											
[10:8]	SMSEL	<p>从机模式选项</p> <table> <tr> <th>SMSEL[2:0]</th><th>模式</th><th>描述</th></tr> <tr> <td>000</td><td>除能模式</td><td>预分频器直接用内部时钟计时。</td></tr> <tr> <td>001</td><td>正交解码模式 1</td><td>计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI0 边沿转换取决于 TI1 的电平。</td></tr> <tr> <td>010</td><td>正交解码模式 2</td><td>计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI1 边沿转换取决于 TI0 的电平。</td></tr> <tr> <td>011</td><td>正交解码模式 3</td><td>计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。在此模式下的一个通道边沿转换取决于另外一个通道的电平。</td></tr> <tr> <td>100</td><td>重启模式</td><td>计数器在 STI 信号上升沿，依计数模式从 0 或 CRR 影子寄存器的值重新开始计数。寄存器也将被更新。</td></tr> <tr> <td>101</td><td>暂停模式</td><td>当所选择的触发输入信号 STI 是高电平时，计数器开始计数。当 STI 信号转换成低电平时，计数器立即停止计数，而不会产生复位。计数器开始和停止计数是由 STI 信号控制的。</td></tr> <tr> <td>110</td><td>触发模式</td><td>在所选择的 STI 触发信号上升沿处，计数器从初始值开始计数。只有计数器的开启是由 STI 信号控制的。</td></tr> <tr> <td>111</td><td>STIED</td><td>计数器使用所选的触发信号 STI 的上升沿计时。</td></tr> </table>	SMSEL[2:0]	模式	描述	000	除能模式	预分频器直接用内部时钟计时。	001	正交解码模式 1	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI0 边沿转换取决于 TI1 的电平。	010	正交解码模式 2	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI1 边沿转换取决于 TI0 的电平。	011	正交解码模式 3	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。在此模式下的一个通道边沿转换取决于另外一个通道的电平。	100	重启模式	计数器在 STI 信号上升沿，依计数模式从 0 或 CRR 影子寄存器的值重新开始计数。寄存器也将被更新。	101	暂停模式	当所选择的触发输入信号 STI 是高电平时，计数器开始计数。当 STI 信号转换成低电平时，计数器立即停止计数，而不会产生复位。计数器开始和停止计数是由 STI 信号控制的。	110	触发模式	在所选择的 STI 触发信号上升沿处，计数器从初始值开始计数。只有计数器的开启是由 STI 信号控制的。	111	STIED	计数器使用所选的触发信号 STI 的上升沿计时。
SMSEL[2:0]	模式	描述																											
000	除能模式	预分频器直接用内部时钟计时。																											
001	正交解码模式 1	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI0 边沿转换取决于 TI1 的电平。																											
010	正交解码模式 2	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI1 边沿转换取决于 TI0 的电平。																											
011	正交解码模式 3	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。在此模式下的一个通道边沿转换取决于另外一个通道的电平。																											
100	重启模式	计数器在 STI 信号上升沿，依计数模式从 0 或 CRR 影子寄存器的值重新开始计数。寄存器也将被更新。																											
101	暂停模式	当所选择的触发输入信号 STI 是高电平时，计数器开始计数。当 STI 信号转换成低电平时，计数器立即停止计数，而不会产生复位。计数器开始和停止计数是由 STI 信号控制的。																											
110	触发模式	在所选择的 STI 触发信号上升沿处，计数器从初始值开始计数。只有计数器的开启是由 STI 信号控制的。																											
111	STIED	计数器使用所选的触发信号 STI 的上升沿计时。																											
[0]	TSE	<p>定时器同步使能</p> <p>0: 无动作</p> <p>1: 主机定时器 (当前定时器) 将产生一个延时以通过 MTO 信号同步其从机定时器</p>																											

定时器触发配置寄存器 – TRCFR

该寄存器定义了 GPTM 的时钟设置和触发源选项。

偏移量: 0x008  
复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							ECME
								RW 0
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							ETIPOL
								RW 0
类型 / 复位	15	14	13	12	11	10	9	8
	保留位		ETIPSC			ETF		
			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位				TRSEL			
					RW 0	RW 0	RW 0	RW 0

位	字段	描述
[24]	ECME	外部时钟模式使能 0: 外部时钟模式除能 1: 外部时钟模式使能 设置 ECME 位为 1 效果等同于定时器被配置在 STI 触发从机模式下，触发源来自 GTn_ETI 引脚，使用的是 GTn_ETI 引脚的外部时钟输入。
[16]	ETIPOL	外部触发极性 0: GTn_ETI 作用在高电平或上升沿 1: GTn_ETI 作用在低电平或下降沿
[13:12]	ETIPSC	外部触发预分频器 分频器使能来减小 ETIP 频率 00: 分频器 OFF 01: ETIP 频率 / 2 10: ETIP 频率 / 4 11: ETIP 频率 / 8

位	字段	描述
[11:8]	ETF	<p>外部触发滤波器</p> <p>这些位定义了分频比用来采样 GTn_ETI 信号。GPTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出滤波信号所需的有效转换次数。</p> <p>0000: 无滤波，采样时钟是 <math>f_{DTS}</math></p> <p>0001: <math>f_{\text{sampling}} = f_{\text{CLKIN}}, N = 2</math></p> <p>0010: <math>f_{\text{sampling}} = f_{\text{CLKIN}}, N = 4</math></p> <p>0011: <math>f_{\text{sampling}} = f_{\text{CLKIN}}, N = 8</math></p> <p>0100: <math>f_{\text{sampling}} = f_{DTS} / 2, N = 6</math></p> <p>0101: <math>f_{\text{sampling}} = f_{DTS} / 2, N = 8</math></p> <p>0110: <math>f_{\text{sampling}} = f_{DTS} / 4, N = 6</math></p> <p>0111: <math>f_{\text{sampling}} = f_{DTS} / 4, N = 8</math></p> <p>1000: <math>f_{\text{sampling}} = f_{DTS} / 8, N = 6</math></p> <p>1001: <math>f_{\text{sampling}} = f_{DTS} / 8, N = 8</math></p> <p>1010: <math>f_{\text{sampling}} = f_{DTS} / 16, N = 5</math></p> <p>1011: <math>f_{\text{sampling}} = f_{DTS} / 16, N = 6</math></p> <p>1100: <math>f_{\text{sampling}} = f_{DTS} / 16, N = 8</math></p> <p>1101: <math>f_{\text{sampling}} = f_{DTS} / 32, N = 5</math></p> <p>1110: <math>f_{\text{sampling}} = f_{DTS} / 32, N = 6</math></p> <p>1111: <math>f_{\text{sampling}} = f_{DTS} / 32, N = 8</math></p>
[3:0]	TRSEL	<p>触发源选择</p> <p>这些位用来选择用于计数器同步的触发输入源 (STI)</p> <p>0000: 通过设置 UEVG 位软件触发</p> <p>0001: 通道 0 滤波输入 (TI0S0)</p> <p>0010: 通道 1 滤波输入 (TI1S1)</p> <p>0011: 外部触发输入 (ETIF)</p> <p>1000: 通道 0 边沿检测器 (TI0BED)</p> <p>1001: 内部定时器模块触发器 0 (ITI0)</p> <p>1010: 内部定时器模块触发器 1 (ITI1)</p> <p>1011: 内部定时器模块触发器 2 (ITI2)</p> <p>其它: 默认 0</p>

注：这些位必须在未被使用时更新，即设置 SMSEL 字段为 0x00 除能从机模式。

**表 34. GPTM 内部触发器连接**

从机定时器模块	ITI0	ITI1	ITI2
GPTM0	GPTM1	MCTM0	MCTM1
GPTM1	GPTM0	MCTM0	MCTM1

## 定时器控制寄存器 – CTR

该寄存器定义了定时器使能位 (TME)、CRR 缓冲器使能位 (CRBE) 和通道 PDMA 选择位 (CHCCDS)。

偏移量: 0x010

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							CHCCDS
类型 / 复位							RW 0
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位						CRBE	TME
类型 / 复位						RW 0	RW 0

位	字段	描述
[16]	CHCCDS	通道 PDMA 事件选择 0: 通道 PDMA 请求来自通道捕捉 / 比较事件 1: 通道 PDMA 请求来自更新事件
[1]	CRBE	计数器重载寄存器缓冲器使能位 0: 计数器重载寄存器立即被更新 1: 直到更新事件发生时计数器重载寄存器才会被更新
[0]	TME	定时器使能位 0: GPTM 关闭 1: GPTM 开启 – GPTM 功能正常 若 GPTM 工作在除单脉冲模式和从机触发模式之外的其它模式，当 TME 位被清零，计数器停止计数且不再产生功耗。而在单脉冲模式和从机触发模式时，TME 位可通过硬件自动置 1，所有的 GPTM 寄存器正常工作。



通道 0 输入配置寄存器 – CH0ICFR

该寄存器定义了通道 0 输入模式配置。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	TI0SRC	保留位						
类型 / 复位	RW	0						
	23	22	21	20	19	18	17	16
	保留位				CH0PSC		CH0CCS	
类型 / 复位					RW	0	RW	0
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位				TI0F			
类型 / 复位					RW	0	RW	0

位	字段	描述
[31]	TI0SRC	通道 0 输入源 TI0 选择 0: GTn_CH0 引脚连接到通道 0 输入 TI0 1: GTn_CH0、GTn_CH1 和 GTn_CH2 引脚异或操作后的输出连接到通道 0 输入 TI0
[19:18]	CH0PSC	通道 0 捕捉输入源预分频器设置 这些位定义了通道 0 捕捉输入的有效事件。注意，一旦通道控制寄存器 CHCTR 中的通道 0 捕捉 / 比较使能位 CH0E 位被清零，则预分频器将被复位。 00: 无分频，每个有效事件为一个通道 0 捕捉输入信号 01: 每 2 个事件为一个通道 0 捕捉输入信号 10: 每 4 个事件为一个通道 0 捕捉输入信号 11: 每 8 个事件为一个通道 0 捕捉输入信号
[17:16]	CH0CCS	通道 0 捕捉 / 比较选择 00: 通道 0 配置为输出 01: 通道 0 配置为来自 TI0 信号的一个输入 10: 通道 0 配置为来自 TI1 信号的一个输入 11: 通道 0 配置为来自触发控制器产生的 TRCED 信号的一个输入 注：仅当 CH0E 位清零时，才可以访问 CH0CCS 字段。

位	字段	描述
[3:0]	TI0F	<p>通道 0 输入源 TI0 滤波器设置</p> <p>这些位定义了用来采样 TI0 信号的分频比。GPTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出滤波信号所需的有效转换次数。</p> <p>0000: 无滤波，采样时钟是 <math>f_{DTS}</math></p> <p>0001: <math>f_{sampling} = f_{CLKIN}</math>, <math>N = 2</math></p> <p>0010: <math>f_{sampling} = f_{CLKIN}</math>, <math>N = 4</math></p> <p>0011: <math>f_{sampling} = f_{CLKIN}</math>, <math>N = 8</math></p> <p>0100: <math>f_{sampling} = f_{DTS} / 2</math>, <math>N = 6</math></p> <p>0101: <math>f_{sampling} = f_{DTS} / 2</math>, <math>N = 8</math></p> <p>0110: <math>f_{sampling} = f_{DTS} / 4</math>, <math>N = 6</math></p> <p>0111: <math>f_{sampling} = f_{DTS} / 4</math>, <math>N = 8</math></p> <p>1000: <math>f_{sampling} = f_{DTS} / 8</math>, <math>N = 6</math></p> <p>1001: <math>f_{sampling} = f_{DTS} / 8</math>, <math>N = 8</math></p> <p>1010: <math>f_{sampling} = f_{DTS} / 16</math>, <math>N = 5</math></p> <p>1011: <math>f_{sampling} = f_{DTS} / 16</math>, <math>N = 6</math></p> <p>1100: <math>f_{sampling} = f_{DTS} / 16</math>, <math>N = 8</math></p> <p>1101: <math>f_{sampling} = f_{DTS} / 32</math>, <math>N = 5</math></p> <p>1110: <math>f_{sampling} = f_{DTS} / 32</math>, <math>N = 6</math></p> <p>1111: <math>f_{sampling} = f_{DTS} / 32</math>, <math>N = 8</math></p>

## 通道 1 输入配置寄存器 – CH1ICFR

该寄存器定义了通道 1 输入模式配置。

偏移量: 0x024

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位				CHIPSC		CH1CCS	
			RW	0	RW	0	RW
15	14	13	12	11	10	9	8
保留位							
7	6	5	4	3	2	1	0
保留位				TIIF			
			RW	0	RW	0	RW

位	字段	描述
[19:18]	CHIPSC	通道 1 捕捉输入源预分频器设置 这些位定义了通道 1 捕捉输入的有效事件。注意，一旦通道控制寄存器 CHCTR 中的通道 1 捕捉 / 比较使能位 CH1E 被清零，则预分频器将被复位。 00: 无分频，每个有效事件为一个通道 1 捕捉输入信号 01: 每 2 个事件为一个通道 1 捕捉输入信号 10: 每 4 个事件为一个通道 1 捕捉输入信号 11: 每 8 个事件为一个通道 1 捕捉输入信号
[17:16]	CH1CCS	通道 1 捕捉 / 比较选项 00: 通道 1 被配置为输出 01: 通道 1 被配置为来自 TI1 信号的一个输入 10: 通道 1 被配置为来自 TI0 信号的一个输入 11: 通道 1 被配置为来自触发控制器产生的 TRCED 信号的一个输入 注：仅当 CH1E 位清零时，才可以访问 CH1CCS 字段。
[3:0]	TIIF	通道 1 输入源 TI1 滤波器设置 这些位定义了用来采样 TI1 信号的分频比。GPTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出滤波信号所需的有效转换次数。 0000: 无滤波，采样时钟是 $f_{DTS}$ 0001: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 2$ 0010: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 4$ 0011: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 8$ 0100: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 6$ 0101: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 8$ 0110: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 6$ 0111: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 8$ 1000: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 6$ 1001: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 8$ 1010: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 5$ 1011: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 6$ 1100: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 8$ 1101: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 5$ 1110: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 6$ 1111: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 8$

## 通道 2 输入配置寄存器 – CH2ICFR

该寄存器定义了通道 2 输入模式配置。

偏移量: 0x028

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位				CH2PSC		CH2CCS	
				RW	0	RW	0
15	14	13	12	11	10	9	8
保留位							
7	6	5	4	3	2	1	0
保留位				TI2F			
				RW	0	RW	0

位	字段	描述
[19:18]	CH2PSC	通道 2 捕捉输入源预分频器设置 这些位定义了通道 2 捕捉输入的有效事件。注意，一旦通道控制寄存器 CHCTR 中的通道 2 捕捉 / 比较使能位 CH2E 被清零，则预分频器将被复位。 00: 无分频，每个有效事件为一个通道 2 捕捉输入信号 01: 每 2 个事件为一个通道 2 捕捉输入信号 10: 每 4 个事件为一个通道 2 捕捉输入信号 11: 每 8 个事件为一个通道 2 捕捉输入信号
[17:16]	CH2CCS	通道 2 捕捉 / 比较选项 00: 通道 2 被配置为输出 01: 通道 2 被配置为来自 TI2 信号的一个输入 10: 通道 2 被配置为来自 TI3 信号的一个输入 11: 通道 2 被配置为来自触发控制器产生的 TRCED 信号的一个输入 注：仅当 CH2E 位清零时，才可以访问 CH2CCS 字段。
[3:0]	TI2F	通道 2 输入源 TI2 滤波器设置 这些位定义了用来采样 TI2 信号的分频比。GPTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出滤波信号所需的有效转换次数。 0000: 无滤波，采样时钟是 $f_{DTS}$ 0001: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 2$ 0010: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 4$ 0011: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 8$ 0100: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 6$ 0101: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 8$ 0110: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 6$ 0111: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 8$ 1000: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 6$ 1001: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 8$ 1010: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 5$ 1011: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 6$ 1100: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 8$ 1101: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 5$ 1110: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 6$ 1111: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 8$

## 通道 3 输入配置寄存器 – CH3ICFR

该寄存器定义了通道 3 输入模式配置。

偏移量: 0x02C

复位值: 0x0000\_0000

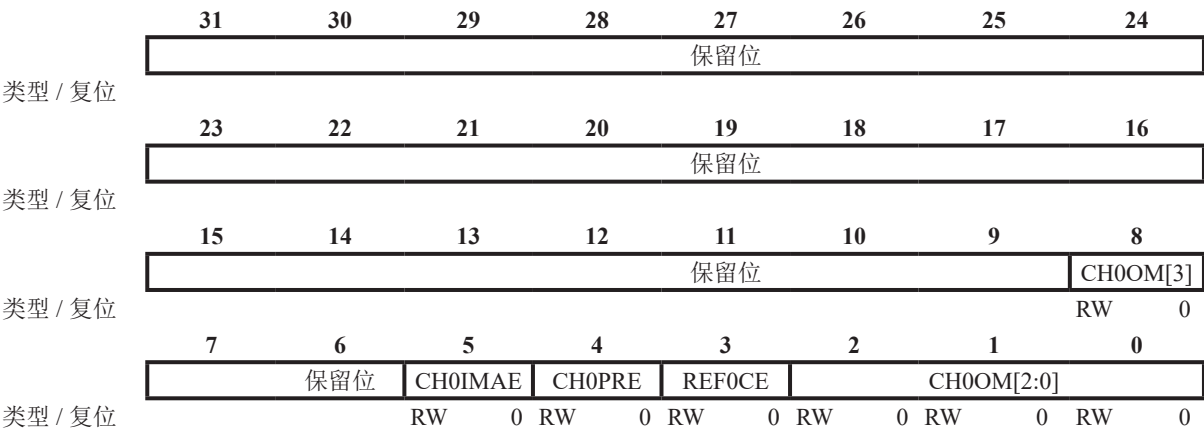
31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位				CH3PSC		CH3CCS	
类型 / 复位				RW	0	RW	0
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位				TI3F			
类型 / 复位				RW	0	RW	0

位	字段	描述
[19:18]	CH3PSC	通道 3 捕捉输入源预分频器设置 这些位定义了通道 3 捕捉输入的有效事件。注意，一旦通道控制寄存器 CHCTR 中的通道 3 捕捉 / 比较使能位 CH3E 被清零，则预分频器将被复位。 00: 无分频，每个有效事件为一个通道 3 捕捉输入信号 01: 每 2 个事件为一个通道 3 捕捉输入信号 10: 每 4 个事件为一个通道 3 捕捉输入信号 11: 每 8 个事件为一个通道 3 捕捉输入信号
[17:16]	CH3CCS	通道 3 捕捉 / 比较选项 00: 通道 3 被配置为输出 01: 通道 3 被配置为来自 TI3 信号的一个输入 10: 通道 3 被配置为来自 TI2 信号的一个输入 11: 通道 3 被配置为来自触发控制器产生的 TRCED 信号的一个输入 注：仅当 CH3E 位清零时，才可以访问 CH3CCS 字段。
[3:0]	TI3F	通道 3 输入源 TI3 滤波器设置 这些位定义了用来采样 TI3 信号的分频比。GPTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出滤波信号所需的有效转换次数。 0000: 无滤波，采样时钟是 $f_{DTS}$ 0001: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 2$ 0010: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 4$ 0011: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 8$ 0100: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 6$ 0101: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 8$ 0110: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 6$ 0111: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 8$ 1000: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 6$ 1001: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 8$ 1010: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 5$ 1011: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 6$ 1100: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 8$ 1101: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 5$ 1110: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 6$ 1111: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 8$

通道 0 输出配置寄存器 – CH0OCFR

该寄存器定义了通道 0 输出模式配置。

偏移量: 0x040  
复位值: 0x0000\_0000



位	字段	描述
[5]	CH0IMAE	通道 0 立即有效使能位 0: 无动作 1: 单脉冲立即有效模式使能 无论 CNTR 和 CH0CCR 值的比较结果如何, 在一个有效触发事件发生后, CH0OREF 会立即强制为比较匹配电平。 有效时间将在下一个上溢或下溢事件到来时自动结束。 注: 只有当通道 0 被配置工作在 PWM 模式 1 或 PWM 模式 2 时, CH0IMAE 位可用。
[4]	CH0PRE	通道 0 捕捉 / 比较寄存器 (CH0CCR) 预载使能位 0: CH0CCR 预载功能除能 当 CH0PRE 位清零, CH0CCR 寄存器将立即被更新为新的值, 且立即可用。 1: CH0CCR 预载功能使能 直到更新事件发生后, 新的 CH0CCR 值才会被传送到影子寄存器中。
[3]	REF0CE	通道 0 参考输出清除使能位 0: CH0OREF 正常执行, 不受 ETIF 信号影响 1: CH0OREF 在来自于 GTn_ETI 引脚的 ETIF 信号的高电平处被强制为 0

位	字段	描述
[8][2:0]	CH0OM[3:0]	<p>通道 0 输出模式设置</p> <p>这些位定义了输出参考信号 CH0OREF 的功能类型</p> <p>0000: 无变化</p> <p>0001: 比较匹配时输出 0</p> <p>0010: 比较匹配时输出 1</p> <p>0011: 比较匹配时输出翻转</p> <p>0100: 强制无效 – CH0OREF 强制为 0</p> <p>0101: 强制有效 – CH0OREF 强制为 1</p> <p>0110: PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH0CCR, 通道 0 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH0CCR, 通道 0 处于无效电平, 否则将处于有效电平。</li> </ul> <p>0111: PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH0CCR, 通道 0 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH0CCR, 通道 0 处于有效电平, 否则将处于无效电平。</li> </ul> <p>1110: 非对称 PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH0CCR, 通道 0 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH0CCR, 通道 0 处于无效电平, 否则将处于有效电平。</li> </ul> <p>1111: 非对称 PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH0CCR, 通道 0 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH0CCR, 通道 0 处于有效电平, 否则将处于无效电平。</li> </ul> <p>注: 当通道 0 用作非对称 PWM 输出模式时, 计数器配置寄存器中的计数器模式选择位必须配置为中心对齐计数模式 (CMSEL = 0x1 / 0x2 / 0x3)。</p>

通道 1 输出配置寄存器 – CH1OCFR

该寄存器定义了通道 1 输出模式配置。

偏移量：0x044

复位值：0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	保留位							CH1OM[3]	
类型 / 复位								RW	0
	7	6	5	4	3	2	1	0	
	保留位		CH1IMAE	CH1PRE	REF1CE	CH1OM[2:0]			
类型 / 复位			RW	0	RW	0	RW	0	RW
									0

位	字段	描述
[5]	CH1IMAE	通道 1 立即有效使能位 0: 无动作 1: 单脉冲立即有效模式使能 无论 CNTR 和 CH1CCR 值的比较结果如何，在一个有效触发事件发生后，CH1OREF 会立即强制为比较匹配电平。 有效时间将在下一个上溢或下溢事件到来时自动结束。 注：只有当通道 1 被配置工作在 PWM 模式 1 或 PWM 模式 2 时，CH1IMAE 位可用。
[4]	CH1PRE	通道 1 捕捉 / 比较寄存器 (CH1CCR) 预载使能位 0: CH1CCR 预载功能除能 当 CH1PRE 位清零，CH1CCR 寄存器将立即被更新为新的值，且立即可用。 1: CH1CCR 预载功能使能 直到更新事件发生后，新的 CH1CCR 值才会被传送到影子寄存器中。
[3]	REF1CE	通道 1 参考输出清除使能位 0: CH1OREF 正常执行，不受 ETIF 信号影响 1: CH1OREF 在来自于 GTn_ETI 引脚的 ETIF 信号的高电平处被强制为 0

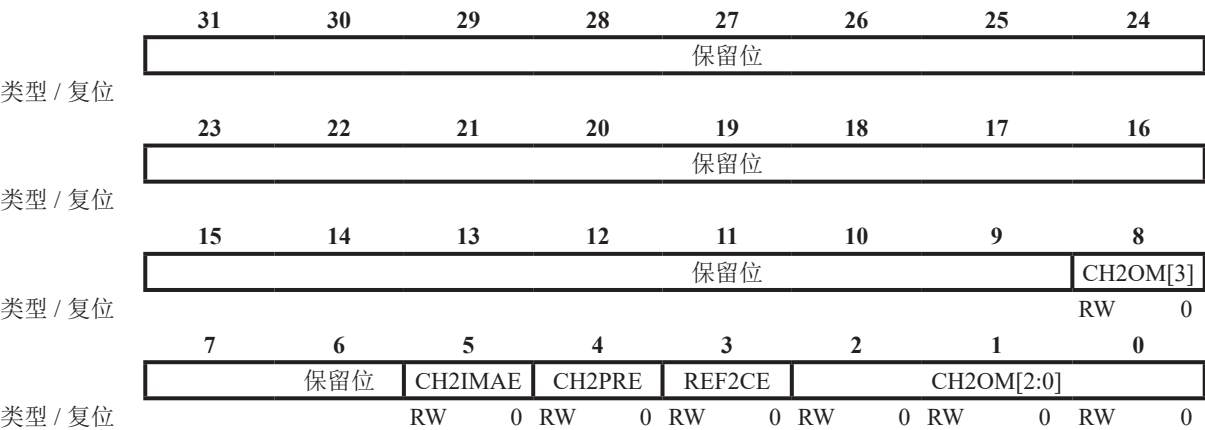


位	字段	描述
[8][2:0]	CH1OM[3:0]	<p>通道 1 输出模式设置 这些位定义了输出参考信号 CH1OREF 的功能类型</p> <p>0000: 无变化</p> <p>0001: 比较匹配时输出 0</p> <p>0010: 比较匹配时输出 1</p> <p>0011: 比较匹配时输出翻转</p> <p>0100: 强制无效 – CH1OREF 强制为 0</p> <p>0101: 强制有效 – CH1OREF 强制为 1</p> <p>0110: PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH1CCR, 通道 1 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH1CCR, 通道 1 处于无效电平, 否则将处于有效电平。</li> </ul> <p>0111: PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH1CCR, 通道 1 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH1CCR, 通道 1 处于有效电平, 否则将处于无效电平。</li> </ul> <p>1110: 非对称 PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH1CCR, 通道 1 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH1CCR, 通道 1 处于无效电平, 否则将处于有效电平。</li> </ul> <p>1111: 非对称 PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH1CCR, 通道 1 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH1CCR, 通道 1 处于有效电平, 否则将处于无效电平。</li> </ul> <p>注: 当通道 1 用作非对称 PWM 输出模式时, 计数器配置寄存器中的计数器模式选择位必须配置为中心对齐计数模式 (CMSEL = 0x1 / 0x2 / 0x3)。</p>

通道 2 输出配置寄存器 – CH2OCFR

该寄存器定义了通道 2 输出模式配置。

偏移量： 0x048  
复位值： 0x0000\_0000



位	字段	描述
[5]	CH2IMAE	通道 2 立即有效使能位 0: 无动作 1: 单脉冲立即有效模式使能 无论 CNTR 和 CH2CCR 值的比较结果如何，在一个有效触发事件发生后，CH2OREF 会立即强制为比较匹配电平。 有效时间将在下一个上溢或下溢事件到来时自动结束。 注：只有当通道 2 被配置工作在 PWM 模式 1 或 PWM 模式 2 时，CH2IMAE 位可用。
[4]	CH2PRE	通道 2 捕捉 / 比较寄存器 (CH2CCR) 预载使能位 0: CH2CCR 预载功能除能 当 CH2PRE 位清零，CH2CCR 寄存器将立即被更新为新的值，且立即可用。 1: CH2CCR 预载功能使能 直到更新事件发生后，新的 CH2CCR 值才会被传送到影子寄存器中。
[3]	REF2CE	通道 2 参考输出清除使能位 0: CH2OREF 正常执行，不受 ETIF 信号影响 1: CH2OREF 在来自于 GTn_ETI 引脚的 ETIF 信号的高电平处被强制为 0

位	字段	描述
[8][2:0]	CH2OM[3:0]	<p>通道 2 输出模式设置 这些位定义了输出参考信号 CH2OREF 的功能类型</p> <p>0000: 无变化 0001: 比较匹配时输出 0 0010: 比较匹配时输出 1 0011: 比较匹配时输出翻转 0100: 强制无效 – CH2OREF 强制为 0 0101: 强制有效 – CH2OREF 强制为 1 0110: PWM 模式 1 – 在向上计数期间, 当 CNTR &lt; CH2CCR, 通道 2 处于有效电平, 否则将处于无效电平。 – 在向下计数期间, 当 CNTR &gt; CH2CCR, 通道 2 处于无效电平, 否则将处于有效电平。 0111: PWM 模式 2 – 在向上计数期间, 当 CNTR &lt; CH2CCR, 通道 2 处于无效电平, 否则将处于有效电平。 – 在向下计数期间, 当 CNTR &gt; CH2CCR, 通道 2 处于有效电平, 否则将处于无效电平。 1110: 非对称 PWM 模式 1 – 在向上计数期间, 当 CNTR &lt; CH2CCR, 通道 2 处于有效电平, 否则将处于无效电平。 – 在向下计数期间, 当 CNTR &gt; CH2CCR, 通道 2 处于无效电平, 否则将处于有效电平。 1111: 非对称 PWM 模式 2 – 在向上计数期间, 当 CNTR &lt; CH2CCR, 通道 2 处于无效电平, 否则将处于有效电平。 – 在向下计数期间, 当 CNTR &gt; CH2CCR, 通道 2 处于有效电平, 否则将处于无效电平。</p> <p>注: 当通道 2 用作非对称 PWM 输出模式时, 计数器配置寄存器中的计数器模式选择位必须配置为中心对齐计数模式 (CMSEL = 0x1 / 0x2 / 0x3)。</p>

通道 3 输出配置寄存器 – CH3OCFR

该寄存器定义了通道 3 输出模式配置。

偏移量： 0x04C  
复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位							CH3OM[3]
类型 / 复位	7	6	5	4	3	2	1	0
	保留位		CH3IMAE	CH3PRE	REF3CE	CH3OM[2:0]		
	RW		0	RW	0	RW	0	RW
	0		0	0	0	0	0	0

位	字段	描述
[5]	CH3IMAE	通道 3 立即有效使能位 0: 无动作 1: 单脉冲立即有效模式使能 无论 CNTR 和 CH3CCR 值的比较结果如何，在一个有效触发事件发生后，CH3OREF 会立即强制为比较匹配电平。 有效时间将在下一个上溢或下溢事件到来时自动结束。 注：只有当通道 3 被配置工作在 PWM 模式 1 或 PWM 模式 2 时，CH3IMAE 位可用。
[4]	CH3PRE	通道 3 捕捉 / 比较寄存器 (CH3CCR) 预载使能位 0: CH3CCR 预载功能除能 当 CH3PRE 位清零，CH3CCR 寄存器将立即被更新为新的值，且立即可用。 1: CH3CCR 预载功能使能 直到更新事件发生后，新的 CH3CCR 值才会被传送到影子寄存器中。
[3]	REF3CE	通道 3 参考输出清除使能位 0: CH3OREF 正常执行，不受 ETIF 信号影响 1: CH3OREF 在来自于 GTn_ETI 引脚的 ETIF 信号的高电平处被强制为 0

位	字段	描述
[8][2:0]	CH3OM[3:0]	<p>通道 3 输出模式设置 这些位定义了输出参考信号 CH3OREF 的功能类型</p> <p>0000: 无变化</p> <p>0001: 比较匹配时输出 0</p> <p>0010: 比较匹配时输出 1</p> <p>0011: 比较匹配时输出翻转</p> <p>0100: 强制无效 – CH3OREF 强制为 0</p> <p>0101: 强制有效 – CH3OREF 强制为 1</p> <p>0110: PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH3CCR, 通道 3 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH3CCR, 通道 3 处于无效电平, 否则将处于有效电平。</li> </ul> <p>0111: PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH3CCR, 通道 3 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH3CCR, 通道 3 处于有效电平, 否则将处于无效电平。</li> </ul> <p>1110: 非对称 PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH3CCR, 通道 3 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH3CCR, 通道 3 处于无效电平, 否则将处于有效电平。</li> </ul> <p>1111: 非对称 PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH3CCR, 通道 3 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH3CCR, 通道 3 处于有效电平, 否则将处于无效电平。</li> </ul> <p>注: 当通道 3 用作非对称 PWM 输出模式时, 计数器配置寄存器中的计数器模式选择位必须配置为中心对齐计数模式 (CMSEL = 0x1 / 0x2 / 0x3)。</p>

## 通道控制寄存器 – CHCTR

该寄存器包含了通道捕捉输入和比较输出功能使能控制位。

偏移量: 0x050

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位	CH3E	保留位	CH2E	保留位	CH1E	保留位	CH0E
	RW 0		RW 0		RW 0		RW 0
类型 / 复位							

位	字段	描述
[6]	CH3E	通道 3 捕捉 / 比较使能位 – 通道 3 被配置为输入 (CH3CCS = 0x01 / 0x02 / 0x03) 0: 输入捕捉模式除能 1: 输入捕捉模式使能 – 通道 3 被配置为输出 (CH3CCS = 0x00) 0: 关闭 – 通道 3 输出信号 CH3O 无作用 1: 开启 – 通道 3 输出信号 CH3O 在相应的输出脚产生
[4]	CH2E	通道 2 捕捉 / 比较使能位 – 通道 2 被配置为输入 (CH2CCS = 0x01 / 0x02 / 0x03) 0: 输入捕捉模式除能 1: 输入捕捉模式使能 – 通道 2 被配置为输出 (CH2CCS = 0x00) 0: 关闭 – 通道 2 输出信号 CH2O 无作用 1: 开启 – 通道 2 输出信号 CH2O 在相应的输出脚产生
[2]	CH1E	通道 1 捕捉 / 比较使能位 – 通道 1 被配置为输入 (CH1CCS = 0x01 / 0x02 / 0x03) 0: 输入捕捉模式除能 1: 输入捕捉模式使能 – 通道 1 被配置为输出 (CH1CCS = 0x00) 0: 关闭 – 通道 1 输出信号 CH1O 无作用 1: 开启 – 通道 1 输出信号 CH1O 在相应的输出脚产生
[0]	CH0E	通道 0 捕捉 / 比较使能位 – 通道 0 被配置为输入 (CH0CCS = 0x01 / 0x02 / 0x03) 0: 输入捕捉模式除能 1: 输入捕捉模式使能 – 通道 0 被配置为输出 (CH0CCS = 0x00) 0: 关闭 – 通道 0 输出信号 CH0O 无作用 1: 开启 – 通道 0 输出信号 CH0O 在相应的输出脚产生

## 通道极性配置寄存器 – CHPOLR

该寄存器包含了通道捕捉输入或比较输出极性的控制。

偏移量: 0x054

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位	CH3P	保留位	CH2P	保留位	CH1P	保留位	CH0P
	RW 0		RW 0		RW 0		RW 0
类型 / 复位							

位	字段	描述
[6]	CH3P	通道 3 捕捉 / 比较极性 – 通道 3 被配置为输入时 (CH3CCS = 0x01 / 0x02 / 0x03) 0: 捕捉事件发生在通道 3 上升沿 1: 捕捉事件发生在通道 3 下降沿 – 通道 3 被配置为输出时 (CH3CCS = 0x00) 0: 通道 3 输出高电平有效 1: 通道 3 输出低电平有效
[4]	CH2P	通道 2 捕捉 / 比较极性 – 通道 2 被配置为输入时 (CH2CCS = 0x01 / 0x02 / 0x03) 0: 捕捉事件发生在通道 2 上升沿 1: 捕捉事件发生在通道 2 下降沿 – 通道 2 被配置为输出时 (CH2CCS = 0x00) 0: 通道 2 输出高电平有效 1: 通道 2 输出低电平有效
[2]	CH1P	通道 1 捕捉 / 比较极性 – 通道 1 被配置为输入时 (CH1CCS = 0x01 / 0x02 / 0x03) 0: 捕捉事件发生在通道 1 上升沿 1: 捕捉事件发生在通道 1 下降沿 – 通道 1 被配置为输出时 (CH1CCS = 0x00) 0: 通道 1 输出高电平有效 1: 通道 1 输出低电平有效
[0]	CH0P	通道 0 捕捉 / 比较极性 – 通道 0 被配置为输入时 (CH0CCS = 0x01 / 0x02 / 0x03) 0: 捕捉事件发生在通道 0 上升沿 1: 捕捉事件发生在通道 0 下降沿 – 通道 0 被配置为输出时 (CH0CCS = 0x00) 0: 通道 0 输出高电平有效 1: 通道 0 输出低电平有效

## 定时器 PDMA / 中断控制寄存器 – DICTR

该寄存器包含了定时器 PDMA 和中断使能控制位。

偏移量: 0x074

复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位					TEVDE	保留位	UEVDE
						RW 0		RW 0
类型 / 复位	23	22	21	20	19	18	17	16
	保留位				CH3CCDE	CH2CCDE	CH1CCDE	CH0CCDE
					RW 0	RW 0	RW 0	RW 0
类型 / 复位	15	14	13	12	11	10	9	8
	保留位					TEVIE	保留位	UEVIE
						RW 0		RW 0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位				CH3CCIE	CH2CCIE	CH1CCIE	CH0CCIE
					RW 0	RW 0	RW 0	RW 0

位	字段	描述
[26]	TEVDE	触发事件 PDMA 请求使能位 0: 除能触发 PDMA 请求 1: 使能触发 PDMA 请求
[24]	UEVDE	更新事件 PDMA 请求使能位 0: 除能更新事件 PDMA 请求 1: 使能更新事件 PDMA 请求
[19]	CH3CCDE	通道 3 捕捉 / 比较 PDMA 请求使能位 0: 除能通道 3 PDMA 请求 1: 使能通道 3 PDMA 请求
[18]	CH2CCDE	通道 2 捕捉 / 比较 PDMA 请求使能位 0: 除能通道 2 PDMA 请求 1: 使能通道 2 PDMA 请求
[17]	CH1CCDE	通道 1 捕捉 / 比较 PDMA 请求使能位 0: 除能通道 1 PDMA 请求 1: 使能通道 1 PDMA 请求
[16]	CH0CCDE	通道 0 捕捉 / 比较 PDMA 请求使能位 0: 除能通道 0 PDMA 请求 1: 使能通道 0 PDMA 请求
[10]	TEVIE	触发事件中断使能位 0: 除能触发事件中断 1: 使能触发事件中断
[8]	UEVIE	更新事件中断使能位 0: 除能更新事件中断 1: 使能更新事件中断
[3]	CH3CCIE	通道 3 捕捉 / 比较中断使能位 0: 除能通道 3 中断 1: 使能通道 3 中断
[2]	CH2CCIE	通道 2 捕捉 / 比较中断使能位 0: 除能通道 2 中断 1: 使能通道 2 中断



位	字段	描述
[1]	CH1CCIE	通道 1 捕捉 / 比较中断使能位 0: 除能通道 1 中断 1: 使能通道 1 中断
[0]	CH0CCIE	通道 0 捕捉 / 比较中断使能位 0: 除能通道 0 中断 1: 使能通道 0 中断

## 定时器事件发生器寄存器 – EVGR

该寄存器包含了软件事件发生控制位。

偏移量: 0x078

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位					TEVG	保留位	UEVG
类型 / 复位					WO 0	WO 0	WO 0
7	6	5	4	3	2	1	0
保留位				CH3CCG	CH2CCG	CH1CCG	CH0CCG
类型 / 复位				WO 0	WO 0	WO 0	WO 0

位	字段	描述
[10]	TEVG	触发事件发生 触发事件 TEV 可由置高此位来发生。它由硬件自动清零。 0: 无动作 1: TEVIF 标志位置位
[8]	UEVG	更新事件发生 更新事件 UEV 可由置高此位来发生。它由硬件自动清零。 0: 无动作 1: 初始化计数器 计数器的值回到 0 或 CRR 预载值, 这取决于当前定时器使用的计数模式。相关寄存器也会被更新。详细描述请参考相关章节。
[3]	CH3CCG	通道 3 捕捉 / 比较发生 通道 3 捕捉 / 比较匹配事件可由置高此位来发生。它由硬件自动清零。 0: 无动作 1: 通道 3 发生捕捉 / 比较匹配事件 如果通道 3 被配置为输入, 计数器的值将被捕捉到 CH3CCR 寄存器, 接着 CH3CCIF 位被置位。如果通道 3 被配置为输出, 则 CH3CCIF 被置位。

位	字段	描述
[2]	CH2CCG	通道 2 捕捉 / 比较发生 通道 2 捕捉 / 比较匹配事件可由置高此位来发生。它由硬件自动清零。 0: 无动作 1: 通道 2 发生捕捉 / 比较匹配事件 如果通道 2 被配置为输入，计数器的值将被捕捉到 CH2CCR 寄存器，接着 CH2CCIF 位被置位。如果通道 2 被配置为输出，则 CH2CCIF 被置位。
[1]	CH1CCG	通道 1 捕捉 / 比较发生 通道 1 捕捉 / 比较匹配事件可由置高此位来发生。它由硬件自动清零。 0: 无动作 1: 通道 1 发生捕捉 / 比较匹配事件 如果通道 1 被配置为输入，计数器的值将被捕捉到 CH1CCR 寄存器，接着 CH1CCIF 位被置位。如果通道 1 被配置为输出，则 CH1CCIF 被置位。
[0]	CH0CCG	通道 0 捕捉 / 比较发生 通道 0 捕捉 / 比较匹配事件可由置高此位来发生。它由硬件自动清零。 0: 无动作 1: 通道 0 发生捕捉 / 比较匹配事件 如果通道 0 被配置为输入，计数器的值将被捕捉到 CH0CCR 寄存器，接着 CH0CCIF 位被置位。如果通道 0 被配置为输出，则 CH0CCIF 被置位。

## 定时器中断状态寄存器 – INTSR

该寄存器存储了定时器中断的状态。

偏移量: 0x07C

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位							
15	14	13	12	11	10	9	8
保留位					TEVIF	保留位	UEVIF
					W0C 0		W0C 0
7	6	5	4	3	2	1	0
CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF
W0C 0	W0C 0	W0C 0	W0C 0	W0C 0	W0C 0	W0C 0	W0C 0

位	字段	描述
[10]	TEVIF	触发事件中断标志位 此位在触发事件发生时通过硬件置位并通过软件清零。 0: 无触发事件发生 1: 触发事件发生
[8]	UEVIF	更新事件中断标志位 此位在更新事件发生时通过硬件置位并通过软件清零。 0: 无更新事件发生 1: 更新事件发生 注: 更新事件在以下情况下发生: – 计数器上溢或下溢 – UEVG 位被置位 – 来自从机触发输入的重启触发事件发生
[7]	CH3OCF	通道 3 过度捕捉标志位 此位由硬件置位, 由软件清零。 0: 没有检测到过度捕捉事件 1: 当 CH3CCIF 位已被置位且还没有被软件清零时, 捕捉事件再次发生。
[6]	CH2OCF	通道 2 过度捕捉标志位 此位由硬件置位, 由软件清零。 0: 没有检测到过度捕捉事件 1: 当 CH2CCIF 位已被置位且还没有被软件清零时, 捕捉事件再次发生。
[5]	CH1OCF	通道 1 过度捕捉标志位 此位由硬件置位, 由软件清零。 0: 没有检测到过度捕捉事件 1: 当 CH1CCIF 位已被置位且还没有被软件清零时, 捕捉事件再次发生。
[4]	CH0OCF	通道 0 过度捕捉标志位 此位由硬件置位, 由软件清零。 0: 没有检测到过度捕捉事件 1: 当 CH0CCIF 位已被置位且还没有被软件清零时, 捕捉事件再次发生。

位	字段	描述
[3]	CH3CCIF	<p>通道 3 捕捉 / 比较中断标志位</p> <p>– 通道 3 配置为输出时：</p> <p>    0：无匹配事件发生</p> <p>    1：计数器 CNTR 内容与 CH3CCR 寄存器内容匹配</p> <p>除中心对齐计数模式外，当计数器的值与 CH3CCR 的值匹配时，此位被硬件置位。通过软件清零。</p> <p>– 通道 3 配置为输入时：</p> <p>    0：无输入捕捉发生</p> <p>    1：输入捕捉发生</p> <p>此位在捕捉事件发生时被硬件置位。通过软件或通过读取 CH3CCR 寄存器清零。</p>
[2]	CH2CCIF	<p>通道 2 捕捉 / 比较中断标志位</p> <p>– 通道 2 配置为输出时：</p> <p>    0：无匹配事件发生</p> <p>    1：计数器 CNTR 内容与 CH2CCR 寄存器内容匹配</p> <p>除中心对齐计数模式外，当计数器的值与 CH2CCR 的值匹配时，此位被硬件置位。通过软件清零。</p> <p>– 通道 2 配置为输入时：</p> <p>    0：无输入捕捉发生</p> <p>    1：输入捕捉发生</p> <p>此位在捕捉事件发生时被硬件置位。通过软件或通过读取 CH2CCR 寄存器清零。</p>
[1]	CH1CCIF	<p>通道 1 捕捉 / 比较中断标志位</p> <p>– 通道 1 配置为输出时：</p> <p>    0：无匹配事件发生</p> <p>    1：计数器 CNTR 内容与 CH1CCR 寄存器内容匹配</p> <p>除中心对齐计数模式外，当计数器的值与 CH1CCR 的值匹配时，此位被硬件置位。通过软件清零。</p> <p>– 通道 1 配置为输入时：</p> <p>    0：无输入捕捉发生</p> <p>    1：输入捕捉发生</p> <p>此位在捕捉事件发生时被硬件置位。通过软件或通过读取 CH1CCR 寄存器清零。</p>
[0]	CH0CCIF	<p>通道 0 捕捉 / 比较中断标志位</p> <p>– 通道 0 配置为输出时：</p> <p>    0：无匹配事件发生</p> <p>    1：计数器 CNTR 内容与 CH0CCR 寄存器内容匹配</p> <p>除中心对齐计数模式外，当计数器的值与 CH0CCR 的值匹配时，此位被硬件置位。通过软件清零。</p> <p>– 通道 0 配置为输入时：</p> <p>    0：无输入捕捉发生</p> <p>    1：输入捕捉发生</p> <p>此位在捕捉事件发生时被硬件置位。通过软件或通过读取 CH0CCR 寄存器清零。</p>

## 定时器计数器寄存器 – CNTR

该寄存器存储了定时器计数器的值。

偏移量: 0x080

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CNTV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CNTV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CNTV	计数器的值

定时器预分频器寄存器 – PSCR

该寄存器定义了定时器预分频器的值以产生计数器时钟。

偏移量: 0x084  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	PSCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PSCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	PSCV	预分频器的值 这些位用来定义预分频器的值以产生计数器的时钟频率 $f_{CK\_CNT}$ 。 $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}$ , $f_{CK\_PSC}$ 代表预分频器时钟源。

定时器计数器重载寄存器 – CRR

该寄存器定义了定时器计数器重载值。

偏移量： 0x088  
复位值： 0x0000\_FFFF

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位									
类型 / 复位										
	15	14	13	12	11	10	9	8		
	CRV									
类型 / 复位	RW	1	RW	1	RW	1	RW	1	RW	1
	7	6	5	4	3	2	1	0		
	CRV									
类型 / 复位	RW	1	RW	1	RW	1	RW	1	RW	1

位	字段	描述
[15:0]	CRV	计数器重载值 CRV 是加载到实际计数器寄存器中的重载值。

通道 0 捕捉 / 比较寄存器 – CH0CCR

该寄存器定义了定时器通道 0 捕捉 / 比较值。

偏移量: 0x090  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH0CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH0CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH0CCV	通道 0 捕捉 / 比较值 – 当通道 0 配置为输出时 CH0CCR 值和计数器的值相比较，比较结果用于触发 CH0OREF 输出信号。 – 当通道 0 配置为输入时 CH0CCR 寄存器存储由上一次通道 0 捕捉事件捕捉到的计数器值。



通道 1 捕捉 / 比较寄存器 – CH1CCR

该寄存器定义了定时器通道 1 捕捉 / 比较值。

偏移量： 0x094  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH1CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH1CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH1CCV	通道 1 捕捉 / 比较值 – 当通道 1 配置为输出时 CH1CCR 值和计数器的值相比较，比较结果用于触发 CH1OREF 输出信号。 – 当通道 1 配置为输入时 CH1CCR 寄存器存储由上一次通道 1 捕捉事件捕捉到的计数器值。

通道 2 捕捉 / 比较寄存器 – CH2CCR

该寄存器定义了定时器通道 2 捕捉 / 比较值。

偏移量： 0x098

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH2CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH2CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH2CCV	通道 2 捕捉 / 比较值 – 当通道 2 配置为输出时 CH2CCR 值和计数器的值相比较，比较结果用于触发 CH2OREF 输出信号。 – 当通道 2 配置为输入时 CH2CCR 寄存器存储由上一次通道 2 捕捉事件捕捉到的计数器值。

通道 3 捕捉 / 比较寄存器 – CH3CCR

该寄存器定义了定时器通道 3 捕捉 / 比较值。

偏移量： 0x09C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH3CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH3CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH3CCV	通道 3 捕捉 / 比较值 – 当通道 3 配置为输出时 CH3CCR 值和计数器的值相比较，比较结果用于触发 CH3OREF 输出信号。 – 当通道 3 配置为输入时 CH3CCR 寄存器存储由上一次通道 3 捕捉事件捕捉到的计数器值。

通道 0 非对称比较寄存器 – CH0ACR

该寄存器定义了定时器通道 0 非对称比较值。

偏移量： 0x0A0  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH0ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH0ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH0ACV	通道 0 非对称比较值 当通道 0 被配置为非对称 PWM 模式且计数器向下计数时，写入该寄存器的值将与计数器的值进行比较。

通道 1 非对称比较寄存器 – CH1ACR

该寄存器定义了定时器通道 1 非对称比较值。

偏移量： 0x0A4  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH1ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH1ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH1ACV	通道 1 非对称比较值 当通道 1 被配置为非对称 PWM 模式且计数器向下计数时，写入该寄存器的值将与计数器的值进行比较。

通道 2 非对称比较寄存器 – CH2ACR

该寄存器定义了定时器通道 2 非对称比较值。

偏移量： 0x0A8  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	CH2ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
类型 / 复位	CH2ACV								
	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH2ACV	通道 2 非对称比较值 当通道 2 被配置为非对称 PWM 模式且计数器向下计数时，写入该寄存器的值将与计数器的值进行比较。

通道 3 非对称比较寄存器 – CH3ACR

该寄存器定义了定时器通道 3 非对称比较值。

偏移量： 0x0AC  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH3ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH3ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH3ACV	通道 3 非对称比较值 当通道 3 被配置为非对称 PWM 模式且计数器向下计数时，写入该寄存器的值将与计数器的值进行比较。

# 15 基本功能定时器 (BFTM)

## 简介

基本功能定时器模块 (BFTM) 是一个 32-bit 向上计数型计数器，用来测量时间间隔并产生单次或重复中断。BFTM 有两种工作模式，即重复模式和单次模式。在内部比较器每次产生比较匹配事件时重复模式都会重启计数器。BFTM 也支持单次模式，当比较匹配事件发生时，计数器会被强制停止计数。

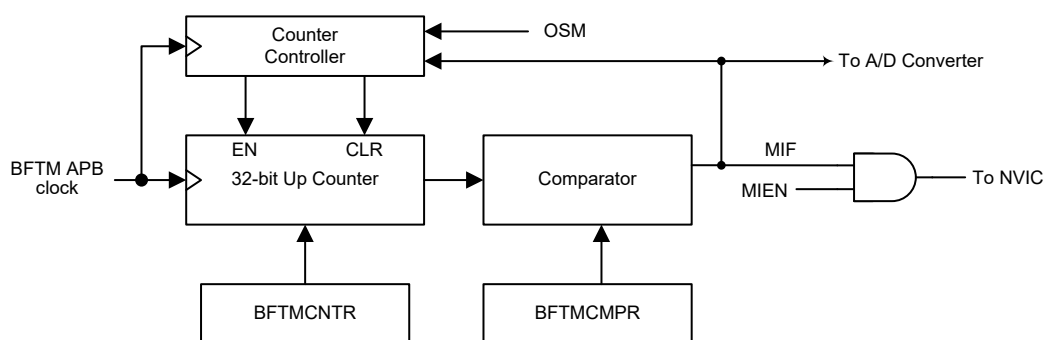


图 73. BFTM 方框图

## 特性

- 32-bit 向上计数型计数器
- 比较匹配功能
- 包含调试模式
- 时钟源：BFTM APB 时钟
- 运行时计数器值可读写
- 单次模式：计数器在比较匹配发生时停止计数
- 重复模式：计数器在比较匹配发生时重启
- 比较匹配中断使能 / 除能控制



## 功能描述

BFTM 是一个由 BFTM APB 时钟 (PCLK) 驱动的 32-bit 向上计数型计数器。任何时候计数器的值都可被更改或读取，即使定时器还在计数。BFTM 支持两种工作模式，即重复模式和单次模式，可测量时间间隔或时间周期。

### 重复模式

BFTM 从 0 开始向上计数到某一指定的比较值，该值由 BFTMCMPR 寄存器预定义。当 BFTM 工作于重复模式时且计数器值已达到 BFTMCMPR 寄存器指定的比较值，定时器将会产生一个比较匹配事件信号 MIF，此时计数器将被复位为 0 并重新开始计数。当 MIF 信号产生时，如果通过设置相关中断控制位 MIEN 为 1 使能比较匹配中断，则 BFTM 比较匹配中断将周期性地产生。如果通过清零 CEN 位除能计数器，计数器将停止计数并保持当前值不变。

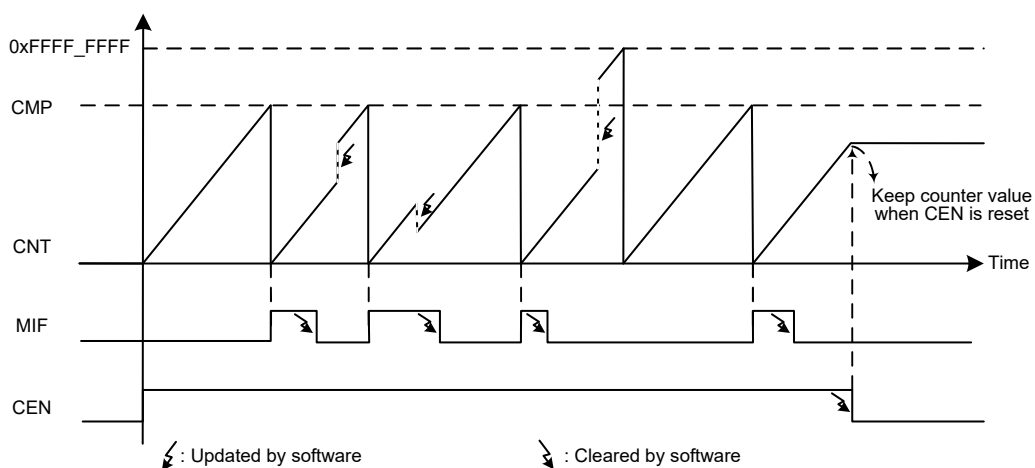


图 74. BFTM – 重复模式

## 单次模式

通过设置 BFTMCR 寄存器中的 OSM 位为 1, BFTM 将工作于单次模式。通过应用程序把 CEN 位置为 1 时, BFTM 开始计数。如果 CEN 位通过应用程序清零, 则计数器保持当前值不变。然而, 如果计数器比较匹配事件发生时 CEN 位通过内部硬件自动清零, 则计数器停止计数且其值将会被复位。

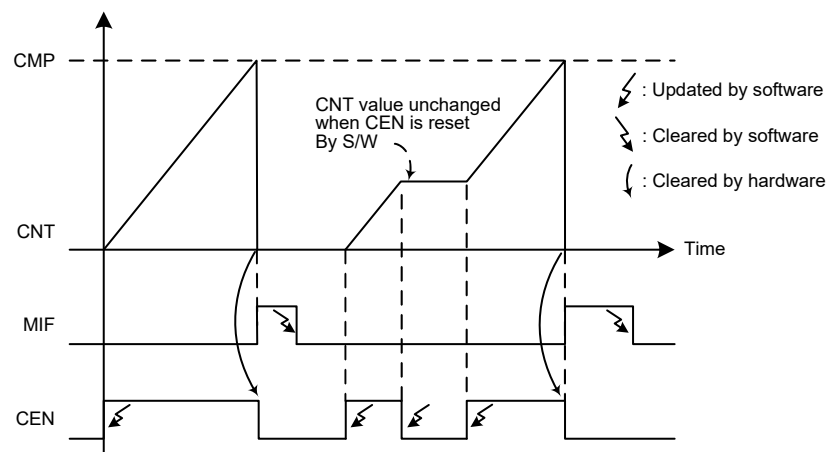


图 75. BFTM – 单次模式

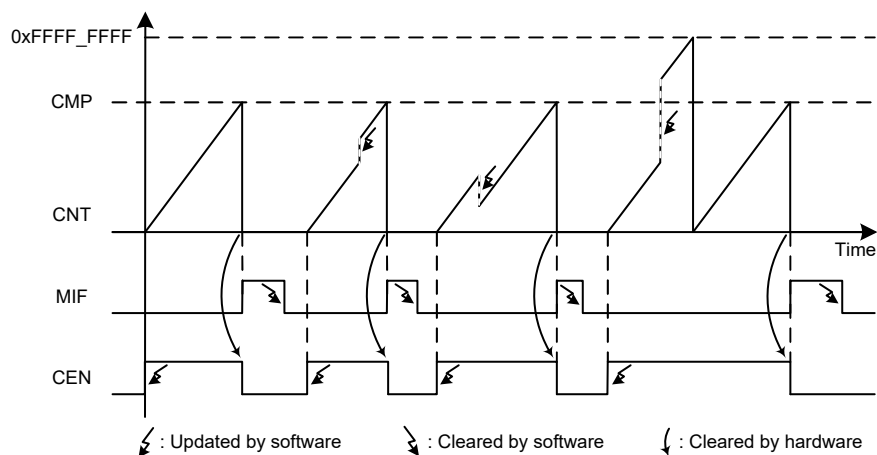


图 76. BFTM – 单次模式计数器更新

## 触发 A/D 转换器开启

当 BFTM 比较匹配事件发生时, 将会产生比较匹配事件中断标志 MIF, 可用作 A/D 转换器输入触发源。

## 寄存器列表

下表所示为 BFTM 寄存器及其复位值。

表 35. BFTM 寄存器列表

寄存器	偏移量	描述	复位值
BFTMCR	0x000	BFTM 控制寄存器	0x0000_0000
BFTMSR	0x004	BFTM 控制寄存器	0x0000_0000
BFTMCNTR	0x008	BFTM 计数器值寄存器	0x0000_0000
BFTMCMR	0x00C	BFTM 比较值寄存器	0xFFFF_FFFF

## 寄存器描述

### BFTM 控制寄存器 – BFTMCR

该寄存器定义了所有的 BFTM 控制位。

偏移量: 0x000

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位							
15	14	13	12	11	10	9	8
保留位							
7	6	5	4	3	2	1	0
保留位				CEN	OSM	MIEN	
				RW	0 RW	0 RW	0

位	字段	描述
[2]	CEN	BFTM 计数器使能控制位 0: BFTM 除能 1: BFTM 使能 该位置 1 时, BFTM 计数器开始计数。通过应用程序把 CEN 位清零时, 不管处于重复模式还是单次模式, 计数器都将停止计数且保持当前值不变。但在单次模式中, 如果是因为比较匹配事件发生, 定时器硬件电路把 CEN 位清零, 计数器将停止计数并复位为 0。
[1]	OSM	BFTM 单次模式选项 0: 计数器工作于重复模式 1: 计数器工作于单次模式
[0]	MIEN	BFTM 比较匹配中断使能控制位 0: 比较匹配中断除能 1: 比较匹配中断使能

**BFTM 状态寄存器 – BFTMSR**

该寄存器定义了 BFTM 的各种状态。

偏移量：0x004

复位值：0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位							MIF
类型 / 复位								W0C 0

位	字段	描述
[0]	MIF	BFTM 比较匹配中断标志位 0: 无比较匹配事件发生 1: 比较匹配事件发生 当计数器值 CNT 等于比较寄存器值 CMP 时，将会发生比较匹配事件，相关中断标志位 MIF 将被置位。写入 0 时，MIF 位被清零。

**BFTM 计数器值寄存器 – BFTMCNTR**

该寄存器定义了 BFTM 计数器的值。

偏移量: 0x008  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	CNT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	CNT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	CNT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CNT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	CNT	BFTM 计数器值 该字段在运行时都可读取或写入，用来存储 32-bit BFTM 计数器的值。

**BFTM 比较值寄存器 – BFTMCMPR**

该寄存器定义了 BFTM 的比较值。

偏移量: 0x00C  
复位值: 0xFFFF\_FFFF

	31	30	29	28	27	26	25	24
	CMP							
类型 / 复位	RW	1	RW	1	RW	1	RW	1
	23	22	21	20	19	18	17	16
	CMP							
类型 / 复位	RW	1	RW	1	RW	1	RW	1
	15	14	13	12	11	10	9	8
	CMP							
类型 / 复位	RW	1	RW	1	RW	1	RW	1
	7	6	5	4	3	2	1	0
	CMP							
类型 / 复位	RW	1	RW	1	RW	1	RW	1

位	字段	描述
[31:0]	CMP	BFTM 比较值 该寄存器用于指定与 BFTM 计数器值作比较的 32-bit BFTM 的比较值。

# 16 马达控制定时器 (MCTM)

## 简介

马达控制定时器由一个 16-bit 向上 / 向下计数器、四个 16-bit 捕捉 / 比较寄存器 (CCR)、一个 16-bit 计数器重载寄存器 (CRR)、一个 8-bit 重复计数器 (REPR) 和几个控制 / 状态寄存器组成。可用于多种用途, 包括通用计时、测量输入信号脉冲宽度或产生输出波形, 如单脉冲或 PWM 输出 (包含带死区时间插入的情况)。MCTM 支持使用正交解码器并带有两个输入口的编码器接口。

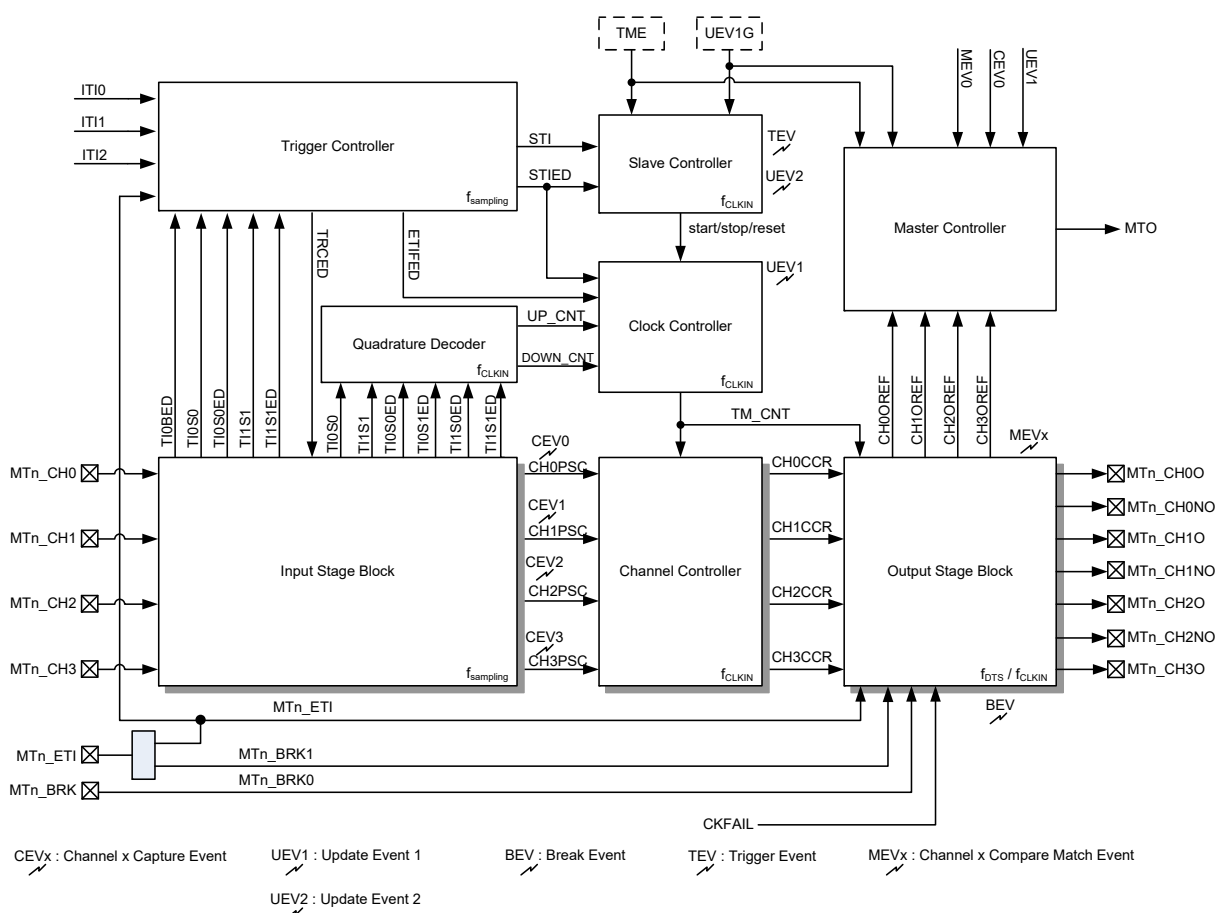


图 77. MCTM 方框图

## 特性

- 16-bit 向上 / 向下自动重载计数器
- 16-bit 可编程预分频器，可以对计数器时钟按 1 ~ 65536 之间的任意值分频
- 多达 4 个独立通道用于：
  - 输入捕捉功能
  - 比较匹配输出
  - PWM 波形生成 – 边沿对齐计数和中心对齐计数模式
  - 单脉冲模式输出
- 带有可编程死区时间插入的互补输出
- 使用正交解码器并带有两个输入口的编码器接口控制器
- 重复计数器仅在给定的计数周期后更新定时器寄存器
- 同步电路使用外部信号控制定时器并使其与其它定时器互相连接
- 下列事件发生时将产生中断 / PDMA：
  - 更新事件 1
  - 更新事件 2
  - 触发事件
  - 输入捕捉事件
  - 输出比较匹配事件
  - 刹车事件 – 仅产生中断
- MCTM 主机 / 从机模式控制器
- 支持三相马达控制和霍尔传感器接口
- 多达两个刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态



功能描述

计时器模式

向上计数

在此模式下，计数器从 0 连续向上计数，一直计数到 CRR 寄存器定义的计数器重载值。一旦计数器的值达到了计数器重载值，定时器模块将发生上溢事件，并从 0 开始重新计数。这一动作会反复执行。向上计数模式时，CNTCFR 寄存器中的计数方向位 DIR 应清零。

当通过设置 EVGR 寄存器中的 UEV1G 位为 1 发生更新事件 1 时，计数器的值将被初始化为 0。

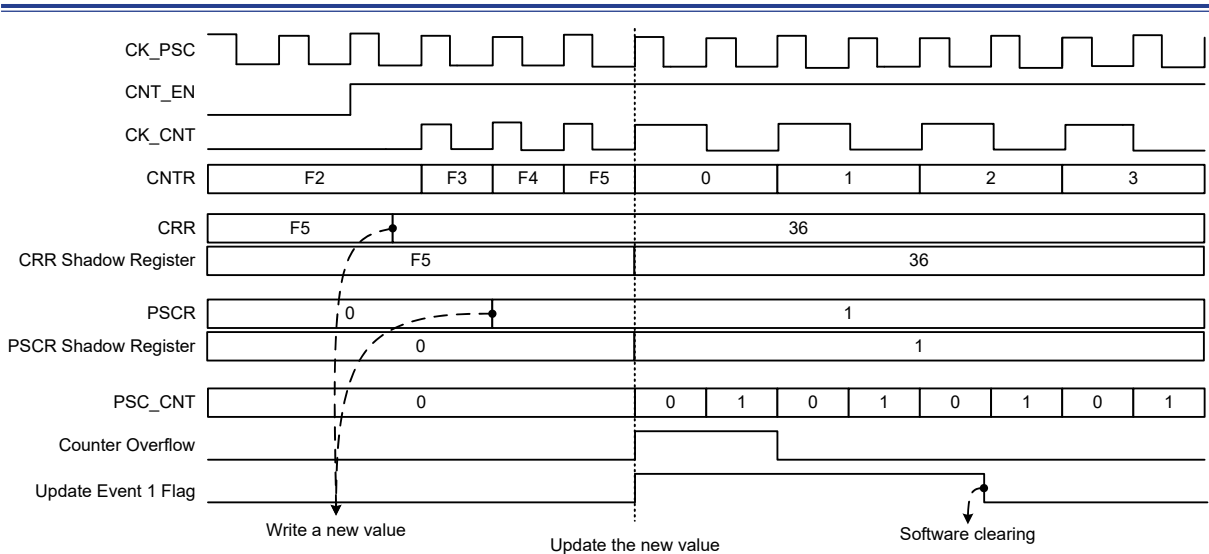


图 78. 向上计数范例

### 向下计数

在此模式下，计数器将从 CRR 寄存器定义的计数器重载值开始向下计数，一直计数到 0。一旦计数器的值达到 0，定时器模块将产生一个下溢事件，并从计数器重载值开始重新计数。这一动作会反复执行。向下计数模式时，CNTCFR 寄存器中的计数方向位 DIR 应置为 1。

当通过设置 EVGR 寄存器中的 UEV1G 位为 1 发生更新事件 1 时，计数器的值将被初始化为计数器重载值。

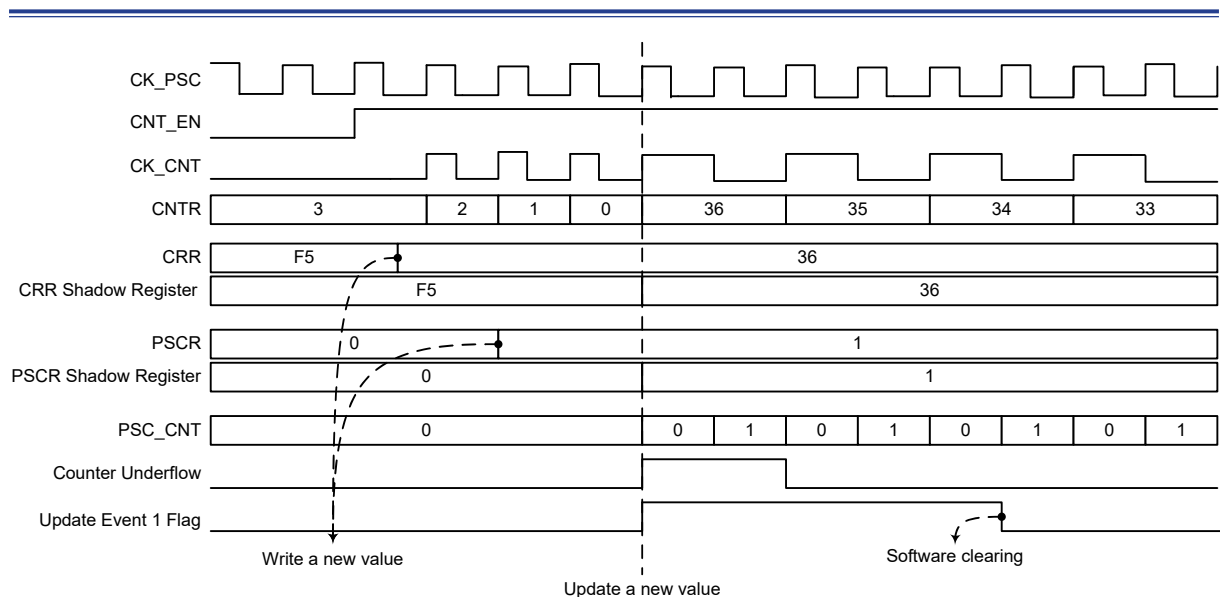


图 79. 向下计数范例

中心对齐计数

在中心对齐计数模式下，计数器将交替地从 0 计数到重载值而后从重载值计数到 0。在向上计数模式下，当计数器计数到计数器重载值时，计数器模块会产生一个上溢事件；而在向下计数模式下，当计数器计数到 0 时，计数器模块会产生一个下溢事件。CNTCFR 寄存器中的 DIR 位是只读位，其值表明了中心对齐计数模式下的计数方向。计数方向由硬件自动更新。

在中心对齐计数模式下，如果将 EVGR 寄存器中的 UEV1G 位置位，那么不管计数器是向上计数还是向下计数，计数器的值都将初始化为 0。

INTSR 寄存器中的 UEV1IF 位在何种情况下被置位取决于 CNTCFR 寄存器中 CMSEL 字段的设置。当 CMSEL 字段设置为向上计数模式时，上溢事件发生会使 UEV1IF 位被置为 1；当 CMSEL 字段设置为向下计数模式时，下溢事件发生会使 UEV1IF 位被置为 1；当 CMSEL 字段设置为中心对齐计数模式时，上溢或下溢事件发生都会使 UEV1IF 位被置为 1。

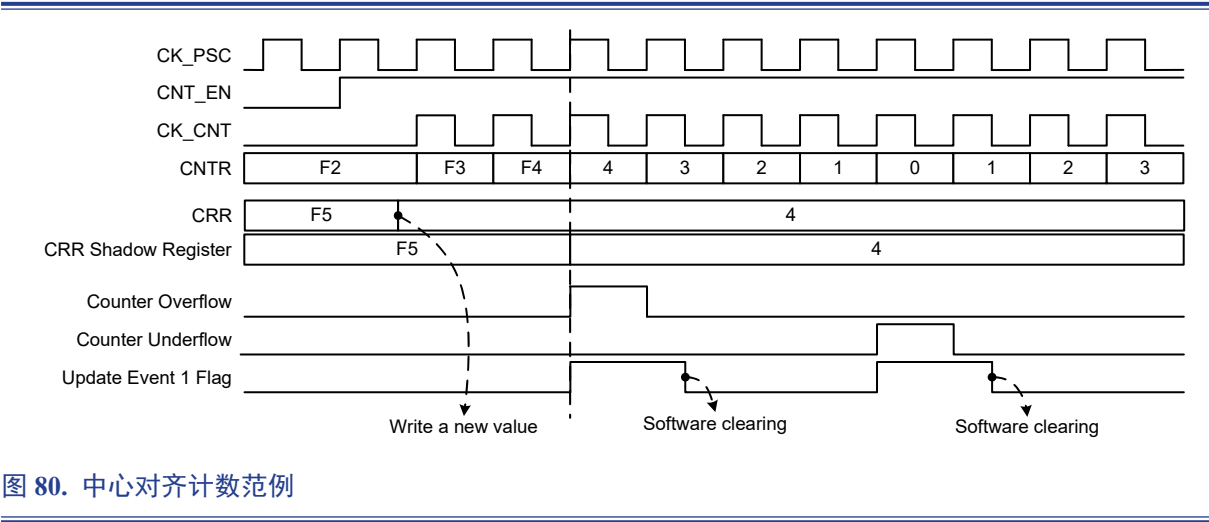


图 80. 中心对齐计数范例

重复向下计数器操作

更新事件 1 通常发生在上溢事件或下溢事件发生时。当分配一个非 0 值到 REPR 寄存器使重复操作有效时，更新事件仅在 REPR 计数器值已达到 0 时才发生。REPR 值在以下情况发生时递减：

- 向上计数模式下，每次计数器上溢时
- 向下计数模式下，每次计数器下溢时
- 中心对齐计数模式下，每次计数器上溢和下溢时

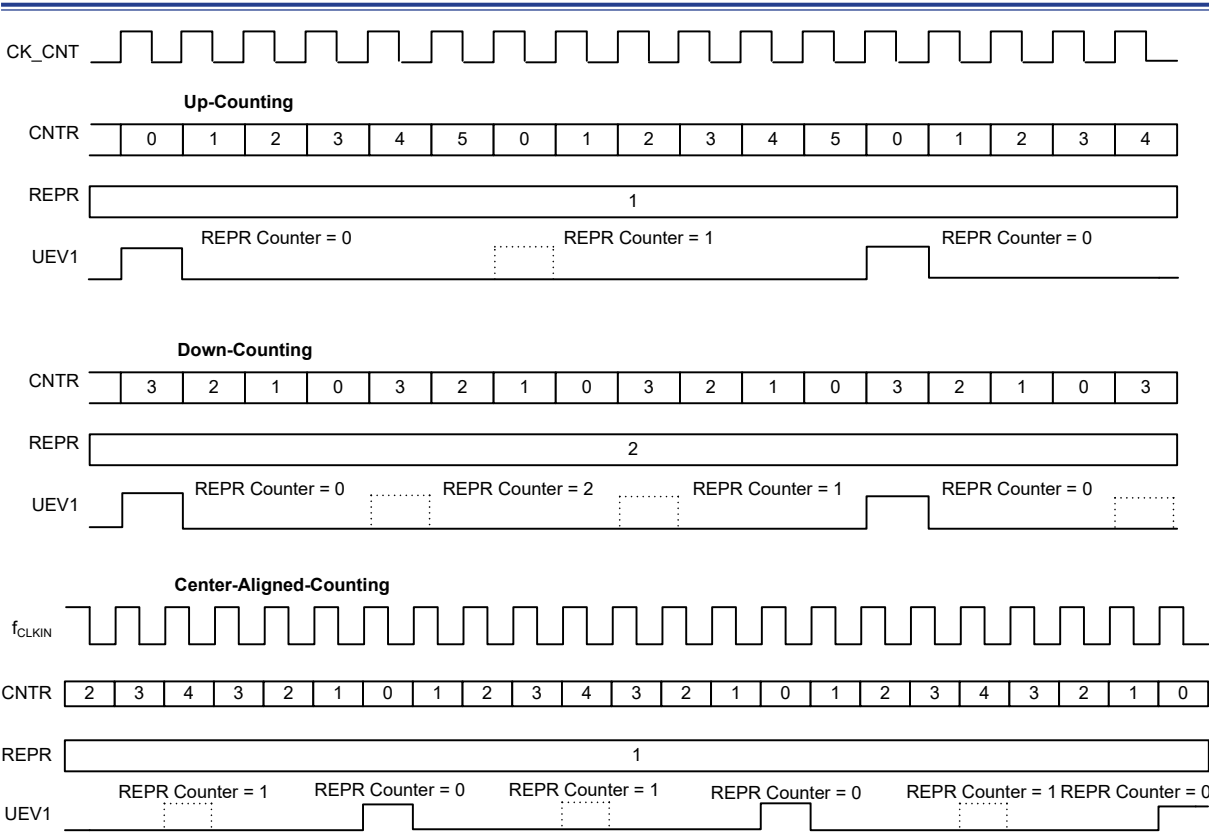


图 81. 更新事件 1 相关重复机制范例

## 时钟控制器

以下描述了定时器模块的时钟控制器，它用来选择内部预分频计数器的时钟源。

### ■ 内部 APB 时钟 $f_{CLKIN}$

当从机模式除能时，默认的内部时钟源是 APB 时钟  $f_{CLKIN}$ ，用来驱动计数器预分频器。当从机模式选择位 SMSEL 被置为 0x4、0x5 或 0x6 时，内部 APB 时钟  $f_{CLKIN}$  将作为驱动计数器预分频器的时钟源。

### ■ 正交解码器

要选择正交解码器模式，应把 MDCFR 寄存器中的 SMSEL 字段设为 0x1、0x2 或 0x3。正交解码器功能使用两个输入口，即 MTn\_CH0 和 MTn\_CH1 引脚来产生时钟脉冲以驱动计数器预分频器。在每一个输入源信号发生转换时，计数器方向位 DIR 的值都会被硬件自动修改。

### ■ STIED

计数器预分频器在每一个 STI 信号的上升沿期间计数。此模式可通过将 MDCFR 寄存器中的 SMSEL 字段设为 0x7 来选择。这时计数器将作为一个事件计数器使用。输入事件，即 STI，可通过把 TRSEL 字段设成除 0x0 以外的可用值来选择。当 STI 信号被选择作为时钟源使用时，在每一个 STI 信号上升沿作用期间，内部边沿检测电路将会产生一个时钟脉冲来驱动计数器预分频器。值得注意的是，如果 TRSEL 字段设成 0x0 来选择软件 UEV1G 位作为触发源，那么当 SMSEL 字段设成 0x7 时，计数器将会被更新而非计数。

### ■ ETIFED

在 ETIF 上升沿作用期间，计数器预分频器将被驱动用来计数。此模式可通过将 TRCFR 寄存器的 ECME 位设为 1 来选择。将 ETIF 信号选择作为时钟源的另一种方法是分别把 SMSEL 字段设成 0x7 和把 TRSEL 字段设成 0x3。当时钟源来自于 ETIF 信号时，含有边沿检测电路的触发控制器将会在每个 ETIF 信号上升沿作用期间产生一个时钟脉冲来作为计数器预分频器的时钟源。

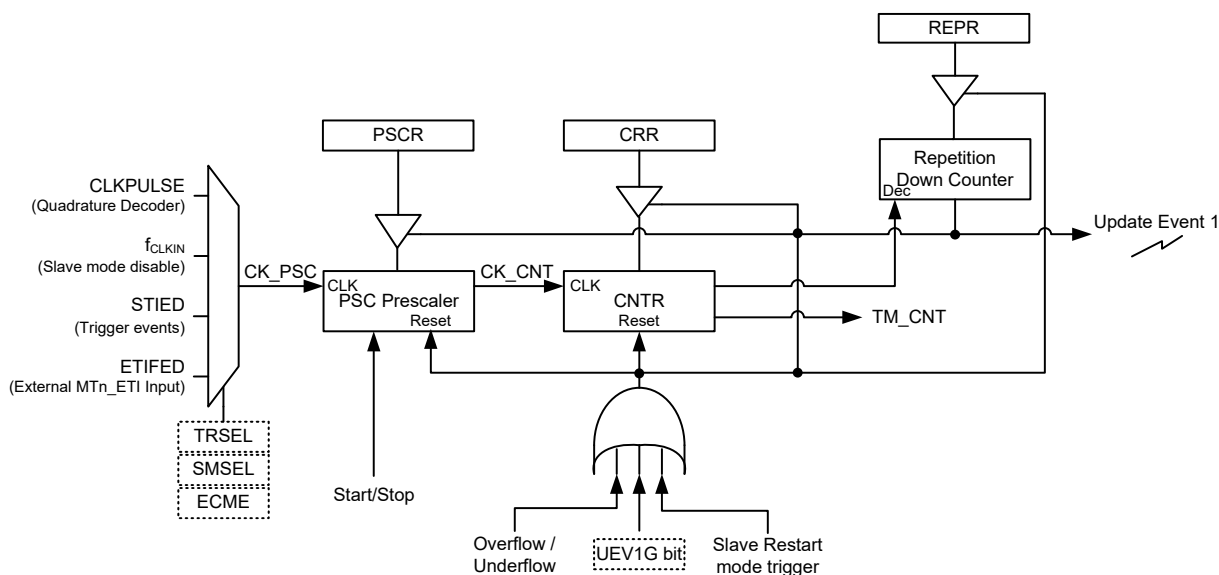


图 82. MCTM 时钟源选择

## 触发控制器

触发控制器用来选择触发源以及设置触发电平和边沿触发条件。外部触发输入信号 MTn\_ETI 的有效极性可由 MCTM 触发配置寄存器 TRCFR 中的外部触发极性控制位 ETIPOL 来进行配置。外部触发输入的频率可通过 TRCFR 寄存器中的外部触发预分频控制位 ETIPSC 来控制分频。对于特殊应用, 如果需要滤波信号时, 触发信号也可通过配置 TRCFR 寄存器中的外部触发滤波器 ETF 选择位进行滤波。内部触发输入, 可通过 TRCFR 寄存器中的触发选择位 TRSEL 进行选择。除了 UEV1G 位软件触发之外的所有触发源, 内部边沿检测电路将会在每个触发信号上升沿作用期间产生一个时钟脉冲, 以激活某些因触发信号上升沿而触发的 MCTM 功能。

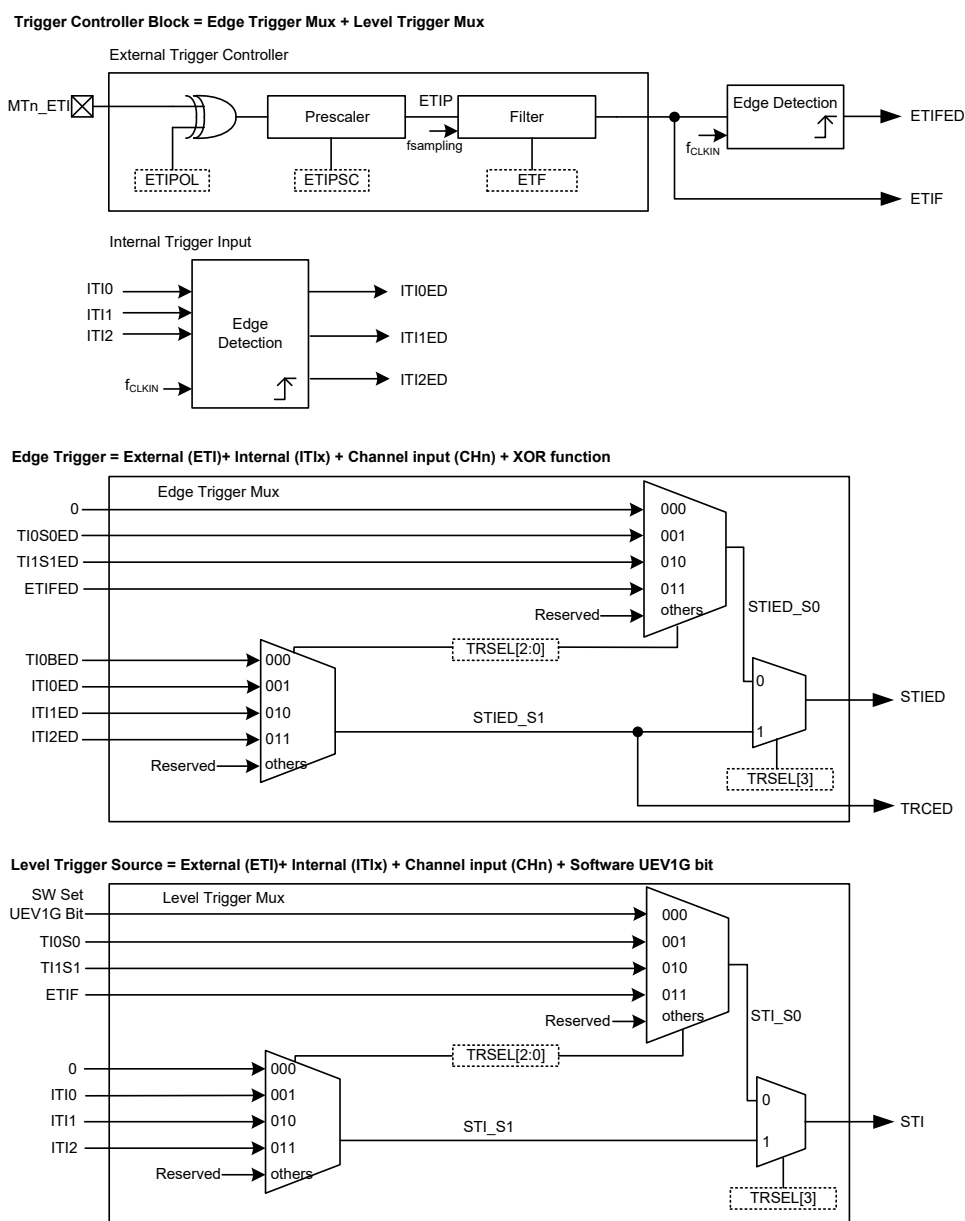


图 83. 触发控制器方框图

从机控制器

在几种模式下，MCTM 可以与一个内部 / 外部触发器进行同步。这些模式包括重启模式、暂停模式和触发模式，是通过 MDCFR 寄存器中的 SMSEL 字段选择的。这些模式的触发输入来自于 STI 信号，通过 TRCFR 寄存器中的 TRSEL 字段选择。从机控制器中的工作模式在相关章节中有所描述。

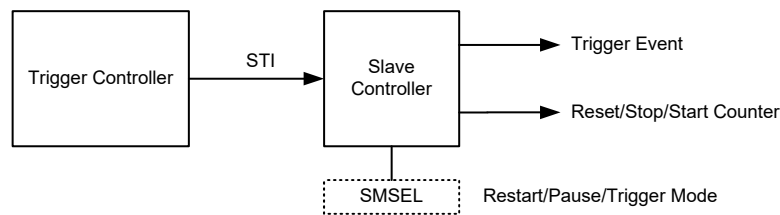


图 84. 从机控制器方框图

重启模式

作为对 STI 信号上升沿的响应，计数器及其预分频器会被重新初始化。当一个 STI 上升沿到来时，如果 UEV1DIS 位被置 1 来除能更新事件，那么更新事件将不会发生，然而当 STI 上升沿到来时，计数器和预分频器仍会被重新初始化。如果 CNTCFR 寄存器中的 UEV1DIS 位被清零来使能更新事件，则更新事件将伴随 STI 上升沿一起发生，所有预载寄存器将被更新。

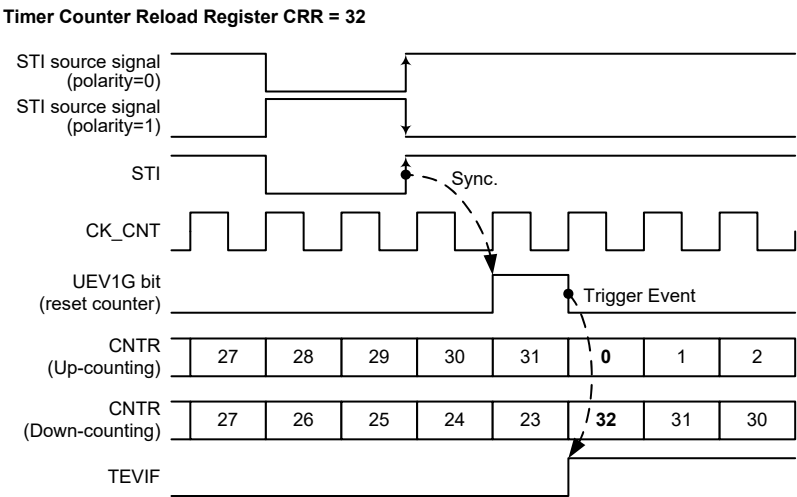


图 85. 重启模式下的 MCTM

### 暂停模式

在暂停模式下，所选择的 STI 输入信号电平用来控制计数器的开始 / 停止操作。当所选的 STI 信号处于高电平时，计数器开始计数；当 STI 信号转换为低电平时，计数器停止计数并保持当前值不变，且不会被复位。因为暂停功能是通过 STI 电平来控制计数器的开始 / 停止操作，所以所选择的 STI 信号不能来自于 TI0BED。

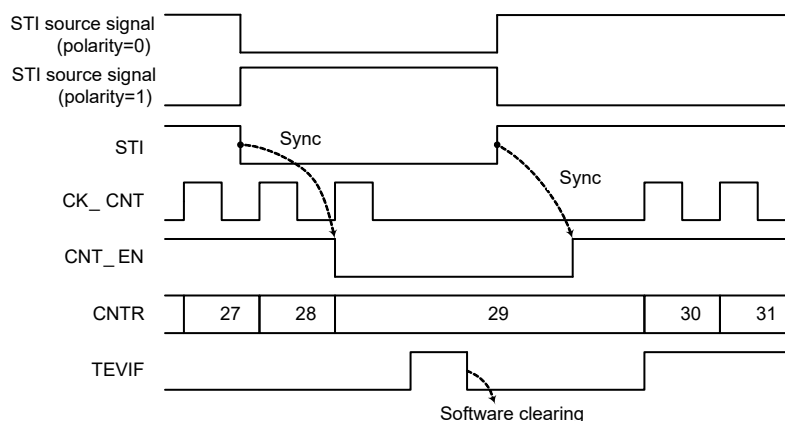


图 86. 暂停模式下的 MCTM

### 触发模式

在计数器停止计数后，当一个 STI 上升沿信号到来时，计数器将从当前值重新开始计数。注意，如果 STI 信号来自于 UEVIG 位软件触发，计数器不会重新计数。当 STI 源信号通过 UEVIG 位选择为软件触发时，不会产生使计数器重新计数的时钟脉冲。还要注意，STI 信号只用来使计数器重新计数，而没有使计数器停止计数的作用。

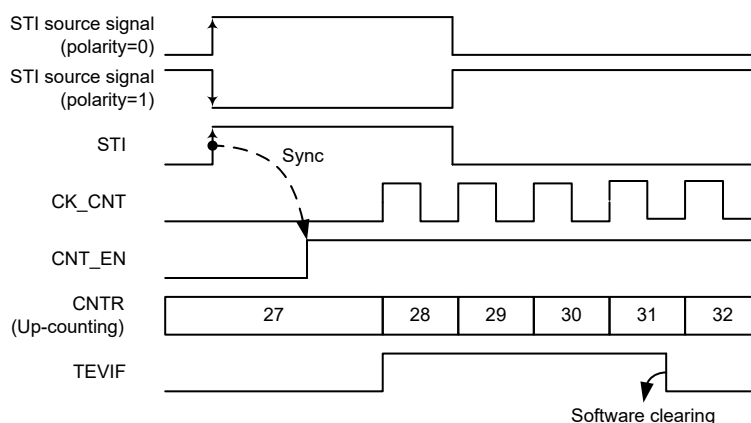


图 87. 触发模式下的 MCTM



## 主机控制器

MCTM 和 GPTM 可在内部连接在一起用于定时器同步或链接。当一个 MCTM 被配置在主机模式下时, MCTM 主机控制器将会产生一个主机触发输出 (MTO) 信号, 通过 MDCFR 寄存器中的 MMSEL 字段选择可以触发或驱动从机模式下的另一个 MCTM 或 GPTM 的定时器复位、重启、停止或为其提供时钟源。

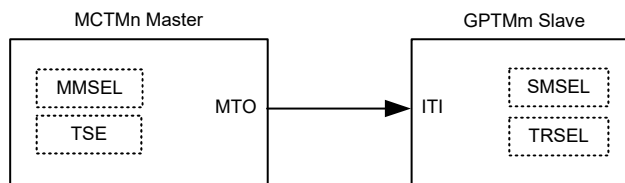


图 88. 主机 MCTMn 和从机 GPTMm 相连接

MDCFR 寄存器中的主机模式选择位 MMSEL 用来选择同步另外一个从机 MCTM 或 GPTM 的 MTO 源。

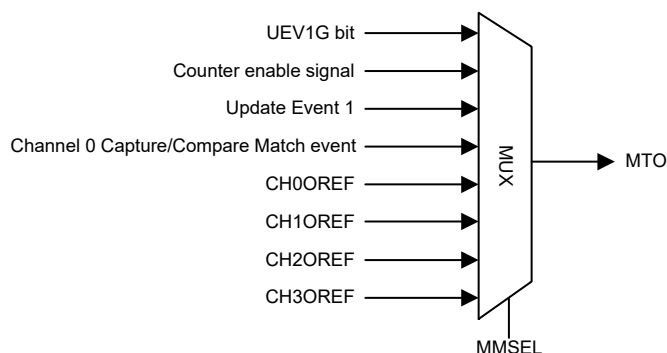


图 89. MTO 选项

例如, 把 MMSEL 字段设为 0x5, 以选择 CH1OREF 信号作为 MTO 信号来同步另外一个从机的 MCTM 或 GPTM。欲知更多详细描述, 请参考相关的 MDCFR 寄存器中的 MMSEL 字段定义。

## 通道控制器

MCTM 有四个独立的通道，用来选择作为捕捉输入或比较匹配输出。每个捕捉输入或比较匹配输出通道都由一个预载寄存器和一个影子寄存器组成。APB 总线是通过读 / 写预载寄存器来进行数据访问的。

在输入捕捉模式下使用时，计数器的值会首先被捕捉到 CHxCCR 影子寄存器中，捕捉事件发生时，其值会被传送到 CHxCCR 预载寄存器中。

在比较匹配输出模式下使用时，CHxCCR 预载寄存器的内容会被复制到相应的影子寄存器中，然后计数器的值会与寄存器的值进行比较。

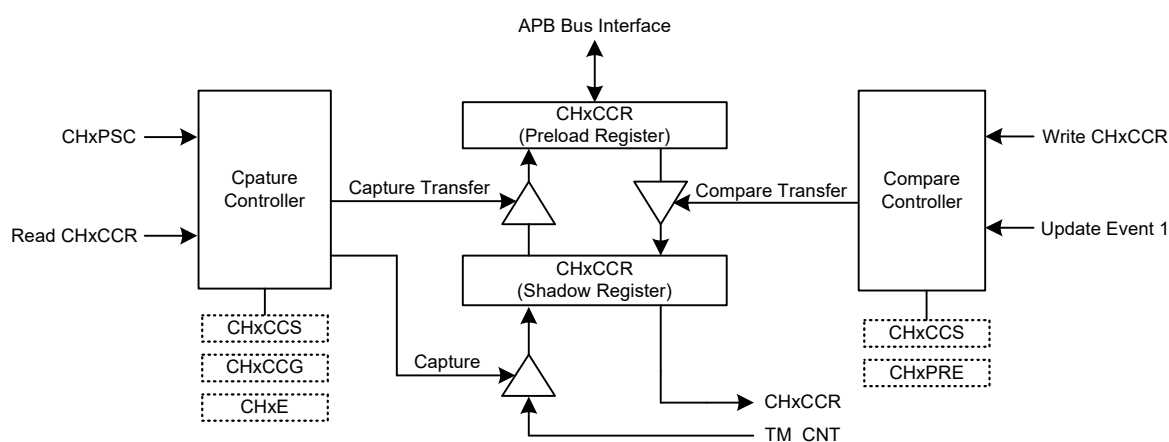


图 90. 捕捉 / 比较方框图

捕捉计数器值传送到 CHxCCR

当通道用作捕捉输入时，计数器的值会在有效输入信号转换发生时，被捕捉到通道捕捉 / 比较寄存器 (CHxCCR) 中。一旦捕捉事件发生，INTSR 寄存器中的 CHxCCIF 标志位会相应的被置位。如果 CHxCCIF 位已经被置位，即标志位还未被软件清零，而此通道的另外一个捕捉事件发生，则相应的通道过度捕捉标志位 CHxOCF 将被置位。

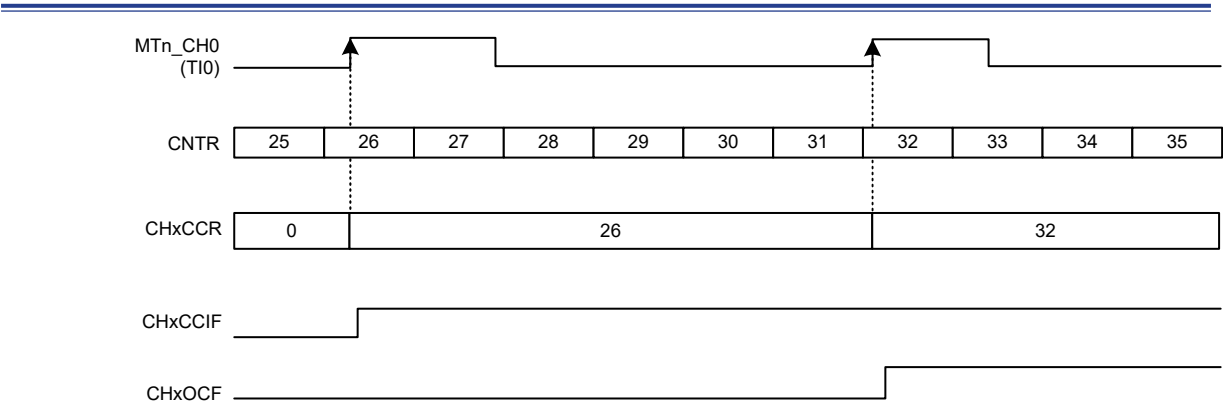


图 91. 输入捕捉模式

脉冲宽度测量

输入捕捉模式也可以用于 MTn\_CHx 引脚上的信号 TIx 脉冲宽度测量。下面的例子说明了在输入捕捉模式下，如何配置 MCTM 使用通道 0 和通道 1 测量在 MTn\_CH0 引脚上的高脉冲宽度和输入周期。基本步骤如下：

- 配置捕捉通道 0 (CH0CCS = 0x1)，选择 TI0 信号作为捕捉输入。
- 配置 CH0P 位为 0，选择 TI0 输入的上升沿作为有效极性。
- 配置捕捉通道 1 (CH1CCS = 0x2)，选择 TI0 信号作为捕捉输入。
- 配置 CH1P 位为 1，选择 TI0 输入的下降沿作为有效极性。
- 把 TRSEL 位置为 0x1 选择 TI0S0 作为触发输入。
- 把 MDCFR 寄存器的 SMSEL 字段置为 0x4，使从机控制器工作在重启模式下。
- 把 CHCTR 寄存器中的 CH0E 和 CH1E 位置为 1 来开启输入捕捉模式。

如下图所示，输入捕捉操作之后，MTn\_CH0 引脚上的高脉冲宽度将被捕捉到 CH1CCR 寄存器中，输入周期将被捕捉到 CH0CCR 寄存器中。

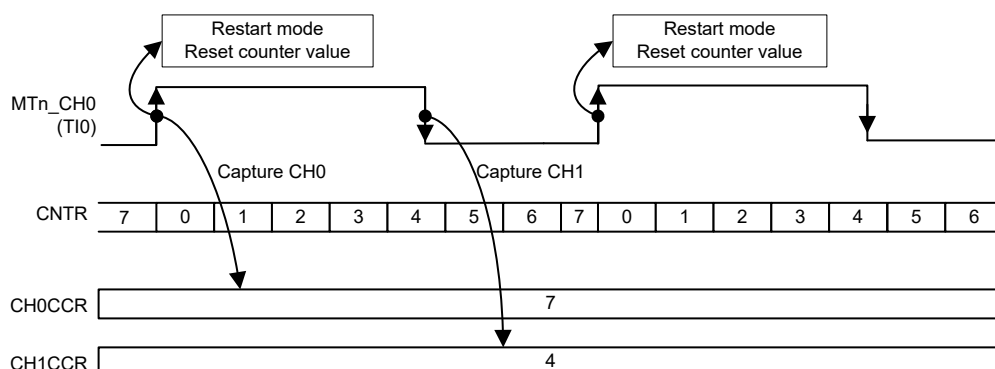


图 92. PWM Pulse 脉冲宽度测量范例

### 输入级

输入级由一个数字滤波器、一个通道极性选择器、边沿检测电路和一个通道预分频器组成。通道 0 输入信号 TI0 可以来自于 MTn\_CH0 信号或 MTn\_CH0、MTn\_CH1 和 MTn\_CH2 信号的异或。通道输入信号 TIx 被一个数字滤波器采样，产生一个滤波输入信号 TIxFP。通道极性和边沿检测模块可以产生一个 TIxSxED 信号提供给输入捕捉功能。有效输入事件数量可通过通道捕捉输入源预分频器设置字段 CHxPSC 设置。

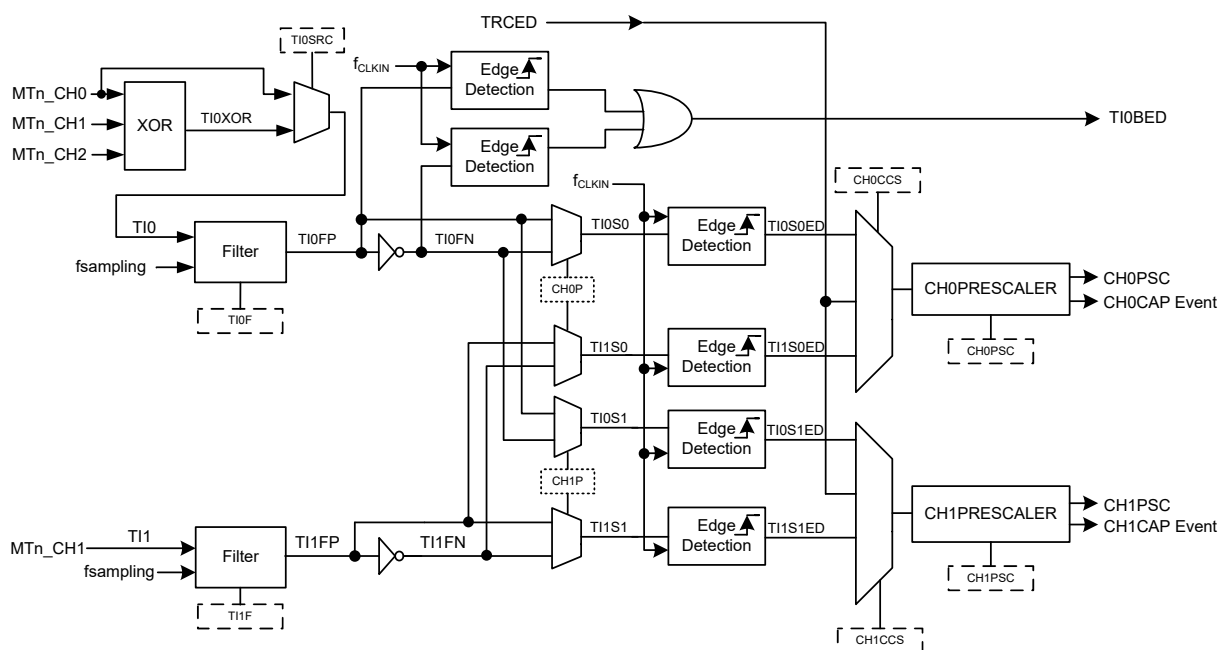


图 93. 通道 0 和通道 1 输入级

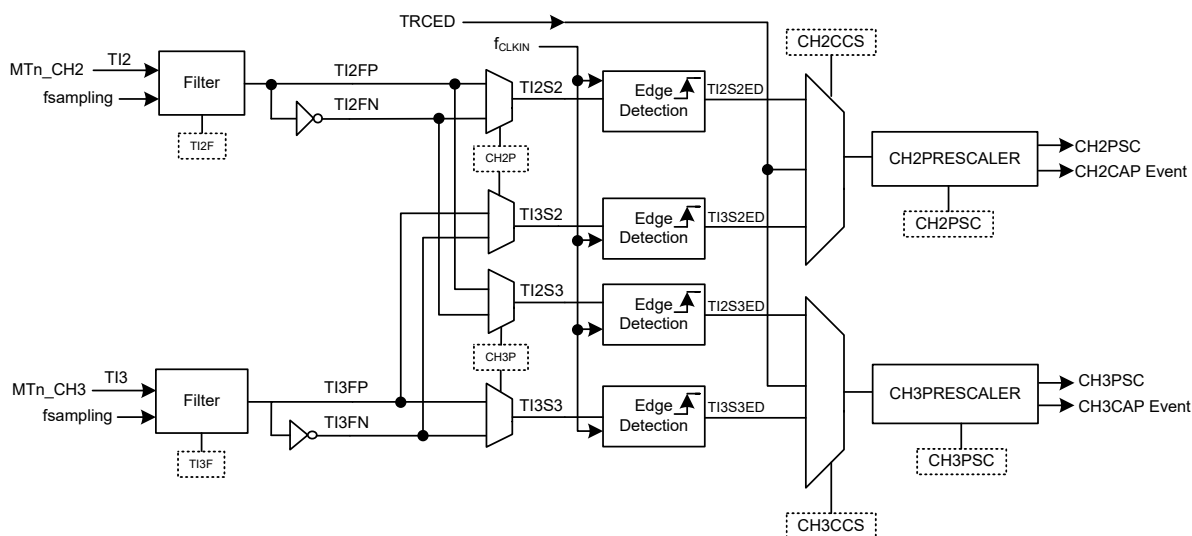


图 94. 通道 2 和通道 3 输入级

## 输出级

MCTM 支持带有死区时间插入的通道 0、1 和 2 的互补输出。除了刹车功能，MCTM 通道 3 的输出功能和 GPTM 通道 3 的输出功能几乎一样。

CHxO 和 CHxNO 通道输出参考信号是 CHxOREF 信号。如下方框图虚线所示，根据相关控制位的配置值，这些通道输出可产生各种波形。

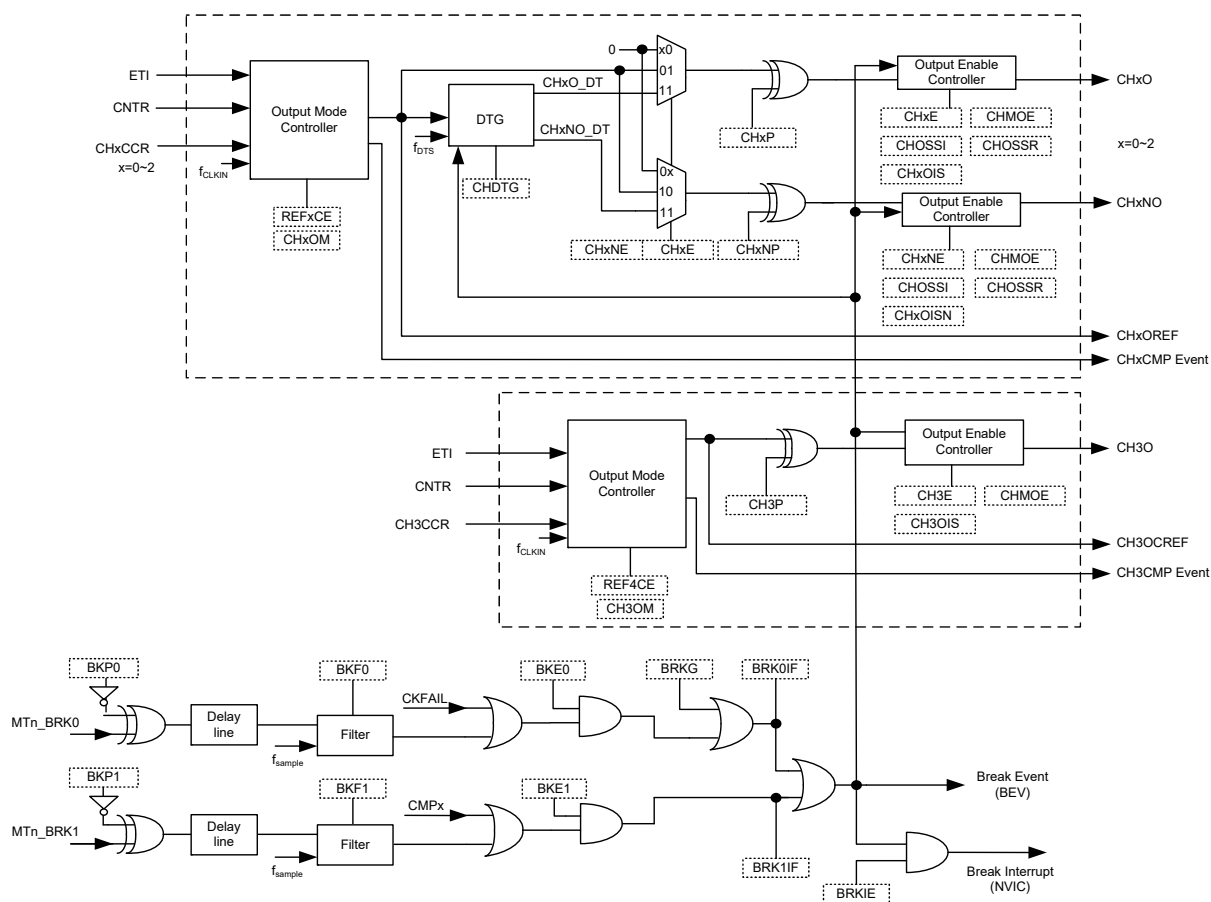


图 95. 输出级方框图

### 通道输出参考信号

当 MCTM 用在比较匹配输出模式时, CHxOREF 信号 (通道 x 输出参考信号) 通过设置 CHxOM 位来定义。CHxOREF 信号有几种输出功能类型, 这些类型包括当计数器的值与 CHxCCR 寄存器的内容匹配时, CHxOREF 输出可为低电平、高电平或者翻转。除此之外, 也有 PWM 模式 1 和 PWM 模式 2 输出。在这些模式中, CHxOREF 信号的电平都是根据计数方向以及计数器值与 CHxCCR 内容的关系而改变。还有两种模式, 不论计数器和 CHxCCR 的值是什么, 输出都会被强制为一个无效或有效的电平。至于更详细的说明请参考相关位的定义。输出类型设置如下表所示。

表 36. 比较匹配输出设置

CHxOM 值	比较匹配输出电平
0x00	无变化
0x01	输出 0
0x02	输出 1
0x03	输出翻转
0x04	强制无效电平
0x05	强制有效电平
0x06	PWM 模式 1
0x07	PWM 模式 2

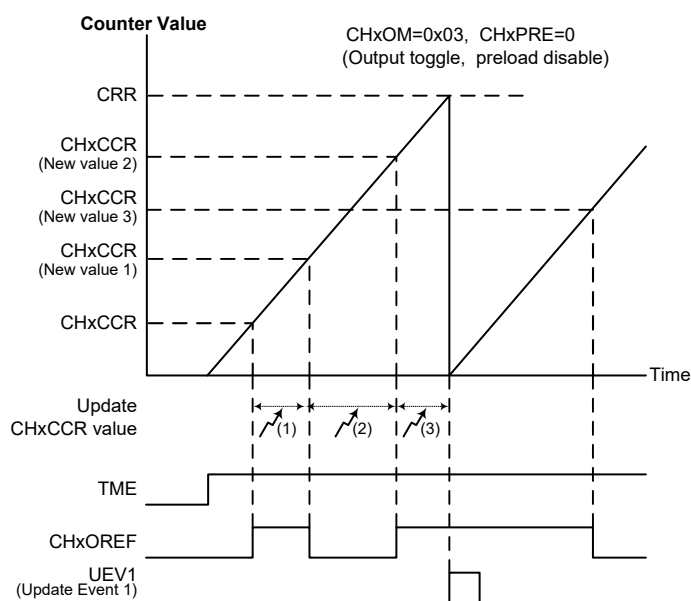


图 96. 翻转模式通道输出参考信号 – CHxPRE = 0

**Counter Value**

CRR

CHxCCR

**CHxOM = 0x06**

CHxOREF

CHxCCIF

**CHxOM = 0x07**

CHxOREF

**Counter Value**

CHxCCR

CRR

CHxCCR = 0x0000

CHxOREF

CHxCCIF

100%

0%

0%

100%

September 14, 2020



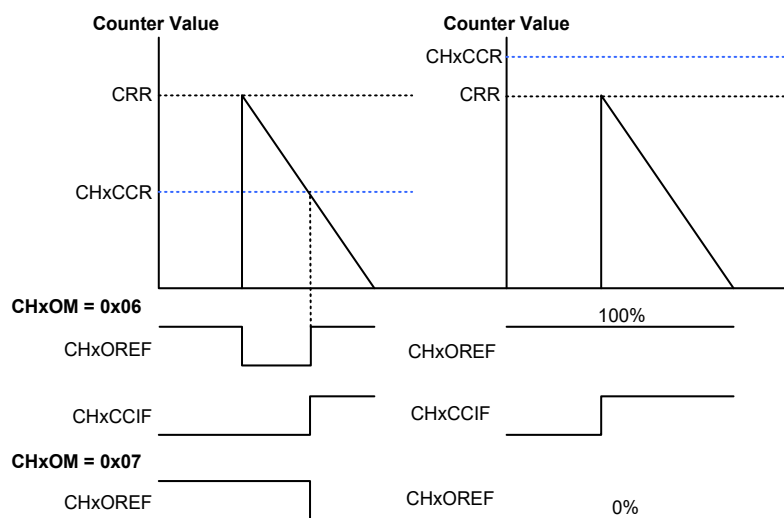


图 99. PWM 模式通道输出参考信号和向下计数模式下的计数器

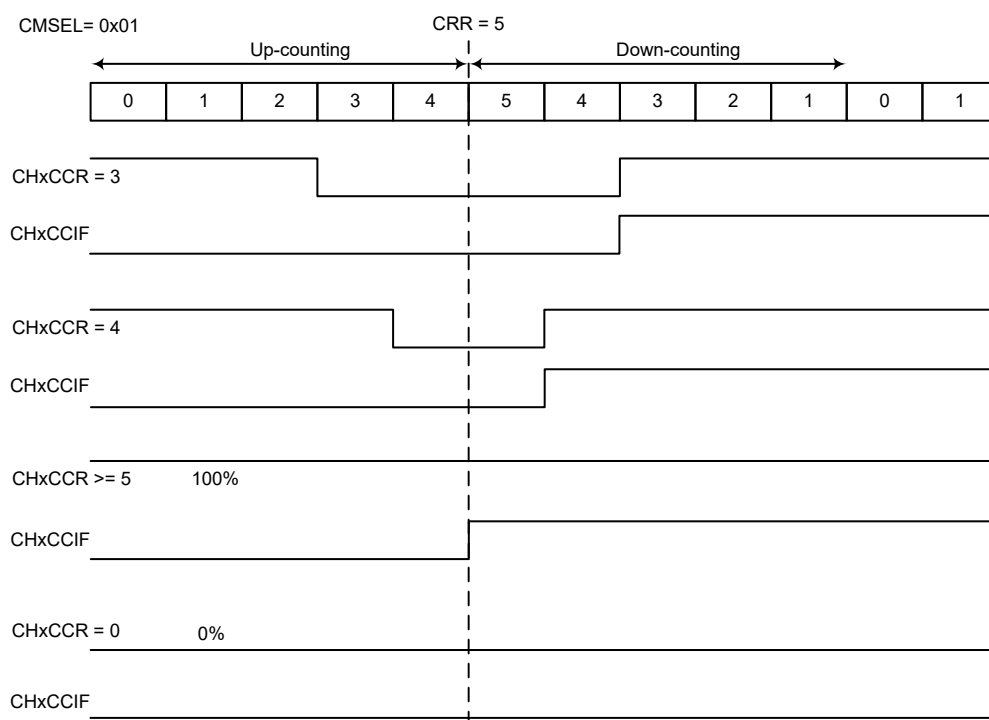


图 100. PWM 模式 1 通道输出参考信号和中心对齐计数模式下的计数器

### 死区时间发生器

通道 0~2 包含一个 8-bit 死区时间发生器功能。通过设置 CHxE 和 CHxNE 位可使得死区时间插入。CHxO 和 CHxNO 信号与 CHxOREF 信号的关系如下：

- CHxO 信号与 CHxOREF 信号相同，除了上升沿相对于参考信号上升沿有一个死区时间延迟。
- CHxNO 信号与 CHxOREF 信号相反，除了上升沿相对于参考信号下降沿有一个死区时间延迟。

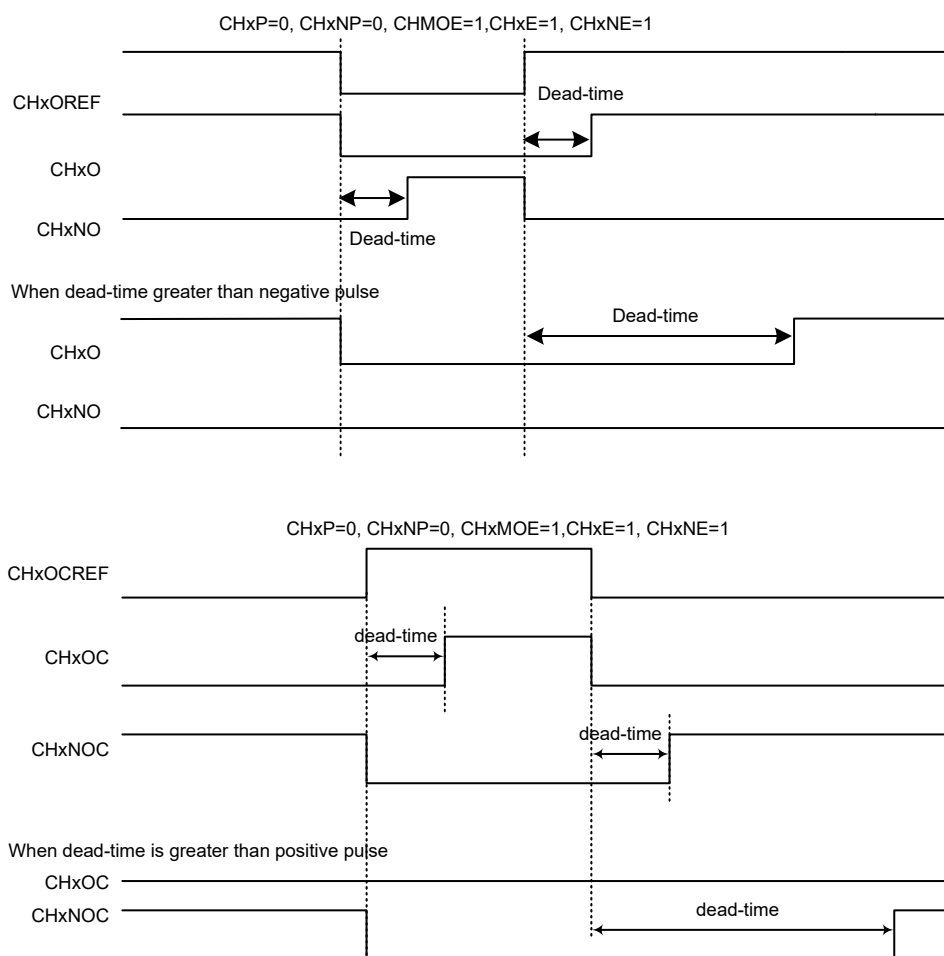


图 101. 带死区时间插入的互补输出

如果延迟比 CHxO 或 CHxNO 的有效输出宽度大，则不会产生相应的 PWM 脉冲。

## 刹车功能

MCTM 包含刹车功能并且最多有两个 MCTM 刹车输入信号。MTn\_BRK0 是该功能的默认刹车信号，来自外部 MTn\_BRK 引脚。而 MTn\_BRK1 是第二个刹车信号，与 MTn\_ETI 引脚共用，默认为关闭状态。通过设置 CHBRKCTR 寄存器中的 BRK1SEL 位可选择 MTn\_ETI 引脚用作额外的刹车信号 MTn\_BRK1。详细方框图如下所示。

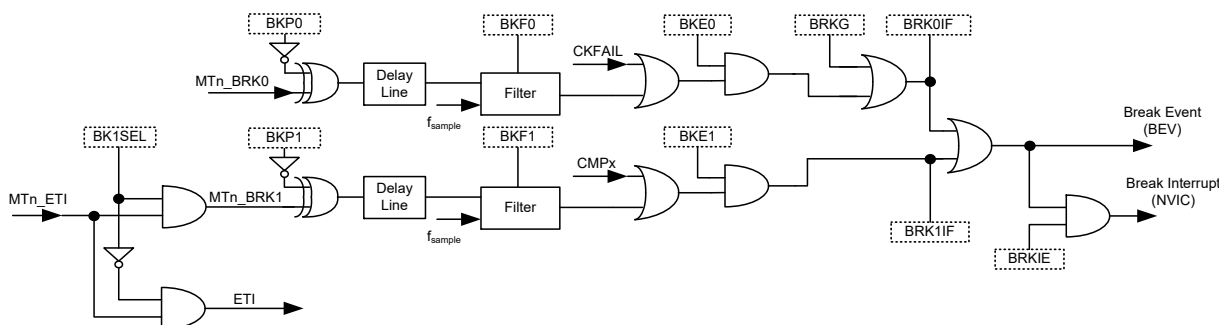


图 102. MCTM 刹车信号方框图

若刹车功能使能，当 MCTM 刹车输入信号输入一个有效电平或时钟监控电路检测到一个时钟失效事件或是比较器输入发生转变时。同时，每个通道输出被强制复位或处于无效或空闲状态。此外，即使在刹车功能除能的情况下，也可通过软件设置 EVGR 寄存器的 BRKG 位来触发刹车事件。

通过设置 CHBRKCTR 寄存器的 BKE<sub>m</sub> 位可使能 MCTM 刹车输入信号。刹车输入极性可通过设置 CHBRKCTR 寄存器的 BKP<sub>m</sub> 位选择。BKE<sub>m</sub> 和 BKP<sub>m</sub> 位可同时修改。

数字滤波器嵌入在输入级和时钟控制器，用于刹车信号。通过设置 CHBRKCTR 寄存器的 BKF<sub>m</sub> 位可使能 MTn\_BRK<sub>m</sub> 信号的输入滤波器。数字滤波器是一个 N 事件计数器，N 指的是能够输出一个滤波信号所需的有效转换次数。

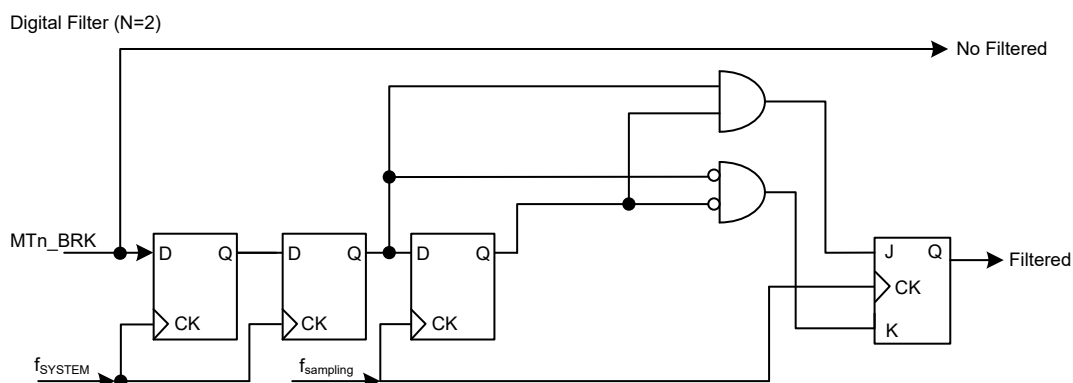


图 103. N = 2 的 MTn\_BRK 引脚数字滤波器方框图

当使用刹车功能时，通道输出使能信号和输出电平变化取决于 CHMOE、CHOSI、CHOSSR、CHxOIS 和 CHxOISN 控制位。一旦发生刹车事件，输出使能位 CHMOE 会被异步清零，刹车中断标志位 BRKmIF 会被置位，如果 BRKmIE 位被置 1 使能了刹车功能中断，那么还会产生一个中断。通道输出步骤如下所示：

- 如果使用互补输出，通道先会输出一个除能或者无效的电平信号，可通过配置 CHBRKCTR 寄存器的 CHOSI 位选择。一个死区时间周期后，输出转换为空闲状态。空闲状态由 CHBRKCFR 寄存器的 CHxOIS / CHxOISN 位决定。
  - 如果未使用互补输出（通道 3），通道将会输出一个空闲状态。
- 主要输出使能控制位 CHMOE 直到刹车事件被清除时才能设置。

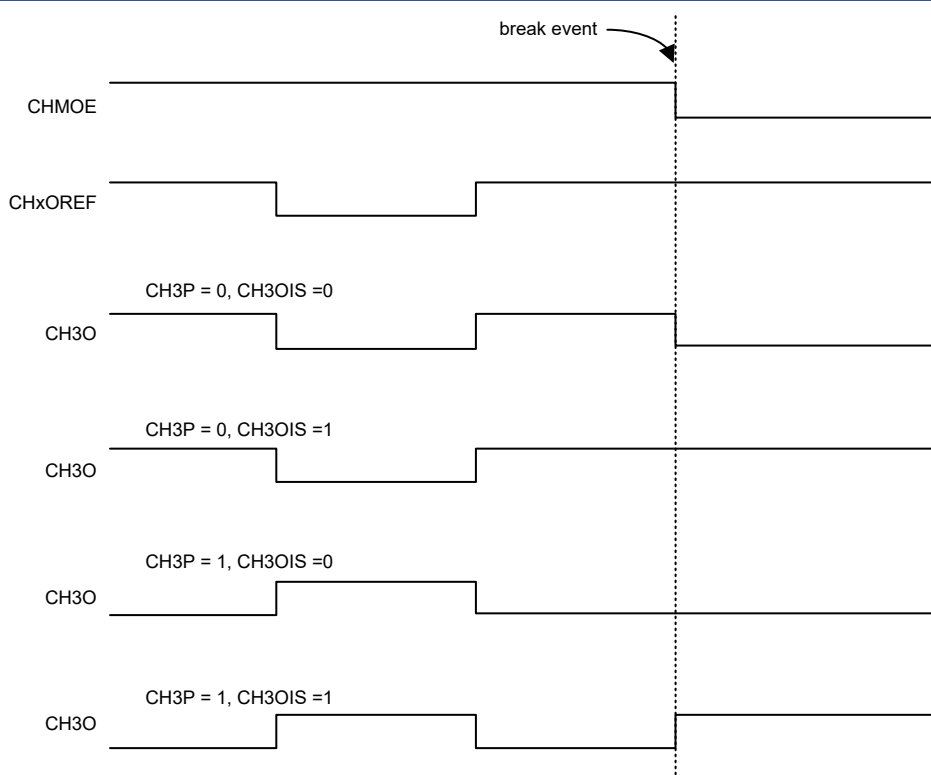


图 104. 刹车事件发生时的通道 3 输出

以下方框图显示了在通过设置 CHxE 和 CHxNE 都为 1 使能互补输出的情况下，刹车事件发生时的互补输出状态。

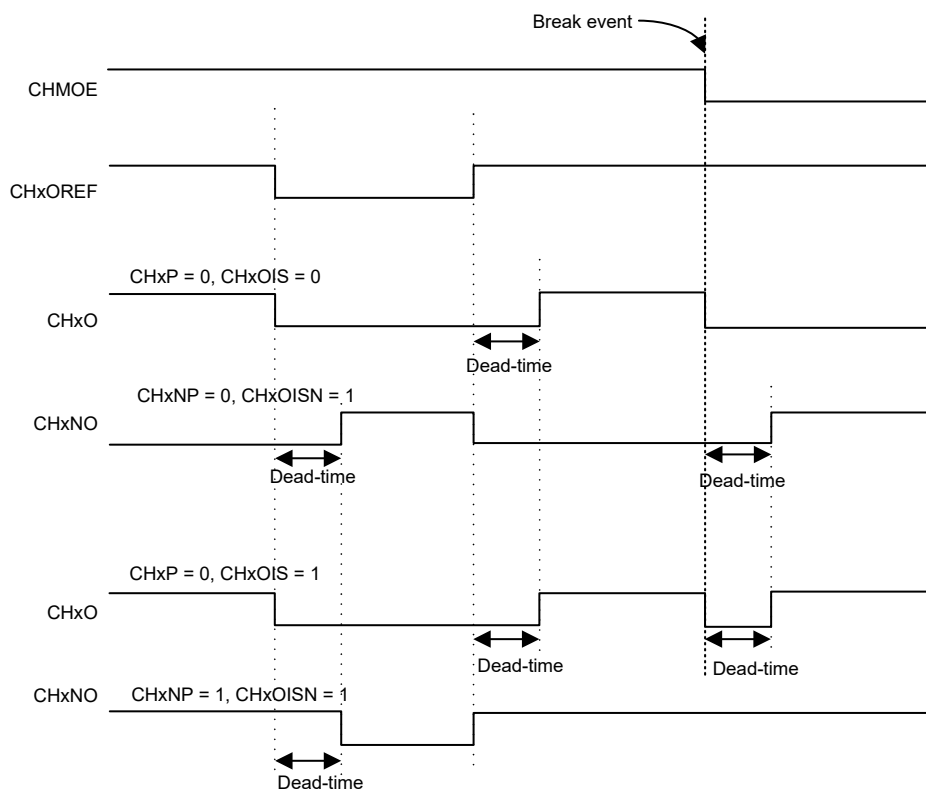


图 105. 刹车事件发生时的通道 0 ~ 2 互补输出

下图显示了在通过设置 CHxE 为 1 使能输出和清除 CHxNE 为 0 除能互补输出的情况下，刹车事件发生时的输出状态。

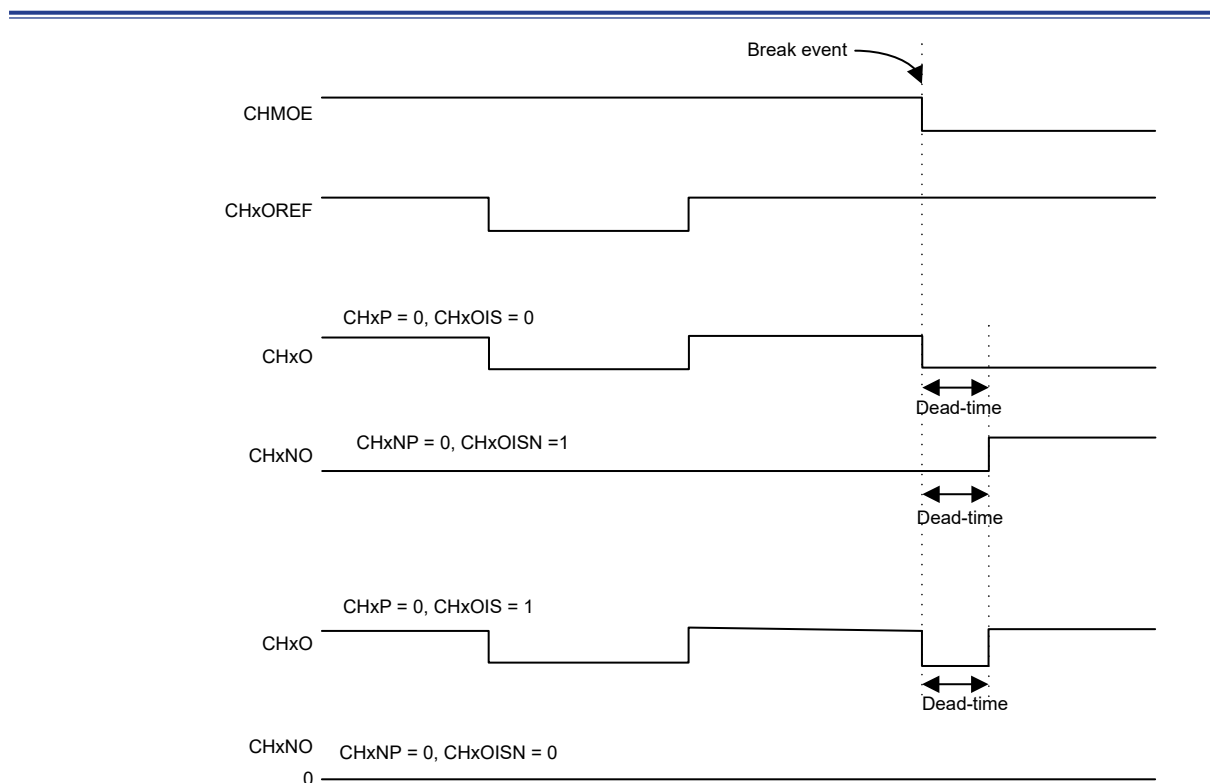


图 106. 刹车事件发生时通道 0 ~ 2 仅一个输出使能

CHxO 和 CHxNO 互补输出不能同时设置为有效电平。硬件将保护 MCTM 电路，仅一个通道被强制输出有效状态。

范例：刹车事件发生后，CHxOIS 和 CHxOISN 都可设置为有效电平，仅 CHxO 产生波形。

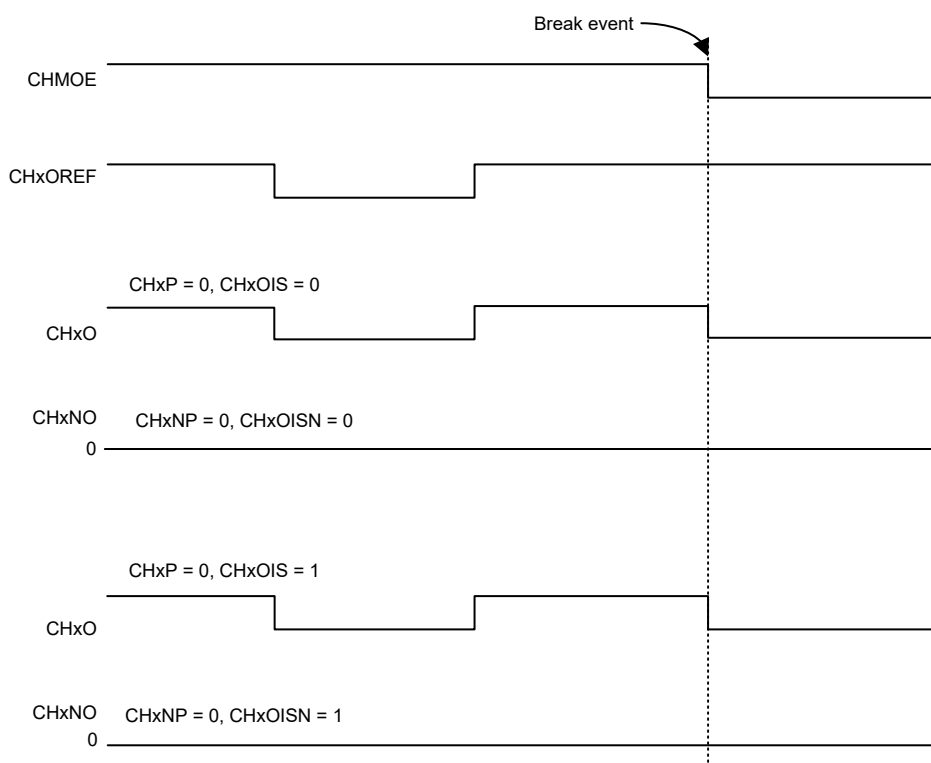


图 107. CHxO 和 CHxNO 都处于有效状态时的硬件保护

如果通过设置 CHBRKCTR 寄存器的 CHAOE 位为 1 使能自动输出使能功能，则可通过更新事件 1 自动设置 CHMOE 位。

### 带有刹车功能的通道互补输出

通道互补输出 CHxO 和 CHxNO 通过 CHxE、CHxNE、CHMOE、CHOSSR 和 CHOSSI 控制位的某种组合使能。

表 37. 刹车事件发生时的互补输出控制位

控制位					输出状态	
CHMOE	CHOSSI	CHOSSR	CHxE	CHxNE	MT_CHx 引脚输出状态	MT_CHxN 引脚输出状态
1 (运行)	x	0	0	0	输出除能 – 浮空 - 非定时器驱动 MT_CHx <sup>(1)</sup> = 浮空 MT_CHx_OEN <sup>(2)</sup> = 1	输出除能 – 浮空 - 非定时器驱动 MT_CHxN = 浮空 MT_CHxN_OEN = 1
		0	0	1	输出除能 – 浮空 - 非定时器驱动 MT_CHx_OEN = 1	输出使能 MT_CHxN = CHx_OREF xor CHxNP MT_CHxN_OEN = 0
		0	1	0	输出使能 MT_CHx = CHx_OREF xor CHxP MT_CHx_OEN = 0	输出除能 – 浮空 - 非定时器驱动 MT_CHxN = 浮空 MT_CHxN_OEN = 1
		0	1	1	输出使能 MT_CHx = CHx_OREF xor CHxP + 死区时间 MT_CHx_OEN = 0	输出使能 MT_CHxN = not CHx_OREF xor CHxNP + 死区时间 MT_CHxN_OEN = 0
		1	0	0	输出除能 – 浮空 - 非定时器驱动 MT_CHx = 浮空 MT_CHx_OEN = 1	输出除能 – 浮空 - 非定时器驱动 MT_CHxN = 浮空 MT_CHxN_OEN = 1
		1	0	1	关闭状态 MT_CHx = CHxP MT_CHx_OEN = 0	输出使能 MT_CHxN = CHx_OREF xor CHxNP MT_CHxN_OEN = 0
		1	1	0	输出使能 MT_CHx = CHx_OCREF xor CHxP MT_CHx_OEN = 0	关闭状态 MT_CHxN = CHxNP MT_CHxN_OEN = 0
		1	1	1	输出使能 MT_CHx = CHx_OREF xor CHxP + 死区时间 MT_CHx_OEN = 0	输出使能 MT_CHxN = not CHx_OREF xor CHxNP + 死区时间 MT_CHxN_OEN = 0
0 (空闲)	x	0	0	0	死区时间之前：输出除能 – 浮空 MT_CHx = 浮空, MT_CHxN = 浮空 MT_CHx_OEN = 1, MT_CHxN_OEN = 1	
		0	0	1		
		0	1	0		
		0	1	1		
		1	0	0	死区时间之前：关闭状态 MT_CHx = CHxP, MT_CHxN = CHxNP MT_CHx_OEN = 0, MT_CHxN_OEN = 0 死区时间之后：输出使能 MT_CHx = CHxOIS, MT_CHxN = CHxOISN MT_CHx_OEN = 0, MT_CHxN_OEN = 0	
		1	0	1		
		1	1	0		
		1	1	1		

注：1. MT\_CHx 引脚是 MCTM 的 I/O 引脚。

2. MT\_CHx\_OEN 和 MT\_CHxN\_OEN 信号是 MCTM I/O 引脚输出使能组合逻辑控制信号，为低电平有效。



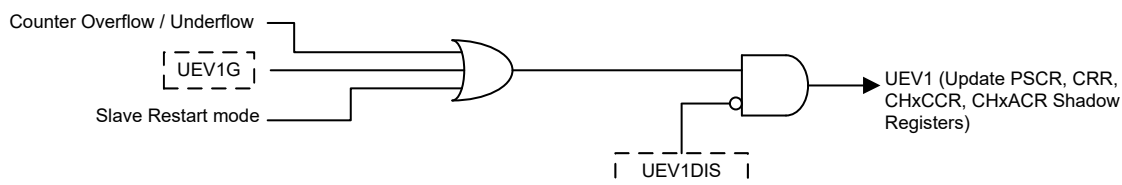
## 更新管理

更新事件划分为两种不同类型：更新事件 1 (UEV1) 和更新事件 2 (UEV2)。更新事件 1 用来把实际寄存器中 CRR、PSCR、CHxACR 和 CHxCCR 的值更新到相应的影子寄存器中。更新事件 1 在计数器上溢 / 下溢、UEV1G 位被置位或者从机重启模式被触发时发生。更新事件 2 用来更新控制位 CHxE、CHxNE 和 CHxOM。更新事件 2 在 STI 上升沿到来时或者相应软件更新控制位被置位时发生。

### 更新事件 1

CNTCFR 寄存器的 UEV1DIS 位控制更新事件 1 是否发生。当更新事件 1 发生，可能会产生相应的更新事件中断，但这取决于更新事件 1 中断功能是否已经通过设置 CNTCFR 寄存器中的 UGDIS 位开启。欲知更多详细信息，请参考 CNTCFR 寄存器中的 UEV1DIS 位和 UGDIS 位的相关定义。

#### Update Event 1 Management



#### Update Event 1 Interrupt Management

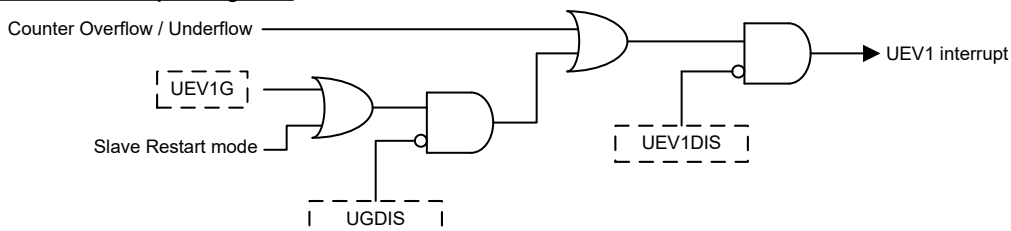


图 108. 更新事件 1 设置方框图

## 更新事件 2

通过设置 CTR 寄存器的 COMPRE 位,可预载用于互补输出的控制位 CHxE、CHxNE 和 CHxOM。当更新事件 2 发生时,CHxE、CHxNE 和 CHxOM 的影子位将会被更新。

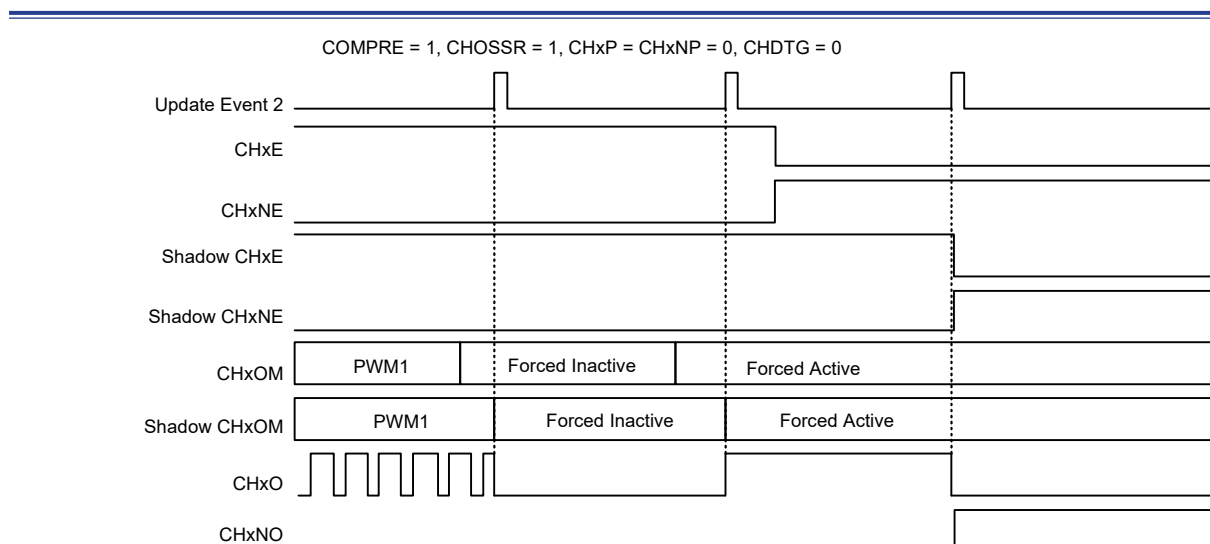


图 109. 更新事件 2 更新 CHxE、CHxNE 和 CHxOM

通过设置 EVGR 寄存器的软件更新位 UEV2G 或者在 CTR 寄存器的 COMUS 位被置位的情况下, STI 信号的上升沿产生时,都可发生更新事件 2。

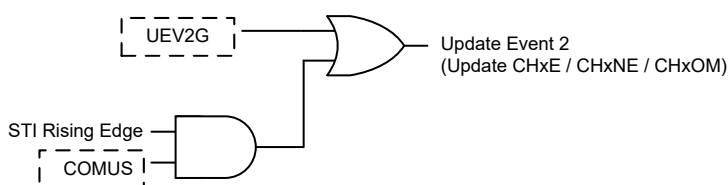


图 110. 更新事件 2 设置方框图

## 正交解码器

正交解码器功能使用两个来自 MTn\_CH0 和 MTn\_CH1 引脚的正交输入 TI0 和 TI1 进行交互以生成计数器的值。在每个输入源信号转换时, DIR 位会被硬件自动更改。输入源可以只来自于 TI0 或 TI1 边沿,也可以是来自于 TI0 和 TI1 边沿,通过设置 SMSEL 字段为 0x01、0x02 或 0x03 来选择。

下表列出了改变计数器方向的几种情况。正交解码器可视为一个带有方向选择性的外部时钟。这意味着，计数器将在 0 和计数器重载值之间连续计数。因此，用户必须在计数器开始计数之前设置 CRR 寄存器。

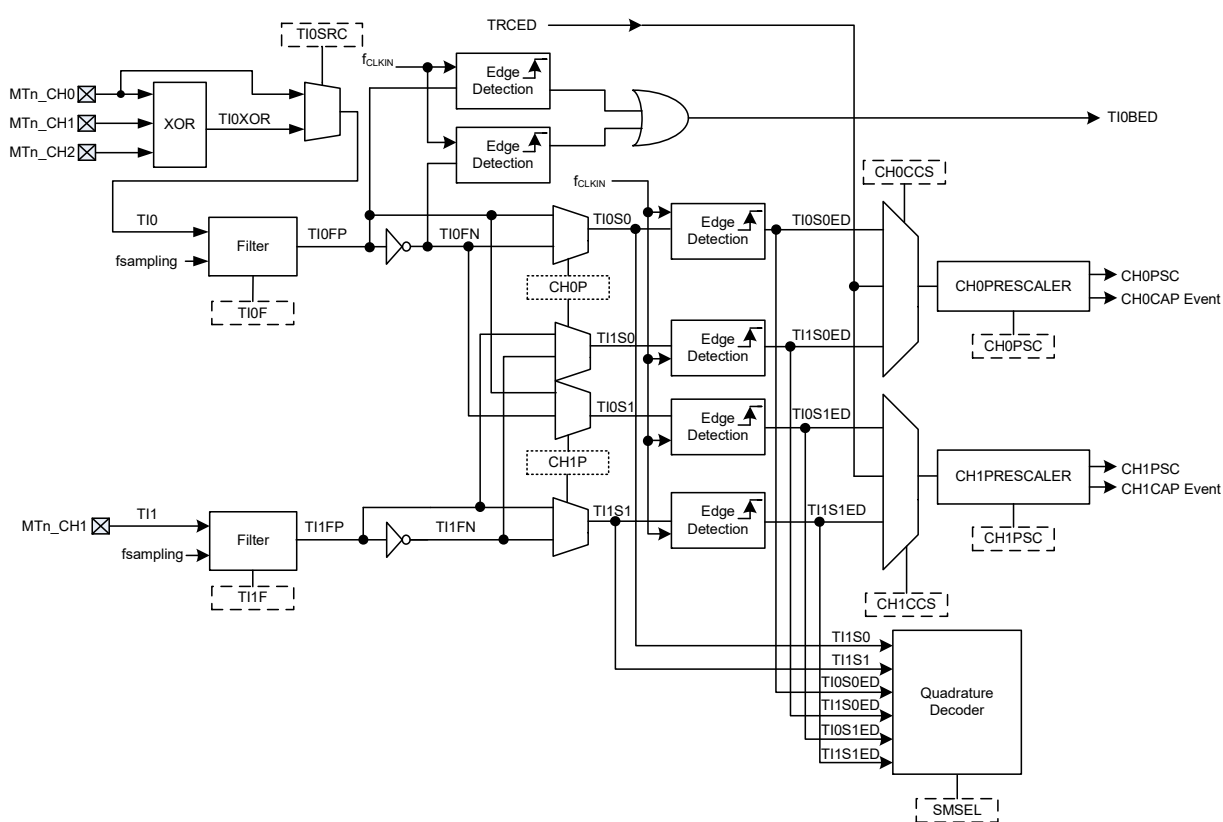


图 111. 输入级和正交解码器方框图

表 38. 计数方向和编码信号

计数模式	电平	TI0S0		TI1S1	
		上升沿	下降沿	上升沿	下降沿
仅 TI0 计数 (SMSEL = 0x01)	TI1S1 = 高	向下	向上	—	—
	TI1S1 = 低	向上	向下	—	—
仅 TI1 计数 (SMSEL = 0x02)	TI0S0 = 高	—	—	向上	向下
	TI0S0 = 低	—	—	向下	向上
TI0 和 TI1 计数 (SMSEL = 0x03)	TI1S1 = 高	向下	向上	X	X
	TI1S1 = 低	向上	向下	X	X
	TI0S0 = 高	X	X	向上	向下
	TI0S0 = 低	X	X	向下	向上

“—”：没有计数；“X”：不可能

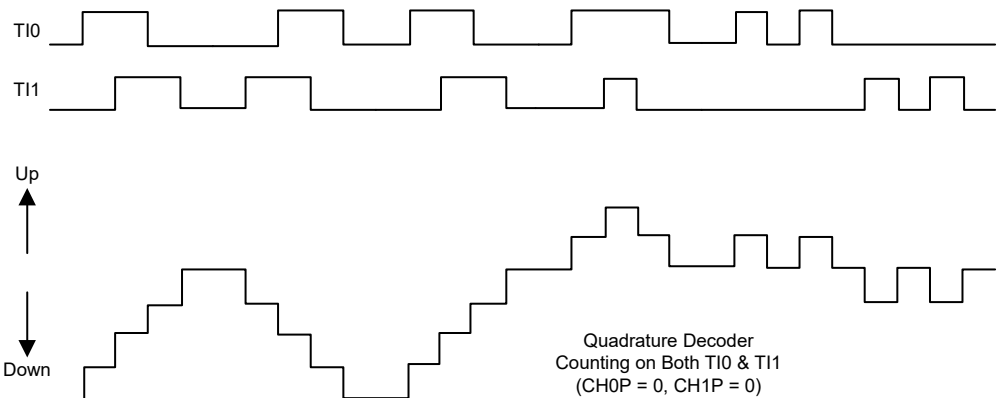


图 112. TI0 和 TI1 正交解码器计数

数字滤波器

数字滤波器嵌入在输入级和时钟控制器,分别用于 MTn\_CH0 ~ MTn\_CH3 和 MTn\_ETI 引脚。MCTM 中的数字滤波器是一个 N 事件计数器, N 指的是能够输出一个滤波信号所需的有效转换次数。根据每个滤波器的选择, N 的值可以是 0、2、4、5、6 或 8。

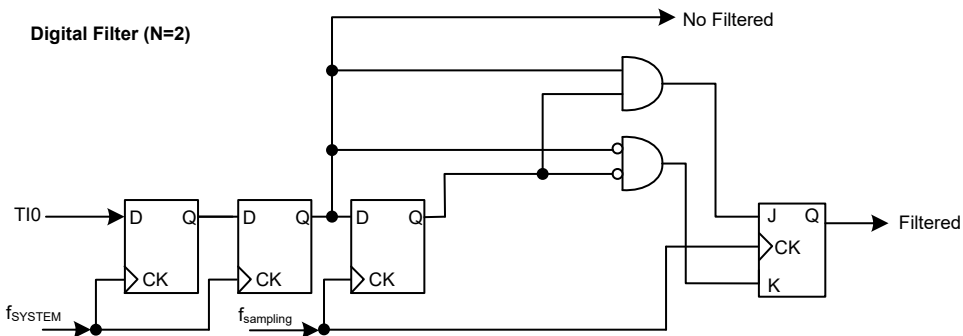


图 113. N = 2 的 MTn\_ETI 引脚数字滤波器方框图

## ETIF 为高电平时清除 CHxOREF

当把 CHxOCFR 寄存器中的 REFxCE 位置 1, 使 ETIF 信号为高电平时, CHxOREF 信号会被强制置为 0。直到下一个更新事件 1 发生时, CHxOREF 信号才会回到其有效电平。

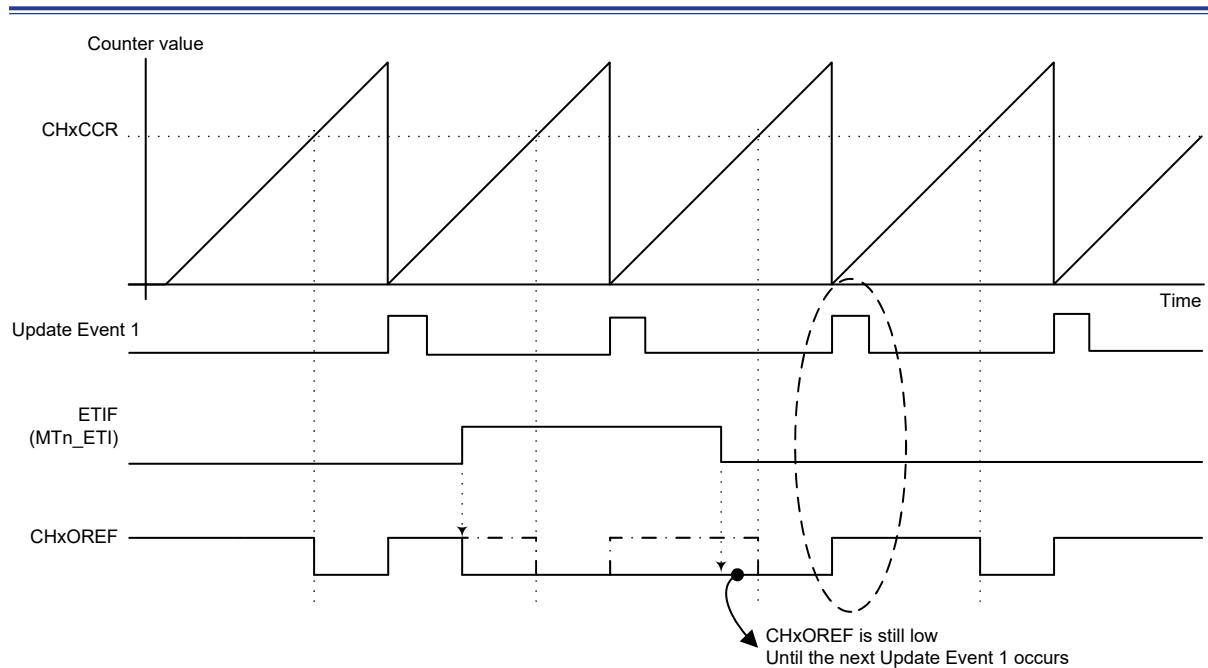


图 114. 通过 ETIF 清除 CHxOREF

## 单脉冲模式

一旦定时器被设置工作在单脉冲模式下，则无需把 CTR 寄存器中的定时器使能位 TME 置为 1 来使能定时器。当 STI 信号上升沿发生时或通过软件把 TME 位设置为 1 时，触发器将会产生一个脉冲。然后一直保持 TME 位为高电平直到更新事件 1 发生或由软件将 TME 位清零。如果使用软件使 TME 位清零，计数器将停止且保持当前值不变。如果 TME 位是由硬件的更新事件 1 自动清零，计数器将被重新初始化。

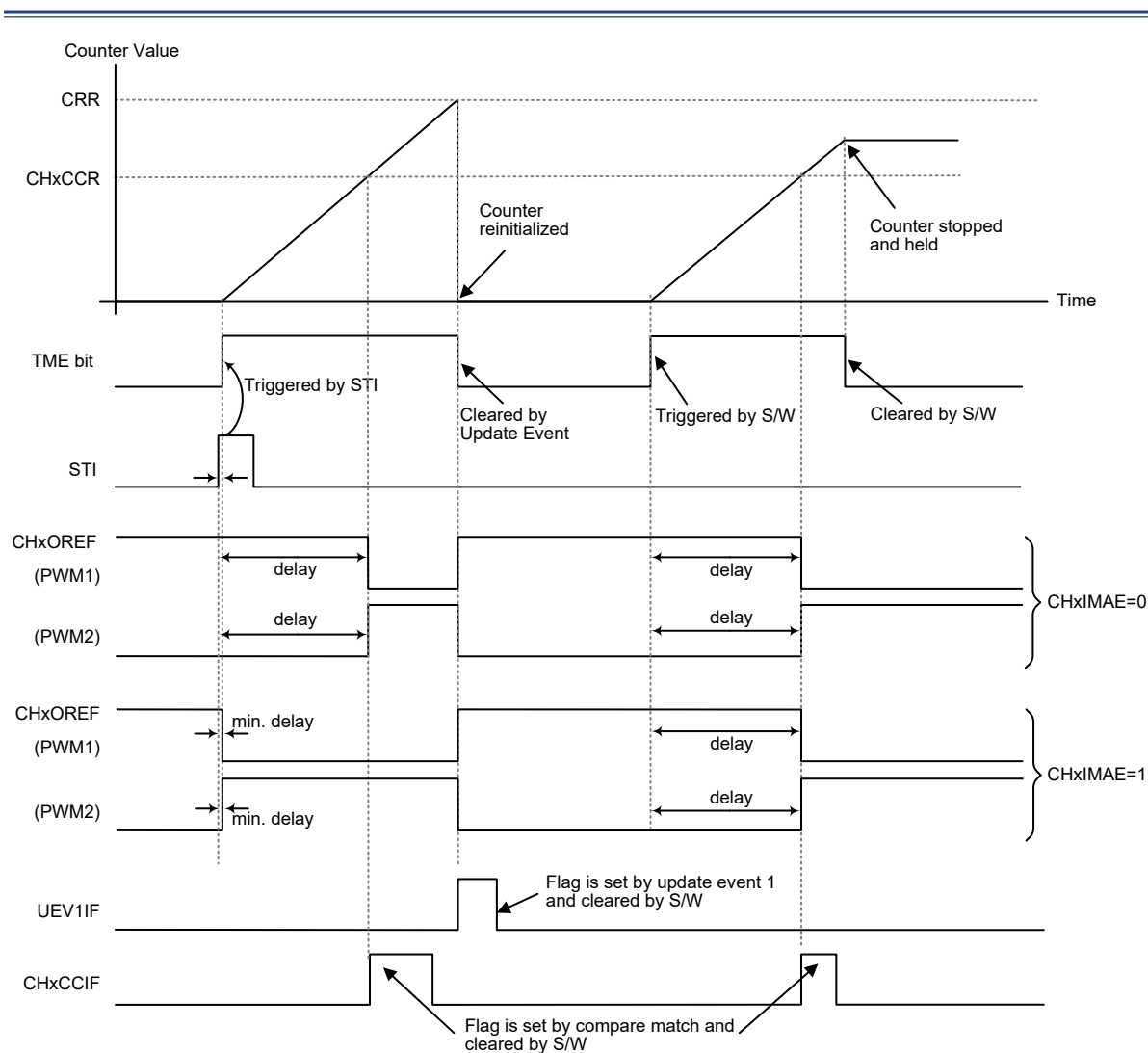


图 115. 单脉冲模式

在单脉冲模式下, STI 有效边沿使 TME 置为 1 时, 将使能计数器。然而, 由于要执行计数器值和 CHxCCR 值的比较结果, 会存在几个时钟延迟。用户可以通过设置 CHxOCFR 寄存器中的 CHxIMAE 位来使延迟时间减小。单脉冲模式下, STI 上升沿触发发生后, CHxOREF 信号将立即被强制进入以下状态, 即 CHxOREF 信号将转变为比较匹配事件发生的电平, 而不考虑比较结果是如何的。只有当输出通道被配置工作在 PWM1 或 PWM2 输出模式下且触发源来自于 STI 信号时, CHxIMAE 位可用。

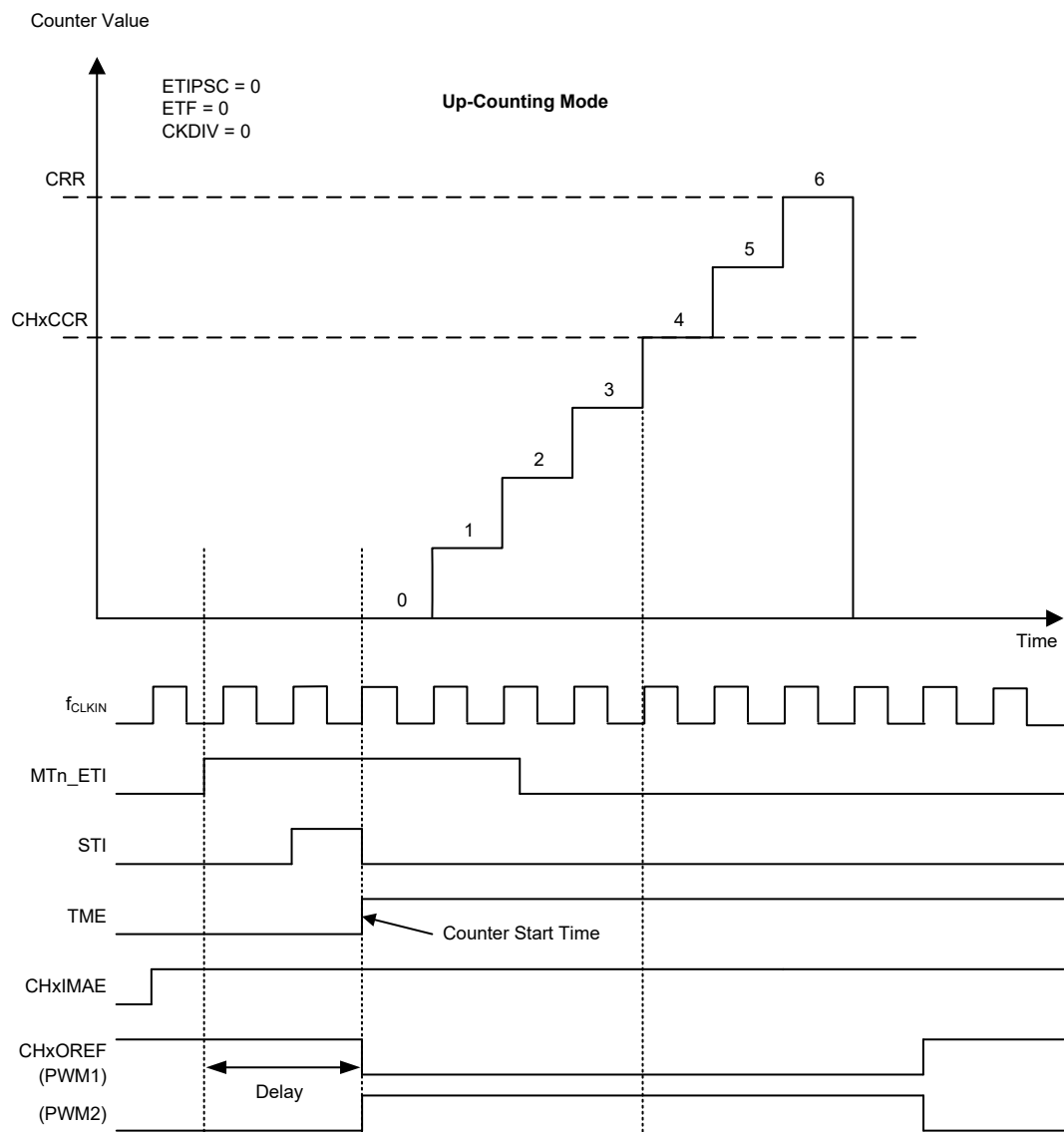


图 116. 立即有效模式的延迟

## 非对称 PWM 模式

非对称 PWM 模式允许两个中心对齐的 PWM 信号产生一个可编程的相位偏移。PWM 频率由 CRR 寄存器的值决定，占空比和相移由 CHxCCR 和 CHxACR 寄存器决定。当计数器向上计数时，PWM 使用 CHxCCR 的值作为向上计数的比较值。当计数器进入向下计数阶段，PWM 使用 CHxACR 的值作为向下计数的比较值。图 117 是在中心对齐计数模式下的非对称 PWM 模式的一个例子。

注：非对称 PWM 模式只能在中心对齐计数模式下运行。

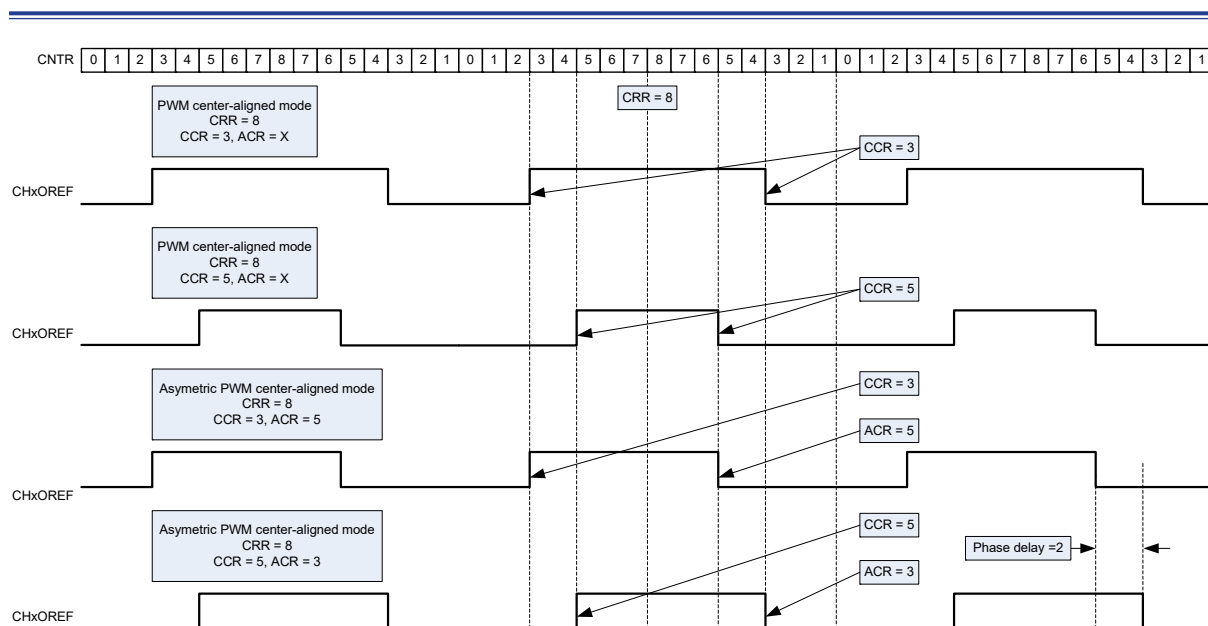


图 117. 中心对齐计数模式下的非对称 PWM 模式



定时器互连

定时器可以在内部互相连接用于使定时器链接和同步。它可通过配置一个定时器工作在主机模式，而配置另一个定时器工作在从机模式来实现。下面是主机 / 从机模式触发器选择的几个示例。

使用一个定时器触发另外一个定时器开始或停止计数

- 使 MCTM0 工作在主机模式，设置通道 0 输出参考信号 CH0OREF 作为触发器输出 (MMSEL = 0x04)
- 配置 MCTM0 CH0OREF 波形
- 使 GPTM0 接收来自于 MCTM0 触发器输出的输入触发器源 (TRSEL = 0x0A)
- 使 GPTM0 工作在暂停模式 (SMSEL = 0x05)
- 通过向 TME 位写入 1 来使能 GPTM0
- 通过向 TME 位写入 1 来使能 MCTM0

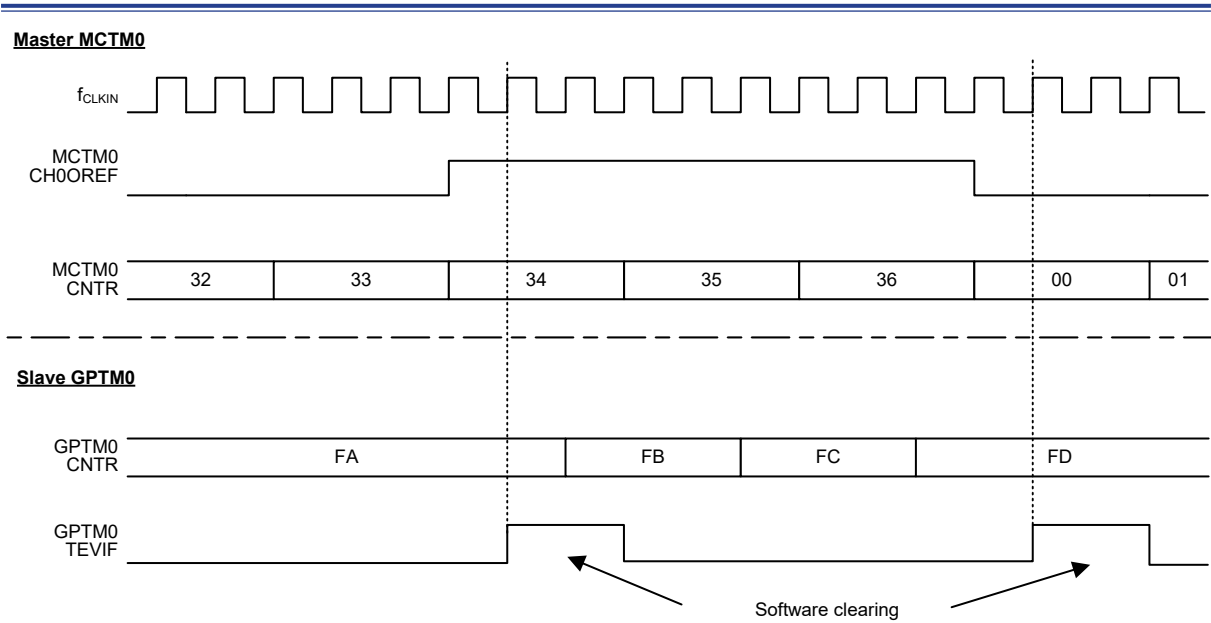


图 118. 用 MCTM0 CH0OREF 信号暂停 GPTM0

#### 使用一个定时器触发另一个定时器开始计数

- 使 MCTM0 工作在主机模式，发送更新事件 UEV 作为触发输出 (MMSEL = 0x02)
- 通过设置 CRR 寄存器配置 MCTM0 周期
- 使 GPTM0 从 MCTM0 触发输出获得输入触发源 (TRSEL = 0x0A)
- 使 GPTM0 工作在从机触发模式 (SMSEL = 0x06)
- 向 TME 位写入 1 开启 MCTM0

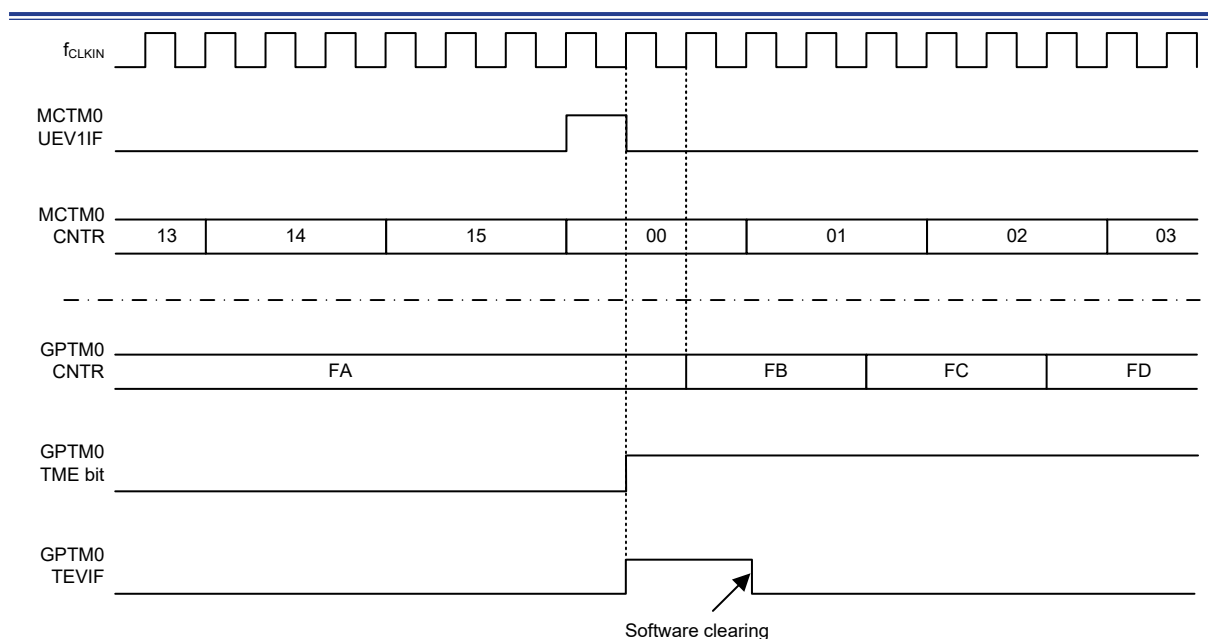


图 119. 用 MCTM0 更新事件 1 触发 GPTM0

#### 启动两个定时器同步响应外部触发

- 使 MCTM0 工作在主机模式下，发送使能信号作为触发输出 (MMSEL = 0x01)
- 使 MCTM0 工作在从机模式下，从 MTn\_CH0 引脚获得其输入触发源 (TRSEL = 0x01)
- 使 MCTM0 工作在从机触发模式 (SMSEL = 0x06)
- 设置 MDCFR 寄存器中的 TSE 位为 1，使能 MCTM0 主机定时器同步功能使其与从机定时器同步
- 使 GPTM0 接收来自于 MCTM0 的触发输出作为其输入触发源 (TRSEL = 0x0A)
- 使 GPTM0 工作在从机触发模式下 (SMSEL = 0x06)

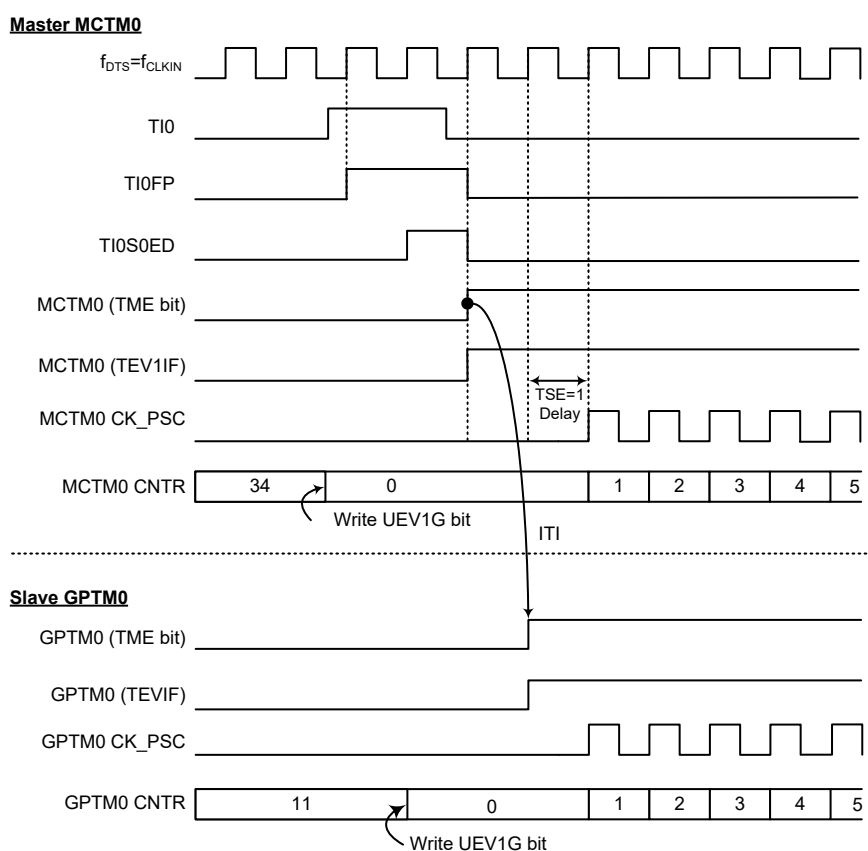


图 120. 用 MCTM0 CH0 输入信号触发 MCTM0 和 GPTM0

使用一个定时器作为霍尔传感器接口来触发另一个带有更新事件 2 的定时器

#### GPTM0:

- 配置通道 0 选择输入 XOR 功能 (TI0SRC = 1)
- 设置通道 0 为输入捕捉模式, TRCED 作为捕捉源 (CH0CCS = 0x03) 并使能通道 0 (CH0E = 1)
- 配置 UEV1G 位作为 MTO 的时钟源 (MMSEL = 0x00)
- 把 TI0BED 连接到 STI (TRSEL = 0x08)
- 使计数器工作在从机重启模式 (SMSEL = 0x04)
- 使能 GPTM0 (TME = 1)

#### MCTM0:

- 选择 GPTM0 MTO 作为 MCTM 的 STI 源 (TRSEL = 0x0A)
- 使能 CHxE、CHxNE 和 CHxOM 预载功能 (COMPRES = 1)
- 选择 STI 上升沿来产生更新事件 2 (COMUS = 1)
- 使能更新事件 2 中断 (UEV2IE = 1)
- 在更新事件 2 中断服务子程序中: 为下一步对 CHxE、CHxNE 和 CHxOM 寄存器写值

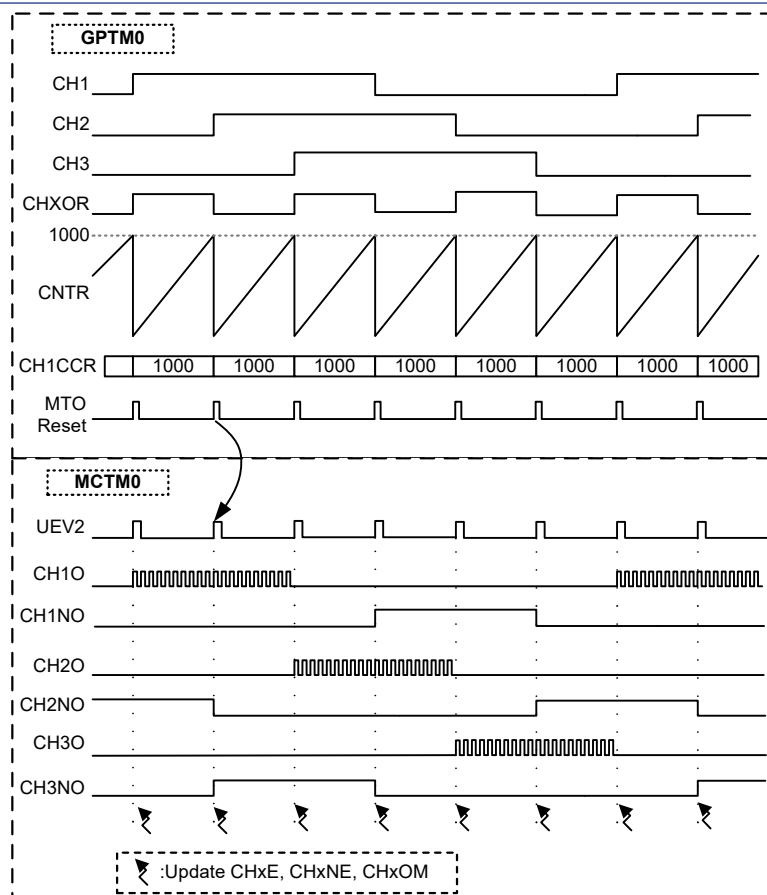


图 121. CH1XOR 输入作为霍尔传感器接口

## 触发 ADC 开启

为了与模拟 / 数字转换器相连接, MCTM 可以输出 MTO 信号或通道比较匹配输出信号 CHxOREF (x = 0 ~ 3) 作为模拟 / 数字转换器的输入触发信号。

## 锁定级别表

除了刹车输入和输出管理, 刹车电路还可内部完成写保护以保护应用。用户可通过 LOCKLV 位选择一种保护级别以保护寄存器相关控制位。MCTM 复位或系统复位发生后, LOCKLV 位只能被写入一次, 然后被保护的位将被锁定且不会再改变, 除非 MCTM 复位或系统复位。

表 39. 锁定级别表

锁定配置	被保护位					
锁定级别 1 (LOCKLV = "01")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
锁定级别 2 (LOCKLV = "10")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
	CHxP	CHxNP	CHOSI	CHOSSR	MCTMEN <sup>(1)</sup>	CKMEN <sup>(2)</sup>
锁定级别 3 (LOCKLV = "11")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
	CHxP	CHxNP	CHOSI	CHOSSR	MCTMEN <sup>(1)</sup>	CKMEN <sup>(2)</sup>
	CHxPRE	CHxOM	—	—	—	—

注: 1. APBCCR1 寄存器中的 MCTMEN 位位于 CKCU 单元, 用于控制 MCTM 单元的时钟源。

2. GCCR 寄存器中的 CKMEN 位位于 CKCU 单元, 用于监控高速外部时钟源 (HSE)。如果 CKMEN 位使能, 当硬件检测到 HSE 时钟停留在低电平 / 高电平状态, 内部硬件会自动切换系统时钟到内部高速 RC 时钟 (HSI) 以保护系统安全。

3. 在 MCTM 单元, CKCU 的 MCTMEN 和 CKMEN 控制位锁定保护模式使能时, 这些位将只被允许使能, 而不能再次除能

## PDMA 请求

MCTM 带有一个 PDMA 数据传输接口。如果相关使能控制位置 1 使能 PDMA 访问, 那么有一些事件可产生 PDMA 请求。这些事件可以是 MCTM 更新事件、触发事件和通道捕捉 / 比较匹配事件。当 MCTM 通道产生 PDMA 请求时, 对于所有通道都可通过通道 PDMA 选择位 CHCCDS 选择是来自通道捕捉 / 比较匹配事件或者 MCTM 更新事件 1。欲知 PDMA 配置信息, 请参考 PDMA 的相关章节。

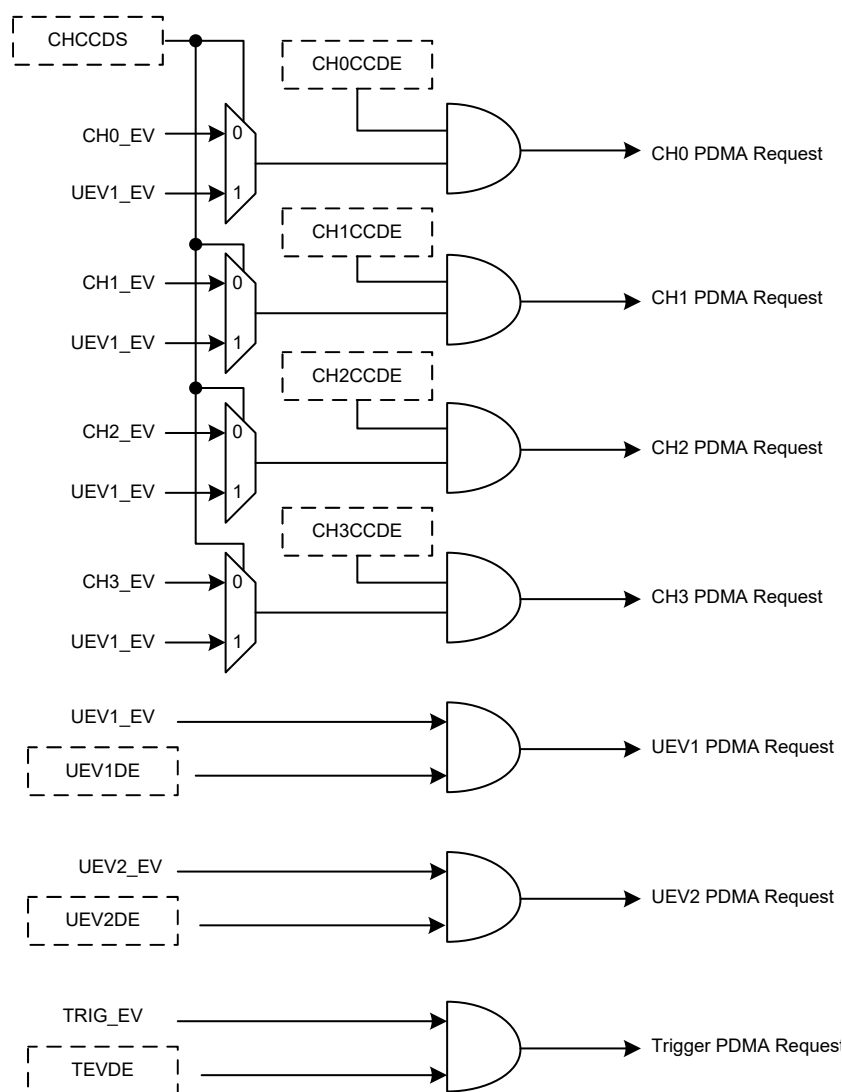


图 122. MCTM PDMA 映射图

## 寄存器列表

下表所示为 MCTM 寄存器及其复位值。

表 40. MCTM 寄存器列表

寄存器	偏移量	描述	复位值
CNTCFR	0x000	定时器计数器配置寄存器	0x0000_0000
MDCFR	0x004	定时器模式配置寄存器	0x0000_0000
TRCFR	0x008	定时器触发配置寄存器	0x0000_0000
CTR	0x010	定时器控制寄存器	0x0000_0000
CH0ICFR	0x020	通道 0 输入配置寄存器	0x0000_0000
CH1ICFR	0x024	通道 1 输入配置寄存器	0x0000_0000
CH2ICFR	0x028	通道 2 输入配置寄存器	0x0000_0000
CH3ICFR	0x02C	通道 3 输入配置寄存器	0x0000_0000
CH0OCFR	0x040	通道 0 输出配置寄存器	0x0000_0000
CH1OCFR	0x044	通道 1 输出配置寄存器	0x0000_0000
CH2OCFR	0x048	通道 2 输出配置寄存器	0x0000_0000
CH3OCFR	0x04C	通道 3 输出配置寄存器	0x0000_0000
CHCTR	0x050	通道控制寄存器	0x0000_0000
CHPOLR	0x054	通道极性配置寄存器	0x0000_0000
CHBRKCFR	0x06C	通道刹车配置寄存器	0x0000_0000
CHBRKCTR	0x070	通道刹车控制寄存器	0x0000_0000
DICTR	0x074	定时器 PDMA / 中断控制寄存器	0x0000_0000
EVGR	0x078	定时器事件发生器寄存器	0x0000_0000
INTSR	0x07C	定时器中断状态寄存器	0x0000_0000
CNTR	0x080	定时器计数器寄存器	0x0000_0000
PSCR	0x084	定时器预分频器寄存器	0x0000_0000
CRR	0x088	定时器计数器重载寄存器	0x0000_FFFF
REPR	0x08C	定时器重复寄存器	0x0000_0000
CH0CCR	0x090	通道 0 捕捉 / 比较寄存器	0x0000_0000
CH1CCR	0x094	通道 1 捕捉 / 比较寄存器	0x0000_0000
CH2CCR	0x098	通道 2 捕捉 / 比较寄存器	0x0000_0000
CH3CCR	0x09C	通道 3 捕捉 / 比较寄存器	0x0000_0000
CH0ACR	0x0A0	通道 0 非对称比较寄存器	0x0000_0000
CH1ACR	0x0A4	通道 1 非对称比较寄存器	0x0000_0000
CH2ACR	0x0A8	通道 2 非对称比较寄存器	0x0000_0000
CH3ACR	0x0AC	通道 3 非对称比较寄存器	0x0000_0000

## 寄存器描述

### 定时器计数器配置寄存器 – CNTCFR

该寄存器定义了 MCTM 计数器配置。

偏移量: 0x000

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							DIR
类型 / 复位								RW 0
	23	22	21	20	19	18	17	16
	保留位						CMSEL	
类型 / 复位							RW 0	RW 0
	15	14	13	12	11	10	9	8
	保留位						CKDIV	
类型 / 复位							RW 0	RW 0
	7	6	5	4	3	2	1	0
	保留位						UGDIS	UEV1DIS
类型 / 复位							RW 0	RW 0

位	字段	描述
[24]	DIR	计数方向 0: 向上计数 1: 向下计数 注: 当定时器工作在中心对齐计数模式下或作为正交解码器使用时, 此位为只读位。
[17:16]	CMSEL	计数器模式选择 00: 边沿对齐计数模式。此模式下可正常向上和向下计数。计数方向由 DIR 位定义。 01: 中心对齐计数模式 1。计数器向上向下交替计数。比较匹配中断标志位在向下计数期间被置位。 10: 中心对齐计数模式 2。计数器向上向下交替计数。比较匹配中断标志位在向上计数期间被置位。 11: 中心对齐计数模式 3。计数器向上向下交替计数。比较匹配中断标志位在向上和向下计数期间被置位。
[9:8]	CKDIV	时钟分频 这两位定义了定时器时钟 ( $f_{CLKIN}$ ) 和死区时钟 ( $f_{DTS}$ ) 之间的频率比。死区时钟也可作为数字滤波器的采样时钟使用。 00: $f_{DTS} = f_{CLKIN}$ 01: $f_{DTS} = f_{CLKIN} / 2$ 10: $f_{DTS} = f_{CLKIN} / 4$ 11: 保留位
[1]	UGDIS	更新事件 1 中断产生除能控制位 0: 以下任何一个事件都可产生一个更新 PDMA 请求或中断 – 计数器上溢 / 下溢 – 设置 UEV1G 位 – 通过从机模式产生更新 1: 只在计数器上溢 / 下溢时产生一个更新 PDMA 请求或中断



位	字段	描述
[0]	UEV1DIS	更新事件 1 除能控制位 0: 以下任何一个事件都可使能更新事件 1 请求 – 计数器上溢 / 下溢 – 设置 UEV1G 位 – 通过从机模式产生更新 1: 除能更新事件 1 – 如果 UEV1G 位被置位或从从机模式收到硬件重启, 那么计数器和预分频器将被重新初始化

定时器模式配置寄存器 – MDCFR

该寄存器定义了 MCTM 主机和从机模式选项以及单脉冲模式。

偏移量: 0x004  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位							SPMSET	
类型 / 复位								RW	0
	23	22	21	20	19	18	17	16	
	保留位					MMSEL			
类型 / 复位						RW	0	RW	0
	15	14	13	12	11	10	9	8	
	保留位					SMSEL			
类型 / 复位						RW	0	RW	0
	7	6	5	4	3	2	1	0	
	保留位							TSE	
类型 / 复位								RW	0

位	字段	描述
[24]	SPMSET	单脉冲模式设置 0: 无论更新事件是否发生, 计数器正常计数。 1: 下一个更新事件到来时, 计数器停止计数, 接着 TME 位被硬件清零。

位	字段	描述																											
[18:16]	MMSEL	<p>主机模式选项 主机模式选项用来选择与其它从机定时器同步的 MTO 信号源。</p> <table> <tr> <th>MMSEL[2:0]</th><th>模式</th><th>描述</th></tr> <tr> <td>000</td><td>复位模式</td><td>复位模式下的 MTO 输出信号由以下条件之一产生： 1. 软件设置 UEV1G 位 2. 从机重启模式下从机触发输入</td></tr> <tr> <td>001</td><td>使能模式</td><td>计数器使能信号作为触发输出。</td></tr> <tr> <td>010</td><td>更新模式</td><td>当 UEVIDIS 位被清零时，更新事件 1 用作触发输出且取决于下列条件之一： 1. 计数器上溢 / 下溢 2. 软件设置 UEV1G 位 3. 从机重启模式下从机触发输入</td></tr> <tr> <td>011</td><td>捕捉 / 比较模式</td><td>当通道 0 捕捉或比较匹配事件发生，将产生一个正脉冲作为主机触发输出。</td></tr> <tr> <td>100</td><td>比较输出 0</td><td>通道 0 输出参考信号 CH0OREF 作为触发输出。</td></tr> <tr> <td>101</td><td>比较输出 1</td><td>通道 1 输出参考信号 CH1OREF 作为触发输出。</td></tr> <tr> <td>110</td><td>比较输出 2</td><td>通道 2 输出参考信号 CH2OREF 作为触发输出。</td></tr> <tr> <td>111</td><td>比较输出 3</td><td>通道 3 输出参考信号 CH3OREF 作为触发输出。</td></tr> </table>	MMSEL[2:0]	模式	描述	000	复位模式	复位模式下的 MTO 输出信号由以下条件之一产生： 1. 软件设置 UEV1G 位 2. 从机重启模式下从机触发输入	001	使能模式	计数器使能信号作为触发输出。	010	更新模式	当 UEVIDIS 位被清零时，更新事件 1 用作触发输出且取决于下列条件之一： 1. 计数器上溢 / 下溢 2. 软件设置 UEV1G 位 3. 从机重启模式下从机触发输入	011	捕捉 / 比较模式	当通道 0 捕捉或比较匹配事件发生，将产生一个正脉冲作为主机触发输出。	100	比较输出 0	通道 0 输出参考信号 CH0OREF 作为触发输出。	101	比较输出 1	通道 1 输出参考信号 CH1OREF 作为触发输出。	110	比较输出 2	通道 2 输出参考信号 CH2OREF 作为触发输出。	111	比较输出 3	通道 3 输出参考信号 CH3OREF 作为触发输出。
MMSEL[2:0]	模式	描述																											
000	复位模式	复位模式下的 MTO 输出信号由以下条件之一产生： 1. 软件设置 UEV1G 位 2. 从机重启模式下从机触发输入																											
001	使能模式	计数器使能信号作为触发输出。																											
010	更新模式	当 UEVIDIS 位被清零时，更新事件 1 用作触发输出且取决于下列条件之一： 1. 计数器上溢 / 下溢 2. 软件设置 UEV1G 位 3. 从机重启模式下从机触发输入																											
011	捕捉 / 比较模式	当通道 0 捕捉或比较匹配事件发生，将产生一个正脉冲作为主机触发输出。																											
100	比较输出 0	通道 0 输出参考信号 CH0OREF 作为触发输出。																											
101	比较输出 1	通道 1 输出参考信号 CH1OREF 作为触发输出。																											
110	比较输出 2	通道 2 输出参考信号 CH2OREF 作为触发输出。																											
111	比较输出 3	通道 3 输出参考信号 CH3OREF 作为触发输出。																											
[10:8]	SMSEL	<p>从机模式选项</p> <table> <tr> <th>SMSEL[2:0]</th><th>模式</th><th>描述</th></tr> <tr> <td>000</td><td>除能模式</td><td>预分频器直接用作内部时钟计时。</td></tr> <tr> <td>001</td><td>正交解码模式 1</td><td>计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI0 边沿转换取决于 TI1 的电平。</td></tr> <tr> <td>010</td><td>正交解码模式 2</td><td>计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI1 边沿转换取决于 TI0 的电平。</td></tr> <tr> <td>011</td><td>正交解码模式 3</td><td>计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。在此模式下的一通道边沿转换取决于另外一个通道的电平。</td></tr> <tr> <td>100</td><td>重启模式</td><td>计数器在 STI 信号上升沿，依计数模式从 0 或 CRR 影子寄存器的值重新开始计数。寄存器也将被更新。</td></tr> <tr> <td>101</td><td>暂停模式</td><td>当所选择的触发输入信号 STI 是高电平时，计数器开始计数。当 STI 信号转换成低电平时，计数器立即停止计数，而不会产生复位。计数器开始和停止计数是由 STI 信号控制的。</td></tr> <tr> <td>110</td><td>触发模式</td><td>在所选择的 STI 触发信号上升沿处，计数器从初始值开始计数。只有计数器的开启是由 STI 信号控制的。</td></tr> <tr> <td>111</td><td>STIED</td><td>计数器使用所选的触发信号 STI 的上升沿计时。</td></tr> </table>	SMSEL[2:0]	模式	描述	000	除能模式	预分频器直接用作内部时钟计时。	001	正交解码模式 1	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI0 边沿转换取决于 TI1 的电平。	010	正交解码模式 2	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI1 边沿转换取决于 TI0 的电平。	011	正交解码模式 3	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。在此模式下的一通道边沿转换取决于另外一个通道的电平。	100	重启模式	计数器在 STI 信号上升沿，依计数模式从 0 或 CRR 影子寄存器的值重新开始计数。寄存器也将被更新。	101	暂停模式	当所选择的触发输入信号 STI 是高电平时，计数器开始计数。当 STI 信号转换成低电平时，计数器立即停止计数，而不会产生复位。计数器开始和停止计数是由 STI 信号控制的。	110	触发模式	在所选择的 STI 触发信号上升沿处，计数器从初始值开始计数。只有计数器的开启是由 STI 信号控制的。	111	STIED	计数器使用所选的触发信号 STI 的上升沿计时。
SMSEL[2:0]	模式	描述																											
000	除能模式	预分频器直接用作内部时钟计时。																											
001	正交解码模式 1	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI0 边沿转换取决于 TI1 的电平。																											
010	正交解码模式 2	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。此模式下的 TI1 边沿转换取决于 TI0 的电平。																											
011	正交解码模式 3	计数器使用由 TI0 和 TI1 信号交互产生的时钟脉冲来驱动计数器预分频器。在此模式下的一通道边沿转换取决于另外一个通道的电平。																											
100	重启模式	计数器在 STI 信号上升沿，依计数模式从 0 或 CRR 影子寄存器的值重新开始计数。寄存器也将被更新。																											
101	暂停模式	当所选择的触发输入信号 STI 是高电平时，计数器开始计数。当 STI 信号转换成低电平时，计数器立即停止计数，而不会产生复位。计数器开始和停止计数是由 STI 信号控制的。																											
110	触发模式	在所选择的 STI 触发信号上升沿处，计数器从初始值开始计数。只有计数器的开启是由 STI 信号控制的。																											
111	STIED	计数器使用所选的触发信号 STI 的上升沿计时。																											
[0]	TSE	<p>定时器同步使能</p> <p>0: 无动作</p> <p>1: 主机定时器 ( 当前定时器 ) 将会产生一个延时以通过 MTO 信号同步其从机定时器</p>																											

定时器触发配置寄存器 – TRCFR

该寄存器定义了 MCTM 的外部时钟设置和触发源选项。

偏移量： 0x008  
复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							ECME
								RW 0
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							ETIPOL
								RW 0
类型 / 复位	15	14	13	12	11	10	9	8
	保留位		ETIPSC			ETF		
			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位				TRSEL			
					RW 0	RW 0	RW 0	RW 0

位	字段	描述
[24]	ECME	外部时钟模式使能 0: 外部时钟模式除能 1: 外部时钟模式使能 以下两种设置效果是一样的： 1. ECME 位置 1 2. 设置 SMSEL = 0x111，连接 STI 到 ETIF (TRSEL = 0x011)
[16]	ETIPOL	外部触发极性 0: MTn_ETI 作用在高电平或上升沿 1: MTn_ETI 作用在低电平或下降沿
[13:12]	ETIPSC	外部触发预分频器 分频器使能来减小 ETIP 频率 00: 分频器 OFF 01: ETIP 频率 / 2 10: ETIP 频率 / 4 11: ETIP 频率 / 8

位	字段	描述
[11:8]	ETF	<p>外部触发滤波器</p> <p>这些位定义了分频比用来采样 MTn_ETI 信号。MPTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出滤波信号所需的有效转换次数。</p> <p>0000: 无滤波，采样时钟是 <math>f_{DTS}</math></p> <p>0001: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 2</math></p> <p>0010: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 4</math></p> <p>0011: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 8</math></p> <p>0100: <math>f_{\text{sampling}} = f_{DTS} / 2</math>, <math>N = 6</math></p> <p>0101: <math>f_{\text{sampling}} = f_{DTS} / 2</math>, <math>N = 8</math></p> <p>0110: <math>f_{\text{sampling}} = f_{DTS} / 4</math>, <math>N = 6</math></p> <p>0111: <math>f_{\text{sampling}} = f_{DTS} / 4</math>, <math>N = 8</math></p> <p>1000: <math>f_{\text{sampling}} = f_{DTS} / 8</math>, <math>N = 6</math></p> <p>1001: <math>f_{\text{sampling}} = f_{DTS} / 8</math>, <math>N = 8</math></p> <p>1010: <math>f_{\text{sampling}} = f_{DTS} / 16</math>, <math>N = 5</math></p> <p>1011: <math>f_{\text{sampling}} = f_{DTS} / 16</math>, <math>N = 6</math></p> <p>1100: <math>f_{\text{sampling}} = f_{DTS} / 16</math>, <math>N = 8</math></p> <p>1101: <math>f_{\text{sampling}} = f_{DTS} / 32</math>, <math>N = 5</math></p> <p>1110: <math>f_{\text{sampling}} = f_{DTS} / 32</math>, <math>N = 6</math></p> <p>1111: <math>f_{\text{sampling}} = f_{DTS} / 32</math>, <math>N = 8</math></p>
[3:0]	TRSEL	<p>触发源选择</p> <p>这些位用来选择用于计数器同步的触发输入源 (STI)</p> <p>0000: 通过设置 UEV1G 位软件触发</p> <p>0001: 通道 0 滤波输入 – TI0S0</p> <p>0010: 通道 1 滤波输入 – TI1S1</p> <p>0011: 外部触发输入 – ETIF</p> <p>1000: 通道 0 边沿检测器 – TI0BED</p> <p>1001: 内部定时器触发器 0 – ITI0</p> <p>1010: 内部定时器触发器 1 – ITI1</p> <p>1011: 内部定时器触发器 2 – ITI2</p> <p>其它: 默认 0</p> <p>注: 这些位只有在未被使用时, 才能进行更新, 即设置 SMSEL 字段为 0x00 除能从机模式。</p>

表 41. MCTM 内部触发器连接

从机定时器模块	ITI0	ITI1	ITI2
MCTM0	MCTM1	GPTM0	GPTM1
MCTM1	MCTM0	GPTM0	GPTM1

## 定时器计数器寄存器 – CTR

该寄存器定义了定时器使能位 (TME)、CRR 缓冲器使能位 (CRBE)、捕捉 / 比较控制位和通道 PDMA 选择位 (CHCCDS)。

偏移量: 0x010

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							CHCCDS
类型 / 复位							RW 0
15	14	13	12	11	10	9	8
保留位						COMUS	COMPRES
类型 / 复位						RW 0	RW 0
7	6	5	4	3	2	1	0
保留位						CRBE	TME
类型 / 复位						RW 0	RW 0

位	字段	描述
[16]	CHCCDS	通道捕捉 / 比较 PDMA 选择 0: 通道 PDMA 请求来自通道捕捉 / 比较事件 1: 通道 PDMA 请求来自更新事件 1
[9]	COMUS	捕捉 / 比较控制更新选择 0: 仅通过设置 UEV2G 位更新 1: 通过设置 UEV2G 位更新或当 STI 信号上升沿发生时更新 该位仅在通过设置 COMPRES 位为 1 使能捕捉 / 比较预载功能时有效。
[8]	COMPRES	捕捉 / 比较预载使能控制位 0: CHxE、CHxNE 和 CHxOM 位不会被预载 1: CHxE、CHxNE 和 CHxOM 位被预载 如果该位置 1, 当更新事件 2 发生时, 相关捕捉 / 比较控制位 CHxE、CHxNE 和 CHxOM 将会被更新。
[1]	CRBE	计数器重载寄存器缓冲器使能位 0: 计数器重载寄存器立即被更新 1: 直到更新事件发生时计数器重载寄存器才会被更新
[0]	TME	定时器使能位 0: MCTM 关闭 1: MCTM 开启 – MCTM 功能正常 该位清零时, 计数器停止计数且 MCTM 在单脉冲模式和从机触发模式以外任何工作模式下无功耗。在这两种模式下, TME 位可通过硬件自动置 1, 允许所有 MCTM 寄存器正常工作。

通道 0 输入配置寄存器 – CH0ICFR

该寄存器定义了通道 0 输入模式配置。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	TI0SRC	保留位						
类型 / 复位	RW	0						
	23	22	21	20	19	18	17	16
	保留位				CH0PSC		CH0CCS	
类型 / 复位					RW	0	RW	0
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位				TI0F			
类型 / 复位					RW	0	RW	0

位	字段	描述
[31]	TI0SRC	通道 0 输入源 TI0 选择 0: MTn_CH0 引脚连接到通道 0 输入 TI0 1: MTn_CH0、MTn_CH1 和 MTn_CH2 引脚异或操作的输出连接到通道 0 输入 TI0
[19:18]	CH0PSC	通道 0 捕捉输入源预分频器设置 这些位定义了通道 0 捕捉输入的有效事件。注意，一旦通道控制寄存器 CHCTR 中的通道 0 捕捉 / 比较使能位 CH0E 位被清零，则预分频器将被复位。 00: 无分频，每个有效事件为一个通道 0 捕捉输入信号 01: 每 2 个事件为一个通道 0 捕捉输入信号 10: 每 4 个事件为一个通道 0 捕捉输入信号 11: 每 8 个事件为一个通道 0 捕捉输入信号
[17:16]	CH0CCS	通道 0 捕捉 / 比较选项 00: 通道 0 配置为输出 01: 通道 0 配置为来自 TI0 信号的一个输入 10: 通道 0 配置为来自 TI1 信号的一个输入 11: 通道 0 配置为来自触发控制器产生的 TRCED 信号的一个输入 注：只有在 CH0E 位清零时，才可以访问 CH0CCS 字段。

位	字段	描述
[3:0]	TI0F	<p>通道 0 输入源 TI0 滤波器设置</p> <p>这些位定义了用来采样 TI0 信号的分频比。MCTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出一个滤波信号所需的有效转换次数。</p> <p>0000: 无滤波，采样时钟是 <math>f_{DTS}</math></p> <p>0001: <math>f_{sampling} = f_{CLKIN}</math>, <math>N = 2</math></p> <p>0010: <math>f_{sampling} = f_{CLKIN}</math>, <math>N = 4</math></p> <p>0011: <math>f_{sampling} = f_{CLKIN}</math>, <math>N = 8</math></p> <p>0100: <math>f_{sampling} = f_{DTS} / 2</math>, <math>N = 6</math></p> <p>0101: <math>f_{sampling} = f_{DTS} / 2</math>, <math>N = 8</math></p> <p>0110: <math>f_{sampling} = f_{DTS} / 4</math>, <math>N = 6</math></p> <p>0111: <math>f_{sampling} = f_{DTS} / 4</math>, <math>N = 8</math></p> <p>1000: <math>f_{sampling} = f_{DTS} / 8</math>, <math>N = 6</math></p> <p>1001: <math>f_{sampling} = f_{DTS} / 8</math>, <math>N = 8</math></p> <p>1010: <math>f_{sampling} = f_{DTS} / 16</math>, <math>N = 5</math></p> <p>1011: <math>f_{sampling} = f_{DTS} / 16</math>, <math>N = 6</math></p> <p>1100: <math>f_{sampling} = f_{DTS} / 16</math>, <math>N = 8</math></p> <p>1101: <math>f_{sampling} = f_{DTS} / 32</math>, <math>N = 5</math></p> <p>1110: <math>f_{sampling} = f_{DTS} / 32</math>, <math>N = 6</math></p> <p>1111: <math>f_{sampling} = f_{DTS} / 32</math>, <math>N = 8</math></p>

## 通道 1 输入配置寄存器 – CH1ICFR

该寄存器定义了通道 1 输入模式选项。

偏移量: 0x024

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位				CHIPSC		CH1CCS	
				RW	0	RW	0
15	14	13	12	11	10	9	8
保留位							
7	6	5	4	3	2	1	0
保留位				TIIF			
				RW	0	RW	0

位	字段	描述
[19:18]	CHIPSC	通道 1 捕捉输入源分频器设置 这些位定义了通道 1 捕捉输入的有效事件。注意，一旦通道控制寄存器 CHCTR 中的通道 1 捕捉 / 比较使能位 CH1E 被清零，则预分频器将被复位。 00: 无分频，每个有效事件为一个通道 1 捕捉输入信号 01: 每 2 个事件为一个通道 1 捕捉输入信号 10: 每 4 个事件为一个通道 1 捕捉输入信号 11: 每 8 个事件为一个通道 1 捕捉输入信号
[17:16]	CH1CCS	通道 1 捕捉 / 比较选项 00: 通道 1 被配置为输出 01: 通道 1 被配置为来自 TI1 信号的一个输入 10: 通道 1 被配置为来自 TI0 信号的一个输入 11: 通道 1 被配置为来自触发控制器产生的 TRCED 信号的一个输入 注：只有在 CH1E 位清零时，才可以访问 CH1CCS 字段。
[3:0]	TIIF	通道 1 输入源 TI1 滤波器设置 这些位定义了用来采样 TI1 信号的分频比。MCTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出一个滤波信号所需的有效转换次数。 0000: 无滤波，采样时钟是 $f_{DTS}$ 0001: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 2$ 0010: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 4$ 0011: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 8$ 0100: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 6$ 0101: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 8$ 0110: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 6$ 0111: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 8$ 1000: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 6$ 1001: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 8$ 1010: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 5$ 1011: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 6$ 1100: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 8$ 1101: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 5$ 1110: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 6$ 1111: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 8$



## 通道 2 输入配置寄存器 – CH2ICFR

该寄存器定义了通道 2 输入模式配置。

偏移量: 0x028

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位				CH2PSC		CH2CCS	
				RW	0	RW	0
15	14	13	12	11	10	9	8
保留位							
7	6	5	4	3	2	1	0
保留位				TI2F			
				RW	0	RW	0

位	字段	描述
[19:18]	CH2PSC	通道 2 捕捉输入源分频器设置 这些位定义了通道 2 捕捉输入的有效事件。注意，一旦通道控制寄存器 CHCTR 中的通道 2 捕捉 / 比较使能位 CH2E 被清零，则预分频器将被复位。 00: 无分频，每个有效事件为一个通道 2 捕捉输入信号 01: 每 2 个事件为一个通道 2 捕捉输入信号 10: 每 4 个事件为一个通道 2 捕捉输入信号 11: 每 8 个事件为一个通道 2 捕捉输入信号
[17:16]	CH2CCS	通道 2 捕捉 / 比较选项 00: 通道 2 被配置为输出 01: 通道 2 被配置为来自 TI2 信号的一个输入 10: 通道 2 被配置为来自 TI3 信号的一个输入 11: 通道 2 被配置为来自触发控制器产生的 TRCED 信号的一个输入 注: 只有在 CH2E 位清零时，才可以访问 CH2CCS 字段。
[3:0]	TI2F	通道 2 输入源 TI2 滤波器设置 这些位定义了用来采样 TI2 信号的分频比。MCTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出一个滤波信号所需的有效转换次数。 0000: 无滤波，采样时钟是 $f_{DTS}$ 0001: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 2$ 0010: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 4$ 0011: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 8$ 0100: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 6$ 0101: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 8$ 0110: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 6$ 0111: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 8$ 1000: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 6$ 1001: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 8$ 1010: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 5$ 1011: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 6$ 1100: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 8$ 1101: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 5$ 1110: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 6$ 1111: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 8$

## 通道 3 输入配置寄存器 – CH3ICFR

该寄存器定义了通道 3 输入模式配置。

偏移量: 0x02C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位				CH3PSC		CH3CCS		
				RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8	
类型 / 复位	保留位								
	7	6	5	4	3	2	1	0	
类型 / 复位	保留位				TI3F				
				RW	0	RW	0	RW	0

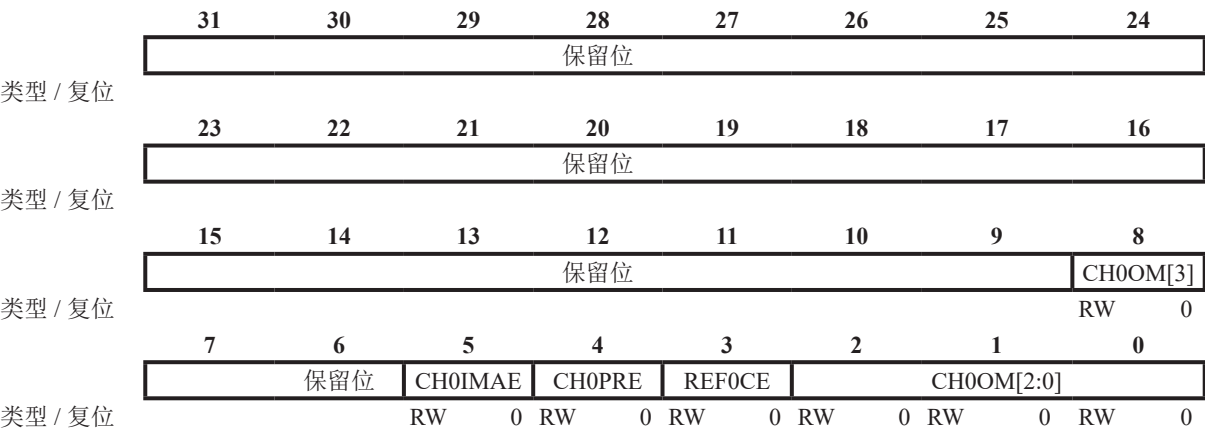
位	字段	描述
[19:18]	CH3PSC	通道 3 捕捉输入源分频器设置 这些位定义了通道 3 捕捉输入的有效事件。注意，一旦通道控制寄存器 CHCTR 中的通道 3 捕捉 / 比较使能位 CH3E 被清零，则预分频器将被复位。 00: 无分频，每个有效事件为一个通道 3 捕捉输入信号 01: 每 2 个事件为一个通道 3 捕捉输入信号 10: 每 4 个事件为一个通道 3 捕捉输入信号 11: 每 8 个事件为一个通道 3 捕捉输入信号
[17:16]	CH3CCS	通道 3 捕捉 / 比较选项 00: 通道 3 被配置为输出 01: 通道 3 被配置为来自 TI3 信号的一个输入 10: 通道 3 被配置为来自 TI2 信号的一个输入 11: 通道 3 被配置为来自触发控制器产生的 TRCED 信号的一个输入 注：只有在 CH3E 位清零时，才可以访问 CH3CCS 字段。
[3:0]	TI3F	通道 3 输入源 TI3 滤波器设置 这些位定义了用来采样 TI3 信号的分频比。GPTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出一个滤波信号所需的有效转换次数。 0000: 无滤波，采样时钟是 $f_{DTS}$ 0001: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 2$ 0010: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 4$ 0011: $f_{\text{sampling}} = f_{\text{CLKIN}}$ , $N = 8$ 0100: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 6$ 0101: $f_{\text{sampling}} = f_{\text{DTS}} / 2$ , $N = 8$ 0110: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 6$ 0111: $f_{\text{sampling}} = f_{\text{DTS}} / 4$ , $N = 8$ 1000: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 6$ 1001: $f_{\text{sampling}} = f_{\text{DTS}} / 8$ , $N = 8$ 1010: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 5$ 1011: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 6$ 1100: $f_{\text{sampling}} = f_{\text{DTS}} / 16$ , $N = 8$ 1101: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 5$ 1110: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 6$ 1111: $f_{\text{sampling}} = f_{\text{DTS}} / 32$ , $N = 8$

通道 0 输出配置寄存器 – CH0OCFR

该寄存器定义了通道 0 输出模式配置。

偏移量：0x040

复位值：0x0000\_0000



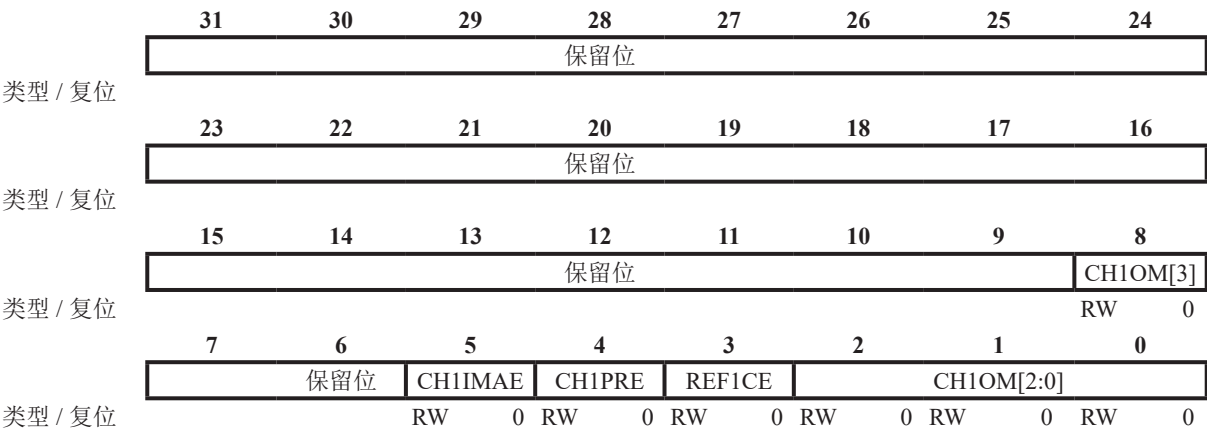
位	字段	描述
[5]	CH0IMAE	通道 0 立即有效使能位 0: 无动作 1: 单脉冲立即有效模式使能 无论 CNTR 和 CH0CCR 值的比较结果如何，在一个有效触发事件发生后，CH0OREF 会立即强制为比较匹配电平。 有效时间将在下一个上溢或下溢事件到来时自动结束。 注：只有当通道 0 被配置工作在 PWM 模式 1 或 PWM 模式 2 时，CH0IMAE 位可用。
[4]	CH0PRE	通道 0 捕捉 / 比较寄存器 (CH0CCR) 预载使能位 0: CH0CCR 预载功能除能 当 CH0PRE 位清零，CH0CCR 寄存器将立即被更新为新的值，且立即可用。 1: CH0CCR 预载功能使能 直到更新事件 1 发生后，新的 CH0CCR 值才会被传送到影子寄存器中。
[3]	REF0CE	通道 0 参考输出清除使能位 0: CH0OREF 正常执行，不受 ETIF 信号影响 1: CH0OREF 在来自于 MTn_ETI 引脚的 ETIF 信号的高电平处被强制为 0

位	字段	描述
[8][2:0]	CH0OM[3:0]	<p>通道 0 输出模式设置</p> <p>这些位定义了输出参考信号 CH0OREF 的功能类型</p> <p>0000: 无变化</p> <p>0001: 比较匹配时输出 0</p> <p>0010: 比较匹配时输出 1</p> <p>0011: 比较匹配时输出翻转</p> <p>0100: 强制无效 – CH0OREF 强制为 0</p> <p>0101: 强制有效 – CH0OREF 强制为 1</p> <p>0110: PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH0CCR, 通道 0 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH0CCR, 通道 0 处于无效电平, 否则将处于有效电平。</li> </ul> <p>0111: PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH0CCR, 通道 0 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH0CCR, 通道 0 处于有效电平, 否则将处于无效电平。</li> </ul> <p>1110: 非对称 PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH0CCR, 通道 0 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH0CCR, 通道 0 处于无效电平, 否则将处于有效电平。</li> </ul> <p>1111: 非对称 PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH0CCR, 通道 0 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH0CCR, 通道 0 处于有效电平, 否则将处于无效电平。</li> </ul> <p>注: 当通道 0 用作非对称 PWM 输出模式时, 计数器配置寄存器中的计数器模式选择位必须配置为中心对齐计数模式 (CMSEL = 0x1 / 0x2 / 0x3)。</p>

通道 1 输出配置寄存器 – CH1OCFR

该寄存器定义了通道 1 输出模式配置。

偏移量： 0x044  
复位值： 0x0000\_0000



位	字段	描述
[5]	CH1MAE	通道 1 立即有效使能位 0: 无动作 1: 单脉冲立即有效模式使能 无论 CNTR 和 CH1CCR 值的比较结果如何，在一个有效触发事件发生后，CH1OREF 会立即强制为比较匹配电平。 有效时间将在下一个上溢或下溢事件到来时自动结束。 注：只有当通道 1 被配置工作在 PWM 模式 1 或 PWM 模式 2 时，CH1MAE 位可用。
[4]	CH1PRE	通道 1 捕捉 / 比较寄存器 (CH1CCR) 预载使能位 0: CH1CCR 预载功能除能 当 CH1PRE 位清零，CH1CCR 寄存器将立即被更新为新的值，且立即可用。 1: CH1CCR 预载功能使能 直到更新事件 1 发生后，新的 CH1CCR 值才会被传送到影子寄存器中。
[3]	REF1CE	通道 1 参考输出清除使能位 0: CH1OREF 正常执行，不受 ETIF 信号影响 1: CH1OREF 在来自于 MTn_ETI 引脚的 ETIF 信号的高电平处被强制为 0

位	字段	描述
[8][2:0]	CH1OM[3:0]	<p>通道 1 输出模式设置</p> <p>这些位定义了输出参考信号 CH1OREF 的功能类型</p> <p>0000: 无变化</p> <p>0001: 比较匹配时输出 0</p> <p>0010: 比较匹配时输出 1</p> <p>0011: 比较匹配时输出翻转</p> <p>0100: 强制无效 – CH1OREF 强制为 0</p> <p>0101: 强制有效 – CH1OREF 强制为 1</p> <p>0110: PWM 模式 1</p> <ul style="list-style-type: none"><li>– 在向上计数期间, 当 CNTR &lt; CH1CCR, 通道 1 处于有效电平, 否则将处于无效电平。</li><li>– 在向下计数期间, 当 CNTR &gt; CH1CCR, 通道 1 处于无效电平, 否则将处于有效电平。</li></ul> <p>0111: PWM 模式 2</p> <ul style="list-style-type: none"><li>– 在向上计数期间, 当 CNTR &lt; CH1CCR, 通道 1 处于无效电平, 否则将处于有效电平。</li><li>– 在向下计数期间, 当 CNTR &gt; CH1CCR, 通道 1 处于有效电平, 否则将处于无效电平。</li></ul> <p>1110: 非对称 PWM 模式 1</p> <ul style="list-style-type: none"><li>– 在向上计数期间, 当 CNTR &lt; CH1CCR, 通道 1 处于有效电平, 否则将处于无效电平。</li><li>– 在向下计数期间, 当 CNTR &gt; CH1ACR, 通道 1 处于无效电平, 否则将处于有效电平。</li></ul> <p>1111: 非对称 PWM 模式 2</p> <ul style="list-style-type: none"><li>– 在向上计数期间, 当 CNTR &lt; CH1CCR, 通道 1 处于无效电平, 否则将处于有效电平。</li><li>– 在向下计数期间, 当 CNTR &gt; CH1ACR, 通道 1 处于有效电平, 否则将处于无效电平。</li></ul> <p>注: 当通道 1 用作非对称 PWM 输出模式时, 计数器配置寄存器中的计数器模式选择位必须配置为中心对齐计数模式 (CMSEL = 0x1 / 0x2 / 0x3)</p>



位	字段	描述
[8][2:0]	CH2OM[3:0]	<p>通道 2 输出模式设置位</p> <p>这些位定义了输出参考信号 CH2OREF 的功能类型</p> <p>0000: 无变化</p> <p>0001: 比较匹配时输出 0</p> <p>0010: 比较匹配时输出 1</p> <p>0011: 比较匹配时输出翻转</p> <p>0100: 强制无效 – CH2OREF 强制为 0</p> <p>0101: 强制有效 – CH2OREF 强制为 1</p> <p>0110: PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH2CCR, 通道 2 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH2CCR, 通道 2 处于无效电平, 否则将处于有效电平。</li> </ul> <p>0111: PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH2CCR, 通道 2 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH2CCR, 通道 2 处于有效电平, 否则将处于无效电平。</li> </ul> <p>1110: 非对称 PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH2CCR, 通道 2 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH2CCR, 通道 2 处于无效电平, 否则将处于有效电平。</li> </ul> <p>1111: 非对称 PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH2CCR, 通道 2 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH2CCR, 通道 2 处于有效电平, 否则将处于无效电平。</li> </ul> <p>注: 当通道 2 用作非对称 PWM 输出模式时, 计数器配置寄存器中的计数器模式选择位必须配置为中心对齐计数模式 (CMSEL = 0x1 / 0x2 / 0x3)。</p>





位	字段	描述
[8][2:0]	CH3OM[3:0]	<p>通道 3 输出模式设置位</p> <p>这些位定义了输出参考信号 CH3OREF 的功能类型。</p> <p>0000: 无变化</p> <p>0001: 比较匹配时输出 0</p> <p>0010: 比较匹配时输出 1</p> <p>0011: 比较匹配时输出翻转</p> <p>0100: 强制无效 – CH3OREF 强制为 0</p> <p>0101: 强制有效 – CH3OREF 强制为 1</p> <p>0110: PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH3CCR, 通道 3 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH3CCR, 通道 3 处于无效电平, 否则将处于有效电平。</li> </ul> <p>0111: PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH3CCR, 通道 3 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH3CCR, 通道 3 处于有效电平, 否则将处于无效电平。</li> </ul> <p>1110: 非对称 PWM 模式 1</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH3CCR, 通道 3 处于有效电平, 否则将处于无效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH3CCR, 通道 3 处于无效电平, 否则将处于有效电平。</li> </ul> <p>1111: 非对称 PWM 模式 2</p> <ul style="list-style-type: none"> <li>– 在向上计数期间, 当 CNTR &lt; CH3CCR, 通道 3 处于无效电平, 否则将处于有效电平。</li> <li>– 在向下计数期间, 当 CNTR &gt; CH3CCR, 通道 3 处于有效电平, 否则将处于无效电平。</li> </ul> <p>注: 当通道 3 用作非对称 PWM 输出模式时, 计数器配置寄存器中的计数器模式选择位必须配置为中心对齐计数模式 (CMSEL = 0x1 / 0x2 / 0x3)。</p>

## 通道控制寄存器 – CHCTR

该寄存器包含了通道捕捉输入和比较输出功能使能控制位。

偏移量: 0x050

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位	CH3E	CH2NE	CH2E	CH1NE	CH1E	CH0NE	CH0E
类型 / 复位	RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW

位	字段	描述
[6]	CH3E	通道 3 捕捉 / 比较使能位 – 通道 3 被配置作为输入 (CH3CCS = 0x01 / 0x02 / 0x03) 0: 输入捕捉模式除能 1: 输入捕捉模式使能 – 通道 3 被配置作为输出 (CH3CCS = 0x00) 0: 关闭 – 通道 3 输出信号 CH3O 无作用 1: 开启 – 通道 3 输出信号 CH3O 在相应的输出脚产生, 取决于 CHMOE、CHOSI、CHOSSR 和 CH3OIS 位的条件。
[5]	CH2NE	通道 2 捕捉 / 比较互补使能位 0: 关闭 – 通道 2 互补输出信号 CH2NO 无作用。CH2NO 电平取决于 CHMOE、CHOSI、CHOSSR、CH2OIS、CH2OISN 和 CH2E 位。 1: 开启 – 通道 2 互补输出信号 CH2NO 在相应的输出脚产生, 取决于 CHMOE、CHOSI、CHOSSR、CH2OIS、CH2OISN 和 CH2E 位的条件。
[4]	CH2E	通道 2 捕捉 / 比较使能位 – 通道 2 被配置作为输入 (CH2CCS = 0x01 / 0x02 / 0x03) 0: 输入捕捉模式除能 1: 输入捕捉模式使能 – 通道 2 被配置作为输出 (CH2CCS = 0x00) 0: 关闭 – 通道 2 输出信号 CH2O 无作用。CH2O 电平取决于 CHMOE、CHOSI、CHOSSR、CH2OIS、CH2OISN 和 CH2NE 位的条件。 1: 开启 – 通道 2 互补输出信号 CH2O 在相应的输出脚产生, 取决于 CHMOE、CHOSI、CHOSSR、CH2OIS、CH2OISN 和 CH2NE 位的条件。
[3]	CH1NE	通道 1 捕捉 / 比较互补使能位 0: 关闭 – 通道 1 互补输出信号 CH1NO 无作用。CH1NO 电平取决于 CHMOE、CHOSI、CHOSSR、CH1OIS、CH1OISN 和 CH1E 位的条件。 1: 开启 – 通道 1 互补输出信号 CH1NO 在相应的输出脚产生, 取决于 CHMOE、CHOSI、CHOSSR、CH1OIS、CH1OISN 和 CH1E 位的条件。

位	字段	描述
[2]	CH1E	<p>通道 1 捕捉 / 比较使能位</p> <p>– 通道 1 被配置作为输入 (CH1CCS = 0x01 / 0x02 / 0x03)</p> <p>0: 输入捕捉模式除能</p> <p>1: 输入捕捉模式使能</p> <p>– 通道 1 被配置作为输出 (CH1CCS = 0x00)</p> <p>0: 关闭 – 通道 1 输出信号无作用。CH1O 电平取决于 CHMOE、CHOSI、CHOSSR、CH1OIS、CH1OISN 和 CH1NE 位的条件。</p> <p>1: 开启 – 通道 1 输出信号 CH1O 在相应的输出脚产生，取决于 CHMOE、CHOSI、CHOSSR、CH1OIS、CH1OISN 和 CH1NE 位的条件。</p>
[1]	CH0NE	<p>通道 0 捕捉 / 比较互补使能位</p> <p>0: 关闭 – 通道 0 互补输出信号 CH0NO 无作用。CH0NO 电平取决于 CHMOE、CHOSI、CHOSSR、CH0OIS、CH0OISN 和 CH0E 位的条件。</p> <p>1: 开启 – 通道 0 互补输出信号 CH0NO 在相应的输出脚产生，取决于 CHMOE、CHOSI、CHOSSR、CH0OIS、CH0OISN 和 CH0E 位的条件。</p>
[0]	CH0E	<p>通道 0 捕捉 / 比较使能位</p> <p>– 通道 0 被配置作为输入 (CH0CCS = 0x01 / 0x02 / 0x03)</p> <p>0: 输入捕捉模式除能</p> <p>1: 输入捕捉模式使能</p> <p>– 通道 0 被配置作为输出 (CH0CCS = 0x00)</p> <p>0: 关闭 – 通道 0 输出信号 CH0O 无作用。CH0O 电平取决于 CHMOE、CHOSI、CHOSSR、CH0OIS、CH0OISN 和 CH0NE 位的条件。</p> <p>1: 开启 – 通道 0 输出信号 CH0O 在相应的输出脚产生，取决于 CHMOE、CHOSI、CHOSSR、CH0OIS、CH0OISN 和 CH0NE 位的条件。</p>

## 通道极性配置寄存器 – CHPOLR

该寄存器包含了通道捕捉输入 / 比较输出极性控制。

偏移量: 0x054

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位	CH3P	CH2NP	CH2P	CH1NP	CH1P	CH0NP	CH0P
	RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW
类型 / 复位							

位	字段	描述
[6]	CH3P	通道 3 捕捉 / 比较极性位 (CH3CCS = 0x01 / 0x02 / 0x03) – 通道 3 被配置为输入时 0: 捕捉事件发生在通道 3 上升沿 1: 捕捉事件发生在通道 3 下降沿 – 通道 3 被配置为输出时 (CH3CCS = 0x00) 0: 通道 3 输出高电平有效 1: 通道 3 输出低电平有效
[5]	CH2NP	通道 2 捕捉 / 比较互补极性位 0: 通道 2 输出高电平有效 1: 通道 2 输出低电平有效
[4]	CH2P	通道 2 捕捉 / 比较极性位 (CH2CCS = 0x01 / 0x02 / 0x03) – 通道 2 被配置为输入时 0: 捕捉事件发生在通道 2 上升沿 1: 捕捉事件发生在通道 2 下降沿 – 通道 2 被配置为输出时 (CH2CCS = 0x00) 0: 通道 2 输出高电平有效 1: 通道 2 输出低电平有效
[3]	CH1NP	通道 1 捕捉 / 比较互补极性位 0: 通道 1 输出高电平有效 1: 通道 1 输出低电平有效
[2]	CH1P	通道 1 捕捉 / 比较极性位 (CH1CCS = 0x01 / 0x02 / 0x03) – 通道 1 被配置为输入时 0: 捕捉事件发生在通道 1 上升沿 1: 捕捉事件发生在通道 1 下降沿 – 通道 1 被配置为输出时 (CH1CCS = 0x00) 0: 通道 1 输出高电平有效 1: 通道 1 输出低电平有效
[1]	CH0NP	通道 0 捕捉 / 比较互补极性位 0: 通道 0 输出高电平有效 1: 通道 0 输出低电平有效

位	字段	描述
[0]	CH0P	通道 0 捕捉 / 比较极性位 (CH0CCS = 0x01 / 0x02 / 0x03) – 通道 0 被配置为输入时 0: 捕捉事件发生在通道 0 上升沿 1: 捕捉事件发生在通道 0 下降沿 – 通道 0 被配置为输出时 (CH0CCS = 0x00) 0: 通道 0 输出高电平有效 1: 通道 0 输出低电平有效

### 通道刹车配置寄存器 – CHBRKCFR

该寄存器定义了使用刹车功能时通道输出空闲状态。

偏移量: 0x06C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位	CH3OIS	CH2OISN	CH2OIS	CH1OISN	CH1OIS	CH0OISN	CH0OIS
类型 / 复位		RW	0 RW	0 RW	0 RW	0 RW	0 RW	0

位	字段	描述
[6]	CH3OIS	MTn_CH3O 输出空闲状态 0: CHMOE = 0, 通道 3 输出 CH3O = 0 1: CHMOE = 0, 通道 3 输出 CH3O = 1
[5]	CH2OISN	MTn_CH2NO 输出空闲状态 0: CHMOE = 0, 经过一个死区时间之后, 通道 2 互补输出 CH2NO = 0 1: CHMOE = 0, 经过一个死区时间之后, 通道 2 互补输出 CH2NO = 1
[4]	CH2OIS	MTn_CH2O 输出空闲状态 0: CHMOE = 0, 经过一个死区时间之后, 通道 2 输出 CH2O = 0 1: CHMOE = 0, 经过一个死区时间之后, 通道 2 输出 CH2O = 1
[3]	CH1OISN	MTn_CH1NO 输出空闲状态 0: CHMOE = 0, 经过一个死区时间之后, 通道 1 互补输出 CH1NO = 0 1: CHMOE = 0, 经过一个死区时间之后, 通道 1 互补输出 CH1NO = 1
[2]	CH1OIS	MTn_CH1O 输出空闲状态 0: CHMOE = 0, 经过一个死区时间之后, 通道 1 输出 CH1O = 0 1: CHMOE = 0, 经过一个死区时间之后, 通道 1 输出 CH1O = 1
[1]	CH0OISN	MTn_CH0NO 输出空闲状态 0: CHMOE = 0, 经过一个死区时间之后, 通道 0 互补输出 CH0NO = 0 1: CHMOE = 0, 经过一个死区时间之后, 通道 0 互补输出 CH0NO = 1

位	字段	描述
[0]	CH0OIS	MTn_CH0O 输出空闲状态 0: CHMOE = 0, 经过一个死区时间之后, 通道 0 输出 CH0O = 0 1: CHMOE = 0, 经过一个死区时间之后, 通道 0 输出 CH0O = 1

## 通道刹车控制寄存器 – CHBRKCTR

该寄存器定义了通道刹车控制位。

偏移量: 0x070  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	CHDTG							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	保留位		CHOSSR	CHOSSI	GFSEL1	GFSEL0	LOCKLV	
类型 / 复位			RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	BKF1				BKF0			
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	保留位	BK1SEL	CHAOE	CHMOE	BKP1	BKE1	BKP0	BKE0
类型 / 复位		RW	0	RW	0	RW	0	RW

位	字段	描述
[31:24]	CHDTG	通道死区时间周期定义 CHDTG[7:5] = 0xx: 通道死区时间 = CHDTG[7:0] × t <sub>dtg</sub> , t <sub>dtg</sub> = t <sub>DTS</sub> CHDTG[7:5] = 10x: 通道死区时间 = (64 + CHDTG[5:0]) × t <sub>dtg</sub> , t <sub>dtg</sub> = 2 × t <sub>DTS</sub> CHDTG[7:5] = 110: 通道死区时间 = (32 + CHDTG[4:0]) × t <sub>dtg</sub> , t <sub>dtg</sub> = 8 × t <sub>DTS</sub> CHDTG[7:5] = 111: 通道死区时间 = (32 + CHDTG[4:0]) × t <sub>dtg</sub> , t <sub>dtg</sub> = 16 × t <sub>DTS</sub>
[21]	CHOSSR	正常运行状态 (CHMOE = 1) 下通道关闭状态 (CHxE = 0, CHxNE = 0) 选择 0: 无效时 MTn_CHxO / MTn_CHxNO 输出除能 – 非定时器驱动 1: 无效时 MTn_CHxO / MTn_CHxNO 输出使能, 电平无效
[20]	CHOSSI	空闲模式 (CHMOE = 0) 下通道关闭状态选择 0: 无效时 MTn_CHxO / MTn_CHxNO 输出除能 – 非定时器驱动 1: 无效时 MTn_CHxO / MTn_CHxNO 输出使能, 空闲电平取决于 CHxOIS 和 CHxOISN 位的条件。
[19]	GFSEL1	刹车信号 1 的抗干扰滤波器选择 0: 无输入抗干扰滤波器 1: 50 ns 的抗干扰滤波器
[18]	GFSEL	刹车信号 0 的抗干扰滤波器选择 0: 无输入抗干扰滤波器 1: 50 ns 的抗干扰滤波器

位	字段	描述
[17:16]	LOCKLV	<p>锁定级别设置</p> <p>这些位提供写保护以防软件出错。复位后，这些位只能被写入一次。</p> <p>00: 锁定功能关闭。寄存器写保护功能除能。</p> <p>01: 锁定级别 1</p> <p>10: 锁定级别 2</p> <p>11: 锁定级别 3</p>
[15:12]	BKF1	<p>刹车信号 1 输入滤波器设置</p> <p>这些位定义了用来采样 MTn_BRK1 信号的分频比。MCTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出一个滤波信号所需的有效转换次数。</p> <p>0000: 无滤波，无需采样时钟</p> <p>0001: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 2</math></p> <p>0010: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 4</math></p> <p>0011: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 8</math></p> <p>0100: <math>f_{\text{sampling}} = f_{\text{DTS}} / 2</math>, <math>N = 6</math></p> <p>0101: <math>f_{\text{sampling}} = f_{\text{DTS}} / 2</math>, <math>N = 8</math></p> <p>0110: <math>f_{\text{sampling}} = f_{\text{DTS}} / 4</math>, <math>N = 6</math></p> <p>0111: <math>f_{\text{sampling}} = f_{\text{DTS}} / 4</math>, <math>N = 8</math></p> <p>1000: <math>f_{\text{sampling}} = f_{\text{DTS}} / 8</math>, <math>N = 6</math></p> <p>1001: <math>f_{\text{sampling}} = f_{\text{DTS}} / 8</math>, <math>N = 8</math></p> <p>1010: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 5</math></p> <p>1011: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 6</math></p> <p>1100: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 8</math></p> <p>1101: <math>f_{\text{sampling}} = f_{\text{DTS}} / 32</math>, <math>N = 5</math></p> <p>1110: <math>f_{\text{sampling}} = f_{\text{DTS}} / 32</math>, <math>N = 6</math></p> <p>1111: <math>f_{\text{sampling}} = f_{\text{DTS}} / 32</math>, <math>N = 8</math></p>
[11:8]	BKF0	<p>刹车信号 0 输入滤波器设置</p> <p>这些位定义了用来采样 MTn_BRK0 信号的分频比。MCTM 中的数字滤波器是一个 N 事件计数器，N 代表能够输出一个滤波信号所需的有效转换次数。</p> <p>0000: 无滤波，无需采样时钟</p> <p>0001: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 2</math></p> <p>0010: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 4</math></p> <p>0011: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 8</math></p> <p>0100: <math>f_{\text{sampling}} = f_{\text{DTS}} / 2</math>, <math>N = 6</math></p> <p>0101: <math>f_{\text{sampling}} = f_{\text{DTS}} / 2</math>, <math>N = 8</math></p> <p>0110: <math>f_{\text{sampling}} = f_{\text{DTS}} / 4</math>, <math>N = 6</math></p> <p>0111: <math>f_{\text{sampling}} = f_{\text{DTS}} / 4</math>, <math>N = 8</math></p> <p>1000: <math>f_{\text{sampling}} = f_{\text{DTS}} / 8</math>, <math>N = 6</math></p> <p>1001: <math>f_{\text{sampling}} = f_{\text{DTS}} / 8</math>, <math>N = 8</math></p> <p>1010: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 5</math></p> <p>1011: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 6</math></p> <p>1100: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 8</math></p> <p>1101: <math>f_{\text{sampling}} = f_{\text{DTS}} / 32</math>, <math>N = 5</math></p> <p>1110: <math>f_{\text{sampling}} = f_{\text{DTS}} / 32</math>, <math>N = 6</math></p> <p>1111: <math>f_{\text{sampling}} = f_{\text{DTS}} / 32</math>, <math>N = 8</math></p>
[6]	BK1SEL	<p>刹车信号 1 选择</p> <p>0: MTn_ETI 引脚选择 ETI 信号</p> <p>1: MTn_ETI 引脚选择 MTn_BRK1 信号</p> <p>注: 刹车信号 1 和 ETI 信号共用 MTn_ETI 引脚。该控制位可将引脚功能切换为 MCTM 的第二个刹车信号。</p>
[5]	CHAOE	<p>通道自动输出使能位</p> <p>0: CHMOE 仅可通过软件设置</p> <p>1: CHMOE 可通过软件设置或自动通过更新事件设置</p>



位	字段	描述
[4]	CHMOE	通道主要输出使能位 刹车事件发生时通过硬件异步清除。 0: MTn_CHxO 和 MTn_CHxNO 除能或强制为空闲状态 1: 如果使能位 CHxE 和 CHxNE 置位, 那么 MTn_CHxO 和 MTn_CHxNO 使能
[3]	BKP1	刹车信号 1 输入极性位 0: 刹车输入低电平有效 1: 刹车输入高电平有效
[2]	BKE1	刹车信号 1 使能位 0: 刹车输入除能 1: 刹车输入使能
[1]	BKP0	刹车信号 0 输入极性位 0: 刹车输入低电平有效 1: 刹车输入高电平有效
[0]	BKE0	刹车信号 0 使能位 0: 刹车输入除能 1: 刹车输入使能

## 定时器 PDMA / 中断控制寄存器 – DICTR

该寄存器包含了定时器 PDMA 和中断使能控制位。

偏移量: 0x074

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位					TEVDE	UEV2DE	UEV1DE
类型 / 复位						RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	保留位				CH3CCDE	CH2CCDE	CH1CCDE	CH0CCDE
类型 / 复位					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	保留位				BRKIE	TEVIE	UEV2IE	UEV1IE
类型 / 复位					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	保留位				CH3CCIE	CH2CCIE	CH1CCIE	CH0CCIE
类型 / 复位					RW 0	RW 0	RW 0	RW 0

位	字段	描述
[26]	TEVDE	触发事件 PDMA 请求使能位 0: 触发 PDMA 请求除能 1: 触发 PDMA 请求使能
[25]	UEV2DE	更新事件 2 PDMA 请求使能位 0: 更新事件 2 PDMA 请求除能 1: 更新事件 2 PDMA 请求使能
[24]	UEV1DE	更新事件 1 PDMA 请求使能位 0: 更新事件 1 PDMA 请求除能 1: 更新事件 1 PDMA 请求使能
[19]	CH3CCDE	通道 3 捕捉 / 比较 PDMA 请求使能位 0: 通道 3 PDMA 请求除能 1: 通道 3 PDMA 请求使能
[18]	CH2CCDE	通道 2 捕捉 / 比较 PDMA 请求使能位 0: 通道 2 PDMA 请求除能 1: 通道 2 PDMA 请求使能
[17]	CH1CCDE	通道 1 捕捉 / 比较 PDMA 请求使能位 0: 通道 1 PDMA 请求除能 1: 通道 1 PDMA 请求使能
[16]	CH0CCDE	通道 0 捕捉 / 比较 PDMA 请求使能位 0: 通道 0 PDMA 请求除能 1: 通道 0 PDMA 请求使能
[11]	BRKIE	刹车事件中断使能位 0: 刹车事件中断除能 1: 刹车事件中断使能
[10]	TEVIE	触发事件中断使能位 0: 触发事件中断除能 1: 触发事件中断使能
[9]	UEV2IE	更新事件 2 中断使能位 0: 更新事件 2 中断除能 1: 更新事件 2 中断使能

位	字段	描述
[8]	UEV1IE	更新事件 1 中断使能位 0: 更新事件 1 中断除能 1: 更新事件 1 中断使能
[3]	CH3CCIE	通道 3 捕捉 / 比较中断使能位 0: 通道 3 中断除能 1: 通道 3 中断使能
[2]	CH2CCIE	通道 2 捕捉 / 比较中断使能位 0: 通道 2 中断除能 1: 通道 2 中断使能
[1]	CH1CCIE	通道 1 捕捉 / 比较中断使能位 0: 通道 1 中断除能 1: 通道 1 中断使能
[0]	CH0CCIE	通道 0 捕捉 / 比较中断使能位 0: 通道 0 中断除能 1: 通道 0 中断使能

## 定时器事件发生器寄存器 – EVGR

该寄存器包含了软件事件发生控制位。

偏移量: 0x078

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位				BRKG	TEVG	UEV2G	UEV1G
				WO	0	WO	0
				WO	0	WO	0
				WO	0	WO	0
				WO	0	WO	0
7	6	5	4	3	2	1	0
保留位				CH3CCG	CH2CCG	CH1CCG	CH0CCG
				WO	0	WO	0
				WO	0	WO	0
				WO	0	WO	0
				WO	0	WO	0

位	字段	描述
[11]	BRKG	软件刹车事件发生 刹车事件 BEV 可由设置此位来使其发生。它由硬件自动清零。 0: 无动作 1: BRK0IF 标志位置位, 接着 CHMOE 位清零
[10]	TEVG	触发事件发生 触发事件 TEV 可由设置此位来使其发生。它由硬件自动清零。 0: 无动作 1: TEVIF 标志位置位
[9]	UEV2G	更新事件 2 发生 更新事件 2 UEV2 可由设置此位来使其发生。它由硬件自动清零。 0: 无动作 1: CTR 寄存器的 COMPRE 位置位时, 更新 CHxE、CHxNE 和 CHxOM 位
[8]	UEV1G	更新事件 1 发生 更新事件 1 UEV1 可由设置此位来使其发生。它由硬件自动清零。 0: 无动作 1: 重新初始化计数器 计数器值返回 0 或 CRR 预载值, 这取决于当前定时器使用的计数器模式。相关寄存器也会被更新。欲知详细描述请参考相关章节。
[3]	CH3CCG	通道 3 捕捉 / 比较发生 通道 3 捕捉 / 比较匹配事件可由此位来使其发生。它由硬件自动清零。 0: 无动作 1: 通道 3 发生捕捉 / 比较匹配事件 如果通道 3 被配置为输入, 计数器的值将被捕捉到 CH3CCR 寄存器, 接着 CH3CCIF 位被置位。如果通道 3 被配置为输出, 则 CH3CCIF 被置位。
[2]	CH2CCG	通道 2 捕捉 / 比较发生 通道 2 捕捉 / 比较匹配事件可由此位来使其发生。它由硬件自动清零。 0: 无动作 1: 通道 2 发生捕捉 / 比较匹配事件 如果通道 2 被配置为输入, 计数器的值将被捕捉到 CH2CCR 寄存器, 接着 CH2CCIF 位被置位。如果通道 2 被配置为输出, 则 CH2CCIF 被置位。

位	字段	描述
[1]	CH1CCG	通道 1 捕捉 / 比较发生器 通道 1 捕捉 / 比较匹配事件可由此位来使其发生。它由硬件自动清零。 0: 无动作 1: 通道 1 发生捕捉 / 比较匹配事件 如果通道 1 被配置为输入, 计数器的值将被捕捉到 CH1CCR 寄存器, 接着 CH1CCIF 位被置位。如果通道 1 被配置为输出, 则 CH1CCIF 被置位。
[0]	CH0CCG	通道 0 捕捉 / 比较发生器 通道 0 捕捉 / 比较匹配事件可通过设置此位发生。它由硬件自动清零。 0: 无动作 1: 通道 0 发生捕捉 / 比较匹配事件 如果通道 0 被配置为输入, 计数器的值将被捕捉到 CH0CCR 寄存器, 接着 CH0CCIF 位被置位。如果通道 0 被配置为输出, 则 CH0CCIF 被置位。

### 定时器中断状态寄存器 – INTSR

该寄存器存储了定时器中断的状态。

偏移量: 0x07C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位			BRK1IF	BRK0IF	TEVIF	UEV2IF	UEV1IF
类型 / 复位				W0C	0	W0C	0	W0C
	7	6	5	4	3	2	1	0
	CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF
类型 / 复位	W0C	0	W0C	0	W0C	0	W0C	0

位	字段	描述
[12]	BRK1IF	刹车事件 1 中断标志位 此位在刹车事件 1 发生时通过硬件被置位并通过软件清零。 0: 无刹车事件 1 发生 1: 刹车事件 1 发生
[11]	BRK0IF	刹车事件 0 中断标志位 此位在刹车事件 0 发生时通过硬件被置位并通过软件清零。 0: 无刹车事件 0 发生 1: 刹车事件 0 发生
[10]	TEVIF	触发事件中断标志位 此位在触发事件发生时通过硬件被置位并通过软件清零。 0: 无触发事件发生 1: 触发事件发生

位	字段	描述
[9]	UEV2IF	更新事件 2 中断标志位 此位在更新事件 2 发生时通过硬件被置位并通过软件清零。 0: 无更新事件 2 发生 1: 更新事件 2 发生
[8]	UEV1IF	更新事件 1 中断标志位 此位在更新事件 1 发生时通过硬件被置位并通过软件清零。 0: 无更新事件 1 发生 1: 更新事件 1 发生 注: 更新事件在以下情况下发生: – 计数器上溢或下溢 – UEV1G 位被置位, 并且 UEV1DIS = 0 – 从机重启模式 (UEV1DIS = 0) 下接收到 STI 上升沿信号
[7]	CH3OCF	通道 3 过度捕捉标志位 此位由硬件置位, 并由软件清零。 0: 没有检测到过度捕捉事件 1: 当 CH3CCIF 位已被置位且还没有被软件清零时, 捕捉事件再次发生。
[6]	CH2OCF	通道 2 过度捕捉标志位 此位由硬件置位, 并由软件清零。 0: 没有检测到过度捕捉事件 1: 当 CH2CCIF 位已被置位且还没有被软件清零时, 捕捉事件再次发生。
[5]	CH1OCF	通道 1 过度捕捉标志位 此位由硬件置位, 并由软件清零。 0: 没有检测到过度捕捉事件 1: 当 CH1CCIF 位已被置位且还没有被软件清零时, 捕捉事件再次发生。
[4]	CH0OCF	通道 0 过度捕捉标志位 此位由硬件置位, 并由软件清零。 0: 没有检测到过度捕捉事件 1: 当 CH0CCIF 位已被置位且还没有被软件清零时, 捕捉事件再次发生。
[3]	CH3CCIF	通道 3 捕捉 / 比较中断标志位 – 通道 3 配置为输出时: 0: 无匹配事件发生 1: 计数器 CNTR 内容与 CH3CCR 寄存器内容匹配 除中心对齐计数模式外, 当计数器的值与 CH3CCR 的值匹配时, 此位被硬件置位。 通过软件清零。 – 通道 3 配置为输入时: 0: 无输入捕捉发生 1: 输入捕捉发生 此位在捕捉事件发生时被硬件置位。通过软件或通过读取 CH3CCR 寄存器清零。
[2]	CH2CCIF	通道 2 捕捉 / 比较中断标志位 – 通道 2 配置为输出时: 0: 无匹配事件发生 1: 计数器 CNTR 内容与 CH2CCR 寄存器内容匹配 除中心对齐计数模式外, 当计数器的值与 CH2CCR 的值匹配时, 此位被硬件置位。 通过软件清零。 – 通道 2 配置为输入时: 0: 无输入捕捉发生 1: 输入捕捉发生 此位在捕捉事件发生时被硬件置位。通过软件或通过读取 CH2CCR 寄存器清零。

位	字段	描述
[1]	CH1CCIF	通道 1 捕捉 / 比较中断标志位 – 通道 1 配置为输出时： 0: 无匹配事件发生 1: 计数器 CNTR 内容与 CH1CCR 寄存器内容匹配 除中心对齐计数模式外，当计数器的值与 CH1CCR 的值匹配时，此位被硬件置位。 通过软件清零。 – 通道 1 配置为输入时： 0: 无输入捕捉发生 1: 输入捕捉发生 此位在捕捉事件发生时被硬件置位。通过软件或通过读取 CH1CCR 寄存器清零。
[0]	CH0CCIF	通道 0 捕捉 / 比较中断标志位 – 通道 0 配置为输出时： 0: 无匹配事件发生 1: 计数器 CNTR 内容与 CH0CCR 寄存器内容匹配 除中心对齐计数模式外，当计数器的值与 CH0CCR 的值匹配时，此位被硬件置位。 通过软件清零。 – 通道 0 配置为输入时： 0: 无输入捕捉发生 1: 输入捕捉发生 此位在捕捉事件发生时被硬件置位。通过软件或通过读取 CH0CCR 寄存器清零。

定时器计数器寄存器 – CNTR

该寄存器存储了定时器计数器的值。

偏移量: 0x080  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	CNTV							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	CNTV							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	CNTV	计数器的值

定时器预分频器寄存器 – PSCR

该寄存器定义了定时器预分频器的值以产生计数器时钟。

偏移量: 0x084  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	PSCV							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PSCV							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PSCV	预分频器的值 这些位用来定义预分频器的值以产生计数器的时钟频率 $f_{CK\_CNT}$ 。 $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}$ , $f_{CK\_PSC}$ 代表预分频器输入时钟源



定时器计数器重载寄存器 – CRR

该寄存器定义了定时器计数器重载值。

偏移量: 0x088  
复位值: 0x0000\_FFFF

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	CRV							
类型 / 复位	RW	1	RW	1	RW	1	RW	1
	7	6	5	4	3	2	1	0
	CRV							
类型 / 复位	RW	1	RW	1	RW	1	RW	1

位	字段	描述
[15:0]	CRV	计数器重载值 CRV 是加载到实际计数器寄存器中的重载值。

## 定时器重复寄存器 – REPR

该寄存器定义了定时器重复计数器的值。

偏移量: 0x08C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	REPV							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[7:0]	REPV	重复计数器的值 这些位允许用户定义比较寄存器的更新速率。

## 通道 0 捕捉 / 比较寄存器 – CH0CCR

该寄存器定义了定时器通道 0 捕捉 / 比较值。

偏移量: 0x090

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH0CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH0CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH0CCV	通道 0 捕捉 / 比较值 – 当通道 0 配置为输出时 CH0CCR 值和计数器的值相比较，比较结果用于触发 CH0OREF 输出信号。 – 当通道 0 配置为输入时 CH0CCR 寄存器存储由上一次通道 0 捕捉事件捕捉到的计数器值。

## 通道 1 捕捉 / 比较寄存器 – CH1CCR

该寄存器定义了定时器通道 1 捕捉 / 比较值。

偏移量: 0x094

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH1CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH1CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH1CCV	通道 1 捕捉 / 比较值 – 当通道 1 配置为输出时 CH1CCR 值和计数器的值相比较，比较结果用于触发 CH1OREF 输出信号。 – 当通道 1 配置为输入时 CH1CCR 寄存器存储由上一次通道 1 捕捉事件捕捉到的计数器值。

## 通道 2 捕捉 / 比较寄存器 – CH2CCR

该寄存器定义了定时器通道 2 捕捉 / 比较值。

偏移量: 0x098

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH2CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH2CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH2CCV	通道 2 捕捉 / 比较值 – 当通道 2 配置为输出时 CH2CCR 值和计数器的值相比较，比较结果用于触发 CH2OREF 输出信号。 – 当通道 2 配置为输入时 CH2CCR 寄存器存储由上一次通道 2 捕捉事件捕捉到的计数器值。

通道 3 捕捉 / 比较寄存器 – CH3CCR

该寄存器定义了定时器通道 3 捕捉 / 比较值。

偏移量： 0x09C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH3CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH3CCV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH3CCV	通道 3 捕捉 / 比较值 – 当通道 3 配置为输出时 CH3CCR 值和计数器的值相比较，比较结果用于触发 CH3OREF 输出信号。 – 当通道 3 配置为输入时 CH3CCR 寄存器存储由上一次通道 3 捕捉事件捕捉到的计数器值。

通道 0 非对称比较寄存器 – CH0ACR

该寄存器定义了定时器通道 0 非对称比较值。

偏移量： 0x0A0  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH0ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH0ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH0ACV	通道 0 非对称比较值 当通道 0 被配置为非对称 PWM 模式且计数器向下计数时，写入该寄存器的值将与计数器的值进行比较。

## 通道 1 非对称比较寄存器 – CH1ACR

该寄存器定义了定时器通道 1 非对称比较值。

偏移量: 0x0A4

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH1ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH1ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH1ACV	通道 1 非对称比较值 当通道 1 被配置为非对称 PWM 模式且计数器向下计数时，写入该寄存器的值将与计数器的值进行比较。



通道 2 非对称比较寄存器 – CH2ACR

该寄存器定义了定时器通道 2 非对称比较值。

偏移量： 0x0A8  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH2ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH2ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH2ACV	通道 2 非对称比较值 当通道 2 被配置为非对称 PWM 模式且计数器向下计数时，写入该寄存器的值将与计数器的值进行比较。

通道 3 非对称比较寄存器 – CH3ACR

该寄存器定义了定时器通道 3 非对称比较值。

偏移量： 0x0AC

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	CH3ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CH3ACV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CH3ACV	通道 3 非对称比较值 当通道 3 被配置为非对称 PWM 模式且计数器向下计数时，写入该寄存器的值将与计数器的值进行比较。

# 17 实时时钟 (RTC)

## 简介

实时时钟 (RTC) 电路包括 APB 接口、一个 32-bit 向上计数器、一个控制寄存器、一个预分频器、一个比较寄存器和一个状态寄存器。除了 APB 接口, 大多数的 RTC 电路位于备份域 ( 如下图的阴影部分所示 )。APB 接口位于  $V_{DD15}$  域。因此, 当  $V_{DD15}$  域掉电, 即单片机进入暂停模式时, 隔离来自电源控制单元的 ISO 信号, 是很有必要的。RTC 计数器用作系统从暂停模式恢复的一个唤醒定时器。详细的 RTC 功能会在以下内容中描述。

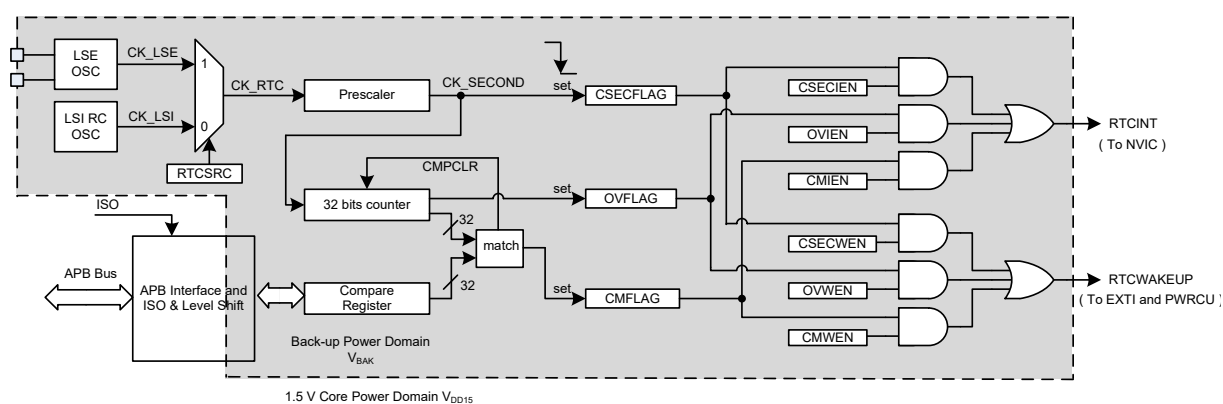


图 123. RTC 方框图

## 特性

- 32-bit 向上计数器用于计算所用的时间
- 可编程时钟预分频器
  - 分频系数: 1, 2, 4, 8, ..., 32768
- 32-bit 比较寄存器用于报警功能
- RTC 时钟源
  - LSE 振荡器时钟
  - LSI 振荡器时钟
- 三个 RTC 中断 / 唤醒设置
  - RTC 秒时钟中断 / 唤醒
  - RTC 比较匹配中断 / 唤醒
  - RTC 计数器溢出中断 / 唤醒
- RTC 中断 / 唤醒事件可搭配电源管理使用使单片机从省电模式中唤醒

功能描述

RTC 相关寄存器复位

RTC 寄存器只能由备份域上电复位, PORB, 或通过设置 BAKCR 寄存器中的 BAKRST 位使备份域软件复位的方式来复位。其它复位事件对清除 RTC 寄存器没有影响。

读取 RTC 寄存器

RTC 控制逻辑电路和相关的寄存器由 V<sub>BAK</sub> 电源电压供电。因此, RTC 电路在 V<sub>DD15</sub> 电源关闭的暂停模式下仍然运行。V<sub>DD15</sub> 电源关闭时, 只有位于 V<sub>DD15</sub> 域的 APB 总线, 通过电平转换电路与位于 V<sub>BAK</sub> 域的电路相连接并通过 ISO 信号隔离。使用 APB 总线访问 RTC 寄存器之前, 隔离功能必须通过 LPCR 寄存器中的 BKISO 位置 1 来除能, LPCR 寄存器在时钟控制单元中有所介绍。

低速时钟配置

默认的 RTC 时钟源, CK\_RTC, 来自于 LSI 振荡器。CK\_RTC 时钟可以来自于外部 32768 Hz 晶体振荡器 LSE 或内部 32 kHz RC 振荡器 LSI, 由 RTCCR 寄存器中的 RTCSRC 位来设置。分频器由 RPRE[3:0] 字段设置 CK\_RTC 时钟分频比, 分频范围是 2<sup>0</sup> ~ 2<sup>15</sup>。例如, 当 CK\_RTC 时钟频率是 32768 Hz 时, 设置预分频值 RPRE[3:0] 为 0x0F, 将产生一个精确的 1 Hz CK\_SECOND 时钟。LSI 和 LSE 振荡器分别由 RTCCR 寄存器中的 LSIEN 和 LSEEN 控制位使能。另外, LSE 振荡器启动模式由 RTCCR 寄存器中的 LSESM 位进行选择, 可以缩短 LSE 振荡器启动时间或降低功耗, 两者需根据特殊应用的需求来权衡。下表是不同启动模式下启动时间和功耗的一个例子, 可作为参考。

表 42. LSE 在不同启动模式下的工作电流和启动时间

启动模式	RTCCR 寄存器中 LSESM 的设置	工作电流	启动时间
正常启动	0	2.0 $\mu$ A	500 ms 以上
快速启动	1	3.5 $\mu$ A	300 ms 以下

@ V<sub>DD</sub> = 3.3 V 和 LSE 时钟 = 32768 Hz; 这些值仅供参考, 实际值取决于外部 32.768 kHz 晶体的规格。

RTC 计数器操作

RTC 提供了一个 32-bit 向上计数器, 它在 CK\_SECOND 时钟的下降沿递增并可以通过 APB 总线从 RTCCNT 寄存器中异步读取计数器的值。还提供了一个 32-bit 比较寄存器, RTCCMP, 用来存储特定值并与 RTCCNT 的内容进行比较。这一操作用于定义一个时间间隔的预设值。当 RTCCNT 内容等于 RTCCMP 寄存器的值时, RTCSR 寄存器中的匹配标志位 CMFLAG 将通过硬件置位并由 RTCIWEN 寄存器中相应的使能位决定发送一个中断或唤醒事件。当比较匹配事件发生时, 由 RTCCR 寄存器中的 CMPCLR 位决定 RTC 计数器复位为 0 或继续计数。例如, 如果将 RPRE[3:0] 设为 0x0F, RTCCMP 设为十进制值 60, CMPCLR 位置 1, 则 CMFLAG 位将每分钟被置位 1 次。此外, 当 RTC 计数器溢出时, RTCSR 寄存器中的 OVFLAG 位将被置位。对 RTCSR 寄存器的读取操作会清除包括 CSECFLAG、CMFLAG 和 OVFLAG 位在内的状态标志位。

中断和唤醒控制

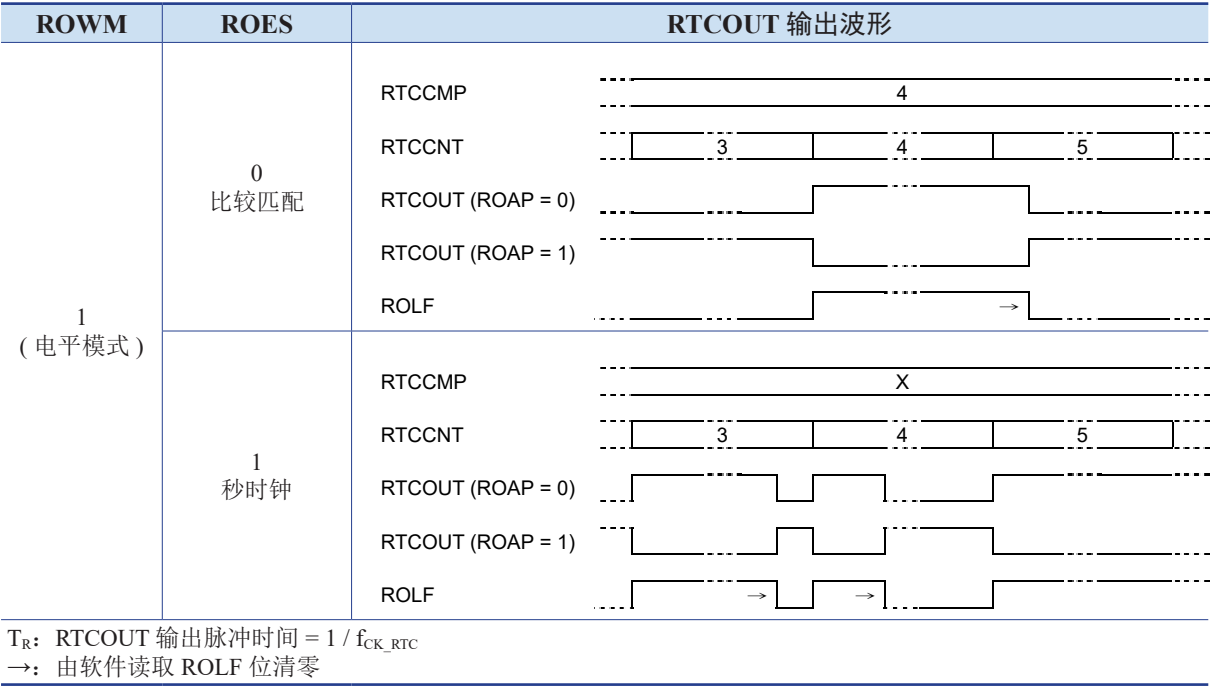
CK\_SECOND 时钟下降沿会引起 RTCSR 寄存器中的 CSECFLAG 位置位, 如果 RTCIWEN 寄存器中的相关中断使能位 CSECIEN 置位, 还会产生中断。当相应的唤醒使能位 CSECWEN 置位时, 将产生唤醒事件来唤醒 HSI / HSE 振荡器、PLL 电路、LDO 以及 CPU 内核。当 RTC 计数器溢出或比较匹配事件发生时, 将产生中断或唤醒事件, 这取决于 RTCIWEN 寄存器中相应的中断使能控制位或唤醒使能控制位, OVIEN / OVWEN 或 CMIEN / CMWEN。欲知详细信息, 请参考相关的寄存器定义。

RTCOUT 输出引脚配置

下表显示了由模式、极性和事件选择设置的 RTCOUT 的输出格式。

表 43. RTCOUT 输出模式和有效电平设置

ROWM	ROES	RTCOUT 输出波形	
0 (脉冲模式)	0 比较匹配	RTCCMP	-----4-----
		RTCCNT	-----3-----4-----5-----
		RTCOUT (ROAP = 0)	-----T <sub>R</sub> -----
		RTCOUT (ROAP = 1)	-----T <sub>R</sub> -----
		ROLF	-----
	1 秒时钟	RTCCMP	-----X-----
		RTCCNT	-----3-----4-----5-----
		RTCOUT (ROAP = 0)	-----T <sub>R</sub> -----T <sub>R</sub> -----T <sub>R</sub> -----
		RTCOUT (ROAP = 1)	-----T <sub>R</sub> -----T <sub>R</sub> -----T <sub>R</sub> -----
		ROLF	-----



寄存器列表

下表显示了 RTC 寄存器及其复位值。注意，此单元中的寄存器都位于 V<sub>BAK</sub> 备份电源域中。

表 44. RTC 寄存器列表

寄存器	偏移量	描述	复位值
RTCCNT	0x000	RTC 计数器寄存器	0x0000_0000
RTCCMP	0x004	RTC 比较寄存器	0x0000_0000
RTCCR	0x008	RTC 控制寄存器	0x0000_0F04
RTCSR	0x00C	RTC 状态寄存器	0x0000_0000
RTCIWEN	0x010	RTC 中断和唤醒使能寄存器	0x0000_0000

寄存器描述

RTC 计数器寄存器 – RTCCNT

该寄存器定义了一个按 CK\_SECOND 时钟递增的 32-bit 向上计数器。

偏移量: 0x000  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	RTCCNTV								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	RTCCNTV								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	RTCCNTV								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	RTCCNTV								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[31:0]	RTCCNTV	RTC 计数器值 读取 RTCCNT 寄存器时，将返回 RTC 计数器的当前值。在 CK_SECOND 下降沿，RTCCNT 寄存器的值将被更新。该寄存器在以下情况被复位： – 备份域软件复位 – 在 BAKCR 寄存器中设置 BAKRST 位 – 备份域上电复位 – PORB – 当 CMPCLR = 1 ( 位于 RTCCR 寄存器 ) 且比较匹配发生 (RTCCNT = RTCCMP) – RTCEN 位从 0 变为 1

RTC 比较寄存器 – RTCCMP

该寄存器定义了一个与 RTC 寄存器的值进行比较的特定值。

偏移量: 0x004  
复位值: 0x0000\_0000 ( 仅由备份域复位来复位 )

	31	30	29	28	27	26	25	24	
	RTCCMPV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	RTCCMPV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	RTCCMPV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	RTCCMPV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	RTCCMPV	RTC 比较匹配值 RTCCNT 寄存器的值与 RTCCMP 寄存器的值相等时，匹配条件发生。如果 RTCIWEN 寄存器中的 CMIEN 位被置位，则将产生中断。当 RTCCR 寄存器中的 CMPCLR 位清零且匹配条件发生时，RTCSR 寄存器中的 CMFLAG 位被置位，而 RTCCNT 寄存器的值不受影响，计数器将继续计数直到溢出为止。当 RTCCR 寄存器中的 CMPCLR 位置 1 且匹配条件发生时，RTCSR 寄存器中的 CMFLAG 位被置位，RTCCNT 寄存器的值将复位到 0 然后计数器将继续计数。



RTC 控制寄存器 – RTCCR

该寄存器定义了 RTC 电路控制位的范围。

偏移量: 0x008  
复位值: 0x0000\_0F04 ( 仅由备份域复位来复位 )

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位		ROLF		ROAP	ROWM	ROES	ROEN
类型 / 复位			RC		0 RW	0 RW	0 RW	0 RW
	15	14	13	12	11	10	9	8
	保留位				RPRE			
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位		LSESM	CMPCLR	LSEEN	LSIEN	RTCSRC	RTCEN
类型 / 复位			RW	0 RW	0 RW	0 RW	1 RW	0 RW

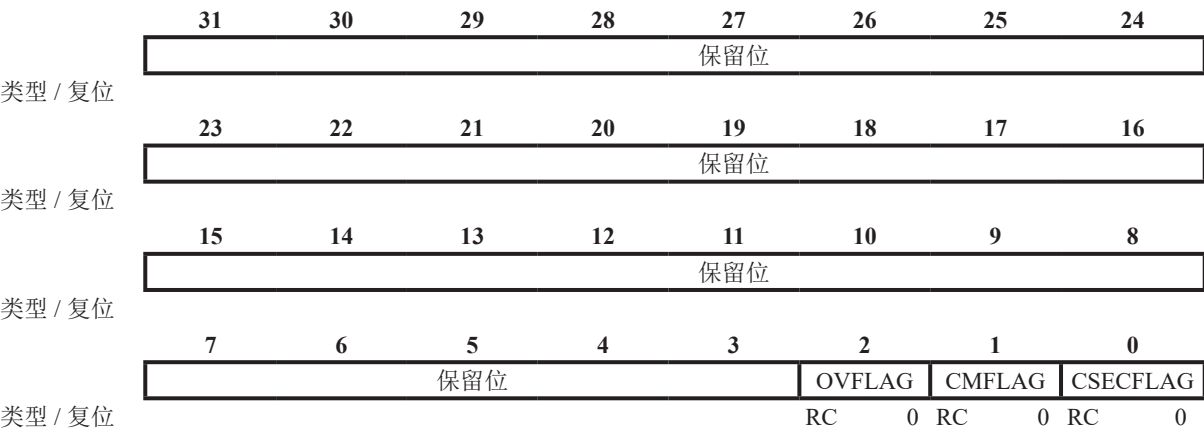
位	字段	描述
[20]	ROLF	RTCOUNT 电平模式标志位 0: RTCOUNT 输出无效 1: RTCOUNT 输出保持在有效电平 当处于电平模式 (ROWM = 1) 且一个 RTCOUNT 输出事件发生时，此位被硬件置位。 软件读取此位会使其清零。软件读取此位后，RTCOUNT 信号将回到无效电平。
[19]	ROAP	RTCOUNT 输出有效极性 0: 高电平有效 1: 低电平有效
[18]	ROWM	RTCOUNT 输出波形模式 0: 脉冲模式 输出脉冲时间是一个 RTC 时钟 (CK_RTC) 周期。 1: 电平模式 在软件读取 ROLF 位使其清零之前，RTCOUNT 信号将保持在有效电平。
[17]	ROES	RTCOUNT 输出事件选择 0: RTC 比较匹配事件被选择 1: RTC 秒时钟 (CK_SECOND) 事件被选择 此位用于选择当 RTC 比较匹配事件还是 RTC 秒时钟 (CK_SECOND) 事件发生时，RTCOUNT 引脚输出 RTCOUNT 信号。
[16]	ROEN	RTCOUNT 输出引脚使能位 0: 除能 RTCOUNT 输出引脚 1: 使能 RTCOUNT 输出引脚 当 ROEN 位设为 1 时，一旦 RTC 比较匹配或 RTC 秒时钟 (CK_SECOND) 事件发生，RTCOUNT 信号将处于一个有效电平状态。有效极性和输出波形模式可分别由 ROAP 和 ROWM 位设置。当 ROEN 位清零时，RTCOUNT 引脚将处于浮空状态。

位	字段	描述
[11:8]	RPRE	RTC 时钟预分频器选择 $CK\_SECOND = CK\_RTC / 2^{RPRE}$ 0000: $CK\_SECOND = CK\_RTC / 2^0$ 0001: $CK\_SECOND = CK\_RTC / 2^1$ 0010: $CK\_SECOND = CK\_RTC / 2^2$ ... 1111: $CK\_SECOND = CK\_RTC / 2^{15}$
[5]	LSESM	LSE 振荡器启动模式 0: 正常启动且功耗较小 1: 快速启动但工作电流较大
[4]	CMPCLR	比较匹配计数器清零 0: 比较匹配条件发生时, 对 32-bit RTC 计数器无影响 1: 比较匹配条件发生时, 32-bit RTC 计数器清零
[3]	LSEEN	LSE 振荡器使能控制位 0: LSE 振荡器除能 1: LSE 振荡器使能
[2]	LSIEN	LSI 振荡器使能控制位 0: LSI 振荡器除能 1: LSI 振荡器使能 LSIEN 位默认值是 1, 意味着 LSI 振荡器在备份域上电后自动使能。 注: 备份域上电后, 内部 LSI RC 振荡器开始振荡。LSI 的振荡频率范围在规格书中电气特性部分有所说明。单片机提供了产品的调整值, 以获得更精确的振荡频率。此过程是先除能 LSI 振荡器, 然后在备份域上电后再重新使能。调整过程完成后, 系统会自动加载产品调整值到 LSI RC 振荡器的频率调整电路中。
[1]	RTCSRC	RTC 时钟源选择 0: LSI 振荡器作为 RTC 时钟源 1: LSE 振荡器作为 RTC 时钟源
[0]	RTCEN	RTC 使能控制位 0: RTC 除能 1: RTC 使能

RTC 状态寄存器 – RTCSR

该寄存器存储了计数器的标志位。

偏移量: 0x00C  
复位值: 0x0000\_0000 (由备份域复位来复位且 RTCEN 位从 1 变为 0)



位	字段	描述
[2]	OVFLAG	计数器溢出标志位 0: 自上一次 RTCSR 寄存器读操作后无计数器溢出发生 1: 自上一次 RTCSR 寄存器读操作后计数器溢出发生 当计数器的值 RTCCNT 从 0xFFFF_FFFF 变为 0x0000_0000 时，此位将被硬件置位，并通过软件读操作清零。建议在 RTC IRQ 处理器中读取此位，在使用软件轮询时应注意。
[1]	CMFLAG	比较匹配条件标志位 0: 自上一次 RTCSR 寄存器读操作后无比较匹配条件发生 1: 自上一次 RTCSR 寄存器读操作后比较匹配条件发生 当寄存器 RTCCNT 的值与寄存器 RTCCMP 的值相等时，此位在 CK_SECOND 时钟下降沿被硬件置位。通过软件读取此位使其清零。建议在相应的 RTC 中断服务程序中对 此位访问 - 软件自由运行时，不要使用软件轮询操作。
[0]	CSECFLAG	CK_SECOND 发生标志位 0: 自上一次 RTCSR 寄存器读操作后 CK_SECOND 没有发生 1: 自上一次 RTCSR 寄存器读操作后 CK_SECOND 发生 此位在 CK_SECOND 时钟下降沿被硬件置位。通过软件读取此位使其清零。建议在相应的 RTC 中断服务程序中对 此位访问 - 软件自由运行时，不要使用软件轮询操作。

## RTC 中断和唤醒使能寄存器 – RTCIWEN

该寄存器包含了中断和唤醒使能位。

偏移量: 0x010

复位值: 0x0000\_0000 ( 仅由备份域复位来复位 )

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位					OVWEN	CMWEN	CSECWEN
类型 / 复位					RW	0	RW
7	6	5	4	3	2	1	0
保留位					OVIEN	CMIEN	CSECIEN
类型 / 复位					RW	0	RW

位	字段	描述
[10]	OVWEN	计数器溢出唤醒使能位 0: 计数器溢出唤醒除能 1: 计数器溢出唤醒使能
[9]	CMWEN	比较匹配唤醒使能位 0: 比较匹配唤醒除能 1: 比较匹配唤醒使能
[8]	CSECWEN	计数器时钟 CK_SECOND 唤醒使能位 0: 计数器时钟 CK_SECOND 唤醒除能 1: 计数器时钟 CK_SECOND 唤醒使能
[2]	OVIEN	计数器溢出中断使能位 0: 计数器溢出中断除能 1: 计数器溢出中断使能
[1]	CMIEN	比较匹配中断使能位 0: 比较匹配中断除能 1: 比较匹配中断使能
[0]	CSECIEN	计数器时钟 CK_SECOND 中断使能位 0: 计数器时钟 CK_SECOND 中断除能 1: 计数器时钟 CK_SECOND 中断使能

# 18 看门狗定时器 (WDT)

## 简介

看门狗定时器是一个硬件定时电路，可用于检测因软件陷入死锁导致的系统锁定。看门狗定时器可工作在复位模式。当计数器向下计数到零时，看门狗定时器将产生复位。因此，软件应在看门狗定时器下溢前重新加载计数器的值。此外，如果在到达 WDT 增量值之前软件重新加载计数器，也会产生复位。这意味着看门狗定时器要防止软件死锁不断地被触发，计数器必须在看门狗定时器的值为 0 ~ WDTD 有限的时间窗口内重新加载。当处理器处于调试或三种休眠模式时，看门狗定时器计数器可以停止。寄存器的写保护功能可以开启来防止看门狗定时器配置的突然改变。

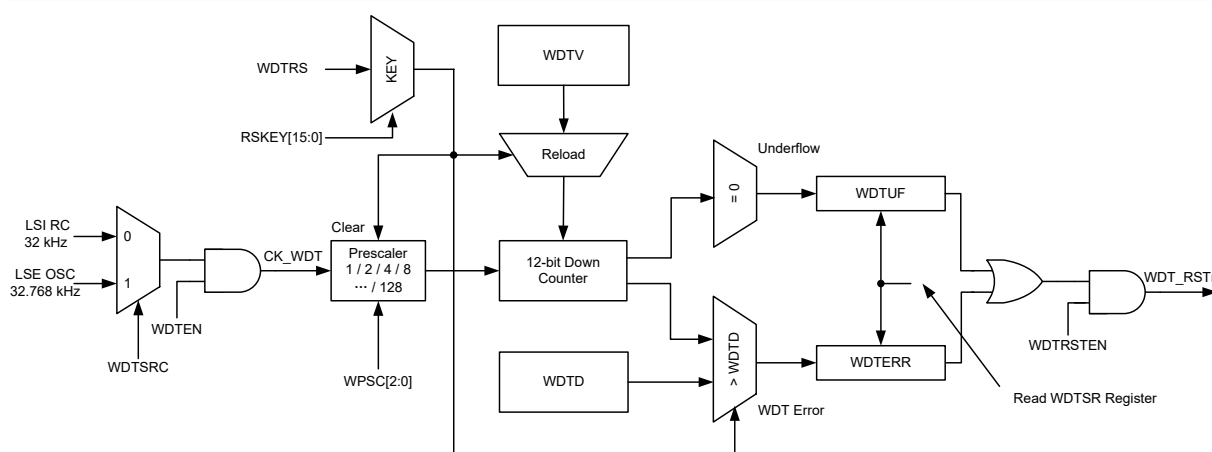


图 124. 看门狗定时器方框图

## 特性

- 时钟源来自于内部 32 kHz RC 振荡器 (LSI) 或外部 32.768 kHz 振荡器 (LSE)
- 进入休眠模式或深度休眠模式 1 时，可以独立设置为继续运行或停止
- 带有 3-bit 预分频器的 12-bit 向下计数器结构
- 为系统提供复位信号
- 有限的重载窗口设置功能，可以定制看门狗定时器的重载时间
- 当处理器在调试模式时，看门狗定时器可能会停止
- 重载锁定键可以防止意外操作
- 配置寄存器写保护功能用于保护计数器值、复位使能、增量值和预分频器

## 功能描述

看门狗定时器由一个 12-bit 向下计数器和一个固定的 3-bit 预分频器。使用 LSE 或 LSI 时钟和 1/128 最大分频值时，最长溢出周期为 16 秒。

看门狗定时器的配置包括一个可编程的计数器重载值、复位使能、窗口值和预分频值。看门狗定时器开始计数之前，这些配置都需通过 WDTMR0 和 WDTMR1 寄存器进行正确设置。为防止这些配置被写入不可预期的值，可以通过向 WDTPR 寄存器中的 PROTECT[15:0] 位写入除 0x35CA 以外的值来使能寄存器写保护功能。在访问配置寄存器之前，向 PROTECT[15:0] 位写入 0x35CA 值，可以除能寄存器写保护功能。读取 PROTECT[0] 位可以得到寄存器写保护功能的使能 / 除能状态。

正常操作期间，看门狗定时器计数器应在下溢之前被重载，以防止看门狗复位的发生。12-bit 向下计数器可通过设置 WDTCSR 寄存器中的 WDTRS 位为 1 和设置锁定键位为 0x5FA0，来载入所需看门狗定时器计数器的值 (WDTV)。

如果软件死锁发生在看门狗定时器重载程序里，重载操作仍将继续执行，因此软件死锁不会被检测到。为了避免这一情况的发生，当看门狗定时器计数器的值在增量值 (WDTD) 以内时必须执行重载操作。如果看门狗定时器计数器的值大于增量值时执行重载操作，会导致看门狗定时器发生错误，如果相关功能控制使能还将产生复位。然而，通过对 WDTD 编程，使其值大于或等于 WDTV 的值，可除能上述的功能。

当看门狗定时器下溢或看门狗定时器发生错误时，WDTSR 寄存器中的 WDTERR 标志位和 WDTUF 标志位会被置位。系统复位或对 WDTSR 寄存器的写 1 操作会清除 WDTERR 标志位和 WDTUF 标志位。

看门狗定时器使用两个时钟：PCLK 和 CK\_WDT。PCLK 时钟用于 APB 访问看门狗寄存器。CK\_WDT 时钟用于看门狗定时器功能和计数。这两个时钟域之间有一些同步逻辑关系。

当系统进入休眠模式或深度休眠模式 1 时，看门狗定时器计数器是继续计数或停止取决于 WDTMR0 寄存器中的 WDTSHLT 位。而在系统进入深度休眠模式 2 时，看门狗定时器计数器将始终停止。看门狗定时器停止计数时，计数值会被保留，使得在系统从这三种休眠模式中被唤醒后可以继续计数。当看门狗定时器有一个工作时钟源，看门狗复位会发生在看门狗定时器运行的任何时间。当系统进入调试模式时，看门狗定时器计数器是继续计数或停止取决于时钟控制单元中 MCUDBGCR 寄存器的 DBWDT 位。

看门狗定时器应按下列方式使用：

- 设置 WDTMR0 寄存器中的看门狗定时器重载值 (WDTV) 和复位选项。
- 设置 WDTMR1 寄存器中的看门狗定时器增量值 (WDTD) 和预分频器选项。
- 向 WDTCR 寄存器中写入  $WDTRS = 1$  和  $RSKEY = 0x5FA0$  来启动看门狗定时器。
- 向 WDTPR 寄存器中写入值来锁定所有的看门狗定时器寄存器，但不包括 WDTCR 和 WDTPR 寄存器。
- 看门狗定时器计数器应重新载入增量值 (WDTD) 以内的重载值。

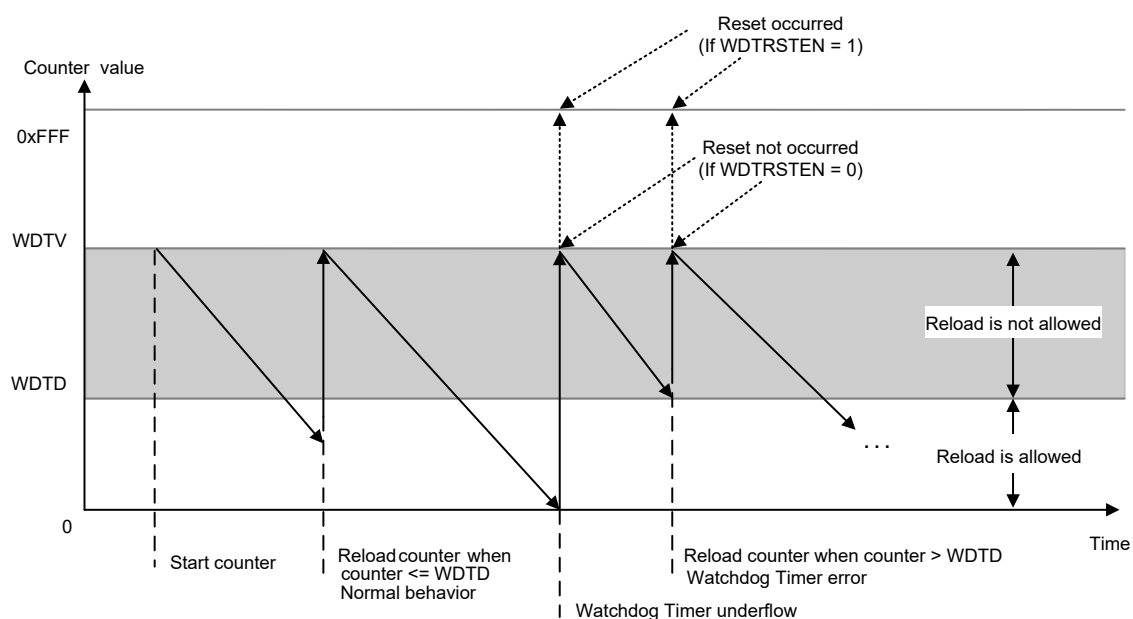


图 125. 看门狗定时器的运转

寄存器列表

下表显示了看门狗定时器寄存器及其复位值。

表 45. 看门狗定时器寄存器列表

寄存器	偏移量	描述	复位值
WDTCR	0x000	看门狗定时器控制寄存器	0x0000_0000
WDTMR0	0x004	看门狗定时器模式寄存器 0	0x0000_0FFF
WDTMR1	0x008	看门狗定时器模式寄存器 1	0x0000_7FFF
WDTSR	0x00C	看门狗定时器状态寄存器	0x0000_0000
WDTPR	0x010	看门狗定时器保护寄存器	0x0000_0000
WDTCSR	0x018	看门狗定时器时钟选择寄存器	0x0000_0000

寄存器描述

看门狗定时器控制寄存器 – WDTCR

该寄存器用来重载看门狗定时器。

偏移量: 0x000  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	RSKEY								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	RSKEY								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	保留位								
类型 / 复位									
	7	6	5	4	3	2	1	0	
	保留位							WDTRS	
类型 / 复位								WO	0

位	字段	描述
[31:16]	RSKEY	看门狗定时器重载锁定键 向 RSKEY[15:0] 位写入 0x5FA0 值来开启 WDT 重载功能，写入其它值将中止写操作。
[0]	WDTRS	看门狗定时器重载 0: 无操作 1: 重载看门狗定时器 此位用来将存储在 WDTMR0 寄存器中的 WDTV 值重载入看门狗定时器计数器。此位由软件置 1 并由硬件自动清零。



看门狗定时器模式寄存器 0 – WDTMR0

该寄存器定义了看门狗定时器计数器重载值和复位使能控制。

偏移量: 0x004  
复位值: 0x0000\_0FFF

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位							WDTEN		
类型 / 复位								RW	0	
	15	14	13	12	11	10	9	8		
	WDTSHLT		WDTRSTEN	保留位	WDTV					
类型 / 复位	RW	0	RW	0	RW	0	RW	1	RW	1
	7	6	5	4	3	2	1	0		
	WDTV									
类型 / 复位	RW	1	RW	1	RW	1	RW	1	RW	1

位	字段	描述
[16]	WDTEN	看门狗定时器运行使能位 0: 看门狗定时器除能 1: 看门狗定时器使能运行 当看门狗定时器除能, 计数器将被重置为硬件默认状态。当 WDTEN 位被置位, 看门狗定时器将重载入 WDTV 值并向下计数。
[15:14]	WDTSHLT	看门狗定时器休眠暂停 00: 看门狗定时器在系统处于休眠模式或深度休眠模式 1 时运行 01: 看门狗定时器在系统处于休眠模式时运行, 在系统处于深度休眠模式 1 时暂停 1x: 看门狗定时器在系统处于休眠模式或深度休眠模式 1 时暂停 注意, 看门狗定时器在深度休眠模式 2 时总是暂停的。适当配置 WDTSHLT 字段可使看门狗定时器在休眠模式或深度休眠模式 1 下停止计数。当看门狗定时器停止计数, 计数值会被保留以便在系统从这三种休眠模式中唤醒后继续计数。若在休眠模式或深度休眠模式 1 中发生看门狗定时器复位将唤醒单片机。
[13]	WDTRSTEN	看门狗定时器复位使能位 0: 看门狗定时器下溢或错误事件对系统复位无影响 1: 看门狗定时器下溢或错误事件触发看门狗定时器系统复位
[11:0]	WDTV	看门狗定时器计数器值 WDTV 定义了载入 12-bit 看门狗向下计数器的值。

看门狗定时器模式寄存器 1 – WDTMR1

该寄存器定义了看门狗增量值和预分频器选项。

偏移量: 0x008  
复位值: 0x0000\_7FFF

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	保留位	WPSC				WDTD			
类型 / 复位		RW	1	RW	1	RW	1	RW	1
	7	6	5	4	3	2	1	0	
	WDTD								
类型 / 复位	RW	1	RW	1	RW	1	RW	1	RW

位	字段	描述
[14:12]	WPSC	看门狗定时器预分频器选项 000: 1/1 001: 1/2 010: 1/4 011: 1/8 100: 1/16 101: 1/32 110: 1/64 111: 1/128
[11:0]	WDTD	看门狗定时器增量值 WDTD 用来定义重载到看门狗定时器中的值所允许的范围。如果看门狗定时器计数器的值小于或等于 WDTD 的值，那么通过向 WDTCSR 寄存器中写入 WDTRIS = 1 和 RSKEY = 0x5FA0 会重载定时器。如果看门狗定时器计数器的值大于 WDTD 的值，那么通过向 WDTCSR 寄存器中写入 WDTRIS = 1 和 RSKEY = 0x5FA0 会导致看门狗定时器发生错误。此功能可通过编程使 WDTD 值大于或等于 WDTV 值来除能。

看门狗定时器状态寄存器 – WDTSR

该寄存器定义了看门狗定时器的状态。

偏移量： 0x00C

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	保留位							
	7	6	5	4	3	2	1	0
类型 / 复位	保留位						WDTERR	WDTUF
							WC	0 WC 0

位	字段	描述
[1]	WDTERR	看门狗定时器错误 0: 自上次对该寄存器进行读操作后无看门狗定时器错误发生 1: 自上次对该寄存器进行读操作后看门狗定时器错误发生 注: 当看门狗定时器计数器的值大于 WDTD 值时, 重载操作会导致看门狗定时器发生错误。 需要注意的是, 该位是写 1 清零标志位。
[0]	WDTUF	看门狗定时器下溢 0: 自上次对该寄存器进行读操作后无看门狗定时器下溢发生 1: 自上次对该寄存器进行读操作后看门狗定时器下溢发生 需要注意的是, 该位是写 1 清零标志位。

看门狗定时器保护寄存器 – WDTPR

该寄存器定义了看门狗定时器保护键的配置。

偏移量: 0x010  
复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	PROTECT							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	PROTECT							
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	PROTECT	看门狗定时器寄存器保护 写操作： 0x35CA：除能看门狗定时器寄存器写保护功能 其它值：使能看门狗定时器寄存器写保护功能 读操作： 0x0000：除能看门狗定时器寄存器写保护功能 0x0001：使能看门狗定时器寄存器写保护功能 该寄存器用来除能 / 使能看门狗定时器配置寄存器的写保护功能。当寄存器写保护功能使能时，除了 WDTCSR 和 WDTPR 以外的所有配置寄存器将变成只读寄存器。此外，读取 PROTECT[0] 位可以获得寄存器写保护功能的使能 / 除能状态。

看门狗定时器时钟选择寄存器 – WDTCSR

该寄存器定义了看门狗定时器时钟选择和锁定配置。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位			WDTLOCK	保留位			WDTSRC
类型 / 复位				RW 0				RW 0

位	字段	描述
[4]	WDTLOCK	看门狗定时器锁定模式 0: 该位只能由任意复位清零，不能由软件清零 1: 该位只能由软件置位一次并锁定看门狗定时器功能 该位在任何时候都可以由软件设置为 1。一旦 WDTLOCK 位被置位，看门狗定时器的功能和寄存器都不能被修改或除能，包括看门狗定时器的时钟源，只有等到系统复位才能将锁定模式除能。
[0]	WDTSRC	看门狗定时器时钟源选择 0: 选择内部 32 kHz RC 振荡器时钟 (LSI) 1: 选择外部 32.768 kHz 晶体振荡器时钟 (LSE) 选择使用软件来控制看门狗定时器的时钟源。

## 19 内部集成电路 (I<sup>2</sup>C)

### 简介

I<sup>2</sup>C 模块是一个允许与外部 I<sup>2</sup>C 接口通信的内部电路，而外部 I<sup>2</sup>C 接口是一个符合工业标准并用于连接外部硬件的两线串行接口。这两条串行线被称为串行数据线 SDA 和串行时钟线 SCL。I<sup>2</sup>C 模块提供了三种数据传输速率：标准模式下 100 kHz、快速模式下 400 kHz 和高速模式下 1 MHz。SCL 周期产生寄存器用于设置不同的占空比以得到不同的 SCL 脉冲。

SDA 线是一条双向数据线，它连接整个 I<sup>2</sup>C 总线，在主机和从机之间用于数据的发送和接收。I<sup>2</sup>C 模块还具有仲裁检测功能，以防止多个主机试图同时传送数据到 I<sup>2</sup>C 总线的情况。

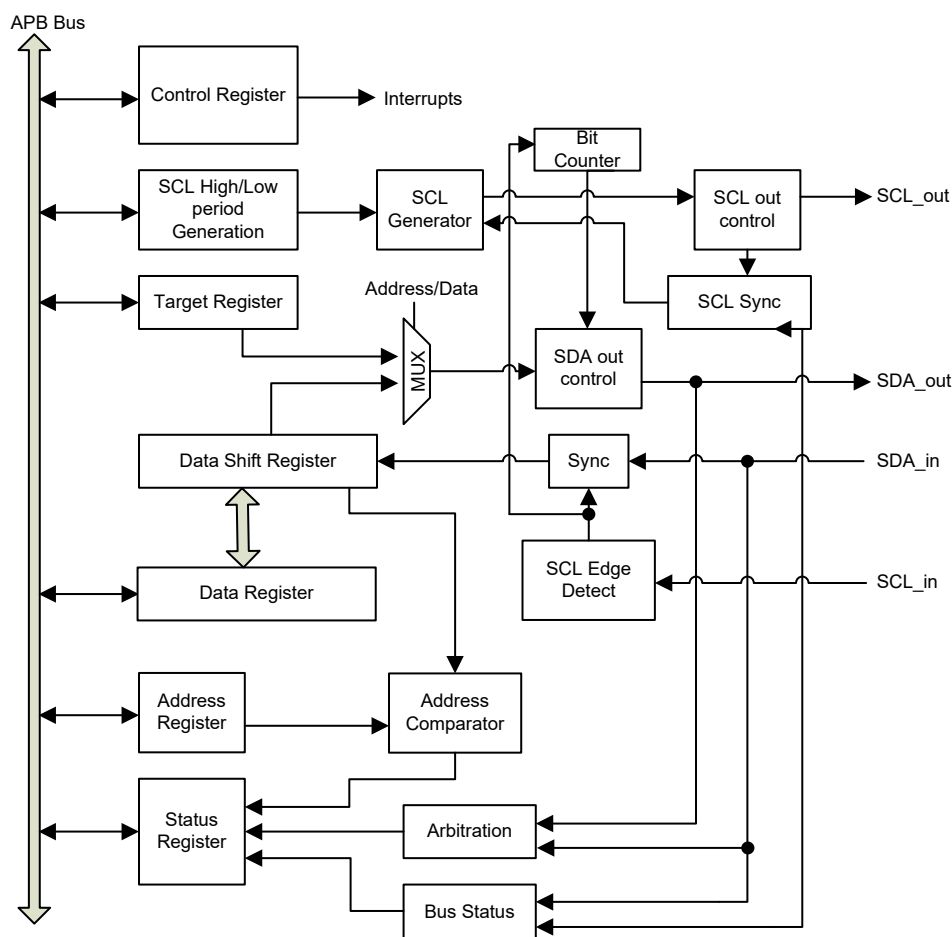


图 126. I<sup>2</sup>C 模块方框图

## 特性

- 双线 I<sup>2</sup>C 串行接口
  - 串行数据线 (SDA) 和串行时钟线 (SCL)
- 多种速率模式
  - 标准模式 – 100 kHz
  - 快速模式 – 400 kHz
  - 高速模式 – 1 MHz
- 主机和从机之间的双向数据传输
- 多主机总线 – 无中心主机
  - 相同的接口可作为主机或从机
- 多主机同时传输的仲裁功能使其在总线上不会产生串行数据损坏
- 时钟同步
  - 允许不同比特率的设备通过一条串行总线进行通信
- 支持 7-bit 和 10-bit 寻址模式以及广播呼叫寻址
- 地址屏蔽功能，可支持多个从机地址寻址
- 超时功能
- 支持 PDMA 接口

## 功能描述

### 双线串行接口

I<sup>2</sup>C 模块有两条外部线，串行数据线 SDA 和串行时钟线 SCL，用来承载连接到总线的设备之间的信息。SCL 和 SDA 线是双向的，且必须连接一个上拉电阻。当 I<sup>2</sup>C 总线处于自由或空闲状态时，两个引脚都处于高电平状态，执行用于多个互连设备所要求的线与功能。

### START 和 STOP 条件

主机可以通过发送一个 START 信号启动传输并通过发送一个 STOP 信号来终止传输。START 信号通常被称为“S”位，定义为当 SCL 线为高电平时，SDA 线上发生从高到低的电平变化。STOP 信号通常被称为“P”位，定义为当 SCL 线为高电平时，SDA 线上从低到高的电平变化。

重复的 START，称为“Sr”位，功能上与 START 条件相同。一个重复的 START 信号允许 I<sup>2</sup>C 接口在不释放总线控制权的情况下与另一从机进行通信或者与同一 I<sup>2</sup>C 设备在不同传输方向上进行通信。

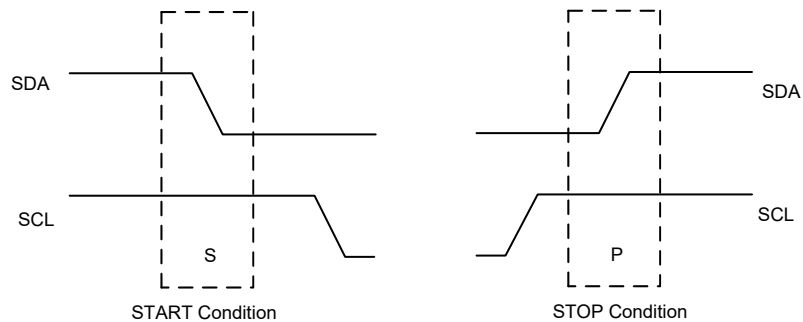


图 127. START 和 STOP 条件

### 数据有效性

在 SCL 时钟的高电平期间，SDA 线上的数据必须保持稳定。只有当 SCL 线上的时钟信号处于低电平状态时，SDA 数据状态才能改变。

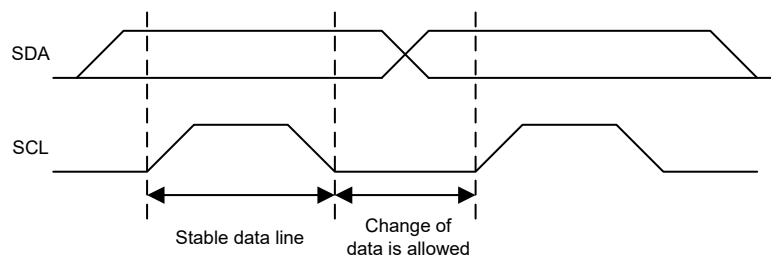


图 128. 数据有效性



## 寻址格式

主机发送地址确认目标从机之后，I<sup>2</sup>C 接口开始传输数据。地址帧在主机发送 START 信号后被发出。I2CCR 寄存器中的寻址模式选择位 ADRM 应该被设置来选择 7-bit 或 10-bit 寻址模式。

### 7-bit 地址格式

7-bit 地址格式由以下各部分组成：与主机通信的 7-bit 长度的从机地址，一个 R/ $\overline{W}$  位和一个 ACK 位。R/ $\overline{W}$  位定义了数据传输方向。

R/ $\overline{W}$  = 0 (写)：主机发送数据到被寻址的从机。

R/ $\overline{W}$  = 1 (读)：主机从被寻址的从机接收数据。

从机地址可通过 I2CADDR 寄存器中的 ADDR 字段分配。如果从机地址与主机发出的地址相匹配，从机将会返回一个确认位 (ACK)。

注意，不允许两个从机有相同的地址。

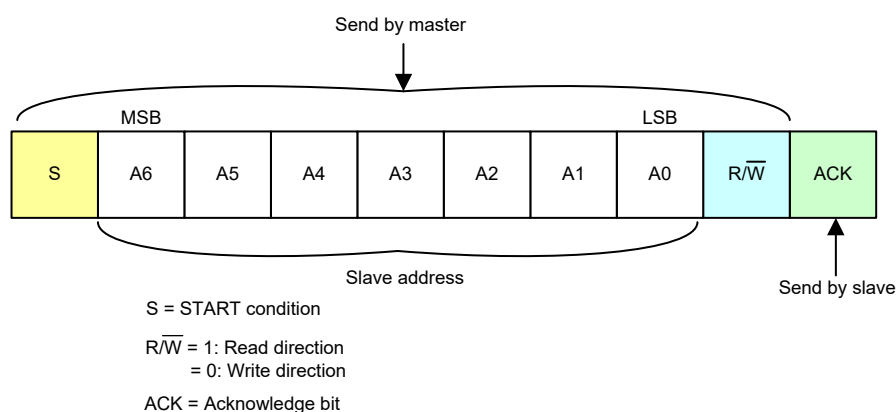


图 129. 7-bit 寻址模式

10-bit 地址格式

因 7-bit 地址范围有限, 为了防止地址冲突, 这里介绍一个 10-bit 地址格式的新方案。该改进的方案与 7-bit 寻址模式一起使用, 增加了十倍左右的可用地址范围。对于 10-bit 寻址模式, START 信号后的前两个字节包括一个头字节和一个地址字节, 用来决定主机会选择哪个从机。头字节由先头的“11110”和从机地址的第 10 位和第 9 位组成。第二个字节由从机的其余 8 位地址组成。

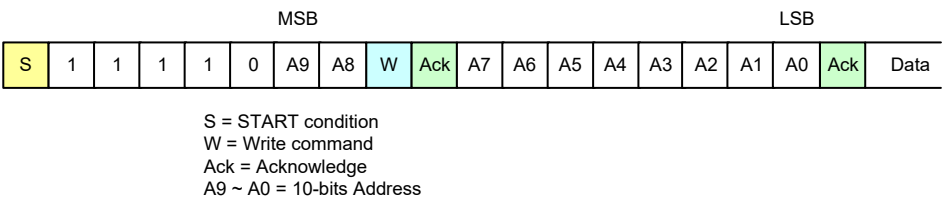


图 130. 10-bit 寻址写发送模式

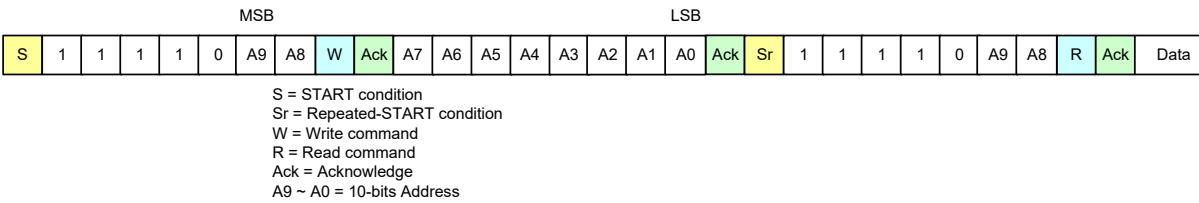


图 131. 10-bit 寻址读接收模式

## 数据传输和确认

一旦从机地址匹配,可以根据由  $R/\bar{W}$  位定义的传输方向,由从机发送或接收数据。每一个字节后面都在第 9 个 SCL 时钟上跟随一个确认位。

如果从机返回一个未确认信号 (NACK) 给主机,则主机会产生一个 STOP 信号来终止数据传输或产生一个重复 START 信号重新开始传输。

如果主机发送一个未确认信号 (NACK) 给从机,从机将释放 SDA 线给主机,主机将产生一个 STOP 信号来终止传输。

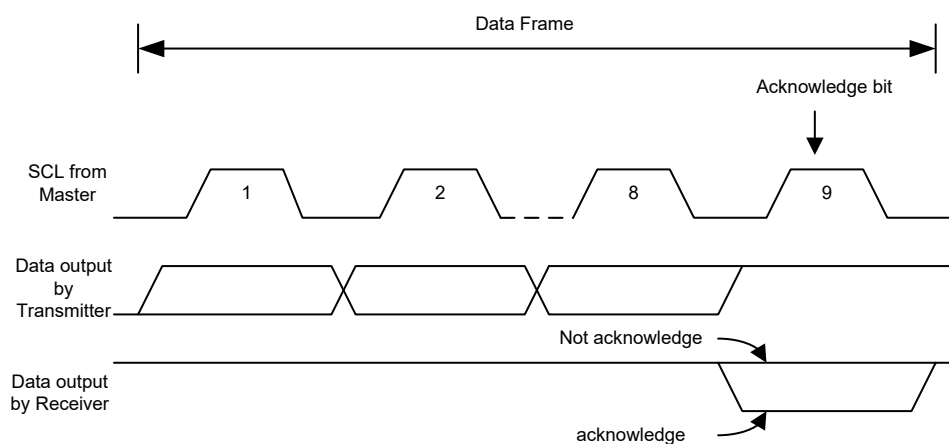


图 132. I<sup>2</sup>C 总线确认

## 时钟同步

正常工作时, 只有一个主机可以产生 SCL 时钟。然而, 当有多个主机试图产生 SCL 时钟, 时钟应该同步以便数据输出可以比较。时钟同步可通过 I<sup>2</sup>C 接口到 SCL 线的线与连接来执行。

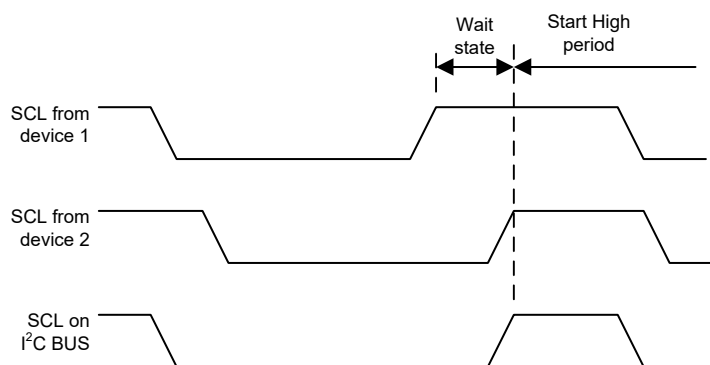


图 133. 仲裁期间时钟同步

## 仲裁

只有在 I<sup>2</sup>C 总线处于自由或空闲状态时, 主机才能开始传输。如果两个或两个以上的主机几乎同时产生 START 信号, 仲裁程序将会发生。

仲裁发生在 SDA 线上且可以持续很多位。仲裁程序赋予发送二进制低位 (逻辑低) 串行数据的设备更高的优先级。其它想要发送二进制高位 (逻辑高) 数据的主机将失去仲裁。只要一个主机失去仲裁, I<sup>2</sup>C 模块就会将 I2CSR 寄存器中的 ARBLOS 位置位, 并且在 I2CIER 寄存器中的中断使能位 ARBLOSIE 被置为 1 时产生中断。同时, 它会停止发送数据, 并监听总线, 以检测 I<sup>2</sup>C 停止信号。当停止信号被检测到, 输掉仲裁的主机会再次尝试访问总线。

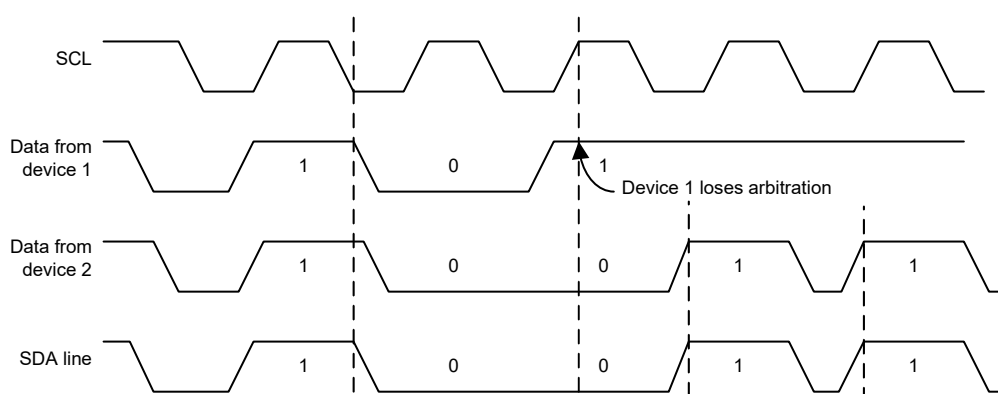


图 134. 两个主机仲裁程序

## 广播呼叫寻址

广播呼叫寻址功能可以用来寻址所有连接到 I<sup>2</sup>C 总线的设备。在寻址帧上，主机可通过设置 TAR 字段值为零且清零 RWD 位来激活广播呼叫寻址功能。

设备可通过设置相应的使能控制位 GCEN 为 1，以使其支持广播呼叫寻址功能。如果 GCEN 位已经置 1 以支持广播呼叫寻址功能，I2CCR 寄存器中的 AA 位也应置 1，当设备接收到值为 00H 的地址帧时将返回一个确认信号。当此条件发生时，广播呼叫标志位 GCS 将被置 1，但 ADRS 标志位不会被置位。

## 总线错误

如果数据正在 I<sup>2</sup>C 总线上传输时，出现意外的 START 或 STOP 条件，将会视作总线错误，数据传输将被中止。当总线错误事件发生，I2CSR 寄存器中相应的总线错误标志位 BUSERR 将被置为 1，且 SDA 和 SCL 线都将被释放。BUSERR 标志位应通过写 1 使其清零以使 I<sup>2</sup>C 模块进入空闲状态。

## 地址屏蔽使能

I<sup>2</sup>C 模块提供给用户地址屏蔽的功能，用来决定在和主机发送来的地址帧作比较时可忽略地址中的哪一位。当未屏蔽地址位和主机发送来的地址帧匹配时，ADRS 标志位被置位。注意，此功能仅在从机模式下可用。

例如，用户将数据传输设置为 7-bit 寻址模式，并设置 I2CADDRMR 寄存器的值为 0x05 和 I2CADDR 寄存器的值为 0x55，这意味着如果总线上 I<sup>2</sup>C 主机发送的地址等于 0x50、0x51、0x54 或 0x55，该 I<sup>2</sup>C 从机地址都将被认为是匹配的，在地址帧匹配之后，I2CSR 寄存器的 ADRS 标志位将被置位。

## 地址捕获

地址捕获寄存器 I2CADDRSR 用于监控整个数据传输期间 I<sup>2</sup>C 总线上的呼叫地址，不论 I<sup>2</sup>C 模块是作为主机还是从机。注意，I2CADDRSR 寄存器是只读寄存器，每个 I<sup>2</sup>C 总线上的呼叫地址自动存储在 I2CADDRSR 寄存器，即使 I<sup>2</sup>C 设备没被寻址。

## 工作模式

I<sup>2</sup>C 模块可工作在以下模式：

- 主机发送
- 主机接收
- 从机发送
- 从机接收

I<sup>2</sup>C 模块默认工作在从机模式。产生 START 信号后接口将自动切换到主机模式。

主机发送模式

开始条件

用户在设置 I2CCR 寄存器中的 I2CEN 位之后, 向 I2CTAR 寄存器中写入目标从机地址和通信方向。开始条件发生之后, I2CSR 寄存器中的 STA 标志位由硬件置位。为了发送下列地址帧, 如果 STA 标志位已被设为 1, 则必须使其清零。STA 标志位通过读取 I2CSR 寄存器来清零。

地址帧

主机发送地址帧且接收到来自于地址匹配的从机的确认信号后 I2CSR 寄存器中的 ADRS 标志位被置位。为了发送下列数据帧, 如果 ADRS 标志位已被设为 1, 则必须使其清零。ADRS 标志位通过读取 I2CSR 寄存器来清零。

数据帧

要发送到从机的数据必须被传送到 I2CDR 寄存器中。

I2CSR 寄存器中的 TXDE 位被置位, 表明 I2CDR 寄存器是空的, 这会使 SCL 线保持在逻辑低电平状态。新数据必须被传送到 I2CDR 寄存器以继续数据的传输。写入数据到 I2CDR 寄存器将清除 TXDE 标志位。

结束 / 继续发送

最后的数据字节发送完成后, I2CCR 寄存器中的 STOP 位被置位来结束发送或通过配置 I2CTAR 寄存器重新分配另外一个从机以开始新的传输。

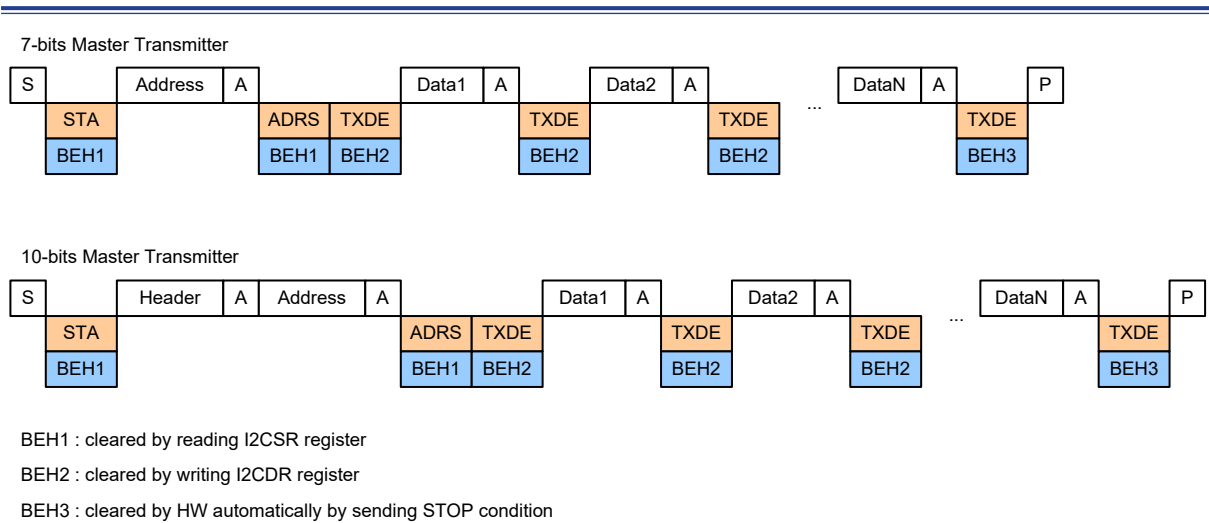


图 135. 主机发送时序图

## 主机接收模式

### 开始条件

目标从机地址和通信方向必须被写入 I2CTAR 寄存器。在开始条件产生之后, I2CSR 寄存器中的 STA 位被硬件置位。为了发送下列地址帧, 如果 STA 标志位已被设为 1, 则必须使其清零。STA 标志位通过读取 I2CSR 寄存器来清零。

### 地址帧

**7-bit 寻址模式:** 主机发送地址帧且接收到来自于地址匹配的从机的确认信号后 I2CSR 寄存器中的 ADRS 标志位被置位。为了接收下列数据帧, 如果 ADRS 标志位已被设为 1, 则必须使其清零。ADRS 标志位通过读取 I2CSR 寄存器来清零。

**10-bit 寻址模式:** 在此模式下, I2CSR 寄存器中的 ADRS 标志位会被置位两次。第一次是当 10-bit 地址被发送且接收到来自于从机的确认信号时, ADRS 位被置位。第二次是当头字节被发送且从机确认信号被接收时, ADRS 位被置位。为了接收下列数据帧, 如果 ADRS 标志位已被设为 1, 则必须使其清零。ADRS 标志位在读取 I2CSR 寄存器之后清零。详细的主机接收器模式时序图如下图所示。

### 数据帧

在主机接收模式, 数据由从机发送。一旦数据被主机接收, I2CSR 寄存器中的 RXDNE 标志位会被置位, 但不会保持 SCL 线的状态。然而, 如果设备接收一个完整的新数据字节且 RXDNE 位已被置 1, 则 I2CSR 寄存器中的 RXBF 位将被置 1 且 SCL 线会保持在一个逻辑低的状态。当此情况发生时, 应读取 I2CDR 寄存器的数据以继续数据的传输。读取 I2CDR 寄存器后, RXDNE 标志位被清零。

结束 / 继续发送

在最后一个数据字节传送完成前，主机应使 I2CCR 寄存器中的 AA 位复位来发送一个 NACK 信号到从机。最后一个来自从机的数据字节被接收后，一旦主机发送完一个 NACK 信号给从机，主机将保持 SCL 线在逻辑低的状态。STOP 位被置位来结束数据传送或重新配置 I2CTAR 寄存器以开始新的传输。

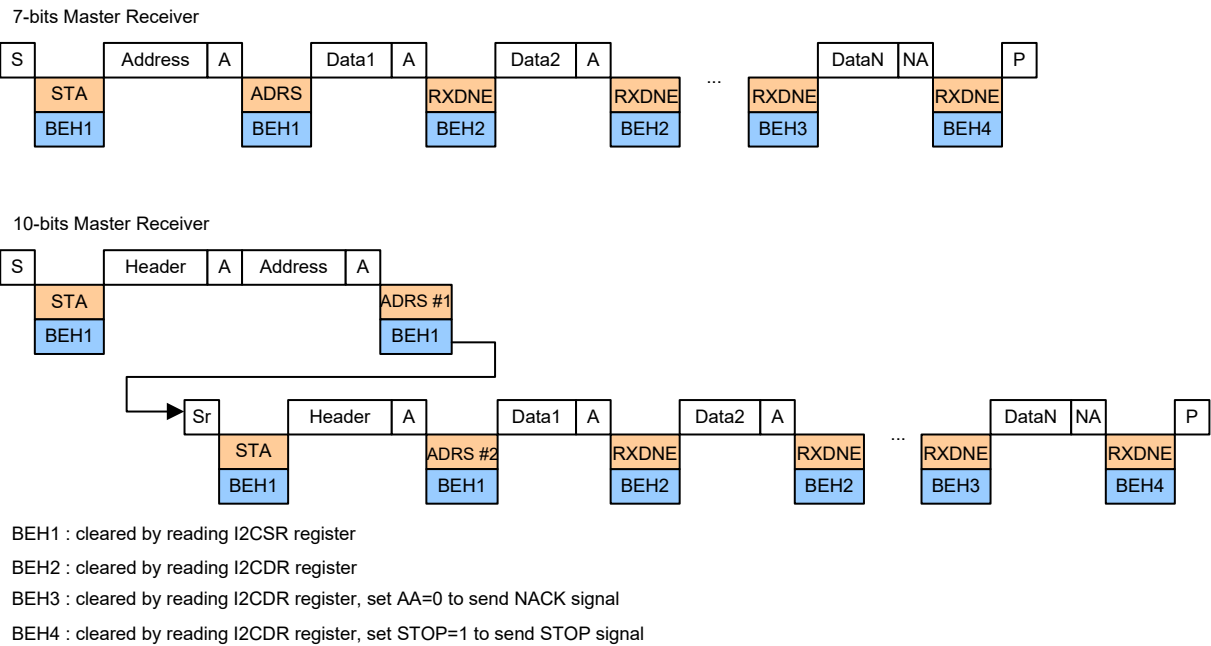


图 136. 主机接收时序图



从机发送模式

地址帧

7-bit 寻址模式，在从机接收到与从机地址相匹配的呼叫地址后，I2CSR 寄存器中的 ADRS 位被置位。10-bit 寻址模式，当第一个头字节和第二个地址字节都匹配时，ADRS 位第一次被置位。应注意当第二个头字节也匹配的时候，ADRS 位将再次被置位。在 ADRS 位已被置 1 后必须清零此位来继续数据的传输。读取 I2CSR 寄存器后，ADRS 位被清零。

数据帧

在从机发送模式，TXDE 位被置位表明 I2CDR 是空的，这将导致 SCL 线处于一个逻辑低的状态。此时新的发送数据必须写入到 I2CDR 寄存器以继续数据的传输。向 I2CDR 中写入一个数据将使 TXDE 位清零。

接收未确认信号

当从机接收到一个未确认信号时，I2CSR 寄存器中的 RXNACK 位被置位，但不会保持 SCL 线的状态。向 RXNACK 写入 1 会使 RXNACK 标志位清零。

STOP 条件

当从机检测到一个 STOP 条件时，I2CSR 寄存器中的 STO 位会被置位以表明 I<sup>2</sup>C 接口发送结束。读取 I2CSR 寄存器可以使 STO 标志位清零。

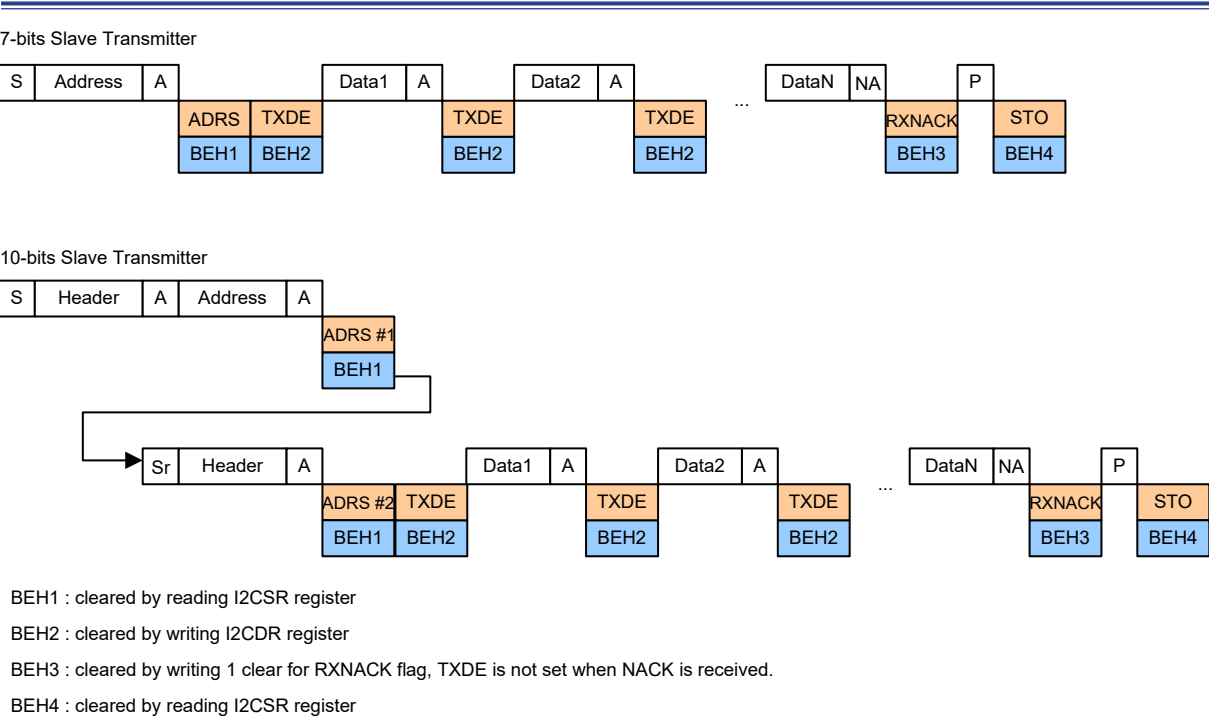


图 137. 从机发送时序图

从机接收模式

地址帧

在从机接收到与从机地址相匹配的呼叫地址后，I2CSR 寄存器中的 ADRS 位被置位。在 ADRS 位已被置 1 后必须使其清零来继续数据的传输。读取 I2CSR 寄存器之后，ADRS 标志位被清零。

数据帧

在从机接收模式，数据由主机发送。一旦一个数据字节被从机接收，I2CSR 寄存器中的 RXDNE 标志位被置位，但不会保持 SCL 线的状态。然而，如果设备接收一个完整的新数据字节且 RXDNE 位被置 1，则 I2CSR 寄存器中的 RXBF 位将被置 1 且 SCL 线会保持在一个逻辑低的状态。当此情况发生时，应读取 I2CDR 寄存器中的数据来继续数据的传输。读取 I2CDR 寄存器之后，RXDNE 标志位被清零。

STOP 条件

当从机检测到 STOP 条件时，I2CSR 寄存器中的 STO 标志位会被置位以表明 I<sup>2</sup>C 接口发送结束。读取 I2CSR 寄存器可以使 STO 位清零。

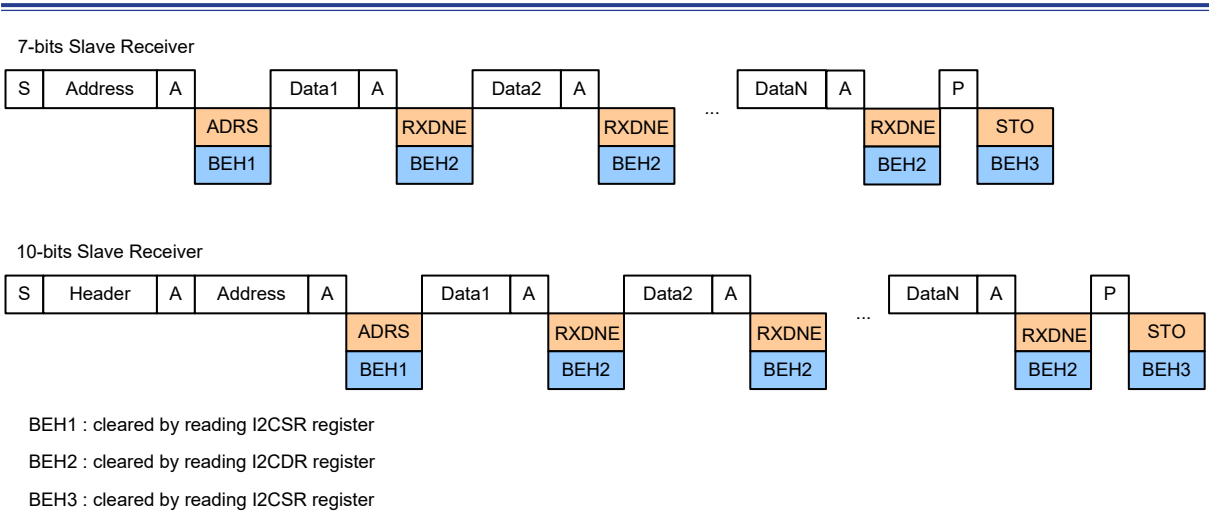


图 138. 从机接收时序图

保持 SCL 线状态的条件

下列条件将使 SCL 线由硬件保持在一个逻辑低电平状态，从而使得所有的 I<sup>2</sup>C 传输停止。数据传输将在下列条件消除后继续进行。

表 46. 保持 SCL 线状态的条件

类型	条件	描述	条件消除后
标志位	TXDE	I <sup>2</sup> C 模块用在发送模式，I2CDR 寄存器需要发送数据。 (注：接收到 NACK 信号后，TXDE 位将不会被置位。)	主机情况： 写数据到 I2CDR 寄存器 设置 TAR 设置 STOP  从机情况： 写数据到 I2CDR 寄存器
	GCS	I <sup>2</sup> C 模块作为从机通过广播呼叫被寻址。	读取 I2CSR 寄存器
	ADRS	主机： I <sup>2</sup> C 模块发送地址帧并接收从从机返回的一个 ACK 信号 (注：请参考图 135 和图 136)  从机： I <sup>2</sup> C 模块作为从机被寻址 (注：请参考图 137 和图 138)	读取 I2CSR 寄存器
	STA	主机发送 START 信号	读取 I2CSR 寄存器
	RXBF	设备接收完整的新数据，同时 RXDNE 标志位已被置位。	读取 I2CDR 寄存器
事件	主机接收 NACK	无论是地址还是数据帧，一旦接收到 NACK 信号，SCL 线将保持在主机模式。	设置 TAR 设置 STOP
	主机在接收模式发送 NACK	在主机接收模式接收最后一个数据字节时发生。 (注：请参考图 136。此情况下，RXNACK 标志将不会被置位。)	设置 TAR 设置 STOP

I<sup>2</sup>C 超时功能

超时功能可减少由于接收错误时钟源造成 I<sup>2</sup>C 被锁住的问题。如果在超时周期里没有接收到 I<sup>2</sup>C 总线时钟源，相关 I<sup>2</sup>C 超时标志位将被置位。超时周期是由一个带有可编程预载值的 16-bit 向下计数型计数器决定的。该超时计数器由 I<sup>2</sup>C 超时时钟 f<sub>I2CTO</sub> 驱动，时钟频率是由 I2CTOUT 寄存器的超时预分频器字段定义的。I2CTOUT 寄存器的 TOUT 字段用于定义超时计数器预载值。通过设置 I2CCR 寄存器的 ENTOUT 位来使能超时功能。当 ENTOUT 位被置为 1 且以下其中一种情况发生时，超时计数器将从预载值开始向下计数：

- I<sup>2</sup>C 主机模块发送 START 信号。
- I<sup>2</sup>C 从机模块检测到 START 信号。
- RXBF、TXDE、RXDNE、RXNACK、GCS 或 ADRS 标志位被置位。

ENTOUT 位清零时，超时计数器将停止计数。然而，当下列所示情况发生时，计数器也会停止计数：

- I<sup>2</sup>C 从机模块没被寻址。
- I<sup>2</sup>C 从机模块检测到 STOP 信号。
- I<sup>2</sup>C 主机模块发送 STOP 信号。
- I2CSR 寄存器的 ARBLOS 或 BUSERR 标志位被置位。

如果超时计数器下溢，I2CSR 寄存器的相关超时标志位 TOUTF 将会被置为 1，若相关中断

使能将产生超时中断。

## PDMA 接口

PDMA 接口集成于 I<sup>2</sup>C 模块。在发送器或接收器模式下可分别通过设置 TXDMAE 或 RXDMAE 位为 1 来使能 PDMA 功能。在发送器模式下, 若数据寄存器为空, TXDMAE 位置 1, PDMA 功能将被激活用于把数据从某一个存储器位置移到 I<sup>2</sup>C 数据寄存器。类似地, 在接收器模式下, 若数据寄存器不为空且 RXDMAE 位置 1, PDMA 功能将被激活用于把数据从 I<sup>2</sup>C 数据寄存器移到指定的存储器位置。

当 I<sup>2</sup>C 模块工作于主机接收器模式且 PDMA 功能使能时, DMA 模式 NACK 控制位 DMANACK 用于确认 NACK 信号是否发送。如果 DMANACK 位被置 1, 并且数据已全部通过 PDMA 接口被接收和移动, NACK 信号将自动发出以正确终止数据传输。

PDMA 配置模式的详细描述, 请参考 PDMA 章节。

## 寄存器列表

下表所示为 I<sup>2</sup>C 寄存器及其复位值。

表 47. I<sup>2</sup>C 寄存器列表

寄存器	偏移量	描述	复位值
I2CCR	0x000	I <sup>2</sup> C 控制寄存器	0x0000_2000
I2CIER	0x004	I <sup>2</sup> C 中断使能寄存器	0x0000_0000
I2CADDR	0x008	I <sup>2</sup> C 地址寄存器	0x0000_0000
I2CSR	0x00C	I <sup>2</sup> C 状态寄存器	0x0000_0000
I2CSHPGR	0x010	I <sup>2</sup> C SCL 高电平周期发生寄存器	0x0000_0000
I2CSLPGR	0x014	I <sup>2</sup> C SCL 低电平周期发生寄存器	0x0000_0000
I2CDR	0x018	I <sup>2</sup> C 数据寄存器	0x0000_0000
I2CTAR	0x01C	I <sup>2</sup> C 目标寄存器	0x0000_0000
I2CADDRMR	0x020	I <sup>2</sup> C 地址屏蔽寄存器	0x0000_0000
I2CADDRSR	0x024	I <sup>2</sup> C 地址捕获寄存器	0x0000_0000
I2CTOUT	0x028	I <sup>2</sup> C 超时寄存器	0x0000_0000

寄存器描述

I<sup>2</sup>C 控制寄存器 – I2CCR

该寄存器定义了相应的 I<sup>2</sup>C 功能使能控制位。

偏移量: 0x000 (0)  
复位值: 0x0000\_2000

类型 / 复位	保留位							
	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	SEQFILTER		COMBFILTEREN		ENTOUT		保留位	
	15	14	13	12	11	10	9	8
类型 / 复位	RW	0	RW	0	RW	1	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	ADRM		保留位		I2CEN		GCEN	
	RW	0			RW	0	RW	0

位	字段	描述
[15:14]	SEQFILTER	SDA 或 SCL 输入连续滤波器配置位 00: 连续滤波器除能 01: 1 个 PCLK 干扰滤波器 1x: 2 个 PCLK 干扰滤波器 注: 该设置将影响 SCL 的频率。详细的描述在 I2CSLPGR 寄存器部分。
[13]	COMBFILTEREN	SDA 或 SCL 输入组合滤波器使能位 0: 组合滤波器除能 1: 组合滤波器使能
[12]	ENTOUT	I <sup>2</sup> C 超时功能使能控制位 0: 超时功能除能 1: 超时功能使能 该位用于使能或除能 I <sup>2</sup> C 超时功能。建议用户在超时计数器设置 ENOUT 位为 1 开始计数之前要配置好 I2CTOUT 寄存器的 PSC 和 TOUT 字段。
[10]	DMANACK	DMA 模式 NACK 控制位 0: 无操作 1: 在 DMA 模式下, I <sup>2</sup> C 主机接收器模块在接收到从机发送器的最后一个字节之后会自动发送一个 NACK 信号
[9]	RXDMAE	DMA 模式 RX 请求使能控制位 0: RX DMA 请求除能 1: RX DMA 请求使能 在接收器模式下, 如果数据寄存器不为空, RXDMAE 位被置 1, 相关 PDMA 通道将被激活用于把数据从数据寄存器移到特定的位置, 该位置由相关 PDMA 寄存器定义。

位	字段	描述
[8]	TXDMAE	DMA 模式 TX 请求使能控制位 0: TX DMA 请求除能 1: TX DMA 请求使能 在发送器模式下，如果数据寄存器不为空，TXDMAE 位被置 1，相关 PDMA 通道将被激活用于把数据从特定的位置移到数据寄存器，该位置由相关 PDMA 寄存器定义。
[7]	ADRM	寻址模式 0: 7-bit 寻址模式 1: 10-bit 寻址模式 当 I²C 主机 / 从机模块工作在 7-bit 寻址模式时，它只能发出和响应一个 7-bit 地址，反之亦然。
[3]	I2CEN	I²C 接口使能位 0: I²C 接口除能 1: I²C 接口使能
[2]	GCEN	广播呼叫使能位 0: 广播呼叫功能除能 1: 广播呼叫功能使能 当设备接收到 0x00 的呼叫地址且 GCEN 和 AA 位都被置 1，则 I²C 接口将作为一个从机且 I2CSR 寄存器中的 GCS 位将被置 1。
[1]	STOP	STOP 条件控制位 0: 无动作 1: 在主机模式下发送 STOP 条件 此位被软件置 1 来产生一个 STOP 条件，通过硬件自动清零。STOP 位只用于主机。
[0]	AA	确认位 0: 在一个字节接收后发送一个未确认信号 (NACK) 1: 在一个字节接收后发送一个确认信号 (ACK)

I<sup>2</sup>C 中断使能寄存器 – I2CIER

该寄存器定义了相应的 I<sup>2</sup>C 中断使能位。

偏移量： 0x004  
复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位					RXBFIE	TXDEIE	RXDNEIE
						RW	0	RW
类型 / 复位	15	14	13	12	11	10	9	8
	保留位				TOUTIE	BUSERRIE	RXNACKIE	ARBLOSIE
					RW	0	RW	0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位				GCSIE	ADRSIE	STOIE	STAIE
					RW	0	RW	0

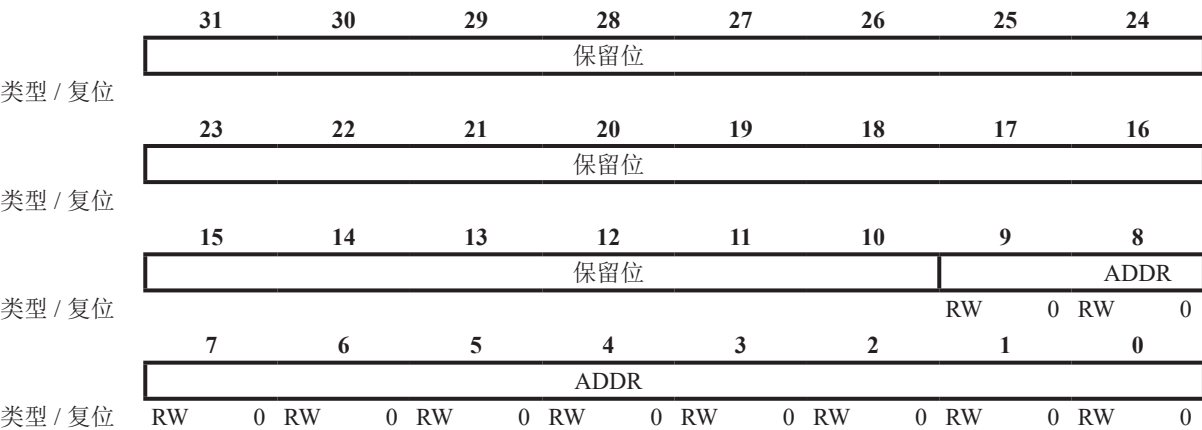
位	字段	描述
[18]	RXBFIE	RX 缓冲器已满中断使能位 0: 中断除能 1: 中断使能
[17]	TXDEIE	发送器模式下数据寄存器空中断使能位 0: 中断除能 1: 中断使能
[16]	RXDNEIE	接收器模式下数据寄存器非空中断使能位 0: 中断除能 1: 中断使能
[11]	TOUTIE	超时中断使能位 0: 中断除能 1: 中断使能
[10]	BUSERRIE	总线错误中断使能位 0: 中断除能 1: 中断使能
[9]	RXNACKIE	接收未确认信号中断使能位 0: 中断除能 1: 中断使能
[8]	ARBLOSIE	I <sup>2</sup> C 多主机模式下仲裁丢失中断使能位 0: 中断除能 1: 中断使能
[3]	GCSIE	广播呼叫从机中断使能位 0: 中断除能 1: 中断使能
[2]	ADRSIE	从机地址匹配中断使能位 0: 中断除能 1: 中断使能

位	字段	描述
[1]	STOIE	STOP 条件检测中断使能位 0: 中断除能 1: 中断使能 此位仅用于 I <sup>2</sup> C 从机模式。
[0]	STAIE	START 条件发送中断使能位 0: 中断除能 1: 中断使能 此位仅用于 I <sup>2</sup> C 主机模式。

I<sup>2</sup>C 地址寄存器 – I2CADDR

该寄存器定义了 I<sup>2</sup>C 设备地址。

偏移量: 0x008  
复位值: 0x0000\_0000



位	字段	描述
[9:0]	ADDR	设备地址 该寄存器定义了 I <sup>2</sup> C 设备地址。当 I <sup>2</sup> C 设备用在 7-bit 寻址模式时，只有 ADDR[6:0] 位与 I <sup>2</sup> C 主机发送的地址相比较。



I²C 状态寄存器 – I2CSR

该寄存器包含了 I²C 的工作状态。

偏移量： 0x00C  
复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位		TXNRX	MASTER	BUSBUSY	RXBF	TXDE	RXDNE
类型 / 复位	15	14	13	12	11	10	9	8
	保留位				TOUTF	BUSERR	RXNACK	ARBLOS
类型 / 复位	7	6	5	4	3	2	1	0
	保留位				GCS	ADRS	STO	STA
					RC	0	RC	0

位	字段	描述
[21]	TXNRX	发送器 / 接收器模式 0: 接收器模式 1: 发送器模式 只读位。
[20]	MASTER	主机模式 0: I²C 在从机模式或空闲 1: I²C 在主机模式 当 I2CTAR 寄存器被赋值且 I²C 总线空闲时，I²C 接口将切换作为 I²C 总线上的主机。 当通过将 I2CEN 位清零或发送一个 STOP 条件给 I²C 总线或检测出 I²C 总线错误来软件除能 I²C 总线时，MASTER 位将由硬件清零。此位通过硬件置位或清零，且为只读位。
[19]	BUSBUSY	总线忙 0: I²C 总线空闲 1: I²C 总线忙 当通过将 I2CEN 位置 1 使能 I²C 接口时，I²C 接口硬件开始检测 I²C 总线的状态。当 SDA 或 SCL 信号被检测到一个逻辑低状态时此位会被置 1，当 STOP 条件被检测到时此位会被清零。
[18]	RXBF	接收器模式下缓冲器已满标志位 0: 数据缓冲器未满 1: 数据缓冲器已满 当数据寄存器 I2CDR 已经存储了一个数据字节，同时数据移位寄存器也已经接收到了一个完整的新数据字节，此位会被置位。RXBF 位通过软件读取 I2CDR 寄存器来清零。

位	字段	描述
[17]	TXDE	发送器模式下数据寄存器为空 0: 数据寄存器 I2CDR 非空 1: 数据寄存器 I2CDR 为空 在发送器模式下, 当 I2CDR 寄存器为空时, 此位被置位。需注意的是, 在地址帧被发送后此位必须置位, 告知发送的数据应该被载入到 I2CDR 寄存器。此位可通过以下方式清零: 在主机和从机模式下通过软件向 I2CDR 寄存器写入数据使其清零; 发送 STOP 信号, 终止数据传输后由硬件自动清零; 主机模式下设置 I2CTAR 寄存器重新开始新的数据传输。
[16]	RXDNE	接收器模式下数据寄存器非空 0: 数据寄存器 I2CDR 为空 1: 数据寄存器 I2CDR 非空 在接收器模式下, 当 I2CDR 寄存器不为空时, 此位被置位。软件读取 I2CDR 寄存器中的数据字节使 RXDNE 位清零。
[11]	TOUTF	超时计数器下溢标志位 0: 无超时计数器下溢发生 1: 超时计数器下溢发生 此位写入 1 将清零 TOUTF 标志位。
[10]	BUSERR	总线错误标志位 0: 无总线错误发生 1: 总线错误发生 在传输过程中, 当 I <sup>2</sup> C 接口检测到一个错误的 START 或 STOP 条件时, 此位将被置位。此位写入 1 将使 BUSERR 标志位清零。 在主机模式下: 一旦总线错误事件发生, SDA 线和 SCL 线都将通过硬件被释放且 BUSERR 位被置位。软件必须在下一个地址字节被发送之前清除 BUSERR 标志位。 在从机模式下: 一旦从机检测到一个错误的 START 或 STOP 条件时, 软件必须在下一个地址字节被接收之前清除 BUSERR 标志位。
[9]	RXNACK	接收未确认信号标志位 0: 从接收器返回确认信号 1: 从接收器返回未确认信号 RXNACK 位表明在主机或从机发送器模式下接收到未确认信号。向此位写入 1 来清除 RXNACK 标志位。
[8]	ARBLOS	仲裁丢失标志位 0: 无仲裁丢失被检测到 1: 位仲裁丢失被检测到 在地址或数据帧发送过程中, 当 I <sup>2</sup> C 接口在主机模式下于当下的时钟周期输掉一个主机总线仲裁时, 此位由硬件置位。向此位写入 1 来清除 ARBLOS 标志位。一旦 ARBLOS 标志位由硬件置位, 则必须在下一次发送之前清除此标志位。
[3]	GCS	广播呼叫从机标志位 0: 无广播呼叫从机发生 1: I <sup>2</sup> C 接口通过广播呼叫命令被寻址 在 7-bit 寻址模式或 10-bit 寻址模式下, I <sup>2</sup> C 接口收到 0x00 或 0x000 的地址时, 如果 GCEN 和 AA 位都被置为 1, 则将切换为广播呼叫从机。此标志位在被读取后自动清零。

位	字段	描述
[2]	ADRS	地址发送 ( 主机模式 ) / 地址接收 ( 从机模式 ) 标志位 主机模式下地址发送 0: 地址帧没被发送 1: 地址帧已被发送 对于 7-bit 寻址模式, 此位在主机接收到从机发送的地址帧确认位后被置位。对于 10-bit 寻址模式, 此位在分别接收到第一个头字节和第二个地址字节的确认位后被置位。应注意若第二个头字节也确认, 此位也会被置位。 从机模式下地址匹配 0: I <sup>2</sup> C 接口没有被寻址 1: I <sup>2</sup> C 接口作为从机被寻址 当 I <sup>2</sup> C 接口接收到与 I2CADDR 寄存器里定义的地址相匹配的呼叫地址, 且 I2CCR 寄存器中的 AA 位被置 1 时, 将切换到从机模式。此标志位在读取 I2CSR 寄存器后被清零。
[1]	STO	STOP 条件检测标志位 0: 无 STOP 条件被检测 1: 从机模式下 STOP 条件被检测 此位只在从机模式下可用, 且在读取 I2CSR 寄存器后自动清零。
[0]	STA	START 条件发送标志位 0: 无 START 条件被发送 1: 在主机模式 START 条件被发送 此位只在主机模式下可用, 且在读取 I2CSR 寄存器后自动清零。

I<sup>2</sup>C SCL 高电平周期发生寄存器 – I2CSHPGR

该寄存器定义了 I<sup>2</sup>C SCL 时钟高电平周期间隔。

偏移量： 0x010  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	SHPG								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
类型 / 复位	SHPG								
	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	SHPG	SCL 时钟高电平周期产生 高电平周期持续时间设置 $SCL_{HIGH} = T_{PCLK} \times (SHPG + d)$ 这里 $T_{PCLK}$ 是 APB 总线外设时钟 (PCLK) 周期，d 值取决于 I <sup>2</sup> C 控制寄存器 (I2CCR) 中 SEQFILTER 的设置。 若 SEQFILTER = 00，d = 6 若 SEQFILTER = 01，d = 8 若 SEQFILTER = 10 或 11，d = 9

I²C SCL 低电平周期产生寄存器 – I2CSLPGR

该寄存器定义了 I²C SCL 时钟低电平周期间隔。

偏移量： 0x014  
复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	SLPG							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	SLPG							
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	SLPG	SCL 时钟低电平周期产生 低电平周期持续时间设置 $SCL_{LOW} = T_{PCLK} \times (SLPG + d)$ 这里 $T_{PCLK}$ 是 APB 总线外设时钟 (PCLK) 周期, d 值取决于 I²C 控制寄存器 (I2CCR) 中 SEQFILTER 的设置。 若 SEQFILTER = 00, d = 6 若 SEQFILTER = 01, d = 8 若 SEQFILTER = 10 或 11, d = 9

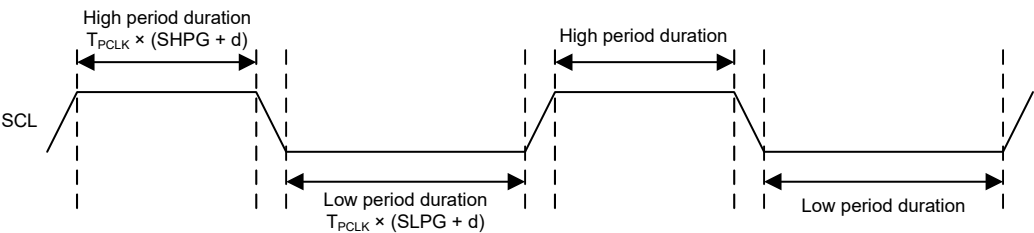


图 139. SCL 时序图

表 48. I²C 时钟设置范例

I²C 时钟	$T_{SCL} = T_{PCLK} \times [ (SHPG + d) + (SLPG + d) ]$ (d = 6) PCLK 时钟下 SHPG + SLPG 的值				
	8 MHz	24 MHz	48 MHz	72 MHz	96 MHz
100 kHz (标准模式)	68	228	468	708	948
400 kHz (快速模式)	8	48	108	168	228
1 MHz (高速模式)	x	12	36	60	84

I<sup>2</sup>C 数据寄存器 – I2CDR

该寄存器定义了由 I<sup>2</sup>C 模块发送和接收的数据。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	DATA							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[7:0]	DATA	I <sup>2</sup> C 数据寄存器 在发送器模式中，发送到从机的一个数据字节可以分配给这些位。如果软件分配新的数据给 I2CDR 寄存器时，TXDE 标志位将被清零。 在接收器模式中，一个数据字节从 MSB 到 LSB 逐位的通过 I <sup>2</sup> C 接口被接收且被储存在数据移位寄存器中。一旦发送了确认位，当 RXDNE 标志位为 0 时，数据移位寄存器的值将被发送到 I2CDR 寄存器。

I<sup>2</sup>C 目标寄存器 – I2CTAR

该寄存器定义了要与之通信的目标设备地址。

偏移量： 0x01C  
复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位					RWD	TAR	
类型 / 复位	7	6	5	4	3	2	1	0
	TAR							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[10]	RWD	读或写方向位 0: 写入目标从机地址 1: 读取目标从机地址 如果在 10-bit 主机接收器模式此位被置 1，那么 I <sup>2</sup> C 接口将在第一个头帧中发起一个值为 11110XX0b 的字节，并由硬件继续在第二个头帧中提供一个值为 11110XX1b 的字节。
[9:0]	TAR	目标从机地址 一旦数据写入该寄存器，I <sup>2</sup> C 接口将会自动发送一个 START 信号和一个目标从机地址。当系统想要发送一个重复的 START 信号给 I <sup>2</sup> C 总线时，建议在一个字节传输完成之后再设置 I2CTAR 寄存器。不允许在地址帧设置 TAR。I2CTAR[9:7] 在 7-bit 寻址模式中不可用。

I²C 地址屏蔽寄存器 – I2CADDRMR

该寄存器定义了 I²C 地址哪个位被屏蔽以及哪个位没有与接收到的地址帧相关位作比较。

偏移量: 0x020  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位									
类型 / 复位										
	15	14	13	12	11	10	9	8		
	保留位						ADDRMR			
类型 / 复位							RW	0	RW	0
	7	6	5	4	3	2	1	0		
	ADDRMR									
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[9:0]	ADDRMR	地址屏蔽控制位 ADDRMR[i] 用于定义 I2CADDR 寄存器 ADDR 字段的第 i 位是否被屏蔽以及是否与接收到的 I²C 总线上的地址帧作比较。该寄存器仅用于 I²C 从机模式。 0: ADDR 第 i 位与 I²C 总线上的地址帧作比较 1: ADDR 第 i 位被屏蔽且没有与 I²C 总线上的地址帧作比较



I²C 地址捕获寄存器 – I2CADDR

该寄存器用于描述 I²C 总线上的地址帧值。

偏移量: 0x024  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	保留位						ADDR		
类型 / 复位							RO	0	RO
	7	6	5	4	3	2	1	0	
	ADDR								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[9:0]	ADDR	地址捕获 一旦 I2CEN 位使能，I²C 总线上的呼叫地址值将被自动载入到该 ADDR 字段。

I<sup>2</sup>C 超时寄存器 – I2CTOUT

该寄存器定义了 I<sup>2</sup>C 超时计数器预载值和时钟预分频比。

偏移量: 0x028  
复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位					PSC		
类型 / 复位	15	14	13	12	11	10	9	8
	TOUT							
类型 / 复位	7	6	5	4	3	2	1	0
	TOUT							

位	字段	描述
[18:16]	PSC	I <sup>2</sup> C 超时计数器预分频选项 该 PSC 字段用于定义 I <sup>2</sup> C 超时计数器时钟频率 f <sub>I2CTO</sub> 。超时时钟频率通过以下公式获得。 $f_{I2CTO} = \frac{f_{PCLK}}{2^{PSC}}$ PSC = 0 → f <sub>I2CTO</sub> = f <sub>PCLK</sub> / 2 <sup>0</sup> = f <sub>PCLK</sub> PSC = 1 → f <sub>I2CTO</sub> = f <sub>PCLK</sub> / 2 <sup>1</sup> = f <sub>PCLK</sub> / 2 PSC = 2 → f <sub>I2CTO</sub> = f <sub>PCLK</sub> / 2 <sup>2</sup> = f <sub>PCLK</sub> / 4 ... PSC = 7 → f <sub>I2CTO</sub> = f <sub>PCLK</sub> / 2 <sup>7</sup> = f <sub>PCLK</sub> / 128
[15:0]	TOUT	I <sup>2</sup> C 超时计数器预载值 TOUT 字段用于定义计数器预载值。 下列情况发生时，计数器值重载： 1. I2CSR 寄存器的 RXBF、TXDE、RXDNE、RXNACK、GCS 或 ADRS 标志位被置位。 2. I <sup>2</sup> C 主机模块发送 START 信号。 3. I <sup>2</sup> C 从机模块检测到 START 信号。 下列情况发生时，计数器停止计数： 1. I <sup>2</sup> C 从机模块没被寻址。 2. I <sup>2</sup> C 主机模块发送 STOP 信号。 3. I <sup>2</sup> C 从机模块检测到 STOP 信号。 4. I2CSR 寄存器的 ARBLOS 或 BUSERR 标志位被置位。

## 20 串行外设接口 (SPI)

### 简介

串行外设接口 SPI 在主从模式下均提供了一个 SPI 协议，即数据传输与发送功能。SPI 接口使用 4 个引脚，其中有串行数据输入和输出线 MISO 和 MOSI，时钟线 SCK 和从机选择线 SEL。SPI 作为主机使用，用 SEL 和 SCK 信号控制数据流来说明数据通信启动和数据采样率。为了接收数据流，SPICR1 寄存器的 DFL 字段定义的 1 位到 16 位数据流在特定的时钟边沿时被锁存且存储在数据寄存器或 RX FIFO。发送数据也是通过类似的方式，但以相反的顺序。模式故障检测功能使其适用于多主机应用。

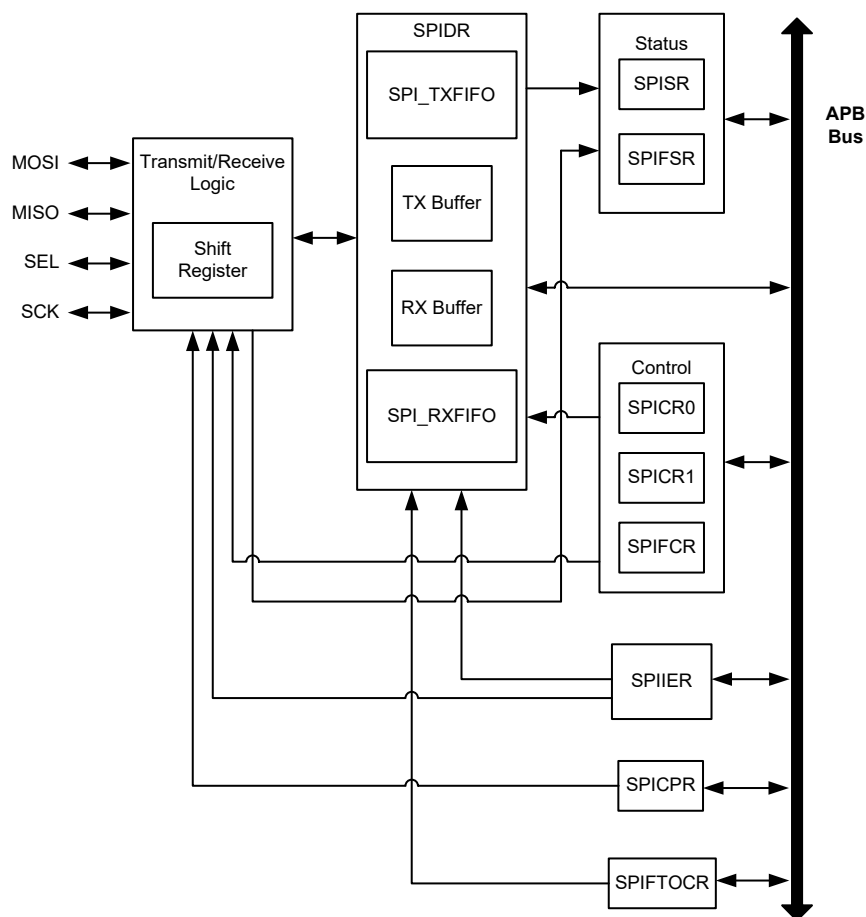


图 140. SPI 方框图

## 特性

- 主机或从机模式
- 主机模式频率高达  $f_{\text{PCLK}} / 2$
- 从机模式频率高达  $f_{\text{PCLK}} / 3$
- 可编程数据帧长度达 16 位
- FIFO 深度：8 级
- MSB 或 LSB 优先传输选择
- 可编程从机选择高或低有效极性
- 多个主机和多个从机工作模式
- 主机模式支持 SPI NOR Flash 双输出读取模式
- 四个错误标志带有各自的中断
  - 读溢出
  - 写冲突
  - 模式故障
  - 从机中止
- 支持 PDMA 接口

## 功能描述

### 主机模式

每个数据帧的数据长度范围是 1 ~ 16 位。被发送的数据的第一位可以是 MSB 或 LSB，这由 SPICR1 寄存器的 FIRSTBIT 位决定。SPI 模块可以由 SPICR1 寄存器中的 MODE 位配置作为主机或从机。当 MODE 位被置位，SPI 模块被配置作为主机且在 SCK 引脚产生串行时钟。在串行时钟边沿发生时，数据流将移位寄存器中的数据发送到 MOSI 引脚。在数据传输的整个过程中，SEL 引脚保持有效电平。当 SPICR1 寄存器中的 SELAP 位被置位，在完整的数据处理过程中 SEL 引脚为高电平有效。当 SPICR1 寄存器中的 SELM 位被置位，SEL 引脚将被硬件自动驱动，在 SEL 有效边沿和 SCK 的第一个边沿之间的时间间隔等于 SCK 周期的一半。

### 从机模式

在从机模式，SCK 引脚作为输入脚且串行时钟来自于外部主机。SEL 引脚也作为输入脚。当 SELAP 位被清零，在完整的数据流接收期间 SEL 引脚为低电平有效。当 SELAP 位被置为 1，在完整数据流接收期间，SEL 引脚为高电平有效。

注：在从机模式，APB 时钟，即  $f_{\text{PCLK}}$ ，必须至少为外部 SCK 时钟输入频率的 3 倍。

SPI 串行帧格式

SPI 接口帧格式取决于时钟极性位 CPOL 和时钟相位位 CPHA 的配置。

- 时钟极性位 – CPOL  
当时钟极性位被清零，SCK 线空闲状态为低电平。当时钟极性位被置高，SCK 线空闲状态为高电平。
- 时钟相位位 – CPHA  
当时钟相位位被清零，数据会在第一个 SCK 时钟转换时被采样。当时钟相位位被置高，数据会在第二个 SCK 时钟转换时被采样。

SPI 接口有四种帧格式。表 49 显示了如何通过 SPICR1 寄存器中的 FORMAT 字段设置这些格式。

表 49. SPI 接口格式设置

FORMAT[2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
其它值	保留位	

CPOL = 0, CPHA = 0

在此格式下，接收到的数据在 SCK 线上升沿被采样，而被发送的数据在 SCK 线下降沿改变数据位。在主机模式中，当数据写入 SPIDR 寄存器时，第一个位被驱动。在从机模式中，当 SEL 信号变为有效电平时，第一个位被驱动。图 141 显示了此格式下单个字节数据传输时序图。

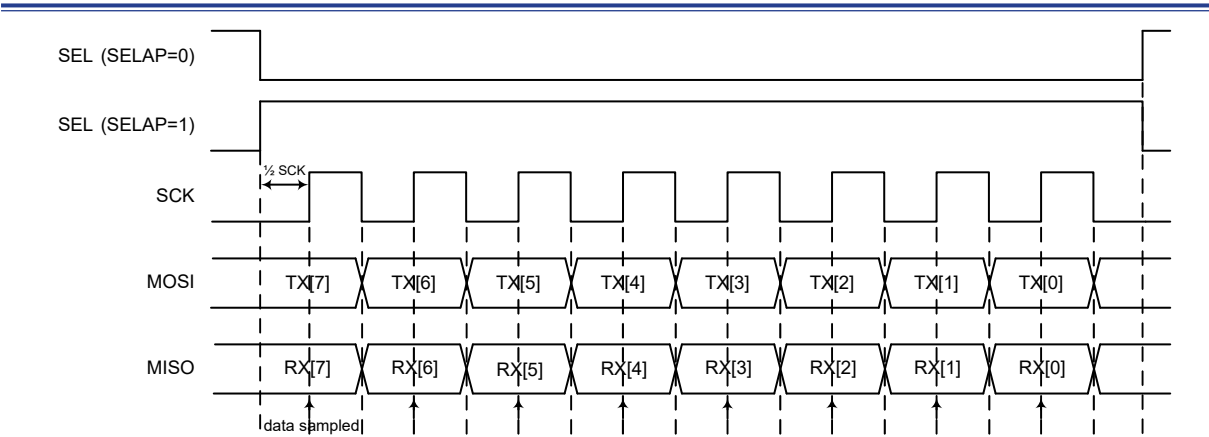


图 141. SPI 单字节数据传输时序图 – CPOL = 0, CPHA = 0

图 142 显示了此格式下连续数据传输的时序图。注意，在每个数据帧之间 SEL 信号必须转变为无效电平。

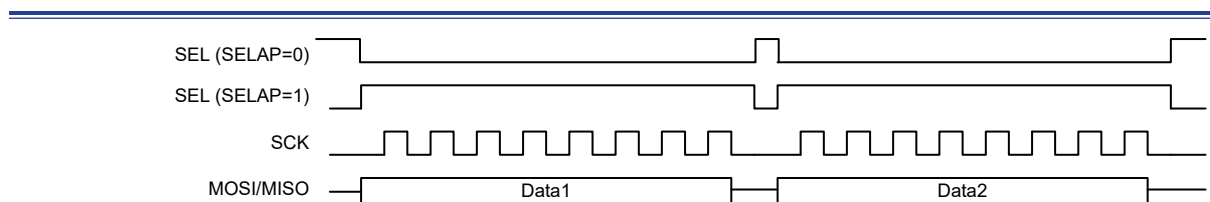


图 142. SPI 连续数据传输时序图 – CPOL = 0, CPHA = 0

### CPOL = 0, CPHA = 1

在此格式下，接收到的数据在 SCK 线下降沿被采样，而被发送的数据在 SCK 线上升沿改变数据位。在主机模式中，当数据写入 SPIDR 寄存器时，第一个位被驱动。在从机模式中，第一个 SCK 时钟上升沿到来时，第一个位被驱动。图 143 显示了单个数据字节传输的时序。

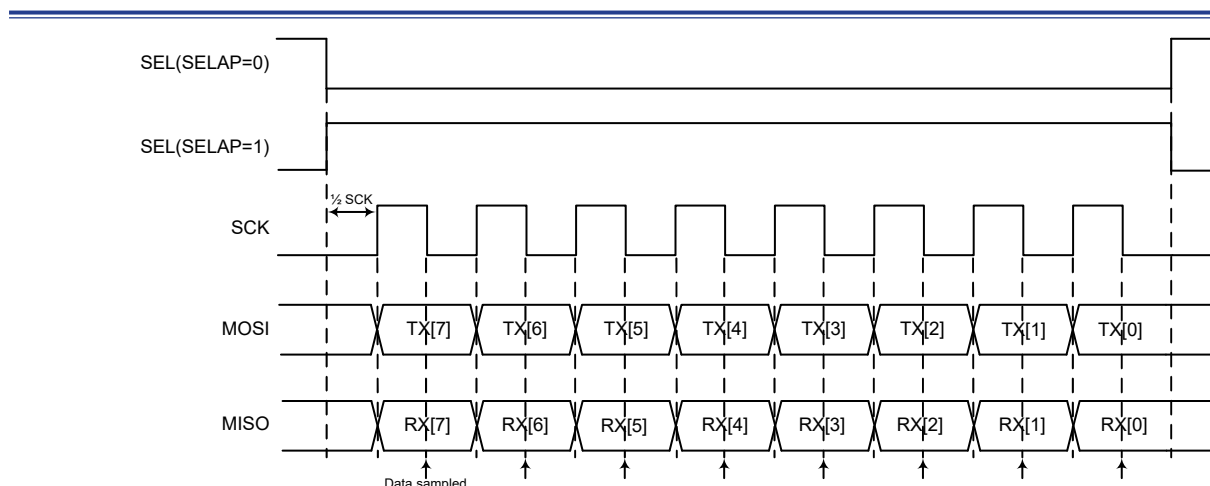


图 143. SPI 单个字节传输时序图 – CPOL = 0, CPHA = 1

图 144 显示了连续数据传输时序图。注意，SEL 信号必须保持在有效电平直到最后一个数据传输结束。

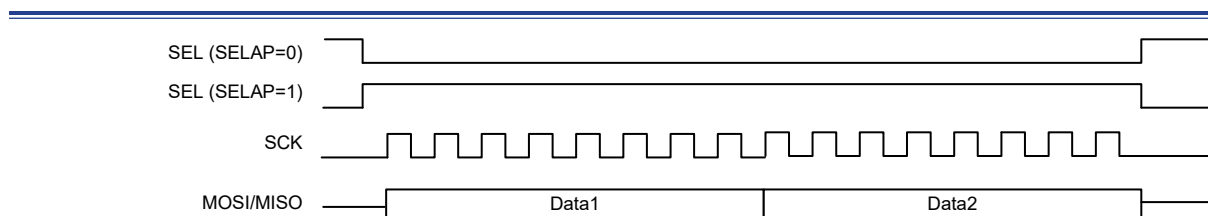


图 144. SPI 连续传输时序图 – CPOL = 0, CPHA = 1

#### CPOL = 1, CPHA = 0

在此格式下，接收到的数据在 SCK 线下降沿被采样，而被发送的数据在 SCK 线上升沿改变数据位。在主机模式中，当数据写入 SPIDR 寄存器时，第一个位被驱动。在从机模式中，在 SEL 信号转换为有效电平时，第一个位被驱动。图 145 显示了此格式下单个字节数据传输的时序图。

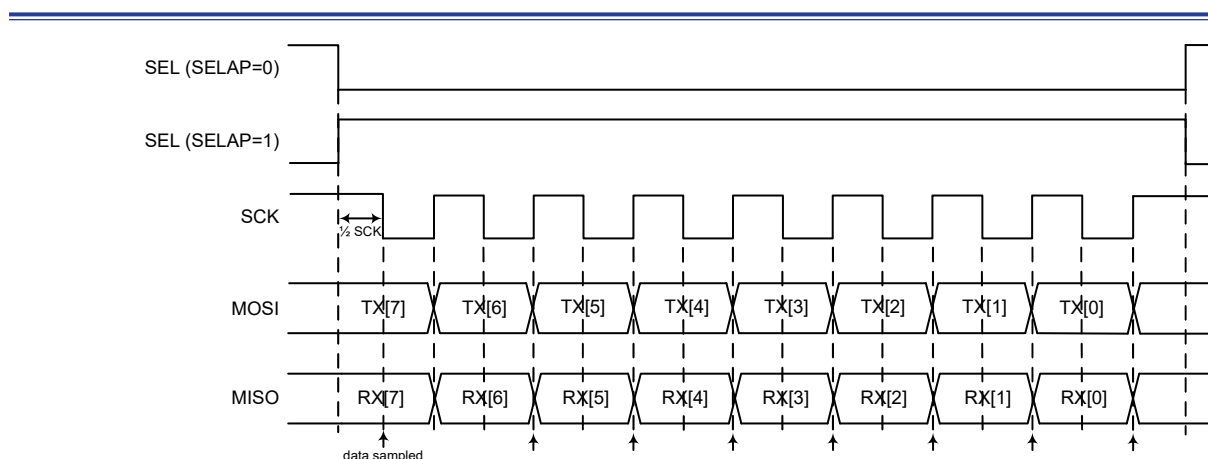


图 145. SPI 单个字节传输时序图 – CPOL = 1, CPHA = 0

图 146 显示于此格式下连续数据传输的时序图。注意，SEL 信号必须在每个数据帧之间转换为无效电平。

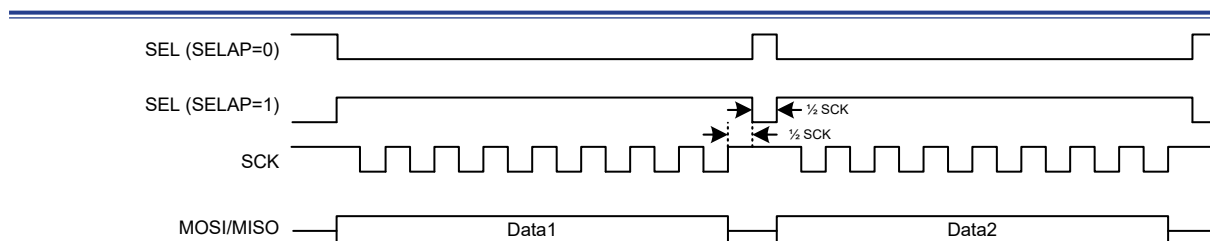


图 146. SPI 连续数据传输时序图 – CPOL = 1, CPHA = 0

### CPOL = 1, CPHA = 1

在此格式下，接收到的数据在 SCK 线上升沿被采样，而被发送的数据在 SCK 线下降沿改变数据位。在主机模式中，当数据写入 SPIDR 寄存器时，第一个位被驱动。在从机模式中，第一个 SCK 时钟下降沿到来时，第一个位被驱动。图 147 显示于此格式下单个字节数据传输的时序图。

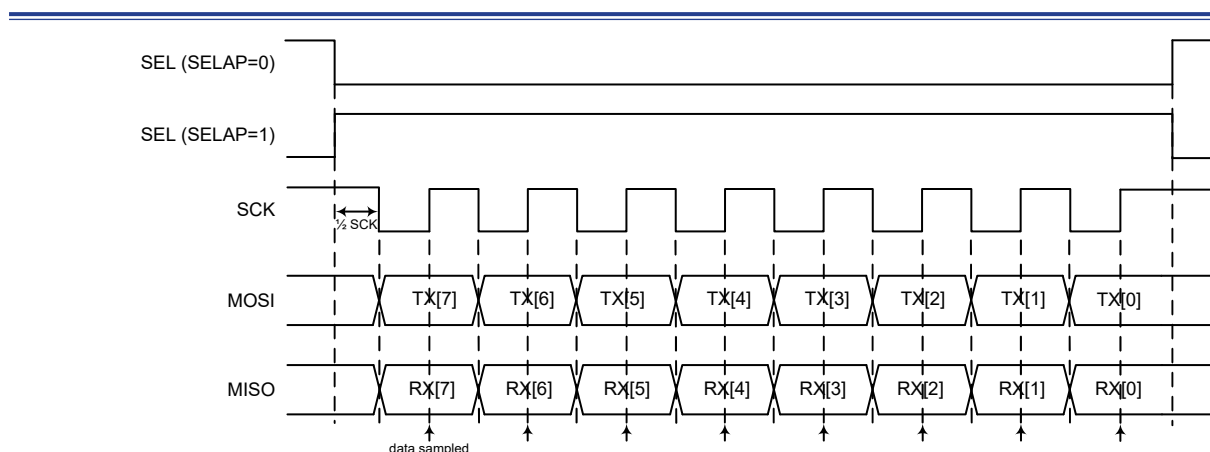


图 147. SPI 单个字节传输时序图 – CPOL = 1, CPHA = 1

图 148 显示于此格式下连续数据传输的时序图。注意，SEL 信号必须保持在有效电平直到最后一个数据传输结束。

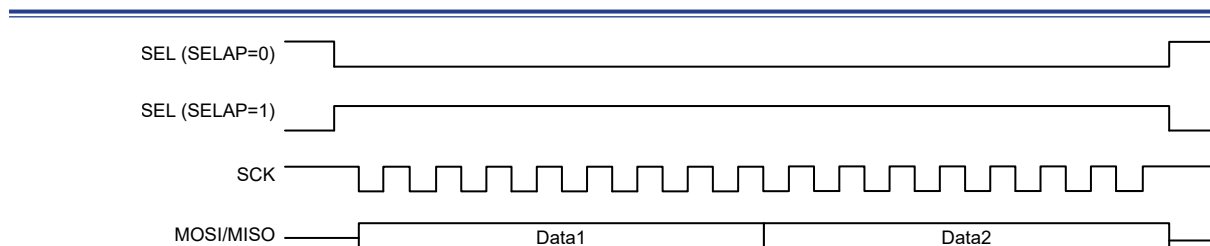


图 148. SPI 连续数据传输时序图 – CPOL = 1, CPHA = 1



## 状态标志

### TX 缓存器空 – TXBE

在非 FIFO 模式下 TX 缓存器为空时，或在 FIFO 模式下当 TX FIFO 数据长度等于或小于 SPIFCR 寄存器中 TXFTLS 字段定义的 TX FIFO 阈值级别时，TXBE 标志位被置位。接着要发送的数据可被载入缓存器。此后，在非 FIFO 模式下 TX 缓存器已经包含了一个新的数据时，或在 FIFO 模式下当 TX FIFO 数据长度大于 SPIFCR 寄存器中 TXFTLS 字段定义的 TX FIFO 阈值级别时，TXBE 标志位被复位。

### 传输寄存器空 – TXE

当 TX 缓存器和 TX 移位寄存器都为空时，TXE 标志位被置位。当 TX 缓存器或 TX 移位寄存器包含新的被发送的数据时，TXE 标志位被复位。

### RX 缓存器非空 – RXBNE

在非 FIFO 模式下 RX 缓存器中的数据为有效接收数据，或在 SPI FIFO 模式下 RX FIFO 数据长度等于或大于 SPIFCR 寄存器中 RXFTLS 字段定义的 RX FIFO 阈值时，RXBNE 标志位被置位。在非 FIFO 模式下当接收到的数据已从 RX 缓存器中被全部读取，或在 FIFO 模式下当 RX FIFO 数据长度小于由 RXFTLS 字段设置的 RX FIFO 阈值时，此标志位将由硬件自动清零。

### 超时标志 – TO

超时功能仅在 SPI FIFO 模式下有效，可通过载 0 到超时计数器寄存器的 TOC 字段来除能。如果 SPI RX FIFO 非空，超时计数器将开始计数，一旦数据从 SPIDR 寄存器被读出，或者接收到新的数据，超时计数器将复位为 0 并再次开始计数。当超时计数器的值等于 SPIFTOCR 寄存器中的 TOC 字段指定的值时，TO 标志位将被置位。此标志位通过向此位写 1 来清零。

### 模式故障标志 – MF

模式故障标志可用于检测 SPI 多主机模式下的 SPI 总线使用情况。在多主机模式下，SPI 模块被配置作为主机且 SEL 信号被设置作为输入信号。当 SEL 引脚突然被另一个主机变换为有效电平时，模式故障标志位将被置位。这意味着另一个 SPI 主机正在请求使用 SPI 总线。因此，当 SPI 模式故障发生，将迫使此 SPI 模块工作在从机模式且除能此 SPI 接口所有的信号来避免 SPI 总线信号冲突。同样地，如果 SPI 主机想要传输数据，有必要通过驱动其它主机的 SEL 信号变为一个有效状态来告知其它的 SPI 主机。SPI 多主机模式下的详细配置图如下图所示。

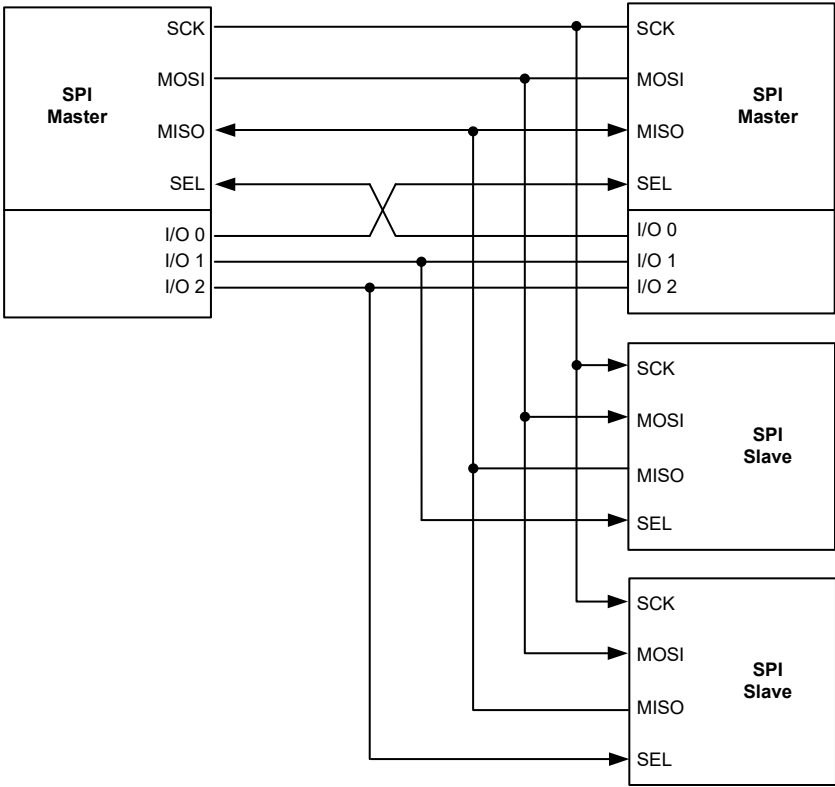


图 149. SPI 多主机从机环境

表 50. SPI 模式故障触发条件

模式故障	描述
触发条件	1. SPI 主机模式。 2. SPICR0 寄存器的 SELOEN = 0 – SEL 引脚被配置为输入模式。 3. 当被外部 SPI 主机驱动时，SEL 信号转换为有效电平。
SPI 行为	1. 模式故障标志位置位。 2. SPICR0 寄存器中的 SPIEN 位被复位。这将除能 SPI 接口和阻止所有来自单片机的输出信号。 3. SPICR1 寄存器中的 MODE 位被复位。这将迫使单片机进入从机模式。

表 51. SPI 主机模式 SEL 引脚状态

	SEL 作为输入 – SELOEN = 0		SEL 作为输出 – SELOEN = 1	
多主机	支持		不支持	
SPI SEL 控制信号	使用另一个 GPIO 代替 SEL 引脚功能		SEL 引脚处于硬件控制或软件控制模式 – 使用 SELM 位设置	
连续传输	情况 1	情况 2	情况 1	情况 2
	不支持	支持	使用硬件控制	硬件或软件控制

情况 1: 每次数据传输之间，SEL 信号必须为无效。  
情况 2: 直到最后一个数据帧传输完成后，SEL 信号才会有效。  
注: 当 SPI 模块工作在从机模式时，SEL 信号始终作为输入且不受 SPICR0 寄存器中 SELOEN 位的影响。

### 写冲突标志 – WC

下列情况将使写冲突标志置位：

- SPIFCR 寄存器中的 FIFOEN 位被清零。  
TX 缓存器和移位寄存器都已经为满时，当有新的数据写入到 SPIDR 寄存器，写冲突标志会被置位。任何被写入 TX 缓存器的新的数据将丢失。
- SPIFCR 寄存器中的 FIFOEN 位被置位。  
TX FIFO 和 TX 移位寄存器都已经为满时，当有新的数据写入到 SPIDR 寄存器，写冲突标志会被置位。任何被写入 TX FIFO 的新的数据都将丢失。

### 读溢出标志 – RO

- SPIFCR 寄存器中的 FIFOEN 位被清零。  
如果再一笔数据被接收时读溢出标志位被置位，表明 RX 移位寄存器和 RX 缓存器都已满。这意味着最新接收的数据不会被移到 SPI 移位寄存器中。所以最新的接收数据将丢失。
- SPIFCR 寄存器中的 FIFOEN 位被置位。  
如果再一笔数据被接收时读溢出标志位被置位，表明 RX 移位寄存器和 RX FIFO 都已满。这意味着最新的接收数据不会被移到 SPI 移位寄存器中。所以最新的接收数据将丢失。

### 从机中止标志 – SA

在 SPI 从机模式，从机中止标志位被置位表明在数据帧传输期间 SEL 引脚被突然转换到一个无效的状态。数据帧长度由 SPICR1 寄存器中的 DFL 字段定义。

### PDMA 接口

PDMA 接口集成在 SPI 模块内。在发送器或接收器模式下分别通过设置 TXDMAE 或 RXDMAE 位为 1 可使能 PDMA 功能。在发送缓存器空标志位 TXBE 被置位且 TXDMAE 位置 1 时，PDMA 功能将被激活用于把数据从用户指定的存储器位置移到 SPI 数据寄存器或者 TX FIFO 直到 TXBE 标志位被清零。当在非 FIFO 模式下发送缓存器为空，或 FIFO 模式下 TX FIFO 包含的数据长度等于或小于 TXFTLS 字段定义的级别时，TXBE 标志位被置位。

同样地，在接收缓存器非空标志位 RXBNE 被置位且 RXDMAE 位置 1，PDMA 功能将被激活用于把数据从 SPI 数据寄存器或者 RX FIFO 移到用户指定的存储器位置直到 RXBNE 标志位被清零。当在非 FIFO 模式下接收缓存器非空，或 FIFO 模式下 RX FIFO 包含的数据长度等于或大于 RXFTLS 字段定义的级别时，RXBNE 标志位被置位。

详细的 PDMA 配置信息，请参考 PDMA 章节。

寄存器列表

下表显示了 SPI 寄存器及其复位值。

表 52. SPI 寄存器列表

寄存器	偏移量	描述	复位值
SPICR0	0x000	SPI 控制寄存器 0	0x0000_0000
SPICR1	0x004	SPI 控制寄存器 1	0x0000_0000
SPIIER	0x008	SPI 中断使能寄存器	0x0000_0000
SPICPR	0x00C	SPI 时钟预分频器寄存器	0x0000_0000
SPIDR	0x010	SPI 数据寄存器	0x0000_0000
SPISR	0x014	SPI 状态寄存器	0x0000_0003
SPIFCR	0x018	SPI FIFO 控制寄存器	0x0000_0000
SPIFSR	0x01C	SPI FIFO 状态寄存器	0x0000_0000
SPIFTOCR	0x020	SPI FIFO 超时计数器寄存器	0x0000_0000

寄存器描述

SPI 控制寄存器 0 – SPICR0

该寄存器定义了 SEL 控制位和 SPI 使能位。

偏移量： 0x000  
复位值： 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
SELHT				GUADT			
RW	0	RW	0	RW	0	RW	0
7	6	5	4	3	2	1	0
GUADTEN	DUALEN	保留位	SSELC	SELOEN	RXDMAE	TXDMAE	SPIEN
RW	0	RW	0	RW	0	RW	0

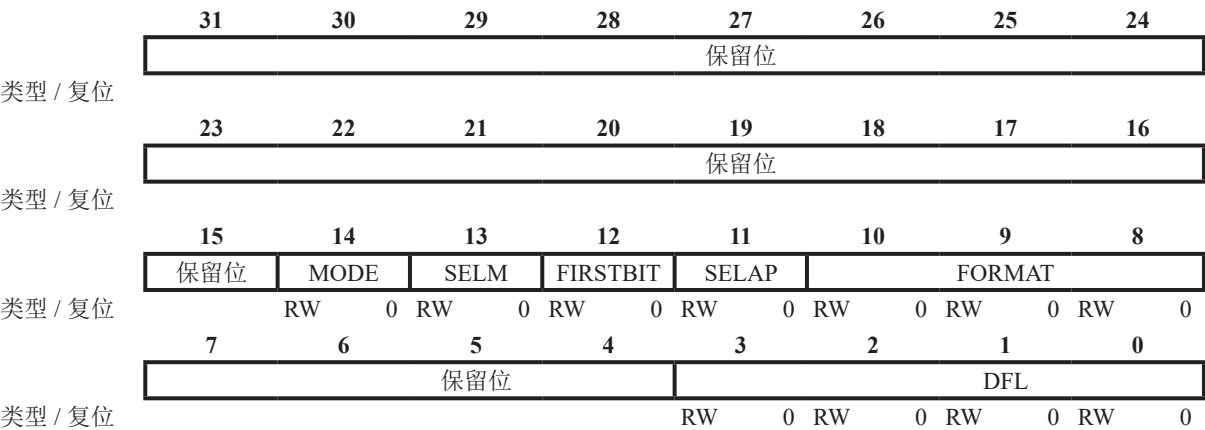
位	字段	描述
[15:12]	SELHT	芯片选择保持时间控制 0x0: 1/2 SCK 0x1: 1 SCK 0x2: 3/2 SCK 0x3: 2 SCK .... 注意，SELHT 位仅适用于主机模式。

位	字段	描述
[11:8]	GUADT	保护时间控制 GUADTEN = 1 0x0: 1 SCK 0x1: 2 SCK 0x2: 3 SCK ... 注意, GUADT 位仅适用于主机模式。
[7]	GUADTEN	保护时间使能位 0: 保护时间为 1/2 SCK 1: 当该位置高, 保护时间由 GUADT 位控制 注意, GUADTEN 位仅适用于主机模式。
[6]	DUALEN	双端口使能位 0: 双端口除能 1: 双端口使能 该控制位用于支持串行 SPI NOR Flash 双输出读取模式。当该位置位, MOSI 信号将改变方向由输出变为输入并接收串行数据流。这意味着 DUALEN 控制位仅适用于主机模式。
[4]	SSELC	软件从机选择控制位 0: 设置 SEL 输出为一个无效状态 1: 设置 SEL 输出为一个有效状态 软件可以通过设置 SSELC 位使 SEL 输出为一个有效或无效状态。有效电平通过 SPICR1 寄存器中的 SELAP 位设置。注意, 只有当 SELOEN 位置 1 使能 SEL 输出且同时 SELM 位清零使用软件控制 SEL 信号, SSELC 位才是可用的。否则, SSELC 位无影响。
[3]	SELOEN	从机选择输出使能位 0: 设置 SEL 信号为输入模式用于多主机模式 1: 设置 SEL 信号为输出模式用于从机选择 SELOEN 仅在主机模式可用, 用来设置 SEL 信号作为输入或输出信号。当 SEL 信号被配置为工作在输出模式时, 根据 SPICR1 寄存器中 SELM 位的设置, 它可作为硬件控制或软件控制模式下的从机选择信号。当 SEL 信号被配置工作在输入模式时, 它可在多主机环境中用作模式故障检测功能。
[2]	RXDMAE	RX PDMA 请求使能位 0: SPI RX 通道 PDMA 请求除能 1: SPI RX 通道 PDMA 请求使能
[1]	TXDMAE	TX PDMA 请求使能位 0: SPI TX 通道 PDMA 请求除能 1: SPI TX 通道 PDMA 请求使能
[0]	SPIEN	SPI 使能位 0: SPI 接口除能 1: SPI 接口使能

SPI 控制寄存器 1 – SPICR1

该寄存器定义了 SPI 的参数，包括数据长度、传输格式、SEL 有效极性 / 模式、LSB / MSB 控制以及主机 / 从机模式。

偏移量：0x004  
复位值：0x0000\_0000



位	字段	描述																		
[14]	MODE	主机或从机模式 0: 从机模式 1: 主机模式																		
[13]	SELM	从机选择模式 0: SEL 信号由软件控制 – 由 SSEL 位控制有效或无效 1: SEL 信号由硬件控制 – 由 SPI 硬件自动产生 注意，只有在主机模式下 SELM 位可用 – MODE = 1																		
[12]	FIRSTBIT	LSB 或 MSB 优先发送 0: MSB 优先发送 1: LSB 优先发送																		
[11]	SELAP	从机选择有效极性 0: SEL 信号低有效 1: SEL 信号高有效																		
[10:8]	FORMAT	SPI 数据传输格式 这些位用来决定 SPI 接口数据传输格式																		
<table><tr><th>FORMAT[2:0]</th><th>CPOL</th><th>CPHA</th></tr><tr><td>001</td><td>0</td><td>0</td></tr><tr><td>010</td><td>0</td><td>1</td></tr><tr><td>110</td><td>1</td><td>0</td></tr><tr><td>101</td><td>1</td><td>1</td></tr><tr><td>其它值</td><td colspan="2">保留位</td></tr></table>			FORMAT[2:0]	CPOL	CPHA	001	0	0	010	0	1	110	1	0	101	1	1	其它值	保留位	
FORMAT[2:0]	CPOL	CPHA																		
001	0	0																		
010	0	1																		
110	1	0																		
101	1	1																		
其它值	保留位																			
CPOL: 时钟极性 0: SCK 空闲状态为低 1: SCK 空闲状态为高																				
CPHA: 时钟相位 0: 数据在第一个 SCK 时钟沿被捕捉 1: 数据在第二个 SCK 时钟沿被捕捉																				

位                  字段                  描述

[3:0]                  DFL                  数据帧长度  
在 1 ~ 16 位之间选择数据传输的数据帧长度

DFL[3:0]	串行 SPI 模式	双线 SPI 模式
0001	1 bit	2 bits
0010	2 bits	4 bits
0011	3 bits	6 bits
0100	4 bits	8 bits
0101	5 bits	10 bits
0110	6 bits	12 bits
0111	7 bits	14 bits
1000	8 bits	16 bits
1001	9 bits	—
...	...	—
1111	15 bits	—
0000	16 bits	—

注：数据位的总数量由 DFL 字段的配置以及所选择的 SPI 设备传输模式共同确定。

以 8-bit 数据传输为例，DFL 设定值计算如下：

在串行 SPI 模式中，数据帧长度 =  $8/1 = 8$ ，DFL = 0x8；

在双线 SPI 模式中，数据帧长度 =  $8/2 = 4$ ，DFL = 0x4；

SPI 中断使能寄存器 – SPIIER

该寄存器包含了相关的 SPI 中断使能控制位。

偏移量： 0x008  
复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位	7	6	5	4	3	2	1	0
	TOIEN	SAIEN	MFIEN	ROIEN	WCIEN	RXBNEIEN	TXEIEN	TXBEIEN
类型 / 复位	RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW	0

位	字段	描述
[7]	TOIEN	超时中断使能位 0: 除能 1: 使能
[6]	SAIEN	从机中止中断使能位 0: 除能 1: 使能
[5]	MFIEN	模式故障中断使能位 0: 除能 1: 使能
[4]	ROIEN	读溢出中断使能位 0: 除能 1: 使能
[3]	WCIEN	写冲突中断使能位 0: 除能 1: 使能
[2]	RXBNEIEN	RX 缓存器非空中断使能位 0: 除能 1: 使能 RXBNE 标志位和 RXBNEIEN 位被置位时，会产生一个中断请求。在 FIFO 模式下，中断请求标志的产生取决于 RX FIFO 触发级别的设置。
[1]	TXEIEN	TX 空中断使能位 0: 除能 1: 使能 TXE 标志位和 TXEIEN 位被置位时，将产生 TX 寄存器空中断请求。
[0]	TXBEIEN	TX 缓存器空中断使能位 0: 除能 1: 使能 TXBE 标志位和 TXBEIEN 位被置位时，TX 缓存器空中断请求将会产生。在 FIFO 模式下，中断请求标志的产生取决于 TX FIFO 触发级别的设置。



SPI 时钟预分频器寄存器 – SPICPR

该寄存器定义了 SPI 时钟预分频比。

偏移量： 0x00C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	CP								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
类型 / 复位	CP								
	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[15:0]	CP	SPI 时钟预分频器 SPI 时钟 SCK 频率由以下等式决定： $f_{SCK} = f_{PCLK} / (2 \times (CP + 1))$ ，CP 范围：0 ~ 65535 注：在 SPI 主机模式下，APB 时钟 ( $f_{PCLK}$ ) 必须至少比 SPI SCK 输出频率快 2 倍。

SPI 数据寄存器 – SPIDR

该寄存器存储了 SPI 接收或发送的数据。

偏移量: 0x010  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位									
类型 / 复位										
	15	14	13	12	11	10	9	8		
	DR									
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0		
	DR									
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	DR	数据寄存器 SPI 数据寄存器用来存储串行总线发送或接收的数据。在非 FIFO 模式下，写数据到 SPI 数据寄存器也会使数据加载到数据发送缓存器即 TX 缓存器内。从 SPI 数据寄存器中读取数据将会返回接收缓存器即 RX 缓存器中的数据。

SPI 状态寄存器 – SPISR

该寄存器包含了相关的 SPI 状态。

偏移量： 0x014

复位值： 0x0000\_0003

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位							BUSY
							RO	0
类型 / 复位	7	6	5	4	3	2	1	0
	TO	SA	MF	RO	WC	RXBNE	TXE	TXBE
	WC	0	WC	0	WC	0	RO	0
							1	RO
								1

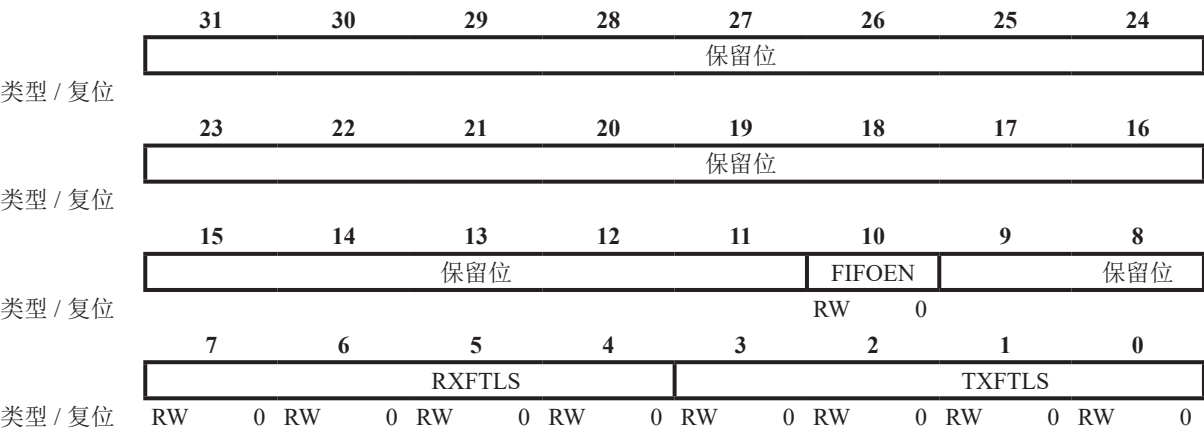
位	字段	描述
[8]	BUSY	SPI 忙标志位 0: SPI 不忙 1: SPI 忙 在主机模式，当 TX 缓存器和 TX 移位寄存器都为空时，此位被复位。当 TX 缓存器或 TX 移位寄存器不为空时，此位被置位。 在从机模式，当 SEL 信号转变为有效电平时，此位被置位。当 SEL 信号转变为无效电平时，此位被复位。
[7]	TO	超时标志位 0: 无 RX FIFO 超时 1: RX FIFO 超时发生 一旦超时计数器的值等于 SPIFTOCR 寄存器中 TOC 字段的设定时，超时标志位将置位，SPIIER 寄存器中的 TOIEN 位使能，则将产生中断。此位通过写 1 使其清零。 注：只有在 SPI FIFO 模式下，超时标志功能可用。
[6]	SA	从机中止标志 0: 无从机中止发生 1: 从机中止发生 此位由硬件置位且通过写 1 清零。
[5]	MF	模式故障标志 0: 无模式故障 1: 模式故障发生 此位由硬件置位且通过写 1 清零。
[4]	RO	读溢出标志 0: 无读溢出 1: 读溢出发生 此位由硬件置位且通过写 1 清零。
[3]	WC	写冲突标志 0: 无写冲突 1: 写冲突发生 此位由硬件置位且通过写 1 清零。

位	字段	描述
[2]	RXBNE	接收缓存器非空标志位 0: RX 缓存器为空 1: RX 缓存器非空 此位表明了非 FIFO 模式下 RX 缓存器的状态。它也用来说明在 FIFO 模式下是否达到 RX FIFO 触发级别。在非 FIFO 模式下，当 SPI RX 缓存器为空时或在 SPI FIFO 模式下当存放在 FIFO 的数据量小于由 SPIFCR 寄存器中的 RXFTLS 字段定义的触发级别时，此位将被清零。
[1]	TXE	发送寄存器空标志位 0: TX 缓存器或 TX 移位寄存器非空 1: TX 缓存器和 TX 移位寄存器都为空
[0]	TXBE	发送缓存器空标志位 0: TX 缓存器非空 1: TX 缓存器为空 在 FIFO 模式下，此位表明存放在 TX FIFO 的数据量等于或小于 SPIFCR 寄存器中 TXFTLS 字段定义的触发级别。

SPI FIFO 控制寄存器 – SPIFCR

该寄存器包含了 SPI FIFO 相关的控制，包括 FIFO 使能控制和 FIFO 触发级别选项。

偏移量： 0x018  
复位值： 0x0000\_0000



位	字段	描述
[10]	FIFOEN	FIFO 使能位 0: FIFO 除能 1: FIFO 使能 当 SPI 接口正在发送时，此位不可被置位或复位。
[7:4]	RXFTLS	RX FIFO 触发级别选择 0000: 触发级别为 0 0001: 触发级别为 1 ... 1000: 触发级别为 8 其它值: 保留 RXFTLS 字段用来定义 RX FIFO 触发级别。当 RX FIFO 中的数据量等于或大于 RXFTLS 字段定义的触发级别时，RXBNE 标志将被置位。
[3:0]	TXFTLS	TX FIFO 触发级别选择 0000: 触发级别为 0 0001: 触发级别为 1 ... 1000: 触发级别为 8 其它值: 保留 TXFTLS 字段用来定义 TX FIFO 触发级别。当 TX FIFO 中的数据量等于或小于 TXFTLS 字段定义的触发级别时，TXBE 标志位被置位。

SPI FIFO 状态寄存器 – SPIFSR

该寄存器包含了 SPI FIFO 相关的状态。

偏移量： 0x01C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	RXFS				TXFS			
类型 / 复位	RO	0	RO	0	RO	0	RO	0

位	字段	描述
[7:4]	RXFS	RX FIFO 状态 0000: RX FIFO 为空 0001: RX FIFO 包含 1 个数据 ... 1000: RX FIFO 包含 8 个数据 其它值: 保留
[3:0]	TXFS	TX FIFO 状态 0000: TX FIFO 为空 0001: TX FIFO 包含 1 个数据 ... 1000: TX FIFO 包含 8 个数据 其它值: 保留

SPI FIFO 超时计数器寄存器 – SPIFTOCR

该寄存器存储了 SPI RX FIFO 超时计数器的值。

偏移量: 0x020  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位									
类型 / 复位										
	15	14	13	12	11	10	9	8		
	TOC									
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0		
	TOC									
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	TOC	超时计数器 超时计数器会在 SPI RX FIFO 接收到数据后从 0 开始计数，且一旦通过软件从 SPIDR 寄存器读取数据，或者接收到新的数据时，会复位计数器。如果 FIFO 没接收到新的数据或数据没有通过软件从 SPIDR 寄存器被读取，超时计数器值连续递增。超时计数器值等于 TOC 设置值时，SPISR 寄存器的 TO 标志位将被置位，若 SPIIER 寄存器的 TOIEN 位已置位，将会产生中断。当 RX FIFO 为空时，超时计数器将被停止。通过清零 TOC 字段，SPI FIFO 超时功能可除能。超时计数器由系统 APB 时钟 f <sub>PCLK</sub> 驱动。

## 21 通用同步异步收发器 (USART)

### 简介

通用同步异步收发器 USART, 提供了一个灵活的同步或异步传输的全双工数据交换。USART 用来转换并行和串行接口之间的数据, 通常也被用作 RS232 标准通信。USART 外设功能支持多种类型的中断。

USART 模块包含一个 8 级的发送器 FIFO (TX FIFO) 和一个 8 级的接收器 FIFO (RX FIFO)。通过读取 USART 状态 & 中断标志位寄存器 USRSIFR, 软件可检测 USART 的错误状态。这些状态包括传输类型和状况以及因奇偶、溢出、帧和暂停事件造成的错误状况。

USART 有一个可编程的波特率发生器, 能对 CK\_APB (CK\_USART) 的 USART 时钟进行分频以产生 USART 发送器和接收器所需的时钟。

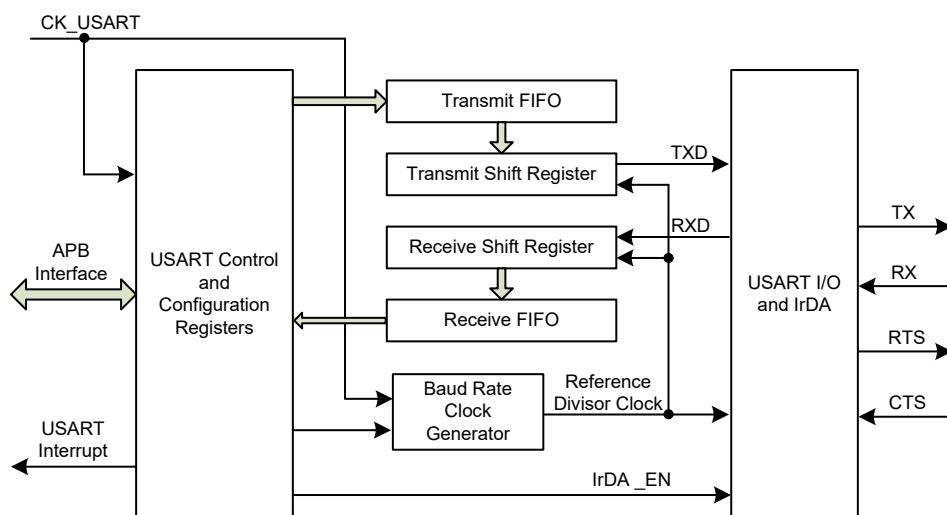


图 150. USART 方框图



## 特性

- 支持异步和时钟同步串行通信模式
- 全双工通信能力
- 可编程波特率时钟频率：
  - 异步模式下高达 ( $f_{\text{CLK}} / 16$ ) MHz，同步模式下高达 ( $f_{\text{CLK}} / 8$ ) MHz
- IrDA SIR 编码器和解码器
  - 支持正常 3 / 16 位持续时间和低功耗持续时间 (1.41 ~ 2.23  $\mu\text{s}$ )
- 支持带有输出使能的 RS485 模式
- 自动硬件流控制模式 – RTS, CTS
- 完全可编程串行通信功能包括：
  - 字长：7、8 或 9-bit 字符
  - 校验位：偶、奇或无奇偶校验位的产生和检测
  - 停止位：1 或 2 个停止位
  - 位顺序：LSB 优先或 MSB 优先传输
- 错误检测：奇偶、溢出和帧错误
- FIFO：
  - 接收器 FIFO：8 级 (最多 9 个数据位)
  - 发送器 FIFO：8 级 (最多 9 个数据位)
- 支持 PDMA 接口

功能描述

串行数据格式

USART 模块对从发送器 FIFO 中读取的数据进行并行到串行的转换，然后发送具有以下格式的数据：起始位、7 ~ 9 个 LSB/MSB 优先的数据位、可选的奇偶校验位和最后 1 ~ 2 个停止位。起始位的极性与数据线空闲状态相反。停止位与数据线空闲状态相同，并在下一步开始情况发生之前提供延迟。开始和停止位都用于异步数据传输过程中的数据同步。

USART 模块对从接收器 FIFO 中读取的数据进行串行到并行的转换。它会首先检查校验位，然后将寻找一个停止位。如果停止位没有找到，USART 模块会认为整个字的传输失败并响应一个帧错误。

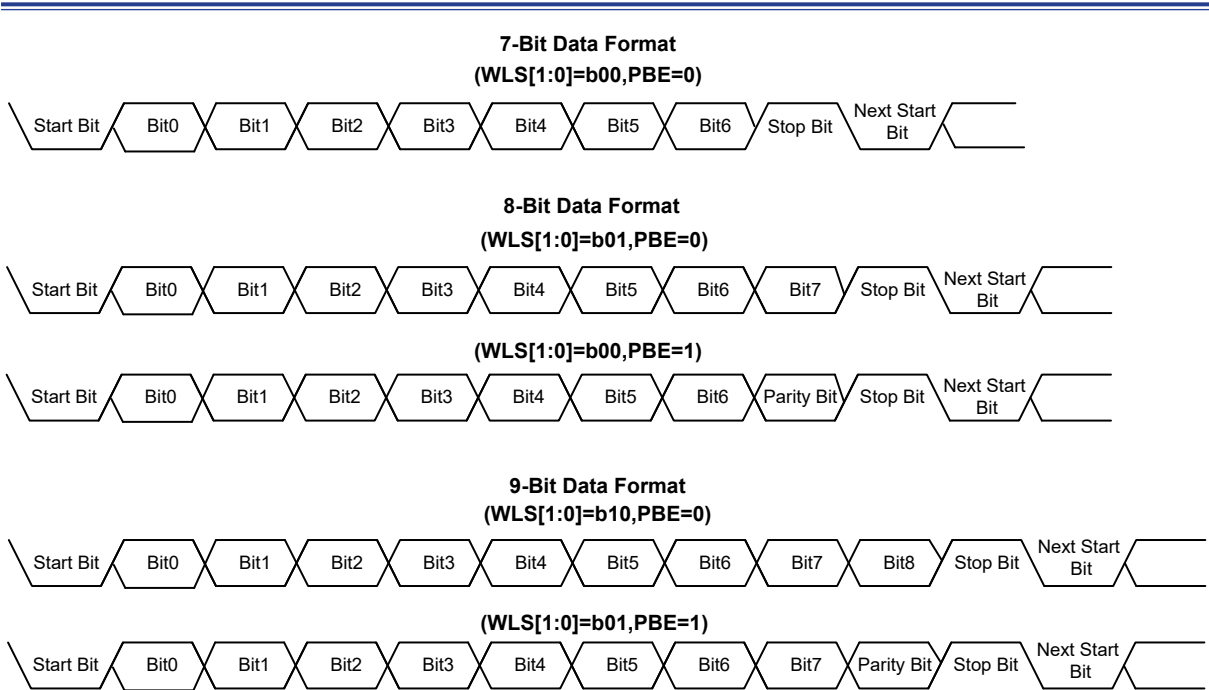


图 151. USART 串行数据格式

波特率发生器

USART 接收器和发送器的波特率都设置为相同值。波特率分频器 BRD 与 USART 时钟 CK\_USART 的关系如下。

波特率时钟 = CK\_USART / BRD

CK\_USART 时钟是连接到 USART 模块的 APB 时钟，而在异步模式中 BRD 的范围是 16 ~ 65535，在同步模式中 BRD 的范围则为 8 ~ 65535。

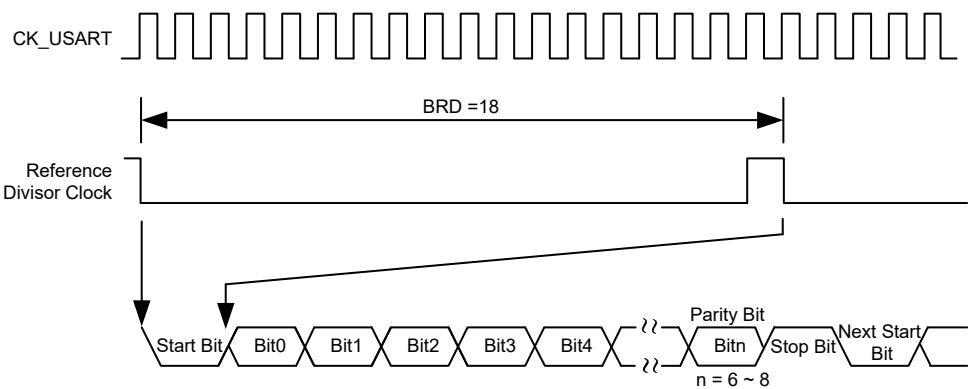


图 152. USART 时钟 CK\_USART 和数据帧时序

表 53. 波特率误差计算 – CK\_USART = 48 MHz

波特率		CK_USART = 48 MHz		
No.	Kbps	实际值	BRD	误差率
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000	16	0.00%

表 54. 波特率误差计算 – CK\_USART = 96 MHz

波特率		CK_USART = 96 MHz		
No.	Kbps	实际值	BRD	误差率
1	2.4	2.4	40000	0.00%
2	9.6	9.6	10000	0.00%
3	19.2	19.2	5000	0.00%
4	57.6	57.6	1667	-0.02%
5	115.2	115.2	833	0.04%
6	230.4	230.2	417	-0.08%
7	460.8	461.5	208	0.16%
8	921.6	923.1	104	0.16%
9	2250	2232.6	43	-0.78%
10	3000	3000	32	0.00%

硬件流控制

USART 支持硬件流控制功能，通过 USRCR 寄存器中的 HFCEN 位置 1 使能。可通过 CTS 输入和 RTS 输出控制 2 个 USART 设备之间的串行数据流。图 153 显示了该模式下的连接图。硬件流控制功能分为两种类型，一个是 RTS 流控制功能，一个是 CTS 流控制功能。

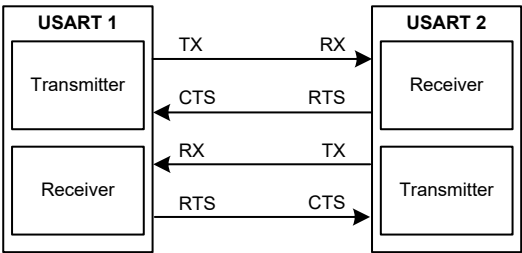


图 153. 两个 USART 之间的硬件流控制

### RTS 流控制功能

在 RTS 流控制中, 当接收数据寄存器为空时, USART RTS 引脚状态为有效电平 (低电平), 表示接收器已做好接收新数据的准备。当 RX FIFO 达到由 USRFCR 寄存器中 RXTL 字段定义的触发级别时, USART RTS 引脚状态为无效电平 (高电平)。图 154 显示了一个 RTS 流控制的范例。

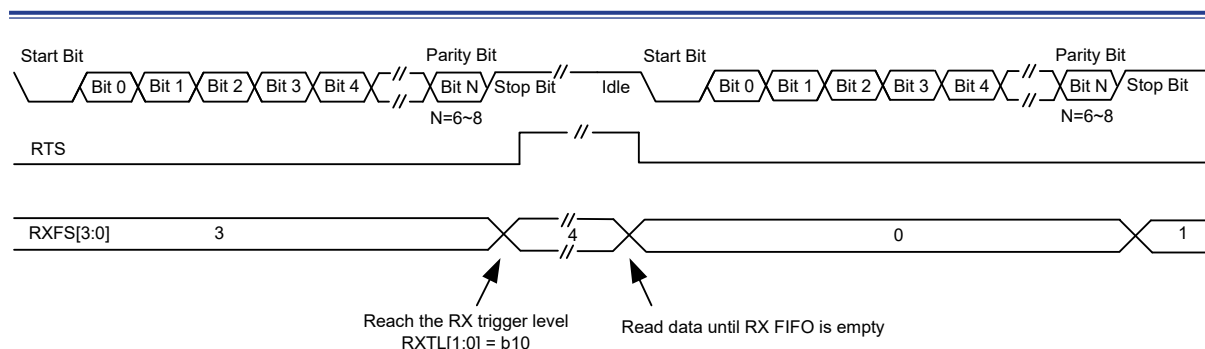


图 154. USART RTS 流控制

### CTS 流控制功能

如果硬件流控制功能使能, USRCR 寄存器中的 URTXEN 位将由 USART CTS 输入信号来控制。如果 USART CTS 引脚强制为逻辑低状态, 则 URTXEN 位将自动置 1 使能数据发送。但是, 如果 USART CTS 引脚强制为逻辑高状态, 则 URTXEN 位将清零除能数据发送。

当 USART CTS 引脚在数据发送过程中强制为逻辑高状态, 则当前数据将继续发送直到停止位发送完成。图 155 显示了一个 CTS 流控制通信的范例。

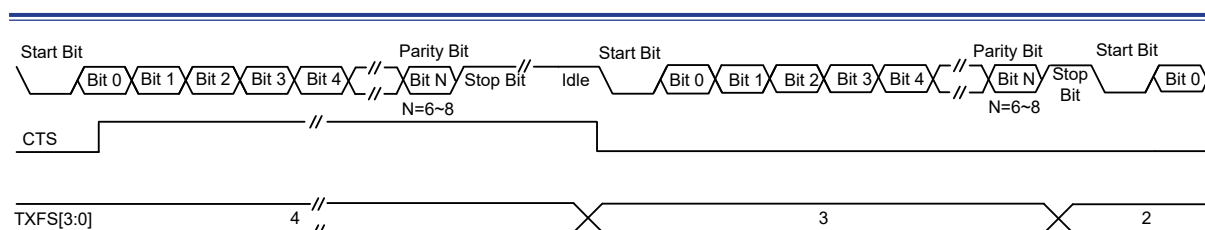


图 155. USART CTS 流控制

## IrDA 模式

USART IrDA 模式提供半双工点对点无线通信。

USART 模块包括一个集成的调制器和解调器，它允许使用红外线收发器进行无线通信。在 IrDA 模式中，发送器指定一个逻辑数据“0”作为一个“高”脉冲，一个逻辑数据“1”作为一个“低”电平，而接收器把一个“低脉冲”解读为逻辑数据“0”，把一个“高”电平解读为逻辑数据“1”。

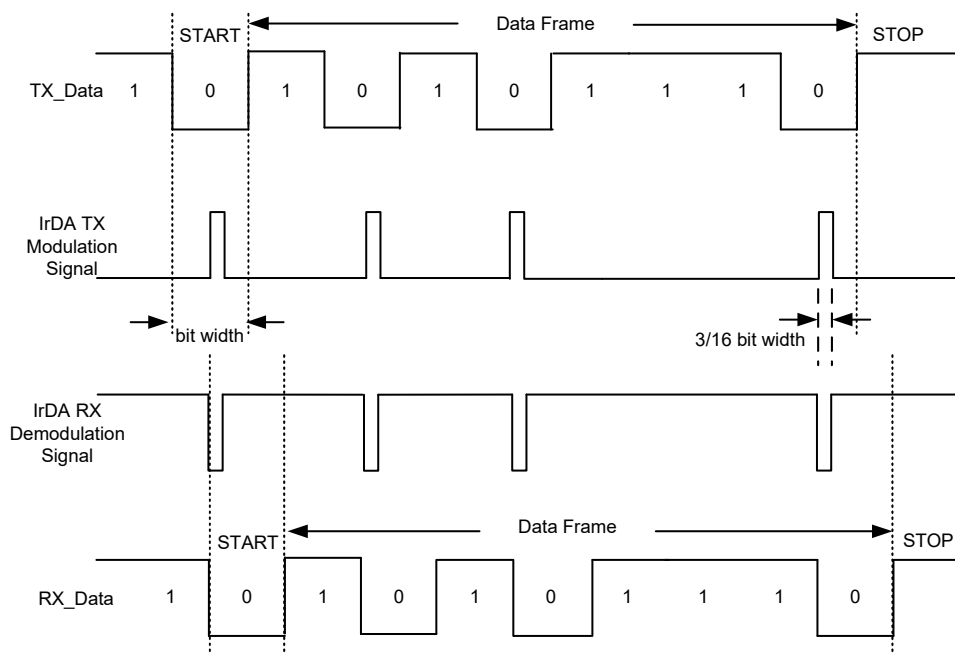


图 156. IrDA 调制与解调

IrDA 模式提供两种工作模式，一种是正常模式，另一种是低功耗模式。

### IrDA 正常模式

在 IrDA 正常模式里, 由发送调制器生成的每个发射脉冲的宽度被定义为波特率时钟周期的 3/16。IrDA 接收器解调器接收到的脉冲宽度取决于 IrDA 接收去抖滤波器, 它是由一个 8-bit 向下计数的计数器定义的。去抖滤波器的计数器的值是由 IrDACR 寄存器的 IrDAPSC 字段定义的。当在接收器引脚检测到一个下降沿时, 去抖滤波器计数器在 CK\_USART 时钟驱动下开始向下计数。当在接收器引脚检测到一个上升沿时, 计数器停止计数, 并重新载入 IrDAPSC 值。当在接收器引脚检测到一个低脉冲下降沿, 且在去抖滤波器计数到零之前检测到上升沿时, 则此低脉冲将被视为干扰噪声, 被丢弃。当在接收器引脚检测到一个低脉冲下降沿, 但在去抖滤波器计数到零之前没有检测到上升沿时, 则在此位的持续时间内, USART 接收器引脚上的输入信号将被视为一个有效数据“0”。IrDAPSC 值必须设置为大于或等于 0x01, 这样 IrDA 接收器解调操作才可以正常进行。可以调整 IrDAPSC 值以符合 USART 波特率的设置, 从而将 IrDA 接收到的宽度小于预分频器设置时间的干扰噪声过滤掉。

### IrDA 低功耗模式

在 IrDA 低功耗模式中, 发送器调制器产生的传输脉冲宽度不是保持在波特率时钟周期的 3/16, 而是固定值, 由下列公式计算。设置 IrDAPSC 字段来调整发送的脉冲宽度以符合外部 IrDA 接收器的最低脉冲宽度规格。

$$T_{\text{IrDA}_L} = 3 \times \text{IrDAPSC} / \text{CK\_USART}$$

注:  $T_{\text{IrDA}_L}$  是在低功耗模式下传输的 IrDA 脉冲宽度。

IrDA 控制寄存器 IrDACR 中的 IrDAPSC 字段是 IrDA 分频器的值。

在 IrDA 低功耗接收模式下的去抖操作类似于 IrDA 正常模式下的操作。对于干扰检测, 在 IrDA 接收器解调过程中, 脉冲宽度小于  $1 \times (\text{IrDAPSC} / \text{CK\_USART})$  的低脉冲将被丢弃, 而低脉冲宽度大于  $2 \times (\text{IrDAPSC} / \text{CK\_USART})$  的信号被视为有效数据可接收。

IrDA 物理层规范规定发送和接收之间最小要有 10 ms 的延时, IrDA 接收器的建立时间应该由软件管理。

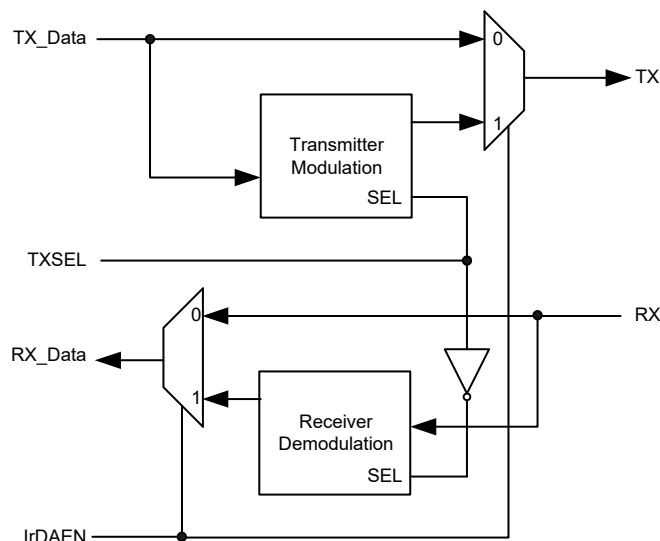


图 157. USART I/O 和 IrDA 方框图

## RS485 模式

在 USART 的 RS485 模式中，接口上的数据通过 2 线式双绞线传输。RS485 收发器通过参照第三方共模参考电压值来解析差分信号的电压电平。如果没有这个共模参考，收发器可能会错误地解析差分信号。这增强了 RS485 接口的抗干扰能力。当 USART 模块工作在 RS485 模式下时，USART RTS 引脚用来控制外部 RS485 收发器，此收发器的极性可以由 RS485 控制寄存器 RS485CR 中的 TXENP 位来配置。

### RS485 自动方向控制模式 – AUD

当 RS485 模式被配置为主机发送器时，其将工作在自动方向控制模式，即 AUD 模式。在 AUD 模式时，RS485 模式下，USART RTS 引脚的极性根据 RS485 控制寄存器的 TXENP 位来配置。此引脚用来控制外部 RS485 收发器使能发送器。



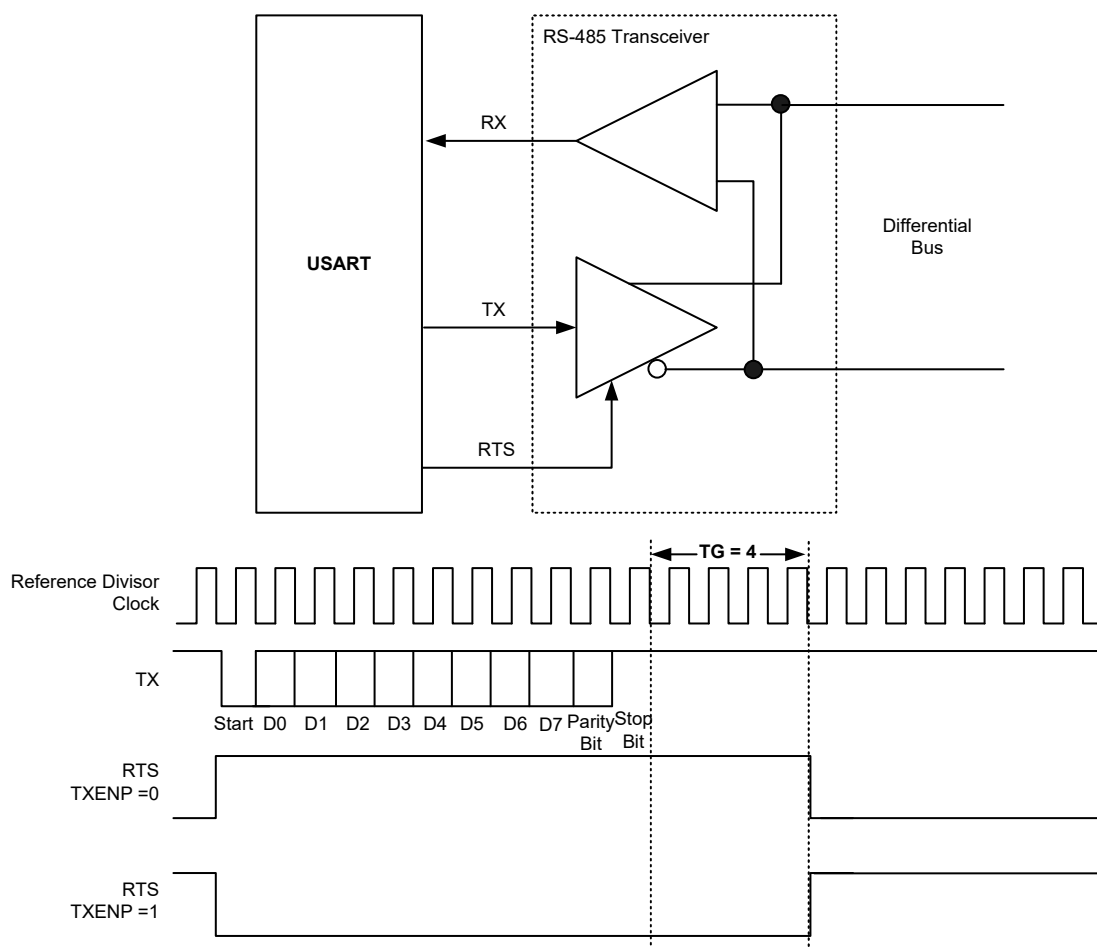


图 158. RS485 接口和波形

### RS485 正常多点通信工作模式 – NMM

当 RS485 接口配置为可寻址从机时,可工作在正常多点通信工作模式,即 NMM。当 RS485CR 寄存器中的 RSNMM 位被置位时,此模式使能。无论 USRCR 寄存器中的 URRXEN 为何值,所有接收到的校验位为“0”的数据都将被忽略,直到检测到第一个带有校验位“1”的地址字节,这时,接收到的地址字节将存入到 RX FIFO 中。一旦检测到第一个字节数据并存入 RX FIFO 时,USRSIFR 寄存器中的 RSADD 标志位将被置位,如果 USRIER 寄存器中的 RSADDIE 位置 1,将产生中断。通过 URRXEN 位决定接收器是否使能来接收后面的数据。当接收器通过 URRXEN 位置 1 使能时,所有接收到的数据都将被存入 RX FIFO,否则,接收器通过 URRXEN 位清零除能时,这些数据将被忽略。

### RS485 自动地址检测工作模式 – AAD

当 RS485 接口配置为可寻址从机时,除了正常多点通信工作模式外,RS485 接口还可以工作在自动地址检测工作模式,即 AAD 模式。此模式由 RS485CR 寄存器中的 RSAAD 位置 1 使能。接收器检测到带有校验位“1”的地址帧后,和在 RS485CR 寄存器中 ADDMATCH 字段定义的 8 位地址值相比较。如果地址数据和 ADDMATCH 值匹配,则将数据存储在 RX FIFO 中并自动置位 URRXEN。当接收器使能,所有接收到的数据都将存储在 RX FIFO 中,直到地址帧和 ADDMATCH 值不匹配的情况发生为止,这时,接收器将自动除能。接收器使能后,清除 URRXEN 位为“0”,可使接收器除能。

## 同步主机模式

USART 同步主机模式中,用全双工方式传输数据,即数据发送和接收同时发生,且只支持主机模式。USART CTS 引脚是同步 USART 发送器的时钟输出脚。在这种模式下,起始位、校验位和停止位持续时间内,没有时钟脉冲被送到 CTS 引脚。同步控制寄存器 SYNCR 中的 CPS 位可以用于决定是在第一个还是第二个时钟脉冲边沿捕捉数据。SYNCR 寄存器中的 CPO 位用来配置 USART 同步模式空闲状态时钟的极性。具体的时序信息如图 160 所示。

在 USART 同步模式下,USART CTS/SCK 时钟输出引脚仅用于发送数据到从机。如果是有效的数据被写入发送数据寄存器 USRDR,USART 同步模式下将自动与相应的时钟输出同步发送数据,并且 USART 接收器也将接收 RX 引脚上的数据。否则,如果没有数据被发送,接收器将无法获得同步数据。

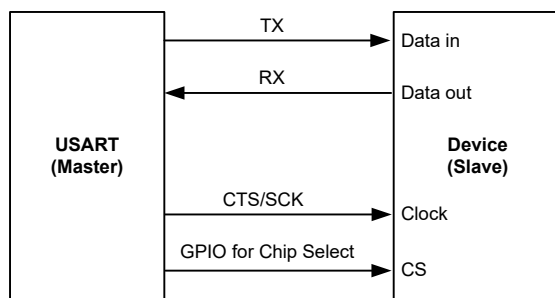


图 159. USART 同步传输范例

注: USART 仅支持同步主机模式。它不能接收或发送与输入时钟同步的数据。USART CTS / SCK 时钟始终是一个输出信号。

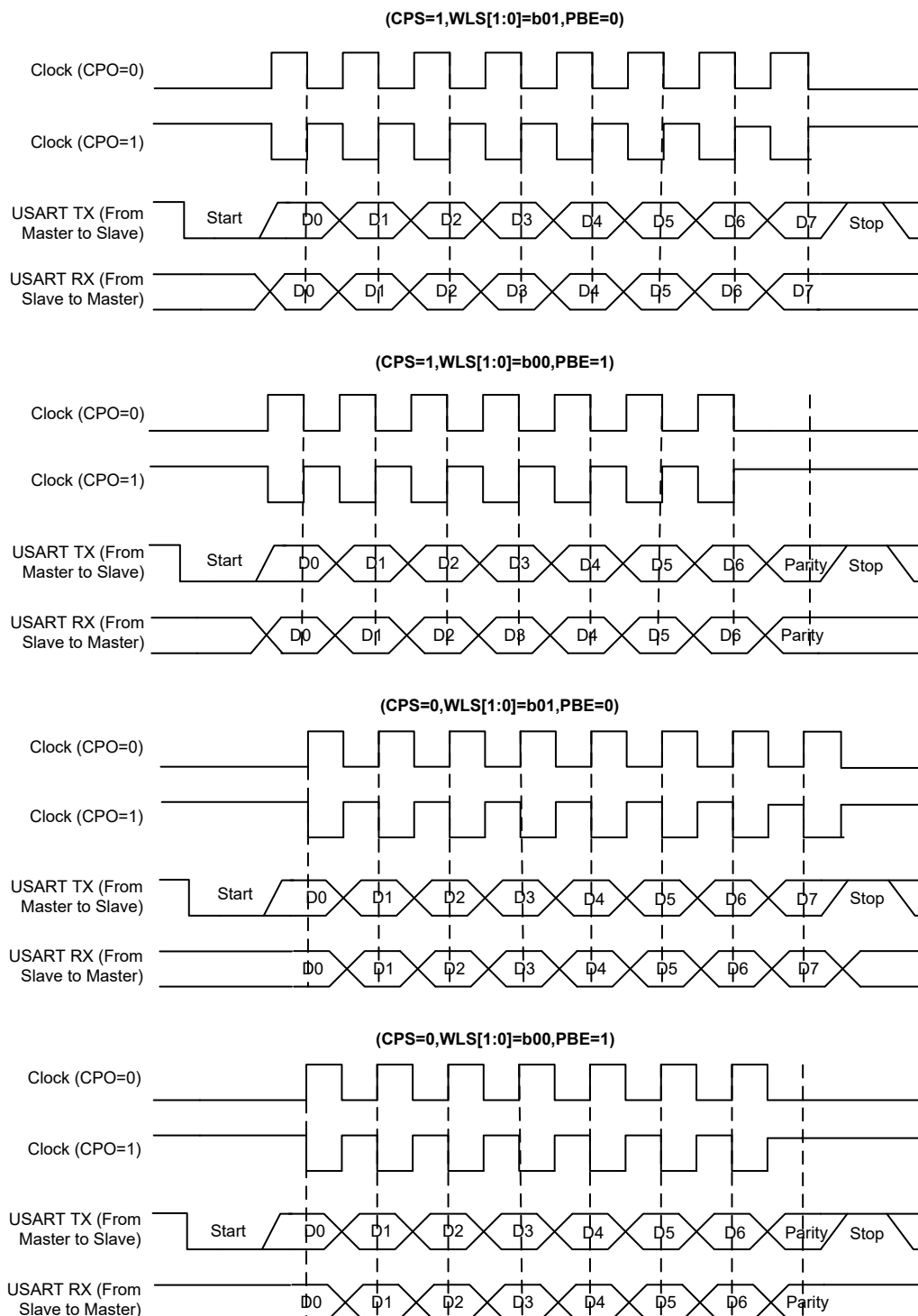


图 160. 8-bit 格式 USART 同步波形

## 中断和状态

下列事件发生时，若相应的中断使能位被置位，USART 模块可产生一个中断：

- 接收器 FIFO 超时中断：USART 接收器 FIFO 非空且在指定的超时间隔内未收到一个新的数据包。
- 接收器线路状态中断：USART 接收器溢出错误、奇偶错误、帧错误或暂停事件发生。
- 发送器 FIFO 阈值级别中断：USART 发送器 FIFO 中要传输的数据量少于指定的阈值级别。
- 发送完成中断：发送 FIFO 为空且发送移位寄存器 TSR 的内容已全部移出。
- 接收器 FIFO 阈值级别中断：FIFO 接收的数据量已达到指定的阈值级别。

## PDMA 接口

PDMA 接口集成在 USART 模块中。在发送器模式或接收器模式下分别通过设置 USRCR 寄存器中的 TXDMAEN 或 RXDMAEN 位为 1 可使能 PDMA 功能。当 USART 发送器 FIFO 中要被发送的数据量少于 USRFCR 寄存器中 TXTL 字段定义的 TX FIFO 的阈值级别且 TXDMAEN 位被置 1 时，PDMA 功能使能，将数据从特定的存储器地址移到 USART TX FIFO 中。

同样地，当接收器 FIFO 中的被接收到的数据量等于 USRFCR 寄存器 RXTL 字段定义的 RX FIFO 阈值级别且 RXDMAEN 位被置 1 时，PDMA 功能使能，将数据从 USART RX FIFO 移到特定的存储器地址中。详细的 PDMA 配置信息，请参考 PDMA 章节。

## 寄存器列表

下表显示了 USART 寄存器及其复位值。

表 55. USART 寄存器列表

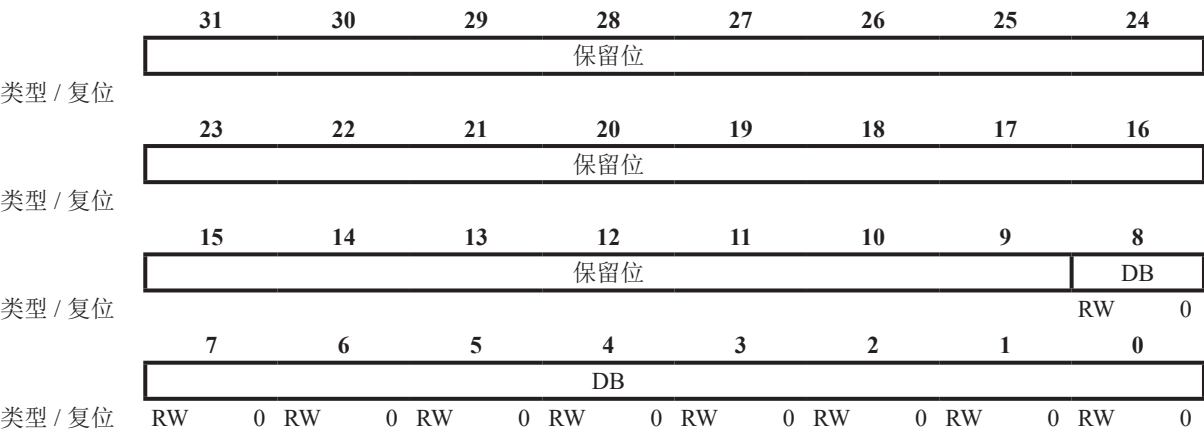
寄存器	偏移量	描述	复位值
USRDR	0x000	USART 数据寄存器	0x0000_0000
USRCR	0x004	USART 控制寄存器	0x0000_0000
USRFCR	0x008	USART FIFO 控制寄存器	0x0000_0000
USRIER	0x00C	USART 中断使能寄存器	0x0000_0000
USRSIFR	0x010	USART 状态 & 中断标志位寄存器	0x0000_0980
USRTPR	0x014	USART 时序参数寄存器	0x0000_0000
IrDACR	0x018	USART IrDA 控制寄存器	0x0000_0000
RS485CR	0x01C	USART RS485 控制寄存器	0x0000_0000
SYNCR	0x020	USART 同步控制寄存器	0x0000_0000
USRDLR	0x024	USART 分频器锁存寄存器	0x0000_0010
USRTSTR	0x028	USART 测试寄存器	0x0000_0000

寄存器描述

USART 数据寄存器 – USRDR

该寄存器用来存取 USART 发送与接收到的 FIFO 数据。

偏移量: 0x000  
复位值: 0x0000\_0000



位	字段	描述
[8:0]	DB	通过此接收器缓冲寄存器读取数据，将返回接收器 FIFO 中的数据。接收器 FIFO 最大容量为 8 × 9 位。通过读取该寄存器，USART 将返回 7、8 或 9 位接收到的数据。DB[8] 位仅在 9-bit 模式下有效，在 8-bit 模式下固定为 0。在 7-bit 模式下，DB[6:0] 位为有效位。 写数据到此发送器缓冲寄存器将会把数据加载到发送器 FIFO 中。发送器 FIFO 的最大容量为 8 × 9 位。通过写入该寄存器，USART 将送出 7、8 或 9 位发送的数据。DB[8] 位只有在 9-bit 模式下有效，在 8-bit 模式下将被忽略。在 7-bit 模式下，DB[6:0] 位为有效位。

## USART 控制寄存器 – USRCR

该寄存器用于定义串口参数，如 USART 的数据长度、奇偶和停止位。该寄存器也包含了 USART 使能控制位和 USART 模式与数据传输模式选择。

偏移量： 0x004

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	RTS	BCB	SPE	EPE	PBE	NSB	WLS	
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	RXDMAEN	TXDMAEN	URRXEN	URTXEN	HFCEN	TRSM	MODE	
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15]	RTS	RTS – Request-To-Send 信号 0: 驱动 USART RTS 引脚为逻辑 1 1: 驱动 USART RTS 引脚为逻辑 0 注意，当 HFCEN 位被复位，该位用于控制 USART RTS 引脚状态。 当 HFCEN 位被置位时，RTS 位用来显示 RTS 引脚状态，此时该引脚状态由硬件流控制功能控制。
[14]	BCB	暂停控制位 当此位被设置为 1，USART TX 引脚上的串行数据输出将被迫进入空白状态 (逻辑 0)。该位仅作用于 USART TX 输出脚，对发送器逻辑无影响。
[13]	SPE	强制奇偶校验使能 0: 除能强制奇偶校验 1: 使能强制奇偶校验 当 PBE 位设置为 1 时，此位才可用。如果 PBE 和 SPE 位都被设置为 1，并且 EPE 位被清零，则被发送的奇偶校验位将被设置为 1。当 PBE 和 SPE 位被设置为 1 并且 EPE 位也被设置为 1，则被传送的奇偶校验位将被清零。
[12]	EPE	偶校验使能 0: 在数据字和奇偶校验位中共有奇数个逻辑“1”被发送或被检测到 1: 在数据字和奇偶校验位中共有偶数个逻辑“1”被发送或被检测到 当 PBE 位设置为 1 时，此位才可用。
[11]	PBE	奇偶校验位使能 0: 在传输过程中，奇偶校验位不会生成 (发送数据) 或被检查 (接收数据) 1: 在传输过程中，奇偶校验位生成或被检查 注：当 WLS 字段设置为“10”来选择 9-bit 数据格式时，对 PBE 位的设置无效。
[10]	NSB	停止位的个数 0: 为传输的数据生成一个停止位 1: 选择 8 位或 9 位字长时，生成两个停止位

位	字段	描述
[9:8]	WLS	字长选择 00: 7 位 01: 8 位 10: 9 位 11: 保留位
[7]	RXDMAEN	USART RX DMA 使能位 0: 除能 1: 使能
[6]	TXDMAEN	USART TX DMA 使能位 0: 除能 1: 使能
[5]	URRXEN	USART RX 使能 0: 除能 1: 使能
[4]	URTXEN	USART TX 使能 0: 除能 1: 使能
[3]	HFCEN	硬件流控制功能使能位 0: 除能 1: 使能
[2]	TRSM	传输模式选择 此位用于选择数据传输协议。 0: LSB 优先 1: MSB 优先
[1:0]	MODE	USART 模式选择 00: 正常模式 01: IrDA 10: RS485 11: 同步

USART FIFO 控制寄存器 – USRFCR

此寄存器定义了包括阈值级别和复位功能以及 USART FIFO 状态的 USART FIFO 控制与配置。

偏移量： 0x008  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位				RXFS			
类型 / 复位					RO	0	RO	0
	23	22	21	20	19	18	17	16
	保留位				TXFS			
类型 / 复位					RO	0	RO	0
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	RXTL		TXTL		保留位		RXR	TXR
类型 / 复位	RW	0	RW	0	RW	0	WO	0

位	字段	描述
[27:24]	RXFS	RX FIFO 状态 RXFS 字段显示了在 RX FIFO 中当前数据的总数。 0000: RX FIFO 为空 0001: RX FIFO 包含 1 个数据 ... 1000: RX FIFO 包含 8 个数据 其它: 保留
[19:16]	TXFS	TX FIFO 状态 TXFS 字段显示了在 TX FIFO 中当前数据的总数。 0000: TX FIFO 为空 0001: TX FIFO 包含 1 个数据 ... 1000: TX FIFO 包含 8 个数据 其它: 保留
[7:6]	RXTL	RX FIFO 阈值级别设定 00: 1 字节 01: 2 字节 10: 4 字节 11: 6 字节 RXTL 字段定义 RX FIFO 触发级别。
[5:4]	TXTL	TX FIFO 阈值级别设定 00: 0 字节 01: 2 字节 10: 4 字节 11: 6 字节 TXTL 字段定义 TX FIFO 触发级别。
[1]	RXR	RX FIFO 复位 此位置位将产生一个复位脉冲以复位 RX FIFO, RX FIFO 将被清空, 即在复位信号之后 RX 指针将复位至 0。在复位脉冲产生后此位自动返回 0。



位	字段	描述
[0]	TXR	TX FIFO 复位 此位置位将产生一个复位脉冲以复位 TX FIFO，TX FIFO 将被清空，即在复位信号之后 TX 指针将复位至 0。在复位脉冲产生后此位自动返回 0。

USART 中断使能寄存器 – USRIER

此寄存器用于使能相关 USART 中断功能。当相应的时间发生且相应的中断使能位置位时，USART 模块产生中断。

偏移量: 0x00C  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24		
	保留位									
类型 / 复位										
	23	22	21	20	19	18	17	16		
	保留位									
类型 / 复位										
	15	14	13	12	11	10	9	8		
	保留位						CTSIE		RXTOIE	
类型 / 复位							RW	0	RW	0
	7	6	5	4	3	2	1	0		
	RSADDIE	BIE	FEIE	PEIE	OEIE	TXCIE	TXDEIE	RXDRIE		
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[9]	CTSIE	CTS – Clear-To-Send 中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 CTSC 位置位时将产生中断。
[8]	RXTOIE	接收器 FIFO 超时中断使能 0: 除能中断 1: 使能中断 接收器 FIFO 超时中断意味着在由 RXTOC 字段所定义的超时持续期间接收器 FIFO 未空且无动作。
[7]	RSADDIE	RS485 地址检测中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 RSADD 位置位时将产生中断。
[6]	BIE	暂停中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 BII 位置位时将产生中断。
[5]	FEIE	帧错误中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 FEI 位置位时将产生中断。

位	字段	描述
[4]	PEIE	奇偶错误中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 PEI 位置位时将产生中断。
[3]	OEIE	溢出错误中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 OEI 位置位时将产生中断。
[2]	TXCIE	发送完成中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 TXC 位置位时将产生中断。
[1]	TXDEIE	发送数据空中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 TXDE 位置位时将产生中断。
[0]	RXDRIE	接收数据就绪中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 USRSIFR 寄存器中的 RXDR 位置位时将产生中断。

## USART 状态 & 中断标志位寄存器 – USRSIFR

该寄存器包含了相应的 USART 状态。

偏移量: 0x010

复位值: 0x0000\_0980

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位				CTSS	CTSC	RSADD	TXC
类型 / 复位					RO	1 WC	0 WC	0 RO
	7	6	5	4	3	2	1	0
	TXDE	RXTOF	RXDR	BII	FEI	PEI	OEI	RXDNE
类型 / 复位	RO	1 WC	0 RO	0 WC	0 WC	0 WC	0 WC	0 RO

位	字段	描述
[11]	CTSS	CTS – Clear-To-Send 状态 0: CTS 引脚无效 1: CTS 引脚有效且保持在逻辑低状态
[10]	CTSC	CTS 状态改变标志位 CTS 输入引脚状态一旦改变, 此位将被置位。若 USRIER 寄存器中的 CTSIE = 1, 则将产生中断。对此位写 1 将清除此标志位。
[9]	RSADD	RS485 地址检测 0: 地址未检测 1: 地址已检测 当接收器检测到地址时此位置位。若 USRIER 寄存器中的 RSADDIE = 1, 则将产生中断。对此位写 1 将清除此标志位。 注: 此位仅用于 RS485 模式, 通过 USRCR 寄存器中的 MODE[1:0] 设置。
[8]	TXC	发送完成 0: 发送器 FIFO (TX FIFO) 或发送移位寄存器 (TSR) 都不为空 1: TX FIFO 与 TSR 寄存器均为空 若 USRIER 寄存器中的 TXCIE = 1, 则将产生中断。向 USRDR 寄存器写入一个新数据将清除此标志位。
[7]	TXDE	发送器数据 FIFO 为空 0: TX FIFO 级别高于阈值 1: TX FIFO 级别等于或低于阈值 当发送器 FIFO 级别等于或低于阈值级别时 TXDE 位将置位, 阈值由 USRFCR 寄存器中的 TXTL 字段设置。当新数据写入 USRDR 寄存器且 TX FIFO 级别高于阈值设定时此位清零。
[6]	RXTOF	接收器 FIFO 超时标志位 0: RX FIFO 超时未发生 1: RX FIFO 超时发生 若 RX FIFO 为非空, 并且在 RXTOC 字段指定的超时期间 RX FIFO 中没有发生动作, 则 RXTOF 将被置位。如果发生了 RX FIFO 超时的情况, 则必须在读取 RX FIFO 之前清除此标志位。对此位写 1 将清除此标志位。

位	字段	描述
[5]	RXDR	接收器 FIFO 就绪标志位 0: RX FIFO 级别低于阈值 1: RX FIFO 级别等于或高于阈值 当 FIFO 接收数据量达到指定的阈值级别时 RXDR 位将置位，阈值由 USRFCR 寄存器中的 RXTL 字段设置。当从 USRDR 寄存器中读取数据且 RX FIFO 级别低于阈值时此位清零。
[4]	BII	暂停中断指示位 当接收数据输入保持为“空白状态”(逻辑 0)的时间长于一个完整字传输所用时间，此位将置位。完整字传输时间包括起始位、数据位、奇偶和停止位的总持续时间。对此位写 1 将清除此标志位。
[3]	FEI	帧错误指示位 接收字符没有有效停止位，即检测到最后一个数字位或奇偶位后面的停止位为逻辑 0 时，此位置位。对此位写 1 将清除此标志位。
[2]	PEI	奇偶错误指示位 接收字符没有有效奇偶位时，此位置位。对此位写 1 将清除此标志位。
[1]	OEI	溢出错误指示位 在 RX FIFO 已满，且 RX 移位寄存器又收到一个新的字符，将发生溢出错误。发生溢出错误事件之后，若 RX 移位寄存器又接收到新字符，移位寄存器中的数据将被覆盖，而 RX FIFO 中的数据不会被覆盖。OEI 位用于溢出事件发生时的即时显示。对此位写 1 将清除此标志位。
[0]	RXDNE	RX FIFO 数据未空 0: RX FIFO 为空 1: RX FIFO 中至少有一个接收数据字

USART 时序参数寄存器 – USRTPR

该寄存器包含了 USART 的时序参数，包括发送器时间保护参数和接收器 FIFO 超时值以及 RX FIFO 超时功能使能控制。

偏移量:	0x014
复位值:	0x0000_0000
	3130292827262524
类型 / 复位	保留位
	2322212019181716
类型 / 复位	保留位
	15141312111098
类型 / 复位	TG
	RW0RW0RW0RW0RW0RW0RW0RW0RW0RW0RW0RW0
	76543210
类型 / 复位	RXTOENRXTOC
	RW0RW0RW0RW0RW0RW0RW0RW0RW0RW0RW0RW0

位	字段	描述
[15:8]	TG	发送器时间保护 发送器时间保护计数器由波特率时钟驱动。当 TX FIFO 发送数据时，计数器被复位，然后开始计数。只有当计数器的内容等于 TG 值时，才允许执行进一步的字发送。
[7]	RXTOEN	接收器 FIFO 超时计数器使能 0: 接收器 FIFO 超时计数器除能 1: 接收器 FIFO 超时计数器使能
[6:0]	RXTOC	接收器 FIFO 超时计数器比较值 RX FIFO 超时计数器由波特率时钟驱动。当 RX FIFO 接收到新数据时，计数器被复位，然后开始计数。一旦计数器的内容等于超时计数器比较值 RXTOC，且 USRIER 寄存器的 RXTOIE 位为 1 时，将产生接收器 FIFO 超时中断 RXTOI。新收到的数据或空的 RX FIFO 被读取后，将清除 RX FIFO 超时计数器。

## USART IrDA 控制寄存器 – IrDACR

该寄存器用于控制 USART 的 IrDA 模式。

偏移量: 0x018

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	IrDAPSC								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	保留位		RXINV	TXINV	LB	TXSEL	IrDALP	IrDAEN	
类型 / 复位			RW	0	RW	0	RW	0	RW

位	字段	描述
[15:8]	IrDAPSC	IrDA 预分频值 此字段包含 8-bit 去抖预分频值。 去抖向下计数器由 USART 时钟 CK_USART 驱动。计数周期由 IrDAPSC 字段定义。 IrDAPSC 字段必须设置为大于或等于 0x01 的值，使去抖计数器正常工作。如果脉冲宽度小于 IrDAPSC 字段定义的时间段，则脉冲将被视为干扰噪声并丢弃。 00000000: 保留 – 不可用 00000001: CK_USART 时钟 1 分频 00000010: CK_USART 时钟 2 分频 00000011: CK_USART 时钟 3 分频 ...
[5]	RXINV	RX 信号反转控制 0: 无反转 1: RX 输入信号反转
[4]	TXINV	TX 信号反转控制 0: 无反转 1: TX 输出信号反转
[3]	LB	IrDA 回送模式 0: 除能 IrDA 回送模式 1: 使能 IrDA 回送模式用于自检
[2]	TXSEL	收发器选择 0: 使能 IrDA 接收器 1: 使能 IrDA 发送器
[1]	IrDALP	IrDA 低功耗模式 选择 IrDA 工作模式。 0: 正常模式 1: IrDA 低功耗模式
[0]	IrDAEN	IrDA 使能控制 0: 除能 IrDA 模式 1: 使能 IrDA 模式

USART RS485 控制寄存器 – RS485CR

该寄存器用于控制 USART 的 RS485 模式。

偏移量： 0x01C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	ADDMATCH								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	保留位					RSAAD	RSNMM	TXENP	
类型 / 复位						RW	0	RW	0

位	字段	描述
[15:8]	ADDMATCH	RS485 自动地址匹配值 此字段是用于存储 RS485 自动地址检测工作模式的地址匹配值。
[2]	RSAAD	RS485 自动地址检测工作模式控制 0: 除能 1: 使能
[1]	RSNMM	RS485 正常多点通信工作模式控制 0: 除能 1: 使能
[0]	TXENP	USART RTS / TXE 引脚极性 0: 在 RS485 发送模式下 RTS / TXE 高电平有效 1: 在 RS485 发送模式下 RTS / TXE 低电平有效

## USART 同步控制寄存器 – SYNCR

该寄存器用于控制 USART 的同步模式。

偏移量: 0x020

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位				CPO	CPS	保留位	CLKEN
类型 / 复位				RW	0	RW	0

位	字段	描述
[3]	CPO	时钟极性 0: CTS / SCK 引脚空闲状态为低 1: CTS / SCK 引脚空闲状态为高 在同步模式下选择 USART CTS / SCK 引脚上的时钟输出极性。与 CPS 位共同定义所需的时钟空闲状态。
[2]	CPS	时钟相位 0: 数据在第一个时钟边沿被捕捉 1: 数据在第二个时钟边沿被捕捉 在同步模式下该位允许用户选择 USART CTS / SCK 引脚上的时钟输出相位。与 CPO 位共同定义数据捕捉边沿。
[0]	CLKEN	时钟使能 0: CTS / SCK 引脚除能 1: CTS / SCK 引脚使能 使能 / 除能 USART CTS / SCK 引脚。



## USART 分频器锁存寄存器 – USRDLR

该寄存器用来定义 USART 时钟分频比，以产生适当的波特率。

偏移量： 0x024

复位值： 0x0000\_0010

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	BRD							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	BRD							
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	BRD	波特率分频器 这 16 位定义了 USART 时钟分频比例。 波特率 = CK_USART / BRD CK_USART 时钟是连接到 USART 模块的系统时钟。 在异步模式中 BRD = 16 ~ 65535; 在同步模式中 BRD = 8 ~ 65535。

## USART 测试寄存器 – USRTSTR

该寄存器控制 USART 调试模式。

偏移量: 0x028

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位						LBM	
类型 / 复位							RW	0 RW 0

位	字段	描述
[1:0]	LBM	回送测试模式选择 00: 正常工作模式 01: 保留位 10: 自动回应模式 11: 回送模式

## 22 通用异步收发器 (UART)

### 简介

通用异步收发器 UART，提供了一个灵活的异步传输的全双工数据交换。UART 用来转换并行和串行接口之间的数据，通常也被用作 RS232 标准通信。UART 外设功能支持多种类型的中断。

UART 模块包含一个发送数据寄存器 TDR 和发送移位寄存器 TSR 以及一个接收数据寄存器 RDR 和接收移位寄存器 RSR。通过读取 UART 状态 & 中断标志位寄存器 URSIFR，软件可检测 UART 的错误状态。这些状态包括传输类型和状况以及因奇偶、溢出、帧和暂停事件造成的错误状况。

UART 有一个可编程的波特率发生器，能对 CK\_APB (CK\_UART) 的 UART 时钟进行分频以产生 UART 发送器和接收器所需的时钟。

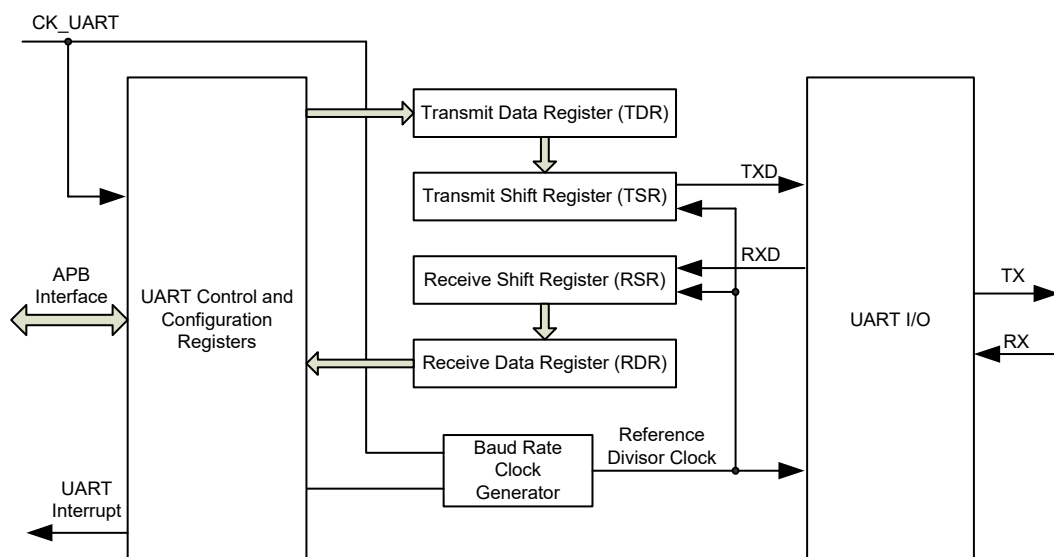


图 161. UART 方框图

特性

- 支持异步串行通信模式
- 全双工通信能力
- 可编程波特率时钟频率高达 (f<sub>PCLK</sub> / 16) MHz
- 完全可编程串行通信功能包括：
  - 字长：7、8 或 9-bit 字符
  - 校验位：偶、奇或无奇偶校验位的产生和检测
  - 停止位：1 或 2 个停止位
  - 位顺序：LSB 优先或 MSB 优先传输
- 错误检测：奇偶、溢出和帧错误
- 支持 PDMA 接口

功能描述

串行数据格式

UART 模块对写入发送数据寄存器中的数据进行并行到串行的转换，然后发送具有以下格式的数据：起始位、7~9 个 LSB/MSB 优先的数据位、可选的奇偶校验位和最后 1~2 个停止位。起始位的极性与数据线空闲状态相反。停止位与数据线空闲状态相同，并在下一步开始情况发生之前提供延迟。开始和停止位都用于异步数据传输过程中的数据同步。

UART 模块对从接收数据寄存器中读取的数据进行串行到并行的转换。它会首先检查校验位，然后将寻找一个停止位。如果停止位没有找到，UART 模块会认为整个字的传输失败并响应帧错误。

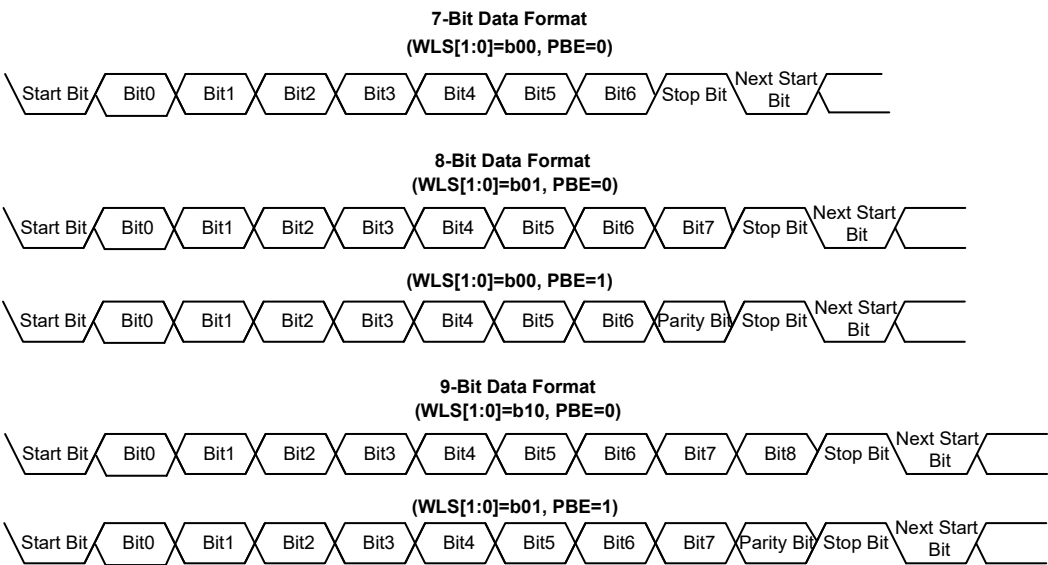


图 162. UART 串行数据格式

## 波特率发生器

UART 接收器和发送器的波特率都设置为相同值。波特率分频器 BRD 与 UART 时钟 CK\_UART 的关系如下。

$$\text{波特率时钟} = \text{CK\_UART} / \text{BRD}$$

CK\_UART 时钟是连接到 UART 模块的 APB 时钟, BRD 的范围是 16 ~ 65535。

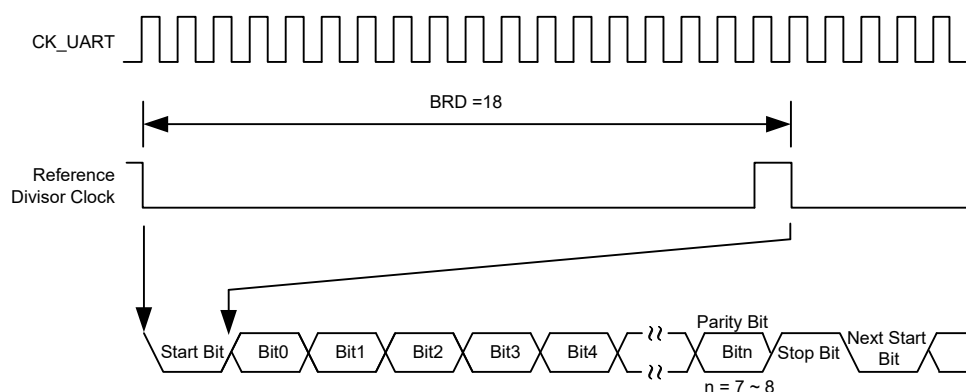


图 163. UART 时钟 CK\_UART 和数据帧时序

表 56. 波特率误差计算 – CK\_UART = 48 MHz

波特率		CK_UART = 48 MHz		
No.	Kbps	实际值	BRD	误差率
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000	16	0.00%

表 57. 波特率误差计算 – CK\_UART = 96 MHz

波特率		CK_UART = 96 MHz		
No.	Kbps	实际值	BRD	误差率
1	2.4	2.4	40000	0.00%
2	9.6	9.6	10000	0.00%
3	19.2	19.2	5000	0.00%
4	57.6	57.6	1667	-0.02%
5	115.2	115.2	833	0.04%
6	230.4	230.2	417	-0.08%
7	460.8	461.5	208	0.16%
8	921.6	923.1	104	0.16%
9	2250	2232.6	43	-0.78%
10	3000	3000	32	0.00%

## 中断和状态

下列事件发生时，若相应的中断使能位被置位，UART 模块可产生一个中断：

- 接收器线路状态中断：UART 接收器溢出错误、奇偶错误、帧错误或暂停事件发生。
- 发送数据寄存器为空中断：发送数据寄存器的内容传输至发送移位寄存器 TSR。
- 发送完成中断：发送数据寄存器 TDR 为空且发送移位寄存器 TSR 的内容已全部移出。
- 接收数据就绪中断：接收移位寄存器 RDR 的内容已全部传入 URDR 寄存器且读取准备就绪。

## PDMA 接口

PDMA 接口集成在 UART 模块中。在发送器模式或接收器模式下分别通过设置 URCR 寄存器中的 TXDMAEN 或 RXDMAEN 位为 1 可使能 PDMA 功能。当 UART 发送数据寄存器 TDR 为空且 TXDMAEN 位被置 1 时，PDMA 功能使能，将数据从特定的存储器地址移到 UART 发送数据寄存器 TDR 中。

同样地，当接收的数据已进入接收数据寄存器 RDR 中且 RXDMAEN 位被置 1 时，PDMA 功能使能，将数据从 UART 接收数据寄存器 RDR 移到特定的存储器地址中。详细的 PDMA 配置信息，请参考 PDMA 章节。

寄存器列表

下表显示了 UART 寄存器及其复位值。

表 58. UART 寄存器列表

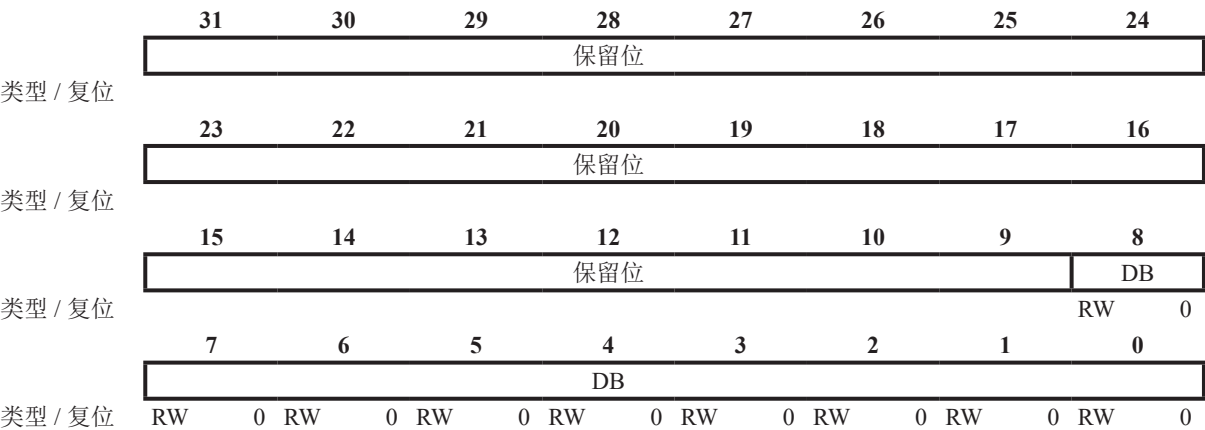
寄存器	偏移量	描述	复位值
URDR	0x000	UART 数据寄存器	0x0000_0000
URCR	0x004	UART 控制寄存器	0x0000_0000
URIER	0x00C	UART 中断使能寄存器	0x0000_0000
URSIFR	0x010	UART 状态 & 中断标志位寄存器	0x0000_0180
URDLR	0x024	UART 分频器锁存寄存器	0x0000_0010
URTSTR	0x028	UART 测试寄存器	0x0000_0000

寄存器描述

UART 数据寄存器 – URDR

该寄存器用来存储 UART 发送与接收到的数据。

偏移量: 0x000  
复位值: 0x0000\_0000



位	字段	描述
[8:0]	DB	通过读取该寄存器，UART 将返回 7、8 或 9 位接收到的数据。DB[8] 位仅在 9-bit 模式下有效，在 8-bit 模式下固定为 0。在 7-bit 模式下，寄存器的 DB[6:0] 包含可用位。通过写入该寄存器，UART 将送出 7、8 或 9 位发送的数据。DB[8] 位只有在 9-bit 模式下有效，在 8-bit 模式下将被忽略。在 7-bit 模式下，寄存器 DB[6:0] 包含可用位。

## UART 控制寄存器 – URCR

该寄存器用于定义串口参数，如 UART 的数据长度、奇偶和停止位。该寄存器也包含了 UART 使能控制位和 UART 模式与数据传输模式选择。

偏移量： 0x004

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24				
类型 / 复位	保留位											
	23	22	21	20	19	18	17	16				
类型 / 复位	保留位											
	15	14	13	12	11	10	9	8				
类型 / 复位	保留位	BCB	SPE	EPE	PBE	NSB	WLS					
	RW		0	RW		0	RW		0	RW		0
	7	6	5	4	3	2	1	0				
类型 / 复位	RXDMAEN	TXDMAEN	URRXEN	URTXEN	保留位	TRSM	保留位					
	RW		0	RW		0	RW		0			

位	字段	描述
[14]	BCB	暂停控制位 当此位被设置为 1，UART TX 引脚上的串行数据输出将被迫进入空白状态（逻辑 0）。该位仅作用于 UART TX 输出脚，对发送器逻辑无影响。
[13]	SPE	强制奇偶校验使能 0：除能强制奇偶校验 1：使能强制奇偶校验 当 PBE 位设置为 1 时，此位才可用。如果 PBE 和 SPE 位都被设置为 1，并且 EPE 位被清零，则被发送的奇偶校验位将被设置为 1。当 PBE 和 SPE 位被设置为 1 并且 EPE 位也被设置为 1，则被传送的奇偶校验位将被清零。
[12]	EPE	偶校验使能 0：在数据字和奇偶校验位中共有奇数个逻辑“1”被发送或被检测到 1：在数据字和奇偶校验位中共有偶数个逻辑“1”被发送或被检测到 当 PBE 位设置为 1 时，此位才可用。
[11]	PBE	奇偶校验位使能 0：在传输过程中，奇偶校验位不会生成（发送数据）或被检查（接收数据） 1：在传输过程中，奇偶校验位生成或被检查 注：当 WLS 字段设置为“10”来选择 9-bit 数据格式时，对 PBE 位的写操作没有任何影响。
[10]	NSB	停止位的个数 0：为传输的数据生成一个停止位 1：选择 8 位或 9 位字长时，生成两个停止位
[9:8]	WLS	字长选择 00：7 位 01：8 位 10：9 位 11：保留位
[7]	RXDMAEN	UART RX DMA 使能位 0：除能 1：使能

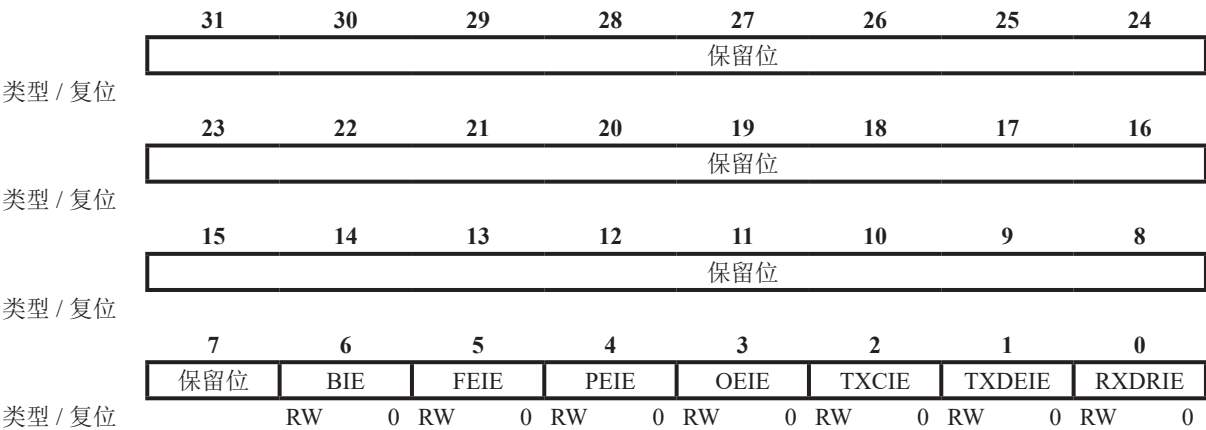


位	字段	描述
[6]	TXDMAEN	UART TX DMA 使能位 0: 除能 1: 使能
[5]	URRXEN	UART RX 使能 0: 除能 1: 使能
[4]	URTXEN	UART TX 使能 0: 除能 1: 使能
[2]	TRSM	传输模式选择 此位用于选择数据传输协议。 0: LSB 优先 1: MSB 优先

UART 中断使能寄存器 – URIER

此寄存器用于使能相关 UART 中断功能。当相应的事件发生且相应的中断使能位置位时，UART 模块产生中断。

偏移量： 0x00C  
复位值： 0x0000\_0000



位	字段	描述
[6]	BIE	暂停中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 URSIFR 寄存器中的 BII 位置位时将产生中断。
[5]	FEIE	帧错误中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 URSIFR 寄存器中的 FEI 位置位时将产生中断。
[4]	PEIE	奇偶错误中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 URSIFR 寄存器中的 PEI 位置位时将产生中断。
[3]	OEIE	溢出错误中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 URSIFR 寄存器中的 OEI 位置位时将产生中断。
[2]	TXCIE	发送完成中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 URSIFR 寄存器中的 TXC 位置位时将产生中断。
[1]	TXDEIE	发送数据空中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 URSIFR 寄存器中的 TXDE 位置位时将产生中断。
[0]	RXDRIE	接收数据就绪中断使能 0: 除能中断 1: 使能中断 如果该位置位，当 URSIFR 寄存器中的 RXDR 位置位时将产生中断。

## UART 状态 & 中断标志位寄存器 – URSIFR

该寄存器包含了相应的 UART 状态。

偏移量: 0x010

复位值: 0x0000\_0180

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	保留位							TXC	
类型 / 复位								RO	1
	7	6	5	4	3	2	1	0	
	TXDE	保留位	RXDR	BII	FEI	PEI	OEI	保留位	
类型 / 复位	RO	1	RO	0	WC	0	WC	0	WC

位	字段	描述
[8]	TXC	发送完成 0: 发送数据寄存器 TDR 或发送移位寄存器 TSR 都不为空 1: 发送数据寄存器 TDR 与发送移位寄存器 TSR 都为空 当发送数据寄存器 TDR 与发送移位寄存器 TSR 都为空。若 URIER 寄存器中的 TXCIE = 1, 则将产生中断。向 URDR 寄存器写入新数据将清除此标志位。
[7]	TXDE	发送数据寄存器为空 0: 发送数据寄存器不为空 1: 发送数据寄存器为空 当发送数据寄存器 TDR 的内容传输至发送移位寄存器 TSR 时 TXDE 位将置位。若 URIER 寄存器中的 TXEIE = 1 将产生中断, 向 URDR 寄存器写入一个新数据将清除此位。
[5]	RXDR	RX 数据就绪标志位 0: 接收数据寄存器为空 1: 读取接收数据寄存器中的所接收数据就绪 当接收移位寄存器 RSR 的内容传输至 URDR 寄存器时 RXDE 位将置位。当从 URDR 寄存器中读取数据时此位清零。若 URIER 寄存器中的 RXDRIE = 1 则产生中断。
[4]	BII	暂停中断指示位 当接收数据输入保持为“空白状态”(逻辑 0)的时间长于一个完整字传输所用时间, 此位将置位。完整字传输时间包括起始位、数据位、奇偶和停止位的总持续时间。对此位写 1 将清除此标志位。
[3]	FEI	帧错误指示位 接收字符没有有效停止位, 即检测到最后一个数字位或奇偶位后面的停止位为逻辑 0 时, 此位置位。对此位写 1 将清除此标志位。
[2]	PEI	奇偶错误指示位 接收字符没有有效奇偶位时, 此位置位。对此位写 1 将清除此标志位。

位	字段	描述
[1]	OEI	溢出错误指示位 当接收数据寄存器已满，且接收移位寄存器又收到一个新的字符，将发生溢出错误。 当发生溢出错误事件之后，若又接收到新字符，移位寄存器中的数据将会被覆盖，接收移位寄存器中的数据将不传送至接收数据寄存器。OEI 位用于溢出事件发生时的即时显示。对此位写 1 将清除此标志位。

UART 分频器锁存寄存器 – URDLR

该寄存器用来定义 UART 时钟分频比，以产生适当的波特率。

偏移量: 0x024

复位值: 0x0000\_0010

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	BRD							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	BRD							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[15:0]	BRD	波特率分频器 这 16 位定义了 UART 时钟分频比例。 波特率 = CK_UART / BRD CK_UART 时钟是连接到 UART 模块的系统时钟。 在 UART 模式中 BRD = 16 ~ 65535。

UART 测试寄存器 – URTSTR

该寄存器控制 UART 调试模式。

偏移量：0x028

复位值：0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位						LBM	
类型 / 复位							RW	0 RW 0

位	字段	描述
[1:0]	LBM	回送测试模式选择 00: 正常工作模式 01: 保留位 10: 自动回应模式 11: 回送模式

## 23 智能卡接口 (SCI)

### 简介

该智能卡接口 (SCI) 符合 ISO 7816-3 标准。此接口包括卡插入 / 移除检测、SCI 数据传输控制逻辑和数据缓冲器、内部定时 / 计数器和控制逻辑电路来完成所需的智能卡操作。智能卡接口充当一个智能卡读卡器, 以便与外部智能卡通信。智能卡接口的所有功能是由一系列控制寄存器和状态寄存器以及一些相关中断控制的。中断可使单片机注意到 SCI 的传输状态。

由于 ISO 7816-3 标准数据协议的复杂性, 无法在此提供全面的规范, 因此, 读者应该查阅其它外部信息以详细了解本标准。

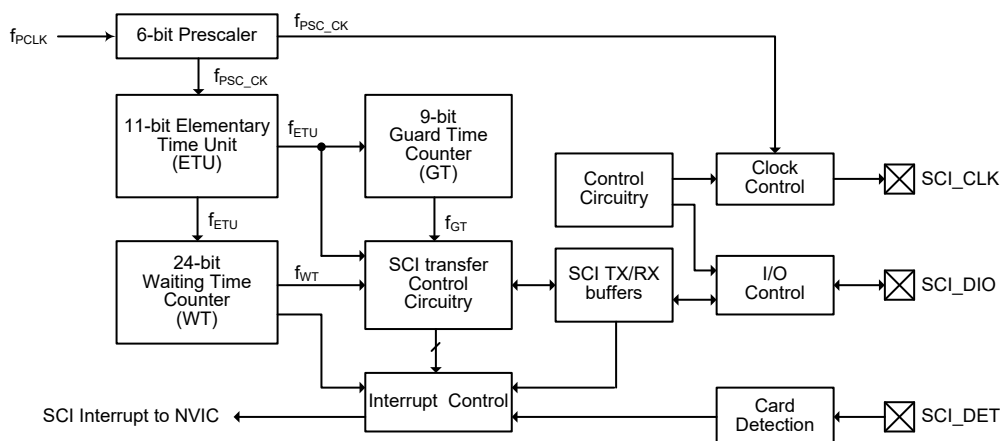


图 164. SCI 方框图

## 特性

- 支持 ISO 7816-3 标准
- 字符传输模式
- 一个发送缓冲器和一个接收缓冲器
- 11-bit 基本时间单元 (ETU) 计数器
- 9-bit 保护时间计数器
- 24-bit 通用等待时间计数器
- 奇偶检验位产生和检测
- 发送和接收模式下检测到奇偶检验错误时自动重发字符
- 支持 PDMA 存取当发送或者接收完成时

## 功能描述

要与外部智能卡通信，内部智能卡接口有一系列外部引脚，即 SCI\_CLK、SCI\_DIO 和 SCI\_DET。SCI\_CLK 是时钟输出信号引脚，与串行数据引脚 SCI\_DIO 一起用来与外部智能卡通信。SCI\_CLK 和 SCI\_DIO 的工作模式可以选择为 SCI 数据传输模式，由 SCI 控制电路自动驱动，或手动模式，由应用程序分别设置内部 CLK 和 DIO 寄存器位来控制。SCI\_DET 是外部智能卡检测的输入引脚，当外部智能卡插入或移除被检测到时，如果相应的中断控制位使能，会产生一个中断信号发送到单片机。

为实现正确的数据传输，一些定时用途的设定程序必须在智能卡接口开始与外部智能卡进行通信之前执行。有三个内部计数器，基本时间单元 (ETU)、保护时间计数器 (GT) 和等待时间计数器 (WT)，用于智能卡接口操作的定时相关功能。

### 基本时间单元计数器

基本时间单元 (ETU) 是一个 11-bit 向上计数型计数器，产生的时钟  $f_{ETU}$  可以作为 SCI 数据发送和接收的工作频率。ETU 的时钟源来自智能卡时钟  $f_{PSC\_CK}$ ，此时钟源自一个 6-bit 分频器产生。SCI 的数据传输是一个基于字符帧的协议，基本上包括一个起始位、8 位数据和一个奇偶校验位。ETU 产生的时间周期  $t_{ETU}$  ( $1/f_{ETU}$ )，是一个字符位的时间单元。有一个与基本时间单元相关的寄存器 ETUR，用来存储 ETU 的预期内容。每次 ETUR 寄存器被写入，ETU 电路都将重新加载新写入的值，并重新开始计数。基本时间单元  $t_{ETU}$  由如下公式取得。波特率的定义遵循 ISO 7816-3 标准。

$$1\text{etu} = t_{ETU} = \frac{E_i}{D_i} \times \frac{1}{f}$$

其中:

- $etu$  是由接口提供给智能卡的 SCI\_DIO 数据位的额定时间
- $D_i$  是比特率调整因数
- $F_i$  是时钟频率转换因数
- $f$  是由接口提供给智能卡的时钟信号 SCI\_CLK 的频率值

$D_i$  是基于 4-bit  $D_i$  字段编码的十进制值, 如下表所示。

**表 59. 基于  $D_i$  字段编码的  $D_i$  十进制值**

D <sub>i</sub> 字段	0001	0010	0011	0100	0101	0110	1000	1001
D <sub>i</sub> (十进制)	1	2	4	8	16	32	12	20

$F_i$  是基于 4-bit  $F_i$  字段编码的十进制值, 如下表所示。

**表 60. 基于  $F_i$  字段编码的  $F_i$  十进制值**

F <sub>i</sub> 字段	0000	0001	0010	0011	0100	0101	0110	1001	1010	1011	1100	1101
F <sub>i</sub> (十进制)	372	372	558	744	1116	1488	1860	512	768	1024	1536	2048

上表中的  $F_i$  和  $D_i$  的值, 在首次插入外部智能卡时, 将从外部智能卡发送到智能卡接口的复位应答包中得到。当智能卡接口收到  $F_i$  和  $D_i$  信息,  $F_i$  和  $D_i$  的值可以通过查询上面两个表得到。在获得  $F_i$  和  $D_i$  值时, 写入到 ETUR 寄存器的值可以由  $F_i/D_i$  计算。下表显示了由  $F_i/D_i$  比例计算得到的 ETU 可能值。

**表 61. 由  $F_i/D_i$  比例计算得到的 ETU 可能值**

$\frac{F_i}{D_i}$	372	558	774	1116	1488	1860	512	768	1024	1536	2048
1	372	558	744	1116	1488	1860	512	768	1024	1536	2048
2	186	279	372	558	744	930	256	384	512	768	1024
4	93	139.5	186	279	372	465	128	192	256	384	512
8	46.5	69.75	93	139.5	186	232.5	64	96	128	192	256
16	23.25	34.87	46.5	69.75	93	116.2	32	48	64	96	128
32	11.625	17.43	23.25	34.87	46.5	58.13	16	24	32	48	64
12	31	46.5	62	93	124	155	42.66	64	85.33	128	170.6
20	18.6	27.9	37.2	55.8	74.4	93	25.6	38.4	51.2	76.8	102.4



补偿模式

由于 ETUR 寄存器的值是由上面的公式得到的，计算结果可能不是一个整数。如果计算的结果不是一个整数，小于整数  $n$  但大于整数  $(n-1)$ ，应写入 ETUR 寄存器的是整数  $n$  还是  $(n-1)$  取决于结果是接近  $n$  还是  $(n-1)$ 。这里所说的整数  $n$  是十进制数。

如果计算的结果接近  $(n-0.5)$ ，为了成功的数据传输，应设置 ETUR 寄存器中的补偿使能控制位 COMP 为 1 以使能补偿模式。当结果接近  $(n-0.5)$ ，补偿模式使能时，写入 ETUR 寄存器的值应是  $n$ 。然后 ETU 时间单元序列会由  $n$  个时钟周期和接着  $(n-1)$  个时钟周期这样交替产生。因此产生了  $(n-0.5)$  个时钟周期的平均时间单元，并允许 ETU 时间间隔减小为半个时钟周期。注意，当在 SCI 数据传输模式下出现起始位时，ETU 将重新载入 ETUR 寄存器的值并重新开始计数。

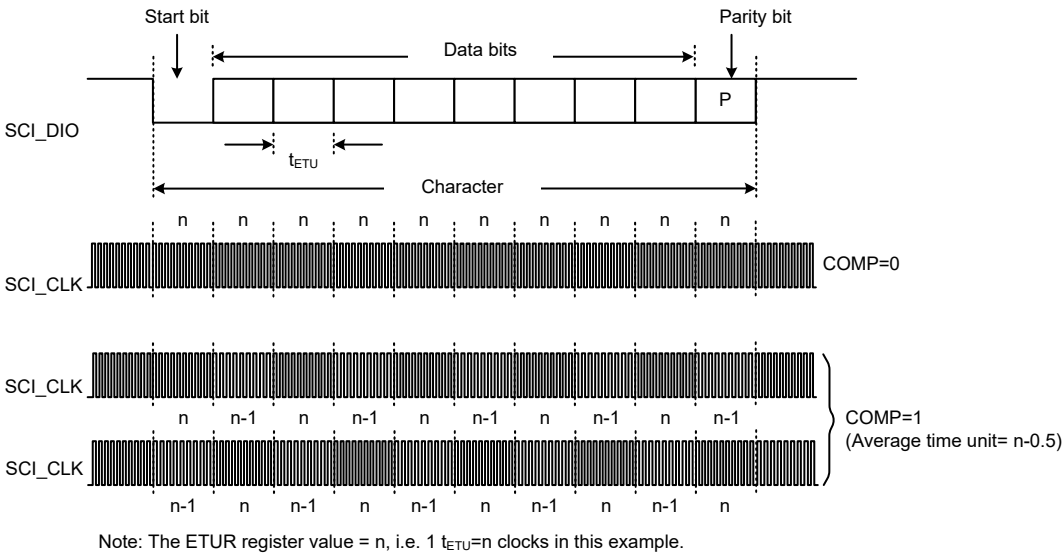


图 165. 字符帧和补偿模式

保护时间计数器

保护时间计数器 (GT) 是一个 9-bit 向上计数型计数器，在 SCI 数据传输的两个连续字符的前沿之间产生一个最低时间长度，即字符帧  $t_{GT}$ 。保护时间计数器的时钟源来自于 ETU，即方框图中的  $f_{ETU}$ 。有一个与保护时间计数器相关的寄存器 GTR，用来存储保护时间计数器的预期值。保护时间值将在当前保护时间周期结束时重新加载。注意，从智能卡收最后一个字符到从智能卡接口发送下一个字符之间的保护时间应该通过应用程序合理设置。因为第一个字符被发送时不插入保护时间。

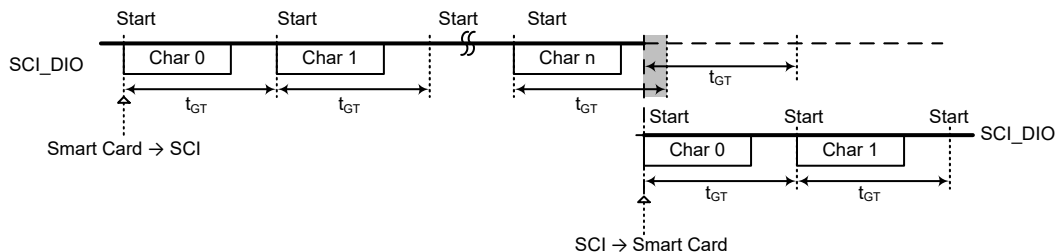


图 166. 保护时间

## 等待时间计数器

等待时间计数器 WT 是一个 24-bit 向下计数型计数器，生成数据传输的最大持续时间为  $t_{WT}$ ，等待时间计数器的时钟源来自于 ETU，即  $f_{ETU}$ 。

有一个用于等待时间计数器的寄存器 WTR，用来存储等待时间计数器的预期值。等待时间计数器可用于 SCI 数据传输模式和手动模式，可以在特定条件重新加载。等待时间计数器的功能由 CR 寄存器中的 WTEN 位控制。当 SCI 设置在 SCI 数据传输模式且 WTEN 位置 1 使等待时间计数器使能时，如果检测到起始位，则更新的 WTR 寄存器值将被加载到等待时间计数器。注意，在 SCI 数据传输模式，在外部智能卡未插入之前 WTEN 位不能置 1 使能等待时间计数器。

如果 SCI 被配置为手动模式，等待时间计数器可作为普通定时器使用，定时器可通过置位或清零 WTEN 位来控制使能或除能。如果等待时间计数器被使能，则更新的 WTR 寄存器值将不会被载入等待时间计数器中。当等待时间计数器由 WTEN 位置 0 除能且一个新的值被写入 WTR 寄存器，新的值将立即载入到等待时间计数器，当 WTEN 位再次置 1 时，计数器将自动开始计数。

在传输过程中，软件可以改变等待时间的值。例如，在 T=1 模式，块等待时间  $t_{BWT}$  应该在最后的传输字符的起始位出现前加载到 WTR 寄存器。当最后一个字符传输完成时，软件应把字符等待时间值  $t_{CWT}$ ，写入到 WTR 寄存器中。

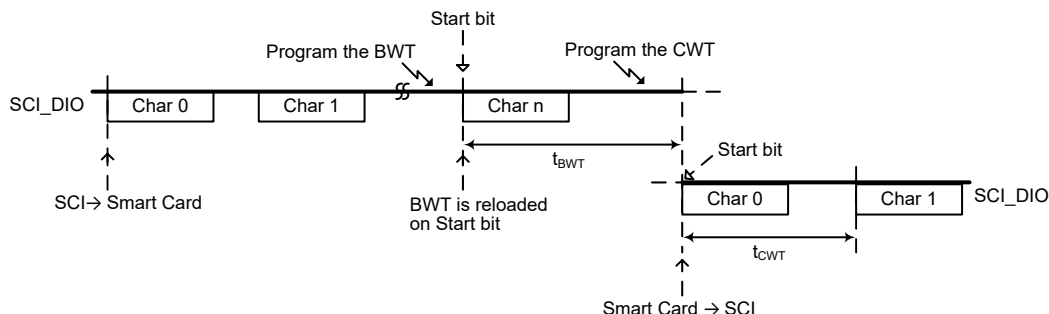


图 167. 字符和块等待时间 – CWT 和 BWT

## 智能卡时钟和数据选择

SCI 通过一系列的外部引脚与外部智能卡通信。这些引脚是串行数据引脚 SCI\_DIO、外部时钟引脚 SCI\_CLK 和智能卡检测输入引脚 SCI\_DET。

SCI 串行数据引脚 SCI\_DIO, 可选择由 SCI 硬件电路或软件通过相关的控制位来控制, 这取决于 SCI 是工作在 SCI 传输模式还是工作在手动模式。工作模式通过 CR 寄存器中的 SCIM 位进行选择。当清零 CR 寄存器的 SCIM 位时, SCI 工作在手动模式, SCI\_DIO 引脚状态由 CCR 寄存器中的 CDIO 位控制。但当 SCI 工作在 SCI 传输模式时, SCI\_DIO 引脚状态由 SCI 传输电路控制。

SCI 时钟输出引脚 SCI\_CLK 可选择由 6-bit SCI 预分频器或软件位进行控制, 取决于 CCR 寄存器中的 CLKSEL 位的情况。当 CLKSEL 位清零时, SCI\_CLK 引脚状态由 CCR 寄存器中的 CCLK 位控制。但当 CLKSEL 位置 1 时, SCI\_CLK 信号来自于 6-bit 预分频器的输出信号。预分频器的分频比由 PSCR 寄存器中的 PSC 字段控制。

## 智能卡检测

当有外部智能卡插入时, 内部卡检测器可以检测到这个插入动作, 如果 IER 寄存器中对应的中断控制位 CARDIRE 置 1, 将产生卡插入中断。类似地, 如果卡被移除, 内部卡检测器同样会检测到移除动作, 如果 IER 寄存器中的中断控制位 CARDIRE 置 1, 相关中断使能, 将产生卡移除中断。

卡检测器可支持两种检测开关机制。一种是没有卡时开关常 OFF 机制, 另一种是没有卡时开关常 ON 机制。在注意到哪个卡检测机制类型被选择后, 应设置 CR 寄存器中的 DETCNF 位来选择开关机制使其正确检测卡是否存在。当卡实际存在于 SCI\_DET 引脚时, 无论由 DETCNF 位设置为哪个卡开关类型, SR 寄存器中卡插入 / 移除标志位 CPREF 都将被置 1。注意, 卡检测器没有硬件去抖电路。SCI\_DET 引脚电平的任何变化都将引起 CPREF 位发生变化。所需的去抖时间由应用程序进行处理。

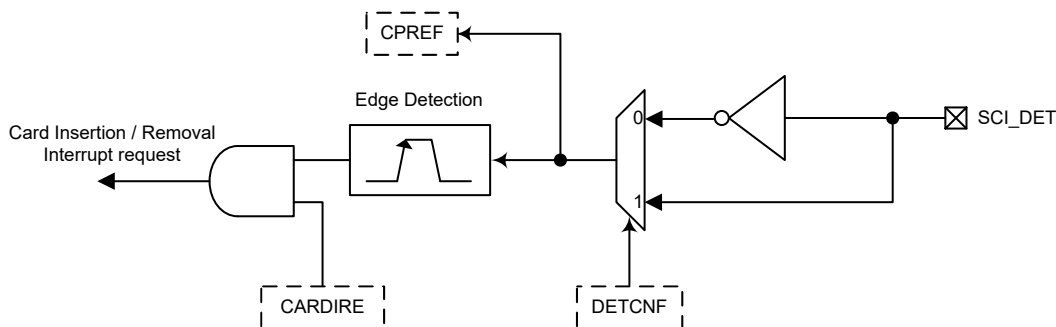


图 168. SCI 卡检测方框图

## SCI 数据传输模式

与外部智能卡进行 SCI 数据传输有两个工作模式。一个是 SCI 模式而另一个是手动模式。数据传输模式通过 CR 寄存器中的 SCI 模式选择位 SCIM 选择。当 SCIM 位设置为 1 时, SCI 模式使能, 数据将自动由 SCI 传输电路进行传输。否则, 如果 SCIM 位设置为 0, 数据传输在手动模式中运行。SCI 传输接口是半双工接口, 通过 SCI\_CLK 和 SCI\_DIO 引脚与外部智能卡进行通信。复位后, SCI 传输接口处于接收模式, SCI 传输被除能。当选择了 SCI 模式, 数据传输通过 SCI\_CLK 和 SCI\_DIO 引脚由 SCI 传输电路自动驱动。

有两个与数据发送和接收相关的数据寄存器, TXB 和 RXB, 分别存储发送和接收的数据。如果在 SCI 传输模式下, 一个字符被写入到 TXB 寄存器, 复位后 SCI 传输接口将自动从接收模式切换到发送模式。当 SCI 发送或接收完成后, SR 寄存器中相应的请求标志 TXCF 或 RXCF 将被设置为 1。如果发送缓冲器是空的, SR 寄存器中的发送缓冲器空标志 TXBEF 将被设置为 1。

## 奇偶校验功能

SCI 传输接口支持奇偶发生器和奇偶校验功能。当在数据传输过程中发生奇偶校验错误, SR 寄存器相应的请求标志 PARF 将被设置为 1。一旦 PARF 位被设置为 1, 如果 IER 寄存器中的相关中断控制位 PARE 使能, 则 IPR 寄存器中的奇偶校验错误挂起标志 PARP 也将被设置为 1。

如果 SCI 发送的数据被智能卡接收器所接收, 且在发送过程中没有奇偶校验错误, SCI 接口发送请求标志 TXCF 将被设置为 1, SCI 奇偶错误请求标志位 PARF 位将被清除为 0。如果由外部智能卡发送的数据被 SCI 接口所接收, 且在接收过程中无奇偶校验错误, SCI 接口接收请求标志 RXCF 将被设置为 1, PARF 位将保持为零。

### 重发功能

当奇偶校验错误发生时, SCI 传输电路支持字符重发功能。字符重发功能由 CR 寄存器中的 CREP 位置 1 使能。在数据传输过程中, 如果发生奇偶错误, 则重发功能将使能。重发次数可由 CR 寄存器中的 RETRY 位选择为 4 或 5。

当 CREP 位置 1, 字符重发功能将使能。以 4 次重发为例, 当 CREP 位置 1 且 RETRY 位也置 1 时, 在发送模式下, 如果产生奇偶校验错误信号, 则 SCI 将重复发送数据, 最多重发 4 次。如果 SCI 接口被告知在 4 次发送都存在奇偶校验错误, SCI 接口奇偶校验错误标志 PARF 将在相同的数据发送 4 次后被设置为 1, 但 TXCF 将不会被置位。此时在发送缓冲器中的数据将被载入到发送移位寄存器, 发送缓冲器为空, 将使 TXBEF 标志位设为 1。

同样地, 如果 SCI 接口处于接收模式, 当字符重发功能使能, 它会告知外部智能卡有一个奇偶校验错误, 最多 4 次。如果 SCI 告知外部智能卡在 4 次发送过程中一直存在奇偶校验错误, 奇偶错误标志位 PARF 和接收请求标志位 RXCF 都将被置 1。

如果 CREP 位清零, 字符重发功能将除能。当 SCI 工作在接收模式时, 当带有奇偶错误的数被接收时, PARF 和 RXCF 位都将置 1。如果在发送模式下 SCI 被告知有奇偶错误, PARF 位将被置 1, 但 TXCF 位不会被置位。

### 手动数据传输模式

当设置 SCIM 位为 0, 选择在手动模式下传输数据。在手动模式下, 数据是由 CCR 寄存器中的控制位 CDIO 控制。在手动模式下, CDIO 位的值将立即反映在 SCI\_DIO 引脚上。注意, 在手动模式下的字符重发功能以及相关的标志位都不能使用, 所有的数据传输是由应用程序处理。用来驱动外部智能卡的时钟来自于 SCI\_CLK 引脚, 由内部时钟源, 即 6-bit 预分频器的输出  $f_{PSC\_CK}$ , 或 CCR 寄存器中的控制位 CCLK 提供。时钟源由 CCR 寄存器的 CLKSEL 位选择。当 CLKSEL 被设置 1, 选择 6-bit 预分频器的输出  $f_{PSC\_CK}$  作为智能卡的时钟源。如果用户想手动来处理时钟, 应先设置 CLKSEL 位为 0, 然后 CCLK 位的值将出现在 SCI\_CLK 引脚上。

### 数据传输方向协议

如果智能卡所使用的方向协议与智能卡接口的协议相同, 当接收中断使能且无奇偶校验错误时, SCI 将产生接收中断。否则, SCI 产生接收中断且奇偶校验错误标志将被置 1。通过检查奇偶校验错误标志, SCI 可以知道数据方向协议正确与否。

中断发生器

SCI 产生 SCI 中断有几个条件。当这些条件存在，将会产生一个中断脉冲引起单片机的注意。这些条件包括智能卡插入 / 移除、等待时间计数器下溢、奇偶校验错误、字符发送或接收结束和发送缓冲器为空。当智能卡中断由这些条件的任何一个产生，如果 NVIC 中 SCI 总中断和 SCI 相应的中断控制同时使能，程序将在返回到主程序之前跳转到相应的中断向量。

对于 SCI 的中断事件，相应的挂起标志可以由相应的中断使能控制位屏蔽。当相关的中断使能控制除能，相应的中断挂起标志不受请求标志影响，不会产生中断。如果相关的中断使能控制位使能，相应的中断挂起标志将受中断请求标志影响，并产生中断。挂起标志寄存器 IPR 是只读的，一旦挂起标志由应用程序读取，它会自动清零，而相关的请求标志由应用程序手动清零。

为了响应 SCI 中断，除了 SCI 相应的中断使能控制位被置位，NVIC 中 SCI 总中断使能控制位也必须被置位。如果 SCI 总中断控制位没有被置位，将不会响应中断。

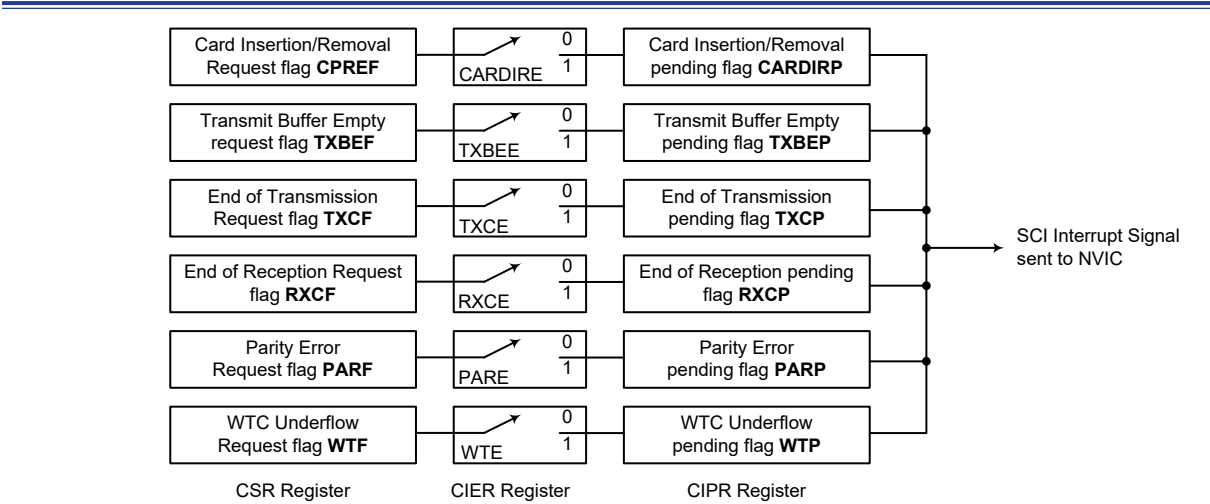


图 169. SCI 中断结构

PDMA 接口

PDMA 接口集成在 SCI 模块中。在发送或接收模式下通过对应设置 CR 寄存器中的 TXDMA 或 RXDMA 位为 1 可使能 PDMA 功能。当发送缓冲器为空从而置位发送缓冲器空标志位 TXBEF，且 TXDMA 位被置 1 时，PDMA 功能使能，将数据从特定的存储器地址移到 SCI 发送缓冲器中。同样地，当接收到数据从而置位接收完成标志位 RXCF，且 RXDMA 位被置 1 时，PDMA 功能使能，将数据从 SCI 接收缓冲器移到特定的存储器地址中。

详细的 PDMA 配置信息，请参考 PDMA 章节。

## 寄存器列表

有几个与智能卡功能相关的寄存器。一些寄存器控制 SCI 所有功能以及中断，而一些寄存器包含状态位，用来表示智能卡的数据传输情况和错误条件。另外有两个 SCI 发送和接收相关的寄存器，分别用于存放将要发送到外部智能卡的数据，或从外部智能卡收到的数据。下表显示了 SCI 相关寄存器和复位值。

**表 62. SCI 寄存器列表**

寄存器	偏移量	描述	复位值
CR	0x000	SCI 控制寄存器	0x0000_0000
SR	0x004	SCI 状态寄存器	0x0000_0080
CCR	0x008	SCI 通信控制寄存器	0x0000_0008
ETUR	0x00C	SCI 基本时间单元寄存器	0x0000_0174
GTR	0x010	SCI 保护时间寄存器	0x0000_000C
WTR	0x014	SCI 等待时间寄存器	0x0000_2580
IER	0x018	SCI 中断使能寄存器	0x0000_0000
IPR	0x01C	SCI 中断挂起寄存器	0x0000_0000
TXB	0x020	SCI 发送缓冲器	0x0000_0000
RXB	0x024	SCI 接收缓冲器	0x0000_0000
PSCR	0x028	SCI 预分频寄存器	0x0000_0000



寄存器描述

SCI 控制寄存器 – CR

该寄存器包含 SCI 控制位。

偏移量: 0x000  
复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位						RXDMA	TXDMA
							RW 0	RW 0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位	DETCNF	ENSCI	RETRY	SCIM	WTEN	CREP	CONV
		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

位	字段	描述															
[9]	RXDMA	SCI RX DMA 使能位 0: 除能 1: 使能															
[8]	TXDMA	SCI TX DMA 使能位 0: 除能 1: 使能															
[6]	DETCNF	卡开关类型选择位 0: 如果没有卡, 开关是常 ON 的 1: 如果没有卡, 开关是常 OFF 的 <table><tr><th>DETCNF</th><th>SCI_DET 引脚</th><th>状态</th></tr><tr><td>0</td><td>1</td><td>无卡插入</td></tr><tr><td>0</td><td>0</td><td>卡插入</td></tr><tr><td>1</td><td>1</td><td>卡插入</td></tr><tr><td>1</td><td>0</td><td>无卡插入</td></tr></table>	DETCNF	SCI_DET 引脚	状态	0	1	无卡插入	0	0	卡插入	1	1	卡插入	1	0	无卡插入
DETCNF	SCI_DET 引脚	状态															
0	1	无卡插入															
0	0	卡插入															
1	1	卡插入															
1	0	无卡插入															
[5]	ENSCI	此位由应用程序置位和清零, 以设置智能卡检测器的开关类型。 SCI 有限状态机制 (FSM) 使能位 0: SCI FSM 除能并强制进入初始状态 1: SCI FSM 使能															
[4]	RETRY	一个奇偶错检验误发生时字符传输重发次数选择 0: 奇偶检验错误发生时数据最多重发 5 次 1: 奇偶检验错误发生时数据最多重发 4 次 此位仅当 CREP 被置 1 时可用。当此位被置 1, 一旦奇偶校验错误发生, 数据将被发送或接收 4 次。如果此位被置 0, 一旦奇偶校验错误发生, 数据将被传输 5 次。															



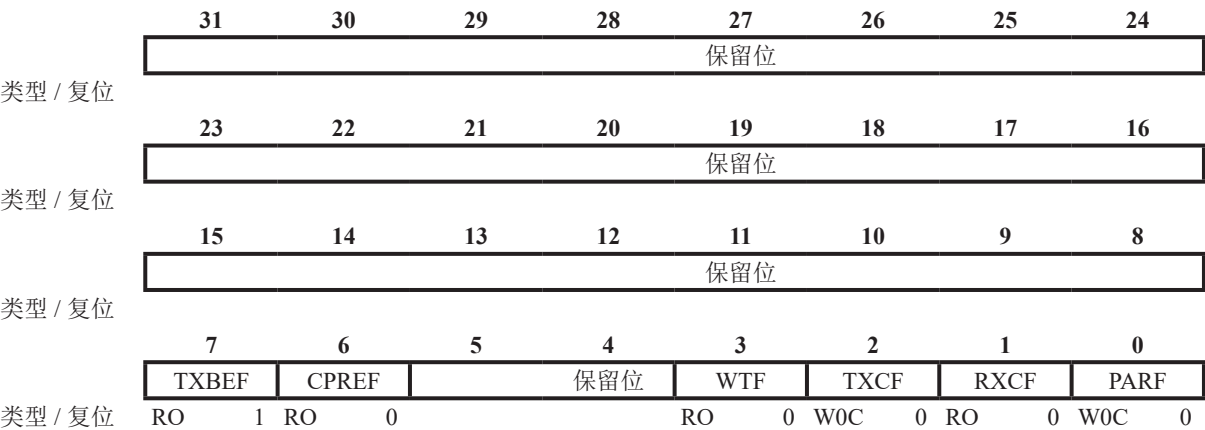
位	字段	描述
[3]	SCIM	<p>SCI 模式选择位</p> <p>0: 手动模式下 SCI 数据传输 1: SCI 模式下 SCI 数据传输</p> <p>此位由应用程序置位或清零用于选择 SCI 数据传输模式。如果此位清零, SCI_DIO 引脚状态与 CCR 寄存器中的 CDIO 位相同。如果置 1, SCI_DIO 引脚由内部 SCI 控制电路驱动。在数据传输类型从手动模式切换到 SCI 模式前, CDIO 位必须被置 1 来避免 SCI 故障。</p>
[2]	WTEN	<p>等待时间计数器使能控制位</p> <p>0: 停止计数 1: 开始计数</p> <p>此位由应用程序置位和清零。当 WTEN 位清零, 写访问 WTR 寄存器将载入值到等待时间计数器中。如果此位置 1, 等待时间计数器将使能, 在每次起始位出现时自动重新加载。</p>
[1]	CREP	<p>奇偶校验错误条件下的字符自动重复使能控制位</p> <p>0: 奇偶校验错误时不重试 1: 奇偶校验错误时自动重试</p> <p>CREP 位由应用程序置位和清零。当 CREP 位被清除为 0, 在接收模式中收到的数据发生奇偶校验错误后, RXCF 和 PARF 标志将被置位, 而在发送模式下 PARF 被置位, TXCF 不会被置位。如果 CREP 位被置 1, 字符传输将自动被重试 4 或 5 次取决于 RETRY 位的值。在发送模式, 如果发生奇偶校验错误, 则字符将会被重复发送。在第 4 或第 5 次发送结束时奇偶校验错误标志 PARF 将被置位, 但 TXCF 不会被置位。在接收模式中, 如果接收到的数据有一个奇偶校验错误, SCI 将通知外部智能卡重复发送 4 或 5 次, 然后在第 4 或第 5 次接收结束时 PARF 和 RXCF 标志都将被置位。</p>
[0]	CONV	<p>数据方向协议选择位</p> <p>0: LSB 先发送; 在 SCI_DIO 引脚的数据“1”是逻辑高电平, 奇偶校验位跟随在 MSB 后 1: MSB 先发送; 在 SCI_DIO 引脚的数据“1”是逻辑低电平, 奇偶校验位跟随在 LSB 后</p> <p>该位由应用程序置位和清零, 以选择数据是 LSB 首先发送还是 MSB 首先发送。当协议的方向与外部智能卡指定的方向相同时, 只有 RXCF 被设置为 1, 无奇偶校验错误。否则, 在收到数据后 RXCF 和 PARF 都将被设置为 1。</p>

SCI 状态寄存器 – SR

该寄存器包含 SCI 状态位。

偏移量： 0x004

复位值： 0x0000\_0080



位	字段	描述
[7]	TXBEF	发送缓冲器空请求标志位 0：发送缓冲器不为空 1：发送缓冲器空 此位用来表示发送缓冲器是否为空，由硬件自动置位和清零。
[6]	CPREF	卡存在请求标志位 0：没有卡 1：有卡 此位用来表示是否有卡，由硬件自动置位和清零。ENSCI 位置位时，卡存在检测功能使能。
[3]	WTF	等待时间计数器下溢请求标志位 0：等待时间计数器无下溢 1：等待时间计数器下溢 此位由应用程序置位和清零，表示等待时间计数器是否下溢。
[2]	TXCF	字符发送请求标志位 0：无字符发送请求 1：字符已发送 此位由硬件置位，由应用程序写入 0 清零。
[1]	RXCF	字符接收请求标志位 0：无字符接收请求 1：字符已接收 此位自动由硬件置位，由应用程序读访问 RXB 寄存器后清零。不管奇偶校验的结果为何，当接收到字符，RXCF 位将设置为 1。 当字符已收到，存储在 RXB 寄存器的已接收到的数据应被转移到应用程序指定的数据存储器。如果在下一个字符移入结束前 RXB 寄存器的内容没被读取，存储在 RXB 寄存器的数据将被覆盖。

位	字段	描述
[0]	PARF	奇偶校验错误请求标志位 0: 无奇偶校验错误请求 1: 奇偶校验错误已发生 此位自动由硬件置位，由应用程序清零。当接收到字符，奇偶校验电路检查校验正确与否。如果奇偶校验的结果是不正确的，奇偶校验错误请求标志 PARF 将被设置为 1。否则，PARF 位将保持为零。在发送模式，当外部智能卡被告知 SCI 发送的字符有奇偶校验错误，PARF 位也会被置位。

### SCI 通信控制寄存器 – CCR

该寄存器指定了 SCI 引脚的设置和时钟选择。

偏移量: 0x008

复位值: 0x0000\_0008

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
CLKSEL	保留位			CDIO	CCLK	保留位	
RW	0			RW	1	RW	0

位	字段	描述
[7]	CLKSEL	智能卡时钟选择位 0: CCLK 位的内容为外部 SCI_CLK 引脚的状态 1: 外部 SCI_CLK 引脚的时钟输出来自 f <sub>PSC_CLK</sub> 时钟 此位用来选择外部 SCI_CLK 引脚的时钟源。它由应用程序置位和清零。建议在 CLKSEL 位由 1 切换到 0 之前，写入一个已知值到 CCLK 位使时钟激活后处于一个已知电平。
[3]	CDIO	SCI_DIO 引脚控制位 0: SCI_DIO 引脚是逻辑电平 0 1: SCI_DIO 引脚处于开漏状态 仅当 CR 寄存器中 SCIM 位清零用来配置 SCI 工作在手动模式时，此位才可用。手动模式下，由应用程序置位和清零以控制外部 SCI_DIO 引脚状态。读取此位将返回 SCI_DIO 引脚的当前状态。
[2]	CCLK	SCI_CLK 引脚控制位 0: SCI_CLK 引脚为逻辑电平 0 1: SCI_CLK 引脚为逻辑电平 1 此位仅在 SCI 工作在手动传输模式时可用。手动模式下，由应用程序置位和清零以控制外部 SCI_CLK 引脚状态。读取此位将返回寄存器中此位的当前值而非 SCI_CLK 引脚的当前状态。在 CLKSEL 位由 1 切换到 0 之前，写入某一已知值到 CCLK 位以确保持钟保持在一个已知电平值。

SCI 基本时间单元寄存器 – ETUR

该寄存器定义的 ETU 值按基本时间单元计数器章节所述公式计算。它还包括 ETU 时间间隔补偿功能的控制位。

偏移量: 0x00C  
复位值: 0x0000\_0174

	31	30	29	28	27	26	25	24				
	保留位											
类型 / 复位												
	23	22	21	20	19	18	17	16				
	保留位											
类型 / 复位												
	15	14	13	12	11	10	9	8				
	COMP	保留位				ETU						
类型 / 复位	RW	0				RW	0	RW	0	RW	1	
	7	6	5	4	3	2	1	0				
	ETU											
类型 / 复位	RW	0	RW	1	RW	1	RW	1	RW	0	RW	0

位	字段	描述
[15]	COMP	基本时间单元补偿模式使能控制位 0: 补偿模式除能 1: 补偿模式使能 此位由应用程序置位和清零，用来控制 ETU 补偿功能。有关补偿功能的细节请参照基本时间单元章节。
[10:0]	ETU	一个字符数据位的 ETU 值 该字段由应用程序配置用于修改 ETU 的值。注意，ETU 的值必须在 0x00C ~ 0x7FF 范围内。为获得十进制的最大 ETU 值 2048，需向此位写入 0x000 值。

SCI 保护时间寄存器 – GTR

该寄存器定义了保护时间值。该保护时间值可从复位应答包获取，详细参考保护时间计数器章节。

偏移量： 0x010  
复位值： 0x0000\_000C

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	保留位							GT	
类型 / 复位								RW	0
	7	6	5	4	3	2	1	0	
	GT								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[8:0]	GT	字符保护时间值 此字段由应用程序配置用于修改保护时间。在当前保护时间周期结束时，更新的 GT 值将被载入到 GT 计数器中。注意，GT 值必须在 0x00C~0x1FF 范围内。

SCI 等待时间寄存器 – WTR

该寄存器定义了等待时间值。该等待时间值可从复位应答包获取，详细参考等待时间计数器章节。

偏移量： 0x014  
复位值： 0x0000\_2580

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	WT							
	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
类型 / 复位	WT							
	RW	0	RW	0	RW	1	RW	0
	7	6	5	4	3	2	1	0
类型 / 复位	WT							
	RW	1	RW	0	RW	0	RW	0

位	字段	描述
[23:0]	WT	字符等待时间值通过 ETU 表示 (0/16777215)。 此字段由应用程序配置用于修改等待时间。等待时间计数器更新值的重载条件已经在等待时间计数器章节描述过，更多的细节请参考等待时间计数器章节。注意，WT 的值在 0x00_0000 ~ 0xFF_FFFF 之间。

SCI 中断使能寄存器 – IER

该寄存器定义了 SCI 中控制所有中断事件的中断控制使能位。

偏移量： 0x018

复位值： 0x0000\_0000

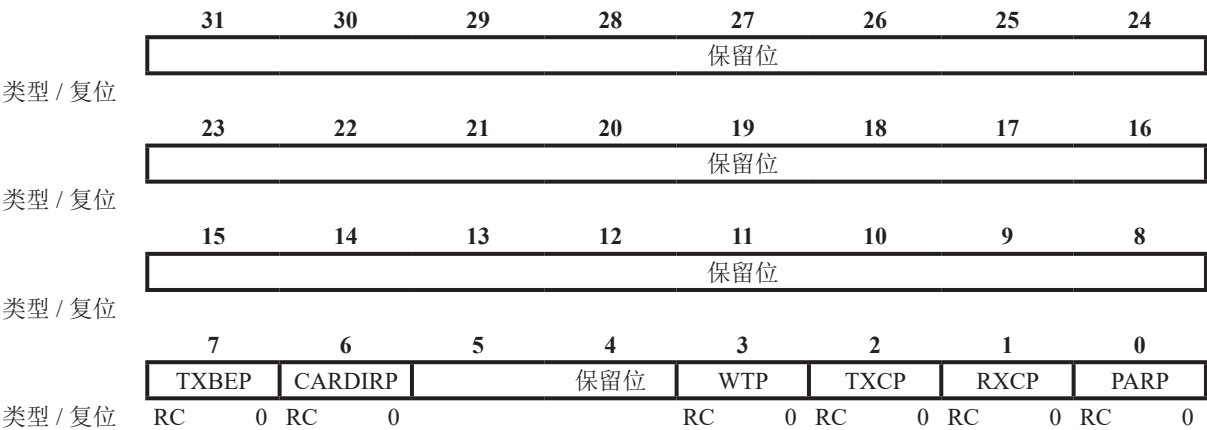
	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	TXBEE	CARDIRE	保留位		WTE	TXCE	RXCE	PARE
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[7]	TXBEE	发送缓冲器空中断使能控制位 0: 除能 1: 使能 此位由应用程序置位和清零来控制发送缓冲器空中断。如果此位置 1，当发送缓冲器为空时，将产生发送缓冲器空中断。
[6]	CARDIRE	卡插入 / 移除中断使能控制位 0: 除能 1: 使能 此位由应用程序置位和清零来控制卡插入 / 移除中断。如果此位置 1，当外部智能卡被插入或移除时，将产生卡插入 / 移除中断。
[3]	WTE	产生等待时间计数器下溢中断使能控制位 0: 除能 1: 使能 此位由应用程序置位和清零来控制产生等待时间计数器下溢中断。如果此位置 1，当等待时间计数器下溢时，将产生等待时间计数器下溢中断。
[2]	TXCE	字符发送完成中断使能控制位 0: 除能 1: 使能 此位由应用程序置位和清零来控制字符发送完成中断。如果此位置 1，在字符发送完成时将产生字符发送完成中断。
[1]	RXCE	字符接收完成中断使能控制位 0: 除能 1: 使能 此位由应用程序置位和清零来控制字符接收完成中断。如果此位置 1，在字符接收完成时将产生字符接收完成中断。
[0]	PARE	奇偶校验错误中断使能控制位 0: 除能 1: 使能 此位由应用程序置位和清零来控制奇偶校验错误中断。如果此位置 1，在奇偶校验错误发生时将产生奇偶校验错误中断。

SCI 中断挂起寄存器 – IPR

此寄存器包含 SCI 所有中断事件的中断挂起标志位。这些标志位可以通过相应的中断使能控制位屏蔽。

偏移量： 0x01C  
复位值： 0x0000\_0000



位	字段	描述
[7]	TXBEP	发送缓冲器空中断挂起标志位 0: 无中断挂起 1: 中断挂起 此位由硬件置位，通过使用应用程序读访问该寄存器时清零。用来表示发送缓冲器空中断挂起与否。如果相应的中断使能控制位被设置为 1 且发送缓冲器为空，该位会被置为 1，表示发送缓冲器空中断正在等待处理。
[6]	CARDIRP	卡插入 / 移除中断挂起标志位 0: 无中断挂起 1: 中断挂起 此位由硬件置位，通过使用应用程序读访问该寄存器时清零。用来表示外部智能卡插入 / 移除中断挂起与否。如果相应的中断使能控制位被设置为 1 且有卡插入或移除动作发生，该位会被置为 1，表示卡插入 / 移除中断正在等待处理。
[3]	WTP	等待时间计数器下溢中断挂起标志位 0: 无中断挂起 1: 中断挂起 此位由硬件置位，通过使用应用程序读访问该寄存器时清零。用来表示等待时间计数器下溢中断挂起与否。如果相应的中断使能控制位被设置为 1 且有等待时间计数器下溢发生，该位会被置为 1，表示等待时间计数器下溢中断正在等待处理。
[2]	TXCP	字符发送完成中断挂起标志位 0: 无中断挂起 1: 中断挂起 此位由硬件置位，通过使用应用程序读访问该寄存器时清零。用来表示字符发送完成中断挂起与否。如果相应的中断使能控制位被设置为 1 且字符发送完成，该位会被置为 1，表示字符发送完成中断正在等待处理。
[1]	RXCP	字符接收完成中断挂起标志位 0: 无中断挂起 1: 中断挂起 此位由硬件置位，通过使用应用程序读访问该寄存器时清零。用来表示字符接收完成中断挂起与否。如果相应的中断使能控制位被设置为 1 且字符接收完成，该位会被置为 1，表示字符接收完成中断正在等待处理。

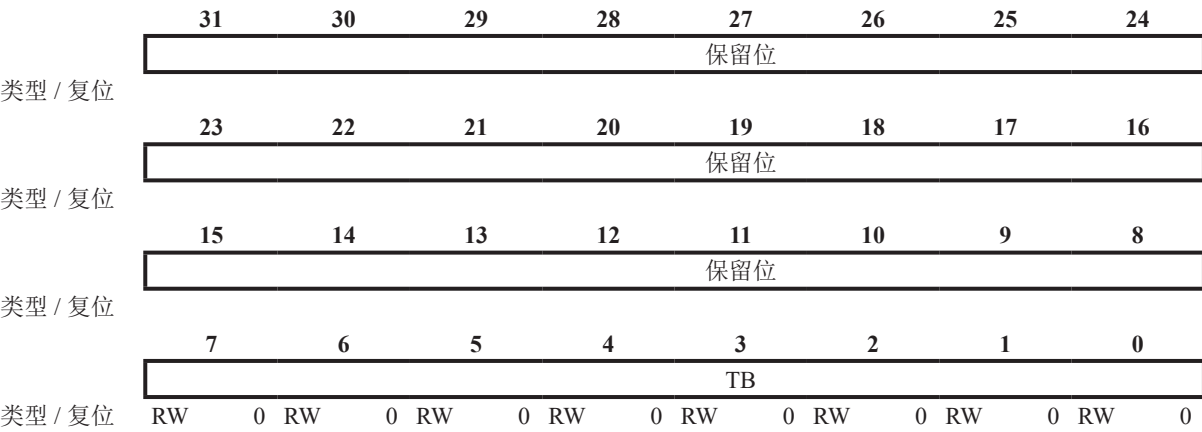


位	字段	描述
[0]	PARP	奇偶校验错误中断挂起标志位 0: 无中断挂起 1: 中断挂起 此位由硬件置位，通过使用应用程序读访问该寄存器时清零。用来表示奇偶校验错误中断挂起与否。如果相应的中断使能控制位被设置为 1 且奇偶校验错误发生时，该位会被置为 1，表示奇偶校验错误中断正在等待处理。

SCI 发送缓冲器 – TXB

该寄存器用来存储要发送的 SCI 数据。

偏移量: 0x020  
复位值: 0x0000\_0000



位	字段	描述
[7:0]	TB	要发送的 SCI 数据字节

SCI 接收缓冲器 – RXB

该寄存器用来存储接收到的 SCI 数据。

偏移量： 0x024  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	RB							
类型 / 复位	RO	0	RO	0	RO	0	RO	0

位	字段	描述
[7:0]	RB	接收到的 SCI 数据字节

SCI 预分频器寄存器 – PSCR

该寄存器定义了 SCI 内部时钟的预分频器的分频比。

偏移量: 0x028  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
类型 / 复位	保留位							
	23	22	21	20	19	18	17	16
类型 / 复位	保留位							
	15	14	13	12	11	10	9	8
类型 / 复位	保留位							
	7	6	5	4	3	2	1	0
类型 / 复位	保留位		PSC					
			RW	0	RW	0	RW	0
					RW	0	RW	0
						RW	0	RW
							RW	0
								0

位	字段	描述
[5:0]	PSC	SCI 预分频器分频比 0: $f_{PSC\_CK} = f_{PCLK}$ 1~63: $f_{PSC\_CK} = \frac{f_{PCLK}}{2 \times PSC}$

## 24 USB 设备控制器 (USB)

### 简介

该 USB 设备控制器符合 USB 2.0 全速规范。其具有一个控制端点 (端点 0) 和 7 个可配置的端点 (EP1 ~ EP7)。一个 1024-byte 的 EP\_SRAM 用作端点缓冲器。每个端点缓冲器的大小可通过设置相应寄存器来得到, 为各种应用提供较大的灵活性。内置的 USB 全速收发器减小了整个系统的复杂性并且降低了成本。USB 包含了挂起和恢复功能, 满足了低功耗产品的需求。下图显示了 USB 方框图。

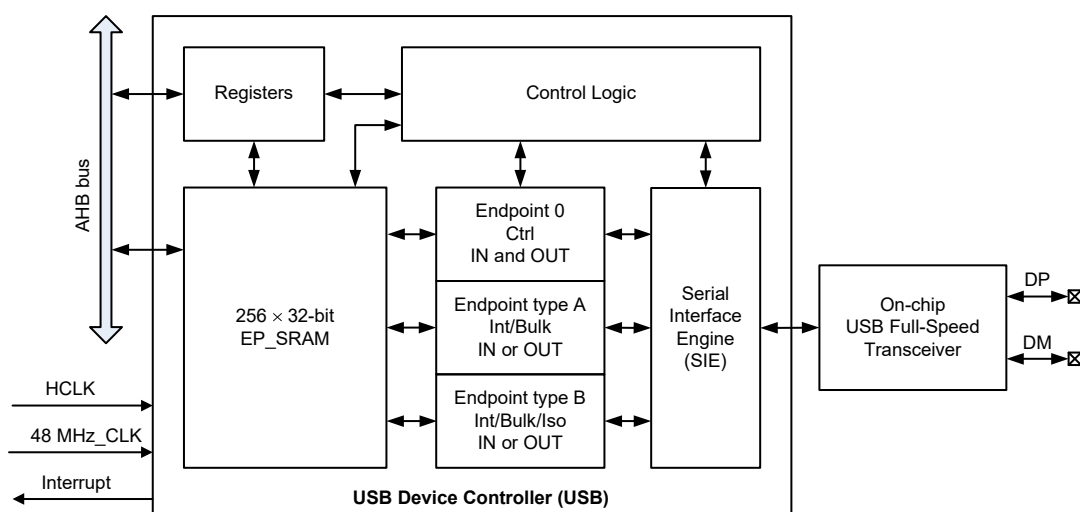


图 170. USB 方框图

### 特性

- 符合 USB 2.0 全速 (12 Mbps) 设备规范
- 完全集成的 USB 全速收发器
- 1 个控制端点 (EP0) 用于控制传输
- 3 个单缓冲端点 (EP1 ~ EP3) 用于批量和中断传输
- 4 个双缓冲端点 (EP4 ~ EP7) 用于批量、中断和等时传输
- 1024 字节 EP\_SRAM 用作端点数据缓冲器

功能描述

端点

USB 端点 0 是唯一一个用于 USB 控制传输的双向端点。该设备也包含 7 个单向的端点，用于其它 USB 传输类型。有三个端点 (EP1 ~ EP3) 支持单缓冲功能用于批量和中断 IN 或 OUT 数据传输。有另外四个端点 (EP4 ~ EP7) 支持单缓冲或双缓冲功能用于批量、中断和等时 IN 或 OUT 数据传输。7 个单向端点 (EP1 ~ EP7) 的地址可由应用软件设置。下表列出了端点特性。

表 63. 端点特性

端点编号	端点地址	传输类型	方向	缓冲器类型
0	固定	控制	IN 和 OUT	单缓冲
1 ~ 3	可设置	中断 / 批量	IN 或 OUT	单缓冲
4 ~ 7	可设置	中断 / 批量 / 等时	IN 或 OUT	单或双缓冲

EP\_SRAM

USB 控制器包含一个专用的存储器空间 EP\_SRAM，用作 USB 端点缓冲器。EP\_SRAM 与 AHB 总线相连，可由 CPU 和 PDMA 访问。EP\_SRAM 基址是 0x400A\_A000，偏移量范围是 0x000 ~ 0x3FF。EP\_SRAM 的前两个字被预留用于端点 0 临时存储 8-byte SETUP 数据。因此，端点缓冲器的有效起始地址应从 0x400A\_A008 开始并且对齐到 4-byte 边界。每个端点缓冲器容量都是可编程的。下表列出了符合 USB 2.0 全速设备规范的 USB 端点缓冲器的最大容量。

表 64. USB 数据类型和缓冲器容量

传输类型	方向	支持的缓冲器容量	带宽	CRC	重试
控制	双向	8, 16, 32, 64	不确定	Yes	自动
批量	单向	8, 16, 32, 64	不确定	Yes	Yes
中断	单向	≤ 64	不确定	Yes	Yes
等时	单向	< 512	确定	Yes	No

在下面端点缓冲器分配范例中，端点“4”设置作为双缓冲的批量 IN 端点，而端点“5”设置作为双缓冲的批量 OUT 端点。每个端点缓冲器的容量是 64 字节。

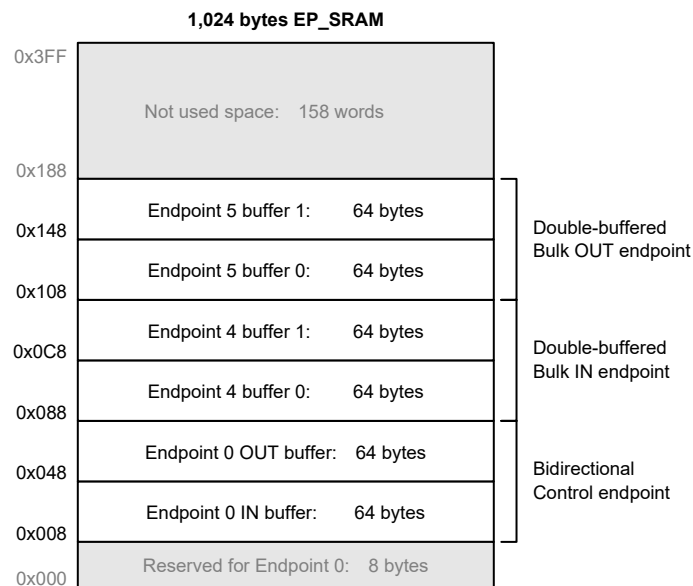


图 171. 端点缓冲器分配范例

串行接口引擎 – SIE

串行接口引擎 SIE 与 USB 全速收发器和内部 USB 控制电路相连，为发送和接收数据提供一个临时缓冲器。当 USB 模块接收数据、发送数据或发送恢复信号用于远程控制时，SIE 也可解码 SE0 信号、SE1 信号、J-state、K-state、USB RESET 事件和数据包结束事件信号 EOP。SIE 会检测 SOF 包的数量并对 USB 控制电路产生 SOF 中断信号，SIE 包含从并行到串行或串行到并行的数据格式转换功能。SIE 还包含 CRC 检测和产生、PID 验证 / 产生、bit-stuffing 和 debit-stuffing 功能。

双缓冲

当要进行等时传输或大批量传输时，建议使用有双缓冲功能的端点并使能其双缓冲功能。在 OUT 事务处理期间，双缓冲中的一个缓冲器用来存储之前由 USB 主机发送的数据用于单片机处理；并且在另一个缓冲器中，硬件会确保端点继续接收当前数据包，进行 IN 事务处理时亦然。使用双缓冲器功能可达到数据传输的最大速率。关于双缓冲功能的用法请参考 USBEPnCSR 寄存器中的 UDBTG 和 MDBTG 控制位的相关描述，其中 n 的范围是 4 ~ 7。

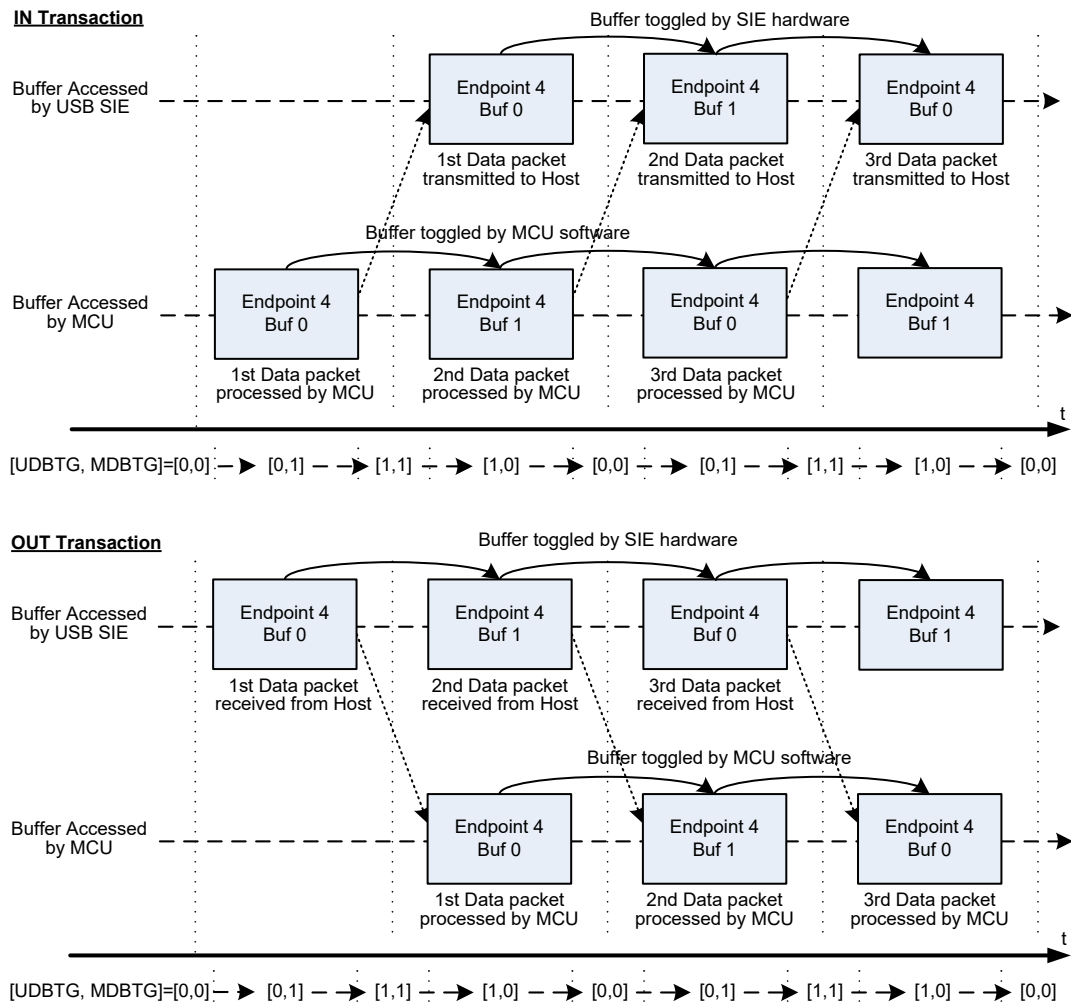


图 172. 双缓冲工作范例

## 挂起模式和唤醒

根据 USB 的规格, USB 设备必须在经过 3 ms 总线空闲时间后进入挂起模式。当 USB 设备进入挂起模式, USB 总线的电流消耗不能大于 500  $\mu$ A, 以满足挂起模式下电流规格要求。如果总线空闲状态持续 3 ms, USB 控制电路将产生挂起中断。此时, USBCSR 寄存器中的 LPMODE 和 PDWN 位必须由软件置 1。通过 LPMODE 位决定 USB 控制器是否进入低功耗模式 (将 USB 总线保持在复位状态), 而 PDWN 位用来决定内置的 USB 全速收发器是否关闭。

有两种方式使 USB 主机唤醒 USB 设备。一种是发送一个 USB 复位信号 SE0, 另一种是发送一个 USB 设备恢复信号 K-state。唤醒信号产生后, 不论是检测到 SE0 信号还是检测到 K-state 信号, USB 设备都将被唤醒。

## 远程唤醒

USB 设备具有远程唤醒功能, 可以通过置 USBCSR 寄存器中的 GENRSM 位为 1 来发送一个恢复请求信号以唤醒 USB 主机。一旦 USB 主机从 USB 设备接收到远程唤醒信号, 将发送一个恢复信号给 USB 设备。

## 寄存器列表

下表显示了 USB 寄存器及其复位值。

表 65. USB 寄存器列表

寄存器	偏移量	描述	复位值
USBCSR	0x000	USB 控制和状态寄存器	0x0000_00X6
USBIER	0x004	USB 中断使能寄存器	0x0000_0000
USBISR	0x008	USB 中断状态寄存器	0x0000_0000
USBFCR	0x00C	USB 帧计数寄存器	0x0000_0000
USBDEVAR	0x010	USB 设备地址寄存器	0x0000_0000
USBEP0CSR	0x014	USB 端点 0 控制和状态寄存器	0x0000_0002
USBEP0IER	0x018	USB 端点 0 中断使能寄存器	0x0000_0000
USBEP0ISR	0x01C	USB 端点 0 中断状态寄存器	0x0000_0000
USBEP0TCR	0x020	USB 端点 0 传输计数寄存器	0x0000_0000
USBEP0CFGR	0x024	USB 端点 0 配置寄存器	0x8000_0002
USBEP1CSR	0x028	USB 端点 1 控制和状态寄存器	0x0000_0002
USBEP1IER	0x02C	USB 端点 1 中断使能寄存器	0x0000_0000
USBEP1ISR	0x030	USB 端点 1 中断状态寄存器	0x0000_0000
USBEP1TCR	0x034	USB 端点 1 传输计数寄存器	0x0000_0000
USBEP1CFGR	0x038	USB 端点 1 配置寄存器	0x1000_03FF
USBEP2CSR	0x03C	USB 端点 2 控制和状态寄存器	0x0000_0002
USBEP2IER	0x040	USB 端点 2 中断使能寄存器	0x0000_0000
USBEP2ISR	0x044	USB 端点 2 中断状态寄存器	0x0000_0000
USBEP2TCR	0x048	USB 端点 2 传输计数寄存器	0x0000_0000
USBEP2CFGR	0x04C	USB 端点 2 配置寄存器	0x1000_03FF
USBEP3CSR	0x050	USB 端点 3 控制和状态寄存器	0x0000_0002
USBEP3IER	0x054	USB 端点 3 中断使能寄存器	0x0000_0000
USBEP3ISR	0x058	USB 端点 3 中断状态寄存器	0x0000_0000
USBEP3TCR	0x05C	USB 端点 3 传输计数寄存器	0x0000_0000



寄存器	偏移量	描述	复位值
USBEP3CFGR	0x060	USB 端点 3 配置寄存器	0x1000_03FF
USBEP4CSR	0x064	USB 端点 4 控制和状态寄存器	0x0000_0002
USBEP4IER	0x068	USB 端点 4 中断使能寄存器	0x0000_0000
USBEP4ISR	0x06C	USB 端点 4 中断状态寄存器	0x0000_0000
USBEP4TCR	0x070	USB 端点 4 传输计数寄存器	0x0000_0000
USBEP4CFGR	0x074	USB 端点 4 配置寄存器	0x1000_03FF
USBEP5CSR	0x078	USB 端点 5 控制和状态寄存器	0x0000_0002
USBEP5IER	0x07C	USB 端点 5 中断使能寄存器	0x0000_0000
USBEP5ISR	0x080	USB 端点 5 中断状态寄存器	0x0000_0000
USBEP5TCR	0x084	USB 端点 5 传输计数寄存器	0x0000_0000
USBEP5CFGR	0x088	USB 端点 5 配置寄存器	0x1000_03FF
USBEP6CSR	0x08C	USB 端点 6 控制和状态寄存器	0x0000_0002
USBEP6IER	0x090	USB 端点 6 中断使能寄存器	0x0000_0000
USBEP6ISR	0x094	USB 端点 6 中断状态寄存器	0x0000_0000
USBEP6TCR	0x098	USB 端点 6 传输计数寄存器	0x0000_0000
USBEP6CFGR	0x09C	USB 端点 6 配置寄存器	0x1000_03FF
USBEP7CSR	0x0A0	USB 端点 7 控制和状态寄存器	0x0000_0002
USBEP7IER	0x0A4	USB 端点 7 中断使能寄存器	0x0000_0000
USBEP7ISR	0x0A8	USB 端点 7 中断状态寄存器	0x0000_0000
USBEP7TCR	0x0AC	USB 端点 7 传输计数寄存器	0x0000_0000
USBEP7CFGR	0x0B0	USB 端点 7 配置寄存器	0x1000_03FF

寄存器描述

USB 控制和状态寄存器 – USBCSR

该寄存器定义了 USB 控制位和 USB 数据线状态。

偏移量: 0x000  
复位值: 0x0000\_00X6

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位				DPWKEN	DPPUEN	SRAMRSTC	ADRSET
类型 / 复位	7	6	5	4	3	2	1	0
	RXDM	RXDP	GENRSM	保留位	LPMODE	PDWN	FRES	保留位
类型 / 复位	RO	X RO	X RW	0	RW	0 RW	1 RW	1

位	字段	描述
[11]	DPWKEN	DP 唤醒使能位 0: DP 唤醒除能 1: DP 唤醒使能
[10]	DPPUEN	DP 上拉使能位 0: DP 上拉除能 1: DP 上拉使能
[9]	SRAMRSTC	EP_SRAM 复位条件 0: 当 (DP, DM) = (0, 0) 时, 复位 EP_SRAM 1: 无论 (DP, DM) 为何状态, 用户都可访问 EP_SRAM
[8]	ADRSET	设备地址控制位 此位用来控制 USB SIE 何时将设备地址更新为 USBDEVAR 寄存器的值。 0: 当有一个地址被写入 USBDEVAR 寄存器之后, SIE 立即更新设备地址。 1: USB 主机通过 IN 操作成功从设备读取数据之后 SIE 更新设备地址。 此位在设备地址更新后由 SIE 清零。
[7]	RXDM	接收的 DM 线状态 在挂起程序结束时, 此位用来观察 DM 数据线的状态以判断唤醒事件是否已经发生。
[6]	RXDP	接收的 DP 线状态 在挂起程序结束时, 此位用来观察 DP 数据线的状态以判断唤醒事件是否已经发生。
[5]	GENRSM	恢复请求发生控制位 通过向此位写 1 来产生一个恢复请求信号给 USB 主机。USB 远程唤醒功能总是使能的。USB 主机发送的恢复信号被接收时, 此位清零。
[3]	LPMODE	低功耗模式控制位 此位用来决定 USB 工作模式。此位置位 1 将使 USB 强制进入低功耗模式。当 USB 总线唤醒事件被硬件检测到时, 此位应由软件清零。 0: 退出低功耗模式 1: 进入低功耗模式

位	字段	描述
[2]	PDWN	暂停模式控制位 此位用来关闭 USB 总线功能。此位置位将暂停全速 USB PHY 收发器，这将断开 USB PHY 收发器与 USB 总线的连接。 0: 退出暂停模式 1: 进入暂停模式
[1]	FRES	强制 USB 复位控制 此位用来复位 USB 电路。置位此位将强制 USB 进入复位状态直到软件清零此位。如果 USBIER 寄存器中的相关中断使能位置 1，则将产生 USB 复位中断。所有相关的 USB 寄存器都将复位到默认值。 0: 解除 USB 复位 1: 强制 USB 复位

表 66. 恢复事件检测

[RXDP, RXDM] 状态	唤醒事件	恢复后软件需进行的动作
00	Root 复位	无
10	无 ( 总线噪声 )	返回到挂起模式
01	Root 恢复	无
11	不允许 ( 总线噪声 )	返回到挂起模式

USB 中断使能寄存器 – USBIER

该寄存器定义了 USB 中断使能控制。

偏移量： 0x004  
复位值： 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	EP7IE	EP6IE	EP5IE	EP4IE	EP3IE	EP2IE	EP1IE	EP0IE
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0
类型 / 复位	7	6	5	4	3	2	1	0
	保留位		ESOFIE	SUSPIE	RSMIE	URSTIE	SOFIE	UGIE
类型 / 复位	保留位		RW	0	RW	0	RW	0
	保留位		RW	0	RW	0	RW	0

位	字段	描述
[15:8]	EPnIE	端点 n 中断使能控制 (n = 0 ~ 7) 0: 除能 1: 使能
[5]	ESOFIE	预期帧开始 (ESOF) 中断使能控制位 0: 除能 1: 使能
[4]	SUSPIE	挂起中断使能控制位 0: 除能 1: 使能
[3]	RSMIE	恢复中断使能控制位 0: 除能 1: 使能
[2]	URSTIE	USB 复位中断使能控制位 0: 除能 1: 使能
[1]	SOFIE	帧开始 (SOF) 中断使能控制位 0: 除能 1: 使能
[0]	UGIE	USB 总中断使能控制位 0: 除能 1: 使能 此位必须置 1 来使能相关 USB 中断功能。如果此位清零, 相关中断将不会产生, 然而, 相关中断标志位仍然会置起。

## USB 中断状态寄存器 – USBISR

该寄存器定义了 USB 中断状态。

偏移量: 0x008

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	EP7IF	EP6IF	EP5IF	EP4IF	EP3IF	EP2IF	EP1IF	EP0IF
类型 / 复位	WC	0	WC	0	WC	0	WC	0
	7	6	5	4	3	2	1	0
	保留位		ESOFIF	SUSPIF	RSMIF	URSTIF	SOFIF	保留位
类型 / 复位			RW	0	WC	0	WC	0

位	字段	描述
[15:8]	EPnIF	端点 n 中断标志位 (n = 0 ~ 7) 此位由硬件置位, 表示相关端点中断产生。 向此位写 1 清零。注意, 仅当 USBEPnISR 寄存器中的端点中断状态位是 0 时, 此中断标志位才可清零。
[5]	ESOFIF	预期帧开始 (ESOF) 中断标志位 当一个 SOF 包预期被接收时, 此位由硬件置位。USB 主机每毫秒发送一个 SOF 包。如果 USB 设备硬件没有正确接收 SOF 包, 当 USBIER 寄存器中的 ESOFIE 位置 1 时, 将产生 ESOF 中断。如果产生三个连续的 ESOF 中断, 也就是说 SOF 包已经被丢失 3 次, 那么 SUSPIF 位将被置 1。如果定时器还没有被锁, 也会导致 SOF 包发生丢失, 此位会被置 1。 此位可读写。但只可写入 0, 写 1 将无效。
[4]	SUSPIF	挂起中断标志位 当超过 3 ms 没有数据传输时, 此位将由硬件置位, 说明 USB 主机已经发送了挂起请求。USB 复位后挂起条件检测功能立刻使能。 此位通过写 1 清零。
[3]	RSMIF	恢复中断标志位 此位由硬件置位, 表示设备恢复已经发生。 此位通过写 1 清零。
[2]	URSTIF	USB 复位中断标志位 当检测到 USB 复位时, 此位由硬件置位。当发生 USB 复位, 内部协议状态机将复位, 当 USBIER 寄存器中的 URSTIE 位置 1 时, 将产生 USB 复位中断。数据接收和发送都将除能直到 URSTIF 位清零。除了 USB 设备地址 (USBDEVAR) 之外, USB 设置相关的寄存器 (USBCSR、USBIER、USBISR、USBFCR 和 USBDEVAR) 都不会因 USB 复位事件而复位, 这确保 USB 复位中断能安全执行, 并且让 USB 复位发生之前的任何数据处理可以完全由软件访问。因此, 单片机必须合理地复位这些寄存器。USB 端点相关的寄存器 (USBEPnCSR、USBEPnISR 和 USBEPnTCR) 也因复位事件而复位, 然而, 端点配置 (USBEPnCFGR) 及中断使能 (USBEPnIER) 寄存器则不受 USB 复位的影响, 仍然保持不变。 此位通过写 1 清零。

位	字段	描述
[1]	SOFIF	帧开始 SOF 中断标志位 当 SOF 包被接收时，此位由硬件置位。 此位通过写 1 清零。

## USB 帧计数寄存器 – USBFCR

该寄存器定义了丢失 SOF (Start-of-Frame) 的数量和 USB 帧计数。

偏移量: 0x00C

复位值: 0x0000\_0000

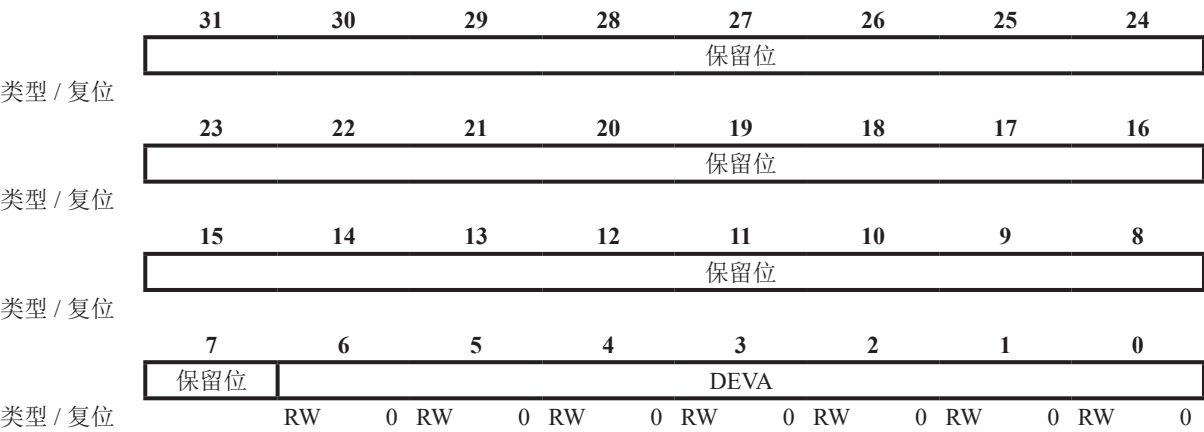
	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位					LSOF		SOFLCK
类型 / 复位						RO	0 RO	0 RO 0
	15	14	13	12	11	10	9	8
	保留位					FRNUM		
类型 / 复位						RO	0 RO	0 RO 0
	7	6	5	4	3	2	1	0
	FRNUM							
类型 / 复位	RO	0 RO	0 RO	0 RO	0 RO	0 RO	0 RO	0

位	字段	描述
[18:17]	LSOF	丢失 SOF 包的数量 ESOFIF 位每置位一次，这些位由硬件写入且加 1。这些位用来计算丢失的 SOF 包的数量。当 SOF 包被接收到时，这些位被清零。
[16]	SOFLCK	SOF 锁存标志位 当 SOF 包在帧定时器超时之前被接收，此位由硬件置位。一旦此标志位置 1，由 USB 主机发送的帧数量将被载入到 USBFCR 寄存器中的帧数量字段。如果在 1 ms 帧时间段内，没有 SOF 包被接收，那么此位将清零。
[10:0]	FRNUM	帧数量 此字段存储由 USB 主机发送的帧数量。

USB 设备地址寄存器 – USBDEVAR

该寄存器定义了 USB 设备地址。

偏移量： 0x010  
复位值： 0x0000\_0000

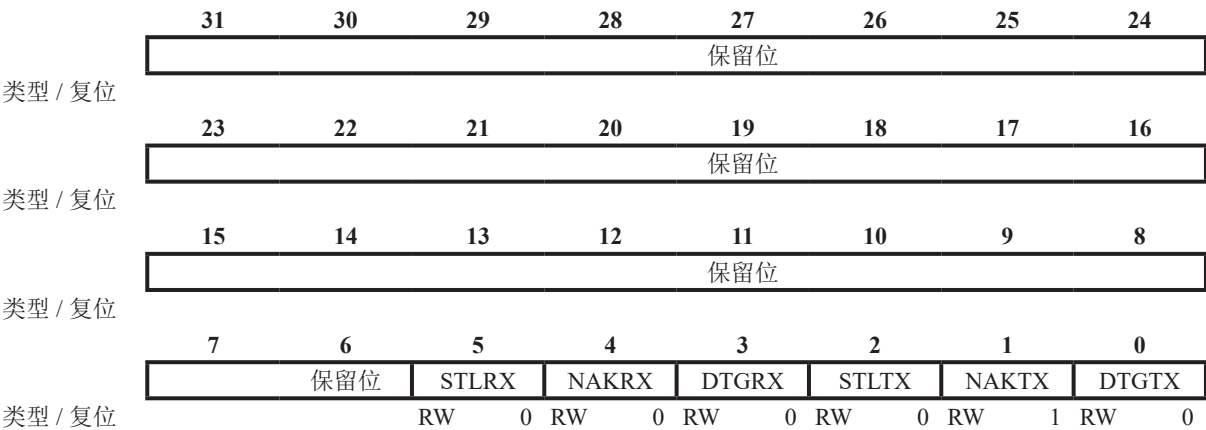


位	字段	描述
[6:0]	DEVA	设备地址 此字段用来定义 USB 设备地址。当 USB 复位事件发生时，此字段清零。

USB 端点 0 控制和状态寄存器 – USBEP0CSR

该寄存器定义了端点 0 控制和状态位。

偏移量： 0x014  
复位值： 0x0000\_0002



位	字段	描述
[5]	STLRX	接收 (OUT) 传输的 STALL 状态 当功能错误被检测到，此位由应用程序置 1，在 OUT 事务处理的握手阶段将返回一个 STALL 信号。也就是说，来自于 USB 主机的控制请求不被 USB 设备支持。当 SETUP 令牌被接收时，STALL 状态由硬件电路清零。 此位可读写，仅当写 1 时状态切换。
[4]	NAKRX	接收 (OUT) 传输的 NAK 状态 在 ACK 信号被发送给主机之后，此位由硬件电路从 0 切换到 1，在 OUT 事务处理的握手阶段将产生一个 NAK 信号。这意味着 USB 设备将暂时不能接收来自 USB 主机的数据。因此，会需要更多的时间来处理所接收的数据。 此位可读写，仅当写 1 时状态切换。
[3]	DTGRX	接收 (OUT) 传输的数据切换状态 此位包含数据切换位的预期值 (0 = DATA0, 1 = DATA1)，用于下一个要接收的数据包。当当前有效数据包被接收，且 USB 设备已发送给 USB 主机相应的 ACK 信号，硬件将切换此位，设备将准备接收下一个数据包。对于端点 0，在 SETUP 令牌已被接收且端点 0 已被寻址后，硬件会将此位切换到 1。此位也可由软件对其初始化以适用于特定的应用。 此位可读写，仅当写 1 时状态切换。
[2]	STLTX	发送 (IN) 传输的 STALL 状态 当功能错误被检测到，此位由应用程序置 1，将返回一个 STALL 信号以响应 IN 令牌。这意味着 USB 设备不能发送数据。当新的 SETUP 令牌被接收到时，STALL 状态由硬件电路清除。 此位可读写，仅当写 1 时状态切换。
[1]	NAKTX	发送 (IN) 传输的 NAK 状态 在 ACK 信号被设备接收之后，此位由硬件电路从 0 切换到 1，在 IN 事务处理的握手阶段将产生一个 NAK 信号。这意味着 USB 设备暂时不能向 USB 主机发送数据。因此，应用软件有更多的时间为发送数据作准备。 此位可读写，仅当写 1 时状态切换。



位	字段	描述
[0]	DTGTX	发送 (IN) 传输的数据切换状态 此位包含数据切换位所需要的值 (0 = DATA0, 1 = DATA1)，用于下一个要发送的数据包。当当前数据包被 USB 设备发送，且由 USB 主机发送的 ACK 信号被接收，硬件将切换此位，并将发送下一个数据包。对于端点 0，在 SETUP 令牌已被接收且端点 0 已被寻址后，硬件会将此位切换到 1。在特定条件下，此位也可由软件对其初始化。 此位可读写，仅当写 1 时状态切换。

## USB 端点 0 中断使能控制寄存器 – USBEP0IER

该寄存器定义了端点 0 中断使能控制位。

偏移量: 0x018

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位				ZLRXIE	SDERIE	SDRXIE	STRXIE
类型 / 复位					RW	0	RW	0
	7	6	5	4	3	2	1	0
	URIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[11]	ZLRXIE	零长度数据接收中断使能控制位 0: 除能 1: 使能
[10]	SDERIE	SETUP 数据错误中断使能控制位 0: 除能 1: 使能
[9]	SDRXIE	SETUP 数据接收中断使能控制位 0: 除能 1: 使能
[8]	STRXIE	SETUP 令牌接收中断使能控制位 0: 除能 1: 使能
[7]	URIE	USB 错误中断使能控制位 0: 除能 1: 使能
[6]	STLIE	STALL 发送中断使能控制位 0: 除能 1: 使能

位	字段	描述
[5]	NAKIE	NAK 发送中断使能控制位 0: 除能 1: 使能
[4]	IDTXIE	IN 数据发送中断使能控制位 0: 除能 1: 使能
[3]	ITRXIE	IN 令牌接收中断使能控制位 0: 除能 1: 使能
[2]	ODOVIE	OUT 数据缓冲器溢出中断使能控制位 0: 除能 1: 使能
[1]	ODRXIE	OUT 数据接收中断使能控制位 0: 除能 1: 使能
[0]	OTRXIE	OUT 令牌接收中断使能控制位 0: 除能 1: 使能

## USB 端点 0 中断状态寄存器 – USBEP0ISR

该寄存器定义了端点 0 中断状态。

偏移量： 0x01C

复位值： 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位							
15	14	13	12	11	10	9	8
保留位				ZLRXIF	SDERIF	SDRXIF	STRXIF
				WC	0	WC	0
7	6	5	4	3	2	1	0
UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
WC	0	WC	0	WC	0	WC	0

位	字段	描述
[11]	ZLRXIF	零长度数据接收中断标志位 当接收到一个零长度数据包时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。
[10]	SDERIF	SETUP 数据错误中断标志位 当 SETUP 数据包的长度不是 8 字节时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。
[9]	SDRXIF	SETUP 数据接收中断标志位 当接收到来自于 USB 主机的一个 SETUP 数据包时，此位被硬件置位。当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。在下一个 SETUP 数据包被接收前，如果已接收到的 SETUP 数据没有被应用软件访问，则 SETUP 数据缓冲器将被覆盖。
[8]	STRXIF	SETUP 令牌接收中断标志位 当 SETUP 令牌被接收时，此位由硬件置位。写 1 时清零。
[7]	UERIF	USB 错误中断标志位 在端点 0 事务处理过程中，当有错误发生时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。
[6]	STLIF	STALL 发送中断标志位 当 STALL 信号被发送以响应 IN 或 OUT 事务处理时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。
[5]	NAKIF	NAK 发送中断标志位 当 NAK 信号被发送以响应 IN 或 OUT 事务处理时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。
[4]	IDTXIF	IN 数据发送中断标志位 当数据包被发送到 USB 主机且从 USB 主机发送的 ACK 信号被接收时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。
[3]	ITRXIF	IN 令牌接收中断标志位 当接收到来自于 USB 主机的 IN 令牌时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。

位	字段	描述
[2]	ODOVIF	OUT 数据缓冲器溢出中断标志位 当接收到数据字节的数量大于端点缓冲器容量时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。
[1]	ODRXIF	OUT 数据接收中断标志位 当成功接收到来自于 USB 主机的数据包且 ACK 信号被发送到 USB 主机时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。
[0]	OTRXIF	OUT 令牌接收中断标志位 当接收到来自于 USB 主机的 OUT 令牌时，此位由硬件置位。 当 SETUP 令牌被接收时此位由硬件清零，或者通过写 1 清零。

USB 端点 0 传输计数寄存器 – USBEP0TCR

该寄存器定义了端点 0 数据传输字节数。

偏移量： 0x020  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位	RXCNT							
类型 / 复位		RO	0	RO	0	RO	0	RO	0
	15	14	13	12	11	10	9	8	
	保留位								
类型 / 复位									
	7	6	5	4	3	2	1	0	
	保留位	TXCNT							
类型 / 复位		RW	0	RW	0	RW	0	RW	0

位	字段	描述
[22:16]	RXCNT	接收字节数 这个字段包含了在之前的 SETUP 事务处理中被端点 0 接收的数据字节总数。
[6:0]	TXCNT	发送字节总数 这个字段包含了在下个 IN 令牌中端点 0 要发送的数据字节总数。如果此字段为 0，表示将发送 0 长度数据包。

## USB 端点 0 配置寄存器 – USBEP0CFGR

该寄存器定义了端点 0 的配置选项。

偏移量： 0x024

复位值： 0x8000\_0002

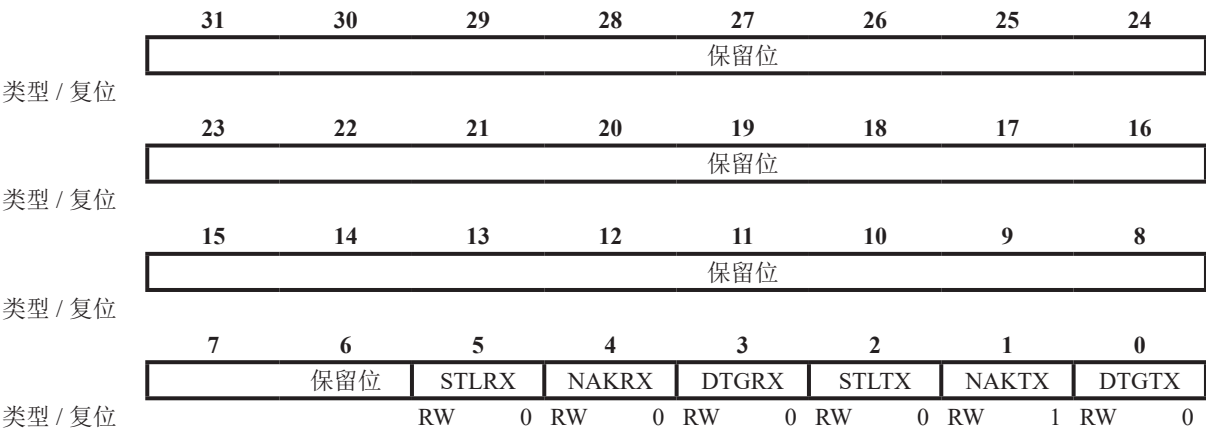
	31	30	29	28	27	26	25	24
	EPEN	保留位				EPADR		
类型 / 复位	RO 1				RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	保留位							EPLEN
类型 / 复位								RW 0
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EPBUFA							
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 0

位	字段	描述
[31]	EPEN	端点使能控制位 此位由硬件始终置 1，使端点 0 一直使能。
[27:24]	EPADR	端点地址 该字段由硬件始终清 0。
[16:10]	EPLEN	端点缓冲器长度 该字段用来指定控制传输包的大小，在 USB 全速标准规格中，可设置为 8、16、32、或 64 个字节。
[9:0]	EPBUFA	端点缓冲器开始地址 此字段用来指定在 EP_SRAM 中的端点 0 缓冲器的起始地址。起始地址从 0x008 开始并 对齐到 4-byte 边界。 端点 0 IN 缓冲器起始地址 = EPBUFA 端点 0 OUT 缓冲器起始地址 = EPBUFA + EPLEN

USB 端点 1 ~ 3 控制和状态寄存器 – USBEPnCSR, n = 1 ~ 3

该寄存器定义了端点 1 ~ 3 控制和状态位。

偏移量: 0x028 (n = 1), 0x03C (n = 2), 0x050 (n = 3)  
复位值: 0x0000\_0002



位	字段	描述
[5]	STLRX	接收传输的 STALL 位 当功能错误被检测到，此位由应用程序置 1。 此位可读写，仅当写 1 时状态切换。此位也可由软件对其初始化以适用于特定的应用。
[4]	NAKRX	接收传输的 NAK 位 当已接收到一个数据包且在 OUT 事务处理的握手阶段已发送一个 ACK 信号到主机后，此位由硬件电路从 0 切换到 1 从而将产生一个 NAK 信号。这意味着 USB 设备需要更多的时间来处理已经接收的数据，将暂时不能接收来自 USB 主机的数据。 此位可读写，仅当写 1 时状态切换。
[3]	DTGRX	接收传输的数据切换位 此位包含数据切换位的预期值 (0 = DATA0, 1 = DATA1)，用于下一个要接收的数据包。 当当前有效数据包被接收且 USB 设备发送给 USB 主机相应的 ACK 信号后，硬件将切换此位，设备将准备接收下一个数据包。 此位可读写，仅当写 1 时状态切换。在特定条件下，此位也可由软件对其初始化。
[2]	STLTX	发送传输的 STALL 位 当功能错误被检测到，此位由应用程序置 1。 此位可读写，仅当写 1 时状态切换。在特定情况下，此位也可由软件对其初始化。
[1]	NAKTX	发送传输的 NAK 位 当一个数据包已被发送且来自于主机 ACK 信号在 IN 事务处理的握手阶段已被接收，此位由硬件电路从 0 切换到 1 从而产生一个 NAK 信号。这意味着 USB 设备暂时不能向 USB 主机发送数据直到应用软件准备好要发送的数据为止。 此位可读写，仅当写 1 时状态切换。
[0]	DTGTX	发送传输的数据切换位 此位包含数据切换位所需要的值 (0 = DATA0, 1 = DATA1)，用于下一个要发送的数据包。 当当前数据包被 USB 设备发送，由 USB 主机发送的 ACK 信号被接收后，硬件将切换此位，并将发送下一个数据包。 此位可读写，仅当写 1 时状态切换。在特定情况下，此位也可由软件对其初始化。

## USB 端点 1 ~ 3 中断使能寄存器 – USBEPnIER, n = 1 ~ 3

该寄存器定义了端点 1 ~ 3 中断使能控制位。

偏移量: 0x02C (n = 1), 0x040 (n = 2), 0x054 (n = 3)

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位							
15	14	13	12	11	10	9	8
保留位							
7	6	5	4	3	2	1	0
UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW

位	字段	描述
[7]	UERIE	USB 错误中断使能控制位 0: 除能 1: 使能
[6]	STLIE	STALL 发送中断使能控制位 0: 除能 1: 使能
[5]	NAKIE	NAK 发送中断使能控制位 0: 除能 1: 使能
[4]	IDTXIE	IN 数据发送中断使能控制位 0: 除能 1: 使能
[3]	ITRXIE	IN 令牌接收中断使能控制位 0: 除能 1: 使能
[2]	ODOVIE	OUT 数据缓冲器溢出中断使能控制位 0: 除能 1: 使能
[1]	ODRXIE	OUT 数据接收中断使能控制位 0: 除能 1: 使能
[0]	OTRXIE	OUT 令牌接收中断使能控制位 0: 除能 1: 使能

## USB 端点 1 ~ 3 中断状态寄存器 – USBEPnISR, n = 1 ~ 3

该寄存器定义了端点 1 ~ 3 点中断状态。

偏移量: 0x030 (n = 1), 0x044 (n = 2), 0x058 (n = 3)

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位							
15	14	13	12	11	10	9	8
保留位							
7	6	5	4	3	2	1	0
UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
WC	0	WC	0	WC	0	WC	0

位	字段	描述
[7]	UERIF	USB 错误中断标志位 当在事务处理期间有错误发生时, 此位由硬件置位。通过写 1 清零。
[6]	STLIF	STALL 发送中断标志位 当 STALL 令牌被发送来响应 IN 或 OUT 令牌时, 此位由硬件置位。通过写 1 清零。
[5]	NAKIF	NAK 发送中断标志位 当 NAK 令牌被发送来响应 IN 或 OUT 令牌时, 此位由硬件置位。通过写 1 清零。
[4]	IDTXIF	IN 数据发送中断标志位 当数据包被成功发送给主机以作为对 IN 令牌的响应且 ACK 令牌被端点接收到时, 此位由硬件置位。通过 1 时清零。
[3]	ITRXIF	IN 令牌接收中断标志位 当端点接收到来自于主机的 IN 令牌时, 此位由硬件置位。通过写 1 清零。
[2]	ODOVIF	OUT 数据缓冲器溢出中断标志位 当接收到的数据字节数大于相关的端点 OUT 数据缓冲器的容量时。此位由硬件置位。通过写 1 清零。
[1]	ODRXIF	OUT 数据接收中断标志位 对于 OUT 令牌, 当成功接收到主机发送的数据包和当端点 n ACK 信号被发送给主机时, 此位由硬件电路置位。通过写 1 清零。
[0]	OTRXIF	OUT 令牌接收中断标志位 当端点接收到来自主机的 OUT 令牌时, 此位被置位。通过写 1 清零。



USB 端点 1 ~ 3 传输计数寄存器 – USBEPnTCR, n = 1 ~ 3

该寄存器定义了端点 1 ~ 3 传输字节数。

偏移量: 0x034 (n = 1), 0x048 (n = 2), 0x05C (n = 3)

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							TCNT
类型 / 复位								RW 0
	7	6	5	4	3	2	1	0
	TCNT							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[8:0]	TCNT	传输字节数 该字段包含了在之前的 OUT 事务处理中由端点 n 接收到的数据字节数，或在下一个 IN 事务处理中由端点 n 发送的数据字节数。

USB 端点 1 ~ 3 配置寄存器 – USBEPnCFGR, n = 1 ~ 3

该寄存器定义了端点 1 ~ 3 的配置。

偏移量: 0x038 (n = 1), 0x04C (n = 2), 0x060 (n = 3)  
复位值: 0x1000\_03FF

	31	30	29	28	27	26	25	24
	EPEN	保留位	EPTYPE	EPDIR	EPADR			
类型 / 复位	RW 0		RW 0	RW 1	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	保留位						EPLEN	
类型 / 复位							RW 0	
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	EPBUFA							
类型 / 复位	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

位	字段	描述
[31]	EPEN	端点使能控制位 0: 除能端点 n 1: 使能端点 n
[29]	EPTYPE	端点传输类型 此位由硬件清零，用来定义端点 n 传输类型为中断或批量传输类型。
[28]	EPDIR	端点传输方向 0: OUT 1: IN
[27:24]	EPADR	端点地址 EPADR 字段由应用软件分配用来指定端点 n 地址。需注意的是，EPADR 字段不能清为 0，否则，端点将被除能。
[16:10]	EPLEN	端点缓冲器长度 此字段用来定义端点 n 数据包大小。此字段定义的值必须是 4 的倍数。此字段最多可达 64 字节，这也是 USB 全速标准规格定义的最大容载量。注意，EPLEN 值不能清除为 0，否则将除能端点。
[9:0]	EPBUFA	端点缓冲器起始地址 此字段用来定义端点 n 数据缓冲器的起始地址。在容量为 1024 字节的 EP_SRAM 中，起始地址范围是 0x008 ~ 0x3FC，且该字段值必须是 4 的倍数。

## USB 端点 4 ~ 7 控制和状态寄存器 – USBEPnCSR, n = 4 ~ 7

该寄存器定义了端点 4 ~ 7 控制和状态位。

偏移量: 0x064 (n = 4), 0x078 (n = 5), 0x08C (n = 6), 0x0A0 (n = 7)

复位值: 0x0000\_0002

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
UDBTG	MDBTG	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
RW	0	RW	0	RW	0	RW	0
类型 / 复位							

位 字段 描述

[7] UDBTG USB 双缓冲器切换位

双缓冲器功能使能时，UDBTG 和 MDBTG 位用来指定哪个数据缓冲器被 USB SIE 硬件访问，哪个数据缓冲器被 MCU 软件访问。在当前缓冲器工作完成时，UDBTG 位将由 SIE 硬件电路切换。此位被 SIE 切换后，硬件电路会自动发送一个 NAK 信号给 USB 主机。因此，数据传输将暂时停止直到另一个缓冲器内的数据被正确设置，在这之后 MDBTG 位通过 CPU 应用软件切换。

下表显示了双缓冲器的操作和在 IN 或 OUT 事务处理时 UDBTG 和 MDBTG 位的状态。

事务处理类型	UDBTG	MDBTG	SIE 读缓冲器	MCU 写缓冲器
IN	0	0	无 *	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	0	EP_BUF1	EP_BUF0
	1	1	无 *	EP_BUF1

事务处理类型	UDBTG	MDBTG	SIE 写缓冲器	MCU 读缓冲器
OUT	0	0	无 *	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	0	EP_BUF1	EP_BUF0
	1	1	无 *	EP_BUF1

\* 表示 USB 设备通过硬件电路发送一个 NAK 信号给 USB 主机。

用于双缓冲功能的 UDBTG 和 MDBTG 位的设置步骤如下例所示：

[UDBTG, MDBTG] = [0, 0]→[0, 1]→[1, 1]→[1, 0]→[0, 0]→[0, 1]→[1, 1]→[1, 0]→...

[6] MDBTG MCU 双缓冲器切换位

如果双缓冲功能使能，MDBTG 位用来指定哪个数据缓冲器被 MCU 访问。在被 MCU 访问的当前缓冲器中的数据正确设置之后，此位可通过 CPU 应用软件改变状态来切换到另一个缓冲器。双缓冲功能操作与 UDBTG 和 MDBTG 位的关系请参考前面 UDBTG 位定义里的表格。

位	字段	描述
[5]	STLRX	接收传输的 STALL 位 当功能错误被检测到，此位由应用程序置 1。 此位可读写，仅当写 1 时状态切换。在特定条件下，此位也可由软件对其初始化。
[4]	NAKRX	接收传输的 NAK 位 当已接收到一个数据包且在 OUT 事务处理的握手阶段已发送一个 ACK 信号到主机后，此位由硬件电路从 0 切换到 1 从而将产生一个 NAK 信号。这意味着 USB 设备需要更多的时间来处理已经接收的数据，将暂时不能接收来自 USB 主机的数据。如果端点被定义成等时传输类型，此位将不可用。经过完整的事务处理之后，硬件不会改变 NAKRX 位的状态。此位可读写，仅当写 1 时状态切换。
[3]	DTGRX	接收传输的数据切换位 如果端点不用于等时传输，此位可用。此位包含数据切换位的预期值 (0 = DATA0, 1 = DATA1)，用于下一个要接收的数据包。当当前有效数据包被接收且 USB 设备已发送给 USB 主机相应的 ACK 信号后，硬件将切换此位，设备将准备接收下一个数据包。如果端点设置为等时传输类型，因为无数据切换，所以此位不可用。只有 DATA0 包在正常等时传输器中可以传输。 此位可读写，仅当写 1 时状态切换。在特定条件下，此位也可由软件对其初始化。
[2]	STLTX	发送传输的 STALL 位 当功能错误被检测到，此位由应用程序置 1。 此位可读写，仅当写 1 时状态切换。在特定情况下，此位也可由软件对其初始化。
[1]	NAKTX	发送传输的 NAK 位 当一个数据包已被发送，且来自于主机的 ACK 信号在 IN 事务处理对此端点寻址的握手阶段已被接收，此位由硬件电路从 0 切换到 1 从而将产生一个 NAK 信号。这意味着 USB 设备暂时不能向 USB 主机发送数据直到应用软件准备好要发送的数据为止。如果端点定义为等时传输类型，则此位不可用。在一个完整的事务处理之后，硬件不会改变 NAKTX 位的状态。 此位可读写，仅当写 1 时状态切换。在特定情况下，此位也可由软件对其初始化。
[0]	DTGTx	发送传输的数据切换位 如果端点不用于等时传输，此位可用。此位包含数据切换位所需要的值 (0 = DATA0, 1 = DATA1)，用于下一个要发送的数据包。当当前数据包已被 USB 设备发送，且由 USB 主机发送的 ACK 信号已被接收，硬件将切换此位，并将发送下一个数据包。如果端点设置为等时传输类型，因为无数据切换，所以此位不可用。在正常等时传输中只有 DATA0 包可以传输。 此位可读写，仅当写 1 时状态切换。在特定情况下，此位也可由软件对其初始化。

## USB 端点 4 ~ 7 中断使能寄存器 – USBEPnIER, n = 4 ~ 7

该寄存器定义了端点 4 ~ 7 中断使能控制位。

偏移量: 0x068 (n = 4), 0x07C (n = 5), 0x090 (n = 6), 0x0A4 (n = 7)

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW

位	字段	描述
[7]	UERIE	USB 错误中断使能控制位 0: 除能 1: 使能
[6]	STLIE	STALL 发送中断使能控制位 0: 除能 1: 使能
[5]	NAKIE	NAK 发送中断使能控制位 0: 除能 1: 使能
[4]	IDTXIE	IN 数据发送中断使能控制位 0: 除能 1: 使能
[3]	ITRXIE	IN 令牌接收中断使能控制位 0: 除能 1: 使能
[2]	ODOVIE	OUT 数据缓冲器溢出中断使能控制位 0: 除能 1: 使能
[1]	ODRXIE	OUT 数据接收中断使能控制位 0: 除能 1: 使能
[0]	OTRXIE	OUT 令牌接收中断使能控制位 0: 除能 1: 使能

## USB 端点 4 ~ 7 中断状态寄存器 – USBEPnISR, n = 4 ~ 7

该寄存器定义了端点 4 ~ 7 端点中断状态。

偏移量: 0x06C (n = 4), 0x080 (n = 5), 0x094 (n = 6), 0x0A8 (n = 7)

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
WC	0	WC	0	WC	0	WC	0
类型 / 复位							

位	字段	描述
[7]	UERIF	USB 错误中断标志位 当在事务处理期间有错误发生时，此位由硬件置位。 通过写 1 清零。
[6]	STLIF	STALL 发送中断标志位 当 STALL 令牌被发送来响应 IN 或 OUT 令牌时，此位由硬件置位。 通过写 1 清零。
[5]	NAKIF	NAK 发送中断标志位 当 NAK 令牌被发送来响应 IN 或 OUT 令牌时，此位由硬件置位。 通过写 1 清零。
[4]	IDTXIF	IN 数据发送中断标志位 当数据包已被成功发送给主机作为对 IN 令牌的响应且 ACK 令牌被端点接收到时，此位由硬件置位。通过写 1 清零。
[3]	ITRXIF	IN 令牌接收中断标志位 当端点接收到来自主机的 IN 令牌时，此位由硬件置位。 通过写 1 清零。
[2]	ODOVIF	OUT 数据缓冲器溢出中断标志位 当接收到的数据字节数大于相关的端点 OUT 数据缓冲器的大小时，此位由硬件置位。 通过写 1 清零。
[1]	ODRXIF	OUT 数据接收中断标志位 对于 OUT 令牌，当成功接收到主机发送的数据包且 ACK 信号已发送给主机时，此位由硬件电路置位。通过写 1 清零。
[0]	OTRXIF	OUT 令牌接收中断标志位 当端点接收到来自主机的 OUT 令牌时，此位由硬件置位。 通过写 1 清零。

USB 端点 4 ~ 7 传输计数寄存器 – USBEPnTCR, n = 4 ~ 7

该寄存器定义了端点 4 ~ 7 端点传输字节数。

偏移量: 0x070 (n = 4), 0x084 (n = 5), 0x098 (n = 6), 0x0AC (n = 7)  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位						TCNT1	
类型 / 复位							RW	0
	23	22	21	20	19	18	17	16
	TCNT1							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	保留位						TCNT0	
类型 / 复位							RW	0
	7	6	5	4	3	2	1	0
	TCNT0							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[25:16]	TCNT1	缓冲器 1 传输字节数 该字段包含了在之前的 OUT 事务处理中由端点 n 缓冲器 1 接收到的数据字节数，或在下一个 IN 事务处理中由端点 n 缓冲器 1 要发送的数据字节数。
[9:0]	TCNT0	缓冲器 0 传输字节数 该字段包含了在之前的 OUT 事务处理中由端点 n 缓冲器 0 接收到的数据字节数，或在下一个 IN 事务处理中由端点 n 缓冲器 0 要发送的数据字节数。当端点配置为单缓冲传输类型时，仅 TCNT0 字段用于端点数据传输计数。

## USB 端点 4~7 配置寄存器 – USBEPnCFGR, n = 4~7

该寄存器定义了端点 4~7 端点的配置。

偏移量: 0x074 (n = 4), 0x088 (n = 5), 0x09C (n = 6), 0x0B0 (n = 7)

复位值: 0x1000\_03FF

	31	30	29	28	27	26	25	24
	EPEN	保留位	EPTYPE	EPDIR	EPADR			
类型 / 复位	RW 0		RW 0	RW 1	RW 0	RW 0	RW 0	
	23	22	21	20	19	18	17	16
	SDBS	保留位			EPLEN			
类型 / 复位	RW 0				RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	EPBUFA							
类型 / 复位	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

位	字段	描述
[31]	EPEN	端点使能控制位 0: 除能端点 n 1: 使能端点 n
[29]	EPTYPE	端点传输类型 0: 中断或批量传输类型 1: 等时传输类型
[28]	EPDIR	端点传输方向 0: OUT 1: IN
[27:24]	EPADR	端点地址 EPADR 字段由应用软件分配用来指定端点 n 的地址。需注意的是, EPADR 字段不能清为 0, 否则, 端点 n 将被除能。
[23]	SDBS	单缓冲或双缓冲选择 0: 单缓冲 1: 双缓冲 如果 SDBS 位被置为 1, 端点缓冲器大小是 EPLEN 值的两倍: – 端点缓冲器 0 起始地址是 EPBUFA – 端点缓冲器 1 起始地址是 (EPBUFA + EPLEN)
[19:10]	EPLEN	端点缓冲器长度 此字段用来定义端点 n 数据包大小。此字段定义的值必须是 4 的倍数。注意, 如果 EPLEN 值为 0, 则端点将除能。
[9:0]	EPBUFA	端点缓冲器起始地址 此字段用来定义端点 n 数据缓冲器的起始地址。在容量为 1024 字节的 EP_SRAM 中, 起始地址范围是 0x008 ~ 0x3FC, 且该字段值必须是 4 的倍数。



## 25 外设直接存储器访问 (PDMA)

### 简介

外设直接存储器访问电路 PDMA 提供了 12 个单向通道用于专用外设执行“外设 → 存储器”和“存储器 → 外设”的数据传输。应用程序同时也支持和要求如 FLASH→SRAM 或 SRAM→SRAM 类型的“存储器 → 存储器”数据传输。每个 PDMA 通道配置都是独立的。PDMA 通道传输分为多个块处理，且每个块的大小等于块的长度乘以数据的宽度。

### 特性

- 12 个单向 PDMA 通道
- 支持存储器 → 外设、外设 → 存储器和存储器 → 存储器类型的数据传输
- 8-bit、16-bit 和 32-bit 宽度数据传输
- 带有可配置通道优先级的软硬件需求的数据传输
- 线性、环形和非递增地址模式
- 4 个传输事件标志：传输完成、半传输、块结束和传输错误
- 自动重载功能

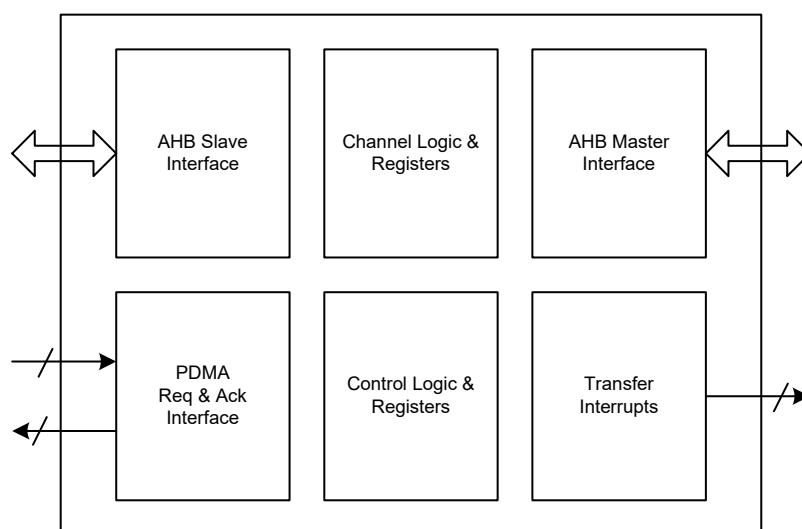


图 173. PDMA 方框图

## 功能描述

### AHB 主机

PDMA 是一个 AHB 主机，通过总线矩阵与其它 AHB 外设相连，如 FLASH 存储器、SRAM 存储器和 AHB-to-APB 总线桥。CPU 和 PDMA 通过总线矩阵可以同时访问不同的 AHB 从机。

### PDMA 通道

有 12 个单向 PDMA 通道用于外设和存储器之间的数据传输。每个 PDMA 通道的配置和操作都是独立的。对于一个双向数据传输应用，需要两个 PDMA 通道。每个 PDMA 通道可支持多个外设设备，但使用同一套寄存器。因此，一个 PDMA 通道在同一时间内只能服务一个外设。PDMA 通道的相关寄存器被限制只能由 32-bit 操作访问，否则，将发生系统硬故障。

### PDMA 请求映射

来自众多外设的请求，如 ADC、SPI、I<sup>2</sup>C 和 USART 等等，在进入 PDMA 之前先进行简单的逻辑与。这意味着一个 PDMA 通道一次只能使能一个请求。请参考图 174 – PDMA 请求映射架构。详细的外设 IP 请求映射列表如表 67 所示。外设 DMA 请求可通过相应外设寄存器中的 DMA 控制位独立激活或禁用。

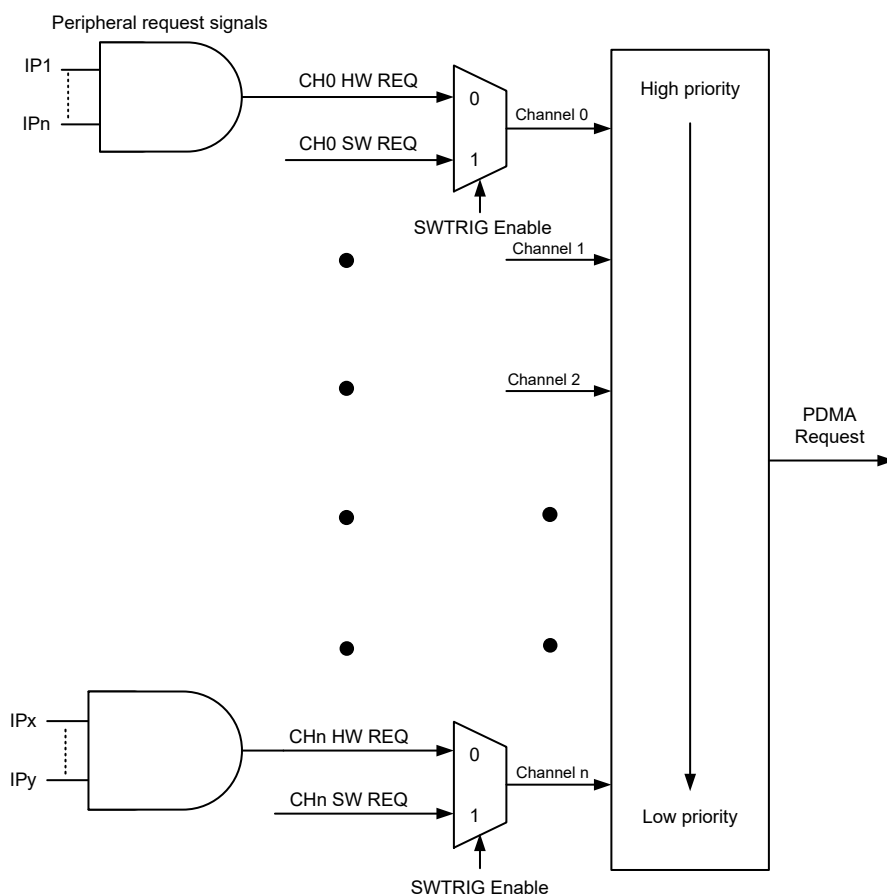


图 174. PDMA 请求映射架构

表 67. PDMA 通道配置

IP (x = 0, 1)	PDMA 通道编号											
	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10	CH11
ADC	ADC											
SPIx	SPI0_RX	SPI0_TX			SPI1_RX	SPI1_TX						
USARTx			USR0_RX	USR0_TX					USR1_RX	USR1_TX		
UARTx	UR0_RX	UR0_TX			UR1_RX	UR1_TX						
SCIx									SCI0_RX	SCI0_TX	SCI1_RX	SCI1_TX
I2Cx							I2C1_RX	I2C1_TX			I2C0_RX	I2C0_TX
MCTMx	MT0_CH0	MT0_TRIG MT1_CH0	MT0_CH1 MT1_CH2	MT0_CH3 MT1_UEV1	MT0_CH2 MT1_CH1	MT0_UEV1 MT1_CH3	MT1_UEV2	MT0_UEV2 MT1_TRIG				
GPTMx	GT0_CH1 GT0_CH3	GT0_UEV	GT0_CH2	GT0_CH0 GT0_TRIG					GT1_CH0	GT1_CH1 GT1_UEV	GT1_CH2 GT1_TRIG	GT1_CH3
I2S			I2S_RX	I2S_TX								
CSIF	CSIF											
SDIO							SDIO_RX	SDIO_TX				
AES											AES_OUT	AES_IN

通道传输

PDMA 通道传输被分成多个块处理，在每个块处理结束后，PDMA 仲裁发生。虽然这些通道传输都可以被激活，但一次只能通过总线进行一个块处理传输。通道传输顺序取决于每个 PDMA 通道的优先级设置。总的传输大小由块处理数和块大小计算而得。块大小等于块长度和数据位宽度的乘积。对于一个有效的传输，建议块长度为 4 的倍数。

总传输数据大小可由下列等式算出：

PDMA 通道总传输数据大小 = 块处理数 × ( 块长度 × 数据宽度 )

通道优先级

PDMA 提供四个优先级，即非常高、高、中、低，可由应用软件设置。PDMA 提供了两种方式来决定通道优先级。一种由应用软件配置选项进行设置，另一种由固定硬件编号决定。PDMA 仲裁处理器将首先检查请求 PDMA 提供数据传输服务的通道的软件配置优先级。如果多个通道有相同的优先级，则经过仲裁后，编号较小的通道比编号大的通道具有高优先级。

注意，当其它低优先级通道请求挂起时，最高优先级通道不会一直占用 PDMA 服务。在一个块处理完成后，最高优先级通道将被跳过一个块处理的时间。接着执行由第二优先级通道请求的块处理。由于第二优先级通道在当前块处理完成时会被排除，所以在第二优先级通道的当前块处理完成时，PDMA 仲裁处理器将重新检查除第二优先级通道外的其它请求通道。因此，较高优先级通道的块数据处理将被服务，且在当前块处理完成时，此通道将从优先级仲裁中排除。PDMA 将用上述的方法继续传输数据直到所有请求的通道数据传输完成。下图的例子显示了 PDMA 通道仲裁和安排。

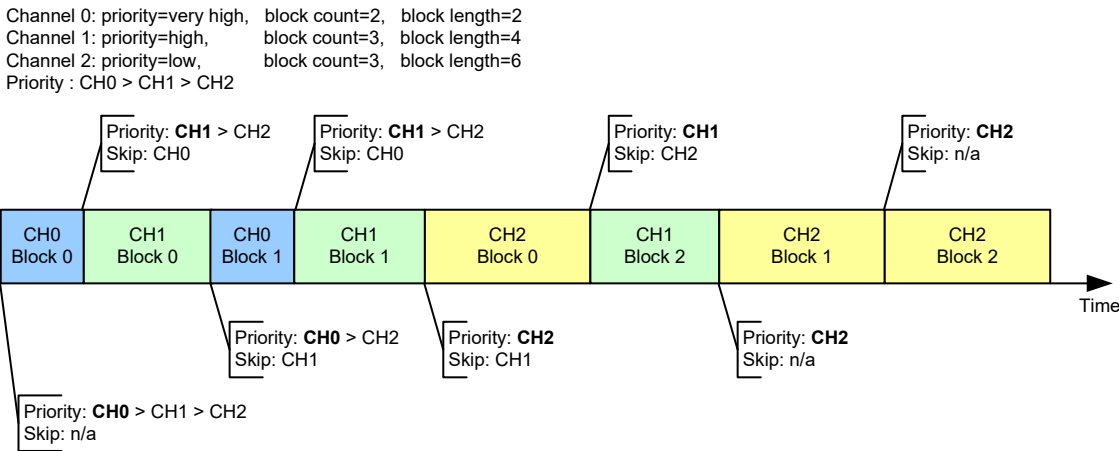


图 175. PDMA 通道仲裁和安排范例

### 传输请求

对于“外设 → 存储器”或“存储器 → 外设”传输，一个外设硬件请求将触发一个专用 PDMA 通道的块处理。但当软件请求发生时，相关专用 PDMA 通道的一个完整数据传输会被触发。应“存储器 → 存储器”数据复制应用的软件需求，建议 PDMA 通道被配置为较低的优先级和较小的块长度。

### 地址模式

PDMA 提供了三种地址模式，即线性地址、环形地址和固定地址模式。这些不同的地址模式用来支持不同种类的源地址和目标地址配置。下表显示了详细的地址模式组合。

表 68. PDMA 地址模式

源地址模式	目标地址模式
线性递增 / 递减地址	线性递增 / 递减地址
线性递增 / 递减地址	环形递增 / 递减地址
线性递增 / 递减地址	固定地址
环形递增 / 递减地址	线性递增 / 递减地址
环形递增 / 递减地址	环形递增 / 递减地址
固定地址	线性递增 / 递减地址
固定地址	固定地址

#### 线性地址模式

在数据被传输后，当前地址将以 1、2 或 4 为间隔递增或递减，取决于数据位的宽度设置。

#### 环形地址模式

在数据被传输后，当前地址将以 1、2 或 4 为间隔递增或递减，取决于数据位的宽度设置。当块处理完成，所设置的起始地址将加载到当前地址中。

#### 固定地址

数据传输后，当前地址仍然不变。

## 自动重载

当 PDMA 通道 n 控制寄存器 PDMACHnCR 中的自动重载控制位 AUTORLn 被置位时, 在当前 PDMA 通道数据传输完全完成时, 通道 n 当前地址和通道 n 当前传输大小将分别自动载入对应的初始值。通道 n 仍然有效且下一个相关的 PDMA 请求在不需要软件重新配置的情况下可以被服务。

## 传输中断

每个 PDMA 通道都有 5 种传输事件, 当传输事件发生时可以产生相应中断。这些传输事件分别为块处理结束 (BE)、半传输 (HT)、传输完成 (TC)、传输错误 (TE) 和总传输事件 (GE)。在 PDMA 中断使能寄存器 PDMAIER 中设置相关控制位可使能相关中断事件。如果 BE、HT、TC 或 TE 四个中断事件中有任意一个发生时, 将产生总中断事件 GE。清零 BE、HT、TC 或 TE 事件标志位也会清零 GE 标志位。清零 GE 标志位将自动清零所有其它事件标志位。当 PDMA 访问一个系统预留地址空间或 PDMA 接收一个请求但相应传输大小设置为 0 时, 将产生 TE 中断事件。

## 寄存器列表

下表显示了 PDMA 寄存器和复位值。

**表 69. PDMA 寄存器列表**

寄存器	偏移量	描述	复位值
<b>PDMA 通道 0 寄存器</b>			
PDMACH0CR	0x000	PDMA 通道 0 控制寄存器	0x0000_0000
PDMACH0SADR	0x004	PDMA 通道 0 源地址寄存器	0x0000_0000
PDMACH0DADR	0x008	PDMA 通道 0 目标地址寄存器	0x0000_0000
PDMACH0TSR	0x010	PDMA 通道 0 传输大小寄存器	0x0000_0000
PDMACH0CTSR	0x014	PDMA 通道 0 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 1 寄存器</b>			
PDMACH1CR	0x018	PDMA 通道 1 控制寄存器	0x0000_0000
PDMACH1SADR	0x01C	PDMA 通道 1 源地址寄存器	0x0000_0000
PDMACH1DADR	0x020	PDMA 通道 1 目标地址寄存器	0x0000_0000
PDMACH1TSR	0x028	PDMA 通道 1 传输大小寄存器	0x0000_0000
PDMACH1CTSR	0x02C	PDMA 通道 1 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 2 寄存器</b>			
PDMACH2CR	0x030	PDMA 通道 2 控制寄存器	0x0000_0000
PDMACH2SADR	0x034	PDMA 通道 2 源地址寄存器	0x0000_0000
PDMACH2DADR	0x038	PDMA 通道 2 目标地址寄存器	0x0000_0000
PDMACH2TSR	0x040	PDMA 通道 2 传输大小寄存器	0x0000_0000
PDMACH2CTSR	0x044	PDMA 通道 2 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 3 寄存器</b>			
PDMACH3CR	0x048	PDMA 通道 3 控制寄存器	0x0000_0000
PDMACH3SADR	0x04C	PDMA 通道 3 源地址寄存器	0x0000_0000
PDMACH3DADR	0x050	PDMA 通道 3 目标地址寄存器	0x0000_0000
PDMACH3TSR	0x058	PDMA 通道 3 传输大小寄存器	0x0000_0000
PDMACH3CTSR	0x05C	PDMA 通道 3 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 4 寄存器</b>			
PDMACH4CR	0x060	PDMA 通道 4 控制寄存器	0x0000_0000

寄存器	偏移量	描述	复位值
PDMACH4SADR	0x064	PDMA 通道 4 源地址寄存器	0x0000_0000
PDMACH4DADR	0x068	PDMA 通道 4 目标地址寄存器	0x0000_0000
PDMACH4TSR	0x070	PDMA 通道 4 传输大小寄存器	0x0000_0000
PDMACH4CTSR	0x074	PDMA 通道 4 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 5 寄存器</b>			
PDMACH5CR	0x078	PDMA 通道 5 控制寄存器	0x0000_0000
PDMACH5SADR	0x07C	PDMA 通道 5 源地址寄存器	0x0000_0000
PDMACH5DADR	0x080	PDMA 通道 5 目标地址寄存器	0x0000_0000
PDMACH5TSR	0x088	PDMA 通道 5 传输大小寄存器	0x0000_0000
PDMACH5CTSR	0x08C	PDMA 通道 5 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 6 寄存器</b>			
PDMACH6CR	0x090	PDMA 通道 6 控制寄存器	0x0000_0000
PDMACH6SADR	0x094	PDMA 通道 6 源地址寄存器	0x0000_0000
PDMACH6DADR	0x098	PDMA 通道 6 目标地址寄存器	0x0000_0000
PDMACH6TSR	0x0A0	PDMA 通道 6 传输大小寄存器	0x0000_0000
PDMACH6CTSR	0x0A4	PDMA 通道 6 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 7 寄存器</b>			
PDMACH7CR	0x0A8	PDMA 通道 7 控制寄存器	0x0000_0000
PDMACH7SADR	0x0AC	PDMA 通道 7 源地址寄存器	0x0000_0000
PDMACH7DADR	0x0B0	PDMA 通道 7 目标地址寄存器	0x0000_0000
PDMACH7TSR	0x0B8	PDMA 通道 7 传输大小寄存器	0x0000_0000
PDMACH7CTSR	0x0BC	PDMA 通道 7 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 8 寄存器</b>			
PDMACH8CR	0x0C0	PDMA 通道 8 控制寄存器	0x0000_0000
PDMACH8SADR	0x0C4	PDMA 通道 8 源地址寄存器	0x0000_0000
PDMACH8DADR	0x0C8	PDMA 通道 8 目标地址寄存器	0x0000_0000
PDMACH8TSR	0x0D0	PDMA 通道 8 传输大小寄存器	0x0000_0000
PDMACH8CTSR	0x0D4	PDMA 通道 8 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 9 寄存器</b>			
PDMACH9CR	0x0D8	PDMA 通道 9 控制寄存器	0x0000_0000
PDMACH9SADR	0x0DC	PDMA 通道 9 源地址寄存器	0x0000_0000
PDMACH9DADR	0x0E0	PDMA 通道 9 目标地址寄存器	0x0000_0000
PDMACH9TSR	0x0E8	PDMA 通道 9 传输大小寄存器	0x0000_0000
PDMACH9CTSR	0x0EC	PDMA 通道 9 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 10 寄存器</b>			
PDMACH10CR	0x0F0	PDMA 通道 10 控制寄存器	0x0000_0000
PDMACH10SADR	0x0F4	PDMA 通道 10 源地址寄存器	0x0000_0000
PDMACH10DADR	0x0F8	PDMA 通道 10 目标地址寄存器	0x0000_0000
PDMACH10TSR	0x100	PDMA 通道 10 传输大小寄存器	0x0000_0000
PDMACH10CTSR	0x104	PDMA 通道 10 当前传输大小寄存器	0x0000_0000
<b>PDMA 通道 11 寄存器</b>			
PDMACH11CR	0x108	PDMA 通道 11 控制寄存器	0x0000_0000
PDMACH11SADR	0x10C	PDMA 通道 11 源地址寄存器	0x0000_0000
PDMACH11DADR	0x110	PDMA 通道 11 目标地址寄存器	0x0000_0000
PDMACH11TSR	0x118	PDMA 通道 11 传输大小寄存器	0x0000_0000

寄存器	偏移量	描述	复位值
PDMACH11CTSR	0x11C	PDMA 通道 11 当前传输大小寄存器	0x0000_0000
<b>PDMA 总寄存器</b>			
PDMAISR0	0x120	PDMA 中断状态寄存器 0	0x0000_0000
PDMAISR1	0x124	PDMA 中断状态寄存器 1	0x0000_0000
PDMAISCR0	0x128	PDMA 中断状态清除寄存器 0	0x0000_0000
PDMAISCR1	0x12C	PDMA 中断状态清除寄存器 1	0x0000_0000
PDMAIER0	0x130	PDMA 中断使能寄存器 0	0x0000_0000
PDMAIER1	0x134	PDMA 中断使能寄存器 1	0x0000_0000

## 寄存器描述

### PDMA 通道 n 控制寄存器 – PDMACHnCR, n = 0 ~ 11

该寄存器定义了 PDMA 通道 n 数据传输配置。

偏移量: 0x000 (0), 0x018 (1), 0x030 (2), 0x048 (3), 0x060 (4), 0x078 (5), 0x090 (6), 0x0A8 (7), 0x0C0 (8),  
0x0D8 (9), 0x0F0 (10), 0x108 (11)  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	保留位				AUTORLn	FIXAENn	CHnPRI		
类型 / 复位					RW	0 RW	0 RW	0 RW	0
	7	6	5	4	3	2	1	0	
	SRCAMODn	SRCAINCn	DSTAMODn	DSTAINCn	DWIDTHn		SWTRIGn	CHnEN	
类型 / 复位	RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW	0

位	字段	描述
[11]	AUTORLn	通道 n 自动重载使能控制位 0: 除能自动重载功能 1: 使能自动重载功能 该位设为 1 将使能自动重载功能, 当传输完成, 通道 n 当前地址和当前传输大小将重新载入对应的起始值且 PDMA 通道 n 仍有效。如果该位设为 0, 在传输完成后, 通道 n 当前地址和当前传输大小的值将保持不变且 PDMA 通道 n 将除能。
[10]	FIXAENn	通道 n 固定地址使能控制位 0: 在环形地址模式下除能固定地址功能 1: 在环形地址模式下使能固定地址功能 注意, 该位仅当源地址或目标地址处于环形地址模式下时才有效。例如, 源地址模式设置为线性地址模式, 目标地址模式设置为环形模式。如果该位置位使能固定地址功能, 则源地址模式仍会处于线性地址模式, 但是目标地址将会处于固定地址模式而不是环形地址模式。



位	字段	描述
[9:8]	CHnPRI	<p>通道 n 优先级</p> <p>00: 低 01: 中 10: 高 11: 非常高</p> <p>CHnPRI 字段通过应用程序来配置通道的优先级。如果含有相同的软件配置优先等级的通道超过 1 个, 则更小通道编号的通道将在仲裁后优先传输一个数据块。</p>
[7]	SRCAMODn	<p>通道 n 源地址模式选择位</p> <p>0: 线性地址模式 1: 环形地址模式</p> <p>在线性地址模式下, 在一个完整的数据传输过程中当前源地址的值根据 SRCAINCn 位的值可以递增或递减。在环形地址模式下, 当前源地址的值在数据传输过程中也是根据 SRCAINCn 位的值来决定是递增或递减, 当一块的数据传输完成时 PDMACHnSADR 寄存器的低 16-bit 的值将会被加载为当前源地址的值。</p>
[6]	SRCAINCn	<p>通道 n 源地址增减控制位</p> <p>0: 递增 1: 递减</p> <p>该位用于定义在线性地址模式下的一个完整的数据传输过程中或在环形地址模式下的一个数据块传输过程中当前源地址是递增还是递减。</p>
[5]	DSTAMODn	<p>通道 n 目标地址模式选择位</p> <p>0: 线性地址模式 1: 环形地址模式</p> <p>在线性地址模式下, 当前目标地址的值在一个完整的数据传输过程中根据 DSTAINCn 位的值可以递增或递减。在环形地址模式下, 当前地址在一个数据块传输过程中也是根据 DSTAINCn 位的值来决定是递增或递减, 当一块的数据传输完成时所设置的起始地址将被加载到当前目标地址中。</p>
[4]	DSTAINCn	<p>通道 n 目标地址增减控制位</p> <p>0: 递增 1: 递减</p> <p>该位用于定义在线性地址模式下的一个完整的数据传输过程中或环形地址模式下一块数据传输过程中当前目标地址是递增还是递减。</p>
[3:2]	DWIDTHn	<p>数据位宽度选择位</p> <p>00: 8-bit 01: 16-bit 10: 32-bit 11: 保留</p> <p>该位用来选择 PDMA 通道 n 的数据位宽度。</p>
[1]	SWTRIGn	<p>软件触发控制位</p> <p>0: 无操作 1: 软件触发传输请求</p> <p>置位该位将会在 PDMA 通道 n 产生一个“存储器 → 存储器”软件传输请求。当传输完成该位会自动清零。</p>
[0]	CHnEN	<p>通道 n 使能控制位</p> <p>0: 除能 PDMA 通道 n 1: 使能 PDMA 通道 n</p> <p>置位该位将会在 PDMA 通道 n 上使能一个软件或硬件传输请求。当传输完成且自动重载功能除能时, 该位将自动由硬件清零。然而, 如果 AUTORLn 位设为 1, 使能自动重载功能, 在传输完成后该位将保持高以使能 PDMA 通道 n 功能用于下一个传输请求而不会由硬件自动清零。</p>



**PDMA 通道 n 源地址寄存器 – PDMACHnSADR, n = 0 ~ 11**

该寄存器定义了 PDMA 通道 n 的源地址。

偏移量: 0x004 (0), 0x01C (1), 0x034 (2), 0x04C (3), 0x064 (4), 0x07C (5), 0x094 (6), 0x0AC (7), 0x0C4 (8),  
0x0DC (9), 0x0F4 (10), 0x10C (11)  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	SADRn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	SADRn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	SADRn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	SADRn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	SADRn	通道 n 源地址 该寄存器用于定义 PDMA 通道 n 的 32-bit 源地址。

**PDMA 通道 n 目标地址寄存器 – PDMACHnDADR, n = 0 ~ 11**

该寄存器定义了 PDMA 通道 n 的目标地址。

偏移量: 0x008 (0), 0x020 (1), 0x038 (2), 0x050 (3), 0x068 (4), 0x080 (5), 0x098 (6), 0x0B0 (7), 0x0C8 (8),  
0x0E0 (9), 0x0F8 (10), 0x110 (11)  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	DADRn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	DADRn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	DADRn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	DADRn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	DADRn	通道 n 目标地址 该寄存器用于定义 PDMA 通道 n 的 32-bit 目标地址。

PDMA 通道 n 传输大小寄存器 – PDMACHnTSR, n = 0 ~ 11

该寄存器用于定义块处理的数量和长度。

偏移量: 0x010 (0), 0x028 (1), 0x040 (2), 0x058 (3), 0x070 (4), 0x088 (5), 0x0A0 (6), 0x0B8 (7), 0x0D0 (8),  
0x0E8 (9), 0x100 (10), 0x118 (11)  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	BLKCNTn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	BLKCNTn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	保留位								
类型 / 复位									
	7	6	5	4	3	2	1	0	
	BLKLENn								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:16]	BLKCNTn	通道 n 块处理数量 BLKCNTn 代表通道 n 完整传输的块处理的数量。一个完整传输的容量是由 BLKCNTn 和 BLKLENn 的值计算得到。BLKCNTn 最大值为 65535。
[7:0]	BLKLENn	通道 n 块长度 BLKLENn 代表数据块的长度。数据宽度由 PDMACHnCR 寄存器中的 DWIDTHn 字段来定义。BLKLENn 最大值为 255。

**PDMA 通道 n 当前传输大小寄存器 – PDMACHnCTSR, n = 0 ~ 11**

该寄存器用于定义当前块处理数量。

偏移量: 0x014 (0), 0x02C (1), 0x044 (2), 0x05C (3), 0x074 (4), 0x08C (5), 0x0A4 (6), 0x0BC (7), 0x0D4 (8),  
0x0EC (9), 0x104 (10), 0x11C (11)  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24								
	CBLKCNTn															
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	23	22	21	20	19	18	17	16								
	CBLKCNTn															
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	15	14	13	12	11	10	9	8								
	保留位															
类型 / 复位																
	7	6	5	4	3	2	1	0								
	保留位															
类型 / 复位																

位	字段	描述
[31:16]	CBLKCNTn	通道 n 当前块数量 CBLKCNTn 是一个 16-bit 的只读值，它定义了待传输的数据块的数量。一个数据块传输完成后，CBLKCNTn 值将会减 1。给 PDMACHnTSR 寄存器中的 BLKCNTn 写一个新值将会更新 CBLKCNTn 的值。

## PDMA 中断状态寄存器 0 – PDMAISR0

该寄存器用于定义 PDMA 通道 0 ~ 5 相关中断状态。

偏移量: 0x120

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位		TEISTA5	TCISTA5	HTISTA5	BEISTA5	GEISTA5	TEISTA4
类型 / 复位			RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	TCISTA4	HTISTA4	BEISTA4	GEISTA4	TEISTA3	TCISTA3	HTISTA3	BEISTA3
类型 / 复位	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	GEISTA3	TEISTA2	TCISTA2	HTISTA2	BEISTA2	GEISTA2	TEISTA1	TCISTA1
类型 / 复位	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	HTISTA1	BEISTA1	GEISTA1	TEISTA0	TCISTA0	HTISTA0	BEISTA0	GEISTA0
类型 / 复位	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

位	字段	描述
[29], [24], [19], [14], [9], [4]	TEISTAn	通道 n 传输错误中断状态位 (n = 0 ~ 5) 0: 传输错误未发生 1: 传输错误发生 该位由硬件置位, 写 1 到 PDMAISR0 寄存器中的相关中断状态清零位将会清零该位。 当 PDMA 访问系统预留地址空间或 PDMA 接收到一个请求但是对应的传输容量为 0 时, 将会发生传输错误。
[28], [23], [18], [13], [8], [3]	TCISTAn	通道 n 传输完成中断状态位 (n = 0 ~ 5) 0: 传输完成未发生 1: 传输完成发生 该位由硬件置位, 写 1 到 PDMAISR0 寄存器的相关中断状态清零位将会清零该位。 当 PDMA 完成数据传输任务时将发生传输完成事件。
[27], [22], [17], [12], [7], [2]	HTISTAn	通道 n 半传输中断状态位 (n = 0 ~ 5) 0: 半传输事件未发生 1: 半传输事件发生 该位由硬件置位, 写 1 到 PDMAISR0 寄存器的相关中断状态清零位将会清零该位。 当 PDMA 完成一半的数据传输任务时将发生半传输完成事件。
[26], [21], [16], [11], [6], [1]	BEISTAn	通道 n 块处理结束中断状态位 (n = 0 ~ 5) 0: 块处理结束事件未发生 1: 块处理结束事件发生 该位由硬件置位, 写 1 到 PDMAISR0 寄存器的相关中断状态清零位将会清零该位。 当 PDMA 完成数据块处理任务时将发生块处理结束事件。
[25], [20], [15], [10], [5], [0]	GEISTAn	通道 n 总传输中断状态位 (n = 0 ~ 5) 0: TE、TC、HT 或 BE 事件未发生 1: TE、TC、HT 或 BE 事件发生 该位由硬件置位, 写 1 到 PDMAISR0 寄存器的相关中断状态清零位 GEICLRn 将会清零该位。如果 BE、HT、TC 或 TE 中的任意一个事件发生将发生一个总传输事件。 清零 BE、HT、TC 或 TE 中的任意一个事件中断标志位会将 GE 中断标志位清零。注意, 如果写 1 到 PDMAISR0 寄存器中的 GEICLRn 位, GE 中断状态标志位和 BE、HT、TC 或 TE 事件中断标志位都将清零。

## PDMA 中断状态寄存器 1 – PDMAISR1

该寄存器用于定义 PDMA 通道 6 ~ 11 相关中断状态。

偏移量: 0x124

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位		TEISTA11	TCISTA11	HTISTA11	BEISTA11	GEISTA11	TEISTA10
类型 / 复位			RO	0	RO	0	RO	0
	23	22	21	20	19	18	17	16
	TCISTA10	HTISTA10	BEISTA10	GEISTA10	TEISTA9	TCISTA9	HTISTA9	BEISTA9
类型 / 复位	RO	0	RO	0	RO	0	RO	0
	15	14	13	12	11	10	9	8
	GEISTA9	TEISTA8	TCISTA8	HTISTA8	BEISTA8	GEISTA8	TEISTA7	TCISTA7
类型 / 复位	RO	0	RO	0	RO	0	RO	0
	7	6	5	4	3	2	1	0
	HTISTA7	BEISTA7	GEISTA7	TEISTA6	TCISTA6	HTISTA6	BEISTA6	GEISTA6
类型 / 复位	RO	0	RO	0	RO	0	RO	0

位	字段	描述
[29], [24], [19], [14], [9], [4]	TEISTAn	通道 n 传输错误中断状态位 (n = 6 ~ 11) 0: 传输错误未发生 1: 传输错误发生 该位由硬件置位, 写 1 到 PDMAISR1 寄存器中的相关中断状态清零位将会清零该位。 当 PDMA 访问系统预留地址空间或 PDMA 接收到一个请求但是对应的传输容量为 0 时, 将会发生传输错误。
[28], [23], [18], [13], [8], [3]	TCISTAn	通道 n 传输完成中断状态位 (n = 6 ~ 11) 0: 传输完成未发生 1: 传输完成发生 该位由硬件置位, 写 1 到 PDMAISR1 寄存器的相关中断状态清零位将会清零该位。 当 PDMA 完成数据传输任务时将发生传输完成事件。
[27], [22], [17], [12], [7], [2]	HTISTAn	通道 n 半传输中断状态位 (n = 6 ~ 11) 0: 半传输事件未发生 1: 半传输事件发生 该位由硬件置位, 写 1 到 PDMAISR1 寄存器的相关中断状态清零位将会清零该位。 当 PDMA 完成一半的数据传输任务时将发生半传输完成事件。
[26], [21], [16], [11], [6], [1]	BEISTAn	通道 n 块处理结束中断状态位 (n = 6 ~ 11) 0: 块处理结束事件未发生 1: 块处理结束事件发生 该位由硬件置位, 写 1 到 PDMAISR1 寄存器的相关中断状态清零位将会清零该位。 当 PDMA 完成数据块处理任务时将发生块处理结束事件。
[25], [20], [15], [10], [5], [0]	GEISTAn	通道 n 总传输中断状态位 (n = 6 ~ 11) 0: TE、TC、HT 或 BE 事件未发生 1: TE、TC、HT 或 BE 事件发生 该位由硬件置位, 写 1 到 PDMAISR1 寄存器的相关中断状态清零位 GEICLRn 将会清零该位。如果 BE、HT、TC 或 TE 中的任意一个事件发生将发生一个总传输事件。 清零 BE、HT、TC 或 TE 中的任意一个事件中断标志位会将 GE 中断标志位清零。注意, 如果写 1 到 PDMAISR1 寄存器中的 GEICLRn 位, GE 中断状态标志位和 BE、HT、TC 或 TE 事件中断标志位都将清零。

## PDMA 中断状态清除寄存器 0 – PDMAISR0

该寄存器用来将 PDMAISR0 寄存器中的相关中断状态位清零。

偏移量: 0x128

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位		TEICLR5	TCICLR5	HTICLR5	BEICLR5	GEICLR5	TEICLR4
类型 / 复位			WC	0	WC	0	WC	0
	23	22	21	20	19	18	17	16
	TCICLR4	HTICLR4	BEICLR4	GEICLR4	TEICLR3	TCICLR3	HTICLR3	BEICLR3
类型 / 复位	WC	0	WC	0	WC	0	WC	0
	15	14	13	12	11	10	9	8
	GEICLR3	TEICLR2	TCICLR2	HTICLR2	BEICLR2	GEICLR2	TEICLR1	TCICLR1
类型 / 复位	WC	0	WC	0	WC	0	WC	0
	7	6	5	4	3	2	1	0
	HTICLR1	BEICLR1	GEICLR1	TEICLR0	TCICLR0	HTICLR0	BEICLR0	GEICLR0
类型 / 复位	WC	0	WC	0	WC	0	WC	0

位	字段	描述
[29], [24], [19], [14], [9], [4]	TEICLRn	通道 n 传输错误中断状态清除 (n = 0 ~ 5) 0: 无操作 1: 将 PDMAISR0 寄存器中对应的 TEISTAn 位清零 写 1 到 TEICLRn 位会将 PDMAISR0 寄存器中的 TEISTAn 位清零。写 1 之后该位会自动清零。
[28], [23], [18], [13], [8], [3]	TCICLRn	通道 n 传输完成中断状态清除 (n = 0 ~ 5) 0: 无操作 1: 将 PDMAISR0 寄存器中对应的 TCISTAn 位清零 写 1 到 TCICLRn 位会将 PDMAISR0 寄存器中的 TCISTAn 位清零。写 1 之后该位会自动清零。
[27], [22], [17], [12], [7], [2]	HTRICLRn	通道 n 半传输中断状态清除 (n = 0 ~ 5) 0: 无操作 1: 将 PDMAISR0 寄存器中对应的 HTISTAn 位清零 写 1 到 HTRICLRn 位会将 PDMAISR0 寄存器中的 HTISTAn 位清零。写 1 之后该位会自动清零。
[26], [21], [16], [11], [6], [1]	BEICLRn	通道 n 块处理结束中断状态清除 (n = 0 ~ 5) 0: 无操作 1: 将 PDMAISR0 寄存器中对应的 BEISTAn 位清零 写 1 到 BEICLRn 位会将 PDMAISR0 寄存器中的 BEISTAn 位清零。写 1 之后该位会自动清零。
[25], [20], [15], [10], [5], [0]	GEICLRn	通道 n 总传输事件中断状态清除 (n = 0 ~ 5) 0: 无操作 1: 将 PDMAISR0 寄存器中对应的 TEISTAn、TCISTAn、HTISTAn、BEISTAn 和 GEISTAn 位清零 写 1 到 GEICLRn 位会将 PDMAISR0 寄存器中的 GEISTAn、TEISTAn、TCISTAn、HTISTAn 和 BEISTAn 位清零。写 1 之后该位会自动清零。

## PDMA 中断状态清除寄存器 1 – PDMAISR1

该寄存器用来将 PDMAISR1 寄存器中的相关中断状态位清零。

偏移量： 0x12C

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位		TEICLR11	TCICLR11	HTICLR11	BEICLR11	GEICLR11	TEICLR10	
类型 / 复位			WC	0	WC	0	WC	0	WC
	23	22	21	20	19	18	17	16	
	TCICLR10	HTICLR10	BEICLR10	GEICLR10	TEICLR9	TCICLR9	HTICLR9	BEICLR9	
类型 / 复位	WC	0	WC	0	WC	0	WC	0	WC
	15	14	13	12	11	10	9	8	
	GEICLR9	TEICLR8	TCICLR8	HTICLR8	BEICLR8	GEICLR8	TEICLR7	TCICLR7	
类型 / 复位	WC	0	WC	0	WC	0	WC	0	WC
	7	6	5	4	3	2	1	0	
	HTICLR7	BEICLR7	GEICLR7	TEICLR6	TCICLR6	HTICLR6	BEICLR6	GEICLR6	
类型 / 复位	WC	0	WC	0	WC	0	WC	0	WC

位	字段	描述
[29], [24], [19], [14], [9], [4]	TEICLRn	通道 n 传输错误中断状态清除 (n = 6 ~ 11) 0: 无操作 1: 将 PDMAISR1 寄存器中对应的 TEISTAn 位清零 写 1 到 TEICLRn 位会将 PDMAISR1 寄存器中的 TEISTAn 位清零。写 1 之后该位会自动清零。
[28], [23], [18], [13], [8], [3]	TCICLRn	通道 n 传输完成中断状态清除 (n = 6 ~ 11) 0: 无操作 1: 将 PDMAISR1 寄存器中对应的 TCISTAn 位清零 写 1 到 TCICLRn 位会将 PDMAISR1 寄存器中的 TCISTAn 位清零。写 1 之后该位会自动清零。
[27], [22], [17], [12], [7], [2]	HTRICLRn	通道 n 半传输中断状态清除 (n = 6 ~ 11) 0: 无操作 1: 将 PDMAISR1 寄存器中对应的 HTISTAn 位清零 写 1 到 HTRICLRn 位会将 PDMAISR1 寄存器中的 HTISTAn 位清零。写 1 之后该位会自动清零。
[26], [21], [16], [11], [6], [1]	BEICLRn	通道 n 块处理结束中断状态清除 (n = 6 ~ 11) 0: 无操作 1: 将 PDMAISR1 寄存器中对应的 BEISTAn 位清零 写 1 到 BEICLRn 位会将 PDMAISR1 寄存器中的 BEISTAn 位清零。写 1 之后该位会自动清零。
[25], [20], [15], [10], [5], [0]	GEICLRn	通道 n 总传输事件中断状态清除 (n = 6 ~ 11) 0: 无操作 1: 将 PDMAISR1 寄存器中对应的 TEISTAn、TCISTAn、HTISTAn、BEISTAn 和 GEISTAn 位清零 写 1 到 GEICLRn 位会将 PDMAISR1 寄存器中的 GEISTAn、TEISTAn、TCISTAn、HTISTAn 和 BEISTAn 位清零。写 1 之后该位会自动清零。



## PDMA 中断使能寄存器 0 – PDMAIER0

该寄存器用于使能或除能 PDMA 通道 0 ~ 5 相关中断。

偏移量: 0x130

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位		TEIE5	TCIE5	HTIE5	BEIE5	GEIE5	TEIE4
类型 / 复位			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	TCIE4	HTIE4	BEIE4	GEIE4	TEIE3	TCIE3	HTIE3	BEIE3
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	GEIE3	TEIE2	TCIE2	HTIE2	BEIE2	GEIE2	TEIE1	TCIE1
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	HTIE1	BEIE1	GEIE1	TEIE0	TCIE0	HTIE0	BEIE0	GEIE0
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

位	字段	描述
[29], [24], [19], [14], [9], [4]	TEIE <sub>n</sub>	通道 <b>n</b> 传输错误中断使能控制位 ( <b>n</b> = 0 ~ 5) 0: 传输错误中断除能 1: 传输错误中断使能 该位通过软件置位和清零。
[28], [23], [18], [13], [8], [3]	TCIE <sub>n</sub>	通道 <b>n</b> 传输完成中断使能控制位 ( <b>n</b> = 0 ~ 5) 0: 传输完成中断除能 1: 传输完成中断使能 该位通过软件置位和清零。
[27], [22], [17], [12], [7], [2]	HTIE <sub>n</sub>	通道 <b>n</b> 半传输中断使能控制位 ( <b>n</b> = 0 ~ 5) 0: 半传输中断除能 1: 半传输中断使能 该位通过软件置位和清零。
[26], [21], [16], [11], [6], [1]	BEIE <sub>n</sub>	通道 <b>n</b> 块处理结束中断使能控制位 ( <b>n</b> = 0 ~ 5) 0: 块处理结束中断除能 1: 块处理结束中断使能 该位通过软件置位和清零。
[25], [20], [15], [10], [5], [0]	GEIE <sub>n</sub>	通道 <b>n</b> 总传输事件中断使能控制位 ( <b>n</b> = 0 ~ 5) 0: 总传输事件中断除能 1: 总传输事件中断使能 该位通过软件置位和清零。

## PDMA 中断使能寄存器 1 – PDMAIER1

该寄存器用于使能或除能 PDMA 通道 6 ~ 11 相关中断。

偏移量： 0x134

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位		TEIE11	TCIE11	HTIE11	BEIE11	GEIE11	TEIE10
类型 / 复位			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	TCIE10	HTIE10	BEIE10	GEIE10	TEIE9	TCIE9	HTIE9	BEIE9
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	GEIE9	TEIE8	TCIE8	HTIE8	BEIE8	GEIE8	TEIE7	TCIE7
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	HTIE7	BEIE7	GEIE7	TEIE6	TCIE6	HTIE6	BEIE6	GEIE6
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

位	字段	描述
[29], [24], [19], [14], [9], [4]	TEIE <sub>n</sub>	通道 <b>n</b> 传输错误中断使能控制位 ( <b>n</b> = 6 ~ 11) 0: 传输错误中断除能 1: 传输错误中断使能 该位通过软件置位和清零。
[28], [23], [18], [13], [8], [3]	TCIE <sub>n</sub>	通道 <b>n</b> 传输完成中断使能控制位 ( <b>n</b> = 6 ~ 11) 0: 传输完成中断除能 1: 传输完成中断使能 该位通过软件置位和清零。
[27], [22], [17], [12], [7], [2]	HTIE <sub>n</sub>	通道 <b>n</b> 半传输中断使能控制位 ( <b>n</b> = 6 ~ 11) 0: 半传输中断除能 1: 半传输中断使能 该位通过软件置位和清零。
[26], [21], [16], [11], [6], [1]	BEIE <sub>n</sub>	通道 <b>n</b> 块处理结束中断使能控制位 ( <b>n</b> = 6 ~ 11) 0: 块处理结束中断除能 1: 块处理结束中断使能 该位通过软件置位和清零。
[25], [20], [15], [10], [5], [0]	GEIE <sub>n</sub>	通道 <b>n</b> 总传输事件中断使能控制位 ( <b>n</b> = 6 ~ 11) 0: 总传输事件中断除能 1: 总传输事件中断使能 该位通过软件置位和清零。

## 26 外部总线接口 (EBI)

### 简介

外部总线接口可用于访问外部并行接口设备，如 SRAM、Flash 和 LCD 模块。该接口存储映射于 Cortex®-M3 内部地址总线。数据和地址线可多路复用，减少了连接外部设备所需引脚数量。总线读 / 写时序可调整，以符合外部设备时序规格。注意，该接口只支持异步 8-bit 或 16-bit 总线接口。

### 特性

- 可编程接口，适用于多种存储器类型
  - 异步静态随机存取存储器 – SRAM
  - 只读存储器 – ROM
  - NOR Flash 存储器
  - 8-bit 或 16-bit 并行总线 CPU 接口设备
- 将 AHB 事务转换为相应的外部设备协议
- 4 个存储区且每个存储区有独立的片选控制
- 可编程时序，支持广泛的设备
  - 可编程等待状态或外部异步就绪信号的控制
  - 可编程总线周转周期
  - 可编程输出使能以及每个存储器 Bank 写使能周期延长
  - 每个存储器 Bank 都有单独的高有效或低有效接口控制信号设置
- 支持页读取模式
- 当 AHB 事务宽度和外部设备存储器宽度不同时，支持自动转换
- 写缓冲器可减少因 AHB 写突发事件而停滞的状况
- 支持复用和非复用的地址线和数据线配置
  - 多达 25 条地址线
  - 多达 16-bit 总线宽度

## 功能描述

EBI 模块方框图如图 176 所示。EBI 允许内部 CPU 和其它总线矩阵主机外设访问外部存储器或设备。EBI 可自动将内部 AHB 事务转换为外部设备协议。特别是当选择的外部存储器为 16-bit 或 8-bit 宽度时, AHB 的 32-bit 事务将自动分割成连续的 16-bit 或 8-bit 事务进行访问。

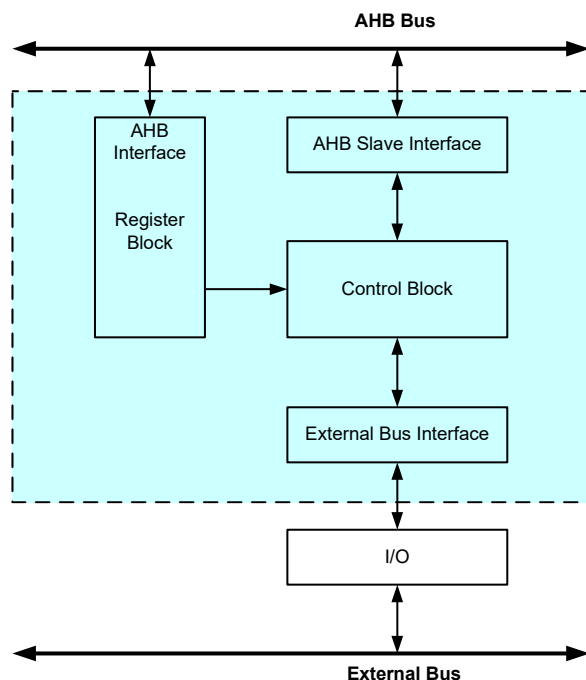


图 176. EBI 方框图

EBI 支持复用和非复用寻址模式。非复用寻址模式工作效率更高且速度更快, 但引脚需求量较大。复用寻址模式工作速度较慢, 需要一个外部地址锁存设备, 但引脚需求量较小。16 个 EBI\_AD 引脚的功能取决于使用何种复用寻址模式。在复用模式下, 这些引脚既可以用作地址线也可用作数据线。在非复用 8-bit 地址模式下, 这 16 个 EBI\_AD 引脚包含地址线和数据线。在复用寻址模式下, 如果需要更多地址位和数据位时, 借助一个外部锁存器仅使用 16 个 EBI\_AD 引脚便可实现多达 24-bit 的地址位或 16-bit 的数据位。此外, 无论在何种寻址模式下, 有多达 25 条非复用地址线在 EBI\_A 上可能能用于地址连接。所有支持的模式的具体操作将在接下来的部分介绍。AHB 时钟 (HCLK) 为 EBI 参考时钟。

非复用 8-bit 数据 8-bit 地址模式

此模式支持 8-bit 地址和 8-bit 数据。地址位于 EBI\_AD 线的高 8 位，数据位于低 8 位。可通过设置 EBICR 寄存器的 MODE 字段为 D8A8 来选择此模式。8-bit 模式读和写时序分别如图 177 和图 178 所示。

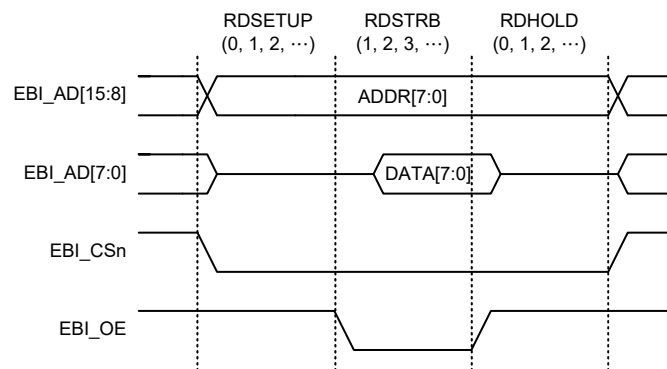


图 177. EBI 非复用 8-bit 数据 8-bit 地址读操作

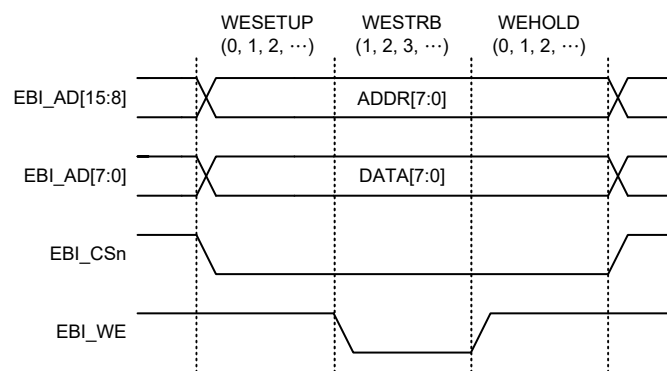


图 178. EBI 非复用 8-bit 数据 8-bit 地址写操作

非复用 16-bit 数据 N-bit 地址模式

此模式下, 16-bit 数据由 16 个 EBI\_AD 线提供, 地址由 EBI\_A 线提供。可通过设置 EBICR 寄存器的 MODE 字段为 D16 来选择此模式。当 EBI\_A 上的 N 个地址线使能时, 读和写信号分别如图 179 和图 180 所示。

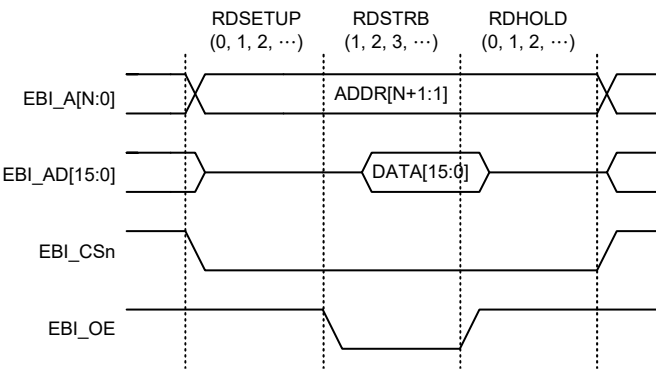


图 179. EBI 非复用 16-bit 数据 N-bit 地址读操作

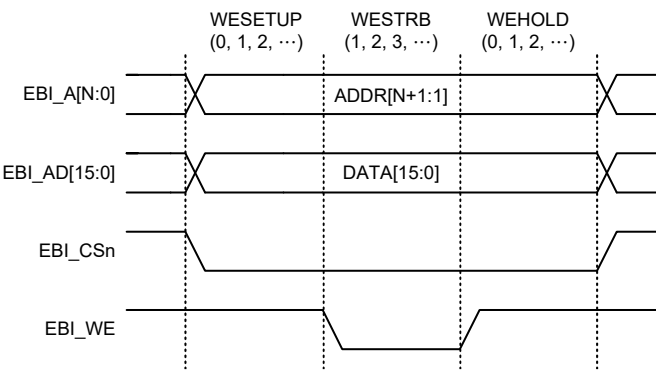


图 180. EBI 非复用 16-bit 数据 N-bit 地址写操作

由于内部 AHB 地址 (HADDR) 为字节地址 (8-bit), 而 16-bit 宽度的外部设备是按字寻址 (16-bit), 因此实际发给外部设备的地址会根据数据宽度的不同而不同, 如下表所示。

存储器宽度	发给 EBI 的数据地址
8-bit	HADDR[N:0] → EBI_A[N:0]
16-bit	HADDR[N+1:1] → EBI_A[N:0]

对于 16-bit 的外部设备宽度, EBI 在内部使用 HADDR[N+1:1] 产生地址 EBI\_A[N:0] 给外部设备。无论外部设备存储器宽度为 16-bit 还是 8-bit, EBI\_A[0] 必须连接到外部设备地址的 A[0]。

复用 16-bit 数据 16-bit 地址模式

此模式支持 16-bit 地址和 16-bit 数据,但需要使用一个外部锁存器以及一个额外的 EBI\_ALE 信号。16-bit 地址和 16-bit 数据通过 EBI\_AD 引脚复用。EBI 地址锁存器设置方框图如图 181 所示。可通过设置 EBICR 寄存器的 MODE 字段为 D16A16ALE 来选择此模式。

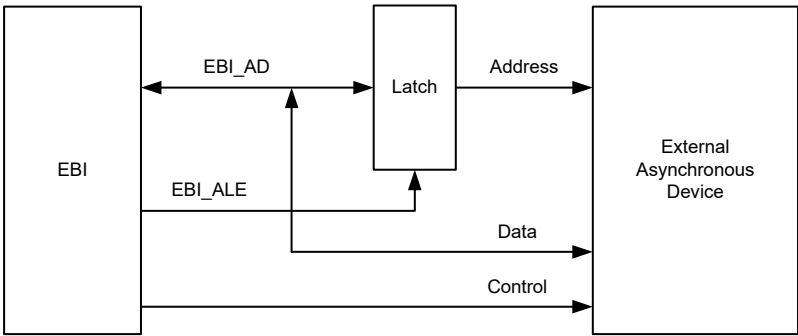


图 181. EBI 地址锁存器设置方框图

事务开始时地址被输出至 EBI\_AD 线。外部地址锁存器由 EBI\_ALE 信号控制并存储地址。接着根据所需操作数据可被读取或写入。读和写信号分别如图 182 和图 183 所示。

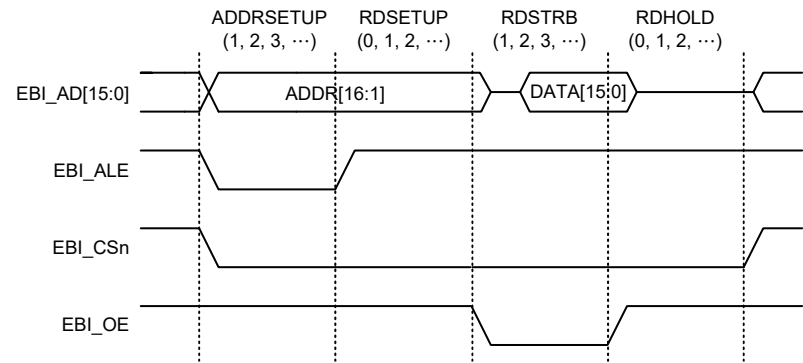


图 182. EBI 复用 16-bit 数据 16-bit 地址读操作

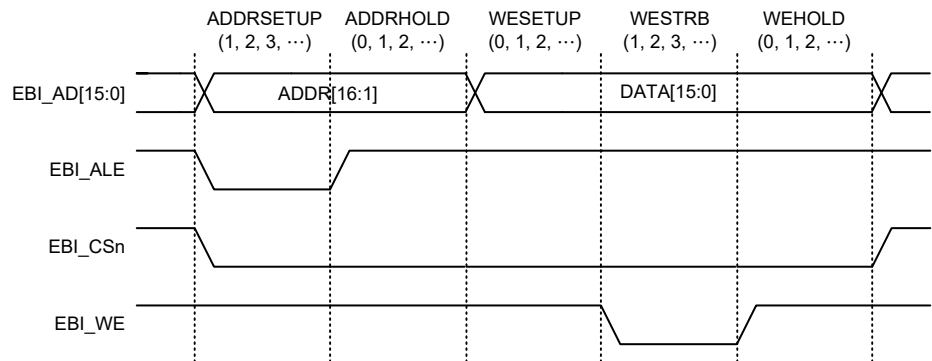


图 183. EBI 复用 16-bit 数据 16-bit 地址写操作

### 复用 8-bit 数据 24-bit 地址模式

此模式允许 24-bit 地址和 8-bit 数据通过 EBI\_AD[15:0] 线复用以减少引脚使用量, 并使用 EBI\_ALE 信号解码 8-bit 数据和 24-bit 地址。EBI\_AD 线的高 8 位, 即 EBI\_AD[15:8], 依次用作地址的最高 8 位和最低 8 位。EBI\_AD 线的低 8 位, 即 EBI\_AD[7:0], 用于地址的中间 8 位和 8 位的数据。可通过设置 EBICR 寄存器的 MODE 字段为 D8A24ALE 来选择此模式。读和写信号分别如图 184 和图 185 所示。

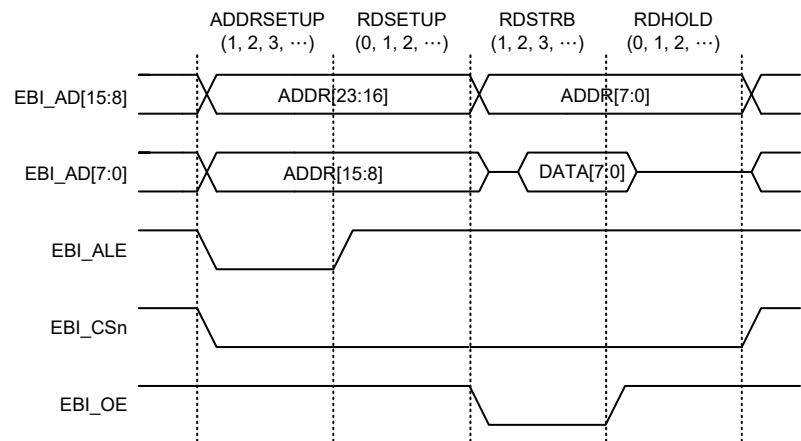


图 184. EBI 复用 8-bit 数据 24-bit 地址读操作



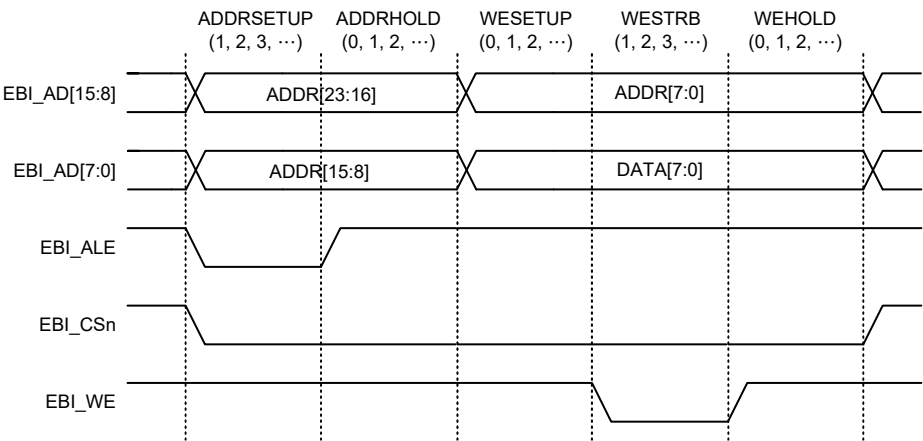


图 185. EBI 复用 8-bit 数据 24-bit 地址写操作

页读取操作

页面模式读取操作相对于传统的异步读取操作性能再次提高。在支持页面模式的设备中，首先进行一个初始的异步读取访问，接着，仅通过改变低位地址便可快速读取相邻的地址。例如，对于支持页模式具有 16 个页地址成员的设备，地址 A[3:0] 将被使用。若改变地址 A[4] 或更高位都将停止页读取操作并开始新一轮异步读取访问。页读取模式相比随机地址模式，对相邻地址的读取速度更快。

通过设置 EBIRTRn 寄存器的 PGEN 位为 1 可使能页面模式操作。若该模式已使能，EBIPCR 寄存器的 RDPG 字段用于定义页内访问的时间，EBIPCR 寄存器的 PAGELEN 字段用于定义一页的地址成员数量。EBIPCR 寄存器的 INCHIT 字段用于定义页面击中是发生在任何一个地址成员还是发生在递增的地址上。宽范围的 AHB 读取被自动转换为多个窄的外部设备读取，由此引发的连续读取可触发页面读取模式。下图显示了适用于所有寻址模式的典型的页面读取时序。

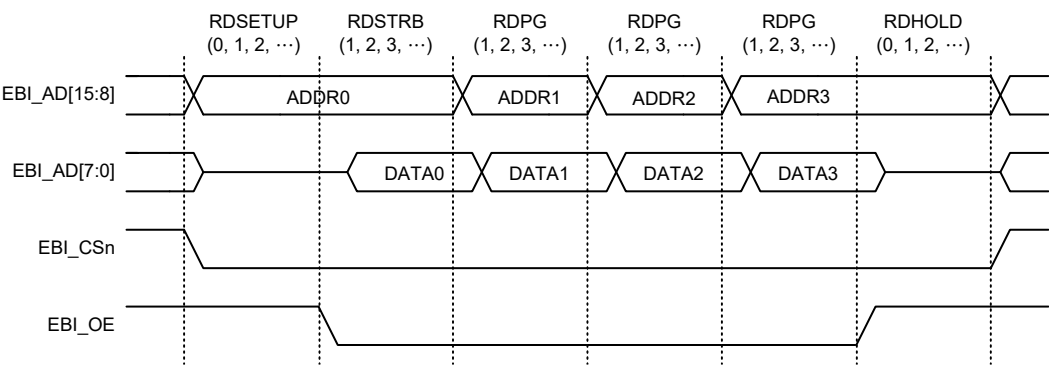


图 186. EBI 非复用 8-bit 数据 8-bit 地址模式页读取操作

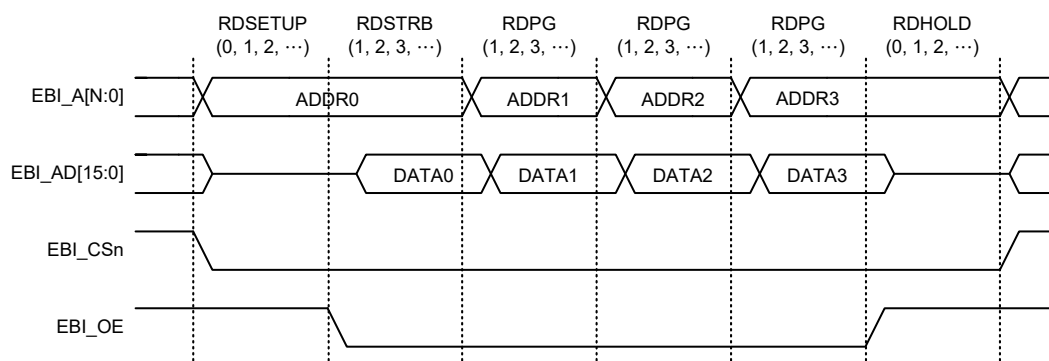


图 187. EBI 非复用 16-bit 数据 N-bit 地址模式页读取操作

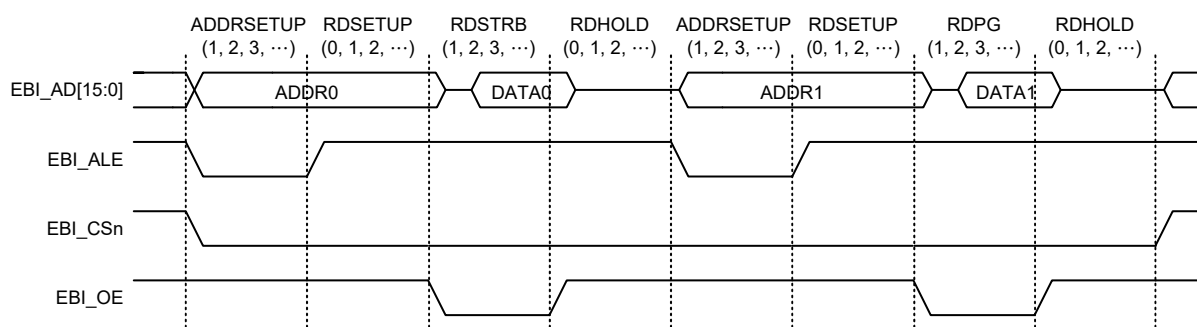


图 188. EBI 复用 16-bit 数据 16-bit 地址模式页读取操作

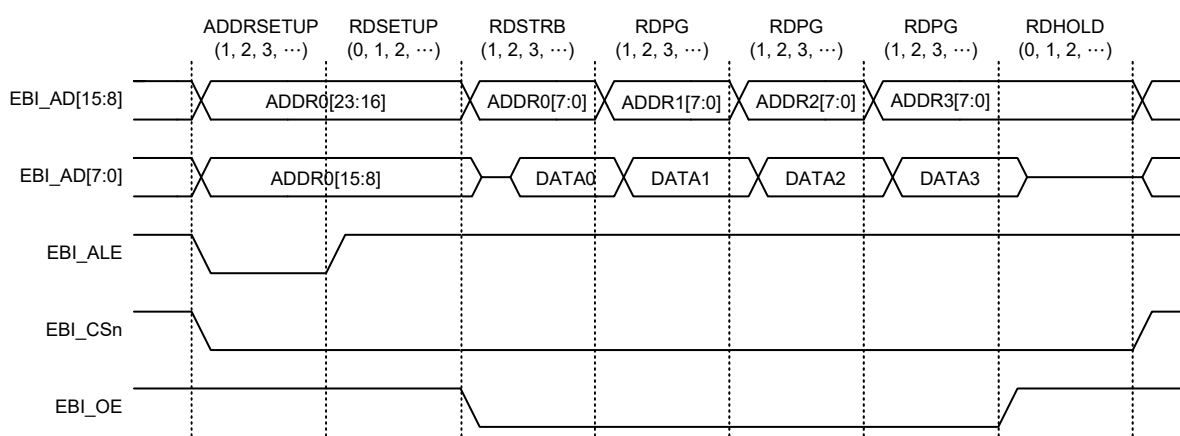
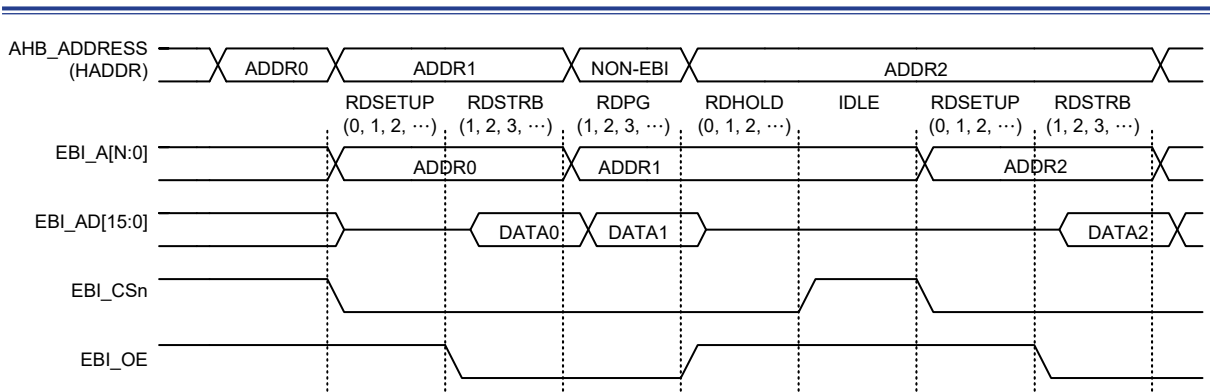


图 189. EBI 复用 8-bit 数据 24-bit 地址模式页读取操作

EBIPCR 寄存器的 PAGEOPEN 字段定义了页读取保持使能的最长时间。若 PAGEOPEN 定义的时间还未超出, 都可以通过 RDPG 时序启动一个新的对 OPEN 页的读取事务。页读取事务在以下情况发生时允许关闭页面模式:

- 在连续的读取事务过程中超过了 PAGEOPEN 定义的时间
- EBI 事务插入一个写操作或一个非页内读取操作
- 无新的 EBI 事务

在图 190 中, 由于插入了一个 AHB 传输的非相关地址导致 ADDR2 未及时送到, 因此只有 ADDR1 为页内时序。此时页面模式关闭, ADDR2 为非页内时序, 被视为一个正常读取访问。



Note1: HADDR is AHB bus address input.

图 190. EBI 页面关闭范例

### 写缓冲器和 EBI 状态

此 EBI 有一个 32-bit 宽的写缓冲器。此缓冲器可用于防止 CPU 或 PDMA 访问一个潜在低速的外部设备的 AHB 写突发事务造成的停滞。

EBISR 寄存器的 EBIBUSY 状态位用于指示 EBI 的 AHB 事务是否仍保持有效。当执行一个 AHB 读或写操作时, EBIBUSY 位保持为 1, 直到与外部设备进行的此事务已结束。

### 总线周转和空闲周期

EBI\_AD 线可由 EBI 或外部设备驱动, 这取决于 EBI 总线的周期状态。RDHOLD 时序参数用于定义总线周转时间, 设置此参数时应根据外部设备的特性以确保有足够的时间。EBI 默认设置是在访问同一个 Bank 的各个 EBI 事务之间插入一个 IDLE 周期。图 191 显示了在两个连续的读事务之间插入一个 IDLE 周期。在 IDLE 状态也可提供所需总线周转时间的情况下, RDHOLD 参数可设置为 0。为了增强 EBI 访问性能, 可通过设置 EBICR 寄存器的 NOIDLEn 位为 1 来除能 IDLE 状态自动插入功能。图 192 显示了非复用地址模式下两个连续的读操作范例。

以下情况时会自动插入一个 IDLE 周期:

- 当 NOIDLEn 位为 0 时, 访问同一个 Bank 的两个外部设备事务之间。
- 访问不同 Bank 的两个外部设备事务之间。
- EBI\_AD 线上一个读操作和一个紧接着的写操作之间。
- EBI 无外部事务请求时。

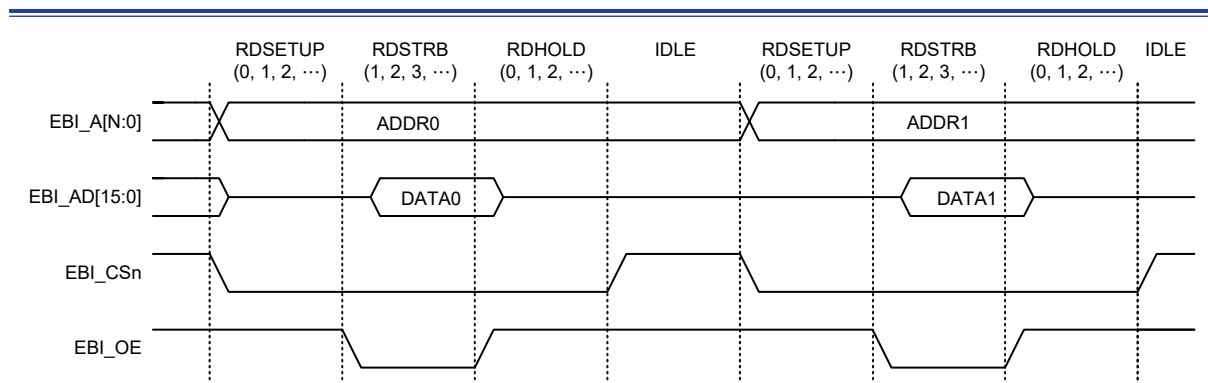


图 191. EBI 在同一个 Bank 的各个事务之间插入一个 IDLE 周期 (NOIDLE = 0)

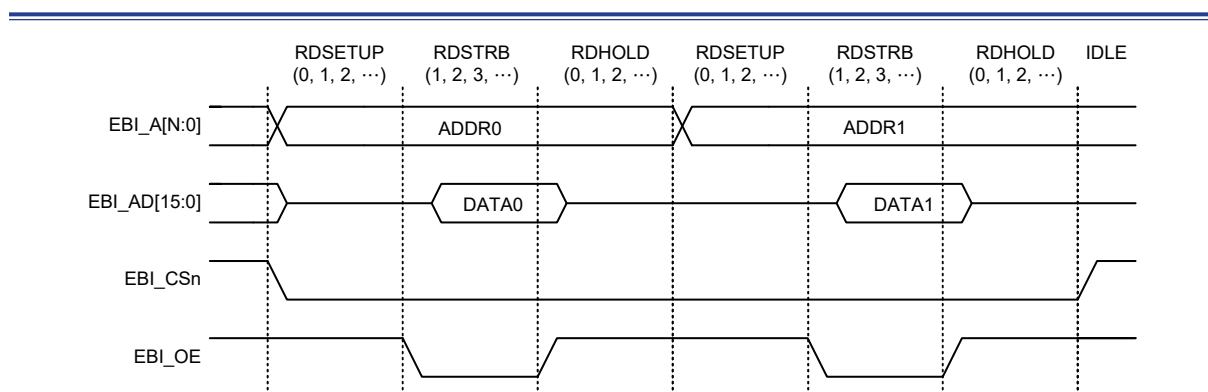


图 192. EBI 在同一个 Bank 的各个事务之间不插入 IDLE 周期 (NOIDLE = 1)

### AHB 事务宽度转换

AHB 事务与外部设备的映射取决于外部设备的数据宽度以及是否支持外部设备字节通道。表 70 显示了 AHB 事务与外部设备的 EBI 映射。EBI 可自动将不同的 AHB 事务宽度转换为匹配外部设备总线容量的外部设备事务。

- 若 AHB 主机 (CPU 或 PDMA) 事务宽度大于外部总线事务宽度, EBI 会将 AHB 事务分割且转换为连续多个外部事务, 这些外部事务有连续递增的地址并且从 AHB 事务的最低位开始。

- 若 AHB 主机 (CPU 或 PDMA) 事务宽度小于外部总线事务宽度, EBI 的行为取决于字节通道是否可用。当字节通道可用时, 使用字节通道选择要读取的数据; 当字节通道不可用时, 将根据外部设备的数据总线宽度读取, 忽略多余的数据。当字节通道可用时, 使用字节通道选择要写入的数据; 当字节通道不可用时, EBI 自动执行读 - 修改 - 写时序。

**表 70. AHB 事务宽度与外部设备事务的 EBI 映射**

AHB 事务	8-bit 外部设备事务	16-bit 外部设备事务 ( 字节通道可用 )	16-bit 外部设备事务 ( 字节通道不可用 )
8-bit 读操作	1 × 8-bit 读操作	1 × 8-bit 读操作 ( 使用字节通道 )	1 × 16-bit 读操作 ( EBI 忽略多余的数据 )
16-bit 读操作	2 × 8-bit 读操作	1 × 16-bit 读操作	1 × 16-bit 读操作
32-bit 读操作	4 × 8-bit 读操作	2 × 16-bit 读操作	2 × 16-bit 读操作
8-bit 写操作	1 × 8-bit 写操作	1 × 8-bit 写操作 ( 使用字节通道 )	1 × 16-bit 读操作 ; 1 × 16-bit 写操作 ( EBI 执行读 - 修改 - 写 时序 )
16-bit 写操作	2 × 8-bit 写操作	1 × 16-bit 写操作	1 × 16-bit 写操作
32-bit 写操作	4 × 8-bit 写操作	2 × 16-bit 写操作	2 × 16-bit 写操作

**表 71. AHB 事务宽度与外部设备事务宽度的 EBI 映射 ( 使用字节通道 EBI\_BL[1:0] )**

外部总线 宽度	来自 AHB 主机的访问		访问外部总线接口 (EBI)			
	访问类型	地址 HADDR[1:0] <sup>(1)</sup>	访问 分割	EBI_A[1:0] 输出值	EBI_AD[15:0] 有效数据	EBI_BL[1:0] 输出值
8-bit	字节 (8-bit)	0b00	无分割	0b00	EBI_AD[7:0]	0b10
		0b01	无分割	0b01		
		0b10	无分割	0b10		
		0b11	无分割	0b11		
	半字 (16-bit)	0b00	1/2 访问	0b00		
			2/2 访问	0b01		
	半字 (16-bit)	0b10	1/2 访问	0b10		
			2/2 访问	0b11		
	字 (32-bit)	0b00	1/4 访问	0b00		
			2/4 访问	0b01		
			3/4 访问	0b10		
			4/4 访问	0b11		
16-bit	字节 (8-bit)	0b00	无分割	0bx0	EBI_AD[7:0]	0b10
		0b01	无分割	0bx0	EBI_AD[15:8]	0b01
		0b10	无分割	0bx1	EBI_AD[7:0]	0b10
		0b11	无分割	0bx1	EBI_AD[15:8]	0b01
	半字 (16-bit)	0b00	无分割	0bx0	EBI_AD[15:0]	0b00
		0b10	无分割	0bx1	EBI_AD[15:0]	0b00
	字 (32-bit)	0b00	1/2 访问	0bx0	EBI_AD[15:0]	0b00
			2/2 访问	0bx1	EBI_AD[15:0]	0b00

注: 1. HADDR 为 AHB 总线地址输入。

2. 在该表格中, 字节通道极性为低有效。

EBI Bank 访问

EBI 被分为 4 个不同的地址区域 (Bank), 每一个区域有各自单独的 EBI\_CS<sub>n</sub> 线。当访问其中一个区域时, 对应的 EBI\_CS<sub>n</sub> 线置位。这样就有多达 4 个独立的设备可共用 EBI 线且分别通过 EBI\_CS<sub>n</sub> 线识别。每个 Bank 可通过 EBICR 寄存器单独使能或除能。且每个 Bank 可单独定义外部设备的行为, 包括数据宽度、时序定义、页面模式操作和引脚极性。每个 Bank 可访问的数据空间高达 64 MB, 如图 193 所示。EBI 区域的地址从存储器映射图的 0x60000000 开始, 还可用于执行代码。当通过 EBI 区域的这个地址开始执行代码时, CPU 通过系统总线接口抓取指令。这将导致性能降低, 因为 CPU 访问堆栈、SRAM 和外设时也需要使用系统总线接口。

为了提高通过 EBI 运行代码的效率, EBI 的 Bank 0 也映射于从 0x1B000000 起的代码空间。当通过这个代码空间运行代码时, Cortex®-M3 通过 I/D-Code 总线接口抓取指令, 则系统总线接口可用于数据访问。通过 I/D-Code 总线抓取指令可提高性能。

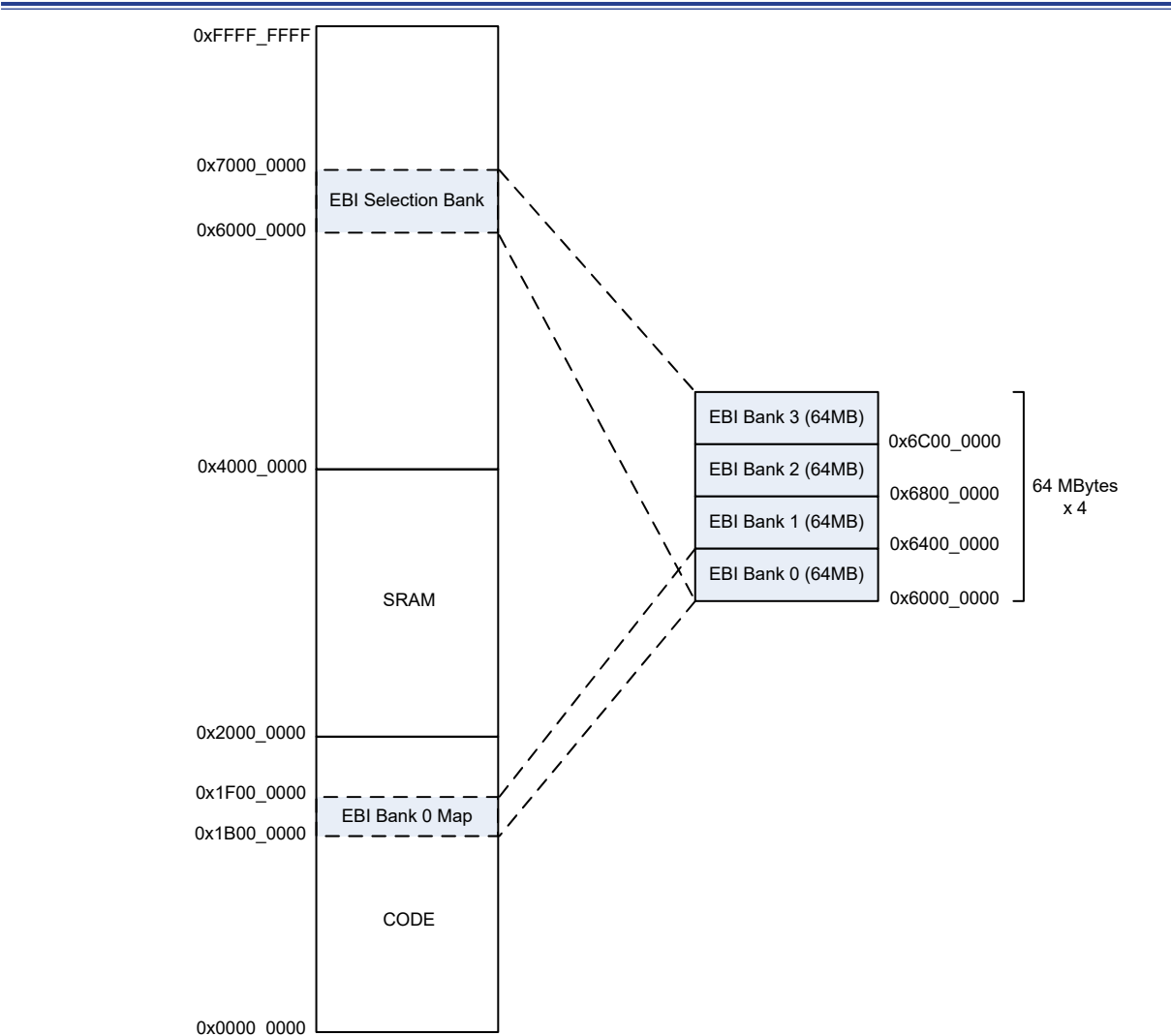


图 193. EBI Bank 存储器映射

## EBI 就绪

一些外部设备可通过插入一个等待信号来指示设备还没结束写或读操作。当外部设备已通过 EBICR 寄存器里对应的 ARDYEN 位使能后，EBI 的 EBI\_ARDY 输入信号可为低速外部设备延长读或写周期。EBI\_ARDY 信号的极性通过 EBIPR 寄存器中的 ARDYPOL 位进行配置。若 ARDYPOL 位设置为低有效，则在 EBI\_ARDY 引脚保持为高期间读或写周期被延长。当外部设备未撤销 EBI\_ARDY 信号时，还提供一个超时检查功能预防系统锁存的情况。若在超时计数周期内未撤销 EBI\_ARDY 信号，将产生一个总线异步就绪超时中断。此超时周期的默认值为 32 个 HLCK 时钟周期。异步就绪超时功能可通过 EBICR 寄存器里对应的 ARDYTDIS 位置位来除能。注意，每个存储器 Bank 可单独设置各自的等待行为。

## PDMA 请求

EBI 只支持软件触发 PDMA 服务。

## 寄存器列表

下表显示了 EBI 寄存器和复位值。

表 72. EBI 寄存器列表

寄存器	偏移量	描述	复位值
EBICR	0x000	EBI 控制寄存器	0x0000_0000
EBIPCR	0x004	EBI 页控制寄存器	0x0000_0F00
EBISR	0x008	EBI 状态寄存器	0x0000_0010
EBIATR0	0x010	EBI 地址时序寄存器 0	0x0000_0F0F
EBIRTR0	0x014	EBI 读时序寄存器 0	0x000F_3F0F
EBIWTR0	0x018	EBI 写时序寄存器 0	0x000F_3F0F
EBIPR0	0x01C	EBI 极性寄存器 0	0x0000_0000
EBIATR1	0x020	EBI 地址时序寄存器 1	0x0000_0F0F
EBIRTR1	0x024	EBI 读时序寄存器 1	0x000F_3F0F
EBIWTR1	0x028	EBI 写时序寄存器 1	0x000F_3F0F
EBIPR1	0x02C	EBI 极性寄存器 1	0x0000_0000
EBIATR2	0x030	EBI 地址时序寄存器 2	0x0000_0F0F
EBIRTR2	0x034	EBI 读时序寄存器 2	0x000F_3F0F
EBIWTR2	0x038	EBI 写时序寄存器 2	0x000F_3F0F
EBIPR2	0x03C	EBI 极性寄存器 2	0x0000_0000
EBIATR3	0x040	EBI 地址时序寄存器 3	0x0000_0F0F
EBIRTR3	0x044	EBI 读时序寄存器 3	0x000F_3F0F
EBIWTR3	0x048	EBI 写时序寄存器 3	0x000F_3F0F
EBIPR3	0x04C	EBI 极性寄存器 3	0x0000_0000
EBIENR	0x050	EBI 中断使能寄存器	0x0000_0000
EBIIFR	0x054	EBI 中断标志寄存器	0x0000_0000
EBIIFCR	0x058	EBI 中断清除寄存器	0x0000_0000

## 寄存器描述

### EBI 控制寄存器 – EBICR

该寄存器定义了 EBI bank 的控制设置。

偏移量： 0x000

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	IDLET				BLN3	BLN2	BLN1	BLN0
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	ARDYTDIS3	ARDYEN3	ARDYTDIS2	ARDYEN2	ARDYTDIS1	ARDYEN1	ARDYTDIS0	ARDYEN0
类型 / 复位								
	15	14	13	12	11	10	9	8
	NOIDLE3	NOIDLE2	NOIDLE1	NOIDLE0	BANKEN3	BANKEN2	BANKEN1	BANKEN0
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	MODE3		MODE2		MODE1		MODE0	
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[31:28]	IDLET	IDLE 时间 该位设置了 EBI 事务之间的周期数。若该位为 0，硬件插入一个周期，该周期以 HCLK 时钟周期为单位。
[27]	BLN3	字节通道使能 3 0：除能 EBI 字节通道功能 1：使能 EBI 字节通道功能 使能或除能 Bank 3 的字节通道功能。
[26]	BLN2	字节通道使能 2 0：除能 EBI 字节通道功能 1：使能 EBI 字节通道功能 使能或除能 Bank 2 的字节通道功能。
[25]	BLN1	字节通道使能 1 0：除能 EBI 字节通道功能 1：使能 EBI 字节通道功能 使能或除能 Bank 1 的字节通道功能。
[24]	BLN0	字节通道使能 0 0：除能 EBI 字节通道功能 1：使能 EBI 字节通道功能 使能或除能 Bank 0 的字节通道功能。
[23]	ARDYTDIS3	异步就绪超时除能 3 0：使能 EBI 异步就绪超时控制功能 1：除能 EBI 异步就绪超时控制功能 使能或除能 Bank 3 异步就绪超时功能。默认的异步就绪超时周期为 32 个 HCLK 时钟周期且不可改变。
[22]	ARDYEN3	异步就绪使能 3 0：除能 EBI 异步就绪控制功能 1：使能 EBI 异步就绪控制功能 使能或除能 Bank 3 异步就绪功能。



位	字段	描述
[21]	ARDYTDIS2	异步就绪超时除能 2 0: 使能 EBI 异步就绪超时控制功能 1: 除能 EBI 异步就绪超时控制功能 使能或除能 Bank 2 异步就绪超时功能。默认的异步就绪超时周期为 32 个 HCLK 时钟周期且不可改变。
[20]	ARDYEN2	异步就绪使能 2 0: 除能 EBI 异步就绪控制功能 1: 使能 EBI 异步就绪控制功能 使能或除能 Bank 2 异步就绪功能。
[19]	ARDYTDIS1	异步就绪超时除能 1 0: 使能 EBI 异步就绪超时控制功能 1: 除能 EBI 异步就绪超时控制功能 使能或除能 Bank 1 异步就绪超时功能。默认的异步就绪超时周期为 32 个 HCLK 时钟周期且不可改变。
[18]	ARDYEN1	异步就绪使能 1 0: 除能 EBI 异步就绪控制功能 1: 使能 EBI 异步就绪控制功能 使能或除能 Bank 1 异步就绪功能。
[17]	ARDYTDIS0	异步就绪超时除能 0 0: 使能 EBI 异步就绪超时控制功能 1: 除能 EBI 异步就绪超时控制功能 使能或除能 Bank 0 异步就绪超时功能。默认的异步就绪超时周期为 32 个 HCLK 时钟周期且不可改变。
[16]	ARDYEN0	异步就绪使能 0 0: 除能 EBI 异步就绪控制功能 1: 使能 EBI 异步就绪控制功能 使能或除能 Bank 0 异步就绪功能。
[15]	NOIDLE3	无 IDLE 3 0: 使能 IDLE 状态插入 1: 除能 IDLE 状态插入 使能或除能 Bank 3 各事务之间空闲状态插入功能。
[14]	NOIDLE2	无 IDLE 2 0: 使能 IDLE 状态插入 1: 除能 IDLE 状态插入 使能或除能 Bank 2 各事务之间空闲状态插入功能。
[13]	NOIDLE1	无 IDLE 1 0: 使能 IDLE 状态插入 1: 除能 IDLE 状态插入 使能或除能 Bank 1 各事务之间空闲状态插入功能。
[12]	NOIDLE0	无 IDLE 0 0: 使能 IDLE 状态插入 1: 除能 IDLE 状态插入 使能或除能 Bank 0 各事务之间空闲状态插入功能。
[11]	BANKEN3	Bank 3 使能 0: 除能 1: 使能 该位使能或除能 Bank 3。
[10]	BANKEN2	Bank 2 使能 0: 除能 1: 使能 该位使能或除能 Bank 2。

位	字段	描述
[9]	BANKEN1	Bank 1 使能 0: 除能 1: 使能 该位使能或除能 Bank 1。
[8]	BANKEN0	Bank 0 使能 0: 除能 1: 使能 该位使能或除能 Bank 0。
[7:6]	MODE3	设置 EBI Bank 3 访问模式 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[5:4]	MODE2	设置 EBI Bank 2 访问模式 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[3:2]	MODE1	设置 EBI Bank 1 访问模式 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[1:0]	MODE0	设置 EBI Bank 0 访问模式 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16

EBI 页控制寄存器 – EBIPCR

该寄存器定义了 EBI 页读取配置。

偏移量： 0x004  
复位值： 0x0000\_0F00

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	PAGEOPEN							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	保留位				RDPG			
类型 / 复位					RW	1	RW	1
	7	6	5	4	3	2	1	0
	保留位			INCHIT	保留位		PAGELEN	
类型 / 复位				RW	0			RW
						0		0

位	字段	描述
[23:16]	PAGEOPEN	页面打开最长时间 设置页面打开的最大保持时间。以一个 HCLK 时钟周期为单位。注意，若页读取模式已使能，PAGEOPEN 字段不可设置为 0。
[11:8]	RDPG	页读取访问时间 0000: 1 个 HCLK 时钟周期 0001: 1 个 HCLK 时钟周期 0010: 2 个 HCLK 时钟周期 ..... 1111: 15 个 HCLK 时钟周期 设置一个页读取访问时间的周期数，以一个 HCLK 时钟周期为单位。
[4]	INCHIT	递增地址击中 0: 页击中发生在页内任何一个地址 1: 页击中仅发生在递增的地址 设置页击中发生在页内任何一个地址或是仅发生在递增的地址。
[1:0]	PAGELEN	页长度 00: 一页有 4 个地址成员 01: 一页有 8 个地址成员 10: 一页有 16 个地址成员 11: 一页有 32 个地址成员 设置一页中地址成员的数量。

EBI 状态寄存器 – EBISR

该寄存器定义了 EBI 状态。

偏移量： 0x008

复位值： 0x0000\_0010

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	保留位								
类型 / 复位									
	15	14	13	12	11	10	9	8	
	保留位							EBISMRST	
类型 / 复位								RW	0
	7	6	5	4	3	2	1	0	
	保留位			EBIARDY	保留位			EBIBUSY	
类型 / 复位				RO	1				RO 0

位	字段	描述
[8]	EBISMRST	EBI 状态机复位 0: 正常 1: 复位 EBI 状态机 该位写 1 将使 EBI 内部状态机复位到初始状态且保持初始的寄存器设置。
[4]	EBIARDY	EBI 异步就绪状态 0: EBI_ARDY 无效 ( 设备处于忙碌状态 ) 1: EBI_ARDY 有效 ( 设备处于就绪状态 ) 该位指示 EBI_ARDY 引脚的状态。
[0]	EBIBUSY	EBI 忙碌 0: EBI 空闲 1: EBI 忙碌 该位置位用于指示 EBI 忙于 AHB 事务。

**EBI 地址时序寄存器 n – EBIATRn, n = 0 ~ 3**

该寄存器定义了每个 Bank n (n = 0 ~ 3) 的地址时序。

偏移量: 0x010 (n = 0), 0x20 (n = 1), 0x30 (n = 2), 0x40 (n = 3)

复位值: 0x0000\_0F0F

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位				ADDRHOLD			
类型 / 复位					RW	1	RW	1
	7	6	5	4	3	2	1	0
	保留位				ADDRSETUP			
类型 / 复位					RW	1	RW	1

位	字段	描述
[11:8]	ADDRHOLD	地址保持时间 设置 EBI_ALE 置位后 EBI_AD 总线上地址保持的周期数。该字段可清为 0。
[3:0]	ADDRSETUP	地址设置时间 设置 EBI_ALE 置位前 EBI_AD 总线上地址设置的周期数。若清为 0，硬件插入一个周期，该周期以 HCLK 时钟周期为单位。

EBI 读时序寄存器 n – EBIRTRn, n = 0 ~ 3

该寄存器定义了每个 Bank n (n = 0 ~ 3) 的读时序设置。

偏移量: 0x014 (n = 0), 0x24 (n = 1), 0x34 (n = 2), 0x44 (n = 3)  
复位值: 0x000F\_3F0F

	31	30	29	28	27	26	25	24
	保留位							PGEN
类型 / 复位								RW 0
	23	22	21	20	19	18	17	16
	保留位				RDHOLD			
类型 / 复位					RW 1	RW 1	RW 1	RW 1
	15	14	13	12	11	10	9	8
	保留位		RDSTRB					
类型 / 复位			RW 1	RW 1	RW 1	RW 1	RW 1	RW 1
	7	6	5	4	3	2	1	0
	保留位				RDSETUP			
类型 / 复位					RW 1	RW 1	RW 1	RW 1

位	字段	描述
[24]	PGEN	页使能 0: 页读取除能 1: 页读取使能 该位用于使能对应 Bank 的页读取模式。
[19:16]	RDHOLD	读保持时间 该位用于设置 EBI_OE 信号无效后 EBI_CS <sub>n</sub> 保持有效的周期数。该时间用于总线周转。
[13:8]	RDSTRB	读选通时间 设置 EBI_OE 保持有效的周期数。经过此特定的时间后，数据被读取。若该字段清为 0，硬件插入一个周期，该周期以 HCLK 时钟周期为单位。
[3:0]	RDSETUP	读设置时间 设置 EBI_OE 有效前地址设置所用的周期数，该周期以 HCLK 时钟周期为单位。

**EBI 写时序寄存器 n – EBIWTRn, n = 0 ~ 3**

该寄存器定义了每个 Bank n (n = 0 ~ 3) 的写时序设置。

偏移量: 0x018 (n = 0), 0x28 (n = 1), 0x38 (n = 2), 0x48 (n = 3)  
复位值: 0x000F\_3F0F

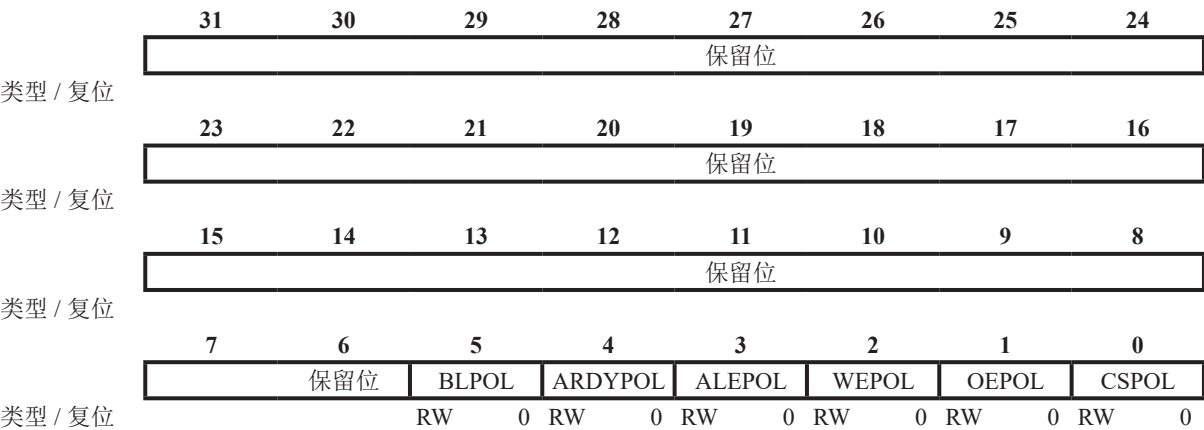
	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位				WEHOLD			
类型 / 复位					RW	1	RW	1
	15	14	13	12	11	10	9	8
	保留位		WESTRB					
类型 / 复位			RW	1	RW	1	RW	1
	7	6	5	4	3	2	1	0
	保留位				WESETUP			
类型 / 复位					RW	1	RW	1

位	字段	描述
[19:16]	WEHOLD	写保持时间 设置 EBI_WE 信号无效后 EBI_CS <sub>n</sub> 保持有效的周期数。
[13:8]	WESTRB	写选通时间 设置 EBI_WE 保持有效的周期数。若该字段清为 0，硬件插入一个周期，该周期以 HCLK 时钟周期为单位。
[3:0]	WESETUP	写设置时间 设置 EBI_WE 有效前地址设置所用的周期数，该周期以 HCLK 时钟周期为单位。

**EBI 极性寄存器 n – EBIPRn, n = 0 ~ 3**

该寄存器定义了每个 Bank n (n = 0 ~ 3) EBI 控制信号的极性。

偏移量: 0x01C (n = 0), 0x2C (n = 1), 0x3C (n = 2), 0x4C (n = 3)  
复位值: 0x0000\_0000



位	字段	描述
[5]	BLPOL	字节通道极性 0: EBI_BL 低有效 1: EBI_BL 高有效 该位设置 EBI_BL 引脚的极性。
[4]	ARDYPOL	异步就绪极性 0: EBI_ARDY 低有效 1: EBI_ARDY 高有效 该位设置 EBI_ARDY 引脚的极性。
[3]	ALEPOL	地址锁存极性 0: EBI_ALE 低有效 1: EBI_ALE 高有效 该位设置 EBI_ALE 引脚的极性。
[2]	WEPOL	写使能极性 0: EBI_WE 低有效 1: EBI_WE 高有效 该位设置 EBI_WE 引脚的极性。
[1]	OEPOL	输出使能极性 0: EBI_OE 低有效 1: EBI_OE 高有效 该位设置 EBI_OE 引脚的极性。
[0]	CSPOL	片选极性 0: EBI_CS 低有效 1: EBI_CS 高有效 该位设置 EBI_CS <sub>n</sub> 引脚的极性。



**EBI 中断使能寄存器 – EBIENR**

该寄存器定义了 EBI 中断使能制。

偏移量： 0x050

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位					ACCRSTIEN	ACCDISIEN	ARDYTOIEN
类型 / 复位						RW 0	RW 0	RW 0

位	字段	描述
[2]	ACCRSTIEN	EBI 状态机复位情况下处理事务时使能中断 (EBISR 寄存器的 EBISMRST 位设置为 1)
[1]	ACCDISIEN	访问除能的 Bank 时使能中断
[0]	ARDYTOIEN	EBI 异步就绪超时时使能中断

EBI 中断标志寄存器 – EBIIFR

该寄存器定义了 EBI 中断标志。

偏移量： 0x054  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位						ACCERRIF	ARDYTOIF
类型 / 复位							RO 0	RO 0

位	字段	描述
[1]	ACCERRIF	当 EBI 访问除能的 Bank 或者 EBI 状态机复位时此位置 1。
[0]	ARDYTOIF	当 EBI 异步就绪超时计数器达到预设值时此位置 1。

**EBI 中断清除寄存器 – EBIFCR**

该寄存器定义了 EBI 中断清除功能。

偏移量： 0x058  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位								
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位								
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位								
	7	6	5	4	3	2	1	0
	保留位						ACCERRIC	ARDYTOIC
类型 / 复位							WO 0	WO 0

位	字段	描述
[1]	ACCERRIC	写 1 清除访问除能的 Bank 或 EBI 状态机复位时产生的中断标志位。若软件读取此位仅返回 0 值。
[0]	ARDYTOIC	写 1 清除 EBI 异步就绪超时中断标志位。若软件读取此位仅返回 0 值。

## 27 内置音频接口 (I²S)

### 简介

I²S 是一个同步通信接口, 可用作主机或从机与其它音频外设, 如 ADC 或 DAC, 进行数据交换。I²S 支持多种数据格式。除了立体声 I²S 对齐、左对齐和右对齐模式外, 还有 8/16/24/32-bit 采样大小的单声道 PCM 模式。当 I²S 工作于主机模式时, 若使用小数分频器, 该接口可提供一个精确的采样频率输出并且支持频率控制功能和输出频率微调功能, 从而避免了因不同设备之间的累积频率误差造成的系统问题。

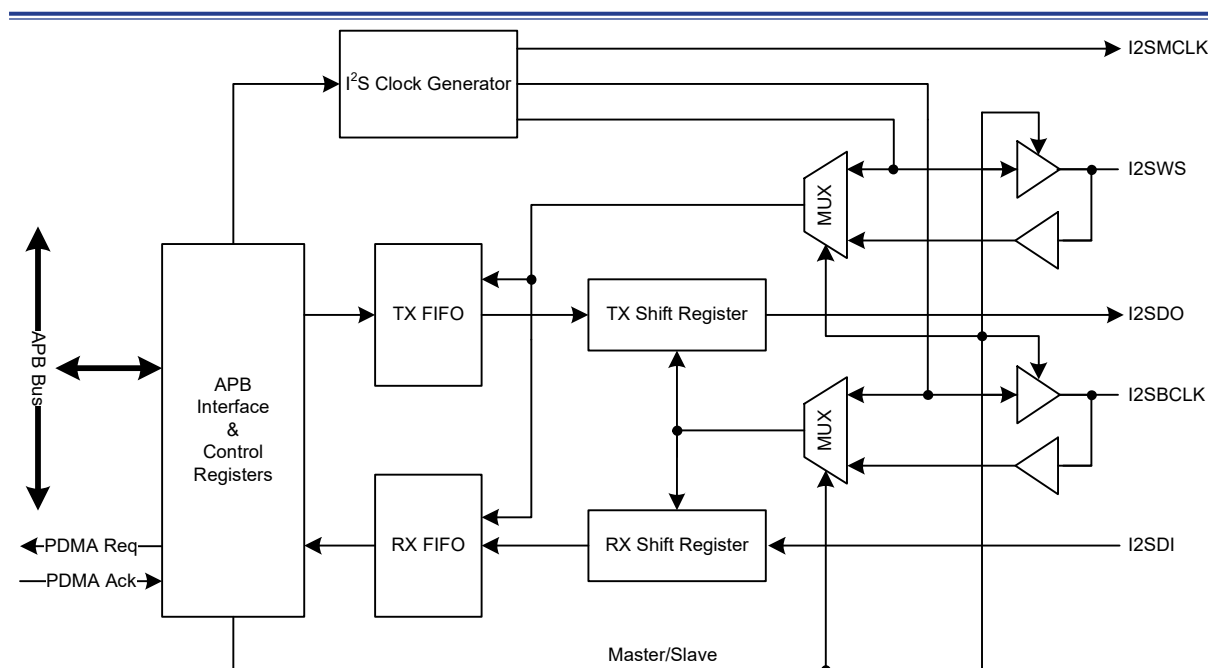


图 194. I²S 方框图

### 特性

- 主机或从机模式
- 单声道和立体声
- I²S 对齐模式、左对齐和右对齐模式
- 使用 32-bit 通道扩展支持 8/16/24/32-bit 采样大小
- 8 × 32-bit TX & RX FIFO, 支持 PDMA
- 8-bit 小数时钟分频器, 带频率控制

## 功能描述

### I<sup>2</sup>S 主机和从机模式

I<sup>2</sup>S 可工作于从机或主机模式。在 I<sup>2</sup>S 模块中，这两个模式的区别在于决定数据传输时序的字选择 (WS) 信号。

- 在主机模式下，字选择信号由内部时钟频率发生器产生。
- 在从机模式下，字选择信号通过 I2S\_WS 引脚输入。
- 当 I<sup>2</sup>S 总线使能时，字选择信号和位时钟信号由总线主机连续发出。
- 静音控制位使发送通道处于静音状态。当静音模式使能时，发送通道 FIFO 正常运行，但输出数据流将被丢弃使用 0 替代。该位并不影响接收通道，所以数据接收正常进行。

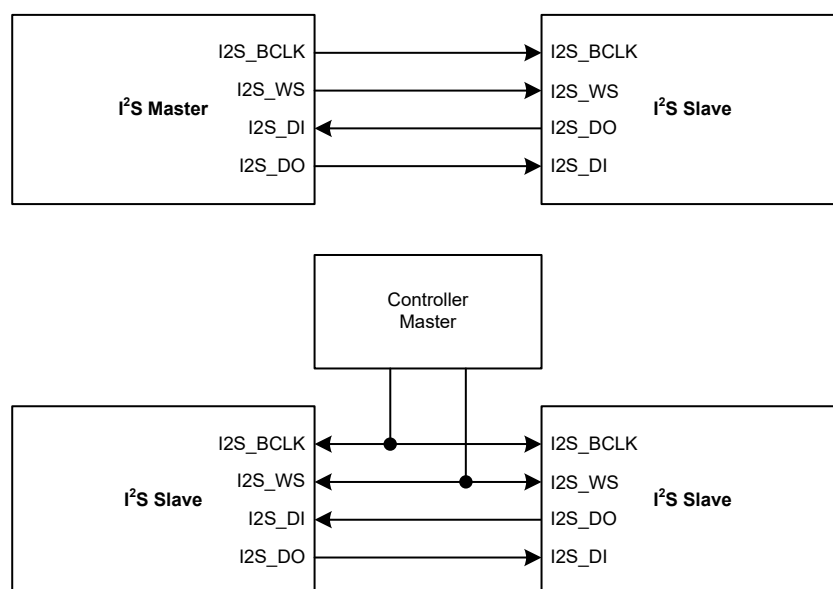


图 195. I<sup>2</sup>S 主 / 从机配置简化图

## I<sup>2</sup>S 时钟频率发生器

I<sup>2</sup>S 的主时钟 (I2S\_MCLK) 和位时钟 (I2S\_BCLK) 由 I2SCDR 寄存器里的相关值决定。I<sup>2</sup>S 位时钟频率设置决定于所需的音频采样频率、所用的数据格式 (立体声或单声道) 以及数据大小。主时钟频率 (I2S\_MCLK) 产生需要使用一个小数时钟分频器, 此分频器将 I<sup>2</sup>S PCLK 频率进行分频。合理设置分子 (X) 和分母 (Y) 的值从而产生一个主时钟 (I2S\_MCLK) 两倍的频率值。分频器的输出频率除以 2 从而使输出时钟占空比更均匀。I<sup>2</sup>S 时钟发生器方框图如图 196 所示。小数频率分频器公式如下:

$$I2S\_MCLK = 1/2 \times PCLK \times (X/Y), X/Y \leq 1, X = 1 \sim 255, Y = 1 \sim 255$$

$$I2S\_BCLK = I2S\_MCLK / (N + 1), N = 0 \sim 255$$

由于小数频率分频器是一个完全数字实现的功能, 分频器输出时钟转换同步于输入时钟源。对于一些分频器设置, 将产生一些频率抖动。用户在选择 X 和 Y 值时需注意这个现象。当选择的分数中 X 可整除 Y, 可完全避免频率抖动, 如 2/4、2/6、3/9 等。

下面的表格显示了不同时钟源和采样频率情况下的推荐设置值以减少时钟抖动。

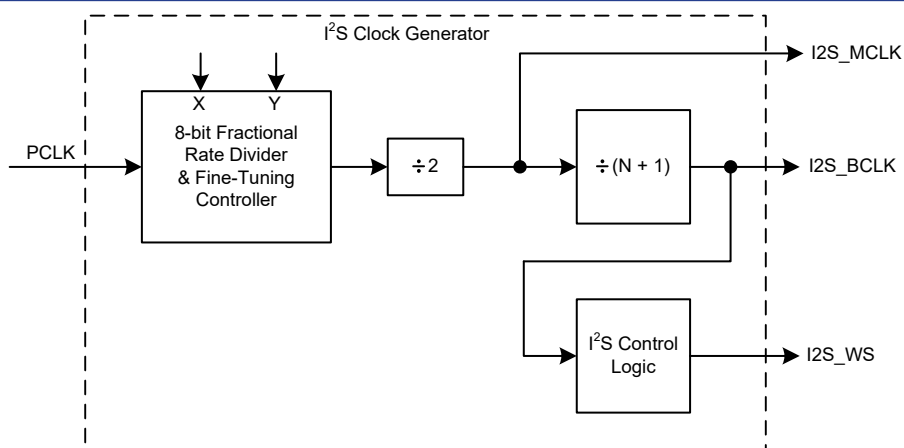


图 196. I<sup>2</sup>S 时钟发生器方框图

表 73.  $F_s$  建议值列表 @ 8 MHz PCLK

	512 $F_s$		384 $F_s$		256 $F_s$		192 $F_s$		128 $F_s$		64 $F_s$	
$F_s$ (Hz)	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	—	—	96	125	64	125	48	125	32	125	16	125
11,025	—	—	—	—	170	241	118	223	90	255	42	238
12,000	—	—	—	—	96	125	72	125	48	125	24	125
16,000	—	—	—	—	—	—	96	125	64	125	32	125
22,050	—	—	—	—	—	—	—	—	170	241	90	255
24,000	—	—	—	—	—	—	—	—	96	125	48	125
32,000	—	—	—	—	—	—	—	—	—	—	64	125
44,100	—	—	—	—	—	—	—	—	—	—	170	241
48,000	—	—	—	—	—	—	—	—	—	—	96	125
96,000	—	—	—	—	—	—	—	—	—	—	—	—
192,000	—	—	—	—	—	—	—	—	—	—	—	—

表 74.  $F_s$  建议值列表 @ 48 MHz PCLK

	512 $F_s$		384 $F_s$		256 $F_s$		192 $F_s$		128 $F_s$		64 $F_s$	
$F_s$ (Hz)	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	36	211	16	125	18	211	8	125	10	234	2	94
11,025	4	17	6	34	2	17	6	68	2	34	2	68
12,000	32	125	24	125	16	125	12	125	8	125	4	125
16,000	86	252	32	125	36	211	16	125	18	211	10	234
22,050	8	17	6	17	4	17	6	34	2	17	2	34
24,000	64	125	48	125	32	125	24	125	16	125	8	125
32,000	142	208	64	125	86	252	32	125	36	211	18	211
44,100	238	253	170	241	8	17	6	17	4	17	2	17
48,000	—	—	96	125	64	125	48	125	32	125	16	125
96,000	—	—	—	—	—	—	96	125	64	125	32	125
192,000	—	—	—	—	—	—	—	—	—	—	64	125

表 75.  $F_s$  建议值列表 @ 72 MHz PCLK

	512 $F_s$		384 $F_s$		256 $F_s$		192 $F_s$		128 $F_s$		64 $F_s$	
$F_s$ (Hz)	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	24	211	18	211	12	211	10	234	6	211	2	141
11,025	8	51	2	17	4	51	2	34	2	51	2	102
12,000	36	211	16	125	18	211	8	125	10	234	2	94
16,000	38	167	36	211	24	211	18	211	12	211	6	211
22,050	74	236	4	17	8	51	2	17	4	51	2	51
24,000	86	252	32	125	36	211	16	125	18	211	10	234
32,000	76	167	86	252	38	167	36	211	24	211	12	211
44,100	106	169	8	17	74	236	4	17	8	51	4	51
48,000	142	208	64	125	86	252	32	125	36	211	18	211
96,000	—	—	—	—	142	208	64	125	86	252	36	211
192,000	—	—	—	—	—	—	—	—	142	208	86	252

表 76.  $F_s$  建议值列表 @ 96 MHz PCLK

	512 $F_s$		384 $F_s$		256 $F_s$		192 $F_s$		128 $F_s$		64 $F_s$	
$F_s$ (Hz)	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	18	211	8	125	10	234	4	125	2	94	2	188
11,025	2	17	6	68	2	34	6	136	2	68	2	136
12,000	16	125	12	125	8	125	6	125	4	125	2	125
16,000	36	211	16	125	18	211	8	125	10	234	2	94
22,050	4	17	6	34	2	17	6	68	2	34	2	68
24,000	32	125	24	125	16	125	12	125	8	125	4	125
32,000	86	252	32	125	36	211	16	125	18	211	10	234
44,100	8	17	86	252	4	17	6	34	2	17	2	34
48,000	64	125	48	125	32	125	24	125	16	125	8	125
96,000	—	—	96	125	64	125	48	125	32	125	16	125
192,000	—	—	—	—	—	—	96	125	64	125	32	125

I²S 接口格式

I²S 对齐立体声模式

在标准的 I²S 对齐模式下，立体音频采样数据的最高位 (MSB) 在 WS 信号变化后的第二个 BCLK 时钟上升时出现。在立体声模式下，WS 低电平状态表示正在传输的是左通道数据，高电平状态表示右通道数据。图 197 和图 198 显示了标准 I²S 对齐立体声模式的格式。

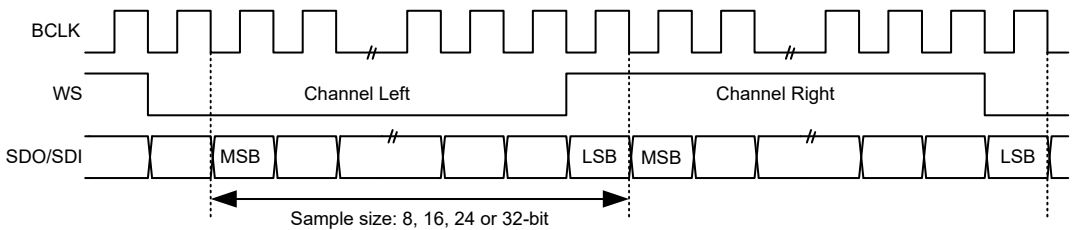


图 197. I²S 对齐立体声模式波形

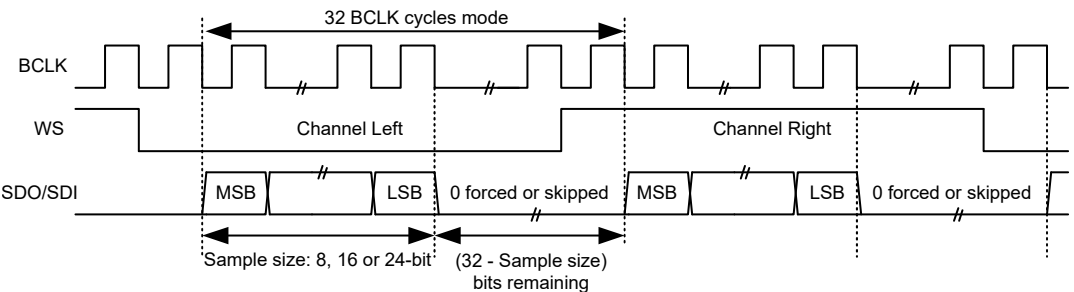


图 198. I²S 对齐立体声模式波形 (32-bit 通道扩展)



### 左对齐立体声模式

在左对齐模式下，立体音频采样数据的最高位 (MSB) 在 WS 信号变化后的第一个 BCLK 时钟上升时出现。图 199 和图 200 显示了左对齐立体声模式的格式。

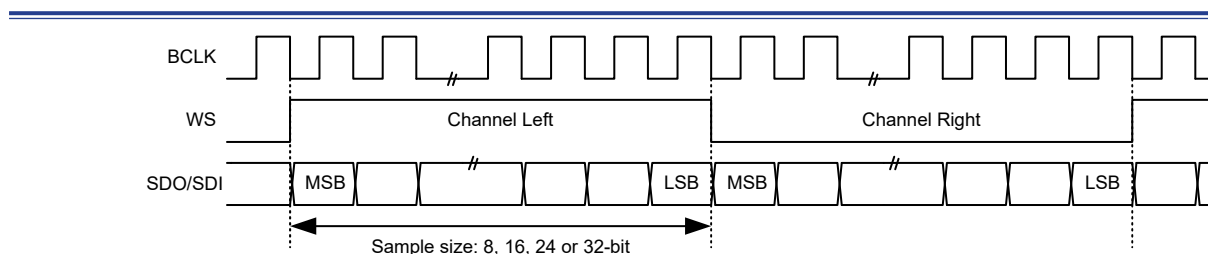


图 199. 左对齐立体声模式波形

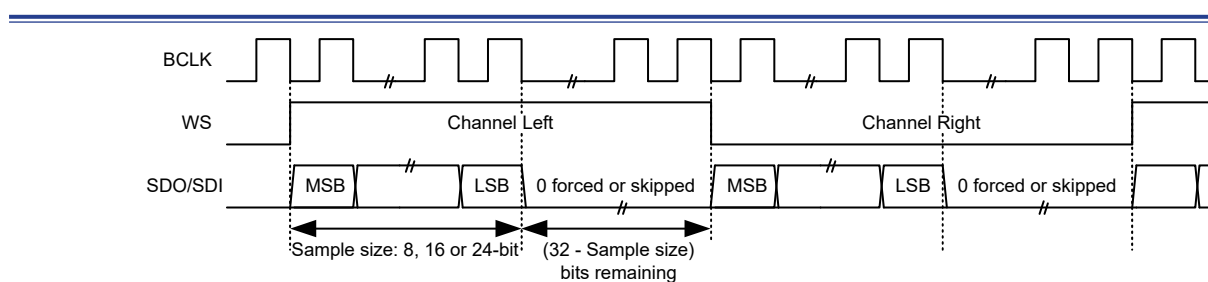


图 200. 左对齐立体声模式波形 ( 32-bit 通道扩展 )

### 右对齐立体声模式

在右对齐模式下，立体音频采样数据的最低位 (LSB) 在 WS 信号变化前的那一个 BCLK 时钟上升沿出现，MSB 优先传输。图 201 和图 202 显示了右对齐立体声模式的格式。

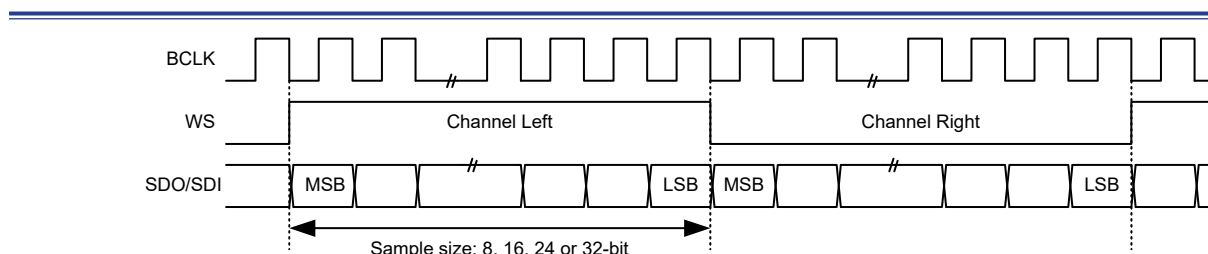


图 201. 右对齐立体声模式波形

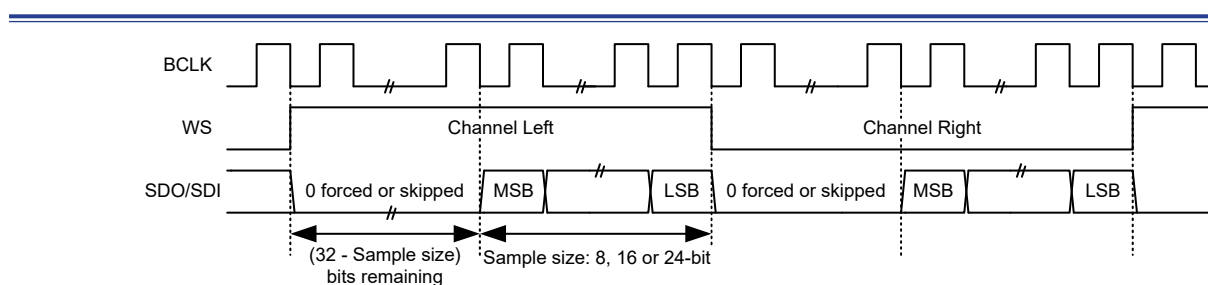


图 202. 右对齐立体声模式波形 ( 32-bit 通道扩展 )

### I<sup>2</sup>S 对齐单声道模式

在 I<sup>2</sup>S 对齐单声道模式下，单声道音频采样数据的最高位 (MSB) 在 WS 信号下降沿后的第二个 BCLK 时钟上升出现。图 203 和图 204 显示了 I<sup>2</sup>S 对齐单声道模式的格式。

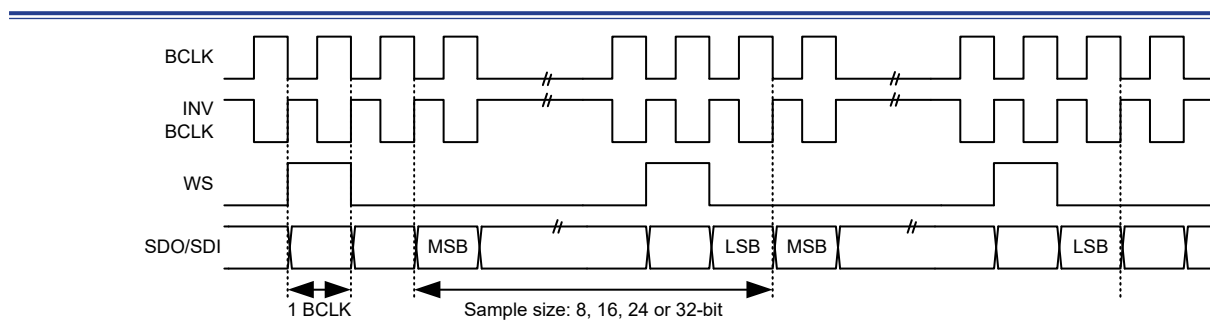


图 203. I<sup>2</sup>S 对齐单声道模式波形

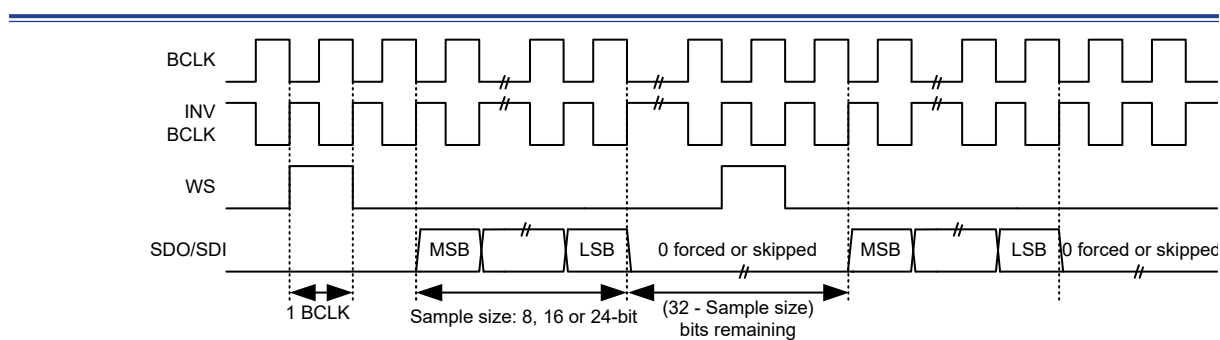


图 204. I<sup>2</sup>S 对齐单声道模式波形 (32-bit 通道扩展)

### 左对齐单声道模式

在左对齐单声道模式下，单声道音频采样时数据的最高位 (MSB) 在 WS 信号下降沿后的第一个 BCLK 上升沿时出现。图 205 和图 206 显示了左对齐单声道模式的格式。

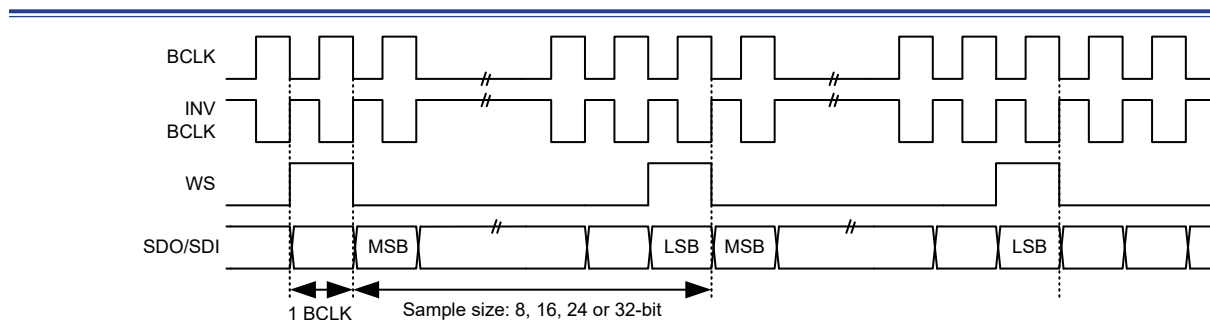


图 205. 左对齐单声道模式波形

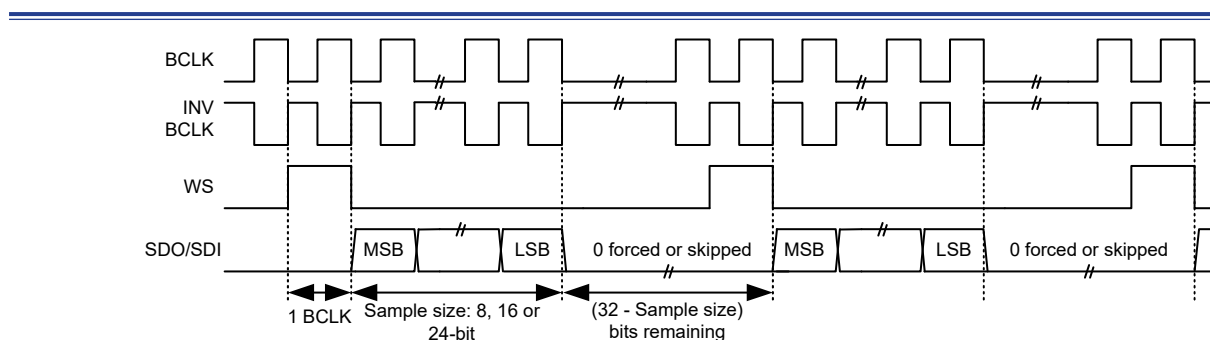


图 206. 左对齐单声道模式波形 ( 32-bit 通道扩展 )

### 右对齐单声道模式

在右对齐单声道模式下，单声道音频采样数据的最低位 (LSB) 在 WS 信号上升沿之前的最后一个 BCLK 上升沿出现。图 207 和图 208 显示了右对齐单声道模式的格式。

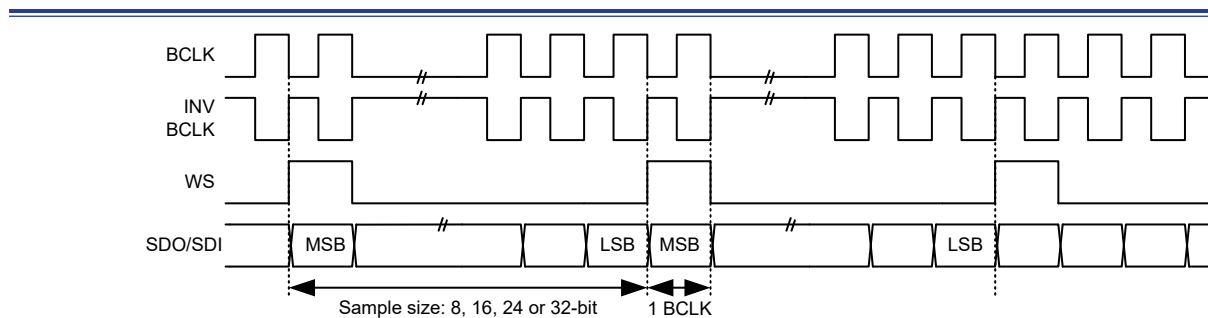


图 207. 右对齐单声道模式波形

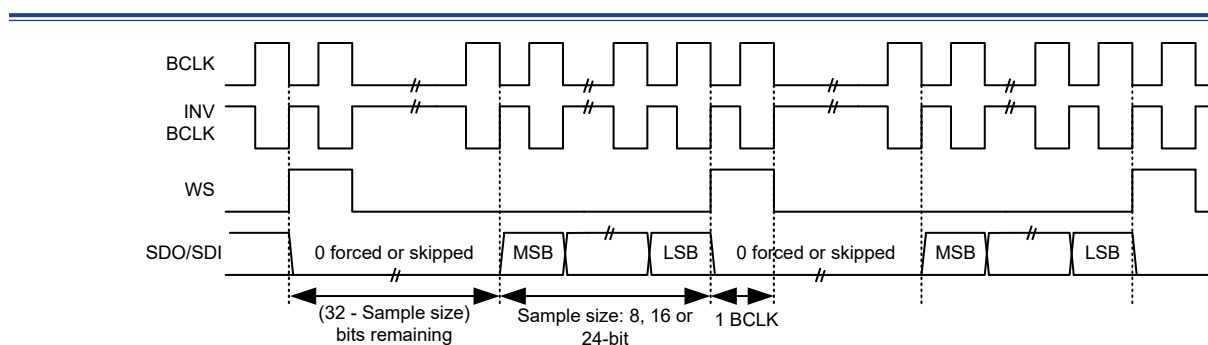


图 208. 右对齐单声道模式波形 ( 32-bit 通道扩展 )

### I<sup>2</sup>S 对齐重复模式

在 I<sup>2</sup>S 对齐重复模式下，单声道音频采样数据的最高位 (MSB) 在 WS 信号变化后的第二个 BCLK 时钟上升沿出现。在此模式下，相同的数据被发送两次，一次在 WS 信号为低时发送，一次在 WS 信号为高时发送。图 209 和图 210 显示了 I<sup>2</sup>S 对齐重复模式的格式。

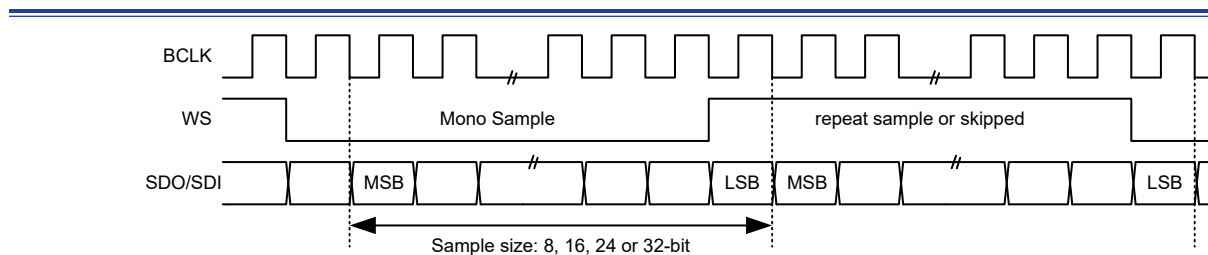


图 209. I<sup>2</sup>S 对齐重复模式波形

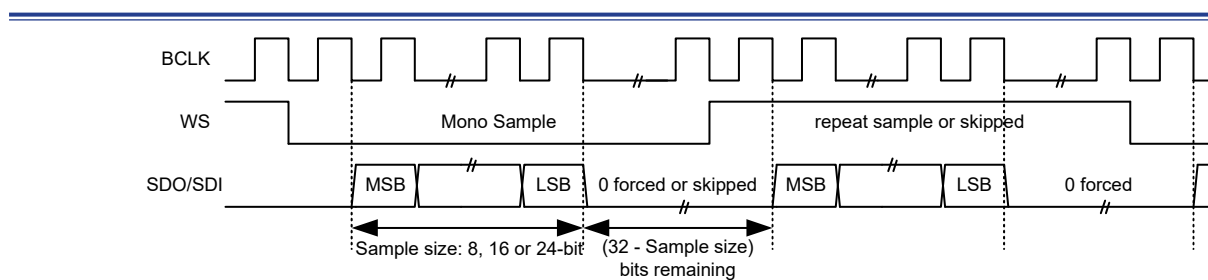


图 210. I<sup>2</sup>S 对齐重复模式波形 (32-bit 通道扩展)

FIFO 控制和分配

I<sup>2</sup>S 通过 FIFO 控制器处理发送和接收的音频数据。每个发送或接收的 FIFO 深度为 8 个字，即 8 × 32-bit，用作数据缓冲。传输格式取决于立体声 / 单声道模式以及采样大小的设置。详细的 FIFO 内容格式如图 211 所示。FIFO 控制器包含的比较器会将当前 FIFO 级别与预设的深度进行比较。当前的 TX FIFO 或 RX FIFO 级别状态可分别通过 I<sup>2</sup>S 状态寄存器 I2SSR 的 TXFS 和 RXFS 字段查询。

				FIFO Pointer Address
Mono 8-bit Data				
7	N+3	0	7 N+2 0 7 N+1 0 7 N 0	M
Mono 16-bit Data				
15	N+1	0	15 N 0	M
Mono 24-bit Data				
31	23	N	0	M
Mono 32-bit Data				
31	N	0		M
Stereo 8-bit Data				
7	LEFT+1	0	7 RIGHT+1 0 7 LEFT 0 7 RIGHT 0	M
Stereo 16-bit Data				
15	LEFT	0	15 RIGHT 0	M
Stereo 24-bit Data				
31	23	LEFT	0	M
31	23	RIGHT	0	M+1
Stereo 32-bit Data				
31	LEFT	0		M
31	RIGHT	0		M+1

图 211. 各个模式下 FIFO 数据内容分配

PDMA 和中断

当 RX FIFO 中接收的数据深度级别等于或大于 I<sup>2</sup>S FIFO 控制寄存器 I2SFCR 中 RXFTLS 字段定义的级别时, 相关的 RXFTL 标志位将置位, 接着产生一个 I<sup>2</sup>S RX PDMA 请求。若 I<sup>2</sup>S RX PDMA 请求使能位或 RX FIFO 触发级别中断使能位置位时, 将产生一个 MCU 中断。当 TX FIFO 中发送的数据深度级别等于或小于 I<sup>2</sup>S FIFO 控制寄存器 I2SFCR 中 TXFTLS 字段定义的级别时, 相关的 TXFTL 标志位将置位, 并产生一个 I<sup>2</sup>S TX PDMA 请求。若 I<sup>2</sup>S TX PDMA 请求使能位或 TX FIFO 触发级别中断使能位置位时, 将产生一个 MCU 中断。

I<sup>2</sup>S 发送器和接收器有各自的 PDMA 请求, 分配给两个不同的 PDMA 通道。当 I<sup>2</sup>S 发送器 PDMA 请求使能, 即 TXDMAEN 为 1 时, 自动请求当 TX FIFO 空间可用且 TXFTL 有效时将数据发送到分配好的 I<sup>2</sup>S TX PDMA 通道。当 I<sup>2</sup>S 接收器 PDMA 请求使能, 即 RXDMAEN 为 1 时, 自动请求当 RX FIFO 接收到数据且 RXFTL 有效时将数据发送到分配好的 I<sup>2</sup>S RX PDMA 通道。

寄存器列表

下表显示了 I<sup>2</sup>S 寄存器和复位值。

表 77. I<sup>2</sup>S 寄存器列表

寄存器	偏移量	描述	复位值
I2SCR	0x000	I <sup>2</sup> S 控制寄存器	0x0000_0000
I2SIER	0x004	I <sup>2</sup> S 中断使能寄存器	0x0000_0000
I2SCDR	0x008	I <sup>2</sup> S 时钟分频寄存器	0x0000_0000
I2STXDR	0x00C	I <sup>2</sup> S TX 数据寄存器	0x0000_0000
I2SRXDR	0x010	I <sup>2</sup> S RX 数据寄存器	0x0000_0000
I2SFCR	0x014	I <sup>2</sup> S FIFO 控制寄存器	0x0000_0000
I2SSR	0x018	I <sup>2</sup> S 状态寄存器	0x0000_0809
I2SRCNTR	0x01C	I <sup>2</sup> S 频率计数值寄存器	0x0000_0000



## 寄存器描述

### I<sup>2</sup>S 控制寄存器 – I2SCR

该寄存器定义了相关的 I<sup>2</sup>S 功能使能控制。

偏移量: 0x000

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位				MCKINV	BCKINV	RCSEL	RCEN
				RW 0	RW 0	RW 0	RW 0
15	14	13	12	11	10	9	8
CLKDEN	RXDMAEN	TXDMAEN	TXMUTE	CHANNEL	REPEAT	MCLKEN	BITEXT
RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
7	6	5	4	3	2	1	0
FORMAT		SMPSIZE		MS	RXEN	TXEN	I2SEN
RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

位	字段	描述
[19]	MCKINV	MCLK 反相使能 0: 除能 1: 使能
[18]	BCKINV	BCLK 反相使能 0: 除能 1: 使能
[17]	RCSEL	频率控制选择 (仅适用于主机模式) 0: 慢速 1: 快速
[16]	RCEN	频率控制使能 (仅适用于主机模式) 0: 除能 1: 使能
[15]	CLKDEN	时钟分频器使能 (仅适用于主机模式) 0: 除能 1: 使能 在 I <sup>2</sup> S 接口主机模式下时钟分频器用于产生 MCLK 和 BCLK 时钟。
[14]	RXDMAEN	RX PDMA 请求使能 0: 除能 1: 使能
[13]	TXDMAEN	TX PDMA 请求使能 0: 除能 1: 使能
[12]	TXMUTE	TX 静音使能 0: 除能 1: 使能

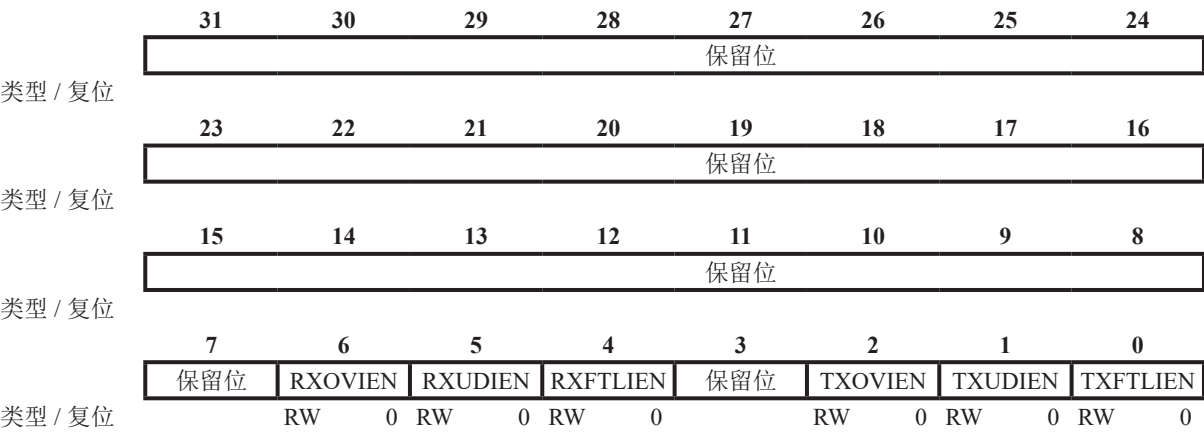
位	字段	描述
[11]	CHANNEL	立体声道或单声道 0: 立体声道 1: 单声道 注: 该位应该在 I <sup>2</sup> S 除能时预先设置。
[10]	REPEAT	重复模式 0: 除能 1: 使能 此模式仅适用于 I <sup>2</sup> S 对齐立体声配置。在两个通道都发送单声道数据, 但只接收左通道的数据, 忽略右通道的数据。重复模式只有在 CHANNEL 被设置为立体声道时可用。 注: 该位应该在 I <sup>2</sup> S 除能时预先设置。
[9]	MCLKEN	MCLK 输出使能 (仅适用于主机模式) 0: 除能 1: 使能 注: 该位应该在 I <sup>2</sup> S 除能时预先设置。
[8]	BITEXT	32-bit 通道使能 0: 除能 1: 使能 此位置位将迫使通道大小变成 32-bit。若采样大小为 8/16/24-bit, 剩下的位在 TX 模式下用 0 取代, 在 RX 模式下被忽略。 注: 该位应该在 I <sup>2</sup> S 除能时预先设置。
[7:6]	FORMAT	数据格式 00: I <sup>2</sup> S 对齐 01: 左对齐 10: 右对齐 11: 保留 注: 该位应该在 I <sup>2</sup> S 除能时预先设置。
[5:4]	SMPSIZE	采样大小 00: 8-bit 01: 16-bit 10: 24-bit 11: 32-bit 注: 该位应该在 I <sup>2</sup> S 除能时预先设置。
[3]	MS	主机或从机模式 0: 主机模式 1: 从机模式 注: 该位应该在 I <sup>2</sup> S 除能时预先设置。
[2]	RXEN	RX 使能 0: 除能 1: 使能
[1]	TXEN	TX 使能 0: 除能 1: 使能
[0]	I2SEN	I <sup>2</sup> S 使能 0: 除能 1: 使能

I²S 中断使能寄存器 – I2SIER

该寄存器定义了相关的 I²S 中断使能位。

偏移量：0x004

复位值：0x0000\_0000



位	字段	描述
[6]	RXOVIEN	RX FIFO 溢出中断使能 0: 除能 1: 使能
[5]	RXUDIEN	RX FIFO 下溢中断使能 0: 除能 1: 使能
[4]	RXFTLIEN	RX FIFO 触发级别中断使能 0: 除能 1: 使能
[2]	TXOVIEN	TX FIFO 溢出中断使能 0: 除能 1: 使能
[1]	TXUDIEN	TX FIFO 下溢中断使能 0: 除能 1: 使能
[0]	TXFTLIEN	TX FIFO 触发级别中断使能 0: 除能 1: 使能

I²S 时钟分频寄存器 – I2SCDR

该寄存器定义了 I²S 时钟分频比。

偏移量： 0x008  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	N_DIV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	X_DIV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	Y_DIV								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[23:16]	N_DIV	分频器 N 值用于产生 BCLK 0x00: 分频器 1 0x01: 分频器 2 ... 0xFF: 分频器 256 注：该位应该在 I²S 除能时预先设置。
[15:8]	X_DIV	分频器 X 值用于产生 MCLK (X = 1 ~ 255) && (X/Y ≤ 1) 注：该位应该在 I²S 除能时预先设置。
[7:0]	Y_DIV	分频器 Y 值用于产生 MCLK (Y = 1 ~ 255) && (X/Y ≤ 1) 注：该位应该在 I²S 除能时预先设置。

## I<sup>2</sup>S TX 数据寄存器 – I2STXDR

该寄存器用于定义 I<sup>2</sup>S 发送数据。

偏移量: 0x00C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	TXDR								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	TXDR								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	TXDR								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	TXDR								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO

位	字段	描述
[31:0]	TXDR	TX 数据寄存器

## I<sup>2</sup>S RX 数据寄存器 – I2SRXDR

该寄存器用于存储 I<sup>2</sup>S 接收的数据。

偏移量: 0x010

复位值: 0x0000\_0000

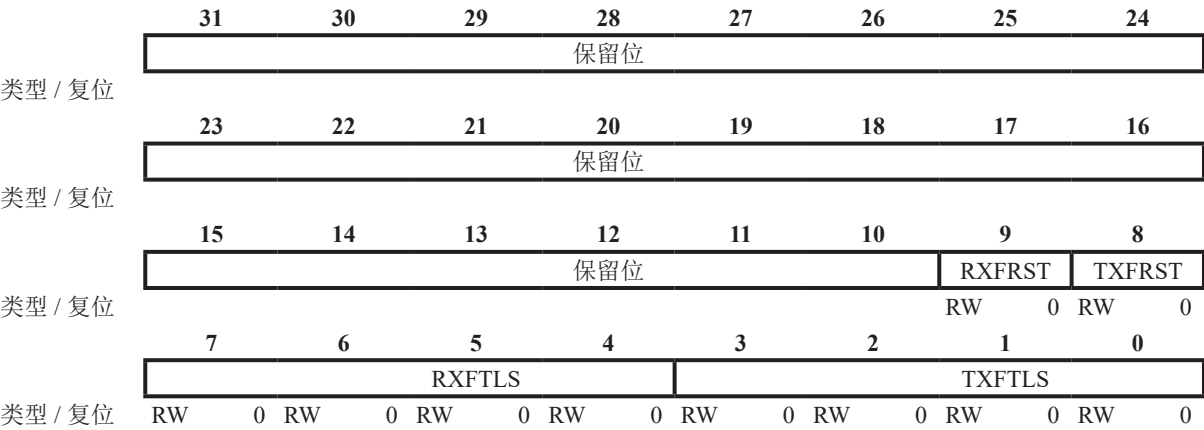
	31	30	29	28	27	26	25	24	
	RXDR								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	RXDR								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	RXDR								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	RXDR								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[31:0]	RXDR	RX 数据寄存器

I²S FIFO 控制寄存器 – I2SFCR

该寄存器定义了相关的 I²S FIFO 控制位。

偏移量： 0x014  
复位值： 0x0000\_0000



位	字段	描述
[9]	RXFRST	RX FIFO 复位 设置该位将复位 RX FIFO。
[8]	TXFRST	TX FIFO 复位 设置该位将复位 TX FIFO。
[7:4]	RXFTLS	RX FIFO 触发级别选择 0000: 触发级别为 0 0001: 触发级别为 1 ... 0111: 触发级别为 7 1xxx: 触发级别为 8 当 RX FIFO 中的数据量等于或大于 RXFTLS 字段定义的级别，则 RXFTL 标志位将置位。
[3:0]	TXFTLS	TX FIFO 触发级别选择 0000: 触发级别为 0 0001: 触发级别为 1 ... 0111: 触发级别为 7 1xxx: 触发级别为 8 当 TX FIFO 中的数据量等于或小于 TXFTLS 字段定义的级别，则 TXFTL 标志位将置位。

## I<sup>2</sup>S 状态寄存器 – I2SSR

该寄存器定义了相关的 I<sup>2</sup>S 状态。

偏移量: 0x018

复位值: 0x0000\_0809

	31	30	29	28	27	26	25	24
	RXFS							
类型 / 复位	RO	0	RO	0	RO	0	RO	0
	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位						RO	0	RO
	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位				RO	0	RO	1	WC
	7	6	5	4	3	2	1	0
	保留位							
类型 / 复位				RO	0	RO	1	WC

位	字段	描述
[31:28]	RXFS	RX FIFO 状态 0000: RX FIFO 为空 0001: RX FIFO 包含 1 个数据 ... 1000: RX FIFO 包含 8 个数据 其它: 保留
[27:24]	TXFS	TX FIFO 状态 0000: TX FIFO 为空 0001: TX FIFO 包含 1 个数据 ... 1000: TX FIFO 包含 8 个数据 其它: 保留
[18]	CLKRDY	时钟分频器输出就绪标志位 0: 未就绪 1: 就绪
[17]	TXBUSY	TX 忙标志位 0: 不忙 1: 忙
[16]	CHS	通道状态 0: 左通道 1: 右通道
[12]	RXFFUL	RX FIFO 满标志位 0: RX FIFO 未滿 1: RX FIFO 已滿
[11]	RXFEMT	RX FIFO 空标志位 0: RX FIFO 非空 1: RX FIFO 为空

位	字段	描述
[10]	RXFOV	RX FIFO 上溢标志位 0: RX FIFO 未上溢 1: RX FIFO 上溢 该位由硬件置位，通过写 1 清零。
[9]	RXFUD	RX FIFO 下溢标志位 0: RX FIFO 未下溢 1: RX FIFO 下溢 该位由硬件置位，通过写 1 清零。
[8]	RXFTL	RX FIFO 触发级别标志位 0: RX FIFO 中的数据小于触发级别 1: RX FIFO 中的数据等于或大于触发级别 该位由硬件置位，通过写 1 清零。
[4]	TXFFUL	TX FIFO 满标志位 0: TX FIFO 未滿 1: TX FIFO 已滿
[3]	TXFEMT	TX FIFO 空标志位 0: TX FIFO 非空 1: TX FIFO 为空
[2]	TXFOV	TX FIFO 上溢标志位 0: TX FIFO 未上溢 1: TX FIFO 上溢 该位由硬件置位，通过写 1 清零。
[1]	TXFUD	TX FIFO 下溢标志位 0: TX FIFO 未下溢 1: TX FIFO 下溢 该位由硬件置位，通过写 1 清零。
[0]	TXFTL	TX FIFO 触发级别标志位 0: TX FIFO 中的数据大于触发级别 1: TX FIFO 中的数据等于或小于触发级别 1 该位由硬件置位，通过写 1 清零。



I²S 频率计数值寄存器 – I2SRCNTR

该寄存器定义了 I²S 频率控制计数值。

偏移量： 0x01C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24				
	保留位											
类型 / 复位												
	23	22	21	20	19	18	17	16				
	保留位				RCNTR							
类型 / 复位					RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8				
	RCNTR											
类型 / 复位												
	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0				
	RCNTR											
类型 / 复位												
	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[19:0]	RCNTR	频率计数值 该值必须大于 0 以实现有效的频率微调控制。 $RCSEL = 1, RCNTR = 1 \sim (2Y - 1): MCLK' = (1 + \frac{1}{2Y - 1}) \times MCLK$ $RCSEL = 1, RCNTR > (2Y - 1): MCLK' = (1 + \frac{1}{RCNTR}) \times MCLK$ $RCSEL = X, RCNTR = 0: MCLK' = MCLK$ $RCSEL = 0, RCNTR > (2Y + 1): MCLK' = (1 - \frac{1}{RCNTR}) \times MCLK$ $RCSEL = 0, RCNTR = 1 \sim (2Y + 1): MCLK' = (1 - \frac{1}{2Y + 1}) \times MCLK$

## 28 循环冗余校验 (CRC)

### 简介

循环冗余校验 (CRC) 计算单元是一种错误检测技术测试算法，用于验证数据传输或存储数据的正确性。CRC 计算将数据流或数据块作为输入，并生成一个 16-bit 或 32-bit 的输出余数。通常情况下，一个数据流带有 CRC 后缀码，当被发送或存储时该数据流可用作校验和。因此，被接收或重新存储的数据流都是通过上述相同的生成多项式计算得到的。若新的 CRC 码结果与先前计算的不匹配，这意味着数据流出错。

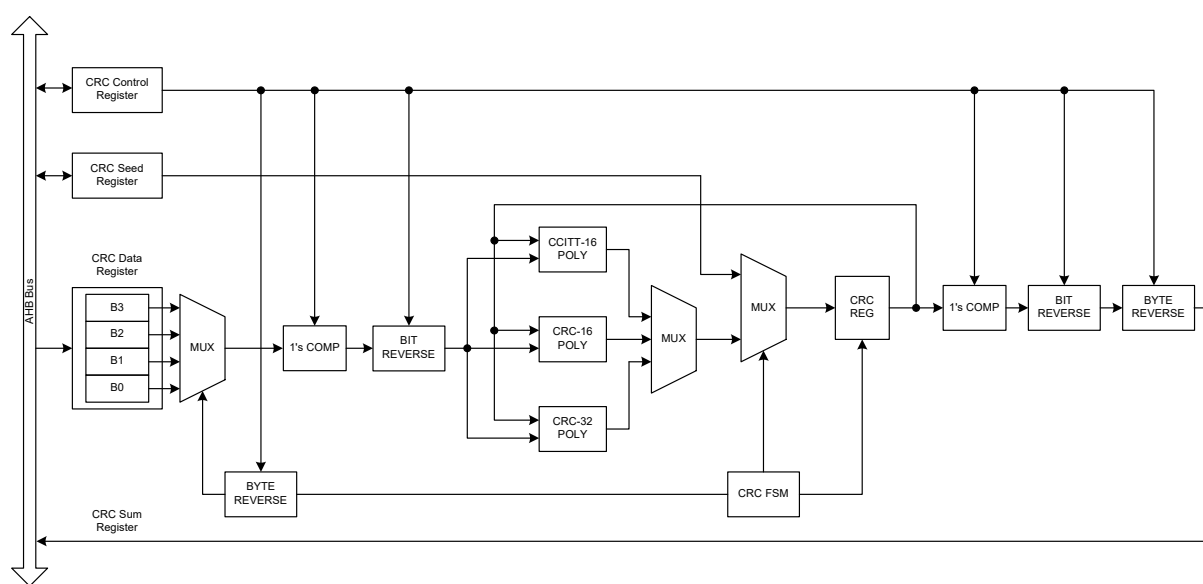


图 212. CRC 方框图

## 特性

- 支持 CRC16 多项式:  $0x8005$ ,  $X^{16} + X^{15} + X^2 + 1$
- 支持 CCITT CRC16 多项式:  $0x1021$ ,  $X^{16} + X^{12} + X^5 + 1$
- 支持 IEEE-802.3 CRC32 多项式:  $0x04C11DB7$ ,  
 $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 支持对数据和校验和进行反码、按字节反序以及按位反序操作
- 支持字节、半字和字数据大小
- 可编程 CRC 初始种子值
- 对 8-bit 数据执行 CRC 计算需要一个 AHB 时钟周期, 32-bit 数据需要 4 个 AHB 时钟周期
- 支持 PDMA 对指定存储器区块进行 CRC 计算

## 功能描述

此 CRC 单元仅允许 CRC16、CCITT CRC16 和 IEEE-802.3 CRC32 多项式的计算。在此单元内, 生成多项式有各自固定的二进制值。因此, 基于其他生成多项式的 CRC 值不可计算。

### CRC 计算

CRC 计算单元具有一个 32-bit 的写 CRC 数据寄存器 (CRCDR) 和读 CRC 校验和寄存器 CRCCSR。CRCDR 寄存器用于输入新数据 (写访问), 而 CRCCSR 寄存器用于保持先前的 CRC 计算的结果 (读访问)。每次对 CRC 数据寄存器 CRCDR 进行写操作, 都会将存储在 CRCCSR 中的前个 CRC 值和新的 CRC 值结合起来。CRC 方框图如图 212 所示。CRC 单元计算 CRC 数据寄存器值是按字节进行的, 默认的字节和位的顺序为大端模式。CRCDR 寄存器可以通过字、右对齐半字节和右对齐字节的方式进行写访问。对于其它寄存器仅允许以 32-bit 格式进行访问。计算持续时间取决于数据宽度:

- 输入数据为 32-bit: 需要 4 个 AHB 时钟周期
- 输入数据为 16-bit: 需要 2 个 AHB 时钟周期
- 输入数据为 8-bit: 需要 1 个 AHB 时钟周期

### CRC 计算按字节反序与位反序

字节重新排序和字节内位反序操作可出现在数据用于 CRC 计算之前或 CRC 校验和输出后, 可通过 CRCCR 寄存器相应的设定字段配置。这些操作发生在字或半字节访问时。字节写访问时硬件会忽略 CRCCR 寄存器中的 DATBYRV 位, 但在字节写入时出现按位反序仍可置位 DATBIRV 位。图 213 为按字节和位反序操作范例。

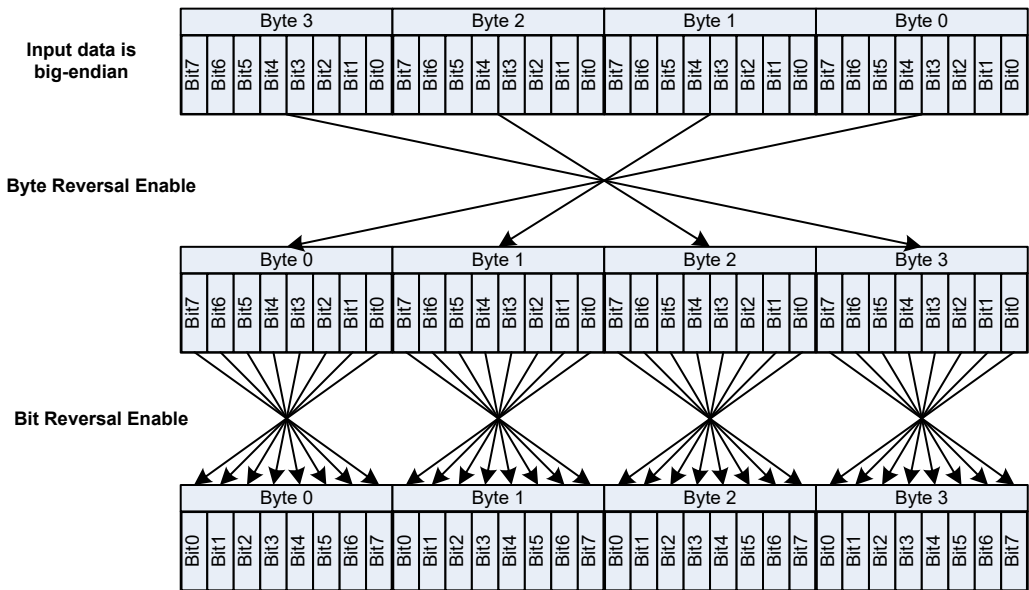


图 213. CRC 数据按字节反序和按位反序范例

与 PDMA 搭配工作

可使用软件触发 PDMA 通道可用于将数据传入 CRC 单元。若需计算大量的数据块，建议先使用 PDMA 来传输其容量允许的数据量，再使用软件写方式来传输剩余的字节。要将数据写入 CRC 单元，应在非递增地址模式下，PDMA 应按字访问的方式将数据从存储器源地址写入 CRC 数据寄存器 CRCDR。通过软件可将剩下任意字节写入 CRC 数据寄存器 CRCDR 且可从 CRC 校验和寄存器 CRCCSR 中读取 CRC 计算结果的值。

寄存器列表

下面表格显示了 CRC 的寄存器和复位值。

表 78. CRC 寄存器列表

寄存器	偏移量	描述	复位值
CRCCR	0x000	CRC 控制寄存器	0x0000_0000
CRCSDR	0x004	CRC 种子值寄存器	0x0000_0000
CRCCSR	0x008	CRC 校验和寄存器	0x0000_0000
CRCDR	0x00C	CRC 数据寄存器	0x0000_0000

寄存器描述

CRC 控制寄存器 – CRCCR

该寄存器定义了相关的 CRC 功能使能控制。

偏移量: 0x000  
复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位	7	6	5	4	3	2	1	0
	SUMCMPL	SUMBYRV	SUMBIRV	DATCMPL	DATBYRV	DATBIRV	POLY	
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[7]	SUMCMPL	对校验和进行反码 0: 除能 1: 使能
[6]	SUMBYRV	对校验和进行字节反序 0: 除能 1: 使能
[5]	SUMBIRV	对校验和进行位反序 0: 除能 1: 使能
[4]	DATCMPL	对数据进行反码 0: 除能 1: 使能
[3]	DATBYRV	对数据进行字节反序 0: 除能 1: 使能
[2]	DATBIRV	对数据进行位反序 0: 除能 1: 使能
[1:0]	POLY	CRC 多项式 00: CRC-CCITT (0x1021) 01: CRC-16 (0x8005) 1X: CRC-32 (0x04C11DB7)

CRC 种子值寄存器 – CRCSDR

该寄存器定义了 CRC 种子值。

偏移量: 0x004  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	SEED								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	SEED								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	SEED								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	SEED								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO

位	字段	描述
[31:0]	SEED	CRC 种子值 根据 CRCCR 寄存器中多项式的设置往此寄存器中填入 16/32-bit 的种子值。

CRC 校验和寄存器 – CRCCSR

此寄存器包含了 CRC 校验和。

偏移量: 0x008  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	CHKSUM								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	CHKSUM								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	CHKSUM								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	CHKSUM								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[31:0]	CHKSUM	CRC 校验和数据 当所有数据已写入 CRCDR 寄存器，根据 CRCCR 寄存器里多项式的设置通过此寄存器获得 16/32-bit 校验和。

CRC 数据寄存器 – CRCDR

该寄存器定义 CRC 数据。

偏移量: 0x00C  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	CRCDATA								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	CRCDATA								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	CRCDATA								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	CRCDATA								
类型 / 复位	WO	0	WO	0	WO	0	WO	0	WO

位	字段	描述
[31:0]	CRCDATA	CRC 输入数据 允许使用字节、半字或字写入方式。根据写入方式选择反码、按字节反序或按位反序操作。



## 29 AES 加密 / 解密接口 (AES)

### 简介

AES 核心提供加密和解密功能。它支持 128 位输入数据，但需要注意的是，若输入数据不足，硬件不会对数据位进行填补，因此用户需在最开始通过软件完成。

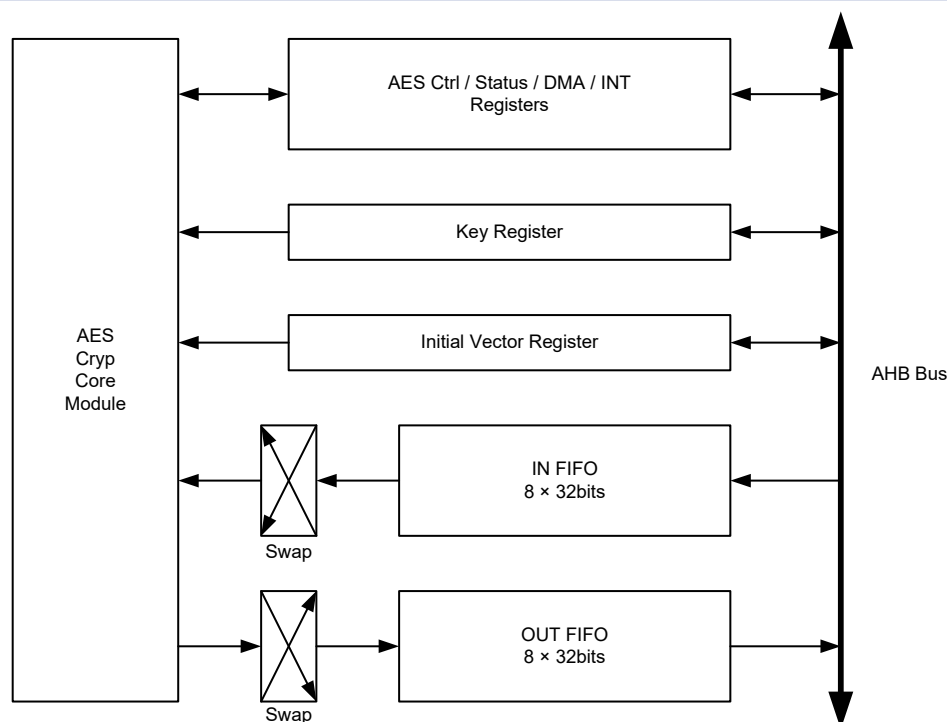


图 214. AES 方框图

### 特性

- 支持 AES 加密 / 解密功能
- 支持 AES ECB / CBC / CTR 三种模式
- 支持 128 位、192 位和 256 位密钥长度
- 支持 4 字长度的初始矢量用于 CBC 和 CTR 模式
- 8 × 32 位 (输入 / 输出 FIFO) 用于 2 个 AES 数据块
- 支持按字的数据交换功能
- 支持 PDMA 接口

## 功能描述

### AES 模式

#### AES 电码本模式 (AES-ECB)

ECB 模式加密, 输入 FIFO 中的 128 位明文数据, 经过字交换处理后, 发送到 AES 核心进行加密操作。AES 核心使用 128 位、192 位或 256 位密钥进行加密处理。加密后产生的密文再次经过字交换处理后, 存入输出 FIFO。

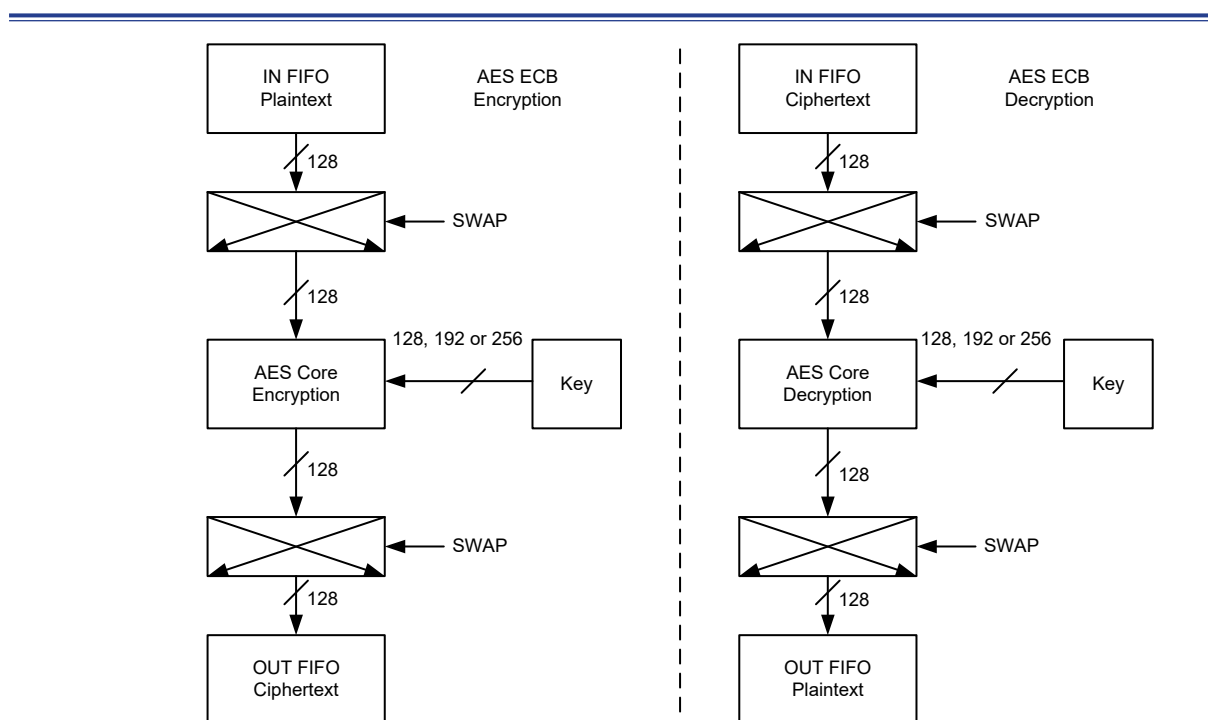


图 215. AES-ECB 模式

### AES 密码分组链接模式 (AES-CBC)

CBC 模式加密，明文切分成若干小段，每一个明文段与上一个的密文段异或运算后，其结果再与密钥进行加密。通过第一次加密运算得到第一个初始矢量起始值。明文经过字交换处理后与初始矢量进行异或运算，其结果与密钥进行加密运算。当生成的密文数据存入输出 FIFO 时，同时初始矢量值会由输出密文数据更新。

CBC 模式解密，每一个明文段经过解密后与上一个的密文段进行异或运算。通过第一次解密运算得到第一个初始矢量起始值。密文经过字交换处理及解密之后，与初始矢量进行异或运算。当异或后的解密数据存至输出 FIFO 时，同时初始矢量值会由解密输入数据更新，用于下一轮的密文解密。

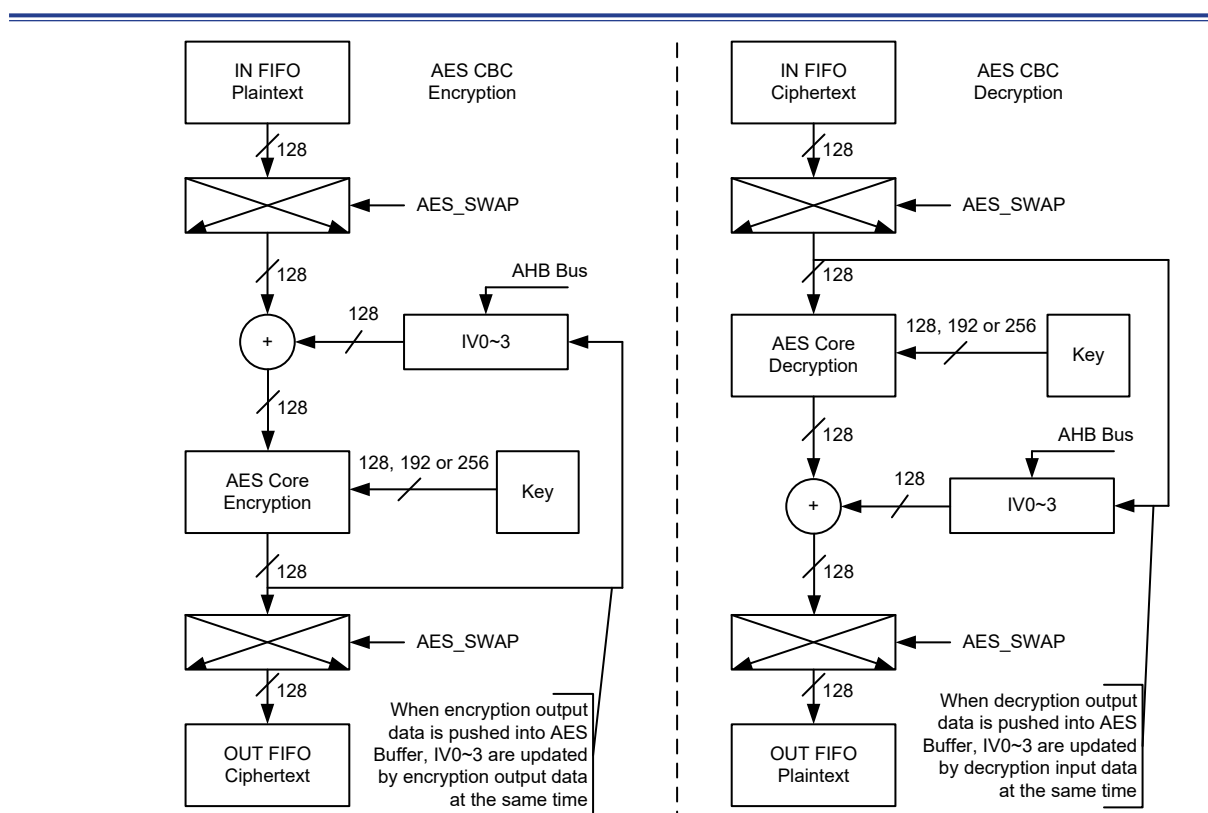


图 216. AES-CBC 模式

### AES 计算器模式 (AES-CTR)

在 CTR 模式中，初始矢量计数器值在增 1 之后，被发送至 AES 核心进行加密得到密文。在加密和解密时，AES 核心使用相同的方向设置。

CTR 模式加密和解密，输入 FIFO 中的数据进行字交换处理后与密文执行异或，异或的结果再经过字交换后发送到输出 FIFO。同时初始矢量值增 1 用于产生下一轮密文。

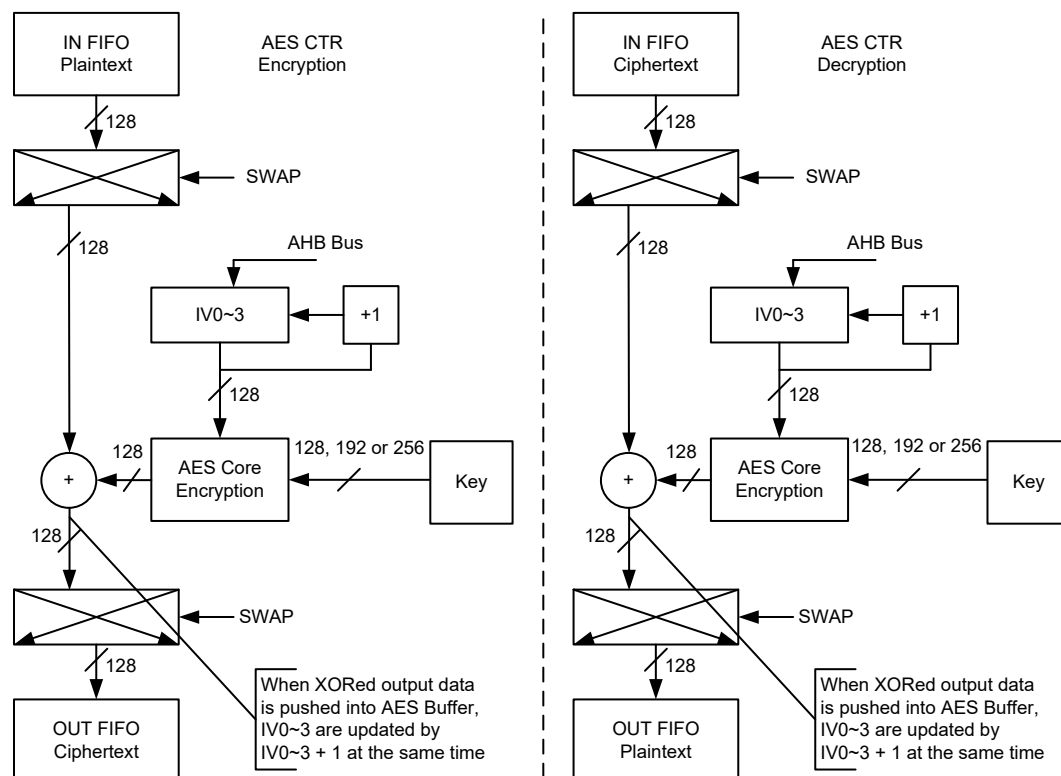


图 217. AES-CTR 模式

AES 状态

提供了五种状态标志位可供用户监测 AES 的工作情况。当输入 FIFO 为空时, IFEMPTY 位被置位; 当输入 FIFO 未滿时, IFNFULL 位会被置位。当输出 FIFO 中有数据时, OFNEMPTY 位被置位; 当输出 FIFO 存滿数据时, OFFULL 位会被置位。当 AES 核心在进行加密, 解密或者当在进行密钥扩展时, BUSY 位被置位。

AES PDMA 接口

AES 支持 32 位 PDMA 数据传输。当输入 FIFO 为空, AES 将发送一个输入 FIFO 请求到 PDMA。当输出 FIFO 已滿, AES 将发送一个输出 FIFO 请求到 PDMA。

AES 中断

当输入 FIFO 小于一个 AES 数据块 ( 4×32 位 ), IFINT 标志位被置位。当输出 FIFO 中有数据时, OFINT 标志位置位。在 AES 使能状态下, 若 IFINT 中断使能, IFINT 被置位或 OFINT 中断使能, OFINT 被置位, 都将产生 AES 中断。

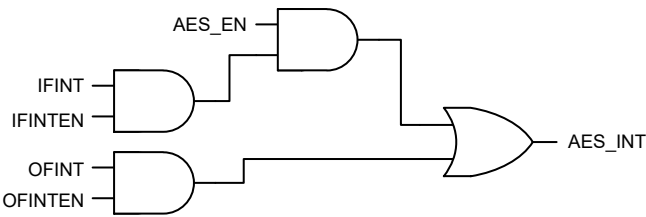


图 218. AES 中断

AES 初始矢量

ECB 模式时, 不会用到初始矢量 (IV0 ~ 3)。CBC 和 CTR 模式时, 初始矢量会在第一个输入数据块时被初始化。完成第一个输入数据块的操作后, 初始矢量通过硬件自动更新作为下一轮数据块的初始矢量。CTR 模式的初始矢量包括随机数、初始矢量和计数器。在每一个 AES 数据块处理完成后, 计数器值加一。

Nonce ( 32 bits )	IV0
Initial Vector ( 64 bits )	IV1, 2
Counter ( 32 bits )	IV3

图 219. CTR 模式初始矢量

AES 字交换

AES 支持字交换功能。交换动作发生在输入 FIFO 和 AES 块数据之间以及输出 FIFO 和 AES 块数据之间。若需要使用字交换功能，需设置 AESCR 寄存器中的 SWAP 位为高以使能字交换功能。

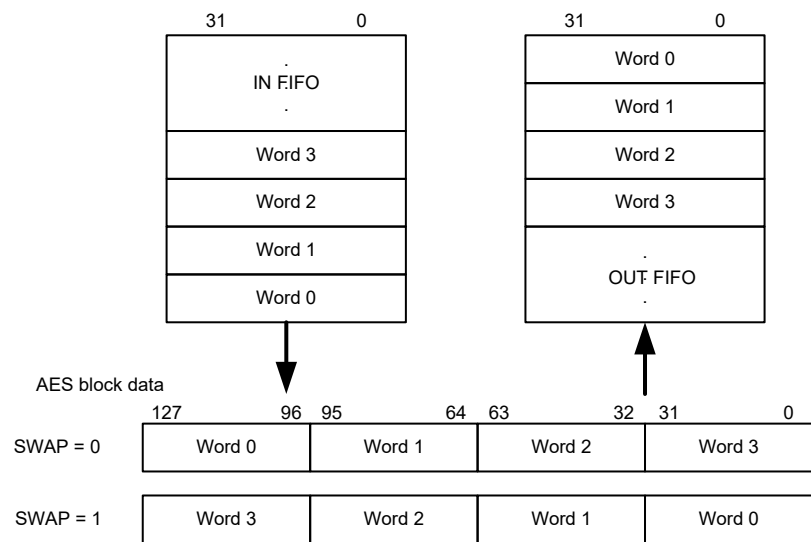


图 220. AES 字交换功能

寄存器列表

下面表格显示了 AES 的寄存器和复位值。

表 79. AES 寄存器列表

寄存器	偏移量	描述	复位值
AESCR	0x000	AES 控制寄存器	0x0000_0000
AESSR	0x004	AES 状态寄存器	0x0000_0003
AESDMAR	0x008	AES DMA 寄存器	0x0000_0000
AESISR	0x00C	AES 中断状态寄存器	0x0000_0000
AESIER	0x010	AES 中断使能寄存器	0x0000_0000
AESDINR	0x014	AES 数据输入寄存器	0x0000_0000
AESDOUTR	0x018	AES 数据输出寄存器	0x0000_0000
AESKEY0	0x01C	AES 密钥寄存器 0	0x0000_0000
AESKEY1	0x020	AES 密钥寄存器 1	0x0000_0000
AESKEY2	0x024	AES 密钥寄存器 2	0x0000_0000
AESKEY3	0x028	AES 密钥寄存器 3	0x0000_0000
AESKEY4	0x02C	AES 密钥寄存器 4	0x0000_0000
AESKEY5	0x030	AES 密钥寄存器 5	0x0000_0000
AESKEY6	0x034	AES 密钥寄存器 6	0x0000_0000
AESKEY7	0x038	AES 密钥寄存器 7	0x0000_0000
AESIVR0	0x03C	AES 初始矢量寄存器 0	0x0000_0000

寄存器	偏移量	描述	复位值
AESIVR1	0x040	AES 初始矢量寄存器 1	0x0000_0000
AESIVR2	0x044	AES 初始矢量寄存器 2	0x0000_0000
AESIVR3	0x048	AES 初始矢量寄存器 3	0x0000_0000

## 寄存器描述

### AES 控制寄存器 – AESCR

该寄存器定义了 AES 控制设置。

偏移量: 0x000

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位					FFLUSH	保留位	SWAP
类型 / 复位					RW 0	RW 0	
7	6	5	4	3	2	1	0
保留位	KEYSIZE		KEYSTART	MODE		DIR	AESEN
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

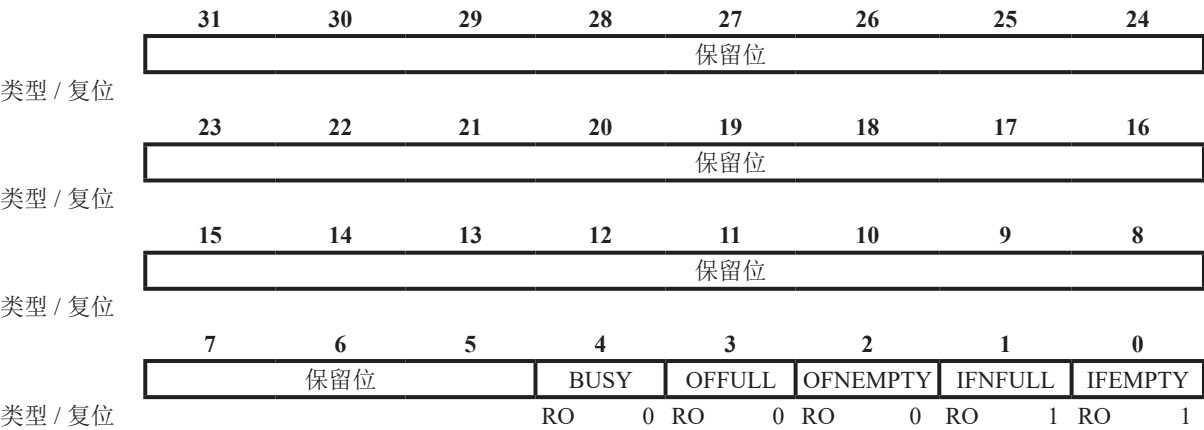
位	字段	描述
[10]	FFLUSH	AES 输入 / 输出 FIFO 清理 0: 无动作 1: 清理 FIFO 此位由硬件自动清零。该位只能在 AES 除能模式下被置位。
[8]	SWAP	AES 数据交换功能 0: 不交换 1: 字交换 该设定用于输入 / 输出 FIFO 数据。
[6:5]	KEYSIZE	AES 密钥长度 00: 128 位 01: 192 位 1x: 256 位
[4]	KEYSTART	AES 密钥开始 0: 密钥未开始 1: 密钥开始 此位由硬件自动清零。该位只有在 AES 使能模式才有效。
[3:2]	MODE	AES 功能模式 00: ECB 模式 01: CBC 模式 1x: CTR 模式

位	字段	描述
[1]	DIR	AES 方向 0: 加密 1: 解密
[0]	AESEN	AES 使能 0: AES 除能 1: AES 使能

AES 状态寄存器 – AESSR

该寄存器指示 AES 状态。

偏移量: 0x004  
复位值: 0x0000\_0003



位	字段	描述
[4]	BUSY	AES 忙标志位 0: AES 空闲 1: AES 忙 当 AES 在进行加密、解密及密钥扩展时, AES 为忙状态。
[3]	OFFULL	输出 FIFO 满标志位 0: 输出 FIFO 未满 1: 输出 FIFO 已满
[2]	OFNEMPTY	输出 FIFO 非空标志位 0: 输出 FIFO 空 1: 输出 FIFO 非空
[1]	IFNFULL	输入 FIFO 未满标志位 0: 输入 FIFO 已满 1: 输入 FIFO 未满
[0]	IFEMPTY	输入 FIFO 空标志位 0: 输入 FIFO 非空 1: 输入 FIFO 空



## AES DMA 寄存器 – AESDMAR

该寄存器用于 DMA 设置。

偏移量: 0x008

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位						OFDMAEN	IFDMAEN
类型 / 复位						RW	0 RW 0

位	字段	描述
[1]	OFDMAEN	输出 FIFO DMA 使能 0: DMA 除能 1: DMA 使能
[0]	IFDMAEN	输入 FIFO DMA 使能 0: DMA 除能 1: DMA 使能

## AES 中断状态寄存器 – AESISR

该寄存器用于储存 AES 的中断状态位。

偏移量: 0x00C

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位						OFINT	IFINT
类型 / 复位						RO 0	RO 0

位	字段	描述
[1]	OFINT	输出 FIFO 中断状态 0: 未发生输出 FIFO 中断 1: 发生输出 FIFO 中断
[0]	IFINT	输入 FIFO 中断状态 0: 未发生输入 FIFO 中断 1: 发生输入 FIFO 中断

## AES 中断使能寄存器 – AESIER

该寄存器用于设置 AES 中断。

偏移量: 0x010

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位						OFINTEN	IFINTEN
类型 / 复位						RW	0 RW 0

位	字段	描述
[1]	OFINTEN	输出 FIFO 中断使能位 0: 中断除能 1: 中断使能
[0]	IFINTEN	输入 FIFO 中断使能位 0: 中断除能 1: 中断使能

## AES 数据输入寄存器 – AESDINR

该寄存器用于定义数据输入值。

偏移量： 0x014

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	DIN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	DIN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	DIN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	DIN								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	DIN	AES 数据输入 0x0000_0000 ~ 0xFFFF_FFFF

## AES 数据输出寄存器 – AESDOUTR

该寄存器用于储存 AES 数据输出。

偏移量： 0x018

复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	DOUT								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	DOUT								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	DOUT								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	DOUT								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[31:0]	DOUT	AES 数据输出 0x0000_0000 ~ 0xFFFF_FFFF

### AES 密钥寄存器 n – AESKEYRn, n = 0 ~ 7

该寄存器用于定义密钥数据字 n。

偏移量: 0x01C ~ 0x038

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	KeyData								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	KeyData								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	KeyData								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	KeyData								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	KeyData	密钥数据 0x0000_0000 ~ 0xFFFF_FFFF

### AES 初始矢量寄存器 n – AESIVRn, n = 0 ~ 3

该寄存器用于定义初始矢量数据字 n。

偏移量: 0x03C ~ 0x048

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	IVData								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	IVData								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	IVData								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	IVData								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	IVData	初始矢量数据 0x0000_0000 ~ 0xFFFF_FFFF

## 30 SDIO 主机控制器 (SDIO)

### 简介

SDIO 主机控制器支持多媒体卡 (MMC)、SD 存储卡和 SD I/O 卡。SDIO 是由一条时钟线、一条 CMD 线和 DAT0 ~ DAT3 4 条数据线组成的 6-pin 高级通信接口。

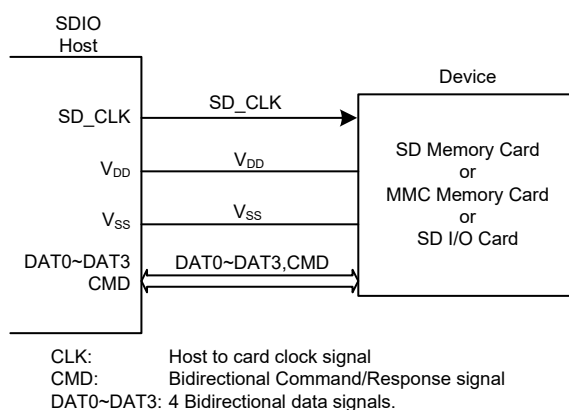


图 221. SDIO 总线拓扑

### 特性

- 支持两种不同的数据总线模式：1-bit (默认) 和 4-bit
- 支持两种不同的速率模式：正常速率模式 (默认) 和高速模式
- SD 时钟频率能高达系统频率
- 不支持 SPI 模式和 MMC 流模式

### 功能描述

SDIO 包含一个命令寄存器、参数寄存器、响应寄存器、数据缓冲器、超时计数器和错误检测逻辑。SDIO 支持单数据块和多数数据块搭配 PDMA 的传输，减少大数据传输过程中的干扰。

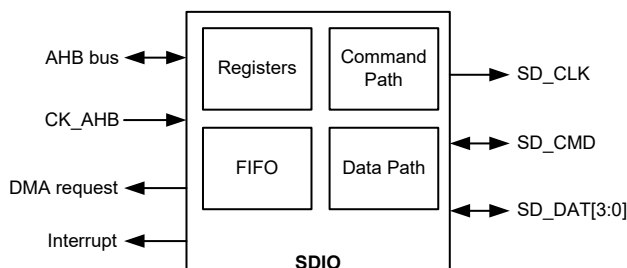


图 222. SDIO 方框图

## SD 时钟

SDIO 控制器驱动 SD\_CLK 时钟并将时钟传送给卡。当 CK\_AHB 工作在 96 MHz 时, SD\_CLK 频率可为 48 MHz (高速模式) 和 24 MHz (正常速率模式)。在正常速率模式下, CMD 和 DAT0 ~ DAT3 线在 SD\_CLK 的下降沿时发生变化, 在 SD\_CLK 的上升沿时锁存。在高速模式下, CMD 和 DAT0 ~ DAT3 线在 SD\_CLK 上升沿时发生改变和锁存。

$$SD\_CLK = CK\_AHB / (CLKPRE + 1), CLKPRE = 0 \sim 255$$

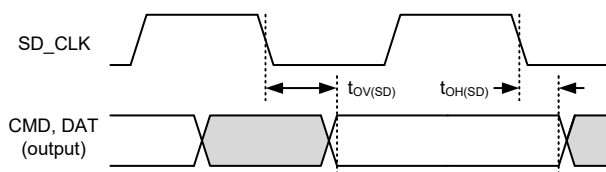


图 223. 正常速率时序

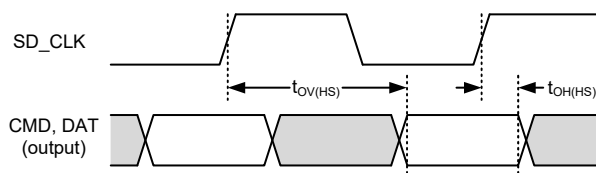


图 224. 高速时序

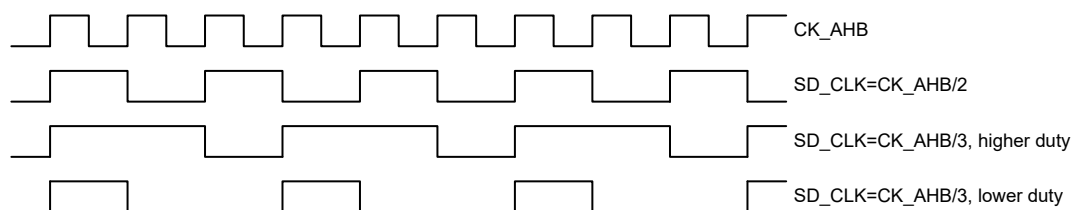


图 225. SD\_CLK 占空比周期

## SD 协议

SD 通信是通过 SD 总线上的基本操作，即命令、响应和由一个 START 信号初始化传输并通过发送一个 STOP 信号来中止传输的数据位流。CMD 线是高位优先传送的。

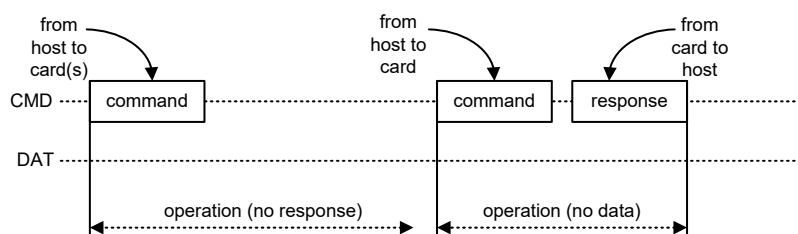


图 226. “无响应”和“无数据”操作

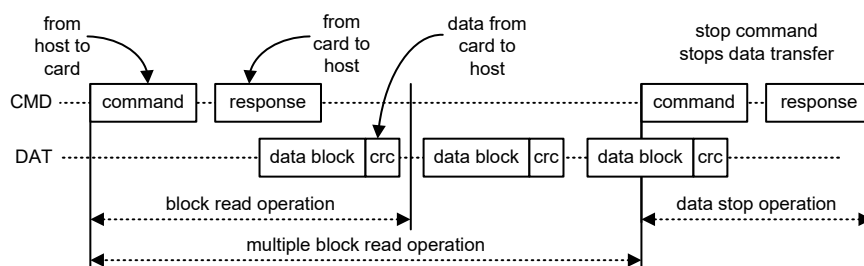


图 227. 多数据块读操作

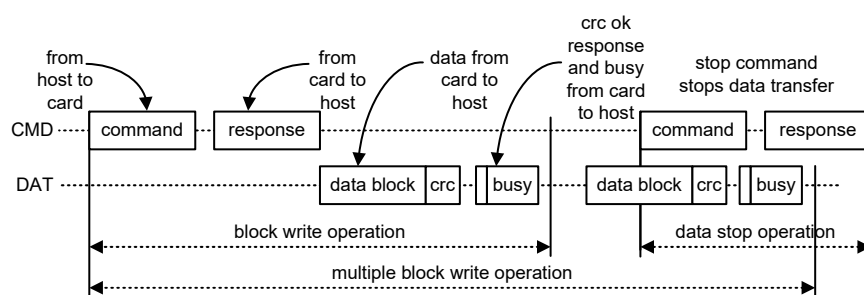


图 228. 多数据块写操作



命令

命令的长度固定为 48 位。每个命令以起始位 (总是为 0 ) 开始, 以结束位 (总是为 1 ) 完成。每个命令都受 CRC 位段检测错误的保护, 可以检测传输错误并重新操作。

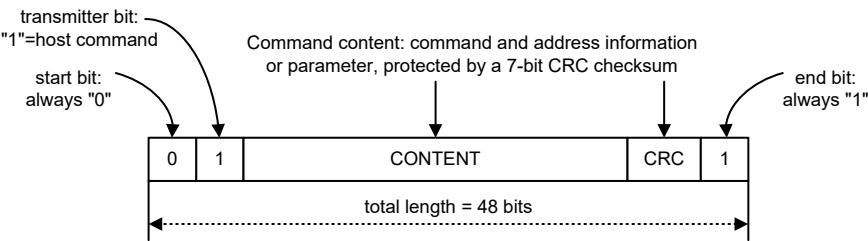


图 229. 命令格式

表 80. 命令格式

位位置	[47]	[46]	[45:40]	[39:8]	[7:1]	[0]
宽度	1	1	6	32	7	1
值	"0"	"1"	X	X	X	"1"
描述	起始位	传输位	命令索引	参数	CRC7	结束位

响应

响应内容决定了响应的长度是 48 位或 136 位。

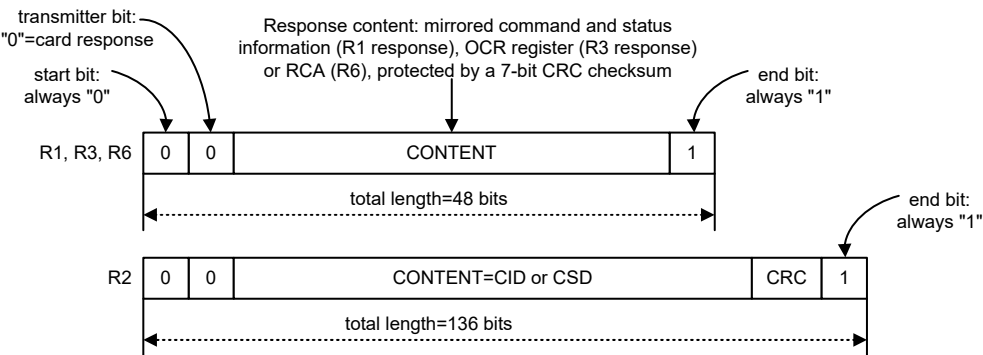


图 230. 响应格式

表 81. R1 响应格式

位位置	[47]	[46]	[45:40]	[39:8]	[7:1]	[0]
宽度	1	1	6	32	7	1
值	“0”	“0”	X	X	X	“1”
描述	起始位	传输位	命令索引	卡状态	CRC7	结束位

表 82. R2 响应格式

位位置	[135]	[134]	[133:128]	[127:8]	[7:1]	[0]
宽度	1	1	6	120	7	1
值	“0”	“0”	“111111”	X	X	“1”
描述	起始位	传输位	保留位	CID 或 CSD 寄存器	CRC7	结束位

表 83. R3 响应格式

位位置	[47]	[46]	[45:40]	[39:8]	[7:1]	[0]
宽度	1	1	6	32	7	1
值	“0”	“0”	“111111”	X	“111111”	“1”
描述	起始位	传输位	保留位	OCR 寄存器	保留位	结束位

表 84. R6 响应格式

位位置	[47]	[46]	[45:40]	[39:24]	[23:8]	[7:1]	[0]
宽度	1	1	6	16	16	7	1
值	“0”	“0”	“000011”	X	X	X	“1”
描述	起始位	传输位	命令索引	新 RCA	卡状态	CRC7	结束位

表 85. R7 响应格式

位位置	[47]	[46]	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	[0]
宽度	1	1	6	20	4	8	7	1
值	“0”	“0”	“001000”	“00000”	X	X	X	“1”
描述	起始位	传输位	命令索引	保留位	可接受电压	回传	CRC7	结束位

数据

具有通用数据和宽数据两种数据格式。通用数据宽度为 8-bit，低位优先传输的格式。然而，有个别字节是高位优先传输。数据块的 CRC 保护算法为 16-bit CCITT 多项式。

宽数据是从最高位开始移位。当选用宽总线，每个时钟周期四条数据线上将传输 4 位数据。每条 DAT 线都有起始位、结束位和 CRC 位段。CRC 位段分别计算和检测每条 DAT 线。CRC 状态响应和忙标志位将仅通过 DAT0 线从 SD 设备传输到主控制器。

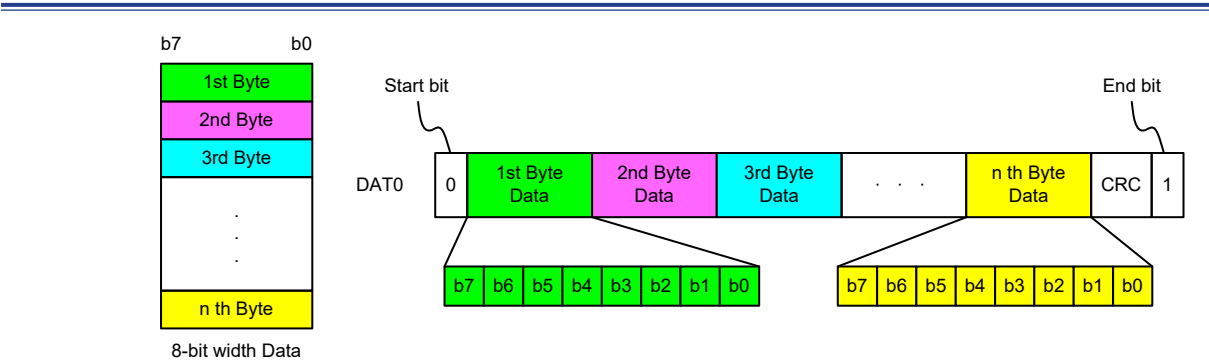


图 231. 标准总线的通用数据格式 – 仅使用 DAT0

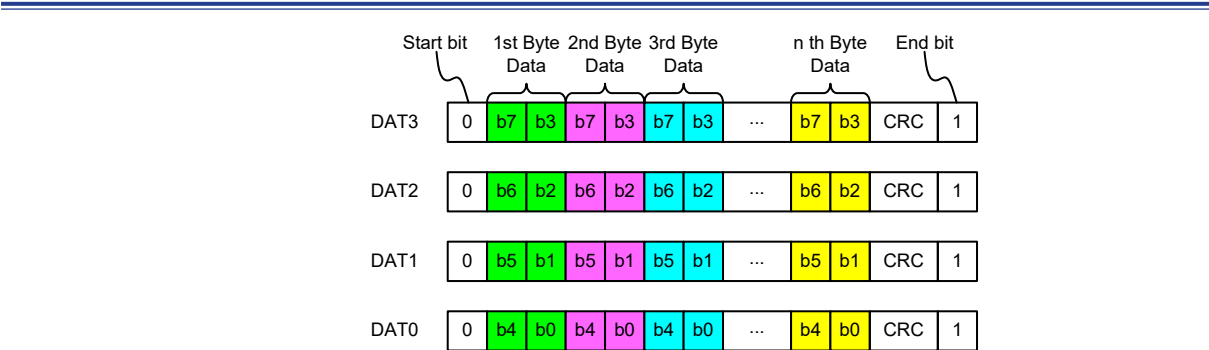


图 232. 宽总线的通用数据格式 – 使用 DAT0 ~ DAT3

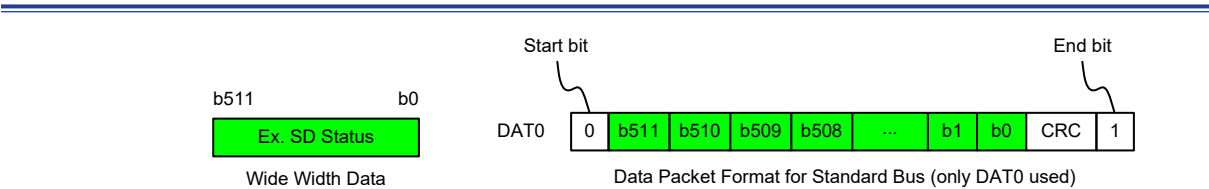


图 233. 标准总线的宽数据格式 – 仅使用 DAT0

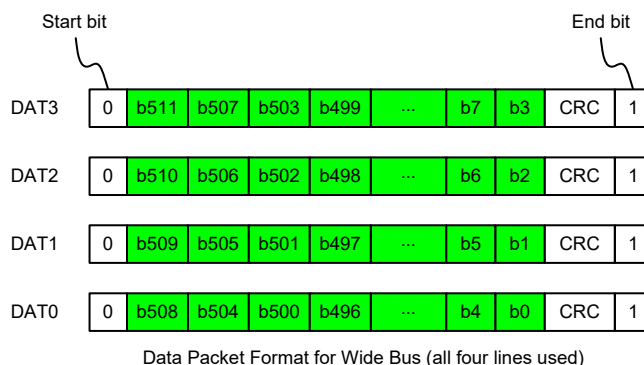


图 234. 宽总线的宽数据格式 – 使用 DAT0 ~ DAT3

## 缓冲器状态

SDIO 包含了一个  $8 \times 32\text{-bit}$  数据缓冲器，该缓冲器由数据读操作和写操作共用。缓冲器深度可通过读取当前状态寄存器的 BLSTA 位段来检验，通过存取数据端口寄存器可对缓冲器中数据进行读取或写入。

一旦缓冲器有  $4 \times 32\text{-bit}$  数据被读取或有  $4 \times 32\text{-bit}$  数据写入到缓冲器，状态寄存器的 BHSTA 位将置位。当 FIFO 已满，状态寄存器的 BFSTA 位将置位，当 FIFO 为空，状态寄存器的 BESTA 位将置位。所有的缓冲器状态标志位的使能可通过状态使能寄存器的相关位设置。

## 中断

当事件发生时，如果中断使能寄存器中的相关位使能，SDIO 可发送一个缓冲器半满、缓冲器全满或缓冲器为空的中断给 NVIC。

如果状态使能寄存器中的相关位使能，通过读取状态寄存器的 CISTA 位能够检查设备卡中断请求是否发送到主控制器。

## DMA 请求

如果传输模式寄存器中的 DMAEN 位置位，若在写操作时有  $4 \times 32\text{-bit}$  数据写入到缓冲器或读操作时缓冲器有  $4 \times 32\text{-bit}$  数据被读取，SDIO 将发送一个 DMA 请求到 PDMA。

表 86. SDIO 命令寄存器字段和值

命令	缩写	字段和值	描述
CMD0	GO_IDLE_STATE	RESP_TYPE = 0 DAT_PRESENT = 0	无响应操作
CMD2	ALL_SEND_CID	RESP_TYPE = 1 DAT_PRESENT = 0	长响应操作
CMD9	SEND_CSD		
CMD10	SEND_CID		
CMD1	SEND_OP_COND	RESP_TYPE = 2 DAT_PRESENT = 0	短响应操作
CMD3	SEND_RELATIVE_ADDR		
CMD5 <sup>(注)</sup>	IO_SEND_OP_COND		
CMD7	SELECT/DESELECT_CARD		
CMD8	SEND_IF_COND		
CMD12	STOP_TRANSMISSION		
CMD13	SEND_STATUS		
CMD16	SET_BLOCKLEN		
CMD32	ERASE_WR_BLK_START		
CMD33	ERASE_WR_BLK_END		
CMD38	ERASE		
CMD52 <sup>(注)</sup>	IO_RW_DIRECT		
CMD55	APP_CMD		
ACMD6	SET_BUS_WIDTH		
ACMD23	SET_WR_BLK_ERASE_COUNT		
ACMD41	SD_SEND_OP_COND		
CMD6	SWITCH_FUNC	RESP_TYPE = 2 DAT_PRESENT = 1	数据块操作
CMD17	READ_SINGLE_BLOCK		
CMD18	READ_MULTIPLE_BLOCK		
CMD24	WRITE_SINGLE_BLOCK		
CMD25	WRITE_MULTIPLE_BLOCK		
CMD53 <sup>(注)</sup>	IO_RW_EXTEND		
ACMD13	SD_STATUS		
ACMD51	SEND_SCR		

注：SD I/O 卡特定命令。

## 寄存器列表

下表显示了 SDIO 寄存器及其复位值。

**表 87. SDIO 寄存器列表**

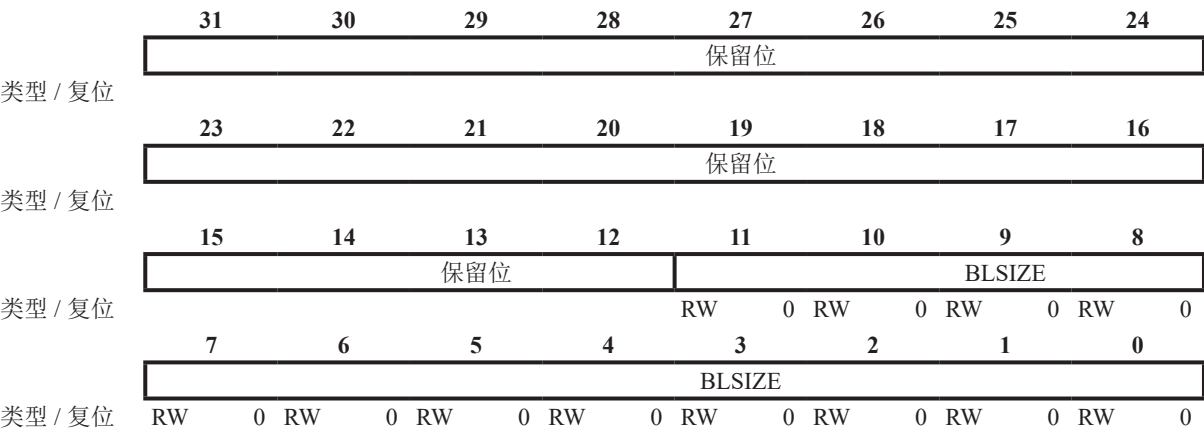
寄存器	偏移量	描述	复位值
BLSIZE	0x000	块大小寄存器	0x0000_0000
BLCNT	0x004	块计数寄存器	0x0000_0000
ARG	0x008	参数寄存器	0x0000_0000
TMR	0x00C	传输模式寄存器	0x0000_0000
CMD	0x010	命令寄存器	0x0000_0000
RESP0	0x014	响应寄存器 0	0x0000_0000
RESP1	0x018	响应寄存器 1	0x0000_0000
RESP2	0x01C	响应寄存器 2	0x0000_0000
RESP3	0x020	响应寄存器 3	0x0000_0000
DR	0x024	数据端口寄存器	0xFFFF_FFFF
PSR	0x028	当前状态寄存器	0x0000_0800
CR	0x02C	控制寄存器	0x0000_0000
CLKCR	0x038	时钟控制寄存器	0x0000_0000
TMOCR	0x03C	超时控制寄存器	0x00FF_FFFF
SWRST	0x040	软件复位寄存器	0x0000_0000
SR	0x044	状态寄存器	0x0000_0000
SER	0x048	状态使能寄存器	0x0000_0000
IER	0x04C	中断使能寄存器	0x0000_0000

寄存器描述

块大小寄存器 – BLSIZE

该寄存器用于配置一个数据块的字节数。

偏移量: 0x000  
复位值: 0x0000\_0000

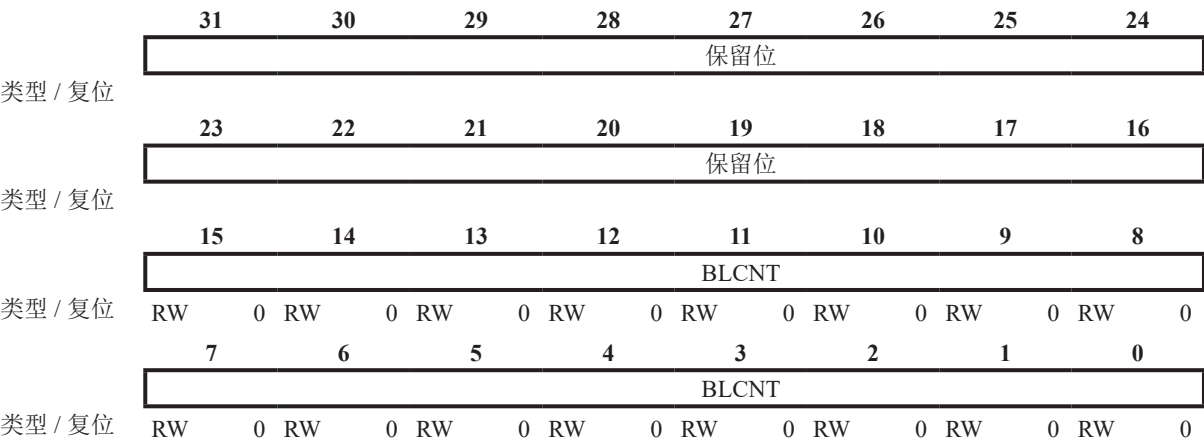


位	字段	描述
[11:0]	BLSIZE	传输块大小 000h: 无数据传输 001h: 1-byte 002h: 2-byte ... 1FFh: 511-byte 200h: 512-byte ... 800h: 2048-byte 801h ~ FFFh: 保留 BLSIZE 表示 CMD17、CMD18、CMD24、CMD25 和 CMD53 命令的数据传输块大小。

块计数寄存器 – BLCNT

该寄存器用于配置数据块的计数。

偏移量: 0x004  
复位值: 0x0000\_0000



位	字段	描述
[15:0]	BLCNT	当前传输块计数 0000h: 停止计数 0001h: 1 块 0002h: 2 块 ... FFFFh: 65535 块 当传输模式寄存器中的块计数使能位置位时该寄存器使能，且该寄存器只在多块传输中有效。



参数寄存器 – ARG

该寄存器包含了 SD 命令参数。

偏移量： 0x008  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	ARG								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	ARG								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	ARG								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	ARG								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

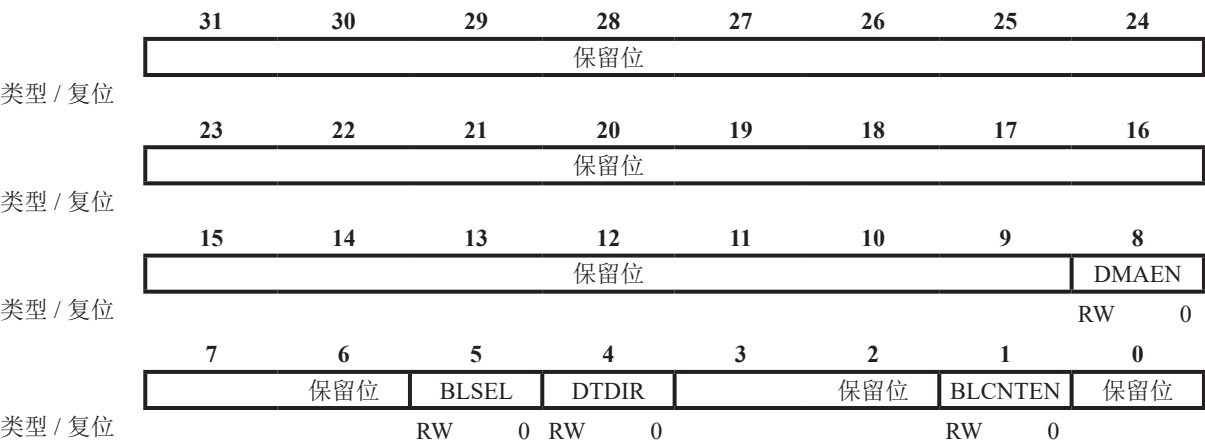
位	字段	描述
[31:0]	ARG	命令参数 该寄存器包含了命令参数，将命令参数作为命令信息的一部分发送到 SD 设备。

传输模式寄存器 – TMR

该寄存器用于控制数据传输操作。

偏移量： 0x00C

复位值： 0x0000\_0000



位	字段	描述
[8]	DMAEN	DMA 使能 0: 除能 1: 使能
[5]	BLSEL	多块 / 单块选择位 0: 单块 1: 多块
[4]	DTDIR	数据传输方向选择位 0: 写 – 主控制器到卡 1: 读 – 卡到主控制器
[1]	BLCNTEN	块计数使能 0: 除能 1: 使能

BLSEL	BLCNTEN	BLCNT	功能
0	无关	无关	单块传输
1	0	无关	无限传输
1	1	不为 0	多块传输
1	1	为 0	停止多块传输

命令寄存器 – CMD

写该寄存器触发 SD 命令产生。

偏移量: 0x010  
复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位							
类型 / 复位	23	22	21	20	19	18	17	16
	保留位							
类型 / 复位	15	14	13	12	11	10	9	8
	保留位		CMDIDX					
类型 / 复位	RW		0	RW	0	RW	0	RW
	0		RW	0	RW	0	RW	0
类型 / 复位	7	6	5	4	3	2	1	0
	CMDTYP		DPSEL	CIDXCEN	CCRCCEN	保留位	RTSEL	
类型 / 复位	RW		0	RW	0	RW	0	RW
	0		RW	0	RW	0	RW	0

位	字段	描述
[13:8]	CMDIDX	命令索引 CMDIDX 须作为命令信息 (CMD0 ~ 63, ACMD0 ~ 63) 中的一部分发送到 SD 设备发送到命令。
[7:6]	CMDTYP	命令类型 00b: 正常 – 其它命令 01b: 暂停 – 在 CCCR 中, 通过 CMD52 命令写入 “总线暂停” 10b: 重新开始 – 在 CCCR 中, 通过 CMD52 命令写入 “功能选择” 11b: 中止 – 在 CCCR 中, 通过 CMD12 和 CMD52 命令写入 “中止 I/O”
[5]	DPSEL	数据当前状态 0: 无数据 1: 有数据 该位设置为 1 表明数据存在, 通过 DAT 线传输。
[4]	CIDXCEN	命令索引检查使能 0: 除能 1: 使能 如果该位设置为 1, SDIO 主机控制器应检查响应的索引字段确定其是否与命令索引值相同。
[3]	CCRCCEN	命令 CRC 校验使能 0: 除能 1: 使能 如果该位设置为 1, 在响应时, SDIO 主控制器应检查 CRC 字段。

位	字段	描述																								
[1:0]	RTSEL	响应类型选择 00b: 无响应 01b: 响应长度 136 10b: 响应长度 48 11b: 响应长度 48 – 响应后检查忙碌																								
<table><tr><th>响应类型</th><th>索引检查使能</th><th>CRC 校验使能</th><th>响应类型名</th></tr><tr><td>00</td><td>0</td><td>0</td><td>无响应</td></tr><tr><td>01</td><td>0</td><td>1</td><td>R2</td></tr><tr><td>10</td><td>0</td><td>0</td><td>R3, R4</td></tr><tr><td>10</td><td>1</td><td>1</td><td>R1, R5, R6, R7</td></tr><tr><td>11</td><td>1</td><td>1</td><td>R1b, R5b</td></tr></table>			响应类型	索引检查使能	CRC 校验使能	响应类型名	00	0	0	无响应	01	0	1	R2	10	0	0	R3, R4	10	1	1	R1, R5, R6, R7	11	1	1	R1b, R5b
响应类型	索引检查使能	CRC 校验使能	响应类型名																							
00	0	0	无响应																							
01	0	1	R2																							
10	0	0	R3, R4																							
10	1	1	R1, R5, R6, R7																							
11	1	1	R1b, R5b																							

响应寄存器 n – RESPn, n = 0 ~ 3

这些寄存器用于存储 SD 设备的响应。

偏移量: 0x014 (0), 0x18 (1), 0x1C (2), 0x20(3)  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	RESPn							
类型 / 复位	RO	0	RO	0	RO	0	RO	0
	23	22	21	20	19	18	17	16
	RESPn							
类型 / 复位	RO	0	RO	0	RO	0	RO	0
	15	14	13	12	11	10	9	8
	RESPn							
类型 / 复位	RO	0	RO	0	RO	0	RO	0
	7	6	5	4	3	2	1	0
	RESPn							
类型 / 复位	RO	0	RO	0	RO	0	RO	0

位	字段	描述															
[31:0]	RESPn	命令响应															
		<table><tr><th>寄存器</th><th>长响应</th><th>短响应</th></tr><tr><td>RESP0</td><td>命令响应 [127:96]</td><td>命令响应 [31:0]</td></tr><tr><td>RESP1</td><td>命令响应 [95:64]</td><td>—</td></tr><tr><td>RESP2</td><td>命令响应 [63:32]</td><td>—</td></tr><tr><td>RESP3</td><td>命令响应 [31:0]</td><td>—</td></tr></table>	寄存器	长响应	短响应	RESP0	命令响应 [127:96]	命令响应 [31:0]	RESP1	命令响应 [95:64]	—	RESP2	命令响应 [63:32]	—	RESP3	命令响应 [31:0]	—
寄存器	长响应	短响应															
RESP0	命令响应 [127:96]	命令响应 [31:0]															
RESP1	命令响应 [95:64]	—															
RESP2	命令响应 [63:32]	—															
RESP3	命令响应 [31:0]	—															

数据端口寄存器 – DR

该 32-bit 数据端口寄存器用于访问内部缓冲器。

偏移量: 0x024  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	DR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	DR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	DR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	DR								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	DR	缓冲器数据 通过该 32-bit 数据端口寄存器访问主控制器 8 × 32-bit 的缓冲器。

## 当前状态寄存器 – PSR

该寄存器是只读寄存器，包含了主控制器的当前状态。

偏移量： 0x028

复位值： 0x0000\_0800

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
BLSTA				BURAW	BORAW	保留位	
RO	0	RO	0	RO	0	RO	0
15	14	13	12	11	10	9	8
保留位				BERAW	BFRAW	RTASTA	WTASTA
				RO	1	RO	0
7	6	5	4	3	2	1	0
保留位					DLASTA	CIDSTA	CICSTA
					RO	0	RO

位	字段	描述
[23:20]	BLSTA	缓冲器深度状态 0000: 缓冲器无数据 0001: 缓冲器内有 1 × 32-bit 数据 ... 1000: 缓冲器内有 8 × 32-bit 数据 其它值: 保留
[19]	BURAW	缓冲器下溢原状态 0: 缓冲器未下溢 1: 缓冲器下溢 如果该状态位设置为 1, 表示数据缓冲器发生下溢。软件复位 DPSM 可清零此位。
[18]	BORAW	缓冲器上溢原状态 0: 缓冲器未上溢 1: 缓冲器上溢 如果该状态位设置为 1, 表示数据缓冲器发生上溢。软件复位 DPSM 可清零此位。
[11]	BERAW	缓冲器为空原状态 0: 缓冲器不为空 1: 缓冲器为空 如果该状态位设置为 1, 表示数据缓冲器为空。在这种情况下, 任意读操作将使缓冲器下溢状态位置位。
[10]	BFRAW	缓冲器已满原状态 0: 缓冲器未满 1: 缓冲器已满 如果该状态位设置为 1, 表示数据缓冲器已满。在这种情况下, 任意写操作将使缓冲器上溢状态位置位。
[9]	RTASTA	读传输有效状态 0: 无有效数据 1: 传输数据 该状态位用于检测读传输完成。

位	字段	描述
[8]	WTASTA	写传输有效状态 0: 无有效数据 1: 传输数据 该状态位用于表示写传输是否有效。若该位为 0，表示主控制器不存在有效写入的数据。
[2]	DLASTA	DAT 线有效状态 0: DAT 线无效 1: DAT 线有效 该位表示在 SD 总线中是否有任意一条 DAT 线在使用。
[1]	CIDSTA	命令禁止 (DAT) 状态 0: 可通过 DAT 线发送一个命令 1: 不可通过 DAT 线发送一个命令 如果 DAT 线有效或读传输有效置位，将产生该状态位。若该位为 0，表示主控制器可发送下一条 SD 命令。
[0]	CICSTA	命令禁止 (CMD) 状态 0: 仅可通过 CMD 线发送命令 1: 不可发送命令 如果该位为 0，表示 CMD 线未使用，主控制器可通过 CMD 线发送 SD 命令。

控制寄存器 – CR

该寄存器用于定义数据速率模式和传输宽度。

偏移量: 0x02C  
复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位							
类型 / 复位							
7	6	5	4	3	2	1	0
保留位				HSMEN		DATWID	保留位
类型 / 复位				RW		0	RW 0

位	字段	描述
[2]	HSMEN	高速模式使能 0: 正常速率模式 1: 高速模式 如果该位设置为 1，在 SD 时钟的上升沿主控制器输出，CMD 和 DAT 线发生改变或是锁存。
[1]	DATWID	数据传输宽度 0: 1-bit 模式 1: 4-bit 模式 该位选择主控制器的数据宽度。

时钟控制寄存器 – CLKCR

该寄存器用于配置 SD 时钟频率和使能控制。

偏移量: 0x038  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
类型 / 复位	保留位								
	23	22	21	20	19	18	17	16	
类型 / 复位	保留位								
	15	14	13	12	11	10	9	8	
类型 / 复位	CLKPRE								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
类型 / 复位	保留位				CLKDUTY	CLKEN	CLKSPEN	CLKSPLV	
					RW	0	RW	0	RW

位	字段	描述
[15:8]	CLKPRE	SD 时钟预分频器 00h: SD_CLK = CK_AHB 01h: SD_CLK = CK_AHB / 2 02h: SD_CLK = CK_AHB / 3 ... FFh: SD_CLK = CK_AHB / 256
[3]	CLKDUTY	SD 时钟占空比周期 0: 更高 1: 更低 如果 SD_CLK 预分频率不为 2 的倍数，此位设置为 1 将获得更低的占空比周期。
[2]	CLKEN	SD 时钟使能 0: 除能 1: 使能 当该位设置为 0，主控制应停止 SD_CLK。
[1]	CLKSPEN	SD 时钟停止使能 0: 除能 1: 使能 如果该位设置为 1，若是 SD 总线空闲，SD_CLK 将保持为高或低电平。
[0]	CLKSPLV	SD 时钟停止电平 0: 低电平 1: 高电平 如果 SD 时钟停止功能使能，当该位设置为 1 时，SD_CLK 将保持为高电平，当该位设置为 0 时，SD_CLK 将保持为低电平。



## 超时控制寄存器 – TMOCR

该寄存器用于定义数据超时计数器值。

偏移量: 0x03C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	TMOVAL								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	TMOVAL								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	TMOVAL								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[23:0]	TMOVAL	数据超时计数器值 该值决定了 DAT 线超时时间隙。

软件复位寄存器 – SWRST

对该寄存器的每位写 1 将产生复位脉冲。完成软件复位需要一定的时间，SD 主控制器需确认这些位为零。

偏移量:	0x040
复位值:	0x0000_0000
	3130292827262524
类型 / 复位	保留位
	2322212019181716
类型 / 复位	保留位
	15141312111098
类型 / 复位	保留位
	76543210
类型 / 复位	保留位RSTDATRSTCMDRSTALL
	RW0RW0RW0

位	字段	描述
[2]	RSTDAT	DAT 线软件复位 0: 无复位请求 1: 复位请求 如果该位设置为 1，仅数据电路部分被复位。
[1]	RSTCMD	CMD 线软件复位 0: 无复位请求 1: 复位请求 如果该位设置为 1，仅命令电路部分被复位。
[0]	RSTALL	总的软件复位 0: 无复位请求 1: 复位请求 该复位影响整个主控制器。

## 状态寄存器 – SR

该寄存器包含了主控制器的状态位。

偏移量: 0x044

复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位					CICESTA	CIDESTA	CISTA
						WC 0	WC 0	RO 0
类型 / 复位	23	22	21	20	19	18	17	16
	保留位	DEESTA	DCESTA	DTESTA	CIESTA	CEESTA	CCESTA	CTESTA
		WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0
类型 / 复位	15	14	13	12	11	10	9	8
	ERRSTA	保留位						
	RO 0							
类型 / 复位	7	6	5	4	3	2	1	0
	BESTA	BFSTA	BHSTA	BUSTA	BOSTA	保留位	TCSTA	CCSTA
	WC 0	WC 0	WC 0	WC 0	WC 0		WC 0	WC 0

位	字段	描述
[26]	CICESTA	命令禁止 (CMD) 错误状态位 0: 无错误 1: 错误 命令禁止 (CMD) 为高时, 当命令寄存器或参数寄存器进行写操作时此位将置位。
[25]	CIDESTA	命令禁止 (DAT) 错误状态位 0: 无错误 1: 错误 命令禁止 (DAT) 为高时, 当命令寄存器 (使用 DAT 线)、块大小寄存器和传输模式寄存器进行写操作时此位将置位。
[24]	CISTA	卡中断状态位 0: 未发生卡中断 1: 发生卡中断 当主控制器被采样时, SD 设备将发出一个卡中断, 该位将被置位。
[22]	DEESTA	数据结束位错误标志位 0: 无错误 1: 错误 通过 DAT 线读数据的结束位或 CRC 状态的结束位检测到 0, 将发生数据结束位错误。
[21]	DCESTA	数据 CRC 校验错误状态位 0: 无错误 1: 错误 通过 DAT 线读数据的传输过程中检测到 CRC 校验错误或写 CRC 状态有除 “010” 以外的值, 将发生数据 CRC 校验错误。
[20]	DTESTA	数据超时错误状态位 0: 无错误 1: 超时 当检测到以下超时条件中任意一个, 该位将被置位。 (1) R1b 或 R5b 类型忙碌超时 (2) 写 CRC 状态后忙碌超时 (3) 写 CRC 状态超时 (4) 读数据超时

位	字段	描述
[19]	CIESTA	命令索引错误状态位 0: 无错误 1: 错误 在命令响应中, 如果命令索引错误发生, 该位将被置位。
[18]	CEESTA	命令结束位错误状态位 0: 无错误 1: 结束位错误产生 当检测到命令响应的结束位为 0 时, 该位将被置位。
[17]	CCESTA	命令 CRC 错误状态位 0: 无错误 1: CRC 错误发生 当检测到以下超时条件中任意一个, 该位将被置位。 (1) 命令响应中检测到 CRC 错误 (2) 在发送一个命令时, CMD 线冲突发生
[16]	CTESTA	命令超时错误状态位 0: 无错误 1: 超时 仅在命令结束位后 64 个 SD_CLK 内无响应返回, 该位将被置位。
[15]	ERRSTA	错误发生 0: 无错误 1: 错误 如果状态寄存器中的任意错误状态位被置位, 该位将被置位。
[7]	BESTA	缓冲器为空状态位 0: 缓冲器不为空 1: 缓冲器为空 该位通过写 1 清零。
[6]	BFSTA	缓冲器已满状态位 0: 缓冲器未满 1: 缓冲器已满 该位通过写 1 清零。
[5]	BHSTA	缓冲器半满状态位 0: 缓冲器未半满 / 空 1: 缓冲器半满 / 空 该位通过写 1 清零。
[4]	BUSTA	缓冲器下溢状态位 0: 缓冲器未下溢 1: 缓冲器下溢 该位通过写 1 清零。
[3]	BOSTA	缓冲器上溢状态位 0: 缓冲器未上溢 1: 缓冲器上溢 该位通过写 1 清零。
[1]	TCSTA	传输完成状态位 0: 未完成 1: 命令执行完成 当一个带忙碌检查的读 / 写传输命令完成时, 该位将被置位。该位通过写 1 清零。
[0]	CCSTA	命令完成状态位 0: 无命令完成 1: 命令完成 命令响应的结束位获得时, 该位将被置位。该位通过写 1 清零。命令超时错误比命令完成具有更高的优先级。

## 状态使能寄存器 – SER

该寄存器用于使能定义在状态寄存器中的状态位。

偏移量: 0x048

复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位					CICESEN	CIDENSEN	CISEN
						RW	0	RW
类型 / 复位	23	22	21	20	19	18	17	16
	保留位	DEESEN	DCESEN	DTESEN	CIESEN	CEESEN	CCESSEN	CTESEN
		RW	0	RW	0	RW	0	RW
类型 / 复位	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位	7	6	5	4	3	2	1	0
	BESSEN	BFSEN	BHSEN	BUSEN	BOSEN	保留位	TCESEN	CCSEN
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[26]	CICESEN	命令禁止 (CMD) 错误状态位使能 0: 屏蔽 1: 使能
[25]	CIDENSEN	命令禁止 (DAT) 错误状态位使能 0: 屏蔽 1: 使能
[24]	CISEN	卡中断状态位使能 0: 屏蔽 1: 使能
[22]	DEESEN	数据结束位错误状态位使能 0: 屏蔽 1: 使能
[21]	DCESEN	数据 CRC 错误状态位使能 0: 屏蔽 1: 使能
[20]	DTESEN	数据超时错误状态位使能 0: 屏蔽 1: 使能
[19]	CIESEN	命令索引错误状态位使能 0: 屏蔽 1: 使能
[18]	CEESEN	命令结束位错误状态位使能 0: 屏蔽 1: 使能
[17]	CCESSEN	命令 CRC 错误状态位使能 0: 屏蔽 1: 使能
[16]	CTESEN	命令超时错误状态位使能 0: 屏蔽 1: 使能

位	字段	描述
[7]	BESEN	缓冲器为空状态位使能 0: 屏蔽 1: 使能
[6]	BFSEN	缓冲器已满状态位使能 0: 屏蔽 1: 使能
[5]	BHSEN	缓冲器半满状态位使能 0: 屏蔽 1: 使能
[4]	BUSEN	缓冲器下溢状态位使能 0: 屏蔽 1: 使能
[3]	BOSEN	缓冲器上溢状态位使能 0: 屏蔽 1: 使能
[1]	TCSEN	传输完成状态使能 0: 屏蔽 1: 使能
[0]	CCSEN	命令完成状态使能 0: 屏蔽 1: 使能

## 中断使能寄存器 – IER

这些状态位是共用一条中断线中的某一个位。这些位中任意一个位置位将使能中断。

偏移量: 0x04C

复位值: 0x0000\_0000

类型 / 复位	31	30	29	28	27	26	25	24
	保留位					CICEIEN	CIDEIEN	CIEN
						RW	0	RW
类型 / 复位	23	22	21	20	19	18	17	16
	保留位	DEEIEN	DCEIEN	DTEIEN	CIEIEN	CEEIEN	CCEIEN	CTEIEN
		RW	0	RW	0	RW	0	RW
类型 / 复位	15	14	13	12	11	10	9	8
	保留位							
类型 / 复位	7	6	5	4	3	2	1	0
	BEIEN	BFIEN	BHIEN	BUIEN	BOIEN	保留位	TCIEN	CCIEN
	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[26]	CICEIEN	命令禁止 (CMD) 错误中断使能 0: 屏蔽 1: 使能
[25]	CIDEIEN	命令禁止 (DAT) 错误中断使能 0: 屏蔽 1: 使能
[24]	CIEN	卡中断状态中断使能 0: 屏蔽 1: 使能
[22]	DEEIEN	数据结束位错误中断使能 0: 屏蔽 1: 使能
[21]	DCEIEN	数据 CRC 错误中断使能 0: 屏蔽 1: 使能
[20]	DTEIEN	数据超时错误中断使能 0: 屏蔽 1: 使能
[19]	CIEIEN	命令索引错误中断使能 0: 屏蔽 1: 使能
[18]	CEEIEN	命令结束位错误中断使能 0: 屏蔽 1: 使能
[17]	CCEIEN	命令 CRC 错误中断使能 0: 屏蔽 1: 使能
[16]	CTEIEN	命令超时错误中断使能 0: 屏蔽 1: 使能

位	字段	描述
[7]	BEIEN	缓冲器为空中断使能 0: 屏蔽 1: 使能
[6]	BFIEN	缓冲器已满中断使能 0: 屏蔽 1: 使能
[5]	BHIEN	缓冲器半满中断使能 0: 屏蔽 1: 使能
[4]	BUIEN	缓冲器下溢中断使能 0: 屏蔽 1: 使能
[3]	BOIEN	缓冲器上溢中断使能 0: 屏蔽 1: 使能
[1]	TCIEN	传输完成中断使能 0: 屏蔽 1: 使能
[0]	CCIEN	命令完成中断使能 0: 屏蔽 1: 使能



# 31 CMOS 传感器接口 (CSIF, 仅适用于 HT32F22366)

## 简介

CMOS 传感器接口, 即 CSIF, 提供了一个接口用于从 CMOS 传感器捕捉图像。单片机可利用这个 CMOS 传感器接口直接连接到 CMOS 传感器。CSIF 支持完成图像捕捉的垂直 SYNC (VSYNC) 模式和水平 SYNC (HSYNC) 模式。CSIF 包含有窗口捕捉和子采样功能以及两个 FIFO, 每个容量为 8×32 位, FIFO 中的数据可通过外设直接存储器访问电路 (PDMA) 或 CPU 移入内部 SRAM。CSIF 不支持图像数据转换或解码, 但可把 CMOS 传感器接收到的图像数据清晰地传输到内部。

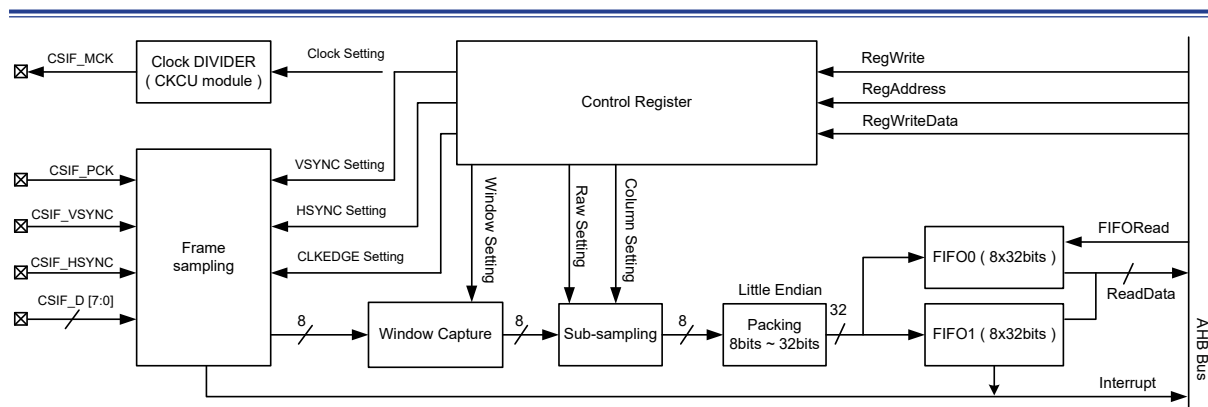


图 235. CSIF 方框图

## 特性

- 可达 2048×2048 输入分辨率
- 支持 8-bit YUV422 和 Raw RGB 格式
- 可达 32 MHz 输入像素时钟频率 @ HCLK = 96 MHz
- VSYNC 和 HSYNC 不同设置用于图像捕捉
- 硬件窗口捕捉功能
- 分级硬件子采样功能
- 两个 FIFO, 每个容量为 8×32 位, 可通过 PDMA 或 CPU 读取

功能描述

CSIF 信号

该信号有 5 种类型。CSIF 具有不同的内部设置以便捕捉不同格式的图像。

表 88. CSIF 信号

信号	方向	描述
CSIF_MCK	输出	输出时钟到 CMOS 传感器
CSIF_PCK	输入	来自 CMOS 传感器像素时钟
CSIF_VSYNC	输入	来自 CMOS 传感器的垂直 SYNC 信号
CSIF_HSYNC	输入	来自 CMOS 传感器的水平 SYNC 信号
CSIF_D [7:0]	输入	来自 CMOS 传感器的像素数据

为了获得不同图像帧速率，应用软件可设置时钟控制单元 CKCU 中的 GCFGR 寄存器 CSIFMPRE 字段，为 CMOS 传感器产生各种不同时钟。出于功耗和其它设计的考虑，CSIF\_MCK 预分频器置于 CKCU。欲知 CSIF\_MCK 预分频器设置信息详情，请参考 CKCU 章节。

表 89. CSIF\_MCK 输出设置 – 请参考 CKCU 章节

寄存器名称	寄存器字段	CSIF_MCK 输出
GCFGR	CSIFMPRE [4:0]	$CK\_PLL/(CSIFMPRE + 1)/2$

CSIF 帧时序

该模块提供垂直 SYNC (VSYNC) 模式和水平 SYNC (HSYNC) 模式的设置。VSYNC 信号有两种类型，一种是短脉冲有效 (HSYNC 信号无重叠)，另一种是帧正确时才有效 (HSYNC 信号有重叠)。所需类型通过 VSYNCTYP 位选择。HSYNC 信号也有两种类型，一种是持续有效，另一种是帧正确时才有效。所需类型通过 HSYNCTYP 位选择。VSYNCPOL 和 HSYNCPOL 位用于决定 VSYNC 和 HSYNC 信号极性。由 CLKEDGE 位决定是在 CSIF\_PCK 上升沿还是下降沿采样数据。IMG\_SFD 字段定义了需丢弃的线数目，IMG\_SLD 字段定义了需丢弃的像素数目。用户可使用 IMG\_SLD 和 IMG\_SFD 字段丢弃像素和线以获得确切的有效的图像数据。图像垂直高度等于 IMG\_HGH 值，以 1 递增。图像水平宽度等于 MG\_WID 字段的值，以 1 递增。因此，用户可选择不同设置匹配 CMOS 传感器输出时序以捕捉图像。

如果 CSIF 使能位 CSIF\_EN 置 1，图像帧开始且 EOF\_FLG 和 CAP\_STS 标志位清零时，CSIF 开始捕捉图像。若 EOF\_FLG 或 CAP\_STS 标志位被置位，CSIF 将不捕捉图像数据不论图像帧是否已开始。有两种方案可捕捉下一个图像数据。一种是清零 EOF\_FLG 和 CAP\_STS 标志位，另一种是先通过清除 CSIF\_EN 位除能 CSIF 功能然后通过设置 CSIF\_EN 位重新使能 CSIF 功能。

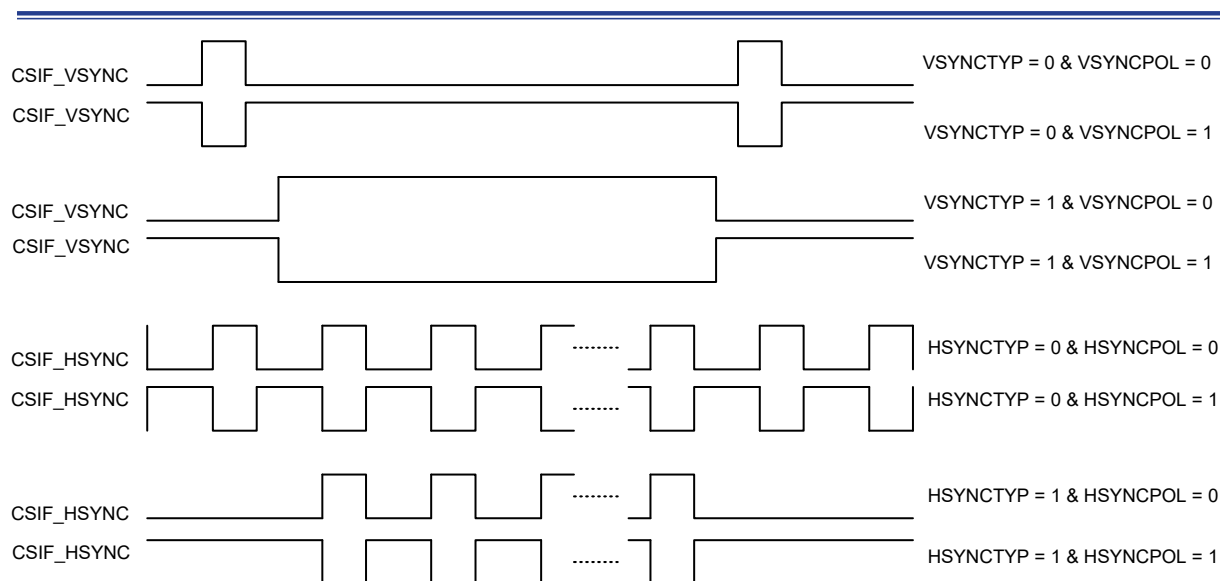


图 236. VSYNC 和 HSYNC 时序图

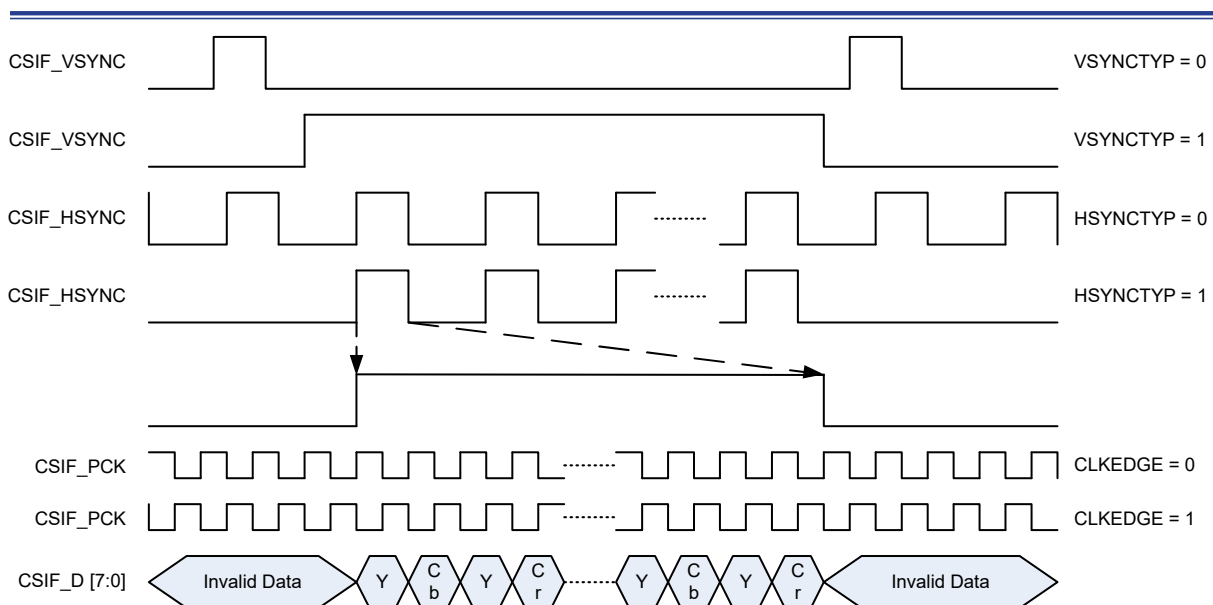


图 237. CSIF 帧时序图

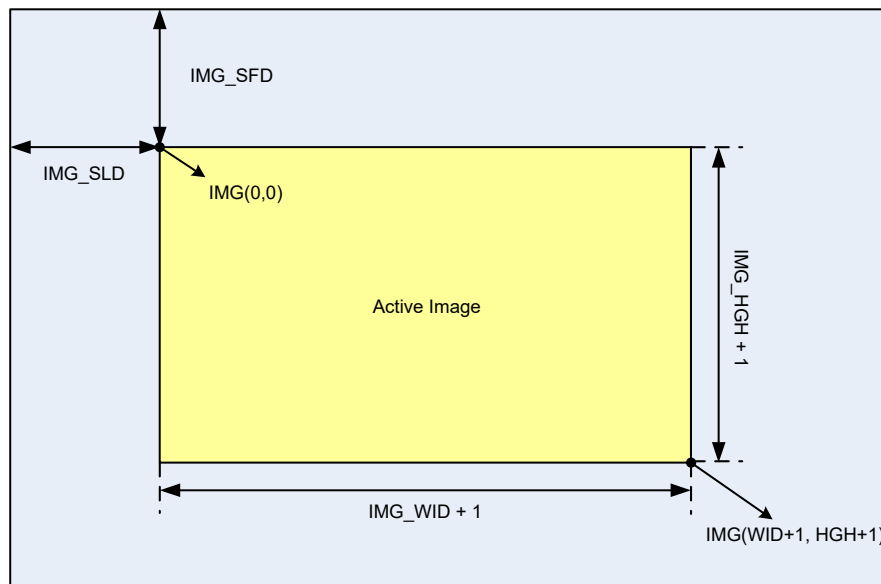


图 238. 图像结构

## 像素数据格式

CSIF 提供两种像素数据格式设置，Raw RGB 格式和 YUV422 格式。Raw RGB 格式是每个像素点一个字节数据，YUV422 格式是每个像素点两个字节数据。像素数据是在 CSIF\_PCK 时钟上升沿还是下降沿被采样由 CLKEDGE 位决定。每个像素数据字节是一个数据包，每四个字节填充小尾数模式 (P3P2P1P0)，存储在 FIFO。软件可用该设置匹配 CMOS 传感器数据格式。

表 90. 像素数据格式 – 无窗口捕捉和子采样

模式	BYTE3	BYTE2	BYTE1	BYTE0
Raw RGB 1 字节 / 像素	R1	G1	R0	G0
	R3	G3	R2	G2
YUV422 2 字节 / 像素	V0	Y1	U0	Y0
	V1	Y3	U1	Y2

## 窗口捕捉

CSIF 提供窗口捕捉功能，该功能可以减小图像大小进而减少存储器使用。该系列单片机配置了 5 个寄存器来完成窗口捕捉功能。由 WIN\_HSTR 和 WIN\_VSTR 字段定义的位置，是有效图像区域的起点。WIN\_WID 和 WIN\_HGH 字段定义了图像捕捉区域。注意，WIN\_WID 和 WIN\_HGH 字段定义的窗口区域是基于窗口起点，必须是在 IMG\_WID 和 IMG\_HGH 字段定义的有效图像区域。如果窗口区域部分超出 IMG\_WID 和 IMG\_HGH 字段定义的有效图像区域，有效图像区域的像素点数据将被丢弃。

表 91. 窗口捕捉设置

设置	寄存器	描述
WIN_EN	CSIFWCR0[31]	窗口捕捉使能
WIN_HSTR	CSIFWCR0[10:0]	窗口水平起点
WIN_VSTR	CSIFWCR0[26:16]	窗口垂直起点
WIN_WID	CSIFWCR1[10:0]	窗口宽度
WIN_HGH	CSIFWCR1[26:16]	窗口高度

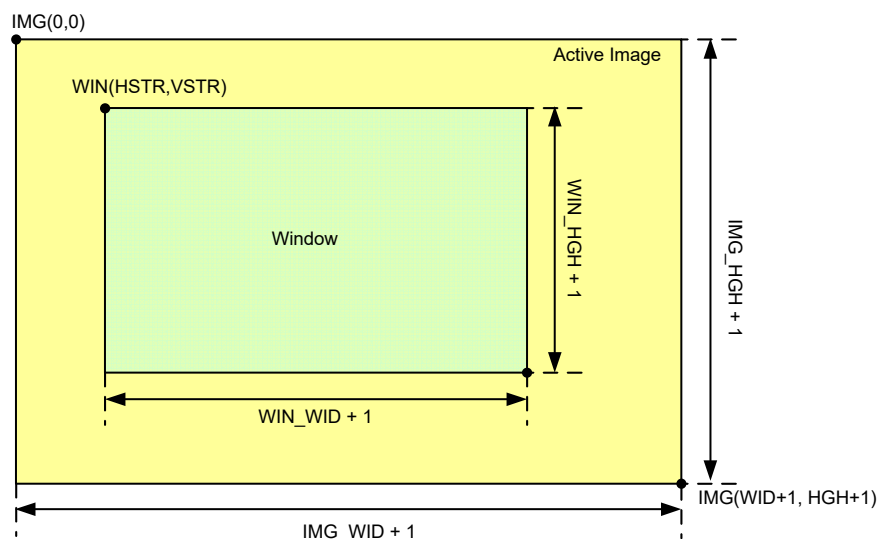


图 239. 正确窗口捕捉

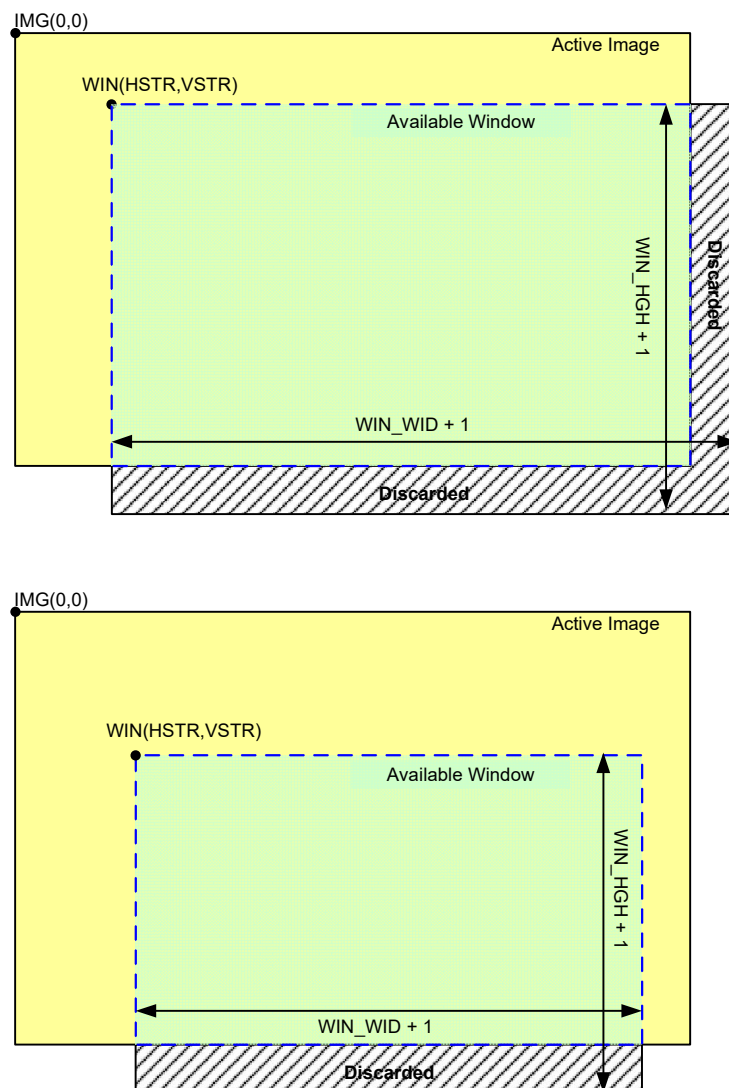


图 240. 错误的窗口捕捉

### 行和列子采样

CSIF 提供捕捉窗口分级行列子采样功能。该功能根据应用算法获得分级图像采样。CSML 和 RSML 字段用于定义子采样的行列长度。CSM 和 RSM 用于控制子采样的行列采样功能。CSIF 根据 CSM 和 RSM 使能位和 CSML、RSML 字段定义的子域连续采样数据。如果 CSM 或 RSM 采样使能控制位清除，CSIF 将会丢弃相关数据。

表 92. 行和列子采样设置

寄存器设置	描述	注释
SMP_EN	行和列子采样使能	CSIFSMP [31]
CSML [4:0]	列子采样长度	列子采样长度 =CSML+1
CSM [31:0]	列采样使能控制	列 n 采样使能控制： CSM [n], n=0~31 1: 使能采样 0: 除能采样
RSML [4:0]	行子采样长度	行子采样长度 =RSML+1
RSM [31:0]	行采样使能控制	行 n 采样使能控制： RSM [n], n=0~31 1: 使能采样 0: 除能采样

R \ C	0	1	2	3	4	5	6	7
0	G1	R	G1	R	G1	R	G1	R
1	B	G2	B	G2	B	G2	B	G2
2	G1	R	G1	R	G1	R	G1	R
3	B	G2	B	G2	B	G2	B	G2
4	G1	R	G1	R	G1	R	G1	R
5	B	G2	B	G2	B	G2	B	G2
6	G1	R	G1	R	G1	R	G1	R
7	B	G2	B	G2	B	G2	B	G2

Column Sample Length  
CSML [4:0] = 0x07  
Column Sample  
CSM [7:0] = 0011\_1111  
C[7].....C[0]

Row Sample Length  
RSML [4:0] = 0x07  
Row Sample  
RSM [7:0] = 0011\_1111  
R[7].....R[0]

R \ C	0	1	2	3	4	5	6	7	8	9
0	G1	R	G1	R	G1	R	G1	R	G1	R
1	B	G2	B	G2	B	G2	B	G2	B	G2
2	G1	R	G1	R	G1	R	G1	R	G1	R
3	B	G2	B	G2	B	G2	B	G2	B	G2
4	G1	R	G1	R	G1	R	G1	R	G1	R
5	B	G2	B	G2	B	G2	B	G2	B	G2
6	G1	R	G1	R	G1	R	G1	R	G1	R
7	B	G2	B	G2	B	G2	B	G2	B	G2
8	G1	R	G1	R	G1	R	G1	R	G1	R
9	B	G2	B	G2	B	G2	B	G2	B	G2

Column Sample Length  
CSML [4:0] = 0x09  
Column Sample  
CSM [9:0] = 00\_1100\_1111  
C[9] .....C[0]

Row Sample Length  
RSML [4:0] = 0x09  
Row Sample  
RSM [9:0] = 00\_1100\_1111  
R[9] .....R[0]

R \ C	0	1	2	3	4	5	6	7
0	Y	Cb	Y	Cr	Y	Cb	Y	Cr
1	Y	Cb	Y	Cr	Y	Cb	Y	Cr
2	Y	Cb	Y	Cr	Y	Cb	Y	Cr
3	Y	Cb	Y	Cr	Y	Cb	Y	Cr
4	Y	Cb	Y	Cr	Y	Cb	Y	Cr
5	Y	Cb	Y	Cr	Y	Cb	Y	Cr
6	Y	Cb	Y	Cr	Y	Cb	Y	Cr
7	Y	Cb	Y	Cr	Y	Cb	Y	Cr

Column Sample Length  
CSML [4:0] = 0x01  
Column Sample  
CSM [1:0] = 01  
C[1]C[0]

Row Sample Length  
RSML [4:0] = 0x00  
Row Sample  
RSM [0] = 1  
R[0]

图 241. 行和列子采样范例

## PDMA 数据发送 – Rx 触发 PDMA

CSIF 包含有两个 FIFO，每个 FIFO 有 8 字容量，即 8×32 位。当一个 FIFO 满时，CSIF 将会发送一个请求到 PDMA 请求服务。CSIF 每次总是发送 8 字数据，通过 PDMA 传送到 SRAM 以减少 PDMA 延迟时间。如果 FIFO 的有效数据少于 8 字，CSIF 会自动用 0x0000 把剩余位填满。

## 中断和状态

该系列单片机提供几种中断和相关状态寄存器，用于确认当前 CSIF 状态以便之后的决定和动作。CSIF 中断包括帧开始中断，帧结束中断，捕捉开始中断，捕捉状态中断，坏帧中断，FIFO 溢出中断，FIFO 空中断和 FIFO 满中断。帧开始中断如果使能，当接收到通过 VSYNC POL 和 VSYNCTYP 位决定的 CSIF\_VSYNC 信号时，中断发生。如果相关中断使能，CSIF 接收到所有通过 IMG\_WID、IMG\_HGH 和 IMG\_FMT 设置的动态图像数据位，帧结束中断发生。相关中断使能且接收到首个动态图像，捕捉开始中断发生。所有要被捕捉的像素数据已接收到且通过 PDMA 或 CPU 接口传送到 SRAM，捕捉状态中断发生。CSIF 检测到异常的 VSYNC 和 HSYNC 信号，坏帧中断发生。两个 FIFO 已满且有数据继续写入时，FIFO 溢出中断请求发生。两个 FIFO 为空，FIFO 空中断请求状态置 1；其中一个已满，清零。当一个 FIFO 存储图像数据已满，FIFO 满中断请求标志位置位；该 FIFO 的数据被读取时，标志位清零。

表 93. 中断和状态

中断状态	寄存器	描述
SOF_FLG	CSIFSR [0]	帧开始
EOF_FLG	CSIFSR [1]	帧结束
CAP_STA	CSIFSR [2]	捕捉开始
CAP_STS	CSIFSR [3]	捕捉状态：结束和没结束
BAD_FRM	CSIFSR [4]	坏帧
FIFO_OVR	CSIFSR [8]	FIFO 溢出
FIFO_EMP	CSIFSR [9]	FIFO 空
FIFO_FUL	CSIFSR [10]	FIFO 满

表 94. 中断状态

中断状态	描述	建立标准	清除标准
SOF_FLG	帧开始	CSIF_EN=1, EOF_FLG=0, CAP_STS=0, 有效 VSYNC 信号开始	写 1 清除
EOF_FLG	帧结束	CSIF_EN=1, 所有动态图像数据已被接收	写 1 清除
CAP_STA	捕捉开始	CSIF_EN=1, EOF_FLG=0, CAP_STS=0 动态图像起点首个数据 (丢弃 IMG_SLD 和 IMG_SFD 之后) 已被捕捉	写 1 清除
CAP_STS	捕捉状态	CSIF_EN=1, 所有被捕捉数据已接收到且通过 PDMA 传送	写 1 清除
BAD_FRM	坏帧	坏帧验收	写 1 清除
FIFO_OVR	FIFO 溢出	CSIF_EN=1, 两个 FIFO 已满。新数据字写入 FIFO	CSIF_EN 位下降沿
FIFO_EMP	FIFO 空	CSIF_EN=1, 两个 FIFO 为空	一个 FIFO 已满
FIFO_FUL	FIFO 满	CSIF_EN=1, 其中一个 FIFO 已满	从已满 FIFO 读取数据的一字



## 寄存器列表

以下表格显示了 CSIF 寄存器及其复位值。

**表 95. CSIF 寄存器列表**

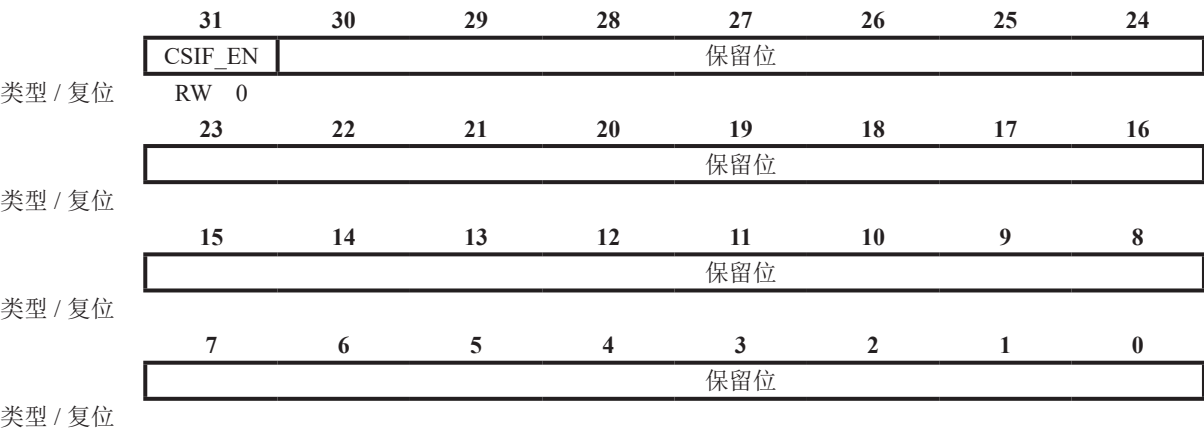
寄存器	偏移量	描述	复位值
CSIFENR	0x000	CSIF 使能寄存器	0x0000_0000
CSIFCR	0x004	CSIF 控制寄存器	0x0000_0004
CSIFIMGWH	0x008	CSIF 图像宽度和高度寄存器	0x0000_0000
CSIFWCR0	0x00C	CSIF 窗口捕捉寄存器 0	0x0000_0000
CSIFWCR1	0x010	CSIF 窗口捕捉寄存器 1	0x0000_0000
CSIFSMP	0x014	CSIF 子采样寄存器	0x0000_0000
CSIFSMPCOL	0x018	CSIF 列子采样寄存器	0x0000_0000
CSIFSMPCROW	0x01C	CSIF 行子采样寄存器	0x0000_0000
CSIFFIFO0	0x020	CSIF FIFO 寄存器 0	0x0000_0000
CSIFFIFO1	0x024	CSIF FIFO 寄存器 1	0x0000_0000
CSIFFIFO2	0x028	CSIF FIFO 寄存器 2	0x0000_0000
CSIFFIFO3	0x02C	CSIF FIFO 寄存器 3	0x0000_0000
CSIFFIFO4	0x030	CSIF FIFO 寄存器 4	0x0000_0000
CSIFFIFO5	0x034	CSIF FIFO 寄存器 5	0x0000_0000
CSIFFIFO6	0x038	CSIF FIFO 寄存器 6	0x0000_0000
CSIFFIFO7	0x03C	CSIF FIFO 寄存器 7	0x0000_0000
CSIFIER	0x040	CSI 中断使能寄存器	0x0000_0000
CSIFSR	0x044	CSIF 状态寄存器	0x0000_0000

寄存器描述

CSIF 使能寄存器 – CSIFENR

该寄存器定义了 CSIF 的使能控制。

偏移量: 0x000  
复位值: 0x0000\_0000



位	字段	描述
[31]	CSIF_EN	CSIF 使能控制位 0: 除能 CSIF 1: 使能 CSIF

31 CMOS 传感器接口 (CSIF, 仅适用于 HT32F22366)

## CSIF 控制寄存器 – CSIFCR

该寄存器包含 CSIF 控制位的各种类型包括图像格式、图像帧和线路延迟、采样时钟边沿选择、同步极性和类型等。

偏移量: 0x004

复位值: 0x0000\_0004

	31	30	29	28	27	26	25	24	
	保留位								
类型 / 复位									
	23	22	21	20	19	18	17	16	
	IMG_SFD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	IMG_SLD								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	HSYNCPOL	VSYNCPOL	PDMA_DIS	IMG_FMT	CLKEDGE	HSYNCTYP	VSYNCTYP	保留位	
类型 / 复位	RW	0	RW	0	RW	0	RW	0	

位	字段	描述
[23:16]	IMG_SFD	图像帧延时 帧延时 = 0x00~0xFF IMG_SFD 字段定义跳过的线数量，以指定有效的图像起点。
[15:8]	IMG_SLD	图像线延时 线延时 = 0x00~0xFF MG_SLD 字段定义跳过的线数量，以指定有效的图像起点。
[7]	HSYNCPOL	HSYNC 极性 0: HSYNC 为高电平 1: HSYNC 为低电平
[6]	VSYNCPOL	VSYNC 极性 0: VSYNC 为高电平 1: VSYNC 为低电平
[5]	PDMA_DIS	除能 PDMA 0: PDMA 模式 1: CPU 模式
[4]	IMG_FMT	图像格式 0: RAW RGB 1: YUV422
[3]	CLKEDGE	像素时钟采样边沿 0: 下降沿采样数据 1: 上升沿采样数据
[2]	HSYNCTYP	HSYNC 类型 0: 连续的 1: 帧有效期间有效
[1]	VSYNCTYP	VSYNC 类型 0: 脉冲 (没有与 HSYNC 重叠) 1: 帧有效期间有效 (与 HSYNC 重叠)

CSIF 图像宽度和高度寄存器 – CSIFIMGWH

该寄存器指定有效图像的宽度和高度设置。

偏移量: 0x008  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24
	保留位					IMG_HGH		
类型 / 复位						RW	0	RW
	23	22	21	20	19	18	17	16
	IMG_HGH							
类型 / 复位	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	保留位					IMG_WID		
类型 / 复位						RW	0	RW
	7	6	5	4	3	2	1	0
	IMG_WID							
类型 / 复位	RW	0	RW	0	RW	0	RW	0

位	字段	描述
[26:16]	IMG_HGH	图像高度 有效图像高度字段内容的范围为 0x0000~0x07FF。相应指定图像高度的范围实际为 0x0001~0x0800。
[10:0]	IMG_WID	图像宽度 有效图像宽度字段内容的范围为 0x0000~0x07FF。相应指定图像宽度的范围实际为 0x0001~0x0800。

CSIF 窗口捕捉寄存器 0 – CSIFWCR0

该寄存器包含窗口捕捉功能使能控制和指定的窗口捕捉起点。

偏移量： 0x00C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24
	WIN_EN	保留位					WIN_VSTR	
类型 / 复位	RW 0						RW 0 RW 0 RW 0	0
	23	22	21	20	19	18	17	16
	WIN_VSTR							
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	保留位					WIN_HSTR		
类型 / 复位						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	WIN_HSTR							
类型 / 复位	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

位	字段	描述
[31]	WIN_EN	窗口捕捉使能控制 0: 除能窗口捕捉 1: 使能窗口捕捉
[26:16]	WIN_VSTR	捕捉窗口的垂直起点 WIN_VSTR 字段内容为 0x0000~0x07FF，基于指定图像面积内的有效图像起点。
[10:0]	WIN_HSTR	捕捉窗口的水平起点 WIN_HSTR 字段内容为 0x0000~0x07FF，基于指定图像面积内的有效图像起点。

CSIF 窗口捕捉寄存器 1 – CSIFWCR1

该寄存器指定窗口捕捉的高度和宽度设置。

偏移量： 0x010  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	保留位					WIN_HGH			
类型 / 复位						RW	0	RW	0
	23	22	21	20	19	18	17	16	
	WIN_HGH								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	保留位					WIN_WID			
类型 / 复位						RW	0	RW	0
	7	6	5	4	3	2	1	0	
	WIN_WID								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[26:16]	WIN_HGH	窗口高度 WIN_HGH 字段内容范围为 0x0000~0x07FF，相应窗口的高度为 0x0001~0x0800。实际窗口高度应在有效图像面积内。
[10:0]	WIN_WID	窗口宽度 WIN_WID 字段内容范围为 0x0000~0x07FF，相应窗口的宽度为 0x0001~0x0800。实际窗口宽度应在有效图像面积内。

## CSIF 子采样寄存器 – CSIFSMP

该寄存器指定行和列子采样长度以及子采样功能使能控制。

偏移量: 0x014

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24					
	SMP_EN	保留位											
类型 / 复位	RW	0											
	23	22	21	20	19	18	17	16					
	保留位			RSML									
类型 / 复位				RW	0	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8					
	保留位			CSML									
类型 / 复位				RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0					
	保留位												
类型 / 复位													

位	字段	描述
[31]	SMP_EN	CSIF 行和列子采样使能控制 0: 除能子采样功能 1: 使能子采样功能
[20:16]	RSML	行子采样长度 0: 1 位 1: 2 位 ...
[12:8]	CSML	列子采样长度 0: 1 位 1: 2 位 ...
		31: 32 位

**CSIF 列子采样寄存器 – CSIFSMPCOL**

该寄存器定义了列采样使能控制。

偏移量: 0x018  
复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	CSM								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	CSM								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	CSM								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	CSM								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	CSM	列采样使能控制 0: 除能采样 1: 使能采样 第 n 列采样使能控制位 CSM[n]，用来决定相关的列数据被采样还是被丢弃。



**CSIF 行子采样寄存器 – CSIFSMPROW**

该寄存器定义了行采样使能控制。

偏移量： 0x01C  
复位值： 0x0000\_0000

	31	30	29	28	27	26	25	24	
	RSM								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	RSM								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	RSM								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	RSM								
类型 / 复位	RW	0	RW	0	RW	0	RW	0	RW

位	字段	描述
[31:0]	RSM	行采样使能控制 0: 除能采样 1: 使能采样 第 n 行采样使能控制位 RSM[n]，用来决定相关的行数据被采样还是被丢弃。

## CSIF FIFO 寄存器 n – CSIFFIFO<sub>n</sub>, n=0~7

该寄存器存储第 n 个 FIFO 数据字。

偏移量: 0x020~0x03C

复位值: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	FIFOData								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	FIFOData								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	FIFOData								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	FIFOData								
类型 / 复位	RO	0	RO	0	RO	0	RO	0	RO

位	字段	描述
[31:0]	FIFOData	FIFO 数据 FIFO 数据通过 PDMA 或 CPU 接口读取。

## CSIF 中断使能寄存器 – CSIFIER

该寄存器定义了 CSIF 中断使能控制位。

偏移量: 0x040

复位值: 0x0000\_0200

31	30	29	28	27	26	25	24
保留位							
类型 / 复位							
23	22	21	20	19	18	17	16
保留位							
类型 / 复位							
15	14	13	12	11	10	9	8
保留位					FIFOFUL	FIFOEMPE	FIFOOVRE
类型 / 复位					RW	0	RW
7	6	5	4	3	2	1	0
保留位			BADFRME	CAPSTSE	CAPSTAE	EOFFLGE	SOFFLGE
类型 / 复位			RW	0	RW	0	RW

位	字段	描述
[10]	FIFOFUL	FIFO 满中断使能位 0: 除能中断 1: 使能中断
[9]	FIFOEMPE	FIFO 空中断使能位 0: 除能中断 1: 使能中断
[8]	FIFOOVRE	FIFO 溢出中断使能位 0: 除能中断 1: 使能中断
[4]	BADFRME	坏帧中断使能位 0: 除能中断 1: 使能中断
[3]	CAPSTSE	捕捉状态中断使能位 0: 除能中断 1: 使能中断
[2]	CAPSTAE	捕捉开始中断使能位 0: 除能中断 1: 使能中断
[1]	EOFFLGE	帧结束中断使能位 0: 除能中断 1: 使能中断
[0]	SOFFLGE	帧开始中断使能位 0: 除能中断 1: 使能中断

## CSIF 状态寄存器 – CSIFSR

该寄存器包含 CSIF 状态。

偏移量: 0x044

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
保留位							
23	22	21	20	19	18	17	16
保留位							
15	14	13	12	11	10	9	8
保留位					FIFO_FUL	FIFO_EMP	FIFO_OVR
					RO	0	RO
7	6	5	4	3	2	1	0
保留位			BAD_FRM	CAP_STS	CAP_STA	EOF_FLG	SOF_FLG
			WC	0	WC	0	WC

位	字段	描述
[10]	FIFO_FUL	FIFO 满标志位 0: FIFO 未满 1: 一个 FIFO 是满的 一旦数据字从满 FIFO 读取, FIFO 满标志位将清零。
[9]	FIFO_EMP	FIFO 空标志位 0: FIFO 不是空的 1: 两个 FIFO 是空的 当一个 FIFO 完全布满数据字, FIFO 空标志位将清零。
[8]	FIFO_OVR	FIFO 溢出标志位 0: FIFO 没有溢出 1: FIFO 溢出 如果两个 FIFO 是满的, 一个新的数据字写入 FIFO, FIFO 溢出标志位置 1。通过将 CSIF_EN 清零, 使 FIFO 溢出标志位清零。
[4]	BAD_FRM	坏帧标志位 0: 接收到的帧大小是正确的 1: 接收到的帧大小是不正确的 当接收到的帧大小小于 CSIF 定义的有效图像的大小, 坏帧标志位将被置 1, 且相关的图像数据应被丢弃。该位通过写 1 来清零。
[3]	CAP_STS	捕捉状态标志位 0: 正在捕捉 1: 捕捉完成 当所有捕捉到的图像数据已经接收完, 且通过 PDMA 传送到 SRAM 时, 捕捉状态标志位将被置 1。该位通过写 1 来清零。
[2]	CAP_STA	捕捉启动标志位 0: 未启动捕捉 1: 启动捕捉 当第一个动态图像数据已经接收完, 相关 CSIF VSYNC 和 HSYNC 信号有效时, 捕捉启动标志位将被置 1。该位通过写 1 来清零。

位	字段	描述
[1]	EOF_FLG	帧结束标志位 0: 帧未结束 1: 帧已结束 当所有动态图像数据已经接收完，帧结束标志位将被置 1。该位通过写 1 来清零。
[0]	SOF_FLG	帧启动标志位 0: 帧未启动 1: 帧已启动 当 CSIF 接收到一个可用的 VSYNC 信号，帧启动标志位将被置 1，意味着将开始接收帧数据。该位通过写 1 来清零。

Copyright© 2020 by HOLTEK SEMICONDUCTOR INC.

使用指南中所出现的信息在出版当时相信是正确的，然而 **Holtek** 对于说明书的使用不负任何责任。文中提到的应用目的仅仅是用来做说明，**Holtek** 不保证或表示这些没有进一步修改的应用将是适当的，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。**Holtek** 产品不授权使用于救生、维生从机或系统中做为关键从机。**Holtek** 拥有不事先通知而修改产品的权利，对于最新的信息，请参考我们的网址 <http://www.holtek.com/zh/>.