PY32F420 系列参考手册

32 位 ARM®Cortex®-M0+微控制器

Puya Confidential

Puya Semiconductor (Shanghai) Co., Ltd.

目录

1. 文档约定 31

1.1 寄存器相关缩写词列表 31

1.2 外设可用性 31

2. 系统架构框图 32

3. 存储器和总线架构 33

3.1 系统架构 33

3.1.1 I_bus 33

3.1.2 D_bus 33

3.1.3 S_bus 34

3.1.4 DMA BUS 34

3.1.5 总线矩阵 34

3.1.6 AHB/APB 总线桥 34

3.2 存储器组织架构 34

3.2.1 简介 34

3.2.2 存储器映射 35

3.3 嵌入式 SRAM 38

3.4 嵌入式 FLASH 38

3.5 位段 38

3.6 启动配置 39

3.6.1 系统存储启动程序 39

3.6.2 物理重映射 40

4. 嵌入式 FLASH 接口 41

4.1 FLASH 简介 41

4.2 FLASH 主要特征 41

4.3 FLASH 功能描述 41

4.3.1 闪存结构 41

4.3.2 读访问延迟 42

4.3.3 自适应实时存储器加速器 43

4.3.4 Flash 擦除和编程操作 43

4.3.5 闪存解锁 44

4.3.6 闪存擦除操作 44

Puya Confidential

4.3.7 闪存写操作 45
4.4 产品唯一身份标识码（UID） 46
4.5 FLASH 选项字节 46
4.5.1 Flash 选项字节编程 46
4.5.2 修改选项字节 47
4.5.3 修改选项字节 BFB（Boot from Bank） 47
4.5.4 加载选项字节 47
4.6 FLASH 用户数据字节（OTP） 49
4.7 FLASH 存储区保护 49
4.7.1 闪存读保护（RDP） 50
4.7.2 闪存写保护（WRP） 51
4.7.3 专有代码读出保护（PCROP） 51
4.7.4 安全内存区域（SECPROT） 52
4.7.5 强制从 Main flash 启动 52
4.8 VERIFY READ 功能 52
4.9 FLASH 中断 53
4.10 FLASH 寄存器描述 53
4.10.1 Flash 访问控制寄存器（FLASH_ACR） 53
4.10.2 Flash 密钥寄存器（FLASH_KEYR） 54
4.10.3 Flash 选项密钥寄存器（FLASH_OPTKEYR） 54
4.10.4 Flash 状态寄存器（FLASH_SR） 55
4.10.5 Flash 控制寄存器（FLASH_CR） 56
4.10.6 Flash 选项 1 寄存器（FLASH_OPTR1） 58
4.10.7 Flash 选项 2 寄存器（FLASH_OPTR2） 59
4.10.8 Flash WRP 地址寄存器（FLASH_WRPR0） 60
4.10.9 Flash PCROP 地址寄存器（FLASH_PCROPR0） 60
4.10.10 Flash SECR 地址寄存器（FLASH_SECR0） 61
4.10.11 Flash WRP 地址寄存器（FLASH_WRPR1） 61
4.10.12 Flash PCROP 地址寄存器（FLASH_PCROPR1） 62
4.10.13 Flash SECR 地址寄存器（FLASH_SECR1） 62
4.10.14 Flash 选项字节 BFB 寄存器（FLASH_OPTBFBR） 62
5. 电源控制（PWR） 64
5.1 简介 64
5.2 电源结构 64

Puya Confidential

5.2.1 系统主要电源 65
5.2.2 PWR 电源检测 65
5.3 系统低功耗模式 67
5.3.1 正常运行模式 70
5.3.2 低功耗运行模式 70
5.3.3 低功耗模式进入和退出 71
5.3.4 睡眠模式 71
5.3.5 低功耗睡眠模式 72
5.3.6 停止模式 72
5.3.7 待机模式 73
5.3.8 停止模式和待机模式的调试 74
5.4 寄存器描述 75
5.4.1 电源控制寄存器 1（PWR_CR1） 75
5.4.2 电源控制寄存器 2（PWR_CR2） 76
5.4.3 电源控制寄存器 3（PWR_CR3） 77
5.4.4 电源控制寄存器 4（PWR_CR4） 78
5.4.5 电源状态寄存器（PWR_SR） 78
5.4.6 电源状态清除寄存器（PWR_SCR） 79
6. 复位与时钟控制（RCC） 81
6.1 RCC 简介 81
6.2 RCC 复位 81
6.2.1 电源复位 81
6.2.2 系统复位 81
6.2.3 备份域复位 82
6.2.4 备份域复位之外的复位的统一处理 82
6.2.5 复位标志表 83
6.3 RCC 时钟 84
6.3.1 时钟源 84
6.3.2 时钟结构 85
6.3.3 时钟安全系统（CSS） 85
6.3.4 时钟输出 86
6.3.5 外设时钟使能寄存器 86
6.4 RCC 寄存器 86
6.4.1 RCC 时钟控制寄存器（RCC_CR） 86

Puya Confidential

6.4.2 RCC 内部时钟源校准寄存器（RCC_ICSCR）.....88
6.4.3 RCC 时钟配置寄存器 1（RCC_CFGR1）.....89
6.4.4 RCC 时钟配置寄存器 2（RCC_CFGR2）.....92
6.4.5 RCC PLL 时钟配置寄存器 1（RCC_PLLCFGR1）.....92
6.4.6 RCC 外部时钟源配置寄存器（RCC_ECSCR）.....93
6.4.7 RCC 时钟中断使能寄存器（RCC_CIER）.....94
6.4.8 RCC 时钟中断标志寄存器（RCC_CIFR）.....94
6.4.9 RCC 时钟中断清零寄存器（RCC_CICR）.....95
6.4.10 RCC AHB 外设复位寄存器 1（RCC_AHBRSTR1）.....96
6.4.11 RCC AHB 外设复位寄存器 2（RCC_AHBRSTR2）.....97
6.4.12 RCC APB 外设复位寄存器 1（RCC_APBRSTR1）.....97
6.4.13 RCC APB 外设复位寄存器 2（RCC_APBRSTR2）.....98
6.4.14 RCC APB 外设复位寄存器 3（RCC_APBRSTR3）.....98
6.4.15 RCC APB 外设复位寄存器 4（RCC_APBRSTR4）.....99
6.4.16 RCC AHB 外设时钟使能寄存器 1（RCC_AHBENR1）.....100
6.4.17 RCC AHB 外设时钟使能寄存器 2（RCC_AHBENR2）.....101
6.4.18 RCC APB 外设时钟使能寄存器 1（RCC_APBENR1）.....101
6.4.19 RCC APB 外设时钟使能寄存器 2（RCC_APBENR2）.....102
6.4.20 RCC APB 外设时钟使能寄存器 3（RCC_APBENR3）.....103
6.4.21 RCC APB 外设时钟使能寄存器 4（RCC_APBENR4）.....104
6.4.22 RCC 外设独立时钟配置寄存器 1（RCC_CCIPR1）.....105
6.4.23 RCC 外设独立时钟配置寄存器 2（RCC_CCIPR2）.....106
6.4.24 RCC 外设独立时钟配置寄存器 3（RCC_CCIPR3）.....107
6.4.25 RCC 备份域控制寄存器（RCC_BDCR）.....107
6.4.26 RCC 控制/状态寄存器（RCC_CSR）.....109
7. CRC 计算单元（CRC）.....111
7.1 CRC 简介.....111
7.2 CRC 主要特性.....111
7.3 CRC 功能描述.....111
7.3.1 CRC 结构框图.....111
7.3.2 CRC 操作.....111
7.4 CRC 寄存器.....112
7.4.1 CRC 数据寄存器（CRC_DR）.....112
7.4.2 CRC 独立数据寄存器（CRC_IDR）.....112

Puya Confidential

7.4.3 CRC 控制寄存器（CRC_CR） 112
8. 通用 IO（GPIO） 114
8.1 GPIO 简介 114
8.2 GPIO 主要特征 114
8.3 GPIO 功能说明 114
8.3.1 通用 IO（GPIO） 116
8.3.2 IO 引脚复用器和映射 116
8.3.3 IO 端口控制寄存器 116
8.3.4 IO 端口数据寄存器 117
8.3.5 IO 数据位操作 117
8.3.6 GPIO 锁定机制 117
8.3.7 IO 复用功能输入/输出 117
8.3.8 外部中断线/唤醒线 118
8.3.9 输入配置 118
8.3.10 输出配置 118
8.3.11 复用功能配置 119
8.3.12 模拟配置 119
8.3.13 使用 PD0、PD1（HSE 振荡器引脚）作为 GPIO 120
8.3.14 使用 PD3（BOOT0）作为 GPIO 120
8.3.15 使用 PD11 作为 GPIO 120
8.4 GPIO 寄存器 121
8.4.1 GPIO 端口模式寄存器（GPIOx_MODER）(x= A..D) 121
8.4.2 GPIO 端口输出类型寄存器（GPIOx_OTYPER）(x= A..D) 121
8.4.3 GPIO 端口输出速度寄存器（GPIOx_OSPEEDR）(x= A..D) 122
8.4.4 GPIO 端口上拉/下拉寄存器（GPIOx_PUPDR）(x= A..D) 122
8.4.5 GPIO 端口输入数据寄存器（GPIOx_IDR）(x= A..D) 122
8.4.6 GPIO 端口输出数据寄存器（GPIOx_ODR）(x= A..D) 123
8.4.7 GPIO 端口置位/复位寄存器（GPIOx_BSRR）(x= A..D) 123
8.4.8 GPIO 端口配置锁定寄存器（GPIOx_LCKR）(x= A..D) 123
8.4.9 GPIO 复用功能低寄存器（GPIOx_AFRL）(x= A..D) 124
8.4.10 GPIO 复用功能高寄存器（GPIOx_AFRH）(x= A..D) 125
8.4.11 GPIO 端口位复位寄存器（GPIOx_BRR）(x= A..D) 125
9. 外设互联 126

Puya Confidential

9.1 简介 126
9.2 互联详情 126
9.2.1 定时器输入触发（ITR） 126
9.2.2 定时器外部触发（ETR） 126
9.2.3 清除定时器 OCxREF 信号 127
9.2.4 定时器输入捕获 127
9.2.5 定时器刹车 128
9.2.6 定时器系统刹车 128
9.2.7 低功耗定时器触发 129
9.2.8 比较器消隐 129
9.2.9 ADC 硬件触发输入 129
9.2.10 DAC 硬件触发输入 130
10. 系统配置控制器（SYSCFG） 132
10.1 SYSCFG 主要特性 132
10.2 SYSCFG 寄存器 132
10.2.1 SYSCFG 配置寄存器 1（SYSCFG_CFGR1） 132
10.2.2 SYSCFG 配置寄存器 2（SYSCFG_CFGR2） 133
10.2.3 SYSCFG 配置寄存器 3（SYSCFG_CFGR3） 133
10.2.4 SYSCFG 配置寄存器 4（SYSCFG_CFGR4） 134
10.2.5 SYSCFG 外部中断配置寄存器 1（SYS_EXTICR1） 134
10.2.6 SYSCFG 外部中断配置寄存器 2（SYS_EXTICR2） 135
10.2.7 SYSCFG 外部中断配置寄存器 3（SYS_EXTICR3） 136
10.2.8 SYSCFG 外部中断配置寄存器 4（SYS_EXTICR4） 137
10.2.9 SYSCFG GPIOA 滤波使能寄存器（PA_ENS） 138
10.2.10 SYSCFG GPIOB 滤波使能寄存器（PB_ENS） 139
10.2.11 SYSCFG GPIOC 滤波使能寄存器（PC_ENS） 139
10.2.12 SYSCFG GPIOD 滤波使能寄存器（PD_ENS） 139
11. DMA 控制器（DMA） 141
11.1 DMA 简介 141
11.2 DMA 主要特性 141
11.3 DMA 功能描述 142
11.3.1 DMA 顶层结构图 142
11.3.2 CPU 和 DMA 总线共享 142

Puya Confidential

11.3.3 DMA 传输 142
11.3.4 DMA 仲裁器 142
11.3.5 DMA 通道 143
11.3.6 可编程数据位宽，数据对齐和大小端 144
11.3.7 错误处理 146
11.3.8 DMA 外设请求映射 146
11.4 DMA 中断 147
11.5 DMA 寄存器 147
11.5.1 DMA 中断状态寄存器（DMA_ISR） 147
11.5.2 DMA 中断标志清零寄存器（DMA_IFCR） 149
11.5.3 DMA 通道 x 控制寄存器（DMA_CCRx）(x=1~5) 151
11.5.4 DMA 通道 x 数据传输个数寄存器（DMA_CNDTRx）(x=1~5) 153
11.5.5 DMA 通道 x 外设地址寄存器（DMA_CPARx）(x=1~5) 154
11.5.6 DMA 通道 x 存储地址寄存器（DMA_CMARx）(x=1~5) 154
11.5.7 DMA 通道 x 块循环传输配置寄存器（DMA_CCCFGRx）(x=1~5) 155
12. 中断和事件 156
12.1 嵌套向量中断控制器（NVIC） 156
12.1.1 NVIC 主要特性 156
12.1.2 系统滴答（SysTick）校准值寄存器 156
12.1.3 中断和异常向量表 156
12.2 外部扩展中断/事件控制器（EXTI） 158
12.2.1 EXTI 主要特性 158
12.2.2 EXTI 唤醒事件管理 159
12.2.3 EXTI 功能说明 159
12.2.4 EXTI 外部中断/事件线映射 161
12.3 EXTI 寄存器 162
12.3.1 EXTI 中断屏蔽寄存器（EXTI_IMR） 162
12.3.2 EXTI 事件屏蔽寄存器（EXTI_EMR） 162
12.3.3 EXTI 上升沿触发选择寄存器（EXTI_RTSR） 163
12.3.4 EXTI 下降沿触发选择寄存器（EXTI_FTSR） 163
12.3.5 EXTI 软件中断事件寄存器（EXTI_SWIER） 163
12.3.6 EXTI 挂起寄存器（EXTI_PR） 164
13. 模拟/数字转换（ADC） 165

Puya Confidential

13.1 ADC 简介 165
13.2 ADC 主要特性 165
13.3 ADC 功能描述 167
13.3.1 ADC 框图 167
13.3.2 单端和差分输入通道 167
13.3.3 ADC 校准 168
13.3.4 ADC 开-关控制（ADEN、ADDIS、ADRDY） 169
13.3.5 转换启动(ADSTART，JADSTART) 170
13.3.6 停止正在进行的转换（ADSTP、JADSTP） 171
13.3.7 ADC 时序（单次/连续模式，硬件/软件触发） 172
13.3.8 写 ADC 控制位的限制 174
13.3.9 ADC 时钟 174
13.3.10 ADC 通道选择（SQRx，JSQR） 175
13.3.11 动态低功耗特性 176
13.3.12 转换模式 179
13.3.13 注入通道管理 181
13.3.14 模拟看门狗 182
13.3.15 数据处理 185
13.3.16 非过载非 DMA 模式 189
13.3.17 DMA 模式 189
13.3.18 过采样器 190
13.3.19 可编程采样时间 196
13.3.20 外部触发转换 196
13.3.21 可配置分辨率 197
13.3.22 温度传感器和内部参考电压 198
13.3.23 电池监视 199
13.3.24 双 ADC 模式 199
13.4 ADC 中断 212
13.5 ADC 寄存器 213
13.5.1 ADC 中断和状态寄存器（ADC_ISR） 213
13.5.2 ADC 中断使能寄存器（ADC_IER） 214
13.5.3 ADC 控制寄存器（ADC_CR） 216
13.5.4 ADC 配置寄存器（ADC_CFGR） 218
13.5.5 ADC 配置寄存器 2（ADC_CFGR2） 221

Puya Confidential

13.5.6 ADC 采样时间寄存器 1（ADC_SMPR1） 223
13.5.7 ADC 采样时间寄存器 2（ADC_SMPR2） 223
13.5.8 ADC 采样时间寄存器 3（ADC_SMPR3） 224
13.5.9 ADC 看门狗阈值寄存器 1（ADC_TR1） 224
13.5.10 ADC 看门狗阈值寄存器 2（ADC_TR2） 225
13.5.11 ADC 看门狗阈值寄存器 3 (ADC_TR3) 225
13.5.12 ADC 规则序列寄存器 1(ADC_SQR1) 226
13.5.13 ADC 规则序列寄存器 2(ADC_SQR2) 226
13.5.14 ADC 规则序列寄存器 3(ADC_SQR3) 227
13.5.15 ADC 规则序列寄存器 4(ADC_SQR4) 228
13.5.16 ADC 规则数据寄存器（ADC_DR) 228
13.5.17 ADC 注入序列寄存器(ADC_JSQR) 228
13.5.18 ADC offset y 寄存器(ADC_OFRy) 229
13.5.19 ADC 注入数据寄存器（ADC_JDRy) 230
13.5.20 ADC 模拟看门狗 2 配置寄存器（ADC_AWD2CR) 231
13.5.21 ADC 模拟看门狗 3 配置寄存器（ADC_AWD3CR) 231
13.5.22 ADC 差分模式选择寄存器（ADC_DIFSEL) 232
13.5.23 ADC 校准因子寄存器（ADC_CALFACT) 233
13.5.24 ADC 增益补偿寄存器（ADC_GCOMP) 234
13.5.25 ADC 模拟控制寄存器（ADC_ANACR) 235
13.5.26 ADC 公共寄存器 235
14. 数字/模拟转换（DAC） 240
14.1 DAC 简介 240
14.2 DAC 主要特性 240
14.3 DAC 功能描述 241
14.3.1 DAC 框图 241
14.3.2 DAC 校准 241
14.3.3 DAC 通道使能 242
14.3.4 DAC 输出缓存使能 242
14.3.5 DAC 数据格式 242
14.3.6 DAC 转换 243
14.3.7 DAC 输出电压 243
14.3.8 DAC 触发选择 244
14.3.9 DMA 功能 244

Puya Confidential

14.3.10 DAC 噪声生成 245
14.3.11 DAC 三角波生成 246
14.3.12 DAC 斜坡发生器 247
14.3.13 双 DAC 通道转换 249
14.3.14 DAC 输出时钟 252
14.4 DAC 寄存器 253
14.4.1 DAC 控制寄存器（DAC_CR） 253
14.4.2 DAC 软件触发寄存器（DAC_SWTRIGR） 254
14.4.3 DAC 通道 1 的 12 位右对齐数据保持寄存器（DAC_DHR12R1） 255
14.4.4 DAC 通道 1 的 12 位左对齐数据保持寄存器（DAC_DHR12L1） 255
14.4.5 DAC 通道 1 的 8 位右对齐数据保持寄存器（DAC_DHR8R1） 256
14.4.6 DAC 通道 2 的 12 位右对齐数据保持寄存器（DAC_DHR12R2） 256
14.4.7 DAC 通道 2 的 12 位左对齐数据保持寄存器（DAC_DHR12L2） 256
14.4.8 DAC 通道 2 的 8 位右对齐数据保持寄存器（DAC_DHR8R2） 256
14.4.9 DAC 双 DAC 的 12 位右对齐数据保持寄存器（DAC_DHR12RD） 257
14.4.10 DAC 双 DAC 的 12 位左对齐数据保持寄存器（DAC_DHR12LD） 257
14.4.11 DAC 双 DAC 的 8 位右对齐数据保持寄存器（DAC_DHR8RD） 257
14.4.12 DAC 通道 1 数据输出寄存器（DAC_DOR1） 258
14.4.13 DAC 通道 2 数据输出寄存器（DAC_DOR2） 258
14.4.14 DAC 状态寄存器（DAC_SR） 258
14.4.15 DAC 输出选择寄存器（DAC_OC) 259
14.4.16 DAC 通道 1 斜坡配置寄存器（DAC_RAMPCFGR1） 260
14.4.17 DAC 通道 2 斜坡配置寄存器（DAC_RAMPCFGR2） 260
14.4.18 DAC 斜坡延迟配置寄存器（DAC_RAMPDLYS） 260
14.4.19 DAC 斜坡控制寄存器（DAC_RAMPCR） 261
14.4.20 DAC 时钟控制寄存器（DAC_CLKCR） 262
15. 电压参考缓冲器(V_{REFBUF}) 263
15.1 V_{REFBUF} 简介 263
15.2 V_{REFBUF} 功能描述 263
15.3 V_{REFBUF} 寄存器 263
15.3.1 V_{REFBUF} 控制寄存器（VREFBUF_CR） 263
16. 比较器（COMP） 264
16.1 COMP 简介 264

Puya Confidential

16.2 COMP 主要特性 264
16.3 COMP 功能描述 265
16.3.1 COMP 模块框图 265
16.3.2 COMP 的输入和输出 266
16.3.3 COMP 的时钟和复位 266
16.3.4 COMP 的滤波功能 267
16.3.5 COMP 的锁定机制 267
16.3.6 COMP 的输出消隐 267
16.3.7 COMP 的窗口模式 268
16.3.8 COMP 的迟滞功能 269
16.3.9 COMP 的低功耗模式 270
16.4 COMP 中断 270
16.5 COMP 寄存器 270
16.5.1 COMP1 控制和状态寄存器（COMP1_CSR） 270
16.5.2 COMP1 滤波寄存器（COMP1_FR） 272
16.5.3 COMP2 控制和状态寄存器（COMP2_CSR） 272
16.5.4 COMP2 滤波寄存器（COMP2_FR） 274
16.5.5 COMP3 控制和状态寄存器（COMP3_CSR） 274
16.5.6 COMP3 滤波寄存器（COMP3_FR） 276
16.5.7 COMP4 控制和状态寄存器（COMP4_CSR） 276
16.5.8 COMP4 滤波寄存器（COMP4_FR） 278
17. 运算放大器(OPA) 279
17.1 OPA 简介 279
17.2 OPA 主要特性 279
17.3 OPA 功能描述 279
17.3.1 模块框图 279
17.3.2 OPA 输出重定向到内部 ADC 通道 279
17.3.3 OPA 复位和时钟 279
17.3.4 初始配置 280
17.3.5 信号连接 280
17.3.6 OPA 模式 280
17.3.7 OPA 低功耗模式 287
17.4 OPA 寄存器 287
17.4.1 OPA1 控制/状态寄存器（OPA1_CSR） 287

Puya Confidential

17.4.2 OPA2 控制/状态寄存器（OPA2_CSR） 289
17.4.3 OPA3 控制/状态寄存器（OPA3_CSR） 291
17.4.4 OPA4 控制/状态寄存器（OPA4_CSR） 293
18. 增强型脉冲宽度调制器（EPWM） 295
18.1 EPWM 简介 295
18.2 EPWM 主要特征 295
18.2.1 结构框图 296
18.2.2 时基子模块功能特征 299
18.2.3 计数器比较子模块功能特征 300
18.2.4 动作限定子模块功能特征 300
18.2.5 死区发生器子模块功能特征 300
18.2.6 PWM 斩波子模块功能特征 301
18.2.7 跳闸区子模块功能特征 301
18.2.8 事件触发子模块功能特征 301
18.2.9 数字比较子模块功能特征 302
18.2.10 高分辨率功能特征 302
18.3 EPWM 功能描述 302
18.3.1 EPWM 引脚输入描述 303
18.3.2 时基（TB）子模块描述 307
18.3.3 计数器比较（CC）子模块描述 319
18.3.4 动作限定(AQ)子模块功能描述 324
18.3.5 死区发生器(DB)子模块功能描述 334
18.3.6 PWM 斩波（PC）子模块功能描述 339
18.3.7 跳闸区（TZ）子模块功能描述 342
18.3.8 事件触发（ET）子模块功能描述 347
18.3.9 数字比较(DC)子模块功能描述 351
18.3.10 高分辨率(HR)功能描述 358
18.3.11 寄存器保护机制描述 372
18.4 EPWMx 寄存器描述 377
18.4.1 时基控制寄存器（EPWMx_TBCTL）（x=1,2,3,4,5,6） 377
18.4.2 时基控制寄存器 2（EPWMx_TBCTL2）（x=1,2,3,4,5,6） 378
18.4.3 时基计数器寄存器（EPWMx_TBCTR）（x=1,2,3,4,5,6） 379
18.4.4 时基状态寄存器（EPWMx_TBSTS）（x=1,2,3,4,5,6） 379
18.4.5 计数器比较控制寄存器（EPWMx_CMPCTL）（x=1,2,3,4,5,6） 380

Puya Confidential

18.4.6 计数器比较控制寄存器 2（EPWMx_CMPCTL2）（x=1,2,3,4,5,6）.....381
18.4.7 死区发生器控制寄存器（EPWMx_DBCTL）（x=1,2,3,4,5,6）.....382
18.4.8 死区发生器控制寄存器 2（EPWMx_DBCTL2）（x=1,2,3,4,5,6）.....384
18.4.9 动作限定控制寄存器（EPWMx_AQCTL）（x=1,2,3,4,5,6）.....385
18.4.10 选择动作限定触发器事件源寄存器（EPWMx_AQTSRCSEL）（x=1,2,3,4,5,6）.....386
18.4.11 PWM 斩波控制寄存器（EPWMx_PCCTL）（x=1,2,3,4,5,6）.....387
18.4.12 谷捕获控制寄存器（EPWMx_VCAPCTL）（x=1,2,3,4,5,6）.....388
18.4.13 各计数器配置寄存器（EPWMx_VCNTCFG）（x=1,2,3,4,5,6）.....389
18.4.14 EPWM 配置寄存器（EPWMx_HRCNFG）（x=1,2,3,4,5,6）.....390
18.4.15 使能 MEP 校准逻辑寄存器（EPWMx_HRPWR）（x=1,2,3,4,5,6）.....392
18.4.16 MEP 步骤数寄存器（EPWMx_HRMSTEP）（x=1,2,3,4,5,6）.....392
18.4.17 EPWM 配置寄存器 2（EPWMx_HRCNFG2）（x=1,2,3,4,5,6）.....392
18.4.18 高分辨率周期控制寄存器（EPWMx_HRPCTL）（x=1,2,3,4,5,6）.....393
18.4.19 EPWM 转换高分辨率余数寄存器（EPWMx_TRREM）（x=1,2,3,4,5,6）.....394
18.4.20 全局 EPWM 装载控制寄存器（EPWMx_GLDCTL）（x=1,2,3,4,5,6）.....395
18.4.21 全局 EPWM 装载事件配置寄存器（EPWMx_GLDCFG）（x=1,2,3,4,5,6）.....396
18.4.22 EPWMx 链接寄存器（EPWMx_EPWMXLINK）（x=1,2,3,4,5,6）.....397
18.4.23 输出 A 的动作限定控制寄存器（EPWMx_AQCTLA）（x=1,2,3,4,5,6）.....399
18.4.24 输出 A 的附加动作限定控制寄存器（EPWMx_AQCTLA2）（x=1,2,3,4,5,6）.....399
18.4.25 输出 B 的动作限定控制寄存器（EPWMx_AQCTLB）（x=1,2,3,4,5,6）.....400
18.4.26 输出 B 的附加动作限定控制寄存器（EPWMx_AQCTLB2）（x=1,2,3,4,5,6）.....401
18.4.27 动作限定软件强制寄存器（EPWMx_AQSFRC）（x=1,2,3,4,5,6）.....402
18.4.28 动作限定连续软件强制寄存器（EPWMx_AQCSFRC）（x=1,2,3,4,5,6）.....402
18.4.29 死区发生器上升沿延迟高分辨率寄存器（EPWMx_DBREDHR）（x=1,2,3,4,5,6）.....403
18.4.30 死区发生器上升沿延迟寄存器（EPWMx_DBRED）（x=1,2,3,4,5,6）.....403
18.4.31 死区发生器下降沿延迟高分辨率寄存器（EPWMx_DBFEDHR）（x=1,2,3,4,5,6）.....403
18.4.32 死区发生器下降沿延迟寄存器（EPWMx_DBFED）（x=1,2,3,4,5,6）.....404
18.4.33 时基相位偏移寄存器（EPWMx_TBPHS）（x=1,2,3,4,5,6）.....404
18.4.34 高分辨率时基周期寄存器（EPWMx_TBPRDHR）（x=1,2,3,4,5,6）.....404
18.4.35 时基周期寄存器（EPWMx_TBPRD）（x=1,2,3,4,5,6）.....405
18.4.36 计数比较模块 A 寄存器（EPWMx_CMPA）（x=1,2,3,4,5,6）.....405
18.4.37 计数比较模块 B 寄存器（EPWMx_CMPB）（x=1,2,3,4,5,6）.....406
18.4.38 计数比较模块 C 寄存器（EPWMx_CMPC）（x=1,2,3,4,5,6）.....407
18.4.39 计数比较模块 D 寄存器（EPWMx_CMPD ）（x=1,2,3,4,5,6）.....407

Puya Confidential

18.4.40 全局 EPWM 装载控制寄存器 2（EPWMx_GLDCTL2）（x=1,2,3,4,5,6）.....407
18.4.41 软件谷模式延迟寄存器（EPWMx_SWVDELVAL）（x=1,2,3,4,5,6）.....408
18.4.42 跳闸区选择寄存器（EPWMx_TZSEL）（x=1,2,3,4,5,6）.....408
18.4.43 跳闸区数字比较器配置寄存器（EPWMx_TZDCSEL）（x=1,2,3,4,5,6）.....409
18.4.44 跳闸区控制寄存器（EPWMx_TZCTL）（x=1,2,3,4,5,6）.....410
18.4.45 跳闸区控制寄存器 2（EPWMx_TZCTL2）（x=1,2,3,4,5,6）.....411
18.4.46 跳闸区控制数字比较器 A 寄存器（EPWMx_TZCTLDCA）（x=1,2,3,4,5,6）.....412
18.4.47 跳闸区控制数字比较器 B 寄存器（EPWMx_TZCTLDCB）（x=1,2,3,4,5,6）.....413
18.4.48 跳闸区中断使能寄存器（EPWMx_TZEINT）（x=1,2,3,4,5,6）.....413
18.4.49 跳闸区标志寄存器（EPWMx_TZFLG）（x=1,2,3,4,5,6）.....414
18.4.50 跳闸区 CBC 标志寄存器（EPWMx_TZCBCFLG）（x=1,2,3,4,5,6）.....415
18.4.51 跳闸区 OST 标志寄存器（EPWMx_TZOSTFLG）（x=1,2,3,4,5,6）.....415
18.4.52 清除跳闸区寄存器（EPWMx_TZCLR）（x=1,2,3,4,5,6）.....416
18.4.53 清除跳闸区 CBC 寄存器（EPWMx_TZCBCCLR）（x=1,2,3,4,5,6）.....417
18.4.54 清除跳闸区 OST 寄存器（EPWMx_TZOSTCLR）（x=1,2,3,4,5,6）.....418
18.4.55 强制跳闸区标志寄存器（EPWMx_TZFRC）（x=1,2,3,4,5,6）.....418
18.4.56 事件触发选择寄存器（EPWMx_ETSEL）（x=1,2,3,4,5,6）.....419
18.4.57 事件触发预分配寄存器（EPWMx_ETPS）（x=1,2,3,4,5,6）.....421
18.4.58 事件触发标志寄存器（EPWMx_ETFLG）（x=1,2,3,4,5,6）.....422
18.4.59 清除事件触发寄存器（EPWMx_ETCLR）（x=1,2,3,4,5,6）.....423
18.4.60 强制事件触发寄存器（EPWMx_ETFRC）（x=1,2,3,4,5,6）.....423
18.4.61 事件触发中断预分频寄存器（EPWMx_ETINTPS）（x=1,2,3,4,5,6）.....424
18.4.62 事件触发 SOC 预分频寄存器（EPWMx_ETSOCPS）（x=1,2,3,4,5,6）.....424
18.4.63 事件触发计数器初始化控制寄存器（EPWMx_ETCNTINITCTL）（x=1,2,3,4,5,6）.....425
18.4.64 事件触发计数器初始化寄存器（EPWMx_ETCNTINIT）（x=1,2,3,4,5,6）.....426
18.4.65 数字比较跳闸选择寄存器（EPWMx_DCTRIPSEL）（x=1,2,3,4,5,6）.....426
18.4.66 数字比较 A 控制寄存器（EPWMx_DCACTL）（x=1,2,3,4,5,6）.....427
18.4.67 数字比较 B 控制寄存器（EPWMx_DCBCTL）（x=1,2,3,4,5,6）.....427
18.4.68 数字比较滤波器控制寄存器（EPWMx_DCFCTL）（x=1,2,3,4,5,6）.....428
18.4.69 数字比较捕获控制寄存器（EPWMx_DCCAPCTL）（x=1,2,3,4,5,6）.....429
18.4.70 数字比较滤波器偏移寄存器（EPWMx_DCFOFFSET）（x=1,2,3,4,5,6）.....430
18.4.71 数字比较滤波器偏移计数器寄存器（EPWMx_DCFOFFSETCNT）（x=1,2,3,4,5,6）..430
18.4.72 数字比较滤波器窗口寄存器（EPWMx_DCFWINDOW）（x=1,2,3,4,5,6）.....431
18.4.73 数字比较滤波器窗口计数器寄存器（EPWMx_DCFWINDOWCNT）（x=1,2,3,4,5,6）431

Puya Confidential

18.4.74 数字比较计数器捕获寄存器（EPWMx_DCCAP）（x=1,2,3,4,5,6）.....431
18.4.75 数字比较 A 高跳闸选择寄存器（EPWMx_DCAHTRIPSEL）（x=1,2,3,4,5,6）.....432
18.4.76 数字比较 A 低跳闸选择寄存器（EPWMx_DCALTRIPSEL）（x=1,2,3,4,5,6）.....433
18.4.77 数字比较 B 高跳闸选择寄存器（EPWMx_DCBHTRIPSEL）（x=1,2,3,4,5,6）.....434
18.4.78 数字比较 B 低跳闸选择寄存器（EPWMx_DCBLTRIPSEL）（x=1,2,3,4,5,6）.....435
18.4.79 EPWM 锁定寄存器（EPWMx_EPWMLOCK）（x=1,2,3,4,5,6）.....436
18.4.80 硬件谷模式延迟寄存器（EPWMx_HWVDELVAL）（x=1,2,3,4,5,6）.....437
18.4.81 硬件谷计数器寄存器（EPWMx_VCNTVAL）（x=1,2,3,4,5,6）.....437
18.5 EPWM 共用寄存器描述.....437
18.5.1 INPUTXBAR1/2 选择寄存器（EPWM_IXBSEL1_2）.....437
18.5.2 INPUTXBAR3/4 选择寄存器（EPWM_IXBSEL3_4）.....438
18.5.3 INPUTXBAR5/6 选择寄存器（EPWM_IXBSEL5_6）.....438
18.5.4 INPUTXBAR7/8 选择寄存器（EPWM_IXBSEL7_8）.....438
18.5.5 INPUTXBAR9/10 选择寄存器（EPWM_IXBSEL9_10）.....439
18.5.6 INPUTXBAR11/12 选择寄存器（EPWM_IXBSEL11_12）.....439
18.5.7 INPUTXBAR13/14 选择寄存器（EPWM_IXBSEL13_14）.....439
18.5.8 INPUTXBAR 输出动作选择寄存器（EPWM_IXBAQSEL）.....440
18.5.9 INPUTXBAR 选择锁定寄存器（EPWM_IXBSEL_LOCK）.....440
18.5.10 INPUTXBAR 动作采样周期控制寄存器（EPWM_IXBCTRL）.....441
18.5.11 HRPMWXBAR4 MUX 选择寄存器 0（EPWM_EPXB4_0）.....442
18.5.12 HRPMWXBAR4 MUX 选择寄存器 1（EPWM_EPXB4_1）.....443
18.5.13 HRPMWXBAR5 MUX 选择寄存器 0（EPWM_EPXB5_0）.....443
18.5.14 HRPMWXBAR5 MUX 选择寄存器 1（EPWM_EPXB5_1）.....444
18.5.15 HRPMWXBAR7 MUX 选择寄存器 0（EPWM_EPXB7_0）.....445
18.5.16 HRPMWXBAR7 MUX 选择寄存器 1（EPWM_EPXB7_1）.....446
18.5.17 HRPMWXBAR8 MUX 选择寄存器 0（EPWM_EPXB8_0）.....447
18.5.18 HRPMWXBAR8 MUX 选择寄存器 1（EPWM_EPXB8_1）.....447
18.5.19 HRPMWXBAR9 MUX 选择寄存器 0（EPWM_EPXB9_0）.....448
18.5.20 HRPMWXBAR9 MUX 选择寄存器 1（EPWM_EPXB9_1）.....449
18.5.21 HRPMWXBAR10 MUX 选择寄存器 0（EPWM_EPXB10_0）.....449
18.5.22 HRPMWXBAR10 MUX 选择寄存器 1（EPWM_EPXB10_1）.....450
18.5.23 HRPMWXBAR11 MUX 选择寄存器 0（EPWM_EPXB11_0）.....451
18.5.24 HRPMWXBAR11 MUX 选择寄存器 1（EPWM_EPXB11_1）.....452
18.5.25 HRPMWXBAR12 MUX 选择寄存器 0（EPWM_EPXB12_0）.....452

Puya Confidential

18.5.26 HRPMWXBAR12 MUX 选择寄存器 1（EPWM_EPXB12_1）.....453
18.5.27 EPWMXBAR4 MUX 使能寄存器（EPWM_EPXB4_MUX）.....454
18.5.28 EPWMXBAR5 MUX 使能寄存器（EPWM_EPXB5_MUX）.....454
18.5.29 EPWMXBAR7 MUX 使能寄存器（EPWM_EPXB7_MUX）.....454
18.5.30 EPWMXBAR8 MUX 使能寄存器（EPWM_EPXB8_MUX）.....455
18.5.31 EPWMXBAR9 MUX 使能寄存器（EPWM_EPXB9_MUX）.....455
18.5.32 EPWMXBAR10 MUX 使能寄存器（EPWM_EPXB10_MUX）.....455
18.5.33 EPWMXBAR11 MUX 使能寄存器（EPWM_EPXB11_MUX）.....456
18.5.34 EPWMXBAR12 MUX 使能寄存器（EPWM_EPXB12_MUX）.....456
18.5.35 EPWMXBAR 输出极性选择寄存器（EPWM_EPXB_POL）.....456
18.5.36 HRPMMXBAR 锁定寄存器（EPWM_EPXBLOCK）.....457
18.5.37 EPWM1_4 SYNCI 选择寄存器（EPWM_SYNCI_SEL1_4）.....457
18.5.38 EPWM5_6 SYNCI 选择寄存器（EPWM_SYNCI_SEL5_6）.....458
18.5.39 EPWM1_4 SYNCO 选择寄存器（EPWM_SYNCO_SEL1_4）.....458
18.5.40 EPWM5_6 SYNCO 选择寄存器（EPWM_SYNCO_SEL5_6）.....458
18.5.41 外部 ADC 控制寄存器（EPWM_ADCOUT）.....459
18.5.42 EPWM_SCOUT 配置寄存器（EPWM_SCOUT_CFG）.....460
18.5.43 HRPMW SYNCSOC 锁定寄存器（EPWM_SYNCSOCLOCK）.....460
18.5.44 EALLOW 寄存器（EPWM_EALLOW）.....461
18.5.45 EPWM 时钟使能寄存器（EPWM_CLKEN）.....461
18.5.46 XBAR 状态寄存器 1（EPWM_XBARFLG1）.....462
18.5.47 XBAR 状态寄存器 2（EPWM_XBARFLG2）.....464
18.5.48 XBAR 状态清除寄存器 1（EPWM_XBARCLR1）.....466
18.5.49 XBAR 状态清除寄存器 2（EPWM_XBARCLR2）.....467

19. 高级定时器（TIM1/TIM8）.....468

19.1 TIM1/TIM8 简介.....468
19.2 TIM1/TIM8 主要特征.....468
19.3 TIM1/TIM8 功能描述.....469
19.3.1 TIM1/TIM8 模块框图.....469
19.3.2 时基单元.....469
19.3.3 计数器模式.....471
19.3.4 重复计数器.....478
19.3.5 外部触发输入.....479
19.3.6 时钟选择.....479

Puya Confidential

19.3.7 捕获/比较通道 482
19.3.8 输入捕获模式 484
19.3.9 PWM 输入模式 484
19.3.10 强置输出模式 485
19.3.11 输出比较模式 485
19.3.12 PWM 模式 486
19.3.13 互补输出和死区插入 490
19.3.14 使用刹车功能 492
19.3.15 双向刹车输入 496
19.3.16 在外部事件时清除 OCxREF 信号 497
19.3.17 产生六步 PWM 输出 498
19.3.18 单脉冲模式 499
19.3.19 可重触发的单脉冲模式 500
19.3.20 定时器输入异或功能 500
19.3.21 与霍尔传感器的接口 500
19.3.22 TIMx 定时器和外部触发的同步 502
19.3.23 DMA 突发模式 505
19.3.24 TIM1/TIM8 DMA 请求 506
19.3.25 定时器同步 506
19.3.26 调试模式 506
19.3.27 计数器多点触发 ADC 506
19.4 TIM1/TIM8 中断 507
19.5 TIM1/TIM8 寄存器 507
19.5.1 TIMx 控制寄存器 1（TIMx_CR1）（x=1/8） 507
19.5.2 TIMx 控制寄存器 2（TIMx_CR2）（x=1/8） 509
19.5.3 TIMx 从模式控制寄存器（TIMx_SMCR）（x=1/8） 511
19.5.4 TIMx DMA/中断使能寄存器（TIMx_DIER）（x=1/8） 514
19.5.5 TIMx 状态寄存器（TIMx_SR）（x=1/8） 515
19.5.6 TIMx 事件产生寄存器（TIMx_EGR）（x=1/8） 518
19.5.7 TIMx 捕获/比较模式控制寄存器 1（TIMx_CCMR1）（x=1/8） 519
19.5.8 TIMx 捕获/比较模式控制寄存器 2（TIMx_CCMR2）（x=1/8） 523
19.5.9 TIMx 捕获/比较使能寄存器（TIMx_CCER）（x=1/8） 525
19.5.10 TIMx 计数器（TIMx_CNT）（x=1/8） 527
19.5.11 TIMx 预分频器（TIMx_PSC）（x=1/8） 528

Puya Confidential

19.5.12 TIMx 自动重装载寄存器（TIMx_ARR）（x=1/8）.....528
19.5.13 TIMx 重复计数寄存器（TIMx_RCR）（x=1/8）.....528
19.5.14 TIMx 捕获/比较寄存器 1（TIMx_CCR1）（x=1/8）.....529
19.5.15 TIMx 捕获/比较寄存器 2（TIMx_CCR2）（x=1/8）.....529
19.5.16 TIMx 捕获/比较寄存器 3（TIMx_CCR3）（x=1/8）.....530
19.5.17 TIMx 捕获/比较寄存器 4（TIMx_CCR4）（x=1/8）.....530
19.5.18 TIMx 刹车和死区寄存器（TIMx_BDTR）（x=1/8）.....531
19.5.19 TIMx 捕获/比较寄存器 5（TIMx_CCR5）（x=1/8）.....535
19.5.20 TIMx 捕获/比较寄存器 6（TIMx_CCR6）（x=1/8）.....535
19.5.21 TIMx 捕获/比较模式控制寄存器 3（TIMx_CCMR3）（x=1/8）.....536
19.5.22 TIMx 死区时间寄存器 2（TIMx_DTR2）（x=1/8）.....537
19.5.23 TIMx 输入选择寄存器（TIMx_TISEL）（x=1/8）.....538
19.5.24 TIMx 备用功能选项寄存器 1（TIMx_AF1）（x=1/8）.....539
19.5.25 TIMx 备用功能选项寄存器 2（TIMx_AF2）（x=1/8）.....541
19.5.26 TIMx 多点触发 ADC 控制寄存器（TIMx_MTADCCR）（x=1/8）.....543
19.5.27 TIMx DMA 控制寄存器（TIMx_DCR）（x=1/8）.....545
19.5.28 TIMx 连续模式的 DMA 地址（TIMx_DMAR）（x=1/8）.....546

20. 通用定时器（TIM2/TIM3）.....547

20.1 TIM2/TIM3 简介.....547
20.2 TIM2/TIM3 主要特征.....547
20.3 TIM2/TIM3 功能描述.....548
20.3.1 TIM2/TIM3 模块框图.....548
20.3.2 时基单元.....548
20.3.3 计数器模式.....550
20.3.4 外部触发输入.....557
20.3.5 时钟选择.....557
20.3.6 捕获/比较通道.....560
20.3.7 输入捕获模式.....561
20.3.8 PWM 输入模式.....562
20.3.9 强置输出模式.....563
20.3.10 输出比较模式.....563
20.3.11 PWM 模式.....564
20.3.12 互补输出和死区插入.....566
20.3.13 使用刹车功能.....568

Puya Confidential

20.3.14 双向刹车输入 572
20.3.15 在外部事件时清除 OCxREF 信号 573
20.3.16 单脉冲模式 574
20.3.17 编码器接口模式 575
20.3.18 定时器输入异或功能 577
20.3.19 TIMx 定时器和外部触发的同步 577
20.3.20 定时器同步 579
20.3.21 DMA 突发模式 584
20.3.22 TIM2/3DMA 请求 585
20.3.23 调试模式 585
20.3.24 TIM2/3 低功耗模式 585
20.3.25 计数器多点触发 ADC 585
20.4 TIM2/3 中断 586
20.5 TIM2/3 寄存器 586
20.5.1 TIMx 控制寄存器 1（TIMx_CR1）（x=2/3） 586
20.5.2 TIMx 控制寄存器 2（TIMx_CR2）（x=2/3） 588
20.5.3 TIMx 从模式控制寄存器（TIMx_SMCR）（x=2/3） 590
20.5.4 TIMx DMA/中断使能寄存器（TIMx_DIER）（x=2/3） 592
20.5.5 TIMx 状态寄存器（TIMx_SR）（x=2/3） 594
20.5.6 TIMx 事件产生寄存器（TIMx_EGR）（x=2/3） 596
20.5.7 TIMx 捕获/比较模式控制寄存器 1（TIMx_CCMR1）（x=2/3） 598
20.5.8 TIMx 捕获/比较模式控制寄存器 2（TIMx_CCMR2）（x=2/3） 601
20.5.9 TIMx 捕获/比较使能寄存器（TIMx_CCER）（x=2/3） 602
20.5.10 TIMx 计数器（TIMx_CNT）（x=2） 603
20.5.11 TIMx 计数器（TIMx_CNT）（x=3） 604
20.5.12 TIMx 预分频器（TIMx_PSC）（x=2/3） 604
20.5.13 TIMx 自动重装载寄存器（TIMx_ARR）（x=2） 604
20.5.14 TIMx 自动重装载寄存器（TIMx_ARR）（x=3） 605
20.5.15 TIMx 捕获/比较寄存器 1（TIMx_CCR1）（x=2） 605
20.5.16 TIMx 捕获/比较寄存器 1（TIMx_CCR1）（x=3） 605
20.5.17 TIMx 捕获/比较寄存器 2（TIMx_CCR2）（x=2） 606
20.5.18 TIMx 捕获/比较寄存器 2（TIMx_CCR2）（x=3） 606
20.5.19 TIMx 捕获/比较寄存器 3（TIMx_CCR3）（x=2） 607
20.5.20 TIMx 捕获/比较寄存器 3（TIMx_CCR3）（x=3） 607

Puya Confidential

20.5.21 TIMx 捕获/比较寄存器 4（TIMx_CCR4）（x=2） 607
20.5.22 TIMx 捕获/比较寄存器 4（TIMx_CCR4）（x=3） 608
20.5.23 TIMx 刹车和死区寄存器（TIMx_BDTR）（x=2/3） 608
20.5.24 TIMx 死区时间寄存器 2（TIMx_DTR2）（x=2/3） 612
20.5.25 TIM_TYP1 编码器控制寄存器（TIMx_ECR）（x=2/3） 613
20.5.26 TIMx 输入选择寄存器（TIMx_TISEL）（x=2/3） 614
20.5.27 TIMx 备用功能选项寄存器 1（TIMx_AF1）（x=2/3） 615
20.5.28 TIMx 备用功能选项寄存器 2（TIMx_AF2）（x=2/3） 617
20.5.29 TIMx 多点触发 ADC 控制寄存器（TIMx_MTADCCR）（x=2/3） 619
20.5.30 TIMx DMA 控制寄存器（TIMx_DCR）（x=2/3） 621
20.5.31 TIMx 连续模式的 DMA 地址（TIMx_DMAR）（x=2/3） 622
21. 基本定时器（TIM6/TIM7） 623
21.1 TIM6/TIM7 简介 623
21.2 TIM6/TIM7 主要特征 623
21.3 TIM6/TIM7 功能描述 623
21.3.1 TIM6/TIM7 模块框图 623
21.3.2 时基单元 624
21.3.3 计数器模式 625
21.4 TIM6/TIM7 调试模式 628
21.5 TIM6/TIM7 寄存器 628
21.5.1 TIMx 控制寄存器 1（TIMx_CR1）（x=6/7） 628
21.5.2 TIMx 控制寄存器 2（TIMx_CR2）（x=6/7） 629
21.5.3 TIMx DMA/中断使能寄存器（TIMx_DIER）（x=6/7） 629
21.5.4 TIMx 状态寄存器（TIMx_SR）（x=6/7） 630
21.5.5 TIMx 事件产生寄存器（TIMx_EGR）（x=6/7） 630
21.5.6 TIMx 计数寄存器（TIMx_CNT）（x=6/7） 630
21.5.7 TIMx 预分频寄存器（TIMx_PSC）（x=6/7） 631
21.5.8 TIMx 自动重载寄存器（TIMx_ARR）（x=6/7） 631
22. 低功耗定时器（LPTIM） 633
22.1 LPTIM 简介 633
22.2 LPTIM 主要特性 633
22.3 LPTIM 功能描述 634
22.3.1 模块框图 634

Puya Confidential

22.3.2 复位和时钟 634
22.3.3 毛刺滤波器 634
22.3.4 预分频 635
22.3.5 触发选择 635
22.3.6 操作模式 635
22.3.7 波形产生 637
22.3.8 寄存器更新 637
22.3.9 计数器模式 638
22.3.10 定时器使能 638
22.3.11 定时器复位 638
22.3.12 编码器模式 639
22.3.13 调试模式 639
22.4 LPTIM 低功耗模式 640
22.5 LPTIM 中断 640
22.6 LPTIM 寄存器 641
22.6.1 LPTIM 中断和状态寄存器（LPTIM_ISR） 641
22.6.2 LPTIM 中断清零寄存器（LPTIM_ICR） 642
22.6.3 LPTIM 中断使能寄存器（LPTIM_IER） 642
22.6.4 LPTIM 配置寄存器（LPTIM_CFGR） 643
22.6.5 LPTIM 控制寄存器（LPTIM_CR） 646
22.6.6 LPTIM 比较寄存器（LPTIM_CMP） 646
22.6.7 LPTIM 自动重载寄存器（LPTIM_ARR） 647
22.6.8 LPTIM 计数器寄存器（LPTIM_CNT） 647
22.6.9 LPTIM 选择寄存器（LPTIM_OR） 648
23. 实时时钟（RTC） 649
23.1 RTC 简介 649
23.2 RTC 主要特性 649
23.3 RTC 功能描述 650
23.3.1 概述 650
23.3.2 寄存器复位 651
23.3.3 读 RTC 寄存器 651
23.3.4 配置 RTC 寄存器 651
23.3.5 RTC 标志设置 652
23.3.6 RTC 时序 652

Puya Confidential

23.3.7 侵入检测 653
23.3.8 RTC 校准 653
23.3.9 备份域寄存器擦除 656
23.4 RTC 寄存器 657
23.4.1 RTC 控制寄存器高位（RTC_CRH） 657
23.4.2 RTC 控制寄存器低位（RTC_CRL） 657
23.4.3 RTC 预分频装载寄存器高位（RTC_PRLH） 659
23.4.4 RTC 预分频装载寄存器低位（RTC_PRLL） 659
23.4.5 RTC 预分频余数寄存器高位（RTC_DIVH） 659
23.4.6 RTC 预分频余数寄存器低位（RTC_DIVL） 660
23.4.7 RTC 计数寄存器高位（RTC_CNTH） 660
23.4.8 RTC 计数寄存器低位（RTC_CNTL） 661
23.4.9 RTC 闹钟寄存器高位（RTC_ALRH） 661
23.4.10 RTC 闹钟寄存器低位(RTC_ALRL) 661
23.4.11 RTC 入侵配置寄存器（RTC_TMPCFGR） 662
23.4.12 RTC 入侵控制/状态寄存器（RTC_TMPCSR） 662
23.4.13 RTC 时钟校准寄存器(RTC_CALIBR) 663
23.4.14 RTC 备份寄存器（RTC_BKPDR） 664
24. 独立看门狗（IWDG） 665
24.1 IWDG 简介 665
24.2 IWDG 主要特性 665
24.3 IWDG 功能说明 665
24.3.1 模块框图 665
24.3.2 硬件看门狗 666
24.3.3 寄存器保护 666
24.3.4 调试模式 666
24.3.5 低功耗冻结 666
24.4 IWDG 寄存器 666
24.4.1 IWDG 键值寄存器（IWDG_KR） 666
24.4.2 IWDG 预分频寄存器（IWDG_PR） 667
24.4.3 IWDG 重装载寄存器（IWDG_RLR） 667
24.4.4 IWDG 状态寄存器（IWDG_SR） 668
25. 窗口看门狗（WWDG） 669

Puya Confidential

25.1 WWDG 简介 669
25.2 WWDG 主要特性 669
25.3 WWDG 功能描述 669
25.4 WWDG 超时程序 670
25.5 WWDG 调试模式 671
25.6 WWDG 寄存器 671
25.6.1 WWDG 控制寄存器（WWDG_CR） 671
25.6.2 WWDG 配置寄存器（WWDG_CFR） 671
25.6.3 WWDG 状态寄存器（WWDG_SR） 672
26. 串行外设接口（SPI） 673
26.1 SPI 简介 673
26.2 SPI 主要特征 673
26.3 SPI 功能描述 674
26.3.1 SPI 框图 674
26.3.2 一个主机和一个从机的通信 674
26.3.3 标准多从机通信 676
26.3.4 多主机通信 677
26.3.5 从机选择（NSS）引脚管理 678
26.3.6 通讯格式 679
26.3.7 SPI 配置 680
26.3.8 使能 SPI 流程 681
26.3.9 数据发送和接收 681
26.3.10 状态标准 685
26.3.11 错误标志 686
26.3.12 TI 模式 687
26.3.13 SPI CRC 687
26.4 SPI 中断 689
26.5 SPI 寄存器 689
26.5.1 SPI 控制寄存器 1（SPI_CR1） 689
26.5.2 SPI 控制寄存器 2（SPI_CR2） 691
26.5.3 SPI 状态寄存器（SPI_SR） 692
26.5.4 SPI 数据寄存器（SPI_DR） 694
26.5.5 SPI CRC 多项式寄存器（SPI_CRCPR） 694
26.5.6 SPI 接收 CRC 寄存器（SPI_RXCRC） 695

Puya Confidential

26.5.7 SPI 发送 CRC 寄存器（SPI_TXCRC）.....695
27. 内部集成电路接口（I²C）.....697
27.1 I²C 简介.....697
27.2 I²C 主要特性.....697
27.3 I²C 功能描述.....699
27.3.1 简介.....699
27.3.2 时钟.....699
27.3.3 包错误校验（PEC）.....699
27.3.4 I2C 的从模式.....700
27.3.5 I2C 的主模式.....702
27.3.6 错误标志.....708
27.3.7 DMA 功能.....709
27.3.8 支持从 Stop 模式唤醒.....710
27.3.9 系统管理总线 SMbus.....710
27.4 I²C 中断.....713
27.5 I²C 寄存器描述.....713
27.5.1 I²C 控制寄存器 1（I2C_CR1）.....713
27.5.2 I²C 控制寄存器 2（I2C_CR2）.....716
27.5.3 I²C 自身地址寄存器 1（I2C_OAR1）.....717
27.5.4 I2C 自身地址寄存器 2（I2C_OAR2）.....718
27.5.5 I²C 数据寄存器（I2C_DR）.....719
27.5.6 I²C 状态寄存器 1（I2C_SR1）.....719
27.5.7 I²C 状态寄存器 2（I2C_SR2）.....723
27.5.8 I²C 时钟控制寄存器（I2C_CCR）.....724
27.5.9 I²C 上升时间寄存器（I2C_TRISE）.....725
27.5.10 I²C 超时寄存器（I2C_TIMEOUTR）.....726
27.5.11 I2C 自身地址寄存器 3（I2C_OAR3）.....727
28. 通用同步收发器(USART).....728
28.1 USART 简介.....728
28.2 USART 主要特征.....728
28.3 USART 功能描述.....729
28.3.1 USART 功能框图.....729
28.3.2 USART 输入/输出引脚说明.....729

Puya Confidential

28.3.3 USART 时钟 730
28.3.4 USART 字符说明 730
28.3.5 USART FIFO 和阈值 732
28.3.6 USART 发送器 732
28.3.7 USAET 接收器 735
28.3.8 波特率生成 740
28.3.9 USART 接收容忍度 741
28.3.10 USART 自动波特率检测 742
28.3.11 USART 多处理器通信 743
28.3.12 Modbus 通信 744
28.3.13 奇偶校验控制 745
28.3.14 LIN（局域互联网）模式 746
28.3.15 USART 同步模式 748
28.3.16 USART 单线半双工通信 750
28.3.17 接收器超时 750
28.3.18 智能卡 750
28.3.19 IrDA SIR ENDEC 功能模块 752
28.3.20 DMA 连续通讯 754
28.3.21 多处理器通讯错误标志和中断产生 756
28.3.22 硬件流控制 756
28.4 USART 中断 758
28.5 USART 寄存器 758
28.5.1 USART 状态寄存器（USART_SR） 758
28.5.2 USART 数据寄存器（USART_DR） 762
28.5.3 USART 波特率寄存器（USART_BRR） 762
28.5.4 USART 控制寄存器 1（USART_CR1） 763
28.5.5 USART 控制寄存器 2（USART_CR2） 766
28.5.6 USART 控制寄存器 3 (USART_CR3) 768
28.5.7 USART 保护时间和预分频寄存器（USART_GTPR) 770
28.5.8 USART 接收超时寄存器（USART_RTOR） 771
29. 通用异步收发器（UART） 773
29.1 UART 简介 773
29.2 UART 主要特性 773
29.3 UART 功能描述 774

Puya Confidential

29.3.1 UART 框图 774
29.3.2 UART（RS232）串行协议 774
29.3.3 UART 9 位数据传输 776
29.3.4 UART 波特率 778
29.3.5 UART 接收容忍度 779
29.3.6 UART DMA 通信 779
29.4 UART 中断 780
29.5 UART 寄存器 781
29.5.1 UART 数据寄存器（UART_DR） 781
29.5.2 UART 波特率寄存器（UART_BRR） 782
29.5.3 UART 状态寄存器（UART_SR） 782
29.5.4 UART 控制寄存器 1（UART_CR1） 785
29.5.5 UART 控制寄存器 2（UART_CR2） 786
29.5.6 UART 控制寄存器 3（UART_CR3） 787
29.5.7 UART 接收地址寄存器（UART_RAR） 788
29.5.8 UART 发送地址寄存器（UART_TAR） 789
29.5.9 UART 波特率小数寄存器（UART_BRRF） 789
30. 低功耗通用异步收发器（LPUART） 790
30.1 LPUART 主要特性 790
30.2 LPUART 功能描述 791
30.2.1 LPUART 框图 791
30.2.2 LPUART 信号 791
30.2.3 LPUART 字符描述 792
30.2.4 LPUART 发送 794
30.2.5 LPUART 接收 795
30.2.6 LPUART 时钟源选择 796
30.2.7 LPUART 波特率产生 796
30.2.8 LPUART 接收容忍度 797
30.2.9 LPUART 多处理器通信 798
30.2.10 LPUART 奇偶校验 799
30.2.11 LPUART 单线半双工通信 799
30.2.12 DMA 连续通信 800
30.2.13 RS232 硬件流控制和 RS485 驱动器使能 802
30.2.14 LPUART 低功耗模式 804

Puya Confidential

30.3 LPUART 中断 805
30.4 LPUART 寄存器 806
30.4.1 LPUART 控制寄存器 1（LPUART_CR1） 806
30.4.2 LPUART 控制寄存器 2（LPUART_CR2） 808
30.4.3 LPUART 控制寄存器 3（LPUART_CR3） 809
30.4.4 LPUART 波特率寄存器（LPUART_BRR） 811
30.4.5 LPUART 请求寄存器（LPUART_RQR） 811
30.4.6 LPUART 中断和状态寄存器（LPUART_ISR） 812
30.4.7 LPUART 中断标志清零寄存器（LPUART_ICR） 814
30.4.8 LPUART 接收数据寄存器（LPUART_RDR） 815
30.4.9 LPUART 发送数据寄存器（LPUART_TDR） 815
30.4.10 LPUART 预分频器寄存器（LPUART_PRESC） 815
30.4.11 LPUART 采样模式选择寄存器（LPUART_MOD） 816
31. FD 控制器局域网路(FDCAN) 817
31.1 FDCAN 简介 817
31.2 FDCAN 主要特性 817
31.3 FDCAN 功能描述 818
31.3.1 FDCAN 框图 818
31.3.2 FDCAN 工作模式 819
31.3.3 FDCAN 时间戳生成 824
31.3.4 FDCAN 超时计数器 825
31.3.5 FDCAN 接收处理 825
31.3.6 FDCAN 发送处理 831
31.3.7 FDCAN FIFO 确认处理 835
31.3.8 FDCAN 消息 RAM 836
31.4 FDCAN 软件初始化 843
31.5 FDCAN 寄存器 844
31.5.1 FDCAN 字节序寄存器（FDCAN_ENDN） 844
31.5.2 FDCAN 数据位时间和预分频寄存器（FDCAN_DBTP） 844
31.5.3 FDCAN 测试寄存器（FDCAN_TEST） 845
31.5.4 FDCAN RAM 看门狗寄存器（FDCAN_RWD） 846
31.5.5 FDCAN CC 控制寄存器（FDCAN_CCCR） 847
31.5.6 FDCAN 标称位时间和预分频寄存器（FDCAN_NBTP） 848
31.5.7 FDCAN 时间戳计数器配置寄存器（FDCAN_TSCC） 849

Puya Confidential

31.5.8 FDCAN 时间戳计数器值寄存器（FDCAN_TSCV）.....850
31.5.9 FDCAN 超时计数器配置寄存器（FDCAN_TOCC）.....850
31.5.10 FDCAN 超时计数器值寄存器（FDCAN_TOCV）.....851
31.5.11 FDCAN 错误计数器寄存器（FDCAN_ECR）.....851
31.5.12 FDCAN 协议状态寄存器（FDCAN_PSR）.....852
31.5.13 FDCAN 发送器延迟补偿寄存器（FDCAN_TDCR）.....854
31.5.14 FDCAN 中断寄存器（FDCAN_IR）.....855
31.5.15 FDCAN 中断使能寄存器（FDCAN_IE）.....857
31.5.16 FDCAN 中断线选择寄存器（FDCAN_ILS）.....860
31.5.17 FDCAN 中断线使能寄存器（FDCAN_ILE）.....862
31.5.18 FDCAN 全局过滤器配置寄存器（FDCAN_GFC）.....863
31.5.19 FDCAN 标准 ID 过滤器配置寄存器（FDCAN_SIDFC）.....863
31.5.20 FDCAN 扩展 ID 过滤器配置寄存器（FDCAN_XIDFC）.....864
31.5.21 FDCAN 扩展 ID 与掩码寄存器（FDCAN_XIDAM）.....864
31.5.22 FDCAN 高优先级消息状态寄存器（FDCAN_HPMS）.....865
31.5.23 FDCAN 新数据 1 寄存器（FDCAN_NDAT1）.....865
31.5.24 FDCAN 新数据 2 寄存器（FDCAN_NDAT2）.....866
31.5.25 FDCAN 接收 FIFO0 配置寄存器（FDCAN_RXF0C）.....866
31.5.26 FDCAN 接收 FIFO0 状态寄存器（FDCAN_RXF0S）.....867
31.5.27 FDCAN 接收 FIFO0 确认寄存器（FDCAN_RXF0A）.....867
31.5.28 FDCAN 接收缓冲区配置寄存器（FDCAN_RXBC）.....868
31.5.29 FDCAN 接收 FIFO1 配置寄存器（FDCAN_RXF1C）.....868
31.5.30 FDCAN 接收 FIFO1 状态寄存器（FDCAN_RXF1S）.....869
31.5.31 FDCAN 接收 FIFO1 确认寄存器（FDCAN_RXF1A）.....870
31.5.32 FDCAN 接收缓冲区和 FIFO 元素大小配置寄存器（FDCAN_RXESC）.....870
31.5.33 FDCAN 发送缓冲区配置寄存器（FDCAN_TXBC）.....871
31.5.34 FDCAN 发送 FIFO/队列状态寄存器（FDCAN_TXFQS）.....872
31.5.35 FDCAN 发送缓冲区元素大小配置寄存器（FDCAN_TXESC）.....872
31.5.36 FDCAN 发送缓冲区请求挂起寄存器（FDCAN_TXBRP）.....873
31.5.37 FDCAN 发送缓冲区添加请求寄存器（FDCAN_TXBAR）.....874
31.5.38 FDCAN 发送缓冲区取消请求寄存器（FDCAN_TXBCR）.....874
31.5.39 FDCAN 发送缓冲区发送已发生寄存器（FDCAN_TXBTO）.....875
31.5.40 FDCAN 发送缓冲区取消已完成寄存器（FDCAN_TXBCF）.....875
31.5.41 FDCAN 发送缓冲区发送中断使能寄存器（FDCAN_TXBTIE）.....875

Puya Confidential

31.5.42 FDCAN 发送缓冲区取消已完成中断使能寄存器（FDCAN_TXBCIE） 876
31.5.43 FDCAN 发送事件 FIFO 配置寄存器（FDCAN_TXEFC） 876
31.5.44 FDCAN 发送事件 FIFO 状态寄存器（FDCAN_TXEFS） 877
31.5.45 FDCAN 发送事件 FIFO 确认寄存器（FDCAN_TXEFA） 877
32. MCU 调试接口 879
32.1 DBGMCU 简介 879
32.2 DBGMCU 主要特性 879
32.3 DBGMCU 功能描述 879
32.3.1 支持调试低功耗模式 879
32.4 DBGMCU 寄存器描述 879
32.4.1 DBGMCU ID 编码（DBGMCU_IDCODE） 879
32.4.2 DBGMCU 调试配置寄存器 1（DBGMCU_CR1） 880
32.4.3 DBGMCU 调试配置寄存器 2（DBGMCU_CR2） 882
33. 版本历史 883

Puya Confidential

1. 文档约定

1.1 寄存器相关缩写词列表

缩写	描述
读/写(RW)	软件可以读写此位
只读(R)	软件只能读取此位
只写(W)	软件只能写入此位，读此位将返回复位值
读取/写入 0 清零(RC_W0)	软件可以读取此位，也可以通过写 0 清除此位，写 1 对此位无影响
读取/写入 1 清零(RC_W1)	软件可以读取此位，也可以通过写 1 清除此位，写 0 对此位无影响
读取/写入清零(RC_W)	软件可以读取此位，也可以通过写入寄存器来清除该位，写入该位的值并不重要
读取/读取清零(RC_R)	软件可以读取此位。读取此位会将其自动清 0，写入此位对其值无影响
读取/读取置位(RS_R)	软件可以读取此位。读取此位会自动将其置 1，写入此位对其值无影响
读取/置位(RS)	软件可以读取此位，也可以将其置 1，写 0 对此位无影响
切换(T)	软件可以通过写入 1 来切换此位，写入 0 无影响
保留(Res.)	保留位，必须保持在复位值

1.2 外设可用性

有关各型号产品的外设可用性以及数量信息，请参见相关器件数据手册。

Puya Confidential

2. 系统架构框图

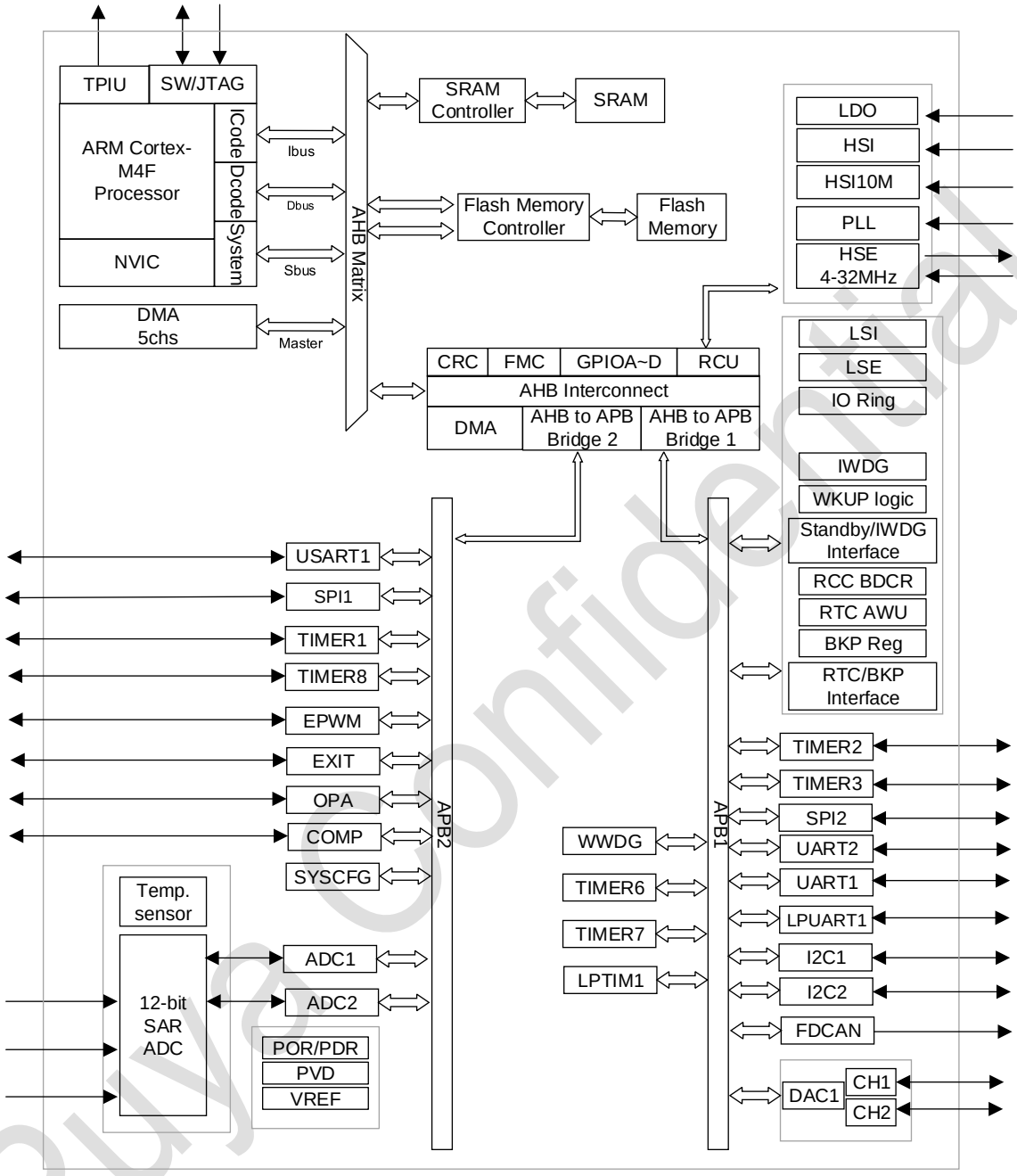


图 2-1 系统框图

Puya Confidential

3. 存储器和总线架构

3.1 系统架构

系统采用 32 位多层 AHB 总线矩阵，可以保证多主机和多从机之间的并行通信：

- 四条主控总线：
  - Cortex®-M4F 内核 I -Code 总线、D-Code 总线和系统总线
  - DMA 总线
- 四条被控总线：
  - 内部 Flash I-Code 总线
  - 内部 Flash D-Code 总线
  - 内部 SRAM 总线
  - AHB 外设总线（包括 AHB 到 APB 的总线桥和 APB 外设）

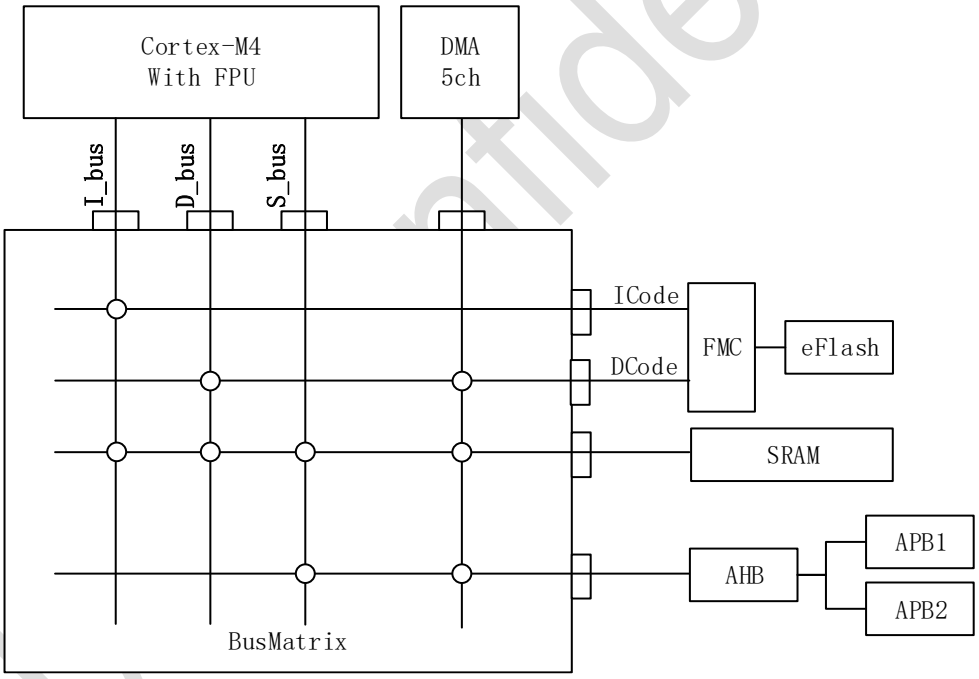


图 3-1 总线架构

3.1.1 I_bus

此总线用于将 Cortex®-M4F 内核的指令总线连接到总线矩阵。内核通过此总线获取指令。此总线访问的对象是包含代码的存储器（内部 Flash/SRAM）。

3.1.2 D_bus

此总线用于将 Cortex®-M4F 数据总线连接到总线矩阵。内核通过此总线进行立即数加载和调试访问。此总线访问的对象是包含代码或数据的存储器（内部 Flash/SRAM）。

Puya Confidential

3.1.3 S_bus

此总线用于将 Cortex®-M4F 内核的系统总线连接到总线矩阵。此总线用于访问位于外设或 SRAM 中的数据。此总线访问的对象是 48 KB 的内部 SRAM、包括 AHB 外设在内的 APB1 外设、APB2 外设。

3.1.4 DMA BUS

此总线用于将 DMA 存储器总线主接口连接到总线矩阵。DMA 通过此总线访问的对象是 16KB 的内部 SRAM、包括 AHB 外设在内的 APB1 外设、APB2 外设。

3.1.5 总线矩阵

总线矩阵用于主控总线之间的访问仲裁管理。仲裁采用循环调度算法。系统包含 4 个主设备 Master，分别为 ARM® Cortex®-M4F 的 I_BUS、D_BUS、S_BUS、DMA；及 4 个从设备 Slave，分别为内部的 FLASH 存储器、内部的 SRAM，AHB 外设（AHB2APB 桥 1 对应的外设和 AHB2APB 桥 2 对应的外设）。

3.1.6 AHB/APB 总线桥

借助两个 AHB/APB 总线桥 APB1 和 APB2，可在 AHB 总线与两个 APB 总线之间实现完全同步的连接，从而灵活选择外设频率。

每次芯片复位后，所有外设时钟都被关闭（SRAM 和 Flash 接口除外）。使用外设前，必须在 RCC_AHBENRx 或 RCC_APBENRx 寄存器中使能其时钟。

有关 AHB 和 APB 外设地址映射信息，参考 3.2 章节。

3.2 存储器组织架构

3.2.1 简介

Arm® Cortex®-M4F 处理器采用哈佛结构，可以使用相互独立的总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和 I/O 端口都在同一个线性的 4 GB 的地址空间之内。

各字节按小端格式在存储器中编码。字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。

有关外设寄存器映射的详细信息，请参见相关章节。

可寻址的存储空间分为 8 个主要块，每个块为 512 MB。

Puya Confidential

3.2.2 存储器映射

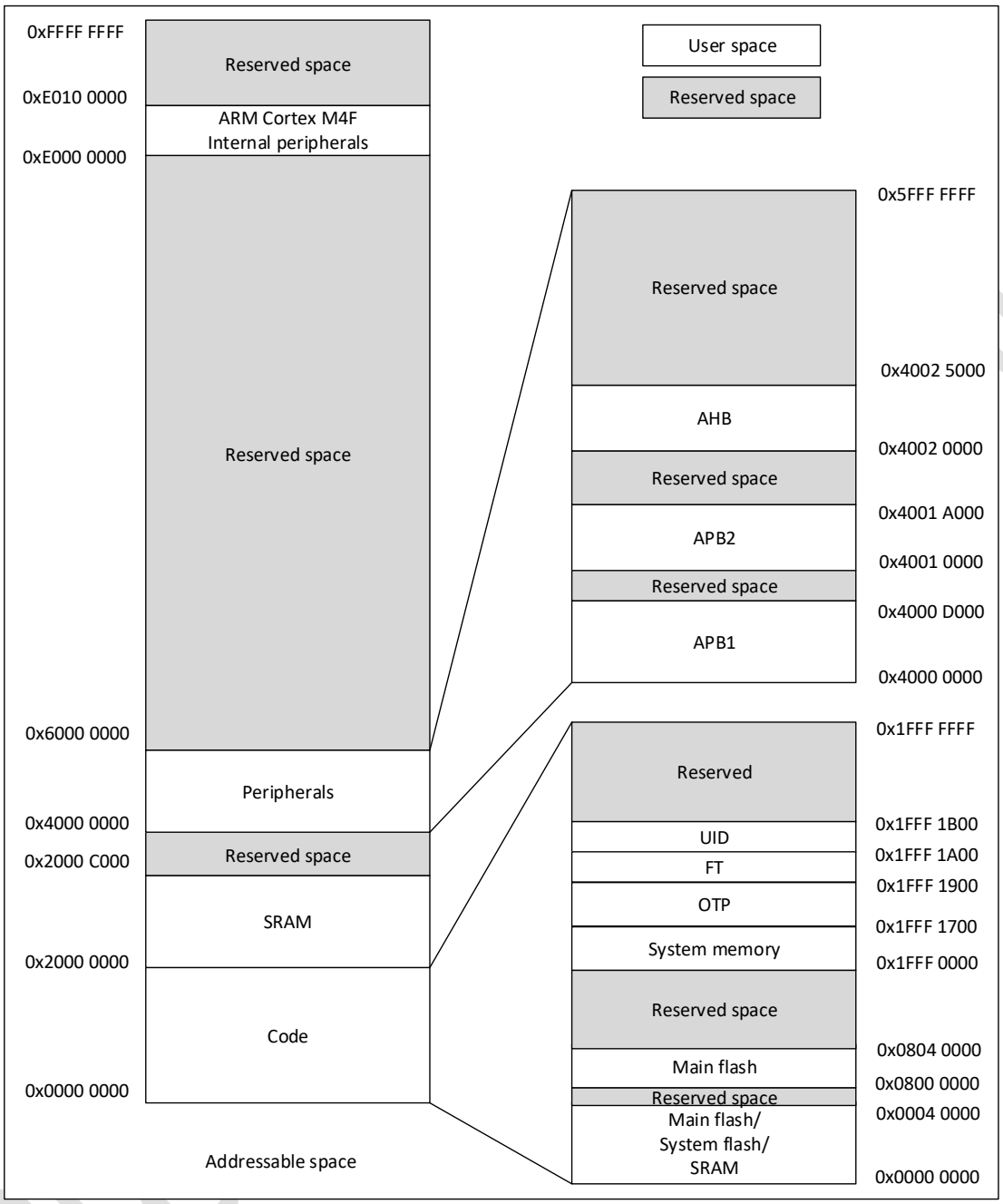


图 3-2 存储器映射

其他所有没有分配给片上存储器和外设的存储器空间都是保留的地址空间。详细的存储和寄存器空间映射参考下面的表格。

下表中给出了产品中可使用的外设及对应的地址边界。

表 3-1 存储器地址

类型	存储器地址	Size	Memory Area	描述
SRAM	0x2000 C000-0x3FFF FFFF	511 MB	保留	1.CPU 读写该空间时产生 Response error，进而进入 Hard-Fault 异常； 2.DMA 访问时产生 TEIF 状态位；
	0x2000 0000-0x2000 BFFF	48 KB	SRAM	SRAM 为 48 KB，地址空间为 0x2000 0000-0x2000 BFFF
Code	0x1FFF 1A00 - 0x1FFF 1AFF	256 bytes	UID	Unique ID
	0x1FFF 1900 - 0x1FFF 19FF	256 bytes	FT	
	0x1FFF 1700 - 0x1FFF 18FF	512 bytes	OTP	
	0x1FFF 0000 - 0x1FFF 16FF	5.75 KB	System memory	存放 Boot loader
	0x0804 0000-0x1FFE FFFF	127 MB	保留	-
	0x0800 0000-0x0803 FFFF	256 KB	Main flash memory	-
	0x0004 0000-0x07FF FFFF	127 MB	保留	1.CPU 读写该空间时产生 Response error，进而进入 HardFault 异常； 2.DMA 访问时产生 TEIF 状态位；
	0x0000 0000-0x0003 FFFF	256 KB	根据 Boot 配置选择是： 1）Main flash memory 2）System memory 3）SRAM	-

1. 上述空间标注为**保留**的空间，无法进行写操作，读为 0，且产生 response error。

表 3-2 外设寄存器地址

存储器起止地址	外设	总线	最大频率
0x4002 5000 - 0x4002 FFFF	保留	AHB	176/144 MHz
0x4002 4C00 - 0x4002 4FFF	GPIOD		
0x4002 4800 - 0x4002 4BFF	GPIOC		
0x4002 4400 - 0x4002 47FF	GPIOB		
0x4002 4000 - 0x4002 43FF	GPIOA		
0x4002 3400 - 0x4002 3FFF	保留		
0x4002 3000 - 0x4002 33FF	CRC		
0x4002 2400 - 0x4002 2FFF	保留		
0x4002 2000 - 0x4002 23FF	FMC		
0x4002 1400 - 0x4002 1FFF	保留		
0x4002 1000 - 0x4002 13FF	RCC		
0x4002 0400 - 0x4002 0FFF	保留		
0x4002 0000 - 0x4002 03FF	DMA1		
0x4001 A000 - 0x4001 FFFF	保留	APB2	176/144 MHz
0x4001 8000 - 0x4001 9FFF	EPWM		
0x4001 7400 - 0x4001 7FFF	保留		
0x4001 7000 - 0x4001 73FF	OPAMP		
0x4001 6C00 - 0x4001 6FFF	COMP		
0x4001 3C00 - 0x4001 6BFF	保留		

Puya Confidential

0x4001 3800 - 0x4001 3BFF	USART1		
0x4001 3400 - 0x4001 37FF	TIMER8		
0x4001 3000 - 0x4001 33FF	SPI1		
0x4001 2C00 - 0x4001 2FFF	TIMER1		
0x4001 2800 - 0x4001 2BFF	ADC2		
0x4001 2400 - 0x4001 27FF	ADC1		
0x4001 2000 - 0x4001 23FF	VREFBUF		
0x4001 0800 - 0x4001 23FF	保留		
0x4001 0400 - 0x4001 07FF	EXTI		
0x4001 0000 - 0x4001 03FF	SYSCFG		
0x4000 D000 - 0x4000 FFFF	保留		
0x4000 CC00 - 0x4000 CFFF	CAN		
0x4000 B000 - 0x4000 CBFF	保留		
0x4000 AC00 - 0x4000 AFFF	CANMEM		
0x4000 8400 - 0x4000 ABFF	保留		
0x4000 8000 - 0x4000 83FF	LPUART1		
0x4000 7C00 - 0x4000 7FFF	LPTIM1		
0x4000 7800 - 0x4000 7BFF	保留		
0x4000 7400 - 0x4000 77FF	DAC		
0x4000 7000 - 0x4000 73FF	PWR		
0x4000 5800 - 0x4000 6FFF	保留		
0x4000 5800 - 0x4000 5FFF	I2C2		
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 5000 - 0x4000 53FF	UART2	APB1	176/144 MHz
0x4000 4C00 - 0x4000 4FFF	UART1		
0x4000 3C00 - 0x4000 4BFF	保留		
0x4000 3800 - 0x4000 3BFF	SPI2		
0x4000 3400 - 0x4000 37FF	保留		
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		
0x4000 2800 - 0x4000 2BFF	RTC		
0x4000 1800 - 0x4000 27FF	保留		
0x4000 1400 - 0x4000 17FF	TIMER7		
0x4000 1000 - 0x4000 13FF	TIMER6		
0x4000 0800 - 0x4000 0FFF	保留		
0x4000 0400 - 0x4000 07FF	TIMER3		
0x4000 0000 - 0x4000 03FF	TIMER2		

1. 上表 AHB 标注为保留的地址空间，无法写操作，读回为 0，且产生 HardFault。

Puya Confidential

3.3 嵌入式 SRAM

PY32F420 系列器件内置最大 48 KB 的静态 SRAM，根据不同产品定义，SRAM 容量会有差异，具体请参考数据手册。它可以以字节、半字（16 位）或全字（32 位）访问。SRAM 的起始地址是 0x2000 0000。

3.4 嵌入式 Flash

Flash 存储器有两个不同的物理区域组成：

■ Main flash 区域，256 KB，有两个 Bank 组成，它包含应用程序和用户数据。可根据不同产品定义，Flash 容量会有差异，具体请参考数据手册。用于存储用户程序和用户数据。软件对设定范围外空间的访问会产生 HardFault。

■ Information 区域，6.75 KB，它包括以下部分：

- FT：256 bytes，由于存放芯片 FT 信息：
- OTP：512 bytes，用于存放芯片的 OTP
- UID：256 bytes，用于存放芯片的 UID
- System memory（系统存储器）：5.75 KB，用于存放 Boot loader

Flash 接口实现基于 AHB 协议的指令读取和数据访问，它也通过寄存器实现了 Flash 的基本编程/擦写等操作。

3.5 位段

Cortex®-M4F 存储器映射包括两个位段区域。这些区域将存储器别名区域中的每个字映射到存储器位段区域中的相应位。在别名区域写入字时，相当于对位段区域的目标位执行读-修改-写操作。

在 PY32F420 器件中，外设寄存器和 SRAM 均映射到一个位段区域，这样可实现单个位段的读写操作。这些操作仅适用于 Cortex®-M4F 访问，对于其它总线主接口（如 DMA）无效。

可通过一个映射公式说明别名区域中的每个字与位段区域中各个位之间的对应关系。映射公式为：

bit_word_addr = bit_band_base + (byte_offset x 32) + (bit_number × 4)

其中：

— bit_word_addr 代表别名区域中将映射到目标位的字的地址

— bit_band_base 代表别名区域的起始地址

— byte_offset 代表目标位所在位段区域中的字节编号

— bit_number 代表目标位的位位置（0-7）

示例

下例说明如何将 SRAM 地址 0x20000300 处字节的位 2 映射到别名区域：

0x22006008 = 0x22000000 + (0x300*32) + (2*4)

对地址 0x22006008 执行写操作相当于在 SRAM 地址 0x20000300 处字节的位 2 执行读-修改-写操作。

Puya Confidential

对地址 0x22006008 执行读操作将返回 SRAM 地址 0x20000300 处字节的位 2 的值（0x01 表示位置位，0x00 表示位复位）。

有关位段的详细信息，请参见 Cortex®-M4F 编程手册（DUI508B_cortex_M4F_ugrm.pdf）。

3.6 启动配置

通过配置位 BOOT0 Pin 以及选项字节的配置，可选择三种不同的启动模式，如下表所示：

表 3-3 Boot 配置

Boot mode configuration					Mode
BOOT_LOCK	nBOOT1 FLASH_OPTR2[8]	nBOOT0 FLASH_OPTR2[14]	BOOT0 Pin	nSWBOOT0 FLASH_OPTR2[13]	
1	X	X	X	X	选择 Main flash 作为启动区
0	X	X	0	1	选择 Main flash 作为启动区
0	X	1	X	0	选择 Main flash 作为启动区
0	0	X	1	1	选择 SRAM 作为启动区
0	0	0	X	0	选择 SRAM 作为启动区
0	1	X	1	1	选择 System flash 作为启动区
0	1	0	X	0	选择 System flash 作为启动区

在系统复位后，BOOT0 引脚或用户选项配置位（取决于 FLASH_OPTR 寄存器中的 nSWBOOT0 位值）的值和 nBOOT1 位将被锁存。从待机模式退出时，BOOT0 引脚的值将被被重新锁存；因此，在待机模式下 BOOT0 引脚应保持为需要的启动配置。

根据选定的启动模式，主闪存存储器、系统存储器或 SRAM 可以按照以下方式访问：

- 从主闪存存储器启动：主闪存存储器被映射到启动空间（0x0000 0000），但仍然能够在它原有的地址（0x0800 0000）访问它，即闪存存储器的内容可以在两个地址区域访问，0x0000 0000 或 0x0800 0000。
- 从系统存储器启动：系统存储器被映射到启动空间（0x0000 0000），但仍然能够在它原有的地址（0x1FFF 0000）访问它。
- 从内置 SRAM 启动：SRAM 被映射到启动空间（0x0000 0000），但仍然能够在它原有的地址（0x2000 0000）访问它。

在启动后，CPU 从地址 0x0000 0000 获取堆栈顶的地址，并从启动存储器的 0x0000_0004 指示的地址开始执行代码。因为固定的存储器映像，代码区始终从地址 0x0000_0000 开始（通过 ICode 和 DCode 总线访问），而数据区（SRAM）始终从地址 0x2000_0000 开始（通过系统总线访问）。Cortex M4F 的 CPU 始终从 ICode 总线获取复位向量，即启动仅适合于从代码区开始（典型地从 Flash 启动）。

PY32F420 系列微控制器实现了一个特殊的机制，可以从片上 SRAM 启动。当从片上 SRAM 启动，在应用程序的初始化代码中，必须使用 NVIC 的异常表和偏移寄存器，重新映射向量表到 SRAM 中。

3.6.1 系统存储启动程序

系统存储器中包含内嵌的引导加载程序，可用于对闪存存储器编程。引导加载程序是通过 USART/UART 接口对闪存存储器进行重新编程。

Puya Confidential

3.6.2 物理重映射

选择启动引脚后，应用程序软件可以将某些存储器设定为从代码空间进行访问（这样，可通过 ICode 总线而非系统总线执行代码）。这样的修改通过在 SYSCFG 控制器中编程 SYSCFG.MEM_MODE 来实现。因此可重映射以下存储器：

- Main flash
- 系统存储器
- 嵌入式 SRAM

表 3-4 重映射地址

地址	Main flash 启动/重映射	嵌入式 SRAM 中的启动/重映射	系统存储器中的启动/重映射
0x2000 0000 - 0x2000 BFFF	SRAM	SRAM	SRAM
0x1FFF 0000 - 0x1FFF 1D00	系统存储器	系统存储器	系统存储器
0x0804 0000 - 0x1FFE FFFF	保留	保留	保留
0x0800 0000 - 0x0803 FFFF	Flash	Flash	Flash
0x0004 0000 - 0x07FF FFFF	保留	保留	保留
0x0000 0000 - 0x0003 FFFF	Flash (使用别名)	SRAM (使用别名)	系统存储器(5.75 KB) (使用别名)

Puya Confidential

4. 嵌入式 Flash 接口

4.1 Flash 简介

Flash 接口可管理 CPU 通过 AHB I-Code 和 D-Code 总线对 Flash 进行访问。该接口可针对 Flash 执行擦除和编程操作，并实施读写保护机制。

Flash 接口通过指令预取和缓存机制加速代码执行。

4.2 Flash 主要特征

- 存储器组织结构：
  - Main flash memory：最大 256 KB
  - Information memory：6.75 KB
  - 页（page）大小：256 bytes
  - 山区（sector）大小：8 KB
  - 读宽度为 64 位
  - 写宽度位 32 位
- Flash 读操作
- Flash 编程操作（页编程，编程的最小单位是页）
- Flash 擦除操作（页擦/扇区擦/全擦，擦除的最小单位是页）
- 读/写保护
- I-Code 上的预取操作
- I-Code 上的分支缓存
- D-Code 上的数据缓存
- 选项字节加载功能
- Flash 编程只能写 0，不能写 1，擦除以后的值是 1

4.3 Flash 功能描述

4.3.1 闪存结构

Flash 存储器由 64 位宽的 cell 组成，具有支持边读边写功能（RWW）的双 Bank 架构，可以用作程序和数据的存储。页大小为 256 bytes，扇区大小为 8 KB。

从功能上，Flash 存储器分为 Main flash 和 information block，前者容量最大是 256 KB，后者容量为 6.75 KB，如下：

Puya Confidential

表 4-1 闪存结构及边界地址

Area	Space Name			Address	Size (bytes)
Main flash	Bank0	Sector 0	Page 0~31	0x0800 0000 - 0x0800 1FFF	8 KB
		Sector 1	Page 32~63	0x0800 2000 - 0x0800 3FFF	8 KB
		Sector 2	Page 64~95	0x0800 4000 - 0x0800 5FFF	8 KB
		Sector 3	Page 96~127	0x0800 6000 - 0x0800 7FFF	8 KB
		…	…	…	…
		Sector 15	Page 480~511	0x0801 E000 - 0x0801 FFFF	8 KB
	Bank1	Sector 16	Page 512~543	0x0802 0000 - 0x0802 1FFF	8 KB
		Sector 17	Page 544~575	0x0802 2000 - 0x0802 3FFF	8 KB
		Sector 18	Page 576~607	0x0802 4000 - 0x0802 5FFF	8 KB
		…	…	…	…
		Sector 31	Page 992~1023	0x0803 E000 - 0x0803 FFFF	8 KB
Information block	Bank0	System boot memory		0x1FFF 0000 - 0x1FFF 16FF	5.75 KB
		OTP0		0x1FFF 1700 - 0x1FFF 17FF	256 bytes
		OTP1		0x1FFF 1800 - 0x1FFF 18FF	256 bytes
		FT		0x1FFF 1900 - 0x1FFF 19FF	256 bytes
		UID		0x1FFF 1A00 - 0x1FFF 1AFF	256 bytes

4.3.2 读访问延迟

为了正确的从 Flash 存储器读取指令/数据，需要根据 CPU 时钟（HCLK）的频率，设置寄存器 FLASH_ACR 的等待周期数（Latency）。

在芯片复位后（包括上电复位，系统复位，退出 standby 复位），HCLK 时钟频率为 8 MHz，Flash 读周期数为 0。

当改变 CPU 时钟频率或范围时，必须按照以下软件步骤来调整访问 FLASH 存储器所需要的等待周期数。

提高 CPU 频率时的操作步骤

1. 将新的等待周期数编程到 FLASH_ACR 寄存器中的 LATENCY 位
2. 通过读取 FLASH_ACR 寄存器，检查新的等待周期是否设置成功
3. 通过改写 RCC_CFGR1 寄存器中的 SW 位来修改 CPU 时钟源
4. 如有需要，可通过改写 RCC_CFGR1 中的 HPRE 位来修改 CPU 时钟预分频器
5. 通过读取 RCC_CFGR1 寄存器中相应的时钟源状态（SWS 位）和/或 AHB 预分频值（HPRE 位），检查新的 CPU 时钟源和/或新的 CPU 时钟预分频值是否设置成功

降低 CPU 频率时的操作步骤

1. 通过改写 RCC_CFGR1 寄存器中的 SW 位来修改 CPU 时钟源
2. 如有需要，可通过改写 RCC_CFGR1 中的 HPRE 位来修改 CPU 时钟预分频器
3. 通过读取 RCC_CFGR1 寄存器中相应的时钟源状态（SWS 位）和/或 AHB 预分频值（HPRE 位），检查新的 CPU 时钟源和/或新的 CPU 时钟预分频值是否设置成功
4. 将新的等待周期数编程到 FLASH_ACR 中的 LATENCY 位
5. 通过读取 FLASH_ACR 寄存器，检查新的等待周期是否设置成功

Puya Confidential

4.3.3 自适应实时存储器加速器

专有的自适应实时（ART）存储器加速器面向 ARM® Cortex®-M4F 处理器 进行了优化。该加速器很好的体现了 ARM®Cortex®-M4F 的固有性能优势，克服了通常条件下，高速的处理器在运行中需要经常等待 Flash 读取的情况。

为了发挥处理器的全部性能，该加速器将实施指令预取队列和分支缓存，从而提高了 64 位 Flash 的程序执行速度。根据 CoreMark 基准测试，凭借 ART 加速器所获得的性能相当于 Flash 在 CPU 频率高达 176 MHz 时以 0 个等待周期执行程序。

- **指令预取**

每个 Flash 读操作可读取 64 位，可以是 2 行 32 位指令，也可以是 4 行 16 位指令，具体取决于烧写在 Flash 中的程序。因此对于顺序执行的代码，至少需要 4 个 CPU 周期来执行前一次读取的 64 位指令行。在 CPU 请求当前指令行时，可使用 I-Code 总线的预取操作读取 Flash 中的下一个连续存放的 64 位指令行。

可将 Flash 访问控制寄存器（FLASH_ACR）中的 PRFTEN 位置 1，来使能预取功能。当访问 Flash 至少需要一个等待状态时，此功能非常有用。

- **指令缓存存储器**

为了减少因指令跳转而产生的时间损耗，可将 64 行 64 位的指令保存到指令缓存存储器中。每当出现指令缺失（即请求的指令未存在于当前使用的指令行、预取指令行或指令缓存存储器中）时，系统会将新读取的行复制到指令缓存存储器中。如果 CPU 请求的指令已存在于指令缓存区中，则无需任何延时即可立即获取。指令缓存存储器存满后，可采用 LRU（最近最少使用）策略确定指令缓存存储器中待替换的指令行。此特性非常适用于包含循环的代码。

可将 Flash 访问控制寄存器（FLASH_ACR）中的指令缓存使能（ICEN）位置 1，来使能此特性。

- **数据缓存存储器**

在 CPU 流水线执行阶段，将通过 D-Code 总线访问 Flash 中的数据缓冲区。CPU 流水线只有获得请求的数据后才会继续执行。为了减少因此而产生的时间损耗，通过 AHB 数据总线 D-Code 进行的访问优先于通过 AHB 指令总线 I-Code 进行的访问。此特性的工作原理与指令缓存存储器类似，但保留的数据大小限制在 16 行 64 位以内。

可将 Flash 访问控制寄存器（FLASH_ACR）中的数据缓存使能（DCEN）位置 1，来使能数据缓存存储器。

4.3.4 Flash 擦除和编程操作

通过 ICP（In-circuit programming）或者 IAP（In-application programming）可以对 Flash 进行编程操作。

ICP：用来更新整个 Flash 存储器的内容，可以使用 JTAG/SWD 协议或者 boot loader，把用户应用装入 MCU 中。

IAP：可以使用芯片支持的通讯接口，下载要编程的数据到 Flash 中。IAP 允许用户在应用运行时，再次编程 Flash 存储器。

如果在进行 Flash 编程或擦写操作时，发生了复位，则 Flash 存储器的内容是不被保护的。

Puya Confidential

在对一个 BANK 进行编程或擦写操作期间，可以对另外一个 BANK 进行读访问。对正在编程或擦写操作中 BANK 再进行编程或擦写操作时，会产生 HardFault。对正在编程或擦写操作中 BANK 再进行读操作时，会拉住总线。

如果使能 FLASH_OPTR2. VERIFY_READ_EN，总线会被 hold 住，直到这次的擦或写完成。

对于编程或擦写操作，必须打开 HSI。

4.3.5 闪存解锁

在复位后，Flash 控制寄存器（FLASH_CR）不允许执行写操作，以防止因电气干扰等原因出现对 Flash 的意外操作。每次对 Flash 的编程或擦写操作，都必须通过写 FLASH_KEYR 来解锁，以启用对 FLASH_CR 寄存器的访问。

具体步骤如下：

步骤 1：向 FLASH_KEYR 寄存器写入 KEY1=0x4567 0123

步骤 2：向 FLASH_KEYR 寄存器写入 KEY2=0xCDEF 89AB

任何错误的时序都会锁定住 FLASH_CR 寄存器，直到下一次复位。在错误的 KEY 时序时，会产生总线 HardFault 中断。这样的错误包括第一个写周期的 KEY1 不匹配，或者 KEY1 匹配但第二个写周期的 KEY2 不匹配。

FLASH_CR 寄存器可以通过软件写 FLASH_CR 寄存器的 LOCK 位被再次锁定。

注意：当 FLASH_SR 寄存器的 BSY 位被置位时，FLASH_CR 寄存器不能进行写操作。

4.3.6 闪存擦除操作

Flash 存储器可以按照页进行擦除操作，或者进行扇区和 bank 的片擦（bank0 片擦对 Flash 信息区不起作用）。

闪存页擦除

页擦除用来对 256 bytes 的 Main flash 进行擦除操作，但对信息区不起作用。

当某个页被 WRP 保护，它是不会被擦除的，此时 WRPERR 位被置位。

页擦除操作的具体步骤如下：

1) 检查 FLASH_SR 寄存器 BSYx 位，确认没有正在进行的 Flash 操作
2) 向 FLASH_KEYR 寄存器依次写 KEY1 和 KEY2，解除 FLASH_CR 寄存器的保护
3) 设置 FLASH_CR 寄存器的 PER 位和 EOPIE（如果需要产生 EOP 中断）位
4) 向该页写任意数据（必须 32 位数据）
5) 等待 BSY 位被清零
6) 检查 EOP 标志位被置位
7) 清除 EOP 标志

闪存扇区擦除

扇区擦除用来对 8 KB 的 Main flash 进行擦除操作，但对信息区不起作用。

当某个扇区被 WRP 保护，它是不会被擦除的，此时 WRPERR 位被置位。

扇区擦除操作的具体步骤如下：

1) 检查 BSYx 位，确认是否没有正在进行的 Flash 操作
2) 向 FLASH_KEYR 寄存器依次写 KEY1 和 KEY2，解除 FLASH_CR 寄存器保护
3) 设置 FLASH_CR 寄存器的 SER 位和 EOPIE（如果需要产生 EOP 中断）位

Puya Confidential

4) 向该扇区写任意数据（必须 32 位数据）

5) 等待 BSY 位被清零

6) 检查 EOP 标志位被置位

7) 清除 EOP 标志

闪存 Bank 擦除

Bank erase 用来对 128 KB 的 Main flash 中进行擦除操作，但对信息区不起作用。

当 WRP 被使能，Bank erase 功能无效，不会产生 Bank erase 操作，并且 WEPERR 位被置位。

Bank erase 操作的具体步骤如下：

1) 检查 BSYx 位，确认是否没有正在进行的 Flash 操作

2) 向 FLASH_KEYR 寄存器依次写 KEY1、KEY2，解除 FLASH_CR 寄存器保护

3) 设置 FLASH_CR 寄存器的 MER1 位或 MER0 位和 EOPIE（如果需要产生 EOP 中断）位

4) 向对应 Bank Flash 的任意 Flash 空间写任意数据（必须 32 位数据）

5) 等待 BSY 位被清零

6) 检查 EOP 标志位被置位

7) 清除 EOP 标志

注：如果使能 FLASH_OPTR2. VERIFY_READ_EN，1）FLASH 的系统时钟需要设置在 25 M 以下，2）进行 sector erase 或者 bank erase，触发地址必须为所擦 sector 或 bank 的头 page 地址。

4.3.7 闪存写操作

Flash 存储器每次以 32 位（word）为单位（进行 half word 或者 byte 操作会产生 HardFault）进行整个页的编程操作。当 FLASH_CR 寄存器的 PG 位被置位，CPU 向 FLASH 存储器地址空间写 32 位数据时，编程操作开始启动。

如果要编程的 Flash 地址空间，是被 FLASH_BANK0_WRPR 或 FLASH_BANK1_WRPR 寄存器设置为保护的区域，则编程操作会被忽略掉，同时 FLASH_SR 寄存器 WRPERR 位会被置位。Program 操作的结束，FLASH_CR 寄存器的 EOP 位会被置位。

具体 Flash program 的操作步骤如下所示：

1) 检查 FLASH_SR 寄存器的 BSYx 位，判断是否当前没有正在继续的 Flash 操作

2) 如果没有正在进行的 Flash erase 或者 program 操作，则软件读出该 Page 的 64 个 word（如果需要保留该 page 已有的数据，则进行该步骤，否则跳过该步骤）

3) 向 FLASH_KEYR 寄存器依次写 KEY1 和 KEY2，解除 FLASH_CR 寄存器的保护

4) 设置 FLASH_CR 寄存器的 PG 位和 EOPIE（如果需要产生 EOP 中断）位

5) 向目标地址进行第 1 到第 63 个 word 的编程操作（只接受 32bit 的编程）

6) 设置 FLASH_CR 寄存器的 PGSTRT

7) 写第 64 个 word

8) 等待 FLASH_SR 寄存器的 BSY 位被清零

9) 检查 FLASH_SR 寄存器的 EOP 标志位（当 program 操作已经成功，该位被置位），然后软件清除该位

10) 如果不再有编程操作，则软件清除 PG 位

当上述步骤 7）成功执行，则编程操作自动启动，同时 BSY 位被硬件置位。

Puya Confidential

4.4 产品唯一身份标识码（UID）

唯一身份标识码典型应用场景：

- 用作序列号
- 对内部闪存编程时，将其用作密钥或加密原语以提高代码的安全性
- 激活安全自举过程等

产品唯一身份标识提供了一个对于任何设备都唯一的参考号码。

用户永远不能改变这些位。唯一身份标识符也可以以单字节/半字/字等不同方式进行读取，然后使用自定义的算法连接起来。

本产品 UID 内容如下表所示：

基址：0x1FFF 1A00

表 4-2 UID 标识码

偏移地址	描述	UID Bits							
		7	6	5	4	3	2	1	0
0	Lot Numer	Lot Number ASCII 码							
1	Lot Numer	Lot Number ASCII 码							
2	Lot Numer	Lot Number ASCII 码							
3	Lot Numer	Lot Number ASCII 码							
4	Wafer Number	Wafer Number							
5	Lot Numer	Lot Number ASCII 码							
6	Lot Numer	Lot Number ASCII 码							
7	Lot Numer	Lot Number ASCII 码							
8	内部编码	内部编码							
9	Y 坐标低位	Y 坐标低位							
10	X 坐标低位	X 坐标低位							
11	X,Y 坐标高地址	Y 坐标高位				X 坐标高位			
12	CP pass ID	CP pass ID							
13	CRC8	0x1FFF 1A00 - 0x1FFF 1AFF 字节地址的数据的 CRC8 校验值							
14	内部编码	内部编码							
15	内部编码	内部编码							

4.5 Flash 选项字节

4.5.1 Flash 选项字节编程

复位后，FLASH_CR 寄存器中与选项字节相关的位是被写保护的。当对选项字节进行相关操作前，FLASH_CR 寄存器中的 OPTLOCK 位必须被清零。

以下步骤用来解锁该寄存器：

1. 通过解锁流程，解除 FLASH_CR 寄存器的写保护
2. 向 FLASH_OPTKEYR 寄存器，写 OPTKEY1=0x0819 2A3B
3. 向 FLASH_OPTKEYR 寄存器，写 OPTKEY2=0x4C5D 6E7F

Puya Confidential

任何错误的时序都会锁住 FLASH_CR 寄存器，直到下一次复位。在错误的 KEY 时序时，会产生总线 HardFault 中断。

User option（information Flash 的选项字节）可以通过软件写 FLASH_CR 寄存器的 OPTLOCK 位，被保护住，以防止不要的 erase/program 操作。

如果软件设置 Lock 位，则 OPTLOCK 位也被自动置位。

4.5.2 修改选项字节

选项字节的编程操作，和 Main flash 的操作不一样。为修改选项字节，需要进行如下步骤：

1. 用之前描述的步骤，清零 OPTLOCK 位
2. 检查 BSY 位，确认没有正在进行的 Flash 操作
3. 向选项字节寄存器 FLASH_OPTR1/FLASH_OPTR2/ FLASH_BANK0_WRPR/ FLASH_BANK1_WRPR /FLASH_PCROP0SR/ FLASH_PCROP0ER/ FLASH_PCROP1SR/ FLASH_PCROR1ER 写期望的值（1~8 个 word）
4. 设置 OPTSTRT 位和 EOPIE 位
5. 向 0x1FFF 1B00 地址写任意数据（必须 32bit 数据）
6. 等待 BSY 位被清零
7. 检查 EOP 标志位被置位
8. 清除 EOP 标志

任何对选项字节的改动，硬件都会先把选项字节对应的整个 page erase 掉，然后用 FLASH_OPTR1、FLASH_OPTR2 、 FLASH_BANK0_WRPR 、 FLASH_BANK1_WRPR 、 FLASH_PCROR0SR 、FLASH_PCROR0ER 、 FLASH_PCROR1SR 、 FLASH_PCROR1ER 、 FLASH_SECER0 、FLASH_SECER1、FLASH_OPTBFBR 寄存器的值，写入选项字节中。并且，硬件自动计算相应的补码，并把计算值写入选项字节的相应区域。

4.5.3 修改选项字节 BFB（Boot from Bank）

选项字节的编程操作，和 Main flash 的操作不一样。为修改选项字节，需要进行如下步骤：

1. 用之前描述的步骤，清零 OPTLOCK 位
2. 检查 BSY 位，确认没有正在进行的 FLASH 操作
3. 向选项字节寄存器 FLASH_OPTBFBR 写期望的值（1 个 word）
4. 设置 OPTSTRT 位和 EOPIE 位
5. 向 0x1FFF 1C00 地址写任意数据（必须 32bit 数据）
6. 等待 BSY 位被清零
7. 检查 EOP 标志位被置位
8. 清除 EOP 标志

任何对选项字节的改动，硬件都会先把选项字节对应的整个 page erase 掉，然后用 FLASH_OPTBFBR 寄存器的值，写入选项字节中。并且，硬件自动计算相应的补码，并把计算值写入选项字节的相应区域。

4.5.4 加载选项字节

在 BSY 位被清零后，所有新的选项字节被写入了 FLASH information 存储器中，但是未应用于芯片系统。对选项字节寄存器进行读操作，仍然返回上一次被装载的选项字节里的值。仅当他们（新值）被装载后，才对芯片系统起作用。

Puya Confidential

选项字节的装载，在以下两种情况下进行：

- 当 FLASH_CR 寄存器中的 OBL_LAUNCH 位被置位
- 在上电复位后（POR）

“装载选项字节”进行的操作是：对 information memory 区域的选项字节进行读操作，再把读出的数据存储在内部 option 寄存器中（FLASH_OPTR1、FLASH_OPTR2、FLASH_BANK0_WRPR、FLASH_BANK1_WRPR、FLASH_PCROR0SR、FLASH_PCROR0ER、FLASH_PCROR1SR、FLASH_PCROR1ER，FLASH_SECR0，FLASH_SECR1，FLASH_OPTBFBR）。

每个 option 位在它相同的双字地址(下一个 half word)有相应的补码。在选项字节装载期间，会对 option bit 和其反码进行验证，确保装载被正确进行。

如果正反码匹配，则选项字节被复制到 option 寄存器中。

如果正反码不匹配，则 FLASH_SR 寄存器的 OPTVERR 状态位被置位。不匹配的值被写入 option 寄存器：

- 对于 OPTR1 和 OPTR2
  - RDP 位写成 0xFF（即 level 1）
  - 其余不匹配的值都写成 1
- 对于 WRP option，不匹配的值是默认值“无保护”
- 对于 PCROP option，不匹配的值是默认值“全闪存保护”
- 对于 SECPRT option，不匹配的值是缺省值“全闪存保护”

在系统复位后，选项字节的内容被复制到下面的 option 寄存器（软件可读可写）：

- FLASH_OPTR1
- FLASH_OPTR2
- FLASH_BANK0_WRPR
- FLASH_SECR0
- FLASH_PCROP0SR
- FLASH_PCROP0ER
- FLASH_BANK1_WRPR
- FLASH_SECR1
- FLASH_PCROP1SR
- FLASH_PCROP1ER
- FLASH_OPTBFBR

这些寄存器也被用来修改选项字节。

表 4-3 Factory config. bytes

NO.	Address	Contents
1	0x1FFF 1A20	1.2 V V_{REFINT} value
2	0x1FFF 1A28	1.024 V V_{REFBUF} value
3	0x1FFF 1A30	2.048 V V_{REFBUF} value
4	0x1FFF 1A34	2.5 V V_{REFBUF} value
5	0x1FFF 1A54	4.96 V V_{REFBUF} value
6	0x1FFF 1A80	HSI 8 MHz Trimming

Puya Confidential

7	0x1FFF 1A84	HSI 16 MHz Trimming
8	0x1FFF 1A8C	HSI 24 MHz Trimming
9	0x1FFF 1A90	HSI 48 MHz Trimming
10	0x1FFF 1AB8	常温温度传感器数据
11	0x1FFF 1ABC	高温温度传感器数据

4.6 Flash 用户数据字节（OTP）

芯片内的 Flash 的 information 区域的部分区间作为 Flash USER OTP memory bytes。
共计 2 Page。

表 4-4 USER OTP 结构

Page	Word	Address	Contents
1	0	0x1FFF 1700	Bit[31:16]:存放用户数据 Bit[15:0]: USER OTP MEMORY LOCK
	1	0x1FFF 1704	存放用户数据
	…	…	存放用户数据
	64	0x1FFF 17FC	存放用户数据
2	0	0x1FFF 1800	Bit[31:16]:存放用户数据 Bit[15:0]: USER OTP MEMORY LOCK
	1	0x1FFF 1804	存放用户数据
	…	…	存放用户数据
	64	0x1FFF 18FC	存放用户数据

这 2 个 Page 配置在 information 区域，对本 Page 区域写和擦是按照 Main flash 的方法来处理。另外，Main flash 区域的 mass erase 对本区域无效。
设定 USER OTP MEMORY LOCK 内容不会立刻更新，直到上电复位（POR/PDR），从会起到保护功能。
对本 Page Write 有如下保护。

表 4-5 Flash USER OTP 写保护

USER OTP MEMORY_LOCK	Write protection
0xAA55	读：可以 program 和擦操作：不可以
除(0xAA55)之外的任何值	读、program 和擦操作：可以

4.7 Flash 存储区保护

对 FLASH main memory 的保护包括以下几种机制：

- 读保护（RDP），防止来自外部的访问。
- 写保护（WRP）控制，以防止不要的写操作（由于程序存储器指针的混乱）。写保护的粒度设计为 8 KB。
- 选项字节写保护，有专门的解锁设计。

Puya Confidential

4.7.1 闪存读保护（RDP）

通过设置 RDP 选项字节，并进行系统复位（POR 或者 OBL 复位）装载新的 RDP 选项字节，可以激活读保护功能。RDP 保护 Main flash memory 和备份寄存器。

如果通过 SWD 的 debug 仍在连接时，读保护被设置，需要进行上电复位而不是系统复位。

当 RDP 选项字节和反码成对正确存在于选项字节时，Main flash memory 会被保护。

表 4-6 闪存读保护状态

RDP 选项字节值	RDP 选项字节反码值	读保护等级
0xAA	0x55	Level 0
除[0xAA,0x55]、[0xCC,0x33]组合的任何值		Level 1
0xCC	0x33	Level 2

无论任何保护级别，System memory 都是可读的，但不能进行 program 和 erase 操作。

Level 0: no protection

对 Main flash 的读、编程和擦除操作是可以的，对选项字节和备份寄存器也是可以进行任何操作。

Level 1: Read protection

当选项字节里的 RDP 及其反码包含任何除了[0xAA,0x55], [0xCC,0x33] 之外的组合，则 level 1 读保护生效，Level 1 是默认的保护级别。

- 用户模式：在用户模式下执行的程序（Boot from Main flash），可以对 Main flash、选项字节和备份寄存器进行所有操作。
- Debug、从 SRAM 自举以及从 system memory 自举：在 debug 模式，或者当从 SRAM 或者 system memory 启动，Main flash 和备份寄存器是不能被访问的。在这些模式下，对 Main flash 读或者写访问会产生总线 HardFault 中断。

Level 2：禁止调试

在此级别下，具有保护级别 1 保护之外，在 debug 模式，或者当从 SRAM 或者 system memory 自举，不可再用。只能从 Main flash 自举，允许对 Flash 主存储器执行所有操作。

另外，可对选项字节执行读取操作。对选项字节进行编程或擦除时只能修改 FLASH_OPTBFBR.BFB 寄存器。因此，无法删除级别 2。尝试修改选项字节其他选项字节时，不会写入成功保持原值，该操作不会产生写错误。

注：在复位时调试功能也被禁止。

读保护级别可以改变：

- 从 Level 0 到 Level 1，将 RDP 字节的值更改为除 0xAA，0xCC 之外的任何值
- 从 Level 0 到 Level 2，将 RDP 字节的值更改为 0xCC
- 从 Level 1 到 Level 0，将 RDP 字节的值更改为 0xAA
- 从 Level 1 到 Level 2，将 RDP 字节的值更改为 0xCC

Level 1 变为 Level 0，会触发硬件 mass erase Main flash。

Level 2 无法改变成 0 或者 1

Puya Confidential

表 4-7 访问状态与保护级别和执行模式的关系

Area	Protection level	User execution (Boot from Flash)			Debug/ Boot from SRAM/ Boot from Loader		
		Read	Write	Erase	Read	Write	Erase
Main memory	0	Yes	Yes	Yes	Yes	Yes	Yes
	1	Yes	Yes	Yes	No	No	No
	2	Yes	Yes	Yes	**N/A**	**N/A**	**N/A**
System memory	x	Yes	No	No	Yes	No	No
选项字节	x	Yes	Yes	N/A	Yes	Yes	N/A
OTP	x	Yes	Yes	Yes	Yes	Yes	Yes
Backup registers	0	Yes	Yes	Yes	Yes	Yes	Yes
	1	Yes	Yes	Yes	No	No	No
	2	Yes	Yes	Yes	**N/A**	**N/A**	**N/A**

注:

(1) Information 区域只读，和保护等级及执行模式无关。

(2) RDP 从 Level 1 修改为 Level 0，会触发硬件对 Main flash 和 Backup register 的 mass erase。

4.7.2 闪存写保护（WRP）

Flash 可以被设置成写保护，以应对不想要的写操作。定义 WRP 寄存器每 bit 的控制粒度为 8 KB 的写保护（WRP）区域，即 1 个扇区大小。具体参见 WRP 寄存器的描述。

当 WRP 的区域被激活，则不允许进行 erase 或者 program 操作。相应的，即使只有一个区域被设定为写保护，则片擦功能不起作用。此外，如果尝试对设为写保护的区域进行擦除或编程操作，则 FLASH_SR 寄存器的写保护错误标识（WRPERR）会被置位。

注：写保护仅对 Main flash 起作用，对 system memory 不起作用。

4.7.3 专有代码读出保护（PCROP）

除了闪存之外，还可以防止来自第三方的读写。保护区仅执行：它只能由 CPU 作为指令代码访问，而严格禁止所有其他访问（DMA、调试和 CPU 数据读取、写入和擦除）。当 RDP 保护从级别 1 更改为级别 0 时，包括 PCROP 区域也会被擦除（请参阅更改读取保护级别）。

每个 PCROP 区域由与物理闪存地址相关的起始页和结束页地址定义。这些地址定义在 PCROP 地址寄存器 FLASH PCROP 起始地址寄存器（FLASH_PCROP0SR，FLASH_PCROP1SR）、FLASH PCROP 结束地址寄存器（FLASH_PCROP0ER，FLASH_PCROP1ER）。

通过 D-code 总线对 PCROP 保护区执行的任何读取访问都会触发 RDERR 标志错误。

任何受 PCROP 保护的地址也是写保护的，对这些地址之一的任何写访问都会触发 WRPERR。

任何 PCROP 区域也受到擦除保护。因此，对该区域中的页面的任何擦除都是不可能的（包括包含该区域的开始地址和结束地址的页面）。此外，如果一个区域受到 PCROP 保护，则不能执行软件 Bank Mass 擦除。

只有当 RDP 从级别 1 更改为级别 0 时，才能禁用 PCROP。如果通过修改用户选项试图清除 PCROP 或缩小 PCROP 区域，可启动选项编程，但 PCROP 区域保持不变。相反，可以增加 PCROP 区域。

Puya Confidential

表 4-8 PCROP 保护

PCROPx 寄存器值(x = 0,1)	PCROP 保护区
PCROPxSR >PCROPxER	无 PCROP 区域。
PCROPxSR <=PCROPxER	PCROPxSR 和 PCROPxER 之间的区域受到保护。

4.7.4 安全内存区域（SECPROT）

安全内存区域定义了一个只能在启动时执行一次的代码区域，除非发生新的复位，否则该区域不能再被执行。

安全内存区域的主要目的是保护 Flash 存储器中特定部分免受未经授权的访问。这使得实现软件安全服务成为可能，例如安全密钥存储或安全启动。安全内存区域位于主 FLASH 存储器中，专门用于执行受信任的代码。当该区域没有被保护时，安全内存区域的行为与主 FLASH 存储器的其余部分相同。当该区域被保护时（即设置了 FLASH_CR 寄存器的 SEC_PROT0/1 位），任何尝试在安全内存区域内进行编程或擦除的操作都会产生写保护错误（设置 WRPERR 标志），任何尝试读取该区域的操作都会产生读错误（设置 RDERR 标志）。

安全内存区域的大小由 FLASH_SECR0（或 FLASH_SECR2）寄存器的 SEC_SIZEER0[8:0]（或 SEC_SIZEER1[8:0]）定义。仅在 RDP Level 0 下可以修改。当从 RDP Level 1 切换到 Level 0 时，其内容也会被擦除。定义的可保护内存区域如下：

起始地址：Bank0(1)基地址（包含）

结束地址：Bank0(1)基地址 + (0x400 * SEC_SIZEER0(1))（不包含）

表 4-9 启动保护区域

SECERx 寄存器值(x = 0,1)	启动保护区
SEC_SECERx = 0x000	无启动保护区域。
0x000 < SEC_SIZEERx	Bank0(1)基地址到 Bank0(1)基地址 + (0x400 * SEC_SIZEER0(1))之间的区域受到保护。

4.7.5 强制从 Main flash 启动

为了提高安全性，FLASH_OPTR1 寄存器的 BOOT_LOCK 选项位允许强制系统从主闪存启动，从而无视其他启动选项。重置 BOOT_LOCK 只有在以下情况：

- RDP 设置为级别 0
- RDP 级别 1 修改为级别 0

4.8 Verify read 功能

Verify read 功能通过设置 FLASH_OPTR2. VERIFY_READ_EN 打开，使能以后，擦写完成时硬件会对写入的数据自动进行校验（如果是 sector erase 或 bank erase，会将当前的 sector 或 bank 全进行校验）。

如果校验时存在数据错误，硬件会自动再进行一次擦写，再次擦写的时间与电压值与之前会有不同。擦完以后，还会对硬件进行校验，如果还存在数据错误硬件还会进行擦写。直到擦写次数达到上限，硬件会将 FLASH_SR.VRERSERR 或 FLASH_SR.VRPRGERR 位置位。

如果置位了 FLASH_CR.VRERSERRIE/ FLASH_CR.VRPRGERRIE 还会产生中断。

注：使能 FLASH_OPTR2. VERIFY_READ_EN 后，RWW 功能会被关闭。

Puya Confidential

4.9 Flash 中断

表 4-10 Flash 中断请求

中断事件	事件标志	时间标志/中断清除方法	控制位使能
操作结束	EOP	Write EOP=1	EOPIE
写保护错误	WRPERR	Write WRPERR=1	ERRIE
选项字节写保护	OPTWERR	Write OPTWERR=1	ERRIE
Read error	RDERR	Write RDERR=1	RDERRIE
Erase verify read error	VRERSERR	Wirte VRERSERR=1	VRERSERRIE
program verify read error	VRPRGERR	Wirte VRPRGERR =1	VRPRGERRIE

以下事件没有单独的中断标识，但会产生 HardFault：

- 解锁 Flash memory 的 FLASH_CR 寄存器的流程错误
- 解锁 Flash 选项字节的写操作流程错误
- Flash program，erase 操作未进行 32 位数据的对齐
- 擦写过程中再次发起擦写请求

4.10 Flash 寄存器描述

4.10.1 Flash 访问控制寄存器（FLASH_ACR）

偏移地址：0x00

复位值：0x0000 0700

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	DCEN	ICEN	PRFTEN	Res.	Res.	Res.	Res.	LATENCY[3:0]			
					RW	RW	RW					RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:11	Reserved	-	-	保留
10	DCEN	RW	1	数据缓存使能 0：数据缓存禁止 1：数据缓存使能 注：该位的初始值，还受 FLASH_OPTR2.ACC_CTRL[2]位影响，如果 FLASH_OPTR2.ACC_CTRL[2]设为 0，则该位的默认值为 0
9	ICEN	RW	1	指令缓存使能 0：指令缓存禁止 1：指令缓存使能 注：该位的初始值，还受 FLASH_OPTR2.ACC_CTRL[1]位影响，如果 FLASH_OPTR2.ACC_CTRL[1]设为 0，则该位的默认值为 0
8	PRFTEN	RW	1	预取使能位 0：预取禁止

Puya Confidential

				1：预取使能 注：该位的初始值，还受 FLASH_OPTR2.ACC_CTRL 位[0]影响，如果 FLASH_OPTR2.ACC_CTRL[0]设为 0，则该位的默认值为 0
7:4	Reserved	-	-	保留
3:0	LATENCY[3:0]	RW	4'h0	Flash 读操作对应的等待状态： 0：Flash 读操作没有等待状态（HCLK <= 25 MHz） 1：Flash 读操作有 1 个等待状态（25 MHz < HCLK <= 50 MHz） 2：Flash 读操作有 3 个等待状态（50 MHz < HCLK <= 75 MHz） 3：Flash 读操作有 4 个等待状态（75 MHz < HCLK <= 100 MHz ） 4：Flash 读操作有 5 个等待状态（100 MHz < HCLK <= 125 MHz） 5：Flash 读操作有 6 个等待状态（125 MHz < HCLK <= 150 MHz） 6：Flash 读操作有 7 个等待状态（150 MHz < HCLK <= 175 MHz） 7：Flash 读操作有 8 个等待状态（175 MHz < HCLK） ... Flash 可以配置比上表设置更高的延迟，为了降低功耗。比如，在 HCLK >176 MHz 的频率条件下，设置 7~15 个等待状态。

4.10.2 Flash 密钥寄存器（FLASH_KEYR）

偏移地址：0x08

复位值：0x0000 0000

所有寄存器位是只写，读出返回 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEY[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:0	KEY[31:0]	W	32'h0	下面的值必须被连续的写入，才能解锁 FLASH_CR 寄存器，并使能 Flash 的编程/擦除操作 KEY1：0x4567 0123 KEY2：0xCDEF 89AB

4.10.3 Flash 选项密钥寄存器（FLASH_OPTKEYR）

偏移地址：0x0C

复位值：0x0000 0000

所有寄存器位是只写，读出返回 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEY[31:16]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPTKEY[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:0	OPTKEY[31:0]	W	32'h0	下面的值必须被连续的写入，才能解锁 Flash 的选项寄存器，并使能选项字节的编程/擦除操作 KEY1：0x0819 2A3B KEY2：0x4C5D 6E7F

Puya Confidential

4.10.4 Flash 状态寄存器（FLASH_SR）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
USRLOCK1	USRLOCK0	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BSY1	BSY0
R	R													R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTVERR	OPTWERR	VRERSERR	VRPRGERR	RDERR	Res.	Res.	Res.	Res.	Res.	Res.	WRPERR	Res.	Res.	Res.	EOP
RC_W1	RC_W1	RC_W1	RC_W1	RC_W1							RC_W1				RC_W1

Bit	Name	RW	Reset Value	Function
31	USRLOCK1	R	0	根据上电读 user data1 区域第一个 word 的低 16 位的值来指示 user data1 区域是否可写 0：读到的值不为 0xaa55，user data 可写 1：读到的值为 0xaa55，user data 不可写 该位仅能被上电复位所复位
30	USRLOCK0	R	0	根据上电读 user data0 区域第一个 word 的低 16 位的值来指示 user data0 区域是否可写 0：读到的值不为 0xaa55，user data 可写 1：读到的值为 0xaa55，user data 不可写 该位仅能被上电复位所复位
29:18	Reserved	-	-	保留
17	BSY1	R	0	Bank1 Busy 位 该位表示 Flash bank1 的操作正在进行。该位在 Flash 操作的开始被硬件置位，当操作完成或者错误产生时由硬件清零。
16	BSY0	R	0	Bank0 Busy 位 该位表示 Flash bank0 的操作正在进行。该位在 Flash 操作的开始被硬件置位，当操作完成或者错误产生时由硬件清零。
15	OPTVERR	RC_W1	0	选项字节加载错误 当选项字节及其反码不匹配时，硬件置位该位。装载不匹配的选项字节，被强制成安全值，参考 Flash 选项字节。 软件写 1，清零。
14	OPTWERR	RC_W1	0	选项写入错误 当选项字节写入无效配置时，硬件会设置此位。通过写入 1 可以清除此位。 会置位 OPTWERR 的情况如下： RDP =2 时，修改选项字节，写入 0x1FFF 1B00 地址。
13	VRERSERR	RC_W1	0	擦校验错误位 0：无擦校验读错误 1：擦校验读失败 软件写 1，清零。
12	VRPRGERR	RC_W1	0	写校验错误位 0：无写校验读错误

Puya Confidential

				1：写校验读失败 软件写 1，清零。
11	RDERR	RC_W1	0	Flash 读取错误 当要通过数据读取的地址属于 Flash 的读取保护区时由硬件置位（PCROP 保护，SECPRT 保护）。硬件置位该位。 写 1，清零该位
10:5	Reserved	-	-	保留
4	WRPERR	RC_W1	0	写保护错误 当要被编程/擦除的地址处于被写保护的 Flash 区域时（WRP），硬件置位该位。 软件写 1，清零。
3:1	Reserved	-	-	保留
0	EOP	RC_W1	0	当 FLASH 的编程/擦除操作成功完成，硬件置位。该位仅当 FLASH_CR 寄存器的 EOPIE 位使能才会被置位。 软件写 1，清零。

4.10.5 Flash 控制寄存器（FLASH_CR）

偏移地址：0x14

复位值：0xC000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPTLOCK	SECPROT1	SECPROT0	OBL_LAUNCH	Res	ERRIE	EOPIE	Res.	RDERRIE	VRERSERRIE	VRPRGERRIE	PGSTRT	UPGSTRT	OPTSTRT	Res.
RS	RS	RS	RS	RC_W1		RW	RW		RW	RW	RW	RW	RW	RW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SER	Res.	Res.	Res.	Res.	Res.	UPER.	UPG	Res.	MER	PER	PG
				RW						RW	RW		RW	RW	RW

Bit	Name	RW	Reset Value	Function
31	LOCK	RS	1	FLASH_CR 寄存器锁定位。 软件对该位只能置位。当置位后，FLASH_CR 寄存器被锁住。当成功给出解锁流程后，该位被硬件清零。 【软件要在编程/擦除操作完成后，置位该位】 如果解锁操作失败，该位仍然保持置位状态，直到下一次系统复位。
30	OPTLOCK	RS	1	选项字节锁定位。 软件对该位只能置位。当置位后，FLASH_CR 寄存器中与选项字节有关的位被锁住。当成功给出解锁流程后，该位被硬件清零。 【软件要在编程/擦除操作完成后，置位该位】 如果解锁操作失败，该位仍然保持置位状态，直到下一次系统复位。
29	SECPROT1	RS		bank1 内存区域保护位。 该位用于锁定 bank1 中安全区域的访问。当退出安全区域时，由软件设置该位，并且只能写入一次。

Puya Confidential

28	SECPROT0	RS		bank0 内存区域保护位。 该位用于锁定 bank0 中安全区域的访问。当退出安全区域时，由软件设置该位，并且只能写入一次。
27	OBL_LAUNCH	RC_W1	0	强制选项字节加载 当置位时，该位强制系统进行选项字节的重装载。该位仅当选项字节装载被完成后被硬件清零。如果 OPTLOCK 位被置位，该位不能被写. 0：选项字节加载完成 1：产生选项字节加载请求，产生系统复位，进行选项字节的重装载。
26	Reserved			保留
25	ERRIE	RW	0	错误中断使能 当 FLASH_SR 寄存器的 WRPERR 位被置位，如果该位使能，则产生中断请求。 0：无中断产生 1：有中断产生
24	EOPIE	RW	0	操作结束中断使能 当 FLASH_SR 寄存器的 EOP 位被置位，该位使能中断的产生。 0：EOP 中断关闭 1：EOP 中断使能
23	Reserved	-	-	保留
22	RDERRIE	RW	0	PCROP SECPRT 读取错误中断使能 当 FLASH_SR 中的 RDERR 位被置位，如果该位使能，则产生中断请求。 0：无中断产生 1：有中断产生
21	VRERSERRIE	RW	0	擦校验错误中断使能位 0: 擦校验失败不会产生中断 1: 擦校验失败会产生中断
20	VRPRGERRIE	RW	0	写校验错误中断使能位 0：写校验失败不会产生中断 1：写校验失败会产生中断
19	PGSTRT	RW	0	Main flash memory 的编程操作的启动位。 该位启动了 Main flash memory 的编程操作，软件置位，在 FLASH_SR 寄存器的 BSY0 或 BSY1 位被清零后，硬件清零该位。
18	UPGSTRT	RW	0	Flash 信息区的用户数据区编程操作的启动位。 该位启动了 Flash 信息区的用户数据区编程操作，软件置位，在 FLASH_SR 寄存器的 BSY0 位被清零后，硬件清零该位。
17	OPTSTRT	RW	0	Flash 选项字节修改的启动位

Puya Confidential

				该位启动了对选项字节的修改。软件置位，在 FLASH_SR 寄存器的 BSY0 位被清零后，硬件清零该位。 注意：当对 Flash 选项字节进行修改时，硬件自动把整个 page 进行 erase 操作，再进行 program 操作，其中也包括自动进行反码的写入。
16:12	Reserved	-	-	保留
11	SER	RW	0	Sector erase 操作 0：未选择 Flash 的扇区擦除操作 1：选择 Flash 的扇区擦除操作
10:6	Reserved	-	-	保留
5	UPER	RW	0	用户数据页擦除操作 0：未选择 Flash 用户数据页的页擦除操作 1：选择 Flash 用户数据页的页擦除操作 注：如果用户数据区不能擦写，则该配置无效。
4	UPG	RW	0	用户数据区编程操作 0：未选择用户数据区的编程操作 1：选择用户数据区的编程操作 注：1）如果用户数据区不能擦写，则该配置无效。 2）如果进入 Iprun 模式，该寄存器无法写入。
3	Reserved	-	-	保留
2	MER	RW	0	mass erase 操作 0：未选择 Flash 的 mass erase 操作 1：选择 Flash 的 mass erases 操作
1	PER	RW	0	Page erase 操作 0：未选择 Flash 的 page erase 操作 1：选择 Flash 的 page erase 操作
0	PG	RW	0	Program 操作 0：未选择 Flash 的 program 操作 1：选择 Flash 的 program 操作

4.10.6 Flash 选项 1 寄存器（FLASH_OPTR1）

偏移地址：0x20

复位值：0xXXXX XXXX。

在上电复位（POR/OBL_LAUNCH）释放后，从 Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	SWD_MODE		Res.	Res.	Res.	BOOT_LOCK	Res	nSW-BOOT0	nBOOT1	nBOOT0
						RW	RW				RW		RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WWDG_SW.	IWDG_SW	IWDG_STOP	IWDG_STDBY	Nrst_mode		nRST STOP	nRST STDBY	RDP[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留

Puya Confidential

25:24	SWD_MODE	RW	-	SWD 模式选择 00：PA11 作为 SWD 时钟 01：PA14 作为 SWD 时钟 其他：PA11 作为 SWD 时钟
23:21	Reserved	-	-	保留
20	BOOT_LOCK	RW	-	用于强制从用户 Flash 区域启动 0：基于 pad/选项位配置启动 1：从主闪存强制启动
19	Reserved	-	-	保留
18	nSWBOOT0	RW	-	软件 BOOT0 0：BOOT0 取自选项位 nBOOT0 1：BOOT0 取自 PD3/BOOT0 引脚
17	nBOOT1	RW	-	启动配置与 BOOT0 引脚决定从闪存主存储器、SRAM 或系统存储器中选择启动模式。 请参阅启动配置该节。
16	nBOOT0	RW	-	nBOOT0 选项位 0：nBOOT0=0 1：nBOOT0=1
15	WWDG_SW	RW	-	0：硬件窗口看门狗 1：软件窗口看门狗
14	IWDG_SW	RW	-	0：硬件独立看门狗 1：软件独立看门狗
13	IWDG_STOP	RW	-	设置 IWDG 在 Stop 模式下定时器运行状态 0：冻结定时器 1：正常运行
12	IWDG_STDBY	RW	-	设置 IWDG 在 Standby 模式下定时器运行状态 0：冻结定时器 1：正常运行
11:10	NRST_MODE	RW	-	PD11 pad mode 00：Reset 输入/输出 01：Reset 输入 10：GPIO 11：Reset 输入/输出
9	nRST_STOP	RW	-	0：进入 Stop 模式时产生 Reset 1：无 Reset 产生
8	nRST_STDBY	RW	-	0：进入 Standby 模式时产生 Reset 1：无 Reset 产生 该位一定要为 0，不可写 1
7:0	RDP	RW	-	0xAA：level 0，读保护无效 非 0xAA 和 0xCC：level 1，读保护有效 0xCC：level 2，读保护有效

4.10.7 Flash 选项 2 寄存器（FLASH_OPTR2）

偏移地址：0x24

复位值：0x0000 00XX。

Puya Confidential

在上电复位（POR/OBL_LAUNCH）释放后，从 Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VERIFY_READ_EN	Pll_unlock_rsten	ACC_CTRL		
											RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:5	Reserved	-	-	保留
4	VERIFY_READ_EN	RW	-	Verify read 使能 0：关闭 verify read 功能，打开 rww 功能 1：打开 verify read 功能，关闭 rww 功能
3	Pll_unlock_rsten	RW	-	Pll_unlock_rsten 要不要产生 reset 0：不产生 reset 1：产生 reset
2:0	ACC_CTRL	RW	-	ACC 上电初始值控制 ACC_CTRL [2]: 数据缓存使能控制 ACC_CTRL [1]: 指令缓存使能控制 ACC_CTRL [0]: 预取使能控制

4.10.8 Flash WRP 地址寄存器（FLASH_WRPR0）

偏移地址：0x28

复位值：0x0000 XXXX

在上电复位（POR/ OBL_LAUNCH）释放后，从 Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BK0_WRP[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	BK0_WRPx	RW	0	Bank0 sector 写保护 0：sector x(x=0~15)，有写保护，不允许进行编程和擦除操作 1：sector x(x=0~15)，无写保护

4.10.9 Flash PCROP 地址寄存器（FLASH_PCROPR0）

偏移地址：0x2C

复位值：0xXXXX XXXX

在上电复位（POR/ OBL_LAUNCH）释放后，从 Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP0ER[8:0]								
							RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Puya Confidential

Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP0SR[8:0]								
							RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24:16	PCROP0ER	RW	-	PCROP 区域结束地址(以 Page 为单位)
15:9	Reserved	-	-	保留
8:0	PCROP0SR	RW	-	PCROP 区域起始地址(以 Page 为单位)

4.10.10 Flash SECR 地址寄存器（FLASH_SECR0）

偏移地址：0x30

复位值：0xXXXX 0000

在上电复位（POR/ OBL_LAUNCH）释放后，从 Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	SEC_SIZE0[8:0]								
							RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24:16	SEC_SIZE0	RW	0	设置 bank0 安全内存区域使用的页数。 安全区域从 0x0800 0000 开始，其大小为 SEC_SIZE * 页大小。此字段只能在 level 0 中更改。在 level 1 中进行的任何修改都不会更改寄存器的值。
15:0	Reserved	-	-	保留

4.10.11 Flash WRP 地址寄存器（FLASH_WRPR1）

偏移地址：0x34

复位值：0x0000 XXXX

在上电复位（POR/ OBL_LAUNCH）释放后，从 Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BK_WRP1[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	BK_WRPx	RW	0	Bank sector 写保护 0：sector x(x=0~15)，有写保护，不允许进行编程和擦除操作

Puya Confidential

				1：sector x(x=0~15)，无写保护

4.10.12 Flash PCROP 地址寄存器（FLASH_PCROPR1）

偏移地址：0x38

复位值：0xXXXX XXXX

在上电复位（POR/ OBL_LAUNCH）释放后，从 Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1ER[8:0]								
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1SR[8:0]								
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24:16	PCROP1ER	RW		PCROP 区域结束地址(以 Page 为单位)
15:9	Reserved	-	-	保留
8:0	PCROP1SR	RW		PCROP 区域起始地址(以 Page 为单位)

4.10.13 Flash SECR 地址寄存器（FLASH_SECR1）

偏移地址：0x3C

复位值：0xXXXX 0000

在上电复位（POR/ OBL_LAUNCH）释放后，Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	SEC_SIZE1[8:0]								
							RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24:16	SEC_SIZE1	RW	0	设置 bank1 安全内存区域使用的页数。 安全区域从 0x0802 0000 开始，其大小为 SEC_SIZE * 页大小。此字段只能在 level 0 中更改。在 level 1 中进行的任何修改都不会更改寄存器的值。
15:0	Reserved	-	-	保留

4.10.14 Flash 选项字节 BFB 寄存器（FLASH_OPTBFBR）

偏移地址：0x60

复位值：0x0000 000X

在上电复位（POR/ OBL_LAUNCH）释放后，Flash 信息区的选项字节区域读出相应的值，写入到该寄存器相应的 option bit。

Puya Confidential

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

Bit	Name	RW	Reset Value	Function
31:1	Reserved	-	-	保留
0	BFB	RW	-	0：从 Bank0 启动(Bank0 映射在 0x0800 0000 Bank1 映射在 0x0802 0000) 1：从 Bank1 启动（Bank1 映射在 0x0800 0000 Bank0 映射在 0x0802 0000）

Puya Confidential

5. 电源控制（PWR）

5.1 简介

PWR 模块主要完成如下功能：

- 芯片上电和掉电流程控制
- 模拟 PMU 控制信号产生
- 存储器低功耗控制信号产生
- 低功耗模式进入和唤醒时序控制
- PVD 功能

5.2 电源结构

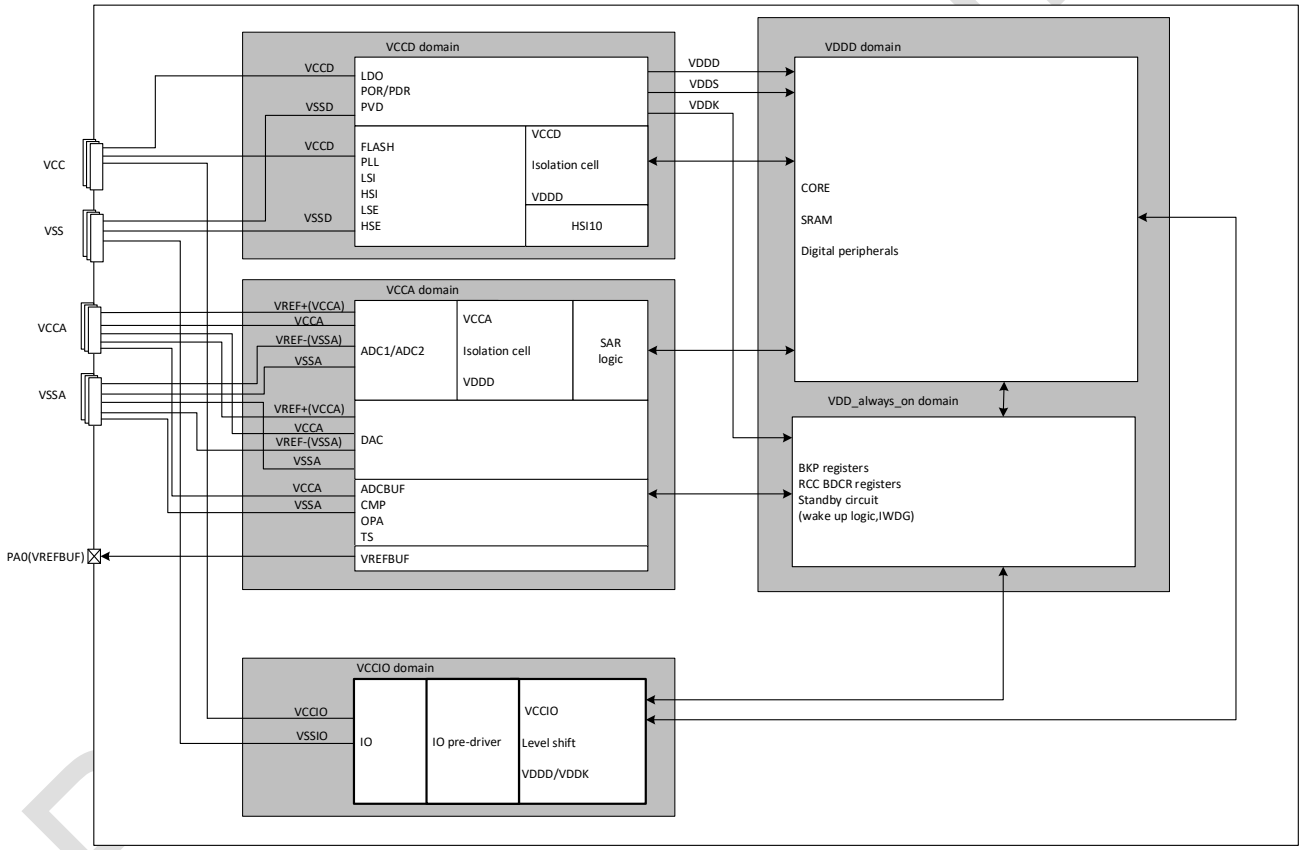


图 5-1 电源框图

- 芯片工作电压（V_{CC}）为 2.7 V ~ 5.5 V
- V_{DDD} 域数字逻辑的上电复位由 VDD_AON 产生
- 可以使用任意时钟进入低功耗模式，唤醒后时钟源会被切换为 HSI。（HSI 时钟需要被打开）
- Standby 模式下，可以配置使能 IWDG 工作，其计数时钟为 LSI，所以 LSI 时钟需要在待机模式下输出，即 LSI 时钟也会输出 VDD_AON 域时钟。V_{DDD} 输出的相关信号需要在 V_{DDD} 掉电时隔离为配置值

5.2.1 系统主要电源

5.2.1.1 独立 A/D 转换器电源和参考电压

为了提高转换精度，ADC 配有独立电源，可以单独滤波并屏蔽 PCB 上的噪声。

- ADC 电压源从单独的 V_{CCA} 引脚输入。
- V_{SSA} 引脚提供了独立的电源接地连接。

为了提高低压输入的精度，用户可以在 V_{REF} 上连接单独的 ADC 外部参考电压输入。

备份域

- RCC_BDCR 寄存器
- RTC 模块(RTC_CR 寄存器除外)
- BKP 寄存器
- LSE 振荡器

备份域访问

复位后，备份域（RTC 寄存器和 RTC 备份寄存器）将受到保护，以防止意外的写访问。要使能对备份域（RTC 和 RTC 备份寄存器）的访问，需要按如下步骤配置：

- 将 RCC_APBENR4 寄存器中的 PWREN 位置 1，使能电源接口时钟
- 将 PWR_CR1 寄存器中的 DBP 位置 1，使能对备份域的访问
- 配置 RCC 备份域控制寄存器（RCC_BDCR）中的 RTCSEL 寄存器选择 RTC 时钟源
- 配置 RCC 备份域控制寄存器（RCC_BDCR）中的 RTCEN，使能 RTC 时钟

V_{DDD} 调压器

嵌入式线性 VR 为备份域和待机（唤醒）电路以外的所有数字电路供电。

通过软件激活时，调压器在复位后始终处于使能状态。根据应用模式的不同，可采用三种不同的模式工作：

- 在 Run 模式中，调压器为内核、存储器和数字外设提供全功率。在该模式下，调压器输出电压可通过软件调节为不同的电压值（可通过 PWR_CR 寄存器的 MR_VSEL 位配置）。
- 在停止模式下，主调压器或低功耗调压器为 V_{DDD} 域提供低功率电压，从而保存寄存器和内部 SRAM 的内容。可以将调压器置于主调压器模式（MR）或低功耗模式（LPR）。
- 在待机模式中，调压器掉电。除待机（唤醒）电路和备份域外，SRAM 寄存器的内容是否保持，可以由 PWR_CR3.RRS 控制。

5.2.2 PWR 电源检测

POR/PDR

芯片包含 power-on reset（POR）、power-down reset（PDR）模块，提供电源相关复位。其中 POR/PDR 在所有 power 模式下都默认工作。POR/PDR 输出复位低电平有效。

Puya Confidential

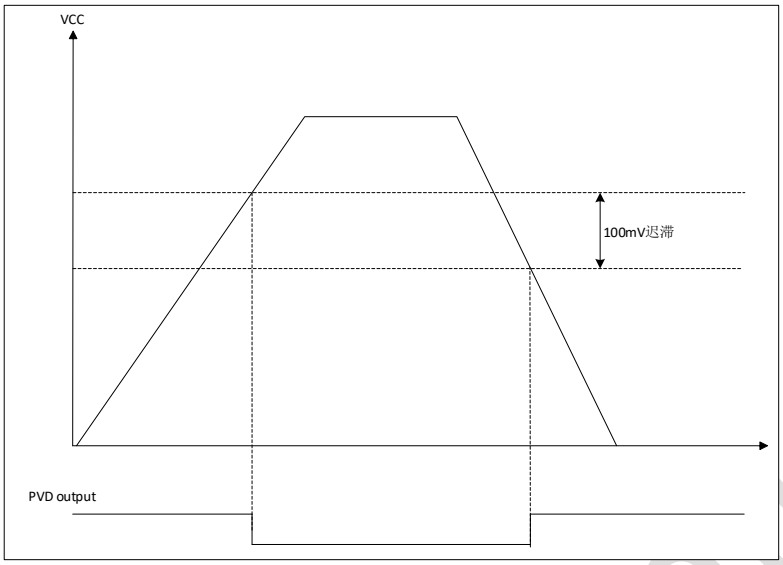


图 5-2 POR/PDR 阈值

PVD

PVD 模块是检测 V_{CC} 是否低于 PWR_CR2.PVDT 设置的阈值。

通过配置 PWR_CR2.PVDE 寄存器使能 PVD 功能。

电源控制/状态寄存器（PWR_CSR）中的 PVDO 标志用来表明 V_{CC} 是高于还是低于 PVD 的电压阀值。该事件在内部连接到 EXTI 的输入，如果该中断在外部中断寄存器中是使能的，该事件就会产生中断。当 V_{CC} 下降到 PVD 阀值以下或当 V_{CC} 上升到 PVD 阀值之上时，根据 EXTI 的上升/下降沿触发设置，就会产生 PVD 中断。实际应用中，这一特性可用于执行紧急关闭任务。

动态电压调节

该设备采用嵌入式线性稳压器——主稳压器（MR）为大部分数字电路供电。主稳压器在运行模式和睡眠模式下工作。

该设备支持动态电压调节功能，可在运行模式下优化功耗。为逻辑电路供电的主稳压器输出电压，可根据系统最高运行频率进行动态调整。

主调节器（MR）在以下范围内运行：

- 高电压（Boost）模式，CPU 运行频率最高可达 176 MHz。
- 中电压（Normal）模式，CPU 运行频率最高可达 144 MHz。

MR 模式从低电压切换到高电压：

MR 模式下，系统时钟频率从低频到高频，电压从低压到高压的切换步骤如下：

1）电源控制寄存器 1 (PWR_CR1) 中的 LPMS_ACTIVE 位配置为 0

2）根据支持的频率，参考 MR_VSEL 中支持的频率配置合适的 MR 电压模式

3）等待 5 μs

4）根据支持的频率配置 FLASH_ACR.LATENCY 的值

5）切换系统时钟频率

MR 模式从高电压切换到低电压：

MR 模式下，系统时钟频率从高频到低频，电压从高压到低压的切换步骤如下：

1）电源控制寄存器 1（PWR_CR1）中的 LPMS_ACTIVE 位配置为 0

Puya Confidential

2）切换系统频率

3）根据支持的频率配置 FLASH_ACR.LATENCY 的值

4）根据支持的频率，参考 MR_VSEL 中支持的频率配置合适的 MR 电压模式

5.3 系统低功耗模式

默认情况下，系统复位或上电复位后，微控制器进入运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。系统提供了多个低功耗模式，可在 CPU 不需要运行时（例如等待外部事件时）节省功耗。由用户根据应用选择具体的低功耗模式，在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡。

器件有五种低功耗模式：

- **睡眠模式**（Sleep mode）：CPU 时钟关闭，内核停止工作，外设包括 Cortex®-M4 的外设（NVIC，Sys Tick）等可以配置为运行；
- **低功耗运行模式**（Low-power run）：HCLK 配置为低于 2 MHz，V_{DDD} 由 LPR 供电(FLASH 不处于低功耗模式)，代码从 SRAM 或 Flash 存储器中执行。调节器处于低功耗模式，以最小化调节器的运行电流。
- **低功耗睡眠模式**（Low-power sleep）：从 LP run 进入 LP sleep 模式，内核的时钟关闭；
- **停止模式**（Stop）：SRAM 和所有的寄存器内容被保持，在 V_{DDD} 域的时钟（HSI/HSE/PLL）被关闭。LSI/LSE 仍然可以工作。
  - 一些具有唤醒功能的外设，可以在停止模式下开启 HSI16 振荡器来检测唤醒条件。
  - 如果设备被配置为唤醒后进入 LP run 模式，需要保证系统时钟 HCLK 配置为低于 2 MHz。
- **待机模式**（Standby）：V_{DDD} 域被关闭。可以选择保留 SRAM 的内容：
  - 当 PWR_CR3.RRS=1 时，进入待机模式并保持 SRAM 内容。在这种情况下，SRAM 由 VDD_STDBY 供电。
  - 当 PWR_CR3.RRS=0 时，进入待机模式。在这种情况下，VDD_STDBY 也被关闭。
- V_{DDD} 域所有时钟关闭，PLL/HSI/HSE 被关闭；LSI 和 LSE 可以根据唤醒源选择是否开启。
- VDD_AON 电源域外设仍然可以工作。当退出 Standby 模式，系统时钟仍然为 HSI 时钟。
- 此外，可通过下列方法之一降低运行模式的功耗：
  - 降低系统时钟速度
  - 不使用 APB 和 AHB 外设时，将对应的外设时钟关闭。

Puya Confidential

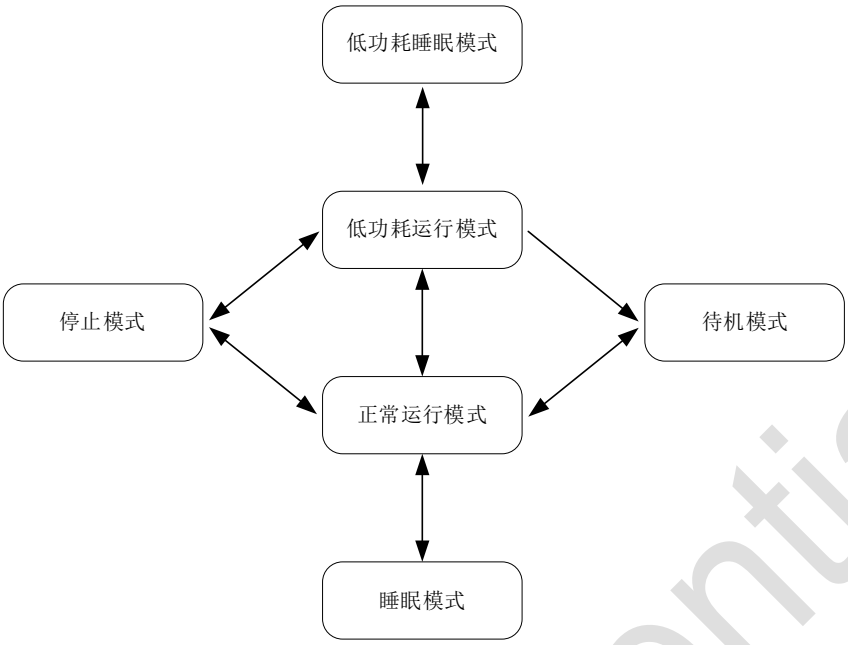


图 5-3 低功耗模式状态转移图

表 5-1 低功耗模式概要

大模式	模式	时钟状态	进入条件	退出条件	唤醒延时	唤醒后系统时钟	MR	LPR	DLPR
							状态	状态	状态
RUN 模式	RUN	任意时钟	-	-	-	-	开	关	关
SLEEP 模式	SLEEP	CPU 时钟关闭	WFI 或者从中断返回 SLEEPDEEP = 0	任意中断	6 个系统时钟 (1)	进入前时钟			
			WFE	唤醒事件				关	
LP 模式	LPRUN	系统时钟:HSI8M 4 分频	LPMS_ACTIVE=1	LPMS_ACTIVE=0	2us(2)		关	开	
	LPSLEEP	CPU 时钟关闭	WFI 或者从中断返回 SLEEPDEEP = 0 LPMS_ACTIVE=1	任意中断	6 个系统时钟 (1)				
			WFE LPMS_ACTIVE=1	唤醒事件			关		
STOP 模式	STOP0	HSI/LSI/LSE 可以选择打开 其他时钟关闭	LPMS_SLP=0 SLEEPDEEP=1 WFI 或者从中断返回或者 WFE	任意 EXTI 特定外设唤醒事件	7us（3）	HSI	开	关	关
	STOP1	除 LSI/LSE 外 其他时钟关闭	LPMS_SLP=0 SLEEPDEEP=1 WFI 或者从中断返回或者 WFE		7/9us（4）		关	开	
STANDBY 模式	sram 保持	除 LSI/LSE 外 其他时钟关闭	LPMS_SLP=1 SLEEPDEEP=1 RRS=1 WFI 或者从中断返回或者 WFE	WKUP IO RTC PIN 复位 IWDG 复位	22us(5)	HSI	关	关	开

Puya Confidential

	sram 不保持	除 LSI/LSE 外其他时钟关闭	LPMS_SLP=1 SLEEPDEEP=1 RRS=0 WFI 或者从中断返回或者 WFE				关	关	关

注：

1.系统时钟为进入 SLEEP 前配置的 sys_clk（参考 RCC 模块）周期。

下表为不同模式下，各个功能模块是否存在的情况：注 (1)

表 5-2 各工作模式下的功能

Peripheral	Run	Sleep	LP Run	LP Sleep	Stop0 (MR 供电)		Stop1 (LPR 供电)		Standby	
					工作	唤醒	工作	唤醒	工作	唤醒
CPU Core	Y	-	Y	-	-	-	-	-	-	-
Flash memory	Y	Y(2)	Y	Y(2)	-	-	-	-	-	-
SRAM	Y	Y(3)	Y	Y(3)	Y	-	Y	-	O(4)	-
BKP	Y	Y	Y	Y	Y	-	Y	-	Y	-
PVD	O	O	O	O	O	O	O	O	-	-
DMA	O	O	O	O	-	-	-	-	-	-
HSI8/16/24/48M	O	O	O	O	O(5)	-	-	-	-	-
HSE	O	O	O	O	-	-	-	-	-	-
LSI	O	O	O	O	O	-	O	-	O	-
LSE	O	O	O	O	O	-	O	-	O	-
PLL	O	O	-	-	-	-	-	-	-	-
HSE Clock Security System (CSS)	O	O	O	O	-	-	-	-	-	-
RTC	O	O	O	O	O	O	O	O	O	O
Number of TAMP pin	1	1	1	1	1	O	1	O	1	O
USART1	O	O	O	O	O	-	O	-	-	-
UARTx（x=1,2）	O	O	O	O	-	-	-	-	-	-
LPUART1	O	O	O	O	O(6)	O(6)	O(6)	O(6)	-	-
I2Cx(x=1,2)	O	O	O	O	O(7)	O(7)	O(7)	O(7)	-	-
SPIx(x=1,2)	O	O	O	O	-	-	-	-	-	-
VREFBUF	O	O	O	O	-	-	-	-	-	-
OPAx(x=1,2)	O	O	O	O	-	-	-	-	-	-
COMPx(x=1,2)	O	O	O	O	O	O	O	O	-	-
Temperature sensor	O	O	O	O	-	-	-	-	-	-
Timers	O	O	O	O	-	-	-	-	-	-
PWMx（x=1,2,3）	O	O	O	O	-	-	-	-	-	-
LPTIM	O	O	O	O	O	O	O	O	-	-
IWDG	O	O	O	O	O	O	O	O	O	O
WWDG	O	O	O	O	-	-	-	-	-	-
SysTick timer	O	O	O	O	-	-	-	-	-	-
CRC	O	O	O	O	-	-	-	-	-	-
GPIOs	O	O	O	O	O	O	O	O	O(8)	O(9)

Puya Confidential

注：

1. Y = YES(ENABLE)；O= Optional(默认关闭，可以软件使能)； - = not available
2. FLASH 在 SLEEP 和 LP_SLEEP 模式下，可由软件选择时钟是否打开；
3. SRAM 时钟可被门控开或者关；
4. SRAM 可以被保持；
5. 在 Stop0 模式下，HSI 可由软件控制开关；
6. LPUART 在 Stop 模式下可以接收，产生 START 和地址匹配中断唤醒
7. I^2C 在 Stop 模式下提供地址匹配和超时中断唤醒
8. 在 standby 模式下，IO 可以配置为上拉下拉或者浮空
9. Standby 模式下具有唤醒功能的 IO：PA0、PC13、PB5、PA2、PC5

5.3.1 正常运行模式

在正常运行模式下，可以通过降低系统时钟（SYSCLK，HCLK，PCLK）来降低功耗；在进入 SLEEP 模式前，也可以通过分频来降低外设时钟频率；

在正常运行模式下，不同的外设和存储器，都可以通过门控单独关闭时钟，达到降低功耗作用；在进入 Sleep 前，可以关闭不是用作唤醒的外设时钟；

5.3.2 低功耗运行模式

为了进一步降低运行模式下的功耗，可以关闭 MR，V_{DDD} 域由 LPR 供电。同时切换系统时钟为 HSI 8M, 降低 HCLK 的工作频率,不能超过 2 MHz；注意：若发生系统复位退出 LPRUN 模式，硬件会强制把 LDO 切换至 MR 模式，同时把时钟切换为 HSI，之后软件复位重新执行。

低功耗运行模式下的 IO 状态

在低功耗运行模式下，所有 IO 状态保持和运行状态一致。

低功耗运行模式的进入

1. 可选：程序跳转到 SRAM 运行，同时配置 Flash 进入低功耗模式；
2. 降低系统时钟，要低于 2 MHz；
3. 设置 V_{DDD} 使用 LPR 供电，MR 关闭。

低功耗运行模式的退出

1. 设置 V_{DDD} 使用 MR 供电；
2. 等待 MR 上电完成；
3. 恢复正常的系统时钟。

表 5-3 低功耗运行模式

低功耗运行	描述
模式进入	降低 HCLK 低于 2MHz LPMS_ACTIVE = 1
模式退出	LPMS_ACTIVE = 0 等待 MR 上电完成 恢复正常时钟
唤醒延时	MR 开启到稳定时间

Puya Confidential

5.3.3 低功耗模式进入和退出

进入：

MCU 通过执行 WFI（等待中断）或 WFE（等待事件）指令进入低功耗模式，也可通过在从 ISR 返回时将 Cortex®-M4 系统控制寄存器中的 SLEEPONEXIT 位置 1 进入低功耗模式。

仅当没有中断或事件挂起时，才可通过 WFI 或 WFE 进入低功耗模式，即 CPU 在进入低功耗前需要清零挂起中断或者事件。

在进入低功耗模式的进程中，出现唤醒信号，系统会根据当前状态判断如何处理：

1. 在系统时钟关闭之前，退出进入流程，处理唤醒中断或事件；
2. 在系统时钟关闭之后，先保持住唤醒信号，再正常进入低功耗模式，进入低功耗模式后，再按正常唤醒流程唤醒。

退出：

MCU 根据进入睡眠和停止模式的方式退出这两种低功耗模式：

- 如果使用 WFI 指令或从 ISR 返回进入低功耗模式，通过 NVIC 确认的任何外设中断都能唤醒器件。
- 如果使用 WFE 指令进入低功耗模式，MCU 将在有事件发生时立即退出低功耗模式。唤醒事件可通过以下方式产生：
  - NVIC IRQ 中断：
  - SEVONPEND = 0 时，通过使能 NVIC 和对应的外设的中断使能控制位，可以从 WFE 唤醒。当系统唤醒后，必须清除 NVIC 和外设中的中断挂起位；
  - SEVONPEND = 1 时，只用使能外设的中断使能控制位，NVIC 中的中断使能可以选择是否使能，就可以从 WFE 唤醒。当系统唤醒后，必须清除外设中的中断挂起位，如果 NVIC 中的中断使能，也要同时清除；
  - 所有的 NVIC 中断都可以唤醒 CPU，NVIC 中没有使能的也可以。
  - 事件：

    配置一个外部或内部的 EXTI 线为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置 1，不必清零 EXTI 外设的中断挂起位或 NVIC IRQ 通道挂起位。可能需要清零外设中的中断标志。该模式唤醒所需的时间最短，因为没有时间花费在中断的进入或退出上。

当发生外部复位（NRST 引脚）、IWDG 复位、使能的 WKUPx 引脚上出现上升沿或者触发 RTC 唤醒事件时，MCU 退出待机低功耗模式。

从待机模式唤醒后，程序将按照复位（启动 boot 引脚采样、复位向量获取等）后的方式重新执行。

5.3.4 睡眠模式

进入睡眠模式之前，必须清零所有的中断挂起位。

睡眠模式下的 IO 状态

在睡眠模式下，所有 IO 状态保持和运行状态一致。

表 5-4 睡眠模式

睡眠模式	描述
模式进入	WFI 或 WFE，并且： --SLEEPDEEP=0 以及 --没有中断挂起。
	从最低优先级的 ISR 返回，并且：

Puya Confidential

	--SLEEPDEEP=0 以及 --SLEEPONEXIT=1 --没有中断挂起。
模式退出	使用 WFI 或从最低优先级 ISR 返回进入：中断（使能） 使用 WFE 进入且 SEVONPEN=0：唤醒事件 使用 WFE 进入且 SEVONPEN=1：外设使能中断（即使 NVIC 中禁止）
唤醒延时	无

5.3.5 低功耗睡眠模式

在低功耗睡眠模式下，所有 IO 状态保持和运行状态一致。

表 5-5 低功耗睡眠模式

低功耗睡眠	描述
模式进入	低功耗睡眠模式只能从低功耗运行模式进入 WFI 或 WFE，并且： --SLEEPDEEP=0 --没有中断挂起。
	低功耗睡眠模式只能从低功耗运行模式进入 从最低优先级的 ISR 返回，并且： --SLEEPDEEP=0 --没有中断挂起。
模式退出	使用 WFI 或从最低优先级 ISR 返回进入：中断（使能） 使用 WFE 进入且 SEVONPEN=0：唤醒事件 使用 WFE 进入且 SEVONPEN=1：外设使能中断（即使 NVIC 中禁止） 从低功耗睡眠模式退出，系统进入低功耗运行模式。
唤醒延时	无

5.3.6 停止模式

在停止模式下，V_{DDD} 域的所有高频时钟都会停止，停止的时钟包括 PLL、HSE 和 HSI，停止顺序为 PLL->HSE->HSI。一些有唤醒能力的外设（I2C、LPUART）可以打开 HSI 接收数据帧，如果接收到的数据帧不是唤醒帧则关闭 HSI。在这种情况下，HSI 只提供给到发出请求的外设。

停止模式下的 IO 状态

在停止模式下，所有 IO 状态保持和运行状态一致。

停止模式的进入

如果正在执行 Flash 编程，停止模式的进入将延迟到存储器访问结束后执行（通过软件保证）。

进入停止模式前，如果正在访问 APB 域，停止模式的进入将延迟到 APB 访问结束。（通过软件保证）

在停止模式 0 下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗（IWDG）：IWDG 通过写入其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位。
- 实时时钟（RTC）：通过 RCC 备份域控制寄存器（RCC_BDCR）中的 RTCEN 位进行配置。
- 内部 RC 振荡器（LSI RC）：通过 RCC 备份域控制寄存器（RCC_BDCR）中的 LSION 位进行配置。
- 外部振荡器（LSE OSC）：通过 RCC 备份域控制寄存器（RCC_BDCR）中的 LSEON 位进行配置。

Puya Confidential

一些时钟源为 LSI/LSE 或者可以请求唤醒 HSI 的外设可以在 Stop 模式下工作，但是会增加功耗。

停止模式的退出

退出停止模式，将选择 HSI 作为系统时钟。如果配置为退出停止模式后，进入低功耗运行模式，需要在进入停止模式前，将系统时钟配置为低于 2 MHz；同时选择 V_{DDD} 的供电方式 MR（Stop0）或者 LPR（Stop1）。

在进入 Stop 时，根据配置的 V_{DDD} 供电模式不同，启动后需要的延时不同。

表 5-6 停止模式

停止模式	描述
模式进入	WFI 或 WFE，并且： --没有中断（针对 WFI）或者事件（针对 WFE）挂起； --SLEEPDEEP=1； --将 PWR_CR 寄存器中的 LPMS_SLP 配置为 00（STOP0）
	从最低优先级的 ISR 返回，并且： --SLEEPDEEP=1； --没有中断挂起； --将 PWR_CR 寄存器中的 LPMS 配置为 00（STOP0）
模式退出	使用 WFI 或从 ISR 返回进入：任何配置为中断模式的 EXTI line（同时使能 NVIC 中对应的 EXTI 中断）。中断源为外部中断或者具有唤醒功能的外设产生的中断。 使用 WFE 进入且 SEVONPEND=0：任何配置为事件模式的 EXTI line。 使用 WFE 进入且 SEVONPEND=1： --任何配置为中断模式的 EXTI line（即使在 NVIC 中断对应的 EXTI 中断禁止）。中断源为外部中断或者具有唤醒功能的外设产生的中断。 --唤醒事件 NRST 复位 IWDG 复位
唤醒延时	HSI 唤醒时间+LDO 切换的时间+Flash 唤醒的时间（根据配置时间可有重合）

5.3.7 待机模式

待机模式基于核的 SLEEPDEEP 来控制，并且调压器关闭（除非 SRAM 内容保持）。通过配置 PWR_CR3.RRS 来控制 SRAM 内容是否保持。在这种情况下，超低功耗调压器打开，只用于 SRAM 内容保持。PLL，HSI，HSE 都关闭，其中 PLL 和 HSE 由软件控制关闭，HSI 由硬件关闭。

除 VBKP 域（RTC 寄存器和备份寄存器）和待机电路的寄存器外，根据配置 SRAM1 可选择是否由 PD_STDBY 单独供电。

待机模式下的 IO 状态

在待机模式，IO 可以配置为上拉下拉，或者模拟模式；

WKUP IO 和 RST IO，都处于工作状态；

待机模式的进入

在待机模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗（IWDG）：IWDG 通过写入其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位。
- 内部 RC 振荡器（LSI RC）：通过 RCC 备份域控制寄存器（RCC_BDCR）中的 LSION 位进行配置。

Puya Confidential

- 外部 32.768 kHz 振荡器（LSE OSC）：通过 RCC 备份域控制寄存器（RCC_BDCR）中的 LSEON 位进行配置。

表 5-7 待机模式

待机模式	描述
模式进入	WFI 或 WFE，并且： --没有中断（针对 WFI）或者事件（针对 WFE）挂起； --SLEEPDEEP=1； --将 PWR_CR 寄存器中的 LPMS 配置为 11； --配置 PWR_CSR 寄存器中的 WUF 位清零； --将与选择唤醒源对应的标志清零；
	从 ISR 返回，并且： --SLEEPDEEP=1； --SLEEPONEXIT=1； --没有中断挂起； --将 PWR_CR 寄存器中的 LPMS 配置为 11； --将 PWR_CSR 寄存器中的 WUF 位清零； --将与选择唤醒源对应的标志清零；
模式退出	WKUPx 引脚（x=1，2，3，4，5）
	RTC 唤醒事件产生
	NRST 引脚复位
	IWDG 复位
唤醒延迟	HSI 唤醒时间+LDO 上电时间+LDO 等待时间（可配置 0 μs/5 μs/10 μs/20 μs）+Flash 等待时间 3 μs（根据配置时间可有重合）+内部校准值加载时间

在待机模式下，除以下各部分外，其余 IO 都处于高阻态：

- 复位引脚
- 被设置为防侵入或者校准输出时的 TAMPER 引脚
- 5 个 WKUP 引脚（PA9/PA7/PA0/PA10/PC13）（如果使能）

5.3.8 停止模式和待机模式的调试

在停止和待机模式，默认情况下，调试功能将中断，因为 CPU 核无时钟。但如果配置了 DBGMCU_CRx 中相关寄存器，则即使 CPU 进入 SLEEPDEEP 模式，仍可进行调试。

Puya Confidential

5.4 寄存器描述

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

5.4.1 电源控制寄存器 1（PWR_CR1）

偏移地址：0x00

复位值：0x0200 0000

该寄存器可由 POR 复位；

该寄存器在退出待机模式后会被复位；

该寄存器可由系统复位清除；

该寄存器可由 PWR 软复位清除；

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	STDBY_MRRDY_WAIT		HSION_CTRL	Res.	Res.	Res	Res	Res.	Res.	Res.	Res.	MR_VSEL
				RW	RW	RW									RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	FLS_WUPT		Res.	Res.	Res.	DBP	Res.	Res.	Res.	VR_MODE_SLP	Res.	Res.	LPMS_SLP	LPMS_ACTIVE
		RW	RW				RW				RW			RW	RW

Bit	Name	RW	Reset Value	Function
31:28	Reserved			保留
27:26	STDBY_MRRDY_WAIT	RW	2'b00	待机模式唤醒，MR ready 后，需要等待的时间。 00：3 μs；01：7 μs；10：15 μs；11：31 μs
25	HSION_CTRL	RW	1'b1	从停止模式唤醒时，HSI 打开时间控制。 0：等待 MR 稳定后，使能 HSI； 1：与 VR 同时打开，即唤醒时立刻使能 HSI。
24:17	Reserved			保留
16	MR_VSEL	RW	1'b0	MR 模式电压选择。 0：Normal 电压模式：144 MHz 1：Boost 电压模式：176 MHz
15:14	Reserved		2'b00	保留
13:12	FLS_WUPT	RW	2'b00	停止模式唤醒时序中，在 HSI8M 稳定后，在 FLASH 操作前需要等待时间。 00：3 μs；01：5 μs；10：2 μs；11：0 μs；
11:9	Reserved			保留
8	DBP	RW	1'b0	backup 域寄存器写保护。 0：不可访问 RTC 备份寄存器； 1：可以访问 RTC 备份寄存器；
7:5	Reserved		3'b00	保留

Puya Confidential

4	VR_MODE_SLP	RW	1'b0	Stop 模式时，调压器工作模式，进入低功耗模式后自动切换 0：MR 模式 1：LPR 模式
3:2	Reserved		2'b00	保留
1	LPMS_SLP	RW	1'b0	低功耗模式选择 0：进入 Stop 模式 1：进入 Standby 模式
0	LPMS_ACTIVE	RW	1'b0	低功耗模式选择 0：Run 模式 1：Low-power run 模式

5.4.2 电源控制寄存器 2（PWR_CR2）

偏移地址：0x04

复位值：0x0000 0000

该寄存器可由 POR 复位；

该寄存器在退出待机模式后会被复位；

该寄存器可由系统复位清除；

该寄存器可由 PWR 软复位清除；

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	FLT_CTRL			FLTDIS	Res.	Res.	Res.	Res.	PVDT			PVDE
				RW			RW					RW			RW

Bit	Name	RW	Reset Value	Function
31:12	Reserved	-	-	-
11:9	FLT_CTRL	RW	3’b000	PVD 滤波时间配置。 111：保留； 110：滤波时间大约为 1024 个滤波时钟（LSI 或者 LSE 时，大约为 30.7 ms）； 101：滤波时间大约为 128 个滤波时钟（LSI 或者 LSE 时，大约为 3.8 ms）； 100：滤波时间大约为 64 个滤波时钟（LSI 或者 LSE 时，大约为 1.92 ms）； 011：滤波时间大约为 16 个滤波时钟（LSI 或者 LSE 时，大约为 480 μs）； 010：滤波时间大约为 4 个滤波时钟（LSI 或者 LSE 时，大约为 120 μs）； 001：滤波时间大约为 2 个滤波时钟（LSI 或者 LSE 时，大约为 60 μs）； 000：滤波时间大约为 1 个滤波时钟（LSI 或者 LSE 时，大约为 30 μs）；

Puya Confidential

8	FLTEN	RW	0	PVD 数字滤波使能。 0：PVD 数字滤波禁止； 1：PVD 数字滤波使能；
7:4	Reserved	-	-	-
3:1	PVDT[2:0]	RW	0	PVD 电平选择。 由软件写入，用于选择 PVD 的电压阈值。 000：保留 001：VPVD1 (around 2.9 V) 010：VPVD2 (around 3.05 V) 011：VPVD3 (around 3.3 V) 100：VPVD4 (around 3.75 V) 101：VPVD5 (around 4.0 V) 110：VPVD6 (around 4.25 V) 111：VPVD7 (around 4.5 V) 当 SYSCFG.CFG 中的 PVD_LOCK=1，则 PVDT 写保护。 当 SYSCFG.CFG 中的 PVD_LOCK=0，可以配置 PVDT 注：一旦 PVD_LOCK =1 后，只有系统复位才能解决写保护
0	PVDE	RW	0	PVD 使能 0：禁止 PVD； 1：使能 PVD； 当 SYSCFG.CFG 中的 PVD_LOCK=1,则 PVDE 写保护。 当 SYSCFG.CFG 中的 PVD_LOCK=0，可以配置 PVDE 注：一旦 PVD_LOCK =1 后，只有系统复位才能解决写保护

5.4.3 电源控制寄存器 3（PWR_CR3）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRS.
															RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWUPx				
											RW	RW	RW	RW	RW

该寄存器可由 POR 复位；

该寄存器在退出待机模式后会被复位；

该寄存器不可由系统复位清除；

该寄存器不可由 PWR 软复位清除；

Bit	Name	RW	Reset Value	Function
31:17	Res.	-	-	-
16	RRS	RW	0	SRAM 在 Standby 模式下是否保持 0：不保持 1：保持

Puya Confidential

15:5	Res.	-	-	-
4:0	EWUPx	RW	0	使能 WKUPx 引脚.(X=1-5) 此位由软件置 1 和清零。 0：WKUPx 引脚用于通用 I/O。 WKUP 引脚上的事件不会从待机模式唤醒器件。 1：WKUPx 引脚用于从待机模式唤醒，唤醒沿由 WPx 决定。 WKUP 1：PC13 WKUP 2：PA10 WKUP 3：PA0 WKUP 4：PA7 WKUP 5：PA9

5.4.4 电源控制寄存器 4（PWR_CR4）

偏移地址：0x0

复位值：0x0000 0000

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

该寄存器可由 POR 复位；

该寄存器在退出待机模式后会被复位；

该寄存器不可由系统复位清除；

该寄存器不可由 PWR 软复位清除；

Bit	Name	RW	Reset Value	Function
31:10	Res.	-	-	-
9:0	WUPx	RW	0	WKUPx 引脚极性选择，此位由软件置 1 和清零。 00：WKUP 引脚上升沿唤醒； 01：WKUP 引脚下降沿唤醒； 10：WKUP 引脚上升沿和下降沿均可唤醒； 11：保留

5.4.5 电源状态寄存器（PWR_SR）

偏移地址：0x10

复位值：0x0080 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR_RDY	PVDO.	Res.	Res.	Res.	Res.	Res.	SBF
								R	R						R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUFX				
											R	R	R	R	R

该寄存器可由 POR 复位；

该寄存器在退出待机模式后会被复位；

Puya Confidential

该寄存器不可由系统复位清除；

该寄存器不可由 PWR 软复位清除；

Bit	Name	RW	Reset Value	Function
31:24	Res.	-	-	-
23	MR_RDY	R	1'b1	用于指示 MR 工作状态， 1：MR 模式稳定工作； 0：MR 模式关闭；
22	PVDO	R	1'b0	PVD 输出标志。 此位通过硬件置 1 和清零。仅当通过 PVDE 位使能 PVD 时此位才有效。 0：V_{CC} 高于 PVDT[2:0]位选择的 PVD 阈值。 1：V_{CC} 低于 PVDT[2:0]位选择的 PVD 阈值。 注：PVD 在进入待机模式时停止。因此，待机模式后或复位后，此位等于 0，直到 PVDE 位置 1 然后根据检测电压阈值输出对应值。
21:17	Res.		3'b0	
16	SBF			待机模式标志。 此位由硬件置 1，清零则只能通过 POR/PDR 或将 PWR_SCR 寄存器中的 CSBF 位置 1 来实现。 0：器件未进入待机模式 1：器件已进入待机模式
15:5	Res.		11'b0	
4:0	WUFx	R	5’b0	WKUPx(x=0-4)唤醒标志。 该位通过硬件置 1，清零则只能通过 POR/PDR 或将 PWR_SCR 寄存器中的 CWUFx 位置 1 清零。 0：未发生唤醒事件 1：收到 WKUPx 唤醒事件

5.4.6 电源状态清除寄存器（PWR_SCR）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSBF
															W
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CWUF5	CWUF4	CWUF3	CWUF2	CWUF1
											W	W	W	W	W

该寄存器可由 POR 复位；

该寄存器在退出待机模式后会被复位；

该寄存器可由系统复位清除；

该寄存器可由 pwr 软复位清除；

Bit	Name	RW	Reset Value	Function
31:17	Res.			-
16	CSBF	W	0	SBF 唤醒标志清除信号。 通过设置该位为 1，清除 PWR_SCR 中的 SBF 信号

Puya Confidential

15:5	Res.			
4:0	CWUFx	W	0	WKUPx 唤醒标志清除信号。 通过设置该位为 1，清除 PWR_SCR 中的 WUFx 信号

Puya Confidential

6. 复位与时钟控制（RCC）

6.1 RCC 简介

本模块完成如下功能：

- 产生系统时钟和系统复位
- 产生各个模块时钟和复位
- 产生时钟源控制信号

6.2 RCC 复位

6.2.1 电源复位

只要发生以下事件之一，就会产生电源复位：

- 上电/掉电复位（POR/PDR 复位）
- 在退出待机模式时

电源复位会将所有寄存器复位。

复位入口向量在存储器映射中固定在地址 0x0000_0004。

6.2.2 系统复位

除了时钟控制寄存器 CSR 中的复位标志和备份域中的寄存器外，系统复位会将其他全部寄存器都复位为复位值。

只要发生以下事件之一，就会产生系统复位：

- NRST 引脚低电平（外部复位）
- 窗口看门狗计数结束（WWDG 复位）
- 独立看门狗计数结束（IWDG 复位）
- 软件复位（SYSRESETREQ 复位）
- 低功耗管理复位（NRST_STDBY/NRST_STOP）
- 选项字节加载复位
- 锁相环失锁复位(PLL unlock)

6.2.2.1 NRST 引脚（外部复位）

通过特定的选项位（NRST_MODE），NRST 引脚可配置为：

- 复位输入/输出

复位输入/输出引脚上的任何有效复位信号都会传播到设备内部逻辑，所有系统复位源都会通过脉冲发生器驱动到此引脚。GPIO 功能（PD11）不可用。脉冲发生器保证每个内部复位源在 NRST 引脚上输出的最小复位脉冲持续时间为 20 μs。该模式在每次设备上电复位期间（直到加载选项字节）始终处于有效状态(独立于选项字节设置)。

- 复位输入

Puya Confidential

在此模式下，NRST 引脚上的任何有效复位信号都会传播到设备内部逻辑，设备内部产生的复位。在此配置中，GPIO 功能（PD11）不可用。

- GPIO

在此模式下，引脚可以用作 PD11 标准 GPIO。引脚的复位功能不可用。复位只能从设备内部复位源进行，并且不会传播到此引脚。

6.2.2.2 软件复位

可通过查看 RCC 时钟控制和状态寄存器（RCC_CSR）中的复位标志确定。要对器件进行软件复位，必须将 CPU 中断和复位控制寄存器中的 SYSRESETREQ 位置 1。

6.2.2.3 低功耗管理复位

引发低功耗管理复位的方式有两种：

- 进入待机模式

此复位的使能方式是清零用户选项字节中的 nRST_STDBY 位。且 PWR_CR1.LPMS_SLP 寄存器配置为待机模式，进入低功耗模式时，器件将复位，而非进入待机模式。

- 进入停止模式

此复位的使能方式是清零用户选项字节中的 nRST_STOP 位。且 PWR_CR1.LPMS_SLP 寄存器配置为停止模式，进入低功耗模式时，器件将复位，而非进入停止模式。

6.2.2.4 选项字节加载复位

当 FLASH_CR 寄存器中设置 OBL_LAUNCH 位（位 27）时，会生成选项字节加载复位。此位用于通过软件启动选项字节加载。

6.2.2.5 锁相环失锁复位

当选项字节里的 Pll_unlock_rsten 选择了使能锁相环失锁复位时（Pll_unlock_rsten=1），如果锁相环失锁，则会引起系统复位。

6.2.3 备份域复位

备份域复位会将所有备份域寄存器复位，备份域寄存器包括 RCC_BDCR 备份寄存器。

只要发生以下事件之一，就会产生备份域复位：

- 软件复位：通过将 RCC 备份域控制寄存器（RCC_BDCR）中的 BDRST 位置 1 触发。
- 在电源 V_{CC} 已掉电后，又再上电。

6.2.4 备份域复位之外的复位的统一处理

NRST 引脚复位输入/输出模式下，其他源复位都作用于 NRST 引脚，该引脚在复位过程中时钟保持低电平。

电路如下图所示：

Puya Confidential

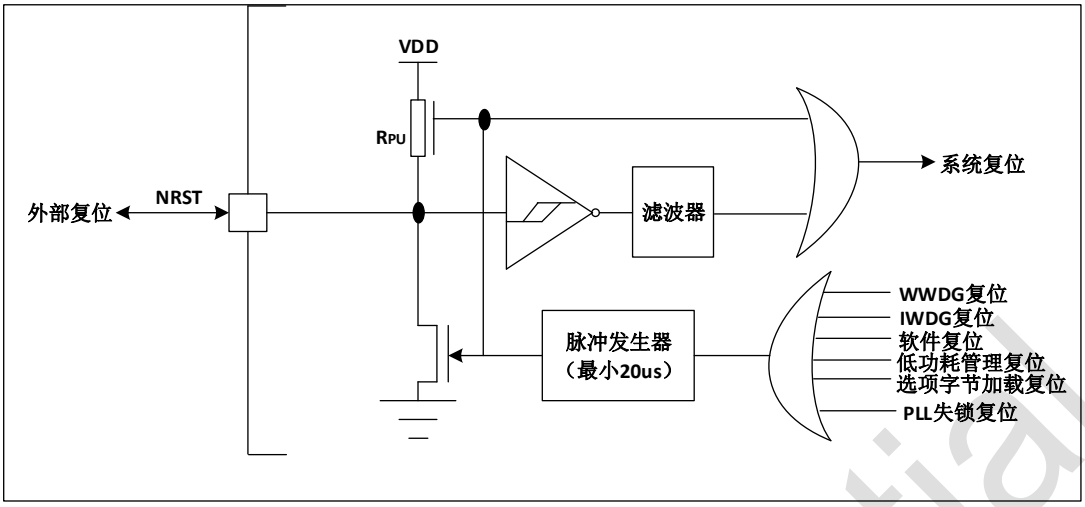


图 6-1 复位电路

当系统发生内部复位时(高电平有效)，脉冲发生器开始产生脉冲信号，并保证该脉冲信号持续至少 20μs。此脉冲信号使 N 管导通。N 管导通会不断拉低 NRST 引脚电压，当拉低到 VIL 时，NRST 引脚产生低电平信号。该信号经过施密特触发器和滤波后，产生系统复位（低电平有效）。

注 1：除外部 PIN 复位源之外的复位源，即脉冲发生器的源，不会经过 30 μs 的滤波后产生系统复位；

注 2：脉冲发生器的复位源，在产生系统复位后，除了本身复位标志置位外，外部引脚复位标志也会置位。

6.2.5 复位标志表

复位标志与复位源之间的对应关系如下表所示：

表 6-1 复位标志表

复位源	PWRRSTF	PINRSTF	SFTRSTF	WWDGRSTF	IWDGRSTF	PLLULRSTF	OBLRSTF	LPWRRSTF
电源复位 (POR/PDR)	√							
待机模式唤醒	√							
引脚复位		√						
软件复位			√					
WWDG 复位				√				
IWDG 复位					√			
PLL 失锁						√		
选项字节修改							√	
低功耗管理复位								√

Puya Confidential

6.3 RCC 时钟

6.3.1 时钟源

系统存在如下时钟源：

6.3.1.1 HSE 时钟

外部高频 OSC.频率范围 4 ~ 32 MHz

HSE 时钟的稳定时间由模拟 HSE 模块计数，模拟 HSE 模块在启动并且稳定后产生 HSE_RDY 信号。

HSE 时钟可以有两个工作模式：

- 外部 XTAL OSC+内部起振电路（HSEBYP=0）
- 通过 OSC_IN IO 输入的外部时钟（HSEBYP=1）

为保证 CLOCK 输出第一个时钟就为稳定时钟， HSE 内部包含一个计数器， 计数完成后输出第一个时钟上升沿。HSEBYP=1 时计数周期是 HSEBYP=0 的一半。

6.3.1.2 HSI 时钟

内部 HSI RC 振荡器。相较于 HSE，功耗低，稳定时间短，但精度低。模拟 HSI 模块在启动并且稳定后产生 HSI_RDY 信号。

上电复位后，在加载阶段，HSI 的校准值加载到 RCC_ICSCR.HSI_TRIMCR 寄存器。

HSI 上电后默认是 8 MHz，可以通过 RCC_ICSCR.HSI_FS 寄存器选择 16 MHz，24 MHz 和 48 MHz。

从停止模式唤醒后，HSI 将作为系统时钟源。

6.3.1.3 PLL 时钟

PLL 模块是用来倍频 HSI 或者 HSE 时钟，PLL 输入时钟频率范围为 8 ~ 48 MHz。输出时钟频率范围为 32 ~ 176 MHz。

PLL lockout 时间典型值为 25 μs，最大值为 50 μs。

6.3.1.4 LSE 时钟

外部 32.768 kHz OSC，用作低功耗时钟。

可以通过配置 LSEDRV 在稳定时间和功耗之间做平衡。LSE 稳定信号由 LSE 模拟模块产生。

与 HSE 来源类似，LSE 也有两个来源：

- 32.768 kHz XTAL+内部起振电路（LSEBYP=0）
- 通过 OSC32_IN 输入的外部时钟（LSEBYP=1）

为保证 CLOCK 输出第一个时钟就为稳定时钟，LSE 内部包含一个计数器，计数完成后输出第一个时钟上升沿。LSEBYP=1 时计数周期是 LSEBYP=0 的一半。

在 LSEBYP=1 情况下，MCO 输出的时钟为原始外灌时钟，且不会受 LSE_ON 信号的控制。但使用 LSE 时钟的逻辑会受 LSE_ON 信号控制。

Puya Confidential

6.3.1.5 LSI 时钟

内部低频 32.768 kHz 时钟，用作 IWDG 时钟，也可用作 PVD 时钟、LPTIM 时钟、COMP1~4 时钟。

6.3.1.6 HSI10M 时钟

该时钟作为低精度时钟，用作 NRST 引脚的滤波计数。每次产生系统复位，经过脉冲发生器后，都通过 NRST 引脚返回，所以每次复位产生时都会使能 HSI10M 时钟。

6.3.2 时钟结构

时钟结构如下图所示：

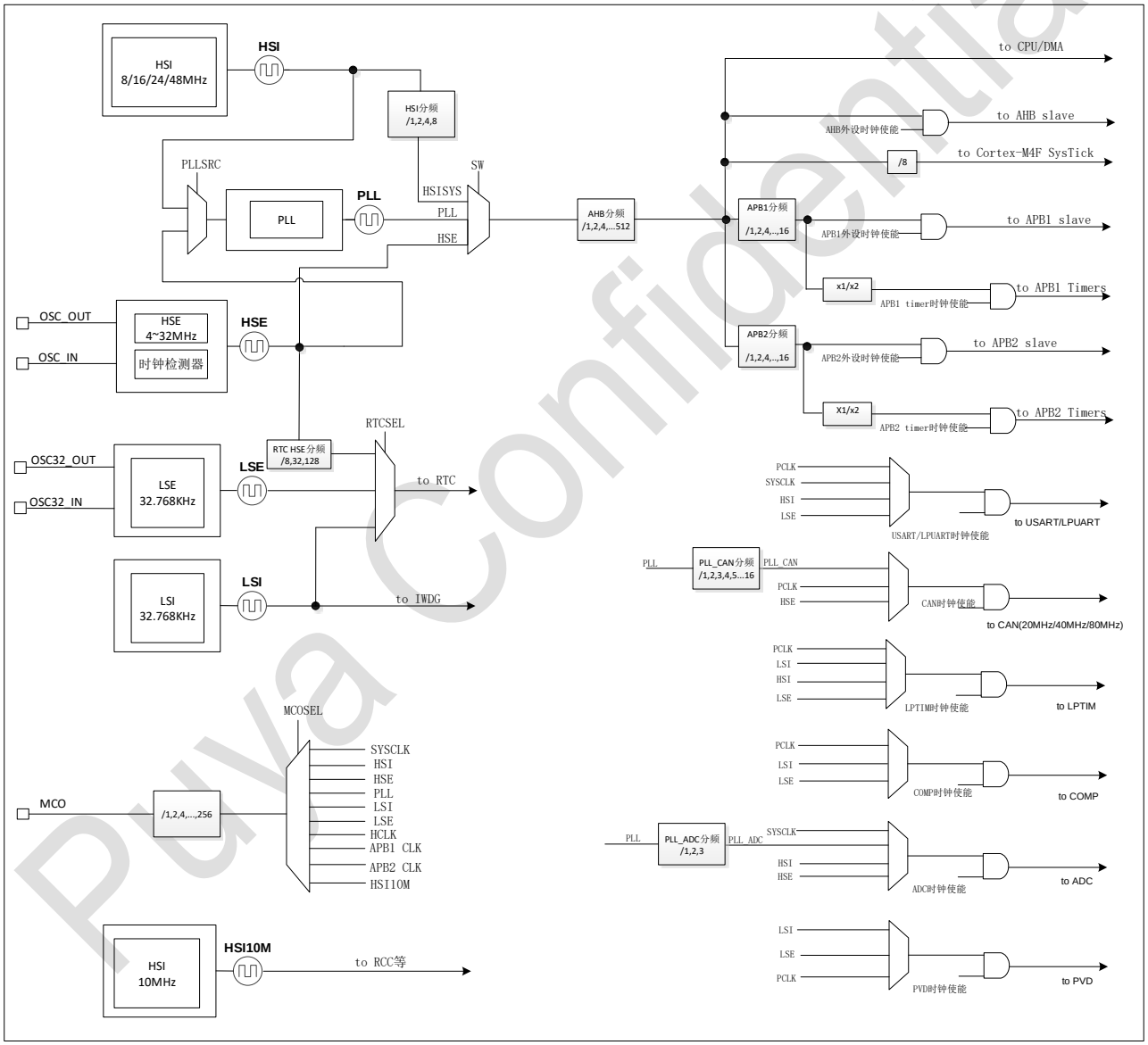


图 6-2 系统时钟结构图

6.3.3 时钟安全系统（CSS）

时钟安全系统可以通过软件激活。在这种情况下，时钟检测器在 HSE 振荡器启动延迟后启用，并在该振荡器停止时禁用。

Puya Confidential

如果在 HSE 时钟上检测到故障，HSE 振荡器将被自动禁用，时钟故障事件将发送到定时器（TIM1/TIM8/TIM2/TIM3）的中断输入，并生成一个中断以通知软件故障（时钟安全系统中断 CSSI），允许 MCU 执行救援操作。CSSI 通过 NMI（不可屏蔽中断）异常向量链接到 Cortex®-M4F。

注：一旦启用 CSS，如果 HSE 时钟出现故障，则会发生 CSS 中断，并自动生成 NMI。除非 CSS 中断挂起位被清除，否则 NMI 将无限期执行。因此，在 NMI ISR 中，用户必须通过在时钟中断清零寄存器（RCC_CICR）中设置 CSSC 位来清除 HSECSSC 中断标志

如果 HSE 振荡器直接或间接用作系统时钟（间接表示：它用作 PLL 输入时钟，PLL 时钟用作系统时钟），则检测到的故障会导致系统时钟切换到 HSI 振荡器，并禁用 HSE 振荡器。如果 HSE 时钟是用作系统时钟的 PLL 的时钟源，则 PLL 也被禁用。

6.3.4 时钟输出

为了方便板级应用，节省 BOM 成本，以及 debug 等的需求，需要芯片提供时钟输出功能。即把下列的 MCO 信号（并分频）通过 GPIO 的复用功能实现时钟输出功能。

- HSI
- HSE
- PLL
- LSE
- LSI
- HSI_10M
- SYSCLK
- HCLK
- PCLK1
- PCLK2

6.3.5 外设时钟使能寄存器

每个外围时钟可以由 RCC_AHBENRx、RCC_APBENRx 寄存器的 xxxxEN 位启用。

当外围时钟未激活时，不支持外围寄存器的读或写访问。

使能位具有同步机制，在使能位被设置之后，在时钟被激活之前有 2 个时钟周期的延迟。

注意：在为外围设备启用时钟后，软件必须等待一段时间后才能访问外围寄存器。

6.4 RCC 寄存器

该模块的寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。RCC_BDCR 除外，只能 32 位访问。

6.4.1 RCC 时钟控制寄存器（RCC_CR）

偏移地址：0x00

复位值：0x0000 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	PLL RDY	PLLON	Res.	Res.	Res.	Res.	HSE CSSON	HSE BYP	HSE RDY	HSEON
						R	RW					RS	RW	R	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	HSIDIV[1:0]		HSIRDY	HSI	HSION	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

						KERON									
			RW	RW	R	RW	RW								

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25	PLLRDY	R	0	PLL 时钟就绪标志。 硬件置位，表明 PLL 时钟已锁定。 0：PLL 未锁定； 1：PLL 已锁定； 该寄存器通过上电复位和系统复位清零。 注：将 PLLON 清零后，PLLRDY 将会立即转换为低电平。
24	PLLON	RW	0	PLL 使能。 由软件置位和清零。 当系统进入停止或待机模式时，硬件会清零该位。当 PLL 时钟被用作（SWS 为 PLL）或将要被用作系统时钟（SW 选择 PLL）时，该位不能被清零。 0：PLL 关闭； 1：PLL 使能； 该寄存器通过上电复位和系统复位清零。
23:20	Reserved	-	-	保留
19	HSECSSON	RS	0	HSE 时钟安全系统使能。 由软件置位，用于使能 HSE 时钟安全系统。当 HSECSSON 置 1 时，时钟监测器将在 HSE 振荡器就绪时由硬件使能，并在检出 HSE 时钟故障时由硬件禁止。 0：时钟安全系统关闭（时钟检测关闭）； 1：时钟安全系统开启（如果 HSE 时钟稳定则时钟检测开启，否则关闭时钟检测）； 该位仅置 1，并通过上电复位和系统复位清零。
18	HSEBYP	RW	0	HSE 旁路振荡器，选择管脚输入时钟。 该位只有当 HSEON=0 时才能写。 管脚输入时钟必须通过 HSEON 位使能才能为器件所用。 0：不旁路 HSE 振荡器，外部高速时钟选择晶振； 1：旁路 HSE 振荡器，外部高速时钟选择外部管脚输入时钟源；
17	HSERDY	R	0	HSE 时钟就绪标志。 该位由硬件置位表明 HSE 振荡器稳定。 0：HSE 振荡器未就绪； 1：HSE 振荡器已就绪； 该寄存器通过上电复位和系统复位清零。 注：在将 HSEON 清零后，HSERDY 将会立即转换为低电平。
16	HSEON	RW	0	HSE 振荡器使能。 当系统进入停止、待机模式时，硬件会清零该位，关闭 HSE 晶振。当 HSE 直接或者间接作为系统时钟源时，该位不能被置 0. 0：HSE 晶振关闭； 1：HSE 晶振使能； 该寄存器通过上电复位和系统复位清零。

Puya Confidential

15:13	Reserved	-	-	保留
12:11	HSIDIV[1:0]	RW	2'h0	HSI 时钟分频系数。 软件控制这些位设定 HSI 的分频系数，产生 HSISYS 时钟。 00：1 01：2 10：4 11：8 该寄存器通过上电复位和系统复位清零。
10	HSIRDY	R	0	HSI 时钟就绪标志。 硬件置位表明 HSI 振荡器稳定。该位只有当 HSION=1 时才有效。 0：HSI 振荡器未就绪； 1：HSI 振荡器已就绪； 该寄存器通过上电复位和系统复位置位。 注: 在将 HSION 清零后，HSIRDY 将在 3 个 HSI 时钟之后拉低。
9	HSIKERON	RW	0	一些 IP 内核的高速内部时钟使能位。 该位由软件置 1 和复位，即使在停止模式下也可强制 HSI 振荡器开启，从而可快速用作一些可唤醒 IP（如 USART 或 I2C1/I2C2）的内核时钟。该位对 HSION 的值没有任何影响。 0：对 HSI 振荡器没有影响。 1：HSI 振荡器即使在停止模式下也被强制使能。 该寄存器通过上电复位和系统复位清零。
8	HSION	RW	1	HSI 时钟使能。 硬件在进入停止、待机模式时，会根据需要清零该寄存器停止 HSI 时钟。 当 HSI 直接或者间接作为系统时钟时，该寄存器不能为 0. 0：HSI 振荡器关闭； 1：HSI 振荡器使能； 硬件在如下情况会使能 HSI 振荡器： 1） 硬件从停止或者待机模式唤醒后； 2） HSE 直接/间接作为系统时钟但出现 HSE CSS fail； 该寄存器通过上电复位和系统复位置位。
7:0	Reserved	-	-	保留

6.4.2 RCC 内部时钟源校准寄存器（RCC_ICSCR）

偏移地址：0x08

复位值：0x0000 8200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSI_FS[1:0]	
														RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HSI_TRIMCR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:18	Reserved	-	-	保留
17:16	HSI_FS[1:0]	RW	2'b00	HSI 频率选择： 00：8 MHz

Puya Confidential

				01：16 MHz 10：24 MHz 11：48 MHz 上电后，默认选择 8 MHz，在选项字节加载完成后，硬件通过加载 Flash trimming 页数据切换成 8 MHz。在系统复位后，硬件同样切换成 8 MHz。
15:0	HSI_TRIMCR[15:0]	RW	16'h8200	HSI 时钟频率校准值。 上电后硬件用 HSI 8MHz 的默认校准值，在上电加载 Trimming 时会把 HSI 8MHz 的校准值写入该寄存器中。 当存在多个 HSI 频率时，在配置 HSI_FS 寄存器修改频率时，软件同时需要通过读出存放在信息区相应地址的数据，写入该寄存器，实现 HSI 特定输出频率下的校准。 8 MHz 校准值存放地址：0x1FFF 1A80 16 MHz 校准值存放地址：0x1FFF 1A84 24 MHz 校准值存放地址：0x1FFF 1A8C 48 MHz 校准值存放地址：0x1FFF 1A90 在系统复位后，硬件会将该寄存器切换成对应 HSI8 MHz 的 Trimming 值（不是默认值）。

注意：

1. HSI_FS、HSI_TRIMCR 需要同时配置。
2. 连续两次写该寄存器的时间间隔不能少于 6 个 HSI 时钟周期。

6.4.3 RCC 时钟配置寄存器 1（RCC_CFGR1）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCO2PRE[3:0]				MCO2SEL[2:0]				MCO1PRE[3:0]				MCO1SEL[2:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	PPRE1[2:0]			HPRE[3:0]				Res.	Res.	SWS[1:0]		Res.	Res.	SW[1:0]	
	RW	RW	RW	RW	RW	RW	RW			RW	RW			RW	RW

Bit	Name	RW	Reset Value	Function
31:28	MCO2PRE[3:0]	RW	0	MCO2 分频系数。软件控制这些位，设置 MCO 输出的分频系数： 0xxx：1 1000：2 1001：4 1010：8 1011：16 1100：32 1101：64 1110：128 1111：256 推荐在 MCO2 输出使能前，设置这些位。 该寄存器通过上电复位和系统复位清零。
27:24	MCO2SEL[3:0]	RW	0	MCO2 输出时钟选择。

Puya Confidential

				0000：无输出 0001：SYSCLK 0010：HSI 0011：HSE 0100：PLL 0101：LSI 0110：LSE 0111：无输出 1000：无输出 1001：HSI10M 1010：HCLK 1011：PCLK1 1100：PCLK2 1101：无输出 1110：无输出 1111：无输出 注 1：如果不特殊控制，输出均为稳定时钟； 注 2：若 HSE 配置 HSE_RDYSEL 和寄存器，则可以输出 startup 期间的时钟； 注 3：MCO2 输出选择系统时钟时，需要保证输出时频率不超过 IO 的最大允许频率。 该寄存器通过上电复位和系统复位清零
23:20	MCO1PRE[3:0]	RW	0	MCO1 分频系数。软件控制这些位，设置 MCO2 输出的分频系数： 0xxx：1 1000：2 1001：4 1010：8 1011：16 1100：32 1101：64 1110：128 1111：256 推荐在 MCO1 输出使能前，设置这些位。 该寄存器通过上电复位和系统复位清零。
19:16	MCO1SEL[3:0]	RW	0	MCO1 输出时钟选择。 0000：无输出 0001：SYSCLK 0010：HSI 0011：HSE 0100：PLL 0101：LSI 0110：LSE 0111：无输出 1000：无输出

Puya Confidential

				1001：HSI10M 1010：HCLK 1011：PCLK1 1100：PCLK2 1101：无输出 1110：无输出 1111：无输出 注 1：如果不特殊控制，输出均为稳定时钟； 注 2：若 HSE 配置 HSE_RDYSEL 和寄存器，则可以输出 startup 期间的时钟； 注 3：MCO1 输出选择系统时钟时，需要保证输出时频率不超过 IO 的最大允许频率。 该寄存器通过上电复位和系统复位清零。
15	Reserved	-	-	保留
14:12	PPRE1[2:0]	RW	3’b0	APB1 时钟 PCLK1 基于 HCLK 的分频系数。 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频 该寄存器通过上电复位和系统复位清零。
11:8	HPRE[3:0]	RW	4’b0	AHB 时钟 HCLK 基于 SYSCLK 的分频系数。 0xxx：不分频 1000：2 分频 1001：4 分频 1010：8 分频 1011：16 分频 1100：64 分频 1101：128 分频 1110：256 分频 1111：512 分频 为了保证系统正常工作，需要根据 VR 电源情况配置合适频率。 注：建议逐级切换分频系数。
7:6	Reserved	-	-	保留
5:4	SWS[1:0]	R	0	系统时钟源选择状态位。 该位由硬件控制，表明系统时钟源的选择。 00：HSISYS 01：HSE 10：PLL 11：保留 该寄存器通过上电复位和系统复位清零。
3:2	Reserved	-	-	保留
1:0	SW[1:0]	RW	0	系统时钟源选择。 该位由硬件和软件控制，表明系统时钟源的选择。

Puya Confidential

				00：HSISYS 01：HSE 10：PLL 11：保留 硬件配置为 HSISYS 的情况包括： 1）系统从停止、待机模式退出 2）软件配置 001（HSE），出现 HSE 故障 3）软件配置 010（PLL），且 HSE 作为 PLL 参考时钟输入，出现 HSE 故障 该寄存器通过上电复位和系统复位清零（选择 HSISYS）

6.4.4 RCC 时钟配置寄存器 2（RCC_CFGR2）

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	PPRE2[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
					RW	RW	RW								

Bit	Name	RW	Reset Value	Function
31:11	Reserved	-	-	保留
10:8	PPRE2[2:0]	RW	3’h0	APB2 时钟 PCLK2 基于 HCLK 的分频系数。 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频 该寄存器通过上电复位和系统复位清零
7:0	Reserved	-	-	保留

6.4.5 RCC PLL 时钟配置寄存器 1（RCC_PLLCFGR1）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLLMUL[3:0]				Res.	PLLSRC
										RW	RW	RW	RW		RW

Bit	Name	RW	Reset Value	Function
31:6	Reserved	-	-	保留
5:2	PLLMUL[3:0]	RW	4’h0	PLL 倍频系数 0000：x2 0001：x3 0010：x4

Shanghai Puya Confidential

				… 1110：x16 1111：保留(不可设置为该值,否则无法保证 PLL 正常)
1	Reserved	-	-	保留
0	PLLSRC	RW	0	PLL 时钟源选择： 0：HSI 1：HSE

6.4.6 RCC 外部时钟源配置寄存器（RCC_ECSCR）

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSE_FILT_ENB	HSE_RDYSEL [1:0]		HSE_DRIVER [1:0]	
											RW	RW	RW	RW	RW

该寄存器需要在 HSE 使能（HSEON=1）前配置，如果需要改变配置，必须先关闭 HSE，重新配置后再使能 HSE。

Bit	Name	RW	Reset Value	Function
31:5	Reserved	-	-	保留
4	HSE_FILT_ENB	RW	0	模拟 HSE 滤波控制。 0：使能滤波 1：禁止滤波 该寄存器通过上电复位和系统复位清零。
3:2	HSE_RDYSEL [1:0]	RW	2’h0	HSE 稳定时间选择。 HSEBYP=0: 00：4096 个 HSE 时钟； 01：2048 个 HSE 时钟； 10：8192 个 HSE 时钟； 11：不计稳定时间，直接输出； HSEBYP=1: 00：2048 个 HSE 时钟； 01：1024 个 HSE 时钟； 10：4096 个 HSE 时钟； 11：不计稳定时间，直接输出； 该寄存器通过上电复位和系统复位清零。
1:0	HSE_DRV[1:0]	RW	2’h0	HSE 驱动能力配置。 00: gm 3.9 mA/V 01: gm 7.2 mA/V 10: gm 9.7 mA/V 11: gm 11.2 mA/V 该寄存器通过上电复位和系统复位清零。

Puya Confidential

6.4.7 RCC 时钟中断使能寄存器（RCC_CIER）

偏移地址：0x3C

复位值：0x0000 0000

该寄存器通过上电复位和系统复位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE
											RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:5	Reserved	-	-	保留
4	PLLRDYIE	RW	0	PLL 时钟就绪中断使能。 0：禁止； 1：使能；
3	HSERDYIE	RW	0	HSE 时钟就绪中断使能。 0：禁止； 1：使能；
2	HSIRDYIE	RW	0	HSI 时钟就绪中断使能。 0：禁止； 1：使能；
1	LSERDYIE	RW	0	LSE 时钟就绪中断使能。 0：禁止； 1：使能；
0	LSIRDYIE	RW	0	LSI 时钟就绪中断使能。 0：禁止； 1：使能；

6.4.8 RCC 时钟中断标志寄存器（RCC_CIFR）

偏移地址：0x40

复位值：0x0000 0000

该寄存器通过上电复位和系统复位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSE CSSF	Res.	Res.	Res.	PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF
							R				R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8	HSECSSF	R	0	HSE 时钟安全系统中断标识位。 当硬件检测 HSE OSC 时钟失败时置位该寄存器。

Puya Confidential

				0：HSE 时钟检测失败中断未产生； 1：HSE 时钟检测失败中断产生； 写 HSECSSC 寄存器为 1 清零该位。
7:5	Reserved	-	-	保留
4	PLLRDYF	R	0	PLL 就绪中断标志。 当 PLL 锁定并且 PLLRDYDIE=1 时，硬件置位该寄存器。 0：PLL 锁定中断未产生； 1：PLL 锁定中断产生； 写 PLLRDYC 寄存器为 1 清零该位。
3	HSERDYF	R	0	HSE 时钟就绪中断标志。 当 HSE 时钟稳定并且 HSERDYIE=1 时，硬件置位该寄存器。 0：HSE 时钟就绪中断未产生； 1：HSE 时钟就绪中断产生； 写 HSERDYC 寄存器为 1 清零该位。
2	HSIRDYF	R	0	HSI 时钟就绪中断标志。 HSI 时钟稳定并且 HSIRDYIE=1 时，硬件置位该寄存器。 0：HSI 时钟就绪中断未产生； 1：HSI 时钟就绪中断产生； 写 HSIRDYC 寄存器为 1 清零该位。
1	LSERDYF	R	0	LSERDY 时钟就绪中断标志。 LSE 时钟稳定并且 LSERDYDIE=1 时，硬件置位该寄存器。 0：LSERDY 时钟就绪中断未产生； 1：LSERDY 时钟就绪中断产生； 写 LSERDYC 寄存器为 1 清零该位。
0	LSIRDYF	R	0	LSIRDY 时钟就绪中断标志。 LSI 时钟稳定并且 LSIRDYIE=1 时，硬件置位该寄存器。 0：LSIRDY 时钟就绪中断未产生； 1：LSIRDY 时钟就绪中断产生； 写 LSIRDYC 寄存器为 1 清零该位。

6.4.9 RCC 时钟中断清零寄存器（RCC_CICR）

偏移地址：0x44

复位值：0x0000 0000

该寄存器通过上电复位和系统复位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSE CSSC	Res.	Res.	Res.	PLL RDYC	HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC
							W				W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8	HSECSSC	W	0	HSE 时钟安全系统（CSS）中断标志清零。 0：没影响； 1：清零 HSECSSF 标志；
7:5	Reserved	-	-	保留
4	PLLRDYC	W	0	PLL 就绪中断标志清零。 0：无影响； 1：清零 PLLRDYF 标志；
3	HSERDYC	W	0	HSE 就绪中断标志清零。 0：无影响； 1：清零 HSERDYF 标志；
2	HSIRDYC	W	0	HSI 就绪中断标志清零。 0：无影响； 1：清零 HSIRDYF 标志；
1	LSERDYC	W	0	LSE 就绪中断标志清零。 0：无影响； 1：清零 LSERDYF 标志；
0	LSIRDYC	W	0	LSI 就绪中断标志清零。 0：无影响； 1：清零 LSIRDYF 标志；

6.4.10 RCC AHB 外设复位寄存器 1（RCC_AHBRSTR1）

偏移地址：0x48

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC RST
															RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1 RST
															RW

Bit	Name	RW	Reset Value	Function
31:17	Reserved	-	-	保留
16	CRCRST	RW	0	CRC 模块复位。 0：无影响； 1：该模块复位；
15:1	Reserved	-	-	保留
0	DMARST	RW	0	DMA 模块复位。 0：无影响； 1：该模块复位；

Puya Confidential

6.4.11 RCC AHB 外设复位寄存器 2（RCC_AHBRSTR2）

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
												RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:4	Reserved	-	-	保留
3	GPIODRST	RW	0	GPIOD 模块复位。 0：无影响； 1：该模块复位；
2	GPIOCRST	RW	0	GPIOC 模块复位。 0：无影响； 1：该模块复位；
1	GPIOBRST	RW	0	GPIOB 模块复位。 0：无影响； 1：该模块复位；
0	GPIOARST	RW	0	GPIOA 模块复位。 0：无影响； 1：该模块复位；

6.4.12 RCC APB 外设复位寄存器 1（RCC_APBRSTR1）

偏移地址：0x58

复位值：0x0000 0000

该寄存器由软件置位和清零，软件置位后，该模块维持复位直到软件将复位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8RST	TIM7RST	TIM6RST	Res.	Res.	TIM3RST	TIM2RST	TIM1RST	Res.
							RW	RW	RW			RW	RW	RW	

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8	TIM8RST	RW	0	TIM8 模块复位。 0：无影响； 1：该模块复位；
7	TIM7RST	RW	0	TIM7 模块复位。 0：无影响； 1：该模块复位；
6	TIM6RST	RW	0	TIM6 模块复位。 0：无影响； 1：该模块复位；
5:4	Reserved	-	-	保留

Puya Confidential

3	TIM3RST	RW	0	TIM3 模块复位。 0：无影响； 1：该模块复位；
2	TIM2RST	RW	0	TIM2 模块复位。 0：无影响； 1：该模块复位；
1	TIM1RST	RW	0	TIM1 模块复位。 0：无影响； 1：该模块复位；
0	Reserved	-	-	保留

6.4.13 RCC APB 外设复位寄存器 2（RCC_APBRSTR2）

偏移地址：0x5C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM RST
															RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPWM RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							RW								

Bit	Name	RW	Reset Value	Function
31:17	Reserved	-	-	保留
16	LPTIMRST	RW	0	LPTIM 模块时钟使能 0：无影响； 1：该模块复位
15:9	Reserved	-	-	保留
8	EPWMRST	RW	0	EPWM 模块复位 0：无影响； 1：该模块复位
7:0	Reserved	-	-	保留

6.4.14 RCC APB 外设复位寄存器 3（RCC_APBRSTR3）

偏移地址：0x60

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	CAN RST	Res.	Res.	Res.	Res.	SPI2 RST	SPI1 RST	Res.	Res.	Res.	Res.	I2C2 RST	I2C1 RST
			RW					RW	RW					RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	LPUART RST	Res.	Res.	Res.	Res.	Res.	USARTRST	Res.	Res.	Res.	Res.	UART2RST	UART1RST
			RW						RW					RW	RW

Bit	Name	RW	Reset Value	Function
31:29	Reserved	-	-	保留
28	CANRST	RW	0	CAN 模块复位

Puya Confidential

				0：无影响； 1：该模块复位
27:23	Reserved	-	-	保留
23	SPI2RST	RW	0	SPI2 模块复位 0：无影响； 1：该模块复位
22	SPI1RST	RW	0	SPI1 模块复位 0：无影响； 1：该模块复位
21:18	Reserved	-	-	保留
17	I2C2RST	RW	0	I2C2 模块复位 0：无影响； 1：该模块复位
16	I2C1RST	RW	0	I2C1 模块复位 0：无影响； 1：该模块复位
15:13	Reserved	-	-	保留
12	LPUARTRST	RW	0	LPUART 模块复位 0：无影响； 1：该模块复位
11:7	Reserved	-	-	保留
6	USARTRST	RW	0	USART 模块复位 0：无影响； 1：该模块复位
5:2	Reserved	-	-	保留
1	UART2RST	RW	0	UART2 模块复位 0：无影响； 1：该模块复位
0	UART1RST	RW	0	UART1 模块复位 0：无影响； 1：该模块复位

6.4.15 RCC APB 外设复位寄存器 4（RCC_APBRSTR4）

偏移地址：0x64

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPRST	Res.	Res.	Res.	OPA RST	Res.	Res.	Res.	DAC RST
							RW				RW				RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	ADC2RST	ADC1RST	Res.	PWR RST	VREF BUF RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGRST
			RW	RW		RW	RW								RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24	COMPRST	RW	0	COMP 模块复位 0：无影响； 1：该模块复位
23:21	Reserved	-	-	保留
20	OPARST	RW	0	OPA 模块复位 0：无影响； 1：该模块复位
19:17	Reserved	-	-	保留
16	DACRST	RW	0	DAC 模块复位 0：无影响； 1：该模块复位
15:13	Reserved	-	-	保留
12	ADC2RST	RW	0	ADC2 模块复位 0：无影响； 1：该模块复位
11	ADC1RST	RW	0	ADC1 模块复位 0：无影响； 1：该模块复位
10	Reserved	-	-	保留
9	PWRRST	RW	0	PWR 模块复位 0：无影响； 1：该模块复位
8	VREFBUFRST	RW	0	VREFBUF 模块复位 0：无影响； 1：该模块复位
7:1	Reserved	-	-	保留
0	SYSCFGRST	RW	0	SYSCFG 模块复位 0：无影响； 1：该模块复位

6.4.16 RCC AHB 外设时钟使能寄存器 1（RCC_AHBENR1）

偏移地址：0x70

复位值：0x0000 0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC EN
															RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM SMEN	Res.	Res.	Res.	Res.	DMA EN
										RW					RW

Bit	Name	RW	Reset Value	Function
31:17	Reserved	-	-	保留
16	CRCEN	RW	0	CRC 模块时钟使能。

Puya Confidential

				0：禁止； 1：使能；
15:6	Reserved	-	-	保留
5	SRAMSMEN	RW	1	在 Sleep 模式下，SRAM 的时钟使能控制 0：在 Sleep 模式该模块时钟关闭 1：在 Sleep 模式该模块时钟使能 注：该位仅影响 Sleep 模式该模块的时钟使能，在 Run 模式，该模块时钟不会关闭
4	Reserved	-	1	保留 注：该位初始值为 1，仅可写初始值
3:1	Reserved	-	-	保留
0	DMAEN	RW	0	DM1 模块时钟使能。 0：禁止； 1：使能；

6.4.17 RCC AHB 外设时钟使能寄存器 2（RCC_AHBENR2）

偏移地址：0x74

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
												RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:4	Reserved	-	-	保留
3	GPIODEN	RW	0	GPIOD 模块时钟使能。 0：禁止； 1：使能
2	GPIOCEN	RW	0	GPIOC 模块时钟使能。 0：禁止； 1：使能
1	GPIOBEN	RW	0	GPIOB 模块时钟使能。 0：禁止； 1：使能
0	GPIOAEN	RW	0	GPIOA 模块时钟使能。 0：禁止； 1：使能

6.4.18 RCC APB 外设时钟使能寄存器 1（RCC_APBENR1）

偏移地址：0x80

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8 EN	TIM7 EN	TIM6 EN	Res.	Res.	TIM3 EN	TIM2 EN	TIM1 EN	Res.
							RW	RW	RW			RW	RW	RW	

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8	TIM8EN	RW	0	TIM8 模块时钟使能。 0：禁止； 1：使能
7	TIM7EN	RW	0	TIM7 模块时钟使能。 0：禁止； 1：使能
6	TIM6EN	RW	0	TIM6 模块时钟使能。 0：禁止； 1：使能
5:4	Reserved	-	-	保留
3	TIM3EN	RW	0	TIM3 模块时钟使能。 0：禁止； 1：使能
2	TIM2EN	RW	0	TIM2 模块时钟使能。 0：禁止； 1：使能
1	TIM1EN	RW	0	TIM1 模块时钟使能。 0：禁止； 1：使能
0	Reserved	-	-	保留

6.4.19 RCC APB 外设时钟使能寄存器 2（RCC_APBENR2）

偏移地址：0x84

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WWDGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIMEN
RS															RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPWMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							RW								

Bit	Name	RW	Reset Value	Function
31	WWDGEN	RS	0	WWDG 模块时钟使能 0：禁止 1：使能 该寄存器由系统复位清零，软件写 0 无效
30:17	Reserved	-	-	保留
16	LPTIMEN	RW	0	LPTIM 模块时钟使能 0：禁止 1：使能
15:9	Reserved	-	-	保留
8	EPWMEN	RW	0	EPWM 模块时钟使能 0：禁止

Puya Confidential

				1：使能
7:0	Reserved	-	-	保留

6.4.20 RCC APB 外设时钟使能寄存器 3（RCC_APBENR3）

偏移地址：0x88

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	CAN EN	Res.	Res.	Res.	Res.	SPI2EN	SPI1EN	Res.	Res.	Res.	Res.	I2C2EN	I2C1EN
			RW					RW	RW					RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**11**	**10**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	LPUART EN	Res.	Res.	Res.	Res.	Res.	USART EN	Res.	Res.	Res.	Res.	UART2 EN	UART1 EN
			RW						RW					RW	RW

Bit	Name	RW	Reset Value	Function
31:29	Reserved	-	-	保留
28	CANEN	RW	0	CAN 模块时钟使能 0：禁止 1：使能
27:24	Reserved	-	-	保留
23	SPI2EN	RW	0	SPI2 模块时钟使能 0：禁止 1：使能
22	SPI1EN	RW	0	SPI1 模块时钟使能 0：禁止 1：使能
21:18	Reserved	-	-	保留
17	I2C2EN	RW	0	I2C2 模块时钟使能 0：禁止 1：使能
16	I2C1EN	RW	0	I2C1 模块时钟使能 0：禁止 1：使能
15:13	Reserved	-	-	保留
12	LPUARTEN	RW	0	LPUART 模块时钟使能 0：禁止 1：使能
11:7	Reserved	-	-	保留
6	USARTEN	RW	0	USART 模块时钟使能 0：禁止 1：使能
5:2	Reserved	-	-	保留
1	UART2EN	RW	0	UART2 模块时钟使能 0：禁止 1：使能
0	UART1EN	RW	0	UART1 模块时钟使能 0：禁止

Puya Confidential

				1：使能

6.4.21 RCC APB 外设时钟使能寄存器 4（RCC_APBENR4）

偏移地址：0x8C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPEN	Res.	Res.	Res.	OPAEN	Res.	Res.	Res.	DACEN
							RW				RW				RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	ADC2EN	ADC1EN	Res.	PWREN	VREFBUFEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
			RW	RW		RW	RW								RW

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24	COMPEN	RW	0	COMP 模块时钟使能 0：禁止 1：使能
23:21	Reserved	-	-	保留
20	OPAEN	RW	0	OPA 模块时钟使能 0：禁止 1：使能
19:17	Reserved	-	-	保留
16	DACEN	RW	0	DAC 模块时钟使能 0：禁止 1：使能
15:13	Reserved	-	-	保留
12	ADC2EN	RW	0	ADC2 模块时钟使能 0：禁止 1：使能
11	ADC1EN	RW	0	ADC1 模块时钟使能 0：禁止 1：使能
10	Reserved	-	-	保留
9	PWREN	RW	0	PWR 模块时钟使能 0：禁止 1：使能
8	VREFBUFEN	RW	0	V_{REFBUF} 模块时钟使能 0：禁止 1：使能
7:1	Reserved	-	-	保留
0	SYSCFGEN	RW	0	SYSCFG 模块时钟使能 0：禁止 1：使能

Puya Confidential

6.4.22 RCC 外设独立时钟配置寄存器 1（RCC_CCIPR1）

偏移地址：0x98

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PVDSEL [1:0]		TIMCLK CTRL	COMP4SEL [1:0]		COMP3SEL [1:0]		COMP2SEL [1:0]		COMP1SEL [1:0]		Res.	Res.	Res.	Res.
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	LPUART SEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USARTSEL [1:0]	
		RW	RW											RW	RW

Bit	Name	RW	Reset Value	Function
31	Reserved	-	-	保留
30:29	PVDSEL[1:0]	RW	2'h0	PVD 滤波时钟选择。 00：PVD 滤波时钟选择 LSI； 01：PVD 滤波时钟选择 LSE; 1x：PVD 滤波时钟选择 PWR PCLK； 注：在配置 PWR_CR.PVDE=1 使能 PVD 功能之前需要先配置该寄存器选择 PVD 滤波时钟，并在整个过程中时钟选择不会变化。
28	TIMCLKCTRL	RW	0	TIMER PCLK 频率控制。 0：TIMER PCLK 为系统 PCLK*2，但频率不会超过 HCLK； 1：TIMER PCLK 为系统 PCLK*1；
27:26	COMP4SEL[1:0]	RW	2'h0	COMP4 时钟源选择。 00：PCLK； 01：LSI； 1x：LSE；
25:24	COMP3SEL[1:0]	RW	2'h0	COMP3 时钟源选择。 00：PCLK； 01：LSI； 1x：LSE；
23:22	COMP2SEL[1:0]	RW	2'h0	COMP2 时钟源选择。 00：PCLK； 01：LSI； 1x：LSE；
21:20	COMP1SEL[1:0]	RW	2'h0	COMP1 时钟源选择。 00：PCLK； 01：LSI； 1x：LSE；
19:14	Reserved	-	-	保留
13:12	LPUARTSEL[1:0]	RW	2'h0	LPUART 时钟源选择。 00：PCLK； 01：SYSCLK； 10：HSI； 11：LSE;
11:2	Reserved	-	-	保留

Puya Confidential

1:0	USARTSEL[1:0]	RW	2'h0	USART 时钟源选择。 00：PCLK； 01：SYSCLK； 10：HSI； 11：LSE;

6.4.23 RCC 外设独立时钟配置寄存器 2（RCC_CCIPR2）

偏移地址：0x9C

复位值：0x0040 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPWM_ TBCLKSYNC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC_HSEDIV _SEL[1:0]		Res.	Res.	Res.	Res.	LPTIMSEL [1:0]	
RW								RW	RW					RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ADC2KEN	Res.	Res.	Res.	ADC1KEN	ADC12SEL [1:0]		Res.	Res.	Res.	Res.	Res.	Res.	CAN CKSEL[1:0]	
	RW				RW	RW	RW							RW	RW

Bit	Name	RW	Reset Value	Function
31	EPWM_ TBCLKSYNC	RW	0	使能 EPWM 时基时钟同步 设置后将所有使能的 PWM 模块同步到基准时钟 (TBCLK) 0：禁止 1：使能
30:24	Reserved	-	-	保留
23:22	RTC_HSEDIV_SEL[1:0]	RW	2'b01	RTC 时钟源选择为 HSE 时钟的分频 00：32 分频 01：128 分频 1x：8 分频
21:18	Reserved	-	-	保留
17:16	LPTIMSEL[1:0]	RW	2'b0	LPTIM 时钟源选择。 00：PCLK； 01：LSI； 10：HSI； 11：LSE；
15	Reserved	-	-	保留
14	ADC2KEN	RW	0	ADC2 内核时钟使能 0：禁止 1：使能
13:11	Reserved	-	-	保留
10	ADC1KEN	RW	0	ADC1 内核时钟使能 0：禁止 1：使能
9:8	ADC12SEL[1:0]	RW	2'b0	ADC1、ADC2 时钟源选择。 00：SYSCLK； 01：PLL_ADC； 10：HSI； 11：HSE；
7:2	Reserved	-	-	保留

Puya Confidential

1:0	CANCKSEL[1:0]	RW	2’b0	CAN 模块时钟选择。 00：PLL_CAN 01：APB1 CLK 1x：HSE

6.4.24 RCC 外设独立时钟配置寄存器 3（RCC_CCIPR3）

偏移地址：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PLL_CAN_DIV[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				RW	RW	RW	RW								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL_ADC_DIV[1:0]	
														RW	RW

Bit	Name	RW	Reset Value	Function
31:28	Reserved	-	-	保留
27:24	PLL_CAN_DIV[3:0]	RW	4’h0	由 PLL 分频产生 PLL_CAN 的分频系数 0x0：1 0x1：2 0x2：3 … 0xF：16
23:2	Reserved	-	-	保留
1:0	PLL_ADC_DIV[1:0]	RW	2’b0	由 PLL 分频产生 PLL_ADC 的分频系数 00：1 01：2 1x：3

6.4.25 RCC 备份域控制寄存器（RCC_BDCR）

偏移地址：0xB0

复位值：0x0200 0000

当 PWR_CR1.DBP 为 1 时，才允许写该寄存器。

该寄存器在 VBKP 域实现。

BDRST 可被软件清零，也可被系统复位清零。

LSE_RDYSEL，LSEDRV，LSI_TRIMCR 仅能通过 BPOR 复位。

其他位由备份域上电复位（BPOR）和备份域软复位（BDRST）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	LSE_RDYSEL[1:0]		LSI RDY	LSI ON	LSI_TRIMCR[8:0]									BD RST
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	Res.	Res.	LSEDRV[1:0]		LSE BYP	LSE RDY	LSE ON
RW						RW	RW				RW	RW	RW	R	RW

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29:28	LSE_RDYSEL[1:0]	RW	2’b0	LSE 晶振稳定时间选择。

Puya Confidential

				LSEBYP=0： 00：4096 个 LSE 时钟周期； 01：2048 个 LSE 时钟周期； 10：8192 个 LSE 时钟周期； 11：不计稳定时间，直接输出； LSEBYP=1： 00：2048 个 LSE 时钟周期； 01：1024 个 LSE 时钟周期； 10：4096 个 LSE 时钟周期； 11：不计稳定时间，直接输出；
27	LSIRDY	R	0	LSI OSC 稳定标志。 0：LSI 未稳定 1：LSI 已稳定
26	LSION	RW	0	LSI OSC 使能。 0：禁止 1：使能 软件置位，软件清零。在硬件使能 IWDG（通过选项字节）时，硬件会置位该位。
25:17	LSI_TRIMCR[8:0]	RW	9’h100	内部低速时钟频率校准。 上电后芯片硬件会把出厂信息（存放在 0x1FFF 1E24）写入该寄存器中，使 LSI 可以输出精准的 32.768 kHz 频率。 软件通过对该寄存器值进行改写，每增（减）1，使 LSI 的输出频率增（减）约 0.2%。 该寄存器复位仅受备份域上电复位影响。
16	BDRST	RW	0	备份域软复位。 0：无影响； 1：复位；
15	RTCEN	RW	0	RTC、TAMPER、BKPREG 时钟使能。 0：禁止； 1：使能；
14:10	Reserved	-	-	保留
9:8	RTCSEL[1:0]	RW	2’b0	RTC 时钟源选择。 00：No clock 01：LSE 10：LSI 11：HSE 分频，8 分频/32 分频/128 分频（默认） 一旦 RTC 时钟源选择后不能再改变，除非备份域复位被清零。
7:5	Reserved	-	-	保留
4:3	LSEDRV[1:0]	RW	2’b0	LSE 驱动能力设置 00：gm 2.5 μA/V 01：gm 3.75 μA/V 10：gm 8.5 μA/V 11：gm 13.5 μA/V

Puya Confidential

2	LSEBYP	RW	0	LSE OSC bypass 0：不旁路，低速外部时钟选择晶振； 1：旁路，低速外部时钟选择外部接口输入时钟； 注：只有当外部 32.768KHz OSC 禁止（LSEON=0 并且 LSERDY=0）时才能写该位。
1	LSERDY	R	0	LSE OSC 就绪 硬件配置该位为 1 表明 LSE 时钟就绪。 在 LSEON 清零后，该位需要 6 个 LSE 时钟后再清零。
0	LSEON	RW	0	LSE OSC 使能。 0：禁止； 1：使能；

6.4.26 RCC 控制/状态寄存器（RCC_CSR）

偏移地址：0xB8

复位值：0xXX0X 0010

该寄存器中复位标志位只能由电源复位清零，其他由系统复位

引脚复位无法清零 NRST_FLTDIS，但其他的复位可以清零

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PWRRSTF	PINRSTF	OBLRSTF	Res.	Res.	Res.	Res.	Res.	PLLULRSTF	Res.	Res.	Res.
R	R	R	R	R	R	R	R					R			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RMVF	Res.	Res.	EPWM_PRST_RELEASE	EPWM_SFTRST_RELEASE	Res.	Res.	Res.	NRST_FLTDIS
							RW			R	R				R

Bit	Name	RW	Reset Value	Function
31	LPWRRSTF	R	0	低功耗复位标志。 当进入非法的停止/待机低功耗模式时，硬件置位该寄存器。 RMVF 置 1 会清零该位。
30	WWDGRSTF	R	0	窗口看门狗复位标志。 RMVF 置 1 会清零该位。
29	IWDGRSTF	R	0	独立看门狗复位标志。 RMVF 置 1 会清零该位。
28	SFTRSTF	R	0	软复位标志。 RMVF 置 1 会清零该位。
27	PWRRSTF	R	1	POR/PDR 复位标志。 RMVF 置 1 会清零该位。
26	PINRSTF	R	0	外部 NRST 管脚复位标志。 RMVF 置 1 会清零该位。
25	OBLRSTF	R	0	选项字节加载复位标志。 RMVF 置 1 会清零该位。
24:20	Reserved	-	-	保留
19	PLLULRSTF	R	0	PLLUL(锁相环失锁)复位标志 RMVF 置 1 会清除该位
18:9	Reserved	-	-	保留

Puya Confidential

8	RMVF	RW	0	软件置位清零复位标志。 写该位为 1 的操作会清零复位标志，软件写 1 后该位会维持为 1，硬件不会清零。
7:6	Reserved	-	-	保留
5	EPWM_PRST_RELEASE	R	0	EPWM 总线（APB）复位状态 0：EPWM 正在总线复位 1：EPWM 总线复位已释放
4	EPWM_SFTRST_RELEASE	R	1	EPWM 软件复位状态 0：EPWM 正在软件复位 1：EPWM 软件复位已释放
3:1	Reserved	-	-	保留
0	NRST_FLTDIS	RW	0	NRST 滤波禁止 0：使能 HSI_10M，且滤波 20 μs 宽度功能使能 1：滤波功能禁止，且 HSI_10M 保持关闭

Puya Confidential

7. CRC 计算单元（CRC）

7.1 CRC 简介

循环冗余校验（CRC）单元可以利用 32 位的以太网多项式计算出 CRC 码。

在其他的应用中，基于 CRC 的技术用于验证数据传输或存储的完整性。在功能安全标准的领域内，基于 CRC 的技术提供了一种验证闪存完整性的方法。CRC 计算单元可帮助软件签名的计算。

7.2 CRC 主要特性

$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- 仅支持对 32 位的数据的计算
- CRC 初始值为 0xFFFFFFFF
- 具有一个通用 8 位寄存器（可用于存放临时数据）
- 具备输入缓存功能
- 具有一个 32 位数据寄存器用于输入 / 输出
- CRC 计算时间：4 个 AHB 时钟周期（32 位数据）

7.3 CRC 功能描述

7.3.1 CRC 结构框图

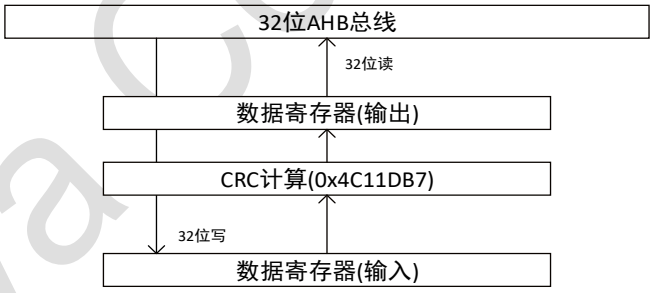


图 7-1 CRC 模块的结构

7.3.2 CRC 操作

CRC 计算单元含有 1 个 32 位数据寄存器（CRC_DR）。

对于写操作，它用于保存输入数据；对于读操作，它用于保存之前的 CRC 计算结果。

每一次写入数据寄存器，其计算结果是前一次 CRC 计算结果和新计算结果的组合（对整个 32 位字进行 CRC 计算，而不是逐字节地计算）。

输入缓存允许用户在写入第一个数据之后立刻写入第二个数据，无需等待之前的数据计算完毕。

可以通过向寄存器 CRC_CR 的 RESET 位写 1 来重置寄存器 CRC_DR 为 0xFFFFFFFF。该操作不影响寄存器 CRC_IDR 内的数据。

Puya Confidential

CRC_IDR 寄存器可以用来保存与 CRC 计算相关的临时值。它不受 CRC_CR 寄存器中 RESET 位的影响。

7.4 CRC 寄存器

7.4.1 CRC 数据寄存器（CRC_DR）

偏移地址：0x00

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	DR[31:0]	RW	32'hFFFFFFFF	数据寄存器位。 写入 CRC 计数器的新数据时，作为输入寄存器。 读取时返回之前的 CRC 计算的结果。

7.4.2 CRC 独立数据寄存器（CRC_IDR）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7:0	IDR[7:0]	RW	8'h0	通用 8 位数据寄存器位。可临时存放数据。 寄存器 CRC_CR 的 RESET 位产生的 CRC 复位对本寄存器无影响。 注：此寄存器不参与 CRC 计算，可以存放任何数据。

7.4.3 CRC 控制寄存器（CRC_CR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RESET
															W

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:1	Reserved	-	-	保留
0	RESET	W	0	软件往此位写 1 将会复位 CRC 模块。 软件只能写 1，由硬件清零。

Puya Confidential

8. 通用 IO（GPIO）

8.1 GPIO 简介

每个通用 IO 端口包括：

4 个 32 位配置寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR 和 GPIOx_PUPDR）

2 个 32 位数据寄存器（GPIOx_IDR 和 GPIOx_ODR）

1 个 32 位置位/复位寄存器（GPIOx_BSRR)、一个 32 位复位寄存器（GPIOx_BRR）

1 个 32 位锁定寄存器（GPIOx_LCKR)

2 个 32 位复用功能选择寄存器（GPIOx_AFRH 和 GPIOx_AFRL）。

8.2 GPIO 主要特征

- 受控 IO 最多达 60 个
- 输出状态：推挽或开漏 + 上拉/下拉
- 从输出数据寄存器（GPIOx_ODR）或外设（复用功能输出）输出到引脚
- 可为每个 IO 选择不同的速度
- 输入状态：浮空、上拉/下拉、模拟
- 将引脚输入到数据寄存器（GPIOx_IDR）或外设（复用功能输入）
- 置位/复位寄存器（GPIOx_BSRR），对 GPIOx_ODR 具有按位写权限
- 复位寄存器（GPIOx_BRR），对 GPIOx_ODR 具有按位写权限
- 锁定寄存器（GPIOx_LCKR），可冻结 IO 配置
- 复用功能输入/输出选择寄存器（一个 IO 最多可具有 16 个复用功能）
- 快速翻转，每次翻转最快只需要两个时钟周期
- 引脚复用非常灵活，允许将 IO 引脚用作 GPIO 或多种外设功能中的一种

8.3 GPIO 功能说明

根据数据手册中列出的每个 IO 端口的特性，可通过软件将通用 IO（GPIO）端口的各个端口位分别配置为多种模式：

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟功能
- 具有上拉或下拉功能的开漏输出
- 具有上拉或下拉功能的推挽输出
- 具有上拉或下拉功能的复用功能推挽
- 具有上拉或下拉功能的复用功能开漏

Puya Confidential

每个 IO 端口位均可自由编程，但 IO 端口寄存器必须按字、半字或字节进行访问。GPIOx_BSRR 和 GPIOx_BRR 寄存器允许对任何 GPIO 寄存器的读/更改的独立访问。这样，在读和更改访问之间产生 IRQ 时不会发生危险。

下图显示了 5 V 兼容和基本 IO 端口位的基本结构。表 8-1 给出了可能的端口位配置方案。

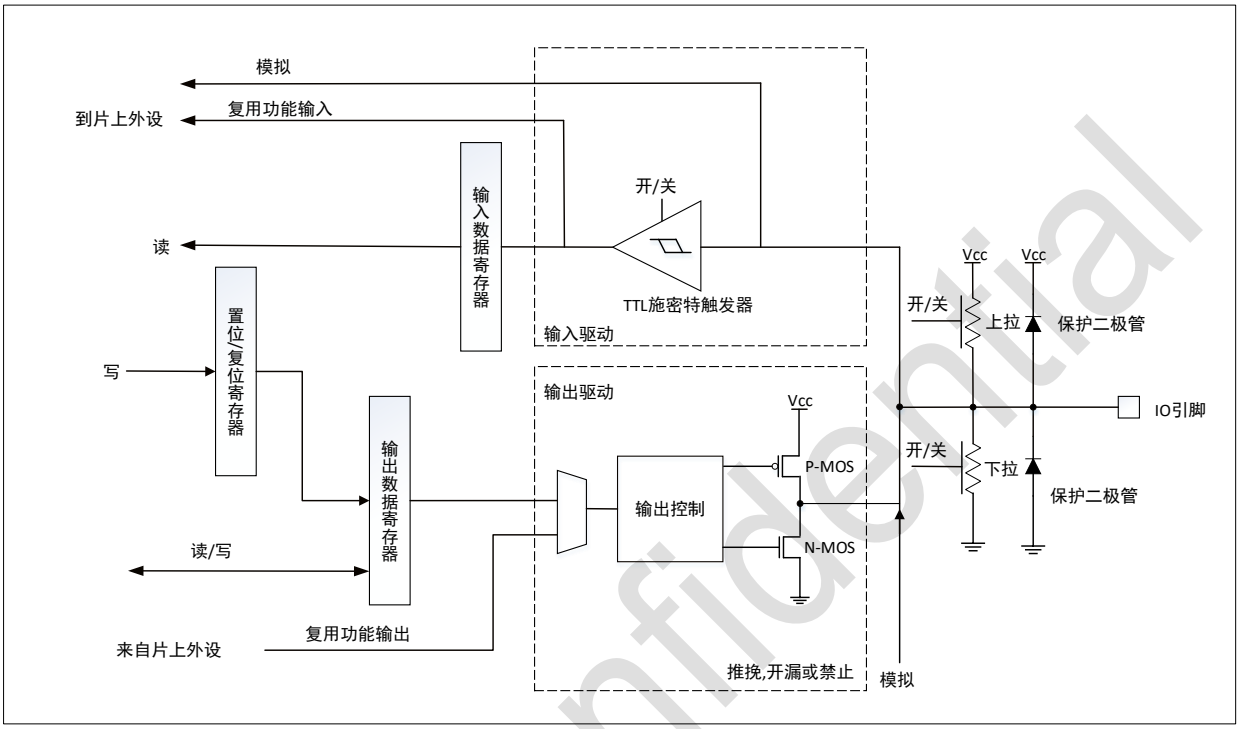


图 8-1 IO 端口基本结构

表 8-1 端口位配置表

MODE(i)[1:0]	OTYPE(i)	OSPEED(i)[1:0]		PUPD(i)[1:0]		IO configuration	
01	0	SPEED[1:0]		0	0	GP output	PP
	0			0	1	GP output	PP+PU
	0			1	0	GP output	PP+PD
	0			1	1	Reserved	
	1			0	0	GP output	OD
	1			0	1	GP output	OD+PU
	1			1	0	GP output	OD+PD
	1			1	1	Reserved(GP output OD)	
10	0	SPEED[1:0]		0	0	AF	PP
	0			0	1	AF	PP+PU
	0			1	0	AF	PP+PD
	0			1	1	Reserved	
	1			0	0	AF	OD
	1			0	1	AF	OD+PU
	1			1	0	AF	OD+PD
	1			1	1	Reserved	
00	x	x	x	0	0	Input	Floating
	x	x	x	0	1	Input	PU
	x	x	x	1	0	Input	PD
	x	x	x	1	1	Reserved(input floating)	
11	x	x	x	0	0	Input/Output	Analog
	x	x	x	0	1	Reserved	
	x	x	x	1	0	Input/Output	Analog,PD
	x	x	x	1	1	Reserved	

GP = general-purpose, PP = push-pull, PU = pull-up, PD = pull-down, OD = open-drain, AF = alternate function.

Puya Confidential

8.3.1 通用 IO（GPIO）

在复位期间及复位刚刚完成后，复用功能尚未激活，大部分 IO 端口被配置为模拟模式。复位后，调试引脚处于复用功能上拉/下拉状态，BOOT0 引脚处于输入模式下拉状态：

- PD6：JTDI 处于上拉状态
- PD7：JTDO 处于浮空状态无上下拉
- PA11：JTCK/SWCLK 处于下拉状态
- PA14：JTCK/SWCLK 处于下拉状态
- PA13：JTMS/SWDIO 处于上拉状态
- PA12：NJTRST 处于上拉状态
- PD3：BOOT0 处于下拉状态

输入数据寄存器（GPIOx_IDR）每隔 1 个 AHB 时钟周期捕获一次 IO 引脚的数据。

所有 GPIO 引脚都具有内部弱上拉及下拉电阻，可根据 GPIOx_PUPDR 寄存器中的值来打开/关闭。

8.3.2 IO 引脚复用器和映射

IO 引脚通过一个复用器连接到板载外设/模块，该复用器一次仅允许一个外设的复用功能（AF）连接到 IO 引脚。这可以确保共用同一个 IO 引脚的外设之间不会发生冲突。

每个 IO 引脚都有一个复用器，该复用器采用 16 路复用功能输入（AF0 到 AF15），可通过 GPIOx_AFRL（用于引脚 0 到 7）和 GPIOx_AFRH（用于引脚 8 到 15）寄存器对这些输入进行配置。

要将 IO 配置成所需功能，请按照以下步骤操作：

调试功能

系统复位后，调试相关的引脚复用为调试功能，供系统调试接口使用。

GPIO

在 GPIOx_MODER 寄存器中将所需 IO 配置为输出或输入。

外设复用功能

- 在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中，设置 IO 的复用功能
- 通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDR 寄存器，分别选择输出类型、上拉/下拉以及输出速度
- 在 GPIOx_MODER 寄存器中将所需 IO 配置为复用功能

额外功能

- 对于 ADC，OPA，COMP，DAC 和 VREFBUF，需要通过寄存器 GPIOx_MODER 配置相应的 IO 为模拟模式，并在 ADC，OPA，COMP，DAC 和 V_{REFBUF} 中配置相应功能；需要特别注意，对于模拟附加功能（如 OPA、DAC、V_{REFBUF}）的输出，需要先配置相应 IO 为模拟模式，再使能外设的控制寄存器。
- 对于额外功能 WKUPx 和 OSC，需要在 PWR 和 RCC 模块配置相应功能，这些功能优先于 GPIO 设定。

8.3.3 IO 端口控制寄存器

每个 GPIO 有 4 个 32 位控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR），可配置多达 16 个 IO。GPIOx_MODER 寄存器用于选择 IO 工作模式（输入、输出、

Puya Confidential

复用、模拟）。GPIOx_OTYPER 用于选择输出类型（推挽或开漏）。GPIOx_OSPEEDR 寄存器用于设定 IO 速度。GPIOx_PUPDR 寄存器用于选择上拉/下拉。

8.3.4 IO 端口数据寄存器

每个 GPIO 都具有 2 个 16 位数据寄存器：输入和输出数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。GPIOx_ODR 用于存储待输出数据，可对其进行读/写访问。IO 输入数据存储到 GPIOx_IDR 中，它是一个只读寄存器。

8.3.5 IO 数据位操作

置位复位寄存器(GPIOx_BSRR)是一个 32 位寄存器,它允许应用程序在输出数据寄存器(GPIOx_ODR)中对各个单独的数据位执行置位和复位操作。

GPIOx_ODR 中的每个数据位对应于 GPIOx_BSRR 中的两个控制位: BS(i)和 BR(i)。当写入 1 时, BS(i)位会置位对应的 ODR(i)位。当写入 1 时，BR(i)位会清零 ODR(i)对应的位。

在 GPIOx_BSRR 中向任何位写入 0 都不会对 GPIOx_ODR 中的对应位产生任何影响。如果在 GPIOx_BSRR 中同时尝试对某个位执行置位和清零操作，则置位操作优先。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各个位的值是一个“单次”操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器提供了一种执行原子按位处理的方法。

在对 GPIOx_ODR 进行位操作时，软件无需禁止中断：在一次原子 AHB 写访问中，可以修改一个或多个位。

复位寄存器（GPIOx_BRR）是一个 32 位寄存器，它允许应用程序在输出数据寄存器（GPIOx_ODR）中对各个单独的数据位执行复位操作。复位寄存器的作用与置位复位寄存器相似，但仅能复位输出数据寄存器的一个或多个位。

8.3.6 GPIO 锁定机制

通过将特定的写序列应用到 GPIOx_LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 GPIOx_MODER 、 GPIOx_OTYPER 、 GPIOx_OSPEEDR 、 GPIOx_PUPDR 、 GPIOx_AFRL 和 GPIOx_AFRH。

要对 GPIOx_LCKR 寄存器执行写操作，必须用特定的写/读序列，软件以正确的锁定序列操作此寄存器的 LCKR[16]后，会使用 LCKR[15:0]的值来锁定 IO 的配置（在写序列期间，LCKR[15:0]的值必须保持不变）。某个或多个端口被锁定后，无法解锁，直到发生系统复位，或 GPIO 被复位。每个 GPIOx_LCKR 位都会冻结控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH）中的对应位。

锁定序列只能通过字访问对 GPIOx_LCKR 寄存器操作。

具体操作参考 GPIOx_LCKR 寄存器描述。

8.3.7 IO 复用功能输入/输出

有两个寄存器可用来从每个 IO 可用的 16 个复用功能输入/输出中进行选择。

不同的复用信号有不同的输入/输出方向，具体情况由各自的外设模块决定。

Puya Confidential

8.3.8 外部中断线/唤醒线

所有端口都具有外部中断功能。要使用外部中断线，必须将端口配置为输入模式，请参见中断和事件章节。

8.3.9 输入配置

当 IO 口配置为输入：

- 输出缓冲器不使能
- 施密特触发器输入使能
- 根据寄存器 GPIOx_PUPDR 配置可使能/不使能上下拉电阻
- 每个 AHB 时钟采样引脚电平，并存放于输入数据寄存器
- 读取输入数据寄存器可得知引脚电平

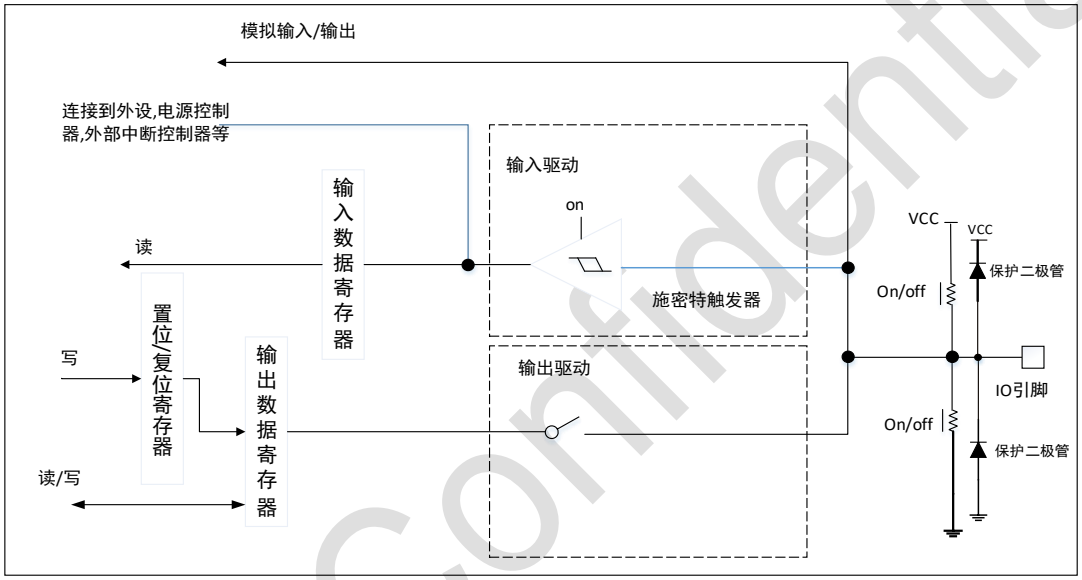


图 8-2 输入浮空/上拉/下拉配置

8.3.10 输出配置

当 IO 端口被配置为输出时：

- 输出缓冲器使能

– 开漏模式：输出寄存器上的’0’导通 N-MOS，而输出寄存器上的’1’将端口置于高阻状态（PMOS 不导通）。

– 推挽模式：输出寄存器上的’0’导通 N-MOS，而输出寄存器上的’1’将导通 P-MOS。

- 施密特触发输入使能
- 根据寄存器 GPIOx_PUPDR 配置可使能/不使能上下拉电阻
- 每个 AHB 时钟采样引脚电平，并存放于输入数据寄存器
- 读取输入数据寄存器可得知引脚电平
- 读取输出数据寄存器可得知上一次写 GPIO 的值

Puya Confidential

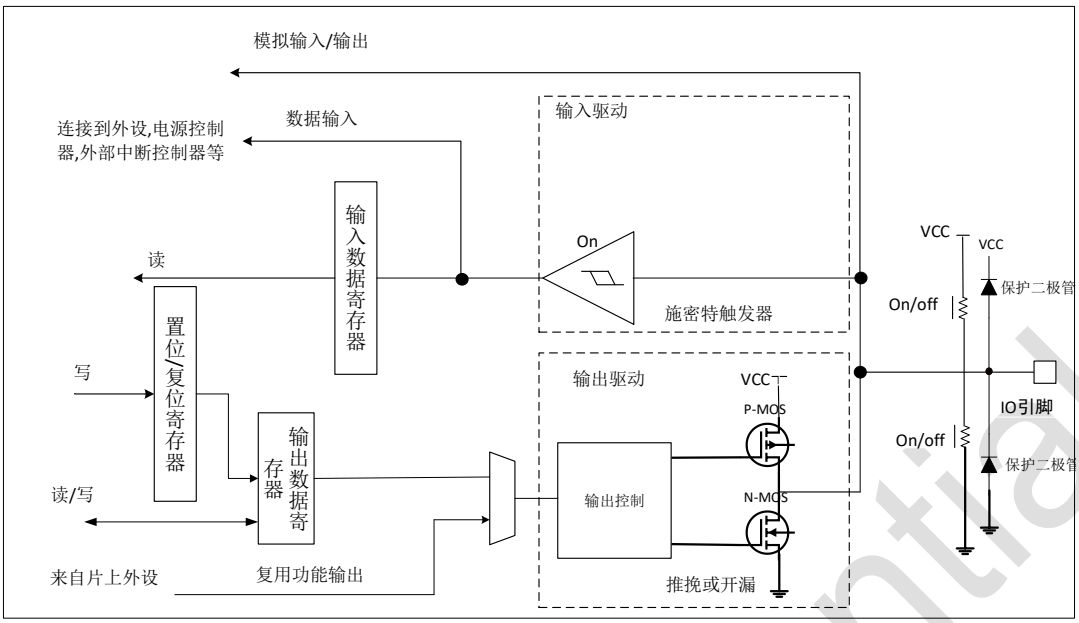


图 8-3 输出配置

8.3.11 复用功能配置

当 IO 端口被配置为复用功能时：

- 输出缓冲器使能
- 输出缓冲器的输出数据和输出使能由外设模块驱动
- 施密特触发输入使能
- 根据寄存器 GPIOx_PUPDR 配置可使能/不使能上下拉电阻
- 每个 AHB 时钟采样引脚电平，并存放于输入数据寄存器
- 读取输出数据寄存器可得知上一次写 GPIO 的值

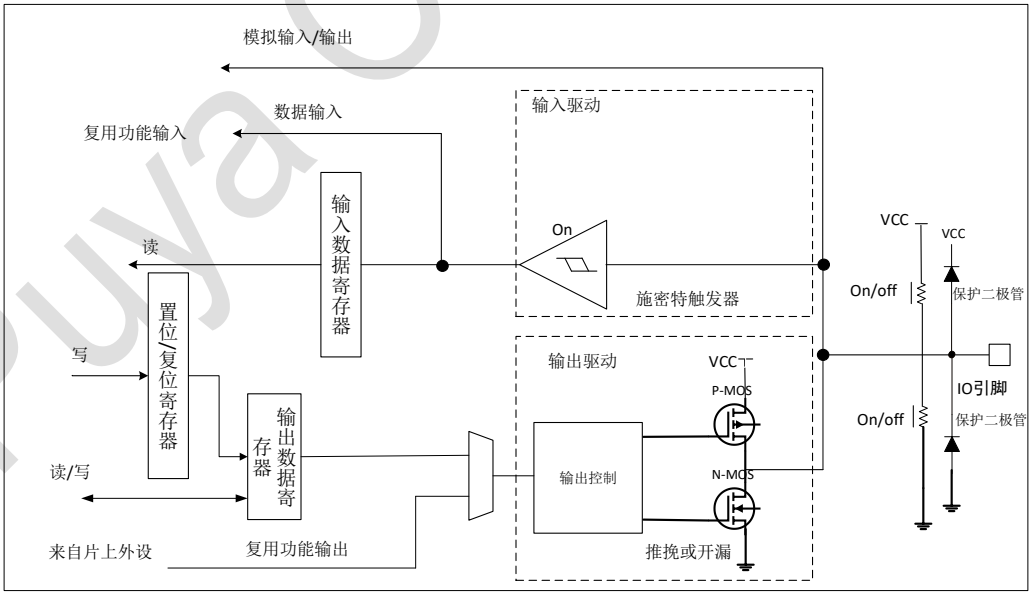


图 8-4 复用功能配置

8.3.12 模拟配置

当 IO 端口被配置为模拟模式时：

Puya Confidential

- 输出缓冲器不使能；
- 施密特触发器输入不使能，实现了每个模拟 IO 引脚上的零功耗。施密特触发输出值被强置为'0'；
- 弱上拉和下拉电阻被禁止（需要软件设定）；
- 读取输入数据寄存器时数值为'0'。

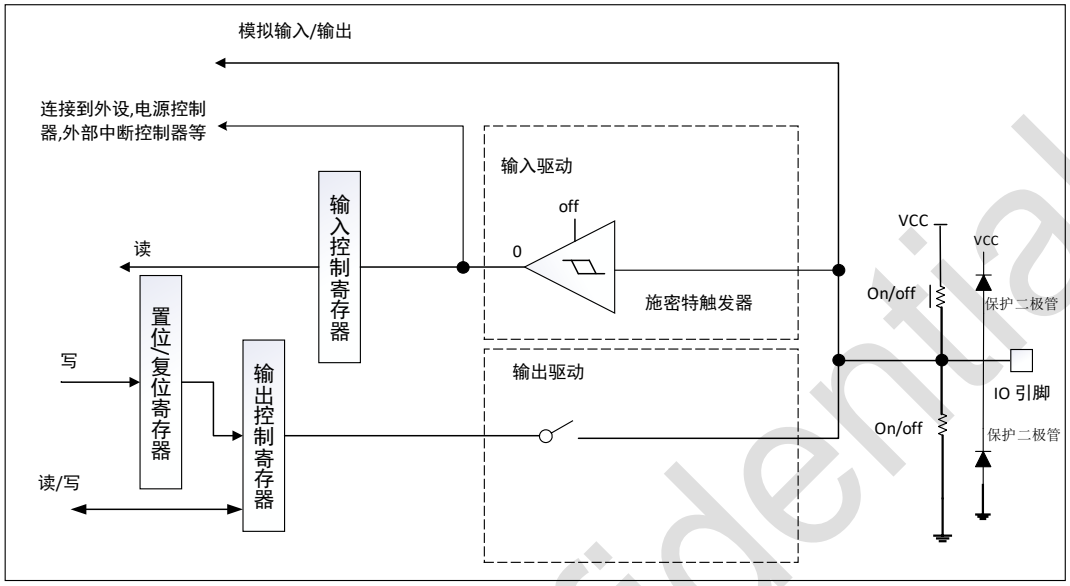


图 8-5 高阻抗模拟配置

8.3.13 使用 PD0、PD1（HSE 振荡器引脚）作为 GPIO

当 HSE 功能不使能（复位后的默认），相应的管脚可以当作正常的 GPIO。

当 HSE 功能使能（RCC 寄存器中置位 HSEON），相关引脚的功能由振荡器控制，不受这些 GPIO 寄存器的设定影响。

当晶振配置为用户外部时钟模式，只有 OSC_IN（PD0）用于接收外部时钟输入，而 OSC_OUT（PD1）脚仍然可以用作正常 GPIO。

8.3.14 使用 PD3（BOOT0）作为 GPIO

选项字节中的 nSWBOOT0 位选择：

- nSWBOOT0=1，复位中为 BOOT0 引脚。
- nSWBOOT0=0，复位中为 GPIO 引脚。

注：不同封装的默认模式请参考数据手册中的引脚配置章节。

8.3.15 使用 PD11 作为 GPIO

PD11 可以用作复位引脚（NRST）或 GPIO。根据用户选项字节中的 NRST_MODE 位，它切换到以下模式：

- 复位输入/输出：NRST_MODE=2'b11 或 NRST_MODE=2'b00
- 仅复位输入：NRST_MODE=2'b01
- GPIO PD11 模式：NRST_MODE=2'b10

注：不同封装的默认模式请参考数据手册中的引脚配置章节。

Puya Confidential

8.4 GPIO 寄存器

该章节详细描述 GPIO 的寄存器功能

有关寄存器位、寄存器偏移地址和复位值的汇总，请参见 GPIO 寄存器映射表；

可通过字节（8 位）、半字（16 位）或字（32 位）对 GPIO 寄存器进行访问。

GPIOA 基地址: 0x4002 4000

GPIOB 基地址: 0x4002 4400

GPIOC 基地址: 0x4002 4800

GPIOD 基地址: 0x4002 4C00

8.4.1 GPIO 端口模式寄存器（GPIOx_MODER）(x= A..D)

偏移地址：0x00

复位值：0xEABF_FFFF（端口 A）、0xFFFF_FFFF（端口 B）、0xFFFF FFFF（端口 C）

0x00FF AF3F（端口 D）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	MODEy[1:0]	RW	PD3：2’b00 PA11~PA14：2’b10 PD6、PD7：2’b10 其他：2’b11	y = 15..0 软件通过这些位配置相应的 IO 模式 00：输入模式 01：通用输出模式 10：复用功能模式 11：模拟模式(复位状态)

8.4.2 GPIO 端口输出类型寄存器（GPIOx_OTYPER）(x= A..D)

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	OTy	RW	0	y = 15..0 软件配置 IO 的输出类型 0：推挽输出（复位状态） 1：开漏输出

Puya Confidential

8.4.3 GPIO 端口输出速度寄存器（GPIOx_OSPEEDR）(x= A..D)

偏移地址：0x0C

复位值：0x0C00 0000（端口 A）、0x0000 0000（端口 B）、0x0000 0000（端口 C）、

0x0000 0000（端口 D）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	OSPEEDy[1:0]	RW	PA13：2’b11 其他：2’b00	y = 15..0 软件配置 IO 口的输出速度 00：非常低 01：低速 10：高速 11：非常高

8.4.4 GPIO 端口上拉/下拉寄存器（GPIOx_PUPDR）(x= A..D)

偏移地址：0x0C

复位值：0x2580 0000（端口 A）、0x0000 0000（端口 B）、0x0000 0000（端口 C）、

0x0000 1080（端口 D）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	PUPDy [1:0]	RW	PA11、PA14：2’b10 PA12、PA13：2’b01 PD6：2’b01 PD3：2’b10 其他：2’b00	y = 15..0 软件配置 IO 口上拉或者下拉 00：无上下拉 01：上拉 10：下拉 11：保留

8.4.5 GPIO 端口输入数据寄存器（GPIOx_IDR）(x= A..D)

偏移地址：0x10

复位值：0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留

Puya Confidential

15:0	IDy	R		y = 15..0 这是只读的，读出值位对应 IO 口的状态

8.4.6 GPIO 端口输出数据寄存器（GPIOx_ODR）(x= A..D)

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15: 0	ODy	RW	0	y = 15..0 对 GPIOx_BSRR 或 GPIOx_BRR 寄存器 (x=A,B,C,D,E,F)，可以分别对各个 ODR 位进行独立的设置/清除。

8.4.7 GPIO 端口置位/复位寄存器（GPIOx_BSRR）(x= A..D)

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:16	BRy	W	0	y = 15..0 读返回值是 0 0: 对应的 ODRy 位不受影响 1: 清除对应的 ODRy 位 注：如果同时写 BSy 和 BRy 的对应位，BSy 位起作用
15: 0	BSy	W	0	y = 15..0 读返回值是 0 0: 对应的 ODRy 位不受影响 1: 设置对应的 ODRy 位

8.4.8 GPIO 端口配置锁定寄存器（GPIOx_LCKR）(x= A..D)

当执行正确的写序列置位 LCKK 时，该寄存器用来锁定端口位的配置。LCKR[15:0]用于锁定 GPIO 端口的配置。在规定的写入操作期间，不能改变 LCKR[15:0]。对相应的端口执行锁定序列后，在下次系统复位前将不能再更改端口位的配置。

注：特殊写时序用来写 GPIOx_LCKR 寄存器。在锁定时序中只允许字访问。

每个锁定位冻结一种特定的配置寄存器（控制和复用功能寄存器）

偏移地址：0x1C

Puya Confidential

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:17	Reserved	-	-	保留
16	LCKK	RW	0	该位可随时读出，它只能通过锁键写入序列修改 0：端口配置锁键位未激活 1：端口配置锁键位被激活，下次系统复位前 GPIOx_LCKR 寄存器被锁定 锁键值写序列： 写 1->写 0->写 1->读 0->读 1，最后一个读可省略，但可以用来确认锁键已被激活。 注：在操作锁键的写入序列时，不能改变 LCK[15:0]的值。锁键时序的任何错误都会终止锁键被激活。对端口的任何一位首次锁键时序之后，读 LCKK 位都是返回 1,直到系统复位或者 GPIO 复位。
15: 0	LCKy	RW	0	y = 15..0 只能在 LCKK 位为 0 时写入。 0：不锁定端口的配置 1：锁定端口配置

8.4.9 GPIO 复用功能低寄存器（GPIOx_AFRL）(x= A..D)

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	AFSELy[3:0]	RW	4'h0	y = 7..0 配置复用功能 IO AFSELy 选择: 0000：AF0　1000：AF8 0001：AF1　1001：AF9 0010：AF2　1010：AF10 0011：AF3　1011：AF11 0100：AF4　1100：AF12

Puya Confidential

				0101：AF5	1101：AF13
				0110：AF6	1110：AF14
				0111：AF7	1111：AF15

8.4.10 GPIO 复用功能高寄存器（GPIOx_AFRH）(x= A..D)

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	AFSELy[3:0]	RW	**4'h0**	y = 15..8 配置复用功能 IO AFSELy 选择: 0000：AF0　1000：AF8 0001：F1　1001：AF9 0010：AF2　1010：AF10 0011：AF3　1011：AF11 0100：AF4　1100：AF12 0101：AF5　1101：AF13 0110：AF6　1110：AF14 0111：AF7　1111：AF15

8.4.11 GPIO 端口位复位寄存器（GPIOx_BRR）(x= A..D)

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**2**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	BRy	RW	**0**	y = 15..0 读返回值是 0 0：对应的 ODy 位受影响 1：清除对应的 ODy 位

Puya Confidential

9. 外设互联

9.1 简介

一些外围设备之间有连接关系。

这允许在外设之间进行自主通信，从而节省 CPU 资源，同时也节省电源消耗。此外这些硬件连接消除了软件延迟。根据外设的不同，这些互联可以在运行、睡眠和停止的模式下运行。

9.2 互联详情

9.2.1 定时器输入触发（ITR）

目的

一些 TIMx 定时器在内部连接在一起，用于定时器的同步或者连接。当一个定时器为主模式时，它可以重置、启动、停止从模式定时器。

触发信号

在可配置定时器事件发生后，输出（来自主设备）出现在 TIMx_TRGO 信号和 EPWMx_SINCO 上。输入（到从设备）位于 TIMx_ITR0/ITR1/ITR2/ITR3 信号上。

表 9-1 定时器输入触发

TIMx 内部触发	Timer input trigger source assignment			
	TIM1	TIM2	TIM3	TIM8
timx_itr0	Reserved	TIM1_TRGO	TIM1_TRGO	TIM1_TRGO
timx_itr1	TIM2_TRGO	Reserved	TIM2_TRGO	TIM2_TRGO
timx_itr2	TIM3_TRGO	TIM3_TRGO	Reserved	TIM3_TRGO
timx_itr3	TIM8_TRGO	TIM8_TRGO	TIM8_TRGO	Reserved
timx_itr4	EPWM1_SYNCO	EPWM1_SYNCO	EPWM1_SYNCO	EPWM1_SYNCO
timx_itr5	EPWM2_SYNCO	EPWM2_SYNCO	EPWM2_SYNCO	EPWM2_SYNCO
timx_itr6	EPWM3_SYNCO	EPWM3_SYNCO	EPWM3_SYNCO	EPWM3_SYNCO
timx_itr7	EPWM4_SYNCO	EPWM4_SYNCO	EPWM4_SYNCO	EPWM4_SYNCO
timx_itr8	EPWM5_SYNCO	EPWM5_SYNCO	EPWM5_SYNCO	EPWM5_SYNCO
timx_itr9	EPWM6_SYNCO	EPWM6_SYNCO	EPWM6_SYNCO	EPWM6_SYNCO

9.2.2 定时器外部触发（ETR）

目的

ADC 模拟看门狗信号输出、比较器的输出，以及定时器 ETR 引脚可以连接到定时器的外部触发 ETR。

触发信号

ADC 模拟看门狗输出 ADC1_AWD，ADC2_AWD、比较器的输出 COMP1_OUT、COMP2_OUT、COMP3_OUT、COMP4_OUT。

表 9-2 定时器外部触发

TIMx/PWM 外部触发	Timer input trigger source assignment			
	TIM1	TIM2	TIM3	TIM8
timx_etr0	TIM1_ETR	TIM2_ETR	TIM3_ETR	TIM8_ETR
timx_etr1	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
timx_etr2	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
timx_etr3	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT

Puya Confidential

timx_etr4	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT
timx_etr5	-	-	-	-
timx_etr6	-	-	-	-
timx_etr7	-	-	-	-
timx_etr8	ADC1_AWD1	TIM3_ETR	TIM2_ETR	ADC1_AWD1
timx_etr9	ADC1_AWD2	ADC1_AWD2	ADC1_AWD2	ADC1_AWD2
timx_etr10	ADC1_AWD3	ADC1_AWD3	ADC1_AWD3	ADC1_AWD3
timx_etr11	ADC2_AWD1	ADC2_AWD1	ADC2_AWD1	ADC2_AWD1
timx_etr12	ADC2_AWD2	ADC2_AWD2	ADC2_AWD2	ADC2_AWD2
timx_etr13	ADC2_AWD3	ADC2_AWD3	ADC2_AWD3	ADC2_AWD3

相关功耗模式

这些互连在运行、睡眠、低功耗运行和低功耗睡眠功耗模式下工作。

9.2.3 清除定时器 OCxREF 信号

目的

比较器的输出，可以用于清除高级定时器 TIM1/TIM8、通用定时器 TIM2/TIM3 的 OCxREF 信号。

触发信号

比较器的输出 COMP1_OUT、COMP2_OUT、COMP3_OUT、COMP4_OUT.

表 9-3 清除定时器 OCxREF 信号

Timx OCREF CLR	Timer OCREF clear signals assignment			
	TIM1	TIM2	TIM3	TIM8
timx_ocref_clr0	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
timx_ocref_clr1	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
timx_ocref_clr2	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT
timx_ocref_clr3	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT

相关功耗模式

这些互连在运行、睡眠、低功耗运行和低功耗睡眠功耗模式下工作。

9.2.4 定时器输入捕获

目的

比较器的输出，可以用作高级定时器 TIM1/TIM8，通用定时器 TIM2/TIM3 的输入捕获通道。

表 9-4 输入捕获通道 1（TI1）源

TIMx TI1	Timer TI1 signals assignment			
	TIM1	TIM2	TIM3	TIM8
timx_ti1_in0	TIM1 external TI1 input pins	TIM2 external TI1 input pins	TIM3 external TI1 input pins	TIM8 external TI1 input pins
timx_ti1_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
timx_ti1_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
timx_ti1_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT
timx_ti1_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT
timx_ti1_in5	MCO1_CLK	MCO2_CLK	MCO1_CLK	MCO2_CLK

表 9-5 输入捕获通道 2（TI2）源

TIMx TI2	Timer TI2 signals assignment			
	TIM1	TIM2	TIM3	TIM8
timx_ti2_in0	TIM1 external TI2 input pins	TIM2 external TI2 input pins	TIM3 external TI2 input pins	TIM8 external TI2 input pins
timx_ti2_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
timx_ti2_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
timx_ti2_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT
timx_ti2_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT

Puya Confidential

表 9-6 输入捕获通道 3（TI3）源

TIMx TI3	Timer TI3 signals assignment			
	TIM1	TIM2	TIM3	TIM8
timx_ti3_in0	TIM1 external TI3 input pins	TIM2 external TI3 input pins	TIM3 external TI3 input pins	TIM8 external TI3 input pins
timx_ti3_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
timx_ti3_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
timx_ti3_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT
timx_ti3_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT

表 9-7 输入捕获通道 4（TI4）源

TIMx TI4	Timer TI4 signals assignment			
	TIM1	TIM2	TIM3	TIM8
timx_ti4_in0	TIM1 external TI4 input pins	TIM2 external TI4 input pins	TIM3 external TI4 input pins	TIM8 external TI4 input pins
timx_ti4_in1	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
timx_ti4_in2	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
timx_ti4_in3	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT
timx_ti4_in4	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT

相关功耗模式

这些互连在运行、睡眠、低功耗运行和低功耗睡眠功耗模式下工作。

9.2.5 定时器刹车

目的

比较器的输出，可以用作高级定时器 TIM1/TIM8，通用定时器 TIM2/TIM3 的刹车。

触发信号

比较器的输出 COMP1_OUT、COMP2_OUT、COMP3_OUT、COMP4_OUT，以及刹车引脚输入 TIMx_BRK。

表 9-8 定时器刹车输入

TIMx 刹车	Timer break signals assignment			
	TIM1 break(tim1_bk)	TIM2 break(tim2_bk)	TIM3 break(tim3_bk)	TIM8 break(tim8_bk)
TIM_BKIN	TIM1_BKIN pin	TIM2_BKIN pin	TIM3_BKIN pin	TIM8_BKIN pin
tim_brk_cmp1	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_brk_cmp2	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_brk_cmp3	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_brk_cmp4	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT

表 9-9 定时器刹车 2 输入

TIMx 刹车	Timer break2 signals assignment			
	TIM1 break2(tim1_bk2)	TIM2 break2(tim2_bk2)	TIM3 break2(tim3_bk2)	TIM8 break2(tim8_bk2)
TIM_BKIN2	TIM1_BKIN2pin	TIM2_BKIN2pin	TIM3_BKIN2pin	TIM8_BKIN2pin
tim_brk2_cmp1	COMP1_OUT	COMP1_OUT	COMP1_OUT	COMP1_OUT
tim_brk2_cmp2	COMP2_OUT	COMP2_OUT	COMP2_OUT	COMP2_OUT
tim_brk2_cmp3	COMP3_OUT	COMP3_OUT	COMP3_OUT	COMP3_OUT
tim_brk2_cmp4	COMP4_OUT	COMP4_OUT	COMP4_OUT	COMP4_OUT

相关功耗模式

这些互连在运行、睡眠、低功耗运行和低功耗睡眠功耗模式下工作。

9.2.6 定时器系统刹车

目的

HSE CSS、CPU LOCKUP，PVD 报警可面向 TIM1、TIM2、TIM3、TIM8 生成系统错误。

刹车功能的目的是保护由这些定时器生成的 PWM 信号所驱动的功率器件。

Confidential

表 9-10 定时器系统刹车

System error signal source					
tim_sys_brk inputs	**System error signal source**	**TIM1**	**TIM2**	**TIM3**	**TIM8**
tim_sys_brk0	Cortex®-M4 with FPU LOCKUP	Y	Y	Y	Y
tim_sys_brk1	Programmable Voltage Detector （PVD）	Y	Y	Y	Y
tim_sys_brk4	Clock Security System（CSS）	Y	Y	Y	Y

相关功耗模式

这些互连在运行、睡眠、低功耗运行和低功耗睡眠功耗模式下工作。

9.2.7 低功耗定时器触发

目的

RTC 报警、外部信号（LPTIM1_ETR）、COMPx_OUT 可用作启动 LPTIM 计数器的触发器。

表 9-11 LPTIM 外部触发

LPTIM trigger input signal (TRIGSEL[3:0])	LPTIM1 trigger source assignment
lptim1_ext_trig0	LPTIM1_ETR
lptim1_ext_trig1	RTC_ALRA_TRG
lptim1_ext_trig2	COMP1_OUT
lptim1_ext_trig3	COMP2_OUT
lptim1_ext_trig4	COMP3_OUT
lptim1_ext_trig5	COMP4_OUT

9.2.8 比较器消隐

目的

高级控制定时器 TIM1、TIM8 和通用定时器 TIM2、TIM3 和 EPWM 可用作 COMP 的消隐窗口输入。

触发信号

定时器输出信号 TIMx_CHx/EPWMx_BLANK 是 COMP 的消隐源输入。

表 9-12 比较器消隐输入

Comparator blanking signal BLANKSEL[2:0]	Comparator blanking source assignment			
	COMP1	**COMP2**	**COMP3**	**COMP4**
1	TIM1_CH5	TIM1_CH5	TIM1_CH5	TIM3_CH4
2	TIM2_CH3	TIM2_CH3	TIM2_CH3	TIM8_CH5
3	TIM3_CH3	TIM3_CH3	TIM3_CH3	TIM2_CH4
4	TIM8_CH5	TIM8_CH5	TIM8_CH5	TIM1_CH5
5	EPWM1_BLANK	EPWM1_BLANK	EPWM1_BLANK	EPWM1_BLANK
6	EPWM2_BLANK	EPWM2_BLANK	EPWM2_BLANK	EPWM2_BLANK
7	EPWM3_BLANK	EPWM3_BLANK	EPWM3_BLANK	EPWM3_BLANK
8	EPWM4_BLANK	EPWM4_BLANK	EPWM4_BLANK	EPWM4_BLANK
9	EPWM5_BLANK	EPWM5_BLANK	EPWM5_BLANK	EPWM5_BLANK
10	EPWM6_BLANK	EPWM6_BLANK	EPWM6_BLANK	EPWM6_BLANK

相关功耗模式

这些互连在运行、睡眠、低功耗运行和低功耗睡眠功耗模式下工作。

9.2.9 ADC 硬件触发输入

目的

通用定时器 TIM2/TIM3、高级定时器 TIM1/TIM8，EPWM 和 EXTI 可用于生成 ADC 触发事件。

Puya Confidential

触发信号

输出（来自定时器）位于 TIMx_TRGO、或 TIMx_CHx、EPWMx_SOCA/B 信号事件上，以及 EXTI Line11、EXTI Line15 事件。

输入（到 ADC）位于 EXT[15:0] 和 JEXT[15:0] 信号上，选择分别由寄存器 ADC_CFGR.EXTSEL[3:0]和 ADC_JSQR.JEXTSEL[3:0]选择

表 9-13 ADC 硬件触发源

ADC trigger selection EXTSEL[4:0] or JEXTSEL[4:0]	ADC triggers signals assignment	
	ADC1/2	
	Regular	Injected
0	TIM1_CH1	TIM1_TRGO
1	TIM1_CH2	TIM1_CH4
2	TIM1_CH3	TIM2_TRGO
3	TIM2_CH2	TIM2_CH1
4	TIM3_TRGO	TIM3_CH4
5	tim1_multi_trg2adc	tim1_multi_trg2adc
6	EXTI11	EXTI15
7	TIM8_TRGO	TIM8_CH4
8	TIM8_TRGO2	TIM1_TRGO2
9	TIM1_TRGO	TIM8_TRGO
10	TIM1_TRGO2	TIM8_TRGO2
11	TIM2_TRGO	TIM3_CH3
12	tim2_multi_trg2adc	TIM3_TRGO
13	tim3_multi_trg2adc	TIM3_CH1
14	tim8_multi_trg2adc	tim2_multi_trg2adc
15	TIM3_CH4	tim3_multi_trg2adc
16	TIM1_CH4	tim8_multi_trg2adc
17	TIM1_CH5	TIM1_CH1
18	LPTIM_OUT	TIM1_CH2
19	TIM8_CH4	LPTIM_OUT
20	EPWM1_SOCA	EPWM1_SOCA
21	EPWM1_SOCB	EPWM1_SOCB
22	EPWM2_SOCA	EPWM2_SOCA
23	EPWM2_SOCB	EPWM2_SOCB
24	EPWM3_SOCA	EPWM3_SOCA
25	EPWM3_SOCB	EPWM3_SOCB
26	EPWM4_SOCA	EPWM4_SOCA
27	EPWM4_SOCB	EPWM4_SOCB
28	EPWM5_SOCA	EPWM5_SOCA
29	EPWM5_SOCB	EPWM5_SOCB
30	EPWM6_SOCA	EPWM6_SOCA
31	EPWM6_SOCB	EPWM6_SOCB

9.2.10 DAC 硬件触发输入

目的

通用定时器 TIM2/TIM3、高级定时器 TIM1/TIM8 、EPWM 以及 EXTI 可用于生成 DAC 外部触发事件。

触发信号

输出（来自定时器）位于 TIMx_TRGO 事件上，以及 EXTI Line9 事件。

输入（到 DAC）位于 TSEL [3:0]信号上，选择分别由寄存器 DAC_CR1.TSEL[3:0]选择。

Puya Confi

表 9-14 DAC 触发源

DAC trigger selection (TSELx[3:0], STRSTTRIG SELx[3:0])	DAC triggers signals assignment	
	DAC	
	update/reset	inc
0	SW	SW
1	TIM8_TRGO	TIM8_TRGO
2	TIM1_TRGO	TIM1_TRGO
3	TIM1_TRGO2	TIM1_TRGO2
4	TIM2_TRGO	TIM2_TRGO
5	TIM8_TRGO2	TIM8_TRGO2
6	EXTI9	EXTI10
7	EPWM1_SYNCPER	EPWM1_SYNCPER
8	TIM3_TRGO	TIM3_TRGO
9	EPWM2_SYNCPER	EPWM2_SYNCPER
10	EPWM3_SYNCPER	EPWM3_SYNCPER
11	EPWM4_SYNCPER	EPWM4_SYNCPER
12	COMP1_OUT	EPWM5_SYNCPER
13	COMP2_OUT	EPWM6_SYNCPER
14	COMP3_OUT	COMP1_OUT
15	COMP4_OUT	COMP2_OUT

Puya Confidential

10. 系统配置控制器（SYSCFG）

10.1 SYSCFG 主要特性

SYSCFG 模块主要完成如下功能：

- I²C IO 噪声滤波器控制
- I²C Fm+模式控制
- 所有 IO 噪声滤波器控制
- EXTI IO 选择
- 根据不同 boot 模式，映射初始程序区
- DMA 外设通道选择
- 定时器的刹车输入控制

10.2 SYSCFG 寄存器

10.2.1 SYSCFG 配置寄存器 1（SYSCFG_CFGR1）

偏移地址：0x00

复位值：0x0000 000x（x 取决于 BOOT0 引脚以及选项字节里启动模式配置）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C2_FLTEN	I2C1_FLTEN	I2C2_FMP	I2C1_FMP	MEM_MODE[1:0]	
										RW	RW	RW	RW	RW	

Bit	Name	RW	Reset Value	Function
31:6	Reserved	-	-	保留
5	I2C2_FLTEN	RW	0	I2C2 滤波使能 0：I2C2 滤波关闭 1：I2C2 滤波开启
4	I2C1_FLTEN	RW	0	I2C1 滤波使能 0：I2C1 滤波关闭 1：I2C1 滤波开启
3	I2C2_FMP	RW	0	I2C2 Fm+模式驱动能力激活 0：I2C2 被 AF 通道所选择的引脚的 Fm+驱动模式不使能 1：I2C2 被 AF 通道所选择的引脚的 Fm+驱动模式使能 适用 IO: PA5、PA6、PA7； PB1、PB2、PB8、PB9、PB10、PB11,PB14,PB15, PC2、PC3、PC4、PC10、PC11、PC12； PD0、PD1、PD2、PD6、PD7、PD8、PD9
2	I2C1_FMP	RW	0	I2C1 Fm+模式驱动能力激活 0：I2C1 被 AF 通道所选择的引脚的 Fm+驱动模式不使能

Puya Confidential

				1: I2C1 被 AF 通道所选择的引脚的 Fm+驱动模式使能 适用 IO: PA1、PA2、PA3、PA4、PA8、PA9、PA10、PA13、PA14、PA15； PB0、PB3、PB4、PB5、PB6、PB7、PB8、PB9、PB13； PC0、PC1、PC5、PC6、PC7、PC8、PC9、PC13、PC14、PC15‘ PD2、PD4、PD5、PD10
1:0	MEM_MODE [1:0]	RW	X	启动模式选择位 控制存储器的 0x0000 0000 地址的映射。在复位后，这些位采用实际实际启动模式配置值。 x0：Main flash，映射到 0x0000 0000 01：System flash，映射到 0x0000 0000 11：SRAM，映射到 0x0000 0000

10.2.2 SYSCFG 配置寄存器 2（SYSCFG_CFGR2）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDL	Res.	CLL
													RS		RS

Bit	Name	RW	Reset Value	Function
31:3	Reserved	-	-	保留
2	PVDL	RS	0	PVD 锁定使能位 软件置位，系统复位清零。它可以被用作使能和锁定 PVD 连接给 TIM1/8/2/3 的刹车输入，也锁定 PWR_CR 寄存器的 PVDE。 软件写 0 到该位不会改变该位的值 0：PVD 中断不与 TIM1/8/2/3 的刹车输入连接。PVDE 位可以被软件写入。 1：PVD 中断与 TIM1/8/2/3 的刹车输入连接。PVDE 位只读。
1	Reserved	-	-	保留
0	CLL	RS	0	Cortex-M4F LOCKUP 使能位 软件置位，系统复位清零。它可以使能和锁定 Cortex-M4F 的 LOCKUP（HardFault）输出给 TIM1/8/2/3 的刹车输入。 软件写 0 到该位不会改变该位的值 0：Cortex-M4F 的 LOCKUP 输出不与 TIM1/8/2/3 的刹车输入连接； 1：Cortex-M4F 的 LOCKUP 输出与 TIM1/8/2/3 的刹车输入连接；

10.2.3 SYSCFG 配置寄存器 3（SYSCFG_CFGR3）

偏移地址：0x8

复位值：0x7F7F 7F7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DMA1_MAP4[6:0]							Res.	DMA1_MAP3[6:0]						

Puya Confidential

	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	DMA1_MAP2[6:0]							Res.	DMA1_MAP1[6:0]						
	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31	Reserved	-	-	保留
30:24	DMA_MAP4[6:0]	RW	7'h7F	DMA1 通道 4 请求映射。详见 DMA 章节
23	Reserved	-	-	保留
22:16	DMA_MAP3[6:0]	RW	7'h7F	DMA1 通道 3 请求映射。详见 DMA 章节
15	Reserved	-	-	保留
14:8	DMA_MAP2[6:0]	RW	7'h7F	DMA1 通道 2 请求映射。详见 DMA 章节
7	Reserved	-	-	保留
6:0	DMA_MAP1[6:0]	RW	7'h7F	DMA1 通道 1 请求映射。详见 DMA 章节

10.2.4 SYSCFG 配置寄存器 4（SYSCFG_CFGR4）

偏移地址：0xc

复位值：0x0000 007F

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1_MAP5[6:0]						
									RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:7	Reserved	-	-	保留
6:0	DMA_MAP5[6:0]	RW	7'h7F	DMA 通道 5 请求映射。详见 DMA 章节

10.2.5 SYSCFG 外部中断配置寄存器 1（SYS_EXTICR1）

偏移地址：0x14

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	EXTI3[1:0]		Res.	Res.	EXTI2[1:0]		Res.	Res.	EXTI1[1:0]		Res.	Res.	EXTI0[1:0]	
		RW	RW			RW	RW			RW	RW			RW	RW

Bit	Name	RW	Reset Value	Function
31:14	Reserved	-	-	保留
13:12	EXTI3[1:0]	RW	2'b0	EXTI3 配置 这些位可由软件读写，用于选择 EXTI3 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[3]引脚 01：PB[3]引脚 10：PC[3]引脚

Puya Confidential

				11：PD[3]引脚
11:10	Reserved	-	-	保留
9:8	EXTI2[1:0]	RW	2’b0	EXTI2 配置 这些位可由软件读写，用于选择 EXTI2 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[2]引脚 01：PB[2]引脚 10：PC[2]引脚 11：PD[2]引脚
7:6	Reserved	-	-	保留
5:4	EXTI1[1:0]	RW	2’b0	EXTI1 配置 这些位可由软件读写，用于选择 EXTI1 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[1]引脚 01：PB[1]引脚 10：PC[1]引脚 11：PD[1]引脚
3:2	Reserved	-	-	保留
1:0	EXTI0[1:0]	RW	2’b0	EXTI0 配置 这些位可由软件读写，用于选择 EXTI0 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[0]引脚 01：PB[0]引脚 10：PC[0]引脚 11：PD[0]引脚

10.2.6 SYSCFG 外部中断配置寄存器 2（SYS_EXTICR2）

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	EXTI7[1:0]		Res.	Res.	EXTI6[1:0]		Res.	Res.	EXTI5[1:0]		Res.	Res.	EXTI4[1:0]	
		RW	RW			RW	RW			RW	RW			RW	RW

Bit	Name	RW	Reset Value	Function
31:14	Reserved	-	-	保留
13:12	EXTI7[1:0]	RW	2’b0	EXTI7 配置 这些位可由软件读写，用于选择 EXTI7 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[7]引脚

Nova Confidential

				01：PB[7]引脚 10：PC[7]引脚 11：PD[7]引脚
11:10	Reserved	-	-	保留
9:8	EXTI6[1:0]	RW	2’b0	EXTI6 配置 这些位可由软件读写，用于选择 EXTI6 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[6]引脚 01：PB[6]引脚 10：PC[6]引脚 11：PD[6]引脚
7:6	Reserved	-	-	保留
5:4	EXTI5[1:0]	RW	2’b0	EXTI5 配置 这些位可由软件读写，用于选择 EXTI5 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[5]引脚 01：PB[5]引脚 10：PC[5]引脚 11：PD[5]引脚
3:2	Reserved	-	-	保留
1:0	EXTI4[1:0]	RW	2’b0	EXTI4 配置 这些位可由软件读写，用于选择 EXTI4 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[4]引脚 01：PB[4]引脚 10：PC[4]引脚 11：PD[4]引脚

10.2.7 SYSCFG 外部中断配置寄存器 3（SYS_EXTICR3）

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	EXTI11[1:0]		Res.	Res.	EXTI10[1:0]		Res.	Res.	EXTI9[1:0]		Res.	Res.	EXTI8[1:0]	
		RW	RW			RW	RW			RW	RW			RW	RW

Bit	Name	RW	Reset Value	Function
31:14	Reserved	-	-	保留

Puya Confidential

13:12	EXTI11[1:0]	RW	2’b0	EXTI11 配置 这些位可由软件读写，用于选择 EXTI11 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[11]引脚 01：PB[11]引脚 10：PC[11]引脚 11：PD[11]引脚
11:10	Reserved	-	-	保留
9:8	EXTI10[1:0]	RW	2’b0	EXTI10 配置 这些位可由软件读写，用于选择 EXTI10 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[10]引脚 01：PB[10]引脚 10：PC[10]引脚 11：PD[10]引脚
7:6	Reserved	-	-	保留
5:4	EXTI9[1:0]	RW	2’b0	EXTI9 配置 这些位可由软件读写，用于选择 EXTI9 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[9]引脚 01：PB[9]引脚 10：PC[9]引脚 11：PD[9]引脚
3:2	Reserved	-	-	保留
1:0	EXTI8[1:0]	RW	2’b0	EXTI8 配置 这些位可由软件读写，用于选择 EXTI8 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[8]引脚 01：PB[8]引脚 10：PC[8]引脚 11：PD[8]引脚

10.2.8 SYSCFG 外部中断配置寄存器 4（SYS_EXTICR4）

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	EXTI15[1:0]		Res.	Res.	EXTI14[1:0]		Res.	Res.	EXTI13[1:0]		Res.	Res.	EXTI12[1:0]	
		RW	RW			RW	RW			RW	RW			RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:14	Reserved	-	-	保留
13:12	EXTI15[1:0]	RW	2’b0	EXTI15 配置 这些位可由软件读写，用于选择 EXTI15 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[15]引脚 01：PB[15]引脚 10：PC[15]引脚 11：保留
11:10	Reserved	-	-	保留
9:8	EXTI14[1:0]	RW	2’b0	EXTI14 配置 这些位可由软件读写，用于选择 EXTI14 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[14]引脚 01：PB[14]引脚 10：PC[14]引脚 11：保留
7:6	Reserved	-	-	保留
5:4	EXTI13[1:0]	RW	2’b0	EXTI13 配置 这些位可由软件读写，用于选择 EXTI13 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[13]引脚 01：PB[13]引脚 10：PC[13]引脚 11：保留
3:2	Reserved	-	-	保留
1:0	EXTI12[1:0]	RW	2’b0	EXTI12 配置 这些位可由软件读写，用于选择 EXTI12 外部中断的输入源。参考 EXTI 外部中断事件映射。 00：PA[12]引脚 01：PB[12]引脚 10：PC[12]引脚 11：保留

10.2.9 SYSCFG GPIOA 滤波使能寄存器（PA_ENS）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA_ENS[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	PA_ENS[15:0]	RW	16'h0	GPIOA0~15 的噪声滤波使能 0：关闭噪声滤波功能 1：开启噪声滤波功能

10.2.10 SYSCFG GPIOB 滤波使能寄存器（PB_ENS）

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PB_ENS[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	PB_ENS[15:0]	RW	16'h0	GPIOB0~15 的噪声滤波使能 0：关闭噪声滤波功能 1：开启噪声滤波功能

10.2.11 SYSCFG GPIOC 滤波使能寄存器（PC_ENS）

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC_ENS[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	PC_ENS[15:0]	RW	16'h0	GPIOC0~15 的噪声滤波使能 0：关闭噪声滤波功能 1：开启噪声滤波功能

10.2.12 SYSCFG GPIOD 滤波使能寄存器（PD_ENS）

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PD_ENS[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	PD_ENS[11:0]	RW	12'h0	GPIOD0~11 的噪声滤波使能 0：关闭噪声滤波功能 1：开启噪声滤波功能

Puya Confidential

11. DMA 控制器（DMA）

11.1 DMA 简介

直接存储器存取（DMA）用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输。无须 CPU 干预，数据可以通过 DMA 快速地移动，节省了 CPU 的资源，进行其他操作。

DMA 控制器有 5 条 DMA 通道，每条通道负责管理来自 1 个或者多个外设对存储器访问的请求。DMA 控制器包括处理 DMA 请求的仲裁器，用于处理各个 DMA 请求的优先级。

11.2 DMA 主要特性

- 5 个独立可配置的通道
- 每个通道通过配置可以连接任一外设的硬件 DMA 请求，每个通道都同样支持软件触发。
- 在同一个 DMA 模块上，多个请求间的优先权可以通过软件编程设置，优先权设置相等时由硬件决定（通道号越低优先级越高）
- 独立数据源和目标数据区的传输宽度(字节、半字、全字)，模拟打包和拆包的过程。源和目标地址必须按数据传输宽度对齐
- 可编程的源和目标地址，地址可选递增，递减或不变
- 每个通道都有 4 个事件标志(传输完成、块传输完成、半块传输完成，传输错误)，这 4 个事件标志进行“逻辑或”，成为一个单独的中断请求
- 支持存储器和存储器间、外设和存储器、存储器和外设、外设和外设的数据传输
- SRAM、APB 和 AHB 外设均可作为访问的源和目标，FLASH 只能作为源不能作为目标
- 支持单次触发模式和四种循环模式
  - 外设地址保持，存储器地址保持
  - 外设地址重新加载，存储器地址保持
  - 外设地址保持，存储器地址重新加载
  - 外设和存储器地址都重新加载
- 单次模式可编程传输数量 0~65535
- 循环模式支持无限循环和有限循环（1~255）
- 支持单一传输和批量传输
  - 单一传输：搬运 1 次数据回复 1 次 ACK；
  - 批量传输：搬运配置的数据量后回复 1 次 ACK（所有数据搬运结束后释放总线）；
- 支持存储器到储存器模式的两种传输方式
  - 快速模式：获得仲裁后始终占据总线，直到传输完所有数据后释放总线；
  - 轮换模式：传输 1 次数据后释放总线重新仲裁；
- 支持在循环模式下进入块传输完成中断后暂停传输

Puya Confidential

11.3 DMA 功能描述

11.3.1 DMA 顶层结构图

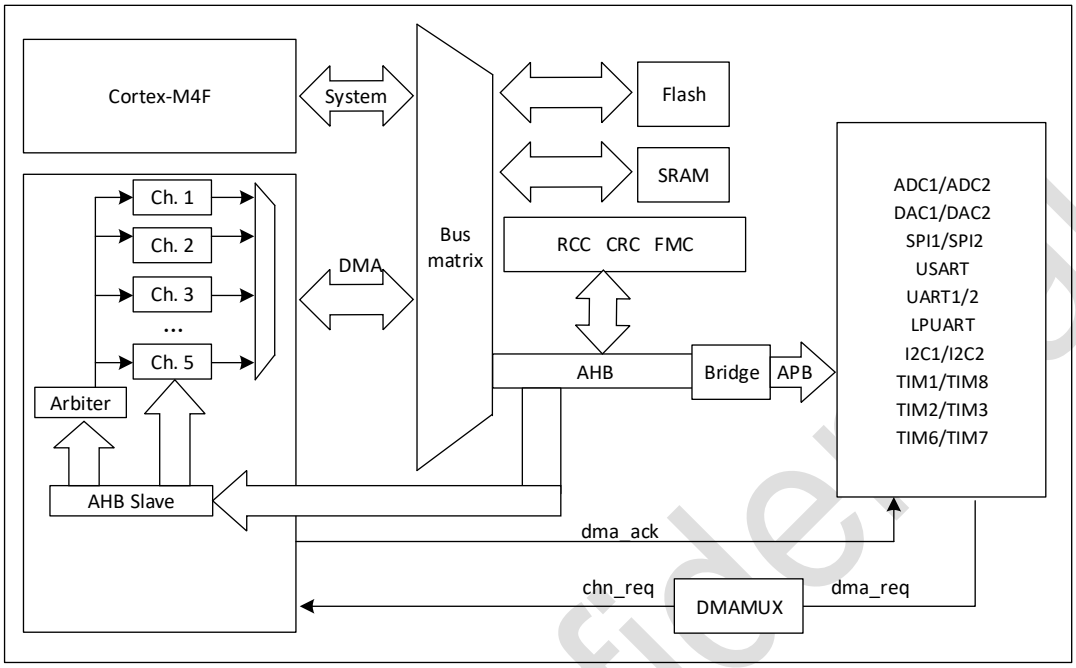


图 11-1 DMA 框图

11.3.2 CPU 和 DMA 总线共享

DMA 控制器和 CPU 共享 AMBA 数据总线，执行直接存储器数据传输。当 CPU 和 DMA 同时访问相同的目标（RAM 或外设）时，DMA 请求会暂停 CPU 访问系统总线达若干个周期，总线仲裁器执行循环调度，以保证 CPU 至少可以得到一半的系统总线带宽。

11.3.3 DMA 传输

在完成一个事件后，外设向 DMA 控制器发送一个请求信号。DMA 控制器根据通道的优先权处理请求。当 DMA 控制器开始访问发出请求的外设时，DMA 控制器在传输结束后发送给它一个应答信号。当从 DMA 控制器得到应答信号时，外设立即释放它的请求。一旦外设释放了这个请求，DMA 控制器撤销应答信号。如果有更多的请求时，外设可以启动下一个传送。

总之，每次 DMA 传送由三个操作组成：

- 从外设数据寄存器或者从当前外设/存储器地址寄存器取数，第一次传输时的开始地址是 DMA_CPARx 或 DMA_CMARx 寄存器指定的外设基地址或存储器单元。
- 存数据到外设寄存器或者当前外设/存储器地址寄存器指示的存储器地址，第一次传输时的开始地址是 DMA_CPARx 或 DMA_CMARx 寄存器指定的外设及地址或存储器单元。
- 执行一次 DMA_CNDTRx 寄存器的递减操作，该寄存器表明未完成的操作数目。

11.3.4 DMA 仲裁器

仲裁器根据通道请求的优先级来启动外设/存储器的访问。

优先权管理分 2 个阶段：

- 软件：每个通道的优先权可以在 DMA_CCRx 寄存器中设置，有 5 个等级：

PUYA Confidential

  - 0
  - 1
  - 2
  - 3
  - 4

注：优先级 4>3>2>1>0。

- 硬件: 如果 2 个请求有相同的软件优先级，则较低编号的通道比较高编号的通道有更高的优先权。比如，通道 2 优先于通道 3。

11.3.5 DMA 通道

每个通道都可以在有固定地址的外设寄存器和存储器地址之间执行 DMA 传输。DMA 的传输数据量是可编程的，最大为 65535。包含要传输的数据数量的寄存器，在每次传输后递减。

可编程数据大小

外设和存储器的传输数据宽度可以通过 DMA_CCRx 寄存器中的 MSIZE 和 PSIZE 位编程。

可编程地址增量方式

通过设置 DMA_CCRx 寄存器中的 PINC 和 MINC 标志位，外设和存储器的指针在每次传输后可以有选择的完成自动增量。当设置位增量模式时，下一个要传输的地址将是前一个地址加上增量值，增量值取决于所选的数据宽度为 1，2，4。

第一个传输的地址是存放在 DMA_CPARx/DMA_CMARx 寄存器中地址。在传输过程中，这些寄存器保持他们的初始值，软件不能改变和读出当前正在传输的地址（它在内部的当前外设/存储器地址寄存器中）。

当通道配置为单块传输时，传输结束后(即传输计数变为 0)将不再产生 DMA 操作。要开始新的 DMA 传输，需要在关闭 DMA 通道的情况下，在 DMA_CNDTRx 寄存器中重新写入传输数目。

在循环模式下，最后一次传输结束时，DMA_CNDTRx 寄存器的内容会自动被重新加载为其初始数值，内部的当前外设/存储器地址寄存器根据 MNORLD 和 PNORLD 的配置保持地址或者重新加载地址。

循环模式

循环模式用于处理循环缓冲区和连续的数据传输（如 ADC 的 scan 模式）。在 DMA_CCRx 寄存器中的 CIRC 位用于开启这一功能。当启动了循环模式，数据传输的数目变为 0 时，将会自动地被恢复配置通道时设置的初值，DMA 操作将会继续进行。

循环模式地址重产生方式有如下 4 种情况：

- 外设地址保持，存储器地址保持
- 外设地址重新加载，存储器地址保持
- 外设地址保持，存储器地址重新加载
- 外设和存储器地址都重新加载

循环模式分为以下两种循环次数：

- 无限循环：通道使能，且 DMA_CCRx.CIRC 为 1，且 DMA_CCCFGRx.NBT 为 0，单块传输数据量为 DMA CNDTRx.NDT；
- 有限循环：通道使能，且 DMA_CCRx.CIRC 为 1，且 DMA_CCCFGRx.NBT 不为 0。
  - DMA_CCCFGRx.NBT 大于 0 时，单块传输数据量为 DMA_CNDTRx.NDT；

Puya Confidential

— DMA_CCCFGRx.NBT 等于 0 时，若 DMA_CCCFGRx.NDTL 等于 0 则传输结束；若 DMA_CCCFGRx.NDTL 大于 0 则继续传输一次 DMA_CCCFGRx.NDTL 个数的数据后传输结束；

存储器到存储器模式

DMA 通道的操作可以在没有外设请求的情况下进行，这种操作就是存储器到存储器模式。

当设置了 DMA_CCRx 寄存器中的 MEM2MEM 位之后，在软件设置了 DMA_CCRx 寄存器中的 EN 位启动 DMA 通道时，DMA 传输将马上开始。当 DMA_CNDTRx 寄存器变为 0 时，DMA 传输结束。

存储器到存储器模式分为以下两种模式

- 快速模式：获得仲裁后始终占据总线，直到传输完所有数据后释放总线，该模式不能与循环模式同时使用；
- 轮换模式：传输 1 次数据后释放总线重新仲裁，该模式下其他通道也有获得仲裁的机会；

传输类型

根据外设的传输需求，支持以下两种数据传输方式

- 单一传输：传输 1 次数据回复 1 次 dma_ack
- 批量传输：传输 NDT 次数据回复 1 次 dma_ack，使用该方式传输数据的通道会在传输完所有数据后再释放总线

块传输完成中断后暂停

在循环模式下，可以设置 DMA_CCRx.BTCSUSP 为 1 使通道在进入到块传输完成中断后暂停传输数据，在清除块传输完成中断后恢复数据传输。

通道配置步骤

按如下步骤配置 DMA 通道：

- 在 DMA_CPARx 寄存器中设置外设寄存器的地址。发生外设数据传输请求时，这个地址将是数据传输的源或目标。
- 在 DMA_CMARx 寄存器中设置数据存储器的地址。发生外设数据传输请求时，传输的数据将从这个地址读出或写入这个地址。
- 在 DMA_CNDTRx 寄存器中设置要传输的数据量。在每个数据传输后，这个数值递减。
- 在 DMA_CCRx 寄存器的 PL 位中设置通道的优先级。
- 在 DMA_CCRx 寄存器中设置数据传输的方向、循环模式、外设和存储器的增量模式、外设和存储器的数据宽度、传输一半产生中断或传输完成产生中断。
- 在 DMA_CNBTRx 寄存器中设置循环传输次数。在有限循环时，每个块传输后，这个数值递减。
- 设置 DMA_CCRx 寄存器的 ENABLE 位，启动该通道。

一旦启动了 DMA 通道，即可响应连到该通道上的外设的 DMA 请求。

当传输一半的数据后，半块传输标志（HBTIF）被置 1，当设置了允许半块传输中断位（HBTIE）时，将产生一个中断请求。单块数据传输结束后，块传输完成标志（BTCIF）被置 1，当设置了允许块传输完成中断位（BTCIE）时，将产生一个中断请求。多块数据全部传输结束后，传输完成标志（TCIF）被置 1，当设置了允许传输完成中断位（TCIE）时，将产生一个中断请求。

11.3.6 可编程数据位宽，数据对齐和大小端

当存储器数据宽度 MSIZE 和外设数据宽度 PSIZE 不同时，DMA 按照下表进行数据对齐：

Puya Confidential

表 11-1 数据宽度和大小端（PINC=MINC=1）

源宽度	目标宽度	传输数目	源：地址/数据	传输操作	目标：地址/数据
8	8	4	0x0/B0 0x1/B1 0x2/B2 0x3/B3	1:在 0x0 读 B0[7:0],在 0x0 写 B0[7:0] 2:在 0x1 读 B1[7:0],在 0x1 写 B1[7:0] 3:在 0x2 读 B2[7:0],在 0x2 写 B2[7:0] 4:在 0x3 读 B3[7:0],在 0x3 写 B3[7:0]	0x0/B0 0x1/B1 0x2/B2 0x3/B3
8	16	4	0x0/B0 0x1/B1 0x2/B2 0x3/B3	1:在 0x0 读 B0[7:0],在 0x0 写 00B0[7:0] 2:在 0x1 读 B1[7:0],在 0x2 写 00B1[7:0] 3:在 0x2 读 B2[7:0],在 0x4 写 00B2[7:0] 4:在 0x3 读 B3[7:0],在 0x6 写 00B3[7:0]	0x0/00B0 0x2/00B1 0x4/00B2 0x6/00B3
8	32	4	0x0/B0 0x1/B1 0x2/B2 0x3/B3	1:在 0x0 读 B0[7:0],在 0x0 写 000000B0[31:0] 2:在 0x1 读 B1[7:0],在 0x4 写 000000B1[31:0] 3:在 0x2 读 B2[7:0],在 0x8 写 000000B2[31:0] 4:在 0x3 读 B3[7:0],在 0xC 写 000000B3[31:0]	0x0/000000B0 0x4/000000B1 0x8/000000B2 0xC/000000B3
16	8	4	0x0/B1B0 0x2/B3B2 0x4/B5B4 0x6/B7B6	1:在 0x0 读 B1B0[15:0],在 0x0 写 B0[7:0] 2:在 0x2 读 B3B2[15:0],在 0x1 写 B2[7:0] 3:在 0x4 读 B5B4[15:0],在 0x2 写 B4[7:0] 4:在 0x6 读 B7B6[15:0],在 0x3 写 B6[7:0]	0x0/B0 0x1/B2 0x2/B4 0x3/B6
16	16	4	0x0/B1B0 0x2/B3B2 0x4/B5B4 0x6/B7B6	1:在 0x0 读 B1B0[15:0],在 0x0 写 B1B0[15:0] 2:在 0x2 读 B3B2[15:0],在 0x2 写 B3B2[15:0] 3:在 0x4 读 B5B4[15:0],在 0x4 写 B5B4[15:0] 4:在 0x6 读 B7B6[15:0],在 0x6 写 B7B6[15:0]	0x0/B1B0 0x2/B3B2 0x4/B5B4 0x6/B7B6
16	32	4	0x0/B1B0 0x2/B3B2 0x4/B5B4 0x6/B7B6	1:在 0x0 读 B1B0[7:0],在 0x0 写 0000B1B0[31:0] 2:在 0x2 读 B3B2[7:0],在 0x4 写 0000B3B2[31:0] 3:在 0x4 读 B5B4[7:0],在 0x8 写 0000B5B4[31:0] 4:在 0x6 读 B7B6[7:0],在 0xC 写 0000B7B6[31:0]	0x0/0000B1B0 0x4/000B3B2 0x8/0000B5B4 0xC/0000B7B6
32	8	4	0x0/B3B2B1B0 0x4/B7B6B5B4 0x8/BBBAB9B8 0xC/BFBEBDBC	1:在 0x0 读 B3B2B1B0 [31:0],在 0x0 写 B0[7:0] 2:在 0x4 读 B7B6B5B4 [31:0],在 0x1 写 B4[7:0] 3:在 0x8 读 BBBAB9B8 [31:0],在 0x2 写 B8[7:0] 4:在 0xc 读 BFBEBDBC [31:0],在 0x3 写 BC[7:0]	0x0/B0 0x1/B4 0x2/B8 0x3/BC
32	16	4	0x0/B3B2B1B0 0x4/B7B6B5B4 0x8/BBBAB9B8 0xC/BFBEBDBC	1:在 0x0 读 B3B2B1B0 [31:0],在 0x0 写 B1B0[7:0] 2:在 0x4 读 B7B6B5B4 [31:0],在 0x2 写 B5B4[7:0] 3:在 0x8 读 BBBAB9B8 [31:0],在 0x4 写 B9B8[7:0] 4:在 0xc 读 BFBEBDBC [31:0],在 0x6 写 BDBC[7:0]	0x0/B1B0 0x2/B5B4 0x4/B9B8 0x6/BDBC
32	32	4	0x0/B3B2B1B0 0x4/B7B6B5B4 0x8/BBBAB9B8 0xC/BFBEBDBC	1:在 0x0 读 B3B2B1B0 [31:0],在 0x0 写 B3B2B1B0[7:0] 2:在 0x4 读 B7B6B5B4 [31:0],在 0x2 写 B7B6B5B4[7:0] 3:在 0x8 读 BBBAB9B8 [31:0],在 0x4 写 BBBAB9B8[7:0] 4:在 0xc 读 BFBEBDBC [31:0],在 0x6 写 BFBEBDBC[7:0]	0x0/B3B2B1B0 0x4/B7B6B5B4 0x8/BBBAB9B8 0xC/BFBEBDBC

Puya Confidential

不支持字节/半字访问

如果 DMA 以字节或半字写入不支持字节或半字写操作的 AHB 设备时（即 HSIZE 不适于该模块），不会发生错误，DMA 将按照下面两个例子写入 32 位 HWDATA 数据：

- 当 HSIZE=半字时，写入半字'0xABCD'，DMA 将设置 HWDATA 总线为'0xABCDABCD'。
- 当 HSIZE=字节时，写入字节'0xAB'，DMA 将设置 HWDATA 总线为'0xABABABAB'。

假定 AHB/APB 桥是一个 AHB 的 32 位从机，它不处理 HSIZE 参数，它将按照下述方式把任何 AHB 上的字节或半字按 32 位传送到 APB 上：

- 一个 AHB 上对地址 0x0（或 0x1、0x2 或 0x3）的写字节数据'0xB0'操作，将转换到 APB 上对地址 0x0 的写字数据'0xB0B0B0B0'操作。
- 一个 AHB 上对地址 0x0（或 0x2）的写半字数据'0xB1B0'操作，将转换到 APB 上对地址 0x0 的写字数据'0xB1B0B1B0'操作。

11.3.7 错误处理

读写一个保留的地址区域，将会产生 DMA 传输错误。在 DMA 读写操作时，发生 DMA 传输错误时，硬件会自动清除发生错误的通道所对应的通道配置寄存器（DMA_CCRx）的 EN 位，该通道操作被停止。此时，在 DMA_IFR 寄存器中对应该通道的传输错误中断标志位（TEIF）将被置位，如果在 DMA_CCRx 寄存器中设置了传输错误中断允许位，则将产生中断。

11.3.8 DMA 外设请求映射

DMA 外设请求映射到 DMA（5 通道）的各个通道，由 SYSCFG 的 DMAx_MAP 寄存器控制，每个外设请求通过配置可以映射到 5 个通道中的任意一个。

序号与外设请求的关系如下表所示：

表 11-2 DMA 外设请求映射

请求 MUX 输入序号	源	请求 MUX 输入序号	源	请求 MUX 输入序号	源
0	ADC1	16	I2C1_RX	32	TIM2_TRIG
1	ADC2	17	I2C1_TX	33	TIM3_CH1
2	DAC1	18	I2C2_RX	34	TIM3_CH2
3	DAC2	19	I2C2_TX	35	TIM3_CH3
4	SPI1_RX	20	TIM1_CH1	36	TIM3_CH4
5	SPI1_TX	21	TIM1_CH2	37	TIM3_UP
6	SPI2_RX	22	TIM1_CH3	38	TIM3_TRIG
7	SPI2_TX	23	TIM1_CH4	39	TIM6_UP
8	USART1_RX	24	TIM1_UP	40	TIM7_UP
9	USART1_TX	25	TIM1_TRIG	41	TIM8_CH1
10	UART1_RX	26	TIM1_COM	42	TIM8_CH2
11	UART1_TX	27	TIM2_CH1	43	TIM8_CH3
12	UART2_RX	28	TIM2_CH2	44	TIM8_CH4
13	UART2_TX	29	TIM2_CH3	45	TIM8_UP
14	LPUART1_RX	30	TIM2_CH4	46	TIM8_TRIG
15	LPUART1_TX	31	TIM2_UP	47	TIM8_COM

Puya Confidential

11.4 DMA 中断

每个 DMA 通道都可以在 DMA 块传输过半、块传输完成、传输完成和传输错误时产生中断。为应用的灵活性考虑，通过设置寄存器的不同位来打开这些中断。

表 11-3 DMA 中断请求

中断事件	事件标志位	使能控制位
块传输过半	HBTIF	HBTIE
块传输完成	BTCIF	BTCIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

注：当 DMA_CNDTRx 寄存器为 1 时，不会置 HBTIFx 位

11.5 DMA 寄存器

11.5.1 DMA 中断状态寄存器（DMA_ISR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEIF5	HBTIF5	BTCIF5	TCIF5
												R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HBTIF4	BTCIF4	TCIF4	TEIF3	HBTIF3	BTCIF3	TCIF3	TEIF2	HBTIF2	BTCIF2	TCIF2	TEIF1	HBTIF1	BTCIF1	TCIF1
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:20	Reserved	-	-	保留
19	TEIF5	R	0	通道 5 传输错误（TE）标志。 硬件置位，软件写 DMA_CTEIF5=1 清零。 0：无 TE 事件； 1：发生 TE 事件；
18	HBTIF5	R	0	通道 5 半块传输完成（HBT）标志。 硬件置位，软件写 DMA_CHBTIF5=1 清零。 0：无 HBT 事件； 1：发生 HBT 事件；
17	BTCIF5	R	0	通道 5 块传输完成（BTC）标志。 硬件置位，软件写 DMA_CBTCIF5=1 清零。 0：无 BTC 事件； 1：发生 BTC 事件；
16	TCIF5	R	0	通道 5 所有块传输完成（TC）中断标志。 硬件置位，软件写 DMA_CTCIF5=1 清零。 0：无 TC 事件； 1：发生 TC 事件；

Puya Confidential

15	TEIF4	R	0	通道 4 传输错误（TE）标志。 硬件置位，软件写 DMA_TEIF4=1 清零。 0：无 TE 事件； 1：发生 TE 事件；
14	HBTIF4	R	0	通道 4 半块传输完成（HBT）标志。 硬件置位，软件写 DMA_CHBTIF4=1 清零。 0：无 HBT 事件； 1：发生 HBT 事件；
13	BTCIF4	R	0	通道 4 块传输完成（BTC）标志。 硬件置位，软件写 DMA_CBTCIF4=1 清零。 0：无 BTC 事件； 1：发生 BTC 事件；
12	TCIF4	R	0	通道 4 所有块传输完成（TC）中断标志。 硬件置位，软件写 DMA_CTCIF4=1 清零。 0：无 TC 事件； 1：发生 TC 事件；
11	TEIF3	R	0	通道 3 传输错误（TE）标志。 硬件置位，软件写 DMA_CTEIF3=1 清零。 0：无 TE 事件； 1：发生 TE 事件；
10	HBTIF3	R	0	通道 3 半块传输完成（HBT）标志。 硬件置位，软件写 DMA_CHBTIF3=1 清零。 0：无 HBT 事件； 1：发生 HBT 事件；
9	BTCIF3	R	0	通道 3 块传输完成（BTC）标志。 硬件置位，软件写 DMA_CBTCIF3=1 清零。 0：无 BTC 事件； 1：发生 BTC 事件；
8	TCIF3	R	0	通道 3 所有块传输完成（TC）中断标志。 硬件置位，软件写 DMA_CTCIF3=1 清零。 0：无 TC 事件； 1：发生 TC 事件；
7	TEIF2	R	0	通道 2 传输错误（TE）标志。 硬件置位，软件写 DMA_CTEIF2=1 清零。 0：无 TE 事件； 1：发生 TE 事件；
6	HBTIF2	R	0	通道 2 block 半传输完成（HBT）标志。 硬件置位，软件写 DMA_CHBTIF2=1 清零。 0：无 HBT 事件； 1：发生 HBT 事件；
5	BTCIF2	R	0	通道 2 块传输完成（BTC）标志。 硬件置位，软件写 DMA_CBTCIF2=1 清零。 0：无 BTC 事件； 1：发生 BTC 事件；
4	TCIF2	R	0	通道 2 所有块传输完成（TC）中断标志。

Puya Confidential

				硬件置位，软件写 DMA_CTCIF2=1 清零。 0：无 TC 事件； 1：发生 TC 事件；
3	TEIF1	R	0	通道 1 传输错误（TE）标志。 硬件置位，软件写 DMA_CTEIF1=1 清零。 0：无 TE 事件； 1：发生 TE 事件；
2	HBTIF1	R	0	通道 1 半块传输完成（HBT）标志。 硬件置位，软件写 DMA_CHBTIF1=1 清零。 0：无 HBT 事件； 1：发生 HBT 事件；
1	BTCIF1	R	0	通道 1 块传输完成（BTC）标志。 硬件置位，软件写 DMA_CBTCIF1=1 清零。 0：无 BTC 事件； 1：发生 BTC 事件；
0	TCIF1	R	0	通道 1 所有块传输完成（TC）中断标志。 硬件置位，软件写 DMA_CTCIF1=1 清零。 0：无 TC 事件； 1：发生 TC 事件；

11.5.2 DMA 中断标志清零寄存器（DMA_IFCR)

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTEIF5	CHBTIF5	CBTCIF5	CTCIF5
												W	W	W	W
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTEIF4	CHBTIF4	CBTCIF4	CTCIF4	CTEIF3	CHBTIF3	CBTCIF3	CTCIF3	CTEIF2	CHBTIF2	CBTCIF2	CTCIF2	CTEIF1	CHBTIF1	CBTCIF1	CTCIF1
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:20	Reserved	-	-	保留
19	CTEIF5	W	0	通道 5 传输错误标志清零。 0：无效果； 1：清零 TEIF5；
18	CHBTIF5	W	0	通道 5 半块传输标志清零。 0：无效果； 1：清零 HBTIF5；
17	CBTCIF5	W	0	通道 5 块传输完成标志清零。 0：无效果； 1：清零 BTCIF5；
16	CTCIF5	W	0	通道 5 所有块传输完成标志清零。 0：无效果； 1：清零 TCIF5；
15	CTEIF4	W	0	通道 4 传输错误标志清零。 0：无效果；

Puya Confidential

				1：清零 TEIF4；
14	CHBTIF4	W	0	通道 4 半块传输标志清零。 0：无效果； 1：清零 HBTIF4；
13	CBTCIF4	W	0	通道 4 块传输完成标志清零。 0：无效果； 1：清零 BTCIF4；
12	CTCIF4	W	0	通道 4 所有块传输完成标志清零。 0：无效果； 1：清零 TCIF4；
11	CTEIF3	W	0	通道 3 传输错误标志清零。 0：无效果； 1：清零 TEIF3；
10	CHBTIF3	W	0	通道 3 半块传输标志清零。 0：无效果； 1：清零 HBTIF3；
9	CBTCIF3	W	0	通道 3 块传输完成标志清零。 0：无效果； 1：清零 BTCIF3；
8	CTCIF3	W	0	通道 3 所有块传输完成标志清零。 0：无效果； 1：清零 TCIF3；
7	CTEIF2	W	0	通道 2 传输错误标志清零。 0：无效果； 1：清零 TEIF2；
6	CHBTIF2	W	0	通道 2 半块传输标志清零。 0：无效果； 1：清零 HBTIF2；
5	CBTCIF2	W	0	通道 2 块传输完成标志清零。 0：无效果； 1：清零 BTCIF2；
4	CTCIF2	W	0	通道 2 所有块传输完成标志清零。 0：无效果； 1：清零 TCIF2；
3	CTEIF1	W	0	通道 1 传输错误标志清零。 0：无效果； 1：清零 TEIF1；
2	CHBTIF1	W	0	通道 1 半块传输标志清零。 0：无效果； 1：清零 HBTIF1；
1	CBTCIF1	W	0	通道 1 块传输完成标志清零。 0：无效果； 1：清零 BTCIF1；
0	CTCIF1	W	0	通道 1 所有块传输完成标志清零。 0：无效果；

Puya Confidential

				1：清零 TCIF1；

11.5.3 DMA 通道 x 控制寄存器（DMA_CCRx）(x=1~5)

偏移地址：0x08+(x-1)*0x14(x 为通道编号)

复位值：0x000y 0000(y 为通道编号-1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	M2MARB	BTCSUSP	MNORLD	PNORLD	MEM2MEM	PL[2:0]		
								RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSIZE[1:0]		PSIZE[1:0]		MINC[1:0]		PINC[1:0]		TRANST	CIRC	DIR	TEIE	HBTIE	BTCIE	TCIE	EN
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	-	保留
23	M2MARB	RW	0	通道 x 存储器到存储器模式下仲裁方式（mem2mem=1 时有效） 0：传输完 1 次数据后不释放总线；（不支持循环模式） 1：传输完 1 次数据后释放总线； EN=0 时软件可读可写； EN=1 时软件只可读；
22	BTCSUSP	RW	0	通道 x 循环模式下进入 BTC 中断后处理方式 0：不暂停传输； 1：暂停传输； 该位配置为 1 时，进入 BTC 中断后软件清除 BTC 中断后恢复传输； EN=0 时软件可读可写； EN=1 时软件只可读；
21	MNORLD	RW	0	通道 x 的存储器地址循环模式（CICR=1 时有效） 0：存储器地址重新加载使能； 1：存储器地址重新加载禁止； EN=0 时软件可读可写； EN=1 时软件只可读；
20	PNORLD	RW	0	通道 x 的外设地址循环模式（CICR=1 时有效） 0：外设地址重新加载使能； 1：外设地址重新加载禁止； EN=0 时软件可读可写； EN=1 时软件只可读；
19	MEM2MEM	RW	0	通道 x 存储器到存储器模式。 0：禁止； 1：存储器到存储器模式使能； EN=0 时软件可读可写； EN=1 时软件只可读；

Puya Confidential

18:16	PL	RW	3'hy	通道 x 优先级配置。 000：通道优先级为 0； 001：通道优先级为 1； 010：通道优先级为 2； 011：通道优先级为 3； 100：通道优先级为 4； 其他：保留 EN=0 时软件可读可写； EN=1 时软件只可读；
15:14	MSIZE	RW	2'h0	通道 x 存储器数据宽度。 00：8 位； 01：16 位； 10：32 位； 11：保留； EN=0 时软件可读可写； EN=1 时软件只可读；
13:12	PSIZE	RW	2'h0	通道 x 外设数据宽度。 00：8 位； 01：16 位； 10：32 位； 11：保留； EN=0 时软件可读可写； EN=1 时软件只可读；
11:10	MINC	RW	2'h0	通道 x 存储器地址增量模式。 00：不变； 01：递增； 10：递减； 11：保留； EN=0 时软件可读可写； EN=1 时软件只可读；
9:8	PINC	RW	2'h0	通道 x 外设地址增量模式。 00：不变； 01：递增； 10：递减； 11：保留； EN=0 时软件可读可写； EN=1 时软件只可读；
7	TRANST	RW	0	通道 x 的数据传输方式（mem2mem=0 时有效） 0：单一传输； 1：批量传输； EN=0 时软件可读可写； EN=1 时软件只可读；
6	CIRC	RW	0	循环模式 0：禁止； 1：使能；

Puya Confidential

				EN=0 时软件可读可写； EN=1 时软件只可读；
5	DIR	RW	0	通道 x 数据传输方向。 0：从外设读； 1：从存储器读； EN=0 时软件可读可写； EN=1 时软件只可读；
4	TEIE	RW	0	通道 x 传输错误中断（TE）使能。 0：禁止； 1：TE 中断使能；
3	HBTIE	RW	0	通道 x block 半传输中断（HBT）使能。 0：禁止； 1：HBT 中断使能；
2	BTCIE	RW	0	通道 x block 传输完成中断（BTC）使能。 0：禁止； 1：BTC 中断使能；
1	TCIE	RW	0	通道 x 所有 block 传输完成中断（TC）使能。 0：禁止； 1：TC 中断使能；
0	EN	RW	0	通道 x 使能. 0：禁止； 1：通道 x 使能； 当所有 block 传输完成或者出现传输错误时硬件清零； 软件可读可写； 注 1：在无效配置，如配置 NDT=0，但配置 EN=1 时，硬件会清零 EN，以保证后续传输正常； 注 2：配置 DMA 寄存器遵循先配置后控制的原则，即在最后配置 EN=1；

11.5.4 DMA 通道 x 数据传输个数寄存器（DMA_CNDTRx）(x=1~5)

偏移地址：0x0C+(x-1)*0x14(x 为通道编号)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	NDT	RW	16'h0	通道 x 数据传输数量。 数据传输数量为 0~65535。该寄存器只在通道不工作（DMA_CCRx.EN=0）时写入。通道使能后

Puya Confidential

				该寄存器为只读，表明剩余传输字节数。该寄存器值在每次 DMA 传输后递减。 数据传输结束后，寄存器的内容或者变为 0，或者当该通道配置为循环模式时，寄存器的内容将被自动重新加载为之前配置时的数值。 当该寄存器值为 0 时，即使 DMA 通道开始，都不会传输数据。 EN=0 时软件可读可写； EN=1 时软件只可读；

11.5.5 DMA 通道 x 外设地址寄存器（DMA_CPARx）(x=1~5)

偏移地址：0x10+(x-1)*0x14(x 为通道编号)

复位值：0x0000 0000

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

Bit	Name	RW	Reset Value	Function
31:0	PA	RW	32’h0	通道 x 外设地址。 通道 x 外设数据寄存器的基址，作为数据传输的源或目标。 当 PSIZE=2’b01，不使用 PA[0]位。操作自动与半字地址对齐。 当 PSIZE=2’b10，不使用 PA[1:0]位。操作自动与字地址对齐。 EN=0 时软件可读可写； EN=1 时软件只可读；

11.5.6 DMA 通道 x 存储地址寄存器（DMA_CMARx）(x=1~5)

偏移地址：0x14+(x-1)*0x14(x 为通道编号)

复位值：0x0000 0000

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

Bit	Name	RW	Reset Value	Function
31:0	MA	RW	32’h0	通道 x 存储器地址。 通道 x 存储器地址，作为数据传输的源或目标。 当 MSIZE=2’b01，不使用 MA[0]位。操作自动与半字地址对齐。 当 MSIZE=2’b10，不使用 MA[1:0]位。操作自动与字地址对齐。 EN=0 时软件可读可写； EN=1 时软件只可读；

Puya Confidential

11.5.7 DMA 通道 x 块循环传输配置寄存器（DMA_CCCFGRx）(x=1~5)

偏移地址：0x18+(x-1)*0x14(x 为通道编号)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDTL[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBT[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	NDTL	RW	16'h0	有限循环模式下通道 x 最后一次块数据传输数量。 数据传输数量为 0~65535。 该寄存器只在通道不工作（DMA CCRx.EN=0）时写入。通道使能后该寄存器为只读，表明循环模式最后一次剩余传输数量。 该寄存器值在循环模式最后一个块传输时随每笔传输递减。 EN=0 时软件可读可写; EN=1 时软件只可读;
15:8	Reserved	-	-	保留
7:0	NBT	RW	8'h0	通道 x 块循环传输数量。 在循环模式下，该寄存器初始值为 0 时块循环传输数量为无穷，该寄存器初始值大于 0 时块循环传输数量为该寄存器值（1~255）。该寄存器只在通道不工作（DMA_CCRx.EN=0）时写入。通道使能后该寄存器为只读，表明剩余传输块数。 该寄存器值在每个 block 传输完成后递减。 EN=0 时软件可读可写; EN=1 时软件只可读; 注：有限循环传输数据个数=NDT*NBT+NDTL

Puya Confidential

12. 中断和事件

12.1 嵌套向量中断控制器（NVIC）

嵌套向量中断控制器（NVIC）和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。

嵌套向量中断控制器管理着包括内核异常等中断。更多关于异常和 NVIC 编程的说明请参考 Cortex®-M4F 编程手册。

12.1.1 NVIC 主要特性

- 63 个可屏蔽中断通道(不包含 16 个 Cortex®-M4F 的中断线)
- 8 个可编程的优先等级(使用了 3 位中断优先级)
- 低延迟的异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

12.1.2 系统滴答（SysTick）校准值寄存器

系统嘀嗒校准值固定为 9000，当系统嘀嗒时钟设定为 9 MHz，产生 1ms 时间基准。

12.1.3 中断和异常向量表

向量编号	优先级	优先级类型	名称	地址	说明
-	-	-	-	0x0000_0000	保留
-	-3	固定	Reset	0x0000_0004	复位
-	-2	NMI	RCC HSE 时钟安全（HSE CSS）	0x0000_0008	不可屏蔽中断
-	-1	固定	硬件失效(HardFault)	0x0000_000C	所有类型的失效
-	0	可编程	存储管理(MemManage)	0x0000_0010	存储器管理
	1	可编程	总线错误(BusFault)	0x0000_0014	预取指失败，存储器访问失败
-	2	可编程	错误应用(UsageFault)	0x0000_0018	未定义的指令或非法状态
-	-	-	-	0x0000_001C	保留
-	-	-	-	0x0000_0020	保留
-	-	-	-	0x0000_0024	保留
-	-	-	-	0x0000_0028	保留
-	3	可编程	SVCall	0x0000_002C	通过 SWI 指令的系统服务调用
-	4	可编程	调试监控(DebugMonitor)	0x0000_0030	调试监控器
-	-	-	-	0x0000_0034	保留
-	5	可编程	PendSV	0x0000_0038	可挂起的系统服务
-	6	可编程	SysTick	0x0000_003C	系统嘀嗒定时器
0	7	可编程	WWDG	0x0000_0040	窗口看门狗定时器中断

Confidential

1	8	可编程	PVD	0x0000_0044	连到 EXTI 的电源电压检测(PVD)中断
2	9	可编程	TAMPER	0x0000_0048	侵入检测中断
3	10	可编程	RTC	0x0000_004C	实时时钟(RTC)全局中断
4	11	可编程	FMC	0x0000_0050	闪存 FMC 全局中断
5	12	可编程	RCC	0x0000_0054	复位和时钟控制(RCC)中断
6	13	可编程	EXTI0	0x0000_0058	EXTI 线 0 中断
7	14	可编程	EXTI1	0x0000_005C	EXTI 线 1 中断
8	15	可编程	EXTI2	0x0000_0060	EXTI 线 2 中断
9	16	可编程	EXTI3	0x0000_0064	EXTI 线 3 中断
10	17	可编程	EXTI4	0x0000_0068	EXTI 线 4 中断
11	18	可编程	EXTI9_5	0x0000_006C	EXTI 线[9:5]中断
12	19	可编程	EXTI15_10	0x0000_0070	EXTI 线[15:10]中断
13	20	可编程	DMA1 通道 1	0x0000_0074	DMA1 通道 1 全局中断
14	21	可编程	DMA1 通道 2	0x0000_0078	DMA1 通道 2 全局中断
15	22	可编程	DMA1 通道 3	0x0000_007C	DMA1 通道 3 全局中断
16	23	可编程	DMA1 通道 4	0x0000_0080	DMA1 通道 4 全局中断
17	24	可编程	DMA1 通道 5	0x0000_0084	DMA1 通道 5 全局中断
18	25	可编程	ADC1	0x0000_0088	ADC1 全局中断
19	26	可编程	ADC2	0x0000_008C	ADC2 全局中断
20	27	可编程	DAC	0x0000_0090	DAC 全局中断
21	28	可编程	TIM1_BRK	0x0000_0094	TIMER1 中止中断
22	29	可编程	TIM1_UP	0x0000_0098	TIMER1 更新中断
23	30	可编程	TIM1_TRG_COM	0x0000_009C	TIMER1 触发和通信中断
24	31	可编程	TIM1_CC	0x0000_00A0	TIMER1 捕获比较中断
25	32	可编程	TIM2	0x0000_00A4	TIMER2 全局中断
26	33	可编程	TIM3	0x0000_00A8	TIMER3 全局中断
27	34	可编程	TIM6	0x0000_00AC	TIM6 全局中断
28	35	可编程	TIM7	0x0000_00B0	TIM7 全局中断
29	36	可编程	TIM8_BRK	0x0000_00B4	TIMER8 中止中断
30	37	可编程	TIM8_UP	0x0000_00B8	TIMER8 更新中断
31	38	可编程	TIM8_TRG_COM	0x0000_00BC	TIMER8 触发和通信中断
32	39	可编程	TIM8_CC	0x0000_00C0	TIMER8 捕获比较中断
33	40	可编程	LPTIM	0x0000_00C4	LPTIM 全局中断
34	41	可编程	SPI1	0x0000_00C8	SPI1 全局中断
35	42	可编程	SPI2	0x0000_00CC	SPI2 全局中断
36	43	可编程	USART1	0x0000_00D0	USART1 全局中断
37	44	可编程	UART1	0x0000_00D4	UART 全局中断
38	45	可编程	UART2	0x0000_00D8	UART2 全局中断

Puya Confidential

39	46	可编程	LPUART	0x0000_00DC	LPUART 全局和唤醒中断
40	47	可编程	RTCAlarm	0x0000_00E0	连到 EXTI 的 RTC 闹钟中断
41	48	可编程	I2C1_EV	0x0000_00E4	I2C1 事件中断
42	49	可编程	I2C1_ER	0x0000_00E8	I2C1 错误中断
43	50	可编程	I2C2_EV	0x0000_00EC	I2C2 事件中断
44	51	可编程	I2C2_ER	0x0000_00F0	I2C2 错误中断
45	52	可编程	CAN_INT0	0x0000_00F4	CAN 中断 0
46	53	可编程	CAN_INT1	0x0000_00F8	CAN 中断 1
47	54	可编程	COMP1	0x0000_00FC	COMP1 全局中断
48	55	可编程	COMP2	0x0000_0100	COMP2 全局中断
49	56	可编程	COMP3	0x0000_0104	COMP3 全局中断
50	57	可编程	COMP4	0x0000_0108	COMP4 全局中断
51	58	可编程	EPWM1	0x0000_010C	EPWM1 全局中断
52	59	可编程	EPWM2	0x0000_0110	EPWM2 全局中断
53	60	可编程	EPWM3	0x0000_0114	EPWM3 全局中断
54	61	可编程	EPWM4	0x0000_0118	EPWM4 全局中断
55	62	可编程	EPWM5	0x0000_011C	EPWM5 全局中断
56	63	可编程	EPWM6	0x0000_0120	EPWM6 全局中断
57	64	可编程	EPWM1_TZ	0x0000_0124	EPWM1 跳闸中断
58	65	可编程	EPWM2_TZ	0x0000_0128	EPWM2 跳闸中断
59	66	可编程	EPWM3_TZ	0x0000_012C	EPWM3 跳闸中断
60	67	可编程	EPWM4_TZ	0x0000_0130	EPWM4 跳闸中断
61	68	可编程	EPWM5_TZ	0x0000_0134	EPWM5 跳闸中断
62	69	可编程	EPWM6_TZ	0x0000_0138	EPWM6 跳闸中断

12.2 外部扩展中断/事件控制器（EXTI）

扩展中断和事件控制器，通过可配置线和直接输入线，管理着 CPU 和系统唤醒功能，并输出下述请求信号：

- 事件请求，送给 CPU 的事件输入
- 唤醒请求，送给功耗管理控制模块

EXTI 唤醒请求允许系统从停止模式唤醒，事件请求也可以在运行模式使用。

EXTI 允许管理多达 26 个可配置线和直接输入线（22 个可配置线和 4 个直接输入线）。

有 26 个能产生事件/中断请求的边沿检测器。其中 22 个可配置线，每个输入线可以独立地配置输入触发方式(上升沿或下降沿或者双边沿触发)。4 个可配置线和直接输入线，每个输入线都可以独立地被屏蔽。挂起寄存器保持着可配置线的中断请求。

12.2.1 EXTI 主要特性

EXTI 控制器的主要特性如下：

Puya Confidential

- 可通过 GPIO 和指定模块（如 PVD/RTC/COMP）唤醒低功耗模式
- 可配置线
  - 可选有效触发沿（上升沿/下降沿）
  - 中断挂起标志位
  - 独立中断和事件屏蔽位
  - 可软件触发
- 直接输入线
  - 固定的上升沿触发
  - 无中断挂起标志位
  - 独立中断和事件屏蔽位
  - 无软件触发

12.2.2 EXTI 唤醒事件管理

本产品可以处理外部或内部事件来唤醒内核（WFE）。唤醒事件可以通过下述配置产生：

- 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时在 Cortex-M4F 的系统控制寄存器中使能 SEVONPEND 位。当 CPU 从 WFE 恢复后，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。
- 配置一个外部或内部 EXTI 线为事件模式，当 CPU 从 WFE 恢复后，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

12.2.3 EXTI 功能说明

要产生中断，必须先配置好并使能中断线。22 个可配置线根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写'1'允许中断请求。当外部中断线上发生了期待的边沿时，将产生一个中断请求，对应的挂起位也随之被置'1'。在挂起寄存器的对应位写'1'，将清除该中断请求。如果需要产生事件，必须先配置好并使能事件线。根据需要的边沿检测通过设置 2 个触发寄存器，同时在事件屏蔽寄存器的相应位写'1'允许事件请求。当事件线上发生了需要的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置'1'。通过在软件中断事件寄存器写'1'，也可以通过软件产生中断/事件请求。

Puya Confidential

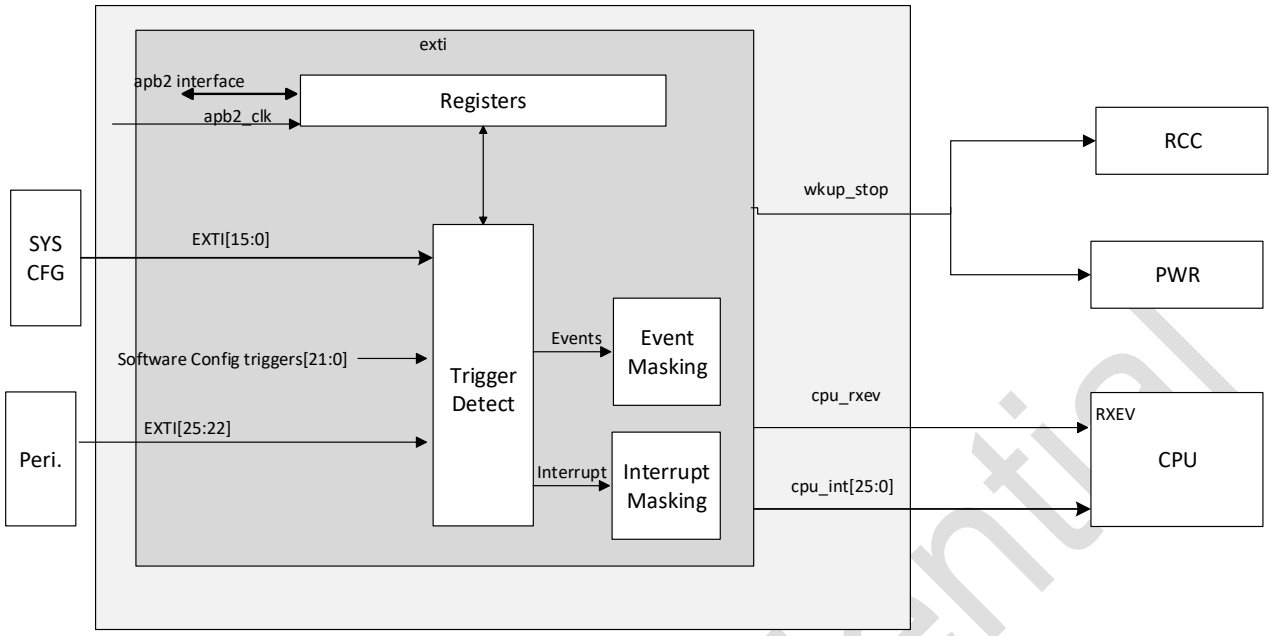


图 12-1 EXTI 框图

硬件中断选择

- 配置中断线的屏蔽位（EXTI_IMR）
- 配置所选中断线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）;

硬件事件选择

- 配置事件线的屏蔽位（EXTI_EMR）
- 配置事件线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）

软件中断/事件的选择

- 配置中断/事件线屏蔽位（EXTI_IMR，EXTI_EMR）
- 设置软件中断寄存器的请求位（EXTI_SWIER）

Puya Confidential

12.2.4 EXTI 外部中断/事件线映射

60 个通用 I/O 端口以下图的方式连接到 16 个外部中断/事件线上：

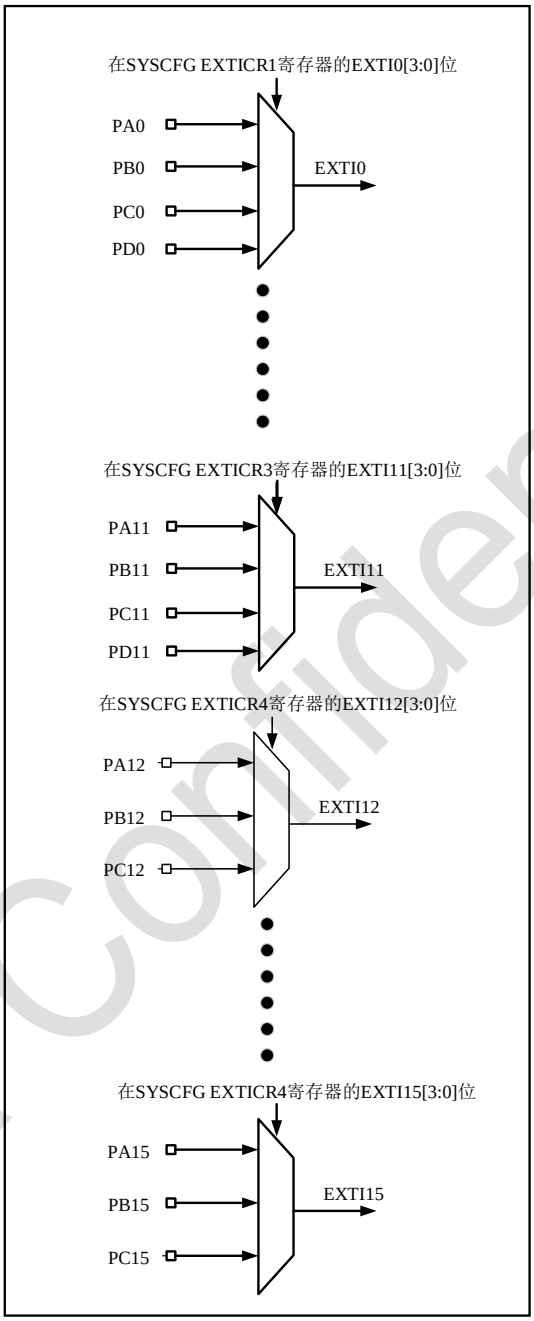


图 12-2 EXTI GPIO 复用

注：通过 SYSCFG_EXTICRx 配置 GPIO 线上的外部中断/事件，必须先使能 SYSCFG 时钟。

表 12-1 EXTI 线连接

EXTI	Line 源	Line 类型
0-15	GPIO	可配置线
16	PVD	可配置线
17	RTC alarm 事件	可配置线
18	COMP1 输出	可配置线

Puya Confidential

19	COMP2 输出	可配置线
20	COMP3 输出	可配置线
21	COMP4 输出	可配置线
22	I2C1 唤醒	直接输入线
23	I2C2 唤醒	直接输入线
24	LPTIM 唤醒	直接输入线
25	LPUART 唤醒	直接输入线

12.3 EXTI 寄存器

12.3.1 EXTI 中断屏蔽寄存器（EXTI_IMR）

偏移地址：0x00

复位值：0x03C0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	IMR25	IMR24	IMR23	IMR22	IMR21	IMR20	IMR19	IMR18	IMR17	IMR16
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25:0	IMRx(x=0-25)	RW	26’h3C0_0000	**IMRx**：线 x 上的中断屏蔽 (Interrupt Mask on line x) 0：屏蔽来自线 x 上的中断请求 1：开放来自线 x 上的中断请求

12.3.2 EXTI 事件屏蔽寄存器（EXTI_EMR）

偏移地址：0x04

复位值：0x0000 0000

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

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25:0	EMRx(x=0-25)	RW	26’h0	**EMRx**：线 x 上的事件屏蔽 (Event Mask on line x) 0：屏蔽来自线 x 上的事件请求 1：开放来自线 x 上的事件请求。

Puya Confidential

12.3.3 EXTI 上升沿触发选择寄存器（EXTI_RTSR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTSR 21	RTSR 20	RTSR 19	RTSR 18	RTSR 17	RTSR 16
										RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTSR 15	RTSR 14	RTSR 13	RTSR 12	RTSR 11	RTSR 10	RTS R9	RTS R8	RTS R7	RTS R6	RTSR 5	RTSR 4	RTSR 3	RTSR 2	RTSR 1	RTSR 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:22	Reserved	-	-	保留
21:0	RTSRx(x=0-21)	RW	22'h0	**RTSRx**：线 x 上的上升沿触发事件配置位 0：禁止输入线 x 上的上升沿触发(中断和事件) 1：允许输入线 x 上的上升沿触发(中断和事件)

12.3.4 EXTI 下降沿触发选择寄存器（EXTI_FTSR）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTSR 21	FTSR 20	FTS R 19	FTS R 18	FTS R 17	FTS R 16
										RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FTS R15	FTS R14	FTSR1 3	FTS R12	FTSR1 1	FTS R10	FTS R9	FTSR8	FTS R7	FTSR6	FTSR 5	FTSR 4	FTS R3	FTS R2	FTS R1	FTS R0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:22	Reserved	-	-	保留
21:0	FTSRx(x=0-21)	RW	22'h0	**FTSRx**：线 x 上的下降沿触发事件配置位 0：禁止输入线 x 上的下降沿触发(中断和事件) 1：允许输入线 x 上的下降沿触发(中断和事件)

12.3.5 EXTI 软件中断事件寄存器（EXTI_SWIER）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIE R21	SWIE R20	SWIE R19	SWIE R18	SWIE R17	SWIE R16
										RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SWIE R15	SWIE R14	SWIE R13	SWIE R12	SWIE R11	SWIE R10	SWI ER9	SWI ER8	SWI ER7	SWI ER6	SWIE R5	SWIE R4	SWIE R3	SWIE R2	SWIE R1	SWIE R0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:22	Reserved	-	-	保留

Puya Confidential

21:0	SWIERx(x=0-21)	RW	22'h0	**SWIERx**：线 x 上的软件中断 当该位为'0'时，写'1'将设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许 产生该中断，则此时将产生一个中断。 注：通过清除 EXTI_PR 的对应位(写入'1')，可以清除该位为'0'。

12.3.6 EXTI 挂起寄存器（EXTI_PR）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR21	PR20	PR19	PR18	PR17	PR16
										RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1

Bit	Name	RW	Reset Value	Function
31: 22	Reserved	-	-	保留
21:0	PRx(x=0-21)	RC_W1	22'h0	**PRx(x=0-21)**：挂起位 0：没有发生触发请求 1：发生了选择的触发请求 当在外部中断线上发生了选择的边沿事件，该位被置'1'。在该位中写入'1'可以清除它，也可以 通过改变边沿检测的极性清除。

Puya Confidential

13. 模拟/数字转换（ADC）

13.1 ADC 简介

芯片具有 2 个 12 位的 SARADC（逐次逼近型模数转换器）。每个 ADC 有 22 个多路复用通道，可以转换来自 14 个外部通道和 8 个内部通道的模拟信号，外部通道包含 4 对差分通道。模拟看门狗允许应用程序来检测输入电压是否超出用户设定的高低阈值。各种通道的 A/D 转换可以配置成单次、连续或间断转换模式。 ADC 转换的结果可以按照左对齐或右对齐的方式存储在 16 位数据寄存器中。

13.2 ADC 主要特性

- 高性能:
  - 12-bit、10-bit、8-bit 和 6-bit 分辨率可配置；
  - ADC 转换时间：0.25 us@12-bit（4 Msps，2.5 T SMP，f_{ADC} = 64 MHz）
  - 自校准（软件启动）
  - 支持规则转换和注入转换
  - 可编程采样时间
  - 可配置数据对齐模式（左对齐或者右对齐）
  - 规则通道转换支持 DMA 请求。
  - 配置支持 16 个规则序列转换
  - 配置支持 4 个注入序列转换
  - 配置支持 4 个注入通道转换数据寄存器
- 过采样器
  - 16 位数据寄存器
  - 过采样率可在 2x 到 256x 之间进行调整
  - 可编程数据最多可移位 8 位
- 数据预处理
  - 增益补偿
  - Offset 补偿
- 动态低功耗特性:
  - 自动延迟转换模式：防止以低频 PCLK 运行产生溢出
  - 为低功耗操作，降低工作频率，而仍然维持合适的 ADC 性能
- 模拟输入通道:
  - ADC1 连接 14 个外部通道，8 个内部通道
  - ADC2 连接 14 个外部通道，8 个内部通道
  - 每个 ADC 有 4 组差分对（对应 PGA1/2/3/4 的 P 和 N 端），其他外部通道和内部通道都是单端输入；差分对也可以配置成单端输入
  - ADC 外部的差分对通道快速通道，其他通道慢速通道
- 启动转换方式:

Puya Confidential

- 软件启动
- 可配置极性的硬件触发

- 转换模式：
  - 单次模式：1 次触发对所有选中的通道执行一次转换
  - 连续模式：1 次触发连续转换被选择的通道，直到软件终止
  - 间断模式：1 次触发连续转换子序列通道，多次触发直到转换所有通道
- 模拟看门狗
- 中断的产生：
  - 在单个通道采样结束
  - 在单个通道转换结束(规则/注入通道)
  - 在规则序列或注入序列转换结束
  - 模拟看门狗 1，2，3 事件
  - 过载事件
  - 校准完成
  - AD 上电完成

Puya Confidential

13.3 ADC 功能描述

13.3.1 ADC 框图

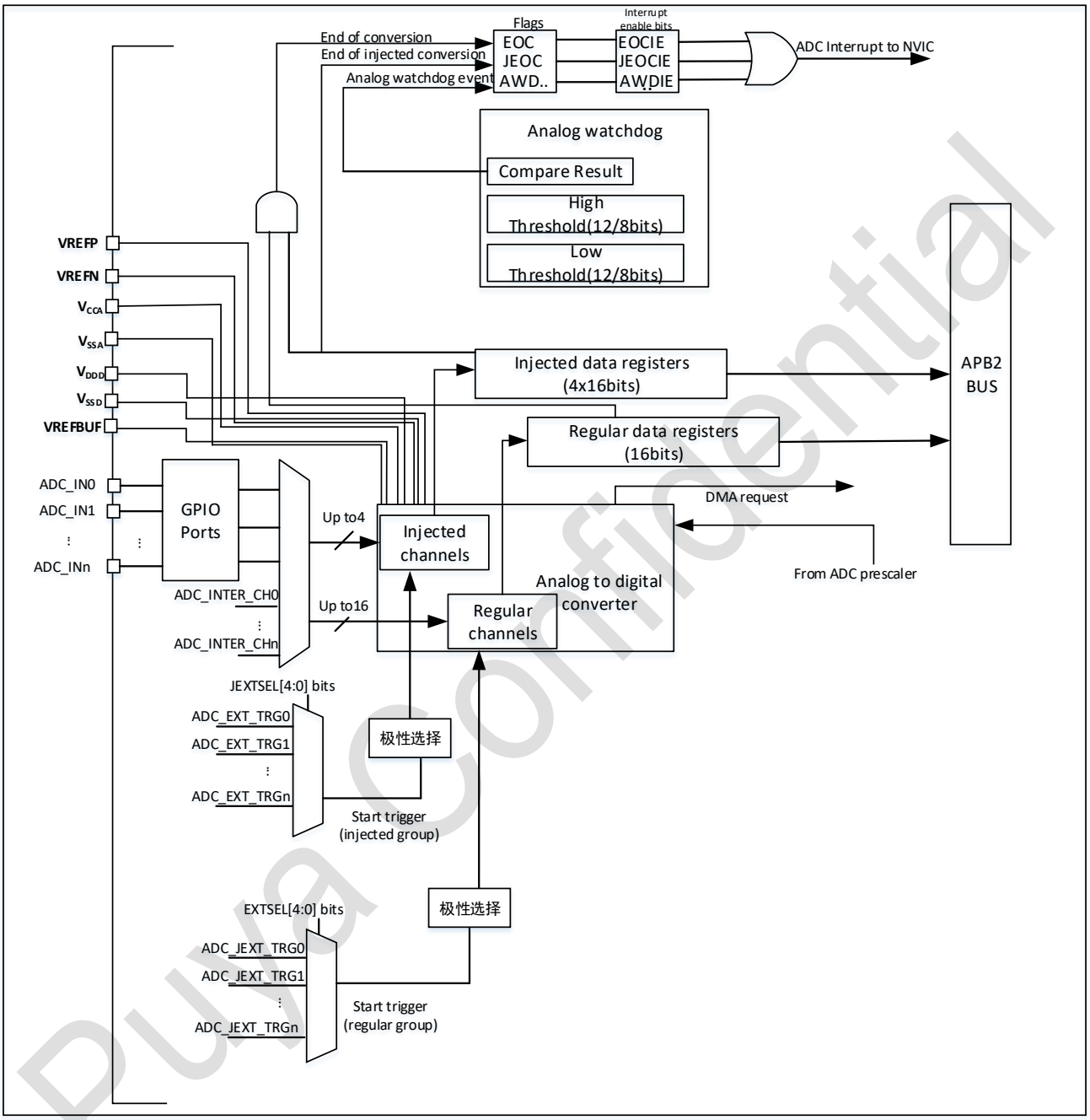


图 13-1 ADC 框图

13.3.2 单端和差分输入通道

ADC 部分通道可配置为单端输入或差分输入，通过 ADC_DIFSEL 寄存器中 DIFSEL 配置。当 DIFSEL[x]（x=6,8,10,12(ADC1)；(ADC2)）为 1 时，选择转换的通道为差分输入模式，否则为单端输入模式。必须当 ADC 被禁用时（ADEN=0）配置。请注意 DIFSEL 对应的单端通道始终保持为 0。

在单端模式下，通道 i（i=0-21）的模拟电压为 ADCy_INx（VINP[i]）和 V_{REF-}之差。

Puya Confidential

在差分模式下，通道 i（i=0-21）的模拟电压为 ADCy_INx（VINP[i]）和 ADCy_INx+1（VINN[i]）的电压差。

差分模式下，输入电压范围为 V_{REF-}到 V_{REF+}，达到 2xVREF+的全刻度。当 VINP[i]等于 V_{REF-}，VINN[i]等于 V_{REF+} 则最大负端输入差分电压（V_{REF-}）对应 0x000 输出。当 VINP[i]等于 V_{REF+}，VINN[i]等于 V_{REF-}则最大正端输入差分电压（V_{REF+}）对应 0xFFF 输出。当 VINP[i]和 VINN[i]连在一起，零输入差分电压对应 0x800 输出。

差分模式下的 ADC 灵敏度比单端模式下小两倍。内部通道（如 VSENSE 和 V_{REFINT}）只采用单端模式。完整的每个 ADC 输入通道连接描述，见相关章节描述。

内部通道只采用单端模式。

警告：

在差分输入模式下配置通道“i”时，其负端输入电压 VINN[i]已连接到另一个通道，因此，此通道不再可用于单端模式或差分模式，必须不能配置转换。ADC1/ADC2 共享一些通道：这可能导致另一个 ADC 上的通道不可用。唯一的例外是当主 ADC 和从 ADC 在交叉模式下运行。

下图为单端/差分通道的示意图：

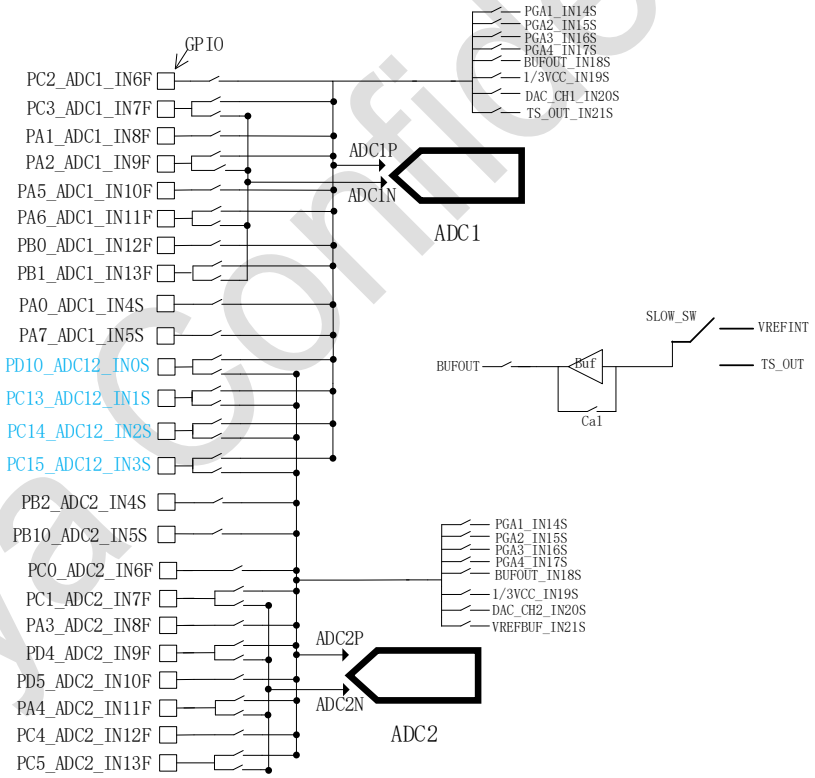


图 13-2 ADC1、ADC2 通道共享关系

13.3.3 ADC 校准

该 ADC 具有校准功能（软件启动），用户可通过软件使能 ADCAL 进行校准。在校准期间，ADC 计算一个用于 ADC 内部的校准因子。在 ADC 校准期间、未完成校准前，应用不能使用 ADC 模块。在使用 ADC 转换前，建议用户进行校准操作。校准用于消除芯片和芯片之间的，由于工艺变化引起的失调误差和失配误差。

Puya Confidential

ADC 软件校准

软件设置 ADCAL=1 可启动校准，校准只能在 ADC 未使能时（ADEN=0）启动，且仅支持选择 PCLK 作为 ADC 的时钟（校准时 ADC 时钟最大不能超过 16 MHz）。当校准完成后，ADCAL 被硬件清 0。

当 ADC 的工作条件发生改变时（V_{CC} 改变是失调误差和失配误差的主要因素，温度改变次之），推荐进行再次校准操作。

通过设置 ADC_CFGR2 寄存器中 CALNUM[2:0]来配置校准过程的重复次数，并对结果进行平均以得到更精确的校准结果。

内部校准寄存器可通过设置 ADC_CR 的 RSTCAL 复位，该位由硬件清除置 0。在校准寄存器被初始化后(即 RSTCAL 置 1 后)，该位即被清除。

校准的软件操作过程：

1. 确认 ADC_CR.ADEN=0
2. 设置 ADC_CR.RSTCAL（该步骤是可选的）；
3. 设置校准次数 ADC_CFGR2.CALNUM（该步骤是可选的）；
4. 设置 ADC_CR.ADCAL=1
5. 等待 ADC_SR.EOCAL=1

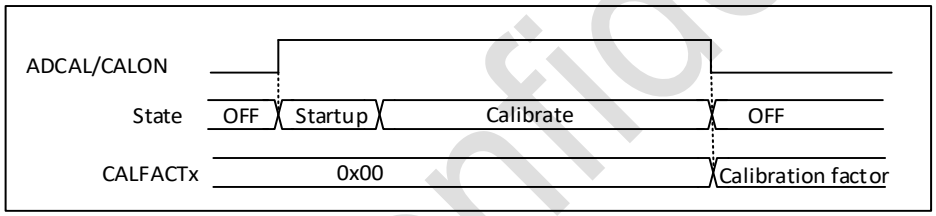


图 13-3 ADC 校准

- 软件写校准因子

1. 确保 ADEN=1，且没有转换正在进行，同时 ADCAL=0。
2. 使能 WRVLD，FACTSEL[4:0]选择写入哪个电容，设置 WCALFACT [8:0]写入校准因子。
3. ADEN 使能，进行 AD 转换时，校准因子自动注入模拟 ADC。

- 软件读校准因子

1. 确保 ADEN=1，且没有转换正在进行，同时 ADCAL=0。
2. 使能 RDVLD，FACTSEL[4:0]选择写入哪个电容。
3. 读 RCALFACT [8:0]校准因子。

13.3.4 ADC 开-关控制（ADEN、ADDIS、ADRDY）

芯片上电复位后，ADC 模块不使能，处于不工作状态(ADEN=0)。

ADEN 位用于控制位开启或关闭 ADC。ADC 启用后，在开始准确转换之前，ADC 需要一个稳定时间 t_{STAB}，如下图。有两个控制位用于启用或禁用 ADC：

- ADEN=1 时，启用 ADC。当 ADC 准备好进行操作时，标志 ADRDY 将被设置。
- ADDIS=1 时，禁用 ADC。一旦模拟 ADC 实际上被禁用，硬件会自动清除 ADEN 和 ADDIS。

ADC 规则转换由设置 ADSTART 来启动，根据 EXTEN 的配置，可以立即开始转换（软件启动）或者等待硬件触发后开始转换。

Puya Confidential

ADC 注入转换由设置 JADSTART 来启动，根据 JEXTEN 的配置，可以立即开始转换（软件启动）或者等待硬件触发后开始转换。

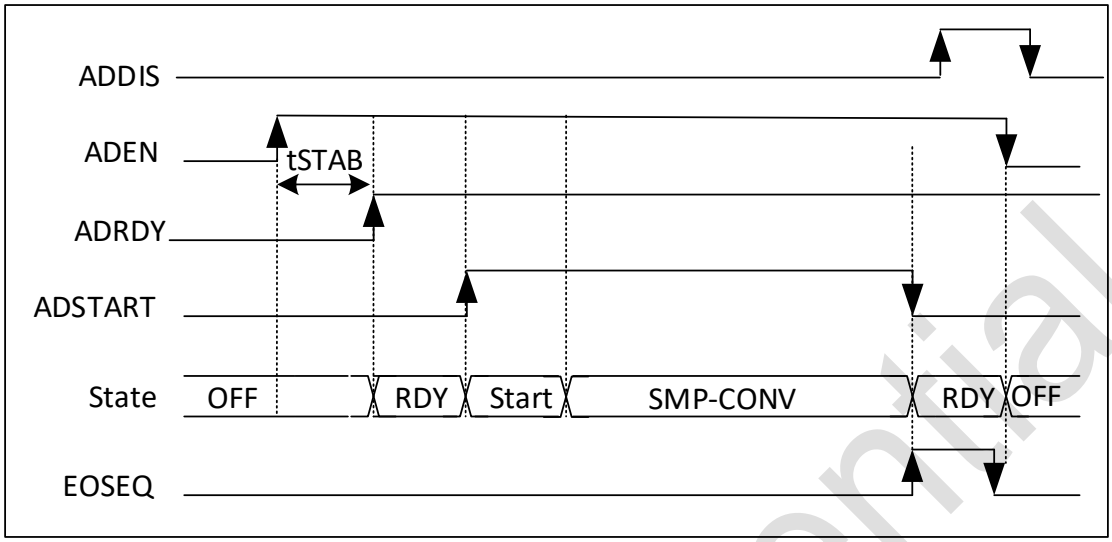


图 13-4 使能/禁止 ADC

■ 软件启用 ADC 的步骤

步骤：

1. 通过写入'1'清除 ADC_ISR 寄存器中的 ADRDY 位。
2. 设置 ADEN=1 或 ADCAL 以启用 ADC。(ADCAL 启用 ADC 进行校准)
3. 等待直到 ADRDY=1（在 ADC 上电结束后，ADRDY 将被设置）。这可以通过关联的中断来完成（设置 ADRDYIE=1）。(ADCAL 启用 ADC 进行校准后，硬件自清零)
4. 通过写入'1'清除 ADC_ISR 寄存器中的 ADRDY 位（可选步骤）。

注意事项：

1. ADEN 或 ADCAL 从 0 置 1 后，SARADC 模拟需要等待 3us 的时间。

■ 软件禁用 ADC 的步骤

1. 检查 ADSTART 和 JADSTART 是否均为 0，确保没有正在进行的转换。如有需要，通过设置 ADSTP=1 和 JADSTP=1 停止任何正在进行的规则和注入转换，然后等待直到 ADSTP=0 和 JADSTP=0。
2. 设置 ADDIS=1 以禁用 ADC。
3. 等待直至 ADEN=0，即模拟 ADC 实际上被禁用（一旦 ADEN=0，ADDIS 将自动重置）。

13.3.5 转换启动(ADSTART，JADSTART)

软件通过设置 ADSTART=1 启动 ADC 规则转换。

当 ADSTART 设置，则：

■ 当 EXTEN=0x0（软件触发）时，立即开始

■ 当 EXTEN 不等于 00 时，在选中的规则硬件触发源的下一个有效边沿开始

软件通过设置 JADSTART=1 启动 ADC 注入转换。

设置 JADSTART 后，转换开始:

■ 当 JEXTEN[1:0] = 00（软件触发）时，立即开始

Puya Confidential

■ 如果 JEXTEN[1:0]不等于 00 时，在选中的注入硬件触发源的下一个活动边沿开始

注意:在自动注入模式下（JAUTO=1），使用 ADSTART 位启动规则转换，然后是自动注入转换（JADSTART 必须保持清除）。

ADSTART 和 JADSTART 位也用于说明目前 ADC 转换操作是否正在进行。

ADSTART 位由硬件清除：

■ 当软件触发（CONT=0，EXTSEL=0x0）：

1. 在规则转换序列结束时（EOS 信号产生），硬件清零 ADSTART。
2. 若 DISCEN=1，在子序列转换结束后，硬件清零 ADSTTART。

■ 在所有的情况下（CONT=X，EXTSEL=X）,在软件设置 ADSTP 后，硬件清零 ADSTART。

注：在连续模式（CONT=1）下，ADSTART 位不能在 EOS 产生时硬件清除，因为序列会自动重新开始转换。当硬件触发时（CONT=0 且 EXTSEL ≠0x00），则当 EOS 标志设置后，ADSTART 不会被硬件清 0，避免软件重新设置 ADSTART 位且要确保无硬件触发事件错过。

JADSTART 由硬件清除:

1. 当软件触发（JEXTSEL = 0x0）：
   1) 在注入转换序列结束(JEOS 产生)时，硬件清零 JADSTART。
   2) 若 JDISCEN = 1，子序列转换结束，硬件清零 JADSTART。
2. 在所有情况下(JEXTSEL=x)，软件设置 JADSTP 后，硬件清零 JADSTART。

注意：当选择软件触发时，如果 EOC 标志未清零，则不应设置 ADSTART 位。

13.3.6 停止正在进行的转换（ADSTP、JADSTP）

通过设置 ADC_CR 寄存器中的 ADSTP=1 可以停止当前正在进行的规则转换，通过设置 ADC_CR 寄存器中的 JADSTP=1 可以停止当前正在进行的注入转换。

停止转换将重置正在进行的 ADC 操作。然后可以重新配置 ADC（例如：改变通道选择或触发方式），为新的 AD 转换做好准备。

请注意，在规则转换仍在运行时可以停止注入转换，反之亦然。例如，这允许在规则转换仍在运行时重新配置注入的转换序列和触发方式（反之亦然）。

当 ADSTP 位由软件设置时，任何正在进行的规则转换被中止，转换结果被丢弃(ADC_DR 寄存器不更新当前转换)。

当 JADSTP 位由软件设置时，任何正在进行的注入转换将被中止，转换结果将被丢弃(ADC_JDRy 寄存器不更新当前转换)。

扫描序列也被中止和重置（这意味着重新启动 ADC 将启动一个新序列）。

一旦 ADC 重置完成，ADSTP/ADSTART（在规则转换的情况下），或 JADSTP/JADSTART（在注入转换的情况下)被硬件清除。若软件需要检测 ADC 是否重置,可以通过轮询 ADSTART(或 JADSTART)是否被重置。

注意：在自动注入模式下（JAUTO=1），设置 ADSTP 位会同时终止规则转换和注入转换（不能使用 JADSTP）。

Puya Confidential

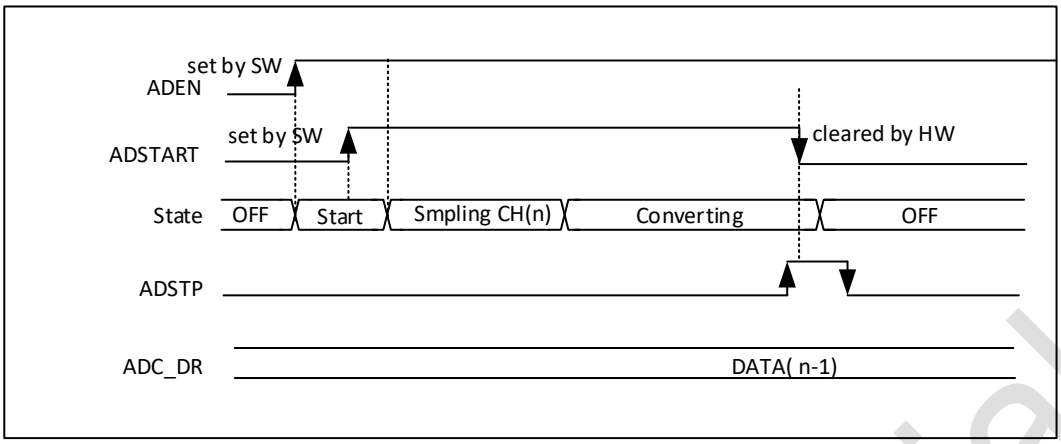


图 13-5 ADSTP 停止规则转换

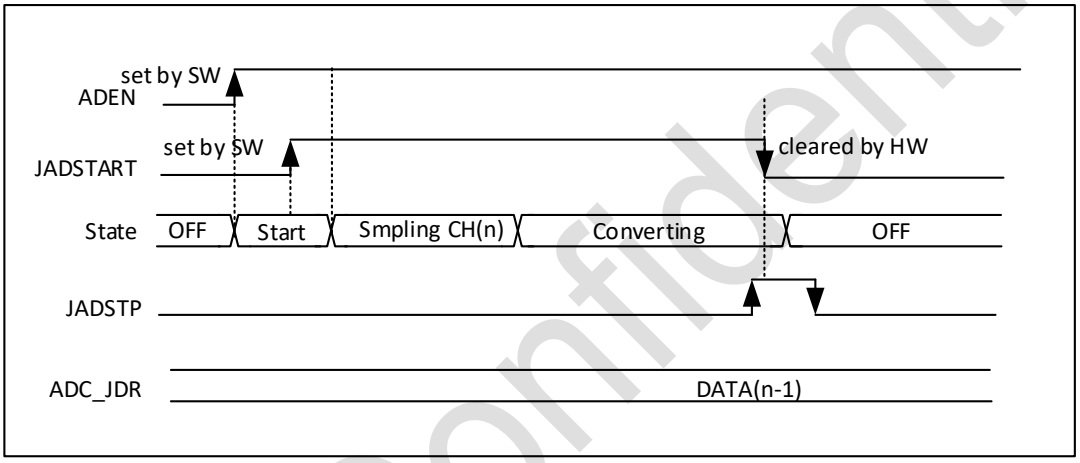


图 13-6 JADSTP 停止注入转换

13.3.7 ADC 时序（单次/连续模式，硬件/软件触发）

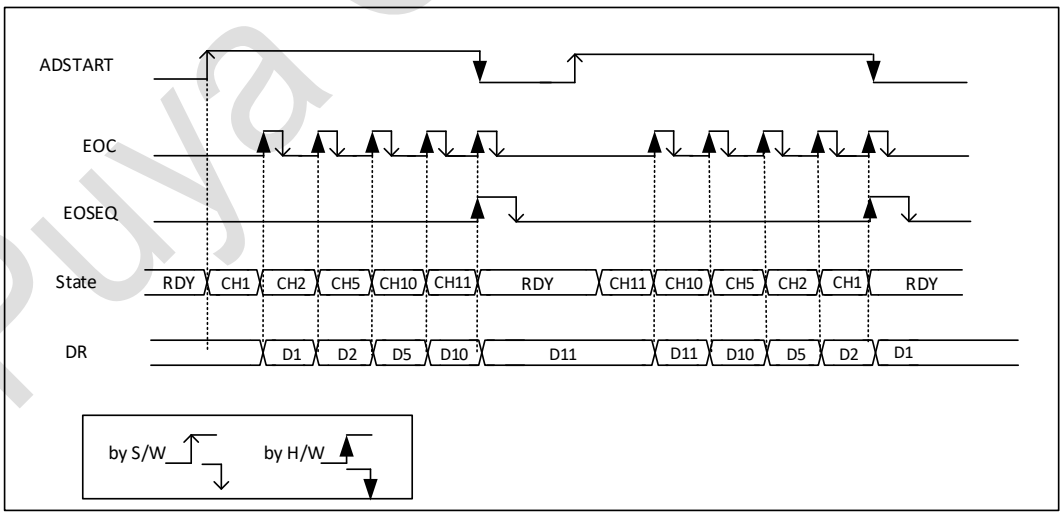


图 13-7 单次模式，软件触发

1. EXTEN=0x0，CONT=0
2. SQ1~5 分别配置为通道 1，2，5，10，11，AUTDLY=0

Puya Confidential

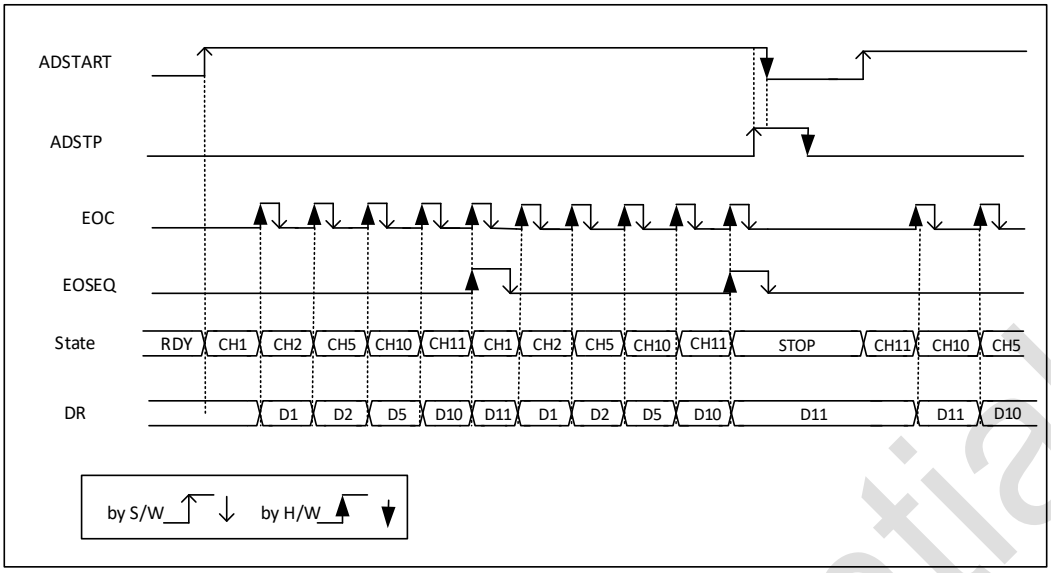


图 13-8 连续模式，软件触发

1. EXTEN=0x0, CONT=1,
2. SQ1~5 分别配置为通道 1，2，5，10，11, AUTDLY=0

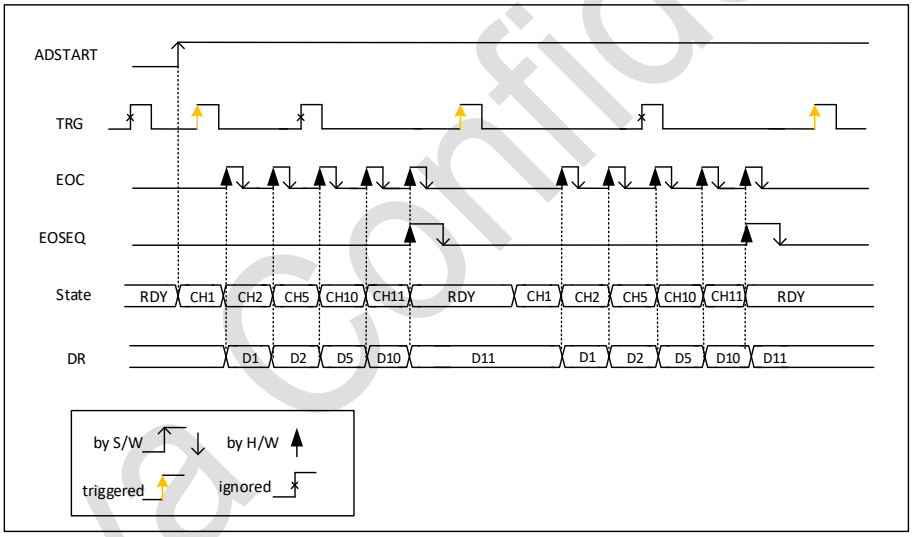


图 13-9 单次转换一个序列，硬件触发

1. EXTSEL=TRGx，EXTEN=0x1（上升沿），CONT=0
2. SQ1~5 分别配置为通道 1，2，5，10，11，AUTDLY=0

Puya Confidential

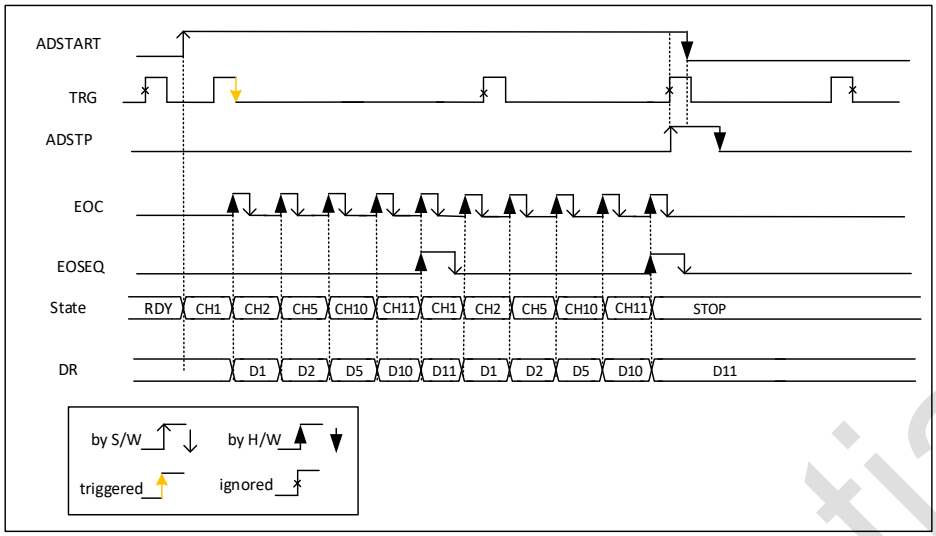


图 13-10 连续转换一个序列，硬件触发

1. EXTSEL=TRGx，EXTEN=0x2（下降沿），CONT=1
2. SQ1~5 分别配置为通道 1，2，5，10，11，AUTDLY =0

13.3.8 写 ADC 控制位的限制

软件写 ADC 寄存器控制位，需要遵循以下原则：

1. ADC 已启用且没有待处理的禁能 ADC 请求（即 ADEN 必须为 1 且 ADDIS 为 0），软件才被允许写入 ADC_CR 寄存器中的 ADSTART、JADSTART 和 ADDIS 控制位。
2. 当 ADC 已启用，可能正在转换，并且没有待处理的禁能 ADC 请求（即 ADSTART 或 JADSTART 必须等于 1 且 ADDIS 为 0）时，软件才被允许写入 ADC_CR 寄存器中的 ADSTP 或 JADSTP 控制位。
3. 在 ADEN=1 且 ADSTART 为 0 时，软件才能配置 ADDIS=1；
4. 软件写 ADC_CFGR，ADC_SMPRx，ADC_SQRy，ADC_JDRy，ADC_OFRy，ADC_IER，ADC_TR，ADC_JSQR 等寄存器，需要遵循以下原则：

1. 规则序列相关寄存器，在 ADEN=1 且 ADSTART=0 时配置；
2. 注入序列相关寄存器，在 ADEN=1 且 JADSTART=0 时配置；

注意：硬件并未提供写操作保护措施，若违反上述原则配置寄存器，ADC 的行为可能会进入未知状态。为了从这种状况中恢复，需要将 ADC 禁用（将 ADEN 清零，并同时清零 ADC_CR 寄存器中的所有位）。

13.3.9 ADC 时钟

ADC 具有双时钟域架构，ADC 时钟（ADC_CLK）独立于 APB 时钟（PCLK）。ADC_CLK 可由两种时钟源产生，如下图所示：

Puya Confidential

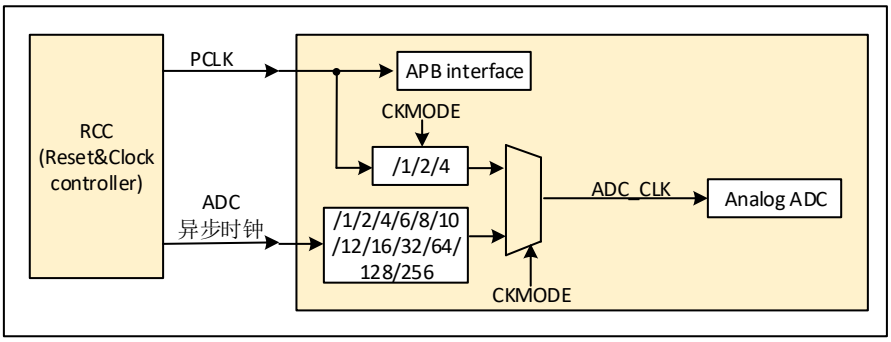


图 13-11 ADC 时钟结构

模拟 ADC 的输入时钟（ADC_CLK）有两个可选时钟源：

1. ADC_CLK 可选“ADC 异步时钟”时钟源，该时钟源独立于 APB 时钟，与 APB 时钟异步。要选择此时钟方案，必须将 ADC_CCR 寄存器的 CKMODE[1:0]位复位。
2. ADC 时钟可选 PCLK（ADC 总线接口时钟）或其分频时钟源，该时钟源与 APB 时钟同步，可选时钟分频因子（1、2 或 4，具体根据 CKMODE[1:0]位而定）。要选择此时钟方案，ADC_CCR 寄存器的 CKMODE[1:0] 位不得为“00”。

在选项 1 中，对 ADC_CCR 寄存器中的 PRESC[3:0]位进行编程时，生成的 ADC 时钟最后会除以预分频系数（1、2、4、6、8、10、12、16、32、64、128、256）。

选项 1 的优点在于无论选择哪种 APB 时钟方案，都可以达到最大 ADC 时钟频率。

选项 2 的优势在于不必进行时钟域同步。如果 ADC 由 timer 触发，并且应用要求 ADC 精确触发（不存在任何不确定性），可使用此选项（否则，同步两个时钟域会为触发时刻带来不确定性）。

注意：选择 CKMODE[1:0]=11（PCLK 1 分频）时，用户必须确保 PCLK 的占空比为 50%。可通过选择占空比为 50%的系统时钟并在 RCC 不配置 APB 预分频器来实现(请参见复位和时钟控制器部分)。

13.3.10 ADC 通道选择（SQRx，JSQR）

ADC1 有 14 个外部通道和 8 个内部通道；ADC2 有 14 个外部通道和 8 个内部通道。

表 13-1 ADC1/2 内部通道

ADC1			ADC2		
通道	**Name**	**Source**	**通道**	**Name**	**Source**
ADC1_CH14	ADC1_INTER_CH0	OPA1_VOUT	ADC2_CH14	ADC2_INTER_CH0	OPA1_VOUT
ADC1_CH15	ADC1_INTER_CH1	OPA2_VOUT	ADC2_CH15	ADC2_INTER_CH1	OPA2_VOUT
ADC1_CH16	ADC1_INTER_CH2	OPA3_VOUT	ADC2_CH16	ADC2_INTER_CH2	OPA3_VOUT
ADC1_CH17	ADC1_INTER_CH3	OPA4_VOUT	ADC2_CH17	ADC2_INTER_CH3	OPA4_VOUT
ADC1_CH18	ADC1_INTER_CH4	BUF_OUT	ADC2_CH18	ADC2_INTER_CH4	BUF_OUT
ADC1_CH19	ADC1_INTER_CH5	Vcc/3	ADC2_CH19	ADC2_INTER_CH5	Vcc/3
ADC1_CH20	ADC1_INTER_CH6	DAC_OUT1	ADC2_CH20	ADC2_INTER_CH6	DAC_OUT2
ADC1_CH21	ADC1_INTER_CH7	VREFINT	ADC2_CH21	ADC2_INTER_CH7	VREFBUF(PA0)

ADC 通道选择可通过 ADC_SQRx 或 ADC_JSQR 配置，配置值和通道对应关系见 IOMUX。

ADC 将转换分为两组：规则转换和注入转换。每个组包含一个转换序列，该序列可按任意顺序在任意通道上完成。例如，可按以下顺序对序列进行转换： ADC1_IN3、ADC1_IN8、ADC1_IN2、ADC1_IN2、ADC1_IN0、ADC1_IN2、ADC1_IN2、ADC1_IN15。所有通道都可以按注入或规则通道组进行转换。

Puya Confidential

- 一个规则转换组最多由 16 个转换构成。转换序列的规则通道及其转换顺序在 ADC_SQRx 寄存器中选择。规则转换组中的序列长度必须写入 L[3:0]。
- 一个注入转换组最多由 4 个转换构成。注入通道及其转换顺序在 ADC_JSQR 寄存器中选择。注入转换组中的序列长度必须写入 JL[1:0]位。

在规则转换期间，不得修改 ADC_SQRx 寄存器，必须先通过写入 ADSTP=1 来停止 ADC 的规则转换，再修改 ADC_SQRx 寄存器。

13.3.11 动态低功耗特性

自动延迟转换模式（AUTDLY）

自动延迟转换对于简化软件以及优化应用程序的性能非常有用，因为程序在低频运行时可能会遇到 ADC 过载的风险。

当在 ADC_CFGR 寄存器中设置 AUTDLY 为 1 时，只有在当前 ADC 转换数据处理完成后才开始新的转换：

- 对于规则转换：一旦 ADC_DR 寄存器被读取或 EOC 位被清除，才开始新的转换
- 对于注入转换：当 JEOS 位被清除时，才开始新的转换

这是一种自适应 ADC 速度和自适应系统读取 ADC 数据速度的方法。

在每个规则转换(无论 DISCEN=0 或 1)后，ADC 进行等待状态，等待 AD 转换数据处理完成。

在每个注入转换序列(无论 JDISCEN=0 或 1)后，ADC 进行等待状态，等待 AD 转换数据处理完成。

注：在注入序列的每次转换结束后不会进行等待状态，而是在整个注入序列结束后进入等待状态。

在等待状态时，将忽略在此期间发生的硬件触发事件(针对同一组转换)。

注：对于软件触发，在等待状态期间，仍然可以通过置位 ADSTART 或 JADSTART 来重新启动转换：在启动新的转换之前，由软件读取数据以避免数据覆盖丢失。

注意：ADSTP/JADSTP 后 EOC/JEOS 即使有效，也不再屏蔽触发

在不同组的转换之间不插入等待状态（规则转换后跟注入转换，反之亦然）：

- 如果在规则转换的等待状态期间发生注入触发，则注入转换立即开始。
- 一旦注入序列完成，ADC 在启动新的规则转换之前等待前一个规则转换的等待状态结束。

在自动注入模式（JAUTO=1）中，行为略有不同，只有在注入转换序列的等待状态结束时(当 JEOS 被清除时)，新的规则转换才能开始。这是为了确保软件可以在开始一个新序列之前读取给定序列的所有数据。

要在 JAUTO=1，CONT=1, AUTDLY =1 下停止转换，请执行以下步骤:

1. 等待 JEOS=1（不再重新启动转换）
2. 清零 JEOS,
3. 设置 ADSTP = 1
4. 读取规则数据。

如果不遵守此过程，在上例的 CONT=1 的情况下，如在设置 ADSTP 后清零 JEOS (步骤 2 和 3 颠倒)，可以重新启动新的规则序列。

在 AUTDLY 模式中，如果硬件规则触发事件发生在已经在进行的规则序列期间，或者发生在序列最后一次规则转换之后的等待状态期间，则忽略该事件。如果发生在此等待状态之后，则认为该触发事件是挂起的。

Puya Confidential

在 AUTDLY 模式中，如果硬件注入的触发事件发生在已经在进行的注入序列期间，或者发生在序列的最后一次注入转换之后的等待状态期间，则忽略该事件。

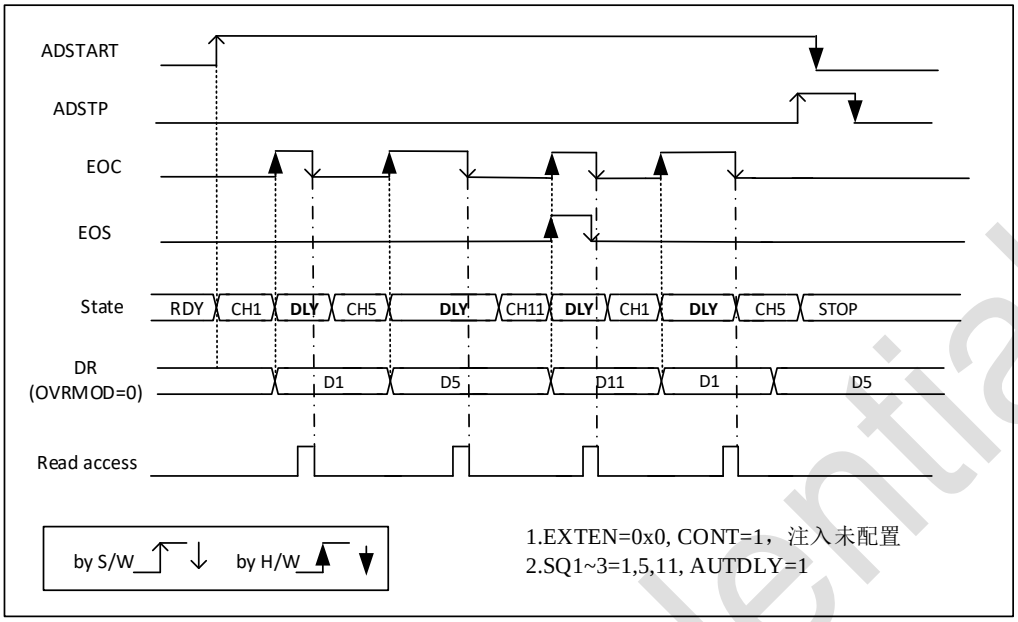


图 13-12 AUTDLY 模式

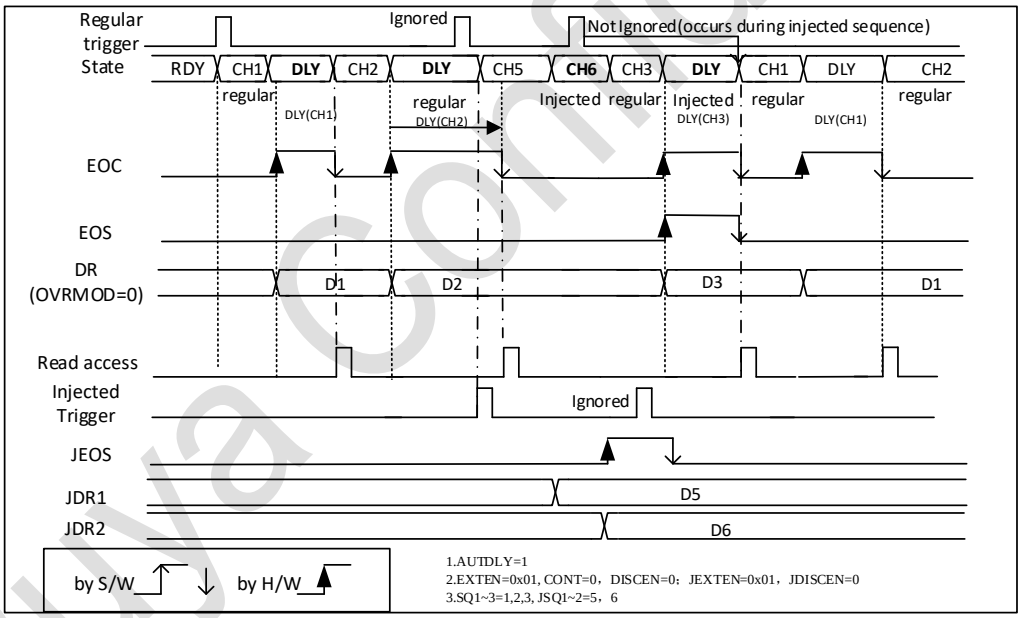


图 13-13 规则序列被注入触发中断

Puya Confidential

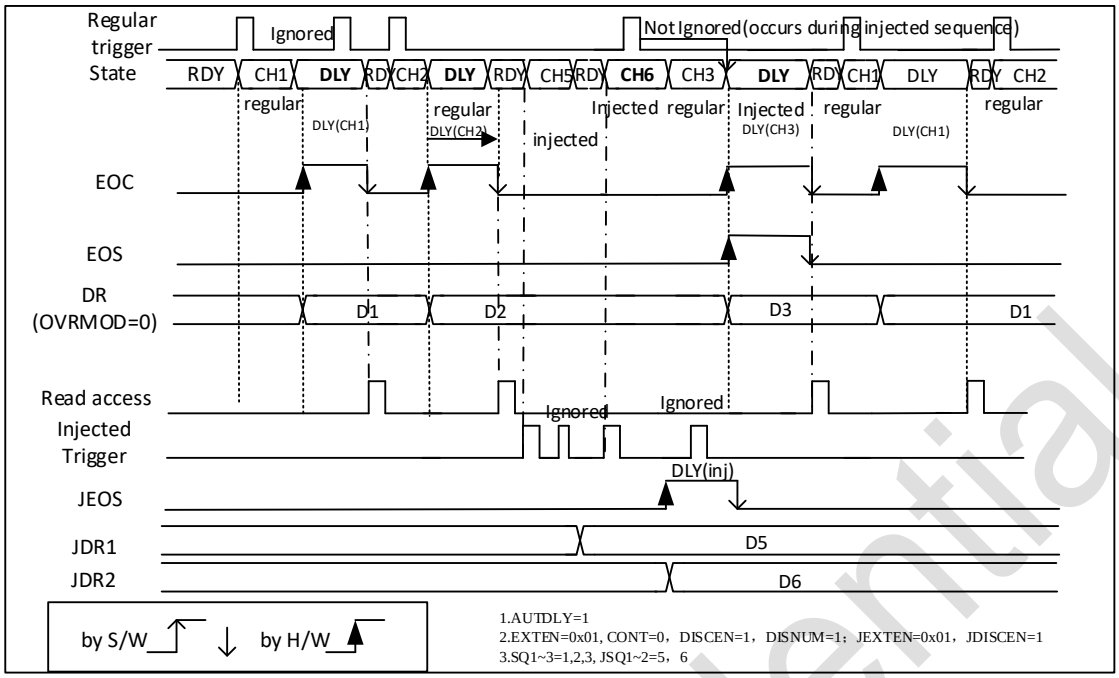


图 13-14 规则序列被注入触发中断

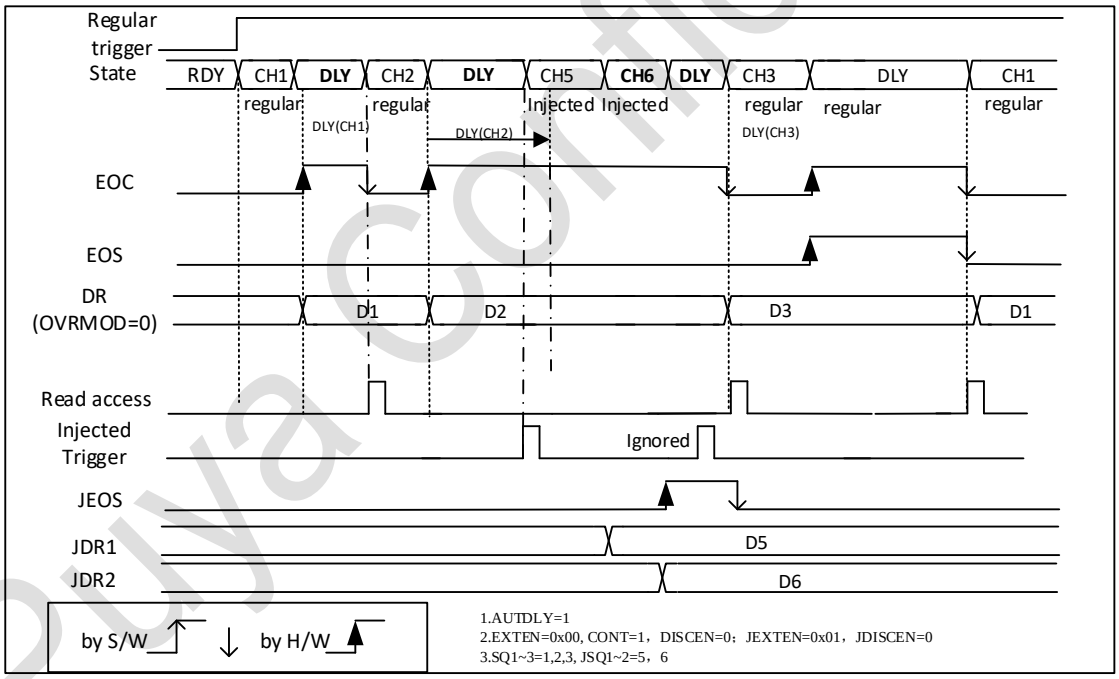


图 13-15 规则序列被注入触发中断

Puya Confidential

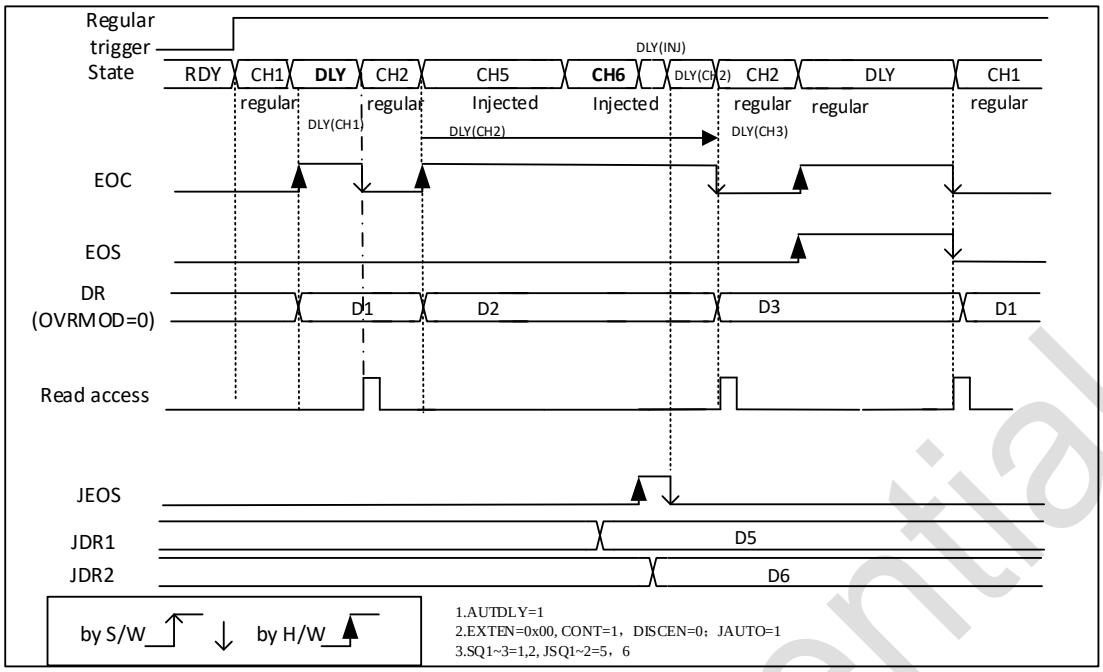


图 13-16 规则序列被注入触发中断

13.3.12 转换模式

单次转换模式（CONT=0）

单次转换模式下，ADC 执行一次序列转换，转换所有被选的通道。当 ADC_CFGR 寄存器中的 CONT=0，DISCEN=0 时，ADC 为单次转换模式。

该模式可由下述方法启动：

- 在 ADC_CR 寄存器中设置 ADSTART 位（规则转换）
- 在 ADC_CR 寄存器中设置 JADSTART 位（注入转换）
- ADSTART=1 且硬件触发事件（规则转换）
- JADSTART=1 且硬件触发事件（注入转换）

规则转换，每次转换完成后：

- 转换的数据结果存放到 16 位寄存器 ADC_DR 中。
- EOC（转换结束标志）标志置位
- 若 EOCIE 位置位则产生一个中断。

规则转换，所有通道序列转换完成后：

- EOS（序列结束）标志置位
- 若 EOSIE 位置位则产生一个中断

注入转换，每次转换完成后：

- 转换的数据结果存放到 16 位寄存器 ADC_JDRx 中
- JEOC（转换结束标志）标志置位
- 若 JEOCIE 位置位则产生一个中断。

注入转换，所有通道序列转换完成后：

- JEOS（序列结束）标志置位
- 若 JEOSIE 位置位则产生一个中断

Puya Confidential

转换结束后，ADC 转换停止直到新的触发事件或 ADSTART/JADSTART 重新置位。

若转换单一通道，则可配置序列长度为 1。

连续转换模式（CONT=1）

该模式仅针对规则转换。

在连续转换模式中，当软件或硬件触发事件产生，ADC 执行一个序列转换。转换所有的通道且自动重新开始执行相同的序列转换。

当寄存器 ADC_CFGR 中的 CONT=1 时，ADC 选择为连续转换模式。

ADC 转换可由下述两种方法启动：

- 在 ADC_CR 寄存器中设置 ADSTART 位
- ADSTART=1 且硬件触发事件硬件触发事件

在序列通道的转换期间，每次转换完成后：

- 转换的数据结果存放到 16 位寄存器 ADC_DR 中
- EOC（转换结束标志）标志置位
- 若 EOCIE 位置位则产生一个中断

通道序列转换完成后：

- EOS（序列结束）标志置位
- 若 EOSIE 位置位则产生一个中断

一次序列转换结束后，ADC 立即重新转换相同的序列通道。

注：若转换单一通道，则可编程一个长度为 1 的一个转换序列。

ADC 不能同时处于间断转换模式和连续转换模式（DISCEN=1，CONT=1）

注入通道不支持连续转换模式，唯一的例外是当连续自动注入时（JAUTO=1，CONT=1）。

间断转换模式（DISCEN=1/JDISCEN=1）

间断转换模式时，ADC 将选择的一个序列分成子序列(序列长度 1-8)，通过多次外部触发子序列，完成整个序列转换，该模式能够运行在规则组和注入组。DISCNUM 规定了规则通道子序列长度，ADC_SQRx 规定了一个规则序列所有转换通道，其中 L[3:0]位规定了该序列长度。 ADC_JSQR 寄存器规定了一个注入组序列所有转换通道，其中 JL[1:0]位规定了该序列长度。采样结束，则产生 EOSMP，如设置了 EOSMPIE，则产生中断。

注意：注入通道组的子序列为 1.该模式下禁止 DISCEN 与 JDISCEN 同时使能.

- 规则组

一个外部触发信号启动 ADC_SQRx 描述通道 n(n≤8)次转换，直到此序列所有转换完成为止。总序列长度由 L[3:0]定义。

例如：

n=3，被转换的通道 = 0, 1, 2, 3, 6, 7, 9, 10

第一次触发：转换的序列为 0，1，2，每个通道转换结束产生 EOC 事件。

第二次触发：转换的序列为 3，6，7，每个通道转换结束产生 EOC 事件。

第三次触发：转换的序列为 9，10，每个通道转换结束产生 EOC 事件。序列结束产生 EOS 事件。

第四次触发：转换的序列 0，1，2，每个通道转换结束产生 EOC 事件。

注：当一规则组以间断模式转换时，转换序列结束后不自动从头开始。当所有子组被转换完成，下一次触发启动第一个子组的转换。在上面的例子中，第四次触发重新转换第一子组的通道 0，1 和 2。

Puya Confidential

在软件同时使能连续转换模式和间断转换模式时，硬件会屏蔽这种情况(DISCEN =1，CONT=1)

- 注入组

一个外部触发信号启动 ADC_JSQR 描述通道 1 次转换，直到序列中所有转换完成为止。总序列长度 JL[1:0]位定义。

例如：

n=1，被转换的通道 = 1，2，3

第一次触发：通道 1 被转换，每个通道转换结束产生 JEOC 事件。

第二次触发：通道 2 被转换，每个通道转换结束产生 JEOC 事件。

第三次触发：通道 3 被转换，每个通道转换结束产生 JEOC 事件。序列结束产生 JEOS 事件。

第四次触发：通道 1 被转换，每个通道转换结束产生 JEOC 事件。

注：1 当完成所有注入通道转换，下个触发启动第 1 个注入通道的转换。在上述例子中，第四个触发重新转换第 1 个注入通道 1。

2 不能同时使用自动注入和间断模式。

3 必须避免同时为规则和注入组设置间断模式。间断模式只能作用于一组转换。

13.3.13 注入通道管理

注入通道外部触发优先级高于规则通道外部触发，即注入通道外部触发可以中断正在进行中的规则通道转换。注入通道有两种中断方式：触发注入和自动注入。

触发注入

触发注入模式，JAUTO 必须为 0。

- 通过外部触发或通过设置 ADC_CR 寄存器中的 ADSTART 位启动一组规则通道的转换。
- 如果发生外部注入触发，或者在转换一组规则通道期间设置了 ADC_CR 寄存器中的 JADSTART 位，则当前转换重置并启动注入通道序列 (所有注入通道都转换一次)。
- 然后，从上次中断的规则转换中恢复规则组通道的规则转换。
- 如果在注入转换期间发生规则触发事件，则不会中断注入转换，而是在注入序列的末尾执行规则序列。

注意：当使用触发注入时，必须确保触发事件之间的间隔长于注入序列。例如，如果序列长度为 30 个 ADC 时钟周期，则触发器之间的最小间隔必须为 31 个 ADC 时钟周期。

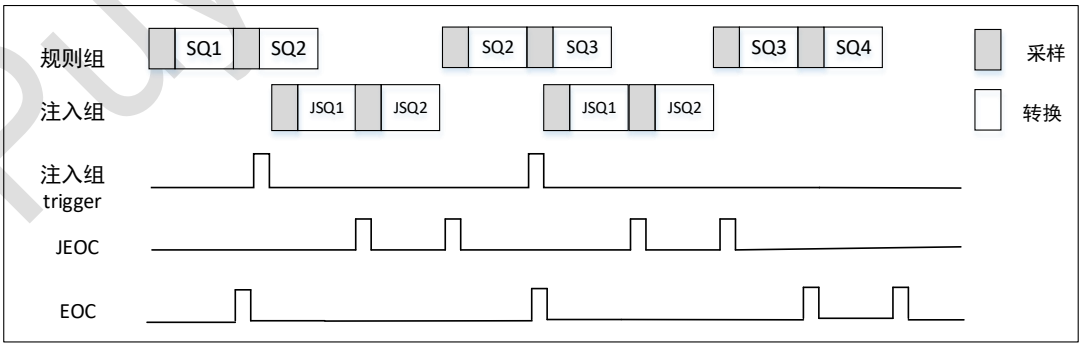


图 13-17 ADC 触发注入示意图

Puya Confidential

自动注入

如果设置了 ADC_CFGR 寄存器中的 JAUTO 位，则注入组中的通道在规则通道组之后自动转换。这可以用来转换在 ADC_SQRy 和 ADC_JSQR 中编程的最多 20 个转换通道（规则通道最多 16 个注入最多 4 个）。

在这种模式下，将 ADC_CR 寄存器中的 ADSTART 位设置为 1 启动规则转换，然后是注入转换(JADSTART 必须保持清除)。设置 ADSTP 位会终止规则转换和注入转换(不能使用 JADSTP 位)。

在此模式下，必须禁用注入通道上的外部触发。

如果除了 JAUTO 位之外还设置了 CONT 位，则连续转换规则通道，然后是注入通道。

注意:不可能同时使用自动注入和不连续模式。

当 DMA 在 JAUTO 模式下用于读取规则转换序列数据时，需要将其编程为循环模式(在 DMA_CCRx 寄存器中设置 CIRC 位)。如果 CIRC 位复位(单发模式)，在 DMA 传输完成事件发生时，JAUTO 序列将停止。

13.3.14 模拟看门狗

三个 AWD 模拟看门狗监测某些通道是否保持在配置的电压范围内（窗口）

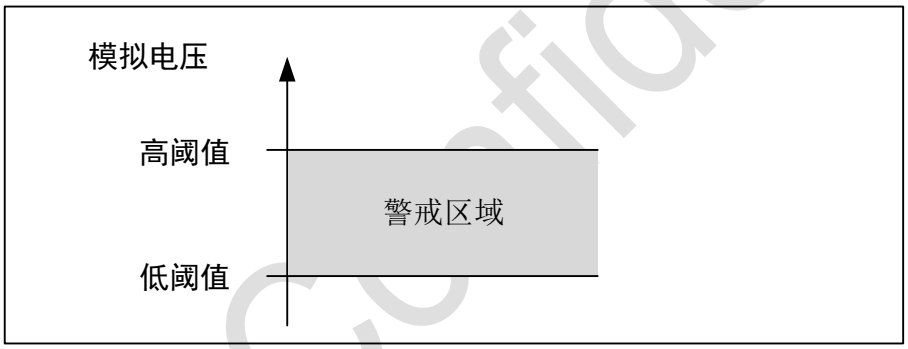


图 13-18 模拟看门狗保护区域

AWDx 标志和中断

通过在 ADC_IER 寄存器（x=1,2,3）中设置 AWDxIE，可以为 3 个模拟看门狗中的每一个启用中断。AWDx（x=1,2,3）标志由软件写 1 来清除。在数据对齐之前将 ADC 转换结果与低阈值和高阈值进行比较。

模拟看门狗 1 描述

通过设置的 AWD1EN 位启用 AWD 模拟看门狗 1。该看门狗监视一个选定的通道或所有启用的通道是否保持在配置的电压范围（窗口）内。

下表显示了应如何配置 ADC_CFGR 寄存器以启用一个或多个通道上的模拟看门狗。

表 13-2 模拟看门狗通道选择

模拟看门狗保护通道	AWD1SGL bit	AWD1EN bit	JAWD1EN bit
None	x	0	0
所有注入通道	0	0	1
所有规则通道	0	1	0
所有规则和注入通道	0	1	1
单个注入通道	1	0	1

Puya Confidential

单个规则通道	1	1	0
单个规则/注入通道	1	1	1

通过 AWD1CH[4:0]选择看门狗保护通道，这些被保护通道必须在规则或注入序列进行转换。

如果 ADC 转换的模拟电压低于较低阈值或高于较高阈值，则设置 AWD1 模拟看门狗状态位。这些阈值被写在模拟看门狗 1 的 ADC_TR1 寄存器的位 HT1[11:0]和 LT1[11:0]中。当转换分辨率小于 12 位的数据时（根据位 RES[1:0]），阈值的 LSB 必须保持清除，因为内部比较总是在完整的 12 位原始转换数据上执行（左对齐）。

下表描述了如何针对模拟看门狗 1 的所有可能的分辨率执行比较。

表 13-3 模拟看门狗 1 比较

分辨率(RES[1:0])	模拟看门狗比较：		描述
	原始转换左对齐	阈值	
00：12bit	DATA[11:0]	LT1[11:0]和 HT1[11:0]	-
01：10bit	DATA[11:2]，00	LT1[11:0]和 HT1[11:0]	用户必须将 LT1[1:0]和 HT1[1:0]配置为 00
10：8bit	DATA[11:4]，0000	LT1[11:0]和 HT1[11:0]	用户必须将 LT1[3:0]和 HT1[3:0]配置为 00
11：6bit	DATA[11:6]，000000	LT1[11:0]和 HT1[11:0]	用户必须将 LT1[5:0]和 HT1[5:0]配置为 00

有关模拟看门狗比较的更多详细信息，请参阅“增益补偿”一节。

看门狗 1 的模拟看门狗滤波器

当 ADC 仅配置一个输入通道（不允许在扫描模式下选择多个通道）时，可以通过 ADC_TR1 寄存器配置有效的 ADC 转换数据间隔：

- 当转换后的数据属于 ADC_TR1 中定义的间隔时，会生成 DMA 请求。
- 否则，RDATA 寄存器在每次转换时都会更新。如果数据超出范围，比 ADC_TR1 的 AWDFILT 位中指定的值高出若干倍，则设置 AWDx 标志，并发出相应的中断。

看门狗 2 和 3 的描述

第 2 和第 3 模拟看门狗更灵活，可以通过对 AWDxCH[20:0]（x=2，3）中的相应位进行编程来保护几个选定的通道。当 AWDxCH[20:0]（x=2,3）的任何位被设置时，相应的看门狗被启用。它们被限制为 8 位的分辨率，并且只有阈值的 8 个 MSB 可以被编程为 HTx[7:0]和 LTx[7:0]。下表描述了如何针对所有可能的分辨率执行比较。

表 13-4 模拟看门狗 2 和 3 比较

分辨率(RES[1:0])	模拟看门狗比较：		描述
	原始转换左对齐	阈值	
00：12bit	DATA[11:4]	LTx[7:0]和 HTx[7:0]	数据[3:0]与比较无关
01：10bit	DATA[11:4]	LTx[7:0]和 HTx[7:0]	数据[3:2]与比较无关
10：8bit	DATA[11:4]	LTx[7:0]和 HTx[7:0]	-
11：6bit	DATA[11:6]，00	LTx[7:0]和 HTx[7:0]	用户必须将 LTx[1:0]和 HTx[1:0]配置为 00

注：有关模拟看门狗比较的更多详细信息，请参阅“增益补偿”一节。

Puya Confidential

ADC_AWDx_OUT 信号输出产生

每个模拟看门狗都与内部硬件信号 ADC_AWDx_OUT（x=看门狗编号）相关联，该信号直接连接到一些片上 timer 的 ETR 输入（外部触发）。请参阅片上 timer 部分，了解如何选择 ADC_AWDx_OUT 信号作为 ETR。

启用相关的模拟看门狗时，ADC_AWDx_OUT 被激活：

- 当被保护转换超出编程阈值时，设置 ADC_AWDx_OUT。
- 当接下来的受保护转换结束且值在编程设定的阈值范围内时，ADC_AWDx_OUT 会复位（但若后续受保护转换超出仍设定阈值，则该信号保持为 1）。
- 当禁用 ADC 时（当设置 ADDIS=0 时），ADC_AWDx_OUT 被复位。请注意，停止规则或注入的转换（设置 ADSTP=1 或 JADSTP=1）对 ADC_AWDx_OUT 的生成没有影响。

注意：AWDx 标志由硬件设置，由软件复位：AWDx 标志对 ADC_AWDx_OUT 的生成没有影响（例如：如果软件未清除 AWDx，AWDx 标志保持为 1，ADC_AWDx_OUT 可以切换）。

ADC_AWDx_OUT 信号由 PCLK 域生成。 AWD 比较在每次 ADC 转换结束时执行。规则、注入 ADC_AWDx_OUT 同理。

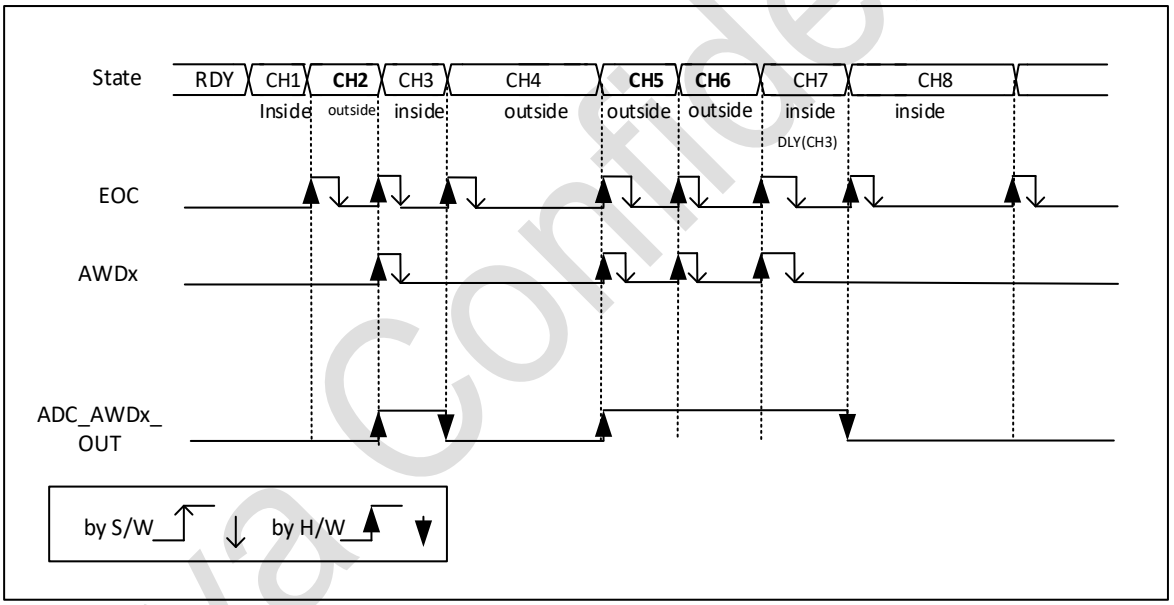


图 13-19 选择所有通道模拟看门狗

Puya Confidential

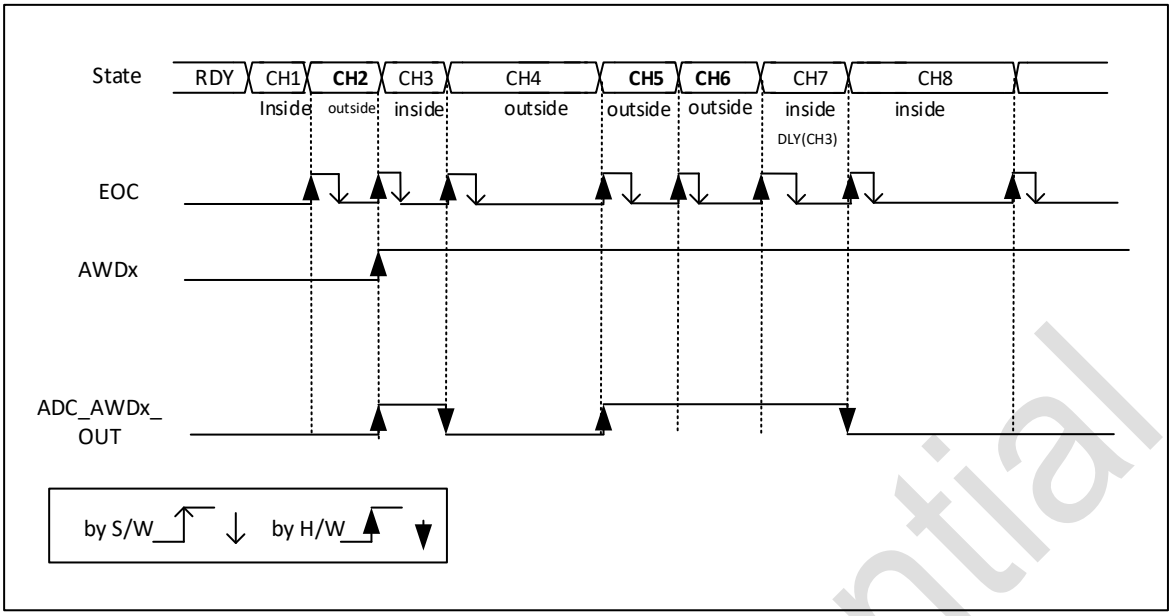


图 13-20 软件不清除 AWDx 的情况

带增益和偏移补偿的模拟看门狗

当增益和偏移补偿被启用时，模拟看门狗在补偿数据之后比较阈值。

注意：启用偏移补偿时（ADC_OFRy 寄存器中的 OFFSETy_EN 设置为 1），数据过载或欠载可能导致错误的看门狗结果。启用过载保护时（在 ADC_OFRy 中将 SATEN 设置为 1），看门狗会提供正确的结果。但是这会阻止使用有符号数据格式。

13.3.15 数据处理

数据对齐

ADC_CFGR 寄存器中的 ALIGN 位用于选择转换后存储的数据的对齐方式，可选择左对齐和右对齐两种方式。转换的数据值已经减去了 ADC_OFRy 寄存器中自定义的偏移量，因此结果可以是一个负值。SEXT 位表示扩展的符号值。

特殊情况：左对齐时，除分辨率为 6 位外，其他分辨率数据以半字为基础对齐。分辨率 6 位时，数据以字节为基础进行对齐。

注意：在过采样模式下不支持左对齐。当 ROVSE 和/或 JOVSE 位被设置时，ALIGN 位值被忽略，ADC 仅提供右对齐数据。

表 13-5 数据对齐和分辨率

ALIGN	RES	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0x0	0x0				DATA[11:0]											
	0x1	0x0						DATA[9:0]									
	0x2	0x0								DATA[7:0]							
	0x3	0x0										DATA[5:0]					
1	0x0	DATA[11:0]												0x0			
	0x1	DATA[9:0]										0x0					
	0x2	DATA[7:0]								0x0							
	0x3	0x0								DATA[5:0]						0x0	

Puya Confidential

偏移（offset）

通过设置 ADC_OFRy 寄存器中 OFFSETy_EN = 1，可以将偏移 y（y = 1、2、3、4）施加到通道。应用可通过 ADC_OFRy 寄存器的位 OFFSETy_CH[4:0]选择偏移的通道。在这种情况下，转换结果减去 OFFSETy[11:0]中的用户定义的偏移量，结果可能是负值，因此读取的数据是有符号的，并且 SEXT 位表示扩展的符号值。

注意：过采样模式下不支持偏移校正。设置 ROVSE 和/或 JOVSE 位时，ADC_OFRy 寄存器中 OFFSETy_EN 位的值被忽略（视为复位）。

下表描述了如何针对模拟看门狗 1 的所有可能的分辨率执行比较。

表 13-6 Offset 计算值 VS 分辨率

分辨 (RES[1:0])	原始转换结果和 offset 减法		Result	描述
	原始转换结果左对齐	Offset		
00:12bit	DATA[11:0]	OFFSET[11:0]	有符号 12bit 数	-
01:10bit	{DATA[11:2],00}	OFFSET[11:0]	有符号 10bit 数	用户必须配置 OFFSET[1:0]=00
10:8bit	{DATA[11:4],0000}	OFFSET[11:0]	有符号 8bit 数	用户必须配置 OFFSET[3:0]=0000
11:6bit	{DATA[11:6],000000}	OFFSET[11:0]	有符号 8bit 数	用户必须配置 OFFSET[5:0]=000000

从与通道“i”对应的 ADC_DR（规则通道）或 ADC_JDRy（注入通道，y=1,2,3,4）读数据时：

- 如果对应通道的偏移之一被使能（位 OFFSETy_EN = 1），则读数据时有符号的。
- 如果此通道的四个偏移均未启用，则读数据无符号。

下图所示为有符号数和无符号数数据对齐。

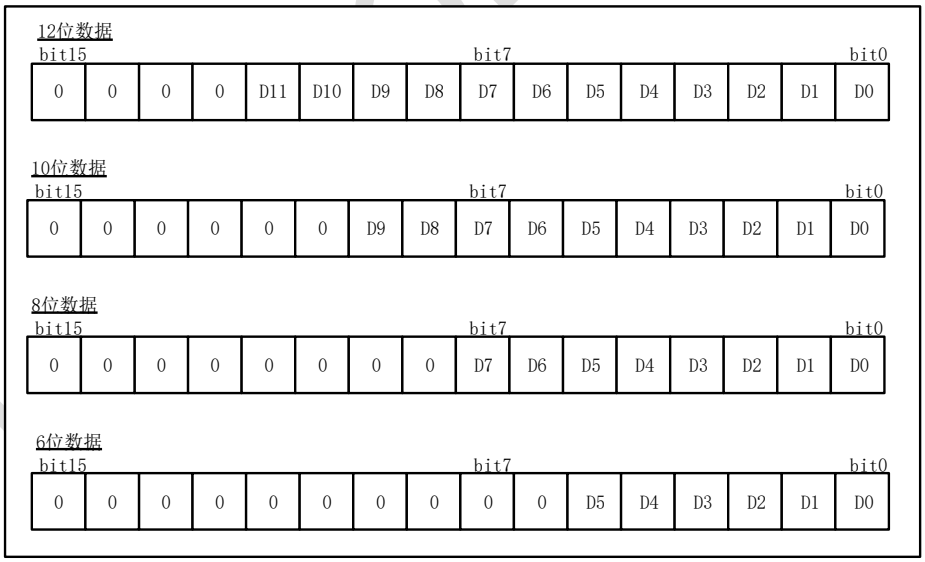


图 13-21 右对齐（offset 未使能，无符号数）

Puya confidential

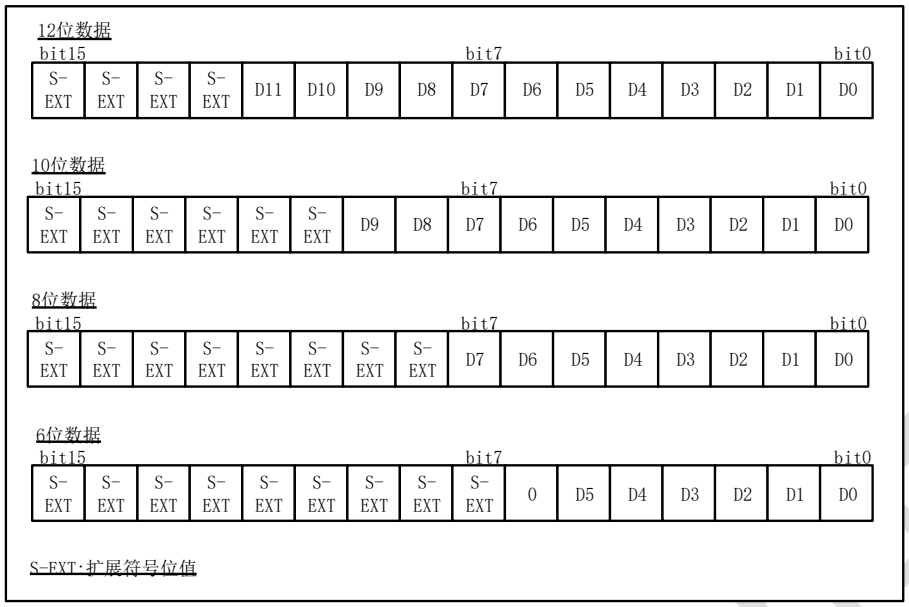


图 13-22 右对齐（offset 使能，有符号数）

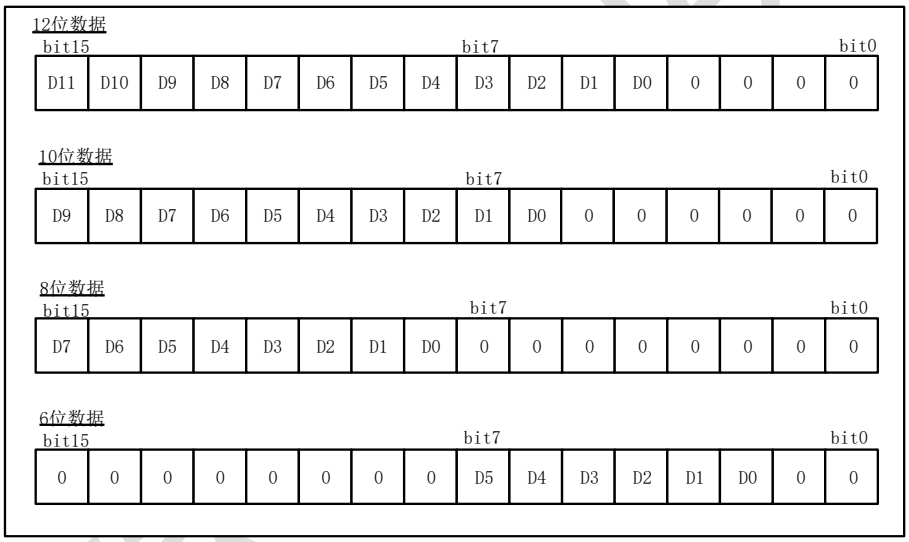


图 13-23 左对齐（offset 未使能，无符号数）

Puya Confidential

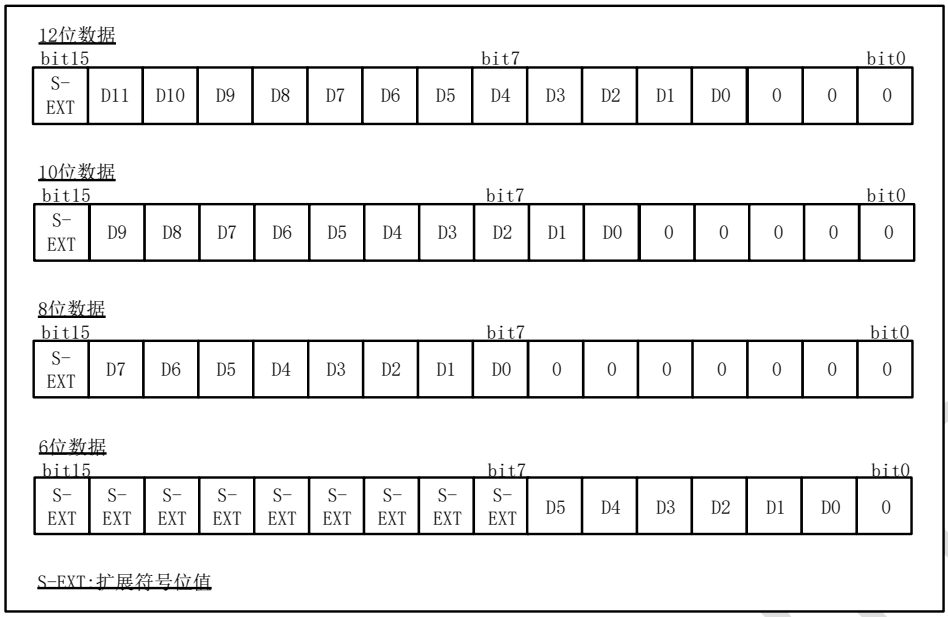


图 13-24 左对齐（offset 使能，有符号数）

增益补偿

当设置 ADC_CFGR2 寄存器 GCOMP 位时，对所有转换后的数据进行增益补偿。每次转换后，数据将使用以下公式进行计算。

DATA = (DATA(adc result) × GCOMPCOEFF)/4095

由于 GCOMPCOEFF 范围为 0-16383，因此实际增益补偿因子范围 0-3.999756。

在将结果数据存储在 RDATA 或 JDATAx 寄存器中之前，将计算 LSB−1 值将四舍五入，并将误差降至最低。

增益补偿对于过采样也是有效的。当增益补偿用于过采样模式时，在累加和右移操作之后执行增益计算，以最小化功耗（增益计算仅进行一次，而不是每次转换）。

偏移（offset）补偿

当在偏移操作期间在 ADC_OFRy 寄存器中设置 SATEN 位时，数据是无符号的。所有偏移数据在 0x000 处饱和（在 12 位分辨率）。当 OFFSETPOS 位被设置时，偏移方向是正的，并且数据在 0xFFF 饱和（在 12 位分辨率）。在 8 位分辨率中，数据分别在 0x00 和 0xFF 处饱和。

在偏移和增益补偿之后，对无符号数执行模拟看门狗比较。为了正确的看门狗操作，偏移补偿后的数据必须为无符号格式（ADC_OFRy 寄存器中的 SATEN 位设置为 1）。

ADC 过载（OVR，OVRMOD）

ADC 过载标志(OVR) 是指一个缓冲区过载事件，当转换好的数据未被 CPU 或 DMA 及时读取时，另一个转换数据已经有效时，就发生了 ADC 过载。

若 EOC 还为‘1’的情况下，这时一个新的转换已经完成，那么 ADC_ISR 寄存器中的 OVR 标志会被置位，表明 ADC 过载。当 ADC_IER 寄存器中的 OVRIE 置位时，产生一个 ADC 过载中断。

当过载事件发生时，ADC 会继续转换除非软件停止并复位这个序列转换，设置 ADC_CR 寄存器中的 ADSTP 为 1 来停止 ADC 转换。OVR 标志可用软件写 1 清除。

当发生过载事件时，可通过对 ADC_CFGR 寄存器中的 OVRMOD 位设置 ADC 数据寄存器中的数据是被保持还是被覆盖：

- OVRMOD=0

Puya Confidential

一个过载事件发生时，数据寄存器的不被覆盖：之前的数据被保持，新的转换数据丢弃。若 OVR 保持为 1，则后续的转换会被执行但结果都被丢弃。

- OVRMOD=1

数据寄存器用最新转换结果覆盖，先前未读的数据丢失。若 OVR 保持为 1，则后续的转换被执行且 ADC_DR 寄存器存放着最新转换的结果值。

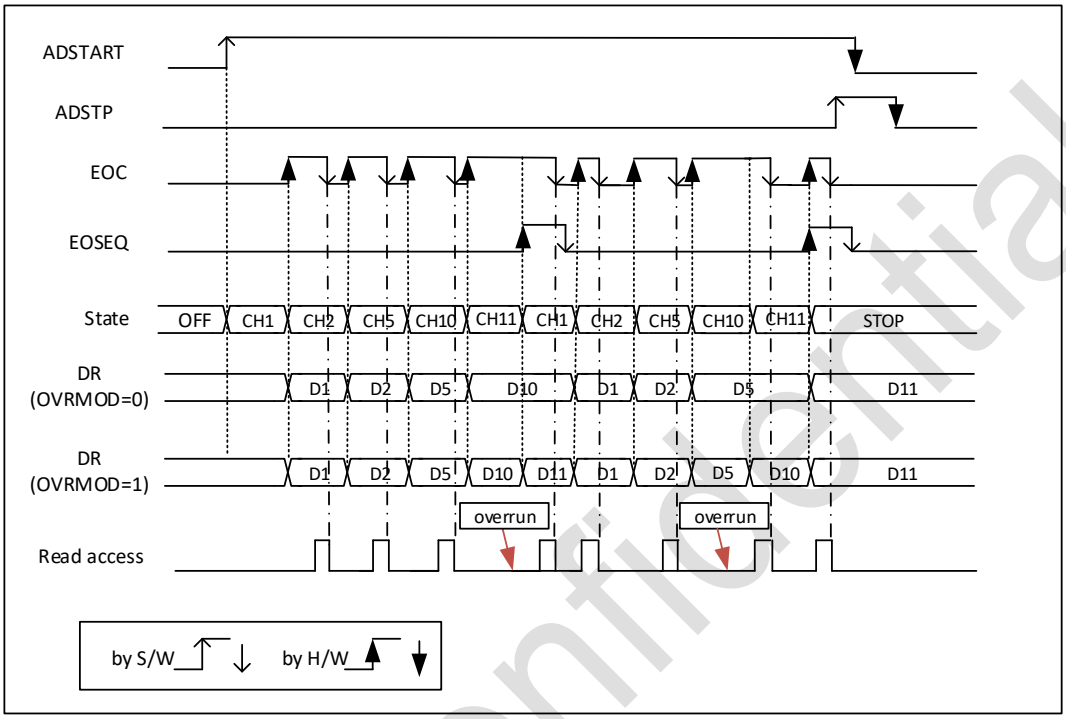


图 13-25 过载示意图

非 DMA 模式

若 ADC 的转换足够慢，转换序列可由软件来控制。这种情况下，软件应用 EOC 标志及其关联的中断去处理每个转换数据。当每次转换结束时，在 ADC_ISR 寄存器中的 EOC 位置位，此时可读 ADC_DR 寄存器的转换值。ADC_CFGR 寄存器中的 OVRMOD 位可配为 0 来管理过载事件。

13.3.16 非过载非 DMA 模式

如需要转换一个或多个通道且不用每次转换结果都要读取，这种情况下，OVRMOD 位必须置为 1 且软件应忽略 OVR 标志。当 OVRMOD=1 时，过载事件不能阻止 ADC 继续转换且 ADC_DR 寄存器中的数据一直为最新转换数据。

13.3.17 DMA 模式

因为所有通道的转换结果数据存放到一个单一的数据寄存器中， 故当转换通道超过 1 个时用 DMA 方式会更有效。这样可以避免丢失存在 ADC_DR 寄存器中的转换结果。当 DMA 模式开启时 (ADC_CFGR 寄存器中的 DMAEN =1)，每次转换结束时都会产生一个 DMA 请求。这样就允许把在 ADC_DR 寄存器中的转换数据传送到软件指定的目标地址中。

尽管如此，因 DMA 不能够及时为 DMA 请求服务而产生的过载 (OVR=1) 时，ADC 就会停止产生 DMA 请求且新的转换数据也不会再由 DMA 进行传输（当 OVR=0 时，会继续传输)。这也可以认为所有传输到 RAM 中的数据都是有效的(因无效的数据再也不传输了)。

根据 OVRMOD 位的配置，ADC_DR 寄存器中的数据可选择为：保持或覆盖。

Puya Confidential

DMA 传输请求会被阻止直到软件清除 OVR 位。

有两种不同的 DMA 模式，其取决于 ADC_CFGR 寄存器中的 DMACFG 位的配置：

- DMA 一次模式（one shot mode）（DMACFG=0）
  当 DMA 编程用于传输固定长度的数据时，可选用该模式。
- DMA 循环模式（circular mode）（DMACFG=1）
  当 DMA 编程为循环模式时，可选用该模式。

DMA 单次模式（DMACFG=0）

在这种模式下，ADC 在每次转换的数据有效时产生一次 DMA 请求。一旦 DMA 已达到最后一个 DMA 传输时，即使 ADC 转换已再次启动，ADC 停止产生 DMA 请求。（产生 DMA_EOT 中断时，下一次的 ADC 转换有可能已开始)

当 DMA 传输完成（配置在 DMA 控制器中的所有传输已经完成）：

- ADC 数据寄存器的内容冻结
- 任何进行中的转换终止，且结果丢弃
- 不给 DMA 控制器发出新的 DMA 请求。假如仍有 ADC 转换启动，这种方式可避免产生一个 ADC 过载错误
- ADC 扫描序列停止并复位
- DMA 停止

DMA 连续模式(DMACFG=1)

在这种模式下，即使 DMA 完成最后一个数据传输，ADC 也会在每次转换的数据有效时产生一次 DMA 请求。这允许 DMA 配置为循环模式来处理连续模拟输入数据流。

13.3.18 过采样器

过采样单元执行数据预处理以卸载 CPU。它能够处理多个转换，并将它们平均为单个数据(增大数据宽度，最高可达 16 位)。

过采样器提供了以下公式，其中 N 和 M 可以调整：

$$Result = \frac{1}{M} \times \sum_{n=0}^{n=N-1} Conversion(t_n)$$

过采样器允许通过硬件执行以下功能：平均、降低数据速率、提高信噪比、基本滤波。

过采样率 N 是通过 ADC_CFGR2 寄存器中的 OVSR[2:0]位定义的，可在 2x 到 256x 的范围内调节。除法系数 M 由高达 8 位的右位移位组成，并且使用 ADC_CFGR2 寄存器中的 OVSS[3:0]位来定义。

求和单元可以产生高达 20 位的结果 (256x 12 位结果) , 该结果首先右移。然后，在最终传输到 ADC_DR 数据寄存器之前被截断为 16 个最低有效位，将移位留下的最低有效位四舍五入到最接近的值。

注意：如果移位后的中间结果超过 16 位，则该结果被原样截断而不饱和。

Puya Confidential

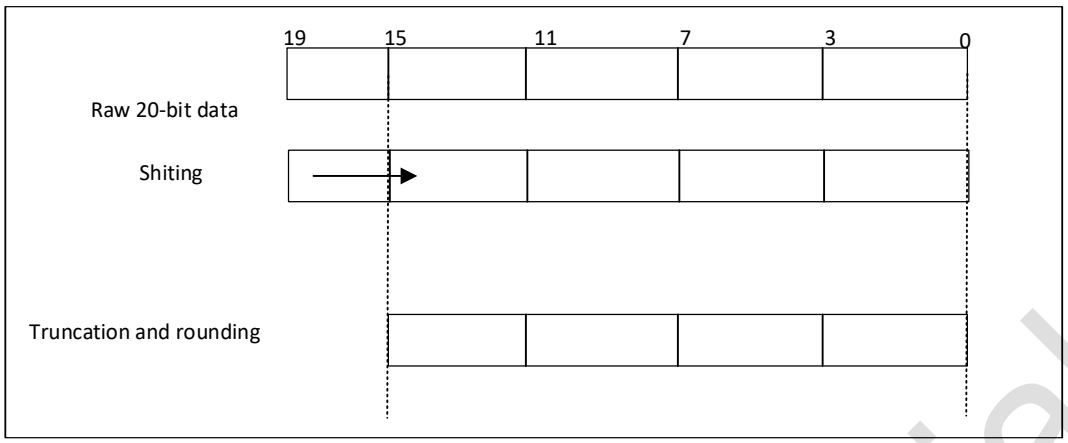


图 13-26 20bit 截断为 16bit

下图给出了处理的数字示例，从原始的 20 位累加数据到最终的 16 位结果。

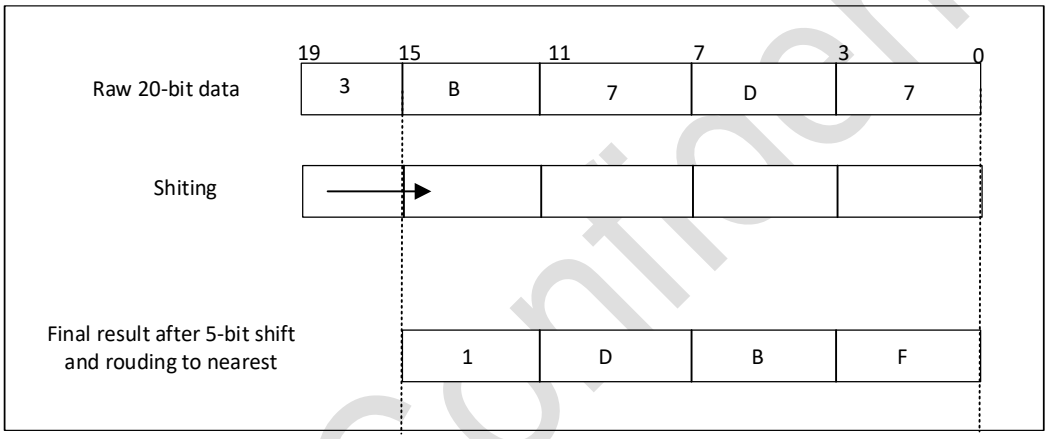


图 13-27 具有 5 位移位和舍入的数值示例

下表给出了原始转换数据为 0xFFF 的各种 N 和 M 组合的数据格式。

表 13-7 最大输出结果与 N 和 M 的关系（灰色单元格表示截断）

Over sam-pling ratio	Max Raw data	No-shift OVSS = 0000	1-bit shift OVSS = 0001	2-bit shift OVSS = 0010	3-bit shift OVSS = 0011	4-bit shift OVSS = 0100	5-bit shift OVSS = 0101	6-bit shift OVSS = 0110	7-bit shift OVSS = 0111	8-bit shift OVSS = 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x020
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

过采样模式下的转换时序没有变化：在整个过采样序列中，采样时间保持相等。每 N 个转换提供一个新数据，通过 N x TCONV = N x（tSMPL+tSAR）等效延迟。标志设置如下：

- 采样阶段结束（EOSMP）设置在每个采样阶段之后

Public Confidential

- 当过采样结果可用时，每 N 次转换发生一次转换结束（EOC）
- 序列结束（EOS）发生在过采样数据序列完成之后（即在 N x 总序列长度转换之后）

过采样时支持的 ADC 操作模式（单个 ADC 模式）

在过采样模式下，大多数 ADC 操作模式都保持不变：

- Single 模式/连续模式转换
- 通过软件或外部触发启动 ADC 转换
- 转换期间停止 ADC（中止）
- 通过 CPU/DMA 读数据
- 低功耗模式（AUTDLY）

可编程分辨率：在这种情况下，减少的转换值（根据 ADC_CFGR 寄存器中的 RES [1:0]位）以与 12 位转换相同的方式进行累加、截断、取整和移位。

注意：使用过采样数据时，数据对齐不可用。ADC_CFGR 中的 ALIGN 位被忽略，并且数据总是以右对齐的方式提供。过采样模式下不支持偏移校正。设置 ROVSE 和/或 JOVSE 位时，ADC_OFRy 寄存器中 OFFSETy_EN 位的值被忽略（视为重置）。

模拟看门狗

保持模拟看门狗功能（AWDSGL 和 AWDEN 位），但有以下区别：

- 忽略 RES[1:0]位，始终使用完整的 12 位值 HT[11:0]和 LT[11:0]进行比较
- 对 16 位过采样结果 ADC_DR[15:4]中的最高有效 12 位执行比较

注意：使用高移位值时必须小心，这会减小比较范围。例如，如果过采样结果被移位 4 位，从而产生 12 位的数据右对齐，则有效的模拟看门狗比较只能在 8 位上执行。在 ADC_DR[11:4]和 HTx[0:7]，LTx[0:7]进行比较，并且 HTx[11:8]，LTx[11:8]必须复位。

触发模式

平均器也可以用于基本的过滤。虽然不是一个非常强大的滤波器（缓慢的滚降和有限的阻带衰减），但它可以用作陷波滤波器来抑制恒定的寄生频率（通常来自电源或开关模式电源）。为此，可以使用 ADC_CFGR2 中的 TROVS 位启用特定的间断模式，以便能够具有由用户定义的过采样频率，并且独立于转换时间本身。

下图显示了在间断模式下如何响应触发启动转换。如果设置了 TROVS 位，则忽略 DISCEN 位的内容并将其视为 1。

Puya Confidential

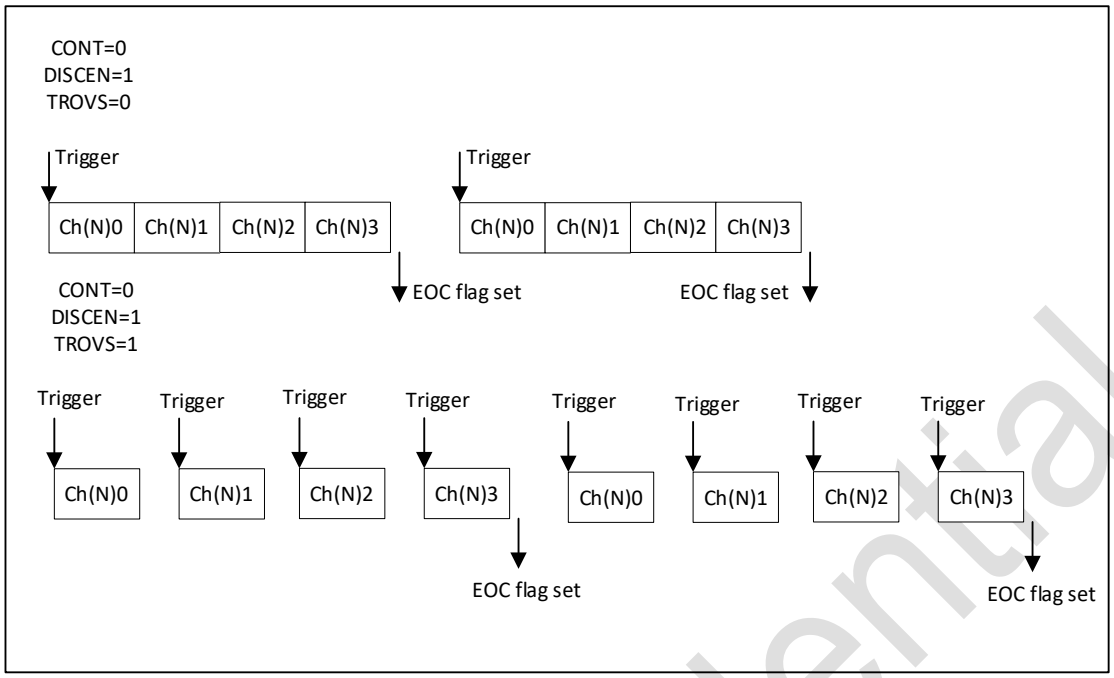


图 13-28 规则的过采样模式触发（TROVS 位=1）

过采样时注入和规则序列管理

在过采样模式下，注入序列和规则序列行为可能有所区别。如果两个序列必须同时使用,可以对它们启用过采样，但有一些限制（这与唯一的累加单元有关）。

仅对规则通道进行过采样

规则过采样模式位 ROVSM 定义了如果规则过采样序列被注入的转换中断，则如何恢复该序列：

- 在连续模式中，累加从最后一个有效数据重新开始（注入的触发而导致之前转换中止)。这确保了无论注入频率如何都将完成过采样（提供至少完成一次规则转换在触发之间)；
- 在恢复模式下，累加从 0 重新开始（忽略以前的转换结果)。该模式允许保证用于过采样的所有数据在单个时隙内背靠背地转换。必须注意使注入触发周期高于过采样周期长度。如果不遵守此条件，则过采样将无法完成，规则序列将被阻塞。

Puya Confidential

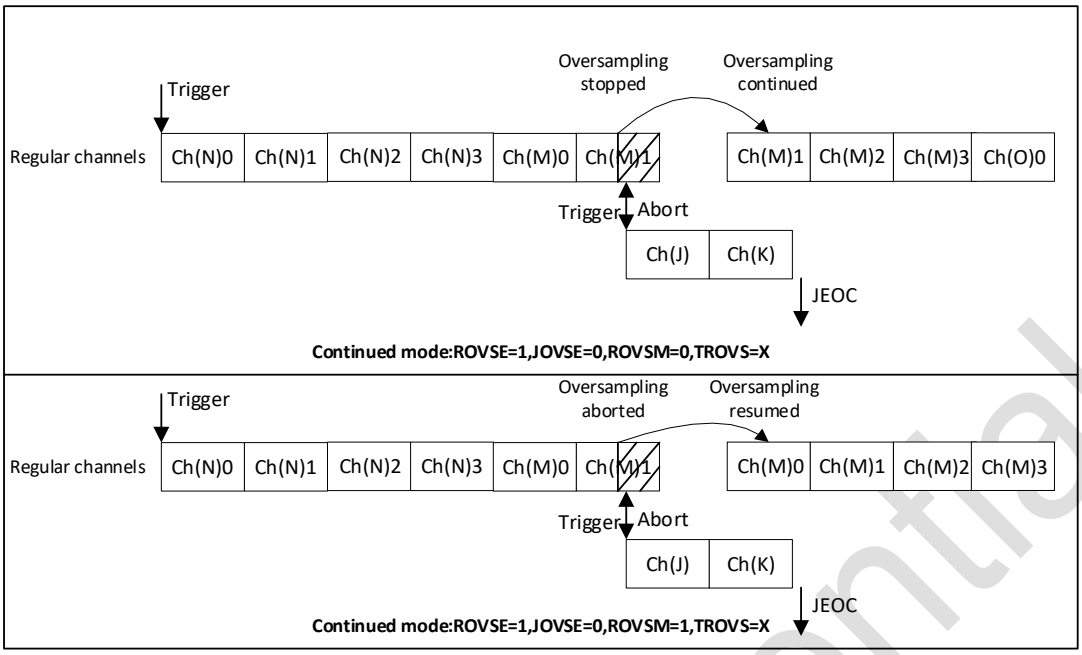


图 13-29 4 倍过采样率模式

仅对注入通道进行过采样

注入过采样模式位 JOVSE 仅为注入序列中的转换启用过采样。

规则通道和注入通道过采样

可以同时设置 ROVSE 和 JOVSE 位。在这种情况下，规则过采样模式被强制为恢复模式（忽略 ROVSM 位），如下图所示

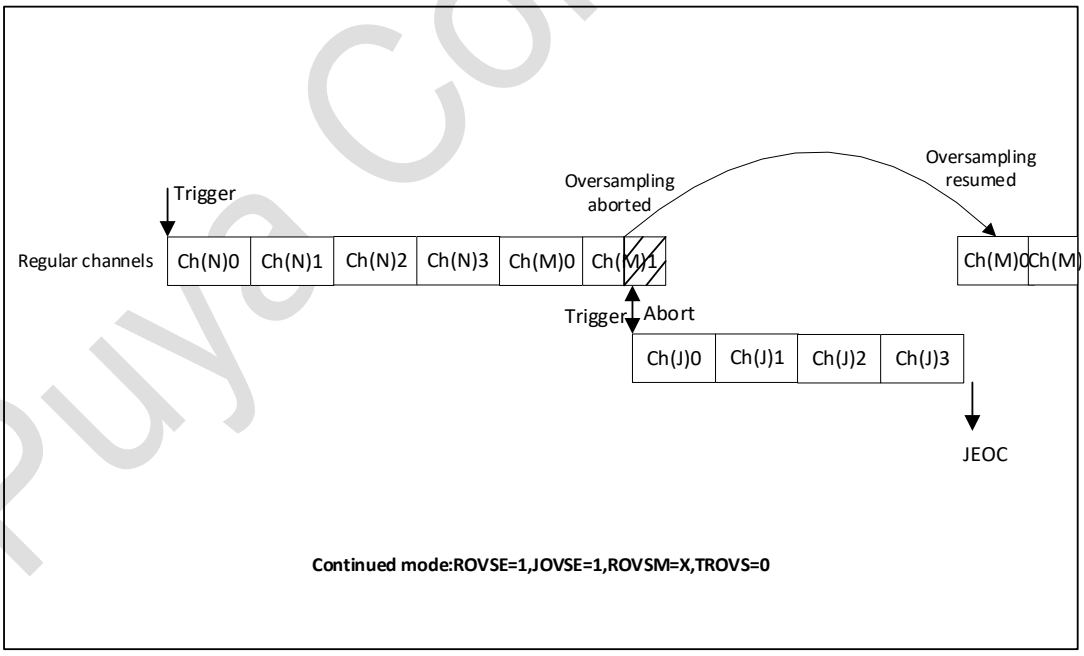


图 13-30 同时使用规则和注入过采样模式

TROVS 规则过采样被注入中断

已经触发转换规则模式被注入转换中断。在这种情况下，必须禁用注入模式过采样模式，并忽略 ROVSM 位（强制恢复模式）。JOVSE 位必须复位。该行为如下图所示。

Puya Confidential

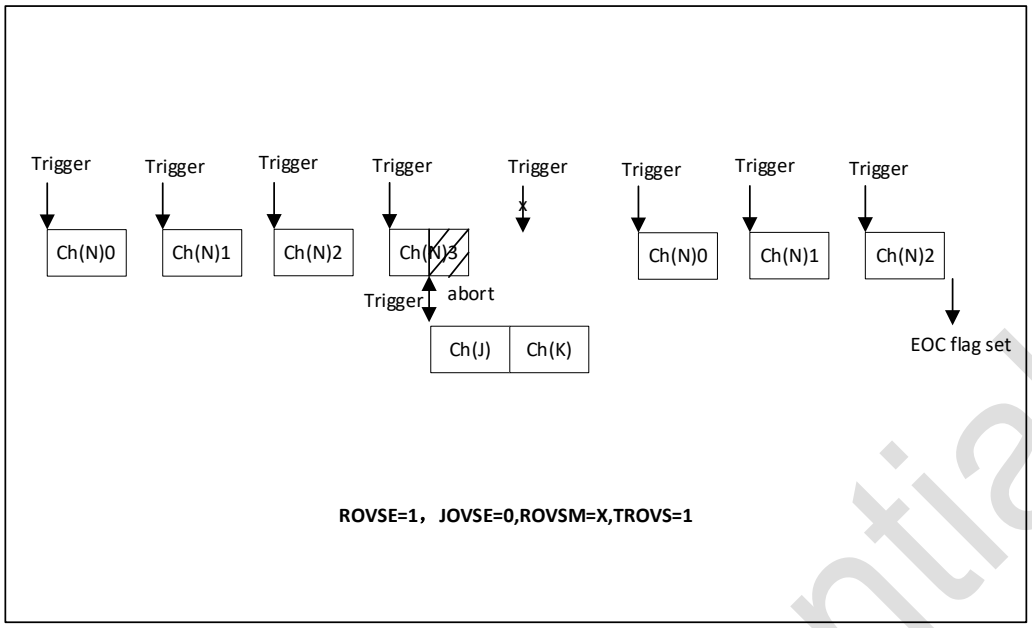


图 13-31 规则过采样被注入触发

自动注入模式

可以对自动注入的序列进行过采样，并将所有转换结果存储在寄存器中，以节省 DMA 资源。此模式仅在规则和注入过采样均使能时可用：JAUTO=1、ROVSE=1 和 JOVSE=1，不支持其他组合。ROVSM 位在自动注入模式下被忽略。下图显示了转换的顺序。

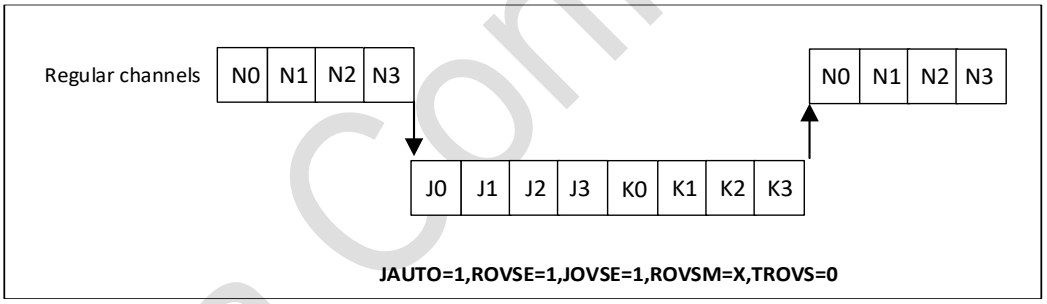


图 13-32 自动注入模式过采样

也可以使用 TROVS 位启用触发模式。在这种情况下，ADC 必须配置如下：JAUTO=1，DISCEN=0，JDISCEN=0，ROVSE=1，JOVSE=1 和 TROVSE=1。（该配置也可与 SCAN 组合，但禁止与 CONT 组合(使能 CONT 会导致功能异常)）。

模式组合小结

下表总结了所有组合，包括不支持的模式。

表 13-8 过采样器模式小结

规则过采样 ROVSE	注入过采样 JOVSE	过采样模式 ROVSM;0=continued 1=resumed	触发规则模式 TROVS	描述
1	0	0	0	Regular continued mode
1	0	0	1	不支持
1	0	1	0	Regular resumed mode
1	0	1	1	Triggered regular resumed

Puya Confidential

				mode
1	1	0	X	不支持
1	1	1	0	Injected and regular resumed mode
1	1	1	1	不支持
0	1	X	X	注入过采样

13.3.19 可编程采样时间

ADC 会在若干个 ADC_CLK 周期内对输入电压进行采样，采样周期数通过 ADC_SMPRx 寄存器中的 SMP[2:0] 位配置。每个通道均可以使用不同的采样时间进行采样。

总转换时间计算如下：

t_{CONV} = （采样时间 + (转换分辨率+0.5) ） x ADC 时钟周期

例如：当 ADC_CLK = 16MHz，分辨率为 12 位，且采样时间为 3.5 个 ADC 时钟周期：

t_{CONV} = (3.5 + (12+0.5)) x ADC 时钟周期= 16 x ADC 时钟周期 = 1 μs

EOSMP 标志位用来表明采样阶段的结束。

对采样时间的限制：

- 对于每个通道，SMP[2:0]位必须按照数据手册中 ADC 特性部分所指定的最小采样时间进行编程。

13.3.20 外部触发转换

转换或转换序列可以由软件或外部事件（例如定时器捕获、输入引脚）触发。如果 EXTEN[1:0]位（用于规则转换）或 JEXTEN[1:0]bit（用于注入转换）不等于 00，则外部事件能够触发具有所选极性的转换。

一旦软件设置了位 ADSTART = 1，规则软件触发选择就有效，而一旦软件设置位 JADSTART = 1，注入的软件触发选择就生效。在转换过程中发生的任何软硬件触发都将被忽略。

- 如果位 ADSTART=0，任何发生的规则硬件触发都会被忽略。
- 如果位 JADSTART=0，任何注入的硬件触发将被忽略。当软件设置 ADSTART=1 时，触发选择有效。

下表提供了 EXTEN[1:0]和 JEXTEN[1:0]值与触发极性之间的对应关系。

表 13-9 规则外部触发配置触发极性

EXTEN[1:0]	触发方式
00	硬件触发检测禁用，软件触发检测启用(上升沿)
01	上升沿检测的硬件触发
10	下降沿检测的硬件触发
11	硬件触发，在上升沿和下降沿都有检测

注意：规则和注入触发的极性不能随时更改

表 13-10 注入外部触发配置触发极性

JEXTEN[1:0]	触发方式
00	硬件触发检测禁用，软件触发检测启用(上升沿)
01	上升沿检测的硬件触发
10	下降沿检测的硬件触发
11	硬件触发，在上升沿和下降沿都有检测

Puya Confidential

表 13-11 ADC12 外部触发事件选择

ADC trigger selection EXTSEL[4:0] or JEXTSEL[4:0]	Regular		Injected	
	Name	Source	Name	Source
0	ADC_EXT_TRG0	tim1_ch1	ADC_JEXT_TRG0	tim1_trgo
1	ADC_EXT_TRG1	tim1_ch2	ADC_JEXT_TRG1	tim1_ch4
2	ADC_EXT_TRG2	tim1_ch3	ADC_JEXT_TRG2	tim2_trgo
3	ADC_EXT_TRG3	tim2_ch2	ADC_JEXT_TRG3	tim2_ch1
4	ADC_EXT_TRG4	tim3_trgo	ADC_JEXT_TRG4	tim3_ch4
5	ADC_EXT_TRG5	tim1_multi_trg2adc	ADC_JEXT_TRG5	tim1_multi_trg2adc
6	ADC_EXT_TRG6	exti11	ADC_JEXT_TRG6	exti15
7	ADC_EXT_TRG7	tim8_trgo	ADC_JEXT_TRG7	tim8_ch4
8	ADC_EXT_TRG8	tim8_trgo2	ADC_JEXT_TRG8	tim1_trgo2
9	ADC_EXT_TRG9	tim1_trgo	ADC_JEXT_TRG9	tim8_trgo
10	ADC_EXT_TRG10	tim1_trgo2	ADC_JEXT_TRG10	tim8_trgo2
11	ADC_EXT_TRG11	tim2_trgo	ADC_JEXT_TRG11	tim3_ch3
12	ADC_EXT_TRG12	tim2_multi_trg2adc	ADC_JEXT_TRG12	tim3_trgo
13	ADC_EXT_TRG13	tim3_multi_trg2adc	ADC_JEXT_TRG13	tim3_ch1
14	ADC_EXT_TRG14	tim8_multi_trg2adc	ADC_JEXT_TRG14	tim2_multi_trg2adc
15	ADC_EXT_TRG15	tim3_ch4	ADC_JEXT_TRG15	tim3_multi_trg2adc
16	ADC_EXT_TRG16	tim1_ch4	ADC_JEXT_TRG16	tim8_multi_trg2adc
17	ADC_EXT_TRG17	tim1_ch5	ADC_JEXT_TRG17	tim1_ch1
18	ADC_EXT_TRG18	lptim_out	ADC_JEXT_TRG18	tim1_ch2
19	ADC_EXT_TRG19	tim8_ch4	ADC_JEXT_TRG19	lptim_out
20	ADC_EXT_TRG20	EPWM1_SOCA	ADC_JEXT_TRG20	EPWM1_SOCA
21	ADC_EXT_TRG21	EPWM1_SOCB	ADC_JEXT_TRG21	EPWM1_SOCB
22	ADC_EXT_TRG22	EPWM2_SOCA	ADC_JEXT_TRG22	EPWM2_SOCA
23	ADC_EXT_TRG23	EPWM2_SOCB	ADC_JEXT_TRG23	EPWM2_SOCB
24	ADC_EXT_TRG24	EPWM3_SOCA	ADC_JEXT_TRG24	EPWM3_SOCA
25	ADC_EXT_TRG25	EPWM3_SOCB	ADC_JEXT_TRG25	EPWM3_SOCB
26	ADC_EXT_TRG26	EPWM4_SOCA	ADC_JEXT_TRG26	EPWM4_SOCA
27	ADC_EXT_TRG27	EPWM4_SOCB	ADC_JEXT_TRG27	EPWM4_SOCB
28	ADC_EXT_TRG28	EPWM5_SOCA	ADC_JEXT_TRG28	EPWM5_SOCA
29	ADC_EXT_TRG29	EPWM5_SOCB	ADC_JEXT_TRG29	EPWM5_SOCB
30	ADC_EXT_TRG30	EPWM6_SOCA	ADC_JEXT_TRG30	EPWM6_SOCA
31	ADC_EXT_TRG31	EPWM6_SOCB	ADC_JEXT_TRG31	EPWM6_SOCB

13.3.21 可配置分辨率

可通过降低 ADC 分辨率来执行快速转换。ADC_CFGR 寄存器的 RES 位用于选择数据寄存器中可用的位数。每种分辨率的最小转换时间如下：

- 12 位：3.5 + 12.5 = 16 ADCCLK 周期
- 10 位：3.5 + 10.5 = 14 ADCCLK 周期
- 8 位：3.5 + 8.5 = 12 ADCCLK 周期
- 6 位：3.5 + 6.5 = 10 ADCCLK 周期

PUYA Confidential

13.3.22 温度传感器和内部参考电压

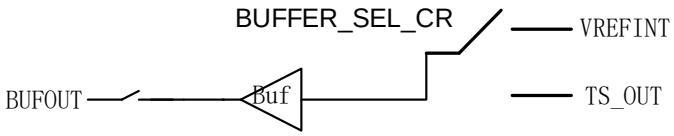


图 13-33 VTS 和 V_{REFINT} 通道复用 BUFFER

温度传感器和内部参考电压可通过复用 1 个 BUFFER 输入到 ADC，可以降低阻抗，减少建立时间，减少采样时间。BUFOUT 通道默认选择温度传感器旁路模式。

温度传感器可用于测量器件结点温度（TJ）。温度传感器内部连接到 ADC1_CH21 通道，此通道把传感器输出的电压转换成数字值。温度传感器不使用时，可将传感器置于掉电模式。

温度传感器输出电压随温度线性变化，由于生产过程的变化，温度变化曲线的偏移在不同芯片上会有不同。内部温度传感器更适合于检测温度的变化，而不是测量绝对的温度。如果需要测量精确的温度，应该使用一个外置的温度传感器。

注意：必须设置 VSENSESEL，VREFEN 激活内部通道：ADC1_CH21 (温度传感器) 和 ADC12_CH18 (V_{REFINT})的转换。

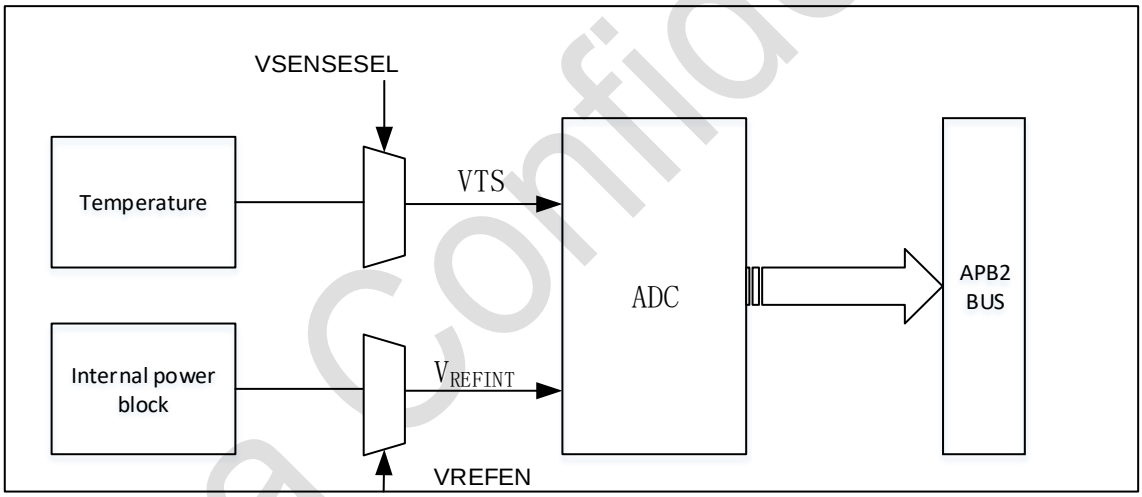


图 13-34 VTS 和 V_{REFINT} 通道

读温度

要使用传感器，请执行以下操作：

1. 选择 ADC1_CH21
2. 选择一个采样时间，该采样时间要大于数据手册中所指定的最低采样时间。
3. 在 ADC_CCR 寄存器中 VSENSESEL 位置 1，以便将温度传感器从掉电模式中唤醒。
4. 通过将 ADEN 位置 1，然后启动 ADC 转换
5. 读取 ADC 数据寄存器中生成的 VTS 数据
6. 使用以下公式计算温度：

$$Temperature(in\ ℃) = \frac{125\ ℃ - 30\ ℃}{TS_{CAL2} - TS_{CAL1}} \times (TS_{DATA} - TS_{CAL1}) + 30\ ℃$$

TS_{CAL2} 代表 125 ℃温度传感器的校准值

Puya Confidential

TS_{CAL1} 代表 30 ℃温度传感器的校准值

TS_{DATA} 是 ADC 转换的实际输出值

注： 传感器从断电模式下唤醒时到能正确输出 VTS 要有一个启动时间，ADC 从上电后启动也有一个启动时间， 若要减少这个延时， 则需要在同一时间的设置 ADEN 和 VSENSESEL 位。

利用内部的参考电压计算实际的 V_{CC} 电压

下面公式可以给出真实 V_{CC} 的电压值：

$$VREFINT = 1.2V = \frac{ADC_DATA}{4095} \times VCC$$

V_{REFINT} 固定值为 1.2 V。

ADC_DATA 是 V_{REFINT} 通道的转换数据。

把一个 ADC 测量到的通道电压相对值转化为绝对电压值

ADC 是根据模拟电源输入和转换通道上的电压比例给出一个数字值。大部分应用是需要把这个比例转换成一个电压值。对于 V_{CC} 可知的情况下并且 ADC 转换值是右对齐的，可用下面公式得到这个绝对电压值：

$$VCHANNEL = \frac{ADC_DATAx}{4095} \times VCC$$

$V_{CHANNEL}$ 是通道电压；

ADC_DATAx 是 ADC_DR 里面的转换数据；

4095 表示为 12 位。

13.3.23 电池监视

由于 V_{CC} 电压可能不准，因此 V_{CC} 引脚需要内部连接到桥接分配器，以确保 ADC 正确运行。

桥接器自动使能将 V_{CC}/3 连接到 ADC12_CH19 输入通道。

13.3.24 双 ADC 模式

双 ADC 模式可用于具有两个或两个以上 ADC 的设备。

在双 ADC 模式下，根据 ADCx_CCR 寄存器中的位 dual[4:0]选择的模式，ADCx 主 ADC 到从 ADC 启动交替或同时触发转换。

实现了四种可能的模式：

- 同步注入模式
- 同步规则模式
- 交错模式
- 交替触发模式

还可以通过以下方式组合使用这些模式：

- 同步注入模式 + 同步规则模式
- 同步规则模式 + 交替触发模式
- 同步注入模式 + 交错模式

在双 ADC 模式下（当 ADCx_CCR 寄存器中的位 dual[4:0]不等于零时），ADC_CFGR 寄存器的位 CONT、AUTDLY、DISCEN、DISCNUM[2:0]、JDISCEN、JAUTO、EXTEN[1:0]、JEXTEN[1:0]、ADSTART、ADSTP(然而，从 ADC 的位 ADSTART 或位 ADSTP 不需要与主 ADC 位同时清除)、JADSTART 、

Puya Confidential

JADSTP（然而，从 ADC 的位 JADSTART 或位 JADSTP 不需要与主 ADC 位同时被清除）被主 ADC 和从 ADC 共享：从 ADC 中的位始终等于主 ADC 的相应位。

要在双 ADC 模式下启动转换，用户必须仅对主 ADC 的 EXTSEL、JEXTSEL 位进行编程，以配置软件或硬件触发以及规则或注入触发。

在双 ADC 模式下，通过读取 ADC 公共数据寄存器（ADCx_CDR），可以并行读取主 ADC 和从 ADC 的转换数据。还可以通过读取双模式状态寄存器（ADCx_CSR）来并行读取状态位。

在双 ADC 模式下，master ADC 和 slave ADC 需要将 ADC_CCR 寄存器中的 MDMA[1:0]，DMACFG，DELAY，DUAL[4:0]配置一致。

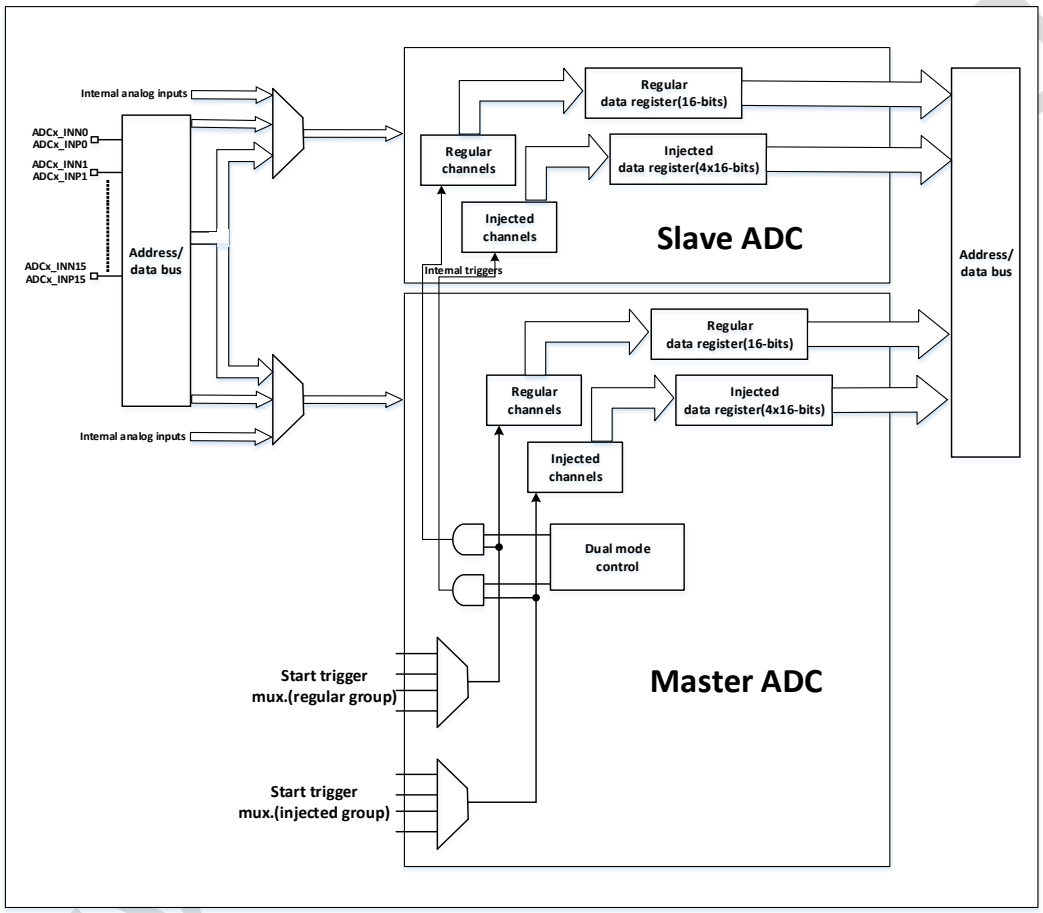


图 13-35 双 ADC 框图

1. Slave ADC 上也存在外部触发，但出于本图的目的未显示。
2. ADC 公共数据寄存器（ADCx_CDR）包含 master ADC 和 slave ADC 规则转换数据。

同步注入模式

该模式由 DUAL[4:0]=00101 选择。该模式转换一组注入通道。外部触发源来自主 ADC 的注入组多路复用器（由 ADC_JSQR 寄存器中的 JEXTSEL 位选择）。

注意：不要在两个 ADC 上转换同一个通道（转换同一通道时，两个 ADC 的采样时间不能重叠）。在同步模式中，必须转换具有相同长度的序列，主/从 ADC 规则序列长度一致 （序列对应的通道不一致，序列的采样时间和分辨率一致）。

注意：独立规则不允许读 ADCx_CDR 寄存器，读各自 ADC 的 ADC_DR 寄存器。

注意：该模式不支持规则软件触发

可以在一个或所有 ADC 上执行规则转换。在这种情况下，它们彼此独立并在注入事件发生时中断，它们在注入的转换组结束时恢复。

- 在主 ADC 上注入转换事件序列（JEOS）结束时，转换后的数据存储到主 ADC_JDRy 寄存器中，并生成 JEOS 中断（如果启用）
- 在从 ADC 上注入的转换事件序列（JEOS）结束时，转换后的数据被存储到从属 ADC_JDRy 寄存器中，并生成 JEOS 中断（如果启用）
- 如果主注入序列的持续时间等于从注入序列的连续时间，则软件可以仅启用两个 JEOS 中断中的一个（例如：主 JEOS），并读取两个转换后的数据（来自主 ADC_JDRy 和从 ADC_JDRy 寄存器）。

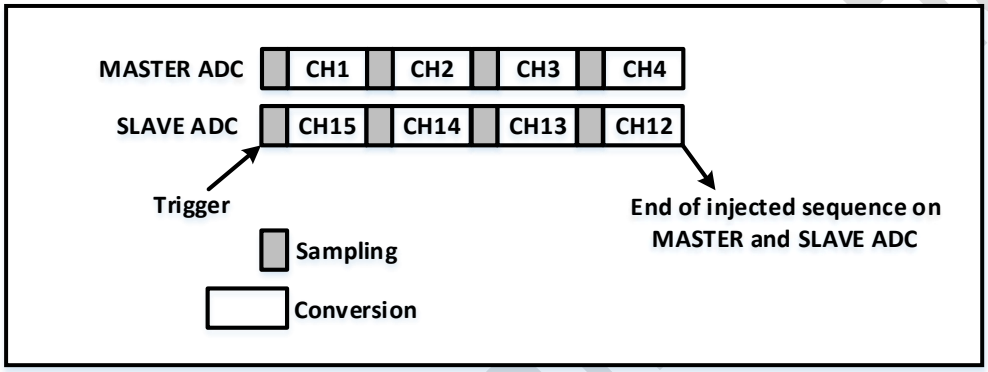


图 13-36 同步注入模式 4 通道：双 ADC 模式

如果 JDISCEN=1，则注入序列的每次同步转换都需要发生注入触发事件。

此模式可以与 AUTDLY 模式组合使用：

- 一旦同步注入转换序列结束，只有在主 ADC 和从 ADC 的 JEOS 位都已被清除的情况下（延迟阶段），才接受新的注入触发事件。忽略所有发生在正在转换的注入序列和相关联的延迟阶段期间的注入触发事件。
- 一旦主 ADC 的规则转换序列已经结束，则只有当主数据寄存器（ADC_DR）已经被读取时，才接受主 ADC 的新的规则触发事件。主 ADC 任何新的规则触发事件发生在规则序列转换期间以及相关的延迟阶段期间都被忽略。对于从 ADC 上的规则序列也是一样。

具有独立注入的同步规则模式

该模式通过 DUAL[4:0]=00110 来选择。

此模式是在一组规则通道上执行的。外部触发源来自主 ADC 的规则组多路复用器（由 ADC_CFGR 寄存器中的 EXTSEL 位选择），同时触发被提供给从 ADC。

在这种模式下，支持独立的注入转换。注入请求（在主设备上或在从设备上）将中止当前的同步转换，一旦注入的转换完成，将重新启动被中断通道。

注意：不要在两个 ADC 上转换同一个通道（转换同一通道时，两个 ADC 的采样时间不能重叠）。在同步模式中，必须转换具有相同长度的序列，主/从 ADC 规则序列长度一致 （序列对应的通道不一致，序列的采样时间和分辨率一致）。

Puya Confidential

当软件收到中断通知时，可以读取数据：

- 在主 ADC 上的每个转换事件（EOC）结束时，生成主 EOC 中断（如果启用了 EOCIE），并且软件可以读取主 ADC 的 ADC_DR。
- 在从 ADC 上的每个转换事件（EOC）结束时，生成从 EOC 中断（如果启用了 EOCIE），并且软件可以读取从 ADC 的 ADC_DR。
- 如果主规则序列的持续时间等于从序列的持续期（如图 14-57 所示），则软件可以仅启用两个 EOC 中断中的一个（例如：主 EOC），并从公共数据寄存器（ADCx_CDR）读取两个转换后的数据。

也可以使用 DMA 来读取规则数据，有两种可行的方法：

- 使用两个 DMA 通道（一个用于主通道，一个用于从通道）。在这种情况下，MDMA[1:0]必须保持清除状态。
  - 配置 DMA 主 ADC 通道去读取主 ADC 的 ADC_DR。在主 ADC 的每个 EOC 事件生成 DMA 请求。
  - 配置 DMA 从 ADC 通道去读取从 ADC 的 ADC_DR。在从 ADC 的每个 EOC 事件生成 DMA 请求。
- 使用 MDMA 模式，使一个 DMA 通道可以自由用于其他用途：
  - 配置 MDMA[1:0]=0b10 或 0b11（取决于分辨率）。
  - 使用单个 DMA 通道（主通道中的一个）。配置 DMA 主 ADC 通道以读取公共 ADC 寄存器（ADCx_CDR）
  - 每次发生主 EOC 事件和从 EOC 事件时，都会生成一个 DMA 请求。此时，从 ADC 转换后的数据在 ADCx_CDR 32 位寄存器的上半字中可用，而主 ADC 转换后数据在 ADC_CCR 寄存器的下半字中可使用。
  - DMA 读取 ADCx_CCR 寄存器时，两个 EOC 标志均被清除。

注意：在 MDMA 模式下（MDMA[1:0]=0b10 或 0b11），用户必须使得主序列中与从序列中相同数量的转换。否则，剩余的转换将不会生成 DMA 请求。

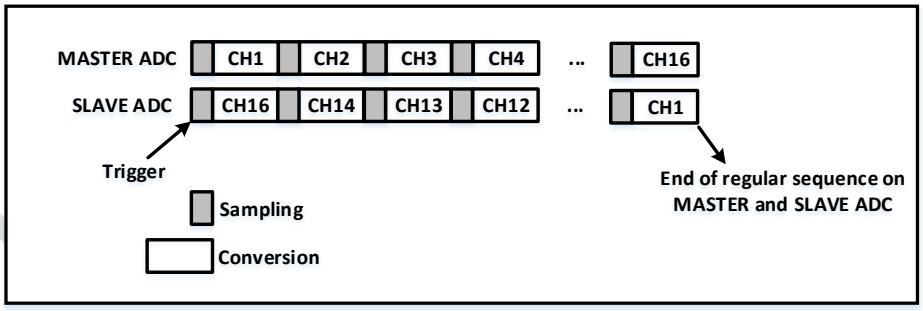


图 13-37 同步规则模式 16 通道：双 ADC 模式

如果 DISCEN=1，则规则序列的每“n”个同步转换都需要一个规则触发事件（“n”由 DISCNUM 定义）。该模式可以和 AUTDLY 模式组合：

- 一旦序列的同步转换结束，只有在公共数据寄存器 ADCx_CDR（或主 ADC 的规则数据寄存器）已被读取（延迟阶段）的情况下，才开始序列中的下一次转换。
- 一旦同步规则转换序列结束，只有在公共数据寄存器（ADCx_CDR）已被读取（延迟阶段）的情况下，才接受新的规则触发事件。在正在进行的规则序列和相关联的延迟阶段期间发生的任何新的规

Confidential

则触发事件都被忽略。

可以使用 DMA 处理数据在同步规则和 AUTDLY 模式组合时，假设使用多 DMA 模式：MDMA 必须设置为 0b10 或 0b11。

当同步规则模式与 AUTDLY 模式相结合时，用户必须确保：

- 主序列中的转换次数等于从序列中的转换次数。
- 对于序列的每次同步转换，从 ADC 的转换的长度小于主 ADC 的转换长度。请注意，序列的长度取决于要转换的通道数量以及每个通道的采样时间和分辨率。

注意：同步规则模式和 AUTDLY 模式的组合仅限于仅对规则通道进行编程的使用情况：禁止在此组合模式中对注入的通道进行编程。

具有独立注入的交错模式

该模式通过 DUAL[4:0]=00111 来选择。

此模式只能在规则组（通常为一个通道）上启动。外部触发源来自主 ADC 的规则通道多路复用器。

在外部触发发生后：

- 主 ADC 立即启动
- 从 ADC 在 delay 几个 ADC 时钟后在主 ADC 采样阶段结束后开始。

在交错模式下区分两个转换的最小延迟配置在 ADCx_CCR 寄存器中的 DELAY 位中。该延迟在主转换的采样阶段结束后的半个周期开始计数。如果互补 ADC 仍在采样它的输入电压(在给定时间内只有一个 ADC 可以对输入信号进行采样)，则另一个 ADC 不能开始转换。

- 最小 DELAY 可以是 1，以确保在主 ADC 采样阶段的模拟开关的打开和从 ADC 采样阶段模拟开关的关闭之间存在至少一个 ADC 时钟周期。
- 最大 DELAY 等于与所选分辨率相对应的 ADC 时钟周期数。然而，用户必须正确地计算该延迟以确保当另一个 ADC 仍在对其输入进行采样时，对应 ADC 不会开始转换。

如果在主 ADC 和从 ADC 上都设置了 CONT 位，则两个 ADC 的所选规则通道将被连续转换。

当软件会收到中断通知，可以在从 ADC 上的每个转换事件 (EOC) 结束时读取数据。当生成从和主 EOC 中断（如果启用了 EOCIE），软件可以读取从/主 ADC 的 ADC_DR。

注意：可以仅启用从 ADC 的 EOC 中断并读取公共数据寄存器（ADCx_CDR）。但在这种情况下，用户必须确保转换的持续时间是兼容的，以确保在序列内部：在新的主转换重新启动之前，主转换之后总是跟随在从转换之后。建议使用 MDMA 模式。

注意:该模式下 Master ADC 不会产生 OVR 标志，Slave ADC 正常产生 OVR 标志，Slave ADC 的 OVR 标志同时屏蔽 Master ADC 和 Slave ADC 的数据更新，MDMA 请求。

注意:该模式下 Master ADC 和 Slave ADC 的 ADSTART 是一致的，在软件触发时只有 Master ADC 的 ADSTART 和 Slave ADC ADSTART 同时清零。

也可以通过 DMA 传输规则数据。在这种情况下，无法使用每个 ADC 上的单独 DMA 请求，必须使用 MDMA 模式，如下所示：

- 配置 MDMA[1:0]=0b10 或 0b11（取决于分辨率）
- 使用单个 DMA 通道（主通道中的一个）。配置 DMA 主 ADC 通道以读取公共 ADC 寄存器（ADCx_CDR）。
- 每次发生主 EOC 事件和从 EOC 事件时，都会生成一个 DMA 请求。此时，从 ADC 转换后的数据

Puya Confidential

在 ADCx_CDR 32 位寄存器的上半字中可用，而主 ADC 转换后数据在 ADC_CCR 寄存器的下半字中可使用。

- DMA 读取 ADCx_CCR 寄存器时，两个 EOC 标志均被清除。

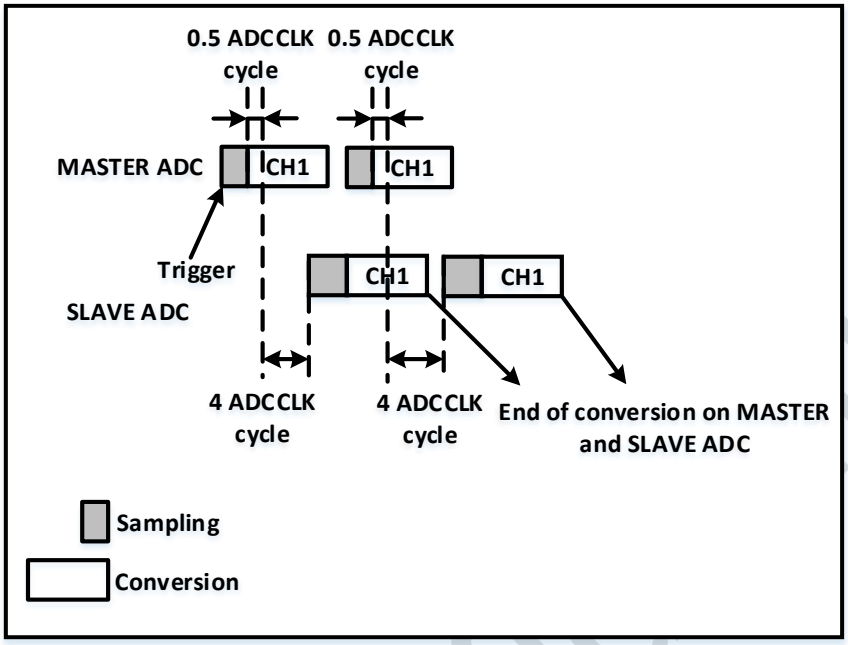


图 13-38 交错模式 1 通道在连续转换模式：双 ADC 模式

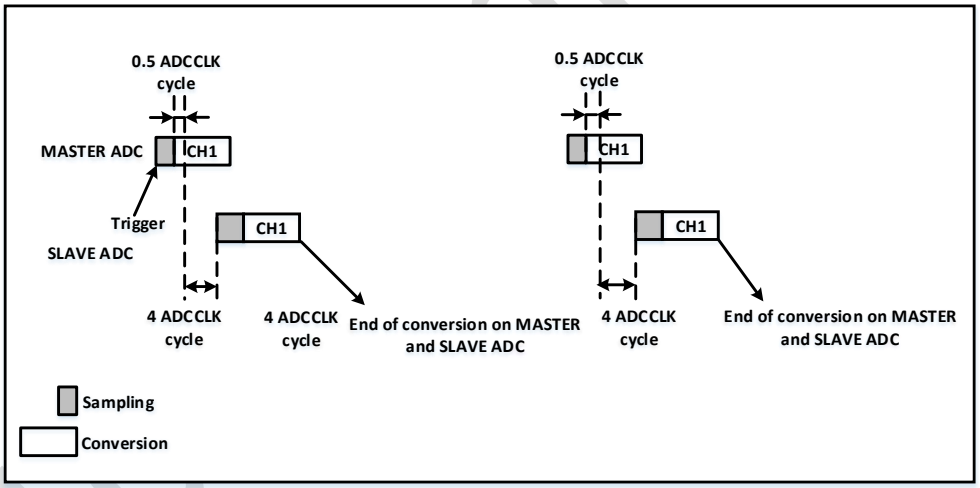


图 13-39 交错模式 1 通道在单次转换模式：双 ADC 模式

如果 DISCEN=1，则规则序列的每“n”个同时转换（“n”由 DISCNUM 定义）都需要发生一个规则触发事件。

在这种模式下，支持注入转换。当注入完成时（在主 ADC 或从 ADC 上），主设备和从设备的规则转换都被中止，序列从主 ADC 重新启动

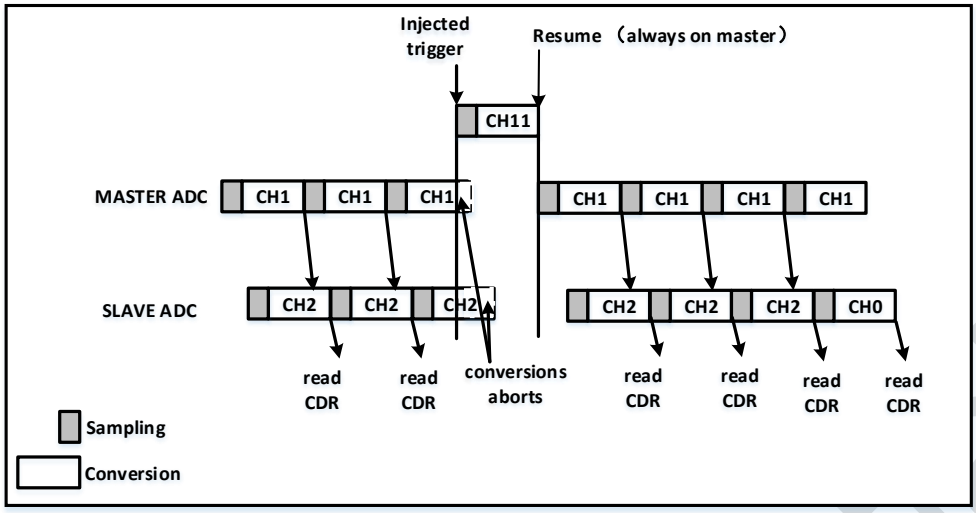


图 13-40 注入中断交错转换

注意: 交错模式（dual=3, dual=7）中非 autdly 模式时，容易发生数据过载，为防止数据过载，用户可根据下式约束 ADC 时钟和 ADC 总线时钟，如下：

分辨率 ADC 时钟 cycle 数 > DELAY ADC 时钟 cycle 数+ 7 ADC 总线时钟+ CPU/DMA 读数据 ADC 总线时钟数

如 12bit 分辨率, DELAY 配置为 0, 读数据总线时钟数 7；则 12/fadc clk > 14 /fpclk ，即 fadc clk < 12/14 fpclk

表 13-12 DELAY bit 和 ADC 分辨率关系(Tadc_ker_ck 即 ADC CLK)

DELAY bits	12 - bit resolution	10 - bit resolution	8 - bit resolution	6 - bit resolution
0	1 * Tadc_ker_ck	1 * Tadc_ker_ck	1 * Tadc_ker_ck	1 * Tadc_ker_ck
1	2 * Tadc_ker_ck	2 * Tadc_ker_ck	2 * Tadc_ker_ck	2 * Tadc_ker_ck
10	3 * Tadc_ker_ck	3 * Tadc_ker_ck	3 * Tadc_ker_ck	3 * Tadc_ker_ck
11	4 * Tadc_ker_ck	4 * Tadc_ker_ck	4 * Tadc_ker_ck	4 * Tadc_ker_ck
100	5 * Tadc_ker_ck	5 * Tadc_ker_ck	5 * Tadc_ker_ck	5 * Tadc_ker_ck
101	6 * Tadc_ker_ck	6 * Tadc_ker_ck	6 * Tadc_ker_ck	6 * Tadc_ker_ck
110	7 * Tadc_ker_ck	7 * Tadc_ker_ck	7 * Tadc_ker_ck	6 * Tadc_ker_ck
111	8 * Tadc_ker_ck	8 * Tadc_ker_ck	8 * Tadc_ker_ck	6 * Tadc_ker_ck
1000	9 * Tadc_ker_ck	9 * Tadc_ker_ck	8 * Tadc_ker_ck	6 * Tadc_ker_ck
1001	10 * Tadc_ker_ck	10 * Tadc_ker_ck	8 * Tadc_ker_ck	6 * Tadc_ker_ck
1010	11 * Tadc_ker_ck	10 * Tadc_ker_ck	8 * Tadc_ker_ck	6 * Tadc_ker_ck
1011	12 * Tadc_ker_ck	10 * Tadc_ker_ck	8 * Tadc_ker_ck	6 * Tadc_ker_ck
其他	12 * Tadc_ker_ck	10 * Tadc_ker_ck	8 * Tadc_ker_ck	6 * Tadc_ker_ck

交替触发模式

该模式通过 DUAL[4:0]=01001 来选择。

此模式只适用于注入通道组。外部触发源来自主 ADC 的注入通道多路器。

此模式仅在选择硬件触发时可用：JEXTEN[1:0]不得为 00

注意：

1. 交替触发模式，软件需要确保没有冗余的无效触发。

Puya Confidential

2. 该模式下的 AUTDLY 模式，必须清除 master ADC 的 JEOS 标志才能响应外部触发

注入间断模式禁用（两个 ADC 的 JDISCEN 均为 0）

1. 当第一次触发发生时，组中所有注入的主 ADC 通道都被转换。
2. 当第二次触发发生时，组中所有注入的从 ADC 通道都被转换。
3. 等等

如果启用了 JEOS 中断，则在组中主 ADC 的所有注入通道转换之后生成 JEOS 中断。

如果启用了 JEOS 中断，则在组中从 ADC 的所有注入通道都已转换后生成。

JEOC 中断（如果启用）在每次注入转换后生成。

如果在该组中的所有注入通道已经被转换之后，又发生一次注入外部触发，则交替触发重新启动转换主 ADC 注入通道组。

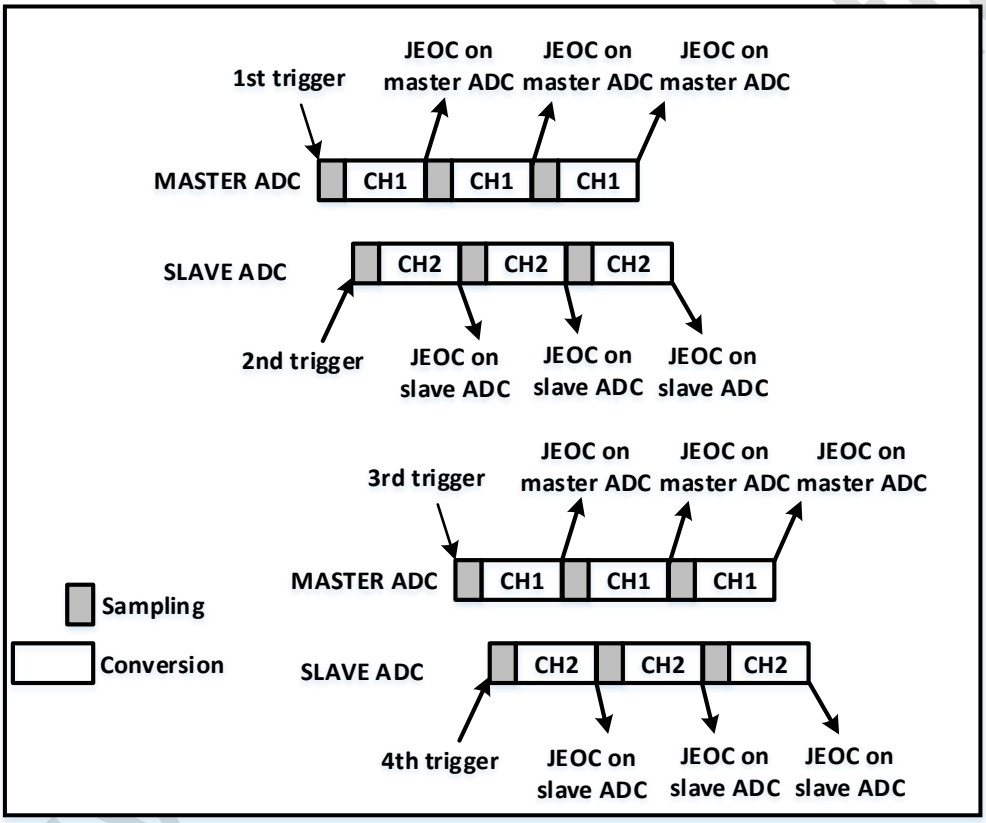


图 13-41 交替触发转换: 每个 ADC 注入组

注意：可以在一个或所有 ADC 上启用规则转换。在硬件触发且 Master ADC 和 Slave ADC 触发源不同时，规则转换是相互独立的。当 ADC 必须执行注入转换时，规则转换被中断，当注入的转换完成时，规则通道将恢复。2 个触发事件之间的时间间隔必须大于或等于 1 个 ADC 时钟周期。在同一 ADC 上启动转换的 2 个触发事件之间的最小时间间隔与单 ADC 模式中的时间间隔相同。

启用注入间断模式（两个 ADC 的 JDISCEN=1）

如果主 ADC 和从 ADC 都启用了注入的间断模式：

- 当第一触发发生时，主 ADC 的第 1 注入通道被转换。
- 当第二次触发发生时，从 ADC 的第 1 个注入通道被转换。
- 等等

如果启用了 JEOS 中断，则在主 ADC 的所有注入通道之后生成 JEOS 中断。

如果启用了 JEOS 中断，则在从 ADC 的所有注入通道都已转换后生成。

如果启用了 JEOC 中断，在每次注入转换后生成。

如果在转换组中所有注入的通道完成后，又发生另一个外部触发，则重新启动交替触发模式。

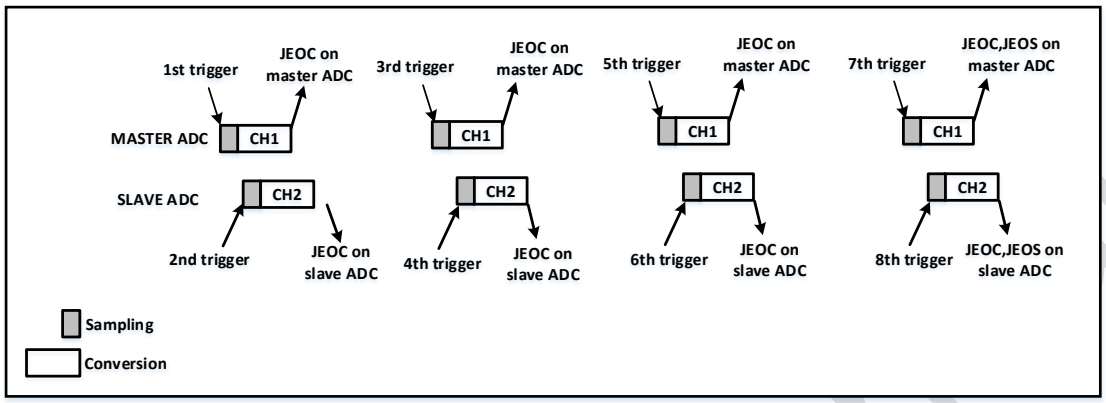


图 13-42 交替触发转换：4 注入通道(每个 ADC)在间断模式

组合的同步规则/注入模式

该模式通过 DUAL[4:0]=00001 来选择。

可以中断同步规则组的同时转换启动同步注入转换。

注意：在组合的同步规则/注入模式中，必须转换具有相同长度的序列，或者确保触发之间的间隔长于两个序列中最长序列转换时间。否则具有最短序列的 ADC 可能重新启动，而具有最长序列的 ADC 正在完成先前的转换。

组合的同步规则+交替触发模式

该模式通过编程位 DUAL[4:0]=00010 来选择。

注意：

1. 组合的同步规则+交替触发交替触发模式，在 master 注入转换期间的注入触发(即使 slave 空闲)也会被忽略，slave 注入转换期间的注入触发同样会被忽略，这是为了保证规则序列同步恢复。
2. JDISCEN 模式时，JEOS flag 是 master 和 slave ADC 注入序列都结束后置位，即等待 slave ADC 注入序列结束后 master ADC 和 slave 同时置位 JEOS；

可以中断规则组的同步转换，以启动注入组的交替触发转换。下图显示了交替触发中断同步规则的行为。注入的交替转换模式在注入触发后立即开始。如果规则转换已经在运行，为了确保注入转换结束后的规则的同步，所有（主/从）ADC 的规则转换被中断时同时停止，并同步恢复。

注意：在组合的同步规则+交替触发模式中，必须转换具有相同长度的序列，或者确保触发之间的间隔长于两个序列中的最长的序列转换时间。否则，具有最短序列的 ADC 可以重新启动，而具有最长序列的 ADC 正在完成先前的转换。

Puya Confidential

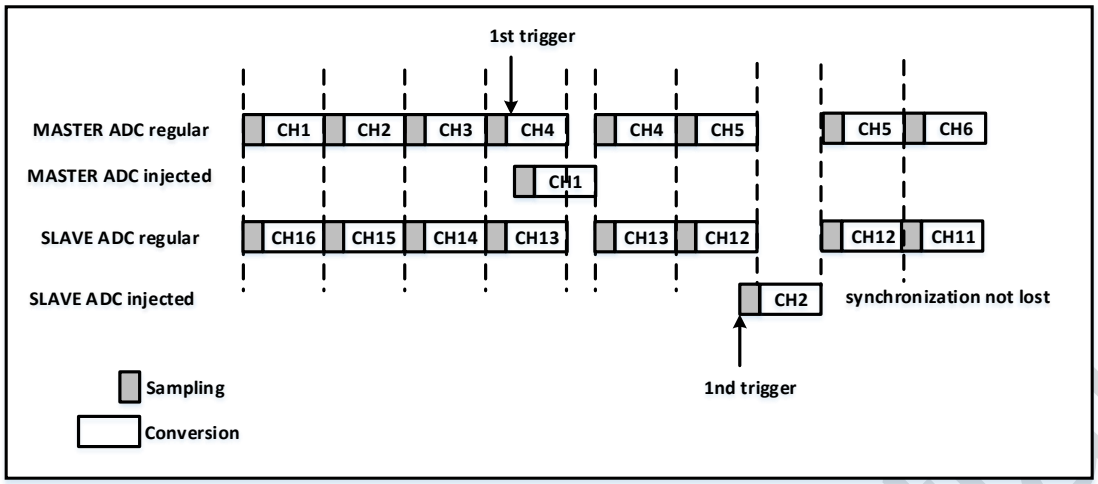


图 13-43 交替触发 + 同步规则

如果在已中断规则转换的注入转换期间发生触发事件，则忽略该触发。图 14-64 显示了这种情况下的行为（注意，第 6 个触发器被忽略，因为相关的交替转换尚未完成）。

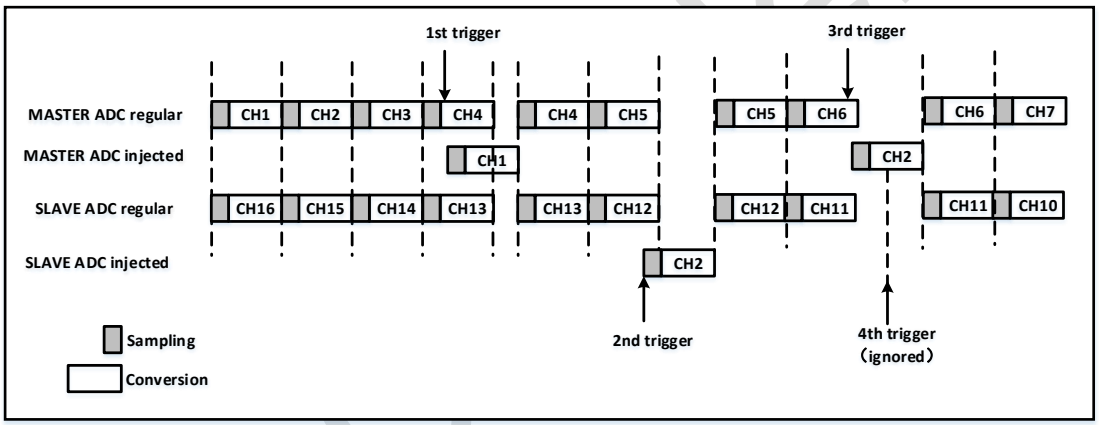


图 13-44 注入转换期间发生触发事件

组合的同步注入+交错模式

该模式由 DUAL[4:0]=00011 选择。

可以通过同步注入事件中断交错模式。在这种情况下，交错模式转换立即中断，同时同步注入转换立即开始。在注入序列结束后，重新开始被中断的交错模式。当规则的交错模式转换恢复时，执行的第一个规则转换总是主 ADC 的规则转换。

警告：在这种模式下，必须通过单次读取公共数据寄存器来读取规则数据，否则不能保证主从数据的一致性。

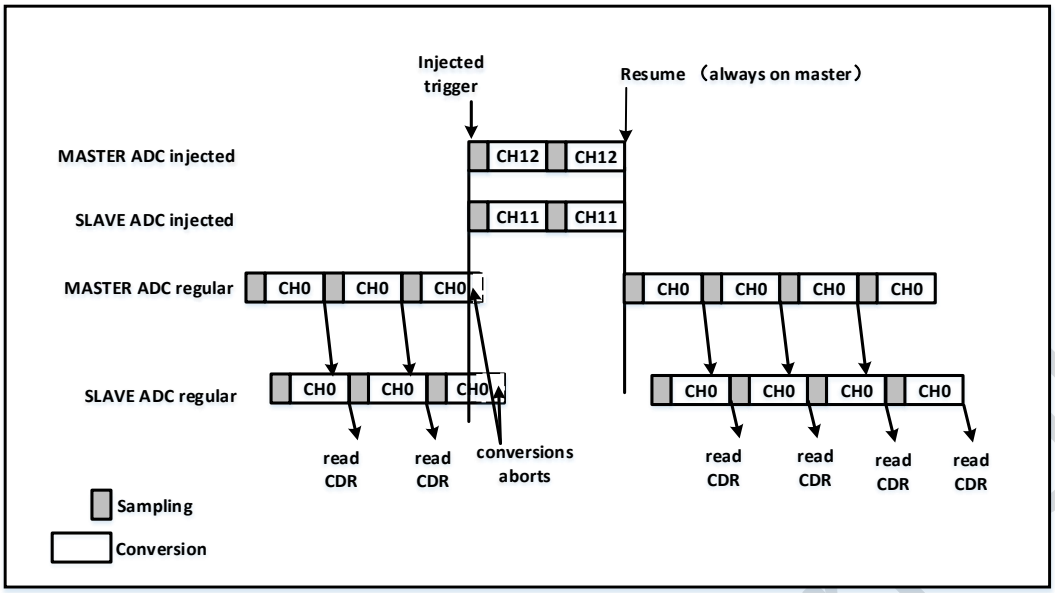


图 13-45 交错单个通道 CH0 注入序列 CH11,CH12

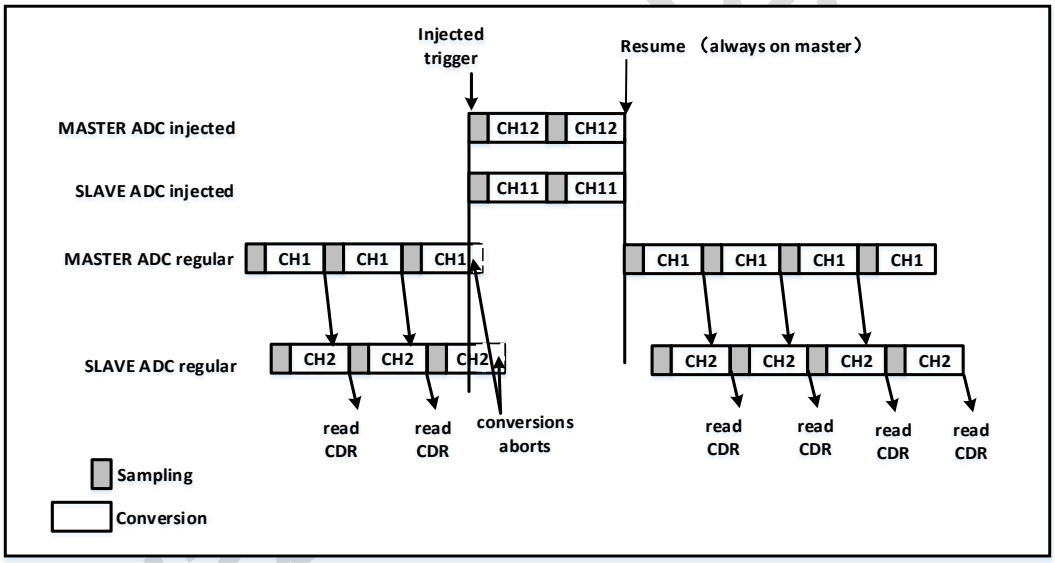


图 13-46 交错两个通道（CH0,CH1）注入序列 CH11，CH12

--Case1：主 ADC 先被中断

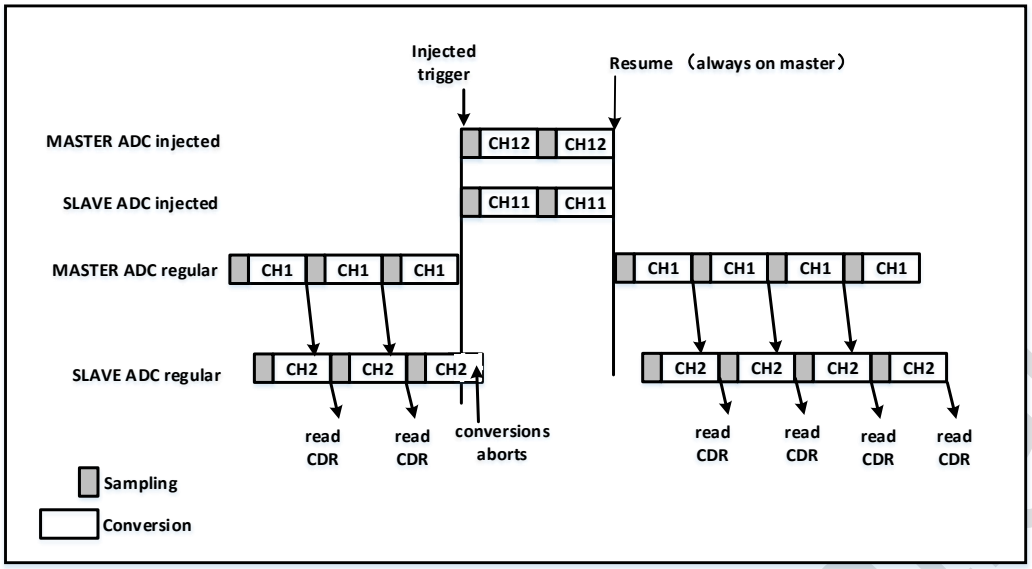


图 13-47 交错两个通道（CH0，CH1）注入序列 CH11,CH12

--Case1: 从 ADC 先被中断

双 ADC 模式 DMA 请求

在所有双 ADC 模式中，可以使用两个 DMA 通道（一个用于主通道，一个用于从通道）来传输数据，就像在 single 模式中一样。

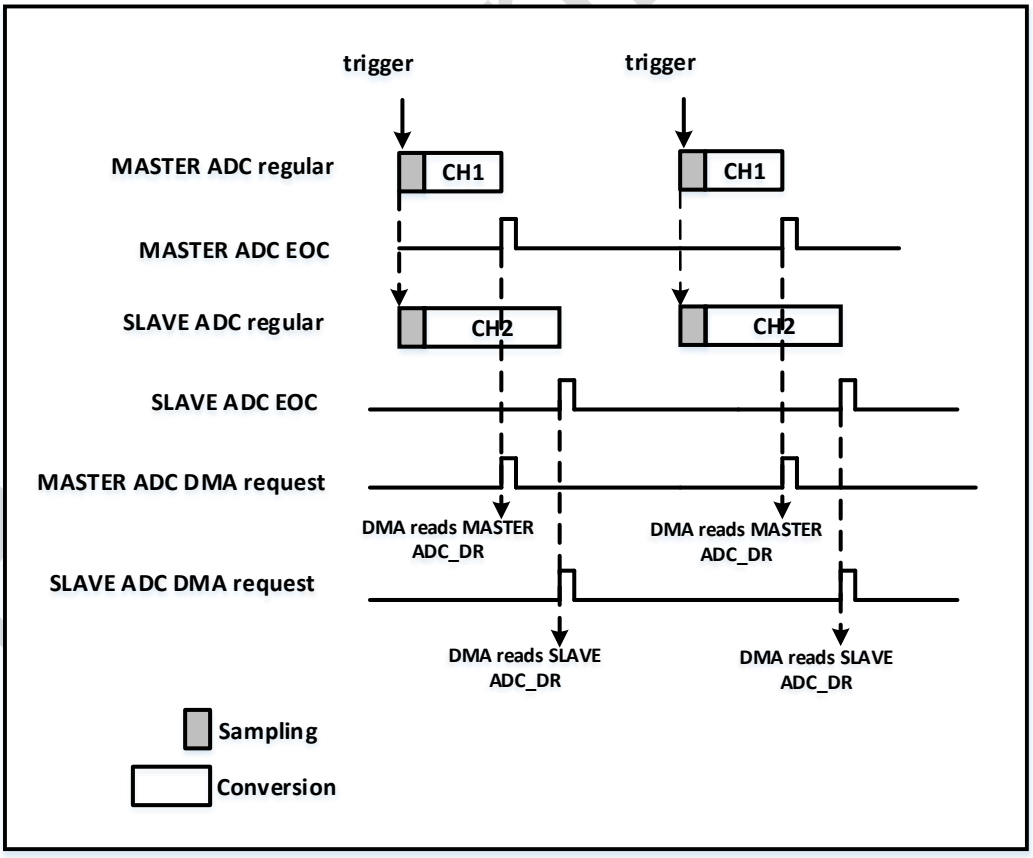


图 13-48 同步规则模式下 DMA 请求当 MDMA=0b00

在同步规则和交错模式中，也可以节省一个 DMA 通道，使用单个 DMA 通道传输两个数据。为此，必须在 ADCx_CCR 寄存器中配置 MDMA 位：

- MDMA=0b10：每次主 EOC 事件和从 EOC 事件发生时，都会生成一个 DMA 请求。此时，两个数据都可用，并且 32 位寄存器 ADCx_CDR 包含表示两个 ADC 转换数据的两个半字。从 ADC 数据在上半字，主 ADC 数据在下半字。当分辨率为 10 位或 12 位时，该模式用于交错模式和同步规则模式。

例如：

交错模式：每次有 2 个数据可用时，生成一个 DMA 请求：

第 1 个 DMA 请求：ADCx_CDR[31:0]=SLV_ADC_DR[15:0]|MST_ADC_DR[15:0]

第 2 个 DMA 请求：ADCx_CDR[31:0]=SLV_ADC_DR[15:0]|MST_ADC_DR[15:0]

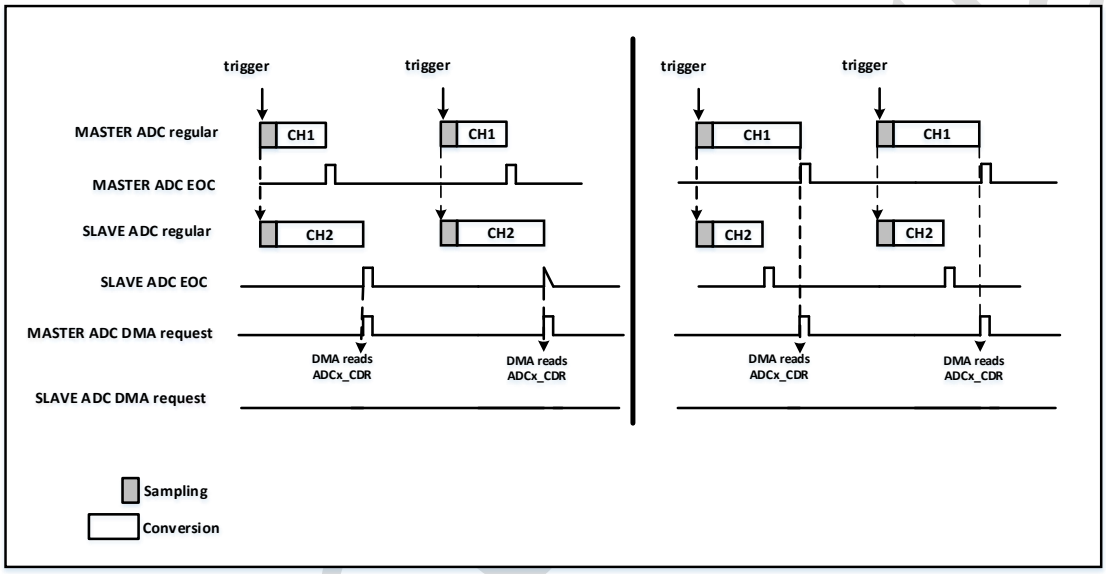


图 13-49 同步规则 DMA 请求当 MDMA=0b10

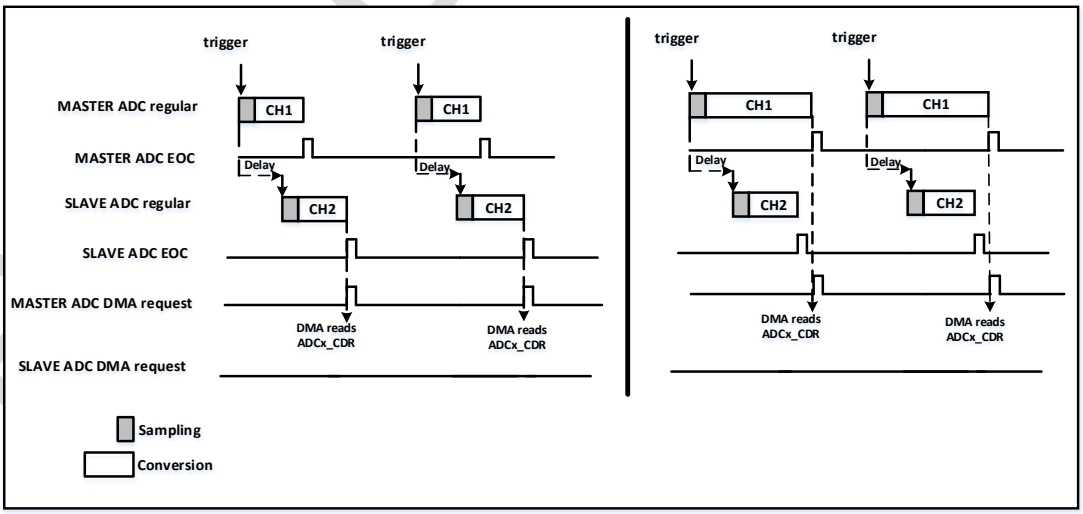


图 13-50 交错模式 DMA 请求当 MDMA=0b10

注意：当使用 MDMA 模式时，用户必须注意正确配置主 ADC 转换和从 ADC 转换的持续时间，以便在新转换执行之前生成 DMA 请求并读取这两个数据（主+从）

- MDMA=0b11：此模式类似于 MDMA=0b10。唯一的区别在于，对于每个 DMA 请求（两个数据可

用），表示两个 ADC 转换数据的两个字节作为半字传输。当分辨率为 6 位或分辨率为 8 位且无符号数（必须禁用所有相关通道的 offset）时，此模式用于交错和同步规则模式。

例子：

交错模式：每次有 2 个数据可用时，生成一个 DMA 请求：

第 1 个 DMA 请求：ADCx_CDR[15:0]=SLV_ADC_DR[7:0]|MST_ADC_DR[7:0]

第 2 个 DMA 请求：ADCx_CDR[15:0]=SLV_ADC_DR[7:0]|MST_ADC_DR[7:0]

过载检测

在双 ADC 模式下（当 dual[4:0]不等于 b00000 时），如果在其中一个 ADC 上检测到过载，则不再发出 DMA 请求，以确保传输到 RAM 的所有数据都是有效的（无论 MDMA 配置如何，都会发生这种行为）。可能发生的情况是，与一个 ADC 相对应的 EOC 位保持置位，因为该 ADC 的数据寄存器具有有效数据。

选择 MDMA 模式时的 DMA 一次性模式/DMA 循环模式

当选择 MDMA 模式（0b10 或 0b11）时，还必须将 ADCx_CCR 寄存器的 DMACFG 选择配置在 DMA 单触发模式还是循环模式，如上述章节中所述：使用 DMA 管理转换（主从 ADC_CFCR 的 DMAFG 不相关）。

双 ADC 模式停止转换

用户必须设置主 ADC 的控制位 ADSTP/JADSTP，停止双 ADC 模式下两个 ADC 的转换。从 ADC 的 ADSTP 在双 ADC 模式下没有影响。

一旦两个 ADC 都被有效地停止，主 ADC 和从 ADC 的 ADSTART/JADSTART 都被硬件清除。

13.4 ADC 中断

ADC 中断可由以下任一事件产生：

- 任何一次的规则转换结束（EOC 标志)
- 规则序列转换结束（EOS 标志)
- 任何一次的注入转换结束（JEOC 标志)
- 注入序列转换结束（JEOS 标志)
- 当模拟看门狗检测发生（AWD 标志)
- 当采样阶段结束发生（EOSMP 标志)
- 当数据过载发生（OVR 标志)
- 上电结束（ADRDY 标志)
- 校准结束（EOCAL 标志）

独自的中断使能位用于灵活设置 ADC 中断。

表 13-13 ADC 中断

中断事件	事件标志	使能控制
规则转换结束	EOC	EOCIE

Puya Confidential

规则序列转换结束	EOS	EOSIE
注入转换结束	JEOC	JEOCIE
注入序列转换结束	JEOS	JEOSIE
模拟看门狗状态置位	AWD	AWDIE
采样阶段结束	EOSMP	EOSMPIE
过载	OVR	OVRIE
AD 上电结束	ADRDY	ADRDYIE
校准结束	EOCAL	EOCALIE

13.5 ADC 寄存器

13.5.1 ADC 中断和状态寄存器（ADC_ISR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	EOCAL	Res.	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
				RC_W1		RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1

Bit	Name	RW	Reset Value	Function
31:12	Reserved	-	-	保留
11	EOCAL	RC_W1	0	校准完成标志 0：校准未启动或者校准未完成 1：校准完成
10	Reserved	-	-	保留
9	AWD3	RC_W1	0	模拟看门狗 3 标志。当转换的电压超过/低于 ADC_TR3 寄存器的字段 LT3[7:0]和 HT3[7:0]中值时，该位由硬件设置。它通过软件写入 1 而被清除。 0：未发生模拟看门狗 3 事件（或软件已确认并清除标志事件） 1：发生模拟看门狗 3 事件
8	AWD2	RC_W1	0	模拟看门狗 2 标志。当转换的电压超过/低于 ADC_TR2 寄存器的字段 LT2[7:0]和 HT2[7:0]中值时，该位由硬件设置。它通过软件写入 1 而被清除。 0：未发生模拟看门狗 2 事件（或软件已确认并清除标志事件） 1：发生模拟看门狗 2 事件
7	AWD1	RC_W1	0	模拟看门狗 1 标志。当转换的电压超过/低于 ADC_TR1 寄存器的字段 LT1[7:0]和 HT1[7:0]中值时，该位由硬件设置。它通过软件写入 1 而被清除。 0：未发生模拟看门狗 1 事件（或软件已确认并清除标志事件）

Puya Confidential

				1：发生模拟看门狗 1 事件
6	JEOS	RC_W1	0	注入序列转换结束标志。当注入序列的所有通道转换结束时，该位由硬件设置。它通过软件写入 1 而被清除。 0：注入序列转换未完成（或标志事件已被软件确认并清除） 1：注入序列转换已完成
5	JEOC	RC_W1	0	注入通道转换结束标志。当每次注入通道转换结束时，新数据可通过相应的 ADC_JDRy 寄存器中访问时，该位由硬件设置。通过软件写入 1 或通过读取相应的 ADC_JDRy 寄存器来清除。 0：注入通道转换未完成（或标志事件已被软件确认并清除） 1：注入通道转换已完成
4	OVR	RC_W1	0	规则通道过载 当过载发生时，硬件置位该位。当 EOC 标志已置起表明一次新的转换已完成。该位写 1 清 0 0：无过载发生（或软件已应答和清除该位） 1：过载已发生
3	EOS	RC_W1	0	规则序列结束标志 规则序列转换结束时硬件置位该位。软件写 1 清 0 0：转换序列没有完成（或者软件已经应答和清除该标志） 1：转换序列完成
2	EOC	RC_W1	0	转换结束标志 当每个规则通道每次转换结束后，新的数据结果可以从 ADC_DR 寄存器读到时，硬件置位该位。软件写 1 清 0 或读 ADC_DR 寄存器清 0 0：通道转换没有完成（或者软件已经应答和清除该标志） 1：通道转换已完成
1	EOSMP	RC_W1	0	采样结束标志，在每次规则通道转换的采样阶段结束时，硬件置位该位，软件写 1 清 0 0：不处在采样阶段结束时（或者软件已经应答和清除该标志） 1：采样阶段结束
0	ADRDY	RC_W1	0	ADC 就绪 ADC 使能后（位 ADEN=1）以及 ADC 达到准备好接收转换请求的状态时，会通过硬件将该位置 1。 通过软件写入 1 可将该位清零。 0：ADC 未准备好开始转换（或标志事件已通过软件确认并清零） 1：ADC 已准备好开始转换

13.5.2 ADC 中断使能寄存器（ADC_IER）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	EOCALIE	Res.	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
				RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:12	Reserved	-	-	保留
11	EOCALIE	RW	0	校准完成中断使能 0：校准完成中断禁止 1：校准完成中断使能
10	Reserved	-	-	保留
9	AWD3IE	RW	0	模拟看门狗 3 标志中断使能。该位由软件设置和清除以使能/禁止模拟看门狗 3 标志中断。 0：禁止模拟看门狗 3 事件中断 1：开启模拟看门狗 3 事件中断
8	AWD2IE	RW	0	模拟看门狗 2 标志中断使能。该位由软件设置和清除以使能/禁止模拟看门狗 2 标志中断。 0：禁止模拟看门狗 2 事件中断 1：开启模拟看门狗 2 事件中断
7	AWD1IE	RW	0	模拟看门狗 1 标志中断使能。该位由软件设置和清除以使能/禁止模拟看门狗 1 标志中断。 0：禁止模拟看门狗 1 事件中断 1：开启模拟看门狗 1 事件中断
6	JEOSIE	RW	0	注入序列转换结束中断使能。该位由软件设置和清除以使能/禁止注入序列转换结束中断。 0：禁止 JEOS 中断 1：使能 JEOS 中断。当 JEOS 置位时，产生一个中断。
5	JEOCIE	RW	0	注入通道转换结束中断使能。该位由软件设置和清除以使能/禁止注入通道转换结束中断。 0：禁止 JEOC 中断 1：使能 JEOC 中断。当 JEOC 置位时，产生一个中断。 注意：只有当 JADSTART = 0 时，软件才被允许写入该位（这确保了没有注入转换正在进行）。
4	OVRIE	RW	0	ADC 规则通道过载中断使能位 软件清除或置起过载中断使能 0：ADC 过载中断不使能 1：ADC 过载中断使能 当 ADSTART=0 时(确保没有任何转换正在进行)软件可以写该位
3	EOSIE	RW	0	规则序列结束中断使能 该位由软件设置和清除以使能/禁止规则序列结束中断 0：禁止 EOS 中断 1：使能 EOS 中断。当 EOS 置位时，产生一个中断。 注意：只有当 ADSTART = 0 时，软件才被允许写入该位（这确保了没有规则转换正在进行）。
2	EOCIE	RW	0	

Puya Confidential

				规则转换结束中断使能 该位由软件设置和清除以使能/禁止规则转换结束中断 0：禁止 EOC 中断 1：使能 EOC 中断。当 EOC 置位时，产生一个中断。 注意：只有当 ADSTART = 0 时，软件才被允许写入该位（这确保了没有规则转换正在进行）。
1	EOSMPIE	RW	0	采样结束中断使能 该位由软件设置和清除以使能/禁止采样结束中断 0：禁止 EOSMP 中断 1：使能 EOSMP 中断。当 EOSMP 置位时，产生一个中断。 注意：只有当 ADSTART = 0 时，软件才被允许写入该位（这确保了没有规则转换正在进行）。
0	ADRDYIE	RW	0	中断使能位 0：ADC 就绪中断不使能 1：ADC 就绪中断使能 当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)软件可以写该位

13.5.3 ADC 控制寄存器（ADC_CR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	Res.	Res.	Res.	RSTCAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
RS				RS											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
										RS	RS	RS	RS	RS	RS

Bit	Name	RW	Reset Value	Function
31	ADCAL	RS	0	ADC 校准启动。 软件设置启动 ADC 校准，校正完成后硬件自动清 0。在校准失败或者校准成功时由硬件清除 0：校准完成 1：写 1 校正 ADC，读为 1 表明校准正在进行 注意：仅当 ADEN=0 时，才允许软件通过设置 ADCAL 来启动校准。仅当 ADEN 无效且无外部触发时（未启用 ADC 且无转换正在进行），才允许软件通过写入 ADC_CALFACT 来更新校准因子
30:28	Reserved	-	-	保留
27	RSTCAL	RS	0	校准复位使能位 该位由软件置位，由硬件清 0。在校准寄存器被初始化后（即 RSTCAL 置 1 后），该位即被清除。 0：校准寄存器已初始化 1：初始化校准寄存器

Puya Confidential

				注：当正在进行转换时，如果设置 RSTCAL，清除校准寄存器需要额外的周期。
26:6	Reserved	-	-	保留
5	JADSTP	RS	0	ADC 停止注入转换命令。 软件置位停止和丢弃正在进行的注入转换（JADSTP 命令）。 当转换被丢弃并且准备接受新的转换命令时硬件会清除该位。 0: 没有正在进行的 ADC 停止注入转换命令 1: 写 1 停止 ADC 注入转换，读为 1 表明一个 JADSTP 命令正在进行中。 软件写该位为 0 为无效操作。
4	ADSTP	RS	0	ADC 停止规则转换命令。 软件置位停止和丢弃正在进行的转换（ADSTP 命令）。 当转换被丢弃并且准备接受新的转换命令时硬件会清除该位。 0：没有正在进行的 ADC 停止转换命令 1：写 1 停止 ADC 规则转换，读为 1 表明一个 ADSTP 命令正在进行中。 软件写该位为 0 为无效操作。
3	JADSTART	RS	0	ADC 触发启动注入转换 该位由软件设置，用于启动注入通道的 ADC 转换。根据 EXTEN[1：0]的配置来决定转换是软件立即启动，还是由硬件触发事件来启动。 0：复位状态 1：开始转换注入通道 该位由硬件清除的情况： – 在单次转换模式，选择软件驱动时(JEXTEN=00)：序列转换完成时（JEOS 标志置位） – 其他情况下：执行 JADSTP 命令之后，同时 JADSTP 标志又被硬件清 0 之时; 0: 没有正在进行的 ADC 转换 1: 写 1 启动 ADC 注入转换，读为 1 表明 ADC 正在操作，可能正在转换。 注：软件只有当 ADEN=1 且 ADDIS=0 时才能配置 JADSTART=1. 软件写该位为 0 为无效操作。
2	ADSTART	RS	0	ADC 启动命令。 软件置位该位启动 ADC 转换。根据 EXTEN[1：0]的配置来决定转换是软件立即启动，还是由硬件触发事件来启动。 该位由硬件清除的情况： – 在单次转换模式 (CONT=0, DISCEN=0), 选择软件驱动时(EXTEN=00)：序列转换完成时（EOSEQ 标志置位） – 在非连续转换模式 (CONT=0, DISCEN=1),当软件驱动时(EXTEN=00)：转换结束标志（EOC 标志置位）

Puya Confidential

				– 其他情况下：执行 ADSTP 命令之后，同时 ADSTP 标志又被硬件清 0 之时 0: 没有正在进行的 ADC 转换 1: 写 1 启动 ADC 规则转换，读为 1 表明 ADC 正在操作，可能正在转换。 注：软件只有当 ADEN=1 且 ADDIS=0 时才能配置 ADSTART=1. 软件写该位为 0 为无效操作。
1	ADDIS	RS	0	ADEN 禁止使能。 软件置位禁止 ADC。当 ADC 被禁止（ADEN 被硬件清零同时），硬件清除该位。软件写该位为 0 为无效操作。 0：没有 ADDIS 1: 写 1 禁止 ADC，读 1 表示 ADDIS 指令正在执行 注：设置 ADDIS 为 1 有效只能在 ADEN=1 并且 ADSTART=0 时（确保没有转换进行）。
0	ADEN	RS	0	开/关 A/D 转换器 ADC 转换器工作使能，当置 1 后 ADC 转换器唤醒，在转换器上电到开始转换有一个延迟 t_{STAB}。当置 0 后 ADC 转换器处于掉电状态。 0：禁止 ADC 转换，ADC 转换器进入断电模式 1：使能 ADC 转换器。

13.5.4 ADC 配置寄存器（ADC_CFGR）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	Res.	JDISCEN	DISCNUM[2:0]			DISCEN
	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALIGN	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL[4:0]					RES[1:0]		Res.	DMACFG	DMAEN
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW

Bit	Name	RW	Reset Value	Function
31	Reserved	-	-	保留
30:26	AWD1CH[4:0]	RW	5’h0	模拟看门狗 1 通道选择位 该位由软件写设置，用于选择模拟看门狗的输入通道。 00000：ADC 模拟输入通道 0 00001：ADC 模拟输入通道 1 …… 01111：ADC 模拟输入通道 15 10000：ADC 模拟输入通道 16 10001：ADC 模拟输入通道 17 10010: ADC 模拟输入通道 18 10011: ADC 模拟输入通道 19 10100: ADC 模拟输入通道 20

Confidential

				保留所有其他数值。 注：模拟看门狗 1 通道选择对应的 ADC 模拟输入通道与 ADC 通道选择一致
25	JAUTO	RW	0	自动注入使能 该位用于开启或关闭规则通道组转换结束后自动进行注入通道组转换 0：关闭自动的注入通道组转换 1：开启自动的注入通道组转换
24	JAWD1EN	RW	0	注入通道模拟看门狗 1 使能。在注入通道上开启模拟看门狗。 0：在注入通道上禁用模拟看门狗 1 1：在注入通道上使用模拟看门狗 1
23	AWD1EN	RW	0	规则通道模拟看门狗 1 使能。在规则通道上开启模拟看门狗。 0：在规则通道上禁用模拟看门狗 1 1：在规则通道上使用模拟看门狗 1
22	AWD1SGL	RW	0	单一通道看门狗 1 使能。 该位用于开启或关闭由 AWD1CH[4:0]定义的通道上的模拟看门狗 1。 0：在所有的通道上使用模拟看门狗 1 1：在单一通道上使用模拟看门狗 1
21	Reserved	-	-	保留
20	JDISCEN	RW	0	注入通道间断模式使能。 该位用于开启或关闭注入通道组上的间断模式。 0：注入通道组上禁用间断模式 1：注入通道组上使用间断模式
19:17	DISCNUM[2:0]	RW	3’h0	间断模式通道计数 这些位用于定义在接收到外部触发后，在间断模式下要转换的规则通道的数量。 000:1 通道 001:2 通道 … 111:8 通道
16	DISCEN	RW	0	规则通道的间断模式使能 该位用于开启或关闭规则通道组上的间断模式 0：规则通道组上禁用间断模式 1：规则通道组上使用间断模式
15	ALIGN	RW	0	数据对齐控制位 0：右对齐 1：左对齐
14	AUTDLY	RW	0	自动延迟转换模式 此位由软件设置和清除，以启用/禁用自动延迟转换模式。 0：自动延迟转换模式关闭 1：自动延迟转换模式打开

13	CONT	RW	0	连续转换使能 如果设置了此位，则转换将连续进行直到该位被清除。 0：单次转换模式 1：连续转换模式
12	OVRMOD	RW	0	过载管理模式。 软件可设置和清除该位，配置数据过载管理的方式。 0：当过载发生时，ADC_DR 寄存器保留旧值 1：当过载发生时，ADC_DR 寄存器会被上一次转换结果覆盖掉 注：仅当 ADSTART=0 时（确保没有正在进行的转换）允许软件写这些位。
11:10	EXTEN[1:0]	RW	2’h0	规则通道的外部触发启用和极性选择 这些位由软件设置和清除，以选择外部触发极性并启用规则组的触发。 00：禁用硬件触发检测（可以通过软件启动转换） 01：上升沿的硬件触发检测 10：下降沿的硬件触发器检测 11：上升沿和下降沿都有硬件触发检测 注意：只有确保没有规则转换正在进行，软件才允许写入这些位。
9:5	EXTSEL[4:0]	RW	5’h0	规则通道外部触发事件选择位。选择启动规则通道组转换的外部事件 ADC 外部触发事件如下：（详情见上述章节） 00000：事件 1 00001：事件 2 00010：事件 3 00011：事件 4 … 10000：事件 16 10001：事件 17 其它：预留
4:3	RES[1:0]	RW	2’h0	分辨率（Resolution）控制位。通过软件写入这些位可选择转换的分辨率。 00：12 位 01：10 位 10：8 位 11：6 位
2	Reserved	-	-	保留
1	DMACFG	RW	0	DMA 访问配置。 软件可设置和清除该位，在两种 DMA 模式操作中选择并在 DMAEN = 1 时有效。 0：DMA 单次模式选择 1：DMA 循环模式选择

				注：仅当 ADSTART=0 时（确保没有正在进行的转换）允许软件写这些位。
0	DMAEN	RW	0	DMA 使能位 0：禁止 DMA 模式 1：使能 DMA 模式

13.5.5 ADC 配置寄存器 2（ADC_CFGR2）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CALNUM[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GCOMP
	RW	RW	RW												RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]				OVSR[2:0]			JOVSE	ROVSE
					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31	Reserved	-	-	保留
30:28	CALNUM[2:0]	RW	3'h0	校准平均次数 000：1 次校准和平均 001：2 次校准和平均 010：4 次校准和平均 011：8 次校准和平均 100：16 次校准和平均 101：32 次校准和平均 其它：保留
27:17	Reserved	-	-	保留
16	GCOMP	RW	0	增益补偿模式 此位由软件设置和清除，以启用增益补偿模式。 0：增益补偿未启用在 ADC 工作模式 1：增益补偿已启用并应用于所有通道 注意：只有当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)，软件才允许写入此位
15:11	Reserved	-	-	保留
10	ROVSM	RW	0	规则过采样模式 该位由软件设置和清除，以选择规则过采样模式。 0：继续模式：触发注入转换时，过采样暂时停止，并在注入序列后继续（注入序列期间保持过采样缓冲） 1：恢复模式：触发注射转换时，当前过采样被中止，并在注入序列后从开始恢复（过采样缓冲区在注入序列开始转换时清零） 注意：只有当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)，软件才允许写入此位
9	TROVS	RW	0	触发规则过采样 此位由软件设置并清除，以启用触发过采样

Puya Confidential

				0：通道的所有过采样转换在触发后连续完成 1：通道的每个过采样转换都需要一个新的触发 注意：只有当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)，软件才允许写入此位
8:5	OVSS[3:0]	RW	4’b0	过采样移位 该位字段由软件设置和清除，以定义应用于原始过采样结果的右移。 0000：无移位 0001：移位 1 位 0010：移位 2 位 0011：移位 3 位 0100：移位 4 位 0101：移位 5 位 0110：移位 6 位 0111：移位 7 位 1000: 移位 8 位 其他：预留 注意：只有当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)，软件才允许写入此位。
4:2	OVSR[2:0]	RW	3’h0	过采样率 此位字段由软件设置并清除，以定义过采样率。 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x 注意：只有当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)，软件才允许写入此位
1	JOVSE	RW	0	注入过采样使能 该位由软件设置和清除，以启用注入过采样。 0：禁用注入过采样 1：启用注入过采样 注意：只有当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)，软件才允许写入此位
0	ROVSE	RW	0	规则过采样使能 启用此位由软件设置并清除，以启用规则过采样。 0：禁用规则过采样 1：启用规则过采样 注意：只有当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)，软件才允许写入此位

Puya Confidential

13.5.6 ADC 采样时间寄存器 1（ADC_SMPR1）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:0]	
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMP5[2:0]		SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29:0	SMP[9:0][2:0]	RW	30'h0	通道 x 采样时间选择 这些位由软件编写，用于为每个通道单独选择采样时间。 在采样周期期间，通道选择位必须保持不变。 000：2.5 ADC 时钟周期 001：6.5 ADC 时钟周期 010：12.5 ADC 时钟周期 011：24.5 ADC 时钟周期 100：47.5 ADC 时钟周期 101：92.5 ADC 时钟周期 110：247.5 ADC 时钟周期 111：640.5 ADC 时钟周期

13.5.7 ADC 采样时间寄存器 2（ADC_SMPR2）

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP19[2:0]			SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:0]	
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMP15[2:0]		SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29:0	SMP[19:10][2:0]	RW	30'h0	通道 x 采样时间选择 这些位由软件编写，用于为每个通道单独选择采样时间。 在采样周期期间，通道选择位必须保持不变。 000：2.5 ADC 时钟周期 001：6.5 ADC 时钟周期 010：12.5 ADC 时钟周期 011：24.5 ADC 时钟周期 100：47.5 ADC 时钟周期 101：92.5 ADC 时钟周期 110：247.5 ADC 时钟周期 111：640.5 ADC 时钟周期

Puya confidential

13.5.8 ADC 采样时间寄存器 3（ADC_SMPR3）

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMP21[2:0]			SMP20[2:0]		
										RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:6	Reserved	-	-	保留
5:0	SMP20,21 [2:0]	RW	3'h0	通道 20、21 采样时间选择 这些位由软件编写，用于为每个通道单独选择采样时间。 在采样周期期间，通道选择位必须保持不变。 000：2.5 ADC 时钟周期 001：6.5 ADC 时钟周期 010：12.5 ADC 时钟周期 011：24.5 ADC 时钟周期 100：47.5 ADC 时钟周期 101：92.5 ADC 时钟周期 110：247.5 ADC 时钟周期 111：640.5 ADC 时钟周期

13.5.9 ADC 看门狗阈值寄存器 1（ADC_TR1）

偏移地址：0x24

复位值：0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT1[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	AWDFILT[2:0]			LT1[11:0]											
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:28	Reserved	-	-	保留
27:16	HT1[11:0]	RW	12'hFFF	模拟看门狗 1 高阈值 软件可配，定义模拟看门狗 1 高阈值 注意：只有当 ADSTART 且 JADSTART=0 时，软件才允许写入这些位（这确保了没有正在进行的转换）。
15	Reserved	-	-	保留
14:12	AWDFILT	RW	3'h0	模拟看门狗滤波参数 该位由软件设置和清除。 000：无过滤 001：连续两次检测生成 AWD 标志或中断 … 111：八个连续检测生成 AWD 标志或中断 注意：只有当 ADSTART 且 JADSTART=0 时，软件才允许写入此位（这确保没有转换正在进行）。

Puya Confidential

11:0	LT1[11:0]	RW	12’h0	模拟看门狗 1 低阈值 软件可配，定义模拟看门狗 1 低阈值 注意：只有当 ADSTART 且 JADSTART=0 时，软件才允许写入这些位（这确保了没有正在进行的转换）。

13.5.10 ADC 看门狗阈值寄存器 2（ADC_TR2）

偏移地址：0x28

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	-	保留
23:16	HT2[7:0]	RW	8’hFF	模拟看门狗 2 高阈值 软件可配，定义模拟看门狗 2 高阈值 注意：只有当 ADSTART 且 JADSTART=0 时，软件才允许写入这些位（这确保了没有正在进行的转换）。
15:8	Reserved	-	-	保留
7:0	LT2[7:0]	RW	8’h0	模拟看门狗 2 低阈值 软件可配，定义模拟看门狗 2 低阈值 注意：只有当 ADSTART 且 JADSTART=0 时，软件才允许写入这些位（这确保了没有正在进行的转换）。

13.5.11 ADC 看门狗阈值寄存器 3 (ADC_TR3)

偏移地址：0x2C

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	-	保留
23:16	HT3[7:0]	RW	8’hFF	模拟看门狗 3 高阈值 软件可配，定义模拟看门狗 3 高阈值 注意：只有当 ADSTART 且 JADSTART=0 时，软件才允许写入这些位（这确保了没有正在进行的转换）。
15:8	Reserved	-	-	保留
7:0	LT3[7:0]	RW	8’h0	模拟看门狗 3 低阈值 软件可配，定义模拟看门狗 3 低阈值

Puya Confidential

				注意：只有当 ADSTART 且 JADSTART=0 时，软件才允许写入这些位（这确保了没有正在进行的转换）。

13.5.12 ADC 规则序列寄存器 1(ADC_SQR1)

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4]
			RW	RW	RW	RW	RW		RW	RW	RW	RW	RW		RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ2[3:0]				Res.	SQ1[4:0]					Res.	Res.	LT3[7:0]			
RW	RW	RW	RW		RW	RW	RW	RW	RW			RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:29	Reserved	-	-	保留
28:24	SQ4[4:0]	RW	5'h0	规则序列第 4 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 4 次。
23	Reserved	-	-	保留
22:18	SQ3[4:0]	RW	5'h0	规则序列第 3 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 3 次。
17	Reserved	-	-	保留
16:12	SQ2[4:0]	RW	5'h0	规则序列第 2 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 2 次。
11	Reserved	-	-	保留
10:6	SQ1[4:0]	RW	5'h0	规则序列第 1 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 1 次。
5:4	Reserved	-	-	保留
3:0	L[3:0]	RW	4'h0	规则通道序列长度 软件配置这些位的值。这些位定义了在规则通道转换序列中通道数目。 0000：1 个转换 0001：2 个转换 …… 1111：16 个转换

某些通道未进行物理连接，因此不得选择进行转换。

13.5.13 ADC 规则序列寄存器 2(ADC_SQR2)

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4]
			RW	RW	RW	RW	RW		RW	RW	RW	RW	RW		RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ7[3:0]				Res.	SQ6[4:0]					Res.	SQ5[4:0]				
RW	RW	RW	RW		RW	RW	RW	RW	RW		RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:29	Reserved	-	-	保留
28:24	SQ9[4:0]	RW	5'h0	规则序列第 9 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 9 次。
23	Reserved	-	-	保留
22:18	SQ8[4:0]	RW	5'h0	规则序列第 8 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 8 次。
17	Reserved	-	-	保留
16:12	SQ7[4:0]	RW	5'h0	规则序列第 7 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 7 次。
11	Reserved	-	-	保留
10:6	SQ6[4:0]	RW	5'h0	规则序列第 6 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 6 次。
5	Reserved	-	-	保留
4:0	SQ5[4:0]	RW	5'h0	规则序列第 5 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 5 次。

某些通道未进行物理连接，因此不得选择进行转换。

13.5.14 ADC 规则序列寄存器 3(ADC_SQR3)

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4]
			RW	RW	RW	RW	RW		RW	RW	RW	RW	RW		RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SQ12[3:0]				Res.	SQ11[4:0]					Res.	SQ10[4:0]				
RW	RW	RW	RW		RW	RW	RW	RW	RW		RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:29	Reserved	-	-	保留
28:24	SQ14[4:0]	RW	5'h0	规则序列第 14 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 14 次。
23	Reserved	-	-	保留
22:18	SQ13[4:0]	RW	5'h0	规则序列第 13 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 13 次。
17	Reserved	-	-	保留
16:12	SQ12[4:0]	RW	5'h0	规则序列第 12 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 12 次。
11	Reserved	-	-	保留
10:6	SQ11[4:0]	RW	5'h0	规则序列第 11 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 11 次。
5	Reserved	-	-	保留
4:0	SQ10[4:0]	RW	5'h0	规则序列第 10 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 10 次。

某些通道未进行物理连接，因此不得选择进行转换。

Puya Confidential

13.5.15 ADC 规则序列寄存器 4(ADC_SQR4)

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
					RW	RW	RW	RW	RW		RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:11	Reserved	-	-	保留
10:6	SQ16[4:0]	RW	5’b0	规则序列第 16 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 16 次。
5	Reserved	-	-	保留
4:0	SQ15[4:0]	RW	5’h0	规则序列第 15 次转换 这些位由软件写入，任一 ADC 通道被分配为规则转换序列中的第 15 次。

某些通道未进行物理连接，因此不得选择进行转换。

13.5.16 ADC 规则数据寄存器（ADC_DR)

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	RDATA[15:0]	R	16’h0	规则通道转换结果 软件可读这些位的值。这些位为只读，包含了规则通道的转换结果。数据是左或右对齐。

13.5.17 ADC 注入序列寄存器(ADC_JSQR)

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:1]			
RW	RW	RW	RW	RW		RW	RW	RW	RW	RW		RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
JSQ2[0]	Res	JSQ1[4:0]					JEXTEN[1:0]		JEXTSEL[4:0]					JL[1:0]	
RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:27	JSQ4[4:0]	RW	5’h0	注入序列第 4 次转换

Puya Confidential

				这些位由软件写入，任一 ADC 通道被分配为注入转换序列中的第 4 次。
26	Reserved	-	-	保留
25:21	JSQ3[4:0]	RW	5’h0	注入序列第 3 次转换 这些位由软件写入，任一 ADC 通道被分配为注入转换序列中的第 3 次。
20	Reserved	-	-	保留
19:15	JSQ2[4:0]	RW	5’h0	注入序列第 2 次转换 这些位由软件写入，任一 ADC 通道被分配为注入转换序列中的第 2 次。
14	Reserved	-	-	保留
13:9	JSQ1[4:0]	RW	5’h0	注入序列第 1 次转换 这些位由软件写入，任一 ADC 通道被分配为注入转换序列中的第 1 次。
8:7	JEXTEN[1:0]	RW	2’h0	注入通道的外部触发启用和极性选择 这些位由软件设置和清除，以选择外部触发极性并启用注入组的触发。 00:禁用硬件触发检测（可以通过软件启动转换） 01:上升沿的硬件触发检测 10:下降沿的软件触发检测 11:上升沿和下降沿都有硬件触发检测 注意：只有当 ADSTART=0 且 JADSTART=0 时(确保没有任何转换正在进行)，软件才允许写入此位。
6:2	JEXTSEL[4:0]	RW	5’h0	注入通道组外部触发事件选择位 ADC 的外部触发事件如下：（详情见上述章节） 00000：事件 1 00001：事件 2 00010：事件 3 00011：事件 4 … 10000：事件 16 10001：事件 17 10010：事件 18 其它：保留
1:0	JL[1:0]	RW	2’h0	注入通道序列长度 这些位由软件编写，用于定义注入通道转换序列中的转换总数。 00：1 个转换 01：2 个转换 10：3 个转换 11：4 个转换

某些通道未进行物理连接，因此不得选择进行转换。

13.5.18 ADC offset y 寄存器(ADC_OFRy)

偏移地址：0x60+0x04*(y-1), (y=1 to 4)

复位值：0x0000 0000

(SATEN、OFFSETPOS 只有 ADC_OFR1 有，并且 ADC_OFR1-4 共用)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFF-SETy_EN	OFFSETy_CH[4:0]					SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Puya Confidential

Res.	Res.	Res.	Res.	OFFSETy[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31	OFFSETy_EN	RW	0	OFFSET y 使能 该位由软件写入，以启用或禁用 offset 值 OFFSETy[11:0]。 注意：只有当确保没有正在进行的转换时，软件才允许写入这些位。
30:26	OFFSETy_CH[4:0]	RW	5'h0	数据偏移 y 的通道选择 这些位由软件写入，以定义 OFFSETy[11:0]的偏移将应用的通道。 注意：只有当确保没有正在进行的转换时，软件才允许写入这些位。 某些通道没有物理连接，因此不能为数据偏移量 y 选择。
25	SATEN	RW	0	饱和使能 此位由软件设置和清除，以使能偏移在 0x000 和 0xFFF 饱和。 0：无饱和控制，偏移结果可以有符号 1：启用饱和，偏移结果无符号且在 0x000 和 0xFFF 处饱和 注意：只有当确保没有正在进行的转换时，软件才允许写入这些位。
24	OFFSETPOS	RW	0	正偏移 此位由软件设置和清除，以使能正偏移。 0：负偏移 1：正偏移 注意：只有当确保没有正在进行的转换时，软件才允许写入这些位。
23:12	Reserved	-	-	保留
11:0	OFFSETy[11:0]	RW	12'h0	OFFSETy_CH[4:0]通道的数据偏移 y。 这些位由软件写入，以定义在转换通道（可以是规则的或注入的）时从原始转换数据中减去的偏移 y。应用数据偏移 y 的通道必须在位 OFFSETy_CH[4:0]中编程。转换结果可以从 ADC_DR（规则转换）或 ADC_JDRy 寄存器（注入转换）中读取。 注意：只有当确保没有正在进行的转换时，软件才允许写入这些位。 如果多个偏移（OFFSETy）指向同一通道，则仅考虑具有最低 x 值的偏移进行减法。 例如：如果 OFFSET1_CH[4:0]=4 且 OFFSET2_CH[4:0]=4，则这是在转换通道 4 时减去的 OFFSET1[11:0]。

13.5.19 ADC 注入数据寄存器（ADC_JDRy)

偏移地址：0x80+0x04*(y-1)（y=1 to 4）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
JDATA[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	JDATA[15:0]	R	16'h0	注入转换数据

Puya Confidential

				这些位只读。注入通道 y 的转换结果放于此寄存器。数据是左对齐或者右对齐的。

13.5.20 ADC 模拟看门狗 2 配置寄存器（ADC_AWD2CR)

偏移地址：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[21:16]					
										RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AWD2CH[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:22	Reserved	-	-	保留
21:0	AWD2CH[21:0]	RW	22'h0	模拟看门狗 2 通道选择 这些位由软件设置和清除。它们启用并选择要由模拟看门狗 2 保护的输入通道。 AWD2CH[i]=0：AWD2 未监控 ADC 模拟输入通道 i AWD2CH[i]=1：AWD2 监控 ADC 模拟输出通道 i 当 AWD2CH[21:0]=000..0 时，模拟看门狗 2 被禁用 注意：AWD2CH 选择的通道也必须选择到 SQRi 或 JSQRi 寄存器中。 注意：只有当确保没有正在进行的转换时，软件才允许写入这些位。 某些通道未进行物理连接，不得为模拟看门狗选择。

13.5.21 ADC 模拟看门狗 3 配置寄存器（ADC_AWD3CR)

偏移地址：0xA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[21:16]					
										RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AWD3CH[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:22	Reserved	-	-	保留
21:0	AWD3CH[21:0]	RW	22'h0	模拟看门狗 3 通道选择 这些位由软件设置和清除。它们启用并选择要由模拟看门狗 3 保护的输入通道。 AWD3CH[i]=0：AWD3 未监控 ADC 模拟输入通道 i AWD3CH[i]=1：AWD3 监控 ADC 模拟输出通道 i 当 AWD3CH[21:0]=000..0 时，模拟看门狗 3 被禁用 注意：AWD3CH 选择的通道也必须选择到 SQRi 或 JSQRi 寄存器中。 注意：只有当确保没有正在进行的转换时，软件才允许写入这些位。 某些通道未进行物理连接，不得为模拟看门狗选择。

Puya Confidential

13.5.22 ADC 差分模式选择寄存器（ADC_DIFSEL)

偏移地址：0xB0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DIFFSEL[12]	Res.	DIFFSEL[10]	Res.	DIFFSEL[8]	Res.	DIFFSEL[6]	Res.	Res.	Res.	Res.	Res.	Res.
			RW		RW		RW		RW						

Bit	Name	RW	Reset Value	Function
31:13	Reserved	-	-	保留
12	DIFSEL[12]	RW	1'h0	通道 12 的差分模式。 这些位由软件设置和清除。它们允许选择通道配置为单端模式还是差分模式。 DIFSEL[i]=0：模拟输入通道 i 配置为单端模式 DIFSEL[i]=1：模拟输入通道 i 配置为差分模式 注意：DIFSEL 对应通道或连接到单端 i/O 端口或连接到内部通道必须保持复位值（单端输入模式）。 只有当 ADC 被禁用时，软件才允许写入这些位。
11	Reserved	-	-	保留
10	DIFSEL[10]	RW	1'h0	通道 10 的差分模式。 这些位由软件设置和清除。它们允许选择通道配置为单端模式还是差分模式。 DIFSEL[i]=0：模拟输入通道 i 配置为单端模式 DIFSEL[i]=1：模拟输入通道 i 配置为差分模式 注意：DIFSEL 对应通道或连接到单端 i/O 端口或连接到内部通道必须保持复位值（单端输入模式）。 只有当 ADC 被禁用时，软件才允许写入这些位。
9	Reserved	-	-	保留
8	DIFSEL[8]	RW	1'h0	通道 8 的差分模式。 这些位由软件设置和清除。它们允许选择通道配置为单端模式还是差分模式。 DIFSEL[i]=0：模拟输入通道 i 配置为单端模式 DIFSEL[i]=1：模拟输入通道 i 配置为差分模式 注意：DIFSEL 对应通道或连接到单端 i/O 端口或连接到内部通道必须保持复位值（单端输入模式）。 只有当 ADC 被禁用时，软件才允许写入这些位。
7	Reserved	-	-	保留
6	DIFSEL[6]	RW	1'h0	通道 6 的差分模式。 这些位由软件设置和清除。它们允许选择通道配置为单端模式还是差分模式。 DIFSEL[i]=0：模拟输入通道 i 配置为单端模式 DIFSEL[i]=1：模拟输入通道 i 配置为差分模式 注意：DIFSEL 对应通道或连接到单端 i/O 端口或连接到内部通道必须保持复位值（单端输入模式）。

Puya Confidential

				只有当 ADC 被禁用时，软件才允许写入这些位。
5:0	Reserved	-	-	保留

13.5.23 ADC 校准因子寄存器（ADC_CALFACT)

偏移地址：0xB4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCALFACT [8:0]									Res.	CAPSUC	OFFSUC	Res.	Res.	Res.	Res.
R	R	R	R	R	R	R	R	R		RC_W1	RC_W1				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RDVLD	WRVLD	FACTSEL[4:0]					WCALFACT [8:0]								
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:23	RCALFACT [8:0]	R	9’h00	溢出标志位读寄存器； 也是读校准结果寄存器（补码表示）。 当 RDVLD 有效时，读取 ADC 校准结果。只有 ADCAL 为 0 时，软件才能读取。 当 RDVLD 有效时，读取溢出标志位。只有 ADCAL 为 0 时，软件才能读取。
22	Reserved	-	-	保留
21	CAPSUC	RC_W1	0	电容校准状态位。 表示 ADC 电容校准是否成功。硬件置 1；软件写 1 置 0；
20	OFFSUC	RC_W1	0	Offset 校准状态位。 表示 ADC offset 校准是否成功。硬件置 1；软件写 1 置 0；
19:16	Reserved	-	-	保留
15	RDVLD	RW	0	溢出标志位读使能。 1：使能读溢出标志 0：复位状态 也是校准因子读使能。 1：使能读校准因子 0：复位状态
14	WRVLD	RW	0	校准因子写使能。 1：使能写校准因子 0：复位状态
13:9	FACTSEL[4:0]	RW	5’h0	写校准值,溢出标志选择位 当 RDVLD/WRVLD 有效时，选择需要读的校准结果/写入的校准因子加载到哪个电容。 {cov_error,cal_error}由硬件置位，软件清零。读{cov_error,cal_error}时需使能 RDVLD。写{cov_error,cal_error}时需使能 WDVLD，通过写 WCALFACT[1]清零 cov_error，通过写 WCALFACT[0]清零 cal_error。 5’d1：dos 5’d2：dcpm5 5’d3：dcpm4 5’d4：dcpm3 5’d5：dcpm2

Puya Confidential

				5’d6：dcpm1 5’d7：dcpn4 5’d8：dcpn3 5’d9：dcpn2 5’d10：dcpn1 5’d11：dcpk3 5’d12：dcm5 5’d13：dcm4 5’d14：dcm3 5’d15：dcm2 5’d16：dcm1 5’d17：dcnm0 5’d18：{cov_error,cal_error}: cov_error：当 AD 转换时，将校准结果补偿到 AD 转换时溢出，硬件置位该位，软件写 1 清零。 cal_error：当校准时，校准过程中产生溢出，硬件置位该位 ，软件写 1 清零。
8:0	WCALFACT [8:0]	RW	9’h0	写校准因子寄存器（补码表示）。 当 WRVLD 有效时，WCALFACT 值加载到 ADC 中。 注意(ADCAL 为 0 时，软件才能写入)。

13.5.24 ADC 增益补偿寄存器（ADC_GCOMP)

偏移地址：0xC0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	GCOMPCOEFF[13:0]													
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:14	Reserved	-	-	保留
13:0	GCOMPCOEFF[13:0]	RW	14’h0	增益补偿系数 这些位由软件设置和清除，以编程增益补偿系数。 00 1000 0000 0000：增益因子为 0.5 … 01 0000 0000 0000：增益因子为 1 10 0000 0000 0000：增益因子为 2 11 0000 0000 0000：增益因子为 3 … 该系数除以 4096，得到范围从 0 到 3.999756 的增益因子。 注意：此增益补偿仅在 ADC_CFGR2 寄存器的 GCOMP 位为 1 时应用。

Puya Confidential

13.5.25 ADC 模拟控制寄存器（ADC_ANACR)

偏移地址：0xC4

复位值：0x0000 0000

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

Bit	Name	RW	Reset Value	Function
31:5	Reserved	-	-	保留
4	VREFSEL	RW	0	ADC 参考电压选择 0：V_{REFP} 作为参考电压（V_{REFP} 连接到 V_{CCA}） 1：V_{REFBUF} 作为参考电压
3:1	PWRMODE	RW	3’b0	ADC 内比较器的功耗模式配置 000：typical 6 μA (default) 001：typical 7 μA 010：typical 8 μA 011：typical 9 μA 100：typical 2 μA 101：typical 3 μA 110：typical 4 μA
0	Reserved	-	-	保留

13.5.26 ADC 公共寄存器

13.5.26.1 ADCx 公共状态寄存器（ADC_CSR，x=12)

偏移地址：0x304（只有 Master ADC 有）

复位值：0x0000 0000

该寄存器提供不同 ADC 的状态位。然而，它是只读的且不允许清除不同的状态位。相反，每个状态位必须通过在相应的 ADC_ISR 寄存器中写入 0 来清除。

一个接口控制 ADC1 和 ADC2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	AWD3_SLV	AWD2_SLV	AWD1_SLV	JEOS_SLV	JEOC_SLV	OVR_SLV	EOS_SLV	EOC_SLV	EOSMP_SLV	ADRDY_SLV
						R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
						R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25	AWD3_SLV	R	0	从 ADC 的模拟看门狗 3 标志。 该位是对应 ADC_ISR 寄存器中 AWD3 位的副本。
24	AWD2_SLV	R	0	从 ADC 的模拟看门狗 2 标志。 该位是对应 ADC_ISR 寄存器中 AWD2 位的副本。

Puya Confidential

23	AWD1_SLV	R	0	从 ADC 的模拟看门狗 1 标志。 该位是对应 ADC_ISR 寄存器中 AWD1 位的副本。
22	JEOS_SLV	R	0	从 ADC 的注入序列转换结束标志。 该位是对应 ADC_ISR 寄存器中 JEOS 位的副本。
21	JEOC_SLV	R	0	从 ADC 的注入通道转换结束标志。 该位是对应 ADC_ISR 寄存器中 JEOC 位的副本。
20	OVR_SLV	R	0	从 ADC 的 ADC 过载 该位是对应 ADC_ISR 寄存器中 OVR 位的副本。
19	EOS_SLV	R	0	从 ADC 的规则序列结束标志 该位是对应 ADC_ISR 寄存器中 EOS 位的副本。
18	EOC_SLV	R	0	从 ADC 的转换结束标志 该位是对应 ADC_ISR 寄存器中 EOC 位的副本。
17	EOSMP_SLV	R	0	从 ADC 的采样结束标志 该位是对应 ADC_ISR 寄存器中 EOSMP 位的副本。
16	ADRDY_SLV	R	0	从 ADC 的 ADC ready。 该位是对应 ADC_ISR 寄存器中 ADRDY 位的副本。
15:10	Reserved	-	-	保留
9	AWD3_MST	R	0	主 ADC 的模拟看门狗 3 标志。 该位是对应 ADC_ISR 寄存器中 AWD3 位的副本。
8	AWD2_MST	R	0	主 ADC 的模拟看门狗 2 标志。 该位是对应 ADC_ISR 寄存器中 AWD2 位的副本。
7	AWD1_MST	R	0	主 ADC 的模拟看门狗 1 标志。 该位是对应 ADC_ISR 寄存器中 AWD1 位的副本。
6	JEOS_MST	R	0	主 ADC 的注入序列转换结束标志。 该位是对应 ADC_ISR 寄存器中 JEOS 位的副本。
5	JEOC_MST	R	0	主 ADC 的注入通道转换结束标志。 该位是对应 ADC_ISR 寄存器中 JEOC 位的副本。
4	OVR_MST	R	0	主 ADC 的 ADC 过载 该位是对应 ADC_ISR 寄存器中 OVR 位的副本。
3	EOS_ MST	R	0	主 ADC 的规则序列结束标志 该位是对应 ADC_ISR 寄存器中 EOS 位的副本。
2	EOC_ MST	R	0	主 ADC 的转换结束标志 该位是对应 ADC_ISR 寄存器中 EOC 位的副本。
1	EOSMP_ MST	R	0	主 ADC 的采样结束标志 该位是对应 ADC_ISR 寄存器中 EOSMP 位的副本。
0	ADRDY_ MST	R	0	主 ADC 的 ADC ready。 该位是对应 ADC_ISR 寄存器中 ADRDY 位的副本。

13.5.26.2 ADCx 公共控制寄存器（ADC_CCR，x=12）

偏移地址：0x308（只有 Master ADC 有）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	BUFFER_SEL_CR	BUFFER_EN	VSENSESEL	VREFEN	PRESC[3:0]				CKMODE	
						RW	RW	RW	R W	RW	RW	RW	RW	RW	RW

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDMA[1:0]		DMACFG	Res.	DELAY[3:0]				Res.	Res.	Res.	DUAL[4:0]				
RW	RW	RW		RW	RW	RW	RW				RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25	BUFFER_SEL_CR	RW	0	VTS 和 V_{REFINT} 共享的 BUFFER 选择 0：选择 VTS 1：选择 V_{REFINT}
24	BUFFER_EN	RW	0	VTS 和 VREFINT 共享的 BUFFER 使能 0：关闭 BUFFER; 1：开启 BUFFER;
23	VSENSESEL	RW	0	VTS 选择 此位由软件控制 VTS 的设置和清除。 0：温度传感器通道禁用 1：温度传感器通道启用
22	VREFEN	RW	0	V_{REFINT} 使能 该位由软件设置和清除，以启用/禁用 V_{REFINT} 通道。 0：V_{REFINT} 通道禁用 1：V_{REFINT} 通道启用
21:18	PRESC[3:0]	RW	4’h0	ADC 预分频系数，针对异步时钟模式 由软件置 1 和清零，以选择 ADC 的时钟频率。 0000：输入 ADC 时钟未分频 0001：输入 ADC 时钟 2 分频 0010：输入 ADC 时钟 4 分频 0011：输入 ADC 时钟 6 分频 0100：输入 ADC 时钟 8 分频 0101：输入 ADC 时钟 10 分频 0110：输入 ADC 时钟 12 分频 0111：输入 ADC 时钟 16 分频 1000：输入 ADC 时钟 32 分频 1001：输入 ADC 时钟 64 分频 1010：输入 ADC 时钟 128 分频 1011：输入 ADC 时钟 256 分频 其他值：保留
17:16	CKMODE[1:0]	RW	2’h0	ADC 时钟模式 此位由软件置 1 和清零，用于定义为模拟 ADC 提供时钟的方式： 00：ADCCLK（异步时钟模式），由 RCC 配置源 01：PCLK/2（同步时钟模式） 10：PCLK/4（同步时钟模式） 11：PCLK（同步时钟模式）。使能此配置时，PCLK 的时钟占空比必须为 50%

Puya Confidential

				在所有同步时钟模式下，从定时器触发到转换开始的延迟过程中，不存在抖动。 注: 仅当 ADC 已禁止时 (ADCAL=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。
15:14	MDMA[1:0]	RW	2'h0	双 ADC 模式的 DMA 该位字段由软件设置和清除。有关更多详细信息，请参阅 DMA 章节 00：禁用 MDMA 模式 01：保留 10：为 12 位和 10 位分辨率启用 MDMA 模式 11：为 8 位和 6 位分辨率启用了 MDMA 模式 注意：只有当 ADSTART=0 时，软件才允许写入这些位（这确保了没有进行规则转换）。
13	DMACFG	RW	0	DMA 配置（用于双 ADC 模式） 此位由软件设置和清除，用于选择两种 DMA 操作模式，仅当 DMAEN=1 时有效。 0：选择 DMA 单次模式 1：选择 DMA 循环模式 有关更多详细信息，请参阅：使用 DMA 管理转换 注意：只有当 ADSTART=0 时，软件才允许写入这些位（这确保没有规则转换正在进行）。
12	Reserved	-	-	保留
11:8	DELAY[3:0]	RW	4'h0	两个采样阶段之间的延迟 这些位由软件设置和清除。这些比特用于双 ADC 交错模式。 有关 ADC 分辨率与 DELAY 位值的关系，请参阅表 13-12。 注意：只有当 ADC 被禁用时（ADCAL=0，JADSTART=0，ADSTART=0、ADSTP=0，ADDIS=0 和 ADEN=0），软件才允许写入这些位。
7:5	Reserved	-	-	保留
4:0	DUAL[4:0]	RW	5'h0	双 ADC 模式选择 这些位由软件写入以选择双 ADC 模式。 所有 ADC 独立： 00000：独立模式 00001 至 01001：双 ADC 模式，主从 ADC 协同工作 00001：组合的同步规则+同步注入模式 00010：组合的同步规则+交替触发模式 00011：组合的交错模式+同步注入模式 00100：保留 00101：仅同步注入模式 00110：仅同步规则模式

Puya Confidential

				00111：仅交错模式 01001：仅交替触发模式 所有其他组合是保留的，不得编程 注意：只有当 ADC 被禁用时（ADCAL=0，JADSTART=0， ADSTART=0 、 ADSTP=0 ，ADDIS=0 和 ADEN=0），软件才允许写入这些位。

13.5.26.3 ADCx 公共规则数据寄存器 (ADC_CDR,x=12)

偏移地址：0x30C（只有 Master ADC 有）

复位值：0x0000 0000

一个接口控制 ADC1 和 ADC2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDATA_SLV[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RDATA_MST[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31：16	RDATA_SLV[15:0]	R	16’h0	从 ADC 规则通道转换数据 在双 ADC 模式中，这些位包含从 ADC 的规则数据。请参阅双 ADC 模式。 数据对齐所述：数据寄存器、数据对齐和偏移（ADC_DR、OFFSETy、OFFSETy_CH、ALIGN）
15:0	RDATA_MST[15:0]	R	16’h0	主 ADC 的规则数据。 在双 ADC 模式中，这些位包含主 ADC 的规则数据。请参阅双 ADC 模式。 数据对齐所述：数据对齐：数据寄存器、数据对齐和偏移（ADC_DR、OFFSETy、OFFSETy_CH、ALIGN）。 在 MDMA=0b11 模式中，位 15:8 包含 SLV_ADC_DR[7:0]，位 7:0 包含 MST_ADC_DR[7:0]。

Puya Confidential

14. 数字/模拟转换（DAC）

14.1 DAC 简介

数字/模拟转换模块（DAC）是 12 位数字输入，电压输出的数字/模拟转换器。DAC 可以配置为 8 位或 12 位模式，也可以与 DMA 控制器配合使用。DAC 工作在 12 位模式时，数据可以设置成左对齐或右对齐。DAC 模块有 2 个输出通道，每个通道都有单独的转换器。在双 DAC 模式下，2 个通道可以独立地进行转换，也可以同时进行转换并同步地更新 2 个通道的输出。DAC 可以通过引脚输入参考电压 V_{REF+} 以获得更精确的转换结果。也选择内部参考电压 V_{REFBUF}。

14.2 DAC 主要特性

- 1 个 DAC 接口，2 个输出通道
- 12 位模式下数据左对齐或者右对齐
- 同步更新功能
- 噪声波形生成
- 三角波形生成
- 斜坡发生器生成
- 双 DAC 通道同时或者分别转换
- 每个通道都有 DMA 功能
- 支持 DMA 下溢错误检测
- 外部触发转换
- 参考电压
  - 输入参考电压 V_{REF+}
  - 内部参考电压 V_{REFBUF}

Puya Confidential

14.3 DAC 功能描述

14.3.1 DAC 框图

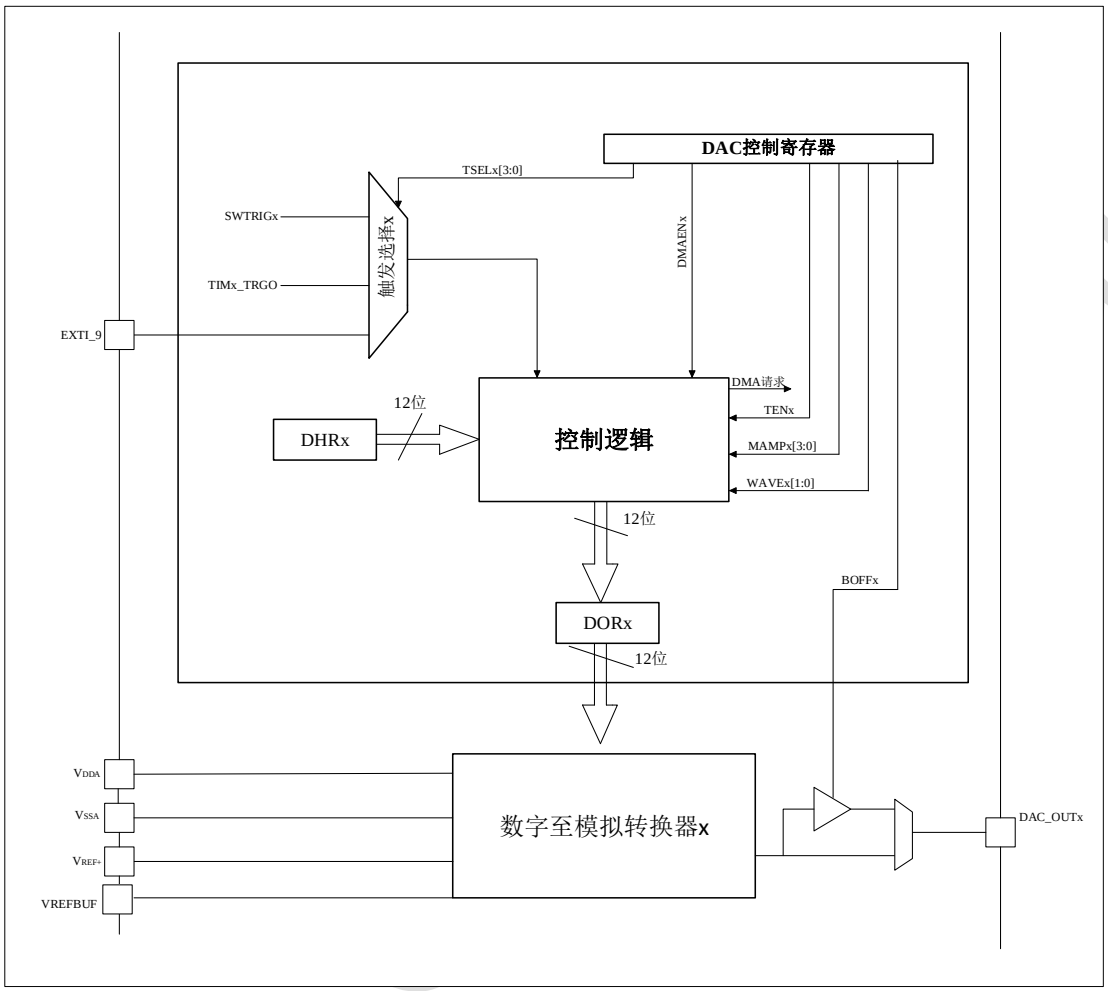


图 14-1 DAC 模块框图

14.3.2 DAC 校准

校准方式有 2 种:

1）输入 DOR=12’h800，扫描 DAC_OS_TRIMCR 改变 VCM 电压，DAC buffer 输出给 ADC，通过 ADC 测量 DAC 输出；

a) 将 ADC 使能，选择内部通道模式，内部通道选择 DAC 输入（ADC 需要先校准好）

b) 配置 DAC 的 ENx=1，DAC_OS_TRIMCR 为 6’h11, DAC BUFF 关闭。

c) 固定 DAC 的 DACCxDOR [11:0]=0X7FF（2047），调节 DAC_OS_TRIMCR_VDDK[5:0]的值，直至 ADC 的输出值为 2047±1，此时校准完成。

2）输入 DOR=12’h800，扫描 DAC_OS_TRIMCR 改变 VCM 电压，DAC buffer 断开负端反馈支路，改为比较器，数字电路采样比较器输出进行判断（若 DAC_CMPOUT 为 0，将 DAC_OS_TRIMCR+1 直到 DAC_CMPOUT 为 1）。

14.3.3 DAC 通道使能

将 DAC_CR 寄存器的 ENx 位置‘1’即可打开对 DAC 通道 x 的供电。经过一段启动时间（t_{WAKEUP}）（数值参见数据手册），DAC 通道 x 即被使能。

注意：ENx 位只会使能 DAC 通道 x 的模拟部分，即便该位被置‘0’，DAC 通道 x 的数字部分仍然工作。

注意：DAC 不支持动态修改模式 WAVE，修改 WAVE 前需关闭 DAC。

14.3.4 DAC 输出缓存使能

DAC 集成了 1 个输出缓存，可以用来减少输出阻抗，无需外部运放即可直接驱动外部负载。两个 DAC 复用同一个缓存可以通过设置 DAC_OC 寄存器的 DAC_EXTEN 位来使能或者关闭。

14.3.5 DAC 数据格式

根据选择的配置模式，数据按照下文所述写入指定的寄存器：

单 DAC 通道 x，有 3 种情况：

- 8 位数据右对齐：用户须将数据写入寄存器 DAC_DHR8Rx[7:0]位(实际是存入寄存器 DAC_DHRx[11:4]位)
- 12 位数据左对齐：用户须将数据写入寄存器 DAC_DHR12Lx[15:4]位(实际是存入寄存器 DAC_DHRx[11:0]位)
- 12 位数据右对齐：用户须将数据写入寄存器 DAC_DHR12Rx[11:0]位(实际是存入寄存器 DAC_DHRx[11:0]位)

根据对 DAC_DHRyyyx 寄存器的操作(yyy 指不同的数据格式和对齐方式)，经过相应的移位后，写入的数据被转存到 DAC_DHRx 寄存器中(DHRx 是内部的数据保存寄存器 x)。随后，DAC_DHRx 寄存器的内容通过软件触发或外部触发自动地传送到 DAC_DORx 寄存器。

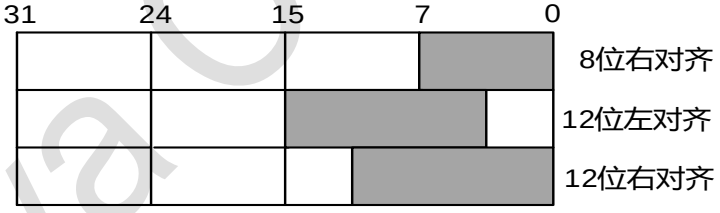


图 14-2 单 DAC 通道模式的数据寄存器

双 DAC 通道，有 3 种情况：

- 8 位数据右对齐：用户须将 DAC 通道 1 数据写入寄存器 DAC_DHR8RD[7:0]位（实际是存入寄存器 DAC_DHR1[11:4]位），将 DAC 通道 2 数据写入寄存器 DAC_DHR8RD[15:8]位（实际是存入寄存器 DAC_DHR2[11:4]位）
- 12 位数据左对齐：用户须将 DAC 通道 1 数据写入寄存器 DAC_DHR12LD[15:4]位（实际是存入寄存器 DAC_DHR1[11:0]位），将 DAC 通道 2 数据写入寄存器 DAC_DHR12LD[31:20]位（实际是存入寄存器 DAC_DHR2[11:0]位）
- 12 位数据右对齐：用户须将 DAC 通道 1 数据写入寄存器 DAC_DHR12RD[11:0]位（实际是存入寄存器 DAC_DHR1[11:0]位），将 DAC 通道 2 数据写入寄存器 DAC_DHR12RD[27:16]位（实际是存入寄存器 DAC_DHR2[11:0]位）

Puya Confidential

根据对 DAC_DHRyyyD（yyy 指不同的数据格式和对齐方式）寄存器的操作，经过相应的移位后，写入的数据被转存到 DAC_DHR1 和 DAC_DHR2 寄存器中（DAC_DHR1 和 DAC_DHR2 是内部的数据保存寄存器）。随后，DAC_DHR1 和 DAC_DHR2 的内容通过软件触发或外部事件触发自动地传送到 DAC_DORx 寄存器。

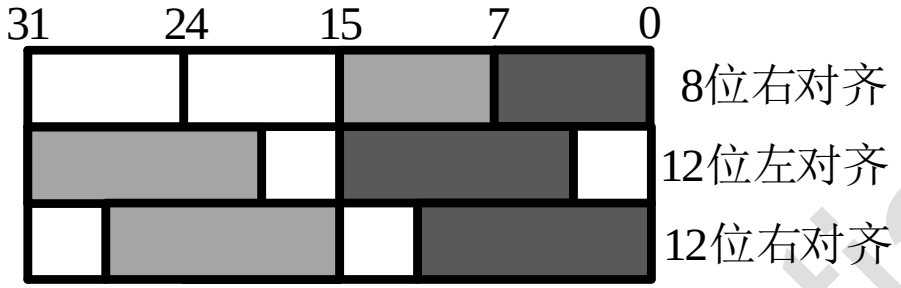


图 14-3 双 DAC 通道模式的数据寄存器

14.3.6 DAC 转换

不能直接对寄存器 DAC_DORx 写入数据，任何输出到 DAC 通道 x 的数据都必须写入 DAC_DHRx 寄存器(数据实际写入 DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx、DAC_DHR8RD、DAC_DHR12LD、或者 DAC_DHR12RD 寄存器)。

如果没有选中硬件触发（寄存器 DAC_CR 的 TENx 位置‘0’），存入寄存器 DAC_DHRx 的数据会在 1 个 APB1 时钟周期后自动传到寄存器 DAC_DORx 中。如果选中硬件触发（寄存器 DAC_CR 的 TENx 位置‘1’），数据传输在触发发生以后 3 个 APB1 时钟周期完成。

一旦数据从 DAC_DHRx 寄存器装入 DAC_DORx 寄存器，在经过建立时间（$t_{SETTLING}$）之后，输出即有效，这段时间的长短随着电源电压和模拟输出负载的不同会有所变化。

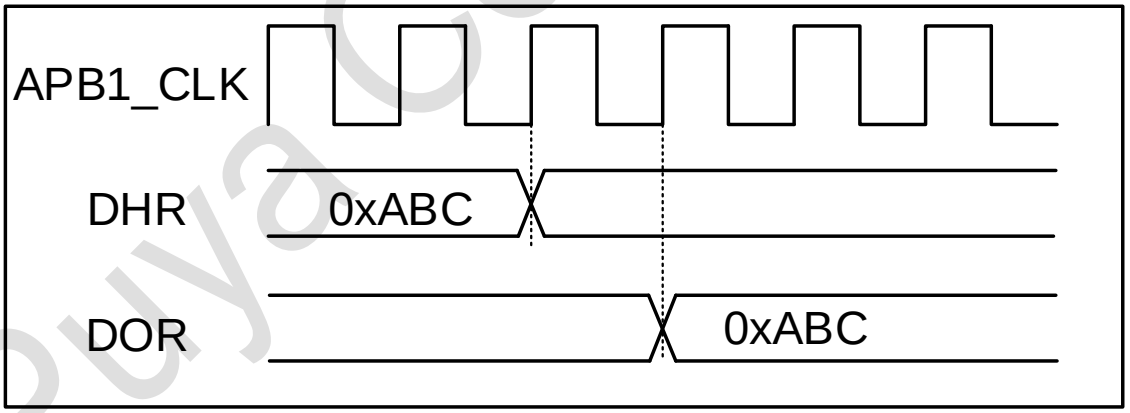


图 14-4 TEN=0 时转换时序图

14.3.7 DAC 输出电压

数字输入经过 DAC 被线性地转换为模拟电压输出，其范围为 0 到 V_{REF+}。

任一 DAC 通道引脚上的输出电压满足下面的关系：

DAC 输出 = V_{REF+} x（DOR / 4095)。

Puya Confidential

14.3.8 DAC 触发选择

如果 TENx 位被置 1，DAC 转换可以由某外部事件触发(定时计数器、外部中断线和软件操作)。配置控制位 TSELx[3:0]可以选择 14 个触发事件之一触发 DAC 转换。斜坡发生器触发不用配置 TENx。

表 14-1 外部触发

TSELx[3:0]/ RAMPSTPTRIGSELx[3:0]/ RAMPUPTRIGSELx[3:0]	DAC CH1 触发信号选择		DAC CH2 触发信号选择	
	update/reset	inc	update/reset	inc
0	sw(SWTRIG1)	sw(SWTRIG1)	sw(SWTRIG2)	sw(SWTRIG2)
1	TIM8_TRGO	TIM8_TRGO	TIM8_TRGO	TIM8_TRGO
2	TIM1_TRGO	TIM1_TRGO	TIM1_TRGO	TIM1_TRGO
3	TIM1_TRGO2	TIM1_TRGO2	TIM1_TRGO2	TIM1_TRGO2
4	TIM2_TRGO	TIM2_TRGO	TIM2_TRGO	TIM2_TRGO
5	TIM8_TRGO2	TIM8_TRGO2	TIM8_TRGO2	TIM8_TRGO2
6	EXTI9	EXTI10	EXTI9	EXTI10
7	EPWM1_SYNCPER	EPWM1_SYNCPER	EPWM1_SYNCPER	EPWM1_SYNCPER
8	TIM3_TRGO	TIM3_TRGO	TIM3_TRGO	TIM3_TRGO
9	EPWM2_SYNCPER	EPWM2_SYNCPER	EPWM2_SYNCPER	EPWM2_SYNCPER
10	EPWM3_SYNCPER	EPWM3_SYNCPER	EPWM3_SYNCPER	EPWM3_SYNCPER
11	EPWM4_SYNCPER	EPWM4_SYNCPER	EPWM4_SYNCPER	EPWM4_SYNCPER
12	COMP1_OUT	EPWM5_SYNCPER	COMP1_OUT	EPWM5_SYNCPER
13	COMP2_OUT	EPWM6_SYNCPER	COMP2_OUT	EPWM6_SYNCPER
14	COMP3_OUT	COMP1_OUT	COMP3_OUT	COMP1_OUT
15	COMP4_OUT	COMP2_OUT	COMP4_OUT	COMP2_OUT

每次 DAC 侦测到来自选中的定时器 TRGO 或者外部中断线 9（事件）的上升沿输出时，最近存放在寄存器 DAC_DHRx 中的数据会被传送到寄存器 DAC_DORx 中。在 3 个 APB1 时钟周期后，寄存器 DAC_DORx 更新为新值。

如果选择软件触发，一旦 SWTRIG 位置‘1’，转换即开始。在数据从 DAC_DHRx 寄存器传送到 DAC_DORx 寄存器后，SWTRIG 位由硬件自动清‘0’。

注意：

1、不能再 ENx 为‘1’时改变 TSELx[3:0]位。

2、如果选择软件触发，数据从寄存器 DAC_DHRx 传送到寄存器 DAC_DORx 只需要 1 个 APB1 时钟周期。

3、使能输出缓存时，触发的频率不能超过 1 M；关闭输出缓存时，触发的频率不能超过 15 M。

14.3.9 DMA 功能

DMA 请求

任一 DAC 通道都具有 DMA 功能。2 个 DMA 通道可分别用于 2 个 DAC 通道的 DMA 请求。

如果 DAC_CR.DMAENx 位置‘1’，一旦有外部触发(而不是软件触发)发生，则会在 3 个时钟周期后产生一个 DMA 请求，然后 DAC_DHRx 寄存器的数据被传送到 DAC_DORx 寄存器中。

在双 DAC 模式下(即使用 DAC_DHR12RD 或 DAC_DHR12LD 或 DAC_DHR8RD 寄存器)，DAC_CR.DMAENx 中仅一位应被置位。

Puya Confidential

DMA 下溢检测

DAC 的 DMA 请求没有缓冲队列，因此如果第二个外部触发在接收到对第一个外部触发的应答（第一个请求）之前到达，则不会发出新的 DMA 请求，并将 DAC_SR 寄存器中的 DMA 下溢标志 DMAUDRx 将置“1”，报告错误。然后禁用 DMA 数据传输，不再处理任何 DMA 请求。DAC 通道继续转换原数据。软件应通过对寄存器 DMAUDRx 写入“1”来清除标志，将所用 DMA 通道的 DMAEN 位置零，并重新初始化 DMA 和 DAC 通道，以便正确地重新开始 DMA 传输。同时软件应修改 DAC 触发转换频率来减轻 DMA 工作负载，以避免再次发生 DMA 下溢情况。

对于各 DAC 通道，如果使能 DAC_CR 寄存器中相应的 DMAUDRIEx 位，还将产生相应的中断。

14.3.10 DAC 噪声生成

可以利用线性反馈移位寄存器（Linear Feedback Shift Register LFSR）产生幅度变化的伪噪声。设置 DAC_CR.WAVE[1:0]位为‘01’选择 DAC 噪声生成功能。寄存器 LFSR 的预装入值为 0xAAA。按照特定算法，在每次触发事件后 3 个 APB1 时钟周期之后更新该寄存器的值。

设置 DAC_CR.MAMPx[3:0]位可以屏蔽部分或者全部 LFSR 的数据，这样得到的 LFSR 的值与 DAC_DHRx 的数值相加，如果得到的数值有溢出，则将寄存器所能保留的最大值写入 DAC_DORx 中，如果得到的数值没有溢出，则将该值写入 DAC_DORx 中。

如果寄存器 LFSR 值为 0x000，则会注入‘1’(防锁定机制)。

将 DAC_CR.WAVEx[1:0]位置‘0’可以复位 LFSR 波形的生成算法。

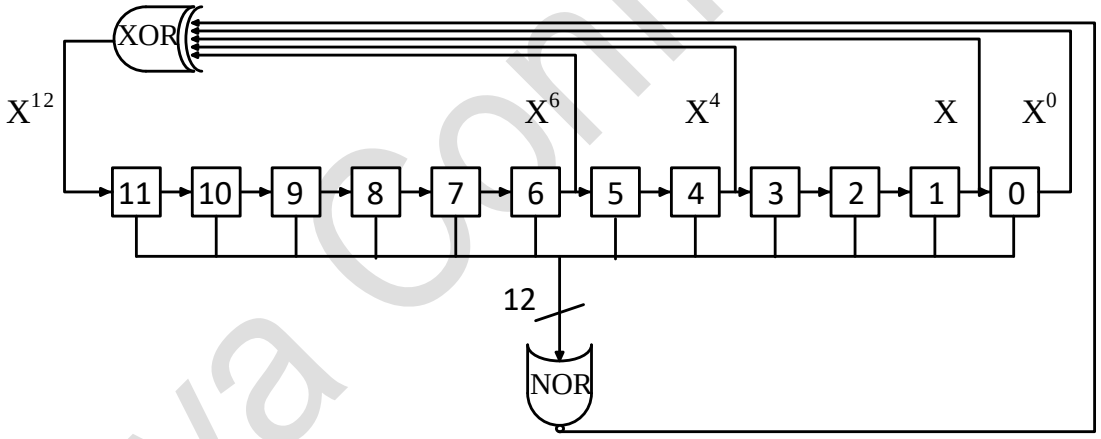


图 14-5 DAC LFSR 寄存器算法

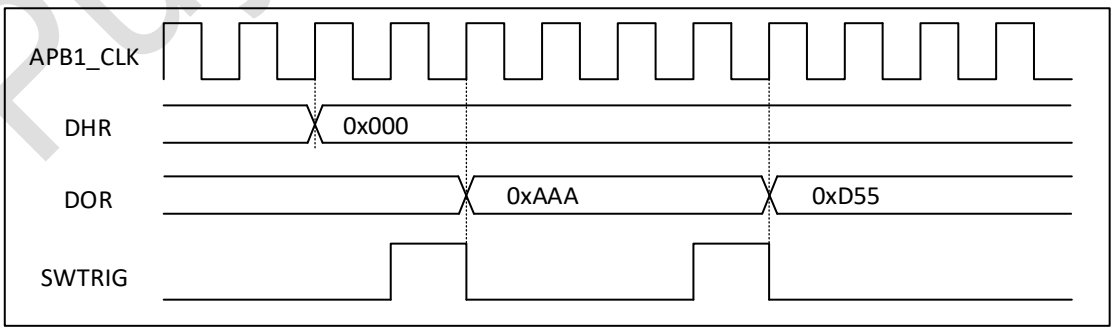


图 14-6 带 LFSR 波形生成的 DAC 转换（软件触发）

Puya Confidential

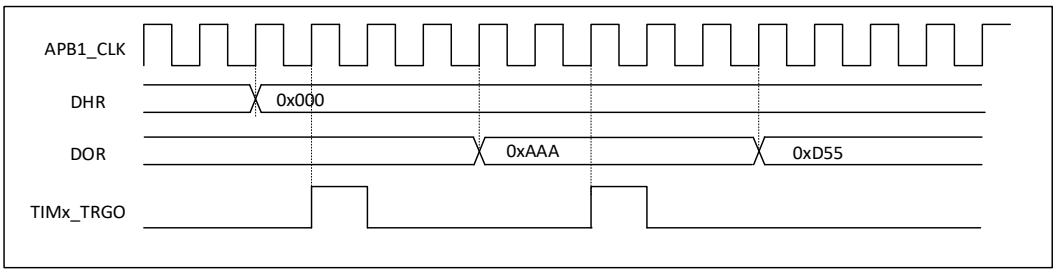


图 14-7 带 LFSR 波形生成的 DAC 转换（硬件触发）

注意：

1. 为了产生噪声，必须使能 DAC 触发，即设 DAC_CR 寄存器的 TENx 位为‘1’；
2. 若 DAC_DHR 的值发生了改变，则 LFSR 寄存器的值会复位为 0xAAA；
3. 若 DAC_CR.TENx 从“1”变为“0”，则 LFSR 寄存器的值也会复位为 0xAAA。

14.3.11 DAC 三角波生成

可以在 DC 或者缓慢变化的信号上加上一个小幅度的三角波。设置 DAC_CR.WAVEx[1:0]位为“10”选择 DAC 的三角波生成功能。设置 DAC_CR.MAMPx[3:0]位来选择三角波的幅度。内部的三角波计数器每次触发事件之后 3 个 APB1 时钟周期累加 1。一旦达到设置的最大幅度，则计数器开始递减，达到 0 后再开始累加，周而复始。

将 DAC_CR.WAVEx[1:0]位置‘0’可以复位三角波的生成。

注意：当基值为寄存器 DAC_DORx 的最大值时，不论 MAMPx 为何值，DAC_DORx 的输出均为其最大值。

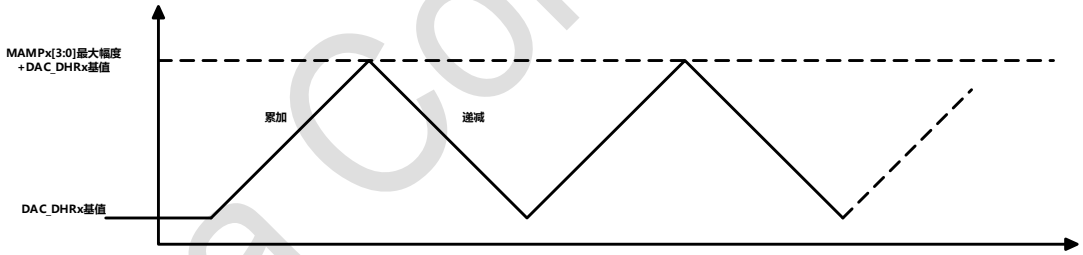


图 14-8 DAC 三角波生成

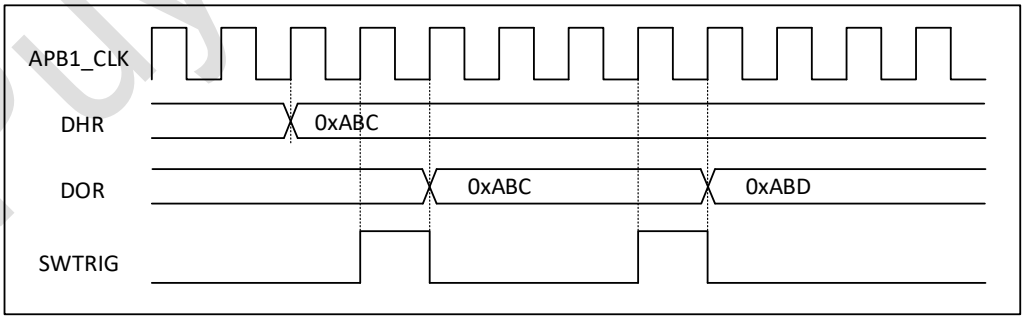


图 14-9 带三角生成的 DAC 转换（软件触发）

Puya Confidential

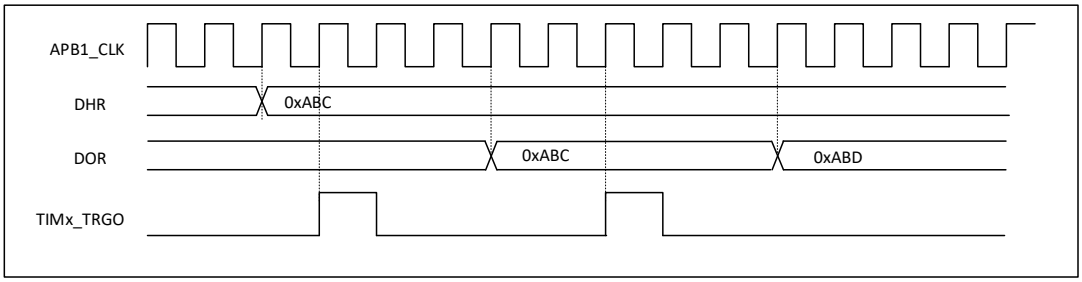


图 14-10 带三角生成的 DAC 转换(硬件触发)

注意：

- 为了产生三角波，必须使能 DAC 触发，即设 DAC_CR 寄存器的 TENx 位为‘1’；
- DAC_CR.MAMP[3:0]位必须在使能 DAC 之前设置，否则其值不能修改；
- 改变 DAC_DHR 的值，三角波计数器复位为 0；
- 若 DAC_CR.TENx 从“1”变为“0”，三角波计数器也复位为 0。

14.3.12 DAC 斜坡发生器

斜坡发生器为 12 位 DAC 生成下降斜坡输入。在此模式下，12 位 DAC 使用 RAMPSTS[11:0]递减计数器计数。通过设置 WAVEx=11 可启用斜坡发生器。

当设置 WAVEx=11 时，RAMPSTS[11:0]寄存器会从 RAMPREFAx（斜坡基值影子寄存器）或 RAMPREFSx 加载初值（通过 RAMPLOADSELx 选择）。

当设置 WAVEx=11 时，RAMPREFSx[11:0]值加载到 RAMPREFAx 寄存器，RAMPHSTEPVALSx[11:0]加载到 RAMPHSTEPVALAx[11:0]（斜坡步进值影子寄存器）寄存器，RAMPDEALYDATAx 加载到 RAMPDEALYDATASx（斜坡延迟影子寄存器）寄存器，并保持直至接收到斜坡启动触发(RAMPSTART)(外部触发 inc)信号。接收到该信号后，DAC 会在每个 DAC CLK 时钟周期从 RAMPSTS[11:0]中减去 RAMPHSTEPVALSx[11:0]设定值。

为防止减法操作在斜坡启动触发（RAMPSTART）（外部触发 inc）后的单个 PCLK 周期就启动，可通过 RAMPDEALYDATAx [11:0]暂缓 RAMPSTS 减法操作。当斜坡启动触发（RAMPSTAR：外部触发 inc）发生时，RAMPDEALYDATAx 寄存器值会在每个 PCLK 周期递减直至归零，此时 RAMPSTS 减法操作才会正式启动。

14.3.12.1 斜坡发生器常规行为

斜坡发生器在以下信号上升沿触发状态更新：wave=11，斜坡启动触发（RAMPSTART），斜坡复位触发（RAMPSTOP）；

在 wave=11 上升沿，将 RAMPHREFS、RAMPHSTEPVALSx 及 RAMPDEALYDATAx 加载到影子寄存器，将 RAMPHREFAx 加载到 RAMPSTS；

在斜坡启动触发 (RAMPSTART) 上升沿, RAMPHREFSx、RAMPHSTEPVALSx、RAMPDEALYDATAx 更新影子寄存器。RAMPHREFSx 更新 RAMPSTS，当延迟递减计数器归零后启动斜坡递减。

在斜坡复位触发(RAMPSTOP)上升沿(RAMPLOADSEL=1),将 RAMPHREFSx、RAMPHSTEPVALSx、RAMPDEALYDATAx 加载到影子寄存器。在 RAMPHREFSx 加载到 RAMPSTS 值并停止递减。

在斜坡复位触发（RAMPSTOP）上升沿（RAMPLOADSE=0），将 RAMPHREFAx 加载到 RAMPSTS 并停止递减。

Puya Confidential

另外，当 RAMPSTS 递减至零时，该寄存器将保持零值直至下次斜坡启动触发（RAMPSTART）信号触发。状态迁移逻辑见下图

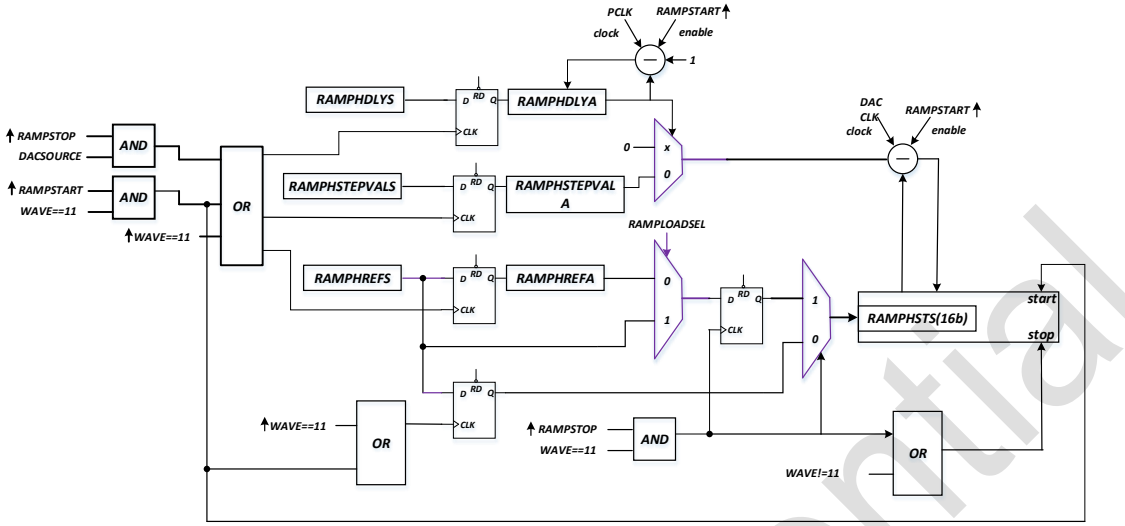


图 14-11 斜坡发生模块框图

14.3.12.2 斜坡发生器行为模型（极端情况）

由于斜坡发生器会在斜坡启动触发（RAMPSTART）和斜坡复位触发（RAMPSTOP）的每个上升沿发生状态变化，因此当这两个事件同时发生或发生时间非常接近时，可能会出现以下情况：

情况 1：斜坡复位触发（RAMPSTOP）上升沿比斜坡启动触发（RAMPSTART）上升沿早一个或多个周期发生。

•在斜坡复位触发（RAMPSTOP）上升沿事件发生时，RAMPSTS 停止递减。

•在斜坡启动触发（RAMPSTART）上升沿事件发生时，当延迟计数器减至 0 时，RAMPSTS 开始递减。

情况 2：斜坡复位触发（RAMPSTOP）上升沿与斜坡启动触发（RAMPSTART）上升沿同时发生。

•斜坡启动触发（RAMPSTART）上升沿事件具有优先权，当延迟计数器减至 0 时，RAMPSTS 开始递减。

•斜坡复位触发（RAMPSTOP）上升沿事件会被忽略，不会使 RAMPSTS 停止。

情况 3：斜坡复位触发（RAMPSTOP）上升沿在斜坡启动触发（RAMPSTART）上升沿之后、但在延迟计数器减至 0 之前的一个或多个周期内发生。

•当延迟计数器减至 0 时，RAMPSTS 不会递减。

情况 4：在斜坡启动触发（RAMPSTART）上升沿触发后，当延迟计数器减至 0 的同时，斜坡复位触发（RAMPSTOP）上升沿发生。

•RAMPSTS 不会递减。

Puya Confidential

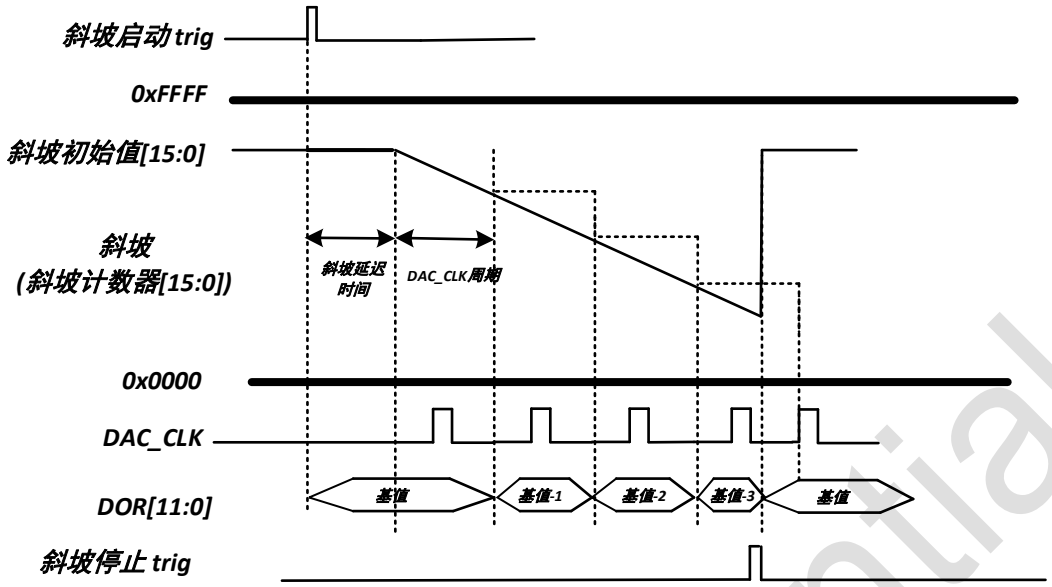


图 14-12 斜坡功能示意图

14.3.13 双 DAC 通道转换

在需要 2 个 DAC 同时工作的情况下，为了更有效地利用总线带宽，DAC 集成了 3 个供双 DAC 模式使用的寄存器：DHR8RD、DHR12RD 和 DHR12LD，只需要访问一个寄存器即可同时完成驱动 2 个 DAC 通道的操作。

对于双 DAC 通道转换和这些专用寄存器，共有 11 种转换模式可用。这些转换模式在只使用一个 DAC 通道的情况下，仍然可通过独立的 DAC_DHRx 寄存器操作。

所有模式详述于以下章节。

不使用波形发生器的独立触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’；
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为不同值，分别配置 2 个 DAC 通道的不同触发源；
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

当发生 DAC 通道 1 触发事件时，（延迟 3 个 APB1 时钟周期后）寄存器 DHR1 的值传入寄存器 DAC_DOR1 中。

当发生DAC通道2触发事件时，(延迟3个APB1时钟周期后)寄存器DHR2的值传入寄存器DAC_DOR2中。

使用相同 LFSR 的独立触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’；
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为不同值，分别配置 2 个 DAC 通道的不同触发源；
- 设置 2 个 DAC 通道的 DAC_CR.WAVEx[1:0]位为“01”，并设置 DAC_CR.MAMPx[3:0]为相同的 LFSR 屏蔽值；

Puya Confidential

- 将双 DAC 通道转换数据装入所需的 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

当发生 DAC 通道 1 触发事件时，具有相同屏蔽的 LFSR1 计数器值与 DAC_DHR1 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR1 中，然后更新 LFSR1 计数器。

当发生 DAC 通道 2 触发事件时，具有相同屏蔽的 LFSR2 计数器值与 DAC_DHR2 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR2 中，然后更新 LFSR2 计数器。

使用不同 LFSR 的独立触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’；
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为不同值，分别配置 2 个 DAC 通道的不同触发源；
- 设置 2 个 DAC 通道的 DAC_CR.WAVEx[1:0]位为“01”，并设 DAC_CR.MAMPx[3:0]为不同的 LFSR 屏蔽值；
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器（DHR12RD、DHR12LD 或者 DHR8RD）。

当发生 DAC 通道 1 触发事件时，按照 DAC_CR.MAMP1[3:0]所设屏蔽的 LFSR1 计数器值与 DAC_DHR1 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR1 中，然后更新 LFSR1 计数器。

当发生 DAC 通道 2 触发事件时，按照 DAC_CR.MAMP2[3:0]所设屏蔽的 LFSR2 计数器值与 DAC_DHR2 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR2 中，然后更新 LFSR2 计数器。

产生相同三角波的独立触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’；
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为不同值，分别配置 2 个 DAC 通道的不同触发源；
- 设置 2 个 DAC 通道的 DAC_CR.WAVEx[1:0]位为“1x”，并设 DAC_CR.MAMPx[3:0]为相同的三角波幅值；
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器(DHR12RD、DHR12LD 或 DHR8RD)。

当发生 DAC 通道 1 触发事件时，相同的三角波幅值加上 DAC_DHR1 寄存器值，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR1 中，然后更新 DAC 通道 1 三角波计数器。

当发生 DAC 通道 2 触发事件时，相同的三角波幅值加上 DAC_DHR2 寄存器值，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR2 中，然后更新 DAC 通道 2 三角波计数器。

产生不同三角波的独立触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’；
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为不同值，分别配置 2 个 DAC 通道的不同触发源；
- 设置 2 个 DAC 通道的 DAC_CR.WAVEx[1:0]位为“1x”，并设 DAC_CR.MAMPx[3:0]为不同的三角波幅值。
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

Puya Confidential

当发生 DAC 通道 1 触发事件时，DAC_CR.MAMP1[3:0]所设的三角波幅值加上 DAC_DHR1 寄存器值，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR1 中，然后更新 DAC 通道 1 三角波计数器。
当发生 DAC 通道 2 触发事件时，DAC_CR.MAMP2[3:0]所设的三角波幅值加上 DAC_DHR2 寄存器值，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR2 中，然后更新 DAC 通道 2 三角波计数器。

同时非硬件转换

按照下列过程设置 DAC 转换模式：

- 将双 DAC 通道转换数据装入所需的 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

在此配置下，一个 APB1 时钟周期后，DAC_DHR1 和 DAC_DHR2 寄存器的数值即被分别传入 DAC_DOR1 和 DAC_DOR2 寄存器中。

不使用波形发生器的同时触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’；
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为相同值，分别配置 2 个 DAC 通道使用相同触发源；
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

当发生触发事件时，延迟 3 个 APB1 时钟周期后，DAC_DHR1 和 DAC_DHR2 寄存器的数值分别传入 DAC_DOR1 和 DAC_DOR2 寄存器。

使用相同 LFSR 的同时触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’；
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为相同值，分别配置 2 个 DAC 通道使用相同触发源；
- 设置 2 个 DAC 通道的 DAC_CR.WAVEx[1:0]位为“01”，并设 DAC_CR.MAMPx[3:0]为相同的 LFSR 屏蔽值；
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器(DHR12RD、DHR12LD 或 DHR8RD)；

当发生触发事件时，DAC_CR.MAMP1[3:0]所设屏蔽的 LFSR1 计数器值与 DAC_DHR1 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入 DAC_DOR1 寄存器中，然后更新 LFSR1 计数器。
同样，DAC_CR.MAMP1[3:0]所设屏蔽的 LFSR2 计数器值与 DAC_DHR2 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR2 中，然后更新 LFSR2 计数器。

使用不同 LFSR 的同时触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’；
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为相同值，分别配置 2 个 DAC 通道使用相同触发源；
- 设置 2 个 DAC 通道的 DAC_CR.WAVEx[1:0]位为“01”，并设 DAC_CR.MAMPx[3:0]为不同的 LFSR 屏蔽值；
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

当发生触发事件时，具有相同屏蔽的 LFSR1 计数器值与 DAC_DHR1 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR1 中，然后更新 LFSR1 计数器。

Puya Confidential

同时，具有相同屏蔽的 LFSR2 计数器值与 DAC_DHR2 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR2 中，然后更新 LFSR2 计数器。

使用相同三角波发生器的同时触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’：
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为相同值，分别配置 2 个 DAC 通道使用相同触发源。
- 设置 2 个 DAC 通道的 DAC_CR.WAVEx[1:0]位为“1x”，并设 DAC_CR.MAMPx[3:0]为相同的三角波幅值。
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

当发生触发事件时，相同的三角波幅值与 DAC_DHR1 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR1 中，然后更新三角波计数器。

同时，相同的三角波幅值与 DAC_DHR2 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR2 中，然后更新三角波计数器。

使用不同三角波发生器的同时触发

按照下列顺序设置 DAC 转换模式：

- 分别设置 2 个 DAC 通道的触发使能位 DAC_CR.TEN1 和 DAC_CR.TEN2 为‘1’：
- 通过设置 DAC_CR.TSEL1[3:0]和 DAC_CR.TSEL2[3:0]位为相同值，分别配置 2 个 DAC 通道使用相同触发源。
- 设置 2 个 DAC 通道的 DAC_CR.WAVEx[1:0]位为“1x”，并设 DAC_CR.MAMPx[3:0]为不同的三角波幅值。
- 将双 DAC 通道转换数据装入所需的 DHR 寄存器(DHR12RD、DHR12LD 或 DHR8RD)。

当发生触发事件时，DAC_CR.MAMP1[3:0]所设的三角波幅值与 DAC_DHR1 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR1 中，然后更新三角波计数器。

同时，DAC_CR.MAMP2[3:0]所设的三角波幅值与 DAC_DHR2 寄存器值相加，延迟 3 个 APB1 时钟周期后，结果传入寄存器 DAC_DOR2 中，然后更新三角波计数器。

14.3.14 DAC 输出时钟

DAC 输出时钟（DAC_CLK）为模拟端采样数据时钟，该时钟只有在 DAC_CR.ENx 为“1”时有效。在选择硬件触发时，检测到硬件触发 TRGO 上升沿后的 5 个 APB1 周期置“1”，在保持 3 个 APB1 周期后清 0；在选择软件触发时，检测到软件触发上升沿后的 3 个 APB1 周期置“1”，在保持 3 个 APB1 周期后清 0；在选择不触发时，检测到数据变化的 2 个 APB1 周期置“1”，在保持 3 个 APB1 周期后清 0。

注意：当 TENx 从“1”变为“0”的 1 个 APB1 周期后，DAC_DOR 的数据会更新为 DAC_DHR 寄存器中的数据。当 DAC_DOR 更新为新值时，会产生一个 DAC_CLK；反之，则不会产生 DAC_CLK。

当需要配置 DAC 15MPS 采样率时，DAC_CLKCR 推荐配置如下表

f_{PCLK}	DACCLK_SETUP	DACCLK_WIDTH	DACCLK_HOLD	DAC_CLK 周期
200 MHz	5	3	5	13
176 MHz	4	3	4	11
144 MHz	3	3	3	9
120 MHz	2	3	3	8

Puya Confidential

14.4 DAC 寄存器

14.4.1 DAC 控制寄存器（DAC_CR）

地址偏移：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL2[3:0]				TEN2	Res.	EN2
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[1:0]		TSEL1[3:0]				TEN1	Res.	EN1
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW

Bit	Name	RW	Reset Value	Function
31	Reserved	-	-	保留
30	DMAUDRIE2	RW	0	**DMAUDRIE2：**DAC 通道 2 DMA 下溢中断使能 0：屏蔽 DAC 通道 2 DMA 下溢中断； 1：使能 DAC 通道 2 DMA 下溢中断。
29	DMAEN2	RW	0	**DMAEN2**：DAC 通道 2 DMA 使能 0：屏蔽 DAC 通道 2 DMA 模式； 1：使能 DAC 通道 2 DMA 模式。
28：25	MAMP2	RW	4’h0	**MAMP2[3：0]：**DAC 通道 2 屏蔽/增幅选择器 0000：不屏蔽 LSFR 位 0/三角波增幅值等于 1； 0001：不屏蔽 LSFR 位[1：0]/三角波增幅值等于 3； 0010：不屏蔽 LSFR 位[2：0]/三角波增幅值等于 7； 0011：不屏蔽 LSFR 位[3：0]/三角波增幅值等于 15； 0100：不屏蔽 LSFR 位[4：0]/三角波增幅值等于 31； 0101：不屏蔽 LSFR 位[5：0]/三角波增幅值等于 63； 0110：不屏蔽 LSFR 位[6：0]/三角波增幅值等于 127； 0111：不屏蔽 LSFR 位[7：0]/三角波增幅值等于 255； 1000：不屏蔽 LSFR 位[8：0]/三角波增幅值等于 511； 1001：不屏蔽 LSFR 位[9：0]/三角波增幅值等于 1023； 1010：不屏蔽 LSFR 位[10：0]/三角波增幅值等于 2047； ≥1011：不屏蔽 LSFR 位[11：0]/三角波增幅值等于 4095；
24：23	WAVE2	RW	2’h0	**WAVE2[1：0]：**DAC 通道 2 噪声/三角波生成使能 00：关闭波形发生器； 01：使能噪声波形发生器； 10：使能三角波发生器。 11：使能斜坡发生器
22：19	TSEL2	RW	4’h0	**TSEL2[3：0]：**DAC 通道 2 出发选择 有关触发器配置和映射的详细信息，见 DAC 触发选择 **update/reset。**
18	TEN2	RW	0	**TEN2**：DAC 通道 2 触发使能 0：关闭 DAC 通道 2 触发 1：使能 DAC 通道 2 触发
17	Reserved	-	-	保留
16	EN2	RW	0	**EN2**：DAC 通道 2 使能

Puya Confidential

				0：关闭 DAC 通道 2； 1：使能 DAC 通道 2；
15	Reserved	-	-	保留
14	DMAUDRIE1	RW	0	**DMAUDRIE1：**DAC 通道 1 DMA 下溢中断使能 0：屏蔽 DAC 通道 1 DMA 下溢中断； 1：使能 DAC 通道 1 DMA 下溢中断。
13	DMAEN1	RW	0	**DMAEN1**：DAC 通道 1 DMA 使能 0：屏蔽 DAC 通道 1 DMA 模式； 1：使能 DAC 通道 1 DMA 模式。
12：9	MAMP1	RW	4’h0	**MAMP1[3：0]：**DAC 通道 1 屏蔽/增幅选择器 由软件设置这些位，用来在噪声生成模式下选择屏蔽位，在三角波生成模式下选择波形的增幅值。 0000：不屏蔽 LSFR 位 0/三角波增幅值等于 1； 0001：不屏蔽 LSFR 位[1：0]/三角波增幅值等于 3； 0010：不屏蔽 LSFR 位[2：0]/三角波增幅值等于 7； 0011：不屏蔽 LSFR 位[3：0]/三角波增幅值等于 15； 0100：不屏蔽 LSFR 位[4：0]/三角波增幅值等于 31； 0101：不屏蔽 LSFR 位[5：0]/三角波增幅值等于 63； 0110：不屏蔽 LSFR 位[6：0]/三角波增幅值等于 127； 0111：不屏蔽 LSFR 位[7：0]/三角波增幅值等于 255； 1000：不屏蔽 LSFR 位[8：0]/三角波增幅值等于 511； 1001：不屏蔽 LSFR 位[9：0]/三角波增幅值等于 1023； 1010：不屏蔽 LSFR 位[10：0]/三角波增幅值等于 2047； ≥1011：不屏蔽 LSFR 位[11：0]/三角波增幅值等于 4095；
8：7	WAVE1	RW	2’h0	**WAVE1[1：0]：**DAC 通道 1 噪声/三角波生成使能 该 2 位由软件设置和清除。 00：关闭波形发生器； 01：使能噪声波形发生器； 10：使能三角波发生器。 11：使能斜坡发生器
6：3	TSEL1	RW	4’h0	**TSEL1[3：0]：**DAC 通道 1 出发选择 有关触发器配置和映射的详细信息，见 DAC 触发选择表格中 **update/reset。**
2	TEN1	RW	0	**TEN1**：DAC 通道 1 触发使能 0：关闭 DAC 通道 1 触发 1：使能 DAC 通道 1 触发
1	Reserved	-	-	保留
0	EN1	RW	0	**EN1**：DAC 通道 1 使能 0：关闭 DAC 通道 1； 1：使能 DAC 通道 1；

14.4.2 DAC 软件触发寄存器（DAC_SWTRIGR)

地址偏移：0x04

复位值：0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SW TRIG1
														W	W

Bit	Name	RW	Reset Value	Function
31:2	Reserved	-	-	保留
1	SWTRIG2	W	0	**SWTRIG2**：DAC 通道 2 软件触发 0：关闭 DAC 通道 2 软件触发； 1：使能 DAC 通道 2 软件触发。
0	SWTRIG1	W	0	**SWTRIG1**：DAC 通道 1 软件触发 0：关闭 DAC 通道 1 软件触发； 1：使能 DAC 通道 1 软件触发。

14.4.3 DAC 通道 1 的 12 位右对齐数据保持寄存器（DAC_DHR12R1）

地址偏移：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Res.	Res.	Res.	Res.	DACC1DHR[11:0]										
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:12	Reserved	-	-	保留
11:0	DACC1DHR[11:0]	RW	12’h0	**DACC1DHR[11:0]**：DAC 通道 1 的 12 位右对齐数据 该位由软件写入，表示 DAC 通道 1 的 12 位数据。

14.4.4 DAC 通道 1 的 12 位左对齐数据保持寄存器（DAC_DHR12L1）

地址偏移：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW				

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:4	DACC1DHR[11:0]	RW	12’h0	**DACC1DHR[11:0]**：DAC 通道 1 的 12 位左对齐数据 该位由软件写入，表示 DAC 通道 1 的 12 位数据。
3:0	Reserved	-	-	保留

Puya Confidential

14.4.5 DAC 通道 1 的 8 位右对齐数据保持寄存器（DAC_DHR8R1）

地址偏移：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7:0	DACC1DHR[7:0]	RW	8’h0	**DACC2DHR[7:0]**：DAC 通道 2 的 8 位右对齐数据 该位由软件写入，表示 DAC 通道 1 的 8 位数据。

14.4.6 DAC 通道 2 的 12 位右对齐数据保持寄存器（DAC_DHR12R2）

地址偏移：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Res.	Res.	Res.	Res.	DACC2DHR[11:0]										
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:12	Reserved	-	-	保留
11:0	DACC2DHR[11:0]	RW	12’h0	**DACC2DHR[11:0]**：DAC 通道 2 的 12 位右对齐数据 该位由软件写入，表示 DAC 通道 2 的 12 位数据。

14.4.7 DAC 通道 2 的 12 位左对齐数据保持寄存器（DAC_DHR12L2）

地址偏移：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW				

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:4	DACC2DHR[11:0]	RW	8’h0	**DACC2DHR[11:0]**：DAC 通道 2 的 12 位左对齐数据 该位由软件写入，表示 DAC 通道 2 的 12 位数据。
3:0	Reserved	-	-	保留

14.4.8 DAC 通道 2 的 8 位右对齐数据保持寄存器（DAC_DHR8R2）

地址偏移：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7:0	DACC2DHR[7:0]	RW	8'h0	**DACC2DHR[7:0]**：DAC 通道 2 的 8 位右对齐数据 该位由软件写入，表示 DAC 通道 2 的 8 位数据。

14.4.9 DAC 双 DAC 的 12 位右对齐数据保持寄存器（DAC_DHR12RD）

地址偏移：0x20

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:28	Reserved	-	-	保留
27:16	DACC2DHR[11:0]	RW	12'h0	**DACC2DHR[11:0]**：DAC 通道 2 的 12 位右对齐数据 该位由软件写入，表示 DAC 通道 2 的 12 位数据。
15:12	Reserved	-	-	保留
11:0	DACC1DHR[11:0]	RW	12'h0	**DACC1DHR[11:0]**：DAC 通道 1 的 12 位右对齐数据 该位由软件写入，表示 DAC 通道 1 的 12 位数据。

14.4.10 DAC 双 DAC 的 12 位左对齐数据保持寄存器（DAC_DHR12LD）

地址偏移：0x24

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW				

Bit	Name	RW	Reset Value	Function
31:20	DACC2DHR[11:0]	RW	12'h0	**DACC2DHR[11:0]**：DAC 通道 2 的 12 位左对齐数据 该位由软件写入，表示 DAC 通道 2 的 12 位数据。
19:16	Reserved	-	-	保留
15:4	DACC1DHR[11:0]	RW	12'h0	**DACC1DHR[11:0]**：DAC 通道 1 的 12 位左对齐数据 该位由软件写入，表示 DAC 通道 1 的 12 位数据。
3:0	Reserved	-	-	保留

14.4.11 DAC 双 DAC 的 8 位右对齐数据保持寄存器（DAC_DHR8RD）

地址偏移：0x28

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DACC2DHR[7:0]								DACC1DHR[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:8	DACC2DHR[7:0]	RW	8'h0	**DACC2DHR[7:0]**：DAC 通道 2 的 8 位右对齐数据 该位由软件写入，表示 DAC 通道 2 的 8 位数据。
7:0	DACC1DHR[7:0]	RW	8'h0	**DACC1DHR[7:0]**：DAC 通道 1 的 8 位右对齐数据 该位由软件写入，表示 DAC 通道 1 的 8 位数据。

14.4.12 DAC 通道 1 数据输出寄存器（DAC_DOR1）

地址偏移：0x2C

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Res.	Res.	Res.	DAC_CMPOUT	DACC1DOR[11:0]										
			R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:13	Reserved	-	-	保留
12	DAC_CMPOUT	R	0	DAC_CMPOUT:校准 DAC 时，调节 trimm 值直到 DAC_CMPOUT=1 时即为校准成功。
11:0	DACC1DOR[11:0]	R	12'h0	**DACC1DOR[11:0]**：DAC 通道 1 输出数据 该位只读，表示 DAC 通道 1 的输出数据。

14.4.13 DAC 通道 2 数据输出寄存器（DAC_DOR2）

地址偏移：0x30

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Res.	Res.	Res.	Res.	DACC2DOR[11:0]										
				R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:12	Reserved	-	-	保留
11:0	DACC2DOR[11:0]	R	12'h0	**DACC2DOR[11:0]**：DAC 通道 2 输出数据 该位只读，表示 DAC 通道 2 的输出数据。

14.4.14 DAC 状态寄存器（DAC_SR）

地址偏移：0x34

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res.	Res.	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		RC_W1													
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

		RC_W1													

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29	DMAUDR2	RC_W1	0	**DMAUDR2**：DAC 通道 2 DMA 下溢标志 该位由硬件置 1，软件写 1 清 0 0：DAC 通道 2 没有 DMA 下溢情况发生； 1：DAC 通道 2 有 DMA 下溢情况发生。 注意：只有在 DMAUDR2 置“1”后，软件才能写“1”清“0”
28:14	Reserved	-	-	保留
13	DMAUDR1	RC_W1	0	**DMAUDR1**：DAC 通道 1 DMA 下溢标志 该位由硬件置 1，软件写 1 清 0 0：DAC 通道 1 没有 DMA 下溢情况发生； 1：DAC 通道 1 有 DMA 下溢情况发生。 注意：只有在 DMAUDR1 置“1”后，软件才能写“1”清“0”
12:0	Reserved	-	-	保留

14.4.15 DAC 输出选择寄存器（DAC_OC)

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DAC_OS_TRIMCR[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	DAC_EXTCH_CR[1:0]		DAC_CALMODE[1:0]		DAC_INTER2_CR	DAC_INTER1_CR	Res.	DAC_VREFSEL	DAC_EXTSRC_CR	DAC_EXTEN	
					RW	RW	RW	RW	RW	RW		RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29:24	DAC_OS_TRIMCR[5:0]	RW	6’h20	DAC CH1、CH2 共享 BUFFER 校准寄存器
23:11	Reserved	-	-	保留
10:9	DAC_EXTCH_CR[1:0]	RW	2’h0	DAC 电压输出,选择 00：DAC_VOUT_EXT 01：DAC_VOUT1_EXT 10：DAC_VOUT2_EXT 11：同 00
8:7	DAC_CALMODE[1:0]	RW	2’h0	DAC 校准选择 00：DAC offset 校准禁止 01：DAC offset 校准用 ADC 10：DAC offset 校准用内部 COMP 11：同 00
6	DAC_INTER2_CR	RW	0	DAC CH2 内部输出使能 0:禁止 DAC CH2 内部输出 1:使能 DAC CH2 内部输出
5	DAC_INTER1_CR	RW	0	DAC CH1 内部输出使能 0:禁止 DAC CH1 内部输出

Puya Confidential

				1:使能 DAC CH1 内部输出
4	Reserved	-	-	保留
3	DAC_VREFSEL	RW	0	DAC CH1、CH2 参考电压选择： 0：选择 V_{CC} 作为 DAC CHx 参考电压 1：选择 V_{REFBUF} 作为 DAC CHx 参考电压
2	DAC_EXTSRC_CR	RW	0	DAC 外部输出选择 0：DAC1_VOUT 连到 BUFFER 1：DAC2_VOUT 连到 BUFFER
1:0	DAC_EXTEN	RW	2'h0	DAC 外部输出选择 00：外部输出禁止 01：外部输出使能且经过输出 BUFFER 10：外部输出使能且不经过输出 BUFFER 其他=00

14.4.16 DAC 通道 1 斜坡配置寄存器（DAC_RAMPCFGR1）

偏移地址：0x58

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RAMPHSTEPVALS1[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	RAMPREFS1[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:28	Reserved	-	-	保留
27:16	RAMPHSTEPVALS1[11:0]	RW	12'h0	DAC 通道 1 斜坡步进值
15:12	Reserved	-	-	保留
11:0	RAMPREFS1[11:0]	RW	12'h0	DAC 通道 1 斜坡基值

14.4.17 DAC 通道 2 斜坡配置寄存器（DAC_RAMPCFGR2）

偏移地址：0x5C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RAMPHSTEPVALS2[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	RAMPREFS2[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:28	Reserved	-	-	保留
27:16	RAMPHSTEPVALS2[11:0]	RW	12'h0	DAC 通道 2 斜坡步进值
15:12	Reserved	-	-	保留
11:0	RAMPREFS2[11:0]	RW	12'h0	DAC 通道 2 斜坡基值

14.4.18 DAC 斜坡延迟配置寄存器（DAC_RAMPDLYS）

偏移地址：0x64

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	RAMPDEALYDATA2 [11:0]												RAMPDELAY_EN2

Puya Confidential

			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	RAMPDELAYDATA1 [11:0]												RAMPDELAY_EN1
			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:29	Reserved	-	-	保留
28:17	RAMPDEALYDATA2 [11:0]	RW	12'h0	DAC CH2 斜坡延迟值
16	RAMPDELAY_EN2	RW	0	DAC CH2斜坡延迟使能 1：启动斜坡延迟 0：关闭斜坡延迟
15:13	Reserved	-	-	保留
12:1	RAMPDELAYDATA1 [11:0]	RW	12'h0	DAC CH1 斜坡延迟值
0	RAMPDELAY_EN1	RW	0	DAC CH1斜坡延迟使能 1：启动斜坡延迟 0：关闭斜坡延迟

14.4.19 DAC 斜坡控制寄存器（DAC_RAMPCR）

偏移地址：0x68

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RAMPSTPTRIGSEL2[3:0]				RAMPUPTRIGSEL2[3:0]				RAMPLOADSEL2
							RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RAMPSTPTRIGSEL1[3:0]				RAMPUPTRIGSEL1[3:0]				RAMPLOADSEL1
							RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24:21	RAMPSTPTRIGSEL2[3:0]	RW	4'h0	RAMPSTPTRIGSEL2[3:0]：DAC通道2斜坡停止触发选择 有关触发器配置和映射的详细信息，见DAC触发选择update/reset。 映射与 TSEL2[3:0]相同。
20:17	RAMPUPTRIGSEL2[3:0]	RW	4'h0	RAMPUPTRIGSEL2[3:0]：DAC通道2斜坡更新触发选择 有关触发配置和映射的详细信息，见DAC触发inc。
16	RAMPLOADSEL2	RW	0	RAMPLOADSEL2：DAC通道2斜坡基值选择 0：斜坡递减计数基值从斜坡基值影子寄存器加载 1：斜坡递减计数基值从斜坡基值寄存器加载
15:9	Reserved	-	-	保留
8:5	RAMPSTPTRIGSEL1[3:0]	RW	4'h0	RAMPSTPTRIGSEL1[3:0]：DAC通道1斜坡停止触发选择 有关触发器配置和映射的详细信息，见DAC触发update/reset。 映射与 TSEL1[3:0]相同。
4:1	RAMPUPTRIGSEL1[3:0]	RW	4'h0	RAMPUPTRIGSEL1[3:0]：DAC通道1斜坡更新触发选择

Puya Confidential

				有关触发配置和映射的详细信息，见DAC触发inc。
0	RAMPLOADSEL1	RW	0	RAMPLOADSEL1：DAC通道1斜坡基值选择 0：斜坡递减计数基值从斜坡基值影子寄存器加载 1：斜坡递减计数基值从斜坡基值寄存器加载

14.4.20 DAC 时钟控制寄存器（DAC_CLKCR)

偏移地址：0x6C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACCLK_HOLD[2:0]			DACCLK_WIDTH[2:0]			DACCLK_SETUP[2:0]		
							RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8:6	DACCLK_HOLD[2:0]	RW	3’h3	DAC CLK 保持时间配置寄存器
5:3	DACCLK_WIDTH[2:0]	RW	3’h3	DAC CLK 高电平时间配置寄存器
2:0	DACCLK_SETUP[2:0]	RW	3’h2	DAC CLK 启动时间配置寄存器

Puya Confidential

15. 电压参考缓冲器(V_{REFBUF})

15.1 V_{REFBUF} 简介

本芯片内置有电压参考缓冲器 V_{REFBUF}，用作 ADC 参考电压，同时也可以通过 PA0 输出该电压。

15.2 V_{REFBUF} 功能描述

电压参考缓冲器 V_{REFBUF} 支持五档电压，可通过 VREFBUF_CR 的 VREFBUFSEL 配置:

- VREFBUFSEL=000：1.024 V
- VREFBUFSEL =001：2.048 V
- VREFBUFSEL =010：2.5 V
- VREFBUFSEL =011：2.9 V
- VREFBUFSEL =100：4.096 V

通过置位 VREFBUF_CR 的 VREFBUFEN 使能 V_{REFBUF}，用户必须设置了 VREFBUFSEL，V_{REFBUF} 输出即达到期望值。

15.3 V_{REFBUF} 寄存器

15.3.1 V_{REFBUF} 控制寄存器（VREFBUF_CR)

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VREFBUFSEL[2:0]			VREFBUFEN
												RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:4	Reserved	-	-	保留
3:1	VREFBUFSEL	RW	3'h0	V_{REFBUF} 电压档位选择 000：1.024V 001：2.048V 010：2.5V 011：2.9V 100：4.096V 其他：保留
0	VREFBUFEN	RW	0	V_{REFBUF} 使能，当 VREFBUFEN==0 时，电压输出高阻 软件写 0 置 0，写 1 置 1， 0：禁用 V_{REFBUF} 1：使能 V_{REFBUF}

Puya Confidential

16. 比较器（COMP）

16.1 COMP 简介

芯片内集成 4 个通用比较器（General purpose comparators）COMP，分别是 COMP1、COMP2、COMP3 和 COMP4。这 4 个模块可以作为单独的模块，也可以与 timer 组合在一起使用。

用作比较器模拟功能的 I/O，必须在 GPIO 寄存器中被配置成模拟模式。

比较器可以使用在：

- 低功耗模式唤醒
- 模拟信号调节
- 和定时器的 PWM 输出连接时，构成逐周期电流控制回路

16.2 COMP 主要特性

- 支持电压比较功能，每个比较器有可配置的正极或者负极输入，以实现灵活的电压选择
  - 多路 I/O 引脚
  - DAC
  - 温度传感器输出
- 可编程速度和功耗
- 可编程迟滞功能
- 配置寄存器写保护（LOCK 功能）
- 输出可以被连接到 I/O 或者 timer 的输入
  - OCREF_CLR 事件（逐周期电流控制）
  - 为快速 PWM 关断的刹车
- 每个 COMP 具有中断产生能力，用作芯片从低功耗模式（睡眠/低功耗睡眠/停止模式）唤醒（通过 EXTI）
- 提供软件可配置数字滤波时间以增强芯片抗干扰能力
- 支持输出消隐以降低开关噪声
- 支持窗口比较功能

Puya Confidential

16.3 COMP 功能描述

16.3.1 COMP 模块框图

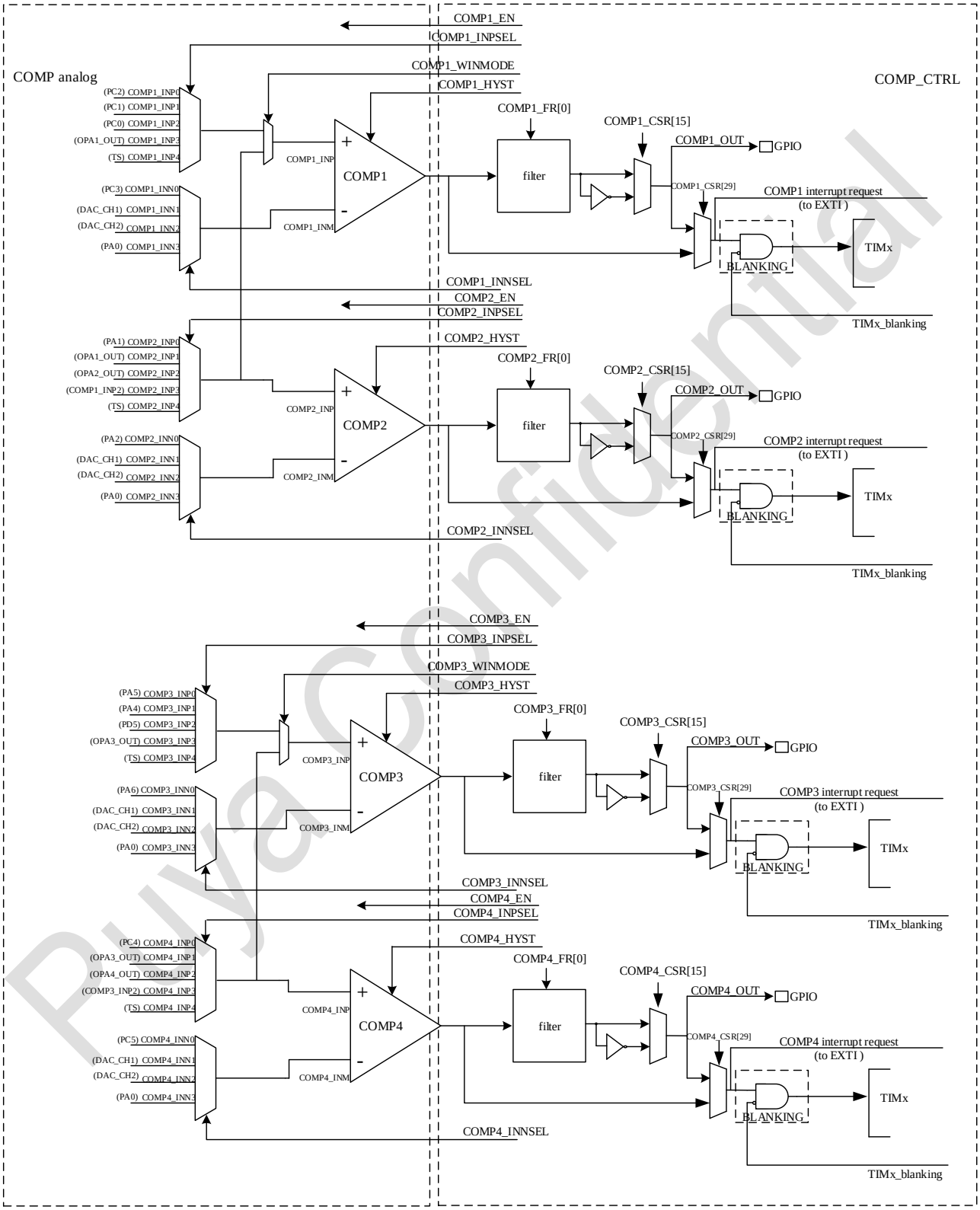


图 16-1 COMP 结构框图

Puya Confidential

16.3.2 COMP 的输入和输出

比较器输出可以通过在 GPIO 的可选功能连接到 I/O 引脚。

输出也可以在内部连接到各种定时器的输入，达到以下目的：

- 连接刹车输入时，PWM 信号的紧急关断刹车
- 使用 OCREF_CLR 输入的 Cycle-by-cycle 电流控制
- 时序测量的输入捕获

比较器输出到其他模块的连接关系，见下表。

表 16-1 COMP 模块数字输出

COMP1_OUT	COMP2_OUT	COMP3_OUT	COMP4_OUT
PA0	PA2	-	-
PA6	PA7	-	-
PA10	PA10	PA10	-
PA11	PA12	-	-
PB8	PB9	PB2	-
PB11	PB10	-	-
PB13	PB14	PB15	PC6
PD1	PD1	PD1	PD1
PD11	PD11	PD11	PD11
TIM1_ETR	TIM1_ETR	TIM1_ETR	TIM1_ETR
TIM2_ETR	TIM2_ETR	TIM2_ETR	TIM2_ETR
TIM3_ETR	TIM3_ETR	TIM3_ETR	TIM3_ETR
TIM8_ETR	TIM8_ETR	TIM8_ETR	TIM8_ETR
TIM1_OCREF_CLR0	TIM1_OCREF_CLR1	TIM1_OCREF_CLR2	TIM1_OCREF_CLR3
TIM2_OCREF_CLR0	TIM2_OCREF_CLR1	TIM2_OCREF_CLR2	TIM2_OCREF_CLR3
TIM3_OCREF_CLR0	TIM3_OCREF_CLR1	TIM3_OCREF_CLR2	TIM3_OCREF_CLR3
TIM8_OCREF_CLR0	TIM8_OCREF_CLR1	TIM8_OCREF_CLR2	TIM8_OCREF_CLR3
TIM1_TI1_IN1	TIM1_TI1_IN2	TIM1_TI1_IN3	TIM1_TI1_IN4
TIM2_TI1_IN1	TIM2_TI1_IN2	TIM2_TI1_IN3	TIM2_TI1_IN4
TIM3_TI1_IN1	TIM3_TI1_IN2	TIM3_TI1_IN3	TIM3_TI1_IN4
TIM8_TI1_IN1	TIM8_TI1_IN2	TIM8_TI1_IN3	TIM8_TI1_IN4
TIM1_TI2_IN1	TIM1_TI2_IN2	TIM1_TI2_IN3	TIM1_TI2_IN4
TIM2_TI2_IN1	TIM2_TI2_IN2	TIM2_TI2_IN3	TIM2_TI2_IN4
TIM3_TI2_IN1	TIM3_TI2_IN2	TIM3_TI2_IN3	TIM3_TI2_IN4
TIM8_TI2_IN1	TIM8_TI2_IN2	TIM8_TI2_IN3	TIM8_TI2_IN4
TIM1_TI3_IN1	TIM1_TI3_IN2	TIM1_TI3_IN3	TIM1_TI3_IN4
TIM2_TI3_IN1	TIM2_TI3_IN2	TIM2_TI3_IN3	TIM2_TI3_IN4
TIM3_TI3_IN1	TIM3_TI3_IN2	TIM3_TI3_IN3	TIM3_TI3_IN4
TIM8_TI3_IN1	TIM8_TI3_IN2	TIM8_TI3_IN3	TIM8_TI3_IN4
TIM1_TI4_IN1	TIM1_TI4_IN2	TIM1_TI4_IN3	TIM1_TI4_IN4
TIM2_TI4_IN1	TIM2_TI4_IN2	TIM2_TI4_IN3	TIM2_TI4_IN4
TIM3_TI4_IN1	TIM3_TI4_IN2	TIM3_TI4_IN3	TIM3_TI4_IN4
TIM8_TI4_IN1	TIM8_TI4_IN2	TIM8_TI4_IN3	TIM8_TI4_IN4
TIM1_BKIN1_CMP1	TIM1_BKIN1_CMP2	TIM1_BKIN1_CMP3	TIM1_BKIN1_CMP4
TIM2_BKIN1_CMP1	TIM2_BKIN1_CMP2	TIM2_BKIN1_CMP3	TIM2_BKIN1_CMP4
TIM3_BKIN1_CMP1	TIM3_BKIN1_CMP2	TIM3_BKIN1_CMP3	TIM3_BKIN1_CMP4
TIM8_BKIN1_CMP1	TIM8_BKIN1_CMP2	TIM8_BKIN1_CMP3	TIM8_BKIN1_CMP4
TIM1_BKIN2_CMP1	TIM1_BKIN2_CMP2	TIM1_BKIN2_CMP3	TIM1_BKIN2_CMP4
TIM2_BKIN2_CMP1	TIM2_BKIN2_CMP2	TIM2_BKIN2_CMP3	TIM2_BKIN2_CMP4
TIM3_BKIN2_CMP1	TIM3_BKIN2_CMP2	TIM3_BKIN2_CMP3	TIM3_BKIN2_CMP4
TIM8_BKIN2_CMP1	TIM8_BKIN2_CMP2	TIM8_BKIN2_CMP3	TIM8_BKIN2_CMP4

16.3.3 COMP 的时钟和复位

COMP 模块有两个时钟源：

1. PCLK（APB 时钟），用于配置寄存器

2. COMP 时钟，用于模拟比较器输出后的电路（模拟输出的锁存电路、滤毛刺电路等）的时钟，可选择为 PCLK、LSE 或者 LSI。当需要在停止模式下工作时，选择 LSE 或者 LSI。

Puya Confidential

COMP 模块的复位信号源有：

1．APB 复位，系统复位

2．COMP 模块软复位。

16.3.4 COMP 的滤波功能

如果芯片的工作环境恶劣，比较器的输出会出现噪声信号。使能数字滤波模块，则比较器的输出波形中脉宽小于 COMPx_FR.FLTCNT（x=1、2、3、4）设定时间的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输出信号与输入信号相同。

注意：设置 COMP 滤波时间，开启滤波使能应在 COMPx_CSR.EN 使能前完成。

滤波示意如下所示：

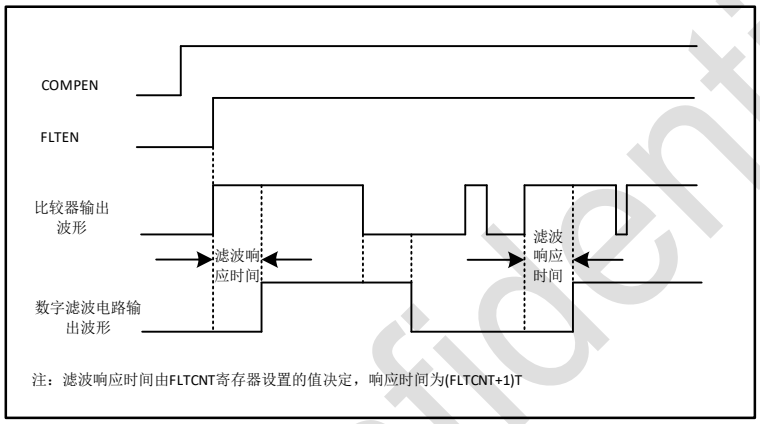


图 16-2 COMP 滤波示意图

16.3.5 COMP 的锁定机制

比较器可以用于安全目的，例如过电流或热保护。对于具有特定功能安全要求的应用程序，有必要确保比较器编程在伪寄存器访问或程序反损坏的情况下不会被更改。为此，比较器控制寄存器和状态寄存器可以被写保护（只读）。一旦编程完成，就可以设置 COMPx_CSR.LOCK（x=1、2、3、4）位。这导致整个寄存器变为只读，包括 COMPx_CSR.LOCK（x=1、2、3、4）位。写入保护只能通过 MCU 复位来移除。

16.3.6 COMP 的输出消隐

消隐功能的目的是防止电流在调节 PWM 周期开始时因短电流尖峰而跳闸（通常是电源开关反并联二极管中的恢复电流）。通过设置定时器的输出比较信号来实现消隐功能。每个比较器通道的消隐源由软件通过相应 COMPx_CSR（x=1、2、3、4）寄存器的 BLANKSEL[2:0]位单独选择。反相消隐信号与比较器输出进行逻辑“与”运算，以产生比较器通道的输出。

Puya Confidential

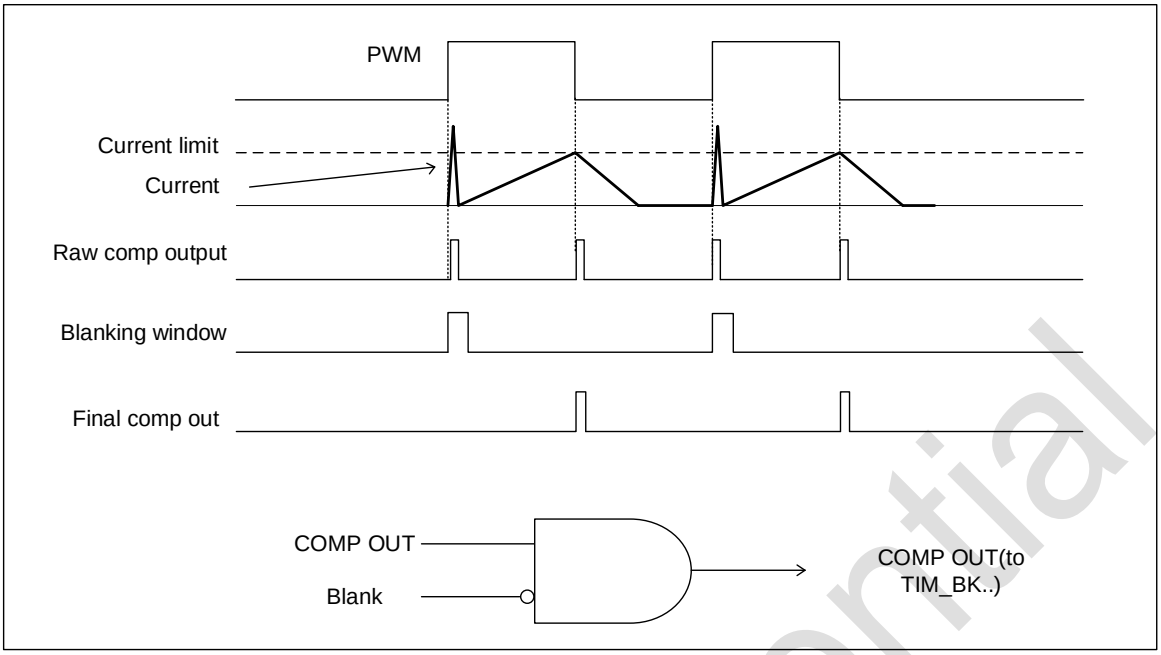


图 16-3 比较器消隐

16.3.7 COMP 的窗口模式

处于窗口模式的比较器简称为窗口比较器，窗口比较器的作用是监测模拟电压是否在低和高阈值范围内。可以使用 COMP1 和 COMP2 或者 COMP3 和 COMP4 创建窗口比较器。被监测的模拟电压同时连接到两个比较器的正输入端，高阈值和低阈值分别连接到两个比较器的负输入端。

通过使能 COMP1_CSR.WINMODE 或 COMP3_CSR.WINMODE 位，可以将 COMP1 和 COMP2 或者 COMP3 和 COMP4 的正输入端连接到一起，起到节省一个 I/O 引脚的作用。

例如当 COMP1_CSR.WINMODE 被置 1 时，比较器 1 的正输入端将接入 COMP2_INP，当 WINMODE 被清 0 时，COMP2 的正输入端将接入 COMP2_INP。这样做可以省掉一个 IO 的引脚。

Puya Confidential

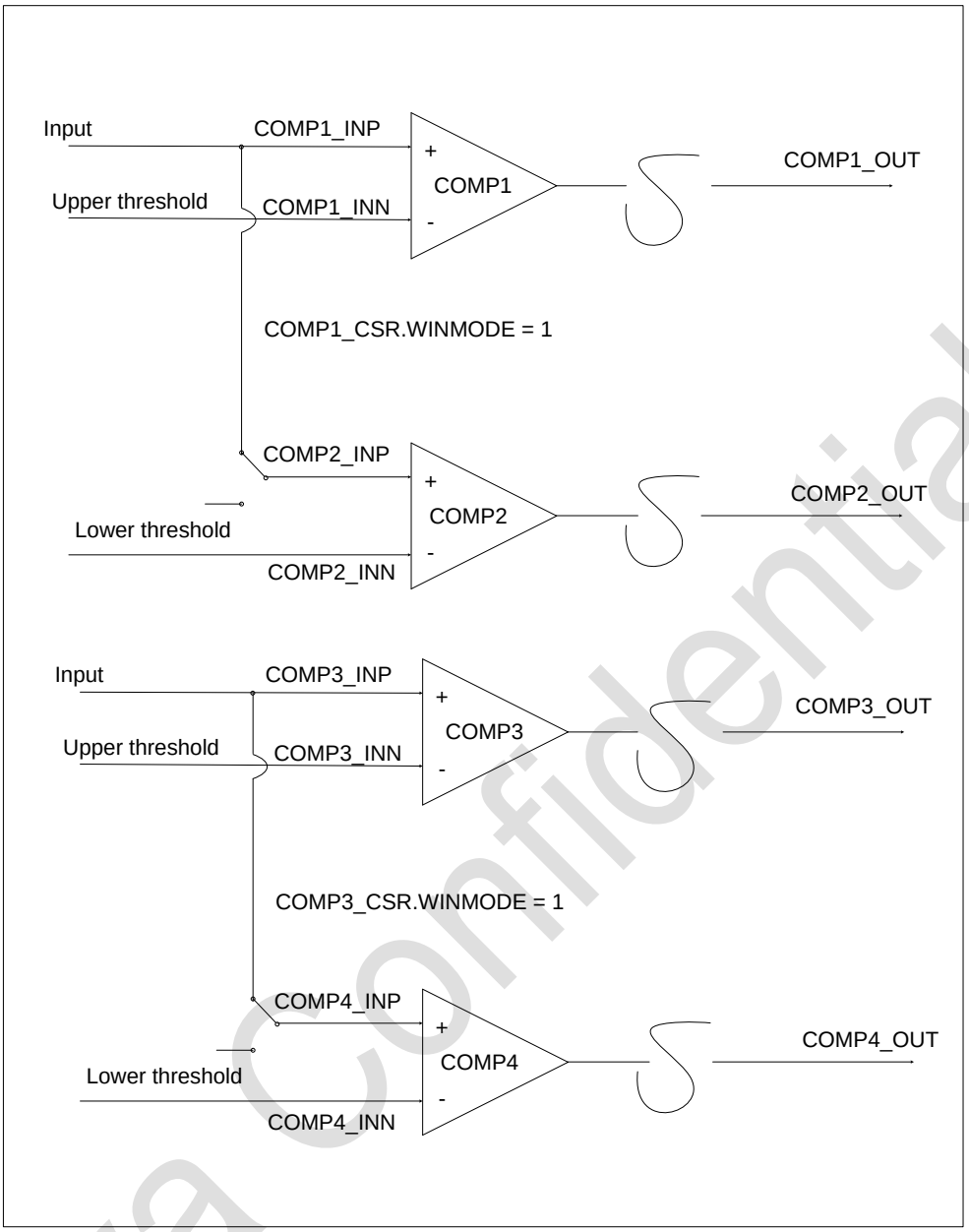


图 16-4 窗口比较器

16.3.8 COMP 的迟滞功能

比较器具有可编程的迟滞，可避免在有噪声信号的情况下出现虚假的输出跳变。 如果不需要迟滞模式（例如，从低功耗模式退出时）则可以将其禁用，以便能够使用外部组件强制迟滞值。COMP1、COMP2、COMP3 和 COMP4 有独立的迟滞功能使能信号 COMP1_HYST、COMP2_HYST、COMP3_HYST 和 COMP4_HYST。

Puya Confidential

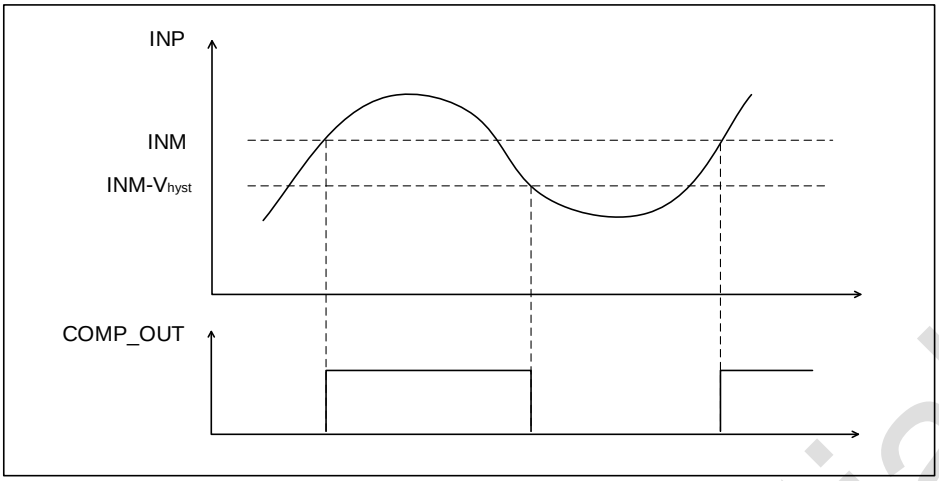


图 16-5 比较器的迟滞示意图

16.3.9 COMP 的低功耗模式

模式	描述
睡眠模式	对 COMP 无影响。 比较器中断可以使设备退出睡眠模式
低功耗运行模式	无影响
低功耗睡眠模式	无影响。 比较器中断可以使设备退出低功耗睡眠模式
停止模式 0	对比较器无影响 比较器中断可使器件退出停止 0 模式
停止模式 1	对比较器无影响 比较器中断可使器件退出停止 1 模式
停止模式 2	比较器不工作
待机模式	COMP 寄存器掉电，在退出待机模式后必须重新初始化

16.4 COMP 中断

比较器的输出在芯片内部连接到 EXTI 控制器（extended interrupts and events）。每个比较器有它自己的 EXTI 线，并能够产生中断或者事件。相同的机制被用作低功耗唤醒。

16.5 COMP 寄存器

16.5.1 COMP1 控制和状态寄存器（COMP1_CSR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	COMP1_OUT	TIM_EXTI_OUT_SEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WINMODE	PWRMODE	Res.	HYST
RW	R	RW										RW	RW		RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY	BLANKSEL[3:0]				INPSEL[2:0]			Res.	Res.	Res.	INNSEL[1:0]		Res.	Res.	EN
RW	RW	RW	RW	RW	RW	RW	RW				RW	RW			RW

Bit	Name	RW	Reset Value	Function
31	LOCK	RW	0	COMP1_CSR 寄存器锁 该位软件设置，硬件系统复位或模块软复位清除。锁住整个 COMP1_CSR，当锁住时，所有位只能读不能写，当解锁后，软件可以写控制位。 0：未锁定 1：锁定
30	COMP1_OUT	R	0	COMP1 输出状态 该位只读，它反映了 COMP1 在经过极性选择的输出电平。
29	TIM_EXTI_OUT_SEL	RW	0	COMP1 输出到 TIM/EXTI 选择。 0：不经过滤波，模拟端比较结果（可以消隐）直接输出到 TIM/EXTI 1：比较器内部数字输出选择与外部输出一致（可以滤波，可以消隐）
28:20	Reserved	-	-	保留
19	WINMODE	RW	0	COMP1 窗口模式使能 0：关闭窗口模式; 1：使能窗口模式，COMP1 的正极输入连接 COMP2 的正极输入
18	PWRMODE	RW	0	COMP1 功耗模式选择 0：高速 1：中速
17	Reserved	-	-	保留
16	HYST	RW	0	COMP1 迟滞功能使能控制 0：关闭迟滞功能 1：使能迟滞功能
15	POLARITY	RW	0	COMP1 输出极性选择 软件可读可写 0：不反向 1：反向
14：11	BLANKSEL	RW	4’h0	COMP1 消隐信号选择 该位由软件选择比较器 1 的消隐源。 0000：无消隐功能 0001：tim1_oc5 0010：tim2_oc3 0011：tim3_oc3 0100：tim8_oc5 0101：EPWM1 0110：EPWM2 0111：EPWM3 1000：EPWM4 1001：EPWM5 1010：EPWM6

Puya Confidential

				其他：保留
10:8	INPSEL	RW	3'h0	COMP1 正极输入选择 000：PC2 001：PC1 010：PC0 011：OPA1 100：TS 其他：保留
7:5	Reserved	-	-	保留
4:3	INNSEL	RW	2'h0	COMP1 负极输入选择 00：PC3 01：DAC_CH1 10：DAC_CH2 11：PA0(COMP123_VINN)
2:1	Reserved	-	-	保留
0	EN	RW	0	COMP1 使能位 软件可读可写（如果没有被锁定） 0：关闭 COMP1 1：使能 COMP1

16.5.2 COMP1 滤波寄存器（COMP1_FR）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTCNT1[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLTEN1
															RW

Bit	Name	RW	Reset Value	Function
31:16	FLTCNT1[15:0]	RW	16'h0	比较器 1 采样滤波计数器 采样时钟为 APB 或 LSI 或 LSE。滤波计数值可配置。采样次数达到滤波计数值时，结果一致输出。 采样计数周期=FLTCNT[15:0]
15:1	Reserved	-	-	保留
0	FLTEN1	RW	0	比较器 1 数字滤波功能配置 0：禁止数字滤波功能 1：使能数字滤波功能 Note: 该位必须在 COMP1 的 EN 为 0 时置位

16.5.3 COMP2 控制和状态寄存器（COMP2_CSR）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	COMP2_OUT	TIM_EXTI_OUT_SEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWRMODE	Res.	HYST
RW	R	RW											RW		RW

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY	BLANKSEL[3:0]				INPSEL[2:0]			Res.	Res.	Res.	INNSEL[1:0]		Res.	Res.	EN
RW	RW	RW	RW	RW	RW	RW	RW				RW	RW			RW

Bit	Name	RW	Reset Value	Function
31	LOCK	RW	0	COMP2_CSR 寄存器锁 该位软件设置，硬件系统复位或模块软复位清除。锁住整个 COMP2_CSR，当锁住时，所有位只能读不能写，当解锁后，软件可以写控制位。 0：未锁定 1：锁定
30	COMP2_OUT	R	0	COMP2 输出状态 该位只读，它反映了 COMP2 在经过极性选择的输出电平。
29	TIM_EXTI_OUT_SEL	RW	0	COMP2 输出到 TIM/EXTI 选择。 0：不经过滤波，ana_out（可以消隐）直接输出到 TIM/EXTI 1：选择输出到 PAD 上的 comp_out（可以滤波，可以消隐）输出到 TIM/EXTI
28：19	Reserved	-	-	保留
18	PWRMODE	RW	0	COMP2 功耗模式选择 选择了功耗和由此而来的 COMP2 的速度 0：高速 1：中速
17	Reserved	-	-	保留
16	HYST	RW	0	COMP2 迟滞功能使能控制 0：关闭迟滞功能 1：使能迟滞功能
15	POLARITY	RW	0	COMP2 极性选择 0：不反向 1：反向
14:11	BLANKSEL	RW	4’h0	COMP2 消隐信号选择 该位由软件选择比较器 2 的消隐源。 0000：无消隐功能 0001：tim1_oc5 0010：tim2_oc3 0011：tim3_oc3 0100：tim8_oc5 0101：EPWM1 0110：EPWM2 0111：EPWM3 1000：EPWM4 1001：EPWM5 1010：EPWM6 其他：保留

Puya Confidential

10：8	INPSEL	RW	3’h0	COMP2 非反向输入的信号选择 000：PA1 001：OPA1 010：OPA2 011：COMP1_INP2 100：TS 其他：保留
7:5	Reserved	-	-	保留
4:3	INNSEL	RW	2’h0	COMP2 反向输入的信号选择 00：PA2 01：DAC_CH1 10：DAC_CH2 11：PA0（COMP123）
2:1	Reserved	-	-	保留
0	EN	RW	0	COMP2 使能位 0：关闭 COMP2 1：使能 COMP2

16.5.4 COMP2 滤波寄存器（COMP2_FR）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTCNT2[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLTEN2
															RW

Bit	Name	RW	Reset Value	Function
31:16	FLTCNT2	RW	16’h0	比较器 2 采样滤波计数器 采样时钟为 APB 或 LSI 或 LSE。滤波计数值可配置。采样次数达到滤波计数值时，结果一致输出。 采样计数周期=FLTCNT[15:0]
15:1	Reserved	-	-	保留
0	FLTEN2	RW	0	比较器 2 数字滤波功能配置 0：禁止数字滤波功能 1：使能数字滤波功能 注：该位必须在 COMP2 的 EN 为 0 时置位

16.5.5 COMP3 控制和状态寄存器（COMP3_CSR）

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	COMP3_OUT	TIM_EXTI_OUT_SEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WINMODE	PWRMODE	Res.	HYST
RW	R	RW										RW	RW		RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POLARITY	BLANKSEL[3:0]				INPSEL[2:0]			Res.	Res.	Res.	INNSEL[1:0]		Res.	Res.	EN
RW	RW	RW	RW	RW	RW	RW	RW				RW	RW			RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31	LOCK	RW	0	COMP3_CSR 寄存器锁定 该位软件设置，硬件系统复位或模块软复位清除。锁住整个 COMP3_CSR，当锁住时，所有位只能读不能写，当解锁后，软件可以写控制位。 0：未锁定 1：锁定
30	COMP3_OUT	R	0	COMP3 输出状态 该位只读，它反映了 COMP3 在经过极性选择的输出电平。
29	TIM_EXTI_OUT_SEL	RW	0	COMP3 输出到 TIM/EXTI 选择。 0：不经过滤波，ana_out（可以消隐）直接输出到 TIM/EXTI 1：选择输出到 PAD 上的 comp_out（可以滤波，可以消隐）输出到 TIM/EXTI
28：20	Reserved	-	-	保留
19	WINMODE	RW	0	COMP3 窗口模式使能 0：关闭窗口模式； 1：使能窗口模式，COMP3 的正极输入连接 COMP4 的正极输入
18	PWRMODE	RW	0	COMP3 功耗模式选择 选择了功耗和由此而来的 COMP3 的速度 0：高速 1：中速
17	Reserved	-	-	保留
16	HYST	RW	0	COMP3 迟滞功能使能控制 0：关闭迟滞功能 1：使能迟滞功能
15	POLARITY	RW	0	COMP3 极性选择 0：不反向 1：反向
14:11	BLANKSEL	RW	4’h0	COMP3 消隐信号选择 该位由软件选择比较器 3 的消隐源。 0000：无消隐功能 0001：tim1_oc5 0010：tim2_oc3 0011：tim3_oc3 0100：tim8_oc5 0101：EPWM1 0110：EPWM2 0111：EPWM3 1000：EPWM4 1001：EPWM5 1010：EPWM6

Puya Confidential

				其他：保留
10：8	INPSEL	RW	3’h0	COMP3 正极输入选择 000：PA5 001：PA4 010：PD5 011：OPA3 100：TS 其他：保留
7:5	Reserved	-	-	保留
4:3	INNSEL	RW	2’h0	COMP3 反向输入的信号选择 00：PA6 01：DAC_CH1 10：DAC_CH2 11：PA0（COMP123）
2:1	Reserved	-	-	保留
0	EN	RW	0	COMP3 使能位 0：关闭 COMP3 1：使能 COMP3

16.5.6 COMP3 滤波寄存器（COMP3_FR）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTCNT3[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLTEN3
															RW

Bit	Name	RW	Reset Value	Function
31:16	FLTCNT3	RW	16’h0	比较器 3 采样滤波计数器 采样时钟为 APB 或 LSI 或 LSE。滤波计数值可配置。采样次数达到滤波计数值时，结果一致输出。 采样计数周期=FLTCNT[15:0]
15:1	Reserved	-	-	保留
0	FLTEN3	RW	0	比较器 3 数字滤波功能配置 0：禁止数字滤波功能 1：使能数字滤波功能 注：该位必须在 COMP3 的 EN 为 0 时置位

16.5.7 COMP4 控制和状态寄存器（COMP4_CSR）

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	COMP2_OUT	TIM_EXTI_OUT_SEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWRMODE	Res.	HYST
RW	R	RW											RW		RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POLARITY	BLANKSEL[2:0]			Res.	INPSEL[2:0]			Res.	Res.	Res.	INNSEL[1:0]		Res.	Res.	EN

Puya Confidential

RW	RW	RW	RW		RW	RW	RW				RW	RW			RW

Bit	Name	RW	Reset Value	Function
31	LOCK	RW	0	COMP4_CSR 寄存器锁定 软件置位，系统复位清零。当被置位，则会锁定 COMP4_CSR 寄存器的所有 32 位 0：未锁定，可读写整个寄存器 1：锁定，整个寄存器只读
30	COMP4_OUT	R	0	COMP4 输出状态 该位只读，它反映了 COMP4 在经过极性选择的输出电平。
29	TIM_EXTI_OUT_SEL	RW	0	COMP4 输出到 TIM/EXTI 选择。 0：不经过滤波，ana_out（可以消隐）直接输出到 TIM/EXTI 1：选择输出到 PAD 上的 comp_out（可以滤波，可以消隐）输出到 TIM/EXTI
28：19	Reserved	-	-	保留
18	PWRMODE	RW	0	COMP4 功耗模式选择 选择了功耗和由此而来的 COMP2 的速度 0：高速 1：中速
17	Reserved	-	-	保留
16	HYST	RW	0	COMP4 迟滞功能使能控制 0：关闭迟滞功能 1：使能迟滞功能
15	POLARITY	RW	0	COMP4 极性选择 0：不反向 1：反向
14:11	BLANKSEL	RW	4’h0	COMP4 消隐信号选择 该位由软件选择比较器 4 的消隐源。 0000：无消隐功能 0001：tim1_oc5 0010：tim2_oc3 0011：tim3_oc3 0100：tim8_oc5 0101：EPWM1 0110：EPWM2 0111：EPWM3 1000：EPWM4 1001：EPWM5 1010：EPWM6 其他：保留
10:8	INPSEL	RW	3’h0	COMP4 正极输入选择 000：PC4 001：OPA3

Puya Confidential

				010：OPA4 011：COMP3_INP2 100：TS 其他：保留
7:5	Reserved	-	-	-
4:3	INNSEL	RW	2’h0	COMP4 反向输入的信号选择 00：PC5 01：DAC_CH1 10：DAC_CH2 11：PA0（COMP123）
2:1	Reserved	-	-	-
0	EN	RW	0	COMP4 使能位 0：关闭 COMP4 1：使能 COMP4

16.5.8 COMP4 滤波寄存器（COMP4_FR）

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTCNT4[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLTEN4
															RW

Bit	Name	RW	Reset Value	Function
31:16	FLTCNT4	RW	16’h0	比较器 4 采样滤波计数器 采样时钟为 APB 或 LSI 或 LSE。滤波计数值可配置。采样次数达到滤波计数值时，结果一致输出。 采样计数周期=FLTCNT[15:0]
15:1	Reserved	-	-	保留
0	FLTEN4	RW	0	比较器 4 数字滤波功能配置 0：禁止数字滤波功能 1：使能数字滤波功能 注：该位必须在 COMP4 的 EN 为 0 时置位

Puya Confidential

17. 运算放大器(OPA)

17.1 OPA 简介

内嵌 4 个运算放大器，每个运算放大器有两个输入和一个输出。三个 I/O 可以连接到外部引脚，从而实现任何类型的外部互连，也可以将输出连接到内部 ADC 和 COMP。

17.2 OPA 主要特性

- 轨对轨输入和输出电压范围
- 低输入偏置电流
- 低输入偏移电压
- 高频增益带宽
- 高速模式可实现更好的压摆率

17.3 OPA 功能描述

17.3.1 模块框图

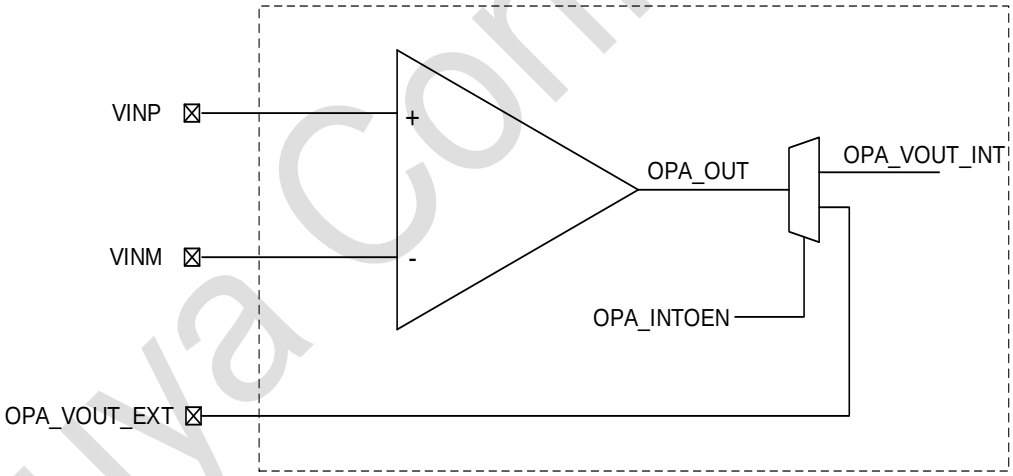


图 17-1 OPA 模块运算放大器模式框图

17.3.2 OPA 输出重定向到内部 ADC 通道

通过设置 OPAx_CSR 寄存器中的 OPAx_INTOEN 位，运算放大器输出可以在内部重定向到 ADC 通道和 COMP 通道。在这种情况下，映射到 OPAx_VOUT 的 GPIO 被释放可以用于其他目的。ADC 可以在内部测量 OPAx_VOUT 电压，COMP 可以在内部比较 OPAx_VOUT 电压。

17.3.3 OPA 复位和时钟

RCC 提供的 OPA 时钟与 PCLK2（APB2 时钟）同步。在 RCC 控制器中提供时钟使能控制位。OPAx_CSR.OPAEN（x=1、2、3、4）位用来启用和禁用 OPA 模块。如果需要更改 OPA 寄存器配置，

Puya Confidential

必须在使能 OPAEN 位之前，以避免对输出产生影响。当不再需要运算放大器的输出时，可以禁用 OPA 以节省功耗。禁用 OPA 时，将保留先前设置的所有配置。

17.3.4 初始配置

运算放大器的默认配置是一种功能模式，其中三个输入/输出连接到外部引脚。

一旦设置了 OPAx_CSR 寄存器中的 OPAEN 位，运算放大器即可工作。两个输入引脚和输出引脚按照下表中的定义进行连接。

注意：输入和输出引脚必须在相应的 GPIOx_MODER 寄存器配置为模拟模式。

17.3.5 信号连接

运算放大器引脚的连接由 OPAx_CSR（x=1、2、3、4）寄存器决定。下表中描述了 4 个运算放大器的连接。

表 17-1 运算放大器可能的连接

Signal	IO pin	Internal	Comment
OPA1_VINN	PC3/PC1		
OPA1_VINP	PC2/PC0		
OPA1_VOUT	PC15	ADC1_CH16/ ADC2_CH16/COMP1_INP/COMP2_INP	由 OPA1_CSR 寄存器中的 OPA1_INTOEN 位控制
OPA2_VINN	PA2/PD4		
OPA2_VINP	PA1/PA3		
OPA2_VOUT	PC14	ADC1_CH17/ADC2_CH17/COMP2_INP	由 OPA2_CSR 寄存器中的 OPA2_INTOEN 位控制
OPA3_VINN	PA6/PA4		
OPA3_VINP	PA5/PD5		
OPA3_VOUT	PB2	ADC1_CH18/ADC2_CH18/COMP3_INP/COMP4_INP	由 OPA3_CSR 寄存器中的 OPA3_INTOEN 位控制
OPA4_VINN	PB1		
OPA4_VINP	PB0		
OPA4_VOUT	PB10	ADC1_CH19/ADC2_CH19/COMP4_INP	由 OPA4_CSR 寄存器中的 OPA4_INTOEN 位控制

17.3.6 OPA 模式

运算放大器的输入和输出都可以通过 GPIO 访问。运算放大器可用于多种配置环境：

- 独立模式(外部增益设置模式)
- 跟随器模式
- 可编程增益放大（PGA）模式

17.3.6.1 独立模式(外部增益设置模式)

OPA 独立模式配置流程：

1) 配置 OPAx_MODE 为 00
2) 配置 OPAx_VBSEL 为 0
3) 配置 OPAx_VPSEL/OPAx_VNSEL 来选择 VINP/VINN 端

Puya Confidential

4) 选择配置 OPAx_EXTOEN 位来使能 OPAx_VOUT_EXT
5) 只要 OPAx_EN 位被配置，电压就输出到 OPAx_VOUT_EXT 端

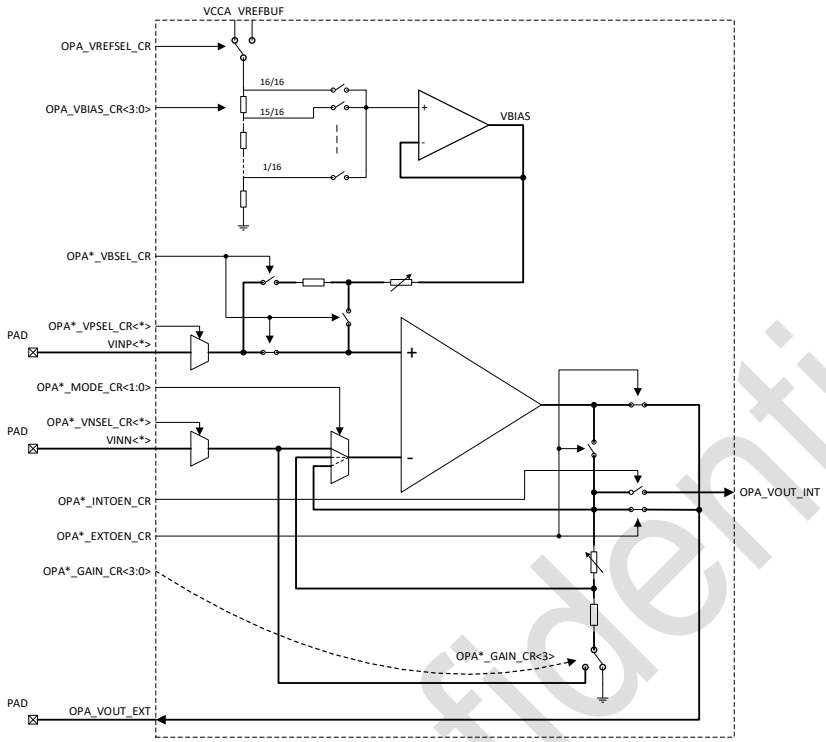


图 17-2 独立模式（外部增益设置模式）

17.3.6.2 跟随器配置模式

OPA 跟随器模式配置流程：

1) 配置 OPAx_MODE 为 10
2) 配置 OPAx_VBSEL 为 0
3) 配置 OPAx_VPSEL 来选择 VINP 端
4) 配置 OPAx_EXTOEN 位来选择是否使能 OPAx_VOUT_EXT
5) 配置 OPAx_INTOEN 位来选择是否使能 OPAx_VOUT_INT
6) 只要 OPAx_EN 位被配置，OPAx_VINP 上的电压被缓冲到引脚 OPAx_VOUT_EXT 或者 OPAx_VOUT_INT 上。

Puya Confidential

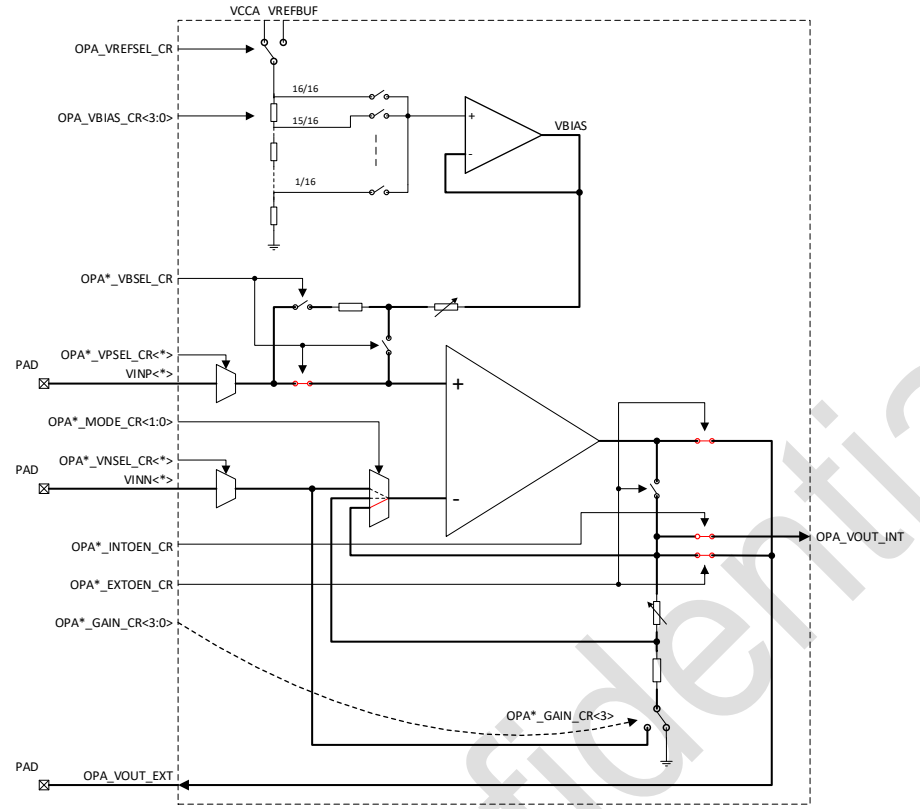


图 17-3 跟随器模式内部输出

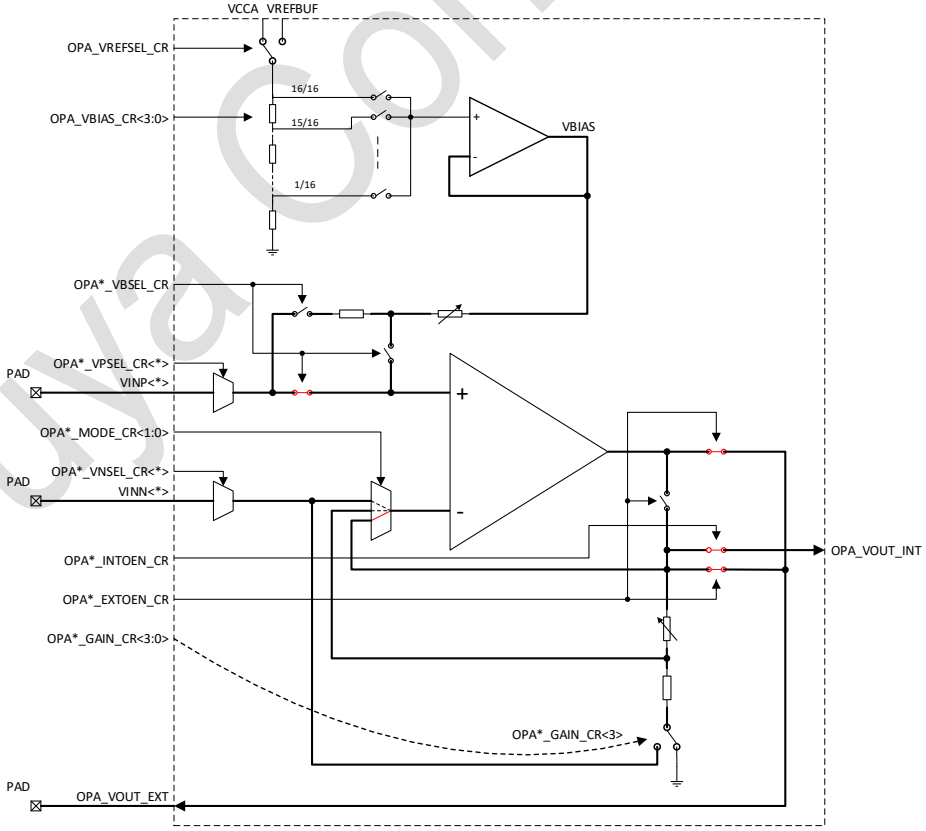


图 17-4 跟随器模式内部和外部输出

Puya Confidential

17.3.6.3 可编程增益放大器模式

PGA 模式有三种配置方式，分别为：

1、单端非反相模式

2、带外部偏置的单端反向/非反相模式

3、带内部偏置的差分模式

OPA 单端非反相模式配置流程：

1) 配置 OPAx_MODE 为 01
2) 配置 OPAx_PGA_GAIN 来选择增益(4'b0000~4'b0110)
3) 配置 OPAx_VBSEL 为 0
4) 配置 OPAx_VPSEL 来选择 VINP 端
5) 配置 OPAx_EXTOEN 位来选择是否使能 OPAx_VOUT_EXT
6) 配置 OPAx_INTOEN 位来选择是否使能 OPAx_VOUT_INT
7) 只要 OPAx_EN 位被配置，OPAx_VINP 引脚上的电压就会根据所选的增益进行放大并在 OPAx_VOUT_EXT 或者 OPAx_VOUT_INT 引脚上可见。

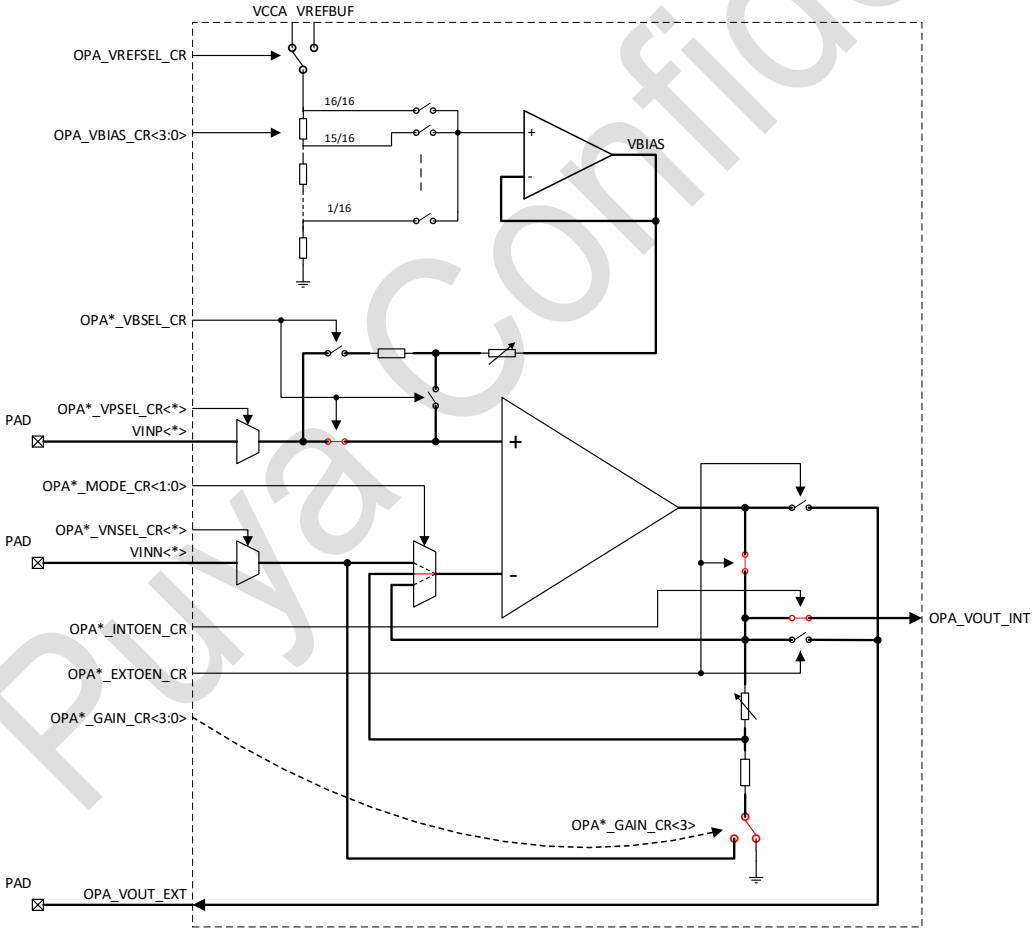


图 17-5 PGA 单端输入，内部输出模式（正向增益）

Puya Confidential

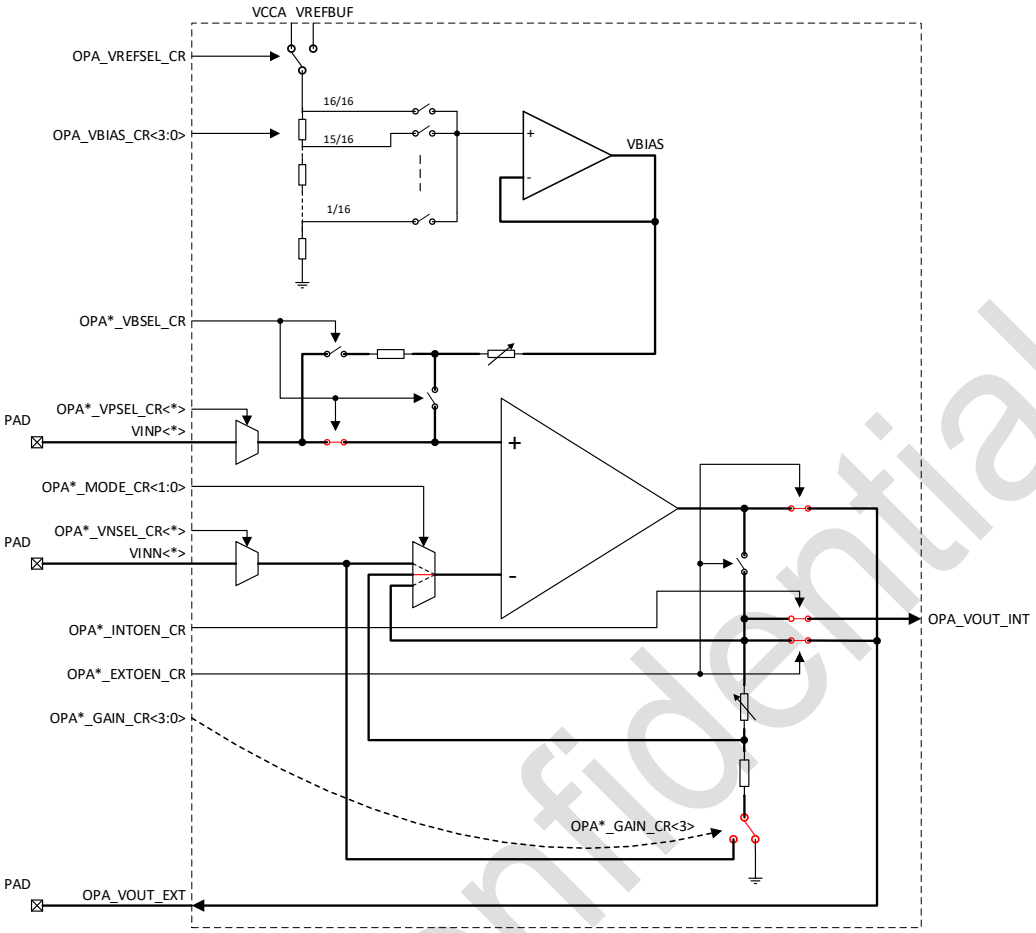


图 17-6 PGA 单端输入，外部输出模式（正向增益）

OPA 带外部偏置的单端反向/非反相模式配置流程：

1) 配置 OPAx_MODE 为 01
2) 配置 OPAx_PGA_GAIN 来设置增益（4’b1000~4’b1110）
3) 配置 OPAx_VBSEL 为 0
4) 配置 OPAx_VPSEL 来选择 VINP 端，配置 OPAx_VNSEL 来选择 VINN 端
5) 配置 OPAx_EXTOEN 位来选择是否使能 OPAx_VOUT_EXT
6) 配置 OPAx_INTOEN 位来选择是否使能 OPAx_VOUT_INT
7) 只要 OPAx_EN 位被设置, OPAx_VINP 引脚上的电压就会根据所选的增益进行放大并在 OPAx_VOUT_EXT 或者 OPAx_VOUT_INT 引脚上可见。

Puya Confidential

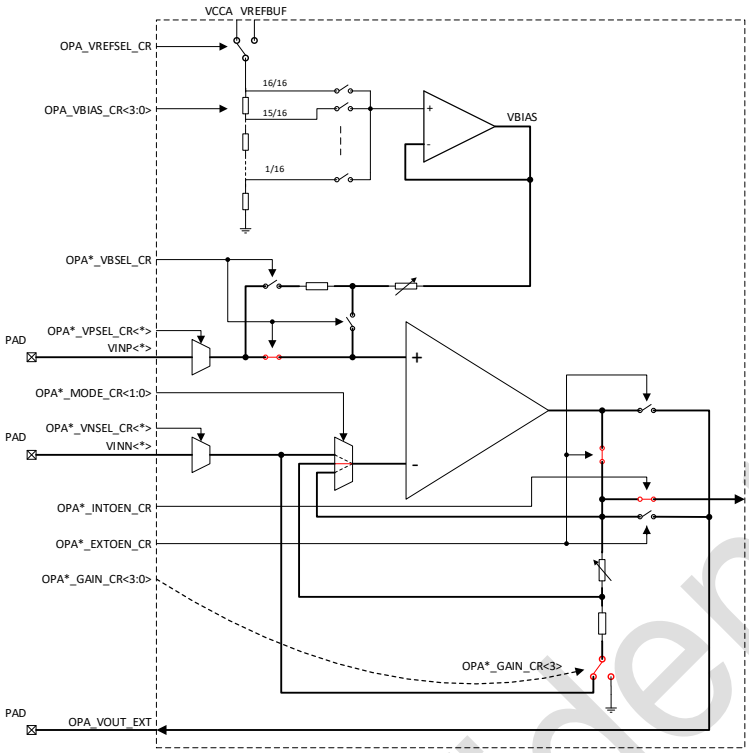


图 17-7 PGA 带外部偏置的单端输入，内部输出，反相增益（1/3/7/11/15/23/31）或者非反相增益(2/4/8/12/16/24/32)

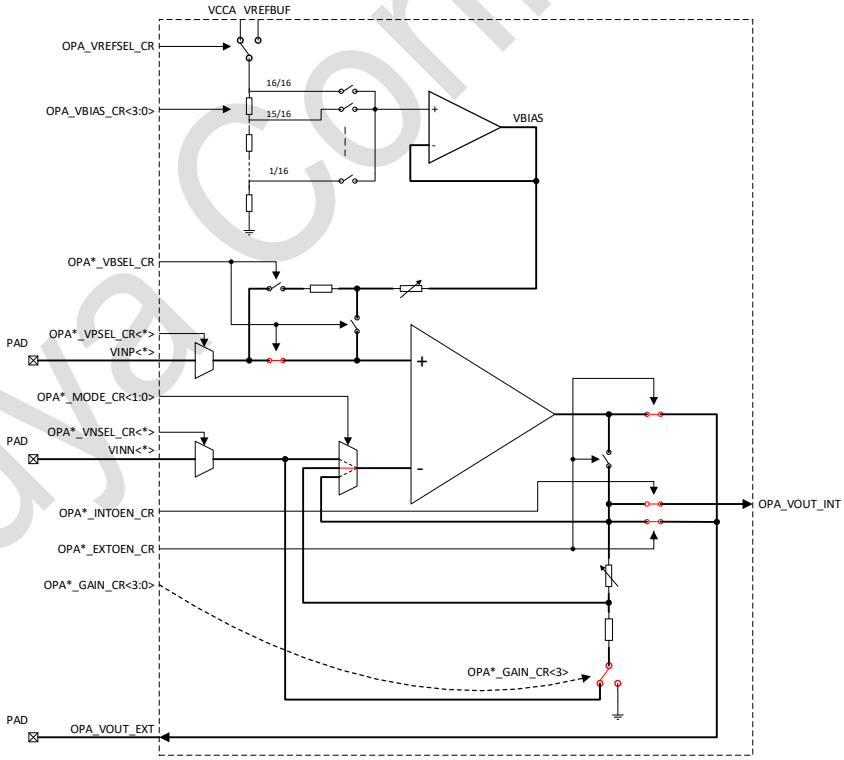


图 17-8 PGA 带外部偏置的单端输入，外部输出，反相增益（1/3/7/11/15/23/31）或者非反相增益(2/4/8/12/16/24/32)

Puya Confidential

带内部偏置的差分模式的配置流程：

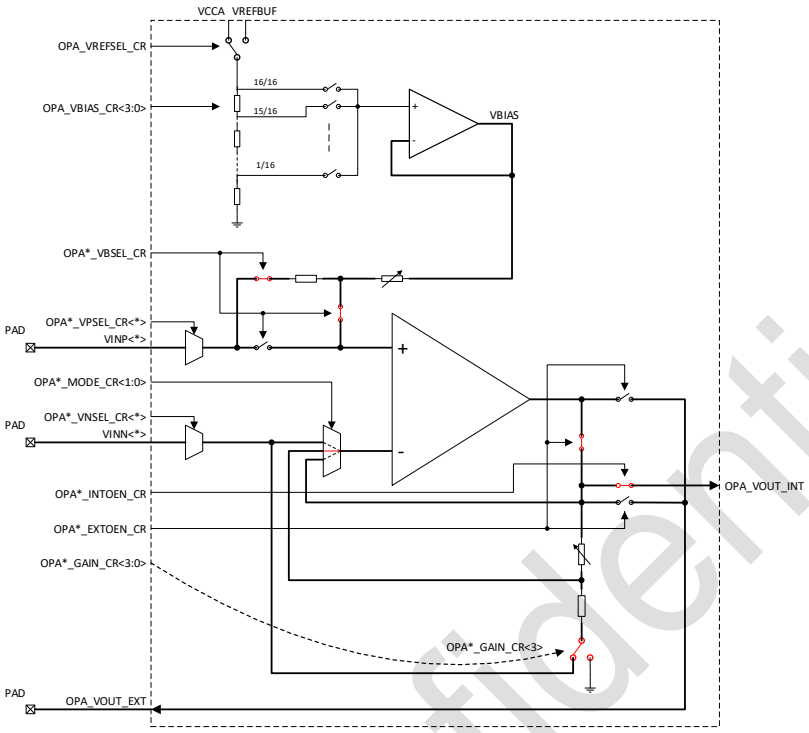


图 17-9 PGA 带内部偏置的差分模式，内部输出，非反相增益（2/4/8/12/16/24/32/）

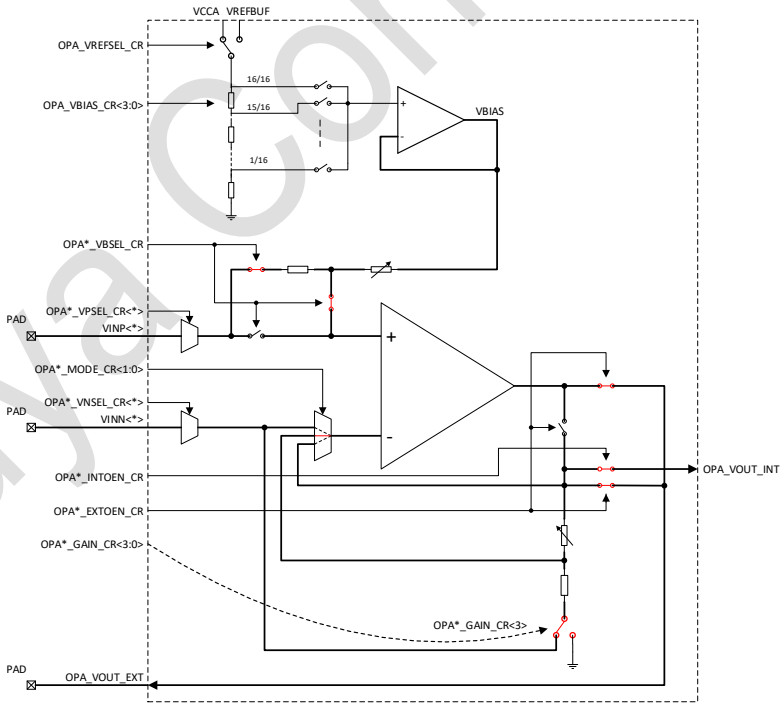


图 17-10 PGA 带内部偏置的差分模式，外部输出，有或没有内部输出，非反相增益（2/4/8/12/16/24/32/）

Puya Confidential

17.3.7 OPA 低功耗模式

表 17-2 低功耗模式对 OPA 的影响

模式	描述
睡眠模式	无影响
低功耗运行模式	无影响
低功耗睡眠模式	无影响
停止模式	OPA 不工作
待机模式	OPA 寄存器掉电，在退出 Standby 模式后必须重新初始化。

17.4 OPA 寄存器

17.4.1 OPA1 控制/状态寄存器（OPA1_CSR）

偏移地址 0x00

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPA1_LOCK	Res.	OPA1_VBSEL	Res.	OPA_VREF_SEL	OPA_VBIAS[3:0]				Res.	Res.	Res.	OPA1_PGA_GAIN[3:0]			
RW		RW		RW	RW	RW	RW	RW				RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPA1_MODE[1:0]		Res.	Res.	Res.	Res.	OPA1_EXTOEN	OPA1_INTOEN	Res.	Res.	OPA1_VNSEL	Res.	Res.	OPA1_VPSEL	Res.	OPA1_EN
RW	RW					RW	RW			RW			RW		RW

Bit	Name	RW	Reset Value	Function
31	OPA1_LOCK	RW	0	该位写一次。它是由软件设置的。它只能通过系统复位或模块软复位来清除。 此位用于将 OPA1_CSR 寄存器配置为只读。 0：OPA1_CSR 可读写 1：OPA1_CSR 只读
30	Reserved	-	-	保留
29	OPA1_VBSEL	RW	0	OPA1 共模电压选择 0：共模电压选择外接电压 1：共模电压选择内置电压
28	Reserved	-	-	保留
27	OPA_VREF_SEL	RW	0	OPA 参考电压选择 0：电压来自 V_{CC} 1：电压来自 V_{REFBUF}
26：23	OPA_VBIAS	RW	4’h0	偏置电压选择 0000：1/16 的 V_{CC}/V_{REFBUF} 0001：2/16 的 V_{CC}/V_{REFBUF} … 1111：16/16 的 V_{CC}/V_{REFBUF}
22:20	Reserved	-	-	保留
19:16	OPA1_PGA_GAIN	RW	4’h0	OPA1 可配置放大器增益值

Puya Confidential

				0000：正向内部增益=2 0001：正向内部增益=4 0010：正向内部增益=8 0011：正向内部增益=12 0100：正向内部增益=16 0101：正向内部增益=24 0110：正向内部增益=32 0111：未使用 1000：反相增益=-1，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=2，VINP 引脚用于输入，VINN 引脚用于偏置 1001：反相增益=-3，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=4，VINP 引脚用于输入，VINN 引脚用于偏置 1010：反相增益=-7，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=8，VINP 引脚用于输入，VINN 引脚用于偏置 1011：反相增益=-11，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=12，VINP 引脚用于输入，VINN 引脚用于偏置 1100：反相增益=-15，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=16，VINP 引脚用于输入，VINN 引脚用于偏置 1101：反相增益=-23，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=24，VINP 引脚用于输入，VINN 引脚用于偏置 1110：反相增益=-31，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=32，VINP 引脚用于输入，VINN 引脚用于偏置 1111：未使用
15:14	OPA1_MODE	RW	2’h0	OPA1 工作模式 00：独立模式 01：PGA 模式 10：跟随器模式 11：保留
13：10	Reserved	-	-	保留
9	OPA1_EXTOEN	RW	0	OPA1 外部输出使能 0：OPA1 外部输出关闭 1：OPA1 外部输出使能
8	OPA1_INTOEN	RW	0	OPA1 内部输出使能 。 0：OPA1 内部输出关闭 1：OPA1 内部输出使能
7:6	Reserved	-	-	保留
5	OPA1_VNSEL	RW	0	反向输入选择 0：OPA1 的 VINN0 引脚连接到 OPA1 VINN 输入 1：OPA1 的 VINN1 引脚连接到 OPA1 VINN 输入
4:3	Reserved	-	-	保留
2	OPA1_VPSEL	RW	0	非反向输入选择

Puya Confidential

				0：OPA1 的 VINP0 引脚连接到 OPA1 VINP 输入 1：OPA1 的 VINP1 引脚连接到 OPA1 VINP 输入
1	Reserved	-	-	保留
0	OPA1_EN	RW	0	OPA1 使能 0：禁止 OPA1 1：使能 OPA1

17.4.2 OPA2 控制/状态寄存器（OPA2_CSR）

偏移地址 0x04

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPA2_LOCK	Res.	OPA2_VBSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPA2_PGA_GAIN[3:0]			
RW		RW										RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPA2_MODE[1:0]		Res.	Res.	Res.	Res.	OPA2_EXTOEN	OPA2_INTOEN	Res.	OPA2_VNSEL[1:0]		Res.	Res.	OPA2_VPSEL	Res.	OPA2_EN
RW	RW					RW	RW		RW	RW			RW		RW

Bit	Name	RW	Reset Value	Function
31	OPA2_LOCK	RW	0	该位写一次。它是由软件设置的。它只能通过系统复位或模块软复位来清除。 此位用于将 OPA2_CSR 寄存器配置为只读。 0：OPA2_CSR 可读写 1：OPA2_CSR 只读
30	Reserved	-	-	保留
29	OPA2_VBSEL	RW	0	OPA2 共模电压选择 0：共模电压选择外接电压 1：共模电压选择内置电压
28:20	Reserved	-	-	保留
19:16	OPA2_PGA_GAIN	RW	4’h0	OPA2 可配置放大器增益值 0000：正向内部增益=2 0001：正向内部增益=4 0010：正向内部增益=8 0011：正向内部增益=12 0100：正向内部增益=16 0101：正向内部增益=24 0110：正向内部增益=32 0111：未使用 1000：反相增益=-1，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=2，VINP 引脚用于输入，VINN 引脚用于偏置 1001：反相增益=-3，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=4，VINP 引脚用于输入，VINN 引脚用于

Puya Confidential

				偏置 1010：反相增益=-7，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=8，VINP 引脚用于输入，VINN 引脚用于偏置 1011：反相增益=-11，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=12，VINP 引脚用于输入，VINN 引脚用于偏置 1100：反相增益=-15，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=16，VINP 引脚用于输入，VINN 引脚用于偏置 1101：反相增益=-23，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=24，VINP 引脚用于输入，VINN 引脚用于偏置 1110：反相增益=-31，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=32，VINP 引脚用于输入，VINN 引脚用于偏置 1111：未使用
15:14	OPA2_MODE	RW	2’h0	OPA2 工作模式 00：独立模式 01：PGA 模式 10：跟随器模式 11：保留
13:10	Reserved	-	-	保留
9	OPA2_EXTOEN	RW	0	OPA2 外部输出使能 0：OPA2 外部输出关闭 1：OPA2 外部输出使能
8	OPA2_INTOEN	RW	0	OPA2 内部输出使能 。 0：OPA2 内部输出关闭 1：OPA2 内部输出使能
7	Reserved	-	-	保留
6：5	OPA2_VNSEL	RW	2’h0	反向输入选择 00：OPA2 的 VINN0 引脚连接到 OPA2 VINN 输入 01：OPA2 的 VINN1 引脚连接到 OPA2 VINN 输入 10：OPA1 的 VINN1 引脚连接到 OPA2_VINN 输入 11：保留
4:3	Reserved	-	-	保留
2	OPA2_VPSEL	RW	0	非反向输入选择 0：OPA2 的 VINP0 引脚连接到 OPA2 VINP 输入 1：OPA2 的 VINP1 引脚连接到 OPA2 VINP 输入
1	Reserved	-	-	保留
0	OPA2_EN	RW	0	OPA2 使能

Puya Confidential

				0：禁止 OPA2 1：使能 OPA2

17.4.3 OPA3 控制/状态寄存器（OPA3_CSR）

偏移地址 0x08

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPA3_LOCK	Res.	OPA3_VBSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPA3_PGA_GAIN[3:0]			
RW		RW										RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPA3_MODE[1:0]		Res.	Res.	Res.	Res.	OPA3_EXTOEN	OPA3_INTOEN	Res.	OPA3_VNSEL[1:0]		Res.	Res.	OPA3_VPSEL	Res.	OPA3_EN
RW	RW					RW	RW			RW			RW		RW

Bit	Name	RW	Reset Value	Function
31	OPA3_LOCK	RW	0	该位写一次。它是由软件设置的。它只能通过系统复位或模块软复位来清除。 此位用于将 OPA3_CSR 寄存器配置为只读。 0：OPA3_CSR 可读写 1：OPA3_CSR 只读
30	Reserved	-	-	保留
29	OPA3_VBSEL	RW	0	OPA3 共模电压选择 0：共模电压选择外接电压 1：共模电压选择内置电压
28:20	Reserved	-	-	保留
19:16	OPA3_PGA_GAIN	RW	4'h0	OPA3 可配置放大器增益值 0000：正向内部增益=2 0001：正向内部增益=4 0010：正向内部增益=8 0011：正向内部增益=12 0100：正向内部增益=16 0101：正向内部增益=24 0110：正向内部增益=32 0111：未使用 1000：反相增益=-1，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=2，VINP 引脚用于输入，VINN 引脚用于偏置 1001：反相增益=-3，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=4，VINP 引脚用于输入，VINN 引脚用于偏置 1010：反相增益=-7，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=8，VINP 引脚用于输入，VINN 引脚用于偏置 1011：反相增益=-11，VINP 引脚用于偏置，VINN 引脚用

Puya Confidential

				于输入/正向增益=12，VINP 引脚用于输入，VINN 引脚用于偏置 1100：反相增益=-15，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=16，VINP 引脚用于输入，VINN 引脚用于偏置 1101：反相增益=-23，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=24，VINP 引脚用于输入，VINN 引脚用于偏置 1110：反相增益=-31，VINP 引脚用于偏置，VINN 引脚用于输入/正向增益=32，VINP 引脚用于输入，VINN 引脚用于偏置 1111：未使用
15:14	OPA3_MODE	RW	2’h0	OPA3 工作模式 00：独立模式 01：PGA 模式 10：跟随器模式 11：保留
13:10	Reserved	-	-	保留
9	OPA3_EXTOEN	RW	0	OPA3 外部输出使能 0：OPA3 外部输出关闭 1：OPA3 外部输出使能
8	OPA3_INTOEN	RW	0	OPA3 内部输出使能 。 0：OPA3 内部输出关闭 1：OPA3 内部输出使能
7	Reserved	-	-	保留
6：5	OPA3_VNSEL	RW	2’h0	反向输入选择 00：OPA3 的 VINN0 引脚连接到 OPA3 VINN 输入 01：OPA3 的 VINN1 引脚连接到 OPA3 VINN 输入 10：OPA1 的 VINN1 引脚连接到 OPA3_VINN 输入 11：保留
4:3	Reserved	-	-	保留
2	OPA3_VPSEL	RW	0	非反向输入选择 0：OPA3 的 VINP0 引脚连接到 OPA3 VINP 输入 1：OPA3 的 VINP1 引脚连接到 OPA3 VINP 输入
1	Reserved	-	-	保留
0	OPA3_EN	RW	0	OPA3 使能 0：禁止 OPA3 1：使能 OPA3

Puya Confidential

17.4.4 OPA4 控制/状态寄存器（OPA4_CSR）

偏移地址 0x0C

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPA4_LOCK	Res.	OPA4_VBSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPA4_PGA_GAIN[3:0]			
RW		RW										RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPA4_MODE[1:0]		Res.	Res.	Res.	Res.	OPA4_EXTOEN	OPA4_INTOEN	Res.	Res.	OPA4_VNSEL	Res.	Res.	Res.	Res.	OPA4_EN
RW	RW					RW	RW			RW					RW

Bit	Name	RW	Reset Value	Function
31	OPA4_LOCK	RW	0	该位写一次。它是由软件设置的。它只能通过系统复位或模块软复位来清除。 此位用于将 OPA4_CSR 寄存器配置为只读。 0：OPA4_CSR 可读写 1：OPA4_CSR 只读
30	Reserved	-	-	保留
29	OPA4_VBSEL	RW	0	OPA4 共模电压选择 0：共模电压选择外接电压 1：共模电压选择内置电压
28:20	Reserved	-	-	保留
19:16	OPA4_PGA_GAIN	RW	4'h0	OPA4 可配置放大器增益值 0000：正向内部增益=2 0001：正向内部增益=4 0010：正向内部增益=8 0011：正向内部增益=12 0100：正向内部增益=16 0101：正向内部增益=24 0110：正向内部增益=32 0111：未使用 1000：反相增益=-1，VINP 引脚用于偏置，VINN 引脚用于输入/ 正向增益=2，VINP 引脚用于输入，VINN 引脚用于偏置 1001：反相增益=-3，VINP 引脚用于偏置，VINN 引脚用于输入/ 正向增益=4，VINP 引脚用于输入，VINN 引脚用于偏置 1010：反相增益=-7，VINP 引脚用于偏置，VINN 引脚用于输入/ 正向增益=8，VINP 引脚用于输入，VINN 引脚用于偏置 1011：反相增益=-11，VINP 引脚用于偏置，VINN 引脚用于输入/ 正向增益=12，VINP 引脚用于输入，VINN 引脚用于偏置

Puya Confidential

				1100：反相增益=-15，VINP 引脚用于偏置，VINN 引脚用于输入/ 正向增益=16，VINP 引脚用于输入，VINN 引脚用于偏置 1101：反相增益=-23，VINP 引脚用于偏置，VINN 引脚用于输入/ 正向增益=24，VINP 引脚用于输入，VINN 引脚用于偏置 1110：反相增益=-31，VINP 引脚用于偏置，VINN 引脚用于输入/ 正向增益=32，VINP 引脚用于输入，VINN 引脚用于偏置 1111：未使用
15:14	OPA4_MODE	RW	2’h0	OPA4 工作模式 00：独立模式 01：PGA 模式 10：跟随器模式 11：保留
13:10	Reserved	-	-	保留
9	OPA4_EXTOEN	RW	0	OPA4 外部输出使能 0：OPA4 外部输出关闭 1：OPA4 外部输出使能
8	OPA4_INTOEN	RW	0	OPA4 内部输出使能 。 0：OPA4 内部输出关闭 1：OPA4 内部输出使能
7:6	Reserved	-	-	保留
5	OPA4_VNSEL	RW	0	反向输入选择 0：OPA4 的 VINN0 引脚连接到 OPA4_VINN 输入 1：OPA3 的 VINN1 引脚连接到 OPA4_VINN 输入
4:1	Reserved	-	-	保留
0	OPA4_EN	RW	0	OPA4 使能 0：禁止 OPA4 1：使能 OPA4

Puya Confidential

18. 增强型脉冲宽度调制器（EPWM）

18.1 EPWM 简介

增强型脉冲宽度调制器（EPWM）可以由多个 EPWMx 子模块组成，EPWMx（x 实例化编号）代表一个完整的 PWM 通道，由两个 PWM 输出组成：EPWMA 和 EPWMB。定时器由一个 16 位计数器组成，可生成各种 PWM 波形；单元间可实现软件同步和硬件同步；支持同步输出以同步片外计数器；各基准值寄存器支持预加载功能；外部事件支持消隐和窗口模式；支持同步 DAC；支持多相单元间交错移相控制。

本章包括每个子模块的概述和信息：

- 时基（TB）子模块
- 计数器比较（CC）子模块
- 动作限定（AQ）子模块
- 死区生成器（DB）子模块
- PWM 斩波器（PC）子模块
- 跳闸区（TZ）子模块
- 事件触发（ET）子模块
- 数字比较（DC）子模块

注意：高分辨率（HR）是一个增强功能，并不是一个独立的子模块。详情参阅高分辨率(HR)功能描述章节。

注意：EPWM 模块的寄存器均带有“EPWMx_”前缀，但为了使图片更简洁，在图片中的寄存器描述忽略前缀。

18.2 EPWM 主要特征

- 具有两个 PWM 输出，分别为 EPWMxA 和 EPWMxB，可以将其配置为单边模式、双边对称模式和双边非对称模式
- 通过软件对 PWM 信号进行异步覆写控制
- 一个专用 16 位时基计数器，可以控制 PWM 的周期和频率
- 通过编程调整 PWM 信号的相位，使其与其他 EPWM 模块的信号产生滞后或超前的效果
- 上升沿和下降沿延迟的死区生成分别独立控制
- 采用高频载波信号进行 PWM 调制，适合用于脉冲变压器的栅极驱动
- 可以逐周期实现硬件锁定相位关系
- 在故障情况下，跳闸区既可分配给逐周期跳闸，也可分配给单次跳闸
- 跳闸状态可以强制 PWM 输出高、低逻辑电平或高阻抗状态
- 编程事件分频可以减小 CPU 响应中断的开销
- 所有事件都是 ADCSOC 和 CPU 中断的触发源

Puya Confidential

18.2.1 结构框图

多个 EPWM 模块的输入/输出信号连接如下图所示。后续章节将对这些信号进行详细描述。EPWM 模块的连接顺序可能与下图所示不同。有关特定设备的同步方案，请参阅时基子模块功能描述中同步章节。

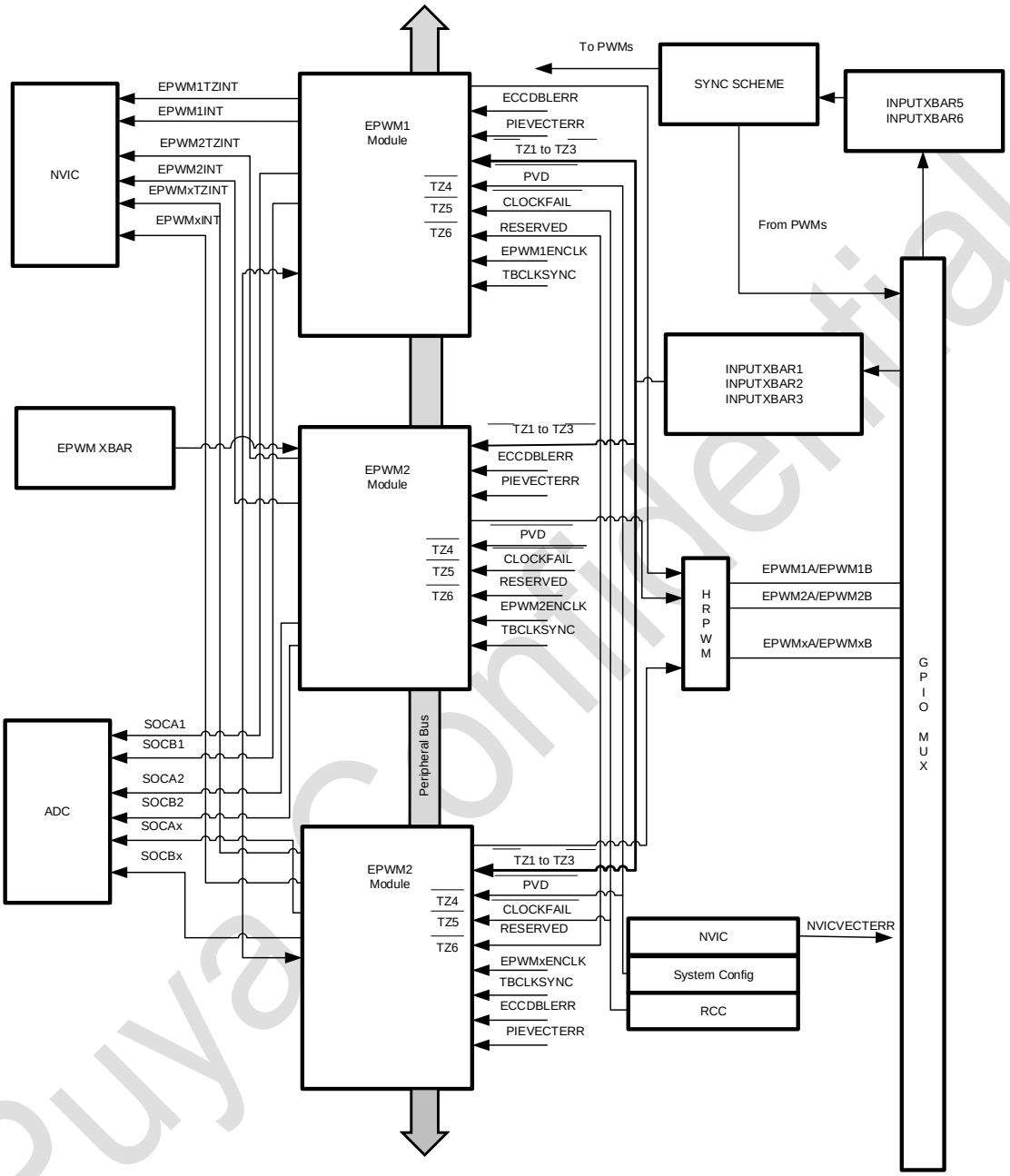


图 18-1 多个 EPWM 模块结构框图

每个 EPWM 模块由 8 个子模块组成，并通过下图所示的信号在系统内连接。

Puya Confidential

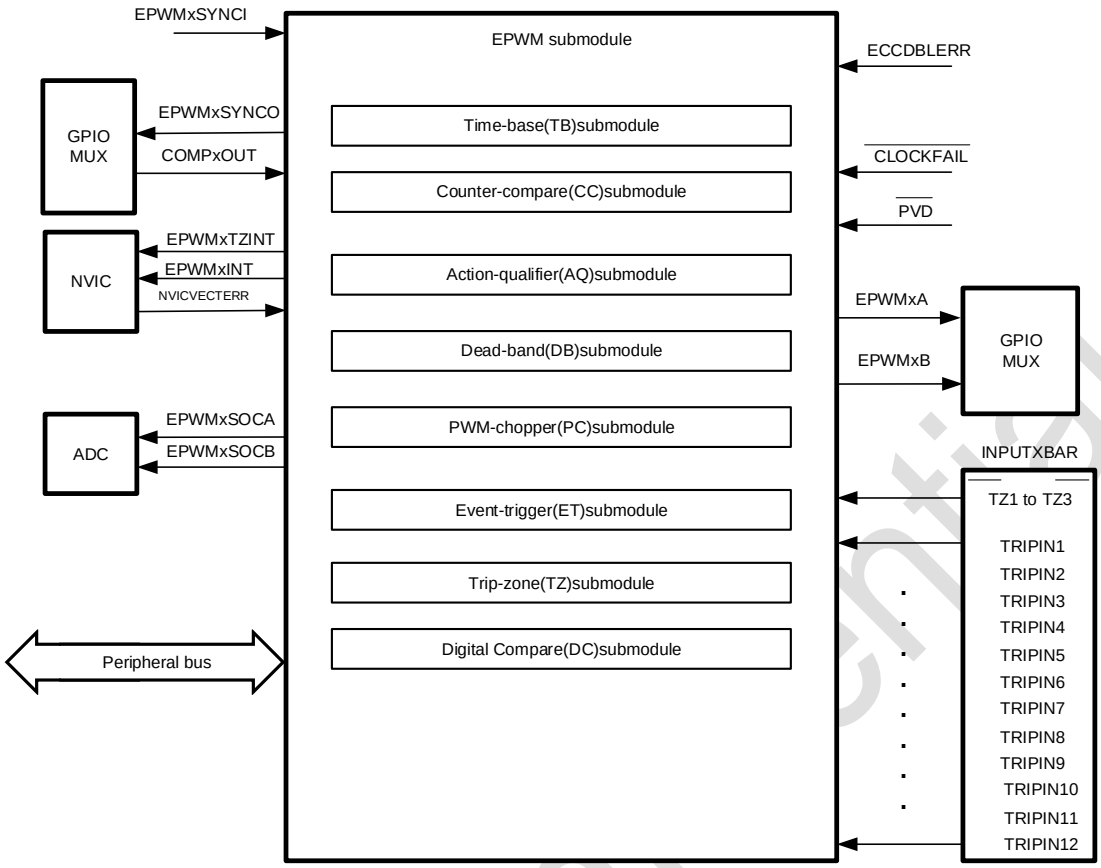


图 18-2 每一个 EPWM 的子模块和信号连接

下图展示了单个 EPWM 模块的更多内部细节。

Puya Confidential

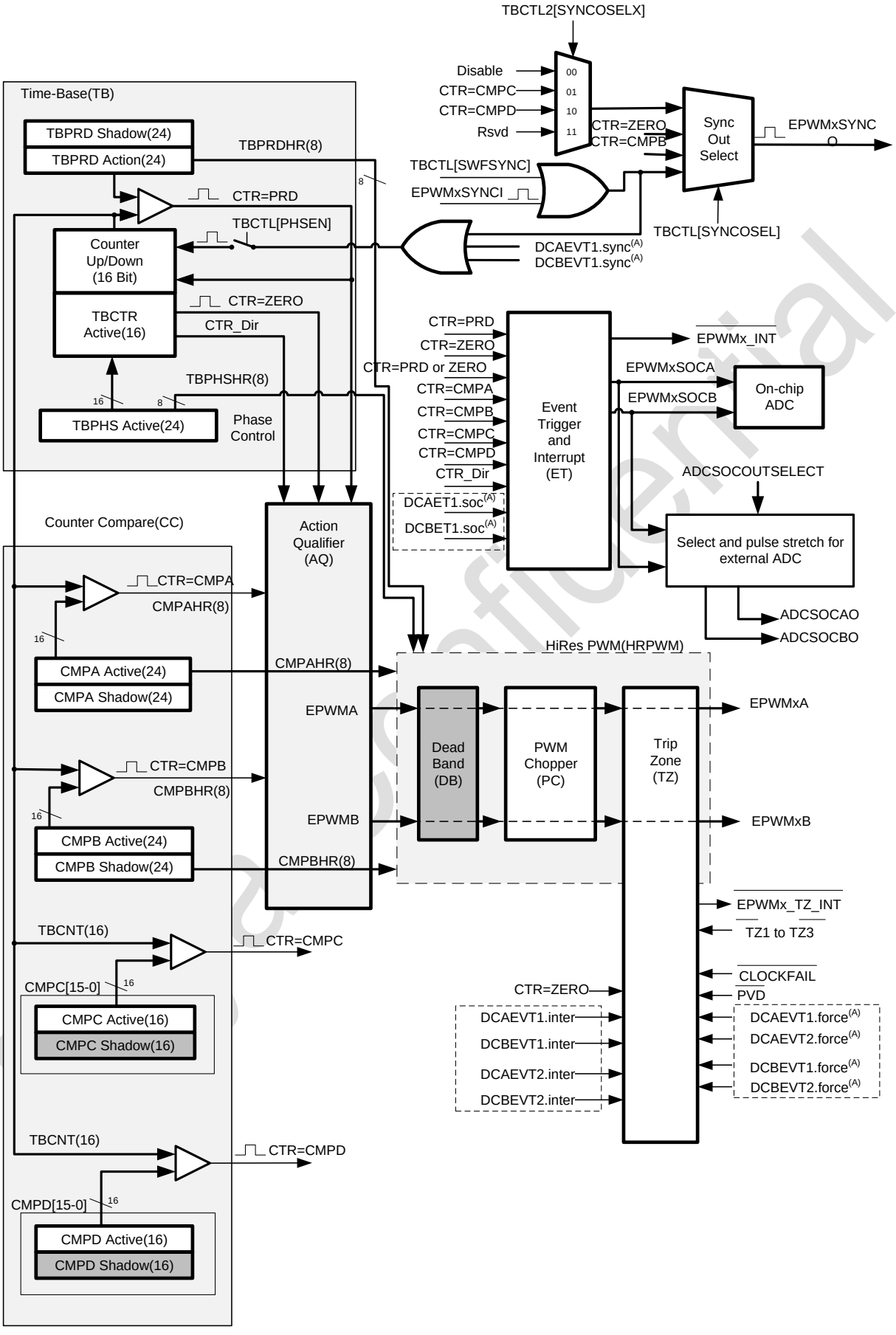


图 18-3 EPWM 模块与关键内部信号互连

Confidential

下表为内部一些关键信号的说明

表 18-1 内部一些关键信号说明

信号类型	信号名称	描述
PWM 输出信号	EPWMxA	输出到器件外的 PWM 信号
	EPWMxB	
跳闸区信号	TZ1n 至 TZ6n	每一个器件上的子模块可以配置为使用或忽略任何跳闸区信号，且这些输入信号通知 EPWM 模块外部的故障情况。这允许配置在时钟故障或 CPU 停止时的跳闸动作。 TZ1n 至 TZ3n：可以通过 GPIO 外设和 INPUT X-BAR 配置为异步输入。 TZ4n：电压检测逻辑（PVD）。 TZ5n： 连接系统时钟失效逻辑。 TZ6n： 连接到 CPU 的 HALT 输出。
时基同步信号	EPWMxSYNCI	同步信号连接各 EPWM 模块。每个模块都可以配置为使用或忽略同步输入。
	EPWMxSYNCO	外部的 SYNC 信号： EXTSYNCIN1 或 EXTSYNCIN2 可被用来发出 SYNC 信号到每一个 EPWM 模块。EPWMxSYNCO 信号：这些模块也可以将 EPWMxSYNCO 信号发出到 GPIO，同时也可以作为其他 EPWM 模块的 SYNC 输入，详情请参见同步章节。EPWMxSYNCSEL 信号：每个 EPWM 模块都可以产生该信号，且该信号用于同步 GPDAC 和 COMP。其功能由 HRPCTL 寄存器配置，但是与 EPWM 没有关系。有关 GPDAC 和 COMP 使用 EPWMxSYNCSEL 的内容，详情请参见对应的章节。
	EP-WMxSYNCSEL	
ADC 转换开始信号	EPWMxSOCA	每个 EPWM 模块都具有 EPWMxSOCA 和 EPWMxSOCB 信号。任何 EPWM 模块都可以触发转换开始，且由 ET 子模块配置哪一个事件触发转换开始。
	EPWMxSOCB	
比较器输出信号	COMPxOUT	COMP 模块的输出信号可以通过输入 X-BAR 传递到一个或所有 12 个跳闸输入（TRIPIN1-TRIPIN12），并且与跳闸区信号搭配，可以产生数字比较事件。
外设总线	-	外设总线宽 32 位，同时支持 EPWM 寄存器的 16 位和 32 位写入。

18.2.2 时基子模块功能特征

- 时基计数器模式：
  - 向上计数模式
  - 向下计数模式
  - 中心对齐计数模式
- 时基时钟：通过对 EPWMCLK 预分频得到
- 通过配置 EPWMx_TBCTR 频率或周期，可以控制事件发生的频率
- 产生 EPWMx_TBCTR=0 或 EPWMx_TBCTR= EPWMx_TBPRD 事件
- 可以配置相对于其它 EPWM 模块的时基相位
- 通过软件或硬件实现 EPWM 模块之间的 EPWMx_TBCTR 同步
- 在同步事件后，配置 EPWMx_TBCTR 方向为向上或向下
- 可以将相同的值同时写入多个 EPWM 模块 EPWMx_TBPRD 寄存器
- 在仿真器暂停器件时，可以配置 EPWMx_TBCTR 的行为
- EPWM 模块同步输出信号源：
  - 同步输入信号
  - EPWMx_TBCTR=0
  - EPWMx_TBCTR=CMPB

Puya Confidential

- 不产生同步输出信号

- 配置 EPWM 模块中寄存器的单次和全局装载

18.2.3 计数器比较子模块功能特征

- 可以指定 EPWMxA 和 EPWMxB 输出切换电平的时刻
- 可以指定 EPWMxA 和 EPWMxB 输出的占空比，通过配置 CMPA 和 CMPB 寄存器来控制占空比
- 可以将相同的值同时写入多个 EPWM 模块 EPWMx_CMPx 寄存器
- 配置 EPWM 模块中寄存器的单次和全局装载
- 通过额外的比较器（CMPC/D），指定可编程延迟的中断和 SOC 生成
- 影子缓冲新的比较值，可以避免在 PWM 有效周期内产生毛刺或波形失真
- 使用 CMPx 寄存器，基于可编程时间戳生成事件：
  - EPWMx_TBCTR= CMPA
  - EPWMx_TBCTR= CMPB
  - EPWMx_TBCTR= CMPC
  - EPWMx_TBCTR= CMPD

18.2.4 动作限定子模块功能特征

- 可以通过软件强制控制 PWM 输出状态
- 可以通过软件配置和控制 PWM 死区
- 当发生时基计数器比较，跳闸区子模块或比较器事件时，可以执行以下动作：
  - EPWMxA 和 EPWMxB 切换为高电平
  - EPWMxA 和 EPWMxB 切换为低电平
  - EPWMxA 和 EPWMxB 电平翻转
  - 无任何动作
- 在以下事件中，进行定性和生成动作（置位、清除、翻转）：
  - EPWMx_TBCTR=0
  - EPWMx_TBCTR= EPWMx_TBPRD
  - EPWMx_TBCTR= CMPA
  - EPWMx_TBCTR= CMPB
- T1、T2 事件：基于比较器、跳闸或同步事件的触发事件，当以上事件同时发生时管理优先级
- 提供在 EPWMx_TBCTR 递增和递减时独立控制事件的能力
- 配置 EPWM 模块中寄存器的单次和全局装载

18.2.5 死区发生器子模块功能特征

- 从单个 EPWMxA 输入生成相应的具有死区关系的信号对，即 EPWMxA 和 EPWMxB
- 可以使能半周期时钟模式，从而加倍分辨率
- 允许 EPWMxB 输出相对于 EPWMxA 相移
- 将信号对编程为：
  - 高电平有效（AH）
  - 低电平有效（AL）
  - 高电平有效互补（AHC）

Puya Confidential

- 低电平有效互补（ALC）
- 插入可编程的上升沿延迟(RED) 和下降沿延迟 (FED) 的值
- 可以控制上下开关之间，传统的互补死区关系
- 可以完全绕过死区发生器子模块；
  - PWM 波形可以没有任何修改地通过死区发生器子模块
- 配置 EPWM 模块中寄存器的单次和全局装载

18.2.6 PWM 斩波子模块功能特征

- 可编程的斩波频率
- 首个脉冲可编程脉宽，第二个和后续的脉冲可编程占空比
- 可以完全绕过 PWM 斩波子模块；
  - PWM 波形可以没有任何修改地通过 PWM 斩波子模块

18.2.7 跳闸区子模块功能特征

- 跳闸输入 TZ1n 至 TZ6n 可被灵活地映射到任意 EPWM 模块
- 支持软件强制跳闸
- 配置 EPWM 模块对跳闸信号或数字比较事件做出反应：
  - 一个跳闸信号或数字比较事件
  - 所有跳闸信号或数字比较事件
  - 不做出反应
- 在故障状态下，可以配置跳闸动作为：
  - EPWMxA 和/或 EPWMxB 强制输出为高阻态
  - EPWMxA 和/或 EPWMxB 强制输出为高电平
  - EPWMxA 和/或 EPWMxB 强制输出为低电平
  - 忽略任何跳闸动作
- DC 子模块 DCAEVT1/2 或 DCBEVT1/2 强制事件和每个跳闸区输入都可以被分配给单次或逐周期操作
- 单次跳闸（OST Trip）：支持用于短路和过流保护。
- 逐周期跳闸（CBC Trip）：支持用于限流操作。
- 使能跳闸区发起中断，任何一个跳闸区输入都可以产生中断
- 支持基于片上模拟比较器和/或 TZ1n 至 TZ3n 的数字比较跳闸（DC）
- 可以完全绕过跳闸区子模块

18.2.8 事件触发子模块功能特征

- 使能触发中断或 ADCSOC 的 PWM 事件
- 允许软件强制触发中断和 ADCSOC
- 使用预分频电路指定事件触发的频率（每 2 次至最多每 15 次）
- 事件计数器和标志位提供了事件生成的完整可见性
- 接收由 TB、CC 和 DC 子模块的事件输入。
- 使用时基方向信息来区分（上/下）事件
- 轮询、设置、或清除事件标志位

Puya Confidential

18.2.9 数字比较子模块功能特征

- 利用比较器（COMP）模块输出和跳闸区信号来产生事件，这些信号通过输入 EPWMXBAR 传入。
- 通过以下信号产生 DCxH 和 DCxL 信号：
  - 内部中断控制
  - TZ1n、TZ2n、TZ3n 输入
  - 使用 Input X-BAR、EPWM X-BAR 或者经过外部使用 GPIO 外设输入的 COMP 模块输出
- 可以指定事件滤波选项来捕获 EPWMx_TBCTR 并产生空白窗口或基于捕获值在 PWM 输出或 EPWMx_TBCTR 插入延迟
- 事件滤波（空白窗口逻辑电路）可用来过滤输入信号，以移除噪声
- DCxH 和 DCxL 信号可以触发事件，这些事件可以选择进行过滤或直接传递给 TZ、ET 和 TB 子模块，从而：
  - 生成用于同步 EPWM 模块 EPWMx_TBCTR 的同步事件
  - 强制产生一个事件
  - 生成 ADCSOC
  - 生成跳闸区中断

18.2.10 高分辨率功能特征

（1）可以用于控制 PWM 信号的占空比和相位

（2）具有扩展时钟分辨率能力

（3）死区高分辨率控制

* 允许在半周期时钟运行时，对 RED 和 FED 进行控制

（4）允许对 PWM 输出进行使能高分辨率切换

（5）通过翻转 EPWMxA 信号输出可以使能对 EPWMxB 信号输出的高分辨率控制

（6）应用在 EPWMxA 和 EPWMxB 输出（PWM 的 A 和 B 信号路径）上

（7）自检诊断软件模式

* 该模块用于检查 MEP 逻辑的运行状况，即检查是否按设计运行,关于 MEP 的详细描述请查看高分辨率(HR)功能描述章节。

（8）通过扩展 EPWMx_ CMPA、EPWMx_CMPB 和 EPWMx_TBPHS 寄存器对 PWM 信号进行细致的边沿定位或时间颗粒度控制

（9）可以对 PWM 输出信号的周期、占空比和相位进行高分辨率控制

18.3 EPWM 功能描述

EPWM 可以由多个 EPWMx 子模块组成，EPWMx（x 实例化编号）代表一个完整的 PWM 通道，由两个 PWM 输出组成：EPWMA 和 EPWMB。每个 EPWM 模块由一个以 1 开始的数值表示。例如，EPWM1 是第一个实例，EPWM3 是系统中的第三个实例，而 EPWMx 表示任何实例。EPWM 模块通过时钟同步方案链接在一起，允许它们在需要时作为单个系统运行。子模块的数量取决于器件并基于目标应用的需求。子模块也可以独立运行。

Puya Confidential

18.3.1 EPWM 引脚输入描述

使用 INPUT X-BAR 和 EPWM XBAR 才能将器件引脚上的信号接入本模块，本器件上的大部分 GPIO 均可配置为模块的输入。此外，TRIP4 至 TRIP12（除了 TRIP6）信号在经过 INPUT X-BAR 之后，还必须经过 EPWM X-BAR 才能到达 EPWM 模块。

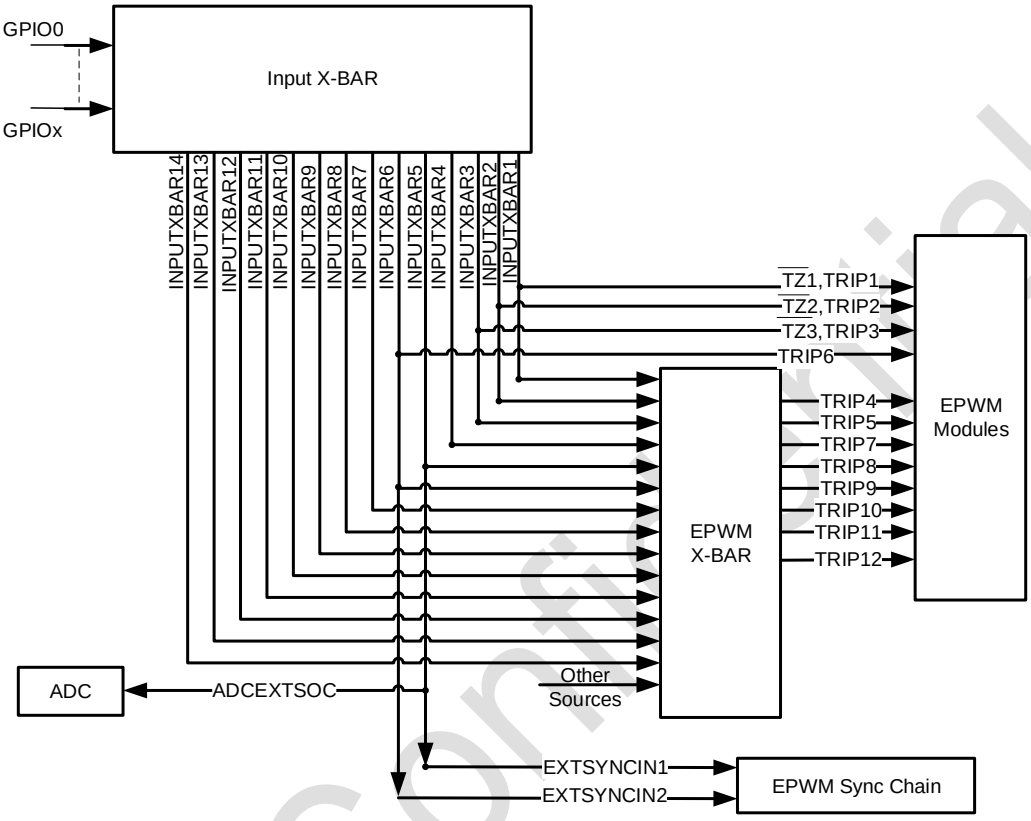


图 18-4 EPWMx 子模块输入连接关系

18.3.1.1 INPUT X-BAR

Input X-BAR 由 14 个相互独立的功能模块构成，每一个功能模块内部结构如下图所示。

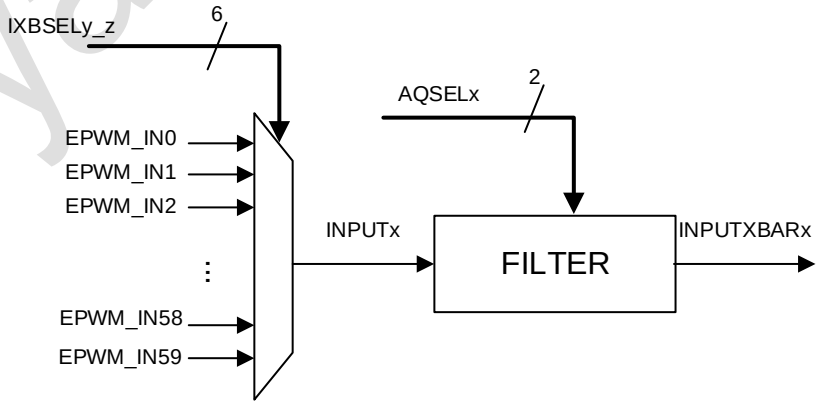


图 18-5 INPUT X-BAR 结构框图

选择器的来源是 GPIO，通过 EPWMx_IXBSELy_z 寄存器可以配置选择器输出。14 路输出 INPUTx 信号可以独立选择从 EPWMx_IN0 到 EPWMx_IN59 中的任意一个，如下表所示。

Puya Confidential

表 18-2 INPUT X-BAR 的 GPIO 选择源

SEL(6bits)	Signal
0	EPWM_IN0
1	EPWM_IN1
2	EPWM_IN2
3	EPWM_IN3
4	EPWM_IN4
5	EPWM_IN5
6	EPWM_IN6
7	EPWM_IN7
8	EPWM_IN8
9	EPWM_IN9
10	EPWM_IN10
11	EPWM_IN11
12	EPWM_IN12
13	EPWM_IN13
14	EPWM_IN14
15	EPWM_IN15
16	EPWM_IN16
17	EPWM_IN17
18	EPWM_IN18
19	EPWM_IN19
20	EPWM_IN20
21	EPWM_IN21
22	EPWM_IN22
23	EPWM_IN23
24	EPWM_IN24
25	EPWM_IN25
26	EPWM_IN26
27	EPWM_IN27
28	EPWM_IN28
29	EPWM_IN29
30	EPWM_IN30
31	EPWM_IN31
32	EPWM_IN32
33	EPWM_IN33
34	EPWM_IN34
35	EPWM_IN35
36	EPWM_IN36
37	EPWM_IN37
38	EPWM_IN38
39	EPWM_IN39
40	EPWM_IN40
41	EPWM_IN41
42	EPWM_IN42
43	EPWM_IN43
44	EPWM_IN44
45	EPWM_IN45
46	EPWM_IN46

Puya Confidential

47	EPWM_IN47
48	EPWM_IN48
49	EPWM_IN49
50	EPWM_IN50
51	EPWM_IN51
52	EPWM_IN52
53	EPWM_IN53
54	EPWM_IN54
55	EPWM_IN55
56	EPWM_IN56
57	EPWM_IN57
58	EPWM_IN58
59	EPWM_IN59
60	Reserved
61	Reserved
62	Reserved
63	Reserved

INPUTx 信号经过 FILTER 模块进行输入类型处理。输入类型方案的设计非常灵活。通过配置 EPWM_AQSELx 寄存器，为每个 INPUTx 选择类型。可以将类型指定为异步信号、仅与 PCLK 同步或者是通过采样窗口进行信号判定。

■ 异步输入

此模式满足于一些特殊异步需求，如 EPWM 的 trip zone 等。

■ 同步输入

同步输入为默认配置，仅将异步输入的信号，经过 PCLK 时钟的寄存器做 2 拍同步后的信号作为输出。

■ 采样窗口

此模式，信号先经过 PCLK 的 2 拍同步后再做窗口采样处理。采样窗口可以通过 EPWM_AQSELx 设置为 3 次采样判定或者 6 次采样判定，同时可以通过 EPWM_ISAPRD0/1 寄存器选择 INPUT0~7 和 INPUT8~14 的输入采样时钟分频。当连续 3 或 6 个周期数值一致，输入的改变电平就会被输出出去。窗口内的采样周期数总是比采集的样本数少一个。对于三采样窗口，采样窗口宽度为两个采样周期宽。同样，对于六采样窗口，采样窗口宽度为五个采样周期宽。

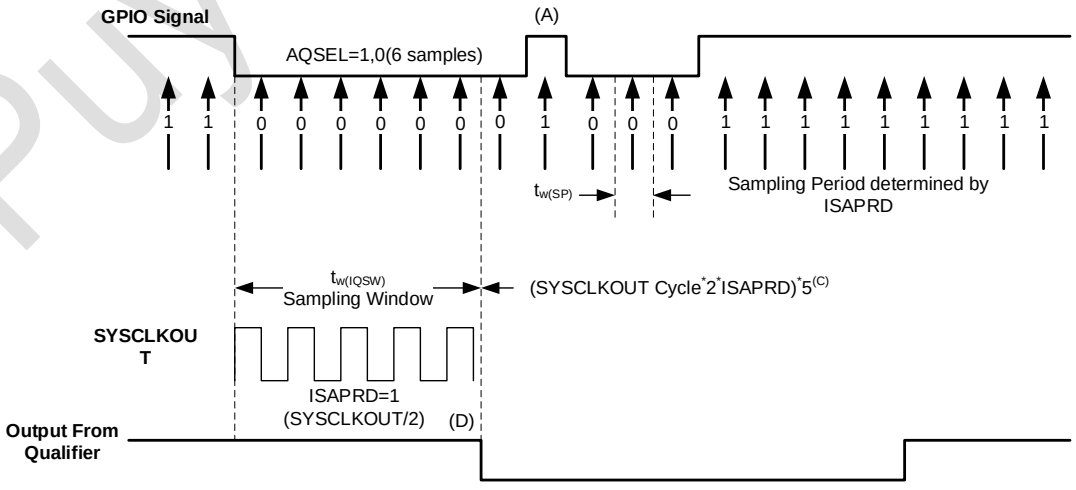


图 18-6 INPUTXBAR 窗口采样示意图

Puya Confidential

由于只需要任何一个可用的 GPIO 即可实现灵活的信号路由，因此使用 Input X-BAR 可以减轻外设复用的一些限制。

18.3.1.2 EPWM X-BAR

EPWM X-BAR 由 8 个相互独立的功能模块构成，每一个功能模块内部结构如下图所示。

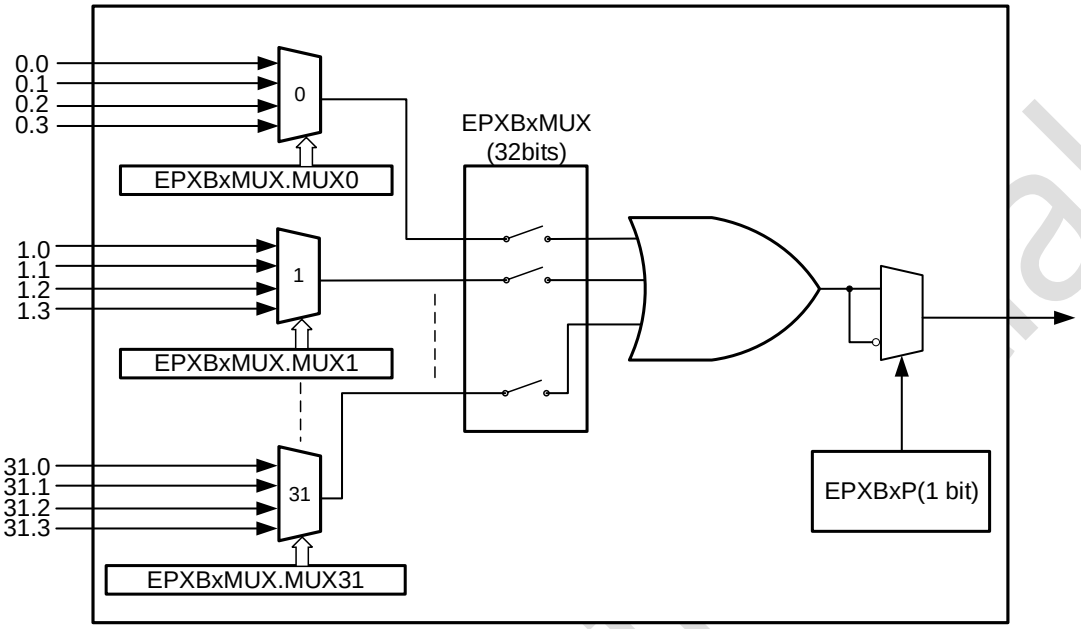


图 18-7 EPWM X-BAR 结构框图

EPWMXBAR 可以选择各种触发源，使其成为 8 个专用的 EPWM 跳闸输入中的任意一个，即 TRIP4/5/7/8/9/10/11/12。每个输出 TRIPx，可以在 MUX 上选择单个信号，或者使用逻辑或连接，最多 32 个信号，具体参考表 18-3 EPWM X-BAR 配置表。TRIPx 与 EPWM 模块的数字比较（DC）子模块连接，执行跳闸和同步等操作。

信号传输步骤：

1. 选择需要传输到 EPWM 的信号：对于每个 TRIPx，在 EPWM_EPXBx_0 或 EPWM_EPXBx_1 寄存器中选择一个信号作为每个 MUX 的输入（总共有 32 个 MUX）。
2. 使能多路复用器：在 EPWM_EPXBx_MUX 寄存器中使能多路复用器。在传输到 EPWM 模块的 TRIPx 信号前，所有的多路复用器进行逻辑或运算。
3. （可选）反相信号：在 EPWM_EPXB_POL 寄存器中对信号进行反相。

表 18-3 EPWM X-BAR 配置表

MUX	0	1	2	3
0	comp1_out	comp1_out or comp2_out	adc1_awd1	tim1_ch1
1	comp2_out	INPUTXBAR1	Reserved	adc1_awd1
2	comp3_out	comp3_out or comp4_out	adc1_awd2	tim2_ch1
3	comp4_out	INPUTXBAR2	Reserved	adc1_awd2
4	comp1_out	comp1_out or comp3_out	adc1_awd3	tim3_ch1
5	comp3_out	INPUTXBAR3	Reserved	adc1_awd3
6	comp2_out	comp2_out or comp4_out	adc2_awd1	tim8_ch1
7	comp4_out	INPUTXBAR4	Reserved	adc2_awd1
8	comp1_out	comp1_out or comp4_out	adc2_awd2	tim1_ch2

Puya Confidential

9	comp4_out	INPUTXBAR5	Reserved	adc2_awd2
10	comp2_out	comp2_out or comp3_out	adc2_awd3	tim2_ch2
11	comp3_out	INPUTXBAR6	Reserved	adc2_awd3
12	Reserved	Reserved	Reserved	tim1_ch3
13	Reserved	ADCSOCAO	Reserved	tim1_ch4
14	Reserved	Reserved	Reserved	EPWMx_SCOUT
15	Reserved	ADCSOCBO	Reserved	tim1_ch5
16	Reserved	Reserved	Reserved	tim1_ch6
17	Reserved	INPUTXBAR7	Reserved	CPUHALT
18	Reserved	Reserved	Reserved	tim2_ch3
19	Reserved	INPUTXBAR8	Reserved	tim2_ch4
20	Reserved	Reserved	Reserved	tim8_ch2
21	Reserved	INPUTXBAR9	Reserved	tim8_ch3
22	Reserved	Reserved	Reserved	tim8_ch4
23	Reserved	INPUTXBAR10	Reserved	tim8_ch5
24	Reserved	Reserved	Reserved	tim8_ch6
25	Reserved	INPUTXBAR11	Reserved	tim3_ch2
26	Reserved	Reserved	Reserved	tim3_ch3
27	Reserved	INPUTXBAR12	Reserved	tim3_ch4
28	Reserved	Reserved	Reserved	Reserved
29	Reserved	INPUTXBAR13	Reserved	Reserved
30	Reserved	Reserved	Reserved	Reserved
31	Reserved	INPUTXBAR14	Reserved	Reserved

注意：“保留”的信号不可使用，所有的未使用和保留的信号都设定为 0。

注释：compx_out 为芯片内部比较器的输出信号；compx_out or compy_out 代表两个比较器的输出做或逻辑；adcx_awdy 为芯片内部 ADC 模块的看门狗信号；timx_chy 为芯片内部 timer 的通道输出。

18.3.1.3 X-BAR 标志位

所有 EPWMX-BAR 模块的输入信号都有一个标志，用来指示哪些输入信号已被触发。这使得软件能够在事件发生时检查输入标志。对应信号的位置可查看寄存器 EPWM_XBARFLG1 和 EPWM_XBARFLG2。除了 ADCSOCA_FLG 和 ADCSOCB_FLG，其余的 flag 均为对应信号的上升沿触发。ADCSOCA_FLG 和 ADCSOCB_FLG 的触发沿根据寄存器 SOCA_POL 和 SOCB_POL 而自动调整，当选择高电平有效时为上升沿触发，低电平有效时为下降沿触发。触发后所有标志位将保持置位状态，直至通过相应的 EPWM_XBARCLRx 寄存器清除。

18.3.2 时基（TB）子模块描述

时基子模块用于确定该 EPWM 模块的所有事件时序，所以每个 EPWM 模块都有自己的时基子模块。内部的同步逻辑允许多个 EPWM 模块的时基协同工作，构成一个系统。

Puya Confidential

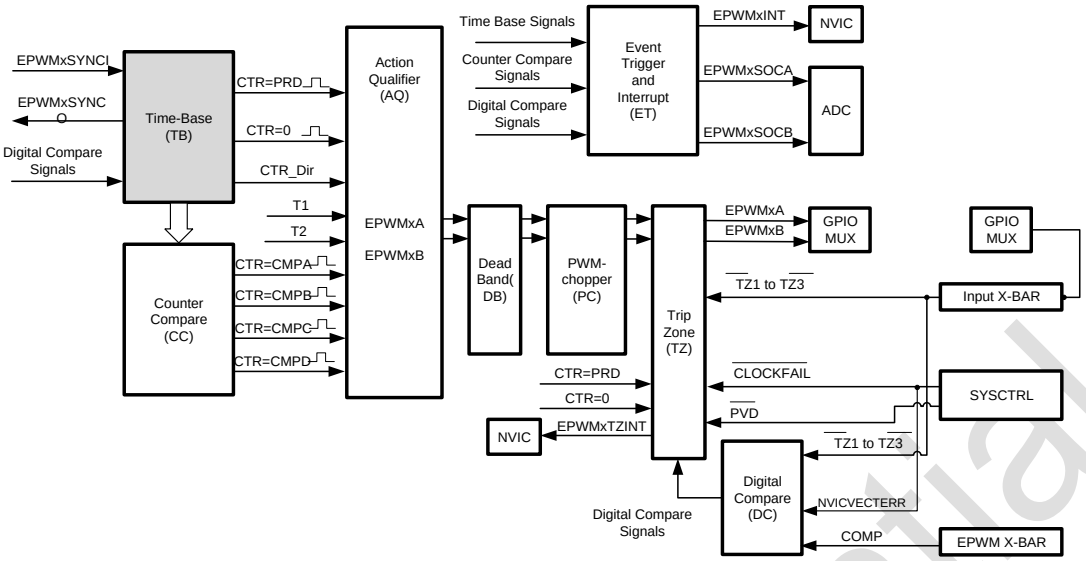


图 18-8 EPWMx 中的时基子模块

18.3.2.1 结构框图

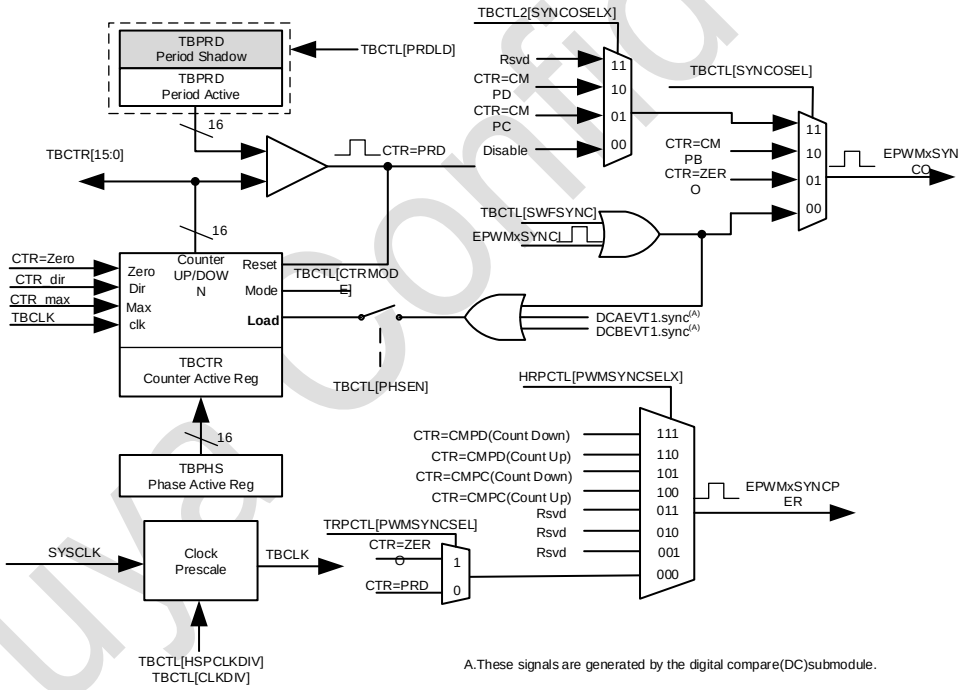


图 18-9 时基子模块结构框图

表 18-4 时基于模块关键信号

信号类型	信号名称	描述
时基时钟	TBCLK	通过对 EPWMCLK 预分频得到，可以用于所有 EPWM 子模块，且该时钟决定了 EPWMx_TBCTR 递增或递减的速率。
时基计数器方向	TBCTR_DIR	表示 EPWM 时基计数器当前的方向。该信号在计数器递增时为高电平，递减时为低电平。
时基计数器等于最大值	TBCTR_MAX	当 EPWMx_TBCTR 达到最大值，即 EPWMx_TBCTR=0xFFFF 时，产生该事件。该信号仅用作状态位。

Puya Confidential

时基计数器等于给定周期	TBCTR=TBPRD	计数器值等于活动周期寄存器值，即 EPWMx_TBCTR= EPWMx_TBPRD 时，产生该信号。
时基计数器等于 0	TBCTR=0	计数器值等于零，即 EPWMx_TBCTR=0 时，产生该信号。
时基计数器等于活动计数器比较 B 寄存器	TBCTR=CMPB	该事件由数值比较子模块产生，并用于同步输出逻辑。
时基同步信号	EPWMxSYNCI 时基同步输入	该信号为脉冲输入，用于将 EPWMx_TBCTR 与另一个或者多个 EPWMx_TBCTR 同步。每个 EPWMx 模块可以灵活选择自己的同步源，使得 EPWMx 模块之间可以任意组合成同步的小系统。当同步源选择 reserved 选项关闭同步输入。关于器件的同步详情，请参见同步章节。
	EPWMxSYNCO 时基同步输出	该输出脉冲用于同步同步链中其他 EPWM 模块的 EPWMx_TBCTR， EPWM 模块通过以下事件源产生该信号： （1） EPWMx_TBCTR=0 （2） EPWMx_TBCTR= CMPB （3） EPWMxSYNCI 注意：EPWM3SYNCO 和 EPWM6SYNCO 保留
	EPWMxSYNCPER 时基外设同步输出	该信号用于将 COMP 与 PWM 同步。

18.3.2.2 PWM 周期和频率计算

通过 EPWMx_TBCTR 寄存器和计数器模式控制 PWM 事件的频率，其计数器模式通过 EPWMx_TBCTL 寄存器来选择。如下图所示，当 EPWMx_TBPRD 设置为 4 时，在三种工作模式下的周期 T_{PWM} 和频率 F_{PWM} 关系。由 TBCLK 决定每一步的时间增量。

（1）向上计数模式

在该模式下，EPWMx_TBCTR 从 0 开始递增，直到达到 EPWMx_TBPRD 寄存器中的值。当达到 EPWMx_TBPRD 值时，时基计数器重置为 0 并再次开始增加。

在该模式下，周期 T_{PWM} 和频率 F_{PWM} 计算公式为：

$$T_{PWM} = (TBPRD+1) * T_{TBCLK}$$

$$FPWM = \frac{1}{TPWM}$$

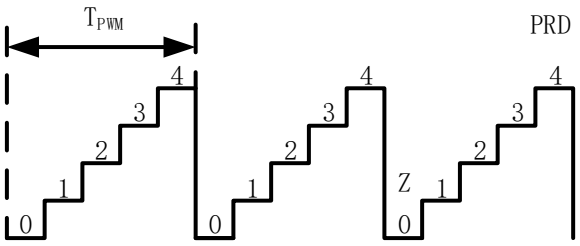


图 18-10 向上计数模式的时基频率和周期

Puya Confidential

（2）向下计数模式

在该模式下，EPWMx_TBCTR 从 EPWMx_TBPRD 值开始递减，直到达到 0。当它达到 0 时，EPWMx_TBCTR 重置为 EPWMx_TBPRD 值，并再次开始递减。

在该模式下，周期 T_{PWM} 和频率 F_{PWM} 计算公式详情请参见向上计数模式。

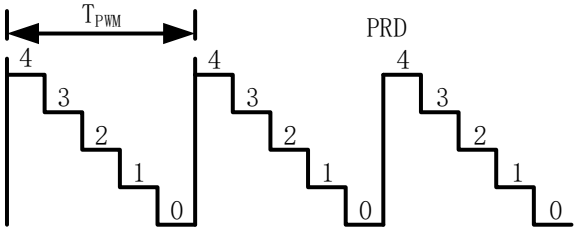


图 18-11 向下计数模式的时基频率和周期

（3）中心对齐计数模式

在该模式下,EPWMx_TBCTR 从 0 开始递增,一直到 EPWMx_TBPRD 值。当到达 EPWMx_TBPRD 值后，EPWMx_TBCTR 开始递减直到为 0。此时，计数器重复该模式并开始递增。

在该模式下，周期 T_{PWM} 和频率 F_{PWM} 计算公式为：

$$T_{PWM} = 2 * TBPRD * T_{TBCLK}$$

$$FPWM = \frac{1}{TPWM}$$

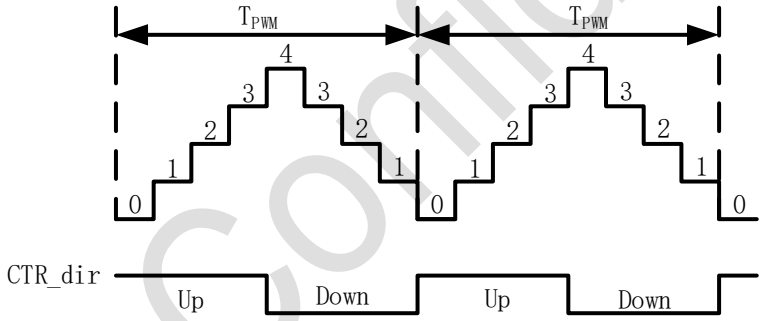


图 18-12 中心对齐计数模式的时基频率和周期

18.3.2.3 时基周期影子寄存器

时基周期寄存器 (EPWMx_TBPRD) 有一个影子寄存器。影子寄存器功能允许寄存器更新与硬件同步。以下定义用于描述 EPWM 模块中的所有影子寄存器：

- 活动寄存器（Active Register）：活动寄存器控制硬件，并负责硬件引起或调用的动作。
- 影子寄存器（Shadow Register）：影子寄存器为活动寄存器提供临时保持位置，对任何控制硬件没有直接影响。影子寄存器为活动寄存器提供缓冲，即提供一个临时保存位置，所以它对任何硬件控制没有直接影响。为了防止由于软件异步修改寄存器而导致的数据损坏或误操作，影子寄存器会在一个特定时间点上，将内容转移到活动寄存器。

影子周期寄存器的存储器地址与活动寄存器相同。写入或读取哪个寄存器由 EPWMx_TBCTL[PRDLD] 位决定。该位决定启用和禁用 EPWMx_TBPRD 影子寄存器：

- 时基周期影子模式：当 EPWMx_TBCTL[PRDLD] = 0 时，EPWMx_TBPRD 影子寄存器被启用。对 EPWMx_TBPRD 存储器地址的读取和写入操作指向影子寄存器。当其时基计数器等于零 (EPWMx_TBCTR = 0x00) 和/或由 EPWMx_TBCTL2[PRDLDSYNC] 位确定的同步事件发生时，

Puya Confidential

影子寄存器的内容被传输到活动寄存器（EPWMx_TBPRD (Active) ← EPWMx_TBPRD (shadow)）。仅当 EPWMx_TBCTL[PRDLD] = 0 时，PRDLDSYNC 位有效。默认情况下，EPWMx_TBPRD 影子寄存器被启用。SYNC 输入的源在同步章节中解释。

通过配置全局加载配置寄存器（EPWMx_GLDCFG）中的相应位，全局加载控制机制也可用于时基周期寄存器。当选择全局加载模式时，对于所有启用此模式的寄存器，从影子寄存器到活动寄存器的内容传输发生在由全局影子到活动加载控制寄存器（EPWMx_GLDCTL）中的配置位定义的同一事件上。全局加载控制机制在全局装载章节中解释。

- 时基周期立即加载模式：如果选择立即加载模式（EPWMx_TBCTL[PRDLD] = 1），则对 EPWMx_TBPRD 存储器地址的读取或写入直接指向活动寄存器。

18.3.2.4 时基计数器工作模式

时基计数器四种工作模式如下：

- 向上计数模式（非对称）
- 向下计数模式（非对称）
- 中心对齐计数模式（对称）
- 冻结模式（时基计数器保持当前值不变）

在向上计数模式、向下计数模式和中心对齐计数模式下，产生事件的时间以及时基响应 EPWMxSYNCI 信号的方式如下图所示。

向上计数模式

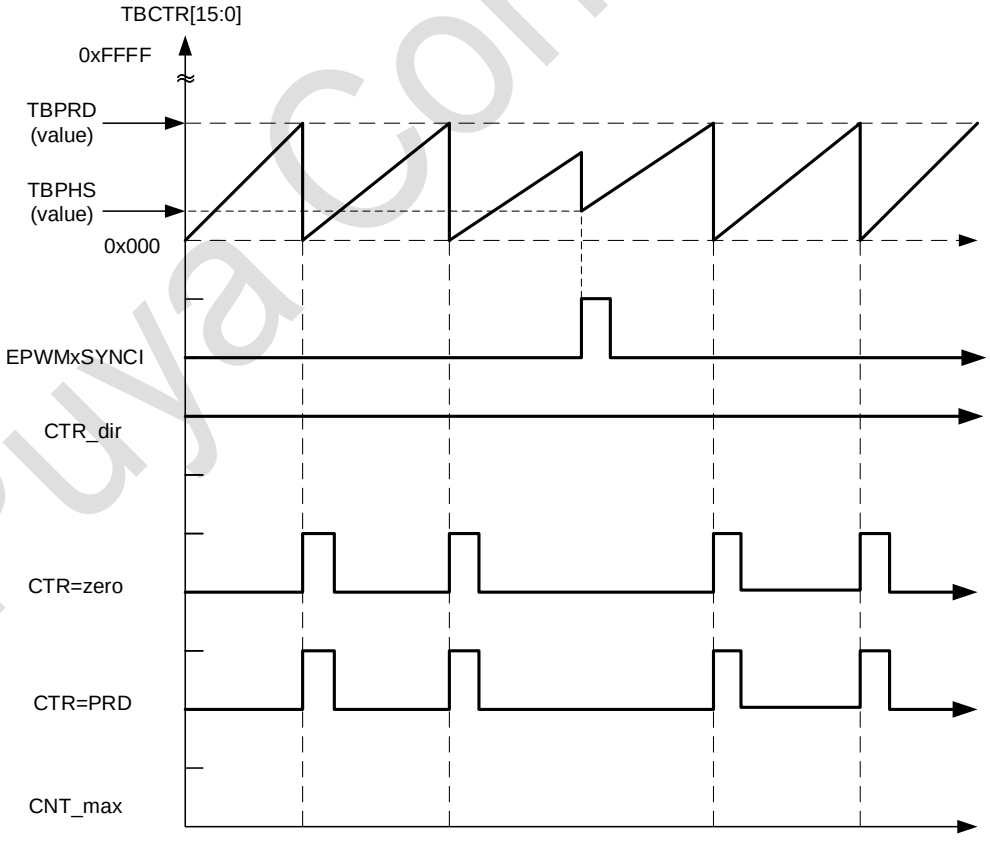


图 18-13 向上计数模式波形图

Puya Confidential

向下计数模式

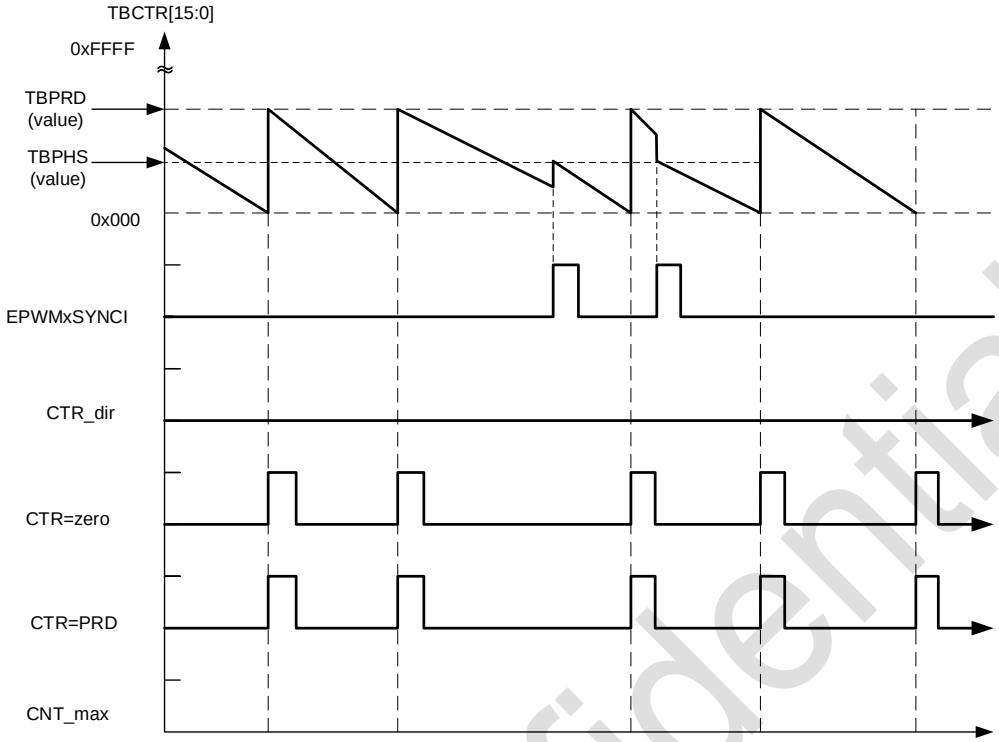


图 18-14 向下计数模式波形图

中心对齐计数模式

在该模式下 PHSDIR=0 时，时基计数器计数方向在同步事件后向下计数。

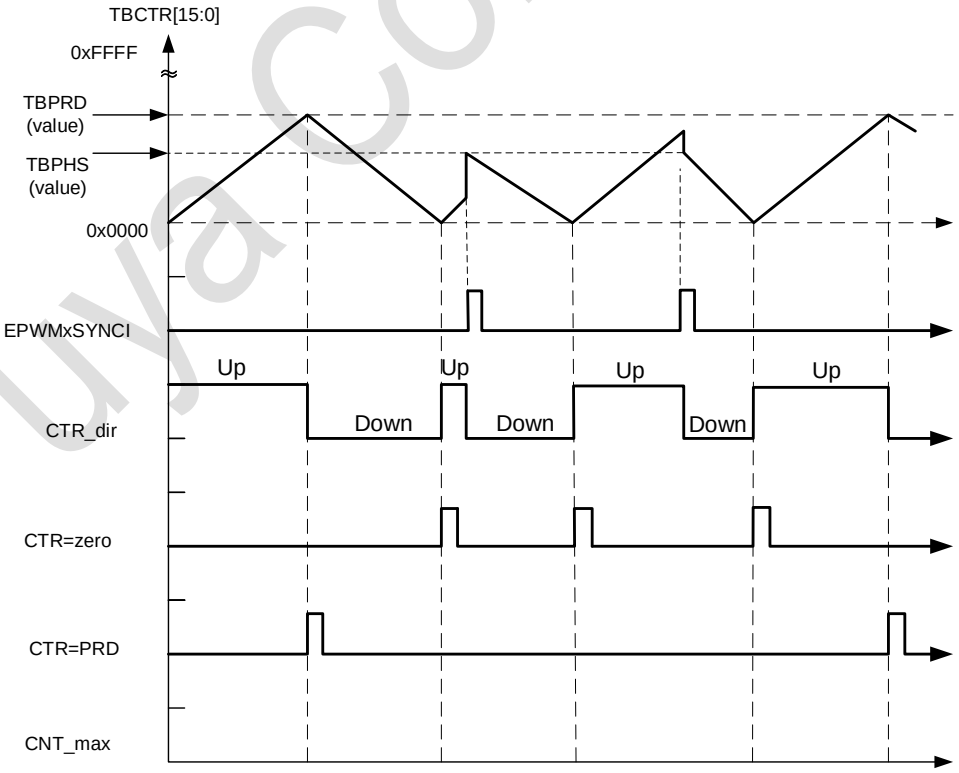


图 18-15 中心对齐计数模式波形图（PHSDIR=0）

Puya Confidential

在该模式下 PHSDIR=1 时，时基计数器计数方向在同步事件后向上计数。

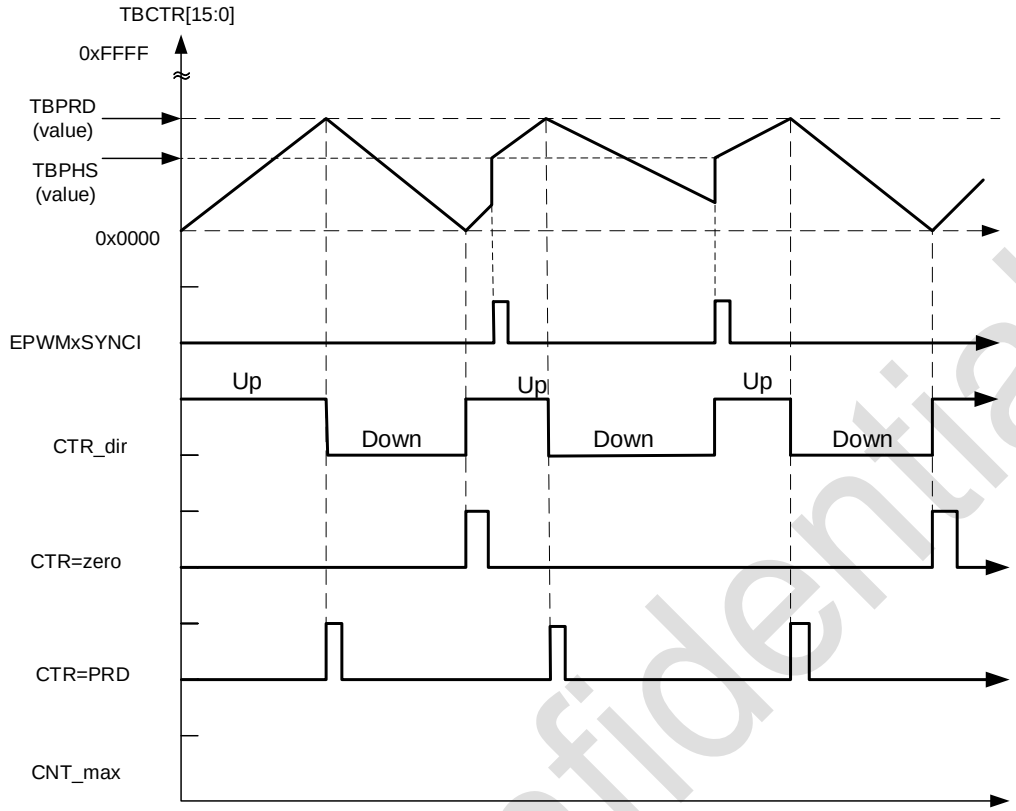


图 18-16 中心对齐计数模式波形图（PHSDIR=1）

冻结模式

在该模式下，时基计数器维持当前值不变。

18.3.2.5 同步

时基时钟同步

外设时钟（RCC 模块）使能寄存器中的 EPWM_TBCLKSYNC 位可以全局同步所有已使能的 EPWM 模块，使其与 TBCLK 同步。当置位时，所有已使能的 EPWM 模块时钟同时启动，并与 TBCLK 的第一个上升沿对齐。为了最佳同步的 TBCLK，每个 EPWM 模块的预分频器必须设置成一样的。

使能 EPWMCLK 的操作步骤如下:

1. 使能 RCC_APBENR2 寄存器中的 EPWMEN 位。
2. 使能 EPWMx_CLKEN 寄存器中对应 EPWMx 模块的时钟使能位（默认为 1）。
3. 清除 RCC 模块中 EPWM_TBCLKSYNC。
4. 配置 EPWM 模块。
5. 设置 RCC 模块中 EPWM_TBCLKSYNC。

时基计数器同步

同步方案增加了 EPWM 模块同步的灵活性。每个 EPWM 模块都有一个同步输入/输出和外设同步输出。每个 EPWM 模块的同步输入源和同步输出选择最高可达 32 个。时基计数器同步示意图如下图所

Puya Confidential

示，EXTSYNCIN1/2 分别源自 INPUTXBAR5/6，可以选择 GPIO 作为同步输入。在使用 EPWM_SYNCI_SELx_y 寄存器选择同步链时，建议最长路径不超过 4 个 EPWM 模块传播路径。

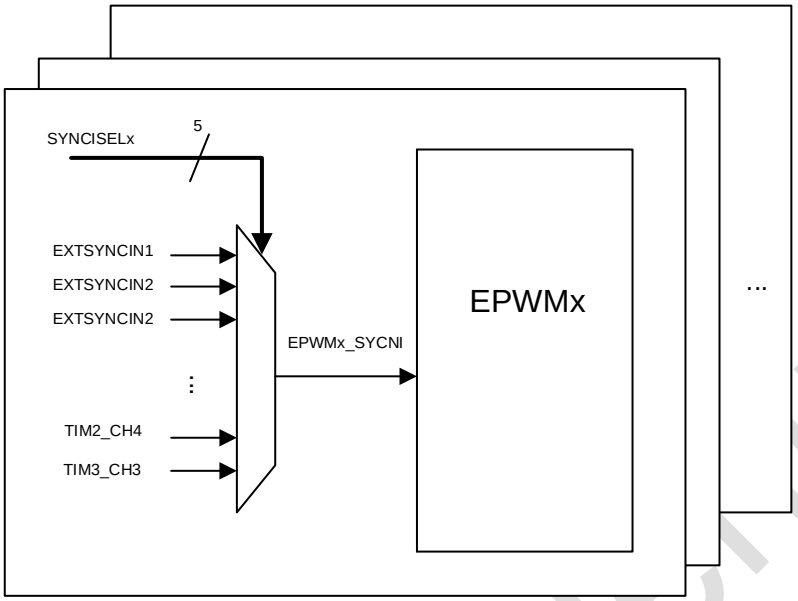


图 18-17 时基计数器同步结构示意图

6 个 EPWM 的同步输入选择源如下表：

表 18-5 EPWMx 同步输入选择

SYNCISEL	EPWM1	EPWM2	EPWM3	EPWM4	EPWM5	EPWM6
0	EXTSYNCIN1	EPWM1_SYNC_OUT	EPWM2_SYNC_OUT	EPWM1_SYNC_OUT	EPWM4_SYNC_OUT	EPWM5_SYNC_OUT
1	EXTSYNCIN1	EXTSYNCIN1	EXTSYNCIN1	EXTSYNCIN1	EXTSYNCIN1	EXTSYNCIN1
2	EXTSYNCIN2	EXTSYNCIN2	EXTSYNCIN2	EXTSYNCIN2	EXTSYNCIN2	EXTSYNCIN2
3	EPWM1_SYNC_OUT	EPWM1_SYNC_OUT	EPWM1_SYNC_OUT	EPWM1_SYNC_OUT	EPWM1_SYNC_OUT	EPWM1_SYNC_OUT
4	EPWM2_SYNC_OUT	EPWM2_SYNC_OUT	EPWM2_SYNC_OUT	EPWM2_SYNC_OUT	EPWM2_SYNC_OUT	EPWM2_SYNC_OUT
5	reserved	reserved	reserved	reserved	reserved	reserved
6	EPWM4_SYNC_OUT	EPWM4_SYNC_OUT	EPWM4_SYNC_OUT	EPWM4_SYNC_OUT	EPWM4_SYNC_OUT	EPWM4_SYNC_OUT
7	EPWM5_SYNC_OUT	EPWM5_SYNC_OUT	EPWM5_SYNC_OUT	EPWM5_SYNC_OUT	EPWM5_SYNC_OUT	EPWM5_SYNC_OUT
8	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
9	TIM1_TRGO	TIM1_TRGO	TIM1_TRGO	TIM1_TRGO	TIM1_TRGO	TIM1_TRGO
10	TIM8_TRGO	TIM8_TRGO	TIM8_TRGO	TIM8_TRGO	TIM8_TRGO	TIM8_TRGO
11	TIM1_TRGO2	TIM1_TRGO2	TIM1_TRGO2	TIM1_TRGO2	TIM1_TRGO2	TIM1_TRGO2
12	TIM8_TRGO2	TIM8_TRGO2	TIM8_TRGO2	TIM8_TRGO2	TIM8_TRGO2	TIM8_TRGO2
13	TIM2_TRGO	TIM2_TRGO	TIM2_TRGO	TIM2_TRGO	TIM2_TRGO	TIM2_TRGO
14	TIM3_TRGO	TIM3_TRGO	TIM3_TRGO	TIM3_TRGO	TIM3_TRGO	TIM3_TRGO
15	TIM6_TRGO	TIM6_TRGO	TIM6_TRGO	TIM6_TRGO	TIM6_TRGO	TIM6_TRGO
16	TIM7_TRGO	TIM7_TRGO	TIM7_TRGO	TIM7_TRGO	TIM7_TRGO	TIM7_TRGO
17	TIM1_CH1	TIM1_CH1	TIM1_CH1	TIM1_CH1	TIM1_CH1	TIM1_CH1

Puya Confidential

18	TIM1_CH2	TIM1_CH2	TIM1_CH2	TIM1_CH2	TIM1_CH2	TIM1_CH2
19	TIM1_CH3	TIM1_CH3	TIM1_CH3	TIM1_CH3	TIM1_CH3	TIM1_CH3
20	TIM1_CH4	TIM1_CH4	TIM1_CH4	TIM1_CH4	TIM1_CH4	TIM1_CH4
21	TIM1_CH5	TIM1_CH5	TIM1_CH5	TIM1_CH5	TIM1_CH5	TIM1_CH5
22	TIM1_CH6	TIM1_CH6	TIM1_CH6	TIM1_CH6	TIM1_CH6	TIM1_CH6
23	TIM8_CH1	TIM8_CH1	TIM8_CH1	TIM8_CH1	TIM8_CH1	TIM8_CH1
24	TIM8_CH2	TIM8_CH2	TIM8_CH2	TIM8_CH2	TIM8_CH2	TIM8_CH2
25	TIM8_CH3	TIM8_CH3	TIM8_CH3	TIM8_CH3	TIM8_CH3	TIM8_CH3
26	TIM8_CH4	TIM8_CH4	TIM8_CH4	TIM8_CH4	TIM8_CH4	TIM8_CH4
27	TIM8_CH5	TIM8_CH5	TIM8_CH5	TIM8_CH5	TIM8_CH5	TIM8_CH5
28	TIM8_CH6	TIM8_CH6	TIM8_CH6	TIM8_CH6	TIM8_CH6	TIM8_CH6
29	TIM2_CH3	TIM2_CH3	TIM2_CH3	TIM2_CH3	TIM2_CH3	TIM2_CH3
30	TIM2_CH4	TIM2_CH4	TIM2_CH4	TIM2_CH4	TIM2_CH4	TIM2_CH4
31	TIM3_CH3	TIM3_CH3	TIM3_CH3	TIM3_CH3	TIM3_CH3	TIM3_CH3

注释：EXTSYNCIN1 的源为 INPUTXBAR5；EXTSYNCIN2 的源为 INPUTXBAR6；EPWMx_SYNC_OUT 为 EPWM 子模块的同步输出信号； timx_chy 为芯片内部 timer 的通道输出；timx_trgoy 为芯片内部 timer 的 triger 输出信号。

每个 EPWM 模块可以配置为使用或忽略同步输入。如果设置了 EPWMx_TBCTL[PHSEN] 位，则当以下条件之一发生时，EPWM 模块的时基计数器 (EPWMx_TBCTR) 会自动加载相位寄存器 (EPWMx_TBPHS) 的内容：

- EPWMxSYNCI：同步输入脉冲：当检测到输入同步脉冲时，相位寄存器的值被加载到计数器寄存器中 (EPWMx_TBPHS → EPWMx_TBCTR)。此操作在下一个有效的时基时钟 (TBCLK) 边沿发生。从内部控制模块到目标模块的延迟由下式给出：
  - 如果 (TBCLK = EPWMCLK)：2 x EPWMCLK
  - 如果 (TBCLK < EPWMCLK)：1 x TBCLK
- 软件强制同步脉冲：向 EPWMx_TBCTL[SWFSYNC] 控制位写入 1 会调用软件强制同步。此脉冲与同步输入信号进行 OR 运算，因此具有与 EPWMxSYNCI 上的脉冲相同的效果。
- 数字比较事件同步脉冲：DCAEVT1 和 DCBEVT1 数字比较事件可以配置为生成同步脉冲，这些脉冲具有与 EPWMxSYNCI 相同的影响。

注意: 如果 EPWMxSYNCI 信号保持高电平，同步不会持续发生。EPWMxSYNCI 是上升沿触发的。

这个功能使 EPWM 模块之间能够自动实现时基同步。通过同步不同的 EPWM 模块，可以对它们生成的波形增加超前或滞后的相位控制。在中心对齐计数模式下，同步事件发生后通过 PHSDIR 位配置时基计数器的计数的方向。新的方向独立于同步事件之前的方向。在向上或向下计数模式下会忽略 PHSDIR 位。

清零 EPWMx_TBCTL[PHSEN] 位可以让 EPWM 模块忽略同步输入脉冲，但是同步脉冲仍然可以传至 EPWMxSYNCO 并用同步其他 EPWM 模块。

Puya Confidential

从 EPWM 子模块输出的 EPWMx_SYNCO 还需要经过一个选择器才能到 EPWMx SYNC_OUT，当然，选择器也有更多的信号源可以输出至 EPWMx SYNC_OUT，可以通过 EPWM_SYNCO_SELx_y 来决定。除了 SELx=0 之外，其他选项为芯片内部信号源，暂不公开。建议 EPWM_SYNCO_SELx_y 保持默认值 0。

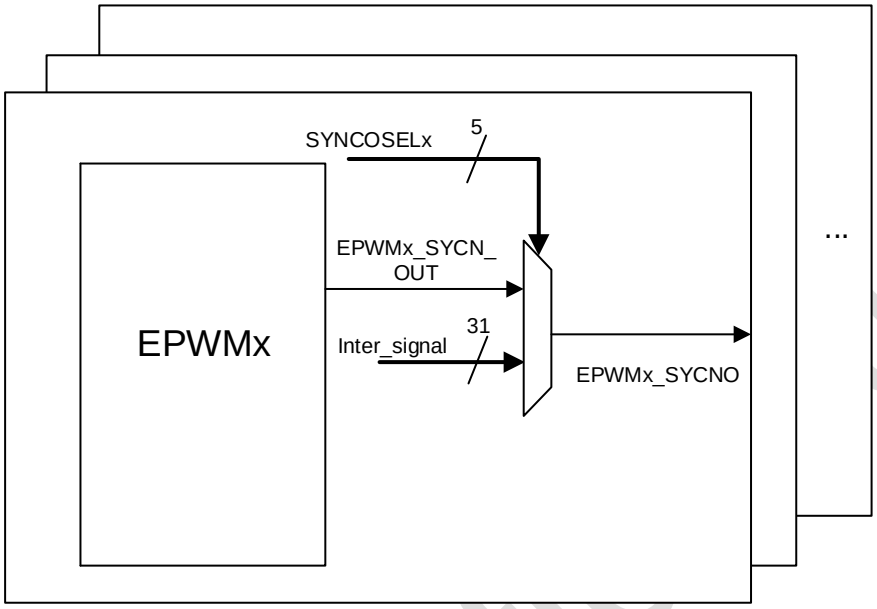


图 18-18 同步输出结构示意图

6 个 EPWM 的同步输出选择源如下表：

表 18-6 EPWMx 同步输出源选择

SYNCOSEL	EPWM1	EPWM2	EPWM3	EPWM4	EPWM5	EPWM6
0	EPWM1_SYNC_OUT	EPWM2_SYNC_OUT	保留	EPWM4_SYNC_OUT	EPWM5_SYNC_OUT	保留

片外同步输出

EPWM 可以将同步输出信号输出至片外，实现与片外计数器同步的效果。其输出信号 EPWM_SCOUT 可以通过寄存器 SCOUT_SEL[2:0]选择输出源，如下表

表 18-7 片外同步输出源选择

SELECT(3-bit)	Signal
0	EPWM1_SYNCO
1	EPWM2_SYNCO
2	reserved
3	EPWM4_SYNCO
4	EPWM5_SYNCO
5	reserved

同步输出信号默认是经过至少 8 个周期展宽，也可以通过 SCOUT_EXP_BYPASS 选择跳过展宽模块。

Puya Confidential

锁相多个 EPWM 模块的时基时钟

通过 TBCLKSYNC 位来全局同步该器件上所有已使能的 EPWM 模块的时钟。当 TBCLKSYNC=0 时，停止所有 EPWM 模块的时基时钟。当 TBCLKSYNC=1 时，在对齐的 TBCLK 上升沿使能所有 EPWM 时基时钟。为了最佳同步的 TBCLK，每个 EPWM 模块的预分频器必须设置成一样的。
同步 EPWM 时钟操作步骤如下：

1. 使能 RCC_APBENR2 寄存器中的 EPWMEN 位。
2. 使能 EPWMx_CLKEN 寄存器中对应 EPWMx 模块的时钟使能位（默认为 1）。
3. 清除 TBCLKSYNC，任何已使能的 EPWM 模块中的时基时钟都将被停止
4. 配置期望的 EPWM 模式和预分频值
5. 设置 TBCLKSYNC

18.3.2.6 同时写入多个 EPWMx_TBPRD 和 EPWMx_CMPx 寄存器

在变频应用中，EPWMx_TBPRD 和 EPWMx_CMPx 寄存器需要在 EPWM 模块之间同时写入。这避免了在修改完所有 EPWM 的寄存器之前，EPWMx_TBCTR=0 或 T EPWMx_BCTR= EPWMx_TBPRD 脉冲强制影子寄存器装载到活动寄存器，从而导致部分寄存器从新的影子值加载，而其他寄存器从旧的影子值加载。为了支持这一点，在 EPWMx_TBPRD、EPWMx_CMPx 寄存器增加了 EPWM 模块间寄存器的链接方案。

请参阅 EPWMx_PWMXLINK 的寄存器描述，以查看相应 EPWM 的链接寄存器位字段值。使用 EPWMx_PWMXLINK 的一个例子是通过 EPWM2 的 EPWMx_PWMXLINK [CMPALINK] 寄存器位字段将 EPWM2 的 EPWMx_CMPA 与 EPWM1 的 EPWMx_CMPA 链接起来。在这种情况下，对 EPWM1 的 EPWMx_CMPA 的写入也会更改 EPWM2 的 EPWMx_CMPA 值。

18.3.2.7 全局装载

如果使能全局装载功能，则对于所有使能此模式的寄存器，影子寄存器内容到活动寄存器的转移发生在同一时刻，由 EPWMx_GLDCTL[GLDMODE] 位域决定该时刻。当 EPWMx_GLDCTL[GLD]=1 时，忽略各寄存器自己的影子寄存器到活动寄存器的加载事件选择，而由 GLDCFG 寄存器使能的对应寄存器的全局加载配置。

当 EPWMx_GLDCTL[GLD]=1 且 EPWMx_GLDCFG 寄存器中相应位为 0 时，相应寄存器不受全局加载模式影响。影子寄存器装载到活动寄存器的时刻由各寄存器自己的影子寄存器到活动寄存器加载事件选择位决定。

Puya Confidential

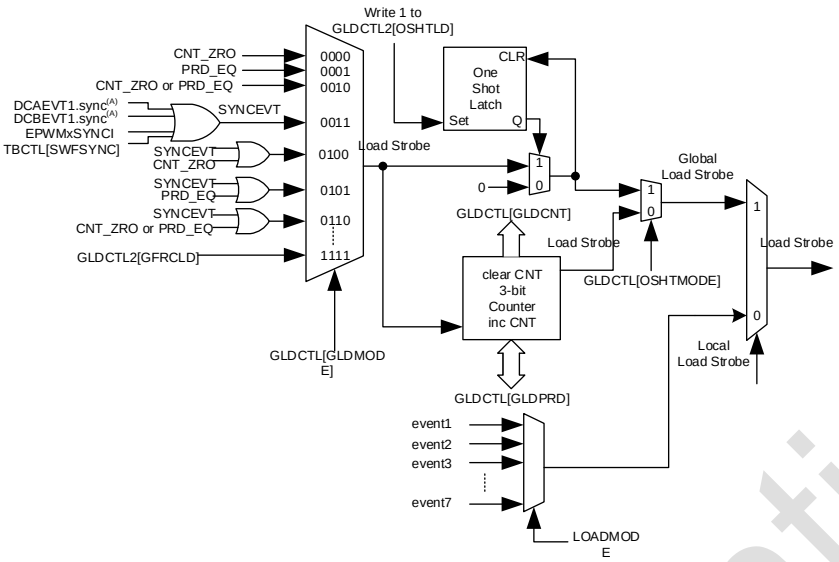


图 18-19 全局装载结构框图

注意：仅在 PHSEN 置位时，SYNCEVT 信号才会通过同步逻辑传递。

脉冲预分频功能

该特性提供了如下配置：EPWMx_GLDCTL[GLDMODE] 位域所选择的全局加载事件每发生“N”次，就发生一次影子寄存器到活动寄存器的转移。对于不能或没有配置为使用全局加载机制的寄存器，即 EPWMx_GLDCTL[GLD]=0 或 EPWMx_GLDCFG 寄存器中相应位为 0 时，此预分频功能不可用。

单次装载模式

限制影子寄存器到活动寄存器的转移只发生一次。当 OSHTLD=1 时，所有配置为使用全局加载机制的寄存器，在 EPWMx_GLDCTL[GLDMODE] 位域所选事件出现时，发生影子寄存器到活动寄存器的转移，同时 OSHTLD 位被自动清零，若要再次发生全局加载，需手动将该位置 1。

使用 GFRCLD 位可以通过软件强制影子寄存器到活动寄存器的转移。GLDCTL2 寄存器也可以通过 GLDCTL2LINK 位链接到多个 EPWM 模块。结合上述的单次装载特性，这提供了一种特定的 EPWM 事件，甚至单个时钟周期内，精确地更新一个或多个 EPWM 模块内的多个 EPWM 寄存器的方法，可以用于变频应用和/或多相交替应用中。

注意：在高分辨率模式下禁止单次装载模式。

单次同步模式

需要配置 EPWMx_TBCTL2[OSHTSYNCMODE] 位和 EPWMx_TBCTL2[OSHTSYNC]位才能使能单次同步模式以生成同步输出信号（SYNCO）脉冲，如下图所示。

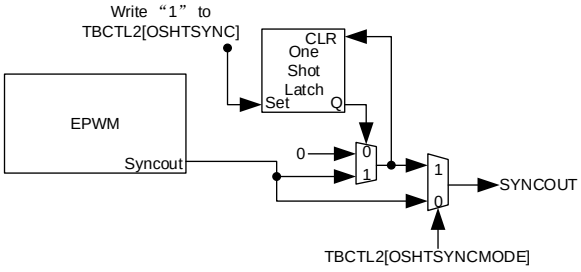


图 18-20 单次同步模式结构框图

Puya Confidential

18.3.3 计数器比较（CC）子模块描述

下图展示了 EPWM 模块内部的计数比较子模块。

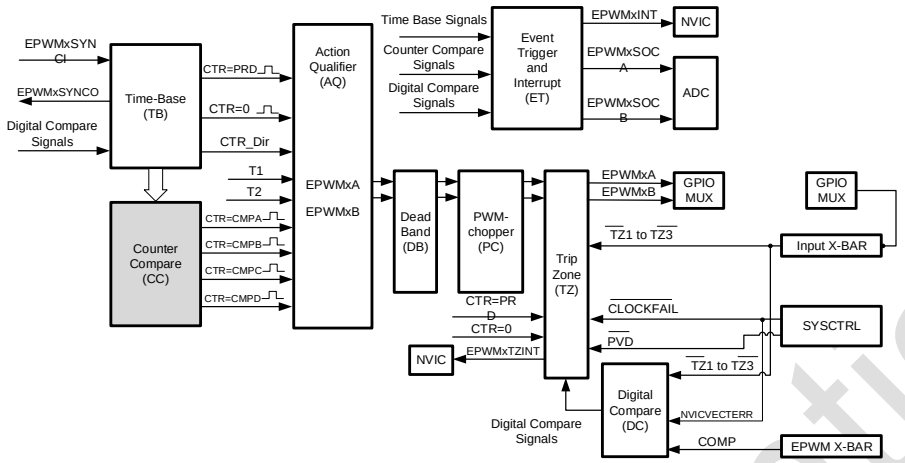


图 18-21 计数比较子模块

18.3.3.1 结构框图

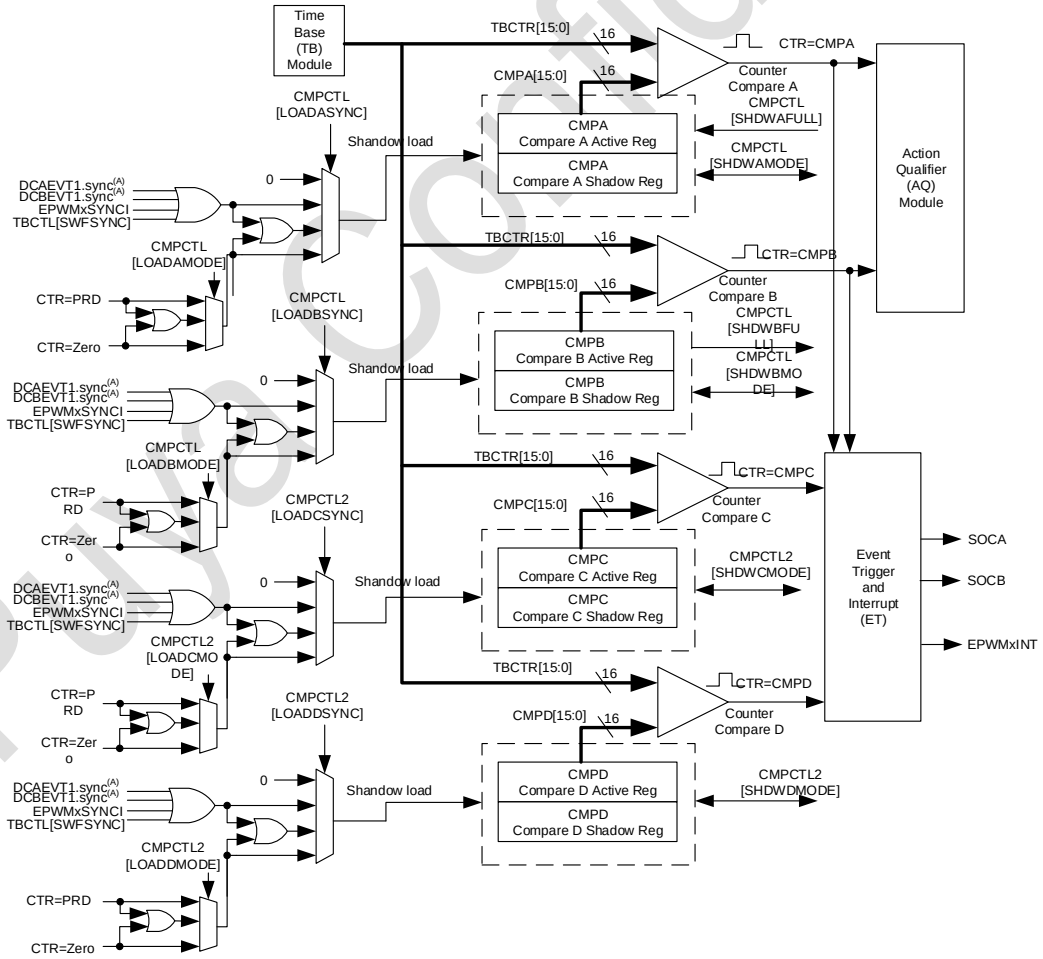


图 18-22 计数器比较子模块结构框图

Puya Confidential

18.3.3.2 工作模式

CC 子模块可以在向上计数模式、向下计数模式和中心对齐模式下生成比较事件。三种计数模式下的计数器比较事件波形如下图所示，产生事件以及 EPWMxSYNCI 信号起作用的时刻。

向上计数模式

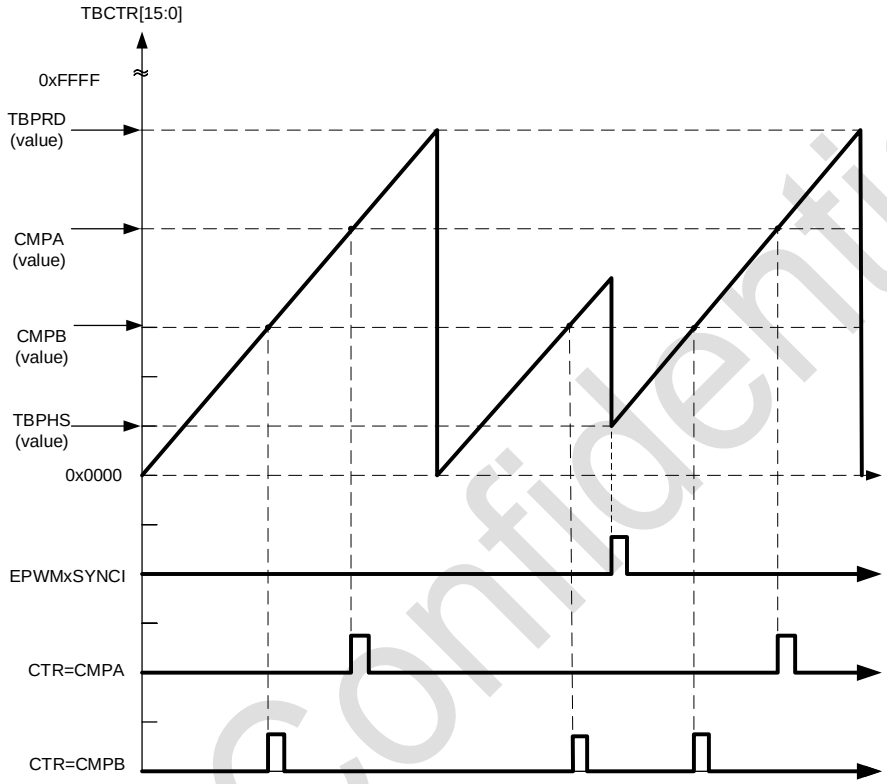


图 18-23 向上计数模式时序图

向下计数模式

EPWMxSYNCI 外部同步事件会导致 EPWMx_TBCTR 计数值跳变。这会导致比较事件被跳过，如下图示，同步事件导致 EPWMx_TBCTR= CMPA 事件被跳过。这种跳过现象被视为正常功能，必须纳入考虑。

Puya Confidential

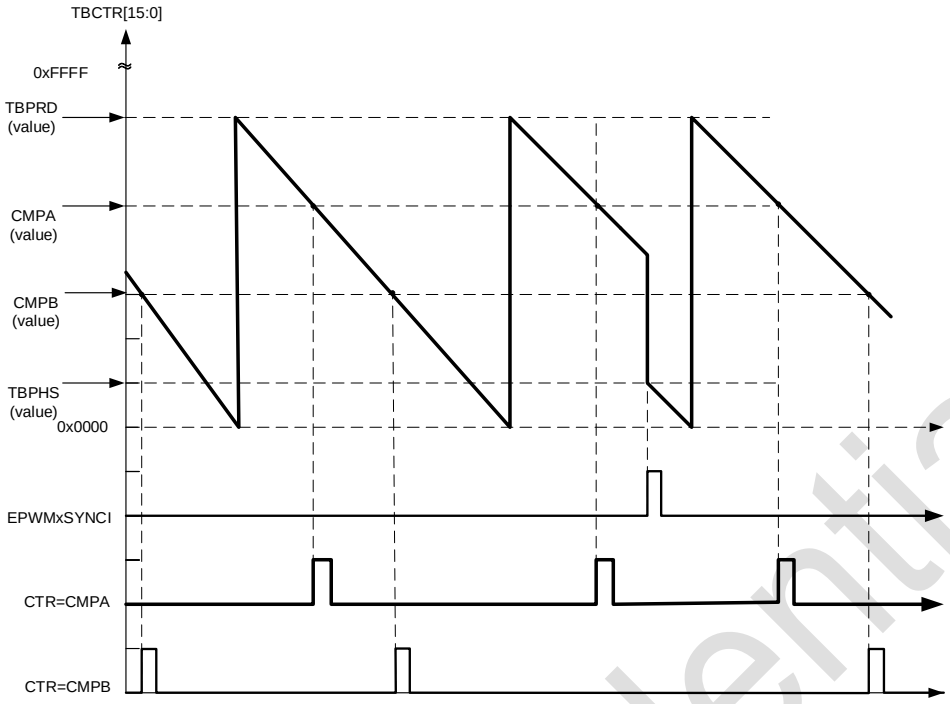


图 18-24 向下计数模式时序图

中心对齐计数模式

如下图所示，EPWMx_ TBCTR 计数方向在同步事件后向下计数。

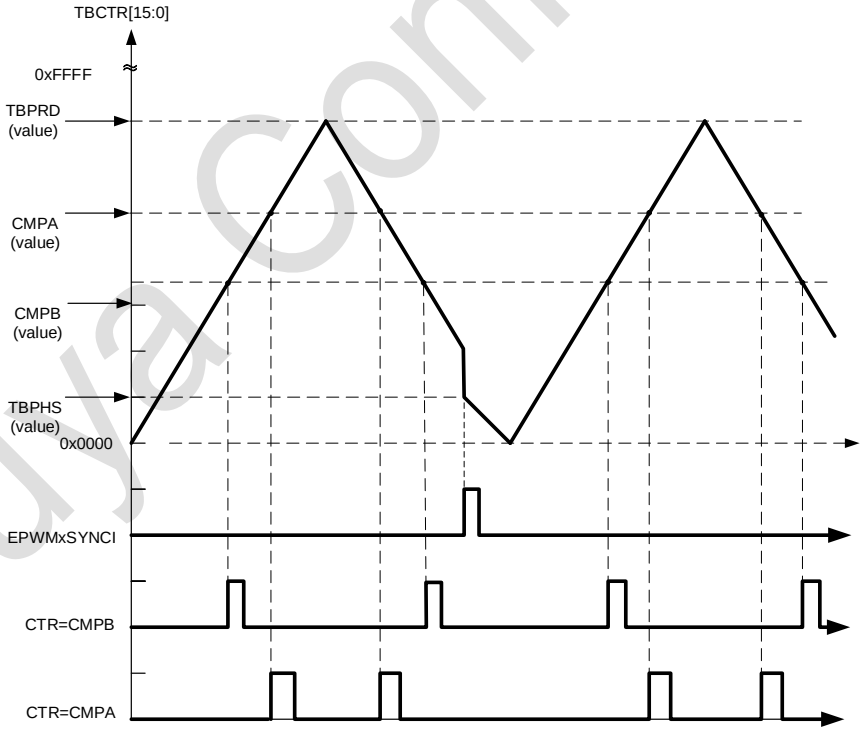


图 18-25 中心对齐计数模式时序图（PHSDIR=0）

如下图所示，EPWMx_TBCTR 计数方向在同步事件后向上计数。

Puya Confidential

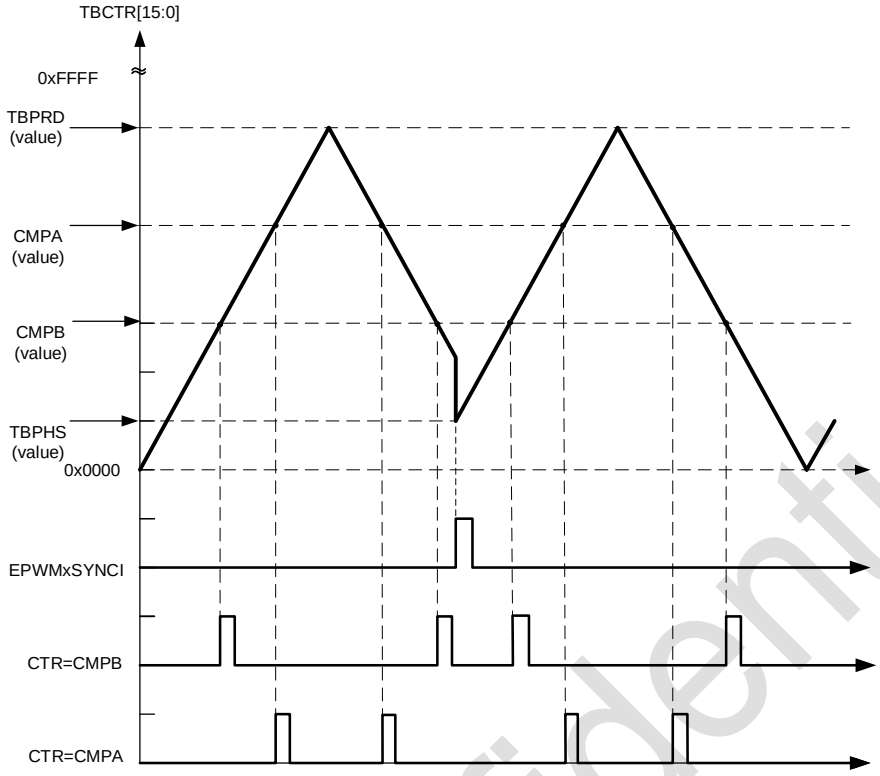


图 18-26 中心对齐计数模式时序图（PHSDIR=1）

18.3.3.3 操作要点

CC 子模块负责产生事件，以供动作限定和事件触发子模块使用。存在四种独立的比较事件：

- EPWMx_TBCTR= CMPA
- EPWMx_TBCTR= CMPB
- EPWMx_TBCTR= CMPC（仅供事件触发器使用）
- EPWMx_TBCTR= CMPD （仅供事件触发器使用）

对于向上计数或向下计数模式：每周期内每个事件仅发生一次

对于中心对齐计数模式：

- 若比较值等于 0x00 或 EPWMx_TBPRD，则每周期内每个事件发生一次
- 若比较值在 0x00 到 EPWMx_TBPRD 之间，则每周期内每个事件发生两次

这些事件被送入 AQ 子模块，在该子模块中，它们根据计数器的方向定性，并在使能时转换为相应的动作，详情请参见动作限定(AQ)子模块功能描述章节。

CMPA 和 CMPB 寄存器分别有一个相关联的，可以使寄存器更新与硬件同步的影子寄存器。为了防止由于软件异步修改寄存器而导致的波形失真或虚假操作，当使用影子寄存器时，活动寄存器的更新只发生在特定时间点上。活动寄存器和影子寄存器有相同的内存地址。由 SHDWAMODE 位和 SHDWBMODE 位决定写入或读取的寄存器。影子模式和立即装载模式的行为描述如下：

（1）影子模式

当 SHDWAMODE =0、SHDWBMODE =0 时，使能 CMPA 和 CMPB 寄存器的影子模式。默认使能 CMPA 和 CMPB 寄存器的影子模式。

Puya Confidential

当使能影子寄存器时，影子寄存器的内容将在以下事件之一转移到活动寄存器，这些事件由 LOADAMODE 位域、LOADBMODE 位域、LOADASYNC 位域和 LOADBSYNC 位域决定：

- EPWMx_TBCTR=0
- EPWMx_TBCTR= EPWMx_TBPRD
- EPWMx_TBCTR=0 且 EPWMx_TBCTR= EPWMx_TBPRD
- 同步事件：由 DCAEVT1 或 DCBEVT1 或 EPWMx 同步输入或软件强制同步脉冲引起的
- LOADAMODE/LOADBMODE 选择的事件或同步事件

该子模块只使用活动寄存器的内容来产生送给动作限定的事件。

（2）立即模式

当 SHDWAMODE=1、SHDWBMODE=1 时，即选择立即加载模式时，那么对寄存器的读写将直接访问活动寄存器。

附加比较器

CC 子模块能够基于两个比较寄存器额外产生两路独立的比较事件，然后提供给 ET 子模块：

- EPWMx_TBCTR= CMPC
- EPWMx_TBCTR= CMPD

CMPC 和 CMPD 寄存器分别有一个默认使能的相关联影子寄存器。活动寄存器和影子寄存器有相同的内存地址。如果 CMPC 和 CMPD 活动寄存器的值与 EPWMx_TBCTR 相等，则 CC 子模块分别产生一个 EPWMx_TBCTR 等于计数比较 C 事件或计数比较 D 事件。通过 SHDWCMODE 位域和 SHDWDMODE 位域来使能或禁止 CMPC 和 CMPD 影子寄存器，影子模式和立即装载模式的行为描述如下：

（1）影子模式

清零 SHDWCMODE 位和 SHDWDMODE 位使能 CMPC 和 CMPD 寄存器的影子模式。默认使能 CMPC 和 CMPD 寄存器的影子模式。

当使能影子寄存器时，影子寄存器的内容将在以下事件之一转移到活动寄存器，这些事件由 LOADCMODE 位域、LOADDMODE 位域、LOADCSYNC 位域和 LOADDSYNC 位域决定：

- EPWMx_TBCTR=0
- EPWMx_TBCTR= EPWMx_TBPRD
- EPWMx_TBCTR=0 且 EPWMx_TBCTR= EPWMx_TBPRD
- 同步事件：由 DCAEVT1 或 DCBEVT1 或 EPWMx 同步输入或软件强制同步脉冲引起
- LOADCMODE/LOADDMODE 选择的事件或同步事件

该子模块只使用活动寄存器的内容来产生送给动作限定的事件。

（2）立即模式

当 SHDWCMODE=1 或 SHDWDMODE=1，即选择立即加载模式时，那么对寄存器的读写将直接访问活动寄存器。

全局装载

可以通过在 EPWMx_GLDCFG 寄存器中配置相应位，使得所有的计数器比较寄存器可以使用全局装载控制机制。如果使能全局装载功能，则使能此模式的寄存器中的影子寄存器内容装载到活动寄存器，装载动作将同时发生在由 EPWMx_GLDCTL 寄存器定义的事件上。全局加载控制机制详情请参见全局装载章节。

Puya Confidential

18.3.4 动作限定(AQ)子模块功能描述

动作限定子模块用于波形构造和 PWM 生成，并决定将哪些事件转换为各种动作类型，从而在 EPWMxA 和 EPWMxB 输出上产生所需的开关波形。

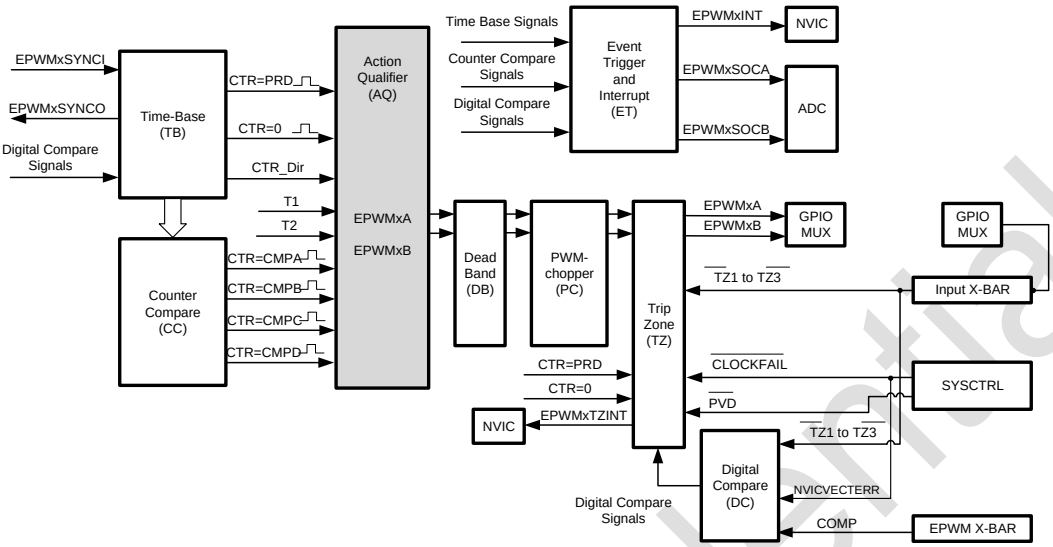


图 18-27 动作限定子模块

18.3.4.1 结构框图

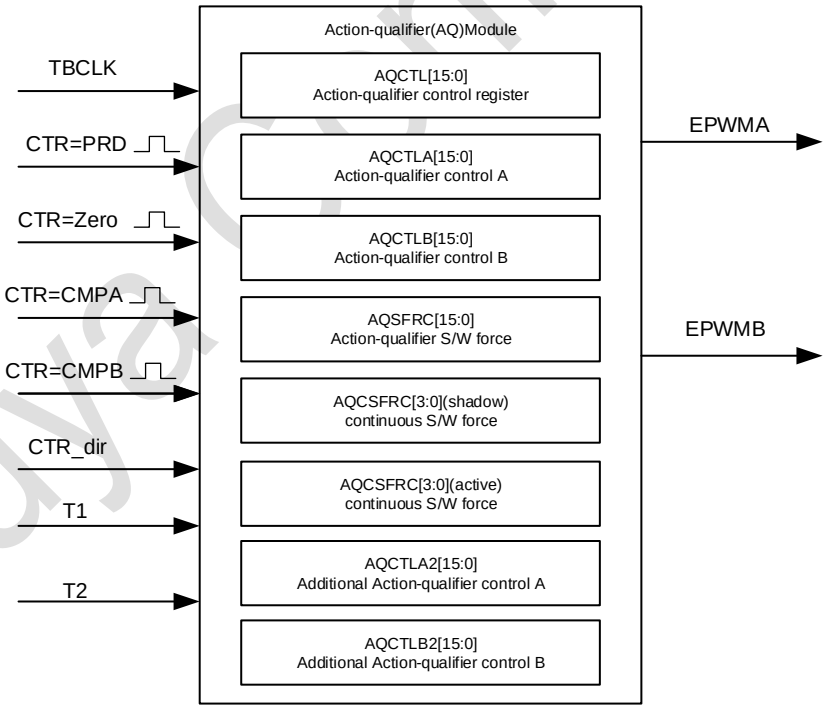


图 18-28 动作限定子模块结构框图

18.3.4.2 输入与输出

动作限定子模块输入

Puya Confidential

AQ 子模块可以用作输入的事件如下所示：

表 18-8 输入事件

输入事件	描述
EPWMx_TBCTR=0	时基计数器等于 0，即 EPWMx_TBCTR 寄存器等于 0
EPWMx_TBCTR= EPWMx_TBPRD	时基计数器等于周期值，即 EPWMx_TBCTR 寄存器等于 EPWMx_TBPRD 寄存器
EPWMx_TBCTR= CMPA	时基计数器等于计数器比较 A，即 EPWMx_TBCTR 寄存器等于 CMPA 寄存器
EPWMx_TBCTR= CMPB	时基计数器等于计数器比较 B，即 EPWMx_TBCTR 寄存器等于 CMPB 寄存器
T1 事件	基于比较器、跳闸、或同步输入事件
T2 事件	基于比较器、跳闸、或同步输入事件
软件强制事件	软件发起的异步事件，由 EPWMx_AQSFRC 和 EPWMx_AQCSFRC 寄存器控制

注意：如果不期望 CSFA/B 位在影子模式下工作，则必须配置 RLDCSP 位为禁用影子模式。

动作限定子模块输出

AQ 子模块控制 EPWMxA 和 EPWMxB 两个输出在特定事件发生时的行为。该子模块的事件输入由计数器方向（向上或向下）进一步定性，这可以在计数器向上计数阶段和向下计数阶段产生相互独立的动作。

表 18-9 可能施加在 EPWMxA 和 EPWMxB 上的动作

动作	描述
置高	EPWMxA 或 EPWMxB 输出被置为高电平
置低	EPWMxA 或 EPWMxB 输出被置位低电平
翻转	将 EPWMxA 或 EPWMxB 输出信号的状态进行翻转。若 EPWMxA 或 EPWMxB 当前为高，则将其拉低。反之将其拉高
无动作	保持 EPWMxA 和 EPWMxB 当前的电平不变。虽然该选项阻止事件在 EPWMxA 和 EPWMxB 输出上产生动作，但这些事件仍然可以触发中断或 ADCSOC。详情请参见事件触发子模块功能描述章节

EPWMxA 或 EPWMxB 输出指定的动作都是相互独立的，在给定的输出上所有事件均可生成动作。例如，EPWMx_TBCTR= CMPA 和 EPWMx_TBCTR= CMPB 都可作用于 EPWMxA 输出。通过相关 PWM 控制寄存器可以配置所有限定器动作。

EPWM_AQTSRCSEL 寄存器用于选择 T1 和 T2 事件源。AQ 子模块的跳闸或数字比较事件的 T1/T2 选择和配置，独立于 TZ 子模块中该事件的配置。一个特定的跳闸事件可以被配置为在 TZ 子模块中引起或不引起跳闸动作，动作限定可以使用相同的事件来生成 T1/T2 以控制 PWM 的产生。

Puya Confidential

18.3.4.3 工作模式

本章中图片所示为使用静态比较寄存器值的 PWM 行为。在运行系统中，在每个周期结束时 CMPA 和 CMPB 活动寄存器从各自的影子寄存器中更新一次，可以指定在 EPWMx_TBCTR=0 或 EPWMx_TBCTR= EPWMx_TBPRD 时发生更新。在特定情况下，可以将基于旧值的动作额外生效一个周期，或者将基于新值的动作延迟一个周期。以下 PWM 配置可以避免这种情况：

向上（或向下）计数模式

（1）生成非对称 PWM

要实现占空比为 0%-100% 的非对称 PWM，使用以下配置：

- 在 EPWMx_TBCTR= EPWMx_TBPRD 时装载 CMPA 或 CMPB
- 使用 EPWMx_TBCTR=0 动作拉高 PWM
- 使用向上（或向下）计数时的比较动作拉低 PWM
- 调节比较值在 0 到 EPWMx_TBPRD+1 之间变化，可实现 0%-100% 的 PWM 占空比

（2）生成带死区的非对称 PWM

要实现占空比为 0%-100% 的非对称 PWM，使用以下配置：

- 当 CMPA 值过于接近 0 或 EPWMx_TBPRD 时，如果 CMPx 小于死区或 CMPx 大于（EPWMx_TBPRD-死区），则 EPWMx_AQCTL 寄存器为 CMPx 指定的动作不起作用
- 为了避免这种情况，只有在 CAU 或 CAD 事件（均为置高或置低时）产生高脉冲或低脉冲时才能改变 EPWMx_AQCTL 寄存器设置
- 确保使能影子模式且软件更新与 PWM 载波周期同步

中心对齐计数模式

（1）生成对称 PWM

- 在等于零时装载 CMPA 或 CMPB，使用大于 1 的 CMPA 或 CMPB 值
- 在等于周期时装载 CMPA 或 CMPB，使用小于 EPWMx_TBPRD-1 的 CMPA 或 CMPB 值

在一个 PWM 周期中始终至少有一个 TBCLK 周期的脉冲的存在，当它很短时，系统往往会忽略它。

（2）生成非对称 PWM

要实现占空比为 50%-0% 的非对称 PWM，使用以下配置：

- 在 EPWMx_TBCTR= EPWMx_TBPRD 时装载 CMPA 或 CMPB
- 使用 EPWMx_TBCTR= EPWMx_TBPRD 动作拉低 PWM
- 使用向上计数时的比较动作拉高 PWM
- 调节比较值在 0 到 EPWMx_TBPRD 之间变化，可实现 50%-0% 的 PWM 占空比

（3）生成带死区的非对称 PWM

Puya Confidential

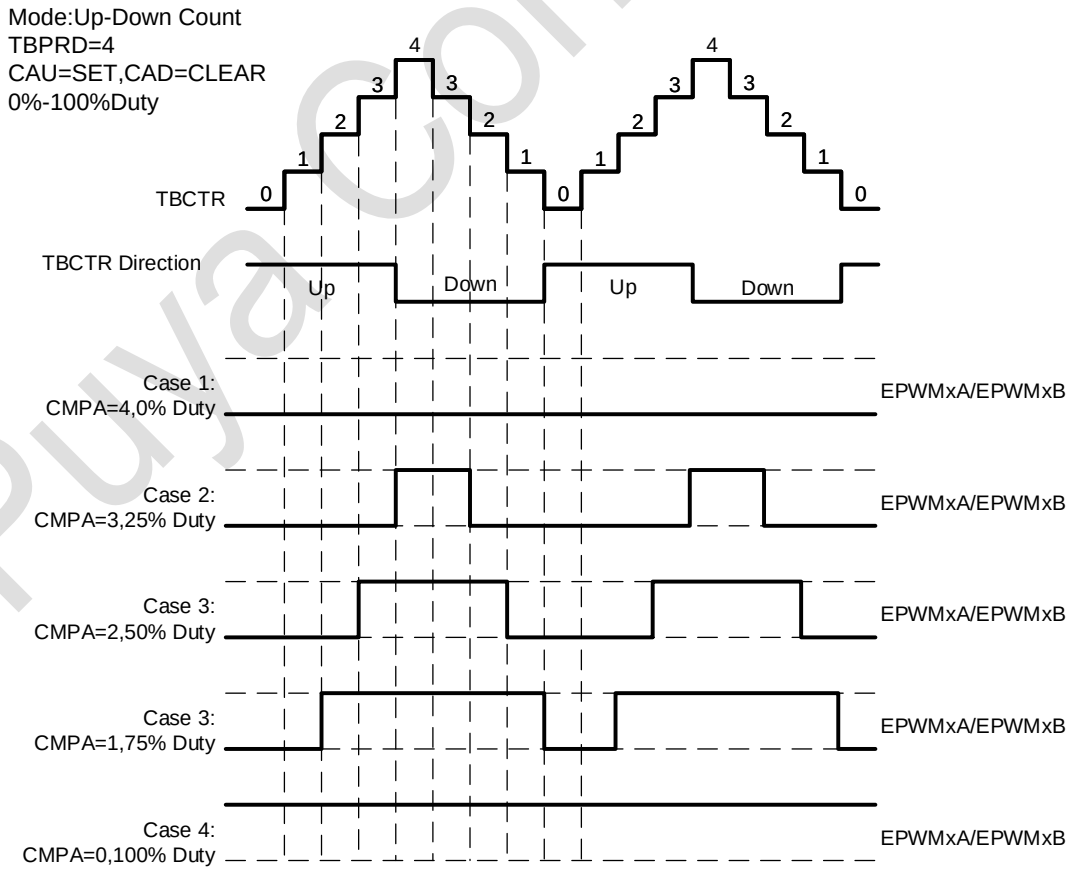
要实现占空比为 0%~100% 的非对称 PWM，使用以下配置：

- 当 CMPA 值过于接近 0 或 EPWMx_TBPRD 时，如果 CMPx 小于死区/2 或 CMPx 大于（EPWMx_TBPRD-死区/2），则 EPWMx_AGCTL 寄存器为 CMPx 指定的动作不起作用
- 为了避免这种情况，只有在在 CAU 或 CAD 事件（均为置高或置低时）产生脉冲或低脉冲时才能改变 EPWMx_AGCTL 寄存器设置
- 确保使能影子模式且软件更新与 PWM 载波周期同步

使用 EPWMx_TBCTR 的中心对齐计数模式生成对称 PWM 波形如下图所示。在该模式下，0%-100%的调制是通过在波形向上计数部分和向下计数部分相等的比较匹配来实现的。使用 CMPA 进行比较示例：

- 当计数器向上计数时，CMPA 匹配将 PWM 输出拉高
- 当计数器向下计数时，比较匹配将 PWM 信号拉低
- 当 CMPA=0 时，PWM 信号在整个周期内为高电平，形成 100% 占空比波形
- 当 CMPA= EPWMx_TBPRD 时，PWM 信号为低，达到 0% 占空比

在实际使用此配置时，需要使用大于或等于 1 的 CMPA/B 值才允许在 EPWMx_TBCTR=0 时装载 CMPA/B。需要使用小于或等于 EPWMx_TBPRD-1 的 CMPA/B 值才允许在 EPWMx_TBCTR= EPWMx_TBPRD 时装载 CMPA/B。因此，在 PWM 周期中至少有一个 TBCLK 周期的脉冲，当它很短时，系统往往会忽略它。



Puya Confidential

图 18-29 中心对齐计数模式对称时序图

常见动作限定配置的 PWM 波形如下图所示，在图和示例中使用 EPWMx_TBPRD、CMPA 和 CMPB 指写入各自寄存器 EPWMx_TBPRD、CMPA 和 CMPB 的值，且硬件使用活动寄存器；CMPX 指 CMPA 或 CMPB；EPWMxA 和 EPWMxB 指 EPWMx 输出信号。

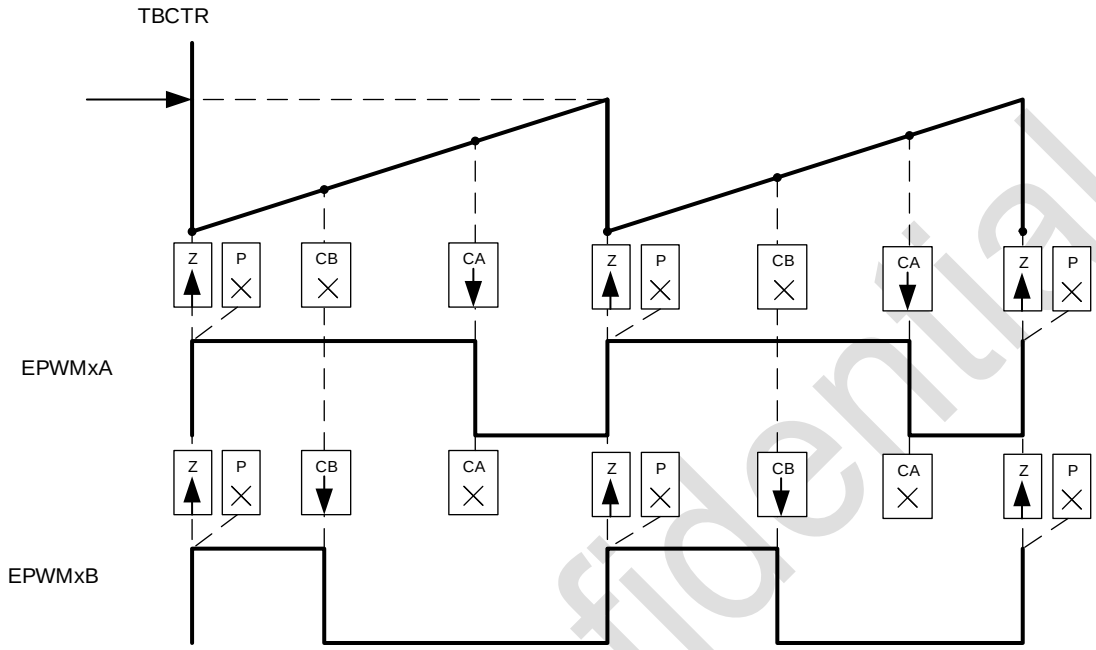


图 18-30 向上计数模式，EPWMxA 和 EPWMxB 上独立调制的单边非对称波形图（高电平有效）

注意：

(1) PWM 周期 = TTBCLK * （EPWMx_TBPRD+1）。

(2) 通过 CMPA/B 设置 EPWMxA/B 的占空比调制且高电平有效。

(3) EPWMx_TBCTR 等于 0 和周期的动作不是同时发生的，它们被一个 TBCLK 周期分隔开。EPWMx_TBCTR 从周期结束为 0。

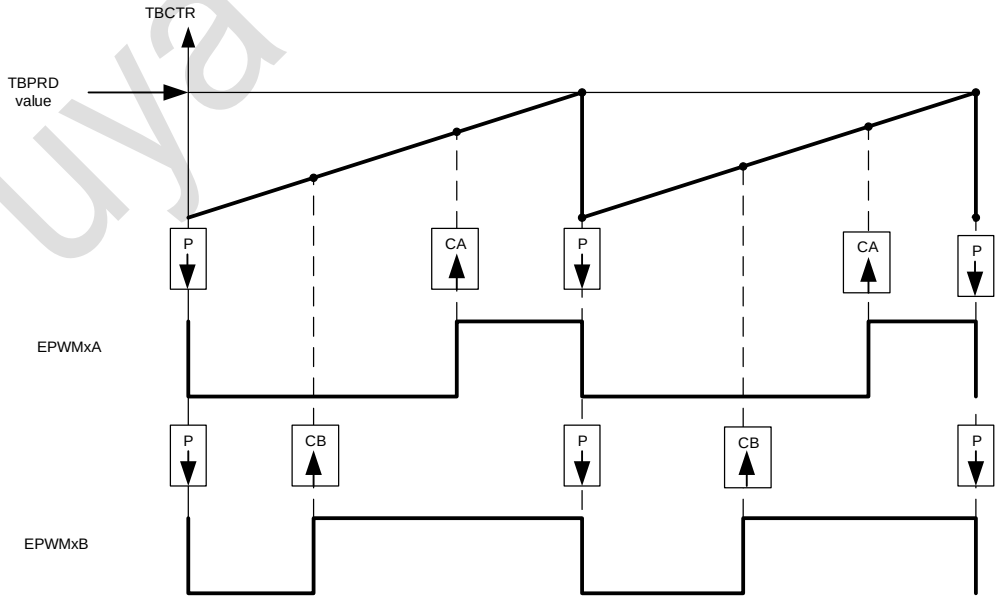


图 18-31 向上计数模式，EPWMxA 和 EPWMxB 上独立调制的单边非对称波形图（低电平有效）

Puya Confidential

注意：

(1) PWM 周期 = TTBCLK * （EPWMx_TBPRD+1）。

(2) 通过 CMPA/B 设置 EPWMxA/B 的占空比调制且低电平有效。

(3) EPWMx_TBCTR 等于 0 和周期的动作不是同时发生的，它们被一个 TBCLK 周期分隔开。EPWMx_TBCTR 从周期结束为 0。

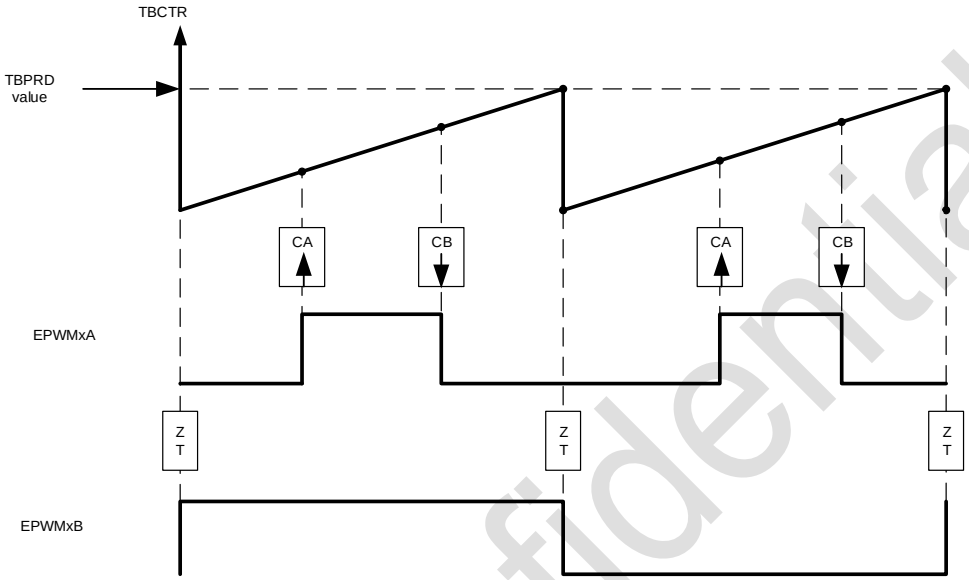


图 18-32 向上计数模式，EPWMxA 上独立调制的脉冲放置型非对称波形图

注意：

(1) PWM 频率 = 1/[TTBCLK * （EPWMx_TBPRD+1）]。

(2) 可以在 PWM 周期内（0~ EPWMx_TBPRD）任意位置设置脉冲。

(3) 高电平占空比与（CMPB- CMPA）成比例关系。

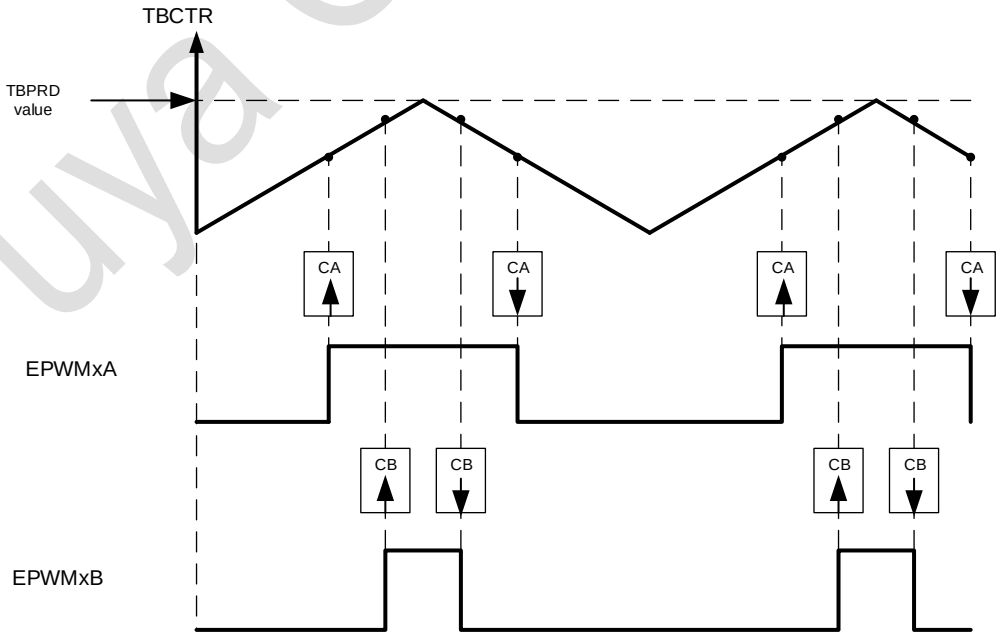


图 18-33 中心对齐计数模式，EPWMxA 和 EPWMxB 上独立调制的双边沿对称波形图（低电平有效）

Puya Confidential

注意：

（1）PWM 周期 = 2* TTBCLK * EPWMx_TBPRD。

（2）通过 CMPA/B 设置 EPWMxA/B 的占空比调制且低电平有效。

（3）独立电源开关可以由输出 EPWMxA 和 EPWMxB 驱动。

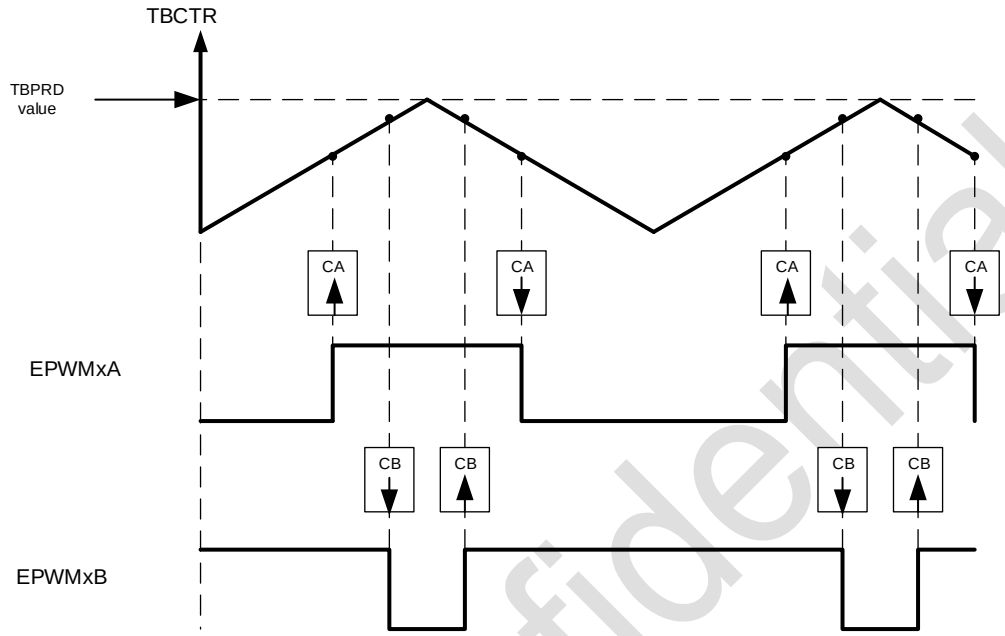


图 18-34 中心对齐计数模式，EPWMxA 和 EPWMxB 上独立调制的双边沿对称波形图（互补）

注意：

（1）PWM 周期 = 2* TTBCLK * EPWMx_TBPRD。

（2）通过 CMPA/B 设置 EPWMxA/B 的占空比调制且低电平有效。

（3）互补电源开关可以由输出 EPWMxA 和 EPWMxB 驱动。

（4）通过软件完全可以编程边沿放置死区（死区= CMPB- CMPA），经典边沿延迟方法也适用于死区模块。

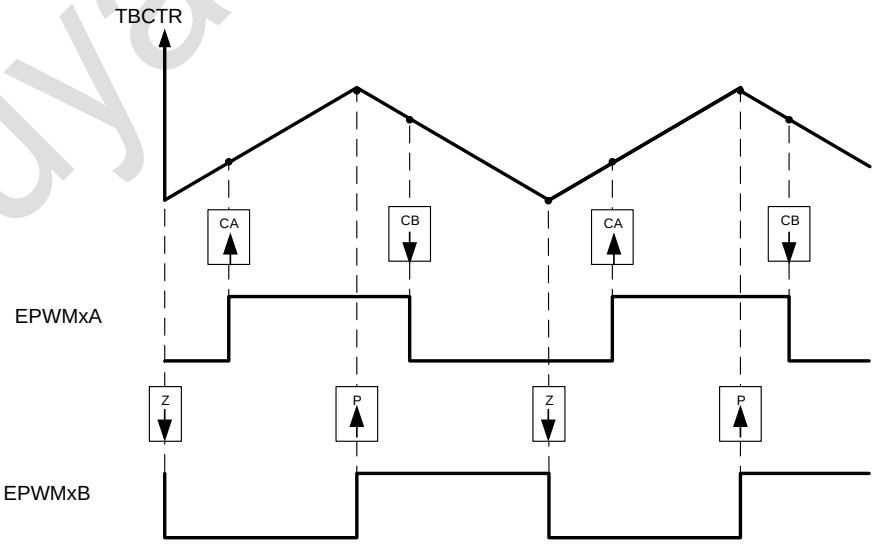


图 18-35 中心对齐计数模式，EPWMxA 和 EPWMxB 上独立调制的双边沿非对称波形图（低电平有效）

Puya Confidential

注意：

(1) PWM 周期 = 2* TTBCLK * EPWMx_TBPRD。

(2) 可以 PWM 周期内不对称定位上升沿和下降沿。

(3) 通过 CMPA 和 CMPB 设置 EPWMxA 的占空比调制，EPWMxB 的占空比调制固定为 50%。（CMPA+ CMPB）与 EPWMxA 低电平占空比成正比。

(4) 如果将该示例更改为高电平有效，则需要在 CMPA 上拉低和在 CMPB 上置高。

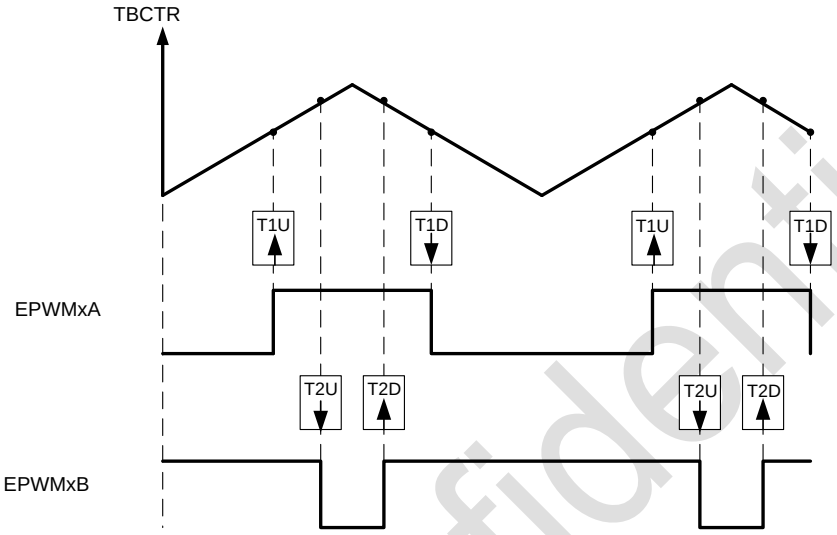


图 18-36 中心对齐计数模式，使用 T1 和 T2 事件生成 PWM 波形图

注意：

(1) PWM 周期 = 2* TTBCLK * EPWMx_TBPRD。

(2) 当 EPWMx_TBCTR 向上计数和向下计数时，将独立 TZ1 和 TZ2 事件动作用来产生 EPWMxA 和 EPWMxB 输出。

18.3.4.4 事件优先级

EPWM 动作限定可以同时接收多个事件。硬件会为这些事件分配一个优先级。后发生的事件优先级更高，软件强制事件优先级最高。各种模式的动作限定事件优先级如下表格所示，优先级为 1 的优先级最高，数值越大则优先级最低。

在向上计数模式中，计数器方向总是向上，因此不会接收到向下计数事件。而在向下计数模式中，计数器方向总是向下，因此不会接收到向上计数事件。在中心对齐计数模式中，EPWMx_TBCTR 的优先级随着方向的变化略有变化。

表 18-10 向上或向下计数模式的动作限定事件优先级

计数模式	优先级	事件
向上计数模式	1	软件强制事件
	2	EPWMx_TBCTR= EPWMx_TBPRD
	3	向上计数时的 T1
	4	向上计数时的 T2
	5	向上计数时计数器等于 CMPB

Puya Confidential

	6	向上计数时计数器等于 CMPA
	7	EPWMx_TBCTR=0
向下计数模式	1	软件强制事件
	2	EPWMx_TBCTR=0
	3	向下计数时的 T1
	4	向下计数时的 T2
	5	向上计数时计数器等于 CMPB
	6	向上计数时计数器等于 CMPA
	7	EPWMx_TBCTR= EPWMx_TBPRD

表 18-11 中心对称模式的动作限定事件优先级

计数模式	**优先级**	**事件**	
		EPWMx_TBCTR 向上计数时的事，EPWMx_TBCTR=0 增至 EPWMx_TBCTR= EPWMx_TBPRD：	**EPWMx_TBCTR 向下计数时的事件， EPWMx_TBCTR= EPWMx_TBPRD 减至 EPWMx_TBCTR=1：**
中心对齐计数模式	1	软件强制事件	软件强制事件
	2	向上计数时的 T1	向下计数时的 T1
	3	向上计数时的 T2	向下计数时的 T2
	4	向上计数时计数器等于 CMPB	向下计数时计数器等于 CMPB
	5	向上计数时计数器等于 CMPA	向下计数时计数器等于 CMPA
	6	EPWMx_TBCTR=0	EPWMx_TBCTR= EPWMx_TBPRD

设置 CMPA/CMPB 大于周期时，事件将按如下表格所示发生：

表 18-12 CMPA/CMPB 大于周期时的动作

计数模式	**事件**	
	向上计数时比较事件	**向下计数时比较事件**
向上计数模式	当 CMPA/B≤EPWMx_TBPRD 时，在比较值匹配时产生 EPWMx_TBCTR= CMPA 或 EPWMx_TBCTR= CMPB 事件。当 CMPA/B> EPWMx_TBPRD 时，不会产生事件。	不会产生事件
向下计数模式	不会产生事件	当 CMPA/B< EPWMx_TBPRD 时，在比较值匹配时产生 EPWMx_TBCTR= CMPA 或 EPWMx_TBCTR= CMPB 事件。当 CMPA/B≥EPWMx_TBPRD 时，在周期匹配时产生 EPWMx_TBCTR= EPWMx_TBPRD 事件。
中心对齐计数模式	当 CMPA/B< EPWMx_TBPRD 且计数器向上计数时，在比较值匹配时产生 EPWMx_TBCTR= CMPA 或 EPWMx_TBCTR= CMPB 事件。当 CMPA/B≥EPWMx_TBPRD 时，在周期匹配时产生 EPWMx_TBCTR= EPWMx_TBPRD 事件。	当 CMPA/B< EPWMx_TBPRD 且计数器向下计数时，在比较值匹配时产生 EPWMx_TBCTR= CMPA 或 EPWMx_TBCTR= CMPB 事件。当 CMPA/B≥EPWMx_TBPRD 时，在周期匹配时产生 EPWMx_TBCTR= EPWMx_TBPRD 事件。

Puya confidential

18.3.4.5 影子模式

为了实现必须在周期结束时发生的动作限定模式更改(即使在相位变化时),动作限定也支持影子模式,并且支持在同步事件时发生影子寄存器到活动寄存器的加载。通过 SHDWAQAMODE 位和 SHDWAQBMODE 位来使能和禁止该寄存器的影子模式。影子模式和立即装载模式的行为描述如下:

(1) 影子模式

通过 SHDWAQAMODE 位和 SHDWAQBMODE 位来使能 EPWMx_AQCTLA 和 EPWMx_AQCTLB 寄存器的影子模式。默认禁止 EPWMx_AQCTLA 和 EPWMx_AQCTLB 寄存器的影子模式。

当使能影子寄存器时，影子寄存器的内容将在以下事件之一转移到活动寄存器，这些事件由 LDAQAMODE 位域、LDAQBMODE 位域、LDAQASYNC 位域和 LDAQBSYNC 位域决定:

- EPWMx_TBCTR=0
- EPWMx_TBCTR= EPWMx_TBPRD
- EPWMx_TBCTR=0 且 EPWMx_TBCTR= EPWMx_TBPRD
- 同步事件：由 DCxEVT1 或 DCBEVT1 或 EPWMx 同步输入或软件强制同步脉冲引起的
- LDAQAMODE/LDAQBMODE 选择的事件或同步事件

(2) 立即模式

如下图所示，当 SHDWAQAMODE=0 或 SHDWAQBMODE=0，即选择立即加载模式时，那么对寄存器的读或写将直接访问活动寄存器。

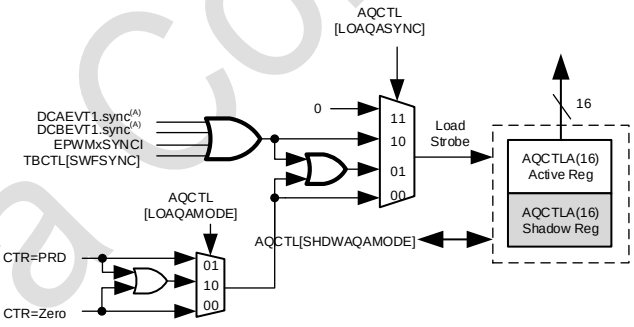


图 18-37 SHDWAQAMODE/SHDWAQBMODE 结构框图

注意:

(1) 由 EPWM DC 子模块基于 TRIPIN 输入的电平生成 DCxEVT1.sync 事件。

(2)当 CMPA 或 CMPB=0,且动作限定对 EPWMx_AQCTLA 和 EPWMx_AQCTLB 寄存器的动作与影子寄存器装载到活动寄存器同一时刻发生时，例如，当 CMPA=0 且 EPWMx_AQCTLA 在 EPWMx_TBCTR=0 时发生影子寄存器装载到活动寄存器时，两个事件就会产生冲突。因此，建议 EPWMx_AQCTLA 和 EPWMx_AQCTLB 寄存器在 EPWMx_TBCTR=0 边界使能影子模式时,使用非零的计数器比较值。

全局装载

Puya Confidential

可以通过在 EPWMx_GLDCFG 寄存器中配置相应位，也可以对 EPWMx_AQCTLA、EPWMx_AQCTLA2、EPWMx_AQCTLB、EPWMx_AQCTLB2 和 EPWMx_AQCSPRC 寄存器使用全局加载机制。如果使能全局装载，则相关寄存器的影子寄存器内容装载到活动寄存器同时发生在由 EPWMx_GLDCTL 寄存器定义的事件上。全局加载控制机制详情请参见全局装载章节。

18.3.5 死区发生器(DB)子模块功能描述

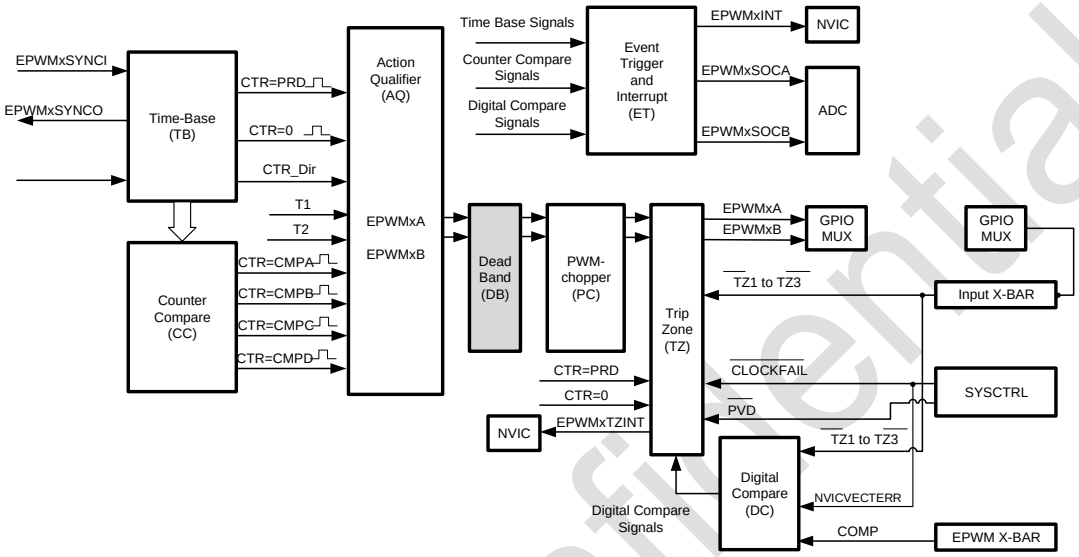


图 18-38 死区发生子模块

18.3.5.1 结构框图

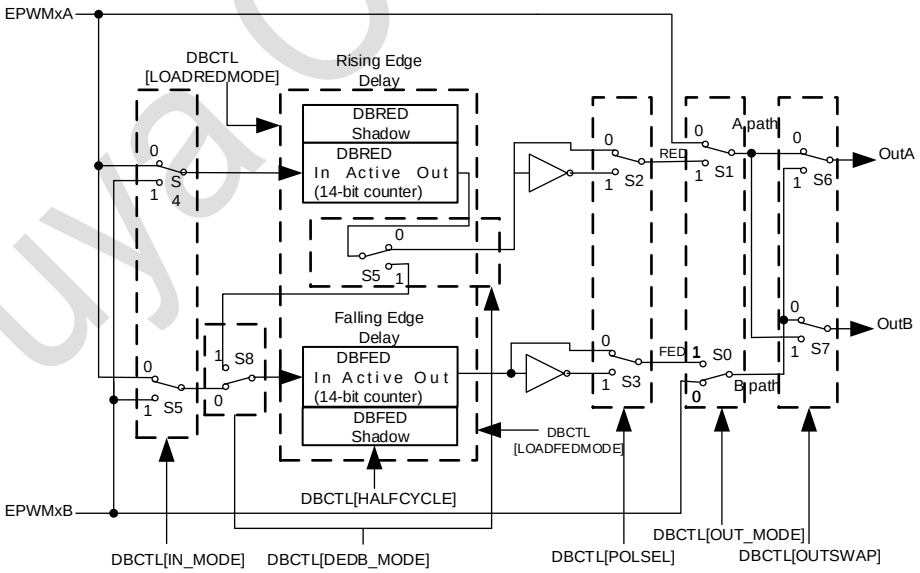


图 18-39 死区发生器子模块结构框图

注释：S0 和 S1 由 OUT_MODE 位控制，S2 和 S3 由 POLSEL 位控制，S4 和 S5 由 IN_MODE 位控制，S6 和 S7 由 OUTSWAP 位控制，S8 由 DEDB_MODE 位控制。

Puya Confidential

18.3.5.2 工作模式

- 通过 S4 和 S5 可以选择 RED 和 FED 的 PWM 输入源；S8 可以让上升沿延迟和下端延迟用于由 DBCTL[4]位（S4 开关）选择的输入源并输出到 B 信号路径；S2 和 S3 可以选择输出的极性；S1 和 S2 可以跳过 S4,S5,S8,S2,S3 以及上升、下降延迟逻辑；S6 和 S7 可以独立决定 OUTA 和 OUTB 的信号选择 A path 和 B path 输出。

死区的延时计数器为 14 位，通过 DBREDHR 和 DBFEDHR 寄存器使能高分辨率死区上升沿延迟和下降沿延迟，同时死区高分辨率寄存器和死区寄存器有影子寄存器。选择通过 S6、S7 和 S8，RED 和 FED 可以同时出现在 A 通道和 B 通道输出上。此外，RED 和 FED 可以一起应用于 A 通道或 B 通道输出，以实现 B 通道相对于 A 通道的相移。使用死区子模块附加操作模式实现 B 通道相对于 A 通道的相移，在选择 RED 和 FED 延迟相对于 EPWMxA 和 EPWMxB 输出的工作占空比方面存在局限性。

注意：

(1) 使能高分辨率死区时，将不能使用 PWM 斩波。

(2) 高分辨率死区 RED 和 FED 功能需要使能半周期时钟，即 HALFCYCLE=1。

(3) RED 和 FED 无法同时作用于 EPWMxA 和 EPWMxB。RED 和 FED 同时作用时，只作用于 PWM 输出 A（OutA）或 PWM 输出 B（OutB）。

(4) 使 B 通道产生相对于 A 通道的相移。通过使用 DEDB_MODE 位，根据 EPWMxA 派生得到 EPWMxB，并将 RED 和 FED 相应的相移量。如果 EPWMxA 的占空比小于相移量，则 EPWMxA 的下陷沿超前于 EPWMxB 的上升沿。建议确保所需的相移量小于输入死区模块的当前波形占空比。

影子模式

通过 SHDWDBRED 位和 SHDWDBFED 位来使能 EPWMx_DBREDHR 和 EPWMx_DBFED 寄存器的影子模式。默认禁止 DBRED 和 DBFED 寄存器的影子模式。

此外，可以将 EPWMx_DBCTL 寄存器影子化。通过 SHDWDBCTLMODE 位使能 EPWMx_DBCTL 寄存器的影子模式。

当使能影子寄存器时，影子寄存器的内容将在以下事件之一转移到活动寄存器，这些事件由 LOADREDMODE 位域、LOADFEDMODE 位域和 LOADDBCTLMODE 位域决定：

- EPWMx_TBCTR=0
- EPWMx_TBCTR= EPWMx_TBPRD
- EPWMx_TBCTR=0 或 EPWMx_TBCTR= EPWMx_TBPRD

注意：必须在对 DBRED 和 DBFED 寄存器的值进行编程之前使能影子模式。如果在这些寄存器进行编程后使能影子寄存器，则这些寄存器被加载为 0。

全局装载

可以通过在 EPWMx_GLDCFG 寄存器中配置相应位，也可以对 EPWMx_DBCTL、EPWMx_DBRED：EPWMx_DBREDHR、EPWMx_DBFED：EPWMx_DBFEDHR 寄存器使用全局加载机制。如果使能全

Puya Confidential

局装载功能，则对于所有使能此模式的寄存器，其影子寄存器内容装载到活动寄存器同时发生在由 EPWMx_GLDCTL 寄存器定义的事件上。

注意：

（1）当 DBRED 或 DBFED 活动寄存器在死区计数器正在计数时，加载了新的影子值，新的 DBRED 或 DBFED 不会影响当前边沿，只会影响下一个 PWM 边沿。

（2）当全局装载时刻设定为 EPWMx_TBCTR=0x00 时，死区值不应为 0。同理，当全局装载时刻为 EPWMx_TBCTR= EPWMx_TBPRD 时，死区值不应为周期值。

（3）EPWMx_TBPRDHR 寄存器不能与全局装载一起使用。当在应用中必须改变高分辨率周期时，用户必须在 EPWM 中断服务程序 (ISR) 写入单独的周期寄存器且该 ISR 必须与 PWM 切换周期同步，其中全局装载单次位也被写入。

18.3.5.3 操作要点

虽然死区发生器支持所有组合模式，但是经典死区工作模式只有七种。这些模式假设 IN_MODE 位域已配置 EPWMxA In 为 RED 和 FED 的输入源。通过改变输入源可以实现非传统或增强模式。经典死区工作模式可以分为以下几类：

（1）同时绕过 FED 和 RED：模式 1

允许用户在 PWM 信号路径上完全禁止 DB 子模块。

（2）经典死区极性设置：模式 2-5

这些典型极性配置可以解决现有工业电源开关门驱动所需的所有高电平有效和低电平有效模式，其时序图如 0%<占空比<100% 典型模式的死区时序图所示。需要注意，如果要生成 0%<占空比<100% 典型模式的死区时序图所示的等效波形，则需要配置 AQ 子模块以生成 EPWMxA 所示的信号。

（3）绕过 RED：模式 6

绕过 RED 组合详情请参见死区工作模式分类章节。

（4）绕过 FED：模式 7

绕过 FED 组合详情请参见死区工作模式分类章节。

死区工作模式分类

表 18-13 死区配置

模式		模式描述	配置位						
			OUT_MODE		POLSEL		OUTSWAP		DEDB_MODE
			S0	S1	S2	S3	S6	S7	S8
典型模式	1	EPWMxA 和 EPWMxB 直通，同时绕过 RED 和 FED，即无延迟.	0	0	X	X	-	-	-
	2	高电平有效互补（AHC）	1	1	0	1	-	-	-
	3	低电平有效互补（ALC）	1	1	1	0	-	-	-
	4	高电平有效（AH）	1	1	0	0	-	-	-
	5	低电平有效（AL）	1	1	1	1	-	-	-
	6	EPWMxA 输出 = EPWMxA 输入 EPWMxB 输出 = EPWMxA 输入带 FED	1	0	0/1	0/1	-	-	-

Via Confidential

	7	EPWMxA 输出 = EPWMxA 输入带 RED EPWMxB 输出 = EPWMxB 输入	0	1	0/1	0/1	-	-	-
附加模式		EPWMxA 和 EPWMxB 信号由 OUT_MODE 位域定义。	-	-	-	-	0	0	0
		EPWMxA = 由 OUT_MODE 位域所定义的 A 路径。 EPWMxB = 由 OUT_MODE 位域所定义的上升沿延迟或绕过延迟的 A 路径。	-	-	-	-	0	1	0
		EPWMxA = 由 OUT_MODE 位域所定义的下降沿延迟或绕过延迟的 B 路径。 EPWMxB = 由 OUT_MODE 位域所定义的 B 路径。	-	-	-	-	1	0	0
		EPWMxA = 由 OUT_MODE 位域所定义的下降沿延迟或绕过延迟的 B 路径。 EPWMxB = 由 OUT_MODE 位域所定义的上升沿延迟或绕过延迟的 A 路径。	-	-	-	-	1	1	0
		RED 仅作用于 A 路径上 S4 开关（IN_MODE[4] 位）所选的 EPWMxA 或 EPWMxB。 FRD 仅作用于 B 路径上 S5 开关（IN_MODE[5] 位）所选的 EPWMxA 或 EPWMxB。	-	-	-	-	X	X	0
		RED 和 FRD 作用于 S4 开关（IN_MODE[4]位）所选的信号源，且仅输出到 B 信号路径。 （1）	-	-	-	-	X	X	1

注意：（1）如果该位置位，用户则可以设置 OUT_MODE=01（使 A 路径等于 InA）或者设置 OUTSWAP=1x（使 EPWMxA 输出信号等于 B 路径），否则 EPWMxA 无信号。

上升沿延迟与下降沿延迟

DB 子模块支持独立的 RED 和 FED 值。通过 DBRED 和 DBFED 寄存器配置延迟的量，它们是 14 位寄存器，其值表示信号边沿延迟的 TBCLK 周期数。

Puya Confidential

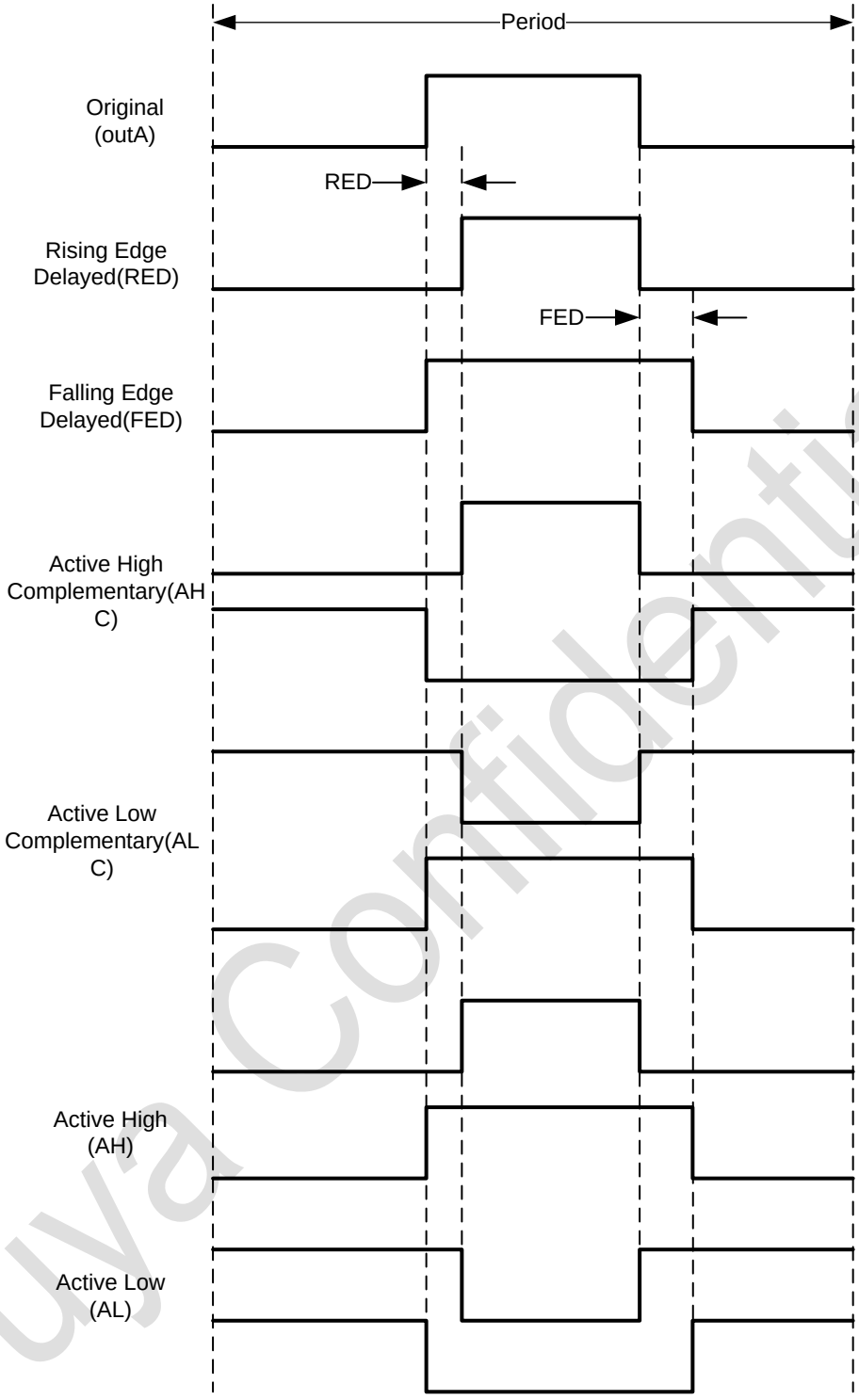


图 18-40 典型模式的死区时序图（0%<占空比<100%）

■ 开启全周期时钟

上升沿延迟计算公式为：RED=DBRED*T_{TBCLK}

下降沿延迟计算公式为：FED=DBFED*T_{TBCLK}

■ 开启半周期时钟

上升沿延迟计算公式为：RED=DBRED*（T_{TBCLK}/2）

下降沿延迟计算公式为：FED=DBFED*（T_{TBCLK}/2）

其中 T_{TBCLK} 是 EPWMCLK 预分频后的 TBCLK 的周期。

Puya Confidential

各种 TBCLK 选项的延迟值如下表格所示，其 EPWM 输入时钟频率为 100 MHz。

表 18-14 各种 TBCLK 选项的延迟值

延迟值（DBRED 和 DBFED）	TBCLK			单位
	EPWMCLK 4 分频	EPWMCLK 2 分频	EPWMCLK 不分频	
1	0.04	0.02	0.01	μs
5	0.2	0.1	0.05	
10	0.4	0.2	0.1	
100	4	2	1	
200	8	4	2	
400	16	8	4	
500	20	10	5	
600	24	12	6	
700	28	14	7	
800	32	16	8	
900	36	18	9	
1000	40	20	10	

18.3.6 PWM 斩波（PC）子模块功能描述

PWM 斩波器子模块允许高频载波信号对 AQ 和 DB 生成的 PWM 波形进行调制，该功能可用于需要基于脉冲变压器的门极驱动来控制功率开关元件。

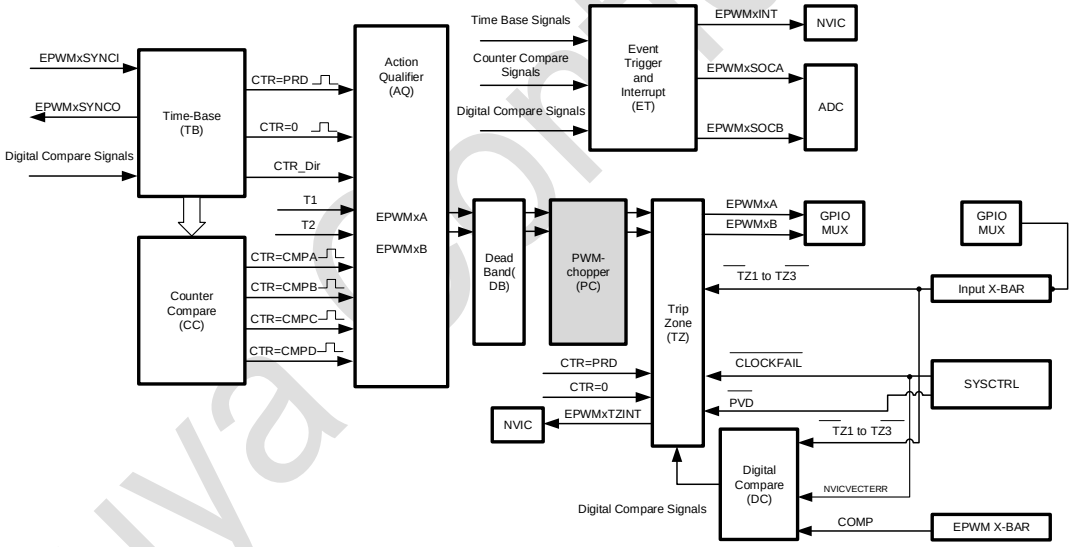


图 18-41 PWM 斩波子模块

Puya Confidential

18.3.6.1 结构框图

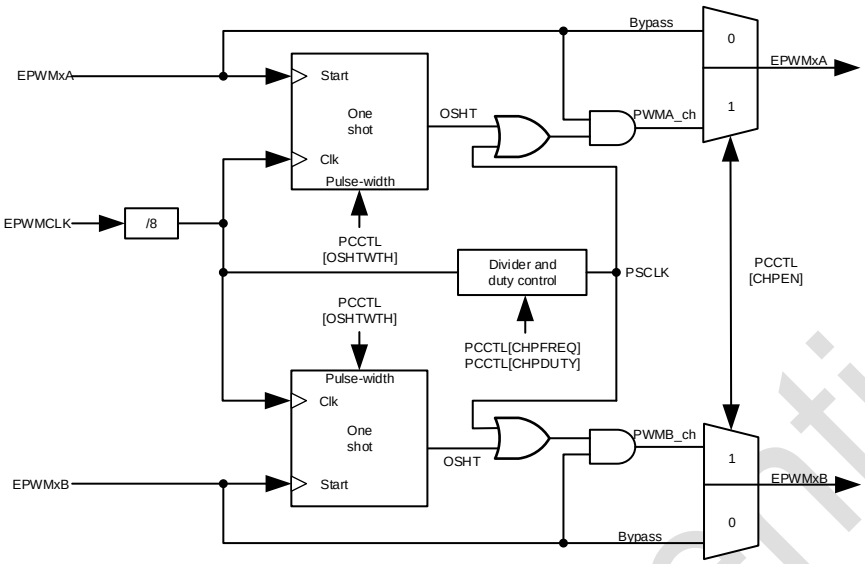


图 18-42 斩波子模块结构框图

18.3.6.2 操作要点

图 18-42 展示了 PWM 斩波子模块的细节。载波时钟由 EPWMCLK 分频产生，其频率与占空比分别通过 EPWMx_PCCTL 寄存器中的 CHPFREQ 与 CHPDUTY 位控制。单脉冲触发单元可生成高能量首脉冲以确保功率开关快速可靠导通，后续维持脉冲则保障功率开关保持导通状态。单脉冲宽度由 OSHTWTH 位配置。通过 CHPEN 位可使能或完全旁路（禁用）PWM 斩波子模块。

18.3.6.3 波形

如下图所示仅有斩波动作的简化波形，其中未显示占空比和单次脉冲控制。

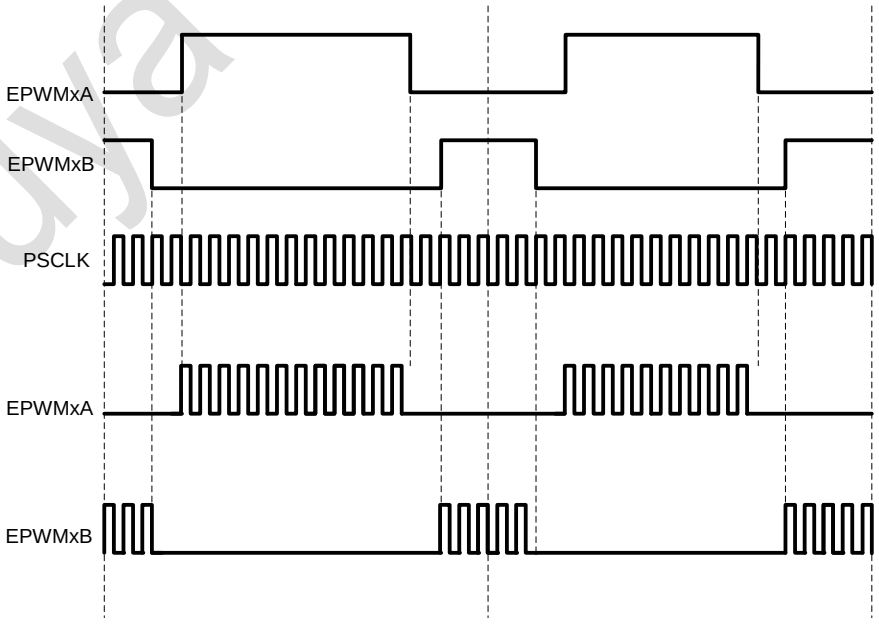


图 18-43 简单 PC 子模块时序图（仅有斩波动作）

Puya Confidential

单次脉冲

可以将首个脉冲的宽度编程为任何一种脉冲宽度值（共 16 种）。$T_{EPWMCLK}$ 是系统时钟的周期，OSHTWTH 长度为 4 位，其值为 1 到 16，所以首个脉冲宽度或周期计算公式为：$T_{1stpulse} = 8 * T_{EPWMCLK} * OSHTWTH$

如下所示首个脉冲和后续持续脉冲的 PWM 时序图以及 EPWMCLK 为 100MHz 时的脉宽值。

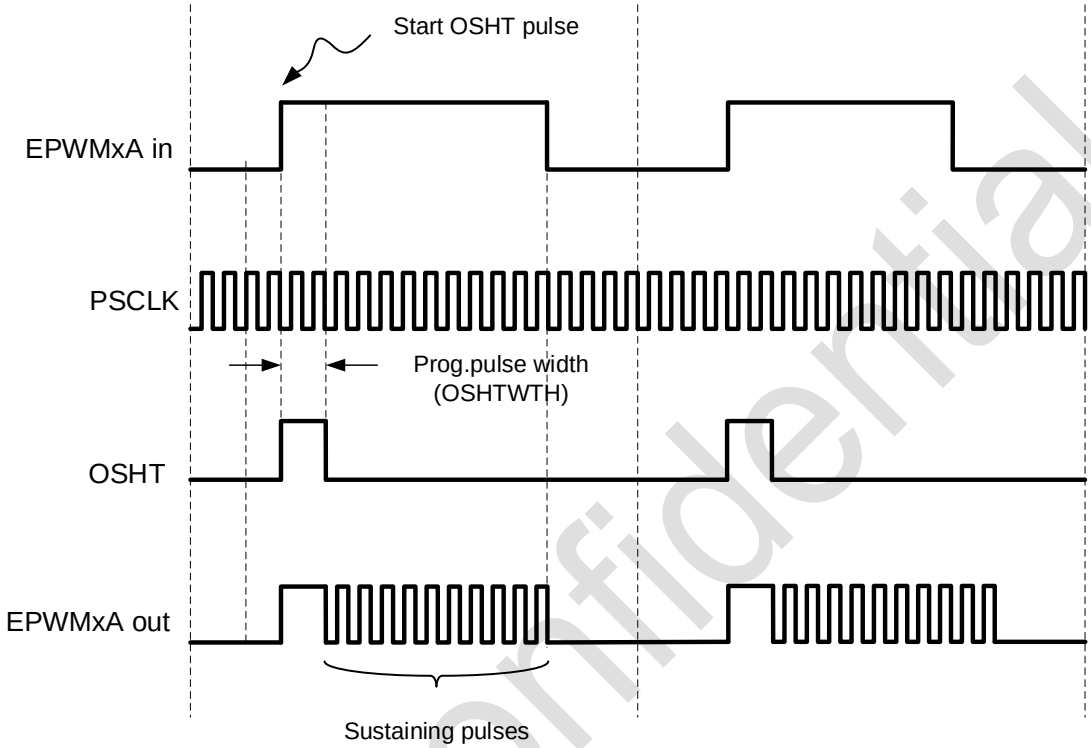


图 18-44 首个脉冲和后续持续脉冲的 PWM 时序图

表 18-15 EPWMCLK 为 100 MHz 时的脉宽值

EPWMCLK	OSHTWTH	脉宽值
100MHz	0x00	80 ns
	0x01	160 ns
	0x02	240 ns
	0x03	320 ns
	0x04	400 ns
	0x05	480 ns
	0x06	560 ns
	0x07	640 ns
	0x08	720 ns
	0x09	800 ns
	0x0a	880 ns
	0x0b	960 ns
	0x0c	1040 ns
	0x0d	1120 ns
	0x0e	1200 ns
	0x0f	1280 ns

占空比控制

当设计基于脉冲变压器的门极驱动时，需要了解包括饱和度在内的变压器和相关电路的磁性能或特性。可编程第二个及以后续脉冲的占空比以辅助门极驱设计人员。这些持续脉冲确保在开启期间维持功率开关门极驱动强度和极性，可以使用软件控制来调整或优化设计可编程的占空比。

Puya Confidential

通过 CHPDUTY 位域配置任何一种占空比控制（共 7 种），其占空比范围为 12.5%~87.5%。

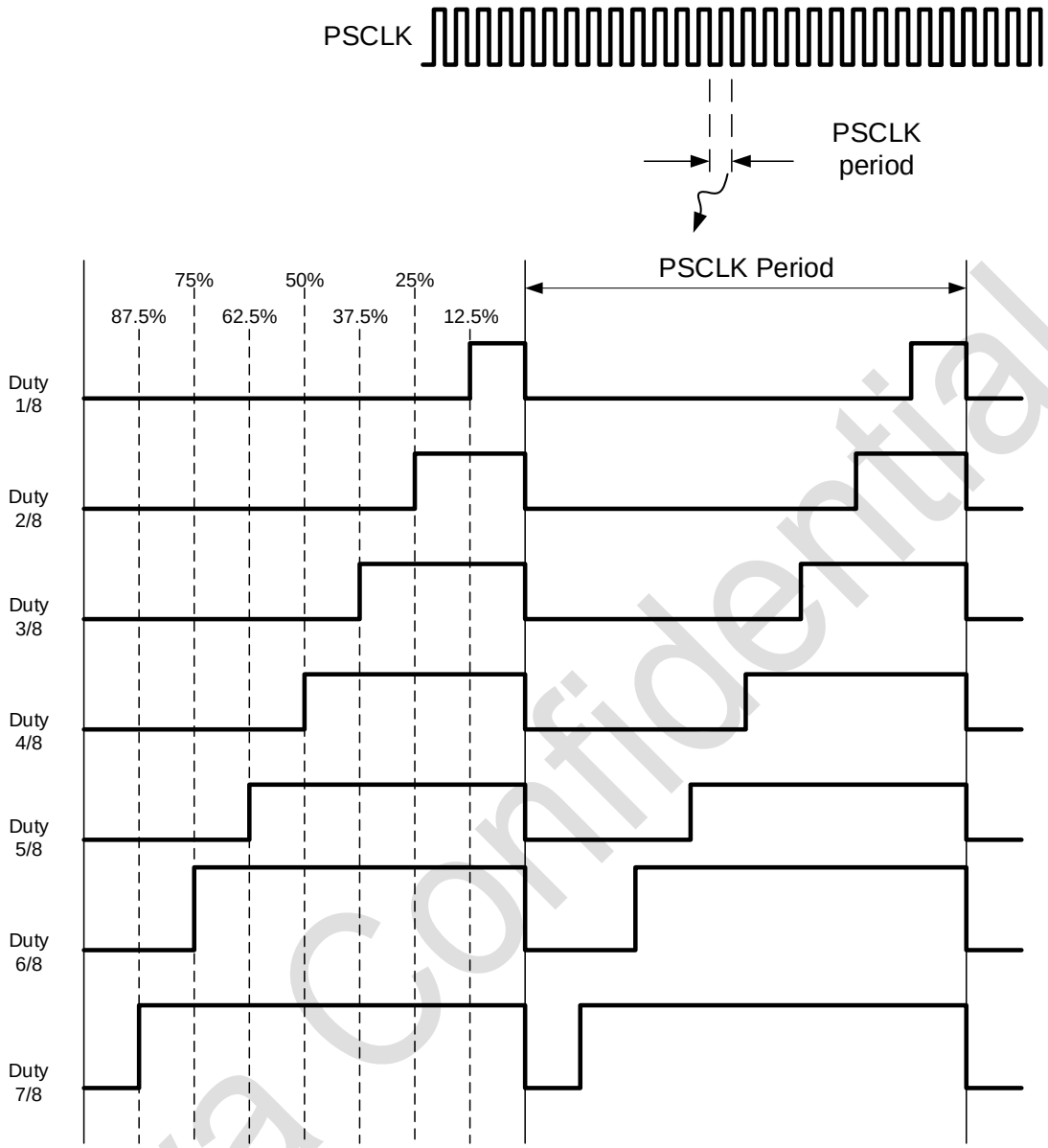


图 18-45 占空比控制的 PWM 时序图

18.3.7 跳闸区（TZ）子模块功能描述

每个 EPWM 模块均连接六路 TZn 信号（TZ1n 至 TZ6n）。其中 TZ1n 至 TZ3n 源自 GPIO 复用器，TZ4n 连接电压检测逻辑（PVD），TZ5n 连接至系统时钟故障检测逻辑，TZ6n 则源自 CPU 的 HALT 输出，在 DBG 模块的 DBG_EPWM_EN 位可选择 CPU HALT 时 EPWM 的工作状态，具体可参考 DGB 模块。这些信号用于指示外部故障或跳闸条件，EPWM 输出可配置为在故障发生时执行相应的保护动作。

Puya Confidential

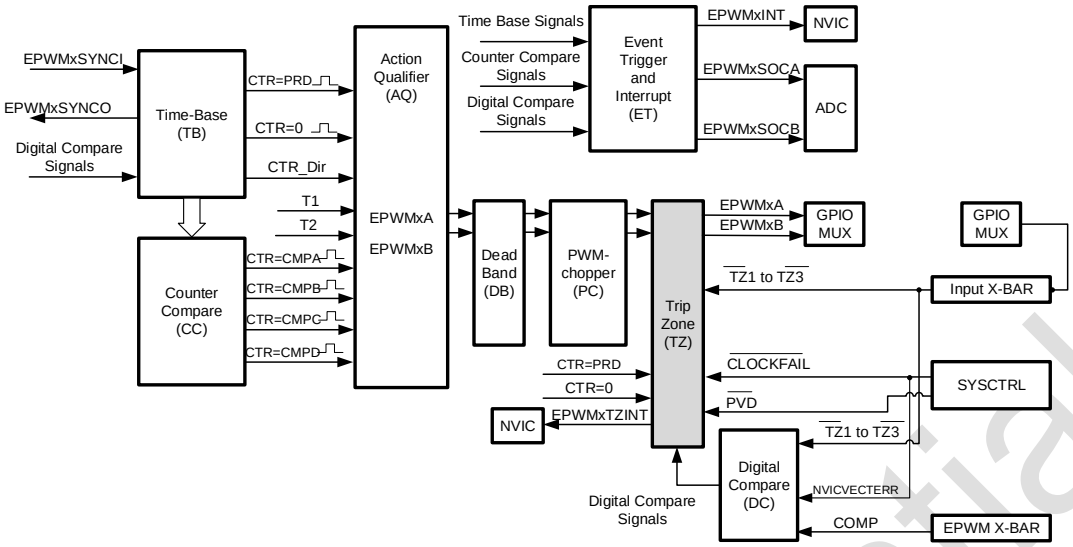


图 18-46 错误区子模块

18.3.7.1 控制逻辑

下图展示了 TZ 子模块的控制逻辑结构框图，DCxEVTy 信号详情请参见 DC 子模块章节。

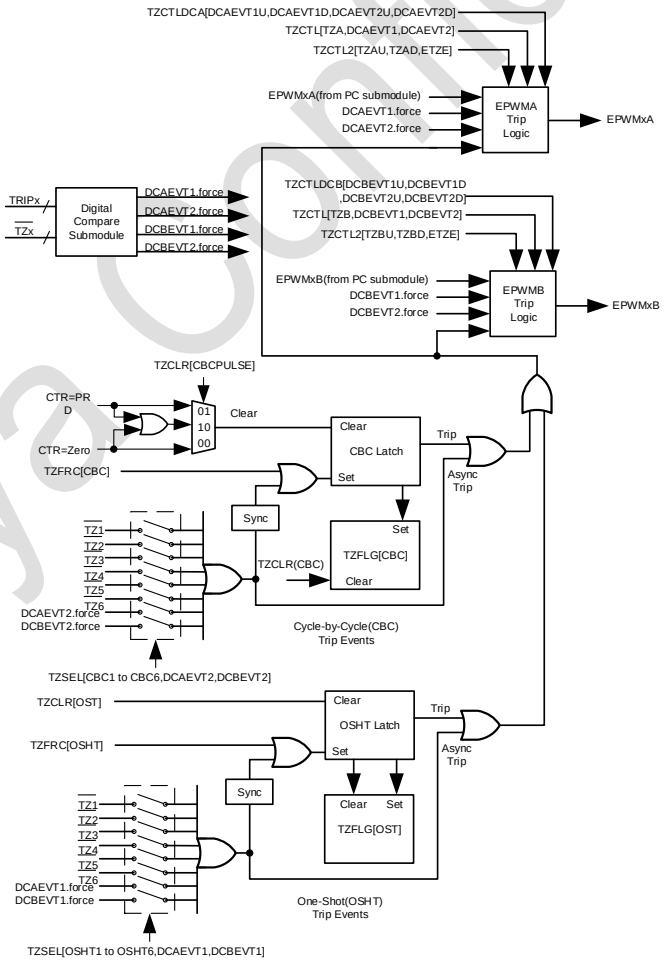


图 18-47 控制逻辑结构框图

Puya Confidential

18.3.7.2 中断

通过 CBC、OST 和 DCxEVTy 标志位来检测 EPWMx 跳闸区中断的来源。在使用多个中断源生成 EPWMx 跳闸区中断时，可以读取和清除这些标志来识别和处理引发中断的具体事件。

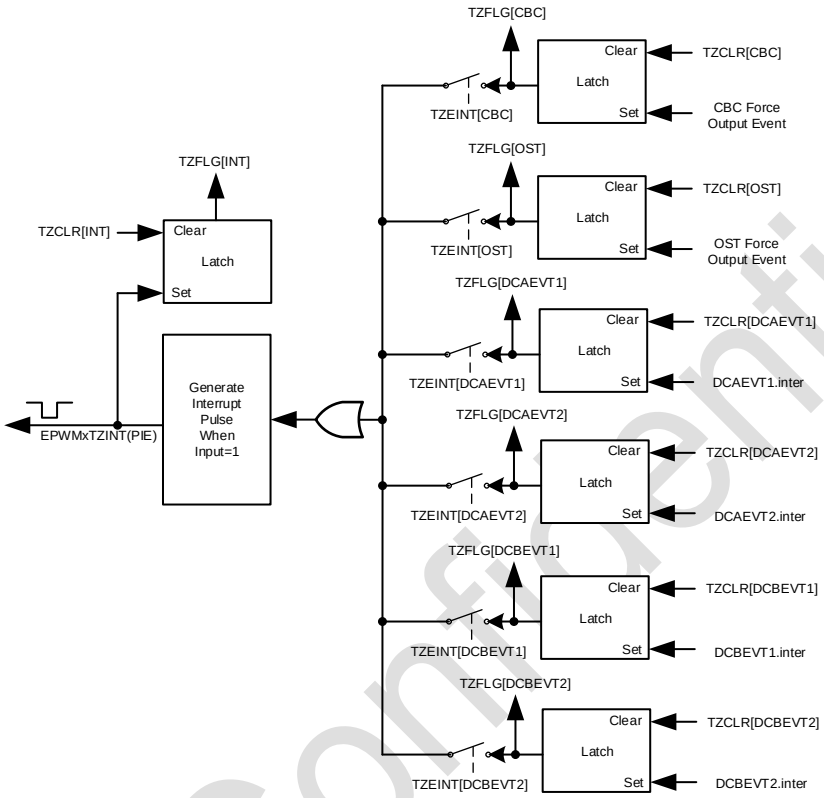


图 18-48 中断逻辑结构框图

18.3.7.3 操作要点

TZ1n 至 TZ6n 信号 (TZxn 信号) 为低电平有效输入信号。发生以下情况表明发生了跳闸事件： TZxn 信号中任何一个信号变低时，或者基于 EPWMx_TZDCSEL 寄存器选择的 DCxEVTy 强制事件发生时。每个 EPWM 模块可以单独配置为使用或忽略每个跳闸区信号或数字比较事件。由 EPWMx_TZSEL 寄存器决定使用的跳闸区信号或数字比较事件。TZxn 信号可以选择是否与 EPWMCLK 同步，并在 INPUTX-BAR Filter 模块内进行数字滤波。为了触发 EPWM 模块的故障条件，需要在 TZxn 输入最小 3*TBCLK 的低电平脉冲宽度。当脉冲宽度小于此值时，OST 或 CBC 锁存器无法锁存跳闸条件。异步跳闸能够保证即使由于任何原因时钟缺失，仍然可以确保 TZxn 输入上存在的有效事件跳闸，GPIOs 或外设必须正确配置。

每个 TZxn 输入可以单独配置为 EPWM 模块提供逐周期或一次性错误事件。DCAEVT1 和 DCBEVT1 事件可以配置为直接错误关断 EPWM 模块或向模块提供一次性错误事件。同样，DCAEVT2 和 DCBEVT2 事件也可以配置为直接错误关断 EPWM 模块或向模块提供逐周期错误事件。此配置分别由 DCAEVT1/2、DCBEVT1/2、CBCn 和 OSHTn 控制位 (其中 n 对应于错误输入) 决定。

Puya Confidential

单次跳闸（OST）事件

当发生一个 OST 事件时，立即对 EPWMxA 和 EPWMxB 输出执行 TZA 和 TZB 位域中指定的操作，详情请参见跳闸事件发生时采取的动作表格。当计数器向上计数或向下计数时，通过配置 EPWMx_TZCTL2 寄存器相应位域指定发生跳闸事件时的独立动作，这些动作仅在 ETZE=1 的情况下才会生效。

如果发生 OST 事件，则错误标志 EPWMx_TZFLG[OST] 位置位，并在 EPWMx_TZEINT 寄存器和中断控制中使能了中断时，还会生成 EPWMx_TZINT 中断。在 EPWMx_TZOSTFLG 寄存器中，引起 OST 的那个事件对应标志也被置位。可以通过手动写入 EPWMx_TZOSTCLR 寄存器中的对应位，从而清零 EPWMx_TZOSTFLG 寄存器位（如果需要）。

当使能 OST 中断（OST=1）且通过 EPWMx_TZSEL 寄存器选择单次跳闸源为 DCxEVT1 时，无需使能 DCxEVT1 中断（DCxEVT1 /DCxEVT1=1），因为数字比较事件通过 OST 机制触发中断。

注意：在确认 OST 跳闸源变为非活动状态后，应清零 EPWMx_TZFLG 和 EPWMx_TZOSTFLG 寄存器标志。否则，当使能中断时，可能发生 OST 中断而 OST 标志为 0，这取决于何时清除标志。

当跳闸事件发生时采取的动作，可以通过 EPWMx_TZCTL、EPWMx_TZCTL2、EPWMx_TZCTLDCA 和 EPWMx_TZCTLDCE 寄存器分别独立地配置每个 EPWM 输出引脚。可以采取的动作如下所示：

表 18-16 跳闸事件发生时采取的动作

EPWMx_TZCTL 寄存器位域设置	EPWMxA 和/或 EPWMxB 动作	状态
00	高阻态	已跳闸
01	强制输出为高电平	已跳闸
10	强制输出为低电平	已跳闸
11	输出保持原状态	无操作

逐周期跳闸（CBC）事件

当发生一个 CBC 事件时，立即对 EPWMxA 和 EPWMxB 输出执行 TZA 和 TZB 位域中指定的操作，详情请参见跳闸事件发生时采取的动作表格。当计数器向上计数或向下计数时，通过配置 EPWMx_TZCTL2 寄存器相应位域指定发生跳闸事件时的独立动作，这些动作仅在 ETZE=1 的情况下才会生效。

如果发生 CBC 事件，则 EPWMx_TZFLG[CBC] 位置位，并在 EPWMx_TZEINT 寄存器和 中断控制外设中使能了中断时，还会生成 EPWMx 跳闸区中断。在 EPWMx_TZCBCFLG 寄存器中，引起 CBC 事件的那个事件对应标志也被置位。

当使能 CBC 中断（EPWMx_TZEINT[CBC]=1）且通过 EPWMx_TZSEL 寄存器选择逐周期跳闸源为 DCXEVT2 时，无需使能 DCXEVT2 中断，因为数字比较事件通过 CBC 机制触发中断。

Puya Confidential

当不存在跳闸事件时，将会根据 CBCPULSE 寄存器的选择自动清除 CBC 跳闸锁存的状态。因此，在这种模式下，每个 EPWM 周期都复位或清除跳闸事件。在通过手动写入 EPWMx_TZCBCCLR 寄存器标志位和 EPWMx_TZCLR[CBC] 位清零之前，EPWMx_TZCBCFLG 寄存器标志位和 EPWMx_TZCLR[CBC] 位会保持置位。如果清除 EPWMx_TZCBCFLG 寄存器标志位和 EPWMx_TZCLR[CBC] 位后 CBC 事件仍然存在跳闸事件，则立即再次置位这些标志位。

数字比较事件（DCxEVTy）

基于 EPWMx_TZDCSEL 寄存器选择的 DCxH 和 DCxL 信号组合，可以产生数字比较事件 DCxEVTy。通过 EPWMx_DCTRIPSEL 寄存器选择 DCxH 和 DCxL 信号的源信号为跳闸区输入引脚，COMPx 或其他信号。DC 子模块信号详情请参见数字比较子模块功能描述章节。

当发生 DCxEVTy 时，立即对 EPWMxA 和 EPWMxB 输出执行 DCxEVTy 位域中指定的操作，详情请参见跳闸事件发生时采取的动作表格。当计数器向上计数或向下计数时，通过配置 EPWMx_TZCTLDCA 和 EPWMx_TZCTLDCB 寄存器相应位域指定发生跳闸事件时的独立动作，这些动作仅在 ETZE=1 的情况下才会生效。

如果发生数字比较跳闸事件，则 EPWMx_TZFLG[DCxEVTy] 位置位，并在 EPWMx_TZEINT 寄存器和 中断控制外设中使能了中断时，还会生成 EPWMx 跳闸区中断。

当不存在数字比较跳闸事件时，引脚上的指定状态将被自动清除。在通过手动写入 EPWMx_TZCLR[DCxEVTy] 位清零之前，EPWMx_TZFLG[DCXEVTy] 位和 EPWMx_TZFLG[DCBEVTy] 位会保持置位。如果清除 EPWMx_TZFLG[DCXEVTy] 位和 EPWMx_TZFLG[DCBEVTy] 位后 CBC 事件仍然存在跳闸事件，则立即再次置位这些标志位。

跳闸区配置

表 18-17 跳闸区配置选项

条件	EPWM1 寄存器配置	EPWM2 寄存器配置
TZ1n 上的 OST 事件将 EPWM1A 和 EPWM1B 输出都拉低，并强制 EPWM2A 和 EPWM2B 输出为高电平	（1） OSHT1=1：选择 TZ1n 作为 EPWM1 单次跳闸源 （2） TZA=10：跳闸事件发生时，强制 EPWM1A 输出为低电平 （3） TZB=10：跳闸事件发生时，强制 EPWM1B 输出为低电平	（1） OSHT1=1：选择 TZ1n 作为 EPWM2 单次跳闸源 （2） TZA=01：跳闸事件发生时，强制 EPWM2A 输出为高电平 （3） TZB=01：跳闸事件发生时，强制 EPWM2B 输出为高电平
（1） TZ5n 上的 CBC 事件将 EPWM1A 和 EPWM1B 输出都拉低 （2） TZ1n 或 TZ6n 上的 OST 事件将 EPWM2A 置为高阻态	（1） CBC5 =1：选择 TZ5n 作为 EPWM1 逐周期跳闸源 （2） TZA=10：跳闸事件发生时，强制 EPWM1A 输出为低电平 （3） TZB=10：跳闸事件发生时，强制 EPWM1B 输出为低电平	（1） OSHT1=1：选择 TZ1n 作为 EPWM2 单次跳闸源 （2） OSHT6=1：选择 TZ6n 作为 EPWM2 单次跳闸源 （3） TZA=00：跳闸事件发生时，EPWM2A 被置为高阻态 （4） TZB=11：跳闸事件发生时，EPWM2B 保持原状态，即忽略跳闸事件

Puya Confidential

当配置 EPWM X-BAR/INPUT X-BAR 和 GPIO 选项时，需要注意更改 X-BAR 输入选择可能会引起意外事件发送。建议将 INPUT X-BAR 和 GPIO 配置在使能 EPWM 跳闸区之前设置完成。当必须要在使能 EPWM 跳闸区后更改 INPUT X-BAR/GPIO 配置时，用户可以关闭跳闸（通过清零 EPWMx_TZSEL 寄存器），并在 INPUT X-BAR 选择更改后重新配置 EPWMx_TZSEL 寄存器。

18.3.8 事件触发（ET）子模块功能描述

事件触发子模块负责管理由时基子模块、计数器比较子模块和数字比较子模块产生的事件，当选定事件发生时，它将会产生一个中断信号给 CPU，并向 ADC 发送一个转换开始脉冲。

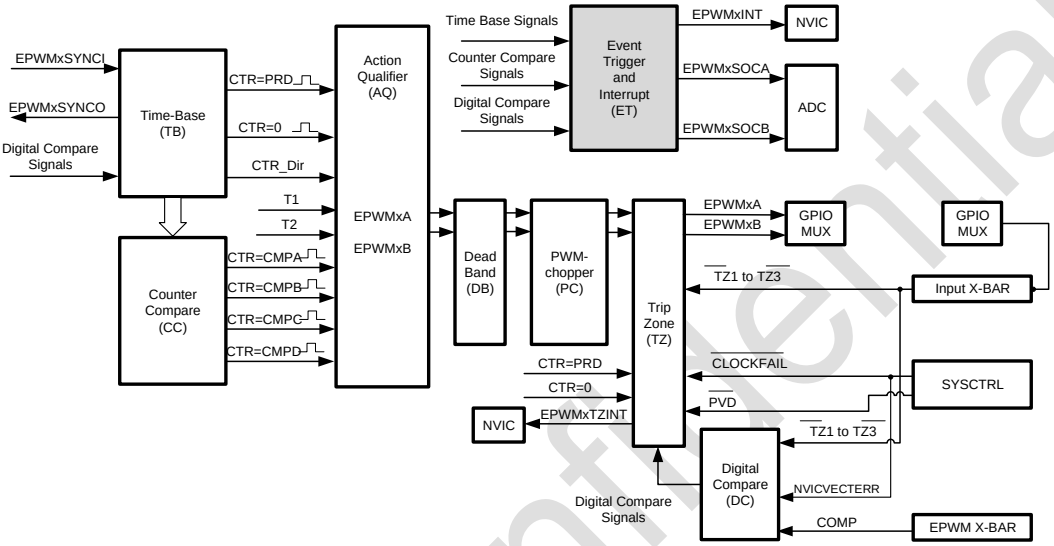


图 18-49 事件触发子模块

18.3.8.1 结构框图

如下图所示，ET 子模块可以监视各种事件条件，根据预分频设置决定在每次事件、每两次事件直到每十五次事件发出中断请求和 ADCSOC。

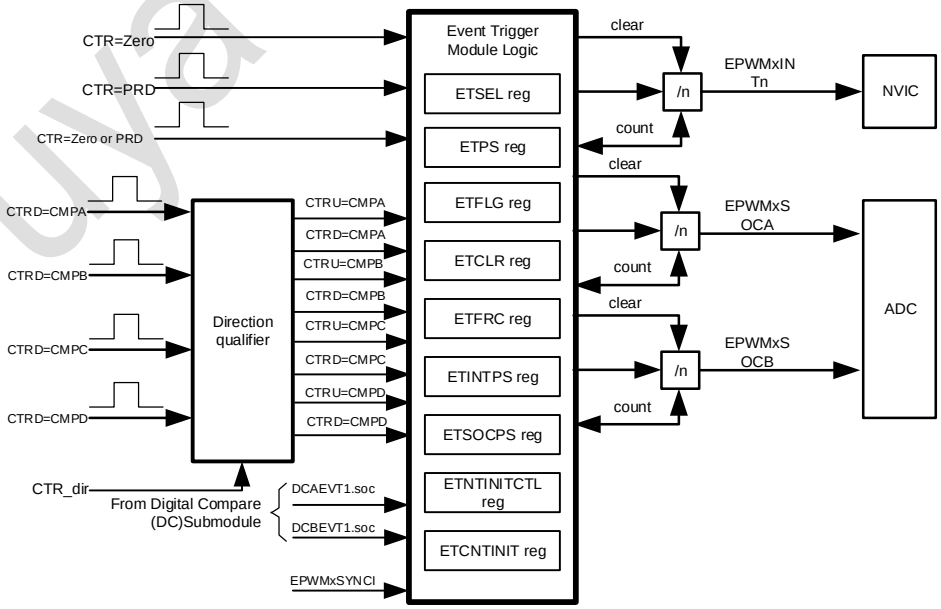


图 18-50 事件触发子模块结构框图

Puya Confidential

表 18-18 ET 子模块补充说明

寄存器	描述
EPWMx_ETSEL	选择触发中断或 ADC 转换开始的事件
EPWMx_ETPS	配置上面提到的事件分频选项
EPWMx_ETFLG	该寄存器表示所选事件和预缩放事件的状态
EPWMx_ETCLR	该寄存器用于以软件方式清零 ETFLG 寄存器中的标志位
EPWMx_ETFRC	这些位用于软件强制产生事件，用于调试或软件干预
EPWMx_ETINTPS	配置中断事件预分频选项，支持计数多达 15 个事件的周期
EPWMx_ETSOCPS	配置 SOC 事件预分频选项，支持计数多达 15 个事件的周期
EPWMx_ETCNTINITCTL	该寄存器使能使用同步事件或使用软件强制使能 ETCNTINIT 寄存器初始化
EPWMx_ETCNTINIT	在同步事件或软件强制时，该寄存器允许用户使用编程值对中断、SOCA 或 SOCB 计数器进行初始化

18.3.8.2 事件触发的脉冲发生器

SOCA 脉冲触发器

事件触发 SOCA 脉冲触发器的操作如下图所示，其工作方式如下所示：

- 增强功能：包括 EPWMx_ETSELZ 中定义的 SOCASELCMP 和 SOCBSELCMP 位分别使能 CMPC 和 CMPD 事件，从而引起转换的启动。
- SOCPSSEL 位可以决定 SOCACNT2 位和 SOCAPRD2 位是否接管控制权
- SOCA 事件计数器和 SOCA 周期的行为类似于中断发生器，会连续产生脉冲。也就是说，在产生脉冲时，锁存 EPWMx_ETFLG[SOCA] 位，但中断发生器不会停止后续脉冲产生
- SOCAEN 位使能为 0 时禁止产生脉冲时，仍然计数输入事件，直至达到周期值。
- 通过 SOCASEL 和 SOCBSEL 位域可以单独配置触发 SOCA 和 SOCB 脉冲的事件，其可以使用事件如下：
- 与中断产生逻辑相关的事件相同
- 来自 DC 子模块的 DCAEVT1.soc 和 DCBEVT1.soc 事件信号
- SOCA 事件计数器 2 初始化方案类似于中断生成器，都包括相应的使能、值初始化以及同步事件或软件强制选项

Puya Confidential

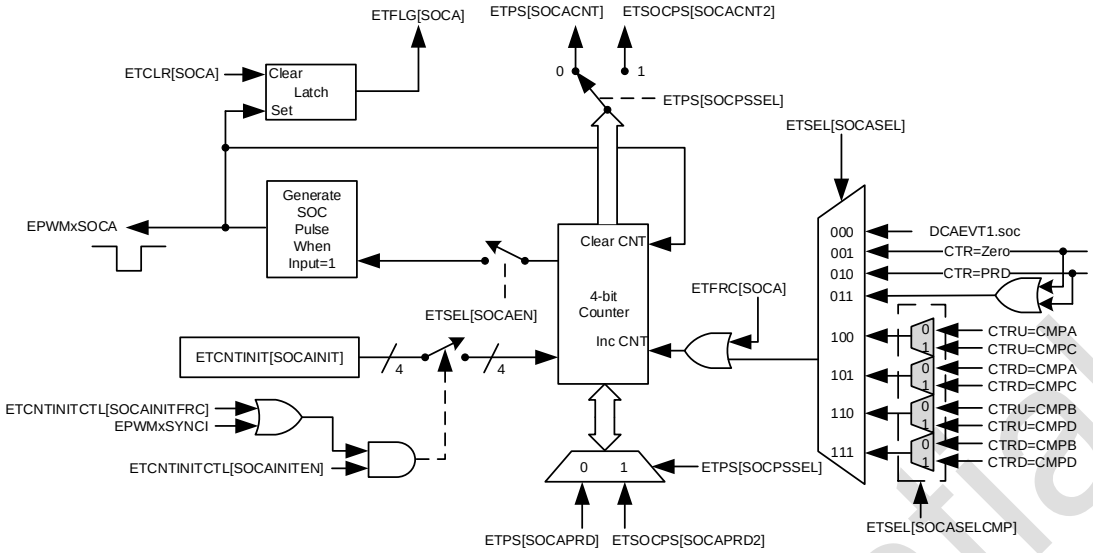


图 18-51 SOCA 脉冲触发器结构框图

SOCB 脉冲触发器

事件触发 SOCB 脉冲触发器的操作如下图所示，其工作方式详情请参见 SOCA 脉冲触发器小节。

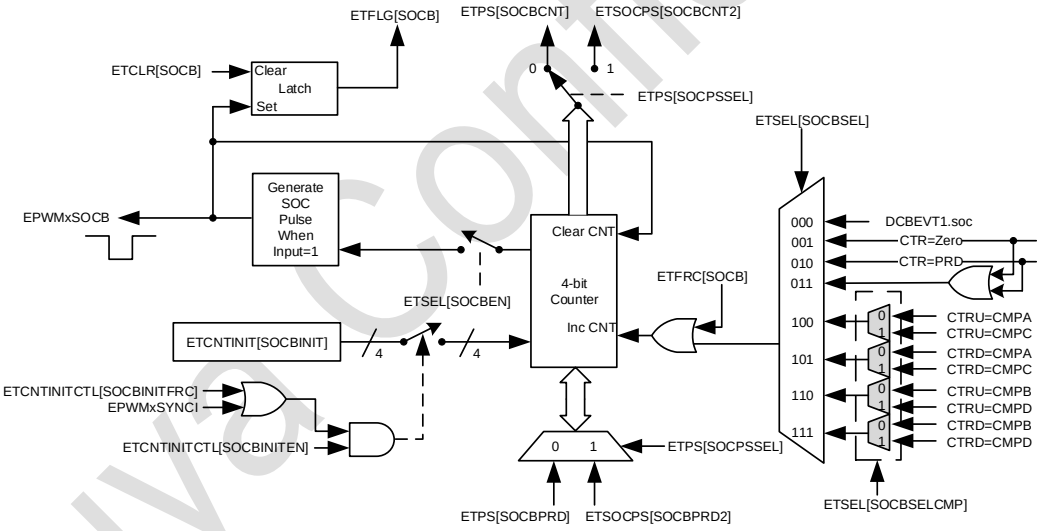


图 18-52 SOCB 脉冲触发器结构框图

18.3.8.3 中断

事件触发的中断生成逻辑如下图所示，通过 INTPRD 位域选择产生中断脉冲所需的事件数为：不产生中断和每次/每两次/每三次事件产生中断。

INTPSSEL 位决定是否使用 ETINTPS 寄存器的 INTCNT2 位和 INTPRD2 位来确定事件频率为每 0...15 个事件产生一次中断。通过 INTSEL 位域和 INTSELCMP 位配置可以导致中断的事件如下表格所示。

Puya Confidential

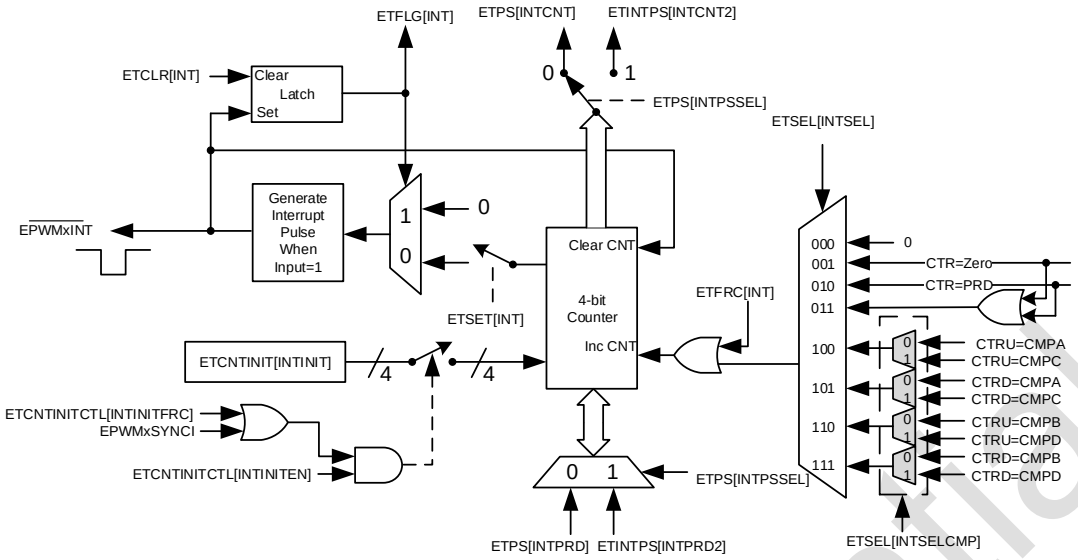


图 18-53 中断发生器结构框图

表 18-19 中断发生器补充说明

事件	描述
TBCTR=0	时基计数器等于 0
TBCTR= TBPRD	时基计数器等于时基周期
TBCTR=0 或 TBCTR= TBPRD	0 或等于时基周期
TBCTRU= CMPA	计数器向上计数时，时基计数器等于 CMPA
TBCTRD= CMPA	计数器向下计数时，时基计数器等于 CMPA
TBCTRU= CMPB	计数器向上计数时，时基计数器等于 CMPB
TBCTRD= CMPB	计数器向下计数时，时基计数器等于 CMPB
TBCTRU= CMPC	计数器向上计数时，时基计数器等于 CMPC
TBCTRD= CMPC	计数器向下计数时，时基计数器等于 CMPC
TBCTRU= CMPD	计数器向上计数时，时基计数器等于 CMPD
TBCTRD= CMPD	计数器向下计数时，时基计数器等于 CMPD

基于 INTPSSEL 位所做的选择，已经发生的事件数可以通过 INTCNT 位域或 INTCNT2 位域来读取。即当指定的事件发生时，基于 INTPSSEL 位所做的选择，INTCNT 位域或 INTCNT2 位域会不断增加直到达到 INTPRD 或 INTPRD2 指定的值。如果 INTCNT=INTPRD，则计数器停止计数并产生计数器对应的输出（中断）。计数器只有在中断发送到中断控制器时才会清零。

当 INTCNT 达到 INTPRD 或 INTCNT2 达到 INTPRD2 时，计数器会发生的行为如表格所示。

表 18-20 EPWMx_ETSEL[INTEN]和 EPWMx_ETFLG[INT]对计数器的行为

条件	行为
INTEN=0 或 INT=1	当计数器到达周期值（INTCNT=INTPRD）时，计数器停止对事件的计数。
INTEN=1 且 INT=0	计数器（INTCNT）计满后，产生一个中断脉冲且 INT=1， 计数器清零。事件计数器再次开始计数事件。
INTEN=1 且 INT=1	计数器将保持输出高电平，直到清除 INT 位。这允许在一个中断被处理时，下一个中断处于等待状态。

以下方式适用于 INTCNT/INTCNT2 和 INTPRD/INTPRD2：

Puya Confidential

- 向 INTPRD 位域写入 0 会自动清零计数器且重置计数器输出，因此不会产生中断
- 所有写入 INTPRD 位域的操作会使 INTCNT 保留原来的值
- INTCNT 在溢出时复位
- ETFRC[INT]=1 会使事件计数器增加
- 当计数器到达周期值时，计数器的行为与表格描述的相同
- 如果 INTPRD=0，则禁用计数器且忽略 ETFRC[INT] 位，因此不会检测到事件

当使用 INTCNT 和 INTPRD 时，每次/每两次/每三次事件可以产生一次中断。当使用 INTCNT2 和 INTPRD2 时，每次事件至每 15 次事件（最多）可以产生一次中断。

可以基于 INTINITEN 位的选择，通过 INTINIT 位域中的值对 INTCNT2 进行初始化。当 INTINITEN 置位时，在发生同步事件或 INTINITFRC 位域决定的软件强制事件时，使用 INTINIT 位域的内容对 INTCNT2 进行初始化。

18.3.9 数字比较(DC)子模块功能描述

在 EPWM 系统内，数字比较子模块与其他子模块交互的信号如结构框图所示。

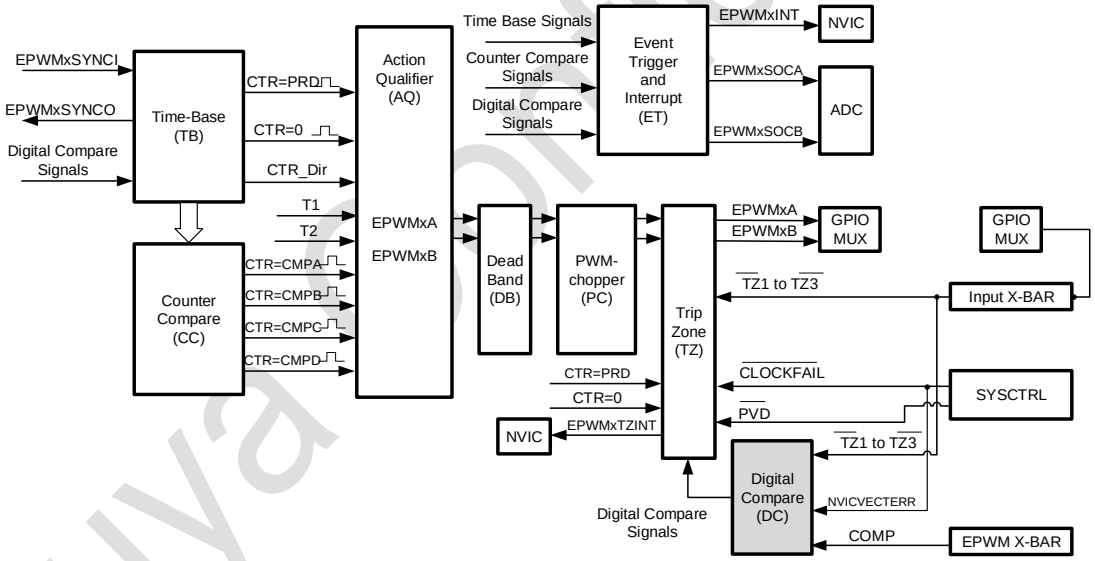


图 18-54 数字比较（DC）子模块

该模块用于将外部的信号 (如 COMPx 信号) 与 EPWM 模块进行比较，直接产生 PWM 事件/动作并反馈给 TB、TZ 和 ET 子模块。该模块支持用于滤除 DC 事件信号中的噪声或不需要的脉冲的空白窗口功能。

Puya Confidential

18.3.9.1 结构框图

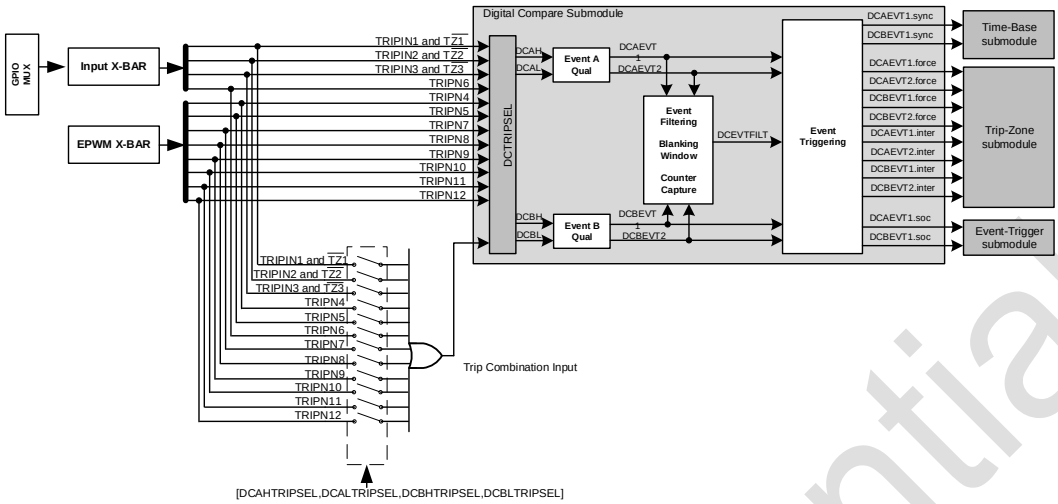


图 18-55 数字比较子模块结构框图

18.3.9.2 输入与输出

如下图所示，在该器件上，任何 GPIO 引脚都可以映射为 TZ 子模块和 DC 子模块的跳闸输入和/或跳闸区输入。通过 EPWM_IXBSELx_y 寄存器可以决定选择 GPIO 引脚为跳闸区输入/跳闸输入。

注意：为了避免误锁存跳闸信号，用户有责任在为相应的 EPWM 外设使能时钟和配置跳闸输入之前，将所选引脚驱动到正确状态。

Puya Confidential

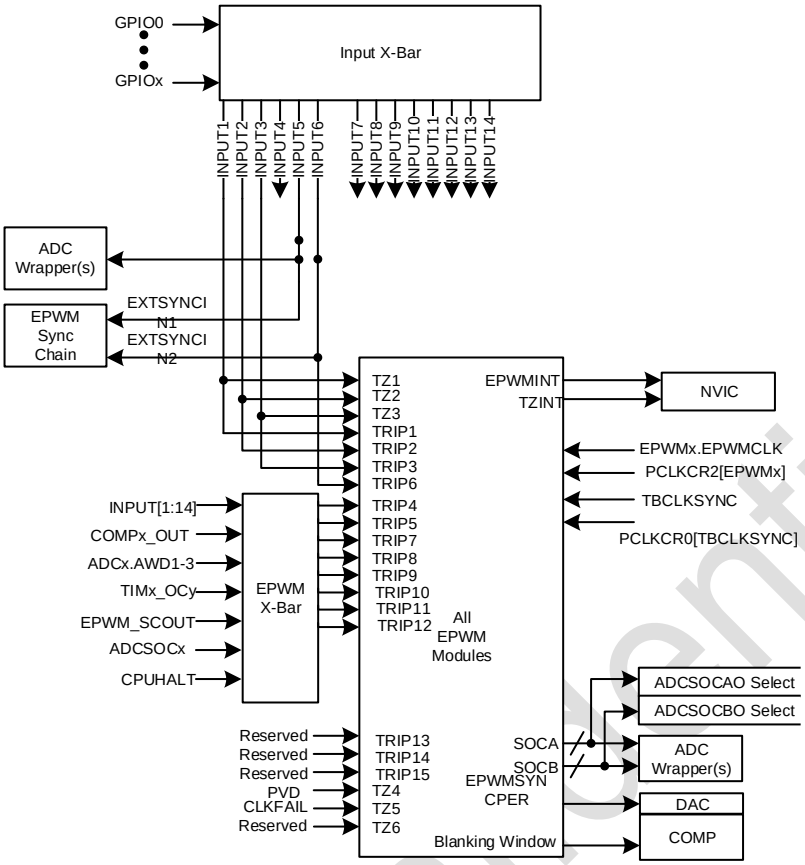


图 18-56 GPIO 复用器到跳闸输入结构框图

18.3.9.3 使用 COMP 跳闸 PWM

增强型跳闸动作

为了允许多个 COMP 能够同一时间影响 DCxEVTy 事件和跳闸动作，通过或逻辑电路将来自 EPWM 模块外部的全部跳闸输入组合在一起，并通过 EPWMx_DCTRIPSEL 寄存器选择作为跳闸输入组合输入到 DCxH 和 DCxL。

可以通过 EPWMx_DCxHTRIPSEL 和 EPWMx_DCxLTRIPSEL 寄存器进行选择将哪些跳闸输入放入该组合逻辑电路，以生成 DCxH 和 DCxL 信号。选择作为跳闸输入组合的输入将会传递到 DCTRIPSEL 寄存器中。

逐周期方式关闭 PWM

当使用 COMP 以逐周期方式跳闸 PWM 时，为了防止在一个 PWM 周期中断首的比较器跳闸状态延续到下一个周期，应该采取一些措施来阻止。比较器可以给处于下游的 EPWM 模块发送跳闸状态信号。对于峰值电流模式控制等应用，期望每个 PWM 周期只有一次跳闸事件。在某些条件下，如果不采取预防措施，持续或延迟的触发事件（在 PWM 周期接近结束时到达）可能会延续到下一个 PWM 周期。如果比较器数字滤波器或 EPWM 数字比较（DC）子模块配置为对比较器错误信号进行限定，则在 EPWM 错误逻辑能够响应错误信号的逻辑变化之前，会引入“N”个时钟周期的限定。一旦 EPWM 跳闸条件被限定，即使比较器错误信号已取消断言，该错误条件也会保持活动状态 N 个时钟周期。如果限

Puya Confidential

定的比较器跳闸信号在 PWM 周期结束前的 N 个时钟周期内保持断言状态，则错误条件直到下一个 PWM 周期开始后才会被清除。因此，新的 PWM 周期一开始就会检测到错误条件。

为避免这种不需要的错误条件，应用程序可以采取措施确保 EPWM 跳闸逻辑看到的限定跳闸信号在每个 PWM 周期结束前被取消断言。这可以通过以下 3 种方法实现：

- 系统设计，使得在 PWM 周期结束前的 N 个时钟周期内不断言比较器错误。
- 使用 EPWM 事件滤波器在 EPWMSYNCPER 信号之前至少两个时钟周期激活比较器跳闸信号的消隐，并持续消隐到下一个 PWM 周期至少 N 个时钟周期。
- 此器件上的 EPWM 模块包括在 CMPC 或 CMPD 匹配时（使用 EPWMx_HRPCTL）生成 EPWMSYNCPER 的能力，以便在 PWM 周期内任意放置 EPWMSYNCPER。

18.3.9.4 操作要点

数字比较事件

如 DC 子模块功能特征所示，可以通过 EPWMx_DCTRIPSEL 寄存器选择模拟比较器模块的 COMPX 等信号和跳闸输入 TZ1n 至 TZ3n 来产生 DCxH 和 DCxL 信号。使用 EPWMx_TZDCSEL 寄存器限定对所选 DCxH 和 DCxL 信号的动作，从而产生 DCxEVTy 事件。

然后，DCxEVTy 事件可以被过滤以提供事件信号的过滤版本 (DCEVTFILT)，或者可以旁路滤波，过滤详情请参见事件过滤小节。DCxEVTy 事件信号或经过滤波的 DCEVTFILT 信号都可以产生对 PWM 同步信号、ADCSOC、TZ 子模块或 TZ 中断的强制事件。

当 TZxn 信号用于数字化较事件跳闸功能时，将该信号视为正常的输入信号且可以定义为低电平有效或高电平有效输入。在 TZxn 信号或者 DCxEVTy 强制信号中任一信号有效时，EPWM 输出被异步跳闸。为了保持锁存状态，需要至少 3*TBCLK 同步脉冲宽度。如果脉冲宽度小于 3*TBCLK 同步脉冲宽度，则跳闸状态可能会被 CBC 或 OST 锁存器锁存或不锁存。

表 18-21 事件信号对其他模块的影响

信号	描述
强制信号	通过 EPWMx_TZCTL， EPWMx_TZCTLDCA 和 EPWMx_TZCTLDCB 寄存器将 DCAEVTy.force 信号配置为强制跳闸区状态可以直接影响 EPWMxA 引脚的输出，或通过 EPWMx_TZSEL 寄存器将 DCAEVTy.force 信号配置为单次跳闸/逐周期跳闸源时可以使用 EPWMx_TZCTL 或 EPWMx_TZCTL2 寄存器配置来影响跳闸动作。在 EPWMx_TZCTL， EPWMx_TZCTL2， EPWMx_TZCTLDCA 和 EPWMx_TZCTLDCB 寄存器中具有冲突操作时的优先级如 EPWMxA/EPWMxB 优先级冲突操作表格所示。
中断信号	DCAEVTy.interrupt 信号生成给中断控制器跳闸区中断。需要在 EPWMx_TZEINT 寄存器中设置 DCxEVTy 位才能使能中断。如果发生其中一个事件，则会触发 EPWMx_TZINT，必须置位 EPWMx_TZCLR 寄存器中的对应位来清除中断。
soc 信号	DCxEVT1.soc 信号与 ET 子模块交互，可以使用 SOCASEL 位和 SOCBSEL 位选择生成 SOCA 和 SOCB 脉冲的事件。
同步信号	DCxEVT1.sync 信号与 EPWMx 同步输入信号和 SWFSYNC 位相或，从而生成同步脉冲给 EPWMx_TBCTR。

Puya Confidential

表 18-22 EPWMxA/EPWMxB 优先级冲突操作

输出	优先级冲突操作
EPWMxA	（1）优先级最高为 TZA，其次为 DCAEVT1，优先级最低为 DCAEVT2 （2）优先级最高为 TZAU，其次为 DCAEVT1U，优先级最低为 DCAEVT2U （3）优先级最高为 TZAD，其次为 DCAEVT1D，优先级最低为 DCAEVT2D
EPWMxB	（1）优先级最高为 TZB，其次为 DCBEVT1，优先级最低为 DCBEVT2 （2）优先级最高为 TZBU，其次为 DCBEVT1U，优先级最低为 DCBEVT2U （3）优先级最高为 TZBD，其次为 DCBEVT1D，优先级最低为 DCBEVT2D

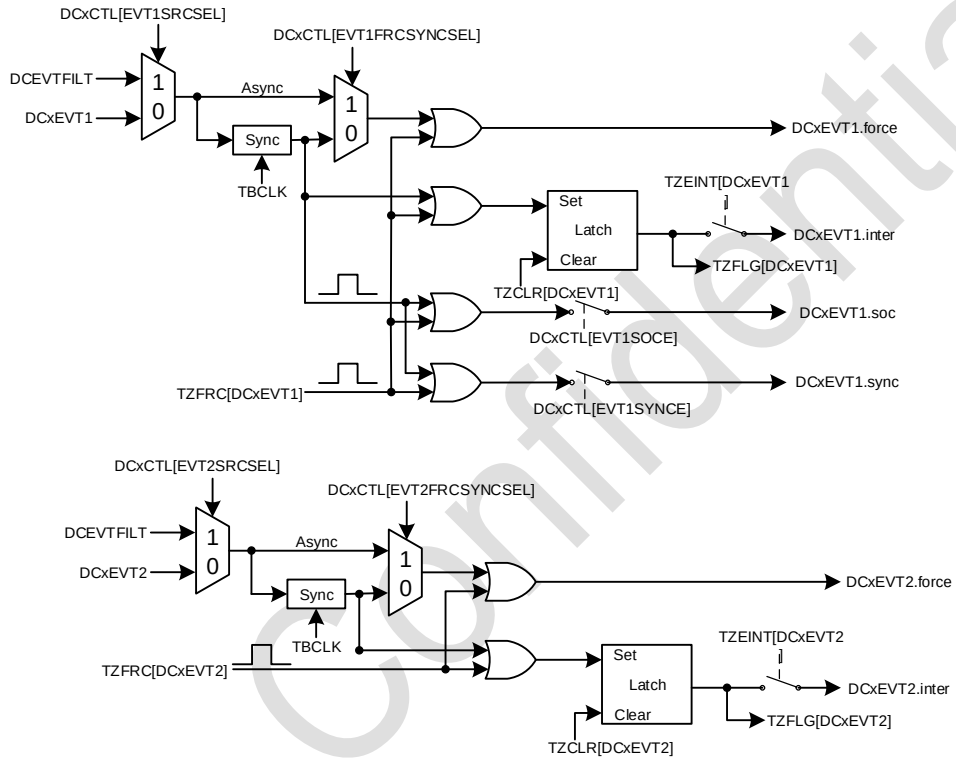


图 18-57 DCxEVTy 事件触发结构框图

事件过滤

（1）空白控制逻辑

该逻辑用于定义空白窗口，在窗口处于活动状态时忽略信号上发生的所有事件。通过 EPWMx_DCFCTL、EPWMx_DCFOFFSET 和 EPWMx_DCFWINDOW 寄存器来配置空白窗口，其中通过 BLANKE 位使能空白窗口，并按照 PULSESEL 位域的选择将空白窗口与所选择的脉冲同时对齐。

为了消除噪声，DCxEVTy 事件可以使用事件过滤逻辑电路来过滤，选择性地在一段时间内空白事件去除噪声。该方式可以用于选择模拟比较器输出来触发 DCxEVTy 事件，且空白控制逻辑用于在跳闸 PWM 输出、产生中断或 ADCSOC 之前滤除信号上潜在的噪声。

通过 SRCSEL 位域选择滤波块信号源为 DCxEVTy 信号。将 EPWMx_TBCTR 计数偏移值编程到 EPWMx_DCFOFFSET 寄存器中，该寄存器确定空白窗口在 EPWMx_TBCTR=0 或 EPWMx_TBCTR= EPWMx_TBPRD 之后的开始点。空白窗口的持续时间（以偏移计数器到期后的

Puya Confidential

TBCLK 计数数量表示) 通过应用程序写入 EPWMx_DCFWINDOW 寄存器。在空白窗口结束之前和结束之后，事件仍然可以像以前一样生成强制、中断、同步和转换开始信号。

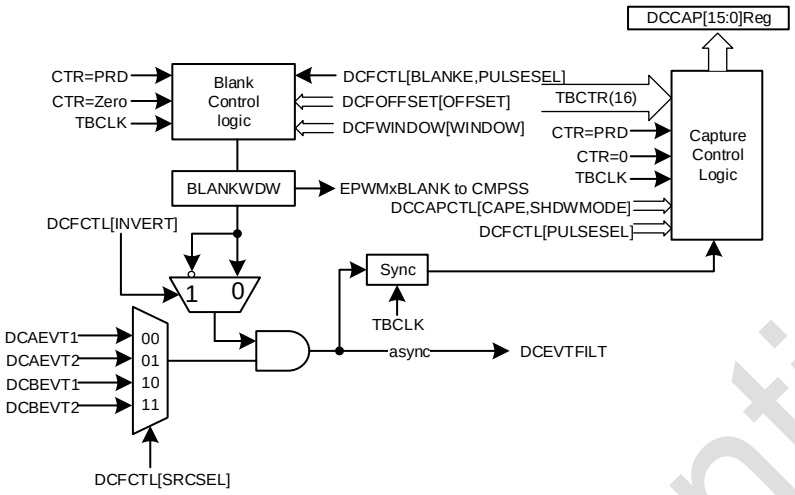


图 18-58 事件过滤结构框图

（2）捕获控制逻辑

事件过滤还可以捕获基于 EPWMx_DCCAPCTL 寄存器所选 DCxEVTy 事件的计数值。

如果使能空白控制逻辑，则所选 DCxEVTy 事件触发将计数器捕获到活动寄存器中，CPU 从活动寄存器中读取。如果使能影子模式，则在 PULSESEL 位域所选的事件中将活动寄存器的值复制到影子寄存器，CPU 从影子寄存器中读取。

在所选的 DCxEVTy 事件之后直到 CAPMODE 位指定的事件发生之前，不会发生其他捕获事件。CAPMODE 可以通过以下方式进行配置：

- 通过 CAPCLR 位清除 CAPSTS 位之前，不会发生其他捕获事件
- 在 PULSESEL 位域所选的事件发生之前，不会发生其他捕获事件

在 PWM 周期内空白窗口和偏移的几种时序条件如下图所示。当空白窗口跨越 EPWMx_TBCTR=0 或 EPWMx_TBCTR= EPWMx_TBPRD 边界时，下一个窗口仍然从相同的偏移值处开始，即在 EPWMx_TBCTR=0 或 EPWMx_TBCTR= EPWMx_TBPRD 脉冲之后。

Puya Confidential

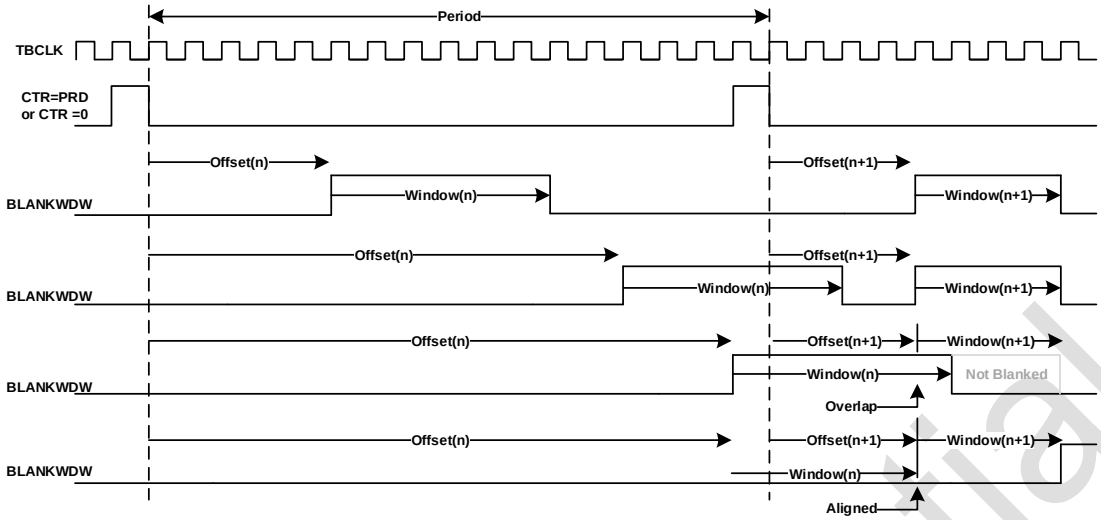


图 18-59 空白窗口时序图

谷底开关功能

事件过滤描述了事件过滤逻辑以及谷底开关功能，谷底开关功能可以在不增加任何外部电路的情况下实现可编程谷底开关。该功能提供一个片上硬件机制，支持捕获振荡周期，可以精确延迟 PWM 开关时刻，且在延迟生效前允许可编程的边沿数，此外，还提供多种触发器和事件选择以及易于适应变化的系统/操作条件下的最佳性能。

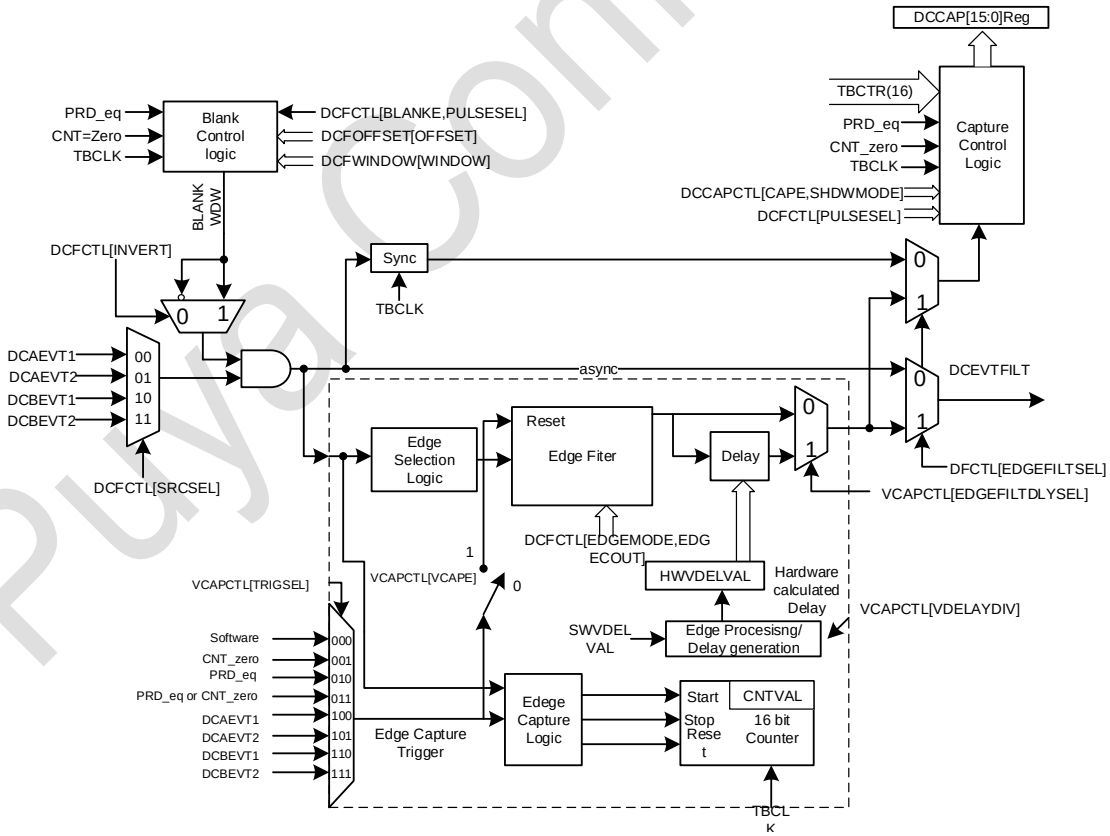


图 18-60 谷底开关结构框图

Puya Confidential

DCxEVTy 信号需要进一步处理以支持谷底开关，使能谷底开关功能操作步骤如下：

表 18-23 使能谷底开关功能操作步骤

步骤	操作	描述
1	选择滤波块信号源	通过 SRCSEL 位域选择谷底开关的输入源，可以为此输入添加空白窗口（空白控制逻辑）。在这里，将谷底开关的输入选择为比较器输出或外部输入。
2	选择边沿模式和数量	通过 EDGEMODE 和 EDGECOUNT 位域配置边沿滤波器捕获上升沿、下降沿或双边沿的边沿数量。
3	选择事件重启边沿滤波器	根据 TRIGSEL 位域选择正确的事件来复位并重启沿滤波器，边沿捕获事件由选定的边沿触发或发送。
4	VCAPE=1	使能谷捕获逻辑
5	选择开始计数的边沿	通过 STARTEDGE 位域选择振荡周期测量捕获的开始边沿，即 16 位计数器开始计数
6	选择停止计数的边沿	通过 STOPEDGE 位域选择 16 位计数器停止计数的边沿。振荡周期信息由捕获的计数器值提供，且 STARTEDGE 值必须始终小于 STOPEDGE 值。
7	配置并应用捕获延迟	对边沿滤波后的 DCxEVTy 信号进行配置并应用捕获延迟。谷时基计数器的值可以直接应用、与用于偏移调整的软件编程值一起应用或只有一小部分延迟可以应用/不应用软件谷延迟值。这可以正确应用于谷值点的延迟。
8	配置边沿滤波器输出延迟	置位 EDGEFILTDLYSEL 位，根据上面捕获的值应用硬件延迟

18.3.10 高分辨率(HR)功能描述

HRPWM 基于微边沿定位器 (MEP) 技术。MEP 逻辑能够通过细分传统 PWM 发生器的一个粗系统时钟来非常精细地定位边沿。时间步进精度约为 130 ps 量级。有关特定器件上的典型 MEP 步进大小，请参阅器件数据手册。EPWM 还具有自检软件诊断模式，用于检查 MEP 逻辑在所有操作条件下是否按设计运行。

Puya Confidential

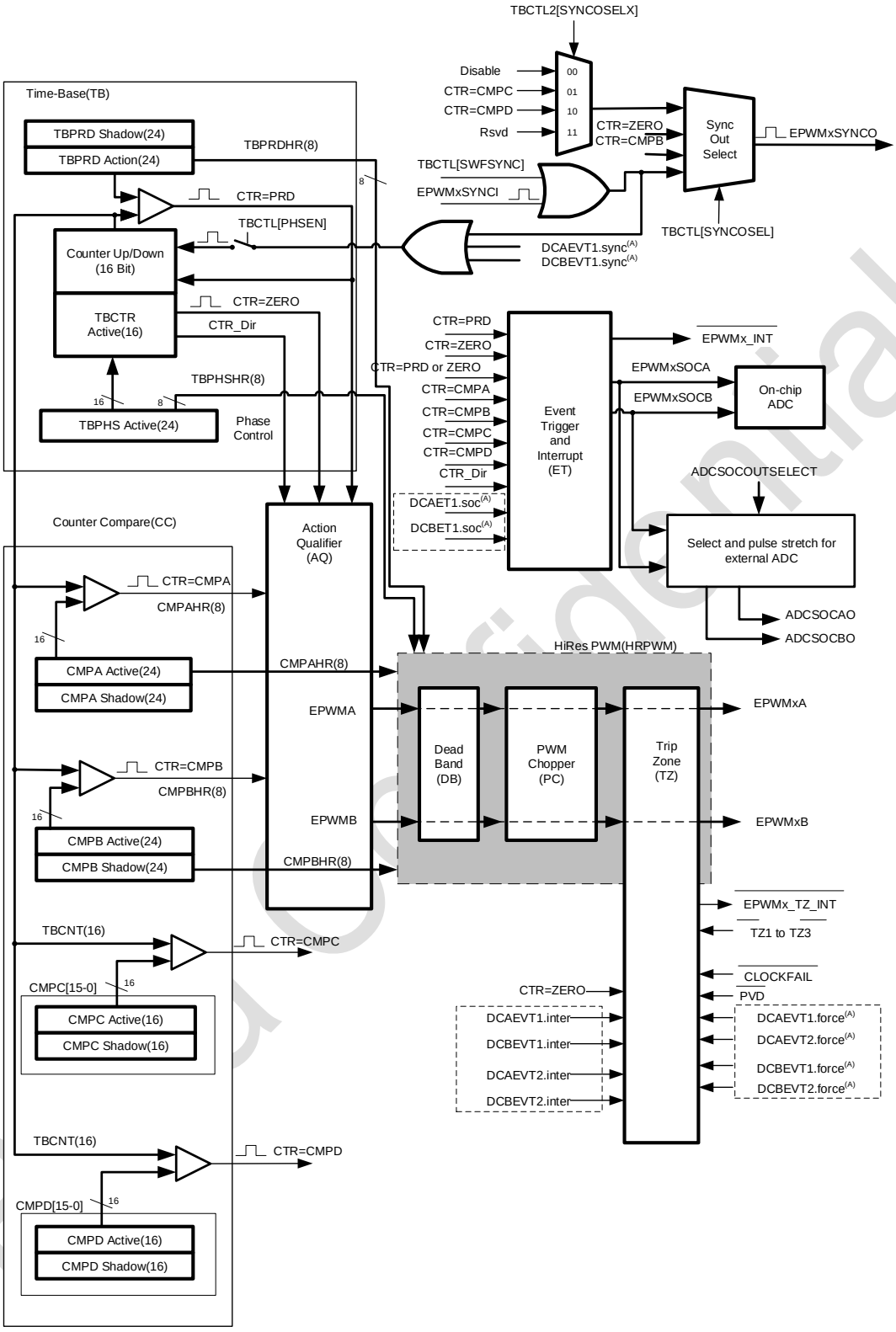


图 18-61 HRPWM 系统接口

Proprietary

18.3.10.1 结构框图

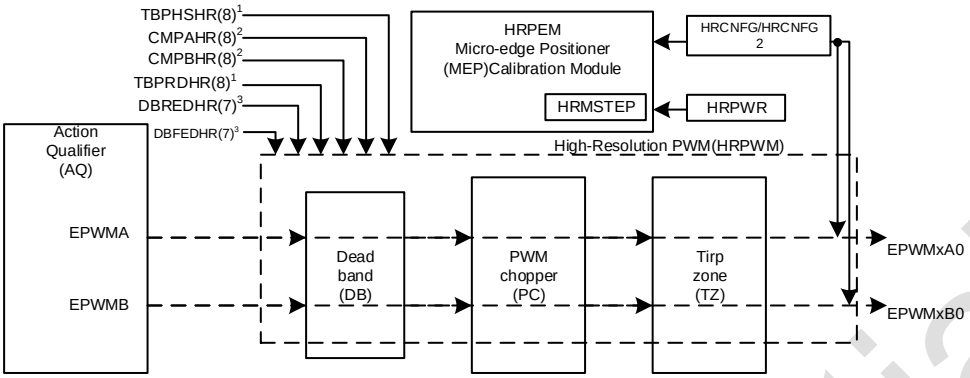


图 18-62 HRPWM 结构框图

HRPWM 功能并不是一个单独的模块来实现，而且由输出路径延时、DB、PC、TZ 以及校准模块（只有 EPWM1 含有）共同完成，如上图所示。其中，TBPHSHR 和 TBPRDHR 来自 TB 时基子模块，CMPAHR 和 CMPBHR 来自 CC 计数器比较子模块， DBREDHR 和 DBFEDHR 来自 DB 死区发生器子模块。

18.3.10.2 PWM 分辨率计算

PWM 外设用于实现数学上等价于数模转换器(DAC)的功能。常规生成的 PWM 时序图如图所示，其有效分辨率与 PWM 频率或周期和系统时钟频率有关。常规生成的 PWM 的分辨率计算：

PWM 分辨率（%） = $F_{PWM}/F_{EPWMCLK} * 100\%$

PWM 分辨率（bits） = $log2\ (T_{PWM}/T_{EPWMCLK})$

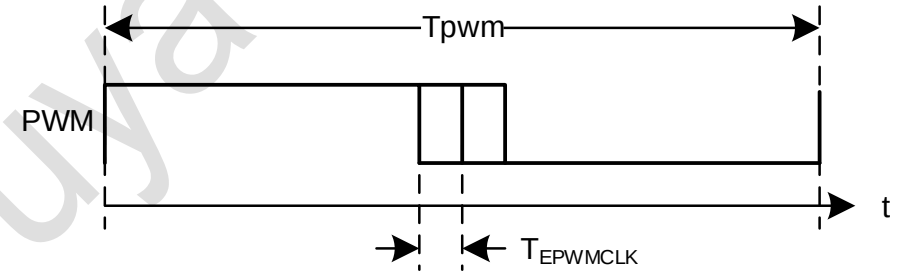


图 18-63 常规生成的 PWM 时序图

如果所需的 PWM 工作频率不能在 PWM 模式下提供足够的分辨率，则考虑使用 EPWM 功能。假设常规分辨率在 EPWMCLK 为 100MHz 条件下且 MEP 步进为 180 ps。不同 PWM 频率下，以位数为单位的分辨率如下表所示。MEP 的典型单最大性能指标请参见表格手册。

Puya Confidential

表 18-24 不同频率下的 PWM 和 HRPWM 分辨率

PWM 频率	PWM 位数	PWM 百分比	HRPWM 位数	HRPWM 百分比
20 kHz	12.3	0.02%	18.1	0.00%
50 kHz	11	0.05%	16.8	0.00%
100 kHz	10	0.10%	15.8	0.00%
150 kHz	9.4	0.15%	15.2	0.00%
200 kHz	9	0.20%	14.8	0.00%
250 kHz	8.6	0.25%	14.4	0.01%
1000 kHz	6.6	1.00%	12.4	0.02%
1500 kHz	6.1	1.50%	11.9	0.03%
2000 kHz	5.6	2.00%	11.4	0.04%

18.3.10.3 MEP 技术

MEP 逻辑能如下所示: MEP 步进由 CMPAHR 寄存器的高 8 位控制且 CMPBHR 寄存器也适用于时钟的传输管理。控制 CMPA 寄存器值等于粗调步进的数量，CMPAHR 寄存器值等于微调步进的数量左移 8 位，然后将结果存储在该寄存器值的高 8 位中。

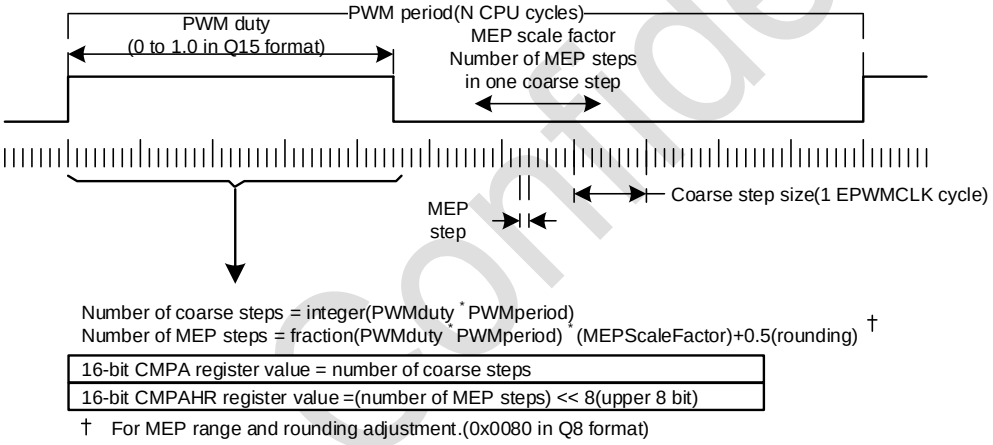


图 18-64 MEP 逻辑图

粗步进与 MEP 步进计算公式:

- 粗步进=int（PWM 周期 * PWM 占空比）
- MEP 步进= fraction（PWM 周期 * PWM 占空比）* MEP 比例系数+0.5。其中 0.5 用于范围和四舍五入的测量（考虑 0x0000 的 Q8 格式）

需要配置常规 PWM 寄存器以产生给定频率和极性的常规 PWM，才能产生 HRPWM 波形。为了扩展边沿的分辨率，可以使 常规 PWM 寄存器与高分辨率寄存器一同工作。在多种编程组合中，只有少数为所需且实用的编程组合。

18.3.10.4 控制 HRPWM 功能

通过以下扩展寄存器控制 EPWM 的 MEP，这些寄存器与用于控制 PWM 操作的 EPWMx_CMPA、EPWMx_CMPB、EPWMx_TBPHS、EPWMx_TBPRD、EPWMx_DBRED 和 EPWMx_DBFED 寄存器连接:

Puya Confidential

表 18-25 HRPWM 扩展寄存器

寄存器名	描述
CMPAHR	计数比较 A 高分辨率扩展寄存器，该寄存器用于动作限定通道 A 的输出，与 CMPA 寄存器无关
CMPBHR	计数比较 B 高分辨率扩展寄存器，该寄存器用于动作限定通道 B 的输出，与 CMPB 寄存器无关
TBPHSHR	高分辨率时基相位偏移寄存器
TBPRDHR	高分辨率时基周期寄存器，只适用于部分器件
DBREDHR	死区上升沿延迟高分辨率寄存器
DBFEDHR	死区下降沿延迟高分辨率寄存器

注意：不得使用 TBPHSHR。应使用 TRREM（HRPWM 余数寄存器）来模拟 TBPHSHR 的功能。

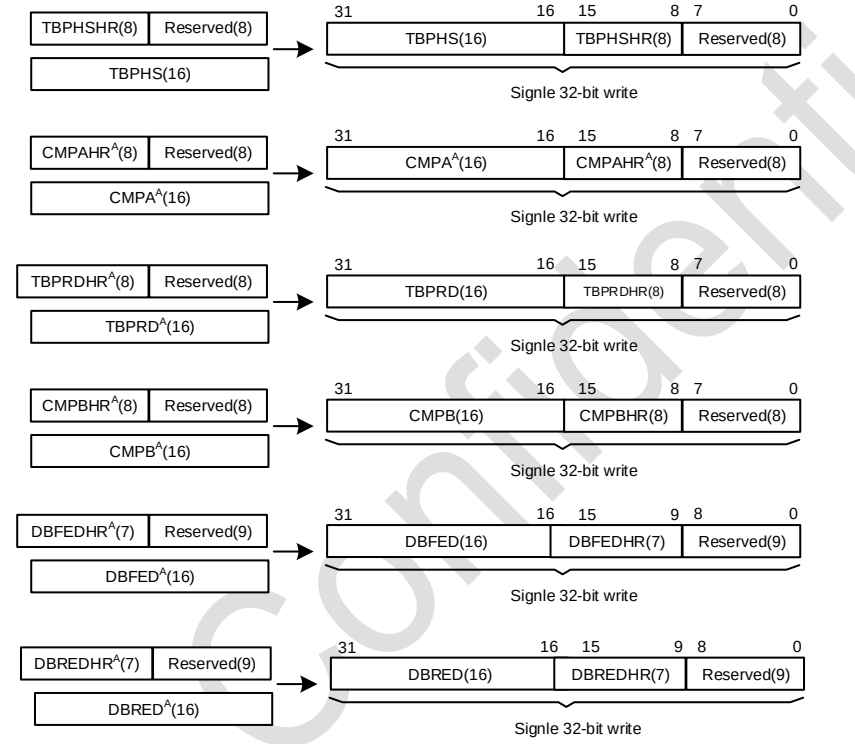


图 18-65 HRPWM 扩展寄存器和内置配置

死区上升沿延迟和下降沿延迟的 HRPWM 能力仅适用于死区半周期时钟操作。MEP 步进的数量位(15:9)是占空比和相位高分辨率寄存器尺寸的一半。

HRPWM 功能是通过通道 A 和 B 的 PWM 信号路径进行控制。通过正确配置 HRCNFG2 寄存器，可以在死区信号路径上支持 HRPWM。EPWM 与 8 位扩展寄存器的接口结构框图请参见 PWM 主要信号章节。

18.3.10.5 时钟源

通过 EPWM1CLK 时钟来驱动 HRPWM 模块和 HRCAL 模块，因此，必须先使能 EPWM1CLK 才能使能 HRCAL 或任何 HRPWM 模块。

Puya Confidential

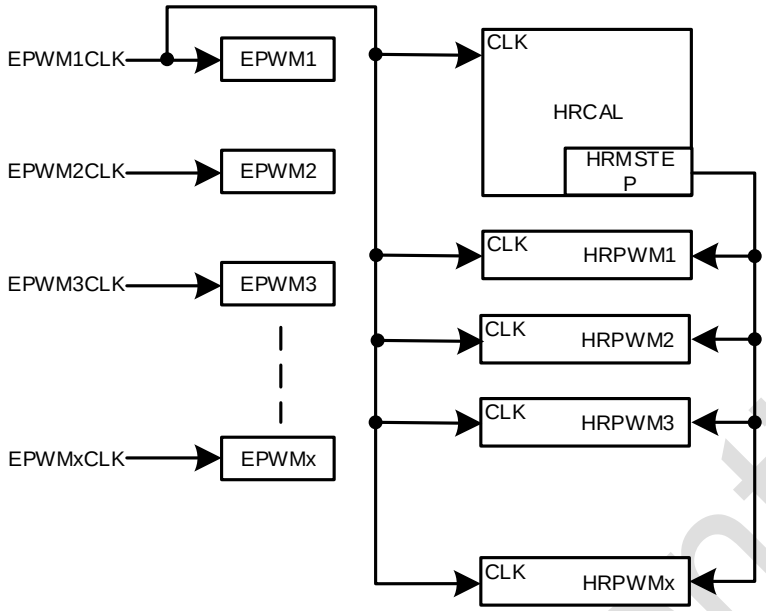


图 18-66 EPWM 和 HRCAL 时钟源

18.3.10.6 配置 HRPWM

如果将 PWM 配置为常规 PWM（提供给定频率和极性），可以通过编程 HRCNFG 寄存器来配置 HRPWM。HRCNFG 寄存器可以提供以下配置选项。

边沿模式

可以编程的 MEP 为上升沿（RE）、下降沿（FE）或双边沿（BE）提供了精确的位置控制。其中，FE 和 RE 由 CMPAHR 或 CMPBHR 寄存器控制，用于需要占空比控制的电源拓扑，BE 用于需要相移的拓扑，例如 TBPRDHR 或 TBPHSHR 寄存器控制的相移全桥。

控制模式

可以将 MEP 编程为以下两种选择：

- 在占空比控制情况下，由 CMPAHR 或 CMPBHR 寄存器控制
- 由 TBPHSHR 寄存器控制

CMPAHR 或 CMPBHR 寄存器可以与 RE 或 FE 控制模式同时使用。TBPHSHR 寄存器可以与 BE 控制模式同时使用。当在周期控制 TBPHSHR 寄存器情况下控制 MEP 时，也可以通过各自的高分辨率寄存器来控制占空比和相位。

影子模式

提供了与常规 PWM 模式相同的双缓冲/影子化选项，只有在由 CMPxHR 和 TBPHSHR 寄存器操作时有效且可以选择与 CMPA/CMPB 寄存器的常规加载选项相同。在使用 TBPHSHR 寄存器时无效。

控制高分辨率 B 信号

PWM 通道的 B 信号可通过 EPWMxB 引脚输出 EPWMxA 信号的反相来实现高分辨率输出。HRPWM 模块支持在独立于 A 信号的 B 信号路径上实现高分辨率功能。

交换高分辨率输出

该模式允许交换高分辨率 A 和 B 输出。模式可以选择以下方式：

Puya Confidential

- 允许 A 和 B 输出信号保持不变
- 交换 A 和 B 输出信号，即将 A 输出到 B，B 输出到 A

自动转换模式

自动转换模式只能与 SFO 软件一起使用。

使能自动转换: CMPAHR= [fraction（PWM 周期 * PWM 占空比）]<<8。在后台代码中，SFO 软件将计算 MEP 比例系数，并将计算出的每粗步进的 MEP 步进数自动更新到 HRMSTEP 寄存器中。MEP 校准模块通过使用 HRMSTEP 和 CMPAHR 寄存器中的值自动计算合适的 MEP 步进数（表示为小数的占空比），并相应地移动高分辨率 PWM 信号边沿。

禁止自动转换：CMPAHR=[fraction（PWM 周期 * PWM 占空比）* MEP 比例系数+0.5]<<8。在该模式下，所有计算都需要由用户的代码执行且 HRMSTEP 寄存器将被忽略。高分辨率周期与高分辨率占空比的自动转换行为相同。如果使能高分辨率周期模式，则必须始终使能自动转换。

注意：当 EPWM 模块配置为中心对齐计数模式时，必须将高分辨率相关寄存器的影子模式设置为在 EPWMx_TBCTR=0 和 EPWMx_TBCTR= EPWMx_TBPRD 时装载。只有当 EPWMx_TBCTR=0 时，用户的新值才会加载到影子寄存器中（寄存器的影子模式必须同时设置为 EPWMx_TBCTR=0 和 EPWMx_TBCTR= EPWMx_TBPRD）。EPWMx_TBCTR= EPWMx_TBPRD 事件用于高分辨率功能的特定内部逻辑。

自动转换模式执行 CMPBHR、DBREDHR 和 DBFEDHR 的计算：

使能自动转换：CMPAHR=[fraction（PWM 周期 * PWM 占空比）]<<8。在后台代码中，SFO 软件将计算 MEP 比例系数，并将计算出的每组步进的 MEP 步进数自动更新到 HRMSTEP 寄存器中。然后，MEP 校准模块通过使用 HRMSTEP 和 CMPBHR / DBREDHR/DBFEDHR 寄存器中的值自动计算合适的 MEP 步进数（表示为小数的占空比），并相应地移动高分辨率 PWM 信号边沿。

禁止自动转换: CMPBHR 行为与 CMPAHR 一致，即 CMPBHR=[frac（PWM 周期 * PWM 占空比）*MEP 比例系数+0.5]<<8

18.3.10.7 工作原理

MEP 逻辑能够将边沿放置在 255 个（8 位）离散时间点之一上，典型 MEP 步进请参见数据手册。为了确保应用时间步进以及各种条件（宽范围的 PWM 频率和系统时钟频率等）下保持边沿的放置精度，MEP 与 TBM 和 CCM 寄存器一起工作。EPWM 支持的典型频率范围如下所示。

表 18-26 EPWM 支持的典型频率范围

系统时钟（MHz）	最大分辨率（位数）(1)	每 EPWMCLK 的 MEP 步进(2)(3)	PWM 频率	
			最低（Hz）(4)	最高（MHz）
60	10.9	93	916	3
70	10.6	79	1068	3.5
80	10.4	69	1221	4
90	10.3	62	1373	4.5
100	10.1	56	1526	5

Puya Confidential

注意：

（1）所给出以位数为单位的分辨率，是针对于所述的最大 PWM 频率

（2）TBCLK=EPWMCLK

（3）表中数据基于 180ps 的 MEP 时间分辨率，其中 MEP 步进=TEPWMCLK/180ps。MEP 的范围请参见数据手册。

（4）PWM 为非对称向上计数模式且 PWM 最低频率是基于一个最大周期值 65535。

边沿放置

数字控制器在一个典型的功率控制回路中发出一个以每单位或百分比表示的占空比命令。如果一个工作点要求占空比为 40.5%，要求变换器 PWM 频率为 1.25 MHz。在常规 PWM 生成中（系统时钟为 100MHz），占空比的选择大约是 40.5%左右。如下表所示，占空比最接近 40.5%数值的是在计数值 32，此时占空比为 40%，这相当于边沿放置在 320ns。

假设 MEP 的步进分辨率为 180ps，利用 MEP 可以得到更接近期望点 324ns 的边沿位置。在 CMPA 计数值之外，MEP 的 22 步进将边沿放置在 323.96ns，导致几乎零误差。

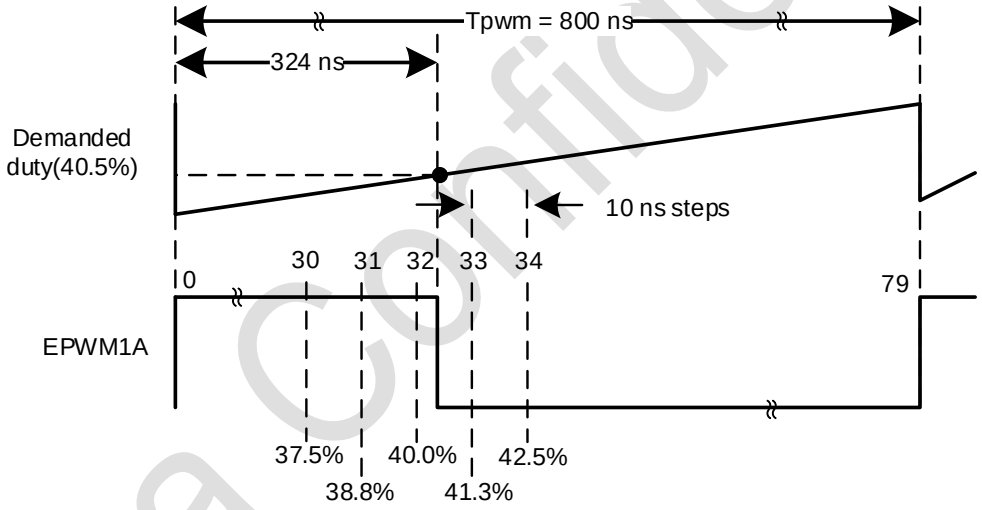


图 18-67 请求占空比为 40.5%时所需的 PWM 波形

表 18-27 CMPA 与占空比（左三列）以及[CMPA:CMPAHR]与占空比（右四列）

CMPA 计数值 (1) (2) (3)	高电平时间	占空比	CMPA 计数值	CMPAHR 计数值	高电平时间	占空比
28	280 ns	35.00%	32	18	323.24ns	40.41%
29	290 ns	36.30%	32	19	323.42ns	40.43%
30	300 ns	37.50%	32	20	323.60ns	40.45%
31	310 ns	38.80%	32	21	323.78ns	40.47%
32	320 ns	40.00%	32	22	323.96ns	40.50%
33	330 ns	41.30%	32	23	324.14ns	40.52%
34	340 ns	42.50%	32	24	324.32ns	40.54%
...	...	...	32	25	324.50ns	40.56%
32.4	324ns	40.50%	32	26	324.68ns	40.59%

Puya Confidential

注意：

（1） TBCLK 为 100MHz，10ns。

（2） MEP 的步进分辨率为 180ps，MEP 典型值和最大值请参见数据手册。

（3） 对于 PWM 周期寄存器值为 80，TPWM=80×10ns=800ns，FPWM=1/TPWM=1.25MHz

使用[CMPB:CMPBHR]进行占空比控制时，其操作方式以及公式与 [CMPA:CMPAHR]寄存器组合一致。

缩放的考量

使用标准的CMPA和MEP (CMPAHR) 寄存器的资源来精确地在时间上定位边沿的机制已经演示过了。在实际应用中，需要将小数的占空比写入到最终的整数（[CMPA:CMPAHR]寄存器组合）表示，并无缝地将该映射函数提供给 CPU。

要做到这一点，首先所涉及的映射或缩放步骤都需要进行检查。在控制软件中，通常使用每单位或百分比表示占空比。这样有利于执行所有的数学计算都无需关心最终用时钟计数或 ns 表示高电平时间的绝对占空比。此外，它使代码更易于兼容不同 PWM 频率的多种转换器类型。

假设条件：

- TBCLK=10 ns（100 MHz）
- FPWM=1.25 MHz (1/800 ns)
- PWM 占空比为 40.5%
- 粗步进的 PWM 周期为 80 (800 ns/10 ns)
- 每个粗步进的 MEP 步进为 180ps 时，MEP 比例系数为 55（10ns/180ps）
- 用于将 CMPAHR 保持在 1-255 之间和小数舍入的常数为 0.5（默认值）(0080h，Q8 格式)

需要以下两步缩放过程便于实现映射方案：

- CMPA 寄存器的百分比整数部分占空比转换
- CMPA 寄存器值 = int（PWM 周期 * PWM 占空比）=32。int 表示整数部分
- CMPAHR 寄存器值 = 32 = 0x20
- CMPAHR 寄存器小数部分转换
- CMPAHR 寄存器值 = [frac（PWM 周期 * PWM 占空比）* MEP 比例系数 + 0.5]<<8=22.5*256=5760，即 0x1680。 frac 表示小数部分
- CMPAHR 寄存器值 = 0x1680，也即 0x1600，因为硬件忽略低 8 位。

当 AUTOCONV 置位且 MEP 比例系数在 HRMSTEP 寄存器中时，CMPAHR 寄存器值 = frac（PWM 周期 * PWM 占空比）。由硬件自动完成其余转换计算，且 PWM 通道将输出正确的经 MEP 缩放的信号边沿。当 AUTOCONV 未置位时，必须通过软件执行上述计算。

MEP 比例系数受系统时钟和 DSP 运行情况的影响。PUYA 提供的 MEP 比例系数 C 函数利用 EPWM 的内置诊断返回最佳比例系数，因其变化缓慢，优化过程可以在后台循环中进行。

Puya Confidential

CPU 的 32 位数能能力可以将在内存中配置的 CMPx 和 CMPxHR 寄存器作为单个拼接的值来写入，即[CMPA:CMPAHR]，[CMPB:CMPBHR]等

注意：

（1） 实际实现利用 Arm 的 32 位 CPU 体系结构，与上述步骤有所不同。

（2） 对于连一个周期都要考虑的时间关键控制循环，建议使用优化了周期的函数（汇编版本）。它接受一个 Q15 占空比值作为输入并写入单个[CMPA:CMPAHR]值。

限制占空比

在高分辨率模式下，MEP 并非在 100% 的 PWM 周期内都有效。MEP 可工作在以下情况：

1 未使能高分辨率 PWM 周期控制

MEP 在周期开始的三个 EPWMCLK 周期之后工作

2 使用 EPWMx_HRPCTL 寄存器使能高分辨率 PWM 周期控制

(1)向上计数模式

MEP 工作于周期开始的三个 EPWMCLK 周期后到周期结束的三个 EPWMCLK 周期之间

(2)中心对齐计数模式

向上计数时，MEP 工作于 EPWMx_TBCTR=0 三个周期后到 EPWMx_TBCTR= EPWMx_TBPRD 三个周期之间

向下计数时，MEP 工作于 EPWMx_TBCTR= EPWMx_TBPRD 三个周期后到 EPWMx_TBCTR=0 三个周期之间

3 使用 DBREDHR 或 DBFEDHR

— DBRED 或 DBFED（对应边沿高分辨率放置的寄存器）必须大于或等于 7。占空比范围限制如下图所示，它限制了 MEP 的占空比。例如，在占空比为 0%时，不可以使用精确的边缘控制。如果禁止高分辨率 PWM 周期控制，则可以使常规的 PWM 占空比控制完全工作到 0%占空比，即使 HRPWM 功能在前三个周期中不可用。但在大多数应用中，控制器调节点通常不会设计为接近 0%的占空比，所以可以避免该问题。要更好地了解可用占空比范围，请参见下表。如果使能高分辨率 PWM 周期控制，则占空比不能落入被限制的范围内，否则，可能在 EPWMxA 输出上存在未定义的行为。

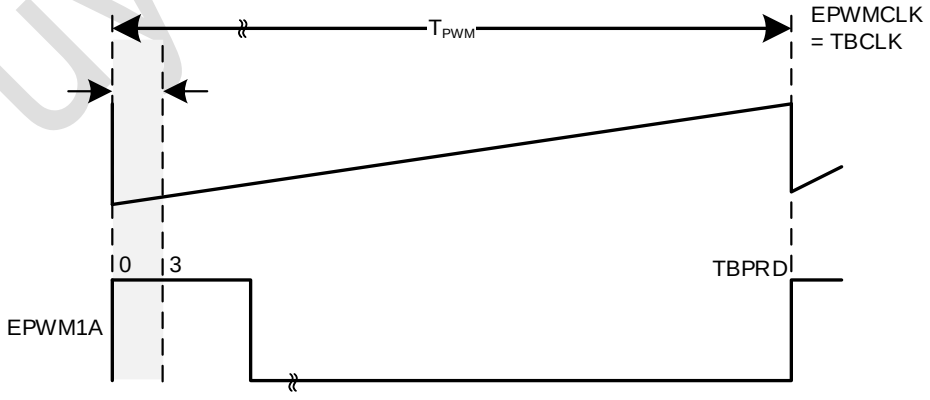


图 18-68 低百分比占空比范围限制示例 (HRPCTL[HRPE] = 0)

Puya Confidential

表 18-28 三个周期的占空比范围限制

PWM 周期 (1)	占空比	
	最小值	最大值 (2)
200kHz	0.60%	99.40%
400kHz	1.20%	98.80%
600kHz	1.80%	98.20%
800kHz	2.40%	97.60%
1000kHz	3%	97%
1200kHz	3.60%	96.40%
1400kHz	4.20%	95.80%
1600kHz	4.80%	95.20%
1800kHz	5.40%	94.60%
2000kHz	6%	94%

注意：

（1）TBCLK=EPWMCLK=100MHz

（2）该限制仅适用于使能高分辨率 PWM 周期控制时

当应用要求 EPWM 工作到最小占空比限制以下时，可以配置 EPWM 工作在向下计数模式，可以在 HRPE=0 时，上升沿位置由 MEP 控制，如下图所示，可以在这种配置下解决最小占空比限制的问题，但存在一个相同百分比的最大占空比限制。

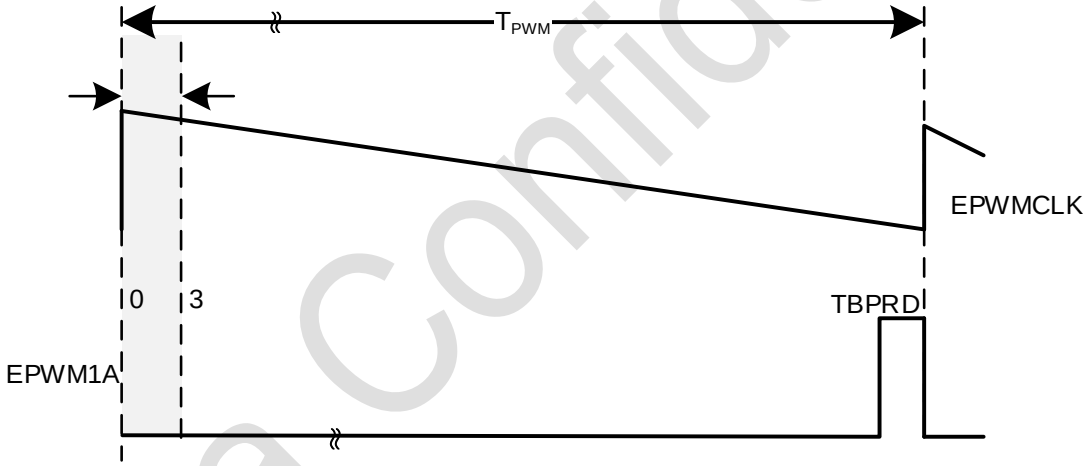


图 18-69 高百分比占空比范围限制示例 (HRPCTL[HRPE] = 0)

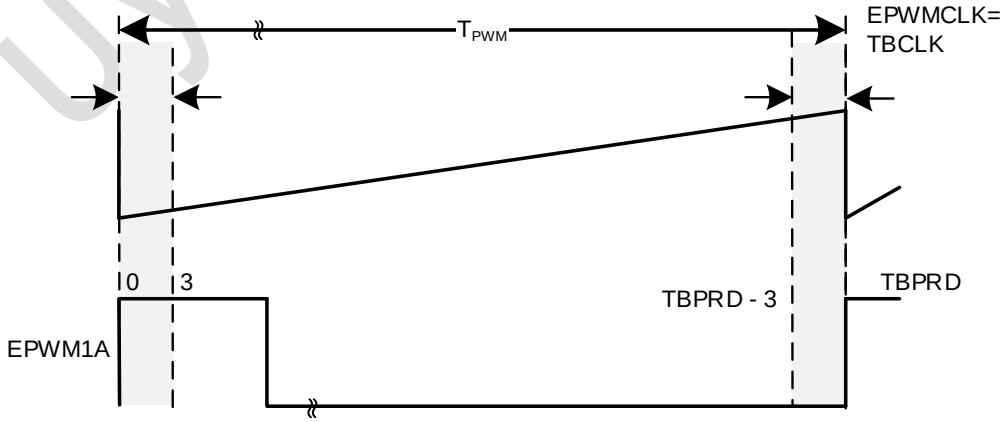


图 18-70 向上计数占空比范围限制示例 (HRPCTL[HRPE] = 1)

Puya Confidential

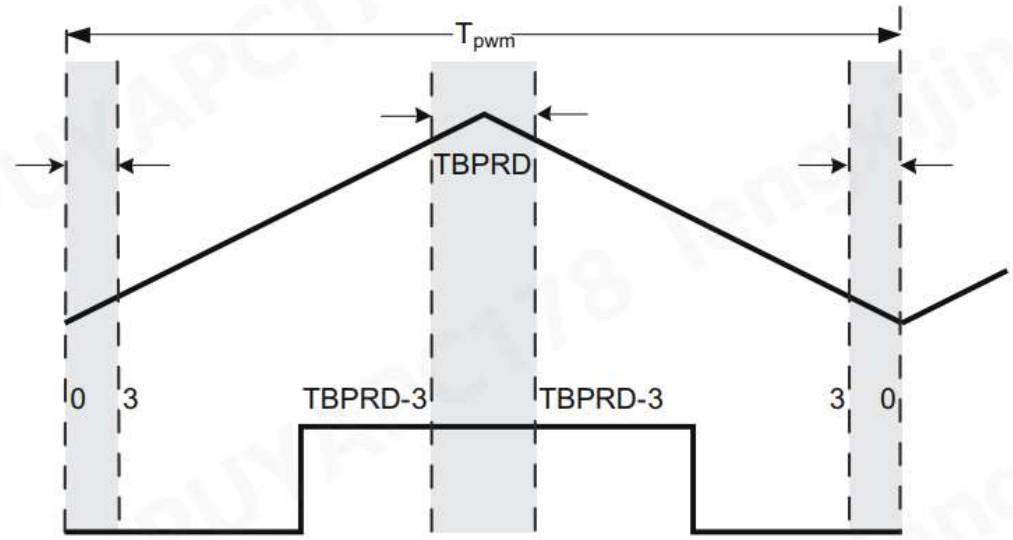


图 18-71 向上-向下计数占空比范围限制示例 (HRPCTL[HRPE] = 1)

高分辨率周期

如果需要使用高分辨率 PWM 周期，则 HRPWM 模块必须按照特定顺序进行初始化。

将 CMPA 及其影子寄存器和相应的 HRCNFG 位域一起用于 EPWMxA 上的高分辨率操作步骤如下所示。将 A 通道替换为 B 通道的对应寄存器即可进行 EPWMxB 的高分辨率操作。

表 18-29 高分辨率 PWM 周期操作步骤

步骤	描述
1	使能 EPWMn 时钟
2	使能 HRPWM 时钟
3	禁止同步 EPWM 时钟（EPWMx_TBCLKSYNC）
4	配置 EPWMx 寄存器： （1） EPWMx 仅可配置为向上或中心对齐计数模式。高分辨率 PWM 周期与向下计数模式不兼容。 （2） EPWMx_TBPRD 和 CC 寄存器必须配置为影子装载。 （3） CMPCTL [LOADAMODE] 向上计数模式： LOADAMODE=01，在 EPWMx_TBCTR= EPWMx_TBPRD 时装载 中心对齐计数模式： LOADAMODE=10，在 EPWMx_TBCTR=0 或 EPWMx_TBCTR= EPWMx_TBPRD 时装载
5	配置 EPWMx_HRCNFG 寄存器： （1） HRLOAD=10，在 EPWMx_TBCTR=0 或 EPWMx_TBCTR= EPWMx_TBPRD 时装载 （2）使能自动转换（AUTOCONV=1） （3） MEP 逻辑同时控制 A 路信号两个边沿（EDGMODE =11）
6	高分辨率 PWM 周期的 TBPHS:TBPHSHR 同步，将 TBPSHRLOADE 位和 PHSEN 位置位。在中心对齐计数模式下，不管 TBPHSHR 寄存器中任何内容，这些位都必须置位。
7	使能高分辨率 PWM 周期控制（HRPE=1）
8	使能同步 PWM 时钟 EPWMx_TBCLKSYNC
9	软件强制产生一个同步脉冲（SWFSYNC =1）

10	HRMSTEP 在使能自动转换时必须包含精确的 MEP 比例系数，即每个 EPWMCLK 粗步进的 MEP 步进数，该系数可以通过 SFO 库软件章节描述的 SFO()函数获得。
11	写入 TBPRDHR 寄存器可以控制高分辨率 PWM 周期

如果使能高分辨率周期模式，则 EPWMxSYNC 脉冲会给 PWM 带来 ±1-2 个周期抖动，在向上计数模式下，为±1 TBCLK 周期的抖动，在中心对齐计数模式下，为±2 TBCLK 周期的抖动。所以不能将 EPWMx_TBCTR=0 或 EPWMx_TBCTR= CMPB 设置为同步输出信号源。当 EPWMx_TBCTL=0 时，软件同步脉冲只能在高分辨率周期初始化时发出一次；在 PWM 运行时应用同步脉冲会导致输出抖动。

若仅在 EPWMxA 或 EPWMxB 上使能高分辨率 PWM 周期控制，在向上计数模式下，非高分辨率输出具有±1 TBCLK 周期的抖动，在中心对齐计数模式下，具有±2 TBCLK 周期的抖动。

限制 MEP 占空比小节中描述的占空比换算程序也是用与高分辨率 PWM 周期。

假设条件：

- TBCLK=10ns（100MHz）
- FPWM=1.75MHz
- 周期为 571.482
- 每个粗步进的 MEP 步进为 180ps 时，MEP 比例系数为 55
- 用于将 TBPRDHR 保持在 1-255 之间和小数舍入的常数为 0.5（默认值）

问题：

在向上计数模式下：

- 如果 EPWMx_TBPRD = 571，则 PWM 频率 = 174.82kHz (周期 = (571+1) * T_{TBCLK})。
- 如果 EPWMx_TBPRD = 570，则 PWM 频率 = 175.13kHz (周期 = (570+1) * T_{TBCLK})。

在向上-向下计数模式下：

- 如果 EPWMx_TBPRD = 286，则 PWM 频率 = 174.82kHz (周期 = (286*2) * T_{TBCLK})。
- 如果 EPWMx_TBPRD = 285，则 PWM 频率 = 175.44kHz (周期 = (285*2) * T_{TBCLK})。

解决方案（在 180ps 下，每粗步进有 55 HRP 步进时）：

（1）EPWMx_TBPRD 寄存器的百分比整数周期值转换

- 整数周期值= int（PWM 周期）* TTBCLK =571* TTBCLK
- 向上计数模式：EPWMx_TBPRD=周期值-1=570=0x23A
- 中心对齐计数模式：EPWMx_TBPRD=周期值/2=285=0x11D

（2） TBPRDHR 寄存器小数值转换

- 中心对齐计数模式：TBPRDHR 寄存器值 = frac（PWM 周期 * MEP 比例系数+0.5）
  - 当使能自动转换且 HRMSTEP = MEP 比例系数=55 时，TBPRDHR 寄存器值 = frac（PWM 周期）<<8，将值移动到 TBPRDHR 的高字节，即 TBPRDHR 寄存器值 = frac（571.428）<<8=0x6D
  - 自动转换将自动执行计算，从而 TBPRDHR MEP 延迟按硬件缩放到=[（TBPRDHR[15:0]>>8）

Puya Confidential

*HRMSTEP+0x80]<<8=0x6D*55+0x80=0x17EB>>8

- 周期 MEP 延迟=0x17 MEP 步进

- 中心对齐计数模式：EPWMx_TBPRD=frac（PWM 周期 * MEP 比例系数+0.5）
  - 当使能自动转换且 HRMSTEP = MEP 比例系数=55 时，TBPRDHR 寄存器值 = frac（PWM 周期/2) <<8，将值移动到 TBPRDHR 的高字节，即 TBPRDHR 寄存器值 = frac (285.714) <<8=0x86
  - 自动转换将自动执行计算，从而 TBPRDHR MEP 延迟按硬件缩放到=[（TBPRDHR[15:0]>>8）*HRMSTEP+0x80]<<8=0x86*55+0x80=0x279A>>8
  - 周期 MEP 延迟=0x27 MEP 步进

18.3.10.8 高分辨率死区

配置高分辨率死区上升沿延迟和下降沿延迟

如果将 PWM 配置为提供给定频率、极性的常规 PWM，并在半周期时钟模式下使能死区，则通过在该 EPWM 模块寄存器空间中对 EPWMx_HRCNFG2 寄存器进行编程，即可使能 DBRED 和 DBFED 的高分辨率操作。提供两种配置选项：

- 控制模式
  - 在高分辨率模式下，选择 DBRED 和 DBFED 在活动寄存器中加载影子值的时间事件
  - 选择脉冲以匹配 EPWMx_DBCTL[LOADREDMODE，LOADFEDMODE]位中的选择
- 边沿模式
  - MEP 可通过编程实现对 DBRED、DBFED 或同时对死区的双边沿，即 DBRED 信号上升沿和 DBFED 信号下降沿，进行精确的位置控制

高分辨率死区操作

在向上模式下，死区模块在使能任何高分辨率模式情况下都不可以使用。

假设条件：

- TBCLK=10 ns（100 MHz）
- 在半周期模式下，使能死区：TBCLK=EPWMCLK
- FPWM=1.33MHz
- 占空比为 50%
- 死区上升沿延迟为占空比之外 5%
- 以 ns 为单位时，死区上升沿延迟 = 0.05*375=18.75ns

注意：与使用 HRPWM 时占空比限制一样，高分辨率死区必须在 DBRED 和 DBFED 的值大于 3 时才能使用。

死区延迟值是 DBRED 和 DBFED 的函数：

- 当使能半周期时钟时，RED 和 FED 计算公式变为：
  - RED= TBCLK/2 * DBRED
  - FED=TBCLK/2 * DBFED

Puya Confidential

DBRED 和 DBFED 计数值：
以 ns 为单位时，DBRED=18.75ns
所需 DBRED = RED/（TBCLK/2）=3.75

解决方案（每个粗步进的 MEP 步进为 180ps 时）：
DBRED 寄存器的整数死区值转换
整数 DBRED 值= int [RED/（TBCLK/2）]=3
DBREDHR 寄存器的小数值转换
DBREDHR 值 = frac[（所需 DBRED）*MEP 比例系数+0.5]<<8=frac（3.75）*55+0.5<<8 = 41.75*256 = 0x29C MEP 步进，硬件会忽略上述计算得到的低 9 位 DBREDHR 值

注意：当 AUTOCONV 置位且 MEP 比例系数在 HRMSTEP 寄存器中时，DBREDHR：DBRED =frac（所需 DBRED）<<8。由硬件自动完成剩余的转换计算，PWM 输出通道会出现正确的 MEP 缩放后的信号边沿。当 AUTOCONV 未置位时，必须由软件实现上述计算。

18.3.10.9 比例系数优化软件

MEP 逻辑能够在 255 个离散时间点之一上放置边沿。这些步进的尺寸约为 130ps 数量级，器件的典型 MEP 步进请参见数据手册。MEP 步进受最坏情况的工艺参数、工作温度和电压的影响，MEP 步进随电压升高、温度降低而减小，反之则增大。MEP 步进根据 HRPWM 特性的应用程序可以使用 PUYA 提供的 MEP 比例系数软件函数。在 EPWM 运行时，SFO 函数有助于每个 EPWMCLK 周期的步进计算。

EPWM 模块内置自检和诊断功能,以找到最佳 MEP 比例系数。PUYA 还提供了一个 C 可调用的 SFO 库，帮助确定最佳比例系数。

有关 SFO 库软件的详细信息，参考 EPWM 软件应用手册。

18.3.11 寄存器保护机制描述

为了保护关键的 PWM 寄存器，避免在代码失控时被意外损坏，增加了寄存器锁保护机制。在失控代码发生时，保护关键 PWM 寄存器不被意外写操作损坏。EPWM 根据不同的寄存器域，设置的多种不同的 LOCK：
当 EPWMx_EPWMLOCK 寄存器中 KEY 位域的值设置为 0xA5A5 时，才能成功写入 EPWMx_EPWMLOCK 寄存器，详情请参见 EPWMx_EPWMLOCK 寄存器描述。
EPWMx_EPWMLOCK 寄存器定义的锁定位以及相应的寄存器，如下所示：

- HRLOCK：锁定的寄存器为 EPWMx_HRCNFG、EPWMx_HRPWR、 EPWMx_HRMSTEP 和 EPWMx_HRPCTL
- GLLOCK：锁定的寄存器为 EPWMx_GLDCTL 和 EPWMx_GLDCFG
- TZCFGLOCK：锁定的寄存器为 EPWMx_TZSEL、EPWMx_TZDCSEL、EPWMx_TZCTL、 EPWMx_TZCTL2、EPWMx_TZCTLDCx 和 EPWMx_TZEINT

Puya Confidential

- TZCLRLOCK: 锁定的寄存器为 EPWMx_TZCLR、 EPWMx_TZCBCCLR、 EPWMx_TZOSTCLR 和 EPWMx_TZFRC
- DCLOCK: 锁定的寄存器为 EPWMx_DCTRIPSEL、EPWMx_DCxCTL、 EPWMx_DCFCTL、 EPWMx_DCCAPCTL、 EPWMx_DCxHTRIPSEL 和 EPWMx_DCxLTRIPSEL

当 EPWM_EALLOW 中 KEY 设置为 0x5a5a 时，才能写 EALLOW。EALLOW 为 1 时才能对相关的寄存器进行写操作，为 0 时不可写入。

- EALLOW：锁定全部的 EPWM 共用寄存器（以“EPWM_”为前缀）和部分 EPWM 模块级寄存器（以“EPWMx_”为前缀）。

INPUTLOCKx 位为 1 时代表锁定，不可写入。一旦 INPUTLOCKx 为 1，则必须通过复位才能清 0。EPWM_IXBSEL_LOCK[FILTERLOCK]同样。

- INPUTLOCKx：锁定的位域为 INPUTSELx。
- FILTERLOCK：锁定的寄存器为 EPWM_IXBAQSEL 和 EPWM_IXBCTRL。

当 EPWM_EPXBLOCK 中 KEY 设置为 0x5a5a 时，才能写 EPXBLOCK。EPXBLOCK 为 1 时才能对相关的寄存器进行写操作。一旦 EPXBLOCK 为 1，则必须通过复位才能清 0。

- EPXBLOCK：锁定的寄存器为 EPWM_EPXBx_0，EPWM_EPXBx_1，EPWM_EPXBx_MUX，EPWM_EPXB_POL。

在 EPWM_SYNCSOCLOCK 寄存器中有两个位域：ADCSOCLOCK 和 SYNCILOCK，分别锁定 ADCSOC 和 EPWM_SYNCI 相关寄存器。

- ADCSOCLOCK：锁定的寄存器为 EPWM_ADCOUT。
- SYNCILOCK：锁定的寄存器为 EPWM_SYNCI_SEL1_4 和 EPWM_SYNCI_SEL5_6。

注意：在含有 KEY 位域的寄存器中，当 KEY 匹配时，如果对该寄存器进行 16 位写操作，无论是写入该寄存器的上半部分或下半部分的 16 位，则都会被忽略，只有 32 位写操作才能成功。

详细寄存器与保护锁的关系如下表

表 18-30 模块级寄存器与保护锁

寄存器名称	偏移地址	保护锁
EPWMx_TBCTL	0x00	
EPWMx_TBCTL2	0x02	
EPWMx_TBCTR	0x08	
EPWMx_TBSTS	0x0A	
EPWMx_CMPCTL	0x10	
EPWMx_CMPCTL2	0x12	
EPWMx_DBCTL	0x18	
EPWMx_DBCTL2	0x1A	
EPWMx_AQCTL	0x20	
EPWMx_AQTSRCSEL	0x22	
EPWMx_PCCTL	0x28	
EPWMx_VCAPCTL	0x30	
EPWMx_VCNTCFG	0x32	

Puya Confidential

EPWMx_HRCNFG	0x40	EALLOW，HRLOCK
EPWMx_HRPWR	0x42	EALLOW，HRLOCK
EPWMx_HRMSTEP	0x4C	EALLOW，HRLOCK
EPWMx_HRCNFG2	0x4E	EALLOW，HRLOCK
EPWMx_HRPCTL	0x5A	EALLOW，HRLOCK
EPWMx_TRREM	0x5C	EALLOW，HRLOCK
EPWMx_GLDCTL	0x68	EALLOW，GLLOCK
EPWMx_GLDCFG	0x6A	EALLOW，GLLOCK
EPWMx_EPWMXLINK	0x70	
EPWMx_AQCTLA	0x80	
EPWMx_AQCTLA2	0x82	
EPWMx_AQCTLB	0x84	
EPWMx_AQCTLB2	0x86	
EPWMx_AQSFRC	0x8E	
EPWMx_AQCSFRC	0x92	
EPWMx_DBREDHR	0xA0	
EPWMx_DBRED	0xA2	
EPWMx_DBFEDHR	0xA4	
EPWMx_DBFED	0xA6	
EPWMx_TBPHS	0xC0	
EPWMx_TBPRDHR	0xC4	
EPWMx_TBPRD	0xC6	
EPWMx_CMPA	0xD4	
EPWMx_CMPB	0xD8	
EPWMx_CMPC	0xDE	
EPWMx_CMPD	0xE2	
EPWMx_GLDCTL2	0xE8	
EPWMx_SWVDELVAL	0xEE	
EPWMx_TZSEL	0x100	EALLOW，TZCFGLOCK
EPWMx_TZDCSEL	0x104	EALLOW，TZCFGLOCK
EPWMx_TZCTL	0x108	EALLOW，TZCFGLOCK
EPWMx_TZCTL2	0x10A	EALLOW，TZCFGLOCK
EPWMx_TZCTLDCA	0x10C	EALLOW，TZCFGLOCK
EPWMx_TZCTLDCB	0x10E	EALLOW，TZCFGLOCK
EPWMx_TZEINT	0x11A	EALLOW，TZCFGLOCK
EPWMx_TZFLG	0x126	
EPWMx_TZCBCFLG	0x128	
EPWMx_TZOSTFLG	0x12A	
EPWMx_TZCLR	0x12E	EALLOW，TZCLRLOCK
EPWMx_TZCBCCLR	0x130	EALLOW，TZCLRLOCK

Puya Confidential

EPWMx_TZOSTCLR	0x132	EALLOW，TZCLRLOCK
EPWMx_TZFRC	0x136	EALLOW，TZCLRLOCK
EPWMx_ETSEL	0x148	
EPWMx_ETPS	0x14C	
EPWMx_ETFLG	0x150	
EPWMx_ETCLR	0x154	
EPWMx_ETFRC	0x158	
EPWMx_ETINTPS	0x15C	
EPWMx_ETSOCPS	0x160	
EPWMx_ETCNTINITCTL	0x164	
EPWMx_ETCNTINIT	0x168	
EPWMx_DCTRIPSEL	0x180	EALLOW，DCLOCK
EPWMx_DCACTL	0x186	EALLOW，DCLOCK
EPWMx_DCBCTL	0x188	EALLOW，DCLOCK
EPWMx_DCFCTL	0x18E	EALLOW，DCLOCK
EPWMx_DCCAPCTL	0x190	EALLOW，DCLOCK
EPWMx_DCFOFFSET	0x192	
EPWMx_DCFOFFSETCNT	0x194	
EPWMx_DCFWINDOW	0x196	
EPWMx_DCFWINDOWCNT	0x198	
EPWMx_DCCAP	0x19E	
EPWMx_DCAHTRIPSEL	0x1A4	EALLOW，DCLOCK
EPWMx_DCALTRIPSEL	0x1A6	EALLOW，DCLOCK
EPWMx_DCBHTRIPSEL	0x1A8	EALLOW，DCLOCK
EPWMx_DCBLTRIPSEL	0x1AA	EALLOW，DCLOCK
EPWMx_EPWMLOCK	0x1F4	
EPWMx_HWVDELVAL	0x1FA	
EPWMx_VCNTVAL	0x1FC	

表 18-31 共用寄存器和保护锁

寄存器名称	偏移地址	保护锁
EPWM_IXBSEL1_2	0x0000	EALLOW, INPUTLOCKx
EPWM_IXBSEL3_4	0x0004	EALLOW, INPUTLOCKx
EPWM_IXBSEL5_6	0x0008	EALLOW, INPUTLOCKx
EPWM_IXBSEL7_8	0x000C	EALLOW, INPUTLOCKx
EPWM_IXBSEL9_10	0x0010	EALLOW, INPUTLOCKx
EPWM_IXBSEL11_12	0x0014	EALLOW, INPUTLOCKx
EPWM_IXBSEL13_14	0x0018	EALLOW, INPUTLOCKx
EPWM_IXBAQSEL	0x001C	EALLOW, FILTERLOCK
EPWM_IXBSEL_LOCK	0x0020	EALLOW
EPWM_IXBCTRL	0x0024	EALLOW, FILTERLOCK

Puya Confidential

EPWM_EPXB4_0	0x0040	EALLOW, EPXBLOCK
EPWM_EPXB4_1	0x0044	EALLOW, EPXBLOCK
EPWM_EPXB5_0	0x0048	EALLOW, EPXBLOCK
EPWM_EPXB5_1	0x004C	EALLOW, EPXBLOCK
EPWM_EPXB7_0	0x0050	EALLOW, EPXBLOCK
EPWM_EPXB7_1	0x0054	EALLOW, EPXBLOCK
EPWM_EPXB8_0	0x0058	EALLOW, EPXBLOCK
EPWM_EPXB8_1	0x005C	EALLOW, EPXBLOCK
EPWM_EPXB9_0	0x0060	EALLOW, EPXBLOCK
EPWM_EPXB9_1	0x0064	EALLOW, EPXBLOCK
EPWM_EPXB10_0	0x0068	EALLOW, EPXBLOCK
EPWM_EPXB10_1	0x006C	EALLOW, EPXBLOCK
EPWM_EPXB11_0	0x0070	EALLOW, EPXBLOCK
EPWM_EPXB11_1	0x0074	EALLOW, EPXBLOCK
EPWM_EPXB12_0	0x0078	EALLOW, EPXBLOCK
EPWM_EPXB12_1	0x007C	EALLOW, EPXBLOCK
EPWM_EPXB4_MUX	0x0080	EALLOW, EPXBLOCK
EPWM_EPXB5_MUX	0x0084	EALLOW, EPXBLOCK
EPWM_EPXB7_MUX	0x0088	EALLOW, EPXBLOCK
EPWM_EPXB8_MUX	0x008C	EALLOW, EPXBLOCK
EPWM_EPXB9_MUX	0x0090	EALLOW, EPXBLOCK
EPWM_EPXB10_MUX	0x0094	EALLOW, EPXBLOCK
EPWM_EPXB11_MUX	0x0098	EALLOW, EPXBLOCK
EPWM_EPXB12_MUX	0x009C	EALLOW, EPXBLOCK
EPWM_EPXB_POL	0x00A0	EALLOW, EPXBLOCK
EPWM_EPXBLOCK	0x00A4	EALLOW
EPWM_SYNCI_SEL1_4	0x00C0	EALLOW, SYNCILOCK
EPWM_SYNCI_SEL5_6	0x00C4	EALLOW, SYNCILOCK
EPWM_SYNCO_SEL1_4	0x00C8	EALLOW
EPWM_SYNCO_SEL5_6	0x00CC	EALLOW
EPWM_ADCOUT	0x00d0	EALLOW , ADCSOCLOCK
EPWM_SCOUT_CFG	0x00d4	EALLOW
EPWM_SYNCSOCLOCK	0x00d8	EALLOW
EPWM_EALLOW	0x00dC	
EPWM_CLKEN	0x00e0	EALLOW
EPWM_XBARFLG1	0x00f0	
EPWM_XBARFLG2	0x00f4	
EPWM_XBARCLR1	0x00f8	
EPWM_XBARCLR2	0x00fc	

Puya Confidential

18.4 EPWMx 寄存器描述

寄存器组地址

表 18-32 寄存器组地址

Device Registers	Start Address	End Address
EPWM1Regs	0x4001_8000	0x4001_83FF
EPWM2Regs	0x4001_8400	0x4001_87FF
EPWM3Regs	0x4001_8800	0x4001_8BFF
EPWM4Regs	0x4001_8C00	0x4001_8FFF
EPWM5Regs	0x4001_9000	0x4001_93FF
EPWM6Regs	0x4001_9400	0x4001_97FF
EPWMRegs	0x4001_9800	0x4001_9BFF

注意：寄存器的位宽有 32 位和 16 位区分，建议 32 位寄存器使用 32 位读/写操作，建议 16 位寄存器使用 16 位读/写操作，否则不能确保读/写正确。

注意：EPWM 模块的寄存器均带有“EPWMx_”前缀，但为了使寄存器描述更简洁，在 Function 中的寄存器描述忽略前缀。

18.4.1 时基控制寄存器（EPWMx_TBCTL）（x=1,2,3,4,5,6）

偏移地址：0x00

复位值：0x0083

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE_SOFT[1:0]		PHSDIR	CLKDIV[2:0]			HSPCLKDIV[2:0]			SWFSYNC	SYNCOSEL[1:0]		PRDLD	PHSEN	CTRMODE[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RS	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:14]	FREE_SOFT	RW	2'b0	配置仿真模式（Emulation Mode Configure）这些位域配置仿真事件期间 EPWM 时基计数器的行为。 00：时基计数器在下一次递增或递减后停止 01：时基计数器在完成整个循环后停止（1）向上计数模式：TBCTR 与周期相等时停止（2）向下计数模式：TBCTR=0x00 时停止（3）中心对齐计数模式：TBCTR=0x00 时停止 1x：自由运行
[13]	PHSDIR	RW	1'b0	配置计数方向（Count Direction Configure）该位表示在同步事件发生且从相位（TBPHS）寄存器加载新的相位值后，时基计数器（TBCTR）的计数方向，与 TBCTR 在同步事件前的方向无关。该位仅适用于 TBCTR 配置为中心对齐计数模式，故该位在向上计数模式和向下计数模式中会被忽略。 0：向下计数（同步事件后） 1：向上计数（同步事件后）
[12:10]	CLKDIV[2:0]	RW	3'b0	配置时基时钟预分频系数（Time Base Clock Pre-Scale Factor Configure） TBCLK = EPWMCLK / (HSPCLKDIV * CLKDIV) 000：分频系数为 1（默认） 001：分频系数为 2 010：分频系数为 4 011：分频系数为 8

Puya Confidential

				100：分频系数为 16 101：分频系数为 32 110：分频系数为 64 111：分频系数为 128
[9:7]	HSPCLKDIV	RW	3'b1	高速时基时钟预分频系数（High Speed Time Base Clock Pre-Scale Factor） TBCLK = EPWMCLK / (HSPCLKDIV * CLKDIV) 000：分频系数为 1 001：分频系数为 2（默认） 010：分频系数为 4 011：分频系数为 6 100：分频系数为 8 101：分频系数为 10 110：分频系数为 12 111：分频系数为 14
[6]	SWFSYNC	RS	1'b0	软件强制同步脉冲（Sync Pulse Forced by Software） 该位仅在 SYNCOSEL=00 时才会影响到 EPWMx 同步输出。 0：无效，读取返回 0 1：强制产生一个同步脉冲
[5:4]	SYNCOSEL	RW	2'b0	选择同步输出信号源（Sync Output Source Select） 00：EPWMx 同步输入或 SWFSYNC 01：TBCTR =0 10：TBCTR =CMPB 11：由 SYNCOSELX 位决定 EPWMx 同步输出
[3]	PRDLD	RW	1'b0	选择时基周期装载模式（Time-Base Period Load Mode Select） 选择 TBPRD 寄存器装载模式（TBPRD register Load Mode Select） 0：影子模式。当由 PRDLDSYNC 决定的 TBCTR =0 且/或同步事件时， TBPRD 寄存器装载影子寄存器的值，读写 TBPRD 寄存器实际访问的是影子寄存器 1：立即模式。立即绕过影子寄存器，读写 TBPRD 寄存器实际访问的是活动寄存器
[2]	PHSEN	RW	1'b0	使能相位（Phase Enable） 使能 TBPOF 寄存器的值装载 TBCNT（The value of TBPOF register to load TBCNT Enable） 0：禁止 1：使能，并允许 TBPRD 寄存器装载影子寄存器同步事件（出现软件强制同步信号或 EPWMx 同步输入信号时）
[1:0]	CTRMODE	RW	2'b11	选择计数器模式（Counter Mode Select） TBCTR 在正常工作后不会更改工作模式。更改 TBCTR 工作模式后，将会在下一个 TBCLK 边沿生效且 TBCTR 值从更改前的值开始递增减。 00：向上计数模式 01：向下计数模式 10：中心对齐模式 11：冻结计数模式（默认）

18.4.2 时基控制寄存器 2（EPWMx_TBCTL2） （x=1,2,3,4,5,6）

偏移地址：0x02

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRDLDSYNC[1:0]		SYNCOSELX[1:0]		Res.	Res.	Res.	Res.	OSHTSYNC	OSHTSYNCMODE	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

RW	RW	RW	RW	-	-	-	-	RS	RW	-	-	-	-	-	-

Bit	Name	RW	ResetValue	Function
[15:14]	PRDLDSYNC	RW	2'b0	选择同步事件周期装载条件（Period Load Condition on SYNC event Select） 该位域仅在 PRDLD=0 时有效。满足以下条件时，TBPRD 影子寄存器的值被装载到活动寄存器： 00：仅当 TBCTR =0 时 01：当 TBCTR =0 或发生同步事件时 10：仅发生同步事件时 11：保留
[13:12]	SYNCOSELX	RW	2'b0	选择扩展同步输出信号（Extended SYNCOUT Select） 00：禁止 EPWMx 同步输出信号 01：同步输出信号为 CMPC 10：同步输出信号为 CMPD 11：保留
[11:8]	reserved		4'b0	保留
[7]	OSHTSYNC	RS	1'b0	单次同步脉冲（Oneshot Sync Pulse），读取返回 0 0：无效 1：允许通过一个同步脉冲
[6]	OSHTSYNCMODE	RW	1'b0	使能单次同步模式（Oneshot Sync Mode Enable） 0：禁止 1：使能
[5:0]	reserved		6'b0	保留

18.4.3 时基计数器寄存器（EPWMx_TBCTR）（x=1,2,3,4,5,6）

偏移地址：0x08

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBCTR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:0]	TBCTR[15:0]	RW	16'b0	时基计数器（Time Base Counter）

18.4.4 时基状态寄存器（EPWMx_TBSTS）（x=1,2,3,4,5,6）

偏移地址：0x0A

复位值： 0x0001

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTRMAX	SYNCI	CTRDIR
-	-	-	-	-	-	-	-	-	-	-	-	-	RC_W1	RC_W1	R

Bit	Name	RW	ResetValue	Function
[15:3]	reserved		13'b0	保留

Puya Confidential

[2]	CTRMAX	RC_W1	1'b0	时基计数器达到最大值标志（Time-base Counter Reached the Max Value Flag）写 1 清除锁存的事件。 0：TBCTR 从未达到最大值 1：TBCTR 已未达到最大值 0xFFFF
[1]	SYNCI	RC_W1	1'b0	外部同步事件标志，写 1 清除锁存的事件。 0：写 0 无效，没有外部同步事件发生 1：有外部同步事件发生
[0]	CTRDIR	R	1'b1	计数方向标志（Count Direction Flag） 0：正在进行向下计数 1：正在进行向上计数

18.4.5 计数器比较控制寄存器（EPWMx_CMPCTL）（x=1,2,3,4,5,6）

偏移地址：0x10

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	LOADBSYNC[1:0]		LOADASYNC[1:0]		SHDWBFULL	SHDWAFULL	Res.	SHDWBMODE	Res.	SHDWAMODE	LOADBMODE[1:0]		LOADAMODE[1:0]	
-	-	RW	RW	RW	RW	R	R	-	RW	-	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:14]	reserved		2'b0	保留
[13:12]	LOADBSYNC	RW	2'b0	配置 CMPB 装载条件（CMPB Load Condition Configure） 该位仅在 SHDWBMODE=0 的条件下才有效。在发生同步事件时，满足以下条件，可以将 CMPB 的值从影子寄存器装载到活动寄存器中： 00：在设置 LOADBMODE 位时装载 01：在设置 LOADBMODE 位和发生同步事件时装载 10：仅接收到同步事件时装载 11：保留
[11:10]	LOADASYNC	RW	2'b0	配置 CMPA 装载条件（CMPA Load Condition Configure） 该位只有在 SHDWBMODE=0 的情况下才有效。在发生同步事件时，满足以下条件，可以将 CMPA 的值从影子寄存器装载到活动寄存器中： 00：在设置 LOADAMODE 位时装载 01：在设置 LOADAMODE 位和发生同步事件时装载 10：仅接收到同步事件时装载 11：保留
[9]	SHDWBFULL	R	1'b0	CMPB 影子满状态（CMPB Shadow Full Status） 在发生装载脉冲时，该位将自动清零。 0：CMPB 影子寄存器未满 1：CMPB 影子寄存器已满，进行 CPU 写操作将会覆盖当前影子值
[8]	SHDWAFULL	R	1'b0	CMPA 影子满状态（CMPA Shadow Full Status） 在发生装载脉冲时，该位将自动清零。该位设置条件： （1）如果对 CMPA 寄存器进行 32-位写操作或对 CMPA 寄存器进行 16-位写操作，则设置该位。 （2）如果对 CMPA 寄存器进行 16-位写操作时，则该位不会被设置。 0：CMPA 影子寄存器未满 1：CMPA 影子寄存器已满，进行 CPU 写操作将会覆盖当前影子值

Puya Confidential

[7]	reserved		1'b0	保留
[6]	SHDWBMODE	RW	1'b0	CMPB 装载模式（Counter-Compare B (CMPB) Load Mode） 0：影子模式。在该模式下，所有由 CPU 进行的写操作都会访问影子寄存器。CMPB 以双缓冲区方式工作。 1：立即模式。为了进行立即比较模式，所有写入和读取操作都可以直接访问活动寄存器。仅适用于活动比较寄存器 B。
[5]	reserved		1'b0	保留
[4]	SHDWAMODE	RW	1'b0	CMPA 装载模式（Counter-Compare A (CMPA) Load Mode） 0：影子模式。在该模式下，所有由 CPU 进行的写操作都会访问影子寄存器。CMPA 以及缓冲区方式工作。 1：立即模式。为了进行立即比较模式，所有写入和读取操作都可以直接访问活动寄存器。仅适用于活动比较寄存器 A。
[3:2]	LOADBMODE	RW	2'b0	选择 CMPB 装载事件（Counter-Compare B (CMPB) Load Event Select） 该位域在 SHDWBMODE=1 时无效。 满足以下条件时，将影子寄存器的值装载到活动 CMPB 寄存器中： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载操作（冻结）
[1:0]	LOADAMODE	RW	2'b0	选择 CMPA 装载事件（Counter-Compare A (CMPA) Load Event Select）该位域在 SHDWAMODE=1 时无效。 满足以下条件时，将影子寄存器的值装载到活动 CMPA 寄存器中： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载操作（冻结）

18.4.6 计数器比较控制寄存器 2（EPWMx_CMPCTL2）（x=1,2,3,4,5,6）

偏移地址：0x12

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	LOADDSYNC[1:0]		LOADCSYNC[1:0]		Res.	Res.	Res.	SHDWDMODE	Res.	SHDWCMODE	LOADDMODE[1:0]		LOADCMODE[1:0]	
-	-	RW	RW	RW	RW	-	-	-	RW	-	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:14]	reserved		2'b0	保留
[13:12]	LOADDSYNC	RW	2'b0	配置 CMPD 装载条件（CMPD Load Condition Configure） 该位只有在 SHDWDMODE=0 的情况下才有效。在发生同步事件时，满足以下条件，可以将 CMPD 的值从影子寄存器装载到活动寄存器中： 00：在设置 LOADDMODE 位装载 01：在设置 LOADDMODE 位和发生同步事件时装载 10：仅接收到同步事件时装载 11：保留
[11:10]	LOADCSYNC	RW	2'b0	配置 CMPC 装载条件（CMPC Load Condition Configure） 该位只有在 SHDWCMODE=0 的情况下才有效。在发生同步事件时，满足以下

Puya Confidential

				条件，可以将 CMPC 的值从影子寄存器装载到活动寄存器中： 00：在设置 LOADCMODE 位装载 01：在设置 LOADCMODE 位和发生同步事件时装载 10：仅接收到同步事件时装载 11：保留
[9:7]	reserved		3'b0	保留
[6]	SHDWDMODE	RW	1'b0	CMPD 装载模式（Counter-Compare D (CMPD) Load Mode） 0：影子模式。在该模式下，所有由 CPU 进行的写操作都会访问影子寄存器。CMPD 以双缓冲区方式工作。 1：立即模式。为了进行立即比较模式，所有写入和读取操作都可以直接访问活动寄存器。仅适用于活动比较寄存器 D。
[5]	reserved		1'b0	保留
[4]	SHDWCMODE	RW	1'b0	CMPC 装载模式（Counter-Compare C (CMPC) Load Mode） 0：影子模式。在该模式下，所有由 CPU 进行的写操作都会访问影子寄存器。CMPC 以双缓冲区方式工作。 1：立即模式。为了进行立即比较模式，所有写入和读取操作都可以直接访问活动寄存器。仅适用于活动比较寄存器 C。
[3:2]	LOADDMODE	RW	2'b0	选择 CMPD 装载事件（Counter-Compare D (CMPD) Load Event Select）该位域在立即模式下无效。满足以下条件时，将影子寄存器的值装载到活动 CMPD 寄存器中： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载操作（冻结）
[1:0]	LOADCMODE	RW	2'b0	选择 CMPC 装载事件（Counter-Compare C (CMPC) Load Event Select）该位域在立即模式下无效。满足以下条件时，将影子寄存器的值装载到活动 CMPC 寄存器中： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载操作（冻结）

18.4.7 死区发生器控制寄存器（EPWMx_DBCTL）（x=1,2,3,4,5,6）

偏移地址：0x18

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HALFCYCLE	DEDBMODE	OUTSWAP[1:0]		SHDWDBFEDMODE	SHDWDBREDMODE	LOADFEDMODE[1:0]		LOADREDMODE[1:0]		INMODE[1:0]		POLSEL[1:0]		OUTMODE[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15]	HALFCYCLE	RW	1'b0	使能半周期时钟（Half Cycle Clocking Enable） 0：使能全周期时钟，此时，死区计数器时钟为 TBCLK 1：使能半周期时钟，此时，死区计数器时钟为 2*TBCLK
[14]	DEDB_MODE	RW	1'b0	死区双边延时 B 模式控制（Dead Band Dual-Edge Delay B Mode Control） 该位控制 S8 开关。如果该位置位，用户则可

Puya Confidential

				以设置 OUT_MODE=01（使 A 路径等于 InA）或者设置 OUTSWAP=1x（使 EPWMxA 输出信号为 B 路径），否则 EPWMxA 无效。 0：上升沿延迟用于由 A 信号路径 DBCTL[4]位（S4 开关）选择的 InA/InB，下端延迟用于由 B 信号路径 DBCTL[5]位（S5 开关）选择的 InA/InB 1：上升沿延迟和下端延迟用于由 DBCTL[4]位（S4 开关）选择的输入源并输出到 B 信号路径
[13:12]	OUTSWAP	RW	2'b0	死区输出交换控制（Dead Band Output Swap Control） [12]: 控制 S6 开关。 [13]: 控制 S7 开关。 00：由 OUT_MODE 位域定义 EPWMxA 输出信号和 EPWMxB 输出信号 01：由 OUT_MODE 位域定义 EPWMxA 输出信号为 A 路径。由 OUT_MODE 位域定义 EPWMxB 输出信号为 A 路径，即等于 A 信号路径的上升沿延迟或延迟旁路。 10：由 OUT_MODE 位域定义 EPWMxA 输出信号为 B 路径，即等于 B 信号路径的下端延迟或延迟旁路。由 OUT_MODE 位域定义 EPWMxB 输出信号为 B 路径。 11：由 OUT_MODE 位域定义 EPWMxA 输出信号为 B 路径，即等于 B 信号路径的下端延迟或延迟旁路。由 OUT_MODE 位域定义 EPWMxB 输出信号为 A 路径，即等于 A 信号路径的上升沿延迟或延迟旁路。
[11]	SHDWDBFED_MODE	RW	1'b0	死区下降沿装载模式（FED Dead-Band Load Mode） 0：立即模式。为了立即进行下降沿延迟死区操作，所有通过 CPU 进行的写入或读取都可以直接访问活动寄存器。仅适用于活动 DBFED 寄存器。 1：影子模式。所有通过 CPU 的写操作都会访问影子寄存器。将以双缓冲区方式操作。
[10]	SHDWDBRED_MODE	RW	1'b0	死区上升沿装载模式（RED Dead-Band Load Mode） 0：立即模式。为了立即进行下降沿延迟死区操作，所有通过 CPU 进行的写入或读取都可以直接访问活动寄存器。仅适用于活动 DBRED 寄存器。 1：影子模式。所有通过 CPU 的写操作都会访问影子寄存器。将以双缓冲区方式操作。
[9:8]	LOADFEDMODE	RW	2'b0	选择下降沿装载事件（Falling Edge Load Event Select）该位或在立即模式下无效。满足以下条件时，将从影子寄存器装载到活动 DBFED 寄存器： 00：在计数器等于 0 时装载 01：在计数器等于周期时装载 10：在计数器等于 0 或计数器等于周期时装载 11：禁止装载（冻结模式）
[7:6]	LOADREDMODE	RW	2'b0	选择上升沿装载事件（Rising Edge Load Event Select）该位或在立即模式下无效。满足以下条件时，将从影子寄存器装载到活动 DBRED 寄存器： 00：在计数器等于 0 时装载

Puya Confidential

				01：在计数器等于周期时装载 10：在计数器等于 0 或计数器等于周期时装载 11：禁止装载（冻结模式）
[5:4]	IN_MODE	RW	2'b0	死区输入选择（Dead-Band Input Select）[4]：控制 S4 开关。[5]：控制 S5 开关。该位或可以选择上升沿延迟和下降沿延迟的输入源。默认上升沿延迟和下降沿延迟的输入源为 EPWMxA In 以便产生典型死区波形。 00：上升沿延迟和下降沿延迟的输入源为 EPWMxA In 01：下降沿延迟的输入源为 EPWMxA In，上升沿延迟输入源为 EPWMxB In 10：上升沿延迟输入源为 EPWMxA In，下降沿延迟的输入源为 EPWMxB In 11：上升沿延迟和下降沿延迟的输入源为 EPWMxB In
[3:2]	POLSEL	RW	2'b0	选择翻转的延迟信号（Invert Delayed Signal Select）[2]：控制 S2 开关。[3]：控制 S3 开关。通过该位域可以在 EPWMA 延迟信号或 EPWMB 延迟信号传输出死区发生器子模块前，对其进行翻转。OUT_MODE=11 和 IN_MODE=00 对应数字电动机控制逆变器的一个支路上发现的典型上下开关控制。其他增强模式不是典型使用模式。 00：高电平有效（AH）模式（默认）。未对 EPWMA 延迟信号和 EPWMB 延迟信号进行翻转。 01：低电平有效互补（ALC）模式。仅对 PWMA 延迟信号进行翻转。 10：高电平有效互补（AHC）模式。仅对 PWMB 延迟信号进行翻转。 11：低电平有效（AL）模式。都对 PWM 延迟信号 A 和 PWMB 延迟信号进行翻转。
[1:0]	OUT_MODE	RW	2'b0	死区输出选择（Dead-Band Output Select）[0]：控制 S0 开关。[1]：控制 S1 开关 00：忽略或完全禁止 DBM。在该情况下，POLSEL 位和 IN_MODE 位无效。 01：A 路径等于 A 信号路径的延迟旁路（InA），B 路径等于 B 信号路径中的下降沿延迟。 10：A 路径等于 A 信号路径中的上升沿延迟，B 路径等于 B 信号路径的延迟旁路（InB）。 11：完全使能 DBM，上升沿延迟和下降沿延迟处于活动状态。

18.4.8 死区发生器控制寄存器 2（EPWMx_DBCTL2）（x=1,2,3,4,5,6）

偏移地址：0x1A

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SHDWDBCTLMODE	LOADDBCTLMODE[1:0]	
-	-	-	-	-	-	-	-	-	-	-	-	-	RW	RW	RW

Puya Confidential

Bit	Name	RW	ResetValue	Function
[15:3]	reserved		13'b0	保留
[2]	SHDWDBCTLMODE	RW	1'b0	DBCTL 装载模式（DBCTL Load Mode） 0：立即模式。所有通过 CPU 写入或读取操作都可以直接访问活动寄存器。仅适用于活动 DBCTL 寄存器。 1：影子模式。除了对 DBCTL[5:0]的所有写入和读取操作都会传输到影子寄存器，其他位/域仍然访问活动寄存器。
[1:0]	LOADDBCTLMODE	RW	2'b0	选择 DBCTL 装载事件（DBCTL Load Event Select） 该位域在立即模式下无效。 00：在计数器等于 0 时装载 01：在计数器等于周期时装载 10：在计数器等于 0 或计数器等于周期时装载 11：禁止装载（冻结模式）

18.4.9 动作限定控制寄存器（EPWMx_AQCTL）（x=1,2,3,4,5,6）

偏移地址：0x20

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LDAQBSYNC[1:0]		LDAQASYNC[1:0]		Res.	SHDWAQBMODE	Res.	SHDWAQAMODE	LDAQBMODE[1:0]		LDAQAMODE[1:0]	
-	-	-	-	RW	RW	RW	RW	-	RW	-	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11:10]	LDAQBSYNC	RW	2'b0	配置 AQCTLB 装载条件（AQCTLB Load Condition Configure） 该位只有在 SHDWAQBMODE=1 的情况下才有效。 满足以下条件下，将控制 AQCTLB 寄存器在何时进行影子寄存器装载到活动寄存器操作： 00：在设置 LDAQBMODE 位时装载 01：在设置 LDAQBMODE 位和发生同步事件时装载 10：在接收到同步事件时装载 11：保留
[9:8]	LDAQASYNC	RW	2'b0	配置 AQCTLA 装载条件（AQCTLA Load Condition Configure） 该位只有在 SHDWAQAMODE=1 的情况下才有效。 满足以下条件下，将控制 AQCTLA 寄存器在何时进行影子寄存器装载到活动寄存器操作： 00：在设置 LDAQAMODE 位时装载 01：在设置 LDAQAMODE 位和发生同步事件时装载 10：在接收到同步事件时装载 11：保留
[7]	reserved		1'b0	保留
[6]	SHDWAQBMODE	RW	1'b0	动作模块 B 装载模式（Action Qualifier B Load Mode） 0：立即模式。在该模式下，所有写入或读取都可以直接访问活动寄存器。仅适用于活动动作限定寄存器。 1：影子模式。所有写操作由 CPU 通过访问影子寄存器进行。AQCTLB 寄存器用于双缓冲区操作。

Puya Confidential

[5]	reserved		1'b0	保留
[4]	SHDWAQAMODE	RW	1'b0	动作模块 A 装载模式（Action Qualifier A Load Mode） 0：立即模式。在该模式下，所有写入或读取都可以直接访问活动寄存器。仅适用于活动动作限定寄存器。 1：影子模式。所有写操作由 CPU 通过访问影子寄存器进行。AQCTLA 寄存器用于双缓冲区操作。
[3:2]	LDAQBMODE	RW	2'b0	选择动作模块 B 装载事件（Action Qualifier B Load Event Select） 该位域在立即模式下无效。满足以下条件时，将从影子寄存器装载到活动 AQCTLB 寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载（冻结模式）
[1:0]	LDAQAMODE	RW	2'b0	选择动作模块 A 装载事件（Action Qualifier A Load Event Select） 该位域在立即模式下无效。满足以下条件时，将从影子寄存器装载到活动 AQCTLA 寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载（冻结模式）

18.4.10 选择动作限定触发器事件源寄存器（EPWMx_AQTSRCSEL）（x=1,2,3,4,5,6）

偏移地址：0x22

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T2SEL[3:0]				T1SEL[3:0]			
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7:4]	T2SEL	RW	4'b0	配置 T2 事件源（T2 Event Source Configure） 0000：DCAEVT1 0001：DCAEVT2 0010：DCBEVT1 0011：DCBEVT2 0100：TZ1 0101：TZ2 0110：TZ3 0111：EPWMx 同步输入 1000：DCEVTFILT 其他：保留
[3:0]	T1SEL	RW	4'b0	配置 T1 事件源（T1 Event Source Configure） 0000：DCAEVT1 0001：DCAEVT2 0010：DCBEVT1

Puya Confidential

				0011：DCBEVT2 0100：TZ1 0101：TZ2 0110：TZ3 0111：EPWMX 同步输入 1000：DCEVTFILT 其他：保留

18.4.11 PWM 斩波控制寄存器（EPWMx_PCCTL）（x=1,2,3,4,5,6）

偏移地址：0x28

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CHPDUTY[2:0]			CHPFREQ[2:0]			OSHTWTH[3:0]				CHPEN
-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bits	Name	RW	ResetValue	Function
[15:11]	reserved		5'b0	保留
[10:8]	CHPDUTY	RW	3'b0	配置新波时钟周期占空比（Chopping Clock Duty Cycle Configure） 000：占空比为 12.5% 001：占空比为 25% 010：占空比为 37.5% 011：占空比为 50% 100：占空比为 62.5% 101：占空比为 75% 110：占空比为 87.5% 111：保留
[7:5]	CHPFREQ	RW	3'b0	斩波时钟分频（Chopping Clock Prescaler） 000：分频系数为 1，TBCLK=100 MHz 时，斩波时钟为 12.5 MHz 001：分频系数为 2，TBCLK=100 MHz 时，斩波时钟为 6.25 MHz 010：分频系数为 3，TBCLK=100 MHz 时，斩波时钟为 4.16 MHz 011：分频系数为 4，TBCLK=100 MHz 时，斩波时钟为 3.12 MHz 100：分频系数为 5，TBCLK=100 MHz 时，斩波时钟为 2.50 MHz 101：分频系数为 6，TBCLK=100 MHz 时，斩波时钟为 2.08 MHz 110：分频系数为 7，TBCLK=100 MHz 时，斩波时钟为 1.78 MHz 111：分频系数为 8，TBCLK=100 MHz 时，斩波时钟为 1.56 MHz
[4:1]	OSHTWTH	RW	4'b0	配置单次脉冲宽度（One-Shot Pulse Width Configure）0000：EPWMCLK / 8 宽度*1，EPWMCLK=100MHz 时，单次脉冲宽度为 80 ns 0001：EPWMCLK / 8 宽度*2，EPWMCLK=100MHz 时，单次脉冲宽度为 160 ns 0010：EPWMCLK / 8 宽度*3，EPWMCLK=100MHz 时，单次脉冲宽度为 240 ns 0011：EPWMCLK / 8 宽度*4，EPWMCLK=100MHz 时，单次脉冲宽度为 320 ns 0100：EPWMCLK / 8 宽度*5，EPWMCLK=100MHz 时，单次脉冲宽度为 400 ns

Puya Confidential

				0101：EPWMCLK / 8 宽度*6，EPWMCLK=100MHz 时，单次脉冲宽度为 480 ns 0110：EPWMCLK / 8 宽度*7，EPWMCLK=100MHz 时，单次脉冲宽度为 560 ns 0111：EPWMCLK / 8 宽度*8，EPWMCLK=100MHz 时，单次脉冲宽度为 640 ns 1000：EPWMCLK / 8 宽度*9，EPWMCLK=100MHz 时，单次脉冲宽度为 720 ns 1001：EPWMCLK / 8 宽度*10，EPWMCLK=100MHz 时，单次脉冲宽度为 800 ns 1010：EPWMCLK / 8 宽度*11，EPWMCLK=100MHz 时，单次脉冲宽度为 880 ns 1011：EPWMCLK / 8 宽度*12，EPWMCLK=100MHz 时，单次脉冲宽度为 960 ns 1100：EPWMCLK / 8 宽度*13，EPWMCLK=100MHz 时，单次脉冲宽度为 1040 ns 1101：EPWMCLK / 8 宽度*14，EPWMCLK=100MHz 时，单次脉冲宽度为 1120 ns 1110：EPWMCLK / 8 宽度*15，EPWMCLK=100MHz 时，单次脉冲宽度为 1200 ns 1111：EPWMCLK / 8 宽度*16，EPWMCLK=100MHz 时，单次脉冲宽度为 1280 ns
[0]	CHPEN	RW	1'b0	使能斩波功能（Chopping Function Enable） 0：禁止 1：使能

18.4.12 谷捕获控制寄存器（EPWMx_VCAPCTL）（x=1,2,3,4,5,6）

偏移地址：0x30

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	EDGEFILTDLYSEL	VDELAYDIV[2:0]			Res.	Res.	TRIGSEL[2:0]			VCAPSTART	VCAPE
-	-	-	-	-	RW	RW	RW	RW	-	-	RW	RW	RW	RW	RW

Bits	Name	RW	ResetValue	Function
[15:11]	reserved		5'b0	保留
[10]	EDGEFILTDLYSEL	RW	1'b0	边沿滤波器输出延迟（Edge Filter Output Delay） 0：无延迟 1：边沿滤波器输出 HWV/DELVAL 延迟
[9:7]	VDELAYDIV	RW	3'b0	配置各值延迟模式分频（Valley Delay Mode Divide Configure） 通过使用这些位域可以调整连续边沿捕获之间的延迟值。 000：硬件各延迟值 = 软件各延迟值 001：硬件各延迟值 = 容时基计数器 + 软件各延迟值 010：硬件各延迟值 = 容时基计数器右移 1 位 + 软件各延迟值 011：硬件各延迟值 = 容时基计数器右移 2 位 + 软件各延迟值

Puya Confidential

				100：硬件各延迟值 = 容时基计数器右移 4 位 + 软件各延迟值 其他：保留
[6:5]	reserved		2'b0	保留
[4:2]	TRIGSEL	RW	3'b0	选择捕获序列触发事件（Capture Sequence Trigger Event Select） 通过这些位域所选事件来触发谷捕获序列，如果发生所选事件，则捕获序列立即触发，且无论当前捕获状态都将进行重新触发捕获序列。 根据 DCFCTL[SRCSEL]位域所选事件进行捕获。 000：通过软件写入 TRIGSEL 位来触发捕获序列 001：TBCTR=0 事件 010：TBCTR=TBPRD 事件 011：TBCTR=0 或 TBCTR=TBPRD 事件 100：DCAEVT1 事件 101：DCAEVT2 事件 110：DCBEVT1 事件 111：DCBEVT2 事件 注意：TRIGSEL 位和 SRCSEL 位不能同时选择同一事件。
[1]	VCAPSTART	RW	1'b0	启动谷值捕获（Valley Capture Start） 在选择 TRIGSEL 位进行软件触发条件下，向该位写入 1 可以通过软件触发一个捕获序列。 0：无效 1：当 TRIGSEL=000 时，触发一次捕获序列
[0]	VCAPE	RW	1'b0	使能谷值捕获（Valley Capture Enable） 0：禁止 1：使能

18.4.13 各计数器配置寄存器（EPWMx_VCNTCFG）（x=1,2,3,4,5,6）

偏移地址：0x32

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOPEDGESTS	Res.	Res.	Res.	STOPEDGE[3:0]				STARTEDGESTS	Res.	Res.	Res.	STARTEDGE[3:0]			
R	-	-	-	RW	RW	RW	RW	R	-	-	-	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15]	STOPEDGESTS	R	1'b0	停止计数的边沿状态（Stop Count Edge Status）发生 TRIGSEL 位选择的触发事件时，该位进行复位。如果触发捕获序列且出现停止计数的边沿，则设置该位。 0：未出现停止计数的边沿 1：出现停止计数的边沿
[14:12]	reserved		3'b0	保留
[11:8]	STOPEDGE	RW	4'b0	选择停止计数的边沿（Stop Count Edge Select） 当激活计数器操作且出现 TRIGSEL 位选择的触发事件，通过这些位域选择各计数器停止计数的边沿。 0000：未停止计数 0001：第 1 个边沿停止计数 0010：第 2 个边沿停止计数 0011：第 3 个边沿停止计数

Puya Confidential

				... 1111：第 15 个边沿停止计数
[7]	STARTEDGESTS	R	1'b0	启动计数的边沿状态（Start Count Edge Status） 出现 TRIGSEL 位选择的触发事件时，该位进行复位。如果触发捕获序列且出现开始计数的边沿，则设置该位。 0：未出现开始计数的边沿 1：出现开始动计数的边沿
[6:4]	reserved		3'b0	保留
[3:0]	STARTEDGE	RW	4'b0	选择启动计数的边沿（Start Count Edge Select） 当激活计数器操作且出现 TRIGSEL 位选择的触发事件，通过这些位域选择各计数器开始计数的边沿。 0000：未开始计数 0001：第 1 个边沿开始计数 0010：第 2 个边沿开始计数 0011：第 3 个边沿开始计数 ... 1111：第 15 个边沿开始计数

18.4.14 EPWM 配置寄存器（EPWMx_HRCNFG）（x=1,2,3,4,5,6）

偏移地址：0x40

复位值： 0x0000

该寄存器仅适用于具有 EPWM 功能的 PWM 模块。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	HRLOADB[1:0]		CTLMODEB	EDGMODEB[1:0]		SWAPAB	AUTOCONV	SELOUTB	HRLOAD[1:0]		CTLMODE	EDGMODE[1:0]	
-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:13]	reserved		3'b0	保留
[12:11]	HRLOADB	RW	2'b0	选择 B 路信号影子值装载事件（EPWMxB Shadow Value Load Event Select） 该位域用于选择将 CMPBHR 的值装载到活动寄存器的 时间事件。 满足以下条件时，将 CMPBHR 的值装载到活动寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：保留
[10]	CTLMODEB	RW	1'b0	选择 B 路信号控制模式（EPWMxB Control Mode Select） 该位用于选择控制 B 路信号边沿位置的寄存器 0：周期或占空比控制模式（默认），即边沿位置由 TBPRDHR 寄存器或 CMPBHR 寄存器控制 1：相位控制模式，即边沿位置由 TBPHSHR 寄存器控制
[9:8]	EDGMODEB	RW	2'b0	选择 B 路信号边沿模式（EPWMxB Edge Mode Select） 该位域用于选择由 MEP 逻辑控制 B 路信号边沿。 00：禁用 EPWM 功能（默认） 01：MEP 控制 B 路信号的上升沿，此时，该上升沿由 CMPBHR 寄存器控制

Puya Confidential

				10：MEP 控制 B 路信号的下降沿，此时，该下降沿由 CMPBHR 寄存器控制 11：MEP 控制 B 路信号的上升沿和下降沿，此时，两 个边沿由 TBPRDHR 寄存器或 TBPHSHR 寄存器控制
[7]	SWAPAB	RW	1'b0	交换 A 路和 B 路输出信号（EPWM A & B Output Signals Swap） 该位可以交换 EPWMxA 和 EPWMxB 输出信号。 0：EPWMxA 和 EPWMxB 输出信号保持不变 1：交换 EPWMxA 和 EPWMxB 输出信号
[6]	AUTOCONV	RW	1'b0	使能自动转换（Auto Convert Enable） 该位用于选择将 TBPHSHR 寄存器、TBPRDHR 寄存 器或 CMPAHR 寄存器中周期、分数占空比或相位通过 软件计算进行手动缩放或通过 HRMSTEP 寄存器的 MEP 比例系数进行自动缩放。 SFO 库函数会根据实际需求，计算合适的 MEP 比例系 数并自动更新到 HRMSTEP 寄存器中。 在应用软件手动缩放分数占空比或相位时，必须禁用 自动转换。 0：禁止 1：使能
[5]	SELOUTB	RW	1'b0	配置 B 路输出信号反相（EPWMxB Output Invert Configure）该位配置 EPWMxB 通道的输出信号类型。在进行反相操作时，会考虑使用高分辨率模式，且反相后的 EPWMxB 信号将包含由高分辨率模式下的任何变化。 0：B 路输出正常信号 1：B 路输出与 A 路相反的信号
[4:3]	HRLOAD	RW	2'b0	选择 A 路信号影子值装载事件（EPWMxA Shadow Value Load Event Select）该位用于选择将 CMPAHR 的值装载到活动寄存器的时间事件。满足以下条件时，将 CMPAHR 的值装载到活动寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：保留
[2]	CTLMODE	RW	1'b0	选择 A 路信号控制模式（EPWMxA Control Mode Select）该位用于选择控制 A 路信号边沿位置的寄存器 0：周期或占空比控制模式（默认），即边沿位置由 TBPRDHR 寄存器或 CMPAHR 控制 1：相位控制模式，即边沿位置由 TBPHSHR 控制
[1:0]	EDGMODE	RW	2'b0	选择 A 路信号边沿模式（EPWMxA Edge Mode Select）该位用于选择由 MEP 逻辑控制 A 路信号边沿。 00：禁用 EPWM 功能（默认） 01：MEP 控制 A 路信号的上升沿，此时，该上升沿由 CMPAHR 控制 10：MEP 控制 A 路信号的下降沿，此时，该下降沿由 CMPAHR 控制 11：MEP 同时控制 A 路信号的上升沿和下降沿，此时，两个边沿分别由 TBPRDHR 寄存器或 TBPHSHR 控制

Puya Confidential

18.4.15 使能 MEP 校准逻辑寄存器（EPWMx_HRPWR）（x=1,2,3,4,5,6）

偏移地址：0x42

复位值： 0x0000

该寄存器仅适用于具有高分辨率功能的 EPWM 模块。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALPWRON	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
RW	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bits	Name	RW	ResetValue	Function
[15]	CALPWRON	RW	1'b0	使能 MEP 校准逻辑（MEPCalibration Logic Enable）该位仅适用于 EPWM1。可以通过禁止 MEP 校准逻辑来降低功耗。 0：禁止 1：使能
[14:0]	reserved		15'b0	保留

18.4.16 MEP 步骤数寄存器（EPWMx_HRMSTEP）（x=1,2,3,4,5,6）

偏移地址：0x4C

复位值： 0x0000

该寄存器仅允许 16 位访问且仅适用于具有高分辨率功能的 EPWM 模块。在 32 位访问模式下会导致调试器出现错误。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HRMSTEP[7:0]							
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bits	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7:0]	HRMSTEP	RW	8'b0	MEP 步进数（number of MEPsteps）在使能自动转换条件下，该位域自动将 CMPAHR 寄存器、 CMPBHR 寄存器、DBFEDHR 寄存器、DBREDHR 寄存 器、TBPRDHR 寄存器或 TBPHSHR 寄存器中的值转换成 EPWM 输出上的缩放微边沿延迟。在每次校准结束时，SFO 校准软件会将适当的值写入该位 域。

18.4.17 EPWM 配置寄存器 2（EPWMx_HRCNFG2）（x=1,2,3,4,5,6）

偏移地址：0x4E

复位值： 0x0000

该寄存器仅允许 16 位访问且仅适用于具有高分辨率功能的 EPWM 模块。在 32 位访问模式下会导致调试器出现错误。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTLMODEDBFED[1:0]		CTLMODEDBRED[1:0]		EDGMODEDB[1:0]	
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:6]	reserved		10'b0	保留

Puya Confidential

[5:4]	CTLMODEDBFED	RW	2'b0	选择 DBFEDHR 装载事件（DBFEDHR Load Event Select）该位用于选择将 DBFEDHR 的值装载到活动寄存器的时间事件。满足以下条件时，将 DBFEDHR 的值装载到活动寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：保留
[3:2]	CTLMODEDBRED	RW	2'b0	选择 DBREDHR 装载事件（DBREDHR Load Event Select）该位用于选择将 DBREDHR 的值装载到活动寄存器的时间事件。满足以下条件时，将 DBREDHR 的值装载到活动寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：保留
[1:0]	EDGMODEDB	RW	2'b0	选择死区边沿模式（Dead Band Edge Mode Select）该位域用于选择由 MEP 逻辑控制死区边沿。 00：禁用 EPWM 功能（默认） 01：MEP 控制死区的上升沿，此时，该上升沿由 DBREDHR 寄存器控制 10：MEP 控制死区的下降沿，此时，该下降沿由 DBREDHR 寄存器控制 11：MEP 同时控制死区的上升沿和下降沿，此时，两个边沿分别由 DBREDHR 寄存器和 DBFEDHR 寄存器控制

18.4.18 高分辨率周期控制寄存器（EPWMx_HRPCTL）（x=1,2,3,4,5,6）

偏移地址：0x5A

复位值： 0x0000

该寄存器仅适用于具有 EPWM 功能的 PWM 模块。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWMSYNCSELX[2:0]			Res.	TBPHSHRLOADE	PWMSYNCSEL	HRPE
-	-	-	-	-	-	-	-	-	RW	RW	RW	-	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:7]	reserved		7'b0	保留
[6:4]	PWMSYNCSELX	RW	3'b0	选择扩展的信号（Extended Signal Select） 000：通过 PWMSYNCSEL 位定义 EPWMSYNCPER 信号（默认） 100：计数器的值等于 CMPC 寄存器的值并且向上计数 101：计数器的值等于 CMPC 寄存器的值并且向下计数 110：计数器的值等于 CMPD 寄存器的值并且向上计数 111：计数器的值等于 CMPD 寄存器的值并且向下计数 其他：保留
[3]	reserved		1'b0	保留
[2]	TBPHSHRLOADE	RW	1'b0	使能高分辨率相位同步（High Resolution Phase Synchronization Enable） 该位可以通过同步输入、软件强制同步脉冲或数字比较事件进行 EPWM 模块的高分辨率相位同步，以实现控制多个相同频率的

Puya Confidential

				EPWM 模块进行高分辨率相位对齐。 TBCTL[PHSEN]位与 TBPHSHRLOADE 位可以 独立工作。如果要使用高分辨率相位和周期控制功能，则需要同时设置 TBCTL[PHSEN]位与 TBPHSHRLOADE 位。当高分辨率周期处于中心对齐计数模式时，如果 TBPHSHR 为 0，则 TBCTL[PHSEN]位与 TBPHSHRLOADE 位必须 置位。如果只使能高分辨率占空比，则不需要设置该位。 0：禁止 1：使能，此时，通过使用 TBPHSHR 寄存器内 容进行同步。
[1]	PWMSYNCSEL	RW	1'b0	选择 EPWMSYNCPER 源（EPWMSYNCPER Source Select）通过该位选择用于驱动 COMP 和 GPDAC 模块的 EPWMSYNCPER 信号源。 0：TBCTR=TBPRD 为 EPWMSYNCPER 信号源 1：TBCTR=0 为 EPWMSYNCPER 信号源
[0]	HRPE	RW	1'b0	使能高分辨率 PWM 周期（High Resolution Period Enable）如果需要使用 EPWM 周期功能则需要将 CTRMODE 位设置为除向下计数模式之外的模式。 0：禁止， 1：使能，EPWM 可以同时控制高分辨率的占空比和频率

18.4.19 EPWM 转换高分辨率余数寄存器（EPWMx_TRREM）（x=1,2,3,4,5,6）

偏移地址：0x5C

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TREEM[10:0]										
-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:11]	reserved		11'b0	保留
[10:0]	TREEM	RW	11'b0	转换器剩余部分(Translator Remainder Portion) ：这个 11 位值用于跟踪 EPWM 算法计算中的余数部分。 此值用于跟踪 EPWM 硬件计算中的余数部分。 注：1. 在 SYNCIN 或 TBCTL[SWFSYNC]事件或 DC 事件（如果已启用）发生时，TRREM 寄存器的低 8 位可自动用 TBPHSHR 值进行初始化。用户也可以通过 CPU 写入一个值。 2. TRREM 寄存器更新的优先级，最高优先级：同步（软件或硬件）TBPHSHR 复制到 TRREM；次高优先级：EPWM 硬件（更新 TRREM 寄存器）；最低优先级： CPU 写入 TRREM 寄存器。 3. 在非对称模式下，TRREM 寄存器的第 10 位未使用。该位可强制置零。 在向上计数模式和中心对齐模式下，TRREM 将分别被初始化为 0x0 和 0x100。 非对称模式：TRREM[7:0] = TBPHSHR[15:8]，且 TRREM[10,9,8] = 0,0,0；对称模式：TRREM[7:0] = TBPHSHR[15:8]，且 TRREM[10,9,8] = 0,0,1

Puya Confidential

18.4.20 全局 EPWM 装载控制寄存器（EPWMx_GLDCTL）（x=1,2,3,4,5,6）

偏移地址：0x68

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	GLDCNT[2:0]			GLDPRD[2:0]			Res.	OSHTMODE	GLDMODE[3:0]				GLD
-	-	-	R	R	R	RW	RW	RW	-	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:13]	reserved		3'b0	保留
[12:10]	GLDCNT	R	3'b0	选择全局装载事件数（The Number of Global Load Events Select） 该位或选择已发生的选定事件数量。 000：无事件发生 001：发生 1 个事件 010：发生 2 个事件 011：发生 3 个事件 100：发生 4 个事件 101：发生 5 个事件 110：发生 6 个事件 111：发生 7 个事件
[9:7]	GLDPRD	RW	3'b0	选择全局装载选通事件（Global Load Strobe Event Select） 该位或用于选择在发生特定数量的事件后，何时生成装载全局装载脉冲。 000：禁止计数器，不生成脉冲 001：在 GLDCNT=001 时生成脉冲 010：在 GLDCNT=010 时生成脉冲 011：在 GLDCNT=011 时生成脉冲 100：在 GLDCNT=100 时生成脉冲 101：在 GLDCNT=101 时生成脉冲 110：在 GLDCNT=110 时生成脉冲 111：在 GLDCNT=111 时生成脉冲
[6]	reserved		1'b0	保留
[5]	OSHTMODE	RW	1'b0	使能单次装载模式（One Shot Load Mode Enable）单次装载模式仅在 GLD=1 条件下使用。 0：禁止，所有选择的全局装载脉冲会连续进行影子寄存器到活动寄存器的装载行为 1：使能，直到 OSHTLD=1，否则所有全局装载脉冲一直被阻止
[4:1]	GLDMODE	RW	4'b0	选择全局装载脉冲（Global Load Pulse Select）该位或用于选择在影子寄存器到活动寄存器模式重装载模式中期间同时进行全局装载脉冲操作。 0000：在计数器等于 0 时进行装载 0001：在计数器等于周期时进行装载 0010：在计数器等于 0 时计数器等于周期时进行装载 0011：在 SYNCEVT 信号时进行装载，SYNCEVT 信号为 EPWMx 同步输入、软件强制同步脉冲。DCAEVT1.sync 和 DCBEVT1.sync 进行逻辑或操作

Puya Confidential

				0100：在 SYNCEVT 信号或计数器等于 0 时进行装载 0101：在 SYNCEVT 信号或计数器等于周期时进行装载 0110：在 SYNCEVT 信号、计数器等于 0 时计数器等于周期时进行装载 1111：在写入 GFROLD 位时进行装载其他：保留
[0]	GLD	RW	1'b0	选择全局装载事件（Global Load Event Select）该位用于控制影子寄存器到活动寄存器的装载行为。 0：所有影子寄存器装载行为按照各自指定的重新装载控制位进行 1：所有影子寄存器装载行为由 GLDMODE 位定义

18.4.21 全局 EPWM 装载事件配置寄存器（EPWMx_GLDCFG）（x=1,2,3,4,5,6）

偏移地址：0x6A

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	AQCSPRC	AQCTLB_AQCTLB2	AQCTLA_AQCTLA2	DBCTL	DBFED_DBFEDHR	DBRED_DBREDHR	CMPD	CMPC	CMPB_CMPBHR	CMPA_CMPAHR	TBPRD_TBPRDRHR
-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:11]	reserved		5'b0	保留
[10]	AQCSPRC	RW	1'b0	配置 AQCSPRC 全局装载事件（AQCSPRC Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[9]	AQCTLB_AQCTLB2	RW	1'b0	配置 AQCTLB_AQCTLB2 全局装载事件（AQCTLB_AQCTLB2 Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[8]	AQCTLA_AQCTLA2	RW	1'b0	配置 AQCTLA_AQCTLA2 全局装载事件（AQCTLA_AQCTLA2 Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[7]	DBCTL	RW	1'b0	配置 DBCTL 全局装载事件（DBCTL Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[6]	DBFED_DBFEDHR	RW	1'b0	配置 DBFED_DBFEDHR 全局装载事件（DBFED_DBFEDHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置

Puya Confidential

[5]	DBRED_DBREDHR	RW	1'b0	配置 DBRED_DBREDHR 全局装载事件（DBRED_DBREDHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[4]	CMPD	RW	1'b0	配置 CMPD 全局装载事件（CMPD Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[3]	CMPC	RW	1'b0	配置 CMPC 全局装载事件（CMPC Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[2]	CMPB_CMPBHR	RW	1'b0	配置 CMPB_CMPBHR 全局装载事件（CMPB_CMPBHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[1]	CMPA_CMPAHR	RW	1'b0	配置 CMPA_CMPAHR 全局装载事件（CMPA_CMPAHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置
[0]	TBPRD_TBPRDRHR	RW	1'b0	配置 TBPRD_TBPRDRHR 全局装载事件（TBPRD_TBPRDRHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置

18.4.22 EPWMx 链接寄存器（EPWMx_EPWMXLINK）（x=1,2,3,4,5,6）

偏移地址：0x70

复位值： 0x0000

该寄存器用于配置 EPWMx 模块之间的链接关系，且每个 EPWMx 模块的默认复位值不同。为了防止无意中链接模块，复位值将每个 EPWMx 模块链接到自身。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GLDCTL2LINK[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPD LINK[3:0]			
RW	RW	RW	RW	-	-	-	-	-	-	-	-	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPCLINK[3:0]				CMPBLINK[3:0]				CMPALINK[3:0]				TBPRDLINK[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	ResetValue	Function
[31:28]	GLDCTL2LINK	RW	4'b0	GLDCTL2 链接：该位域用于在所选的 EPWMx 模块在进行写操作，并根据以下选择，写入 GLDCTL2 寄存器同时写入当前 EPWMx 的 GLDCTL2 寄存器。0000=EPWM1， 0001=EPWM2， ...直到最后一个 EPWM 实例，其他=保留
[27:20]	reserved		8'b0	保留
[19:16]	CMPD LINK	RW	4'b0	CMPD 链接：该位域用于在所选的 EPWMx 模块在进行写操作，并根据以下选择，写入 CMPD 寄存器同时写入当前 EPWMx 的 CMPD 寄存器。 0000=EPWM1， 0001=EPWM2， ...直到最后一个 EPWM 实例， 其他=保留
[15:12]	CMPCLINK	RW	4'b0	CMPC 链接：该位域用于在所选的 EPWMx 模块在进行写操作，并根据以下选择，写入 CMPC 寄存器同时写入当前 EPWMx 的 CMPC 寄存器。 0000=EPWM1， 0001=EPWM2， ...直到最后一个 EPWM 实例， 其他 = 保留
[11:8]	CMPBLINK	RW	4'b0	CMPB_CMPBHR 链接：该位域用于在所选的 EPWMx 模块在进行写操作，并根据以下选择，写入 CMPB_CMPBHR 寄存器同时写入当前 EPWMx 的 CMPB 和 CMPBHR 寄存器。 0000=EPWM1， 0001=EPWM2， ...直到最后一个 EPWM 实例， 其他=保留
[7:4]	CMPALINK	RW	4'b0	CMPA_CMPAHR 链接：该位域用于在所选的 EPWMx 模块在进行写操作，并根据以下选择，写入 CMPA_CMPAHR 寄存器同时写入当前 EPWMx 的 CMPA 和 CMPAHR 寄存器。 0000=EPWM1， 0001=EPWM2， ...直到最后一个 EPWM 实例， 其他=保留
[3:0]	TBPRDLINK	RW	4'b0	TBPRD_TBPRDIR 链接：该位域用于在所选的 EPWMx 模块在进行写操作，并根据以下位选择，写入 TBPRD:TBPRDIR 寄存器同时写入当前 EPWMx 的 TBPRD_TBPRDIR 寄存器。 0000=EPWM1， 0001=EPWM2， ...直到最后一个 EPWM 实例， 其他=保留

Puya C

18.4.23 输出 A 的动作限定控制寄存器（EPWMx_AQCTLA）（x=1,2,3,4,5,6）

偏移地址：0x80

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CBD[1:0]		CBU[1:0]		CAD[1:0]		CAU[1:0]		PRD[1:0]		ZRO[1:0]	
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11:10]	CBD	RW	2'b0	向下计数时 TBCTR=CMPB 动作： 00 = 无操作， 01 = 清零（EPWMxA 低）， 10 = 设置（EPWMxA 高）， 11 = 翻转
[9:8]	CBU	RW	2'b0	向上计数时 TBCTR=CMPB 动作： 00 = 无操作， 01 = 清零， 10 = 设置， 11 = 翻转
[7:6]	CAD	RW	2'b0	向下计数时 TBCTR=CMPA 动作： 00 = 无操作， 01 = 清零， 10 = 设置， 11 = 翻转
[5:4]	CAU	RW	2'b0	向上计数时 TBCTR=CMPA 动作： 00 = 无操作， 01 = 清零， 10 = 设置， 11 = 翻转
[3:2]	PRD	RW	2'b0	TBCTR=TBPRD 时动作： 00 = 无操作， 01 = 清零， 10 = 设置， 11 = 翻转
[1:0]	ZRO	RW	2'b0	TBCTR=0 时动作： 00 = 无操作，01 = 清零， 10 = 设置， 11 = 翻转 中心对齐模式下 TBCTR=0 时方向为向上计数

18.4.24 输出 A 的附加动作限定控制寄存器（EPWMx_AQCTLA2）（x=1,2,3,4,5,6）

偏移地址：0x82

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T2D[1:0]		T2U[1:0]		T1D[1:0]		T1U[1:0]	

-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7:6]	T2D	RW	2'b0	向下计数时 T2 事件动作: 00 = 无操作, 01 = 清零（EPWMxA 低）, 10 = 设置（EPWMxA 高）, 11 = 翻转
[5:4]	T2U	RW	2'b0	向上计数时 T2 事件动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转
[3:2]	T1D	RW	2'b0	向下计数时 T1 事件动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转
[1:0]	T1U	RW	2'b0	向上计数时 T1 事件动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转; 中心对齐模式下 TBCTR=0 时方向为向上计数

18.4.25 输出 B 的动作限定控制寄存器（EPWMx_AQCTLB）（x=1,2,3,4,5,6）

偏移地址：0x84

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CBD[1:0]		CBU[1:0]		CAD[1:0]		CAU[1:0]		PRD[1:0]		ZRO[1:0]	
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved	-	4'b0	保留
[11:10]	CBD	RW	2'b0	向下计数时 TBCTR=CMPB 动作: 00 = 无操作, 01 = 清零（EPWMxB 低）, 10 = 设置（EPWMxB 高）, 11 = 翻转
[9:8]	CBU	RW	2'b0	向上计数时 TBCTR=CMPB 动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转

[7:6]	CAD	RW	2'b0	向下计数时 TBCTR=CMPA 动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转
[5:4]	CAU	RW	2'b0	向上计数时 TBCTR=CMPA 动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转
[3:2]	PRD	RW	2'b0	TBCTR=TBPRD 时动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转
[1:0]	ZRO	RW	2'b0	TBCTR=0 时动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转; 中心对齐模式下 TBCTR=0 时方向为向上计数

18.4.26 输出 B 的附加动作限定控制寄存器（EPWMx_AQCTLB2）（x=1,2,3,4,5,6）

偏移地址：0x86

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T2D[1:0]		T2U[1:0]		T1D[1:0]		T1U[1:0]	
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7:6]	T2D	RW	2'b0	向下计数时 T2 事件动作: 00 = 无操作, 01 = 清零（EPWMxB 低）, 10 = 设置（EPWMxB 高）, 11 = 翻转
[5:4]	T2U	RW	2'b0	向上计数时 T2 事件动作: 00 = 无操作, 01 = 清零, 10 = 设置, 11 = 翻转
[3:2]	T1D	RW	2'b0	向下计数时 T1 事件动作: 00 = 无操作, 01 = 清零,

				10 = 设置， 11 = 翻转
[1:0]	T1U	RW	2'b0	向上计数时 T1 事件动作： 00 = 无操作， 01 = 清零， 10 = 设置， 11 = 翻转； 中心对齐模式下 TBCTR=0 时方向为向上计数

18.4.27 动作限定软件强制寄存器（EPWMx_AQSFRC）（x=1,2,3,4,5,6）

偏移地址：0x8E

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLDCSF[1:0]		OTSFB	ACTSFB[1:0]		OTSFA	ACTSFA[1:0]	
-	-	-	-	-	-	-	-	RW	RW	RS	RW	RW	RS	RW	RW

Bit	Name	RW	ResetValue	Function
[15:8]	reserved	RO	8'b0	保留
[7:6]	RLDCSF	RW	2'b0	选择 AQCSPFRC 重装载模式： 00=在 TBCTR=0 时装载， 01=在 TBCTR=TBPRD 时装载， 10=在 TBCTR=0 或 TBCTR=TBPRD 时装载， 11=立即装载（不使用影子寄存器）
[5]	OTSFB	R/S	1'b0	在 B 路输出进行一次软件强制事件： 0=无效（读取返回 0）， 1=使能（自动清除）
[4:3]	ACTSFB	RW	2'b0	配置当发生一次软件强制事件后，B 路输出的动作： 00=无操作， 01=清除（低电平）， 10=设置（高电平）， 11=翻转
[2]	OTSFA	R/S	1'b0	在 A 路输出进行一次软件强制事件： 0=无效（读取返回 0）， 1=使能（自动清除）
[1:0]	ACTSFA	RW	2'b0	配置当发生一次软件强制事件后，A 路输出的动作： 00=无操作， 01=清除（低电平）， 10=设置（高电平）， 11=翻转

18.4.28 动作限定连续软件强制寄存器（EPWMx_AQCSFRC）（x=1,2,3,4,5,6）

偏移地址：0x92

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSFB[1:0]		CSFA[1:0]	
-	-	-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:4]	reserved		12'b0	保留
[3:2]	CSFB	RW	2'b0	配置 B 路输出连续软件强制：在立即模式下连续软件强制在下一 TBCLK 边沿生效。在影子模式下，连续软件强制影子加载到活动寄存器后的下一个 TBCLK 边沿生效。 00=禁止且无影响， 01=强制低电平， 10=强制高电平， 11=禁止且无影响
[1:0]	CSFA	RW	2'b0	配置 A 路输出连续软件强制：在立即模式下连续软件强制在下一 TBCLK 边沿生效。在影子模式下，连续软件强制影子加载到活动寄存器后的下一个 TBCLK 边沿生效。 00=禁止且无影响， 01=强制低电平， 10=强制高电平， 11=禁止且无影响

18.4.29 死区发生器上升沿延迟高分辨率寄存器（EPWMx_DBREDHR）（x=1,2,3,4,5,6）

偏移地址：0xA0

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBREDHR[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
RW	RW	RW	RW	RW	RW	RW	-	-	-	-	-	-	-	-	-

Bit	Name	RW	ResetValue	Function
[15:9]	DBREDHR	RW	7'b0	死区上升沿高分辨率延迟
[8:0]	reserved		9'b0	保留

18.4.30 死区发生器上升沿延迟寄存器（EPWMx_DBRED）（x=1,2,3,4,5,6）

偏移地址：0xA2

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DBRED[13:0]													
-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:14]	reserved		2'b0	保留
[13:0]	DBRED	RW	14'b0	死区上升沿延迟

18.4.31 死区发生器下降沿延迟高分辨率寄存器（EPWMx_DBFEDHR）（x=1,2,3,4,5,6）

偏移地址：0xA4

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBFEDHR[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	-	-	-	-	-	-	-	-	-

Bit	Name	RW	ResetValue	Function
[15:9]	DBFEDHR	RW	7'b0	死区下降沿高分辨率延迟
[8:0]	reserved		9'b0	保留

18.4.32 死区发生器下降沿延迟寄存器（EPWMx_DBFED）（x=1,2,3,4,5,6）

偏移地址：0xA6

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DBFED[13:0]													
-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:14]	reserved		2'b0	保留
[13:0]	DBFED	RW	14'b0	死区下降沿延迟

18.4.33 时基相位偏移寄存器（EPWMx_TBPHS）（x=1,2,3,4,5,6）

偏移地址：0xC0

复位值： 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBPHS[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHSHR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[31:16]	TBPHS	RW	16'b0	时基相位偏移：该寄存器用于调整 PWM 相位，即设置所选 EPWM 与提供同步输入信号的时基计数器之间的相位偏移量。如果 TBCTL[PHSEN]=0，则 TBCTR 不会装载相位值并且忽略同步事件。如果 TBCTL[PHSEN]=1，则在发生同步事件时（可以由软件强制同步或 EPWMx 同步输入信号产生），TBCTR 装载相位值。
[15:0]	TBPHSHR	RW	16'b0	高分辨率时基相位偏移：忽略 TBPHSHR 寄存器的低 8 位，忽略写入操作，读取操作返回 0。必须通过 TRREM 寄存器来模拟 TBPHSHR 功能，而不是使用 TBPHSHR。

18.4.34 高分辨率时基周期寄存器（EPWMx_TBPRDHR）（x=1,2,3,4,5,6）

偏移地址：0xC4

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPRDHR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Confidential

Bit	Name	RW	ResetValue	Function
[15:0]	TBPRDHR	RW	16'b0	高分辨率时基周期：该寄存器用于存储高分辨率时基周期值的高 8 位，且不受 PRDLD 位的影响。对 TBPRDHR 进行读取和写入操作都是对影子寄存器进行的。TBPRDHR 寄存器仅在高分辨率周期功能启用时使用。此寄存器仅适用于支持高分辨率周期控制的 EPWM 模块。 该寄存器的低 8 位被忽略——写入操作被忽略，读取操作返回零

18.4.35 时基周期寄存器（EPWMx_TBPRD）（x=1,2,3,4,5,6）

偏移地址：0xC6

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPRD[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:0]	TBPRD	RW	16'b0	时基周期：该寄存器确定 TBCTR 周期，从而设置 PWM 频率。该寄存器的影子寄存器可以通过 TBCTL[PRDLD]位来使能/禁止，在默认情况下，该寄存器用于使能影子寄存器。如果 TBCTL[PRDLD]=0，则使能影子寄存器，任何读取/写入操作都将自动进入影子寄存器。此时，在 TBCTR=0 时，活动寄存器从影子寄存器中装载。如果 TBCTL[PRDLD]=1，则禁止影子寄存器，任何读取/写入操作都将自动进入活动寄存器。活动寄存器和影子寄存器的内存映射地址相同。

18.4.36 计数比较模块 A 寄存器（EPWMx_CMPA）（x=1,2,3,4,5,6）

偏移地址：0xD4

复位值： 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMPA[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPAHR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[31:16]	CMPA	RW	16'b0	计数比较模块 A：CMPA 寄存器用于与 TBCTR 进行连续比较。如果两个值相等，则计数器比较模块就会生成一个 TBCTR 等于计数比较模块 A 事件。将该事件发送到动作限定进行限定和转换为一个或多个动作。AQCTLA 寄存器和 AQCTLB 寄存器的配置决定这些动作可以应用于 EPWMxA 输出信号或 EPWMxB 输出信号。AQCTLA 寄存器和 AQCTLB 寄存器可以定义的动作如下所示： 无操作：忽略 TBCTR 等于计数比较模块 A 事件 清除：设置 EPWMxA 或 EPWMxB 输出信号为低电平 设置：设置 EPWMxA 或 EPWMxB 输出信号为高电平 翻转：将 EPWMxA 或 EPWMxB 输出信号的状态进行翻转 通过 SHDWAMODE 位来使能 CMPA 影子寄存器，在默认情况下使能影子模式。当 SHDWAMODE=0 时，使能 CMPA 影子寄存器，此时，任

Confidential

				何写入或读取操作都会自动进入到影子寄存器。在该情况下，根据 LOADAMODE 位可以确定影子寄存器何时装载到活动寄存器中。当 SHDWAMODE=1 时，禁止 CMPA 影子寄存器，此时，任何写入或读取操作将直接进入到活动寄存器。在写入之前，可以根据 SHDWAFULL 位判断 CMPA 影子寄存器当前状态是否已满。在任何一种模式下，活动寄存器和影子寄存器的内存映射地址相同。
[15:0]	CMPAHR	RW	16'b0	计数比较 A 高分辨率扩展：忽略 CMPAHR 寄存器的低 8 位。该寄存器高 8 位用于存储和操作计数器比较 A 值的高分辨率部分。CMPA 寄存器可以通过单个 32-位的读取/写入操作来访问。与 CMPA 寄存器一样，通过 SHDWAMODE 位来使能影子寄存器，详细内容请参见 CMPA 寄存器描述。

18.4.37 计数比较模块 B 寄存器（EPWMx_CMPB）（x=1,2,3,4,5,6）

偏移地址：0xDB

复位值： 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMPB[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPBHR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[31:16]	CMPB	RW	16'b0	计数比较模块 B：CMPB 寄存器用于与 TBCTR 进行连续比较。如果两个值相等，则计数器比较模块就会生成一个 TBCTR 等于计数比较模块 B 事件。将该事件发送到动作限定进行限定和转换为一个或多个动作。AQCTLA 寄存器和 AQCTLB 寄存器的配置决定这些动作可以应用于 EPWMxA 输出信号或 EPWMxB 输出信号。AQCTLA 寄存器和 AQCTLB 寄存器可以定义的动作如下所示： 无操作：忽略 TBCTR 等于计数比较模块 A 事件 清除：设置 EPWMxA 或 EPWMxB 输出信号为低电平 设置：设置 EPWMxA 或 EPWMxB 输出信号为高电平 翻转：将 EPWMxA 或 EPWMxB 输出信号的状态进行翻转 通过 SHDWBMODE 位来使能 CMPB 影子寄存器，在默认情况下使能影子模式。当 SHDWBMODE=0 时，使能 CMPB 影子寄存器，此时，任何写入或读取操作都会自动进入到影子寄存器。在该情况下，根据 LOADBMODE 位可以确定影子寄存器何时装载到活动寄存器中。当 SHDWBMODE=1 时，禁止 CMPB 影子寄存器，此时，任何写入或读取操作将直接进入到活动寄存器。在写入之前，可以根据 SHDWBFULL 位判断 CMPB 影子寄存器当前状态是否已满。在任何一种模式下，活动寄存器和影子寄存器的内存映射地址相同。
[15:0]	CMPBHR	RW	16'b0	计数比较 B 高分辨率扩展：忽略 CMPBHR 寄存器的低 8 位。

18.4.38 计数比较模块 C 寄存器（EPWMx_CMPC）（x=1,2,3,4,5,6）

偏移地址：0xDE

复位值： 0x0000

该寄存器的链接功能始终为 16 位访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPC[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:0]	CMPC	RW	16'b0	计数比较模块 C：CMPC 寄存器用于与 TBCTR 进行连续比较。如果两个值相等，则计数器比较模块就会生成一个 TBCTR 等于计数比较 C 事件。通过 SHDWCMODE 位来使能 CMPC 影子寄存器，在默认情况下使能影子模式。当 SHDWCMODE=0 时，使能 CMPC 影子寄存器，此时，任何写入或读取操作都会自动进入到影子寄存器。在该情况下，根据 LOADCMODE 位可以确定影子寄存器何时装载到活动寄存器中。当 SHDWCMODE=1 时，禁止 CMPC 影子寄存器，此时，任何写入或读取操作将直接进入到活动寄存器。在任何一种模式下，活动寄存器和影子寄存器的内存映射地址相同。

18.4.39 计数比较模块 D 寄存器（EPWMx_CMPD ）（x=1,2,3,4,5,6）

偏移地址：0xE2

复位值： 0x0000

该寄存器的链接功能始终为 16-位访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPD [15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:0]	CMPD	RW	16'b0	计数比较模块 D：CMPD 寄存器用于与 TBCTR 进行连续比较。如果两个值相等，则计数器比较模块就会生成一个 TBCTR 等于计数比较 D 事件。通过 SHDWDMODE 位来使能 CMPD 影子寄存器，在默认情况下使能影子模式。当 SHDWDMODE=0 时，使能 CMPD 影子寄存器，此时，任何写入或读取操作都会自动进入到影子寄存器。在该情况下，根据 LOADDMODE 位可以确定影子寄存器何时装载到活动寄存器中。当 SHDWDMODE=1 时，禁止 CMPD 影子寄存器，此时，任何写入或读取操作将直接进入到活动寄存器。在任何一种模式下，活动寄存器和影子寄存器的内存映射地址相同。

18.4.40 全局 EPWM 装载控制寄存器 2（EPWMx_GLDCTL2）（x=1,2,3,4,5,6）

偏移地址：0xE8

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GFRCLD	OSHTLD
-	-	-	-	-	-	-	-	-	-	-	-	-	-	RS	RS

Bit	Name	RW	ResetValue	Function
[15:2]	reserved		14'b0	保留
[1]	GFRCLD	RS	1'b0	使能单次模式下的强制装载事件：该位用于在全局装载模式下，通过测试或软件强制来装载事件。 0：无效，读取返回 0 1：在事件预分频计数器的输入位置上强制一个装载事件
[0]	OSHTLD	RS	1'b0	使能单次模式下的重装载事件：该位允许一个装载脉冲事件并阻止后续的时间。 0：无效，读取返回 0 1：使能单次锁存条件。在选择的装载脉冲发生时，会发生一次影子到活动的重装载并清除锁存器。

18.4.41 软件谷模式延迟寄存器（EPWMx_SWVDELVAL）（x=1,2,3,4,5,6）

偏移地址：0xEE

复位值： 0x0000

用户通过设置该寄存器的值，可以根据 VDELAYDIV 位定义硬件计算延迟值。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWVDELVAL[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:0]	SWVDELVAL	RW	16'b0	软件谷延迟值：用户通过设置该寄存器的值，可以根据 VDELAYDIV 位定义硬件计算延迟值。

18.4.42 跳闸区选择寄存器（EPWMx_TZSEL）（x=1,2,3,4,5,6）

偏移地址：0x100

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCBEVT1	DCAEVT1	OSHT6	OSHT5	OSHT4	OSHT3	OSHT2	OSHT1	DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15]	DCBEVT1	RW	1'b0	选择数字比较器 B 输出事件 1 为单次跳闸源：通过该位选择单次跳闸源是否为 DCBEVT1。 0：不是 1：是
[14]	DCAEVT1	RW	1'b0	选择数字比较器 A 输出事件 1 为单次跳闸源：通过该位选择单次跳闸源是否为 DCAEVT1。 0：不是 1：是
[13]	OSHT6	RW	1'b0	选择跳闸区 6 为单次跳闸源：通过该位选择单次跳闸源是否为 TZ6。 0：不是 1：是
[12]	OSHT5	RW	1'b0	选择跳闸区 5 为单次跳闸源：通过该位选择单次跳闸源是否为 TZ5。 0：不是

Confidential

				1：是
[11]	OSHT4	RW	1'b0	选择跳闸区 4 为单次跳闸源：通过该位选择单次跳闸源是否为 TZ4。 0：不是 1：是
[10]	OSHT3	RW	1'b0	选择跳闸区 3 为单次跳闸源：通过该位选择单次跳闸源是否为 TZ3。 0：不是 1：是
[9]	OSHT2	RW	1'b0	选择跳闸区 2 为单次跳闸源：通过该位选择单次跳闸源是否为 TZ2。 0：不是 1：是
[8]	OSHT1	RW	1'b0	选择跳闸区 1 为单次跳闸源：通过该位选择单次跳闸源是否为 TZ1。 0：不是 1：是
[7]	DCBEVT2	RW	1'b0	选择数字比较器 B 输出事件 2 为逐周期跳闸源：通过该位选择逐周期跳闸源是否为 DCBEVT2。 0：不是 1：是
[6]	DCAEVT2	RW	1'b0	选择数字比较器 A 输出事件 2 为逐周期跳闸源：通过该位选择逐周期跳闸源是否为 DCAEVT2。 0：不是 1：是
[5]	CBC6	RW	1'b0	选择跳闸区 6 为逐周期跳闸源：通过该位选择逐周期跳闸源是否为 TZ6。 0：不是 1：是
[4]	CBC5	RW	1'b0	选择跳闸区 5 为逐周期跳闸源：通过该位选择逐周期跳闸源是否为 TZ5。 0：不是 1：是
[3]	CBC4	RW	1'b0	选择跳闸区 4 为逐周期跳闸源：通过该位选择逐周期跳闸源是否为 TZ4。 0：不是 1：是
[2]	CBC3	RW	1'b0	选择跳闸区 3 为逐周期跳闸源：通过该位选择逐周期跳闸源是否为 TZ3。 0：不是 1：是
[1]	CBC2	RW	1'b0	选择跳闸区 2 为逐周期跳闸源：通过该位选择逐周期跳闸源是否为 TZ2。 0：不是 1：是
[0]	CBC1	RW	1'b0	选择跳闸区 1 为逐周期跳闸源：通过该位选择逐周期跳闸源是否为 TZ1。 0：不是 1：是

18.4.43 跳闸区数字比较器配置寄存器（EPWMx_TZDCSEL）（x=1,2,3,4,5,6）

偏移地址：0x104

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DCBEVT2[2:0]			DCBEVT1[2:0]			DCAEVT2[2:0]			DCAEVT1[2:0]		

-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11:9]	DCBEVT2	RW	3'b0	配置数字比较器 B 输出事件 2：参考 DCAEVT1 描述
[8:6]	DCBEVT1	RW	3'b0	配置数字比较器 B 输出事件 1：参考 DCAEVT1 描述
[5:3]	DCAEVT2	RW	3'b0	配置数字比较器 A 输出事件 2：参考 DCAEVT1 描述
[2:0]	DCAEVT1	RW	3'b0	配置数字比较器 A 输出事件 1： 000：禁止数字比较器 A 输出事件 1 001：在 DCAH 信号为低电平时触发 010：在 DCAH 信号为高电平时触发 011：在 DCAL 信号为低电平时触发 100：在 DCAL 信号为高电平时触发 101：在 DCAL 信号为高电平，DCAH 信号为低电平时触发 110：保留 111：保留

18.4.44 跳闸区控制寄存器（EPWMx_TZCTL）（x=1,2,3,4,5,6）

偏移地址：0x108

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DCBEVT2[1:0]		DCBEVT1[1:0]		DCAEVT2[1:0]		DCAEVT1[1:0]		TZB[1:0]		TZA[1:0]	
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11:10]	DCBEVT2	RW	2'b0	配置 EPWMxB 的数字比较器 B 输出事件 2 动作： 00：EPWMxB 为高阻态 01：强制 EPWMxB 输出为高电平 10：强制 EPWMxB 输出为低电平 11：禁止跳闸动作
[9:8]	DCBEVT1	RW	2'b0	配置 EPWMxB 的数字比较器 B 输出事件 1 动作： 00：EPWMxB 为高阻态 01：强制 EPWMxB 输出为高电平 10：强制 EPWMxB 输出为低电平 11：禁止跳闸动作
[7:6]	DCAEVT2	RW	2'b0	配置 EPWMxA 的数字比较器 A 输出事件 2 动作： 00：EPWMxA 为高阻态 01：强制 EPWMxA 输出为高电平 10：强制 EPWMxA 输出为低电平 11：禁止跳闸动作
[5:4]	DCAEVT1	RW	2'b0	配置 EPWMxA 的数字比较器 A 输出事件 1 动作： 00：EPWMxA 为高阻态 01：强制 EPWMxA 输出为高电平 10：强制 EPWMxA 输出为低电平

				11：禁止跳闸动作
[3:2]	TZB	RW	2'b0	配置跳闸事件发生时 EPWMxB 动作：该位或定义在发生跳闸事件时对 EPWMxB 输出采取的动作，通过 TZSEL 寄存器可以确定哪些跳闸区引脚可以引发事件。 00：EPWMxB 为高阻态 01：强制 EPWMxB 输出为高电平 10：强制 EPWMxB 输出为低电平 11：EPWMxB 保持原状态
[1:0]	TZA	RW	2'b0	配置跳闸事件发生时 EPWMxA 动作：该位或定义在发生跳闸事件时对 EPWMxA 输出采取的动作，通过 TZSEL 寄存器可以确定哪些跳闸区引脚可以引发事件。 00：EPWMxA 为高阻态 01：强制 EPWMxA 输出为高电平 10：强制 EPWMxA 输出为低电平 11：EPWMxA 保持原状态

18.4.45 跳闸区控制寄存器 2（EPWMx_TZCTL2）（x=1,2,3,4,5,6）

偏移地址：0x10A

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETZE	Res.	Res.	Res.	TZBD[2:0]			TZBU[2:0]			TZAD[2:0]			TZAU[2:0]		
RW	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15]	ETZE	RW	1'b0	使能 TZCTL2： 0：通过 TZCTL 寄存器中定义的跳闸动作 1：通过 TZCTL2 寄存器、TZCTLDCA 寄存器和 TZCTLDCB 寄存器中定义的跳闸动作，忽略 TZCTL 寄存器中的设置。
[14:12]	reserved		3'b0	保留
[11:9]	TZBD	RW	3'b0	计数方向向下时，配置跳闸事件发生时 EPWMxB 动作： 000：EPWMxB 输出为高阻态 001：强制 EPWMxB 输出为高电平 010：强制 EPWMxB 输出为低电平 011：将 EPWMxB 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作
[8:6]	TZBU	RW	3'b0	计数方向向上时，配置跳闸事件发生时 EPWMxB 动作： 000：EPWMxB 输出为高阻态 001：强制 EPWMxB 输出为高电平 010：强制 EPWMxB 输出为低电平 011：将 EPWMxB 输出状态进行翻转 100：保留 101：保留

				110：保留 111：禁止跳闸动作
[5:3]	TZAD	RW	3'b0	计数方向向下时，配置跳闸事件发生时 EPWMxA 动作： 000：EPWMxA 输出为高阻态 001：强制 EPWMxA 输出为高电平 010：强制 EPWMxA 输出为低电平 011：将 EPWMxA 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作
[2:0]	TZAU	RW	3'b0	计数方向向上时，配置跳闸事件发生时 EPWMxA 动作： 000：EPWMxA 输出为高阻态 001：强制 EPWMxA 输出为高电平 010：强制 EPWMxA 输出为低电平 011：将 EPWMxA 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作

18.4.46 跳闸区控制数字比较器 A 寄存器（EPWMx_TZCTLDCA）（x=1,2,3,4,5,6）

偏移地址：0x10C

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DCAEVT2D[2:0]			DCAEVT2U[2:0]			DCAEVT1D[2:0]			DCAEVT1U[2:0]		
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11:9]	DCAEVT2D	RW	3'b0	计数方向向下时，配置 EPWMxA 的数字比较器 A 输出事件 2 动作：参考 DCAEVT1U 描述
[8:6]	DCAEVT2U	RW	3'b0	计数方向向上时，配置 EPWMxA 的数字比较器 A 输出事件 2 动作：参考 DCAEVT1U 描述
[5:3]	DCAEVT1D	RW	3'b0	计数方向向下时，配置 EPWMxA 的数字比较器 A 输出事件 1 动作：参考 DCAEVT1U 描述
[2:0]	DCAEVT1U	RW	3'b0	计数方向向上时，配置 EPWMxA 的数字比较器 A 输出事件 1 动作： 000：EPWMxA 输出为高阻态 001：强制 EPWMxA 输出为高电平 010：强制 EPWMxA 输出为低电平 011：将 EPWMxA 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作

18.4.47 跳闸区控制数字比较器 B 寄存器（EPWMx_TZCTLDCB）（x=1,2,3,4,5,6）

偏移地址：0x10E

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DCBEVT2D[2:0]			DCBEVT2U[2:0]			DCBEVT1D[2:0]			DCBEVT1U[2:0]		
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11:9]	DCBEVT2D	RW	3'b0	计数方向向下时，配置 EPWMxB 的数字比较器 B 输出事件 2 动作：参考 DCBEVT1U 描述
[8:6]	DCBEVT2U	RW	3'b0	计数方向向上时，配置 EPWMxB 的数字比较器 B 输出事件 2 动作：参考 DCBEVT1U 描述
[5:3]	DCBEVT1D	RW	3'b0	计数方向向下时，配置 EPWMxB 的数字比较器 B 输出事件 1 动作：参考 DCBEVT1U 描述
[2:0]	DCBEVT1U	RW	3'b0	计数方向向上时，配置 EPWMxB 的数字比较器 B 输出事件 1 动作： 000：EPWMxB 输出为高阻态 001：强制 EPWMxB 输出为高电平 010：强制 EPWMxB 输出为低电平 011：将 EPWMxB 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作

18.4.48 跳闸区中断使能寄存器（EPWMx_TZEINT）（x=1,2,3,4,5,6）

偏移地址：0x11A

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	Res.
-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	-

Bit	Name	RW	ResetValue	Function
[15:7]	reserved		9'b0	保留
[6]	DCBEVT2	RW	1'b0	使能数字比较器 A 输出事件 2 中断： 0：禁止 1：使能
[5]	DCBEVT1	RW	1'b0	使能数字比较器 B 输出事件 1 中断： 0：禁止 1：使能
[4]	DCAEVT2	RW	1'b0	使能数字比较器 A 输出事件 2 中断： 0：禁止 1：使能
[3]	DCAEVT1	RW	1'b0	使能数字比较器 A 输出事件 1 中断：

				0：禁止 1：使能
[2]	OST	RW	1'b0	使能 OST 中断：单次跳闸事件将会导致中断。 0：禁止 1：使能
[1]	CBC	RW	1'b0	使能 CBC 中断：逐周期跳闸事件将会导致中断。 0：禁止 1：使能
[0]	reserved		1'b0	保留

18.4.49 跳闸区标志寄存器（EPWMx_TZFLG）（x=1,2,3,4,5,6）

偏移地址：0x126

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
[15:7]	reserved		9'b0	保留
[6]	DCBEVT2	R	1'b0	数字比较器 B 输出事件 2 锁存状态： 0：DCBEVT2 未发生跳闸事件 1：DCBEVT2 发生跳闸事件
[5]	DCBEVT1	R	1'b0	数字比较器 B 输出事件 1 锁存状态： 0：DCBEVT1 未发生跳闸事件 1：DCBEVT1 发生跳闸事件
[4]	DCAEVT2	R	1'b0	数字比较器 A 输出事件 2 锁存状态： 0：DCAEVT2 未发生跳闸事件 1：DCAEVT2 发生跳闸事件
[3]	DCAEVT1	R	1'b0	数字比较器 A 输出事件 1 锁存状态： 0：DCAEVT1 未发生跳闸事件 1：DCAEVT1 发生跳闸事件
[2]	OST	R	1'b0	单次跳闸事件锁存状态：通过将适当的值写入 TZCLR 寄存器来清除此位。 0：未发生单次跳闸事件 1：在单次跳闸源引脚上发生单次跳闸事件
[1]	CBC	R	1'b0	逐周期跳闸事件锁存状态：如果设置了该位，则会一直保持设置状态，直到手动清除该位。当清除此位但是仍然存在逐周期跳闸事件时，此位就会立即再次设置。当 TBCTR=0 时，若跳闸条件不再存在，则自动清除信号条件。无论在周期中任一位置，只有 TBCTR=0 时，才能清除信号条件。通过将适当的值写入 TZCLR 寄存器来清除此位。 0：未发生逐周期跳闸事件 1：发生逐周期跳闸事件，该事件是由于选择一个信号作为逐周期跳闸源引起的
[0]	INT	R	1'b0	跳闸中断锁存状态：未清除该位之前，不会再生成中断。当设置 CBC 位或 OST 位前设置此位，将会生成另一个中断脉冲。清除所有状态标志位可以防止进一步的中断。通过将适当的值写入 TZCLR 寄存器来清除此位。

				0：未产生中断 1：因为跳闸条件而产生中断

18.4.50 跳闸区 CBC 标志寄存器（EPWMx_TZCBCFLG）（x=1,2,3,4,5,6）

偏移地址：0x128

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7]	DCBEVT2	R	1'b0	数字比较器 B 输出事件 2 触发锁存标志： 0：DCBEVT2 未发生跳闸事件 1：DCBEVT2 发生跳闸事件
[6]	DCAEVT2	R	1'b0	数字比较器 A 输出事件 2 触发锁存标志： 0：DCAEVT2 未发生跳闸事件 1：DCAEVT2 发生跳闸事件
[5]	CBC6	R	1'b0	CBC6 锁存标志： 0：CBC6 未发生跳闸事件 1：CBC6 发生跳闸事件
[4]	CBC5	R	1'b0	CBC5 锁存标志： 0：CBC5 未发生跳闸事件 1：CBC5 发生跳闸事件
[3]	CBC4	R	1'b0	CBC4 锁存标志： 0：CBC4 未发生跳闸事件 1：CBC4 发生跳闸事件
[2]	CBC3	R	1'b0	CBC3 锁存标志： 0：CBC3 未发生跳闸事件 1：CBC3 发生跳闸事件
[1]	CBC2	R	1'b0	CBC2 锁存标志： 0：CBC2 未发生跳闸事件 1：CBC2 发生跳闸事件
[0]	CBC1	R	1'b0	CBC1 锁存标志： 0：CBC1 未发生跳闸事件 1：CBC1 发生跳闸事件

18.4.51 跳闸区 OST 标志寄存器（EPWMx_TZOSTFLG）（x=1,2,3,4,5,6）

偏移地址：0x12A

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEVT1	DCAEVT1	OST6	OST5	OST4	OST3	OST2	OST1
-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7]	DCBEVT1	R	1'b0	数字比较器 B 输出事件 1 锁存标志： 0：DCBEVT1 未发生跳闸事件 1：DCBEVT1 发生跳闸事件
[6]	DCAEVT1	R	1'b0	数字比较器 A 输出事件 1 锁存标志： 0：DCAEVT1 未发生跳闸事件 1：DCAEVT1 发生跳闸事件
[5]	OST6	R	1'b0	OST6 锁存标志： 0：OST6 未发生跳闸事件 1：OST6 发生跳闸事件
[4]	OST5	R	1'b0	OST5 锁存标志： 0：OST5 未发生跳闸事件 1：OST5 发生跳闸事件
[3]	OST4	R	1'b0	OST4 锁存标志： 0：OST4 未发生跳闸事件 1：OST4 发生跳闸事件
[2]	OST3	R	1'b0	OST3 锁存标志： 0：OST3 未发生跳闸事件 1：OST3 发生跳闸事件
[1]	OST2	R	1'b0	OST2 锁存标志： 0：OST2 未发生跳闸事件 1：OST2 发生跳闸事件
[0]	OST1	R	1'b0	OST1 锁存标志： 0：OST1 未发生跳闸事件 1：OST1 发生跳闸事件

18.4.52 清除跳闸区寄存器（EPWMx_TZCLR）（x=1,2,3,4,5,6）

偏移地址：0x12E

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CBCPULSE[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
RW	RW	-	-	-	-	-	-	-	RS	RS	RS	RS	RS	RS	RS

Bit	Name	RW	ResetValue	Function
[15:14]	CBCPULSE	RW	2'b0	清除 CBC 锁存脉冲：通过该位域可以选择清除 CBC 锁存的脉冲。 00：计数器等于 0 的脉冲 01：计数器等于周期的脉冲 10：计数器等于 0 或计数器等于周期的脉冲 11：不清除 CBC 锁存
[13:7]	reserved		7'b0	保留
[6]	DCBEVT2	RS	1'b0	清除数字比较器 B 输出事件 2 锁存标志： 0：无效，读取返回 0 1：清除 DCBEVT2 跳闸条件
[5]	DCBEVT1	RS	1'b0	清除数字比较器 B 输出事件 1 锁存标志： 0：无效，读取返回 0

				1：清除 DCBEVT1 跳闸条件
[4]	DCAEVT2	RS	1'b0	清除数字比较器 A 输出事件 2 锁存标志： 0：无效，读取返回 0 1：清除 DCAEVT2 跳闸条件
[3]	DCAEVT1	RS	1'b0	清除数字比较器 A 输出事件 1 锁存标志： 0：无效，读取返回 0 1：清除 DCAEVT1 跳闸条件
[2]	OST	RS	1'b0	清除 OST 锁存标志： 0：无效，读取返回 0 1：清除 OST 跳闸条件
[1]	CBC	RS	1'b0	清除 CBC 锁存标志： 0：无效，读取返回 0 1：清除 CBC 跳闸条件
[0]	INT	RS	1'b0	清除全局中断标志：未清除 INT 位之前，不会再生成中断。清除 INT 位后，设置任何标志位将会生成另一个中断脉冲。清除所有状态标志位可以防止进一步的中断。 0：无效，读取返回 0 1：清除 INT 位

18.4.53 清除跳闸区 CBC 寄存器（EPWMx_TZCBCCLR）（x=1,2,3,4,5,6）

偏移地址：0x130

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
-	-	-	-	-	-	-	-	RS	RS	RS	RS	RS	RS	RS	RS

Bit	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7]	DCBEVT2	RS	1'b0	清除 CBC 选择的数字比较器 B 输出事件 2 标志： 0：无效，读取返回 0 1：清除 DCBEVT2 位
[6]	DCAEVT2	RS	1'b0	清除 CBC 选择的数字比较器 A 输出事件 2 标志： 0：无效，读取返回 0 1：清除 DCAEVT2 位
[5]	CBC6	RS	1'b0	清除 CBC6 锁存标志： 0：无效，读取返回 0 1：清除 CBC6 位
[4]	CBC5	RS	1'b0	清除 CBC5 锁存标志： 0：无效，读取返回 0 1：清除 CBC5 位
[3]	CBC4	RS	1'b0	清除 CBC4 锁存标志： 0：无效，读取返回 0 1：清除 CBC4 位
[2]	CBC3	RS	1'b0	清除 CBC3 锁存标志： 0：无效，读取返回 0

				1：清除 CBC3 位
[1]	CBC2	RS	1'b0	清除 CBC2 锁存标志： 0：无效，读取返回 0 1：清除 CBC2 位
[0]	CBC1	RS	1'b0	清除 CBC1 锁存标志： 0：无效，读取返回 0 1：清除 CBC1 位

18.4.54 清除跳闸区 OST 寄存器（EPWMx_TZOSTCLR）（x=1,2,3,4,5,6）

偏移地址：0x132

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEVT1	DCAEVT1	OST6	OST5	OST4	OST3	OST2	OST1
-	-	-	-	-	-	-	-	RS	RS	RS	RS	RS	RS	RS	RS

Bit	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7]	DCBEVT1	RS	1'b0	清除 OST 选择的数字比较器 B 输出事件 1 标志： 0：无效，读取返回 0 1：清除 DCBEVT1 位
[6]	DCAEVT1	RS	1'b0	清除 OST 选择的数字比较器 A 输出事件 1 标志： 0：无效，读取返回 0 1：清除 DCAEVT1 位
[5]	OST6	RS	1'b0	清除 OST6 锁存标志： 0：无效，读取返回 0 1：清除 OST6 位
[4]	OST5	RS	1'b0	清除 OST5 锁存标志： 0：无效，读取返回 0 1：清除 OST5 位
[3]	OST4	RS	1'b0	清除 OST4 锁存标志： 0：无效，读取返回 0 1：清除 OST4 位
[2]	OST3	RS	1'b0	清除 OST3 锁存标志： 0：无效，读取返回 0 1：清除 OST3 位
[1]	OST2	RS	1'b0	清除 OST2 锁存标志： 0：无效，读取返回 0 1：清除 OST2 位
[0]	OST1	RS	1'b0	清除 OST1 锁存标志： 0：无效，读取返回 0 1：清除 OST1 位

18.4.55 强制跳闸区标志寄存器（EPWMx_TZFRC）（x=1,2,3,4,5,6）

偏移地址：0x136

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	Res.
-	-	-	-	-	-	-	-	-	RS	RS	RS	RS	RS	RS	-

Bit	Name	RW	ResetValue	Function
[15:7]	reserved		9'b0	保留
[6]	DCBEVT2	RS	1'b0	数字比较器 B 输出事件 2 强制标志： 0：无效，读取返回 0 1：强制触发 DCBEVT2 跳闸条件并设置 DCBEVT2 位
[5]	DCBEVT1	RS	1'b0	数字比较器 B 输出事件 1 强制标志： 0：无效，读取返回 0 1：强制触发 DCBEVT1 跳闸条件并设置 DCBEVT1 位
[4]	DCAEVT2	RS	1'b0	数字比较器 A 输出事件 2 强制标志： 0：无效，读取返回 0 1：强制触发 DCAEVT2 跳闸条件并设置 DCAEVT2 位
[3]	DCAEVT1	RS	1'b0	数字比较器 A 输出事件 1 强制标志： 0：无效，读取返回 0 1：强制触发 DCAEVT1 跳闸条件并设置 DCAEVT1 位
[2]	OST	RS	1'b0	软件强制单次跳闸事件： 0：无效，读取返回 0 1：强制触发单次跳闸事件并设置 OST 位
[1]	CBC	RS	1'b0	软件强制逐周期跳闸事件： 0：无效，读取返回 0 1：强制触发逐周期跳闸事件并设置 CBC 位
[0]	reserved		1'b0	保留

18.4.56 事件触发选择寄存器（EPWMx_ETSEL）（x=1,2,3,4,5,6）

偏移地址：0x148

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOCBEN	SOCBSEL[2:0]			SOCAEN	SOCASEL[2:0]			Res.	INTSELCMP	SOCBSELCMP	SOCASELCMP	INTEN	INTSEL[2:0]		
RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15]	SOCBEN	RW	1'b0	使能 B 路触发 ADC 转换脉冲（EPWMxSOCB）： 0：禁止 1：使能
[14:12]	SOCBSEL	RW	3'b0	选择生成 EPWMxSOCB 的事件：通过该位域可以选择在生成 EPWMxSOCB 事件的条件。根据 SOCBSELCMP 位选择 EPWMxSOCB 比较器事件。 000：在 DCBEVT1.soc 事件时生成 001：在 TBCTR=0 时生成 010：在 TBCTR=TBPRD 时生成 011：在 TBCTR=0 或 TBCTR=TBPRD 时生成（用于中心对齐计数模

				式） 100：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 101：当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 110：当计数器向上计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成 111：当计数器向下计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成
[11]	SOCAEN	RW	1'b0	使能 A 路触发 ADC 转换脉冲（EPWMxSOCA）： 0：禁止 1：使能
[10:8]	SOCASEL	RW	3'b0	选择生成 EPWMxSOCA 的事件：通过该位域可以选择在生成 EPWMxSOCA 事件的条件。根据 SOCASELCMP 位选择 EPWMxSOCA 比较器事件。 000：在 DCAEVT1.soc 事件时生成 001：在 TBCTR=0 时生成 010：在 TBCTR=TBPRD 时生成 011：在 TBCTR=0 或 TBCTR=TBPRD 时生成（用于中心对齐计数模式） 100：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 101：当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 110：当计数器向上计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成 111：当计数器向下计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成
[7]	reserved		1'b0	保留
[6]	INTSELCMP	RW	1'b0	选择 EPWMxINT 比较器： 0=使用 CMPA/CMPB， 1=使用 CMPC/CMPD
[5]	SOCBSELCMP	RW	1'b0	选择 EPWMxSOCB 比较器： 0=使用 CMPA/CMPB， 1=使用 CMPC/CMPD
[4]	SOCASELCMP	RW	1'b0	选择 EPWMxSOCA 比较器： 0=使用 CMPA/CMPB， 1=使用 CMPC/CMPD
[3]	INTEN	RW	1'b0	使能生成 EPWM 中断： 0：禁止 1：使能
[2:0]	INTSEL	RW	3'b0	选择 EPWM 中断：通过该位域可以选择生成 EPWM 中断事件的条件。 000：保留 001：在 TBCTR=0 时生成

				010：在 TBCTR=TBPRD 时生成 011：在 TBCTR=0 或 TBCTR=TBPRD 时生成（用于中心对齐计数模式） 100：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 101：当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 110：当计数器向上计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成 111：当计数器向下计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成

18.4.57 事件触发预分配寄存器（EPWMx_ETPS）（x=1,2,3,4,5,6）

偏移地址：0x14C

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOCBCNT[1:0]		SOCBPRD[1:0]		SOCACNT[1:0]		SOCAPRD[1:0]		Res.	Res.	SOCPSSEL	INTPSSEL	INTCNT[1:0]		INTPRD[1:0]	
R	R	RW	RW	R	R	RW	RW	-	-	RW	RW	R	R	RW	RW

Bits	Name	RW	ResetValue	Function
[15:14]	SOCBCNT	R	2'b0	EPWMxSOCB 事件计数器：通过该位域选择已发生 ETSEL[SOCBSEL]事件的数量。 00：无 SOCBSEL 事件发生 01：发生 1 个 SOCBSEL 事件 10：发生 2 个 SOCBSEL 事件 11：发生 3 个 SOCBSEL 事件
[13:12]	SOCBPRD	RW	2'b0	选择 EPWMxSOCB 周期：该位用于确定生成 EPWMxSOCB 脉冲前的条件。必须使 ETSEL[SOCBEN]=1 才能生成脉冲，即使设置 ETFLG[SOCB]位，也将生成 EPWMxSOCB 脉冲。此时，将会自动清除 SOCBCNT 位。 00：SOCBCNT=00，不产生脉冲 01：SOCBCNT=01，在第 1 个事件发生时产生脉冲 10：SOCBCNT=10，在第 2 个事件发生时产生脉冲 11：SOCBCNT=11，在第 3 个事件发生时产生脉冲
[11:10]	SOCACNT	R	2'b0	EPWMxSOCA 事件计数器：通过该位域选择已发生 ETSEL[SOCASEL]事件的数量。 00：无 SOCASEL 事件发生 01：发生 1 个 SOCASEL 事件 10：发生 2 个 SOCASEL 事件 11：发生 3 个 SOCASEL 事件
[9:8]	SOCAPRD	RW	2'b0	选择 EPWMxSOCA 周期：该位域用于确定生成 EPWMxSOCA 脉冲前的条件。必须使 ETSEI[SOCAEN]=1 才能生成脉冲，即使设置 ETFLG[SOCA]位，也将在成 EPWMxSOCA 脉冲。此时，将会自动清除 ETPS[SOCACNT]位。 00：SOCACNT=00，不产生脉冲 01：SOCACNT=01，在第 1 个事件发生时产生脉冲

				10：SOCACNT=10，在第 2 个事件发生时产生脉冲 11：SOCACNT=11，在第 3 个事件发生时产生脉冲
[7:6]	reserved		2'b0	保留
[5]	SOCPSSEL	RW	1'b0	选择 EPWMxSOC A/B 预分频： 0：通过 SOCACNT 位/SOCBCNT 位和 SOCAPRD 位/SOCBPRD 位来确定事件频率为每 0...3 个事件产生一次脉冲 1：通过 SOCACNT2 位/SOCBCNT2 位和 SOCAPRD2 位/SOCBPRD2 位来确定事件频率为每 0...15 个事件产生一次脉冲
[4]	INTPSSEL	RW	1'b0	选择 EPWM 中断预分频： 0：通过 ETPS[INTCNT, and INTPRD]位来确定事件频率为每 0...3 个事件产生一次脉冲 1：通过 ETINTPS[INTCNT2, and INTPRD2]位来确定事件频率为每 0...15 个事件产生一次脉冲
[3:2]	INTCNT	R	2'b0	EPWM 中断事件计数器：通过该位域选择已发生 INTSEL 事件的数量。该位域会在生成中断脉冲时自动清零，当 ETSEI[INT]=0 或 ETFLG[INT]=1 时，计数器在 ETPS[INTCNT] = ETPS[INTPRD]时停止计数。 00：无 INTSEL 事件发生 01：发生 1 个 INTSEL 事件 10：发生 2 个 INTSEL 事件 11：发生 3 个 INTSEL 事件
[1:0]	INTPRD	RW	2'b0	选择 EPWM 中断周期：该位域用于确定生成中断前的条件。必须使 ETSEL[INT]=1 才能生成中断，未使用 ETSEL[INT]位清除 ETFLG[INT]标志位前不会产生中断。如果产生中断，则会自动清除 INTECNT 位。 00：当 INTECNT=00 时，不产生中断且忽略 ETFRG[INT]位 01：当 INTECNT=01 时，在第 1 个事件发生时产生中断 10：当 INTECNT=10 时，在第 2 个事件发生时产生中断 11：当 INTECNT=11 时，在第 3 个事件发生时产生中断

18.4.58 事件触发标志寄存器（EPWMx_ETFLG）（x=1,2,3,4,5,6）

偏移地址：0x150

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOCB	SOCA	Res.	INT
-	-	-	-	-	-	-	-	-	-	-	-	R	R	-	R

Bits	Name	RW	ResetValue	Function
[15:4]	reserved		12'b0	保留
[3]	SOCB	R	1'b0	EPWMxSOCB 标志：当设置该位时，在 EPWMxSOCB 输出仍继续产生转换开始脉冲。 0：未产生脉冲 1：产生脉冲
[2]	SOCA	R	1'b0	EPWMxSOCA 标志：当设置该位时，在 EPWMxSOCA 输出仍继续产生转换开始脉冲。 0：未产生脉冲 1：产生脉冲
[1]	reserved		1'b0	保留

[0]	INT	R	1'b0	EPWM 中断锁存标志：需要清除该位才能产生新的中断，且当 INT 位仍处于设置状态时，最多可以有一个中断挂起。如果有中断挂起，则会在清除 INT 位之后才能生成中断。 0：未产生中断 1：产生中断

18.4.59 清除事件触发寄存器（EPWMx_ETCLR）（x=1,2,3,4,5,6）

偏移地址：0x154

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOCB	SOCA	Res.	INT
-	-	-	-	-	-	-	-	-	-	-	-	RS	RS	-	RS

Bits	Name	RW	ResetValue	Function
[15:4]	reserved		12'b0	保留
[3]	SOCB	RS	1'b0	清除 EPWMxSOCB 标志： 0：无效，读取返回 0 1：清除 SOCB 位
[2]	SOCA	RS	1'b0	清除 EPWMxSOCA 标志： 0：无效，读取返回 0 1：清除 SOCA 位
[1]	reserved		1'b0	保留
[0]	INT	RS	1'b0	清除 EPWM 中断标志： 0：无效，读取返回 0 1：清除 INT 位

18.4.60 强制事件触发寄存器（EPWMx_ETFRC）（x=1,2,3,4,5,6）

偏移地址：0x158

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOCB	SOCA	Res.	INT
-	-	-	-	-	-	-	-	-	-	-	-	RS	RS	-	RS

Bit	Name	RW	ResetValue	Function
[15:4]	reserved		12'b0	保留
[3]	SOCB	RS	1'b0	强制产生 SOCB 脉冲：该位用于测试。如果在 ETSEL 寄存器中使能该事件，则会产生 EPWMxSOCB 脉冲。为了确定是否生成 EPWMxSOCB 事件，无论是否生成脉冲都会设置 SOCB 位。 0：无效，读取返回 0 1：产生一个脉冲并设置 SOCB 位
[2]	SOCA	RS	1'b0	强制产生 SOCA 脉冲：该位用于测试。如果在 ETSEL 寄存器中使能该事件，则会产生 EPWMxSOCA 脉冲。为了确定是否生成 EPWMxSOCA 事件，无论是否生成脉冲都会设置 SOCA 位。 0：无效，读取返回 0 1：产生一个脉冲并设置 SOCA 位

[1]	reserved		1'b0	保留
[0]	INT	RS	1'b0	强制产生中断：如果在 ETSEL 寄存器中使能该事件，则会产生中断。 0：无效，读取返回 0 1：产生中断并设置 INT 位

18.4.61 事件触发中断预分频寄存器（EPWMx_ETINTPS）（x=1,2,3,4,5,6）

偏移地址：0x15C

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTCNT2[3:0]				INTPRD2[3:0]			
-	-	-	-	-	-	-	-	R	R	R	R	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:8]	reserved		8'b0	保留
[7:4]	INTCNT2	R	4'b0	EPWM 中断事件计数器 2：当 INTPSSEL=1 时，该位用于选择已发生 INTSEL 事件的数量。 0000：无 INTSEL 事件发生 0001：发生 1 个 INTSEL 事件 0010：发生 2 个 INTSEL 事件 0011：发生 3 个 INTSEL 事件 … 1111：发生 15 个 INTSEL 事件
[3:0]	INTPRD2	RW	4'b0	选择 EPWM 中断周期 2：当 INTPSSEL=1 时，该位用于确定生成中断前的条件。 0000：INTCNT2=0000，不产生中断 0001：INTCNT2=0001，在第 1 个事件发生时产生中断 0010：INTCNT2=0010，在第 2 个事件发生时产生中断 0011：INTCNT2=0011，在第 3 个事件发生时产生中断 ... 1111：INTCNT2=1111，在第 15 个事件发生时产生中断...

18.4.62 事件触发 SOC 预分频寄存器（EPWMx_ETSOCPS）（x=1,2,3,4,5,6）

偏移地址：0x160

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOCBCNT2[3:0]				SOCBPRD2[3:0]				SOCACNT2[3:0]				SOCAPRD2[3:0]			
R	R	R	R	RW	RW	RW	RW	R	R	R	R	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	SOCBCNT2	R	4'b0	EPWMxSOCB 事件计数器：当 SOCPSSEL=1 时，该位用于选择已发生 SOCASEL 事件的数量。 0000：无 SOCASEL 事件发生 0001：发生 1 个 SOCASEL 事件 0010：发生 2 个 SOCASEL 事件 0011：发生 3 个 SOCASEL 事件

				… 1111：发生 15 个 SOCASEL 事件
[11:8]	SOCBPRD2	RW	4'b0	EPWMxSOCB 事件计数器 2：当 SOCPSSEL=1 时，该位域用于确定生成 SOCB 脉冲前的条件。 0000：SOCBCNT2=0000，不产生中断 0001：SOCBCNT2=0001，在第 1 个事件发生时产生 SOCB 脉冲 0010：SOCBCNT2=0010，在第 2 个事件发生时产生 SOCB 脉冲 0011：SOCBCNT2=0011，在第 3 个事件发生时产生 SOCB 脉冲 … 1111：SOCBCNT2=1111，在第 15 个事件发生时产生 SOCB 脉冲
[7:4]	SOCACNT2	R	4'b0	EPWMxSOCA 事件计数器 2：当 SOCPSSEL=1 时，该位用于选择已发生 SOCASEL 事件的数量。 0000：无 SOCASEL 事件发生 0001：发生 1 个 SOCASEL 事件 0010：发生 2 个 SOCASEL 事件 0011：发生 3 个 SOCASEL 事件 … 1111：发生 15 个 SOCASEL 事件
[3:0]	SOCAPRD2	RW	4'b0	选择 EPWMxSOCA 周期 2：当 SOCPSSEL=1 时，该位用于确定生成 SOCA 脉冲前的条件。 0000：SOCACNT2=0000，不产生中断 0001：SOCACNT2=0001，在第 1 个事件发生时产生 SOCA 脉冲 0010：SOCACNT2=0010，在第 2 个事件发生时产生 SOCA 脉冲 0011：SOCACNT2=0011，在第 3 个事件发生时产生 SOCA 脉冲 … 1111：SOCACNT2=1111，在第 15 个事件发生时产生 SOCA 脉冲

18.4.63 事件触发计数器初始化控制寄存器（EPWMx_ETCNTINITCTL）（x=1,2,3,4,5,6）

偏移地址：0x164

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOCBINTEN	SOCAINTEN	INTINTEN	SOCBINITFRC	SOCAINITFRC	INTINITFRC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
RW	RW	RW	RS	RS	RS	-	-	-	-	-	-	-	-	-	-

Bit	Name	RW	ResetValue	Function
[15]	SOCBINTEN	RW	1'b0	使能 EPWMxSOCB 计数器 2 初始化： 0：无效 1：EPWMxSOCB 计数器在同步事件或软件强制时，通过 SOCBINTFRC 内容进行初始化
[14]	SOCAINTEN	RW	1'b0	使能 EPWMxSOCA 计数器 2 初始化： 0：无效 1：EPWMxSOCA 计数器在同步事件或软件强制时，通过 SOCAINTFRC 内容进行初始化
[13]	INTINTEN	RW	1'b0	使能 EPWM 中断计数器 2 初始化： 0：无效

				1：EPWM 中断计数器在同步事件或软件强制时，通过 INTINTFRC 内容进行初始化
[12]	SOCBINITFRC	RS	1'b0	强制 EPWMxSOCB 计数器 2 初始化： 0：无效 1：强制 EPWMxSOCB 计数器通过 SOCBINIT 内容进行初始化
[11]	SOCAINITFRC	RS	1'b0	强制 EPWMxSOCA 计数器 2 初始化： 0：无效 1：强制 EPWMxSOCB 计数器通过 SOCAINIT 内容进行初始化
[10]	INTINITFRC	RS	1'b0	强制 EPWM 中断计数器 2 初始化： 0：无效 1：强制 EPWMxSOCB 计数器通过 INTINIT 内容进行初始化
[9:0]	reserved		10'b0	保留

18.4.64 事件触发计数器初始化寄存器（EPWMx_ETCNTINIT）（x=1,2,3,4,5,6）

偏移地址：0x168

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SOCBINT[3:0]				SOCAINT[3:0]				INTINT[3:0]			
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11:8]	SOCBINT	RW	4'b0	EPWMxSOCB 计数器 2 初始化：当发生同步事件或软件强制时，该寄存器内容被用于初始化 EPWMxSOCB 计数器 2。
[7:4]	SOCAINT	RW	4'b0	EPWMxSOCA 计数器 2 初始化：当发生同步事件或软件强制时，该寄存器内容被用于初始化 EPWMxSOCA 计数器 2。
[3:0]	INTINT	RW	4'b0	EPWM 中断计数器 2 初始化：当发生同步事件或软件强制时，该寄存器内容被用于初始化 PWM 中断计数器 2。

18.4.65 数字比较跳闸选择寄存器（EPWMx_DCTRIPSEL）（x=1,2,3,4,5,6）

偏移地址：0x180

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCBLCOMPSEL[3:0]				DCBHCOMPSEL[3:0]				DCALCOMPSEL[3:0]				DCAHCOMPSEL[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	DCBLCOMPSEL	RW	4'b0	选择数字比较器 B 低位输入：参考 DCAHCOMPSEL 描述
[11:8]	DCBHCOMPSEL	RW	4'b0	选择数字比较器 B 高位输入：参考 DCAHCOMPSEL 描述
[7:4]	DCALCOMPSEL	RW	4'b0	选择数字比较器 A 低位输入：参考 DCAHCOMPSEL 描述
[3:0]	DCAHCOMPSEL	RW	4'b0	选择数字比较器 A 高位输入： 0000：跳闸输入 1 0001：跳闸输入 2 0010：跳闸输入 3 0011：跳闸输入 4 …

Puya Confidential

				1011：跳闸输入 12 1100：保留 1101：保留 1110：保留 1111：跳闸输入组合。通过 DCAHTRIPSEL 寄存器所选的全部跳闸输入进行逻辑或运行。

18.4.66 数字比较 A 控制寄存器（EPWMx_DCACTL）（x=1,2,3,4,5,6）

偏移地址：0x186

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EVT2FRCSYNCSEL	EVT2SRCSEL	Res.	Res.	Res.	Res.	EVT1SYNCE	EVT1SOCE	EVT1FRCSYNCSEL	EVT1SRCSEL
-	-	-	-	-	-	RW	RW	-	-	-	-	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:10]	reserved		6'b0	保留
[9]	EVT2FRCSYNCSEL	RW	1'b0	选择 DCAEVT2 强制同步信号： 0：与 EPWMCLK 同步 1：异步传递
[8]	EVT2SRCSEL	RW	1'b0	选择 DCAEVT2 信号源： 0：DCAEVT2 信号 1：DCEVTFILT 信号
[7:4]	reserved		4'b0	保留
[3]	EVT1SYNCE	RW	1'b0	使能 DCAEVT1 生成 SYNC： 0：禁止 1：使能
[2]	EVT1SOCE	RW	1'b0	使能 DCAEVT1 生成 SOC： 0：禁止 1：使能
[1]	EVT1FRCSYNCSEL	RW	1'b0	选择 DCAEVT1 强制同步信号： 0：与 EPWMCLK 同步 1：异步传递
[0]	EVT1SRCSEL	RW	1'b0	选择 DCAEVT1 信号源： 0：DCAEVT1 信号 1：DCEVTFILT 信号

18.4.67 数字比较 B 控制寄存器（EPWMx_DCBCTL）（x=1,2,3,4,5,6）

偏移地址：0x188

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EVT2FRCSYNCSEL	EVT2SRCSEL	Res.	Res.	Res.	Res.	EVT1SYNCE	EVT1SOCE	EVT1FRCSYNCSEL	EVT1SRCSEL
-	-	-	-	-	-	RW	RW	-	-	-	-	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:10]	reserved		6'b0	保留

[9]	EVT2FRCSYNCSEL	RW	1'b0	选择 DCBEVT2 强制同步信号： 0：与 EPWMCLK 同步 1：异步传递
[8]	EVT2SRCSEL	RW	1'b0	选择 DCBEVT2 信号源： 0：DCBEVT2 信号 1：DCEVTFILT 信号
[7:4]	reserved		4'b0	保留
[3]	EVT1SYNCE	RW	1'b0	使能 DCBEVT1 生成 SYNC： 0：禁止 1：使能
[2]	EVT1SOCE	RW	1'b0	使能 DCBEVT1 生成 SOC： 0：禁止 1：使能
[1]	EVT1FRCSYNCSEL	RW	1'b0	选择 DCBEVT1 强制同步信号： 0：与 EPWMCLK 同步 1：异步传递
[0]	EVT1SRCSEL	RW	1'b0	选择 DCBEVT1 信号源： 0：DCBEVT1 信号 1：DCEVTFILT 信号

18.4.68 数字比较滤波器控制寄存器（EPWMx_DCFCTL）（x=1,2,3,4,5,6）

偏移地址：0x18E

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDGESTATUS[2:0]			EDGECOUNT[2:0]			EDGEMODE[1:0]		Res.	EDGEFILTSEL	PULSESEL[1:0]		BLANKINV	BLANKE	SRCSEL[1:0]	
R	R	R	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:13]	EDGESTATUS	R	3'b0	边沿状态：该位反映当前捕获的边沿总数。当 EDGESTATUS 与 EDGECOUNT 匹配时，该位被设置为 0，并且在 DCEVTFILT 信号上生成一个 TBCLK 宽脉冲。边沿计数器可以通过将 000 写入 EDGECOUNT 值来复位。
[12:10]	EDGECOUNT	RW	3'b0	选择计数边沿数：该位用于选择在生成 DCEVTFILT 信号上的 TBCLK 宽脉冲之前计算边缘数。 000：不计算边沿 001：计算 1 个边沿 010：计算 2 个边沿 011：计算 3 个边沿 100：计算 4 个边沿 101：计算 5 个边沿 110：计算 6 个边沿 111：计算 7 个边沿
[9:8]	EDGEMODE	RW	2'b0	选择边沿模式： 00：上升沿 01：下降沿

				10：双边沿 11：保留
[7]	reserved		1'b0	保留
[6]	EDGEFILTSEL	RW	1'b0	选择边沿滤波器： 0：未选择 1：选择
[5:4]	PULSESEL	RW	2'b0	选择空白和捕获对齐脉冲： 00：TBCTR=TBPRD 01：TBCTR=0 10：TBCTR=0 或 TBCTR=TBPRD 11：保留
[3]	BLANKINV	RW	1'b0	翻转空白窗口： 0：未翻转 1：翻转
[2]	BLANKE	RW	1'b0	使能空白窗口： 0：禁止 1：使能
[1:0]	SRCSEL	RW	2'b0	选择滤波块信号源： 00：DCAEVT1 信号 01：DCAEVT2 信号 10：DCBEVT1 信号 11：DCBEVT2 信号

18.4.69 数字比较捕获控制寄存器（EPWMx_DCCAPCTL）（x=1,2,3,4,5,6）

偏移地址：0x190

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPMODE	CAPCLR	CAPSTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SHDWMODE	CAPE
RW	RS	R	-	-	-	-	-	-	-	-	-	-	-	RW	RW

Bit	Name	RW	ResetValue	Function
[15]	CAPMODE	RW	1'b0	计数器捕获模式： 0：当 CAPE=1 且发生 DCEVTFILT 事件时，将当前计数器值捕获到活动寄存器中。如果发生跳闸事件，则在下一个 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）重新触发捕获机制前，将一直忽略进一步的捕跳闸/获事件。影子模式：如果 SHDWMODE=0，当 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）时，活动寄存器值将被复制到影子寄存器中，则 CPU 读取该寄存器将返回影子寄存器值。活动模式：如果 SHDWMODE=1，则 CPU 读取该寄存器将返回活动寄存器值。1：当 CAPE=1 且发生 DCEVTFILT 事件时，将当前计数器值捕获到活动寄存器中。如果发生跳闸事件，则设置 CAPSTS 位并在清除该标志位前一直忽略进一步的捕跳闸/获事件。通过 CAPCLR 位可以清除该标志位并重新触发捕获机制。影子模式：SHDWMODE=0，当 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）时，活动寄存器值将被复制到影子寄存器中，则 CPU 读取该寄存器将返回影子

				寄存器值。活动模式：SHDWMODE=1，则 CPU 读取该寄存器将返回活动寄存器值。
[14]	CAPCLR	RS	1'b0	清除 DC 捕获锁存标志： 0：无效 1：清除 CAPSTS 位
[13]	CAPSTS	R	1'b0	DC 捕获事件锁存标志： 0：未发生 1：发生
[12:2]	reserved		11'b0	保留
[1]	SHDWMODE	RW	1'b0	选择 TBCTR 计数器捕获影子模式： 0：影子模式。当 TBCTR=0 或 TBCTR=TBPRD 时（由 PULSESEL 位定义），DCCAP 活动寄存器的值将被复制到影子寄存器。此时，从 DCCAP 寄存器中读取的值为影子寄存器值。1：活动模式。在该模式下，禁用影子寄存器且从 DCCAP 寄存器中读取的值始终为活动寄存器值。
[0]	CAPE	RW	1'b0	使能 TBCTL 计数器捕获： 0：禁止 1：使能

18.4.70 数字比较滤波器偏移寄存器（EPWMx_DCFOFFSET）（x=1,2,3,4,5,6）

偏移地址：0x192

复位值：0x0000

该寄存器为影子寄存器，活动寄存器在 PULSESEL 位定义的参考点装载。当活动寄存器装载时，初始化空白偏移计数器并开始向下计数。如果偏移计数器停止，则应用空白窗口。当空白窗口处于活动状态时，就会重新使能空白窗口计数器。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCFOFFSET[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:0]	DCFOFFSET	RW	16'b0	空白窗口：该位域用于指定空白窗口参考点（由 PULSESEL 位定义 TBCTR=0 或 TBCTR=TBPRD）到空白窗口应用点的 TBCLK 周期数。

18.4.71 数字比较滤波器偏移计数器寄存器（EPWMx_DCFOFFSETCNT）（x=1,2,3,4,5,6）

偏移地址：0x194

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCFOFFSETCNT[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
[15:0]	DCFOFFSETCNT	R	16'b0	空白偏移计数器：这 16 位位域用于表示当前偏移计数器的值。当该计数器向下计数达到 0 之后将会停止计数，直到下一个 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）重新装载计数器。即

				使设备由于仿真停止而暂停，该计数器也会继续向下计数，因此，偏移计数器不受自由或软件仿真位影响。

18.4.72 数字比较滤波器窗口寄存器（EPWMx_DCFWINDOW）（x=1,2,3,4,5,6）

偏移地址：0x196

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCFWINDOW[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:0]	DCFWINDOW	RW	16'b0	空白窗口宽度：0000：无空白窗口生成 0001...FFFF：指定空白窗口宽度，单位为 TBCLK 周期。当偏移计数器达到指定值时，空白窗口开始。此时，空白窗口计数器装载并且开始向下计数。需要注意避免当偏移计数器到达指定值并且空白窗口当前处于活动状态时，空白窗口计数器不会重新启动且空白窗口会提前结束的情况。空白窗口可以在 PWM 周期中的某个时间段内将信号暂时关闭。

18.4.73 数字比较滤波器窗口计数器寄存器（EPWMx_DCFWINDOWCNT）（x=1,2,3,4,5,6）

偏移地址：0x198

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCFWINDOWCNT[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
[15:0]	DCFWINDOWCNT	R	16'b0	空白窗口计数器：这 16 位位域用于表示当前窗口计数器的值。该计数器向下计数达到 0 之后就会停止计数，直到偏移计数器再次达到 0 时重新装载。

18.4.74 数字比较计数器捕获寄存器（EPWMx_DCCAP）（x=1,2,3,4,5,6）

偏移地址：0x19E

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCCAP[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
[15:0]	DCCAP	R	16'b0	数字比较时基计数器捕获：必须使 CAPE=1 才能使能 TBCTL 计数器捕获。当使能该功能时，在 DCEVTFLT 事件的上开沿触发时，会记录 TBCTR 的值。在 PULSESEL 位定义下一个 TBCTR=0 或 TBCTR=TBPRD 事件前，会忽略后面的捕获事件。通过 SHDWMODE 位使能 DCCAP 寄存器的影子模式（默认）。影子模式：SHDWMODE=0，当 TBCTR=0 或 TBCTR=TBPRD 事件（由

Puya Confidential

				PULSESEL 位定义）时，活动寄存器值将被复制到影子寄存器中，则 CPU 读取该寄存器将返回影子寄存器值。活动模式：SHDWMODE=1，则 CPU 读取该寄存器将返回活动寄存器值。活动寄存器和影子寄存器的内存映射地址相同。

18.4.75 数字比较 A 高跳闸选择寄存器（EPWMx_DCAHTRIPSEL）（x=1,2,3,4,5,6）

偏移地址：0x1A4

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9	TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11]	TRIPINPUT12	RW	1'b0	TRIP 输入 12： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[10]	TRIPINPUT11	RW	1'b0	TRIP 输入 11： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[9]	TRIPINPUT10	RW	1'b0	TRIP 输入 10： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[8]	TRIPINPUT9	RW	1'b0	TRIP 输入 9： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[7]	TRIPINPUT8	RW	1'b0	TRIP 输入 8： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[6]	TRIPINPUT7	RW	1'b0	TRIP 输入 7： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[5]	TRIPINPUT6	RW	1'b0	TRIP 输入 6： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[4]	TRIPINPUT5	RW	1'b0	TRIP 输入 5： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[3]	TRIPINPUT4	RW	1'b0	TRIP 输入 4： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[2]	TRIPINPUT3	RW	1'b0	TRIP 输入 3： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器

[1]	TRIPINPUT2	RW	1'b0	TRIP 输入 2： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器
[0]	TRIPINPUT1	RW	1'b0	TRIP 输入 1： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器

18.4.76 数字比较 A 低跳闸选择寄存器（EPWMx_DCALTRIPSEL）（x=1,2,3,4,5,6）

偏移地址：0x1A6

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9	TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11]	TRIPINPUT12	RW	1'b0	TRIP 输入 12： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[10]	TRIPINPUT11	RW	1'b0	TRIP 输入 11： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[9]	TRIPINPUT10	RW	1'b0	TRIP 输入 10： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[8]	TRIPINPUT9	RW	1'b0	TRIP 输入 9： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[7]	TRIPINPUT8	RW	1'b0	TRIP 输入 8： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[6]	TRIPINPUT7	RW	1'b0	TRIP 输入 7： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[5]	TRIPINPUT6	RW	1'b0	TRIP 输入 6： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[4]	TRIPINPUT5	RW	1'b0	TRIP 输入 5： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[3]	TRIPINPUT4	RW	1'b0	TRIP 输入 4： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[2]	TRIPINPUT3	RW	1'b0	TRIP 输入 3：

				0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[1]	TRIPINPUT2	RW	1'b0	TRIP 输入 2： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器
[0]	TRIPINPUT1	RW	1'b0	TRIP 输入 1： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器

18.4.77 数字比较 B 高跳闸选择寄存器（EPWMx_DCBHTRIPSEL）（x=1,2,3,4,5,6）

偏移地址：0x1A8

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9	TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11]	TRIPINPUT12	RW	1'b0	TRIP 输入 12： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[10]	TRIPINPUT11	RW	1'b0	TRIP 输入 11： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[9]	TRIPINPUT10	RW	1'b0	TRIP 输入 10： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[8]	TRIPINPUT9	RW	1'b0	TRIP 输入 9： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[7]	TRIPINPUT8	RW	1'b0	TRIP 输入 8： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[6]	TRIPINPUT7	RW	1'b0	TRIP 输入 7： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[5]	TRIPINPUT6	RW	1'b0	TRIP 输入 6： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[4]	TRIPINPUT5	RW	1'b0	TRIP 输入 5： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[3]	TRIPINPUT4	RW	1'b0	TRIP 输入 4： 0：未选择

Confidential

				1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[2]	TRIPINPUT3	RW	1'b0	TRIP 输入 3： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[1]	TRIPINPUT2	RW	1'b0	TRIP 输入 2： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器
[0]	TRIPINPUT1	RW	1'b0	TRIP 输入 1： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器

18.4.78 数字比较 B 低跳闸选择寄存器（EPWMx_DCBLTRIPSEL）（x=1,2,3,4,5,6）

偏移地址：0x1AA

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9	TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[15:12]	reserved		4'b0	保留
[11]	TRIPINPUT12	RW	1'b0	TRIP 输入 12： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[10]	TRIPINPUT11	RW	1'b0	TRIP 输入 11： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[9]	TRIPINPUT10	RW	1'b0	TRIP 输入 10： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[8]	TRIPINPUT9	RW	1'b0	TRIP 输入 9： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[7]	TRIPINPUT8	RW	1'b0	TRIP 输入 8： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[6]	TRIPINPUT7	RW	1'b0	TRIP 输入 7： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[5]	TRIPINPUT6	RW	1'b0	TRIP 输入 6： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[4]	TRIPINPUT5	RW	1'b0	TRIP 输入 5： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器

Confidential

[3]	TRIPINPUT4	RW	1'b0	TRIP 输入 4： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[2]	TRIPINPUT3	RW	1'b0	TRIP 输入 3： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[1]	TRIPINPUT2	RW	1'b0	TRIP 输入 2： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器
[0]	TRIPINPUT1	RW	1'b0	TRIP 输入 1： 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器

18.4.79 EPWM 锁定寄存器（EPWMx_EPWMLOCK）（x=1,2,3,4,5,6）

偏移地址：0x1F4

复位值： 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCLOCK	TZCLRLOCK	TZCFGLOCK	GLLOCK	HRLOCK
-	-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
[31:16]	KEY	W	16'b0	关键字（KEY） 只有将该位域的值设置为 0xA5A5 才能成功写入此寄存器。 当 KEY 匹配时，如果对该寄存器进行 16-位写操作，无 论是该寄存器的[15:0]还是[31:16]，则都会被忽略，只有 32 位写操作才能成功。
[15:5]	reserved		11'b0	保留
[4]	DCLOCK	RW	1'b0	数字比较锁定（Digital Compare LOCK） 数字比较寄存器地址映射范围为 0x180...0x1B2。 0：数字比较寄存器受到 EALLOW 保护 1：锁定数字比较寄存器且不可进行写操作
[3]	TZCLRLOCK	RW	1'b0	跳闸区清除锁定（Trip-zone Configure LOCK） 跳闸区清除寄存器地址映射范围为 0x12E...0x13G。 0：跳闸区清除寄存器受到 EALLOW 保护 1：锁定跳闸区清除寄存器且不可进行写操作
[2]	TZCFGLOCK	RW	1'b0	跳闸区配置锁定（Trip-zone Configure LOCK） 跳闸区配置寄存器地址映射范围为 0x100...0x11A。 0：跳闸区配置寄存器受到 EALLOW 保护 1：锁定跳闸区配置寄存器且不可进行写操作
[1]	GLLOCK	RW	1'b0	全局 EPWM 装载锁定（Global EPWM Load LOCK） 全局 EPWM 装载寄存器地址映射范围为 0x68...0x6A。 0：全局 EPWM 装载寄存器受到 EALLOW 保护 1：锁定全局 EPWM 装载寄存器且不可进行写操作
[0]	HRLOCK	RW	1'b0	高精度功能寄存器锁定（High Resolution EPWM LOCK）EPWM 寄存器地址映射范围为 0x40...0x5A。 0：高精度功能寄存器受到 EALLOW 保护 1：锁定 高精度功能 寄存器且不可进行写操作

18.4.80 硬件谷模式延迟寄存器（EPWMx_HWVDELVAL）（x=1,2,3,4,5,6）

偏移地址：0x1FA

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HWVDELVAL[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
[15:0]	HWVDELVAL	R	16'b0	硬件谷延迟值 这些位域用于表示 VDELAYDIV 位中计算的硬件谷延迟值。当触发谷捕获序列并更新谷捕获 1/2 值时，该硬件谷延迟值也同时更新。

18.4.81 硬件谷计数器寄存器（EPWMx_VCNTVAL）（x=1,2,3,4,5,6）

偏移地址：0x1FC

复位值： 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCNTVAL[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
[15:0]	VCNTVAL	R	16'b0	谷时基计数器：这些位域用于表示当在 VCNTCFG 寄存器中选择了 STOPEDGE 位时，该寄存器记录并保存捕获的谷计数器值。

18.5 EPWM 共用寄存器描述

18.5.1 INPUTXBAR1/2 选择寄存器（EPWM_IXBSEL1_2）

偏移地址：0x00

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL2[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL1[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:22]	reserved		10'b0	保留
[21:16]	INPUTSEL2	RW	6'b0	参考 INPUTSEL1
[15:6]	reserved		10'b0	保留
[5:0]	INPUTSEL1[5:0]	RW	6'b0	参考 INPUTSEL1 inputxbar1 的选择输入： 0x0 选择 EPWM_IN0 0x1 选择 EPWM_IN1 …

Puya Confidential

				0x3b 选择 EPWM_IN59 大于 0x3b 选择 EPWM_IN0

18.5.2 INPUTXBAR3/4 选择寄存器（EPWM_IXBSEL3_4）

偏移地址：0x04

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL4[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL3[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:22]	reserved		10'b0	保留
[21:16]	INPUTSEL4	RW	6'b0	参考 INPUTSEL1
[15:6]	reserved		10'b0	保留
[5:0]	INPUTSEL3	RW	6'b0	参考 INPUTSEL1

18.5.3 INPUTXBAR5/6 选择寄存器（EPWM_IXBSEL5_6）

偏移地址：0x08

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL6[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL5[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:22]	reserved		10'b0	保留
[21:16]	INPUTSEL6	RW	6'b0	参考 INPUTSEL1
[15:6]	reserved		10'b0	保留
[5:0]	INPUTSEL5	RW	6'b0	参考 INPUTSEL1

18.5.4 INPUTXBAR7/8 选择寄存器（EPWM_IXBSEL7_8）

偏移地址：0x0C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL8[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL7[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	Access	ResetValue	Function
[31:22]	reserved		10'b0	保留
[21:16]	INPUTSEL8	RW	6'b0	参考 INPUTSEL1
[15:6]	reserved		10'b0	保留
[5:0]	INPUTSEL7	RW	6'b0	参考 INPUTSEL1

18.5.5 INPUTXBAR9/10 选择寄存器（EPWM_IXBSEL9_10）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL10[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL9[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:22]	reserved		10'b0	保留
[21:16]	INPUTSEL10	RW	6'b0	参考 INPUTSEL1
[15:6]	reserved		10'b0	保留
[5:0]	INPUTSEL9	RW	6'b0	参考 INPUTSEL1

18.5.6 INPUTXBAR11/12 选择寄存器（EPWM_IXBSEL11_12）

偏移地址：0x14

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL12[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL11[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:22]	reserved		10'b0	保留
[21:16]	INPUTSEL12	RW	6'b0	参考 INPUTSEL1
[15:6]	reserved		10'b0	保留
[5:0]	INPUTSEL11	RW	6'b0	参考 INPUTSEL1

18.5.7 INPUTXBAR13/14 选择寄存器（EPWM_IXBSEL13_14）

偏移地址：0x18

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL14[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Puya Confidential

Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPUTSEL13[5:0]					
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:22]	reserved		10'b0	保留
[21:16]	INPUTSEL14	RW	6'b0	参考 INPUTSEL1
[15:6]	reserved		10'b0	保留
[5:0]	INPUTSEL13	RW	6'b0	参考 INPUTSEL1

18.5.8 INPUTXBAR 输出动作选择寄存器（EPWM_IXBAQSEL）

偏移地址：0x1C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	AQSEL14[1:0]		AQSEL13[1:0]		AQSEL12[1:0]		AQSEL11[1:0]		AQSEL10[1:0]		AQSEL9[1:0]	
-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AQSEL8[1:0]		AQSEL7[1:0]		AQSEL6[1:0]		AQSEL5[1:0]		AQSEL4[1:0]		AQSEL3[1:0]		AQSEL2[1:0]		AQSEL1[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:28]	reserved		4'b0	保留
[27:26]	AQSEL14	RW	2'b0	参考 AQSEL1
[25:24]	AQSEL13	RW	2'b0	参考 AQSEL1
[23:22]	AQSEL12	RW	2'b0	参考 AQSEL1
[21:20]	AQSEL11	RW	2'b0	参考 AQSEL1
[19:18]	AQSEL10	RW	2'b0	参考 AQSEL1
[17:16]	AQSEL9	RW	2'b0	参考 AQSEL1
[15:14]	AQSEL8	RW	2'b0	参考 AQSEL1
[13:12]	AQSEL7	RW	2'b0	参考 AQSEL1
[11:10]	AQSEL6	RW	2'b0	参考 AQSEL1
[9:8]	AQSEL5	RW	2'b0	参考 AQSEL1
[7:6]	AQSEL4	RW	2'b0	参考 AQSEL1
[5:4]	AQSEL3	RW	2'b0	参考 AQSEL1
[3:2]	AQSEL2	RW	2'b0	参考 AQSEL1
[1:0]	AQSEL1	RW	2'b0	INPUTXBAR1 的输出选择类型 0：同步方式 1：3 次采样判定 2：6 次采样判定 3：异步方式

18.5.9 INPUTXBAR 选择锁定寄存器（EPWM_IXBSEL_LOCK）

偏移地址：0x20

复位值： 0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FILTERLOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
RW	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	INPUTLOCK14	INPUTLOCK13	INPUTLOCK12	INPUTLOCK11	INPUTLOCK10	INPUTLOCK9	INPUTLOCK8	INPUTLOCK7	INPUTLOCK6	INPUTLOCK5	INPUTLOCK4	INPUTLOCK3	INPUTLOCK2	INPUTLOCK1
-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31]	FILTERLOCK	RW	1'b0	对 EPWM_IXBCRTL 寄存器的锁定标志。本寄存器被置 1 之后只能通过复位清 0 0：寄存器没有被锁定 1：寄存器被锁定
[30:14]	reserved		17'b0	保留
[13]	INPUTLOCK14	RW	1'b0	参考 INPUTLOCK1
[12]	INPUTLOCK13	RW	1'b0	参考 INPUTLOCK1
[11]	INPUTLOCK12	RW	1'b0	参考 INPUTLOCK1
[10]	INPUTLOCK11	RW	1'b0	参考 INPUTLOCK1
[9]	INPUTLOCK10	RW	1'b0	参考 INPUTLOCK1
[8]	INPUTLOCK9	RW	1'b0	参考 INPUTLOCK1
[7]	INPUTLOCK8	RW	1'b0	参考 INPUTLOCK1
[6]	INPUTLOCK7	RW	1'b0	参考 INPUTLOCK1
[5]	INPUTLOCK6	RW	1'b0	参考 INPUTLOCK1
[4]	INPUTLOCK5	RW	1'b0	参考 INPUTLOCK1
[3]	INPUTLOCK4	RW	1'b0	参考 INPUTLOCK1
[2]	INPUTLOCK3	RW	1'b0	参考 INPUTLOCK1
[1]	INPUTLOCK2	RW	1'b0	参考 INPUTLOCK1
[0]	INPUTLOCK1	RW	1'b0	对 INPUTXBAR1_SEL 寄存器的锁定标志。本寄存器被置 1 之后只能通过复位清 0 0：寄存器没有被锁定 1：寄存器被锁定

18.5.10 INPUTXBAR 动作采样周期控制寄存器（EPWM_IXBCTRL）

偏移地址：0x24

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISAPRD1[7:0]								ISAPRD0[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:16]	reserved		16'b0	保留

Puya Confidential

[15:8]	ISAPRD1	RW	8'b0	inputxbar8~14 的输入采样周期 0x00: 周期 = 0 系统时钟 0x01: 周期 = 2 系统时钟 0x02: 周期 = 4 系统时钟 0x03: 周期 = 6 系统时钟 ... 0xFF: 周期 = 510 系统时钟
[7:0]	ISAPRD0	RW	8'b0	inputxbar0~7 的输入采样周期 0x00: 周期 = 0 系统时钟 0x01: 周期 = 2 系统时钟 0x02: 周期 = 4 系统时钟 0x03: 周期 = 6 系统时钟 ... 0xFF: 周期 = 510 系统时钟

18.5.11 HRPMWXBAR4 MUX 选择寄存器 0（EPWM_EPXB4_0）

偏移地址：0x40

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15[1:0]		MUX14[1:0]		MUX13[1:0]		MUX12[1:0]		MUX11[1:0]		MUX10[1:0]		MUX9[1:0]		MUX8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX7[1:0]		MUX6[1:0]		MUX5[1:0]		MUX4[1:0]		MUX3[1:0]		MUX2[1:0]		MUX1[1:0]		MUX0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX15	RW	2'b0	参考 EPWMXBAR4_MUX0
[29:28]	MUX14	RW	2'b0	参考 EPWMXBAR4_MUX0
[27:26]	MUX13	RW	2'b0	参考 EPWMXBAR4_MUX0
[25:24]	MUX12	RW	2'b0	参考 EPWMXBAR4_MUX0
[23:22]	MUX11	RW	2'b0	参考 EPWMXBAR4_MUX0
[21:20]	MUX10	RW	2'b0	参考 EPWMXBAR4_MUX0
[19:18]	MUX9	RW	2'b0	参考 EPWMXBAR4_MUX0
[17:16]	MUX8	RW	2'b0	参考 EPWMXBAR4_MUX0
[15:14]	MUX7	RW	2'b0	参考 EPWMXBAR4_MUX0
[13:12]	MUX6	RW	2'b0	参考 EPWMXBAR4_MUX0
[11:10]	MUX5	RW	2'b0	参考 EPWMXBAR4_MUX0
[9:8]	MUX4	RW	2'b0	参考 EPWMXBAR4_MUX0
[7:6]	MUX3	RW	2'b0	参考 EPWMXBAR4_MUX0
[5:4]	MUX2	RW	2'b0	参考 EPWMXBAR4_MUX0
[3:2]	MUX1	RW	2'b0	参考 EPWMXBAR4_MUX0
[1:0]	MUX0	RW	2'b0	EPWMXBAR4_MUX0 输入选择（参考表格 EPWMX-BAR 配置表） 0：选择输入 0 1：选择输入 1

Puya Confidential

				2：选择输入 2 3：选择输入 3

18.5.12 HRPMWXBAR4 MUX 选择寄存器 1（EPWM_EPXB4_1）

偏移地址：0x44

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX31[1:0]		MUX30[1:0]		MUX29[1:0]		MUX28[1:0]		MUX27[1:0]		MUX26[1:0]		MUX25[1:0]		MUX24[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX23[1:0]		MUX22[1:0]		MUX21[1:0]		MUX20[1:0]		MUX19[1:0]		MUX18[1:0]		MUX17[1:0]		MUX16[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX31	RW	2'b0	参考 EPWMXBAR4_MUX16
[29:28]	MUX30	RW	2'b0	参考 EPWMXBAR4_MUX16
[27:26]	MUX29	RW	2'b0	参考 EPWMXBAR4_MUX16
[25:24]	MUX28	RW	2'b0	参考 EPWMXBAR4_MUX16
[23:22]	MUX27	RW	2'b0	参考 EPWMXBAR4_MUX16
[21:20]	MUX26	RW	2'b0	参考 EPWMXBAR4_MUX16
[19:18]	MUX25	RW	2'b0	参考 EPWMXBAR4_MUX16
[17:16]	MUX24	RW	2'b0	参考 EPWMXBAR4_MUX16
[15:14]	MUX23	RW	2'b0	参考 EPWMXBAR4_MUX16
[13:12]	MUX22	RW	2'b0	参考 EPWMXBAR4_MUX16
[11:10]	MUX21	RW	2'b0	参考 EPWMXBAR4_MUX16
[9:8]	MUX20	RW	2'b0	参考 EPWMXBAR4_MUX16
[7:6]	MUX19	RW	2'b0	参考 EPWMXBAR4_MUX16
[5:4]	MUX18	RW	2'b0	参考 EPWMXBAR4_MUX16
[3:2]	MUX17	RW	2'b0	参考 EPWMXBAR4_MUX16
[1:0]	MUX16	RW	2'b0	EPWMXBAR4_MUX16 输入选择（参考表格 EPWMX-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.13 HRPMWXBAR5 MUX 选择寄存器 0（EPWM_EPXB5_0）

偏移地址：0x48

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15[1:0]		MUX14[1:0]		MUX13[1:0]		MUX12[1:0]		MUX11[1:0]		MUX10[1:0]		MUX9[1:0]		MUX8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX7[1:0]		MUX6[1:0]		MUX5[1:0]		MUX4[1:0]		MUX3[1:0]		MUX2[1:0]		MUX1[1:0]		MUX0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	Access	ResetValue	Function
[31:30]	MUX15	RW	2'b0	参考 EPWMXBAR5_MUX0
[29:28]	MUX14	RW	2'b0	参考 EPWMXBAR5_MUX0
[27:26]	MUX13	RW	2'b0	参考 EPWMXBAR5_MUX0
[25:24]	MUX12	RW	2'b0	参考 EPWMXBAR5_MUX0
[23:22]	MUX11	RW	2'b0	参考 EPWMXBAR5_MUX0
[21:20]	MUX10	RW	2'b0	参考 EPWMXBAR5_MUX0
[19:18]	MUX9	RW	2'b0	参考 EPWMXBAR5_MUX0
[17:16]	MUX8	RW	2'b0	参考 EPWMXBAR5_MUX0
[15:14]	MUX7	RW	2'b0	参考 EPWMXBAR5_MUX0
[13:12]	MUX6	RW	2'b0	参考 EPWMXBAR5_MUX0
[11:10]	MUX5	RW	2'b0	参考 EPWMXBAR5_MUX0
[9:8]	MUX4	RW	2'b0	参考 EPWMXBAR5_MUX0
[7:6]	MUX3	RW	2'b0	参考 EPWMXBAR5_MUX0
[5:4]	MUX2	RW	2'b0	参考 EPWMXBAR5_MUX0
[3:2]	MUX1	RW	2'b0	参考 EPWMXBAR5_MUX0
[1:0]	MUX0	RW	2'b0	EPWMXBAR5_MUX0 输入选择（参考表格 EPWM X-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.14 HRPMWXBAR5 MUX 选择寄存器 1（EPWM_EPXB5_1）

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX31[1:0]		MUX30[1:0]		MUX29[1:0]		MUX28[1:0]		MUX27[1:0]		MUX26[1:0]		MUX25[1:0]		MUX24[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX23[1:0]		MUX22[1:0]		MUX21[1:0]		MUX20[1:0]		MUX19[1:0]		MUX18[1:0]		MUX17[1:0]		MUX16[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX31	RW	2'b0	参考 EPWMXBAR5_MUX16
[29:28]	MUX30	RW	2'b0	参考 EPWMXBAR5_MUX16
[27:26]	MUX29	RW	2'b0	参考 EPWMXBAR5_MUX16
[25:24]	MUX28	RW	2'b0	参考 EPWMXBAR5_MUX16
[23:22]	MUX27	RW	2'b0	参考 EPWMXBAR5_MUX16
[21:20]	MUX26	RW	2'b0	参考 EPWMXBAR5_MUX16
[19:18]	MUX25	RW	2'b0	参考 EPWMXBAR5_MUX16
[17:16]	MUX24	RW	2'b0	参考 EPWMXBAR5_MUX16
[15:14]	MUX23	RW	2'b0	参考 EPWMXBAR5_MUX16
[13:12]	MUX22	RW	2'b0	参考 EPWMXBAR5_MUX16
[11:10]	MUX21	RW	2'b0	参考 EPWMXBAR5_MUX16

Puya Confidential

[9:8]	MUX20	RW	2'b0	参考 EPWMXBAR5_MUX16
[7:6]	MUX19	RW	2'b0	参考 EPWMXBAR5_MUX16
[5:4]	MUX18	RW	2'b0	参考 EPWMXBAR5_MUX16
[3:2]	MUX17	RW	2'b0	参考 EPWMXBAR5_MUX16
[1:0]	MUX16	RW	2'b0	EPWMXBAR5_MUX16 输入选择（参考表格 EPWM X-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.15 HRPMWXBAR7 MUX 选择寄存器 0（EPWM_EPXB7_0）

偏移地址：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15[1:0]		MUX14[1:0]		MUX13[1:0]		MUX12[1:0]		MUX11[1:0]		MUX10[1:0]		MUX9[1:0]		MUX8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX7[1:0]		MUX6[1:0]		MUX5[1:0]		MUX4[1:0]		MUX3[1:0]		MUX2[1:0]		MUX1[1:0]		MUX0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX15	RW	2'b0	参考 EPWMXBAR7_MUX0
[29:28]	MUX14	RW	2'b0	参考 EPWMXBAR7_MUX0
[27:26]	MUX13	RW	2'b0	参考 EPWMXBAR7_MUX0
[25:24]	MUX12	RW	2'b0	参考 EPWMXBAR7_MUX0
[23:22]	MUX11	RW	2'b0	参考 EPWMXBAR7_MUX0
[21:20]	MUX10	RW	2'b0	参考 EPWMXBAR7_MUX0
[19:18]	MUX9	RW	2'b0	参考 EPWMXBAR7_MUX0
[17:16]	MUX8	RW	2'b0	参考 EPWMXBAR7_MUX0
[15:14]	MUX7	RW	2'b0	参考 EPWMXBAR7_MUX0
[13:12]	MUX6	RW	2'b0	参考 EPWMXBAR7_MUX0
[11:10]	MUX5	RW	2'b0	参考 EPWMXBAR7_MUX0
[9:8]	MUX4	RW	2'b0	参考 EPWMXBAR7_MUX0
[7:6]	MUX3	RW	2'b0	参考 EPWMXBAR7_MUX0
[5:4]	MUX2	RW	2'b0	参考 EPWMXBAR7_MUX0
[3:2]	MUX1	RW	2'b0	参考 EPWMXBAR7_MUX0
[1:0]	MUX0	RW	2'b0	EPWMXBAR7_MUX0 输入选择（参考表格 EPWM X-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

Puya Confidential

18.5.16 HRPMWXBAR7 MUX 选择寄存器 1（EPWM_EPXB7_1）

偏移地址：0x54

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX31[1:0]		MUX30[1:0]		MUX29[1:0]		MUX28[1:0]		MUX27[1:0]		MUX26[1:0]		MUX25[1:0]		MUX24[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX23[1:0]		MUX22[1:0]		MUX21[1:0]		MUX20[1:0]		MUX19[1:0]		MUX18[1:0]		MUX17[1:0]		MUX16[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX31	RW	2'b0	参考 EPWMXBAR7_MUX16
[29:28]	MUX30	RW	2'b0	参考 EPWMXBAR7_MUX16
[27:26]	MUX29	RW	2'b0	参考 EPWMXBAR7_MUX16
[25:24]	MUX28	RW	2'b0	参考 EPWMXBAR7_MUX16
[23:22]	MUX27	RW	2'b0	参考 EPWMXBAR7_MUX16
[21:20]	MUX26	RW	2'b0	参考 EPWMXBAR7_MUX16
[19:18]	MUX25	RW	2'b0	参考 EPWMXBAR7_MUX16
[17:16]	MUX24	RW	2'b0	参考 EPWMXBAR7_MUX16
[15:14]	MUX23	RW	2'b0	参考 EPWMXBAR7_MUX16
[13:12]	MUX22	RW	2'b0	参考 EPWMXBAR7_MUX16
[11:10]	MUX21	RW	2'b0	参考 EPWMXBAR7_MUX16
[9:8]	MUX20	RW	2'b0	参考 EPWMXBAR7_MUX16
[7:6]	MUX19	RW	2'b0	参考 EPWMXBAR7_MUX16
[5:4]	MUX18	RW	2'b0	参考 EPWMXBAR7_MUX16
[3:2]	MUX17	RW	2'b0	参考 EPWMXBAR7_MUX16
[1:0]	MUX16	RW	2'b0	EPWMXBAR7_MUX16 输入选择 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

Puya Confidential

18.5.17 HRPMWXBAR8 MUX 选择寄存器 0（EPWM_EPXB8_0）

偏移地址：0x58

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15[1:0]		MUX14[1:0]		MUX13[1:0]		MUX12[1:0]		MUX11[1:0]		MUX10[1:0]		MUX9[1:0]		MUX8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX7[1:0]		MUX6[1:0]		MUX5[1:0]		MUX4[1:0]		MUX3[1:0]		MUX2[1:0]		MUX1[1:0]		MUX0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX15	RW	2'b0	参考 EPWMXBAR8_MUX0
[29:28]	MUX14	RW	2'b0	参考 EPWMXBAR8_MUX0
[27:26]	MUX13	RW	2'b0	参考 EPWMXBAR8_MUX0
[25:24]	MUX12	RW	2'b0	参考 EPWMXBAR8_MUX0
[23:22]	MUX11	RW	2'b0	参考 EPWMXBAR8_MUX0
[21:20]	MUX10	RW	2'b0	参考 EPWMXBAR8_MUX0
[19:18]	MUX9	RW	2'b0	参考 EPWMXBAR8_MUX0
[17:16]	MUX8	RW	2'b0	参考 EPWMXBAR8_MUX0
[15:14]	MUX7	RW	2'b0	参考 EPWMXBAR8_MUX0
[13:12]	MUX6	RW	2'b0	参考 EPWMXBAR8_MUX0
[11:10]	MUX5	RW	2'b0	参考 EPWMXBAR8_MUX0
[9:8]	MUX4	RW	2'b0	参考 EPWMXBAR8_MUX0
[7:6]	MUX3	RW	2'b0	参考 EPWMXBAR8_MUX0
[5:4]	MUX2	RW	2'b0	参考 EPWMXBAR8_MUX0
[3:2]	MUX1	RW	2'b0	参考 EPWMXBAR8_MUX0
[1:0]	MUX0	RW	2'b0	EPWMXBAR8_MUX0 输入选择（参考表格 EPWM X-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.18 HRPMWXBAR8 MUX 选择寄存器 1（EPWM_EPXB8_1）

偏移地址：0x5C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX31[1:0]		MUX30[1:0]		MUX29[1:0]		MUX28[1:0]		MUX27[1:0]		MUX26[1:0]		MUX25[1:0]		MUX24[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX23[1:0]		MUX22[1:0]		MUX21[1:0]		MUX20[1:0]		MUX19[1:0]		MUX18[1:0]		MUX17[1:0]		MUX16[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX31	RW	2'b0	参考 EPWMXBAR8_MUX16

Puya Confidential

[29:28]	MUX30	RW	2'b0	参考 EPWMXBAR8_MUX16
[27:26]	MUX29	RW	2'b0	参考 EPWMXBAR8_MUX16
[25:24]	MUX28	RW	2'b0	参考 EPWMXBAR8_MUX16
[23:22]	MUX27	RW	2'b0	参考 EPWMXBAR8_MUX16
[21:20]	MUX26	RW	2'b0	参考 EPWMXBAR8_MUX16
[19:18]	MUX25	RW	2'b0	参考 EPWMXBAR8_MUX16
[17:16]	MUX24	RW	2'b0	参考 EPWMXBAR8_MUX16
[15:14]	MUX23	RW	2'b0	参考 EPWMXBAR8_MUX16
[13:12]	MUX22	RW	2'b0	参考 EPWMXBAR8_MUX16
[11:10]	MUX21	RW	2'b0	参考 EPWMXBAR8_MUX16
[9:8]	MUX20	RW	2'b0	参考 EPWMXBAR8_MUX16
[7:6]	MUX19	RW	2'b0	参考 EPWMXBAR8_MUX16
[5:4]	MUX18	RW	2'b0	参考 EPWMXBAR8_MUX16
[3:2]	MUX17	RW	2'b0	参考 EPWMXBAR8_MUX16
[1:0]	MUX16	RW	2'b0	EPWMXBAR8_MUX16 输入选择 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.19 HRPMWXBAR9 MUX 选择寄存器 0（EPWM_EPXB9_0）

偏移地址：0x60

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15[1:0]		MUX14[1:0]		MUX13[1:0]		MUX12[1:0]		MUX11[1:0]		MUX10[1:0]		MUX9[1:0]		MUX8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX7[1:0]		MUX6[1:0]		MUX5[1:0]		MUX4[1:0]		MUX3[1:0]		MUX2[1:0]		MUX1[1:0]		MUX0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX15	RW	2'b0	参考 EPWMXBAR9_MUX0
[29:28]	MUX14	RW	2'b0	参考 EPWMXBAR9_MUX0
[27:26]	MUX13	RW	2'b0	参考 EPWMXBAR9_MUX0
[25:24]	MUX12	RW	2'b0	参考 EPWMXBAR9_MUX0
[23:22]	MUX11	RW	2'b0	参考 EPWMXBAR9_MUX0
[21:20]	MUX10	RW	2'b0	参考 EPWMXBAR9_MUX0
[19:18]	MUX9	RW	2'b0	参考 EPWMXBAR9_MUX0
[17:16]	MUX8	RW	2'b0	参考 EPWMXBAR9_MUX0
[15:14]	MUX7	RW	2'b0	参考 EPWMXBAR9_MUX0
[13:12]	MUX6	RW	2'b0	参考 EPWMXBAR9_MUX0
[11:10]	MUX5	RW	2'b0	参考 EPWMXBAR9_MUX0
[9:8]	MUX4	RW	2'b0	参考 EPWMXBAR9_MUX0
[7:6]	MUX3	RW	2'b0	参考 EPWMXBAR9_MUX0
[5:4]	MUX2	RW	2'b0	参考 EPWMXBAR9_MUX0

Puya Confidential

[3:2]	MUX1	RW	2'b0	参考 EPWMXBAR9_MUX0
[1:0]	MUX0	RW	2'b0	EPWMXBAR9_MUX0 输入选择（参考表格 EPWMX-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.20 HRPMWXBAR9 MUX 选择寄存器 1（EPWM_EPXB9_1）

偏移地址：0x64

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX31[1:0]		MUX30[1:0]		MUX29[1:0]		MUX28[1:0]		MUX27[1:0]		MUX26[1:0]		MUX25[1:0]		MUX24[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX23[1:0]		MUX22[1:0]		MUX21[1:0]		MUX20[1:0]		MUX19[1:0]		MUX18[1:0]		MUX17[1:0]		MUX16[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX31	RW	2'b0	参考 EPWMXBAR9_MUX16
[29:28]	MUX30	RW	2'b0	参考 EPWMXBAR9_MUX16
[27:26]	MUX29	RW	2'b0	参考 EPWMXBAR9_MUX16
[25:24]	MUX28	RW	2'b0	参考 EPWMXBAR9_MUX16
[23:22]	MUX27	RW	2'b0	参考 EPWMXBAR9_MUX16
[21:20]	MUX26	RW	2'b0	参考 EPWMXBAR9_MUX16
[19:18]	MUX25	RW	2'b0	参考 EPWMXBAR9_MUX16
[17:16]	MUX24	RW	2'b0	参考 EPWMXBAR9_MUX16
[15:14]	MUX23	RW	2'b0	参考 EPWMXBAR9_MUX16
[13:12]	MUX22	RW	2'b0	参考 EPWMXBAR9_MUX16
[11:10]	MUX21	RW	2'b0	参考 EPWMXBAR9_MUX16
[9:8]	MUX20	RW	2'b0	参考 EPWMXBAR9_MUX16
[7:6]	MUX19	RW	2'b0	参考 EPWMXBAR9_MUX16
[5:4]	MUX18	RW	2'b0	参考 EPWMXBAR9_MUX16
[3:2]	MUX17	RW	2'b0	参考 EPWMXBAR9_MUX16
[1:0]	MUX16	RW	2'b0	EPWMXBAR9_MUX16 输入选择 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.21 HRPMWXBAR10 MUX 选择寄存器 0（EPWM_EPXB10_0）

偏移地址：0x68

复位值： 0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15[1:0]		MUX14[1:0]		MUX13[1:0]		MUX12[1:0]		MUX11[1:0]		MUX10[1:0]		MUX9[1:0]		MUX8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX7[1:0]		MUX6[1:0]		MUX5[1:0]		MUX4[1:0]		MUX3[1:0]		MUX2[1:0]		MUX1[1:0]		MUX0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX15	RW	2'b0	参考 EPWMXBAR10_MUX0
[29:28]	MUX14	RW	2'b0	参考 EPWMXBAR10_MUX0
[27:26]	MUX13	RW	2'b0	参考 EPWMXBAR10_MUX0
[25:24]	MUX12	RW	2'b0	参考 EPWMXBAR10_MUX0
[23:22]	MUX11	RW	2'b0	参考 EPWMXBAR10_MUX0
[21:20]	MUX10	RW	2'b0	参考 EPWMXBAR10_MUX0
[19:18]	MUX9	RW	2'b0	参考 EPWMXBAR10_MUX0
[17:16]	MUX8	RW	2'b0	参考 EPWMXBAR10_MUX0
[15:14]	MUX7	RW	2'b0	参考 EPWMXBAR10_MUX0
[13:12]	MUX6	RW	2'b0	参考 EPWMXBAR10_MUX0
[11:10]	MUX5	RW	2'b0	参考 EPWMXBAR10_MUX0
[9:8]	MUX4	RW	2'b0	参考 EPWMXBAR10_MUX0
[7:6]	MUX3	RW	2'b0	参考 EPWMXBAR10_MUX0
[5:4]	MUX2	RW	2'b0	参考 EPWMXBAR10_MUX0
[3:2]	MUX1	RW	2'b0	参考 EPWMXBAR10_MUX0
[1:0]	MUX0	RW	2'b0	EPWMXBAR10_MUX0 输入选择（参考表格 EPWMX-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.22 HRPMWXBAR10 MUX 选择寄存器 1（EPWM_EPXB10_1）

偏移地址：0x6C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX31[1:0]		MUX30[1:0]		MUX29[1:0]		MUX28[1:0]		MUX27[1:0]		MUX26[1:0]		MUX25[1:0]		MUX24[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX23[1:0]		MUX22[1:0]		MUX21[1:0]		MUX20[1:0]		MUX19[1:0]		MUX18[1:0]		MUX17[1:0]		MUX16[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX31	RW	2'b0	参考 EPWMXBAR10_MUX16
[29:28]	MUX30	RW	2'b0	参考 EPWMXBAR10_MUX16
[27:26]	MUX29	RW	2'b0	参考 EPWMXBAR10_MUX16
[25:24]	MUX28	RW	2'b0	参考 EPWMXBAR10_MUX16

Puya Confidential

[23:22]	MUX27	RW	2'b0	参考 EPWMXBAR10_MUX16
[21:20]	MUX26	RW	2'b0	参考 EPWMXBAR10_MUX16
[19:18]	MUX25	RW	2'b0	参考 EPWMXBAR10_MUX16
[17:16]	MUX24	RW	2'b0	参考 EPWMXBAR10_MUX16
[15:14]	MUX23	RW	2'b0	参考 EPWMXBAR10_MUX16
[13:12]	MUX22	RW	2'b0	参考 EPWMXBAR10_MUX16
[11:10]	MUX21	RW	2'b0	参考 EPWMXBAR10_MUX16
[9:8]	MUX20	RW	2'b0	参考 EPWMXBAR10_MUX16
[7:6]	MUX19	RW	2'b0	参考 EPWMXBAR10_MUX16
[5:4]	MUX18	RW	2'b0	参考 EPWMXBAR10_MUX16
[3:2]	MUX17	RW	2'b0	参考 EPWMXBAR10_MUX16
[1:0]	MUX16	RW	2'b0	EPWMXBAR10_MUX16 输入选择（参考表格 EPWM X-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.23 HRPMWXBAR11 MUX 选择寄存器 0（EPWM_EPXB11_0）

偏移地址：0x70

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15[1:0]		MUX14[1:0]		MUX13[1:0]		MUX12[1:0]		MUX11[1:0]		MUX10[1:0]		MUX9[1:0]		MUX8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX7[1:0]		MUX6[1:0]		MUX5[1:0]		MUX4[1:0]		MUX3[1:0]		MUX2[1:0]		MUX1[1:0]		MUX0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX15	RW	2'b0	参考 EPWMXBAR11_MUX0
[29:28]	MUX14	RW	2'b0	参考 EPWMXBAR11_MUX0
[27:26]	MUX13	RW	2'b0	参考 EPWMXBAR11_MUX0
[25:24]	MUX12	RW	2'b0	参考 EPWMXBAR11_MUX0
[23:22]	MUX11	RW	2'b0	参考 EPWMXBAR11_MUX0
[21:20]	MUX10	RW	2'b0	参考 EPWMXBAR11_MUX0
[19:18]	MUX9	RW	2'b0	参考 EPWMXBAR11_MUX0
[17:16]	MUX8	RW	2'b0	参考 EPWMXBAR11_MUX0
[15:14]	MUX7	RW	2'b0	参考 EPWMXBAR11_MUX0
[13:12]	MUX6	RW	2'b0	参考 EPWMXBAR11_MUX0
[11:10]	MUX5	RW	2'b0	参考 EPWMXBAR11_MUX0
[9:8]	MUX4	RW	2'b0	参考 EPWMXBAR11_MUX0
[7:6]	MUX3	RW	2'b0	参考 EPWMXBAR11_MUX0
[5:4]	MUX2	RW	2'b0	参考 EPWMXBAR11_MUX0
[3:2]	MUX1	RW	2'b0	参考 EPWMXBAR11_MUX0

Puya Confidential

[1:0]	MUX0	RW	2'b0	EPWMXBAR11_MUX0 输入选择（参考表格 EPWMX-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.24 HRPMWXBAR11 MUX 选择寄存器 1（EPWM_EPXB11_1）

偏移地址：0x74

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX31[1:0]		MUX30[1:0]		MUX29[1:0]		MUX28[1:0]		MUX27[1:0]		MUX26[1:0]		MUX25[1:0]		MUX24[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX23[1:0]		MUX22[1:0]		MUX21[1:0]		MUX20[1:0]		MUX19[1:0]		MUX18[1:0]		MUX17[1:0]		MUX16[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX31	RW	2'b0	参考 EPWMXBAR11_MUX16
[29:28]	MUX30	RW	2'b0	参考 EPWMXBAR11_MUX16
[27:26]	MUX29	RW	2'b0	参考 EPWMXBAR11_MUX16
[25:24]	MUX28	RW	2'b0	参考 EPWMXBAR11_MUX16
[23:22]	MUX27	RW	2'b0	参考 EPWMXBAR11_MUX16
[21:20]	MUX26	RW	2'b0	参考 EPWMXBAR11_MUX16
[19:18]	MUX25	RW	2'b0	参考 EPWMXBAR11_MUX16
[17:16]	MUX24	RW	2'b0	参考 EPWMXBAR11_MUX16
[15:14]	MUX23	RW	2'b0	参考 EPWMXBAR11_MUX16
[13:12]	MUX22	RW	2'b0	参考 EPWMXBAR11_MUX16
[11:10]	MUX21	RW	2'b0	参考 EPWMXBAR11_MUX16
[9:8]	MUX20	RW	2'b0	参考 EPWMXBAR11_MUX16
[7:6]	MUX19	RW	2'b0	参考 EPWMXBAR11_MUX16
[5:4]	MUX18	RW	2'b0	参考 EPWMXBAR11_MUX16
[3:2]	MUX17	RW	2'b0	参考 EPWMXBAR11_MUX16
[1:0]	MUX16	RW	2'b0	EPWMXBAR11_MUX16 输入选择 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.25 HRPMWXBAR12 MUX 选择寄存器 0（EPWM_EPXB12_0）

偏移地址：0x78

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15[1:0]		MUX14[1:0]		MUX13[1:0]		MUX12[1:0]		MUX11[1:0]		MUX10[1:0]		MUX9[1:0]		MUX8[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Puya Confidential

MUX7[1:0]		MUX6[1:0]		MUX5[1:0]		MUX4[1:0]		MUX3[1:0]		MUX2[1:0]		MUX1[1:0]		MUX0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX15	RW	2'b0	参考 EPWMXBAR12_MUX0
[29:28]	MUX14	RW	2'b0	参考 EPWMXBAR12_MUX0
[27:26]	MUX13	RW	2'b0	参考 EPWMXBAR12_MUX0
[25:24]	MUX12	RW	2'b0	参考 EPWMXBAR12_MUX0
[23:22]	MUX11	RW	2'b0	参考 EPWMXBAR12_MUX0
[21:20]	MUX10	RW	2'b0	参考 EPWMXBAR12_MUX0
[19:18]	MUX9	RW	2'b0	参考 EPWMXBAR12_MUX0
[17:16]	MUX8	RW	2'b0	参考 EPWMXBAR12_MUX0
[15:14]	MUX7	RW	2'b0	参考 EPWMXBAR12_MUX0
[13:12]	MUX6	RW	2'b0	参考 EPWMXBAR12_MUX0
[11:10]	MUX5	RW	2'b0	参考 EPWMXBAR12_MUX0
[9:8]	MUX4	RW	2'b0	参考 EPWMXBAR12_MUX0
[7:6]	MUX3	RW	2'b0	参考 EPWMXBAR12_MUX0
[5:4]	MUX2	RW	2'b0	参考 EPWMXBAR12_MUX0
[3:2]	MUX1	RW	2'b0	参考 EPWMXBAR12_MUX0
[1:0]	MUX0	RW	2'b0	EPWMXBAR12_MUX0 输入选择（参考表格 EPWMX-BAR 配置表） 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.26 HRPMWXBAR12 MUX 选择寄存器 1（EPWM_EPXB12_1）

偏移地址：0x7C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX31[1:0]		MUX30[1:0]		MUX29[1:0]		MUX28[1:0]		MUX27[1:0]		MUX26[1:0]		MUX25[1:0]		MUX24[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUX23[1:0]		MUX22[1:0]		MUX21[1:0]		MUX20[1:0]		MUX19[1:0]		MUX18[1:0]		MUX17[1:0]		MUX16[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:30]	MUX31	RW	2'b0	参考 EPWMXBAR12_MUX16
[29:28]	MUX30	RW	2'b0	参考 EPWMXBAR12_MUX16
[27:26]	MUX29	RW	2'b0	参考 EPWMXBAR12_MUX16
[25:24]	MUX28	RW	2'b0	参考 EPWMXBAR12_MUX16
[23:22]	MUX27	RW	2'b0	参考 EPWMXBAR12_MUX16
[21:20]	MUX26	RW	2'b0	参考 EPWMXBAR12_MUX16
[19:18]	MUX25	RW	2'b0	参考 EPWMXBAR12_MUX16
[17:16]	MUX24	RW	2'b0	参考 EPWMXBAR12_MUX16

Puya Confidential

[15:14]	MUX23	RW	2'b0	参考 EPWMXBAR12_MUX16
[13:12]	MUX22	RW	2'b0	参考 EPWMXBAR12_MUX16
[11:10]	MUX21	RW	2'b0	参考 EPWMXBAR12_MUX16
[9:8]	MUX20	RW	2'b0	参考 EPWMXBAR12_MUX16
[7:6]	MUX19	RW	2'b0	参考 EPWMXBAR12_MUX16
[5:4]	MUX18	RW	2'b0	参考 EPWMXBAR12_MUX16
[3:2]	MUX17	RW	2'b0	参考 EPWMXBAR12_MUX16
[1:0]	MUX16	RW	2'b0	EPWMXBAR12_MUX16 输入选择 0：选择输入 0 1：选择输入 1 2：选择输入 2 3：选择输入 3

18.5.27 EPWMXBAR4 MUX 使能寄存器（EPWM_EPXB4_MUX）

偏移地址：0x80

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUXEN[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUXEN[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:0]	MUXEN	RW	32'b0	EPWMXBAR4 的 32 个 MUX 使能位（EPWMXBAR4_MUXEN），对应的 bit 为代表该 mux 参与 EPWMXBAR4 的输出，0 代表关闭对 EPWMXBAR4 的输出的参与

18.5.28 EPWMXBAR5 MUX 使能寄存器（EPWM_EPXB5_MUX）

偏移地址：0x84

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUXEN[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUXEN[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:0]	MUXEN	RW	32'b0	参考 EPWMXBAR4_MUXEN

18.5.29 EPWMXBAR7 MUX 使能寄存器（EPWM_EPXB7_MUX）

偏移地址：0x88

复位值： 0x0000 0000

Puya Confidential

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

Bit	Name	Access	ResetValue	Function
[31:0]	MUXEN	RW	32'b0	参考 EPWMXBAR4_MUXEN

18.5.30 EPWMXBAR8 MUX 使能寄存器（EPWM_EPXB8_MUX）

偏移地址：0x8C

复位值： 0x0000 0000

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

Bit	Name	Access	ResetValue	Function
[31:0]	MUXEN	RW	32'b0	参考 EPWMXBAR4_MUXEN

18.5.31 EPWMXBAR9 MUX 使能寄存器（EPWM_EPXB9_MUX）

偏移地址：0x90

复位值： 0x0000 0000

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

Bit	Name	Access	ResetValue	Function
[31:0]	MUXEN	RW	32'b0	参考 EPWMXBAR4_MUXEN

18.5.32 EPWMXBAR10 MUX 使能寄存器（EPWM_EPXB10_MUX）

偏移地址：0x94

复位值： 0x0000 0000

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

Puya Confidential

Bit	Name	Access	ResetValue	Function
[31:0]	MUXEN	RW	32'b0	参考 EPWMXBAR4_MUXEN

18.5.33 EPWMXBAR11 MUX 使能寄存器（EPWM_EPXB11_MUX）

偏移地址：0x98

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUXEN[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUXEN[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:0]	MUXEN	RW	32'b0	参考 EPWMXBAR4_MUXEN

18.5.34 EPWMXBAR12 MUX 使能寄存器（EPWM_EPXB12_MUX）

偏移地址：0x9C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUXEN[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MUXEN[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:0]	MUXEN	RW	32'b0	参考 EPWMXBAR4_MUXEN

18.5.35 EPWMXBAR 输出极性选择寄存器（EPWM_EPXB_POL）

偏移地址：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPXB12P	EPXB11P	EPXB10P	EPXB9P	EPXB8P	EPXB7P	EPXB5P	EPXB4P
-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:8]	reserved		24'b0	保留
[7]	EPXB12P	RW	1'b0	参考 EPXB4P
[6]	EPXB11P	RW	1'b0	参考 EPXB4P

Puya Confidential

[5]	EPXB10P	RW	1'b0	参考 EPXB4P
[4]	EPXB9P	RW	1'b0	参考 EPXB4P
[3]	EPXB8P	RW	1'b0	参考 EPXB4P
[2]	EPXB7P	RW	1'b0	参考 EPXB4P
[1]	EPXB5P	RW	1'b0	参考 EPXB4P
[0]	EPXB4P	RW	1'b0	EPWMXBAR4 的输出极性 0：高电平有效 1：低电平有效

18.5.36 HRPMMXBAR 锁定寄存器（EPWM_EPXBLOCK）

偏移地址：0xA4

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPXBLOCK
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW

Bits	Name	Access	ResetValue	Function
[31:16]	KEY	W	16'b0	当 KEY=0x5a5a 时，bit 0 才可以被写入。本位读始终为 0.
[15:1]	reserved		15'b0	保留
[0]	EPXBLOCK	RW	1'b0	EPWMXBAR 寄存器锁定位，锁定后以下寄存器写操作受限：EPWM_EPXBx_0, EPWM_EPXBx_1, EPWM_EPXBx_MUX, EPWM_EPXB_POL。 0：允许写寄存器 1：拒绝写寄存器 本寄存器被置 1 之后只能通过复位清 0

18.5.37 EPWM1_4 SYNCI 选择寄存器（EPWM_SYNCI_SEL1_4）

偏移地址：0xC0

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SYNCISEL4[4:0]					Res.	Res.	Res.	SYNCISEL3[4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SYNCISEL2[4:0]					Res.	Res.	Res.	SYNCISEL1[4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:29]	reserved		3’b0	保留
[28:24]	SYNCISEL4	RW	5‘b0	参考 SYNCISEL1
[23:21]	reserved		3’b0	保留
[20:16]	SYNCISEL3	RW	5‘b0	参考 SYNCISEL1
[15:13]	reserved		3’b0	保留
[12:8]	SYNCISEL2	RW	5‘b0	参考 SYNCISEL1

Puya Confidential

[7:5]	reserved		3’b0	保留
[4:0]	SYNCISEL1	RW	5‘b0	HRPMW1 的 SYNCI 输入源选择，参考表格 EPWMx 同步输入选择

18.5.38 EPWM5_6 SYNCI 选择寄存器（EPWM_SYNCI_SEL5_6）

偏移地址：0xC4

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SYNCISEL6[4:0]					Res.	Res.	Res.	SYNCISEL5[4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:13]	reserved		19’b0	保留
[12:8]	SYNCISEL6	RW	5‘b0	参考 SYNCISEL1
[7:5]	reserved		3’b0	保留
[4:0]	SYNCISEL5	RW	5‘b0	参考 SYNCISEL1

18.5.39 EPWM1_4 SYNCO 选择寄存器（EPWM_SYNCO_SEL1_4）

偏移地址：0xC8

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SYNCOSEL4[4:0]					Res.	Res.	Res.	SYNCOSEL3[4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SYNCOSEL2[4:0]					Res.	Res.	Res.	SYNCOSEL1[4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:29]	reserved		3’b0	保留
[28:24]	SYNCOSEL4	RW	5‘b0	参考 SYNCOSEL1
[23:21]	reserved		3’b0	保留
[20:16]	SYNCOSEL3	RW	5‘b0	参考 SYNCOSEL1
[15:13]	reserved		3’b0	保留
[12:8]	SYNCOSEL2	RW	5‘b0	参考 SYNCOSEL1
[7:5]	reserved		3’b0	保留
[4:0]	SYNCOSEL1	RW	5‘b0	HRPMW1_SYNCO 的输出源选择，参考表表 18-6 EPWMx 同步输出源选择

18.5.40 EPWM5_6 SYNCO 选择寄存器（EPWM_SYNCO_SEL5_6）

偏移地址：0xCC

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	SYNCOSEL6[4:0]					Res.	Res.	Res.	SYNCOSEL5[4:0]				
-	-	-	RW	RW	RW	RW	RW	-	-	-	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:13]	reserved		19'b0	保留
[12:8]	SYNCOSEL6	RW	5'b0	参考 SYNCOSEL1
[7:5]	reserved		3'b0	保留
[4:0]	SYNCOSEL5	RW	5'b0	参考 SYNCOSEL1

18.5.41 外部 ADC 控制寄存器（EPWM_ADCOUT）

偏移地址：0xD0

复位值： 0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	SOCB_POL	SOCA_POL	SOC6B_EN	SOC5B_EN	SOC4B_EN	SOC3B_EN	SOC2B_EN	SOC1B_EN	SOC6A_EN	SOC5A_EN	SOC4A_EN	SOC3A_EN	SOC2A_EN	SOC1A_EN
-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:14]	reserved		18'b0	保留
[13]	SOCB_POL	RW	1'b0	ADCSOCBO 输出极性 0：低电平有效 1：高电平有效
[12]	SOCA_POL	RW	1'b0	ADCSOCAO 输出极性 0：低电平有效 1：高电平有效
[11]	SOC6B_EN	RW	1'b0	参考 SOC1B_EN
[10]	SOC5B_EN	RW	1'b0	参考 SOC1B_EN
[9]	SOC4B_EN	RW	1'b0	参考 SOC1B_EN
[8]	SOC3B_EN	RW	1'b0	参考 SOC1B_EN
[7]	SOC2B_EN	RW	1'b0	参考 SOC1B_EN
[6]	SOC1B_EN	RW	1'b0	EPWM1 输出的 soc1b 至 ADCSOCBO 的使能位 0：代表没有被选择 1：代表被选择
[5]	SOC6A_EN	RW	1'b0	参考 SOC1A_EN
[4]	SOC5A_EN	RW	1'b0	参考 SOC1A_EN
[3]	SOC4A_EN	RW	1'b0	参考 SOC1A_EN
[2]	SOC3A_EN	RW	1'b0	参考 SOC1A_EN
[1]	SOC2A_EN	RW	1'b0	参考 SOC1A_EN

Puya Confidential

[0]	SOC1A_EN	RW	1'b0	EPWM1 输出的 soc1a 至 ADCSOCAO 的使能位（HRPMW1SOCA_EN） 0：代表没有被选择 1：代表被选择

18.5.42 EPWM_SCOUT 配置寄存器（EPWM_SCOUT_CFG）

偏移地址：0xD4

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCOUT_EXP_BYPASS	SCOUT_SEL[2:0]		
-	-	-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:4]	reserved		28'b0	保留
[3]	SCOUT_EXP_BYPASS	RW	1'b0	0：EPWM_SCOUT 输出经过至少 8 个周期的脉冲展宽 1：跳过 EPWM_SCOUT 脉冲展宽模块
[2:0]	SCOUT_SEL	RW	3'b0	EPWM_SCOUT 输出选择 参考表表 18-7 片外同步输出源选择

18.5.43 HRPMW SYNCSOC 锁定寄存器（EPWM_SYNCSOCLOCK）

偏移地址：0xD8

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADCSOCLOCK	SYNCILOCK
-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW	RW

Bit	Name	Access	ResetValue	Function
[31:2]	reserved		30'b0	保留
[1]	ADCSOCLOCK	RW	1'b0	对 EPWM_ADCOUT 寄存器的锁定标志。本寄存器被置 1 之后只能通过复位清 0 0：寄存器没有被锁定 1：寄存器被锁定
[0]	SYNCILOCK	RW	1'b0	对 EPWM_SYNCI_SEL1_4, EPWM_ SYNCI_SEL5_6 寄存器的锁定标志。本寄存器被置 1 之后只能通过复位清 0 0：寄存器没有被锁定 1：寄存器被锁定

Puya Confidential

18.5.44 EALLOW 寄存器（EPWM_EALLOW）

偏移地址：0xDC

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EALLOW_KEY[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EALLOW
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW

Bit	Name	Access	ResetValue	Function
[31:16]	KEY	W	16'b0	当 KEY=0x5a5a 时，bit 0 才可以被写入。本位读始终为 0.
[15:1]	reserved		15'b0	保留
[0]	EALLOW	RW	1'b0	寄存器锁定位，锁定后所有 EPWM 共用寄存器以及部分 EPWMx 寄存器写操作受限,详见寄存器保护机制描述 0：拒绝写寄存器 1：允许写寄存器

18.5.45 EPWM 时钟使能寄存器（EPWM_CLKEN）

偏移地址：0xE0

复位值：0x0000 003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPWM6_CLK_EN	EPWM5_CLK_EN	EPWM4_CLK_EN	EPWM3_CLK_EN	EPWM2_CLK_EN	EPWM1_CLK_EN
-	-	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Access	ResetValue	Function
[31:6]	Reserved		26'b0	保留
[5]	EPWM6_CLK_EN	RW	1'b1	EPMW6 的时钟使能信号 1：开启时钟 0：关闭时钟
[4]	EPWM5_CLK_EN	RW	1'b1	EPMW5 的时钟使能信号 1：开启时钟 0：关闭时钟
[3]	EPWM4_CLK_EN	RW	1'b1	EPMW4 的时钟使能信号 1：开启时钟 0：关闭时钟
[2]	EPWM3_CLK_EN	RW	1'b1	EPMW3 的时钟使能信号 1：开启时钟 0：关闭时钟

Puya Confidential

[1]	EPWM2_CLK_EN	RW	1'b1	EPMW2 的时钟使能信号 1：开启时钟 0：关闭时钟
[0]	EPWM1_CLK_EN	RW	1'b1	EPMW1 的时钟使能信号 1：开启时钟 0：关闭时钟

18.5.46 XBAR 状态寄存器 1（EPWM_XBARFLG1）

偏移地址：0xF0

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIM1_CH6_FLG	TIM1_CH5_FLG	TIM1_CH4_FLG	TIM1_CH3_FLG	TIM1_CH2_FLG	TIM1_CH1_FLG	ADC2_AWD3_FLG	ADC2_AWD2_FLG	ADC2_AWD1_FLG	ADC1_AWD3_FLG	ADC1_AWD2_FLG	ADC1_AWD1_FLG	INPUT14_FLG	INPUT13_FLG	INPUT12_FLG	INPUT11_FLG
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INPUT10_FLG	INPUT9_FLG	INPUT8_FLG	INPUT7_FLG	ADCSOCB_FLG	ADCSOCA_FLG	INPUT6_FLG	INPUT5_FLG	INPUT4_FLG	INPUT3_FLG	INPUT2_FLG	INPUT1_FLG	COMP4_OUT_FLG	COMP3_OUT_FLG	COMP2_OUT_FLG	COMP1_OUT_FLG
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Access	ResetValue	Function
[31]	TIM1_CH6_FLG	R	1'b0	TIM1_CH6 触发标志 1:已经触发 0:没有触发
[30]	TIM1_CH5_FLG	R	1'b0	TIM1_CH5 触发标志 1:已经触发 0:没有触发
[29]	TIM1_CH4_FLG	R	1'b0	TIM1_CH4 触发标志 1:已经触发 0:没有触发
[28]	TIM1_CH3_FLG	R	1'b0	TIM1_CH3 触发标志 1:已经触发 0:没有触发
[27]	TIM1_CH2_FLG	R	1'b0	TIM1_CH2 触发标志 1:已经触发 0:没有触发
[26]	TIM1_CH1_FLG	R	1'b0	TIM1_CH1 触发标志 1:已经触发 0:没有触发
[25]	ADC2_AWD3_FLG	R	1'b0	ADC2_AWD3 触发标志 1:已经触发 0:没有触发
[24]	ADC2_AWD2_FLG	R	1'b0	ADC2_AWD2 触发标志 1:已经触发 0:没有触发

Puya Confidential

[23]	ADC2_AWD1_FLG	R	1'b0	ADC2_AWD1 触发标志 1:已经触发 0:没有触发
[22]	ADC1_AWD3_FLG	R	1'b0	ADC1_AWD3 触发标志 1:已经触发 0:没有触发
[21]	ADC1_AWD2_FLG	R	1'b0	ADC1_AWD2 触发标志 1:已经触发 0:没有触发
[20]	ADC1_AWD1_FLG	R	1'b0	ADC1_AWD1 触发标志 1:已经触发 0:没有触发
[19]	INPUT14_FLG	R	1'b0	INPUT14 触发标志 1:已经触发 0:没有触发
[18]	INPUT13_FLG	R	1'b0	INPUT13 触发标志 1:已经触发 0:没有触发
[17]	INPUT12_FLG	R	1'b0	INPUT12 触发标志 1:已经触发 0:没有触发
[16]	INPUT11_FLG	R	1'b0	INPUT11 触发标志 1:已经触发 0:没有触发
[15]	INPUT10_FLG	R	1'b0	INPUT10 触发标志 1:已经触发 0:没有触发
[14]	INPUT9_FLG	R	1'b0	INPUT9 触发标志 1:已经触发 0:没有触发
[13]	INPUT8_FLG	R	1'b0	INPUT8 触发标志 1:已经触发 0:没有触发
[12]	INPUT7_FLG	R	1'b0	INPUT7T 触发标志 1:已经触发 0:没有触发
[11]	ADCSOCB_FLG	R	1'b0	ADCSOCB 触发标志 1:已经触发 0:没有触发
[10]	ADCSOCA_FLG	R	1'b0	ADCSOCA 触发标志 1:已经触发 0:没有触发
[9]	INPUT6_FLG	R	1'b0	INPUT6 触发标志 1:已经触发 0:没有触发

Puya Confidential

[8]	INPUT5_FLG	R	1'b0	INPUT5 触发标志 1:已经触发 0:没有触发
[7]	INPUT4_FLG	R	1'b0	INPUT4 触发标志 1:已经触发 0:没有触发
[6]	INPUT3_FLG	R	1'b0	INPUT3 触发标志 1:已经触发 0:没有触发
[5]	INPUT2_FLG	R	1'b0	INPUT2 触发标志 1:已经触发 0:没有触发
[4]	INPUT1_FLG	R	1'b0	INPUT1 触发标志 1:已经触发 0:没有触发
[3]	COMP4_OUT_FLG	R	1'b0	COMP4_OUT 触发标志 1:已经触发 0:没有触发
[2]	COMP3_OUT_FLG	R	1'b0	COMP3_OUT 触发标志 1:已经触发 0:没有触发
[1]	COMP2_OUT_FLG	R	1'b0	COMP2_OUT 触发标志 1:已经触发 0:没有触发
[0]	COMP1_OUT_FLG	R	1'b0	COMP1_OUT 触发标志 1:已经触发 0:没有触发

18.5.47 XBAR 状态寄存器 2（EPWM_XBARFLG2）

偏移地址：0xF4

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CPUHALT_FLG	EPWMSCOUT_FLG	TIM8_CH6_FLG	TIM8_CH5_FLG	TIM8_CH4_FLG	TIM8_CH3_FLG	TIM8_CH2_FLG	TIM8_CH1_FLG	TIM3_CH4_FLG	TIM3_CH3_FLG	TIM3_CH2_FLG	TIM3_CH1_FLG	TIM2_CH4_FLG	TIM2_CH3_FLG	TIM2_CH2_FLG	TIM2_CH1_FLG
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Access	ResetValue	Function
[31:16]	reserved		16'b0	保留
[15]	CPUHALT_FLG	R	1'b0	CPUHALT 触发标志 1:已经触发 0:没有触发

Puya Confidential

[14]	EPWM_SCOUT_FLG	R	1'b0	EPWM_SCOUT 触发标志 1:已经触发 0:没有触发
[13]	TIM8_CH6_FLG	R	1'b0	TIM8_CH6 触发标志 1:已经触发 0:没有触发
[12]	TIM8_CH5_FLG	R	1'b0	TIM8_CH5 触发标志 1:已经触发 0:没有触发
[11]	TIM8_CH4_FLG	R	1'b0	TIM8_CH4 触发标志 1:已经触发 0:没有触发
[10]	TIM8_CH3_FLG	R	1'b0	TIM8_CH3 触发标志 1:已经触发 0:没有触发
[9]	TIM8_CH2_FLG	R	1'b0	TIM8_CH2 触发标志 1:已经触发 0:没有触发
[8]	TIM8_CH1_FLG	R	1'b0	TIM8_CH1 触发标志 1:已经触发 0:没有触发
[7]	TIM3_CH4_FLG	R	1'b0	TIM3_CH4 触发标志 1:已经触发 0:没有触发
[6]	TIM3_CH3_FLG	R	1'b0	TIM3_CH3 触发标志 1:已经触发 0:没有触发
[5]	TIM3_CH2_FLG	R	1'b0	TIM3_CH2 触发标志 1:已经触发 0:没有触发
[4]	TIM3_CH1_FLG	R	1'b0	TIM3_CH1 触发标志 1:已经触发 0:没有触发
[3]	TIM2_CH4_FLG	R	1'b0	TIM2_CH4 触发标志 1:已经触发 0:没有触发
[2]	TIM2_CH3_FLG	R	1'b0	TIM2_CH3 触发标志 1:已经触发 0:没有触发
[1]	TIM2_CH2_FLG	R	1'b0	TIM2_CH2 触发标志 1:已经触发 0:没有触发
[0]	TIM2_CH1_FLG	R	1'b0	TIM2_CH1 触发标志 1:已经触发 0:没有触发

Puya Confidential

18.5.48 XBAR 状态清除寄存器 1（EPWM_XBARCLR1）

偏移地址：0xF8

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIM1_CH6_CLR	TIM1_CH5_CLR	TIM1_CH4_CLR	TIM1_CH3_CLR	TIM1_CH2_CLR	TIM1_CH1_CLR	ADC2_AWD3_CLR	ADC2_AWD2_CLR	ADC2_AWD1_CLR	ADC1_AWD3_CLR	ADC1_AWD2_CLR	ADC1_AWD1_CLR	INPUT14_CLR	INPUT13_CLR	INPUT12_CLR	INPUT11_CLR
RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INPUT10_CLR	INPUT9_CLR	INPUT8_CLR	INPUT7_CLR	ADCSOCB_CLR	ADCSOCA_CLR	INPUT6_CLR	INPUT5_CLR	INPUT4_CLR	INPUT3_CLR	INPUT2_CLR	INPUT1_CLR	COMP4_OUT_CLR	COMP3_OUT_CLR	COMP2_OUT_CLR	COMP1_OUT_CLR
RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1

Bit	Name	Access	ResetValue	Function
[31]	TIM1_CH6_CLR	RC_W1	1'b0	写 1 清除 TIM1_CH6_FLG，写 0 没有作用
[30]	TIM1_CH5_CLR	RC_W1	1'b0	写 1 清除 TIM1_CH5_FLG，写 0 没有作用
[29]	TIM1_CH4_CLR	RC_W1	1'b0	写 1 清除 TIM1_CH4_FLG，写 0 没有作用
[28]	TIM1_CH3_CLR	RC_W1	1'b0	写 1 清除 TIM1_CH3_FLG，写 0 没有作用
[27]	TIM1_CH2_CLR	RC_W1	1'b0	写 1 清除 TIM1_CH2_FLG，写 0 没有作用
[26]	TIM1_CH1_CLR	RC_W1	1'b0	写 1 清除 TIM1_CH1_FLG，写 0 没有作用
[25]	ADC2_AWD3_CLR	RC_W1	1'b0	写 1 清除 ADC2_AWD3_FLG，写 0 没有作用
[24]	ADC2_AWD2_CLR	RC_W1	1'b0	写 1 清除 ADC2_AWD2_FLG，写 0 没有作用
[23]	ADC2_AWD1_CLR	RC_W1	1'b0	写 1 清除 ADC2_AWD1_FLG，写 0 没有作用
[22]	ADC1_AWD3_CLR	RC_W1	1'b0	写 1 清除 ADC1_AWD3_FLG，写 0 没有作用
[21]	ADC1_AWD2_CLR	RC_W1	1'b0	写 1 清除 ADC1_AWD2_FLG，写 0 没有作用
[20]	ADC1_AWD1_CLR	RC_W1	1'b0	写 1 清除 ADC1_AWD1_FLG，写 0 没有作用
[19]	INPUT14_CLR	RC_W1	1'b0	写 1 清除 INPUT14_FLG，写 0 没有作用
[18]	INPUT13_CLR	RC_W1	1'b0	写 1 清除 INPUT13_FLG，写 0 没有作用
[17]	INPUT12_CLR	RC_W1	1'b0	写 1 清除 INPUT12_FLG，写 0 没有作用
[16]	INPUT11_CLR	RC_W1	1'b0	写 1 清除 INPUT11_FLG，写 0 没有作用
[15]	INPUT10_CLR	RC_W1	1'b0	写 1 清除 INPUT10_FLG，写 0 没有作用
[14]	INPUT9_CLR	RC_W1	1'b0	写 1 清除 INPUT9_FLG，写 0 没有作用
[13]	INPUT8_CLR	RC_W1	1'b0	写 1 清除 INPUT8_FLG，写 0 没有作用
[12]	INPUT7_CLR	RC_W1	1'b0	写 1 清除 INPUT7_FLG，写 0 没有作用
[11]	ADCSOCB_CLR	RC_W1	1'b0	写 1 清除 ADCSOCB_FLG，写 0 没有作用
[10]	ADCSOCA_CLR	RC_W1	1'b0	写 1 清除 ADCSOCA_FLG，写 0 没有作用
[9]	INPUT6_CLR	RC_W1	1'b0	写 1 清除 INPUT6_FLG，写 0 没有作用
[8]	INPUT5_CLR	RC_W1	1'b0	写 1 清除 INPUT5_FLG，写 0 没有作用
[7]	INPUT4_CLR	RC_W1	1'b0	写 1 清除 INPUT4_FLG，写 0 没有作用
[6]	INPUT3_CLR	RC_W1	1'b0	写 1 清除 INPUT3_FLG，写 0 没有作用
[5]	INPUT2_CLR	RC_W1	1'b0	写 1 清除 INPUT2_FLG，写 0 没有作用
[4]	INPUT1_CLR	RC_W1	1'b0	写 1 清除 INPUT1_FLG，写 0 没有作用
[3]	COMP4_OUT_CLR	RC_W1	1'b0	写 1 清除 COMP4_OUT_FLG，写 0 没有作用
[2]	COMP3_OUT_CLR	RC_W1	1'b0	写 1 清除 COMP3_OUT_FLG，写 0 没有作用

Puya Confidential

[1]	COMP2_OUT_CLR	RC_W1	1'b0	写 1 清除 COMP2_OUT_FLG，写 0 没有作用
[0]	COMP1_OUT_CLR	RC_W1	1'b0	写 1 清除 COMP1_OUT_FLG，写 0 没有作用

18.5.49 XBAR 状态清除寄存器 2（EPWM_XBARCLR2）

偏移地址：0xFC

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CPUHALT_CLR	EPWM_SCOUT_CLR	TIM8_CH6_CLR	TIM8_CH5_CLR	TIM8_CH4_CLR	TIM8_CH3_CLR	TIM8_CH2_CLR	TIM8_CH1_CLR	TIM3_CH4_CLR	TIM3_CH3_CLR	TIM3_CH2_CLR	TIM3_CH1_CLR	TIM2_CH4_CLR	TIM2_CH3_CLR	TIM2_CH2_CLR	TIM2_CH1_CLR
RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1	RC_W1

Bit	Name	Access	ResetValue	Function
[31:16]	reserved		16'b0	保留
[15]	CPUHALT_CLR	RC_W1	1'b0	写 1 清除 CPUHALT_FLG，写 0 没有作用
[14]	EPWM_SCOUT_CLR	RC_W1	1'b0	写 1 清除 EPWM_SCOUT_FLG，写 0 没有作用
[13]	TIM8_CH6_CLR	RC_W1	1'b0	写 1 清除 TIM8_CH6_FLG，写 0 没有作用
[12]	TIM8_CH5_CLR	RC_W1	1'b0	写 1 清除 TIM8_CH5_FLG，写 0 没有作用
[11]	TIM8_CH4_CLR	RC_W1	1'b0	写 1 清除 TIM8_CH4_FLG，写 0 没有作用
[10]	TIM8_CH3_CLR	RC_W1	1'b0	写 1 清除 TIM8_CH3_FLG，写 0 没有作用
[9]	TIM8_CH2_CLR	RC_W1	1'b0	写 1 清除 TIM8_CH2_FLG，写 0 没有作用
[8]	TIM8_CH1_CLR	RC_W1	1'b0	写 1 清除 TIM8_CH1_FLG，写 0 没有作用
[7]	TIM3_CH4_CLR	RC_W1	1'b0	写 1 清除 TIM3_CH4_FLG，写 0 没有作用
[6]	TIM3_CH3_CLR	RC_W1	1'b0	写 1 清除 TIM3_CH3_FLG，写 0 没有作用
[5]	TIM3_CH2_CLR	RC_W1	1'b0	写 1 清除 TIM3_CH2_FLG，写 0 没有作用
[4]	TIM3_CH1_CLR	RC_W1	1'b0	写 1 清除 TIM3_CH1_FLG，写 0 没有作用
[3]	TIM2_CH4_CLR	RC_W1	1'b0	写 1 清除 TIM2_CH4_FLG，写 0 没有作用
[2]	TIM2_CH3_CLR	RC_W1	1'b0	写 1 清除 TIM2_CH3_FLG，写 0 没有作用
[1]	TIM2_CH2_CLR	RC_W1	1'b0	写 1 清除 TIM2_CH2_FLG，写 0 没有作用
[0]	TIM2_CH1_CLR	RC_W1	1'b0	写 1 清除 TIM2_CH1_FLG，写 0 没有作用

Puya Confidential

19. 高级定时器（TIM1/TIM8）

19.1 TIM1/TIM8 简介

高级控制定时器（TIM1/TIM8）由一个 16 位的自动装载计数器组成，计数器由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度（输入捕获），或者产生输出波形（输出比较、PWM、嵌入死区时间的互补 PWM）。使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。高级控制定时器（TIM1）和通用定时器（TIMy）是完全独立的，它们不共享任何资源。它们可以同步操作。

19.2 TIM1/TIM8 主要特征

TIM1/TIM8 定时器的功能包括：

- 16 位递增、递减、递增/递减的自动装载计数器
- 16 位可编程(可以实时修改)的预分频器，计数器时钟频率的分频系数为 1～65536 之间的任意数值
- 多达 6 个独立通道：
  - 输入捕获（5/6 通道不支持）
  - 输出比较
  - PWM 生成(边缘或中间对齐模式)
  - PWM 移相（CH1/CH2/CH3 支持此功能）
  - 单脉冲模式输出
- 带有可编程死区时间的互补输出
- 通过外部信号来控制定时器与定时器之间互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 2 路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
- 如下事件发生时产生中断/DMA：
  - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
  - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
  - 输入捕获
  - 输出比较
- 支持针对定位的增量（正交）编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理
- 支持计数器多点触发 ADC

Puya Confidential

19.3 TIM1/TIM8 功能描述

19.3.1 TIM1/TIM8 模块框图

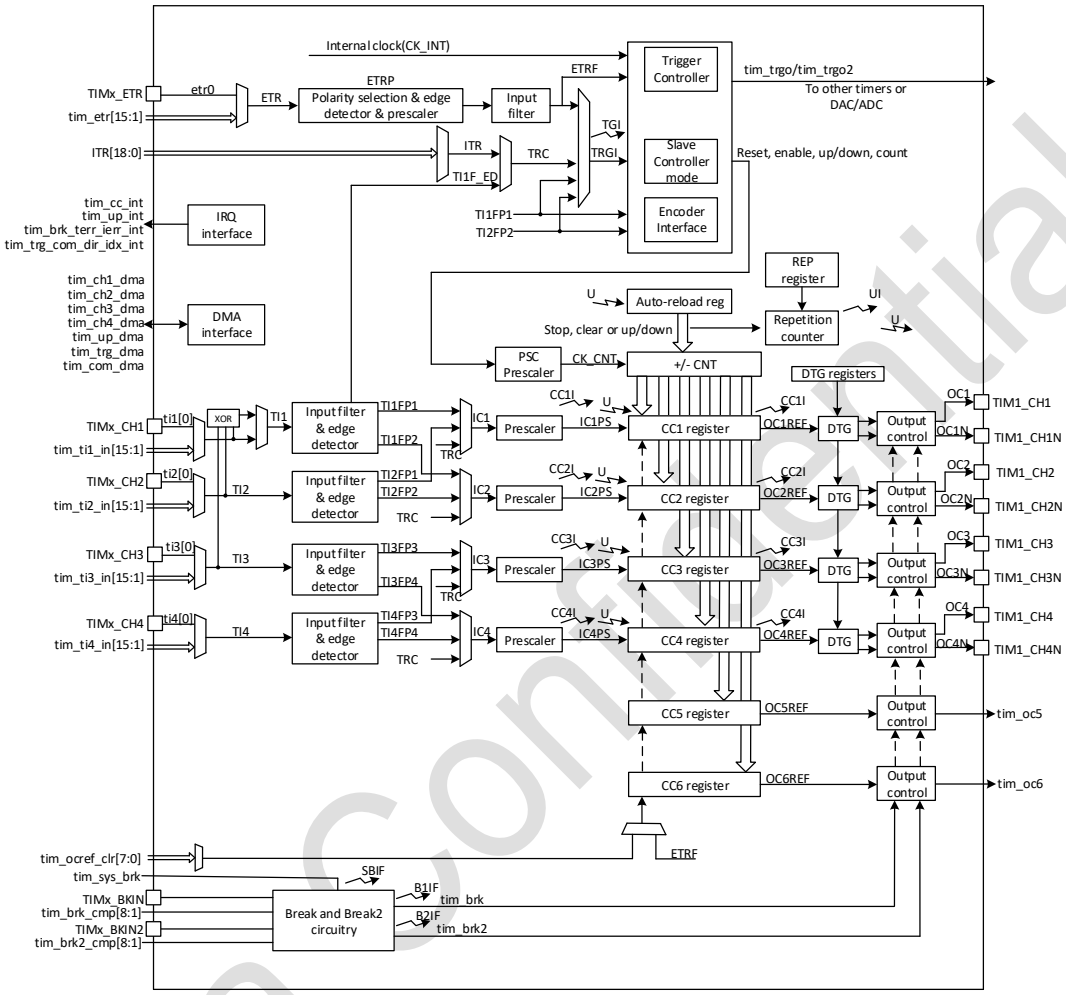


图 19-1 高级定时器框图

19.3.2 时基单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以递增计数、递减计数或者交替进行递增和递减计数。计数器的时钟可以被预分频器分频。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

时基单元包含：

- 计数器寄存器（TIMx_CNT）
- 预分频器寄存器（TIMx_PSC）
- 自动装载寄存器（TIMx_ARR）
- 重复计数寄存器（TIMx_RCR）

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问它的预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位（ARPE）的设置，预装载寄存器的内容被立即或在每次的更新事件（UEV）时传送到影子寄存器。当计数器达到溢出条件（上溢或者下溢）并当 TIMx_CR1 寄存器中的

Puya Confidential

UDIS 位等于 0 时，会产生更新事件。更新事件也可以由软件及其他条件产生。后续会详细描述每一种配置下更新事件的产生。

计数器由预分频器分频后的时钟输出 CK_CNT 驱动，仅当设置了 TIMx_CR1 寄存器中的计数器使能位（CEN），CK_CNT 才对计数器有效。(更多有关使能计数器的细节，请参见从模式控制器的描述)。

注意，在设置了 TIMx_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个（在 TIMx_PSC 寄存器中的) 16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频参数将在下一次更新事件到来时被采用。

下面几张图给出了在预分频器运行时，更改计数器参数的例子。

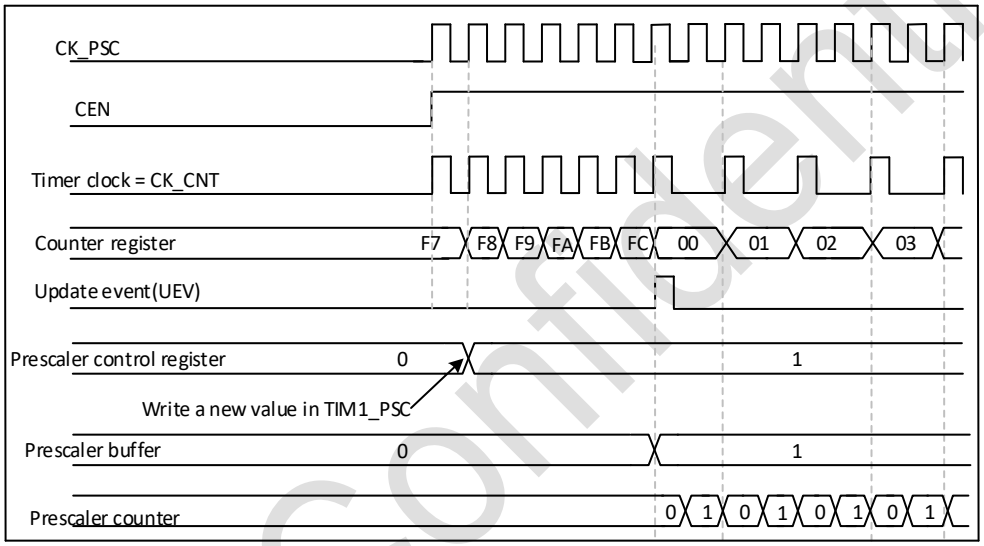


图 19-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

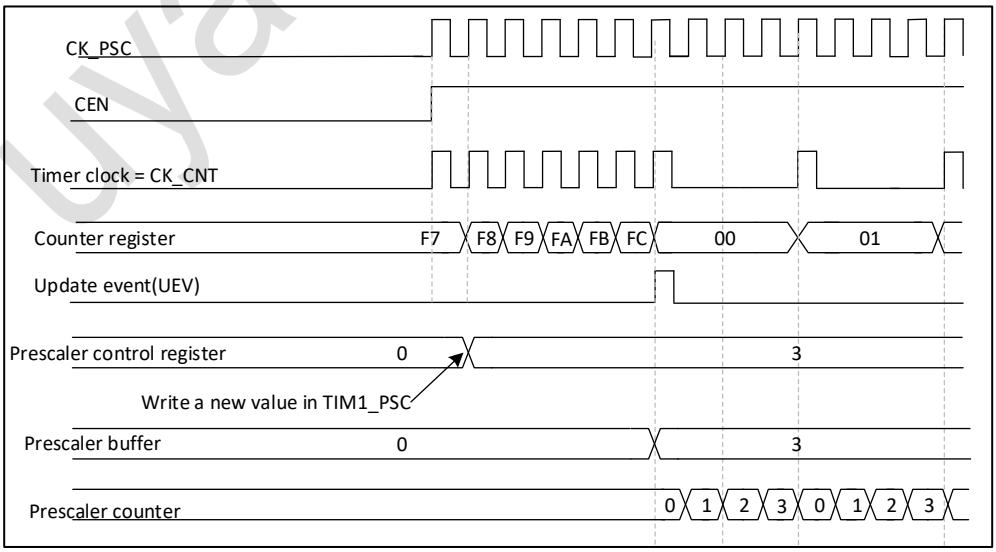


图 19-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

Puya Confidential

19.3.3 计数器模式

递增计数模式

在递增计数模式中，计数器从 0 计数到自动加载值（TIMx_ARR 的内容），然后重新从 0 开始计数并且产生一个计数上溢事件。

如果使用了重复计数器，那么更新事件(UEV)需要在上溢次数达到所配置的重复计数寄存器的值加一（即 TIMx_RCR+1）时才会产生；如果没有使用重复计数器（即 TIMx_RCR=0），那么每次计数上溢都会产生更新事件。

而在 TIMx_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。

通过设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清‘0’之前，将不会产生更新事件。即使这样，在应该产生更新事件时，计数器仍会被清‘0’，同时预分频器内部的计数器也被清‘0’（但预分频器的数值不变）。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），通过设置 UG 位可以产生一个更新事件 UEV，但不会置起 UIF 标志位（即不会产生中断或 DMA 请求）。这是为了避免在捕获事件时清除计数器，同时产生更新和捕获中断。

当发生一个更新事件时，所有以下的寄存器都被更新，硬件同时（依据 URS 位）设置更新标志位（TIMx_SR 寄存器中的 UIF 位）：

- 重复计数器被重新加载为 TIMx_RCR 寄存器的内容。
- 自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。
- 预分频器的缓冲区被置入预装载寄存器的值（TIMx_PSC 寄存器的内容）。

下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下的动作。

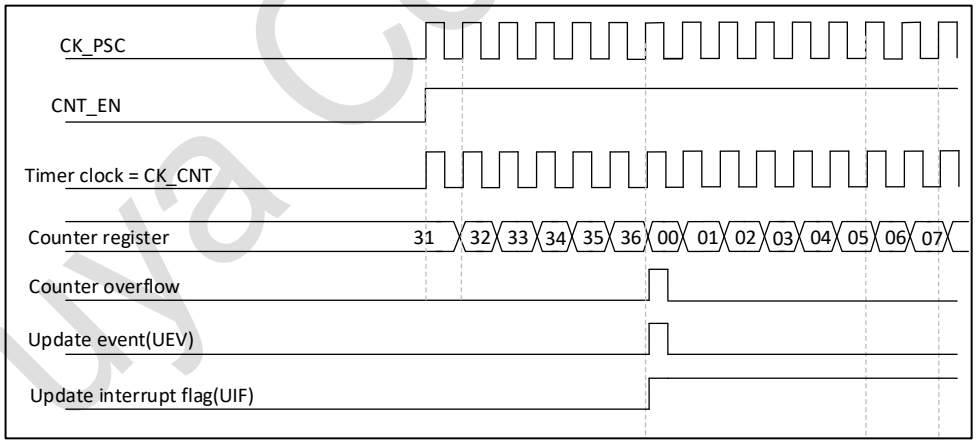


图 19-4 计数器时序图，内部时钟分频因子为 1

Puya Confidential

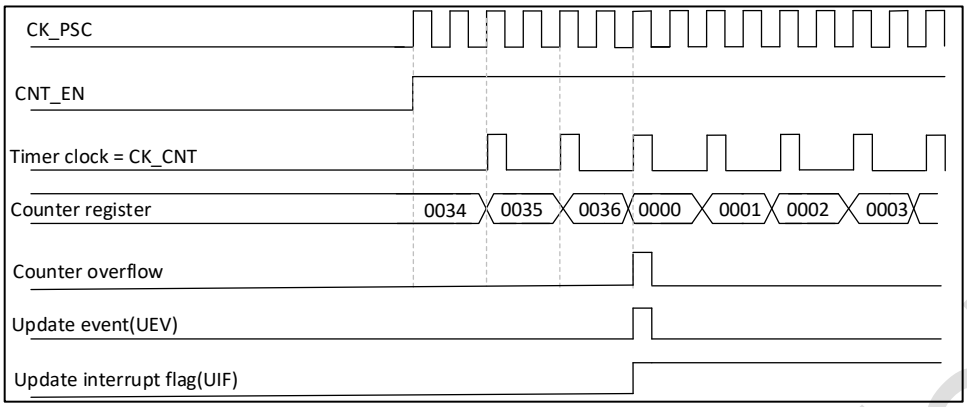


图 19-5 计数器时序图，内部时钟分频因子为 2

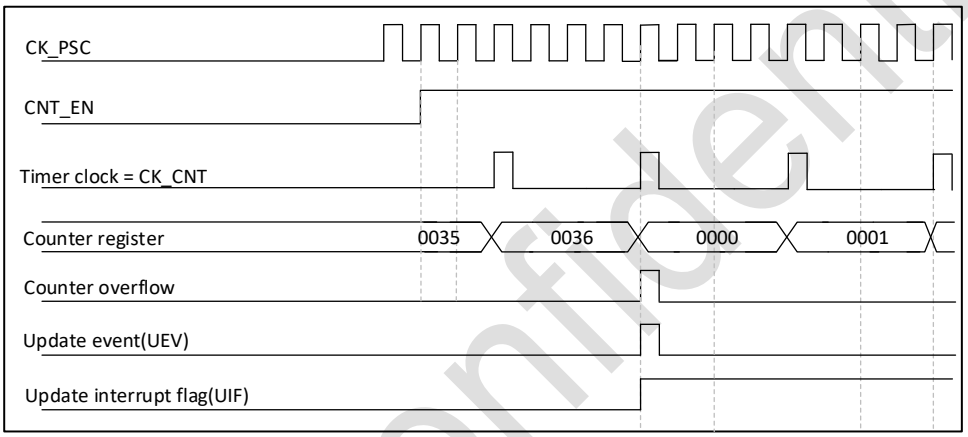


图 19-6 计数器时序图，内部时钟分频因子为 4

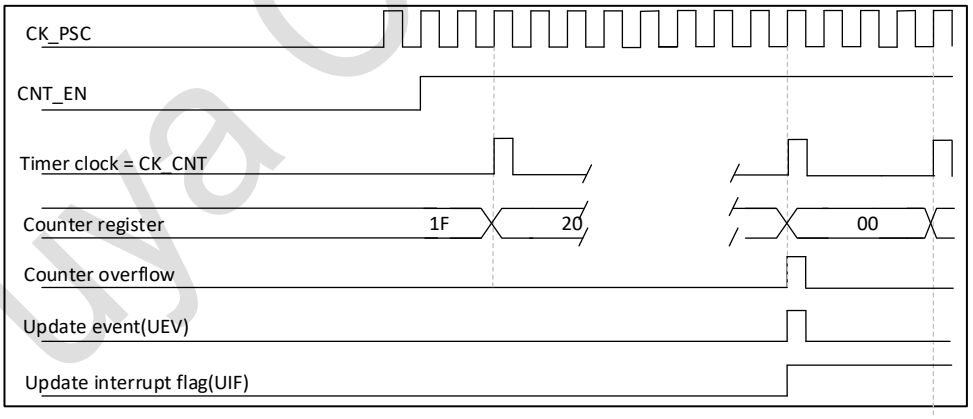


图 19-7 计数器时序图，内部时钟分频因子为 N

Puya Confidential

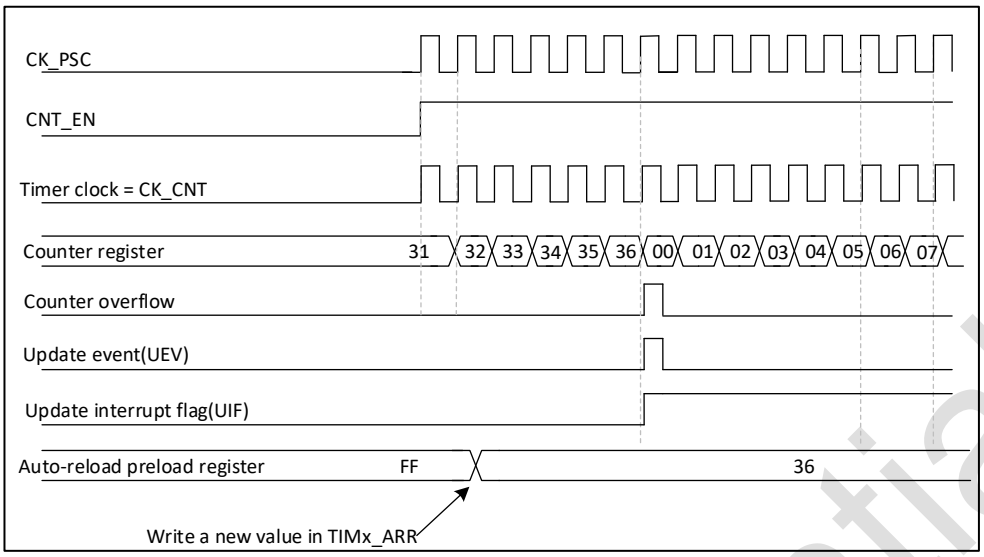


图 19-8 计数器时序图，当 ARPE=0 时的更新事件（没有预装 TIMx_ARR）

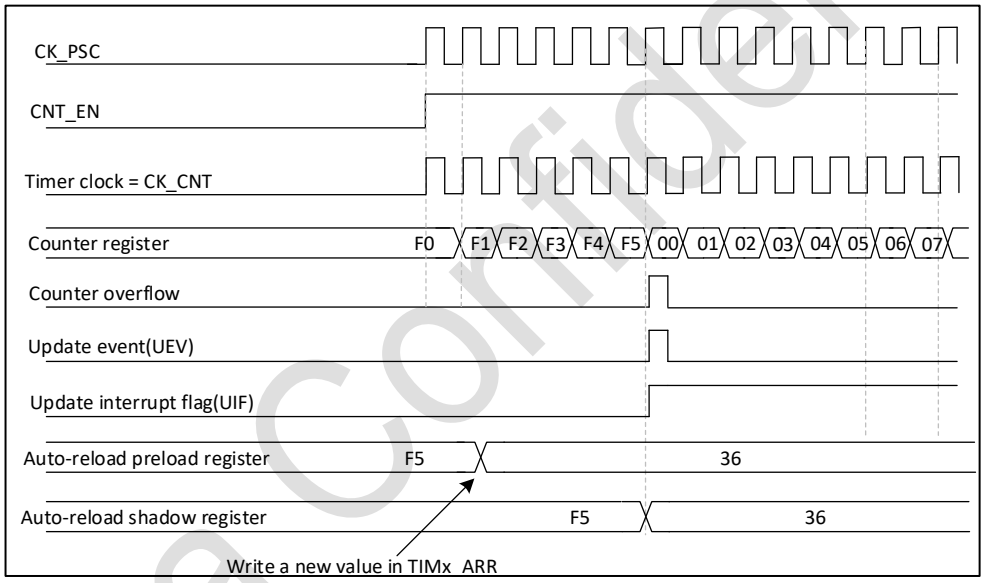


图 19-9 计数器时序图，当 ARPE=1 时的更新事件（预装了 TIMx_ARR）

递减计数模式

在递减计数模式中，计数器从自动加载值（TIMx_ARR 的内容）开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数下溢事件。

如果使用了重复计数器，那么更新事件（UEV）需要在下溢次数达到所配置的重复计数寄存器的值加一（即 TIMx_RCR+1）时才会产生；如果没有使用重复计数器（即 TIMx_RCR=0），那么每次计数下溢都会产生更新事件。

而在 TIMx_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。

通过设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清‘0’之前不会产生更新事件。即使这样，在应该产生更新事件时，计数器仍会从当前自动加载值重新开始计数，同时预分频器内部的计数器被清‘0’（但预分频系数不变）。

Puya Confidential

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），通过设置 UG 位可以产生一个更新事件 UEV，但不置起 UIF 标志位（即不会产生中断或 DMA 请求），这是为了避免在发生捕获事件时清除计数器，同时产生更新和捕获中断。

当发生一个更新事件时，所有以下的寄存器都被更新，硬件同时（依据 URS 位）设置更新标志位（TIMx_SR 寄存器中的 UIF 位）：

- 重复计数器被重新加载为 TIMx_RCR 寄存器中的内容。
- 预分频器的缓冲区被置入预装载寄存器的值（TIMx_PSC 寄存器的内容）。
- 当前的自动加载寄存器被更新为预装载值（TIMx_ARR 寄存器中的内容）。

注：自动加载值在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIMx_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

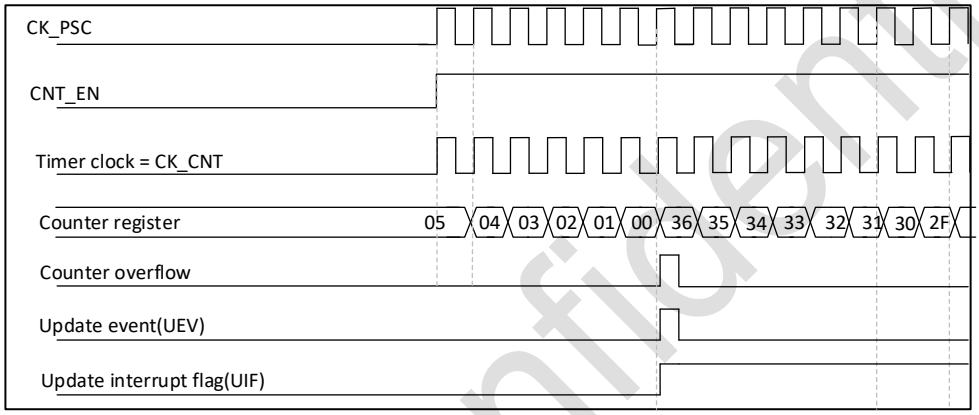


图 19-10 计数器时序图，内部时钟分频因子为 1

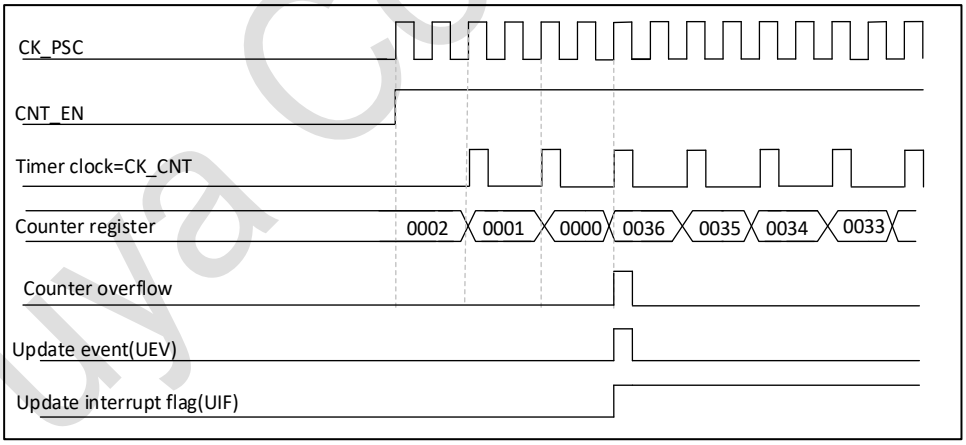


图 19-11 计数器时序图，内部时钟分频因子为 2

Puya Confidential

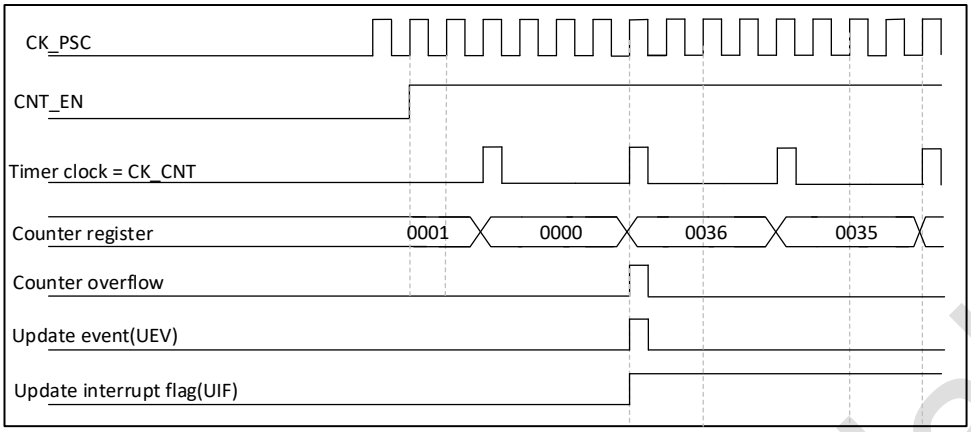


图 19-12 计数器时序图，内部时钟分频因子为 4

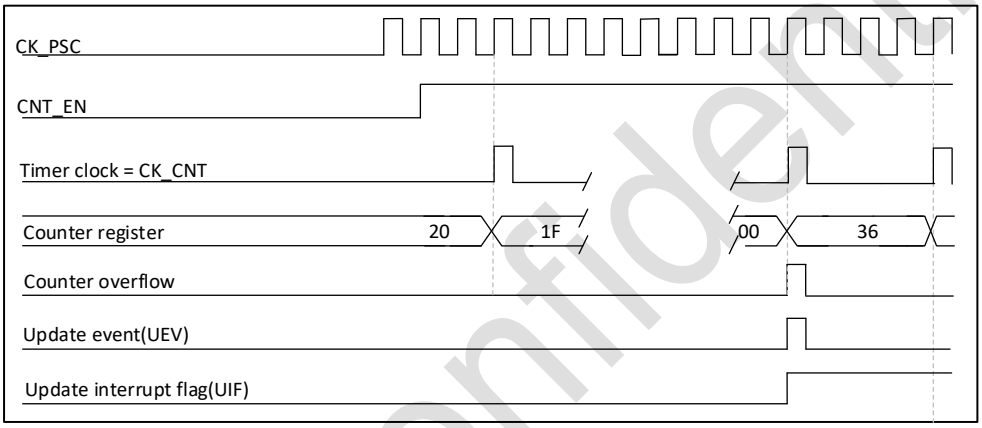


图 19-13 计数器时序图，内部时钟分频因子为 N

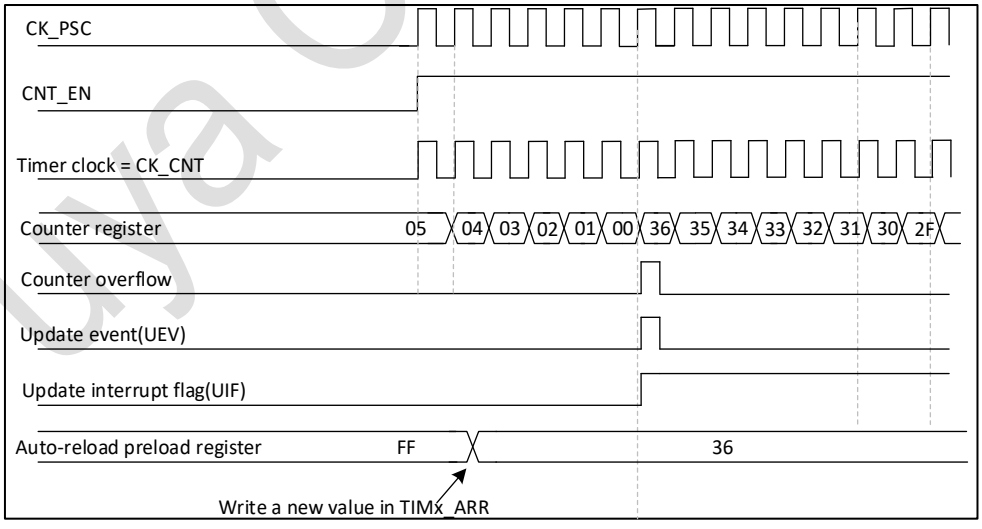


图 19-14 计数器时序图，当没有使用重复计数器时的更新事件

Puya Confidential

中央对齐模式（递增/递减计数）

在中央对齐模式，计数器从 0 开始计数到自动加载的值（TIMx_ARR 寄存器）减 1，产生一个计数器上溢事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

通过配置 TIMx_CR1 寄存器中的 CMS 位不为‘00’可以得到中央对齐模式。在通道配置为输出模式下的输出比较标志位在以下几种计数过程中会被置起：向下计数时（中央对齐模式 1，CMS=‘01’）、向上计数时（中央对齐模式 2，CMS=‘10’）、向上和向下计数时（中央对齐模式 3，CMS=‘11’）。

在此模式下，不能写入 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过（软件或者使用从模式控制器）设置 TIMx_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器内部计数器也重新从 0 开始计数。

设置 TIMx_CR1 寄存器中的 UDIS 位可以禁止更新事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。虽然 UDIS 位被清为 0 之前不会产生更新事件，但是计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），通过设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志（因此不产生中断和 DMA 请求），这是为了避免在发生捕获事件时清除计数器，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且（根据 URS 位的设置）更新标志位（TIMx_SR 寄存器中的 UIF 位）也被设置：

- 重复计数器被重置为 TIMx_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载（TIMx_PSC 寄存器）的值。
- 当前的自动加载寄存器被更新为预装载值（TIMx_ARR 寄存器中的内容）。注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值（计数器被装载为新的值）。

以下是一些计数器在不同时钟频率下的操作的例子：

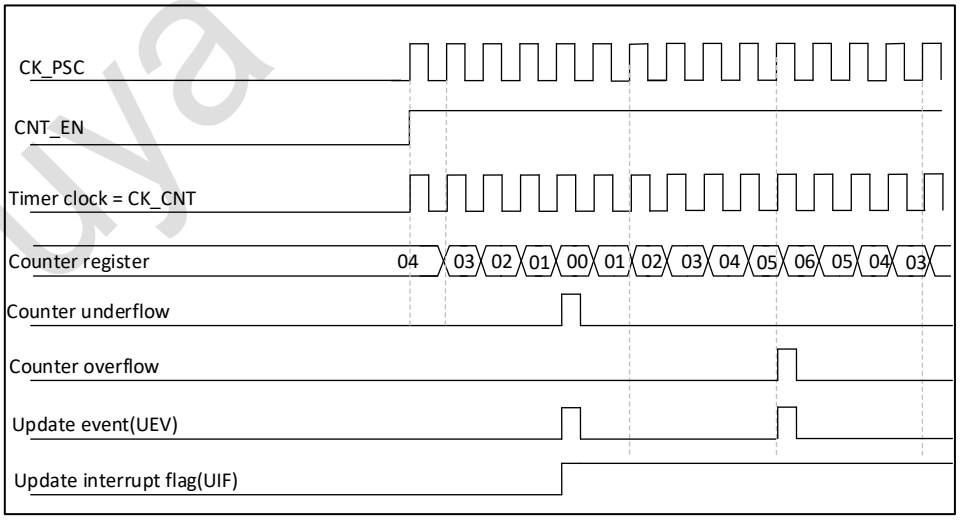


图 19-15 计数器时序图，内部时钟分频因子为 1，TIMx_ARR=0x6

Puya Confidential

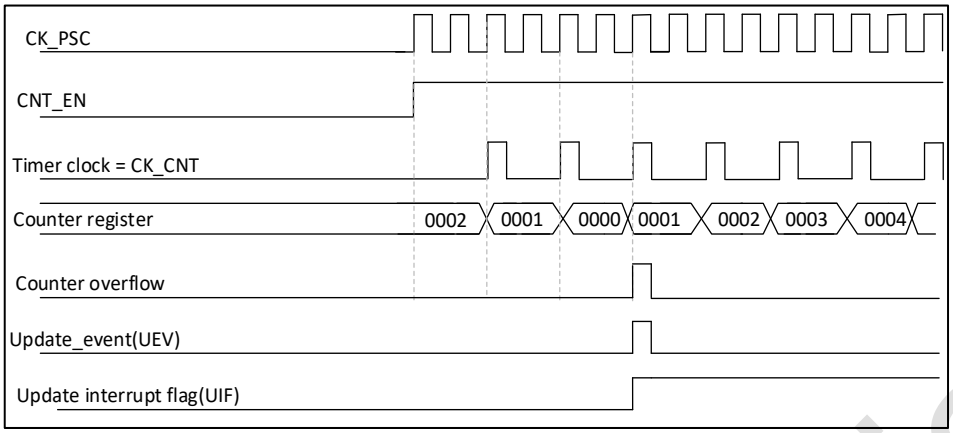


图 19-16 计数器时序图，内部时钟分频因子为 2

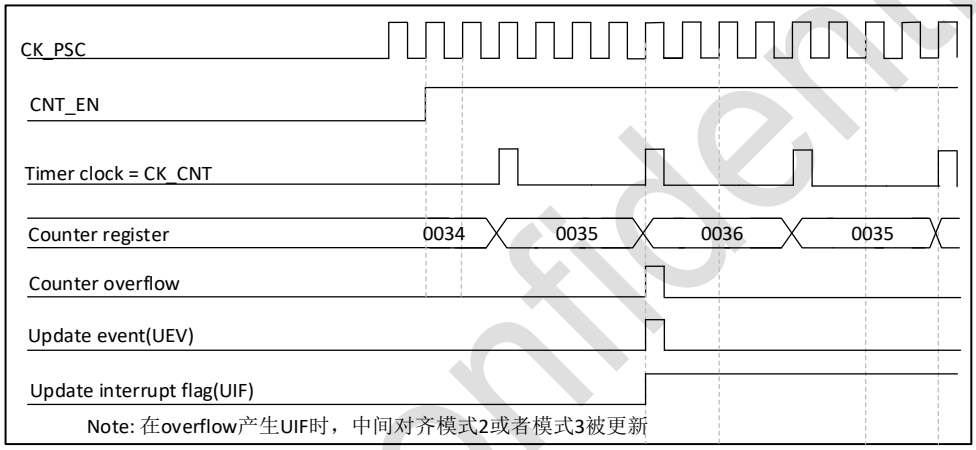


图 19-17 计数器时序图，内部时钟分频因子为 4，TIMx_ARR=0x36

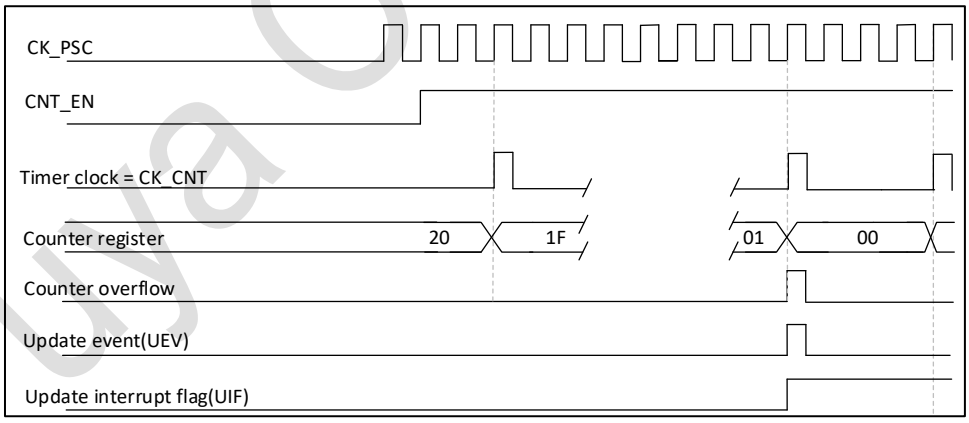


图 19-18 计数器时序图，内部时钟分频因子为 N

Puya Confidential

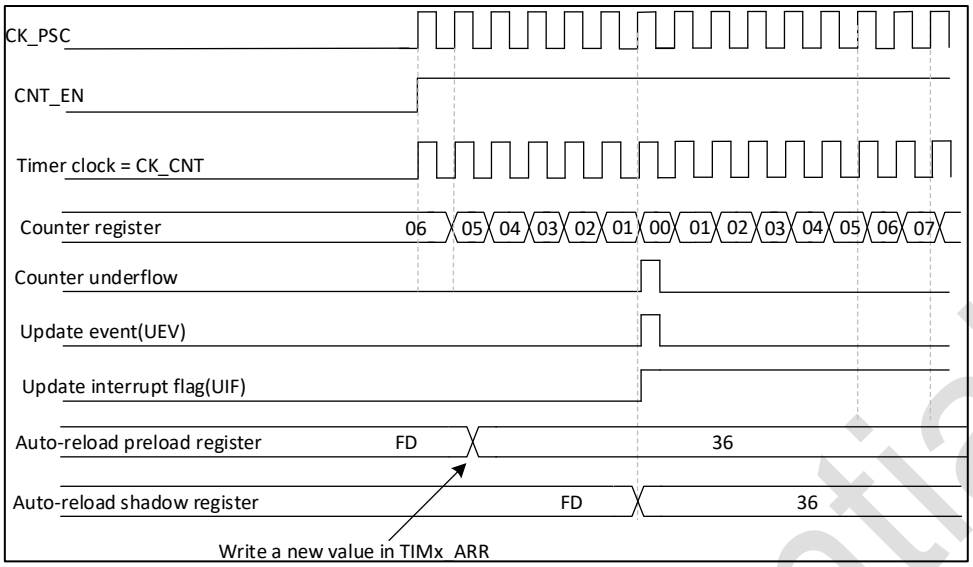


图 19-19 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

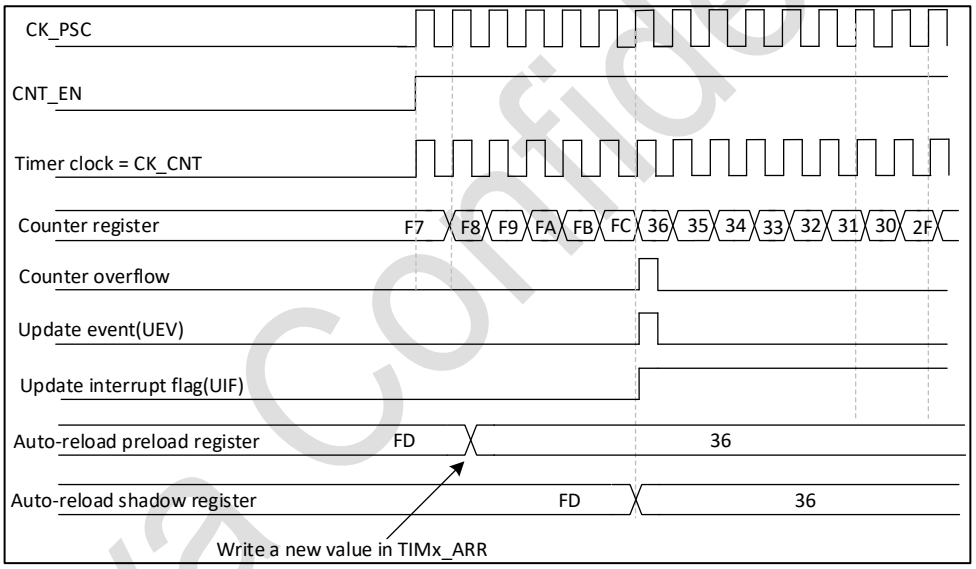


图 19-20 计数器时序图，ARPE=1 时的更新事件（计数器溢出）

19.3.4 重复计数器

“时基单元”解释了计数器上溢/下溢时更新事件（UEV）是如何产生的，事实上它只能在重复计数器计数达到 0 的时候产生。这个特性对产生 PWM 信号非常有用。

这意味着在每 N+1 个计数上溢或下溢时，数据才会从预装载寄存器传输到影子寄存器（TIMx_ARR 自动重载寄存器，TIMx_PSC 预装载寄存器，还有在比较模式下的捕获/比较寄存器 TIMx_CCRx），N 是 TIMx_RCR 重复计数寄存器中的值。

重复计数器在下述任一条件成立时递减：

- 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，
- 中央对齐模式下每次上溢和每次下溢时。

Puya Confidential

中央对齐模式限制了 PWM 的最大循环周期为 128 (TIMx_RCR 为 8 位计数器) , 但它能够在每个 PWM 周期 2 次更新占空比。

重复计数器是自动重载的，重复速率由 TIMx_RCR 寄存器的值定义。当更新事件由软件产生(通过设置 TIMx_EGR 中的 UG 位)或者通过硬件的从模式控制器产生，无论重复计数器的值是多少，立即发生更新事件，并且 TIMx_RCR 寄存器中的内容被重载入到重复计数器。

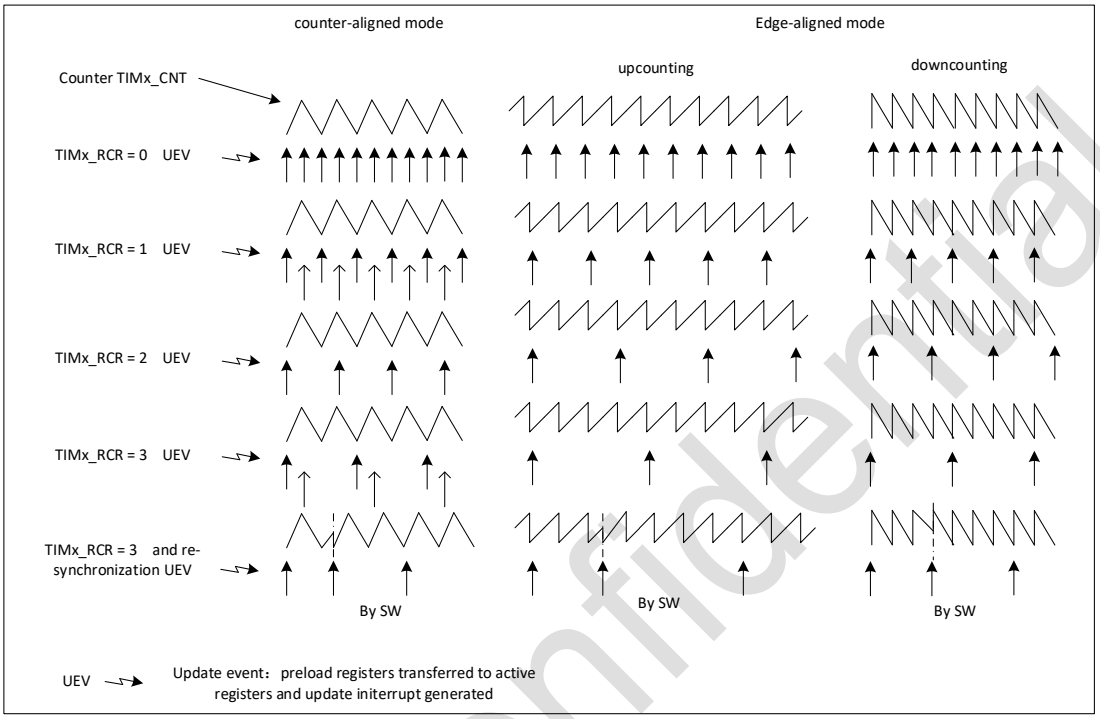


图 19-21 不同模式下更新速率的例子，及 TIMx_RCR 的寄存器设置

19.3.5 外部触发输入

定时器外部触发输入 tim_etr，可用于:

- 外部时钟
- 从模式的触发
- 在周期电流调节中作为 PWM 复位输入

19.3.6 时钟选择

计数器时钟可由下列时钟源提供:

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚（TI1 和 TI2）
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入（ITRx)：使用一个定时器作为另一个定时器的预分频器。如可以配置一个定时器而作为另一个定时器的预分频器。
- 编码器模式

内部时钟源（CK_INT）

如果禁止了从模式控制器（SMS=0000），则 CEN、DIR（TIMx_CR1 寄存器）和 UG 位（TIMx_EGR 寄存器）是事实上的控制位，并且只能被软件修改（UG 位仍被自动清除）。只要 CEN 位被写成‘1’，预分频器的时钟就由内部时钟 CK_INT 提供。

Puya Confidential

下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。

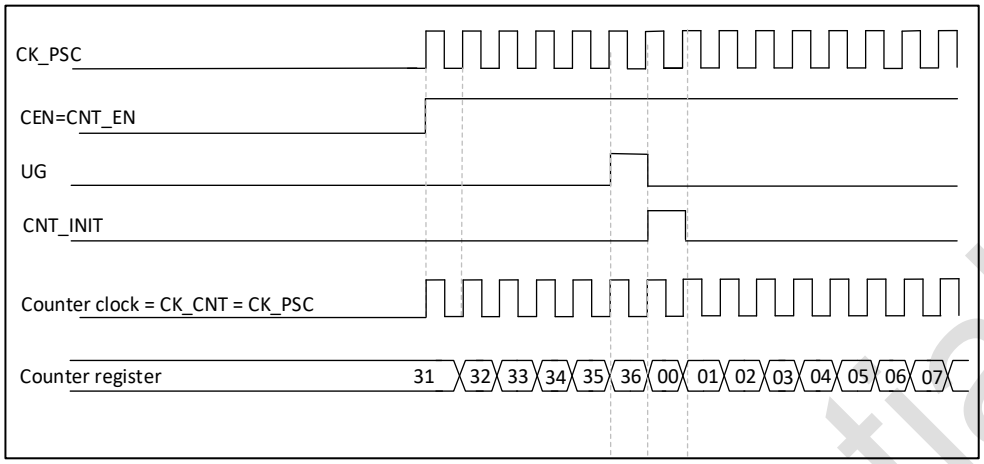


图 19-22 一般模式下的控制电路，内部时钟分频因子为 1

外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

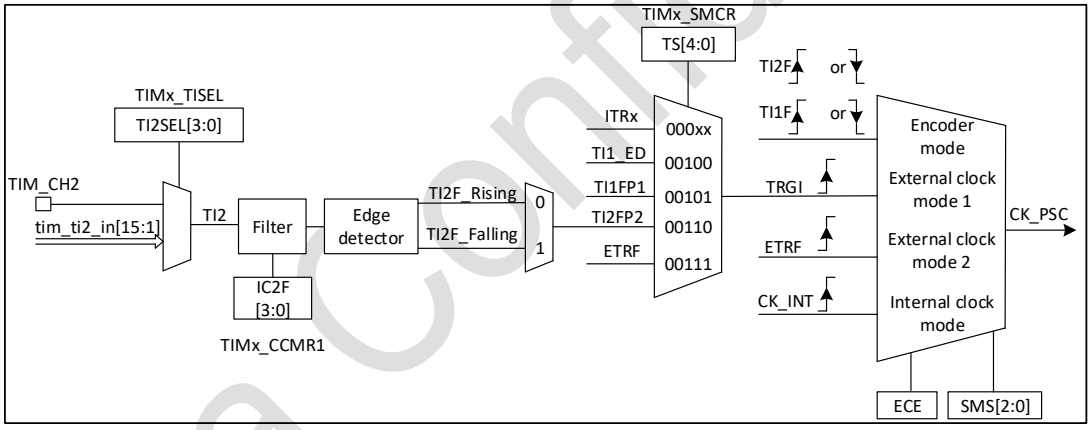


图 19-23 TI2 外部时钟连接例子

例如，要配置计数器在 TI2 输入端的上升沿向上计数，使用下列步骤：

1.配置 TIMx_CCMR1 寄存器 CC2S=01，使得通道 2 检测 TI2 输入端的上升沿；

2.配置 TIMx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽 (如果不需要滤波器，保持 IC2F=0000)；

3.配置 TIMx_CCER 寄存器的 CC2P=0，选定上升沿极性；

4.配置 TIMx_SMCR 寄存器的 SMS=111，选择定时器为外部时钟模式 1；

5.配置 TIMx_SMCR 寄存器中的 TS=00110，选定 TI2 作为触发输入源；

6.设置 TIMx_CR1 寄存器的 CEN=1，启动计数器。

注：捕获预分频器不用作触发，所以不需要对它进行配置

当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的同步电路。

Puya Confidential

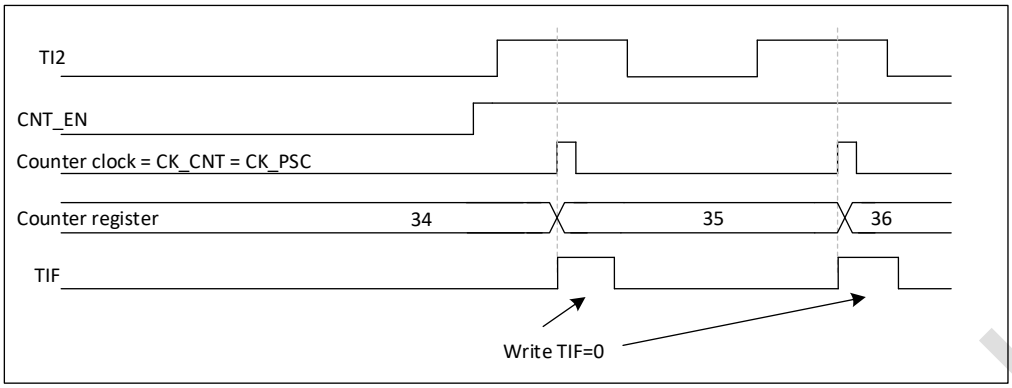


图 19-24 外部时钟模式 1 下的控制电路

外部时钟源模式 2

选定此模式的方法为：令 TIMx_SMCR 寄存器中的 ECE=1，计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图：

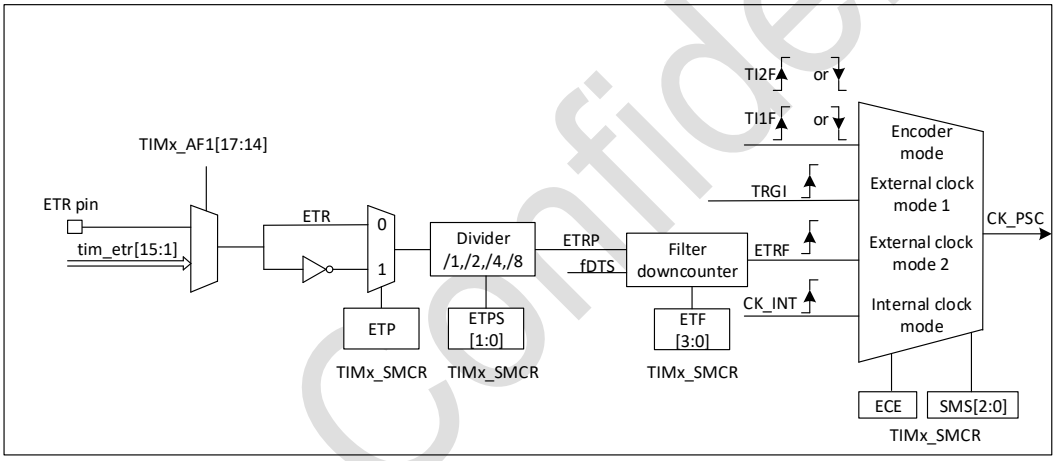


图 19-25 外部触发输入框图

例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：

1.本例中不需要滤波器，置 TIMx_SMCR 寄存器中的 ETF[3:0]=0000；

2.设置预分频器，置 TIMx_SMCR 寄存器中的 ETPS[1:0]=01；

3.选择 ETR 输入端的上升沿，置 TIMx_SMCR 寄存器中的 ETP=0；

4.开启外部时钟模式 2，写 TIMx_SMCR 寄存器中的 ECE=1；

5.启动计数器，写 TIMx_CR1 寄存器中的 CEN=1；

计数器在每 2 个 ETR 上升沿计数一次。

在 ETR 的上升沿和计数器实际时钟之间的延时取决于 ETRP 信号的同步电路。

Puya Confidential

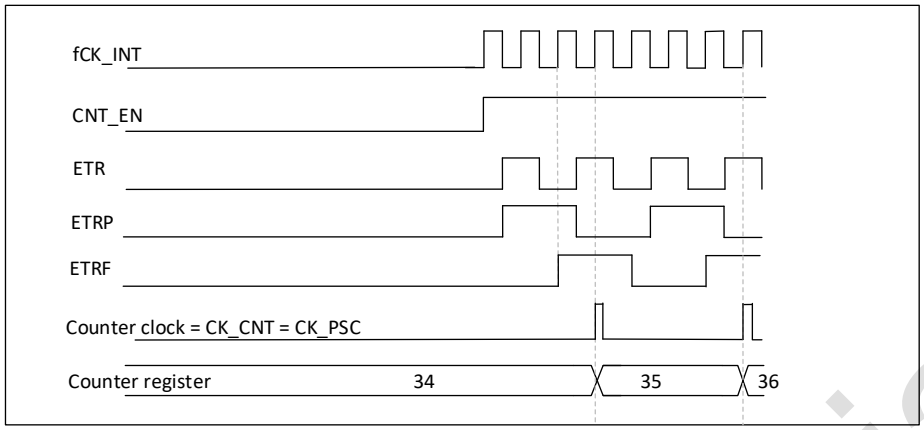


图 19-26 外部时钟模式 2 下的控制电路

19.3.7 捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器（包含影子寄存器），包括捕获的输入部分（数字滤波、多路复用和预分频器，通道 5 和 6 除外），和输出部分（比较器和输出控制）。

下面几张图是捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘监测器产生一个信号（TIxFPx），它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器（ICxPS）。

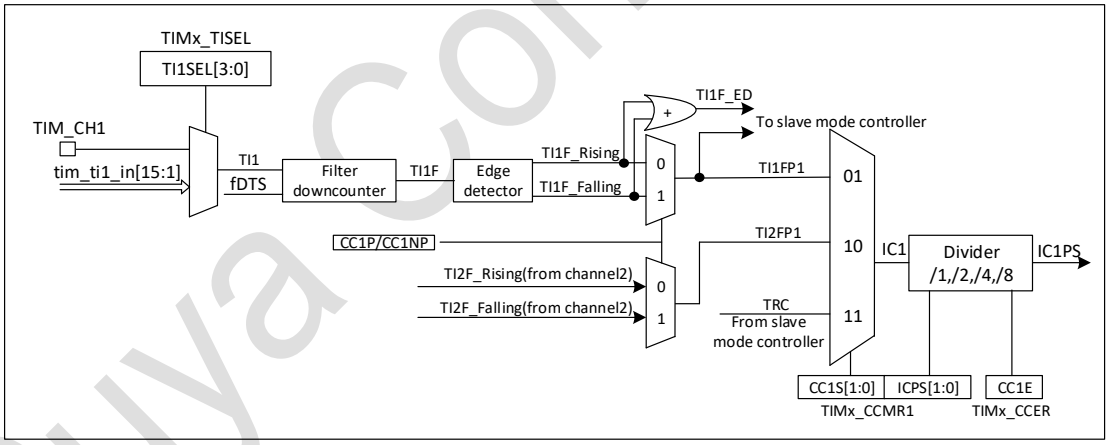


图 19-27 捕获/比较通道（如：通道 1 输入部分）

输出部分产生一个中间波形 OCxREF（高有效）作为基准，链的末端决定最终输出信号的极性。

Puya Confidential

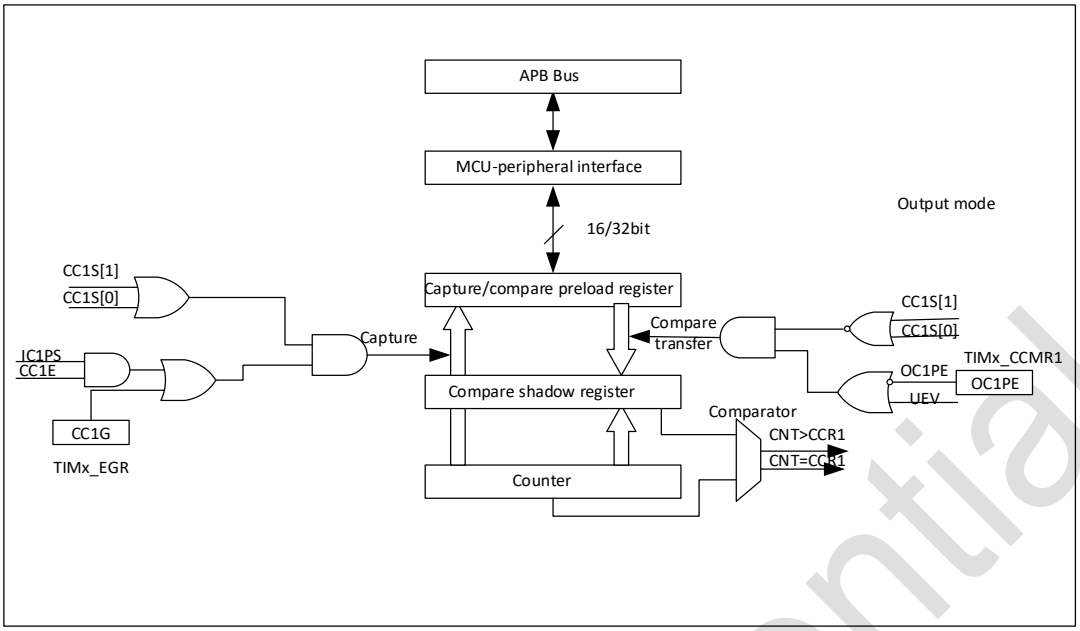


图 19-28 捕获/比较通道 1 的主电路

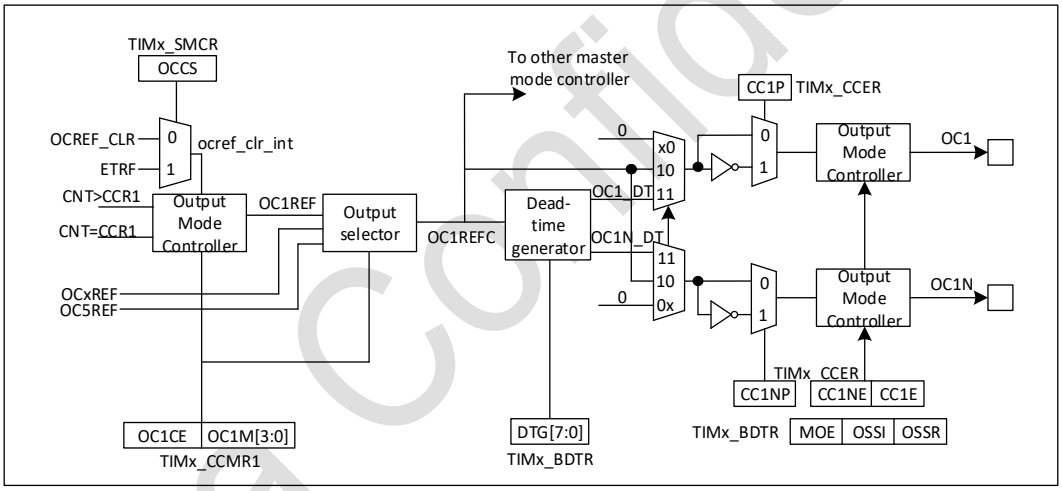


图 19-29 捕获/比较通道的输出部分（通道 1 至 4）

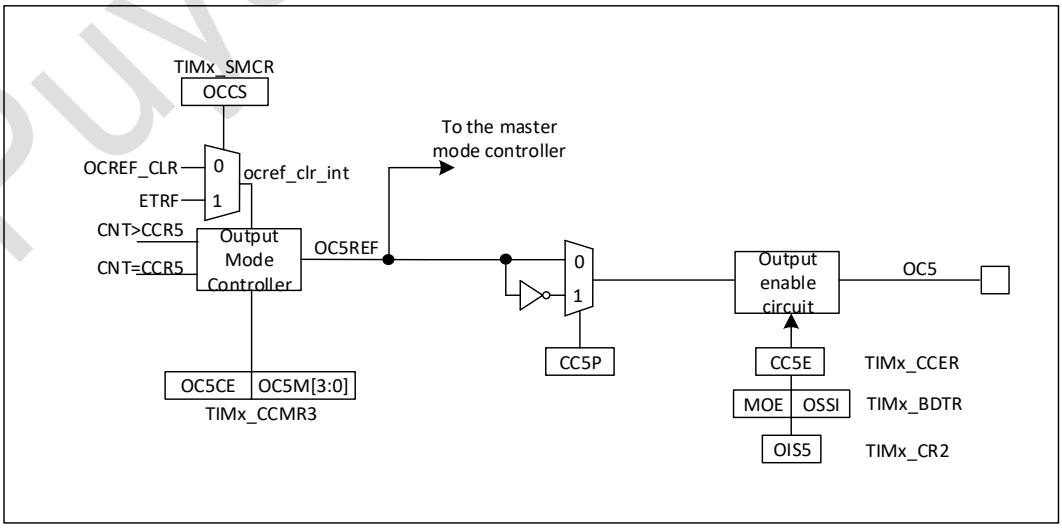


图 19-30 捕获/比较通道的输出部分（通道 5，6）

Puya Confidential

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

19.3.8 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器(TIMx_CCRx)中。当发生捕获事件时，相应的 CCxIF 标志(TIMx_SR 寄存器)被置 1，如果使能中断或者 DMA 操作，则将产生中断或者 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高，那么过捕获标志 CCxOF(TIMx_SR 寄存器)被置 1。通过写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIMx_CCMR1 必须连接到 TI1 输入，所以写入 TIMx_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为‘00’，通道就被配置为输入，并且 TIMx_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽（即输入为 TIx 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位）。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以（以 f_{DTS} 频率）连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0（设置为上升沿）。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIMx_CCMR1 寄存器的 IC1PS=00）。
- 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，可以通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求，也可通过设置 TIMx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志位被设置（中断标志）。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

为了处理过捕获，建议在过捕获标志被置起之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：设置 TIMx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

19.3.9 PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，其余操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，用户可以测量输入到 TI1 上的 PWM 信号的周期（TIMx_CCR1 寄存器）和占空比（TIMx_CCR2 寄存器），具体步骤如下

Puya Confidential

- 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01（选中 TI1）。
- 选择 TI1FP1 的有效极性（用来捕获数据到 TIMx_CCR1 中和清除计数器）：置 CC1P=0（上升沿有效）。
- 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10（选中 TI1）。
- 选择 TI1FP2 的有效极性(捕获数据到 TIMx_CCR2)：置 CC2P=1（下降沿有效）。
- 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101（选择 TI1FP1）。
- 配置从模式控制器为复位模式：置 TIMx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMx_CCER 寄存器中 CC1E=1 且 CC2E=1。

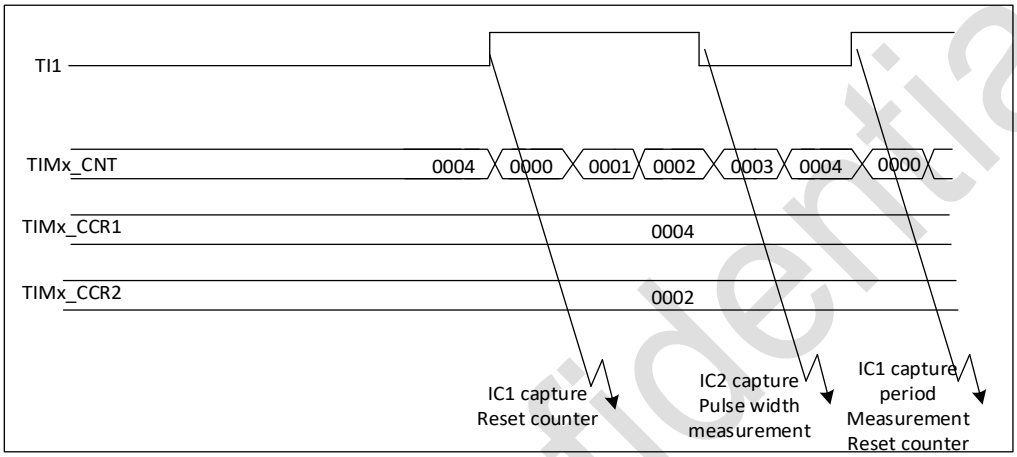


图 19-31 PWM 输入模式时序

因为只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIMx_CH1/TIMx_CH2 信号。

19.3.10 强置输出模式

在输出模式（TIMx_CCMRx 寄存器中 CCxS=00）下，输出比较信号（OCxREF 和相应的 OCx/OCxN）能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRx 寄存器中相应的 OCxM=0101，即可强置输出比较信号（OCxREF/OCx）为有效状态。这样 OCxREF 被强置为高电平（OCxREF 始终为高电平有效），同时 OCx 得到 CCxP 极性相反的信号。

例如： CCxP=0（OCx 高电平有效），则 OCx 被强置为高电平。

置 TIMx_CCMRx 寄存器中的 OCxM=0100，可强置 OCxREF 信号为低。

该模式下，在 TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

19.3.11 输出比较模式

此项功能是用来控制一个输出波形，或者表明一段给定的的时间已经到时。通道 1 到 4 可以输出，但通道 5 和 6 仅在 MCU 内部使用（例如用于产生混合波形或用于 ADC 触发）。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平（OCxM=0000）、被设置成有效电平（OCxM=0001）、被设置成无效电平（OCxM=0010）或进行

Puya Confidential

翻转（OCxM=0011）。

- 设置中断状态寄存器中的标志位（TIMx_SR 寄存器中的 CCxIF 位）。
- 若设置了相应的中断屏蔽（TIMx_DIER 寄存器中的 CCxIE 位），则产生一个中断。
- 若设置了相应的使能位（TIMx_DIER 寄存器中的 CCxDE 位，TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能），则产生一个 DMA 请求。

可以通过配置 TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。同步的精度可以达到计数器的一个计数周期。输出比较模式（在单脉冲模式下）也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟（内部，外部，预分频器）。
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
   - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=0011
   - 置 OCxPE = 0 禁用预装载寄存器
   - 置 CCxP = 0 选择极性为高电平有效
   - 置 CCxE = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器（OCxPE='0'，否则 TIMx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新）。下图给出了一个例子。

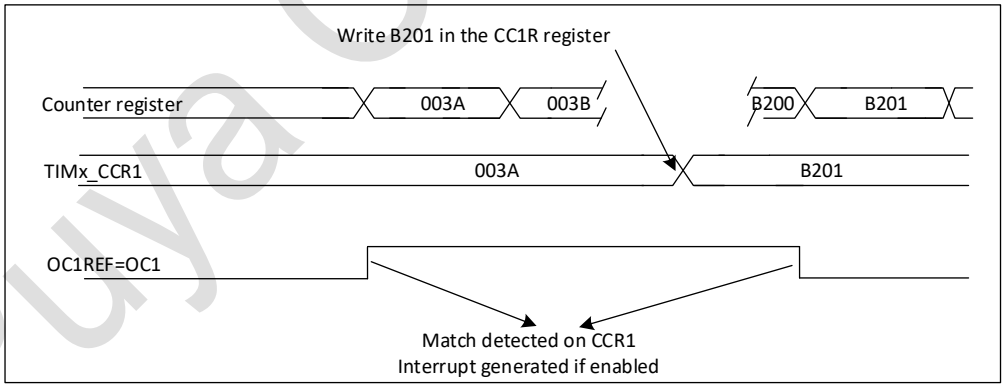


图 19-32 输出比较模式，翻转 OC1

19.3.12 PWM 模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比的信号。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入'0110'（PWM 模式 1）或'0111'（PWM 模式 2），能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMx_CCMRx 寄存器的 OCxPE 位使能相

Puya Confidential

应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位，（在向上计数或中心对称模式中）使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，用户必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。OCx 的输出使能通过（TIMx_CCER 和 TIMx_BDTR 寄存器中）CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 TIMx_CCER 寄存器的描述。

在 PWM 模式（模式 1 或模式 2）下，TIMx_CNT 和 TIMx_CCRx 始终在进行比较，（依据计数器的计数方向）以确定是否符合 TIMx_CCRx≤TIMx_CNT 或者 TIMx_CNT≤TIMx_CCRx。

根据 TIMx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

PWM 边沿对齐模式

■ 向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。

下面是一个 PWM 模式 1 的例子。当 TIMx_CNT<TIMx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCRx 中的比较值大于自动重装载值（TIMx_ARR），则 OCxREF 保持为‘1’。如果比较值为 0，则 OCxREF 保持为‘0’。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

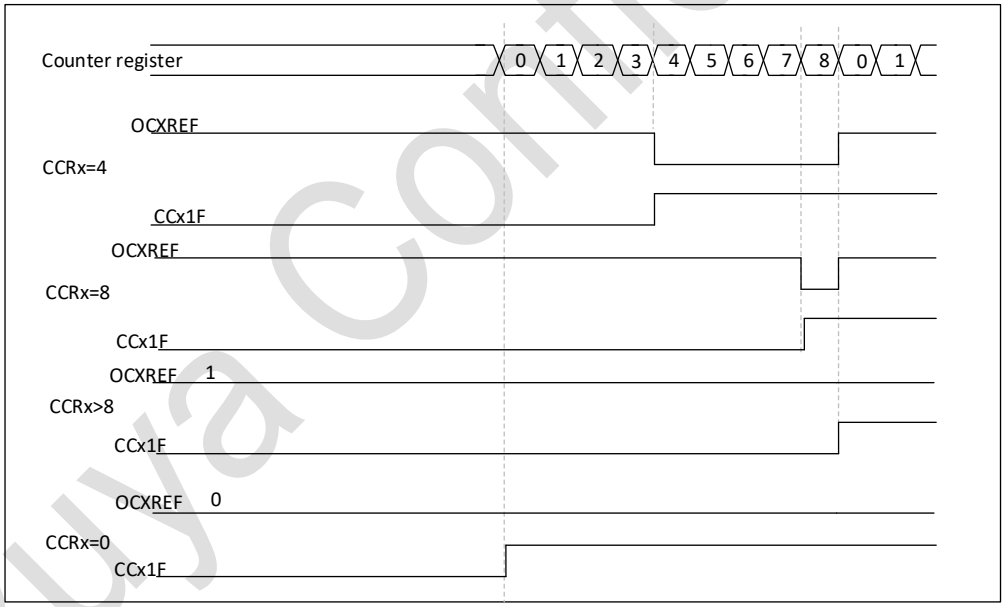


图 19-33 边沿对齐的 PWM 波形(ARR=8)

■ 向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1，当 TIMx_CNT>TIMx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重装载值，则 OCxREF 保持为‘1’。该模式下不能产生 0%的 PWM 波形。

Puya Confidential

PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为‘00’时为中央对齐模式（CMS 位的所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位（DIR）由硬件更新，不要用软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子

- TIMx_ARR=8
- PWM 模式 1
- TIMx_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

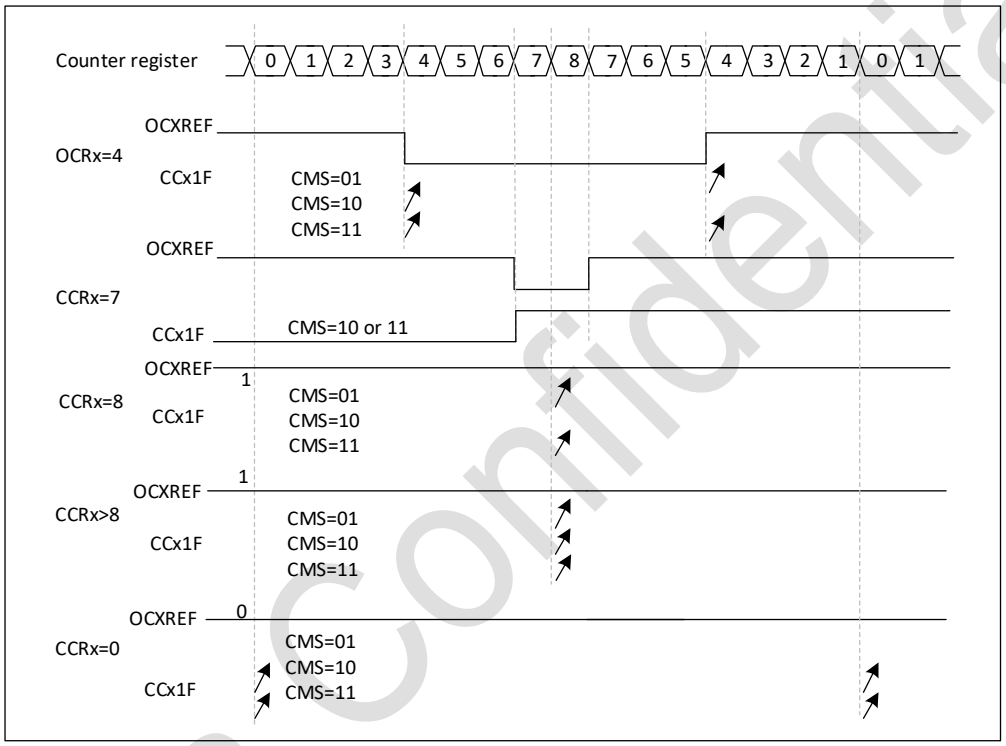


图 19-34 中央对齐的 PWM 波形(APR=8)

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIMx_CR1 寄存器中 DIR 位的当前值。此外，不要通过软件同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：

– 如果写入计数器的值大于自动重加载的值(TIMx_CNT>TIMx_ARR)，则方向不会被更新。

例如，如果计数器正在向上计数，它就会继续向上计数。

– 如果将 0 或者 TIMx_ARR 的值写入计数器，方向被更新，但不会产生更新事件 UEV。

- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新(设置 TIMx_EGR 位中的 UG 位)，并且不要在计数进行过程中修改计数器的值。

PWM 移相模式

PWM 移相模式是指将脉冲信号的相位进行调整。在 PWM 中央对齐模式中，可以通过调整移相功能，将输出信号的相位相对于输入信号进行偏移。这一功能可以用于控制电子设备的输出波形，实现相位差的调整。

Puya Confidential

在 PWM 中央对齐模式（CMS 不为 00）下使能 PWM 移相模式，通过在各 PWM 周期产生一次中断（根据 TIMx_AF1 中的 INTR_SEL 选择中断位置），更新寄存器输出比较值 CCR1、CCR2、CCR3（其中 CCRx[15:0]为向上计数模式的比较值，CCRx[31:16]为向下计数模式的比较值）产生 PWM 移相功能。PWM 移相结构图如下图。

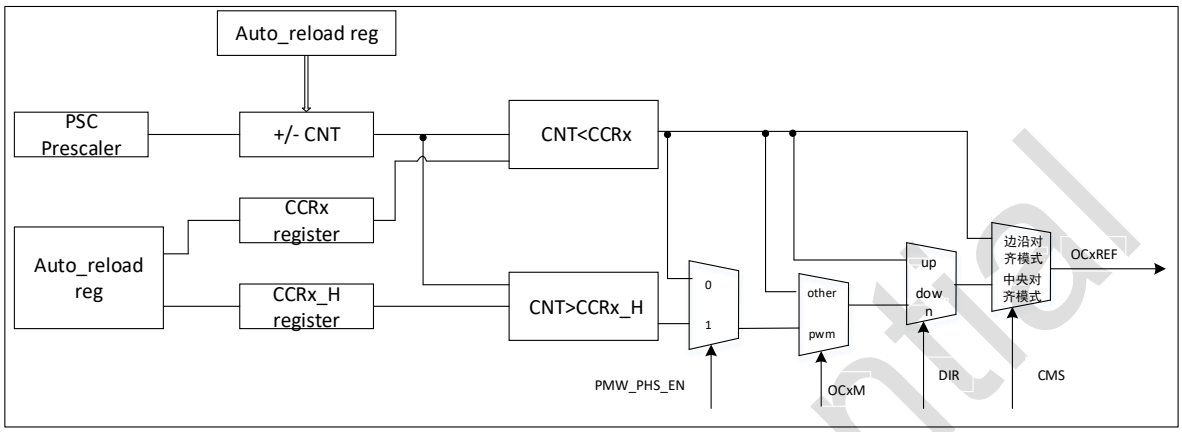


图 19-35 PWM 移相模式结构概览

下图为 PWM2 输出模式，使能预装载功能（OCxPE=1），timx_intr 在下溢时产生中断（上溢/下溢中断可用 INTR_SEL 选择），将预先设置的 CCRx/CCRx_H 的预装载值在更新事件到来时更新到寄存器 CCRx_SHA/CCRx_H_SHA 中。通过每次更新中断后设置 CCRx_SHA/CCRx_H_SHA 寄存器的值来实现 PWM 移相的功能。

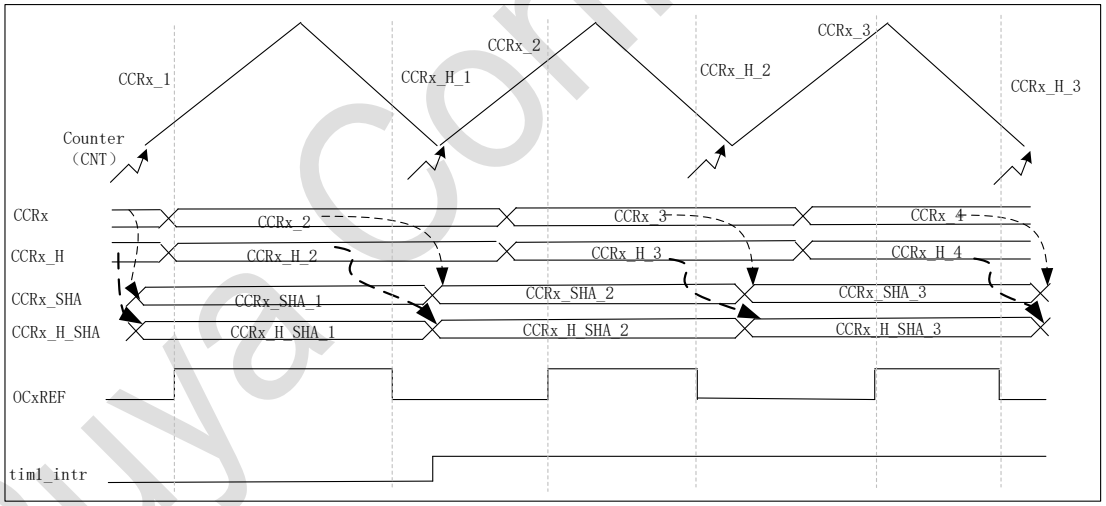


图 19-36 PWM2 输出模式，使能预装载功能(OCxPE=1)

注：PWM 移相模式更新 CCRx_SHA/CCRx_H_SHA 寄存器，有两种方法可以选择：

预装载功能使能时(OCxPE=1)，只有在更新事件到来时才会将预先设置的预装载值更新到 CCRx_SHA /CCRx_H_SHA 寄存器中。

预装载功能无效（OCxPE=0），CCRx_SHA /CCRx_H_SHA 的值可以实时进行更新（使用此方式时，向上计数模式时不能更改 CCRx_SHA（向下计数模式时不能更改 CCRx_H_SHA），否则，会不满足 PWM 移相功能。如下图为 PWM2 输出模式下，预装载功能无效(OCxPE=0)，timx_intr 在下溢时产生中断，向上计数时实时更新寄存器 CCRx，此种情况不满足 PWM 移相模式。

Puya Confidential

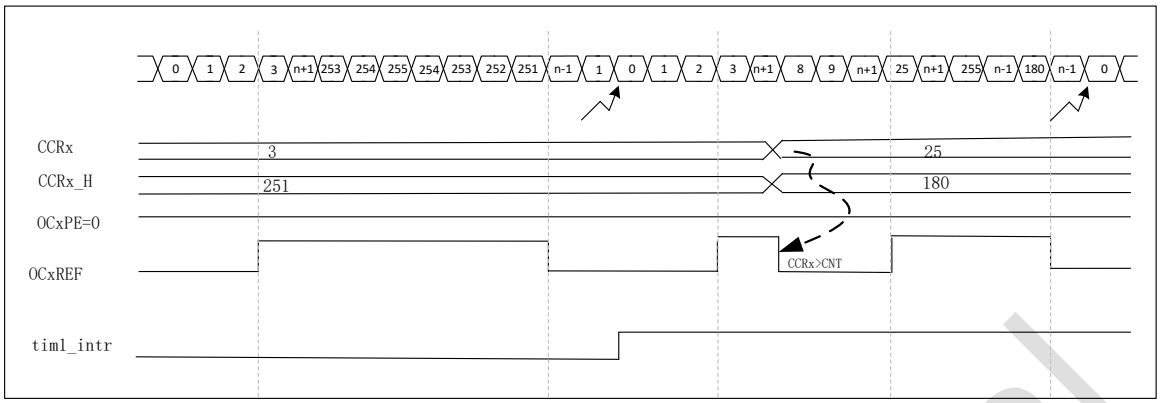


图 19-37 PWM2 输出模式下，预装载功能无效(OCxPE=0)

PWM 移相配置步骤：

假定计数器选择内部时钟源:

- 选择中央对齐模式：根据 TIMx_CR1 寄存器中的 CMS 选择中央对齐模式
- 选择预装载使能：置 TIMx_CCMRx 寄存器中的 OCxPE=1 使能输出比较 1、2、3 预装载使能。
- 配置比较寄存器 TIMx_CCRx[31:0]，置 UG 位产生一个更新事件，更新比较寄存器值。
- 置 TIMx_AF1 寄存器中的 PWM_PHS_EN=1,PWM 移相使能，此例中选择 ISR_SEL=10 下溢中断。
- 配置 TIMx_ARR 自动重装载寄存器。
- 选择 PWM2 输出模式：置 TIMx_CCMRx 中的 OCxM=4’b0111。
- 置 TIMx_CR1 寄存器中的 CEN=1 使能计数器，如果设置了 UIE 位，则会产生更新中断。
- 预先配置 TIMx_CCRx，出现更新事件后，预装载值更新到本地寄存器中。

19.3.13 互补输出和死区插入

高级控制定时器（TIM1/TIM8）能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。

这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

配置 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性（主输出 OCx 或互补输出 OCxN）。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是，在转换到 IDLE 状态时（MOE 下降到 0）死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。通过配置 TIMx_BDTR 寄存器中的 DTG[7:0]位，可以控制所有通道的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效：

- OCx 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度（OCx 或者 OCxN），则不会产生相应的脉冲。

Puya Confidential

下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。(假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

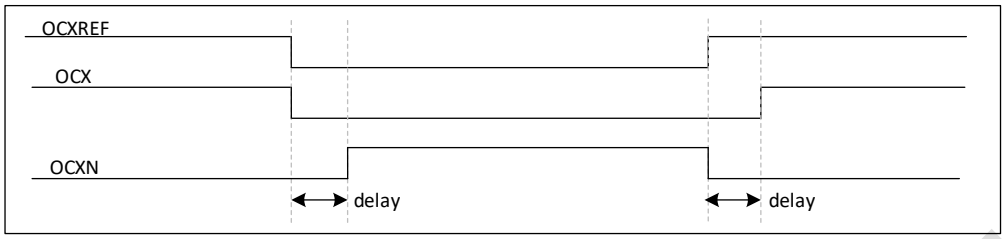


图 19-38 带死区插入的互补输出

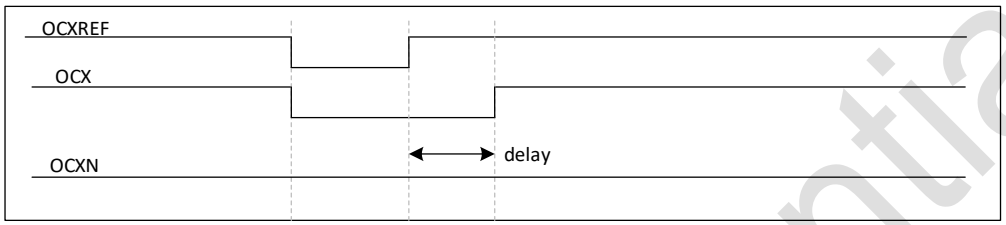


图 19-39 死区波形延迟大于负脉冲

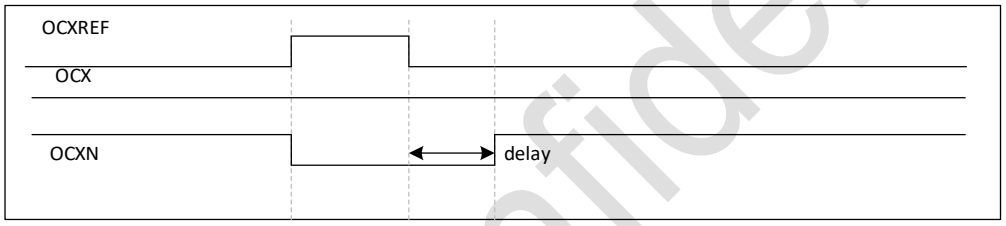


图 19-40 死区波形延迟大于正脉冲

当 DTAE=1，上升沿死区时间由 TIMx_BDTR 的 DTG[7:0]位配置，下降沿死区时间由 TIMx_DTR2 的 DTGF[7:0]位配置。DTAE 的写入需要在计数器使能前，当 CEN=1 时，不允许修改 DTAE。

可以通过预加载的机制在 PWM 工作过程中修改死区时间，当 TIMX_DTR2 的 DTPE 被置位时死区时间配置寄存器 DTG[7:0]和 DTGF[7:0]被预加载，在下一个更新事件时预加载的数据被加载到缓存寄存器中。

注：如果在计数器被使能时，DTPE 位被置位，则自最后一次更新以来的写的新数据都会被丢弃，而使用之前的值

Puya Confidential

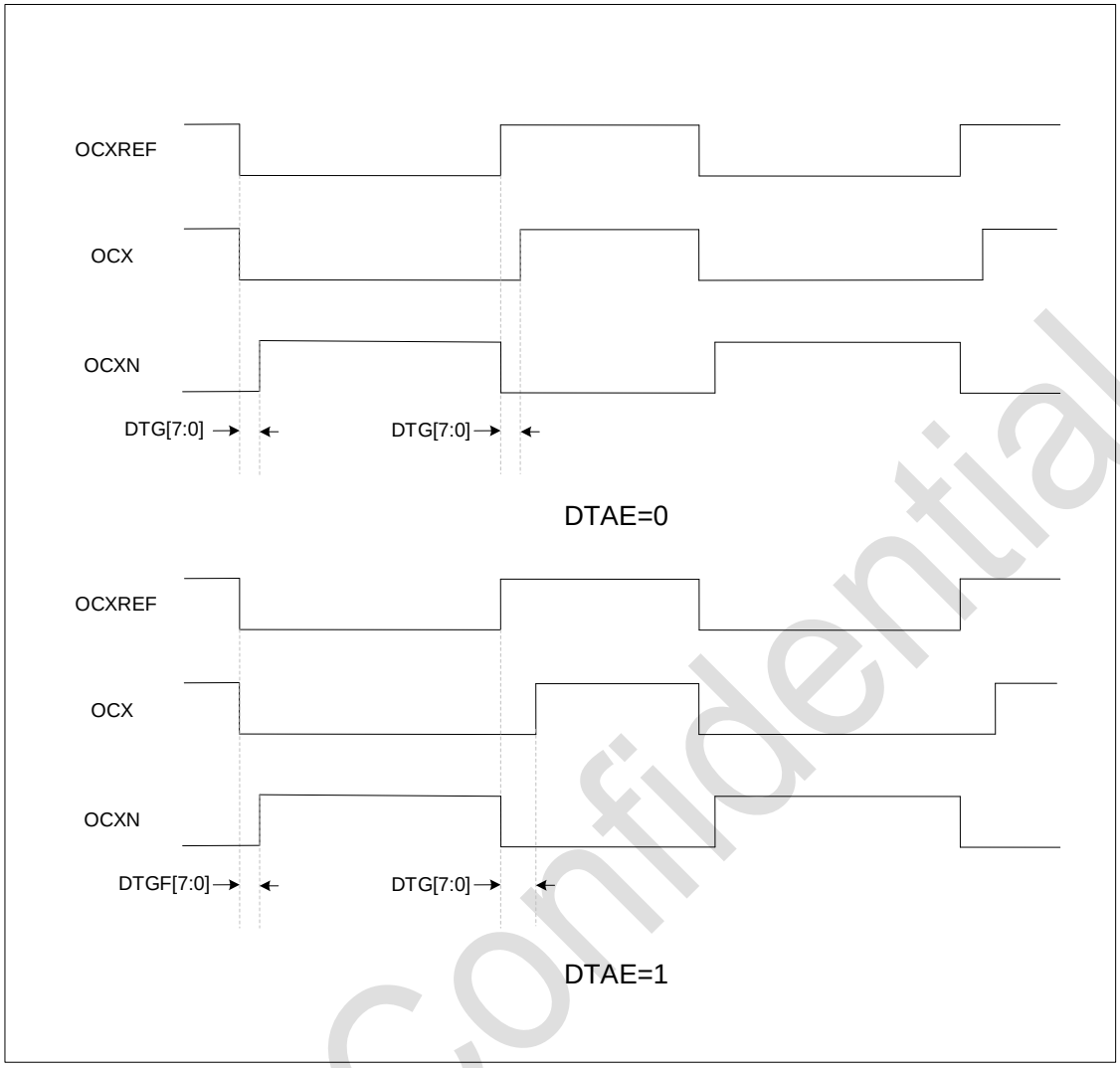


图 19-41 非对称死区时间

重定向 OCxREF 到 OCx 或 OCxN

在输出模式下(强置、输出比较或 PWM),通过配置 TIMx_CCER 寄存器的 CCxE 和 CCxNE 位,OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。

这个功能可以在互补输出处于无效电平时，在某个输出上送出一个特殊的波形（例如 PWM 或者静态有效电平）。另一个作用是，让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

注：当只使能 OCxN（CCxE=0, CCxNE=1）时，它不会反相，当 OCxREF 有效时立即变高。例如，如果 CCxNP=0，则 OCxN=OCxREF。另一方面，当 OCx 和 OCxN 都被使能时(CCxE=CCxNE=1)，当 OCxREF 为高时 OCx 有效；而 OCxN 相反，当 OCxREF 低时 OCxN 变为有效。

19.3.14 使用刹车功能

刹车功能的目的是保护由 TIMER 输出的 PWM 驱动的电源开关。两路刹车的输入通常是连接到三相逆变器和功率级芯片的故障输出。当刹车信号有效时，会立即关断 PWM 的所有输出，并将它们强制到预先设置好的一个安全的电平状态。部分 MCU 内部的事件也可以作为触发信号来关闭输出。

Puya Confidential

刹车事件有两路通道，通道 1 包括了系统级的故障（clock failure、ECC、奇偶校验等）和应用层的故障（通过输入引脚或者内置的比较器），并且可以在一个死区区间后强制输出到预先设置好的电平（无论工作还是空闲）。

在刹车过程中输出使能信号和输出电平由以下多个 bit 控制

- TIMx_BDTR 的 MOE 位可以通过软件或 2 路刹车事件复位来实现使能和关闭输出
- TIMx_BDTR 的 OSSI 位用于定义当处于空闲状态时 timer 是否还控制输出，或者释放对 GPIO 控制器的控制（高阻状态）

TIMx_CR2 的 OISx 和 OISxN 位用于无论工作还是空闲状态下都将输出设置为无效电平，OCx 和 OCxN 输出在同一时刻不能同时设置为有效电平，无论 OISx 和 OISxN 为何值。

当从复位中退出时，刹车功能被关闭并且 MOE 为 0。刹车功能的开启可通过使能 TIMxBDTR 的 BKE 位。刹车输入的极性可通过 BKP 配置。BKEx 和 BKPx 可同时修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的，在实际信号（作用在输出端）和同步控制位（在 TIMx_BDTR 寄存器中）之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当它为低时写 MOE=1，则读出它之前必须先插入一个延时（空指令）才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

刹车通道 1 的源；

- 外部源可连接到其中一个 TIMx_BKIN 引脚上（选择 GPIO 和配置寄存器），再加上极性选择和滤波
- 内部源包含 2 部分
  - 来自刹车比较器信号（tim_brk_cmpx）
  - 来自系统刹车请求

刹车通道 2 的源；

- 外部源可连接到其中一个 TIMx_BKIN2 引脚上（选择 GPIO 和配置寄存器），极性选择和滤波
- 内部源来自刹车比较器信号（tim_brk2_cmpx）

Puya Confidential

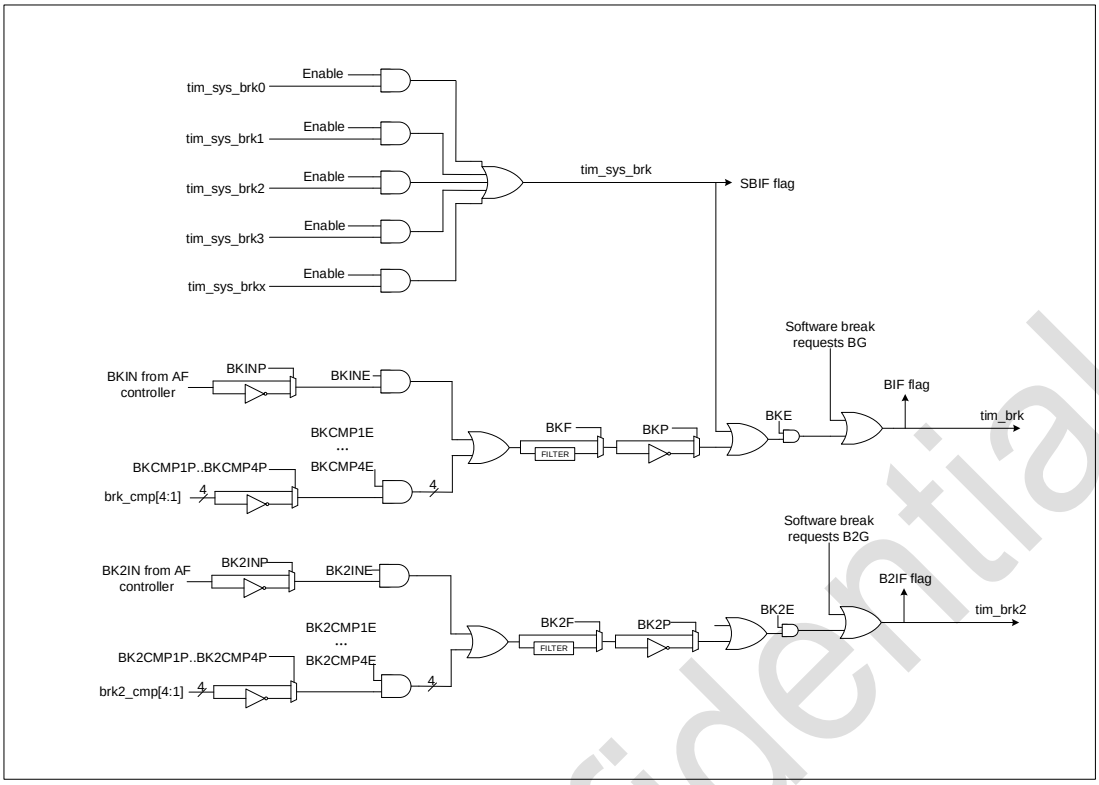


图 19-42 刹车电路

注意：只有没有滤波的时候才能保证是异步的操作，如果滤波开启了，必须使用一个故障保护时钟模式来保证刹车中断事件得到处理（例如使用内部 PLL 或 CSS，无滤波）。

当发生刹车时（在刹车输入端出现选定的电平），有下述动作：

- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者释放对 GPIO 控制器的控制（由 OSSI 位选择）。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIMx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放输出控制，否则使能输出始终为高。
- 当使用互补输出时：
  - 输出首先被置于复位状态即无效的状态（取决于极性）。这是异步操作，即使定时器没有时钟时，此功能也有效。
  - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下， OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些（大约 2 个时钟周期）。
  - 如果 OSSI=0，定时器释放输出，否则保持使能输出；或一旦 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 刹车状态标志会被拉高（TIMx_SR 的 SBIF,BIF）。如果设置了 TIMx_DIER 寄存器中的 BIE 位，当刹车状态标志（TIMx_SR 寄存器中的 BIF 位）为‘1’时，则产生一个中断。如果设置了 TIMx_DIER 寄存器中的 TDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位；例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置‘1’；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

Puya Confidential

注：当 AOE 为 1 时如果 MOE 被 CPU 复位了，输出会进入空闲状态，并被强制到无效电平或者高阻（取决于 OSSI 的值），如果 MOE 和 AOE 均被 CPU 复位，输出将进入无效状态并且由 OISx 位来驱动输出电平。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能同时（自动地或者通过软件）设置 MOE。同时，状态标志 BIF 不能被清除。

除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区持续时间，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。用户可以通过设置 TIMx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看刹车和死区寄存器(TIMx_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

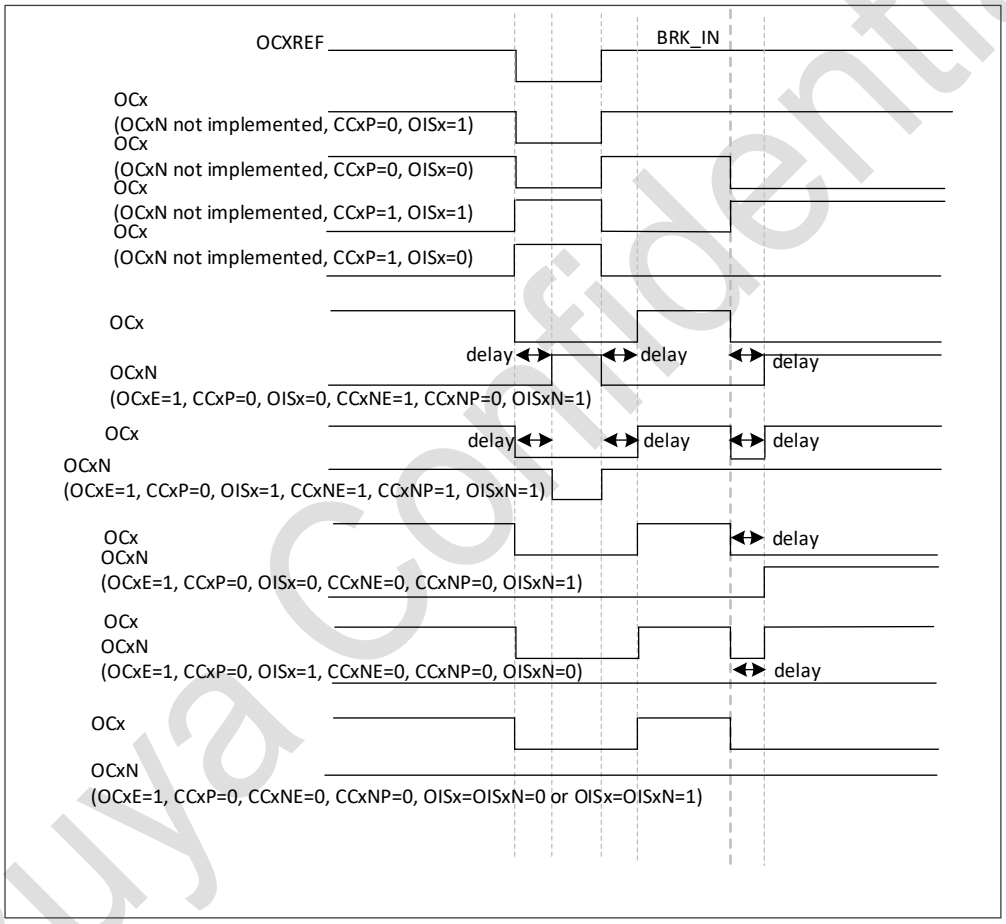


图 19-43 响应刹车的输出

Puya Confidential

19.3.15 双向刹车输入

TIM_TYP1 具备双向的刹车 I/O，如下图所示：

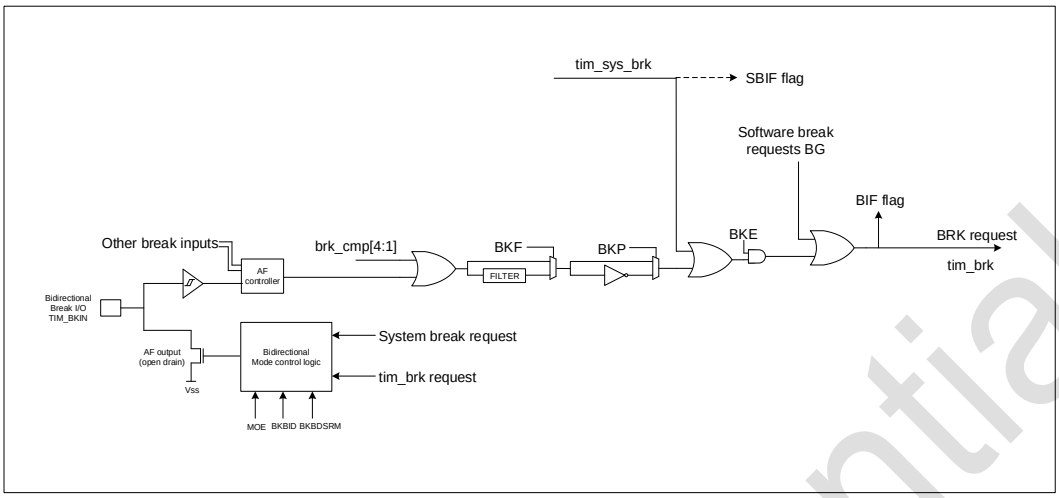


图 19-44 输出重定向（刹车 2 无请求时）

I/O 支持特性：

- 提供一种板级的全局刹车信号，通过唯一的既可做输入也可做输出的引脚，向外部 MCU 和栅极驱动器发送故障信号
- 当需要合并多个内部和外部的刹车源时，内部的刹车源和多个外部的开漏源进行或运算后作为刹车事件的的唯一触发信号

通过 TIMxBDTR 的 BKBID 和 BK2BID 位控制 tim_brk 和 tim_brk2 的输入为双向模式。BKBID 和 BK2BID 可以使用 TIMxBDTR 的 LOCK 位（LOCK level 1）设置为 1，实现被锁定为只读模式。

tim_brk 和 tim_brk2 输入都可使用双向模式，并且要求 I/O 配置成开漏模式且极性为低有效(通过 BKINP、BKP、BK2INP、BK2P 控制位）。无论来自系统的（例如 CCS）、还是板级外设的、还是刹车输入强制一个低电平表示刹车输入故障事件，都可以作为刹车请求。但出于安全的考虑，当极性位未有效配置时（配置成高电平有效时），双向模式会被禁止。

软件刹车事件 (BG、B2G) 也会引起刹车 I/O 被拉低，用于向外部设备显示本 timer 进入到了刹车状态。当然当 BKE 或 B2KE 为 1 时才有可以。当 BKE 或 B2KE 为 0，发生软件刹车事件，输出会被置为安全的状态并产生刹车标志信号，但是对 TIMx_BKIN 和 TIMx_BKIN2 I/O 没有影响

一种安全的解除机制可以防止系统被限制性锁死（在刹车输入口的低电平触发了一个刹车，该刹车又将输入口强制成低电平）

当 BKDSRM（BK2DSRM）位被置为 1 时，可以释放刹车输出控制，清除故障标志信号并提供重新配置系统的可能性。

任何情况下都不能关闭刹车保护电路：

- 刹车输入路径一直有效：即使 BKDSRM（BK2DSRM）位被置 1 和开漏控制被释放，刹车事件还是有效的。这防止在还处于刹车状态时 PWM 输出可以重启
- 当输出被使能的时候（MOE=1）,BKDSRM（BK2DSRM）位无法解除刹车保护

Puya Confidential

表 19-4 刹车保护状态

MOE	BKBID（BK2BID）	BKDSRM（BK2DSRM）	刹车保护状态
0	0	X	启用
0	1	0	启用
0	1	1	解除
1	X	X	启用

19.3.15.1 对刹车电路进行启用和重启用

默认情况下，刹车电路（在输入或双向模式下）处于启用状态

在发生刹车（或刹车 2）后需要遵循下面的程序来重启动保护：

- BKDSRM(BK2DSRM)位必须置 1 用于释放输出控制
- 软件必须要等到系统刹车状态消失然后清除 SBIF 状态标志（或者在重启用前系统地清除它）
- 软件必须轮询 BKDSRM(BK2DSRM)位直到它被硬件清除（当应用刹车状态消失）

通过这种方式，刹车电路被激活，并且 MOE 位可以被重新使能 PWM 输出

19.3.16 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIMx_CCMRx 寄存器中对应的 OCxCE 位为‘1’，能够用 OCREF_CLR_INT 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次计数溢出所产生的更新事件 UEV。该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

而 OCREF_CLR_INT 可以通过配置 TIMx_SMCR 寄存器中的 OCCS 位，在 OCREF_CLR 和 ETRF(ETR 滤波后)之间选择。

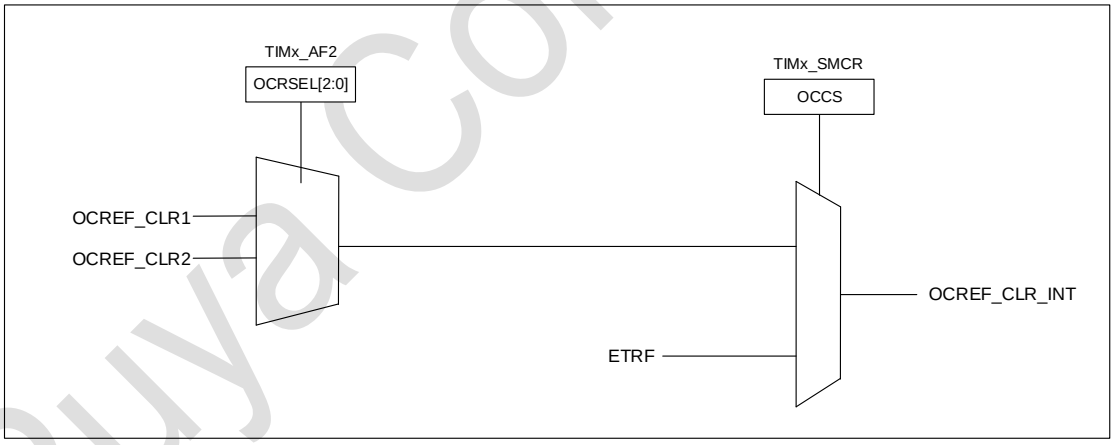


图 19-45 OCREF_CLR 输入选择

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时， ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭： TIMx_SMCR 寄存器中的 ETPS[1:0]=00。
2. 必须禁止外部时钟模式 2： TIMx_SMCR 寄存器中的 ECE=0。
3. 外部触发极性（ETP）和外部触发滤波器（ETF）可以根据需要配置。

下图显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子中，定时器 TIMx 被置于 PWM1 模式。

Puya Confidential

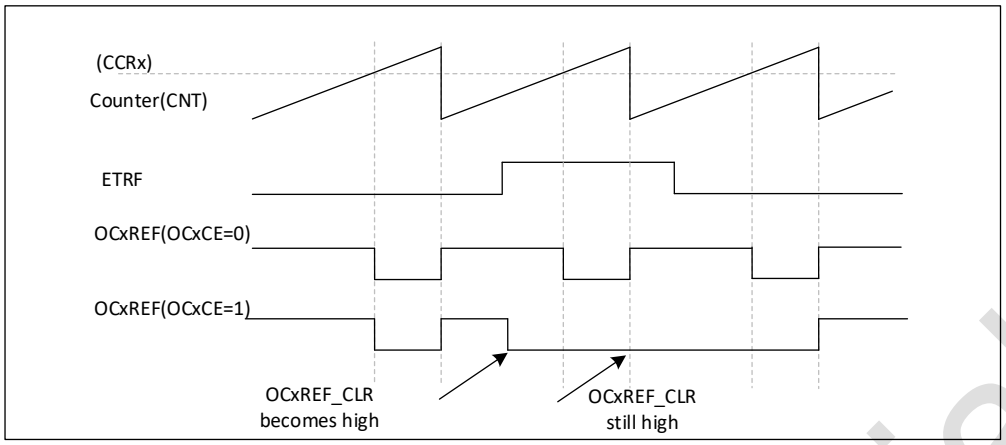


图 19-46 清除 TIMx 的 OCxREF

19.3.17 产生六步 PWM 输出

当在一个通道上需要互补输出时，预装载位有 OCxM、CCxE 和 CCxNE。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器。这样你就可以预先设置好下一步骤配置，并在同一个时刻同时更改所有通道的配置。COM 可以通过设置 TIMx_EGR 寄存器的 COMG 位由软件产生，或在 TRGI 上升沿由硬件产生。

当发生 COM 事件时会设置一个标志位（TIMx_SR 寄存器中的 COMIF 位），这时如果已设置了 TIMx_DIER 寄存器的 COMIE 位，则产生一个中断；如果已设置了 TIMx_DIER 寄存器的 COMDE 位，则产生一个 DMA 请求。

下图显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出。

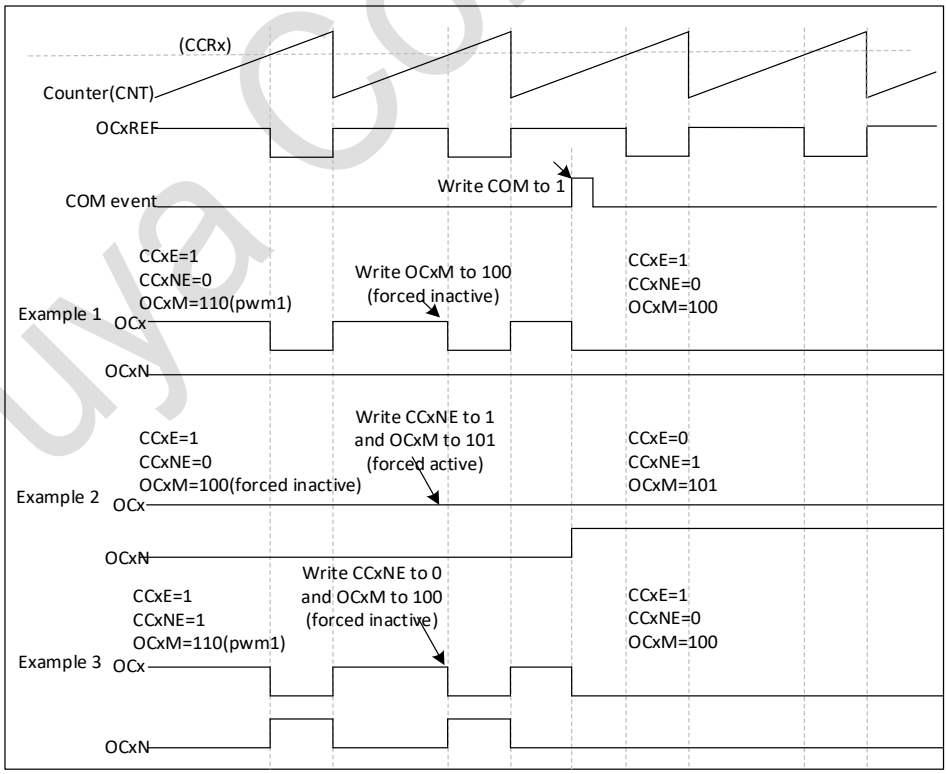


图 19-47 产生六步 PWM，使用 COM 的例子（OSSR=1）

Puya Confidential

19.3.18 单脉冲模式

单脉冲模式（OPM）是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须如下配置：

- 向上计数方式：计数器 CNT < CCRx ≤ ARR（特别地，0 < CCRx），
- 向下计数方式：计数器 CNT > CCRx。

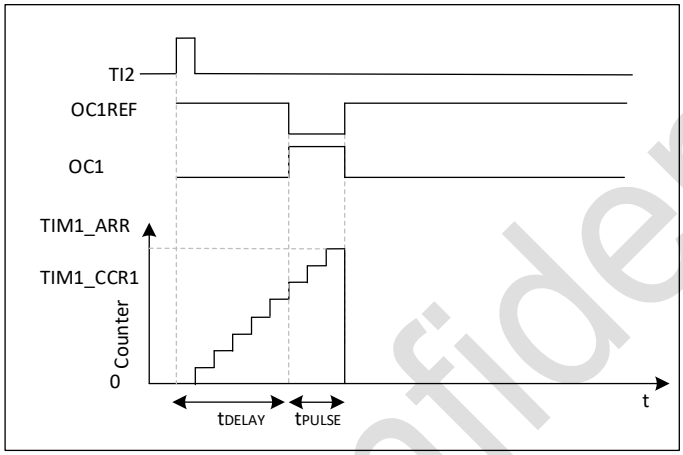


图 19-48 单脉冲模式的例子

例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 tDELAY 之后，在 OC1 上产生一个长度为 tPULSE 的正脉冲。

假定 TI2FP2 作为触发:

- 置 TIMx_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映射到 TI2。
- 置 TIMx_CCER 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- 置 TIMx_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发（TRGI）。
- 置 TIMx_SMCR 寄存器中的 SMS=110(触发模式)，TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定（要考虑时钟频率和计数器预分频器）

- t_{DELAY} 由 TIMx_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义（TIMx_ARR - TIMx_CCR1）。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 0 到 1 的波形；首先要置 TIMx_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMR1 中的 OC1PE=1 和 TIMx_CR1 寄存器中的 ARPE；然后在 TIMx_CCR1 寄存器中填写比较值，在 TIMx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲，所以必须设置 TIMx_CR1 寄存器中的 OPM=1，在下一个更新事件（当计数器从自动装载值翻转到 0）时停止计数。当 OPM=0 时，重复模式被选中。

Puya Confidential

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY}。如果要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF（和 OCx）直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

19.3.19 可重触发的单脉冲模式

该模式允许计数器响应激励启动并产生长度可编程的脉冲，与上节所述的不可再触发单脉冲模式有以下区别：

- 触发一发生，脉冲就开始（无可编程延迟）。
- 如果在前一个触发所产生的脉冲完成之前发生新的触发，则延长脉冲。

要使用可重触发的单脉冲模式，计时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0]=“1000”（组合模式-复位+触发），OCxM[3:0]位设置为“1000”或“1001”（可重新触发 OPM 模式 1 或 2）。

如果此时计时器配置为递增计数模式，则相应的 CCRx 必须设置为 0（ARR 寄存器设置脉冲长度）。

如果计时器配置为递减计数模式，CCRx 必须大于或等于 ARR。

注：出于兼容性原因，OCxM[3:0]和 SMS[3:0]位字段被分成两部分，最高有效位与 3 个最低有效位位置不连续。

该模式不得与中央对齐计数模式一起使用。TIMx_CR1 中必须为 CMS[1:0]=00。

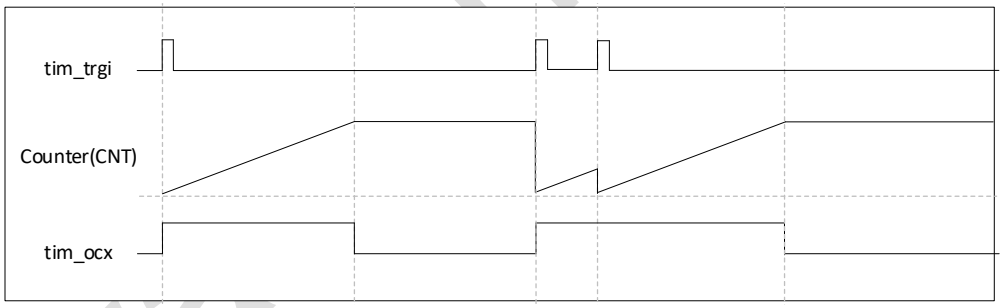


图 19-49 可重触发的单脉冲模式的例子

19.3.20 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。下节给出了此特性用于连接霍尔传感器的例子。

19.3.21 与霍尔传感器的接口

使用高级控制定时器（TIM1/TIM8）产生 PWM 信号驱动马达时，可以用另一个通用 TIMx（TIM2、TIM3、TIM4）定时器作为“接口定时器”来连接霍尔传感器，见下图，3 个定时器输入引脚（TI1、TI2、TI3）通过一个异或门连接到 TI1 输入通道（通过设置 TIMx_CR2 寄存器中的 TI1S 位来选择），此时“接口定时器”被用来捕获这个信号。

从模式控制器被配置成复位模式，从输入是 TI1F_ED。每当 3 个输入之一变化时，计数器重新从 0 开始计数。这样产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”上的捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（如下图）。捕获值反映了两个输入变化间的时间延迟，给出了马达速度的信息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以（通过触发一个 COM 事件）用于改变高级定时器 TIM1/TIM8 各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。

因此“接口定时器”通道必须编程为在一个指定的延时（输出比较或 PWM 模式）之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1/TIM8。

举例：霍尔输入连接到 TIMx 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIMx 的 PWM 配置。

- 置 TIMx_CR2 寄存器的 TI1S 位为‘1’，配置三个定时器输入逻辑异或到 TI1 输入；
- 时基编程：置 TIMx_ARR 为其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔；
- 设置通道 1 为捕获模式（选中 TRC）：置 TIMx_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器；
- 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIMx_CCMR1 寄存器中的 OC2M=111 和 CC2S=00；
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIMx_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIM1/TIM8 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的（TIMx_CR2 寄存器中 CCPC=1），同时触发输入控制 COM 事件（TIMx_CR2 寄存器中 CCUS=1）。在一次 COM 事件后，写入下一步的 PWM 控制位（CCxE、OCxM），这可以在处理 OC2REF 上升沿的中断子程序里实现。

下图显示了这个实例

Puya Confidential

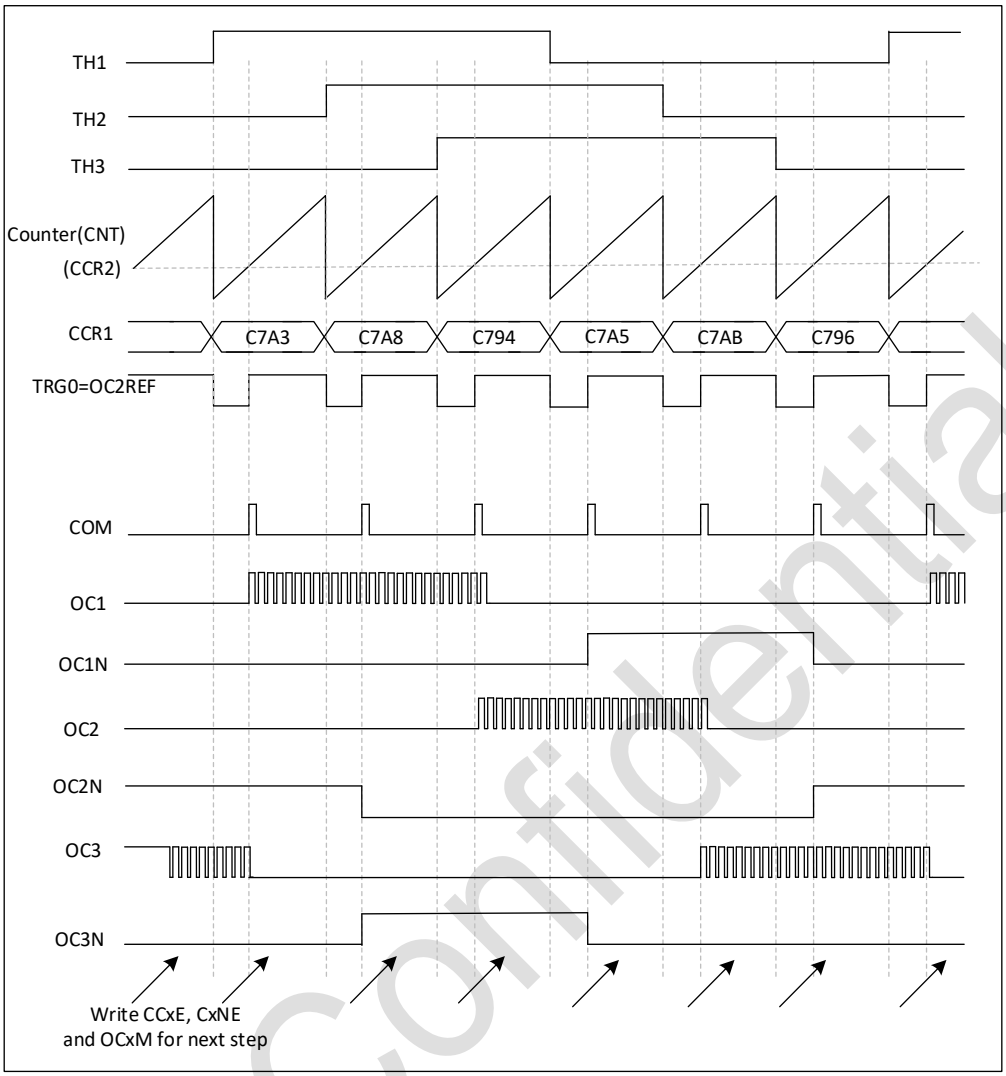


图 19-50 霍尔传感器接口的实例

19.3.22 TIMx 定时器和外部触发的同步

TIMx 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式、触发、复位+触发和门控+复位模式。

从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 UDIS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（TIMx_ARR，TIMx_CCRx）都被更新了。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）。
- 置 TIMx_SMCR 寄存器中 SMS=0100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=00101，选择 TI1 作为输入源。

Puya Confidential

- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）被设置，根据 TIMx_DIER 寄存器中 TIE（中断使能）位和 TDE（DMA 使能）位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

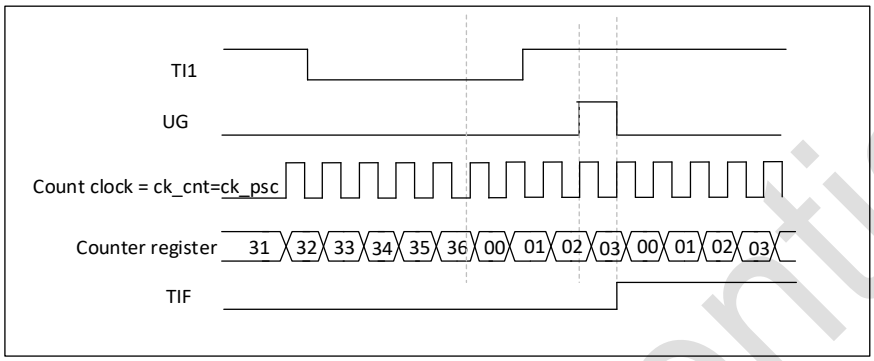


图 19-51 复位模式下的控制电路

从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽（本例中，不需要滤波，所以保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 以确定极性（只检测低电平）。
- 置 TIMx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标置。

TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。

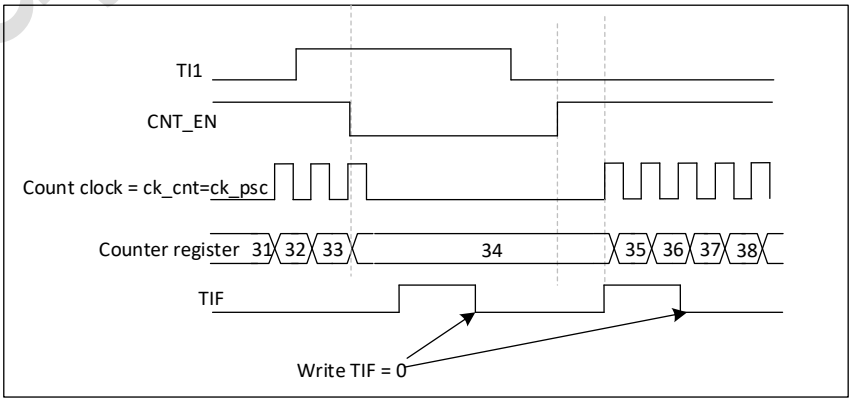


图 19-52 门控模式下的控制电路

Puya Confidential

从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC2S=01。置 TIMx_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

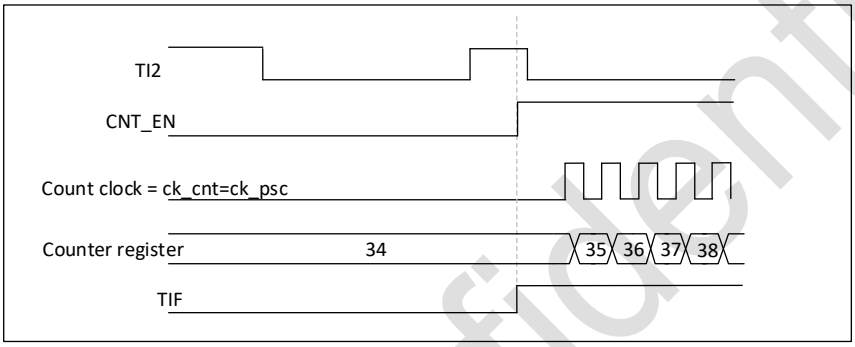


图 19-53 触发器模式下的控制电路

从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可以与另一种从模式（外部时钟模式 1 和编码器模式除外）一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。不建议使用 TIMx_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次：

1. 通过 TIMx_SMCR 寄存器配置外部触发输入电路：
   - ETF=0000：没有滤波
   - ETPS=00：不用预分频器
   - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。
2. 按如下配置通道 1，检测 TI1 的上升沿：
   - IC1F=0000：没有滤波
   - 触发操作中不使用捕获预分频器，不需要配置
   - 置 TIMx_CCMR1 寄存器中 CC1S=01，选择输入捕获源
   - 置 TIMx_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）
3. 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时， TIF 标志被设置，计数器开始在 ETR 的上升沿计数。

ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

Puya Confidential

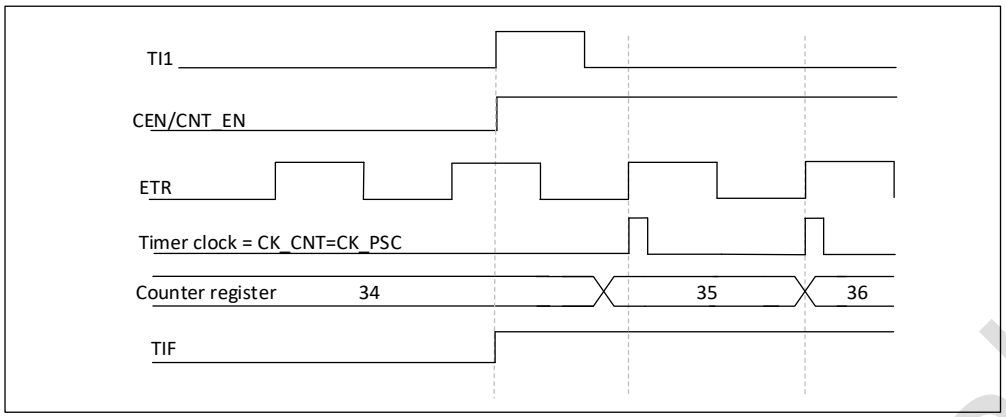


图 19-54 外部时钟模式 2 + 触发模式下的控制电路

19.3.23 DMA 突发模式

TIMx 定时器当发生单个事件时有生成多个 DMA 请求的能力。主要目的是能够多次重新编程定时器的部分功能而无需软件的开销。但是也可以在固定的间隔时间内读取一行中的几个寄存器。

DMA 控制器的目的地是唯一的并且必须指向虚拟寄存器 TIMx_DMAR.当给定的定时器事件发生，定时器发起一系列 DMA 请求。每个写入 TIMx_DMAR 寄存器的操作实际上是重定向一个定时器寄存器的。

TIMx_DCR 寄存器中的 DBL[4:0]位设置 DMA 突发长度。当对 TIMx_DMR 地址进行读取或写入访问时，定时器会识别突发传输，即传输次数（以半个字或字节为单位）

TIMx_DCR 的 DBA[4:0]位定义了 DMA 传输的 DMA 基地址(当通过 TIMx_DMAR 地址完成读写操作)。DBA 定义为从 TIMx_CR1 寄存器地址开始的偏移量。

例子：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

作为一个例子，定时器 DMA 突发特性是用于发生更新事件时更新 CCRx 寄存器中的内容，通过 DMA 传输半字到 CCRx。

通过以下步骤来完成：

1. 如下配置对应的 DMA 通道：

1) DMA 通道外设地址是 DMAR 寄存器的地址
2) DMA 通道存储地址是 RAM 中的 BUFF 地址，包含了 DMA 传给 CCRx 的数据
3) 传输的数据个数=3（见注释）
4) 轮询模式关闭

2. 通过配置 DBA 和 DBL 实现配置 DCR 寄存器：DBL=3，DBA=0XE
3. 使能定时器更新 DMA 请求（UDE=1）
4. 使能定时器
5. 使能 DMA 通道

这此示例适用于每个 CCRx 寄存器更新一次的情况。如果每个 CCRx 都更新两次，数据的传输个数就需要设置为 6。举例假设 RAM 的 buffer 中包含了 data1、data2、data3、data4、data5 和 data6。数

Puya Confidential

据依次传输到 CCRx 中：第一次更新 DMA 请求，data1 传输给 CCR2，data2 传输给 CCR3，data3 传输给 CCR4；第二次更新 DMA 请求，data4 传输给 CCR2，data5 传输给 CCR3，data6 传输给 CCR4。

注：空值可以写入到保留寄存器

19.3.24 TIM1/TIM8 DMA 请求

TIM1/TIM8 可以生成 DMA 请求，如下表所示：

表 19-1 DMA 请求

DMA 请求源	DMA 使能控制位
更新	UDE
比较/捕获 1	CC1DE
比较/捕获 2	CC2DE
比较/捕获 3	CC3DE
比较/捕获 4	CC4DE
COM	COMDE
触发	TDE

19.3.25 定时器同步

所有 TIMx 定时器在内部相连，用于定时器同步或链接。详见第 20.3.20 节：定时器同步。

19.3.26 调试模式

当微控制器进入调试模式时，根据 DBG 模块中 DBG_TIMx_STOP 的设置。

TIMx 计数器可以或者继续正常操作，或者停止。

调试模式下的行为可以使用调试支持模块（DBG）中每个定时器的专用配置进行编程

为了安全的考虑，当计数器停止时，输出被禁止（如果 MOE 被复位了）。输出即可被强制到一个无效电平（OSSI=1），或者由 GPIO 控制器接管（OSSI=0），典型的是强制为高阻。

更多的细节，见 DBG 模块描述

19.3.27 计数器多点触发 ADC

TIMx 定时器在设置 MTADCE 后，可以开启多点触发 ADC 功能。计数器可以工作在向上计数、向下计数和向上向下计数模式。通过 TIMx_MTADCCR 可以选择在一个计数周期内，计数器的零点（CNT 为 0 时）触发、顶点（ARR 值）触发、通道的 CCRx 和 CCRxH 点触发的任意组合，其中 CCRx 和 CCRxH 区分计数器向上计数过程中触发还是向下计数过程触发。满足触发点后在 timx_multi_trg2adc 接口产生一个时钟长度的脉冲信号。

下面的例子为选择了零点、顶点、上升 CCR1 点和下降 CCR4H 点。

假定计数器选择内部时钟源：

- 选择中央对齐模式：根据 TIMx_CR1 寄存器中的 CMS 选择中央对齐模式。
- 配置 TIMx_ARR 自动重装载寄存器。
- 按需配置比较寄存器 TIMx_CCRx[31:0]，置 UG 位产生一个更新事件，更新比较寄存器值。
- 配置触发点 TIMx_CCR1[31:0]和 TIMx_CCR4[31:0]。
- 使能多点触发功能和使能对应触发点，设置 MTADCE，MTZEROE，MTARRE，UMTCCR1E，DMTCCR4HE。
- 置 TIMx_CR1 寄存器中的 CEN=1 使能计数器，如果设置了 UIE 位，则会产生更新中断。

Puya Confidential

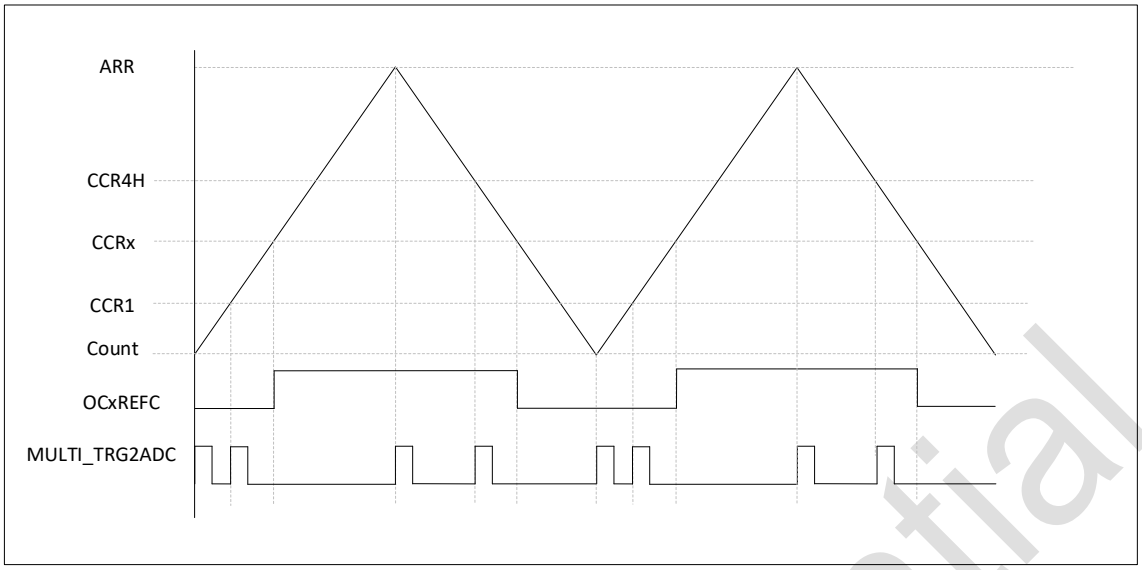


图 19-55 中央对其模式下使能多个触发点

19.4 TIM1/TIM8 中断

中断事件	事件标志	中断使能控制位
更新	UIF	UIE
比较/捕获 1	CC1IF	CC1IE
比较/捕获 2	CC2IF	CC2IE
比较/捕获 3	CC3IF	CC3IE
比较/捕获 4	CC4IF	CC4IE
COM	COM	COMIE
触发	TIF	TIE
刹车	BIF	BIE
系统刹车	SBIF	/

19.5 TIM1/TIM8 寄存器

19.5.1 TIMx 控制寄存器 1（TIMx_CR1）（x=1/8）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:10	Reserved	-	-	保留
9:8	CKD[1:0]	RW	2'h0	时钟分频因子这 2 位定义在定时器时钟 (CK_INT) 频率和死区时间的分频比和死区发生器与数字滤波器（ETR，Tix）所用的采样时钟（t_{DTS}）之间的分频比例。 00：$t_{DTS} = t_{CK_INT}$

Puya Confidential

				01：t_{DTS} = 2 x t_{CK_INT} 10：t_{DTS} = 4 x t_{CK_INT} 11：保留，不要使用这个配置
7	ARPE	RW	0	自动重装载预装载允许位 0：TIMx_ARR 寄存器没有缓冲； 1：TIMx_ARR 寄存器被装入缓冲器。
6:5	CMS	RW	2'h0	选择中央对齐模式 00：边沿对齐模式。计数器依据方向位（DIR）向上或向下计数。 01：中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMRx 寄存器中 CCxS=00）的输出比较中断标志位，只在计数器向下计数时被设置。 10：中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMRx 寄存器中 CCxS=00）的输出比较中断标志位，只在计数器向上计数时被设置。 11：中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMRx 寄存器中 CCxS=00）的输出比较中断标志位，在计数器向上和向下计数时均被设置。 注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	RW	0	计数方向 0：计数器递增计数； 1：计数器递减计数。 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	OPM	RW	0	单脉冲模式 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件(清除 CEN 位)时，计数器停止。
2	URS	RW	0	更新请求源 软件通过该位选择 UEV 事件的源 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： − 计数器溢出/下溢 − 设置 UG 位 − 从模式控制器产生的更新 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0：允许 UEV。更新(UEV)事件由下述任一事件产生： − 计数器溢出/下溢 − 设置 UG 位 − 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。 (译注：更新影子寄存器) 1：禁止 UEV。不产生更新事件，影子寄存器(ARR、 PSC、 CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。

Puya Confidential

0	CEN	RW	0	使能计数器 0：禁止计数器； 1：使能计数器。 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

19.5.2 TIMx 控制寄存器 2（TIMx_CR2）（x=1/8）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	MMS[3]	Res.	MMS2[3:0]				Res.	$\overline{OIS6}$	Res.	$\overline{OIS5}$
-	-	-	-	-	-	RW	-	RW	RW	RW	RW	-	RW	-	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25	MMS[3]	RW	0	详见 MMS[2:0]描述
24	Reserved	-	-	保留
23:20	MMS2[3:0]	RW	4’h0	主模式选择 2 这四位用于选择同步信息（TRGO2）发送给 ADC，可能的组合如 0000：复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO2)。如果触发输入（复位模式下的从模式控制器）产生复位，则 TRGO 上的信号相对实际的复位会有一个延迟。 0001：使能 –计数器使能信号 CNT_EN 可被用于作为触发输出（TRGO）。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式（见 TIMx_SMCR 寄存器中 MSM 位的描述）。 0010：更新– 更新事件被选为触发输出（TRGO2）。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 0011：比较脉冲 – 一旦发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时（即是它已经为高），触发输出送出一个正脉冲（TRGO2）。 0100：比较 – OC1REFC 信号被用于作为触发输出（TRGO2）。 0101：比较 – OC2REFC 信号被用于作为触发输出（TRGO2）。 0110：比较 – OC3REFC 信号被用于作为触发输出（TRGO2）。 0111：比较 – OC4REFC 信号被用于作为触发输出（TRGO2）。 1000：比较 – OC5REFC 信号被用于作为触发输出（TRGO2）。 1001：比较 – OC6REFC 信号被用于作为触发输出（TRGO2）。 1010：比较脉冲 - OC4REFC 信号上升沿或下降沿被用于作为触发输出（TRGO2）。

Puya Confidential

				1011：比较脉冲 – OC6REFC 信号上升沿或下降沿被用于作为触发输出（TRGO2） 1100：比较脉冲 - OC4REFC 信号上升沿或 OC6REFC 信号上升沿用于作为触发输出（TRGO2） 1101：比较脉冲 – OC4REFC 信号上升沿或 OC6REFC 信号下降沿用于作为触发输出（TRGO2） 1110：比较脉冲 – OC5REFC 信号上升沿或 OC6REFC 信号上升沿用于作为触发输出（TRGO2） 1111：比较脉冲 – OC5REFC 信号上升沿或 OC6REFC 信号下降沿用于作为触发输出（TRGO2） 注：从定时器或 ADC 的时钟必须在接收主定时器事件之前就打开，并在接受时不要改变
19	Reserved	-	-	保留
18	OIS6	RW	0	输出空闲状态 6（OC6 输出）。参见 OIS1 位。
17	Reserved	-	-	保留
16	OIS5	RW	0	输出空闲状态 5(OC5 输出)。参见 OIS1 位。
15	OIS4N	RW	0	输出空闲状态 4(OC4N 输出)。参见 OIS1N 位。
14	OIS4	RW	0	输出空闲状态 4(OC4 输出)。参见 OIS1 位。
13	OIS3N	RW	0	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。
12	OIS3	RW	0	输出空闲状态 3(OC3 输出)。参见 OIS1 位。
11	OIS2N	RW	0	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。
10	OIS2	RW	0	输出空闲状态 2(OC2 输出)。参见 OIS1 位。
9	OIS1N	RW	0	输出空闲状态 1(OC1N 输出) 0：当 MOE=0 时，死区后 OC1N=0； 1：当 MOE=0 时，死区后 OC1N=1。 注：已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
8	OIS1	RW	0	输出空闲状态 1(OC1 输出) 0：当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=0； 1：当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=1。 注：已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、 2 或 3 后，该位不能被修改。
7	TI1S	RW	0	TI1 选择 0：TIMx_CH1 引脚连到 TI1 输入； 1：TIMx_CH1、 TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入。
6:4	MMS	RW	3’h0	主模式选择 用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 0000： 复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。

Puya Confidential

				0001： 使能 – 计数器使能信号 CNT_EN 可被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时， TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMx_SMCR 寄存器中 MSM 位的描述)。 0010： 更新 – 更新事件被选为触发输出(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 0011: 比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 0100： 比较 – OC1REFC 信号被用于作为触发输出(TRGO)。 0101： 比较 – OC2REFC 信号被用于作为触发输出(TRGO)。 0110： 比较 – OC3REFC 信号被用于作为触发输出(TRGO)。 0111： 比较 – OC4REFC 信号被用于作为触发输出(TRGO)。 1000：编码器时钟输出-编码器时钟信号作为触发输出（TRGO）。其他的 SMS 值可能导致预期之外的效果。 1001-1111：保留 注意： 1.从定时器和 ADC 的时钟必须先被使能以接收主定时器的信号，并在接收时不要改变。 2.若主从定时器不在同一总线上，主模式应该配置为能被从定时器采到的宽度。
3	CCDS	RW	0	捕获/比较的 DMA 选择 0：当发生 CCx 事件时，送出 CCx 的 DMA 请求； 1：当发生更新事件时，送出 CCx 的 DMA 请求。
2	CCUS	RW	0	捕获/比较控制更新选择 0：如果捕获/比较控制位是预装载的（CCPC=1），只能通过设置 COM 位更新它们； 1：如果捕获/比较控制位是预装载的（CCPC=1），可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。 注：该位只对具有互补输出的通道起作用。
1	Reserved	-	-	保留
0	CCPC	RW	0	捕获/比较预装载控制位 0：CCxE，CCxNE 和 OCxM 位不是预装载的； 1：CCxE，CCxNE 和 OCxM 位是预装载的；设置该位后，它们只在设置了 COM 位后被更新（COMG 位设置或 TIM_TRGI 上检测到的上升沿，具体取决于 CCUS 位）。 注：该位只对具有互补输出的通道起作用。

19.5.3 TIMx 从模式控制寄存器（TIMx_SMCR）（x=1/8）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS[3]	Res.	Res.	Res.	SMS[3]
											RW				RW

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:21	Reserved	-	-	保留
20	TS[3]	RW	1'h0	详见 TS[2:0]描述
19:17	Reserved	-	-	保留
16	SMS[3]	RW	0	详见 SMS[2:0]描述
15	ETP	RW	0	外部触发极性 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0：ETR 不反相，高电平或上升沿有效； 1：ETR 被反相，低电平或下降沿有效。
14	ECE	RW	0	外部时钟使能位 该位启用外部时钟模式 2 0：禁止外部时钟模式 2； 1：使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动。 注 1：设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF（SMS=111 和 TS=00111）具有相同功效。 注 2：下述从模式可以与外部时钟模式 2 同时使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF（TS 位不能是'00111'）。 注 3：外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。
13:12	ETPS	RW	2'h0	外部触发预分频 外部触发信号 ETRP 的频率必须最多是 TIMxCLK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。 00：关闭预分频； 01： ETRP 频率除以 2； 10： ETRP 频率除以 4； 11： ETRP 频率除以 8。
11:8	ETF	RW	4'h0	外部触发滤波 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6

Puya Confidential

				1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8
7	MSM	RW	0	主/从模式 0：无作用； 1：触发输入（TRGI）上的事件被延迟了，以允许在当前定时器（通过 TRGO）与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
6:4	TS	RW	3’h0	触发选择 选择用于同步计数器的触发输入。 0000：内部触发 0（ITR0) 0001：内部触发 1（ITR1) 0010：内部触发 2（ITR2) 0011：内部触发 3（ITR3) 0100：TI1 边沿检测（TI1F_ED） 0101：滤波后的定时器输入 1（TI1FP1） 0110：滤波后的定时器输入 2（TI1FP2） 0111：外部触发输入（ETRF） 1000：内部触发 4(ITR4) 1001：内部触发 5(ITR5) 1010：内部触发 6(ITR6) 1011：内部触发 7(ITR7) 1100：内部触发 8(ITR8) 1101：内部触发 9(ITR9) 1110：内部触发 10(ITR10) 1111：内部触发 11(ITR11) 其它：保留 更多有关 ITRx 的细节，参见外设互联章节 注: 这些位只能在未用到(如 SMS=0000)时被改变，以避免在改变时产生错误的边沿检测。
3	OCCS	RW	0	OCREF 清除选择位 这位选择了 OCREF 的清除源。 0：OCREF_CLR_INT 连接 OCREF_CLR 输入 1：OCREF_CLR_INT 连接 ETRF
2:0	SMS	RW	3’h0	从模式选择 当选择了外部信号，触发信号（TRGI）的有效边沿与选中的外部输入极性相关（见输入控制寄存器和控制寄存器的说明） 0000：关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。 0001：正交编码器模式 1，x2 模式– 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/下计数。

Puya Confidential

				0010：正交编码器模式 2，x2 模式– 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/下计数。 0011：正交编码器模式 3，x4 模式–根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。 0100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 0101：门控模式 – 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 0110：触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位)，只有计数器的启动是受控的。 0111：外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 1000：复位+触发组合模式：-触发输入（TRGI）的上升沿初始化计数器，生成寄存器更新并启动计数器 注：如果 TI1F_ED 被选为触发输入(TS=00100)时，不要使用门控模式。这是因为， TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。 注：必须在接收主定时器事件之前，使能从设备（定时器，ADC）来接收 TRGO，并且时钟频率不能再接收过程中改变 注：在编码器模式下，不要使用 UEV 作为 tim_trgo 输出信号，（即 mms 不能配置为 010）

19.5.4 TIMx DMA/中断使能寄存器（TIMx_DIER）（x=1/8）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:15	Reserved	-	-	保留
14	TDE	RW	0	允许触发 DMA 请求 0：禁止触发 DMA 请求; 1：允许触发 DMA 请求。
13	COMDE	RW	0	允许 COM 的 DMA 请求 0：禁止 COM 的 DMA 请求; 1：允许 COM 的 DMA 请求。
12	CC4DE	RW	0	允许捕获/比较 4 的 DMA 请求 0：禁止捕获/比较 4 的 DMA 请求; 1：允许捕获/比较 4 的 DMA 请求。
11	CC3DE	RW	0	允许捕获/比较 3 的 DMA 请求 0：禁止捕获/比较 3 的 DMA 请求;

Puya Confidential

				1：允许捕获/比较 3 的 DMA 请求。。
10	CC2DE	RW	0	允许捕获/比较 2 的 DMA 请求 0：禁止捕获/比较 2 的 DMA 请求; 1：允许捕获/比较 2 的 DMA 请求。
9	CC1DE	RW	0	允许捕获/比较 1 的 DMA 请求 0：禁止捕获/比较 1 的 DMA 请求; 1：允许捕获/比较 1 的 DMA 请求。。
8	UDE	RW	0	允许更新的 DMA 请求 0：禁止更新的 DMA 请求; 1：允许更新的 DMA 请求。
7	BIE	RW	0	允许刹车中断 0：禁止刹车中断; 1：允许刹车中断。
6	TIE	RW	0	触发中断使能 0：禁止触发中断; 1：使能触发中断。
5	COMIE	RW	0	允许 COM 中断 0：禁止 COM 中断; 1：允许 COM 中断。
4	CC4IE	RW	0	允许捕获/比较 4 中断 0：禁止捕获/比较 4 中断; 1：允许捕获/比较 4 中断。
3	CC3IE	RW	0	允许捕获/比较 3 中断 0：禁止捕获/比较 3 中断; 1：允许捕获/比较 3 中断。
2	CC2IE	RW	0	允许捕获/比较 2 中断 0：禁止捕获/比较 2 中断; 1：允许捕获/比较 2 中断。
1	CC1IE	RW	0	允许捕获/比较 1 中断 0：禁止捕获/比较 1 中断; 1：允许捕获/比较 1 中断
0	UIE	RW	0	允许更新中断 0：禁止更新中断; 1：允许更新中断。

19.5.5 TIMx 状态寄存器（TIMx_SR）（x=1/8）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	IC4IF	IC3IF	IC2IF	IC1IF	IC4IR	IC3IR	Res.	Res.	Res.	Res.	IC2IR	IC1IR	CC6IF	CC5IF
		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0					RC_W0	RC_W0	RC_W0	RC_W0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1F	UIF
		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29	IC4IF	RC_W0	0	下降沿捕获 4 标志 参见 IC1IF 描述。
28	IC3IF	RC_W0	0	下降沿捕获 3 标志 参见 IC1IF 描述。
27	IC2IF	RC_W0	0	下降沿捕获 2 标志 参见 IC1IF 描述。
26	IC1IF	RC_W0	0	下降沿捕获 1 标志 仅当相应的通道被配置为输入捕获时且由下降沿触发捕获事件，该标记可由硬件置 1。它由软件清‘0’或通过读 TIMx_CCR1 清‘0’。 0：无重复捕获产生； 1：发生下降沿捕获事件。
25	IC4IR	RC_W0	0	上升沿捕获 4 标志 参见 IC1IR 描述。
24	IC3IR	RC_W0	0	上升沿捕获 3 标志 参见 IC1IR 描述。
23:20	Reserved	-	-	保留
19	IC2IR	RC_W0	0	上升沿捕获 2 标志 参见 IC1IR 描述。
18	IC1IR	RC_W0	0	上升沿捕获 1 标志 仅当相应的通道被配置为输入捕获时且由上升沿触发捕获事件，该标记可由硬件置 1。它由软件清‘0’或通过读 TIMx_CCR1 清‘0’。 0：无重复捕获产生； 1：发生上升沿捕获事件。
17	CC6IF	RC_W0	0	捕获/比较 6 中断标记 参考 CC1IF 描述。 注：通道 6 只能配置为输出
16	CC5IF	RC_W0	0	捕获/比较 5 中断标记 参考 CC1IF 描述。 注：通道 5 只能配置为输出
15:14	Reserved	-	-	保留
13	SBIF	RC_W0	0	系统刹车中断标志 当系统刹车输入一旦有效，该标志信号会硬件置位。当系统刹车输入无效时可通过软件清零 该标志必须复位来重启 PWM 操作 0：无刹车事件发生 1：系统刹车输入处于有效。如果 TIMx_DIER 寄存器的 BIE 位=1 则会产生中断
12	CC4OF	RC_W0	0	捕获/比较 4 重复捕获标记 参见 CC1OF 描述。
11	CC3OF	RC_W0	0	捕获/比较 3 重复捕获标记 参见 CC1OF 描述。
10	CC2OF	RC_W0	0	捕获/比较 2 重复捕获标记

Puya Confidential

				参见 CC1OF 描述。
9	CC1OF	RC_W0	0	捕获/比较 1 重复捕获标记 仅当相应的通道被配置为输入捕获时，该标记可由硬件置 1。写 0 可清除该位。 0：无重复捕获产生； 1: 计数器的值被捕获到 TIMx_CCR1 寄存器时， CC1IF 的状态已经为‘1’。
8	B2IF	RC_W0	0	刹车 2 中断标记 一旦刹车输入有效，由硬件对该位置‘1’。如果刹车输入无效，则该位可由软件清‘0’。 0：无刹车事件产生； 1：刹车输入上检测到有效电平。如果 TIM_DIER 的 BIE=1 则产生中断
7	BIF	RC_W0	0	刹车中断标记 一旦刹车输入有效，由硬件对该位置‘1’。如果刹车输入无效，则该位可由软件清‘0’。 0：无刹车事件产生； 1：刹车输入上检测到有效电平。如果 TIM_DIER 的 BIE=1 则产生中断
6	TIF	RC_W0	0	触发器中断标记 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿)时由硬件对该位置‘1’。它由软件清‘0’。 0：无触发器事件产生； 1：触发中断等待响应。
5	COMIF	RC_W0	0	COM 中断标记 一旦产生 COM 事件(当捕获/比较控制位： CCxE、 CCxNE、 OCxM 已被更新)该位由硬件置‘1’。它由软件清‘0’。 0：无 COM 事件产生； 1： COM 中断等待响应。
4	CC4IF	RC_W0	0	捕获/比较 4 中断标记 参考 CC1IF 描述。
3	CC3IF	RC_W0	0	捕获/比较 3 中断标记 参考 CC1IF 描述。
2	CC2IF	RC_W0	0	捕获/比较 2 中断标记 参考 CC1IF 描述。
1	CC1IF	RC_W0	0	捕获/比较 1 中断标记 如果通道 CC1 配置为输出模式： 0：无匹配发生； 1： TIMx_CNT 的值与 TIMx_CCR1 的值匹配。 当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下， CC1IF 位变高 如果通道 TI1 配置为输入模式：

Puya Confidential

				当捕获事件发生时该位由硬件置‘1’，它由软件清‘0’或通过读 TIMx_CCR1 清‘0’。 0：无输入捕获产生； 1：计数器值已被捕获（拷贝）至 TIMx_CCR1（在 IC1 上检测到与所选极性相同的边沿）。
0	UIF	RC_W0	0	更新中断标记 当产生更新事件时该位由硬件置‘1’。它由软件清‘0’。 0：无更新事件产生； 1：更新中断等待响应。当寄存器被更新时该位由硬件置’1’： − 若 TIMx_CR1 寄存器的 UDIS=0，当重复计数器数值上溢或下溢时（重复计数器=0 时产生更新事件）。 − 若 TIMx_CR1 寄存器的 URS=0、 UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。 − 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。(参考从模式控制寄存器(TIMx_SMCR))。

19.5.6 TIMx 事件产生寄存器（TIMx_EGR）（x=1/8）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	Reserved
8	B2G	W	0	产生刹车 2 事件 该位由软件置’1’，用于产生一个刹车事件，由硬件自动清’0’。 0：无动作； 1：产生一个刹车 2 事件。此时 MOE=0、 B2IF=1
7	BG	W	0	产生刹车事件 该位由软件置’1’，用于产生一个刹车事件，由硬件自动清’0’。 0：无动作； 1：产生一个刹车事件。此时 MOE=0、 BIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
6	TG	W	0	产生触发事件 该位由软件置’1’，用于产生一个触发事件，由硬件自动清’0’。 0：无动作； 1： TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
5	COMG	W	0	捕获/比较事件，产生控制更新 该位由软件置’1’，由硬件自动清’0’。 0：无动作； 1：当 CCPC=1，允许更新 CCxE、 CCxNE、 OCxM 位。

Puya Confidential

				注： 该位只对拥有互补输出的通道有效。
4	CC4G	W	0	产生捕获/比较 4 事件 参考 CC1G 描述。
3	CC3G	W	0	产生捕获/比较 3 事件 参考 CC1G 描述。
2	CC2G	W	0	产生捕获/比较 2 事件 参考 CC1G 描述。
1	CC1G	W	0	产生捕获/比较 1 事件 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0：无动作； 1：在通道 1 上产生一个捕获/比较事件： 若通道 1 配置为输出： 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 1 配置为输入： 当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	W	0	产生更新事件 该位由软件置'1'，由硬件自动清'0'。 0：无动作； 1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0'；若 DIR=1(向下计数)则计数器取 TIMx_ARR 的值。

19.5.7 TIMx 捕获/比较模式控制寄存器 1（TIMx_CCMR1）（x=1/8）

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式或输出比较模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
							RW								RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	CO2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

■ 输出比较模式

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24	OC2M[3]	RW	0	详见 OC2M[2:0]描述
23:17	Reserved	-	-	保留
16	OC1M[3]	RW	0	详见 OC1M[2:0]描述
15	OC2CE	RW	0	输出比较 2 清零使能 (Output Compare 2 clear enable)
14:12	OC2M	RW	3'h0	输出比较 2 模式 (Output Compare 2 mode)(支持移相模式)
11	OC2PE	RW	0	输出比较 2 预装载使能 (Output Compare 2 preload enable)

Puya Confidential

10	OC2FE	RW	0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	2'h0	捕获/比较 2 选择。 (Capture/Compare 2 selection) 该位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC2 通道被配置为输出； 01：CC2 通道被配置为输入，IC2 映射在 TI2 上； 10：CC2 通道被配置为输入，IC2 映射在 TI1 上； 11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7	OC1CE	RW	0	输出比较 1 清'0'使能 (Output Compare 1 clear enable) 0： OC1REF 不受 ETRF 输入的影响； 1：一旦检测到 ETRF 输入高电平，清除 OC1REF=0。
6:4	OC1M	RW	3'h0	输出比较 1 模式（Output Compare 1 mode）(支持移相模式) 这些位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 0000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用； 0001：匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1(TIMx_CCR1)相同时，强制 OC1REF 为高。 0010：匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1(TIMx_CCR1)相同时，强制 OC1REF 为低。 0011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 0100：强制为无效电平。强制 OC1REF 为低。 0101：强制为有效电平。强制 OC1REF 为高。 0110：PWM 模式 1 移相模式： -在向上计数时，一旦 TIM1_CNT<TIM1_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIM1_CNT>TIM1_CCR1_H 时通道 1 为无效电平，TIM1_CNT<=TIM1_CCR1_H 时通道 1 为有效电平。 非移相模式： - 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平(OC1REF=0)，否则为有效电平(OC1REF=1)。 0111： PWM 模式 2 移相模式：

Puya Confidential

				-在向上计数时，一旦 TIM1_CNT<TIM1_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIM1_CNT>TIM1_CCR1_H 时通道 1 为有效电平，TIM1_CNT<=TIM1_CCR1_H 时通道 1 为无效电平。 非移相模式： - 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。 1000：可重触发 OPM 模式 1-在递增计数模式下，通道处于有效状态，直到检测到触发事件（tim_trgi 信号）。然后，如在 PWM 模式 1 中那样执行比较，并且通道在下一次更新时再次有效。在递减计数模式下，通道处于无效状态，直到检测到触发事件（tim_trgi 信号）。然后，如在 PWM 模式 1 中那样执行比较，并且信道在下一次更新时再次变为无效 1001：可重触发 OPM 模式 2-在递增计数模式下，信道处于无效状态，直到检测到触发事件（tim_trgi 信号）。然后，如在 PWM 模式 2 中那样执行比较，并且通道在下一次更新时再次变为无效。在递减计数模式下，信道处于有效状态，直到检测到触发事件（tim_trgi 信号）。然后，如在 PWM 模式 2 中那样执行比较，并且信道在下一次更新时再次变为有效 注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2：在 PWM 模式中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时， OC1REF 电平才改变。 注 3：当通道有互补输出时，该位段可以预加载。如果 TIMx_CR2 的 CCPC 位置 1，然后仅当 COM 事件生成时 OC1M 有效位从预加载中更新新的值
3	OC1PE	RW	0	输出比较 1 预装载使能 (Output Compare 1 preload enable) 0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。 1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2：仅在单脉冲模式下(TIMx_CR1 寄存器的 OPM=1)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。
2	OC1FE	RW	0	输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快输出对触发输入事件的响应。必须在单脉冲模式（TIMx_CR1 寄存器的 OPM 位置 1）中使用，当触发到来时实现脉冲快速输出。

Puya Confidential

				0：根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。仅能用于 PWM 模式 1 和 PWM 模式 2
1:0	CC1S	RW	2'h0	捕获/比较 1 选择。(Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC1 通道被配置为输出； 01：CC1 通道被配置为输入，IC1 映射在 TI1 上； 10：CC1 通道被配置为输入，IC1 映射在 TI2 上； 11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

■ 输入捕获模式

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:12	IC2F	RW	4'h0	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	2'h0	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	2'h0	捕获/比较 2 选择 (Capture/Compare 2 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC2 通道被配置为输出； 01：CC2 通道被配置为输入，IC2 映射在 TI2 上； 10：CC2 通道被配置为输入，IC2 映射在 TI1 上； 11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注 CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	RW	4'h0	输入捕获 1 滤波器 (Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6

Puya Confidential

				1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8
3:2	IC1PSC	RW	2'h0	输入/捕获 1 预分频器 这 2 位定义了 CC1 输入(IC1)的预分频系数。 一旦 CC1S=00，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01：每 2 个事件触发一次捕获； 10：每 4 个事件触发一次捕获； 11：每 8 个事件触发一次捕获。
1:0	CC1S	RW	2'h0	捕获/比较 1 选择 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC1 通道被配置为输出； 01：CC1 通道被配置为输入，IC1 映射在 TI1 上； 10：CC1 通道被配置为输入，IC1 映射在 TI2 上； 11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

19.5.8 TIMx 捕获/比较模式控制寄存器 2（TIMx_CCMR2）（x=1/8）

偏移地址：0x1C

复位值：0x0000 0000

参见 TIM_CCMR1 的说明。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
							RW								RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	CO4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

■ 输出比较模式

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24	OC4M[3]	RW	0	详见 OC4M[2:0]描述
23:17	Reserved	-	-	保留
16	OC3M[3]	RW	0	详见 OC3M[2:0]描述
15	OC4CE	RW	0	输出比较 4 清零使能 (Output Compare 4 clear enable)
14:12	OC4M	RW	3'h0	输出比较 4 模式 (Output Compare 4 mode) (不支持移相模式)
11	OC4PE	RW	0	输出比较 4 预装载使能 (Output Compare 4 preload enable)

Puya Confidential

10	OC4FE	RW	0	输出比较 4 快速使能 (Output Compare 4 fast enable)
9:8	CC4S	RW	2'h0	捕获/比较 4 选择。 (Capture/Compare 4 selection) 该 2 位定义通道的方向(输入/输出)，及输入脚的选择: 00：CC4 通道被配置为输出; 01：CC4 通道被配置为输入，IC4 映射在 TI4 上; 10：CC4 通道被配置为输入，IC4 映射在 TI3 上; 11：CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7	OC3CE	RW	0	输出比较 3 清'0'使能
6:4	OC3M	RW	3'h0	输出比较 3 模式（Output Compare 3 mode）参考 OC1M
3	OC3PE	RW	0	输出比较 3 预装载使能
2	OC3FE	RW	0	输出比较 3 快速使能（Output Compare 3 fast enable）
1:0	CC3S	RW	2'h0	捕获/比较 3 选择。 这 2 位定义通道的方向（输入/输出），及输入脚的选择: 00：CC3 通道被配置为输出; 01：CC3 通道被配置为输入，IC3 映射在 TI3 上; 10：CC3 通道被配置为输入，IC3 映射在 TI4 上; 11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIMx_CCER 寄存器的 CC3E=0)才是可写的。

■ 输入捕获模式

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:12	IC4F	RW	4'h0	输入捕获 4 滤波器
11:10	IC4PSC	RW	2'h0	输入/捕获 4 预分频器
9:8	CC4S	RW	2'h0	捕获/比较 4 选择 这 2 位定义通道的方向(输入/输出)，及输入脚的选择: 00：CC4 通道被配置为输出; 01：CC4 通道被配置为输入，IC4 映射在 TI4 上; 10：CC4 通道被配置为输入，IC4 映射在 TI3 上; 11：CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7:4	IC3F	RW	4'h0	输入捕获 3 滤波器
3:2	IC3PSC	RW	2'h0	输入/捕获 3 预分频器
1:0	CC3S	RW	2'h0	捕获/比较 3 选择 这 2 位定义通道的方向(输入/输出)，及输入脚的选择: 00：CC3 通道被配置为输出; 01：CC3 通道被配置为输入，IC3 映射在 TI3 上; 10：CC3 通道被配置为输入，IC3 映射在 TI4 上;

Puya Confidential

				11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIMx_CCER 寄存器的 CC3E=0)才是可写的。

19.5.9 TIMx 捕获/比较使能寄存器（TIMx_CCER）（x=1/8）

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										RW	RW			RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:22	Reserved	-	-	保留
21	CC6P	RW	0	输入/捕获 6 输出极性 (Capture/Compare 6 output polarity) 参考 CC1P 的描述。
20	CC6E	RW	0	输入/捕获 6 输出使能 (Capture/Compare 6 output enable) 参考 CC1E 的描述。
19:18	Reserved	-	-	保留
17	CC5P	RW	0	输入/捕获 5 输出极性 (Capture/Compare 5 output polarity) 参考 CC1P 的描述。
16	CC5E	RW	0	输入/捕获 5 输出使能 (Capture/Compare 5 output enable) 参考 CC1E 的描述。
15	CC4NP	RW	0	输入/捕获 4 互补输出极性参考 CC1NP 的描述。
14	CC4NE	RW	0	输入/捕获 4 互补输出使能参考 CC1NE 的描述。
13	CC4P	RW	0	输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	RW	0	输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	RW	0	输入/捕获 3 互补输出极性参考 CC1NP 的描述。
10	CC3NE	RW	0	输入/捕获 3 互补输出使能参考 CC1NE 的描述。
9	CC3P	RW	0	输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	RW	0	输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	RW	0	输入/捕获 2 互补输出极性参考 CC1NP 的描述。
6	CC2NE	RW	0	输入/捕获 2 互补输出使能参考 CC1NE 的描述。
5	CC2P	RW	0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	RW	0	输入/捕获 1 互补输出极性

Puya Confidential

				0：OC1N 高电平有效； 1：OC1N 低电平有效。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 3 或 2 且 CC1S=00(通道配置为输出)则该位不能被修改。
2	CC1NE	RW	0	输入/捕获 1 互补输出使能 0: 关闭 - OC1N 禁止输出,因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。 1：开启 - OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	RW	0	输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出： 0：OC1 高电平有效； 1：OC1 低电平有效。 CC1 通道配置为输入： CC1NP/CC1P 位选择作为触发或捕获信号的 TI1FP1 和 TI2FP1 的极性。 00：不反相/上升沿： TIxFP1 上升沿有效（捕获、复位模式下触发、外部时钟或触发模式下）； TIxFP1 不反相（门控模式、编码器模式）。 01：反相/下降沿： TIxFP1 下降沿有效（捕获、复位模式下触发、外部时钟或触发模式下）； TIxFP1 反相（门控模式、编码器模式）。 10：保留，不要使用这个配置。 11：不反相/双沿 TIxFP1 上升和下降沿都有效（捕获、复位模式下触发、外部时钟或触发模式下）； TIxFP1 不反相（门控模式）。这个配置不能应用于编码器模式下。 注： 1.对于互补输出通道，这一位是预载的。如果 TIMx_CR2 寄存器中的 CCPC 位被设置，那么 CC1P 的实际有效位只有在 com 事件发生时才会加载预载值。 2.一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	RW	0	输入/捕获 1 输出使能 (Capture/Compare 1 output enable) CC1 通道配置为输出： 0： 关闭 - OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、 OSSI、OSSR、 OIS1、 OIS1N 和 CC1NE 位的值。 1： 开启 - OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、 OSSR、OIS1、 OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。 0：捕获禁止； 1：捕获使能。 注：

Puya Confidential

				对于互补输出通道，这一位是预载的。如果 TIMx_CR2 寄存器中的 CCPC 位被设置，那么 CC1E 的实际有效位只有在 com 事件发生时才会加载预载值。

表 19-2 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

控制位					输出状态	
MOE 位	**OSSI 位**	**OSSR 位**	**CCxE 位**	**CCxNE 位**	**OCx输出状态**	**OCxN输出状态**
1	x	0	0	0	输出禁止（与定时器断开） OCx=0, OCx_EN=0	输出禁止（与定时器断开） OCxN=0, OCxN_EN=0
		0	0	1	输出禁止（与定时器断开） OCx=0, OCx_EN=0	OCxREF+极性， OCxN=OCREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF+极性， OCx=OCREF xor CCxP, OCx_EN=1	输出禁止（与定时器断开） OCxN=0, OCxN_EN=0
		0	1	1	OCxREF+极性+死区，， OCx_EN=1	OCxREF（取反）+极性+死区， OCxN_EN=1
		1	0	0	输出禁止（与定时器断开） OCx=CCxP, OCx=0 OCx_EN=0	输出禁止（与定时器断开） OCxN=CCxNP, OCx=0 OCxN_EN=0
		1	0	1	关闭状态(输出使能且为无效电平) OCx=CCxP, OCx_EN=1	OCxREF+极性， OCxN=OCREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF+极性， OCx=OCREF xor CCxP, OCx_EN=1	关闭状态（输出使能且为无效电平） OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCxREF+极性+死区， OCx_EN=1	OCxREF（取反）+极性+死区， OCxN_EN=1
0	0	x	x	x	输出禁止（与定时器断开）	
	1		0	0		
	1		0	1	关闭状态（输出使能但为无效电平） 异步的：OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCx_EN=1;（如果刹车或刹车2触发了） 然后若时钟存在（仅刹车1触发时有效，不需要刹车2）：经过一个死区时间后OCx=OISx, OCxN=OISxN，假设OISx与OISxN并不都对应OCx和OCxN的有效电平 注：刹车2仅用于OSSI=OSSR=1	
	1		1	0		
	1	x	1	1		

特别的，当死区有效时，输出总是按照死区内输出的规则输出，尽管此时 moe 可能已经被软件或硬件的方式置为 1。

如果一个通道的 2 个输出都没有使用(CCxE = CCxNE = 0)，那么 OISx, OISxN, CCxP 和 CCxNP 都必须清零。

19.5.10 TIMx 计数器（TIMx_CNT）（x=1/8）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CNT[15:0]	RW	16'h0	计数器的值

19.5.11 TIMx 预分频器（TIMx_PSC）（x=1/8）

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	PSC[15:0]	RW	16'h0	预分频器的值 计数器的时钟频率(CK_CNT)等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被 TIM_EGR 的 UG 位清'0'或被工作在复位模式的从控制器清'0'。

19.5.12 TIMx 自动重装载寄存器（TIMx_ARR）（x=1/8）

偏移地址：0x2C

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:16]			
												RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:20	Reserved	-	-	保留
19:0	ARR[19:0]	RW	20'hFFFF	自动重装载的值 (Prescaler value) 自动重装载的值为空时，计数器不工作。 该寄存器保持自动加载值

19.5.13 TIMx 重复计数寄存器（TIMx_RCR）（x=1/8）

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REP[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留

Puya Confidential

15:0	REP[15:0]	RW	16'h0	重复计数器的值 (Repetition counter value) 开启了预装载功能后，这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。 每次向下计数器达到 0，会产生一个更新事件并且重复计数器重新从 REP 值开始计数。对 TIMx_RCR 寄存器写入的新值只在下次更新事件发生时才起作用。 这意味着在 PWM 模式中，(REP+1)对应着： - 在边沿对齐模式下，PWM 周期的数目； - 在中心对称模式下，PWM 半周期的数目；

19.5.14 TIMx 捕获/比较寄存器 1（TIMx_CCR1）（x=1/8）

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1_H[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR1[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	CCR1_H[31:16]	RW	0	输出比较 1 寄存器的值 若 CC1 通道配置为中央对齐 pwm 模式输出： CCR1_H 包含了装入向下计数比较 1 寄存器的值 (预装载值) 。
15:0	CCR1[15:0]	RW	16'h0	捕获/比较通道 1 的值 (Capture/Compare 1 value) 若 CC1 通道配置为输出： CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIMx_CCMR1 寄存器(OC1PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。 若 CC1 通道配置为输入： CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。 TIMx_CCR1 寄存器只读

19.5.15 TIMx 捕获/比较寄存器 2（TIMx_CCR2）（x=1/8）

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2_H[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR2[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	CCR2_H[31:16]	RW	0	输出比较 2 寄存器的值 若 CC2 通道配置为中央对齐 pwm 模式输出：

Puya Confidential

				CCR2_H 包含了装入向下计数比较 2 寄存器的值 (预装载值)。
15:0	CCR2[15:0]	RW	16'h0	捕获/比较通道 2 的值 (Capture/Compare 2 value) 若 CC2 通道配置为输出: CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIMx_CCMR1 寄存器(OC2PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。 若 CC2 通道配置为输入: CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。 TIMx_CCR2 寄存器只读

19.5.16 TIMx 捕获/比较寄存器 3（TIMx_CCR3）（x=1/8）

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3_H[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR3[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	CCR3_H[31:16]	RW	0	输出比较 3 寄存器的值 若 CC3 通道配置为中央对齐 pwm 模式输出: CCR3_H 包含了装入向下计数比较 3 寄存器的值（预装载值）。
15:0	CCR3[15:0]	RW	16'h0	捕获/比较通道 3 的值 (Capture/Compare 3 value) 若 CC3 通道配置为输出: CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。 如果在 TIMx_CCMR2 寄存器(OC3PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC3 端口上产生输出信号。 若 CC3 通道配置为输入: CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。 TIMx_CCR3 寄存器只读

19.5.17 TIMx 捕获/比较寄存器 4（TIMx_CCR4）（x=1/8）

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4_H[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR4[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:16	CCR4_H[31:16]	RW	0	输出比较 4 寄存器的值 若 CC4 通道配置为中央对齐 PWM 模式输出： CCR4_H 包含了装入向下计数比较 4 寄存器的值（预装载值）。
15:0	CCR4[15:0]	RW	16'h0	捕获/比较通道 4 的值 (Capture/Compare 4 value) 若 CC4 通道配置为输出： CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。 如果在 TIMx_CCMR2 寄存器(OC4PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 4 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC4 端口上产生输出信号。 若 CC4 通道配置为输入： CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。 TIMx_CCR4 寄存器只读

19.5.18 TIMx 刹车和死区寄存器（TIMx_BDTR）（x=1/8）

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F [3:0]				BKF [3:0]			
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

注释：根据锁定设置，BKBID/BK2BID/BK2P/BK2E/BK2F/BKF/AOE/BKP/BKE/OSSI、OSSR 和 DTG[7:0]位均可被写保护，有必要在第一次写入 TIMx_BDTR 寄存器时对它们进行配置。

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	Reserved
29	BK2BID	RW	0	刹车 2 双向性（Break2 bidirectional） 参考 BKBID 描述
28	BKBID	RW	0	刹车双向性（Break bidirectional） 0：刹车输入 BRK 在输入模式中 1：刹车输入 BRK 在双向模式中 在双向模式中（BKBID 设置为 1），刹车输入被既配置为输入模式也是开漏输出模式。任何有效的刹车事件会将刹车输入电平拉低，向外部设备指示内部发生了刹车事件 注：一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。 注：任何对该位的写操作，都会延迟一个 APB 时钟生效。
27	BK2DSRM	RW	0	刹车 2 解除（Break2 disarm） 参考 BKDSRM 描述
26	BKDSRM	RW	0	刹车解除（Break disarm） 0：刹车输入 BRK 启用

Puya Confidential

				1：刹车输入 BRK 解除 当没有刹车源时该位会被硬件清零 BKDSRM 位必须由软件置位来释放双向输出控制（开漏输出高阻状态），然后轮询它直到被硬件复位，表示故障状态已经消失。 注：任何对该位的写操作，都会延迟一个 APB 时钟生效。
25	BK2P	RW	0	刹车 2 极性（Break 2 polarity） 0：刹车输入 BRK2 有效电平为低电平 1：刹车输入 BRK2 有效电平为高电平 注：一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。 注：任何对该位的写操作，都会延迟一个 APB 时钟生效。
24	BK2E	RW	0	刹车 2 使能（Break 2 enable） 该位使能刹车 2 的保护 0：刹车 2 功能关闭 1：刹车功能开启 注：刹车 2 必须在 OSSR=OSSI=1 时才能使用 注：一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。 注：任何对该位的写操作，都会延迟一个 APB 时钟生效。
23:20	BK2F[3:0]	RW	4’h0	刹车 2 滤波（Break 2 filter） 这些位定义了对 BRK2 信号采样的频率和对 BRK2 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，BRK2 是异步的 0001：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8

Puya Confidential

				注：一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。
19:16	BKF[3:0]	RW	4'h0	刹车滤波（Break filter） 这些位定义了对 BRK 信号采样的频率和对 BRK 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，BRK2 是异步的 0001：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，=6 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8 注：一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。
15	MOE	RW	0	主输出使能 (Main output enable) 一旦刹车输入有效，该位被硬件异步清'0'。根据 AOE 位的设置值，该位可以由软件清'0'或被自动置 1。它仅对配置为输出的通道有效。 0：禁止 OC 和 OCN 输出或强制为空闲状态； 1：如果设置了相应的使能位(TIMx_CCER 寄存器的 CCxE、CCxNE 位)，则开启 OC 和 OCN 输出。 有关 OC/OCN 使能的细节，参见 TIMx 捕获/比较使能寄存(TIMx_CCER)。
14	AOE	RW	0	自动输出使能 0：MOE 只能被软件置'1'； 1：MOE 能被软件置'1'或在下一个更新事件被自动置'1'(如果刹车输入无效)。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。
13	BKP	RW	0	刹车输入极性 0：刹车输入低电平有效； 1：刹车输入高电平有效。

Puya Confidential

				注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。 注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
12	BKE	RW	0	刹车功能使能 0：禁止刹车输入(BRK 及 CCS 时钟失效事件)； 1：开启刹车输入(BRK 及 CCS 时钟失效事件)。 注：当设置了 LOCK 级别 1 时(TIMx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。 注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
11	OSSR	RW	0	运行模式下"关闭状态"选择 该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。 参考 OC/OCN 使能的详细说明(TIMx 捕获/比较使能寄存器(TIMx_CCER))。 0：当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，首先开启 OC/OCN 并输出无效电平，然后置 OC/OCN 使能输出信号=1。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
10	OSSI	RW	0	空闲模式下"关闭状态"选择 (Off-state selection for Idle mode) 该位用于当 MOE=0 且通道设为输出时。 参考 OC/OCN 使能的详细说明(TIMx 捕获/比较使能寄存器(TIMx_CCER))。 0：当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1， OC/OCN 首先输出其空闲电平，然后 OC/OCN 使能输出信号=1。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
9:8	LOCK	RW	2'h0	锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。 00：锁定关闭，寄存器无写保护； 01：锁定级别 1，不能写入 TIMx_BDTR 寄存器的 DTG、BKBID、BK2BID、BKE、 BKP、 AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位； 10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出， CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCxNP 位)以及 OSSR/OSSI 位； 11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出， CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位)； 注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIMx_BDTR 寄存器，则其内容冻结直至复位。
7:0	DTG	RW	8'h0	死区发生器设置 (Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间:

Puya Confidential

				DTG[7:5]=0xx => DT=DTG[7:0] × Tdtg， Tdtg = TDTS； DTG[7:5]=10x => DT=(64+DTG[5:0]) × Tdtg， Tdtg = 2 × TDTS； DTG[7:5]=110 => DT=(32+DTG[4:0]) × Tdtg， Tdtg = 8 × TDTS； DTG[7:5]=111 => DT=(32+DTG[4:0])× Tdtg， Tdtg = 16 × TDTS； 例：若 TDTS = 125ns(8MHZ)，可能的死区时间为： 0 到 15875ns，若步长时间为 125ns； 16us 到 31750ns，若步长时间为 250ns； 32us 到 63us，若步长时间为 1us； 64us 到 126us，若步长时间为 2us； 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 1、 2 或 3，则不能修改这些位。

19.5.19 TIMx 捕获/比较寄存器 5（TIMx_CCR5）（x=1/8）

偏移地址：0x48

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR5[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CCR5	RW	16’h0	捕获/比较通道 5 的值(Capture/Compare 5 value) 若 CC5 通道配置为输出： CCR5 包含了装入当前捕获/比较 5 寄存器的值(预装载值)。 如果在 TIMx_CCMR3 寄存器(OC5PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 5 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC5 端口上产生输出信号。 该寄存器保持 CCR5[15:0]的比较值，CCR5[19:16]位被复位

19.5.20 TIMx 捕获/比较寄存器 6（TIMx_CCR6）（x=1/8）

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR6[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CCR6	RW	16’h0	捕获/比较通道 6 的值(Capture/Compare 6 value) 若 CC6 通道配置为输出： CCR6 包含了装入当前捕获/比较 6 寄存器的值(预装载值)。

Puya Confidential

				如果在 TIMx_CCMR3 寄存器(OC6PE 位)中未选择预装载功能,写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时,此预装载值才传输至当前捕获/比较 6 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较,并在 OC6 端口上产生输出信号。

19.5.21 TIMx 捕获/比较模式控制寄存器 3（TIMx_CCMR3）（x=1/8）

偏移地址：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							RW								RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6CE	OC6M[2:0]			OC6PE	CO6FE	Res.	Res.	OC5CE	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	RW		

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24	OC6M[3]	RW	0	详见 OC6M[2:0]描述
23:17	Reserved	-	-	保留
16	OC5M[3]	RW	0	详见 OC5M[2:0]描述
15	OC6CE	RW	0	输出比较 6 清零使能（Output Compare 6 clear enable)
14:12	OC6M	RW	3'h0	输出比较 6 模式 (Output Compare 6 mode) 参照 OC5M[3:0]描述
11	OC6PE	RW	0	输出比较 6 预装载使能 (Output Compare 6 preload enable)
10	OC6FE	RW	0	输出比较 6 快速使能 (Output Compare 6 fast enable)
9:8	Reserved	-	-	保留
7	OC5CE	RW	0	输出比较 5 清'0'使能 (Output Compare 5 clear enable)
6:4	OC5M	RW	3'h0	输出比较 5 模式(OutputCompare5mode) 这些位定义了输出参考信号 OC5REF 的动作，而 OC5REF 决定了 OC5、OC5N 的值。OC5REF 是高电平有效，而 OC5、OC5N 的有效电平取决于 CC5P、CC5NP 位。 0000：冻结。输出比较寄存器 TIMx_CCR5 与计数器 TIMx_CNT 间的比较对 OC5REF 不起作用; 0001：匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1(TIMx_CCR5)相同时，强制 OC5REF 为高。 0010：匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1(TIMx_CCR5)相同时，强制 OC5REF 为低。 0011：翻转。当 TIMx_CCR5=TIMx_CNT 时，翻转 OC5REF 的电平。 0100：强制为无效电平。强制 OC5REF 为低。 0101：强制为有效电平。强制 OC5REF 为高。 0110：PWM 模式 1 - 在向上计数时，一旦 TIMx_CNT<TIMx_CCR5 时通道 5 为有效电平，否则为无效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR5 时通道 5 为无效电平(OC5REF=0)，否则为有效电平(OC5REF=1)。

Puya Confidential

				0111：PWM 模式 2 - 在向上计数时，一旦 TIMx_CNT<TIMx_CCR5 时通道 5 为无效电平，否则为有效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR5 时通道 5 为有效电平，否则为无效电平。 1000：可重触发 OPM 模式 1-在递增计数模式下，通道处于有效状态，直到检测到触发事件（TRGI 信号）。然后，如在 PWM 模式 1 中那样执行比较，并且通道在下一次更新时再次有效。在递减计数模式下，通道处于无效状态，直到检测到触发事件（TRGI 信号）。然后，如在 PWM 模式 1 中那样执行比较，并且信道在下一次更新时再次变为不活动 1001：可重触发 OPM 模式 2-在递增计数模式下，信道处于无效状态，直到检测到触发事件（TRGI 信号）。然后，如在 PWM 模式 2 中那样执行比较，并且通道在下一次更新时再次变为无效。在递减计数模式下，信道处于有效状态，直到检测到触发事件（TRGI 信号）。然后，如在 PWM 模式 2 中那样执行比较，并且信道在下一次更新时再次变为有效 其它：保留 注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2：在 PWM 模式中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时， OC5REF 电平才改变。 注 3: 当通道有互补输出时,该位段可以预加载。如果 TIMx_CR2 的 CCPC 位置 1，然后仅当 COM 事件生成时 OC5M 有效位从预加载中更新新的值 注 4：使用可重复 OPM 模式时，计数模式不要配置为中央计数模式
3	OC5PE	RW	0	输出比较 5 预装载使能（Output Compare 5 preload enable)
2	OC5FE	RW	0	输出比较 5 快速使能（Output Compare 5 fast enable)
1:0	Reserved	-	-	保留

19.5.22 TIMx 死区时间寄存器 2（TIMx_DTR2）（x=1/8）

Address offset:0x54

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DTPE	DTAE
														RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	DTGF[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:18	Reserved	-	-	Reserved
17	DTPE	RW	0	死区时间预加载使能（Deadtime preload enable） 0：死区时间的值不可以预加载 1：死区时间的值可以预加载 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，2，3，则该位不能被修改。
16	DTAE	RW	0	死区时间不对称使能（Deadtime asymmetric enable） 0：上升沿和下降沿的死区时间相同，都由 DTG[7:0]配置 1：上升沿死区时间由 DTG[7:0]配置，下降沿死区时间由 DTGF[7:0]配置 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，2，3，则该位不能被修改。
15:8	Rserved	-	-	Reserved

Puya Confidential

7:0	DTGF [7:0]	RW	8'h0	下降沿死区时间生成器（Dead-time falling edge generator setup） 该位段定义了插入互补输出时下降沿的死区时间，假设 DT 表示其持续时间： DTGF[7:5]=0xx => DTF=DTGF[7:0] × Tdtg， Tdtg = T_{DTS}； DTGF[7:5]=10x => DTF=(64+DTGF[5:0]) × Tdtg， Tdtg = 2 × TDTS； DTGF[7:5]=110 => DTF=(32+DTGF[4:0]) × Tdtg， Tdtg = 8 × TDTS； DTGF[7:5]=111 => DTF=(32+DTGF[4:0])× Tdtg， Tdtg = 16 × TDTS； 例：若 TDTS = 125ns(8MHZ)，可能的死区时间为： 0 到 15875ns，若步长时间为 125ns； 16us 到 31750ns，若步长时间为 250ns； 32us 到 63us，若步长时间为 1us； 64us 到 126us，若步长时间为 2us； 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 1、 2 或 3，则不能修改这些位

19.5.23 TIMx 输入选择寄存器（TIMx_TISEL）（x=1/8）

偏移地址：0x5C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				RW	RW	RW	RW					RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				RW	RW	RW	RW					RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:28	Reserved	-	-	保留
27:24	TI4SEL	RW	4'h0	TI4 输入选择. 0000：TIM1_CH4 0001：COMP1_OUT 0010：COMP2_OUT 0011：COMP3_OUT 0100：COMP4_OUT 其他：保留
23:20	Reserved	-	-	保留
19:16	TI3SEL	RW	4'h0	TI3 输入选择. 0000：TIM1_CH3 0001：COMP1_OUT 0010：COMP2_OUT 0011：COMP3_OUT 0100：COMP4_OUT 其他：保留
15:12	Reserved	-	-	保留
11:8	TI2SEL	RW	4'h0	TI3 输入选择. 0000：TIM1_CH3 0001：COMP1_OUT 0010：COMP2_OUT 0011：COMP3_OUT 0100：COMP4_OUT 其他：保留

Puya Confidential

7:4	Reserved	-	-	保留
3:0	TI1SEL	RW	4'h0	TI1 输入选择. 0000：TIM1_CH1 0001：COMP1_OUT 0010：COMP2_OUT 0011：COMP3_OUT 0100：COMP4_OUT 0101：MCO1_CLK(TIM1)/ MCO2_CLK(TIM8) 其他：保留

19.5.24 TIMx 备用功能选项寄存器 1（TIMx_AF1）（x=1/8）

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWM_PHS_EN	INTR_SEL		ETRSEL[3:2]	
											RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ETRSEL[1:0]		BKCMP4P	BKCMP3P	BKCMP2P	BKCMP1P	BKINP	Res.	Res.	Res.	Res.	BKCMP4E	BKCMP3E	BKCMP2E	BKCMP1E	BKINE
RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:21	Reserved	-	-	Reserved
20	PWM_PHS_EN	RW	0	PWM 移相使能 0：关 1：开
19:18	INTR_SEL	RW	0	00：上溢或下溢中断 01：上溢中断 10：下溢中断 11：保留 注：不使用移相功能时，此位配置为 0
17:14	ETRSEL	RW	4'h0	外部触发源选择（etr_in source selection） 该位段用于选择 ETR 输入源 0000：TIMx_ETR 0001：tim_etr1 0010：tim_ etr2 0011：tim_ etr 3 0100：tim_ etr 4 0101：tim_ etr 5 0110：tim_ etr 6 0111：tim_ etr 7 1000：tim_ etr 8 1001：tim_ etr 9 1010：tim_ etr 10 1011：tim_ etr 11 1100：tim_ etr 12 1101：tim_ etr 13

Puya Confidential

				1110：tim_ etr 14 1111：tim_ etr 15 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
13	BKCMP4P	RW	0	tim_brk_cmp4 输入极性 该位选择 tim_brk_cmp4 输入极性，必须与 BKP 极性位同时配置 0：tim_brk_cmp4 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：tim_brk_cmp4 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
12	BKCMP3P	RW	0	tim_brk_cmp3 输入极性 该位选择 tim_brk_cmp3 输入极性，必须与 BKP 极性位同时配置 0：tim_brk_cmp3 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：tim_brk_cmp3 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
11	BKCMP2P	RW	0	tim_brk_cmp2 输入极性 该位选择 brk_cmp2 输入极性，必须与 BKP 极性位同时配置 0：tim_brk_cmp2 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：tim_brk_cmp2 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
10	BKCMP1P	RW	0	tim_brk_cmp1 输入极性 该位选择 brk_cmp1 输入极性，必须与 BKP 极性位同时配置 0：tim_brk_cmp1 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：tim_brk_cmp1 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
9	BKINP	RW	0	BKIN 输入极性（TIMx_BKIN input polarity） 该位选择 BKIN 输入极性的备用功能，必须与 BKP 极性位同时配置 0：BKIN 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：BKIN 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
8:5	Reserved	-	-	Reserved
4	BKCMP4E	RW	0	tim_brk_cmp4 输入使能（tim_brk_cmp4 enable）

Puya Confidential

				该位用于刹车输入时使能 tim_brk_cmp4。tim_brk_cmp4 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk_cmp4 输入关闭 1：tim_brk_cmp4 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
3	BKCMP3E	RW	0	tim_brk_cmp3 输入使能（tim_brk_cmp3 enable） 该位用于刹车输入时使能 tim_brk_cmp3。tim_brk_cmp3 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk_cmp3 输入关闭 1：tim_brk_cmp3 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
2	BKCMP2E	RW	0	tim_brk_cmp2 输入使能（tim_brk_cmp2 enable） 该位用于刹车输入时使能 tim_brk_cmp2。tim_brk_cmp2 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk_cmp2 输入关闭 1：tim_brk_cmp2 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
1	BKCMP1E	RW	0	tim_brk_cmp1 输入使能（tim_brk_cmp1 enable） 该位用于刹车输入时使能 tim_brk_cmp1。tim_brk_cmp1 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk_cmp1 输入关闭 1：tim_brk_cmp1 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
0	BKINE	RW	1	BKIN 输入使能（TIMx_BKIN input enable） 该位用于刹车输入时使能 BKIN 的备用功能。BKIN 输入与其它刹车源进行或逻辑作为刹车输入。 0：BKIN 输入关闭 1：BKIN 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改。

19.5.25 TIMx 备用功能选项寄存器 2（TIMx_AF2）（x=1/8）

偏移地址：0x64

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCRSEL[2:0]		
													RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	BK2 CMP4 P	BK2 CMP3 P	BK2 CMP2 P	BK2 CMP1 P	BK2INP	Res.	Res.	Res.	Res.	BK2 CMP4 E	BK2 CMP3 E	BK2 CMP2 E	BK CMP1 E	BK2INE
		RW	RW	RW	RW	RW					RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:19	Reserved	-	-	Reserved
18:16	OCRSEL	RW	3'h0	OCREF 复位源选择（ocref_clr source selection） 该位段用于选择 ocref_clr 输入源 0000：ocref_clr0 0001：ocref_clr1 0010：ocref_clr2 0011：ocref_clr3 其它：保留 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
15:14	Reserved	-	-	Reserved
13	BK2CMP4P	RW	0	tim_brk2_cmp4 输入极性 该位选择 tim_brk2_cmp4 输入极性，必须与 BK2P 极性位同时配置 0：tim_brk2_cmp4 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：tim_brk2_cmp4 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
12	BK2CMP3P	RW	0	tim_brk2_cmp3 输入极性 该位选择 tim_brk2_cmp3 输入极性，必须与 BK2P 极性位同时配置 0：tim_brk2_cmp3 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：tim_brk2_cmp3 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
11	BK2CMP2P	RW	0	tim_brk2_cmp2 输入极性 该位选择 tim_brk2_cmp2 输入极性，必须与 BK2P 极性位同时配置 0：tim_brk2_cmp2 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：tim_brk2_cmp2 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
10	BK2CMP1P	RW	0	tim_brk2_cmp1 输入极性 该位选择 tim_brk2_cmp1 输入极性，必须与 BK2P 极性位同时配置 0：tim_brk2_cmp1 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：tim_brk2_cmp1 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
9	BK2INP	RW	0	BKIN2 输入极性（TIMx_BKIN2 input polarity） 该位选择 BKIN 输入极性的备用功能，必须与 BK2P 极性位同时配置

Puya Confidential

				0：BKIN2 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：BKIN2 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
8:5	Reserved	-	-	Reserved
4	BK2CMP4E	RW	0	tim_brk2_cmp4 输入使能（tim_brk2_cmp4 enable） 该位用于刹车输入时使能 tim_brk2_cmp4。tim_brk2_cmp4 输入与其它刹车源进行或逻辑作为刹车输入。 0：brk2_cmp4 输入关闭 1：brk2_cmp4 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
3	BK2CMP3E	RW	0	tim_brk2_cmp3 输入使能（tim_brk2_cmp3 enable） 该位用于刹车输入时使能 tim_brk2_cmp3。tim_brk2_cmp3 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk2_cmp3 输入关闭 1：tim_brk2_cmp3 输入开启 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
2	BK2CMP2E	RW	0	tim_brk2_cmp2 输入使能（tim_brk2_cmp2 enable） 该位用于刹车输入时使能 tim_brk2_cmp2。tim_brk2_cmp2 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk2_cmp2 输入关闭 1：tim_brk2_cmp2 输入开启 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
1	BK2CMP1E	RW	0	tim_brk2_cmp1 输入使能（tim_brk2_cmp1 enable） 该位用于刹车输入时使能 tim_brk2_cmp1。tim_brk2_cmp1 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk2_cmp1 输入关闭 1：tim_brk2_cmp1 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
0	BK2INE	RW	1	BKIN2 输入使能（TIMx_BKIN input enable） 该位用于刹车输入时使能 BKIN2 的备用功能。BKIN2 输入与其它刹车源进行或逻辑作为刹车输入。 0：BKIN2 输入关闭 1：BKIN2 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改。

19.5.26 TIMx 多点触发 ADC 控制寄存器（TIMx_MTADCCR）（x=1/8）

偏移地址：0x6C

复位值：0x0000 0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MTADCE	DMTCCR4HE	DMTCCR4E
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMTCCR3HE	DMTCCR3H	DMTCCR2HE	DMTCCR2H	DMTCCR1HE	DMTCCR1H	UMTCCR4HE	UMTCCR4E	UMTCCR3HE	UMTCCR3H	UMTCCR2HE	UMTCCR2H	UMTCCR1HE	UMTCCR1H	MTARRE	MTZEROE
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:19	Reserved	-	-	保留
18	MTADCE	RW	1'b0	组合触发 ADC 总使能（multi triger adc en） 1：使能组合触发 adc 功能 0：禁用组合触发 adc 功能
17	DMTCCR4HE	RW	1'b0	向下计数，CNT 为 CCR4H 时触发 ADC 使能（count down multi triger CCR4H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
16	DMTCCR4E	RW	1'b0	向下计数，CNT 为 CCR4 时触发 ADC 使能（count down multi triger CCR4 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
15	DMTCCR3HE	RW	1'b0	向下计数，CNT 为 CCR3H 时触发 ADC 使能（count down multi triger CCR3H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
14	DMTCCR3E	RW	1'b0	向下计数，CNT 为 CCR3 时触发 ADC 使能（count down multi triger CCR3 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
13	DMTCCR2HE	RW	1'b0	向下计数，CNT 为 CCR2H 时触发 ADC 使能（count down multi triger CCR1H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
12	DMTCCR2E	RW	1'b0	向下计数，CNT 为 CCR2 时触发 ADC 使能（count down multi triger CCR1 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
11	DMTCCR1HE	RW	1'b0	向下计数，CNT 为 CCR1H 时触发 ADC 使能（count down multi triger CCR1H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
10	DMTCCR1E	RW	1'b0	向下计数，CNT 为 CCR1 时触发 ADC 使能（count down multi triger CCR1 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
9	UMTCCR4HE	RW	1'b0	向上计数，CNT 为 CCR4H 时触发 ADC 使能（count up multi triger CCR4H enable），CEN 和 MTADCE 同时为 1 有效。

Puya Confidential

				1：使能该触发点。 0：禁用该出发点。
8	UMTCCR4E	RW	1'b0	向上计数，CNT 为 CCR4 时触发 ADC 使能（count up multi triger CCR4 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
7	UMTCCR3HE	RW	1'b0	向上计数，CNT 为 CCR3H 时触发 ADC 使能（count up multi triger CCR3H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
6	UMTCCR3E	RW	1'b0	向上计数，CNT 为 CCR3 时触发 ADC 使能（count up multi triger CCR3 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
5	UMTCCR2HE	RW	1'b0	向上计数，CNT 为 CCR2H 时触发 ADC 使能（count up multi triger CCR2H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
4	UMTCCR2E	RW	1'b0	向上计数，CNT 为 CCR2 时触发 ADC 使能（count up multi triger CCR2 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
3	UMTCCR1HE	RW	1'b0	向上计数，CNT 为 CCR1H 时触发 ADC 使能（count up multi triger CCR1H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
2	UMTCCR1E	RW	1'b0	向上计数，CNT 为 CCR1 时触发 ADC 使能（count up multi triger CCR1 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
1	MTARRE	RW	1'b0	CNT 为 ARR 时触发 ADC 使能（multi triger ARR enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
0	MTZEROE	RW	1'b0	CNT 为 0 时触发 ADC 使能（multi triger zero enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。

19.5.27 TIMx DMA 控制寄存器（TIMx_DCR）（x=1/8）

偏移地址：0x3DC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			RW	RW	RW	RW	RW				RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:13	Reserved	-	-	保留
12:8	DBL	RW	5'h0	DMA 连续传送长度（DMA burst length) 这些位定义了 DMA 在连续模式下的传送长度(当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送)，即：定义传输的次数，传输可以是半字(双字节)或字节： 00000：1 次传输 00001：2 次传输 00010：3 次传输 .... 10001：24 次传输
7:5	Reserved	-	-	保留
4:0	DBA	RW	5'h0	DMA 基地址（DMA base address） 这些位定义了 DMA 在连续模式下的基地址(当对 TIMx_DMAR 寄存器进行读或写时)， DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量： 00000： TIMx_CR1， 00001： TIMx_CR2， 00010： TIMx_SMCR， ……..

19.5.28 TIMx 连续模式的 DMA 地址（TIMx_DMAR）（x=1/8）

偏移地址：0x3E0

复位值：0x0000 000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAB[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	DMAB	RW	0	DMA 连续传送寄存器 对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作: TIMx_CR1 地址 + (DBA + DMA 索引)x4 其中： "TIMx_CR1 地址"是控制寄存器 1(TIMx_CR1)所在的地址； "DBA"是 TIMx_DCR 寄存器中定义的基地址； "DMA 索引"是由 DMA 自动控制的偏移量，它取决于 TIMx_DCR 寄存器中定义的 DBL。

Puya Confidential

20. 通用定时器（TIM2/TIM3）

20.1 TIM2/TIM3 简介

通用控制定时器（TIMx）由一个 32 位或 16 位的自动装载计数器组成，计数器由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度（输入捕获），或者产生输出波形（输出比较、PWM、嵌入死区时间的互补 PWM）。使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。通用控制定时器（TIMx）和高级定时器（TIMy）是完全独立的，它们不共享任何资源。它们可以同步操作。

20.2 TIM2/TIM3 主要特征

主要功能包括：

- 32 位（TIM2）或 16 位递增、递减、递增/递减的自动装载计数器
- 16 位可编程(可以实时修改)的预分频器，计数器时钟频率的分频系数为 1～65536 之间的任意数值
- 多达 4 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边缘或中央对齐模式）
  - 单脉冲模式输出
- 通过外部信号来控制定时器与定时器之间互联的同步电路
- 2 路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
- 如下事件发生时产生中断/DMA：
  - 更新：计数器向上溢出/向下溢出，计数器初始化（通过软件或者内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持针对定位的增量（正交）编码器
- 触发输入作为外部时钟或者按周期的电流管理
- 支持计数器多点触发 ADC

Puya Confidential

20.3 TIM2/TIM3 功能描述

20.3.1 TIM2/TIM3 模块框图

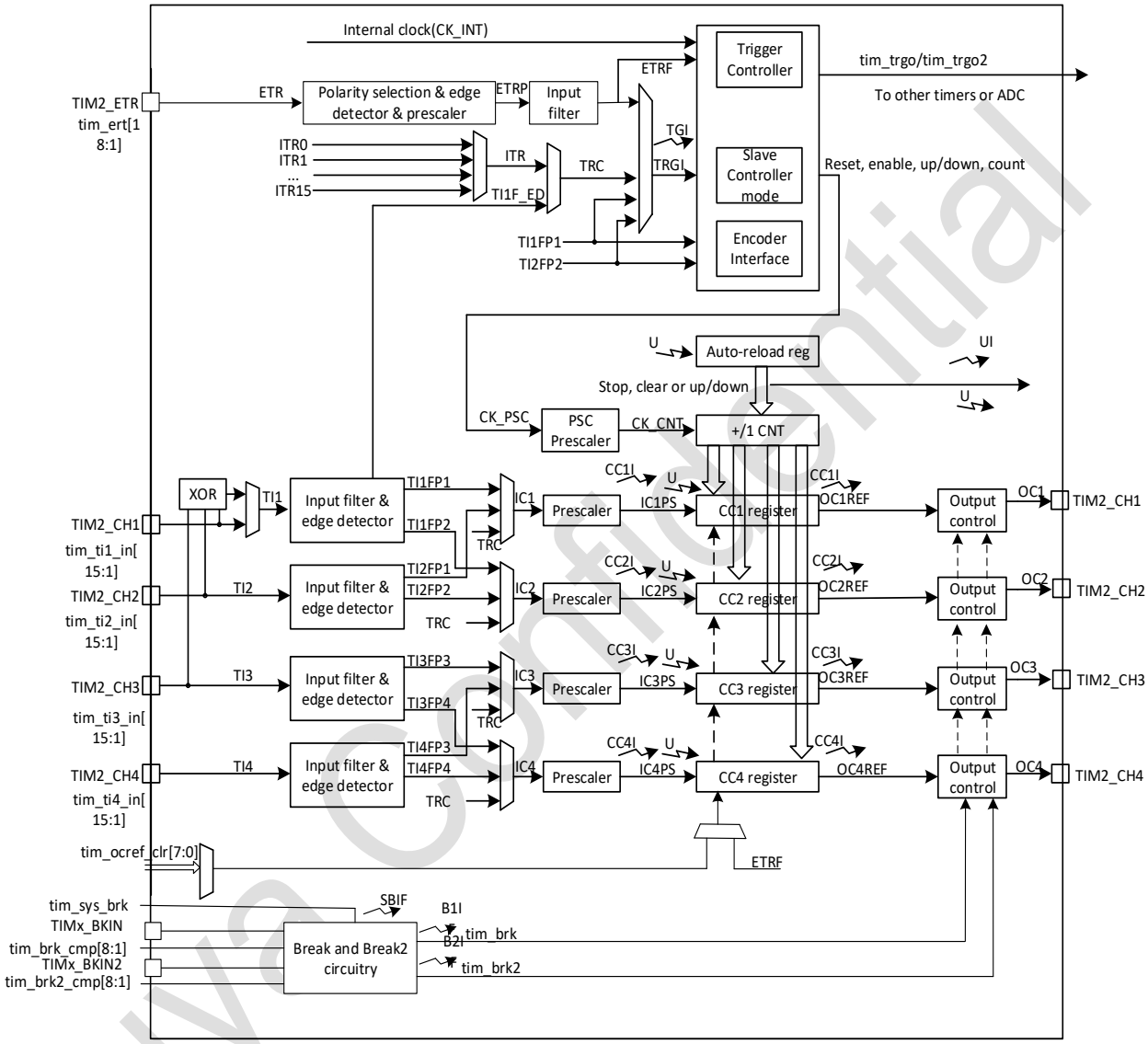


图 20-1 通用定时器框图（TIM2/TIM3）

20.3.2 时基单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以递增计数、递减计数或者递增递减双向计数。计数器的时钟可以被预分频器分频。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

时基单元包含：

- 计数器寄存器（TIMx_CNT）
- 预分频器寄存器（TIMx_PSC）
- 自动装载寄存器（TIMx_ARR）

Puya Confidential

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问它的预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位(ARPE)的设置，预装载寄存器的内容被立即或在每次的更新事件（UEV)时传送到影子寄存器。当计数器达到溢出条件(上溢或者下溢)并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，会产生更新事件。更新事件也可以由软件及其他条件产生。后续会详细描述每一种配置下更新事件的产生。

计数器由预分频器分频后的时钟输出 CK_CNT 驱动，仅当设置了 TIMx_CR1 寄存器中的计数器使能位(CEN)，CK_CNT 才对计数器有效。(更多有关使能计数器的细节，请参见从模式控制器的描述)。

注意，在设置了 TIMx_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个（在 TIMx_PSC 寄存器中的) 16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频参数将在下一次更新事件到来时被采用。

下边几张图给出了在预分频器运行时，更改计数器参数的例子。

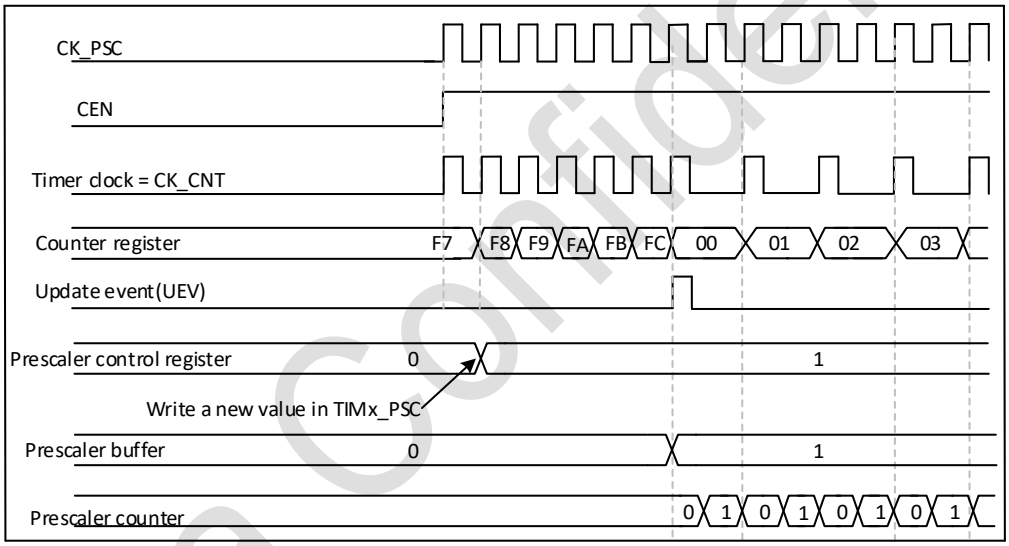


图 20-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

Puya Confidential

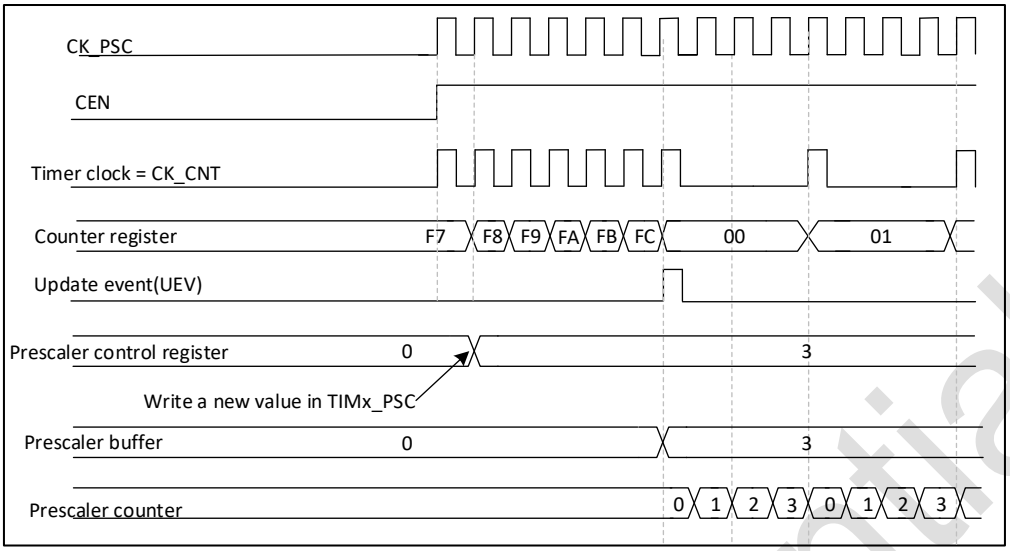


图 20-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

20.3.3 计数器模式

递增计数模式

在递增计数模式中，计数器从 0 计数到自动加载值（TIMx_ARR 的内容），然后重新从 0 开始计数并且产生一个计数上溢事件。而在 TIMx_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。

通过设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清‘0’之前，将不会产生更新事件。即使这样，在应该产生更新事件时，计数器仍会被清‘0’，同时预分频器内部的计数器也被清‘0’（但预分频器的数值不变）。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），通过设置 UG 位可以产生一个更新事件 UEV，但不会置起 UIF 标志位（即不会产生中断或 DMA 请求）。这是为了避免在捕获事件时清除计数器，同时产生更新和捕获中断。

当发生一个更新事件时，所有以下的寄存器都被更新，硬件同时(依据URS位)设置更新标志位(TIMx_SR 寄存器中的 UIF 位）：

- 自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。
- 预分频器的缓冲区被置入预装载寄存器的值（TIMx_PSC 寄存器的内容）。

下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下的动作。

Puya Confidential

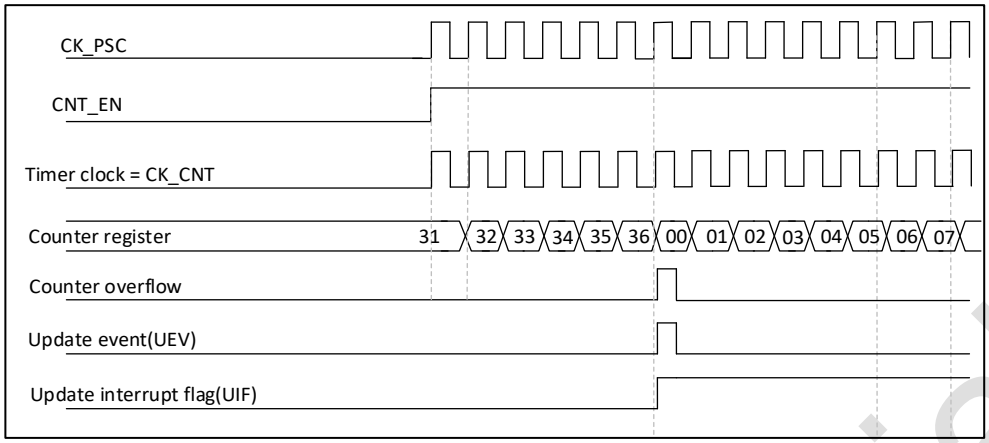


图 20-4 计数器时序图，内部时钟分频因子为 1

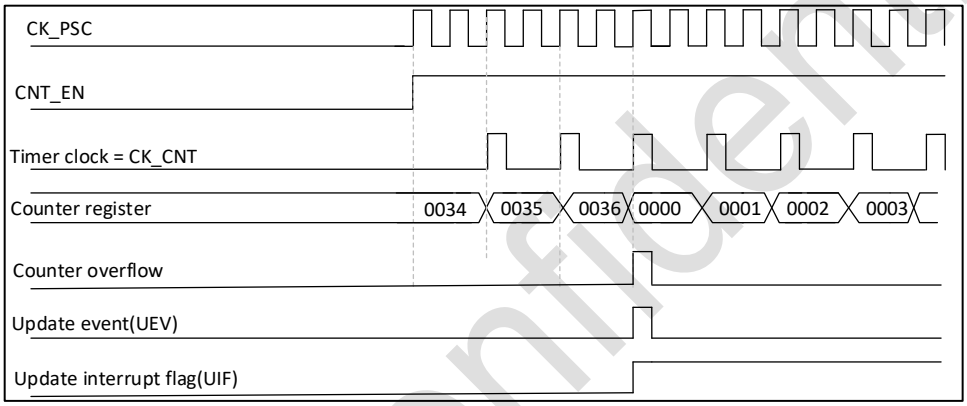


图 20-5 计数器时序图，内部时钟分频因子为 2

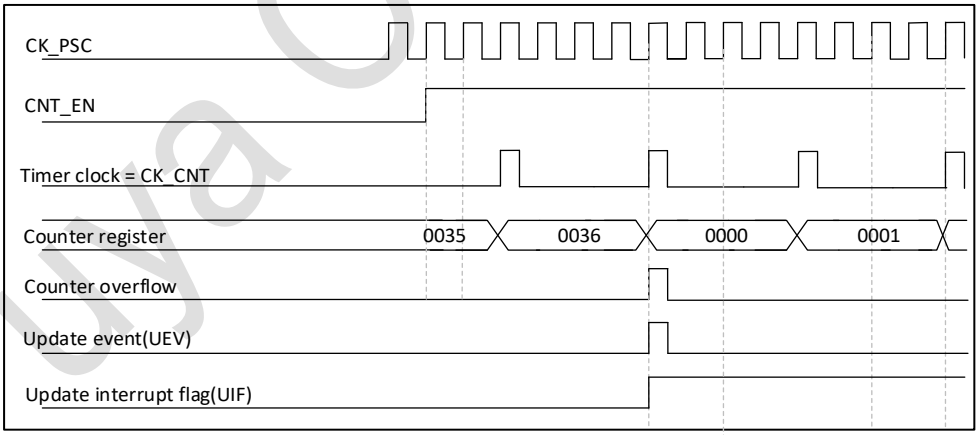


图 20-6 计数器时序图，内部时钟分频因子为 4

Puya Confidential

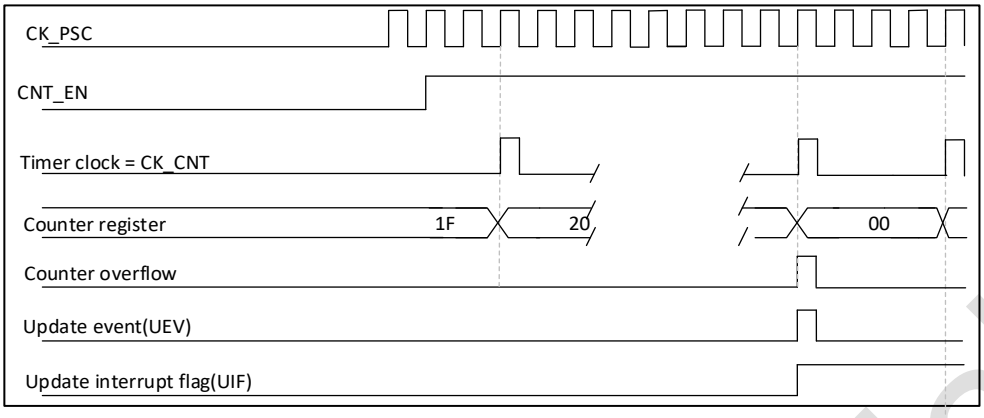


图 20-7 计数器时序图，内部时钟分频因子为 N

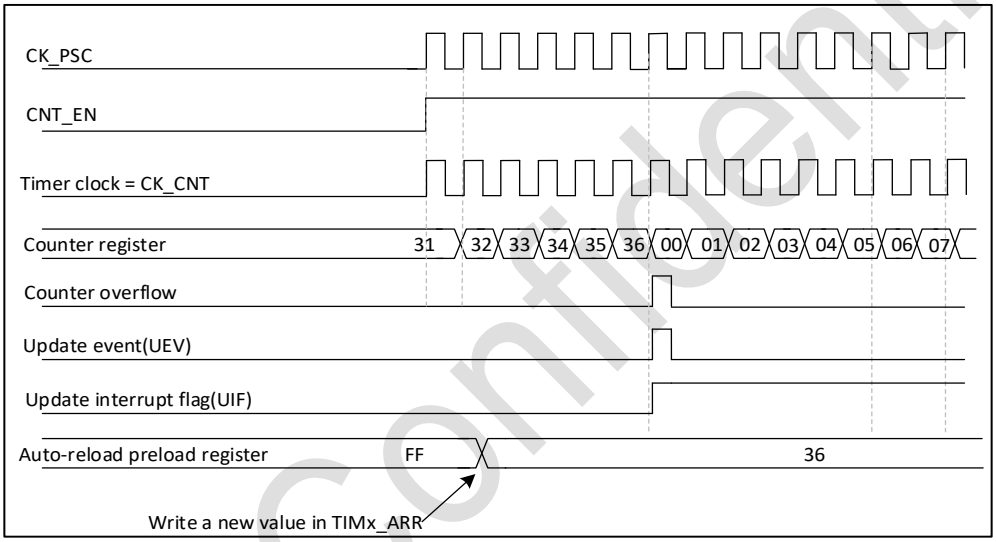


图 20-8 计数器时序图，当 ARPE=0 时的更新事件（没有预装 TIMx_ARR）

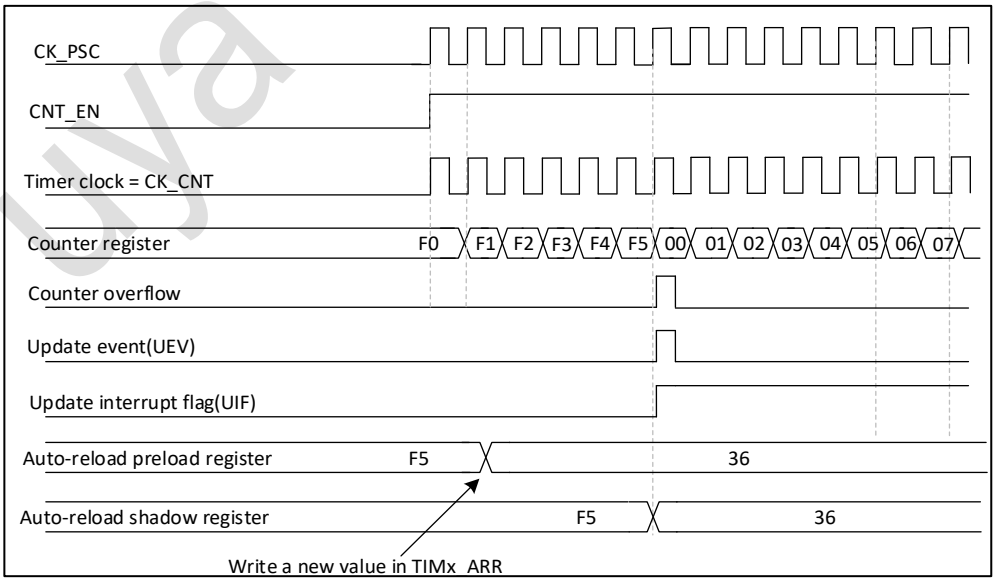


图 20-9 计数器时序图，当 ARPE=1 时的更新事件（预装了 TIMx_ARR）

Puya Confidential

递减计数模式

在递减计数模式中，计数器从自动加载值（TIMx_ARR 的内容）开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数下溢事件。而在 TIMx_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。

通过设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清'0'之前不会产生更新事件。即使这样，在应该产生更新事件时，计数器仍会从当前自动加载值重新开始计数，同时预分频器内部的计数器被清'0'（但预分频系数不变）。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），通过设置 UG 位可以产生一个更新事件 UEV，但不置起 UIF 标志位（即不会产生中断或 DMA 请求），这是为了避免在发生捕获事件时清除计数器，同时产生更新和捕获中断。

当发生一个更新事件时，所有以下的寄存器都被更新，硬件同时（依据 URS 位）设置更新标志位（TIMx_SR 寄存器中的 UIF 位）：

- 预分频器的缓冲区被置入预装载寄存器的值（TIMx_PSC 寄存器的内容）。
- 当前的自动加载寄存器被更新为预装载值（TIMx_ARR 寄存器中的内容）。

注：自动加载值在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIMx_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

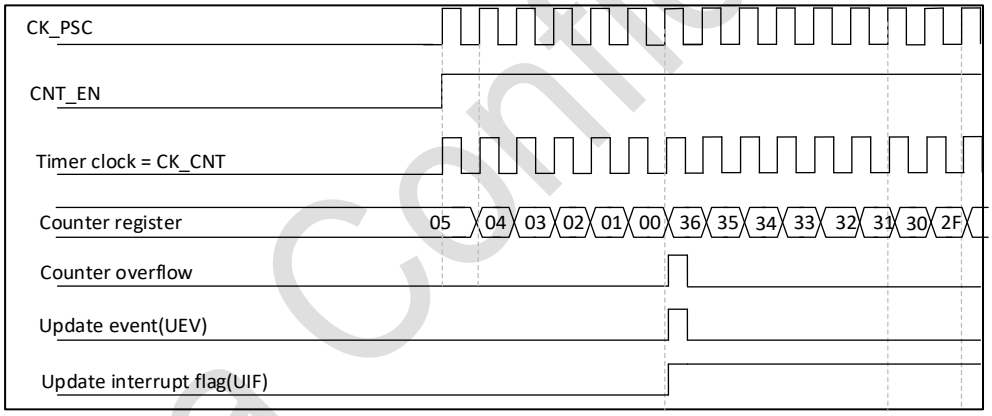


图 20-10 计数器时序图，内部时钟分频因子为 1

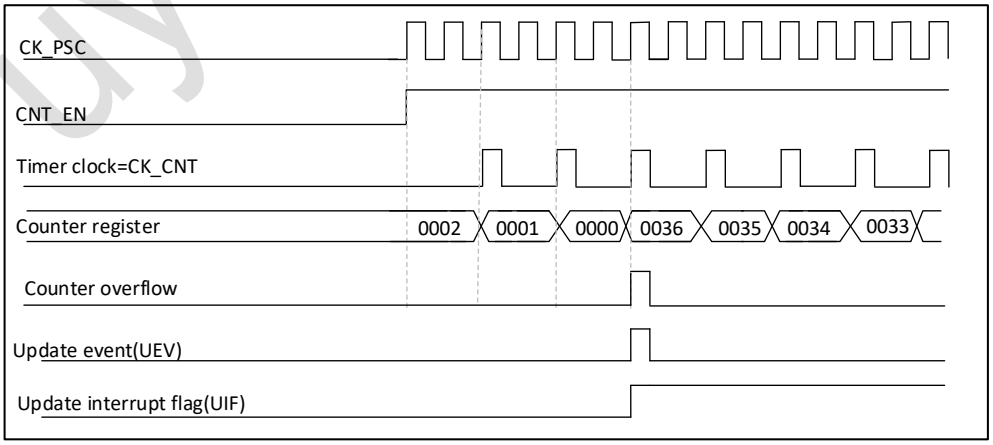


图 20-11 计数器时序图，内部时钟分频因子为 2

Puya Confidential

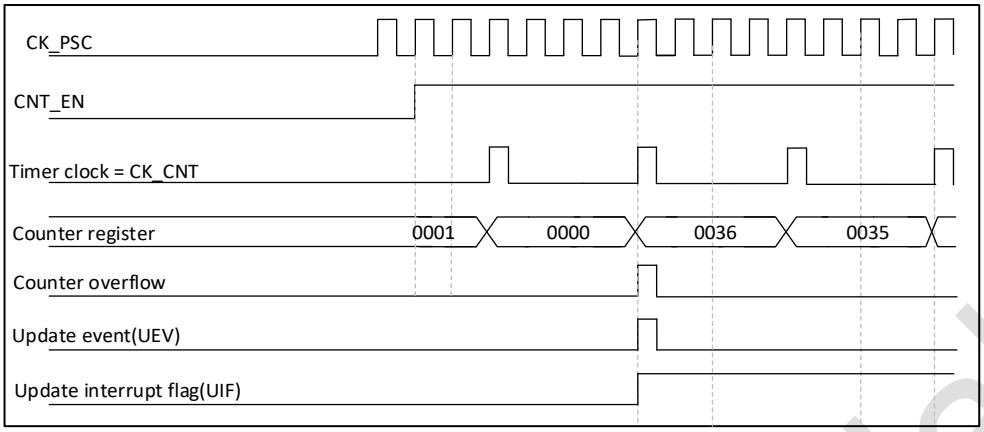


图 20-12 计数器时序图，内部时钟分频因子为 4

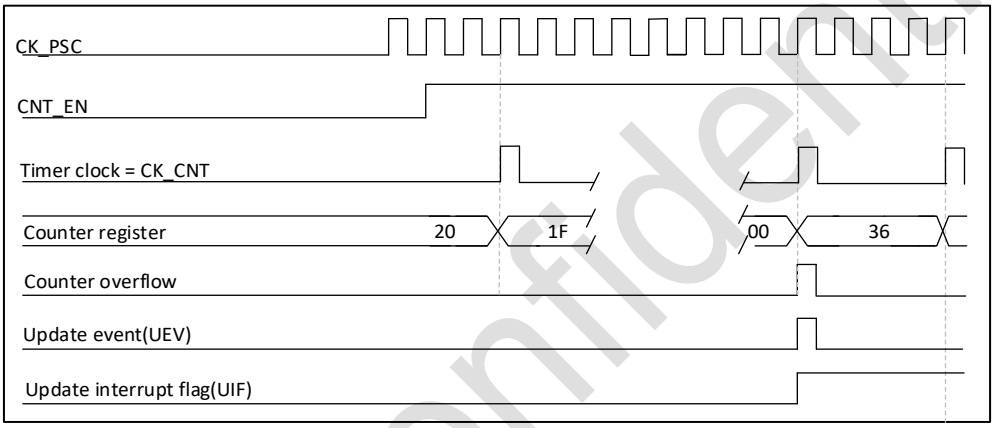


图 20-13 计数器时序图，内部时钟分频因子为 N

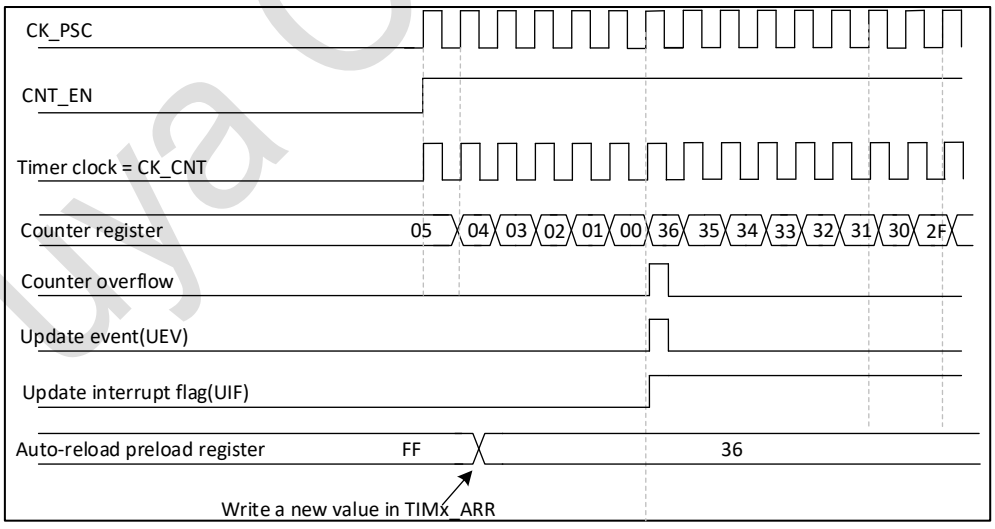


图 20-14 计数器时序图，当没有使用重复计数器时的更新事件

中央对齐模式(递增/递减计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值（TIMx_ARR 寄存器）减 1，产生一个计数器上溢事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

Puya Confidential

通过配置 TIMx_CR1 寄存器中的 CMS 位不为‘00’可以得到中央对齐模式。在通道配置为输出模式下的输出比较标志位在以下几种计数过程中会被置起：向下计数时（中央对齐模式 1，CMS=‘01’）、向上计数时（中央对齐模式 2，CMS=‘10’）、向上和向下计数时（中央对齐模式 3，CMS=‘11’）。

在此模式下，不能写入 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过(软件或者使用从模式控制器)设置 TIMx_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器内部计数器也重新从 0 开始计数。

设置 TIMx_CR1 寄存器中的 UDIS 位可以禁止更新事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。虽然 UDIS 位被清为 0 之前不会产生更新事件，但是计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），通过设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志（因此不产生中断和 DMA 请求），这是为了避免在发生捕获事件时清除计数器，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且（根据 URS 位的设置）更新标志位（TIMx_SR 寄存器中的 UIF 位）也被设置：

- 预分频器的缓存器被加载为预装载（TIMx_PSC 寄存器）的值。
- 当前的自动加载寄存器被更新为预装载值（TIMx_ARR 寄存器中的内容）。注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值（计数器被装载为新的值）。

以下是一些计数器在不同时钟频率下的操作的例子：

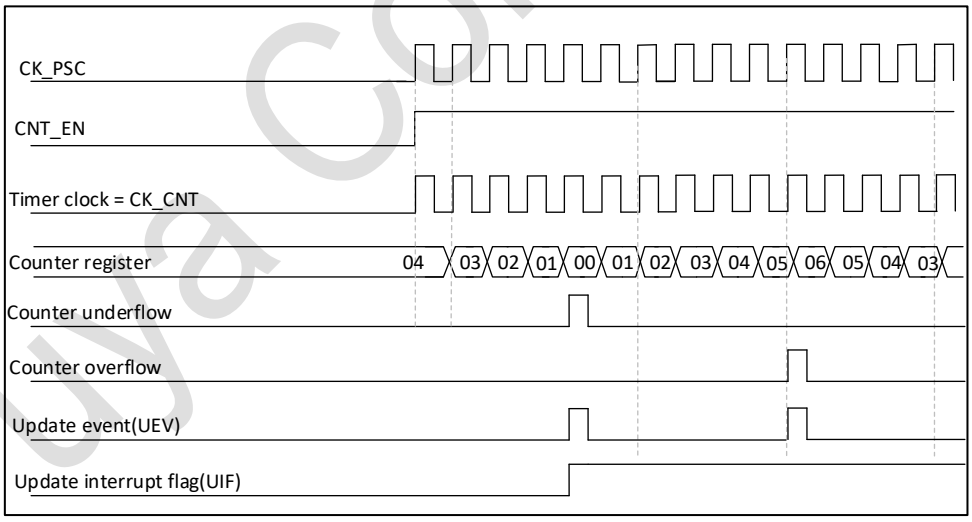


图 20-15 计数器时序图，内部时钟分频因子为 1，TIMx_ARR=0x6

Puya Confidential

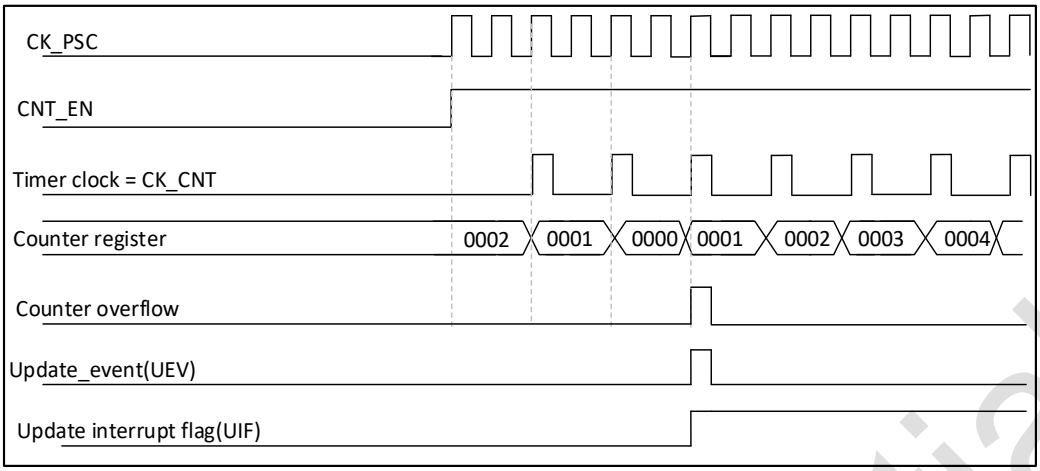


图 20-16 计数器时序图，内部时钟分频因子为 2

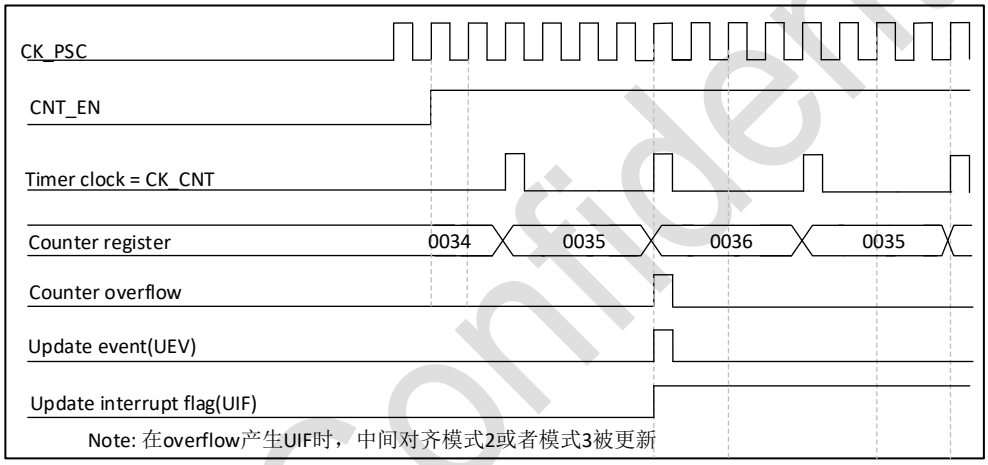


图 20-17 计数器时序图，内部时钟分频因子为 4，TIMx_ARR=0x36

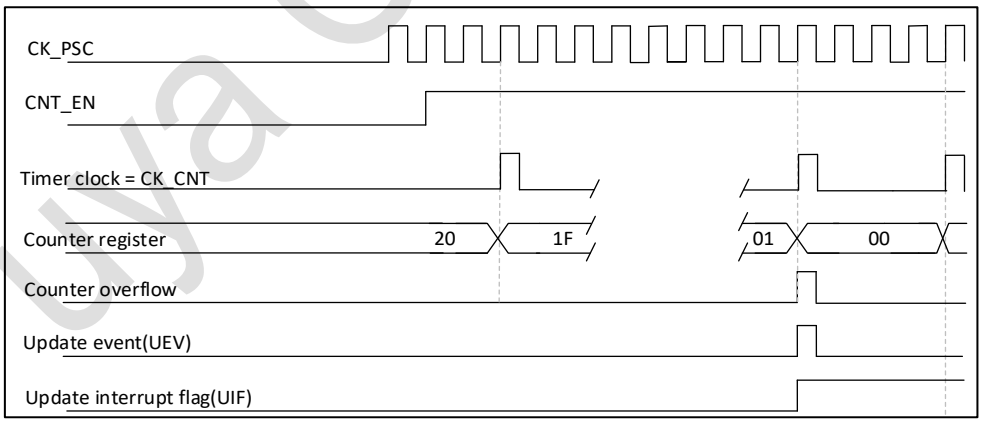


图 20-18 计数器时序图，内部时钟分频因子为 N

Puya Confidential

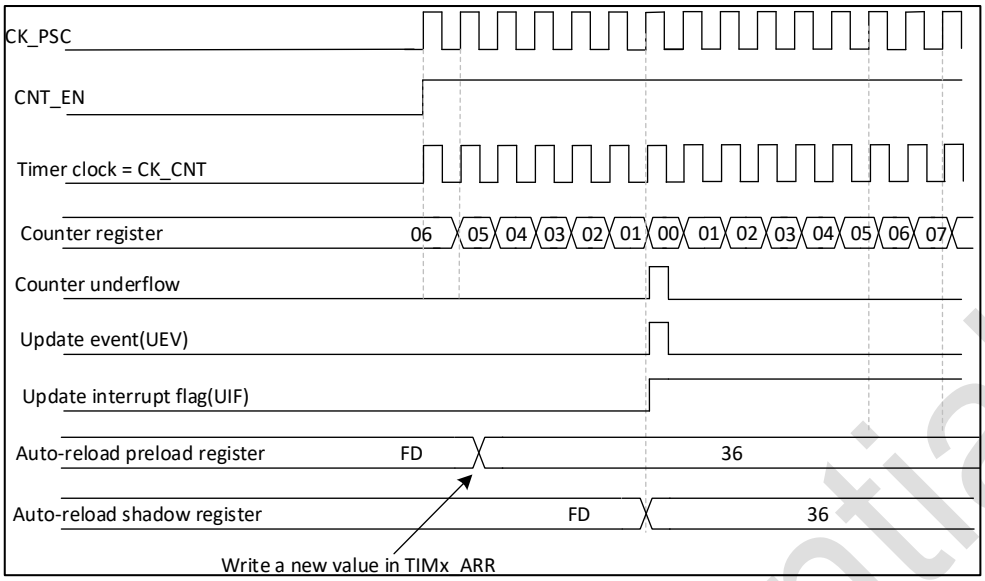


图 20-19 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

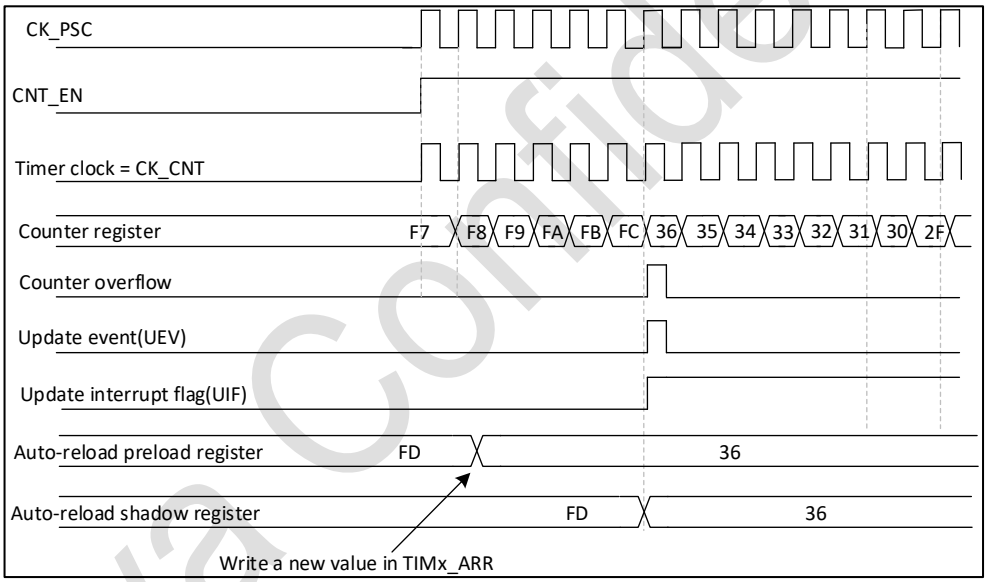


图 20-20 计数器时序图，ARPE=1 时的更新事件（计数器溢出）

20.3.4 外部触发输入

定时器外部触发输入 tim_etr，可用于：

- 外部时钟
- 从模式的触发
- 在周期电流调节中作为 PWM 复位输入

20.3.5 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚（TI1 和 TI2）
- 外部时钟模式 2：外部触发输入 ETR

Puya Confidential

- 内部触发输入（ITRx）：使用一个定时器作为另一个定时器的预分频器。如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。
- 编码器模式

内部时钟源（CK_INT）

如果禁止了从模式控制器（SMS=0000），则 CEN、DIR（TIMx_CR1 寄存器）和 UG 位（TIMx_EGR 寄存器）是事实上的控制位，并且只能被软件修改（UG 位仍被自动清除）。只要 CEN 位被写成‘1’，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。

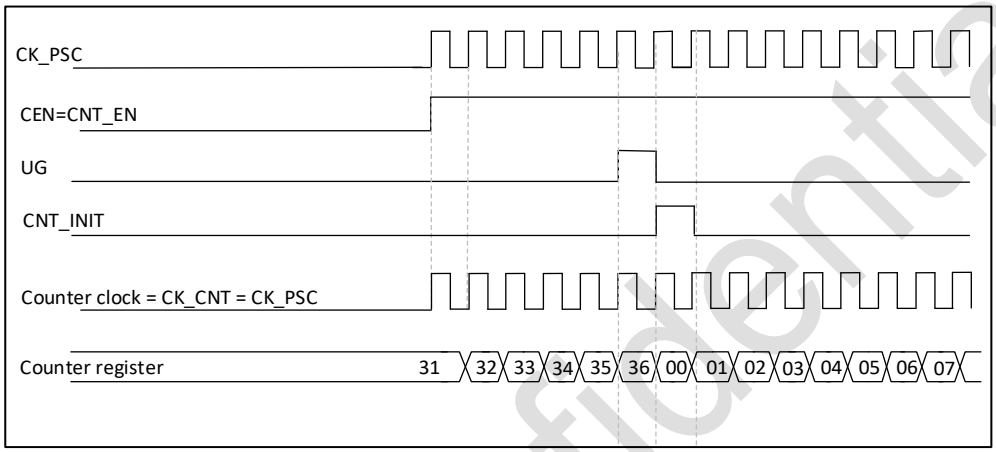


图 20-21 一般模式下的控制电路，内部时钟分频因子为 1

外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

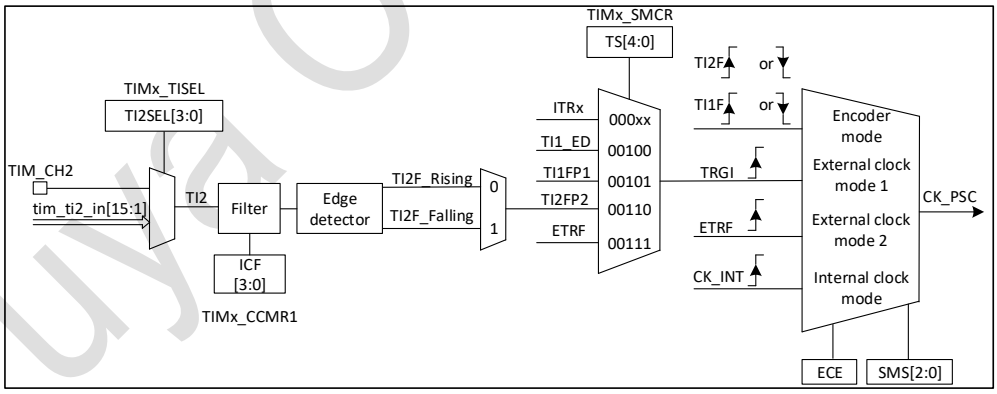


图 20-22 TI2 外部时钟连接例子

例如，要配置计数器在 TI2 输入端的上升沿向上计数，使用下列步骤：

1.配置 TIMx_CCMR1 寄存器 CC2S=01，使得通道 2 检测 TI2 输入端的上升沿；

2.配置 TIMx_CCMR1 寄存器的 IC2F[3:0],选择输入滤波器带宽(如果不需要滤波器，保持 IC2F=0000)；

3.配置 TIMx_CCER 寄存器的 CC2P=0，选定上升沿极性；

4.配置 TIMx_SMCR 寄存器的 SMS=111，选择定时器为外部时钟模式 1；

5.配置 TIMx_SMCR 寄存器中的 TS=00110，选定 TI2 作为触发输入源；

6.设置 TIMx_CR1 寄存器的 CEN=1，启动计数器。

Puya Confidential

注： 捕获预分频器不用作触发，所以不需要对它进行配置

当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的同步电路。

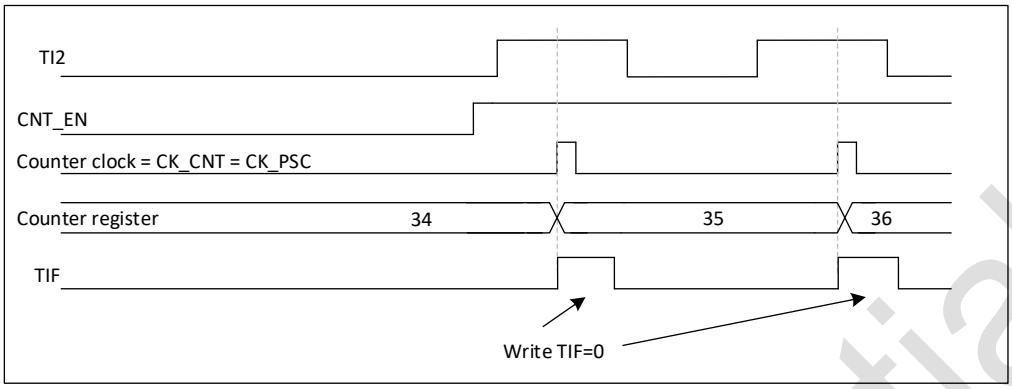


图 20-23 外部时钟模式 1 下的控制电路

外部时钟源模式 2

选定此模式的方法为：令 TIMx_SMCR 寄存器中的 ECE=1，计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图：

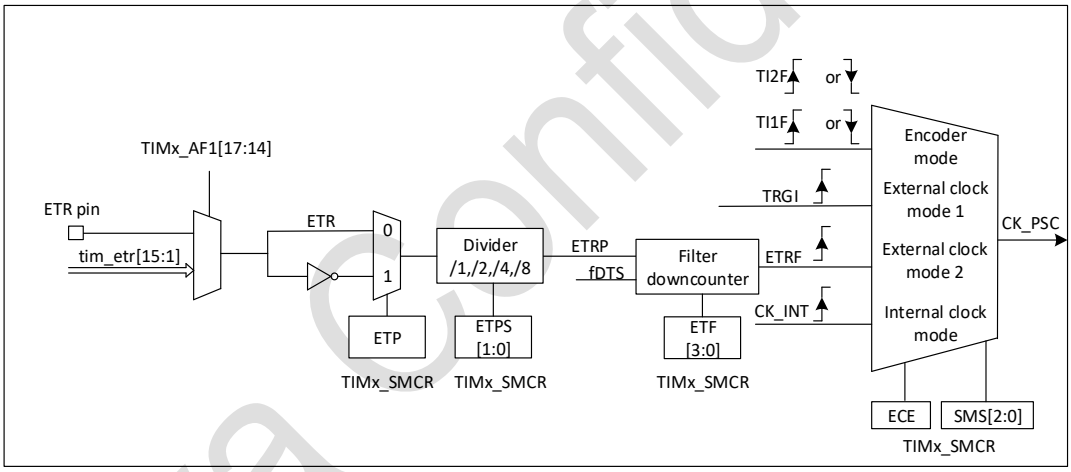


图 20-24 外部触发输入框图

例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：

1.本例中不需要滤波器，置 TIMx_SMCR 寄存器中的 ETF[3:0]=0000；

2.设置预分频器，置 TIMx_SMCR 寄存器中的 ETPS[1:0]=01；

3.选择 ETR 输入端的上升沿，置 TIMx_SMCR 寄存器中的 ETP=0；

4.开启外部时钟模式 2，写 TIMx_SMCR 寄存器中的 ECE=1；

5.启动计数器，写 TIMx_CR1 寄存器中的 CEN=1；

计数器在每 2 个 ETR 上升沿计数一次。

在 ETR 的上升沿和计数器实际时钟之间的延时取决于 ETRP 信号的同步电路。

Puya Confidential

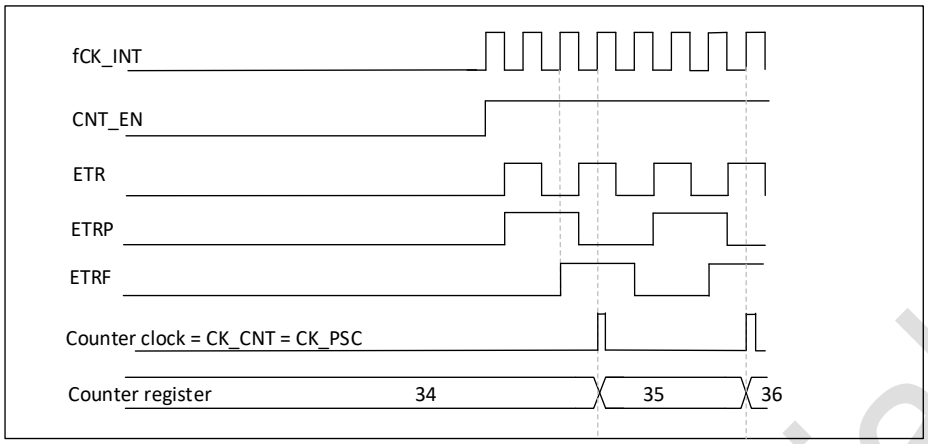


图 20-25 外部时钟模式 2 下的控制电路

20.3.6 捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器（包含影子寄存器），包括捕获的输入部分（数字滤波、多路复用和预分频器），和输出部分（比较器和输出控制）。

下面几张图是捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘监测器产生一个信号（TIxFPx），它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器（ICxPS）。

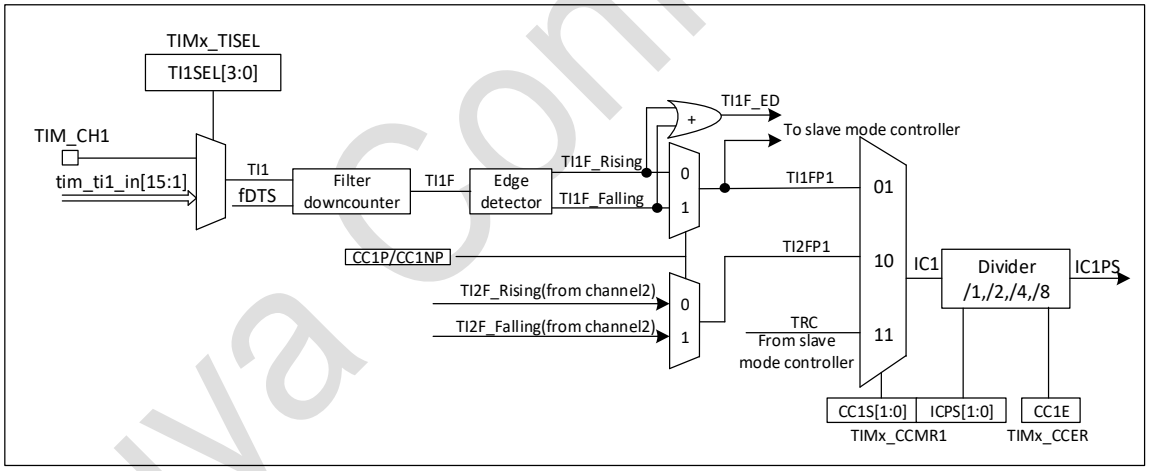


图 20-26 捕获/比较通道(如：通道 1 输入部分)

输出部分产生一个中间波形 OCxREF(高有效)作为基准，链的末端决定最终输出信号的极性。

Puya Confidential

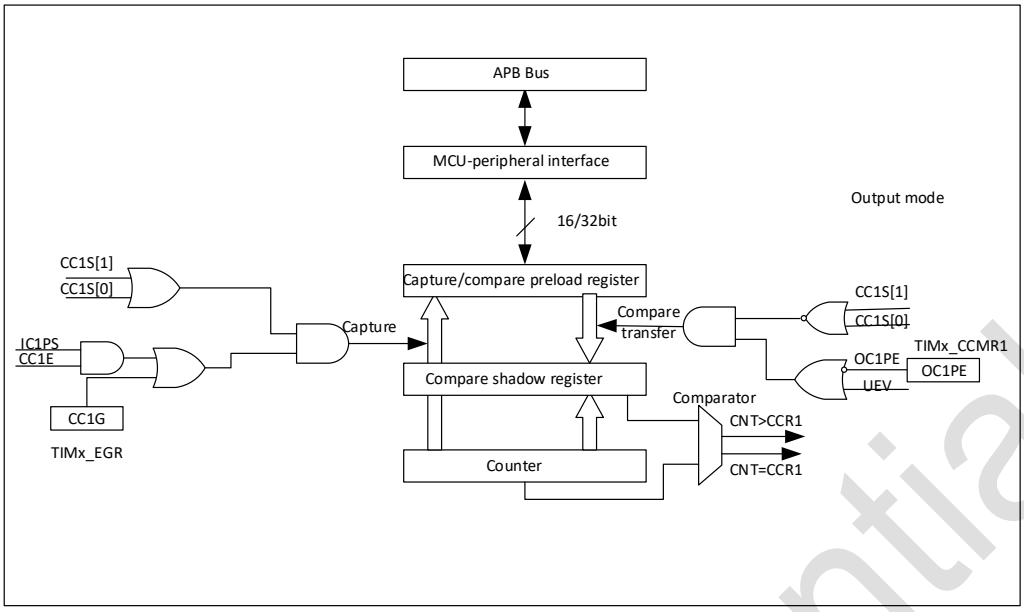


图 20-27 捕获/比较通道 1 的主电路

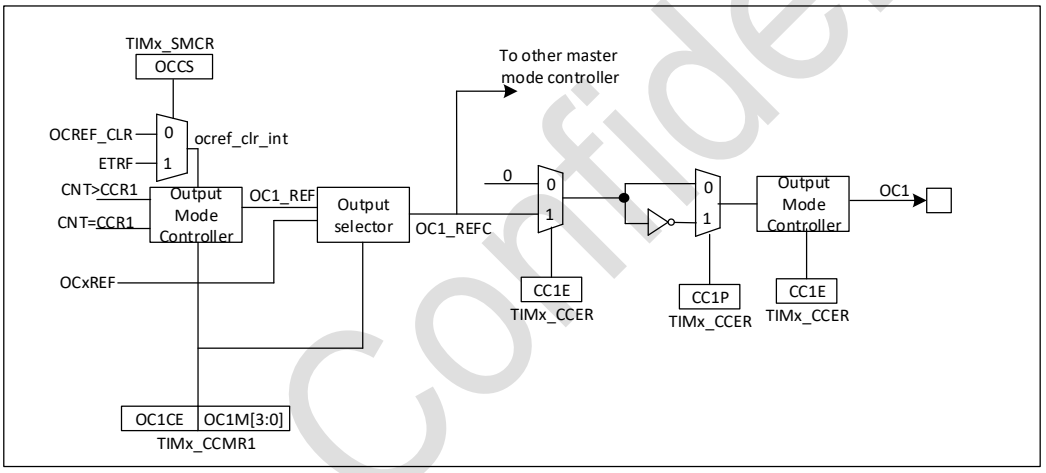


图 20-28 捕获/比较通道的输出部分（通道 1 至 4）

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

20.3.7 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIMx_CCRx）中。当发生捕获事件时，相应的 CCxIF 标志（TIMx_SR 寄存器）被置 1，如果使能中断或者 DMA 操作，则将产生中断或者 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高，那么过捕获标志 CCxOF（TIMx_SR 寄存器）被置 1。通过写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIMx_CCMR1 必须连接到 TI1 输入，所以写入 TIMx_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为‘00’，通道就被配置为输入，并且 TIMx_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽（即输入为 TIx 时，输入滤波器控制位是

Puya Confidential

TIMx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以（以 f_{DTS} 频率）连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011。

- 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0（设置为上升沿）。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写 TIMx_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，可以通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求，也可通过设置 TIMx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志位被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

为了处理过捕获，建议在过捕获标志被置起之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注： 设置 TIMx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

20.3.8 PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，其余操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，用户可以测量输入到 TI1 上的 PWM 信号的周期（TIMx_CCR1 寄存器）和占空比（TIMx_CCR2 寄存器），具体步骤如下

- 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01（选中 TI1）。
- 选择 TI1FP1 的有效极性（用来捕获数据到 TIMx_CCR1 中和清除计数器）：置 CC1P=0（上升沿有效）。
- 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- 选择 TI1FP2 的有效极性（捕获数据到 TIMx_CCR2）：置 CC2P=1(下降沿有效)。
- 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMx_CCER 寄存器中 CC1E=1 且 CC2E=1。

Puya Confidential

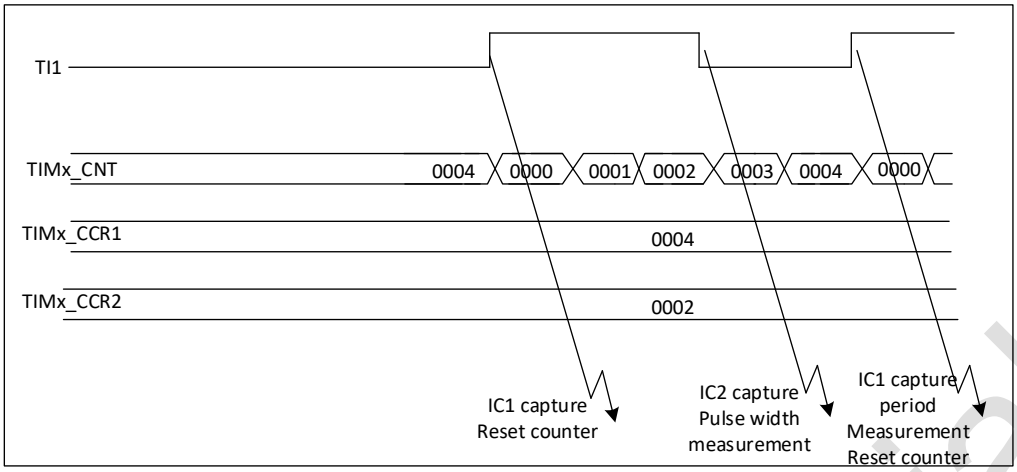


图 20-29 PWM 输入模式时序

因为只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIMx_CH1/TIMx_CH2 信号。

20.3.9 强置输出模式

在输出模式（TIMx_CCMRx 寄存器中 CCxS=00）下，输出比较信号（OCxREF 和相应的 OCx/OCxN）能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号（OCxREF/OCx）为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性相反的信号。例如： CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。

置 TIMx_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

该模式下，在 TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

20.3.10 输出比较模式

此项功能是用来控制一个输出波形，或者表明一段给定的的时间已经到时。当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位(TIMx_DIER 寄存器中的 CCxDE 位，TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

可以通过配置 TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟(内部，外部，预分频器)。

Puya Confidential

2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
   - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
   - 置 OCxPE = 0 禁用预装载寄存器
   - 置 CCxP = 0 选择极性为高电平有效
   - 置 CCxE = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器(OCxPE='0'，否则 TIMx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

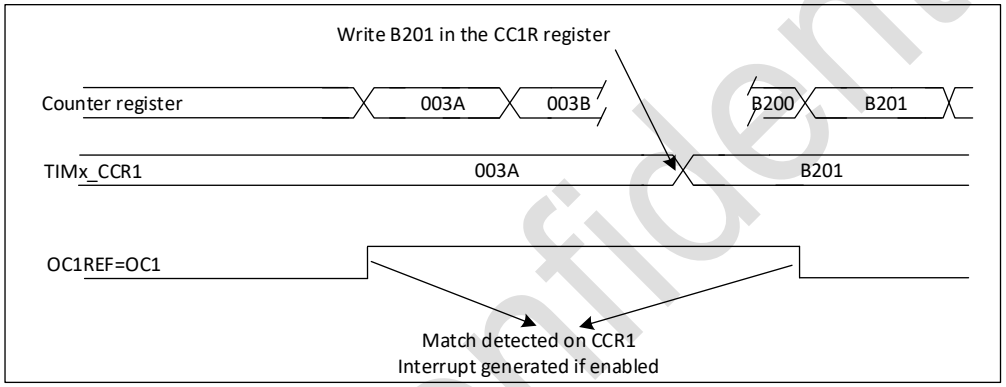


图 20-30 输出比较模式，翻转 OC1

20.3.11 PWM 模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比的信号。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入'110' (PWM 模式 1)或'111'(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位， (在向上计数或中心对称模式中)使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，用户必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。OCx 的输出使能通过(TIMx_CCER)CCxE、CCxNE 控制。详见 TIMx_CCER 寄存器的描述。

在 PWM 模式(模式 1 或模式 2)下，TIMx_CNT 和 TIMx_CCRx 始终在进行比较，(依据计数器的计数方向)以确定是否符合 TIMx_CCRx≤TIMx_CNT 或者 TIMx_CNT≤TIMx_CCRx。

根据 TIMx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

Puya Confidential

PWM 边沿对齐模式

- 向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。

下面是一个 PWM 模式 1 的例子。当 TIMx_CNT<TIMx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCRx 中的比较值大于自动重装载值(TIMx_ARR)，则 OCxREF 保持为‘1’。如果比较值为 0，则 OCxREF 保持为‘0’。 下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

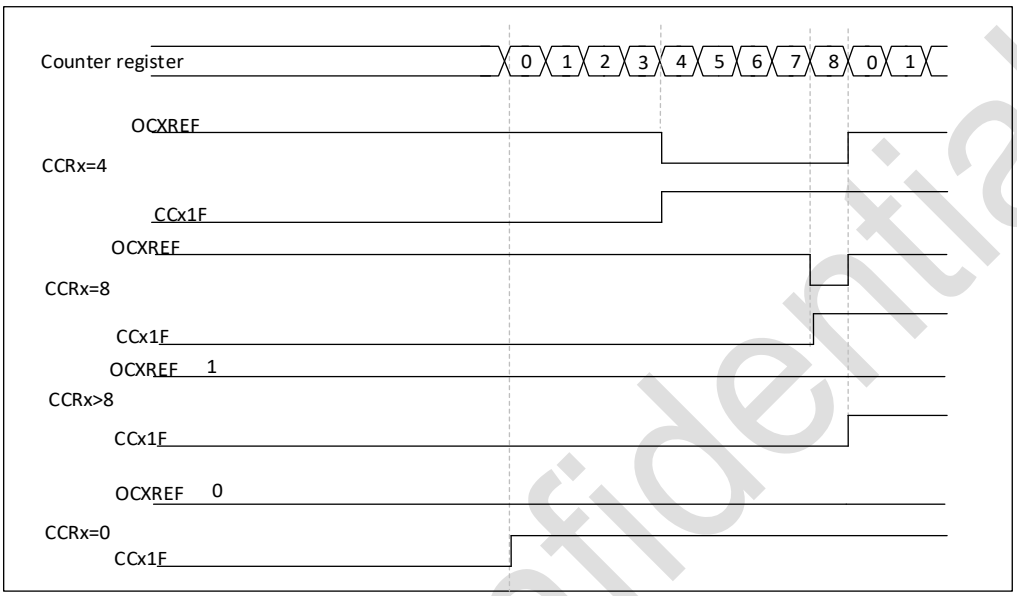


图 20-31 边沿对齐的 PWM 波形（ARR=8）

- 向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1，当 TIMx_CNT>TIMx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重装载值，则 OCxREF 保持为‘1’。该模式下不能产生 0%的 PWM 波形。

PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为‘00’时为中央对齐模式（CMS 位的所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位(DIR)由硬件更新，不要用软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子

- TIMx_ARR=8
- PWM 模式 1
- TIMx_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

Puya Confidential

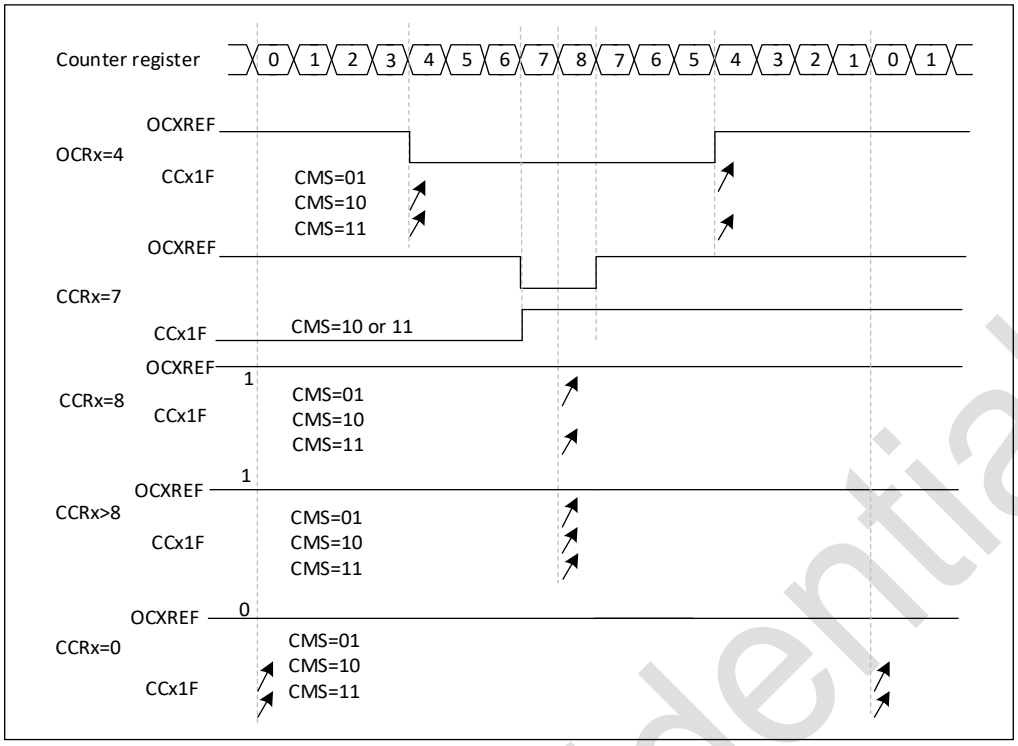


图 20-32 中央对齐的 PWM 波形(APR=8)

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIMx_CR1 寄存器中 DIR 位的当前值。此外，不要通过软件同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：

– 如果写入计数器的值大于自动重加载的值（TIMx_CNT>TIMx_ARR），则方向不会被更新。

例如，如果计数器正在向上计数，它就会继续向上计数。

– 如果将 0 或者 TIMx_ARR 的值写入计数器，方向被更新，但不会产生更新事件 UEV。

- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新（设置 TIMx_EGR 位中的 UG 位），并且不要在计数进行过程中修改计数器的值。

20.3.12 互补输出和死区插入

控制定时器能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。

这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

配置 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性（主输出 OCx 或互补输出 OCxN）。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是，在转换到 IDLE 状态时（MOE 下降到 0）死区被激活。

Puya Confidential

同时设置 CCxE 和 CCxNE 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。通过配置 TIMx_BDTR 寄存器中的 DTG[7:0]位，可以控制所有通道的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效：

- OCx 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度（OCx 或者 OCxN），则不会产生相应的脉冲。

下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。(假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

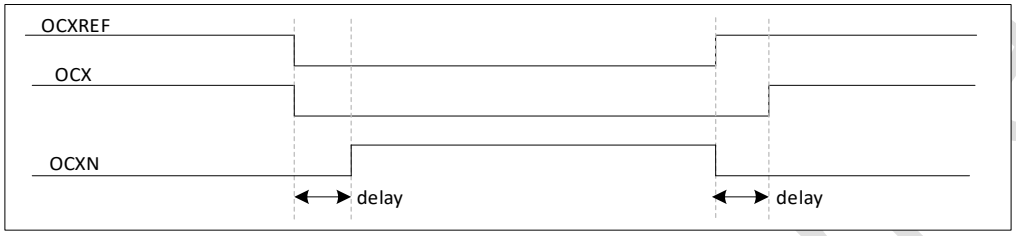


图 20-33 带死区插入的互补输出

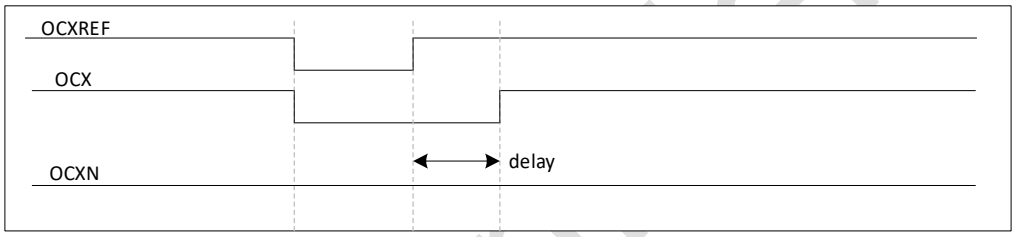


图 20-34 死区波形延迟大于负脉冲

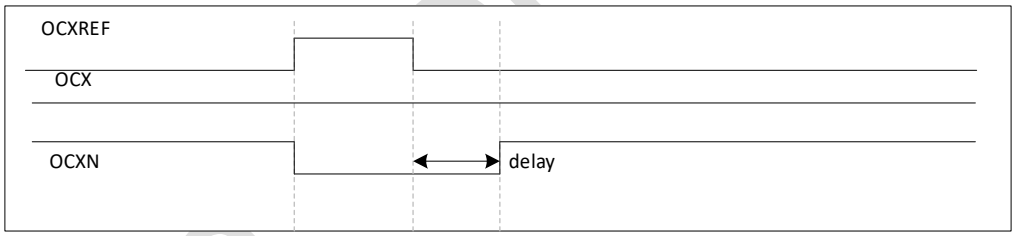


图 20-35 死区波形延迟大于正脉冲

当 DTAE=1，上升沿死区时间由 TIMx_BDTR 的 DTG[7:0]位配置，下降沿死区时间由 TIMx_DTR2 的 DTGF[7:0]位配置。DTAE 的写入需要在计数器使能前，当 CEN=1 时，不允许修改 DTAE。

可以通过预加载的机制在 PWM 工作过程中修改死区时间，当 TIMX_DTR2 的 DTPE 被置位时死区时间配置寄存器 DTG[7:0]和 DTGF[7:0]被预加载，在下一个更新事件时预加载的数据被加载到缓存寄存器中。

注：如果在计数器被使能时，DTPE 位被置位，则自最后一次更新以来的写的新数据都会被丢弃，而使用之前的值

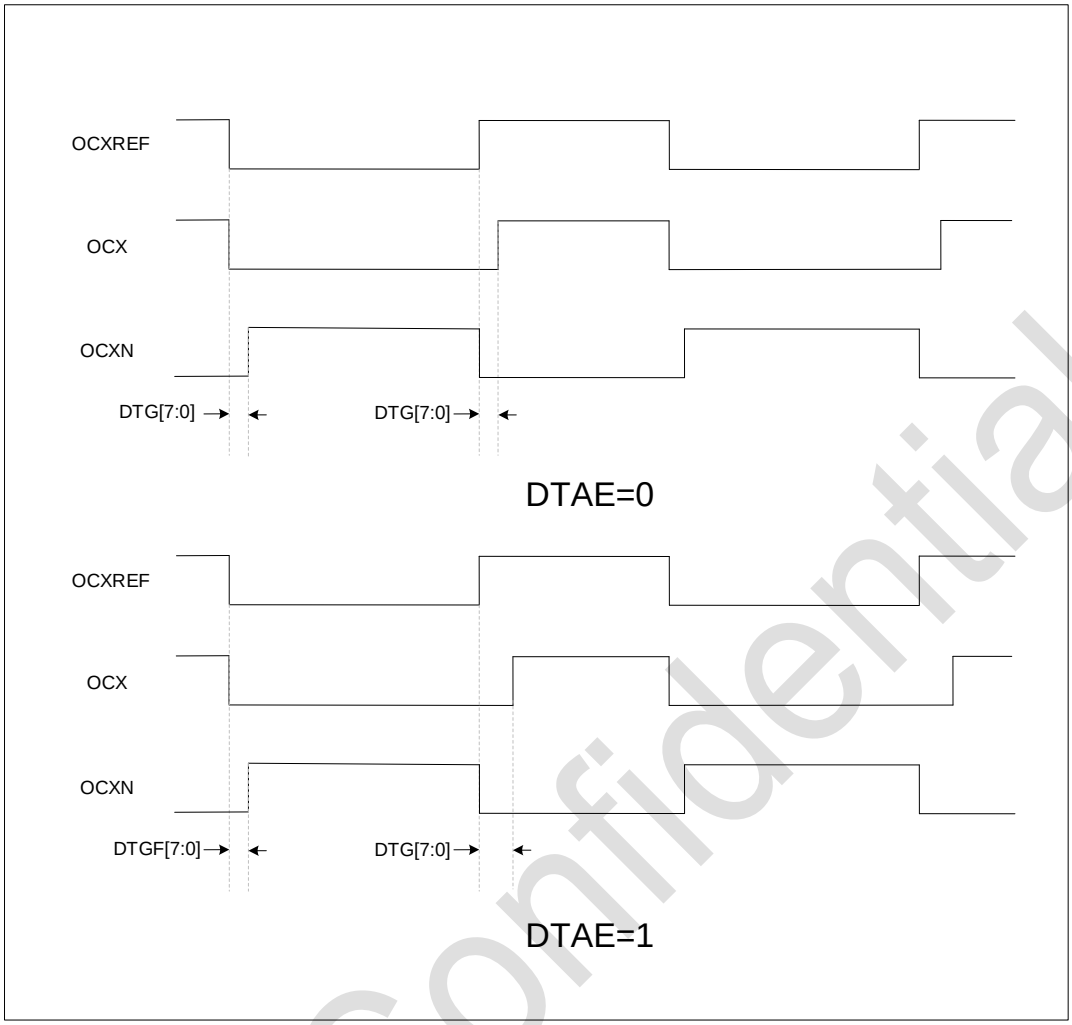


图 20-36 非对称死区时间

重定向 OCxREF 到 OCx 或 OCxN

在输出模式下（强置、输出比较或 PWM），通过配置 TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。

这个功能可以在互补输出处于无效电平时，在某个输出上送出一个特殊的波形（例如 PWM 或者静态有效电平）。另一个作用是，让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

注：当只使能 OCxN（CCxE=0, CCxNE=1）时，它不会反相，当 OCxREF 有效时立即变高。例如，如果 CCxNP=0，则 OCxN=OCxREF。另一方面，当 OCx 和 OCxN 都被使能时（CCxE=CCxNE=1），当 OCxREF 为高时 OCx 有效；而 OCxN 相反，当 OCxREF 低时 OCxN 变为有效。

20.3.13 使用刹车功能

刹车功能的目的是保护由 TIMER 输出的 PWM 驱动的电源开关。两路刹车的输入通常是连接到三相逆变器和功率级芯片的故障输出。当刹车信号有效时，会立即关断 PWM 的所有输出，并将它们强制到预先设置好的一个安全的电平状态。部分 MCU 内部的事件也可以作为触发信号来关闭输出。

刹车事件有两路通道，通道 1 包括了系统级的故障（clock failure、ECC、奇偶校验等）和应用层的故障（通过输入引脚或者内置的比较器），并且可以在一个死区区间后强制输出到预先设置好的电平（无论工作还是空闲）。

Puya Confidential

在刹车过程中输出使能信号和输出电平由以下多个 bit 控制

- TIMx_BDTR 的 MOE 位可以通过软件或 2 路刹车事件复位来实现使能和关闭输出
- TIMx_BDTR 的 OSSI 位用于定义当处于空闲状态时 timer 是否还控制输出，或者释放对 GPIO 控制器的控制（高阻状态）

TIMx_CR2 的 OISx 和 OISxN 位用于无论工作还是空闲状态下都将输出设置为无效电平，OCx 和 OCxN 输出在同一时刻不能同时设置为有效电平，无论 OISx 和 OISxN 为何值。

当从复位中退出时，刹车功能被关闭并且 MOE 为 0。刹车功能的开启可通过使能 TIMxBDTR 的 BKE 位。刹车输入的极性可通过 BKP 配置。BKEx 和 BKPx 可同时修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的，在实际信号（作用在输出端）和同步控制位（在 TIMx_BDTR 寄存器中）之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当它为低时写 MOE=1，则读出它之前必须先插入一个延时（空指令）才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

刹车通道 1 的源；

- 外部源可连接到其中一个 TIMx_BKIN 引脚上（选择 GPIO 和配置寄存器），再加上极性选择和滤波
- 内部源包含 2 部分
  - 来自刹车比较器信号（tim_brk_cmpx）
  - 来自系统刹车请求

刹车通道 2 的源；

- 外部源可连接到其中一个 TIMx_BKIN2 引脚上（选择 GPIO 和配置寄存器），极性选择和滤波
- 内部源来自刹车比较器信号（tim_brk2_cmpx）

Puya Confidential

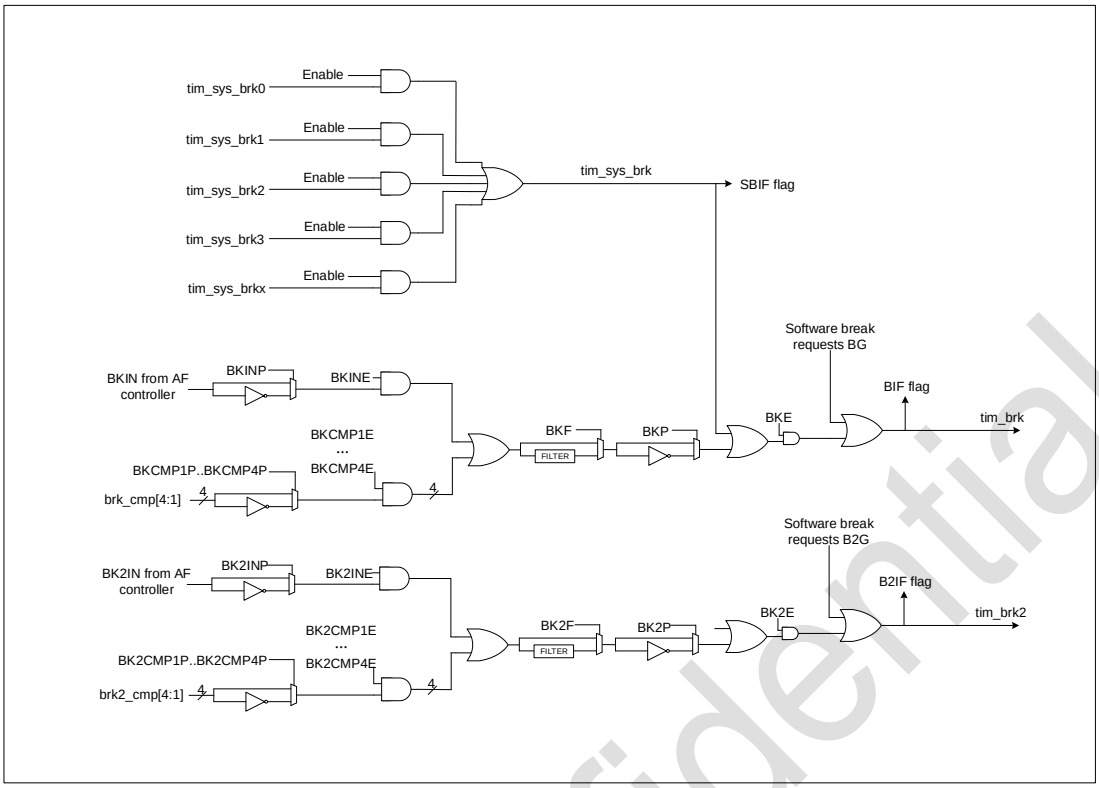


图 20-37 刹车电路

注意：只有没有滤波的时候才能保证是异步的操作，如果滤波开启了，必须使用一个故障保护时钟模式来保证刹车中断事件得到处理（例如使用内部 PLL 或 CSS，无滤波）。

当发生刹车时(在刹车输入端出现选定的电平)，有下述动作：

- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者释放对 GPIO 控制器的控制（由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIMx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放输出控制，否则使能输出始终为高。
- 当使用互补输出时：
  - 输出首先被置于复位状态即无效的状态（取决于极性)。这是异步操作，即使定时器没有时钟时，此功能也有效。
  - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些（大约 2 个时钟周期)。
  - 如果 OSSI=0，定时器释放输出，否则保持使能输出；或一旦 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 刹车状态标志会被拉高（TIMx_SR 的 SBIF,BIF)。如果设置了 TIMx_DIER 寄存器中的 BIE 位，当刹车状态标志(TIMx_SR 寄存器中的 BIF 位)为‘1’时，则产生一个中断。如果设置了 TIMx_DIER 寄存器中的 TDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位；例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置‘1’；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

Puya Confidential

注：当 AOE 为 1 时如果 MOE 被 CPU 复位了，输出会进入空闲状态，并被强制到无效电平或者高阻（取决于 OSSI 的值），如果 MOE 和 AOE 均被 CPU 复位，输出将进入无效状态并且由 OISx 位来驱动输出电平。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能同时(自动地或者通过软件)设置 MOE。同时，状态标志 BIF 不能被清除。

除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区持续时间， OCx/OCxN 极性和被禁止的状态， OCxM 配置，刹车使能和极性)。用户可以通过设置 TIMx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看刹车和死区寄存器(TIMx_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

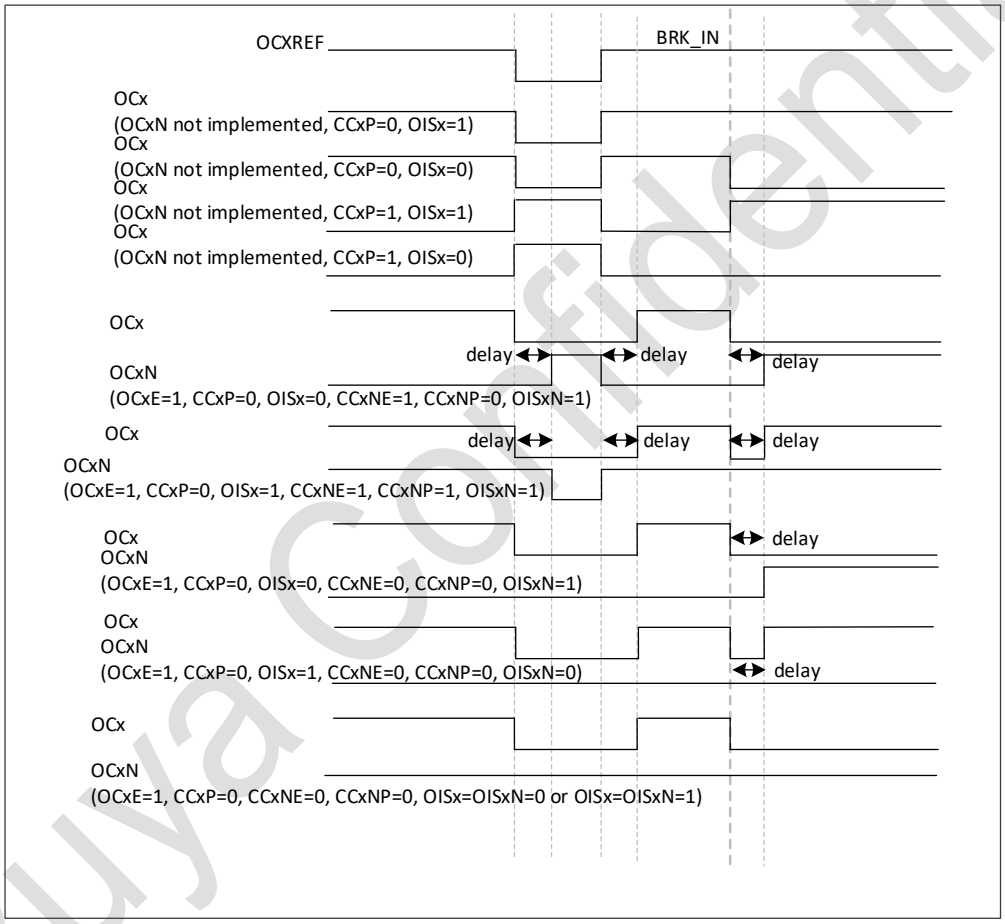


图 20-38 响应刹车的输出

Puya Confidential

20.3.14 双向刹车输入

TIM_TYP1 具备双向的刹车 I/O，如下图所示：

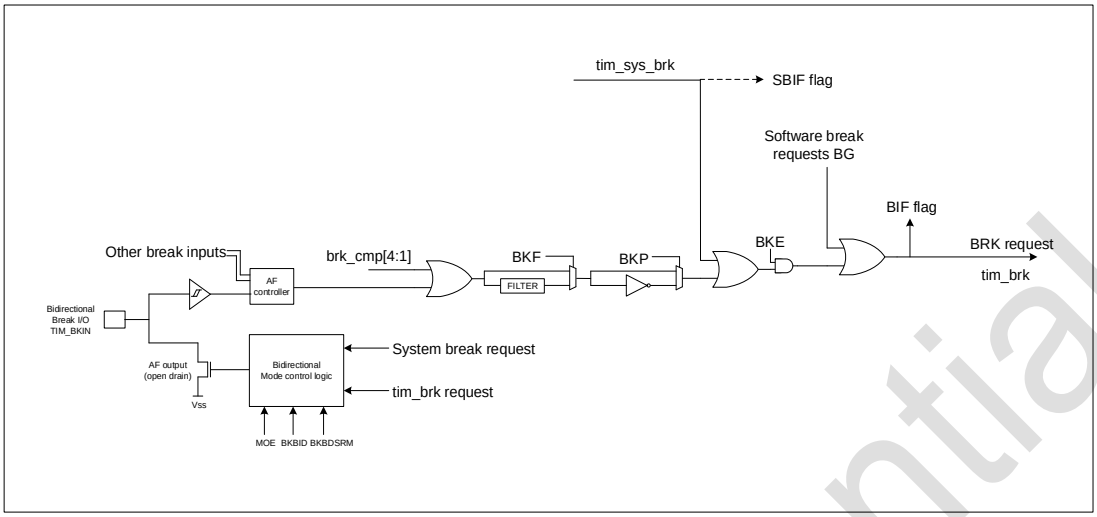


图 20-39 输出重定向（刹车 2 无请求时）

I/O 支持特性：

- 提供一种板级的全局刹车信号，通过唯一的既可做输入也可做输出的引脚，向外部 MCU 和栅极驱动器发送故障信号
- 当需要合并多个内部和外部的刹车源时，内部的刹车源和多个外部的开漏源进行或运算后作为刹车事件的的唯一触发信号

通过 TIMxBDTR 的 BKBID 和 BK2BID 位控制 tim_brk 和 tim_brk2 的输入为双向模式。BKBID 和 BK2BID 可以使用 TIMxBDTR 的 LOCK 位（LOCK level 1）设置为 1，实现被锁定为只读模式。

tim_brk 和 tim_brk2 输入都可使用双向模式，并且要求 I/O 配置成开漏模式且极性为低有效(通过 BKINP、BKP、BK2INP、BK2P 控制位）。无论来自系统的（例如 CCS）、还是板级外设的、还是刹车输入强制一个低电平表示刹车输入故障事件，都可以作为刹车请求。但出于安全的考虑，当极性位未有效配置时（配置成高电平有效时），双向模式会被禁止。

软件刹车事件 (BG、B2G) 也会引起刹车 I/O 被拉低，用于向外部设备显示本 timer 进入到了刹车状态。当然当 BKE 或 B2KE 为 1 时才有可以。当 BKE 或 B2KE 为 0，发生软件刹车事件，输出会被置为安全的状态并产生刹车标志信号，但是对 TIMx_BKIN 和 TIMx_BKIN2 I/O 没有影响

一种安全的解除机制可以防止系统被限制性锁死（在刹车输入口的低电平触发了一个刹车，该刹车又将输入口强制成低电平）

当 BKDSRM（BK2DSRM）位被置为 1 时，可以释放刹车输出控制，清除故障标志信号并提供重新配置系统的可能性。

任何情况下都不能关闭刹车保护电路：

- 刹车输入路径一直有效：即使 BKDSRM（BK2DSRM）位被置 1 和开漏控制被释放，刹车事件还是有效的。这防止在还处于刹车状态时 PWM 输出可以重启
- 当输出被使能的时候（MOE=1）,BKDSRM（BK2DSRM）位无法解除刹车保护

Puya Confidential

表 20-4 刹车保护状态

MOE	BKBID（BK2BID）	BKDSRM（BK2DSRM）	刹车保护状态
0	0	X	启用
0	1	0	启用
0	1	1	解除
1	X	X	启用

20.3.14.1 对刹车电路进行启用和重启用

默认情况下，刹车电路（在输入或双向模式下）处于启用状态

在发生刹车（或刹车 2）后需要遵循下面的程序来重启动保护：

- BKDSRM（BK2DSRM）位必须置 1 用于释放输出控制
- 软件必须要等到系统刹车状态消失然后清除 SBIF 状态标志（或者在重启用前系统地清除它）
- 软件必须轮询 BKDSRM(BK2DSRM)位直到它被硬件清除（当应用刹车状态消失）

通过这种方式，刹车电路被激活，并且 MOE 位可以被重新使能 PWM 输出

20.3.15 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIMx_CCMRx 寄存器中对应的 OCxCE 位为‘1’，能够用 OCREF_CLR_INT 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次计数溢出所产生的更新事件 UEV。该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

而 OCREF_CLR_INT 可以通过配置 TIMx_SMCR 寄存器中的 OCCS 位，在 OCREF_CLR 和 ETRF（ETR 滤波后）之间选择。

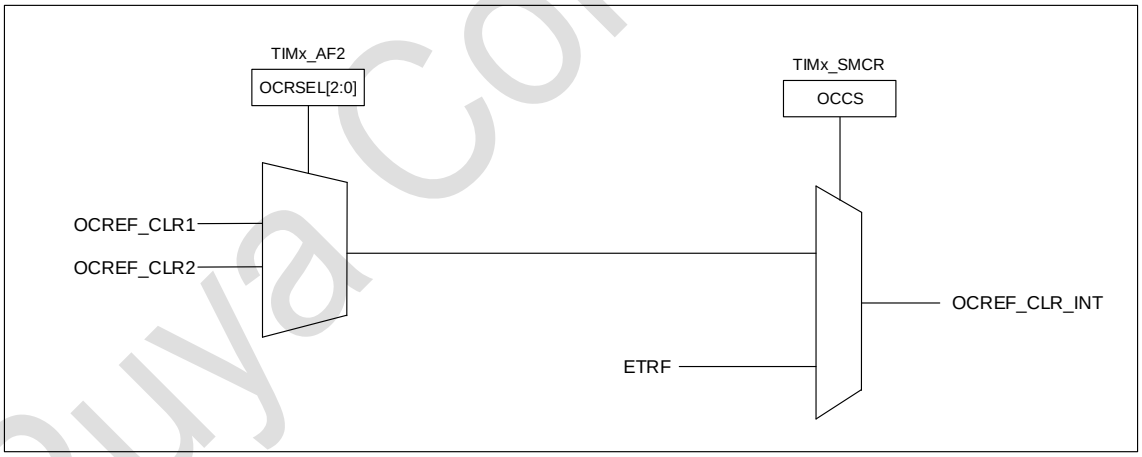


图 20-40 OCREF_CLR 输入选择

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭：TIMx_SMCR 寄存器中的 ETPS[1:0]=00。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE=0。
3. 外部触发极性（ETP）和外部触发滤波器（ETF）可以根据需要配置。

下图显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子中，定时器 TIMx 被置于 PWM1 模式。

Puya Confidential

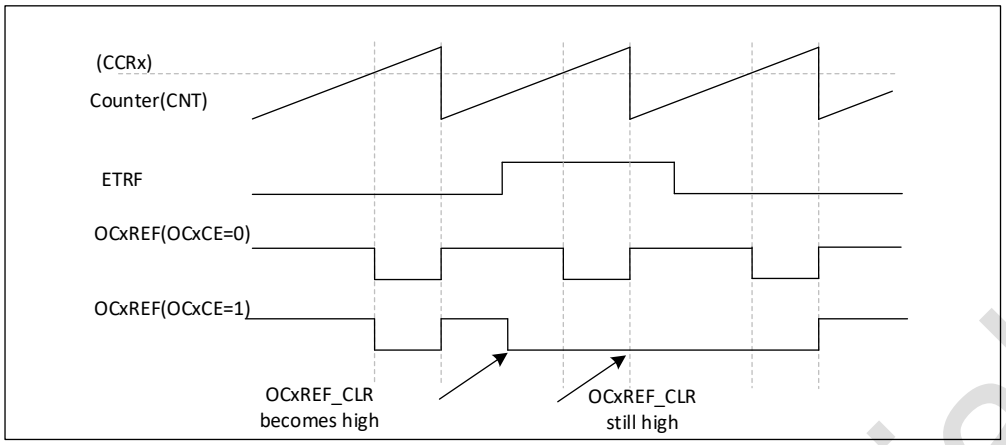


图 20-41 清除 TIMx 的 OCxREF

20.3.16 单脉冲模式

单脉冲模式（OPM）是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须如下配置：

- 向上计数方式：计数器 CNT < CCRx ≤ ARR（特别地，0 < CCRx），
- 向下计数方式：计数器 CNT > CCRx。

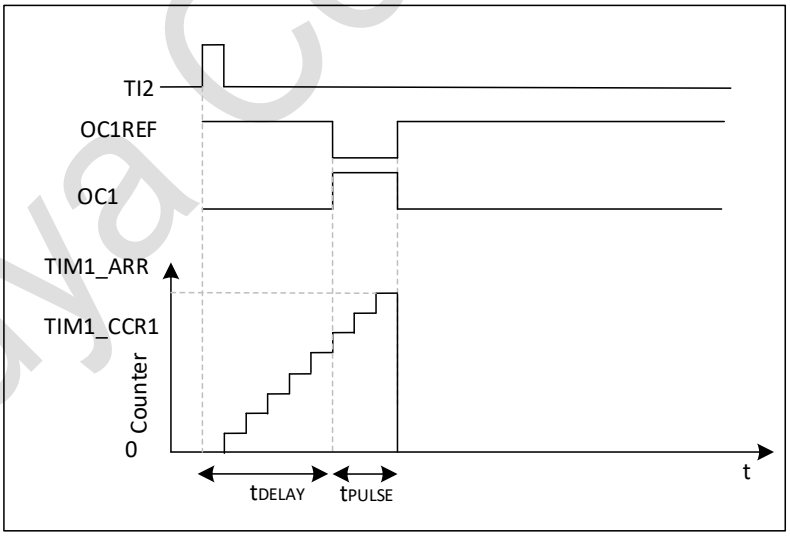


图 20-42 单脉冲模式的例子

例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

假定 TI2FP2 作为触发:

Puya Confidential

- 置 TIMx_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映射到 TI2。
- 置 TIMx_CCER 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- 置 TIMx_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发（TRGI）。
- 置 TIMx_SMCR 寄存器中的 SMS=110（触发模式），TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定（要考虑时钟频率和计数器预分频器）

- t_{DELAY} 由 TIMx_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义（TIMx_ARR - TIMx_CCR1）。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 0 到 1 的波形；首先要置 TIMx_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMR1 中的 OC1PE=1 和 TIMx_CR1 寄存器中的 ARPE；然后在 TIMx_CCR1 寄存器中填写比较值，在 TIMx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中， TIMx_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲，所以必须设置 TIMx_CR1 寄存器中的 OPM=1，在下一个更新事件（当计数器从自动装载值翻转到 0）时停止计数。

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 tDELAY。

如果要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF(和 OCx)直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。 OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

20.3.17 编码器接口模式

正交编码器

选择编码器接口模式的方法是：如果计数器只在 TI1 的边沿计数，则置 TIMx_SMCR 寄存器中的 SMS=0001；如果只在 TI2 边沿计数，则置 SMS=0010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=0011。

通过设置 TIMx_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参看下表，假定计数器已经启动(TIMx_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端(TI1 或者 TI2)的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_ARR 寄存器的自动装载值之间连续计数(根据方向，或是 0 到 ARR 计数，或是 ARR 到 0 计数)。所以在开始计数之前必须配置 TIMx_ARR；同样，捕获器、比较器、预分频器、重复计数器、触发输出

Puya Confidential

特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表 20-1 计数方向与编码器信号的关系（CC1P=CC2P=0）

有效沿	SMS[3:0]	反向信号电平 (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 信号		TI2FP2 信号	
			Rising	Falling	Rising	Falling
仅在 TI1 双沿计数 x2 模式	001	高	减	加	-	-
		低	加	减	-	-
仅在 TI2 双沿计数 x2 模式	010	高	-	-	加	减
		低	-	-	减	加
在 TI1 和 TI2 双沿计数 x4 模式	011	高	减	加	加	减
		低	加	减	减	加

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

- CC1S='01'(TIMx_CCMR1 寄存器，TI1FP1 映射到 IC1)
- CC2S='01'(TIMx_CCMR2 寄存器，TI2FP2 映射到 IC2)
- CC1P='0' (TIMx_CCER 寄存器，TI1FP1 不反相，TI1FP1=TI1)
- CC2P='0' (TIMx_CCER 寄存器，TI2FP2 不反相，TI2FP2=TI2)
- SMS='011'(TIMx_SMCR 寄存器，所有的输入均在上升沿和下降沿有效).
- CEN='1'(TIMx_CR1 寄存器，计数器使能)

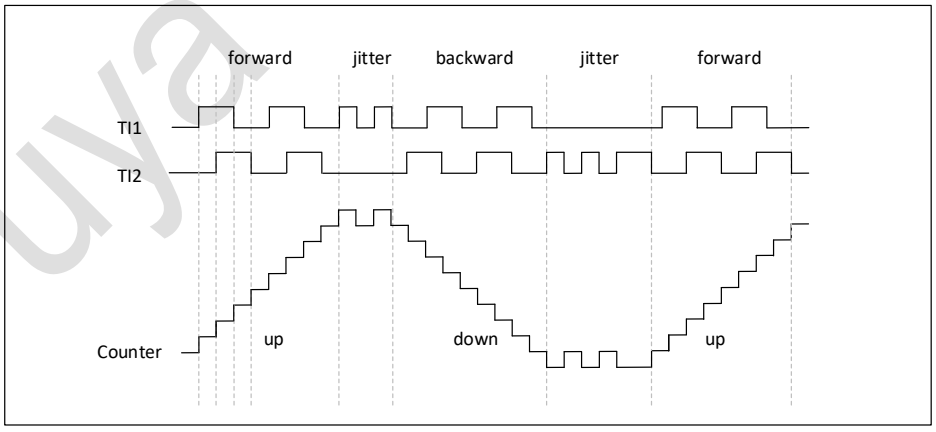


图 20-43 编码器模式下的计数器操作实例

下图为当 IC1FP1 极性反相时计数器的操作实例(CC1P='1'，其他配置与上例相同)

Puya Confidential

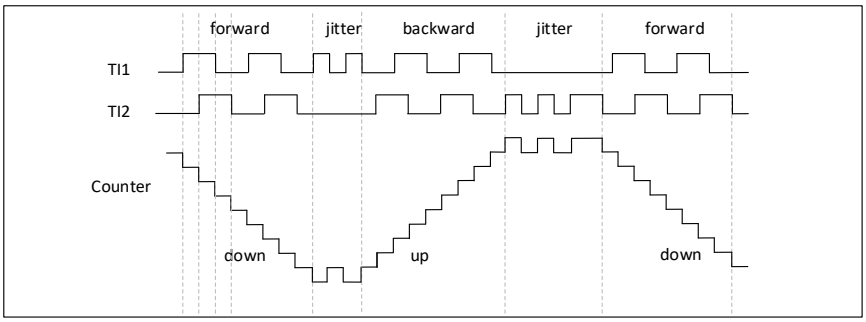


图 20-44 IC1FP1 反相的编码器接口模式实例

下图表示的是不同模式下当转向翻转时的计数值情况：

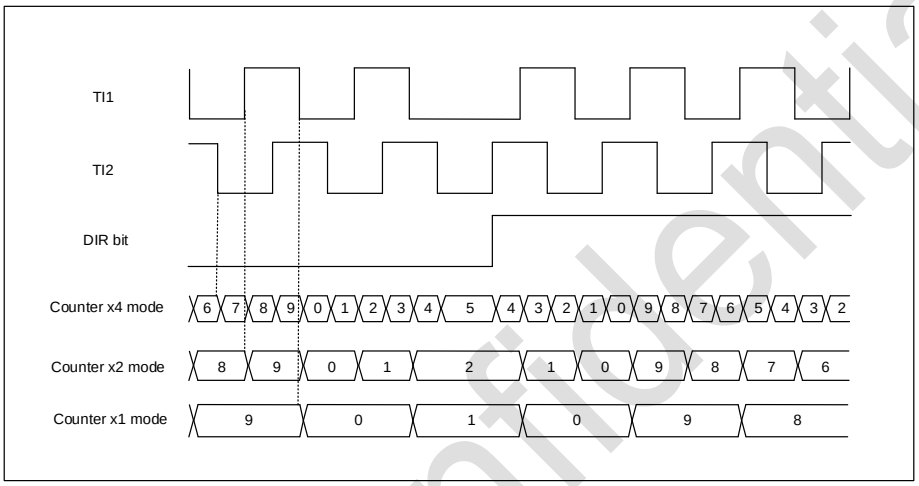


图 20-45 正交编码器计数模式

当定时器配置成编码器接口模式时，可以提供传感器当前位置的信息。通过将第二个定时器配置在捕获模式，可以测量两个编码器事件的间隔，获得动态的信息（速度，加速度，减速度）。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器（捕获信号必须是周期的并且可以由另一个定时器产生）；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

20.3.18 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。

20.3.19 TIMx 定时器和外部触发的同步

TIMx 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式、触发、复位+触发和门控+复位模式。

从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 UDIS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(TIMx_ARR，TIMx_CCRx)都被更新了。

在以下的例子中， TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保

Puya Confidential

持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）。

- 置 TIMx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=00101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）被设置，根据 TIMx_DIER 寄存器中 TIE（中断使能）位和 TDE（DMA 使能）位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

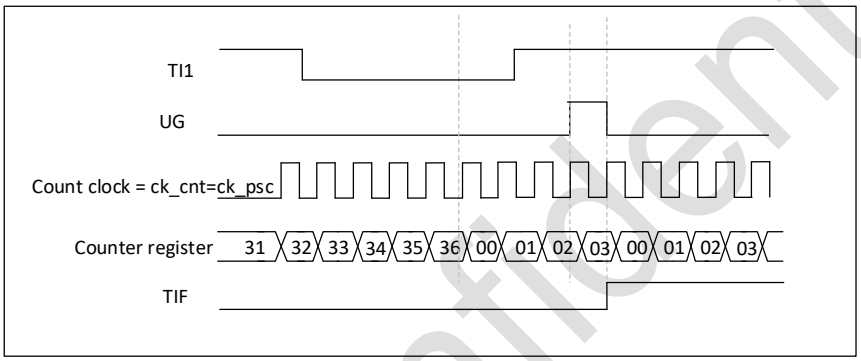


图 20-46 复位模式下的控制电路

从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽（本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标置。

TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。

Puya Confidential

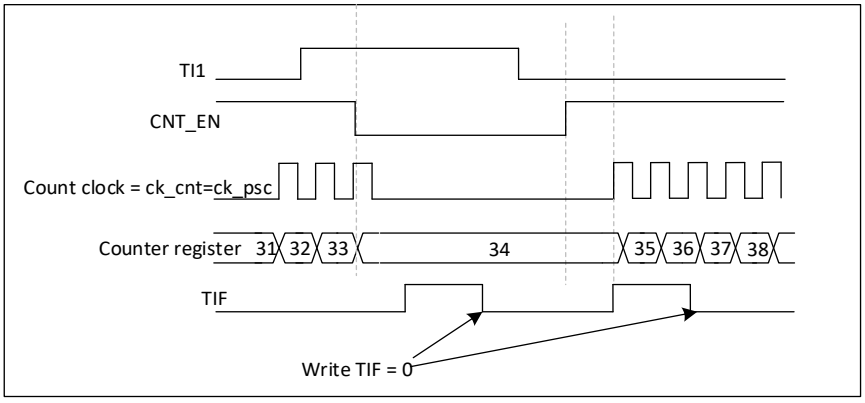


图 20-47 门控模式下的控制电路

从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC2S=01。置 TIMx_CCER 寄存器中 CC2P=1 以确定极性（只检测低电平）。
- 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

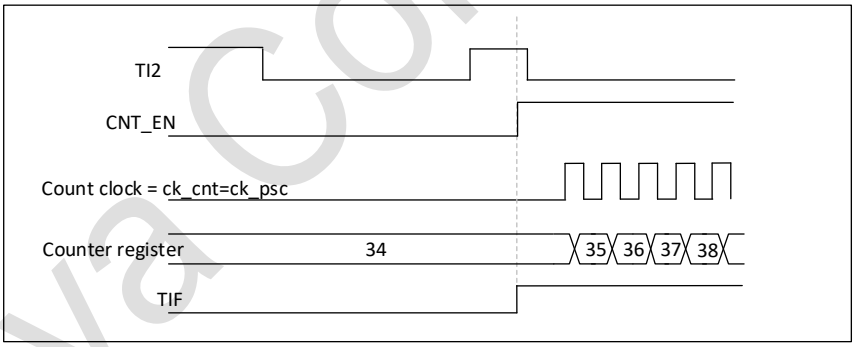


图 20-48 触发器模式下的控制电路

20.3.20 定时器同步

所有 TIMx 定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况。

Puya Confidential

使用一个定时器作为另一个定时器的预分频器

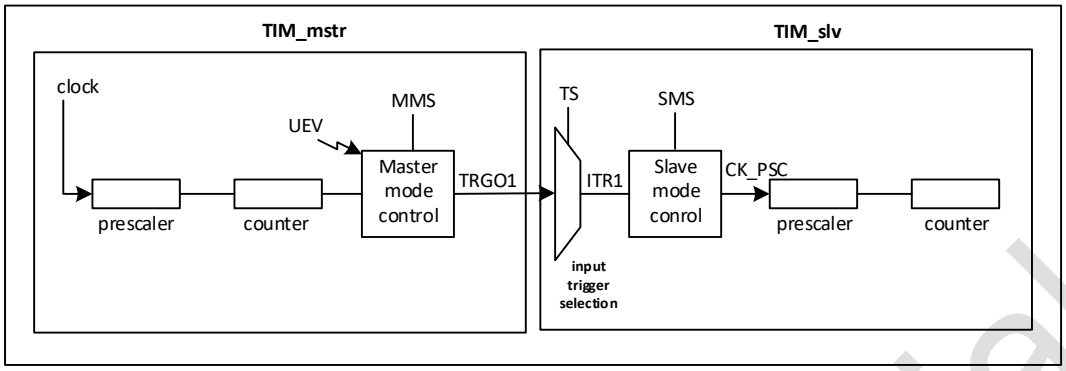


图 20-49 主/从定时器的例子

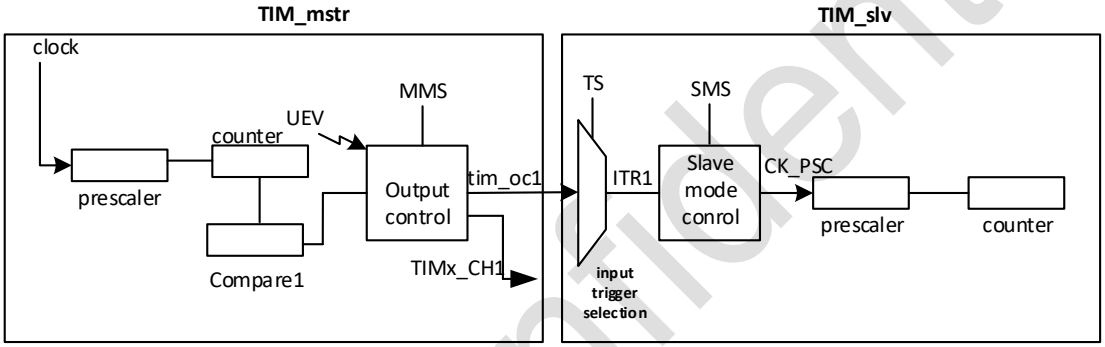


图 20-50 仅有 1 个通道定时器的主从连接示例

仅具有一个通道的计时器（见上图）不具有主模式。但是，tim_oc1 输出信号可以作为从定时器的触发器。tim_oc1 信号脉冲宽度必须编程为目标定时器的至少 2 个时钟周期，以确保从属定时器检测到触发。例如，如果目标定时器时钟比源定时器慢 4 倍，则 OC1 脉冲宽度必须为 8 个时钟周期。

如：可以配置 TIM_mstr 作为 TIM_slv 的预分频器。参考上图，进行下述操作：

- 配置 TIM_mstr 为主模式，它可以在每一个更新事件 UEV 时输出一个周期性的触发信号。在 TIM_mst_CR2 寄存器的 MMS='010'时，每当产生一个更新事件时在 TRGO1 上输出一个上升沿信号。
- 连接 TIM_mstr 的 TRGO1 输出至 TIM_slv，设置 TIM_slv_SMCR 寄存器的 TS='00000'，配置 TIM_slv 为使用 ITR1 作为内部触发的从模式。
- 然后把从模式控制器置于外部时钟模式 1(TIM_slv_SMCR 寄存器的 SMS=111)；这样 TIM_slv 即可由 TIM_mstr 周期性的上升沿(即 TIM_mstr 的计数器溢出)信号驱动。
- 最后，必须设置相应 CEN 位分别启动两个定时器。

注：如果 OCx 已被选中为 TIM_mstr 的触发输出(MMS=1xx)，它的上升沿用于驱动 TIM_slv 的计数器。

使用一个定时器使能另一个定时器

在这个例子中，TIM_slv 的使能由 TIM_mstr 的输出比较控制。只当 TIM_mstr 的 OC1REF 为高时，TIM_slv 才对分频后的内部时钟计数。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3(fCK_CNT=fCK_INT/3)得到。

Puya Confidential

- 配置 TIM_mstr 为主模式，送出它的输出比较参考信号(OC1REF)为触发输出(TIM_mstr_CR2 寄存器的 MMS=100)
- 配置 TIM_mstr 的 OC1REF 波形(TIM_mst_CCMR1 寄存器)
- 配置 TIM_slv 从 TIM_mstr 获得输入触发(TIM_slv_SMCR 寄存器的 TS=00000)
- 配置 TIM_slv 为门控模式(TIM_slv_SMCR 寄存器的 SMS=101)
- 置 TIM_slv_CR1 寄存器的 CEN=1 以使能 TIM_slv
- 置 TIM_mst_CR1 寄存器的 CEN=1 以启动 TIM_mstr

注： TIM_slv 的时钟不与 TIM_mstr 的时钟同步，这个模式只影响 TIM_slv 计数器的使能信号。

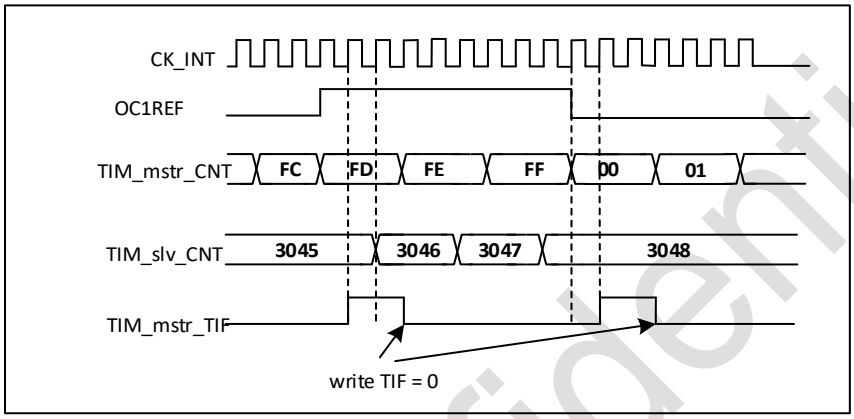


图 20-51 TIM_mstr 的 OC1REF 控制 TIM_slv

在上图的例子中，在 TIM_slv 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动 TIM_mstr 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 TIMx_EGR 寄存器的 UG 位即可复位定时器。

在下一个例子中，需要同步 TIM_mstr 和 TIM_slv。TIM_mstr 是主模式并从 0 开始，TIM_slv 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写‘0’到 TIM_mstr_CR1 的 CEN 位将禁止 TIM_mstr，TIM_slv 随即停止。

- 配置 TIM_mstr 为主模式，送出输出比较 1 参考信号（OC1REF）做为触发输出（TIM_mst_CR2 寄存器的 MMS=100）。
- 配置 TIM_mstr 的 OC1REF 波形（TIM_mstr_CCMR1 寄存器）。
- 配置 TIM_slv 从 TIM_mstr 获得输入触发（TIM_slv_SMCR 寄存器的 TS=00000）
- 配置 TIM_slv 为门控模式（TIM_slv_SMCR 寄存器的 SMS=101）
- 置 TIM_mstr_EGR 寄存器的 UG=‘1’，复位 TIM_mstr。
- 置 TIM_slv_EGR 寄存器的 UG=‘1’，复位 TIM_slv。
- 写‘0XE7’至 TIM_slv 的计数器（TIM_slv_CNT），初始化它为 0xE7。
- 置 TIM_slv_CR1 寄存器的 CEN=‘1’以使能 TIM_slv。
- 置 TIM_mstr_CR1 寄存器的 CEN=‘1’以启动 TIM_mstr。
- 置 TIM_mstr_CR1 寄存器的 CEN=‘0’以停止 TIM_mstr。

Puya Confidential

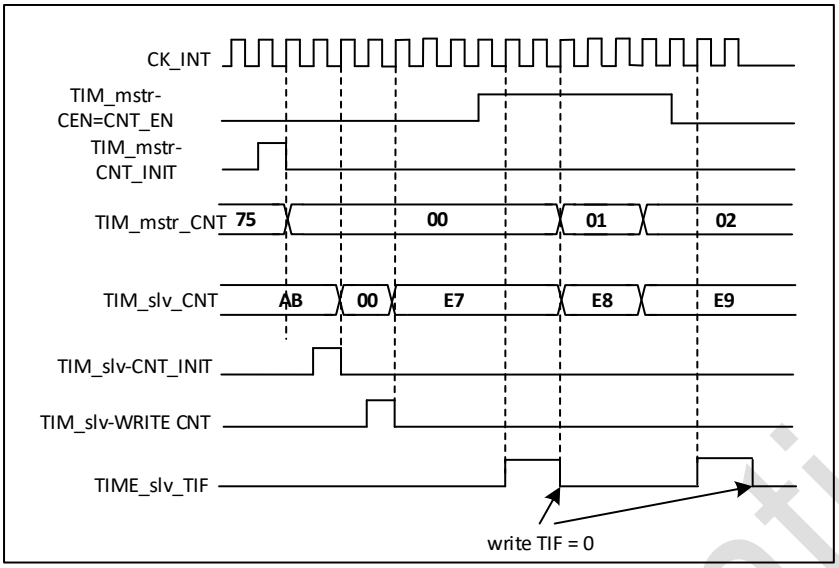


图 20-52 通过使能 TIM_mstr 可以控制 TIM_slv

使用一个定时器去启动另一个定时器

在这个例子中，使用 TIM_mstr 的更新事件使能 TIM_slv。一旦 TIM_mstr 产生更新事件，TIM_slv 即从它当前的数值（可以是非 0）按照分频的内部时钟开始计数。在收到触发信号时，TIM_slv 的 CEN 位被自动地置‘1’，同时计数器开始计数直到写‘0’到 TIM_slv_CR1 寄存器的 CEN 位。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3($f_{CK_CNT}=f_{CK_INT}/3$)。

- 配置 TIM_mstr 为主模式，送出它的更新事件（UEV）做为触发输出（TIM_mstr_CR2 寄存器的 MMS=010）。
- 配置 TIM_mstr 的周期(TIM_mstr_ARR 寄存器)。
- 配置 TIM_slv 从 TIM_mstr 获得输入触发(TIM_slv_SMCR 寄存器的 TS=00000)
- 配置 TIM_slv 为触发模式(TIM_slv_SMCR 寄存器的 SMS=110)
- 置 TIM_mstr_CR1 寄存器的 CEN=1 以启动 TIM_mstr。

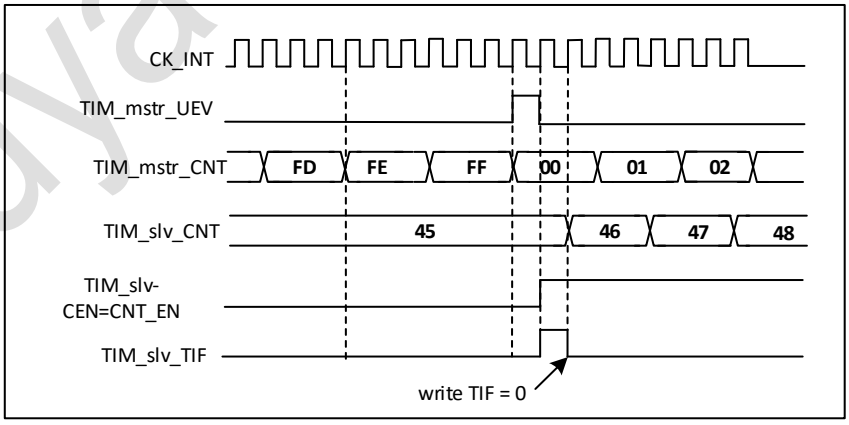


图 20-53 使用 TIM_mstr 的更新触发 TIM_slv

在上一个例子中，可以在启动计数之前初始化两个计数器。下图显示在与 0 相同配置情况下，使用触发模式而不是门控模式（TIM_slv_SMCR 寄存器的 SMS=110）的动作。

Puya Confidential

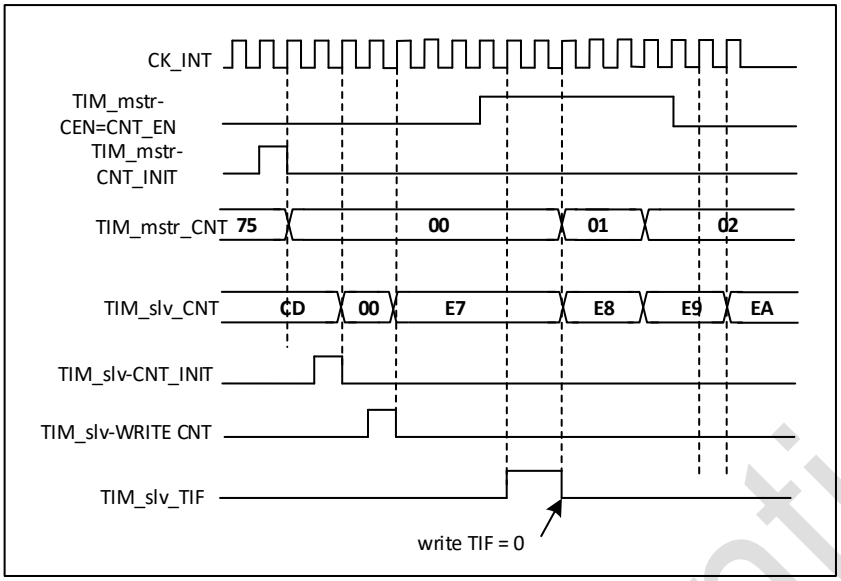


图 20-54 利用 TIM_mstr 的使能触发 TIM_slv

使用一个外部触发同步地启动 2 个定时器

这个例子中当 TIM_mstr 的 TI1 输入上升时使能 TIM_mstr，使能 TIM_mstr 的同时使能 TIM_slv。保证计数器的对齐，TIM_mstr 必须配置为主/从模式(对应 TI1 为从，对应 TIM_slv 为主)：

- 配置 TIM_mstr 为主模式，送出它的使能做为触发输出(TIM_mstr_CR2 寄存器的 MMS=001)。
- 配置 TIM_mstr 为从模式，从 TI1 获得输入触发(TIM_mstr_SMCR 寄存器的 TS=00100)。
- 配置 TIM_mstr 为触发模式（TIM_mstr_SMCR 寄存器的 SMS=110）。
- 配置 TIM_mstr 为主/从模式，TIM_mstr_SMCR 寄存器的 MSM=1。
- 配置 TIM_slv 从 TIM_mstr 获得输入触发（TIM_slv_SMCR 寄存器的 TS=00000）
- 配置 TIM_slv 为触发模式（TIM_slv_SMCR 寄存器的 SMS=110）。

当 TIM_mstr 的 TI1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TIF 标志也同时被设置。

注：在这个例子中，在启动之前两个定时器都被初始化（设置相应的 UG 位），两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器（TIMx_CNT）在定时器间插入一个偏移。下图中能看到主/从模式下在 TIM_mstr 的 CNT_EN 和 CK_PSC 之间有个延迟。

Puya Confidential

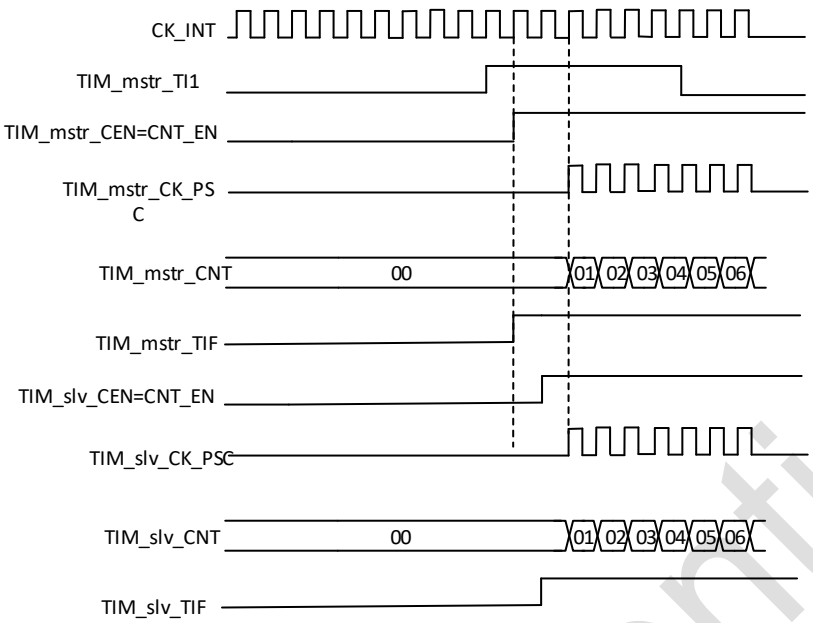


图 20-55 使用 TIM_mstr 的 TI1 输入触发 TIM_mstr 和 TIM_slv

20.3.21 DMA 突发模式

TIMx 定时器当发生单个事件时有生成多个 DMA 请求的能力。主要目的是能够多次重新编程定时器的部分功能而无需软件的开销。但是也可以在固定的间隔时间内读取一行中的几个寄存器。

DMA 控制器的目的地是唯一的并且必须指向虚拟寄存器 TIMx_DMAR.当给定的定时器事件发生，定时器发起一系列 DMA 请求。每个写入 TIMx_DMAR 寄存器的操作实际上是重定向一个定时器寄存器的。

TIMx_DCR 寄存器中的 DBL[4:0]位设置 DMA 突发长度。当对 TIMx_DMR 地址进行读取或写入访问时，定时器会识别突发传输，即传输次数（以半个字或字节为单位）

TIMx_DCR 的 DBA[4:0]位定义了 DMA 传输的 DMA 基地址(当通过 TIMx_DMAR 地址完成读写操作)。DBA 定义为从 TIMx_CR1 寄存器地址开始的偏移量。

例子：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

作为一个例子，定时器 DMA 突发特性是用于发生更新事件时更新 CCRx 寄存器中的内容，通过 DMA 传输半字到 CCRx。

通过以下步骤来完成：

1. 如下配置对应的 DMA 通道：

1) DMA 通道外设地址是 DMAR 寄存器的地址

2) DMA 通道存储地址是 RAM 中的 BUFF 地址，包含了 DMA 传给 CCRx 的数据

3) 传输的数据个数=3

4) 轮询模式关闭

2. 通过配置 DBA 和 DBL 实现配置 DCR 寄存器：DBL=3，DBA=0XE

3. 使能定时器更新 DMA 请求（UDE=1）

Puya Confidential

4. 使能定时器
5. 使能 DMA 通道

这个例子此示例适用于每个 CCRx 寄存器更新一次的情况。如果每个 CCRx 都更新两次，数据的传输个数就需要设置为 6。举例假设 RAM 的 buffer 中包含了 data1，data2，data3，data4，data5 and data6。数据依次传输到 CCRx 中：第一次更新 DMA 请求，data1 传输给 CCR2，data2 传输给 CCR3，data3 传输给 CCR4；第二次更新 DMA 请求，data4 传输给 CCR2，data5 传输给 CCR3，data6 传输给 CCR4。

注：空值可以写入到保留寄存器。

20.3.22 TIM2/3DMA 请求

TIM2/3 可以生成 DMA 请求，如下表所示：

表 20-2 DMA 请求

DMA 请求源	DMA 使能控制位
更新	UDE
比较/捕获 1	CC1DE
比较/捕获 2	CC2DE
比较/捕获 3	CC3DE
比较/捕获 4	CC4DE
触发	TDE

20.3.23 调试模式

当微控制器进入调试模式时，根据 DBG 模块中 DBG_TIMx_STOP 的设置，

TIMx 计数器可以或者继续正常操作，或者停止。详见随后的 DBG 节。

调试模式下的行为可以使用调试支持模块（DBG）中每个定时器的专用配置进行编程

为了安全的考虑，当计数器停止时，输出被禁止（如果 MOE 被复位了）。输出即可被强制到一个无效电平（OSSI=1），或者由 GPIO 控制器接管（OSSI=0），典型的是强制为高阻。

更多的细节，见 DBG 模块描述。

20.3.24 TIM2/3 低功耗模式

模式	说明
Sleep	无效果，外围设备处于活动状态。中断可能导致设备退出睡眠状态
Stop	定时器操作停止，寄存器内容保留。不能产生中断
Standby	定时器已断电，退出待机模式后必须重新初始化。

20.3.25 计数器多点触发 ADC

TIMx 定时器在设置 MTADCE 后，可以开启多点触发 ADC 功能。计数器可以工作在向上计数、向下计数和向上向下计数模式。通过 TIMx_MTADCCR 可以选择在一个计数周期内，计数器的零点（CNT 为 0 时）触发、顶点（ARR 值）触发、通道的 CCRx 和 CCRxH 点触发的任意组合，其中 CCRx 和 CCRxH 区分计数器向上计数过程中触发还是向下计数过程触发。满足触发点后在 timx_multi_trg2adc 接口产生一个时钟长度的脉冲信号。

下面的例子为选择了零点、顶点、上升 CCR1 点和下降 CCR4H 点。

假定计数器选择内部时钟源:

- 选择中央对齐模式：根据 TIMx_CR1 寄存器中的 CMS 选择中央对齐模式。
- 配置 TIMx_ARR 自动重装载寄存器。

Puya Confidential

- 按需配置比较寄存器 TIMx_CCRx[31:0]，置 UG 位产生一个更新事件，更新比较寄存器值。
- 配置触发点 TIMx_CCR1[31:0]和 TIMx_CCR4[31:0]。
- 使能多点触发功能和使能对应触发点，设置 MTADCE，MTZEROE，MTARRE，UMTCCR1E，DMTCCR4HE。
- 置 TIMx_CR1 寄存器中的 CEN=1 使能计数器，如果设置了 UIE 位，则会产生更新中断。

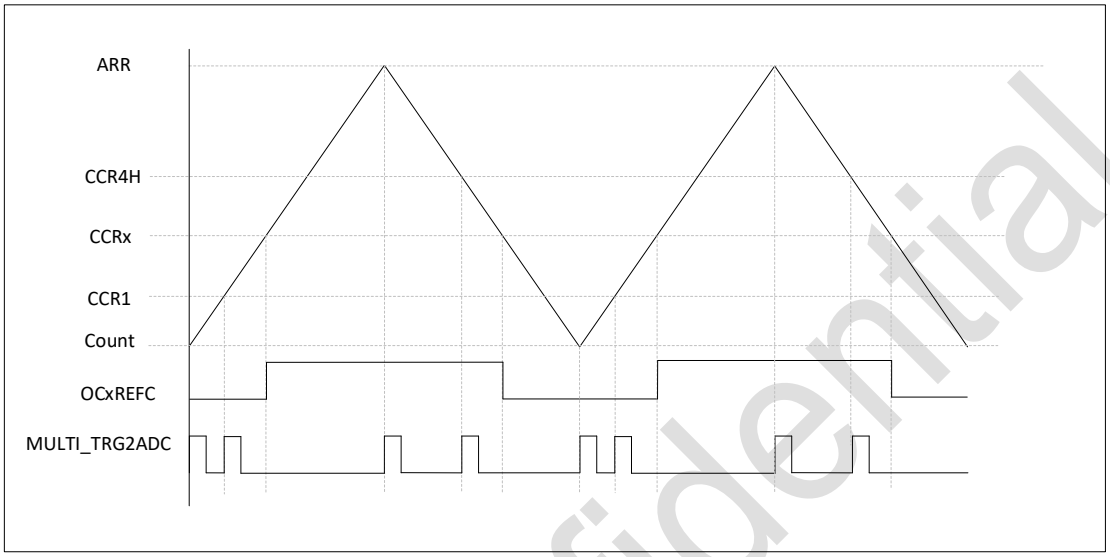


图 20-56 中央对其模式下使能多个触发点

20.4 TIM2/3 中断

中断事件	事件标志	中断使能控制位
更新	UIF	UIE
比较/捕获 1	CC1IF	CC1IE
比较/捕获 2	CC2IF	CC2IE
比较/捕获 3	CC3IF	CC3IE
比较/捕获 4	CC4IF	CC4IE
触发	TIF	TIE

20.5 TIM2/3 寄存器

20.5.1 TIMx 控制寄存器 1（TIMx_CR1）（x=2/3）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:10	Reserved	-	-	保留
9:8	CKD	RW	2’h0	时钟分频因子（Clock division）

Puya Confidential

				这2位定义在定时器时钟(CK_INT)频率和死区时间和死区发生器与数字滤波器(ETR,TIx)所用的采样时钟 (t_{DTS})之间的分频比例。 00：$t_{DTS} = t_{CK_INT}$ 01：$t_{DTS} = 2 \times t_{CK_INT}$ 10：$t_{DTS} = 4 \times t_{CK_INT}$ 11：保留，不要使用这个配置
7	ARPE	RW	0	自动重装载预装载允许位 0：TIMx_ARR 寄存器没有缓冲； 1：TIMx_ARR 寄存器被装入缓冲器。
6:5	CMS	RW	2’h0	选择中央对齐模式 00: 边沿对齐模式。计数器依据方向位(DIR)递增或递减计数。 01：中央对齐模式 1。计数器交替地递增或递减计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 10：中央对齐模式 2。计数器交替地递增和递减计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器递增计数时被设置。 11：中央对齐模式 3。计数器交替地递增和递减计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 **注：**在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	RW	0	方向 (Direction) 0：计数器向上计数； 1：计数器向下计数。 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	OPM	RW	0	单脉冲模式 (One pulse mode) 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件(清除 CEN 位)时，计数器停止。
2	URS	RW	0	更新请求源 (Update request source) 软件通过该位选择 UEV 事件的源 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： − 计数器溢出/下溢 − 设置 UG 位 − 从模式控制器产生的更新 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0	禁止更新 (Update disable) 软件通过该位允许/禁止 UEV 事件的产生

Puya Confidential

				0：允许 UEV。更新(UEV)事件由下述任一事件产生： - 计数器溢出/下溢 - 设置 UG 位 - 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。 (译注：更新影子寄存器) 1：禁止 UEV。不产生更新事件，影子寄存器(ARR、PSC、 CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	RW	0	使能计数器 (Counter enable) 0：禁止计数器； 1：使能计数器。 注: 在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

20.5.2 TIMx 控制寄存器 2（TIMx_CR2） （x=2/3）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	MMS[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						RW									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res	CCPC
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25	MMS[3]	RW	0	详见 MMS[2:0]描述
24:15	Reserved	-	-	保留
14	OIS4	RW	0	输出空闲状态 4(OC4 输出)。参见 OIS1 位。
13	OIS3N	RW	0	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。
12	OIS3	RW	0	输出空闲状态 3(OC3 输出)。参见 OIS1 位。
11	OIS2N	RW	0	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。
10	OIS2	RW	0	输出空闲状态 2(OC2 输出)。参见 OIS1 位。
9	OIS1N	RW	0	输出空闲状态 1(OC1N 输出) (Output Idle state 1) 0：当 MOE=0 时，死区后 OC1N=0； 1：当 MOE=0 时，死区后 OC1N=1。 注：已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、 2 或 3 后，该位不能被修改。
8	OIS1	RW	0	输出空闲状态 1(OC1 输出) (Output Idle state 1) 0：当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=0； 1：当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=1。 注：已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、 2 或 3 后，该位不能被修改。

Puya Confidential

7	TI1S	RW	0	TI1 选择 (TI1 selection) 0： TIMx_CH1 引脚连到 TI1 输入； 1: TIMx_CH1、 TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入。
6:4	MMS	RW	3'h0	主模式选择 (Master mode selection) 用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 0000：复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。 0001：使能 – 计数器使能信号 CNT_EN 可被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMx_SMCR 寄存器中 MSM 位的描述)。 0010: 更新 – 更新事件被选为触发输出(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 0011：比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 0100：比较 – OC1REFC 信号被用于作为触发输出(TRGO)。 0101：比较 – OC2REFC 信号被用于作为触发输出(TRGO)。 0110：比较 – OC3REFC 信号被用于作为触发输出(TRGO)。 0111：比较 – OC4REFC 信号被用于作为触发输出(TRGO)。 1000：编码器时钟输出-编码器时钟信号作为触发输出（TRGO）。 1001-1111：保留 注意： 1.从定时器和 ADC 的时钟必须先被使能以接收主定时器的信号，并在接收时不要改变。 2.若主从定时器不在同一总线上，主模式应该配置为能被从定时器采到的宽度。
3	CCDS	RW	0	捕获/比较的 DMA 选择 (Capture/compare DMA selection) 0：当发生 CCx 事件时，送出 CCx 的 DMA 请求； 1：当发生更新事件时，送出 CCx 的 DMA 请求。
2	CCUS	RW	0	捕获/比较控制更新选择 (Capture/compare control update selection) 0：如果捕获/比较控制位是预装载的(CCPC=1)，只能通过设置 COM 位更新它们； 1：如果捕获/比较控制位是预装载的(CCPC=1)，可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。 注：该位只对具有互补输出的通道起作用。
1	Reserved	-	-	保留
0	CCPC	RW	0	捕获/比较预装载控制位 (Capture/compare preloaded control) 0：CCxE，CCxNE 和 OCxM 位不是预装载的；

Puya Confidential

				1：CCxE，CCxNE 和 OCxM 位是预装载的；设置该位后，它们只在设置了 COM 位后被更新（COMG 位设置或 tim_trgi 上检测到的上升沿，具体取决于 CCUS 位）。 注：该位只对具有互补输出的通道起作用。

20.5.3 TIMx 从模式控制寄存器（TIMx_SMCR）（x=2/3）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	TS[3]	Res.	Res.	Res.	Res.
											RW				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:21	Reserved	-	-	保留
20	TS[3]	RW	1’h0	详见 TS[2:0]描述
19:16	Reserved	-	-	保留
15	ETP	RW	0	外部触发极性（External trigger polarity） 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0： ETR 不反相，高电平或上升沿有效； 1： ETR 被反相，低电平或下降沿有效。
14	ECE	RW	0	外部时钟使能位（External clock enable） 该位启用外部时钟模式 2 0：禁止外部时钟模式 2； 1：使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动。 注 1：设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=00111)具有相同功效。 注 2：下述从模式可以与外部时钟模式 2 同时使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF(TS 位不能是‘00111’)。 注 3：外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。
13:12	ETPS	RW	2’h0	外部触发预分频（External trigger prescaler） 外部触发信号 ETRP 的频率必须最多是 TIMxCLK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。 00：关闭预分频； 01：ETRP 频率除以 2； 10：ETRP 频率除以 4； 11：ETRP 频率除以 8。
11:8	ETF[3:0]	RW	4’h0	外部触发滤波（External trigger filter） 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=4

Puya Confidential

				0011：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8
7	MSM	RW	0	主/从模式（Master/slave mode) 0：无作用； 1：触发输入（TRGI）上的事件被延迟了，以允许在当前定时器（通过 TRGO）与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
6:4	TS	RW	3’h0	触发选择（Trigger selection) 选择用于同步计数器的触发输入。 0000：内部触发 0（ITR0） 0001：内部触发 1（ITR1）（TIM3/TIM4 支持） 0010：内部触发 2（ITR2）（TIM2/TIM4 支持） 0011：内部触发 3（ITR3）（TIM2/TIM3 支持） 0100：TI1 边沿检测（TI1F_ED） 0101：滤波后的定时器输入 1（TI1FP1） 0110：滤波后的定时器输入 2（TI1FP2） 0111：外部触发输入（ETRF） 1000：内部触发 4(ITR4) 1001：内部触发 5(ITR5) 1010：内部触发 6(ITR6) 1011：内部触发 7(ITR7) 1100：内部触发 8(ITR8) 1101：内部触发 9(ITR9) 1110：内部触发 10(ITR10) 1111：内部触发 11(ITR11) 其它：保留 更多有关 ITRx 的细节，参见外设互联章节 注：这些位只能在未用到(如 SMS=0000)时被改变，以避免在改变时产生错误的边沿检测。
3	OCCS	RW	0	OCREF 清除选择位（OCREF clear selection） 这位选择了 OCREF 的清除源。 0：OCREF_CLR_INT 连接 OCREF_CLR 输入 1：OCREF_CLR_INT 连接 ETRF
2:0	SMS	RW	3’h0	从模式选择（Slave mode selection)

Puya Confidential

				当选择了外部信号，触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) 000：关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。 001: 正交编码器模式 1，x2 模式– 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/下计数。 010: 正交编码器模式 2，x2 模式– 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/下计数。 011：正交编码器模式 3，x4 模式–根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。 100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 101: 门控模式 – 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110：触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位)，只有计数器的启动是受控的。 111：外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 注：如果 TI1F_ED 被选为触发输入(TS=00100)时，不要使用门控模式。这是因为， TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。 注：在编码器模式下，不要使用 UEV 作为 tim_trgo 输出信号，（即 mms 不能配置为 010。

20.5.4 TIMx DMA/中断使能寄存器（TIMx_DIER）（x=2/3）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	TERRIE	IERRIE	DIRIE	IDXIE	Res	Res	Res	Res
								RW	RW	RW	RW				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	-	Reserved
23	TERRIE	RW	0	传输错误中断使能（Transition error interrupt enable） 0：传输错误中断不使能 1：传输错误中断使能
22	IERRIE	RW	0	索引错误中断使能（Index error interrupt enable） 0：索引错误中断不使能 1：索引错误中断使能
21	DIRIE	RW	0	方向改变中断使能（Direction change interrupt enable） 0：方向改变中断不使能 1：方向改变中断使能
20	IDXIE	RW	0	索引中断使能（Index interrupt enable） 0：索引中断不使能 1：索引中断使能

Puya Confidential

19:15	Reserved	-	-	Reserved
14	TDE	RW	0	允许触发 DMA 请求（Trigger DMA request enable） 0：禁止触发 DMA 请求； 1：允许触发 DMA 请求。
13	COMDE	RW	0	允许 COM 的 DMA 请求（COM DMA request enable） 0：禁止 COM 的 DMA 请求； 1：允许 COM 的 DMA 请求。
12	CC4DE	RW	0	允许捕获/比较 4 的 DMA 请求(Capture/Compare 4 DMA request enable) 0：禁止捕获/比较 4 的 DMA 请求； 1：允许捕获/比较 4 的 DMA 请求。
11	CC3DE	RW	0	允许捕获/比较 3 的 DMA 请求(Capture/Compare 3 DMA request enable) 0：禁止捕获/比较 3 的 DMA 请求； 1：允许捕获/比较 3 的 DMA 请求。。
10	CC2DE	RW	0	允许捕获/比较 2 的 DMA 请求(Capture/Compare 2 DMA request enable) 0：禁止捕获/比较 2 的 DMA 请求； 1：允许捕获/比较 2 的 DMA 请求。
9	CC1DE	RW	0	允许捕获/比较 1 的 DMA 请求(Capture/Compare 1 DMA request enable) 0：禁止捕获/比较 1 的 DMA 请求； 1：允许捕获/比较 1 的 DMA 请求。。
8	UDE	RW	0	允许更新的 DMA 请求（Update DMA request enable) 0：禁止更新的 DMA 请求； 1：允许更新的 DMA 请求。
7	BIE	RW	0	允许刹车中断（Break interrupt enable) 0：禁止刹车中断； 1：允许刹车中断。
6	TIE	RW	0	触发中断使能（Trigger interrupt enable) 0：禁止触发中断； 1：使能触发中断。
5	COMIE	RW	0	允许 COM 中断（COM interrupt enable) 0：禁止 COM 中断； 1：允许 COM 中断。
4	CC4IE	RW	0	允许捕获/比较 4 中断（Capture/Compare 4 interrupt enable) 0：禁止捕获/比较 4 中断； 1：允许捕获/比较 4 中断。
3	CC3IE	RW	0	允许捕获/比较 3 中断（Capture/Compare 3 interrupt enable) 0：禁止捕获/比较 3 中断； 1：允许捕获/比较 3 中断。
2	CC2IE	RW	0	允许捕获/比较 2 中断（Capture/Compare 2 interrupt enable) 0：禁止捕获/比较 2 中断； 1：允许捕获/比较 2 中断。
1	CC1IE	RW	0	允许捕获/比较 1 中断（Capture/Compare 1 interrupt enable) 0：禁止捕获/比较 1 中断； 1：允许捕获/比较 1 中断
0	UIE	RW	0	允许更新中断（Update interrupt enable)

Puya Confidential

				0：禁止更新中断； 1：允许更新中断。

20.5.5 TIMx 状态寄存器（TIMx_SR）（x=2/3）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	IC4IF	IC3IF	IC2IF	IC1IF	IC4IR	IC3IR	TERRF	IERRF	DIRF	IDXF	IC2IR	IC1IR	Res.	Res.
		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1F	UIF
		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	RC_W0

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29	IC4IF	RC_W0	0	下降沿捕获 4 标志 参见 IC1IF 描述。
28	IC3IF	RC_W0	0	下降沿捕获 3 标志 参见 IC1IF 描述。
27	IC2IF	RC_W0	0	下降沿捕获 2 标志 参见 IC1IF 描述。
26	IC1IF	RC_W0	0	下降沿捕获 1 标志 仅当相应的通道被配置为输入捕获时且由下降沿触发捕获事件，该标记可由硬件置 1。它由软件清‘0’或通过读 TIMx_CCR1 清‘0’。 0：无重复捕获产生； 1：发生下降沿捕获事件。
25	IC4IR	RC_W0	0	上升沿捕获 4 标志 参见 IC1IR 描述。
24	IC3IR	RC_W0	0	上升沿捕获 3 标志 参见 IC1IR 描述。
23	TERRF	RC_W0	0	传输错误中断标志（Transition error interrupt flag） 当编码器模式中出现传输错误，该标志由硬件置位。可通过软件清零或者写清零 0：未检测到编码器传输错误 1：检测到编码器传输错误
22	IERRF	RC_W0	0	索引错误中断标志（Index error interrupt flag） 当检测到索引出现错误，该标志由硬件置位。可通过软件清零或者写清零
21	DIRF	RC_W0	0	方向改变中断标志（Direction change interrupt flag） 当编码器模式中出现方向改变（TIMx_CR 中的 DIR 位改变），该标志由硬件置位。可通过软件清零或者写清零 0：未检测到方向改变

Puya Confidential

				1：检测到方向改变
20	IDXF	RC_W0	0	索引中断标志（Index interrupt flag） 当检测到索引事件，该标志由硬件置位。可通过软件清零或者写清零 0：无索引事件发生 1：有索引事件发生
19	IC2IR	RC_W0	0	上升沿捕获 2 标志 参见 IC1IR 描述。
18	IC1IR	RC_W0	0	上升沿捕获 1 标志 仅当相应的通道被配置为输入捕获时且由上升沿触发捕获事件，该标记可由硬件置 1。它由软件清‘0’或通过读 TIMx_CCR1 清‘0’。 0：无重复捕获产生； 1：发生上升沿捕获事件。
17:14	Reserved	-	-	保留
13	SBIF	RC_W0	0	系统刹车中断标志（System break interrupt flag） 当系统刹车输入一旦有效，该标志信号会硬件置位。当系统刹车输入无效时可通过软件清零 该标志必须复位来重启 PWM 操作 0：无刹车事件发生 1：系统刹车输入处于有效。如果 TIMx_DIER 寄存器的 BIE 位=1 则会产生中断
12	CC4OF	RC_W0	0	捕获/比较 4 重复捕获标记（Capture/Compare 4 overcapture flag） 参见 CC1OF 描述。
11	CC3OF	RC_W0	0	捕获/比较 3 重复捕获标记（Capture/Compare 3 overcapture flag） 参见 CC1OF 描述。
10	CC2OF	RC_W0	0	捕获/比较 2 重复捕获标记（Capture/Compare 2 overcapture flag） 参见 CC1OF 描述。
9	CC1OF	RC_W0	0	捕获/比较 1 重复捕获标记（Capture/Compare 1 overcapture flag） 仅当相应的通道被配置为输入捕获时，该标记可由硬件置 1。写 0 可清除该位。 0：无重复捕获产生； 1：计数器的值被捕获到 TIMx_CCR1 寄存器时， CC1IF 的状态已经为‘1’。
8	B2IF	RC_W0	0	刹车 2 中断标记（Break2 interrupt flag） 一旦刹车输入有效，由硬件对该位置‘1’。如果刹车输入无效，则该位可由软件清‘0’。 0：无刹车事件产生； 1：刹车输入上检测到有效电平。如果 TIM_DIER 的 BIE=1 则产生中断
7	BIF	RC_W0	0	刹车中断标记（Break interrupt flag） 一旦刹车输入有效，由硬件对该位置‘1’。如果刹车输入无效，则该位可由软件清‘0’。

Puya Confidential

				0：无刹车事件产生； 1：刹车输入上检测到有效电平。如果 TIM_DIER 的 BIE=1 则产生中断
6	TIF	RC_W0	0	触发器中断标记 (Trigger interrupt flag) 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿)时由硬件对该位置‘1’。它由软件清‘0’。 0：无触发器事件产生； 1：触发中断等待响应。
5	Reserved	-	-	保留
4	CC4IF	RC_W0	0	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	RC_W0	0	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	RC_W0	0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	RC_W0	0	捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag) 如果通道 CC1 配置为输出模式： 0：无匹配发生； 1： TIMx_CNT 的值与 TIMx_CCR1 的值匹配。 当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下， CC1IF 位变高 如果通道 TI1 配置为输入模式： 当捕获事件发生时该位由硬件置‘1’，它由软件清‘0’或通过读 TIMx_CCR1 清‘0’。 0：无输入捕获产生； 1：计数器值已被捕获(拷贝)至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。
0	UIF	RC_W0	0	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置‘1’。它由软件清‘0’。 0：无更新事件产生； 1：更新中断等待响应。当寄存器被更新时该位由硬件置’1’： − 若 TIMx_CR1 寄存器的 UDIS=0，当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 − 若 TIMx_CR1 寄存器的 URS=0、 UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。 − 若 TIMx_CR1 寄存器的 URS=0、 UDIS=0，当计数器 CNT 被触发事件重新初始化时。 (参考从模式控制寄存器(TIMx_SMCR))。

20.5.6 TIMx 事件产生寄存器（TIMx_EGR）（x=2/3）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	Reserved
8	B2G	W	0	产生刹车 2 事件 (Break generation) 该位由软件置'1'，用于产生一个刹车事件，由硬件自动清'0'。 0：无动作； 1：产生一个刹车 2 事件。此时 MOE=0、 B2IF=1
7	BG	W	0	产生刹车事件 (Break generation) 该位由软件置'1'，用于产生一个刹车事件，由硬件自动清'0'。 0：无动作； 1: 产生一个刹车事件。此时 MOE=0、 BIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
6	TG	W	0	产生触发事件 (Trigger generation) 该位由软件置'1'，用于产生一个触发事件，由硬件自动清'0'。 0：无动作； 1： TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
5	COMG	W	0	捕获/比较事件，产生控制更新 该位由软件置'1'，由硬件自动清'0'。 0：无动作； 1：当 CCPC=1，允许更新 CCxE、 CCxNE、 OCxM 位。 注： 该位只对拥有互补输出的通道有效。
4	CC4G	W	0	产生捕获/比较 4 事件 (Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	W	0	产生捕获/比较 3 事件 (Capture/Compare 3 generation) 参考 CC1G 描述。
2	CC2G	W	0	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	W	0	产生捕获/比较 1 事件 (Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0：无动作； 1：在通道 1 上产生一个捕获/比较事件： 若通道 1 配置为输出： 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 1 配置为输入： 当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	W	0	产生更新事件 (Update generation) 该位由软件置'1'，由硬件自动清'0'。 0：无动作；

Puya Confidential

				1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0'；若 DIR=1(向下计数)则计数器取 TIMx_ARR 的值。

20.5.7 TIMx 捕获/比较模式控制寄存器 1（TIMx_CCMR1）（x=2/3）

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式或输出比较模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OC2CE	OC2M[2:0]			OC2PE	CO2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

■ 输出比较模式

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	OC2CE	RW	0	输出比较 2 清零使能（Output Compare 2 clear enable)
14:12	OC2M	RW	3'h0	输出比较 2 模式（Output Compare 2 mode)
11	OC2PE	RW	0	输出比较 2 预装载使能（Output Compare 2 preload enable)
10	OC2FE	RW	0	输出比较 2 快速使能（Output Compare 2 fast enable)
9:8	CC2S	RW	2'h0	捕获/比较 2 选择。（Capture/Compare 2 selection） 该位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC2 通道被配置为输出； 01：CC2 通道被配置为输入， IC2 映射在 TI2 上； 10：CC2 通道被配置为输入， IC2 映射在 TI1 上； 11：CC2 通道被配置为输入， IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7	OC1CE	RW	0	输出比较 1 清'0'使能（Output Compare 1 clear enable) 0：OC1REF 不受 ETRF 输入的影响； 1：一旦检测到 ETRF 输入高电平，清除 OC1REF=0。
6:4	OC1M	RW	3'h0	输出比较 1 模式(OutputCompare1mode) 这些位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用； 001：匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1(TIMx_CCR1)相同时，强制 OC1REF 为高。 010：匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1(TIMx_CCR1)相同时，强制 OC1REF 为低。 011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 100：强制为无效电平。强制 OC1REF 为低。

Puya Confidential

				101：强制为有效电平。强制 OC1REF 为高。 110：PWM 模式 1 - 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平(OC1REF=0)，否则为有效电平(OC1REF=1)。 111：PWM 模式 2 - 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。
3	OC1PE	RW	0	输出比较 1 预装载使能 (Output Compare 1 preload enable) 0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。 1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作， TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。
2	OC1FE	RW	0	输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快输出对触发输入事件的响应。必须在单脉冲模式（TIMx_CR1 寄存器的 OPM 位置 1）中使用，当触发到来时实现脉冲快速输出。 0：根据计数器与 CCR1 的值， CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此， OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。仅能用于 PWM 模式 1 和 PWM 模式 2
1:0	CC1S	RW	2’h0	捕获/比较 1 选择。 (Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC1 通道被配置为输出； 01：CC1 通道被配置为输入， IC1 映射在 TI1 上； 10：CC1 通道被配置为输入， IC1 映射在 TI2 上； 11：CC1 通道被配置为输入， IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

■ 输入捕获模式

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:12	IC2F	RW	4’h0	输入捕获 2 滤波器（Input capture 2 filter)
11:10	IC2PSC	RW	2’h0	输入/捕获 2 预分频器（Input capture 2 prescaler)
9:8	CC2S	RW	2’h0	捕获/比较 2 选择（Capture/Compare 2 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC2 通道被配置为输出； 01：CC2 通道被配置为输入，IC2 映射在 TI2 上； 10：CC2 通道被配置为输入，IC2 映射在 TI1 上；

Puya Confidential

				11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	RW	4'h0	输入捕获 1 滤波器（Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8
3:2	IC1PSC	RW	2'h0	输入/捕获 1 预分频器（Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。 一旦 CC1S=00，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01：每 2 个事件触发一次捕获； 10：每 4 个事件触发一次捕获； 11：每 8 个事件触发一次捕获。
1:0	CC1S	RW	2'h0	捕获/比较 1 选择（Capture/Compare 1 Selection） 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC1 通道被配置为输出； 01：CC1 通道被配置为输入，IC1 映射在 TI1 上； 10：CC1 通道被配置为输入，IC1 映射在 TI2 上； 11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

Puya Confidential

20.5.8 TIMx 捕获/比较模式控制寄存器 2（TIMx_CCMR2）（x=2/3）

偏移地址：0x1C

复位值：0x0000 0000

参看以上 CCMR1 寄存器的描述

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	CO4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

- 输出比较模式

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	OC4CE	RW	0	输出比较 4 清零使能 (Output Compare 4 clear enable)
14:12	OC4M	RW	3’h0	输出比较 4 模式 (Output Compare 4 mode)
11	OC4PE	RW	0	输出比较 4 预装载使能 (Output Compare 4 preload enable)
10	OC4FE	RW	0	输出比较 4 快速使能 (Output Compare 4 fast enable)
9:8	CC4S	RW	2’h0	捕获/比较 4 选择。 (Capture/Compare 4 selection) 该 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC4 通道被配置为输出； 01：CC4 通道被配置为输入，IC4 映射在 TI4 上； 10：CC4 通道被配置为输入，IC4 映射在 TI3 上； 11：CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7	OC3CE	RW	0	输出比较 3 清’0’使能 (Output Compare 1clear enable)
6:4	OC3M	RW	3’h0	输出比较 3 模式
3	OC3PE	RW	0	输出比较 3 预装载使能 (Output Compare 3 preload enable)
2	OC3FE	RW	0	输出比较 3 快速使能 (Output Compare 3 fast enable)
1:0	CC3S	RW	2’h0	捕获/比较 3 选择。 (Capture/Compare 3 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC3 通道被配置为输出； 01：CC3 通道被配置为输入，IC3 映射在 TI3 上； 10：CC3 通道被配置为输入，IC3 映射在 TI4 上； 11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIMx_CCER 寄存器的 CC3E=0)才是可写的。

- 输入捕获模式

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:12	IC4F	RW	4’h0	输入捕获 4 滤波器 (Input capture 4 filter)
11:10	IC4PSC	RW	2’h0	输入/捕获 4 预分频器 (Input capture 4 prescaler)
9:8	CC4S	RW	2’h0	捕获/比较 4 选择 (Capture/Compare 4 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择：

Puya Confidential

				00：CC4 通道被配置为输出； 01：CC4 通道被配置为输入，IC4 映射在 TI4 上； 10：CC4 通道被配置为输入，IC4 映射在 TI3 上； 11：CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注：CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7:4	IC3F	RW	4'h0	输入捕获 3 滤波器 (Input capture 3 filter)
3:2	IC3PSC	RW	2'h0	输入/捕获 3 预分频器 (Input capture 3 prescaler)
1:0	CC3S	RW	2'h0	捕获/比较 3 选择 (Capture/Compare 3 Selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC3 通道被配置为输出； 01：CC3 通道被配置为输入，IC3 映射在 TI3 上； 10：CC3 通道被配置为输入，IC3 映射在 TI4 上； 11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注：CC3S 仅在通道关闭时(TIMx_CCER 寄存器的 CC3E=0)才是可写的。

20.5.9 TIMx 捕获/比较使能寄存器（TIMx_CCER）（x=2/3）

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CC4NP	CC4NE	CC4P	CC4E	Res	Res	CC3P	CC3E	Res	Res.	CC2P	CC2E	Res	Res.	CC1P	CC1E
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	CC4NP	RW	0	输入/捕获 4 互补输出极性 (Capture/Compare 4 output polarity) 参考 CC1NP 的描述。
14	CC4NE	RW	0	输入/捕获 4 互补输出使能 (Capture/Compare 4 output enable) 参考 CC1NE 的描述。
13	CC4P	RW	0	输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	RW	0	输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11:10	Reserved	RW	0	保留，当前 bit 无实际意义
9	CC3P	RW	0	输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	RW	0	输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7:6	Reserved	RW	0	保留，当前 bit 无实际意义

Puya Confidential

5	CC2P	RW	0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3:2	Reserved	RW	0	保留，当前 bit 无实际意义
1	CC1P	RW	0	输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出： 0：OC1 高电平有效； 1：OC1 低电平有效。 CC1 通道配置为输入： CC1NP/CC1P 位选择作为触发或捕获信号的 TI1FP1 和 TI2FP1 的极性。 00：不反相/上升沿： TIxFP1 上升沿有效（捕获、复位模式下触发、外部时钟或触发模式下）； TIxFP1 不反相（门控模式、编码器模式）。 01：反相/下降沿： TIxFP1 下降沿有效（捕获、复位模式下触发、外部时钟或触发模式下）； TIxFP1 反相（门控模式、编码器模式）。 10：保留，不要使用这个配置。 11：不反相/双沿 TIxFP1 上升和下降沿都有效（捕获、复位模式下触发、外部时钟或触发模式下）； TIxFP1 不反相（门控模式）。这个配置不能应用于编码器模式下。
0	CC1E	RW	0	输入/捕获 1 输出使能 (Capture/Compare 1 output enable) CC1 通道配置为输出： 0: 关闭 - OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、 OSSR、 OIS1、 OIS1N 和 CC1NE 位的值。 1: 开启 - OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、 OSSI、 OSSR、OIS1、 OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。 0：捕获禁止； 1：捕获使能。 注： 对于互补输出通道，这一位是预载的。如果 TIMx_CR2 寄存器中的 CCPC 位被设置，那么 CC1E 的实际有效位只有在 com 事件发生时才会加载预载值。

20.5.10 TIMx 计数器（TIMx_CNT）（x=2）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															

Puya Confidential

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	CNT	RW	32'h0	计数器的值 (Counter value)

20.5.11 TIMx 计数器（TIMx_CNT）（x=3）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT1[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CNT	RW	16'h0	计数器的值 (Counter value)

20.5.12 TIMx 预分频器（TIMx_PSC）（x=2/3）

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSC[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	PSC	RW	16'h0	预分频器的值 (Prescaler value) 计数器的时钟频率(CK_CNT)等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被 TIM_EGR 的 UG 位清'0'或被工作在复位模式的从控制器清'0'。

20.5.13 TIMx 自动重装载寄存器（TIMx_ARR）（x=2）

偏移地址：0x2C

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ARR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	ARR	RW	32'hFFFFFFFF	自动重装载的值 (Prescaler value) 自动重装载的值为空时，计数器不工作。 该寄存器保持自动加载值

Puya Confidential

20.5.14 TIMx 自动重装载寄存器（TIMx_ARR）（x=3）

偏移地址：0x2C

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	ARR[15:0]	RW	16'hFFFF	自动重装载的值（Prescaler value） 自动重装载的值为空时，计数器不工作。 该寄存器保持自动加载值

20.5.15 TIMx 捕获/比较寄存器 1（TIMx_CCR1）（x=2）

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	CCR1	RW	32'h0	捕获/比较通道 1 的值（Capture/Compare 1 value） 若 CC1 通道配置为输出： CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIMx_CCMR1 寄存器(OC1PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。

20.5.16 TIMx 捕获/比较寄存器 1（TIMx_CCR1）（x=3）

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1_H[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	CCR1_H[31:16]	RW	0	输出比较 1 寄存器的值 若 CC1 通道配置为中央对齐 PWM 模式输出： CCR1_H 包含了装入向下计数比较 1 寄存器的值（预装载值）。
15:0	CCR1	RW	16'h0	捕获/比较通道 1 的值（Capture/Compare 1 value） 若 CC1 通道配置为输出：

Puya Confidential

				CCR1 包含了装入当前捕获/比较 1 寄存器的值（预装载值）。 如果在 TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。

20.5.17 TIMx 捕获/比较寄存器 2（TIMx_CCR2）（x=2）

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR2[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	CCR2	RW	32’h0	捕获/比较通道 2 的值（Capture/Compare 2 value) 若 CC2 通道配置为输出： CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIMx_CCMR2 寄存器(OC2PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。

20.5.18 TIMx 捕获/比较寄存器 2（TIMx_CCR2）（x=3）

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2_H[31:16]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCR2[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	CCR2_H[31:16]	RW	0	输出比较 2 寄存器的值 若 CC2 通道配置为中央对齐 PWM 模式输出： CCR2_H 包含了装入向下计数比较 2 寄存器的值（预装载值）。
15:0	CCR2	RW	16’h0	捕获/比较通道 2 的值（Capture/Compare 2 value) 若 CC2 通道配置为输出： CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIMx_CCMR2 寄存器(OC2PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。

Puya Confidential

20.5.19 TIMx 捕获/比较寄存器 3（TIMx_CCR3）（x=2）

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	CCR3	RWO	32’h0	捕获/比较通道 3 的值（Capture/Compare 3 value） 若 CC3 通道配置为输出： CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。 如果在 TIMx_CCMR3 寄存器(OC3PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC3 端口上产生输出信号。

20.5.20 TIMx 捕获/比较寄存器 3（TIMx_CCR3）（x=3）

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3_H[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	CCR3_H[31:16]	RW	0	输出比较 3 寄存器的值 若 CC3 通道配置为中央对齐 pwm 模式输出： CCR3_H 包含了装入向下计数比较 3 寄存器的值（预装载值）。
15:0	CCR3	RW	16’h0	捕获/比较通道 3 的值（Capture/Compare 3 value） 若 CC3 通道配置为输出： CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。 如果在 TIMx_CCMR3 寄存器(OC3PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC3 端口上产生输出信号。

20.5.21 TIMx 捕获/比较寄存器 4（TIMx_CCR4）（x=2）

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Puya Confidential

CCR14[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	CCR4	RW	32'h0	捕获/比较通道 4 的值（Capture/Compare 4 value) 若 CC4 通道配置为输出： CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。 如果在 TIMx_CCMR4 寄存器(OC4PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 4 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC4 端口上产生输出信号。

20.5.22 TIMx 捕获/比较寄存器 4（TIMx_CCR4）（x=3）

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4_H[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	CCR4_H[31:16]	RW	0	输出比较 4 寄存器的值 若 CC4 通道配置为中央对齐 pwm 模式输出： CCR4_H 包含了装入向下计数比较 4 寄存器的值（预装载值）。
15:0	CCR4	RW	16'h0	捕获/比较通道 4 的值（Capture/Compare 4 value) 若 CC4 通道配置为输出： CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。 如果在 TIMx_CCMR4 寄存器(OC4PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 4 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC4 端口上产生输出信号。

20.5.23 TIMx 刹车和死区寄存器（TIMx_BDTR）（x=2/3）

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

注释：根据锁定设置，BKBID/BK2BID/BK2P/BK2E/BK2F/BKF/AOE/BKP/BKE/OSSI、OSSR 和 DTG[7:0]位均可被写保护，有必要在第一次写入 TIMx_BDTR 寄存器时对它们进行配置。

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29	BK2BID	RW	0	刹车 2 双向性（Break2 bidirectional） 参考 BKBID 描述

Puya Confidential

28	BKBID	RW	0	刹车双向性（Break bidirectional） 0：刹车输入 BRK 在输入模式中 1：刹车输入 BRK 在双向模式中 在双向模式中（BKBID 设置为 1），刹车输入被既配置为输入模式也是开漏输出模式。任何有效的刹车事件会将刹车输入电平拉低，向外部设备指示内部发生了刹车事件 注: 一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。 注：任何对该位的写操作，都会延迟一个 APB 时钟生效。
27	BK2DSRM	RW	0	刹车 2 解除（Break2 disarm） 参考 BKDSRM 描述
26	BKDSRM	RW	0	刹车解除（Break disarm） 0：刹车输入 BRK 启用 1：刹车输入 BRK 解除 当没有刹车源时该位会被硬件清零 BKDSRM 位必须由软件置位来释放双向输出控制（开漏输出高阻状态），然后轮询它直到被硬件复位，表示故障状态已经消失。 注：任何对该位的写操作，都会延迟一个 APB 时钟生效。
25	BK2P	RW	0	刹车 2 极性（Break 2 polarity） 0：刹车输入 BRK2 有效电平为低电平 1：刹车输入 BRK2 有效电平为高电平 注: 一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。 注：任何对该位的写操作，都会延迟一个 APB 时钟生效。
24	BK2E	RW	0	刹车 2 使能（Break 2 enable） 该位使能刹车 2 的保护 0：刹车 2 功能关闭 1：刹车功能开启 注：刹车 2 必须在 OSSR=OSSI=1 时才能使用 注: 一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。 注：任何对该位的写操作，都会延迟一个 APB 时钟生效。
23:20	BK2F[3:0]	RW	4’h0	刹车 2 滤波（Break 2 filter） 这些位定义了对 BRK2 信号采样的频率和对 BRK2 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，BRK2 是异步的 0001：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6

Puya Confidential

				1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$, N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$, N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8 注: 一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。
19:16	BKF[3:0]	RW	4'h0	刹车滤波（Break filter） 这些位定义了对 BRK 信号采样的频率和对 BRK 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，BRK2 是异步的 0001：采样频率 $f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK_INT}$， N=4 0011：采样频率 $f_{SAMPLING}=f_{CK_INT}$， N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$, N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$, N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8 注: 一旦 LOCK 级别 1(TIMx_BDTR 寄存器中的 LOCK 位)被设置，则该位不能被修改。
15	MOE	RW	0	主输出使能 (Main output enable) 一旦刹车输入有效，该位被硬件异步清'0'。根据 AOE 位的设置值，该位可以由软件清'0'或被自动置 1。它仅对配置为输出的通道有效。 0：禁止 OC 和 OCN 输出或强制为空闲状态； 1: 如果设置了相应的使能位(TIMx_CCER 寄存器的 CCxE、 CCxNE 位)，则开启 OC 和 OCN 输出。 有关 OC/OCN 使能的细节，参见 TIMx 捕获/比较使能寄存(TIMx_CCER)。
14	AOE	RW	0	自动输出使能 (Automatic output enable) 0： MOE 只能被软件置'1'； 1： MOE 能被软件置'1'或在下一个更新事件被自动置'1'(如果刹车输入无效)。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。
13	BKP	RW	0	刹车输入极性 (Break polarity)

Puya Confidential

				0：刹车输入低电平有效； 1：刹车输入高电平有效。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。 注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
12	BKE	RW	0	刹车功能使能 (Break enable) 0：禁止刹车输入(BRK 及 CCS 时钟失效事件)； 1：开启刹车输入(BRK 及 CCS 时钟失效事件)。 注：当设置了 LOCK 级别 1 时(TIMx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。 注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
11	OSSR	RW	0	运行模式下"关闭状态"选择 (Off-state selection for Run mode) 该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。 参考 OC/OCN 使能的详细说明(TIMx 捕获/比较使能寄存器(TIMx_CCER))。 0：当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，首先开启 OC/OCN 并输出无效电平，然后置 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
10	OSSI	RW	0	空闲模式下"关闭状态"选择 (Off-state selection for Idle mode) 该位用于当 MOE=0 且通道设为输出时。 参考 OC/OCN 使能的详细说明(TIMx 捕获/比较使能寄存器(TIMx_CCER))。 0：当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1， OC/OCN 首先输出其空闲电平，然后 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
9:8	LOCK	RW	2'h0	锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。 00：锁定关闭，寄存器无写保护； 01: 锁定级别 1，不能写入 TIMx_BDTR 寄存器的 DTG、BKBID、BK2BID、BKE、 BKP、 AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位； 10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出， CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCxNP 位)以及 OSSR/OSSI 位； 11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出， CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位)； 注: 在系统复位后，只能写一次 LOCK 位，一旦写入 TIMx_BDTR 寄存器，则其内容冻结直至复位。
7:0	DTG	RW	8'h0	死区发生器设置 (Dead-time generator setup)

Puya Confidential

				这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间： DTG[7:5]=0xx => DT=DTG[7:0] × Tdtg， Tdtg = TDTS； DTG[7:5]=10x => DT=(64+DTG[5:0]) × Tdtg， Tdtg = 2 × TDTS； DTG[7:5]=110 => DT=(32+DTG[4:0]) × Tdtg， Tdtg = 8 × TDTS； DTG[7:5]=111 => DT=(32+DTG[4:0])× Tdtg， Tdtg = 16 × TDTS； 例：若 TDTS = 125ns(8MHZ)，可能的死区时间为： 0 到 15875ns，若步长时间为 125ns； 16us 到 31750ns，若步长时间为 250ns； 32us 到 63us，若步长时间为 1us； 64us 到 126us，若步长时间为 2us； 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 1、 2 或 3，则不能修改这些位。

20.5.24 TIMx 死区时间寄存器 2（TIMx_DTR2）（x=2/3）

Address offset:0x54

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DTPE	DTAE
														RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	DTGF[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:18	Reserved	-	-	保留
17	DTPE	RW	0	死区时间预加载使能（Deadtime preload enable） 0：死区时间的值不可以预加载 1：死区时间的值可以预加载 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，2，3，则该位不能被修改。
16	DTAE	RW	0	死区时间不对称使能（Deadtime asymmetric enable） 0：上升沿和下降沿的死区时间相同，都由 DTG[7:0]配置 1：上升沿死区时间由 DTG[7:0]配置，下降沿死区时间由 DTGF[7:0]配置 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，2，3，则该位不能被修改。
15:8	Rserved	-	-	保留
7:0	DTGF [7:0]	RW	8’h0	下降沿死区时间生成器（Dead-time falling edge generator setup） 该位段定义了插入互补输出时下降沿的死区时间，假设 DT 表示其持续时间： DTGF[7:5]=0xx => DTF=DTGF[7:0] × Tdtg， Tdtg = TDTS； DTGF[7:5]=10x => DTF=(64+DTGF[5:0]) × Tdtg， Tdtg = 2 × TDTS； DTGF[7:5]=110 => DTF=(32+DTGF[4:0]) × Tdtg， Tdtg = 8 × TDTS； DTGF[7:5]=111 => DTF=(32+DTGF[4:0])× Tdtg， Tdtg = 16 × TDTS； 例：若 T_{DTS} = 125ns(8 MHz)，可能的死区时间为： 0 到 15875 ns，若步长时间为 125 ns； 16 µs 到 31750 ns，若步长时间为 250 ns；

Puya Confidential

				32 μs 到 63 μs，若步长时间为 1 μs； 64 μs 到 126 μs，若步长时间为 2 μs； 注：一旦 LOCK 级别（TIMx_BDTR 寄存器中的 LOCK 位）设为 1、2 或 3，则不能修改这些位

20.5.25 TIM_TYP1 编码器控制寄存器（TIMx_ECR）（x=2/3）

Address offset:0x58

Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	PWPRSC[2:0]			PW[7:0]							
					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	IPOS		FIDX	Res	Res	IDIR		IE
								RW	RW	RW			RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:27	Reserved	-	-	保留
26:24	PWPRSC	RW	3'h0	脉冲宽度预分频（Pulse width prescaler） 该位设置脉冲生成器的时钟预分频，公式如下： t_{PWG}= $2^{(PWPRSC[2:0])}$ x tCK_INT
23:16	PW	RW	8'h0	脉冲宽度（Pulse width） 该位定义了脉冲持续时间，公式如下： t_{PW}=PW[7:0] x t_{PWG}
15:8	Reserved	-	-	保留
7:6	IPOS	RW	2'h0	索引定位（Index positioning） 在正交编码器模式中（SMS[3:0] = 0001, 0010, 0011, 1110, 1111），该位用于指示 AB 输入在哪种配置下索引事件会复位计数器 00：AB=00 时，索引会复位计数器 01：AB=01 时，索引会复位计数器 10：AB=10 时，索引会复位计数器 11：AB=11 时，索引会复位计数器 在定向时钟模式或时钟+方向模式中(SMS[3:0] = 1010, 1011, 1100, 1101)，该位段用于指示哪种电平索引会复位计数器。在定向时钟模式中，应用于两路的时钟输入。 x0：当时钟为 0 时，索引复位计数器 x1：当时钟为 1 时，索引复位计数器 注：IPOS[1]位无效
5	FIDX	RW	0	第一次索引（First index） 该位指示是否只考虑第一次索引 0：索引一直有效 1：仅第一次索引可以复位计数器
4:3	Reserved	-	-	保留
1	IDIR	RW	0	索引方向（Index direction） 该位指示索引事件在哪个计数方向时复位计数器 00：无论哪个计数方向，索引均可以复位计数器 01：仅在向上计数时，索引可以复位计数器 10：仅在向下计数时，索引可以复位计数器 11：保留

Puya Confidential

0	IE	RW	0	索引使能（Index enable） 该位指示索引是否复位计数器 0：索引关闭 1：索引使能

20.5.26 TIMx 输入选择寄存器（TIMx_TISEL）（x=2/3）

偏移地址：0x5C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				RW	RW	RW	RW					RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				RW	RW	RW	RW					RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:28	Reserved	-	-	保留
27:24	TI4SEL	RW	4’h0	TI4 输入选择. 0000：TIMx_CH4 0001：COMP1_OUT 0010：COMP2_OUT 0011：COMP3_OUT 0100：COMP4_OUT 其他：保留
23:20	Reserved	-	-	保留
19:16	TI3SEL	RW	4’h0	TI3 输入选择. 0000：TIMx_CH3 0001：COMP1_OUT 0010：COMP2_OUT 0011：COMP3_OUT 0100：COMP4_OUT 其他：保留
15:12	Reserved	-	-	保留
11:8	TI2SEL	RW	4’h0	TI2 输入选择. 0000：TIMx_CH2 0001：COMP1_OUT 0010：COMP2_OUT 0011：COMP3_OUT 0100：COMP4_OUT 其他：保留
7:4	Reserved	-	-	保留
3:0	TI1SEL	RW	4’h0	TI1 输入选择. 0000：TIMx_CH1 0001：COMP1_OUT 0010：COMP2_OUT 0011：COMP3_OUT 0100：COMP4_OUT 0101：MCO1_CLK(TIM2)/ MCO2_CLK(TIM3)

Puya Confidential

				其他：保留

20.5.27 TIMx 备用功能选项寄存器 1（TIMx_AF1）（x=2/3）

偏移地址：0x60

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWM_PHS_EN	INTR_SEL		ETRSEL[3:2]	
											RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		BKCMP4P	BKCMP3P	BKCMP2P	BKCMP1P	BKINP	Res.	Res.	Res.	Res.	BKCMP4E	BKCMP3E	BKCMP2E	BKCMP1E	BKINE
RW	RW	RW	RW	RW	RW					RW	RW	RW	RW	RW	

Bit	Name	RW	Reset Value	Function
31:21	Reserved	-	-	保留
20	PWM_PHS_EN	RW	0	PWM 移相使能 0：关 1：开
19:18	INTR_SEL	RW	0	00：上溢或下溢中断 01：上溢中断 10：下溢中断 11：保留 注：不使用移相功能时，此位配置为 0
17:14	ETRSEL	RW	4’h0	外部触发源选择（etr_in source selection） 该位段用于选择 ETR 输入源 0000：TIMx_ETR 0001：TIM_ETR1 0010：TIM_ ETR2 0011：TIM_ ETR 3 0100：TIM_ ETR 4 0101：TIM_ ETR 5 0110：TIM_ ETR 6 0111：TIM_ ETR 7 1000：TIM_ ETR 8 1001：TIM_ ETR 9 1010：TIM_ ETR 10 1011：TIM_ ETR 11 1100：TIM_ ETR 12 1101：TIM_ ETR 13 1110：TIM_ ETR 14 1111：TIM_ ETR 15 注：一旦 LOCK 级别（TIMx_BDTR 寄存器中的 LOCK 位）被设置为 1，则该位不能被修改
13	BKCMP4P	RW	0	tim_brk_cmp4 输入极性 该位选择 tim_brk_cmp4 输入极性，必须与 BKP 极性位同时配置 0：tim_brk_cmp4 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效）

Puya Confidential

				1：tim_brk_cmp4 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
12	BKCMP3P	RW	0	tim_brk_cmp3 输入极性 该位选择 tim_brk_cmp3 输入极性，必须与 BKP 极性位同时配置 0：tim_brk_cmp3 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：tim_brk_cmp3 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
11	BKCMP2P	RW	0	tim_brk_cmp2 输入极性 该位选择 brk_cmp2 输入极性，必须与 BKP 极性位同时配置 0：tim_brk_cmp2 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：tim_brk_cmp2 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
10	BKCMP1P	RW	0	tim_brk_cmp1 输入极性 该位选择 brk_cmp1 输入极性，必须与 BKP 极性位同时配置 0：tim_brk_cmp1 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：tim_brk_cmp1 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
9	BKINP	RW	0	BKIN 输入极性（TIMx_BKIN input polarity） 该位选择 BKIN 输入极性的备用功能，必须与 BKP 极性位同时配置 0：BKIN 输入极性不翻转（BKP=0 时低有效，BKP=1 时高有效） 1：BKIN 输入极性翻转（BKP=0 时高有效，BKP=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
8:5	Reserved	-	-	保留
4	BKCMP4E	RW	0	tim_brk_cmp4 输入使能（tim_brk_cmp4 enable） 该位用于刹车输入时使能 tim_brk_cmp4。tim_brk_cmp4 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk_cmp4 输入关闭 1：tim_brk_cmp4 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
3	BKCMP3E	RW	0	tim_brk_cmp3 输入使能（tim_brk_cmp3 enable） 该位用于刹车输入时使能 tim_brk_cmp3。tim_brk_cmp3 输入与其它刹车源进行或逻辑作为刹车输入。

Puya Confidential

				0：tim_brk_cmp3 输入关闭 1：tim_brk_cmp3 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
2	BKCMP2E	RW	0	tim_brk_cmp2 输入使能（tim_brk_cmp2 enable） 该位用于刹车输入时使能 tim_brk_cmp2。tim_brk_cmp2 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk_cmp2 输入关闭 1：tim_brk_cmp2 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
1	BKCMP1E	RW	0	tim_brk_cmp1 输入使能（tim_brk_cmp1 enable） 该位用于刹车输入时使能 tim_brk_cmp1。tim_brk_cmp1 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk_cmp1 输入关闭 1：tim_brk_cmp1 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
0	BKINE	RW	1	BKIN 输入使能（TIMx_BKIN input enable） 该位用于刹车输入时使能 BKIN 的备用功能。BKIN 输入与其它刹车源进行或逻辑作为刹车输入。 0：BKIN 输入关闭 1：BKIN 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改。

20.5.28 TIMx 备用功能选项寄存器 2（TIMx_AF2）（x=2/3）

偏移地址：0x64

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCRSEL[2:0]		
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	BK2 CMP4 P	BK2 CMP3 P	BK2 CMP2 P	BK2 CMP1 P	BK2INP	Res.	Res.	Res.	Res.	BK2 CMP4 E	BK2 CMP3 E	BK2 CMP2 E	BK CMP1 E	BK2INE
		RW	RW	RW	RW	RW					RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:19	Reserved	-	-	Reserved
18:16	OCRSEL	RW	3'h0	OCREF 复位源选择（ocref_clr source selection） 该位段用于选择 ocref_clr 输入源 0000：ocref_clr0 0001：ocref_clr1 0010：ocref_clr2 0011：ocref_clr3 其它：保留

Puya Confidential

				注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
15:14	Reserved	-	-	保留
13	BK2CMP4P	RW	0	tim_brk2_cmp4 输入极性 该位选择 tim_brk2_cmp4 输入极性，必须与 BK2P 极性位同时配置 0：tim_brk2_cmp4 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：tim_brk2_cmp4 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
12	BK2CMP3P	RW	0	tim_brk2_cmp3 输入极性 该位选择 tim_brk2_cmp3 输入极性，必须与 BK2P 极性位同时配置 0：tim_brk2_cmp3 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：tim_brk2_cmp3 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
11	BK2CMP2P	RW	0	tim_brk2_cmp2 输入极性 该位选择 tim_brk2_cmp2 输入极性，必须与 BK2P 极性位同时配置 0：tim_brk2_cmp2 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：tim_brk2_cmp2 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
10	BK2CMP1P	RW	0	tim_brk2_cmp1 输入极性 该位选择 tim_brk2_cmp1 输入极性，必须与 BK2P 极性位同时配置 0：tim_brk2_cmp1 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：tim_brk2_cmp1 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
9	BK2INP	RW	0	BKIN2 输入极性（TIMx_BKIN2 input polarity） 该位选择 BKIN 输入极性的备用功能，必须与 BK2P 极性位同时配置 0：BKIN2 输入极性不翻转（BK2P=0 时低有效，BK2P=1 时高有效） 1：BKIN2 输入极性翻转（BK2P=0 时高有效，BK2P=1 时低有效） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
8:5	Reserved	-	-	Reserved
4	BK2CMP4E	RW	0	tim_brk2_cmp4 输入使能（tim_brk2_cmp4 enable） 该位用于刹车输入时使能 tim_brk2_cmp4。tim_brk2_cmp4 输入与其它刹车源进行或逻辑作为刹车输入。

Puya Confidential

				0：brk2_cmp4 输入关闭 1：brk2_cmp4 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
3	BK2CMP3E	RW	0	tim_brk2_cmp3 输入使能（tim_brk2_cmp3 enable） 该位用于刹车输入时使能 tim_brk2_cmp3。tim_brk2_cmp3 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk2_cmp3 输入关闭 1：tim_brk2_cmp3 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
2	BK2CMP2E	RW	0	tim_brk2_cmp2 输入使能（tim_brk2_cmp2 enable） 该位用于刹车输入时使能 tim_brk2_cmp2。tim_brk2_cmp2 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk2_cmp2 输入关闭 1：tim_brk2_cmp2 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
1	BK2CMP1E	RW	0	tim_brk2_cmp1 输入使能（tim_brk2_cmp1 enable） 该位用于刹车输入时使能 tim_brk2_cmp1。tim_brk2_cmp1 输入与其它刹车源进行或逻辑作为刹车输入。 0：tim_brk2_cmp1 输入关闭 1：tim_brk2_cmp1 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改
0	BK2INE	RW	1	BKIN2 输入使能（TIMx_BKIN input enable） 该位用于刹车输入时使能 BKIN2 的备用功能。BKIN2 输入与其它刹车源进行或逻辑作为刹车输入。 0：BKIN2 输入关闭 1：BKIN2 输入开启 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)被设置为 1，则该位不能被修改。

20.5.29 TIMx 多点触发 ADC 控制寄存器（TIMx_MTADCCR）（x=2/3）

偏移地址：0x6C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MTADCE	DMTCCR4HE	DMTCCR4E
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMTCCR3HE	DMTCCR3H	DMTCCR2HE	DMTCCR2H	DMTCCR1HE	DMTCCR1H	UMTCCR4HE	UMTCCR4E	UMTCCR3HE	UMTCCR3H	UMTCCR2HE	UMTCCR2H	UMTCCR1HE	UMTCCR1H	MTARRE	MTZEROE
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:19	Reserved	-	-	保留

Puya Confidential

18	MTADCE	RW	1'b0	组合触发 ADC 总使能（multi triger adc en） 1：使能组合触发 adc 功能 0：禁用组合触发 adc 功能
17	DMTCCR4HE	RW	1'b0	向下计数，CNT 为 CCR4H 时触发 ADC 使能（count down multi triger CCR4H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
16	DMTCCR4E	RW	1'b0	向下计数，CNT 为 CCR4 时触发 ADC 使能（count down multi triger CCR4 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
15	DMTCCR3HE	RW	1'b0	向下计数，CNT 为 CCR3H 时触发 ADC 使能（count down multi triger CCR3H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
14	DMTCCR3E	RW	1'b0	向下计数，CNT 为 CCR3 时触发 ADC 使能（count down multi triger CCR3 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
13	DMTCCR2HE	RW	1'b0	向下计数，CNT 为 CCR2H 时触发 ADC 使能（count down multi triger CCR1H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
12	DMTCCR2E	RW	1'b0	向下计数，CNT 为 CCR2 时触发 ADC 使能（count down multi triger CCR1 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
11	DMTCCR1HE	RW	1'b0	向下计数，CNT 为 CCR1H 时触发 ADC 使能（count down multi triger CCR1H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
10	DMTCCR1E	RW	1'b0	向下计数，CNT 为 CCR1 时触发 ADC 使能（count down multi triger CCR1 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
9	UMTCCR4HE	RW	1'b0	向上计数，CNT 为 CCR4H 时触发 ADC 使能（count up multi triger CCR4H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
8	UMTCCR4E	RW	1'b0	向上计数，CNT 为 CCR4 时触发 ADC 使能（count up multi triger CCR4 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
7	UMTCCR3HE	RW	1'b0	向上计数，CNT 为 CCR3H 时触发 ADC 使能（count up multi triger CCR3H enable），CEN 和 MTADCE 同时为 1 有效。

Puya Confidential

				1：使能该触发点。 0：禁用该出发点。
6	UMTCCR3E	RW	1'b0	向上计数，CNT 为 CCR3 时触发 ADC 使能（count up multi triger CCR3 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
5	UMTCCR2HE	RW	1'b0	向上计数，CNT 为 CCR2H 时触发 ADC 使能（count up multi triger CCR2H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
4	UMTCCR2E	RW	1'b0	向上计数，CNT 为 CCR2 时触发 ADC 使能（count up multi triger CCR2 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
3	UMTCCR1HE	RW	1'b0	向上计数，CNT 为 CCR1H 时触发 ADC 使能（count up multi triger CCR1H enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
2	UMTCCR1E	RW	1'b0	向上计数，CNT 为 CCR1 时触发 ADC 使能（count up multi triger CCR1 enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
1	MTARRE	RW	1'b0	CNT 为 ARR 时触发 ADC 使能（multi triger ARR enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。
0	MTZEROE	RW	1'b0	CNT 为 0 时触发 ADC 使能（multi triger zero enable），CEN 和 MTADCE 同时为 1 有效。 1：使能该触发点。 0：禁用该出发点。

20.5.30 TIMx DMA 控制寄存器（TIMx_DCR）（x=2/3）

偏移地址：0x3DC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			RW	RW	RW	RW	RW				RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:13	Reserved	-	-	保留
12:8	DBL	RW	5'h0	DMA 连续传送长度（DMA burst length) 这些位定义了 DMA 在连续模式下的传送长度(当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送)，即：定义传输的次数，传输可以是半字(双字节)或字节： 00000：1 次传输

Puya Confidential

				00001：2 次传输 00010：3 次传输 .... 例：我们考虑这样的传输： DBL=7 字节， DBA=TIMx_CR1 - 如果 DBL=7 字节， DBA=TIMx_CR1 表示待传输数据的地址，那么传输的地址由下式给出： (TIMx_CR1 的地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL 其中(TIMx_CR1 的地址) + DBA 再加上 7，给出了将要写入或者读出数据的地址，这样数据的传输将发生在从地址(TIMx_CR1 的地址) + DBA 开始的 7 个寄存器。 根据 DMA 数据长度的设置，可能发生以下情况： - 如果设置数据为半字(16 位)，那么数据就会传输给全部 7 个寄存器。 - 如果设置数据为字节，数据仍然会传输给全部 7 个寄存器：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，以此类推。因此对于定时器，用户必须指定由 DMA 传输的数据宽度。
7:5	Reserved	-	-	保留
4:0	DBA	RW	5'h0	DMA 基地址 (DMA base address) 这些位定义了 DMA 在连续模式下的基地址(当对 TIMx_DMAR 寄存器进行读或写时)， DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量： 00000：TIMx_CR1， 00001：TIMx_CR2， 00010：TIMx_SMCR， …….

20.5.31 TIMx 连续模式的 DMA 地址（TIMx_DMAR）（x=2/3）

偏移地址：0x3E0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAB[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	DMAB	RW	0	DMA 连续传送寄存器（DMA register for burst accesses) 对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作： TIMx_CR1 地址 + (DBA + DMA 索引)x4 其中： "TIMx_CR1 地址"是控制寄存器 1(TIMx_CR1)所在的地址； "DBA"是 TIMx_DCR 寄存器中定义的基地址； "DMA 索引"是由 DMA 自动控制的偏移量，它取决于 TIMx_DCR 寄存器中定义的 DBL。

Puya Confidential

21. 基本定时器（TIM6/TIM7）

21.1 TIM6/TIM7 简介

基本定时器 TIM6 和 TIM7 各包含一个 16 位自动装载计数器，由各自的可编程预分频器驱动。

它们可以作为通用定时器提供时间基准，特别地可以为数模转换器（DAC）提供时钟。实际上，它们在芯片内部直接连接到 DAC 并通过触发输出直接驱动 DAC。

TIM6 和 TIM7 是完全独立的，它们不共享任何资源。

21.2 TIM6/TIM7 主要特征

TIM6 和 TIM7 定时器的主要功能包括：

- 16 位自动装载计数器
- 16 位可编程(可以实时修改)的预分频器，计数器时钟频率的分频系数为 1～65535 之间的任意数值
- 触发 DAC 的同步电路
- 在更新事件（计数器溢出）发生时产生中断/DMA

21.3 TIM6/TIM7 功能描述

21.3.1 TIM6/TIM7 模块框图

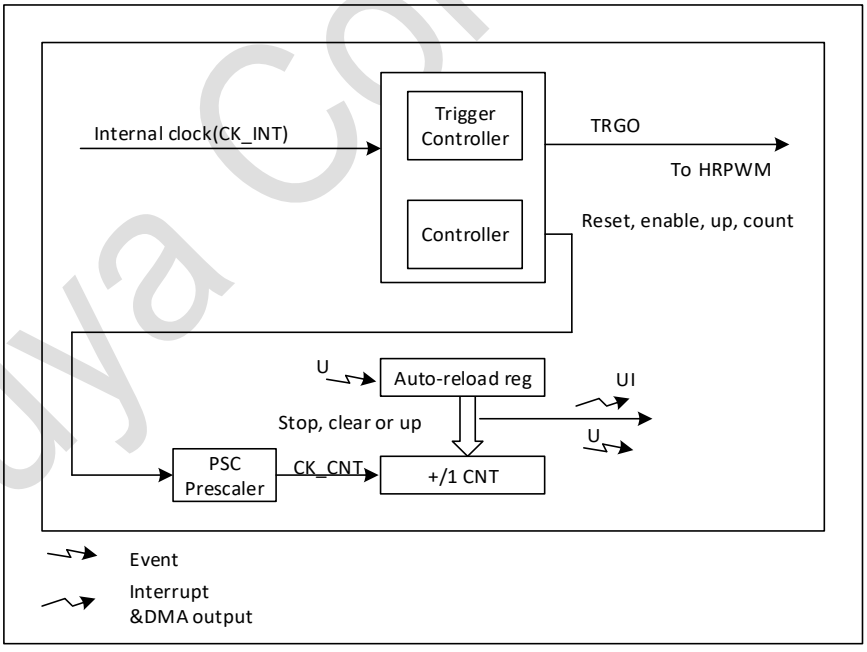


图 21-1 TIM6/TIM7 通用框图

Puya Confidential

21.3.2 时基单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数。计数器的时钟可以被预分频器分频。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

时基单元包含：

- 计数器寄存器（TIMx_CNT）
- 预分频器寄存器（TIMx_PSC）
- 自动装载寄存器（TIMx_ARR）

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问它的预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位(ARPE)的设置，预装载寄存器的内容被立即或在每次的更新事件（UEV) 时传送到影子寄存器。当计数器达到溢出条件并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，会产生更新事件。更新事件也可以由软件及其他条件产生。后续会详细描述每一种配置下更新事件的产生。

计数器由预分频器分频后的时钟输出 CK_CNT 驱动，仅当设置了 TIMx_CR1 寄存器中的计数器使能位(CEN)，CK_CNT 才对计数器有效。(更多有关使能计数器的细节，请参见从模式控制器的描述)。

注意，在设置了 TIMx_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个(在 TIMx_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频参数将在下一次更新事件到来时被采用。

下面几张图给出了在预分频器运行时，更改计数器参数的例子。

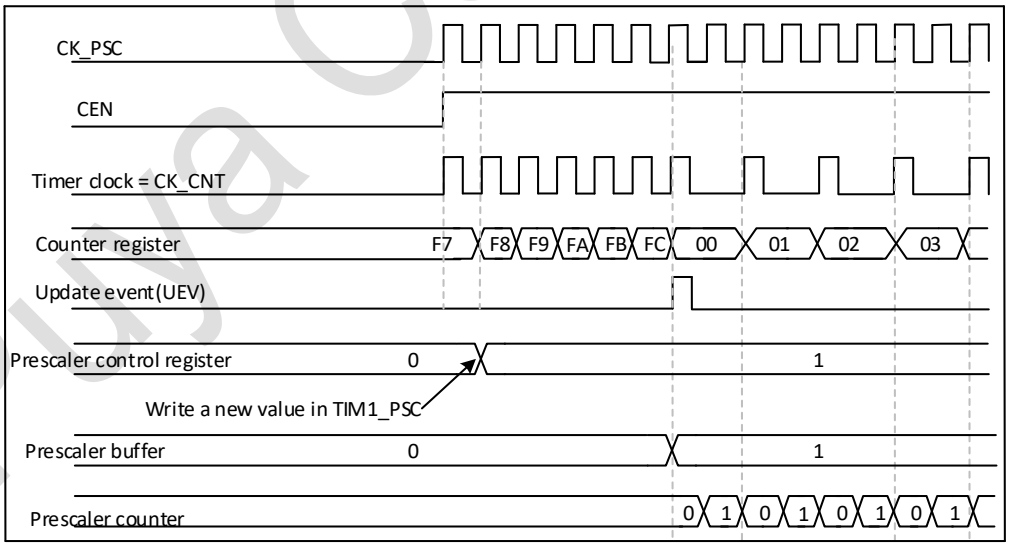


图 21-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

Puya Confidential

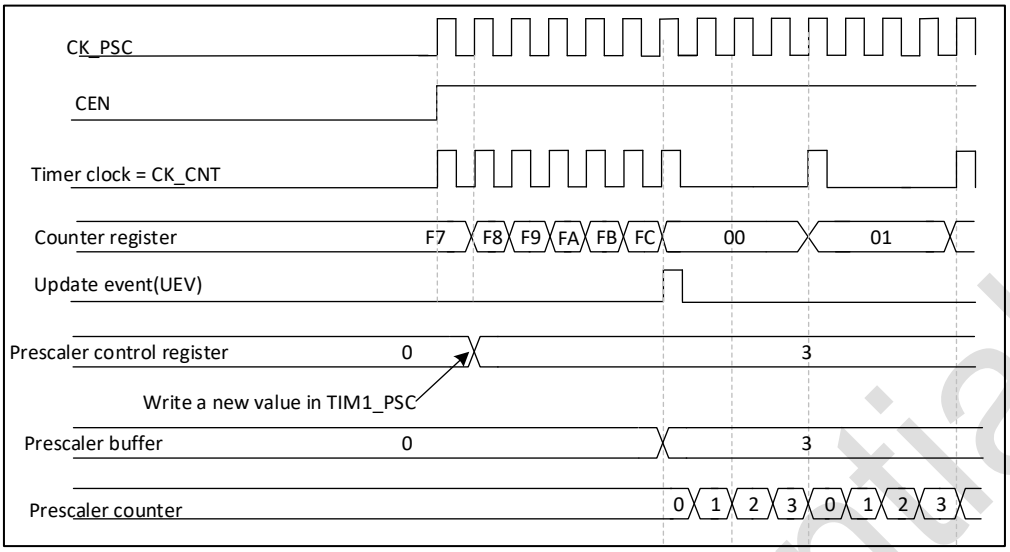


图 21-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

21.3.3 计数器模式

向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值（TIMx_ARR 的内容），然后重新从 0 开始计数并且产生一个计数上溢事件。

而在 TIMx_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。

通过设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清‘0’之前，将不会产生更新事件。即使这样，在应该产生更新事件时，计数器仍会被清‘0’，同时预分频器内部的计数器也被清‘0’（但预分频器的数值不变）。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），通过设置 UG 位可以产生一个更新事件 UEV，但不会置起 UIF 标志位（即不会产生中断请求）。这是为了避免在捕获事件时清除计数器，同时产生更新和捕获中断。

当发生一个更新事件时，所有以下的寄存器都被更新，硬件同时（依据 URS 位）设置更新标志位（TIMx_SR 寄存器中的 UIF 位）：

- 自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。
- 预分频器的缓冲区被置入预装载寄存器的值（TIMx_PSC 寄存器的内容）。

下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下的动作。

Puya Confidential

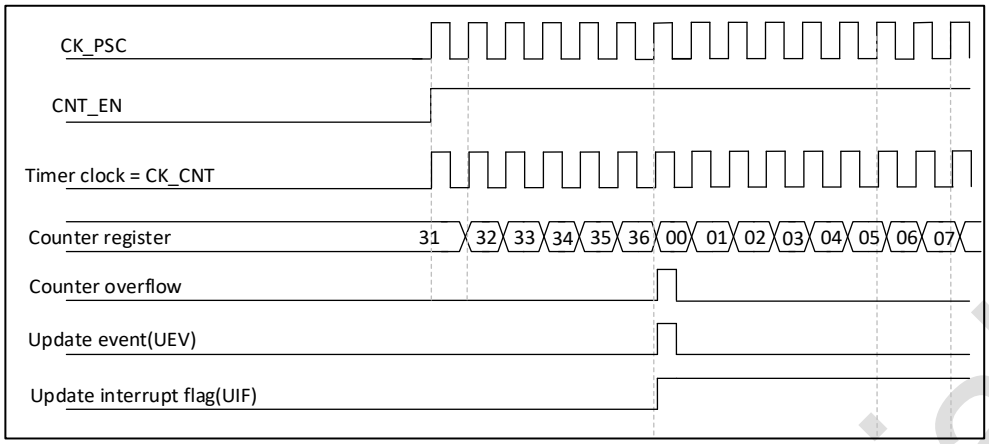


图 21-4 计数器时序图，内部时钟分频因子为 1

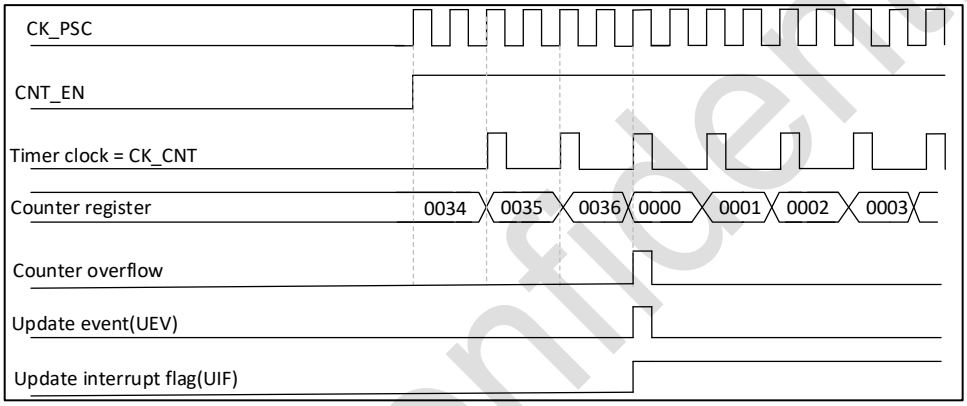


图 21-5 计数器时序图，内部时钟分频因子为 2

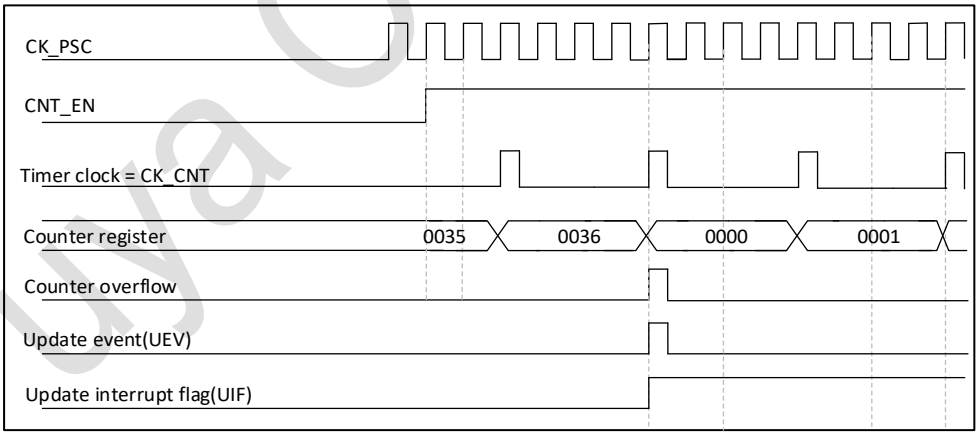


图 21-6 计数器时序图，内部时钟分频因子为 4

Puya Confidential

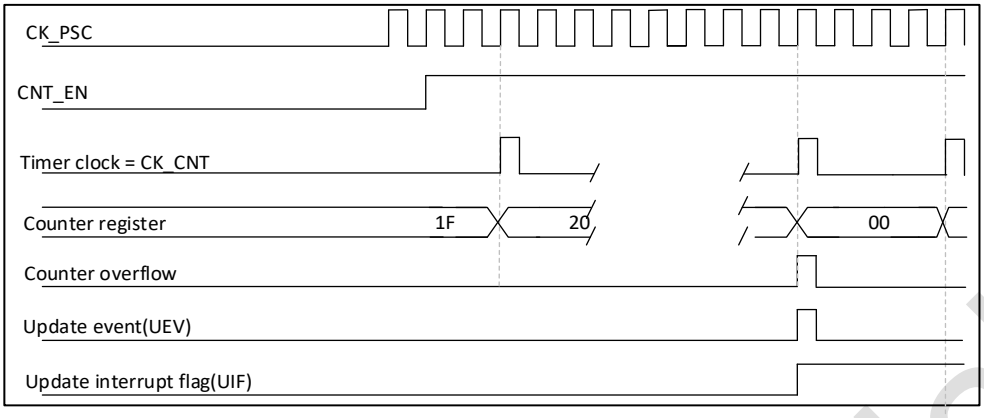


图 21-7 计数器时序图，内部时钟分频因子为 N

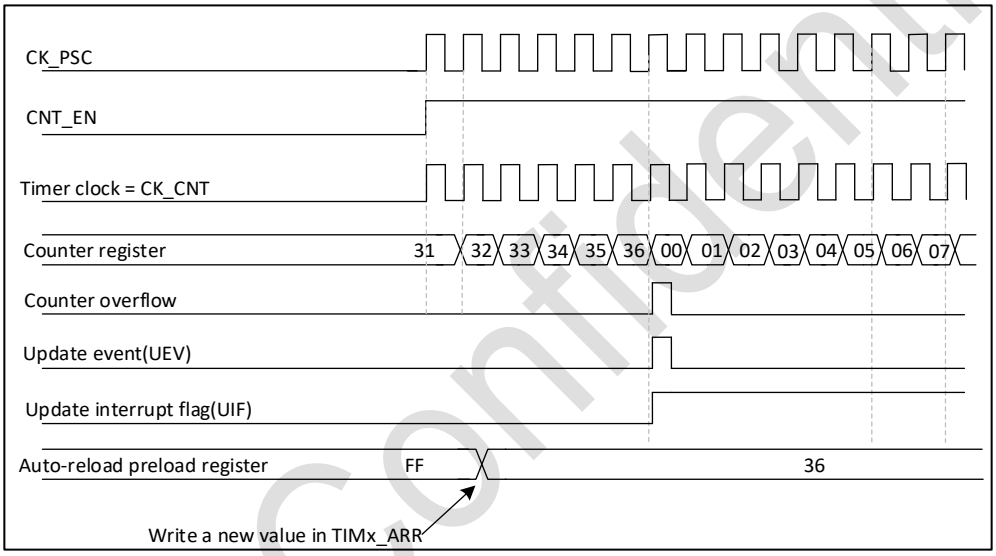


图 21-8 计数器时序图，当 ARPE=0 时的更新事件（没有预装 TIMx_ARR）

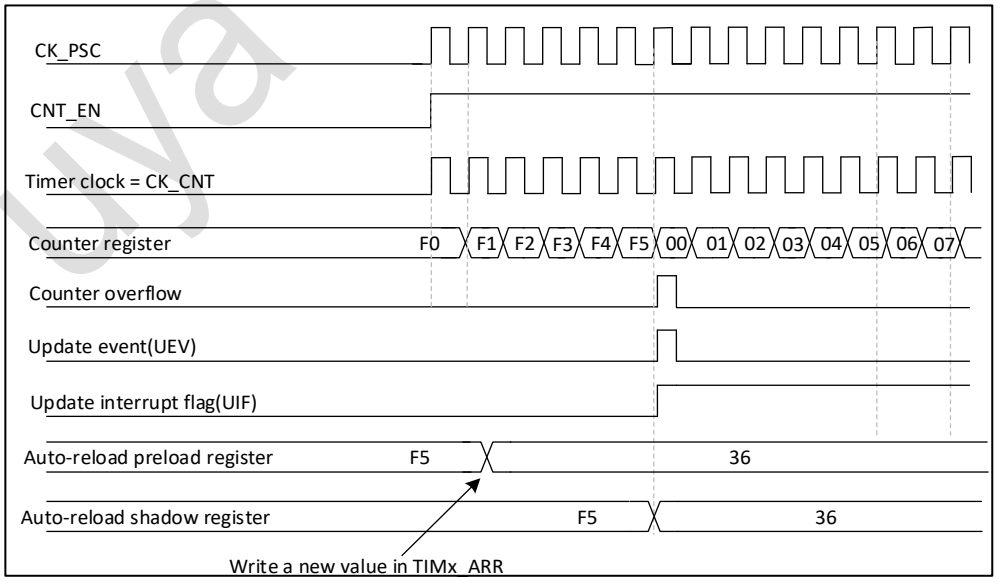


图 21-9 计数器时序图，当 ARPE=1 时的更新事件（预装了 TIMx_ARR）

Puya Confidential

21.4 TIM6/TIM7 调试模式

当微控制器进入调试模式时（Cortex-M4 核心停止），根据 DBG 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。详见随后的 DBG 节。

21.5 TIM6/TIM7 寄存器

21.5.1 TIMx 控制寄存器 1（TIMx_CR1）（x=6/7）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
								RW				RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7	ARPE	RW	0	自动重装载预装载允许位 0：TIMx_ARR 寄存器没有缓冲； 1：TIMx_ARR 寄存器被装入缓冲器。
6:4	Reserved	-	-	保留
3	OPM	RW	0	单脉冲模式 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件（清除 CEN 位）时，计数器停止。
2	URS	RW	0	更新请求源 软件通过该位选择 UEV 事件的源。 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器上溢/下溢 – 设置 UG 位 – 从模式控制器产生的更新 1：如果使能了更新中断或 DMA 请求，则只有计数器上溢/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0	更新禁止 软件通过该位允许/禁止 UEV 事件的产生。 0：允许 UEV。更新（UEV）事件由下述任一事件产生： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 影子寄存器装入预装载值。 1：禁止 UEV。不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则重新初始化计数器和预分频器。

Puya Confidential

0	CEN	RW	0	使能计数器 0：禁止计数器； 1：使能计数器。 注：在软件设置了 CEN 位后，门控模式才能工作。触发模式可以自动地通过硬件设置 CEN 位为 1。 在单脉冲模式下，当产生更新事件时 CEN 被自动清除。

21.5.2 TIMx 控制寄存器 2（TIMx_CR2）（x=6/7）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
									RW	RW	RW				

Bit	Name	RW	Reset Value	Function
31:7	Reserved	-	-	保留
6:4	MMS[2:0]	RW	3’b0	主模式选择 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 000：复位---TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。 001：使能---计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。该触发输出可用于在同一时间启动多个定时器或在一段时间内控制从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。 010：更新---更新事件被选为触发输出（TRGO）。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。
3:0	Reserved	-	-	保留

21.5.3 TIMx DMA/中断使能寄存器（TIMx_DIER）（x=6/7）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
							RW								RW

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8	UDE	RW	0	更新的 DMA 请求使能 0：禁止更新的 DMA 请求 1：允许更新的 DMA 请求
7:1	Reserved	-	-	保留

Puya Confidential

0	UIE	RW	0	更新中断使能 0：禁止更新中断 1：允许更新中断

21.5.4 TIMx 状态寄存器（TIMx_SR）（x=6/7）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
															RC_W0

Bit	Name	RW	Reset Value	Function
31:1	Reserved	-	-	保留
0	UIF	RC_W0	0	当产生更新事件时该位由硬件置 1。它由软件清 0。 0：无更新事件产生； 1：更新事件发生。当寄存器被更新时该位由硬件置 1： – 若 TIMx_CR1 寄存器的 UDIS=0，当计数器上溢时产生更新事件； – 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当 TIMx_EGR 寄存器的 UG=1 时产生更新事件(软件对 CNT 重新初始化)；

21.5.5 TIMx 事件产生寄存器（TIMx_EGR）（x=6/7）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
															W

Bit	Name	RW	Reset Value	Function
31:1	Reserved	-	-	保留
0	UG	W	0	产生更新事件 该位由软件置‘1’，由硬件自动清‘0’。 0：无动作； 1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清‘0’(但是预分频系数不变)。

21.5.6 TIMx 计数寄存器（TIMx_CNT）（x=6/7）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CNT[15:0]	RW	16'b0	计数器值

21.5.7 TIMx 预分频寄存器（TIMx_PSC）（x=6/7）

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	PSC[15:0]	RW	16'b0	预分频器的值。 计数器的时钟频率（CK_CNT）等于 f_{CK_PSC}/(PSC[15:0]+1)。 PSC 包含每次发生更新事件时要装载到有效预分频器寄存器的值（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为"复位模式"时通过触发控制器清零时）。

21.5.8 TIMx 自动重载寄存器（TIMx_ARR）（x=6/7）

TIM6

偏移地址：0x2C

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	ARR[15:0]	RW	16'hFFFF	自动重装载的值 ARR 包含了将要装载入实际的自动重装载寄存器的值。 当自动重装载的值为空时，计数器不工作。

TIM7

偏移地址：0x00

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
								RW				RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7	ARPE	RW	0	自动重装载预装载允许位 0：TIMx_ARR 寄存器没有缓冲； 1：TIMx_ARR 寄存器被装入缓冲器。
6:4	Reserved	-	-	保留
3	OPM	RW	0	单脉冲模式 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件（清除 CEN 位）时，计数器停止。
2	URS	RW	0	更新请求源 软件通过该位选择 UEV 事件的源。 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器上溢/下溢 – 设置 UG 位 – 从模式控制器产生的更新 1：如果使能了更新中断或 DMA 请求，则只有计数器上溢/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0	更新禁止 软件通过该位允许/禁止 UEV 事件的产生。 0：允许 UEV。更新（UEV）事件由下述任一事件产生： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 影子寄存器装入预装载值。 1：禁止 UEV。不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则重新初始化计数器和预分频器。
0	CEN	RW	0	使能计数器 0：禁止计数器； 1：使能计数器。 注：在软件设置了 CEN 位后，门控模式才能工作。触发模式可以自动地通过硬件设置 CEN 位为 1。 在单脉冲模式下，当产生更新事件时 CEN 被自动清除。

Puya Confidential

22. 低功耗定时器（LPTIM）

22.1 LPTIM 简介

LPTIM 是一个 16 位定时器。由于其时钟源的多样性，LPTIM 能够在所有功耗模式下保持运行。考虑到 LPTIM 即使没有内部时钟源也能运行，可以用作“脉冲计数器”，这在某些应用中很有用。LPTIM 引入了一种灵活的时钟方案，可提供所需的功能和性能，同时将功耗降至最低。

22.2 LPTIM 主要特性

- 16 位向上计数器
- 3 位预分频器，具有 8 个可能的分频因子（1，2，4，8，16，32，64，128）
- 可选时钟
  - 内部时钟源：APB 时钟或其他任何振荡器（见 RCC 章节）
  - 通过 LPTIM 输入的外部时钟源（在没有运行嵌入式振荡器的情况下工作，由脉冲计数器应用程序使用）。
- 16 位 ARR 可重载寄存器
- 16 位比较寄存器
- 连续/单次模式
- 可选软件/硬件输入触发
- 可配置数字毛刺滤波器
- 可配置输出：脉冲、PWM
- 可配置 IO 极性
- 编码器模式

Puya Confidential

22.3 LPTIM 功能描述

22.3.1 模块框图

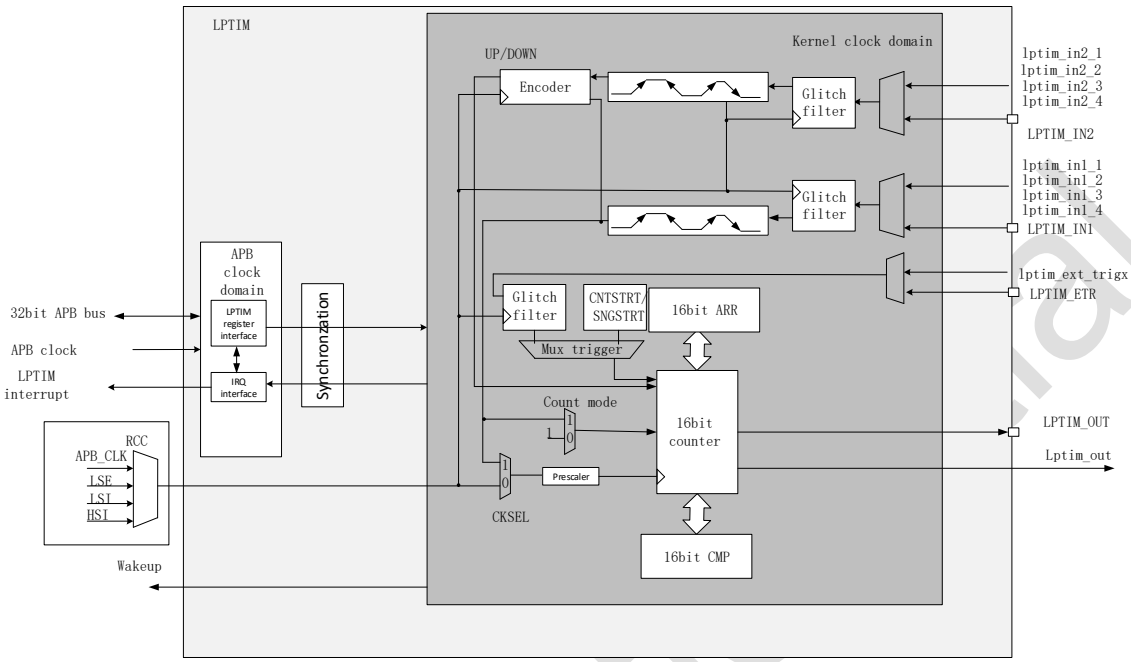


图 22-1 LPTIM 框图

22.3.2 复位和时钟

通过 RCC 模块，可以使用内部时钟信号对 LPTIM 进行时钟控制（该时钟信号可以在 APB、LSI、LSE 中进行选择）。

22.3.3 毛刺滤波器

LPTIM 输入，无论是外部 (映射到 GPIO) 还是内部 (在芯片级映射到其他嵌入式外围设备，如比较器) 触发，都受到数字滤波器的保护，可防止任何毛刺和噪声扰动在 LPTIM 内部传播。这是为了防止意外计数或触发。在激活数字滤波器之前，应首先向 LPTIM 提供一个内部时钟源。这是保证滤波器正常运行所必需的。LPTIM 包含如下数字滤波器：

- 数字滤波器保护 LPTIM 内部触发输入。数字滤波器灵敏度由 TRGFLT 位控制。

注意：数字滤波器灵敏度由组控制。不可能在同一组内单独配置每个数字滤波器灵敏度。

滤波器灵敏度作用于在 LPTIM 某一输入上检测到的连续相等样本的数量，以将信号电平变化视为有效转换。下图为在编程 2 个连续样本的情况下毛刺滤波器行为的示例。

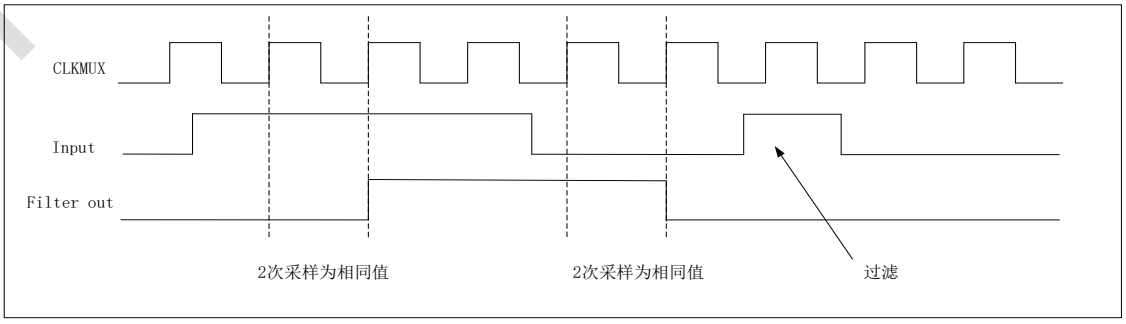


图 22-2 毛刺滤波器行为示例

Puya Confidential

注意：如果没有提供内部时钟信号，则必须通过将 TRGFLT 位设置为“0”来停用数字滤波器。在这种情况下，可以使用外部模拟滤波器（IO 模块实现）来保护 LPTIM 外部输入免受毛刺干扰。

22.3.4 预分频

LPTIM 16 位计数器，由一个可配置的 2 次方预分频器控制驱动。预分频器分频比由 LPTIM_CFGR.PRESC[2:0]控制。

下表列出了所有情况：

表 22-1 预分频器的分频比

Programming	Dividing factor
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

22.3.5 触发选择

LPTIM 计数器可以通过软件启动，也可以在检测到 8 个（可选最大输入个数）触发输入之一的激活边沿后启动。

TRIGEN[1:0]用于确定 LPTIM 触发源：

- 当 TRIGEN[1:0]等于“00”时，一旦 CNTSTRT 或 SNGSTRT 位之一由软件设置，LPTIM 计数器就会启动。TRIGEN[1:0]的其他值可用于配置触发输入使用的有效边沿，支持上升沿和下降沿。一旦检测到有效边沿，LPTIM 计数器就会启动。
- 当 TRIGEN[1:0]不是‘00’时，TRIGSEL[2:0]用于选择 8 个触发输入中的哪一个启动计数器。

外部触发器被视为 LPTIM 的异步信号。因此，在触发检测之后，由于同步，在定时器开始运行之前需要两个计数器时钟周期的延迟。

如果在定时器已经启动时发生新的触发事件，它将被忽略。

在设置 SNGSTRT/CNTSTRT 位之前必须使能定时器。当定时器被禁用时，对这些位的任何写入都将被硬件丢弃。

注意：当通过软件启动计数器时（TRIGEN[1:0] = 00），LPTIM_CR 寄存器更新（设置 SNGSTRT 或 CNTSTRT 位之一）和计数器有效启动之间有 3 个计数时钟周期的延迟。

22.3.6 操作模式

LPTIM 具有两种如下工作模式：

- 连续计数模式：计时器自由运行，从触发事件开始运行，直到计时器被禁止时才停止。
- 单次计数模式：定时器从一个触发事件开始，当达到 ARR 值时停止。

22.3.6.1 单次模式

要使能单次计数，LPTIM_CR.SNGSTRT 寄存器位必须置 1。

设置 SNGSTRT 将启动计数器进行单次计数。

Puya Confidential

一个新的触发事件将重新启动计时器。在计数器启动之后，到达 ARR 之前，任何触发事件都将被忽略。如果选择外部触发，则在设置 SNGSTRT 位后以及计数器寄存器停止（包含零值）后，再产生外部触发事件，LPTIM 将启动计数器进行新的单次计数周期，如下图所示。

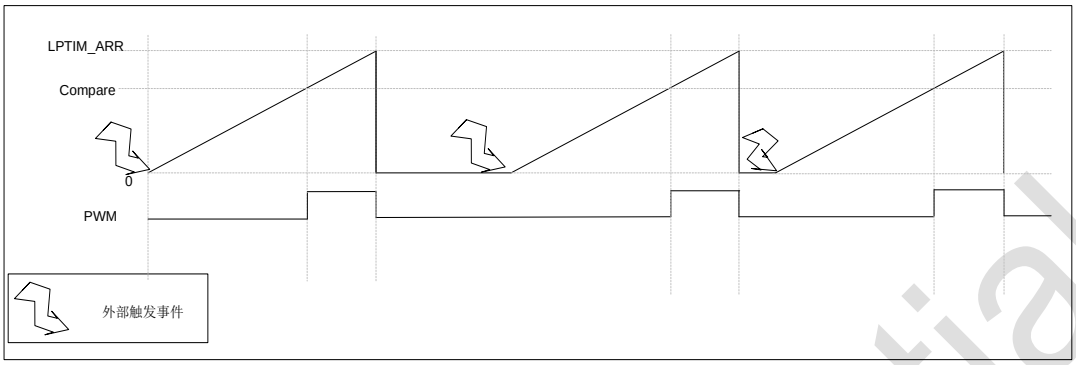


图 22-3 LPTIM 输出,单次计数模式

当 LPTIM_CFGR.WAVE 位置 1 时，一次设置模式被激活。这种情况下，计数器仅在第一次触发后启动一次，随后的任何触发事件均被忽略。如下图所示：

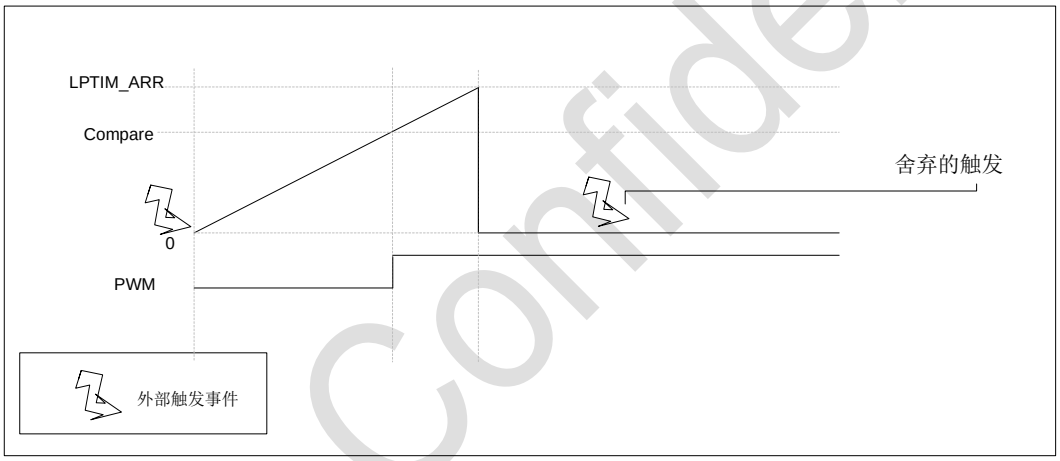


图 22-4 LPTIM 输出,单次计数模式（WAVE=1）

在软件启动（TRIGEN[1:0]=‘00’）的情况下，SNGSTRT 置位将启动计数器进行单次计数。

22.3.6.2 连续计数模式

要使能连续计数，LPTIM_CR.CNTSTRT 位必须置 1。

如果选择外部触发，则在 CNTSTRT 置位后到达的外部触发事件将启动计数器进行连续计数。任何后续的外部触发事件都将被丢弃，如下图所示。

在软件启动（TRIGEN[1:0]=‘00’）的情况下，设置 CNTSTRT 将启动计数器以进行连续计数。

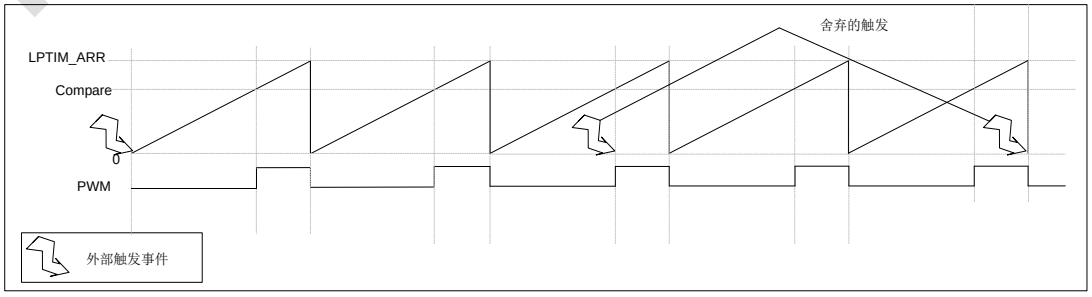


图 22-5 LPTIM 输出，连续计数模式

Puya Confidential

LPTIM_CR.SNGSTRT 和 LPTIM_CR.CNTSTRT 位只能在定时器使能时置位 (LPTIM_CR.ENABLE 为“1”)。可以快速的从单次模式转变为连续模式：

- 如果之前选择了连续模式，则置位 LPTIM_CR.SNGSTRT 会将 LPTIM 切换到单次模式。计数器（如果开始计数）一到达 ARR 就会停止。
- 如果先前选择了单次模式，则置位 LPTIM_CR.CNTSTRT 会将 LPTIM 切换到连续模式。计数器（如果开始计数）一到达 ARR 就会重新启动。

22.3.7 波形产生

两个 16 位寄存器，LPTIM_ARR（自动重载寄存器）和 LPTIM_CMP（比较寄存器），用于在 LPTIM 输出上生成几种不同的波形。

定时器可以生成以下波形：

- PWM 模式：LPTIM 输出置位条件为 LPTIM_CNT 中的计数器值超过了 LPTIM_CMP 中的比较值。一旦 LPTIM_ARR 和 LPTIM_CNT 寄存器之间发生匹配，LPTIM 输出就会复位。
- 单脉冲模式：输出波形类似于第一个脉冲的 PWM 模式，然后输出永久复位。
- 一次设置模式：输出波形类似于单脉冲模式，不同点在于输出会保持在最后一个信号电平（取决于输出配置的极性）。

上述模式要求 LPTIM_ARR 寄存器值严格大于 LPTIM_CMP 寄存器值。

LPTIM 输出波形可通过 WAVE 位进行如下配置：

- 将 WAVE 位重置为“0”会强制 LPTIM 生成 PWM 波形或单脉冲波形，具体取决于设置的位：CNTSTRT 或 SNGSTRT。
- 将 WAVE 位设置为“1”会强制 LPTIM 生成一次设置模式波形。

WAVPOL 位控制 LPTIM 输出极性。更改立即生效，因此，在重新配置极性后，甚至在启用定时器之前，输出默认值将立即改变。

可以产生频率高达 LPTIM 时钟频率除以 2 的信号。下图显示了可在 LPTIM 输出上生成的三种可能波形。此外，它还显示了使用 WAVPOL 位改变极性的效果。

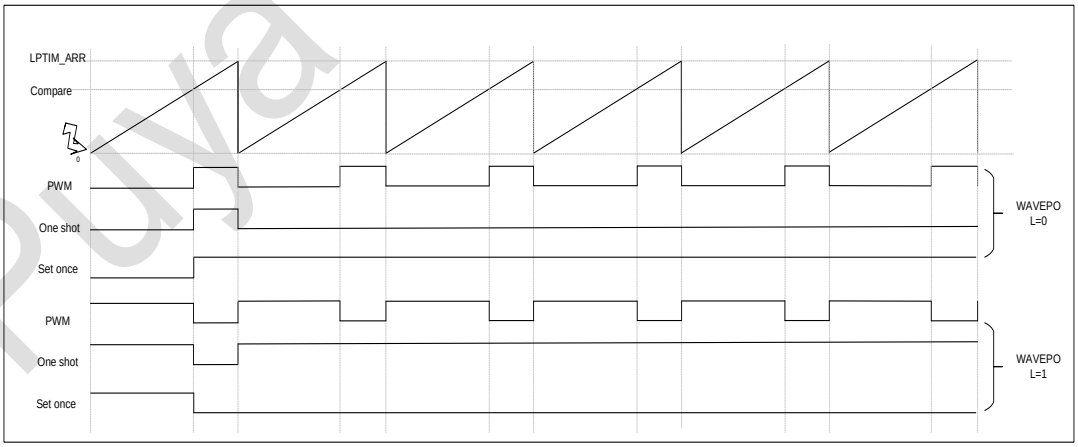


图 22-6 LPTIM 波形产生

22.3.8 寄存器更新

LPTIM_ARR 寄存器在 APB 总线写操作后立即更新，如果定时器已经启动，则与下一个 LPTIM 更新事件同步进行更新。

Puya Confidential

PRELOAD 位控制 LPTIM_ARR 寄存器的更新方式：

- 当 PRELOAD 位被复位为“0”：LPTIM_ARR 寄存器在任何写访问后立即更新。
- 当 PRELOAD 位被设为“1”时：如果定时器已经启动，则 LPTIM_ARR 寄存器将在当前周期结束时更新。

LPTIM APB 接口和 LPTIM 内核逻辑使用不同的时钟，因此在 APB 写入,和写入的值被应用到计数器比较器时，存在一定的延迟。在此延迟周期内，必须避免对这些寄存器进行任何额外的写操作。

LPTIM_ISR 寄存器中的 ARROK 标志和 CMPOK，指示对 LPTIM_ARR 和 LPTIM_CMP 寄存器的写操作是否完成。

对 LPTIM_ARR 和 LPTIM_CMP 寄存器进行写操作后，只有在前一次写操作完成后，才能对同一寄存器执行新的写操作。在 ARROK 标志位或者 CMPOK 标志位置位之前的任何连续写入都将导致不可预测的结果。

22.3.9 计数器模式

可选择以下计数模式：

- LPTIM 由内部时钟源计时

LPTIM 时钟由内部时钟源提供，LPTIM 计数器配置为在每个内部时钟脉冲后更新。

22.3.10 定时器使能

LPTIM_CR.ENABLE 位用于使能/禁用 LPTIM 内核逻辑。置位 ENABLE 位后，需要延迟两个计数器时钟才能使能 LPTIM。

仅当 LPTIM 禁用时，才能修改 LPTIM_CFGR 和 LPTIM_IER 寄存器。

22.3.11 定时器复位

为了将 LPTIM_CNT 寄存器的内容复位，提供如下所述两种复位机制。

注意：异步复位和同步复位应确保互斥使用。

22.3.11.1 异步复位

异步复位由 LPTIM_CR 寄存器的 RSTARE 位控制。当该位被置为 1 时，任何读 LPTIM_CNT 寄存器的访问都将其内容复位为零。异步复位应在未提供 LPTIM 内核时钟的时间范围内触发。

应注意，为了可靠地读取 LPTIM_CNT 寄存器，必须进行 2 次读访问并比较其结果，结果一致，被则认为是读出值是可靠的。

需要注意的是：

- 使能异步复位时，读取两次 LPTIM_CNT 寄存器的结果是不同的（第一次读会复位 LPTIM_CNT）。
- 在 LPTIM 计数时钟选择 PCLK 时，连续访问 2 次也不能保证读出值可靠。

22.3.11.2 同步复位

同步复位由 LPTIM_CR.COUNTRST 位控制。将 COUNTRST 位置 1 后，复位信号送给 LPTIM 的内核时钟域。所以需要注意在复位起作用前，LPTIM 内核时钟域的逻辑电路已过去几个时钟周期了。这将使从复位触发到复位生效，LPTIM 计数器多计了额外几个数。

Puya Confidential

由于 COUNTRST 是 APB 时钟域的，而 LPTIM 计数器位于 LPTIM 内核时钟域，所以当向 COUNTRST 位写入 1 时，需要内核时钟的 3 个时钟周期的延迟来同步来自 APB 时钟域的复位信号。

22.3.12 编码器模式

该模式允许处理来自用于检测旋转元件角位置的正交编码器的信号。编码器接口模式简单地充当具有方向选择的外部时钟。这意味着计数器只是在 0 和 ARR 值之间连续计数（0 到 ARR 或 ARR 到 0，具体取决于方向）。因此必须在启动前配置 LPTIM_ARR。从两个外部输入信号 Input1 和 Input2 生成一个时钟信号来为 LPTIM 计数器提供时钟。这两个信号之间的相位决定了计数方向。

编码器模式仅在 LPTIM 由内部时钟源计数时可用。Input1 和 Input2 输入上的信号频率不得超过 LPTIM 内部时钟的 1/4。这是强制性要求以保证 LPTIM 的正常运行。

方向更改由 LPTIM_ISR 寄存器中的两个向下和向上标志指示。此外，如果启用 DOWNIE/UPIE 位，则可以为两个方向更改事件生成中断。

要激活编码器模式，必须将 ENC 位设置为“1”。LPTIM 必须首先配置为连续模式。

当编码器模式处于活动状态时，LPTIM 计数器会根据增量编码器的速度和方向自动修改。因此，它的内容总是代表编码器的位置。由 UP 和 DOWN 标志指示的计数方向对应于编码器转子的旋转方向。

根据 CKPOL[1:0]位配置的边沿灵敏度，可能会出现不同的计数情况。下表总结了可能的组合，假设 Input1 和 Input2 不同时切换。

表 22-2 计数方式

有效沿	反向信号电平 (input1 for input2 input2 for input1)	input1 信号		input2 信号	
		上升沿	下降沿	上升沿	下降沿
上升沿	高	递减	不计数	递增	不计数
	低	递增	不计数	递减	不计数
下降沿	高	不计数	递增	不计数	递减
	低	不计数	递减	不计数	递增

下图显示了配置双沿灵敏度的编码器模式的计数序列。

在此模式下，预分频器分频比必须等于其复位值 1（PRESC[2:0]位必须为“000”）。

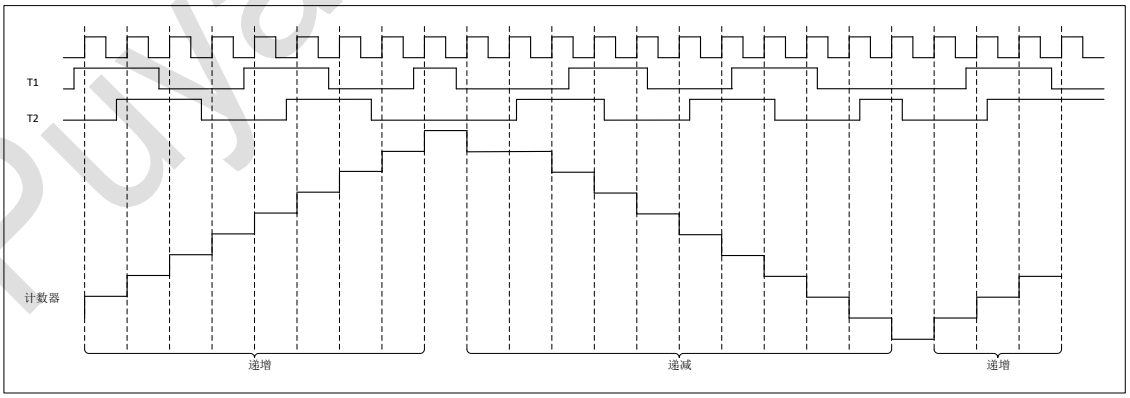


图 22-7 LPTIM 计数模式

22.3.13 调试模式

当芯片进入 debug 模式，取决于 DBG 模块的 DBG_LPTIM_STOP 位的设定，LPTIM 或者继续正常工作，或者停止工作。

Puya Confidential

22.4 LPTIM 低功耗模式

表 22-3 LPTIM 低功耗模式

Mode	Description
Sleep	无影响。 LPTIM 中断（使能后）会退出 Sleep 模式.
Low-power run	无影响。
Low-power sleep	无影响。LPTIM 中断导致设备退出低功耗睡眠模式。
Stop0/Stop1	当 LPTIM 由 LSE 或 LSI 计时时没有影响。LPTIM 中断导致设备退出 Stop 0 和 Stop 1。

注：在使用 LPTIM 唤醒时，需要确保两次唤醒的间隔时间大于 6 个 PCLK 周期，否则将导致不可预知的结果。

22.5 LPTIM 中断

如果下列事件在 LPTIM_IER 寄存器内使能，则这些事件将生成中断/唤醒事件：

- 比较匹配
- 自动重新加载匹配（如果是编码器模式，则无论方向如何）
- 外部触发事件
- 自动重载寄存器写入完成
- 比较寄存器写入完成
- 方向改变（编码器模式），可编程（上/下/两者）。

注意：如果在 LPTIM_ISR 寄存器（状态寄存器）中的相应标志置 1 后，LPTIM_IER 寄存器（中断使能寄存器）中的相应位才被置 1，则不会产生中断。

表 22-4 LPTIM 中断事件

中断事件	描述
比较匹配	当计数器寄存器（LPTIM_CNT）的内容与比较寄存器（LPTIM_CMP）的内容匹配时，会产生中断标志。
自动重载匹配	当计数器寄存器的内容（LPTIM_CNT）与自动重新加载寄存器的内容匹配（LPTIM_ARR），中断标志置位
外部触发事件	当检测到外部触发事件时，中断标志被置位
自动重载寄存器写入完成	当对 LPTIM_ARR 寄存器的写操作完成时，会产生中断标志。
比较寄存器写入完成	当对 LPTIM_CMP 寄存器的写操作完成时，会产生中断标志。
方向改变	在编码器模式下使用。嵌入了两个中断标志以指示方向改变： — UP 标志指示递增计数方向改变 — DOWN 标志表示减计数方向改变

Puya Confidential

22.6 LPTIM 寄存器

22.6.1 LPTIM 中断和状态寄存器（LPTIM_ISR）

地址偏移：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWM	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
									R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:7	Reserved	-	-	保留
6	DOWN	R	0	计数器由递增计数变为递减计数 在编码器模式下，DOWN 位由硬件设置用于通知应用程序计数器已从递增计数变为递减计数。可以通过向 LPTIM_ICR 寄存器中的 DOWNCF 位写入 1 来清除 DOWN 标志。 注意：如果 LPTIM 不支持编码器模式功能，则该位保留。
5	UP	R	0	计数器由递减计数变为递增计数 在编码器模式下，UP 位由硬件设置用于通知应用程序计数器已从递减计数变为递增计数。可以通过向 LPTIM_ICR 寄存器中的 UPCF 位写入 1 来清除 UP 标志。注意：如果 LPTIM 不支持编码器模式功能，则该位保留。
4	ARROK	R	0	自动重载寄存器更新 OK。 ARROK 由硬件设置，以通知应用程序 APB 总线对 LPTIM_ARR 的写操作已成功完成。向 LPTIM_ICR.ARROKCF 写入 1 可清除 ARROK 标志。
3	CMPOK	R	0	比较寄存器更新 OK CMPOK 由硬件置位，用于通知应用程序 APB 总线对 LPTIM_CMP 寄存器的写操作已成功完成。向 LPTIM_ICR.CMPOKCF 写入 1 可清除 ARROK 标志。
2	EXTTRIG	R	0	外部触发边沿事件 EXTTRIG 由硬件置位，用于通知应用程序所选外部触发输入上已出现有效边沿。如果因为计时器已经启动而忽略了触发器，则不会设置此标志。可以通过向 LPTIM_ICR 寄存器中的 EXTTRIGCF 位写入 1 来清除 EXTTRIG 标志。
1	ARRM	R	0	自动重载匹配 ARRM 由硬件设置，通知应用程序 LPTIM_CNT 寄存器值匹配 LPTIM_ARR 寄存器的值。向 LPTIM_ICR 寄存器的 ARRMCF 位写入 1 可清除 ARRM 标志。
0	CMPM	R	0	比较匹配

Puya Confidential

				CMPM 位由硬件置位，用于通知应用程序 LPTIM_CNT 寄存器值达到 LPTIM_CMP 寄存器的值。向 LPTIM_ICR 寄存器的 CMPMCF 位写入 1 可清除 CMPM 标志。

22.6.2 LPTIM 中断清零寄存器（LPTIM_ICR）

地址偏移：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWMCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
									W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:7	Reserved	-	-	保留
6	DOWNCF	W	0	计数方向改变为向下清除标志 向该位写入 1 会清除 LPTIM_ISR 寄存器中的 DOWN 标志。 注意：如果 LPTIM 不支持编码器模式功能，则该位保留。
5	UPCF	W	0	方向改变为向上清除标志 向该位写入 1 会清除 LPTIM_ISR 寄存器中的 UP 标志。 注意：如果 LPTIM 不支持编码器模式功能，则该位保留。
4	ARROKCF	W	0	自动重载寄存器更新 OK 清除标志。 向该位写入 1 可清除 LPTIM_ISR 寄存器中的 ARROK 标志。
3	CMPOKCF	W	0	比较寄存器更新 OK 清除标志 向该位写入 1 会清除 LPTIM_ISR 寄存器中的 CMPOK 标志。
2	EXTTRIGCF	W	0	外部触发有效沿清除标志 向该位写入 1 会清除 LPTIM_ISR 寄存器中的 EXTTRIG 标志
1	ARRMCF	W	0	自动重载匹配清除标志 向该位写入 1 可清除 LPTIM_ISR 寄存器中的 ARRM 标志。
0	CMPMCF	W	0	比较匹配清除标志 向该位写入 1 会清除 LPTIM_ISR 寄存器中的 CMPM 标志。

22.6.3 LPTIM 中断使能寄存器（LPTIM_IER）

地址偏移：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
									RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:7	Reserved	-	-	保留
6	DOWNIE	RW	0	计数器由递增计数变为递减计数中断使能

Puya Confidential

				0：禁用 DOWN 中断 1：使能 DOWN 中断 注意：如果 LPTIM 不支持编码器模式功能，则该位保留。
5	UPIE	RW	0	计数器由递减计数变为递增计数中断使能 0：禁用 UP 中断 1：使能 UP 中断 注意：如果 LPTIM 不支持编码器模式功能，则该位保留。
4	ARROKIE	RW	0	自动重载寄存器更新 OK 中断使能。 0：禁用 ARROK 中断 1： 使能 ARROK 中断
3	CMPOKIE	RW	0	比较寄存器更新 OK 中断使能 0：禁用 CMPOK 中断 1：使能 CMPOK 中断
2	EXTTRIGIE	RW	0	外部触发有效边沿中断使能 0：禁用 EXTTRIG 中断 1：使能 EXTTRIG 中断
1	ARRMIE	RW	0	自动重载匹配中断使能 0：禁用 ARRM 中断 1： 使能 ARRM 中断
0	CMPMIE	RW	0	比较匹配中断使能 0：禁用 CMPM 中断 1：使能 CMPM 中断

22.6.4 LPTIM 配置寄存器（LPTIM_CFGR）

地址偏移：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNT MODE	PRELOAD	WAVPOL	WAVE	Res.	TRIGEN[1:0]		Res.
							RW	RW	RW	RW	RW		RW	RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRIGSEL[2:0]			Res.	PRESC[2:0]			Res.	TRGFLT[1:0]		Res.	CKFLT[1:0]		CKPOL[1:0]		CKSEL
RW	RW	RW		RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:25	Reserved	-	-	保留
24	ENC	RW	0	编码器模式使能 ENC 位控制编码器模式 0：禁用编码器模式 1：启用编码器模式 注意：如果 LPTIM 不支持编码器模式功能，则该位保留。
23	COUNTMODE	RW	0	启用计数器模式 COUNTMODE 位选择 LPTIM 使用哪个时钟源为计数器计时： 0：计数器在每个内部时钟脉冲后递增 1：计数器在 LPTIM Input1 上的每个有效时钟脉冲后递增
22	PRELOAD	RW	0	寄存器更新模式。 预加载位控制 LPTIM_ARR 寄存器更新模式。 0:每次 APB 总线写访问后更新寄存器；

Puya Confidential

				1:寄存器在当前 LPTIM 计数周期结束时更新； 注：本寄存器仅针对 LPTIM_ARR 寄存器。
21	WAVPOL	RW	0	波形极性 WAVPOL 位控制输出极性 0：LPTIM 输出反映了 LPTIM_CNT 和 LPTIM_CMP 寄存器之间的比较结果 1：LPTIM 输出反映了 LPTIM_CNT 和 LPTIM_CMP 寄存器之间比较结果相反的值
20	WAVE	RW	0	波形形状. WAVE 位控制输出形状 0：停用一次设置模式，PWM 还是单脉冲波形具体取决于定时器的启动方式（用于 PWM 的 CNTSTRT 或用于单脉冲波形的 SNGSTRT）。 1：激活一次设置模式
19	TIMOUT	RW	0	TIMOUT 使能 TIMOUT 位控制超时功能 0：定时器已经启动时到达的触发事件将被忽略 1：定时器已经启动时到达的触发事件将复位并重新启动计数器
18:17	TRIGEN[1:0]	RW	0	触发使能和极性 TRIGEN 位控制 LPTIM 计数器是否由外部触发启动。如果选择了外部触发选项，则触发有效边沿可以采用 2 种配置： 00：软件触发（计数启动由软件发起） 01：上升沿为有效沿 10：下降沿为有效沿 11：保留
16	Reserved	-	-	保留
15:13	TRIGSEL[2:0]	RW	0	触发选择器 TRIGSEL 位在以下 8 个可用源中选择将作为 LPTIM 触发事件的触发源： 000：lptim_ext_trig0（LPTIM1_ETR for lptim1） 001：lptim_ext_trig1（RTC 闹钟） 010：lptim_ext_trig2（COMP1_OUT） 011：lptim_ext_trig3（COMP2_OUT） 100：lptim_ext_trig4（COMP3_OUT） 101：lptim_ext_trig5（COMP4_OUT） 110：lptim_ext_trig6（保留） 111：lptim_ext_trig7（保留）
12	Reserved	-	-	保留
11:9	PRESC[2:0]	RW	0	时钟预分频器。 PRESC 位配置预分频器分频系数。分频系数可从以下分频系数中选择: 000:/1； 001:/2； 010:/4； 011:/8；

Puya Confidential

				100:/16； 101:/32； 110:/64； 111:/128；
8	Reserved	-	-	保留
7:6	TRGFLT[1:0]	RW		触发的可配置数字滤波器 TRGFLT 值设置当内部触发器上发生电平变化时应检测到的连续相等采样值的数量，然后才将其视为有效电平转换。必须存在内部时钟源才能使用此功能。 00：任何触发活动电平变化都被视为有效触发 01：触发有效电平变化必须稳定至少 2 个时钟周期才被认为是有效触发。 10：触发有效电平变化必须稳定至少 4 个时钟周期才被认为是有效触发。 11：触发有效电平变化必须稳定至少 8 个时钟周期才被认为是有效触发。
5	Reserved	-	-	保留
4:3	CKFLT[1:0]	RW	0	外部时钟的可配置数字滤波器 CKFLT 值设置在外部时钟信号上发生电平变化时应检测到的连续相等采样数，然后才将其视为有效电平转换。必须存在内部时钟源才能使用此功能 00：任何外部时钟信号电平变化都被视为有效跳变 01：外部时钟信号电平变化必须至少稳定 2 个时钟周期才被认为是有效跳变。 10：外部时钟信号电平变化必须至少稳定 4 个时钟周期才被认为是有效跳变。 11：外部时钟信号电平变化必须至少稳定 8 个时钟周期才被认为是有效跳变。
2:1	CKPOL[1:0]	RW	0	处于活动状态编码器模式选择。 00：如果 LPTIM 配置为编码器模式(置位 ENC 位)，则编码器子模式 1 处于活动状态。 01：如果 LPTIM 配置为编码器模式(置位 ENC 位)，则编码器子模式 2 处于活动状态。 10：如果 LPTIM 配置为编码器模式(置位 ENC 位)，则编码器子模式 3 处于活动状态。 11：保留 注意：如果 LPTIM 不支持编码器模式功能，则该位域保留。
0	CKSEL	RW	0	时钟选择器 CKSEL 位选择 LPTIM 将使用的时钟源： 0：LPTIM 由内部时钟源（APB 时钟或任何嵌入式振荡器）提供时钟 1：LPTIM 由外部时钟源通过 LPTIM Input1 提供时钟

注：配置 CFGR 寄存器时建议一次写所有配置位，若分开配置时，两次配置之间需要间隔 3 个内核时钟。

Puya Confidential

22.6.5 LPTIM 控制寄存器（LPTIM_CR）

地址偏移：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
											RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:5	Reserved	-	-	保留
4	RSTARE	RW	0	读取后复位使能 此位由软件置 1 和清 0。当 RSTARE 设置为“1”时，对 LPTIM_CNT 的任何读取访问寄存器将异步重置 LPTIM_CNT 寄存器内容。
3	COUNTRST	RW	0	计数器复位。 该位由软件置 1，硬件清 0。设置为“1”时，此位将触发 LPTIM_CNT 计数寄存器同步复位。由于此复位的同步特性，它只需要在同步延迟 3 个 LPTIM 内核时钟周期之后释放（LPTIM 内核时钟可能是与 APB 时钟不同）。 注：在 COUNTRST 已被硬件清除为“0”之前，软件绝不能将其设置为“1”.因此，软件在尝试将其设置为“1”之前，应检查 COUNTRST 位是否已清零为“0”
2	CNTSTRT	RW	0	定时器启动连续模式。 该位由软件置位。 如果在进行单次计数模式计数时该位被置 1，则定时器不会在下一个 LPTIM_ARR 和 LPTIM_CNT 寄存器匹配的脉冲模式计数时停止。LPTIM 计数器保持在连续模式下计数。 注：仅当 LPTIM 使能时，此位才能置 1。该位在同步到内核时钟后由硬件清零。
1	SNGSTRT	RW	0	LPTIM 启动单次模式。 该位由软件置位，由硬件清零。该位置 1 将以单脉冲模式启动 LPTIM。 注：仅当 LPTIM 使能时，此位才能置 1。它将由硬件自动复位。
0	ENABLE	RW	0	LPTIM 使能位,由软件设置和清零 0：LPTIM 禁用 1：LPTIM 使能

22.6.6 LPTIM 比较寄存器（LPTIM_CMP）

地址偏移：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CMP	RW	0x0000	比较值。 CMP 是 LPTIM 的比较值 当 LPTIM 使能后才能更新该寄存器。

22.6.7 LPTIM 自动重载寄存器（LPTIM_ARR）

地址偏移：0x018

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	ARR[15:0]	RW	16’h1	自动重新加载值 ARR 是 LPTIM 的自动重载值。 当 LPTIM 使能后才能更新该寄存器。

22.6.8 LPTIM 计数器寄存器（LPTIM_CNT）

地址偏移：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CNT[15:0]	R	0	计数器值 当 LPTIM 以异步时钟运行时，读取 LPTIM_CNT 寄存器可能返回不可靠的值。因此在这种情况下，有必要执行两次连续的读访问并验证返回的两个值是否相同。需要注意的是，对于可靠的 LPTIM_CNT 寄存器读访问，需要两次连续的读操作并比较，相等时，可以认为读取访问是可靠的。

Puya Confidential

22.6.9 LPTIM 选择寄存器（LPTIM_OR）

地址偏移：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN1[2:1]		IN1[2:1]		IN2[0]	IN1[0]
										RW	RW	RW	RW	RW	RW

Bit	Name	RW	ResetValue	Function
31:16	Reserved	-	-	保留
5:4	IN2[2:1]	RW	0	IN2SEL[1:0]：LPTIM 输入 2 选择 IN2SEL 位控制 LPTIM 输入 2 多路复用器，它将 LPTIM 输入 2 连接到以下可用输入之一。 00：COMP1 01：COMP2 其它：保留
3:2	IN1[2:1]	RW	0	IN2SEL[1:0]：LPTIM 输入 1 选择 IN2SEL 位控制 LPTIM 输入 1 多路复用器，它将 LPTIM 输入 1 连接到以下可用输入之一。 00：COMP1 01：COMP2 其它：保留
1	IN2[0]	RW	0	LPTIM 输入 2 重映射 1：连接到 IN2[2:1]指示的 COMP 输出 0：连接到 GPIO
0	IN1[0]	RW	0	LPTIM 输入 1 重映射 1：连接到 IN1[2:1]指示的 COMP 输出 0：连接到 GPIO

Puya Confidential

23. 实时时钟（RTC）

23.1 RTC 简介

实时时钟是一个独立的定时器。RTC 模块拥有一组连续计数的计数器，在相应软件配置下，可提供时钟日历的功能。修改计数器的值可以重新设置系统当前的时间和日期。

RTC 模块和时钟配置系统（RCC_BDCR 寄存器）处于备份域，即在系统复位后，RTC 的设置和时间维持不变。系统复位后，对备份寄存器和 RTC 的访问被禁止，这是为了防止对备份域（BKP）的意外写操作。执行以下操作将使能对备份寄存器和 RTC 的访问：

- 设置寄存器 RCC_APBENR 的 PWREN 位，使能电源时钟
- 设置寄存器 PWR_CR1 的 DBP 位，使能对备份寄存器和 RTC 的访问。

RTC 内还包含了 5 个 32 位的备份寄存器，可用来存储 20 个字节的应用程序数据。备份寄存器处在备份域里。当系统复位时，它们也不会被擦除。

备份寄存器由上电复位（POR）或软件复位（BDRST）所复位，且在检测到侵入事件时，擦除备份寄存器内容。

备份寄存器受 RDP 保护，当 RDP 等级为 1 时，不可写入或读取备份寄存器，当 RDP 等级从 1 变为 0 时，擦除备份寄存器的所有数据。

此外，RTC 内还包含了用来管理侵入检测的寄存器和 RTC 校准功能控制寄存器

23.2 RTC 主要特性

主要功能如下：

- 可编程的预分频系数：分频系数最高为 2^{20}。
- 32 位的可编程计数器，可用于较长时间段的测量。
- 2 个分离的时钟：用于 APB1 接口的 PCLK1 和 RTC 时钟（RTC 时钟的频率必须小于 PCLK1 时钟频率的四分之一）。
- 可以选择以下三种 RTC 的时钟源：
  - HSE 时钟的 128/32/8 分频（默认 128，由 RCC_CCIPR2. RTC_HSEDIV_SEL[1:0]决定）
  - LSE 振荡器时钟
  - LSI 振荡器时钟
- 2 个独立的复位类型：
  - APB1 接口由系统复位
  - RTC 核心（预分频器、闹钟、计数器、分频器、侵入检测电路、校准电路、备份寄存器）只能由备份域复位。
- 3 个专门的可屏蔽中断：
  - 闹钟中断，用来产生一个软件可编程的闹钟中断。
  - RTC 全局中断，包含秒中断、闹钟中断、溢出中断。
  - 侵入事件中断
- 支持 20 字节数据备份寄存器

Puya Confidential

- 拥有 1 个用来存储 RTC 时钟的校准值的校准寄存器
- PC13 引脚可用于侵入检测
- 可在 PC13 引脚（当该引脚不用于侵入检测时）上输出 RTC 校准时钟，RTC 闹钟脉冲或者秒脉冲
- 备份寄存器受 Flash 的 RDP 保护

23.3 RTC 功能描述

23.3.1 概述

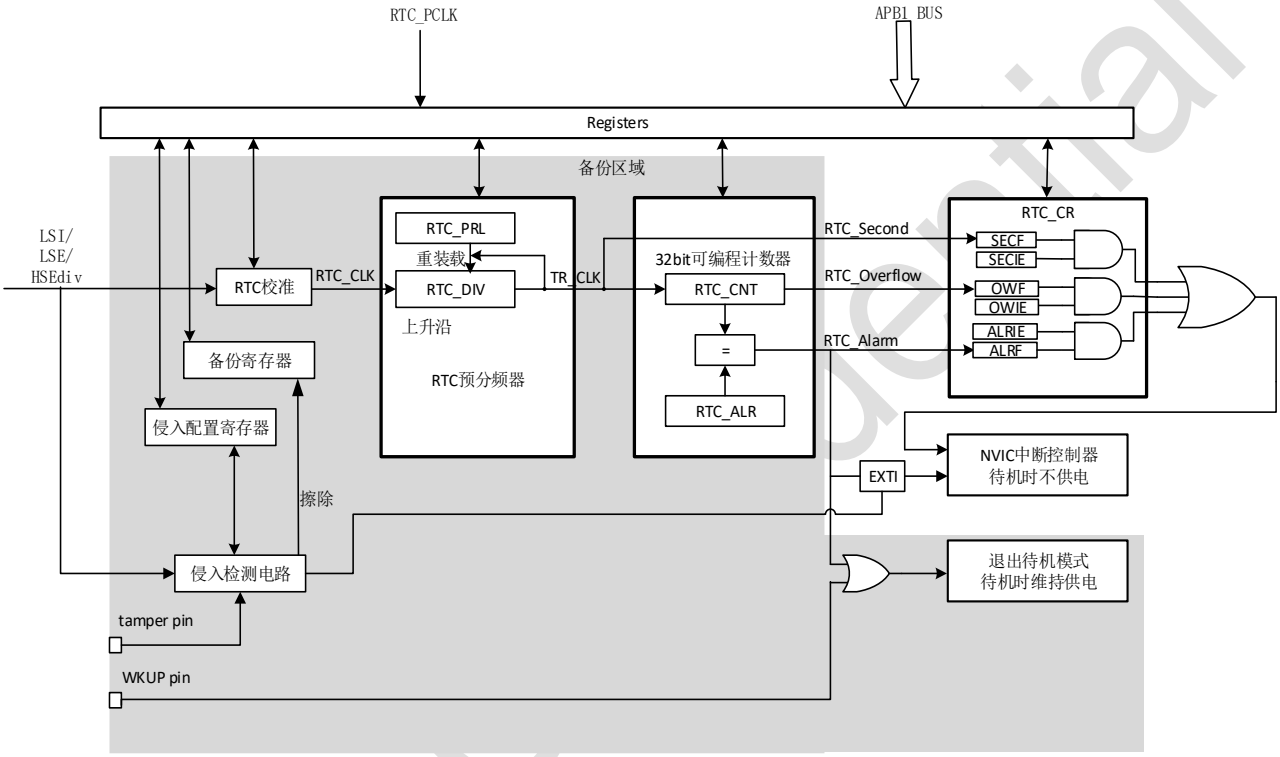


图 23-1 RTC 功能框图

RTC 由两个主要部分组成（参见上图）。第一部分（APB1 接口）用来和 APB1 总线相连。此单元还包含一组 32 位寄存器（RTC_CR，实际分散在两个地址的寄存器），可通过 APB1 总线对其进行读写操作。APB1 接口由 APB1 总线时钟驱动，用来连接 APB1 总线。

另一部分（RTC 核心）由一组可编程计数器组成，分成两个主要模块。

第一个模块是 RTC 的预分频器模块，它可编程产生最长为 1 秒的 RTC 时间基准 TR_CLK。RTC 的预分频模块包含了一个 20 位的可编程分频器（RTC 预分频器）。如果在 RTC_CR 寄存器中设置了相应的允许位，则在每个 TR_CLK 周期中 RTC 产生一个中断（秒中断）（second）。

第二个模块是一个 32 位的可编程计数器，可被初始化为当前的系统时间。系统时间按 TR_CLK 周期累加并与存储在 RTC_ALR 寄存器中的可编程时间相比较，比较匹配时将产生一个闹钟中断（alarm）。

侵入检测电路、RTC 时钟校准电路、备份寄存器也包含在了 RTC 核心里，在停止状态下也能正常工作。

闹钟中断作为停止模式的唤醒信号。在停止模式，RTC 闹钟作为唤醒需要配置 EXTI line17 对应寄存器。

注意：RTC 全局中断和侵入中断并不能唤醒停止模式。

23.3.2 寄存器复位

RTC_PRL，RTC_ALR，RTC_CNT，RTC_DIV，RTC_TMPCFGR，RTC_TMPCSR，RTC_CALIBR，RTC_BKPDRx 寄存器仅能通过备份域复位信号复位，RTC_CR 则由系统复位或电源复位进行异步复位。

23.3.3 读 RTC 寄存器

RTC 核完全独立于 RTC APB1 接口。

软件通过 APB1 接口访问 RTC 的预分频值、计数器值和闹钟值。但是，相关的可读寄存器只在与 RTC 时钟的上升沿同步到 RTC APB1 时钟后的信号有效时更新。RTC 标志也是如此的。

这意味着，如果 APB1 接口曾经被关闭，而读操作又是在刚刚重新开启 APB1 之后，则在第一次的内部寄存器更新之前，从 APB1 上读出的 RTC 寄存器数值可能被破坏了（通常读到 0）。下述几种情况下能够发生这种情形：

- 发生系统复位或电源复位
- 系统刚从停机模式唤醒 (这种情况计数值只是不会更新，因为 CPU 不工作，系统时钟停止，但 RTC 正常计数，计数值不会同步到 V_{DDD} 区)

所有以上情况中，APB1 接口被禁止时（复位、无时钟或断电）RTC 核仍保持运行状态。

因此，若在读取 RTC_CNT，RTC_DIV 时，如果 RTC 的 APB1 接口曾经处于禁止状态，则软件首先必须等待 RTC_CR 寄存器中的 RSF 位（寄存器同步标志）被硬件置'1'。

说明：

RTC_CR 寄存器为 RTC_PCLK 域，CPU 任何时候读能读到稳定值

RTC_CNT 和 RTC_DIV 来源于 RTC_CLK 域。在 RTC 工作后，RTC_DIV 寄存器在每个 RTC_CLK 的上升沿更新；RTC_CNT 和来源于 RTC_CLK 时钟域的标志位同样采用跟 RTC_DIV 寄存器同样的更新信号，虽然这样不是每次更新时 RTC_CNT 的值都改变；

RSF 实现在 RTC_PCLK 域，在 RTC_CLK 同步到 RTC_PCLK 后的脉冲信号有效时置位；

RSF 仅控制 RTC_CNT 和 RTC_DIV 的读取时机（硬件不会控制）

读侵入检测、备份寄存器（RTC_TMPCFGR、RTC_TMPCSR、RTC_BKPDRx）时，无需考虑 RSF 位的值。

23.3.4 配置 RTC 寄存器

必须设置 RTC_CR 寄存器中的 CNF 位，使 RTC 进入配置模式后，才能写入 RTC_PRL、RTC_CNT、RTC_ALR、RTC_CALIBR 寄存器。

另外，对 RTC_PRL、RTC_CNT 、RTC_ALR、RTC_CALIBR 写操作，都必须在前一次写操作结束后进行。可以通过查询 RTC_CR 寄存器中的 RTOFF 状态位，判断 RTC 寄存器是否处于更新中。仅当 RTOFF 状态位是'1'时，才可以写入 RTC 寄存器。

写侵入检测、备份寄存器（RTC_TMPCFGR、RTC_TMPCSR、RTC_BKPDRx）时，无需考虑 CNF 位的值。

配置过程：

- 查询 RTOFF 位，直到 RTOFF 的值变为'1'；(表明前面配置已经完成)
- 置 CNF 值为 1，进入配置模式；(此时 RTOFF 位仍为 1，保证 CPU 在写寄存器时 RTOFF=1)
- 对一个或多个 RTC 寄存器进行写操作；(RTOFF 在此过程仍为 1，RTC_CLK 域寄存器此步骤操

Puya Confidential

作写入 buffer 寄存器）

- 清除 CNF 标志位，退出配置模式；（硬件检测到 CNF 清零后，开始执行上一步骤中的写寄存器操作，开始写入 RTC_CLK 域的寄存器；同时 RTOFF 被清零）
- 查询 RTOFF，直至 RTOFF 位变为’1’以确认写操作已经完成。（buffer 寄存器写入 RTC_CLK 域后置位 RTOFF）

注：仅当 CNF 标志位被清除时，写操作才能进行，这个过程至少需要 3 个 RTC_CLK 周期。(在清除 CNF 标志位后 3 个 RTC_CLK 不能重新启动配置，否则会出现配置错误（此时通过 RTOFF=0 控制））

说明：

1. 在此过程中，CPU 写寄存器时 RTOFF=1；
2. CPU 的写周期为从 CNF=1 到 CNF=0，配置其他寄存器在这两个操作之间；
3. 先将 CNF 写 1 后将 CNF 清零，这个操作清零 RTOFF；只写 CNF=0 或者写 CNF=1 后不清零不会清零 RTOFF；
4. 先将 CNF 写 1 后将 CNF 清零，这个操作启动 buffer 寄存器写 RTC_CLK 域寄存器的过程；
5. RTOFF 实现在 RTC_PCLK 域。

23.3.5 RTC 标志设置

在每一个 RTC 核心的时钟周期中，更改 RTC 计数器之前设置 RTC 秒标志（SECF）。

在计数器到达 0x0000 之前的最后一个 RTC 时钟周期中，设置 RTC 溢出标志（OWF）。（达到 0 之前即为检测到 RTC_CNT 的最大值置位 OWF）

在计数器的值到达闹钟寄存器的值加 1(RTC_ALR+1)之前的 RTC 时钟周期中，设置 RTC_Alarm 和 RTC 闹钟标志(ALRF)。（检测到 RTC_CNT 计数到 RTC_ALR，且在最后一个 RTC_CLK 置位 ALRF）

对 RTC 闹钟和 RTC 秒标记的写操作必须通过以下步骤之一同步：

- 使用 RTC 闹钟中断，并在中断处理程序中修改 RTC 闹钟寄存器（RTC_ALR）和/或 RTC 计数器寄存器（RTC_CNT）。
- 等待 RTC 控制寄存器中的 SECF 位置 1，再更改 RTC 闹钟寄存器（RTC_ALR）和/或 RTC 计数器寄存器（RTC_CNT）。

23.3.6 RTC 时序

RTC 秒和闹钟时序

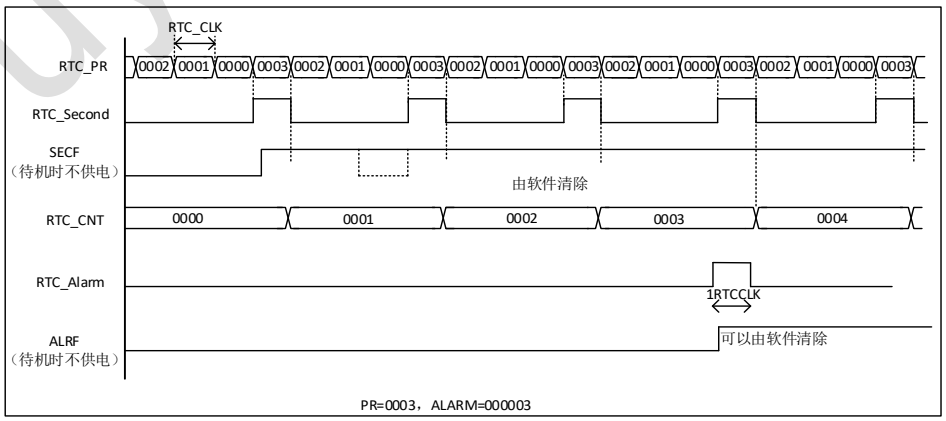


图 23-2 RTC 秒和闹钟时序图

Puya Confidential

SECF 和 ALRF 为 RTC_PCLK 域信号，分别采样 RTC_CLK 域 RTC_Sectond 和 RTC_Alarm 信号产生。

RTC 溢出时序

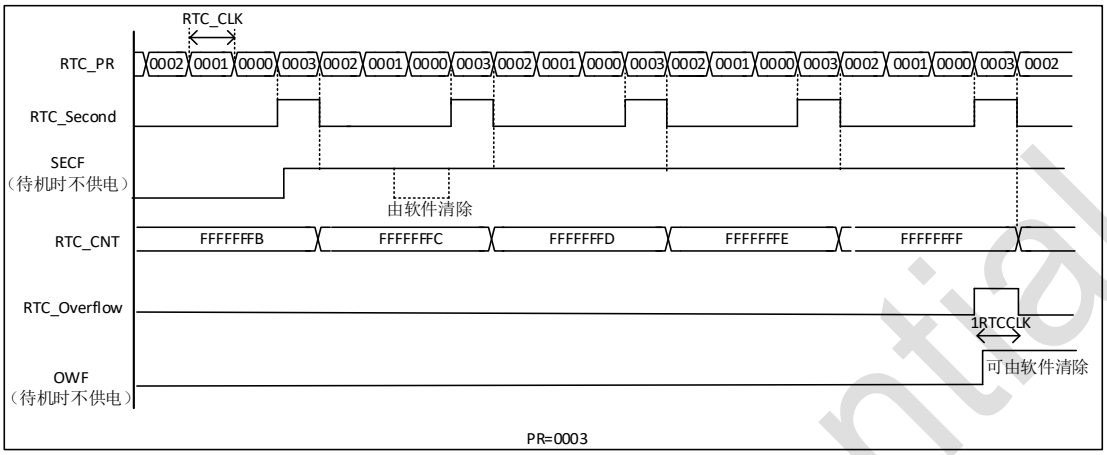


图 23-3 RTC 溢出时序图

SECF 和 OWF 为 RTC_PCLK 域信号，分别采样 RTC_CLK 域 RTC_Sectond 和 RTC_Overflow 号产生。

23.3.7 侵入检测

当侵入检测引脚（PC13）上的信号从 0 变成 1 或者从 1 变成 0（取决于 RTC 入侵配置寄存器 RTC_TMPCFGR 的 TPAL 位），会产生一个侵入检测事件。侵入检测事件将所有数据备份寄存器内容清除。然而为了避免丢失侵入事件，侵入检测信号是边沿检测的信号与侵入检测允许位的逻辑‘与’，从而在侵入检测引脚被允许前发生的侵入事件也可以被检测到。

- 当 TPAL=0 时：如果在使能侵入检测前（通过设置 TPE 位），PC13 已经为高电平，一旦启动侵入检测功能，则会产生一个额外的侵入事件（尽管在 TPE 位置‘1’后并没有出现上升沿）。
- 当 TPAL=1 时：如果在使能侵入检测前（通过设置 TPE 位），PC13 已经为低电平，一旦启动侵入检测功能，则会产生一个额外的侵入事件（尽管在 TPE 位置‘1’后并没有出现下降沿）。

设置 RTC_TMPCSR 寄存器的 TPIE 位为‘1’，当检测到侵入事件时就会产生一个中断。在一个侵入事件被检测到并被清除后，应关闭侵入检测使能，然后在再次写入备份数据寄存器前重新用 TPE 位启动侵入检测功能。这样，可以阻止软件在侵入检测引脚上仍然有侵入事件时对备份数据寄存器进行写操作。这相当于对侵入引脚进行电平检测。

注：

当 V_{CC} 电源断开时，侵入检测功能仍然有效。为了避免不必要的复位数据备份寄存器，侵入检测引脚应该在片外连接到正确的电平。

侵入检测开始后，PC13 引脚的电平会持续被 RTC_CLK 采样。故开启侵入检测前，必须先开启 RTC 时钟。

23.3.8 RTC 校准

为方便测量，RTC 时钟可以经 64 分频输出到侵入检测引脚(PC13)上。通过设置 RTC_CALIBR 的 CCO 位来开启这一功能。

Puya Confidential

通过配置 CAL[6:0]位，此时钟可以最多减慢 121 ppm。

概述

实时时钟(RTC)精度是大多数嵌入式应用程序的要求，但由于外部环境温度变化，时钟的晶体频率变化引起 RTC 精度可能不像预期的那样准确。使用数字时钟校准电路，允许应用程序补偿晶体和温度变化。

RTC 校准方式

RTC 时钟可选择一个额定频率为 32.768 kHz 的石英晶体控制振荡器驱动。晶体振荡器是提供固定频率的最精确电路之一。时钟误差有两个原因：

1. 温度变化
2. 晶体变化

如前所述，大多数时钟芯片通过使用繁琐的微调电容器来补偿晶体频率和温度变化。设计采用周期性计数器校正。数字校准电路每 220 个时钟周期消除 0 至 127 个周期。脉冲被消隐的个数取决于加载到 BKP 的 RTC 时钟校准寄存器 BKP_RTCCR.CAL 的值。

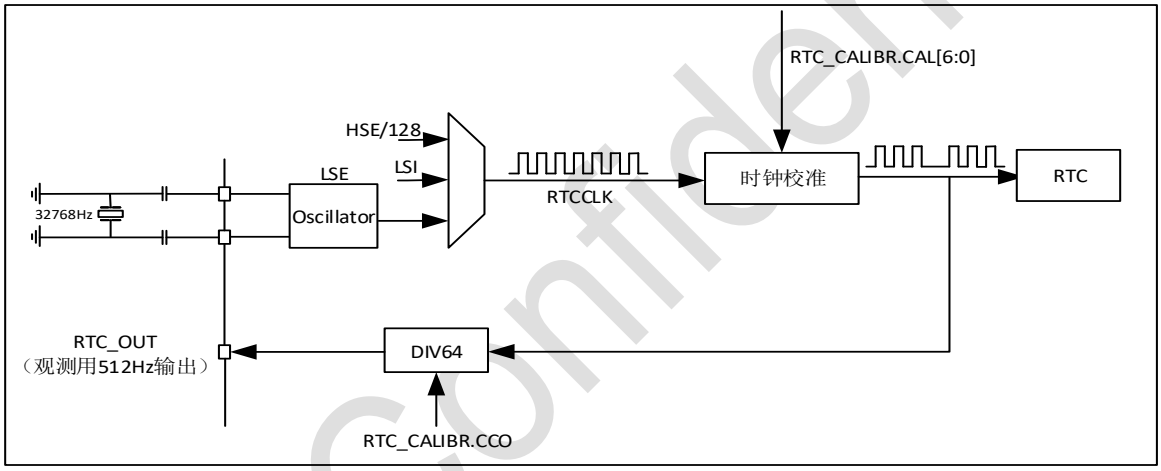


图 23-4 RTC 校准示意图

每个校准步骤的效果是每 1 048 576（220）实际振荡器周期减去 1 个振荡器周期。即校准寄存器中每个校准步骤的调整量为 0.954（1000000/220）ppm。因此，振荡器时钟可以从 0 放慢到 121 ppm。下表显示了每个位在每个月（30 天）中实时表示的 ppm 和秒数。

表 23-1 ppm 计算对照表

校准值	四舍五入到最接近的 ppm	值以秒为单位，每月(30 天)四舍五入到最近的秒	校准值	四舍五入到最接近的 ppm	值以秒为单位，每月(30 天)四舍五入到最近的秒
0	0	0	64	61	158
1	1	2	65	62	161
2	2	5	66	63	163
3	3	7	67	64	166
4	4	10	68	65	168
5	5	12	69	66	171
6	6	15	70	67	173
7	7	17	71	68	176
8	8	20	72	69	178
9	9	22	73	70	180

Puya Confidential

10	10	25	74	71	183
11	10	27	75	72	185
12	11	30	76	72	188
13	12	32	77	73	190
14	13	35	78	74	193
15	14	37	79	75	195
16	15	40	80	76	198
17	16	42	81	77	200
18	17	44	82	78	203
19	18	47	83	79	205
20	19	49	84	80	208
21	20	52	85	81	210
22	21	54	86	82	213
23	22	57	87	83	215
24	23	59	88	84	218
25	24	62	89	85	220
26	25	64	90	86	222
27	26	67	91	87	225
28	27	69	92	88	227
29	28	72	93	89	230
30	29	74	94	90	232
31	30	77	95	91	235
32	31	79	96	92	237
33	31	82	97	93	240
34	32	84	98	93	242
35	33	87	99	94	245
36	34	89	100	95	247
37	35	91	101	96	250
38	36	94	102	97	252
39	37	96	103	98	255
40	38	99	104	99	257
41	39	101	105	100	260
42	40	104	106	101	262
43	41	106	107	102	264
44	42	109	108	103	267
45	43	111	109	104	269
46	44	114	110	105	272
47	45	116	111	106	274
48	46	119	112	107	277
49	47	121	113	108	279
50	48	124	114	109	282
51	49	126	115	110	284
52	50	129	116	111	287
53	51	131	117	112	289
54	51	133	118	113	292
55	52	136	119	113	294
56	53	138	120	114	297

Puya Confidential

57	54	141	121	115	299
58	55	143	122	116	302
59	56	146	123	117	304
60	57	148	124	118	307
61	58	151	125	119	309
62	59	153	126	120	311
63	60	156	127	121	314

如上所述，RTC 时钟校准电路只从晶体时钟中减去周期。基于 RTC 预分频器值默认设置为 32768，更快的晶体频率(>32768 Hz)可以被校准，而较慢的晶体频率（<32768 Hz）不能被补偿（只校准变快的晶体，变慢的会加剧变慢的情况）。所以只有晶体频率量程[32 772,32 768]可以校准。

由于晶体频率可能在 32.768 kHz 附近变化，比如可以通过设置 RTC 预分频为 32766（而不是 32768）来调整晶体频率由 32 768 为 32 766。这样，可以补偿晶体频率为[32770,32766]的范围。

计算所需的校准量

为了确定在给定的应用程序中需要多少校准，需要根据具体情况，借助 RTC 时钟的输出功能。RTC 时钟输出可以用来测量晶体振荡器的精度，如果晶体振荡器的频率是精确的 32768 Hz，则 RTC 时钟输出的频率正好是 512 Hz。

该方法可分为以下步骤:

1.启用低速外部振荡器（LSE），选择 LSE 作为 RTC 时钟源，然后启用 RTC 时钟。

2.在 RTC_OUT 引脚 (PC13) 上输出 RTC 时钟的 64 分频率，用于晶体频率测量。通过将 BKP_RTCCR 中的 CCO 位置 1 来实现。

3.计算晶体频率偏差(ppm)。假设 RTC 预分频器值为 32766，通过将 511.968 Hz (32766/64≈511.968) 的测量偏差除以 511.968，并将结果乘以 100 万，就可以快速计算出 ppm 中的偏差。使用表 1 找到最近的校准值。此表是基于 ppm 表示的变化值的校准值的直接查找表。

4.向 RTC 校准寄存器中写入校准值以补偿晶体偏差。

注意:要将 RTC 预分频器设置为 32766，请将 32765 写入 RTC 预分频器寄存器。

例如，如果在实测中测量到的频率为 511.982 Hz，则 δ 为 0.014。除以 511.968，再乘以 100 万，结果是 27.35 ppm。在这种情况下，最接近的补偿值是 28。误差将从 27.35 ppm（每月~71 秒）减少到 0.65 ppm（每月~1.7 秒）。

注意:由于 RTC 校准的原理是忽略部分 RTC 时钟脉冲，它不能改善短时间内的计数，它只改善长时间内的计数。例如，在没有校准的情况下，使用 RTC 计数 0.01 秒会比有校准的情况下更准确。由于校准周期在考虑的时间范围内可能发生或不发生，因此结果值可能会发生显著变化。因此，根据应用情况，最好不要使用校准。

23.3.9 备份域寄存器擦除

以下方式可以使得备份寄存器被擦除：

①电复位

②寄存器 RCC_BDCR 的 BDRST（备份域软复位）被写入 1

③RDP level 从 1 变为 0

④TEF 标志位为 1(检测到侵入事件)

Puya Confidential

23.4 RTC 寄存器

寄存器宽度为 32 位，只允许字操作。

23.4.1 RTC 控制寄存器高位（RTC_CRH）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OWIE	ALRIE	SECIE
													RW	RW	RW

该寄存器由系统复位复位。

Bit	Name	RW	Reset Value	Function
31:3	Reserved	-	-	保留
2	OWIE	RW	0	溢出中断允许位 0：不允许溢出中断 1：允许溢出中断
1	ALRIE	RW	0	闹钟中断允许位 0：不允许闹钟中断 1：允许闹钟中断
0	SECIE	RW	0	秒中断允许位 0：不允许秒中断 1：允许秒中断

这些位用于屏蔽中断请求。注意：在复位后，所有中断是未使能的，所以在初始化后，写 RTC 寄存器以确保没有正在挂起的中断请求是可能的。但是当外设正在完成前一次的写操作（RTOFF=0）时，是不能写 RTC_CRH 寄存器的。

该控制寄存器控制着 RTC 的功能。某些位必须使用专门的配置流程才能进行写操作。

23.4.2 RTC 控制寄存器低位（RTC_CRL）

偏移地址：0x04

复位值：0x0000 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTOFF	CNF	RSF	OWF	ALRF	SECF
										R	RW	RC_W0	RC_W0	RC_W0	RC_W0

Bit	Name	RW	Reset Value	Function
31:6	Reserved	-	-	保留
5	RTOFF	R	1	RTC 操作关闭（RTC operation OFF），该位只读。 RTC 模块利用该位来指示对其寄存器进行的最后一次操作的状态（指示操作是否完成）。 若此位为'0'，则表示无法对任何的 RTC 寄存器进行写操作。 0：上一次对 RTC 寄存器的写操作仍在进行 1：上一次对 RTC 寄存器的写操作已经完成

Puya Confidential

4	CNF	RW	0	配置标志（Configuration flag）。 此位必须由软件置’1’以进入配置模式，从而允许向 RTC_CNT、RTC_ALR、RTC_PRL、RTC_CALIBR 寄存器写入新值。 只有当此位在被置’1’，并重新由软件清’0’后，才会执行写操作。 0：退出配置模式(开始更新 RTC 寄存器) 1：进入配置模式
3	RSF	RC_W0	0	寄存器同步标志（Registers synchronized flag).该寄存器由硬件置位，软件清零。当 RTC_CNT 寄存器和 RTC_DIV 寄存器更新后，硬件置’1’该位。 在 APB1 复位后，或 APB1 时钟停止后，此位必须由软件清’0’。 要进行任何的读操作之前，用户程序必须等待该位被硬件置’1’，以确保 RTC_CNT、RTC_PRL 已经被同步。 0：寄存器尚未被同步 1：寄存器已经被同步
2	OWF	RC_W0	0	溢出标志（Overflow flag). 当 32 位可编程计数器溢出时，此位由硬件置’1’。如果 RTC_CRH 寄存器中 OWIE=1，则产生中断。 此位只能由软件清’0’，写’1’无效。 0：无溢出； 1：32 位可编程计数器溢出
1	ALRF	RC_W0	0	闹钟标志（Alarm flag)。 当 32 位可编程计数器达到 RTC_ALR 寄存器所设置的预定值，此位由硬件置’1’。如果 RTC_CRH 寄存器中 ALRIE=1，则产生中断。此位只能由软件清’0’，写’1’无效。 0：无闹钟； 1：有闹钟。
0	SECF	RC_W0	0	秒标志（Second flag)。 当 32 位可编程预分频器溢出时，此位由硬件置’1’，同时 RTC 计数器加 1。 因此，此标志为分辨率可编程的 RTC 计数器提供一个周期性的信号(通常为 1 秒)。如果 RTC_CRH 寄存器中 SECIE=1，则产生中断。此位只能由软件清除，写’1’无效。 0：秒标志条件不成立 1：秒标志条件成立

注：

- RTC 的功能是被该控制寄存器控制的。当外设正在继续上一次写操作时（RTOFF=0），是不能写 RTC_CRL 寄存器的。
- 任何标志位都将保持挂起状态，直到适当的 RTC_CR 请求位被软件复位，表示所请求的中断已经被接受。
- 在复位时禁止所有中断，无挂起的中断请求，可以对 RTC 寄存器进行写操作
- 当 APB1 时钟不运行时， OWF、 ALRF、 SECF 和 RSF 位不被更新（无法同步）。

Puya Confidential

- OWF、 ALRF、SECF 和 RSF 位只能由硬件置位，由软件来清零。
- 若 ALRF=1 且 ALRIE=1，则允许产生 RTC 全局中断。如果在 EXTI 控制器中允许产生 EXTI 线 17 中断，则允许产生 RTC 全局中断和 RTC 闹钟中断。
- 若 ALRF=1，如果在 EXTI 控制器中设置了 EXTI 线 17 的中断模式，则允许产生 RTC 闹钟中断；如果在 EXTI 控制器中设置了 EXTI 线 17 的事件模式，则这条线上会产生一个脉冲 (不会产生 RTC 闹钟中断)。

23.4.3 RTC 预分频装载寄存器高位（RTC_PRLH）

PRL 寄存器用来保存 RTC 预分频器周期性的计数值。该寄存器是被 RTC_CR 寄存器的 RTOFF 位写保护的，只有 RTOFF=1，才允许 CPU 进行写操作。

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRL[19:16]			
												W	W	W	W

Bit	Name	RW	Reset Value	Function
31:4	Reserved	-	-	保留
3:0	PRL[19:16]	W	4'h0	RTC 预分频装载值高位 根据以下公式，这些位用来定义计数器的时钟频率： $f_{TR_CLK} = f_{RTCCLK}/(PRL[19:0]+1)$ 注：不推荐使用 0 值，否则无法正确的产生 RTC 中断和标志位

23.4.4 RTC 预分频装载寄存器低位（RTC_PRLL）

偏移地址：0x0C

复位值：0x0000 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRL[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	PRL[15:0]	W	16'h8000	RTC 预分频装载值高位 (RTC prescaler reload value high)。根据以下公式，这些位用来定义计数器的时钟频率： $f_{TR_CLK} = f_{RTCCLK}/(PRL[19:0]+1)$ 注：不推荐使用 0 值，否则无法正确的产生 RTC 中断和标志位。

23.4.5 RTC 预分频余数寄存器高位（RTC_DIVH）

在每个 TR_CLK 周期，RTC_PRL 寄存器的值被重装载到 RTC 预分频计数器里。用户可以通过读取 RTC_DIV 寄存器，以获得预分频计数器的当前值，而不停止分频计数器的工作，从而获得精确的时间测量。

该寄存器只读属性，当 RTC_PRL 或者 RTC_CNT 寄存器的值发生任何变化，该寄存器值将由硬件重新装载。

Puya Confidential

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIV[19:16]			
												R	R	R	R

Bit	Name	RW	Reset Value	Function
31:4	Reserved	-	-	保留
3:0	DIV[19:16]	R	4’h0	RTC 时钟分频器余数高位。

23.4.6 RTC 预分频余数寄存器低位（RTC_DIVL）

偏移地址：0x14

复位值：0x0000 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	DIV[15:0]	R	16’h8000	RTC 时钟分频器

23.4.7 RTC 计数寄存器高位（RTC_CNTH）

RTC 模块有个 32bit 可编程的计数器，该寄存器通过两个寄存器访问，计数基于预分频器产生的 TR_CLK 时间基准为参考进行计数。

RTC_CNT 寄存器用来存放计数器的计数值。寄存器是被写保护的，仅当 RTOFF=1 时 CPU 才能进行写操作。对高 16bit 的 RTC_CNTH 或者低 16bit 的 RTC_CNTL 寄存器进行写操作，直接装载到相应的可编程计数器里，并重装载 RTC 预分频器。当读操作发生，返回计数器的当前值（系统时间）。

偏移地址：0x18

复位值：0x0000 0000

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

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CNT[31:16]	RW	16’h0000	RTC core counter 的高 16bit. 当读 RTC_CNTH 寄存器时，返回 RTC 计数器寄存器的当前值的高 16bit。只有进入配置模式才能对该寄存器进行写操作。

Puya Confidential

23.4.8 RTC 计数寄存器低位（RTC_CNTL）

偏移地址：0x1C

复位值：0x0000 0000

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

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CNT[15:0]	RW	16’h0000	RTC core counter 低 16bit 当读 RTC_CNTL 寄存器时，返回 RTC 计数器寄存器当前值的低 16bit。只有进入配置模式才能对该寄存器进行写操作。

23.4.9 RTC 闹钟寄存器高位（RTC_ALRH）

当可编程计数器（计数）达到存储在 RTC_ALR 寄存器的 32bit 值时，并产生 alarm 中断请求。该寄存器是被 RTOFF 位写保护的，只有 RTOFF=1，才允许写访问。

偏移地址：0x20

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALR[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	ALR[31:16]	RW	16’hFFFF	RTC 闹钟值高 16bit. 该寄存器用来保存由软件写入的闹钟时间的高 16bit。写该寄存器必须进入配置模式。

23.4.10 RTC 闹钟寄存器低位(RTC_ALRL)

偏移地址：0x24

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	ALR[15:0]	RW	16’hFFFF	RTC 闹钟值低 16bit. 该寄存器用来保存由软件写入的闹钟时间的高 16bit。写该寄存器必须进入配置模式。

Puya Confidential

23.4.11 RTC 入侵配置寄存器（RTC_TMPCFGR）

偏移地址：0x30

复位值：0x0000 0000

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

Bit	Name	RW	Reset Value	Function
31:2	Reserved	-	-	保留
1	TPAL	RW	0	侵入检测引脚(PC13)有效电平。 0：侵入检测引脚上的高电平会清除所有数据备份寄存器（TPE=1 时）； 1：侵入检测引脚上的低电平会清除所有数据备份寄存器（TPE=1 时）； 注：当 TPE=1 时，无法更改此位。
0	TPE	RW	0	启动侵入检测引脚。 0：侵入检测引脚作为通用 IO 口使用； 1：侵入检测引脚作为侵入检测使用；

23.4.12 RTC 入侵控制/状态寄存器（RTC_TMPCSR）

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TIF	TEF	Res.	Res.	Res.	Res.	Res.	TPIE	CTI	CTE
						R	R						RW	W	W

Bit	Name	RW	Reset Value	Function
31:10	Reserved	-	-	保留
9	TIF	R	0	侵入中断标志。 当检测到侵入事件且 TPIE 位为 1 时，此位由硬件置 1.通过向 CTI 位写 1 来清除此标志位（同时也清除中断）。如果 TPIE 位被清除，此位也会被清除。 0：无侵入中断 1：产生侵入中断 注： 此标志能被系统复位清零,但不能被备份域复位清零。
8	TEF	R	0	侵入事件标志。 当检测到侵入事件时此位由硬件置 1.通过向 CTE 位写 1 可清除此标志位。 0：无侵入事件 1：检测到侵入事件 注： 侵入事件会复位所有的 RTC_BKPDRx 寄存器。
7:3	Reserved	-	-	保留

Puya Confidential

2	TPIE	RW	0	允许侵入检测引脚中断。 0：禁止侵入检测中断 1：允许侵入监测中断（RTC_TMPCFGR 寄存器的 TPE 位也必须被置 1） 注: 此标志能被系统复位清零,但不能被备份域复位清零.
1	CTI	W	0	清除侵入检测中断。 此位只能写入，读出值为 0. 0：无效 1：清除侵入检测中断和 TIF 侵入检测中断标志
0	CTE	W	0	清除侵入检测事件。 此位只能写入，读出值为 0. 0：无效 1：清除 TEF 侵入检测事件标志（并复位侵入检测器）

23.4.13 RTC 时钟校准寄存器(RTC_CALIBR)

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	ASOS	ASOE	CCO	CAL[6:0]						
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:10	Reserved	-	-	保留
9	ASOS	RW	0	闹钟或者秒输出选择。 当设置了 ASOE 位，ASOS 位可以用于选择在侵入检测引脚上输出的是 RTC 秒脉冲还是闹钟脉冲信号。 0：输出 RTC 闹钟脉冲 1：输出秒脉冲 注：该位只能被备份区复位清零。
8	ASOE	RW	0	允许输出闹钟或者秒脉冲。 根据 ASOS 位配置，该位允许 RTC 闹钟或秒脉冲输出到侵入检测引脚上。 在 ASOE 置 1 之前，应先将 TPE 置 0，否则可能会错误地产生侵入事件。 注：该位只能被备份区复位清零。
7	CCO	RW	0	校准时钟输出。 0：无影响； 1：在侵入检测引脚输出校准后的 RTCCLK 64 分频时钟。 在 CCO 置 1 之前，应先将 TPE 置 0，否则可能会错误地产生侵入事件
6:0	CAL[6:0]	RW	7'h0	校准值。 校准值表示在每 2^{20} 个时钟脉冲内将有多少个时钟脉冲被跳过。这可以用来对 RTC 进行校准，以 1000000/2^{20}ppm 的比例减慢时钟。

Puya Confidential

				RTC 时钟可以被减慢 0 ~ 121ppm。

注：1.当 CCO 和 ASOE 同时配置时，ASOE 优先级高。

2.写该寄存器必须进入配置模式

23.4.14 RTC 备份寄存器（RTC_BKPDR）

偏移地址：0x40, 0x44, 0x48, 0x4C, 0x50

复位值：0x0000 0000

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

Bit	Name	RW	Reset Value	Function
31:0	D[31:0]	RW	32’b0	备份数据。这些位可以被用来写入用户数据。 注：RTC_BKPDRx 寄存器不会被系统复位。可以由备份域复位来复位或（如果侵入检测功能被开启时）由侵入事件复位。

Puya Confidential

24. 独立看门狗（IWDG）

24.1 IWDG 简介

独立看门狗寄存器（简称 IWDG），该模块具有高安全级别、时序精确及灵活使用的特点。IWDG 可用来检测和解决由软件错误引起的故障，并在计数器达到指定的超时值时（TIMEOUT）触发系统复位。

独立看门狗（IWDG）由专用的低速内部时钟（LSI）驱动，即使主时钟发生故障它也仍然有效。

IWDG 最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的场合。

24.2 IWDG 主要特性

- 自由运行的递减计数器
- 时钟由独立的 RC 振荡器提供（可在停止模式和待机模式下工作）
- 看门狗被激活后，则在计数器递减至 0x000 时产生复位
- 具有低功耗冻结功能：在停止模式和待机模式下保持工作状态，计数器保持递减计数
- 具有硬件看门狗功能：在用户选项字节里将 IWDG 配置成硬件启动模式后，IWDG 会在复位后自动使能
- 可自动使能 LSI：IWDG 使能后，自动使能 LSI，并作为 IWDG 的内核时钟

24.3 IWDG 功能说明

24.3.1 模块框图

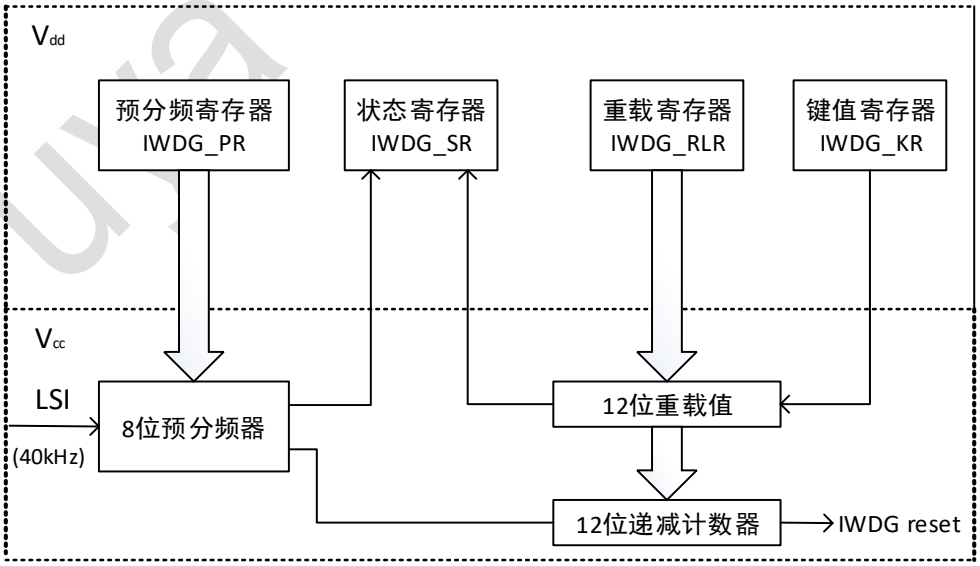


图 24-1 IWDG 模块框图

注：看门狗功能处于 V_{CC} 供电区，即在停机和待机模式时仍能正常工作。

Puya Confidential

在键寄存器（IWDG_KR）中写入 0xCCCC，开始启用独立看门狗；此时计数器开始从其复位值 0xFFF 递减计数。当计数器计数到末尾 0x000 时，会产生一个复位信号(IWDG_RESET)。

无论何时，只要键寄存器 IWDG_KR 中被写入 0xAAAA，IWDG_RLR 中的值就会被重新加载到计数器中从而避免产生 IWDG 复位 。

表 24-1 看门狗超时时间 32.768 kHz 的输入时钟(LSI)

预分频系数	PR[2:0]位	最短时间(ms) PR[11:0] = 0x000	最长时间(ms) PR[11:0] = 0xFFF
/4	0	0.12207	500
/8	1	0.24414	1000
/16	2	0.48828	2000
/32	3	0.97656	4000
/64	4	1.95312	8000
/128	5	3.90957	16000
/256	6	7.81248	32000

注：这些时间是按照 32.768 kHz 时钟给出。实际上，微控制器内部的 RC 频率会存在变化，具体变化范围见数据手册。此外，即使 RC 振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与 RC 振荡器时钟之间的相位差，因此总会有一个完整的 RC 周期是不确定的。通过对 LSI 进行校准可获得相对精确的看门狗超时时间。

24.3.2 硬件看门狗

如果用户在选择字节中启用了“硬件看门狗”功能，在系统上电复位后，看门狗会自动开始运行；如果在计数器计数结束前，若软件没有向键寄存器写入相应的值，则系统会产生复位

24.3.3 寄存器保护

对 IWDG_PR 和 IWDG_RLR 寄存器的写访问是被保护的，要修改这两个寄存器的值，IWDG_KR 必须先被写入 0x5555。对这些寄存器的写其他数将重新开启写保护，比如写入 0xAAAA，IWDG_PR 和 IWDG_RLR 将被再次保护。

IWDG_SR 指示 IWDG_PR 或 IWDG_RLR 的值是否正在更新。

24.3.4 调试模式

如果 CPU 进入调试模式，根据调试模块中的 DBG 模块中 DBG_IWDG_STOP 配置位的状态，IWDG 的计数器继续工作或停止

24.3.5 低功耗冻结

取决于选项字节里的 IWDG_STOP 和 IWDG_STBY 位，决定了 IWDG 停止模式或待机模式下是否继续保持计数。如果 IWDG 在停止或待机模式下保持工作，IWDG 能从当前的低功耗模式下唤醒微控制器。

24.4 IWDG 寄存器

只能字的方式操作以下寄存器。

24.4.1 IWDG 键值寄存器（IWDG_KR）

偏移地址：0x00

复位值：0x0000 0000（由待机模式复位）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	KEY[15:0]	W	16'h0	键值。 软件必须以一定的时间间隔向该寄存器写入 0xAAAA，否则，当计数器计数到 0 时，看门狗会产生复位。 写入 0x5555：允许访问 IWDG_PR 和 IWDG_RLR； 写入 0xCCCC：激活 IWDG（如果选择了硬件看门狗则不受此命令控制）。

24.4.2 IWDG 预分频寄存器（IWDG_PR）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
													RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:3	Reserved	-	-	保留
2:0	PR[2:0]	RW	3'h0	预分频值。 通过配置该寄存器选择计数器时钟的预分频值。 要改变该寄存器，IWDG_SR 寄存器的 PVU 必须为 0。 此寄存器受写保护。向 IWDG_KR 写入 0x5555 可解除写保护。 000：4 分频； 001：8 分频； 010：16 分频； 011：32 分频； 100：64 分频； 101：128 分频； 110：256 分频； 111：256 分频；

24.4.3 IWDG 重装载寄存器（IWDG_RLR）

偏移地址：0x08

复位值：0x0000 0FFF（由待机模式复位）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RL[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:12	Reserved	-	-	保留
11:0	RL[11:0]	RW	12'hFFF	IWDG 计数器重装载值。

Puya Confidential

				当向 IWDG_KR 寄存器写入 0xAAAA 时，RL 值会传送到计数器中。随后计数器从这个值开始递减计数。看门狗超时周期可通过此 RL 值和时钟预分频值来计算。 此寄存器受写保护。向 IWDG_KR 写入 0x5555 可解除写保护。只有当 IWDG_SR.RVU=0 时，才能对寄存器进行修改。

24.4.4 IWDG 状态寄存器（IWDG_SR）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RVU	PVU
														R	R

Bit	Name	RW	Reset Value	Function
31:2	Reserved	-	-	保留
1	RVU	R	0	看门狗计数器重装值更新。 此位由硬件置 1，表明重装载值正在更新。当重装载值更新结束后，此位由硬件清零。
0	PVU	R	0	看门狗预分频值更新。 此位由硬件置 1，表明预分频值正在更新。当预分频值更新结束后，此位由硬件清零。

注：在更新 IWDG_PR 和 IWDG_SR.RLR 前，要分别等待 IWDG_PVU 和 IWDG_SR.RVU 为 0.但在更新 IWDG_PR、IWDG_RLR 后，不必再等待 IWDG_SR.PVU、IWDG_SR.RVU 为 0，可继续执行后续的代码。

Puya Confidential

25. 窗口看门狗（WWDG）

25.1 WWDG 简介

窗口看门狗通常用来监测由外部干扰或不可预见的逻辑条件造成的应用程序跑飞而产生的软件故障。除非递减计数器的值在 T[6]位变成 0 前被刷新，看门狗电路在达到预置的时间周期时，会产生一个复位。在递减计数器达到窗口寄存器数值之前，如果 7 位的递减计数器数值（在控制寄存器中）被刷新，那么也将产生一个复位。这表明递减计数器需要在一个有限的时间窗口中被刷新。

25.2 WWDG 主要特性

- 可编程的自由运行递减计数器
- 若看门狗被启动，满足以下任一条件时，产生系统复位:
  - 递减计数器的值小于 0x40
  - 递减计数器在窗口外被重新装载
- 如果启动了看门狗并且开启早期唤醒中断(EWI)，则当递减计数器等于 0x40 时产生 EWI，软件可在 EWI 中断程序内重装载计数器以避免 WWDG 复位。
- 具有硬件看门狗功能：在用户选项字节里将 WWDG 配置成硬件启动模式后，WWDG 会在复位后自动使能

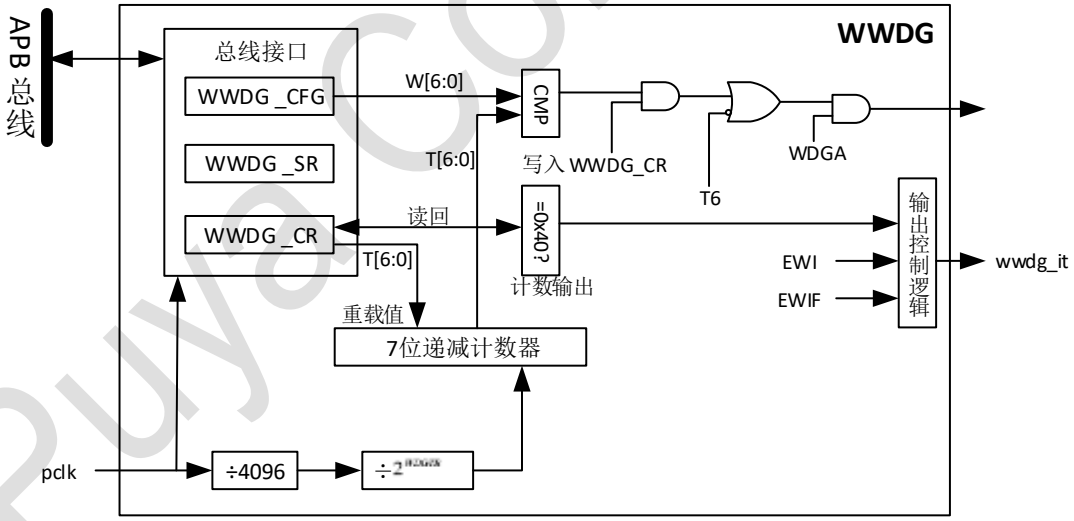


图 25-1 WWDG 模块框图

25.3 WWDG 功能描述

如果看门狗被启动（WWDG_CR 寄存器中的 WDGA 位被置”1”），并且当 7 位（T[6:0]）递减计数器从 0x40 翻转到 0x3F（T[6]位清零）时，产生一个复位。如果软件在计数器值大于窗口寄存器中的数值时重新装载计数器，将产生一个复位。

Puya Confidential

软件在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止微控制器发生复位。只有当计数器值小于窗口寄存器的值时，才能进行写操作。储存在 WWDG_CR 寄存器中的数值必须在 0xFF 和 0xC0 之间：

- 启动看门狗

当选项字节里的 WWDG_SW 选择了软件启动 WWDG 时（WWDG_SW=1），系统复位之后 WWDG 处于关闭状态，将 WWDG_CR 里的 WDGA 为设为 1 即可使能 WWDG。WWDG 一旦被软件启动就无法被关闭，除非发生了系统复位。

当选项字节里的 WWDG_SW 选择了硬件启动 WWDG 时（WWDG_SW=0），系统复位之后 WWDG 时钟自动启动，且 WWDG 处于开启状态，且无法被关闭。

- 控制递减计数器

递减计数器处于自由运行状态，即使看门狗被禁止，递减计数器仍继续递减计数。当看门狗被启用时，T[6]位必须被置 1，以防止立即产生一个复位。

T[5:0]位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频值是未知的。

配置寄存器（WWDG_CFR）中包含窗口的上限值：要避免产生复位，递减计数器必须在其值小于窗口寄存器的数值并且大于 0x3F 时被重新装载。

另一个重装载计数器的方法是利用早期唤醒中断（EWI）。设置 WWDG_CFR 寄存器中的 EWI 位开启该中断。当递减计数器到达 0x40 时产生此中断，相应的中断服务程序（ISR）可以用来加载计数器以防止 WWDG 复位。在 WWDG_SR 寄存器中写’0’可以清除该中断。

注： 可以用 T[6]位产生一个软件复位（设置 WDGA 位为”1”，T[6]位为”0”）。

25.4 WWDG 超时程序

可以使用下图中提供的公式计算窗口看门狗的超时时间。

警告：当写入 WWDG_CR 寄存器时，始终置 T[6]位为’1’以避免立即产生一个复位。

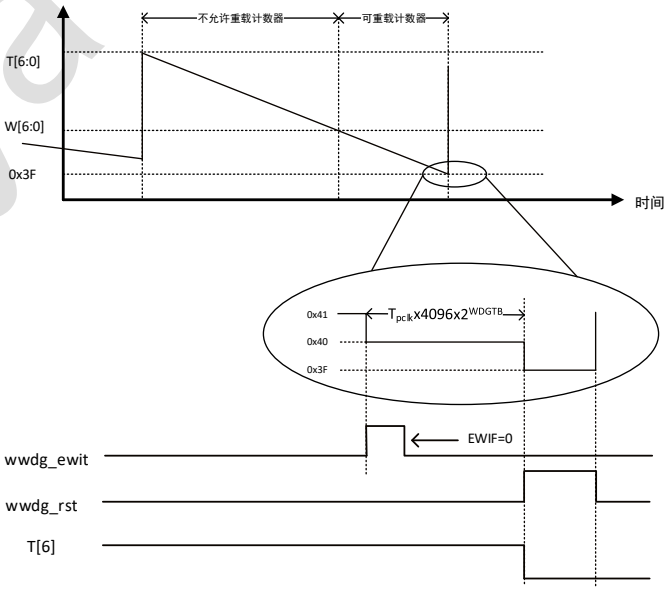


图 25-2 窗口看门狗时序图

Puya Confidential

计算 WWDG timeout 值的公式如下：

$$t_{WWDG} = t_{PCLK} \times 4096 \times 2^{WWDGTB[1:0]} \times (T[5:0] + 1)\ (ms)$$

25.5 WWDG 调试模式

当微控制器进入调试模式时（Cortex-M4F 核心停止），根据调试模块中的 DBG_WWDG_STOP 配置位的状态，决定 WWDG 的计数器继续工作或停止。详见调试模式章节。

25.6 WWDG 寄存器

25.6.1 WWDG 控制寄存器（WWDG_CR）

偏移地址：0x00

复位值：0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
-								RS	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7	WDGA	RS	0	激活位（Activation bit） 此位由软件置”1”，但仅能由硬件在复位后清”0”。当 WDGA=1 时，看门狗可以产生复位。 0：禁止看门狗 1：启用看门狗
6:0	T[6:0]	RW	7’h7F	7 位计数器（MSB 至 LSB） 这些位用来存储看门狗的计数器值。每(4096x2^WDGTB)个 PCLK1 周期减 1。当计数器值从 0x40 变为 0x3F 时(T[6]变成 0)，产生看门狗复位。

25.6.2 WWDG 配置寄存器（WWDG_CFR）

偏移地址：0x04

复位值：0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EWI	WDGTB[1:0]		W[6:0]						
-						RS	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:10	Reserved	-	-	保留
9	EWI	RS	0	提前唤醒中断 此位若置’1’，则当计数器值达到 0x40，即产生中断。 此中断只能由硬件在复位后清除。
8:7	WDGTB[1:0]	RW	2’h0	时基 预分频器的时基可以设置如下：

Puya Confidential

				00: CK 计时器时钟（PCLK1 除以 4096）除以 1 01: CK 计时器时钟（PCLK1 除以 4096）除以 2 10: CK 计时器时钟（PCLK1 除以 4096）除以 4 11: CK 计时器时钟（PCLK1 除以 4096）除以 8
6:0	W[7:0]	RW	7'h7F	7 位窗口值 这些位包含了用来与递减计数器进行比较用的窗口值。

25.6.3 WWDG 状态寄存器（WWDG_SR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
-															RC_W0

Bit	Name	RW	Reset Value	Function
31:1	Reserved	-	-	保留
0	EWIF	RC_W0	0	提前唤醒中断标志 当计数器值达到 0x40 时，此位由硬件置"1"。它必须通过软件写"0"来清除。对此位写"1"无效。若中断未被使能，此位也会被置"1"。

Puya Confidential

26. 串行外设接口（SPI）

26.1 SPI 简介

串行外设接口（SPI）允许芯片与外部设备以半/全双工、同步、串行方式通信。此接口可以被配置成主模式，并为外部从设备提供通信时钟（SCK）。接口还能以多主配置方式工作。它可用于多种用途，包括使用一条双向数据线的双线单工同步传输，还可使用经 CRC 校验的可靠通信。

26.2 SPI 主要特征

- 支持主机或者从机模式
- 3 线全双工同步传输
- 2 线半双工同步传输（有双向数据线）
- 2 线单工同步传输（无双向数据线）
- 8 位或者 16 位传输帧选择
- 支持多主模式
- 8 个主模式波特率预分频系数
- 从模式频率
- 主模式和从模式下均可以由软件或硬件进行 NSS 管理：主/从操作模式的动态改变
- 可编程的时钟极性和相位
- 可编程的数据顺序，MSB 在前或 LSB 在前
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- 支持可靠通信的硬件 CRC
  - 在发送模式下，CRC 值可以被作为最后一个字节发送
  - 在全双工模式中对接收到的最后一个字节自动进行 CRC 校验
  - CRC 错误标志
- 支持 Motorola 和 TI 模式
- 可触发引起中断的主模式故障、过载标志
- 2 个具备 DMA 能力的深度为 4，宽度为 16bit（当数据帧设置为 8bit 时，宽度为 8bit）的嵌入式 Rx 和 Tx FIFOs

Puya Confidential

26.3 SPI 功能描述

26.3.1 SPI 框图

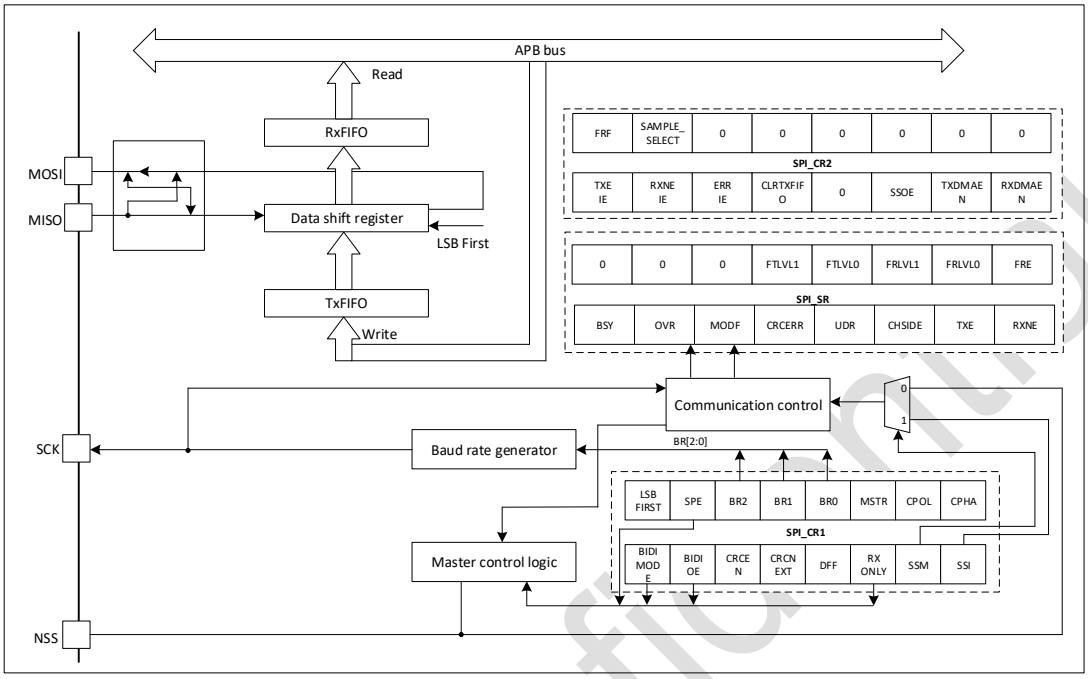


图 26-1 SPI 框图

SPI 通过 4 个引脚与外部器件相连：

MISO：主机输入/从机输出引脚。该引脚在从模式下发送数据，在主模式下接收数据。

MOSI：主机输出/从机输入引脚。该引脚在主模式下发送数据，在从模式下接收数据。

SCK：串口时钟，作为主机的输出，从机的输入。

NSS：从机选择。取决于 SPI 和 NSS 的设定，该引脚可以用作：

- 选择要通讯的从机
- 同步数据帧或者
- 检测多主机间的冲突

SPI 总线允许在一个主机和一个或者多个从机之间的通讯。总线由至少两根线组成：一个是时钟，另一个是用于同步传输的数据。根据应用场景，可以选择增加另外一根数据线和从机选择信号。

26.3.2 一个主机和一个从机的通信

针对不同的应用场景，SPI 可以使用几种不同的配置进行通讯。这些配置使用 2 线、3 线 (软件 NSS)，或者 3 线、4 线（硬件 NSS）。通讯始终由主机启动。

26.3.2.1 全双工通信

SPI 被配置成全双工通讯的情况下，主机和从机的移位寄存器，在 MOSI 和 MISO 之间，使用两个单向的线连到一起。在 SPI 通讯期间，数据在主机提供的时钟沿同步移位。主机通过 MOSI 发送数据，从 MISO 接收来自从机的数据。当数据帧传输完成，结束在主机和从机之间的信息交互。

Puya Confidential

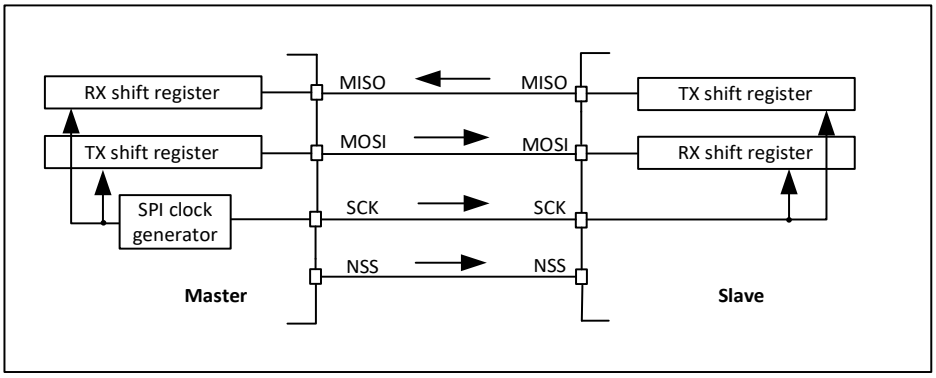


图 26-2 全双工单主机/单从机应用

26.3.2.2 半双工通信

通过设定 BIDIMODE（SPI_CR1 寄存器），SPI 可以工作在半双工模式。在这种配置下，用 1 根数据线完成主机和从机移位寄存器的连接。在通讯过程中，在 SCK 的时钟沿，数据在两个移位寄存器之间以 BIDIOE（SPI_CR1 寄存器）选择的方向同步移位。在该配置下，主机的 MISO 和从机的 MOSI 可被释放作为通用端口给其他应用使用。

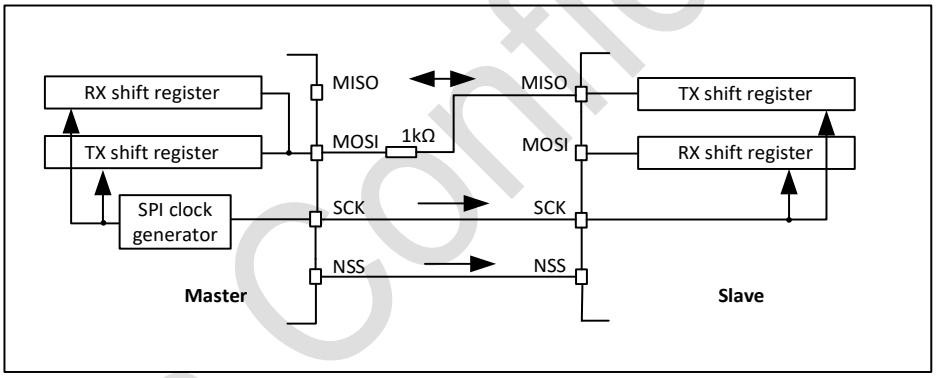


图 26-3 半双工单主机/单从机应用

（1）NSS 引脚可以被使用在主机和从机之间进行硬件控制流。可选的，NSS 也可以不使用。然后该流程就要内部处理。

（2）在该配置下，主机的 MISO 引脚和从机的 MOSI 引脚可以用作 GPIO。

（3）当以双向模式工作的两个节点间的通信方向不是同步变化时，会出现临界情况，新发送器访问共用数据线，而前一个发送器仍保持线路上的相反值（值取决于 SPI 配置和通信数据）。两个节点会出现冲突，在共用线上短暂提供相反的输出电平，直到下一个节点也相应地改变其方向设置。建议此模式下在 MISO 和 MOSI 引脚之间插入串行电阻以在这种情况下保护输出并限制电流在二者之间流过。

26.3.2.3 单工通信

通过使用 RXONLY（SPI_CR1 寄存器）位将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这个配置下，在主机和从机的移位寄存器之间只使用 1 根线。另一对 MISO 和 MOSI 引脚不被使用，可以被释放成 GPIO。

Puya Confidential

- 只发送模式（RXONLY=0）：配置与全双工模式相同。应用忽略在未使用的输入引脚上的信息。这个引脚可以被用作标准的 GPIO。
- 只接收模式（RXONLY=1）：通过置位 RXONLY，应用可以禁止 SPI 输出功能。
  - 在从机模式配置下，MISO 输出被禁止，该引脚被用作 GPIO。在从机选择信号有效时，从机继续从 MOSI 引脚接收数据。接收到的数据事件是否发生取决于数据缓冲区的配置。
  - 在主机模式配置下，MOSI 输出被禁止，引脚可以用作 GPIO。只要 SPI 处于使能状态，便不断生成时钟信号。停止时钟的唯一方式是将 RXONLY 位或 SPE 位清零，直至来自 MISO 引脚的传入模式结束，然后基于相应配置填充数据缓冲区结构。

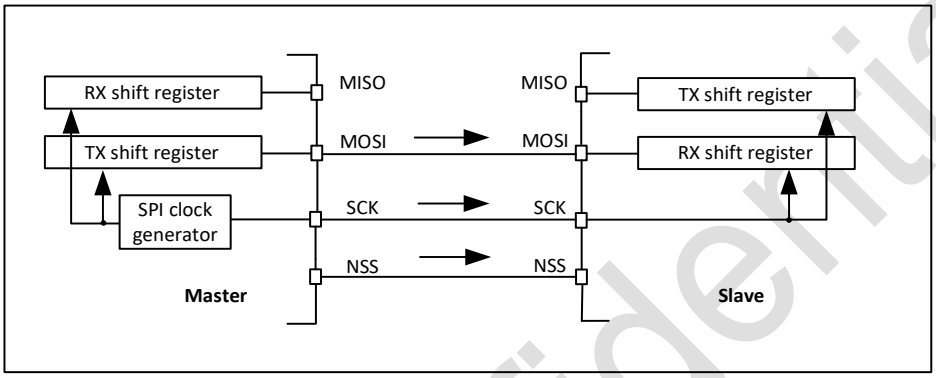


图 26-4 单工单从机/单主机应用

（1）在主机和从机之间可以使用 NSS 引脚进行硬件控制流。可选的，NSS 也可以不使用。然后该流程就要内部处理。

（2）在 Rx 移位寄存器的输入上捕获意外的输入信息。在标准的只发送模式下，所有与传输接收相关的事件都被必须被忽略（如 OVF 标志）。

（3）在该配置下，两边的 MISO 引脚都被用作 GPIO。

注：任何单工通信都可以通过固定半双工模式中的方向设置来实现（使能双向模式，同时 BDIO 位保持不变）。

26.3.3 标准多从机通信

在一个有两个或者更多独立从机的应用场景里，主机使用 GPIO 控制 NSS 来管理从机。主机必须通过拉低连接的从机 NSS，选择某个从机。当完成这个操作，就建立了主机和选定的从机通讯。

Puya Confidential

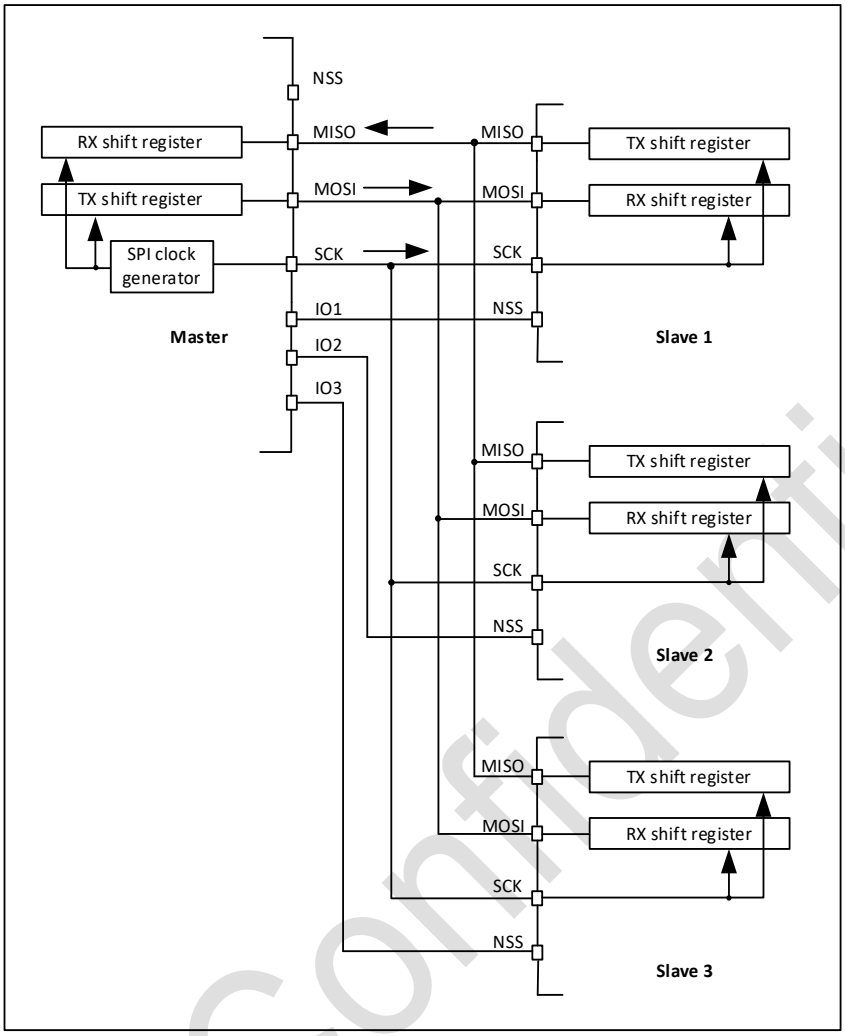


图 26-5 主机与三个独立的从机通信

在这种配置下不使用 NSS 引脚。必须通过 SSM=1，SSI=1 来防止任何 MODF 错误。

由于从机的 MISO 引脚连接到一起，所有从机 MISO 引脚的 GPIO 必须配置为开漏复用功能。

26.3.4 多主机通信

如果 SPI 总线未用于多主机功能，用户可使用内置功能来检测试图同时控制总线的两个节点间是否存在潜在冲突。对于该检测，NSS 引脚配置为硬件输入模式。

由于此时只有一个结点可将其输出施加到公用数据线上，因此该模式连接的 SPI 节点不能超过两个。

当节点无效时，默认情况下均保持从机模式。一旦一个节点要接管对总线的控制，它会将自身切换到主机模式，然后通过专用 GPIO 引脚向其他节点的从机选择输入施加有效电平。会话完成后，有效的从机选择信号将被释放，控制总线的节点会短暂切换回被动从模式，等待下一个会话开始。

如果两个节点同时发出各自的控制请求，则会出现总线冲突（模式故障 MODF 事件）。随后，用户可应用某个简单的仲裁过程（例如，在两个节点上施加不同的预定义超时来推迟下一次尝试）。

Puya Confidential

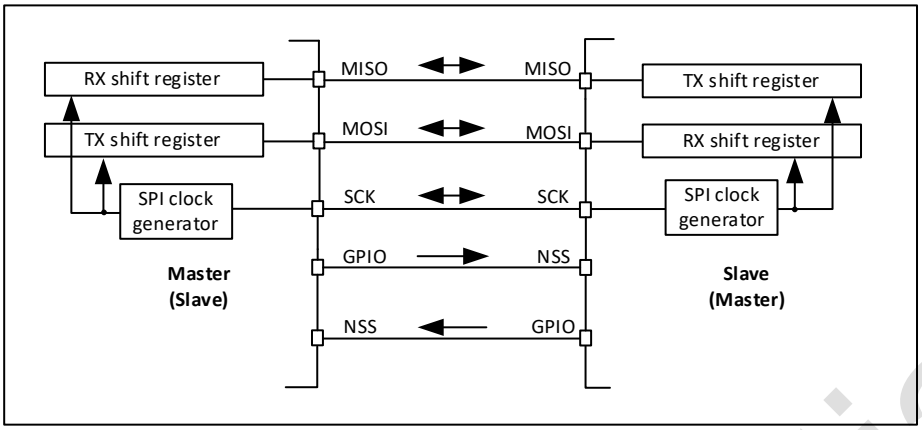


图 26-6 多主机应用

NSS 引脚在两个节点都被配置成硬件输入模式。当被动节点被配置成从机时，其有效电平将使能 MISO 输出控制。

26.3.5 从机选择（NSS）引脚管理

在从机模式，NSS 作为标准的片选输入，使从机能与主机通讯。在主机模式，NSS 既可以作为输出又可以作为输入。当作为输入时，它可以防止多主机的总线冲突，当作为输出时，它可以驱动单个从机的从机选择信号。

可以使用 SPI_CR1 寄存器的 SSM 位设置硬件或者软件从机选择管理：

- 软件 NSS 管理（SSM=1）：在这个配置下，由 SPI_CR1 寄存器中的 SSI 位的值内部驱动从机选择信息。外部 NSS 引脚空闲，可供其他应用使用。
- 硬件 NSS 管理（SSM=0）：在这个情况下，有几个可能的配置：
  - NSS 输出使能（SSM=0，SSOE=1）：这个配置仅在作为主机时使用。硬件管理 NSS 引脚。只要在主机模式下使能 SPI（SPE=1），NSS 信号便会被驱动为低电平，并且会一直保持低电平状态，直至关闭 SPI（SPE =0）。在多主机应用中，SPI 不能进行这种 NSS 配置。
  - NSS 输出禁止（SSM=0，SSOE=0）：如果 MCU 在总线上作为主机，此配置可实现多主模式功能。如果在该模式下将 NSS 引脚拉至低电平，SPI 将进入主模式故障状态，器件将在从模式下自动进行重新配置。在从机模式，NSS 引脚作为标准的片选输入，当 NSS 为低时选择从机。

Puya Confidential

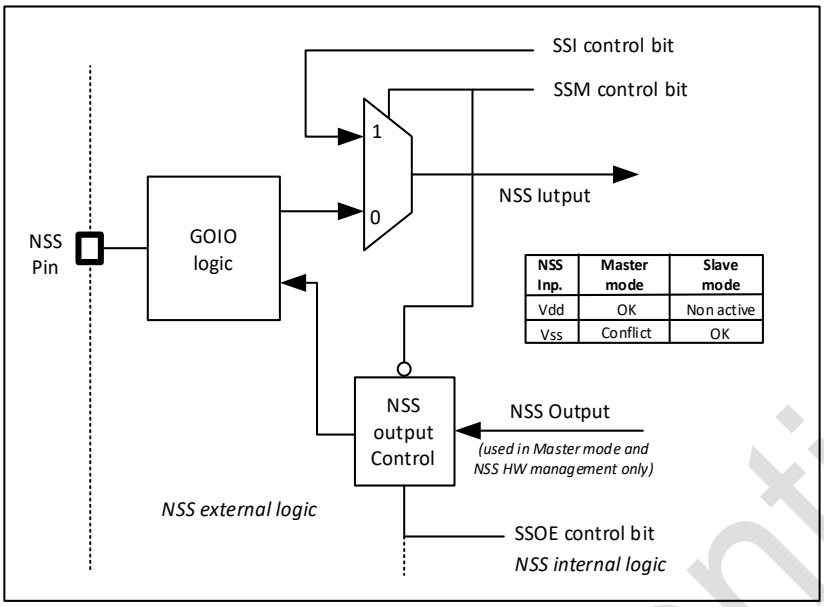


图 26-7 硬件/软件管理从机

26.3.6 通讯格式

在 SPI 通讯期间，接收和发送操作同时进行。SCK 将数据线上的数据移位和采样操作同步。通讯格式取决于时钟相位、时钟极性和数据帧格式。为了能够进行通讯，主机和从机必须遵循相同的通讯格式。

26.3.6.1 时钟极性和相位

通过 SPI_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。

CPOL（时钟极性）位控制不传任何数据时的时钟电平状态。此位对主器件和从器件都有作用。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA（时钟相位）位置 1，则 SCK 引脚上的第二个边沿（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）对 MSBit 采样。即，在第二个时钟边沿锁存数据。如果复位 CPHA 位，则 SCK 引脚上的第一个边沿（如果将 CPOL 位置 1，则为下降沿；如果复位 CPOL 位，则为上升沿）对 MSBit 采样。即，在第一个时钟边沿锁存数据。

CPOL 和 CPHA 的组合选择了数据采样时钟边沿。

在 CPOL/CPHA 改变之前，SPI 必须被关闭（SPE=0）。

SCK 的 IDLE 状态必须对应 SPI_CR1 寄存器选择的极性。

Puya Confidential

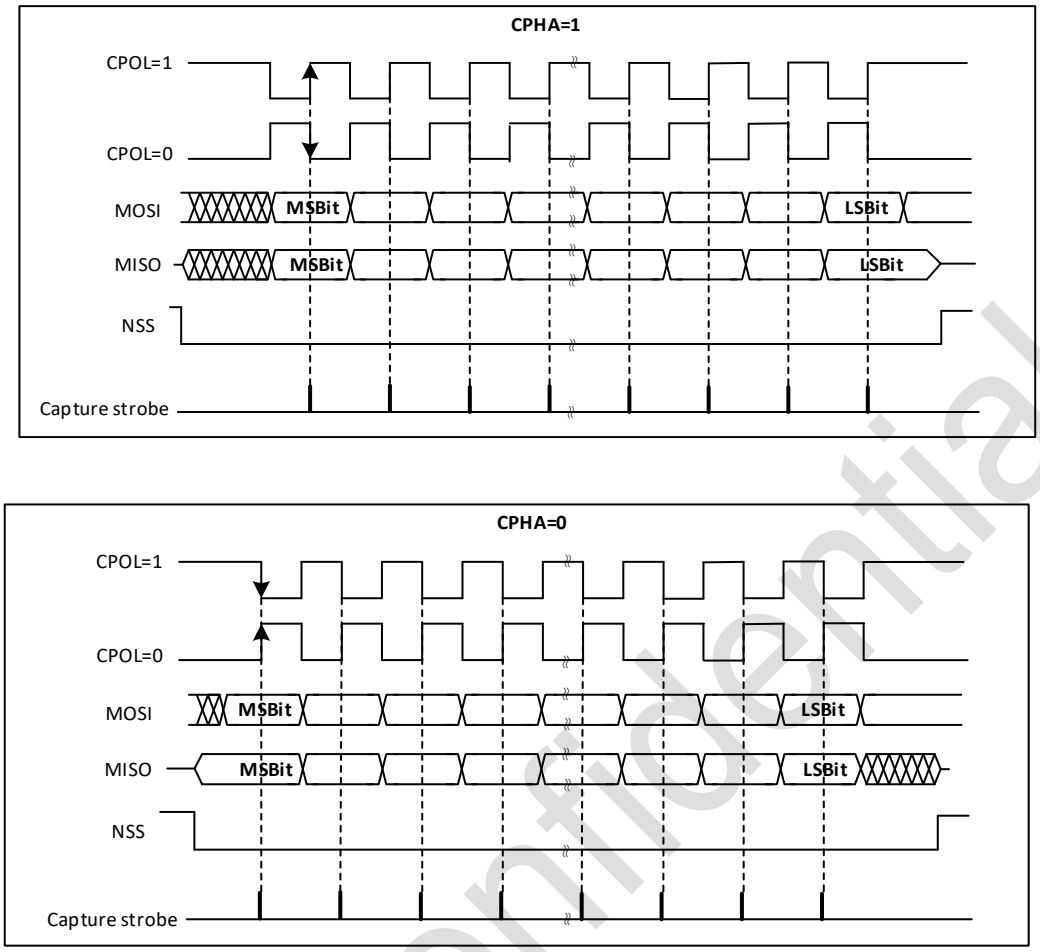


图 26-8 数据时钟时序图

数据位的顺序取决于 LSBFIRST（SPI_CR1 寄存器）的设定。

注：在作为 SPI 主机高速时，外部迟延大时。可以使能 SAMPLE_SELECT 位（SPI_CR2 寄存器），来选择采样时间点。默认主机接收数据为 1/2 个 cycle 采样，使能后将转变为 1 个 cylce 采样。

26.3.6.2 数据帧格式

通过 LSBFIRST（SPI_CR1 寄存器)，SPI 移位寄存器可以设定为 MSB-FIRST 或者 LSB-FIRST。通过使用 DFF（SPI_CR1 寄存器)，选择数据帧的位数，可选择为 8 位或者 16 位长度，该设置对于发送和接收都适用。

26.3.7 SPI 配置

对于主机和从机，SPI 的配置流程几乎一样。对于具体的模式建立，遵循专门的章节介绍。当进行标准的通讯，进行以下步骤：

1. 写相关的 GPIO 寄存器：配置 MOSI、MISO 和 SCK 引脚
2. 写 SPI_CR1 寄存器
   1) 通过 BR[2:0]配置时钟波特率（从机模式不需要）
   2) 配置 CPOL 和 CPHA，定义数据传输和串行时钟之间的关系
   3) 通过 RXONLY 或者 BIDIMODE 和 BIDIOE（RXONLY 和 BIDIMODE 不能同时有效），选择单

Puya Confidential

工或者半双工模式

4) 配置 LSBFIRST 定义帧格式
5) 如果需要 CRC，则配置 CRCL 和 CRCEN 位（SCK 时钟信号处于空闲状态时）
6) 配置 SSM 和 SSI
7) 配置 MSTR 位（在多主机 NSS 配置中，如果主机被配置防止 MODF 错误，要避免 NSS 的冲突状态）
8) 配置 DFF 位，选择数据帧位数

3. 写 SPI_CR2 寄存器
   1) 配置 SSOE（从机模式不需要）
4. 写 SPI_CRCPR 寄存器：需要时配置 CRC 多项式
5. 写相应的 DMA 寄存器（包括 SYSCFG_CFGR3 和 SYSCFG_CFGR4 配置 DMA 通道）：配置 DMA 的 SPI TX 和 Rx 通道

26.3.8 使能 SPI 流程

建议在主机发送时钟前使能 SPI 从机。否则，数据传输可能会不正常。从机的数据寄存器必须包含待发送的数据才能开始与主机通信（在通信时钟的第一个边沿；如果时钟信号连续，则是在正在进行的通信结束前）。使能 SPI 从机前，SCK 信号必须稳定为所选极性对应的空闲状态电平。

当 SPI 被使能并且 TXFIFO 不空，或者向 TXFIFO 进行下一个写操作，全双工模式（或者只发送）的主机开始通讯。

在任何主机只接收模式（RXONLY=1，或者 BIDIMODE=1 且 BIDIOE=0），SPI 使能后，主机开始通讯且主机立即开始提供时钟。

对于 DMA 处理，遵从具体的章节内容。

数据传输和接收流程

26.3.9 数据发送和接收

26.3.9.1 RXFIFO 和 TXFIFO

SPI 所有数据通讯都通过深度为 4，宽度为 16bit（当数据帧设置为 8bit 时，宽度为 8bit）的 FIFO。该特性使 SPI 能够以连续数据流进行工作，并防止由于 CPU 来不及处理数据导致的通讯问题。发送和接收有独立的 FIFO，叫做 TXFIFO 和 RXFIFO。这些 FIFO 被用在所有的 SPI 模式。

FIFO 的处理取决于多种参数，包括：数据交换模式（全双工、半双工）、数据帧格式。

读 SPI_DR 寄存器会得到最早存放在 RXFIFO 中还未被读走的数据结果。写 SPI_DR 寄存器，会在 FIFO 发送队列的最后位置，存入被写的数据。读写访问必须与数据宽度对齐。FTLVL[1:0]和 FRLVL[1:0]位显示了两个 FIFO 当前的占用级别。

对 SPI_DR 寄存器的读访问必须通过 RXNE 事件管理。当 RXFIFO 数据非空，该事件被触发。当 RXNE 被清零，RXFIFO 就被认为是空的。

相似地，写要发送的数据帧，通过 TXE 事件管理。当 TXFIFO 的占用级别小于或者等于总容量的一半时，该事件就会被触发。否则，TXE 被清零，并且 TXFIFO 被认为是满的。

用这样的方式，RXFIFO 和 TXFIFO 都可以存 4 个数据帧。

TXE 和 RXNE 事件都都可以通过查询或者中断方式处理。

Puya Confidential

另一种管理数据交换的方式是使用 DMA。

如果在 RXFIFO 已满时收到下一个数据，将发生上溢事件（OVR 标志）。上溢事件可通过轮询或中断方式来处理。

BSY 位被置 1 表示当前正在处理数据帧。当时钟信号连续运行时，在主机中，BSY 标志在数据帧之间保持置 1 状态，但在从机中，BSY 标志在数据帧传输之间变为低电平并持续最短的一段时间（一个 SPI 时钟）。

某些应用场景下，当向 TXFIFO 中写入数据，可以通过设置 CLRTXFIFO 位，清空 TXFIFO 数据，从而往 TXFIFO 里写新的数据重新进行通信。

26.3.9.2 处理序列

一些数据帧可以通过初始序列传递来完成一条信息。当发送被使能，当主机的 TXFIFO 里有任何数据，序列开始并继续进行。时钟信号被主机连续的提供，直到 TXFIFO 变空，然后停止等待额外的数据。

在只接收模式，即半双工（BIDIMODE=1，BIDIOE=0）或者单工模式（BIDIMODE=0，RXONLY=1），在 SPI 被使能并且只接收模式被激活的时候，主机就立即开始接收。主机一直会提供时钟并连续地接收数据，直到主机停止了 SPI 或者关闭了只接收模式。

当主机以连续的模式（SCK 信号是连续的）进行通讯，主机必须要考虑从机处理数据流的能力。当有必要时，主机必须降低通讯速度，并提供更慢的时钟，或者发送分开的帧。要注意的是，对于主机或者从机来说，没有下溢错误信号，来自于从机的数据被主机处理（即使从机不能及时发送正确的数据）。

每个序列都必须被 NSS 引脚管理，同时在多从机系统中选择要进行通讯的其中一个从机。在一个单从机系统，没有必要使用 NSS 去控制从机，但通常也最好提供 NSS，使通讯的每个数据序列开头同步。NSS 可以由软件和硬件两种方式管理。

当 BSY 被置位，它显示了正在进行的数据帧交互。当专门的帧交互被完成时，RXNE 标志置位。最后一位被采样，并且整个数据帧被存在 RXFIFO 中。

26.3.9.3 关闭 SPI 的流程

当关闭 SPI 时，必须按照本段中介绍的关闭步骤进行操作。当外设时钟停止时，在系统进入低功耗模式前做到这一点是十分重要的。否则会损坏正在进行的交互。在某些模式下，禁止步骤是停止所进行的连续通信的唯一方式。

全双工或者只发送模式下，主机可以在停止提供要发送的数据时完成交互。在这种情况下，在最后的数据交互后，时钟被停止。在这些模式下，SPI 被禁止之前，用户必须使用标准的禁止流程。当 SPI 在主机发送时，如果此时一个帧交互正在进行，或者下一个数据帧存在 TXFIFO 中，此时关闭 SPI，则 SPI 的的状态是不可预测的。

当主机处在只接收模式，停止连续时钟的唯一方法是关闭外设（SPE=0）。这必须在最后一个数据帧传输内的特定时间段，即第一位采样与最后一位传输开始之间完成（以便接收全部数量的预期数据帧并防止在最后一个有效数据帧后读取任何其他的“空”数据）。在该模式下关闭 SPI 时必须遵循特定步骤。

关闭 SPI 后，已接收但未读取的数据始终存储在 RXFIFO 中，这些数据必须在下次使能 SPI 后进行处理，然后才能启动新序列。为防止存在未读取的数据，需确保关闭 SPI 时 RXFIFO 为空，可通过正确的

Puya Confidential

关闭步骤来关闭 SPI，也可以通过控制外设复位专用的特定寄存器以软件复位的方式来初始化所有 SPI 寄存器从而关闭 SPI（RCC_APBRSTx 寄存器）。

标准的禁止流程是基于 BSY 状态，并查看 FTLVL[1:0]，以确保传输彻底完成。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：

- 当 NSS 信号被软件管理，主机要向从机提供正确的 NSS 脉冲。或者
- 最后的数据帧或者 CRC 帧仍在传输过程中，来自 DMA 或者 FIFO 的交互数据流完成。

正确的禁止流程是（只接收模式除外）：

1. 等待 FTLVL[1:0]=00（没有数据要发送）
2. 等待 BSY=0（最后的数据被处理完成）
3. 关闭 SPI（SPE=0）
4. 读数据，直到 FRLVL[1:0]=00（读所有接收到的数据）

对于特定只接收模式，正确的禁止流程是：

1. 在最后一个数据帧传输过程中，通过禁止 SPI（SPE=0），打断接收流程
2. 等待 BSY=0（最后的数据帧已被处理）
3. 读数据，直到 FRLVL[1:0]=00（读所有接收到的数据）

26.3.9.4 DMA 通信

为了以最大速度工作并且方便避免上溢所需的数据寄存器读/写过程，SPI 提供了 DMA 功能，该功能采用了简单的请求/应答协议。

当 TXE 或者 RXNE 置位，会产生 DMA 请求。Tx 和 Rx 缓冲区有独立的请求。

- 发送时，每次 TXE 置为 1，则产生 DMA 请求。然后 DMA 会向 SPI_DR 寄存器写入数据。
- 接收时，每次 RXNE 置为 1，则产生 DMA 请求。然后 DMA 读 SPI_DR 寄存器的数据。

当 SPI 被仅用作发送数据，可以只使能 SPI Tx DMA 通道。在这个情况下，因为被接收到的数据没有被读走，OVR 标志位置位。当 SPI 被仅用作接收数据，可以使能 SPI Rx DMA 通道。

在发送时，当 DMA 已经写入了所有要发送的数据（DMA_ISR 寄存器的 TCIF 标志位被置位），可以通过监测 BSY 标志来确保 SPI 通讯已完成。这是用来避免当禁止 SPI 或者进入停止模式时，最后的传输被破坏。软件必须先等待 FTLVL[1:0]=00，然后再等待 BSY=0。

通过 DMA 开始通讯时，为防止 DMA 通道管理引发错误事件发生，必须遵从以下步骤：

1. 使能 DMA Rx 缓冲区（SPI_CR2 的 RXDMAEN 位）（如果 Rx DMA 被使用）
2. 配置 SYSCFG_CFGR3.DMAx_MAP 寄存器，选择 SPI 使用的 DMA 通道；并配置 DMA 的寄存器（参考 DMA 模块的描述）
3. 使能 DMA Tx 缓冲区（在 SPI_CR2 寄存器的 TXDMAEN 位）（如果 Tx DMA 被使用）
4. 配置 SPE=1 使能 SPI

强制用以下步骤关闭通讯：

1. 如果使能了 DMA，通过操作 DMA 寄存器来关闭 DMA 模块（参考 DMA 模块的描述）
2. 通过 SPI 关闭流程关闭 SPI
3. 通过清除 TXDMAEN 和 RXDMAEN（SPI_CR2 寄存器），关闭 DMA Tx 和 Rx 缓冲区（如果 DMA Tx 和 Rx 被使用）

Puya Confidential

26.3.9.5 时序

本节介绍一些典型的时序，这些时序对于查询、中断或者 DMA 都是有效的。为了简化，假定 LSBFIRST=0, CPOL=0, CPHA=1。也不提供完整的 DMA 操作配置。

1. 激活 NSS 并使能 SPI 后，从机开始控制 MISO；当 NSS 被释放或者 SPI 关闭时从机失去对 MISO 的控制。必须为从机提供充足的时间，以便在传输开始前准备好主机专用的数据。
2. 在主机端，仅在 SPI 使能后，SPI 外设才会控制 MOSI 和 SCK 信号（也包括 NSS 信号）。如果 SPI 被关闭，SPI 外设就从 GPIO 断开，在这些线上的电平值取决于 GPIO 的设定。
3. 在主机端，如果通讯是连续的，则 BSY 在帧之间保持有效。在从机端，BSY 信号在数据帧之间通常会保持至少一个时钟周期的低电平状态。
4. 只有当 TXFIFO 是满的，TXE 信号才被清零。
5. 在 TXDMAEN 位一被置位，DMA 仲裁过程即开始。在 TXEIE 被置位后，产生 TXE 中断。当 TXE 信号有效时，开始向 TxFIFO 传输数据，直到 TxFIFO 变满，或者 DMA 传输完成。
6. 如果所有要被发送的数据装进了 TxFIFO，在 SPI 总线传输前 DMA Tx TCIF 标志就会被拉高。这个标志在 SPI 交互完成之前，一直都为高。
7. 在数据封装模式，TxE 和 RxNE 事件是成对出现的，每个读/写 FIFO 的访问是 16bit 宽，直至数据帧数为偶数。如果 TxFIFO 是 3/4 满，FTLVL 状态停在 FIFO 全满级别。这就是为什么最后一个奇数帧不能在 TxFIFO 变成 1/2 满之前存储。该数据帧以 8-bit 的访问方式（由 DMA 内部信号在 LDMA_TX 置 1 时自动实现）存储在 TxFIFO 中。
8. 为了接收数据封装模式的最后一个奇数数据帧，当最后一个数据帧被处理时，Rx 阈值必须被改变成 8-bit（由 DMA 内部信号在 LDMA_RX 置 1 时自动实现）。

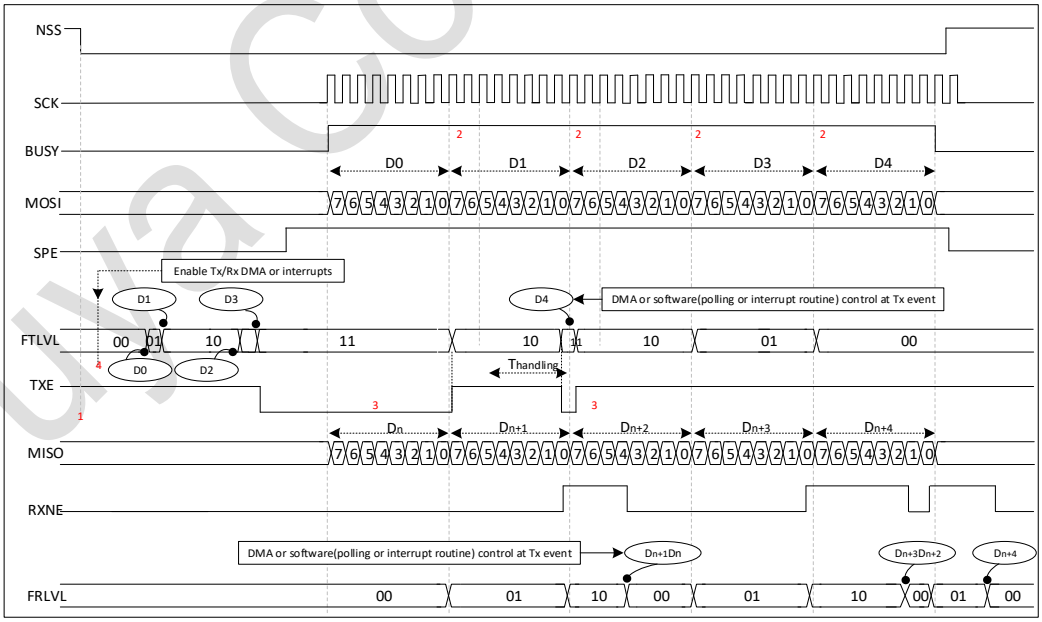


图 26-9 主模式全双工通讯(帧长=8)

Puya Confidential

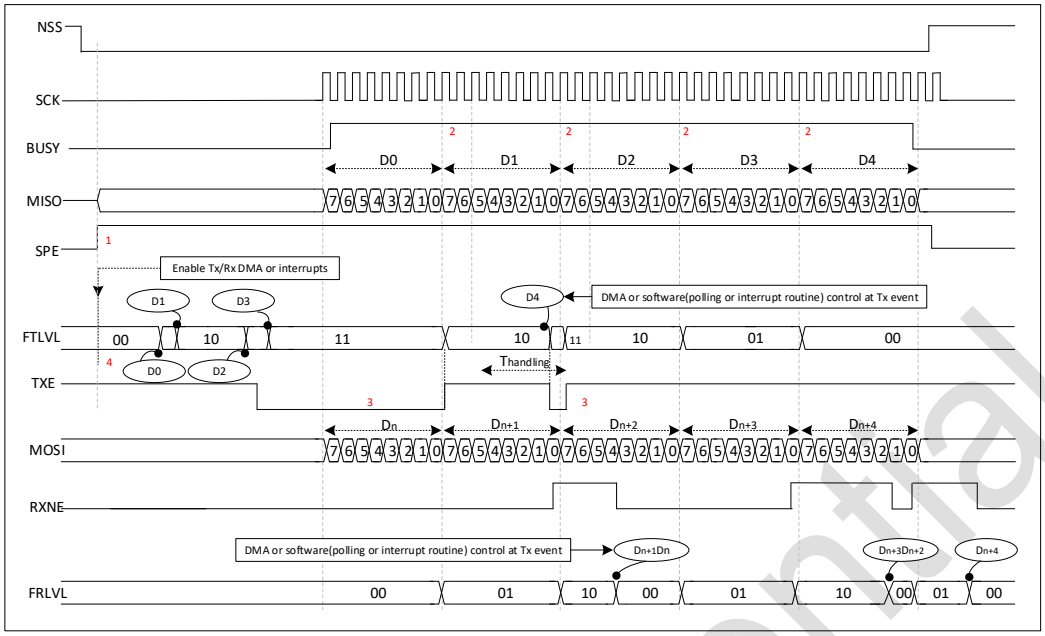


图 26-10 从模式全双工通讯(帧长=8)

26.3.10 状态标准

应用程序通过 3 个状态标志可以完全监控 SPI 总线的状态。

26.3.10.1 发送缓存空标志(TXE)

当 TXFIFO 有足够的空间存放要发送的数据时，TXE 标志位置位。TXE 标志位与 TXFIFO 占用级别有关。该标志位变高并保持高电平，直到 TXFIFO 占用级别小于等于 1/2 FIFO 深度。如果 SPI_CR2 寄存器的 TXEIE 置位，则会产生中断请求。当 TXFIFO 占用级别大于 1/2，该位被自动清零。

26.3.10.2 接收缓存非空(RXNE)

当接收到数据，RXNE 置位。

如果 SPI_CR2 寄存器的 RXNEIE 位置位，则产生中断。

当上述条件不再成立，则 RXNE 被硬件自动清零。

26.3.10.3 忙标志(BSY)

BSY 标志由硬件设置与清除 (写入此位无作用)，此标志表明 SPI 正在进行数据传输 (SPI 总线繁忙)。当它被设置为’1’时，表明 SPI 正忙于通信，但有一个例外：主模式的双向接收模式下（MSTR=1、IDIMODE=1 并且 BIDIOE=0），在接收期间 BSY 标志保持为低（接收 CRC 期间除外）。

在软件要关闭 SPI 模块并进入停止低功耗模式（或关闭设备时钟）之前，可以使用 BSY 标志检测传输是否结束，这样可以避免破坏最后一次传输。

BSY 标志还可以用于在多主机系统中避免写冲突。

除了主模式的双向接收模式（MSTR=1、IDIMODE=1 并且 BIDIOE=0），当传输开始时， BSY 标志被置’1’。

以下情况该标志将被清除为’0’：

Puya Confidential

- 正确关闭 SPI
- 主机模式，当检测到故障时（MODF=1）
- 主机模式，完成了数据发送并且不准备发送任何新数据时
- 从机模式，在每个数据传输之间，BSY 标志置为 0，并保持至少一个 SPI 时钟周期

注：不要使用 BSY 标志处理每个数据发送和接收。使用 TXE 和 RXNE 更合适。

26.3.11 错误标志

26.3.11.1 主模式故障（MODF）

主模式故障（MODF）仅发生在：当 NSS 作为输入信号（SSOE=0），NSS 引脚硬件管理模式下，主机的 NSS 脚被拉低；或者在 NSS 引脚软件管理模式下，SSI 位被置为'0'时，MODF 位被自动置位。主模式故障对 SPI 设备有以下影响：

- MODF 位置为'1'，如果设置了 ERRIE 位，则产生 SPI 中断
- SPE 位被清为'0'。这将停止一切输出，并且关闭 SPI 接口
- MSTR 位被清为'0'，因此强迫此设备进入从模式

下面的步骤用于清除 MODF 位：

1. 当 MODF 位被置为'1'时，执行一次对 SPI_SR 寄存器的读或写操作
2. 然后写 SPI_CR1 寄存器

在有多个 MCU 的系统中，为了避免出现多个从机的冲突，必须先拉高该主机的 NSS 引脚，再对 MODF 位进行清零。在完成清零之后， SPE 和 MSTR 位可以恢复到它们的原始状态。

出于安全的考虑，当 MODF 位为'1'时，硬件不允许设置 SPE 和 MSTR 位。

通常配置下，从机的 MODF 位不能被置为'1'。然而，在多主机配置里，一个设备可以在设置了 MODF 位的情况下，处于从机模式；此时，MODF 位表示可能出现了多主冲突。中断程序可以执行一个复位或返回到默认状态来从错误状态中恢复。

26.3.11.2 上溢（OVF）

当主机或者从机接收了数据，但 RXFIFO 没有足够的空间存储接收到的数据时，产生上溢情况。如果软件或者 DMA 没有足够的时间读走之前接收到的数据（RXFIFO 中存放），该情况就会发生。

当上溢情况发生，新收到的数据不会覆盖以前存放在 RXFIFO 的数据。接收到的新数据被忽略，并且丢弃所有接下来发送的数据。

依次读出 SPI_DR 寄存器和 SPI_SR 寄存器可将 OVR 清除。

26.3.11.3 CRC 错误（CRCERR）

当 SPI_CR1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。如果移位寄存器中接收的值与 SPI_RXCRC 的值不匹配，SPI_SR 寄存器中的 CRCERR 标志将置 1。该标志由软件清零。

26.3.11.4 TI 模式帧格式错误（FRE）

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在通信进行期间出现 NSS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPI_SR 寄存器中的 FRE 标志将置 1。发生错误时不会关闭 SPI，

Puya Confidential

但会忽略 NSS 脉冲，并且 SPI 会等待下一个 NSS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失两个数据字节，因此数据可能会损坏。

读取 SPI_SR 寄存器时，将清零 FRE 标志。如果 ERRIE 位置 1，则检测到 NSS 错误时将生成中断。在这种情况下，由于无法保证数据的一致性，应关闭 SPI，并在重新使能从 SPI 后，由主器件重新发起通信。

26.3.12 TI 模式

26.3.12.1 主模式下的 TI 协议

SPI 接口与 TI 协议兼容。可以使用 SPI_CR2 寄存器的 FRF 位来配置 SPI，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPI_CR1 中的设置无关。NSS 管理也特定于 TI 协议，在这种情况下，无法通过 SPI_CR1 和 SPI_CR2 寄存器（SSM、SSI 和 SSOE）来对 NSS 管理进行配置。

在从机模式下，SPI 波特率预分频器用于控制在当前传输完成时 MISO 引脚切换为高阻态的时刻。可以使用任意波特率，因此可以非常灵活地确定此时刻。但是，波特率通常设置为外部主时钟波特率。MISO 信号变为高阻态的延时（$t_{release}$）取决于内部重新同步以及通过 SPI_CR1 寄存器的 BR[2:0]位设置的波特率值。具体公式如下：

$$\frac{t_{baud_rate}}{2} + 4 \times t_{pclk} < t_{release} < \frac{t_{baud_rate}}{2} + 6 \times t_{pclk}$$

如果从机在数据帧传输期间检测到错位的 NSS 脉冲，FRE 标志将置 1。

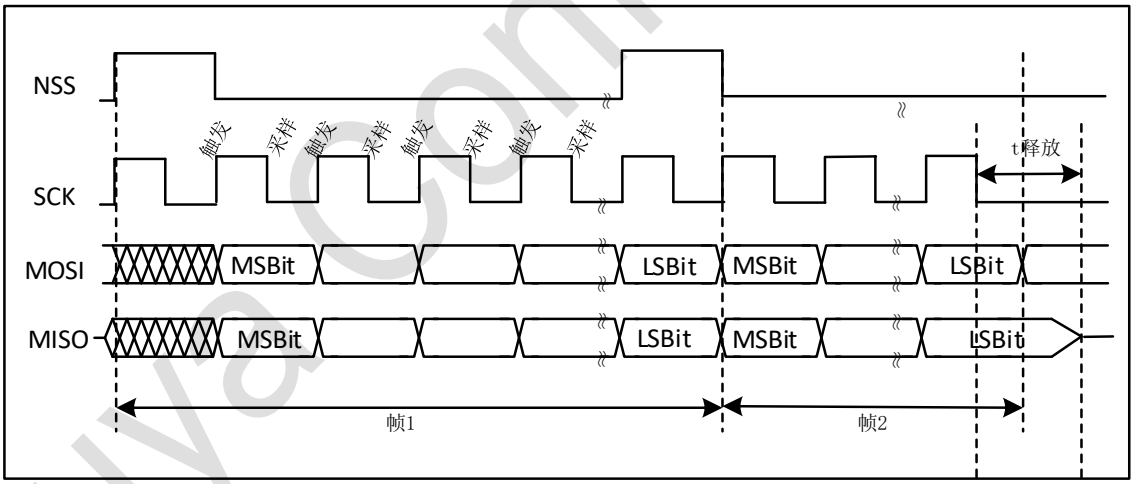


图 26-11 TI 模式传输

26.3.13 SPI CRC

为了检查发送和接收数据的可靠性，利用两个独立的 CRC 计算器。SPI 提供 CRC8 或 CRC16 计算，且数据帧长度为 8 时选择 CRC8，数据帧长度为 16 时选择 CRC16。

26.3.13.1 CRC 原理

在 SPI 启用（SPE = 1）之前，通过设置 SPI_CR1 寄存器中的 CRCEN 位来启用 CRC 计算。CRC 值是使用每个位上的奇数可编程多项式计算的。该计算在由 SPI_CR1 寄存器中的 CPHA 和 CPOL 位定义的采样时钟边沿上进行处理。计算出的 CRC 值在数据块结束时自动校验，并且由 CPU 或 DMA 管理

Puya Confidential

其传输。当检测到根据接收数据内部计算的 CRC 与发送器发送的 CRC 不匹配时，设置 CRCERR 标志以指示数据错误。处理 CRC 计算的正确程序取决于 SPI 配置和选择的传输管理方式。

26.3.13.2 CPU 管理 CRC 传输

通信开始后将一直持续到必须发送或接收 SPIx_DR 寄存器中的最后一个数据帧。之后，SPIx_CR1 寄存器中的 CRCNEXT 位必须置 1，以指示当前处理的数据帧传输后将处理 CRC 帧传输。CRCNEXT 位必须在最后一个数据帧传输结束前置 1。在 CRC 传输期间，CRC 计算将冻结。

所接收的 CRC 以数据字节或数据字的形式存储在 RXFIFO 中。因此仅在 CRC 模式下，接收缓冲区才必须被视为一个 16 位缓冲区且一次仅接收一个数据帧。

CRC 格式的传输通常在数据传输结束时需要多出一个数据帧。但是，当设置一个通过 16 位 CRC 校验的 8 位数据帧时，需要多两帧才能发送完整的 CRC 值。

接收最后一个 CRC 数据后，将执行自动校验，将接收的值与 SPI_RXCRC 寄存器中的值进行比较。软件必须校验 SPI_SR 寄存器中的 CRCERR 标志，以确定数据传输是否损坏。软件通过向 CRCERR 标志写入“0”来将其清零。

接收 CRC 后，CRC 值存储到 RXFIFO 中，且必须在 SPI_DR 寄存器中进行读取，以将 RXNE 标志清零。

26.3.13.3 DMA 管理 CRC 通信

当使能 SPI，并开启带 CRC 的 DMA 模式进行通信时，在通信结束时自动发送和接收 CRC（在仅接收模式下读取 CRC 数据除外），CRCNEXT 位并非一定要通过软件来处理。 SPI 发送 DMA 通道计数器必须设置为要发送的数据帧数，其中不包括 CRC 帧。在接收器侧，接收的 CRC 值在传输结束时通过 DMA 自动处理,但用户必须注意刷新 RXFIFO 中接收的 CRC 信息(因为该信息始终加载到其中)。在全双工模式下，接收 DMA 通道计数器可设置为要接收的数据帧数，其中包括 CRC。

在只接收模式下，DMA 接收通道计数器应仅包含已传输的数据量，而不包括 CRC 计算。之后，基于通过 DMA 实现的完整传输，必须通过软件从 FIFO 中读回所有 CRC 值，因为 FIFO 在该模式下用作单个缓冲区。

如果传输过程中出现故障，则在数据和 CRC 传输结束时，将 SPI_SR 寄存器中的 CRCERR 标志位置 1。

26.3.13.4 CRC 使用流程

使能 CRC 的流程如下：

- 配置 CPOL、CPHA、LSBFIRST、BR、SSM、SSI 和 MSTR 寄存器
- 在 SPI_CRCPR 寄存器输入多项式
- 通过设置 SPI_CR1 寄存器 CRCEN 位使能 CRC 计算，该操作也会清除寄存器 SPI_RXCRCR 和 SPI_TXCRC
- 设置 SPI_CR1 寄存器的 SPE 位启动 SPI 功能
- 启动通信并且维持通信，直到只剩最后一个字节或者半字

Puya Confidential

- 在把最后一个字节或半字写进发送缓冲器时，设置 SPI_CR1 的 CRCNEXT 位，指示硬件在发送完成最后一个数据之后，发送 CRC 的数值。在发送 CRC 数值期间，停止 CRC 计算
- 当最后一个字节或半字被发送后，SPI 发送 CRC 数值，CRCNEXT 位被清除。同样，接收到的 CRC 与 SPI_RXCRCR 值进行比较，如果比较不匹配，则设置 SPI_SR 寄存器的 CRCERR 标志位，当设置了 SPI_CR2 寄存器的 ERRIE 时，则产生中断。

注意：在使能 CRC 后，接收或者发送数据会一直做 CRC 计算。所以，如果数据不再需要 CRC 计算，要及时关闭 CRC。

如果在通信期间关闭了 SPI，再使能 SPI 及 CRC，则必须遵循以下顺序：

关闭 SPI

- 将 CRCEN 清零
- 使能 CECEN
- 使能 SPI

26.4 SPI 中断

表 26-1 SPI 中断请求

中断事件	事件标志	使能控制位
TXFIFO 等待被装载	TXE	TXEIE
数据接收到 RXFIFO 中	RXNE	RXNEIE
主模式故障事件	MODF	ERRIE
上溢错误	OVR	
TI 帧格式错误	FRE	
CRC 校验错误	CRCERR	

26.5 SPI 寄存器

26.5.1 SPI 控制寄存器 1（SPI_CR1）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	BIDIMODE	RW	0	双向数据模式使能。 0：“双线单向”模式 1：“单线双向”模式 注：该位不适用于 I^2S 模式。
14	BIDIOE	RW	0	双向模式输出使能。

Puya Confidential

				与 BIDIMODE 位一起配置“单线双向”模式下数据的输出方向。 0：输出禁止（只接收模式） 1：输出使能（只发送模式） 在主模式下，使用 MOSI 引脚；在从模式下，使用 MISO 引脚。 注：该位不适用于 I^2S 模式。
13	CRCEN	RW	0	硬件 CRC 校验使能。 0：禁止 CRC 计算； 1：启动 CRC 计算。 注：只有在禁止 SPI 时(SPE=0)，才能写该位，否则出错。 注：该位不适用于 I^2S 模式。
12	CRCNEXT	RW	0	下一个发送 CRC。 0：下一个发送的值来自发送缓冲区。 1：下一个发送的值来自发送 CRC 寄存器。 注：在 SPI_DR 寄存器写入最后一个数据后应马上设置该位。 注：该位不适用于 I^2S 模式。
11	DFF	RW	0	数据帧格式 0：使用 8 位数据帧格式进行发送/接收； 1：使用 16 位数据帧格式进行发送/接收。 注：只有当 SPI 禁止(SPE=0)时，才能写该位，否则出错。 注：该位不适用于 I^2S 模式。
10	RXONLY	RW	0	仅接收控制。 该位和 BIDIMODE 位一起决定在“双线单向”模式下的传输方向。在多个从机的配置中，在未被访问的从机上该位置 1，使得只有被访问的从机才有输出，因而不会造成数据线上有数据冲突。 0：全双工（发送和接收） 1：禁止输出（只接收模式） 注：该位不适用于 I^2S 模式。
9	SSM	RW	0	软件从机管理。 当 SSM 置位，NSS 引脚上的电平由 SSI 位的值决定。 0：禁止软件从机管理 1：使能软件从机管理 注：该位不适用于 I^2S 模式和 SPI TI 模式。
8	SSI	RW	0	内部从机选择。 该寄存器只有当 SSM=1 时才有效。该寄存器决定了 NSS 上的电平，在 NSS 引脚上的 I/O 操作无效。 注：该位不适用于 I^2S 模式和 SPI TI 模式。
7	LSBFIRST	RW	0	帧格式。 0：先发送 MSB 1：先发送 LSB

Puya Confidential

				通讯进行时不能改变该寄存器的值。 注：该位不适用于 I²S 模式和 SPI TI 模式。
6	SPE	RW	0	SPI 使能。 0：禁止 SPI 1：使能 SPI 注：该位不适用于 I²S 模式。
5:3	BR[2:0]	RW	3'h0	波特率控制。 000：f_{PCLK}/2 001：f_{PCLK}/4 010：f_{PCLK}/8 011：f_{PCLK}/16 100：f_{PCLK}/32 101：f_{PCLK}/64 110：f_{PCLK}/128 111：f_{PCLK}/256 通讯进行时不能改变该寄存器的值。 注：该位不适用于 I²S 模式。
2	MSTR	RW	0	主机选择。 0：配置为从机 1：配置为主机 通讯进行时不能改变该寄存器的值。 注：该位不适用于 I²S 模式。
1	CPOL	RW	0	时钟极性。 0：空闲状态时，SCK 保持低电平 1：空闲状态时，SCK 保持高电平 通讯进行时不能改变该寄存器的值。 注：除了在 TI 模式下应用 CRC 的情况外，该位不适用于 I²S 模式和 SPI TI 模式。
0	CPHA	RW	0	时钟相位。 0：数据采样从第一个时钟边沿开始 1：数据采样从第二个时钟边沿开始 通讯进行时不能改变该寄存器的值。 注：除了在 TI 模式下应用 CRC 的情况外，该位不适用于 I²S 模式和 SPI TI 模式。

26.5.2 SPI 控制寄存器 2（SPI_CR2）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRF	SAMPLE_SELECT	Res.	Res.	Res.	Res.	Res.	Res.	TXEIE	RXNEIE	ERRIE	CLRTXFIFO	Res.	SSOE	TXDMAEN	RXDMAEN
RW	RW							RW	RW	RW	RW		RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	FRF	RW	0	帧格式： 0：SPI Motorola 模式 1：SPI TI 模式 只有在禁止 SPI（SPE=0）后才能对此位进行操作
14	SAMPLE_SELECT	RW	0	采样方式： 0：半个 cycle 采样 1：1 个 cycle 采样
13:8	Reserved	-	-	保留
7	TXEIE	RW	0	发送缓冲区空中断使能 0：禁止 TXE 中断 1：使能 TXE 中断。TXE=1 时产生中断请求。
6	RXNEIE	RW	0	接收缓冲区非空中断使能 0：禁止 RXNE 中断 1：使能 RXNE 中断。RXNE=1 时产生中断请求。
5	ERRIE	RW	0	错误中断使能。 0：禁止错误中断 1：使能错误中断。当 CRCERR、OVR 或 MODF 为 1 时，产生中断请求。
4	CLRTXFIFO	RW	0	清空 TXFIFO 软件置位，硬件复位 0：没作用 1：清空 TXFIFO 注：只有当 SPI 禁止（SPE=0）时，才能写该位，否则无效。
3	Reserved	-	-	保留
2	SSOE	RW	0	SS 输出使能。 0：禁止在主模式下 SS 输出，该设备可以工作在多主机模式 1：开启主模式下 SS 输出，该设备不能工作在多主机模式。 注：I^2S 模式下不使用。
1	TXDMAEN	RW	0	发送缓冲区 DMA 使能。 0：禁止发送缓冲区 DMA 1：使能发送缓冲区 DMA。当 TXE=1，则发出 DMA 请求。
0	RXDMAEN	RW	0	接收缓冲区 DMA 使能。 0：禁止接收缓冲区 DMA 1：使能接收缓冲区 DMA。当 RXNE=1，则发出 DMA 请求。

26.5.3 SPI 状态寄存器（SPI_SR）

偏移地址：0x08

复位值：0x0000 0002

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTLVL [1:0]		FRLVL [1:0]		FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
			R	R	R	R	R	R	R	R	RC_W0	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
12:11	FTLVL[1:0]	R	2'h0	FIFO 发送级别。硬件置位，硬件清零 00：FIFO 空 01：1/4 FIFO 10：1/2 FIFO 11：FIFO 满（当 FIFO 阈值大于 1/2，即认为是满） 注：I²S 模式下不使用。
10:9	FRLVL[1:0]	R	2'h0	FIFO 接收级别。硬件置位，硬件清零 00：FIFO 空 01：1/4 FIFO 10：1/2 FIFO 11：FIFO 满 注：I²S 模式和带 CRC 校验的 SPI 仅接收模式下不使用。
8	FRE	R	0	FRE：帧格式错误。 该标志用于 SPI TI 从模式。 此标志由硬件置 1，在读取 SPI_SR 时由软件复位。 0：未发生帧格式错误 1：发生帧格式错误
7	BSY	R	0	忙标志。 0：SPI 不忙； 1：SPI 处于通讯，或者发送缓冲非空。
6	OVR	R	0	溢出标志。 0：无溢出错误 1：产生溢出错误 该寄存器由硬件置位，或者软件序列复位（上溢和下溢序列不同）。
5	MODF	R	0	模式故障。 0：无模式故障 1：发生模式故障 该寄存器由硬件置位，或者软件序列复位。 注：I²S 模式下不适用。
4	CRCERR	RC_W0	0	CRC 错误标志。 0：收到的 CRC 值和 SPI_RXCRCR 寄存器中的值匹配； 1：收到的 CRC 值和 SPI_RXCRCR 寄存器中的值不匹配。 该位由硬件置位，由软件写'0'而复位。 注：I²S 模式下不适用。
3	UDR	R	0	下溢标志位。 0：未发生下溢； 1：发生下溢错误。

Puya Confidential

				硬件置位，软件序列清零。 注：SPI 模式下不适用。
2	CHSIDE	R	0	声道控制。 0：发送或者接收左声道； 1：发送或者接收右声道。 注：在 SPI 和 I^2S PCM 模式下不适用。
1	TXE	R	1	发送缓冲空。 0：发送缓冲非空 1：发送缓冲为空
0	RXNE	R	0	接收缓冲非空。 0：接收缓冲非空 1：接收缓冲为空

26.5.4 SPI 数据寄存器（SPI_DR）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	DR[15:0]	RW	16'h0	数据寄存器。 要发送或者接收到的数据。 数据寄存器作为 RxFIFO 和 TxFIFO 的接口。当要读数据，实际访问 RxFIFO，而要写数据，实际访问 TxFIFO。 注：取决于 DFF 位（数据帧宽度选择），数据发送或者接收是 8 位或者 16 位。 对于 8 位数据帧，数据寄存器是基于右对齐的 8 位数据进行发送和接收的。当在接收模式，DR[15:8]硬件置为 0。 对于 16 位数据帧，数据寄存器是 16 位的，整个 DR[15:0]都用作发送和接收。

26.5.5 SPI CRC 多项式寄存器（SPI_CRCPR）

偏移地址：0x10

复位值：0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	CRCPOLY[15:0]	RW	16'h7	CRC 多项式寄存器。 该寄存器包含了 CRC 计算时用到的多项式。

Puya Confidential

				其复位值为 0x0007，根据应用可以设置其他数值。 注：在 I²S 模式下不适用。 注：多项式值只能是奇数，不支持偶数值。

26.5.6 SPI 接收 CRC 寄存器（SPI_RXCRC）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXCRC [15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	RXCRC[15:0]	R	16'h0	接收 CRC 寄存器 在启用 CRC 计算时，RXCRC[15:0]中包含了依据收到的字节计算的 CRC 数值。当在 SPI_CR1 的 CRCEN 位写入'1'时，该寄存器被复位。CRC 计算使用 SPI_CRCPR 中的多项式。 当数据帧格式被设置为 8 位时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；当数据帧格式为 16 位时，寄存器中的所有 16 位都参与计算，并且按照 CRC16 的标准。 注：当 BSY 标志为'1'时读该寄存器，将可能读到不正确的数值。 注：在 I²S 模式下不适用。

26.5.7 SPI 发送 CRC 寄存器（SPI_TXCRC）

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXCRC [15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	TXCRC[15:0]	R	16'h0	接收 CRC 寄存器 在启用 CRC 计算时， RXCRC[15:0]中包含了依据收到的字节计算的 CRC 数值。当在 SPI_CR1 的 CRCEN 位写入'1'时，该寄存器被复位。CRC 计算使用 SPI_CRCPR 中的多项式。 当数据帧格式被设置为 8 位时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；当数据帧格式为 16 位时，寄存器中的所有 16 位都参与计算，并且按照 CRC16 的标准。

Puya Confidential

				注：当 BSY 标志为‘1’时读该寄存器，将可能读到不正确的数值。 注：在 I^2S 模式下不适用。

Puya Confidential

27. 内部集成电路接口（I²C）

27.1 I²C 简介

I²C（Inter-integrated circuit）总线接口连接微控制器和串行 I²C 总线。它控制所有 I²C 总线特定的顺序、协议、仲裁和时序。支持标准（Sm）、快速（Fm）、快速增强模式（Fm+）。根据特定设备的需要，可以使用 DMA 以减轻 CPU 的负担。

27.2 I²C 主要特性

- 从机和主机模式
- 作为主机
  - 产生时钟
  - 产生起始和停止信号
- 作为从机
  - 可编程的 I²C 地址检测
  - 可响应 2 个从地址的双地址能力（其中 1 个可配置掩码）
  - 检测停止位
- 支持 7/10 位寻址模式
- 支持广播呼叫功能
- 支持不同的通讯速度
  - 标准（高至 100 kHz）
  - 快速（高至 400 kHz）
  - 快速增强（高至 1 MHz）
- 状态标志位：
  - 发送/接收模式标志
  - 字节传输完成标志
  - I²C 总线忙标志
- 错误标志位
  - 主机模式仲裁丢失
  - 地址/数据传输后的应答（ACK）失败
  - 起始/停止条件错误
  - 禁止拉长时钟功能时的上溢或下溢
- 2 个中断向量
  - 1 个事件中断，用于地址/数据通讯成功
  - 1 个错误中断
- 可选的拉长时钟功能
- 支持 DMA 功能
- 可配置的 PEC（信息包错误检测）的产生或校验：

Puya Confidential

- 发送模式中 PEC 值可以作为最后一个字节传输
- 用于最后一个接收字节的 PEC 错误校验

■ 支持 SMBus 2.0

- 25 ms 时钟低电平超时延时
- 10 ms 主设备累积时钟低电平扩展时间
- 25 ms 从设备累积时钟低电平扩展时间
- 带 ACK 控制的硬件 PEC 产生/校验
- 支持地址分辨协议（ARP）
- 超时和空闲条件检测

■ 兼容 SMBus

■ 支持地址匹配时从 Stop 模式唤醒

- 唤醒过程中支持可配置的 timeout 时间

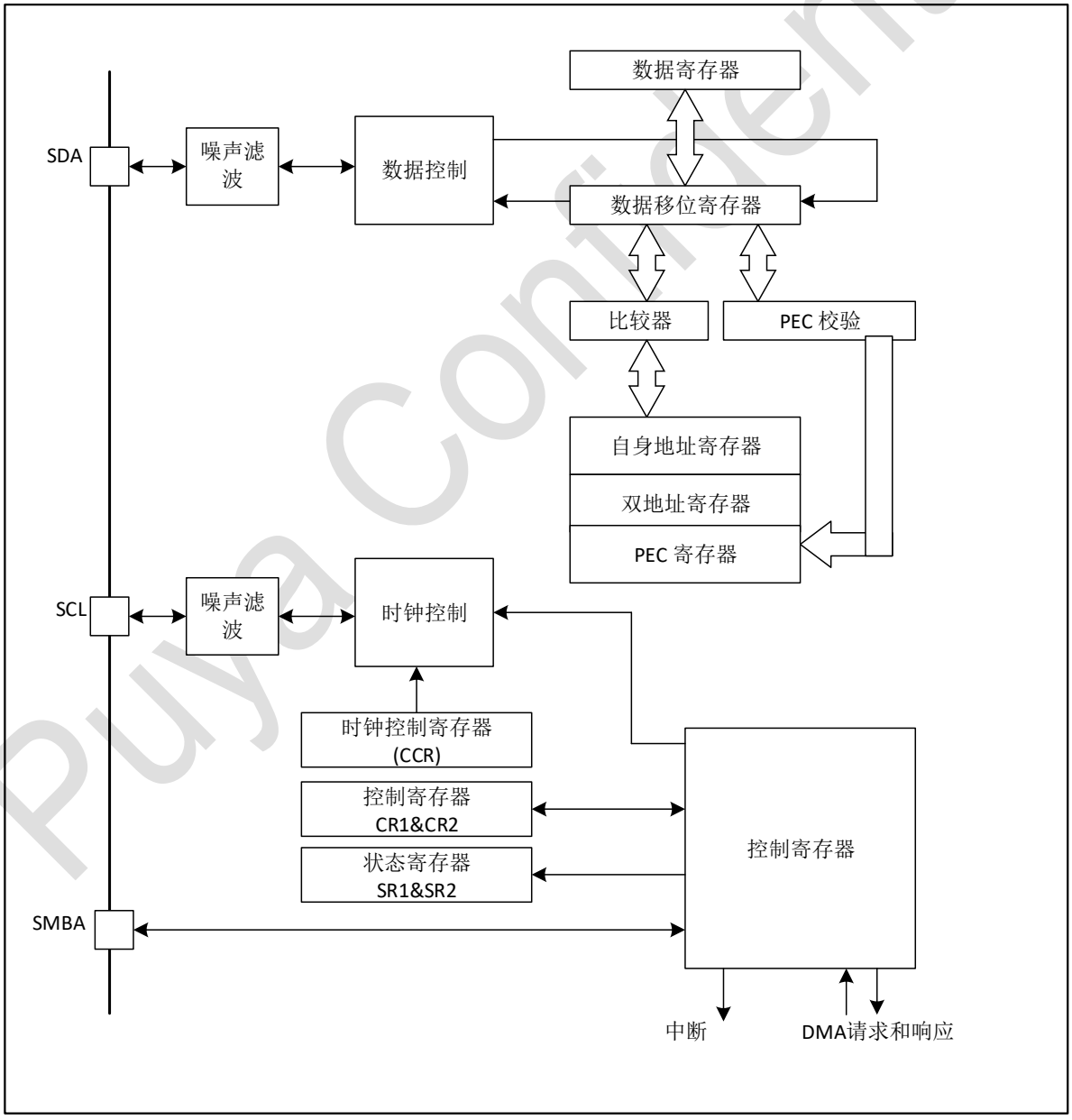


图 27-1 I²C 模块

Puya Confidential

27.3 I²C 功能描述

27.3.1 简介

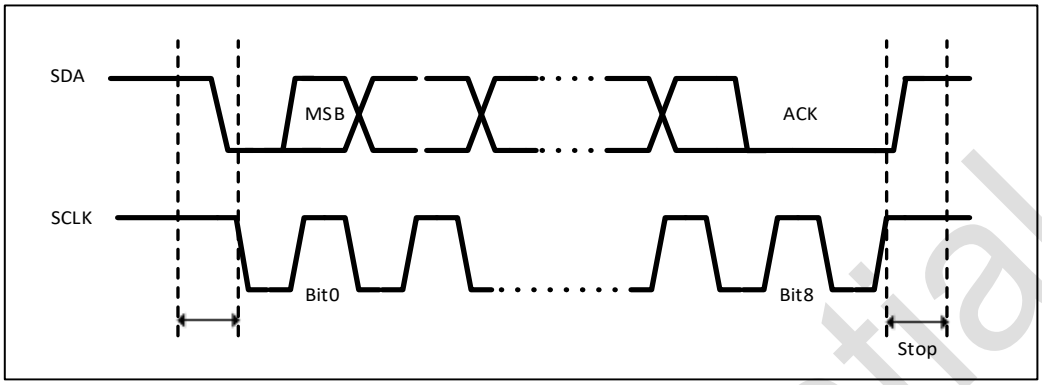


图 27-2 I²C 总线协议

I²C 支持以下四种模式：

- 从发送器模式（Slave transmitter）
- 从接收器模式（Slave receiver）
- 主发送器模式（Master transmitter）
- 主接收器模式（Master receiver）

默认情况下，I²C 接口总是工作在从模式。接口在生成起始条件后，自动地从从模式到主模式；在允许多主机功能的情况下，当仲裁丢失或产生停止信号时，则从主模式切换到从模式

作为主机，I²C 接口启动数据传输，并产生时钟信号。串行数据的传输总是以起始条件开始，并以停止条件结束。起始条件和停止条件都是在主机模式下由软件控制产生。

作为从机，I²C 接口能识别自己的地址（7 位或 10 位）和广播呼叫地址。软件能够控制开启或禁止对广播呼叫地址的识别。

数据和地址按 8 位（字节）进行传输，高位在前。跟在起始条件后的 1 或 2 个字节是地址（7 位模式为 1 个字节，10 位模式为 2 个字节）。地址只在 master 模式发送。

在一个字节传输后的第一个始终时钟内，接收方必须回送一个应答位（ACK）给发送方。

27.3.2 时钟

I²C 模块的输入时钟是 APB 时钟。模块内部生成 SCL 时钟再输出到总线上给从机。

SCL 产生机制如下：有一个计数器根据 CCR 寄存器的值来计数高低电平（CCR 寄存器决定高低电平的长度）。此外，CCR 寄存器还控制标准模式，快速模式和快速增强模式的选择，以及 SCL 高低电平的占空比。

27.3.3 包错误校验（PEC）

包错误校验（PEC）计算器是用于提高通信的可靠性，这个计算器使用一个可编程的多项式对每一位串行数据进行计算。

- PEC 计算由 I2C_CR1 寄存器的 ENPEC 位使能。PEC 使用 CRC-8 算法对所有信息字节进行计算，包括地址和读/写位在内。

Puya Confidential

- 在发送时：在最后一个 TxE 事件时设置 I2C_CR1 寄存器中的 PEC 位，PEC 将在这个字节后被发送。
- 在接收时：在最后一个 RxNE 事件时设置 I2C_CR1 寄存器中的 PEC 位，如果下个接收到的字节不等于内部计算的 PEC，接收器会发送一个 NACK。如果是主接收器，不管校对的结果如何，PEC 后都将发送 NACK。PEC 位必须在接收当前字节的 ACK 脉冲之前设置。

- 在 I2C_SR1 寄存器中可获得 PECERR 错误标记/中断。
- 如果 DMA 和 PEC 计算器都被激活：
  - 在发送时：当 I2C 接口从 DMA 控制器处接收到 EOT(对应 DMA 传输字节数) 信号时，它在最后一个字节后自动发送 PEC。
  - 接收时：当 I2C 接口从 DMA 处接收到一个 EOT_1(对应 DMA 传输字节数-1) 信号时，它将自动把下一个字节作为 PEC，并且将检查它。在接收到 PEC 后产生一个 DMA 请求。
- 为了允许中间 PEC 传输，在 I2C_CR2 寄存器中有一个控制位(LAST 位)用于判别是否真是最后一个 DMA 传输。如果确实是最后一个主接收器的 DMA 请求，在接收到最后一个字节后自动发送 NACK。
- 仲裁丢失时 PEC 计算失效。

27.3.4 I2C 的从模式

默认情况下，I²C 接口总是工作在 slave 模式。从 slave 模式切换到 master 模式，需要产生一个起始条件。为了产生正确的时序，必须在 I2C_CR2 寄存器中设定该模块的输入时钟。输入时钟的频率必须至少是：

标准模式下为：4 MHz

快速模式下为：8 MHz

快速增强模式下为：16 MHz

设置从机地址及地址屏蔽:

一旦检测到起始条件，在 SDA 线上接收到的地址，被送到移位寄存器，并与芯片的地址 OAR1 或者广播呼叫地址(如果 ENGC=1)相比较。如果启用了额外地址（ENDUAL=1）,地址会和芯片的 OAR2 地址进行比较，并且 OAR2 寄存器中的地址可以通过 OA2MSK[2:0]进行屏蔽，屏蔽从 OA2 高第 7 位开始，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6] 或 OA2[7] 与接收到的地址作比较。如果 OA2MSK=7，接收到的所有 7 位地址（保留地址除外）均得到应答。

如果启用了额外地址（ENTRIPLE=1），地址会和芯片的 OAR3 地址进行比较。

地址不匹配（头段或地址不匹配）：I2C 接口将其忽略并等待另一个起始条件。

地址匹配（地址匹配）：I2C 接口产生以下时序：

- 如果 ACK 被软件置'1'，则产生一个应答脉冲
- 硬件置位 ADDR 位，如果设置了 ADDRIE 位，则产生中断
- 如果 ENDUAL=1，软件必须读 DUALF 位，以确认响应了哪个从地址

在从模式下 TRA 位指示当前是处于接收器模式还是发送器模式

Puya Confidential

从机发送模式

在接收到地址并清除 ADDR 位后，（如果地址字节的最低位是 1）Slave 将数据（字节）从 DR 寄存器，经由内部移位寄存器发送到 SDA 上。

Slave 拉低 SCL，直到 ADDR 位被清除，并且待发送数据已写入 DR 寄存器（参考 EV1、EV3）。

当收到应答脉冲时：TxE 位被硬件置位，如果设置了 TXIE 位，则产生一个中断。

如果 TxE 位被置位，但在下一个数据发送结束之前，没有新数据写入到 I2C_DR 寄存器，则 BTF 位被置位。Slave 拉低 SCL，直到 BTF 位被软件清零（再写入 I2C_DR 寄存器）。

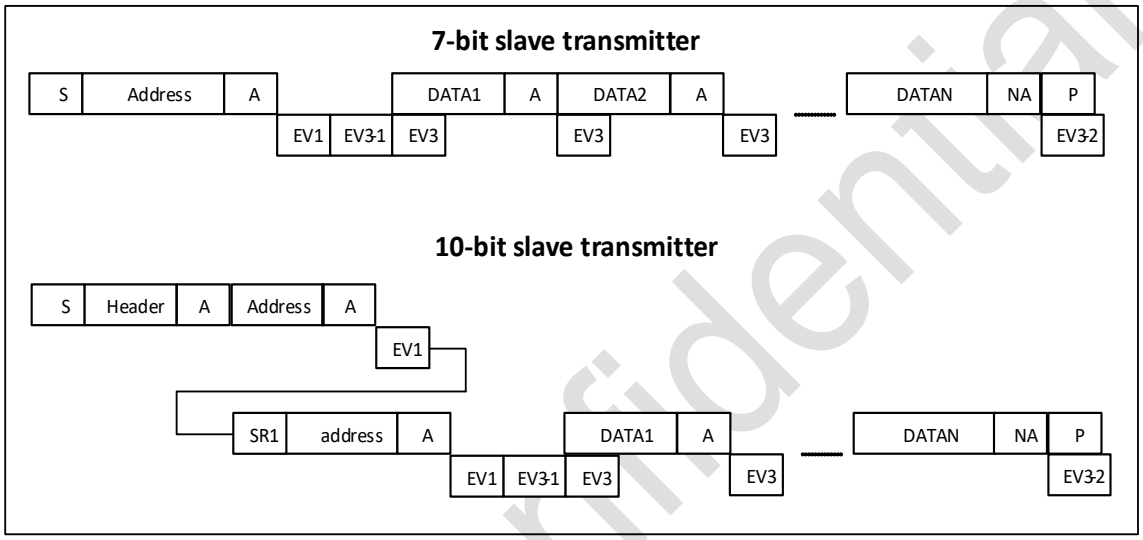


图 27-3 从发送器的传送序列图

说明: S= Start（起始条件），SR= Repeated Start（重复的起始条件），P= Stop（停止条件），A= Acknowledge（响应），NA= Non-acknowledge（不响应），EVx= Event（ITEVFEN= 1 时产生中断）

EV1: ADDR=1，通过先读 SR1 寄存器，再读 SR2 寄存器，清零 ADDR 位

EV3-1：TxE=1，移位寄存器为空，数据寄存器为空，向 DR 寄存器写 Data1

EV3：TxE=1，移位寄存器不为空，数据寄存器为空，向 DR 寄存器写（Data2）清零 TxE

EV3-2: AF=1；软件向 AF 位写 0 清零该位

注:

1. EV1 和 EV3-1 事件拉低 SCL，直到对应的软件序列结束。
2. EV3 的软件序列必须在当前字节传输结束之前完成
3. RxNE 的中断处理需要放在较高优先级。

从机接收模式

在接收到地址并清除 ADDR 后，（如果地址字节的最低位是 0）slave 将通过内部移位寄存器把从 SDA 线接收到的字节存进 DR 寄存器。I^2C 接口在接收到每个字节后都执行下列操作：

- 如果设置了 ACK 位，则产生一个应答脉冲
- 硬件设置 RxNE=1。如果设置了 RXIE 位，则产生一个中断。

Puya Confidential

如果 RxNE 被置位，并且在接收新的数据结束之前，DR 寄存器未被读出，则 BTF 位被置位，在清除 BTF（读 I2C_DR 寄存器）之前，slave 一直拉低 SCL。(见下图)。

注：

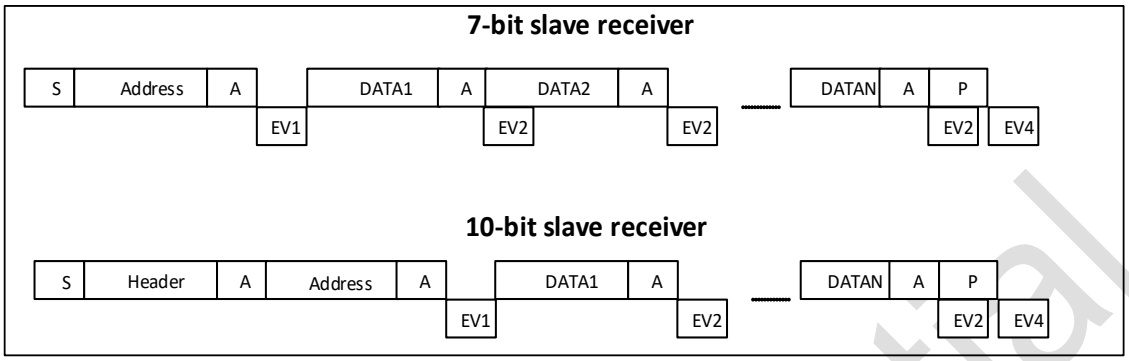


图 27-4 从接收器的传送序列图

说明: S= Start（起始条件），SR= Repeated Start（重复的起始条件），P= Stop（停止条件），A= Acknowledge（响应），NA= Non-acknowledge（不响应），EVx= Event(ITEVFEN= 1 时产生中断)

EV1：ADDR=1，通过先读 SR1，后读 SR2 实现 ADDR 的清零

EV2：RxNE=1，读 DR 寄存器清零该位

EV4：STOPF=1，通过先读 SR1 寄存器，后写 CR1 寄存器实现对该位的清零。

注:

1）EV1 事件拉低 SCL，直到相应软件序列结束。

2）EV2 软件序列必须在当前字节传输完成之前完成。

3）当用户检查 SR1 寄存器内容后，应该对每个置位的标志位，进行完整的清除序列。比如 ADDR 和 STOPF 标志位，需要用以下序列：

如果 ADDR=1，先读 SR1，再读 SR2；如果 STOPF=1，先读 SR1，再写 CR1。

这样做的目的是确保如果 ADDR 和 STOPF 都被置位，都能被发现并且清除掉。

4）RxNE 的中断处理需要放在较高优先级。

关闭通信

在传输完最后一个数据字节后，主机产生一个停止条件，从机检测到该条件时：

- 硬件置位 STOPF，如果设置了 STOPIE 位，则产生一个中断。

通过先读 SR1，后写 CR1，实现对 STOPF 位的清零。

27.3.5 I2C 的主模式

在主模式时，I²C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始，并以停止条件结束。

当通过 START 位在总线上产生了起始条件，设备就进入了主机模式。

以下是主机模式所要求的操作顺序：

- 在 I2C_CR2 寄存器中设定该模块的输入时钟以产生正确的时序
- 配置 I2C_CCR 寄存器
- 配置 I2C_TRISE 寄存器
- 配置 I2C_CR1 寄存器中的 PE 启动外设

Puya Confidential

- 置 I2C_CR1 寄存器中的 START 位为 1，产生起始条件

I²C 模块的输入时钟频率必须至少是：

- 标准模式下为：4 MHz
- 快速模式下为：8 MHz
- 快速增强模式下为：16 MHz

主机产生时钟

CCR 寄存器以输入时钟上升沿计数，产生 SCL 的高低电平。由于从机可能拉长 SCL 信号，在 SCL 上升沿产生后，主机在 I2C_TRISE 寄存器所设置的时间到达时，检测来自总线的 SCL 信号。

- 如果 SCL 是低电平，意味着从机正在拉长 SCL 总线，高电平计数器停止计数，直到 SCL 被检测到高电平。这是为了确保 SCL 参数的最小高电平时间。
- 如果 SCL 是高电平，高电平计数器保持计数。

实际上，即使从机不拉长 SCL，从 SCL 上升沿产生，到 SCL 上升沿被发现，这样的反馈回路也是要花费些时间的。这个回路的时间与 SCL 的上升时间有关系，再加上 SCL 输入路径的模拟噪声滤波延时，以及芯片内部由于用 APB 时钟进行的 SCL 同步时间。反馈回路的最大时间在 TRISE 寄存器中设置，所以无论 SCL 上升时间如何，SCL 的频率保持稳定。

起始条件

当 BUSY=0 时，设置 START=1，I²C 接口将产生一个起始条件，并切换至主机模式（MSL 被置位）。

注：在主机模式下设置 START 位，将在当前字节传输完成后，由硬件产生一个重开始条件。

一旦发出起始条件：

- SB 位被硬件置位，如果设置了 SBIE 位，则会产生一个中断。
- 主机读 I2C_SR1 寄存器，再把从机地址写入 I2C_DR 寄存器。

从机地址发送

主机将从机的地址通过内部移位寄存器被送到 SDA 线上。

- 在 10 位地址模式时，发送一个头段序列产生以下事件：
  - ADD10 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。
  - 然后主机读 SR1 寄存器，并且将第二个地址字节写入 DR 寄存器。
  - ADDR 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。
  - 随后主机读 SR1 寄存器，然后读 SR2 寄存器。
- 在 7 位地址模式时，将送出一个地址字节。

一旦该地址字节被送出，

- ADDR 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。
- 随后主设备等待一次读 SR1 寄存器，跟着读 SR2 寄存器。

根据送出从机地址的最低位，主设备决定进入是发送器模式，还是进入接收器模式。

- 在 7 位地址模式时，
  - 要进入发送器模式，主设备发送从地址时让最低位等于 0。
  - 要进入接收器模式，主设备发送从地址时让最低位等于 1。
- 在 10 位地址模式时
  - 要进入发送器模式，主设备先送头字节（11110xx0），然后送最低位等于 0 的从地址。(头段字节中的 xx 是 10 位地址中的最高 2 位)。

Puya Confidential

— 要进入接收器模式，主设备先送头字节（11110xx0），然后送最低位等于 1 的从地址。然后再重新发送一个起始条件，后面跟着头字节（11110xx1）。(头字节中的 xx 是 10 位地址中的最高 2 位)。

TRA 位指示主设备是在接收器模式还是发送器模式

主机发送模式

在发送了地址和清除了 ADDR 位后,主机通过内部移位寄存器将数据字节从 DR 寄存器发送到 SDA 线上。

主机等待，直到第一个数据字节被写入 DR 寄存器（参见 EV8_1）。

当收到 ACK 脉冲时，TxE 位被硬件置位，如果设置了 TXIE 位，则产生一个中断。

如果 TxE 被置位，且在上一次数据发送结束之前，没有写新的数据字节到 DR 寄存器，则 BTF 被硬件置位。在清除 BTF（写 I2C_DR 寄存器）之前，I2C 接口将保持 SCL 为低电平。

关闭通信

在 DR 寄存器中写入最后一个字节后，通过设置 STOP 位产生一个停止条件（见图的 EV8_2），然后 I^2C 接口将自动回到从模式（MSL 位清除）。

注：当 TxE 或 BTF 位置位时，在出现 EV8_2 事件时产生停止条件。

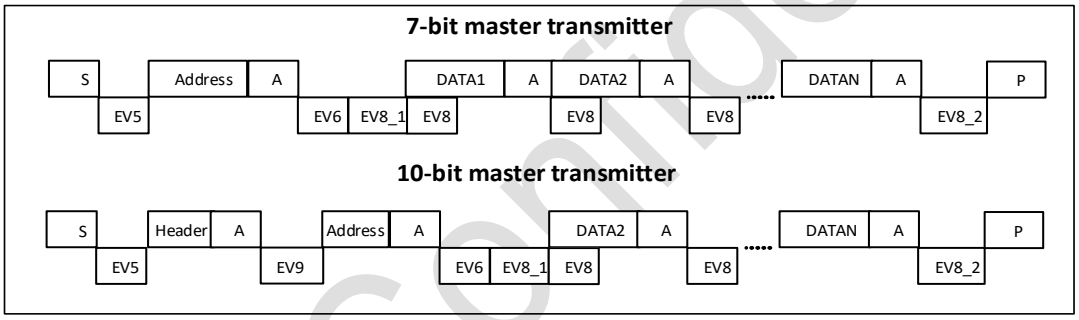


图 27-5 主发送器传送序列图

说明： S= Start（起始条件），SR= Repeated Start（重复的起始条件），P= Stop（停止条件），A= Acknowledge（响应），NA= Non-acknowledge（不响应），EVx= Event(ITEVFEN= 1 时产生中断)

EV5: SB=1，读 SR1 然后将地址写入 DR 寄存器将清除该事件

EV6: ADDR=1,读 SR1，再读 SR2，清除该事件

EV8_1: TXE=1，移位寄存器空

EV8: TXE=1，写 DR 寄存器清除该事件

EV8_2: TXE=1，BTF=1，产生停止条件时由硬件清除

EV9: ADDR10=1,读 SR1 然后写 DR 寄存器清除该事件

注:

1．EV5，EV6, EV8_1 和 EV8_2 事件，拉长 SCL 的低电平，直到相应的软件序列执行结束

2．EV8 软件序列必须在当前字节发送完成前执行完毕。若 EV8 的软件序列不能在当前传输的字节结束前完成，则推荐使用 BTF 代替 TxE

Puya Confidential

主机接收模式

在发送地址和清除 ADDR 之后，I²C 接口进入主接收器模式。在此模式下，I²C 接口从 SDA 线接收数据字节，并通过内部移位寄存器送至 DR 寄存器。在接收到每个字节后，I²C 接口依次执行以下操作：

- 如果 ACK 位被置位，则发出一个应答脉冲。
- 硬件设置 RxNE=1，如果设置了 RXIE 位，则会产生一个中断。

如果 RxNE 位被置位，并且在接收新数据结束前，DR 寄存器中的数据没有被读走，硬件将设置 BTF=1，在清除 BTF 之前 I²C 接口将保持 SCL 为低电平；读出 I2C_SR1 之后再读出 I2C_DR 寄存器将清除 BTF 位。

关闭通信

方法 1：该方法的应用场景是，当 I²C 的中断被设成应用程序中最高优先级

主机在接收到从机最后一个字节后，发送一个 NACK。接收到 NACK 后，从机释放对 SCL 和 SDA 线的控制。主机就可以发送一个停止/重开始条件。

1. 为了在收到最后一个字节后产生一个 NACK 脉冲，在读倒数第二个数据字节之后(在倒数第二个 RxNE 事件之后)必须清除 ACK 位。
2. 为了产生一个停止/重开始条件，软件必须在读倒数第二个数据字节之后(在倒数第二个 RxNE 事件之后)设置停止/开始位。
3. 当接收单个字节时，清除 ACK 和停止条件的产生位要刚好在 EV6 之后(EV6_1 时，清除 ADDR 之后)。在产生了停止条件后，I²C 接口自动回到从模式(MSL 位被清除)。

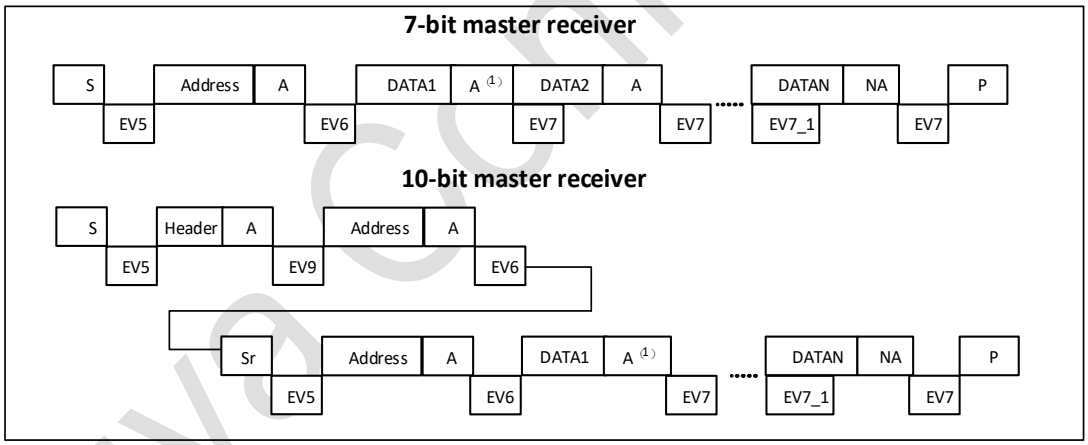


图 27-6 方法 1：主模式发送时的时序

说明：S= Start（起始条件），SR= Repeated Start（重复的起始条件），P= Stop（停止条件），A= Acknowledge（响应），NA= Non-acknowledge（不响应），EVx= Event(ITEVFEN= 1 时产生中断)

EV5：SB=1，读 SR1，再写 DR 寄存器，该位被清零

EV6：ADDR=1，读 SR1，再读 SR2，该位被清零

EV6_1：无相关的标志事件，仅用作 1 个字节的接收。

EV7：RxNE=1，读 DR 寄存器，该位被清零

EV7_1：RxNE=1，读 DR 寄存器，写 ACK=0 并置位 STOP

EV9：ADD10=1，读 SR1 然后写 DR 寄存器清除该事件

注：

Puya Confidential

如果是单个字节接收，则上述标注为（1）的地方会是 NA

EV5，EV6 事件，拉长 SCL 的低电平，直到相应的软件序列执行结束

EV7 软件序列必须在当前字节发送完成前执行完毕。在 EV7，软件序列不能在当前传输的字节传输完成前执行完毕，则推荐使用 BTF 代替 RXNE。

EV6_1 或者 EV7_1 的软件序列必须在当前字节传输的 ACK 位发送之前完成。

方法 2：这个方法的应用场景是，I²C 的中断在应用中不是最高优先级，或者使用查询方式

用这个方法，如果 DataN-2 没有被读，在 DataN-1 之后，通讯会被拉长（RxNE 和 BTF 都被置位）。然后，在读 DR 寄存器的 DataN-2 前，清 ACK 位，以确保 ACK 位在 DataN ACK 之前被清掉。在读 DataN-2 之后，置位 STOP/START 位，并读 DataN-1。在 RxNE 置位后，读 DataN。

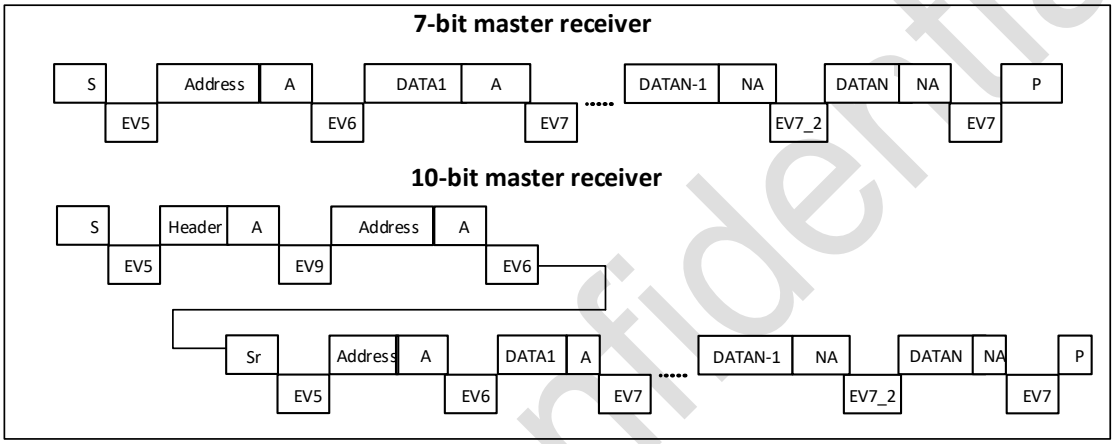


图 27-7 方法 2: N>2 时主模式发送时的时序

说明: S= Start（起始条件），SR= Repeated Start（重复的起始条件），P= Stop（停止条件），A= Acknowledge（响应），NA= Non-acknowledge（不响应），EVx= Event(ITEVFEN= 1 时产生中断)

EV5：SB=1，先读 SR1 寄存器先读 SR2 寄存器，再写 DR 寄存器，清零该位

EV6：ADDR，先读 SR1，再读 SR2，清零该位

EV7：RxNE=1，读 DR 寄存器清零该位

EV7_2：BTF=1，DataN-2 存在 DR 寄存器中，DataN-1 存在移位寄存器中，写 ACK=0，读 DR 寄存器中的 DataN-2。置位 STOP，读 DataN-1

EV9：ADD10=1，读 SR1 然后写 DR 寄存器清除该事件

注：

1. EV5，EV6 事件，拉长 SCL 的低电平，直到相应的软件序列执行结束
2. EV7 软件序列必须在当前字节发送完成前执行完毕。在 EV7，软件序列不能在当前传输的字节传输完成前执行完毕，则推荐使用 BTF 代替 RXNE。

- 当 3 个字节要被读走：
  - RXNE=1，DataN-2 没有读。
  - DataN-1 接收
  - BTF=1，移位和 DR 寄存器满。DR 寄存器存放了 DataN-2，移位寄存器存放了 DataN-1。此时，SCL 拉低，总线上没有其他要被接收的数据

Puya Confidential

- 清零 ACK 位
- 读 DR 寄存器中的 DataN-2，这将启动移位寄存器对 DataN 的接收
- DataN 接收完成（发送 NACK 条件）
- 写 START 或者 STOP 位
- 读 DataN-1
- RxNE=1
- 读 DataN

以上流程是针对 N > 2 的描述。1 个字节和 2 个字节的接收，要用不同的处理方式，参见以下描述：

■ 2 个字节接收的情况

- 置位 POS 和 ACK 位
- 等待 ADDR 置位
- 清零 ADDR 位
- 清零 ACK 位
- 等待 BTF 被置位
- 写 STOP 位
- 读 DR 两次

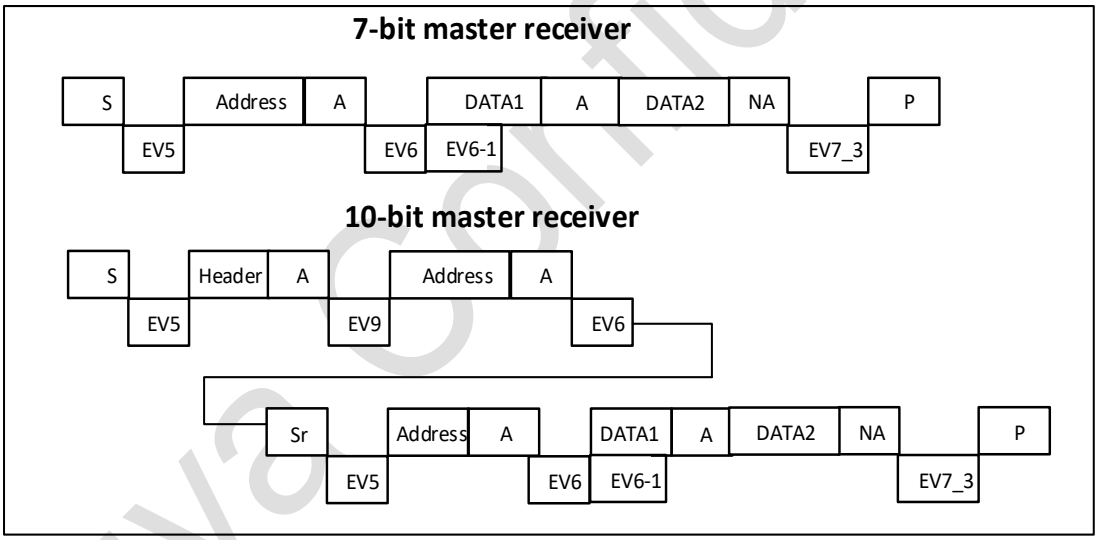


图 27-8 方法 2: N=2 时主模式发送时的时序

说明：S= Start（起始条件），SR= Repeated Start（重复的起始条件），P= Stop（停止条件），A= Acknowledge（响应），NA= Non-acknowledge（不响应），EVx= Event（ITEVFEN=1 时产生中断）

EV5：SB=1，先读 SR1 寄存器，再写 DR 寄存器，清零该位

EV6：ADDR=1，先读 SR1 寄存器，后读 SR2 寄存器，清零 ADDR 位

EV6_1：无相关的标志位事件。在 EV6 后，也就是地址被清零后，ACK 应该被清零

EV7_3：BTF=1，写 STOP=1，之后读两次 DR（Data1 和 Data2）

EV9：ADD10=1，读 SR1 然后写 DR 寄存器清除该事件

注:

1)EV5，EV6 事件，拉长 SCL 的低电平，直到相应的软件序列执行结束

Puya Confidential

2)EV6_1 的软件序列必须在当前字节传输的 ACK 位发送之前完成。

- 单个字节接收的情况
  - 在 ADDR 事件里，清零 ACK 位
  - 清零 ADDR
  - 写 STOP 或者 START 位
  - 在 RxNE 标志置位后，读数据

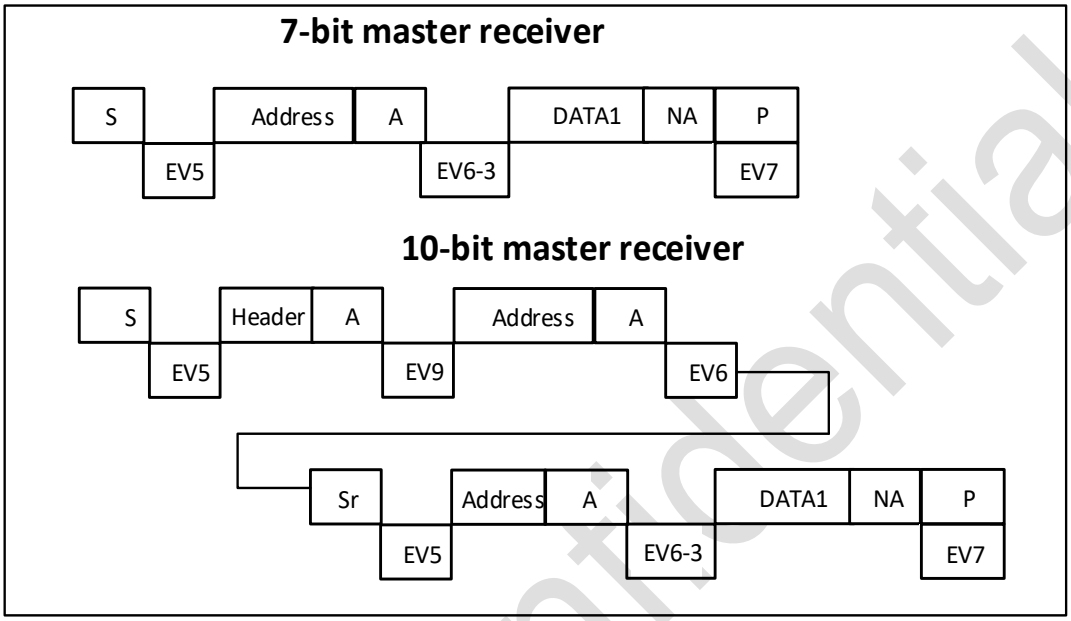


图 27-9 方法 2：N=1 时主模式发送时的时序

说明：S= Start（起始条件），SR= Repeated Start（重复的起始条件），P= Stop（停止条件），A= Acknowledge（响应），NA= Non-acknowledge（不响应），EVx= Event（ITEVFEN= 1 时产生中断)

EV5：SB=1，先读 SR1 寄存器，再写 DR 寄存器，清零该位

EV6_3：ADDR=1，写 ACK=0。先读 SR1 寄存器，后读 SR2 寄存器，清零 ADDR 位。在 ADDR 被清零后，写 STOP=1

EV7：RxNE=1，读 DR 寄存器清零该位

EV9：ADD10=1，读 SR1 然后写 DR 寄存器清除该事件

注:

EV5，EV6，EV8_1 和 EV8_2 事件，拉长 SCL 的低电平，直到相应的软件序列执行结束。

27.3.6 错误标志

总线错误（BERR）

在一个地址或数据字节传输期间，当 I²C 接口检测到一个外部的停止或起始条件则产生总线错误。此时：

- BERR 位被置位为'1'；如果设置了 ITERREN 位，则产生一个中断；
- 在从机模式：数据被丢弃，硬件释放总线。
  - 如果是错误的起始条件，从机认为是一个重开始，并等待地址或停止条件
  - 如果是错误的停止条件，从机按正常的停止条件操作，同时硬件释放总线
- 在主机模式：硬件不释放总线，同时不影响当前的传输状态。此时由软件决定是否要中止当前的传输。

Puya Confidential

应答失败（AF)

当接口检测到一个无应答位时，产生应答错误。此时：

- AF 位被置位，如果设置了 ITERREN 位，则产生一个中断
- 当发送器接收到一个 NACK 时，必须复位通讯。
  - 如果是处于从机模式，硬件释放总线
  - 如果是处于主机模式，软件必须生成一个停止条件或者重开始

仲裁丢失（ARLO)

当 I²C 接口检测到仲裁丢失时产生仲裁丢失错误，此时：

- ARLO 位被硬件置位，如果设置了 ITERREN 位，则产生一个中断
- I²C 接口自动回到从模式(MSL 位被清除)。当 I²C 接口丢失了仲裁，则它无法在同一个传输中响应它的从地址，但它可以在赢得总线的主机发送重开始条件之后响应
- 硬件释放总线

过载/欠载错误（OVR)

在从机模式下，如果禁止时钟延长，I²C 接口正在接收数据时，当它已经接收到一个字节(RxNE=1)，但在 DR 寄存器中前一个字节数据还没有被读出，则发生过载错误。

此时：

- 最后接收的数据被丢弃
- 在过载错误时，软件应清除 RxNE 位，发送器应该重新发送最后一次发送的字节

在从机模式下，如果禁止时钟延长，I²C 接口正在发送数据时，在下一个字节的时钟到达之前，新的数据还未写入 DR 寄存器(TxE=1)，则发生欠载错误。此时：

- 在 DR 寄存器中的前一个字节将被重复发出
- 用户应该确定在发生欠载运行错误时，接收端应丢弃重复接收到的数据。发送端应按 I2C 总线标准在规定的时间更新 DR 寄存器

在发送第一个字节时，必须在清除 ADDR 之后且在第一个 SCL 上升沿之前写入 DR 寄存器；如果不能做到这点，则接收方应该丢弃第一个数据。

27.3.7 DMA 功能

DMA 请求（当被使能时）仅用于数据传输。发送时数据寄存器变空，或接收时数据寄存器变满，则产生 DMA 请求。DMA 必须在当前字节传输结束之前被初始化和使能。DMAEN 位（I2C_CR2 寄存器中）必须在 ADDR 事件发生前使能。

在 master 或者 slave 模式，当时钟延长功能使能，DMAEN 可以在清零 ADDR 之前的 ADDR 事件期间置位。DMA 请求必须在当前字节传输完成之前响应。当 DMA 传输数据长度达到 DMA 设定的值时，DMA 控制器向 I²C 发送 EOT（End of transfer），并产生传输完成中断（如果中断使能位有效）：

- 主机发送模式：在 EOT 中断服务程序中，需禁止 DMA 请求，然后在等到 BTF 事件后，置位 STOP 条件。
- 主机接受模式：当要接收的数据数目大于或等于 2 时，DMA 控制器发送一个硬件信号 EOT_1（对应 DMA 传输字节数 - 1)。如果在 I2C_CR2 寄存器中设置了 LAST 位，硬件在发送完 EOT_1 后的

Puya Confidential

下一个字节，将自动发送 NACK。在中断允许的情况下，用户可以在 DMA 传输完成的中断服务程序中产生一个停止条件。

27.3.8 支持从 Stop 模式唤醒

在 MCU 处于 Stop 模式下，可以通过 I²C 接收地址，若地址匹配则可以唤醒 MCU，反之则保持 STOP 状态。

1）实现以上功能需要在进入 Stop 模式之前使能 I2C_CR1 的 PE 和 WUPEN，并且在 RCC 中使能 I2C 的时钟。

2）Stop 模式唤醒过程还支持可配置的 timeout 时间，可通过设置 I2C_CR1 的 WKUP_CNT 和 WKUP_DIV 进行配置，若使能 ITERREN，则在产生 TIMEOUT 之后唤醒 MCU，若不使能 ITERREN，则在产生 timeout 之后保持 Stop 状态。

27.3.9 系统管理总线 SMbus

系统管理总线（SMBus）是一个双线接口。通过它，各设备之间以及设备与系统的其他部分之间可以互相通信。它基于 I²C 操作原理。SMBus 为系统和电源管理相关的任务提供一条控制总线。一个系统利用 SMBus 可以和多个设备互传信息，而不需使用独立的控制线路。

系统管理总线 (SMBus) 标准涉及三类设备。从设备: 接收或响应命令的设备。主设备: 用来发布命令，产生时钟和终止发送的设备。主机：是一种专用的主设备，它提供与系统 CPU 的主接口。主机必须具有主-从机功能，并且必须支持 SMBus 提醒协议。在一个系统里只允许有一个主机。

利用系统管理总线，设备可提供制造商信息，告诉系统它的型号/部件号，保存暂停事件的状态，报告不同类型的错误，接收控制参数，和返回它的状态。SMBus 为系统和电源管理相关的任务提供控制总线。本产品的 I²C 模块支持 SMbus/PMbus 功能。

表 27-1 SMbus 与 I2C 比较的不同点

SMBus	I²C
最大传输速度 100 kHz	最大传输速度 400 kHz
最小传输速度 10 kHz	无最小传输速度
35 ms 时钟低超时	无时钟超时
固定的逻辑电平	逻辑电平由 V_{CC} 决定
不同的地址类型(保留、动态等)	7 位、10 位和广播呼叫从地址类型
不同的总线协议（快速命令、处理呼叫等)	无总线协议

SMBus 和 I²C 之间的相似点:

- 两条线的总线协议（1 个时钟，1 个数据）+可选的 SMBus 提醒线
- 主-从通信，主设备提供时钟
- 多主机功能
- SMBus 数据格式类似于 I²C 的 7 位地址格式

地址解析协议（ARP）

SMBus 的从地址冲突可以通过给每个从设备动态地分配一个新的唯一地址来解决。

ARP 有以下的属性:

- 地址分配利用标准 SMBus 物理层仲裁机制
- 当设备维持供电期间，分配的地址仍保持不变，允许设备在断电时保留其地址。

Puya Confidential

● 在地址分配后，没有额外的 SMBus 的打包开销(也就是说访问分配地址的设备与访问固定地址的设备所用时间是一样的）。

● 任何一个 SMBus 主设备可以遍历总线。

SMBus 提醒模式（ALERT）

SMBus 提醒是一个带中断线的可选信号，用于那些希望扩展它们的控制能力而牺牲一个引脚的设备。SMBALERT 和 SCL、SDA 信号一样，是一种线与信号。SMBALERT 通常和 SMBus 广播呼叫地址一起使用。与 SMBus 有关的消息为 2 字节。

一个只具有从功能的设备，可以通过 SMBALERT 发信号给主机表示它希望进行通信，这可通过设置 I2C_CR1 寄存器上的 ALERT 位实现。主机处理该中断并通过提醒响应地址 ARA（Alert Response Address，地址值为 0001100x）访问所有 SMBALERT 设备。只有那些将 SMBALERT 拉低的设备能应答 ARA。此状态是由 I2C_SR1 寄存器中的 SMBALERT 状态标记来标识的。主机执行一个修改过的接收字节操作。由从发送设备提供的 7 位设备地址被放在字节的 7 个最高位上，第八个位可以是 0 或 1。

如果多个设备把 SMBALERT 拉低，最高优先级设备（最小的地址）将在地址传输期间通过标准仲裁赢得通信权。在确认从地址后，此设备不得再拉低它的 SMBALERT，如果当信息传输完成后，主机仍看到 SMBALERT 低，就知道需要再次读 ARA。

没有实现 SMBALERT 信号的主机可以定期访问 ARA。

超时错误（TIMEOUT）

在定时规范上 I2C 和 SMBus 之间有很多差别。

SMBus 定义一个时钟低超时，35ms 的超时。SMBus 规定 $T_{LOW:SEXT}$ 为从设备的累积时钟低扩展时间。SMBus 规定 $T_{LOW:MEXT}$ 为主设备的累积时钟低扩展时间。更多超时细节请参考 2.0 版的 SMBus 规范(http://smbus.org/specs/)。I2C_SR1 中的状态标志 TIMEOUT 错误表明了这个特征的状态。

使用 SMBus 模式的接口

将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范规定的时间最大值之前检测出超时情况。

• $t_{TIMEOUT}$ 检查

要使能 $t_{TIMEOUT}$ 检查，必须将 12 位 TIMEOUTA[11:0]位编程为定时器重载值，以检查 $t_{TIMEOUT}$ 参数。必须将 TIDLE 位配置为“0”，以检测 SCL 低电平超时。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 的低电平持续时间超过（TIMEOUTA+1）x 2048 x t_{I2CCLK}，I2C_SR1 寄存器的 TIMEOUT 标志将置 1。（最大 $t_{TIMEOUT}$ = 25 ms）。

注： TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0] 位和 TIDLE 位的配置。

• $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查

必须根据外设配置为主设备还是从设备来配置 TIMEOUTB 定时器，以便为从设备校验 $t_{LOW:SEXT}$，为主设备校验 $t_{LOW:MEXT}$。由于标准只规定了最大值，用户可以为这两个参数选择相同的值。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。

如果 SMBus 外设延展 SCL 的累积时间超过 (TIMEOUTB+1) x 2048 x t_{I2CCLK}，则 I2C_SR1 寄存器中的 TIMEOUT 标志将置 1。

Puya Confidential

请参见下：不同 I2CCLK 频率下的 TIMEOUTB 设置示例。

注： TEXTEN 位置 1 时，不允许更改 TIMEOUTB 配置。

总线空闲检测

要使能 t_{IDLE} 检查，必须将 12 位 TIMEOUTA[11:0] 字段编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位配置为“1”，以检测 SCL 和 SDA 高电平超时。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 和 SDA 线的高电平持续时间超过 (TIMEOUTA+1) x 4 x t_{I2CCLK}，I2C_SR1 寄存器中的 TIMEOUT 标志将置 1。

请参见表:不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）。

注： TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

PMbus

PMbus 是基于 SMbus 而来的，传输逻辑与 SMbus 完全一致，区别在于 PMbus 定义了一些与电源管理相关的功能（由软件完成）。

SMBus: I2C_TIMEOUTR 寄存器配置示例

仅当支持 SMBus 功能时，才涉及本节内容。

将 $t_{TIMEOUT}$ 的最大持续时间配置为 25 ms：

表 27-2 不同 PCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms）

f_{PCLK}	TIMEOUTA[11:0]	TIDLE	TIMEOUTEN	$t_{TIMEOUT}$
8 MHz	0x61	0	1	98 x 2048 x 125 ns = 25 ms
16 MHz	0xC3	0	1	196 x 2048 x 62.5 ns = 25 ms
32 MHz	0x186	0	1	391 x 2048 x 31.25 ns = 25 ms
48 MHz	0x249	0	1	586 x 2048 x 20.08 ns = 25 ms

将 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的最大持续时间配置为 8 ms：

表 27-3 不同 PCLK 频率下的 TIMEOUTB 设置示例

f_{PCLK}	TIMEOUTB[11:0]	TEXTEN	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms
48 MHz	0xBB	1	188 x 2048 x 20.08 ns = 8 ms

将 t_{IDLE} 的最大持续时间配置为 50 µs

表 27-4 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）

f_{PCLK}	TIMEOUTB[11:0]	TIDLE	TIMEOUTEN	T_{TIDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 us
16 MHz	0xC7	1	1	64 x 4 x 62.5 ns = 50 us
48 MHz	0x257	1	1	600 x 4 x 20.08 ns = 50 us

表 27-5 SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
$t_{TIMEOUT}$	检测时钟低电平超时	25	35	ms
$t_{LOW:SEXT}$(1)	累积时钟低电平延长时间（从设备）	-	25	ms
$t_{LOW:MEXT}$(2)	累积时钟低电平延长时间（主设备）	-	10	ms
t_{IDLE}	SCL 和 SDA 高电平超时	-	50	us

Puya Confidential

1. $t_{LOW:SEXT}$ 是一段累积时间，即给定从设备在一条消息的最初起始到停止期间时钟信号可延展的时间。其他从设备或主设备也可能延长时钟，进而导致时钟低电平总延长时间超过 $t_{LOW:SEXT}$。因此，测量该参数时该器件应该是全速主设备寻址的唯一器件。
2. $t_{LOW:MEXT}$ 是一段累积时间，即主设备在消息的每个字节（定义为 START 到 ACK、ACK 到 ACK 或 ACK 到 STOP）内时钟信号可延展的时间。从设备或其他主设备也可能延长时钟，进而导致时钟低电平总时间超过 $t_{LOW:MEXT}$（针对给定字节）。因此，测量该参数时该全速主设备只寻址一个从设备。

27.4 I²C 中断

下表列出了所有的 I2C 中断请求

中断事件	事件标志	开启控制位
起始位已发送（Master）	SB	SBIE
地址已发送（Master）或地址匹配（Slave）	ADDR	ADDRIE
ADD10	10 位头段已发送（主）	ADD10IE
已收到停止（Slave）	STOPF	STOPIE
数据字节传输完成	BTF	BTFIE
接收缓冲区非空	RxNE	RXIE
发送缓冲区空	TxE	TXIE
总线错误	BERR	ITERREN
仲裁丢失（Master）	ARLO	
响应失败	AF	
过载/欠载	OVR	
超时/Tlow 错误	TIMEOUT	

27.5 I²C 寄存器描述

寄存器可以 half-word 或者 word 访问。

27.5.1 I²C 控制寄存器 1（I2C_CR1）

偏移地址：0x00

复位值：0x0008 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUP_CNT[1:0]		WKUP_DIV[1:0]	
												RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SWRST	Res.	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	ENARP	SMBTYPE	WUPEN	SMBUS	PE
RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:20	Res			
19:18	WKUP_CNT[1:0]	RW	2'h2	低功耗唤醒超时计数 00：2 01：8 10：32

Puya Confidential

				11：128 注：超时时间的计算公式为 (WKUP_DIV/FREQ)*WKUP_CNT μs 例：WKUP_DIV 选择 11 即 1024，FREQ 为 8M，CNT 选择 00 即 2，则超时时间为(1024/8)* 2 = 256 μs 注：PCLK 为 72M 时，timeout 时间为 3.5 μs-1817.6 μs PCLK 为 4M 时，timeout 时间为 64 μs - 32768 μs
17:16	WKUP_DIV[1:0]	RW	2'h0	PCLK 的分频系数，分频后的时钟用于计数 STOP 模式唤醒时的超时时间 00：128 01：256 10：512 11：1024
15	SWRST	RW	0	软件复位。 当被置位时，I²C 处于复位状态。在复位释放前，要确保 I²C 的引脚被释放，总线是空闲状态。 0：I²C 模块不处于复位状态 1：I²C 模块处于复位状态 注：该位可以用于错误或锁住状态时重新初始化 I²C。如 BUSY 位为 1，在总线上又没有检测到停止条件时。
14	Reserved	-	-	保留
13	ALERT	RW	0	SMBus 提醒。 0：释放 SMBAlert 引脚使其变高。提醒响应地址头紧跟在 NACK 信号后面； 1：驱动 SMBAlert 引脚使其变低。提醒响应地址头紧跟在 ACK 信号后面； PE=0 时，由硬件清除。
12	PEC	RW	0	数据包出错检查 软件可置位和清零该位，硬件可以在以下情况下清零该位：当传送 PEC 后，起始条件、或者停止条件、或者当 PE=0 时； 0：无 PEC 传输 1：PEC 传输（在发送或接收模式） 注：仲裁丢失时，PEC 的计算失效。
11	POS	RW	0	ACK/PEC 位置（用于数据接收），软件可置位/清零该寄存器，或 PE=0 时由硬件清零。 0：ACK 位控制当前移位寄存器内正在接收的字节的 (N)ACK。PEC 位表明当前移位寄存器内的字节是 PEC 1：ACK 位控制在移位寄存器里接收的下一个字节的 (N)ACK。PEC 位表明在移位寄存器里接收的下一个字节是 PEC 注：POS 位只能用在 2 字节的接收配置中，必须在接收数据之前配置。为了 NACK 第 2 个字节，必须在清除 ADDR 之后清除 ACK 位。为了检测第 2 个字节的 PEC，

Puya Confidential

				必须在配置了 POS 位之后，ADDR 延长事件时设置 PEC 位。
10	ACK	RW	0	应答使能。软件可置位/清零该寄存器，或 PE = 0 时由硬件清零。 0：无应答返回 1：在接收到一个字节后返回一个应答。（匹配地址或数据时）
9	STOP	RW	0	停止条件产生，软件可以置位/清零该寄存器，或者当检测到停止条件时，由硬件清除；当检测到超时错误时，硬件置位。 在主模式下： 0：无停止条件产生 1：在当前字节传输或在当前起始条件发出后产生停止条件 在从模式下： 0：无停止条件产生 1：在当前字节传输后释放 SCL 和 SDA 线
8	START	RW	0	起始条件产生。 软件可置位/清零该寄存器，或当起始条件发出后或 PE=0 时由硬件清零。 主模式： 0：无起始条件产生 1：重复产生起始条件 从模式： 0：无起始条件产生 1：当总线空闲时，产生起始条件（并由硬件自动切换到主机模式）
7	NOSTRETCH	RW	0	禁止时钟延长（从机）。 当 ADDR 或 BTF 标志被置位时，该位用于从机禁止时钟延长，直到被软件复位。 0：允许时钟延长 1：禁止时钟延长
6	ENGC	RW	0	广播呼叫使能。 0：禁止广播呼叫。以 NACK 响应地址 00h 1：允许广播呼叫。以 ACK 响应地址 00h
5	ENPEC	RW	0	PEC 使能。 0：禁止 PEC 计算 1：开启 PEC 计算
4	ENARP	RW	0	ARP 使能。 0：禁止 ARP； 1：使能 ARP； 如果 SMBTYPE=0，使用 SMBus 设备的默认地址； 如果 SMBTYPE=1，使用 SMBus 的主地址。
3	SMBTYPE	RW	0	SMBus 类型。 0：SMBus 设备；

Puya Confidential

				1：SMBus 主机；
2	WUPEN	RW	0	从模式唤醒 STOP 模式使能 0：禁止从模式唤醒 STOP 模式 1：使能从模式唤醒 STOP 模式
1	SMBUS	RW	0	SMBus 模式使能 0：I²C 模式； 1：SMBus 模式；
0	PE	RW	0	I²C 模块使能。 0：禁止 1：I²C 使能 注：如果清除该位时通讯正在进行，在当前通讯结束后，I²C 模块被禁用并返回空闲状态。 由于在通讯结束后 PE=0，所有的位被清除。 在主模式下，通讯结束之前，绝不能清除该位。

27.5.2 I²C 控制寄存器 2（I2C_CR2）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXIE	RXIE	Res.	STOPIE	ADD10IE	BTFIE	ADDRIE	SBIE
								RW	RW		RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	LAST	DMAEN	Res.	Res.	ITERREN	FREQ[7:0]							
			RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	-	保留
23	TXIE	RW	0	数据寄存器为空（发送时）标志中断使能。 0：禁止； 1：允许；
22	RXIE	RW	0	数据寄存器非空（接收时）标志中断使能。 0：禁止； 1：允许；
21	Reserved	-	-	保留
20	STOPIE	RW	0	停止条件检测位（从模式）中断使能。 0：禁止； 1：允许；
19	ADD10IE	RW	0	10 位地址头序列已发送（主模式）中断使能。 0：禁止； 1：允许；
18	BTFIE	RW	0	字节传输结束标志中断使能。 0：禁止； 1：允许；
17	ADDRIE	RW	0	地址已被发送（主模式）/地址匹配（从模式）中断使能。

Puya Confidential

				0：禁止； 1：允许；
16	SBIE	RW	0	起始位标志（主模式）中断使能。 0：禁止； 1：允许；
15:13	Reserved	-	-	保留
12	LAST	RW	0	DMA 最后一次传输。 0：下一次 DMA 的 EOT 不是最后的传输 1：下一次 DMA 的 EOT 是最后的传输 注：该位在主接收模式使用，使在最后一次接收数据时可以产生一个 NACK。
11	DMAEN	RW	0	DMA 请求使能 0：禁止 DMA 请求 1：使能 DMA 请求
10:9	Reserved			保留
8	ITERREN	RW	0	错误中断使能。 0：禁止出错中断； 1：允许出错中断； 在下列条件下，将产生该中断： ■ BERR=1 ■ ARLO=1 ■ AF=1 ■ OVR=1 ■ PECERR=1 ■ TIMEOUT=1； ■ SMBAlert=1.
7:0	FREQ[7:0]	RW	0	I^2C 模块时钟频率。 必须用 APB 时钟频率的值配置该寄存器，以产生与 I^2C 协议兼容的数据 setup 和 hold 时间。运行时钟为偶数时钟。最小允许可设定的频率分别是 100 k 模式下 4 MHz 及以上、400 k 模式下大于等于 8 MHz、1MHz 模式下大于等于 16 MHz。 0000000：禁止 0000001：禁止 ..... 0000011：禁止 0000100：4 MHz ….. …..（可配的最大频率详见 APB 的最大时钟频率） 注：1.主机模式下，116M 117M 禁止使用 2.从机模式下，频率大于等于 116M 时，FREQ 需要设置成 01110011。

27.5.3 I^2C 自身地址寄存器 1（I2C_OAR1）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

Puya Confidential

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDMODE	Res.	Res.	Res.	Res.	Res.	ADD[9:8]		ADD[7:1]							ADD0
RW						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	ADDMODE	RW	0	寻址模式（从模式）。 0：7 位从地址（不响应 10 位地址）； 1：10 位从地址（不响应 7 位地址）；
14:10	Reserved	-	-	保留
9:8	ADD[9:8]	RW	2'h0	接口地址。 7 位地址模式该寄存器无关。 10 位地址模式位地址的 9~8 位。
7:1	ADD[7:1]	RW	7'h0	接口地址的 7~1 位。
0	ADD0	RW	0	接口地址。 7 位地址模式时该寄存器无效。 10 位地址模式时为地址的 0 位。

27.5.4 I2C 自身地址寄存器 2（I2C_OAR2）

偏移地址：0x0C

复位值：0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	OA2MSK[2:0]			ADD2[7:1]							ENDUAL
					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:11	Reserved	-	-	保留
10:8	OA2MSK[2:0]	RW	3'h0	ADD2 屏蔽配置。 000：不屏蔽； 001：ADD2[1] 屏蔽，只比较 ADD2[7:2]； 010：ADD2[2:1] 屏蔽，只比较 ADD2[7:3]； 011：ADD2[3:1] 屏蔽，只比较 ADD2[7:4]； 100：ADD2[4:1] 屏蔽，只比较 ADD2[7:5]； 101：ADD2[5:1] 屏蔽，只比较 ADD2[7:6]； 110：ADD2[6:1] 屏蔽，只比较 ADD2[7]； 111：ADD2[7:1] 屏蔽，ACK 所有收到的 7bit 地址； 注：当 OA2MSK 不为 0 时，保留的 I^2C 地址（0b0000xxx 和 0b1111xxx）不会 ACK，即使地址比对正确。
7:1	ADD2[7:1]	RW	7'h0	接口地址的 7~1 位。 双地址模式下地址的 7~1 位。
0	ENDUAL	RW	0	双地址模式使能位。 0：在 7 位地址模式下，只有 OAR1 被识别； 1：在 7 位地址模式下，OAR1 和 OAR2 都被识别。

Puya Confidential

27.5.5 I²C 数据寄存器（I2C_DR）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7:0	DR[7:0]	RW	8'h0	8 位数据寄存器，芯片内部实际是两个独立的缓存共用一个地址，分别用于存放接收到的数据（RX_DR）、放置要发送到总线的数据（TX_DR）。 发送器模式： 当写一个字节至 DR 寄存器时（实际写到 TX_DR），自动启动数据传输。一旦传输开始（TxE=1），如果能及时把下一个需传输的数据写入 DR 寄存器，I2C 模块将保持连续的数据流。 接收器模式： 接收到的字节被拷贝到 DR 寄存器（实际是 RX_DR）（RxNE=1）。在接收到下一个字节（RxNE=1）之前读出数据寄存器，即可实现连续的数据接收。 注： 1) 在从机模式下，地址不会被拷贝进数据寄存器 2) 硬件不处理写冲突 (如果 TxE=0，仍能写入数据寄存器) 3) 如果在处理 ACK 脉冲时发生 ARLO 事件，接收到的字节不会被拷贝到数据寄存器里，因此不能被读到

27.5.6 I²C 状态寄存器 1（I2C_SR1）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMBALERT	TIMEOUT	Res.	PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Res.	STOPF	ADD10	BTF	ADDR	SB
RC_W0	RC_W0		RC_W0	RC_W0	RC_W0	RC_W0	RC_W0	R	R		R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	SMBALERT	RC_W0	0	SMBus 提醒状态。 主机模式： 0：无 SMBus 提醒； 1：在引脚产生 SMB Alert 提醒事件；

Puya Confidential

				从机模式： 0：没有 SMBAlert 响应地址头序列； 1：收到 SMBAlert 响应地址头序列直到 SMB Alert 变低。 该位由软件写 0 或 PE=0 时由硬件清除。
14	TIMEOUT	RC_W0	0	超时或 T_{low} 错误。 0：无超时错误； 1：SCL 为低的持续时间已达到 25 ms（超时）；或者主机低电平累计时钟扩展时间超过 10 ms（$T_{low:\ next}$）；或从设备低电平累积时钟扩展时间超过 25 ms（$T_{low:sext}$）；或者 T_{idle} 的时间超时；或者 I^2C 模式从 STOP 模式唤醒时可通过设置 I2C_CR1 设置 timeout 时间。 当在从模式下设置该位：从设备复位通讯，硬件释放总线； 当在主模式下设置该位：硬件发出停止条件； 该位由软件写 0 清除，或当 PE=0 时由硬件清除。 注：该功能仅在 SMBUS 模式和 I^2C 模式下从 STOP 模式唤醒过程下有作用。
13	Reserved	-	-	保留
12	PECERR	RC_W0	0	在接收时发生 PEC 错误。 0：无 PEC 错误，接收到 PEC 后返回 ACK（如果 ACK=1） 1：有 PEC 错误，接收到 PEC 后返回 NACK（不管 ACK 为何值） 该位由将软件写 0 清除，或当 PE=0 时由硬件清除。
11	OVR	RC_W0	0	过载/欠载标志。 0：无过载/欠载； 1：出现过载/欠载。 当 NOSTRETCH=1 时，在从模式下该位被硬件置位； 该位由软件写 0 清除，或当 PE=0 时由硬件清除。 注：如果数据寄存器的写操作发生时间非常接近 SCL 的上升沿，发送的数据是不确定的，并发生保持时间错误。
10	AF	RC_W0	0	应答失败标志。 0：没有应答失败； 1：应答失败。 当没有返回应答时，硬件将置位该寄存器。 该位由软件写 0 清除，或当 PE=0 时由硬件清除。
9	ARLO	RC_W0	0	仲裁丢失（主模式）。 0：没有检测到仲裁丢失； 1：检测到仲裁丢失。 当接口失去对总线的控制给另一个主机时，硬件将置位该寄存器。 该位由软件写 0 清除，或在 PE=0 时由硬件清除。

Puya Confidential

				在 ARLO 事件之后，I²C 接口自动切换回从模式（MSL=0）。 注：在 SMBUS 模式下，在从模式下的仲裁仅发生在数据解读，或应答传输区间（不包括地址的应答）。
8	BERR	RC_W0	0	总线出错标志。 0：无起始或者停止条件出错； 1：起始或者停止条件出错。 当接口检测到错误的起始或者停止条件,硬件将该位置 1。 该位由软件写 0 清除，或者在 PE=0 时由硬件清除。
7	TxE	R	0	数据寄存器为空（发送时）标志。 0：数据寄存器非空； 1：数据寄存器为空。 在发送数据时，数据寄存器为空时该位被置 1，在发送地址阶段不设置该位。 软件写数据到 DR 寄存器可清除该位，或在发生一个起始或停止条件后，或当 PE=0 时由硬件自动清除。 如果收到一个 NACK，或下一个要发送的字节是 PEC（PEC=1），该位不被置位。 注：在写入第 1 个要发送的数据后，或设置了 BTF 时写入数据，都不能清除 TxE 位，因为此时数据寄存器为空。
6	RxNE	R	0	数据寄存器非空（接收时）标志。 0：数据寄存器为空； 1：数据寄存器非空。 在接收时，当数据寄存器不为空，置位该寄存器。在接收地址阶段，该寄存器不置位。 软件对数据寄存器的读写操作会清除该寄存器，或当 PE=0 时由硬件清除。 注：当设置了 BTF 时，读取数据不能清除 RxNE 位，因为此时数据寄存器仍为满。
5	Reserved	-	-	保留
4	STOPF	R	0	停止条件检测位（从模式）。 0：没有检测到停止位； 1：检测到停止条件。 在一个应答之后（如果 ACK=1），当从设备在总线上检测到停止条件时，硬件将该位置 1。 将软件读取 I2C_SR1 寄存器后，对 I2C_CR1 寄存器的写操作将清除该位，或当 PE=0 时，硬件清除该位。 注：在收到 NACK 后，STOPF 位不会置位。
3	ADD10	R	0	10 位地址头序列已发送（主模式）。 0：没有 ADD10 事件发生。 1：主设备已经将第一个地址字节发送出去。 在 10 位地址模式下，当主设备将第一个字节发送出去时，硬件将该位置 1. 软件读取 I2C_SR1 寄存器后，对 I2C_DR 寄存器的写操作将清除该位，或当 PE=0 时，硬件清除该位。

Puya Confidential

				主：收到 NACK 后，该寄存器不被置位。
2	BTF	R	0	字节传输发送结束标志位。 0：字节传输发送未完成 1：字节传输发送成功结束 当 NOSTRETCH=0 时，在下列情况下硬件将置位该寄存器（从机模式，NOSTRETCH=0 时；主机模式，与 NOSTRETCH 无关）： — 接收时，当收到一个新字节（包括 ACK 脉冲）且数据寄存器还未被读取（RxNE=1）。 — 发送时，当一个新数据应该被发送，且数据寄存器还未被写入新的数据（TxE=1）。 对数据寄存器的读或写操作将清除该位; 或发送一个起始或停止条件后，或当 PE=0 时，由硬件清除。 注： 在收到一个 NACK 后，BTF 位不会被置位。 如果下一个要传输的字节是 PEC（I2C_SR1.TRA=1,I2C_CR1.PEC=1）,BTF 位不会被置位。
1	ADDR	R	0	地址已被发送（主模式）/地址匹配（从模式）。 读 I2C_SR2 寄存器后，再读 I2C_SR1 寄存器将清除该位；或在传输中发送一个起始或停止条件后，或当 PE=0 时，由硬件清除。 **地址匹配（从机）：** 0：地址不匹配或没有收到地址； 1：收到的地址匹配。 当收到的从地址与 OAR 寄存器或广播呼叫地址，或 SMBus 设备默认地址、或 SMBus 主机识别出 SMBus 提醒地址匹配时，硬件将置位该位。 **注：**在从机模式下，推荐进行完整的清零序列，即在 ADDR 被置位后，先读 SR1 寄存器，再读 SR2 寄存器。 **地址已发送（主机）：** 0：地址发送没有结束； 1：地址发送结束。 — 10 位地址时，当收到地址的第二个字节的 ACK 后置位； — 7 位地址时，当收到 ACK byte 后置位。 **注：**在收到 NACK 后，该寄存器不会被置位。
0	SB	R	0	起始位标志（主模式）。 0：未发送起始条件； 1：起始条件已发送； —当发送起始条件时，置位该寄存器。 —软件读取 I2C_SR1 寄存器后，对数据寄存器的读或写操作将清除该位；或当 PE=0 时，由硬件清除。

Puya Confidential

27.5.7 I²C 状态寄存器 2（I2C_SR2）

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SBUSY
															R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	SMBHOST	SMBDEFAULT	GENCALL	TRIPLEF	TRA	BUSY	MSL
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:17	Reserved	-	-	保留
16	SBUSY	R	0	从机工作标志 0：不处于从机工作状态 1：处于从机工作状态 当 MSL 等于 0 时检测到一个开始条件时硬件置位 当地址不匹配或者总线上检测到一个停止条件（STOPF=1）时硬件清零
15:8	PEC[7:0]	R	8'h0	数据包出错检测寄存器。 当 ENPEC=1 时，该寄存器存放内部的 PEC 的值。
7	DUALF	R	0	双地址标志（从模式）. 0：接收到的地址与 OAR1 匹配; 1：接收到的地址与 OAR2 匹配。 当产生一个停止条件或一个重复的起始条件时，或 PE=0 时，硬件清除该寄存器。
6	SMBHOST	R	0	接收到 SMBus 主机头序列（从模式）标志 0：未收到 SMBus 主机的地址; 1：当 SMBTYPE=1 且 ENARP=1 时，收到 SMBus 主机地址。 当产生一个停止条件或一个重复的起始条件时，或 PE=0 时，硬件清除该寄存器。
5	SMBDEFAULT	R	0	SMBus 从设备默认地址（从模式） 0：未收到 SMBus 设备的默认地址; 1：当 ENARP=1 时，收到 SMBus 设备的默认地址。 当产生一个停止条件或一个重复的起始条件时，或 PE=0 时，硬件清除该寄存器。
4	GENCALL	R	0	广播呼叫地址（从模式） 0：未收到广播呼叫地址; 1：当 ENGC=1 时，收到广播呼叫的地址。 当产生一个停止条件或一个重复的起始条件时，或 PE=0 时，硬件清除该寄存器。
3	TRIPLEF	R	0	三地址标志（从模式）. 0：接收到的地址与 OAR3 不匹配; 1：接收到的地址与 OAR3 匹配。

Puya Confidential

				当产生一个停止条件或一个重复的起始条件时，或 PE=0 时，硬件清除该寄存器。 注：该寄存器如果不支持三地址功能，则固定为 0.
2	TRA	R	0	发送/接收标志 0：接收到数据 1：数据已发送 在整个地址传输阶段的结尾，该寄存器根据地址字节的 RW 位来设定。 当检测到停止条件（STOPF=1），或者重复的起始条件、或者总线仲裁丢失（ARLO=1），或当 PE=0 时，硬件清除该寄存器。
1	BUSY	R	0	总线忙标志 0：在总线上无数据通讯 1：在总线上正在进行数据通讯 当检测到 SDA 或 SCL 为低电平时，硬件置位。 当检测到一个停止条件时，硬件清零。 该寄存器指示当前正在进行的总线通讯，当接口被禁用（PE=0）时该信息仍然被更新。
0	MSL	R	0	主从模式标志 0：从机模式 1：主机模式 —当接口处于主模式（SB=1）时，硬件置位； —当总线上检测到一个停止条件（STOPF=1）、仲裁丢失（ARLO=1）、或当 PE=0 时，硬件清零。

27.5.8 I²C 时钟控制寄存器（I2C_CCR）

偏移地址：0x1c

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F/S	DUTY	F+	Res.	CCR[11:0]											
RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	F/S	RW	0	I²C 主模式选择。 0：标准模式 1：快速模式
14	DUTY	RW	0	快速模式时的占空比。 0：快速模式下：$T_{low}/T_{high}=2$ 1：快速模式下：$T_{low}/T_{high}=16/9$
13	F+	RW	0	I²C 快速增强模式选择。 0：标准模式或快速模式，具体选择哪种由 bit15 决定； 1：快速增强模式；
12	Reserved	-	-	保留

Puya Confidential

11:0	CCR[11:0]	RW	12'h0	快速/标准模式下的时钟控制分频系数（主模式）。 该分频系数用于设置主模式下的 SCL 时钟。 ● 标准模式或者 SMBus 模式： T_{high}=CCR x Tpclk T_{low} =CCR x Tpclk ● 快速模式： ✓ DUTY=0: T_{high}=CCR x Tpclk T_{low} =2 x CCR x Tpclk ✓ DUTY=1(为达到 400KHz): T_{high}=9 x CCR x Tpclk T_{low} =16 x CCR x Tpclk ● 快速增强版模式： ✓ DUTY=0: T_{high} =3 x CCR x Tpclk T_{low} =5 x CCR x Tpclk ✓ DUTY=1: T_{high} =2 x CCR x Tpclk T_{low} =3x CCR x Tpclk 注： 1. 允许设定的最小值为 0x04，在快速模式下允许的最小值为 0x01 2. $T_{high}=t_{r(SCL)}+t_{w(SCLH)}$ 3. $T_{low}=t_{r(SCL)}+t_{w(SCLL)}$ 4. 这些延时没有过滤器 5. 只有当 PE=0 时才能配置该寄存器；

27.5.9 I²C 上升时间寄存器（I2C_TRISE）

偏移地址：0x20

复位值：0x0000 0082

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	THOLDDATA_SEL	THOLDDATA[5:0]						TRISE[7:0]							
	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:15	Reserved	-	-	保留
14	THOLDDATA_SEL	RW	0	数据保持时间选择 0：默认硬件计算数据保持时间 1：通过 THLDDATA 配置数据保持时间
13:8	THOLDDATA	RW	5'h1	在快速/标准/快速增强模式下的最大数据保持时间（发送模式） 这些位是用于数据发送模式下，保证数据保持时间的最小时间。 例如：标准模式允许的最大 SDA 下降时间为 300ns。如果 I2C_CR2.FREQ[6:0]的值等于 0x08，T_{pclk} =125ns，则 TRISE 中配置为 0x03 （300ns/125ns = 2.4 +1 =3.4） 如果结果不为整数，则将整数部分写入 THOLDDATA，以确保配置。

Puya Confidential

7:0	TRISE	RW	8'h2	在快速/标准模式下的最大上升时间（主模式）。 这些位应该提供在主机模式下， SCL 反馈回路的最大持续时间。这样做的目的是无论 SCL 上升沿持续时间多少，SCL 都能保持一个稳定的频率。 这些位必须设置为 I^2C 总线规范里给出的最大的 SCL 上升时间，增长步幅为 1。 例如：标准模式中最大允许 SCL 上升时间为 1000ns。如果 I2C_CR2.FREQ[5:0]的值等于 0x08，T_{pclk}=125ns，则 TRISE 中配置为 0x09（1000ns/125ns = 8 + 1 = 9）。 如果结果不为整数，则将整数部分写入 TRISE，以确保 t_{HIGH} 参数。 注：当 PE=0 时才能设置该寄存器。

27.5.10 I^2C 超时寄存器（I2C_TIMEOUTR）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]											
RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31	TEXTEN	RW	0	时钟信号延展超时使能 0：禁止时钟信号延展超时检测。 1：使能时钟信号延展超时检测 当 I^2C 接口执行 SCL 延展的累积时间超过 $t_{LOW:EXT}$ 时，将检测到超时错误（TIMEOUT = 1)。
30:28	Reserved	-	-	保留
27:16	TIMEOUTB[11:0]	RW	12'h0	总线超时 B 该字段用于配置累积时钟延展超时： 在主模式下，将检测主设备的累积时钟低电平延展时间($t_{LOW:MEXT}$) 在从模式下，将检测从设备的累积时钟低电平延展时间($t_{LOW:SEXT}$) $t_{LOW:EXT}$ = (TIMEOUTB+1) x 2048 x t_{I2CCLK} 注： 仅可在 TEXTEN=0 时写入这些位
15	TIMEOUTEN	RW	0	时钟超时使能 0：禁止 SCL 超时检测。 1：使能 SCL 超时检测：当 SCL 的低电平时间超过 $t_{TIMEOUT}$ (TIDLE=0)，或 SCL 的高电平 时间超过 t_{IDLE}(TIDLE=1) 时，将检测到超时错误 (TIMEOUT=1)。
14:13	Reserved	-	-	保留
12	TIDLE	RW	0	空闲时钟超时检测 0：TIMEOUTA 用于检测 SCL 低电平超时 1：TIMEOUTA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件） 注：仅可在 TIMOUTEN=0 时写入该位。

Puya Confidential

11: 0	TIMEOUTA[11:0]	RW	12'h0	总线超时 A 该字段用于配置： – SCL 低电平超时条件 $t_{TIMEOUT}$（当 TIDLE=0 时） $t_{TIMEOUT}$= (TIMEOUTA+1) x 2048 x t_{I2CCLK} –总线空闲条件，即 SCL 和 SDA 高电平（当 TIDLE=1 时） t_{IDLE}= (TIMEOUTA+1) x 4 x t_{I2CCLK} 注：仅可在 TIMOUTEN=0 时写入这些位。

27.5.11 I2C 自身地址寄存器 3（I2C_OAR3）

注：该寄存器如果不支持三字节地址功能，则固定为 0.

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res	Res	Res	ADD3[7:1]							ENTRIPLE
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7:1	ADD3[7:1]	RW	0	接口地址的 7~1 位。 三地址模式下地址的 7~1 位。
0	ENTRIPLE	RW	0	三地址模式使能位。 0：在 7 位地址模式下，只有 OAR1 被识别； 1：在 7 位地址模式下，OAR1 和 OAR3 都被识别。

Puya Confidential

28. 通用同步收发器(USART)

28.1 USART 简介

通用同步异步收发器（USART）能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 利用分数波特率发生器提供宽范围的波特率选择。

USART 不仅支持同步单向通信和半双工单线通信，以及 LIN（局域互连网络）、智能卡协议、IrDA（红外线数据协会）、SIR ENDEC 规范和调制解调器操作（CTS/RTS），还支持多处理器通信。

使用多缓冲器配置的 DMA 方式，可以实现高速数据通信。

28.2 USART 主要特征

- 全双工异步通信
- NRZ 标准格式
- 可配置 16 或 8 的过采样方法，以实现速度和时钟容忍度之间的最佳选择
- 发送和接收共用可编程波特率
- 两个用于收发数据的内部 FIFO，每个 FIFO 均可由软件使能/禁止，并且均带有一个状态标志位
- 双时钟域，除了 PCLK 以外，还存在的一个 USART_KER_CK 时钟
- 自动波特率检测功能
- 可编程的数据长度（7 位，8 位或者 9 位）
- 可编程的数据传输方向：MSB 或 LSB
- 可配置的停止位（0.5/1/1.5/2 个停止位）
- 支持同步模式和为同步通讯提供时钟输出
- 单线半双工通信
- 通过 DMA 缓冲接收/发送字节
- 独立的发送器和接收器使能位
- 可配置的 TX/RX 引脚交换功能
- 硬件流控：RS232 和 RS485
- 通信控制和错误检测标志
- 奇偶校验控制
  - 发送校验位
  - 对接收数据进行校验
- 多处理器通信。如果地址不匹配，则进入静默模式
  - 静默模式中唤醒（通过空闲总线检测或地址标志检测）
  - 唤醒接收器的方式：地址位，总线空闲
- LIN 主机发送同步断开帧的能力以及 LIN 从机检测断开帧的能力
  - 当 USART 硬件配置成 LIN 时，生成 13 位断开符，检测 10/11 断开符
- IRDA SIR 编码器解码器
  - 在正常模式下支持 3/16 位的持续时间

Puya Confidential

- 智能卡模拟功能
  - 智能卡接口支持 ISO7816-3 标准里定义的异步智能卡协议
  - 智能卡用到 0.5 和 1.5 个停止位
  - NACK 信号宽度可配置为 1/1.5/2 ETU
  - 发送数据支持传输错误重发机制
  - 支持 EGT 可设 0~256，并支持超时中断
- MODBUS 通信
  - TIMOUT 功能
  - CR/LF 字符识别功能

28.3 USART 功能描述

28.3.1 USART 功能框图

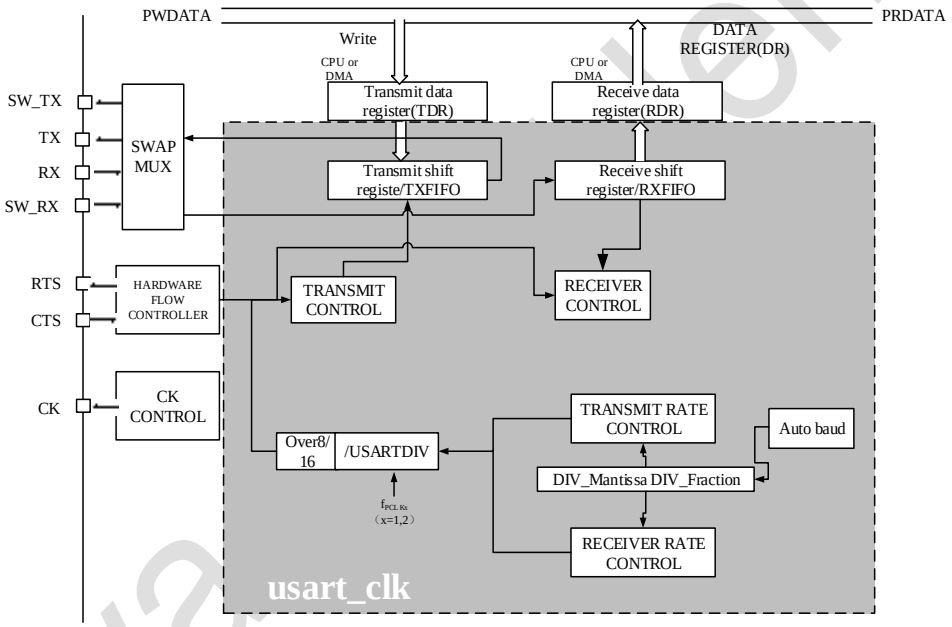


图 28-1 USART 框图

28.3.2 USART 输入/输出引脚说明

USART 接口通过三个引脚与其他设备连接在一起。任何 USART 双向通信至少需要两个脚：接收数据输入（RX）和发送数据输出（TX）。

RX：接收数据串行输入。通过过采样技术来区别数据和噪音，从而恢复数据。

TX：发送数据串行输出。当发送器被禁止时，输出引脚恢复到它的 I/O 端口配置。当发送器被激活，并且不发送数据时，TX 引脚处于高电平。在单线和智能卡模式里，此 I/O 口被同时用于数据的发送和接收。

在 RS232 硬件流控制模式中需要以下引脚：

CTS（clear to send）： 清除发送，若是高电平，在当前数据传输结束时阻断下一次的数据发送。

RTS（request to send）： 发送请求，若是低电平，表明 USART 准备好接收数据。

在 RS485 硬件控制模式下需要以下引脚：

Puya Confidential

DE（Driver Enable）：驱动使能，这个信号用控制外部接收者的传输

在同步主模式和智能卡模式下需要以下引脚：

CK：该引脚在同步主模式和智能卡模式下用作时钟输出。

在同步主模式下，该引脚用于输出发送器数据时钟，以便按照 SPI 主器件模式进行同步发送（起始位和结束位上无时钟脉冲，可通过软件向最后一个数据位发送时钟脉冲）。同时，RX 引脚可同步接收数据。该机制可用于控制带移位寄存器的外设（例如 LCD 驱动器）。时钟相位和极性可通过软件编程。

在智能卡模式下，CK 输出向智能卡提供时钟。

USART 输入/输出引脚的列表如下表所示。

表 28-1 USART 输入/输出引脚

引脚名称	信号类型	说明
USART_RX	输入	串行数据接收输入
USART_TX	输出	发送数据输出引脚
USART_CTS	输入	清除以发送
USART_RTS	输出	请求以发送
USART_DE	输出	驱动器使能
USART_CK	输出	同步主模式和智能卡模式下的时钟输出

注：USART_DE 和 USART_RTS 共用同一个引脚。

28.3.3 USART 时钟

USART 有两个完全独立的时钟域：

- usart_pclk 时钟域

usart_pclk 时钟信号为外设总线接口提供时钟。需要访问 USART 寄存器时，该信号必须有效。

- usart_ker_ck 内核时钟域

usart_ker_ck 是 USART 时钟源。它独立于 usart_pclk，由 RCC 提供。

注意：访问 USART 寄存器时，usart_ker_ck 也是需要使能的。

不支持双时钟域功能时，usart_ker_ck 时钟与 usart_pclk 时钟相同。

usart_pclk 和 usart_ker_ck 之间无任何约束：usart_ker_ck 可快于/慢于 usart_pclk。唯一的限制就是软件以足够快的速度管理通信的能力。

28.3.4 USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位（M0：位 12，M1：14 位）进行编程来将字长设置为 7 位、8 位或 9 位。

- 7 位字符长度：M[1:0] = 10
- 8 位字符长度：M[1:0] = 00
- 9 位字符长度：M[1:0] = 01

注意：在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测），

在 9 位数据长度模式下，不支持自动波特率（0x7F 帧和 0x55 帧检测）。

在默认情况下，信号(TX 或 RX)在起始位工作期间处于低电平状态，在停止位工作期间处于高电平。

通过极性配置控制，可以单独针对每个信号对这些值取反。

Puya Confidential

IDLE character（空闲帧）可理解为整个帧周期内电平均为“1”(停止位的电平也是“1”)，该字符后是下一个数据帧的起始位。

Break character（断开帧）可理解为在一个帧周期内接收到的电平均为“0”。发送器在断开帧的末尾插入 1 或 2 个停止位(逻辑“1”位)以确认起始位。

发送和接收操作由一共用的波特率发生器驱动，当发送器和接收器的使能位分别置位时，分别为其产生时钟。

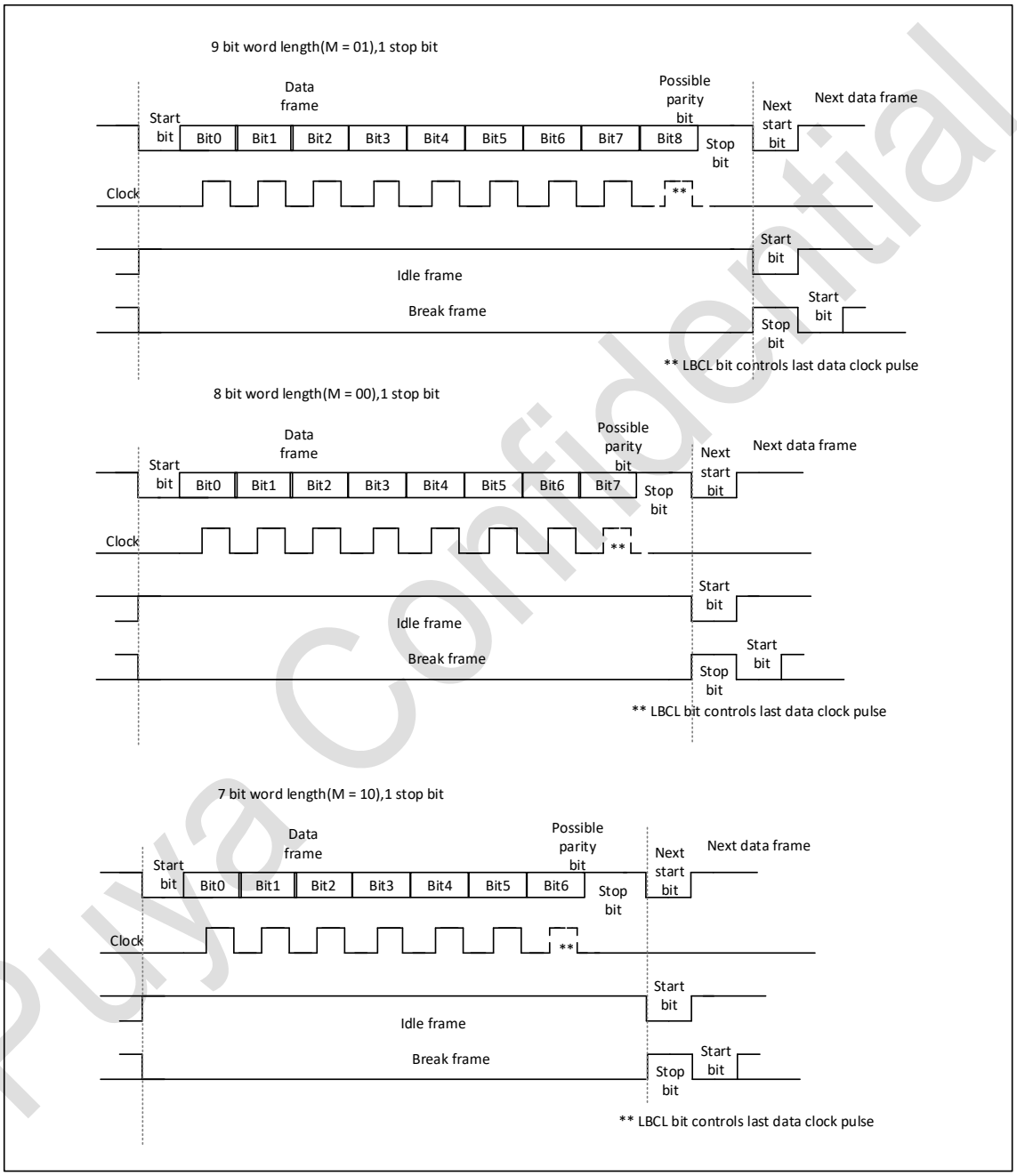


图 28-2 字长编程

Puya Confidential

28.3.5 USART FIFO 和阈值

USART 可工作在 FIFO 模式下。

USART 具有一个发送 FIFO（TXFIFO）和一个接收 FIFO（RXFIFO）。可通过将 USART_CR1 寄存器中的 FIFOEN（位 31）置 1 使能 FIFO 模式。仅 UART 和 SPI 模式下支持该功能。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注意：1、接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。

状态标志位于 USART_SR 寄存器中。

2、在使用 FIFO 功能前需要先置位 USART_CR1.TXFRQ 和 USART_CR1.RXFRQ,以此确保 FIFO 中没有数据。

可以配置触发 TX 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 USART_CR1 控制寄存器中的 RXFTCFG 和 TXFTCFG 位域进行编程。

在这种情况下：

- 当 RXFIFO 中接收到的数据量达到 RXFTCFG 中编程的阈值时，会生成 RX 中断。此时，USART_SR 寄存器中的 RXFT 标志置 1。这意味着已接收到如下数据：RDR 中有 1 个数据，RXFIFO 中有(RXFTCFG - 1)个数据。例如，如果将 RXFTCFG 编程为 100，则在接收到对应于 FIFO 大小的数据量(RXFIFO 中有(配置 FIFO 大小 – 1)个数据，USART_RDR 中有 1 个数据)时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中的空位置数达到在 TXFTCFG 中编程的阈值时，会生成 TX 中断。此时，USART_SR 寄存器中的 TXFT 标志置 1。

28.3.6 USART 发送器

发送器根据 USART_CR1.M 位的状态发送 7 位、8 位或 9 位的数据。当发送使能位（TE）被设置时，发送移位寄存器中的数据在 TX 脚上输出，相应的时钟脉冲在 CK 脚上输出。

28.3.6.1 字符发送

在 USART 发送期间，在 TX 引脚上首先移出数据的最低有效位。在此模式下，USART_DR 寄存器的缓冲区（TDR）位于内部总线和发送移位寄存器之间。

使能 FIFO 模式时，写入到发送数据寄存器(USART_DR)中的数据会在 TXFIFO 中排队。

每个字符之前都有一个低电平的起始位；之后跟着的停止位，其数目可配置。

USART 支持多种停止位的配置：0.5、1、1.5 或者 2 个停止位。

注意：1、在数据传输期间不能复位 TE 位，否则将破坏 TX 脚上的数据，因为波特率计数器停止计数。正在传输的当前数据将丢失。

2、TE 位被激活后将发送一个空闲帧。

28.3.6.2 配置停止位

随每个字符发送的停止位的位数可以通过 USART_CR2 的位 13、12 进行编程。

- 1 个停止位：停止位位数的默认值。

Puya Confidential

- 2 个停止位：可用于常规 USART 模式、单线模式以及调制解调器模式。
- 0.5 个停止位：在智能卡模式下接收数据时使用。
- 1.5 个停止位：在智能卡模式下发送和接收数据时使用。

空闲帧包括了停止位。

断开帧是 10 位低电平(M[1:0] = 00 时)，或者 11 位低电平(M[1:0] = 01 时)，或者 9 位低电平(M[1:0] = 10 时)；后面都是已配置数量的停止位。不可能传输更长的断开帧(长度大于 9/10/11 位低电平)。

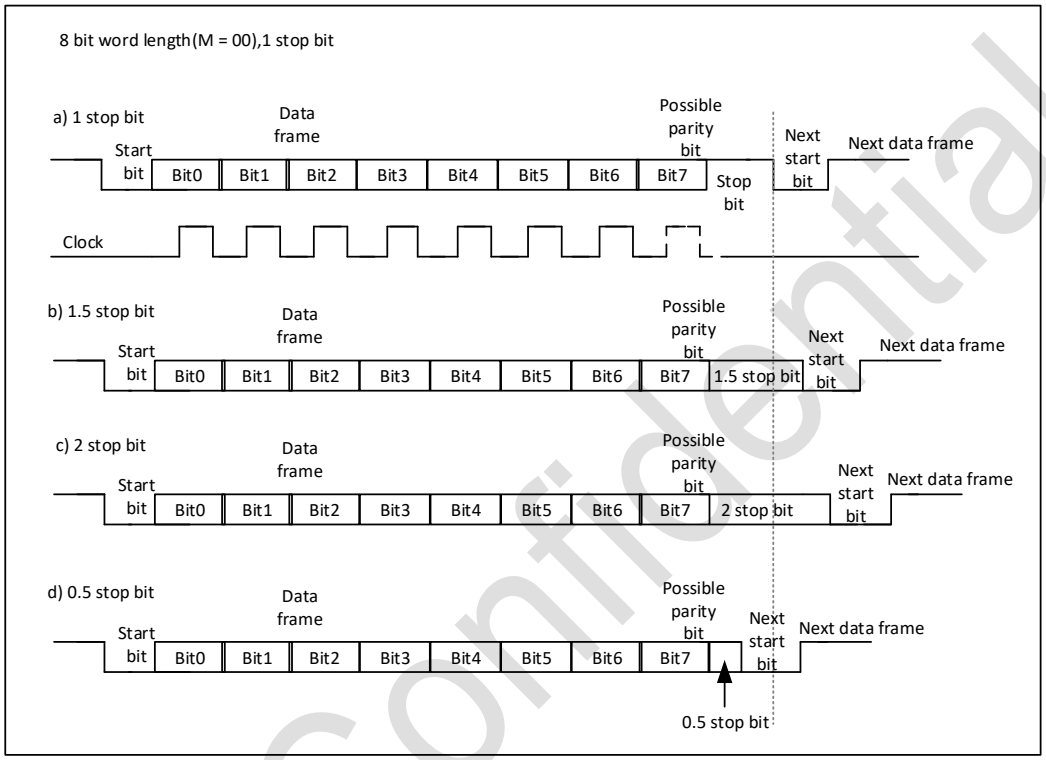


图 28-3 配置停止位

配置步骤：

1) 通过置位 USART_CR1 寄存器的 UE 位来激活 USART。
2) 编程 USART_CR1 的 M 位来定义字长。
3) 在 USART_CR2 中编程停止位的位数。
4) 如果采用多缓冲器通信，配置 USART_CR3 中的 DMA 使能位（DMAT）。按多缓冲器通信中的描述配置 DMA 寄存器。
5) 利用 USART_BRR 寄存器选择要求的波特率。
6) 设置 USART_CR1 中的 TE 位，发送一个空闲帧作为第一次数据发送。
7) 把要发送的数据写进 USART_DR 寄存器(此动作清除 TXE 位)。在只有一个缓冲器的情况下，对每个待发送的数据重复步骤 7。

- 禁止 FIFO 模式时，向 USART_DR 写入数据会将 TXE 标志清零。
- 使能 FIFO 模式时，向 USART_DR 写入数据会为 TXFIFO 中增添一个数据。当 TXFNF 标志置 1 时，对 USART_DR 寄存器的写操作可以执行。该标志会保持置 1，直到 TXFIFO 已满。

8) 在 USART_DR 寄存器中写入最后一个数据字后，要等待 TC=1。

- 禁止 FIFO 模式时，这表示最后一个帧的发送已经完成。
- 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

Puya Confidential

当需要关闭 USART 或需要进入停机模式之前，需要确认传输结束，避免破坏最后一次传输。

28.3.6.3 单字节通讯

■ 禁止 FIFO 模式时

对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1。它表明：

- 数据已从 USART_DR 寄存器传到移位寄存器中且数据发送已开始；
- USART_DR 寄存器为空；
- USART_DR 寄存器中可写入下一个数据，而且不会覆盖前一个数据。

TXEIE 位置 1 时，该标志位会发生中断。

如果此时 USART 正在发送数据，对 USART_DR 寄存器的写操作把数据存进内部的 TDR 寄存器，并在当前传输结束时把该数据复制进移位寄存器。

如果此时 USART 没有在发送数据，处于空闲状态，对 USART_DR 寄存器的写操作直接把数据放进移位寄存器，数据传输开始时，TXE 位立即被置起。

■ 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：

- TXFIFO 未满；
- USART_DR 寄存器为空；
- USART_DR 寄存器中可写入一个数据，而不会覆盖前一个数据。发送时，写 USART_DR 寄存器，会将数据存储在 TXFIFO。该数据随后会在当前发送结束时从 TXFIFO 复制到移位寄存器中。

TXFIFO 未满时，即使在对 USART_DR 寄存器执行完写操作后，TXFNF 标志也保持为 1。该标志在 TXFIFO 已满时清零。TXFNFIE 位置 1 时该标志位会生成中断。

或者当达到 TXFIFO 阈值时，会生成中断并将数据写入 FIFO。这种情况下，CPU 可写入由编程的触发阈值定义的数据块。

当一帧发送完成时（停止位发送后）并且设置了 TXE 位，TC 位被置起，如果 USART_CR1 寄存器中的 TCIE 位被置起时，则会产生中断。

在 USART_DR 寄存器中写入了最后一个数据字后，在关闭 USART 模块之前或设置微控制器进入低功耗模式(详见下图)之前，必须先等待 TC=1。

Puya Confidential

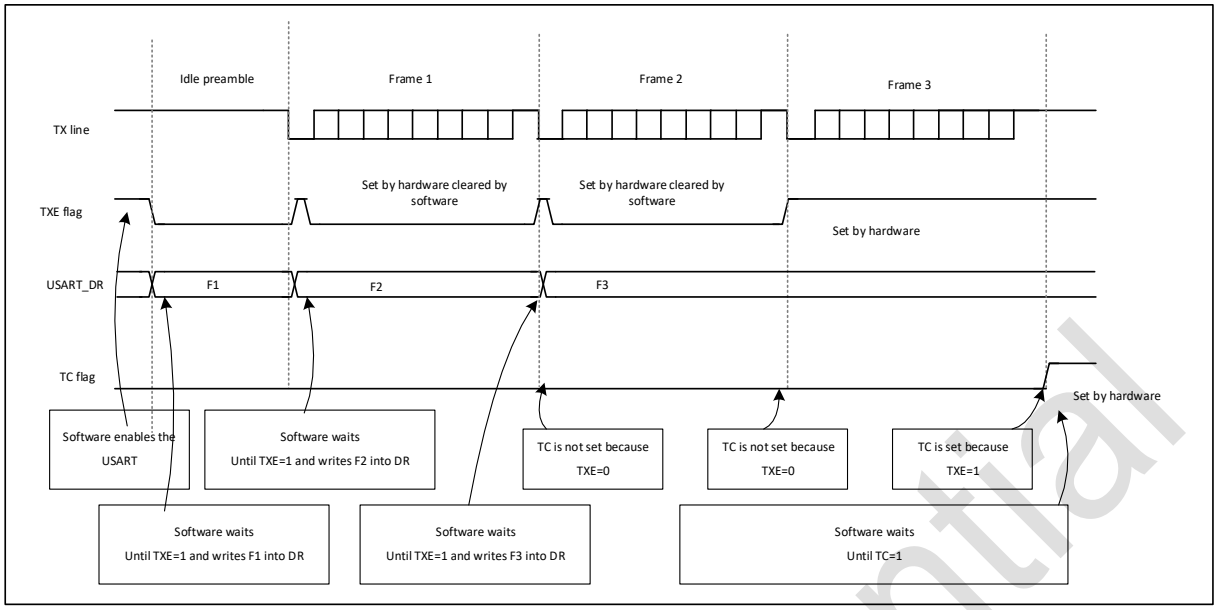


图 28-4 传输期间 TC/TXE 的行为

注意：使能 FIFO 管理时，TXFNF 标志将用于数据发送。

28.3.6.4 断开字符

设置 SBK 可发送一个断开字符。断开帧长度取决 M 位。如果设置 SBK=1，在完成当前数据发送后，将在 TX 线上发送一个断开字符。断开字符发送完成时(发送断开字符后的停止位期间)SBK 被硬件复位。USART 在最后一个断开帧的结束处插入一逻辑'1'，以保证能识别下一帧的起始位。

如果 SBK 位置 1，则在当前发送结束时，会发送一个断开字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送断开字符的优先级也仍高于发送数据的优先级。

注意：如果在开始发送断开帧之前，软件又复位了 SBK 位，断开符号将不被发送。如果要发送两个连续的断开帧，SBK 位应该在前一个断开符号的停止位之后置起。

28.3.6.5 空闲字符

置位 TE 将使得 USART 在第一个数据帧前发送一空闲帧。

28.3.7 USAET 接收器

USART 可以根据 USART_CR1 的 M 位接收 7 位、8 位或 9 位的数据字。

28.3.7.1 起始位检测

在 USART 中，如果辨认出一个特殊的采样序列，那么就认为侦测到一个起始位。该序列为：1 1 1 0 X 0 X 0 X 0 0 0 0

Puya Confidential

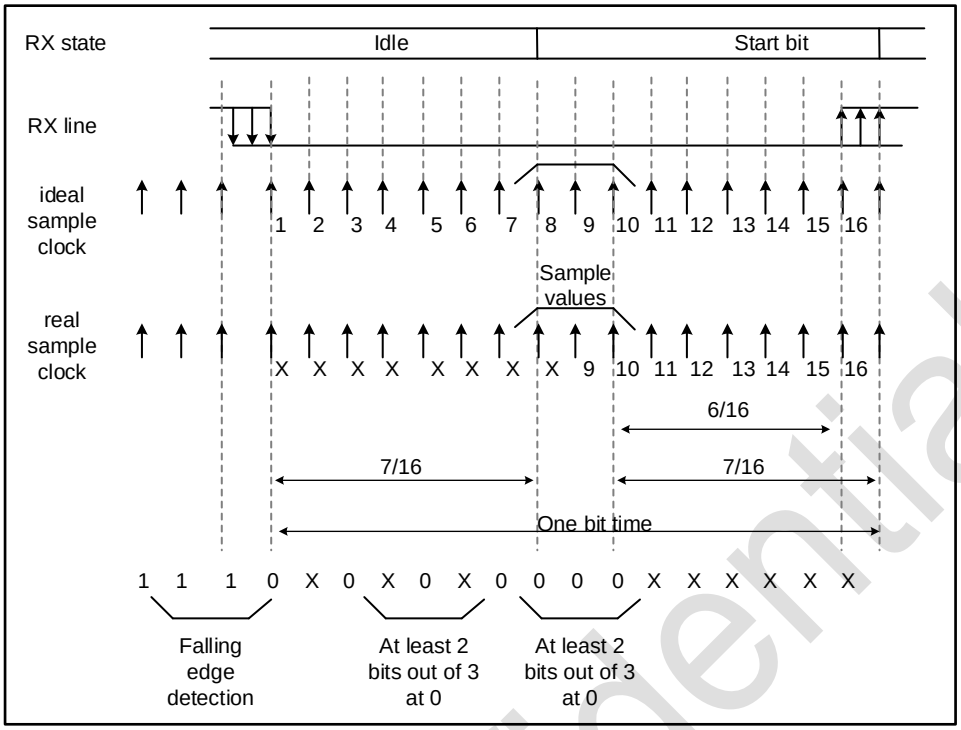


图 28-5 起始位的检测

如果该序列不完整，那么接收端将退出起始位侦测并回到空闲状态（不设置标志位）等待下降沿。如果 3 个采样点都为’0’（在第 3、5、7 位的第一次采样，和在第 8、9、10 的第二次采样都为’0’），则确认收到起始位，这时设置 RXNE 标志位，如果 RXNEIE=1，则产生中断。

如果两次 3 个采样点上仅有 2 个是’0’（第 3、5、7 位的采样点和第 8、9、10 位的采样点），那么起始位仍然是有效的，但是会设置 NE 噪声标志位。如果不能满足这个条件，则中止起始位的侦测过程，接收器会回到空闲状态（不设置标志位）。

如果有一次 3 个采样点上仅有 2 个是’0’（第 3、5、7 位的采样点或第 8、9、10 位的采样点），那么起始位仍然是有效的，但是会设置 NE 噪声标志位。

28.3.7.2 字符接收

在 USART 接收期间，数据的最低有效位首先从 RX 脚移进。在此模式里，USART_DR 寄存器包含的缓冲器位于内部总线和接收移位寄存器之间。

配置步骤：

1. 将 USART_CR1 寄存器的 UE 置 1 来激活 USART。
2. 编程 USART_CR1 的 M 位定义字长
3. 在 USART_CR2 中编写停止位的个数
4. 如果需多缓冲器通信，选择 USART_CR3 中的 DMA 使能位(DMAR)。按多缓冲器通信所要求的配置 DMA 寄存器。
5. 利用波特率寄存器 USART_BRR 选择希望的波特率。
6. 设置 USART_CR1 的 RE 位。激活接收器，使它开始寻找起始位。

PuyaConfidential

当一字符被接收到时：

- 如果已禁止 FIFO 模式，则 RXNE 位置 1。它表明移位寄存器的内容被转移到 RDR。换句话说，数据已经被接收并且可以被读出（包括与之有关的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表明 RXFIFO 非空。读取 USART_DR 寄存器会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（使能 FIFO 模式时为 RXFNEIE）位被设置，则产生中断。
- 在接收期间如果检测到帧错误、噪音错误、奇偶校验错误或上溢错误，错误标志将被置 1。
- 在多缓冲器通信模式下：
  - 如果禁止 FIFO 模式，则 RXNE 标志会在每次接收到字节后置 1。该标志在 DMA 读取数据寄存器时被清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1.每次收到 DMA 请求后，都会从 RXFIFO 取回数据。当 RXFIFO 非空时(即当存在要从 RXFIFO 中读取的数据时)，会触发 DMA 请求。
- 在单缓冲器模式下：
  - 如果禁止 FIFO 模式，则通过软件对 USART_DR 寄存器进行读操作来将 RXNE 标志位清零。RXNE 位必须在下一字符接收结束前被清零，以避免上溢错误。
  - 如果使能 FIFO 模式，则 RXFNE 在 RXFIFO 非空时置 1。每次对 USART_DR 执行完读操作后，都会从 RXFIFO 中取回数据。RXFIFO 为空时，RXFNE 标志将清零。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

注意： 在接收数据时，RE 位不应该被复位。如果 RE 位在接收时被清零，当前字节的接收被丢失。

28.3.7.3 断开字符

当接收到一个断开字符时，USART 像处理帧错误一样处理它。

28.3.7.4 空闲字符

当一空闲帧被检测到时，其处理步骤和接收到普通数据帧一样，但如果 IDLEIE 位被设置将产生一个中断。

28.3.7.5 上溢错误

- 禁止 FIFO 模式

如果 RXNE 未复位时接收到字符，则会发生上溢错误。

RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志都将置 1。

当 RXNE 标志位是 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求，则会发生上溢错误。发生上溢错误时：

Puya Confidential

- ORE 位被置位。
- RDR 内容将不会丢失。读 USART_DR 寄存器仍能得到先前的数据。
- 移位寄存器中以前的内容将被覆盖。随后接收到的数据都将丢失。
- 如果 RXNEIE 位置 1 或 EIE 位都置 1，则会生成中断。
- 顺序执行对 USART_SR 和 USART_DR 寄存器的读操作，可复位 ORE 位

■ 使能 FIFO 模式

移位寄存器准备好传送并且接收 FIFO 已满时，会发生上溢错误

在 RXFIFO 中出现一个空闲位置之前，数据无法从移位寄存器传送到 USART_DR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。

发生上溢错误时：

- ORE 位置 1。
- RXFIFO 中的第一个条目不会丢失。通过读取 USART_DR 寄存器可获得此条目。
- 移位寄存器会被覆盖。之后，上溢期间接收到的任何数据都将丢失。
- 如果 RXFNEIE 位置 1 或 EIE 位都置 1，则会生成中断。

注意：当 ORE 位置位时，表明至少有 1 个数据已经丢失。

禁止 FIFO 模式时，有以下两种可能

■ 如果 RXNE=1，上一个有效数据还在接收寄存器 RDR 上，可以被读出。

■ 如果 RXNE=0，这意味着上一个有效数据已经被读走，RDR 已经没有东西可读。当上一个有效数据在 RDR 中被读取的同时又接收到新的（也就是丢失的）数据时，此种情况可能发生。在读序列期间（在 USART_SR 寄存器读访问和 USART_DR 读访问之间）接收到新的数据，此种情况也可能发生。

28.3.7.6 选择时钟源和合适的过采样方案

时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器采用不同的用户可配置过采样技术（除了同步模式下），可以从噪声中提取有效数据。这可在最大通信速度与抗噪声/时钟误差性能之间实现最佳平衡。

可通过编程 USART_CR3 寄存器中的 OVER8 位来选择采样方法，且采样时钟可以是波特率时钟的 16 倍或 8 倍。

根据应用：

■ 选择 8 倍过采样（OVER8 = 1）以获得更高的速度（高达 usart_ker_ck/8）。这种情况下接收器对时钟偏差的最大容差将会降低。

■ 选择 16 倍过采样（OVER8 = 0）以增加接收器对时钟偏差的容差。在这种情况下，最大速度被限制为最大 usart_ker_ck/16。

帧中检测到噪声时：

■ 在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）的上升沿时 NE 位置 1。

■ 无效数据从移位寄存器传送到 USART_RDR 寄存器。

■ 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时会发出中断。

■ 先读出 USART_SR，再读出 USART_DR 寄存器，将清除 NE 标志位

Puya Confidential

注意：SPI 模式不支持噪声错误。

智能卡、IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下，OVER8 位必须为零。

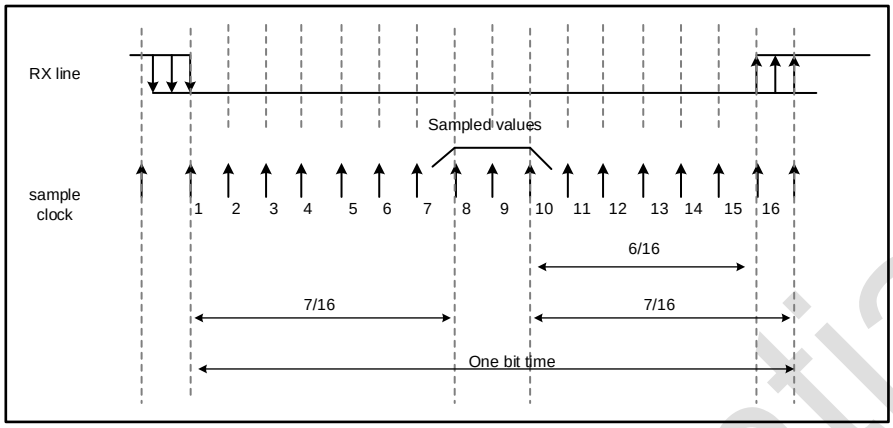


图 28-6 16 位过采样时噪声检测时数据采样

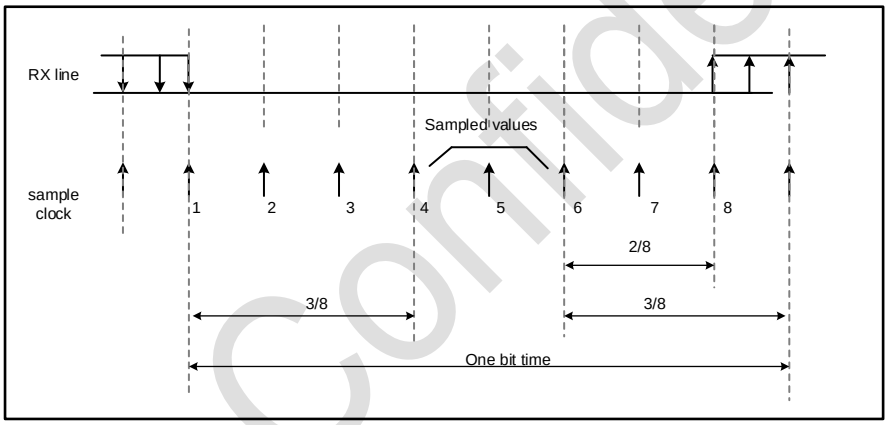


图 28-7 8 位过采样时噪声检测时数据采样

表 28-2 检测噪声的数据采样

采样值	NE 状态	接收的值	数据有效性
000	0	0	有效
001	1	0	无效
010	1	0	无效
011	1	1	无效
100	1	0	无效
101	1	1	无效
110	1	1	无效
111	0	1	有效

28.3.7.7 帧错误

如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。

检测到帧错误被时：

Puya Confidential

- FE 位由硬件置 1
- 无效数据从移位寄存器传送到 USART_DR 寄存器（使能 FIFO 模式时为 RXFIFO）。
- 在单字节通信时，无法中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时会发出中断。
- 顺序执行对 USART_SR 和 USART_DR 寄存器的读操作，可复位 FE 位。

注意：UART 8 位过采样模式和 SPI 模式不支持帧错误。

28.3.7.8 接收期间可配置的停止位

可通过 USART_CR2 的控制位配置要接收的停止位的数量：可以是 1 或 2 个（正常模式下），也可以是 0.5 或 1.5 个（智能卡模式下）。

0.5 个停止位（在智能卡模式下接收时）：不会对 0.5 个停止位进行采样。结果，选择 0.5 个停止位时，无法检测到帧错误和断开帧。

1 个停止位：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。

1.5 个停止位（在智能卡模式下）

在智能卡模式下发送时，设备必须检查数据是否正确发送。因此，必须使能接收器块（USART_CR1 中的 RE = 1）并检查停止位，以测试智能卡是否已检测到奇偶校验错误。发生奇偶校验错误时，智能卡会在采样时将数据信号强制为低电平（即 NACK 信号），该信号被标记为帧错误。之后，FE 标志在 1.5 个停止位的末尾由 RXNE 标志（使能 FIFO 模式时为 RXFNE）置 1。在第 16、第 17 和第 18 次采样时对 1.5 个停止位进行采样（停止位采样开始后维持 1 个波特率时钟周期）。1.5 个停止位可分为 2 个部分：0.5 个波特时钟周期（未发生任何动作），然后是 1 个正常的停止位周期（一半时间处进行采样）。

2 个停止位

采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。如果在第一个停止位期间检测到帧错误，则帧错误标志会置 1。发生帧错误时不检测第 2 个停止位。RXNE 标志（使能 FIFO 模式时为 RXFNE）将在第一个停止位结束时置 1。

28.3.8 波特率生成

分数波特率的产生接收器和发送器的波特率在 USARTDIV 的整数和小数寄存器中的值应设置成相同。

TX / RX 波特率 = fCK /(16*USARTDIV)

这里的 fCK 是给外设的时钟 USARTDIV 是一个无符号的定点数。这 12 位的值设置在 USART_BRR 寄存器。

注：在写入 USART_BRR 之后，波特率计数器会被波特率寄存器的新值替换。因此，不要在通信进行中改变波特率寄存器的数值。

如何从 USART_BRR 寄存器值得到 USARTDIV

例 1：

如果 DIV_Mantissa = 27，DIV_Fraction = 12 (USART_BRR=0x1BC),

则：

Mantissa (USARTDIV) = 27

Fraction (USARTDIV) = 12/16 = 0.75

所以 USARTDIV = 27.75

Puya Confidential

例 2:

要求 USARTDIV = 25.62,

就有:

DIV_Fraction = 16*0.62 = 9.92

最接近的整数是: 10 = 0x0A

DIV_Mantissa = mantissa (25.620) = 25 = 0x19

于是, USART_BRR = 0x19A

例 3:

要求 USARTDIV = 50.99

就有:

DIV_Fraction = 16*0.99 = 15.84

最接近的整数是: 16 = 0x10 => DIV_frac[3:0]溢出 => 进位必须加到小数部分

DIV_Mantissa = mantissa (50.990 + 进位) = 51 = 0x33

于是: USART_BRR = 0x330, USARTDIV=51

波特率		f_{pclk} = 36 MHz			f_{pclk} = 72 MHz		
序号	kbps	实际	置于波特率寄存器中的值	误差%	实际	置于波特率寄存器中的值	误差%
1	2.4	2.400	937.5	0.00%	2.4	1875	0.00%
2	9.6	9.600	234.375	0.00%	9.6	468.75	0.00%
3	19.2	19.200	117.1875	0.00%	19.2	234.375	0.0%
4	57.6	57.600	39.0625	0.00%	57.6	78.125	0.0%
5	115.2	115.385	19.5	0.15%	115.2	39.0625	0.0%
6	230.4	230.769	9.75	0.16%	230.769	19.5	0.16%
7	460.8	461.538	4.875	0.16%	461.538	9.75	0.16%
8	921.6	923.076	2.4375	0.16%	923.076	4.875	0.16%
9	2250	2250.000	1	0.00%	2250	2	0%
10	4500	不可能	不可能	不可能	4500	1	0%

注: CPU 的时钟频率越低，则某一特定波特率的误差也越低。可以达到的波特率上限可以由这组数据得到。

28.3.9 USART 接收容忍度

只有当整体的时钟系统地变化小于 USART 异步接收器能够容忍的范围，USART 异步接收器才能正常地工作。影响这些变化的因素有:

- DTRA: 由于发送器误差而产生的变化(包括发送器端振荡器的变化)
- DQUANT: 接收器端波特率取整所产生的误差
- DREC: 接收器端振荡器的变化
- DTCL: 由于传输线路产生的变化(通常是由于收发器在由低变高的转换时序，与由高变低转换时序之间的不一致性所造成)。

需要满足: DTRA + DQUANT + DREC + DTCL < USART 接收器的容忍度。

对于正常接收数据，USART 接收器的容忍度等于最大能容忍的变化，它依赖于下述选择:

- 由 USART_CR1 寄存器的 M 位定义的 9/10/11 位字符长度
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍采样
- 是否使用分数波特率产生

Puya Confidential

表 28-3 当 DIV_FRA 为 0 时的 USART 接收容忍度

M 位	OVR8=0		OVR8=1	
	认为 NF 是错误	不认为 NF 是错误	认为 NF 是错误	不认为 NF 是错误
00	2.05%	3.59%	2.55%	2.55%
01	1.84%	3.22%	2.32%	2.32%
10	2.29%	4.07%	2.85%	2.85%

表 28-4 当 DIV_FRA 不为 0 时的 USART 接收容忍度

M 位	OVR8=0		OVR8=1	
	认为 NF 是错误	不认为 NF 是错误	认为 NF 是错误	不认为 NF 是错误
00	1.43%	3.24%	2.25%	2.25%
01	1.28%	2.85%	2.05%	2.05%
10	1.59%	3.60%	2.53%	2.53%

注意：当接收的帧恰好包含 10 个（M 位 =00)、11 个（M 位 = 01）或 9 个（M 位 = 10）位时间的空闲帧时，上表中指定的数据可能与特例中的数据略微不同。

28.3.10 USART 自动波特率检测

USART 能够基于一个字符的接收，检测并自动设定 USART_BRR 寄存器的值。自动波特率检测在以下场景是有用处的：

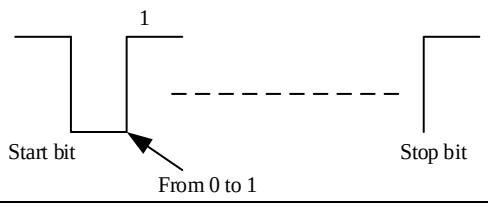
- 系统通讯速率提前未确认
- 系统正在使用相对低精度的时钟源，该机制允许在不测量时钟偏差的情况下，获得正确的波特率时钟源频率必须与被期望的通讯速率兼容。
- 16 倍过采样时，波特率范围为 usart_ker_ck/65535 ~ usart_ker_ck/16。
- 8 倍过采样时，波特率范围为 usart_ker_ck/65535~usart_ker_ck/8。

在激活自动波特率检测之前，自动波特率检测模式必须被选择(通过 USART_CR3 寄存器的 ABRMOD[1:0]位被选择)。基于不同的字符模式，有各种模式。

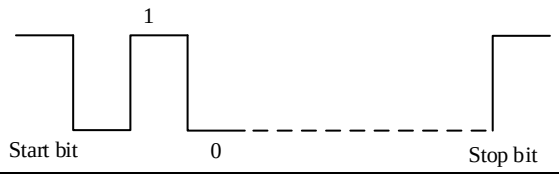
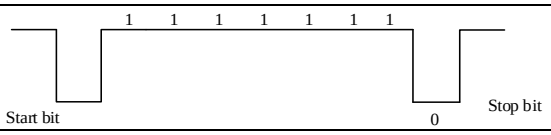
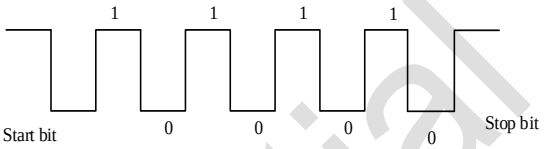
在这些自动波特率模式下，波特率在同步数据接收期间会被测量几次，每次测量都会跟之前进行对比。这些模式是：

- MODE 0：任何以 1 开始的字符。在这种情况下，USART 测量起始位的宽度(下降沿到上升沿)
- MODE 1：任何以 10xx 位开始的字符。在这种情况下，USART 测量起始位和第一个数据位的宽度。该测量在下降沿到下降沿之间进行，以确保在慢速信号上升情况下更好的精度。
- MODE2：0x7F 字符帧。在此情况下，将首先在起始位结束处更新波特率，然后是在位 6 结束时更新波特率。
- MODE3：0x55 字符帧。在此情况下，将首先在开始位结束处更新波特率，然后是位 0 的结束处，最后是在位 6 的结束处。同时，对 RX 线路的每个中间转换执行其它检查。

注意：在使用 mode 1 的时候，只支持正确的开始字符。

ABR 模式	说明	波形
0	接收的字符为“1”位为开头的字符。这种情况下，USART 会测量起始位的持续时间(下降沿到上升沿)	

Puya Confidential

1	以 10xx 模式开头的任何字符。 这种情况下，USART 会测量起始位和第一个数据位的持续时间。从下降沿到下降沿测得的持续时间，**可在信号斜率较小时确保较高的精度。**	
2	0x7F 字符帧。 在此情况下，将首先在起始位结束处更新波特率，然后时在位 6 结束时更新波特率。	
3	0x55 字符帧。 在此情况下，将首先在开始位结束处更新波特率，然后是位 0 的结束处，最后是在位 6 的结束处。同时，对 RX 线路的每个中间转换执行其他检查。	

通过置位 USART_CR3 寄存器的 ABREN 位，可以激活自动波特率检测功能。USART 将等待 RX 上的第一个字符。USART_SR 寄存器中的 ABRF 标志位置位，表明了自动波特率检测的完成。如果通讯线上是有噪声的，则自动波特率检测的正确进行不能被保证。在这种情况下，BRR 的值可能被破坏，ABRE 错误标志位将被置位。如果通信速度不在自动波特率检测范围（数据位持续时间不在 16 个和 65536 个时钟周期（16 倍过采样时）之间，也不在 8 个和 65536 个时钟周期（8 倍过采样时）之间）内，ABRE 错误也会发生。

RXNE 中断显示操作的完成。

注意：自动波特率检测可以通过复位 ABRF 标志(通过写 0)再次启动，如果前后波特率差距过大，建议复位 usart 模块来重新使用自动波特率功能。

注：如果在自动波特期间，清零 UE，BRR 值可能被破坏。

28.3.11 USART 多处理器通信

通过 USART 可以实现多处理器通信（将几个 USART 连在一个网络里）。例如某个 USART 设备可以是主，它的 TX 输出和其他 USART 从设备的 RX 输入相连接；USART 从设备各自的 TX 输出逻辑地“与”在一起，并且和主设备的 RX 输入相连接。

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

未被寻址的设备可启用其静默功能置于静默模式。在静默模式里：

- 任何接收状态位都不会被置 1。
- 所有接收中断被禁止。
- USART_CR1 寄存器中的 RWU 位被置 1。RWU 可以被硬件自动控制或在某个条件下由软件写入。

根据 USART_CR1 寄存器中的 WAKE 位状态，USART 可以用二种方法进入或退出静默模式。

- 如果 WAKE 位为 0：进行空闲总线检测。
- 如果 WAKE 位为 1：进行地址标记检测。

28.3.11.1 空闲帧检测(WAKE=0)

当 RWU 位被写 1 时，USART 进入静默模式。当检测到一空闲帧时，它被唤醒。然后 RWU 被硬件清零，但是 USART_SR 寄存器中的 IDLE 位并不置起。 RWU 还可以被软件写 0。下图给出利用空闲总线检测来唤醒和进入静默模式的一个例子

PUYA Confidential

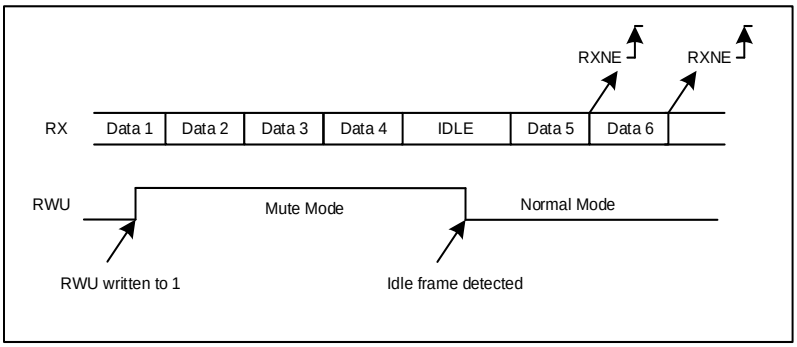


图 28-8 使用空闲帧检测的静默模式

28.3.11.2 4 位/7 位地址标志检测（WAKE=1）

在这个模式里，如果 MSB 是 1，该字节被认为是地址，否则被认为是数据。在一个地址字节中，目标接收器的地址被放在 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

注意：在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成(ADD[5:0]和 ADD[7:0])。

如果接收到的字节与它的编程地址不匹配时，USART 进入静默模式。此时，硬件设置 RWU 位。

接收该字节既不会设置 RXNE 标志也不会产生中断或发出 DMA 请求，因为 USART 已经在静默模式。

使能 FIFO 管理时，软件必须确保在进入静默模式前 RXFIFO 中至少有一个空位置。

当接收到的字节与接收器内编程地址匹配时，USART 退出静默模式。然后 RWU 位被清零，随后的字节被正常接收。由于 RWU 位已清零，RXNE/RXFNE 位会针对地址字符置 1。

当接收缓冲器不包含数据时(USART_SR 的 RXNE/RXFNE=0)，RWU 位可以被写 0 或 1。否则，该次写操作被忽略。下图给出利用地址标记检测来唤醒和进入静默模式的例子。

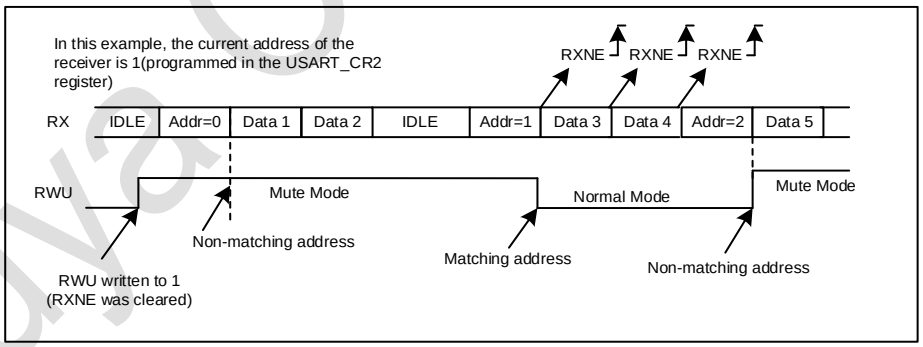


图 28-9 使用地址标记检测的静默模式

28.3.12 Modbus 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。Modbus/RTU 是一个半双工块传输协议。该协议的控制部分（地址识别、块完整性控制和命令解析）必须用软件实现。

USART 为块结束检测提供基本支持，无需软件开销或其他资源。

Modbus/RTU

在此模式下，一个块的结束通过超过 2 个字符时间的“静默”（空闲线路）来识别。此功能通过可编程的超时功能实现。

Puya Confidential

超时功能和中断必须分别通过 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位激活。与 2 个字符时间 (例如 22 个位时间) 的超时相对应的值必须在 RTO 寄存器中编程。如果在此期间接收线路空闲，则在接收到最后一个停止位后，将生成中断，同时通知软件当前块接收已完成。

Modbus/ASCII

在此模式下，块结束通过特定 (CR/LF) 字符序列识别。 USART 通过字符匹配功能管理此机制。通过在 ADD[7:0] 位域中编程 LF ASCII 码以及激活字符匹配中断 (CMIE = 1)，软件可在接收到 LF 时获得通知并检查 DMA 缓存区中的 CR/LF。

28.3.13 奇偶校验控制

设置 USART_CR1 寄存器上的 PCE 位，可以使能奇偶控制(发送时生成一个奇偶位，接收时进行奇偶校验)。根据 M 位定义的帧长度，可能的 USART 帧格式列在下表中。

表 28-5 帧格式

M bit	PCE bit	USART frame
00	0	SB—8 bit data—STB
00	1	SB—7 bit data—PB—STB
01	0	SB—9 bit data—STB
01	1	SB—8 bit data—PB—STB
10	0	SB—7 bit data—STB
10	1	SB—6 bit data—PB—STB

注：SB 为起始位，STB 为停止位，PB 为奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 7 位或者第 8 位或者第 9 位，具体取决于 M 位的值）。

在用地址标记唤醒设备时，地址的匹配只考虑到数据的 MSB 位，而不用关心校验位。（MSB 是数据位中最后发出的，后面紧跟校验位或者停止位）

28.3.13.1 偶校验

校验位使得一帧中的 6 或 7 或 8 个 LSB 数据以及校验位中'1'的个数为偶数。

例如：数据=00110101，有 4 个'1'，如果选择偶校验（在 USART_CR1 中的 PS = 0），校验位将是'0'。

28.3.13.2 奇校验

此校验位使得一帧中的 6 或 7 或 8 个 LSB 数据以及校验位中'1'的个数为奇数。

例如：数据=00110101，有 4 个'1'，如果选择奇校验（在 USART_CR1 中的 PS = 1），校验位将是'1'。

28.3.13.3 接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_SR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。软件先读 USART_SR 寄存器后读 USART_DR 寄存器可以清零该位。但软件在清该位前必须等待 RXNE=1。

Puya Confidential

28.3.13.4 发送时的奇偶校验生成

如果 USART_CR1 的 PCE 位被置位，写进数据寄存器的数据的 MSB 位被校验位替换后发送出去（如果选择偶校验偶数个'1'，如果选择奇校验奇数个'1'）。如果奇偶校验失败，USART_SR 寄存器中的 PE 标志被置'1'，并且如果 USART_CR1 寄存器的 PEIE 在被预先设置的话，中断产生。

28.3.14 LIN（局域互联网）模式

配置 USART_CR2.LINEN=1 选择 LIN 模式。在 LIN 模式下，下列位必须保持位 0：

- USART_CR2 寄存器的 STOP/CLKEN;
- USART_CR3 寄存器的 SCEN/HDSEL/IREN.

28.3.14.1 发送

与一般 USART 发送相比，LIN 发送有如下区别：

- M 位清零以配置 8 位字长度；
- 需要配置 USART_CR2.LINEN=1。置位 SBK 后将发送 13 位 0 作为断开帧。然后发送一位"1"，以允许对下一个起始位检测。

28.3.14.2 接收

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

当接收器被激活时(USART_CR1 的 RE=1)，电路监测 RX 上的起始信号。监测起始位的方法同检测断开符号或数据是一样的。当起始位被检测到后，电路对接下来的位进行采样，方法与数据采样相同（第 8、第 9、第 10 次采样）。如果 10 个（当 USART_CR2 的 LBDL = 0）或 11 个（当 USART_CR2 的 LBDL = 1）连续位都是 0，并且其后跟随分隔符，则 USART_SR 的 LBD 标志被置 1。如果 LBDIE 位 =1，中断产生。在确认断开符号前，要检查分隔符，因为它意味 RX 线已经回到高电平。

如果在第 10 或 11 个采样点之前采样到了 1，检测电路取消当前检测并重新寻找起始位。如果 LIN 模式被禁止，接收器继续如正常 USART 那样工作，不需要考虑检测断开符号。

如果 LIN 模式没有被激活（LINEN=0），接收器仍然正常工作于 USART 模式，不会进行断开检测。

如果 LIN 模式被激活（LINEN=1），只要发生帧错误（例如，在 0 处检测到停止位，这种情况可能出现在任何断开帧中)，接收器即会停止，直到断开符号检测电路接收到" 1"（断开字不完整时）或接收到分隔符(检测完整的断开符号)为止

要一发生帧错误（也就是停止位检测到'0'，这种情况出现在断开帧），接收器就停止，直到断开符号检测电路接收到一个 1（这种情况发生于断开符号没有完整的发出来），或一个定界符（这种情况发生于已经检测到一个完整的断开符号）。

Puya Confidential

本情况：断开帧长度不够：需要舍弃该帧，不设置LBD

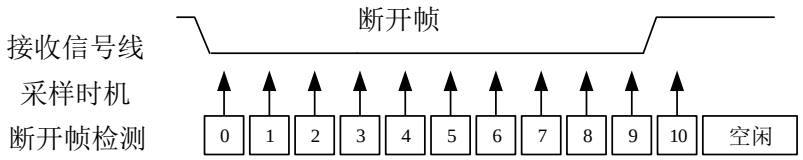
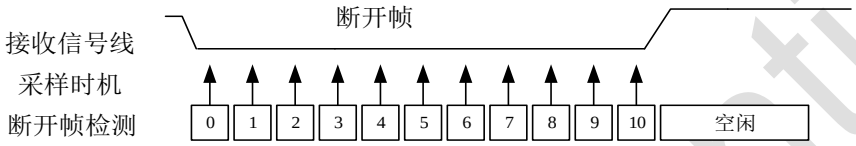


图 28-10 LIN 模式下的断开帧长度不够的情况

本情况：断开帧长度刚好，设置LBD



本情况：断开帧长度很长，设置LBD

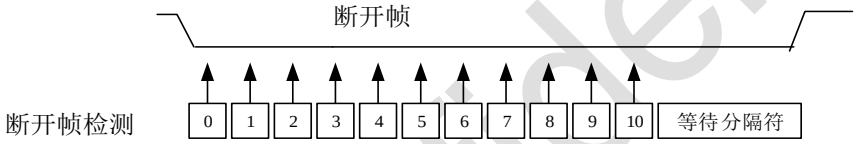
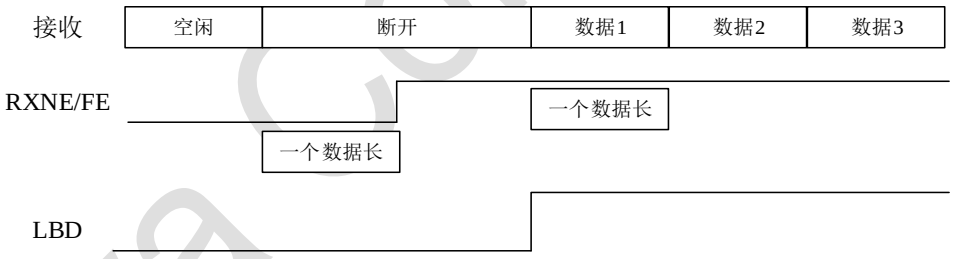


图 28-11 LIN 模式下的断开帧长度够的情况

断开发生在空闲后



断开发生在正在接收数据时

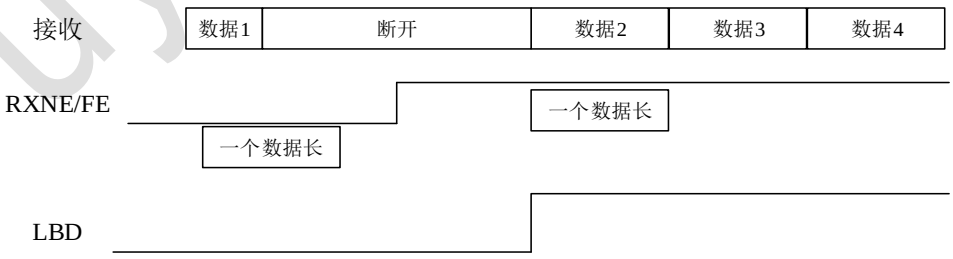


图 28-12 LIN 模式下的断开检测与帧错误的检测

Puya Confidential

28.3.15 USART 同步模式

28.3.15.1 主模式

通过写 USART_CR2 寄存器的 CLKEN 位为 1，选择同步主模式。在同步模式里，下列位必须保持清零状态：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 SCEN,HDSEL 位。

在此模式下，USART 可用于在主模式下控制双向同步串行通信。CK 脚是 USART 发送器时钟的输出。在起始位和停止位期间，CK 脚上没有时钟脉冲。在最后一个有效数据位期间，是否生成时钟取决于 USART_CR2 寄存器中 LBCL 位的状态。USART_CR2 寄存器的 CPOL 位允许用户选择时钟极性，USART_CR2 寄存器上的 CPHA 位允许用户选择外部时钟的相位。

在总线空闲期间，实际数据到来之前以及发送断开符号的时候，外部 CK 时钟不被激活。

同步模式时，USART 发送器和异步模式里工作一模一样。但是因为 CK 是与 TX 同步的（根据 CPOL 和 CPHA），所以 TX 上的数据是随 CK 同步发出的。

同步模式的 USART 接收器工作方式与异步模式不同。如果 RE=1，数据在 CK 上采样（根据 CPOL 和 CPHA 决定在上升沿还是下降沿），不需要任何的过采样。但必须考虑建立时间和持续时间（取决于波特率，1/16 位时间）。

注意：CK 脚同 TX 脚一起联合工作。因而，只有在使能了发送器（TE = 1），并且发送数据时（写入数据至 USART_DR 寄存器）才提供时钟。这意味着在没有发送数据时是不可能接收一个同步数据的。

LBCL，CPOL 和 CPHA 位的正确配置，应该在发送器和接收器都被禁止时；当使能了发送器或接收器时，这些位不能被改变。

建议在同一条指令中设置 TE 和 RE，以减少接收器的建立时间和保持时间。

USART 只支持主模式：它不能用来自其他设备的输入时钟接收或发送数据（CK 永远是输出）。

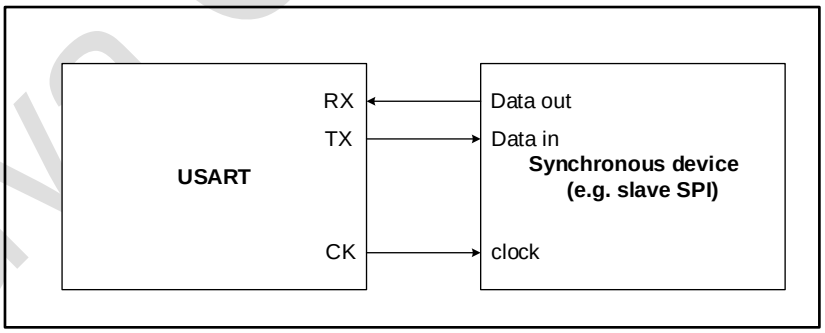


图 28-13 USART 同步传输例子

Puya Confidential

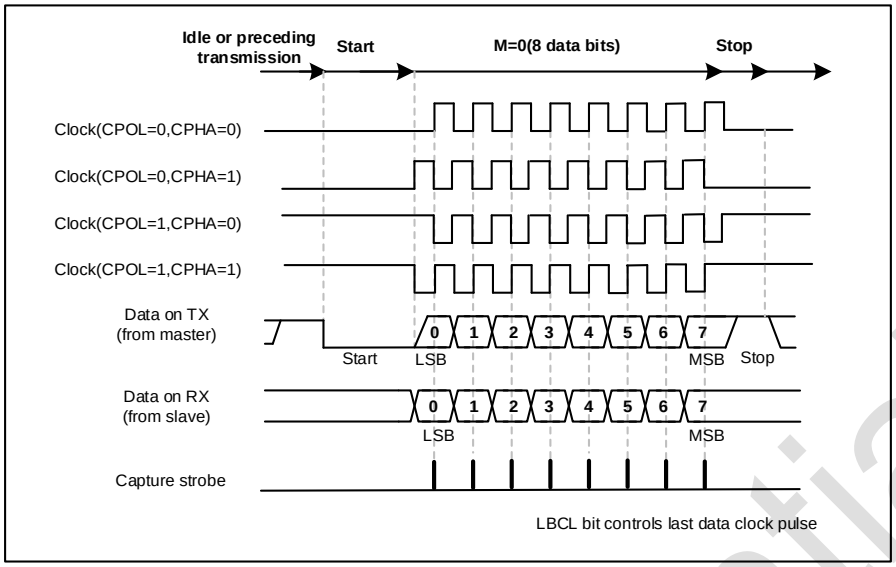


图 28-14 USART 数据时钟时序图（M = 00）

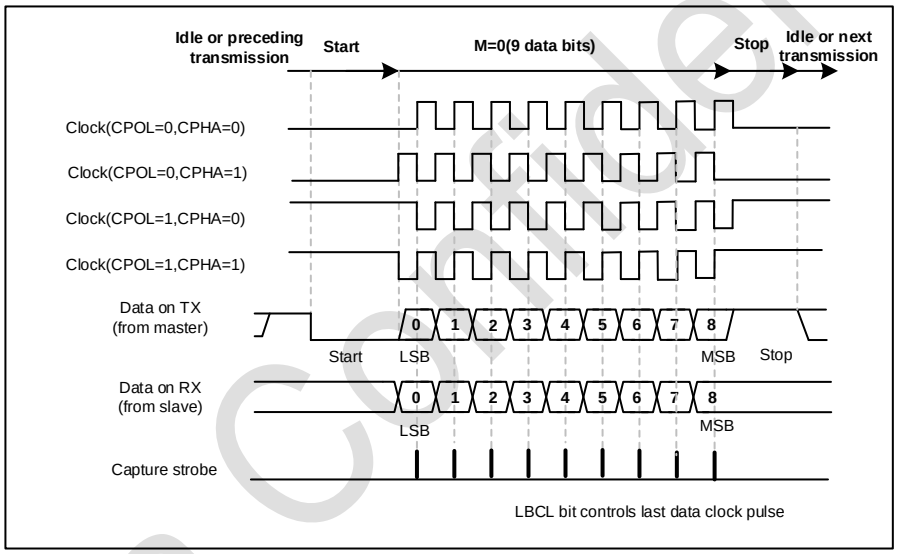


图 28-15 USART 数据时钟时序图（M = 01）

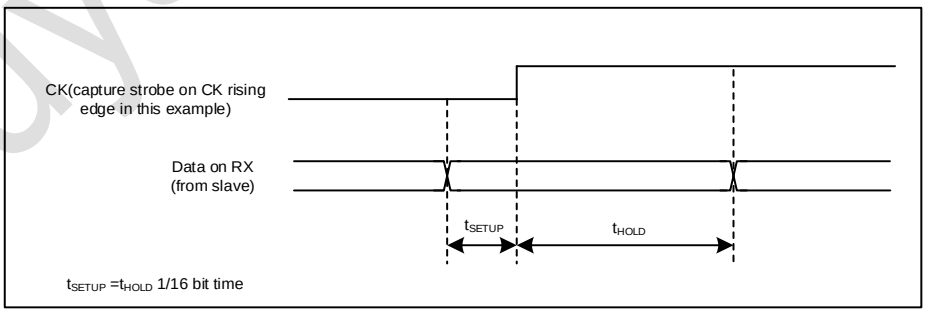


图 28-16 RX 数据建立/保持时间

Puya Confidential

28.3.16 USART 单线半双工通信

单线半双工模式通过设置 USART_CR3 寄存器的 HDSEL 位选择。在这个模式里，下面的位必须保持清零状态：

- USART_CR2 寄存器的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器的 SCEN 和 IREN 位。

USART 可以配置成遵循单线半双工协议。在单线半双工模式下，TX 和 RX 引脚在芯片内部互连。使用 USART_CR3 寄存器中的 HDSEL 位选择半双工和全双工通信。

当 HDSEL 为'1'时

- TX 和 RX 线路从内部相连接
- RX 不再被使用
- 当没有数据传输时，TX 总是被释放。因此，它在空闲状态的或接收状态时表现为一个标准 I/O 口。这就意味该 I/O 在不被 USART 驱动时，必须配置成悬空输入（或开漏的输出高）。

除此以外，通信与正常 USART 模式类似。由软件来管理线上的冲突 (例如通过使用一个中央仲裁器）。特别的是，发送从不会被硬件所阻碍。当 TE 位被设置时，只要数据一写到数据寄存器上，发送就继续。

28.3.17 接收器超时

接收器超时功能可通过将 USART_CR2 控制寄存器中的 RTOEN 位置 1 来使能。

超时间隔通过 USART_RTOR 寄存器中的 RTO 位域进行编程。

接收器超时计数器遵循以下规则开始计数：

STOP = 00 或 STOP = 11 时从停止位结束时开始计数。

STOP = 10 时从第二个停止位结束时开始计数。

STOP = 01 时从停止位开始时开始计数。

经过超时间隔后，USART_SR 寄存器中的 RTOF 标志置 1。如果 USART_CR1 寄存器中的 RTOIE 位置 1，则会产生超时中断。

28.3.18 智能卡

设置 USART_CR3 寄存器的 SCEN 位选择智能卡模式。在智能卡模式下，下列位必须保持清零：

- USART_CR2 寄存器的 LINEN 位
- USART_CR3 寄存器的 HDSEL 位和 IREN 位

此外，CLKEN 位可以被设置，以提供时钟给智能卡。

该接口符合 ISO7816-3 标准，支持智能卡异步协议。T = 0（字符模式）和 T = 1（块模式）都支持。

注意：智能卡模式不能使用 FIFO 功能传输数据

28.3.18.1 T = 0 (字符模式)

USART 应该被设置为：

- 8 位数据位加校验位：此时 USART_CR1 寄存器中 M=01、PCE=1
- 发送和接收时为 1.5 个停止位：即 USART_CR2 寄存器的 STOP=11。

注：也可以在接收时选择 0.5 个停止位，但为了避免在 2 种配置间转换，建议在发送和接收时使用 1.5 个停止位。

在 T = 0（字符）模式下，在保护时间段内每个字符的末尾表示奇偶校验错误。

Puya Confidential

下图给出的例子说明了数据线上，在有校验错误和没校验错误两种情况下的信号。

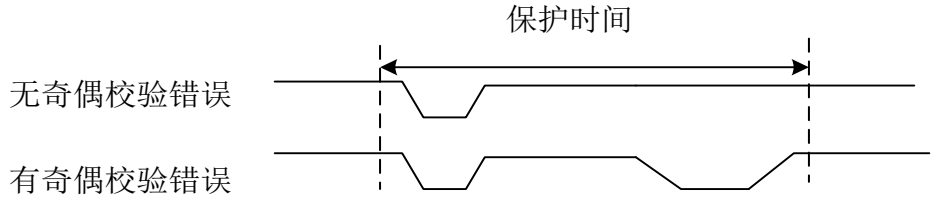


图 28-17 ISO7816-3 异步协议

当与智能卡相连接时，USART 的 TX 驱动一根智能卡也驱动的双向线。为了做到这点，RX 必须和 TX 连接到相同的 I/O 口。在发送开始位和数据字节期间，发送器的输出使能位 TX_EN 被置起，在发送停止位期间被释放(弱上拉)，因此在发现校验错误的情况下接收器可以将数据线拉低。如果 TX_EN 不被使用，在停止位期间 TX 被拉到高电平：这样的话，只要 TX 配置成开漏，接收器也可以驱动这根线。

智能卡是一个单线半双工通信协议

- 从发送移位寄存器发送数据会经过至少 1/2 个波特时钟周期的延迟。在正常操作时，一个满的发送移位寄存器将在下一个波特时钟沿开始向外移出数据。在智能卡模式里，此发送被延迟 1/2 波特时钟。
- 如果在接收一个设置为 1.5 个停止位的数据帧期间，检测到一奇偶校验错误，在完成接收该帧后（即停止位结束时），发送线被拉低一个波特时钟周期。这是告诉智能卡发送到 USART 的数据没有被正确地接收到。此 NACK 信号（拉低发送线一个波特时钟周期）在发送端将产生一个帧错误（发送端被配置成 1.5 个停止位）。应用程序可以根据协议处理重新发送数据。如果设置了 NACK 控制位，发生校验错误时接收器会给出一个 NACK 信号；否则就不会发送 NACK。

注意：若配置为 0.5 个停止位，则不会产生 NACK。

- 智能卡传输中的自动重试：在 USART 检测 NACK 和重复字符的起始位之间插入 2.5 个波特周期的延迟。TC 位在接收到最后一个重复字符后立即置 1（无守护时间）。如果软件想要再次重复它，它必须确保标准要求的最小 2 个波特周期。
- 在传输中，USART 在两个连续字符之间插入保护时间（按照保护时间寄存器中的编程）。由于保护时间是在前一个字符的停止位之后测量的，因此必须将 GT[7:0]寄存器编程为所需的 CGT（字符保护时间，由 7816-3 规范定义）减去 12（一个字符的持续时间）。
- TC 标志的置起可以通过编程保护时间寄存器得以延时。在正常操作时，当发送移位寄存器变空并且没有新的发送请求出现时，TC 被置起。在智能卡模式里，空的发送移位寄存器将触发保护时间计数器，使其递增计数至保护时间寄存器中的值。TC 在这段时间被强制拉低。当保护时间计数器达到保护时间寄存器中的值时，TC 被置高。
- TC 标志的撤销不受智能卡模式的影响。
- 如果发送器检测到一个帧错误(收到接收器的 NACK 信号)，发送器的接收功能模块不会把 NACK 当作起始位检测。根据 ISO 协议，接收到的 NACK 的持续时间可以是 1 或 2 波特时钟周期。
- 在接收器这边，如果一个校验错误被检测到，并且 NACK 被发送，接收器不会把 NACK 检测成起始位。

注意：1. 断开符号在智能卡模式里没有意义。一个带帧错误的 0x00 数据将被当成数据而不是断开符号。

2. 当来回切换 TE 位时，没有 IDLE 帧被发送。ISO 协议没有定义 IDLE 帧。

Puya Confidential

下图详述了 USART 是如何采样 NACK 信号的。在这个例子里， USART 正在发送数据，并且被配置成 1.5 个停止位。为了检查数据的完整性和 NACK 信号， USART 的接收功能块被激活。

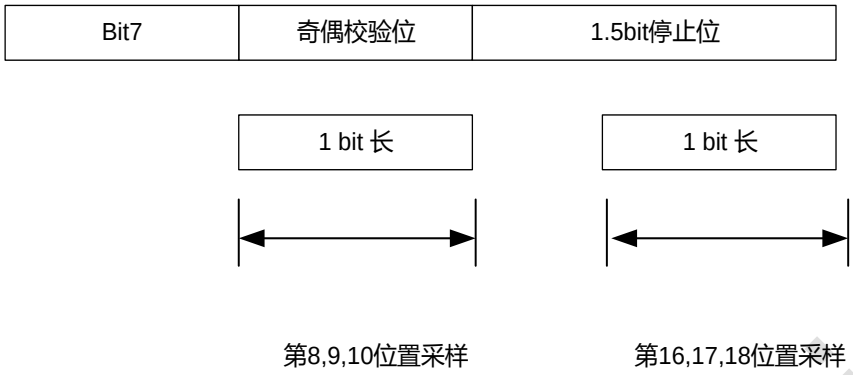


图 28-18 使用 1.5 停止位检测奇偶检验错

USART 可以通过 CK 输出为智能卡提供时钟。在智能卡模式里， CK 不和通信直接关联，而是先通过一个 5 位预分频器简单地用内部的外设输入时钟来驱动智能卡的时钟。分频率在预分频寄存器 USART_GTPR 中配置。 CK 频率可以从 fCK/2 到 fCK/62，这里的 fCK 是外设输入时钟。

28.3.18.2 T = 1（块模式）

在块模式下，向智能卡发出读取请求时，软件必须将 RTOR 寄存器编程为 BWT(块等待时间)- 11 值。如果在此期限到期之前没有收到卡的答复，则会产生超时中断。如果第一个字符在期限到期之前被接收，RXNE/RXFNE 中断将发出信号

注：RXNE/RXFNE 中断必须启用，即使在 DMA 模式下使用 USART 以块模式从智能卡读取。同时，DMA 必须只在接收到第一个字节之后才启用。

在智能卡协议定义中，BWT/CWT 值应该从最后一个字符的开始(起始位)定义。考虑到最后一个字符本身的长度，RTO 寄存器必须分别编程为 BWT -11 或 CWT -11。

块长度计数器用于计算 USART 接收到的所有字符。当 USART 发送时，这个计数器被重置。块的长度由智能卡在块的第三个字节(序言字段)中进行通信。这个值必须编程到 USART_RTOR 寄存器的 BLEN 字段中。当使用 DMA 模式时，在块开始之前，必须将该寄存器字段编程为最小值(0x0)。根据这个值，在接收到的第 4 个字符之后产生中断。软件必须读取 LEN 字段(第三个字节)，其值必须从接收缓冲区读取。

在中断驱动接收模式下，块的长度可以通过软件或编程 BLEN 值来检查。但是，在块开始之前，可以编程 BLEN 的最大值(0xFF)。真正的值是在接收到第三个字符后编程的。

如果块使用 LRC 纵向冗余检查(1 个结尾字节)，则 BLEN = LEN。如果块使用 CRC 机制(2 个结尾字节)，则必须编程 BLEN = LEN+1。总块长度(包括序言、尾声和信息字段)等于 BLEN+4。块的结束通过 EOBF 标志和中断(当设置 EOBIE 位时)向软件发出信号。

28.3.19 IrDA SIR ENDEC 功能模块

通过设置 USART_CR3 寄存器的 IREN 位选择 IrDA 模式。在 IRDA 模式里，下列位必须保持清零：

- USART_CR2 寄存器的 LINEN,STOP 和 CLKEN 位
- USART_CR3 寄存器的 SCEN 和 HDSEL 位。

Puya Confidential

28.3.19.1 IrDA 正常模式

IrDA SIR 物理层规定使用反相归零调制方案（RZI），该方案用一个红外光脉冲代表逻辑'0'。SIR 发送编码器对从 USART 输出的 NRZ（非归零）比特流进行调制。输出脉冲流被传送到一个外部输出驱动器和红外 LED。USART 为 SIR ENDEC 最高只支持到 115.2Kbps 速率。在正常模式里，脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器对来自红外接收器的归零位比特流进行解调，并将接收到的 NRZ 串行比特流输出到 USART。在空闲状态里，解码器输入通常是高（标记状态 marking state）。发送编码器输出的极性和解码器的输入相反。当解码器输入低时，检测到一个起始位。

- IrDA 是一个半双工通信协议。如果发送器忙（也就是 USART 正在送数据给 IrDA 编码器），IrDA 接收线上的任何数据将被 IrDA 解码器忽视。如果接收器忙（也就是 USART 正在接收从 IrDA 解码器来的解码数据），从 USART 到 IrDA 的 TX 上的数据将不会被 IrDA 编码。当接收数据时，应该避免发送，因为将被发送的数据可能被破坏。
- SIR 发送逻辑把'0'作为高脉冲发送，把'1'作为低电平发送。脉冲的宽度规定为正常模式时位周期的 3/16。
- SIR 接收逻辑把高电平状态解释为'1'，把低脉冲解释为'0'。
- 发送编码器输出与解码器输入有着相反的极性。当空闲时，SIR 输出处于低状态。
- SIR 解码器把 IrDA 兼容的接收信号转变成给 USART 的比特流。
- IrDA 规范要求脉冲要宽于 1.41 μs。脉冲宽度是可编程的。接收器端的尖峰脉冲检测逻辑滤除宽度小于 2 个 PSC 周期的脉冲(PSC 是在 IrDA 低功耗波特率寄存器 USART_GTPR 中编程的预分频值)。宽度小于 1 个 PSC 周期的脉冲一定被滤除掉，但是那些宽度大于 1 个而小于 2 个 PSC 周期的脉冲可能被接收或滤除，那些宽度大于 2 个周期的将被视为一个有效的脉冲。当 PSC=0 时，IrDA 编码器/解码器不工作。
- 接收器可以与一低功耗发送器通信。
- 在 IrDA 模式里， USART_CR2 寄存器上的 STOP 位必须配置成 1 个停止位。

28.3.19.2 IrDA 低功耗模式

发送器

在低功耗模式，脉冲宽度不再持续 3/16 个位周期。取而代之，脉冲的宽度是低功耗波特率的 3 倍，它最小可以是 1.42 MHz。通常这个值是 1.8432 MHz（1.42 MHz < PSC < 2.12 MHz）。一个低功耗模式可编程分频器把系统时钟进行分频以达到这个值。

接收器

低功耗模式的接收类似于正常模式的接收。为了滤除尖峰干扰脉冲，USART 应该滤除宽度短于 1 个 PSC 的脉冲。只有持续时间大于 2 个周期的 IrDA 低功耗波特率时钟（USART_GTPR 中的 PSC）的低电平信号才被接受为有效的信号。

注意：1. 宽度小于 2 个大于 1 个 PSC 周期的脉冲可能会也可能不会被滤除。

2. 接收器的建立时间应该由软件管理。IrDA 物理层技术规范规定了在发送和接收之间最小要有 10 ms 的延时（IrDA 是一个半双工协议）。

Puya Confidential

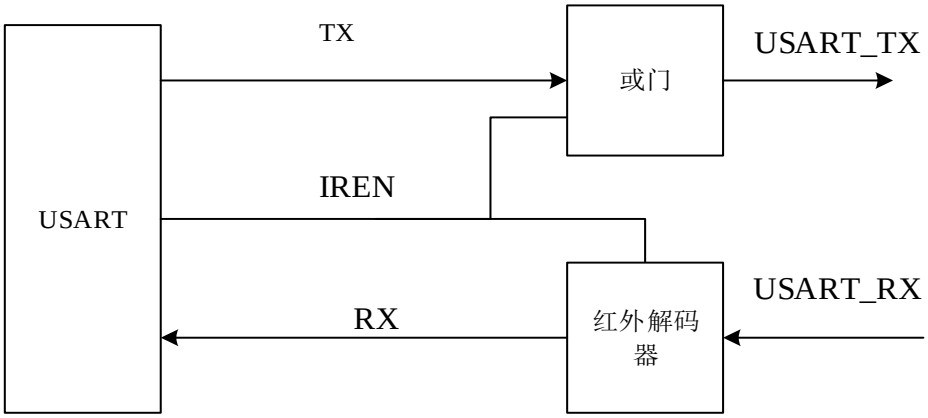


图 28-19 IrDA SIR ENDEC 框图

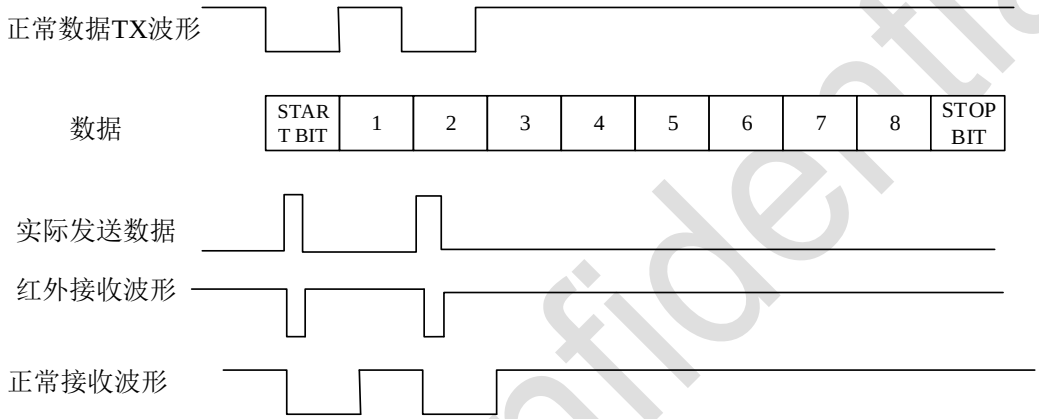


图 28-20 IrDA 数据调制（3/16）—普通模式

28.3.20 DMA 连续通讯

USART 可以利用 DMA 连续通信。RX 缓冲器和 TX 缓冲器的 DMA 请求是分别产生的。

28.3.20.1 DMA 发送

使用 DMA 进行发送，可以通过设置 USART_CR3 寄存器上的 DMAT 位激活。当 TXE 位（使能 FIFO 模式时为 TXFNF 标志）被置为'1'时，DMA 就从指定的 SRAM 区传送数据到 USART_DR 寄存器。为 USART 的发送分配一个 DMA 通道的步骤如下（x 表示通道号）：

1. 在 DMA 控制寄存器上将 USART_DR 寄存器地址配置成 DMA 传输的目的地址。在每个 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据将被传送到这个地址。
2. 在 DMA 控制寄存器上将存储器地址配置成 DMA 传输的源地址。在每个 TXE（使能 FIFO 模式时为 TXFNF）事件后，将从此存储器区读出数据并传送到 USART_DR 寄存器。
3. 在 DMA 控制寄存器中配置要传输的总的字节数。
4. 在 DMA 寄存器上配置通道优先级。
5. 根据应用程序的要求，配置在传输完成一半还是全部完成时产生 DMA 中断。
6. 通过写 0，清 TC 位。
7. 在 DMA 寄存器上激活该通道。

当传输完成 DMA 控制器指定的数据量时，DMA 控制器在该 DMA 通道的中断向量上产生一中断。

Puya Confidential

在发送模式下，当 DMA 传输完所有要发送的数据时，DMA 控制器设置 DMA_SR 寄存器的 TCIF 标志；监视 USART_SR 寄存器的 TC 标志可以确认 USART 通信是否结束，这样可以在关闭 USART 或进入停机模式之前避免破坏最后一次传输的数据；软件需要先等待 TXE=1，再等待 TC=1。

注意：在 DMA 传输结束之前不得将 DMAT 位清零。

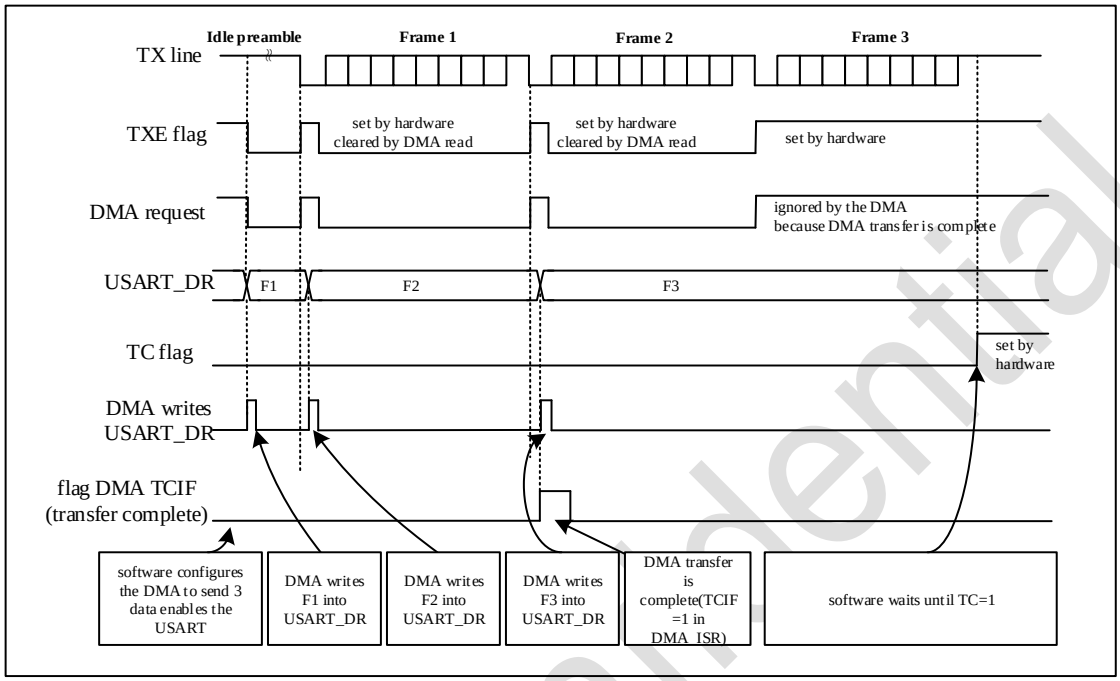


图 28-21 使用 DMA 发送

28.3.20.2 DMA 接收

可以通过设置 USART_CR3 寄存器的 DMAR 位激活使用 DMA 进行接收，每次接收到一个字节，DMA 控制器就就把数据从 USART_DR 寄存器传送到指定的 SRAM 区。为 USART 的接收分配一个 DMA 通道的步骤如下：

1. 通过 DMA 控制寄存器把 USART_DR 寄存器地址配置成传输的源地址。在每个 RXNE（使能 FIFO 模式时为 RXFNE）事件后，将从此地址读出数据并传输到存储器。
2. 通过 DMA 控制寄存器把存储器地址配置成传输的目的地址。在每个 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据将从 USART_DR 传输到此存储器区。
3. 在 DMA 控制寄存器中配置要传输的总的字节数。
4. 在 DMA 寄存器上配置通道优先级。
5. 根据应用程序的要求配置在传输完成一半还是全部完成时产生 DMA 中断。
6. 在 DMA 控制寄存器上激活该通道。

当接收完成 DMA 控制器指定的传输量时，DMA 控制器在该 DMA 通道的中断矢量上产生一中断。

注意：在 DMA 传输结束之前不得将 DMAR 位清零。

Puya Confidential

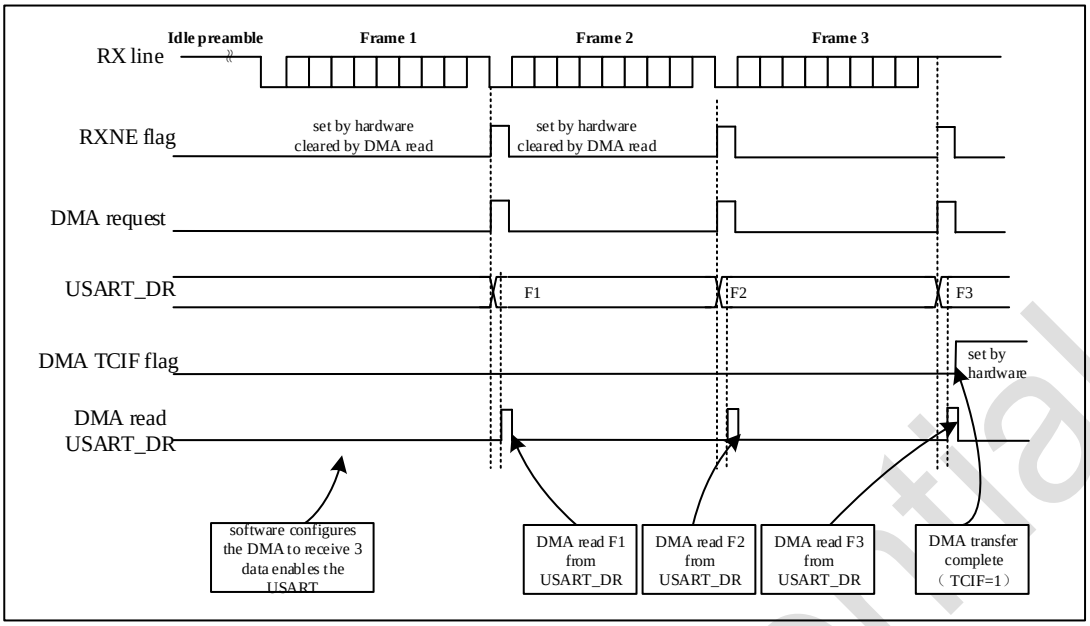


图 28-22 使用 DMA 接收

注意：使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = 1）触发。

28.3.21 多处理器通讯错误标志和中断产生

在多缓冲器通信的情况下，通信期间如果发生任何错误，在当前字节传输后将置起错误标志。如果中断使能位被设置，将产生中断。在单个字节接收的情况下，和 RXNE（使能 FIFO 模式时为 RXFNE）一起被置起的帧错误、溢出错误和噪音标志，有单独的错误标志中断使能位；如果设置了，会在当前字节传输结束后，产生中断。

28.3.22 硬件流控制

利用 CTS 输入和 RTS 输出可以控制 2 个设备间的串行数据流。下图表明在这个模式里如何连接 2 个设备。

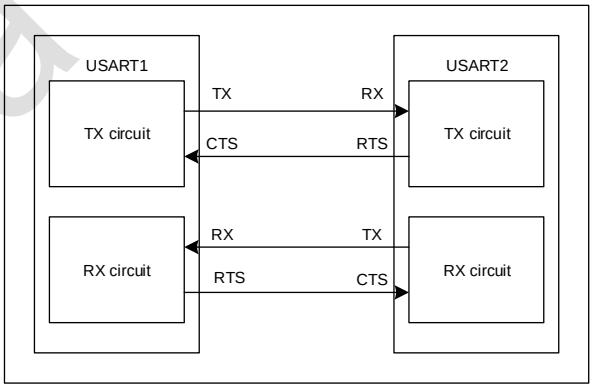


图 28-23 USART 间的硬件流控制

28.3.22.1 RTS 流控制

如果 RTS 流控制被使能（RTSE=1），只要 USART 接收器准备好接收新的数据，RTS 就变成有效（接低电平）。当接收寄存器内有数据到达时，RTS 被释放，由此表明希望在当前帧结束时停止数据传输。

Puya Confidential

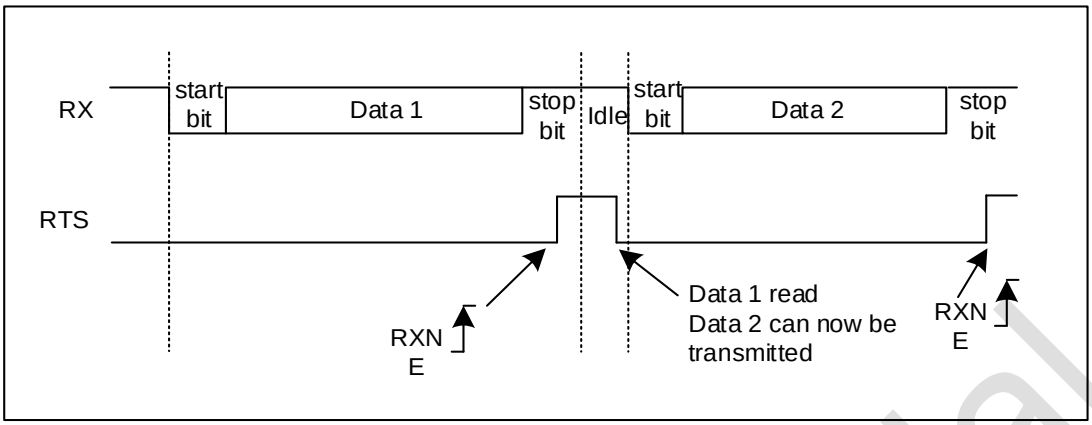


图 28-24 RTS 流控制

28.3.22.2 CTS 流控制

如果 CTS 流控制被使能（CTSE=1），发送器在发送下一帧前检查 CTS 输入。如果 CTS 有效（被拉成低电平），则下一个数据被发送（假设那个数据是准备发送的，也就是 TXE/TXFE=0），否则下一帧数据不被发出去。若 CTS 在传输期间被变成无效，当前的传输完成后停止发送。

当 CTSE=1 时，只要 CTS 输入一变换状态，硬件就自动设置 CTSIF 状态位。它表明接收器是否准备好进行通信。如果设置了 USART_CT3 寄存器的 CTSIE 位，则产生中断。

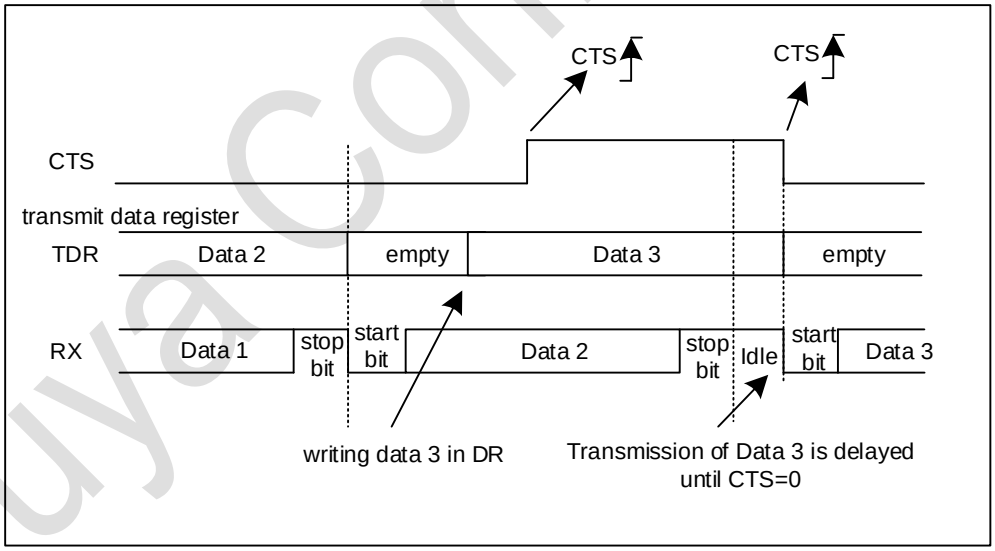


图 28-25 CTS 流控制

注意：为正常运行，必须在当前字符结束前至少 3 个 USART 时钟源周期内将 CTS 拉低。此外还必须注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。

28.3.22.3 RS485 驱动器使能

驱动器使能功能可通过将 USART_CR3 控制寄存器中的 DEM 位置 1 使能。这样用户便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与 START 位开始间的时间。可以通过

Puya Confidential

USART_CR3 控制寄存器中的 DEAT[4:0]位域编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 USART_CR1 控制寄存器中的 DEDT [4:0]位域编程禁止时间。DE 信号的极性可使用 USART_CR3 控制寄存器中的 DEP 位配置。

在 USART 中，DEAT 和 DEDT 以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）

28.4 USART 中断

表 28-6 USART 中断请求

中断向量	标志位	使能位
USART	txe	txeie
	cts	ctsie
	tc	tcie
	rxne	rxneie
	idle	idleie
	pe	peie
	lbd	lbdie
	ne/ore/fe	eie
	txft	txftie
	rxft	rxftie
	rxff	rxffie
	txfe	txfeie
	rtof	rtofie
	eobf	eobfie

28.5 USART 寄存器

28.5.1 USART 状态寄存器（USART_SR)

偏移地址:0x00

复位值:0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WAIT_RPT	TXBUSY	RXBUSY	TXPE	RTOF	EOBF	Res.	TXFE
								R	R	R	RC_W0	RC_W0	RC_W0		R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFF	RXFT	TXFT	ABRRQ	ABRE	ABRF	CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
R	R	R	W	R	R	RC_W0	RC_W0	R	RC_W0	RC_W0	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	-	保留
23	WAIT_RPT	R	0	该寄存器适用于智能卡模式 发送了 NACK 信号，正在等待对方重发数据; 状态机进入发送错误信号状态时置位，收到数据起始位或者进入发送状态时硬件清零; 软件可以查询此位，但是不能改写;
22	TXBUSY	R	0	发送数据忙标志。（发送完成后自动清零） 0：数据发送空闲 1：处于数据发送状态，发送移位寄存器正在发送数据

Puya Confidential

21	RXBUSY	R	0	接收数据忙标志。（接收完成后自动清零） 0：数据接收空闲 1：处于数据接收状态，接收移位寄存器正在接收数据
20	TXPE	RC_W0	0	该寄存器适用于智能卡模式。 发送数据奇偶校验错误标志位。 0：发送数据奇偶校验正确 1：发送数据奇偶校验错误 该位硬件置位，写 0 清零
19	RTOF	RC_W0	0	接收超时标志 已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，此位由硬件置 1。 如果 USART_CR2 寄存器中 RTOIE=1，则会生成中断。 在智能卡模式下，该超时对应于 CWT 或 BWT 时间。 0：未达到超时值 1：已达到超时值，未接收到任何数据 注：如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。 即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 将置 1。 如果 USART 不支持接收器超时功能，该位保留且保持复位值。
18	EOBF	RC_W0	0	块结束标志 接收到完整块后，此位由硬件置 1（例如 T=1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。 0：未达到块结束 1：已达到块结束（字符数） 注：如果不支持智能卡模式，该位保留且保持复位值。
17	Reserved	-	-	保留
16	TXFE	R	0	TXFIFO 为空 当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 TXFRQ 写入 1 将 TXFE 标志置 1。 0：TXFIFO 非空 1：TXFIFO 为空
15	RXFF	R	0	RXFIFO 已满 当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + USART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。 如果 RXFFIE 位为 1，则会生成中断 0：RXFIFO 未满 1：RXFIFO 已满
14	RXFT	R	0	RXFIFO 阈值标志

Puya Confidential

				达到在 RXFTCFG 中编程的阈值时，该位由硬件置 1。这意味着，接收 FIFO 中有 (RXFTCFG - 1) 个数据，USART_RDR 寄存器中有一个数据。 如果 RXFTIE 位为 1，则会生成中断。 0：接收 FIFO 未达到编程的阈值 1：接收 FIFO 已达到编程的阈值
13	TXFT	R	0	TXFIFO 阈值标志 当 TXFIFO 达到在 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个位置），该位由硬件置 1。 如果 TXFTIE 位为 1，则会生成中断。 0： TXFIFO 未达到编程的阈值 1： TXFIFO 已达到编程的阈值
12	ABRRQ	W	0	自动波特率请求。 该位写 1 会复位 ABRF 标志位，并且请求下一帧的自动波特率检测。
11	ABRE	R	0	自动波特率错误标志 当自动波特率检测出错（波特率超出范围或者字符比较错误）时，硬件置位该寄存器。 软件通过写 1 到 ABRRQ 寄存器清零该位。
10	ABRF	R	0	自动波特率检测标志 当自动波特率设置（同时设置 RXNE=1，当中断使能后产生中断），或者自动波特率检测操作出错（ABRE=1，RXNE=1，FE=1）时该位由硬件置 1。 软件通过写 1 到 ABRRQ 位清零该位。
9	CTS	RC_W0	0	CTS 标志 如果配置 CTSE=1，则当 CTS 输入状态变化，该寄存器为 1。软件写 0 清零。当 CTSIE=1 时，产生 CTS 中断。 0：CTS 线上的值未改变 1：CTS 线上的值改变
8	LBD	RC_W0	0	LIN 断开帧检测标志 当检测到 LIN 断开时，硬件配置该寄存器。软件写 0 清零。当 LBDIE=1 时产生中断。 0：未检测到 LIN 断开帧 1：检测到 LIN 断开帧 注：该寄存器如果不支持 LIN 功能，则固定为 0。
7	TXE	R	1	传输寄存器空标志 发送时，当 USART_DR 寄存器数据传送到移位寄存器，硬件置位该寄存器。当 TXEIE=1 时，产生中断。写 USART_DR 寄存器会清零该位。 0：数据未传送到移位寄存器 1：数据传送到移位寄存器
6	TC	RC_W0	1	传送完成标志 传送数据帧完成后，且 TXE=1，则硬件置位该寄存器。TCIE=1 时产生中断。

Puya Confidential

				软件写 USART_DR 寄存器会清零该位（针对多处理器通讯）。软件同时可以写 0 清零。 0：传送未完成 1：传送完成
5	RXNE	RC_W0	0	读数据寄存器不空标志 在非 FIFO 模式时，当移位寄存器值传送到 USART_RDR 寄存器，硬件置位该寄存器。 软件读 USART_DR 寄存器、或者写 0 清零该位。 当 RXNEIE=1 时，产生中断。 在 FIFO 模式时，RXFIFO 非空，RXNE 位由硬件置位。每次对 USART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。RXNE 在 RXFIFO 为空时清零。也可以通过向 USART_CR1 寄存器中的 RXFRQ 位写入 1 将 RXNE 标志位清零。 0：未收到数据 1：接收数据准备好
4	IDLE	R	0	空闲标志。 检测空闲帧，硬件置位该寄存器。当 IDLEIE=1 时产生中断。 软件先读 USART_SR 寄存器后读 USART_DR 寄存器可以清零该位。 0：未检测到空闲帧 1：检测到空闲帧
3	ORE	R	0	溢出错误标志 当 RXNE=1 时，在移位寄存器中接收到的数据正准备转移到 RDR 寄存器时，硬件设置该位。 软件先读 USART_SR 寄存器后读 USART_DR 寄存器可以清零该位。 当 RXNEIE=1 时，产生中断。 0：未产生溢出错误 1：产生溢出错误 注：该寄存器置位时，接收 DR 寄存器内容不会丢失，但移位寄存器内容被覆盖。 当 EIE=1 时，产生 ORE 中断。
2	NE	R	0	噪声错误标志 在数据帧接收到噪声时，硬件置位该寄存器。 软件先读 USART_SR 寄存器后读 USART_DR 寄存器可以清零该位。 0：未检测到噪声错误 1：检测到噪声错误 注：当 RXNE 与 NE 同时产生时，NE=1 时不产生中断，而在设置 RXNE 标志时产生中断。在多缓冲器通讯模式下，当 EIE=1 时 NE=1 会产生中断。
1	FE	R	0	帧错误标志

Puya Confidential

				当检测到不同步、过多的噪声或断开字符时，由硬件设置此位。 软件先读 USART_SR 寄存器后读 USART_DR 寄存器可以清零该位。 0：未检测到帧错误 1：检测到帧错误或断开字符 注：当 RXNE 与 FE 同时产生时，FE=1 时不产生中断，而在设置 RXNE 标志时产生中断。如果当前传输的数据既产生了帧错误，又产生了过载错误，硬件还是会继续该数据的传输，并且只设置 ORE 标志位。 在多缓冲器通讯模式下，当 EIE=1 时 FE=1 会产生中断。
0	PE	R	0	校验值错误 当接收时校验值错误时，硬件置位该寄存器。 软件先读 USART_SR 寄存器后读 USART_DR 寄存器可以清零该位。但软件在清该位前必须等待 RXNE=1. 当 PEIE 时，产生中断。 0：未产生奇偶校验错误 1：产生奇偶校验错误

28.5.2 USART 数据寄存器（USART_DR）

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[8:0]								
							RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8:0	DR[8:0]	RW	9’h0	接收/发送数据寄存器 取决于读还是写操作，前者是接收到的数据，后者是发送的数据。 DR 寄存器物理上由两个寄存器组成（一个是发送的 TDR，一个是接收的 RDR），所以 DR 寄存器实现了读和写的两个功能。 TDR 寄存器在内部总线和输出移位寄存器之间提供了并行的接口，RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口。 当奇偶校验使能打开进行发送操作时，写 MSB 位（bit7 或者 bit8）是无效的，因为已被校验位代替了。 当奇偶校验使能打开进行接收操作时，读出的 MSB 位是被接收到的校验位。

28.5.3 USART 波特率寄存器（USART_BRR）

偏移地址:0x08

复位值:0x0000_0000

Puya Confidential

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_MAN[11:0]												DIV_FRA[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

在自动波特率检测模式，硬件更新该寄存器。

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:4	DIV_MAN[15:4]	RW	12'h0	12 位整数部分波特率
3:0	DIV_FRA[3:0]	RW	4'h0	4 位小数部分波特率

28.5.4 USART 控制寄存器 1（USART_CR1）

偏移地址:0x0C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOEN	TXFRQ	RXFRQ	Res.	DATAINV	TX_INV	RX_INV	RXFTCFG[2:0]			TXFTCFG[2:0]			SWAP	ADDM7	Res.
RW	W	W		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB	M1	UE	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31	FIFOEN	RW	0	FIFO 模式使能 0：FIFO 模式关闭 1：FIFO 模式打开 FIFO 模式可用于标准 UART 通信，SPI 主/从模式和智能卡模式。它不能在 IrDA 和 LIN 模式下启用. 该位只能在 USART 被禁用（UE = 0）时写入。
30	TXFRQ	W	0	发送数据刷新请求 禁止 FIFO 模式时，向该位写入"1"会将 TXE 标志置 1。这可丢弃发送数据。由于错误（NACK）而未发送数据且 USART_SR 寄存器中的 FE 标志有效时，只能在智能卡模式下使用此位。如果 USART 不支持智能卡模式，该位保留且必须保持复位值。 使能 FIFO 时，TXFRQ 位置 1 以清空整个 FIFO。这会将 TXFE 标志置位。在 UART 模式和智能卡模式下都支持清空发送 FIFO。 注意：该位软件写 1，硬件清零
29	RXFRQ	W	0	接收数据刷新请求 写 1 来清除整个接收的 FIFO。 该位置一是不读数据但是会丢弃所有 FIFO 接收数据，避免出现过载情况 注意：该位软件写 1，硬件清零
28	Reserved		-	保留
27	DATAINV	RW	0	二进制数据翻转。 0：1 为高电平，0 为低电平 1：1 为低电平，0 为高电平

Puya Confidential

				注：奇偶校验位同样按此定义翻转。
26	TX_INV	RW	0	TX 引脚电平反转 该位只能在 USART 被禁用(UE = 0)时写入。 该位由软件设置和清除。 0：TX 引脚信号工作使用标准逻辑电平 1：TX 引脚信号值取反 注意：在 IRDA 模式、智能卡模式、自动波特率模式下不支持 TX 引脚互换
25	RX_INV	RW	0	RX 引脚电平反转 该位只能在 USART 被禁用(UE = 0)时写入。 该位由软件设置和清除 0：RX 引脚信号使用标准逻辑电平工作 1：RX 引脚信号值取反 注意：在 IRDA 模式、智能卡模式、自动波特率模式下不支持 RX 引脚互换
24:22	RXFTCFG	RW	3'h0	RXFIFO 阈值配置 000：接收 FIFO 至少接收了 1/8 空间的数据 001：接收 FIFO 至少接收了 2/8 空间的数据 010：接收 FIFO 至少接收了 4/8 空间的数据 011：接收 FIFO 至少接收了 6/8 空间的数据 100：接收 FIFO 至少接收了 7/8 空间的数据 101：接收 FIFO 满 其他：保留
21:19	TXFTCFG	RW	3'h0	TXFIFO 阈值配置 000：TXFIFO 至少有 1/8 的空间可以写数据 001：TXFIFO 至少有 2/8 的空间可以写数据 010：TXFIFO 至少有 4/8 的空间可以写数据 011：TXFIFO 至少有 6/8 的空间可以写数据 100：TXFIFO 至少有 7/8 的空间可以写数据 101：TXFIFO 变空 其他：保留
18	SWAP	RW	0	交换 TX/RX 引脚 0：按标准引脚排列定义使用 TX/RX 引脚 1：交换 TX 和 RX 引脚功能。这使得可以在与另一个 UART 交叉连接的情况下工作。 该位只能在 USART 被禁用（UE = 0）时写入。
17	ADDM7	RW	0	7 位/4 位地址检测 0：4 位地址检测 1：7 位地址检测（8 位数据模式下） 该位只能在 USART 被禁用（UE = 0）时写入。
16	Reserved		-	保留
15	MSB	RW	0	高位有效在前 0：在起始位之后，首先使用数据位 0 发送/接收数据。 1：在起始位之后，首先使用 MSB(位 7/8)发送/接收数据。 该位域只能在 USART 被禁用(UE = 0)时写入。

Puya Confidential

14	M1	RW	0	该位和第 12 位 M0 组成 M 决定数据大小 00：8 位 01：9 位 10：7 位 11：保留 该位只能在 USART 未使能时配置（UE=0）
13	UE	RW	0	USART 使能 当该位清零后，USART 模块会立即停止当前操作。该位由软件置位和清零。 0：USART 禁止 1：USART 使能 注意：在清零 UE 位之前 DMA 通道需要禁止。
12	M0	RW	0	M0：字长 该位定义了数据字的长度，由软件对其设置和清零。 该位只能在 USART 未使能时配置(UE=0)。
11	WAKE	RW	0	唤醒方式 0：被空闲总线唤醒 1：被地址标记唤醒 该位只能在 USART 未使能时配置(UE=0)
10	PCE	RW	0	奇偶校验控制 0：禁止奇偶校验 1：使能奇偶校验 该位只能在 USART 未使能时配置(UE=0)
9	PS	RW	0	奇偶校验选择 0：偶校验 1：奇校验 该位只能在 USART 未使能时配置(UE=0)
8	PEIE	RW	0	PE 中断使能 0：禁止 PE 中断 1：使能 PE 中断
7	TXEIE	RW	0	TXE 中断使能 0：禁止 TXE 中断 1：使能 TXE 中断
6	TCIE	RW	0	TC 中断使能 0：禁止 TC 中断 1：使能 TC 中断
5	RXNEIE	RW	0	RXNE 中断使能 0：禁止 RXNE 中断 1：当 USART_SR 中的 ORE 或者 RXNE 为 1 时，产生 USART 中断
4	IDLEIE	RW	0	IDLE 中断使能 0：禁止 IDLE 中断 1：使能 IDLE 中断
3	TE	RW	0	发送使能 0：禁止发送

Puya Confidential

				1：使能发送 注意：在数据传输过程中，除了在智能卡模式下，如果 TE 位上有个 0 脉冲(即设置为'0'之后，再设置为'1')，会在当前数据字传输完成后，发送一个空闲帧。当 TE 被设置后，在真正发送开始之前，有一位时间的延迟。
2	RE	RW	0	接收使能 0：禁止接收 1：使能接收
1	RWU	RW	0	接收唤醒 该位表明 USART 是否为静默模式。 当接收到静默模式序列，该寄存器置位；如果接收到唤醒序列，该寄存器清零。具体哪种唤醒序列（地址或者空闲帧）由寄存器 USART_CR1.WAKE 位控制。 0：接收器为工作模式 1：接收器为静默模式 注意 1：在设置该位进入静默模式前，USART 要已经先接收了一个数据字节，否则在静默模式下，不能被空闲总线检测唤醒。 2：当配置成地址标志检测唤醒（WAKE=1），在 RXNE 被置位时，不能用软件修改 RWU 位。
0	SBK	RW	0	发送断开帧 软件置位该寄存器，发送断开字节。断开帧的停止位发送后，硬件清零该寄存器。 0：不发送断开字节 1：发送断开字节

28.5.5 USART 控制寄存器 2（USART_CR2）

偏移地址:0x10

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RREPEN	TREPEN	REP_T	ERSW[1:0]		RTOEN	DEP	DEM	ADD[7:1]						
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADD[0]	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	Res.	Res.	Res.	Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW		RW	RW					

Bit	Name	RW	Reset Value	Function
31	Reserved	-	-	保留
30	RREPEN	RW	0	该寄存器适用于智能卡模式 接收数据奇偶校验错的处理方式选择 1：奇偶校验错，根据 T=0 协议自动回发 NACK。单一 BYTE 连续接收次数超过 REP_T 后，置 PE 标志，进行中断 0：奇偶校验错，不自动发送 NACK，置 PE 标志，进行中断 该位只能在 SCEN= 1 时写入。
29	TREPEN	RW	0	该寄存器适用于智能卡模式 发送数据奇偶校验错的处理方式选择。

Puya Confidential

				1：收到奇偶校验出错标志（NACK），根据 T = 0 协议自动进行回发。在单一 byte 重复发送次数超过 REP_T 后，硬件置 TXPE 标志，若开启 PEIE，则产生中断 0：收到 NACK 时不进行自动回发，硬件置 TXPE 标志，若开启 PEI，则产生中断 该位只能在 SCEN= 1 时写入。
28	REP_T	RW	0	该寄存器适用于智能卡模式 控制接收数据奇偶校验出错时自动重发次数 1：3 次 0：1 次 该位只能在 SCEN= 1 时写入。
27:26	ERSW	RW	2’h0	该寄存器适用于智能卡模式。 NACK 宽度选择 11：NACK 宽度为 1 bit 10：NACK 宽度为 1.5 bit 01：NACK 宽度为 2 bit 00：NACK 宽度为 2 bit 该位只能在 SCEN= 1 时写入。
25	RTOEN	RW	0	接收器超时使能 0：禁止接收器超时功能 1：使能接收器超时功能 使能此功能后，如果 RX 线路在 RTOR（接收器超时寄存器）中编程的持续时间内处于空闲状态（无接收），则 USART_ISR 寄存器中的 RTOF 标志置 1。
24	DEP	RW	0	驱动使能极性选择 0：DE 信号为高 1：DE 信号为低 该位只能在 USART 未使能时配置（UE=0）
23	DEM	RW	0	驱动使能模式 0：驱动禁止 1：驱动使能 该位只能在 USART 未使能时配置（UE=0）
22:15	ADD[7:0]	RW	8’h0	USART 地址 地址标记检测时的地址。 在 4 位地址检测时，只有 ADD[3:0]被使用 该位只能在 USART 未使能时配置（UE=0）
14	LINEN	RW	0	LIN 模式使能 0：LIN 模式禁止 1：LIN 模式使能 LIN 模式下，通过使能 SBK 位，发送 LIN 同步断开帧（13 低位）。 该位只能在 USART 未使能时配置（UE=0）
13:12	STOP[1:0]	RW	2’h0	停止位配置 00：1 个停止位 01：0.5 个停止位

Puya Confidential

				10：2 个停止位 11：1.5 个停止位 该位只能在 USART 未使能时配置（UE=0）
11	CLKEN	RW	0	时钟引脚使能 0：时钟引脚禁止 1：时钟引脚使能 该位只能在 USART 未使能时配置（UE=0）
10	CPOL	RW	0	时钟极性 0：总线空闲时 CK 引脚上保持低电平 1：总线空闲时 CK 引脚上保持高电平 该位只能在 USART 未使能时配置（UE=0）
9	CPHA	RW	0	时钟相位 0：在时钟的第一个边沿进行数据捕获； 1：在时钟的第二个边沿进行数据捕获。 该位只能在 USART 未使能时配置（UE=0）
8	LBCL	RW	0	最后一位数据的时钟脉冲是否在时钟引脚输出。 0：最后一位数据的时钟脉冲不在时钟引脚输出 1：最后一位数据的时钟脉冲在时钟引脚输出 该位只能在 USART 未使能时配置（UE=0）
7	Reserved	-	-	保留
6	LBDIE	RW	0	LIN 断开中断使能 0：禁止 LIN 断开中断 1：使能 LIN 断开中断
5	LBDL	RW	0	LIN 断开帧检测长度 0：10 位断开帧检测 1：11 位断开帧检测
4:0	Reserved	-	-	保留

28.5.6 USART 控制寄存器 3 (USART_CR3)

偏移地址:0x14

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]					TXFTIE	RXFTIE	TXFEIE
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFFIE	ABRMOD[1:0]		ABREN	OVER8	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31	Reserved	-	-	保留
30	EOBIE	RW	0	EOB 中断使能 0：禁止 EOB 中断 1：使能 EOB 中断
29	RTOIE	RW	0	RTO 中断使能 0：禁止 RTO 中断 1：使能 RTO 中断

Puya Confidential

28:24	DEAT[4:0]	RW	5'h0	起始位到 DE 拉高信号的时间 以采样时间表示（1/8 位或 1/16 位时间，这个取决于过采样配置） 该位只能在 USART 未使能时配置(UE=0)
23:19	DEDT[4:0]	RW	5'h0	最后一个数据的停止位到 DE 的时间 以采样时间表示（1/8 位或 1/16 位时间，这个取决于过采样配置） 该位只能在 USART 未使能时配置(UE=0)
18	TXFTIE	RW	0	TXFIFO 阈值中断使能 0：禁止 TXFT 中断 1：使能 TXFT 中断
17	RXFTIE	RW	0	RXFIFO 阈值中断使能 0：禁止 RXFT 中断 1：使能 RXFT 中断
16	TXFEIE	RW	0	TXFIFO 空中断使能 0：禁止 TXFE 中断 1：使能 TXFE 中断
15	RXFFIE	RW	0	RXFIFO 满中断使能 0：禁止 RXFF 中断 1：使能 RXFF 中断
14:13	ABRMOD[1:0]	RW	2'h0	自动波特率检测模式 00：从起始位开始测量 01：下降沿到下降沿测量 10：0x7F 帧检测方式测量 11：0x55 帧检测方式测量 当 ABREN=0 或者 UE=0 时，才能配置该寄存器。
12	ABREN	RW	0	自动波特率使能 0：禁止自动波特率 1：使能自动波特率
11	OVER8	RW	0	过采样模式。 0：16 位过采样 1：8 位过采样 该位只能在 USART 未使能时配置(UE=0) 注意：在 LIN、IrDA 和智能卡模式下，该位必须保持清零。
10	CTSIE	RW	0	CTS 中断使能 0：禁止 CTS 中断 1：使能 CTS 中断
9	CTSE	RW	0	CTS 使能 0：禁止 CTS 硬件流控 1：使能 CTS 硬件流控 该位只能在 USART 未使能时配置(UE=0)
8	RTSE	RW	0	RTS 使能 0：禁止 RTS 硬件流控 1：使能 RTS 硬件流控 该位只能在 USART 未使能时配置(UE=0)

Puya Confidential

7	DMAT	RW	0	发送时使能 DMA 0：发送时禁止 DMA 1：发送时使能 DMA
6	DMAR	RW	0	接收时使能 DMA 0：接收时禁止 DMA 1：接收时使能 DMA
5	SCEN	RW	0	智能卡模式使能 0：禁止智能卡模式 1：使能智能卡模式 该位只能在 USART 未使能时配置(UE=0)
4	NACK	RW	0	智能卡 NACK 使能 0：禁止奇偶校验错误时发送 NACK； 1：使能奇偶校验错误时发送 NACK； 注：该寄存器如果不支持智能卡功能，则固定为 0
3	HDSEL	RW	0	半双工选择 0：非半双工模式 1：半双工模式 该位只能在 USART 未使能时配置(UE=0)
2	IRLP	RW	0	IrDA 低功耗 0：正常模式 1：IrDA 低功耗模式 该位只能在 USART 未使能时配置(UE=0)
1	IREN	RW	0	IrDA 模式使能 0：禁止 IrDA 模式 1：使能 IrDA 模式 该位只能在 USART 未使能时配置(UE=0)
0	EIE	RW	0	错误中断使能 0：禁止错误中断 1：使能错误中断，包括帧错误 FE、上溢错误 ORE、噪声 NE

28.5.7 USART 保护时间和预分频寄存器 (USART_GTPR)

偏移地址:0x18

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT[7:0]								PSC[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:8	GT[7:0]	RW	8’h0	保护时间值 以波特时钟为单位的保护时间。 在智能卡模式，需要此功能。 注：该位只能在 USART 未使能时配置（UE=0）
7:0	PSC[7:0]	RW	8’h0	预分频器值

Puya Confidential

				在红外（IrDA）低功耗模式下： PSC[7:0] =红外低功耗波特率。 对系统时钟分频以获得低功耗模式下的频率： 源时钟被寄存器中的值（8 位有效）分频 00000000：Res； 00000001：对源时钟 1 分频； 00000010：对源时钟 2 分频； …… 1. 在红外(IrDA)正常模式下： PSC 只能设置为 00000001。 2. 在智能卡模式下： PSC[4:0]：预分频值 对系统时钟进行分频，给智能卡提供时钟。 寄存器中给出的值(低 5 位有效)乘以 2 后，作为对源时钟的分频因子。 00000：Res； 00001：对源时钟进行 2 分频； 00010：对源时钟进行 4 分频； 00011：对源时钟进行 6 分频； …… 注：位[7:5]在智能卡模式下没有意义。 注：该位只能在 USART 未使能时配置（UE=0）

28.5.8 USART 接收超时寄存器（USART_RTOR）

偏移地址:0x1C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLEN[7:0]								RTO[23:16]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTO[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:24	BLEN	RW	8’h0	块长度 该寄存器定义智能卡模式下 T=1 接收的块长度。 它的值等于 information 字符+Epilogue 长度-1 例如： BLEN =0：0 information 字符+LEC BLEN =1：0 information 字符+CRC BLEN = 255:254 个 information 字符+CRC 在智能卡模式，当 TXE=0 时块长度计算被复位。 这个位域也可以在其他模式中使用。在这种情况下，当 RE = 0(接收器禁用)和/或 EOBF 位被写入 0 时，块长度计数器被重置。
23:0	RTO	RW	24’h0	在 RX 线上没有活动的接收超时时间 在标准模式下，在设定的时间下，在最后接收达到的字符后没有新起始位检测被 RTO 检测，RTOF 标志位置起。

Puya Confidential

				在智能卡模下，这个值用来执行 CWT 和 BWT。在这个情况下，CWT 和 BWT 是从最后接收到的字符起始位开始计算的

Puya Confidential

29. 通用异步收发器（UART）

29.1 UART 简介

本项目设计实现了 2 个 UART 模块，2 个功能完全一致。

UART 是一种可编程通用异步收发器。

29.2 UART 主要特性

- AMBA APB 接口
- 支持 5/6/7/8/9 位串行数据
- 支持 1/2 位 STOP 位（5 位数据时：1/1.5 位 STOP）
- 支持发送地址/数据
- 支持固定奇偶校验
- 支持断开帧
- 起始位错误检测
- 支持可编程分数波特率：可编程串行数据波特率，计算如下：波特率=（串行时钟频率）/（16*除数）
- 支持 4 位小数波特率
- 支持 Tx/Rx 引脚互换功能
- 支持大小端切换 MSB FIRST 功能
- 支持 DMA 传输

Puya Confidential

29.3 UART 功能描述

29.3.1 UART 框图

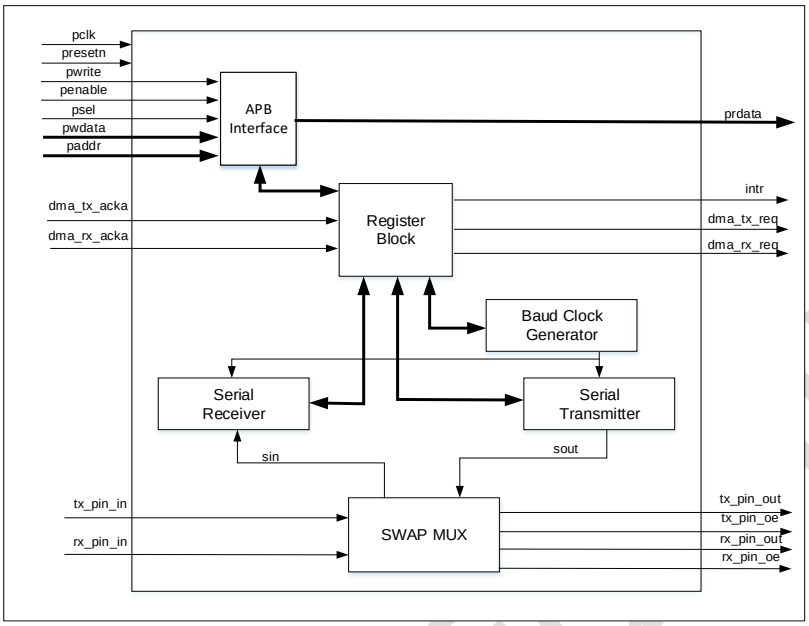


图 29-1 UART 框图

注：1.粗体线表示存在且为总线。

主要模块：

- APB 从接口（APB interface）---APB 总线接口。
- 寄存器块（Register Block）---负责 UART 的主要功能，包括控制、状态和中断生成。
- 波特时钟生成器（Baud Clock Generator）---生成发送器和接收器波特时钟以及输出参考时钟信号。
- 串行发送器（Serial Transmitter）---将写入 UART 的并行数据转换为串行形式，并添加控制寄存器指定的所有附加位，用于传输。
- 串行接收器（Serial Receiver）---将 UART 中接收的控制寄存器指定的串行数据字符转换为并行形式。此块控制：
  - 奇偶校验错误检测
  - 帧错误检测
  - 断开帧检测
- 引脚互换处理（SWAP MUX）---SWAP 使能时的引脚处理

29.3.2 UART（RS232）串行协议

因为 UART 和所选设备之间的串行通信是异步的，所以在串行数据中添加了额外的位（起始和停止）来指示开始和结束。利用这些比特可以使两个设备同步。这种由起始位和停止位组成的串行数据结构被称为一个字符，如下图所示。

Puya Confidential

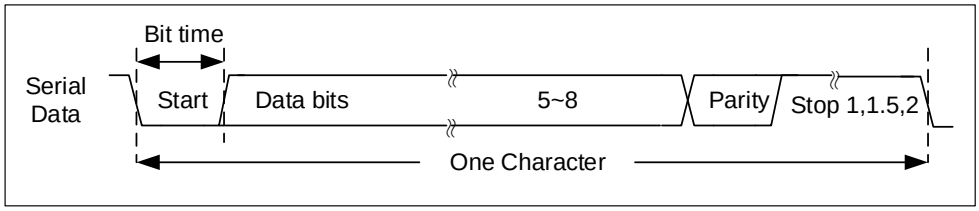


图 29-2 串行数据格式

一个额外的奇偶校验位可以添加到串行字符。该位出现在字符结构中的最后一个数据位之后和停止位之前，以便为 UART 提供对接收到的数据执行简单错误检查的能力。

UART 线路控制寄存器用于控制串行字符特性。数据字的各个位在起始位之后发送，从最低有效位（LSB）开始。它们后面是可选的奇偶校验位，后面是停止位，可以是 1、1.5 或 2。

注：UART 实现的停止位持续时间可能会更长，原因如下：

■ 在某些配置的字符之间插入的空闲时间

传输中的所有比特都是在完全相同的持续时间内传输的；这方面的例外是当使用 1.5 个停止位时的半停止位。该持续时间被称为比特周期或比特时间；一位时间等于十六个波特时钟。

为了确保线路的稳定性，一旦检测到起始位，接收器就在位时间的大约中点对串行输入数据进行采样。因为每个比特传输的波特时钟的确切数量是已知的，所以计算采样的中点并不困难；即在起始位的中点采样之后每十六个波特时钟。

与串行输入去抖动一起，这种采样有助于避免错误起始位的检测。短故障通过去抖动被过滤掉，并且在线路上没有检测到转换。如果故障足够宽，可以通过去抖动来避免滤波，则会检测到下降沿。然而，只有当线路在半个采样周期后再次被采样为低时，才检测到起始位。

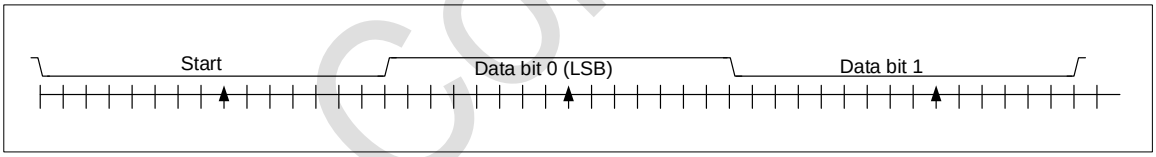


图 29-3 接收器串行数据采样点

作为 16550 标准的一部分，一个可选的波特时钟参考输出信号（baudout_n）为需要它的接收设备提供定时信息。UART 的波特率由单时钟实现中的串行时钟 pclk 以及分频锁存寄存器（BRR）控制。

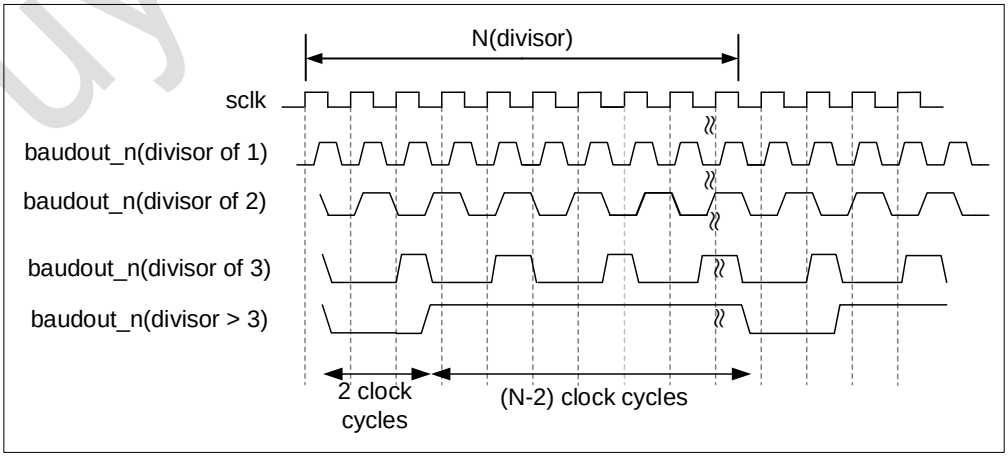


图 29-4 不同除数值的 baudout_n 输出的时序图

Puya Confidential

29.3.3 UART 9 位数据传输

UART 在发送和接收模式下都可以被配置为具有 9 位数据传输。字符中的第 9 位出现在字符的第 8 位之后和奇偶校验位之前。下图显示了字符的串行传输，其中 D8 表示第 9 位，还显示了 9 位模式下的一般串行传输。（此为正常小端模式，如果是大端模式则调转 9 位数据位的顺序）

9 位数据

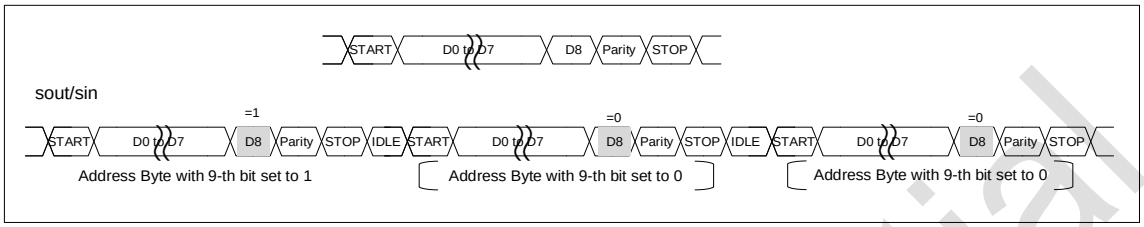


图 29-5 9 位字符

通过启用 9 位数据传输模式，UART 可以用于多点系统，其中一个主机连接到系统中的多个从机。主机与其中一个从机交流。当主机想要将数据块传输到从机时，它首先发送一个地址字节来识别目标从机。地址/数据字节之间的区分是基于输入字符中的第 9 位来完成的。如果第 9 位设置为 0，则该字符表示一个数据字节。如果第 9 位设置为 1，则该字符表示地址字节。所有从系统将地址字节与它们自己的地址进行比较，并且只有目标从机（其中地址匹配）能够从主机接收数据。主机开始向目标从机发送数据字节。未寻址的从机忽略传入的数据，直到接收到新的地址字节。

在上图中，请注意一个地址后面跟着 2 个数据字节。地址字节在第 9 位（D8）设置为 1 的情况下输出，而数据字节在第九位（D8）设置为 0 的情况下发出。奇偶校验位是一个可选字段。

用于 9 位数据传输的 UART 的配置执行以下操作：

- UART_CR3.M_E 位用于启用或禁用 9 位数据传输。
- 在接收的情况下，UART_CR3.ADDR_MATCH 用于在基于硬件和软件的地址匹配之间进行选择。
- UART_CR3.SEND_ADDR 位用于在发送的情况下使能发送地址。
- UART_CR3.TX_MODE 位用于在基于硬件和软件的地址传输之间进行选择。
- UART_TAR 和 UART_RAR 寄存器分别用于发送地址和匹配接收到的地址。
- UART_DR（TDR/RDR）寄存器为 9 位寄存器，用于在 9 位模式下进行数据传输。
- UART_SR.ADDR_RCVD 位用于指示地址接收中断。

29.3.3.1 发送模式

UART 支持两种类型的传输模式：

- 传输模式 0（当 UART_CR3.TX_MODE 设置为 0 时）
- 传输模式 1（当 UART_CR3.TX_MODE 设置为 1 时）

模式 0

在传输模式 0 中，地址被编程在传输地址寄存器（TAR）中，数据被写入传输保持寄存器（TDR）。TDR 寄存器的第 9 位在此模式中不适用。

下图说明了基于 SEND_ADDR（CR3[2]）、TDR 空条件的地址和数据传输。

Puya Confidential

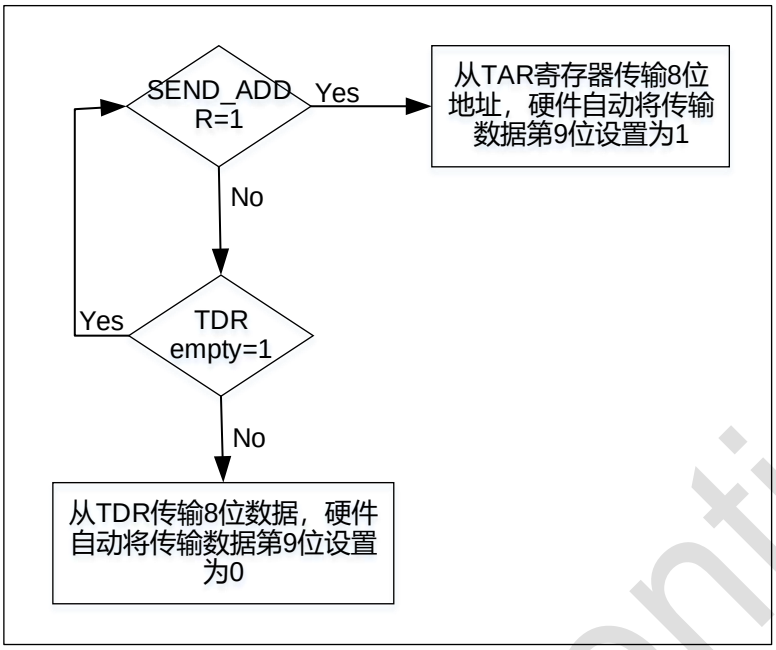


图 29-6 自动地址传输流程图

要向其传输数据的目标从机地址被编程在 TAR 寄存器中。必须启用 SEND_ADDR（CR3[2]）位来传输串行 UART 线上 TAR 寄存器中的目标从机地址，其中第 9 个数据位设置为 1，表示正在发送地址到从机。地址字符开始在线上传输后，UART 清除 SEND_ADDR 位。

传输到目标从机所需的数据通过传输保持寄存器（TDR）进行编程。数据在 UART 线上传输，第 9 个数据位设置为 0，表示正在发送数据到从机。

模式 1

在传输模式 1 中，TDR 寄存器为 9 位宽，地址和数据都通过 TDR 寄存器编程。UART 不区分地址和数据。SEND_ADDR（UART_CR3[2]）位和传输地址寄存器（UART_TAR）不适用于此模式。根据是否必须发送地址/数据，软件必须用 1/0 写入第 9 位。

M_E	TX_MODE	SEND_ADDR	使用场景
0	X	X	发 8 位 TDR 发数据
1	0	0	发“0+8 位 TDR 数据“
1	0	1	发“1+8 位 TAR 地址”
1	1	X	发 “9 位 TDR 数据”

注：该表说明作为发送方可以使用的几种发送场景和对应的配置。

29.3.3.2 接收模式

UART 支持两种接收模式：

- 硬件地址匹配接收模式（当 ADDR_MATCH（UART_CR3[1]）设置为 1 时）
- 软件地址匹配接收模式（当 ADDR_MATCH（UART_CR3[1]）设置为 0 时）

硬件地址匹配接收模式

在硬件地址匹配接收模式中，如果接收字符的第 9 位设置为 1，则 UART 将接收字符与在接收地址寄存器（UART_RAR）中编程的地址进行匹配：

如果接收到的地址与 UART_RAR 寄存器中的编程地址匹配成功，则随后接收数据字节。

Puya Confidential

如果地址匹配失败，则 UART 控制器丢弃数据字符，直到接收到匹配的地址为止。

下图显示了基于地址匹配功能的数据字节接收流程图。

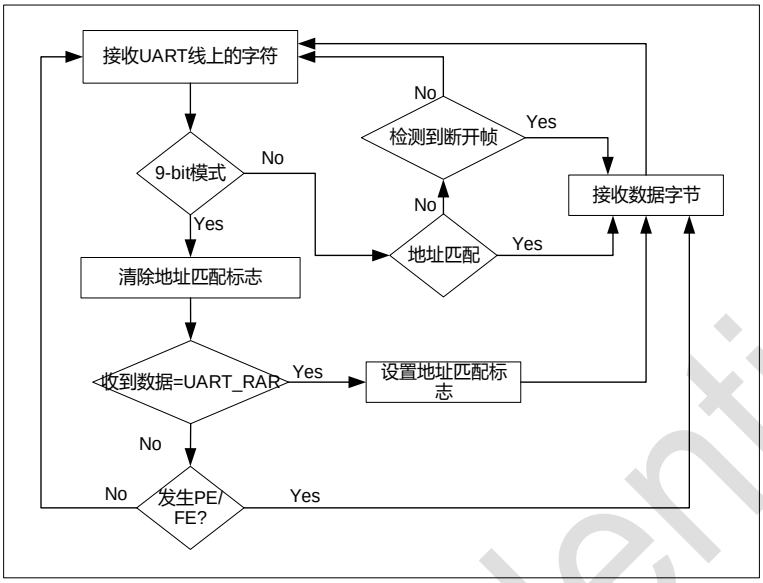


图 29-7 硬件地址匹配接收模式

UART 接收字符，而不管第 9 位数据是否设置为 1。如果接收到的字符的第 9 位被设置为 1，则它清除内部地址匹配标志，然后将接收到的 8 位字符信息与 UART_RAR 寄存器中编程的地址进行比较。

如果接收到的地址字符与 UART_RAR 寄存器中编程的地址匹配成功，则地址匹配标志被设置为 1，并且接收到的字符推送到 UART_RDR 寄存器，并且 UART_SR 寄存器中的 ADDR_RCVD 位被设置为指示地址已被接收，随后的数据字节（接收字符的第 9 位设置为 0）被推送到 UART_RDR。

如果地址与 UART_RAR 寄存器匹配失败并且(奇偶校验或者接收的地址字符中发现帧错误)的情况下，则接收的地址字符串仍然被推送到 UART_RDR 寄存器，ADDR_RCVD 和 PE/FE 错误位被设置为 1。

如果接收到任何中断字符，UART 将其视为一个特殊字符，并将其推送到 UART_RDR 寄存器，而不考虑地址匹配标志。

软件地址匹配接收模式

在这种操作模式中，UART 不执行与 UART_RAR 寄存器的接收地址字符（第 9 位数据设置为 1）的地址匹配。UART 始终接收 9 位数据，移位进 UART_RDR 寄存器。每当接收到地址字节并通过状态寄存器中的 ADDR_RCVD 位指示时，必须由用户自行比较地址。

29.3.4 UART 波特率

UART 的波特率由 PCLK 和分频锁存寄存器（UART_BRR）和小数波特率寄存器 UART_BRRF 控制。

波特率由以下因素决定：

- 串行时钟工作频率（PCLK）
- 所需波特率
- 波特率生成器除数值 divisor（由 UART_BRR 寄存器组成）
- 可接受的波特率误差%error

计算波特率的方程式如下：

波特率=串行时钟工作频率/（16* DIVISOR）——（1）

Piya Confidential

其中，DIVISOR–用于对 BRR 进行编程的数字（十六进制）。

串行时钟频率–UART 的 PCLK 频率。

根据等式（1），DIVISOR 可以计算为：DIVISOR=串行时钟频率/（16*波特率）

（等式算出的小数部分放在 UART_BRRF）

同样从等式（1）中，还可以示出：串行时钟频率= 波特率* 16* 除数

波特率和选定波特率定之间的误差如下所示：

百分比错误=（ |波特率-选定波特率| ）/波特率* 100%

表 29-1 设置波特率时的误差计算

波特率		f_{PCLK} = 4MHz			f_{PCLK} = 24MHz			f_{PCLK} = 48MHz		
序号	kbps	实际	置于波特率寄存器中的值	误差%	实际	置于波特率寄存器中的值	误差%	实际	置于波特率寄存器中的值	误差%
1	2.4	2.404	104	0.16%	2.4	625	0.00%	2.4	1250	0.00%
2	9.6	9.615	26	0.16%	9.615	156	0.16%	9.615	312	0.16%
3	19.2	19.231	13	0.16%	19.231	78	0.16%	19.231	156	0.16%
4	57.6	62.5	4	8.51%	57.692	26	0.16%	57.692	52	0.16%
5	115.2	125	2	8.51%	115.385	13	0.16%	115.385	26	0.16%
6	230.4	250	1	8.51%	250	6	8.51%	230.769	13	0.16%
7	460.8	不可能	不可能	不可能	500	3	8.51%	500	6	8.51%
8	921.6	不可能	不可能	不可能	1500	1	62.76%	1000	3	8.51%
9	2250	不可能	不可能	不可能	不可能	不可能	不可能	3000	1	33.33%
10	4500	不可能	不可能	不可能	不可能	不可能	不可能	不可能	不可能	不可能

注:

1. CPU 的时钟频率越低，则某一特定波特率的误差也越低。可以达到的波特率上限可以由这组数据得到。

2. 在配置时钟的时候，由于波特率时钟是在系统时钟上进行分频的，但得到的时钟频率与实际时钟频率会有误差。配置的时钟误差需要在 2%以内，才可以正常工作。

29.3.5 UART 接收容忍度

只有当整体的时钟系统地变化小于 UART 异步接收器能够容忍的范围，UART 异步接收器才能正常地工作。影响这些变化的因素有：

- DTRA：由于发送器误差而产生的变化(包括发送器端振荡器的变化)
- DQUANT：接收器端波特率取整所产生的误差
- DREC：接收器端振荡器的变化
- DTCL：由于传输线路产生的变化(通常是由于收发器在由低变高的转换时序，与由高变低转换时序之间的不一致性所造成)。

需要满足：DTRA + DQUANT + DREC + DTCL < UART 接收器的容忍度

对于正常接收数据，UART 接收器的容忍度等于最大能容忍的变化为 96%~105%。

29.3.6 UART DMA 通信

UART 可以利用 DMA 连续通讯。Rx 缓冲器和 Tx 缓冲器的 DMA 请求分别产生。

Puya Confidential

29.3.6.1 DMA 发送

通过配置 UART_CR3.DMAT 使能利用 DMA 发送。当发送数据寄存器为空，DMA 就从指定的存储区传送数据到 UART_DR 寄存器。

DMA 发送的配置步骤如下：

- 配置 SYSCFG_CFGR3 或者 SYSCFG_CFGR4 选择 USART 使用的 DMA 通道
- 配置 UART_TDR 寄存器地址作为 DMA 传输的目标地址。在每个发送数据寄存器为空事件后，数据将被传送到这个地址；
- 配置存储器（如 SRAM）地址作为 DMA 传输的源地址。在每次发送数据寄存器为空事件后，将从此存储器读数据加载到 UART_TDR 寄存器；
- 写 DMA 控制寄存器配置传输字节数；
- 写 DMA 寄存器配置通道优先级；
- 使能 DMA 通道；
- 当传输达到配置的传输字节数，DMA 控制器产生中断请求。

29.3.6.2 DMA 接收

通过配置 UART_CR3.DMAR 使能利用 DMA 接收。每收到一个字节，DMA 就把数据从 UART_DR 寄存器传送到指定的 SRAM 区。

DMA 接收的配置步骤如下：

- 配置 SYSCFG_CFGR3 或者 SYSCFG_CFGR4 选择 USART 使用的 DMA 通道
- 配置 UART_RDR 寄存器地址作为 DMA 传输的源地址。在每个接收数据寄存器非空事件后，数据将从此 UART_RDR 寄存器地址读出数据传输到存储器；
- 配置 SRAM 存储器地址作为 DMA 传输的目标地址。在每次接收数据寄存器非空事件后，数据将从 UART_RDR 寄存器传送到此地址；
- 写 DMA 控制寄存器配置传输字节数；
- 写 DMA 寄存器配置通道优先级；
- 使能 DMA 通道；
- 当传输达到配置的传输字节数，DMA 控制器产生中断请求。

29.4 UART 中断

UART 中断输出信号（intr）的断言——只要几个优先中断类型中的一个被启用并激活，就会发生中断。

以下中断类型可以通过 UART_CR2 寄存器启用：

- 接收器数据可用（RXNEIE）
- 发送寄存器空（TXEIE）
- 发送保持寄存器为空（在可编程 TDR 中断模式下）（TDREIE）
- 忙错误检测指示（BUSYERRIE）
- 接收线路状态（LSIE）

Puya Confidential

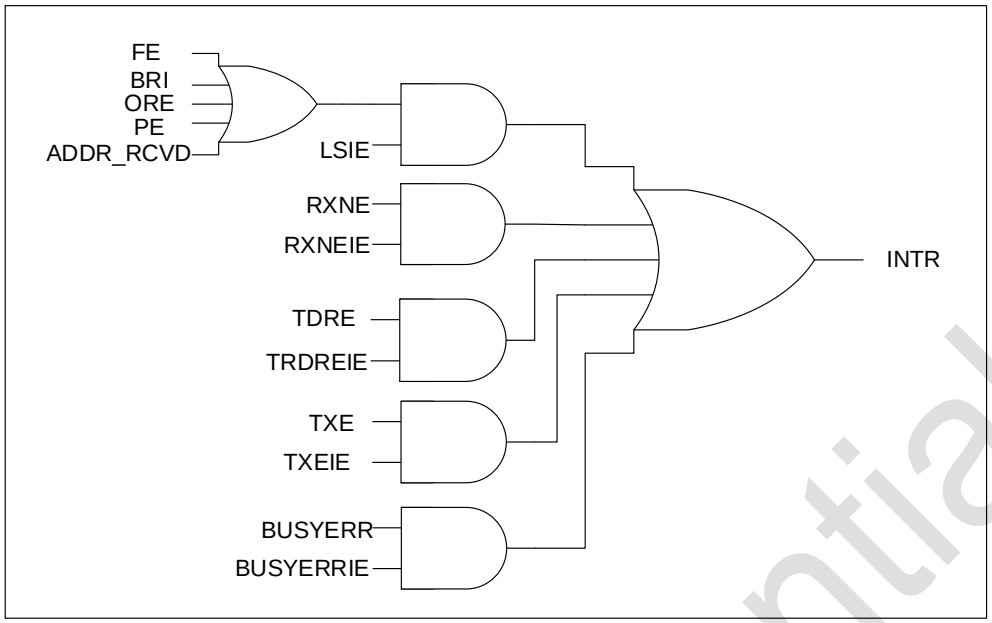


图 29-8 中断映射图

这些中断类型在下表中有详细说明。

表 29-2 中断控制功能

中断置位和清零功能		
中断类型	**中断源**	**中断清零控制**
接收线路状态	溢出/奇偶校验/帧错误中断或地址接收中断	对应位写 1 清 0
接收数据可用 RXNE	接收器数据可用	读取接收器缓冲寄存器
发送保持寄存器为空 TDRE	发送保持寄存器为空	写入 TDR
发送寄存器为空 TXE	发送寄存器为空	写入 TDR
忙错误检测 BUSYERR	UART 繁忙时（BUSY[0]设置为 1），主机尝试写入 CR1 寄存器。	对应位写 1 清 0

29.5 UART 寄存器

术语：设置=设置为 1；清除=清除为 0；

29.5.1 UART 数据寄存器（UART_DR）

地址偏移：0x00

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[8:0]								
							RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8:0	DR[8:0]	RW	9'b0	接收/发送数据寄存器。 一共是由两个寄存器组成（一个是发送的 TDR,一个是接收的 RDR），所以 DR 寄存器实现了读和写的两个功能。 RDR 寄存器是 UART 模式下串行输入端口上接收到的数据字节。只有状态寄存器（SR）中的接收非空（RXNE）位被设置时，该寄存器中的数据才有效。必须在下一个数据到达之前读取 RDR 中的数据，否则它将被覆盖，从而导致溢出错误。 TDR 寄存器是在 UART 模式下串行输出端口上传输的数据。软件保证只有当 TDR 空（UART_SR.TDRE）位被设置时，数据才能再次写入 TDR。 如果 TDRE 被设置，则向 TDR 写入单个字符将清除 TDRE。在再次设置 TDRE 之前对 TDR 的任何额外写入都会导致 TDR 数据被重写。 仅当 UART_CR3.TX_MODE = 1 时，第 9 位才适用。

29.5.2 UART 波特率寄存器（UART_BRR）

地址偏移：0x04

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	BRR	RW	16'b0	该寄存器构成 16 位 divisor，divisor 包含 UART 的波特率除数。输出波特率等于串行时钟（PCLK）频率除以波特率除数值的十六倍，如下所示：波特率=（串行时钟频率）/（16*divisor）。 请注意，当除数锁存寄存器（BRR）设置为零时，波特时钟被禁用，不会发生串行通信。 此外，一旦设置了 BRR，在发送或接收数据之前，应等待至少 8 个时钟周期。

29.5.3 UART 状态寄存器（UART_SR）

地址偏移：0x10

复位值：0x0000_0060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	BUSY_ERR	BUSY	ADDR_RCVD	Res.	TXE	TDRE	BRI	FE	PE	ORE	RXNE
					RC_W1	R	RC_W1		R	O	RC_W1	RC_W1	RC_W1	RC_W1	R

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:11	Reserved	-	-	保留
10	BUSY_ERR	RC_W1	0	忙检测错误，检测到忙时的误操作。 0：无忙误操作错误。 1：UART 繁忙时（UART_SR.BUSY 设置为 1），主机尝试写入 UART_CR1 寄存器。 该位写 1 清零。
9	BUSY	R	0	UART 忙 这表示串行传输正在进行中，清除时表示 UART 处于空闲或非活动状态。 在以下任何一种情况下，此位都将设置为 1（忙）： ---串行接口上正在进行发送 ---串行接口上正在进行接收 ---传输 TDR 中存在的数据，波特除数为非零（BRR 不等于 0） ---接收 RDR 中存在数据 注意：即使可能从另一个设备发送了新字符，UART 忙位也可能被清除。也就是说，如果 UART 在 TDR 和 RDR 中没有数据，并且没有正在进行的传输，并且新字符的起始位刚刚到达 UART。这是由于直到位周期的中间才看到有效的开始，并且该持续时间取决于已编程的波特除数。 0（IDLE）：UART 处于空闲或非活动状态 1（忙）：UART 忙（主动传输数据）
8	ADDR_RCVD	RC_W1	0	地址接收。 如果启用 9 位数据模式（UART_CR3.M_E = 1），则该位用于指示接收数据的第 9 位被设置为 1。该位还可以用于指示输入字符是地址还是数据。 0：表示字符为数据。 1：表示字符为地址。 注意：用户需要确保在下一个地址字节到达之前清除中断（该位写 1 清 0）。 如果清除中断有延迟，则软件将无法区分多个地址相关的中断。
7	Reserved	-	-	保留
6	TXE	R	1	发送为空。 每当当前没有发送数据且 TDR 为空的时候，就会设置此位。如果 TXEIE 使能，则会产生 TXE 中断 0：发送不为空 1：发送为空
5	TDRE	R	1	发送保持寄存器空。 该位指示 TDR 为空。 每当数据从 TDR 传输到发送器移位寄存器且没有新数据写入 TDR 时，该位被设置。如果 TDRIE 使能，则会

Puya Confidential

				产生中断。 0：TDR 非空 1：TDR 为空
4	BRI	RC_W1	0	断开中断位 这用于指示在串行输入数据上检测到中断序列。 如果在 UART 模式下,每当串行输入保持在逻辑“0”状态的时间超过起始时间+数据位+奇偶校验位+停止位的总和时，就会设置它。对该位写 1 将清除 BRI 位。 0：未检测到断开序列 1：检测到断开序列
3	FE	RC_W1	0	帧错误位。 这用于指示接收机中出现帧错误。当接收机在接收的数据中没有检测到有效的停止位时，就会发生帧错误。 应该注意的是，如果发生断开，也将设置帧错误（UART_SR.FE）位，如断开中断（UART_SR.BRI）位所示。之所以会发生这种情况，是因为断开字符通过将输入保持到逻辑 0 的时间长于字符的持续时间而隐式地生成帧错误。对该位写 1 将清除 FE 位。 0：无帧错误 1：帧错误
2	PE	RC_W1	0	奇偶校验错误位。 如果设置了奇偶校验使能（UART_CR1.PEN）位，则该寄存器用于指示接收器发生奇偶校验错误。 应该注意的是，如果发生了断开，在这种情况下，如果奇偶校验生成和检测被启用（UART_CR1.PCE = 1）并且奇偶校验被设置为奇数（UART_CR1.PS = 0），则 PE 也会被设置。对该位写 1 将清除 PE 位。 0：无奇偶校验错误 1：奇偶校验错误
1	ORE	RC_W1	0	溢出错误位。 这用于指示溢出错误的发生。 如果在读取以前的数据之前接收到新的数据字符，则会发生这种情况。 当新字符在从 RDR 读取前一个字符之前到达接收器时，设置 ORE 位。当这种情况发生时，RDR 中的数据将被覆盖。 对该位写 1 将清除 ORE 位。 0：无溢出错误 1：溢出错误
0	RXNE	R	0	数据就绪位。 这用于指示接收器在 RDR 中至少包含一个字符。 读取 RDR 时，此位被清除。 0：数据未就绪 1：数据就绪

Puya Confidential

29.5.4 UART 控制寄存器 1（UART_CR1）

地址偏移：0x14

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MSBFIRST	SWAP	Res.	SBK	SP	PS	PCE	STOP	M[1:0]	
						RW	RW		RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:10	Reserved	-	-	保留
9	MSBFIRST	RW	0	最高有效位在前。 0：起始位后，收发第 0 位数据； 1：起始位后，收发第 5/6/7/8/9 位数据； 在数据传输过程中，不能修改这个位。
8	SWAP	RW	0	TX/RX 引脚互换。 0：TX/RX 引脚按照标准引脚映射定义； 1: TX/RX 引脚互换。此时用作交叉连接其他 UART 时。
7	Reserved	-	-	保留
6	SBK	RW	0	断开控制位。 这用于将断开发送到接收设备。如果设置为 1，串行输出将强制进入间隔（逻辑 0）状态。输出线被强制为低电平，直到 SBK 位被清除。 0：释放串行输出以进行数据传输 1：串行输出被迫进入间隔状态
5	SP	RW	0	固定奇偶校验。 仅当 UART 不忙时可写（UART_SR.BUSY 为 0）。此位用于强制固定奇偶校验值。当 PCE、PS 和 SP 设置为 1 时，奇偶校验位被发送并检查为逻辑 0。如果 PCE 和 SP 被设置为 1，并且 PS 是逻辑 0，则奇偶校验位被发送并被检查为逻辑 1。如果此位设置为 0，则 SP（固定奇偶校验）被禁用。 0：固定奇偶校验已禁用 1：固定奇偶校验已启用
4	PS	RW	0	偶数奇偶校验选择。 仅当 UART 不忙时可写（UART_SR.BUSY 为 0）。当奇偶校验被启用（PCE 设置为 1）时，这用于在偶数和奇数奇偶校验之间进行选择。如果设置为 1，则传输或检查偶数个逻辑“1”。如果设置为零，则传输或检查奇数个逻辑“1”。 0：奇校验 1：偶校验
3	PCE	RW	0	奇偶校验启用。 仅当 UART 不忙时可写（UART_SR.BUSY 为 0）。该位用于控制发送和接收的串行字符中的奇偶校验生成和检测。

Puya Confidential

				0：禁用奇偶校验 1：启用奇偶校验
2	STOP	RW	0	停止位的数量。 仅当 UART 不忙时可写（UART_SR.BUSY 为 0）。这用于选择外围设备将发送和接收的每个字符的停止位数。如果设置为零，则在串行数据中传输一个停止位。如果设置为 1 并且数据位设置为 5（UART_CR1.M 设置为 0），则发送一个半停止位。否则，传输两个停止位。 注意，无论所选择的停止位的数量如何，接收器将仅检查第一个停止位。 注：由于某些配置的字符之间插入的空闲时间和传输方向上的波特时钟除数值，UART 实现的停止位持续时间可能会更长； 0：1 个停止位 1：1.5/2 个停止位
1:0	M[1:0]	RW	2'b0	数据长度选择。 仅当 UART 不忙时可写（UART_SR.BUSY 为 0）。当 UART_CR3 中的 M_E 设置为 0 时，此寄存器用于选择外围设备将发送和接收的每个字符的数据位数。 0x0：每个字符 5 个数据位 0x1：每个字符 6 个数据位 0x2：每个字符 7 个数据位 0x3：每个字符 8 个数据位

29.5.5 UART 控制寄存器 2（UART_CR2）

地址偏移：0x18

复位值：0x0000_0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXEIE	BUSYERRIE	LSIE	TDREIE	RXNEIE
											RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:5	Reserved	-	-	保留
4	TXEIE	RW	0	启用发送空中断。这用于启用/禁用传输发送空中断的生成。 0：禁用发送空中断 1：使能发送空中断
3	BUSYERRIE	RW	1	启用 BUSYERR 状态中断。这用于启用/禁用 BUSYERR 状态中断生成。UART 繁忙时（UART_SR.BUSY 设置为 1），主机尝试写入 UART_CR1 寄存器。 0：禁用 BUSYERR 状态中断 1：使能 BUSYERR 状态中断

Puya Confidential

2	LSIE	RW	0	启用接收器线路状态中断。这用于启用/禁用接收器线路状态中断的生成。 该中断标志是 PE、FE、ORE、BRI、ADDR_RCVD 中断标志的组合。 0：禁用接收器线路状态中断 1：使能接收器线路状态中断
1	TDREIE	RW	0	启用传输保持寄存器空中断。这用于启用/禁用传输保持寄存器空中断的生成。 0：禁用传输保持寄存器空中断 1：使能传输保持寄存器空中断
0	RXNEIE	RW	0	启用接收数据可用中断。这用于启用/禁用接收数据可用中断。 0：禁用接收数据中断 1：使能接收数据中断

29.5.6 UART 控制寄存器 3（UART_CR3）

地址偏移：0x1C

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA_SOFT_ACK	DMAT	DMAR	TX_MODE	SEND_ADDR	ADDR_MATCH	M_E
									RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:7	Reserved	-	-	保留
6	DMA_SOFT_ACK	RW	0	软件清除 DMA 发送请求和 DMA 接收请求。 0：不影响 1：清除 DMA 发送请求和 DMA 接收请求
5	DMAT	RW	0	传送时使能 DMA。 0：禁止； 1：使能发送 DMA；
4	DMAR	RW	0	接收时使能 DMA。 0：禁止； 1：使能接收 DMA；
3	TX_MODE	RW	0	传输模式控制位。该比特用于控制 9-bit 数据传输期间的传输模式的类型。 数值： 1：在这种操作模式下，传输保持寄存器（TDR）为 9 位宽。 用户需要确保 TDR 寄存器的地址/数据写入正确。 地址：第 9 位设置为 1 数据：第 9 位数设置为 0 注意：传输地址寄存器（UART_TAR）不适用于此操作

Puya Confidential

				模式。 0：在此操作模式中，传输保持寄存器（TDR）的宽度为 8 位。 用户需要将地址编程到传输地址寄存器（UART_TAR）中，并将数据编程到 TDR 寄存器中。
2	SEND_ADDR	RW	0	发送地址控制位。此位用作控制旋钮，供用户在传输模式下确定何时发送地址。 0：9 位字符内容来自 TDR 寄存器 1：9 位字符内容：第 9 位设置为 1，其余 8 位将与“传输地址寄存器”中正在编程的内容相匹配。 注： 1.在发出地址字符后，该位由硬件自动清除。用户不应将此位编程为 0。 2.此字段仅在 M_E 位设置为 1 且 TX_MODE 设置为 0 时适用。
1	ADDR_MATCH	RW	0	地址匹配模式。此位用于在接收期间启用地址匹配功能。 ■ 在地址匹配模式下，UART 将等待第 9 位设置为 1 的传入字符。并进一步检查地址是否与“接收地址匹配寄存器”中编程的地址匹配。如果匹配，则后续字符将被视为有效数据，UART 开始接收数据。 ■ 在正常模式下，UART 将开始接收数据，9 位字符将形成。用户负责读取数据并区分 b/n 地址和数据。 0：正常模式 1：地址匹配模式 注：仅当 M_E 设置为 1 时，此字段才适用。
0	M_E	RW	0	9 数据使能 此位用于启用用于发送和接收传输的 9 位数据 0：禁止 9 位数据 1：使能 9 位数据

29.5.7 UART 接收地址寄存器（UART_RAR）

地址偏移：0x20

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RAR[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7:0	RAR[7:0]	RW	8'b0	接收模式下的地址匹配寄存器。 如果在输入字符中第 9 位被设置为 1，则将对照该寄存器值检查剩余的 8 位。如果匹配成功，则第 9 位设置为 0 的后续字符将被视为数据字节，直到接收到下

Puya Confidential

				一个地址字节。 注： 1. 仅当“ADDR_MATCH”（UAR_CR3[1]）和“M_E”（UART_CR3[0]）位设置为 1 时，此寄存器才适用。 2. RAR 可以在任何时间点被编程。但是，当任何接收正在进行时，用户不得更改此寄存器值。

29.5.8 UART 发送地址寄存器（UART_TAR）

地址偏移：0x24

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAR[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:8	Reserved	-	-	保留
7:0	TAR[7:0]	RW	8'b0	这是传输模式下的地址匹配寄存器。 如果 M_E 位被启用，并且“SEND_ADDR”（UART_CR3[2]）位被设置为 1，则 UART 将发送第 9 位设置为 1 的 9 位字符，剩余的 8 位地址将从该寄存器发送。 注： 1. 此寄存器仅用于发送地址。正常数据应通过编程 TDR 寄存器发送。 2. 在 UART 串行通道上开始发送地址后，硬件将自动清除“SEND_ADDR”位

29.5.9 UART 波特率小数寄存器（UART_BRRF）

地址偏移：0x28

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRRF[3:0]			
												RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:4	Reserved	-	-	保留
3:0	BRRF[3:0]	RW	4'b0	波特率小数部分。 小数值由（BRRF）/（2^4）确定。

Puya Confidential

30. 低功耗通用异步收发器（LPUART）

LPUART 是一种 UART，允许在有限功耗下双向 UART 通信。仅需 32.768 kHz LSE 时钟即可进行高达 9600 波特/s 的 UART 通信。当 LPUART 由与 LSE 时钟不同的时钟源驱动时，可以达到更高的波特率。

即使当微控制器处于低功耗模式，能耗极低时，LPUART 也会等待 UART 帧的到来。LPUART 包含所有必要的硬件支持，使在最小功耗下可以进行异步串行通信。

它支持半双工单线通信和调制解调器操作(CTS/RTS)，还支持多处理器通信。

DMA（直接存储器访问）可用于数据发送/接收。

30.1 LPUART 主要特性

- AMBA APB 接口
- 全双工异步通讯
- NRZ 标准模式（标记/空格）
- 波特率可编程
- 32.768 kHz 时钟，波特率范围 300 ~ 9600 波特/s.更高波特率需要更高时钟频率支持
- 双时钟域：PCLK 及专用内核时钟
- 数据字长度可配置（7 位/8 位/9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位数可配置（1/2 位）
- 单线半双工通讯
- 支持 DMA 连续传输
- 传送和接收独立使能
- 独立发送/接收信号极性控制
- Tx/Rx 引脚可以互换
- 支持硬件 RS-485/modem 流控制
- 发送检测标志
  - 接收缓存器满
  - 发送缓存器空
  - 忙标志
  - 传输结束
- 奇偶校验控制
  - 发送时产生奇偶校验位
  - 接收时奇偶校验
- 4 个错误标志
  - 上溢错误
  - 噪声检测
  - 帧结构错误
  - 奇偶校验错误

Puya Confidential

- 中断源标志
  - CTS 改变
  - 发送寄存器空
  - 发送完成
  - 接收数据寄存器非空
  - 检测到总线空闲
  - 溢出错误
  - 帧错误
  - 噪音操作
  - 校验错误
  - 地址字节匹配
- 多处理器通讯：当地址不匹配时 LPUART 进入静默模式
- 支持静默模式唤醒（空闲帧/地址标记检测）

30.2 LPUART 功能描述

30.2.1 LPUART 框图

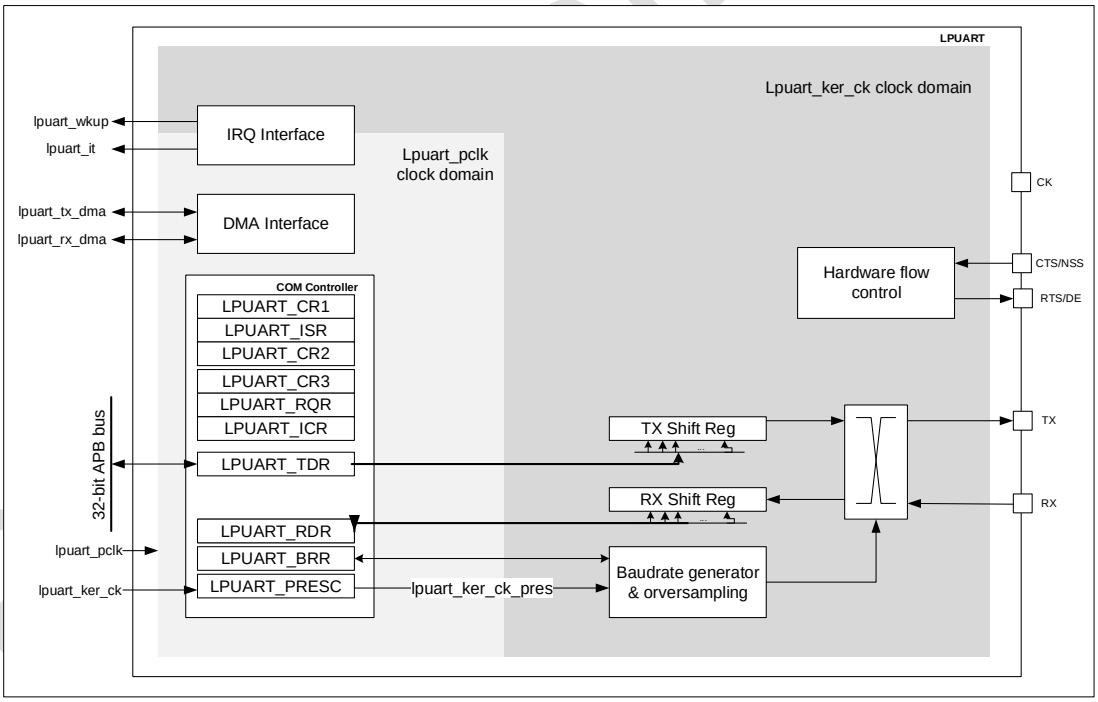


图 30-1 LPUART 框图

30.2.2 LPUART 信号

任何 LPUART 双向通信至少需要两个引脚：接收数据输入（RX）和发送数据输出（TX）。

- RX：接收数据串行输入。
- TX：发送数据输出。当发送器被禁止时，输出引脚恢复到它的 I/O 端口配置。当发送器使能，但不

发送数据时，TX 引脚处于高电平。在单线模式里，此 I/O 口被同时用于数据的发送和接收。

硬件流控制模式

- CTS：低电平开始发送，发送结束变为高电平
- RTS：低电平表明 LPUART 准备好接收数据

RS485 硬件流控制模式

- DE：驱动器使能，激活外部收发器的发送

注：DE 和 RTS 复用同一引脚

30.2.3 LPUART 字符描述

根据{M1，M0}配置，分为 7 位/8 位/9 位三种数据长度。

M[1:0]= 2’b10，7 位；

M[1:0]= 2’b00：8 位；

M[1:0]= 2’b01：9 位；

在默认情况下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

空闲字符定义：在 7 位/8 位/9 位数据长度+停止位时间内为全 1；

断开字符定义：在 7 位/8 位/9 位数据长度时间内为全 0；断开帧结尾，传送器插入 2 个停止位。即 8 位模式，断开帧长度为 10 位 0；9 位模式，断开帧长度为 11 位 0.

发送和接收操作由通用波特率发生器驱动。发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面是各个不同长度帧的波形。

Puya Confidential

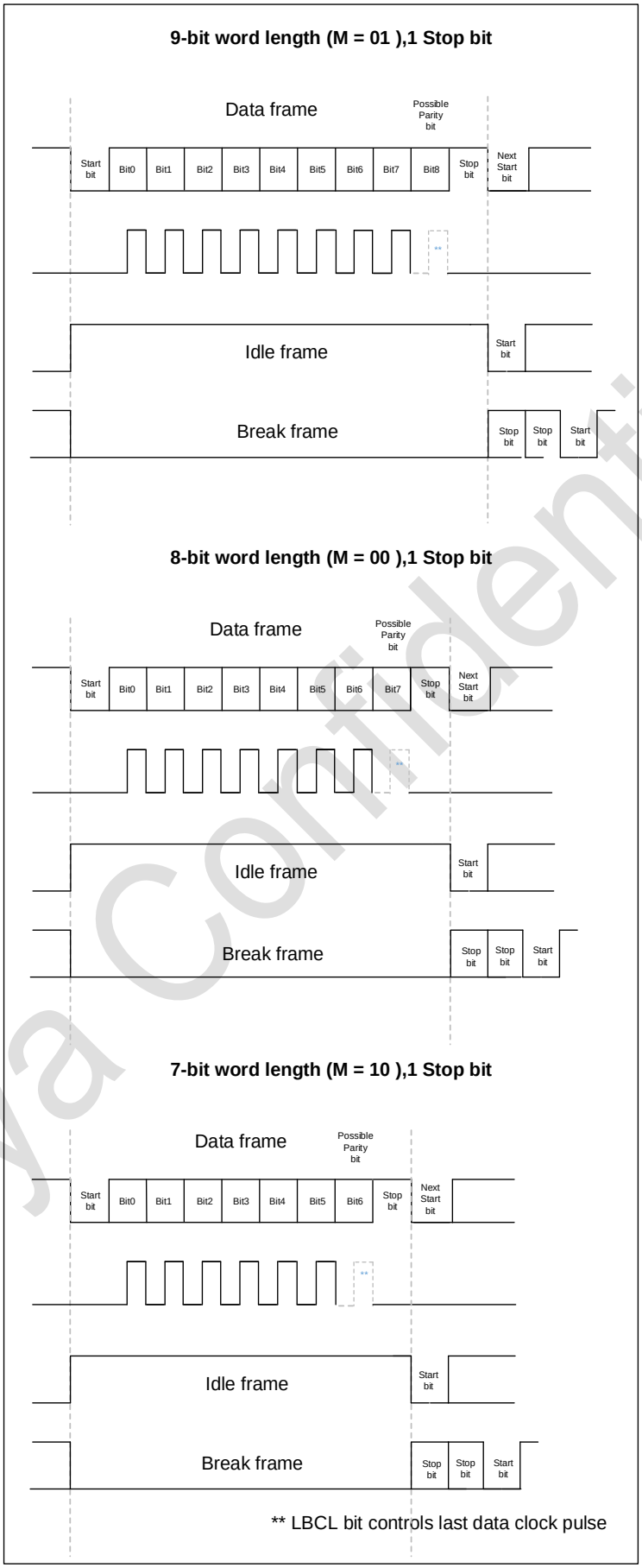


图 30-2 LPUART 字长编程

Puya Confidential

30.2.4 LPUART 发送

LPUART 发送步骤如下所述。

- 配置 LPUART_CR1.{M1,M0}，定义传输数据宽度（7/8/9 位）；
- 配置 LPUART_BRR 寄存器，选择波特率；
- 配置 LPUART_CR2.STOP 寄存器，定义停止位个数(1/2 位);
- 配置 LPUART_CR1.UE=1,使能 LPUART;
- 多处理器传输时，配置 LPUART_CR3.DMAT=1，使能 DMA;
- 配置 TE=1，硬件插入空闲帧时序;
- 写传输数据到 LPUART_TDR 寄存器（该操作清零 TXE 位）。该项操作可能重复多次。
  - 写数据到 LPUART_TDR 会清零 TXE 标志；
- 写最后一个数据到 LPUART_TDR 后，软件等待 TC=1，表明最后一帧发送完成。
  - 表明为最后一帧；
- 硬件产生发送时序：
  - 产生起始位，开始传送
  - 传送移位寄存器输出数据（默认 LSB）到 TX 引脚
  - 停止位（根据配置发送个数）
- 传送完最后 1 帧数据后，产生 TC=1（TXE=1 时）。当 TCIE=1，产生 TC 中断。

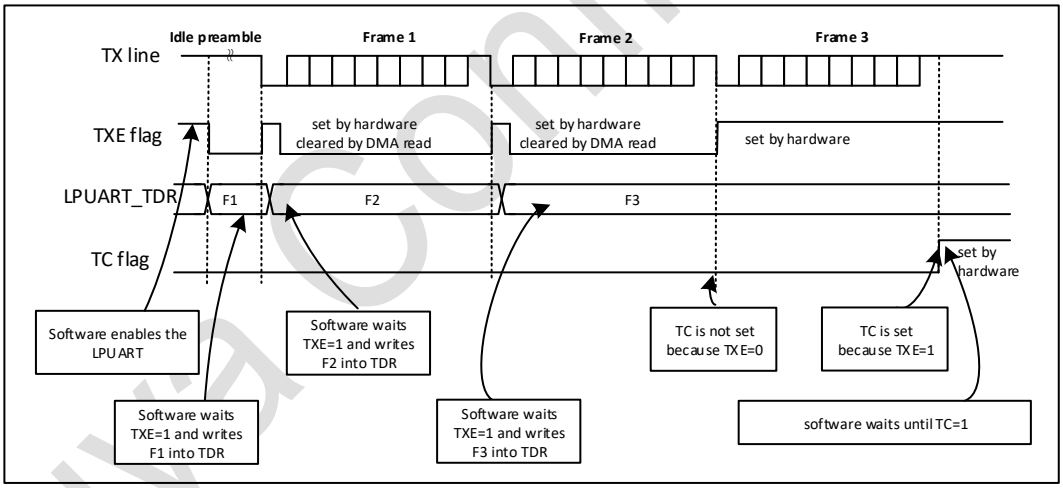


图 30-3 发送时的 TC/TXE 行为 TC/TXE 时序

特殊帧处理

- 断开帧

设置 SBKRQ=1 将会在当前传输完成后传送一个断开帧。断开帧的长度取决于{M1，M0}寄存器的配置。断开帧发送完毕后，硬件会清零 SBKRQ。

- 空闲帧

设置 TE=1，在第一个数据帧之前先发送一个空闲帧。

Puya Confidential

30.2.5 LPUART 接收

起始位检测

在 RX 上检测到下降沿后，硬件开始检测起始位。在该位的中间检测到 0，则判定该位为起始位。如果检测到的是 1，则置位噪声错误标志（NE），当前起始位无效，重新检测起始位。

数据字符接收

字符接收配置步骤：

- 配置 LPUART_CR1.{M1，M0}，选择接收长度；
- 配置 LPUART_BRR 寄存器，选择波特率；
- 配置 USAR_CR2.STOP，选择 1 位或者 2 位停止位；
- 配置 LPUART_CR1.UE=1，使能 LPUART；
- 多处理器传输时，配置 LPUART_CR3.DMAR=1，使能 DMA 接收;
- 配置 LPUART_CR1.RE，使能接收，开始检测起始位；

当接收到字符后：

- 置位 RXNE。表明移位寄存器的内容已被转移到 RDR，即数据可以被读出；
- 当 RXNEIE=1 时，产生中断；
- 在接收时，如果检测到帧错误、噪声错误或溢出错误，则设置相应错误标志位；
- 多处理器通讯：
  - RXNE 在每个字节接收后被置位，并由 DMA 对接收数据寄存器的读操作而清零；
- 在单缓冲器模式：
  - 软件读 LPUART_RDR 寄存器对 RXNE 位清零。RXNE 标志也可以通过写 LPUART_RQR.RXFRQ=1 清零。RXNE 位必须在下一字符接收结束前被清零，以免溢出错误；

特殊帧及错误处理

- 断开字符

当接收到一个断开帧时，LPUART 按错误帧处理。

- 空闲字符

当检测到空闲帧时，按正常帧处理，如果 IDLEIE=1，则产生中断。

- 上溢错误

如果 RXNE 没有被复位，又接收到一个字符，则发生溢出错误。数据只有当 RXNE 位被清零后才能从移位寄存器转移到 RDR 寄存器。每接收到 1 个字节后置位 RXNE 标志。如果下一个数据已收到或先前 DMA 请求还没执行时，RXNE 标志仍为置起状态，溢出错误产生。

溢出错误产生时，执行如下错误处理：

- 置位 ORE；
- RDR 内容不会丢失。读 LPUART_DR 寄存器仍可正常读出先前数据；
- 移位寄存器内容被覆盖，随后接收到的数据丢失；
- 如果 RXNEIE=1，或者 EIE=1，则产生中断；
- 写 ORECF=1 清零 ORE 位；

Puya Confidential

ORE 置位，表明至少 1 个数据已经丢失。有如下两种情况：

- 如果 RXNE=1，上一个有效数据还在接收寄存器 RDR 中，可以被读出；
- 如果 RXNE=0，上一个有效数据已经被读走，RDR 无有效数据可读。当上一个有效数据在 RDR 中被读取的同时又接收到新的（被丢失）的数据时，此种情况可能发生。

30.2.6 LPUART 时钟源选择

LPUART 的时钟源由 RCC_CCIPR.LPUARTxSEL 寄存器配置。在配置 LPUART_CR1.UE=1 前，必须配置时钟源。

时钟源的选择决定了通讯速度范围。

LPUART 双时钟域，一个为内核时钟（lpuart_ker_ck），一个为系统时钟（lpuart_pclk）。当用作从低功耗唤醒时，内核时钟源由 RCC_CCIPR.LPUARTxSEL 寄存器配置。

当不支持双时钟域时，内核时钟与系统时钟相同。

当 MCU 进入低功耗模式，内核时钟源选择低频时，LPUART 可以接收数据并唤醒 MCU，然后软件或者 DMA 可以读取 LPUART_RDR 中数据。

帧错误

接收时没有检测到停止位，从而出现同步失效或大量噪声时，会产生帧错误。

帧错误时做如下处理：

- 硬件设置 FE 位；
- 无效数据从移位寄存器传送到 LPUART_RDR 寄存器；
- 在单字节通讯情况下，不产生中断。但因为 NE 标志位和 RXNE 标志位时同时被设置，FE 位出现上升沿。在多处理器通讯情况下，如果 LPUART_CR3.EIE=1，将产生一个中断。
- 写 LPUART_ICR.FECF=1 可以清零 FE 标志。

配置停止位个数

可配置位 1/2 个停止位：

- 1 停止位：在第 8/9/10 采样点采样停止位；
- 2 停止位：在第一个停止位采样后采样第 2 个停止位。在检测到第 2 个停止位结束后设置 RXNE 和 FE 标志。第 1 个停止位不会检测帧错误。

30.2.7 LPUART 波特率产生

发送和接收配置相同波特率。

$$\text{Tx/Rx 波特率} = \frac{256 \ X \ \text{lpuartckpres}}{\text{LPUARTDIV}}$$

LPUARTDIV 在 LPUART_BRR 寄存器配置。在通讯进行时不要改变波特率的值。LPUART_BRR 寄存器值不能低于 0x300.

fCK 的频率范围为 3 ~ 4096 倍*波特率。

LPUART 时钟源为 32.768 kHz 时，最大波特率为 9600 波特率。当 LPUART 由不同的时钟时，可以达到更高的波特率。例如，LPUART 时钟源频率为 100 MHz，可达到的最大波特率约为 33M 波特。

Puya Confidential

表 30-1 lpuart_ker_ck_pres = 32.768 kHz 时编程波特率的误差计算

Baud rate	lpuart_ker_ck_pres = 32.768 kHz		
期望值	**实际值**	**在波特率寄存器中编程的值**	**误差%**
0.3 KBps	0.3 KBps	0x6D3A	0
0.6 KBps	0.6 KBps	0x369D	0
1200 Bps	1200.087 Bps	0x1B4E	0.007
2400 Bps	2400.17 Bps	0xDA7	0.007
4800 Bps	4801.72 Bps	0x6D3	0.035
9600 Bps	9608.94 Bps	0x369	0.035

表 30-2 lpuart_ker_ck_pres = 100 MHz 时编程波特率的误差计算

Baud rate	lpuart_ker_ck_pres = 100 MHz		
期望值	**实际值**	**在波特率寄存器中编程的值**	**误差%**
38400 波特	38400.04 波特	A2C2A	0.0001
57600 波特	38400.04 波特	6C81C	0.0001
115200 波特	115200.12 波特	3640E	0.0001
230400 波特	230400.23 波特	1B207	0.0001
460800 波特	460804.61 波特	D903	0.001
921600 波特	921625.81 波特	6C81	0.0028
4000000 波特	4000000.00 波特	1900	0
10000000 波特	10000000.00 波特	A00	0
20000000 波特	20000000.00 波特	500	0
33000000 波特	33032258.06 波特	307	0.1

30.2.8 LPUART 接收容忍度

只有当整体的时钟系统地变化小于 LPUART 异步接收器能够容忍的范围，LPUART 异步接收器才能正常地工作。影响这些变化的因素有：

- DTRA：由于发送器误差而产生的变化(包括发送器端振荡器的变化)
- DQUANT：接收器端波特率取整所产生的误差
- DREC：接收器端振荡器的变化
- DTCL：由于传输线路产生的变化(通常是由于收发器在由低变高的转换时序，与由高变低转换时序之间的不一致性所造成)。

需要满足： DTRA + DQUANT + DREC + DTCL+DWU < LPUART 接收器的容忍度

其中 DWU 是使用低功耗模式唤醒时由于采样点偏差引起的误差。

LPUART 接收器可以在指定的最大容忍偏差下正确接收数据（理论值）

表 30-3 LPUART 接收容忍度

{M1,M0}位	768 < BRR < 1024	1024 < BRR < 2048	2048 < BRR < 4096	4096 ≤ BRR
8 位({M1,M0} =00), 1 停止位	1.82%	2.56%	3.90%	4.42%
9 位({M1,M0} =01), 1 停止位	1.69%	2.33%	2.53%	4.14%
7 位({M1,M0} =10), 1 停止位	2.08%	2.86%	4.35%	4.42%
8 位({M1,M0} =00), 2 停止位	2.08%	2.86%	4.35%	4.42%
9 位({M1,M0} =01), 2 停止位	1.82%	2.56%	3.90%	4.42%
7 位({M1,M0} =10), 2 停止位	2.34%	3.23%	4.92%	4.42%

Puya Confidential

注：当接收到的帧中包含恰好是 10/11/9 位的空闲帧（对应 M='b00/11/10'）时，表中指定的数据在特殊情况下可能会略有不同。

30.2.9 LPUART 多处理器通信

通过 LPUART 可以实现多处理器通讯（将几个 LPUART 连在一个网络里）。例如某个 LPUART 设备可以是主，其 TX 输出和其他 LPUART 从机的 RX 输入相连接；LPUART 从机各自的 TX 输出逻辑与在一起，并且和主机的 RX 输入相连接。

未被寻址的设备可以配置 MME 寄存器置于静默模式。在静默模式：

- 不会设置任何接收状态位；
- 禁止所有接收中断；
- LPUART_ISR.RWU=1.通过配置 LPUART_RQR.MMRQ，硬件和软件可以控制 LPUART 进入静默模式；

根据 LPUART_CR1.WAKE 的值，LPUART 可以用两种方式进入或退出静默模式：

- WAKE=0：空闲总线检测；
- WAKE=1：地址标志检测；

空闲总线检测

当 MMRQ 写 1，RMU 自动置 1 后，LPUART 进入静默模式。

当检测到空闲帧时，LPUART 被唤醒。然后 RWU 被硬件清零，但是 LPUART_ISR.IDLE 并不置起。

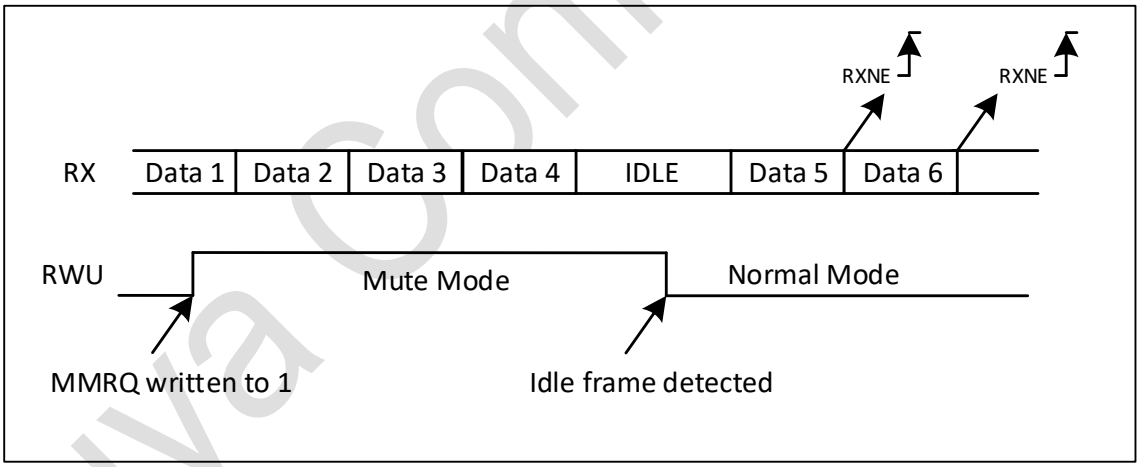


图 30-4 用总线帧检测的静默模式

注：

如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。

如果在线路处于空闲状态时激活 LPUART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标志检测

在这个模式，如果 MSB=1，该字节被认为是地址，否则被认为是数据。在一个地址字节中，目标接收器的地址被放在 4 个或者 7 个 LSB 中（由 ADDM7 寄存器选择几个）。这个 4 位/7 位地址被接收器同 LPUART_CR2 寄存器中 ADD 中配置的地址比较。

Puya Confidential

如果接收到的字节与配置地址不匹配，LPUART 进入静默模式。此时，硬件设置 RWU。接收该字节既不会设置 RXNE 标志也不会产生中断或发出 DMA 请求，因为 LPUART 已经在静默模式。

当 MMRQ 配置为 1 时，LPUART 也会进入静默模式，并设置 RWU=1.

如果接收到的字节与配置地址匹配，LPUART 退出静默模式。此时，清零 RWU，随后正常接收数据。因为 RWU 已被清零，此种情况会设置 RXNE/RXFNE 位。

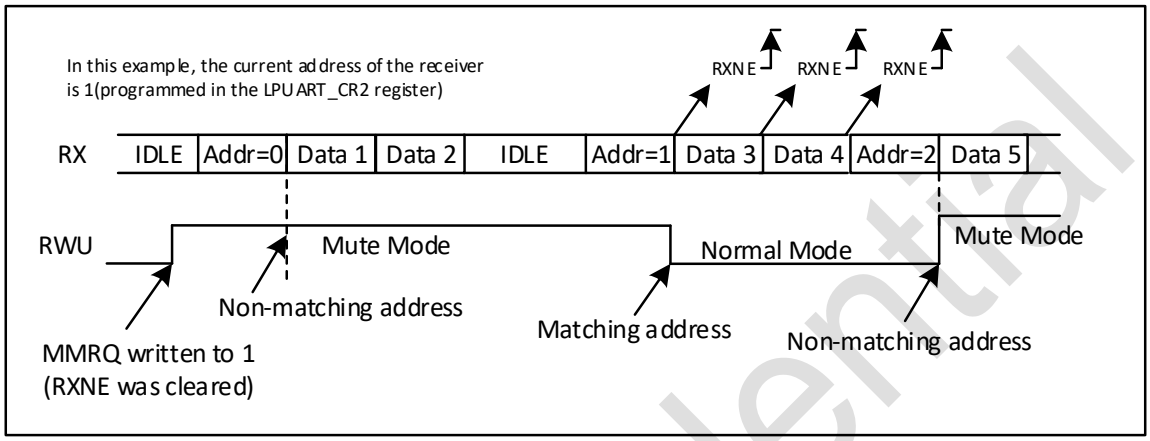


图 30-5 用地址标志检测的静默模式

30.2.10 LPUART 奇偶校验

设置 LPUART_CR1.PCE=1，可以使能奇偶控制（发送时生成一个奇偶位，接收时进行奇偶校验）。根据{M1,M0}位定义的帧长度，支持的 LPUART 帧格式如下表所示：

表 30-4 支持的 LPUART 帧格式

{M1,M0}位	PCE 位	LPUART 帧
00	0	| 开始位 | 8 位数据 | 停止位 |
00	1	|开始位| 7 位数据| 奇偶校验位 |停止位|
01	0	|开始位| 9 位数据|停止位|
01	1	|开始位| 8 位数据|奇偶校验位|停止位|
10	0	|开始位| 7 位数据|停止位|
10	1	|开始位| 6 位数据|奇偶校验位|停止位|

偶校验

偶校验位使得一帧中的 6 或 7 或 8 个 LSB 数据以及校验位中“1”的个数为偶数。

奇校验

奇校验位使得一帧中的 6 或 7 或 8 个 LSB 数据以及校验位中“1”的个数为奇数。

接收校验

奇偶校验错误时设置 PE=1，当 PEIE=1 时产生中断。软件写 PECF=1 清零 PE。

发送校验位

发送时 MSB 由奇偶校验位代替。

30.2.11 LPUART 单线半双工通信

当配置 LPUART_CR3.HDSEL=1 时，LPUART 进入单线半双工通讯模式。

Puya Confidential

在此模式下：

- TX 和 RX 在模块内部连接；
- RX 引脚不用；
- 当无数据传输时释放 TX 引脚作为标准 I/O

30.2.12 DMA 连续通信

LPUART 可以利用 DMA 连续通讯。接收缓冲器和发送缓冲器的 DMA 请求分别产生。

利用 DMA 发送

通过配置 LPUART_CR3.DMAT 使能利用 DMA 发送。当 TXE=1，DMA 就从指定的 SRAM 区传送数据到 LPUART_DR 寄存器。

DMA 发送的配置步骤如下：

- 配置 SYSCFG_CFGR3 或者 SYSCFG_CFGR4 选择 LPUART 使用的 DMA 通道
- 配置 LPUART_TDR 寄存器地址作为 DMA 传输的目标地址。在每个 TXE 事件后，数据将被传送到这个地址；
- 配置存储器（如 SRAM）地址作为 DMA 传输的源地址。在每次 TXE 事件后，将从此存储器读数据加载到 LPUART_TDR 寄存器；
- 写 DMA 控制寄存器配置传输字节数；
- 写 DMA 寄存器配置通道优先级；
- 根据应用，配置在传输完成一半还是全部完成时产生 DMA 中断；
- 写 TCCF=1 清零 LPUART_ISR.TC 标志；
- 使能 DMA 通道；

当传输达到配置的传输字节数，DMA 控制器产生中断请求。

当 DMA 写完所有要传输数据后，设置 TCIF=1.然后软件需要等待 TC=1，表明 LPUART 已经发送完最后一帧数据。

Puya Confidential

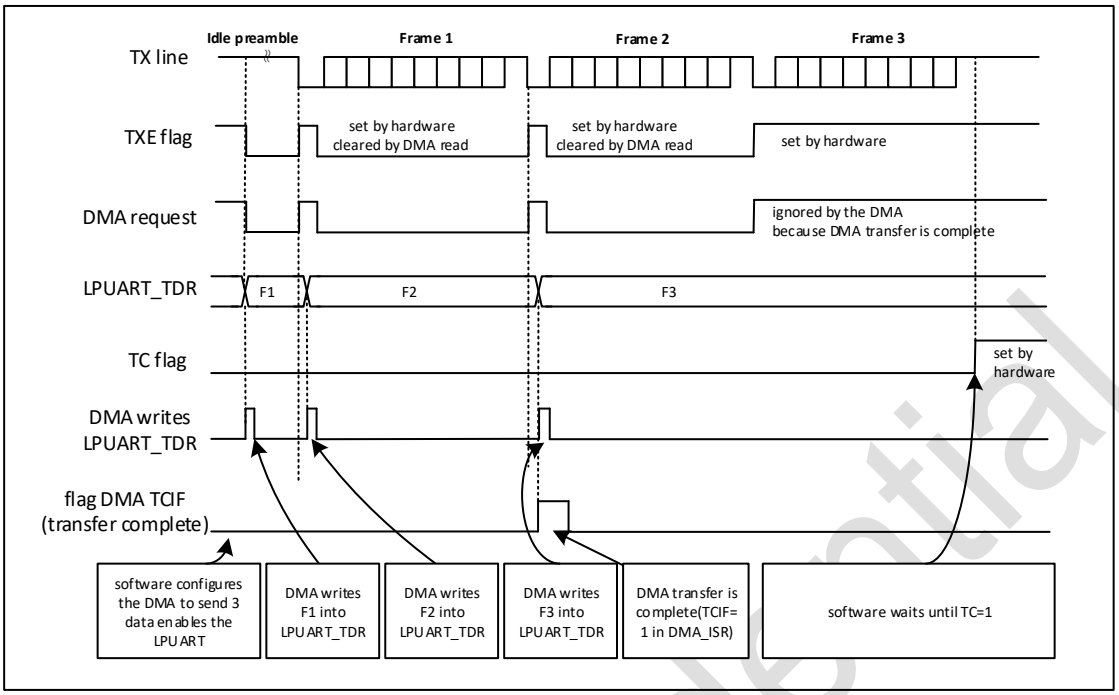


图 30-6 LPUART 用 DMA 发送

利用 DMA 接收

通过配置 LPUART_CR3.DMAR 使能利用 DMA 接收。每收到一个字节，DMA 就把数据从 LPUART_RDR 寄存器传送到指定的 SRAM 区。

DMA 接收的配置步骤如下：

- 配置 SYSCFG_CFGR3 或者 SYSCFG_CFGR4 选择 LPUART 使用的 DMA 通道
- 配置 LPUART_RDR 寄存器地址作为 DMA 传输的源地址。在每个 RXNE 事件后，数据将从此 LPUART_RDR 寄存器地址读出数据传输到存储器；
- 配置 SRAM 存储器地址作为 DMA 传输的目标地址。在每次 RXNE 事件后，数据将从 LPUART_RDR 寄存器传送到此地址；
- 写 DMA 控制寄存器配置传输字节数；
- 写 DMA 寄存器配置通道优先级；
- 根据应用，配置在传输完成一半还是全部完成时产生 DMA 中断；
- 使能 DMA 通道；

当传输达到配置的传输字节数，DMA 控制器产生中断请求。

Puya Confidential

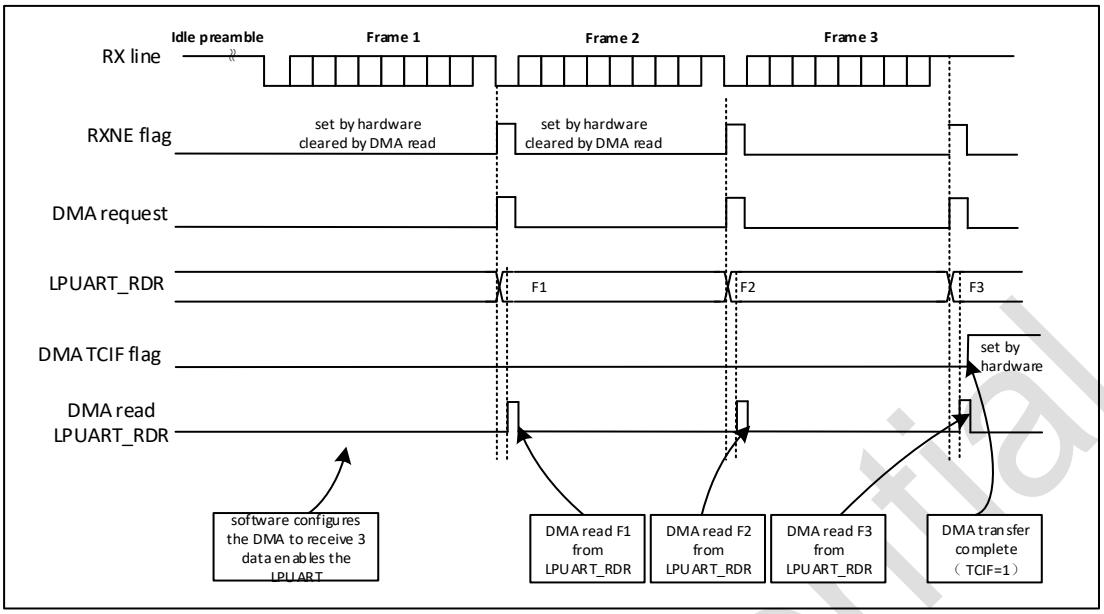


图 30-7 LPUART 用 DMA 接收

错误标志和中断

在多处理器通讯模式，在传输时产生任何错误都会在当前字节传送完成后产生错误标志，如果对应中断使能，则产生中断。

30.2.13 RS232 硬件流控制和 RS485 驱动器使能

通过 CTS 和 RTS 可以控制两个 LPUART 模块的串行数据流。

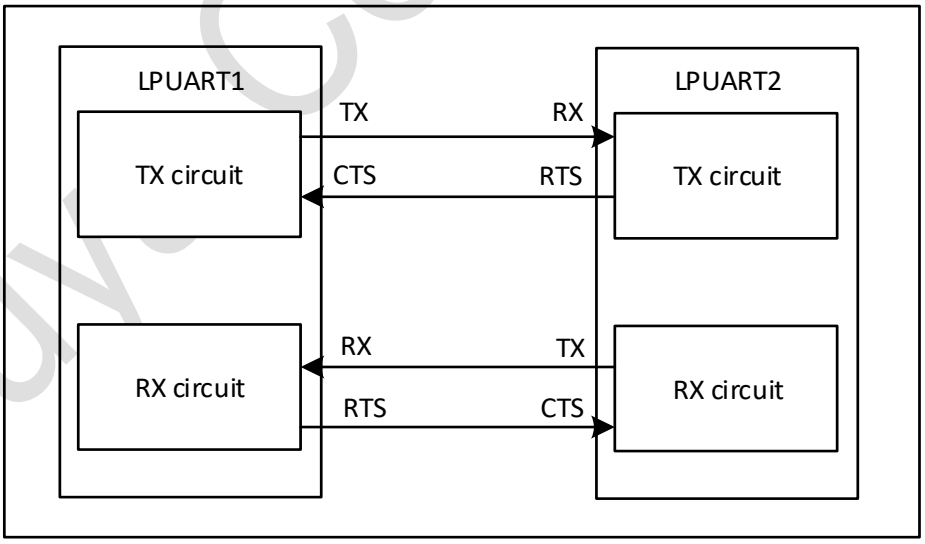


图 30-8 硬件流控制

RS232 RTS 流控制

通过配置 LPUART_CR3.RTSE=1，使能 RTS 流控制。当 LPUART 准备好接收数据时，将 RTS 拉低。当接收寄存器满时，RTS 拉高（无效），表明当前帧是最后一帧。

Puya Confidential

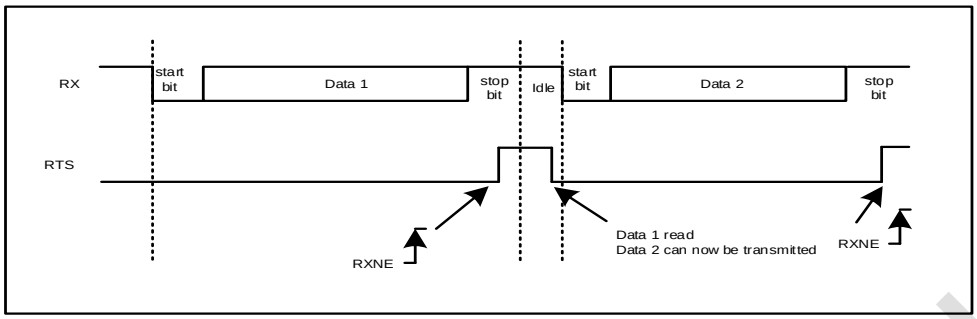


图 30-9 RTS 流控制

RS232 CTS 流控制

通过配置 LPUART_CR3.CTSE=1，使能 CTS 流控制。使能后，在传送下一帧前检测 CTS 信号。如果 CTS=0（有效），传送下一笔数据（假设有数据需要传输）。如果 CTS=1（无效），则当前传输完成（在 stop bit 位前完成）。

当 CTSE=1，只要 CTS 信号反转，硬件就置位 CTSIF。表明接收器准备好通讯或未准备好通讯。当 CTSIE=1 时，产生中断。

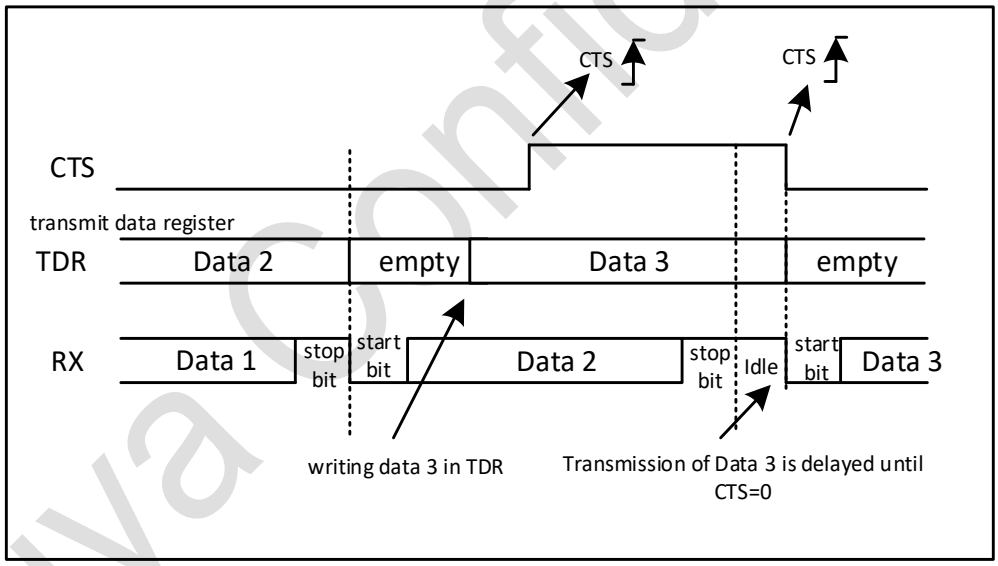


图 30-10 CTS 流控制

注：CTS 在当前帧结束前至少维持 3 个 LPUART 源时钟的高电平。CTSCF 也需要维持至少 2xPCLK 周期。

RS485 驱动使能

配置 LPUART_CR3.DEM=1，使能 DE 功能。通过 DE 信号，用户可以激活外部传输控制。DE 信号的有效极性通过 LPUART_CR3.DEP 寄存器配置。

有效时间由 LPUART_CR1.DEAT 配置，为 DE 信号有效沿与起始位开始之间的时间。

释放时间由 LPUART_CR1.DEDT 配置，为最后一个停止位与 DE 无效之间的时间。

具体时间计算如下所示，其中 P=BRR[20:11]:

Puya Confidential

- DE 插入时间：
  - （1+（DEAT x P））x f_{CK}，P≠0
  - （1+DEAT）x f_{CK}，P=0
- DE 释放时间：
  - （1+（DEDT x P））x f_{CK}，P≠0
  - （1+DEDT）x f_{CK}，P=0

30.2.14 LPUART 低功耗模式

LPUART 支持低功耗模式，使得在 LPUART PCLK 不存在时仍能传输数据。

当 UESM=1，LPUART 能在低功耗模式下唤醒 MCU。

根据缓存产生唤醒中断请求

当 lpuart_pclk 时钟被门控时，如果有特定功能需要 lpuart_pclk 参与，则 LPUART 需要提供一个唤醒中断（lpuart_wkup）：

需要 lpuart_pclk，（DMA/软件）搬空数据寄存器。此时，lpuart_wkup 中断源为 RXNE=1，在进入低功耗模式前，RXNEIE 位必须置 1.

根据 WUS 产生中断唤醒请求

当检测到唤醒事件，硬件产生 WUF 标志。如果 WUFIE=1 时，产生 WUF 中断。在此情况下，WUF 标志置 1 便足以将 MCU 从低功耗模式唤醒。

注：

- 进入低功耗模式前，保证没有 LPUART 传输；
- 不管 MCU 是在低功耗模式还是工作模式，只要检测到唤醒事件，就会置位 WUF 标志；
- 当初始化和使能 LPUART 接收后，必须要通过 REACK 位确认 LPUART 是否真正使能，然后再配置进入低功耗模式；
- 当使能 DMA 接收，在进入低功耗模式前必须关闭 DMA 接收；然后从低功耗唤醒后再重新使能 DMA；

从静默模式进入低功耗模式

如果 LPUART 在进入低功耗模式前处于静默模式：

- 不能用空闲帧唤醒静默模式，因为在低功耗模式下不支持空闲帧检测；
- 用地址匹配的方式唤醒静默模式，则从低功耗模式唤醒也要用地址匹配的方法。当进入低功耗时，如果 RXNE=1，则 LPUART 将不会进入低功耗模式，而维持在静默模式；

低功耗模式时内核时钟关闭

如果低功耗模式下，内核时钟也关闭，则 LPUART 在 RX 线上检测到下降沿时，会产生 lpuart_ker_ck_req 信号去使能内核时钟。

如果唤醒事件 LPUART_CR3.WUS 配置为地址匹配方式：

- 内核时钟使能后，接收完一帧后如果地址匹配，则 MCU 唤醒，开始正常数据接收；

下图为唤醒事件后地址验证匹配的情况：

Puya Confidential

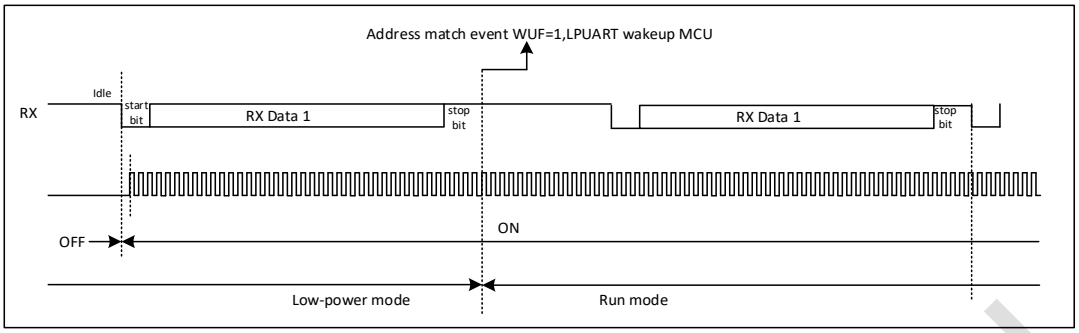


图 30-11 LPUART 地址匹配唤醒

如果唤醒事件后地址验证不匹配，则内核时钟再次关闭，MCU 不会被唤醒，系统仍在低功耗模式。

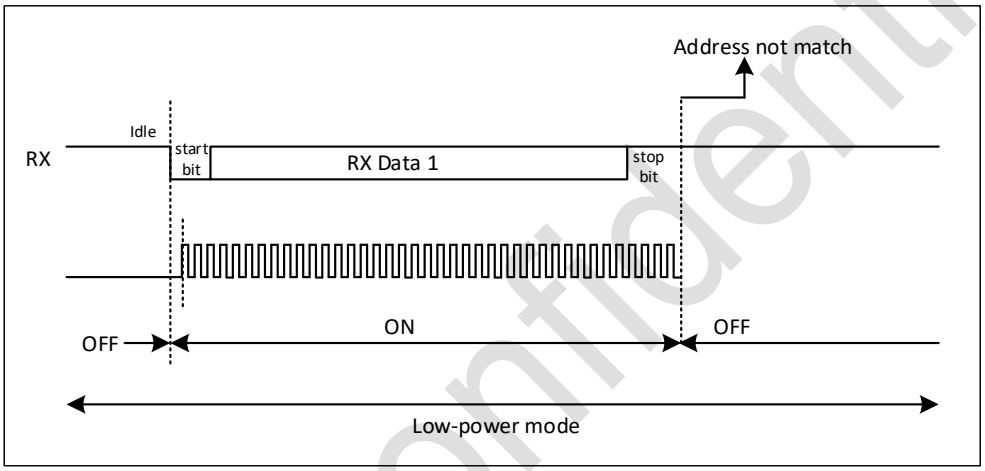


图 30-12 LPUART 地址不匹配无法唤醒

30.3 LPUART 中断

表 30-5 LPUART 中断请求

中断事件	事件标志	使能位	中断清零	中断信号是否有效	
				中断	唤醒
发送数据寄存器空	TXE	TXEIE	写数据到 TDR	YES	NO
CTS 中断	CTSIF	CTSIE	软件配置 CTSCF=1	YES	NO
传送完成	TC	TCIE	写数据到 TDR 或 TCCF=1	YES	NO
接收寄存器未空（读数据准备好）	RXNE（对应标志位 RXFNE）	RXNEIE（对应中断使能 RXFNEIE）	读 RDR 寄存器或 RXFRQ=1	YES	YES
溢出错误	ORE	RXNEIE/RXFNEIE	ORECF=1	YES	NO
空闲帧	IDLE	IDLEIE	IDLECF=1	YES	NO
奇偶校验错误	PE	PEIE	PECF=1	YES	NO
多处理器通讯时，噪声、overrun 和帧错误	NR/ORE/FE	EIE	NECF=1 清零 NE； ORECF=1 清零 ORE； FECF=1 清零 FE；	YES	NO

Puya Confidential

地址字节匹配	CMF	CMIE	CMCF=1	YES	NO
从低功耗模式唤醒	WUF	WUFIE	WUCF=1	YES	YES

30.4 LPUART 寄存器

30.4.1 LPUART 控制寄存器 1（LPUART_CR1）

偏移地址：0x00

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]				
			RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:29	Reserved	-	-	保留
28	M1	RW	0	{M1,M0}的值配置长度，由软件置位或者清零。 2’b00：1 起始位，8 数据位，n 停止位 2’b01：1 起始位，9 数据位，n 停止位 2’b10：1 起始位，7 数据位，n 停止位 只能在 LPUART 未使能时（UE=0）才能写入该位。
27:26	Reserved	-	-	保留
25:21	DEAT[4:0]	RW	5’h0	该寄存器配置驱动使能（DE）信号有效和起始位之间的时间。用采样时间作为单位。 如果不支持 DE 功能，则该位保留。 只能在 LPUART 未使能时（UE=0）才能写入该位
20:16	DEDT[4:0]	RW	5’h0	该寄存器配置发送帧停止位与 DE 信号无效之间的时间。用采样时间作为单位（1/8 或者 1/16 位周期，取决于过采样率）。如果 DEDT 时间内有写 LPUART_TDR 寄存器请求发送，则数据只有当 DEDT 和 DEAT 都结束后才发送。 如果不支持 DE 功能，则该位保留。 只能在 LPUART 未使能时（UE=0）才能写入该位
15	Reserved	-	-	保留
14	CMIE	RW	0	字符匹配中断使能。该位由软件置位或者清零。 0：中断禁止 1：CMF 中断使能
13	MME	RW	0	静音模式使能。 0：接收器工作在工作模式 1：接收器在工作模式和静默模式之间切换
12	M0	RW	0	与 M1 结合配置字长度。

Puya Confidential

				只能在 LPUART 未使能时（UE=0）才能写入该位
11	WAKE	RW	0	接收唤醒方式。 从静默模式唤醒方式。由软件置位或者清零。 0：空闲帧唤醒 1：地址标志唤醒 只能在 LPUART 未使能时（UE=0）才能写入该位
10	PCE	RW	0	奇偶校验控制。 0：奇偶校验禁止 1：奇偶校验使能 奇偶校验位：9bit 数据的第 9 位；8bit 数据的第 8 位；7bit 数据的第 7 位。 只能在 LPUART 未使能时（UE=0）才能写入该位
9	PS	RW	0	奇偶校验选择。由软件置位和清零。 0：偶校验 1：奇校验 只能在 LPUART 未使能时（UE=0）才能写入该位
8	PEIE	RW	0	PE 中断使能。由软件置位和清零。 0：禁止 1：PE 中断使能
7	TXEIE	RW	0	传输寄存器空中断使能。由软件置位和清零。 0：禁止 1：TXE 中断使能
6	TCIE	RW	0	传送结束中断使能。由软件置位和清零。 0：禁止 1：TC 中断使能
5	RXNEIE	RW	0	接收寄存器非空中断使能；由软件置位和清零。 0：禁止 1：ORE 或者 RXNE/RXFNE 中断使能
4	IDLEIE	RW	0	IDLE 中断使能。由软件置位和清零。 0：禁止 1：IDLE 中断使能
3	TE	RW	0	传送使能。 0：传送禁止 1：传送使能
2	RE	RW	0	接收使能。 0：接收禁止 1：接收使能，开始检测 start 位
1	UESM	RW	0	停止低功耗模式下 LPUART 唤醒使能。 该寄存器为 1， LPUART 能唤醒 MCU。

Puya Confidential

				0：LPUART 不能唤醒 MCU 1：LPUART 模式唤醒 MCU。 从低功耗模式唤醒后，该位清零。
0	UE	RW	0	LPUART 使能。 当该位清零后，LPUART 模块会立即停止当前操作。 LPUART 的配置会保持，但所有状态标志位， LPUART_ISR 寄存器值会变为默认值。该位由软件置位和清零。 0：LPUART 预分频器和输出禁止，低功耗模式 1：LPUART 使能 软件需要等待 LPUART_ISR.TC 置位后，才能清零 UE 位，进入低功耗模式； 同时，在清零 UE 位之前 DMA 通道需要禁止。 该位清零时，LPUART 配置寄存器值维持，但 LPUART_ISR 状态寄存器全部复位。

30.4.2 LPUART 控制寄存器 2（LPUART_CR2）

偏移地址：0x04

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:0]								Res.	Res.	Res.	TXOE_ALWAYS_ON	MSBFIRST	DATAINV	TXINV	RXINV
RW	RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
RW		RW	RW								RW				

Bit	Name	RW	Reset Value	Function
31:24	ADD[7:0]	RW	8’b0	LPUART 地址。 该寄存器用于多处理器静默或者停止模式，用作地址唤醒时的地址。传送器发送的 MSB 位为 1. 在正常接收模式时，该寄存器用作字符检测。此时，比对接收内容与 ADD[7:0]相比较，如果匹配，则置位 CMF 标志位。 只有在 RE=0 或者 UE=0 时，该寄存器可以被写。
23:21	Reserved	-	-	保留
20	TXOE_ALWAYS_ON	RW	1	该寄存器用于控制 LPUART 输出使能。 0：硬件自主控制输出使能开/关 1：软件控制输出使能常开 只能在 LPUART 未使能时（UE=0）才能写入该位 注：在半双工模式，需要将此位关闭。
19	MSBFIRST	RW	0	最高有效位在前。 0：起始位后，收发第 0 位数据； 1：起始位后，收发第 7/8/9 位数据；

Puya Confidential

				只能在 LPUART 未使能时（UE=0）才能写入该位
18	DATAINV	RW	0	二进制数据逆处理。 0：以正/正向逻辑收发数据寄存器数据。（1=H，0=L） 1：以反/反向逻辑收发数据寄存器数据。奇偶位也会逆处理。（1=L，0=H） 只能在 LPUART 未使能时（UE=0）才能写入该位
17	TXINV	RW	0	TX 引脚有效电平反相。 0：TX 引脚为标准逻辑电平（V_{DD} =1/空闲, Gnd=0/标记） 1：TX 引脚信号值取反（V_{DD} =0/标记, Gnd=1/空闲） 只能在 LPUART 未使能时（UE=0）才能写入该位
16	RXINV	RW	0	RX 引脚有效电平反相。 0：RX 引脚为标准逻辑电平（V_{DD} =1/空闲, Gnd=0/标记） 1：RX 引脚信号值取反（V_{DD} =0/标记, Gnd=1/空闲） 只能在 LPUART 未使能时（UE=0）才能写入该位
15	SWAP	RW	0	TX/RX 引脚互换。 0：TX/RX 引脚按照标准引脚排列定义 1：TX/RX 引脚互换。此时用作交叉连接其他 LPUART 时 只能在 LPUART 未使能时（UE=0）才能写入该位
14	Reserved	-	-	保留
13:12	STOP[1:0]	RW	2’b0	停止位配置。 00：1 停止位 01，11：保留 10：2 停止位 只能在 LPUART 未使能时（UE=0）才能写入该位
11:5	Reserved	-	-	保留
4	ADDM7	RW	0	7 位/4 位地址检测配置。 0：4 位地址检测 1：7 位地址检测（8 位数据模式） 只能在 LPUART 未使能时（UE=0）才能写入该位
3:0	Reserved	-	-	保留

30.4.3 LPUART 控制寄存器 3（LPUART_CR3）

偏移地址：0x08

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUFIE	WUS[1:0]		Res.	Res.	Res.	Res.
									RW	RW	RW				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
RW	RW	RW	RW		RW	RW	RW	RW	RW			RW			RW

Bit	Name	RW	Reset Value	Function
31:23	Reserved	-	-	保留

Puya Confidential

22	WUFIE	RW	0	低功耗唤醒中断使能。 0：中断禁止 1：WUF 中断使能
21:20	WUS[1:0]	RW	0	低功耗唤醒选择。 00：地址匹配产生 WUF 01：保留 10：起始位检测产生 WUF 11：RXNE 产生 WUF
19:16	Reserved	-	-	保留
15	DEP	RW	0	DE 极性选择。 0：DE 高电平有效 1：DE 低电平有效 只能在 LPUART 未使能时（UE=0）才能写入该位
14	DEM	RW	0	DE 模式使能。 0：禁止 1：DE 模式使能，DE 信号在 RTS 引脚输出 用户通过使能该位来控制外部传输。 只能在 LPUART 未使能时（UE=0）才能写入该位
13	DDRE	RW	0	接收错误时是否禁止 DMA。 0：接收错误时不禁止 DMA。相应出错标志置位，但 RXNE=0. 不会产生 DMA 请求，所以出错数据不会输出。 1：接收错误时禁止 DMA。RXNE 和出错标志位都会置位。在出错标志位清零前，屏蔽 DMA 请求。 只能在 LPUART 未使能时（UE=0）才能写入该位
12	OVRDIS	RW	0	溢出禁止。 0：当接收到新数据前已收数据没读时置位溢出出错标志（PRE） 1：溢出功能禁止。接收到新数据后，RXNE=1，ORE=0，新数据会覆盖 LPUART_RDR 寄存器中数据 只能在 LPUART 未使能时（UE=0）才能写入该位
11	Reserved	-	-	保留
10	CTSIE	RW	0	CTS 中断使能。 0：禁止 1：CTSIF 中断使能
9	CTSE	RW	0	CTS 使能。 0：CTS 硬件流控制禁止 1：CTS 模式使能。当有当 CTS 输入为 0 时，才会传输数据。此时，当数据写入数据寄存器后，要等待 CTS 有效后才会启动传输 只能在 LPUART 未使能时（UE=0）才能写入该位
8	RTSE	RW	0	RTS 使能。 0：RTS 硬件流控制禁止

Puya Confidential

				1：RTS 输出使能，只有当接收缓冲区未满时才会请求下一个数据。当前数据发送完成后，发送操作暂停。如果可以接收数据了，将 RTS 置为有效（0） 只能在 LPUART 未使能时（UE=0）才能写入该位
7	DMAT	RW	0	传送时使能 DMA。 0：禁止 1：传送时使能 DMA
6	DMAR	RW	0	接收时使能 DMA。 0：禁止 1：接收时使能 DMA
5:4	Reserved	-	-	保留
3	HDSEL	RW	0	半双工选择。 0：非半双工模式 1：半双工模式选择 只能在 LPUART 未使能时（UE=0）才能写入该位
2:1	Reserved	-	-	保留
0	EIE	RW	0	错误中断使能。 0：禁止 1：帧错误 FE、溢出错误 ORE、噪声 NF 中断产生；

30.4.4 LPUART 波特率寄存器（LPUART_BRR）

偏移地址：0x0C

复位值：0x0000_0300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BBR[19:16]			
												RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BBR[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:20	Reserved	-	-	保留
19:0	BBR[19:0]	RW	0x300	LPUART 波特率。 禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值

30.4.5 LPUART 请求寄存器（LPUART_RQR）

偏移地址：0x18

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMRQ	SBKRQ	Res.
													W	W	

Bit	Name	RW	Reset Value	Function
31:3	Reserved	-	-	保留
2	MMRQ	W	0	静音模式请求。 该位写 1 使 LPUART 进入静默模式，并置位 RWU 标志。

Puya Confidential

1	SBKRQ	W	0	发送中止请求。 该位写 1 会置位 SBKF，并发送 BREAK 请求。
0	Reserved	-	-	保留

30.4.6 LPUART 中断和状态寄存器（LPUART_ISR）

偏移地址：0x1C

复位值：0x0080_00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
									R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
-					R	R	-	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:23	Reserved	-	-	保留
22	REACK	R	0	接收使能确认标志。 硬件置位/复位该寄存器。表明硬件在进入低功耗模式前准备好接收。 如果 LPUART 不支持从停止低功耗模式唤醒，则该位保留。
21	TEACK	R	0	发送使能确认标志。 硬件置位/复位该寄存器。 当在 LPUART_CR1 寄存器中写入 TE=0 生成空闲帧请求，然后写入 TE=1 以满足 TE=0 的最小周期时，可以使用该寄存器。
20	WUF	R	0	从低功耗模式唤醒标志。 当检测到唤醒事件时硬件写该位为 1. 事件通过 WUS 位域定义。通过向 LPUART_ICR 寄存器中的 WUCF 写入 1，此位由软件清零。如果 LPUART_CR3 寄存器中 WUFIE=1，则会生成中断。 注：当 UESM 清零时，WUF 标志也清零。
19	RWU	R	0	接收到静默模式唤醒。 当接收到静默模式序列，该寄存器置位；如果接收到唤醒序列，该寄存器清零。具体哪种唤醒序列（地址标记或者空闲帧）由寄存器 LPUART_CR1.WAKE 位控制。 0：接收器为工作模式； 1：接收器为静默模式；
18	SBKF	R	0	发送中止标志。 该寄存器表明请求发送断开字符。由软件写 1 到 LPUART_RQR.SBKRQ 寄存器置位该寄存器。硬件在发送完中止字符的停止位后清零该寄存器。 0：不发送断开字符； 1：发送断开字符；
17	CMF	R	0	地址匹配标志。 当接收到 ADD[7:0]值匹配的字符时置位该寄存器。软件写 1 到 CMCF 寄存器会清零该位。

Puya Confidential

16	BUSY	R	0	忙标志。 当 RX 线接收数据时（正确接收到起始位），硬件该寄存器置位。接收结束，硬件清零该寄存器。 0：LPUART 空闲 1：正在进行接收
15:11	Reserved	-	-	保留
10	CTS	R	0	CTS 标志。 0：CTS 线为 1 1：CTS 线为 0
9	CTSIF	R	0	CTS 中断标志。 0：CTS 状态线未改变 1：CTS 状态线值改变
8	Reserved	-	-	保留
7	TXE	R	1	传输寄存器空标志，可以写数据到 LPUART_TDR。 0：数据寄存器满 1：数据寄存器空
6	TC	R	1	传送完成标志。 传送数据帧完成后，且 TXE=1，则硬件置位该寄存器。TCIE=1 时产生中断。 软件写 1 到 TCCF 寄存器清零该位。 0：传送未完成 1：传送完成
5	RXNE	R	0	读数据寄存器不空标志。 当移位寄存器值传送到 LPUART_RDR 寄存器，硬件置位该寄存器。 软件读 LPUART_RDR 寄存器则清零该位。 当 RXNEIE=1 时，产生中断。 0：未收到数据 1：接收数据准备好读出
4	IDLE	R	0	空闲标志。 检测到空闲帧，硬件置位该寄存器。当 IDLEIE=1 时产生中断。 软件写 1 到 IDLECF 清零该位。 0：未检测到 IDLE 帧 1：检测到 IDLE 帧
3	ORE	R	0	溢出错误标志。 当 RXNE=1 时，在移位寄存器中接收到的数据正准备转移到 RDR 寄存器时，硬件设置该位。 软件写 1 到 ORECF 寄存器清零该位。 当 RXNEIE=1 或者 EIE=1 时，产生中断。 注：该寄存器置位时，RDR 寄存器内容不会丢失，但移位寄存器内容被覆盖。 当 EIE=1 时，产生 ORE 中断。
2	NE	R	0	START 位噪声标志。 在数据帧接收到噪声时，硬件置位该寄存器。

Puya Confidential

				软件写 NFCF=1 会清零该位。
1	FE	R	0	帧错误标志。 当检测到不同步、过多的噪声或中断字符时，由硬件设置此位。 软件写 FECF=1 清零该位。 当 EIE=1 时，产生中断。
0	PE	R	0	校验值错误。 当接收时校验值错误时，硬件置位该寄存器。 软件写 PECF=1 清零该位。 当 PEIE=1 时，产生中断。

30.4.7 LPUART 中断标志清零寄存器（LPUART_ICR）

偏移地址：0x20

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											W			W	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
						W			W		W	W	W	W	W

Bit	Name	RW	Reset Value	Function
31:21	Reserved	-	-	保留
20	WUCF	W	0	低功耗唤醒唤醒标志清零。 软件写 1 会清零 WUF 寄存器。
19:18	Reserved	-	-	保留
17	CMCF	W	0	地址匹配标志清零。 软件写 1 会清零 CMF 寄存器。
16:10	Res			
9	CTSCF	W	0	CTS 标志清零。 软件写 1 清零 CTSIF 寄存器。
8:7	Reserved	-	-	保留
6	TCCF	W	0	传送完成标志清零。 软件写 1 清零 TC 寄存器。
5	Reserved	-	-	保留
4	IDLECF	W	0	空闲标志清零。 软件写 1 清零 IDLEF 寄存器。
3	ORECF	W	0	溢出错误标志清零。 软件写 1 清零 ORE 寄存器。
2	NECF	W	0	噪声标志清零。 软件写 1 会清零 NE 寄存器。
1	FECF	W	0	帧错误标志清零。 软件写 1 清零 FE 寄存器。
0	PECF	W	0	校验值错误标志清零。 软件写 1 清零 PE 寄存器。

Puya Confidential

30.4.8 LPUART 接收数据寄存器（LPUART_RDR）

偏移地址：0x24

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:9	Res			
8:0	RDR[8:0]	R	0	接收数据寄存器。

30.4.9 LPUART 发送数据寄存器（LPUART_TDR）

偏移地址：0x28

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
							R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:9	Reserved	-	-	保留
8:0	TDR[8:0]	RW	0	发送数据寄存器。 当校验使能时，该寄存器的 bit7 或者 bit8 在发送时被校验位代替。 注意：建议先开启 TE 和 UE 之后，再写 TDR 数据。

30.4.10 LPUART 预分频器寄存器（LPUART_PRESC）

偏移地址：0x2C

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
												RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:4	Reserved	-	-	保留
3:0	PRESCALER	RW	0	输入时钟预分频寄存器。 0000：不分频 0001：2 分频 0010：4 分频 0011：6 分频 0100：8 分频 0101：10 分频 0110：12 分频

Puya Confidential

				0111：16 分频 1000：32 分频 1001：64 分频 1010：128 分频 1011：256 分频 其它：256 分频

30.4.11 LPUART 采样模式选择寄存器（LPUART_MOD）

偏移地址：0x34

复位值：0x0000_0000

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

Bit	Name	RW	Reset Value	Function
31:1	Reserved	-	-	保留
0	SAMPLE_MOD	RW	0	0：采样模式 0，建议 1200,2400 等低波特率使用。 1：采样模式 1，建议 4800,9600 等高波特率使用。

Puya Confidential

31. FD 控制器局域网路(FDCAN)

31.1 FDCAN 简介

控制器局域网络（CAN）子系统由一个 CAN 模块、一个共享消息 RAM 存储器组成。

CAN 模块符合 ISO 11898-1: 2015（CAN 协议规范第 2.0 版 A、B 部分）和 FD CAN 协议规范第 1.0 版（CAN with Flexible Data-Rate Specification Version 1.0）。

0.8 KB 的消息 RAM 存储器可实现接收过滤器（Rx Filter）、接收 FIFO（Rx FIFO）、接收缓冲区（Rx Buffer）、发送事件 FIFO（Tx Event FIFO）、发送缓冲区（Tx Buffer）功能。

31.2 FDCAN 主要特性

- 适用 ISO 11898-1:2015 标准
- FD CAN 最多支持收发 64 个数据字节
- 支持 CAN 错误日志
- 支持 AUTOSAR 和 SAE J1939
- 支持接收过滤功能
- 两个可配置的接收 FIFO
- 接收到高优先级消息时单独发出信号指示
- 最多 3 个专用接收缓冲区
- 最多 3 个专用发送缓冲区
- 可配置的发送 FIFO 或队列
- 可配置的发送事件 FIFO
- 主机 CPU 可直接访问消息 RAM
- 支持可编程回环测试模式
- 支持可屏蔽模块中断
- 两个时钟域：CAN 内核时钟和 APB 总线接口时钟
- 支持调试

Puya Confidential

31.3 FDCAN 功能描述

31.3.1 FDCAN 框图

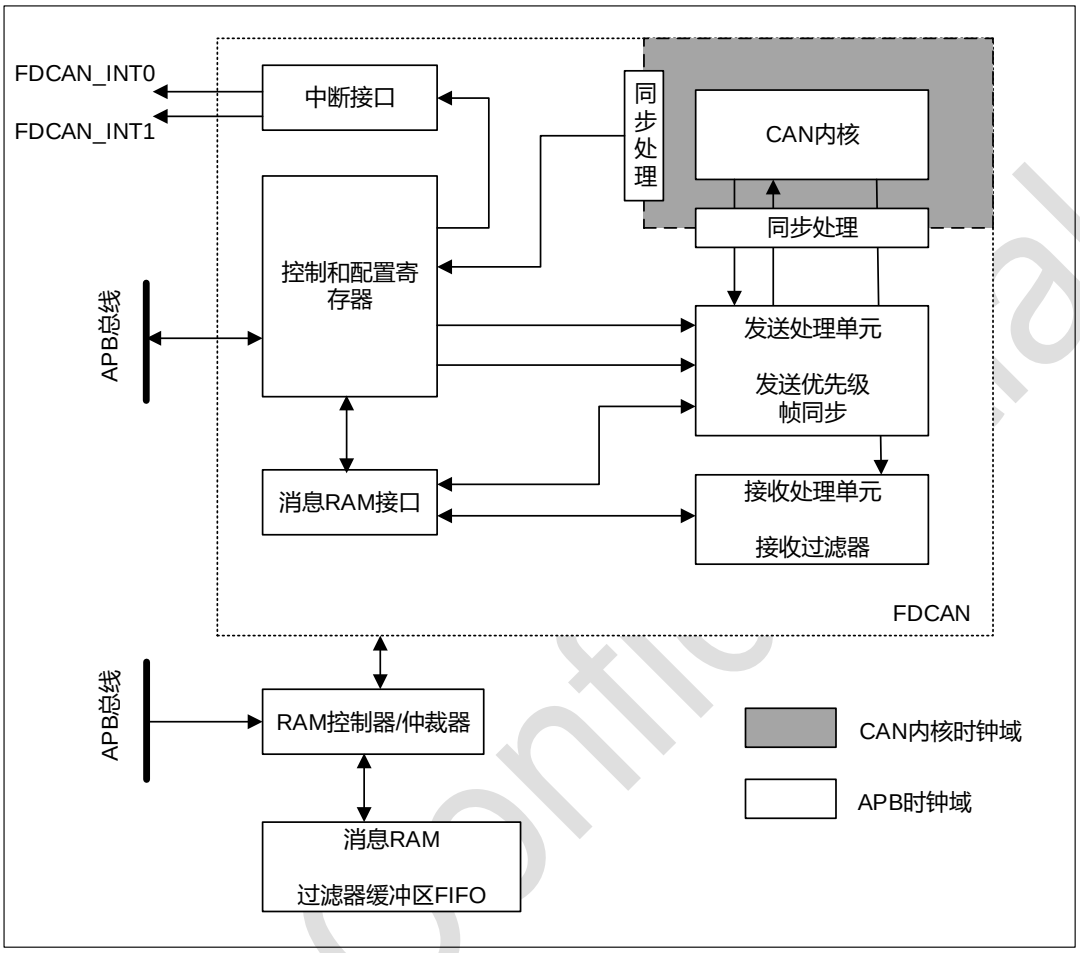


图 31-1 FDCAN 框图

CAN 内核

CAN 内核包含协议控制器、接收与发送移位寄存器。它可处理 ISO 11898-1:2015 的所有协议功能，并支持 11 位标准 ID 和 29 位扩展 ID。

同步处理

同步模块将来自 APB 时钟域的信号同步到 CAN 内核时钟域，以及将来自 CAN 内核时钟域的信号同步到 APB 时钟域。

时钟

APB 时钟域的时钟为 FDCAN 控制逻辑时钟，CAN 内核时钟域的时钟为 FDCAN 通信时钟。为了保证 FDCAN 通信的稳定性，控制逻辑时钟频率必须大于等于通信时钟频率。

发送处理单元

该处理单元控制消息从外部消息 RAM 到 CAN 内核的传输。最多可配置 32 个发送缓冲区进行发送。发送缓冲区可用作专用发送缓冲区、发送 FIFO（或队列），或二者的组合。发送事件 FIFO 将发送时间戳与对应的消息 ID 存储在一起。另外还支持取消发送。

Puya Confidential

接收处理单元

该处理单元控制接收到的消息从 CAN 内核到外部消息 RAM 的传输。接收处理单元支持两个接收 FIFO（每个 FIFO 的大小均可配置）和最多 64 个专用接收缓冲区，用于存储所有通过接收过滤的消息。与接收 FIFO 有所不同，专用接收缓冲区仅用于存储特定 ID 的消息。接收时间戳与其消息存储在一起。对于 11 位标准 ID，最多可定义 128 个过滤器；对于 29 位扩展 ID，最多可定义 64 个过滤器。

APB 接口

将 FDCAN 连接至 APB 总线。

消息 RAM

接口通过 RAM 控制器和仲裁器将 FDCAN 访问连接到外部 0.8 KB 消息 RAM。

双中断线

FDCAN 提供 FDCAN_INT0 和 FDCAN_INT1 两条中断线。对 FDCAN_ILE.EINT0 和 FDCAN_ILE.EINT1 位进行编程，可分别使能或禁止这两条中断线。

31.3.2 FDCAN 工作模式

31.3.2.1 正常工作

FDCAN 完成初始化(初始化流程参见“37.4 FDCAN 软件初始化”章节)且 FDCAN_CCCR.INIT 清零后，FDCAN 会将其自身与 CAN 总线同步并准备好进行通信。接收消息时，通过接收过滤器接收到的消息（包含消息 ID 和 DLC）会存储到专用的接收缓冲区，或接收 FIF0，或接收 FIF1。发送消息时，先初始化或更新专用发送缓冲区和（或）发送 FIFO（队列）。不支持在接收到远程帧后自动发送。

31.3.2.2 FDCAN 操作

FDCAN 帧传输有两种类型，第一种是不带比特率切换的 FDCAN 帧，第二种是带比特率切换的 FDCAN 帧（控制段、数据段和 CRC 段的传输比特率比帧开始和结束时的高）。

经典 CAN（Classical CAN）标准帧中的 r0 位（IDE 位之后，标准帧格式中的保留位）、扩展帧中的 r1 位（RTR 位之后，扩展帧格式中的第一个保留位），在 FDCAN 帧格式中解码为 FDF 位。FDF 位为隐性（逻辑 1）表示 FDCAN 帧，为显性（逻辑 0）表示经典 CAN 帧。在 FDCAN 帧中，FDF 位之后的 res 和 BRS（BitRateSwitch，比特率切换）两个位，决定是否切换 FDCAN 帧中的比特率，并通过 res 显性和 BRS 隐性指示已切换比特率。res 位隐性的编码保留，供将来协议扩展使用。如果 FDCAN 接收到的帧中 FDF 位和 res 位都是隐性，则会通过将 FDCAN_PSR.PXE 置位的方式发出协议异常事件。如果使能协议异常处理（FDCAN_CCCR.PXHD=0），FDCAN 将在下一采样点将工作状态从接收（FDCAN_PSR.ACT=0b10）变为同步（FDCAN_PSR.ACT=0b00）。如果禁止协议异常处理（FDCAN_CCCR.PXHD=1），FDCAN 会将隐性 res 位当作格式错误处理，并将以错误帧进行响应。

FD CAN 模式通过编程 FDCAN_CCCR.FDOE 来使能。如果 FDCAN_CCCR.FDOE=1，则会使能 FD CAN 帧的发送和接收。经典 CAN 帧始终可进行发送和接收。是发送 FD CAN 帧还是经典 CAN 帧，可通过相应发送缓冲区元素中的 FDF 位配置。如果 FDCAN_CCCR.FDOE=0，接收到的帧会被当作经典 CAN 帧，从而会在接收到 FD CAN 帧时发送错误帧。如果 FD CAN 模式已禁止，即使发送缓冲区元素的 FDF 位置 1，也不会发送 FD CAN 帧。仅当 FDCAN_CCCR.INIT 和 FDCAN_CCCR.CCE 均置位

Puya Confidential

时，才能更改 FDCAN_CCCR.FDOE 和 FDCAN_CCCR.BRSE。

如果 FDCAN_CCCR.FDOE=0，则会忽略 FDF 和 BRS 位的设置，帧以经典型 CAN 格式发送。如果 FDCAN_CCCR.FDOE=1 且 FDCAN_CCCR.BRSE=0，则仅会评估发送缓冲区元素的 FDF 位。如果 FDCAN_CCCR.FDOE=1 且 FDCAN_CCCR.BRSE=1，则会使能比特率切换发送 FD CAN 帧。所有 FDF 和 BRS 位置位的发送缓冲区元素均会以比特率切换的 FD CAN 格式发送。

建议仅当以下条件满足时，才在 CAN 操作期间切换模式：

- FD CAN 数据阶段的故障率显著高于 FD CAN 仲裁阶段的故障率。在这种情况下，应禁止发送的 FD CAN 比特率切换。
- 系统启动期间，所有节点在被验证能够以 FD CAN 格式进行通信之前，都将发送经典 CAN 消息。如果通过验证，所有节点会切换为 FD CAN 操作。
- CAN 局部网络（CAN Partial Networking）中的唤醒消息必须以经典 CAN 格式发送。
- EOL 编程（End-of-line programming）时，并非所有节点都支持 FD CAN，非 FD CAN 的节点会保持在静默模式直到编程结束。随后，所有节点切换回经典 CAN 通信。

在 FD CAN 格式中，DLC 代码 0 到 8 的编码（对应的数据字节长度）与经典 CAN 相同，代码 9 到 15 的编码与经典 CAN 不同，如下表所示

表 31-1 FDCAN 中的 DLC 编码

DLC 编码		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
数据字节数	经典 CAN	0	1	2	3	4	5	6	7	8	8	8	8	8	8	8	8
	FD CAN	0	1	2	3	4	5	6	7	8	12	16	20	24	32	48	64

在 FD CAN 帧中，如果 BRS 位为隐性，则位时长将在 BRS（比特率切换）位后，在帧内切换。在 BRS 位之前，在 FD CAN 仲裁段，位时长按照寄存器 FDCAN_NBTP（Nominal Bit Timing & Prescaler Register，标称位时长和预分频寄存器）的定义执行。在接下来的 FD CAN 数据段，位时长按照寄存器 FDCAN_DBTP（Data Bit Timing & Prescaler Register，数据位时长和预分频寄存器）的定义执行。位时长在 CRC 分隔符之后，或检测到错误时从数据段位时长切换回来，以先发生的事件为准。

FD CAN 数据段的最大可配置比特率取决于 CAN 通信时钟频率。举例来说，如果 CAN 通信时钟频率为 20MHz，最短可配置的位时间为 4 个时间片（tq），则数据阶段的比特率为 5 Mbps。

在带比特率切换和不带比特率切换的 FD CAN 两种数据帧格式中，位 ESI（Error Status Indicator，错误状态指示符）的值，由发送开始时发送节点的错误状态决定。如果发送节点为错误被动状态，则 ESI 以隐性位发送，否则以显性位发送。

31.3.2.3 收发器延迟补偿

在 FD CAN 传输的数据段，只有一个节点进行发送，其他所有节点都是接收节点。总线的长度没有影响。当通过引脚 FDCAN_TX 发送时，FDCAN 通过引脚 FDCAN_RX 从其本地 CAN 收发器接收发送的数据。接收数据因发送器延迟而延迟，如果该延迟大于 TSEG1（采样点之前的时间段），则会检测到位错误。为了使数据段位时间可以比发送器延迟更短，引入了延迟补偿。如果没有发送器延迟补偿， FD CAN 帧数据段的比特率将受到发送器延迟的限制。

FDCAN 的协议单元实现了延迟补偿机制来补偿发送器延迟，从而实现在 FD CAN 数据段以更高的比特率进行传输，而不受特定 CAN 收发器延迟的影响。

Puya Confidential

为了检查发送节点的数据段是否存在位错误，会将延迟的发送数据与第二采样点 SSP 处的接收数据进行比较。如果检测到位错误，发送节点将在下一个常规采样点对该位错误作出反应。在仲裁段，延迟补偿始终被禁用。

发送器延迟补偿支持数据位时间短于发射器延迟的配置， ISO 11898-1:2015 对此进行了详细描述，它通过置位 FDCAN_DBTP.TDC 使能。

在第二采样点 SSP 将接收到的位与发送的位进行比较。 SSP 的位置定义为，从 FDCAN 发送输出引脚 FDCAN_TX 通过收发器到接收输入引脚 FDCAN_RX 测得的延迟，与由 FDCAN_TDCR.TDCO 配置的发送器延迟补偿偏移之和。发送器延迟补偿偏移，用于调整 SSP 在已接收到的位中的位置（例如数据段的位时间的一半）。第二采样点的位置向下取整到下一个整数 mtq（Minimum time quantum，即最小时间片，为一个 FDCAN 通信时钟周期）。

FDCAN_PSR.TDCV 显示实际的发送器延迟补偿值。当 FDCAN_CCCR.INIT 置位时，FDCAN_PSR.TDCV 被清零; 当 FDCAN_DBTP.TDC 置位时，FDCAN_PSR.TDCV 在每次发送 FD 帧时都会更新。

对于在 FDCAN 中实现的发送器延迟补偿，必须考虑以下边界条件：

- 测得的 FDCAN_TX 到 FDCAN_RX 的延迟与配置的发送器延迟补偿偏移 FDCAN_TDCR.TDCO 之和，必须小于数据段的 6 个位时间。
- 测得的 FDCAN_TX 到 FDCAN_RX 的延迟与配置的发送器延迟补偿偏移 FDCAN_TDCR.TDCO 之和，必须小于或等于 127mtq。如果此和大于 127mtq，则发送器延迟补偿使用最大值 127mtq。
- 数据段在 CRC 分隔符的采样点结束，将停止在 SSP 检查接收位。

如果通过编程 FDCAN_DBTP.TDC=1 使能发送器延迟补偿，则会在每个 FD CAN 帧的 FDF 位到 res 位的下降沿处开始测量。当在发送节点的 FDCAN_RX 引脚上检测到该边沿时，停止测量。该测量的分辨率为一个 mtq。

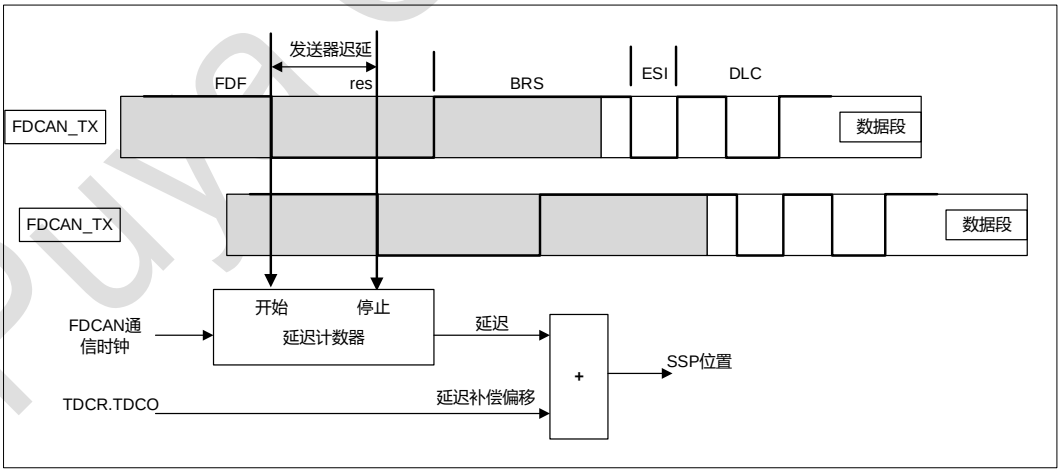


图 31-2 FDCAN 收发器延迟测量

为了避免接收到的 FDF 位中的显性毛刺，在接收到的 RES 位的下降沿之前结束延迟补偿测量，从而导致 SSP 位置提前，可通过编程 FDCAN_TDCR.TDCF 来使能发送器延迟补偿滤波窗口，从而定义 SSP 位置的最小值。FDCAN_RX 引脚上的显性边缘会导致 SSP 位置更加提前，但对于发射器延迟测量而言这将被忽略。当 SSP 位置至少为 FDCAN_TDCR.TDCF 且 FDCAN_RX 引脚为低电平时，测量停止。

Puya Confidential

31.3.2.4 受限工作模式

在受限工作模式下，节点能够接收数据帧和远程帧，并能对有效帧进行确认，但不会发送数据帧、远程帧、主动错误帧或过载帧。在出现错误或过载情况时，不会发送显性位，而是等待出现总线空闲条件，以重新同步到 CAN 通信。错误记录(FDCAN_TDCR.CEL)动作时，错误计数器(FDCAN_TDCR.REC、FDCAN_TDCR.TEC)被冻结。主机可通过置位 FDCAN_CCCR.ASM 将 FDCAN 设置为受限工作模式，仅当 FDCAN_CCCR.CCE 和 FDCAN_CCCR.INIT 均设为 1 时，主机才能将 FDCAN_CCCR.ASM 置位。FDCAN_CCCR.ASM 可随时通过主机清零。

当发送处理单元无法及时从消息 RAM 中读取数据时，会自动进入受限工作模式。要退出受限工作模式，主机 CPU 必须清零 FDCAN_CCCR.ASM。

受限工作模式可用于适应不同 CAN 比特率的应用。在这种情况下，应用程序会测试不同的比特率，并在收到有效帧后退出受限操作模式。

注：受限工作模式不得与回环模式（内部或外部）结合使用。

31.3.2.5 总线监听模式

配置 FDCAN_CCCR.MON 为 1，将 FDCAN 设置为总线监听模式。在总线监听模式下（见 ISO 11898-1:2015,10.14 Bus monitoring），FDCAN 能够接收有效数据帧和有效远程帧，但不能启动发送。在该模式下，FDCAN 仅在 CAN 总线上发送隐性位，如果 FDCAN 需要发送一个显性位（ACK 位、过载标志、主动错误标志），该位将在内部被改道发送，以便 FDCAN 可以监听该显性位，但 CAN 总线可以保持隐性状态。在总线监听模式下，寄存器 FDCAN_TXBRP 保持复位状态。

总线监听模式可用于分析 CAN 总线上的流量，同时又不会因发送显性位而对总线造成影响。 下图显示了总线监听模式下 FDCAN_TX 和 FDCAN_RX 信号与 FDCAN 的连接。

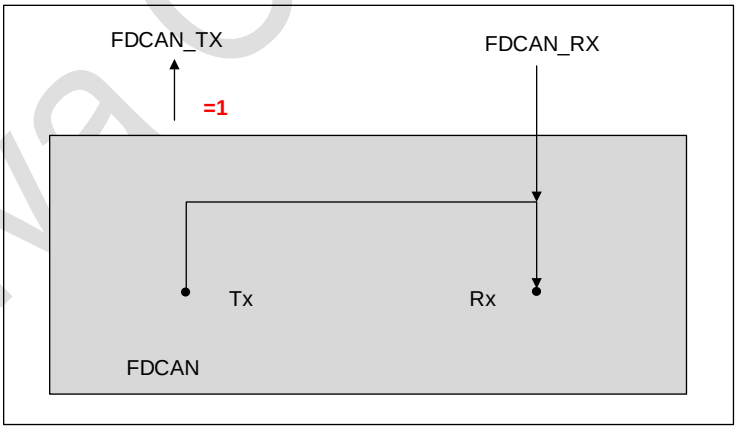


图 31-3 FDCAN 总线监听模式下的引脚控制

31.3.2.6 禁止自动重发（DAR）模式

根据 CAN 规范（见 ISO 11898-1:2015, 8.3.4 Recovery Management），FDCAN 可自动重发仲裁失败或在发送期间被错误干扰的帧。默认使能自动重发。

在 DAR 模式下，所有发送在 CAN 总线上启动后都会自动取消。发送缓冲区发送请求挂起位

Puya Confidential

FDCAN_TXBRP.TRPx 在成功发送后复位，如果取消时发送尚未开始、发送已因仲裁失败而中止、或帧发送期间发生错误，该位也会复位。

- 成功发送：
  - 对应的发送缓冲区发送已发生位 FDCAN_TXBTO.TOx 置位
  - 对应的发送缓冲区取消已完成位 FDCAN_TXBCF.CFx 不置位
- 即使已取消但也成功发送：
  - 对应的发送缓冲区发送已发生位 FDCAN_TXBTO.TOx 置位
  - 对应的发送缓冲区取消已完成位 FDCAN_TXBCF.CFx 置位
- 仲裁失败或帧发送受到干扰：
  - 对应的发送缓冲区发送已发生位 FDCAN_TXBTO.TOx 不置位
  - 对应的发送缓冲区取消已完成位 FDCAN_TXBCF.CFx 置位

如果帧发送成功，并且使能了发送事件的存储，则会写入发送事件 FIFO 元素，其中事件类型 ET=0b10（即使已取消也会进行发送）。

31.3.2.7 掉电（休眠模式）

FDCAN 可通过 CC 控制寄存器 FDCAN_CCCR.CSR 设置为掉电模式。只要时钟停止请求有效，FDCAN_CCCR.CSR 位就读为 1。

当所有挂起的发送请求都完成后，FDCAN 会一直等待直到检测到总线空闲状态。检测到总线空闲状态后，FDCAN 将 FDCAN_CCCR.INIT 设置为 1 以防止任何进一步的 CAN 传输。然后，FDCAN 通过将 FDCAN_CCCR.CSA 置 1 确认其已准备好进入掉电状态。在模块时钟（控制逻辑时钟和通信时钟）关闭之前，可继续访问寄存器，但 FDCAN_CCCR.INIT 保持为 1。

注：在 CAN 总线受到严重干扰的情况下，可能永远不会达到空闲状态，因此 FDCAN 不会置位 FDCAN_CCCR.INIT。这种情况可通过轮询 FDCAN_PSR.ACT 来检测。如果 FDCAN 没有进入空闲状态，软件可以写 CCCR.INIT=1，这将立即停止 FDCAN 的 CAN 通信，无论是否正在进行发送或接收。

要退出掉电模式，应用程序必须在复位 CC 控制寄存器标志 FDCAN_CCCR.CSR 之前开启模块时钟。FDCAN 将通过复位 FDCAN_CCCR.CSA 确认时钟已开启。之后，应用程序可通过复位 FDCAN_CCCR.INIT 重新开始 CAN 通信。

31.3.2.8 测试模式

要使能对 FDCAN 测试寄存器 FDCAN_TEST（请参见章节[FDCAN 测试寄存器（FDCAN_TEST）]）的写访问，必须将 FDCAN_CCCR.TEST 置 1，从而能够配置测试模式和测试功能。

通过编程 FDCAN_TEST.TX，发送引脚 FDCAN_TX 可实现四种输出功能。除了串行数据输出这一默认功能之外，还可驱动 CAN 采样点信号来监测 FDCAN 的位时长，并可输出恒定的显性或隐性电平。从 FDCAN_TEST.RX 可读取引脚 FDCAN_RX 的实际值。这两种功能（FDCAN_TEST.TX 和 FDCAN_TEST.RX）均可用于检查 CAN 总线的物理层。

由于 CAN 通信时钟与控制逻辑时钟之间的同步机制，从配置 FDCAN_TEST.TX 到引脚 FDCAN_TX 实现指定功能之间，可能有几个控制逻辑时钟周期的延迟。这一点还适用于通过 FDCAN_TEST.RX 读

Puya Confidential

取引脚 FDCAN_RX 的情况。

注：

测试模式仅限用于生产测试和自检。对 FDCAN_TX 引脚进行软件控制会干扰所有 CAN 协议功能。不建议实际应用时使用测试模式。

31.3.2.9 外部回环模式

配置 FDCAN_TEST.LBCK 为 1，将 FDCAN 设置为外部回环模式。在回环模式下，FDCAN 将其自身发送的消息作为接收的消息来处理，并将消息存储（如果这些消息通过了接收过滤）到接收缓冲区或接收 FIFO 中。下图显示了外部回环模式下 FDCAN_TX 和 FDCAN_RX 引脚信号与 FDCAN 的连接。

该模式用于硬件自检。为了不受外部影响，FDCAN 在回环模式下忽略 ACK 错误（在数据帧或远程帧的 ACK 域采样到隐性位）。在此模式下，FDCAN_TX 输出的信号在内部直接反馈到接收处理单元，FDCAN_RX 输入引脚的实际值被忽略。从引脚 FDCAN_TX 可以监听到发送的消息。

31.3.2.10 内部回环模式

配置 FDCAN_TEST.LBCK 和 FDCAN_CCCR.MON 为 1，将 FDCAN 设置为内部环回模式。该模式可用于“热自检（Hot Selftest）”，也就是说，FDCAN 可以进行检测，同时又不会影响接入的正在运行的 CAN 系统。在此模式下，引脚 FDCAN_RX 与 FDCAN 断开连接，引脚 FDCAN_TX 保持隐性。下图显示了内部环回模式下 FDCAN_TX 和 FDCAN_RX 引脚信号与 FDCAN 的连接。

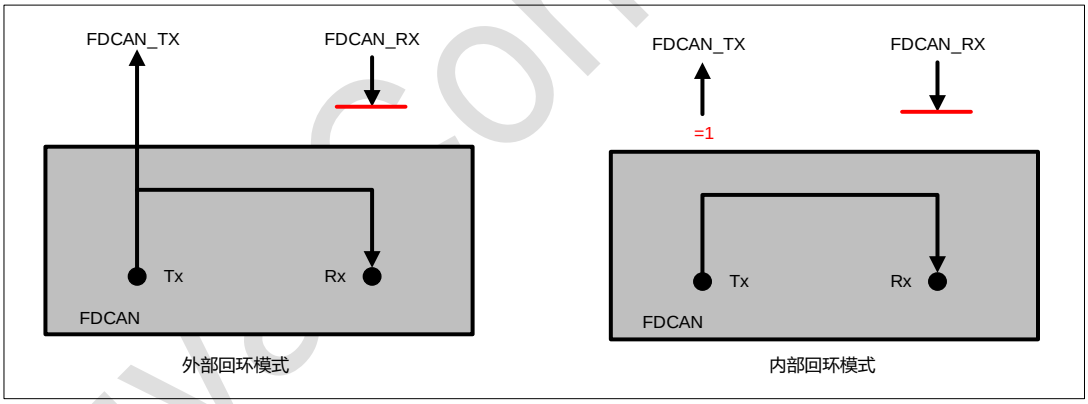


图 31-4 FDCAN 回环模式的引脚控制

31.3.3 FDCAN 时间戳生成

FDCAN 提供了一个 16 位循环计数器来生成内部时间戳。计数时间单位可通过预分频器 FDCAN_TSCC.TCP 配置为 CAN 位时间 1~16 倍。计数器值可通过 FDCAN_TSCV.TCV 读取。对寄存器 FDCAN_TSCV 进行写访问会将计数器复位为 0。时间戳计数器循环时，中断标志 FDCAN_IR.TSW 会置位。

在接收到或发送 SOF（Start Of Frame，帧起始）时，计数值被捕获并存储到接收缓冲区或接收 FIFO（RXTS[15:0]）或发送事件 FIFO（TXTS[15:0]）元素的时间戳部分。

Puya Confidential

31.3.4 FDCAN 超时计数器

FDCAN 提供了一个 16 位超时计数器来指示接收 FIF0、接收 FIF1 和发送事件 FIFO 的超时条件。它作为递减计数器工作，并且与时间戳计数器一样，使用 FDCAN_TSCC.TCP 控制的预分频器。超时计数器通过寄存器 FDCAN_TOCC 配置。实际计数器值可从 FDCAN_TOCV.TOC 读取。仅当 FDCAN_CCCR.INIT=0 时，才会启动超时计数器。当 FDCAN_CCCR.INIT=1 时，超时计数器停止计数，例如当 CAN 进入 Bus_Off 状态时。

工作模式通过 FDCAN_TOCC.TOS 选择。在连续模式下工作时，计数器在 FDCAN_CCCR.INIT 复位时启动。对 FDCAN_TOCV 进行写操作，会将计数器预设为 FDCAN_TOCC.TOP 配置的值，并继续递减计数。

当超时计数器由其中一个 FIFO 控制时，空 FIFO 会将计数器预设为 FDCAN_TOCC.TOP 配置的值。存储了第一个 FIFO 元素后，开始递减计数。对 FDCAN_TOCV 执行写入操作不起作用。

计数器达到 0 时，中断标志 FDCAN_IR.TOO 置位。在连续模式下，计数器在达到 FDCAN_TOCC.TOP 后立即重启。

注：超时计数器的时钟信号来自 CAN 内核采样点信号。因此，由于 CAN 内核同步（重同步）机制的原因，超时计数器递减的时间点可能有所不同。如果在 FD CAN 中使用了比特率切换功能，则超时计数器在仲裁字段和数据字段中使用不同的时钟计数。

31.3.5 FDCAN 接收处理

接收处理单元控制着接收过滤、已接收消息到接收缓冲区或其中一个接收 FIFO（共两个）的传输以及接收 FIFO 放入和获取索引。

31.3.5.1 接收过滤

FDCAN 能够配置两组接收过滤器，一组供标准 ID 使用，另一组供扩展 ID 使用。这两组过滤器可分配给接收缓冲区、接收 FIFO0 或接收 FIFO1。对于接收过滤，每个过滤器列表从第 0 个元素开始执行比较，直到第一个匹配的元素。接收过滤在第一个匹配的元素处停止。不会为此消息评估剩下的过滤器元素。

- 主要特性有：
  - 每个过滤器元素可配置为
    * ID 范围过滤器（起止范围）
    * 一个或两个专用 ID 的过滤器
    * ID 位屏蔽的经典过滤器
  - 每个过滤器元素均可配置为实现接收过滤或拒绝过滤
  - 每个过滤器元素均可单独使能/禁止
  - 按顺序比较过滤器元素，到第一个匹配的过滤器元素后停止执行
- 相关配置寄存器位：
  - 全局过滤器配置（FDCAN_GFC）
  - 标准 ID 过滤器配置（FDCAN_SIDFC）
  - 扩展 ID 过滤器配置（FDCAN_XIDFC）

Puya Confidential

- 扩展 ID 与运算掩码（FDCAN_XIDAM）

■ 根据过滤器元素的配置（SFEC/EFEC），发生匹配时会触发下列操作之一：

- 将接收到的帧存储到接收 FIFO0 或 FIFO1 中
- 将接收到的帧存储到接收缓冲区中
- 拒绝接收到的帧
- 将高优先级消息中断标志 FDCAN_IR.HPM 置 1

将高优先级消息中断标志 FDCAN_IR.HPM 置 1，并将接收到的帧存储到接收 FIF0 或 FIF1 中接收到完整的 ID 后，开始接收过滤。接收过滤完成后，如果找到匹配的接收缓冲区或接收 FIFO，消息处理单元便会开始将接收到的消息数据以 32 位的形式写入匹配的接收缓冲区或接收 FIFO 中。如果 CAN 协议控制器检测到错误条件（例如 CRC 错误），则会丢弃此消息，丢弃后的影响如下：

接收缓冲区

匹配接收缓冲区的新数据标志不会置 1，但接收缓冲区（部分）会被接收到的数据覆盖。有关错误类型，请参见 FDCAN_PSR.LEC 和 FDCAN_PSR.DLEC。

接收 FIFO

匹配接收 FIFO 的放入索引不会更新，但相关接收 FIFO 元素（部分）会被接收到的数据覆盖。有关错误类型，请参见 FDCAN_PSR.LEC 和 FDCAN_PSR.DLEC。如果匹配的接收 FIFO 在覆盖模式下工作，必须考虑接收 FIFO 覆盖模式中介绍的边界条件。

注：当接收的消息写入两个接收 FIFO 之一或写入接收缓冲区中时，未匹配的已接收 ID 会独立于所用过滤器之外进行存储。接收过滤的过程很大程度上取决于所配置的过滤器元素的顺序。

31.3.5.2 ID 范围过滤器

该过滤器会将所有接收到的帧与 SF1ID/SF2ID 和 EF1ID/EF2ID 定义的范围内的消息 ID 进行匹配检查。范围过滤与扩展帧一起使用时，有两种可能的情况：

■ EFT=0b00：应用范围过滤器之前，会将已接收帧的消息 ID 与扩展 ID 与运算掩码（FDCAN_XIDAM）进行与运算

■ EFT=0b11：不会为范围过滤使用扩展 ID 与运算掩码（FDCAN_XIDAM）

31.3.5.3 专用 ID 过滤器

可将过滤器元素配置为过滤一个或两个特定的消息 ID。要过滤一个特定的消息 ID，过滤器元素必须配置为 SF1ID=SF2ID 和 EF1ID=EF2ID。

31.3.5.4 ID 位屏蔽过滤器

ID 位屏蔽过滤用于通过屏蔽已接收消息 ID 的某些位来过滤消息 ID 组。进行 ID 位屏蔽过滤时，SF1ID/EF1ID 用作消息 ID 过滤器，SF2ID/EF2ID 用作过滤器 ID 掩码（ID Mask）。

ID 掩码中为 0 的位将屏蔽掉已配置 ID 过滤器的对应位，即，已接收消息 ID 在该位位置的值不与接收过滤器对应位的值进行比较，只有对应掩码位为 1 的已接收消息 ID 的位才进行比较。

如果所有掩码位均为 1，则仅当接收到的消息 ID 和消息 ID 过滤器完全相同时，才会出现匹配。如果

Puya Confidential

所有掩码位均为 0，则所有消息 ID 都匹配。

31.3.5.5 标准消息 ID 过滤

下图介绍了标准消息 ID（11 位 ID）的过滤流程。章节【标准消息 ID 过滤器元素】介绍了标准消息 ID 过滤器元素。

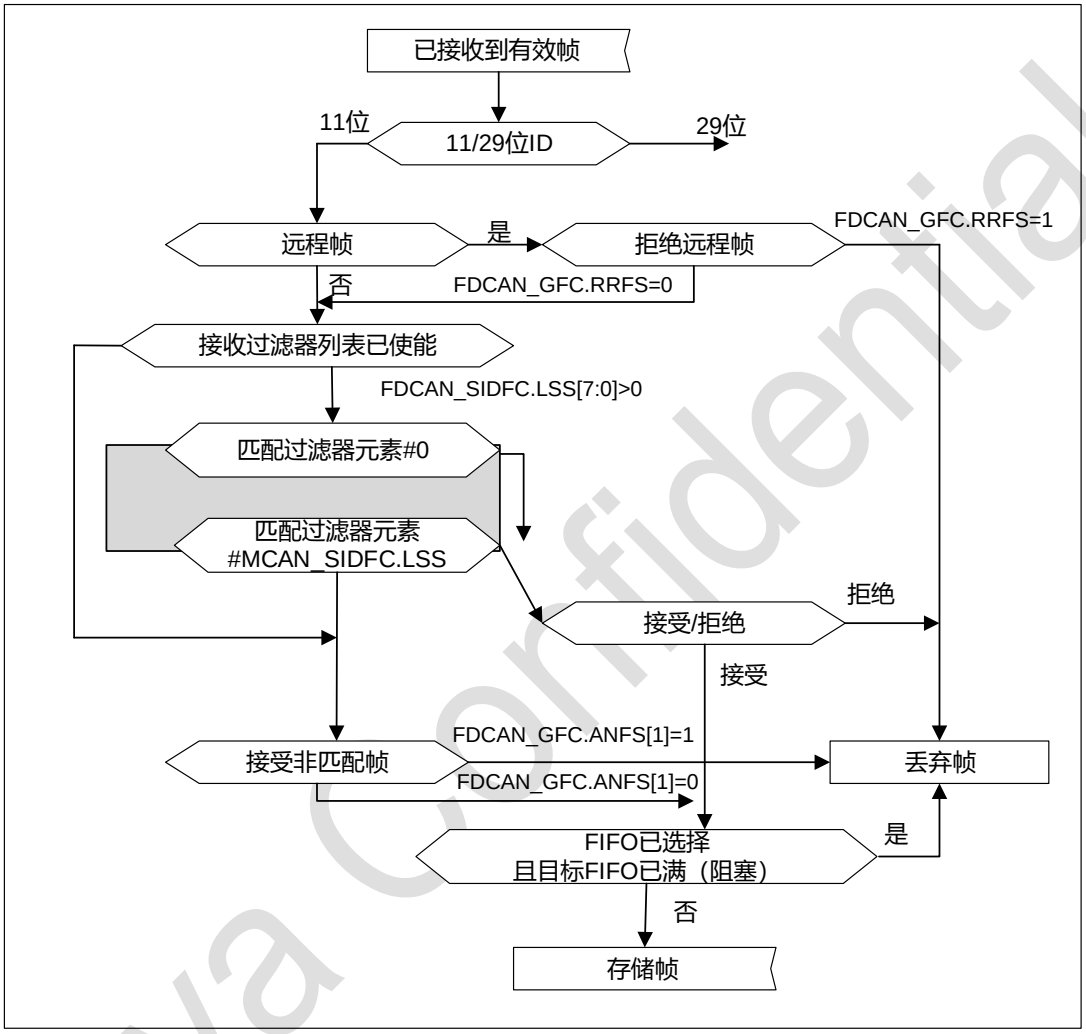


图 31-5 FDCAN 标准消息 ID 过滤器路径

在全局过滤器配置寄存器（FDCAN_GFC）和标准 ID 过滤器配置寄存器（FDCAN_SIDFC）控制下，接收消息的 ID、远程传输请求位（RTR）和 ID 扩展位（IDE）将与配置的过滤器元素列表进行比较。

31.3.5.6 扩展消息 ID 过滤

下图介绍了扩展消息 ID（29 位 ID）的过滤流程。章节【扩展消息 ID 过滤器元素】介绍了扩展消息 ID 过滤器元素。

Puya Confidential

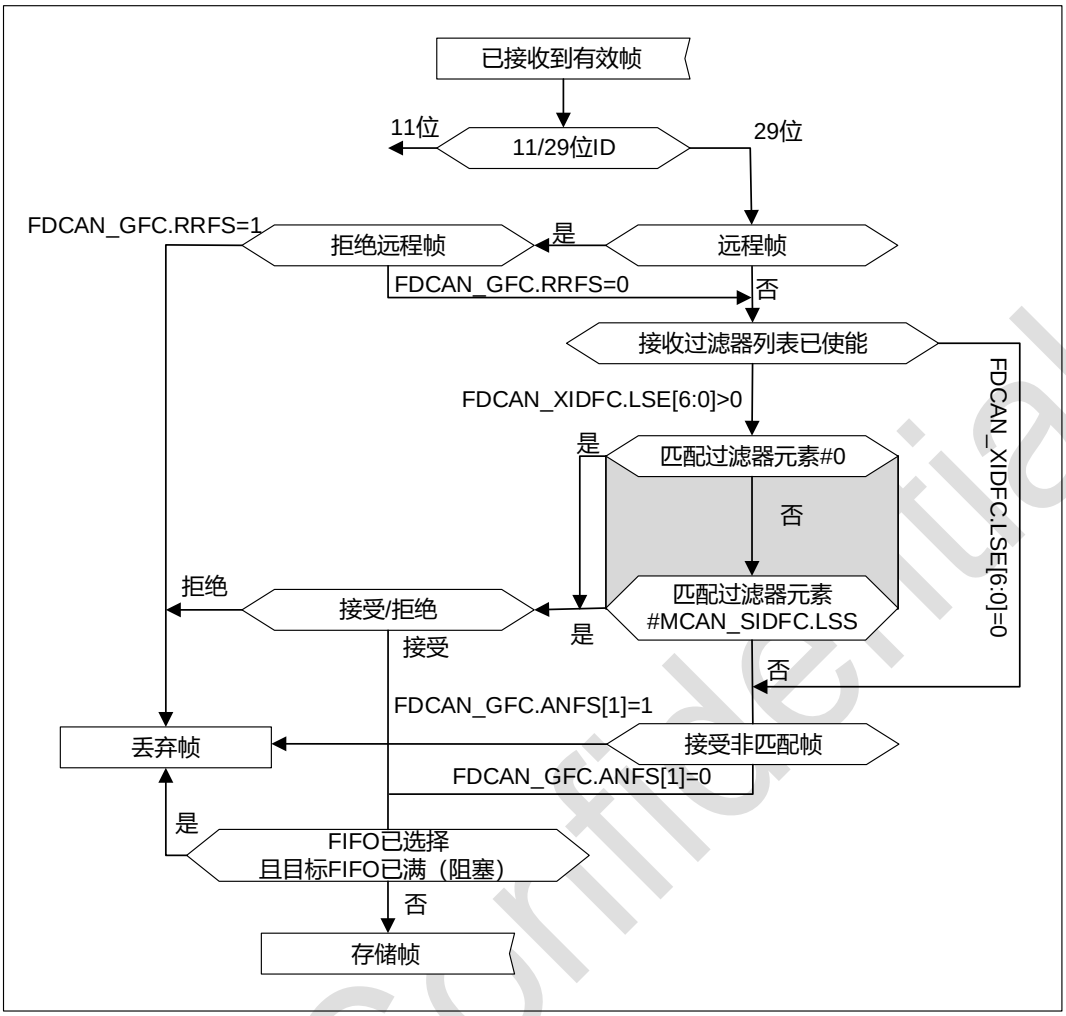


图 31-6 FDCAN 扩展消息 ID 过滤器路径

在全局过滤器配置寄存器（FDCAN_GFC）和扩展 ID 过滤器配置寄存器（FDCAN_XIDFC）控制下，接收消息的 ID、远程传输请求位（RTR）和 ID 扩展位（IDE）将与配置的过滤器元素列表进行比较。扩展 ID 与掩码（FDCAN_XIDAM）会在执行过滤器列表之前与接收到的 ID 进行与运算。

31.3.5.7 **接收 FIFO**

接收 FIFO0 和接收 FIFO1 分别最多可保存 64 个元素（实际元素个数与消息 RAM 容量有关）。两个接收 FIFO 的配置通过寄存器 FDCAN_RXF0C 和 FDCAN_RXF1C 完成。

通过接收过滤的消息会传输到匹配过滤器元素配置的接收 FIFO 中。有关可用于接收 FIFO0 和接收 FIFO1 的过滤器机制的说明，请参见章节【接收过滤】。章节【接收缓冲区和 FIFO 元素】介绍了接收缓冲区和 FIFO 元素。

为了避免接收 FIFO 溢出，可使用接收 FIFO 水位线（Watermark）。当接收 FIFO 填充级别达到 FDCAN_RXFnC.FnWM 配置的接收 FIFO 水位线时，中断标志 FDCAN_IR.RFnW 置位。当接收 FIFO 的放入索引和获取索引相遇时，接收 FIFO 满的条件产生，标志位 FDCAN_RXFnS.FnF 置位，中断标志位 FDCAN_IR.RFnF 置位。

Puya Confidential

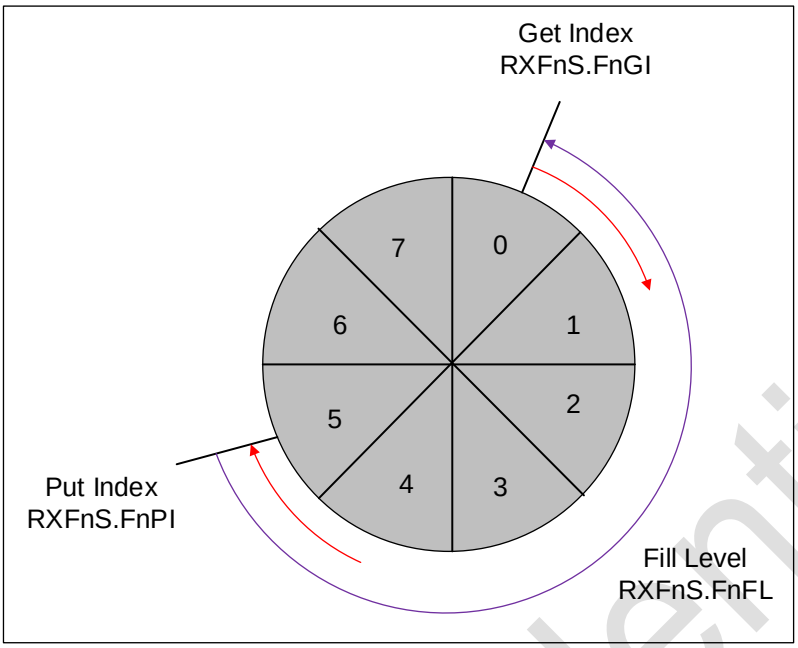


图 31-7 FDCAN 接收 FIFO 状态

当从接收 FIFO 中读取消息时，接收 FIFO 获取索引 FDCAN_RXFnS.FnGI 乘以 FIFO 元素大小后，必须再加上起始地址 FDCAN_RXFnC.FnSA。

表 31-2 FDCAN 接收缓冲器和 FIFO 元素大小

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	数据段（字节）	元素大小（字）
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

31.3.5.8 接收 FIFO 阻塞模式

接收 FIFO 阻塞模式是通过 FDCAN_RXFnC.FnOM=0 配置的，该模式是接收 FIFO 的默认工作模式。

达到接收 FIFO 已满条件时（FDCAN_RXFnS.FnPI=FDCAN_RXFnS.FnGI），标志位 RXFnS.FnF=1，中断标志 FDCAN_IR.RFnF=1。此时，FDCAN 不会继续向相应的接收 FIFO 写入消息，除非从相应的接收 FIFO 读取消息且其获取索引已递增。

如果在相应的接收 FIFO 已满时收到消息，此消息会被丢弃，并会通过 FDCAN_RXFnS.RFnL=1 指示消息丢失。此外，中断标志 FDCAN_IR.RFnL 也会置位。

Puya Confidential

31.3.5.9 接收 FIFO 覆盖模式

接收 FIFO 覆盖模式是通过 FDCAN_RXFnC.FnOM=1 配置的。

达到接收 FIFO 已满条件时（FDCAN_RXFnS.FnPI=FDCAN_RXFnS.FnGI），标志位 RXFnS.FnF=1。此时，FIFO 接收的下一条消息将覆盖最早收到的消息，放入索引和获取索引都会加 1。

如果接收 FIFO 在覆盖模式下工作，并且接收 FIFO 已满，则应至少从获取索引加 1 处开始读取接收 FIFO 元素。这是因为，当 CPU 正在从消息 RAM 读取消息时，恰好有接收到的消息正在被写入消息 RAM，这可能导致 CPU 读到的数据与实际不一致，从接收 FIFO 读取数据时向获取索引中添加偏移量可避免此问题。偏移量取决于 CPU 访问接收 FIFO 的速度。下图显示了读取接收 FIFO 时获取索引的偏移量为 2。在这种情况下，存储在元素 1 和 2 中的两条消息会丢失。

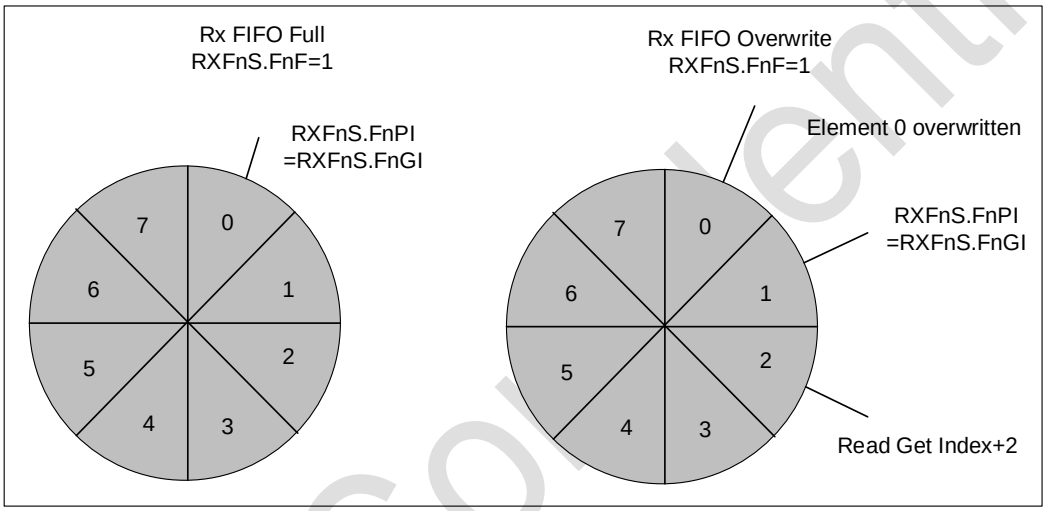


图 31-8 FDCAN 接收 FIFO 溢出处理

从接收 FIFO 读取数据后，必须将读取的最后一个元素的编号写入接收 FIFO 确认索引寄存器 FDCAN_RXFnA.FnA 中，使增加后的获取索引指向下一个元素。如果放入索引未增加到该接收 FIFO 元素编号，接收 FIFO 已满标志会复位（FDCAN_RXFnS.FnF=0）。

31.3.5.10 专用接收缓冲区

FDCAN 支持多达 64 个专用接收缓冲区。专用接收缓冲区的起始地址是通过 RXBC.RBSA 配置的。要使用专用接收缓冲区，对于标准 ID 消息，必须配置标准消息 ID 过滤器元素的 SFEC 位为 0b111、SFID2 位为 0b00（请参见章节【标准消息 ID 过滤器元素】）；对于扩展 ID 消息，必须配置扩展消息 ID 过滤器元素的 EFEC 位为 0b111、 EFID2 位为 0b00（请参见章节【扩展消息 ID 过滤器元素】 ）。 下表为专用接收缓冲区的过滤器配置示例。

接收到的消息被过滤器元素接受后，会存储在过滤器元素引用的消息 RAM 中的接收缓冲区中。存储格式与接收 FIFO 元素的格式相同。此外，中断寄存器中的标志 FDCAN_IR.DRX（专用接收缓冲区中存储的消息）也会置 1。

Puya Confidential

表 31-3 FDCAN 接收缓冲区过滤器配置示例

过滤器元素	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID 消息 1	00	00 0000
1	ID 消息 2	00	00 0001
2	ID 消息 3	00	00 0010

匹配的接收到的消息最后一个字写入消息 RAM 后，寄存器 FDCAN_NDAT1/FDCAN_NDAT2 中对应的新数据标志会置位。只要新数据标志保持置位的状态，相应的接收缓冲区就会被锁定，以免更新为新的匹配消息。新数据标志必须由主机向相应的位写 1 进行复位。

接收缓冲区新数据标志置位时，其对应的消息 ID 过滤器元素将不进行接收过滤，其它的过滤器元素会继续进行接收过滤。其它的消息 ID 过滤器元素可能导致接收到的消息存储到另一接收缓冲区或接收 FIFO 中，或者被拒绝，具体取决于过滤器配置。

31.3.5.11 接收缓冲区处理

- 复位中断标志 FDCAN_IR.DRX
- 读取新数据寄存器
- 从消息 RAM 中读取消息
- 复位已处理消息的新数据标志

31.3.6 FDCAN 发送处理

发送处理单元处理专用发送缓冲区、发送 FIFO 和发送队列的发送请求。它控制着发送消息到 CAN 内核的传输、放入和获取索引以及发送事件 FIFO。最多可为消息发送设置 32 个发送缓冲区，每个发送缓冲区可以独立配置发送模式（经典 CAN 模式或 FD CAN 模式）。发送缓冲区的描述见章节【发送缓冲区元素】。 下表描述了发送帧的可能配置。

表 31-4 FDCAN 发送帧的可能配置

FDCAN_CCCR		发送缓冲区元素		发送帧
BRSE	FDOE	FDF	BRS	
忽略	0	忽略	忽略	经典 CAN
0	1	0	忽略	经典 CAN
0	1	1	忽略	不切换比特率的 FD
1	1	0	忽略	经典 CAN
1	1	1	0	不切换比特率的 FD
1	1	1	1	切换比特率的 FD

注：AUTOSAR 至少需要三个发送队列缓冲区，并支持发送取消。

当发送缓冲区请求挂起寄存器 FDCAN_TXBRP 更新时，发送处理单元会启动发送扫描来检查优先级最高的挂起的发送请求（消息 ID 最小的发送缓冲区）。FDCAN_TXBRP 更新的条件是，一个发送缓冲区发送完成，或主机通过写 FDCAN_TXBAR 添加发送请求，或主机通过写 FDCAN_TXBCR 取消已挂起的发送。

31.3.6.1 发送暂停

发送暂停功能用于 CAN 消息 ID（永久性地）被指定为特定值、并且无法轻易更改的 CAN 系统中。这

Puya Confidential

些消息 ID 的 CAN 仲裁优先级可能高于其他已定义的消息，而在特定的应用中，其相对仲裁优先级应反转。这样可能会导致如下情况：一个 ECU 突发多条较高仲裁优先级的 CAN 消息造成另一个仲裁优先级较低的 ECU CAN 消息被延迟。

举例来说，如果 CAN ECU-1 启用了此功能，并且其应用软件请求发送四条消息，成功发送第一条消息后，将等待两个总线空闲的 CAN 位时间，然后才允许开始发送下一条请求的消息。如果其他 ECU 有挂起的消息，则会在空闲时间开始发送这些消息，不需要与 ECU-1 的下一条消息进行仲裁。接收到消息后，ECU-1 可在接收到的消息释放 CAN 总线后立即开始进行下一次发送。

此功能是通过 FDCAN_CCCR.TXP 控制的。如果此位置位，则每次成功发送消息后，FDCAN 都将暂停两个 CAN 位时间，然后再开始进行下一次发送。这样，即使网络中其他节点的 ID 优先级较低，也能发送消息。默认禁止此功能（FDCAN_CCCR.TXP=0）。此功能可使来自单个节点突发的多条消息变得稀疏一些，避免发送较低优先级消息的 ECU 出现应用程序错误地认为自己请求过多传输。

31.3.6.2 专用发送缓冲区

专用发送缓冲区可完全在主机 CPU 的控制下发送消息。每个专用发送缓冲区都配置了特定的消息 ID。如果多个发送缓冲区配置为相同消息 ID，则会先发送编号最小的发送缓冲区中的消息，这些发送缓冲区应按升序请求发送，即先请求发送缓冲区编号最小的，或者，通过单次写入 FDCAN_TXBAR 同时请求发送。

如果缓冲区中的消息有更新，通过置位 FDCAN_TXBAR.ARn 请求发送。已请求发送的消息，会根据消息 ID 先在内部与发送 FIFO 或发送队列中的消息进行仲裁，然后在外部与 CAN 总线上的消息进行仲裁，最后根据仲裁结果发送出去。

消息 RAM 中分配的专用发送缓冲区元素大小为若干个 32 位字（4 字节）。因此，消息 RAM 中专用发送缓冲区的起始地址，是通过将发送缓冲区索引 (0~31) 乘以元素大小 (字)，再乘以 4 (转换为字节)，然后与 FDCAN_TXBC.TBSA 相加计算得到。

表 31-5 FDCAN 发送缓冲器、FIFO（队列）元素大小

TXESC.TBDS2[2:0]	数据段（字节）	元素大小（字）
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

31.3.6.3 发送 FIFO

发送 FIFO 模式是通过将 FDCAN_TXBC.TFQM 配置为 0 实现的。存储在发送 FIFO 中的消息，是从获取索引 FDCAN_TXFQS.TFGI 指向的消息开始发送的。每次发送后，获取索引会循环递增，直至发送 FIFO 为空。发送 FIFO 会按消息写入的顺序，发送来自不同发送缓冲区但 ID 相同的消息。FDCAN 通过计算获取索引与放入索引之差，得到可用（空闲）的发送 FIFO 元素个数（TXFQS.TFFL）。

Puya Confidential

新消息必须写入以放入索引 FDCAN_TXFQS.TFQPI 指向的发送缓冲区开始的发送 FIFO 中。添加发送请求会使放入索引递增，指向下一空闲的发送 FIFO 元素。当放入索引与获取索引相遇时，会指示发送 FIFO 已满（FDCAN_TXFQS.TFQF=1）。这种情况下，在下一条消息已发送且获取索引递增之前，不应继续向发送 FIFO 写入消息。

如果有单条消息添加到发送 FIFO，通过对发送 FIFO 放入索引对应的 FDCAN_TXBAR 位写 1 来请求发送。

如果有多条（假设有 n 条）消息添加到发送 FIFO，则会写入以放入索引开始的 n 个连续发送缓冲区中。随后通过 FDCAN_TXBAR 请求发送，放入索引循环递增 n。请求发送的发送缓冲区数不应超过发送 FIFO 空闲级别（FDCAN_TXFQS.TFFL）指示的空闲发送缓冲区数。

如果获取索引指向的发送缓冲区的发送请求已取消，那么获取索引会递增到下一个有发送请求挂起的发送缓冲区，并重新计算发送 FIFO 空闲级别（FDCAN_TXFQS.TFFL）。如果取消其他任何发送缓冲区的发送，获取索引和 FIFO 空闲级别保持不变。

消息 RAM 中分配的发送 FIFO 元素大小为若干个 32 位字（4 字节）。因此，下一个可用（空闲）的发送 FIFO 的起始地址，是通过将放入索引 FDCAN_TXFQS.TFQPI（0~31）乘以元素大小（字），再乘以 4，然后与 FDCAN_TXBC.TBSA 相加计算得到。

31.3.6.4 发送队列

发送队列模式是通过将 FDCAN_TXBC.TFQM 配置为 1 实现的。存储在发送队列中的消息，是先从消息 ID 最小（优先级最高）的开始发送的。如果多个发送队列配置为相同消息 ID，如果多个发送队列配置为相同消息 ID，由于发送顺序与存储消息的发送队列的编号有关，而这些发送队列的编号依赖于当前放入索引的状态，因此无法预测发送顺序。

新消息必须写入放入索引 FDCAN_TXFQS.TFQPI 指向的发送缓冲区中。放入索引一直指向发送队列中编号最小的空闲缓冲区。如果发送队列已满（FDCAN_TXFQS.TFQF=1），则放入索引无效，并且在至少有一个请求的消息已发出或挂起的发送请求已取消之前，不应继续向发送队列写入消息。

应用程序可使用寄存器 FDCAN_TXBRP 来代替放入索引，并可将消息放入任何没有挂起发送请求的发送缓冲区中。

消息 RAM 中分配的发送队列缓冲区元素大小为若干个 32 位字（4 字节）。因此，下一个可用（空闲）的发送队列缓冲区的起始地址，是通过将放入索引 FDCAN_TXFQS.TFQPI（0~31）乘以元素大小（字），再乘以 4，然后与 FDCAN_TXBC.TBSA 相加计算得到。

31.3.6.5 混合使用专用发送缓冲区和发送 FIFO

在这种情况下，消息 RAM 中的发送缓冲区会被划分为一组专用发送缓冲区和一个发送 FIFO。专用发送缓冲区的数量是通过 FDCAN_TXBC.NDTB 配置的。分配给发送 FIFO 的发送缓冲区数量是通过 FDCAN_TXBC.TFQS 配置的。如果 FDCAN_TXBC.TFQS 配置为 0，则仅会使用专用发送缓冲区。

Puya Confidential

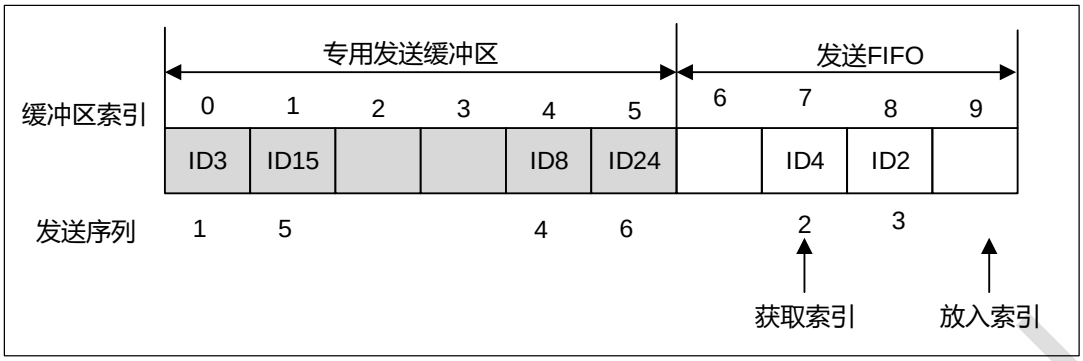


图 31-9 FDCAN 混合配置专用发送缓冲区和发送 FIFO 示例

发送优先级：

- 扫描所有激活了发送请求的专用发送缓冲区和最早挂起发送请求的发送 FIFO 缓冲区（FDCAN_TXEFS.TFGI 指向的缓冲区）
- 消息 ID 最小的缓冲区优先级最高，下次将发送该缓冲区的消息

31.3.6.6 混合使用专用发送缓冲区和发送队列

在这种情况下，消息 RAM 中的发送缓冲区会被划分为一组专用发送缓冲区和一个发送队列。专用发送缓冲区的数量是通过 FDCAN_TXBC.NDTB 配置的。发送队列缓冲区的数量是通过 FDCAN_TXBC.TFQS 配置的。如果 FDCAN_TXBC.TFQS 配置为 0，则仅会使用专用发送缓冲区。

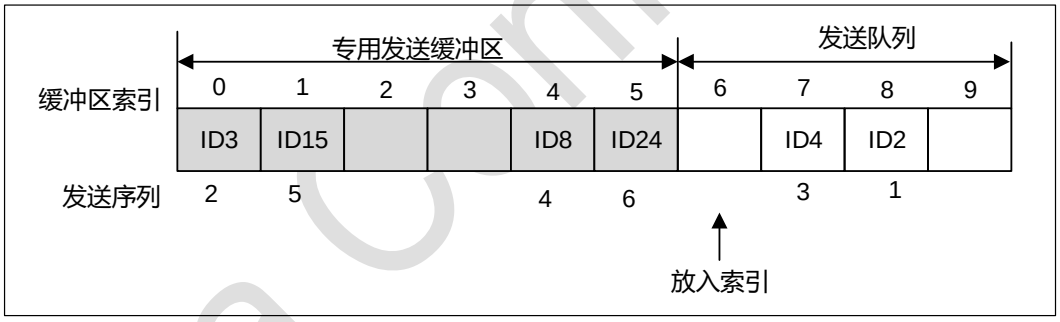


图 31-10 FDCAN 混合配置专用发送缓冲区和发送队列示例

发送优先级：

- 扫描所有激活了发送请求的专用发送缓冲区和发送队列缓冲区
- 消息 ID 最小的发送缓冲区优先级最高，下次将发送该缓冲区的消息

31.3.6.7 发送取消

FDCAN 支持发送取消功能。这个功能特别适用于网关应用和基于 AUTOSAR 的应用。要取消专用发送缓冲区或发送队列缓冲区的发送请求，主机必须向寄存器 FDCAN_TXBCR 中相应的位（发送缓冲区索引对应的位）写 1。发送取消不适用于发送 FIFO 模式。

成功取消后，寄存器 FDCAN_TXBCF 的相应位会置 1。

如果在缓冲区发送过程中请求取消发送，则只要发送正在进行中，FDCAN_TXBRP 相应的位就会保持置 1 状态。如果发送成功，FDCAN_TXBTO 和 FDCAN_TXBCF 相应的位会置 1。如果发送不成功，则不会重复发送，只会将 FDCAN_TXBCF 相应的位置位。

Puya Confidential

注：如果一个挂起的发送在开始发送之前被立即取消，即使该节点中有另一消息挂起，取消后也会有一个短暂的窗口期，在此期间不会开始任何发送。这样，另一节点便可以发送优先级可能比该节点中第二条消息低的消息。

31.3.6.8 发送事件处理

FDCAN 用发送事件 FIFO 来支持发送事件的处理。FDCAN 在 CAN 总线上发送消息后，消息 ID 和时间戳会存储在发送事件 FIFO 元素中。为了将发送事件关联到发送事件 FIFO 元素，已发送的发送缓冲区中的消息标志会被复制到发送事件 FIFO 元素中。

发送事件 FIFO 最多可配置 32 个元素，章节【发送事件 FIFO 元素】描述了发送事件 FIFO 元素。

发送事件 FIFO 的用途是将处理发送状态信息与处理发送消息分开，也就是让发送缓冲区仅保存要发送的消息，而将发送状态单独存储在发送事件 FIFO 中。这样做是有优势的，尤其是在处理动态管理的发送队列时，发送缓冲区可在发送成功后立即用于新消息。重写发送缓冲区之前，不需要保存该发送缓冲区的发送状态信息。

当 FDCAN_IR.TEFF 指示发送事件 FIFO 已满时，在至少已读取一个元素且发送事件 FIFO 获取索引递增之前，不会继续向发送事件 FIFO 写入元素。如果在发送事件 FIFO 已满时发生发送事件，此事件会被丢弃，中断标志 FDCAN_IR.TEFL 置位。

为了避免发送事件 FIFO 溢出，可使用发送事件 FIFO 水位线。当发送事件 FIFO 填充数量达到由 TXEFC.EFWM 配置的发送事件 FIFO 水位线时，中断标志 FDCAN_IR.TEFW 置位。

从发送事件 FIFO 读取数据时，必须将发送事件 FIFO 获取索引 TXEFS.EFGI 乘以 2（发送事件 FIFO 元素大小为两个字），再乘以 4，然后与发送事件 FIFO 起始地址 TXEFC.EFSA 相加，以得到目标数据所在的地址。

31.3.7 FDCAN FIFO 确认处理

接收 FIF0、接收 FIF1 和发送事件 FIFO 的获取索引，是通过对相应 FIFO 确认索引寄存器进行写操作来控制的，请参见章节【FDCAN 接收 FIFO0 确认寄存器（FDCAN_RXF0A）】、章节【FDCAN 接收 FIFO1 确认寄存器(FDCAN_RXF1A)】和章节【FDCAN 发送事件 FIFO 确认寄存器(FDCAN_TXEFA)】。对 FIFO 确认索引进行写操作会将 FIFO 获取索引设置为 FIFO 确认索引加 1，进而会更新 FIFO 填充级别。有两种用例：

1.如果只从 FIFO 读取了一个元素（获取索引指向的元素），则应将获取索引值写入 FIFO 确认索引寄存器中。

2.如果已从 FIFO 中读取了一个序列的元素，则仅需在该序列读取结束时，将最后一个元素的索引值写入 FIFO 确认索引寄存器，即可更新 FIFO 获取索引。

由于 CPU 可自由访问 FDCAN 的消息 RAM，在以任意顺序（不考虑获取索引）读取 FIFO 元素时需要特别注意，特别是在从两个接收 FIFO 之一读取高优先级消息时。在这种情况下，不应对 FIFO 确认索引寄存器进行写操作，否则会导致获取索引指向错误的位置，还会改变 FIFO 填充级别，一些较早的 FIFO 元素可能会丢失。

注: 应用程序必须确保写入到 FIFO 确认索引寄存器中的值是正确的，FDCAN 不会检查数值是否正确。

Puya Confidential

31.3.8 FDCAN 消息 RAM

31.3.8.1 消息 RAM 配置

消息 RAM 的宽度为 32 位。 FDCAN 模块最多可在消息 RAM 中分配 212 个字，如下图所示。实际应用中不要求对所有部分都进行配置，各部分之间的顺序也没有限制。

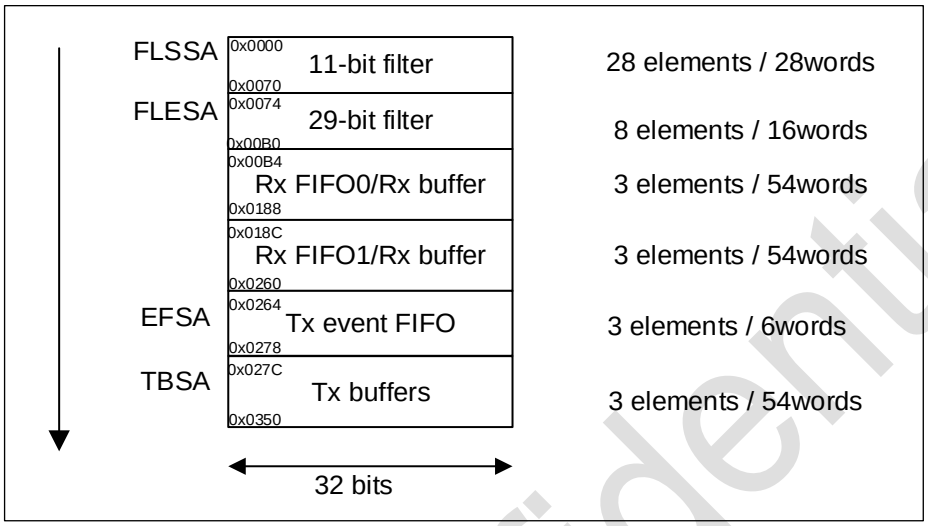


图 31-11 FDCAN 消息 RAM 配置

FDCAN 访问消息 RAM 时需要按 32 位对齐的地址进行寻址，而不是字节对齐的地址。可配置的起始地址是 32 位字地址，即仅评估地址的 15 到 2 位，忽略两个最低有效位。

注: FDCAN 不会检查消息 RAM 的配置是否存在错误，因此配置不同部分的起始地址以及各部分元素的数量时必须多加留意，以免造成数据入侵或丢失。

31.3.8.2 接收缓冲区和 FIFO 元素

最多可在消息 RAM 中配置 2 个接收 FIFO/接收缓冲区。每个接收 FIFO/接收缓冲区均可配置为最多存储 3 个消息。可通过寄存器 FDCAN_RXESC 配置元素大小，FD CAN 消息的数据段最多可达 64 字节。

表 31-6 FDCAN 接收缓冲区和 FIFO 元素

位	31	24	23	16	15	8	7	0
R0	ESI	XTD RTR	ID[28:0]					
R1	ANMF	FIDX[6:0]	Res FDF BRS	DLC[3:0]	RXTS[15:0]			
R2	DB3[7:0]		DB2[7:0]		DB1[7:0]		DB0[7:0]	
R3	DB7[7:0]		DB6[7:0]		DB5[7:0]		DB4[7:0]	
...	...		...		...		...	
Rn	DBm[7:0]		DBm-1[7:0]		DBm-2[7:0]		DBm-3[7:0]	

表 31-7 FDCAN 接收缓冲区和 FIFO 元素说明

位域	标记	说明
R0[31]	ESI	错误状态指示符（Error State Indicator） 0：发送节点为错误主动状态 1：发送节点为错误被动状态
R0[30]	XTD	扩展 ID 标志位（Extended Identifier） 向主机指示接收到的帧的 ID 是标准 ID 还是扩展 ID。

Puya Confidential

		0： 11 位标准 ID 1： 29 位扩展 ID
R0[29]	RTR	远程发送请求（Remote Transmission Request） 向主机指示接收到的是于数据帧还是远程帧。 0：接收到的帧是数据帧 1：接收到的帧是远程帧 注： FD CAN 没有远程帧。在 FD CAN 帧中（FDF=1），显性位 RRS（Remote Request Substitution）替代了 RTR 位。
R0[28:0]	ID[28:0]	ID（Identifier） 标准 ID 或扩展 ID 取决于 XTD 位，标准 ID 存储在 ID[28:18]中。
R1[31]	ANMF	接受非匹配帧（Accepted Non-matching Frame） 可通过 FDCAN_GFC.ANFS 和 FDCAN_GFC.ANFE 使能接受非匹配帧。 0：接收到的帧与过滤器索引 FIDX 指向的过滤器元素匹配 1：接收到的帧与任何接收过滤器元素都不匹配
R1[30:24]	FIDX[6:0]	过滤器索引（Filter Index） 0~127：匹配的接收过滤器元素的索引（ANMF =1 时无效）范围为 0 到 FDCAN_SIDFC.LSS]减 1 或 FDCAN_XIDFC.LSE 减 1。
R1[21]	FDF	FD 格式（FD Format） 0：经典 CAN 帧格式 1： FD CAN 帧格式（新的 DLC 编码和 CRC）
R1[20]	BRS	比特率切换（Bit Rate Switch） 0：接收帧没有比特率切换 1：接收帧有比特率切换
R1[19:16]	DLC[3:0]	数据长度代码 （Data Length Code） 0~8：经典 CAN 和 FD CAN：接收到的帧包含 0~8 个数据字节 9~15：CAN：接收到的帧包含 8 个数据字节 9~15：FD CAN：接收到的帧包含 12/16/20/24/32/48/64 个数据字节
R1[15:0]	RXTS[15:0]	接收时间戳（Rx Timestamp） 接收到 SOF（Start Of Frame）时捕获的时间戳计数器值。 分辨率取决于时间戳计数器预分频器 FDCAN_TSCC.TCP 的配置。
R2[31:24]	DB3[7:0]	数据字节 3（Data Byte 3）
R2[23:16]	DB2[7:0]	数据字节 2（Data Byte 2）
R2[15:8]	DB1[7:0]	数据字节 3（Data Byte 1）
R2[7:0]	DB0[7:0]	数据字节 2（Data Byte 0）
R3[31:24]	DB7[7:0]	数据字节 7（Data Byte 7）
R3[23:16]	DB6[7:0]	数据字节 6（Data Byte 6）
R3[15:8]	DB5[7:0]	数据字节 5（Data Byte 5）
R3[7:0]	DB4[7:0]	数据字节 4（Data Byte 4）
...	...	...
Rn[31:24]	DBm[7:0]	数据字节 m（Data Byte m）
Rn[23:16]	DBm-1[7:0]	数据字节 m-1（Data Byte m-1）
Rn[15:8]	DBm-2[7:0]	数据字节 m-2（Data Byte m-2）
Rn[7:0]	DBm-3[7:0]	数据字节 m-3（Data Byte m-3）

Puya Confidential

31.3.8.3 **发送缓冲区元素**

发送缓冲区可配置为专用发送缓冲区和发送 FIFO（或队列）。如果发送缓冲区由专用发送缓冲区和发送 FIFO（或队列）共享，则专用发送缓冲区在发送缓冲区的起始处，其后是发送 FIFO（或队列）。发送处理单元会评估发送缓冲区配置 FDCAN_TXBC.TFQS 和 FDCAN_TXBC.NDTB，以区分专用发送缓冲区与发送 FIFO（队列）。发送缓冲区元素的结构及相关说明如下表所示。可通过寄存器 FDCAN_TXESC 配置元素大小，FD CAN 消息的数据段最多可达 64 字节。

表 31-8 FDCAN 发送缓冲区元素

位	**31**			**24**	**23**					**16**	**15**	**8**	**7**	**0**
T0	ESI	XTD	RTR	ID[28:0]										
T1	MM[7:0]				EFC	Res	FDF	BRS	DLC[3:0]		MM[15:8]		Res	
T2	DB3[7:0]				DB2[7:0]						DB1[7:0]		DB0[7:0]	
T3	DB7[7:0]				DB6[7:0]						DB5[7:0]		DB4[7:0]	
...	...				...						...		...	
Tn	DBm[7:0]				DBm-1[7:0]						DBm-2[7:0]		DBm-3[7:0]	

表 31-9 FDCAN 发送缓冲区元素说明

位域	**标记**	**说明**
T0[31]	ESI	错误状态指示符（Error State Indicator） 0：FD CAN 格式的 ESI 位仅取决于错误被动标志 1：FD CAN 格式的 ESI 位以隐性位发送 注： 发送缓冲区的 ESI 位会与错误被动标志进行或运算，以确定将发送的 FD 帧 ESI 位的值。 按照 FD CAN 协议规范的要求，错误主动节点可选择发送隐性 ESI 位，但错误被动节点将始终发送隐性 ESI 位。
T0[30]	XTD	扩展 ID 标志位（Extended Identifier） 0：11 位标准 ID 1：29 位扩展 ID
T0[29]	RTR	远程发送请求（Remote Transmission Request） 0：发送数据帧 1：发送远程帧 注： 当 RTR = 1 时，即使 FDCAN_CCCR.FDOE 使能以 FD CAN 格式进行发送，FDCAN 也会按照 ISO 11898-1:2015 的规定发送远程帧。
T0[28:0]	ID[28:0]	ID（Identifier） 标准 ID 或扩展 ID，取决于 XTD 位，标准 ID 必须写在 ID[28:18]中。
T1[31:24]	MM[7:0]	消息标记（Message Marker） 在发送缓冲区配置期间由 CPU 写入。复制到发送事件 FIFO 元素中，用于标识发送消息状态。
T1[23]	EFC	事件 FIFO 控制（Event FIFO Control） 0：不存储发送事件 1：存储发送事件

Puya Confidential

T1[21]	FDF	FD 格式（FD Format） 0：以经典 CAN 格式发送帧 1：以 FD CAN 格式发送帧
T1[20]	BRS	比特率切换（Bit Rate Switch） 0：FD CAN 帧发送时不切换比特率 1：FD CAN 帧发送时切换比特率 注： 仅当 FD CAN 使能（FDCAN_CCCR.FDOE=1）时，才会评估 ESI、 FDF 和 BRS 位。仅当额外设置 FDCAN_CCCR.BRSE =1 时，才会评估 BRS 位。
T1[19:16]	DLC[3:0]	数据长度代码（Data Length Code） 0~8：经典 CAN 和 FD CAN：发送帧包含 0~8 个数据字节 9~15： CAN：发送帧包含 8 个数据字节 9~15： FD CAN：发送帧包含 12/16/20/24/32/48/64 个数据字节
T1[15:8]	MM[15:8]	宽消息标记的高字节（High byte of Wide Message Marker） 在发送缓冲区配置期间由 CPU 写入。复制到发送事件 FIFO 元素中，用于标识发送消息状态。仅当 FDCAN_CCCR.WMM=1 时有效。
T2[31:24]	DB3[7:0]	数据字节 3（Data Byte 3）
T2[23:16]	DB2[7:0]	数据字节 2（Data Byte 2）
T2[15:8]	DB1[7:0]	数据字节 3（Data Byte 1）
T2[7:0]	DB0[7:0]	数据字节 2（Data Byte 0）
T3[31:24]	DB7[7:0]	数据字节 7（Data Byte 7）
T3[23:16]	DB6[7:0]	数据字节 6（Data Byte 6）
T3[15:8]	DB5[7:0]	数据字节 5（Data Byte 5）
T3[7:0]	DB4[7:0]	数据字节 4（Data Byte 4）
...	...	...
Tn[31:24]	DBm[7:0]	数据字节 m（Data Byte m）
Tn[23:16]	DBm-1[7:0]	数据字节 m-1（Data Byte m-1）
Tn[15:8]	DBm-2[7:0]	数据字节 m-2（Data Byte m-2）
Tn[7:0]	DBm-3[7:0]	数据字节 m-3（Data Byte m-3）

31.3.8.4 发送事件 FIFO 元素

每个元素存储已发送消息相关的信息。通过读取发送事件 FIFO，主机 CPU 按消息的发送顺序获取这些信息。关于发送事件 FIFO 的状态信息，可从寄存器 FDCAN_TXEFS 中获取。发送事件 FIFO 元素的结构及相关说明如下表所示。

E1A: 当 FDCAN_CCCR.WMM=0 时，E1A.TXTS[15:0]保存由 FDCAN 的内部时间戳逻辑生成的 16 位时间戳。

E1B：当使能 16 位宽消息标志（FDCAN_CCCR.WMM=1）时， E1B.MM[15:8]为宽消息标志的高 8 位。

Puya Confidential

表 31-10 FDCAN 发送事件 FIFO 元素及说明

位	31			24	23			16	15	8	7	0
E0	ESI	XTD	RTR	ID[28:0]								
E1A	MM[7:0]				EF[1:0]	FDF	BRS	DLC[3:0]	TXTS[15:0]			
E1B	MM[7:0]				EF[1:0]	FDF	BRS	DLC[3:0]	MM[15:8]		Res	

位域	标记	说明
E0[31]	ESI	错误状态指示符（Error State Indicator） 0：发送节点为错误主动状态 1：发送节点为错误被动状态
E0[30]	XTD	扩展 ID 标志位（Extended Identifier） 0：11 位标准 ID 1：29 位扩展 ID
E0[29]	RTR	远程发送请求（Remote Transmission Request） 0：发送帧为数据帧 1：发送帧为远程帧
E0[28:0]	ID[28:0]	标识符（Identifier） 标准 ID 或扩展 ID，取决于 XTD 位。标准 ID 存储在 ID[28:18]中。
E1A/B[31:24]	MM[7:0]	消息标记（Message Marker） 从发送缓冲区复制到发送事件元素，用于标识发送消息状态。
E1A/B[23:22]	EF[1:0]	事件类型（Event Type） 00：保留 01：发送事件 10：取消后仍发送（在 DAR 模式下发送时始终为该值） 11：保留
E1A/B[21]	FDF	FD 格式（FD Format） 0：经典 CAN 帧格式 1：FD CAN 帧格式（新的 DLC 编码和 CRC）
E1A/B[20]	BRS	比特率切换（Bit Rate Switch） 0： FD CAN 帧发送时没有切换比特率 1： FD CAN 帧发送时有切换比特率
E1A/B[19:16]	DLC[3:0]	数据长度代码（Data Length Code） 0~8：经典 CAN 和 FD CAN：发送帧包含 0~8 个数据字节 9~15：经典 CAN：发送帧包含 8 个数据字节 9~15： FD CAN：发送帧包含 12/16/20/24/32/48/64 个数据字节
E1A[15:0]	TXTS[15:0]	发送时间戳（Tx Timestamp） 发送 SOF（Start Of Frame）时捕获的时间戳计数器值。分辨率取决于时间戳计数器预分频器 FDCAN_TSCC.TCP 的配置。
E1B[15:8]	MM[15:8]	宽消息标记的高字节（High byte of Wide Message Marker） 在发送缓冲区配置期间由 CPU 写入。复制到发送事件 FIFO 元素中，用于标识发送消息状态。

31.3.8.5 标准消息 ID 过滤器元素

最多可为 11 位标准 ID 配置 128 个过滤器元素（每个元素大小为 1 个字）。其地址为过滤器元素索引

Puya Confidential

（0~127）乘以 4（转换成字节），加上起始地址 FDCAN_SIDFC.FLSSA。标准消息 ID 元素的结构及相关说明请见下表。

表 31-11 FDCAN 标准消息 ID 过滤器元素及说明

位	31	24	23	16	15	8	7	0
S0	SFT[1:0]	SFEC[2:0]	SFID1[10:0]		Res	SFID2[10:0]		

位域	标记	说明
S0[31:30]	SFT[1:0]	标准过滤器类型（Standard Filter Type） 00：从 SFID1 到 SFID2 的范围过滤器（SFID2 ≥ SFID1） 01：SFID1 或 SFID2 定义的双 ID 过滤器 10：经典过滤器： SFID1=ID， SFID2=掩码 11：禁止该过滤器元素 注：当 SFT=0b11 时，该过滤器元素被禁止，接收过滤继续进行（与 SFEC=0b000 时的行为相同） 。
S0[29:27]	SFEC[2:0]	标准过滤器元素配置（Standard Filter Element Configuration） 所有使能的过滤器元素都用于标准帧的接收过滤。当匹配到一个已使能的过滤器元素，或到达过滤器列表结尾处时，停止接收过滤。如果 SFEC=0b100、 0b101 或 0b110， 出现匹配时中断标志 FDCAN_IR.HPM 置位，并产生中断（如果使能）。在这种情况下，寄存器 FDCAN_HPMS 会更新为优先级匹配的状态。 000：禁止该过滤器元素 001：如果匹配，则存储到接收 FIFO0 中 010：如果匹配，则存储到接收 FIFO1 中 011：如果匹配，则拒绝 ID 100：如果匹配，则设置优先级，不存储 101：如果匹配，则设置优先级并存储到 FIFO0 中 110：如果匹配，则设置优先级并存储到 FIFO1 中 111：存储到接收缓冲区中或作为调试消息，忽略 SFT[1:0]的配置
S0[26:16]	SFID[10:0]	标准过滤器 ID1（Standard Filter ID 1） 标准 ID 过滤器元素的第一个 ID。当为接收缓冲区或调试消息过滤时，此位域定义要存储的标准消息的 ID。接收的 ID 必须完全匹配，不使用掩码机制。
S0[10:0]	SFID2[10:0]	标准过滤器 ID2（Standard Filter ID 2） 该位域具有不同含义，具体视 SFEC 的配置而定： – SFEC=0b001~0b110 时：标准 ID 过滤器元素的第二个 ID – SFEC=0b111 时： 用于接收缓冲区或调试消息的过滤器
	SFID2[10:9]	确定接收的消息存储到接收缓冲区中，还是作为调试消息序列中的消息 A、 B 或 C 进行处理。 00：将消息存储到接收缓冲区中 01：调试消息 A 10：调试消息 B 11：调试消息 C
	SFID2[5:0]	定义存储匹配消息的接收缓冲区到起始地址 RXBC.RBSA 的偏移（缓冲区索引）。

Puya Confidential

31.3.8.6 扩展消息 ID 过滤器元素

最多可为 29 位扩展 ID 配置 64 个过滤器元素（每个元素大小为 2 个字）。访问扩展消息 ID 过滤器元素时，其地址为过滤器元素索引（0~63）乘以 2，再乘以 4（转换成字节），然后加上起始地址 FDCAN_XIDFC.FLESA。扩展消息 ID 元素的结构及相关说明请见下表。

表 31-12 FDCAN 扩展消息 ID 过滤器元素

位	31 24		23 16	15 8	7 0
F0	EFEC[2:0]		EFID1[28:0]		
F1	EFT[1:0]	Res	EFID2[28:0]		

位域	标记	说明
F0[31:29]	EFEC[2:0]	扩展过滤器元素配置 （Extended Filter Element Configuration） 所有使能的过滤器元素都用于扩展帧的接收过滤。当匹配到一个已使能的过滤器元素，或到达过滤器列表结尾处时，停止接收过滤。如果 EFEC=0b100、 0b101 或 0b110，出现匹配时中断标志 FDCAN_IR.HPM 置位，并产生中断（若使能）。在这种情况下，寄存器 FDCAN_HPMS 会更新为优先级匹配的状态。 000：禁止该过滤器元素 001：如果匹配，则存储到接收 FIFO0 中 010：如果匹配，则存储到接收 FIFO1 中 011：如果匹配，则拒绝 ID 100：如果匹配，则设置优先级 101：如果匹配，则设置优先级并存储到 FIFO0 中 110：如果匹配，则设置优先级并存储到 FIFO1 中 111：存储到接收缓冲区中或作为调试消息，忽略 EFT[1:0]的配置
F0[28:0]	EFID[28:0]	扩展过滤器 ID1 扩展 ID 过滤器元素的第一个 ID。当为接收缓冲区或调试消息过滤时，此位域定义要存储的扩展消息的 ID。接收的 ID 必须完全匹配，仅使用 FDCAN_XIDAM 掩码机制。
F1[31:30]	EFT[1:0]	扩展过滤器类型 00：从 EFID1 到 EFID2 的范围过滤器(EFID2 ≥ EFID1) 01： EFID1 或 EFID2 定义的双 ID 过滤器 10：经典过滤器： EFID1=ID， EFID2=掩码 11： 从 EFID1 到 EFID2 的范围过滤器（EFID2 ≥ EFID1）， 不使用 FDCAN_XIDAM 掩码
F1[28:0]	EFID2[28:0]	扩展过滤器 ID2（Extended Filter ID 2） 该位域具有不同含义，具体视 EFEC 的配置而定： – SFEC =0b001~0b110 时：扩展 ID 过滤器元素的第二个 ID – SFEC =0b111 时： 用于接收缓冲区或调试消息的过滤器
	SFID2[10:9]	确定接收的消息存储到接收缓冲区中，还是作为调试消息序列中的消息 A、 B 或 C 进行处理。 00：将消息存储到接收缓冲区中 01：调试消息 A 10：调试消息 B 11：调试消息 C
	SFID2[5:0]	定义存储匹配消息的接收缓冲区到起始地址 RXBC.RBSA 的偏移（缓冲区索引）。

Puya Confidential

31.4 FDCAN 软件初始化

软件初始化是通过置位 FDCAN_CCCR.INIT 开始的。软件复位、硬件复位或进入 Bus_Off 会置位 FDCAN_CCCR.INIT。当 FDCAN_CCCR.INIT 被置位时，CAN 总线上传入和传出的消息都将停止，FDCAN_TX 引脚的状态变为隐性（高电平）。错误逻辑管理单元（EML）的计数器保持不变。将 FDCAN_CCCR.INIT 置位不会改变任何配置寄存器。将 FDCAN_CCCR.INIT 清零可以完成软件初始化，随后，位流处理单元（BSP）等待总线上出现 11 个连续隐性位的序列（Bus_Idle），以此将其自身与 CAN 总线上的数据传输进行同步，然后才能参与总线活动并开始消息传输。

仅当 FDCAN_CCCR.INIT 和 FDCAN_CCCR.CCE 均置位时，才能访问 FDCAN 配置寄存器。FDCAN 配置寄存器如下表所示。

表 31-13 FDCAN 配置寄存器列表

寄存器	说明
FDCAN_DBTP	FDCAN 数据位时间和预分频寄存器
FDCAN_TEST	FDCAN 测试寄存器
FDCAN_RWD	FDCAN RAM 看门狗寄存器
FDCAN_CCCR	FDCAN CC 控制寄存器
FDCAN_NBTP	FDCAN 标称位时间和预分频寄存器
FDCAN_TSCC	FDCAN 时间戳计数器配置寄存器
FDCAN_TOCC	FDCAN 超时计数器配置寄存器
FDCAN_TDCR	FDCAN 发送器延迟补偿寄存器
FDCAN_GFC	FDCAN 全局过滤器配置寄存器
FDCAN_SIDFC	FDCAN 标准 ID 过滤器配置寄存器
FDCAN_XIDFC	FDCAN 扩展 ID 过滤器配置寄存器
FDCAN_XIDAM	FDCAN 扩展 ID 与掩码寄存器
FDCAN_RXF0C	FDCAN 接收 FIFO0 配置寄存器
FDCAN_RXBC	FDCAN 接收缓冲区配置寄存器
FDCAN_RXF1C	FDCAN 接收 FIFO1 配置寄存器
FDCAN_RXESC	FDCAN 接收缓冲区和 FIFO 元素大小配置寄存器
FDCAN_TXBC	FDCAN 发送缓冲区配置寄存器
FDCAN_TXESC	FDCAN 发送缓冲区元素大小配置寄存器
FDCAN_TXEFC	FDCAN 发送事件 FIFO 配置寄存器

FDCAN_CCCR.CCE 仅在 FDCAN_CCCR.INIT 置位时才能被置位或清零，在 FDCAN_CCCR.INIT 清零时自动清零。

FDCAN_CCCR.CCE 置位后，以下寄存器会被复位：

FDCAN_HPMS——高优先级消息状态

FDCAN_RXF0S——接收 FIFO 0 状态

FDCAN_RXF1S——接收 FIFO 1 状态

FDCAN_TXFQS——发送 FIFO/队列状态

FDCAN_TXBRP——发送缓冲区请求挂起

FDCAN_TXBTO——发送缓冲区发送发生

Puya Confidential

FDCAN_TXBCF——发送缓冲区取消完成

FDCAN_TXEFS——发送事件 FIFO 状态

FDCAN_TXBAR——发送缓冲区添加请求

FDCAN_TXBCR——发送缓冲区取消请求

当 CCCR.CCE 置位时，超时计数器值 TOCV.TOC 预设为 TOCC.TOP 配置的值。

以下寄存器仅在 CCCR.CCE = '0' 时可写：

TXBAR - Tx 缓冲区添加请求

TXBCR - Tx 缓冲区取消请求

仅当 FDCAN_CCCR.INIT 和 FDCAN_CCCR.CCE 均已置位时，主机才能将 FDCAN_CCCR.TEST 和 FDCAN_CCCR.MON 置位，主机可随时复位 FDCAN_CCCR.TEST 和 FDCAN_CCCR.MON。仅当 FDCAN_CCCR.INIT 和 FDCAN_CCCR.CCE 均已置位时，才能将 FDCAN_CCCR.DAR 置位或清零。

31.5 FDCAN 寄存器

31.5.1 FDCAN 字节序寄存器（FDCAN_ENDN）

地址偏移：0x004

复位值：0x87654321

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETV[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETV[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:0	ETV	R	0x87654321	字节序测试值

31.5.2 FDCAN 数据位时间和预分频寄存器（FDCAN_DBTP）

地址偏移：0x00C

复位值：0x0000 0A33

仅当 FDCAN_CCCR.CCE 和 FDCAN_CCCR.INIT 位均被置位时，才能对该寄存器进行写访问。CAN 位时间的可编程范围为 4 到 49 个时间片（Time quanta，tq）。CAN 时间片的可编程范围为 1 到 32 个 FDCAN 通信时钟周期（Minimum time quantum，mtq）。tq = (DBRP + 1)mtq。

DTSEG1 为 Prop_Seg 与 Phase_Seg1 之和。DTSEG2 为 Phase_Seg2。

因此，位时间的长度为（编程值）[DTSEG1 + DTSEG2 + 3] tq 或（功能值） [Sync_Seg + Prop_Seg +Phase_Seg1 + Phase_Seg2] tq。

信息处理时间（IPT）为零，这意味着下一位数据在采样点之后的第一个时钟边沿可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDC	Res.	Res.	DBRP[4:0]				
								RW			RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DTSEG1[4:0]					DTSEG2[3:0]				DSJW[3:0]			
			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	-	保留
23	TDC	RW	0	发送器延迟补偿 0：禁止发送器延迟补偿 1：使能发送器延迟补偿
22:21	Reserved	-	-	保留
20:16	DBRP[4:0]	RW	0	数据比特率预分频 生成位时间片的 FDCAN 通信时钟预分频值，位时间为该时间片的倍数。预分频有效值为 0 到 31，当 TDC=1 时，范围被限制在 0 到 1，硬件实际使用值为编程值加 1。
15:13	Reserved	-	-	保留
12:8	DTSEG1[4:0]	RW	5'b01010	采样点之前的数据时间 有效值为 0 到 31，硬件实际使用值为编程值加 1。
7:4	DTSEG2[3:0]	RW	4'b0011	采样点之后的数据时间 有效值为 1 到 15，硬件实际使用值为编程值加 1。
3:0	DSJW[3:0]	RW	4'b0011	数据（重新）同步跳转宽度 有效值为 0 到 15，硬件实际使用值为编程值加 1。

注：当 FDCAN 通信时钟为 8MHz 时，复位值 0x0000 0A33 会将 FDCAN 数据段的比特率配置为 500kbps。通过 DBTP 配置的 FD CAN 数据段比特率不能小于通过 NBTP 配置的仲裁段比特率。

31.5.3 FDCAN 测试寄存器（FDCAN_TEST)

地址偏移：0x010

复位值：0x0000 00X0（位 7 的复位值由 FDCAN_RX 引脚的实际电平决定）

仅当 FDCAN_CCCR.TEST 位被置 1 后，才能对该测试寄存器进行写访问。当 FDCAN_CCCR.TEST 位被清 0 时，所有测试寄存器功能都会设为其复位值。

回环模式和软件控制 FDCAN_TX 引脚属于硬件测试模式。TX 配置为非 0b00 时可能会干扰 CAN 总线上的消息传输。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SVAL	TXBNS[4:0]				
										R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	PVAL	TXBNP[4:0]					RX	TX[1:0]		LBCK	Res.	Res.	Res.	Res.
		R	R	R	R	R	R	R	RW	RW	RW				

Bit	Name	RW	Reset Value	Function
31:22	Reserved	-	-	保留
21	SVAL	R	0	已开始有效 0：TXBNS 的值无效 1：TXBNS 的值有效
20:16	TXBNS[4:0]	R	5’b0	已开始发送的缓冲区编号 最后开始发送的发送缓冲区编号。 当 SVAL 位被置 1 时有效，有效范围为 0~31。
15:14	Reserved	-	-	保留
13	PVAL	R	0	已准备有效

Puya Confidential

				0：TXBNP 的值无效 1：TXBNP 的值有效
12:8	TXBNP[4:0]	R	5’b0	已准备好发送的缓冲区编号 准备好发送的发送缓冲区编号。 当 PVAL 位被置 1 时有效，有效范围为 0~31。
7	RX	R	0	接收引脚 监测到的 FDCAN_RX 引脚的实际值 0：CAN 总线为显性（FDCAN_RX=0） 1：CAN 总线为隐性（FDCAN_RX=1）
6:5	TX[1:0]	RW	2’b0	发送引脚控制 00：复位值， FDCAN_TX 引脚由 CAN 内核控制，在 CAN 位时间结束时更新 01：可以在 FDCAN_TX 引脚监测采样点 10：FDCAN_TX 引脚输出显性电平（0） 11：FDCAN_TX 引脚输出隐性电平（1）
4	LBCK	RW	0	回环模式 0：复位值，禁止回环模式 1：使能回环模式（详见【外部回环模式】 和【内部回环模式】章节）
3:0	Reserved	-	-	保留

31.5.4 FDCAN RAM 看门狗寄存器（FDCAN_RWD）

地址偏移：0x014

复位值：0x0000 0000

RAM 看门狗会监视消息 RAM 的 READY 信号输出。对消息 RAM 进行访问时，会启动消息 RAM 看门狗计数器，该计数器的起始值由 FDCAN_RWD.WDC 位进行配置。

当消息 RAM 通过激活其 READY 信号输出，指示访问成功完成时，该计数器会重新载入 FDCAN_RWD.WDC 的值。如果计数器递减为 0 之前消息 RAM 没有响应，计数器将停止计数，并将中断标志 FDCAN_IR.WDI 位置 1。 RAM 看门狗计数器由 FDCAN 控制逻辑时钟计数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDV[7:0]								WDC[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	ResetValue	Function
31:16	Reserved	-	-	保留
15:8	WDV[7:0]	R	8’b0	看门狗计数值 消息 RAM 看门狗计数器的实际值
7:0	WDC[7:0]	R	8’b0	看门狗配置 消息 RAM 看门狗计数器的起始值。如果使用初始值 0x00，则计数器被禁止。

Puya Confidential

31.5.5 FDCAN CC 控制寄存器（FDCAN_CCCR）

地址偏移：0x018

复位值：0x00000001

寄存器各个位的设置条件请参见【软件初始化】章节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NISO	TXP	EFBI	PXHD	WMM	Res.	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	NISO	RW	0	非 ISO 模式选择 0：使用 ISO 11898-1:2015 规定的 FD CAN 帧格式 1：使用 Bosch FD CAN 规范 V1.0 规定的 FD CAN 帧格式
14	TXP	RW	0	传输暂停 若该位被置为 1，则 FDCAN 在成功传输完一帧数据后会暂停两个位时间，再进行 下一帧数据的传输。 0：禁止传输暂停 1：使能传输暂停
13	EFBI	RW	0	总线同步期间的边沿滤波 0：禁止边沿滤波 1：需要两个连续的显性 tq 才能检测硬同步边沿
12	PXHD	RW	0	协议异常处理禁止 0：使能协议异常处理 1：禁止协议异常处理 注： 当协议异常处理被禁止时， FDCAN 将在检测到协议异常条件时传输错误帧。
11	WMM	RW	0	宽消息标志 当使用 16 位宽消息标志（WMM=1）时，发送事件 FIFO 中的 16 位内部时间戳是无效的。 0：使用 8 位消息标志 1：使用 16 位消息标志，替换发送事件 FIFO 中的 16 位时间戳
10	Reserved	-	-	保留
9	BRSE	RW	0	比特率切换 0：禁止发送时进行比特率切换 1：使能发送时进行比特率切换 注： 当 FD CAN 模式无效（FDOE=0）时， BRSE 的设置不起作用。
8	FDOE	RW	0	FD 模式使能 0：禁止 FD 模式 1：使能 FD 模式

Puya Confidential

7	TEST	RW	0	测试模式使能 0：正常模式，测试寄存器 FDCAN_TEST 保持复位值 1：测试模式，使能写访问测试寄存器 FDCAN_TEST
6	DAR	RW	0	禁止自动重发 0：使能允许自动重发未成功发送的消息 1：禁止自动重发
5	MON	RW	0	总线监听模式 仅当 CCE 和 INIT 位均置 1 时，主机才能将 MON 位置 1。此位可随时通过主机复位。 0：禁止总线监听模式 1：使能总线监听模式
4	CSR	RW	0	时钟停止请求 当已请求时钟停止时，在所有挂起的传输请求均已完成且 CAN 总线达到空闲状 态之后， INIT 位和 CSA 位将会依次被置 1。 0：未请求时钟停止 1：已请求时钟停止
3	CSA	RW	0	时钟停止请求 当已请求时钟停止时，在所有挂起的传输请求均已完成且 CAN 总线达到空闲状 态之后， INIT 位和 CSA 位将会依次被置 1。 0：未请求时钟停止 1：已请求时钟停止
2	ASM	RW	0	传输暂停 若该位被置为 1，则 FDCAN 在成功传输完一帧数据后会暂停两个位时间，再进行 下一帧数据的传输。 0：禁止传输暂停 1：使能传输暂停
1	CCE	RW	0	配置更改使能 0：CPU 对受保护的配置寄存器无写访问权限 1：CPU 对受保护的配置寄存器有写访问权限（当 INIT=1 时）
0	INIT	RW	1	0：正常工作 1：启动初始化 注： 由于两个时钟域之间存在同步机制，INIT 的值在写入后可能会有一定延迟才能被正确读取。因此，在重新设定 INIT 值之前，请确认之前的设定值确实已被写入。

31.5.6 FDCAN 标称位时间和预分频寄存器（FDCAN_NBTP）

地址偏移：0x01C

复位值：0x0600 0A03

仅当 FDCAN_CCCR.CCE 和 FDCAN_CCR.INIT 位均被置 1 时，才能对该寄存器进行写访问。 CAN 标称位时间的可编程范围为 4 到 385 个时间片（Time quanta， tq）。 CAN 时间片的可编程范围为 1 到 512 个 FDCAN 通信时钟周期（Minimum time quantum， mtq）。 tq = (NBRP + 1)mtq。

Puya Confidential

NTSEG1 为 Prop_Seg 与 Phase_Seg1 之和。 NTSEG2 为 Phase_Seg2。

因此，位时间的长度为（编程值） [NTSEG1 + NTSEG2 + 3] tq 或（功能值） [Sync_Seg + Prop_Seg +Phase_Seg1 + Phase_Seg2] tq。

信息处理时间（IPT）为零，这意味着下一位数据在采样点之后的第一个时钟边沿可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSJW[6:0]							NBRP[8:0]								
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NTSEG1[7:0]									NTSEG2[6:0]						
RW	RW	RW	RW	RW	RW	RW	RW	Res.	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:25	NSJW[6:0]	RW	7’h6	标称（重）同步跳转宽度 有效值为 0 到 127，硬件实际使用值为编程值加 1。
24:16	NBRP[8:0]	RW	9’b0	标称比特率预分频 生成位时间片的 FDCAN 通信时钟预分频值，位时间为该时间片的倍数。预分 频有效值为 0 到 511，硬件实际使用值为编程值加 1。
15:8	NTSEG1[7:0]	RW	8’hA	采样点之前的标称时间段 有效值为 1 到 255，硬件实际使用值为编程值加 1。
7	Reserved	-	-	保留
6:0	NTSEG2[6:0]	RW	7’h3	采样点之后的标称时间段 有效值为 1 到 127，硬件实际使用值为编程值加 1。

31.5.7 FDCAN 时间戳计数器配置寄存器（FDCAN_TSCC)

地址偏移：0x020

复位值：0x0000 0000

该寄存器用于配置内部 16 位时间戳计数器。内部时间戳处理的介绍请参见章节【时间戳生成】。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCP[3:0]			
												RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSS[1:0]	
														RW	RW

Bit	Name	RW	Reset Value	Function
31:20	Reserved	-	-	保留
19:16	TCP[3:0]	RW	4’b0	时间戳计数器预分频 将时间戳和超时计数器时间单位配置为 CAN 位时间的 1~16 倍。 有效设定值为 0~15，硬件实际使用值为编程值加 1。
15:2	Reserved	-	-	保留
1:0	TSS[1:0]	R	2’b0	时间戳选择 00：时间戳计数器的值一直为 0x0000 01：时间戳计数器的值根据 TCP 递增 10：禁止设定 11：时间戳计数器的值一直为 0x0000 （等同于设定值“00”）

Puya Confidential

31.5.8 FDCAN 时间戳计数器值寄存器（FDCAN_TSCV）

地址偏移：0x024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSC[15:0]															
RW	RW	RW	RW	RW	Res.	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	TSC[15:0]	RW	16’b0	时间戳计数器 在接收到或发送 SOF（Start Of Frame）时捕获的内部时间戳计数器值。 当 FDCAN_TSCC.TSS=0b01 时，时间戳计数器以 CAN 位时间的 1~16 倍进行递增，具体倍数由 FDCAN_TSCC.TCP 的设定值决定。 计数回卷时，中断标志 FDCAN_IR.TSW 会被置 1。写操作会将计数器清零。 注： 1.“回卷”是指时间戳计数器的值从非零变为零，并不是由于 FDCAN_TSCV 的写操作引起的清零。 2. 字节访问时，对该寄存器的任何一个字节进行写操作都会导致时间戳寄存器被清零。

31.5.9 FDCAN 超时计数器配置寄存器（FDCAN_TOCC）

地址偏移：0x028

复位值：0xFFFF 0000

超时计数器的介绍请参见【超时计数器】章节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TOP[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TOS[1:0]		ETOC
													RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	TOP[15:0]	RW	16’b0	超时周期 超时计数器（递减计数器）的起始值，配置超时周期。
15:3	Reserved	-	-	保留
2:1	TOS[1:0]	RW	2’b0	超时选择 在连续模式下操作时，对 FDCAN_TOCV 进行写访问会将计数器预设为 FDCAN_TOCC.TOP 配置的值，并继续向下递减。当超时计数器由其中一个 FIFO 控制时，空 FIFO 会将计数器预设为 FDCAN_TOCC.TOP 配置的值。当第一个 FIFO 元素被存储后，计数器开始递减计数。

Puya Confidential

				00：连续模式 01：由发送事件 FIFO 控制超时 10：由接收 FIFO0 控制超时 11：由接收 FIFO1 控制超时
0	ETOC	RW	0	超时计数器使能 0：禁止超时计数器 1：使能超时计数器

31.5.10 FDCAN 超时计数器值寄存器（FDCAN_TOCV）

地址偏移：0x02C

复位值：0x0000 FFFF

超时计数器的介绍请参见【超时计数器】章节。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOC[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15:0	TOC[15:0]	RW	16’hFFFF	超时计数器 超时计数器以 CAN 位时间的 1~16 倍进行递减，具体倍数由 FDCAN_TSCC.TCP 的设定值决定。当计数器递减到零时，中断标志位 FDCAN_IR.TOO 会被置 1，计数器停止计数。可以通过 FDCAN_TOCC.TOS 来设置启动和复位（或重启）条件。 注： 字节访问时，当 FDCAN_TOCC.TOS=0b00 时，对该寄存器的任何一个字节进行写操作都会将超时计数器预设为 FDCAN_TOCC.TOP 配置的值。

31.5.11 FDCAN 错误计数器寄存器（FDCAN_ECR）

地址偏移：0x040

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEL[7:0]							
								RC_R	RC_R	RC_R	RC_R	RC_R	RC_R	RC_R	RC_R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC[6:0]							TEC[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	0	
23:16	CEL[7:0]	RC_R	8’b0	CAN 错误记录 当 CAN 协议错误导致发送错误计数器 TEC 或接收错误计数器 REC 递增时，该错误记录计数器 CEL 都会递增。计数器值达到 0xFF 时，会停止计数。只有 RP 位置位但 REC 没有变化时，CEL 不会递增。 CEL 在 REC 和 TEC 增加后递增。

Puya Confidential

				对 CEL 进行读访问会复位计数器。计数器值达到 0xFF 时，停止计数， TEC 或 REC 下一次增加时中断标志 FDCAN_IR.ELO 置位。 注：读寄存器 FDCAN_PSR 会将 CEL 清零。
15	RP	R	0	接收错误被动 0：接收错误计数器低于错误被动级别 128 1：接收错误计数器已达到错误被动级别 128
14:8	REC[6:0]	R	7’b0	接收错误计数器 接收错误计数器的实际状态，计数值范围为 0~127。
7:0	TEC[7:0]	R	8’b0	发送错误计数器 发送错误计数器的实际状态，计数值范围为 0~255。

注：如果 FDCAN_CCCR.ASM 被置为 1，则检测到 CAN 协议错误时，CAN 协议控制器不会使 TEC 和 REC 递增，但 CEL 仍会递增。

31.5.12 FDCAN 协议状态寄存器（FDCAN_PSR）

地址偏移：0x044

复位值：0x0000 0717

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDCV[6:0]						
									R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	PXE	RFDF	RBRS	RESI	DLEC[2:0]			BO	EW	EP	ACT[1:0]		LEC[2:0]		
	RC_R	RC_R	RC_R	RC_R	RS_R	RS_R	RS_R	RW	RW	RW	RW	RW	RS_R	RS_R	RS_R

Bit	Name	RW	Reset Value	Function
31:23	Reserved	-	-	保留
22:16	TDCV[6:0]	R	7’b0	发送器延迟补偿值 第二采样点（Secondary Sample Point，SSP）的位置，由 FDCAN_TX 引脚到 FDCAN_RX 引脚之间测得的迟延值与 FDCAN_TDCR.TDCO 之和决定。SSP 位置在数据段。TDCV 为已发送位的起点与第二采样点之间的 mtq 数，有效值为 0~127 个 mtq。
15	Reserved	-	-	保留
14	PXE	RC_R	0	协议异常事件 0：自上次读访问后未发生协议异常事件 1：已发生协议异常事件 访问类型为 RX：进行读访问时复位。 注：读寄存器 FDCAN_PSR 会将 PXE 清零。
13	RFDF	RC_R	0	收到 FD CAN 消息 此位的置位与接收过滤无关。 0：自从上次被 CPU 清零后未收到 FD CAN 格式的消息 1：收到 FDF 标志为 1 的 FD CAN 格式的消息 注：读寄存器 FDCAN_PSR 会将 RFDF 清零。
12	RBRS	RC_R	0	上次收到的 FD CAN 消息的 BRS 标志 该位与 RFDF 一起置位，与接收过滤无关。

Puya Confidential

				0：上次收到的 FD CAN 消息的 BRS 标志未置位 1：上次收到的 FD CAN 消息的 BRS 标志已置位 注：读寄存器 FDCAN_PSR 会将 RBRS 清零。
11	RESI	RC_R	0	上次收到的 FD CAN 消息的 ESI 标志 该位与 RFDF 一起被设置，与验收过滤器无关。 0：上次收到的 FD CAN 消息的 ESI 标志未置位 1：上次收到的 FD CAN 消息的 ESI 标志已置位 注：读寄存器 FDCAN_PSR 会将 RESI 清零。
10:8	DLEC[2:0]	RS_R	3’b111	数据段最后一次错误码 在设置了 BRS 标志的 FD CAN 帧的数据段中发生的最后一个错误的类型。错误码与 LEC 相同。当设置了 BRS 标志的 FD CAN 帧传输（接收或发送）没有错误时， DLEC 将被清零。 注：读寄存器 FDCAN_PSR 会将 DLEC[2:0]置位为 3’b111。
7	BO	R	0	Buss_Off 状态 0：FDCAN 未处于 Bus_Off 状态 1：FDCAN 处于 Bus_Off 状态
6	EW	R	0	警告状态 0：两个错误计数器均小于错误警告（Error_Warning）上限 96 1：至少有一个错误计数器已达到错误警告（Error_Warning）上限 96
5	EP	R	0	错误被动 0：FDCAN 处于错误主动（Error_Active）状态。FDCAN 正常参与总线通信，并在检测到错误时发送主动错误标志。 1：FDCAN 处于错误被动（Error_Passive）状态
4:3	ACT[1:0]	R	2’b0	通信状态 监测 FDCAN 的通信状态。 00：同步中，节点正在进行 CAN 通信同步 01：空闲，节点既不是接收器也不是发送器 10：接收器，节点作为接收器工作 11：发送器，节点作为发送器工作 注： 协议异常事件会将 ACT 设为 2’b00
2:0	LEC[2:0]	RS_R	3’b111	最后一次错误码 0：没有错误（No Error）。自成功接收或发送使 LEC 复位之后，没有错误发生。 1： 填充错误（Stuff Error）。在已接收到的消息中，某一部分出现了 5 个以上的相同 位，这种情况是不允许的。 2：格式错误（Form Error）。已接收帧的固定格式部分有格式错误。 3：ACK 错误（ACK Error）。 FDCAN 发送的消息没有被其他节点确认。 4：位 1 错误（Bit1 Error）。在消息的发送过程中（仲裁段除外），设备想发送一个隐性电平（逻辑值 1），但是监测到的总线电平为显性（逻辑值 0）。

Puya Confidential

				5：Bit0 错误（Bit0 Error）。在消息的发送过程中（包括 ACK 位、主动错误标志、过载标志），设备想发送一个显性电平（数据或 ID 的逻辑值 0），但是监测到的总线电平为隐性（逻辑值 1）。在总线 Bus_Off 恢复期间，每次监测到 11 位隐性位序列时，都会设置为该状态。这样， CPU 就能够监测总线恢复序列的进程（表明总线未卡在显性状态或持续受到干扰）。 6： CRC 错误（CRC Error）。已接收消息的 CRC 有错误。收到的 CRC 和用接收到的数据计算出的 CRC 不匹配。 7：无变化（No Change）。对协议状态寄存器的任何读访问会将 LEC 重新初始化为 7。 当 LEC 显示值为 7 时， 表示自上次 CPU 对协议状态寄存器进行读访问之后，没有检测到 CAN 总线事件。 注： 1.当设置了 BRS 标志的 FDCAN 帧已经到达数据段时，下一个 CAN 事件（错误或有效帧）将显示在 DLEC 中而不是 LEC 中。FDCANCRC 序列中的固定填充位的错误将显示为格式错误（FormError）而不是填充错误（StuffError）。 2.Bus_Off 恢复序列（请参考 ISO11898-1:2015）不能通过置位或清零 FDCAN_CCCR.INIT 来缩短。如果设备进入 Bus_Off 状态，它将自动置位 FDCAN_CCCR.INIT，停止所有总线活动。一旦 FDCAN_CCCR.INIT 被 CPU 清零，设备在恢复正常操作之前，将等待 129 个总线空闲（129*11 个连续的隐性位）。在 Bus_Off 恢复序列结束时，错误管理计数器将被复位。在清零 FDCAN_CCCR.INIT 后的等待期间，每次监测到一个 11 位隐性序列时，就向 FDCAN_PSR.LEC 写入一个 Bit0 错误码，使 CPU 能够随时检查 CAN 总线是否处于显性状态或持续被干扰，并监测 Bus_Off 恢复序列。FDCAN_ECR.REC 用于对这些序列进行计数。 3. 读寄存器 FDCAN_PSR 会将 LEC[2:0]置位为 3’b111。

31.5.13 FDCAN 发送器延迟补偿寄存器（FDCAN_TDCR）

地址偏移：0x048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	TDCO[6:0]							Res	TDCF[6:0]						
	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:15	Reserved	-	-	保留
14:8	TDCO[6:0]	RW	7’b0	发送器延迟补偿 SSP 偏移 该偏移值是指，从 FDCAN_TX 引脚到 FDCAN_RX 引脚测得的延迟与第二采样点（SSP）之间的距离。有效设定值为 0~127mtq。
7	Reserved	-	-	保留
6:0	TDCF[6:0]	RW	7’b0	发送器延迟补偿滤波窗口宽度

PuyaConfidential

				定义 SSP 位置的最小值，对于发送器延迟的测量，将忽略 FDCAN_RX 引脚上会导致 SSP 位置提前的显性边沿。当 TDCF 设定值大于 TDCO 时，该功能被启用。有效设定值为 0~127mtq。

31.5.14 FDCAN 中断寄存器（FDCAN_IR）

地址偏移：0x050

复位值：0x0000 0000

当如下列表中的一个条件被检测到时，对应标志位会置位。在主机将标志位清零之前，保持置位状态。所有标志位写 1 清零，写 0 无影响。硬复位会将寄存器清零。是否产生中断，由寄存器 FDCAN_IE 控制；哪条中断线产生中断，由寄存器 FDCAN_ILS 控制。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARA	PED	PEA	WDI	BO	EW	EP	ELO	Res.	Res.	DRX	TOO	MRAF	TSW
		RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29	ARA	RW	0	访问保留地址 (Access to Reserved Address) 0：未对保留地址进行访问 1：已对保留地址进行访问
28	PED	RW	0	数据段协议错误（使用数据位时间）(Protocol Error in Data Phase (Data Bit Time isused)) 0：未检测到数据段协议错误 1： 检测到数据段协议错误（FDCAN_PSR.DLEC≠0/7）
27	PEA	RW	0	仲裁段协议错误（使用标称位时间） (Protocol Error in Arbitration Phase (NominalBit Time is used)) 0：未检测到仲裁段协议错误 1： 检测到仲裁段协议错误（FDCAN_PSR.LEC≠0/7）
26	WDI	RW	0	看门狗中断 (Watchdog Interrupt) 0：未发生消息 RAM 看门狗事件 1：因 READY 信号缺失而发生消息 RAM 看门狗事件
25	BO	RW	0	Bus_Off 状态 (Bus_Off Status) 0： Bus_Off 状态未改变 1： Bus_Off 状态已改变
24	EW	RW	0	警告状态 (Warning Status) 0：错误警告（Error_Warning）状态未改变 1：错误警告（Error_Warning）状态已改变
23	EP	RW	0	错误被动 (Error Passive) 0：错误被动（Error_Passive）状态未改变 1：错误被动（Error_Passive）状态已改变
22	ELO	RW	0	错误记录溢出 (Error Logging Overflow) 0： CAN 错误记录计数器未溢出 1： CAN 错误记录计数器已溢出
21:20	Reserved	-	-	保留

Puya Confidential

19	DRX	RW	0	消息存储到专用接收缓冲区（Message stored to Dedicated Rx Buffer) 每当收到的消息已存储到专用接收缓冲区时，都会置位该标志。 0：接收缓存区未更新 1：至少有一条消息已存储到接收缓冲区
18	TOO	RW	0	发生超时（Timeout Occurred) 0：未发生超时 1：已发生超时
17	MRAF	RW	0	消息 RAM 访问失败（Message RAM Access Failure) 接收处理中置位条件： 1） 在接收到后续消息的仲裁段之前，尚未完成对已接收消息的接收过滤或存储。在这种情况下，接收过滤或消息存储被终止，接收处理单元将开始处理后续消息。 2） 无法向消息 RAM 写入消息，消息存储将终止。 在这两种情况下， FIFO 放入索引都不会更新，专用接收缓冲区的新数据标志不会置位，当下一条消息存储到该位置时，已部分存储的消息会被覆盖。 发送处理中置位条件： 1） 发送处理单元无法及时从消息 RAM 读取消息。在这种情况下，消息发送将终止。 在发送处理单元访问失败的情况下， FDCAN 被切换到受限工作模式（详见章节【受限工作模式】 ）。 要退出受限工作模式，主机 CPU 必须复位 FDCAN_CCCR.ASM。 0：未发生消息 RAM 访问失败 1：发生了消息 RAM 访问失败
16	TSW	RW	0	时间戳回卷（Timestamp Wraparound) 0：时间戳计数器未回卷 1：时间戳计数器已回卷
15	TEFL	RW	0	发送事件 FIFO 元素丢失（Tx Event FIFO Element Lost) 0：发送事件 FIFO 元素未丢失 1：发送事件 FIFO 元素有丢失；对空间大小为 0 的发送事件 FIFO 进行写操作时，该位也会置位
14	TEFF	RW	0	发送事件 FIFO 已满（Tx Event FIFO Full) 0：发送事件 FIFO 未满 1：发送事件 FIFO 已满
13	TEFW	RW	0	发送事件 FIFO 到达水位线（Tx Event FIFO Watermark Reached) 0：发送事件 FIFO 的填充级别低于水位线 1：发送事件 FIFO 的填充级别达到水位线
12	TEFN	RW	0	发送事件 FIFO 新条目（Tx Event FIFO New Entry) 0：发送事件 FIFO 无变化 1：发送处理单元写入了发送事件 FIFO 元素
11	TFE	RW	0	发送 FIFO 为空（Tx FIFO Empty) 0：发送 FIFO 非空 1：发送 FIFO 为空
10	TCF	RW	0	发送取消已完成（Transmission Cancellation Finished)

Puya Confidential

				0：发送取消未完成 1：发送取消已完成
9	TC	RW	0	发送完成 (Transmission Completed) 0：发送未完成 1：发送已完成
8	HPM	RW	0	高优先级消息 (High Priority Message) 0：未收到高优先级消息 1：已收到高优先级消息
7	RF1L	RW	0	接收 FIFO1 消息丢失 (Rx FIFO 1 Message Lost) 0：接收 FIFO1 消息未丢失 1：接收 FIFO1 消息有丢失；对空间大小为 0 的接收 FIFO1 进行写操作时，该位也会置位
6	RF1F	RW	0	接收 FIFO1 已满 (Rx FIFO 1 Full) 0：接收 FIFO1 未满 1：接收 FIFO1 已满
5	RF1W	RW	0	接收 FIFO1 到达水位线 (Rx FIFO 1 Watermark Reached) 0：接收 FIFO1 的填充级别低于水位线 1：接收 FIFO1 的填充级别达到水位线
4	RF1N	RW	0	接收 FIFO1 新消息 (Rx FIFO 1 New Message) 0：没有新消息写入接收 FIFO1 1：有新消息写入接收 FIFO1
3	RF0L	RW	0	接收 FIFO0 消息丢失 (Rx FIFO 0 Message Lost) 0：接收 FIFO0 消息未丢失 1：接收 FIFO0 消息有丢失；对空间大小为 0 的接收 FIFO0 进行写操作时，该位也会置位
2	RF0F	RW	0	接收 FIFO0 已满 (Rx FIFO 0 Full) 0：接收 FIFO0 未满 1：接收 FIFO0 已满
1	RF0W	RW	0	接收 FIFO0 到达水位线 (Rx FIFO 0 Watermark Reached) 0：接收 FIFO0 的填充级别低于水位线 1：接收 FIFO0 的填充级别达到水位线
0	RF0N	RW	0	接收 FIFO0 新消息 (Rx FIFO 0 New Message) 0：没有新消息写入接收 FIFO0 1：有新消息写入接收 FIFO0

31.5.15 FDCAN 中断使能寄存器（FDCAN_IE）

地址偏移：0x054

复位值：0x0000 0000

中断使能寄存器中的设置，决定中断寄存器（FDCAN_IR）中的哪些状态变化会指示在中断线上。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	Res.	Res.	DRXE	TOOE	MRAFE	TSWE
		RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29	ARAE	RW	0	访问保留地址使能 (Access to Reserved Address Enable) 0：禁止中断 1：使能中断
28	PEDE	RW	0	数据阶段的协议错误使能 (Protocol Error in Data Phase Enable) 0：禁止中断 1：使能中断
27	PEAE	RW	0	仲裁阶段中的协议错误使能 (Protocol Error in Arbitration Phase Enable) 0：禁止中断 1：使能中断
26	WDIE	RW	0	看门狗中断使能 (Watchdog Interrupt Enable) 0：禁止中断 1：使能中断
25	BOE	RW	0	Bus_Off 状态中断 (Bus_Off Status Interrupt Enable) 0：禁止中断 1：使能中断
24	EWE	RW	0	警告状态中断使能 (Warning Status Interrupt Enable) 0：禁止中断 1：使能中断
23	EPE	RW	0	错误被动中断使能 (Error Passive Interrupt Enable) 0：禁止中断 1：使能中断
22	ELOE	RW	0	错误记录溢出中断使能 (Error Logging Overflow Interrupt Enable) 0：禁止中断 1：使能中断
21:20	Reserved	-	-	保留
19	DRXE	RW	0	消息存储到专用接收缓冲区中断使能 (Message stored to Dedicated RxBuffer Interrupt Enable) 0：禁止中断 1：使能中断
18	TOOE	RW	0	发生超时中断使能 (Timeout Occurred Interrupt Enable) 0：禁止中断 1：使能中断
17	MRAFE	RW	0	消息 RAM 访问失败中断使能 (Message RAM Access Failure Interrupt Enable) 0：禁止中断 1：使能中断
16	TSWE	RW	0	时间戳回卷中断使能 (Timestamp Wraparound Interrupt Enable) 0：禁止中断 1：使能中断
15	TEFLE	RW	0	发送事件 FIFO 元素丢失中断使能 (Tx Event FIFO Element Lost Interrupt Enable) 0：禁止中断

Puya Confidential

				1：使能中断
14	TEFFE	RW	0	发送事件 FIFO 满中断使能 (Tx Event FIFO Full Interrupt Enable) 0：禁止中断 1：使能中断
13	TEFWE	RW	0	发送事件 FIFO 到达水位线中断使能(Tx Event FIFO Watermark Reached Interrupt Enable) 0：禁止中断 1：使能中断
12	TEFNE	RW	0	发送事件 FIFO 新条目中断使能 (Tx Event FIFO New Entry Interrupt Enable) 0：禁止中断 1：使能中断
11	TFEE	RW	0	发送 FIFO 空中断使能 (Tx FIFO Empty Interrupt Enable) 0：禁止中断 1：使能中断
10	TCFE	RW	0	发送取消已完成中断使能 (Transmission Cancellation Finished InterruptEnable) 0：禁止中断 1：使能中断
9	TCE	RW	0	发送完成中断使能 (Transmission Completed Interrupt Enable) 0：禁止中断 1：使能中断
8	HPME	RW	0	高优先级消息中断使能 (High Priority Message Interrupt Enable) 0：禁止中断 1：使能中断
7	RF1LE	RW	0	接收 FIFO1 消息丢失中断使能 (Rx FIFO 1 Message Lost Interrupt Enable) 0：禁止中断 1：使能中断
6	RF1FE	RW	0	接收 FIFO1 满中断使能 (Rx FIFO 1 Full Interrupt Enable) 0：禁止中断 1：使能中断
5	RF1WE	RW	0	接收 FIFO1 到达水位线中断使能 (Rx FIFO 1 Watermark Reached Interrupt Enable) 0：禁止中断 1：使能中断
4	RF1NE	RW	0	接收 FIFO1 新消息中断使能 (Rx FIFO 1 New Message Interrupt Enable) 0：禁止中断 1：使能中断
3	RF0LE	RW	0	接收 FIFO0 消息丢失中断使能 (Rx FIFO 0 Message Lost Interrupt Enable) 0：禁止中断 1：使能中断
2	RF0FE	RW	0	接收 FIFO0 满中断使能 (Rx FIFO 0 Full Interrupt Enable) 0：禁止中断 1：使能中断

Puya Confidential

1	RF0WE	RW	0	接收 FIFO0 到达水位线中断使能(Rx FIFO 0 Watermark Reached Interrupt Enable) 0：禁止中断 1：使能中断
0	RF0NE	RW	0	接收 FIFO0 新消息中断使能 (Rx FIFO 0 New Message Interrupt Enable) 0：禁止中断 1：使能中断

31.5.16 FDCAN 中断线选择寄存器（FDCAN_ILS）

地址偏移：0x058

复位值：0x0000 0000

中断线选择寄存器，将中断寄存器（FDCAN_IR）中特定中断标志生成的中断分配到两条中断线之一。要产生中断，必须通过 FDCAN_ILE.EINT0 和 FDCAN_ILE.EINT1 使能相应的中断线。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	Res.	Res.	DRXL	TOOL	MRAFL	TSWL
		RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29	ARAL	RW	0	访问保留地址中断线 (Access to Reserved Address Line) 0：选择中断线 0 1：选择中断线 1
28	PEDL	RW	0	数据段协议错误中断线 (Protocol Error in Data Phase Line) 0：选择中断线 0 1：选择中断线 1
27	PEAL	RW	0	仲裁段协议错误中断线 (Protocol Error in Arbitration Phase Line) 0：选择中断线 0 1：选择中断线 1
26	WDIL	RW	0	看门狗中断线 (Watchdog Interrupt Line) 0：选择中断线 0 1：选择中断线 1
25	BOL	RW	0	Bus_Off 状态中断线 (Bus_Off Status Interrupt Line) 0：选择中断线 0 1：选择中断线 1
24	EWL	RW	0	警告状态中断线 (Warning Status Interrupt Line) 0：选择中断线 0 1：选择中断线 1
23	EPL	RW	0	错误被动中断线 (Error Passive Interrupt Line) 0：选择中断线 0 1：选择中断线 1
22	ELOL	RW	0	错误记录溢出中断线 (Error Logging Overflow Interrupt Line)

Puya Confidential

				0：选择中断线 0 1：选择中断线 1
21:20	Reserved	-	-	保留
19	DRXL	RW	0	消息存储到专用接收缓冲区中断线 (Message stored to Dedicated Rx Buffer Interrupt Line) 0：选择中断线 0 1：选择中断线 1
18	TOOL	RW	0	发生超时中断线 (Timeout Occurred Interrupt Line) 0：选择中断线 0 1：选择中断线 1
17	MRAFL	RW	0	消息 RAM 访问失败中断线 (Message RAM Access Failure Interrupt Line) 0：选择中断线 0 1：选择中断线 1
16	TSWL	RW	0	时间戳回卷中断线 (Timestamp Wraparound Interrupt Line) 0：选择中断线 0 1：选择中断线 1
15	TEFLL	RW	0	发送事件 FIFO 元素丢失中断线 (Tx Event FIFO Element Lost Interrupt Line) 0：选择中断线 0 1：选择中断线 1
14	TEFFL	RW	0	发送事件 FIFO 已满中断线 (Tx Event FIFO Full Interrupt Line) 0：选择中断线 0 1：选择中断线 1
13	TEFWL	RW	0	发送事件 FIFO 到达水位线中断线 (Tx Event FIFO Watermark Reached Interrupt Line) 0：选择中断线 0 1：选择中断线 1
12	TEFNL	RW	0	发送事件 FIFO 新条目中断线 (Tx Event FIFO New Entry Interrupt Line) 0：选择中断线 0 1：选择中断线 1
11	TFEL	RW	0	发送 FIFO 为空中断线 (Tx FIFO Empty Interrupt Line) 0：选择中断线 0 1：选择中断线 1
10	TCFL	RW	0	发送取消已完成中断线 (Transmission Cancellation Finished Interrupt Line) 0：选择中断线 0 1：选择中断线 1
9	TCL	RW	0	发送完成中断线 (Transmission Completed Interrupt Line) 0：选择中断线 0 1：选择中断线 1
8	HPML	RW	0	高优先级消息中断线 (High Priority Message Interrupt Line) 0：选择中断线 0 1：选择中断线 1
7	RF1LL	RW	0	接收 FIFO1 消息丢失中断线 (Rx FIFO 1 Message Lost Interrupt Line)

Puya Confidential

				0：选择中断线 0 1：选择中断线 1
6	RF1FL	RW	0	接收 FIFO1 已满中断线 (Rx FIFO 1 Full Interrupt Line) 0：选择中断线 0 1：选择中断线 1
5	RF1WL	RW	0	接收 FIFO1 到达水位线中断线 (Rx FIFO 1 Watermark Reached Interrupt Line) 0：选择中断线 0 1：选择中断线 1
4	RF1NL	RW	0	接收 FIFO1 新消息中断线 (Rx FIFO 1 New Message Interrupt Line) 0：选择中断线 0 1：选择中断线 1
3	RF0LL	RW	0	接收 FIFO0 消息丢失中断线 (Rx FIFO 0 Message Lost Interrupt Line) 0：选择中断线 0 1：选择中断线 1
2	RF0FL	RW	0	接收 FIFO0 已已满中断线 (Rx FIFO 0 Full Interrupt Line) 0：选择中断线 0 1：选择中断线 1
1	RF0WL	RW	0	接收 FIFO0 到达水位线中断线 (Rx FIFO 0 Watermark Reached Interrupt Line) 0：选择中断线 0 1：选择中断线 1
0	RF0NL	RW	0	接收 FIFO0 新消息中断线 (Rx FIFO 0 New Message Interrupt Line) 0：选择中断线 0 1：选择中断线 1

31.5.17 FDCAN 中断线使能寄存器（FDCAN_ILE）

地址偏移：0x05C

复位值：0x0000 0000

连接到 CPU 的两条中断线中的每一条，都可以通过设置 EINT0 和 EINT1 分别使能或禁用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EINT1	EINT0
														RW	RW

Bit	Name	RW	Reset Value	Function
31:2	Reserved	-	-	保留
1	EINT1	RW	0	中断线 1 使能 0：中断线 1 禁止 1：中断线 1 使能
0	EINT0	RW	0	中断线 0 使能 0：中断线 0 禁止 1：中断线 0 使能

Puya Confidential

31.5.18 FDCAN 全局过滤器配置寄存器（FDCAN_GFC）

地址偏移：0x080

复位值：0x0000 0000

用于消息 ID 过滤的全局设置。全局过滤器配置，控制标准消息和扩展消息的过滤路径。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
										RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:6	Reserved	-	-	保留
5:4	ANFS[1:0]	RW	2’b0	接受不匹配的标准帧 定义如何处理，已接收的与过滤器列表任何元素都不匹配的 11 位标准 ID 消息。 00：在接收 FIFO0 中接受不匹配的标准帧 01：在接收 FIFO1 中接受不匹配的标准帧 10：拒绝不匹配的标准帧 11：拒绝不匹配的标准帧
3:2	ANFE[1:0]	RW	2’b0	接受不匹配的扩展帧 定义如何处理，已接收的与过滤器列表任何元素都不匹配的 29 位扩展 ID 消息。 00：在接收 FIFO0 中接受不匹配的扩展帧 01：在接收 FIFO1 中接受不匹配的扩展帧 10：拒绝不匹配的扩展帧 11：拒绝不匹配的扩展帧
1	RRFS	RW	0	拒绝远程标准帧 0：过滤 11 位标准 ID 的远程帧 1：拒绝所有 11 位标准 ID 的远程帧
0	RRFE	RW	0	拒绝远程扩展帧 0：过滤 29 位扩展 ID 的远程帧 1：拒绝所有 29 位扩展 ID 的远程帧

31.5.19 FDCAN 标准 ID 过滤器配置寄存器（FDCAN_SIDFC）

地址偏移：0x084

复位值：0x0000 0000

用于 11 位标准 ID 过滤器的设置。标准 ID 过滤器配置，控制标准消息的过滤路径。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LLS[7:0]							
								RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FLSSA[15:2]														Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Name	RW	Reset Value	Function
31:24	Reserved	-	-	保留
23:16	LSS[7:0]	RW	8’b0	标准 ID 过滤器列表大小 0：无标准 ID 过滤器

Puya Confidential

				1~128：标准 ID 过滤器元素数量 ＞128：大于 128 的设定值被解析为 128
15:2	FLSSA[15:2]	RW	14’b0	标准 ID 过滤器列表起始地址 设置标准 ID 过滤器列表的起始地址（32 位字地址）
1:0	Reserved	-	-	保留

31.5.20 FDCAN 扩展 ID 过滤器配置寄存器（FDCAN_XIDFC）

地址偏移：0x088

复位值：0x0000 0000

用于 29 位扩展 ID 过滤器的设置。扩展 ID 过滤器配置，控制扩展消息的过滤路径。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSE[6:0]						
									RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FLESA[15:2]														Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Name	RW	Reset Value	Function
31:23	Reserved	-	-	保留
22:16	LSE[7:0]	RW	8’b0	扩展 ID 过滤器列表大小 0：无扩展 ID 过滤器 1~64：扩展 ID 过滤器元素数量 ＞64：大于 64 的设定值被解析为 64
15:2	FLESA[15:2]	RW	14’b0	扩展 ID 过滤器列表起始地址 扩展 ID 过滤器列表的起始地址（32 位字地址）
1:0	Reserved	-	-	保留

31.5.21 FDCAN 扩展 ID 与掩码寄存器（FDCAN_XIDAM）

地址偏移：0x090

复位值：0x1FFF FFFF

用于 29 位扩展 ID 过滤器的设置。扩展 ID 过滤器配置，控制扩展消息的过滤路径。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EIDM[28:16]												
			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EIDM[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:29	Reserved	-	-	保留
28:0	EIDM[28:0]	RW	29’b0	扩展 ID 掩码 对于扩展帧的接收过滤，会将 EIDM 与已接收帧的 ID 进行与运算。用于屏蔽 SAE J1939 中的 29 位 ID。所有位的复位值都被设置为 1，掩码无效。

Puya Confidential

31.5.22 FDCAN 高优先级消息状态寄存器（FDCAN_HPMS）

地址偏移：0x094

复位值：0x0000 0000

配置为生成优先级事件的消息 ID 过滤器元素，每次匹配时都会更新此寄存器。这可用于监测收到的高优先级消息的状态，并可对这些消息进行快速访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FLST	FIDX[6:0]							MSI[1:0]		BIDX[5:0]					
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	-	保留
15	FLST	R	0	过滤器列表 指示匹配的过滤器列表。 0：标准过滤器列表 1：扩展过滤器列表
14:8	FIDX[6:0]	R	7’b0	过滤器索引 匹配的过滤器元素的索引。范围为 0 到 FDCAN_SIDFC.LSS 减 1 或 FDCAN_XIDFC.LSE 减 1。
7:6	MSI[1:0]	R	2’b0	消息存储器指示符 00：未选择 FIFO 01：FIFO 消息丢失 10：消息存储在 FIFO0 中 11：消息存储在 FIFO1 中
5:0	BIDX[5:0]	R	6’b0	缓存区索引 存储消息的接收 FIFO 元素的索引。仅当 MSI[1]=1 时有效。

31.5.23 FDCAN 新数据 1 寄存器（FDCAN_NDAT1）

地址偏移：0x098

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	ND[31:0]	RW	32’b0	新数据 该寄存器保存接收缓冲区 0 到 31 的新数据标志。当接收帧更新接收缓冲区时，对应的标志位会置 1。在主机将标志位清零之前，标志位保持置 1 状态。通过向对应位写 1 将标志位清零，写入 0 不起任何作用。硬复位会将寄存器清零。 0：接收缓冲区未更新

Puya Confidential

				1：接收缓冲区通过新消息更新

31.5.24 FDCAN 新数据 2 寄存器（FDCAN_NDAT2）

地址偏移：0x09C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	ND[63:32]	RW	32’b0	新数据 该寄存器保存接收缓冲区 32 到 63 的新数据标志。当接收帧更新接收缓冲区时，对应的标志位会置 1。在主机将标志位清零之前，标志位保持置 1 状态。通过向对应位写 1 将标志位清零，写入 0 不起任何作用。硬复位会将寄存器清零。 0：接收缓冲区未更新 1：接收缓冲区通过新消息更新

31.5.25 FDCAN 接收 FIFO0 配置寄存器（FDCAN_RXF0C）

地址偏移：0x0A0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F0OM	F0WM[6:0]							Res.	F0S[6:0]						
RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F0SA[15:2]														Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Name	RW	Reset Value	Function
31	F0OM	RW	0	接收 FIFO0 工作模式 FIFO0 可工作在阻塞模式（Blocking mode）或覆盖模式（Overwrite mode） 0：接收 FIFO0 为阻塞模式 1：接收 FIFO0 为覆盖模式
30:24	F0WM[6:0]	RW	0	接收 FIFO0 水位线 0：禁止接收 FIFO0 水位线中断 1~64：接收 FIFO0 水位线中断（FDCAN_IR.RF0W）的级别 ＞64：禁止接收 FIFO0 水位线中断
23	Reserved	-	-	保留
22:16	F0S[6:0]	RW	0	接收 FIFO0 大小 0：无接收 FIFO0 1~64：接收 FIFO0 元素数量 ＞64：大于 64 的设定值被解析为 64 接收 FIFO0 元素的索引为 0 到 F0S 减 1

Puya Confidential

15:2	F0SA[15:2]	RW	0	接收 FIFO0 起始地址 消息 RAM 中接收 FIFO0 的起始地址（32 位字地址）
1:0	Reserved	-	-	保留

31.5.26 FDCAN 接收 FIFO0 状态寄存器（FDCAN_RXF0S）

地址偏移：0x0A4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	RF0L	F0F	Res.	Res.	F0PI[5:0]					
						R	R			R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	F0GI[5:0]						Res.	F0FL[6:0]						
		R	R	R	R	R	R		R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25	RF0L	R	0	接收 FIFO0 消息丢失 该位是中断标志 FDCAN_IR.RF0L 的副本。当 FDCAN_IR.RF0L 被复位时，该位也被复位 0：接收 FIFO0 无消息丢失 1：接收 FIFO0 有消息丢失；对空间大小为 0 的接收 FIFO0 进行写操作时，该位也会置位 注： 当 FDCAN_RXF0C.F0OM=1 时，覆盖最早的消息，不会置位该标志位。
24	F0F	R	0	接收 FIFO0 满 0：接收 FIFO0 未满 1：接收 FIFO0 已满
23:22	Reserved	-	-	保留
21:16	F0PI[5:0]	R	6’b0	Rx FIFO 0 写入索引 Rx FIFO 0 写入索引指针，范围为 0 到 63。
15:14	Reserved	-	-	保留
13:8	F0GI[5:0]	R	6’b0	接收 FIFO0 获取索引 接收 FIFO0 读取索引指针，范围 0~63
7	Reserved	-	-	保留
6:0	F0FL[6:0]	R	7’b0	接收 FIFO0 填充级别 接收 FIFO0 中存储的元素数量，范围 0~64

31.5.27 FDCAN 接收 FIFO0 确认寄存器（FDCAN_RXF0A）

地址偏移：0x0A8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F0AI[5:0]					
										RW	RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:6	Reserved	-	-	保留
5:0	F0AI[5:0]	RW	6’b0	接收 FIFO0 确认索引 主机从接收 FIFO0 读取消息或消息序列后，必须将从接收 FIFO0 中读取的最后一个元素的缓冲区索引写入 F0AI。此操作会将接收 FIFO0 获取索引 FDCAN_RXF0S.F0GI 设为 F0AI 加 1，并会更新 FIFO0 填充级别 FDCAN_RXF0S.F0FL

31.5.28 FDCAN 接收缓冲区配置寄存器（FDCAN_RXBC）

地址偏移：0x0AC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBSA[15:2]														Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Name	RW	Reset Value	Function
31:16	Reserved	-	0	保留
15:2	RBSA[15:2]	RW	14’b0	接收缓冲区的起始地址 配置消息 RAM 中接收缓冲区的起始地址（32 位地址），也用于引用调试消息 A、B、C
1:0	Reserved	-	0	保留

31.5.29 FDCAN 接收 FIFO1 配置寄存器（FDCAN_RXF1C）

地址偏移：0x0B0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F1OM	F1WM[6:0]							Res	F1S[6:0]						
RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F1SA[15:2]														Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Name	RW	Reset Value	Function
31	F1OM	RW	0	接收 FIFO1 操作模式 FIFO1 可工作在阻塞模式（Blocking mode）或覆盖模式（Overwrite mode） 0：接收 FIFO1 为阻塞模式 1：接收 FIFO1 为覆盖模式
30:24	F1WM[6:0]	RW	7’b0	接收 FIFO1 水印 0：禁止接收 FIFO1 水位线中断 1~64：接收 FIFO1 水位线中断（FDCAN_IR.RF1W）的级别 ＞64：禁止接收 FIFO1 水位线中断
23	Reserved	-	-	保留
22:16	F1S[6:0]	RW	7’b0	接收 FIFO1 大小

Puya Confidential

				0：无接收 FIFO1 1~64：接收 FIFO1 元素数量 ＞64：大于 64 的设定值被解析为 64 接收 FIFO1 元素的索引为 0 到 F1S 减 1
15:2	F1SA[15:2]	RW	14’b0	接收 FIFO1 起始地址 消息 RAM 中接收 FIFO1 的起始地址（32 位字地址）
1:0	Reserved	-	-	保留

31.5.30 FDCAN 接收 FIFO1 状态寄存器（FDCAN_RXF1S）

地址偏移：0x0B4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMS[1:0]		Res.	Res.	Res.	Res.	RF1L	F1F	Res.	Res.	F1PI[5:0]					
R	R					R	R			R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	F1GI[5:0]						Res.	F1FL[6:0]						
		R	R	R	R	R	R		R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:30	DMS[1:0]	R	2’b0	调试消息状态 00：空闲状态，等待接收调试消息 01：已接收调试消息 A 10：已接收调试消息 A、 B 11：已接收调试消息 A、 B、 C
29:26	Reserved	-	-	保留
25	RF1L	R	0	接收 FIFO1 消息丢失 0：接收 FIFO1 无元素丢失 1：接收 FIFO1 有元素丢失；对空间大小为 0 的接收 FIFO1 进行写操作时，该位也会置位 注： 当 FDCAN_RXF1C.F1OM=1 时，覆盖最早的消息，不会置位该标志位。
24	F1F	R	0	接收 FIFO1 满 0：接收 FIFO1 未满 1：接收 FIFO1 已满
23:22	Reserved	-	-	保留
21:16	F1PI[5:0]	R	6’b0	接收 FIFO1 放入索引 接收 FIFO1 写入索引指针，范围 0~63
15:14	Reserved	-	-	保留
13:8	F1GI[5:0]	R	6’b0	接收 FIFO1 获取索引 接收 FIFO1 读取索引指针，范围 0~63
7	Reserved	-	-	保留
6:0	F1FL[6:0]	R	7’b0	接收 FIFO1 填充级别 接收 FIFO1 中存储的元素数量，范围 0~64

Puya Confidential

31.5.31 FDCAN 接收 FIFO1 确认寄存器（FDCAN_RXF1A）

地址偏移：0x0B8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F1AI[5:0]					
										RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:6	Reserved	-	-	保留
5:0	F1AI[5:0]	RW	6’b0	接收 FIFO1 确认索引 主机从接收 FIFO1 读取消息或消息序列后，必须将从接收 FIFO1 中读取的最后一个元素的缓冲区索引写入 F1AI。此操作会将接收 FIFO1 获取索引 FDCAN_RXF1S.F1GI 设为 F1AI 加 1，并会更新 FIFO1 填充级别 FDCAN_RXF1S.F1FL

31.5.32 FDCAN 接收缓冲区和 FIFO 元素大小配置寄存器（FDCAN_RXESC）

地址偏移：0x0BC

复位值：0x0000 0000

配置接收缓冲区和接收 FIFO 元素的数据字节数。大于 8 字节的数据段仅适用于 FD CAN 模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	RBDS[2:0]			Res.	F1DS[2:0]			Res.	F0DS[2:0]		
					RW	RW	RW		RW	RW	RW		RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:11	Reserved	-	-	保留
10:8	RBDS[2:0]	RW	3’b0	接收缓冲区数据段大小 000：8 字节 001：12 字节 010：16 字节 011：20 字节 100：24 字节 101：32 字节 110：48 字节 111：64 字节
7	Reserved	-	-	保留
6:4	F1DS[2:0]	RW	3’b0	接收 FIFO1 数据段大小 000：8 字节 001：12 字节 010：16 字节 011：20 字节

Puya Confidential

				100：24 字节 101：32 字节 110：48 字节 111：64 字节
3	Reserved	-	-	保留
2:0	F0DS[2:0]	RW	3’b0	接收 FIFO0 数据段大小 000：8 字节 001：12 字节 010：16 字节 011：20 字节 100：24 字节 101：32 字节 110：48 字节 111：64 字节

31.5.33 FDCAN 发送缓冲区配置寄存器（FDCAN_TXBC）

地址偏移：0x0C0

复位值：0x0000 0000

仅当 FDCAN_CCCR 寄存器的 CCE 位和 INIT 位均被置 1 时，才能对该寄存器进行写访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TFQM	TFQS[5:0]						Res.	Res.	NDTB[5:0]					
	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TBSA[13:0]														Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Name	RW	Reset Value	Function
31	Reserved	-	-	保留
30	TFQM	RW	0	发送 FIFO/队列模式 0：发送 FIFO 模式 1：发送队列模式
29:24	TFQS[5:0]	RW	6’b0	发送 FIFO/队列大小 0：无发送 FIFO 或队列 1-32：发送 FIFO 队列的缓冲区大小 ＞32：大于 32 的设定值被解析为 32
23:22	Reserved	-	-	保留
21:16	NDTB[5:0]	RW	6’b0	专用发送缓冲区大小 0：无专用发送缓冲区 1-32：专用发送缓冲区大小 ＞32：大于 32 的设定值被解析为 32
15:2	TBSA[13:0]	RW	14’b0	发送缓冲区起始地址 消息 RAM 中发送缓冲区部分的起始地址（32 位字地址）
1:0	Reserved	-	-	保留

注意：TFQS 和 NDTB 之和不得大于 32。FDCAN 没有对该设置错误的监测。消息 RAM 中的发送缓冲区部分从专用发送缓冲区开始。

Puya Confidential

31.5.34 FDCAN 发送 FIFO/队列状态寄存器（FDCAN_TXFQS）

地址偏移：0x0C4

复位值：0x0000 0000

发送 FIFO 或队列的状态与寄存器 FDCAN_TXBRP 中挂起的发送请求有关。因此，由于正在运行的发送扫描（TXBRP 尚未更新），添加或取消请求的效果可能会延迟。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFQF	TFQPI[4:0]				
										R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TFGI[4:0]					Res.	Res.	TFFL[5:0]					
			R	R	R	R	R			R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:22	Reserved	-	-	保留
21	TFQF	R	0	发送 FIFO/队列满 0：发送 FIFO 或队列未满 1：发送 FIFO 或队列已满
20:16	TFQPI[4:0]	R	5’b0	发送 FIFO/队列放入索引 发送 FIFO 或队列的放入索引指针，范围 0-31
15:13	Reserved	-	-	保留
12:8	TFGI[4:0]	R	5’b0	发送 FIFO 获取索引 发送 FIFO 的读取索引指针，范围 0-31。若已配置为发送队列模式（FDCAN_TXBC.TFQM=1），则该位读取值为 0
7:6	Reserved	-	-	保留
5:0	TFFL[5:0]	R	6’b0	发送 FIFO 空闲级别 从 TFGI 开始的连续空闲的发送 FIFO 元素数量，范围 0-32。若已配置为发送队列模式（FDCAN_TXBC.TFQM=1），则该位读取值为 0

注：

在专用发送缓冲区与发送 FIFO 或发送队列组合配置的情况下，放入索引和获取索引是指，从第一个专用发送缓冲区开始的发送缓冲区编号。

例如：对于有 12 个专用发送缓冲区和 20 个发送 FIFO 缓冲区的组合配置，放入索引 15 指向发送 FIFO 的第 4 个缓冲区。

31.5.35 FDCAN 发送缓冲区元素大小配置寄存器（FDCAN_TXESC）

地址偏移：0x0C8

复位值：0x0000 0000

配置发送缓冲区元素的数据字节数。大于 8 字节的数据段仅适用于 FD CAN 模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TBDS[2:0]		
													RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:3	Reserved	-	-	保留

Puya Confidential

2:0	TBDS[2:0]	RW	3’b0	发送缓冲区数据段大小 000： 8 字节 001： 12 字节 010： 16 字节 011： 20 字节 100： 24 字节 101： 32 字节 110： 48 字节 111： 64 字节

注：如果发送缓冲区元素的数据长度（由 DLC 配置），大于 TBDS 配置的数据长度，则发送缓冲区中未定义的数据字节将会以 0xCC（填充字节）发送。

31.5.36 FDCAN 发送缓冲区请求挂起寄存器（FDCAN_TXBRP）

地址偏移：0x0CC

复位值：0x0000 0000

配置发送缓冲区元素的数据字节数。大于 8 字节的数据段仅适用于 FD CAN 模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:0	TRP[31:0]	R	0	发送请求挂起标志 每个发送缓冲区都有自己的发送请求挂起位。这些位通过寄存器 FDCAN_TXBAR 置位；在请求的发送已完成，或请求的发送已通过寄存器 FDCAN_TXBCR 取消后复位。 只有 FDCAN_TXBC 配置了的发送缓冲区对应的 TXBRP 位才会置位。 TXBRP 位置位后会启动发送扫描（见章节【发送处理】），以检查优先级最高的挂起的发送请求（即消息 ID 最小的发送缓冲区）。 取消发送请求会复位相应的 TXBRP 位。如果请求取消时发送已经开始，则无论发送是否成功，在发送结束时 TXBRP 位都会被复位。TXBRP 位复位后，对应的取消请求位也会立即复位。 请求取消后，以下情况通过寄存器 FDCAN_TXBCF 指示取消已完成： • 发送成功及相应的 FDCAN_TXBTO 位置位后 • 取消时发送尚未开始 • 发送因仲裁失败而中止 • 帧发送期间发生错误 在 DAR 模式下，所有发送在失败后都会自动取消。所有失败的发送，对应的 FDCAN_TXBCF 位都会置位。 0：无发送请求挂起 1：发送请求挂起

Puya Confidential

				注： 在发送扫描期间置位的 TXBRP 位，在本次发送扫描期间被忽略，在此期间，如果请求取消这些发送，则发送被立即取消，对应的 TXBRP 位复位。

31.5.37 FDCAN 发送缓冲区添加请求寄存器（FDCAN_TXBAR）

地址偏移：0x0D0

复位值：0x0000 0000

配置发送缓冲区元素的数据字节数。大于 8 字节的数据段仅适用于 FD CAN 模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	AR[31:0]	RW	0	添加请求 每个发送缓冲区都有自己的添加请求位，写 1 置位相应的添加请求位，写 0 无影响。主机可一次设置多个发送缓冲区的发送请求。只有 FDCAN_TXBC 配置了的发送缓冲区对应的 TXBAR 位才会置位。没有运行发送扫描时，这些位立即复位，否则将保持置位到发送扫描完成。 0：未添加发送请求 1：已添加发送请求 注： 如果添加请求的发送缓冲区已有发送请求挂起（FDCAN_TXBRP 寄存器相应位已置位），该添加请求将被忽略。

31.5.38 FDCAN 发送缓冲区取消请求寄存器（FDCAN_TXBCR）

地址偏移：0x0D4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	CR[31:0]	RW	0	取消请求 每个发送缓冲区都有自己的取消请求位，写 1 置位相应的取消请求位，写 0 无影响。主机可一次设置多个发送缓冲区的取消请求。只有 FDCAN_TXBC 配置了的发送缓冲区对应的 TXBCR 位才会置位。这些位保持置位直到对应的 FDCAN_TXBRP 位复位。 0：无取消请求挂起 1：有取消请求挂起

Puya Confidential

31.5.39 FDCAN 发送缓冲区发送已发生寄存器（FDCAN_TXBTO）

地址偏移：0x0D8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:0	TO[31:0]	R	0	发送已发生 每个发送缓冲区都有自己的发送已发生标志位。当发送缓冲区发送完成， FDCAN_TXBRP 相应位清零时，对应的发送已发生标志位置位。向 FDCAN_TXBAR 相应位写 1 请求新的发送时，对应的发送已发生标志位复位。 0：未进行发送 1：已进行发送

31.5.40 FDCAN 发送缓冲区取消已完成寄存器（FDCAN_TXBCF）

地址偏移：0x0DC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:0	CF[31:0]	R	0	取消已完成 每个发送缓冲区都有自己的取消已完成标志位。当通过 FDCAN_TXBCR 请求的取消已完成，FDCAN_TXBRP 相应位清零时，对应的取消已完成标志位置位。如果 FDCAN_TXBRP 相应位在取消时没有置位，取消已完成标志位会立即置位。向 FDCAN_TXBAR 相应位写 1 请求新的发送时，对应的取消已完成标志位复位。 0：无发送缓冲区取消 1：发送缓冲区已取消

31.5.41 FDCAN 发送缓冲区发送中断使能寄存器（FDCAN_TXBTIE）

地址偏移：0x0E0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Puya Confidential

TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	TIE[31:0]	RW	0	发送中断使能 每个发送缓冲区都有自己的发送中断使能位。 0：禁止发送中断 1：使能发送中断

31.5.42 FDCAN 发送缓冲区取消已完成中断使能寄存器（FDCAN_TXBCIE）

地址偏移：0x0E4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:0	CFIE[31:0]	RW	0	取消已完成中断使能 每个发送缓冲区都有自己的取消已完成中断使能位。 0：禁止取消已完成中断 1：使能取消已完成中断

31.5.43 FDCAN 发送事件 FIFO 配置寄存器（FDCAN_TXEFC）

地址偏移：0x0F0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	EFWM[5:0]						Res.	Res.	EFS[5:0]					
		RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EFSA[15:2]														Res.	Res.
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Name	RW	Reset Value	Function
31:30	Reserved	-	-	保留
29:24	EFWM[5:0]	RW	6’b0	事件 FIFO 水位线 0：禁止水位线中断 1~32：发送事件 FIFO 水位线中断（FDCAN_IR.TEFW）的级别 ＞32：禁止水位线中断
23:22	Reserved	-	-	保留
21:16	EFS[5:0]	RW	6’b0	事件 FIFO 大小 0：禁止发送事件 FIFO 1-32：发送事件 FIFO 的元素数量 ＞32：大于 32 的设定值被解析为 32

Puya Confidential

				发送事件 FIFO 元素的索引为 0 到 EFS 减 1
15:2	EFSA[15:2]	RW	14’b0	事件 FIFO 起始地址 消息 RAM 中事件 FIFO 的起始地址（32 位字地址）
1:0	Reserved	-	-	保留

31.5.44 FDCAN 发送事件 FIFO 状态寄存器（FDCAN_TXEFS）

地址偏移：0x0F4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TEFL	EFF	Res.	Res.	Res.	EFPI[4:0]				
						R	R				R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	EFGI[4:0]					Res.	Res.	EFFL[5:0]					
			R	R	R	R	R			R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:26	Reserved	-	-	保留
25	TEFL	R	0	发送事件 FIFO 元素丢失 该位是中断标志 FDCAN_IR. TEFL 的副本。当 FDCAN_IR. TEFL 被复位时，该位也被复位 0：发送事件 FIFO 无元素丢失 1：发送事件 FIFO 有元素丢失；对空间大小为 0 的发送事件 FIFO 进行写操作时，该位也会置位
24	EFF	R	0	事件 FIFO 已满 0：发送事件 FIFO 未满 1：发送事件 FIFO 已满
23:21	Reserved	-	-	保留
20:16	EFPI[4:0]	R	5’b0	事件 FIFO 放入索引 发送事件 FIFO 写入索引指针，范围 0~31
15:13	Reserved	-	-	保留
12:8	EFGI[4:0]	R	5’b0	事件 FIFO 获取索引 发送事件 FIFO 读取索引指针，范围 0~31
7:6	Reserved	-	-	保留
5:0	EFFL[5:0]	R	6’b0	事件 FIFO 填充级别 接收 FIFO0 中存储的元素个数，范围 0~32

31.5.45 FDCAN 发送事件 FIFO 确认寄存器（FDCAN_TXEFA）

地址偏移：0x0F8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EFAI[4:0]				
											RW	RW	RW	RW	RW

Puya Confidential

Bit	Name	RW	Reset Value	Function
31:5	Reserved	-	-	保留
4:0	EFAI[4:0]	RW	5’b0	事件 FIFO 确认索引 主机从发送事件 FIFO 读取元素或元素序列后，必须将从事件 FIFO 中读取的最后一个元素的索引写入 EFAI。此操作会将发送事件 FIFO 获取索引 FDCAN_TXEFS.EFGI 设为 EFAI 加 1，并会更新事件 FIFO 填充级别 FDCAN_TXEFS.EFFL。

Puya Confidential

32. MCU 调试接口

32.1 DBGMCU 简介

DBGMCU 模块给调试者提供以下支持：

- 低功耗模式
- 定时器、看门狗、I²C 的时钟控制
- CAN 的调试功能
- DBGMCU 寄存器还提供芯片 ID 的编码。使用 JTAG 或者 SW 调试接口，或者用户程序都可以访问此 ID 编码。

32.2 DBGMCU 主要特性

- 支持睡眠模式，停止模式和待机模式的调试
- CPU 进入 HALT 时，控制定时器、看门狗停止计数或者继续计数
- CPU 进入 HALT 时，阻止 I²C 的 SMBUS 超时
- CPU 进入 HALT 时，使能 CAN 的“读取时禁止修改功能“

32.3 DBGMCU 功能描述

32.3.1 支持调试低功耗模式

用户可以通过执行 WFI 和 WFE 指令进入低功耗模式。

MCU 支持多种低功耗模式，分别可以关闭 CPU 时钟，或降低 CPU 的功耗。

内核不允许在调试期间关闭 FCLK 或 HCLK。这些时钟对于调试操作是必要的，因此在调试期间，它们必须工作。MCU 使用一种特殊的方式，允许用户在低功耗模式下调试代码。

为实现这一功能，调试器必须先设置一些配置寄存器来改变低功耗模式的特性。

- 在睡眠模式下，调试器必须先置位 DBG_SLEEP 位。这将为 HCLK 提供与 FCLK 相同的时钟。
- 在停止模式下，调试器必须先置位 DBG_STOP 位。这将激活 HSI 时钟，在停止模式下为 FCLK 和 HCLK 提供时钟。
- 在待机模式下，调试器必须先置位 DBG_STDBY 位。这将激活 HSI 时钟，在待机模式下为 FCLK 和 HCLK 提供时钟。

32.4 DBGMCU 寄存器描述

DBGMCU 寄存器映射在外部 PPB 总线。

32.4.1 DBGMCU ID 编码（DBGMCU_IDCODE）

该寄存器由上电复位，系统复位不会复位该寄存器。当内核处于复位状态下时，调试器可以访问该寄存器。

Puya Confidential

Address offset：0xE0042000

Reset value：0x4444_BBBB

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV_ID[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REV_ID[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	RW	Reset Value	Function
31:0	REV_ID[31:0]	R	32'h4444BBBB	MCU 产品 ID。该值保存在 FLASH information 区。

32.4.2 DBGMCU 调试配置寄存器 1（DBGMCU_CR1）

该寄存器由上电复位，系统复位不会复位该寄存器。当内核处于复位状态下时，调试器可以访问该寄存器。

Address offset：0xE0042004

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBG_TIM7_STOP	DBG_TIM6_STOP	Res	DBG_TIM8_STOP	DBG_I2C2_SMBUS_TIMEOUT
											RW	RW			RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBG_I2C1_SMBUS_TIMEOUT	DBG_CAN1_STOP	Res	DBG_TIM3_STOP	DBG_TIM2_STOP	DBG_TIM1_STOP	DBG_WWDG_STOP	DBG_IWDG_STOP	Res	Res	Res	Res	Res	DBG_STDBY	DBG_STOP	DBG_SLEEP
RW	RW		RW	RW	RW	RW	RW						RW	RW	RW

Bit	Name	RW	Reset Value	Function
31:21	Reserved	-	-	保留
20	DBG_TIM7_STOP	RW	0	当 CPU 核处于 halt 状态时，控制 TIM7 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数，输出禁止；
19	DBG_TIM6_STOP	RW	0	当 CPU 核处于 halt 状态时，控制 TIM6 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数，输出禁止；
18	Reserved	-	-	保留
17	DBG_TIM8_STOP	RW	0	当 CPU 核处于 halt 状态时，控制 TIM8 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数，输出禁止；
16	DBG_I2C2_SMBUS_TIMEOUT	RW	0	当 CPU 核处于 halt 状态时，控制 I2C2 SMBUS 超时模式停止。 0：CPU halt 时，I2C2 与正常模式相同； 1：CPU halt 时，I2C2 SMBUS 超时功能禁止；

Puya Confidential

15	DBG_I2C1_SMBUS_TIMEOUT	RW	0	当 CPU 核处于 halt 状态时，控制 I2C1 SMBUS 超时模式停止。 0：CPU halt 时，I2C1 与正常模式相同； 1：CPU halt 时，I2C1 SMBUS 超时功能禁止；
14	DBG_CAN1_STOP	RW	0	当 CPU 核处于 halt 状态时，启动 CAN1 模块的“读取时禁止修改功能”。 0：CPU halt 时，CAN1 与正常模式相同； 1：CPU halt 时，会启用 CAN1 的“读时禁止修改功能”。 读时禁止修改功能描述如下： ECR.CEL[7:0], PSR.PXE, PSR.RFDF, PSR.RBRS, PSR.RESI 不会被复位, PSR.LEC 和 PSR.DLEC 信号不会被置位。并且这些寄存器被重复读取后，也不会被复位或者置位。
13	Reserved	-	-	保留
12	DBG_TIM3_STOP	RW	0	当 CPU 核处于 halt 状态时，控制 TIM3 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数；
11	DBG_TIM2_STOP	RW	0	当 CPU 核处于 halt 状态时，控制 TIM2 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数；
10	DBG_TIM1_STOP	RW	0	当 CPU 核处于 halt 状态时，控制 TIM1 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数；
9	DBG_WWDG_STOP	RW	0	当 CPU 核处于 halt 状态时, 控制 WWDG 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数；
8	DBG_IWDG_STOP	RW	0	当 CPU 核处于 halt 状态时，控制 IWDG 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数；
7:3	Reserved	-	-	保留
2	DBG_STDBY	RW	0	调试待机模式。 0：（FCLK 关，HCLK 关）正常待机模式处理； 1：（FCLK 开，HCLK 开）系统时钟由 HSI 提供。如果产生唤醒条件，MCU 会产生系统复位退出待机模式。
1	DBG_STOP	RW	0	调试停止模式。 0：（FCLK 关，HCLK 关）在停止模式，HCLK 和 FCLK 都会关闭。当从停止模式退出时，时钟配置与复位后相同（系统时钟为 HSI）。随后，软件需要重新配置时钟控制器以使能 PLL 和 HSE 等。 1：（FCLK 开，HCLK 开）。当进入停止模式，系统内部时钟源不会关闭，FCLK 和 HCLK 存在。当

Puya Confidential

				退出停止模式，如果需要改变时钟控制，软件需要重新配置。
0	DBG_SLEEP	RW	0	调试睡眠模式。 0：（FCLK 开，HCLK 关）。在睡眠模式，FCLK 由原先配置好的系统时钟提供，HCLK 关闭。由于睡眠模式不会复位已配置好的时钟系统，因此从睡眠模式退出后，软件不需要重新配置时钟。 1：（FCLK 开，HCLK 开）。在睡眠模式，FCLK 和 HCLK 时钟都由原先配置好的系统时钟提供。

32.4.3 DBGMCU 调试配置寄存器 2（DBGMCU_CR2）

该寄存器由上电复位，系统复位不会复位该寄存器。当内核处于复位状态下时，调试器可以访问该寄存器。

Address offset：0xE0042008

Reset value：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBG_EPWM_EN	DBG_EPWM_CLK_OFF
														RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	DBG_LPTIM_STOP	Res	Res	Res	Res	Res	Res	Res	Res
							RW								

Bit	Name	RW	Reset Value	Function
31: 18	Reserved	-	-	保留
17	DBG_EPWM_EN	RW	0	断点操作（cpu halt）是否作为 EPWM 模块的跳闸与 trip 事件。 0：断点操作（cpu halt）不作为跳闸与 trip 事件； 1：断点操作（cpu halt）操作会作为跳闸与 trip 事件；
16	DBG_EPWM_CLK_OFF	RW	0	当 CPU 核处于 halt 状态时，控制 EPWM 计数停止。 0：CPU halt 时，EPWM 时钟使能，正常计数； 1：CPU halt 时，EPWM 时钟禁止，停止计数，输出禁止；
15:9	Reserved	-	-	保留
8	DBG_LPTIM_STOP	RW	0	当 CPU 核处于 halt 状态时，控制 LPTIM 计数停止。 0：CPU halt 时，计数器时钟使能，正常计数； 1：CPU halt 时，计数器时钟禁止，停止计数，输出禁止；
7:0	Reserved	-	-	保留

Puya Confidential

33. 版本历史

版本	日期	更新记录
V0.1	2026.04.30	预发布版本

Puya Confidential

Puya Semiconductor Co., Ltd.

声　　明

普冉半导体(上海)股份有限公司（以下简称：“Puya”）保留更改、纠正、增强、修改 Puya 产品和/或本文档的权利，恕不另行通知。用户可在下单前获取产品的最新相关信息。

Puya 产品是依据订单时的销售条款和条件进行销售的。

用户对 Puya 产品的选择和使用承担全责，同时若用于其自己或指定第三方产品上的，Puya 不提供服务支持且不对此类产品承担任何责任。

Puya 在此不授予任何知识产权的明示或暗示方式许可。

Puya 产品的转售，若其条款与此处规定不一致，Puya 对此类产品的任何保修承诺无效。

任何带有 Puya 或 Puya 标识的图形或字样是普冉的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代并替换先前版本中的信息。

普冉半导体(上海)股份有限公司 - 保留所有权利