

32 位微控制器

KF32F 系列 用户手册

目录

目录	2
图示列表.....	23
表格列表.....	28
1 文档说明.....	37
1.1 简介.....	37
1.2 寄存器操作缩写说明.....	37
1.3 寄存器地址说明.....	37
1.3.1 偏移地址.....	37
1.3.2 寄存器内容说明.....	37
1.4 寄存器操作说明.....	38
2 芯片介绍.....	39
2.1 芯片特征.....	39
2.2 系统概述.....	40
2.3 指令集.....	40
2.4 在线编程和调试.....	41
2.4.1 ISP 模式.....	41
2.4.2 DPI 模式.....	41
3 I/O 端口介绍 (GPIO)	42
3.1 概述.....	42
3.2 GPIO 结构框图	42
3.3 GPIO 相关寄存器	43
3.3.1 GPIOx_PIR 输入状态寄存器	43
3.3.2 GPIOx_POR 输出状态寄存器.....	44
3.3.3 GPIOx_PUR 上拉使能寄存器.....	44
3.3.4 GPIOx_PDR 下拉使能寄存器.....	44
3.3.5 GPIOx_PODR 开漏输出控制寄存器.....	45
3.3.6 GPIOx_PMOD 端口方向控制寄存器.....	45
3.3.7 GPIOx_OMOD 端口速度控制寄存器	45
3.3.8 GPIOx_LOCK 端口配置锁定寄存器.....	46
3.4 端口方向控制.....	46
3.5 引脚通用输入配置.....	46
3.6 引脚通用输出配置.....	46
3.7 引脚重映射功能配置.....	47
3.8 引脚上/下拉功能.....	47
3.9 开漏输出控制功能.....	47
3.10 工作模式掉电锁存功能.....	47
3.11 引脚锁定.....	47

4 引脚重映射 (GPIO remap)	48
4.1 概述.....	48
4.2 引脚重映射相关寄存器.....	48
4.2.1 GPIOx_RMPLPx 低 8 位引脚重映射控制寄存器.....	48
4.2.2 GPIOx_RMPH Px 高 8 位引脚重映射控制寄存器.....	49
5 振荡器 (OSC)	50
5.1 概述.....	50
5.2 振荡器结构框图.....	51
5.3 振荡器相关寄存器.....	52
5.3.1 OSC_CTL0 振荡器控制寄存器 0.....	52
5.3.2 OSC_CTL1 振荡器控制寄存器 1.....	54
5.3.3 OSC_INT 振荡器中断控制寄存器.....	55
5.3.4 OSC_CTL2 振荡器控制寄存器 2.....	56
5.3.5 OSC_HFOSCCAL0 高频振荡器校准寄存器 0	57
5.3.6 OSC_HFOSCCAL1 高频振荡器校准寄存器 1	58
5.3.7 PLL_CTL PLL 控制寄存器	59
5.4 内部高频振荡器 INTHF	60
5.5 内部低频振荡器 INTLF	60
5.6 外部高频振荡器 EXTHF	60
5.7 外部低频振荡器 EXTLF	61
5.8 PLL 使用.....	62
5.9 内部低功耗 4M 振荡器 LP4M.....	63
5.10 时钟同步.....	63
5.11 外部时钟故障检测.....	64
5.11.1 时钟故障保护检测.....	64
5.11.2 时钟故障保护处理.....	65
5.12 外设模块时钟使能位.....	65
5.13 中断.....	65
6 存储器 (memory)	67
6.1 概述.....	67
6.2 存储器空间映射.....	68
6.3 存储器访问说明.....	69
6.4 ROM 访问说明.....	69
6.5 低功耗 RAM.....	69
7 中断 (interrupt)	70
7.1 概述.....	70
7.2 中断向量表.....	70
7.3 中断相关寄存器.....	73
7.3.1 INT_CTL0 中断控制寄存器 0.....	74
7.3.2 INT_EIE0 中断使能寄存器 0.....	74
7.3.3 INT_EIE1 中断使能寄存器 1.....	75

7.3.4 INT_EIE2 中断使能寄存器 2	77
7.3.5 INT_EIF0 中断标志位寄存器 0	80
7.3.6 INT_EIF1 中断标志位寄存器 1	81
7.3.7 INT_EIF2 中断标志位寄存器 2	83
7.3.8 INT_IPx 中断优先级寄存器 (x=0~18)	85
7.3.9 INT_EINTMASK 外部中断屏蔽寄存器	90
7.3.10 INT_EINTRISE 外部中断上升沿选择寄存器	90
7.3.11 INT_EINTFALL 外部中断下降沿选择寄存器	90
7.3.12 INT_EINTF 外部中断中断标志位寄存器	90
7.3.13 INT_EINTSS0 外部中断源选择寄存器 0	91
7.3.14 INT_EINTSS1 外部中断源选择寄存器 1	93
7.3.15 INT_CTL1 中断控制寄存器 1	95
7.4 中断使用	96
7.5 内核中断类型	97
7.6 中断优先级	97
7.7 中断行为介绍	98
7.8 中断的进入处理	99
7.9 中断的返回	100
7.10 中断堆栈地址对齐方式	101
7.11 中断向量表重映射	101
7.12 中断延迟	101
7.13 外部中断控制	102
7.14 中断唤醒	104
7.15 中断堆栈指针使用	104
7.16 多堆栈使用	104
7.17 芯片自动初始化	105
7.18 超级用户模式进入	105
7.19 中断使用步骤	105
8 系统控制 (system control)	107
8.1 概述	107
8.2 系统控制相关寄存器	108
8.2.1 SYS_PSW 程序状态字寄存器	108
8.2.2 SYS_MCTL 系统模式控制寄存器	108
8.2.3 SYS_ARCTL 应用和复位控制寄存器	109
8.2.4 SYS_VECTOFF 中断向量表重映射控制寄存器	110
8.2.5 SYS_RAMSPA RAM 空间指示寄存器	110
8.2.6 SYS_MEMCTL 程序空间控制寄存器	110
8.3 中断堆栈地址对齐	111
8.4 堆栈指针的使用	112
8.5 应用和复位控制	112
8.6 中断向量表重映射	112
8.7 RAM 空间指示	112
8.8 休眠与休眠唤醒	113

8.9 程序空间控制设置.....	114
8.10 超级用户模式使用.....	114
8.10.1 超级用户的进入和退出.....	114
8.10.2 超级用户权限.....	114
9 DMA 控制器 (DMA0/DMA1)	116
9.1 概述.....	116
9.2 DMA 相关寄存器	117
9.2.1 DMAx_CTLRy DMAx 通道 y 控制寄存器	118
9.2.2 DMAx_PADDRy DMAx 通道 y 外设地址寄存器	119
9.2.3 DMAx_MADDRy DMAx 通道 y 存储器地址寄存器.....	119
9.2.4 DMAx_CPARy DMAx 通道 y 当前外设地址寄存器.....	120
9.2.5 DMAx_CMARy DMAx 通道 y 当前存储器地址寄存器.....	120
9.2.6 DMAx_NCTy DMAx 通道 y 当前剩余数据寄存器.....	120
9.2.7 DMAx_LIFR DMAx 通道 1~7 中断标志寄存器.....	120
9.2.8 DMAx_LIER DMAx1~7 通道中断使能寄存器	121
9.3 DMA 功能	122
9.4 DMA 请求	123
9.5 DMA 触发	125
9.6 数据位宽.....	125
9.7 传输地址.....	125
9.8 数据量和数据空间.....	125
9.9 单数据/块传输和循环模式.....	126
9.10 优先级.....	126
9.11 中断.....	126
9.12 DMA 独立于 CPU 运行.....	127
9.13 DMA 与 CPU 冲突仲裁.....	127
9.14 使用 DMA	127
10 节拍定时器 (SYSTICK)	129
10.1 概述.....	129
10.2 节拍定时器结构框图.....	129
10.3 相关寄存器.....	129
10.3.1 ST_CTL 系统节拍定时器控制寄存器.....	130
10.3.2 ST_RELOAD 系统节拍定时器重加载寄存器	130
10.3.3 ST_CV 系统节拍定时器计数器.....	130
10.3.4 ST_CALI 校验寄存器.....	131
10.4 节拍定时器使用.....	131
10.5 中断请求.....	131
11 基本定时/计数器(T14/T15)	133
11.1 概述.....	133
11.2 基本定时器结构框图.....	133
11.3 基本定时器相关寄存器.....	134

11.3.1 Tx_CNT 定时器寄存器.....	134
11.3.2 Tx_CTL1 Tx 控制寄存器 1.....	134
11.3.3 Tx_CTL2 Tx 控制寄存器 2.....	135
11.3.4 Tx_PRSC Tx 预分频寄存器	137
11.3.5 Tx_PPX 周期寄存器	137
11.3.6 Tx_DIER Tx 中断使能控制寄存器.....	137
11.3.7 Tx_SR Tx 中断状态寄存器	138
11.3.8 Tx_SRIC Tx 中断标志清除寄存器	139
11.4 Tx 计数方式和中断	140
11.5 Tx 工作模式	141
11.5.1 定时模式.....	141
11.5.2 计数模式.....	141
11.6 定时器外部触发功能.....	141
11.6.1 触发模式.....	141
11.6.2 门控模式.....	142
11.6.3 复位模式.....	142
11.7 定时器主从同步.....	142
11.7.1 定时器之间主从关系	142
11.7.2 使用一个定时器作为另一个定时器的预分频器（计数模式 2）	142
11.7.3 使用 TxCK 使能 Tx 定时器（门控模式）	142
11.7.4 使用一个定时器启动另一个定时器（触发模式）	143
11.7.5 使用一个外部触发同步的启动两个定时器(同步触发)	143
11.7.6 使用一个定时器复位另一个定时器（复位模式）	143
11.8 Tx 更新	143
11.9 Tx 单脉冲模式	144
11.10 Tx 定时器关断	144
11.11 Tx 触发 DMA(基本定时器)	144
12 通用定时/计数器(T0/T1/T2/T3/T4/T18/T19/T20/T21/T22/T23).....	145
12.1 概述.....	145
12.2 通用定时器结构框图.....	145
12.3 通用定时器相关寄存器.....	146
12.3.1 Tx_CNT 定时器寄存器.....	146
12.3.2 Tx_CTL1 Tx 控制寄存器 1.....	147
12.3.3 Tx_CTL2 Tx 控制寄存器 2.....	148
12.3.4 Tx_PRSC Tx 预分频器	150
12.3.5 Tx_PPX 周期寄存器	150
12.3.6 Tx_UDTIM 更新计数器	151
12.4 Tx 计数方式和中断	152
12.5 Tx 工作模式	153
12.5.1 定时模式.....	153
12.5.2 计数模式.....	153
12.6 定时器外部触发功能.....	153
12.6.1 触发模式.....	153

12.6.2 门控模式.....	154
12.6.3 复位模式.....	154
12.7 定时器主从同步.....	154
12.7.1 定时器之间主从关系.....	154
12.7.2 使用一个定时器作为另一个定时器的预分频器（计数模式 2）.....	154
12.7.3 使用一个定时器使能另一个定时器（门控模式）.....	154
12.7.4 使用一个定时器启动另一个定时器（触发模式）.....	155
12.7.5 使用一个外部触发同步的启动两个定时器（同步触发）.....	155
12.7.6 使用一个定时器复位另一个定时器（复位模式）.....	155
12.8 Tx 更新.....	155
12.9 Tx 定时器关断.....	158
12.10 Tx 分配给 CCPx.....	158
12.11 T0 低功耗定时器（LP_TIMER）.....	158
12.12 Tx 触发 DMA(通用定时器).....	159
13 高级定时/计数器（T5/T6/T9/T10）.....	160
13.1 概述.....	160
13.2 高级定时器结构框图.....	160
13.3 高级定时器相关寄存器.....	161
13.3.1 Tx_CNT 定时器寄存器.....	161
13.3.2 Tz_CNT 定时器寄存器.....	162
13.3.3 Tx_PPX 周期寄存器.....	162
13.3.4 Tz_PPZ 周期寄存器.....	162
13.3.5 Tx_PRSC/Tz_PRSC/Tz 预分频寄存器.....	163
13.3.6 Tx_CCR0/Tx_CCR1/Tz_CCR0 Tx/Tz 触发 AD 寄存器.....	163
13.3.7 Tx_CTL Tx 控制寄存器.....	164
13.3.8 Tz_CTL Tz 控制寄存器.....	165
13.3.9 Tx_CCTCTL 比较器清零定时器 Tx 控制寄存器.....	167
13.3.10 Tz_CCTCTL 比较器清零定时器 Tz 控制寄存器.....	167
13.3.11 Tx_CCRCTL CCR 功能控制寄存器.....	168
13.4 Tx/Tz 的工作原理.....	169
13.5 Tx/Tz 计数方式和中断.....	169
13.6 Tx/Tz 工作模式.....	171
13.6.1 定时模式.....	171
13.6.2 计数模式.....	171
13.6.3 外部时钟引脚 Tx/Tz 说明.....	171
13.7 Tx/Tz 分频器.....	172
13.8 高级定时器中断.....	172
13.9 Tx/Tz 外部触发功能.....	172
13.9.1 触发模式.....	172
13.9.2 门控模式.....	173
13.9.3 复位模式.....	173
13.10 定时器主从同步.....	173
13.10.1 定时器之间主从关系.....	173

13. 10.2 使用一个定时器作为另一个定时器的预分频器（计数模式 2）	173
13. 10.3 使用一个定时器使能另一个定时器（门控模式）	173
13. 10.4 使用一个定时器启动另一个定时器（触发模式）	174
13. 10.5 使用一个外部触发同步的启动两个定时器（同步触发）	174
13. 10.6 使用一个定时器复位另一个定时器（复位模式）	174
13. 11 联立 Tx 和 Tz	174
13. 12 Tx/Tz 更新	175
13. 13 Tx/Tz 定时器关断	175
13. 14 Tx/Tz 分配给 ECCP	175
13. 15 Tx/Tz 触发 AD	175
13. 16 Tx/Tz 触发 DMA(高级定时器)	176
13. 17 比较器清零定时器 Tx/Tz 功能	176
14 通用捕捉/比较/PWM 模块（CCP0/1/2/3/4/18/19/20/21/22/23）	177
14. 1 概述	177
14. 2 CCP 模块结构框图	178
14. 3 CCP 模块相关寄存器	179
14. 3.1 CCPx_Cy 捕捉寄存器	180
14. 3.2 CCPx_SRIC CCPx 中断标志清除寄存器	181
14. 3.3 CCPx_DF CCP 触发 DMA 请求标志寄存器	181
14. 3.4 CCPx_CTL1 CCP 控制寄存器 1	182
14. 3.5 CCPx_Ry 比较/PWM 占空比寄存器	183
14. 3.6 CCPx_CTL2 CCP 控制寄存器 2	183
14. 3.7 CCPx_CTL3 CCPx 控制寄存器 3	184
14. 3.8 CCPx_EGIF CCPx 中断状态/事件产生寄存器	186
14. 4 捕捉模式	188
14. 4.1 PWM 测量模式	188
14. 4.2 模块相关引脚说明	189
14. 4.3 PWM 异或输入以及霍尔传感器接口功能	189
14. 4.4 比较器的输出触发捕捉	190
14. 4.5 捕捉中断与 DMA 请求	191
14. 5 比较模式	191
14. 5.1 比较功能	191
14. 5.2 比较中断与 DMA 请求	192
14. 6 PWM 模式	192
14. 6.1 PWMx 的周期	192
14. 6.2 PWMx 占空比	193
14. 6.3 PWMx 分辨率	193
14. 6.4 边沿对齐 PWM 信号	193
14. 6.5 中心对齐 PWM 信号	194
14. 6.6 单脉冲输出模式	195
14. 6.7 PWM 信号的产生和中断	195
14. 7 比较器的输出清除输出比较/PWM 信号	196
14. 8 低功耗 CCP（CCP0）	196

15 增强型捕捉/比较/PWM 模块(ECCP5/9).....	197
15.1 概述.....	197
15.2 ECCP 模块结构框图.....	198
15.3 ECCP 模块相关寄存器.....	199
15.3.1 ECCPx_CTL1 ECCPx 控制寄存器 1	200
15.3.2 ECCPx_Ry 比较/PWM 占空比寄存器.....	202
15.3.3 ECCPx_PXUDCTL 更新控制寄存器.....	202
15.3.4 ECCPx_CTL2 ECCP 控制寄存器 2	203
15.3.5 ECCPx_PXDTCTL 死区控制寄存器.....	204
15.3.6 ECCPx_PWMXOC 输出控制寄存器.....	205
15.3.7 ECCPx_PXATRCTL 输出配置寄存器.....	206
15.3.8 ECCPx_PXASCTL0 关断控制寄存器 0.....	207
15.3.9 ECCPx_PXASCTL1 关断控制寄存器 1	208
15.3.10 ECCPx_IE ECCPx 中断使能寄存器	209
15.3.11 ECCPx_EGIF ECCPx 中断状态/事件产生寄存器.....	210
15.3.12 Tx_UDTIMX/Tz_UDTIMZ 更新计数器.....	212
15.3.13 ECCPx_DF 触发 DMA 请求标志寄存器.....	212
15.3.14 ECCPx_Cy ECCPx 捕捉寄存器	213
15.3.15 ECCPx_DE ECCPx 请求 DMA 使能寄存器	213
15.3.16 ECCPx_SRIC ECCPx 中断标志清除寄存器	214
15.3.17 ECCPx_CTL3 ECCP 控制寄存器 3	215
15.3.18 ECCPx_PDCTL ECCP 位置检测控制寄存器.....	216
15.3.19 ECCPx_PXASCTL 关断控制寄存器	216
15.4 捕捉模式.....	218
15.4.1 PWM 测量模式	218
15.4.2 模块相关引脚说明.....	219
15.4.3 与霍尔传感器的接口	219
15.4.4 捕捉中断与 DMA 请求.....	219
15.5 比较模式.....	219
15.5.1 比较功能.....	219
15.5.2 比较中断与 DMA 请求.....	220
15.6 PWM 模式.....	220
15.6.1 PWMx 的周期、占空比及分辨率	222
15.6.1.1 PWMx 周期	222
15.6.1.2 PWMx 占空比	222
15.6.1.3 PWMx 分辨率	223
15.6.2 边沿对齐 PWM 信号	223
15.6.3 中心对齐 PWM 信号	224
15.6.4 PWM 信号产生和中断	225
15.6.5 独立输出模式.....	225
15.6.6 互补输出模式.....	226
15.6.7 单脉冲输出模式.....	227
15.6.8 死区延时.....	227
15.6.9 PWMx 输出控制	229

15.6.10 PWM 更新锁定	229
15.6.11 自动 PWM 相移	229
15.6.12 自动关断模式	230
15.6.13 自动重启模式	231
15.6.14 位置检测	232
15.6.14.1 位置检测输入信号	232
15.6.14.2 输入信号采样功能	232
15.6.14.3 位置检测事件中断标志位	232
16 正交编码脉冲电路 (QEIO/I)	234
16.1 概述	234
16.2 正交编码器模块结构框图	235
16.3 QEI 模块相关寄存器	236
16.3.1 QEIy_CTL0 QEIy 控制寄存器	236
16.3.2 QEIy_CTL1 QEIy 数字滤波控制寄存器	237
16.3.3 Tx_CNT Tx 寄存器	238
16.3.4 Tx_PPX 周期寄存器	238
16.3.5 Tx_PRSTx 预分频器	238
16.3.6 QEIy_DIER QEIy 中断控制寄存器	239
16.4 可编程数字噪声滤波器	240
16.5 正交编码器	240
16.5.1 计数方向状态	241
16.5.2 编码器计数方向	241
16.5.3 正交速率	241
16.6 16 位递增/递减位置计数器	241
16.6.1 使用 PPX 复位位置计数器 TX	242
16.6.2 使用索引脉冲复位位置计数器 Tx	242
16.6.3 相位检测	243
16.6.4 计数检测	243
16.7 QEI 作为 16 位定时/计数器	243
16.8 16 位预分频器	244
16.9 QEI 的使用说明	244
17 模数转换模块 (A/D)	245
17.1 概述	245
17.2 模数(A/D)转换模块原理	246
17.3 ADCx 相关寄存器	247
17.3.1 ADCx_CTL0 ADC 控制寄存器 0	247
17.3.2 ADCx_CTL1 ADCx 控制寄存器 1	249
17.3.3 ADCx_SCANSQ0 常规通道扫描序列设置寄存器	254
17.3.4 ADCx_SCANSQ1 常规通道扫描序列设置寄存器	254
17.3.5 ADCx_SCANSQ2 常规通道扫描序列设置寄存器	255
17.3.6 ADCx_SCANSQ3 常规通道扫描序列设置寄存器	255
17.3.7 ADCx_HSCANSQ 高优先级通道扫描序列设置寄存器	256

17.3.8 ADCx_WDH 模拟看门狗高阈值寄存器	257
17.3.9 ADCx_WDL 模拟看门狗低阈值寄存器	257
17.3.10 ADCx_DATA 常规通道数据寄存器	257
17.3.11 ADCx_HPDATAy 高优先级通道数据寄存器	257
17.3.12 ADCx_HPDOFFy 高优先级通道数据失调寄存器	258
17.3.13 ADCx_STATE ADC 状态寄存器	258
17.3.14 ADC0_DELAY ADC 快速交叉延时寄存器	260
17.4 ADC 电压引脚	262
17.5 ADC 时钟和时序图	262
17.6 ADC 通道	262
17.7 ADC 结果	263
17.8 高优先级通道管理	263
17.9 间隔模式	263
17.10 ADC 数据对齐	264
17.11 ADC 单次转换模式	265
17.12 ADC 连续转换模式	265
17.13 ADC 扫描模式	265
17.14 ADC 模拟看门狗	265
17.15 ADC 外部触发	265
17.16 DMA 请求	266
17.17 内部参考电压	266
17.18 内部信号通道	267
17.19 ADC 中断	267
17.20 双 ADC 模式	267
17.20.1 高优先级通道同步模式	268
17.20.2 常规通道同步模式	268
17.20.3 快速交叉模式	269
17.20.4 交替触发模式	269
17.20.5 混合(常规通道同步+高优先级通道同步模式)	270
17.20.6 混合(常规通道同步+交替触发模式)	270
17.20.7 混合(高优先级通道同步+交叉模式)	271
18 数模转换器模块 (D/A)	272
18.1 概述	272
18.2 数模 (D/A) 转换器模块原理框图	272
18.3 DACx 相关寄存器	273
18.3.1 DACx_CTL DACx 控制寄存器	273
18.3.2 DACx_DAHD DACx 数据保持寄存器	275
18.3.3 DACx_DATA DACx 输出数据寄存器	275
18.3.4 DACx_CTL1 DACx 控制寄存器 1	275
18.3.5 DACx_CAL DACx 校验寄存器	276
18.4 DACxREF 参考电压	277
18.5 DACx 转换	277
18.6 DACx 触发事件选择	277

18.7 DMA 请求	278
18.8 噪声发生器	278
18.9 三角波发生器	279
18.10 DACx 校准功能	280
19 模拟比较器模块 (CMP)	282
19.1 概述	282
19.2 模拟比较器原理	282
19.3 模拟比较器相关的寄存器	285
19.3.1 CMP_CTLx 比较器控制寄存器 0/1/2	285
19.3.2 CMP_CTL3 比较器控制寄存器 3	286
19.3.3 CMP_CTL4 比较器控制寄存器 4	288
19.4 比较器引脚设置	291
19.5 输出极性选择	291
19.6 范围控制	291
19.7 数字滤波器	291
19.7.1 数字滤波器框图	292
19.7.2 数字滤波器时序	292
19.7.3 数字滤波器使用方法	292
19.8 比较器中断	293
19.9 比较器输出关断 PWM 和清零定时器	293
19.10 BEMF 模式和 HALL 模式	293
20 通用全/半双工收发器 (USART)	295
20.1 概述	295
20.1.1 USART 引脚说明	295
20.2 USART 相关寄存器	297
20.2.1 USARTx_CTLR USART 控制寄存器	297
20.2.2 USARTx_BRGR USART 波特率寄存器	300
20.2.3 USARTx_STR USART 状态寄存器	300
20.2.4 USARTx_TBUFR USART 发送数据寄存器	302
20.2.5 USARTx_RBUFR USART 接收数据寄存器	302
20.2.6 USARTx_U7816R 7816 控制寄存器	302
20.2.7 USARTx_IER USART 中断使能寄存器	304
20.2.8 USARTx_ADM USART 地址匹配设置寄存器	305
20.3 功能描述	306
20.3.1 波特率发生器	306
20.3.1.1 波特率产生	306
20.3.1.2 波特率误差分析	306
20.3.1.3 自动波特率检测	308
20.3.2 间隔字符	308
20.3.2.1 间隔符时序	308
20.3.2.2 间隔和同步发送序列	308
20.3.3 USART 全双工模式	309

20.3.3.1	发送数据	309
20.3.3.2	全双工发送的设置	310
20.3.3.3	接收数据	311
20.3.3.4	接收错误	311
20.3.3.5	地址检测	311
20.3.3.6	全双工接收的设置	311
20.3.3.7	RS-485 发送/接收	311
20.3.4	USART 半双工模式	311
20.3.4.1	USART 半双工主控模式	312
20.3.4.2	半双工主控发送	312
20.3.4.3	半双工主控接收	314
20.3.4.4	USART 半双工从动模式	315
20.3.4.5	USART 半双工从动发送	315
20.3.4.6	USART 半双工从动接收	316
20.3.4.7	USARTA 半双工 RS-485 模式	316
20.3.5	硬件流控制	316
20.3.6	IrDA 红外调制	316
20.3.7	7816 模式	318
20.3.7.1	7816 模式发送	318
20.3.7.2	7816 模式接收	319
20.3.8	USART 单线通信模式	320
20.3.8.1	单线发送模式	321
20.3.8.2	单线接收模式	321
20.3.9	USART 的 DMA 读写	321
20.3.10	USART 中断	322
20.3.11	低功耗 USART (LP-USART)	322
21	串行外设接口 (SPI)	324
21.1	概述	324
21.1.1	SPI 功能描述	324
21.1.2	I2S 功能描述	325
21.2	相关寄存器	326
21.2.1	SPIx_BRGR SPI 波特率寄存器	326
21.2.2	SPIx_CTLR SPI 控制寄存器	327
21.2.3	SPIx_BUFR SPI 数据寄存器	328
21.2.4	SPIx_STR SPI 状态寄存器	329
21.3	SPI 操作	331
21.3.1	SPI 工作原理	331
21.3.2	使能 SPI/IO 与外部链接	331
21.3.2.1	使能 SPI/IO	331
21.3.2.2	典型连接	331
21.3.3	主模式	332
21.3.4	从模式	333
21.3.4.1	从动模式	333

21.3.4.2 从动选择同步	334
21.3.5 休眠模式和复位	334
21.3.5.1 休眠模式下的工作	334
21.3.5.2 复位的影响	334
21.3.6 SPI 工作模式设置	334
21.3.6.1 主控工作流程	334
21.3.6.2 从动工作流程	335
21.4 I2S 操作	335
21.4.1 I2S 飞利浦标准	335
21.4.2 MSB 对齐标准	336
21.4.3 LSB 对齐标准	336
21.4.4 PCM 标准	336
21.4.5 时钟发生器	337
21.4.6 状态标志位	337
21.4.6.1 忙标志位(SPIBUSY)	338
21.4.6.2 发送缓存非空标志位(TNE)	338
21.4.6.3 接收缓存非空标志位(RNE)	338
21.4.6.4 声道标志位(CHSIDE)	338
21.4.7 错误标志位	338
21.5 SPI 的 DMA 读写	339
21.6 SPI 中断	339
22 内部集成电路接口 (I2C)	340
22.1 概述	340
22.2 I2C 模块结构框图	341
22.2.1 I2C 引脚说明	341
22.3 I2C 模块相关寄存器	342
22.3.1 I2Cx_CTLR I2C 控制寄存器	342
22.3.2 I2Cx_SR I2C 状态寄存器	344
22.3.3 I2Cx_BUFR I2C 缓存寄存器	346
22.3.4 I2Cx_ADDRy I2C 地址寄存器 (y=0,1,2,3)	346
22.3.5 I2Cx_BRGR I2C 波特率寄存器	346
22.3.6 I2Cx_IER I2C 中断使能寄存器	347
22.4 功能描述	348
22.4.1 从动模式	348
22.4.1.1 寻址	348
22.4.1.2 接收	349
22.4.1.3 发送	350
22.4.1.4 广播呼叫地址支持	351
22.4.2 主控模式	352
22.4.2.1 主控模式进入	352
22.4.2.2 I2C 主模式操作	353
22.4.2.3 波特率发生器	354
22.4.2.4 I2C 主控模式启动条件时序	354

22.4.2.5 I2C 主控模式重复启动条件时序	355
22.4.2.6 I2C 主控模式发送	356
22.4.2.7 I2C 主控模式接收	356
22.4.2.8 停止条件序列	357
22.4.3 时钟仲裁	358
22.4.4 多主控器模式	358
22.4.4.1 多主机通信, 总线冲突与总线仲裁	359
22.4.5 I2C 屏蔽寄存器	359
22.4.6 监控模式	360
22.4.7 SMBus	360
22.4.7.1 介绍	360
22.4.7.2 SMBus 应用用途	361
22.4.7.3 SMBus 提醒模式	361
22.4.7.4 如何使用 SMBus 模式的接口	362
22.4.8 I2C 的 DMA 读写	362
22.4.9 I2C 中断	362
23 EXIC 接口	364
23.1 概述	364
23.2 EXIC 相关寄存器	365
23.2.1 EXIC_CTL0 EXIC 控制寄存器 0	365
23.2.2 EXIC_CTL1 EXIC 控制寄存器 1	366
23.2.3 EXIC_STAT 状态寄存器	367
23.2.4 EXIC_WRITEBUF 写数据缓存寄存器	367
23.2.5 EXIC_READBUF 读数据缓存寄存器	368
23.3 功能描述	369
23.3.1 写功能	369
23.3.2 读功能	369
23.3.3 EXIC 接口 DMA 使用方法	370
23.3.4 EXIC 接口使用方法	370
24 实时时钟 (RTC)	371
24.1 概述	371
24.2 RTC 相关寄存器	373
24.2.1 RTC_CR 实时时钟控制寄存器	373
24.2.2 RTC_ALRA 实时时钟闹钟 A 控制寄存器	375
24.2.3 RTC_ALRB 实时时钟闹钟 B 控制寄存器	375
24.2.4 RTC_TMR 实时时钟时间寄存器	376
24.2.5 RTC_DTR 实时时钟日期寄存器	377
24.2.6 RTC_TMER 实时时钟定时器寄存器	377
24.2.7 RTC_TCR 实时时钟定时器控制寄存器	377
24.2.8 RTC_IER 实时时钟中断使能寄存器	378
24.2.9 RTC_IFR 实时时钟中断标志寄存器	379
24.2.10 RTC_TMBR 实时时钟时间备份寄存器	381

24. 2. 11 RTC_DTBR 实时时钟日期备份寄存器.....	381
24. 3 RTC 模块的复位和寄存器读写	382
24. 4 功能描述.....	382
24. 4. 1 时钟校正.....	382
24. 4. 1. 1 计算方法.....	382
24. 4. 1. 2 校准原理.....	383
24. 4. 1. 3 校准寄存器设定值表.....	384
24. 4. 2 闹钟.....	384
24. 4. 3 时间戳.....	385
24. 4. 4 周期时间节拍中断.....	385
24. 4. 5 进程中断.....	386
24. 4. 6 实时时钟定时器.....	386
24. 4. 6. 1 实时时钟定时器的使能.....	386
24. 4. 6. 2 实时时钟定时器的中断.....	386
24. 4. 6. 3 实时时钟定时器的计数和重载.....	386
24. 4. 6. 4 实时时钟定时器的级联.....	387
24. 4. 6. 5 实时时钟定时器的配置步骤.....	387
24. 4. 7 复位.....	387
25 控制器局域网总线 (CAN0/1/2)	388
25. 1 概要.....	388
25. 2 CAN 模块结构框图	389
25. 3 CAN 模块相关寄存器	390
25. 3. 1 CANx_CTLR CAN 控制寄存器.....	391
25. 3. 2 CANx_BRGR CAN 波特率寄存器	393
25. 3. 3 CANx_RCR CAN 记录寄存器.....	394
25. 3. 4 CANx_EROR CAN 错误寄存器	396
25. 3. 5 CAN 验收滤波器组	396
25. 3. 5. 1 CANx_ACRR CAN 验收代码寄存器	397
25. 3. 5. 2 CANx_MSKR CAN 验收屏蔽寄存器.....	397
25. 3. 5. 3 验收滤波器.....	397
25. 3. 6 CAN 发送缓冲器	398
25. 3. 6. 1 CANx_INFR CAN 数据信息寄存器.....	398
25. 3. 6. 2 CANx_TX0R CAN 数据寄存器 0.....	398
25. 3. 6. 3 CANx_TX1R CAN 数据寄存器 1	399
25. 3. 6. 4 CANx_TX2R CAN 数据寄存器 2.....	399
25. 3. 7 CAN 接收缓冲寄存器组	399
25. 3. 7. 1 标准帧格式 SFF.....	399
25. 3. 7. 2 扩展帧格式 EFF.....	401
25. 3. 8 RAM 接收.....	402
25. 3. 9 CANx_IER CAN 中断使能寄存器.....	404
25. 3. 10 CANx_IFR CAN 中断标志寄存器.....	406
25. 4 CAN 模块功能描述	408
25. 4. 1 模块功能.....	408

25.4.1.1 CAN 系统拓扑结构	409
25.4.1.2 初始化.....	409
25.4.1.3 消息报文.....	409
25.4.1.4 回环模式.....	409
25.4.1.5 安静模式.....	410
25.4.1.6 睡眠模式.....	410
25.4.1.7 信息发送.....	410
25.4.1.8 信息接收.....	410
25.4.1.9 CAN 标志位说明	411
25.4.1.10 DMA 请求	411
25.4.1.11 错误处理.....	412
25.4.1.12 低功耗 CAN	413
25.5 CAN 的使用	413
26 独立看门狗 (IWDG)	415
26.1 概述.....	415
26.2 独立看门狗相关寄存器.....	415
26.2.1 IWDG_CTL 独立看门狗控制寄存器.....	415
26.2.2 IWDG_FD 独立看门狗喂狗寄存器	416
26.3 独立看门狗原理.....	417
26.3.1 独立看门狗的计数时钟	417
26.3.2 独立看门狗的喂狗方式.....	417
26.3.3 独立看门狗的复位周期.....	417
26.4 IWDG 默认使能功能	417
26.5 独立看门狗的使用注意事项.....	417
27 窗口看门狗 (WWDG)	419
27.1 概述.....	419
27.2 窗口看门狗相关寄存器.....	420
27.2.1 WWDG_CTL 窗口看门狗控制寄存器.....	420
27.2.2 WWDG_CNT 窗口看门狗计数寄存器	421
27.3 窗口看门狗原理.....	422
27.4 窗口看门狗中断.....	422
27.5 时钟选择位.....	423
28 USB 模块 (USB)	424
28.1 概述.....	424
28.2 USB 模块结构框图.....	424
28.3 USB 模块相关寄存器.....	425
28.3.1 USB_UCON USB 控制寄存器.....	425
28.3.2 USB_USTAT USB 状态寄存器	427
28.3.3 USB_UADDR USB 地址寄存器	428
28.3.4 USB_UFRM USB 帧编号寄存器	428
28.3.5 USB_UIR USB 中断标志寄存器	429

28.3.6 USB_UIE USB 中断控制寄存器.....	430
28.3.7 USB_UEPn USB 端点 n 控制寄存器.....	431
28.3.8 USB_UDB USB DEBUG 控制寄存器.....	432
28.3.9 USB_BDnSTAT 缓存区描述符 n 的状态寄存器.....	433
28.4 USB RAM.....	435
28.5 缓冲区描述符和缓冲区描述符表 BDT.....	435
28.5.1 BD 状态和配置.....	436
28.5.2 BD 字节计数.....	436
28.5.3 BD 地址验证.....	437
28.5.4 双缓冲（乒乓缓冲）.....	437
28.6 USB 中断.....	438
28.7 SB 时钟源.....	439
28.8 USB 操作概述.....	439
28.8.1 分层框架.....	440
28.8.2 帧.....	440
28.8.3 传输.....	440
28.8.4 枚举.....	440
28.8.5 描述符.....	441
28.8.5.1 设备描述符.....	441
28.8.5.2 配置描述符.....	441
28.8.5.3 接口描述符.....	441
28.8.5.4 端点描述符.....	441
28.8.5.5 字符串描述符.....	441
28.8.6 总线速度.....	441
28.8.7 分类规范和驱动程序速度.....	441
28.9 USB 端点配置概述.....	442
29 CFGL 模块 (CFGL)	443
29.1 概述.....	443
29.2 CFGL 模块结构框图.....	443
29.3 CFGL 相关寄存器.....	444
29.3.1 CFGL1_CTL0 CFGL1 控制寄存器 0.....	444
29.3.2 CFGL2_CTL0 CFGL2 控制寄存器 0.....	446
29.3.3 CFGL1_CTL1 CFGL1 控制寄存器 1.....	448
29.3.4 CFGL2_CTL1 CFGL2 控制寄存器 1.....	451
29.3.5 CFGLIFR CFGL 中断控制寄存器.....	453
29.4 CFGL 模块的配置.....	455
29.4.1 CFGL 数据输入信号.....	455
29.4.2 CFGL 数据门控.....	456
29.4.3 CFGL 逻辑功能.....	456
29.4.4 CFGL 输出同步.....	457
29.4.5 CFGL 输出极性.....	457
29.5 CFGL 中断.....	457
29.6 CFGL 输出.....	458

29.7 CFGL 工作在休眠模式.....	458
30 复位 (RESET)	459
30.1 复位相关寄存器.....	460
30.1.1 RST_CTL0 复位控制寄存器 0.....	460
30.1.2 RST_CTL1 复位控制寄存器 1.....	460
30.1.3 RST_CTL2 复位控制寄存器 2.....	461
30.1.4 RST_CTL3 复位控制寄存器 3.....	461
30.2 上电/掉电复位 POR/PDR.....	462
30.3 BrownOut 复位 BOR.....	462
30.4 IWDT 复位	462
30.5 WWDT 复位.....	463
30.6 NRST 复位	463
30.7 电压检测模块 (PVD)	463
30.8 备份域复位.....	464
31 外设模块时钟使能模块(CLK_EN).....	465
31.1 软件复位.....	465
31.2 概述.....	465
31.3 外设模块时钟使能模块相关寄存器	466
31.3.1 PCLK_CTL0 外设时钟控制寄存器 0	466
31.3.2 PCLK_CTL1 外设时钟控制寄存器 1	466
31.3.3 PCLK_CTL2 外设时钟控制寄存器 2	467
31.3.4 PCLK_CTL3 外设时钟控制寄存器 3	467
32 电源结构及功耗模式 (PM,power manage)	468
32.1 电源配备介绍.....	468
32.2 电源地引脚说明.....	469
32.3 电压调制器.....	469
32.4 VREG 电源域分类.....	470
32.5 电源结构及功耗模式相关寄存器.....	471
32.5.1 PM_CTL0 功耗模式控制寄存器 0.....	471
32.5.2 PM_CTL1 功耗模式控制寄存器 1.....	473
32.5.3 PM_CTL2 功耗模式控制寄存器 2.....	474
32.5.4 PM_STA0 功耗模式状态寄存器 0.....	477
32.5.5 PM_STA1 功耗模式状态寄存器 1.....	478
32.5.6 PM_STAC 功耗模式状态清零寄存器	478
32.5.7 PM_CAL0 校准寄存器 0.....	479
32.5.8 PM_CAL1 校准寄存器 1	480
32.6 工作模式介绍.....	481
32.6.1 工作模式切换关系图.....	483
32.6.2 正常运行模式.....	483
32.6.3 普通休眠模式.....	483
32.6.4 休眠模式间关系表.....	484

32.6.5 停止模式.....	484
32.6.5.1 Stop0 模式.....	484
32.6.5.2 Stop1 模式.....	485
32.6.6 待机模式.....	485
32.6.7 关断模式.....	486
32.6.8 工作模式与电源域关系表.....	488
32.7 低功耗模式外部唤醒引脚 Wakeup Pin.....	489
33 备份域 (BKP)	490
33.1 概述.....	490
33.2 备份域数据寄存器组相关寄存器.....	491
33.2.1 BKP_CTL 备份域控制寄存器.....	491
33.2.2 BKP_INT 备份域中断控制寄存器.....	492
33.2.3 BKP_DATAx 备份域数据寄存器 (x=0~31)	493
33.3 寄存器组的复位.....	494
33.4 寄存器组的读写.....	494
33.5 侵入检测.....	494
33.6 侵入检测信号同步/异步模式.....	494
33.7 外部时钟旁路功能.....	495
34 Flash 程序存储器	496
34.1 FLASH 概述.....	496
34.2 FLASH 相关寄存器.....	497
34.2.1 FLASH_ISPCON0 控制寄存器.....	497
34.2.2 FLASH_ISPCON1 控制寄存器.....	498
34.2.3 FLASH_ISPCMD 控制寄存器.....	498
34.2.4 FLASH_ISPTRG 控制寄存器.....	499
34.2.5 FLASH_CFG 控制寄存器.....	499
34.2.6 FLASH_ISPADDR 地址寄存器.....	500
34.2.7 FLASH_STATE 状态寄存器.....	500
34.2.8 FLASH_NVMUNLOCK 解锁 KEY 寄存器.....	500
34.2.9 FLASH_PROUNLOCK 解锁 KEY 寄存器.....	501
34.2.10 FLASH_CFGUNLOCK 解锁 KEY 寄存器.....	501
34.2.11 FLASH_CSSTART 寄存器.....	501
34.2.12 FLASH_CSSTOP 寄存器.....	501
34.2.13 FLASH_CSRES0 寄存器.....	502
34.2.14 FLASH_CSRES1 寄存器.....	502
34.2.15 FLASH_CSRES2 寄存器.....	502
34.2.16 FLASH_CSRES3 寄存器.....	502
34.3 FLASH 存储映射.....	503
34.4 信息区内容介绍.....	504
34.5 FLASH 读接口.....	504
34.5.1 CPU 时钟频率和 FLASH 读时间.....	504
34.6 FLASH 加密.....	505

34.6.1 调试引脚相关.....	506
34.7 Check_Sum.....	507
34.7.1 功能说明.....	507
34.7.2 Check_sum 算法.....	507
35 循环冗余校验单元 (CRC)	508
35.1 概述.....	508
35.2 CRC 相关寄存器.....	509
35.2.1 CRC_CTL CRC 控制寄存器 0	509
35.2.2 CRC_DATA CRC 数据寄存器.....	510
35.2.3 CRC_RSLT CRC 结果寄存器.....	510
35.2.4 CRC_INIT CRC 初始值寄存器.....	510
35.2.5 CRC_PLN CRC 多项式寄存器.....	510
35.2.6 CRC_RXOR CRC 结果异或值寄存器.....	511
35.2.7 CRC_IDATA CRC 独立数据寄存器	511
35.2.8 CRC_TEMP CRC 缓存寄存器	511
35.3 CRC 操作.....	512
35.4 CRC 多项式操作.....	512
35.5 CRC 输入数据.....	513
35.5.1 输入数据格式.....	513
35.5.2 反序操作.....	513
35.6 CRC 运算结果反序操作.....	513
35.7 CRC 运算初始值设置.....	513
35.8 CRC 独立寄存器.....	514
35.9 常见多项式控制.....	514
35.10 CRC 操作步骤.....	514
36 封装信息	516
36.1 QFN32 封装.....	516
36.2 48 脚 LQFP 封装.....	517
36.3 64 脚 LQFP 封装.....	518
36.4 64 脚 LQFP64 封装 (10.0mm*10.0mm)	519
36.5 100 脚 LQFP 封装.....	520
37 产品标识体系.....	521
38 声明及销售网络.....	522
附录 1 汇编指令集.....	523
条件执行编码.....	525
传送指令.....	526
存取指令.....	527
数学指令.....	529
逻辑指令.....	530
位操作指令.....	531

循环/移位指令.....	532
比较/跳过指令.....	533
程序流指令.....	534
堆栈/出栈指令.....	535
控制指令.....	536
附录 2 用户手册更新记录.....	537

图示列表

图 2-1 ISP 模式编程接口.....	41
图 2-2 DPI 模式编程接口.....	41
图 3-1 GPIO 结构框图.....	42
图 5-1 振荡器结构框图.....	51
图 5-2 外部高频振荡器原理图.....	61
图 5-3 外部低频振荡器原理图.....	62
图 5-4 时钟同步时序图.....	63
图 5-5 时钟故障检测原理图.....	64
图 5-6 时钟故障检测时序图.....	65
图 6-1 存储空间映射.....	68
图 7-1 中断结构框图.....	96
图 7-2 PRIGROUP<1:0>为 00 时.....	98
图 7-3 栈帧内容.....	99
图 7-4 引脚外部中断线映射图.....	103
图 8-1 双字对齐堆栈方式.....	111
图 8-2 字对齐堆栈方式.....	111
图 9-1 DMA 结构框图.....	116
图 10-1 节拍定时器结构框图.....	129
图 10-2 重载值 0x20, SCLK/2 作为计数时钟, 使能 TICKINTEN.....	132
图 11-1 基本定时器结构框图.....	133
图 11-2 基本定时器中断框图.....	139
图 11-3 向上计数, 周期值 0x03.....	140
图 11-4 向下计数, 周期值 0x03.....	140
图 11-5 向上向下计数上下均溢出, 周期值 0x03.....	141
图 12-1 通用定时器结构框图.....	145
图 12-2 向上计数, 周期值 0x03.....	152
图 12-3 向下计数, 周期值 0x03.....	152
图 12-4 向上向下计数上下均溢出, 周期值 0x03.....	153
图 12-5 通用定时器向上计数启动更新和周期更新.....	156
图 12-6 通用定时器向下计数周期更新.....	157
图 12-7 通用定时器向上计数立即更新.....	157
图 12-8 通用定时器向下计数立即更新.....	158
图 13-1 高级定时器结构框图.....	160
图 13-2 向上计数, 计数周期 0x03, 分频器 2 设置 1 分频.....	169
图 13-3 向下计数, 计数周期 0x03, 分频器 2 设置 1 分频.....	170
图 13-4 向上向下计数上下均溢出, 计数周期 0x03, 分频器 2 设置 1 分频.....	170
图 13-5 TxCK 和 TzCK 框图.....	172
图 13-6 CCT 功能原理框图.....	176
图 14-1 CCP 模块结构框图.....	178
图 14-2 通用定时器/CCP 中断框图.....	187
图 14-3 捕捉模式原理框图.....	188

图 14-4 霍尔元件实例原理框图	190
图 14-5 比较模式原理框图	191
图 14-6 PWM 原理框图.....	192
图 14-7 向上计数边沿对齐 PWM.....	194
图 14-8 向下计数边沿对齐 PWM.....	194
图 14-9 中心对齐 PWM.....	195
图 15-1 ECCP 模块结构框图	198
图 15-2 高级定时器/ECCP 中断框图	211
图 15-3 捕捉模式原理框图	218
图 15-4 比较模式原理框图	220
图 15-5 PWM 通道原理框图.....	221
图 15-6 ECCP 向上计数 PWM 输出.....	224
图 15-7 ECCP 向下计数 PWM 输出.....	224
图 15-8 ECCP 中心对齐模式 PWM 输出.....	225
图 15-9 PWMx 输出示例.....	226
图 15-10 ECCP 互补输出模式示例	226
图 15-11 ECCP 互补输出模式应用举例.....	227
图 15-12 ECCP 死区插入模式示例	228
图 15-13 ECCP 自动相移模式	230
图 15-14 ECCP 自动关断模式原理框图.....	230
图 15-15 ECCP 自动重启与软件重启示例.....	231
图 15-16 ECCP 位置检测原理框图	232
图 15-17 信号采样时序图	232
图 15-18 CMPE=0 位置检测时序图	233
图 15-19 CMPE=1 位置检测时序图	233
图 16-1 QEI 模块状态分布图.....	235
图 16-2 QEI 模块结构框图.....	235
图 16-3 QEI 中断框图.....	239
图 16-4 QEI 可编程噪声滤波器.....	240
图 16-5 x4 模式下正交编码信号	240
图 16-6 x2 模式下正交编码信号	241
图 16-7 以 x4 模式为例的索引脉冲复位.....	242
图 16-8 QEI 相位检测错误示意图.....	243
图 16-9 QEI 的启动说明图 1 (X4 模式, 周期值为 10, 1 分频)	244
图 17-1 模数(A/D)转换模块框图.....	246
图 17-2 ADC 转换时序图	262
图 17-3 数据对齐方式	264
图 17-4 电压参考缓冲器	266
图 17-5 ADC 中断逻辑图	267
图 17-6 高优先级通道同步模式	268
图 17-7 常规通道同步模式	268
图 17-8 快速交叉模式	269
图 17-9 交替触发-无间隔交替模式.....	269
图 17-10 交替触发-有间隔交替模式.....	269

图 17-11 混合常规通道同步+高优先级通道同步模式.....	270
图 17-12 混合常规通道同步+交替触发模式 1	270
图 17-13 混合常规通道同步+交替触发模式 2	270
图 17-14 混合高优先级通道同步交叉模式.....	271
图 18-1 数模 DACx 转换器框图.....	272
图 18-2 噪声发生器	278
图 18-3 LFSR 产生波形的 DAC 转换	279
图 18-4 三角波发生器	279
图 18-5 生成三角波波形的 DAC 转换	280
图 18-6 DAC 校准示意图.....	280
图 19-1 模拟比较器 3 框图	282
图 19-2 模拟比较器 0/1/2 框图	283
图 19-3 模拟比较器范围控制	291
图 19-4 数字滤波器时序图	292
图 19-5 数字滤波器时序图	292
图 19-6 CMP 中断逻辑框图.....	293
图 20-1 USART 结构框图.....	295
图 20-2 发送接收临界状态	307
图 20-3 自动波特率检测示意图	308
图 20-4 全双工异步通信数据结构图	309
图 20-5 全双工异步连接框图	309
图 20-6 全双工异步发送（两字符间有空闲位）	310
图 20-7 全双工异步发送（一字符接一字符）	310
图 20-8 USART 半双工异步连接框图.....	312
图 20-9 半双工同步发送（SCKPS = 0）	313
图 20-10 半双工同步发送(SCKPS=1)	314
图 20-11 半双工同步接收（主控模式，RXEN=1，SCKPS=0）	315
图 20-12 半双工同步接收（主控模式，RXEN=1，SCKPS=1）	315
图 20-13 硬件流控制	316
图 20-14 IrDA 数据调制.....	317
图 20-15 7816 连接方式	318
图 20-16 7816 接口时序	318
图 20-17 数据发送流程	319
图 20-18 数据接收流程	320
图 20-19 单线模式连接示意	321
图 20-20 单线模式连接示意	321
图 20-21 USART 中断框图.....	322
图 21-1 SPI 原理框图.....	325
图 21-2 SPI 主从控制器连接（SPIIOS=0）.....	332
图 21-3 SPI 主从控制器连接（SPIIOS=1）.....	332
图 21-4 SPI 主控模式时序图（8 位数据）	333
图 21-5 从动时序图（8 位数据）	334
图 21-6 I2S 飞利浦协议波形(16/32 位全精度).....	336
图 21-7 MSB 对齐 16 位或 32 位全精度.....	336

图 21-8 LSB 对齐 16 位或 32 位全精度	336
图 21-9 PCM 标准波形(16 位)	337
图 21-10 音频采样频率定义	337
图 21-11 SPI 中断框图	339
图 22-1 I2C 模块结构框图	341
图 22-2 七位地址接收模式	349
图 22-3 十位地址接收模式	349
图 22-4 七位地址发送模式	350
图 22-5 十位地址发送模式	350
图 22-6 从动模式广播呼叫模式	352
图 22-7 主控模式结构框图	353
图 22-8 带有仲裁的波特率发生时序	354
图 22-9 第一个启动位时序	355
图 22-10 I2C 协议复合数据帧	355
图 22-11 重复启动时序	355
图 22-12 I2C 协议典型写数据帧格式	356
图 22-13 I2C 主模式七位地址发送时序	356
图 22-14 I2C 协议典型读数据帧时序	357
图 22-15 I2C 主模式接收数据时序图	357
图 22-16 发送模式停止条件	358
图 22-17 主控发送模式下的时钟仲裁时序	358
图 22-18 I2C 发送和应答时的总线冲突时序	359
图 22-19 I2C 中断框图	363
图 23-1 EXIC 模块框图	364
图 23-2 EXIC 数据传输时序	369
图 23-3 EXIC 写时序	369
图 23-4 EXIC 读时序	369
图 24-1 RTC 结构框图	371
图 25-1 CAN 模块结构框图	389
图 25-2 CAN 波特率原理图	393
图 25-3 标准帧验收滤波示意图	397
图 25-4 扩展帧验收滤波示意图	398
图 25-5 信息接收区域顺序	403
图 25-6 区域内信息接收顺序	404
图 25-7 CAN 中断框图	407
图 25-8 CAN 协议帧格式图	408
图 25-9 CAN 网络拓扑	409
图 25-10 错误状态图	412
图 26-1 独立看门狗原理框图	417
图 27-1 窗口看门狗原理框图 (1)	422
图 27-2 窗口看门狗原理图 (2)	422
图 28-1 USB 模块结构框图	424
图 28-2 USB_USTAT FIFO	428
图 28-3 缓冲区描述符示例	436

图 28-4 缓冲模式的缓冲区描述符表映射.....	437
图 28-5 USB 中断逻辑汇集.....	439
图 28-6 USB 事务和中断事件示例.....	439
图 28-7 USB 层.....	440
图 29-1 CFGL 模块结构框图	443
图 29-2 CFGL 数据输入结构框图	455
图 29-3 CFGL 逻辑功能结构框图	457
图 30-1 片内复位电路简化框图	459
图 30-2 POR/PDR 电压说明简图.....	462
图 30-3 BOR 电压说明简图	462
图 30-4 建议 NRST 复位电路	463
图 30-5 电压检测说明简图	464
图 32-1 电源结构框图	468
图 32-2 VREG 域框图.....	470
图 32-3 工作模式切换关系图	483
图 34-1 FLASH 的地址映射.....	503
图 34-2 FLASH 信息区的地址映射.....	503
图 34-3 FLASH 访问方式.....	504
图 35-1 CRC 结构框图	508
图 36-1 QFN32 封装图	516
图 36-2 LQFP48 封装图.....	517
图 36-3 LQFP64 封装图（7.0mm*7.0mm）	518
图 36-4 LQFP64 封装图（10.0mm*10.0mm）	519
图 36-5 LQFP100 封装图.....	520

表格列表

表 1-1 INT_EIE0 中断使能寄存器 0.....	37
表 1-2 GPIOx_POR 输出状态寄存器.....	38
表 3-1 GPIOx 相关的寄存器.....	43
表 3-2 GPIOx_PIR 输入状态寄存器.....	43
表 3-3 GPIOx_POR 输出状态寄存器.....	44
表 3-4 GPIOx_PUR 上拉使能寄存器.....	44
表 3-5 GPIOx_PDR 下拉使能寄存器.....	44
表 3-6 GPIOx_PODR 开漏输出控制寄存器.....	45
表 3-7 GPIOx_PMOD 端口方向控制寄存器.....	45
表 3-8 GPIOx_OMOD 端口速度控制寄存器.....	45
表 3-9 GPIOx_LOCK 端口配置锁定寄存器.....	46
表 4-1 GPIOx 引脚重映射相关寄存器.....	48
表 4-2 GPIOx_RMP Px 重映射控制寄存器.....	48
表 4-3 GPIOx_RMP Px 重映射控制寄存器.....	49
表 5-1 振荡器相关寄存器 1.....	52
表 5-2 振荡器相关寄存器 2.....	52
表 5-3 OSC_CTL0 振荡器控制寄存器 0.....	52
表 5-4 OSC_CTL1 振荡器控制寄存器 1.....	54
表 5-5 OSC_INT 振荡器中断控制寄存器.....	55
表 5-6 OSC_CTL2 振荡器控制寄存器 2.....	56
表 5-7 OSC_HFOSCCAL0 高频振荡器校准寄存器 0.....	57
表 5-8 OSC_HFOSCCAL1 高频振荡器校准寄存器 1.....	58
表 5-9 电流增益选择表.....	59
表 5-10 PLL_CTL PLL 控制寄存器.....	59
表 5-11 时钟稳定条件.....	65
表 6-1 存储器预留空间分配示意.....	68
表 6-2 加密访问权限说明.....	69
表 7-1 内核中断向量.....	70
表 7-2 外设中断向量表.....	71
表 7-3 中断相关寄存器.....	73
表 7-4 INT_CTL0 中断控制寄存器 0.....	74
表 7-5 INT_EIE0 中断使能寄存器 0.....	74
表 7-6 INT_EIE1 中断使能寄存器 1.....	75
表 7-7 INT_EIE2 中断使能寄存器 2.....	77
表 7-8 INT{EIF0 中断标志位寄存器 0.....	80
表 7-9 INT{EIF1 中断标志位寄存器 1.....	81
表 7-10 INT{EIF2 中断标志位寄存器 2.....	83
表 7-11 INT_IP0 中断优先级寄存器 0.....	85
表 7-12 INT_IP1 中断优先级寄存器 1.....	85
表 7-13 INT_IP2 中断优先级寄存器 2.....	85
表 7-14 INT_IP3 中断优先级寄存器 3.....	85

表 7-15 INT_IP4 中断优先级寄存器 4	86
表 7-16 INT_IP5 中断优先级寄存器 5	86
表 7-17 INT_IP6 中断优先级寄存器 6	86
表 7-18 INT_IP7 中断优先级寄存器 7	86
表 7-19 INT_IP8 中断优先级寄存器 8	86
表 7-20 INT_IP9 中断优先级寄存器 9	86
表 7-21 INT_IP10 中断优先级寄存器 10	86
表 7-22 INT_IP11 中断优先级寄存器 11	86
表 7-23 INT_IP12 中断优先级寄存器 12	87
表 7-24 INT_IP13 中断优先级寄存器 13	87
表 7-25 INT_IP14 中断优先级寄存器 14	87
表 7-26 INT_IP15 中断优先级寄存器 15	87
表 7-27 INT_IP16 中断优先级寄存器 16	87
表 7-28 INT_IP17 中断优先级寄存器 17	87
表 7-29 INT_IP18 中断优先级寄存器 18	87
表 7-30 INT_EINTMASK 外部中断屏蔽寄存器	90
表 7-31 INT_EINTRISE 外部中断上升沿选择寄存器	90
表 7-32 INT_EINTFALL 外部中断下降沿选择寄存器	90
表 7-33 INT_EINTF 外部中断标志位寄存器	90
表 7-34 INT_EINTSS0 外部中断源选择寄存器 0	91
表 7-35 INT_EINTSS1 外部中断源选择寄存器 1	93
表 7-36 INT_CTL1 中断控制寄存器 1	95
表 7-37 内核中断类型描述	97
表 7-38 xPSR 寄存器（映射寄存器）	99
表 7-39 中断返回 INTRETURN 行为	100
表 8-1 系统控制相关的寄存器	108
表 8-2 SYS_PSW 程序状态字寄存器	108
表 8-3 SYS_MCTL 系统模式控制寄存器	108
表 8-4 SYS_ARCTL 应用和复位控制寄存器	109
表 8-5 SYS_VECTOFF 中断向量表重映射控制寄存器	110
表 8-6 SYS_RAMSPARAM 空间指示寄存器	110
表 8-7 SYS_MEMCTL 程序空间控制寄存器	110
表 8-8 休眠模式及休眠唤醒设置	113
表 8-9 向量表映射状态	114
表 9-1 DMA 相关寄存器	117
表 9-2 DMAx_CTLRy DMAx 通道 y 控制寄存器	118
表 9-3 DMAx_PADDRy DMAx 通道 y 外设地址寄存器	119
表 9-4 DMAx_MADDRy DMAx 通道 y 存储器地址寄存器	119
表 9-5 DMAx_CPARy DMAx 通道 y 当前外设地址寄存器	120
表 9-6 DMAx_CMARy DMAx 通道 y 当前存储器地址寄存器	120
表 9-7 DMAx_NCTy DMAx 通道 y 当前剩余数据寄存器	120
表 9-8 DMAx_LIFR DMAx 通道 1~7 中断标志寄存器	120
表 9-9 DMAx_LIER DMAx1~7 通道中断使能寄存器	121
表 9-10 DMA0 请求	123

表 9-11 DMA1 请求	124
表 9-12 DMA 与 CPU 冲突关系（高速度模式）	127
表 9-13 DMA 与 CPU 冲突关系（高效模式）	127
表 10-1 系统节拍定时器相关寄存器	129
表 10-2 ST_CTL 寄存器	130
表 10-3 ST_RELOAD 系统节拍定时器重加载寄存器	130
表 10-4 ST_CV 系统节拍定时器寄存器	130
表 10-5 ST_CALI 校验寄存器	131
表 11-1 基本定时器相关寄存器	134
表 11-2 Tx_CNT 定时器寄存器	134
表 11-3 Tx_CTL1 Tx 控制寄存器 1	134
表 11-4 Tx_CTL2 Tx 控制寄存器 2	135
表 11-5 Tx_PRSC Tx 预分频寄存器	137
表 11-6 Tx_PPX 周期寄存器	137
表 11-7 Tx_DIER Tx 中断使能控制寄存器	137
表 11-8 Tx_SR Tx 中断状态寄存器	138
表 11-9 Tx_DIER Tx 中断标志清除寄存器	139
表 12-1 通用定时器相关寄存器 1(x=0,1,2,3,4,18,19,22,23)	146
表 12-2 通用定时器相关寄存器 2(x=20,21)	146
表 12-3 Tx_CNT Tx 定时器寄存器(x=0,1,2,3,4,18,19,22,23)	146
表 12-4 Tx_CNT Tx 定时器寄存器(x=20,21)	147
表 12-5 Tx_CTL1 Tx 控制寄存器 1(x=0,1,2,3,4,18,19,20,21,22,23)	147
表 12-6 Tx_CTL2 Tx 控制寄存器 2(x=0,1,2,3,4,18,19,20,21,22,23)	148
表 12-7 Tx_PRSC Tx 预分频器(x=0,1,2,3,4,18,19,22,23)	150
表 12-8 Tx_PRSC Tx 预分频器(x=20,21)	150
表 12-9 Tx_PPX 周期寄存器(x=0,1,2,3,4,18,19,22,23)	150
表 12-10 Tx_PPX 周期寄存器(x=20,21)	150
表 12-11 Tx_UDTIM 更新计数器(x=0,1,2,3,4,18,19,20,21,22,23)	151
表 13-1 高级定时器相关寄存器	161
表 13-2 Tx_CNT 定时器寄存器	161
表 13-3 Tz_CNT 定时器寄存器	162
表 13-4 Tx_PPX 周期寄存器	162
表 13-5 Tz_PPZ 周期寄存器	162
表 13-6 Tx_PRSC Tx 预分频寄存器	163
表 13-7 Tz_PRSC Tz 预分频寄存器	163
表 13-8 Tx_CCR0 Tx 触发 AD 寄存器 0	163
表 13-9 Tx_CCR1 Tx 触发 AD 寄存器 1	163
表 13-10 Tz_CCR0 Tz 触发 AD 寄存器 0	164
表 13-11 Tx_CTL Tx 控制寄存器	164
表 13-12 Tz_CTL Tz 控制寄存器	165
表 13-13 Tx_CCTCTL 比较器清零定时器 Tx 控制寄存器	167
表 13-14 Tz_CCTCTL 比较器清零定时器 Tz 控制寄存器	167
表 14-1 CCP 模块相关寄存器 1 (x=0,1,2,3,4,18,19,22,23)	179
表 14-2 CCP 模块相关寄存器 2 (x=20,21)	179

表 14-3 CCPx_Cy 捕捉寄存器(x=0,1,2,3,4,18,19,22,23).....	180
表 14-4 CCPx_Cy 捕捉寄存器(x=20,21).....	180
表 14-5 CCPx_SRIC CCPx 中断标志清除寄存器.....	181
表 14-6 CCPx_DF CCP 触发 DMA 请求标志寄存器.....	181
表 14-7 CCPx_CTL1 CCP 控制寄存器 1.....	182
表 14-8 CCPx_Ry 比较/PWM 占空比寄存器(x=0,1,2,3,4,18,19,22,23).....	183
表 14-9 CCPx_Ry 比较/PWM 占空比寄存器(x=20,21).....	183
表 14-10 CCPx_CTL2 CCP 控制寄存器 2(x=0,1,2,3,4,18,19,20,21,22,23).....	183
表 14-11 CCPx_CTL3 CCPx 控制寄存器 3.....	184
表 14-12 CCPx_EGIF CCPx 中断状态/事件产生寄存器.....	186
表 15-1 ECCP 相关寄存器 1.....	199
表 15-2 ECCP 相关寄存器 2.....	199
表 15-3 ECCPx_CTL1 ECCPx 控制寄存器 1.....	200
表 15-4 ECCPx_Ry 比较/PWM 占空比寄存器.....	202
表 15-5 ECCPx_PXUDCTL 更新控制寄存器.....	202
表 15-6 ECCPx_CTL2 ECCP 控制寄存器 2.....	203
表 15-7 ECCPx_PXDTCTL 死区控制寄存器.....	204
表 15-8 ECCPx_PWMXOC 输出控制寄存器.....	205
表 15-9 ECCPx_PXATRCTL 输出配置寄存器.....	206
表 15-10 ECCPx_PXASCTL0 关断控制寄存器 0.....	207
表 15-11 ECCPx_PXASCTL1 关断控制寄存器 1.....	208
表 15-12 ECCPx_IE ECCPx 中断使能寄存器.....	209
表 15-13 ECCPx_EGIF ECCPx 中断状态/事件产生寄存器.....	210
表 15-14 Tx_UDTIMX 更新计数器.....	212
表 15-15 Tz_UDTIMZ 更新计数器.....	212
表 15-16 ECCPx_DF 触发 DMA 请求标志寄存器.....	212
表 15-17 ECCPx_Cy ECCPx 捕捉寄存器.....	213
表 15-18 ECCPx_DE ECCPx 请求 DMA 使能寄存器.....	213
表 15-19 ECCPx_SRIC ECCPx 中断标志清除寄存器.....	214
表 15-20 ECCPx_CTL3 ECCP 控制寄存器 3.....	215
表 15-21 ECCPx_PDCTL ECCP 位置检测控制寄存器.....	216
表 15-22 ECCPx_PXASCTL 关断控制寄存器.....	216
表 15-23 PWMx 通道输出模式控制及有效引脚.....	221
表 15-24 PWM 信号重启恢复时刻表.....	231
表 16-1 QEI 模块相关寄存器.....	236
表 16-2 QEIy_CTL0 QEIy 控制寄存器.....	236
表 16-3 QEIy_CTL1 QEIy 数字滤波控制寄存器.....	237
表 16-4 Tx_CNNTx 寄存器.....	238
表 16-5 Tx_PPX 周期寄存器.....	238
表 16-6 Tx_PRSTx 预分频器.....	238
表 16-7 QEIy_DIER QEIy 中断控制寄存器.....	239
表 17-1 ADC 相关的寄存器.....	247
表 17-2 ADCx_CTL0 ADC 控制寄存器 0.....	247
表 17-3 ADCx_CTL1 ADCx 控制寄存器 1.....	249

表 17-4 ADCx_SCANSQ0 常规通道扫描序列设置寄存器.....	254
表 17-5 ADCx_SCANSQ1 常规通道扫描序列设置寄存器.....	254
表 17-6 ADCx_SCANSQ2 常规通道扫描序列设置寄存器.....	255
表 17-7 ADCx_SCANSQ3 常规通道扫描序列设置寄存器.....	255
表 17-8 ADCx_HSCANSQ 高优先级通道扫描序列设置寄存器.....	256
表 17-9 ADCx_WDH 模拟看门狗高阈值寄存器.....	257
表 17-10 ADCx_WDL 模拟看门狗低阈值寄存器.....	257
表 17-11 ADCx_DATA 常规通道数据寄存器.....	257
表 17-12 ADCx_HPDATAy 高优先级通道数据寄存器.....	257
表 17-13 ADCx_HPDOFFy 高优先级通道数据失调寄存器.....	258
表 17-14 ADCx_STATE ADC 状态寄存器.....	258
表 17-15 ADC0_DELAY ADC 快速交叉延时寄存器.....	260
表 17-16 内部信号通道.....	267
表 18-1 DACx 相关寄存器.....	273
表 18-2 DACx_CTL DACx 控制寄存器.....	273
表 18-3 DACx_DAHD DACx 数据保持寄存器.....	275
表 18-4 DACx_DATA DACx 输出数据寄存器.....	275
表 18-5 DACx_CTL1 DACx 控制寄存器 1.....	275
表 18-6 DACx_CAL DACx 校验寄存器.....	276
表 18-7 DAC 触发事件.....	277
表 19-1 CMP0/1/2/3 正端和负端输入通道.....	284
表 19-2 模拟比较器相关寄存器.....	285
表 19-3 CMP_CTLx 比较器控制寄存器 0/1/2.....	285
表 19-4 CMP_CTL3 比较器控制寄存器 3.....	286
表 19-5 CMP_CTL4 比较器控制寄存器 4.....	288
表 19-6 比较器输出状态与极性关系.....	291
表 20-1 USART 相关的寄存器.....	297
表 20-2 USARTx_CTLR USART 控制寄存器.....	297
表 20-3 USARTx_BRGR USART 波特率寄存器.....	300
表 20-4 USARTx_STR USART 状态寄存器.....	300
表 20-5 USARTx_TBUFR USART 发送数据寄存器.....	302
表 20-6 USARTx_RBUFR USART 接收数据寄存器.....	302
表 20-7 USARTx_U7816R 7816 控制寄存器.....	302
表 20-8 USARTx_IER USART 中断使能寄存器.....	304
表 20-9 USARTx_ADM USART 地址匹配设置寄存器.....	305
表 21-1 SPIx 模块相关寄存器.....	326
表 21-2 SPIx_BRGR SPI 波特率寄存器.....	326
表 21-3 SPIx_CTLR SPI 控制寄存器.....	327
表 21-4 SPIx_BUFR SPI 数据寄存器.....	328
表 21-5 SPIx_STR SPI 状态寄存器.....	329
表 22-1 I2Cx 模块相关寄存器.....	342
表 22-2 I2Cx_CTLR I2C 控制寄存器.....	342
表 22-3 I2Cx_SR I2C 状态寄存器.....	344
表 22-4 I2Cx_BUFR I2C 缓存寄存器.....	346

表 22-5 I2Cx_ADDRy I2C 地址寄存器 (y=0,1,2,3)	346
表 22-6 I2Cx_BRGR I2C 波特率寄存器.....	346
表 22-7 I2Cx_IER I2C 中断使能寄存器	347
表 22-8 SMBUS 与 I2C 的比较.....	360
表 23-1 EXIC 相关寄存器	365
表 23-2 EXIC_CTL0 EXIC 接口控制寄存器 0	365
表 23-3 EXIC_CTL1 EXIC 接口控制寄存器 1	366
表 23-4 EXIC_STAT EXIC 状态寄存器.....	367
表 23-5 EXIC_WRITEBUF 写数据缓存寄存器.....	367
表 23-6 EXIC_READBUF 读数据缓存寄存器.....	368
表 24-1 RTC 相关寄存器.....	373
表 24-2 RTC_CR 实时时钟控制寄存器.....	373
表 24-3 RTC_ALRA 实时时钟闹钟 A 控制寄存器	375
表 24-4 RTC_ALRB 实时时钟闹钟 B 控制寄存器.....	375
表 24-5 RTC_TMR 实时时钟时间寄存器	376
表 24-6 RTC_DTR 实时时钟日期寄存器.....	377
表 24-7 RTC_TMER 实时时钟定时器寄存器.....	377
表 24-8 RTC_TCR 实时时钟定时器控制寄存器	377
表 24-9 RTC_IER 实时时钟中断使能寄存器.....	378
表 24-10 RTC_IFR 实时时钟中断标志寄存器.....	379
表 24-11 RTC_TMBR 实时时钟时间备份寄存器.....	381
表 24-12 RTC_DTBR 实时时钟日期备份寄存器	381
表 24-13 时间校正参考	382
表 24-14 校准例图 (校准周期为 20s, 延迟时间, 校准大小为 3)	383
表 24-15 校准例图 (校准周期为 20s, 加速时间, 校准大小为 3)	383
表 24-16 校准寄存器设定值表 1 (最小分解能=3.052ppm)	384
表 24-17 校准寄存器设定值表 2 (最小分解能=1.017ppm)	384
表 24-18 时间戳通道	385
表 24-19 节拍中断	386
表 25-1 CANx 模块相关寄存器.....	390
表 25-2 寄存器不同模式下功能	390
表 25-3 CANx_CTLR CAN 控制寄存器	391
表 25-4 CANx_BRGR CAN 波特率寄存器.....	393
表 25-5 CANx_RCR CAN 记录寄存器.....	394
表 25-6 CANx_EROR CAN 错误寄存器.....	396
表 25-7 CANx_ACRR CAN 验收代码寄存器.....	397
表 25-8 CANx_MSKR CAN 验收屏蔽寄存器	397
表 25-9 CANINFR CAN 数据信息寄存器.....	398
表 25-10 CANx_TX0R CAN 数据寄存器 0.....	398
表 25-11 CANx_TX1R CAN 数据寄存器 1.....	399
表 25-12 CANx_TX2R CAN 数据寄存器 2.....	399
表 25-13 CANx_RXDATA0 标准帧 CANx 接收数据信息寄存器.....	400
表 25-14 CANx_RXDATA1 标准帧 CANx 接收数据寄存器 1.....	400
表 25-15 CANx_RXDATA2 标准帧 CANx 接收数据寄存器 2.....	400

表 25-16 CANx_RXDATA3 标准帧 CAN 接收数据寄存器 3.....	401
表 25-17 CANx_RXDATA0 扩展帧 CAN 接收数据信息寄存器.....	401
表 25-18 CANx_RXDATA1 扩展帧 CAN 接收数据寄存器 1.....	401
表 25-19 CANx_RXDATA2 扩展帧 CAN 接收数据寄存器 2.....	402
表 25-20 CANx_RXDATA3 扩展帧 CAN 接收数据寄存器 3.....	402
表 25-21 接收数据字节表	402
表 25-22 CAN 和双端口 RAM 对应关系	402
表 25-23 CANx_IER CAN 中断使能寄存器	404
表 25-24 CANx_IFR CAN 中断标志寄存器	406
表 25-25 错误计数器与错误状态关系	412
表 25-26 错误计数器计数条件	413
表 26-1 IWDG 相关寄存器.....	415
表 26-2 IWDG_CTL 独立看门狗控制寄存器.....	415
表 26-3 IWDG_FD 独立看门狗喂狗寄存器.....	416
表 27-1 窗口看门狗相关寄存器	420
表 27-2 WWDT_CTL 窗口看门狗控制寄存器	420
表 27-3 WWDT_CNT 窗口看门狗计数寄存器.....	421
表 28-1 USB 模块相关寄存器.....	425
表 28-2 USB_UCON USB 控制寄存器.....	425
表 28-3 USB_USTAT USB 状态寄存器.....	427
表 28-4 USB_UADDR USB 地址寄存器.....	428
表 28-5 USB_UFRM USB 帧编号寄存器.....	428
表 28-6 USB_UIR USB 中断标志寄存器	429
表 28-7 USB_UIE USB 中断控制寄存器	430
表 28-8 USB_UEPn USB 端点 n 控制寄存器 (n=0,1,2,3,4,5,6,7)	431
表 28-9 USB_UGDB USB DEBUG 控制寄存器.....	432
表 28-10 USB_BDNSTAT 缓存区描述符 n 的状态寄存器	433
表 28-11 DTSEN 位对奇/偶 (DATA0/DATA1) 数据包接收的影响	434
表 28-12 不同缓冲模式的缓冲区描述符分配.....	438
表 29-1 CFGL 相关寄存器	444
表 29-2 CFGL1_CTL0 CFGL1 控制寄存器 0.....	444
表 29-3 CFGL2_CTL0 CFGL2 控制寄存器 0.....	446
表 29-4 CFGL1_CTL1 CFGL1 控制寄存器 1.....	448
表 29-5 CFGL2_CTL1 CFGL2 控制寄存器 1.....	451
表 29-6 CFGLIFR CFGL 中断控制寄存器.....	453
表 29-7 CFGL 数据输入通道对应表	456
表 30-1 复位相关的寄存器	460
表 30-2 RST_CTL0 复位控制寄存器 0.....	460
表 30-3 RST_CTL1 复位控制寄存器 1.....	460
表 30-4 RST_CTL2 复位控制寄存器 2.....	461
表 30-5 RST_CTL3 复位控制寄存器 3.....	461
表 31-1 外设模块时钟使能模块相关寄存器.....	466
表 31-2 PCLK_CTL0 外设时钟控制寄存器 0.....	466
表 31-3 PCLK_CTL1 外设时钟控制寄存器 1.....	466

表 31-4 PCLK_CTL2 外设时钟控制寄存器 2.....	467
表 31-5 PCLK_CTL3 外设时钟控制寄存器 3.....	467
表 32-1 电源介绍	469
表 32-2 功耗模式和电压调制器关系	469
表 32-3 复位相关的寄存器	471
表 32-4 PM_CTL0 功耗模式控制寄存器 0.....	471
表 32-5 PM_CTL1 功耗模式控制寄存器 1.....	473
表 32-6 PM_CTL2 功耗模式控制寄存器 2.....	474
表 32-7 PM_STA0 功耗模式状态寄存器 0.....	477
表 32-8 PM_STA1 功耗模式状态寄存器 1.....	478
表 32-9 PM_STAC 功耗模式状态清零寄存器.....	478
表 32-10 PM_CAL0 校准寄存器 0.....	479
表 32-11 PM_CAL1 校准寄存器 1.....	480
表 32-12 普通休眠模式配置表	484
表 32-13 休眠模式间关系表	484
表 32-14 停止模式配置表	484
表 32-15 停止模式配置表	485
表 32-16 待机模式配置表	486
表 32-17 关断模式配置表	486
表 32-18 工作模式与电源域关系表	488
表 33-1 备份域数据寄存器组相关的寄存器.....	491
表 33-2 BKP_CTL 备份域控制寄存器.....	491
表 33-3 BKP_INT 备份域中断控制寄存器.....	492
表 33-4 BKP_DATAx 备份域数据寄存器 (x=0~31)	493
表 34-1 振荡器相关寄存器	497
表 34-2 FLASH_ISPCON0 控制寄存器.....	497
表 34-3 FLASH_ISPCON1 控制寄存器.....	498
表 34-4 FLASH_ISPCMD 控制寄存器.....	498
表 34-5 FLASH_ISPTRG 控制寄存器.....	499
表 34-6 FLASH_CFG 控制寄存器	499
表 34-7 FLASH_ISPADDR 寄存器.....	500
表 34-8 FLASH_STATE 状态寄存器	500
表 34-9 FLASH_NVMUNLOCK 寄存器.....	500
表 34-10 FLASH_PROUNLOCK 寄存器	501
表 34-11 FLASH_CFGUNLOCK 寄存器.....	501
表 34-12 FLASH_CSSTART 寄存器.....	501
表 34-13 FLASH_CSSTOP 寄存器	501
表 34-14 FLASH_CSRES0 寄存器.....	502
表 34-15 FLASH_CSRES1 寄存器.....	502
表 34-16 FLASH_CSRES2 寄存器.....	502
表 34-17 FLASH_CSRES3 寄存器.....	502
表 34-18 信息区校准值信息	504
表 34-19 开预取时 FLASH 读时间与系统时钟.....	505
表 34-20 关预取时 FLASH 读时间与系统时钟.....	505

表 34-21 FLASH 加密状态表.....	506
表 34-22 调制引脚配置状态表	507
表 35-1 CRC 相关的寄存器	509
表 35-2 CRC_CTL CRC 控制寄存器 0.....	509
表 35-3 CRC_DATA CRC 数据寄存器	510
表 35-4 CRC_RSLT CRC 结果寄存器	510
表 35-5 CRC_INIT CRC 初始值寄存器.....	510
表 35-6 CRC_PLN CRC 多项式寄存器	510
表 35-7 CRC_RXOR CRC 结果异或值寄存器	511
表 35-8 CRC_IDATA CRC 独立数据寄存器	511
表 35-9 CRC_TEMP CRC 缓存寄存器.....	511
表 35-10 常见多项式及对应的 CRC_PLN 寄存器值	512
表 35-11 DATAREV 位反序操作示意	513
表 35-12 RSLTREV 位反序操作示意	513
表 35-13 常见多项式控制	514
附表 1 条件执行编码表.....	525
附表 2 传送指令.....	526
附表 3 存取指令.....	527
附表 4 数学指令.....	529
附表 5 逻辑指令.....	530
附表 6 位操作指令.....	531
附表 7 循环/位移指令.....	532
附表 8 比较/跳过指令.....	533
附表 9 程序流指令.....	534
附表 10 堆栈/出栈指令.....	535
附表 11 控制指令.....	536

1 文档说明

1.1 简介

本文档为 KF32 系列单片机芯片资源详细说明文档。该系列不同的型号拥有不同的芯片封装和资源，具体型号对应的封装及资源情况见对应的数据手册。

本文档主要面向 KF32 应用开发者或爱好者，为其提供了完整的功能说明。

更多详细信息请访问 CHIPON 官网 www.chipon-ic.com。

1.2 寄存器操作缩写说明

以下为本文中寄存器操作使用的字母缩写含义。

read/write (R/W): 软件可读写该位。

read-only (R): 软件只能读该位。

write-only (W): 软件只能写该位，读该位时返回默认值。

clear by write 0 (w0): 写 0 清零，写 1 无效。

clear by write 1 (w1): 写 1 清零，写 0 无效。

uncertain (u): 该位初始状态不确定。

1.3 寄存器地址说明

1.3.1 偏移地址

本文档所提及的寄存器地址计算形式如下：

寄存器地址 = 基地址 + 偏移地址

1.3.2 寄存器内容说明

表 1-1 INT_EIE0 中断使能寄存器 0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
R/W	R/W																R/W	R/W	R/W				R/W	R/W	R/W												
位名																	SYSTICKIE	PENDSVIE					SVCALLIE							ARIFAUPTIE	STACKIE						

如上图所示为寄存器描述的一般形式。其中，第一行为位序号，描述寄存器位置分布，其中 31 为最高位，0 为最低位。复位值行描述寄存器在复位条件下的默认值，0 表示复位时相应位为 0，1 表示复位时相应位为 1，u 表示复位时相应位为不定态，白色表明该位可操作，灰色表明该位不可操作或操作无效。R/W 行为寄存器操作描述，寄存器操作描述见寄存器操作缩写说明，白色且有对应的寄存器操作缩写表明该位可操作，灰色表明该位不可操作或操作无效。位名行描述了对应寄存器的功能缩写，用于区分寄存器不同位置对应的特殊功能，白色且有对应的功能名称的为有功能的寄存器位，灰色为无效位，配置时建议无效位设置成默认值。

1.4 寄存器操作说明

表 1-2 GPIOx_POR 输出状态寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0												
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W											
位名																	PXPOR15	PXPOR14	PXPOR13	PXPOR12	PXPOR11	PXPOR10	PXPOR9	PXPOR8	PXPOR7	PXPOR6	PXPOR5	PXPOR4	PXPOR3	PXPOR2	PXPOR1	PXPOR0												

PXPOR<15:0>: 端口 Px 输出数据

0 = 对应引脚为逻辑低电平

1 = 对应引脚为逻辑高电平

注: 上表中 x=A,B,C,D,E,F,G,H

如上图所示为 IO 口输出状态寄存器, 以该寄存器为例子说明对寄存器操作的事项。

如上所示, 上述寄存器名称为 GPIOx_POR (x=A,B,C,D,E,F,G,H), 假设需要对 PA1 口进行操作, 则操作对象为 GPIOA_POR 寄存器的 PXPOR1, 其它依次类推。

2 芯片介绍

2.1 芯片特征

● CPU

32 位高性能 KungFu32 内核
工作频率最高为 120MHz，可软件调节；
基于 16 位/32 位混合指令的高效指令集；
3 级流水线；
32×32 单周期乘法，32÷32 硬件除法；
支持中断优先级处理，实现自动中断堆栈；
13 个 32 位通用寄存器 R0~R12；
链接寄存器（R13/LR）；
堆栈指针寄存器（R14/MSP/PSP）；
程序计数器（R15/PC）；
24 位系统节拍定时器；

● 存储器

最高 512KByte FLASH，带 ECC 校验；
最高 128KByte RAM，其中 32K Byte 带 ECC 校验；
6 个 256Byte 双端口 RAM，带 ECC 校验；
1 个 512Byte 双端口 RAM，带 ECC 校验；
16K Byte 引导 ROM；
FLASH 可经受 100 000 次写操作；

● 特殊功能

内嵌上电复位电路；
低电压检测及低电压复位；
可编程电压检测；
硬件双看门狗；
6 种时钟源，3 种时钟信号选择；
支持两线串行编程/在线调试；

● I/O 口配置

支持输入输出配置；
支持内置上拉/下拉功能；
支持推挽输出和开漏输出模式；
支持数字/模拟引脚设置；
支持引脚功能重映射；
施密特电平输入；

● 定时器/计数器

定时器 5/6/9/10：16 位高级定时器，其中定

时器 5/6 支持 ECCP5，定时器 9/10 支持 ECCP9；

定时器 0/1/2/3/4/18/19/22/23：16 位通用定时器，其中定时器 0/1/2/3/4/18/19/22/23 支持 CCP0/1/2/3/4/18/19/22/23；

定时器 20/21：32 位通用定时器，其中定时器 20/21 分别支持 CCP20/21；

定时器 14/15：16 位基本定时器；

定时器 7/8：用于 QEI0/1；

● 其它外设

2 个 7 通道 DMA；

1 个硬件 CRC32 模块；

4 个 SPI 总线模块（兼容 I2S）；

4 个 I2C 总线模块（兼容 SMBUS/PMBUS）；

8 个 USART 模块（兼容 7816/LIN/IRDA 功能）；

3 个 CAN2.0B 模块；

1 个 CFGL 可配置逻辑单元模块；

1 个独立的 RTC（万年历）；

1 个 USB 模块；

1 个 EXIC 接口模块；

3 个 12 位 ADC 模块，支持最多 42 个通道；

2 个 12 位 DAC 模块；

2 个 CMP 比较器模块；

● 功耗管理

5 种功耗模式：正常运行模式、休眠模式、停止模式、待机模式、关断模式

● 工作条件

工作电压：1.8V~3.6V

工作温度范围：-40~125°C

2.2 系统概述

该单片机为基于 KF32 内核架构开发的单片机，KF32 为 32 位冯诺依曼三级流水线结构的高性能处理器内核，KF32 内核具有以下特点：

- 三级流水线结构
- 基于 16 位/32 位混合指令的高效指令集
- 支持 13 个 32 位通用寄存器 (R0~R12)，1 个链接寄存器 (R13/LR)，1 个堆栈指针寄存器 (R14/MSP/PSP，R14 可软件选择 MSP/PSP)，1 个程序计数器 (R15/PC)
- 支持 32x32 单周期硬件乘法
- 支持 32÷32 硬件除法
- 支持 8/16/32 位数据访存操作，支持 8/16/32/64 位数据处理
- 支持加减移位和逻辑运算
- 支持相对/绝对跳转，支持条件跳转
- 具有统一的存储空间，32 位地址位宽，支持 4GB 存储空间
- 支持最多 64+16 个中断请求和 16 个中断优先级
- 支持多种休眠模式
- 支持 24 位系统节拍定时器
- 提供了可编程存储器访问权限控制
- 支持多种操作系统 (OS) 特性

2.3 指令集

单片机拥有基于 16 位/32 位混合指令的高效指令集，拥有多种操作模式。具体指令集请参考“附录 1 汇编指令集”。

2.4 在线编程和调试

开发人员和用户可以使用未编程的单片机来制造电路板，然后对其在线编程，调试等。只要有电脑、USB 下载线和编程器，即可在任何时候，任何地点，对电路板上的单片机程序进行更新。

可以通过下列方式实现对单片机的编程或调试。

- ISP 模式进行编程
- DPI 模式进行调试或编程

2.4.1 ISP 模式

在ROM启动模式中可以直接通过串口实现对芯片的编程。该模式接口连接如下图所示。



图 2-1 ISP 模式编程接口

2.4.2 DPI 模式

DPI (Debug/Program Interface) 模式通过 KF32DP 编程器对芯片进行调试或编程。该模式接口连接如下图所示。

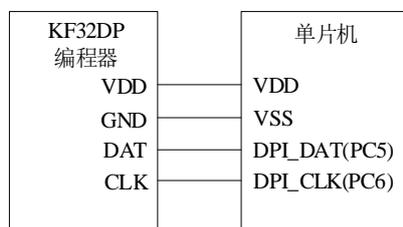


图 2-2 DPI 模式编程接口

3 I/O 端口介绍 (GPIO)

3.1 概述

单片机有不同的管脚封装，包括 PA 口、PB 口、PC 口、PD 口、PE 口、PF 口、PG 口、PH 口和电源等特殊引脚。每个 Px (x=A,B,C,D,E,F,G,H,) 最多有 16 个引脚。

端口特性如下：

- 数字输入
- 数字输出
 - 推挽式输出
 - 开漏输出
- 模拟口
- 独立端口上/下拉控制

3.2 GPIO 结构框图

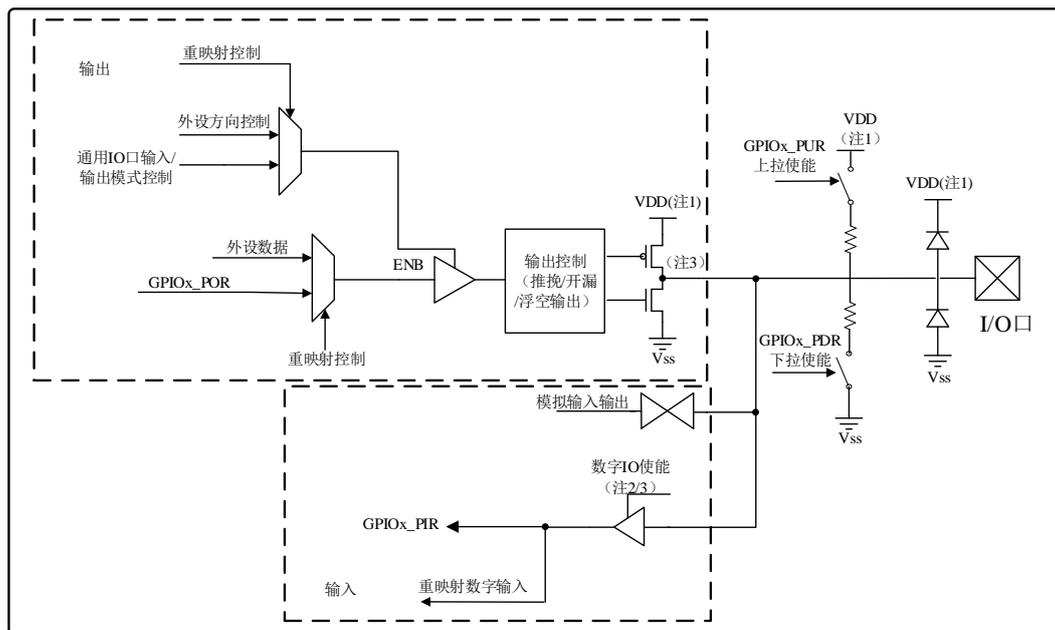


图 3-1 GPIO 结构框图

注 1: VDD 为电源引脚输入电源。

注 2: 当 IO 口模式为 GPIO 输入/输出或重映射模式时，数字 IO 使能都打开。

注 3: 引脚支持低功耗模式下输入，输出电路能够保持电压并维持低功耗前的输出。

3.3 GPIO 相关寄存器

 表 3-1 GPIOx 相关的寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x000	GPIOx_PIR	R	Px 口输入状态寄存器	0x0000 0000
0x004	GPIOx_POR	R/W	Px 口输出状态寄存器	0x0000 0000
0x008	GPIOx_PUR	R/W	Px 口上拉使能寄存器 ²	0x0000 0000
0x00C	GPIOx_PDR	R/W	Px 口下拉使能寄存器	0x0000 0000
0x010	GPIOx_PODR	R/W	Px 口开漏输出控制寄存器	0x0000 0000
0x014	GPIOx_PMOD	R/W	Px 口端口方向控制寄存器 ³	0xFFFF FFFF
0x018	GPIOx_OMOD	R/W	Px 口端口速度控制寄存器	0x0000 0000
0x01C	GPIOx_LOCK	R/W	Px 端口配置锁定寄存器	0x0000 0000

GPIOA 的基地址：0x5000 0000

GPIOB 的基地址：0x5000 0040

GPIOC 的基地址：0x5000 0080

GPIOD 的基地址：0x5000 00C0

GPIOE 的基地址：0x5000 0100

GPIOF 的基地址：0x5000 0140

GPIOG 的基地址：0x5000 0180

GPIOH 的基地址：0x5000 01C0

3.3.1 GPIOx_PIR 输入状态寄存器

 表 3-2 GPIOx_PIR 输入状态寄存器⁴

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																	
R/W																	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
位名																	PXPIR15	PXPIR14	PXPIR13	PXPIR12	PXPIR11	PXPIR10	PXPIR9	PXPIR8	PXPIR7	PXPIR6	PXPIR5	PXPIR4	PXPIR3	PXPIR2	PXPIR1	PXPIR0																	

PXPIR<15:0>：端口 Px 输入数据

0 = 对应引脚为逻辑低电平

1 = 对应引脚为逻辑高电平

¹ 表中 x=A,B,C,D,E,F,G,H

² 上电复位过程中 PC6 会自动上拉，其它情况下 PC6 复位为 0。

³ GPIOC_PMOD 复位值为 0xFFFF EBFF。

⁴ 由于启动程序，GPIOB_PIR、GPIOC_PIR 会受输入的影响。若启动后引脚是悬空的，则 GPIOB_PIR 的 PBPIR3 位=1，GPIOC_PIR 的 PCPIR5=1，PCPIR6 不确定。

3.3.2 GPIOx_POR 输出状态寄存器

表 3-3 GPIOx_POR 输出状态寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0			
R/W	R/W																R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PXPOR15	PXPOR14	PXPOR13	PXPOR12	PXPOR11	PXPOR10	PXPOR9	PXPOR8	PXPOR7	PXPOR6	PXPOR5	PXPOR4	PXPOR3	PXPOR2	PXPOR1	PXPOR0			

PXPOR<15:0>: 端口 Px 输出数据

0 = 对应引脚为逻辑低电平

1 = 对应引脚为逻辑高电平

3.3.3 GPIOx_PUR 上拉使能寄存器

表 3-4 GPIOx_PUR 上拉使能寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0			
R/W	R/W																R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PXPUR15	PXPUR14	PXPUR13	PXPUR12	PXPUR11	PXPUR10	PXPUR9	PXPUR8	PXPUR7	PXPUR6	PXPUR5	PXPUR4	PXPUR3	PXPUR2	PXPUR1	PXPUR0			

PXPUR<15:0>: 端口 Px 上拉使能控制¹

0 = 对应引脚上拉功能禁止

1 = 对应引脚上拉功能使能

3.3.4 GPIOx_PDR 下拉使能寄存器

表 3-5 GPIOx_PDR 下拉使能寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0			
R/W	R/W																R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PXPDR15	PXPDR14	PXPDR13	PXPDR12	PXPDR11	PXPDR10	PXPDR9	PXPDR8	PXPDR7	PXPDR6	PXPDR5	PXPDR4	PXPDR3	PXPDR2	PXPDR1	PXPDR0			

PXPDR<15:0>: 端口 Px 下拉使能控制

0 = 对应引脚下拉功能禁止

1 = 对应引脚下拉功能使能

¹ 由于启动程序的影响, GPIOB_PXPUR3 将被置 1。

3.3.5 GPIOx_PODR 开漏输出控制寄存器

表 3-6 GPIOx_PODR 开漏输出控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0														
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W													
位名																	PXPODR15	PXPODR14	PXPODR13	PXPODR12	PXPODR11	PXPODR10	PXPODR9	PXPODR8	PXPODR7	PXPODR6	PXPODR5	PXPODR4	PXPODR3	PXPODR2	PXPODR1	PXPODR0														

PXPODR<15:0>: 端口 Px 开漏输出控制

0 = 对应引脚为推挽输出

1 = 对应引脚为开漏输出

3.3.6 GPIOx_PMOD 端口方向控制寄存器

 表 3-7 GPIOx_PMOD 端口方向控制寄存器¹²

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	RXPMD15 <1:0>	RXPMD14 <1:0>	RXPMD13 <1:0>	RXPMD12 <1:0>	RXPMD11 <1:0>	RXPMD10 <1:0>	RXPMD9 <1:0>	RXPMD8 <1:0>	RXPMD7 <1:0>	RXPMD6 <1:0>	RXPMD5 <1:0>	RXPMD4 <1:0>	RXPMD3 <1:0>	RXPMD2 <1:0>	RXPMD1 <1:0>	RXPMD0 <1:0>																

RXPMDy<1:0>: 端口 Pxy 端口输出模式控制

00 = 通用 IO 口输入模式

01 = 通用 IO 口输出模式

10 = 重映射 IO 口功能模式

11 = 模拟模式

3.3.7 GPIOx_OMOD 端口速度控制寄存器

表 3-8 GPIOx_OMOD 端口速度控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PXOMD15	PXOMD14	PXOMD13	PXOMD12	PXOMD11	PXOMD10	PXOMD9	PXOMD8	PXOMD7	PXOMD6	PXOMD5	PXOMD4	PXOMD3	PXOMD2	PXOMD1	PXOMD0																

PXOMDy: 端口 Pxy 端口输出速度控制 (y=15~0)

0 = 10MHZ

1 = 20MHZ

¹ 编程引脚 (SPDATA 和 SPCLK) 复位为重映射 IO 口功能模式, 所以 GPIOC_PMOD_PXPMD5<1:0>以及 GPIOC_PMOD_PXPMD6<1:0>复位值都是 10。

² 由于启动程序的影响, GPIOB_PMOD_PXPMD3<1:0>将被设置为 00。

3.3.8 GPIOx_LOCK 端口配置锁定寄存器

表 3-9 GPIOx_LOCK 端口配置锁定寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位名	PXLCKKEY<15:0>																PXLCK15	PXLCK14	PXLCK13	PXLCK12	PXLCK11	PXLCK10	PXLCK9	PXLCK8	PXLCK7	PXLCK6	PXLCK5	PXLCK4	PXLCK3	PXLCK2	PXLCK1	PXLCK0

PXLCKKEY<15:0>: 锁键。该位读数为 0。

将 0x5C5A 写入 PXLCKKEY<15:0>可以实现对 PXLCKy 的操作，否则禁止对 PXLCKy 的操作。

PXLCKy: 端口 Pxy 锁定控制 (x=A,B,C,D,E,F,G,H ;y=15~0)

0 = 对应引脚为未锁定

1 = 对应引脚为锁定配置

3.4 端口方向控制

通过 GPIOx_PMOD 端口方向控制寄存器可以设置 IO 口的使用状态。IO 口共有 4 中使用状态，分别为通用 IO 口输入，通用 IO 口输出，重映射 IO 口模式，和模拟引脚模式。每个 IO 口都可以独立设置。

当配置成通用 IO 口输入/通用 IO 口输出/重映射 IO 口时，可以通过 GPIOx_PIR 输入状态寄存器读出相应的 IO 口的值。

当配置成通用 IO 口输出时，可以通过 GPIOx_POR 输出状态寄存器的值设置对应 IO 口的输出电平状态。

当配置成重映射 IO 口模式时，IO 口状态由相应的外设控制，通过 GPIOx_RMP 可以设置相应引脚的重映射功能由什么外设控制。

当配置成模拟引脚功能时，可用于模拟引脚的输出，此时数字输入读为 0。

3.5 引脚通用输入配置

通过 GPIOx_PMOD 端口方向控制寄存器将 PXPMDy<1:0>设置成 00，可以将 IO 口设置成通用 IO 口输入模式，在该模式下可以通过 GPIOx_PIR 寄存器的 PXPIR<15:0>位读出相应 IO 口的状态。

3.6 引脚通用输出配置

通过 GPIOx_PMOD 端口方向控制寄存器将 PXPMDy<1:0>设置成 01，可以将 IO 口设置成通用 IO 口输出模式，在该模式下可以通过 GPIOx_POR 寄存器通过 PXPOR<15:0>设置相应 IO 口的输出状态 (0 为输出低电平，1 为输出高电平)。在通用 IO 口输出模式中，也可以通过 GPIOx_PIR 寄存器的 PXPIR<15:0>位读出相应 IO 口的状态。

3.7 引脚重映射功能配置

通过 GPIOx_PMOD 端口方向控制寄存器可以将 IO 口设置成引脚重映射模式，在该模式下 IO 口状态由相应的外设控制。外设功能使用时需要将相应的 IO 口设置成引脚重映射状态。通过 GPIOx_RMP 可以设置相应引脚的重映射功能由什么外设控制。在引脚重映射模式中，也可以通过 GPIOx_PIR 读出相应 IO 口的状态。

3.8 引脚上/下拉功能

每个引脚都可以独立配置上/下拉功能。通过 GPIOx_PUR 上拉使能寄存器可以独立设置相应引脚的上拉功能，通过 GPIOx_PDR 下拉使能寄存器可以独立设置相应引脚的下拉功能。

3.9 开漏输出控制功能

每个引脚都可以独立配置成开漏输出功能。通过 GPIOx_PODR 开漏输出控制寄存器可以独立设置相应引脚的开漏输出或者推挽输出模式。

3.10 工作模式掉电锁存功能

使能 PM_CTL0 寄存器的 IOLATCH 位，掉电后 IO 口工作模式保存为之前配置的形式。

3.11 引脚锁定

引脚锁定允许冻结任意 IO 的设置。通过 GPIOx_LOCK 端口配置锁定寄存器可以实现对任意引脚的锁定，当一个引脚上执行了锁定后，在解除锁定或在下一次复位前，将不能再更改相应引脚的配置。

当 PXLCKy 对应位置 1 时，Pxy 引脚的 GPIOx_PIR、GPIOx_POR、GPIOx_PUR、GPIOx_PDR、GPIOx_PODR、GPIOx_PMOD、GPIOx_OMOD、GPIOx_RMP 寄存器对应的 y 引脚的配置将不可更改，直到将 PXLCKy 对应位清零时或者复位后才能对 GPIOx_PIR、GPIOx_POR、GPIOx_PUR、GPIOx_PDR、GPIOx_PODR、GPIOx_PMOD、GPIOx_OMOD、GPIOx_RMP 寄存器对应的 y 引脚控制位进行修改配置。

4 引脚重映射 (GPIO remap)

4.1 概述

单片机提供了灵活的重映射设置功能。通过寄存器 GPIOx_RMP 重映射寄存器可以设置引脚的重映射功能。当使能选择外设功能引脚映射时，引脚为相应外设引脚的输入输出数据。

1

4.2 引脚重映射相关寄存器

表 4-1 GPIOx 引脚重映射相关寄存器²

偏移地址	寄存器	访问	功能描述	复位值
0x020	GPIOx_RMPL	R/W	Px 口低位重映射控制寄存器	0x0000 0000
0x024	GPIOx_RMPH	R/W	Px 口高位重映射控制寄存器	0x0000 0000
0x02C	GPIOx_CTMDCTL0	R/W	Px 端口触摸模式控制寄存器 0	0x0000 0000
0x030	GPIOx_CTMDCTL1	R/W	Px 端口触摸模式控制寄存器 1	0x0000 0000

GPIOA 的基地址: 0x5000 0000

GPIOB 的基地址: 0x5000 0040

GPIOC 的基地址: 0x5000 0080

GIPIOD 的基地址: 0x5000 00C0

GPIOE 的基地址: 0x5000 0100

GPIOF 的基地址: 0x5000 0140

GPIOG 的基地址: 0x5000 0180

GPIOH 的基地址: 0x5000 01C0

4.2.1 GPIOx_RMPLPx 低 8 位引脚重映射控制寄存器

表 4-2 GPIOx_RMP Px 重映射控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W																						
位名	PXRMP7 <3:0>			PXRMP6 <3:0>			PXRMP5 <3:0>			PXRMP4 <3:0>			PXRMP3 <3:0>			PXRMP2 <3:0>			PXRMP1 <3:0>			PXRMP0 <3:0>										

PxRMPy<3:0>: 端口 Pxy 引脚重映射选择 (y=0~7)

0000 = 特殊功能 (CLKOUT/RTC/BKP)

0001 = T0/T1/T2/T3/T4

0010 = T5/T6

0011 = T9/T20/T21/T23

0100 = T14/T15/18/19/22/QEIO

¹ 编程引脚复位为重映射引脚，且重映射为编程引脚功能。

² 本章除特殊说明外，x = A,B,C,D,E,F,G,H; y = 0~15。

- 0101 = USART0/1/2
- 0110 = USART3
- 0111 = SPI0/1
- 1000 = I2C0/1/2
- 1001 = CAN0/1/FLT
- 1010 = 复合功能 1(详见重映射表)
- 1011 = 复合功能 2(详见重映射表)
- 1100 = CFG1
- 1101 = 复合功能 3(详见重映射表)
- 1110 = 复合功能 4(详见重映射表)
- 1111 = 保留

4.2.2 GPIOx_RMP Px 高 8 位引脚重映射控制寄存器

表 4-3 GPIOx_RMP Px 重映射控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位名	PXRMP15 <3:0>			PXRMP14 <3:0>			PXRMP13 <3:0>			PXRMP12 <3:0>			PXRMP11 <3:0>			PXRMP10 <3:0>			PXRMP9 <3:0>			PXRMP8 <3:0>										

PxRMPy<3:0>: 端口 Pxy 引脚重映射选择 (y=8~15)

- 0000 = 特殊功能 (CLKOUT/RTC/BKP)
- 0001 = T0/T1/T2/T3/T4
- 0010 = T5/T6
- 0011 = T9/ T20/T21/T23
- 0100 = T14/T15/18/19/22/QEIO
- 0101 = USART0/1/2
- 0110 = USART3
- 0111 = SPI0/1
- 1000 = I2C0/1/2
- 1001 = CAN0/1FLT
- 1010 = 复合功能 1(详见重映射表)
- 1011 = 复合功能 2(详见重映射表)
- 1100 = CFG1
- 1101 = 复合功能 3(详见重映射表)
- 1110 = 复合功能 4(详见重映射表)
- 1111 = 保留

5 振荡器 (OSC)

5.1 概述

单片机提供 6 种基础时钟振荡器选择, 分别为内部高频 (INTHF)、内部低频 (INTLF)、外部高频 (EXTHF)、外部低频 (EXTLF)、内部的 PLL 和低功耗 4M 时钟 LP4M。内部的 PLL 可以将内部高频 (INTHF) 和外部高频 (EXTHF) 的输出时钟倍频, 提供更高频率的工作时钟选择, 作为系统和外设工作需要的基础时钟。通过寄存器配置, 可以从 6 种振荡器中得到 4 种系统和外设运行时需要的时钟源: 系统主时钟 (SCLK)、低频外设时钟 (LFCLK)、高频外设时钟 (HFCLK) 和 48MHz 时钟 (CK48M) 满足不同的需要。此外, 内部低频振荡器还可以直接用于看门狗定时器、时钟故障检测或是其他低功耗外设的时钟。

系统复位后, INTHF 振荡器被选为系统时钟, 当系统时钟需要切换时, 只有当目标时钟源准备就绪 (时钟源稳定, 延迟标志位置 1), 才会发生时钟源的切换。

振荡器模块具有以下特征:

- 提供 6 种振荡源选择
 - 内部高频振荡器 INTHF (16MHz);
 - 内部低频振荡器 INTLF (32KHz);
 - 外接高频振荡器 EXTHF (4~48MHz);
 - 外部低频振荡器 EXTLF (32.768KHz 的晶振);
 - 内部 PLL (最高 400MHz)
 - 内部低功耗振荡器 LP4M (4MHz)
- 可产生 4 种时钟源
 - 系统主时钟 SCLK
由内部高频 (INTHF)、内部低频 (INTLF)、外部高频 (EXTHF)、外部低频 (EXTLF)、PLL 倍频或内部 4M 振荡器 (LP4M) 产生。
 - 高频外设时钟 HFCLK
由内部高频 (INTHF)、外部高频 (EXTHF)、PLL 倍频或内部 4M 振荡器 (LP4M) 产生。
 - 低频外设时钟 LFCLK
由内部低频 (INTLF) 或外部低频 (EXTLF) 产生。
 - 时钟 CK48M
由内部高频 (INTHF)、外部高频 (EXTHF) 或 PLL 倍频产生
- 外部时钟启动/切换保障
- 带时钟同步功能
- 带时钟故障检测功能

5.2 振荡器结构框图

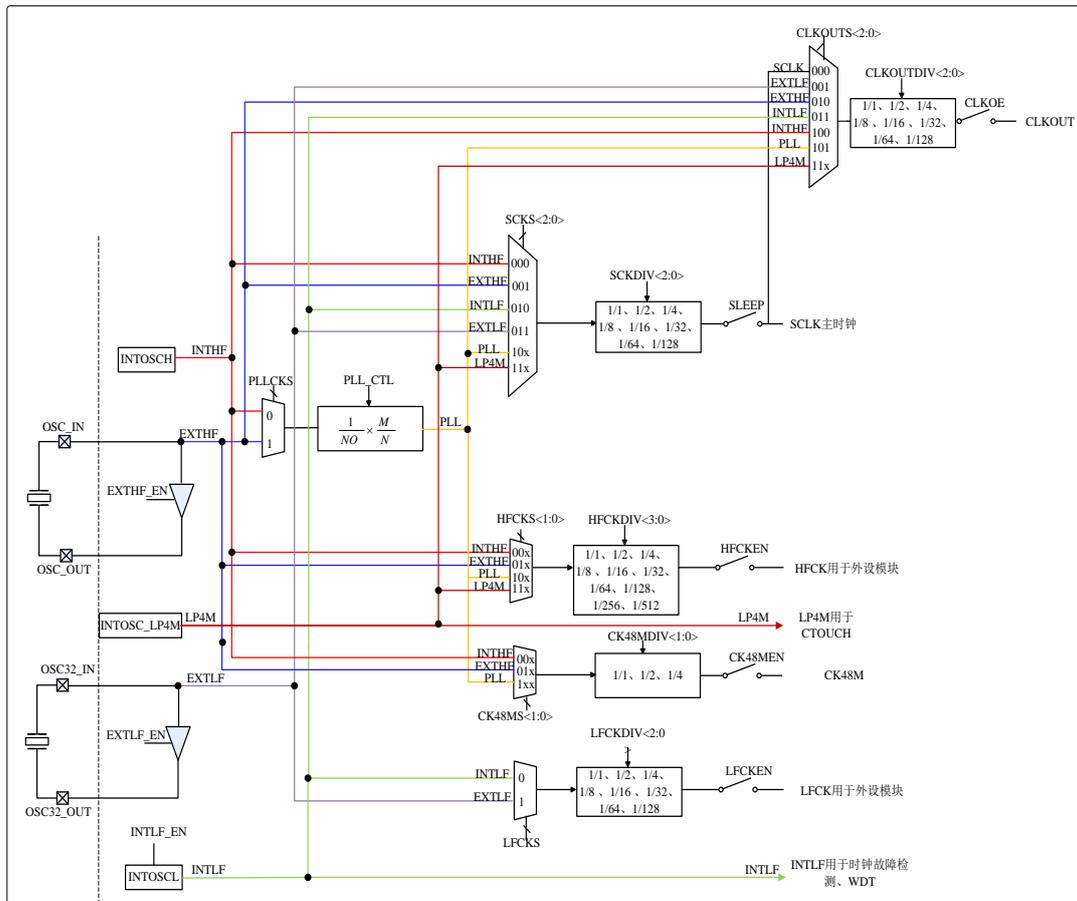


图 5-1 振荡器结构框图¹²³

¹ 内部低频可直接用于时钟故障检测、看门狗、上电复位延迟定时器 (PWRT) 以及低功耗外设。
² 当任意时钟源选择 EXTHF 作为时钟源或软件使能时 EXTHF_EN 使能
³ 当任意时钟源选择 EXTLF 作为时钟源或软件使能时 EXTLF_EN 使能

5.3 振荡器相关寄存器

表 5-1 振荡器相关寄存器 1

偏移地址	寄存器	访问	功能描述	复位值
0x000	OSC_CTL0	R/W	振荡器控制寄存器 0	0x0004 0000
0x004	OSC_CTL1	R/W	振荡器控制寄存器 1	0x0000 0000
0x008	OSC_INT	R/W	振荡器中断控制寄存器	0x0000 0000
0x00C	OSC_CTL2	R/W	振荡器控制寄存器 2	0x0000 0000
0x010	OSC_HFOSCCAL0	R/W	高频振荡器校准寄存器 0	0x0548 0040
0x014	OSC_HFOSCCAL1	R/W	高频振荡器校准寄存器 1	0x0000 0000

基地址：0x4000 0000

表 5-2 振荡器相关寄存器 2

偏移地址	寄存器	访问	功能描述	复位值
0x000	PLL_CTL	R/W	PLL 控制寄存器	0x0001 0010

基地址：0x4000 2580

5.3.1 OSC_CTL0 振荡器控制寄存器 0

表 5-3 OSC_CTL0 振荡器控制寄存器 0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W						R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W		R/W			R/W	R/W	R/W	R/W	R/W
位名						LFCKDIV<2:0>			HFCKDIV<3:0>				SCKDIV<2:0>						PLCKS			HFCKS<2:0>		KFCKEN		LFCKS		LFCKEN	SCKS<2:0>		PWREN		

LFCKDIV<2:0>：低频外设时钟分频选择位

- 000 = 1/1 分频
- 001 = 1/2 分频
- 010 = 1/4 分频
- 011 = 1/8 分频
- 100 = 1/16 分频
- 101 = 1/32 分频
- 110 = 1/64 分频
- 111 = 1/128 分频

HFCKDIV<3:0>：高频外设时钟分频选择位

- 0000 = 1/1 分频
- 0001 = 1/2 分频
- 0010 = 1/4 分频
- 0011 = 1/8 分频
- 0100 = 1/16 分频（默认）
- 0101 = 1/32 分频
- 0110 = 1/64 分频
- 0111 = 1/128 分频

1000 = 1/256 分频

其他 = 1/512 分频

SCKDIV<2:0>: 主时钟 SCLK 分频选择位

000 = 1/1 分频

001 = 1/2 分频

010 = 1/4 分频

011 = 1/8 分频

100 = 1/16 分频 (默认)

101 = 1/32 分频

110 = 1/64 分频

111 = 1/128 分频

PLLCKS: PLL 输入时钟源选择位

0 = 选择 INTHF 作为 PLL 输入时钟

1 = 选择 EXTHF 作为 PLL 输入时钟

HFCKS<2:0>: 高频外设时钟源选择位

00x = 选择 INTHF 作为 HFCLK 时钟

01x = 选择 EXTHF 作为 HFCLK 时钟

10x = 选择 PLL 输出作为 HFCLK 时钟

11x = 选择 LP4M 输出作为 HFCLK 时钟

HFCKEN: 高频外设时钟允许位

0 = HFCLK 时钟信号禁止

1 = HFCLK 时钟信号允许

LFCKS: 低频外设时钟源选择位

0 = 选择 INTLF 作为 LFCLK 时钟

1 = 选择 EXTLF 作为 LFCLK 时钟

LFCKEN: 低频外设时钟允许位

0 = LFCLK 时钟信号禁止

1 = LFCLK 时钟信号允许

SCKS<2:0>: SCLK 主时钟选择位

000 = 选择 INTHF 作为系统时钟

001 = 选择 INTLF 作为系统时钟

010 = 选择 EXTHF 作为系统时钟

011 = 选择 EXTLF 作为系统时钟

10x = 选择 PLL 输出作为系统时钟

11x = 选择 LP4M 输出作为系统时钟

PMWREN: 备份区寄存器读写允许位

0 = 备份区不可读写

1 = 备份区可读写

5.3.2 OSC_CTL1 振荡器控制寄存器 1

表 5-4 OSC_CTL1 振荡器控制寄存器 1

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	1 18	1 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0							1	1	0											0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	W	W	W							R/W	R/W	R/W											R/W	R/W	R/W	R/W	R/W	R/W
位名	CLKOUTDIV <2:0>			CLKOE	CLKOUTS<2:0>									FSCMGAP <1:0>		FSCM										CK48DIV <1:0>		CK48MS <2:0>		CK48MEN		

CLKOUTDIV<2:0>: 从引脚 CLKOUT 输出时钟分频选择位

- 000 = 1/1 分频
- 001 = 1/2 分频
- 010 = 1/4 分频
- 011 = 1/8 分频
- 100 = 1/16 分频
- 101 = 1/32 分频
- 110 = 1/64 分频
- 111 = 1/128 分频

CLKOE: 时钟输出使能位

- 0 = 禁止时钟从 IO 口 (CLKOUT) 输出
- 1 = 允许时钟从 IO 口 (CLKOUT) 输出

CLKOUTS<2:0>: 从引脚 CLKOUT 输出时钟选择位

- 000 = 选择 SCLK 作为输出时钟
- 001 = 选择 EXTIF 作为输出时钟
- 010 = 选择 EXTHF 作为输出时钟
- 011 = 选择 INTIF 作为输出时钟
- 100 = 选择 INTHF 作为输出时钟
- 101 = 选择 PLL 作为输出时钟
- 11x = 选择 LP4M 作为输出时钟

FSCMGAP<1:0>: 时钟故障检测间隔设置, 这里需要设置

- 00 = INTIF/16
- 01 = INTIF/32
- 10 = INTIF/64
- 11 = INTIF/128

FSCM: 时钟故障检测使能

- 0 = 时钟故障检测禁止
- 1 = 时钟故障检测使能

CK48DIV<1:0>: CK48M 时钟分频选择位

- 00 = 1/1 分频
- 01 = 1/2 分频
- 1x = 1/4 分频

CK48MS<2:0>: 48M 时钟输入时钟源选择位

- 00x = 选择 INTHF 作为 CK48M 输入时钟
- 01x = 选择 EXTHF 作为 CK48M 输入时钟

1xx = 选择 PLL 作为 CK48M 输入时钟

CK48MEN: 48M 时钟输入时钟允许位

0 = CK48M 时钟信号禁止

1 = CK48M 时钟信号使能

5.3.3 OSC_INT 振荡器中断控制寄存器

表 5-5 OSC_INT 振荡器中断控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																		R/W		R/W	R/W		R/W	R/W	R/W	R/W	R/W						
位名																		LP4MIE		PLLIE	EHFIE	ELFIE	IHFIE	ILFIE	CKFIF	LP4MIF		PLLIF	EHFIF	ELFIF	IHFIF	ILFIF	

LP4MIE: LP4M 中断使能位

0 = 中断禁止

1 = 中断使能

PLLIE: PLL 中断使能位

0 = 中断禁止

1 = 中断使能

EHFIE: 外部高频中断使能位

0 = 中断禁止

1 = 中断使能

ELFIE: 外部低频中断使能位

0 = 中断禁止

1 = 中断使能

IHFIE: 内部高频中断使能位

0 = 中断禁止

1 = 中断使能

ILFIE: 内部低频中断使能位

0 = 中断禁止

1 = 中断使能

CKFIF: 时钟故障标志位

0 = 没有时钟故障

1 = 检测到时钟故障

LP4MIF: LP4M 中断标志位, 发生时钟故障时清零该标志位

0 = LP4M 未锁定

1 = LP4M 锁定

PLLIF: PLL 中断标志位, 发生时钟故障或 PDRST 置 1 时清零该标志位

0 = PLL 未锁定

1 = PLL 锁定

EHFIF: 外部高频中断标志位, 发生时钟故障时清零该标志位

0 = 外部高频不稳定

1 = 外部高频稳定

ELFIF: 外部低频中断标志位, 发生时钟故障时清零该标志位

0 = 外部低频不稳定

1 = 外部低频稳定

IHFIF: 内部高频中断标志位

0 = 内部高频不稳定

1 = 内部高频稳定

ILFIF: 内部低频中断标志位

0 = 内部低频不稳定

1 = 内部低频稳定

5.3.4 OSC_CTL2 振荡器控制寄存器 2

表 5-6 OSC_CTL2 振荡器控制寄存器 2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W										R/W			R/W	R/W	R/W	R/W	R/W	R/W
位名					PLLDLY <3:0>			EHFDLY <3:0>			ELFDLY <3:0>															LP4MSWEN			PLLSWEN	EXTHFSWEN	EXTLFSWEN	INTHFSWEN	INTLFSWEN	

PLLDLY<3:0>: PLL 启动延迟控制 (时钟为 PLL 的参考时钟)

0000 = 延迟 64 个时钟

0001 = 延迟 128 个时钟

0010 = 延迟 256 个时钟

0011 = 延迟 512 个时钟

0100 = 延迟 1024 个时钟

0101 = 延迟 2048 个时钟

0110 = 延迟 4096 个时钟

0111 = 延迟 8192 个时钟

1000 = 延迟 16384 个时钟

其他 = 延迟 32768 个时钟

EHFDLY<3:0>: 外部高频启动延迟控制

0000 = 延迟 256 个时钟

0001 = 延迟 512 个时钟

0010 = 延迟 1024 个时钟

0011 = 延迟 2048 个时钟

0100 = 延迟 4096 个时钟

0101 = 延迟 8192 个时钟

0110 = 延迟 16384 个时钟

0111 = 延迟 32768 个时钟

其他 = 延迟 65536 个时钟

ELFDLY<3:0>: 外部低频启动延迟控制

0000 = 延迟 256 个时钟

0001 = 延迟 512 个时钟

0010 = 延迟 1024 个时钟

0011 = 延迟 2048 个时钟

- 0100 = 延迟 4096 个时钟
- 0101 = 延迟 8192 个时钟
- 0110 = 延迟 16384 个时钟
- 0111 = 延迟 32768 个时钟
- 其他 = 延迟 65536 个时钟

LP4MSWEN: LP4M 软件使能位

- 0 = 无效
- 1 = 使能 LP4M 工作

PLLSWEN: PLL 软件使能位

- 0 = 无效
- 1 = 使能 PLL 工作

EXTHFSWEN: 外部高频振荡器软件使能位

- 0 = 无效
- 1 = 使能外部高频振荡器工作

EXTLFSWEN: 外部低频振荡器软件使能位

- 0 = 无效
- 1 = 使能外部低频振荡器工作

INTHFSWEN: 内部高频振荡器软件使能位

- 0 = 无效
- 1 = 使能内部高频振荡器工作

INTLFSWEN: 内部低频振荡器软件使能位

- 0 = 无效
- 1 = 使能内部低频振荡器工作

5.3.5 OSC_HFOSCCAL0 高频振荡器校准寄存器 0

表 5-7 OSC_HFOSCCAL0 高频振荡器校准寄存器 0

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	SCALE32X	SCALE16X <4:0>				SCALE8X	SCALE4X	SCALE2X	SCALE1X	SCALEDIV2	SCALEDIV4	SCALEDIV8	SCALEDIV16	SCALEDIV32									PTAT <4:0>				CTAT <4:0>					

SCALE32X: 零温漂的绝对值校准权重位

- 0 = 32uA 电流档关闭
- 1 = 32uA 电流档开启

SCALE16X<4:0>: 零温漂的绝对值校准权重位

- 00000 = 5 个 16uA 档关闭
- 11111 = 5 个 16uA 档开启 (每一位单独控制一个 16uA)

SCALE8X: 零温漂的绝对值校准权重位

- 0 = 8uA 电流档关闭
- 1 = 8uA 电流档开启

SCALE4X: 零温漂的绝对值校准权重位

- 0 = 4uA 电流档关闭

- 1 = 4uA 电流档开启
- SCALE2X: 零温漂的绝对值校准权重位
- 0 = 2uA 电流档关闭
- 1 = 2uA 电流档开启
- SCALE1X: 零温漂的绝对值校准权重位
- 0 = 1uA 电流档关闭
- 1 = 1uA 电流档开启
- SCALEDIV2: 零温漂的绝对值校准权重位
- 0 = 1/2uA 电流档关闭
- 1 = 1/2uA 电流档开启
- SCALEDIV4: 零温漂的绝对值校准权重位
- 0 = 1/4uA 电流档关闭
- 1 = 1/4uA 电流档开启
- SCALEDIV8: 零温漂的绝对值校准权重位
- 0 = 1/8uA 电流档关闭
- 1 = 1/8uA 电流档开启
- SCALEDIV16: 零温漂的绝对值校准权重位
- 0 = 1/16uA 电流档关闭
- 1 = 1/16uA 电流档开启
- SCALEDIV32: 零温漂的绝对值校准权重位
- 0 = 1/32uA 电流档关闭
- 1 = 1/32uA 电流档开启
- PTAT<4:0>: 内部高频振荡器的正温调节位
- 00000 = 5 个正温漂档关闭
- 11111 = 5 个正温漂档开启 (权重从高到低分别是 1,1/2,1/4,1/8,1/16)
- CTAT<4:0>: 内部高频振荡器的负温调节位
- 00000 = 5 个负温漂档关闭
- 11111 = 5 个负温漂档开启 (权重从高到低分别是 1,1/2,1/4,1/8,1/16)

5.3.6 OSC_HFOSCCAL1 高频振荡器校准寄存器 1

表 5-8 OSC_HFOSCCAL1 高频振荡器校准寄存器 1

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值																																
R/W																																
位名																									SELGAIN<1:0>	HSFSEN	HSXTALEXEN	HL5W				

SELGAIN<1:0>: 电流增益选择位

表 5-9 电流增益选择表¹

Frequency(M)	VDD(min)	5p	10p	20p	39p
4	1.62	00	00	00	00
8	1.62	00	00	00	X
16	1.62	00	00	00	X
20	1.8	X	00	00	X
32	2.2	X	00	01	X
40	2.6	X	01	10	X
48	2.6	X	10	11	X

HSFSEN: 高速选择位

0 = 频率大于等于 20MHz

1 = 频率小于 20MHz

HSXTALEXEN: 外部高频晶振输入方法选择位

0 = 使用晶振频率

1 = 外部输入频率

HLSW: 反馈电阻调节位

0 = 频率大于 10MHz

1 = 频率小于 10MHz

5.3.7 PLL_CTL PLL 控制寄存器

表 5-10 PLL_CTL PLL 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
R/W											R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名											OD<1:0>	N<3:0>			M<13:0>										PDRST							

OD<1:0>: PLL 输出分频控制, 用于产生 PLL 倍频的 NO 值

$$NO=2^{OD<1:0>} \quad \text{公式 5-1}$$

N<3:0>: PLL 输入驱动控制, 用于产生 PLL 倍频的 N 值, 修改该位需将 PDRST 置 1。

$$N=N<3:0> \quad \text{公式 5-2}$$

M<13:0>: PLL 反馈驱动控制, 用于产生 PLL 倍频的 M 值, 修改该位需将 PDRST 置 1。

$$M=M<13:0> \quad \text{公式 5-3}$$

PDRST: PLL 复位控制, 当 PLL 的参考频率/N<3:0>/M<13:0>/BP (1 到 0 变化) 后都需要将该位置 1 初始化 PLL, 该位自动清零。

0 = 无作用

1 = PLL 复位控制

PLL 倍频计算公式为:

$$PLLCLK = \frac{F_{REF}}{NO} \times \frac{M}{N} \quad \text{公式 5-4}$$

¹ X 表示禁止这种应用。

5.4 内部高频振荡器 INTHF

内部高频振荡器 INTHF 可以为芯片提供 16MHz 高频时钟。内部高频振荡器 INTHF 可以作为 PLL 输入、作为系统主时钟源 SCLK 的输入和作为外设高频时钟源 HFCLK 的输入。

当任意时钟选择内部高频振荡器 INTHF 作为时钟源时，内部高频振荡器使能。内部高频振荡器也可以通过软件使能位 INTHFSWEN 进行独立使能。

内部高频使能后，会立即产生振荡器稳定信号将标志位 IHFIF 置 1。

将 OSC_INT 寄存器中的 IHFIE 位置 1，可以使能内部高频振荡器中断。内部高频振荡器 INTHF 使能后，会立即产生振荡器稳定信号将标志位 IHFIF 置 1，产生中断。

5.5 内部低频振荡器 INTLF

内部低频振荡器 INTLF 可以为芯片提供 32KHz 的低频时钟。内部低频振荡器 INTLF 可以作为系统主时钟源 SCLK 的输入和作为外设低频时钟源 LFCLK 的输入，此外，内部低频时钟还可以作为独立时钟信号供时钟故障检测、WDT、上电延时定时器 (PWRT)、低功耗外设使用。

当任意时钟选择内部低频振荡器 INTLF 作为时钟源时，内部低频振荡器使能。内部低频振荡器也可以通过软件使能位 INTLFSWEN 或通过 PM_CTL0 的 LSIEN 进行独立使能。需要注意的是，如果需要掉电模式下使用内部低频振荡器 INTLF，需要将 PM_CTL0 的 LSIEN 置 1。

将 OSC_INT 寄存器中的 ILFIE 位置 1，可以使能内部低频振荡器中断。内部低频振荡器 INTLF 使能后，会立即产生振荡器稳定信号将标志位 ILFIF 置 1（除通过 PM_CTL0 寄存器中的 LSIEN 位使能外），产生中断。

5.6 外部高频振荡器 EXTHF

外部高频振荡器 EXTHF 可以为芯片提供 4~48MHz 可选的高频时钟。外部高频振荡器 EXTHF 可以作为 PLL 输入、作为系统主时钟源 SCLK 的输入和作为外设高频时钟源 HFCLK 的输入。

外部高频振荡器原理如下图所示，引脚 OSC_IN 和引脚 OSC_OUT 可以接外部标准晶体、陶瓷谐振器或外接 4MHz~48MHz 的时钟为时钟源。

当任意时钟选择外部高频振荡器 EXTHF 作为时钟源时，都会使能外部高频振荡器 EXTHF。外部高频振荡器也可以通过软件使能位 EXTHFSWEN 进行独立使能。

外部高频使能后，经过一段延时时间后会产生振荡器稳定信号将标志位置 1。延时时间由固定延时和可编程延时决定。固定延时为 xx ms，可编程延时为 0~512 周期。通过 EHF DLY<3:0>设置可编程延时的时间，最终延时为两个部分的延时的和。

将 OSC_INT 寄存器中的 EHFIE 位置 1，使能外部高频振荡器中断，当使能外部高频振荡器 EXTHF 后，EXTHF 输出稳定且经过启动延时后，EHFIF 中断标志位置 1，将会产生中断。

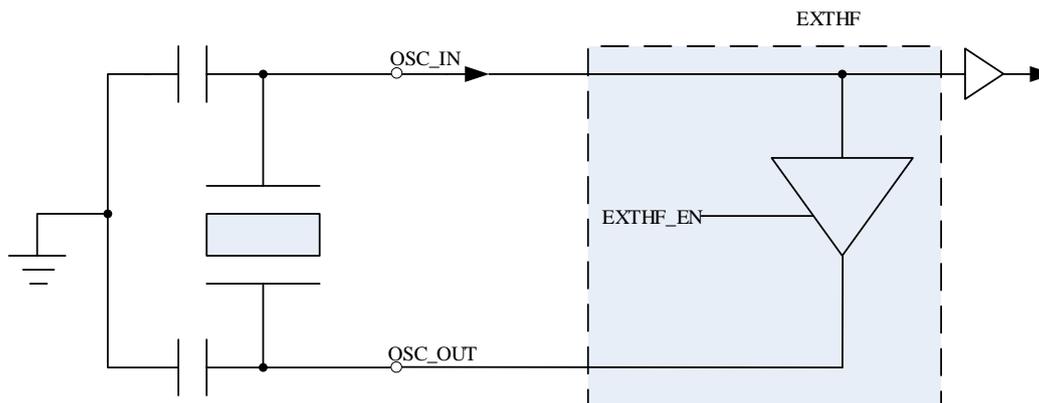


图 5-2 外部高频振荡器原理图

内部说明：内置两个 HSE 模块，可以通过 PM_CAL1 寄存器的 HSE_SEL 位进行选择：

HSE_SEL：外部高频晶振模块选择位

0 = 选择外部高频晶振 1，端口引脚位于 PD14/PD15

1 = 选择外部高频晶振 2，端口引脚位于 PD9/PD10

5.7 外部低频振荡器 EXTLE

外部低频振荡器 EXTLE 可以为芯片提供 32.768KHz 的时钟。外部低频振荡器 EXTLE 可以作为系统主时钟 SCLK 源的输入和作为外设低频时钟源 LFCLK 的输入。

如下图所示，引脚 OSC32_IN 和引脚 OSC32_OUT 接外部钟表石英晶体，主要是以 32.768KHz 的晶体提供时钟源。

当任意时钟选择 EXTLE 作为时钟源时，都会使能 EXTLE。外部低频振荡器也可以通过软件使能位 EXTLFSWEN 或通过 PM_CTL0 的 LSEEN 进行独立使能。需要注意的是，如果需要掉电模式下使用 EXTLE，需要将 PM_CTL0 的 LSEEN 置 1。

外部低频使能后，经过一段延时时间后会产生振荡器稳定信号将标志位置 1。延时时间由固定延时和可编程延时决定。固定延时为 2.5 ms，可编程延时为 0~512 周期。通过 ELFDLY<3:0>设置可编程延时的时间，最终延时为两个部分的延时的和。

将 OSC_INT 寄存器中的 ELFIE 位置 1，使能外部低频振荡器 EXTLE 中断，当使能 EXTLE 后（除通过 PM_CTL0 寄存器中的 LSEEN 位使能），EXTLE 输出稳定且经过启动延时后，ELFIF 中断标志位置 1，将会产生中断。

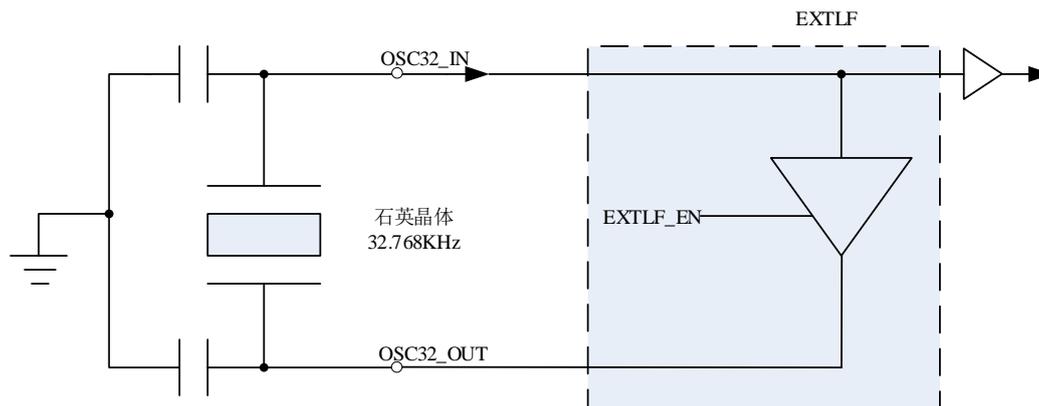


图 5-3 外部低频振荡器原理图

5.8 PLL 使用

芯片提供内置 PLL，可以为芯片提供高达 400MHz 的时钟。PLL 可以通过 PLLCKS 选择内部高频时钟 INTHF 和外部高频时钟 EXTHF 作为输入(F_{REF})，通过 PLL_CTL 的 N<3:0>、M<13:0>和 OD<1:0>可以获得需要的倍频，通过 PDRST 可以初始化 PLL。

PLL 的输出可以作为系统主时钟源 SCLK、外设高频时钟源 HFCLK 和 48M 时钟源的输入。

当任意时钟选择 PLL 作为时钟源输入时，PLL 使能。PLL 也可以通过 OSC_CTL2 软件使能位 PLLSWEN 进行独立使能。

PLL 的输出计算公式为：

$$PLLCLK = F_{REF} \times \frac{1}{NO} \times \frac{M}{N} \quad \text{公式 5-5}$$

PLL 使用过程中需要满足以下条件：

- $1MHz \leq \frac{F_{REF}}{N} \leq 50MHz$
- $200MHz \leq F_{REF} \times \frac{M}{N} \leq 400MHz$
- $M \geq 4$
- $N \geq 1$

在使用 PLL 时，发生以下任意情况都需要软件将 PDRST 置 1，初始化 PLL，否则修改无效（注意：先修改后将 PDRST 置 1）：

- PLL 由不使能到使能
- PLL 的输入参考频率改变
- N<3:0>改变
- M<13:0>改变

PLL 使能后或将 PDRST 置 1 后，经过一段延时时间后会产生振荡器稳定信号将标志位置 1，只要当稳定标志位为 1 时才会使用 PLL 的输出。通过 PLLDLY<3:0>设置可编程延时的时间，延时时间为可编程延时 64~32768 周期。建议在使用 PLL 的时候等待稳定标志位为 1 时再切换使用 PLL，确保使用的是切换后的时钟。

当使用 PLL 时，如果希望修改 PLL 的配置，需要先将切换成其他可用时钟源(如 INTHF)或暂停模块工作，等待 PLL 稳定后再将切换成 PLL 源，否则由于 PLL 切换过程中可能存在

不确定状态可能会导致运行故障，尤其是当 SCLK 选择 PLL 作为时钟源时。

将 OSC_INT 寄存器中的 PLLIE 位置 1，使能 PLL 中断，当使能 PLL 后，PLL 输出经过启动延时后，PLLIF 中断标志位置 1，将会产生中断。

5.9 内部低功耗 4M 振荡器 LP4M

芯片提供一个低功耗 4MHz 的振荡器 LP4M。内部低功耗 4M 振荡器 LP4M 可以作为系统主时钟 SCLK 源的输入、作为外设高频时钟源 HFCLK 的输入。

当任意时钟选择 LP4M 作为时钟源时，LP4M 使能。内部低功耗 4M 振荡器也可以通过软件使能位 LP4MSWEN 进行独立使能。

将 OSC_INT 寄存器中的 LP4MIE 位置 1，可以使能 LP4M 中断。当中断使能时，LP4MIF 在内部低功耗 4M 振荡器 LP4M 使能时置 1，产生中断。

5.10 时钟同步

SCLK、HFCLK 和 LFCLK 时钟都可以由不同时钟源提供，SCLK 提供硬件同步，HFCLK 和 LFCLK 需要软件规避。

- SCLK 切换同步：

当一个时钟源切换到另一个时钟源时，切换硬件同步以避免发生时间竞争。当选择一个新的时钟源，会发生以下过程：

- (1) 写 OSC_CTL 寄存器改变 SCLK 时钟源；
- (2) 在新时钟源稳定标志位置 1 前使用旧时钟源继续工作；
- (3) 当新时钟源稳定，在当前时钟的下一个完整周期的下降沿更新缓存 SCKS_BUF<1:0>；
- (4) 时钟 CLK 保持为低电平，时钟切换电路等待新的时钟的下降沿，允许时钟输出；
- (5) CLK 与新的时钟连接，完成时钟切换。

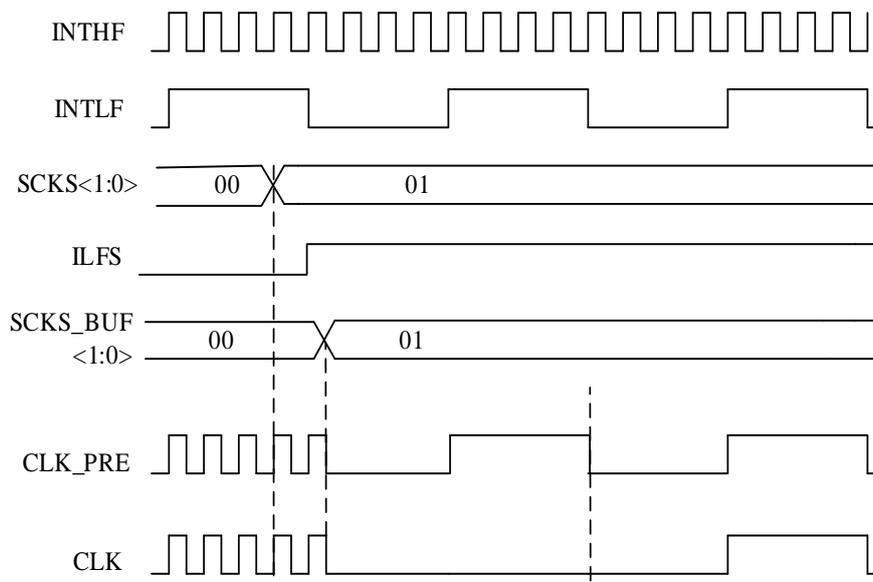


图 5-4 时钟同步时序图

- HFCLK 和 LFCLK 时钟源切换：

建议在使用前先切换好，使用过程中不要切换，或者在切换的过程中先将正在使用的模

块关闭，切换成功后重新使能，否则可能会出现时钟毛刺，影响外设模块运行。

5.11 外部时钟故障检测

当系统时钟 SCLK 直接或间接使用外部振荡器时，故障保护时钟监视（FSCM）能使器件在外部振荡器发生故障时继续运行，时钟故障检测需要使用内部低频时钟，其可以检测出外部振荡器稳定后任何时刻发生的振荡器故障（外部振荡器稳定前不会）。

将 OSC_CTL1 寄存器的 FSCM 位置 1 使能时钟故障检测功能，通过 OSC_CTL1 寄存器的 FSCMGAP<1:0>寄存器可以设置检测间隔，提高检测灵敏度。

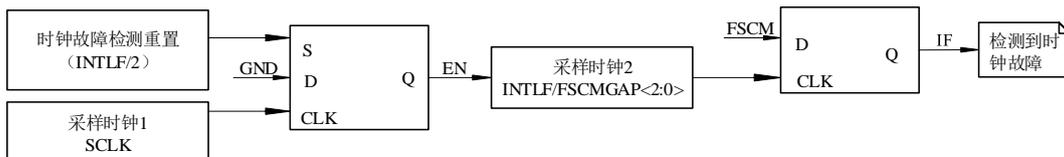


图 5-5 时钟故障检测原理图

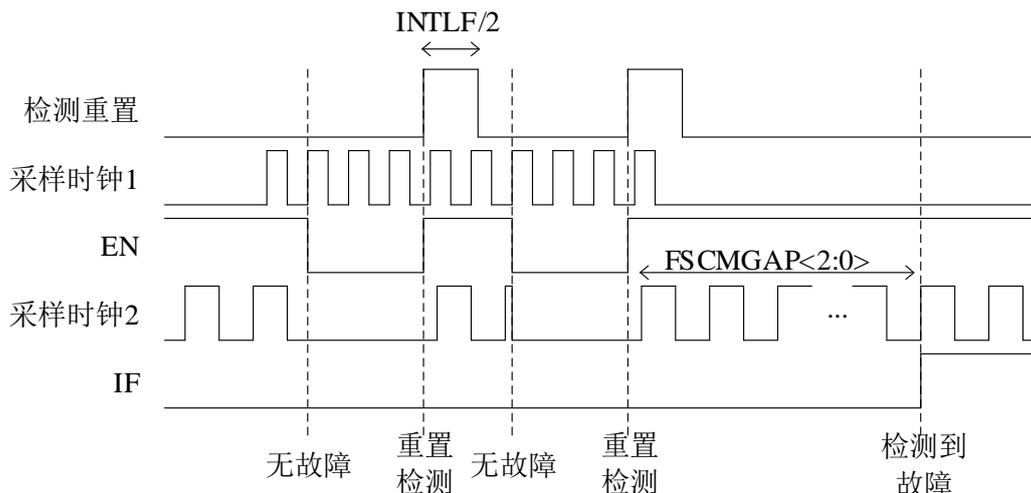
5.11.1 时钟故障保护检测

时钟故障检测模块通过使用外部时钟清除内部检测时钟的采样值的方式来监控外部时钟的工作。

如下图所示为时钟故障检测的时序图。检测重置装置会不断使能检测计数，当外部时钟有效时，则会不断清除检测计数，只有当外部时钟因故障停止后，检测计数计数到一定的值就会则系统检测到时钟故障。

检测重置装置在固定时间内重置（高电平有效）一次 EN 信号，使其保持为高电平（使能状态），而当前采样时钟 1（SCLK）没有故障，则会在采样时钟 1 的上升沿时将 EN 清零，表明当前时钟运行正常。若 SCLK 出现异常停止的情况时，EN 会持续保持高电平。检测重置装置在检测到 EN 为高电平时在间隔最长 2 个 INTLF 的时间间隔内会重置一次，重置时间最长为 2 个 INTLF 的时钟长度。

EN 为采样时钟 2（时钟故障间隔采样时钟）使能，EN 为低时清零采样状态，EN 为高电平时故障间隔采样时钟开始工作，以 INTLF 为时钟，在 FSCMGAP<2:0>设置的采样间隔内，如果 EN 保持为 1 则将 IF（时钟故障标志位）置 1，表明检测到时钟故障。


图 5-6 时钟故障检测时序图¹

5.11.2 时钟故障保护处理

检测到时钟故障后，系统时钟将自动切换至内部高频时钟源(使用默认分频)继续工作，并产生时钟故障中断信号，时钟故障中断连接到 NMI 中断，允许在发生时钟故障的时候进行软件抢救。

发生时钟故障之后，只要标志位 CKFIF 为 1，则无法切换系统时钟（包括分频）。如果需要切换时钟，需要先切换系统时钟源（如切换成内部高频时钟），然后清除标志位 CKFIF，系统就会使用切换后的时钟工作。

5.12 外设模块时钟使能位

为了降低功耗，默认外设时钟禁止。在使用外设模块时，需要使能该外设模块时钟控制信号，否则模块不工作。通过 PCLK_CTLx (x=0,1,2,3) 外设时钟控制寄存器控制相应的外设时钟。当外设时钟禁止时，CPU 无法对相应的模块寄存器进行写操作。

5.13 中断

必须满足下面的条件才能产生对应的时钟中断

- 相应的时钟使能：系统时钟/外设高频/外设低频选中对应的时钟源，或 OSC_CTL2 中的时钟软件使能位使能
- 对应的中断标志使能位置 1
- 对应的中断标志位为 1

当时钟使能时，当时钟达到稳定条件时，相应的中断标志位会置 1，而当时钟未使能时则标志位保持为 0。时钟稳定条件如下。

表 5-11 时钟稳定条件

时钟源	稳定条件
内部高频	内部高频使能
内部低频	内部低频使能
外部高频	使能后，固定延时和可编程延时结束

¹ 系统时钟频率实际比采样时钟大很多，图示为方便分析将频率差异减小。

外部低频	使能后，固定延时和可编程延时结束
PLL	使能后或初始化后，PLL 时钟锁定，固定延时和可编程延时结束
LP4M	LP4M 使能

当任意中断标志位置 1，且相应的中断使能位置 1 时，会产生 OSC 中断，将中断标志位 OSCIF 置 1。当 OSCIE 为 1、OSCIF 为 1 且全局中断 AIE 为 1 时，则可以进入 OSC 中断。

6 存储器 (memory)

6.1 概述

芯片为统一线性编址。芯片采用小端存储格式，低字节为最低有效位，高字节为最高有效位。

6.2 存储器空间映射

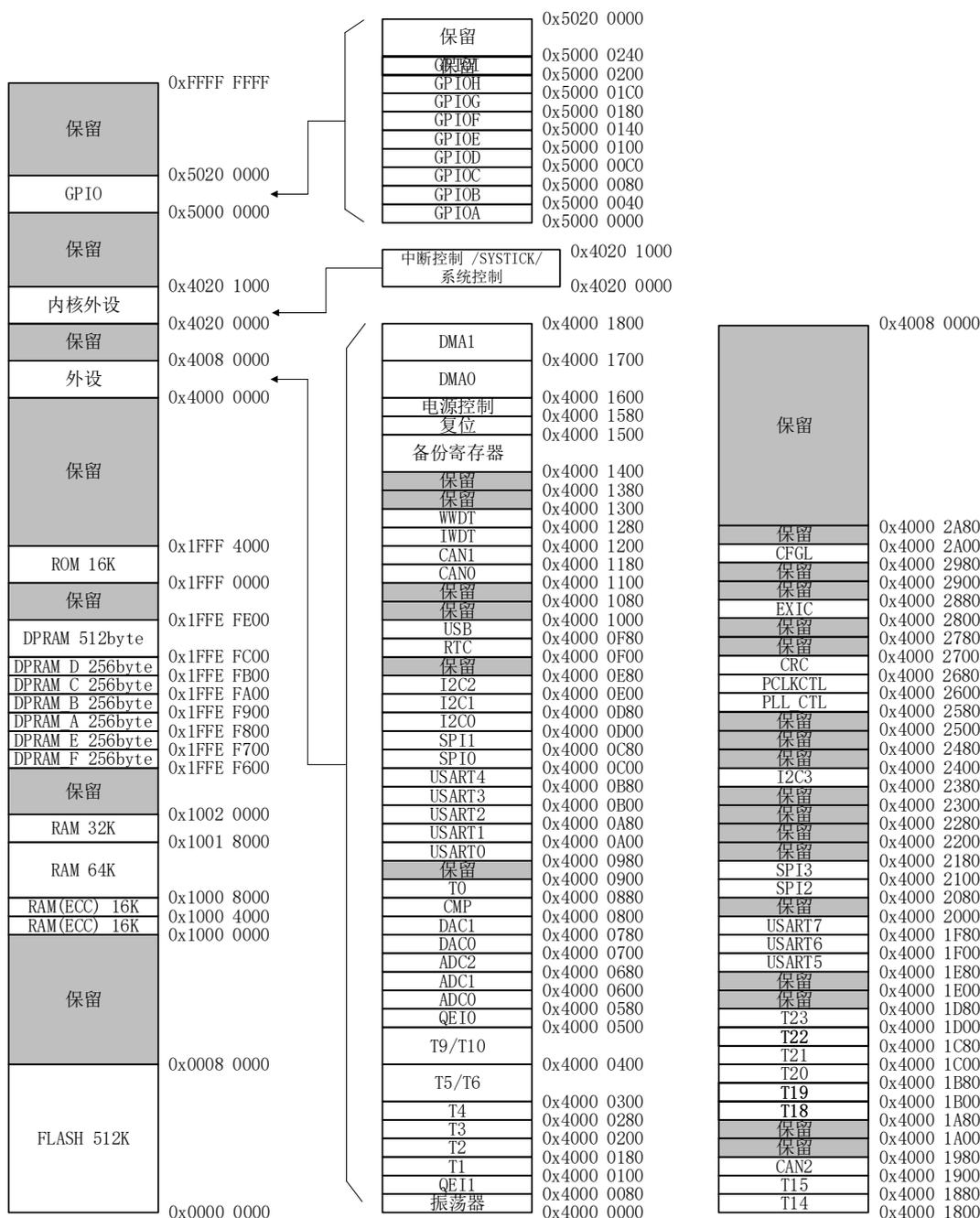


图 6-1 存储空间映射

表 6-1 存储器预留空间分配示意

地址	模块
0x0000 0000 - 0x0007 FFFF	FLASH 空间, 带 ECC 校验
0x1000 0000 - 0x1000 7FFF	单端口 SRAM 空间, 带 ECC 校验
0x1000 8000 - 0x1001 FFFF	单端口 SRAM 空间
0x1FFE F600 - 0x1FFE FDFE	双端口 SRAM 空间, 带 ECC 校验
0x1FFF 0000 - 0x1FFF 3FFF	ROM 空间

0x4000 0000 - 0x4007 FFFF	外设
0x4020 0000 - 0x4020 0FFF	内核外设
0x5000 0000 - 0x501F FFFF	GPIO

6.3 存储器访问说明

芯片拥有 1 个可配置的加密区，为 0x00000000 ~ 0x00003FFF。当使能加密模式时该区域可作为加密区。

当加密区未使能时，LD 指令可以正常访问。当使能加密模式时，在加密区或 ROM 区的 LD 指令可以正常访问加密区内容，而在非加密区和非 ROM 区使用 LD 指令访问加密区的内容时则返回值为 0。加密访问权限可参考表“加密访问权限说明”。

加密区：

- 加密区空间：0x00000000 ~ 0x00003FFF

6.4 ROM 访问说明

ROM 区的程序可以访问所有的存储空间，而非 ROM 区的程序的 LD 操作访问 ROM 区时返回值为 0。

ROM 访问权限可参考表“表 6-2 加密访问权限说明”。

表 6-2 加密访问权限说明

加密模式		ROM	加密区	非加密区
未加密	ROM 区 LD 指令	可访问	可访问	可访问
	FLASH 加密区的 LD 指令	-	可访问	可访问
	非加密区 LD 指令	-	可访问	可访问
加密模式	ROM 区 LD 指令	可访问	可访问	可访问
	加密区 LD 指令	-	可访问	可访问
	非加密区 LD 指令	-	-	可访问

注：-表示访问返回值为 0。

6.5 低功耗 RAM

芯片有 2 块低功耗 RAM，1 块 16K 单端口 SRAM (0x1000 0000-0x1000 3FFF)，和 1 块 256byte 双端口 DPRAM (0x1FFE F800-0x1FFE F8FF)。

在 standby 模式下，可以通过 PM_CTL0 寄存器的 DPRAMSEL 位来配置低功耗的 DPRAM 是否掉电。

在 standby/stop1 模式下，可以通过 PM_CTL0 寄存器的 LPRAMSEL 位来配置低功耗的 SRAM 是否掉电。

当 PM_CTL0 寄存器 DPRAMSEL 和 LPRAMSEL 都为 0 时，执行进入 standby 指令后最终进入 standby1 模式，否则进入 standby0 模式。

7 中断 (interrupt)

7.1 概述

中断特性:

- 支持最多 13 个系统中断，支持最多 64 个外设中断（含软件中断）
- 支持 16 级中断优先级设置
- 支持自动堆栈
- 支持中断入口配置
- 支持嵌套中断
- 支持软件中断

7.2 中断向量表

中断向量表包含堆栈指针的复位值以及所有中断处理程序的入口地址。下表显示了中断在向量表中的顺序。

表 7-1 内核中断向量

向量编号	地址	类型	优先级	中断源描述
0	0x0000 0000	初始 SP	-	初始 SP
1	0x0000 0004	复位	-3	复位
2	0x0000 0008	NMI	-2	不可屏蔽中断，时钟故障检测中断为 NMI 中断
3	0x0000 000C	硬件错误中断	-1	不满足 BREAK 使用条件下使用 BREAK 或在不满足 SVC 使用条件下使用 SVC 指令时产生，可用于系统调试。
4	0x0000 0010	ECC 错误中断	可编程	FLASH/RAM ECC 校验错误
5	0x0000 0014	堆栈错误	可编程	堆栈空间错误
6	0x0000 0018	算术错误	可编程	发生算术错误
7	0x0000 001C	-	可编程	不可使用
8	0x0000 0020	保留	可编程	-
9	0x0000 0024	保留	可编程	-
10	0x0000 0028	保留	可编程	-
11	0x0000 002C	SVC 中断	可编程	超级用户中断，使用 SVC 指令触发，当满足可触发条件时进入该中断，若不满足条件则会进入硬件错误中断。
12	0x0000 0030	保留	可编程	-
13	0x0000 0034	保留	可编程	-
14	0x0000 0038	SOFTSV	可编程	软件中断，可作为上下文切换时的中断
15	0x0000 003C	SYSTICK	可编程	节拍定时器中断

表 7-2 外设中断向量表

向量编号	地址	类型	优先级	中断源描述
16	0x0000 0040	WWDT	可编程	窗口看门狗中断
17	0x0000 0044	EINT16	可编程	连接到 EINT16 的 PVD 中断
18	0x0000 0048	EINT0	可编程	外部事件中断 0
19	0x0000 004C	EINT1	可编程	外部事件中断 1
20	0x0000 0050	EINT2	可编程	外部事件中断 2
21	0x0000 0054	EINT3	可编程	外部事件中断 3
22	0x0000 0058	EINT4	可编程	外部事件中断 4
23	0x0000 005C	EINT<9:5>	可编程	外部事件中断<9:5>
24	0x0000 0060	EINT<15:10>	可编程	外部事件中断<15:10>
25	0x0000 0064	T1	可编程	定时器 1 全局中断
26	0x0000 0068	T3	可编程	定时器 3 全局中断
27	0x0000 006C	T5	可编程	定时器 5 全局中断
28	0x0000 0070	T6	可编程	定时器 6 全局中断
29	0x0000 0074	QEI (T7/T8)	可编程	QEI(T7/T8)全局中断
30	0x0000 0078	CAN2	可编程	CAN2 全局中断
31	0x0000 007C	保留	可编程	保留
32	0x0000 0080	保留	可编程	保留
33	0x0000 0084	T14	可编程	定时器 14 全局中断
34	0x0000 0088	T15	可编程	定时器 15 全局中断
35	0x0000 008C	保留	可编程	保留
36	0x0000 0090	EXIC	可编程	EXIC 全局中断
37	0x0000 0094	ADC0	可编程	ADC0 全局中断
38	0x0000 0098	ADC1	可编程	ADC1 全局中断
39	0x0000 009C	CFGL	可编程	CFGL 全局中断
40	0x0000 00A0	保留	可编程	保留
41	0x0000 00A4	T0	可编程	定时器 0 全局中断
42	0x0000 00A8	DMA0	可编程	DMA0 通道全局中断
43	0x0000 00AC	CMP	可编程	比较器 0/1 全局中断
44	0x0000 00B0	USART0	可编程	USART0 全局中断
45	0x0000 00B4	USART1	可编程	USART1 全局中断
46	0x0000 00B8	SPI0	可编程	SPI0 全局中断
47	0x0000 00BC	SPI1	可编程	SPI1 全局中断
48	0x0000 00C0	DMA1	可编程	DMA1 通道全局中断
49	0x0000 00C4	EINT<19:17>	可编程	EINT17 为 RTC 中断 EINT18 为 TAMPER 的侵入检测中断 EINT19 为 RTC_WKUP 闹钟唤醒中断
50	0x0000 00C8	CAN0	可编程	CAN0 全局中断
51	0x0000 00CC	CAN1	可编程	CAN1 全局中断
52	0x0000 00D0	T9	可编程	定时器 T9 全局中断
53	0x0000 00D4	T10	可编程	定时器 T10 全局中断

向量编号	地址	类型	优先级	中断源描述
54	0x0000 00D8	EINT20	可编程	EINT20 为 AES 中断
55	0x0000 00DC	保留	可编程	保留
56	0x0000 00E0	OSC(CLK)	可编程	振荡器中断
57	0x0000 00E4	I2C0	可编程	I2C0 全局中断
58	0x0000 00E8	I2C1	可编程	I2C1 全局中断
59	0x0000 00EC	I2C2	可编程	I2C2 全局中断
60	0x0000 00F0	USB	可编程	USB 全局中断
61	0x0000 00F4	T2	可编程	T2 全局中断
62	0x0000 00F8	T4	可编程	T4 全局中断
63	0x0000 00FC	保留	可编程	保留
64	0x0000 0100	USART2	可编程	USART2 全局中断
65	0x0000 0104	USART3	可编程	USART3 全局中断
66	0x0000 0108	USART4	可编程	USART4 全局中断
67	0x0000 010C	SPI2	可编程	SPI2 全局中断
68	0x0000 0110	SPI3	可编程	SPI3 全局中断
69	0x0000 0114	ADC2	可编程	ADC2 全局中断
70	0x0000 0118	T18	可编程	T18 全局中断
71	0x0000 011C	T19	可编程	T19 全局中断
72	0x0000 0120	T22/T23	可编程	T22/T23 全局中断
73	0x0000 0124	WKP0/1/2/3/4	可编程	唤醒引脚 0/1/2/3/4 全局中断
74	0x0000 0128	T20	可编程	T20 全局中断
75	0x0000 012C	T21	可编程	T21 全局中断
76	0x0000 0130	I2C3	可编程	I2C3 全局中断
77	0x0000 0134	USART5	可编程	USART5 全局中断
78	0x0000 0138	USART6	可编程	USART6 全局中断
79	0x0000 013C	USART7	可编程	USART7 全局中断

注：中断向量表中保留部分可作为软件中断使用。

7.3 中断相关寄存器

表 7-3 中断相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	INT_CTL0	R/W	中断控制寄存器 0	0x0000 0008
0x004	INT_EIE0	R/W	中断使能寄存器 0 (保留)	0x0000FFFF
0x00C	INT_EIE1	R/W	中断使能寄存器 1	0x0000 0000
0x014	INT_EIE2	R/W	中断使能寄存器 2	0x0000 0000
0x01C	INT{EIF0	R/W	中断标志位寄存器 0	0x0000 0000
0x024	INT{EIF1	R/W	中断标志位寄存器 1	0x0000 0000
0x02C	INT{EIF2	R/W	中断标志位寄存器 2	0x0000 0000
0x034	INT_IP0	R/W	中断优先级控制寄存器 0	0x0000 0000
0x038	INT_IP1	R/W	中断优先级控制寄存器 1	0x0000 0000
0x03C	INT_IP2	R/W	中断优先级控制寄存器 2	0x0000 0000
0x040	INT_IP3	R/W	中断优先级控制寄存器 3	0x0000 0000
0x044	INT_IP4	R/W	中断优先级控制寄存器 4	0x0000 0000
0x048	INT_IP5	R/W	中断优先级控制寄存器 5	0x0000 0000
0x04C	INT_IP6	R/W	中断优先级控制寄存器 6	0x0000 0000
0x050	INT_IP7	R/W	中断优先级控制寄存器 7	0x0000 0000
0x054	INT_IP8	R/W	中断优先级控制寄存器 8	0x0000 0000
0x058	INT_IP9	R/W	中断优先级控制寄存器 9	0x0000 0000
0x05C	INT_IP10	R/W	中断优先级控制寄存器 10	0x0000 0000
0x060	INT_IP11	R/W	中断优先级控制寄存器 11	0x0000 0000
0x064	INT_IP12	R/W	中断优先级控制寄存器 12	0x0000 0000
0x068	INT_IP13	R/W	中断优先级控制寄存器 13	0x0000 0000
0x06C	INT_IP14	R/W	中断优先级控制寄存器 14	0x0000 0000
0x070	INT_IP15	R/W	中断优先级控制寄存器 15	0x0000 0000
0x074	INT_IP16	R/W	中断优先级控制寄存器 16	0x0000 0000
0x078	INT_IP17	R/W	中断优先级控制寄存器 17	0x0000 0000
0x07C	INT_IP18	R/W	中断优先级控制寄存器 18	0x0000 0000
0x080	INT_EINTMASK	R/W	外部中断屏蔽寄存器	0x0000 0000
0x084	INT_EINTRISE	R/W	外部中断上升沿选择寄存器	0x0000 0000
0x088	INT_EINTFALL	R/W	外部中断下降沿选择寄存器	0x0000 0000
0x08C	INT_EINTF	R/W	外部中断标志位寄存器	0x0000 0000
0x094	INT_EINTSS0	R/W	外部中断线中断源选择寄存器 0	0x0000 0000
0x098	INT_EINTSS1	R/W	外部中断线中断源选择寄存器 1	0x0000 0000
0x09C	INT_CTL1	R/W	中断控制寄存器 1	0x0000 0000

基地址: 0x4020 0000

7.3.1 INT_CTL0 中断控制寄存器 0

表 7-4 INT_CTL0 中断控制寄存器 0

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值																																	
R/W		R	R	R	R	R	R	R		R	R	R	R	R	R	R	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R	R		R/W	R/W	R/W
位名		INTACT<6:0>							INTPEND<6:0>						PRIBASE<3:0>						DSALIGN	FAULTMASK	INTPREEMPT	INTPENDING				PRIGROUP<1:0>		AIE			

- INTACT<6:0>**: 记录当前正在处理的中断向量编号（只读）
- INTPEND<6:0>**: 记录当前悬起的优先级最高的中断向量编号（只读）
- PRIBASE<3:0>**: 优先级响应基级设置。当该位不为 0 时，优先级数值大于或等于该位的异常均被忽略，该位为 0 时，无影响（其中低四位保留，只有高四位有效）
- DSALIGN**: 中断自动堆栈双字/字对齐选择
0 = 中断自动堆栈使用双字对齐
1 = 中断自动堆栈使用字对齐
- FAULTMASK**: 硬件错误中断屏蔽位
0 = FAULT 异常未屏蔽
1 = FAULT 异常及优先级可配置中断被屏蔽
- INTPREEMPT**: 为 1 时表示一个悬起的中断将在下一步时进入活动状态（可用于单步执行的调试目的）
- INTPENDING**: 当前有除 NMI 之外的中断挂起标志位
0 = 没有除 NMI 之外的中断挂起
1 = 有除 NMI 之外的中断挂起
- PRIGROUP<1:0>**: 优先级分组, 用于优先级设置中确定抢占优先级和子优先级的控制位数。
00 = 抢占优先级为 3 位, 子优先级为 1 位
01 = 抢占优先级为 2 位, 子优先级为 2 位
10 = 抢占优先级为 1 位, 子优先级为 3 位
11 = 抢占优先级为 0 位, 子优先级为 4 位
- AIE**: 全局可屏蔽中断使能位, 该中断使能控制不包含复位/NMI/硬件错误中断。
0 = 全局可屏蔽中断禁止
1 = 全局可屏蔽中断使能

7.3.2 INT_EIE0 中断使能寄存器 0

表 7-5 INT_EIE0 中断使能寄存器 0¹

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值																																	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W				
位名																	SYSTICKIE	SOFTSVIE	保留	保留	SYCIE	保留	保留	保留			ARTFAULTIE	STACKIE	ECCIE				

¹ 该寄存器 bit15-bit4 位中，未使用的位也可以作为软件中断使用，bit3-bit0 不可使用, bit7 不可使用。

SYSTICKIE: 系统节拍定时器中断使能位, 默认使能。

0 = 系统节拍中断禁止

1 = 系统节拍中断使能

SOFTSVIE: 可挂起超级用户中断使能位, 默认使能。

0 = 可挂起超级用户中断禁止

1 = 可挂起超级用户中断使能

SVCIE: 超级用户中断使能位, 默认使能。

0 = 超级用户中断禁止

1 = 超级用户中断使能

ARIFAULTIE: 算术错误中断 (Arithmetic Fault Interrupt enable) 使能位, 默认使能。

0 = 算术错误中断禁止

1 = 算术错误中断使能

STACKIE: 堆栈错误中断使能位, 默认使能。

0 = 堆栈错误中断禁止

1 = 堆栈错误中断使能

ECCIE: ECC 校验错误中断使能位。

0 = ECC 错误中断禁止

1 = ECC 错误中断使能

7.3.3 INT_EIE1 中断使能寄存器 1

表 7-6 INT_EIE1 中断使能寄存器 1

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	SPI1IE	SPI0IE	USART1IE	USART0IE	CMP1E	DMA0IE	TOIE	保留	CFGLIE	ADCLIE	ADCOIE	EXICIE	保留	T15IE	T14IE	保留	保留	CAN2IE	QE1IE	T6IE	T5IE	T3IE	T1IE	EINT15TO10IE	EINT9TO5IE	EINT4IE	EINT3IE	EINT2IE	EINT1IE	EINT0IE	EINT16IE	WWDTE

SPI1IE: SPI1 中断使能位

0 = 中断禁止

1 = 中断使能

SPI0IE: SPI0 中断使能位

0 = 中断禁止

1 = 中断使能

USART1IE: USART1 中断使能位

0 = 中断禁止

1 = 中断使能

USART0IE: USART0 中断使能位

0 = 中断禁止

1 = 中断使能

CMP1E: CMP0/1/2 中断使能位

0 = 中断禁止

1 = 中断使能

DMA0IE: DMA0 中断使能位

- 0 = 中断禁止
- 1 = 中断使能
- T0IE: T0 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- CFGLIE: CFGL 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- ADC1IE: ADC1 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- ADC0IE: ADC0 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- EXICIE: EXIC 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- T15IE: T15 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- T14IE: T14 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- CAN2IE: CAN2 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- QEIIIE: QEIO/1 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- T6IE: T6 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- T5IE: T5 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- T3IE: T3 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- T1IE: T1 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- EINT15TO10IE: EINT<15:10>中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能

EINT9TO5IE: EINT<9:5>中断使能位

- 0 = 中断禁止
- 1 = 中断使能

EINT4IE: EINT4 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

EINT3IE: EINT3 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

EINT2IE: EINT2 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

EINT1IE: EINT1 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

EINT0IE: EINT0 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

EINT16IE: EINT16 (PVD) 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

WWDTIE: WWDT 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

7.3.4 INT_EIE2 中断使能寄存器 2

表 7-7 INT_EIE2 中断使能寄存器 2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	USART7IE	USART6IE	USART5IE	I2C3IE	T21IE	T20IE	WKPIE	T2T23IE	T19IE	T18IE	ADC2IE	SP13IE	SP12IE	USART4IE	USART3IE	USART2IE	保留	T4IE	T2IE	USBIE	I2C2IE	I2C1IE	I2C0IE	OSCIIE	保留	EINT20IE	T10IE	T9IE	CAN1IE	CAN0IE	EINT19TO17IE	DMA1IE

USART7IE: USART7 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

USART6IE: USART6 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

USART5IE: USART5 中断使能位

- 0 = 中断禁止
- 1 = 中断使能

I2C3IE: I2C3 中断使能位

- 0 = 中断禁止

1 = 中断使能
T21IE: T21 中断使能位
0 = 中断禁止
1 = 中断使能
T20IE: T20 中断使能位
0 = 中断禁止
1 = 中断使能
WKPIE: WKP0/1/2/3/4 中断使能位
0 = 中断禁止
1 = 中断使能
T22T23IE: T22/T23 中断使能位
0 = 中断禁止
1 = 中断使能
T19IE: T19 中断使能位
0 = 中断禁止
1 = 中断使能
T18IE: T18 中断使能位
0 = 中断禁止
1 = 中断使能
ADC2IE: ADC2 中断使能位
0 = 中断禁止
1 = 中断使能
SPI3IE: SPI3 中断使能位
0 = 中断禁止
1 = 中断使能
SPI2IE: SPI2 中断使能位
0 = 中断禁止
1 = 中断使能
USART4IE: USART4 中断使能位
0 = 中断禁止
1 = 中断使能
USART3IE: USART3 中断使能位
0 = 中断禁止
1 = 中断使能
USART2IE: USART2 中断使能位
0 = 中断禁止
1 = 中断使能
T4IE: T4 中断使能位
0 = 中断禁止
1 = 中断使能
T2IE: T2 中断使能位
0 = 中断禁止
1 = 中断使能
USBIE: USB 中断使能位

- 0 = 中断禁止
- 1 = 中断使能
- I2C2IE: I2C2 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- I2C1IE: I2C1 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- I2C0IE: I2C0 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- OSCIE: OSC(CLK)中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- EINT20IE: EINT20 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- T10IE: T10 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- T9IE: T9 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- CAN1IE: CAN1 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- CAN0IE: CAN0 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- EINT19TO17IE: EINT<19:17>中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能
- DMA1IE: DMA1 中断使能位
 - 0 = 中断禁止
 - 1 = 中断使能

7.3.5 INT_EIF0 中断标志位寄存器 0

 表 7-8 INT_EIF0 中断标志位寄存器 0¹

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W		
位名																	SYSTICKIF	SOFTSVIF	保留	保留	SVCIF	保留	保留	保留		ARIFALTIF	STACKIF	ECCIF	HARDAULTIF	NMIIF		

SYSTICKIF: SYSTICK 中断标志位。当系统节拍定时器产生中断或软件置位时该位置 1。

0 = 无中断挂起

1 = 中断挂起

SOFTSVIF: SOFTSV 中断标志位。该位为软件置位。

0 = 无中断挂起

1 = 中断挂起

SVCIF: SVC 中断标志位。当使用 SVC 指令并满足中断条件时该位置 1。如使用 SVC 且不满足 SVC 挂起条件时该位为 0。

0 = 无中断挂起

1 = 中断挂起

ARIFALTIF: ARIFALT 中断标志位。执行除法运算时发生被 0 时该位置 1。

0 = 无中断挂起

1 = 中断挂起

STACKIF: STACK 中断标志位。当使用 PUSH/POP 指令堆栈内容超出 SYS_RAMSPA 寄存器规定的范围时产生堆栈错误中断，该位置 1。

0 = 无中断挂起

1 = 中断挂起

ECCIF: ECC 错误中断标志位，当 FLASH/RAM 出现 ECC 校验错误时该位置 1。

0 = 无中断挂起

1 = 中断挂起

HARDAULTIF: HARDAULT 中断标志位。不满足 BREAK 使用条件下使用 BREAK 或在 不满足 SVC 使用条件下使用 SVC 指令时该位置 1。

0 = 无中断挂起

1 = 中断挂起

NMIIF: NMI 中断标志位。当产生时钟故障检测中断时该位置 1。

0 = 无中断挂起

1 = 中断挂起

¹ 该寄存器 bit15-bit2 位中，保留位也可以作为软件中断使用，bit0/bit1/bit7 不可使用。该寄存器相应的位在相应的中断函数开始运行前会自动清除。

7.3.6 INT_EIF1 中断标志位寄存器 1

 表 7-9 INT_EIF1 中断标志位寄存器¹

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	SPI1IF	SPI0IF	USART1IF	USART0IF	CMPIF	DMA0IF	T0IF	保留	CFGLIF	ADC1IF	ADC0IF	EXICIF	保留	T15IF	T14IF	保留	保留	CAN2IF	QE1IF	T6IF	T5IF	T3IF	T1IF	EINT15T010IF	EINT9T05IF	EINT4IF	EINT3IF	EINT2IF	EINT1IF	EINT0IF	EINT16IF	WWD1IF	

SPI1IF: SPI1 中断标志位

0 = 无中断挂起

1 = 中断挂起

SPI0IF: SPI0 中断标志位

0 = 无中断挂起

1 = 中断挂起

USART1IF: USART1 中断标志位

0 = 无中断挂起

1 = 中断挂起

USART0IF: USART0 中断标志位

0 = 无中断挂起

1 = 中断挂起

CMPIF: CMP 中断标志位

0 = 无中断挂起

1 = 中断挂起

DMA0IF: DMA0 中断标志位

0 = 无中断挂起

1 = 中断挂起

T0IF: T0 中断标志位

0 = 无中断挂起

1 = 中断挂起

CFGLIF: CFGL 中断标志位

0 = 无中断挂起

1 = 中断挂起

ADC1IF: ADC1 中断标志位

0 = 无中断挂起

1 = 中断挂起

ADC0IF: ADC0 中断标志位

0 = 无中断挂起

1 = 中断挂起

EXICIF: EXIC 中断标志位

0 = 无中断挂起

1 = 中断挂起

¹ 该寄存器相应的位在相应的中断开始响应时会自动清除。

- T15IF: T15 中断标志位
0 = 无中断挂起
1 = 中断挂起
- T14IF: T14 中断标志位
0 = 无中断挂起
1 = 中断挂起
- CAN2IF: CAN2 中断标志位
0 = 无中断挂起
1 = 中断挂起
- QE1IF: QE1(T7/T8)中断标志位
0 = 无中断挂起
1 = 中断挂起
- T6IF: T6 中断标志位
0 = 无中断挂起
1 = 中断挂起
- T5IF: T5 中断标志位
0 = 无中断挂起
1 = 中断挂起
- T3IF: T3 中断标志位
0 = 无中断挂起
1 = 中断挂起
- T1IF: T1 中断标志位
0 = 无中断挂起
1 = 中断挂起
- EINT15TO10IF: EINT<15:10>中断标志位
0 = 无中断挂起
1 = 中断挂起
- EINT9TO5IF: EINT<9:5>中断标志位
0 = 无中断挂起
1 = 中断挂起
- EINT4IF: EINT4 中断标志位
0 = 无中断挂起
1 = 中断挂起
- EINT3IF: EINT3 中断标志位
0 = 无中断挂起
1 = 中断挂起
- EINT2IF: EINT2 中断标志位
0 = 无中断挂起
1 = 中断挂起
- EINT1IF: EINT1 中断标志位
0 = 无中断挂起
1 = 中断挂起
- EINT0IF: EINT0 中断标志位
0 = 无中断挂起

1 = 中断挂起

EINT16IF: EINT16(PVD)中断标志位

0 = 无中断挂起

1 = 中断挂起

WWDTIF: WWDT 中断标志位

0 = 无中断挂起

1 = 中断挂起

7.3.7 INT_EIF2 中断标志位寄存器 2

表 7-10 INT_EIF2 中断标志位寄存器 2

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	USART7IF	USART6IF	USART5IF	I2C3IF	T21IF	T20IF	WKPIF	T22T23IF	T19IF	T18IF	ADC2IF	SPI3IF	SPI2IF	USART4IF	USART3IF	USART2IF	保留	T4IF	T2IF	保留	I2C2IF	I2C1IF	I2C0IF	OSCIF	保留	EINT20IF	T10IF	T9IF	CAN1IF	CAN0IF	EINT19T017IF	DMA1IF	

USART7IF: USART7 中断标志位

0 = 无中断挂起

1 = 中断挂起

USART6IF: USART6 中断标志位

0 = 无中断挂起

1 = 中断挂起

USART5IF: USART5 中断标志位

0 = 无中断挂起

1 = 中断挂起

I2C3IF: I2C3 中断标志位

0 = 无中断挂起

1 = 中断挂起

T21IF: T21 中断标志位

0 = 无中断挂起

1 = 中断挂起

T20IF: T20 中断标志位

0 = 无中断挂起

1 = 中断挂起

WKPIF: WKP0/1/2/3/4 中断标志位

0 = 无中断挂起

1 = 中断挂起

T22T23IF: T22/T23 中断标志位

0 = 无中断挂起

1 = 中断挂起

T19IF: T19 中断标志位

0 = 无中断挂起

1 = 中断挂起

T18IF: T18 中断标志位

0 = 无中断挂起
1 = 中断挂起

ADC2IF: ADC2 中断标志位
0 = 无中断挂起
1 = 中断挂起

SPI3IF: SPI3 中断标志位
0 = 无中断挂起
1 = 中断挂起

SPI2IF: SPI2 中断标志位
0 = 无中断挂起
1 = 中断挂起

USART4IF: USART4 中断标志位
0 = 无中断挂起
1 = 中断挂起

USART3IF: USART3 中断标志位
0 = 无中断挂起
1 = 中断挂起

USART2IF: USART2 中断标志位
0 = 无中断挂起
1 = 中断挂起

T4IF: T4 中断标志位
0 = 无中断挂起
1 = 中断挂起

T2IF: T2 中断标志位
0 = 无中断挂起
1 = 中断挂起

I2C2IF: I2C2 中断标志位
0 = 无中断挂起
1 = 中断挂起

I2C1IF: I2C1 中断标志位
0 = 无中断挂起
1 = 中断挂起

I2C0IF: I2C0 中断标志位
0 = 无中断挂起
1 = 中断挂起

OSCIF: OSC(CLK)中断标志位
0 = 无中断挂起
1 = 中断挂起

EINT31TO22IF: EINT<31:22>中断标志位
0 = 无中断挂起
1 = 中断挂起

EINT20IF: EINT20 中断标志位
0 = 无中断挂起
1 = 中断挂起

T10IF: T10 中断标志位

- 0 = 无中断挂起
- 1 = 中断挂起

T9IF: T9 中断标志位

- 0 = 无中断挂起
- 1 = 中断挂起

CAN1IF: CAN1 中断标志位

- 0 = 无中断挂起
- 1 = 中断挂起

CAN0IF: CAN0 中断标志位

- 0 = 无中断挂起
- 1 = 中断挂起

EINT19TO17IF: EINT<19:17>中断标志位

- 0 = 无中断挂起
- 1 = 中断挂起

DMA1IF: DMA1 中断标志位

- 0 = 无中断挂起
- 1 = 中断挂起

7.3.8 INT_IPx 中断优先级寄存器 (x=0~18)

下列的中断优先级寄存器中, PRIx<7:0>的低 4 位值无效。

表 7-11 INT_IP0 中断优先级寄存器 0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI7<7:0>							PRI6<7:0>							PRI5<7:0>							PRI4<7:0>											

表 7-12 INT_IP1 中断优先级寄存器 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI11<7:0>							PRI10<7:0>							PRI9<7:0>							PRI8<7:0>											

表 7-13 INT_IP2 中断优先级寄存器 2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI15<7:0>							PRI14<7:0>							PRI13<7:0>							PRI12<7:0>											

表 7-14 INT_IP3 中断优先级寄存器 3

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI19<7:0>							PRI18<7:0>							PRI17<7:0>							PRI16<7:0>											

表 7-15 INT_IP4 中断优先级寄存器 4

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PRI23<7:0>								PRI22<7:0>								PRI21<7:0>								PRI20<7:0>							

表 7-16 INT_IP5 中断优先级寄存器 5

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PRI27<7:0>								PRI26<7:0>								PRI25<7:0>								PRI24<7:0>							

表 7-17 INT_IP6 中断优先级寄存器 6

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PRI31<7:0>								PRI30<7:0>								PRI29<7:0>								PRI28<7:0>							

表 7-18 INT_IP7 中断优先级寄存器 7

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PRI35<7:0>								PRI34<7:0>								PRI33<7:0>								PRI32<7:0>							

表 7-19 INT_IP8 中断优先级寄存器 8

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PRI39<7:0>								PRI38<7:0>								PRI37<7:0>								PRI36<7:0>							

表 7-20 INT_IP9 中断优先级寄存器 9

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PRI43<7:0>								PRI42<7:0>								PRI41<7:0>								PRI40<7:0>							

表 7-21 INT_IP10 中断优先级寄存器 10

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PRI47<7:0>								PRI46<7:0>								PRI45<7:0>								PRI44<7:0>							

表 7-22 INT_IP11 中断优先级寄存器 11

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PRI51<7:0>								PRI50<7:0>								PRI49<7:0>								PRI48<7:0>							

表 7-23 INT_IP12 中断优先级寄存器 12

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI55<7:0>								PRI54<7:0>								PRI53<7:0>								PRI52<7:0>							

表 7-24 INT_IP13 中断优先级寄存器 13

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI59<7:0>								PRI58<7:0>								PRI57<7:0>								PRI56<7:0>							

表 7-25 INT_IP14 中断优先级寄存器 14

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI63<7:0>								PRI62<7:0>								PRI61<7:0>								PRI60<7:0>							

表 7-26 INT_IP15 中断优先级寄存器 15

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI67<7:0>								PRI66<7:0>								PRI65<7:0>								PRI64<7:0>							

表 7-27 INT_IP16 中断优先级寄存器 16

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI71<7:0>								PRI70<7:0>								PRI69<7:0>								PRI68<7:0>							

表 7-28 INT_IP17 中断优先级寄存器 17

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI75<7:0>								PRI74<7:0>								PRI73<7:0>								PRI72<7:0>							

表 7-29 INT_IP18 中断优先级寄存器 18

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W				
位名	PRI79<7:0>								PRI78<7:0>								PRI77<7:0>								PRI76<7:0>							

PRIx<7:4>: 向量编号为 x 的中断优先级设置。中断有效级低四位无效, 优先级为 16 级。

- 0000 = 中断优先级为 0
- 0001 = 中断优先级为 1
- 0010 = 中断优先级为 2
- 0011 = 中断优先级为 3
- 0100 = 中断优先级为 4
- 0101 = 中断优先级为 5
- 0110 = 中断优先级为 6
- 0111 = 中断优先级为 7

1000 = 中断优先级为 8
1001 = 中断优先级为 9
1010 = 中断优先级为 10
1011 = 中断优先级为 11
1100 = 中断优先级为 12
1101 = 中断优先级为 13
1110 = 中断优先级为 14
1111 = 中断优先级为 15

PRI4<7:4>: 保留, 可作为软件中断设置, 设置同上。。
PRI5<7:4>: STACK 中断优先级控制位, 设置同上。
PRI6<7:4>: ARIFAUULT 中断优先级控制位, 设置同上。
PRI7<7:4>: 保留, 可作为软件中断设置, 设置同上。
PRI8<7:4>: 保留, 可作为软件中断设置, 设置同上。
PRI9<7:4>: 保留, 可作为软件中断设置, 设置同上。
PRI10<7:4>: 保留, 可作为软件中断设置, 设置同上。
PRI11<7:4>: SVC 中断优先级控制位, 设置同上。
PRI12<7:4>: 保留, 可作为软件中断设置, 设置同上。
PRI13<7:4>: 保留, 可作为软件中断设置, 设置同上。
PRI14<7:4>: SOFTSV 中断优先级控制位, 设置同上。
PRI15<7:4>: SYSTICK 中断优先级控制位, 设置同上。
PRI16<7:4>: 16 号中断优先级控制位, 设置同上。
PRI17<7:4>: 17 号中断优先级控制位, 设置同上。
PRI18<7:4>: 18 号中断优先级控制位, 设置同上。
PRI19<7:4>: 19 号中断优先级控制位, 设置同上。
PRI20<7:4>: 20 号中断优先级控制位, 设置同上。
PRI21<7:4>: 21 号中断优先级控制位, 设置同上。
PRI22<7:4>: 22 号中断优先级控制位, 设置同上。
PRI23<7:4>: 23 号中断优先级控制位, 设置同上。
PRI24<7:4>: 24 号中断优先级控制位, 设置同上。
PRI25<7:4>: 25 号中断优先级控制位, 设置同上。
PRI26<7:4>: 26 号中断优先级控制位, 设置同上。
PRI27<7:4>: 27 号中断优先级控制位, 设置同上。
PRI28<7:4>: 28 号中断优先级控制位, 设置同上。
PRI29<7:4>: 29 号中断优先级控制位, 设置同上。
PRI30<7:4>: 30 号中断优先级控制位, 设置同上。
PRI31<7:4>: 31 号中断优先级控制位, 设置同上。
PRI32<7:4>: 32 号中断优先级控制位, 设置同上。
PRI33<7:4>: 33 号中断优先级控制位, 设置同上。
PRI34<7:4>: 34 号中断优先级控制位, 设置同上。
PRI35<7:4>: 35 号中断优先级控制位, 设置同上。
PRI36<7:4>: 36 号中断优先级控制位, 设置同上。
PRI37<7:4>: 37 号中断优先级控制位, 设置同上。
PRI38<7:4>: 38 号中断优先级控制位, 设置同上。
PRI39<7:4>: 39 号中断优先级控制位, 设置同上。

PRI40<7:4>: 40 号中断优先级控制位, 设置同上。
PRI41<7:4>: 41 号中断优先级控制位, 设置同上。
PRI42<7:4>: 42 号中断优先级控制位, 设置同上。
PRI43<7:4>: 43 号中断优先级控制位, 设置同上。
PRI44<7:4>: 44 号中断优先级控制位, 设置同上。
PRI45<7:4>: 45 号中断优先级控制位, 设置同上。
PRI46<7:4>: 46 号中断优先级控制位, 设置同上。
PRI47<7:4>: 47 号中断优先级控制位, 设置同上。
PRI48<7:4>: 48 号中断优先级控制位, 设置同上。
PRI49<7:4>: 49 号中断优先级控制位, 设置同上。
PRI50<7:4>: 50 号中断优先级控制位, 设置同上。
PRI51<7:4>: 51 号中断优先级控制位, 设置同上。
PRI52<7:4>: 52 号中断优先级控制位, 设置同上。
PRI53<7:4>: 53 号中断优先级控制位, 设置同上。
PRI54<7:4>: 54 号中断优先级控制位, 设置同上。
PRI55<7:4>: 55 号中断优先级控制位, 设置同上。
PRI56<7:4>: 56 号中断优先级控制位, 设置同上。
PRI57<7:4>: 57 号中断优先级控制位, 设置同上。
PRI58<7:4>: 58 号中断优先级控制位, 设置同上。
PRI59<7:4>: 59 号中断优先级控制位, 设置同上。
PRI60<7:4>: 60 号中断优先级控制位, 设置同上。
PRI61<7:4>: 61 号中断优先级控制位, 设置同上。
PRI62<7:4>: 62 号中断优先级控制位, 设置同上。
PRI63<7:4>: 63 号中断优先级控制位, 设置同上。
PRI64<7:4>: 64 号中断优先级控制位, 设置同上。
PRI65<7:4>: 65 号中断优先级控制位, 设置同上。
PRI66<7:4>: 66 号中断优先级控制位, 设置同上。
PRI67<7:4>: 67 号中断优先级控制位, 设置同上。
PRI68<7:4>: 68 号中断优先级控制位, 设置同上。
PRI69<7:4>: 69 号中断优先级控制位, 设置同上。
PRI70<7:4>: 70 号中断优先级控制位, 设置同上。
PRI71<7:4>: 71 号中断优先级控制位, 设置同上。
PRI72<7:4>: 72 号中断优先级控制位, 设置同上。
PRI73<7:4>: 73 号中断优先级控制位, 设置同上。
PRI74<7:4>: 74 号中断优先级控制位, 设置同上。
PRI75<7:4>: 75 号中断优先级控制位, 设置同上。
PRI76<7:4>: 76 号中断优先级控制位, 设置同上。
PRI77<7:4>: 77 号中断优先级控制位, 设置同上。
PRI78<7:4>: 78 号中断优先级控制位, 设置同上。
PRI79<7:4>: 79 号中断优先级控制位, 设置同上。

7.3.9 INT_EINTMASK 外部中断屏蔽寄存器

表 7-30 INT_EINTMASK 外部中断屏蔽寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W												R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
位名												EINTM<20:0>																				

EINTM_x: 外部中断线 x 中断屏蔽位(x=0~20)
 0 = 禁止对应的外部中断线上的中断请求
 1 = 使能对应外部中断线上的中断请求

7.3.10 INT_EINTRISE 外部中断上升沿选择寄存器

表 7-31 INT_EINTRISE 外部中断上升沿选择寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W												R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名												EINTRI<20:0>																				

EINTRI_x: 外部中断线 x 上升沿选择位 (x=0~20)
 0 = 禁止外部中断线上的上升沿中断
 1 = 使能外部中断线上的上升沿中断

7.3.11 INT_EINTFALL 外部中断下降沿选择寄存器

表 7-32 INT_EINTFALL 外部中断下降沿选择寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W												R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名												EINTFA<20:0>																				

EINTFA_x: 外部中断线 x 下降沿选择位 (x=0~20)
 0 = 禁止外部中断线上的下降沿中断
 1 = 使能外部中断线上的下降沿中断

7.3.12 INT_EINTF 外部中断中断标志位寄存器

表 7-33 INT_EINTF 外部中断标志位寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W												R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名												EINTIF<20:0>																				

EINTIF_x: 外部中断线 x 中断标志位 (x=0~20)
 0 = 没有发生外部中断
 1 = 发生了外部中断

7.3.13 INT_EINTSS0 外部中断源选择寄存器 0

 表 7-34 INT_EINTSS0 外部中断源选择寄存器 0¹

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W																		
位名	EINTSOU7 <3:0>			EINTSOU6 <3:0>			EINTSOU5 <3:0>			EINTSOU4 <3:0>			EINTSOU3 <3:0>			EINTSOU2 <3:0>			EINTSOU1 <3:0>			EINTSOU0 <3:0>										

EINTSOU0<3:0>: 外部中断线 0 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA0 作为 EINT0 输入
- 0001 = 选择 PB0 作为 EINT0 输入
- 0010 = 选择 PC0 作为 EINT0 输入
- 0011 = 选择 PD0 作为 EINT0 输入
- 0100 = 选择 PE0 作为 EINT0 输入
- 0101 = 选择 PF0 作为 EINT0 输入
- 0110 = 选择 PG0 作为 EINT0 输入
- 0111 = 选择 PH0 作为 EINT0 输入
- 其他= 保留

EINTSOU1<3:0>: 外部中断线 1 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA1 作为 EINT1 输入
- 0001 = 选择 PB1 作为 EINT1 输入
- 0010 = 选择 PC1 作为 EINT1 输入
- 0011 = 选择 PD1 作为 EINT1 输入
- 0100 = 选择 PE1 作为 EINT1 输入
- 0101 = 选择 PF1 作为 EINT1 输入
- 0110 = 选择 PG1 作为 EINT1 输入
- 0111 = 选择 PH1 作为 EINT1 输入
- 其他 = 保留

EINTSOU2<3:0>: 外部中断线 2 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA2 作为 EINT2 输入
- 0001 = 选择 PB2 作为 EINT2 输入
- 0010 = 选择 PC2 作为 EINT2 输入
- 0011 = 选择 PD2 作为 EINT2 输入
- 0100 = 选择 PE2 作为 EINT2 输入
- 0101 = 选择 PF2 作为 EINT2 输入
- 0110 = 选择 PG2 作为 EINT2 输入
- 0111 = 选择 PH2 作为 EINT2 输入
- 其他 = 保留

EINTSOU3<3:0>: 外部中断线 3 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA3 作为 EINT3 输入
- 0001 = 选择 PB3 作为 EINT3 输入

¹ 每种封装能够使用的引脚数量有所差别, 实际以对应封装支持的引脚为准。

0010 = 选择 PC3 作为 EINT3 输入
0011 = 选择 PD3 作为 EINT3 输入
0100 = 选择 PE3 作为 EINT3 输入
0101 = 选择 PF3 作为 EINT3 输入
0110 = 选择 PG3 作为 EINT3 输入
0111 = 选择 PH3 作为 EINT3 输入
其他 = 保留

EINTSOU4<3:0>: 外部中断线 4 中断源选择位, 选择外设中断线的中断输入源。

0000 = 选择 PA4 作为 EINT4 输入
0001 = 选择 PB4 作为 EINT4 输入
0010 = 选择 PC4 作为 EINT4 输入
0011 = 选择 PD4 作为 EINT4 输入
0100 = 选择 PE4 作为 EINT4 输入
0101 = 选择 PF4 作为 EINT4 输入
0110 = 选择 PG4 作为 EINT4 输入
0111 = 选择 PH4 作为 EINT4 输入
其他 = 保留

EINTSOU5<3:0>: 外部中断线 5 中断源选择位, 选择外设中断线的中断输入源。

0000 = 选择 PA5 作为 EINT5 输入
0001 = 选择 PB5 作为 EINT5 输入
0010 = 选择 PC5 作为 EINT5 输入
0011 = 选择 PD5 作为 EINT5 输入
0100 = 选择 PE5 作为 EINT5 输入
0101 = 选择 PF5 作为 EINT5 输入
0110 = 选择 PG5 作为 EINT5 输入
0111 = 选择 PH5 作为 EINT5 输入
其他 = 保留

EINTSOU6<3:0>: 外部中断线 6 中断源选择位, 选择外设中断线的中断输入源。

0000 = 选择 PA6 作为 EINT6 输入
0001 = 选择 PB6 作为 EINT6 输入
0010 = 选择 PC6 作为 EINT6 输入
0011 = 选择 PD6 作为 EINT6 输入
0100 = 选择 PE6 作为 EINT6 输入
0101 = 选择 PF6 作为 EINT6 输入
0110 = 选择 PG6 作为 EINT6 输入
0111 = 选择 PH6 作为 EINT6 输入
其他 = 保留

EINTSOU7<3:0>: 外部中断线 7 中断源选择位, 选择外设中断线的中断输入源。

0000 = 选择 PA7 作为 EINT7 输入
0001 = 选择 PB7 作为 EINT7 输入
0010 = 选择 PC7 作为 EINT7 输入
0011 = 选择 PD7 作为 EINT7 输入
0100 = 选择 PE7 作为 EINT7 输入
0101 = 选择 PF7 作为 EINT7 输入

- 0110 = 选择 PG7 作为 EINT7 输入
- 0111 = 选择 PH7 作为 EINT7 输入
- 其他 = 保留

7.3.14 INT_EINTSS1 外部中断源选择寄存器 1

表 7-35 INT_EINTSS1 外部中断源选择寄存器 1¹

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W															
位名	EINTSOU15 <3:0>			EINTSOU14 <3:0>			EINTSOU13 <3:0>			EINTSOU12 <3:0>			EINTSOU11 <3:0>			EINTSOU10 <3:0>			EINTSOU9 <3:0>			EINTSOU8 <3:0>										

EINTSOU8<3:0>: 外部中断线 8 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA8 作为 EINT8 输入
- 0001 = 选择 PB8 作为 EINT8 输入
- 0010 = 选择 PC8 作为 EINT8 输入
- 0011 = 选择 PD8 作为 EINT8 输入
- 0100 = 选择 PE8 作为 EINT8 输入
- 0101 = 选择 PF8 作为 EINT8 输入
- 0110 = 选择 PG8 作为 EINT8 输入
- 0111 = 选择 PH8 作为 EINT8 输入
- 其他 = 保留

EINTSOU9<3:0>: 外部中断线 9 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA9 作为 EINT9 输入
- 0001 = 选择 PB9 作为 EINT9 输入
- 0010 = 选择 PC9 作为 EINT9 输入
- 0011 = 选择 PD9 作为 EINT9 输入
- 0100 = 选择 PE9 作为 EINT9 输入
- 0101 = 选择 PF9 作为 EINT9 输入
- 0110 = 选择 PG9 作为 EINT9 输入
- 0111 = 选择 PH9 作为 EINT9 输入
- 其他 = 保留

EINTSOU10<3:0>: 外部中断线 10 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA10 作为 EINT10 输入
- 0001 = 选择 PB10 作为 EINT10 输入
- 0010 = 选择 PC10 作为 EINT10 输入
- 0011 = 选择 PD10 作为 EINT10 输入
- 0100 = 选择 PE10 作为 EINT10 输入
- 0101 = 选择 PF10 作为 EINT10 输入
- 0110 = 选择 PG10 作为 EINT10 输入
- 0111 = 选择 PH10 作为 EINT10 输入
- 其他 = 保留

¹ 每种封装能够使用的引脚数量有所差别, 实际以对应封装支持的引脚为准。

EINTSOU11<3:0>: 外部中断线 11 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA11 作为 EINT11 输入
- 0001 = 选择 PB11 作为 EINT11 输入
- 0010 = 选择 PC11 作为 EINT11 输入
- 0011 = 选择 PD11 作为 EINT11 输入
- 0100 = 选择 PE11 作为 EINT11 输入
- 0101 = 选择 PF11 作为 EINT11 输入
- 0110 = 选择 PG11 作为 EINT11 输入
- 0111 = 选择 PH11 作为 EINT11 输入
- 其他 = 保留

EINTSOU12<3:0>: 外部中断线 12 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA12 作为 EINT12 输入
- 0001 = 选择 PB12 作为 EINT12 输入
- 0010 = 选择 PC12 作为 EINT12 输入
- 0011 = 选择 PD12 作为 EINT12 输入
- 0100 = 选择 PE12 作为 EINT12 输入
- 0101 = 选择 PF12 作为 EINT12 输入
- 0110 = 选择 PG12 作为 EINT12 输入
- 0111 = 选择 PH12 作为 EINT12 输入
- 其他 = 保留

EINTSOU13<3:0>: 外部中断线 13 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA13 作为 EINT13 输入
- 0001 = 选择 PB13 作为 EINT13 输入
- 0010 = 选择 PC13 作为 EINT13 输入
- 0011 = 选择 PD13 作为 EINT13 输入
- 0100 = 选择 PE13 作为 EINT13 输入
- 0101 = 选择 PF13 作为 EINT13 输入
- 0110 = 选择 PG13 作为 EINT13 输入
- 0111 = 选择 PH13 作为 EINT13 输入
- 其他 = 保留

EINTSOU14<3:0>: 外部中断线 14 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA14 作为 EINT14 输入
- 0001 = 选择 PB14 作为 EINT14 输入
- 0010 = 选择 PC14 作为 EINT14 输入
- 0011 = 选择 PD14 作为 EINT14 输入
- 0100 = 选择 PE14 作为 EINT14 输入
- 0101 = 选择 PF14 作为 EINT14 输入
- 0110 = 选择 PG14 作为 EINT14 输入
- 0111 = 选择 PH14 作为 EINT14 输入
- 其他 = 保留

EINTSOU15<3:0>: 外部中断线 15 中断源选择位, 选择外设中断线的中断输入源。

- 0000 = 选择 PA15 作为 EINT15 输入
- 0001 = 选择 PB15 作为 EINT15 输入
- 0010 = 选择 PC15 作为 EINT15 输入

- 0011 = 选择 PD15 作为 EINT15 输入
- 0100 = 选择 PE15 作为 EINT15 输入
- 0101 = 选择 PF15 作为 EINT15 输入
- 0110 = 选择 PG15 作为 EINT15 输入
- 0111 = 选择 PH15 作为 EINT15 输入
- 其他 = 保留

7.3.15 INT_CTL1 中断控制寄存器 1

表 7-36 INT_CTL1 中断控制寄存器 1

	031	030	029	028	027	026	025	024	023	022	021	020	019	018	017	016	015	014	013	012	011	010	09	08	07	06	05	04	03	02	01	00																					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
R/W																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																								INTDELY<7:0>																													

INTDELY<7:0>: 中断延时控制。可设置堆栈过程延时时间。在检测到有效挂起中断时，在进栈过程中需要等待这个延迟周期完成才会完成进栈保存过程并开始查找中断入口进入中断服务程序。默认为延迟 16 个时钟周期。（注意：这个延时不等于中断完成整个进栈过程需要的时间，整个进栈延时参考中断延迟小节）

- 00000000 = 0 个延时时间周期（默认）
- 00000001 = 1 个延时时间周期
- 00000010 = 2 个延时时间周期
- 00000011 = 3 个延时时间周期
-
- 00010000 = 16 个延时时间周期
-
- 11111111 = 255 个延时时间周期

7.4 中断使用

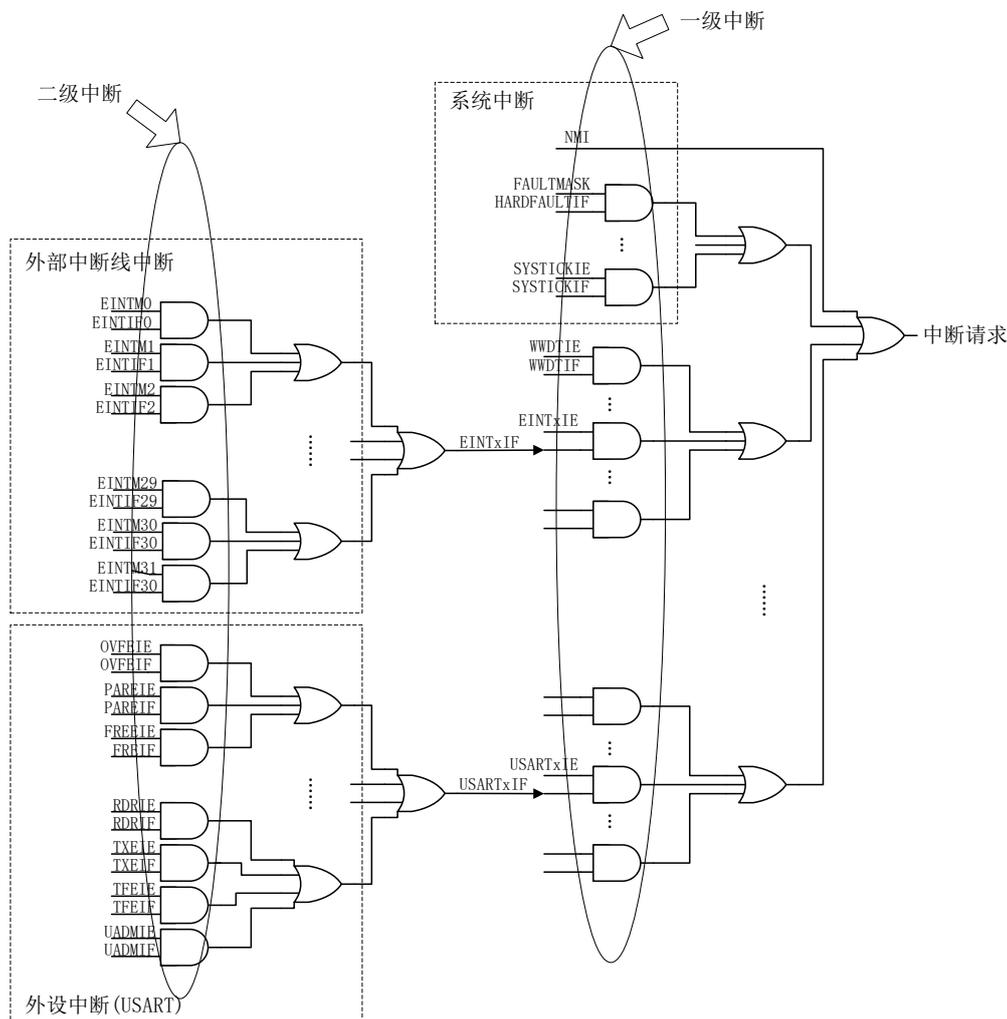


图 7-1 中断结构框图

如上所示为中断的结构框图。中断分成两级中断控制，一级中断确定系统是否响应，二级中断确定是否发出中断请求。其中，一级中断使能为 INT_EIE0/1/2 寄存器，一级中断的标志寄存器为 INT_EIE0/1/2 寄存器，二级中断使能为各外设对应的中断寄存器。

当外设需要使用中断系统时，需要将相应的外设中断使能位使能，即需要将二级中断使能位使能，这样在满足中断条件时候，就能够发出中断请求，将一级中断中断的中断标志位置 1，请求系统响应其中断。当一级中断相应的中断使能也使能时，系统就能够响应相应的中断。

外部中断线的使用和外设类似，其中 INT_EINTMASK 为二级中断使能寄存器，INT_EINTF 为二级中断标志位寄存器。

当中断响应后，会自动将一级中断的中断标志位清零，而二级中断的标志位需要在中断处理函数中软件清零。如果二级中断的标志位没有清零，会出现持续进入该中断的现象。

所有中断都能作为软件中断使用。作为软件中断使用时，只需要将对应的一级中断使能位使能，并软件将对应的标志位置 1 即可。

7.5 内核中断类型

表 7-37 内核中断类型描述

内核中断类型	中断描述
复位	指向复位时的程序入口地址。复位优先级最高，且不可屏蔽。
NMI	不可屏蔽中断，由时钟故障检测中断产生。该中断为除复位外优先级最高的中断。
硬件错误	<p>硬件错误中断，为除了复位和 NMI 外优先级最高的中断。</p> <p>不满足 BREAK 使用条件下使用 BREAK 或在满足 SVC 使用条件下使用 SVC 指令时产生硬件错误中断。具体如下：</p> <p>(1) 当 debug 未使能时使用 BREAK 指令</p> <p>(2) 当前抢占优先级高于或等于 SVC 中断时使用 SVC 指令，或 SVC 中断被禁止时使用 SVC 指令</p> <p>注：在处理硬件错误中断进程中或在硬件错误中断被屏蔽时，执行 BREAK 和 SVC 指令无响应，效果等同于空指令。</p>
ECC 校验错误	当 CPU 读 FLASH 或带 ECC 的 RAM/DPRAM 时，若发生 ECC 校验错误，则产生 ECC 校验错误标志。
堆栈错误	堆栈内容超出 SYS_RAMSPA 寄存器规定的范围时产生堆栈错误。在执行 PUSH/POP 指令时或进入、退出中断使用堆栈指针时检测是否超界。
算术错误	执行除法运算时发生被 0 除则产生算术错误中断。
超级用户模式	<p>调用超级用户指令 (SVC) 时产生，满足进入中断条件时则进入该中断，否则将进入硬件错误中断。SVC 中断需要同时满足下面情况：</p> <p>(1) SVC 中断使能</p> <p>(2) 当前抢占优先级低于 SVC 中断优先级</p> <p>注：</p> <p>当前优先级指当前系统正在响应的中断的优先级，不同于当前挂起等待响应的中断优先级。</p> <p>当前等待中的优先级大于 SVC 优先级，会出现优先响应优先级更高的等待中的中断，这种情况视同抢占，不会引起硬件错误中断。</p> <p>在不可屏蔽中断进程中、或在处理硬件错误中断进程中、或在硬件错误中断被屏蔽时，执行 SVC 指令无响应，效果等同于空指令。</p>
SOFTSV	软件中断。
SYSTICK	节拍定时器中断，节拍定时器产生。

注：SYS_RAMSPA 寄存器说明和 RAM 空间指示说明详见“系统控制 (system control)”章节。

7.6 中断优先级

复位、NMI、硬件错误优先级固定，且比其他中断优先级要高，其中复位优先级最高用 -3 表示，NMI 中断优先级为 -2，硬件错误中断为 -1，除此之外所有的中断优先级都是可配置的。中断优先级的表示方法中，数值越小优先级越高。

单片机共有 16 级用户可配置中断优先级，通过 INT_IPx (x=0~18) 寄存器可以设置相

应的中断优先级，其中优先级 0 为可配置优先级中最高优先级，优先级 15 为最低优先级。当单片机发生不止一个中断事件时，CPU 会最先响应优先级最高的中断。当单片机新产生的中断较当前正在处理的芯片的优先级高且满足抢占要求时，高优先级中断会马上被单片机响应，从而实现抢占功能。

每一个中断的优先级可以分成抢占优先级位和子优先级位，其中抢占优先级为高位，子优先级为低位。实现抢占需要通过 INTCTL0 寄存器中的 PRIGROUP<1:0>位设置抢占优先级和子优先级的优先级位数。抢占优先级即可发生抢占事件的优先级，中断只有拥有比当前响应的中断更高的抢占优先级时才能发生抢占事件，小于或等于都不能发生抢占事件。子优先级不可发生抢占事件，但在相同的抢占优先条件下，单片机会先响应子优先级高的中断事件。抢占优先级和子优先级都一致时，中断最先响应中断向量编号小的中断。

设置抢占优先级和子优先级时，子优先级设计最少为 1 位，最多可设置为 4 位；而抢占优先级最少为 0 位，最多为 3 位。

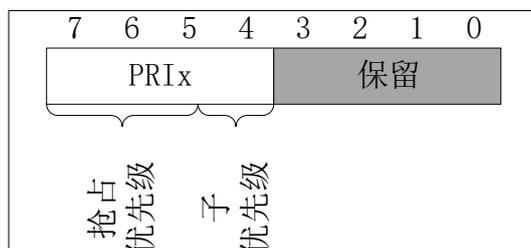


图 7-2 PRIGROUP<1:0>为 00 时

通过 PRIBASE<3:0>可以设置优先级响应的基数，也就是说优先级数值大于或等于该位的异常均被忽略。通过该位可以选择响应的优先级级别，方便对不同优先级实现统一管理。当 PRIBASE<3:0>=0x0000 时，该控制为补齐作用。

7.7 中断行为介绍

中断模块可用于响应多种中断事件。当一个中断事件发生时，不论该中断是否使能，该中断的中断标志位 EIFx (x=0~2) 都会置 1。只有当相应的中断使能位 EIEx (x=0~2) 置 1，且总使能位 (AIE) 置 1 时，该中断才会得到 CPU 的响应。

注意，AIE 为全局优先级可编程中断的总使能，不包含复位、NMI、硬件错误中断的使能。复位和 NMI 不可屏蔽，硬件错误中断可以通过 FAULTMASK 硬件错误中断屏蔽位来控制。当 FAULTMASK 置 1 后，同时会将所有优先级可配置的中断屏蔽掉。

当中断标志位置 1，而中断使能未开启时，保持该悬起状态直到软件清零或复位。若中断使能打开且 CPU 没有正在处理的中断，该中断能够立即得到 CPU 的响应。此时该中断相应的中断向量编号会记录在 INT_CTL0 寄存器中的 INTACT<6:0>位，而 INT_CTL0 寄存器中 INTPEND<6:0>位则记录了当前中断标志位置 1 的所有中断中优先级最高的中断向量编号。

通过 INT_IPx (x=0~18) 可以设置中断的优先级，通过 INT_CTL0 寄存器中的 PRIGROUP<1:0>位设置抢占优先级和子优先级的优先级位数。抢占优先级和子优先级一旦设定，所有的中断优先级设定 INT_IPx (x=0~18) 都使用该设定。当 CPU 正在处理一个低抢占优先级的中断，而一个满足高抢占优先级的中断产生时，CPU 会马上停止当前低优先级中断，堆栈保存相应的信息，开始执行高抢占优先级中断。当抢占优先级一样时，CPU 不会发生抢占现象，会先将当前中断处理完后，接着继续处理当前有效中断中优先级最高的中断。抢占优先级一样时，由子优先级决定优先级的高低。抢占优先级和子优先级都一致时，中断最先响应中断向量编号小的中断。

当响应中断时，CPU 会自动执行堆栈过程。自动进栈的过程如下：依次保存 xPSR（映射寄存器）、PC（返回地址）、LR（R13）、R0~R4（通用寄存器）。中断返回时出栈过程如下：依照后入先出的关系恢复进栈的内容，同时硬件清除中断标志位。硬件堆栈保证了中断发生嵌套时中断处理的准确性。

7.8 中断的进入处理

当有一个优先级足够高的中断挂起，并满足下面的任何一个条件，就进入中断处理：

- 芯片处于线程模式
- 新的中断的优先级高于正在处理的中断，且满足抢占条件

当处理中断时，芯片会将信息压入到当前的堆栈中，8 个数据字的组成的结构为栈帧。栈帧包含以下信息：

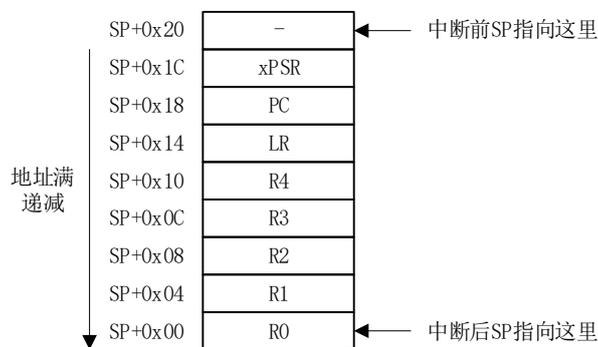


图 7-3 栈帧内容

其中 xPSR 为映射寄存器，该寄存器不可直接读写，内容如下所示。

表 7-38 xPSR 寄存器（映射寄存器）

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
位名	N	Z	C	V																		STKALIGN	INTACT<6:0>									

N：负值标志

- 0 = 非负值
- 1 = 负值

Z：零值标志

- 0 = 非零值
- 1 = 零值

C：进位或借位标志

- 0 = 加法无进位或减法有借位
- 1 = 加法有进位或减法无借位

V：溢出标志

- 0 = 无溢出
- 1 = 溢出

STKALIGN：用于堆栈 8 字节对齐。中断进入时，芯片使用入栈的 SYS_PSW 的 bit<9>指示栈对齐。中断返回时，芯片使用这个位来恢复正确的栈对齐。

0 = 堆栈空间已经是 8 字节对齐

1 = 堆栈空间系统处理成 8 字节对齐

INTACT<6:0>: 记录当前正在处理的中断向量编号

入栈后, 堆栈指针指向栈帧的最低地址单元。栈帧按照双字或字对齐。栈帧包含返回地址 PC。返回地址 PC 为被中止的程序指令的地址。这个值在中断返回时返还给 PC, 使被中断的程序恢复执行。

堆栈过程完成, 芯片会执行一次向量提取, 从向量表中相应的位置读出中断处理的起始地址。在入栈完成后, 芯片开始执行中断处理程序。同时, 芯片向 LR 写入一个 INTRETURN 值。这个值指示了栈帧对应哪个堆栈指针以及中断出现前芯片处于什么工作模式。

当芯片进入中断处理后, 在中断程序执行前, 芯片会自动清除 EIFx (x=0~2) 中断的挂起状态。如果此时中断源尚未消失, 则该中断状态会重新变成挂起状态, 如果进栈操作没有完成前接收到多次相同的中断信号, 则该中断不会重复触发。INTACT<6:0>显示了当前正在处理的中断向量编号, INTPEND<6:0>显示了当前悬起的优先级最高的中断向量编号。当中断开始响应时, INTPEND<6:0>显示的中断可以被处理, 在进入中断程序前 INTPEND<6:0>会更新当前优先级最高的中断向量编号, 而 INTACT<6:0>会更新成当前正在处理的中断向量编号。

注: 中断进栈是会自动进栈保存 R0~R4, LR, PC, xPSR, 当中断程序中使用到其他的寄存器时, 需要额外的手动保存, 防止中断进程破坏原始值。

7.9 中断的返回

当芯片处于处理模式并且执行下面其中一条指令将 INTRETURN 加载到 PC 时, 执行中断返回操作:

- JMP 指令, 例: JMP LR
- POP 指令, 将 LR 值加载到 PC 中, 例: POP {LR}

在中断进栈时, 芯片会将一个 INTRETURN 值保存到 LR 寄存器中。在中断返回时需要识别这个值来判断是一个正常的分支跳转, 还是中断返回操作。INTRETURN<31: 4>全为 1, INTRETURN<3:0>则指出了堆栈返回的模式和堆栈指针的使用情况。如下表所示。

表 7-39 中断返回 INTRETURN 行为

INTRETURN 值	返回行为
0xFFFFFFFF1	返回到处理模式, 中断返回获得 MSP 状态, 返回后使用 MSP
0xFFFFFFFF5	返回到处理模式, 中断返回获得 PSP 状态, 返回后使用 PSP
0xFFFFFFFF9	返回到线程模式, 中断返回获得 MSP 状态, 返回后使用 MSP
0xFFFFFFF9D	返回到线程模式, 中断返回获得 PSP 状态, 返回后使用 PSP
其他	保留

INTRETURN<3>: 记录进入本次中断前服务例程所处状态。

0 = 进入中断前为处理模式

1 = 进入中断前为线程模式

INTRETURN<2>: 记录进入本次中断前服务例程所使用的堆栈指针。

0 = 进入中断前使用 MSP 作为堆栈指针

1 = 进入中断前使用 PSP 作为堆栈指针

注意, 使用 POP 指令实现中断返回时, 确保之前已经正确使用了 PUSH 指令, 且 PUSH

的内容需要包含 LR 的值。

在特殊应用情况下，可通过修改 INTRETURN 的值来达到特殊的返回目的。通过修改 INTRETURN<3>的值可以决定返回处理模式或是线程模式。通过修改 INTRETURN<2>的值可以确定返回后使用 MSP 或时 PSP 作为堆栈指针，当在修改前请确认堆栈指针是否可用。

7.10 中断堆栈地址对齐方式

在自动堆栈过程中，通过 INT_CTL0 中断控制寄存器 0 的 DSALIGN 中断自动堆栈双字/字对齐选择位可以选择堆栈 xSP 地址的对齐方式。xSP 有两种对齐方式，当 DSALIGN 为 0 时（默认为 0），xSP 为双字对齐；当 DSALIGN 为 1 时，xSP 为字对齐。详见“中断堆栈地址对齐方式”。

7.11 中断向量表重映射

SYS_VECTOFF 中断向量表重映射控制寄存器可以设置中断向量表的起始地址，通过该寄存器的值可以实现中断向量表重映射功能。详见“中断向量表重映射”。

7.12 中断延迟

中断延迟是中断能否完成实时处理的重要指标，KF32 系列提供了低延时的中断堆栈过程。完成的堆栈延时包括 3 个部分延时：当前指令完成延时，自动堆栈保护现场延时，获取中断入口并跳转到中断服务程序延时。

- 当前指令完成延时

当中断挂起并且相应的中断使能位使能时，中断将会进入中断处理程序。为保证 CPU 指令运行的完整性，在中断进入中断处理程序之前，CPU 会先完成已经进入译码/执行阶段的指令，并将尚未进入译码/执行阶段的指令舍弃，然后进入到中断处理（注：16 位的指令在这个阶段会全部舍弃掉）。中断退出后重新取指执行进入中断前舍弃的指令。由于从中断标志位置位到开始中断处理需要两个周期，故该延时最大为指令实现周期减 2（即若指令为 2 周期以内实现就不会额外增加延时）。

注意：

- (1) 在执行除法指令时，由于除法运算可能最长需要 12 个周期，为了保证中断的及时响应，在除法执行期间如果检测到中断会打断除法的继续进行，优先响应中断，并在中断退出后重新进行除法运算。
- (2) 在执行 PUSH/POP 指令时，由于指令的运行周期可能较长，为了保证中断的及时响应，在除法 PUSH/POP 期间如果检测到中断会打断该指令继续进行，并将堆栈指针还原，已经出栈或入栈的操作不做处理，优先响应中断，并在中断退出后重新执行 PUSH/POP 指令。
- (3) 在 DMA 占用 RAM 总线时，会暂停 DMA 的操作，优先响应中断响应操作。
- (4) 如果中断太过频繁，以至于相邻中断的时间小于 DIV/PUSH/POP 等指令周期，则可能会出现指令一直被打断，无法执行的情况。

- 自动堆栈保护现场延时

中断处理时会自动进栈保护中断现场，自动进栈过程需要 9 个内核周期。自动进栈保护的同时，内核会启动一个可配置的延时，延时结束后才会获取中断入口。检测到有效中断标志位后下一个周期可配置延时初始化并开始递减。自动进栈延时与可配置延时不冲突，最大延时取两者中延时最大者。注意，只有在可配置延时结束时才能获得中断入口地址。通过 INT_CTL1 中断控制寄存器 1 的 INTDELY<7:0>可以控制进栈过程中自动保存过程的延时时间，默认为延时 0 个时钟周期。如果可配置的延时小于等于 8 时，该过程需要 8 个内核时钟周期，而默认情况下可配置延时为 0 个周期，故默认情况下该过程需要 8 个时钟周期。

- 获取中断入口并跳转到中断服务程序延时

当自动堆栈保存结束并且延时结束时，可以获取中断入口地址。并根据中断入口地址跳转到相应的中断服务程序中，该过程需要至少 5 个内核时钟周期。由于跳转需要重新取指，实际延时时间可能还需要将 FLASH 读取时间加上（这个时间可通过 FLASH_CFG 寄存器的 TCFG<3:0>选择。）

故默认情况下，进栈延时最小为：2+8+5=15 个内核时钟。其中，2 为中断出现后需要经过一个周期后才会进入中断处理过程；8 为堆栈时间，5 为获取中断入口地址并从中断入口获取中断服务程序的时间。

在实际使用过程中，由于获取中断向量表入口地址、获取新的中断处理程序指令等都需要等待 FLASH 的读取，故实际延时会比上述的过程长。

7.13 外部中断控制

外部中断拥有 20 个中断边沿检测器组成，每个输入线都可以独立配置触发事件（可以选择指定的引脚或指定的触发源的上升沿触发或者下降沿触发或双边沿触发），每个外部中断线都可以独立使能和屏蔽。

注意：外部中断标志位寄存器 INT_EINTF 的标志位需要手动清除。

外部中断控制特性：

- 每个中断都有独立的触发和屏蔽
- 每个中断线都有专用的状态位
- 可检测上升沿和下降沿
- 支持最多 20 个中断事件请求
- 支持软件中断

通用 IO 端口以下图的方式连接到 16 个外部中断线上。通过 INT_EINTSS0 外部中断源选择寄存器 0 和 INT_EINTSS1 外部中断源选择寄存器 1 可以使能相应的 IO 口作为外部中断线的输入。

注：每种封装能够使用的引脚数量有所差别，实际以对应封装支持的引脚为准。

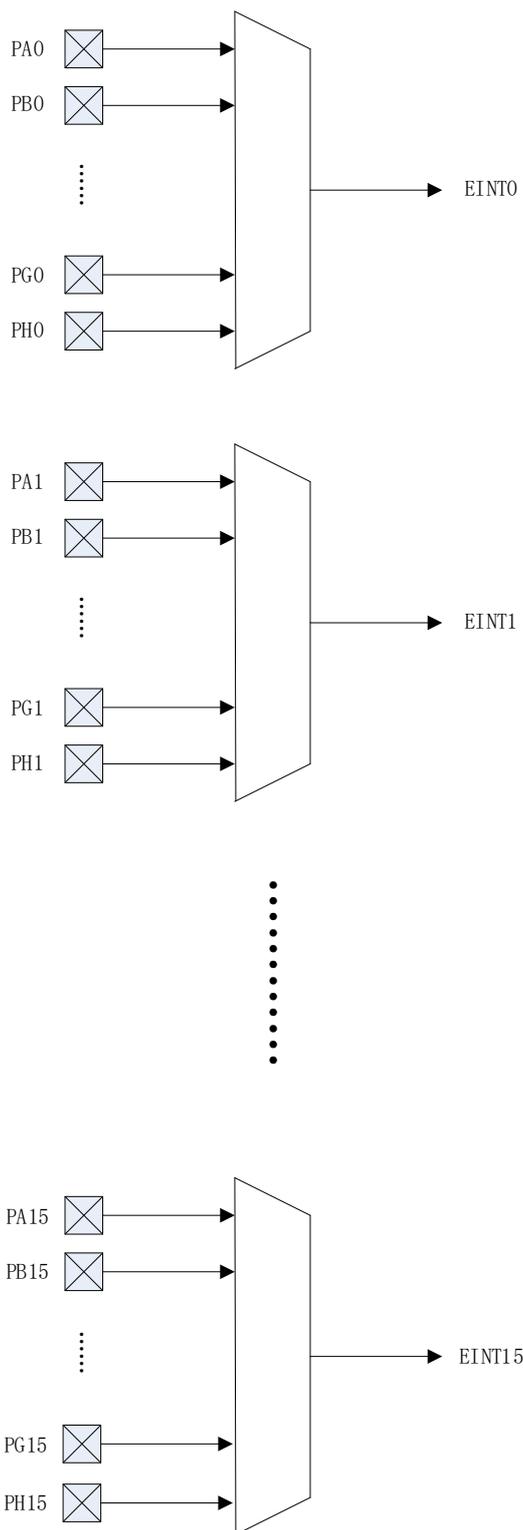


图 7-4 引脚外部中断线映射图

此外，另外的外部中断线连接如下：

- EINT16 连接到 PVD 输出
- EINT17 连接到 RTC 中断（含闹铃唤醒中断）
- EINT18 连接到 TAMPER 侵入检测（任意侵入检测标志位都影响 EINT18）

- EINT19 连接到 RTC 闹钟唤醒事件
- EINT20 连接到 AES 中断

外部中断先的使用:

- (1) 通过 INT_EIE0 和 INT_EIE1 寄存器相应的位使能相应的外设中断线, 其中, EINT20 使用 EINT20IE, EINT19~EINT17 使用 EINT19TO17IE, EINT16 使用 EINT16IE, EINT15~EINT10 使用 EINT15TO10IE, EINT9~EINT5 使用 EINT9TO5IE, EINT4 使用 EINT4IE, EINT3 使用 EINT3IE, EINT2 使用 EINT2IE, EINT1 使用 EINT1IE, EINT0 使用 EINT0IE;
- (2) 通过 INT_EINTMASK 对应的位可以使能外部对应编号的外部中断线;
- (3) 通过 INT_EINTRISE 和 INT_EINTFALL 可以设置外部中断线信号上升沿或下降沿触发中断;
- (4) 通过 EINTSS0/1 可以设置 EINT15~EINT0 选择哪个引脚作为外部中断线的源, EINT20~16 为内部信号源或作为软件中断使用;
- (5) 当检测到外部中断信号时, 会将 INT_EINTF 对应的位置 1, 若 INT_EINTMASK 使能时则将 INT_EIE0 和 INT_EIE1 对应位置 1;
- (6) INT_EINTF 位需要软件清零。

7.14 中断唤醒

CPU 在休眠的时候可以通过中断唤醒。通过 SYS_MCTL 休眠控制寄存器可以控制休眠和唤醒的状态。详见“休眠与休眠唤醒”。

7.15 中断堆栈指针使用

芯片提供两个 SP 指针, 一个为 MSP, 即主堆栈指针; 一个是 PSP, 即进程堆栈指针。系统默认使用 MSP。当芯片初始化时, 系统自动将中断向量表中的初始 SP 指针赋给 MSP, 而 PSP 则处于未初始化状态, 在使用 PSP 之前需要用户手动初始化 PSP 的值。

同一时刻内只能选择一个堆栈指针, 通过 SYS_MCTL 系统模式控制寄存器中的 SPSEL 位可以选择当前使用的指针类型。当 SPSEL 为 0 时选择 MSP 为当前堆栈指针, 这也是默认的堆栈指针; 当 SPSEL 为 1 时选择 PSP 为当前堆栈指针。

设置 SPSEL 之后, 在当前进程中 (没有发生嵌套中断或中断返回时) 可以使用 SPSEL 指定的堆栈指针工作。当发生嵌套中断后, 在嵌套中断进程中默认使用 MSP 指针, 可再次切换成 PSP 指针; 当发生中断返回时, 中断返回使用的指针为 INTRETURN 的值指定的指针, 通过修改 INTRETURN 的值可以修改返回后使用的堆栈指针, 但要注意修改后的指针的数据是否正确。

中断的进入和返回都会更新 SPSEL 位。中断返回时会将会根据 INTRETURN 的值将 SPSEL 的值还原。

7.16 多堆栈使用

通过设置多堆栈, 配合系统节拍定时器可实现多任务处理模式。

将系统节拍定时器设置成需要的时间长度, 当进入系统节拍定时器中断处理时, 可在中

断内执行堆栈切换，以实现多任务切换。多任务切换时注意，如果直接在系统节拍中断处理中切换时，如果系统节拍定时器抢占了更低优先级的中断，则有可能导致被抢占的中断处理延迟，这是不希望看到的。所以系统节拍定时器中断处理时，可以使用一个低优先级软件中断（该软件中断可使用 SOFTSV，或者可以使用未被定义中断），在所有中断都处理完成的最后才做任务切换。

7.17 芯片自动初始化

在芯片自动初始化过程中，芯片会从中断向量表中获取中断向量编号 0 的 MSP 初始化指针值，并赋给 MSP，然后会获取中断向量编号为 1 的复位入口地址赋给 PC。

在芯片上电初始化芯片完成前，如果出现中断向量编号 2~15 的系统中断挂起，则在初始化完成后会将中断全局可屏蔽中断使能位（AIE）使能位置 1，并在进入用户程序后响应最高优先级的中断；如果初始化过程中没有出现中断向量编号 2~15 的系统中断挂起，则在初始化后将中断全局可屏蔽中断使能位（AIE）使能位清零。

7.18 超级用户模式进入

芯片提供了超级用户的功能选择。可使用 SVC 指令或者 SOFTSV 软件中断进入超级用户模式。在中断处理过程中，系统始终为超级用户模式。

SVC 指令为调用超级用户指令，使用该指令时 CPU 需要立即响应 SVC 指令，并进入中断处理进程。在使用 SVC 指令时，若当前中断优先级比 SVC 中断的优先级高或相等时，CPU 无法立即响应 SVC 处理，此时会引起硬件错误中断，并将硬件错误中断标志位置 1，而不会将 SVC 中断标志位置 1。在使用时需要注意 SVC 中断的优先级，同时不可在 SVC 中断处理中嵌套使用 SVC 指令，否则会引起硬件错误中断。若在使用 SVC 指令时，未使能全局中断使能 AIE，也会引起硬件错误中断并进入硬件错误中断处理进程。

当系统处于硬件错误中断或是不可屏蔽中断时，执行 SVC 指令则会被忽略。

SOFTSV 则是可挂起的超级用户调用。使用 SVC 指令时需要 CPU 立即响应超级用户调用，此时会抢占其他低优先级中断进行，这可能是用户不期望的，而使用可挂起的超级用户调用软件中断（或使用其他未定义的中断）则可以解决这个问题。当需要等待当前中断进程处理完成后再执行超级用户调用时，可以使用软件中断 SOFTSV，并将其中断优先级设置为最低，这样就可以在当前所有中断处理完成后再进行超级用户调用。

超级用户权限请参考“超级用户模式使用”。

7.19 中断使用步骤

中断使用最小操作如下：

- (1) 设置中断程序入口，将中断程序入口地址写到中断向量表的指定位置；
- (2) 通过 INT_IPx 设置相应中断的优先级；
- (3) 通过将 EIEx 响应位置 1 使能需要的中断；
- (4) 通过 INT_CTL0 的 AIE 置 1 使能总中断；
- (5) 中断响应自动保存 R0~R4, LR, PC, 其它使用到的通用寄存器需要软件保存；
- (6) EIFx 中断标志位在响应后自动清零，但中断源（如各外设模块内）的中断标志位需要（具体外设标志位的清零方式请参考各模块描述）执行软件清零操作；
- (7) 若使用了除 R0~R4, LR, PC 以外的通用寄存器，需要在中断程序前入栈保存，并

在中断程序结束前出栈恢复：

- (8) 中断结束需要执行中断退出操作（如 JMP LR，详见中断的返回章节）；

8 系统控制 (system control)

8.1 概述

通过 SYS_PSW 程序状态字寄存器可以使用运算结果标志位 N/V/C/Z, 通过 SYS_MCTL 的 SPSEL 可以选择 MSP 或是 PSP 作为堆栈指针, 通过 STKALIGN 可以获得当前堆栈的对齐情况。

通过 SYS_ARCTL 应用和复位控制寄存器可以实现软件产生系统复位。通过 CDMAR 可以清零内核 DMA 状态机的状态, 通过 CDMAAM 可以选择 DMA 与 CPU 读写冲突仲裁模式, 通过 CDMADC 可以选择 CPU 和 DMA 对 BKP 域读写长周期模式控制, 通过 CRWDC 可以选择内核读写外设长周期模式控制。在对 SYS_ARCTL 寄存器进行写操作时, 需要将 0x05FA 写入 SYS_ARCTL 寄存器的 ARKYE<31:16>中, 否则写操作被忽略。

通过 SYS_MCTL 系统模式控制寄存器可以实现系统休眠的控制。通过设置 SLPONMOD 唤醒模式选择位可以选择任意挂起的中断唤醒 CPU, 也可以选择只有当挂起的中断优先级比当前优先级高时才能唤醒 CPU。通过 SLEEPONEXIT 可以实现当中断退出后自动进入休眠 (即使没有使用休眠指令)。

通过 SYS_VECTOFF 中断向量表重映射控制寄存器可以设置中断向量表的首地址, 以此实现中断向量表的重映射。

通过 SYS_RAMSPA 可以设置当前芯片的 RAM 空间的大小。该寄存器在芯片初始化时配置, 用于堆栈溢出中断的判断。

8.2 系统控制相关寄存器

表 8-1 系统控制相关的寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x0B0	SYS_PSW	R/W	程序状态字寄存器	0x0000 0000
0x0B4	SYS_MCTL	R/W	系统模式控制寄存器	0x0000 8200
0x0B8	SYS_ARCTL	R/W	应用复位控制	0xFA05 0010
0x0BC	SYS_VECTOFF	R/W	中断向量表重映射控制寄存器	0x0000 0000
0x0C4	SYS_RAMSPA	R/W	RAM 空间指示寄存器	0x13FF FFFC
0x0C8	SYS_MEMCTL	R/W	程序空间控制寄存器	0x0000 0000

基地址：0x4020 0000

8.2.1 SYS_PSW 程序状态字寄存器

 表 8-2 SYS_PSW 程序状态字寄存器¹

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W																												
位名	N	Z	C	V																												

N: 负值标志

0 = 非负值

1 = 负值

Z: 零值标志

0 = 非零值

1 = 零值

C: 进位或借位标志

0 = 加法无进位或减法有借位

1 = 加法有进位或减法无借位

V: 溢出标志

0 = 无溢出

1 = 溢出

8.2.2 SYS_MCTL 系统模式控制寄存器

表 8-3 SYS_MCTL 系统模式控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																R/W	R/W						R					R/W			R/W	R/W
位名																SPSEL	SUSTA						STACKALIGN				SLPONMOD			SLEEPDEEP	SLEEPONEXIT	

¹ 直接修改该寄存器时，为了确保该寄存器状态能被后续条件跳转指令识别，需要在修改后增加一条 NOP 指令。

SPSEL: 当前有效堆栈指针。在中断处理时默认使用 MSP。注意，系统初始化时会将中断向量的初始 SP 地址赋给 MSP，PSP 是未经过初始化的，用户在使用 PSP 时需要手动将 PSP 初始化。

0 = MSP 是当前的堆栈指针

1 = PSP 是当前的堆栈指针

SUSTA: 超级用户控制位

0 = 用户模式

1 = 超级用户模式

STACKALIGN: 该位不可写，该位的值在每次中断入栈时会自动更新。用于中断自动堆栈 8 字节对齐。中断进入时，内核更新的 SYS_MCTL 的 bit<9>来指示栈对齐情况。中断返回时，芯片使用这个位来恢复正确的栈对齐。

0 = 堆栈空间已经是 8 字节对齐或双字对齐未使能

1 = 堆栈空间由系统自动处理成 8 字节对齐

SLPONMOD: 唤醒模式选择

0 = 只有当挂起的中断优先级比当前优先级高时才能唤醒 CPU

1 = 任意挂起的中断都能唤醒 CPU

SLEEPDEEP: 深度休眠模式控制

0 = 禁止深度休眠模式

1 = 使能深度休眠模式

SLEEPONEXIT: 中断处理返回进入休眠模式

0 = 无影响

1 = 中断返回时进入休眠模式

8.2.3 SYS_ARCTL 应用和复位控制寄存器

表 8-4 SYS_ARCTL 应用和复位控制寄存器

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W								R/W	R/W	R/W	R/W	R/W		W	W							
位名	ARKEY<31:16>																												CDEBUGS	CDMADC	CRWDC	CDMAAM	CDMAR		SYSTEMRST	VECTRST		

ARKEY<31:16>: 应用和复位控制寄存器写操作 key。对该寄存器执行写操作时需要将 0x05FA 写入 ARKEY 中，否则写操作被忽略。读出时该值为 0xFA05。

CDMADC: CPU 和 DMA 读写 BKP 域长周期模式控制

0 = CPU 和 DMA 对 BKP 域读/写分别为 4 个周期

1 = CPU 和 DMA 对 BKP 域读/写分别为 1 个周期

CRWDC: CORE 读写外设长周期模式控制

0 = CORE 对外设读/写为 2 周期，对 RAM 读/写为 1 周期

1 = CORE 对所有区域读/写分别为 1 个周期

CDMAAM: 内核 DMA 与 CPU 读写冲突仲裁模式选择

0 = 高速模式，DMA 与 CPU 同时读写则判断为冲突

1 = 高效模式，DMA 与 CPU 读写占用相同总线时判断为冲突

CDMAR: 内核 DMA 控制状态清零控制

0 = 清零内核 DMA 控制状态

1 = 无作用

SYSTEMRST: 系统复位请求, 该位读数为 0

0 = 无影响

1 = 请求一个系统复位

8.2.4 SYS_VECTOFF 中断向量表重映射控制寄存器

表 8-5 SYS_VECTOFF 中断向量表重映射控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	TBLOFF<31:0>																															

TBLOFF<31:0>: 中断向量表重映射偏移地址。该寄存器定义了中断向量表的起始地址。

注意, TBLOFF<31:0>低两位默认为 0 且不可修改。

8.2.5 SYS_RAMSPA RAM 空间指示寄存器

表 8-6 SYS_RAMSPA RAM 空间指示寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
R/W																																
位名	RAMSPA<25:0>																															

RAMSPA<25:0>: RAM 空间结束地址指示, 用于指示 RAM 空间区大小, 可用于堆栈错误检测, 当堆栈地址超出这个范围时则会将堆栈错误中断标志位置 1, 该寄存器在程序初始化期间根据芯片型号初始化。最低两位固定为 0。

8.2.6 SYS_MEMCTL 程序空间控制寄存器

表 8-7 SYS_MEMCTL 程序空间控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W																																
位名	MEMM<1:0>																															

MEMM<1:0>: FLASH 开始 512 字节空间映射控制, 该寄存器在经过 ROM 启动会会被修改成 0x0000 0002。

00 = 为 ROM 的映射

01 = 为 RAM 的映射

10 = 为 FLASH 的映射

11 = 停止 CPU 运行

8.3 中断堆栈地址对齐

中断堆栈有两种堆栈对齐方式，通过 INT_CTL0 中断控制寄存器 0 的 DSALIGN 位可以配置。

在自动堆栈过程中，通过 INT_CTL0 中断控制寄存器 0 的 DSALIGN 中断自动堆栈双字/字对齐选择位可以选择堆栈 xSP 地址的对齐方式。xSP 有两种对齐方式，当 DSALIGN 为 0 时（默认为 0），xSP 为双字对齐；当 DSALIGN 为 1 时，xSP 为字对齐。

- 双字对齐

堆栈指针最后必须为 0x0 或 0x8 结束。在双字对齐的模式下，如果在中断进栈前指针没有双字对齐，则会自动下移指针（跳过一个字的空间），使指针双字对齐，然后会更新 SYS_MCTL 程序状态字寄存器中的 STACKING 位，将其置 1，并将 STACKING 的值堆栈保存；当中断返回时，将 STACKING 的值出栈恢复，然后会根据 STACKING 位的值恢复成入栈前的指针。

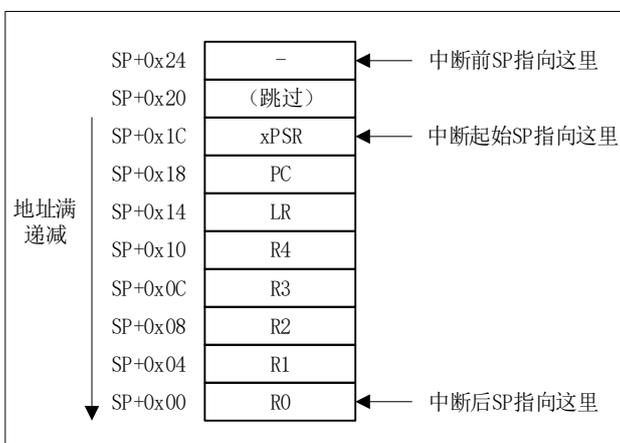


图 8-1 双字对齐堆栈方式

- 字对齐

堆栈指针最后必须为 0x0、0x4、0x8、0xC 结束。在字对齐模式下，中断入栈时 STACKING 位固定为 0，堆栈指针按字对齐。

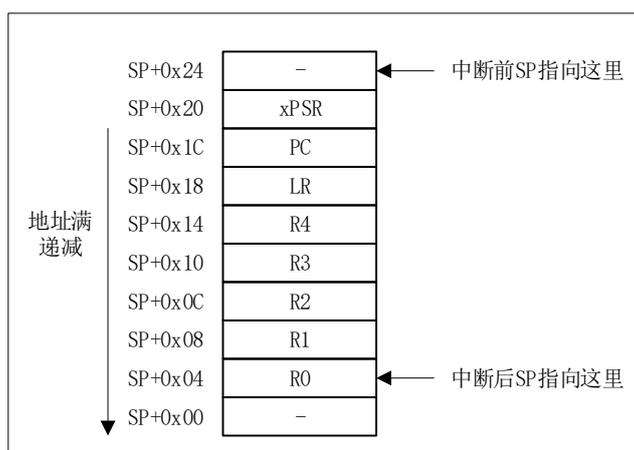


图 8-2 字对齐堆栈方式

注：MSP/PSP 低 2 位固定为 0。

8.4 堆栈指针的使用

芯片提供两个 SP 指针，一个为 MSP，即主堆栈指针；一个是 PSP，即进程堆栈指针。通过 SYS_MCTL 寄存器中的 SPSEL 当前有效堆栈指针位可以选择当前使用的指针类型。详见“中断堆栈指针使用”。

8.5 应用和复位控制

芯片内核提供一种软件复位模式，通过 SYS_ARCTL 应用和复位寄存器的 SYSTEMRST 位来控制。

SYSTEMRST 为系统复位请求。当该位置 1 时，会产生一个系统复位，该复位会将除 debug 电路之外的整个芯片复位。

芯片提供了两个 SP 指针用于用户操作，用户可通过 SPSEL 选择。

内核读写与 DMA 读写产生冲突时会进入自动仲裁机制。该仲裁机制可确保内核和 DMA 轮流获得总线的使用权。通过 CDMAAM 位可以选择仲裁的模式。当 CDMAAM 为 0 时只要内核与 DMA 同时发出了读写操作则判断为冲突，这可以适应需要较高运行频率的场合；当 CDMAAM 为 1 时只有当内核与 DMA 同时占用同一条接口总线时判断为冲突，这为低速高频次的 DMA 的传输提供了更高效的传输方式。RAM（RAM 和 DPRAM）和外设（含所有外设和 GPIO）分别有独立的总线接口。

内核读写操作预留了长周期读写模式。当 CRWDC 为 0 时，为长周期读写模式，此时针对外设读和写会分别增加 1 个周期，针对 BKP 域寄存器的读和写会分别增加 3 个周期，这对 RAM 的读写不变；当 CRWDC 为 1 时，所有的读写不会增加周期。

DMA 的读写操作预留了长周期读写模式。当 CRWDC 为 1 时，为普通模式，不会增加周期。当 CDMADC 为 0 时为长周期模式，此时 DMA 对 BKP 域的读和写分别增加 3 个周期。

为了避免误操作，在对 SYS_ARCTL 应用和复位控制寄存器进行设置时，需要向该寄存器的 ARKEY<31:16> 写入指定值 0x05FA，否则对该寄存器的写操作被忽略。读出 ARKEY<31:16> 的值为 0xFA05。

8.6 中断向量表重映射

中断产生并进入中断处理时，芯片会从中断向量表中获取入口地址并进入到相应的中断处理程序中。默认情况下中断向量表起始地址为 0x0000 0000，通过中断向量表重映射功能。用户可根据实际需要调整中断向量表的入口地址所在的空间。

通过 SYS_VECTOFF 中断向量表重映射控制寄存器设置重映射的起始地址。地址自动字对齐。中断向量表的重映射后新的向量入口地址计算如下：

$$\text{中断入口地址} = \text{SYS_VECTOFF} + \text{中断向量编号} * 4 \quad \text{公式 8-1}$$

8.7 RAM 空间指示

SYS_RAMSPA 寄存器用于堆栈空间错误检测，用户可用 SYS_RAMSPA 定义堆栈空间的最大地址，堆栈空间的最小地址默认从 0x1000 0000 开始，32 位寄存器 SYS_RAMSPA 的高 6 位默认为 000100。若堆栈进程中堆栈指针超出范围则会产生堆栈错误标志。

堆栈进程包括中断堆栈进程、PUSH/POP 堆栈进程。当实际使用的 xSP 大于等于

SYS_RAMSPA 时，判定为堆栈溢出，堆栈错误标志位置 1。由于中断和 PUSH/POP 对堆栈的操作中，进栈操作都是先对指针进行减操作，然后进行访存操作，而出栈都是先访存然后再对指针进行加操作，所以进栈最开始或者出栈最后的指针地址实际是不使用的，所以在进栈中初始指针地址可以等于 SYS_RAMSPA，而出栈后结束地址可以等于 SYS_RAMSPA。

堆栈错误发生时，尽管 xSP 指针错误，PUSH/POP 指令实际上仍继续执行，并使用当前的 xSP 作为堆栈指针，并将堆栈错误标志位置 1。当中断标志位置 1 后，将按照 PUSH/POP 遇到中断的处理方式，当 PUSH/POP 操作内容较多时，PUSH/POP 可能会被打断，优先处理中断进程。

堆栈错误发生时，尽管 xSP 指针错误，中断仍可进入，并使用当前的 xSP（无论溢出与否）作为堆栈指针保存数据（实际有可能无法正确保存），然后正常运行中断子程序。此时如无其他操作，则不会重复触发该中断。但中断返回前如若未能更正错误，则返回时会继续触发错误堆栈错误中断。

8.8 休眠与休眠唤醒

通过 SLEEP 指令可以让 CPU 进入休眠状态。当 CPU 处于休眠状态时，可以通过 SYS_MCTL 系统模式控制寄存器选择多种休眠和唤醒状态。

当 CPU 使用 SLEEP 指令时，CPU 会立即进入休眠模式。除此以外，CPU 还提供了另一种自动进入休眠的模式：中断退出时自动进入休眠模式，不管 CPU 在此前是否处于休眠状态。当 SLEEPONEXIT 置 1 时可以使能中断退出时自动进入休眠模式。该模式下允许 CPU 在休眠后只响应中断而不执行其他操作。在这种模式下工作，如果没有出现嵌套中断时，CPU 将忽略所有的进栈和出栈，缩短了中断的反应时间。

在不同的模式下有不同的唤醒模式。

当 CPU 处于普通休眠模式时，通过 SLPONMOD 唤醒模式选择位可以选择中断唤醒的源，根据不同的唤醒方式 CPU 在唤醒后有不同的处理方式。当 SLPONMOD 为 0 时，只有当挂起的有效中断（使能的中断）优先级比当前优先级高时才能唤醒 CPU，在这种情况下唤醒 CPU 后，CPU 会进入高优先级中断处理模式；当 SLPONMOD 为 1 时，任意挂起的中断都能唤醒 CPU，在这种情况下唤醒 CPU 后，如果触发唤醒的中断使能位使能且优先级更高，则进入高优先级处理模式，如果唤醒的中断优先级没有更高，则唤醒后继续休眠之前的程序运行。如果触发唤醒的中断没有使能，则唤醒 CPU 后将相应的标志位置 1，但不响应该中断。

当 CPU 处于深度休眠模式时，因 CPU 内部时钟停止，则情况与普通休眠有所不同。当 SLPONMOD 为 0 时，只有当挂起的有效中断（使能的中断）优先级比当前优先级高时，CPU 内部时钟会被唤醒，接着 CPU 会被唤醒，并执行中断处理程序。当 SLPONMOD 为 1 时，任意挂起的中断都可以唤醒时钟，如果该中断使能且优先级更高，则进入该中断处理模式，如果该中断使能且优先级低，则继续执行休眠前的程序。如果触发唤醒的中断没有使能，则唤醒 CPU 后将相应的标志位置 1，但不响应该中断。

表 8-8 休眠模式及休眠唤醒设置

休眠模式及休眠唤醒设置	SLEEPDEEP=0	SLEEPDEEP=1
SLPONMOD=0	普通休眠模式，只有挂起的有效中断优先级比当前优先级高时才能唤醒 CPU	深度休眠模式，只有当挂起的有效中断优先级比当前优先级高时，唤醒时钟，再唤醒 CPU

SLPONMOD=1	普通休眠模式，任意挂起的中断都能唤醒 CPU	深度休眠模式，可由任意挂起的中断唤醒时钟，之后与普通中断处理模式相同
------------	------------------------	------------------------------------

注：当 SLPONMOD 为 1 时，在进入休眠前请确认当前没有挂起的中断（即便是未使能的中断），否则将无法进入休眠模式。

8.9 程序空间控制设置

芯片可以通过 SYS_MEMCTL 程序控制寄存器的 MEMM<1:0> 选择存储空间的开始 512 字节的映射关系。默认中断向量表存放在开始的 512 字节空间内，通过不同的中断向量表的映射可以选择不同的复位地址，从而实现不同的启动位置启动。

每次系统复位后，根据 MEMM<1:0> 的值确定启动位置。芯片启动位置由复位地址确定，复位地址由中断向量表确定，而 MEMM<1:0> 可以选择默认中断向量表的映射关系。

表 8-9 向量表映射状态

MEMM<1:0>	程序执行状态
00	向量表映射为 ROM 空间
01	向量表映射为 RAM 空间
10	向量表映射为 FLASH 空间
11	内核不工作 (设置成该模式后需要将外部复位才能重新开始工作)

当 MEMM<10>=00 时，程序空间开始的 512 字节内容为 ROM 空间映射，此时所有对程序开始的 512 字节的访问都将访问 ROM 空间，由于中断向量表默认为存在程序起始空间，故复位地址由 ROM 确定。当 MEMM<10>=01 时，程序空间开始的 512 字节内容为 RAM 空间映射，复位地址由 RAM 确定。当 MEMM<10>=10 时，程序空间开始的 512 字节内容为 FLASH 空间映射，复位地址由 FLASH 确定。当 MEMM<10>=11 保留。

复位后，如果需要访问指定存储介质的起始 512 字节空间，需要修改 SYS_MEMCTL 寄存器的值。

8.10 超级用户模式使用

通过 SUSTA 位可以进入或退出超级用户模式。当 SUSTA 为超级用户模式时，程序拥有访问内核控制寄存器的权限；当程序为用户模式时，用户不能修改内核指定的控制寄存器，且读数为 0。

8.10.1 超级用户的进入和退出

复位状态下，系统默认为超级用户状态。中断处理模式下，系统始终为超级用户模式。通过 SUSTA 可以修改在线程模式下特权权限。当系统处于超级用户下，程序可以通过修改 SUSTA 位将特权模式从超级用户修改成用户模式。需要注意的是，该位本身也必须在超级用户模式下才能修改，故在线程模式下，一旦特权修改成用户权限之后，只能在中断程序中（如使用 SVC 指令）修改该位的值重新回到超级用户模式。

8.10.2 超级用户权限

超级用户模式下才允许访问的寄存器如下：

- 中断相关寄存器
- 系统控制相关寄存器（除 SYS_PSW 外）
- 节拍定时器相关寄存器

中断相关寄存器如下：

- 中断控制寄存器 INT_CTLx
- 中断使能寄存器 INT_EIEx
- 中断标志位寄存器 INT_EIFx
- 中断优先级控制寄存器 INT_IPx
- 外部中断屏蔽寄存器 INT_EINTMASK
- 外部中断上升沿选择寄存器 INT_EINTRISE
- 外部中断下降沿选择寄存器 INT_EINTFALL
- 外部中断中断标志位寄存器 INT_EINTF
- 外部中断源选择寄存器 INT_EINTSSx

系统控制相关寄存器如下：

- 系统模式控制寄存器 SYS_MCTL
- 应用和复位控制寄存器 SYS_ARCTL
- 中断向量表重映射控制寄存器 SYS_VECTOFF
- RAM 空间指示寄存器 SYS_RAMSPA
- 程序空间控制寄存器 SYS_MEMCTL

系统节拍定时器相关寄存器如下：

- 系统节拍定时器控制寄存器 ST_CTL
- 系统节拍定时器重加载寄存器 ST_RELOAD
- 系统节拍定时器计数器 ST_CV
- 系统节拍定时器校验寄存器 ST_CALI

超级用户下才能使用的指令：

MOV SYS,Ra: 用户模式下无效

9 DMA 控制器 (DMA0/DMA1)

9.1 概述

直接存储器访问模块(DMA)用于外设和存储器间直接数据传输,可用于 RAM 和 RAM 之间、RAM 和外设、外设和外设之间的数据传输。DMA 模块将从源地址上读取的数据写入到目标地址空间中,从而完成数据传输,而无需 CPU 的干预。

每个 DMA 模块有如下特性

- 7 个独立可配置的通道
- 支持存储器和存储器、存储器和外设、外设和外设之间的数据传输
- 支持 8bit/16bit/32bit 数据位宽传输
- 支持自动递增的源和目标地址,支持固定的源和目标地址
- 支持循环模式
- 支持传输数据数量设置,最大为 65535
- 支持 4 级通道优先级设置
- 支持外设触发,支持软件触发
- 追踪当前的源指针和目标指针
- 追踪当前未传输的数据量

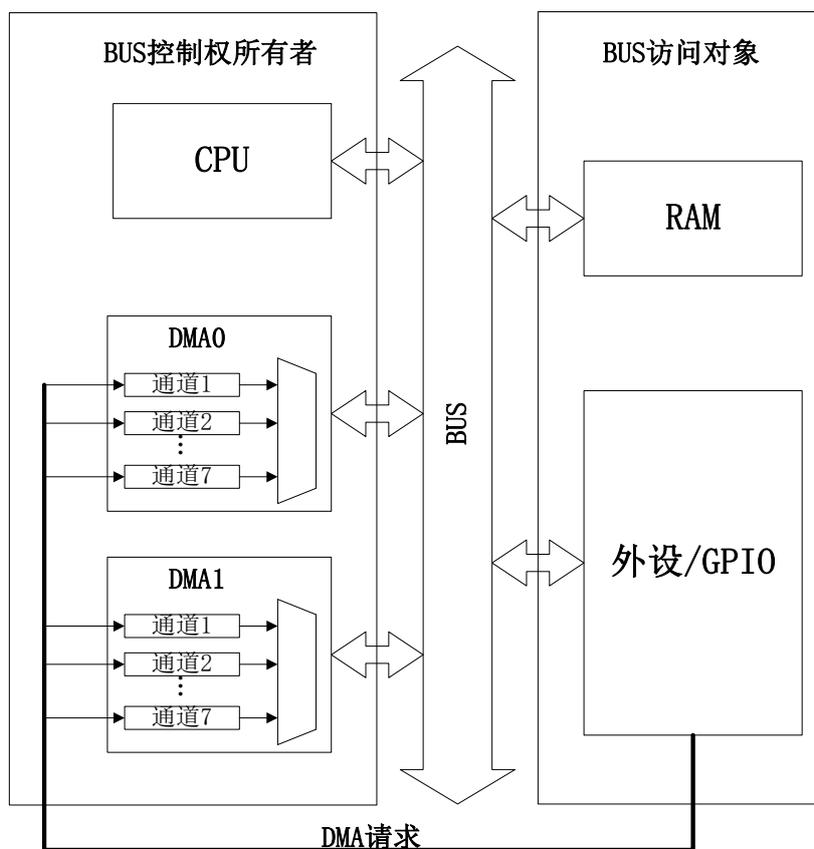


图 9-1 DMA 结构框图

9.2 DMA 相关寄存器

 表 9-1 DMA 相关寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x000	DMAx_CTLR1	R/W	DMAx 通道 1 控制寄存器	0x0002 0000
0x004	DMAx_CTLR2	R/W	DMAx 通道 2 控制寄存器	0x0002 0000
0x008	DMAx_CTLR3	R/W	DMAx 通道 3 控制寄存器	0x0002 0000
0x00C	DMAx_CTLR4	R/W	DMAx 通道 4 控制寄存器	0x0002 0000
0x010	DMAx_CTLR5	R/W	DMAx 通道 5 控制寄存器	0x0002 0000
0x014	DMAx_CTLR6	R/W	DMAx 通道 6 控制寄存器	0x0002 0000
0x018	DMAx_CTLR7	R/W	DMAx 通道 7 控制寄存器	0x0002 0000
0x020	DMAx_PADDR1	R/W	DMAx 通道 1 外设地址寄存器	0x0000 0000
0x024	DMAx_PADDR2	R/W	DMAx 通道 2 外设地址寄存器	0x0000 0000
0x028	DMAx_PADDR3	R/W	DMAx 通道 3 外设地址寄存器	0x0000 0000
0x02C	DMAx_PADDR4	R/W	DMAx 通道 4 外设地址寄存器	0x0000 0000
0x030	DMAx_PADDR5	R/W	DMAx 通道 5 外设地址寄存器	0x0000 0000
0x034	DMAx_PADDR6	R/W	DMAx 通道 6 外设地址寄存器	0x0000 0000
0x038	DMAx_PADDR7	R/W	DMAx 通道 7 外设地址寄存器	0x0000 0000
0x040	DMAx_MADDR1	R/W	DMAx 通道 1 存储器地址寄存器	0x0000 0000
0x044	DMAx_MADDR2	R/W	DMAx 通道 2 存储器地址寄存器	0x0000 0000
0x048	DMAx_MADDR3	R/W	DMAx 通道 3 存储器地址寄存器	0x0000 0000
0x04C	DMAx_MADDR4	R/W	DMAx 通道 4 存储器地址寄存器	0x0000 0000
0x050	DMAx_MADDR5	R/W	DMAx 通道 5 存储器地址寄存器	0x0000 0000
0x054	DMAx_MADDR6	R/W	DMAx 通道 6 存储器地址寄存器	0x0000 0000
0x058	DMAx_MADDR7	R/W	DMAx 通道 7 存储器地址寄存器	0x0000 0000
0x060	DMAx_CPAR1	R	DMAx 通道 1 当前外设地址寄存器	0x0000 0000
0x064	DMAx_CPAR2	R	DMAx 通道 2 当前外设地址寄存器	0x0000 0000
0x068	DMAx_CPAR3	R	DMAx 通道 3 当前外设地址寄存器	0x0000 0000
0x06C	DMAx_CPAR4	R	DMAx 通道 4 当前外设地址寄存器	0x0000 0000
0x070	DMAx_CPAR5	R	DMAx 通道 5 当前外设地址寄存器	0x0000 0000
0x074	DMAx_CPAR6	R	DMAx 通道 6 当前外设地址寄存器	0x0000 0000
0x078	DMAx_CPAR7	R	DMAx 通道 7 当前外设地址寄存器	0x0000 0000
0x080	DMAx_CMAR1	R	DMAx 通道 1 当前存储器地址寄存器	0x0000 0000
0x084	DMAx_CMAR2	R	DMAx 通道 2 当前存储器地址寄存器	0x0000 0000
0x088	DMAx_CMAR3	R	DMAx 通道 3 当前存储器地址寄存器	0x0000 0000
0x08C	DMAx_CMAR4	R	DMAx 通道 4 当前存储器地址寄存器	0x0000 0000
0x090	DMAx_CMAR5	R	DMAx 通道 5 当前存储器地址寄存器	0x0000 0000
0x094	DMAx_CMAR6	R	DMAx 通道 6 当前存储器地址寄存器	0x0000 0000
0x098	DMAx_CMAR7	R	DMAx 通道 7 当前存储器地址寄存器	0x0000 0000

¹ 本章除特殊说明以外, x=0,1, y=1~7。

偏移地址	寄存器	访问	功能描述	复位值
0x0A0	DMAx_NCT1	R	DMAx 通道 1 当前剩余数据寄存器	0x0000 0000
0x0A4	DMAx_NCT2	R	DMAx 通道 2 当前剩余数据寄存器	0x0000 0000
0x0A8	DMAx_NCT3	R	DMAx 通道 3 当前剩余数据寄存器	0x0000 0000
0x0AC	DMAx_NCT4	R	DMAx 通道 4 当前剩余数据寄存器	0x0000 0000
0x0B0	DMAx_NCT5	R	DMAx 通道 5 当前剩余数据寄存器	0x0000 0000
0x0B4	DMAx_NCT6	R	DMAx 通道 6 当前剩余数据寄存器	0x0000 0000
0x0B8	DMAx_NCT7	R	DMAx 通道 7 当前剩余数据寄存器	0x0000 0000
0x0C0	DMAx_LIFR	R/W	DMAx 通道 1~7 中断标志寄存器	0x0000 0000
0x0C4	DMAx_LIER	R/W	DMAx 通道 1~7 中断使能寄存器	0x0000 0000

DMA0 基地址：0x4000 1600

DMA1 基地址：0x4000 1700

9.2.1 DMAx_CTLRy DMAx 通道 y 控制寄存器

表 9-2 DMAx_CTLRy DMAx 通道 y 控制寄存器

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	NUMODT<15:0>															M2M	CPL<1:0>	ONESHOT	PDS<1:0>	MDS<1:0>	PLNC	MLNC	LPM	DDLR	BLKM			DMAEN				

NUMODT<15:0>：需要传输的数据个数

0x0000 = 需要传输 0 个数据

0x0001 = 需要传输 1 个数据

.....

0xFFFF = 需要传输 65535 个数据

M2M：存储器到存储器传输

0 = 未使能存储器到存储器传输

1 = 使能存储器到存储器传输

CPL<1:0>：通道优先级设置

00 = 低优先级

01 = 中优先级

10 = 高优先级

11 = 最高优先级

ONESHOT：单次触发模式使能

0 = 无作用

1 = 使能单次触发，硬件自动清零

PDS<1:0>：外设数据宽度选择

00 = 外设数据为 8 位宽

01 = 外设数据为 16 位宽

1x = 外设数据为 32 位宽

MDS<1:0>：存储器数据宽度选择

- 00 = 存储器数据为 8 位宽
- 01 = 存储器数据为 16 位宽
- 1x = 存储器数据为 32 位宽

PINC: 外设地址增量模式

- 0 = 外设地址增量模式关
- 1 = 外设地址增量模式开

MINC: 存储器地址增量模式

- 0 = 存储器地址增量模式关
- 1 = 存储器地址增量模式开

LPM: 循环模式

- 0 = 循环模式关
- 1 = 开启循环模式

DDIR: 数据方向控制

- 0 = 数据从外设地址 (PADDR) 读取, 存入存储器地址 (MADDR)
- 1 = 数据从存储器地址读取, 存入外设地址

BLKM: 数据块传输模式

- 0 = 数据按照单数据传输
- 1 = 数据按照块传输

DMAEN: DMA 通道 y 使能位

- 0 = 未使能 DMA 通道 y
- 1 = 使能 DMA 通道 y

9.2.2 DMAx_PADDRy DMAx 通道 y 外设地址寄存器

表 9-3 DMAx_PADDRy DMAx 通道 y 外设地址寄存器

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PADDR<31:0>																																

PADDR<31:0>: DMAx 通道 y 的起始外设地址

9.2.3 DMAx_MADDRy DMAx 通道 y 存储器地址寄存器

表 9-4 DMAx_MADDRy DMAx 通道 y 存储器地址寄存器

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	MADDR<31:0>																																

MADDR<31:0>: DMAx 通道 y 的起始存储器地址

9.2.4 DMAx_CPARy DMAx 通道 y 当前外设地址寄存器

表 9-5 DMAx_CPARy DMAx 通道 y 当前外设地址寄存器

复位值	R	0	31	R	0	30	R	0	29	R	0	28	R	0	27	R	0	26	R	0	25	R	0	24	R	0	23	R	0	22	R	0	21	R	0	20	R	0	19	R	0	18	R	0	17	R	0	16	R	0	15	R	0	14	R	0	13	R	0	12	R	0	11	R	0	10	R	0	9	R	0	8	R	0	7	R	0	6	R	0	5	R	0	4	R	0	3	R	0	2	R	0	1	R	0	0
R/W	R	0	31	R	0	30	R	0	29	R	0	28	R	0	27	R	0	26	R	0	25	R	0	24	R	0	23	R	0	22	R	0	21	R	0	20	R	0	19	R	0	18	R	0	17	R	0	16	R	0	15	R	0	14	R	0	13	R	0	12	R	0	11	R	0	10	R	0	9	R	0	8	R	0	7	R	0	6	R	0	5	R	0	4	R	0	3	R	0	2	R	0	1	R	0	0
位名	CPAR<31:0>																																																																																															

CPAR<31:0>: DMAx 通道 y 的当前外设地址

9.2.5 DMAx_CMARy DMAx 通道 y 当前存储器地址寄存器

表 9-6 DMAx_CMARy DMAx 通道 y 当前存储器地址寄存器

复位值	R	0	31	R	0	30	R	0	29	R	0	28	R	0	27	R	0	26	R	0	25	R	0	24	R	0	23	R	0	22	R	0	21	R	0	20	R	0	19	R	0	18	R	0	17	R	0	16	R	0	15	R	0	14	R	0	13	R	0	12	R	0	11	R	0	10	R	0	9	R	0	8	R	0	7	R	0	6	R	0	5	R	0	4	R	0	3	R	0	2	R	0	1	R	0	0
R/W	R	0	31	R	0	30	R	0	29	R	0	28	R	0	27	R	0	26	R	0	25	R	0	24	R	0	23	R	0	22	R	0	21	R	0	20	R	0	19	R	0	18	R	0	17	R	0	16	R	0	15	R	0	14	R	0	13	R	0	12	R	0	11	R	0	10	R	0	9	R	0	8	R	0	7	R	0	6	R	0	5	R	0	4	R	0	3	R	0	2	R	0	1	R	0	0
位名	CMAR<31:0>																																																																																															

CMAR<31:0>: DMAx 通道 y 的当前存储器地址

9.2.6 DMAx_NCTy DMAx 通道 y 当前剩余数据寄存器

表 9-7 DMAx_NCTy DMAx 通道 y 当前剩余数据寄存器

复位值	R	0	31	R	0	30	R	0	29	R	0	28	R	0	27	R	0	26	R	0	25	R	0	24	R	0	23	R	0	22	R	0	21	R	0	20	R	0	19	R	0	18	R	0	17	R	0	16	R	0	15	R	0	14	R	0	13	R	0	12	R	0	11	R	0	10	R	0	9	R	0	8	R	0	7	R	0	6	R	0	5	R	0	4	R	0	3	R	0	2	R	0	1	R	0	0
R/W	R	0	31	R	0	30	R	0	29	R	0	28	R	0	27	R	0	26	R	0	25	R	0	24	R	0	23	R	0	22	R	0	21	R	0	20	R	0	19	R	0	18	R	0	17	R	0	16	R	0	15	R	0	14	R	0	13	R	0	12	R	0	11	R	0	10	R	0	9	R	0	8	R	0	7	R	0	6	R	0	5	R	0	4	R	0	3	R	0	2	R	0	1	R	0	0
位名	NCT<31:16>																																																																																															

NCT<31:16>: DMAx 通道 y 的当前剩余数据量

9.2.7 DMAx_LIFR DMAx 通道 1~7 中断标志寄存器

表 9-8 DMAx_LIFR DMAx 通道 1~7 中断标志寄存器

复位值	0	31	0	30	0	29	0	28	0	27	0	26	0	25	0	24	0	23	0	22	0	21	0	20	0	19	0	18	0	17	0	16	0	15	0	14	0	13	0	12	0	11	0	10	0	9	0	8	0	7	0	6	0	5	0	4	0	3	0	2	0	1	0	0																			
R/W																					R/W	0	20	R/W	0	19	R/W	0	18	R/W	0	17	R/W	0	16	R/W	0	15	R/W	0	14	R/W	0	13	R/W	0	12	R/W	0	11	R/W	0	10	R/W	0	9	R/W	0	8	R/W	0	7	R/W	0	6	R/W	0	5	R/W	0	4	R/W	0	3	R/W	0	2	R/W	0	1	R/W	0	0
位名																					ETIF7	HTIF7	FTIF7	ETIF6	HTIF6	FTIF6	ETIF5	HTIF5	FTIF5	ETIF4	HTIF4	FTIF4	ETIF3	HTIF3	FTIF3	ETIF2	HTIF2	FTIF2	ETIF1	HTIF1	FTIF1																																										

ETIFy: 错误传输中断标志位, 该位写 0 清零, 写 1 无效。

- 0 = 对应通道的 DMA 传输无错误
- 1 = 对应通道的 DMA 传输发生错误

HTIFy: 半传输中断标志位, 该位写 0 清零, 写 1 无效。

- 0 = 对应通道的 DMA 传输未完成一半

1 = 对应通道的 DMA 传输已完成一半

FTIFy: 完成传输中断标志位, 该位写 0 清零, 写 1 无效。

0 = 对应通道的 DMA 传输未完成

1 = 对应通道的 DMA 传输已完成

9.2.8 DMAx_LIER DMAx1~7 通道中断使能寄存器

表 9-9 DMAx_LIER DMAx1~7 通道中断使能寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W												R/W																				
位名												ETIE7	HTIE7	FTIE7	ETIE6	HTIE6	FTIE6	ETIE5	HTIE5	FTIE5	ETIE4	HTIE4	FTIE4	ETIE3	HTIE3	FTIE3	ETIE2	HTIE2	FTIE2	ETIE1	HTIE1	FTIE1

ETIEy: 错误传输中断使能位

0 = 未使能对应通道的 DMA 错误传输中断

1 = 使能对应通道的 DMA 错误传输中断

HTIEy: 半传输中断使能位

0 = 未使能对应通道的 DMA 传输完成一半中断

1 = 使能对应通道的 DMA 传输完成一半中断

FTIEy: 完成传输中断使能位

0 = 未使能对应通道的 DMA 传输完成中断

1 = 使能对应通道的 DMA 传输完成中断

9.3 DMA 功能

DMA 可以独立于 CPU 使用读写总线，实现外设/GPIO 和 RAM 的之间数据的传输。

DMA 使用系统时钟 SCLK 作为工作时钟，因此可以实现高速且高效的数据传输。DMA 可实现以下的传输模式：

- 外设/GPIO 和 RAM 之间
- 外设/GPIO 和外设/GPIO 之间
- RAM 和 RAM 之间

当外设使能 DMA 传输功能后，会发出对应的 DMA 请求。DMA 请求会发送到对应的通道。DMA 接收到对应的请求后会可以使用总线进行数据传输。

DMA 支持单数据传输、块传输和循环传输模式。通过 DMAx_CTLRy 寄存器的 BLKM 可以选择单数据传输模式或块传输模式，通过 DMAx_CTLRy 寄存器的 LPM 可以使能循环模式。通过 DMAx_CTLRy 寄存器的 NUMDOT<15:0>可以设置需要传输的数据，所设置的数为一个块。通过 DMAx_NCTy 可以读出当前剩余的数据量。

通过 DMAx_PADDRy 和 DMAx_MADDRy 可以设置外设地址和存储器地址。这里外设地址和存储器地址只用于区分源地址和目标地址，通过 DMAx_CTLRy 寄存器的 DDIR 位可以确定源地址和目标地址。DMA 支持不同数据位宽的操作模式，通过 DMAx_CTLRy 寄存器的 PDS<1:0>和 MDS<1:0>可以设置外设数据宽度和存储器数据宽度。DMA 支持地址递增模式，通过 DMAx_CTLRy 寄存器的 PINC 和 MINC 可以设置外设地址递增和存储器地址递增。通过 DMAx_CPARy 和 DMAx_CMARy 可以读出当前的外设地址和存储器地址。

DMAx_CTLRy 寄存器的 M2M 位为存储器到存储器传输使能，也可以看成是软件触发位。

通过 DMAx_CTLRy 寄存器的 CPL<1:0>可以设置 DMA 的优先级。所有的 DMA 通道同一个时间只能响应一个通道，多通道同时发出请求时，优先级高的优先响应，优先级相同的通道编号小的优先响应。

通过 DMAx_CTLRy 寄存器的 DMAEN 可以使能相应的 DMA 通道。当 DMAx_CTLRy 寄存器的 NUMDOT<15:0>变为 0 时，视为传输结束。非循环模式下 DMAEN 会硬件清零，如需重新使用 DMA，需要软件重新使能；循环模式下，DMAEN 不会硬件清零。

通过 DMAx_LIER 可以使能 DMA 中断，通过 DMAx_LIFR 可以得到 DMA 的中断标志状态。

9.4 DMA 请求

从外设产生的多个请求经过逻辑或输出到 DMA 控制器，同时只有一个请求有效。外设的 DMA 请求可以通过设置相应外设寄存器中的 DMA 控制位被独立的开启或关闭。各个通道的 DMA 请求分布如下表所示。

表 9-10 DMA0 请求

外设	通道1	通道2	通道3	通道4	通道5	通道6	通道7
USART0	USART0_TX	USART0_RX					
USART1			USART1_TX	USART1_RX			
USART2					USART2_TX	USART2_RX	
USART3	USART3_RX						USART3_TX
USART4				USART4_TX	USART4_RX		
USART5		USART5_TX	USART5_RX				
USART6							
USART7							
I2C				I2C0_TX	I2C0_RX	I2C1_TX	I2C1_RX
ADC0	ADC0H0	ADC0H1	ADC0H2	ADC0H3	ADC0N		
ADC1							
ADC2							
SPI0	SPI0_TX	SPI0_RX					
SPI1			SPI1_TX	SPI1_RX			
SPI2					SPI2_TX	SPI2_RX	
SPI3						SPI3_TX	SPI3_RX
TIMER0		TM0UP	TM0TRIG	CCP0CH1	CCP0CH2	CCP0CH3	CCP0CH4
TIMER1	TM1UP	TM1TRIG	CCP1CH1	CCP1CH2	CCP1CH3	CCP1CH4	
TIMER2	CCP2CH3	CCP2CH4		TM2UP	TM2TRIG	CCP2CH1	CCP2CH2
TIMER3	CCP3CH4		TM3UP	TM3TRIG	CCP3CH1	CCP3CH2	CCP3CH3
TIMER4		TM4TRIG	CCP4CH1	CCP4CH2	CCP4CH3	CCP4CH4	TM4UP
TIMER5	CCP5CH2	CCP5CH3	CCP5CH4	CCP5CLS	TM5UP	TM5TRIG	CCP5CH1
TIMER6						TM6UP	
TIMER9							
TIMER10							
TIMER14	TM14TRGI						TM14UP
TIMER15		TM15UP	TM15TRGI				
TIMER18							
TIMER19							
TIMER20	CCP20CH4		TM20UP	TM20TRIG	CCP20CH1	CCP20CH2	CCP20CH3
TIMER21							
TIMER22							
TIMER23							
DAC	DAC1TRGI				DAC0TRGI		
EXIC							
CAN0-2	CAN0_TX	CAN0_RX	CAN1_TX	CAN1_RX	CAN2_TX	CAN2_RX	

表 9-11 DMA1 请求

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
USART0							
USART1							
USART2							
USART3							
USART4							
USART5							
USART6			USART6_TX	USART6_RX			
USART7					USART7_TX	USART7_RX	
I2C			I2C2_TX	I2C2_RX	I2C3_TX	I2C3_RX	
ADC0							
ADC1	ADC1H0	ADC1H1	ADC1H2	ADC1H3	ADC1N		
ADC2			ADC2H0	ADC2H1	ADC2H2	ADC2H3	ADC2N
SPI0							
SPI1							
SPI2							
SPI3							
TIMER0							
TIMER1							
TIMER2							
TIMER3							
TIMER4							
TIMER5							
TIMER6							
TIMER9	TM9TRIG	CCP9CH1	CCP9CH2	CCP9CH3	CCP9CH4	CCP9CLS	TM9UP
TIMER10	TM10UP						
TIMER14							
TIMER15							
TIMER18	TM18UP	TM18TRIG	CCP18CH1	CCP18CH2	CCP18CH3	CCP18CH4	
TIMER19		TM19UP	TM19TRIG	CCP19CH1	CCP19CH2	CCP19CH3	CCP19CH4
TIMER20							
TIMER21	CCP21CH3	CCP21CH4		TM21UP	TM21TRIG	CCP21CH1	CCP21CH2
TIMER22	CCP22CH2	CCP22CH3	CCP22CH4		TM22UP	TM22TRIG	CCP22CH1
TIMER23	CCP23CH1	CCP23CH2	CCP23CH3	CCP23CH4		TM23UP	TM23TRIG
DAC							
EXIC	RDMA	TDMA					

9.5 DMA 触发

DMA 模块可以用于存储器 and 存储器之间、存储器和外设之间的数据传输，无需 CPU 的干预。

当外设中断满足 DMA 触发条件并发出 DMA 请求时，若此时 DMAx_CTLRy 寄存器的 DMAEN 置 1 使能对应的 DMA 通道，外设请求就可以触发 DMA 传输。

通过 DMAx_CTLRy 寄存器的 M2M 位可以作为软件触发位，触发 DMA 传输。该位置 1 后需要软件清零才会停止继续触发 DMA 传输。

通过 DMAx_CTLRy 寄存器的 ONESHOT 单次触发模式使能位，可以软件产生一次的 DMA 触发，该位在置 1 后会硬件自动清零。

每一个外设对 DMA 的触发都只能触发对应的 DMA 通道，DMAx_CTLRy 寄存器的 M2M 位可以软件触发相应的 DMA 通道，该位常用于存储器到存储器间的传输。

9.6 数据位宽

DMA 模块每一个通道都可以单独设置传输数据的位宽。外设地址空间的数据位宽和存储器地址空间的数据位宽分别通过 DMAx_CTLRy 寄存器的 PDS<1:0>位和 MDS<1:0>位来控制，8 位/16 位/32 位位宽可选，默认传输位宽为 8 位。

外设地址和存储器地址都可以分别作为源地址和目标地址。当作为源地址时候，数据位宽为 8 时数据的低 8 位有效；数据位宽为 16 时数据的低 16 位有效；数据位宽为 32 时数据的 32 位有效。当作为目标数据时，数据位宽为 8 时，只修改目标地址的 8 位数据，其它保持；数据位宽为 16 时，只修改目标地址的 16 位数据，其它保持；数据位宽为 32 时，修改目标地址的 32 位数据。

9.7 传输地址

通过 DMAx_PADDRy 和 DMAx_MADDRy 可以设置外设地址和存储器地址。这里外设地址和存储器地址只用于区分源地址和目标地址，通过 DMAx_CTLRy 寄存器的 DDIR 位可以确定传输方向，即源地址和目标地址，这里操作都是从源地址读取数据写到目标地址中。

DMA 支持地址递增模式，通过 DMAx_CTLRy 寄存器的 PINC 和 MINC 可以设置外设地址递增和存储器地址递增。当地址为非递增方式时，DMA 传输的数据地址就是 DMAx_PADDRy 或 DMAx_MADDRy 指定的地址。

当 DMA 设置成递增模式时，会将 DMAx_PADDRy 和 DMAx_MADDRy 设置的地址作为初始地址进行递增操作，递增的值与位宽相关，且 DMAx_PADDRy 和 DMAx_MADDRy 这两个地址的递增是分开独立计算的。当数据位宽为 8 位时，地址递增加 1；当数据位宽为 16 位时，地址递增加 2；当数据位宽为 32 位时，地址递增加 4。

通过 DMAx_CPARy 和 DMAx_CMARy 可以读出当前的外设地址和存储器地址。

9.8 数据量和数据空间

通过 DMAx_CTLRy 寄存器的 NUMODT 位可以设置需要传输的数据量。该寄存器和自动增模式、源/目标地址共同决定了传输需要的数据空间。当传输开始时，NUMODT 的值加载到传输数量计数器中 (DMAx_NCTy) 执行减计数，当 DMAx_NCTy 为 0 时表明完成传输完成，通过 DMAx_NCTy 寄存器可以得到当前尚未传输的数据量。

9.9 单数据/块传输和循环模式

单数据传输为每次触发只传输一个数据，块传输为每次触发会将 DMAx_CTLRy 寄存器的 NUMODT 设置的数据量全部传输。通过 DMAx_CTLRy 寄存器的 BLKM 可以选择单数据传输或者块传输的模式。

循环模式为只要触发之后，就会持续传输数据。在非循环模式下，当 DMAx_CTLRy 寄存器的 NUMODT 设置的数据全部传输后（即 DMAx_NCTy 计数为 0 时），硬件会自动将 DMAEN 位清零，关闭 DMA 的功能，如需重新使用 DMA 传输，需要软件使能 DMAEN。在循环模式下，硬件不会将 DMAEN 清零，当 NUMODT 设置的数据全部传输后（即 DMAx_NCTy 计数为 0 时），会继续重新开始传输，直到软件关闭 DMAEN 为止。通过 DMAx_CTLRy 寄存器的 LPM 位可以使能循环模式。

9.10 优先级

不同的 DMA 通道之间可以通过优先级来区分哪一个通道的请求优先响应。优先级机制确保在同时存在多个通道的请求时，通道的响应顺序能够按照用户的需要进行调整。

单个 DMA 模块使用时，优先级高的通道会优先响应，优先级相同的时候通道编号小的优先响应；多个 DMA 模块之间的优先级能够同时判断，多个 DMA 模块间优先级最高的通道优先响应，优先级相同的情况下，DMA0 的优先级大于 DMA1 的优先级，以此类推。

优先级可通过 DMAx_CTLRy 寄存器的 CPL<1:0>位来控制，可以设置成 4 种不同的优先级，其中优先级 3 为最高优先级，优先级 0 为最低优先级。

若当前正有一个 DMA 工作时，其他的 DMA 模块需要等待当前 DMA 通道工作结束才能开始工作，无论优先级高低。若当前没有 DMA 工作时，多个 DMA 同时发出请求时，按照优先级响应优先级最高的通道。

当多个 DMA 同时使用连续模式，且优先级不同时，由于 DMA 结束到下一次开始有两个时钟的延时情况，在这段延时如有其它 DMA 请求，则会响应其它的 DMA 请求，所以实际工作会出现两个不同优先级的 DMA 轮流使用的情况。但这种情况也会受到 CPU 执行的影响，当 CPU 执行访存与 DMA 冲突导致 DMA 响应延时，此时由于高优先级通道已经重新发出请求，所以会继续高优先级通道的工作。

9.11 中断

每个 DMA 通道都支持传输完成中断、传输过半中断和传输错误中断。通过寄存器 DMAx_IER 寄存器中的 FTIEy、HTIEy、ETIEy 可以分别设置相应的中断使能，当 DMAx_NCTy 计数器的值减少到设置值的一半时，对应通道的 HTIFy 置 1；当 DMAx_NCTy 计数器的值减到 0 时，对应通道的 FTIFy 置 1；当 DMA 操作的地址空间不属于 RAM 或 SFR 范围（DMA 可操作空间）时，则会产生传输错误中断，将对应通道的 ETIFy 标志位置 1。注意，标志位为写 0 清零，写 1 无效。

DMA 允许操作空间为 RAM(0x1000_0000 到 0x1FFE_FFFF)、SFR (0x4000_0000 到 0x4007_FFFF)、SFR (0x5000_0000 到 0x501F_FFFF)，其他空间暂不支持。

若 DMA 中断产生，且 DMA 全局中断(DMAxIE)使能，并使能全局中断信号 AIE 时，CPU 可以响应该 DMA 中断。

9.12 DMA 独立于 CPU 运行

由于 DMA 可以独立于 CPU 工作，DMA 可以工作在正常运行模式、低功耗运行模式、普通休眠模式、低功耗休眠模式，为低功耗下数据转移提供了高效支持。

9.13 DMA 与 CPU 冲突仲裁

DMA 可以独立于 CPU 工作，当 DMA 与 CPU 同时访问会出现总线冲突的情况，这个时候会进入 DMA 仲裁。该仲裁机制可确保内核和 DMA 轮流获得总线的使用权。

通过 SYS_ARCTL 寄存器 CDMAAM 位可以选择仲裁的模式。当 CDMAAM 为 0 时只要内核与 DMA 同时发出了读写操作则判断为冲突，这可以适应需要较高运行频率的场合；当 CDMAAM 为 1 时只有当内核与 DMA 同时占用同一条接口总线时判断为冲突，这为高频率的 DMA 的传输提供了更高效的传输方式。

冲突关系如表所示。

表 9-12 DMA 与 CPU 冲突关系（高速度模式）

	DMA 访问	无 DMA 访问
CPU 访问	冲突	CPU 访问
无 CPU 访问	DMA 访问	-

表 9-13 DMA 与 CPU 冲突关系（高效模式）

	DMA 访问 RAM	DMA 访问外设/GPIO
CPU 访问 RAM	冲突	同时访问
CPU 访问外设/GPIO	同时访问	冲突

仲裁条件如下：

- 当 CPU 执行指令访存与 DMA 冲突时，CPU 与 DMA 轮流执行，后执行的等待先执行的执行完成后进行轮换。不管是 CPU 还是 DMA，等待期间会暂停相应的操作。
- 当 CPU 处于中断进栈或者出栈的过程中，CPU 默认有更高优先权，DMA 处于等待状态
- 当 CPU 处于取值状态（如程序在 RAM 中运行且 DMA 对 RAM 操作）时，CPU 默认有更高优先权，DMA 处于等待状态

9.14 使用 DMA

DMA 使用系统时钟 SCLK 作为工作时钟，一般而言，按照下列顺序来配置 DMA 通道：

- (1) 配置对应通道的外设地址寄存器；
- (2) 配置对应通道的存储器地址寄存器；
- (3) 配置对应通道的控制寄存器，设置需要传输的数据量、方向、优先级和位宽等；
- (4) 使能 DMA 通道；
- (5) 在非循环模式下，DMAx_NCTy 计数为 0 时，需要重新使能 DMAEN 位

例：如果希望 USART0 发送的数据由 DMA 提供，通过查询表“DMA 请求”可以知道 USART0 发送对应的是 DMA0 通道 1。配置外设地址寄存器 DMA0_PADDR1 为 USART0 发送寄存器地址 0x4000_098C。存储器寄存器地址可被配置为 RAM 中的任意地址。

设置控制寄存器 DMA0_CTLR1 为 0x0003_0F40，则 DMA0 总共会传输 3 个数据到 USART0，优先级为最低级，发送和接收的数据位宽都为 32 位，存储器寄存器地址自增。

使能 DMA0_CTLR1 寄存器中的 DMAEN 位后，一旦 USART0 触发 DMA 请求，DMA0 就可以传输数据了。

10 节拍定时器 (SYSTICK)

10.1 概述

KungFu32 内核提供了一个 24 位的系统节拍定时器 (System Tick Timer)。系统节拍定时器可为系统提供可编程时长的周期性中断，即使是在休眠下也能工作。系统节拍定时器有专用的中断向量。

系统节拍定时器结构如下图所示。系统节拍定时器为递减计数模式，当系统节拍定时器的值为 0 时会产生一个中断，同时系统节拍定时器重载值寄存器 (ST_RELOAD) 的值会装入系统节拍定时器中。对系统节拍定时器重载值寄存器 (ST_RELOAD) 进行设置可以修改产生中断的间隔时长。在使用节拍定时器时，使能前要先向 ST_CV 系统节拍定时器当前值寄存器写任意值，使 COUNTZERO 位及 ST_CV 清零，保证 ST_RELOAD 的值加载到 ST_CV 中。

向 ST_RELOAD 写 0 会使计数器在下一个计数周期禁止。

通过使能 INT_EIE0 寄存器的 SYSTICKIE 位可以使能系统节拍定时器中断，当定时器由 1 变 0 时可以将 INT_EIF0 中的 SYSTICKIF 标志位置 1。

10.2 节拍定时器结构框图

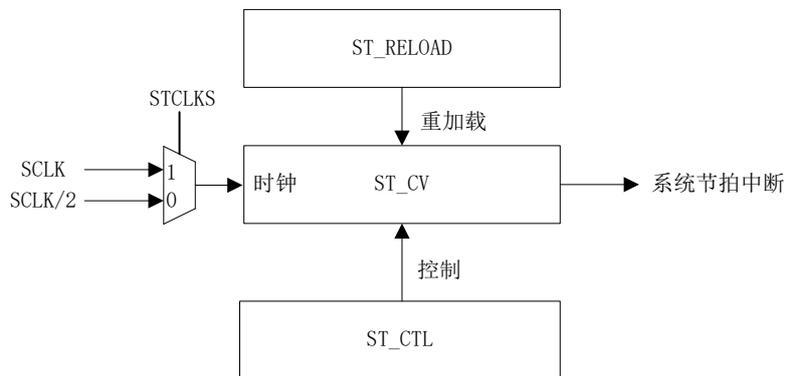


图 10-1 节拍定时器结构框图

10.3 相关寄存器

表 10-1 系统节拍定时器相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x0A0	ST_CTL	R/W	系统节拍定时器控制和状态寄存器	0x0000 0000
0x0A4	ST_RELOAD	R/W	系统节拍定时器重载值寄存器	0x00uu uuuu
0x0A8	ST_CV	R/W	系统节拍定时器当前值寄存器	0x00uu uuuu
0x0AC	ST_CALI	R/W	系统节拍定时器校验寄存器	0xC000 0000

基地址：0x4020 0000

10.3.1 ST_CTL 系统节拍定时器控制寄存器

表 10-2 ST_CTL 寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值																																
R/W	R																R/W										R/W	R/W	R/W			
位名	COUNTZERO																										STCLKS	TICKINTEN	STEN			

COUNTZERO: 定时器计数到 0 时该位置 1，读 ST_CTL 寄存器或写 ST_CV 清零该位。

STCLKS: 系统节拍定时器时钟源选择。该位与 STCLKREF 共同作用，任意一位值为 1 时选择 SCLK 作为时钟。

0 = SCLK/2 作为时钟

1 = SCLK 作为时钟

TICKINTEN: 使能 SYSTICK 中断

0 = 计数到 0 不发送 SYSTICK 中断请求

1 = 计数到 0 发送 SYSTICK 中断请求

STEN: 系统节拍定时器使能位

0 = 禁止系统节拍定时器

1 = 使能系统节拍定时器

10.3.2 ST_RELOAD 系统节拍定时器重加载寄存器

表 10-3 ST_RELOAD 系统节拍定时器重加载寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值																																
R/W	R/W								R/W																							
位名									STRELOAD<23:0>																							

STRELOAD<23:0>: 系统节拍定时器重加载寄存器，当计数器减到 0 时，将此寄存器值加载到定时器上。内核时钟为 100MHz 时，该寄存器配置值为 0x0098_9680 可以得到 100ms 定时。

10.3.3 ST_CV 系统节拍定时器计数器

表 10-4 ST_CV 系统节拍定时器寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值																																
R/W	R/W								R/W																							
位名									STCV<23:0>																							

STCV<23:0>: 系统节拍定时器计数器，读该寄存器可以得到当前计数值，软件写该寄存器时候清零该寄存器。在初始化时，要先向该寄存器写入任意值，确保 COUNTZERO 清零和 STRELOAD 值能加载到该寄存器中。

10.3.4 ST_CALI 校验寄存器

表 10-5 ST_CALI 校验寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	1	1	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	STCLKREF	SKEW							STCALIB<23:0>																								

STCLKREF: 系统时钟选择控制, 该位与校验值配合使用。该位与 ST_CTL 寄存器的 STCLKS 位共同控制选择结果, 两者任意一位值为 1 时选择 SCLK 作为时钟。

0 = 无作用 (由 STCLKS 确定选择结果)

1 = 选择 SCLK 作为节拍定时器的时钟

SKEW: 校准值偏差标志位, 该位在写入校验值时一同写入。

0 = 校验值不是 10ms

1 = 校验值是 10ms

STCALIB<23:0>: 系统节拍定时器校验寄存器, 该寄存器用于存放芯片出厂的 10ms 校验值。

10.4 节拍定时器使用

节拍定时器可按照下列操作步骤使用:

- (1) ST_CTL 的 STEN 写 0 禁止节拍定时器, 防止产生意外的中断请求。
- (2) 向 ST_RELOAD 重加载寄存器写入节拍定时器的周期值, 周期值为重加载值加 1。
- (3) 向 ST_CV 当前值寄存器写入任意值, 以清零当前值寄存器并清零 COUNTZERO 标志位, 确保重载值能加载到当前值寄存器中。
- (4) 使能 ST_CTL 寄存器中 TICKINTEN 位 (如果需要中断请求), 通过 ST_CTL 寄存器中的 STCLKS 位和 ST_CALI 寄存器中的 STCLKREF 位选择需要的时钟, 最后使能 STEN 位。
- (5) 节拍定时器的时钟由 ST_CTL 寄存器的 STCLKS 和 ST_CALI 寄存器的 STCLKREF 共同决定, 当有任意一个寄存器选择 SCLK 作为时钟时节拍定时器使用 SCLK 作为时钟。

节拍定时在计数到 0 的时候会将 ST_CTL 寄存器的 COUNTZERO 位置 1, 读 ST_CTL 寄存器或写 ST_CV 可以清零该位。

10.5 中断请求

将 ST_CTL 寄存器的 TICKINTEN 置 1, 可以使能节拍定时器中断请求。节拍定时在计数到 0 的时候会发送 SYSTICK 中断请求脉冲, 使 INT_EIF0 中断标志位寄存器的 SYSTICKIF 中断标志位置 1。如果同时将 INT_EIE0 寄存器的 SYSTICKIE 置 1, 会使能节拍定时器中断, SYSTICKIF 标志位将在中断响应后自动清零。

电路中, 当 ST_CV 的值为 0 时就会产生中断脉冲, 并产生 COUNTZERO 置位的脉冲。在 ST_CV 为 0 的同时, ST_CTL 寄存器 COUNTZERO 置 1, 且 INT_EIF0 寄存器的 SYSTICKIF 位置 1。

注: ST_CTL 寄存器的 COUNTZERO 位与中断功能无关, 仅仅是 ST_CV 计数到 0 时的

标志，读 ST_CTL 寄存器或写 ST_CV 清零该位。

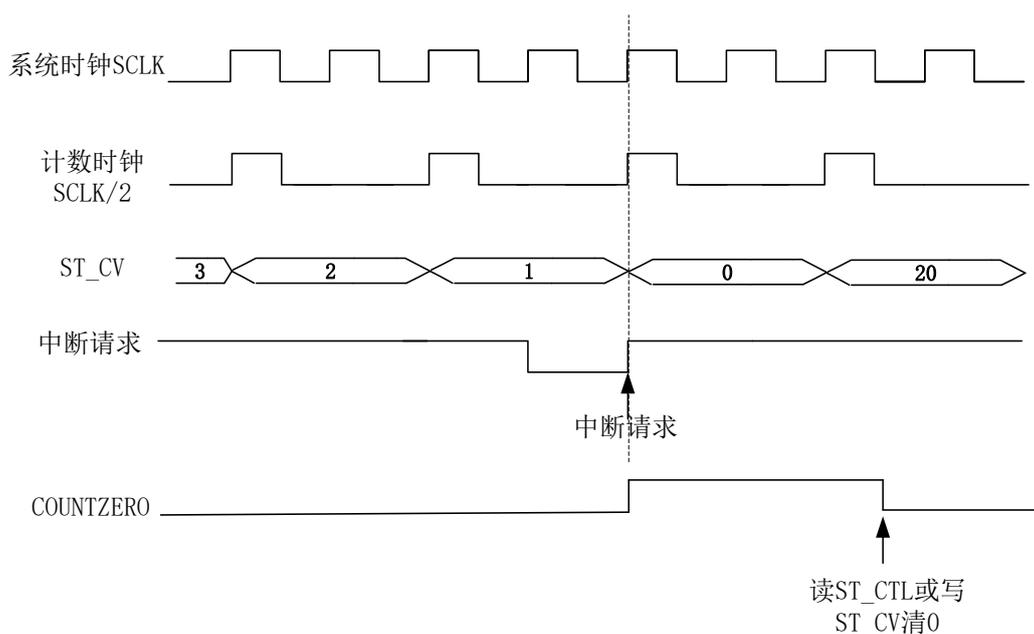


图 10-2 重载值 0x20, SCLK/2 作为计数时钟, 使能 TICKINTEN

11 基本定时/计数器(T14/T15)

11.1 概述

Tx(x=14,15)是一个 16 位的定时/计数器，它有定时和计数两种工作模式，支持 3 种计数方式：向上计数、向下计数和向上向下计数方式。根据不同的模式，计数会产生溢出，将 Tx 溢出中断标志 TXIF 位置 1。

基本定时器主要功能包括：

16 位自动重载计数器

16 位可编程预分频器，用于对输入的时钟按系数为 1~65536 之间任意数值分频在更新事件以及触发事件时产生 DMA 请求

基本定时器可以用于触发 AD 和 DA 模块

11.2 基本定时器结构框图

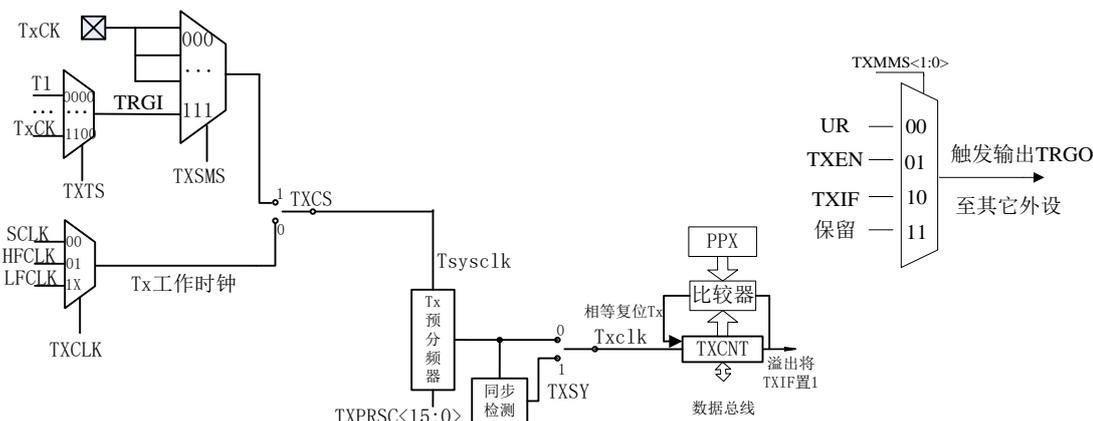


图 11-1 基本定时器结构框图

- TxCK 引脚：外部计数引脚，模块工作在计数模式 1 时，从该引脚输入外部时钟信号。计数模式 2 时，如果 Tx_CTL2 寄存器的 TXTS<2:0>=1100，则 TxCK 引脚作为模块的触发输入 TRGI。该引脚从 IO 口通过重映射功能输入。
- Txclk：计数时钟，用于计数器计数
- Tsysclk：计数时钟源，用于输入分频器以产生计数时钟。

11.3 基本定时器相关寄存器

 表 11-1 基本定时器相关寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x000	Tx_CNT	R/W	TXCNT 寄存器	0x0000 0000
0x004	Tx_CTL1	R/W	Tx 控制寄存器 1	0x0000 0100
0x008	Tx_CTL2	R/W	Tx 控制寄存器 2	0x0000 0000
0x00C	Tx_PRSC	R/W	Tx 预分频寄存器	0x0000 0000
0x010	Tx_PPX	R/W	Tx 周期寄存器	0x0000 FFFF
0x014	Tx_DIER	R/W	Tx 中断使能控制寄存器	0x0000 0000
0x018	Tx_SR	R	Tx 中断状态寄存器	0x0000 0000
0x01C	Tx_SRIC	R/W	Tx 中断状态清除寄存器	0x0000 0000

T14 基地址：0x4000 1800

T15 基地址：0x4000 1880

11.3.1 Tx_CNT 定时器寄存器

表 11-2 Tx_CNT 定时器寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值																	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXCNT<15:0>																

TXCNT<15:0>：16 位的 Tx 定时器（写该寄存器的值只会在向上/向上向下计数方式下的启动时作为初始值载入，读该寄存器的值为模块定时/计数的当前值）

注：建议在定时器工作时不要对 CNT 寄存器进行写操作。关闭定时器使能后,当前计数值会被清零。

11.3.2 Tx_CTL1 Tx 控制寄存器 1

表 11-3 Tx_CTL1 Tx 控制寄存器 1

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																				
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0																				
R/W																							R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																							TXCMS<2:0>			TXDIR	TXCLK<1:0>			TXSY	TXCS	TXEN																				

TXCMS<2:0>：Tx 计数方式选择位。

000 = 向下计数，下溢时产生中断标志。

100 = 向上计数，上溢时产生中断标志。（初始化默认向上计数）

¹ 本章除特殊说明外，x=14,15

- x01 = 向上向下计数，上溢时产生中断标志
- x10 = 向上向下计数，下溢时产生中断标志
- x11 = 向上向下计数，上溢和下溢时都产生中断标志

TXDIR: Tx 计数方向标志位

- 0 = 当前计数方向为向下计数
- 1 = 当前计数方向为向上计数

TXCLK<1:0>: Tx 工作时钟选择位

- 00 = 选用 SCLK 时钟为 Tx 工作时钟
- 01 = 选用 HFCLK 时钟为 Tx 工作时钟
- 1x = 选用 LFCLK 时钟为 Tx 工作时钟

TXSY: Tx 计数模式外部触发脉冲输入同步控制位

- TXCS=0 时，该位被忽略，Tx 使用内部时钟
- TXCS=1 时，
 - 0 = 与外部触发脉冲输入同步
 - 1 = 不与外部触发脉冲输入同步

TXCS: Tx 定时/计数模式选择

- 0 = 定时模式，Tx 时钟由 TXCLK<1:0>决定。
- 1 = 计数模式：
 - TXSMS<2:0>= 111 计数模式 2，定时器时钟为触发输入信号 TRGI
 - TXSMS<2:0>≠ 111 计数模式 1，Tx 时钟为外部时钟 TxCK

TXEN: Tx 启动控制位

- 0 = 停止 Tx
- 1 = 启动 Tx

11.3.3 Tx_CTL2 Tx 控制寄存器 2

表 11-4 Tx_CTL2 Tx 控制寄存器 2

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXTRG	PXSPM	PXSPMST	TXUR	TXMSSYNC	TXTX<3:0>			TXSMS<2:0>			TXMMS<1:0>			TXUDEVT	TXUDEN		

TXTRG: 产生触发事件（该位由软件置‘1’，用于产生一个触发事件，定时器停止计数后由硬件自动清‘0’。）

- 0 = 无动作
- 1 = Tx_SR 寄存器中的 TXTIF=1，若使能相应的中断及 DMA,则产生相应的中断及 DMA 请求

PXSPM: 单脉冲输出模式

- 0 = 单脉冲输出模式未使能
- 1 = 使能单脉冲输出模式

PXSPMST: 单脉冲输出模式选择

- 0 = 单脉冲输出后不关闭定时器使能位
- 1 = 单脉冲输出后关闭定时器使能位

TXUR: 更新事件生成位

0 = 未使能

1 = 初始化定时器 Tx、Tx 预分频器的计数器并更新周期、占空比、预分频器、输出控制寄存器。

(该位硬件自动清零, 此位不受 TXUDEN 控制)

TXMSSYNC: 主从模式同步位

0 = 未使能

1 = 触发输入上的事件将被延迟, 以使主、从定时器同步响应外部触发

TXTS<3:0>: 触发输入 TRGI 选择位

0000 = T1

0001 = T2

0010 = T3

0011 = T4

0100 = T5

0101 = T9

0110 = T14

0111 = T15

1000 = T18

1001 = T19

1010 = T20

1011 = T21

1100 = TxCK

其余 = 保留

(这些位只在 TXSMS<2:0>= 000: 从模式禁止时才可改变。以避免电压转变时发生错误的边沿检测)

TXSMS<2:0>: 从模式选择位

000 = 从模式禁止

100 = 触发模式, 在触发信号 TRGI 出现上升沿时计数器启动(需使能 TXEN)

101 = 门控模式, 当触发信号 TRGI 为高时计数器的计数时钟使能, TRGI 为低时停止计数(计数器没有被复位)

110 = 复位模式, 在触发信号 TRGI 出现上升沿时初始化定时器, 并对寄存器进行更新

111 = 计数模式 2, 定时器时钟为触发信号 TRGI

其它 = 未使用(保留)

TXMMS<1:0>: 主模式选择位

00 = TXUR 位作为触发输出 TRGO

01 = 使能信号 TXEN 作为触发输出 TRGO

10 = TXIF 信号作为触发输出 TRGO

11 = 保留

TXUDEVT: 更新事件控制(在 TXUDEN=1 时有效)

0 = 当定时器 Tx 为 0(向下计数为 PPX 值时)更新以 Tx 为时基的周期、预分频寄存器到缓冲器中

1 = 产生更新事件, 更新以 Tx 为时基的周期、预分频寄存器到缓冲器中, 并将定时器复位(只在该位的上升沿有效对于定时器复位, 除了关断定时器,

其他时候都会保证当前计数值完整，在下一个计数周期复位)

TXUDEN: 更新使能

1 = 允许以 Tx 为时基的周期、预分频寄存器更新

0 = 禁止以 Tx 为时基的周期、预分频寄存器更新

11.3.4 Tx_PRSC Tx 预分频寄存器

表 11-5 Tx_PRSC Tx 预分频寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXCKS<15:0>																

TXCKS<15:0>: Tx 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，它是一个基于 16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一个计数周期开始时被采用。预分频器设置为 0 时，计数器 1:1 分频；预分频器设置为 1 时，计数器 1:2 分频，以此类推。预分频设置为 N，计时器时钟采用 N+1 分频。

11.3.5 Tx_PPX 周期寄存器

表 11-6 Tx_PPX 周期寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PPX<15:0>																

PPX<15:0>: Tx 定时器的周期寄存器。

11.3.6 Tx_DIER Tx 中断使能控制寄存器

表 11-7 Tx_DIER Tx 中断使能控制寄存器

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

TXTDE: 触发事件的 DMA 请求使能位

0 = 禁止触发事件的 DMA 请求

1 = 允许触发事件的 DMA 请求

TXUDE: 更新事件的 DMA 请求使能位

0 = 禁止更新事件的 DMA 请求

1 = 允许更新事件的 DMA 请求

TXIE: Tx 计数溢出中断使能位

0 = 禁止 Tx 计数溢出引起中断

1 = 允许 Tx 计数溢出引起中断

TXTIE: Tx 触发事件中断使能位

0 = 禁止 Tx 触发事件引起中断

1 = 允许 Tx 触发事件引起中断

TXUIE: Tx 更新事件中断使能位

0 = 禁止 Tx 更新事件引起中断

1 = 允许 Tx 更新事件引起中断

11.3.7 Tx_SR Tx 中断状态寄存器

表 11-8 Tx_SR Tx 中断状态寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																																	
位名																										TXUDF	TXTDF	TXIF	TXTIF	TXUIF			

TXTDF: Tx 触发事件触发 DMA 请求标志

0 = 无 Tx 触发事件触发 DMA

1 = Tx 触发事件触发 DMA

TXUDF: Tx 更新事件触发 DMA 请求标志

0 = 无 Tx 更新事件触发 DMA

1 = Tx 更新事件触发 DMA

TXIF: Tx 溢出中断标志

0 = 未发生 Tx 溢出中断

1 = 发生 Tx 溢出中断

TXTIF: Tx 触发事件中断标志

0 = 未发生 Tx 触发中断

1 = 发生 Tx 触发中断

TXUIF: Tx 更新事件中断标志

0 = 未发生 Tx 更新中断

1 = 发生 Tx 更新中断

注: TXTDF 和 TXTIF 位仅仅在 Tx_CTL2 寄存器 TXSMS<2:0>=100/101/110 时 (触发/门控/复位模式) 下有效。

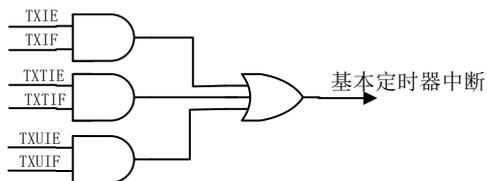


图 11-2 基本定时器中断框图

11.3.8 Tx_SRIC Tx 中断标志清除寄存器

表 11-9 Tx_DIER Tx 中断标志清除寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																										R/W	R/W	R/W				
位名																										TXIC	TXTIC	TXUIC				

TXIC: Tx 溢出中断标志清除位

0 = 不清除 Tx 溢出中断标志

1 = 清除 Tx 溢出中断标志

TXTIC: Tx 触发事件中标志清除位

0 = 不清除 Tx 触发中断标志

1 = 清除 Tx 触发中断标志

TXUIC: Tx 更新事件中标志清除位

0 = 不清除 Tx 更新中断标志

1 = 清除 Tx 更新中断标志

11.4 Tx 计数方式和中断

通过 Tx_CTL1 寄存器的 TXCMS<2:0>位可以选择 Tx 的三种计数方式: 向上计数方式, 向下计数方式和向上向下计数方式。注意, 基本定时器计数溢出标志 TXIF 的产生机制与通用定时器和高级定时器均不同。

向上计数方式下, 每个计数周期 Tx_CNT 寄存器自加 1, 加到 PPX 后再加一, Tx 溢出并将 Tx_CNT 清零, 同时 Tx 的溢出中断标志位 TXIF 置 1, 若使能了溢出中断 TXIE, 则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。将中断标志清除寄存器 Tx_SRIC 中的 TXIC 位置 1, 可以清零 TXIF, TXIC 位需要手动清零。

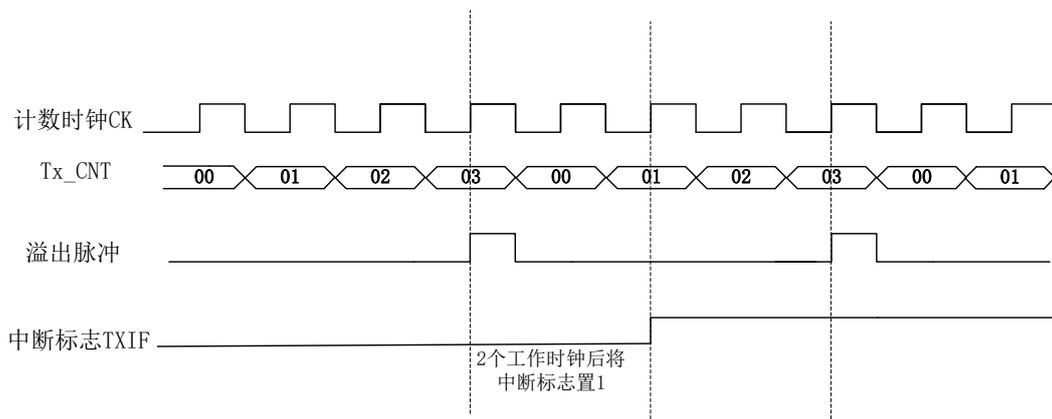


图 11-3 向上计数, 周期值 0x03

向下计数方式下, 每个计数周期 Tx_CNT 寄存器自减 1, 减到 0 后再减一, Tx 溢出并将周期值 PPX 重载入 Tx_CNT 寄存器, 同时 Tx 溢出中断标志位 TXIF 置 1, 若使能了溢出中断 TXIE, 则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。将中断标志清除寄存器 Tx_SRIC 中的 TXIC 位置 1, 可以清零 TXIF, TXIC 位需要手动清零。

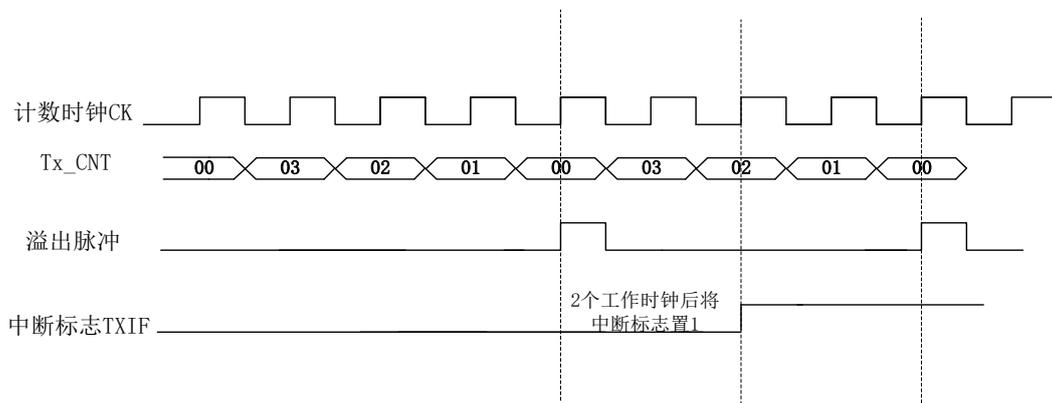


图 11-4 向下计数, 周期值 0x03

向上向下计数方式, 在 TXCMS<2:0>=x01 时, Tx 计数值达到 PPX 后再加一, Tx 溢出 (此时计数值更新为 PPX-1), 将 Tx 溢出中断标志位 TXIF 置 1, 此为上溢中断。在 TXCMS<2:0>=x10 时, Tx 寄存器减到 1 后再减一, Tx 溢出, 将 Tx 溢出中断标志位 TXIF 置 1, 此为下溢中断。在 TXCMS<2:0>=x11 时, 上溢和下溢均会触发 TXIF 溢出中断标志位。将中断标志清除寄存器 Tx_SRIC 中的 TXIC 位置 1, 可以清零 TXIF, TXIC 位需要手动清零。

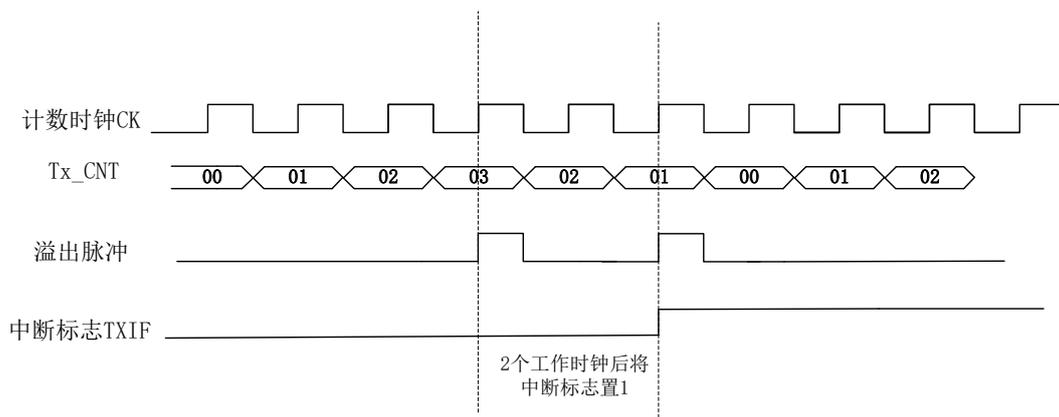


图 11-5 向上向下计数上下均溢出，周期值 0x03

11.5 Tx 工作模式

通过 Tx_CTL1 寄存器的 TXCS 位可以选择模块的工作模式，TXCS 为 0 时模块工作在定时模式，TXCS 为 1 时模块工作在计数模式。定时模式与计数模式的本质区别在于工作时使用的计数时钟源 Tsysclk 不同。具体可以参考本章节“基本定时器结构框图”。

11.5.1 定时模式

Tx_CTL1 寄存器的 TXCS 位为 0，则以 Tx_CTL1 寄存器的 TXCLK<1:0>位选择的工作时钟作为计数时钟源 Tsysclk。此时模块工作在定时模式。

11.5.2 计数模式

TXCS 为 1，则根据 Tx_CTL2 寄存器的 TXSMS<2:0>来选择计数时钟源 Tsysclk，TXSMS<2:0>≠111 时以 TxCK 引脚输入的下降沿进行计数（计数模式 1），TXSMS<2:0>=111 时以 TRGI 输入的下降沿进行计数（计数模式 2）。此时模块工作在计数模式。Tx 的计数模式又有同步计数和异步计数两种方式。

在计数模式下，如果 Tx_CTL1 寄存器的控制位 TXSY 清零，则 Tx 工作在异步计数模式，计数器根据 TxCK 引脚/TRGI 输入的脉冲下降沿进行计数；如果控制位 TXSY 置 1，则 Tx 工作在同步计数模式，用模块内部工作时钟对 TxCK 引脚/TRGI 输入电平进行采样，可以实现 TxCK/TRGI 与内部相位时钟的同步。

Tx 可选择在两个不同的外部计数信号下计数，这两个外部计数信号分别为外部计数引脚 TxCK 和触发输入 TRGI。

11.6 定时器外部触发功能

11.6.1 触发模式

如果使能触发模式(Tx_CTL2 寄存器 TXSMS<2:0>=100),且 Tx_CTL1 寄存器 TXEN=1,则在触发输入信号 TRGI 的上升沿 Tx 开始计数。同时, Tx_SR 寄存器中的 TXTIF 被置位,若使能了 Tx_DIER 寄存器中的触发中断使能位 TXTIE,则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求,见“Tx 触发 DMA(基本定时器)”。将中断标志清除寄存器 Tx_SRIC 中的 TXTIC 位置 1,可以清零 TXTIF, TXTIC 位需要手动清零。

11.6.2 门控模式

如果使能 Tx 门控功能 (Tx_CTL2 寄存器 TXSMS<2:0>=101), 且 Tx_CTL2 寄存器 TXEN=1, 则在触发输入信号 TRGI 为高电平时, 启动 Tx, 如果触发输入信号 TRGI 为低电平, 禁止 Tx (不复位)。TXEN=1 时, 在 TRGI 信号的上升沿以及下降沿, Tx_SR 寄存器中的 TXTIF 被置位, 若使能了 Tx_DIER 寄存器中的触发中断使能位 TXTIE, 则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求, 见“Tx 触发 DMA(基本定时器)”。将中断标志清除寄存器 Tx_SRIC 中的 TXTIC 位置 1, 可以清零 TXTIF, TXTIC 位需要手动清零。

使用该方式可粗略的对触发输入信号 TRGI 的高电平持续时间进行计算。

11.6.3 复位模式

如果使能复位模式(Tx_CTL2 寄存器 TXSMS<2:0>=110), 且 Tx_CTL2 寄存器 TXEN=1, 则在触发输入信号 TRGI 的上升沿复位 Tx 寄存器, 若 Tx_CTL2 寄存器中的更新使能位: TXUDEN=1, 则以 Tx 为时基的周期、预分频寄存器均被更新。同时, Tx_SR 寄存器中的 TXTIF 被置位, 若使能了 Tx_DIER 寄存器中的触发中断使能位 TXTIE, 则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求, 见“Tx 触发 DMA(基本定时器)”。将中断标志清除寄存器 Tx_SRIC 中的 TXTIC 位置 1, 可以清零 TXTIF, TXTIC 位需要手动清零。

11.7 定时器主从同步

11.7.1 定时器之间主从关系

任意一个基本定时器、通用定时器、高级定时器都可以作为触发源去触发其余的定时器, 提供触发源的定时器被称为主定时器, 被触发的定时器被称为从定时器。

此外, 也可以把 TxCK 引脚作为触发源, 用外部信号去触发从定时器。

11.7.2 使用一个定时器作为另一个定时器的预分频器 (计数模式 2)

可以用两个 16 位定时器组成一个 32 位定时器, 例如用定时器 T14 做为定时器 T15 的预分频器, 即 T14 每计数溢出一次, T15 计数一次, 具体配置如下:

- (1) T14_CTL2 寄存器 TXMMS<1:0>=10, T14 为主模式, T14 将在每次 T14IF 溢出中断标志信号产生时输出一个触发信号。
- (2) T15_CTL2 寄存器 TXTS<3:0>=0110, T15 的触发源为 T14, 选择定时器 14 触发。
- (3) T15_CTL2 寄存器 TXSMS<2:0>=111, 同时 T15_CTL1 寄存器的 TXCS 位置 1, 配置 T15 的从模式选择为计数模式 2。
- (4) 使能 T14_CTL1 的 TXEN 和 T15_CTL1 的 TXEN。

11.7.3 使用 TxCK 使能 Tx 定时器 (门控模式)

下例为用 T14CK 来使能 T14 的运行:

- (1) T14_CTL2 寄存器 TXTS<3:0>=1100, 配置 T14 的触发源为 T14CK, 选择 T14CK 触发。
- (2) T14_CTL2 寄存器 TXSMS<2:0>=101, 配置 T14 的从模式使 T14 处于门控模式下。
- (3) 配置 T14_CTL1 的 TXEN=1 使能定时器 T14。

11.7.4 使用一个定时器启动另一个定时器（触发模式）

下例为使用 T14 的溢出中断脉冲 T14IF 启动 T15 的配置过程：

- (1) T14_CTL2 寄存器 TXMMS<1:0>=10，T14 配置为主模式 T14_CTL1 的 TXIF 溢出中断脉冲信号作为触发输出 TRGO。
- (2) 配置 T14 的周期寄存器 T14_PPX。
- (3) T15_CTL2 寄存器 TXTS<3:0>=0110，T15 的触发源为 T14，选择定时器 14 触发。
- (4) T15_CTL2 寄存器 TXSMS<2:0>=100，T15 配置为触发模式。
- (5) 配置 T15_CTL1 的 TXEN=1 使能定时器 T15。
- (6) 配置 T14_CTL1 的 TXEN=1 使能定时器 T14。

11.7.5 使用一个外部触发同步的启动两个定时器(同步触发)

下例为 T14 在 T14CK 输入的上升沿启动，同时启动 T15 的配置：

- (1) T14_CTL2 寄存器 TXMMS<1:0>=01，配置 T14 的主模式，T14 使能信号 TXEN 作为触发输出 TRGO。
- (2) T14_CTL2 寄存器 TXTS<3:0>=1100，配置 T14 的触发输入源，T14CK 的输入。
- (3) T14_CTL2 寄存器 TXSMS<2:0>=100，配置 T14 的从模式，触发模式。
- (4) T14_CTL2 寄存器 TXMSSYNC=1，配置 T14 的主从模式同步位。
- (5) T15_CTL2 寄存器 TXTS<3:0>=0110，配置 T15 的触发输入源，定时器 14 触发。
- (6) T15_CTL2 寄存器 TXSMS<2:0>=100，配置 T15 的从模式，触发模式。
- (7) 配置 T15_CTL1 的 TXEN=1 使能定时器 T15。
- (8) 配置 T14_CTL1 的 TXEN=1 使能定时器 T14。

11.7.6 使用一个定时器复位另一个定时器（复位模式）

下例为使用 T14 的溢出中断脉冲 T14IF 复位 T15 的配置过程：

- (1) T14 配置为主模式，TXMMS<1:0>=10，T14_CTL1 的 TXIF 溢出中断脉冲信号作为触发输出 TRGO。
- (2) 配置 T14 的周期寄存器 T14_PPX。
- (3) T15 的触发源为 T14，T15_CTL2 的 TXTS<3:0>=0110，定时器 14 触发。
- (4) T15_CTL1 的 TXSMS<2:0>=110，T15 配置为复位模式。
- (5) 配置 T15_CTL1 的 TXEN=1 使能定时器 T15。
- (6) 配置 T14_CTL1 的 TXEN=1 使能定时器 T14。

11.8 Tx 更新

定时器开始计数时会产生启动更新事件。在运行过程中，有周期更新和立即更新。使能 Tx_CTL2 寄存器中的 TXUDEN 位，允许以 Tx 为时基的周期、预分频寄存器更新，这时当 TXUDEVT 位为 0，每个周期开始，更新以 Tx 为时基的周期、预分频寄存器到缓冲器中。当 TXUDEN 为 1 时，如果将 TXUDEVT 置 1，在 TXUDEVT 的上升沿立即更新以 Tx 为时基的周期、预分频寄存器到缓冲器中，并将定时器复位。

将 Tx_CTL2 寄存器中的 TXUR 位置 1（该位硬件自动清零，此位不受 TXUDEN 控制），立即初始化定时器 Tx 的计数器并更新周期、预分频寄存器。

发生更新事件时，会将 Tx_SR 寄存器中的更新中断标志位 TXUIF 置位，该位需要软件将 Tx_SRIC 寄存器中的 TXUIC 位置 1 才会清零。如果 Tx_DIER 寄存器中的更新中断使能位 TXUIE 位已经置位，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可

产生 DMA 请求，见“Tx 触发 DMA(基本定时器)”。

11.9 Tx 单脉冲模式

如果使能触发模式 (Tx_CTL2 寄存器 TXSMS<2:0>=100)，且 TXEN=1，这时使能寄存器 Tx_CTL2 中的 PXSPM 位，启动单脉冲模式，触发输入信号 TRGI 的上升沿 Tx 开始计数，一个周期计数完成后产生溢出中断并停止计数。如果 PXSPMST 位为零，则不清除定时器使能位 TXEN，在下次触发输入信号 TRGI 到来时又会重新开始计数。如果 PXSPMST 位置 1，则会清除定时器使能位 TXEN，在下次触发输入信号 TRGI 到来时不会计数，直到 TXEN 再次使能。

11.10 Tx 定时器关断

将寄存器 Tx_CTL1 中的 TXEN 位置 0，定时器关断，此时会将计数值以及预分频器清零，并且将所有缓存寄存器复位为初始值，再次开启时又会更新缓存寄存器。

11.11 Tx 触发 DMA(基本定时器)

Tx 在两种情况下可以触发 DMA，触发事件触发和更新事件触发。

触发事件触发 DMA，需要使能触发模式 (TXSMS<2:0>=100) (门控和复位模式也可以)，并将 Tx_DIER 寄存器的 TXTDE 置 1 使能触发事件 DMA 请求。当触发事件到来时，会将 Tx_SR 寄存器的 TXTDF 置 1 以向 DMA 发送请求。DMA 响应请求后会硬件清零标志位 TXTDF。触发事件也可以用软件触发 (TXTRG 置 1)。

更新事件触发 DMA，需要将 Tx_DIER 寄存器的 TXUDE 置 1 使能更新事件 DMA 请求，配置寄存器产生更新事件时，会将 Tx_SR 寄存器的 TXUDF 置 1 以向 DMA 发送请求。DMA 响应请求后会硬件清零标志位 TXUDF。值得注意的是，使能定时器 (TXEN 置 1) 也会产生一次更新事件。

12 通用定时/计数器(T0/T1/T2/T3/T4/T18/T19/T20/T21/T22/T23)

12.1 概述

Tx(x=0,1,2,3,4,18,19,22,23)是 16 位的定时/计数器, Tx(x=20,21)是 32 位的定时/计数器。它们除位宽不一样外, 其他功能以及实现方式都是一样的。其中 T0 可作为低功耗定时器使用。

通用定时/计数器有定时和计数 2 种工作模式, 支持 3 种计数方式: 向上计数、向下计数和向上向下计数方式。根据不同的模式, 计数会产生溢出, 将 Tx 中断标志位 TXIF 置 1。Tx 属于外部单元, 因此在使用 Tx 中断时, 需使能对应的外设中断。

通用定时/计数器主要功能包括:

- 16 位/32 位自动重载计数器
- 16 位/32 位可编程预分频器, 用于对输入的时钟按系数为 1~65536/1~4294967296 之间任意数值分频
- 基本定时器可以用于触发 AD 和 DA 模块
- 更新事件、触发事件 (触发模式、门控模式、复位模式)、捕捉事件、比较事件
- 可以产生 DMA 请求

12.2 通用定时器结构框图

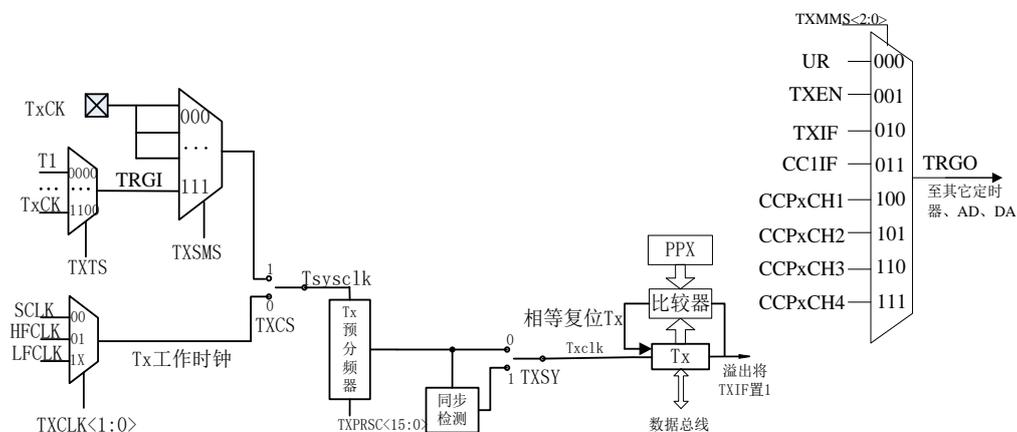


图 12-1 通用定时器结构框图¹

- TxCK 引脚: 外部计数引脚, 模块工作在计数模式 1 时, 从该引脚输入外部时钟信号。计数模式 2 时, 如果 Tx_CTL2 寄存器的 TXTS<2:0>=1100, 则 TxCK 引脚作为模块的触发输入 TRGI。该引脚从 IO 口通过重映射功能输入。
- Tsysclk: 计数时钟源, 用于输入分频器以产生计数时钟。
- Txclk: 计数时钟, 用于计数器计数。

¹ 上图的 CCPxCH1/2/3/4 为 CCP 相关引脚, 见通用捕捉/比较/PWM 模块章节。

12.3 通用定时器相关寄存器

表 12-1 通用定时器相关寄存器 1(x=0,1,2,3,4,18,19,22,23)

偏移地址	寄存器	访问	功能描述	复位值
0x000	Tx_CNT	R/W	Tx 寄存器	0x0000 uuuu
0x004	Tx_CTL1	R/W	Tx 控制寄存器 1	0x0000 0480
0x008	Tx_CTL2	R/W	Tx 控制寄存器 2	0x0000 0000
0x00C	Tx_PRSC	R/W	Tx 预分频寄存器	0x0000 0000
0x010	Tx_PPX	R/W	Tx 周期寄存器	0x0000 FFFF
0x014	Tx_UDTIM	R/W	更新计数器	0x0000 0000

T0 基地址: 0x4000 0880

T1 基地址: 0x4000 0100

T2 基地址: 0x4000 0180

T3 基地址: 0x4000 0200

T4 基地址: 0x4000 0280

T18 基地址: 0x4000 1A80

T19 基地址: 0x4000 1B00

T22 基地址: 0x4000 1C80

T23 基地址: 0x4000 1D00

表 12-2 通用定时器相关寄存器 2(x=20,21)

偏移地址	寄存器	访问	功能描述	复位值
0x0000	Tx_CNT	R/W	Tx 寄存器	0xuuuu uuuu
0x0004	Tx_CTL1	R/W	Tx 控制寄存器 1	0x0000 0480
0x0008	Tx_CTL2	R/W	Tx 控制寄存器 2	0x0000 0000
0x000C	Tx_PRSC	R/W	Tx 预分频寄存器	0x0000 0000
0x0010	Tx_PPX	R/W	Tx 周期寄存器	0xFFFF FFFF
0x0014	Tx_UDTIM	R/W	更新计数器	0x0000 0000

T20 基地址为: 0x4000 1B80

T21 基地址为: 0x4000 1C00¹

12.3.1 Tx_CNT 定时器寄存器

表 12-3 Tx_CNT Tx 定时器寄存器(x=0,1,2,3,4,18,19,22,23)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXCNT<15:0>																	

TXCNT<15:0>: 16 位的 Tx 定时器（写该寄存器的值只会在向上/向上向下计数方式下的启动时作为初始值载入，读该寄存器的值为定时器当前值）

¹ 通用定时器 T0/1/2/3/4/18/19/22/23 与 T20/21 除 Tx_CNT、Tx_PRSC、Tx_PPX 寄存器位数不一样外，其他都一致。

表 12-4 Tx_CNT Tx 定时器寄存器(x=20,21)

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	TXCNT<31:0>																																	

TXCNT<31:0>: 32 位的 Tx 定时器（写该寄存器的值只会在向上/向上向下计数方式下的启动时作为初始值载入，读该寄存器的值为定时器当前值）

注：建议在定时器工作时不要对 CNT 寄存器进行写操作。关闭定时器使能后,当前计数值会被保留。

12.3.2 Tx_CTL1 Tx 控制寄存器 1

表 12-5 Tx_CTL1 Tx 控制寄存器 1(x=0,1,2,3,4,18,19,20,21,22,23)

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R/W																				R	R/W		R/W	R/W	R/W	R	R/W	R/W		R/W	R/W	R/W	R/W
位名														TXIF	TXIE			TXCMS<2:0>			TXDIR	TXCLK<1:0>				TXSY	TXCS		TXEN				

TXIF: Tx 溢出中断标志位

- 0 = 没有发生溢出中断
- 1 = 产生溢出中断

TXIE: Tx 溢出中断使能位

- 0 = 禁止 Tx 溢出中断
- 1 = 使能 Tx 溢出中断

TXCMS<2:0>: Tx 计数方式选择位。

- 000 = 向下计数，下溢时产生中断标志。
- 100 = 向上计数，上溢时产生中断标志。（初始化默认向上计数）
- x01 = 向上向下计数，上溢时产生中断标志
- x10 = 向上向下计数，下溢时产生中断标志
- x11 = 向上向下计数，上溢和下溢时都产生中断标志

TXDIR: Tx 计数方向标志位

- 0 = 当前计数方向为向下计数
- 1 = 当前计数方向为向上计数

TXCLK<1:0>: Tx 工作时钟选择位

- 00 = 选用 SCLK 时钟为 Tx 工作时钟
- 01 = 选用 HFCLK 时钟为 Tx 工作时钟
- 1x = 选用 LFCLK 时钟为 Tx 工作时钟（非 T0）
- 10 = 选用 LFCLK 时钟为 T0 工作时钟（T0）

11 = 选用 INTLF 时钟为 T0 工作时钟 (T0)¹

TXSY: Tx 计数模式外部触发脉冲输入同步控制位

TXCS = 0 该位被忽略, Tx 使用内部时钟

TXCS = 1:

0 = 与外部触发脉冲输入同步

1 = 不与外部触发脉冲输入同步

TXCS: Tx 定时/计数模式选择

0 = 定时模式, Tx 时钟由 TXCLK<1:0>决定。

1 = 计数模式

SMS<2:0>= 111 计数模式 2, 定时器时钟为触发信号 TRGI

SMS<2:0>≠ 111 计数模式 1, Tx 时钟为外部时钟 TxCK

TXEN: Tx 启动控制位

0 = 停止 Tx

1 = 启动 Tx

12.3.3 Tx_CTL2 Tx 控制寄存器 2

表 12-6 Tx_CTL2 Tx 控制寄存器 2(x=0,1,2,3,4,18,19,20,21,22,23)

复位值	0	31	0	30	0	29	0	28	0	27	0	26	0	25	0	24	0	23	0	22	0	21	0	20	0	19	0	18	0	17	0	16	0	15	0	14	0	13	R/W	0	12	R/W	0	11	R/W	0	10	R/W	0	9	R/W	0	8	R/W	0	7	R/W	0	6	R/W	0	5	R/W	0	4	R/W	0	3	R/W	0	2	R/W	0	1	R/W	0	0												
位名																									TXUR	TXMSSYNC	TXTS<3:0>			TXSMS<2:0>		TXMMS<2:0>		TXUDEV	TXUDEN																																																						

TXUR: 更新事件生成位

0 = 未使能

1 = 初始化定时器 Tx 的计数器并更新周期、占空比、预分频器、输出控制寄存器。

(该位硬件自动清零, 此位不受 TXUDEN 控制)

TXMSSYNC: 主从模式同步位

0 = 未使能

1 = 触发输入上的事件将被延迟, 以使主、从定时器同步响应外部触发。

TXTS<3:0>: 触发输入 TRGI 选择位

0000 = T1 (T1 定时器中保留)

0001 = T2

0010 = T3

0011 = T4

0100 = T5

0101 = T9

0110 = T14

0111 = T15

1000 = T18

1001 = T19

¹ T0 是低功耗定时器, T0CLK<1:0>=10 时需要使能 PM_CTL2 寄存器的 CCPCLKLPEN 位。

1010 = T20

1011 = T21

1100 = TxCK

1101 = CCPxCH1 的输入（在串口测量模式下为 USART0_RX 信号）

1110 = CCPxCH2 的输入

1111 = CCPxCH3 的输入

（这些位只在 TXSMS=000：从模式禁止时才可改变。以避免电压转变时发生错误的边沿检测。）

TXSMS<2:0>：从模式选择位

000 = 从模式禁止

100 = 触发模式，在触发信号 TRGI 出现上升沿时计数器启动（需使能 TXEN）

101 = 门控模式，当触发信号 TRGI 为高时计数器的计数时钟使能，TRGI 为低时停止计数（计数器没有被复位）

110 = 复位模式，在触发信号 TRGI 出现上升沿时初始化定时器，并对寄存器进行更新。对于外部引脚信号需要先用计数时钟滤波，所以外部引脚信号脉宽需要大于两个计数时钟。

111 = 计数模式 2，定时器时钟为触发信号 TRGI

其它 = 未使用（保留）

TXMMS<2:0>：主模式选择位

000 = TXUR 位作为触发输出 TRGO

001 = 使能信号 TXEN 作为触发输出 TRGO

010 = TXIF 信号作为触发输出 TRGO（定时器溢出时的中断脉冲信号）

011 = CC1IF 脉冲作为触发输出 TRGO

100 = CCPxCH1 作为触发输出 TRGO

101 = CCPxCH2 作为触发输出 TRGO

110 = CCPxCH3 作为触发输出 TRGO

111 = CCPxCH4 作为触发输出 TRGO

TXUDEVT：更新事件控制位（在 TXUDEN=1 时有效）

0 = 当定时器 Tx 为 0（向上计数时）或为 PPX 值（向下计数）时且 Tx_UDTIMX 计数到设定值时更新以 Tx 为时基的占空比、周期、输出控制、预分频、周期计数寄存器到缓冲器中

1 = 产生更新事件，更新以 Tx 为时基的占空比、周期、输出控制、预分频、周期计数寄存器到缓冲器中，并将定时器复位（只在该位的上升沿有效。对于定时器复位，除了关断定时器，其他时候都会保证当前计数值完整，在下一个计数周期复位）

TXUDEN：更新使能位

0 = 禁止以 Tx 为时基占空比、周期、输出控制、预分频、周期计数寄存器更新

1 = 允许以 Tx 为时基占空比、周期、输出控制、预分频、周期计数寄存器更新

12.3.4 Tx_PRSC Tx 预分频器
表 12-7 Tx_PRSC Tx 预分频器(x=0,1,2,3,4,18,19,22,23)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXCKS<15:0>																

TXCKS<15:0>: Tx 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，它是一个基于 16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一个计数周期开始时被采用。预分频器设置为 0 时，计数器 1:1 分频；预分频器设置为 1 时，计数器 1:2 分频，以此类推。预分频设置为 N，计时器时钟采用 N+1 分频。

表 12-8 Tx_PRSC Tx 预分频器(x=20,21)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	TXCKS<31:0>																															

TXCKS<31:0>: Tx 预分频器可以将计数器的时钟频率按 1 到 4294967296 之间的任意值分频，它是一个基于 32 位寄存器控制的 32 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一个计数周期开始时被采用。预分频器设置为 0 时，计数器 1:1 分频；预分频器设置为 1 时，计数器 1:2 分频，以此类推。预分频设置为 N，计时器时钟采用 N+1 分频。

12.3.5 Tx_PPX 周期寄存器
表 12-9 Tx_PPX 周期寄存器(x=0,1,2,3,4,18,19,22,23)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PPX<15:0>															

PPX<15:0>: 16 位 Tx 定时器的周期寄存器

表 12-10 Tx_PPX 周期寄存器(x=20,21)

复位值	1 31	1 30	1 29	1 28	1 27	1 26	1 25	1 24	1 23	1 22	1 21	1 20	1 19	1 18	1 17	1 16	1 15	1 14	1 13	1 12	1 11	1 10	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PPX<31:0>																															

PPX<31:0>: 32 位 Tx 定时器的周期寄存器

12.3.6 Tx_UDTIM 更新计数器

表 12-11 Tx_UDTIM 更新计数器(x=0,1,2,3,4,18,19,20,21,22,23)

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
R/W																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																								TXUD<7:0>																													

TXUD<7:0>: 当使能Tx的更新功能时,若Tx_CTL2寄存器中的TXUDEN=1, TXUDEVT=0, 则只有在Tx_UDTIM的值向下计数到0时才会更新Tx为时基的预分频器、周期、占空比、输出控制、输出配置寄存器。

12.4 Tx 计数方式和中断

通过 Tx_CTL1 寄存器的 TXCMS<2:0>位可以选择 Tx 的三种计数方式：向上计数方式，向下计数方式和向上向下计数方式。注意，通用定时器计数溢出标志 TXIF 的产生机制与基本定时器和高级定时器均不同。

向上计数方式下，每个计数周期 Tx 寄存器自加 1，加到 PPX 后再加一，Tx 溢出（此时计数值更新为 0），将 Tx 的溢出中断标志位 TXIF 置 1，若使能了溢出中断 TXIE，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。将中断标志清除寄存器 Tx_SRIC 中的 TXIC 位置 1，可以清零 TXIF，TXIC 位需要手动清零。

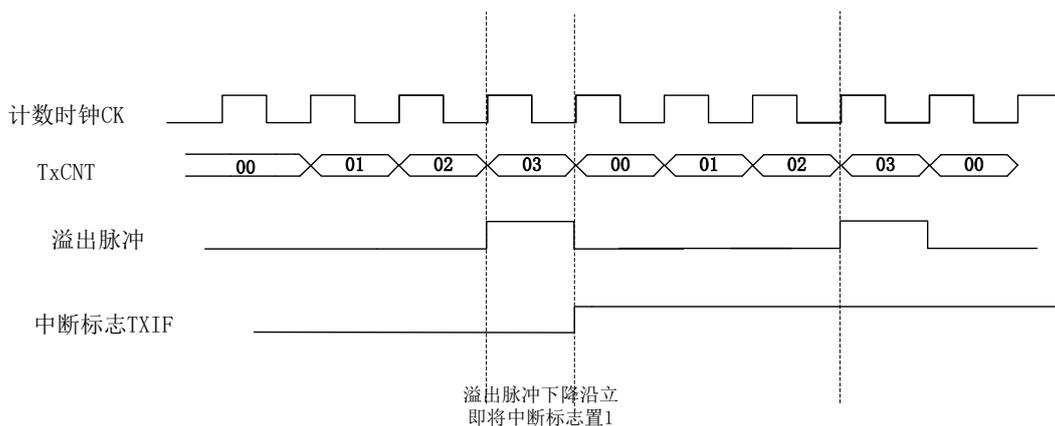


图 12-2 向上计数，周期值 0x03

向下计数方式下，每个计数周期 Tx 寄存器自减 1，减到 0 后再减一，Tx 溢出（此时计数值更新为 PPX），将 Tx 溢出中断标志位 TXIF 置 1，若使能了溢出中断 TXIE，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。将中断标志清除寄存器 Tx_SRIC 中的 TXIC 位置 1，可以清零 TXIF，TXIC 位需要手动清零。

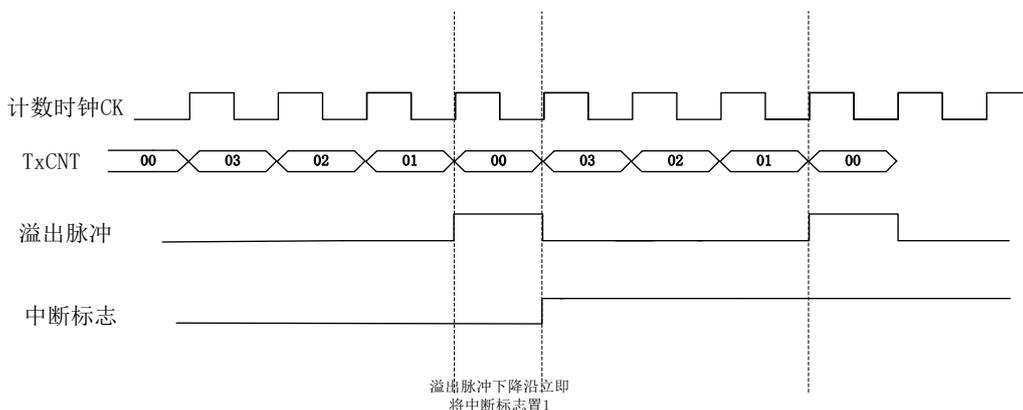


图 12-3 向下计数，周期值 0x03

向上向下计数方式，在 TXCMS<2:0>=x01 时，Tx 计数值达到 PPX 后再加一，Tx 溢出（此时计数值更新为 PPX-1），将 Tx 溢出中断标志位 TXIF 置 1，此为上溢中断。在 TXCMS<2:0>=x10 时，Tx 寄存器减到 1 后再减一，Tx 溢出，将 Tx 溢出中断标志位 TXIF 置 1，此为下溢中断。在 TXCMS<2:0>=x11 时，上溢和下溢均会触发 TXIF 溢出中断标志位。将中断标志清除寄存器 Tx_SRIC 中的 TXIC 位置 1，可以清零 TXIF，TXIC 位需要手动清零。

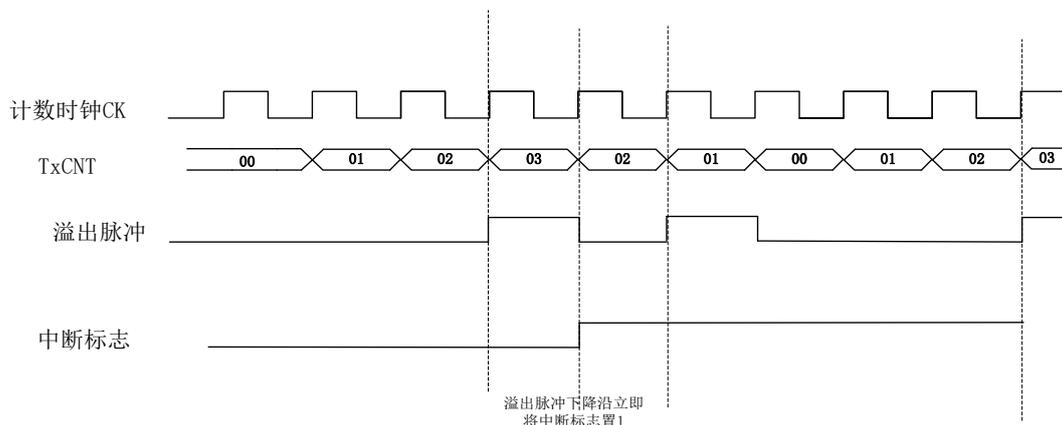


图 12-4 向上向下计数上下均溢出，周期值 0x03

12.5 Tx 工作模式

通过 Tx_CTL1 寄存器的 TXCS 位可以选择模块的工作模式，TXCS 为 0 时模块工作在定时模式，TXCS 为 1 时模块工作在计数模式。定时模式与计数模式的本质区别在于工作时使用的计数时钟源 Tsysclk 不同。具体可以参考本章节“通用定时器结构框图”。

12.5.1 定时模式

Tx_CTL1 寄存器的 TXCS 位为 0，则以 Tx_CTL1 寄存器的 TXCLK<1:0>位选择的工作时钟作为计数时钟源 Tsysclk。此时模块工作在定时模式。

12.5.2 计数模式

TXCS 为 1，则根据 Tx_CTL2 寄存器的 TXSMS<2:0>来选择计数时钟源 Tsysclk，TXSMS<2:0>≠111 时以 TxCK 引脚输入的上升沿进行计数（计数模式 1），TXSMS<2:0>=111 时以 TRGI 输入的上升沿进行计数（计数模式 2）。

Tx 的计数模式又有同步计数和异步计数两种方式。

在计数模式下，如果 Tx_CTL1 寄存器的控制位 TXSY 清零，则 Tx 工作在异步计数模式，计数器根据 TxCK 引脚/TRGI 输入的脉冲上升沿进行计数；如果控制位 TXSY 置 1，则 Tx 工作在同步计数模式，用模块内部工作时钟对 TxCK 引脚/TRGI 输入电平进行采样，可以实现 TxCK/TRGI 与内部相位时钟的同步。

Tx 可选择在两个不同的外部计数信号下计数，这两个外部计数信号分别为外部计数引脚 TxCK 和触发输入 TRGI。

12.6 定时器外部触发功能

12.6.1 触发模式

如果使能触发模式(Tx_CTL2 寄存器 TXSMS<2:0>=100)，且 Tx_CTL1 寄存器 TXEN=1，则在触发输入信号 TRGI 的上升沿 Tx 开始计数。同时，CCPx_EGIF 寄存器中的 TXTIF 被置位，若使能了 CCPx_CTL3 寄存器中的触发中断使能位 TXTIE，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求，见“Tx 触发 DMA(通用定时器)”。将中断标志清除寄存器 CCPx_SRIC 中的 TXTIC 位置 1，可以清零 TXTIF，TXTIC 位需要手动清零。

12.6.2 门控模式

如果使能 Tx 门控功能 (Tx_CTL2 寄存器 TXSMS<2:0>=101), 且 Tx_CTL1 寄存器 TXEN=1, 则在触发输入信号 TRGI 为高电平时, 启动 Tx, 如果触发输入信号 TRGI 为低电平, 禁止 Tx。TXEN=1 时, 在 TRGI 信号的上升沿以及下降沿, CCPx_EGIF 寄存器中的 TXTIF 被置位, 若使能了 CCPx_CTL3 寄存器中的触发中断使能位 TXTIE, 则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求, 见“Tx 触发 DMA(通用定时器)”。将中断标志清除寄存器 CCPx_SRIC 中的 TXTIC 位置 1, 可以清零 TXTIF, TXTIC 位需要手动清零。

使用该方式可粗略的对触发输入信号 TRGI 的高电平持续时间进行计算。

12.6.3 复位模式

如果使能复位模式(Tx_CTL2 寄存器 TXSMS<2:0>=110), 且 Tx_CTL1 寄存器 TXEN=1, 则在触发输入信号 TRGI 的上升沿复位 Tx 寄存器, 若 Tx_CTL2 寄存器中的更新使能位: TXUDEN=1, 则以 Tx 为时基的占空比、周期、预分频器、输出控制、输出配置寄存器均被更新。同时, CCPx_EGIF 寄存器中的 TXTIF 被置位, 若使能了 CCPx_CTL3 寄存器中的触发中断使能位 TXTIE, 则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求, 见“Tx 触发 DMA(通用定时器)”。将中断标志清除寄存器 CCPx_SRIC 中的 TXTIC 位置 1, 可以清零 TXTIF, TXTIC 位需要手动清零。

这里需要注意的是, 若在复位模式下, 触发输入为外部引脚信号, 则需要进过两级的计数时钟下降沿滤波, 所以需要引脚信号至少有两个计数时钟脉宽。

12.7 定时器主从同步

12.7.1 定时器之间主从关系

任意一个基本定时器、通用定时器、高级定时器都可以作为触发源去触发其余的定时器, 提供触发源的定时器被称为主定时器, 被触发的定时器被称为从定时器。

此外, 也可以把 TxCK 引脚作为触发源, 用外部信号去触发从定时器。

12.7.2 使用一个定时器作为另一个定时器的预分频器 (计数模式 2)

如基本定时器章节中主从关系, 可以用两个 16 位定时器组成一个 32 位定时器, 例如用定时器 T1 做为定时器 T3 的预分频器, 即 T1 计数溢出一次, T3 计数一次, 具体配置如下:

- (1) T1_CTL2 寄存器的 TXMMS<2:0>=010, 配置 T1 为主模式, T1 将在每次 T1 溢出中断标志 TXIF 产生时输出一个触发信号。
- (2) T3_CTL2 寄存器的 TXTS<3:0>=0000, T3 的触发源配置为 T1, 选择定时器 1 触发。
- (3) T3_CTL2 寄存器的 TXSMS<2:0>=111, 同时 T3_CTL1 寄存器的 TXCS 位置 1, 配置 T3 的从模式选择为计数模式 2。
- (4) 配置 T3_CTL1 的 TXEN=1 使能定时器 T3。
- (5) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

12.7.3 使用一个定时器使能另一个定时器 (门控模式)

下例为用 CCP1 的输出使能 T3 的运行:

- (1) T1_CTL2 寄存器的 TXMMS<2:0>=100, T1 配置为主模式, CCP1CH1 作为触发输出 TRGO。

- (2) T3_CTL2 寄存器的 TXTS<3:0>=0000，配置 T3 从 T1 获得触发源，选择定时器 1 触发。
- (3) T3_CTL2 寄存器的 TXSMS<2:0>=101，配置 T3 的从模式使 T3 处于门控模式下。
- (4) 配置 T3_CTL1 的 TXEN=1 使能定时器 T3。
- (5) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

12.7.4 使用一个定时器启动另一个定时器（触发模式）

下例为使用 T1 的中断标志位 T1IF 启动 T3 的配置过程：

- (1) T1_CTL2 寄存器的 T1MMS<2:0>=010，T1 配置为主模式，T1 溢出中断脉冲信号 TXIF 作为触发输出 TRGO。
- (2) 配置 T1 的周期寄存器 T1_PPX。
- (3) T3_CTL2 寄存器的 TXTS<3:0>=0000，T3 的触发源为 T1，选择定时器 1 触发。
- (4) T3_CTL2 寄存器的 TXSMS<2:0>=100，T3 配置为触发模式。
- (5) 配置 T3_CTL1 的 TXEN=1 使能定时器 T3。
- (6) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

12.7.5 使用一个外部触发同步的启动两个定时器（同步触发）

下例为 T1 在 CCP1CH1 输入的上升沿启动，同时启动 T3 的配置：

- (1) T1_CTL2 寄存器的 TXMMS<2:0>=001，配置 T1 的主模式，T1 使能信号 TXEN 作为触发输出 TRGO。
- (2) T1_CTL2 寄存器的 TXTS<3:0>=1101，配置 T1 的触发输入源，选择 CCP1CH1 的输入。
- (3) T1_CTL2 寄存器的 TXSMS<2:0>=100，配置 T1 的从模式，触发模式。
- (4) T3_CTL2 寄存器的 TXMSSYNC=1，配置 T3 的主从模式同步位。
- (5) T3_CTL2 寄存器的配置 T3 的触发输入源，TXTS<3:0>=0000，定时器 1 触发。
- (6) T3_CTL2 寄存器的 TXSMS<2:0>=100，配置 T3 的从模式，触发模式。
- (7) 配置 T3_CTL1 的 TXEN=1 使能定时器 T3。
- (8) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

12.7.6 使用一个定时器复位另一个定时器（复位模式）

下例为使用 T1 的中断标志位 T1IF 复位 T3 的配置过程：

- (1) T1_CTL2 寄存器的 T1MMS<2:0>=010，T1 配置为主模式，T1 溢出中断脉冲信号 TXIF 作为触发输出 TRGO。
- (2) 配置 T1 的周期寄存器 T1_PPX。
- (3) T3_CTL2 寄存器的 TXTS<3:0>=0000，T3 的触发源为 T1，选择定时器 1 触发。
- (4) T3_CTL2 寄存器的 TXSMS<2:0>=110，T3 配置为复位模式。
- (5) 配置 T3_CTL1 的 TXEN=1 使能定时器 T3。
- (6) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

12.8 Tx 更新

定时器开始计数时会产生启动更新事件。在运行过程中，有周期更新和立即更新。使能 Tx_CTL2 寄存器中的 TXUDEN 位，允许以 Tx 为时基的占空比、周期、输出控制、预分频、周期计数寄存器更新，这时当 TXUDEVT 位为 0，每一个计数周期开始，Tx_UDTIM 递减计

数 1，计数到 0 时更新以 Tx 为时基的占空比、周期、输出控制、预分频、周期计数寄存器到缓冲器中。当 TXUDEN 为 1 时，如果将 TXUDEVT 置 1，在 TXUDEVT 的上升沿会立即更新以 Tx 为时基的占空比、周期、输出控制、预分频、周期计数寄存器到缓冲器中，并将定时器复位。

将 Tx_CTL2 寄存器中的 TXUR 位置 1(该位硬件自动清零,此位不受 TXUDEN 控制)，立即初始化定时器 Tx 的计数器并更新周期、占空比、预分频器、输出控制、周期计数寄存器。

发生更新事件时，会将 CCPx_EGIF 寄存器中的更新中断标志位 TXUIF 置位，该位需要软件将 CCPx_SRIC 寄存器中的 TXUIC 位置 1 才会清零。如果 CCPx_CTL3 寄存器中的更新中断使能位 TXUIE 位已经置位，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求，见“Tx 触发 DMA(通用定时器)”。

向上计数启动更新和周期更新

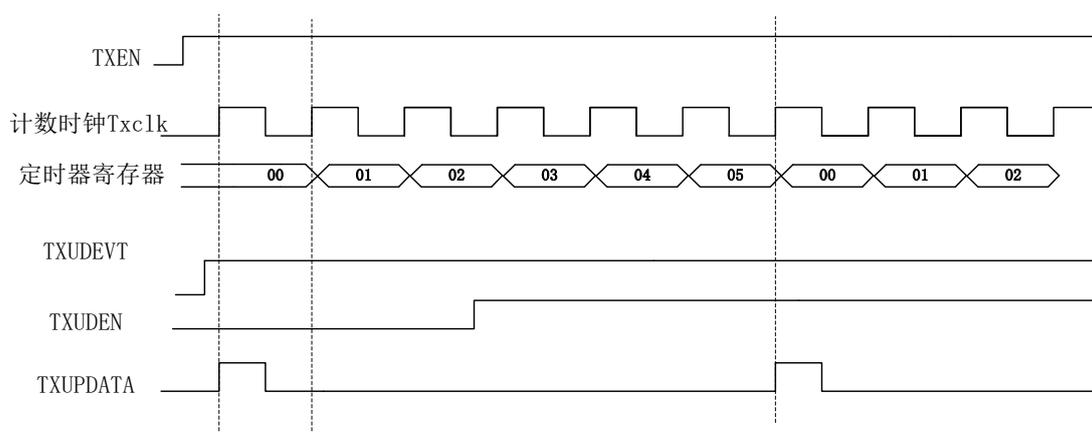


图 12-5 通用定时器向上计数启动更新和周期更新¹

¹ 周期为 0x05，TxCLK 为计数时钟。TXUPDATA 为占空比、周期、输出控制、预分频器寄存器更新时钟。

向下计数周期更新

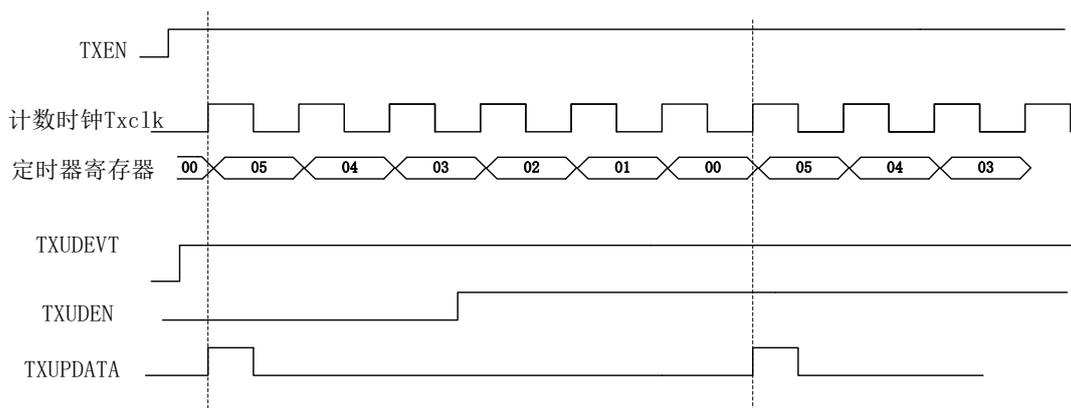


图 12-6 通用定时器向下计数周期更新

向上计数立即更新

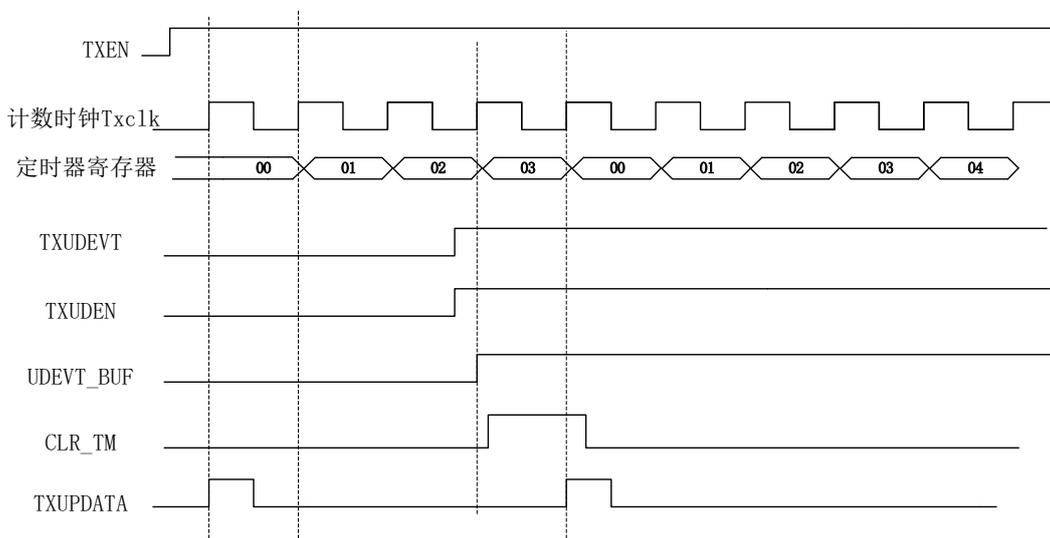
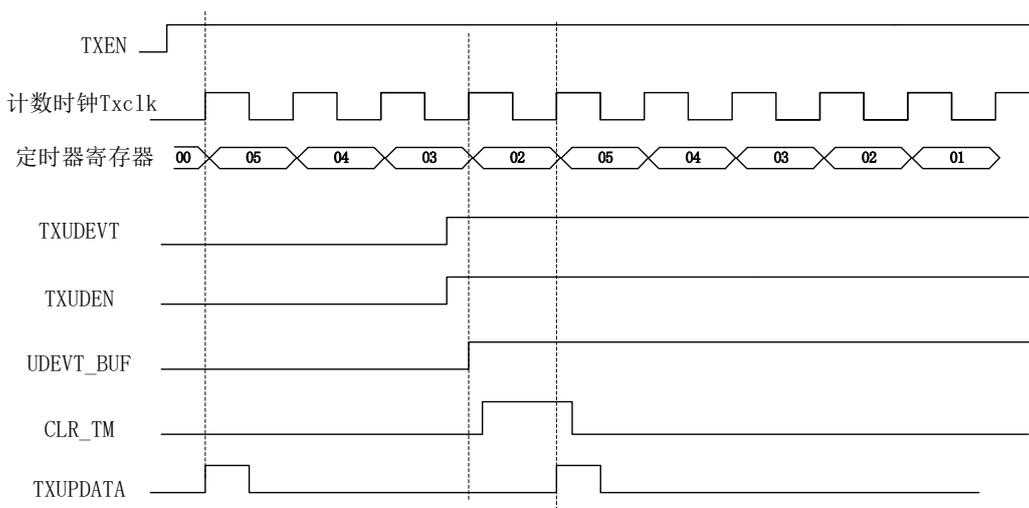


图 12-7 通用定时器向上计数立即更新

向下计数立即更新

图 12-8 通用定时器向下计数立即更新

注：向上向下计数更新时机与向上计数一致。

12.9 Tx 定时器关断

如果将 Tx_CTL1 寄存器的 TXEN 位置 0，定时器将关断。定时器关断时，会将计数值保留，并且将所有缓存寄存器复位为初始值，再次开启时又会更新缓存寄存器。

12.10 Tx 分配给 CCPx

CCPx 模块在使用时，需要使用 Tx 作为时基，其中 T1 分配给 CCP1，T2 分配给 CCP2，依此类推，详细请参照通用捕捉/比较/PWM 模块部分。

12.11 T0 低功耗定时器（LP_TIMER）

定时器 T0 可作为低功耗定时器（LP_TIMER）使用，此时 CCP0 也可以工作于低功耗模式。低功耗的定时器可以在除待机和关断模式以外的模式中工作。当芯片处于停止模式下，内核已经掉电，此时低功耗定时器仍可以保持工作状态。

为了使能定时器进入低功耗模式，需要使能 PM_CTL2 寄存器的 CCP0LPEN 位不让 T0 复位，同时 T0_CTL1 的 TXCLK<1:0>=11 即选用内部低频振荡器 INTLF 作为相应的时钟源并将 PM_CTL2 寄存器的 CCP0CLKLPEN 置 1 允许内部低频振荡器作为 CCP 工作时钟，同时保持时钟源处于工作状态，这样低功耗定时器就可以在低功耗模式下运行。

低功耗模式下的 IO 口请参考数据手册的引脚重映射章节“引脚重映射表-低功耗外设功能重映射”。

注：PM_CTL 寄存器位于备份域，需要将 OSC_CTL0 寄存器的 PMWREN 位置 1 以允许备份域寄存器读写。

12.12 Tx 触发 DMA(通用定时器)

Tx 在 6 种情况下可以触发 DMA，触发事件触发、更新事件触发、CCPxCH1/2/3/4 通道上的捕捉/比较事件触发。

触发事件触发 DMA，需要使能触发模式(TXSMS<2:0>=100)(门控和复位模式也可以)，并将 CCPx_CTL3 寄存器的 TXTDE 置 1 使能触发事件 DMA 请求。当触发事件到来时，会将 CCPx_DF 寄存器的 TXTDF 置 1 以向 DMA 发送请求，DMA 响应请求后会硬件清零标志位 TXTDF。触发事件也可以用软件触发 (TXTRG 置 1)。

更新事件触发 DMA，需要将 CCPx_CTL3 寄存器的 TXUDE 置 1 使能更新事件 DMA 请求，配置寄存器产生更新事件时，会将 CCPx_DF 寄存器的 TXUDF 置 1 以向 DMA 发送请求。DMA 响应请求后会硬件清零标志位 TXUDF。值得注意的是，使能定时器 (TXEN 置 1) 也会产生一次更新事件。

CCPxCH1/2/3/4 通道上的捕捉/比较事件触发 DMA，需要使能 CCPx_CTL3 寄存器相应的 CCyDE，允许 CCPxCHy 通道的捕捉/比较触发 DMA 请求。等待相应捕捉/比较事件发生时，会将 CCPx_DF 寄存器的 CCyDF 置 1 以向 DMA 发送请求。DMA 响应请求后会硬件清零标志位 CCyDF。

13 高级定时/计数器 (T5/T6/T9/T10)

13.1 概述

ECCPx 模块各包含两个计数器 Tx/Tz (x=5,9;z=6,10;Tx 和 Tz 原理相同), 他们是 16 位的定时器, 有 3 种计数方式: 向上计数、向下计数和向上向下计数方式, 可精确配置 1-65535 自由分频进行计数。支持触发其它定时器、AD 及 DMA 等外设。

高级定时/计数器主要功能包括:

- 16 位位自动重载计数器
- 16 位的可编程预分频器 (分频器 1) 和 4 位的可编程后分频器 (分频器 2)
- 高级定时器可用于触发 AD、DA 等模块
- 支持周期更新和立即更新
- 支持比较器清零定时器功能
- 支持主从模式 (触发、门控、复位)
- 可以用来产生 DMA 请求 (更新、TRGI 触发、捕捉/比较、关断事件)

13.2 高级定时器结构框图

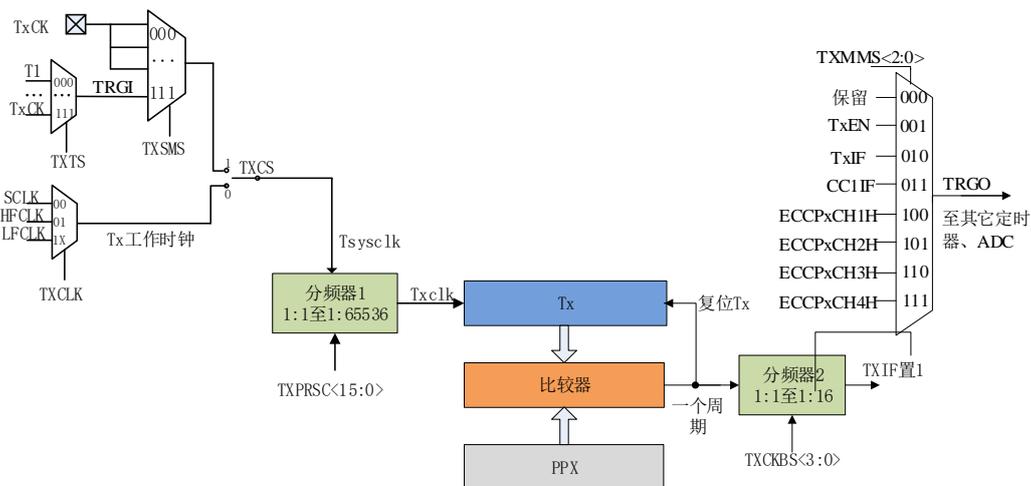


图 13-1 高级定时器结构框图

- TxCK 引脚: 外部计数引脚, 模块工作在计数模式 1 时, 从该引脚输入外部时钟信号。计数模式 2 时, 如果 Tx_CTL2 寄存器的 TXTS<2:0>=1100, 则 TxCK 引脚作为模块的触发输入 TRGI。该引脚从 IO 口通过重映射功能输入。
- Txclk: 计数时钟, 用于计数器计数
- Tsysclk: 计数时钟源, 用于输入分频器以产生计数时钟。

注: 上图右侧的 ECCPxCH1/2/3/4H 为 ECCP 相关引脚, 具体功能参考 ECCP 章节。

13.3 高级定时器相关寄存器

 表 13-1 高级定时器相关寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x000	Tx_CNT	R/W	Tx_CNT 寄存器	0x0000 uuuu
0x004	Tz_CNT	R/W	Tz_CNT 寄存器	0x0000 uuuu
0x008	Tx_PPX	R/W	Tx_PPX 周期寄存器	0x0000 FFFF
0x00C	Tz_PPZ	R/W	Tz_PPZ 周期寄存器	0x0000 FFFF
0x010	Tx_PRSC	R/W	Tx 预分频设置寄存器	0x0000 0000
0x014	Tz_PRSC	R/W	Tz 预分频设置寄存器	0x0000 0000
0x018	Tx_CCR0	R/W	Tx 启动 ADC 设置寄存器 0	0x0000 0000
0x01C	Tx_CCR1	R/W	Tx 启动 ADC 设置寄存器 1	0x0000 0000
0x020	Tz_CCR0	R/W	Tz 启动 ADC 设置寄存器 0	0x0000 0000
0x024	Tx_CTL	R/W	Tx 控制寄存器	0x0000 8080
0x028	Tz_CTL	R/W	Tz 控制寄存器	0x0000 8080
0x2C	ECCPx_PDCTL	R/W	ECCP5 位置检测控制寄存器	0x0000 0000
0x30	ECCPx_PXASCTL	R/W	关断控制寄存器	0x0000 0000
0x34	Tx_CCTCTL	R/W	比较器清零定时器 Tx 控制寄存器	0x0000 0000
0x38	Tz_CCTCTL	R/W	比较器清零定时器 Tz 控制寄存器	0x0000 0000
0x5C	Tx_CCRCTL	R/W	CCR 功能控制寄存器	0x0000 0000

T5/T6 基地址：0x4000 0300

T9/T10 基地址：0x4000 0400

13.3.1 Tx_CNT 定时器寄存器

表 13-2 Tx_CNT 定时器寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXCNT<15:0>																	

TXCNT<15:0>：16 位的 Tx 定时器（写该寄存器的值只会在向上/向上向下计数方式下的启动时作为初始值载入，读该寄存器的值为定时器当前值）

注：建议在定时器工作时不要对 CNT 寄存器进行写操作。关闭定时器使能后,当前计数值会被保留。

¹ 本章除特殊说明，其中 x=5,9, z=6,10, y=1,2,3,4。

13.3.2 Tz_CNT 定时器寄存器

表 13-3 Tz_CNT 定时器寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
R/W	R/W																R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TZCNT<15:0>																		

TZCNT<15:0>: 16 位的 Tz 定时器（写该寄存器的值只会在向上/向上向下计数方式下的启动时作为初始值载入，读该寄存器的值为定时器当前值）

注：建议在定时器工作时不要对 CNT 寄存器进行写操作。关闭定时器使能后,当前计数值会被保留。

13.3.3 Tx_PPX 周期寄存器

表 13-4 Tx_PPX 周期寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
R/W	R/W																R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PPX<15:0>																		

PPX<15:0>: Tx 定时器的周期寄存器

13.3.4 Tz_PPZ 周期寄存器

表 13-5 Tz_PPZ 周期寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
R/W	R/W																R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PPZ<15:0>																		

PPZ<15:0>: Tz 定时器的周期寄存器

13.3.5 Tx_PRSC/Tz_PRSC Tx/Tz 预分频寄存器
表 13-6 Tx_PRSC Tx 预分频寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXCKS<15:0>																															

表 13-7 Tz_PRSC Tz 预分频寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0															
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TZCKS<15:0>																														

TXCKS<15:0>/TZCKS<15:0>: Tx/Tz 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，它是一个基于 16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一个计数周期开始时被采用。预分频器设置为 0 时，计数器 1:1 分频；预分频器设置为 1 时，计数器 1:2 分频，预分频器设置为 N 时，计数器 1:N+1 分频以此类推。预分频计数器又称为分频器 1，与后分频器（分频器 2）对应。

13.3.6 Tx_CCR0/Tx_CCR1/Tz_CCR0 Tx/Tz 触发 AD 寄存器
表 13-8 Tx_CCR0 Tx 触发 AD 寄存器 0

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0															
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXCCR0<15:0>																														

表 13-9 Tx_CCR1 Tx 触发 AD 寄存器 1

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0															
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TXCCR1<15:0>																														

TXCKBS<3:0>: Tx 后分频器（分频器 2）分频比选择位

- 0000 = 分频比为 1:1
- 0001 = 分频比为 1:2
- 0010 = 分频比为 1:3
- 0011 = 分频比为 1:4
- 0100 = 分频比为 1:5
- 0101 = 分频比为 1:6
- 0110 = 分频比为 1:7
- 0111 = 分频比为 1:8
- 1000 = 分频比为 1:9
- 1001 = 分频比为 1:10
- 1010 = 分频比为 1:11
- 1011 = 分频比为 1:12
- 1100 = 分频比为 1:13
- 1101 = 分频比为 1:14
- 1110 = 分频比为 1:15
- 1111 = 分频比为 1:16

TXEN: Tx 使能位

- 0 = 禁止 Tx
- 1 = 使能 Tx

TXSY: Tx 计数模式外部触发脉冲输入同步控制位

- TXCS= 0: 该位被忽略, Tx 使用内部时钟
- TXCS= 1:
 - 0 = 与外部触发脉冲输入同步
 - 1 = 不与外部触发脉冲输入同步

TXCS: Tx 工作模式选择

- 0 = 定时模式, Tx 时钟由 TXCLK<1:0>决定
- 1 = 计数模式

- TXSMS<2:0>= 111 计数模式 2, 定时器时钟为触发信号 TRGI
- TXSMS<2:0>≠ 111 计数模式 1, 定时器时钟为外部时钟 TxCK

13.3.8 Tz_CTL Tz 控制寄存器

表 13-12 Tz_CTL Tz 控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	TZCMS <2:0>		TZOFSI	TZOFSO	TZADAUTO	TZCLK<1:0>		TZDIR	TZCKBS <3:0>			TZEN	TZSY	TZCS			

TZCMS<2:0>: Tz 计数方式选择位

- 000 = 向下计数, 下溢时产生中断标志
- 100 = 向上计数, 上溢时产生中断标志。(初始化默认向上计数)
- x01 = 向上向下计数, 上溢时产生中断标志

x10 = 向上向下计数，下溢时产生中断标志

x11 = 向上向下计数，上溢和下溢时都产生中断标志

TZOFS1: Tz 溢出中断触发 AD 选择，TZADAUTO 为 1 时有效

0 = 禁止上溢中断触发 AD

1 = 使能上溢中断触发 AD

TZOFS0: Tz 溢出中断触发 AD 选择，TZADAUTO 为 1 时有效

0 = 禁止下溢中断触发 AD

1 = 使能下溢中断触发 AD

TZADAUTO: Tz 溢出中断自动触发 AD 控制位，当 AD 转换结束后自动清零

0 = 禁止 Tz 溢出中断自动触发 AD

1 = 使能 Tz 溢出中断自动触发 AD

TZCLK<1:0>: Tz 工作时钟选择位

00 = 选用 SCLK 时钟为 Tz 工作时钟

01 = 选用 HFCLK 时钟为 Tz 工作时钟

1x = 选用 LFCLK 时钟为 Tz 工作时钟

TZDIR: Tz 计数方向标志位

0 = 当前计数方向为向下计数

1 = 当前计数方向为向上计数

TZCKBS<3:0>: Tz 分频器 2 分频比选择位

0000 = 分频比为 1:1

0001 = 分频比为 1:2

0010 = 分频比为 1:3

0011 = 分频比为 1:4

0100 = 分频比为 1:5

0101 = 分频比为 1:6

0110 = 分频比为 1:7

0111 = 分频比为 1:8

1000 = 分频比为 1:9

1001 = 分频比为 1:10

1010 = 分频比为 1:11

1011 = 分频比为 1:12

1100 = 分频比为 1:13

1101 = 分频比为 1:14

1110 = 分频比为 1:15

1111 = 分频比为 1:16

TZON: Tz 使能位

0 = 禁止 Tz

1 = 使能 Tz

TZSY: Tz 计数模式外部触发脉冲输入同步控制位

TZCS= 0: 该位被忽略，Tz 使用内部时钟

TZCS= 1:

0 = 与外部触发脉冲输入同步

1 = 不与外部触发脉冲输入同步

TZCS: Tz 工作模式选择

0 = 定时模式，Tz 时钟为 TZCLK<1:0>确定

1 = 计数模式：

TZSMS<2:0>= 111 计数模式 2，定时器时钟为触发信号 TRGI

TZSMS<2:0>≠ 111 计数模式 1，Tz 时钟为外部时钟 TzCK

13.3.9 Tx_CCTCTL 比较器清零定时器 Tx 控制寄存器

表 13-13 Tx_CCTCTL 比较器清零定时器 Tx 控制寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W																											R/W	R/W	R/W	R/W		
位名																											CCTSEL <1:0>	CRSET	CCTEN			

CCTSEL<1:0>：用于清零定时器的比较器模块选择位

00 = 选择比较器 0

01 = 选择比较器 1

10 = 选择比较器 2

11 = 选择比较器 3

CRSET：CCT 清零信号上升/下降沿选择位

0 = 比较器输出信号检测到上升沿时，清零定时器

1 = 比较器输出信号检测到下降沿时，清零定时器

CCTEN：比较器清零定时器功能使能位

0 = 禁止比较器清零定时器功能

1 = 使能比较器清零定时器功能

13.3.10 Tz_CCTCTL 比较器清零定时器 Tz 控制寄存器

表 13-14 Tz_CCTCTL 比较器清零定时器 Tz 控制寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0						
R/W																					R/W	R/W													R/W	R/W	R/W	R/W
位名																					TZCKSEL <1:0>													CCTSEL <1:0>	CRSET	CCTEN		

CCTSEL<1:0>：用于清零定时器的比较器模块选择位

00 = 选择比较器 0

01 = 选择比较器 1

10 = 选择比较器 2

11 = 选择比较器 3

CRSET：CCT 清零信号上升/下降沿选择位

0 = 比较器输出信号检测到上升沿时，清零定时器

1 = 比较器输出信号检测到下降沿时，清零定时器

CCTEN：比较器清零定时器功能使能位

- 0 = 禁止比较器清零定时器功能
- 1 = 使能比较器清零定时器功能

13.3.11 Tx_CCRCTL CCR 功能控制寄存器

表 13-14 Tx_CCRCTL CCR 功能控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																							
R/W																									R/W																														
位名																									TZCCR0SEL	TXCCR1SEL	TXCCR0SEL	TZCCR0SEN	TXCCR1SEN	TXCCR0SEN	TXCCR1SEL	TXCCR0SEL	TZCCR0SEN	TXCCR1SEN	TXCCR0SEN	TXCCR1SEL	TXCCR0SEL	TZCCR0SEN	TXCCR1SEN	TXCCR0SEN	TXCCR1SEL	TXCCR0SEL	TZCCR0SEN	TXCCR1SEN	TXCCR0SEN	TXCCR1SEL	TXCCR0SEL	TZCCR0SEN	TXCCR1SEN	TXCCR0SEN	TXCCR1SEL	TXCCR0SEL	TZCCR0SEN	TXCCR1SEN	TXCCR0SEN

TZCCR0SEL: Tz_CCR0 触发 AD 计数方向选择位

- 0 = 向上计数且计数值等于 Tz_CCR0 寄存器值时，触发 AD
- 1 = 向下计数且计数值等于 Tz_CCR0 寄存器值时，触发 AD

TXCCR1SEL: Tx_CCR1 触发 AD 计数方向选择位

- 0 = 向上计数且计数值等于 Tx_CCR1 寄存器值时，触发 AD
- 1 = 向下计数且计数值等于 Tx_CCR1 寄存器值时，触发 AD

TXCCR0SEL: Tx_CCR0 触发 AD 计数方向选择位

- 0 = 向上计数且计数值等于 Tx_CCR0 寄存器值时，触发 AD
- 1 = 向下计数且计数值等于 Tx_CCR0 寄存器值时，触发 AD

TZCCR0SEN: Tz_CCR0 触发 AD 计数方向选择使能位

- 0 = 只要计数值等于 Tz_CCR0 寄存器值，就会触发 AD
- 1 = 触发 AD 时机由 TZCCR0SEL 选择

TXCCR1SEN: Tx_CCR1 触发 AD 计数方向选择使能位

- 0 = 只要计数值等于 Tx_CCR1 寄存器值，就会触发 AD
- 1 = 触发 AD 时机由 TXCCR1SEL 选择

TXCCR0SEN: Tx_CCR0 触发 AD 计数方向选择使能位

- 0 = 只要计数值等于 Tx_CCR0 寄存器值，就会触发 AD
- 1 = 触发 AD 时机由 TXCCR0SEL 选择

13.4 Tx/Tz 的工作原理

通过 Tx_CTL 寄存器的 TXEN 启动/禁止 Tx。Tx 有两种工作模式，分别为定时模式和计数模式，由 Tx_CTL 寄存器的 TXCS 位进行选择

Tx 定时模式时的计数时基有 SCLK、HFCLK 和 LFCLK 三种时钟可选，计数模式的计数时基为 TxCK/TRGI，计数时基经过预分频寄存器配置分频比之后作为计数时钟，每个计数时钟结束，计数值自动加一，通过 Tx_PPX 寄存器来设置计数周期。周期设置可以应用在 PWMx 的产生。

Tx 有两个分频器，预分频器（分频器 1）和后分频器（分频器 2）。预分频器用于产生定时器的时钟，后分频器用于控制中断信号产生的时机，每次溢出后分频器加一，直到等于设定值时，产生溢出中断。

定时器 Tx 有 3 种计数方式，向上计数、向下计数和向上向下计数，通过 Tx_CTL 寄存器的 TXDIR 计数方向标志位可以知道当前的计数方向。

当使用 Tx，可以使用自动触发 AD 的功能。AD 触发功能有两种形式，一种是通过 Tx 溢出信号触发，一种是通过 TxCCRy (x=5;y=0,1) 寄存器与 Tx 匹配时产生。

Tz 的工作原理与 Tx 相同，除了 Tz 只有一个触发 AD 寄存器 Tz_CCR0。

13.5 Tx/Tz 计数方式和中断

通过控制寄存器 Tx_CTL 的 TXCMS<2:0>位可以设置计数方式。Tx 可实现三种计数方式，分别是向上计数、向下计数和向上向下计数，前两种实现边沿对称 PWM 信号，最后一种实现中心对称 PWM 信号。通过 Tx_CTL 寄存器的 TXDIR 计数方向标志位可以知道当前的计数方向。注意，高级定时器计数溢出标志 TXIF 的产生机制与基本定时器和通用定时器均不同。

在向上计数方式时，当 Tx_CNT 寄存器与 PPX 相等时再加一，Tx 自动清零，同时发出上溢信号给分频器 2，分频器 2 值加一。

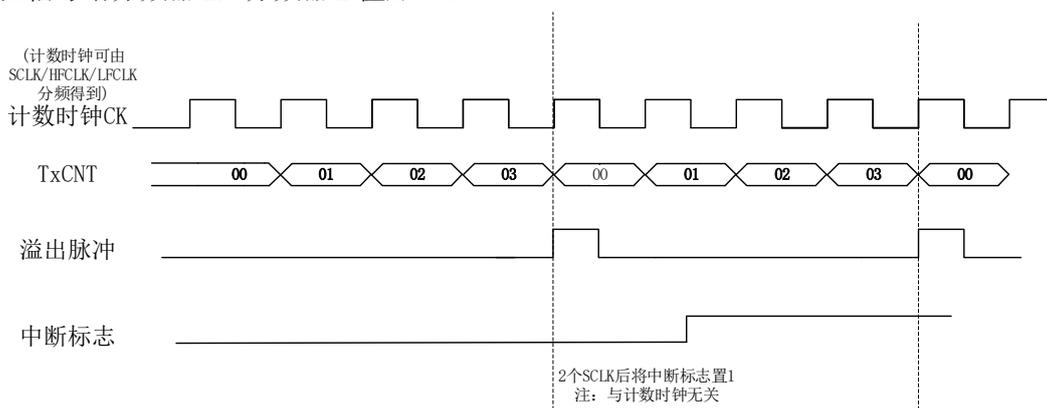


图 13-2 向上计数，计数周期 0x03，分频器 2 设置 1 分频

在向下计数方式时，当 Tx_CNT 寄存器的值计数到 0 时再减一，自动将 PPX 的值载入 Tx_CNT 寄存器中，并发出下溢信号给分频器 2，分频器 2 值加一。

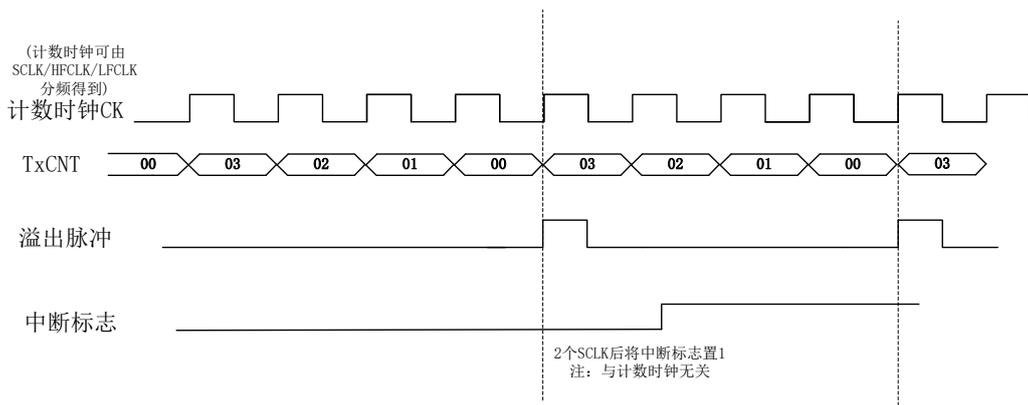


图 13-3 向下计数，计数周期 0x03，分频器 2 设置 1 分频

在向上向下计数方式时，当 Tx_CNT 寄存器与 PPX 寄存器相等时，Tx 不清零，而是自动从当前值开始向下计数，直到计数值为 0，周期寄存器 PPX 的值为计数周期的一半。这种模式通过寄存器 Tx_CTL 的 TXCMS<2:0>位控制产生中断溢出信号给分频器 2，并进行加一计数。在 TXCMS<2:0>=x01 时，Tx 计数值达到 PPX 后再加一，产生中断溢出信号给分频器 2；在 TXCMS<2:0>=x10 时，Tx_CNT 寄存器减到 1 后再减一，Tx 溢出，产生中断溢出信号给分频器 2；在 TXCMS<2:0>=x11 时，上溢和下溢均会产生中断溢出信号给分频器 2。

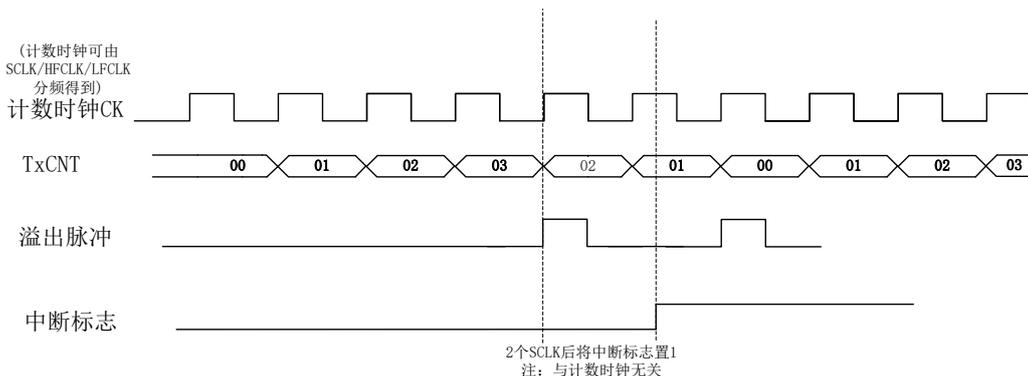


图 13-4 向上向下计数上下均溢出，计数周期 0x03，分频器 2 设置 1 分频

Tx 预分频器 2 的计数值达到设定值后将会使 ECCPx_EGIF 寄存器的溢出中断标志位 TXIF 置 1，如果 ECCPx_IE 寄存器的 TXIE 位为 1，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。

分频器 2 的分频比可通过寄存器 Tx_CTL 中的 TXCKBS<3:0>位设置为 1:1 至 1:16。当其设置为 1:1 时，每次寄存器 Tx_CNT 与 PPX 相等再加一（向上模式，向上向下计数方式中的的上溢触发和上下溢均触发）或者向下计数到 0 再减一（向下模式，向上向下模式中的下溢触发和上下溢均触发），将会使 Tx 溢出中断标志位 TXIF 置 1；当其设置为 1:2 时，寄存器 Tx_CNT 与 PPX 相等再加一或者向下计数到 0 再减一累计两次才会使 TXIF 置 1，以此类推。

将中断标志清除寄存器 ECCPx_SRIC 中的 TXIC 位置 1，可以清零 ECCPx_EGIF 寄存器中的 TXIF，TXIC 位需要手动清零。

13.6 Tx/Tz 工作模式

通过 Tx_CTL/Tz_CTL 寄存器的 TXCS/TZCS 位可以选择模块的工作模式，TXCS/TZCS 为 0 时模块工作在定时模式，TXCS/TZCS 为 1 时模块工作在计数模式。定时模式与计数模式的本质区别在于工作时使用的计数时钟源 Tsysclk 不同。具体可以参考本章节“高级定时器结构框图”。

13.6.1 定时模式

Tx_CTL/Tz_CTL 寄存器的 TXCS/TZCS 位为 0，则以 Tx_CTL/Tz_CTL 寄存器的 TXCLK<1:0>/TZCLK<1:0>位选择的工作时钟作为计数时钟源 Tsysclk。此时模块工作在定时模式。

13.6.2 计数模式

TXCS/TZCS 为 1，则根据 ECCPx_CTL2 寄存器的 TXSMS<2:0>/TZSMS<2:0>来选择计数时钟源 Tsysclk，TXSMS<2:0>/TZSMS<2:0>≠ 111 时以 TxCK/TzCK 引脚输入的上升沿进行计数（计数模式 1），TXSMS<2:0>/TZSMS<2:0>= 111 时以 TRGI 输入的上升沿进行计数（计数模式 2）。

Tx 的计数模式又有同步计数和异步计数两种方式。

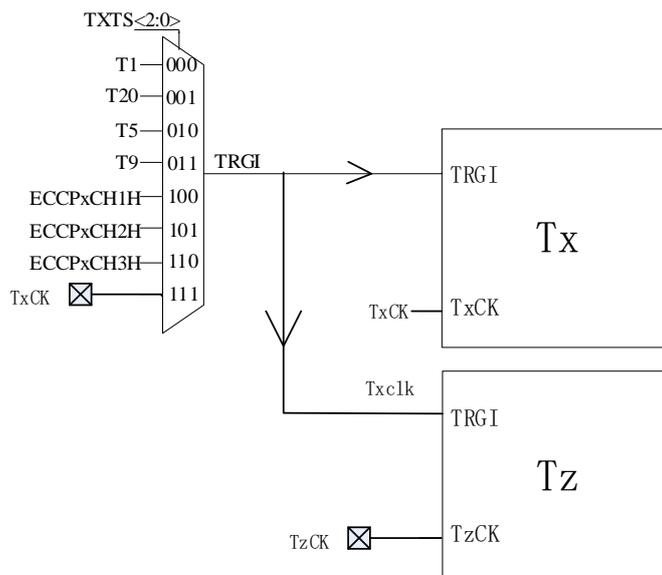
在计数模式下，如果 Tx_CTL/Tz_CTL 寄存器的控制位 TXSY/TZSY 清零，则 Tx/Tz 工作在异步计数模式，计数器根据 TxCK 引脚/Tz 引脚/TRGI 输入的脉冲上升沿进行计数；如果控制位 TXSY/TZSY 置 1，则 Tx/Tz 工作在同步计数模式，用模块内部工作时钟对 TxCK 引脚/Tz 引脚/TRGI 输入电平进行采样，可以实现 TxCK/TRGI 与内部相位时钟的同步。

Tx/Tz 可选择在两个不同的外部计数信号下计数，这两个外部计数信号分别为外部计数引脚 TxCK/TzCK 和触发输入 TRGI。

13.6.3 外部时钟引脚 Tx/Tz 说明

TxCK 和 TzCK 分别是 Tx 定时器和 Tz 定时器的外部时钟引脚，当工作在计数模式 1 时，Tx 的计数时钟源 Tsysclk 为 Tx 引脚，Tz 的计数时钟源 Tsysclk 为 Tz 引脚。

注意，Tx 和 Tz 的触发输入信号 TRGI 是共用的，通过 ECCPx_CTL2 寄存器的 TXTS<2:0>选择触发源。


图 13-5 TxCK 和 TzCK 框图

13.7 Tx/Tz 分频器

Tx 有两个分频器，预分频器（分频器 1）和后分频器（分频器 2）。预分频器用于产生定时器的时钟，后分频器用于控制中断信号产生的时机。

Tx 计数时钟 TXCLK 先经过分频器 1 分频后送到 Tx_CNT 寄存器，进行加一计数，分频器 1 通过设置寄存器 Tx_PRSC 中的 TXCKS<15:0>位，可将 TXCLK 进行 1:1~1:65536 分频。

分频器 2 的分频比可通过寄存器 Tx_CTL 中的 TXCKBS<3:0>位设置为 1:1 至 1:16。分频器 2 可以控制 Tx 中断产生的时机。

定时器关断或发生任何器件复位事件，分频器 1/2 的计数器将被清零。

Tz 的工作原理与 Tx 相同。

13.8 高级定时器中断

Tx 溢出中断、触发事件中断、以及以 Tx 为时基的更新事件中断、ECCP 的 1/2/3 通道捕捉比较中断对应于中断模块 INT_EIF1 中的 Tx 中断。当其中任一中断标志位置 1，且其中断使能，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。

同理，Tz 溢出中断、以 Tz 为时基的更新事件中断和 ECCP 的 4 通道捕捉比较中断对应于中断模块 INT_EIF1 中的 Tz 中断。

13.9 Tx/Tz 外部触发功能

13.9.1 触发模式

如果使能触发模式（ECCPx_CTL2 寄存器的 TXSMS<2:0>=100），且 TXEN=1，则在触发输入信号 TRGI 的上升沿 Tx 开始计数。同时，ECCPx_EGIF 寄存器中的 TXTIF 被置位，若使能了 ECCPx_IE 寄存器中的触发中断使能位 TXTIE，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求，见“Tx/Tz 触发 DMA(高级定时器)”。将中

断标志清除寄存器 ECCPx_SRIC 中的 TXTIC 位置 1，可以清零 TXTIF，TXTIC 位需要手动清零。

Tz 的工作原理与 Tx 相同。

13.9.2 门控模式

如果使能 Tx 门控功能 (ECCPx_CTL2 寄存器的 TXSMS<2:0>=101)，且 TXEN=1，则在触发输入信号 TRGI 为高电平时，启动 Tx，如果触发输入信号 TRGI 为低电平，禁止 Tx。在 TRGI 信号的边沿，ECCPx_EGIF 寄存器中的 TXTIF 被置位，若使能了 ECCPx_IE 寄存器中的触发中断使能位 TXTIE，则 TXIF 将被置位。且可产生 DMA 请求，见“Tx/Tz 触发 DMA(高级定时器)”。将中断标志清除寄存器 ECCPx_SRIC 中的 TXTIC 位置 1，可以清零 TXTIF，TXTIC 位需要手动清零。

使用该方式可粗略的对触发输入信号 TRGI 的高电平持续时间进行计算。

Tz 的工作原理与 Tx 相同。

13.9.3 复位模式

如果使能复位模式 (ECCPx_CTL2 寄存器的 TXSMS<2:0>=110)，且 TXEN=1，则在触发输入信号 TRGI 的上升沿复位 Tx_CNT 寄存器，若 ECCPx_CTL2 寄存器中的更新使能位：TXUDEN=1，则以 Tx 为时基的占空比、周期、输出控制、输出配置寄存器均被更新。同时，ECCPx_EGIF 寄存器中的 TXTIF 被置位，若使能了 ECCPx_IE 寄存器中的触发中断使能位 TXTIE，则 TXIF 将被置位。且可产生 DMA 请求，见“Tx/Tz 触发 DMA(高级定时器)”。将中断标志清除寄存器 ECCPx_SRIC 中的 TXTIC 位置 1，可以清零 TXTIF，TXTIC 位需要手动清零。

Tz 的工作原理与 Tx 相同。

13.10 定时器主从同步

13.10.1 定时器之间主从关系

任意一个基本定时器、通用定时器、高级定时器都可以作为触发源去触发其余的定时器，提供触发源的定时器被称为主定时器，被触发的定时器被称为从定时器。

此外，也可以把 TxCK 引脚作为触发源，用外部信号去触发从定时器。

13.10.2 使用一个定时器作为另一个定时器的预分频器（计数模式 2）

如基本定时器章节中主从关系，可以用两个 16 位定时器组成一个 32 位定时器，例如用定时器 T1 做为定时器 T5 的预分频器，具体配置如下：

- (1) T1_CTL2 寄存器的 TXMMS<2:0>=010，配置 T1 为主模式，选择 T1 在每次 T1 溢出中断标志 TXIF 产生时输出一个触发信号。
- (2) ECCP5_CTL2 寄存器 TXTS<2:0>=000，将 T5 的触发源配置为 T1 定时器 1 触发。
- (3) ECCP5_CTL2 寄存器 TXSMS<2:0>=111，同时将 T5_CTL 寄存器的 TXCS 置 1，配置 T5 工作在计数模式 2（从模式）。
- (4) 配置 T5_CTL1 的 TXEN=1 使能定时器 T5。
- (5) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

13.10.3 使用一个定时器使能另一个定时器（门控模式）

下例为用 CCP1 的输出使能 T5 的运行：

- (1) T1_CTL2 寄存器的 TXMMS<2:0>=100, T1 配置为主模式, CCP1CH1 作为触发输出 TRGO。
- (2) ECCP5_CTL2 寄存器的 TXTS<2:0>=000, 配置 T5 从 T1 获得触发源, 定时器 1 触发。
- (3) ECCP5_CTL2 寄存器的 TXSMS<2:0>=101, 配置 T5 的从模式使 T5 处于门控模式下。
- (4) 配置 T5_CTL1 的 TXEN=1 使能定时器 T5。
- (5) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

13. 10. 4 使用一个定时器启动另一个定时器（触发模式）

下例为使用 T1 的中断标志位 T1IF 启动 T5 的配置过程：

- (1) T1_CTL2 寄存器的 T1MMS<2:0>=010, T1 配置为主模式, T1 溢出中断脉冲信号 TXIF 作为触发输出 TRGO。
- (2) 配置 T1 的周期寄存器 T1_PPX。
- (3) ECCP5_CTL2 寄存器的 TXTS<2:0>=000, T5 的触发源为 T1, 定时器 1 触发。
- (4) ECCP5_CTL2 寄存器的 TXSMS<2:0>=100, T5 配置为触发模式。
- (5) 配置 T5_CTL1 的 TXEN=1 使能定时器 T5。
- (6) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

13. 10. 5 使用一个外部触发同步的启动两个定时器（同步触发）

下例为 T1 在 CCP1CH1 输入的上升沿启动, 同时启动 T5 的配置：

- (1) 配置 T1 的主模式, TXMMS<2:0>=001, T1 使能信号 TXEN 作为触发输出 TRGO。
- (2) 配置 T1 的触发输入源, TXTS<3:0>=1101, CCP1CH1 的输入。
- (3) 配置 T1 的从模式, TXSMS<2:0>=100, 触发模式。
- (4) ECCP5_CTL2 寄存器的 TXMSSYNC=1, 配置 T5 的主从模式同步位。
- (5) ECCP5_CTL2 寄存器的 TXTS<2:0>=000, 配置 T5 的触发输入源, 定时器 1 触发。
- (6) ECCP5_CTL2 寄存器的 TXSMS<2:0>=100, 配置 T5 的从模式, 触发模式。
- (7) 配置 T5_CTL1 的 TXEN=1 使能定时器 T5。
- (8) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

13. 10. 6 使用一个定时器复位另一个定时器（复位模式）

下例为使用 T1 的中断标志位 T1IF 复位 T5 的配置过程：

- (1) T1_CTL2 寄存器的 T1MMS<2:0>=010, T1 配置为主模式, T1 溢出中断脉冲信号 TXIF 作为触发输出 TRGO。
- (2) 配置 T1 的周期寄存器 T1_PPX。
- (3) ECCP5_CTL2 寄存器的 TXTS<3:0>=000, T5 的触发源为 T1, 定时器 1 触发。
- (4) ECCP5_CTL2 寄存器的 TXSMS<2:0>=110, T5 配置为复位模式。
- (5) 配置 T5_CTL1 的 TXEN=1 使能定时器 T5。
- (6) 配置 T1_CTL1 的 TXEN=1 使能定时器 T1。

13. 11 联立 Tx 和 Tz

将 ECCPx_CTL3 寄存器的 TUNITE 位置 1, 无需 Tx 或者 Tz 使能, 硬件自动将定时器 Tx 作为定时器 Tz 的预分频器, 组成一个周期值为 0xFFFF_FFFF 的 32 位定时器。将该位清

零后，定时器自动关断，且计数值清零。

联立 Tx 和 Tz 的使用方法：

- (1) 配置 Tx 定时器的初值、计数方式、预分频、周期等。
- (2) 配置 Tz 定时器的初值计数方式、预分频、周期等。
- (3) 将 ECCPx_CTL3 寄存器的 TUNITE 位置 1。

注：如果 Tx 发生了 N 次计数溢出，Tz 将计数 N-1 次。假设 Tz 计数初值为 0x0 且向上计数，假设 Tx 计数溢出了 5 次，此时 Tz 的计数值为 0x4。

13.12 Tx/Tz 更新

ECCPx 以 Tx/Tz 为时基，更新部分见“增强型捕捉/比较/PWM 模块(ECCP5/9)”章节的“PWM 更新锁定”部分。

13.13 Tx/Tz 定时器关断

如果将 Tx_CTL/Tz_CTL 寄存器的 TXEN/TZON 位置 0，定时器将关断。定时器关断时，会将计数值以及分频器 1/2 保留，并且将所有缓存寄存器复位为初始值，再次开启时又会更新缓存寄存器。

Tx/Tz 定时器关断事件触发 DMA 见“Tx/Tz 触发 DMA(高级定时器)”章节。

13.14 Tx/Tz 分配给 ECCP

以 ECCP5 为例，ECCP5 模块在使用时，需要使用 T5/T6 作为时基，捕捉/比较模式下使用 T5 作为通道 1/2/3 的时基，使用 T6 作为通道 4 的时基。PWM 模式下通道 1/2/3 使用 T5 作为计数时基，通道 4 可选择 T5 或 T6 作为计数时基，详细请参照 ECCP5 部分。

ECCP9 模块在使用时，需要使用 T9/T10 作为时基。

13.15 Tx/Tz 触发 AD

当使用 Tx，可以使用自动触发 AD 的功能。AD 触发功能有两种形式，一种是通过 Tx 溢出信号触发，一种是通过 Tx_CCRy (x=5;y=0,1) 寄存器与 Tx 匹配时产生，这种方法可在周期中的任意位置触发 AD。

以 Tx 触发 AD 为例，设置寄存器 Tx_CTL，选择合适的计数方式，通过 TXOFS<1:0> 选择 Tx 的上溢/下溢触发 AD，通过 TXADAUTO 使能 Tx 溢出触发 AD。TXADAUTO 信号在完成一次 AD 转换后自动清零。

可以设置 Tx_CCRy (y=0,1) 寄存器，当 Tx_CCRy 与 Tx 匹配时，将产生 AD 触发信号将 ADSTART 位置 1。如果要使用该功能，还需要在 ADCx_STATE 寄存器使能相应通道的 CCRy 触发开关。特别的，还可以通过配置 Tx_CCRCTL 寄存器，选择在特定的计数方向且匹配时产生 AD 触发信号。

Tz 触发 AD 与 Tx 相似，不同的是，Tz 只有一个任意位置触发 AD 的控制 Tz_CCR0 寄存器。

13.16 Tx/Tz 触发 DMA(高级定时器)

Tx 在 6 种情况下可以触发 DMA，触发事件 TRGI 触发、Tx 更新事件触发、Tz 更新事件触发、ECCPxCH1/2/3/4 通道上的捕捉/比较事件触发和关断事件触发。

触发事件触发 DMA，需要使能触发模式（TXSMS<2:0>=100），并将 TXTDE 置 1 使能触发事件 DMA 请求。等待触发事件到来即响应 DMA，触发事件也可以用软件触发（TXTRG 置 1）。

Tx 更新事件触发 DMA，需要将 TXUDE 置 1 使能更新事件 DMA 请求，配置寄存器产生更新事件即可响应 DMA。值得注意的是，使能定时器（TXEN 置 1）也会产生一次更新事件。

Tz 更新事件触发 DMA，需要将 TZUDE 置 1 使能更新事件 DMA 请求，配置寄存器产生更新事件即可响应 DMA。值得注意的是，使能定时器（TZON 置 1）也会产生一次更新事件。

ECCPxCH1/2/3/4 通道上的捕捉/比较事件触发 DMA，需要使能相应的 CHyDE，允许 ECCPxCHy 通道的捕捉/比较触发 DMA 请求。等待相应捕捉/比较事件发生即响应 DMA。

关断事件触发 DMA，需要使能 CLSDE，允许关断事件触发 DMA 请求。等待关断事件发生即可响应 DMA。关断事件也可以将 ECCPx_PXASCTL 寄存器的 PXASEy 置 1，直接产生对应通道的关断事件。

13.17 比较器清零定时器 Tx/Tz 功能

比较器清零定时器功能（CCT）可以实现：当检测到模拟比较器 CMP0/1/2/3 的输出信号电平翻转时，定时器硬件执行清零操作。

模拟比较器模块的相关配置参见模拟比较器模块章节；CCT 功能通过 Tx_CCTCTL 和 Tz_CCTCTL 寄存器进行配置。

四个比较器模块均可用于 CCT 功能，通过 CCTSEL<1:0>位对比较器模块进行选择；通过配置 CRSET 位，可以选择检测比较器输出信号的上升沿或者下降沿；CCTEN 位为 CCT 功能的使能位，置 1 以允许比较器清零定时器功能。

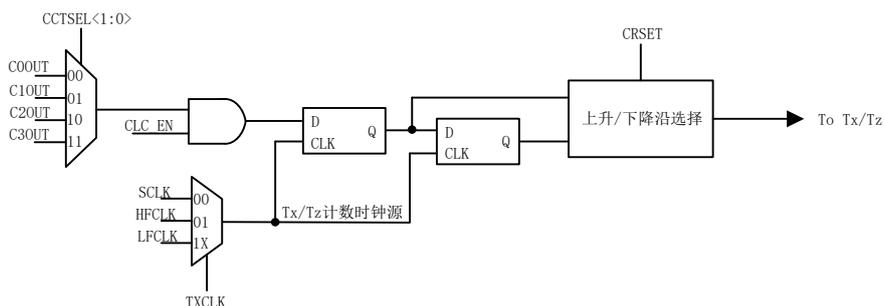


图 13-6 CCT 功能原理框图

注：

- (1) CxOUT 信号为模拟比较器模块的结果输出信号，比较器输出极性、范围控制和滤波功能都将影响该信号的结果。
- (2) CCT 功能使用 Tx/Tz 计数时钟源作为同步时钟，小于 1 个计数时钟源周期的信号将不会被采样，即不会触发清零操作。
- (3) 清零信号保持 1 个计数时钟源周期，之后 Tx/Tz 开始重新计数。

14 通用捕捉/比较/PWM 模块（CCP0/1/2/3/4/18/19/20/21/22/23）

14.1 概述

CCP 模块是通用型捕捉/比较/脉宽调制模块，在通用 CCP 模块中，采用通用定时/计数器做为该 CCP 的计数时基，可以用来实现捕捉功能、比较功能和 PWM 功能。

在 CCP0/1/2/3/4/18/19/22/23 模块中比较寄存器为 16 位的寄存器 CCPx_Ry (x=0,1,2,3,4,18,19,22,23; y=1,2,3,4)，该寄存器也用于 PWM 模式下的占空比设置；在 CCP20/21 中比较寄存器为 32 位的寄存器 CCPx_Ry (x=20,21; y=1,2,3,4)，该寄存器也用于 PWM 模式下的占空比设置。

在 CCP0/1/2/3/4/18/19/22/23 模块中捕捉寄存器为 16 位的寄存器 CCPx_Cy (x=0,1,2,3,4,18,19,22,23; y=1,2,3,4)，该寄存器为只读。在 CCP20/21 模块中捕捉寄存器为 32 位的寄存器 CCPx_Cy (x=20,21; y=1,2,3,4)，该寄存器为只读。

通用 CCP 主要功能包括：

- 16 位/32 位的捕捉功能
- 16 位/32 位的比较功能
- 16 位/32 位的 PWM 功能
- 支持 PWM 测量功能
- 4 个独立的通道
- PWM 支持边沿对其和中心对齐
- 支持单脉冲输出
- 更新事件、触发事件（触发模式、门控模式、复位模式）、捕捉事件、比较事件
- 可以产生 DMA 请求

14.2 CCP 模块结构框图

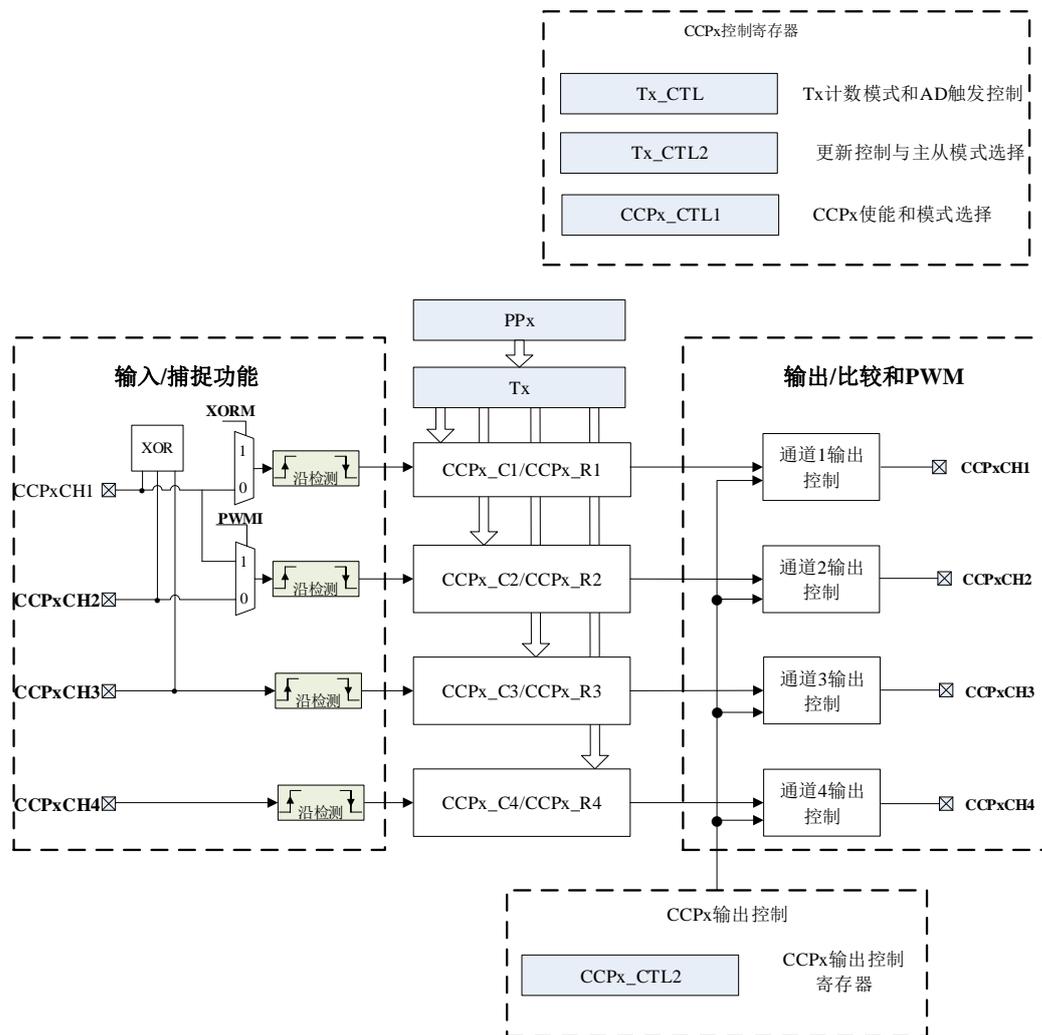


图 14-1 CCP 模块结构框图

- CCPxCHy 引脚¹: CCP 模块的通道引脚, 通过 IO 口重映射功能选择与具体的 IO 口映射。在捕捉模式下, 从 CCPxCHy 引脚输入捕捉源信号去触发捕捉; 在比较模式下, 从 CCPxCHy 引脚输出比较事件 (翻转、置 1、置 0 等); 在 PWM 模式下, 从 CCPxCHy 引脚输出 PWM 波形。此外, 此引脚也可用作定时器的触发输入 TRGI 和触发输出 TRGO。在功能确定后, 引脚的输入输出方向由硬件自动配置。

¹ 本章除特殊说明外, x=0,1,2,3,4,18,19,20,21,22,23; y=1,2,3,4。

14.3 CCP 模块相关寄存器

表 14-1 CCP 模块相关寄存器 1 (x=0,1,2,3,4,18,19,22,23)

偏移地址	寄存器	访问	功能描述	复位值
0x000	Tx_CNT	R/W	Tx 寄存器	0x0000 0000
0x004	Tx_CTL1	R/W	Tx 控制寄存器 1	0x0000 0480
0x008	Tx_CTL2	R/W	Tx 控制寄存器 2	0x0000 0000
0x00C	Tx_PRSC	R/W	Tx 预分频寄存器	0x0000 0000
0x010	Tx_PPX	R/W	Tx 周期寄存器	0x0000 FFFF
0x014	Tx_UDTIM	R/W	更新计数器	0x0000 0000
0x020	CCPx_C1	R	捕捉寄存器 1	0x0000 0000
0x024	CCPx_C2	R	捕捉寄存器 2	0x0000 0000
0x028	CCPx_C3	R	捕捉寄存器 3	0x0000 0000
0x02C	CCPx_C4	R	捕捉寄存器 4	0x0000 0000
0x030	CCPx_SRIC	R/W	CCPx 中断标志清除寄存器	0x0000 0000
0x034	CCPx_DF	R	CCP 触发 DMA 请求标志寄存器	0x0000 0000
0x040	CCPx_CTL1	R/W	CCPx 控制寄存器 1	0x0000 0000
0x044	CCPx_R1	R/W	比较/PWM 占空比寄存器 1	0x0000 0000
0x048	CCPx_R2	R/W	比较/PWM 占空比寄存器 2	0x0000 0000
0x04C	CCPx_R3	R/W	比较/PWM 占空比寄存器 3	0x0000 0000
0x050	CCPx_R4	R/W	比较/PWM 占空比寄存器 4	0x0000 0000
0x054	CCPx_CTL2	R/W	CCPx 控制寄存器 2	0x0000 0000
0x058	CCPx_CTL3	R/W	CCPx 控制寄存器 3	0x0000 0000
0x05C	CCPx_EGIF	R/W	CCPx 中断状态/事件产生寄存器	0x0000 0000

CCP0 基地址: 0x4000 0880

CCP1 基地址: 0x4000 0100

CCP2 基地址: 0x4000 0180

CCP3 基地址: 0x4000 0200

CCP4 基地址: 0x4000 0280

CCP18 基地址: 0x4000 1A80

CCP19 基地址: 0x4000 1B00

CCP22 基地址: 0x4000 1C80

CCP23 基地址: 0x4000 1D00

表 14-2 CCP 模块相关寄存器 2 (x=20,21)

偏移地址	寄存器	访问	功能描述	复位值
0x000	Tx	R/W	Tx 寄存器	0x0000 0000
0x004	Tx_CTL1	R/W	Tx 控制寄存器 1	0x0000 0400
0x008	Tx_CTL2	R/W	Tx 控制寄存器 2	0x0000 0000
0x00C	Tx_PRSC	R/W	Tx 预分频寄存器	0x0000 0000
0x010	Tx_PPX	R/W	Tx 周期寄存器	0xFFFFFFFF
0x014	Tx_UDTIM	R/W	更新计数器	0x0000 0000
0x020	CCPx_C1	R	捕捉寄存器 1	0x0000 0000
0x024	CCPx_C2	R	捕捉寄存器 2	0x0000 0000

偏移地址	寄存器	访问	功能描述	复位值
0x028	CCPx_C3	R	捕捉寄存器 3	0x0000 0000
0x02C	CCPx_C4	R	捕捉寄存器 4	0x0000 0000
0x030	CCPx_SRIC	R/W	CCPx 中断标志清除寄存器	0x0000 0000
0x034	CCPx_DF	R	CCP 触发 DMA 请求标志寄存器	0x0000 0000
0x040	CCPx_CTL1	R/W	CCPx 控制寄存器 1	0x0000 0000
0x044	CCPx_R1	R/W	比较/PWM 占空比寄存器 1	0x0000 0000
0x048	CCPx_R2	R/W	比较/PWM 占空比寄存器 2	0x0000 0000
0x04C	CCPx_R3	R/W	比较/PWM 占空比寄存器 3	0x0000 0000
0x050	CCPx_R4	R/W	比较/PWM 占空比寄存器 4	0x0000 0000
0x054	CCPx_CTL2	R/W	CCPx 控制寄存器 2	0x0000 0000
0x058	CCPx_CTL3	R/W	CCPx 控制寄存器 3	0x0000 0000
0x05C	CCPx_EGIF	R/W	CCPx 中断状态/事件产生寄存器	0x0000 0000

CCP20 基地址: 0x4000 1B80

CCP21 基地址: 0x4000 1C00¹

14. 3. 1 CCPx_Cy 捕捉寄存器

表 14-3 CCPx_Cy 捕捉寄存器(x=0,1,2,3,4,18,19,22,23)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
R/W																	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位名																	CCPXCy<15:0>																	

CCPXCy<15:0>: 在捕捉模式下时, 当对应的 CCPxCHy 引脚发生事件时, CCPx_Cy 寄存器捕捉 Tx 寄存器的 16 位值。

表 14-4 CCPx_Cy 捕捉寄存器(x=20,21)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位名	CCPXCy<31:0>																																

CCPXCy<31:0>: 在捕捉模式下时, 当对应的 CCPxCHy 引脚发生事件时, CCPx_Cy 寄存器捕捉 Tx 寄存器的 32 位值。

¹ CCP0/1/2/3/4/18/19/22/23 与 CCP20/21 除 Tx、Tx_PRSC、Tx_PPX、CCPx_Cy、CCPx_Ry 寄存器位数不一样外, 其他都一致。

14. 3. 2 CCPx_SRIC CCPx 中断标志清除寄存器

表 14-5 CCPx_SRIC CCPx 中断标志清除寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																							
R/W																											R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																											TXIC	TXUIC	TX TIC	CC4IC	CC3IC	CC2IC	CC1IC																						

TXIC: Tx 溢出中断标志清除位

- 0 = 不清除 Tx 溢出中断标志
- 1 = 清除 Tx 溢出中断标志

TXUIC: Tx 更新事件中标志清除位

- 0 = 不清除 Tx 更新事件中标志
- 1 = 清除 Tx 更新事件中标志

TX TIC: Tx 触发事件中标志清除位

- 0 = 不清除 Tx 触发事件中标志
- 1 = 清除 Tx 触发事件中标志

CC4IC: CCPx 通道 4 捕捉/比较中断标志清除位

- 0 = 不清除 CCPx 通道 4 捕捉/比较中断标志
- 1 = 清除 CCPx 通道 4 捕捉/比较中断标志

CC3IC: CCPx 通道 3 捕捉/比较中断标志清除位

- 0 = 不清除 CCPx 通道 3 捕捉/比较中断标志
- 1 = 清除 CCPx 通道 3 捕捉/比较中断标志

CC2IC: CCPx 通道 2 捕捉/比较中断标志清除位

- 0 = 不清除 CCPx 通道 2 捕捉/比较中断标志
- 1 = 清除 CCPx 通道 2 捕捉/比较中断标志

CC1IC: CCPx 通道 1 捕捉/比较中断标志清除位

- 0 = 不清除 CCPx 通道 1 捕捉/比较中断标志
- 1 = 清除 CCPx 通道 1 捕捉/比较中断标志

14. 3. 3 CCPx_DF CCP 触发 DMA 请求标志寄存器

表 14-6 CCPx_DF CCP 触发 DMA 请求标志寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
R/W																											R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位名																											TXUDF	TX TDF	CC4DF	CC3DF	CC2DF	CC1DF																							

TXUDF: Tx 更新事件触发 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

- 0 = 无 Tx 更新事件触发 DMA
- 1 = Tx 更新事件触发 DMA

TX TDF: Tx 触发事件触发 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

0 = 无 Tx 触发事件触发 DMA

1 = Tx 触发事件触发 DMA

CC4DF: CCPx 通道 4 触发 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

0 = CCPx 通道 4 无 DMA 触发

1 = CCPx 通道 4 触发 DMA

CC3DF: CCPx 通道 3 触发 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

0 = CCPx 通道 3 无 DMA 触发

1 = CCPx 通道 3 触发 DMA

CC2DF: CCPx 通道 2 触发 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

0 = CCPx 通道 2 无 DMA 触发

1 = CCPx 通道 2 触发 DMA

CC1DF: CCPx 通道 1 触发 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

0 = CCPx 通道 1 无 DMA 触发

1 = CCPx 通道 1 触发 DMA

14.3.4 CCPx_CTL1 CCP 控制寄存器 1

表 14-7 CCPx_CTL1 CCP 控制寄存器 1

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	CH4M<3:0>			CH3M<3:0>			CH2M<3:0>			CH1M<3:0>							

CHyM<3:0>: CCPx 模块的通道 y 模式选择位

0000 = 捕捉/比较/PWM 关闭 (复位 CCPx 模块)

0001 = 未使用 (保留)

0010 = 比较模式, 匹配时输出电平翻转 (CCyIF 置 1)

0011 = 未使用 (保留)

0100 = 捕捉模式, 在每个下降沿发生捕捉 (CCyIF 置 1)

0101 = 捕捉模式, 在每个上升沿发生捕捉 (CCyIF 置 1)

0110 = 捕捉模式, 每 4 个上升沿发生捕捉 (CCyIF 置 1)

0111 = 捕捉模式, 每 16 个上升沿发生捕捉 (CCyIF 置 1)

1000 = 比较模式, 比较匹配时输出高电平 (CCyIF 置 1)

1001 = 比较模式, 比较匹配时输出低电平 (CCyIF 置 1)

1010 = 比较模式, 比较匹配时产生软件中断 (CCyIF 置 1, CCPxCHy 引脚不受影响)

1011 = 比较模式, 触发特殊事件 (CCyIF 位置 1, CCP 复位对应的时基 Tx/Tz, 如果 ADC 使能将启动 AD 转换。)

11xx = PWM 模式

14.3.5 CCPx_Ry 比较/PWM 占空比寄存器

表 14-8 CCPx_Ry 比较/PWM 占空比寄存器(x=0,1,2,3,4,18,19,22,23)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	CCPXRY<15:0>																																

CCPXRY<15:0>: 在比较模式下, 16 位 CCPx_Ry 寄存器的值将不断与 Tx 寄存器的值相比较。当两者匹配时, CCPx 就会触发相应的事件。

在 PWM 模式下, CCPx_Ry 寄存器作为 PWMxy 的占空比寄存器。

表 14-9 CCPx_Ry 比较/PWM 占空比寄存器(x=20,21)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CCPXRY<31:0>																																	

CCPXRY<31:0>: 在比较模式下, 32 位 CCPx_Ry 寄存器的值将不断与 Tx 寄存器的值相比较。当两者匹配时, CCPx 就会触发相应的事件。

在 PWM 模式下, CCPx_Ry 寄存器作为 PWMxy 的占空比寄存器。

14.3.6 CCPx_CTL2 CCP 控制寄存器 2

表 14-10 CCPx_CTL2 CCP 控制寄存器 2(x=0,1,2,3,4,18,19,20,21,22,23)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
R/W																	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PXOC4CE	PXOC3CE	PXOC2CE	PXOC1CE	PXASS<1:0>				PXOC4<1:0>	PXOC3<1:0>	PXOC2<1:0>	PXOC1<1:0>						

PXOC4CE: 输出比较/PWM 信号清零使能位

0 = 通道 4 (CCPxCH4) 输出比较/PWM 信号/捕捉不受比较器输出影响。

1 = 一旦检测到比较器输出的高电平, 通道 4 (CCPxCH4) 输出比较信号清零/PWM 信号置为无效电平 (强制模式不受影响)。在捕捉模式下, 以比较器输出作为捕捉输入信号。

PXOC3CE: 输出比较/PWM 信号清零使能位

0 = 通道 3 (CCPxCH3) 输出比较/PWM 信号不受比较器输出影响。

1 = 一旦检测到比较器输出的高电平, 通道 3 (CCPxCH3) 输出比较信号清零/PWM 信号置为无效电平 (强制模式不受影响)

PXOC2CE: 输出比较/PWM 信号清零使能位

0 = 通道 2 (CCPxCH2) 输出比较/PWM 信号不受比较器输出影响。

1 = 一旦检测到比较器输出的高电平, 通道 2 (CCPxCH2) 输出比较信号清零/PWM 信号置为无效电平 (强制模式不受影响)

PXOC1CE: 输出比较/PWM 信号清零使能位

0 = 通道 1 (CCPxCH1) 输出比较/PWM 信号不受比较器输出影响。

1 = 一旦检测到比较器输出的高电平, 通道 1 (CCPxCH1) 输出比较信号清零 /PWM 信号置为无效电平 (强制模式不受影响)

PXASS<1:0>: 比较器检测选择位

00 = 检测比较器 1 的输出

01 = 检测比较器 2 的输出

10 = 检测比较器 3 的输出

11 = 保留

PXOC4<1:0>: 通道 4 输出控制

00 = PWM 输出, 高有效

01 = PWM 输出, 低有效

10 = 强制低电平输出

11 = 强制高电平输出

PXOC3<1:0>: 通道 3 输出控制

00 = PWM 输出, 高有效

01 = PWM 输出, 低有效

10 = 强制低电平输出

11 = 强制高电平输出

PXOC2<1:0>: 通道 2 输出控制

00 = PWM 输出, 高有效

01 = PWM 输出, 低有效

10 = 强制低电平输出

11 = 强制高电平输出

PXOC1<1:0>: 通道 1 输出控制

00 = PWM 输出, 高有效

01 = PWM 输出, 低有效

10 = 强制低电平输出

11 = 强制高电平输出

14. 3. 7 CCPx_CTL3 CCPx 控制寄存器 3

表 14-11 CCPx_CTL3 CCPx 控制寄存器 3

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PXSPMST	PXPWMI	PAXORM	PASPM	TXTDE	CC4DE	CC3DE	CC2DE	CC1DE	TXUDE	TXUIE	TXTIE	CC4IE	CC3IE	CC2IE	CC1IE	

PXSPMST: 单脉冲输出模式选择

0 = 单脉冲输出后不关闭定时器使能位

1 = 单脉冲输出后关闭定时器使能位

PXPWMI: PWM 输入测量模式使能位

0 = 未使能

- 1 = 使能 PWM 输入测量模式
- PXXORM: 输入异或使能位
- 0 = 未使能
- 1 = 使能输入异或, 将 CCPxCH1H、CCPxCH2H、CCPxCH3H 的输入信号连接到异或门, 异或门输出映射到捕捉/比较通道 1 上。
- PXSPM: 单脉冲输出模式
- 0 = 单脉冲输出模式未使能
- 1 = 使能单脉冲输出模式
- TXTDE: 允许触发事件的 DMA 请求
- 0 = 禁止触发事件的 DMA 请求
- 1 = 允许触发事件的 DMA 请求
- CC4DE: 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
- CC3DE: 允许捕获/比较 3 的 DMA 请求
- 0 = 禁止捕获/比较 3 的 DMA 请求
- 1 = 允许捕获/比较 3 的 DMA 请求
- CC2DE: 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
- CC1DE: 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
- TXUDE: 允许更新事件的 DMA 请求
- 0 = 禁止更新事件的 DMA 请求
- 1 = 允许更新事件的 DMA 请求
- TXUIE: Tx 更新事件中中断使能
- 0 = 禁止 Tx 触发中断
- 1 = 允许 Tx 触发中断
- TXTIE: Tx 触发事件中中断使能
- 0 = 禁止 Tx 触发中断
- 1 = 允许 Tx 触发中断
- CC4IE: CCPx 通道 4 捕捉/比较中断使能
- 0 = 禁止 CCPx 通道 4 捕捉/比较中断
- 1 = 允许 CCPx 通道 4 捕捉/比较中断
- CC3IE: CCPx 通道 3 捕捉/比较中断使能
- 0 = 禁止 CCPx 通道 3 捕捉/比较中断
- 1 = 允许 CCPx 通道 3 捕捉/比较中断
- CC2IE: CCPx 通道 2 捕捉/比较中断使能
- 0 = 禁止 CCPx 通道 2 捕捉/比较中断
- 1 = 允许 CCPx 通道 2 捕捉/比较中断
- CC1IE: CCPx 通道 1 捕捉/比较中断使能
- 0 = 禁止 CCPx 通道 1 捕捉/比较中断
- 1 = 允许 CCPx 通道 1 捕捉/比较中断

14. 3. 8 CCPx_EGIF CCPx 中断状态/事件产生寄存器
表 14-12 CCPx_EGIF CCPx 中断状态/事件产生寄存器

复位值	0	31	0	30	0	29	0	28	0	27	0	26	0	25	0	24	0	23	0	22	0	21	0	20	0	19	0	18	0	17	0	16	0	15	0	14	0	13	0	12	0	11	0	10	0	9	0	8	0	7	0	6	0	5	0	4	0	3	0	2	0	1	0	0				
R/W																																	R	0	11	R	0	10	R	0	9	R	0	8	R	0	7	R	0	6	R/W	0	5	R/W	0	4	R/W	0	3	R/W	0	2	R/W	0	1	R/W	0	0
位名																																	TXUIF	TXTIF	CC4IF	CC3IF	CC2IF	CC1IF	TXTRG	CC4G	CC3G	CC2G	CC1G	UARXEN																								

TXUIF: Tx 更新事件中断标志

- 0 = 未发生 Tx 更新中断
- 1 = 发生 Tx 更新中断

TXTIF: Tx 触发事件中断标志

- 0 = 未发生 Tx 触发中断
- 1 = 发生 Tx 触发中断

CC4IF: CCPx 通道 4 捕捉/比较中断标志

- 0 = CCPx 通道 4 无捕获/比较事件发生
- 1 = CCPx 通道 4 发生捕获/比较事件

CC3IF: CCPx 通道 3 捕捉/比较中断标志

- 0 = CCPx 通道 3 无捕获/比较事件发生
- 1 = CCPx 通道 3 发生捕获/比较事件

CC2IF: CCPx 通道 2 捕捉/比较中断标志

- 0 = CCPx 通道 2 无捕获/比较事件发生
- 1 = CCPx 通道 2 发生捕获/比较事件

CC1IF: CCPx 通道 1 捕捉/比较中断标志

- 0 = CCPx 通道 1 无捕获/比较事件发生
- 1 = CCPx 通道 1 发生捕获/比较事件

TXTRG: 产生触发事件（该位由软件置‘1’，用于产生一个触发事件，定时器停止计数后由硬件自动清‘0’。）

- 0 = 无动作
- 1 = CCPx_EGIF 寄存器中的 TXTIF=1,若使能相应的中断及 DMA,则产生相应的中断及 DMA

CC4G: 产生通道 4 捕捉/比较事件（该位由软件置‘1’，由硬件自动清‘0’。）

- 0 = 无动作
- 1 = 在通道 4 上产生一个捕捉/比较事件。
(详细描述参考 CC1G)

CC3G: 产生通道 3 捕捉/比较事件（该位由软件置‘1’，由硬件自动清‘0’。）

- 0 = 无动作
- 1 = 在通道 3 上产生一个捕捉/比较事件。
(详细描述参考 CC1G)

CC2G: 产生通道 2 捕捉/比较事件（该位由软件置‘1’，由硬件自动清‘0’。）

- 0 = 无动作
- 1 = 在通道 2 上产生一个捕捉/比较事件。
(详细描述参考 CC1G)

CC1G: 产生通道 1 捕捉/比较事件（该位由软件置‘1’，由硬件自动清‘0’。）

0 = 无动作

1 = 在通道 1 上产生一个捕捉/比较事件。

若通道 1 被配置为比较模式：触发比较模式对应的功能，置 CC1IF=1，若使能相应的中断及 DMA，则产生相应的中断及 DMA 请求

若通道 1 被配置为捕捉模式：当前的计数值被捕捉至 CCPx_R1 寄存器，且置 CC1IF=1，若使能相应的中断及 DMA,则产生相应的中断及 DMA 请求

USARTRXEN: 产生串口接收信号测量事件

0 = 禁止测量串口接收信号

1 = 使能测量串口接收信号

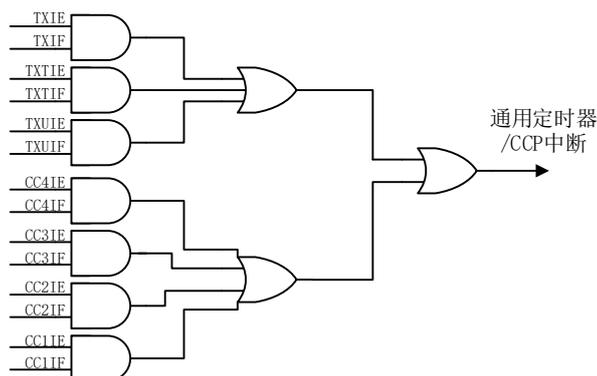


图 14-2 通用定时器/CCP 中断框图

14.4 捕捉模式

在捕捉模式下,当对应的CCPxCHy引脚发生捕捉事件时,CCPx_Cy寄存器捕捉Tx_CNT寄存器的值。CCPxCHy引脚上触发捕捉的事件可被定义为以下四者之一,并且由CCPx_CTL1寄存器中的CHyM<3:0>位配置:

- 每个下降沿
- 每个上升沿
- 每4个上升沿
- 每16个上升沿

通过模式选择位CHyM<3:0>选择事件类型。当一个捕捉事件发生时,CCPx_EGIF寄存器中的CCyIF置1,它必须用CCPx_SRIC寄存器的CCyIC位置1后才会清零。如果在CCPx_Cy寄存器中的值被读取之前发生另一次捕捉,那么之前捕捉的值将被新捕捉的值覆盖。

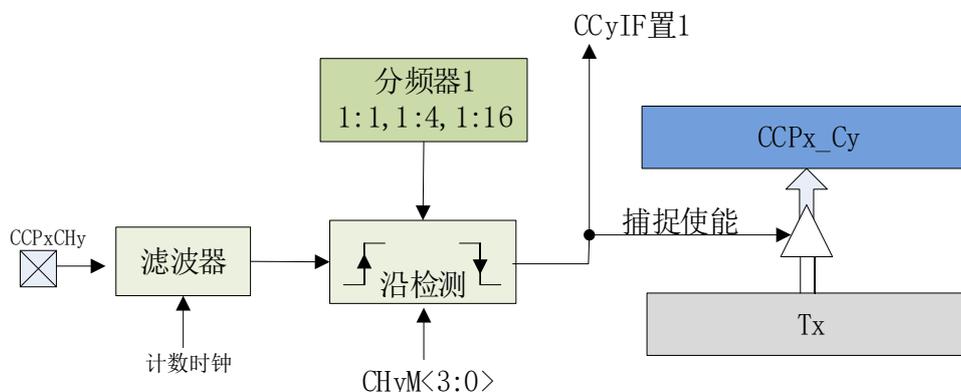


图 14-3 捕捉模式原理框图

CCPxCHy引脚信号经过两级的计数时钟下降沿滤波后进行沿检测,所以至少要有两个计数时钟脉宽的信号才会被捕捉到。

当捕捉模式改变时,可能会产生错误的捕捉中断。用户应该保持CCPxIE寄存器中的CCyIE中断允许位清零以避免产生误中断。在操作模式发生任何改变之后也应清零CCPx_EGIF寄存器中的中断标志位CCyIF。

CCPx_CTL1寄存器中的CHyM<3:0>位指定了4种捕捉预分频器设置(对应的是1,4,16个上升沿发生捕捉)。每当关闭CCPx模块或禁止捕捉模式时,就会清零捕捉预分频器计数器。这意味着任何复位都将清零捕捉预分频计数器。

注:从一种捕捉预分频比切换到另一种捕捉预分频比不会将捕捉预分频计数器清零,但可能会产生误中断。要避免出现这种不期望的操作,应在改变捕捉预分频比前通过将CCPx_CTL1寄存器清零关闭该模块。

14.4.1 PWM 测量模式

该模式是捕捉模式的一个特例,除下列区别外,操作与输入捕捉模式相同:

- (1) 设置CCPx_CTL3寄存器的PXPWMI=1,选择PWM测量模式。(将通道1的PWM输入信号映射到通道1和通道2的输入,即通道1的捕捉信号同时可以触发通道1和通道2的捕捉。)
- (2) Tx的Tx_CTL2寄存器里的TXSMS<2:0>设为110:复位模式,在触发信号TRGI出现上升沿时初始化定时器,并对寄存器进行更新。

- (3) Tx 的 Tx_CTL2 寄存器里的 TXTS<3:0>设为 1101: CCPxCH1 的输入作为触发输入 (TRGI)
- (4) 设置 CCPx_CTL1 的 CH1M <3:0>=0100/0101 在通道 1 的每个下降/上升沿发生捕捉;
- (5) CH2M <3:0>=0101/0100 在通道 2 的每个上升/下降沿发生捕捉。即通道 1 和通道 2 的捕捉沿时反的。

注: 如果配置通道 1 上升沿捕捉, 通道 2 下降沿捕捉, 则 CCPx_C1 寄存器捕捉到的是被测信号的周期值, CCPx_C2 寄存器捕捉到的是被测信号的正脉宽; 如果配置通道 1 下降沿捕捉, 通道 2 上升沿捕捉, 则 CCPx_C1 寄存器捕捉到的是被测信号的正脉宽, CCPx_C2 寄存器捕捉到的是被测信号的周期值。

如果将 CCPx_EGIF 寄存器中的 USARTRXEN 位置 1, 则使能测量串口输入信号:

设置与上面相同, 只是硬件把通道 1 的 PWM 输入 CCPxCH1 信号换成了 USART0_RX 信号。USARTRXEN 置 1 后, 测量模式的测量对象从通道 1 的输入信号变成了 USART0_RX 引脚上的输入信号。

14.4.2 模块相关引脚说明

(E)CCPxCH1、(E)CCPxCH2、(E)CCPxCH3、(E)CCPxCH4 可作为 PWM 输出、捕捉输入、比较输出以及可供主从模式中触发输入 TRGI 或者触发输出 TRGO 选择。在选择引脚用于某种功能后, 其输入输出方向由硬件自动配置。(初始情况下都为输出)

14.4.3 PWM 异或输入以及霍尔传感器接口功能

通过将 CCPx_CTL3 寄存器的 PXXORM 位置 1, 允许 CCPx 的异或输入模式, 将 3 个引脚输入信号 CCPxCH1、CCPxCH2、CCPxCH3 连接到一个异或门的输入端, 三个输入中任意一个发生电平变化都会产生 2 个计数时钟宽度的脉冲, 用作通道 1 的捕捉输入, 捕捉一次后产生捕捉比较中断, 并将定时器清零。下面介绍此特性用于霍尔传感器接口的例子, 这里以通用定时器 T1 作为接口定时器, 霍尔输入连接到 T1 定时器, 要求每次任一霍尔输入上发生变化之后, 改变高级定时器 T5 的 PWM 配置:

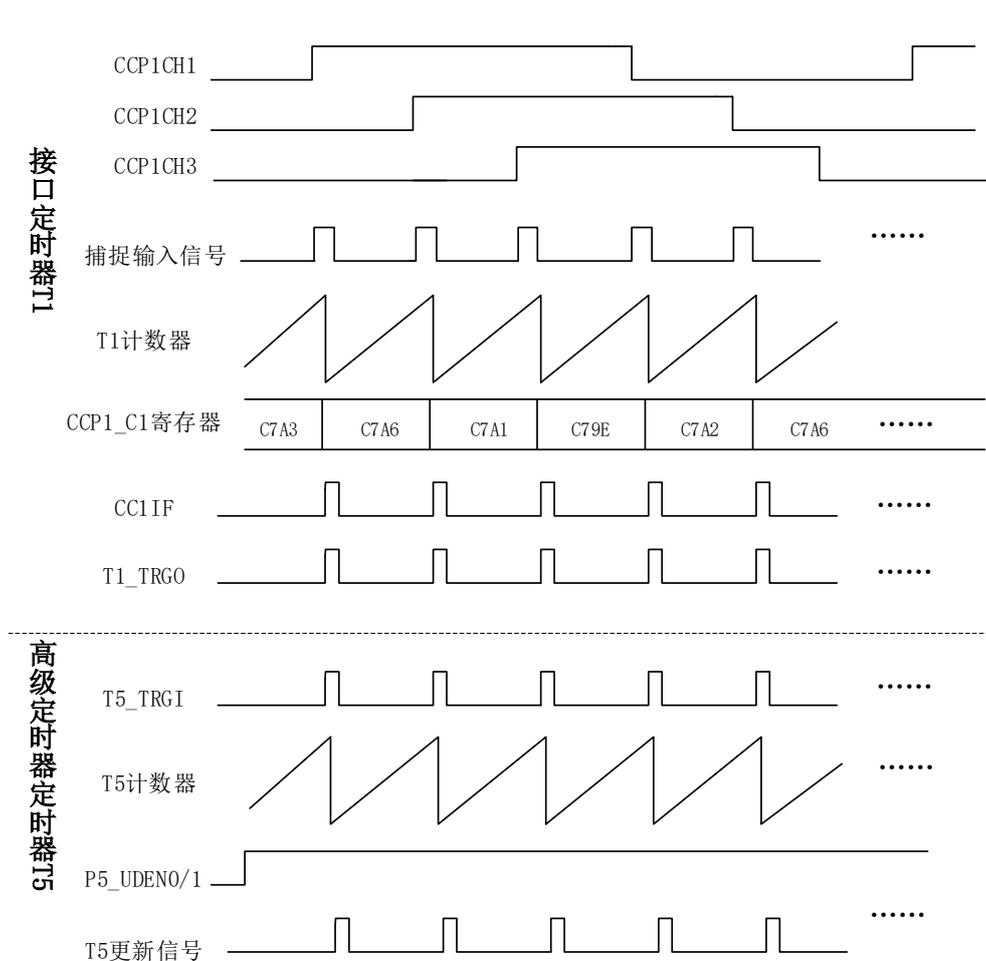


图 14-4 霍尔元件实例原理框图

- (1) 将 CCP1_CTL3 寄存器的 PXXORM 位置 1，将通用定时器 T1 设置为异或输入模式。
- (2) 配置 CCP1_CTL1 寄存器的 CH1M<3:0>=0101，通道 1 每个上升沿发生捕捉。
- (3) 配置 T1_CTL2 寄存器的 TXMMS<2:0>=011，以 CH1IF 脉冲作为触发输出 T1TRGO。
- (4) 配置 ECCP5_CTL2 寄存器的 TXTS<2:0>=000，以定时器 T1 的触发输出 T1TRGO 作为触发输入 T5TRGI。
- (5) 配置 ECCP5_CTL2 寄存器的 TXSMS<2:0>=110，将 T5 从模式设置为复位模式，在触发输入信号 T5TRGI 的上升沿初始化定时器，ECCP5_CTL2 寄存器的 PXUDENO/1 为 1，更新相关寄存器。配置 T5 的周期以及预分频器，确保 T5 计数器周期大于传感器上的两次变化的时间间隔。
- (6) 使能 T1 以及 T5 定时器。

使用高级定时器产生 PWM 信号驱动马达时，可以用另一个通用定时器作为接口定时器来连接霍尔传感器。每当三个输入之一变化时，计数器从 0 开始计数，这样就产生一个由霍尔输入端的任何变化而触发的时间基准，捕获值反映了两个输入变化间的时间延迟，给出马达速度的信息，同时产生捕捉中断去复位高级定时器来更新相应的 PWM 控制寄存器。

14.4.4 比较器的输出触发捕捉

CCP1_CTL2 寄存器的 PXOCyCE 位置 1，比较器的输出可设置成连接到 CCPx 的捕捉

输入通道 y (CCPxCHy), 此模式下可根据比较器的输出变化触发捕捉。(y=1,2,3,4)

14.4.5 捕捉中断与 DMA 请求

当一个捕捉发生时, CCPx_EGIF 寄存器中的 CCyIF 置 1, 它必须用 CCPx_SRIC 寄存器的 CCyIC 位置 1 后才会清零, CCyIC 置 1 后需要软件清零以保证后续中断标志位能被置 1。若 CCPx_CTL3 寄存器中对应的 CCyIE 位为 1, 则中断模块的 INT_EIF1 寄存器中对应的中断标志位置 1。

CCP 捕捉触发 DMA 的使用详细参考通用定时/计数器章节“Tx 触发 DMA(通用定时器)”部分。

14.5 比较模式

14.5.1 比较功能

在比较模式下, 比较寄存器 CCPx_Ry 的值将不断与 Tx 寄存器的值相比较。当两者匹配时, CCPx 模块可能会出现以下几种情况:

- CCPxCHy 的输出电平翻转
- CCPxCHy 输出高电平
- CCPxCHy 输出低电平
- 产生特殊事件触发信号
- 仅产生软件中断

CCPxCHy 引脚的动作取决于 CCPx_CTL1 寄存器中 CHyM<3:0>控制位的值。

所有比较模式都会产生中断。(CCyIE=1 的情况下)

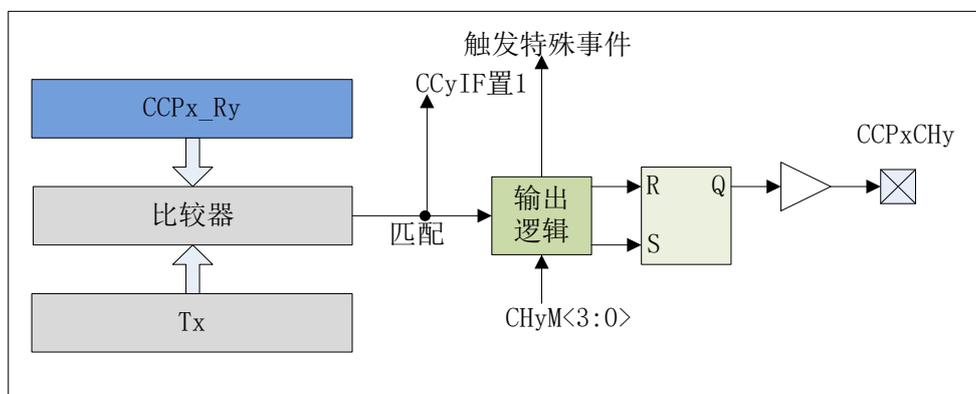


图 14-5 比较模式原理框图

- (1) 当选择输出电平翻转模式 (CHyM<3:0>=0010) 时, 比较匹配时, CCPxCHy 引脚的输出电平翻转, 并将 CCyIF 置 1。
- (2) 当选择普通比较模式时 (CHyM<3:0>=1000 或 1001) 时, 比较匹配时, CCPxCHy 引脚输出高电平或者低电平, 并且将 CCyIF 置 1。
- (3) 当选择了软件中断触发模式时 (CHyM<3:0>=1010) 时, 比较匹配时, 将 CCyIF 置 1, 但是 CCP 模块不会控制 CHy 引脚。
- (4) 当选择了特殊事件触发模式 (CHyM<3:0>=1011) 时, 比较匹配时, CCP 会产生特殊事件触发输出, 将 CCyIF 置 1, 但是 CCP 模块不会控制 CHyH 引脚。此时如果 ADC 已使能, 将启动 AD 转换。但 Tx 寄存器不会立即复位, 直到 Tx 计数脉冲的下一个上升沿才复位。从而使 CCPx_Ry 寄存器实际上成为了 Tx 的可编程周期寄存器。

14.5.2 比较中断与 DMA 请求

在比较模式下，当 CCPx_Ry 的值与 Tx 寄存器的值相等时，CCPx_EGIF 寄存器中的 CCyIF 置 1，它必须用 CCPx_SRIC 寄存器的 CCyIC 位置 1 后才会清零，CCyIC 置 1 后需要软件清零以保证后续中断标志位能被置 1。若 CCPx_CTL3 寄存器中对应的 CCyIE 位为 1，则中断模块的 INT_EIF1 寄存器中对应的中断标志位置 1。

CCP 比较触发 DMA 的使用详细参考通用定时/计数器章节“Tx 触发 DMA(通用定时器)”部分。

14.6 PWM 模式

PWMx 为四通道输出的通用 PWM 模块。通过配置 CCPx_CTL1 寄存器中的 CHyM<3:0> 位可分别使能或关闭 PWMx 模块的各个通道。

下图显示了 PWMx 通道 y 的逻辑框图。其中 PPX 为 PWMx 模块的周期寄存器，CCPx_Ry 为 PWMx 模块占空比设置寄存器，使用 PWMx 时需要将定时器 Tx 配置给 PWMx 做定时用。改变 PPX 和 CCPx_Ry 的值可产生不同的 PWMx 周期和 PWMx 占空比。启动 PWMx 后，在对应的 CCPxCHy 引脚输出 PWM 脉冲。

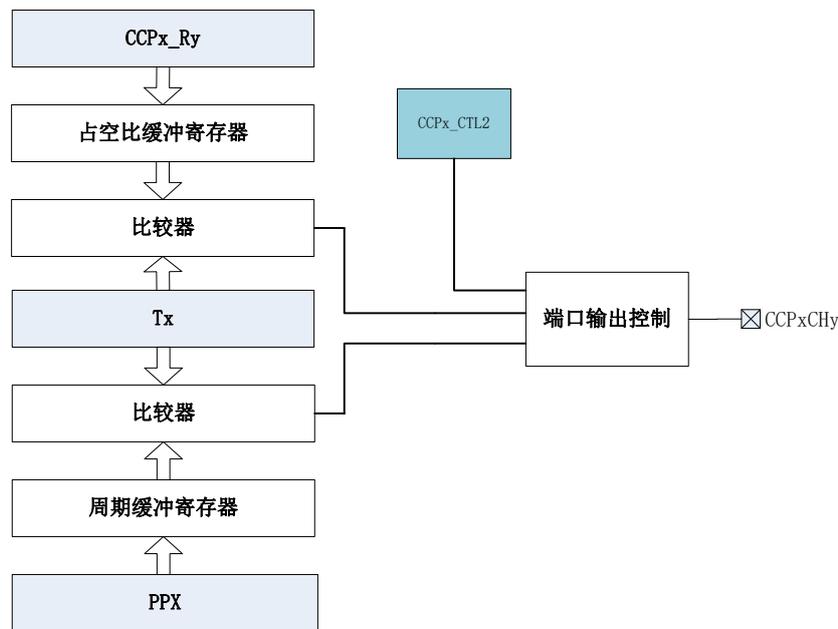


图 14-6 PWM 原理框图

14.6.1 PWMx 的周期

PWMx 的周期通过寄存器 PPX 进行设置，在边沿对齐模式下和中心对齐模式下，周期的计算方式不同，中心对齐模式下得到的周期是周期寄存器设置值的两倍。PWMx 边沿对齐和中心对齐的周期分别通过下式进行计算。

$$\text{PWMx周期} = (\text{PPX} + 1) \cdot \text{Tsyclk} \cdot (\text{Tx 预分频比}) \quad \text{公式 14-1}$$

$$\text{PWMx周期} = \text{PPX} \cdot \text{Tsyclk} \cdot \text{Tx 预分频比} \cdot 2 \quad \text{公式 14-2}$$

当 Tx_CTL2 寄存器中的 TXUDEN=1，且 TXUDEVT=0 时，周期寄存器 PPX 在 Tx 的一个计数周期结束时更新到缓冲器中，所以每个 PWM 周期更新与 PWM 周期同步。（注意：当 Tx 启动时周期寄存器立即更新）。

当 Tx_CTL2 寄存器中的 TXUDEN=1，且 TXUDEVT=1 时，周期寄存器 PPX 立即更新

到缓冲器中，且将定时器 Tx 清零。

14.6.2 PWMx 占空比

PWMx 的占空比通过寄存器 CCPx_Ry 进行设置。在不同模式下占空比计算方式不一致。在边沿对齐模式下，脉冲宽度和占空比通过下式计算：

$$\text{脉冲宽度} = \text{CCPx_Ry} \cdot T_{\text{sysclk}} \cdot T_x \text{ 预分频比} \quad \text{公式 14-3}$$

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM 周期}} = \frac{\text{CCPx_Ry}}{\text{PPx} + 1} \quad \text{公式 14-4}$$

在中心对齐模式下，脉冲宽度和占空比通过下式计算：

$$\text{脉冲宽度} = \text{CCPx_Ry} \cdot T_{\text{sysclk}} \cdot T_x \text{ 预分频比} \cdot 2 \quad \text{公式 14-5}$$

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM 周期}} = \frac{\text{CCPx_Ry}}{\text{PPx}} \quad \text{公式 14-6}$$

占空比寄存器在 Tx 的一个计数周期结束时更新到缓冲器中，占空比和周期寄存器的更新与 PWMx 周期同步。

当 Tx_CTL2 寄存器中的 TXUDEN=1，且 TXUDEVT=0 时，占空比寄存器 CCPx_Ry 在 Tx 的一个计数周期结束时更新到缓冲器中，所以每个占空比寄存器的更新与 PWM 周期同步。

当 Tx_CTL2 寄存器中的 TXUDEN=1，且 TXUDEVT=1 时，占空比寄存器 CCPx_Ry 立即更新到缓冲器中，且将定时器 Tx 清零。

注：

- 1、当 Tx 启动时占空比寄存器立即更新
- 2、Tsysclk 为工作时钟频率的倒数

14.6.3 PWMx 分辨率

分辨率的计算公式如下式所示：

$$\text{分辨率}(\%) = F_{\text{PWM}} / F_{\text{Txclk}} \times 100\% \quad \text{公式 14-7}^1$$

$$\text{分辨率}(\text{bits}) = \text{Log}_2(T_{\text{PWM}} / T_{\text{Txclk}}) \quad \text{公式 14-8}$$

14.6.4 边沿对齐 PWM 信号

当 Tx_CTL 控制寄存器中的 TXCMS<2:0>计数模式选择位设置成 000/100 时，产生的 PWM 信号为边沿对齐的 PWM 信号。在该模式下，PWM 信号的周期由 PPX 周期寄存器决定，占空比由寄存器 CCPx_Ry 决定。当 PWM 输出为高有效时，周期开始时输出高电平，占空比与定时器匹配时电平变低；当 PWM 输出低有效时，周期开始时输出低电平，占空比与定时器匹配电平变高。

在向上计数模式下，如果占空比寄存器设置成 0，那么在整个 PWMx 周期都输出无效电平。如果占空比寄存器设置大于 PPX 周期寄存器的值，那么在整个 PWM 周期都输出有效电平。

在向下计数模式下，如果占空比寄存器设置等于周期值，那么在整个 PWMx 周期都输

¹ Txclk 为 Tsysclk 预分频后的计数时钟

出无效电平。如果占空比寄存器设置大于 PPX 周期寄存器的值，那么在整個 PWM 周期都输出有效电平。

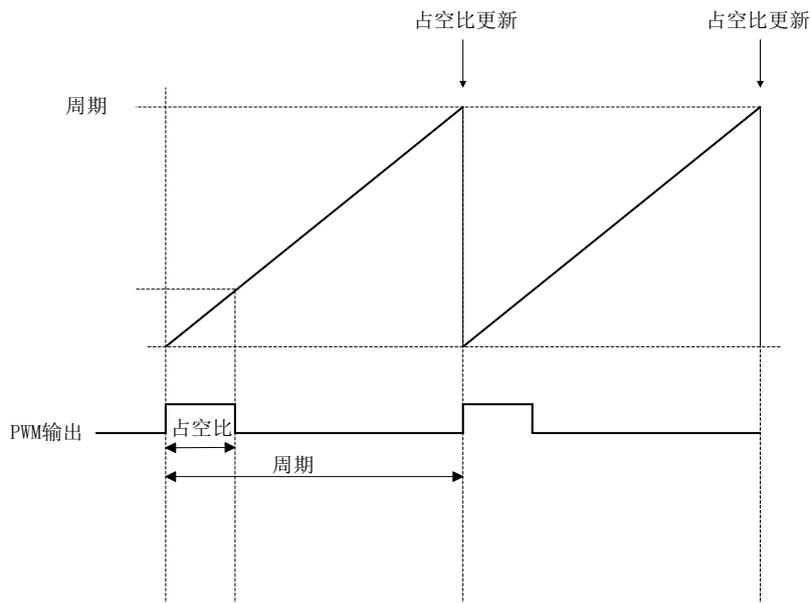


图 14-7 向上计数边沿对齐 PWM

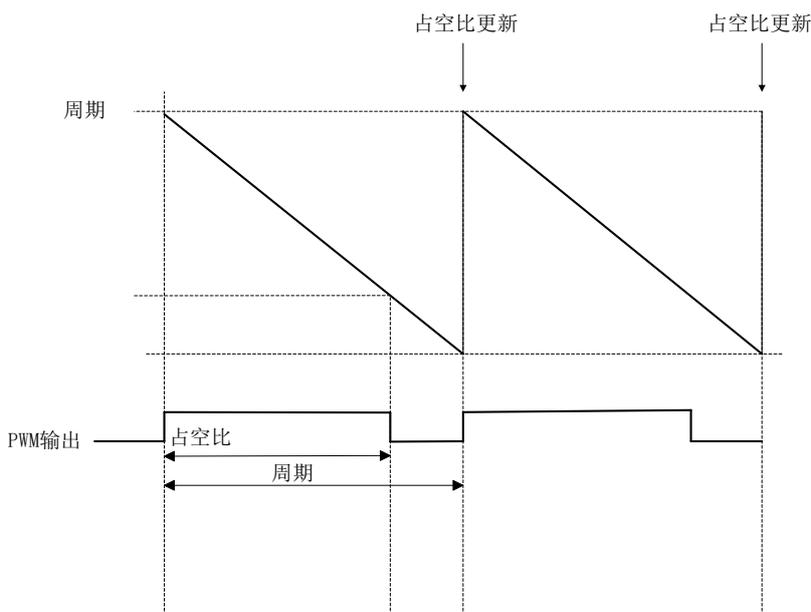
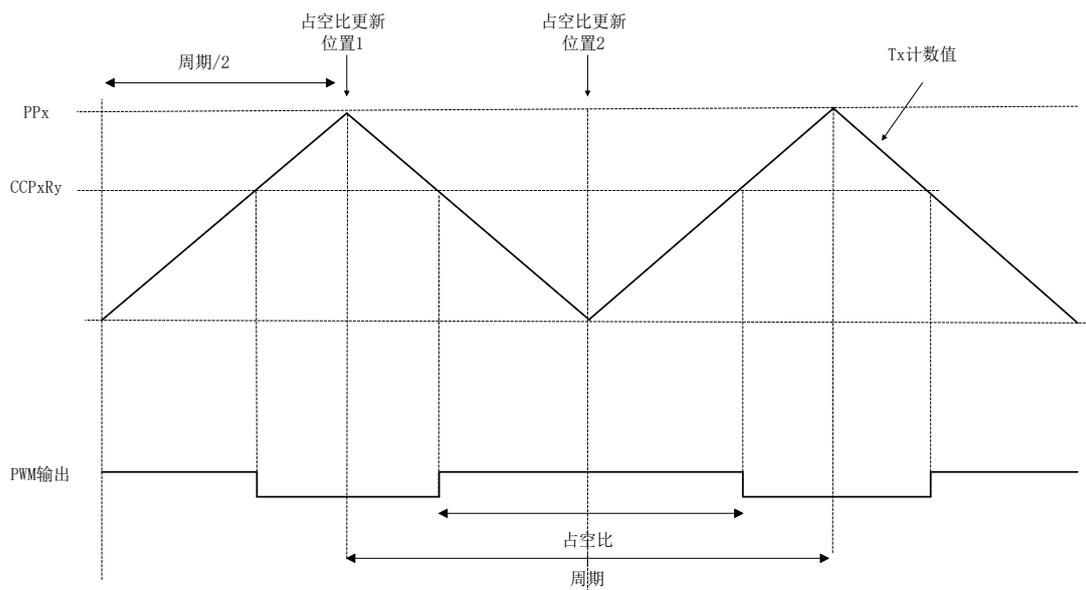


图 14-8 向下计数边沿对齐 PWM

14.6.5 中心对齐 PWM 信号

当 Tx_CTL 控制寄存器中的 TXCMS<2:0>计数模式选择位设置成 x01,x10,x11 时，产生的 PWM 信号为中心对齐的 PWM 信号。在该模式下，PWM 信号的周期由 PPX 周期寄存器决定，此时，周期寄存器中的值表示的是周期的一半。占空比由寄存器 CCPx_Ry 决定。在计数器 Tx 向上计数，计数器 Tx 与占空比匹配时，PWM 输出为低电平；在计数器 Tx 向下计数，计数器 Tx 与占空比匹配时，PWM 输出为高电平。

如果占空比值为 0，则在整個 PWM 输出均为低电平，若占空比值比周期寄存器的值大或者等于周期，则在整個 PWM 输出均为高电平。


图 14-9 中心对齐 PWM

14.6.6 单脉冲输出模式

单脉冲输出模式允许计数器响应一个激励，并产生一个脉宽可程序控制的脉冲。

以下为在 CCPxCH1 引脚产生单脉冲输出的配置：

- (1) Tx_CTL2 寄存器的 TXTS<3:0>=1101、TXSMS<2:0>=100，Tx 在 CCPxCH1 的上升沿开始计数。
- (2) CCPx_CTL3 寄存器的 PXSPM=1，CCPx 模块设为单脉冲输出模式。
- (3) PXOCy<1:0>=00/01：PWM 输出，高有效/低有效。
- (4) 如果 CCPx_CTL3 寄存器的 PXSPMST=0，则单脉冲输出一次后不会关闭定时器使能位，在下次 CCPxCH1 上升沿到来时，依旧会触发定时器开始计数。如果 CCPx_CTL3 寄存器的 PXSPMST=1，则单脉冲输出一次后关闭定时器使能位。

14.6.7 PWM 信号的产生和中断

通过设置 Tx_CTL 控制寄存器中的 TXCMS<2:0>可以选择 Tx 的计数方式，从而产生边沿对齐和中心对齐的 PWM 信号。当 Tx 向上/向下计数时，产生边沿对齐的 PWM 信号；当 Tx 向上向下计数时，产生中心对齐的 PWM 信号。

通过设置 Tx_CTL 寄存器中的 TXCMS<2:0>位控制选择溢出中断，则可以选择在不同的时机触发 PWM 中断操作。当 TXCMS<2:0>=000 时，Tx 为向下计数模式，是下溢时产生中断标志，即 Tx 计数到 0 时产生一个中断信号。TXCMS<2:0>=100 时，Tx 为向上计数模式，是上溢时产生中断标志，也就是在 PPX 和 Tx 相等的时候产生一个中断信号。当 TXCMS<2:0>=x01 时(向上向下计数模式)，是上溢产生中断标志，也就是在计数器 Tx 向上计数到周期值的时候产生一个中断信号。当 TXCMS<2:0>=x10 时(向上向下计数模式)，是下溢产生中断标志，是计数器 Tx 向下递减为 0 的时候产生一个中断信号。当 TXCMS<2:0>=x11 时(向上向下计数模式)，是上溢下溢中断，在这种状态下可以在 PPX 和计数器 Tx 相等以及计数器 Tx 向下计数到 0 时都产生中断信号。在向上向下计数模式时可以使用 3 种中断产生方式。

14.7 比较器的输出清除输出比较/PWM 信号

对于一个给定的通道，设置 CCPx_CTL2 寄存器的 PXOCyCE (y=1,2,3,4) 为“1”，能够用比较器的上升沿高电平将输出比较信号拉低/将 PWM 信号置为无效电平。CCPxCHy 输出信号将保持为低（比较模式）/保持无效信号（PWM 模式），直到比较器输出低电平且同时发生更新事件。

该功能只适用于输出比较及 PWM 模式，不适用于强制输出模式。

14.8 低功耗 CCP (CCP0)

定时器 T0 可作为低功耗定时器 (LP_TIMER) 使用，此时 CCP0 也可以工作于低功耗模式。低功耗的定时器可以在除待机和关断模式以外的模式中工作。当单片机处于停止模式下，内核已经掉电，此时低功耗定时器仍可以保持工作状态。

为了使能定时器/CCP 进入低功耗模式，需要使能 CCP0LPEN 位不让 T0 复位，同时 T0_CTL1 的 TXCLK<1:0>=11 即选用内部低频振荡器 INTLF 作为相应的时钟源并将 CCP0CLKLPEN 置 1 允许内部低频振荡器作为 CCP 工作时钟，同时保持时钟源处于工作状态，这样低功耗定时器就可以在低功耗模式下运行。

低功耗模式下的 IO 口请参考引脚重映射章节“引脚重映射表-低功耗外设功能重映射”。

注：PM_CTL 寄存器位于备份域，需要将 OSC_CTL0 寄存器的 PMWREN 位置 1 以允许备份域寄存器读写。

15 增强型捕捉/比较/PWM 模块(ECCP5/9)

15.1 概述

ECCPx (x=5,9) 模块是增强型捕捉/比较/脉宽调制模块, 可以提供外部信号捕捉、内部比较输出以及 PWM 输出三种功能。在 ECCP 模块中, 采用 16 位的定时器/计数器(ECCP5 为 T5 和 T6, ECCP9 为 T9 和 T10)做为该 ECCP 的计数时基, 在 ECCP5 模块中捕捉寄存器为 16 位的寄存器 ECCPx_Cy (x=5,9;y=1,2,3,4), 比较寄存器为 16 位的寄存器 ECCPx_Ry (x=5,9;y=1,2,3,4), 该寄存器也用于 PWM 模式下的占空比设置。支持部分寄存器的数据更新功能。支持各个通道独立的关断操作。

如下事件发生时产生 DMA:

- 输入捕获
- 输出比较
- 关断事件
- 更新事件

15.2 ECCP 模块结构框图

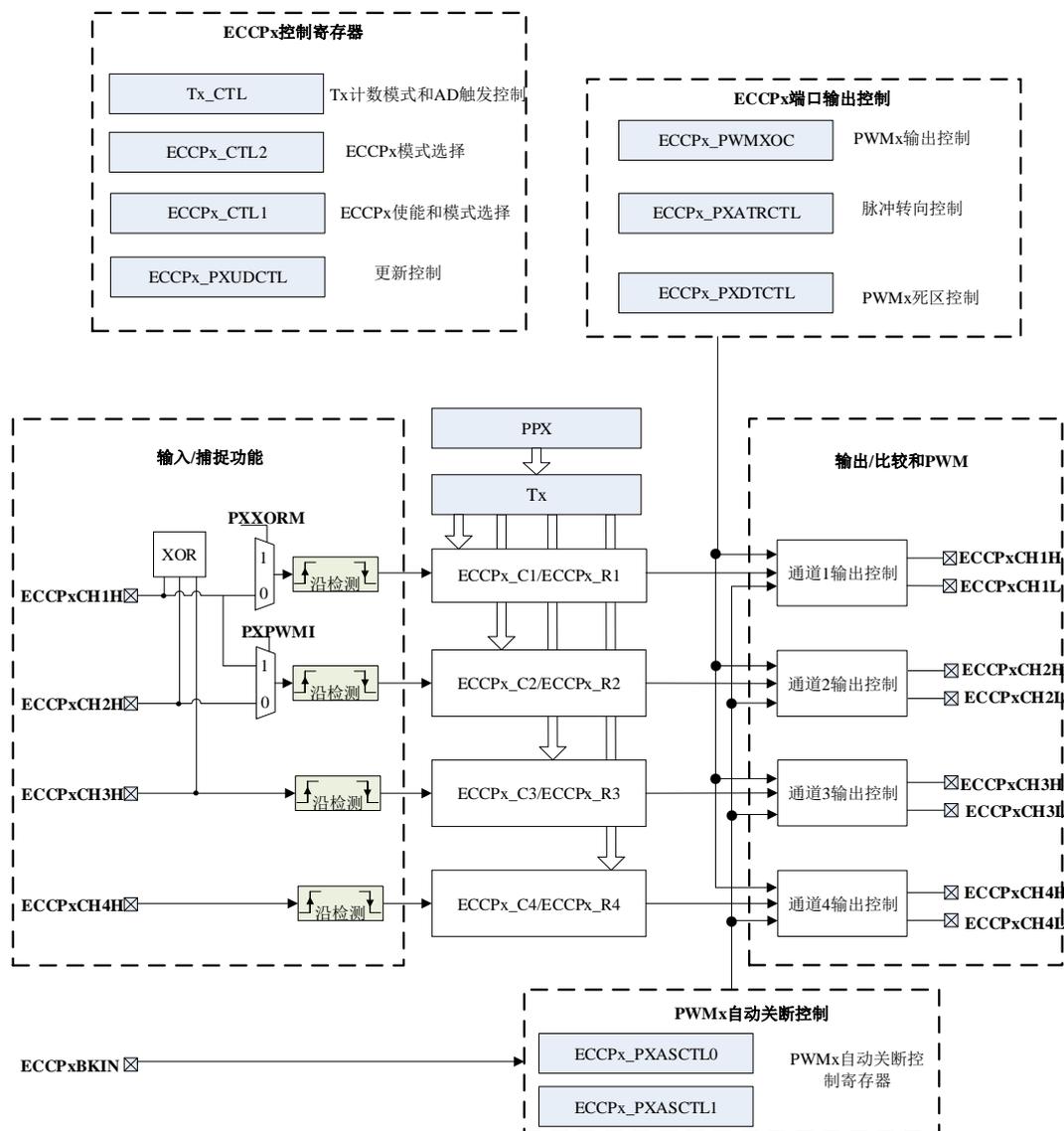


图 15-1 ECCP 模块结构框图

- ECCPxCHyH¹ 引脚: ECCP 模块的通道引脚 H, 通过 IO 口重映射功能选择与具体的 IO 口映射。在捕捉模式下, 从 ECCPxCHyH 引脚输入捕捉源信号去触发捕捉; 在比较模式下, 从 ECCPxCHyH 引脚输出比较事件 (翻转、置 1、置 0 等); 在 PWM 模式下, 从 ECCPxCHyH 引脚输出 PWM 波形。此外, 此引脚也可用作定时器的触发输入 TRGI 和触发输出 TRGO。在功能确定后, 引脚的输入输出方向由硬件自动配置。
- ECCPxCHyL 引脚: ECCP 模块的通道引脚 L, 通过 IO 口重映射功能选择与具体的 IO 口映射。在捕捉模式和比较模式下, ECCPxCHyL 无效; 在 PWM 模式下, 根据不同输出配置, 可以从 ECCPxCHyL 引脚输出不同的 PWM 波形 (与 H 通道同相、反相或者死区延迟)。在功能确定后, 引脚的输入输出方向由硬件自动配置。
- ECCPxBKIN 引脚: ECCP 模块关断引脚, 通过 IO 口映射功能映射到 IO 口。在自动关断/重启模式下, 外部关断信号从该引脚输入。

¹ 本章除特殊说明外, x=5,9, z=6,10, y=1,2,3,4。

15.3 ECCP 模块相关寄存器

表 15-1 ECCP 相关寄存器 1

偏移地址	寄存器	访问	功能描述	复位值
0x000	Tx_CNT	R/W	Tx_CNT 寄存器	0x0000 uuuu
0x004	Tz_CNT	R/W	Tz_CNT 寄存器	0x0000 uuuu
0x008	Tx_PPX	R/W	Tx_PPX 周期寄存器	0x0000 FFFF
0x00C	Tz_PPZ	R/W	Tz_PPZ 周期寄存器	0x0000 FFFF
0x010	Tx_PRSC	R/W	Tx 预分频设置寄存器	0x0000 0000
0x014	Tz_PRSC	R/W	Tz 预分频设置寄存器	0x0000 0000
0x024	Tx_CTL	R/W	Tx 控制寄存器	0x0000 8080
0x028	Tz_CTL	R/W	Tz 控制寄存器	0x0000 8080
0x2C	ECCPx_PDCTL	R/W	位置检测控制寄存器	0x0000 0000
0x30	ECCPx_PXASCTL	R/W	关断控制寄存器	0x0000 0000
0x34	Tx_CCTCTL	R/W	比较器清零定时器 Tx 控制寄存器	0x0000 0000
0x38	Tz_CCTCTL	R/W	比较器清零定时器 Tz 控制寄存器	0x0000 0000
0x5C	Tx_CCRCTL	R/W	CCR 功能控制寄存器	0x0000 0000

定时器 T5/6 基地址：0x4000 0300

定时器 T9/10 基地址：0x4000 0400

表 15-2 ECCP 相关寄存器 2

偏移地址	寄存器	访问	功能描述	复位值
0x000	ECCPx_CTL1	R/W	ECCPx 控制寄存器 1	0x0000 0000
0x004	ECCPx_R1	R/W	比较/PWM 寄存器 1	0x0000 0000
0x008	ECCPx_R2	R/W	比较/PWM 寄存器 2	0x0000 0000
0x00C	ECCPx_R3	R/W	比较/PWM 寄存器 3	0x0000 0000
0x010	ECCPx_R4	R/W	比较/PWM 寄存器 4	0x0000 0000
0x014	ECCPx_PXUDCTL	R/W	更新控制寄存器	0x0000 0000
0x018	ECCPx_CTL2	R/W	ECCPx 控制寄存器 2	0x0000 0000
0x01C	ECCPx_PXDTCTL	R/W	死区控制寄存器	0x0000 0000
0x020	ECCPx_PWMXOC	R/W	PWMx 输出控制寄存器	0x0000 0000
0x024	ECCPx_PXATRCTL	R/W	PWMx 输出配置寄存器	0x0000 0000
0x028	ECCPx_PXASCTL0	R/W	PWMx 关断控制寄存器 0	0x0000 0000
0x02C	ECCPx_PXASCTL1	R/W	PWMx 关断控制寄存器 1	0x0000 0000
0x03C	ECCPx_IE	R/W	ECCPx 中断使能寄存器	0x0000 0000
0x040	ECCPx_EGIF	R/W	ECCPx 中断状态寄存器	0x0000 0000
0x044	Tx_UDTIMX	R/W	Tx 更新计数器	0x0000 0000
0x048	Tz_UDTIMZ	R/W	Tz 更新计数器	0x0000 0000
0x04C	ECCPx_DF	R	触发 DMA 请求标志寄存器	0x0000 0000
0x050	ECCPx_C1	R	捕捉寄存器 1	0x0000 0000
0x054	ECCPx_C2	R	捕捉寄存器 2	0x0000 0000
0x058	ECCPx_C3	R	捕捉寄存器 3	0x0000 0000
0x05C	ECCPx_C4	R	捕捉寄存器 4	0x0000 0000

偏移地址	寄存器	访问	功能描述	复位值
0x064	ECCPx_DE	R/W	触发 DMA 请求使能寄存器	0x0000 0000
0x068	ECCPx_SRIC	R/W	ECCPx 中断清除寄存器	0x0000 0000
0x06C	ECCPx_CTL3	R/W	ECCPx 控制寄存器 3	0x0000 0000

ECCP5 基地址: 0x4000 0380

ECCP9 基地址: 0x4000 0480

15.3.1 ECCPx_CTL1 ECCPx 控制寄存器 1

表 15-3 ECCPx_CTL1 ECCPx 控制寄存器 1

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
R/W																																		
位名																	CH4M<3:0>	CH3M<3:0>	CH2M<3:0>	CH1M<3:0>														

CH4M<3:0>: ECCPxCH4 模式选择位

- 0000 = 该通道捕捉/比较关闭
- 0001 = 未使用 (保留)
- 0010 = 比较模式: 匹配时输出电平翻转 (CC4IF 置 1)
- 0011 = 未使用 (保留)
- 0100 = 捕捉模式: 在每个下降沿发生捕捉 (CC4IF 置 1)
- 0101 = 捕捉模式: 在每个上升沿发生捕捉 (CC4IF 置 1)
- 0110 = 捕捉模式: 每 4 个上升沿发生捕捉 (CC4IF 置 1)
- 0111 = 捕捉模式: 每 16 个上升沿发生捕捉 (CC4IF 置 1)
- 1000 = 比较模式: 比较匹配时高电平 (CC4IF 置 1)
- 1001 = 比较模式: 比较匹配时输出低电平 (CC4IF 置 1)
- 1010 = 比较模式: 比较匹配时产生软件中断 (CCH4IF 置 1, ECCPxCH4H 引脚不受影响)
- 1011 = 比较模式: 触发特殊事件 (CC4IF 位置 1, ECCPx 复位 Tz, 如果 ADC 使能将启动 AD 转换)
- 其它 = 未使用 (保留)

CH3M<3:0>: ECCPxCH3 模式选择位

- 0000 = 该通道捕捉/比较关闭
- 0001 = 未使用 (保留)
- 0010 = 比较模式: 匹配时输出电平翻转 (CC3IF 置 1)
- 0011 = 未使用 (保留)
- 0100 = 捕捉模式: 在每个下降沿发生捕捉 (CC3IF 置 1)
- 0101 = 捕捉模式: 在每个上升沿发生捕捉 (CC3IF 置 1)
- 0110 = 捕捉模式: 每 4 个上升沿发生捕捉 (CC3IF 置 1)
- 0111 = 捕捉模式: 每 16 个上升沿发生捕捉 (CC3IF 置 1)
- 1000 = 比较模式: 比较匹配时输出高电平 (CC3IF 置 1)
- 1001 = 比较模式: 比较匹配时输出低电平 (CC3IF 置 1)
- 1010 = 比较模式: 比较匹配时产生软件中断 (CC3IF 置 1, ECCPxCH3H 引脚)

不受影响)

1011 = 比较模式: 触发特殊事件 (CC3IF 位置 1, ECCPx 复位 Tx, 如果 ADC 使能将启动 AD 转换)

其它 = 未使用 (保留)

CH2M<3:0>: ECCPCH2 模式选择位

0000 = 该通道捕捉/比较关闭

0001 = 未使用 (保留)

0010 = 比较模式: 匹配时输出电平翻转 (CC2IF 置 1)

0011 = 未使用 (保留)

0100 = 捕捉模式: 在每个下降沿发生捕捉 (CC2IF 置 1)

0101 = 捕捉模式: 在每个上升沿发生捕捉 (CC2IF 置 1)

0110 = 捕捉模式: 每 4 个上升沿发生捕捉 (CC2IF 置 1)

0111 = 捕捉模式: 每 16 个上升沿发生捕捉 (CC2IF 置 1)

1000 = 比较模式: 比较匹配时输出高电平 (CC2IF 置 1)

1001 = 比较模式: 比较匹配时输出低电平 (CC2IF 置 1)

1010 = 比较模式: 比较匹配时产生软件中断 (CC2IF 置 1, ECCPxCH2H 引脚不受影响)

1011 = 比较模式: 触发特殊事件 (CC2IF 位置 1, ECCPx 复位 Tx, 如果 ADC 使能将启动 AD 转换)

其它 = 未使用 (保留)

CH1M<3:0>: ECCP 模块的模式选择位

0000 = 该通道捕捉/比较关闭、所有通道 PWM 关闭

0001 = 未使用 (保留)

0010 = 比较模式: 匹配时输出电平翻转 (CC1IF 置 1)

0011 = 未使用 (保留)

0100 = 捕捉模式: 在每个下降沿发生捕捉 (CC1IF 置 1)

0101 = 捕捉模式: 在每个上升沿发生捕捉 (CC1IF 置 1)

0110 = 捕捉模式: 每 4 个上升沿发生捕捉 (CC1IF 置 1)

0111 = 捕捉模式: 每 16 个上升沿发生捕捉 (CC1IF 置 1)

1000 = 比较模式: 比较匹配时输出高电平 (CC1IF 置 1)

1001 = 比较模式: 比较匹配时输出低电平 (CC1IF 置 1)

1010 = 比较模式: 比较匹配时产生软件中断 (CC1IF 置 1, ECCPxCH1H 引脚不受影响)

1011 = 比较模式: 触发特殊事件 (CC1IF 位置 1, ECCPx 复位 Tx, 如果 ADC 使能将启动 AD 转换)

1100 = PWM 自由模式, Tx 作为通道 1/2/3 的时基, Tz 作为通道 4 的时基, Tx 和 Tz 可独立使能生成 PWM 输出。

1101 = PWM 协同模式, Tx 作为通道 1/2/3 的时基, Tz 作为通道 4 的时基, Tx 和 Tz 需要全部使能才能生成 PWM 输出。

1110 = PWM 单时基模式, Tx 作为通道 1/2/3/4 的时基, Tx 使能控制生成 PWM 输出。

注: CH1M 取值为: 1100、1101 和 1110 时对 CH1M、CH2M、CH3M、CH4M 位域同时起作用

15.3.2 ECCPx_Ry 比较/PWM 占空比寄存器

表 15-4 ECCPx_Ry 比较/PWM 占空比寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	ECCPXRY<15:0>																																

ECCPXRY<15:0>: 在比较模式下, 16 位 ECCPx_Ry 寄存器的值将不断与 Tx 寄存器的值相比较。当两者匹配时, ECCP 就会触发相应的事件。

在 PWM 模式下, ECCPx_Ry 寄存器作为 PWMxy 的占空比寄存器。

15.3.3 ECCPx_PXUDCTL 更新控制寄存器

表 15-5 ECCPx_PXUDCTL 更新控制寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																						
R/W																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																								PXMPEN	PXUDEVT3	PXUDEVT2	PXUDEVT1	PXUDEVT0	PXUDEN1	PXUDENO																								

PXMPEN: 相位移动使能位

0 = 关闭 PWM 自动相移功能

1 = 开启 PWM 自动相移功能

PXUDEVT3: 更新事件控制 3 (在 PXUDEN1=1 时有效)

0 = 当定时器 UDTIMZ 为 0 时更新以 Tz 为时基的占空比、周期、预分频、更新计数器寄存器

1 = 产生更新事件, 更新 Tz 为时基的占空比、周期、预分频、更新计数器寄存器, 并将定时器清零

PXUDEVT2: 更新事件控制 2 (在 PXUDEN1=1 时有效)

0 = 当定时器 UDTIMZ 为 0 时更新 Tz 为时基的输出控制寄存器

1 = 立即更新 Tz 为时基的输出控制寄存器

PXUDEVT1: 更新事件控制 1 (在 PXUDEN0=1 时有效)

0 = 当定时器 UDTIMX 为 0 时更新以 Tx 为时基的占空比、周期、预分频、更新计数器寄存器

1 = 产生更新事件, 更新以 Tx 为时基的占空比、周期、预分频、更新计数器寄存器, 并将定时器清零

PXUDEVT0: 更新事件控制 0 (在 PXUDEN0=1 时有效)

0 = 当定时器 UDTIMX 为 0 时更新以 Tx 为时基的输出控制寄存器

1 = 立即更新以 Tx 为时基的输出控制寄存器

PXUDEN1: 更新使能 1 (控制 Tz)

1 = 允许以 Tz 为时基占空比、周期、预分频、更新计数器、输出控制寄存器更新

0 = 禁止以 Tz 为时基占空比、周期、预分频、更新计数器、输出控制寄存器更新

PXUDEN0: 更新使能 0 (控制 Tx)

1 = 允许以 Tx 为时基占空比、周期、预分频、更新计数器、输出控制寄存器更新

0 = 禁止以 Tx 为时基占空比、周期、预分频、更新计数器、输出控制寄存器更新

15.3.4 ECCPx_CTL2 ECCP 控制寄存器 2

表 15-6 ECCPx_CTL2 ECCP 控制寄存器 2

复位值	0	31	0	30	0	29	0	28	0	27	0	26	0	25	0	24	0	23	0	22	0	21	0	20	0	19	0	18	0	17	0	16	0	15	0	14	0	13	0	12	0	11	0	10	0	9	0	8	0	7	0	6	0	5	0	4	0	3	0	2	0	1	0	0																
R/W																																	R/W	0	15	R/W	0	14	R/W	0	13	R/W	0	12	R/W	0	11	R/W	0	10	R/W	0	9	R/W	0	8	R/W	0	7	R/W	0	6	R/W	0	5	R/W	0	4	R/W	0	3	R/W	0	2	R/W	0	1	R/W	0	0
位名																																	TZSMS <2:0>			PXPWMI			PXXORM			PXSPM			TXMSSYNC			TXTS <2:0>			TXSMS <2:0>			TXMMS <2:0>																										

TZSMS<2:0>: Tz 从模式选择位

000 = 从模式禁止

100 = 触发模式, 在触发信号 TRGI 出现上升沿时计数器启动(需使能 TXEN)

101 = 门控模式, 当触发信号 TRGI 为高电平时计数器的计数时钟使能, TRGI 为低电平时停止计数(计数器没有被复位)

110 = 复位模式, 在触发信号 TRGI 出现上升沿时初始化定时器, 并对寄存器进行更新

111 = 计数模式 2, 定时器时钟为触发信号 TRGI

其它 = 未使用(保留)

PXPWMI: PWM 输入测量模式使能位

0 = 未使能

1 = 使能 PWM 输入测量模式

PXXORM: 输入异或使能位

0 = 未使能

1 = 使能输入异或, 将 ECCPxCH1H、ECCPxCH2H、ECCPxCH3H 的输入信号连接到异或门, 异或门输出映射到捕捉/比较通道 1 上。

PXSPM: 单脉冲输出模式

0 = 单脉冲输出模式未使能

1 = 使能单脉冲输出模式

TXMSSYNC: 主从模式同步位

0 = 未使能

1 = 触发输入上的事件将被延迟, 以使主、从定时器同步响应外部触发。

TXTS<2:0>: 输入触发 TRGI 选择位¹

000 = 定时器 1 触发

001 = 定时器 20 触发

¹ 这些位只在 TXSMS=000: 从模式禁止时才可改变。以避免电压转变时发生错误的边沿检测。TX 和 TZ 的触发位选择都根据这一位域配置。

- 010 = 定时器 5 触发
- 011 = 定时器 9 触发
- 100 = ECCPxCH1H 的输入
- 101 = ECCPxCH2H 的输入
- 110 = ECCPxCH3H 的输入
- 111 = TxCK 输入

TXSMS<2:0>: Tx 从模式选择位

- 000 = 从模式禁止
- 100 = 触发模式, 在触发信号 TRGI 出现上升沿时计数器启动(需使能 TXEN)
- 101 = 门控模式, 当触发信号 TRGI 为高时计数器的计数时钟使能, TRGI 为低时停止计数(计数器没有被复位)
- 110 = 复位模式, 在触发信号 TRGI 出现上升沿时初始化定时器, 并对寄存器进行更新
- 111 = 计数模式 2, 定时器时钟为触发信号 TRGI
- 其它 = 未使用(保留)

TXMMS<2:0>: 主模式选择位

- 000 = 未使用(保留)
- 001 = 使能信号 TXEN 作为触发输出 TRGO
- 010 = TXIF 信号作为触发输出 TRGO(定时器溢出时的中断脉冲信号)
- 011 = ECCPx 的 CC1IF 脉冲作为触发输出 TRGO
- 100 = ECCPxCH1H 作为触发输出 TRGO
- 101 = ECCPxCH2H 作为触发输出 TRGO
- 110 = ECCPxCH3H 作为触发输出 TRGO
- 111 = ECCPxCH4H 作为触发输出 TRGO

15.3.5 ECCPx_PXDTCTL 死区控制寄存器

表 15-7 ECCPx_PXDTCTL 死区控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	PXDC<7:0>							PZDC<7:0>									

PXDC<7:0>: Tx(x=5,9)为时基的 PWM 死区延时时间设置位, 用于设置通道 1/2/3 的死区延时时间

PZDC<7:0>: Tz(z=6,10)为时基的 PWM 死区延时时间设置位, 用于设置通道 4 的死区延时的时间

15.3.6 ECCPx_PWMXOC 输出控制寄存器

表 15-8 ECCPx_PWMXOC 输出控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值																																
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
位名																	PXOC4H<1:0>	PXOC4L<1:0>	PXOC3H<1:0>	PXOC3L<1:0>	PXOC2H<1:0>	PXOC2L<1:0>	PXOC1H<1:0>	PXOC1L<1:0>								

PXOC4H<1:0>: 通道 4 中 ECCPxCH4H 端口输出控制

- 00 = PWM 输出, 高有效
- 01 = PWM 输出, 低有效
- 10 = 强制低电平输出
- 11 = 强制高电平输出

PXOC4L<1:0>: 通道 4 中 ECCPxCH4L 端口输出控制

- 00 = PWM 输出, 高有效
- 01 = PWM 输出, 低有效
- 10 = 强制低电平输出
- 11 = 强制高电平输出

PXOC3H<1:0>: 通道 3 中 ECCPxCH3H 端口输出控制

- 00 = PWM 输出, 高有效
- 01 = PWM 输出, 低有效
- 10 = 强制低电平输出
- 11 = 强制高电平输出

PXOC3L<1:0>: 通道 3 中 ECCPxCH3L 端口输出控制

- 00 = PWM 输出, 高有效
- 01 = PWM 输出, 低有效
- 10 = 强制低电平输出
- 11 = 强制高电平输出

PXOC2H<1:0>: 通道 2 中 ECCPxCH2H 端口输出控制

- 00 = PWM 输出, 高有效
- 01 = PWM 输出, 低有效
- 10 = 强制低电平输出
- 11 = 强制高电平输出

PXOC2L<1:0>: 通道 2 中 ECCPxCH2L 端口输出控制

- 00 = PWM 输出, 高有效
- 01 = PWM 输出, 低有效
- 10 = 强制低电平输出
- 11 = 强制高电平输出

PXOC1H<1:0>: 通道 1 中 ECCPxCH1H 端口输出控制

- 00 = PWM 输出, 高有效
- 01 = PWM 输出, 低有效
- 10 = 强制低电平输出
- 11 = 强制高电平输出

PXOC1L<1:0>: 通道 1 中 ECCPxCH1L 端口输出控制

- 00 = PWM 输出, 高有效
- 01 = PWM 输出, 低有效
- 10 = 强制低电平输出
- 11 = 强制高电平输出

15.3.7 ECCPx_PXATRCTL 输出配置寄存器

表 15-9 ECCPx_PXATRCTL 输出配置寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0				
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
R/W																					R/W	R/W	R/W	R/W												
位名																					PXCH4MOD	PXCH3MOD	PXCH2MOD	PXCH1MOD												

PXCH4MOD: PWMx 通道 4 输出配置位

- 0 = 独立输出模式; 默认 ECCPxCH4H 和 ECCPxCH4L 配置为 PWM 同向输出
- 1 = 互补输出模式; ECCPxCH4H 和 ECCPxCH4L 配置为互补输出, 此模式带有死区控制功能

PXCH3MOD: PWMx 通道 3 输出配置位

- 0 = 独立输出模式; 默认 ECCPxCH3H 和 ECCPxCH3L 配置为 PWM 同向输出
- 1 = 互补输出模式; ECCPxCH3H 和 ECCPxCH3L 配置为互补输出, 此模式带有死区控制功能

PXCH2MOD: PWMx 通道 2 输出配置位

- 0 = 独立输出模式; 默认 ECCPxCH2H 和 ECCPxCH2L 配置为 PWM 同向输出
- 1 = 互补输出模式; ECCPxCH2H 和 ECCPxCH2L 配置为互补输出, 此模式带有死区控制功能

PXCH1MOD: PWMx 通道 1 输出配置位

- 0 = 独立输出模式; 默认 ECCPxCH1H 和 ECCPxCH1L 配置为 PWM 同向输出
- 1 = 互补输出模式; ECCPxCH1H 和 ECCPxCH1L 配置为互补输出, 此模式带有死区控制功能

注:

ECCPx_PWMXOC 输出控制寄存器与 ECCPx_PXATRCTL 输出配置寄存器是相互配合工作的。例如, 假设将 4 通道配置成独立输出模式, 但是 4H 输出控制选择高有效, 4L 输出控制选择低有效, 则 4H 与 4L 的输出波形是反相的; 假设将 4 通道配置成互补输出模式, 4H 输出控制选择高有效, 4L 输出控制选择低有效, 4H 与 4L 的输出波形是同相的; 但是将 4H 输出控制选择高有效, 4L 输出控制也选择高有效, 则 4H 与 4L 的输出波形是反相的。即互补输出实质为有效无效状态互补而非高低电平互补。

15. 3. 8 ECCPx_PXASCTL0 关断控制寄存器 0

表 15-10 ECCPx_PXASCTL0 关断控制寄存器 0

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																		
R/W																					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																					PXASE4	PXASE3	PXASE2	PXASE1	PXASS4<1:0>				PXASS3<1:0>				PXASS2<1:0>				PXASS1<1:0>													

PXASE4: 通道 4 自动关闭事件状态位

- 0 = 通道 4 正常工作
- 1 = 发生了关闭事件；通道 4 为关闭状态

PXASE3: 通道 3 自动关闭事件状态位

- 0 = 通道 3 正常工作
- 1 = 发生了关闭事件；通道 3 为关闭状态

PXASE2: 通道 2 自动关闭事件状态位

- 0 = 通道 2 正常工作
- 1 = 发生了关闭事件；通道 2 为关闭状态

PXASE1: 通道 1 自动关闭事件状态位

- 0 = 通道 1 正常工作
- 1 = 发生了关闭事件；通道 1 为关闭状态

PXASS4<1:0>: 通道 4 自动关闭源选择位

- 00 = 禁止自动关断
 - 01 = 比较器 2/3 输出高电平
 - 10 = ECCPxBKIN 引脚上的低电平
 - 11 = ECCPxBKIN 引脚上的低电平和比较器 2/3 输出高电平都可以关断
- 注：通过 ECCPx_PXASCTL 寄存器的 PASS 具体选择 CMP2 或者 CMP3。

PXASS3<1:0>: 通道 3 自动关闭源选择位

- 00 = 禁止自动关断
- 01 = 比较器 3 输出高电平
- 10 = ECCPxBKIN 引脚上的低电平
- 11 = ECCPxBKIN 引脚上的低电平和比较器 3 输出高电平都可以关断

PXASS2<1:0>: 通道 2 自动关闭源选择位

- 00 = 禁止自动关断
- 01 = 比较器 3 输出高电平
- 10 = ECCPxBKIN 引脚上的低电平
- 11 = ECCPxBKIN 引脚上的低电平和比较器 3 输出高电平都可以关断

PXASS1<1:0>: 通道 1 自动关闭源选择位

- 00 = 禁止自动关断
- 01 = 比较器 3 输出高电平
- 10 = ECCPxBKIN 引脚上的低电平
- 11 = ECCPxBKIN 引脚上的低电平和比较器 3 输出高电平都可以关断

15. 3. 9 ECCPx_PXASCTL1 关断控制寄存器 1

表 15-11 ECCPx_PXASCTL1 关断控制寄存器 1

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值																																
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
位名																	PXSS4H<1:0>	PXSS4L<1:0>	PXSS3H<1:0>	PXSS3L<1:0>	PXSS2H<1:0>	PXSS2L<1:0>	PXSS1H<1:0>	PXSS1L<1:0>								

PXSS4H<1:0>: 引脚 ECCPxCH4H 关闭状态控制位

- 00 = 驱动引脚 ECCPxCH4H 为低电平
- 01 = 驱动引脚 ECCPxCH4H 为高电平
- 1x = 驱动引脚 ECCPxCH4H 为三态

PXSS4L<1:0>: 引脚 ECCPxCH4L 关闭状态控制位

- 00 = 驱动引脚 ECCPxCH4L 为低电平
- 01 = 驱动引脚 ECCPxCH4L 为高电平
- 1x = 驱动引脚 ECCPxCH4L 为三态

PXSS3H<1:0>: 引脚 ECCPxCH3H 关闭状态控制位

- 00 = 驱动引脚 ECCPxCH3H 为低电平
- 01 = 驱动引脚 ECCPxCH3H 为高电平
- 1x = 驱动引脚 ECCPxCH3H 为三态

PXSS3L<1:0>: 引脚 ECCPxCH3L 关闭状态控制位

- 00 = 驱动引脚 ECCPxCH3L 为低电平
- 01 = 驱动引脚 ECCPxCH3L 为高电平
- 1x = 驱动引脚 ECCPxCH3L 为三态

PXSS2H<1:0>: 引脚 ECCPxCH2H 关闭状态控制位

- 00 = 驱动引脚 ECCPxCH2H 为低电平
- 01 = 驱动引脚 ECCPxCH2H 为高电平
- 1x = 驱动引脚 ECCPxCH2H 为三态

PXSS2L<1:0>: 引脚 ECCPxCH2L 关闭状态控制位

- 00 = 驱动引脚 ECCPxCH2L 为低电平
- 01 = 驱动引脚 ECCPxCH2L 为高电平
- 1x = 驱动引脚 ECCPxCH2L 为三态

PXSS1H<1:0>: 引脚 ECCPxCH1H 关闭状态控制位

- 00 = 驱动引脚 ECCPxCH1H 为低电平
- 01 = 驱动引脚 ECCPxCH1H 为高电平
- 1x = 驱动引脚 ECCPxCH1H 为三态

PXSS1L<1:0>: 引脚 ECCPxCH1L 关闭状态控制位

- 00 = 驱动引脚 ECCPxCH1L 为低电平
- 01 = 驱动引脚 ECCPxCH1L 为高电平
- 1x = 驱动引脚 ECCPxCH1L 为三态

15. 3. 10 ECCPx_IE ECCPx 中断使能寄存器

表 15-12 ECCPx_IE ECCPx 中断使能寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名																							PDIE	CPIE	TXUIE	TZUIE	TXIE	TZIE	TXTIE	CC4IE	CC3IE	CC2IE	CC1IE

- PDIE:** 位置检测事件中断使能位
 0 = 禁止位置检测事件中断
 1 = 允许位置检测事件中断
- CPIE:** RDA 和 CPD 比较中断使能位
 0 = 禁止 RDA 和 CPD 比较中断
 1 = 允许 RDA 和 CPD 比较中断
- TXUIE:** Tx 更新事件中断使能位
 0 = 禁止 Tx 更新中断
 1 = 允许 Tx 更新中断
- TZUIE:** Tz 更新事件中断使能位
 0 = 禁止 Tz 更新中断
 1 = 允许 Tz 更新中断
- TXIE:** Tx 计数溢出中断使能位
 0 = 禁止 Tx 溢出中断
 1 = 允许 Tx 溢出中断
- TZIE:** Tz 计数溢出中断使能位
 0 = 禁止 Tx 溢出中断
 1 = 允许 Tx 溢出中断
- TXTIE:** Tx 触发事件中断使能位
 0 = 禁止 Tx 触发中断
 1 = 允许 Tx 触发中断
- CC4IE:** ECCPx 通道 4 捕捉/比较中断使能位
 0 = 禁止 ECCPx 通道 4 捕捉/比较中断
 1 = 允许 ECCPx 通道 4 捕捉/比较中断
- CC3IE:** ECCPx 通道 3 捕捉/比较中断使能位
 0 = 禁止 ECCPx 通道 3 捕捉/比较中断
 1 = 允许 ECCPx 通道 3 捕捉/比较中断
- CC2IE:** ECCPx 通道 2 捕捉/比较中断使能位
 0 = 禁止 ECCPx 通道 2 捕捉/比较中断
 1 = 允许 ECCPx 通道 2 捕捉/比较中断
- CC1IE:** ECCPx 通道 1 捕捉/比较中断使能位
 0 = 禁止 ECCPx 通道 1 捕捉/比较中断
 1 = 允许 ECCPx 通道 1 捕捉/比较中断

15. 3. 11 ECCPx_EGIF ECCPx 中断状态/事件产生寄存器

表 15-13 ECCPx_EGIF ECCPx 中断状态/事件产生寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0															
复位值																	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R													
R/W																																															
位名																	PDIF	TXUIF	TZUIF	TXIF	TZIF	TXTIF	CC4IF	CC3IF	CC2IF	CC1IF	TXTRG	CC4G	CC3G	CC2G	CC1G	CPIF															

PDIF: 位置检测事件中断标志

0 = 未发生位置检测事件中断

1 = 发生位置检测事件中断

TXUIF: Tx 更新事件中断标志

0 = 未发生 Tx 更新中断

1 = 发生 Tx 更新中断

TZUIF: Tz 更新事件中断标志

0 = 未发生 Tz 更新中断

1 = 发生 Tz 更新中断

TXIF: Tx 溢出中断标志

0 = 未发生 Tx 溢出中断

1 = 发生 Tx 溢出中断

TZIF: Tz 溢出中断标志

0 = 未发生 Tz 溢出中断

1 = 发生 Tz 溢出中断

TXTIF: Tx 触发事件中断标志

0 = 未发生 Tx 触发中断

1 = 发生 Tx 触发中断

CC4IF: ECCPx 通道 4 捕捉/比较中断标志

0 = ECCPx 通道 4 无捕获/比较事件发生

1 = ECCPx 通道 4 发生捕获/比较事件

CC3IF: ECCPx 通道 3 捕捉/比较中断标志

0 = ECCPx 通道 3 无捕获/比较事件发生

1 = ECCPx 通道 3 发生捕获/比较事件

CC2IF: ECCPx 通道 2 捕捉/比较中断标志

0 = ECCPx 通道 2 无捕获/比较事件发生

1 = ECCPx 通道 2 发生捕获/比较事件

CC1IF: ECCPx 通道 1 捕捉/比较中断标志

0 = ECCPx 通道 1 无捕获/比较事件发生

1 = ECCPx 通道 1 发生捕获/比较事件

TXTRG: 产生触发事件（该位由软件置‘1’，用于产生一个触发事件，定时器停止计数后由硬件自动清‘0’。）

0 = 无动作

1 = ECCPxIF 寄存器中的 TXTIF=1,若使能相应的中断及 DMA,则产生相应的中断及 DMA

CC4G: 产生通道 4 捕捉/比较事件 (该位由软件置‘1’, 由硬件自动清‘0’。)

- 0 = 无动作
- 1 = 在通道 4 上产生一个捕捉/比较事件。
- (详细描述参考 CC1G)

CC3G: 产生通道 3 捕捉/比较事件 (该位由软件置‘1’, 由硬件自动清‘0’。)

- 0 = 无动作
- 1 = 在通道 3 上产生一个捕捉/比较事件。
- (详细描述参考 CC1G)

CC2G: 产生通道 2 捕捉/比较事件 (该位由软件置‘1’, 由硬件自动清‘0’。)

- 0 = 无动作
- 1 = 在通道 2 上产生一个捕捉/比较事件。
- (详细描述参考 CC1G)

CC1G: 产生通道 1 捕捉/比较事件 (该位由软件置‘1’, 由硬件自动清‘0’。)

- 0 = 无动作
- 1 = 在通道 1 上产生一个捕捉/比较事件。
- 若通道 1 被配置为比较模式: 置 CC1IF=1, 若使能相应的中断及 DMA, 则产生相应的中断及 DMA。
- 若通道 1 被配置为捕捉模式: 当前的计数值被捕捉至 ECCPx_C1 寄存器, 且置 CC1IF=1, 若使能相应的中断及 DMA, 则产生相应的中断及 DMA

CPIF: RDA 和 CPD 比较中断标志位

- 0 = 未发生 RDA 和 CPD 匹配事件中断
- 1 = 发生 RDA 和 CPD 匹配事件中断

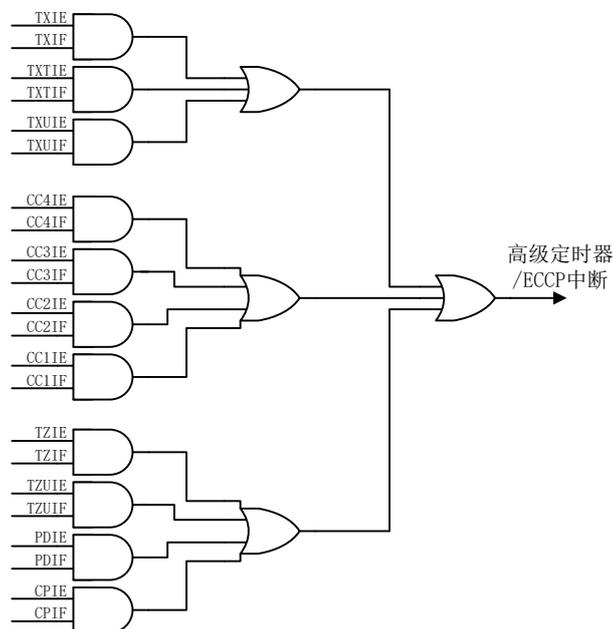


图 15-2 高级定时器/ECCP 中断框图

15. 3. 12 Tx_UDTIMX/Tz_UDTIMZ 更新计数器

表 15-14 Tx_UDTIMX 更新计数器

复位值	0	31	0	30	0	29	0	28	0	27	0	26	0	25	0	24	0	23	0	22	0	21	0	20	0	19	0	18	0	17	0	16	0	15	0	14	0	13	0	12	0	11	0	10	0	9	0	8	0	7	0	6	0	5	0	4	0	3	0	2	0	1	0	0
R/W																																	R/W	0	7	R/W	0	6	R/W	0	5	R/W	0	4	R/W	0	3	R/W	0	2	R/W	0	1	R/W	0	0								
位名																																	UDCNT<7:0>																															

UDCNT<7:0>: Tx 更新计数器的值

表 15-15 Tz_UDTIMZ 更新计数器

复位值	0	31	0	30	0	29	0	28	0	27	0	26	0	25	0	24	0	23	0	22	0	21	0	20	0	19	0	18	0	17	0	16	0	15	0	14	0	13	0	12	0	11	0	10	0	9	0	8	0	7	0	6	0	5	0	4	0	3	0	2	0	1	0	0
R/W																																	R/W	0	7	R/W	0	6	R/W	0	5	R/W	0	4	R/W	0	3	R/W	0	2	R/W	0	1	R/W	0	0								
位名																																	UDCNT<7:0>																															

UDCNT<7:0>: Tz 更新计数器的值

当使能 Tx/Tz 的更新功能时,若 ECCPx_PXUDCTL 寄存器中的 PXUDEVT0=0, 则只有在计数值匹配 Tx_UDTIMX/Tz_UDTIMZ 的值时才会更新 Tx 为时基的输出控制、输出配置寄存器,同理 PXUDEVT2 位。若 ECCPx_PXUDCTL 寄存器中的 PXUDEVT1=0, 则只有在 Tx_UDTIMX/Tz_UDTIMZ 的值向下计数到 0 时才会更新 Tx 为时基的占空比、周期、预分频寄存器到缓冲器中,同理 PXUDEVT3 位。

15. 3. 13 ECCPx_DF 触发 DMA 请求标志寄存器

表 15-16 ECCPx_DF 触发 DMA 请求标志寄存器

复位值	0	31	0	30	0	29	0	28	0	27	0	26	0	25	0	24	0	23	0	22	0	21	0	20	0	19	0	18	0	17	0	16	0	15	0	14	0	13	0	12	0	11	0	10	0	9	0	8	0	7	0	6	0	5	0	4	0	3	0	2	0	1	0	0
R/W																																	R	0	7	R	0	6	R	0	5	R	0	4	R	0	3	R	0	2	R	0	1	R	0	0								
位名																																	TXUDF	TZUDF	TRIGDF	CLSDF	CH4DF	CH3DF	CH2DF	CH1DF																								

TXUDF: Tx 更新事件 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

- 0 = 未发生 Tx 更新 DMA 请求
- 1 = 发生 Tx 更新 DMA 请求

TZUDF: Tz 更新事件 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

- 0 = 未发生 Tz 更新 DMA 请求
- 1 = 发生 Tz 更新 DMA 请求

TRIGDF: Tx 触发事件 DMA 请求标志 (DMA 响应后该位由硬件自动清零)

- 0 = 未发生 Tx 触发 DMA 请求
- 1 = 发生 Tx 触发 DMA 请求

- CLSDF:** 关断事件触发 DMA 标志 (DMA 响应后该位由硬件自动清零)
 0 = 未发生关断事件触发 DMA 标志
 1 = 发生了关断事件, 且 CCPxASDE 使能
- CH4DF:** ECCPx 通道 4 捕捉/比较 DMA 请求标志 (DMA 响应后该位由硬件自动清零)
 0 = ECCPx 通道 4 无捕获/比较事件触发 DMA 发生
 1 = ECCPx 通道 4 发生捕获/比较事件触发 DMA
- CH3DF:** ECCPx 通道 3 捕捉/比较 DMA 请求标志 (DMA 响应后该位由硬件自动清零)
 0 = ECCPx 通道 3 无捕获/比较事件触发 DMA 发生
 1 = ECCPx 通道 3 发生捕获/比较事件触发 DMA
- CH2DF:** ECCPx 通道 2 捕捉/比较 DMA 请求标志 (DMA 响应后该位由硬件自动清零)
 0 = ECCPx 通道 2 无捕获/比较事件触发 DMA 发生
 1 = ECCPx 通道 2 发生捕获/比较事件触发 DMA
- CH1DF:** ECCPx 通道 1 捕捉/比较 DMA 请求标志 (DMA 响应后该位由硬件自动清零)
 0 = ECCPx 通道 1 无捕获/比较事件触发 DMA 发生
 1 = ECCPx 通道 1 发生捕获/比较事件触发 DMA

15. 3. 14 ECCPx_Cy ECCPx 捕捉寄存器

表 15-17 ECCPx_Cy ECCPx 捕捉寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
R/W																	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位名																	ECCPXCy<15:0>																

ECCPXCy<15:0>: 在捕捉模式下时, 当对应的 ECCPxCHy 引脚发生事件时, ECCPx_Cy 寄存器捕捉 Tx_CNT 寄存器的 16 位值。

15. 3. 15 ECCPx_DE ECCPx 请求 DMA 使能寄存器

表 15-18 ECCPx_DE ECCPx 请求 DMA 使能寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																						
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																						
R/W																									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
位名																									TXUDE	TZUDE	TXTDE	CLSDE	CH4DE	CH3DE	CH2DE	CH1DE																						

- TXUDE:** Tx 更新事件的 DMA 请求使能位
 0 = 禁止更新事件的 DMA 请求
 1 = 允许更新事件的 DMA 请求
- TZUDE:** Tz 更新事件的 DMA 请求使能位
 0 = 禁止更新事件的 DMA 请求
 1 = 允许更新事件的 DMA 请求
- TXTDE:** 触发事件的 DMA 请求使能位

- 0 = 禁止触发事件的 DMA 请求
- 1 = 允许触发事件的 DMA 请求
- CLSDDE: 关断事件的 DMA 请求使能位
 - 0 = 禁止关断事件的 DMA 请求
 - 1 = 允许关断事件的 DMA 请求
- CH4DE: 捕获/比较 4 的 DMA 请求使能位
 - 0 = 禁止捕获/比较 4 的 DMA 请求
 - 1 = 允许捕获/比较 4 的 DMA 请求
- CH3DE: 允许捕获/比较 3 的 DMA 请求使能位
 - 0 = 禁止捕获/比较 3 的 DMA 请求
 - 1 = 允许捕获/比较 3 的 DMA 请求
- CH2DE: 允许捕获/比较 2 的 DMA 请求使能位
 - 0 = 禁止捕获/比较 2 的 DMA 请求
 - 1 = 允许捕获/比较 2 的 DMA 请求
- CH1DE: 允许捕获/比较 1 的 DMA 请求使能位
 - 0 = 禁止捕获/比较 1 的 DMA 请求
 - 1 = 允许捕获/比较 1 的 DMA 请求

15. 3. 16 ECCPx_SRIC ECCPx 中断标志清除寄存器

表 15-19 ECCPx_SRIC ECCPx 中断标志清除寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																			
R/W																						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																						PDIC	CPIC	TXUIC	TZUIC	TXIC	TZIC	TX TIC	CC4IC	CC3IC	CC2IC	CC1IC																			

- PDIC: 位置检测事件中断标志清除位
 - 0 = 不清除位置检测事件中断标志
 - 1 = 清除位置检测事件中断标志
- CPIC: RDA 和 CPD 比较中断标志清除位
 - 0 = 不清除 RDA 和 CPD 比较中断标志
 - 1 = 清除 RDA 和 CPD 比较中断标志
- TXUIC: Tx 更新事件中断标志清除位
 - 0 = 不清除 Tx 更新事件中断标志
 - 1 = 清除 Tx 更新事件中断标志
- TZUIC: Tz 更新事件中断标志清除位
 - 0 = 不清除 Tz 更新事件中断标志
 - 1 = 清除 Tz 更新事件中断标志
- TXIC: Tx 溢出中断标志清除位
 - 0 = 不清除 Tx 溢出中断标志
 - 1 = 清除 Tx 溢出中断标志
- TZIC: Tz 溢出中断标志清除位
 - 0 = 不清除 Tz 溢出中断标志

- 1 = 清除 Tz 溢出中断标志
- TXTIC: Tx 触发事件中断标志清除位
 - 0 = 不清除 Tx 触发事件中断标志
 - 1 = 清除 Tx 触发事件中断标志
- CC4IC: ECCPx 通道 4 捕捉/比较中断标志清除位
 - 0 = 不清除 ECCPx 通道 4 捕捉/比较中断标志
 - 1 = 清除 ECCPx 通道 4 捕捉/比较中断标志
- CC3IC: ECCPx 通道 3 捕捉/比较中断标志清除位
 - 0 = 不清除 ECCPx 通道 3 捕捉/比较中断标志
 - 1 = 清除 ECCPx 通道 3 捕捉/比较中断标志
- CC2IC: ECCPx 通道 2 捕捉/比较中断标志清除位
 - 0 = 不清除 ECCPx 通道 2 捕捉/比较中断标志
 - 1 = 清除 ECCPx 通道 2 捕捉/比较中断标志
- CC1IC: ECCPx 通道 1 捕捉/比较中断标志清除位
 - 0 = 不清除 ECCPx 通道 1 捕捉/比较中断标志
 - 1 = 清除 ECCPx 通道 1 捕捉/比较中断标志

15. 3. 17 ECCPx_CTL3 ECCP 控制寄存器 3

表 15-20 ECCPx_CTL3 ECCP 控制寄存器 3

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W																																
位名																	PXRSEN	TUNITE	PXSPMST													

- PXRSEN: PWMx 通道 1/2/3/4 重启使能位
 - 0 = 自动关闭时, PXASE 由软件清零, 以重启 PWM
 - 1 = 自动关闭时, 一旦关闭事件消失, PXASE 位将自动清零, PWM 自动重启
- TUNITE: Tx、Tz 联立使能位
 - 0 = 不联立 Tx、Tz 计数
 - 1 = 将 Tx 和 Tz 联合成一个 32 位计数器
- PXSPMST: 单脉冲输出模式选择
 - 0 = 单脉冲输出后不关闭定时器使能位
 - 1 = 单脉冲输出后关闭定时器使能位

15. 3. 18 ECCPx_PDCTL ECCP 位置检测控制寄存器

表 15-21 ECCPx_PDCTL ECCP 位置检测控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R																		R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位名																			CPD <2:0>			RDA <2:0>			SMPEN		TI2_EN	TI1_EN	TI0EN	CMPE	CPE <1:0>				

CPD<2:0>: TI<2:0>信号状态位, 该位与 RDA<2:0>信号相等时, CPIF 位置 1

RDA<2:0>: 与位置检测输入信号 TI<2:0>比较位, CMPE 置 1 有效

SMPEN: 输入采样功能使能位

0 = 禁止采样功能

1 = 使能采样功能

TI2_EN: TI2 输入使能位

0 = 禁止 TI2 输入检测

1 = 使能 TI2 输入检测

TI1_EN: TI1 输入使能位

0 = 禁止 TI1 输入检测

1 = 使能 TI1 输入检测

TI0_EN: TI0 输入使能位

0 = 禁止 TI0 输入检测

1 = 使能 TI0 输入检测

CMPE: 位置检测比较使能位

0 = 禁止位置检测比较

PDIF 根据 TIx 信号有效沿触发

1 = 使能位置检测比较

PDIF 根据 TIx 信号有效沿, 以及 TIx 和 RDA 的比较结果触发

CPE<1:0>: TIx 信号输入检测沿选择位

00 = 禁止检测功能

01 = 上升沿检测

10 = 下降沿检测

01 = 上升沿和下降沿均检测

15. 3. 19 ECCPx_PXASCTL 关断控制寄存器

表 15-22 ECCPx_PXASCTL 关断控制寄存器

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

PXASS: 比较器关断源选择位, ECCPx_PXASCTL0 选择比较器输出高电平作为关断源时有

效

0 = 选择比较器 3 的输出高电平作为关断源

1 = 选择比较器 2 的输出高电平作为关断源

15.4 捕捉模式

ECCPx 模块可在 ECCPxCH1H、ECCPxCH2H、ECCPxCH3H、ECCPxCH4H 这 4 个通道上发生捕捉。各个通道在捕捉模式下，当对应的 ECCPxCHyH 引脚发生事件时，捕捉 Tx/Tz 寄存器的 16 位值存入 ECCPx_Ry 寄存器。捕捉模式时，通道 1/2/3 以 Tx 为时基，通道 4 以 Tz 为时基。

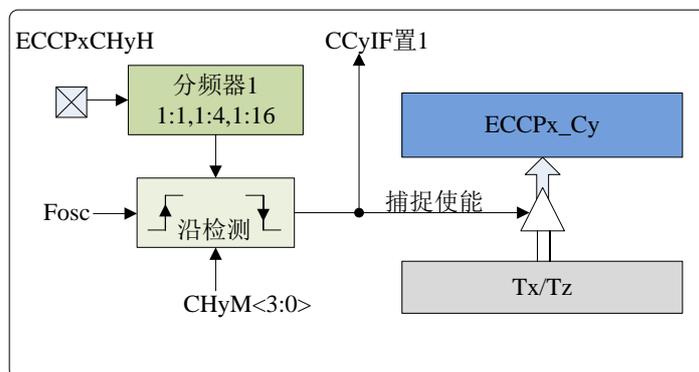


图 15-3 捕捉模式原理框图

触发捕捉的事件可被定义为以下四者之一，并且各通道的捕捉模式由 ECCPx_CTL1 寄存器中的模式选择位 CHyM<3:0>配置：

- 0100 = 每个下降沿
- 0101 = 每个上升沿
- 0110 = 每 4 个上升沿
- 0111 = 每 16 个上升沿

当一个捕捉发生时，ECCPx_EGIF 寄存器中的 CCyIF 置 1，它必须用 ECCPx_SRIC 寄存器的 CCyIC 位置 1 后才会清零。如果在 ECCPx_Cy 寄存器中的值被读取之前发生另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该保持 ECCPx_IE 寄存器中的 CCyIE 中断允许位清零以避免产生误中断。在操作模式发生任何改变之后也应清零 ECCPx_EGIF 寄存器中的中断标志位 CCyIF。

ECCPx_CTL1 寄存器中的 CHyM<3:0>位指定了 4 种预分频器设置（对应的是 1,4,16 个上升沿发生捕捉）。每当关闭 ECCP 模块或禁止捕捉模式时，就会清零预分频器计数器。这意味着任何复位都将清零预分频计数器。

虽然从一种捕捉预分频比切换到另一种捕捉预分频比不会将预分频计数器清零，但可能会产生误中断。因此要避免出现这种不期望的操作，应在改变预分频比前通过将 ECCPx_CTL1 寄存器清零关闭该模块。

15.4.1 PWM 测量模式

该模式是捕捉模式的一个特例，除下列区别外，操作与输入捕捉模式相同：

设置 ECCPx_CTL2 寄存器的 PXPWMI=1，选择 PWM 测量模式。（将通道 1 的 PWM 输入信号映射到通道 1 和通道 2 的输入。）

ECCPx_CTL2 寄存器里的 TXSMS<2:0>设为 110：复位模式，在触发信号 TRGI 出现上升沿时初始化定时器，并对寄存器进行更新。

ECCPx_CTL2 寄存器里的 TXTS<2:0>设为 100：ECCPxCH1 的输入作为触发输入（TRGI）

- (1) 设置 CH1M<3:0>=0100/0101 在通道 1 的每个下降/上升沿发生捕捉；CH2M

$\langle 3:0 \rangle = 0101/0100$ 在通道2的每个上升/下降沿发生捕捉。

通过读取通道 1/2 的捕捉寄存器 ECCPx_C1/2 可以得到输入信号的高低电平的脉宽。

注：如果配置通道1上升沿捕捉，通道2下降沿捕捉，则ECCPx_C1寄存器捕捉到的是被测信号的周期值，ECCPx_C2寄存器捕捉到的是被测信号的正脉宽；如果配置通道1下降沿捕捉，通道2上升沿捕捉，则ECCPx_C1寄存器捕捉到的是被测信号的正脉宽，ECCPx_C2寄存器捕捉到的是被测信号的周期值；

15.4.2 模块相关引脚说明

ECCPxCH1H/L、ECCPxCH2H/L、ECCPxCH3H/L、ECCPxCH4H/L 全部可作为 PWM 输出，其中 ECCPxCH1H、ECCPxCH2H、ECCPxCH3H、ECCPxCH4H 可作为捕捉输入、比较输出以及可供主从模式中触发输入 TRGI 或者触发输出 TRGO 选择。在选择引脚用于某种功能后，其输入输出方向由硬件自动配置。（初始情况下都为输出）

15.4.3 与霍尔传感器的接口

使用增强型 ECCP 产生 PWM 信号驱动马达时，可以用另一个通用定时器作为“接口定时器”来连接霍尔传感器。具体配置见章节“PWM 异或输入以及霍尔传感器接口功能”。

15.4.4 捕捉中断与 DMA 请求

当一个捕捉发生时，ECCPx_EGIF 寄存器中的 CCyIF 置 1，它必须用 ECCPx_SRIC 寄存器的 CCyIC 位置 1 后才会清零，CCyIC 置 1 后需要软件清零以保证后续中断标志位能被置 1。若 ECCPx_IE 寄存器中对应的 CCyIE 位为 1，则中断模块的 INT_EIF1 寄存器中对应的中断标志位置 1。

ECCP 捕捉触发 DMA 的使用详细参考高级定时/计数器章节“Tx/Tz 触发 DMA(高级定时器)”部分。

15.5 比较模式

15.5.1 比较功能

ECCPxCH1H、ECCPxCH2H、ECCPxCH3H、ECCPxCH4H 这 4 个通道均可设为比较模式。比较模式时，通道 1/2/3 以 Tx 为时基，通道 4 以 Tz 为时基。

在比较模式下，16 位 ECCPx_Ry 寄存器的值将不断与 Tx/Tz 寄存器的值相比较。当两者匹配时，ECCP 模块可能会出现以下几种情况：

- ECCPxCHyH 引脚的输出电平翻转
- ECCPxCHyH 引脚输出高电平
- ECCPxCHyH 引脚输出低电平
- 产生软件中断触发信号
- 产生特殊事件触发信号

ECCPxCHyH 引脚的动作取决于 ECCPx_CTL1 寄存器中 CHyM $\langle 3:0 \rangle$ 控制位的值。

所有比较模式都会产生中断。原理图如下图所示：

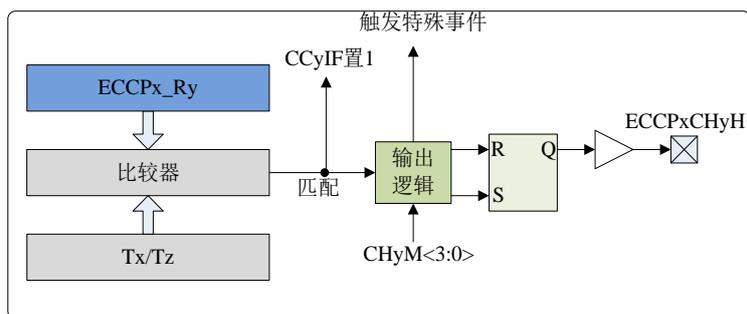


图 15-4 比较模式原理框图

当选择输出电平翻转模式 ($CHyM<3:0>=0010$) 时, 比较匹配时, ECCPxCHyH 引脚的输出电平翻转, 并将 CCyIF 置 1。

当选择普通比较模式时 ($CHyM<3:0>=1000$ 或 1001) 时, 比较匹配时, ECCPxCHyH 引脚输出高电平或者低电平, 并且将 CCyIF 置 1。

当选择了软件中断触发模式时 ($CHyM<3:0>=1010$) 时, 比较匹配时, 将 ECCyIF 置 1, 但是 ECCP 模块不会控制 CHyH 引脚。

当选择了特殊事件触发模式 ($CHyM<3:0>=1011$) 时, 比较匹配时, ECCPx 会立即产生特殊事件触发输出, 将 CCyIF 置 1, 但是 ECCPx 模块不会控制 ECCPxCHyH 引脚。此时如果 ADC 已使能, 将启动 AD 转换。但 Tx/Tz 寄存器不会立即复位, 直到 Tx/Tz 计数脉冲的下一个上升沿才复位。从而使 ECCPx_Ry 寄存器实际上成为了 Tx/Tz 的 16 位可编程周期寄存器。

15.5.2 比较中断与 DMA 请求

在比较模式下, 当 ECCPx_Ry 的值与 Tx/Tz 寄存器的值相等时, ECCPx_EGIF 寄存器中的 CCyIF 置 1, 它必须用 ECCPx_SRIC 寄存器的 CCyIC 位置 1 后才会清零, CCyIC 置 1 后需要软件清零以保证后续中断标志位能被置 1。若 ECCPx_IE 寄存器中对应的 CCyIE 位为 1, 则中断模块的 INT_EIF1 寄存器中对应的中断标志位置 1。

ECCP 比较触发 DMA 的使用详细参考高级定时/计数器章节“Tx/Tz 触发 DMA(高级定时器)”部分。

15.6 PWM 模式

ECCP5 的 PWM 模式为增强型 PWM 模式。共有 4 个通道, 可在 8 个不同的引脚输出 PWM 信号, 分辨率最高 16 位。增强型 PWM 模式的通道 $y(y=1,2,3,4)$ 原理框图如下图所示:

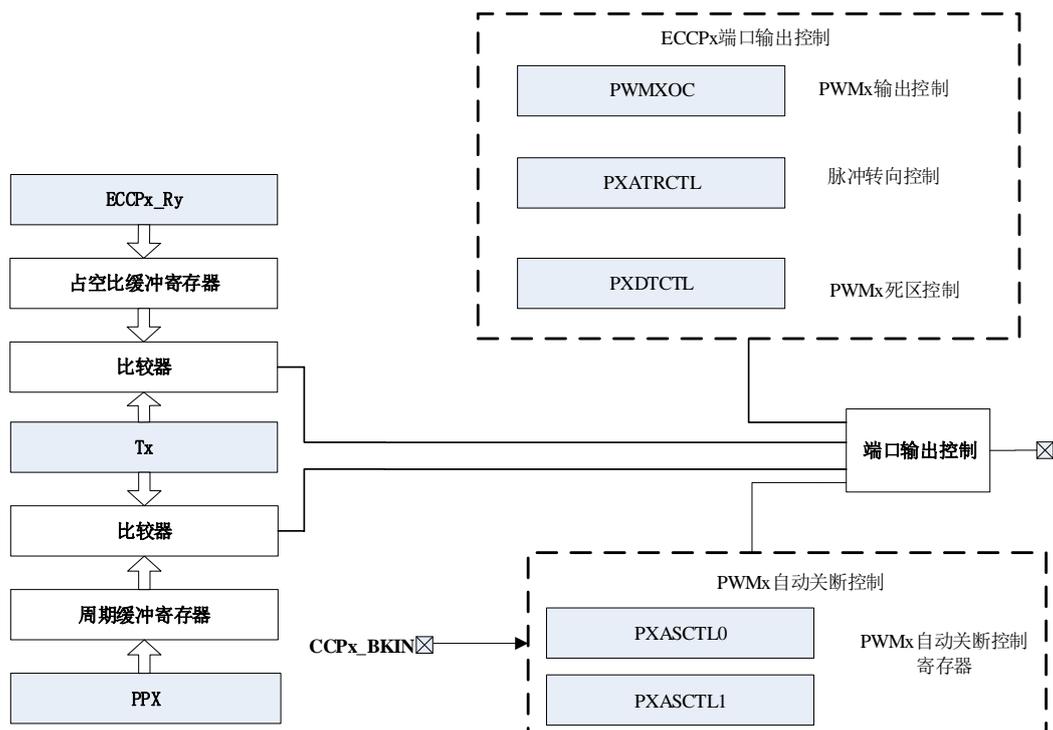


图 15-5 PWM 通道原理框图

通过 ECCPx_CTL1 控制寄存器的 CHyM<3:0>可以使能或禁止 PWMx 的通道功能。在 PWM 自由模式和 PWM 协同模式下，通道 1/2/3 共用 Tx 作为计数时基，通道 4 用 Tz 作为计数时基；在 PWM 单时基模式下，通道 1/2/3/4 共用 Tx 作为计数时基。

每个 PWMx 通道均带有一个死区控制功能，其中通过 ECCPx_PXDTCTL 寄存器可以设置通道 1/2/3/4 的死区时间。每个通道均有独立的死区定时器。

每个 PWMx 通道的占空比设置寄存器均为 16 位 ECCPx_Ry。定时器 Tx 可通过 PPX 设置脉冲周期，定时器 Tz 可通过 PPZ 设置脉冲周期。

每个 PWMx 通道最多可分别在 2 个不同的引脚输出 PWM 信号，其中，通道 4 输出引脚为 ECCPxCH4H/ECCPxCH4L，通道 3 输出引脚为 ECCPxCH3H/ECCPxCH3L，通道 2 输出引脚为 ECCPxCH2H/ECCPxCH2L，通道 1 输出引脚为 ECCPxCH1H/ECCPxCH1L。

每个 PWMx 通道均有 2 种输出模式：独立输出模式和互补输出模式，如表“PWMx 通道输出模式控制及有效引脚”所示。通过 ECCPx_PXATRCTL 中的 PxCHyMOD 位选择相应的输出模式。互补输出模式带有死区控制功能。

通过 ECCPx_PWMXOC 输出控制寄存器可以控制 PWM 通道输出端口是输出 PWM 信号还是强制输出模式。若输出的是 PWM 信号，其输出极性是高有效还是低有效；若为强制输出模式，输出电平是高电平还是低电平。

表 15-23 PWMx 通道输出模式控制及有效引脚

PWMx 输出模式	PxCHyMOD	有效引脚
独立输出模式	0	PWMx 通道 1/2/3/4 最多可在 2 个引脚输出 PWM 信号，不带死区控制
互补输出模式	1	每个通道输出配置为互补输出；互补输出模式带有死区控制功能

注：各种输出模式的详细介绍请参考本节相应部分。

15.6.1 PWMx 的周期、占空比及分辨率

15.6.1.1 PWMx 周期

PWMx 以 Tx/Tz 作为计数时基, Tx/Tz 的周期通过 16 位的周期寄存器 PPX 或 PPZ 进行设置, 其值均可设置为 0~65535。当 PWMx 为自由模式或协同模式时, PWMx 通道 1/2/3 使用 Tx 作为时基, 同时通道 4 使用 Tz 作为时基; 当 PWMx 为单时基模式时, 通道 1/2/3/4 均使用 Tx 作为计数时基。

在边沿对齐模式下和中心对齐模式下, 周期的计算方式不同, 中心对齐模式下得到的周期是周期寄存器设置值的两倍。PWMx 边沿对齐和中心对齐的周期分别通过下式进行计算。

边沿对齐下周期的计算公式:

$$\text{PWM周期} = (\text{PPX} + 1) \cdot \text{Tsysclk} \cdot (\text{Tx 预分频比}) \quad \text{公式 15-1}$$

$$\text{PWM周期} = (\text{PPZ} + 1) \cdot \text{Tsysclk} \cdot (\text{Tz 预分频比}) \quad \text{公式 15-2}$$

中心对齐下周期的计算公式:

$$\text{PWM周期} = \text{PPX} \cdot \text{Tsysclk} \cdot \text{Tx 预分频比} \cdot 2 \quad \text{公式 15-3}$$

$$\text{PWM周期} = \text{PPZ} \cdot \text{Tsysclk} \cdot \text{Tz 预分频比} \cdot 2 \quad \text{公式 15-4}$$

注: T_{sysclk} 为工作时钟频率的倒数

周期寄存器为双缓冲模式, 只有在 PXUDEN0/1 为 1 时才能更新。其中 PXUDEN0 控制以 Tx 为时基的周期更新, PXUDEN1 控制以 Tz 为时基的周期更新。

当 PXUDEN0/1 为 1 时, 通过 PXUDEVTx (x=1, 3) 可以控制周期更新时机。其中 PXUDEN0 控制以 Tx 为时基的周期更新, PXUDEN1 控制以 Tz 为时基的周期更新。

若 PXUDEVT1/3 为 0, 则周期寄存器可以在新的计数周期开始时更新到周期缓冲器中; 若 PXUDEVT1/3 为 1 时, 则立即更新周期到周期缓冲器中, 且 Tx 被清零。在下次立即更新前, 需要软件清零 PXUDEVT1/3, 再置 1 才能开启新一次的立即更新。

注意: 当 PWM 使能, Tx/Tz 启动时周期寄存器立即更新。

15.6.1.2 PWMx 占空比

PWMx 有 4 个 PWM 通道, 每个 PWM 通道均有 1 个占空比设置寄存器, 每个占空比设置寄存器均为 16 位, 通过 ECCPx_Ry 进行设置。

在不同模式下占空比计算方式不一致。在边沿对齐模式和中心对齐模式下, 脉冲宽度和占空比通过下式计算。

边沿对齐下脉冲宽度和占空比的计算公式:

$$\text{脉冲宽度} = \text{CCPx_Ry} \cdot \text{Tsysclk} \cdot (\text{Tx/Tz 预分频比}) \quad \text{公式 15-5}$$

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{\text{CCPx_Ry}}{\text{PPX} + 1} \quad \text{公式 15-6}$$

注: (1) T_{sysclk} 为工作时钟频率的倒数

中心对齐下脉冲宽度和占空比的计算公式:

$$\text{脉冲宽度} = \text{CCPx_Ry} \cdot \text{Tsysclk} \cdot (\text{Tx/Tz 预分频比}) \cdot 2 \quad \text{公式 15-7}$$

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{\text{CCPx_Ry}}{\text{PPX}} \quad \text{公式 15-8}$$

注: (1) T_{sysclk} 为工作时钟频率的倒数

由上述公式可知，占空比为 0%和 100%满足条件如下：

- 边沿对齐模式：
 - 0%：占空比寄存器为 0
 - 100%：占空比寄存器为大于等于 (PPX+1)
- 中心对齐模式：
 - 0%：占空比寄存器为 0
 - 100%：占空比寄存器为大于等于 PPX
- 在互补模式下，占空比为 0%或者 100%时，只有在边界 PWM 变化时插入死区，否则不插入死区。

占空比寄存器为双缓冲模式，只有在 PXUDEN0/1 为 1 时才能更新。其中 PXUDEN0 控制以 Tx 为时基的周期，占空比更新，PXUDEN1 控制以 Tz 为时基的周期占空比更新。

当 PXUDEN0/1 为 1 时，通过 PXUDEVTx (x=1,3) 可以控制周期，占空比更新时机。其中 PXUDEN0 控制以 Tx 为时基的周期，占空比更新，PXUDEN1 控制以 Tz 为时基的周期，占空比更新。

若 PXUDEVT1/3 为 0，则占空比可以在新的计数周期开始时更新到占空比缓冲器中；若 PXUDEVT1/3 为 1 时，则立即更新占空比到占空比缓冲器中，且 Tx 被清零。在下次立即更新前，需要软件清零 PXUDEVT1/3，再置 1 才能开启新一次的立即更新。

(注意：当 PWM 使能，Tx/Tz 启动时占空比寄存器立即更新。)

15.6.1.3 PWMx 分辨率

分辨率的计算公式如下式所示。

$$\text{分辨率}(\%) = F_{PWM} / F_{Txclk} \times 100\% \quad \text{公式 15-9}$$

$$\text{分辨率}(\text{bits}) = \text{Log}_2(T_{PWM} / T_{Txclk}) \quad \text{公式 15-10}$$

注：Txclk 为 Tsysclk 预分频后的计数时钟

15.6.2 边沿对齐 PWM 信号

当 Tx_CTL 控制寄存器中的 TXCMS<2:0>计数方式选择位设置成 000/100 时，产生的 PWM 信号为边沿对齐的 PWM 信号。在该模式下，PWM 信号的周期由 PPX 周期寄存器决定，占空比由 16 位的寄存器 CCPx_Ry 决定。

下图所示为边沿对齐模式下的 PWM 输出示意图。当 PWM 输出为高有效时，周期开始时输出高电平，占空比与定时器匹配时电平变低；当 PWM 输出低有效时，周期开始时输出低电平，占空比与定时器匹配电平变高。

在向上计数方式下，如果占空比寄存器设置成 0，那么在整个 PWMx 周期都输出无效电平。如果占空比寄存器设置大于 PPX 周期寄存器的值，那么在整个 PWM 周期都输出有效电平。

在向下计数方式下，如果占空比寄存器设置等于周期值，那么在整个 PWMx 周期都输出无效电平。如果占空比寄存器设置大于 PPX 周期寄存器的值，那么在整个 PWM 周期都输出有效电平。

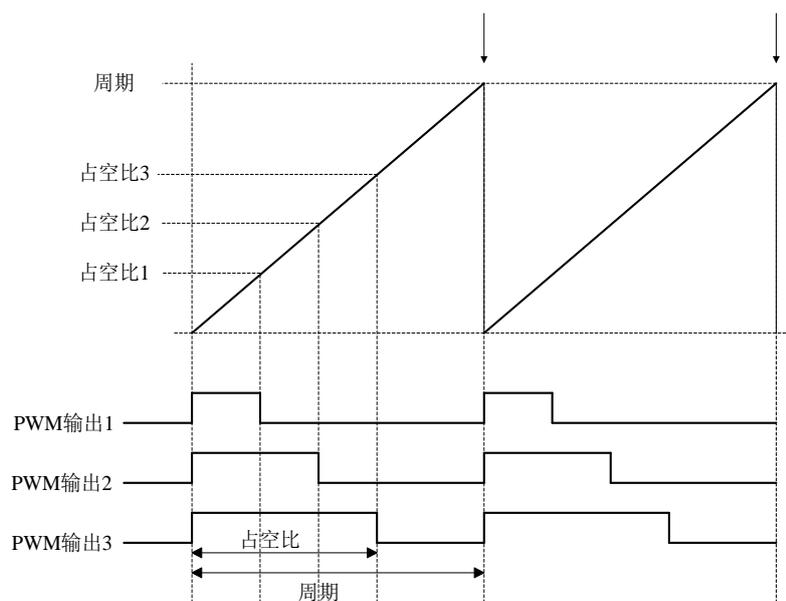


图 15-6 ECCP 向上计数 PWM 输出

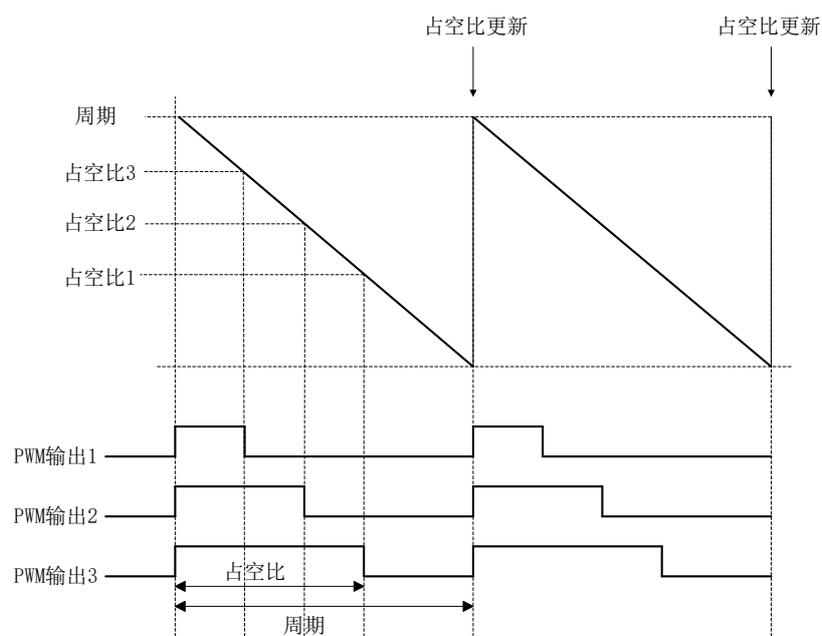


图 15-7 ECCP 向下计数 PWM 输出

15.6.3 中心对齐 PWM 信号

当 Tx_CTL 控制寄存器中的 TXCMS<2:0>计数方式选择位设置成 x01,x10,x11 时,产生的 PWM 信号为中心对齐的 PWM 信号。在该模式下, PWM 信号的周期由 PPX 周期寄存器决定,但此时,周期寄存器中的值表示的是周期的一半。占空比也是由 16 位的寄存器 ECCPx_Ry 决定。在占空比定时器匹配时, PWM 输出翻转。

下图所示为中心对齐模式下的 PWM 输出示意图。当 PWM 输出为高有效时,周期开始时输出高电平,占空比与定时器匹配时电平翻转;当 PWM 输出低有效时,周期开始时输出低电平,占空比与定时器匹配电平翻转。

如果占空比值为 0,则在整个 PWM 输出均为低电平,若占空比值比周期寄存器的值大

或者等于周期，则在整个 PWM 输出均为高电平。

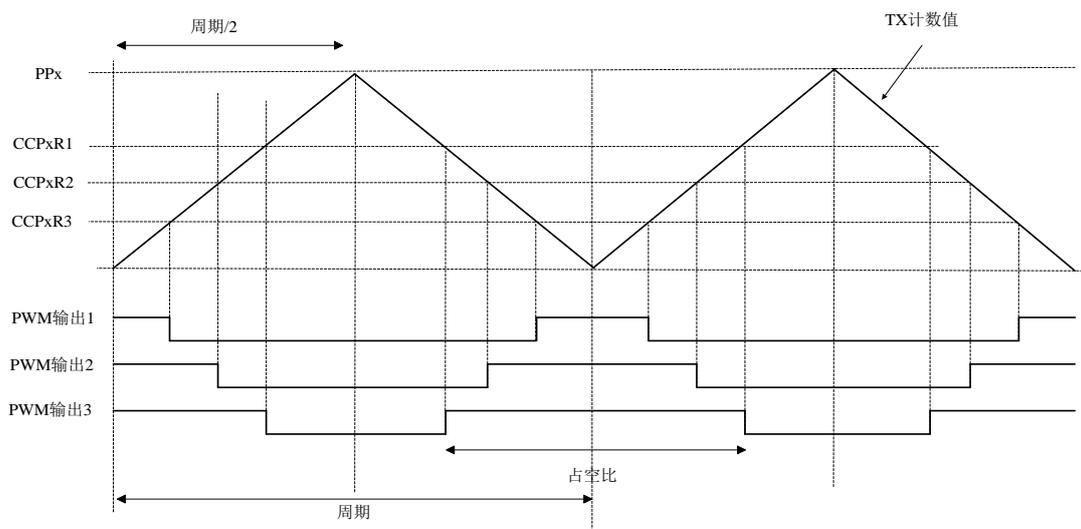


图 15-8 ECCP 中心对齐模式 PWM 输出

15.6.4 PWM 信号产生和中断

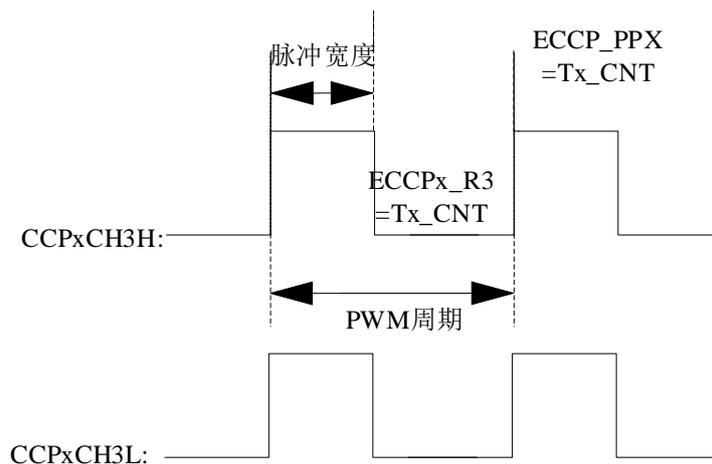
通过设置 Tx_CTL 控制寄存器中的 TXCMS<2:0>可以选择 Tx 的计数方式，从而产生边沿对齐和中心对齐的 PWM 信号。当 Tx 向上/向下计数时，产生边沿对齐的 PWM 信号；当 Tx 向上向下计数时，产生中心对齐的 PWM 信号。

通过设置 Tx_CTL 寄存器中的 TXCMS<2:0>位控制选择溢出中断，则可以选择在不同的时机触发 PWM 中断操作。当 TXCMS<2:0>=000 时，Tx 为向下计数方式，是下溢时产生中断标志，即 Tx 计数到 0 时产生一个中断信号。当 TXCMS<2:0>=100 时(向上计数方式)，是上溢时产生中断标志，也就是在 PPX 和 Tx 相等的时候产生一个中断信号。当 TXCMS<2:0>=x01 时(向上向下计数方式)，是上溢产生中断标志，也就是在计数器 Tx 向上计数到周期值的时候产生一个中断信号。当 TXCMS<2:0>=x10 时(向上向下计数方式)，是下溢产生中断标志，是计数器 Tx 向下递减为 0 的时候产生一个中断信号。当 TXCMS<2:0>=x11 时(向上向下计数方式)，是上溢下溢中断，在这种状态下可以在 PPX 和计数器 Tx 相等以及计数器 Tx 向下计数到 0 时都产生中断信号。在向上计数方式时，使用上溢中断；在向下计数方式时，使用下溢中断；在向上向下计数方式时可以使用 3 种中断产生方式。

15.6.5 独立输出模式

通过将寄存器 ECCPx_PXATRCTL 中的 PXCHyMOD(y=1,2,3,4)位中相应位设置为 0，可以选择独立输出模式。

PWMx 使用独立输出模式时，每个通道最多可同时在 2 个引脚输出同一个 PWM 信号。如下图所示，在独立输出模式下，将 ECCPxCH3H 和 ECCPxCH3L 两路同时设置为 PWM 输出的示例，其它设置情况与此类似。



(a)通道3两路输出有效

图 15-9 PWMx 输出示例

15.6.6 互补输出模式

通过将寄存器 ECCPx_PXATRCTL 的 PXCHyMOD 相应位设置为 1, 可以把 PWMx 相应通道设置为互补输出模式。在此模式下, 每个通道的两个输出为互补输出, 来驱动推挽式负载, 如下图所示为通道 1 带死区控制的互补输出示意图。

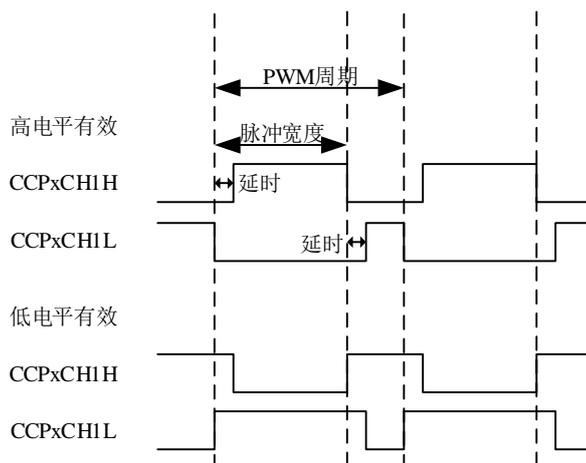


图 15-10 ECCP 互补输出模式示例

同理也可以把通道 2/3/4 设置成互补输出模式。互补输出模式可用于控制互补和全桥控制电路, 如图下图所示, 为通道 1 互补输出模式应用于互补桥和全桥控制电路的示例。互补输出模式应用于两个开关管的互补控制电路, 或使用 2 个 PWM 信号来控制 4 个开关管的全桥控制电路。其他通道实现原理相同。

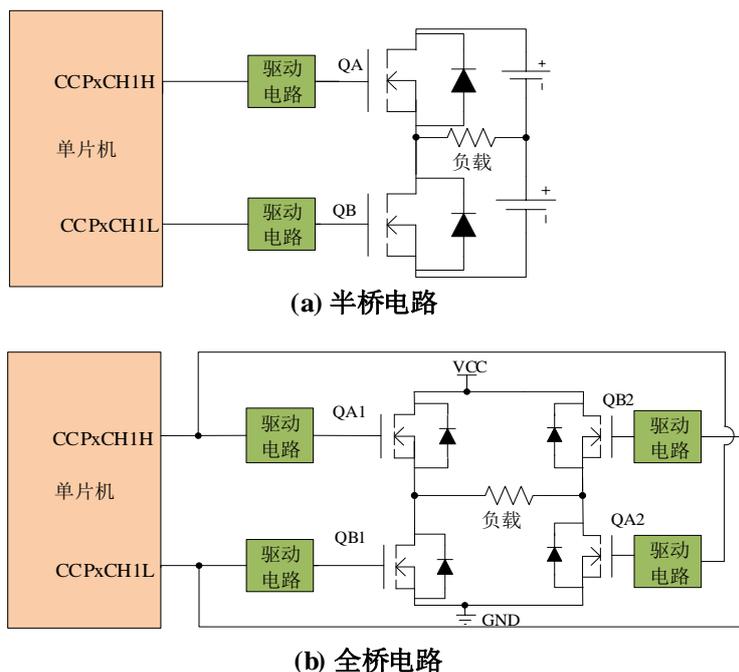


图 15-11 ECCP 互补输出模式应用举例

互补输出模式具有可编程的死区延时功能,可用于产生对外部电路的开关管等元件的导通和截止的时间差异,用来防止在互补驱动电路中产生直通电流,损坏相关电路。ECCPx_PXDTCTL 寄存器中 PXDC<6:0>位的值用来设置以 Tx 为时基的 PWM 通道的死区延时时间,ECCPx_PXDTCTL 寄存器中 PZDC<6:0>位的值用来设置以 Tz 为时基的 PWM 通道的死区延时时间。如果该值大于脉冲宽度,在整个周期内对应的输出将保持无效。

15.6.7 单脉冲输出模式

单脉冲输出模式允许计数器响应一个激励,并产生一个脉宽可程序控制的脉冲。

单脉冲输出模式可以近似看成一个触发模式和一个单脉冲输出模式的结合。触发信号上升沿决定脉冲起始时间,也就是时延时间;PPX 和 ECCPx_Ry 的值共同决定脉冲宽度。

以下是在 ECCPxCH1H 引脚产生单脉冲输出的配置示例:

- (1) 将 Tx 设置为触发模式,触发模式中的上升沿触发。Tx 在 ECCPxCH1H 的上沿开始计数
- (2) 将 ECCPx_CTL2 寄存器中的 PXSPM 位置 1,使能单脉冲输出模式
- (3) 设置 ECCPx_PWMXOC 寄存器的 PXOC1H<1:0>=00/01: PWM 输出高有效或者低有效
- (4) 将寄存器 ECCPx_PXATRCTL 的 PXCH1MOD 位置 0,选择独立输出模式
- (5) 寄存器 ECCPx_CTL1 设置自由输出模式

如果 ECCPx_CTL3 寄存器的 PXSPMST=0,则单脉冲输出一次后不会关闭定时器使能位,在下一次 ECCPxCH1H 上升沿到来时,依旧会触发定时器开始计数。如果 ECCPx_CTL3 寄存器的 PXSPMST=1,则单脉冲输出一次后关闭定时器使能位。

15.6.8 死区延时

在半桥模式输出应用中,ECCPxCH1H 和 ECCPxCH1L 一直以 PWM 频率调制两个开关管,通常开关管的截止比导通需要更多的时间。如果 QA 和 QB 同时导通,两个管子可能会

在一段很短的时间内都处于导通状态，在这很短的时间内，将会产生很大的电流流过两个管子，从而可能导致电路损坏。直到一个管子完全截止才会退出此状态。

为了避免开关期间产生这种具有破坏性的直通电流，可使其中一个管子关闭后再打开另一个管子。在互补输出模式下，使用一个可编程死区延时模块，来避免产生的直通电流破坏电路。如下图所示，该延时在 PWMx 信号从非有效电平到有效电平转换时发生。延时时间通过寄存器 ECCPx_PXDTCTL 进行设置。延时时间计数公式如下式所示。

$$\text{延时时间} = \text{PXDC} \langle 6:0 \rangle \times T \quad \text{公式 15-11}$$

注：T 为与通道计数时基相同的时钟。

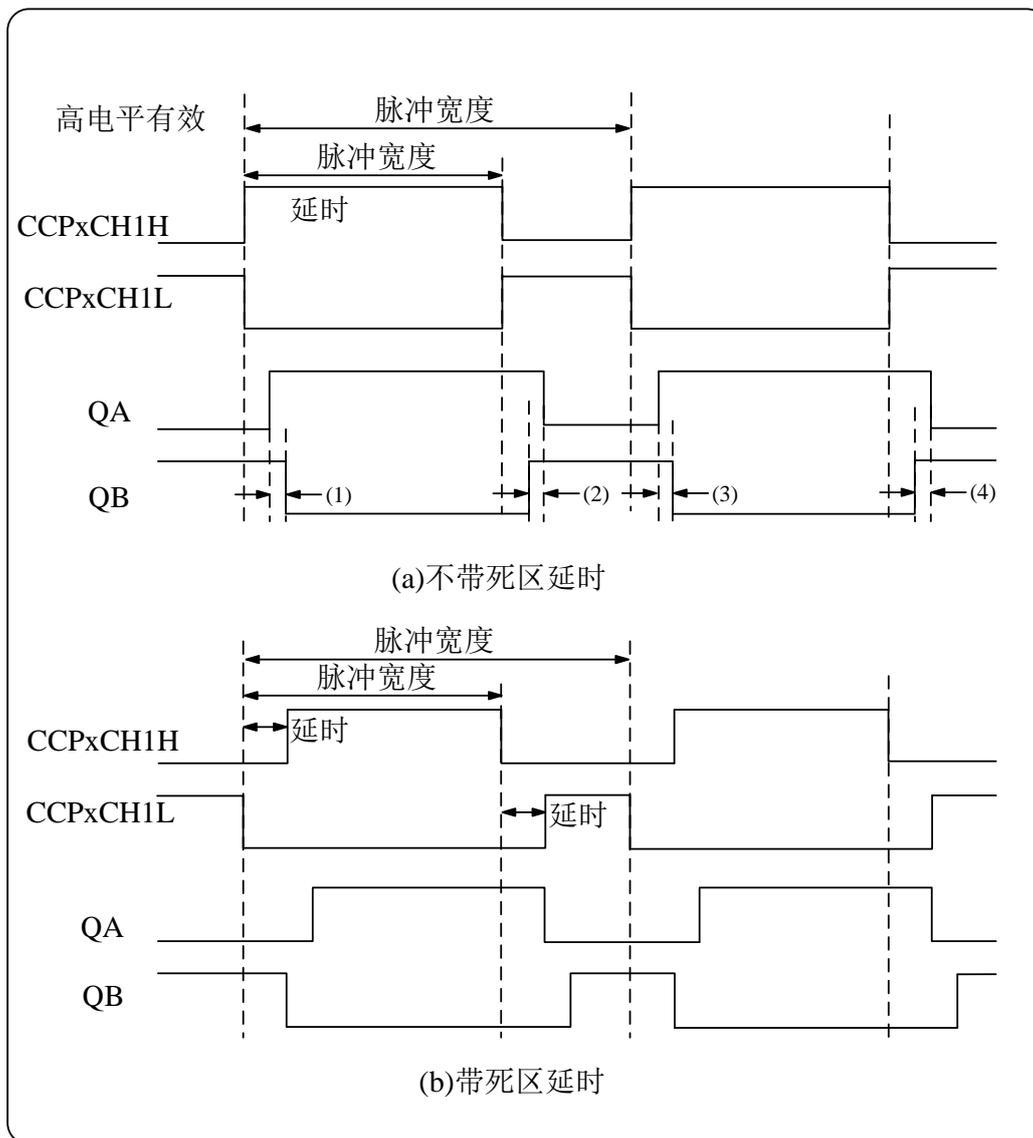


图 15-12 ECCP 死区插入模式示例

在互补模式下，当死区设置为 0 时，死区无效；当占空比为 0%或占 100%时，死区插入与是取决于是否发生了输出变化。若切换前电平和切换后输出电平无变化时不插入死区，若切换前输出电平和切换后输出电平发生变化时，则在切换时插入死区。

(注：切换前后的电平指不改变极性控制条件下的波形。)

更新设置时，若 PXUDEN0/1 为 1，且 PXUDEVT1/3=1 时，相应的死区定时器被清零。

15.6.9 PWMx 输出控制

通过设置 ECCPx_PWMXOC 可以控制 PWMx 通道的输出为 PWM 输出或是强制输出。若设置为高有效时，输出信号不受影响，若设置为低有效时，输出信号为反相信号；当设置 PWM 输出为强制模式时，PWM 输出不受占空比和极性控制的影响。

在独立输出模式下，若通道 1 需要两个输出同相，则控制 ECCPx_PWMXOC 中 PXOC1L<1:0>和 PXOC1H<1:0>分别为 00/00 或 01/01；在互补输出模式下，若通道 1 需要两个输出反相，则控制 ECCPx_PWMXOC 中 PXOC1L<1:0>和 PXOC1H<1:0>分别为 00/01 或 01/00。其它通道设置相同。

通过控制更新控制寄存器 ECCPx_PXUDCTL 可以控制 ECCPx_PWMXOC 的更新。只有 PXUDEN0/1 为 1 才能更新 ECCPx_PWMXOC 寄存器。通过 PXUDEVT0/2 可以控制 ECCPx_PWMXOC 寄存器的更新时机，详见 PWM 更新锁定。

15.6.10 PWM 更新锁定

通过设置 ECCPx_PXUDCTL 更新控制寄存器的 PXUDEN0/1 可以使能或禁止占空比、周期、预分频、更新计数器和输出控制的更新。其中 PXUDEN0/PXUDEVT0/PXUDEVT1 用于控制以 Tx 为时基的通道的占空比、周期、预分频、更新计数器和输出控制寄存器，PXUDEN1/PXUDEVT2/PXUDEVT3 用于控制以 Tz 为时基的通道的占空比、周期、预分频、更新计数器和输出控制寄存器。

当 ECCPx_PXUDCTL 寄存器的 PXUDEN0/1 为 0 时，占空比、周期、预分频、更新计数器和输出控制不能更新，这使得用户能够在新值生效前将所需要的值写入到相应的寄存器中。当 PXUDEN0/1 为 1 时，可以通过 PXUDEVT0/1/2/3 来控制占空比、周期、预分频、更新计数器和输出控制的更新到缓冲中。PXUDEVT1/3 控制周期、预分频、更新计数器和占空比的更新，PXUDEVT0/2 控制输出控制寄存器的更新。若 PXUDEVT1/3 为 0，则周期、预分频、更新计数器和占空比可以在相应的定时器为 0/定时器为 PPX 时更新到缓冲中；若 PXUDEVT1/3 为 1 时，则立即更新周期、预分频、更新计数器和占空比到缓冲中，同时定时器清零。若 PXUDEVT0/2 为 0，则输出控制寄存器可以在定时器为 0/定时器为 PPX 时更新到缓冲中；若 PXUDEVT0/2 为 1 时，则立即更新输出控制寄存器到缓冲中，此时定时器继续计数，不影响占空比和周期的完整性。

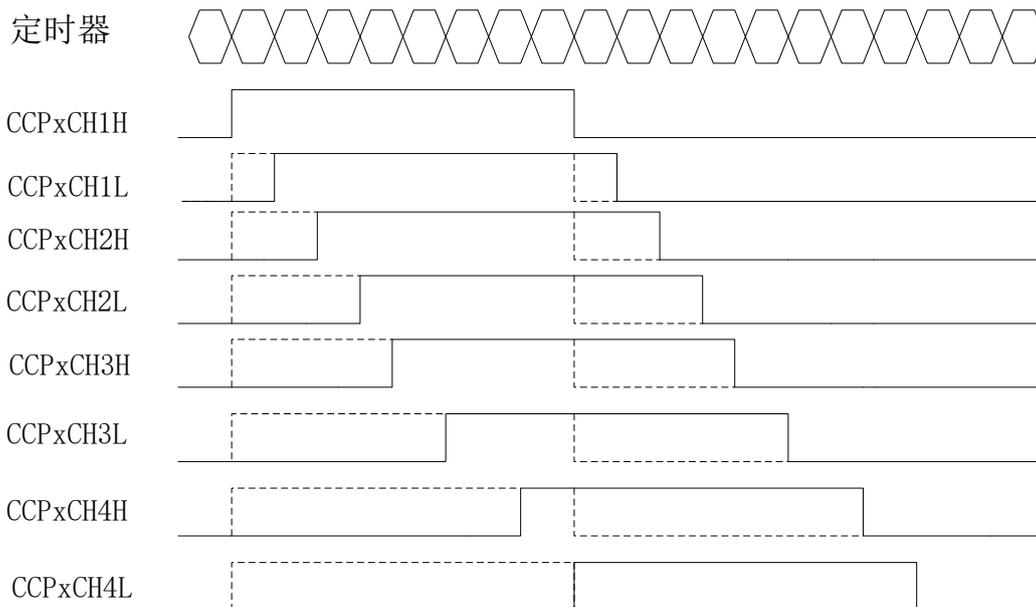
注意，当 PXUDEVT1/3 为 1 时，会清零定时器，此时若 PXUDEVT0/2 为 0，也会引起输出控制的更新。

发生更新事件时，会将 ECCPx_EGIF 寄存器中的更新中断标志位 TXUIF 置位，该位需要软件将 ECCPx_SRIC 寄存器中的 TXUIC 位置 1 才会清零。如果 ECCPx_IE 寄存器中的更新中断使能位 TXUIE 位已经置位，则中断模块部分的寄存器 INT_EIF1 中相应的中断标志位置 1。且可产生 DMA 请求，见“Tx/Tz 触发 DMA(高级定时器)”。

注：PXUDEVT0/1/2/3 是上升沿有效，因此置 1 后必须软件清零再置 1 才能使能下一次的立即更新控制。

15.6.11 自动 PWM 相移

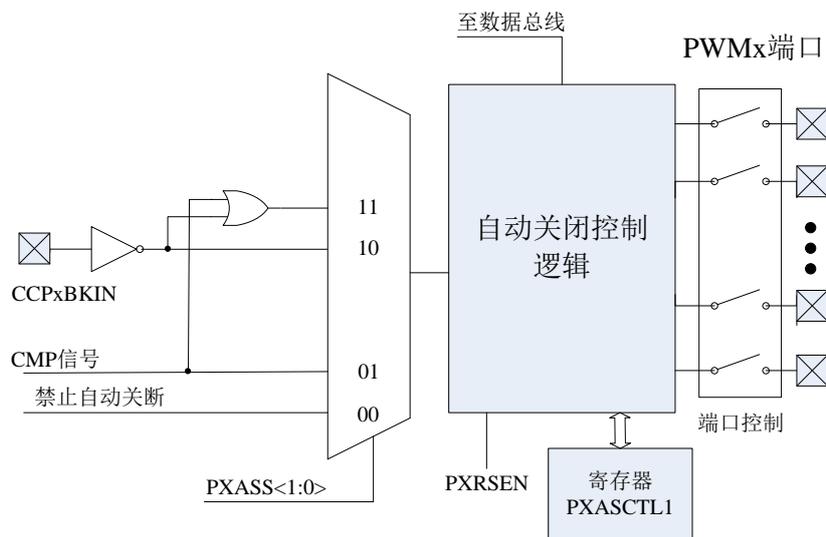
通道上下 2 路输出同时开关可能会导致供应电源电压的起伏，增加电磁干扰的易感性。为了防止此类事件的发生，用户可以通过将 ECCPx_PXUDCTL 的 PXMPEN 位置 1，对于 4 路驱动电路，输出信号会自动移动一个计数时钟。


图 15-13 ECCCP 自动相移模式

15.6.12 自动关断模式

PWMx 模块具有自动关断功能。如下图所示，为自动关断模式的原理框图。使能自动关断功能后，在外部关断事件发生时，该功能自动禁止 PWM 输出，然后四个通道的八个引脚输出电平置于其预定义的状态。此模式用于防止 PWM 破坏应用电路。

自动关断模式具有 3 个关断源：ECCPxBKIN 引脚的逻辑低电平、比较器 2/3 输出高电平和在软件中直接将 PXASEy 位置 1。关断源触发关断的信号是高电平或低电平，而不是上升沿或下降沿，只要关断源的关断电平存在，自动关断状态将保持。


图 15-14 ECCCP 自动关断模式原理框图

通过寄存器 ECCPx_PXASCTL0 选择相应通道的自动关断源。

寄存器 ECCPx_PXASCTL0 中的 PXASEy 位指示通道 y 关断的状态。如果某位为 0，则表示该通道的引脚输出正常的 PWM 信号，如果该位为 1，表示该通道的引脚输出处于关断

状态。

发生关断事件时，将会：

- (1) $PXASEy$ 位被置 1。直到被软件清零或发生自动重启才会将该位清零。
- (2) 使能的 2 个 PWM 引脚将被置于关断电平状态。
- (3) 若 $ECCPx_DE$ 寄存器中对应的 $ECCPxASDE$ 位为 1，则会产生 DMA 请求。

关断时，八引脚输出电平的状态由寄存器 $ECCPx_PXASCTL1$ 决定。通过设定可将输出引脚置为：三态、高电平和低电平三种状态。

打开自动关断功能后，如果关断源产生关断事件，则 $PXASEy$ 标志位被硬件置 1，八路输出被驱动为关断模式电平；关断源清除关断事件后， $PXASEy$ 仍然为 1(如果 $PXRSEN=0$)，八路输出仍然为关断模式电平，直到将 $PXASEy$ 位清零，PWM 重启。

15.6.13 自动重启模式

可将 PWMx 配置为一旦清除自动关断条件就自动重启 PWM。通过将 $ECCPx_CTL3$ 寄存器中的 $PXRSEN$ 位置 1 使能自动重启。

如果使能自动重启，只要自动关断条件有效， $PXASEy$ 位就将保持置 1。当清除自动关闭条件时，将通过硬件将 $PXASEy$ 位清零，并且将恢复常规操作。

如下图所示，在图(a)中， $PXRSEN=0$ ，自动重启模式关闭，通过软件将 $PXASEy$ 标志位清零，PWM 才会重启。在图(b)中， $PXRSEN=1$ ，自动重启模式被打开，关断事件被清除后， $PXASEy$ 标志位由硬件自动清零，然后重启 PWM。

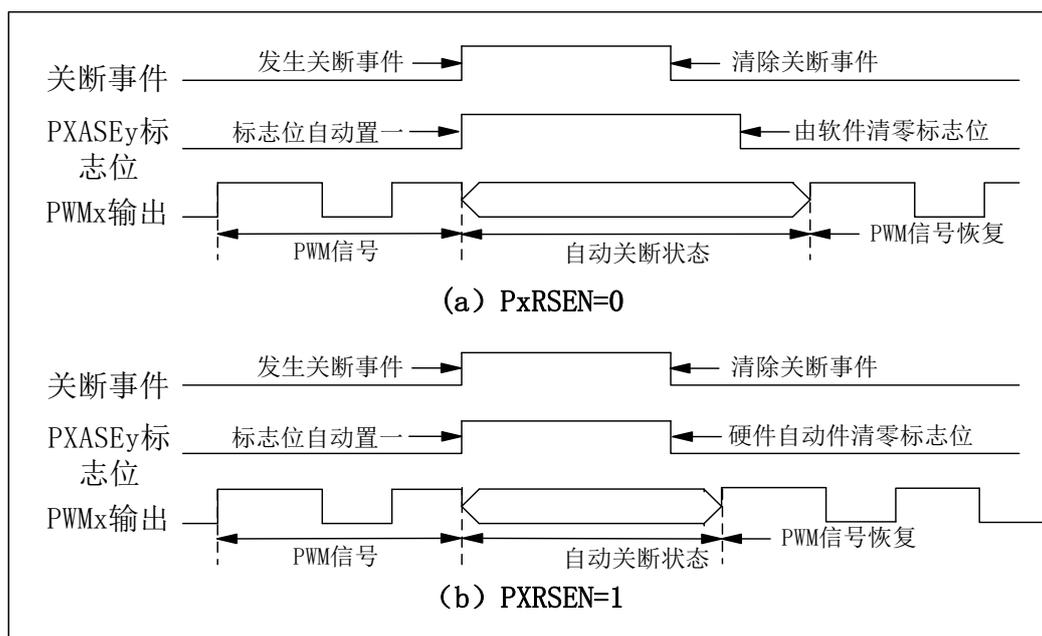


图 15-15 ECCP 自动重启与软件重启示例

如下表所示，在 $PXASEy$ 标志位清零后，PWM 信号恢复时刻和计数方式相关。

表 15-24 PWM 信号重启恢复时刻表

计数方式	PWM 信号恢复时刻
向上计数	$Tx=0$
向下计数	$Tx=PPX$
向上向下计数	$Tx=0$

15.6.14 位置检测

位置检测功能原理框图如下：

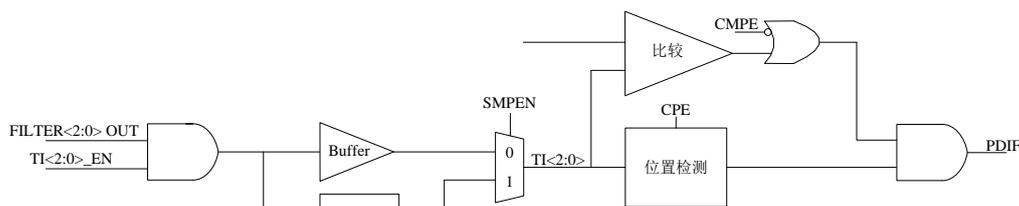


图 15-16 ECCP 位置检测原理框图

15.6.14.1 位置检测输入信号

位置检测的输入信号为模拟比较器模块（CMP0/1/2）使用的数字滤波器输出 FILTER_x_OUT（x=0,1,2），其输入信号可选择比较器输出或者 GPIO 外部输入信号，可以通过使能比较器的数字滤波器对输入信号进行数字滤波处理。

TI<2:0>_EN 位用于使能 TI<2:0>信号的通道检测，当 TI<2:0>=111 时，TI<2:0>通道同时处于检测状态；当 TI<2:0>=011 时，TI2 通道被禁止，TI1 和 TI0 通道处于检测状态。

15.6.14.2 输入信号采样功能

将 SMPEN 位置 1 可启用输入信号采样功能。比较器的输出信号或 GPIO 外部输入信号可能受到外围驱动电路 MOS 开关的干扰，叠加干扰噪声信号。在滤波器功能之外，还会提供一个采样模块，以 T1 为时基，对输入信号进行采样，从而消除噪声。使用采样功能会导致信号存在 0~1 个采样时钟周期的延时。

采样点为 CCP1 模块的 PWM1 占空比翻转点，因此在使用采样功能时，需要使能 CCP1 模块，配置周期、占空比寄存器，不配置 CCP1 的重映射功能。

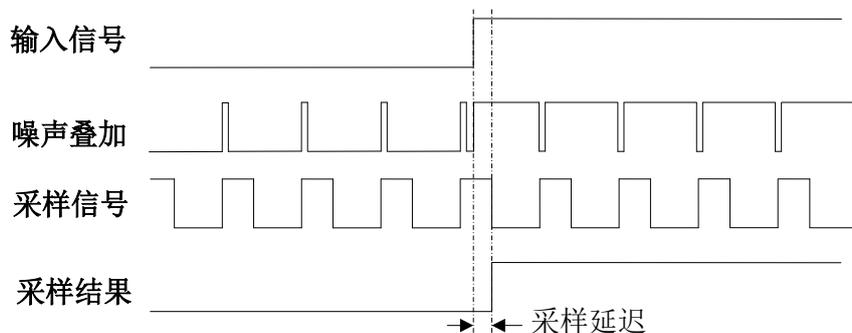


图 15-17 信号采样时序图

15.6.14.3 位置检测事件中断标志位

位置检测事件可以通过如下两种方式产生：

- 检测输入（TI2/TI1/TI0）的有效沿产生
- RDA<2:0>和 CPD<2:0>发生匹配事件，由检测输入的有效沿产生

当 CMPE=0 时，位置检测事件中断标志位 PDIF 在 TI_x（x=0,1,2）信号的有效沿到来时置 1。CPE<1:0>位用于选择 TI_x 信号的有效沿。

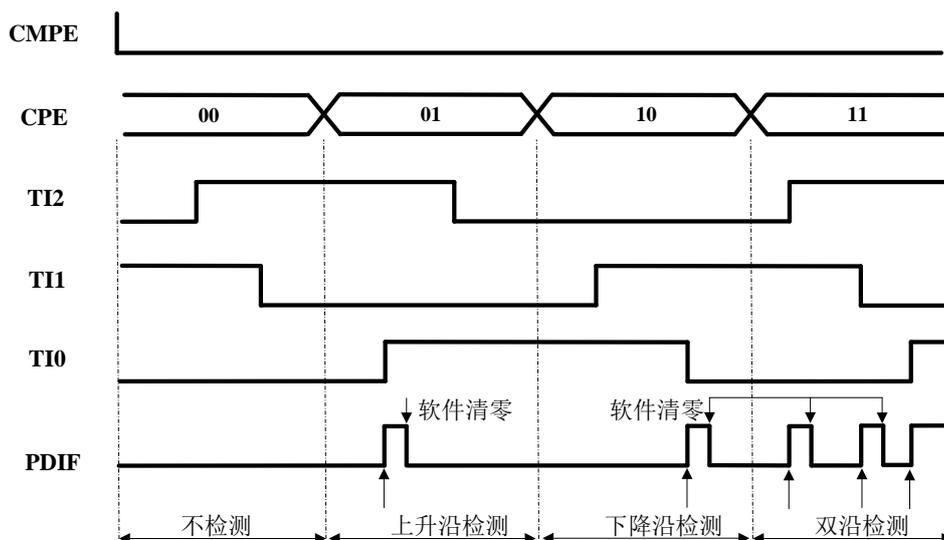


图 15-18 CMPE=0 位置检测时序图

当 CMPE=1 时，在 TI_x ($x=0,1,2$) 信号的有效沿到来时，且 $RDA<2:0>$ 位的值与 TI_x ($x=0,1,2$) 相等时，位置检测事件中标志位 PDIF 置 1。CPE<1:0>位用于选择 TI_x 信号的有效沿。 $RDA<2:0>$ 位的值通过软件配置。

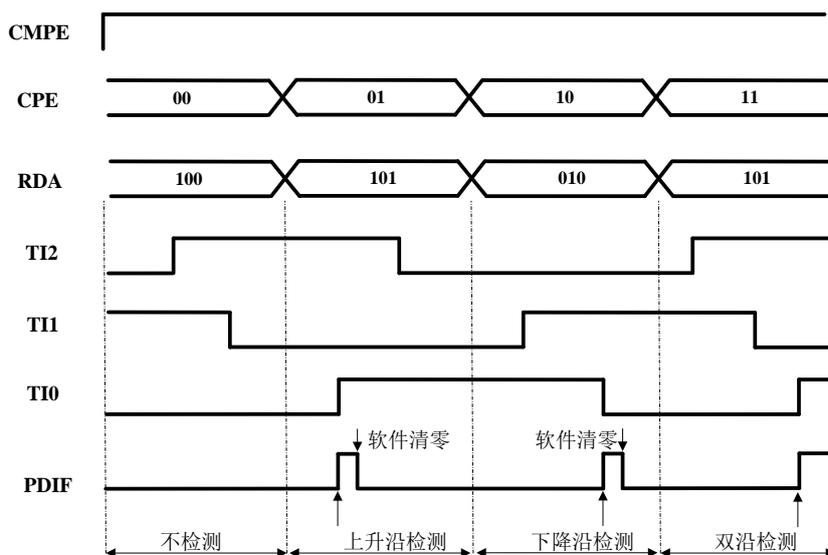


图 15-19 CMPE=1 位置检测时序图

16 正交编码脉冲电路（QEI0/1）

16.1 概述

单片机内部集成有正交编码脉冲电路。正交编码脉冲电路可用于获得旋转机械的位置和速率等信息。

正交编码脉冲是两个频率变化且正交的脉冲。当它由电机轴上的光电编码器产生时（光电编码器具有 3 路输出：A 相、B 相和索引脉冲），电机的旋转方向可以通过检测两个脉冲序列（QEA 和 QEB）中先到达的列来确定，角位置和转速可由脉冲数和脉冲频率（即齿脉冲和圈脉冲）来决定。电机的绝对位置以索引脉冲为基准确定。

QEI 由用于解析 A 相（QEA）和 B 相（QEB）信号的解码器逻辑以及用于累计计数值的递增/递减计数器组成。输入端上的数字噪声滤波器对输入信号进行滤波。

QEI0 的计数时基为定时器 T7，QEI1 的计数时基为定时器 T8。

QEI 的工作特性包括：

- 3 路输入通道，分别为两相信号和索引脉冲输入
- 输入端上的可编程数字噪声滤波器
- 16 位递增/递减位置计数器
- 计数方向状态
- x2 和 x4 计数分辨率
- 两种位置计数器复位模式：
 - 使用周期复位位置计数器
 - 使用索引脉冲复位位置计数器
- 通用 16 位定时器/计数器模式
- 正交编码器接口中断

16.2 正交编码器模块结构框图

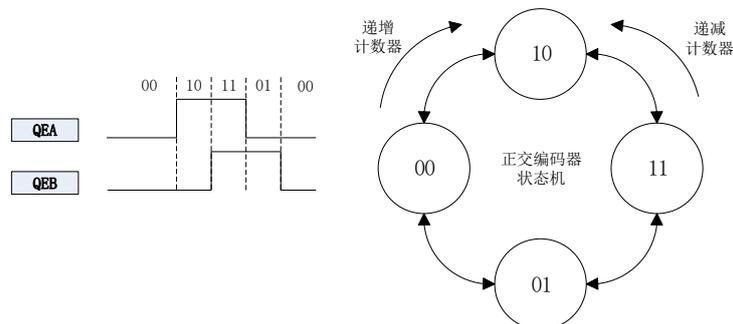


图 16-1 QEI 模块状态分布图

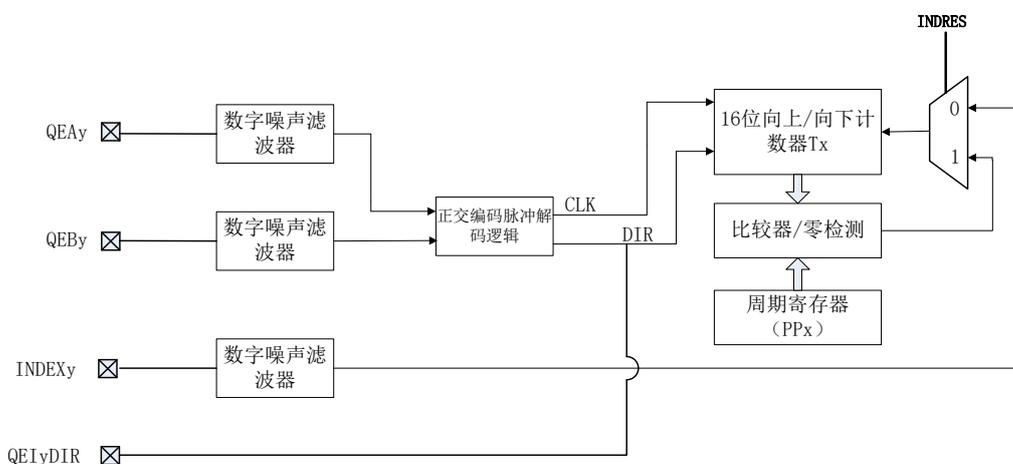


图 16-2 QEI 模块结构框图

- QEAy¹引脚: QEIA 通道引脚, 通过 IO 口重映射功能映射到 IO 口。
- QEB y 引脚: QEIB 通道引脚, 通过 IO 口重映射功能映射到 IO 口。
- INDEX y 引脚: QEI y 的索引脉冲引脚, 通过 IO 口重映射功能映射到 IO 口。索引脉冲通过该引脚输入模块。
- QEI y DIR 引脚: QEI y 的方向输出引脚, 通过 IO 口重映射功能映射到 IO 口。QEI y_CTL0 的 TXDIR 位通过该引脚输出到 IO 口。

¹ 本章除特殊说明外, x=7,8, y=0,1

16.3 QEI 模块相关寄存器

表 16-1 QEI 模块相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	QEIy_CTL0	R/W	QEIy 控制寄存器	0x0000 0000
0x004	QEIy_CTL1	R/W	QEIy 数字滤波控制寄存器	0x0000 0000
0x008	Tx_CNT	R/W	Tx 寄存器	0x0000 0000
0x00C	Tx_PPX	R/W	PPX 周期寄存器	0x0000 0000
0x010	Tx_PRSC	R/W	Tx 预分频器	0x0000 0000
0x014	QEIy_DIER	R/W	QEIy 中断控制寄存器	0x0000 0000

QEI0 基地址：0x4000 0500

QEI1 基地址：0x4000 0080

QEI 模块有 6 个可由用户访问的寄存器。这 6 个寄存器分别是：

- 控制/状态寄存器 (QEIy_CTL)：控制 QEIy 模块的操作并提供模块状态标志。
- 位置计数寄存器 (Tx_CNT)：该寄存器允许读写 16 位位置计数器。
- 最大计数寄存器 (PPX)：PPX 寄存器保存在某些操作中用来与 Tx_CNT 计数器进行比较的值。
- 计数预分频配置寄存器 (Tx_PRSC)：计数器的 16 位预分频比配置。
- 中断状态控制寄存器 (QEIy_DIER)：控制 QEI 的中断使能以及中断标志清除。

16.3.1 QEIy_CTL0 QEIy 控制寄存器

表 16-2 QEIy_CTL0 QEIy 控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																			
R/W																						R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																						TXIF	PHEIF	CNTERRIF	QEIMOD<2:0>			INDRES	TXCS	INDEX	TXDIROUT	TXDIR																		

TXIF：计数器 Tx 溢出中断标志位

0 = 计数器 Tx 未发生向上或者向下溢出

1 = 计数器 Tx 发生向上或者向下溢出（包括索引脉冲 INDEX 触发）

PHEIF：QEIy 相位错误中断标志位

0 = QEIy 相位未出现错误

1 = QEIy 相位出现错误

CNTERRIF：QEIy 计数错误中断标志位

0 = QEIy 计数未出现错误

1 = QEIy 计数出现错误

（仅当 QEIMOD<2:0> = 110 或者 100 时，才能使用该位）

QEIMOD<2:0>：正交编码器接口模式选择位

000 = 正交编码器接口/定时器关闭

001 = 使能计数方向/时钟可选的通用计数器模式

010 = 保留

011 = 保留

- 100 = 使能正交编码器接口 (x2 模式), 索引脉冲复位位置计数器
- 101 = 使能正交编码器接口 (x2 模式), 匹配时复位位置计数器 (PPX)
- 110 = 使能正交编码器接口 (x4 模式), 索引脉冲复位位置计数器
- 111 = 使能正交编码器接口 (x4 模式), 匹配时复位位置计数器 (PPX)

INDRES: 位置计数器复位使能位

- 0 = 索引脉冲不复位位置计数器
- 1 = 索引脉冲复位位置计数器
(仅当 QEIMOD<2:0> = 110 或者 100 时, 才能使用该位)

TXCS: 定时器时钟源选择位

- 0 = 内部时钟 (由 QEICK<1:0>决定)
- 1 = 来自 QEAY 引脚的外部时钟 (下降沿触发计数)
(仅当 QEIMOD<2:0> = 001 时, 才能使用该位)

INDEX: 索引引脚状态位 (只读)

- 0 = 索引引脚为低电平
- 1 = 索引引脚为高电平

TXDIROUT: 位置计数器方向状态输出使能位

- 0 = 禁止位置计数器方向状态输出
- 1 = 使能位置计数器方向状态输出

TXDIR: 位置计数器方向状态位

- 0 = 位置计数器方向为反向 (-)
- 1 = 位置计数器方向为正向 (+)
(当 QEIMOD<2:0> = 1xx 时为只读位)
(当 QEIMOD<2:0> = 001 时为读/写位)

注意: QEI 模块没有独立的使能位, 当 QEIMOD<2:0>≠000 时, QEI 即开始工作。

16.3.2 QEIy_CTL1 QEIy 数字滤波控制寄存器

表 16-3 QEIy_CTL1 QEIy 数字滤波控制寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																						
R/W																									R/W	R/W	R/W	R/W	R/W	R/W																								
位名																									QEICK<1:0>		QEIOUT		QEICK<2:0>		SWPAB																							

QEICK<1:0>: QEIy 工作时钟选择位

- 00 = SCLK
- 01 = HFCLK
- 1x = LFCLK

QEIOUT: 数字滤波器输出使能位

- 0 = 禁止数字滤波器输出
- 1 = 使能 QEAY/QEBY/INDXY 引脚上的数字滤波器输出

QEICK<2:0>: 数字滤波器时钟分频选择位

- 000 = 为 QEAY/QEBY/INDXY 滤波时钟进行 1:1 分频

16.4 可编程数字噪声滤波器

定时器 T7 分配给 QEIO, 定时器 T8 分配给 QEI1; 两个 QEI 模块完全相同, 下面以 QEIO 为例介绍 QEI 模块。

只有输入电平在 3 个连续的时钟上升沿都获得同一个值之后, 才允许经过滤波的输出信号发生变化。结果就是时钟上升沿之间短的噪声尖峰被忽略, 短于两个时钟周期的脉冲被滤除。滤波器时钟是由工作时钟经过可编程分频器分频后得到。将 QEIy_CTL1 寄存器的 QEIOUT 位置 1, 将使能 QEAy、QEBY、和 INDXy 输入的滤波器。时钟位 QEICK<2:0>用于指定滤波器的时钟分频比。

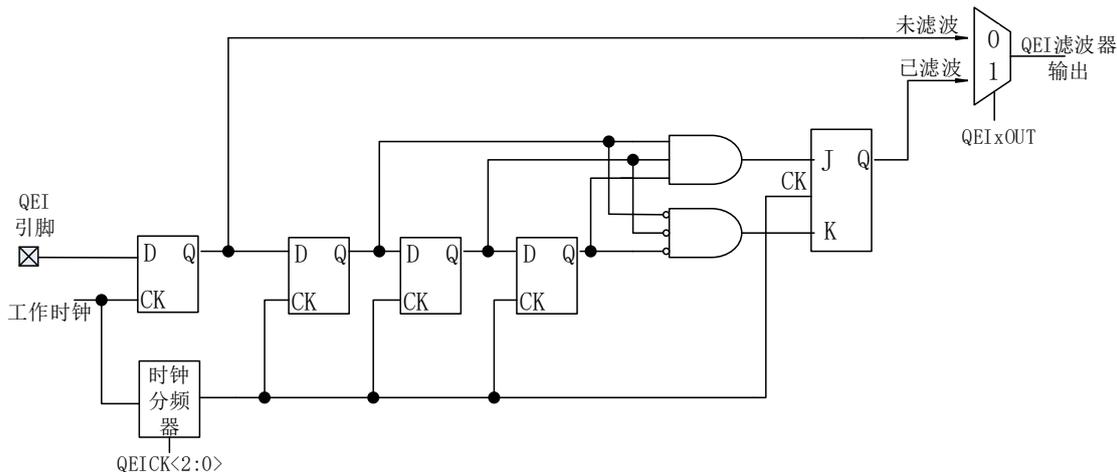


图 16-4 QEI 可编程噪声滤波器

16.5 正交编码器

QEI 正交编码器将已经滤波的输入信号转换为计数信息。QEI 电路将输入信号的分辨率乘以 2 或者 4 的因数 (x2 或 x4 模式)。

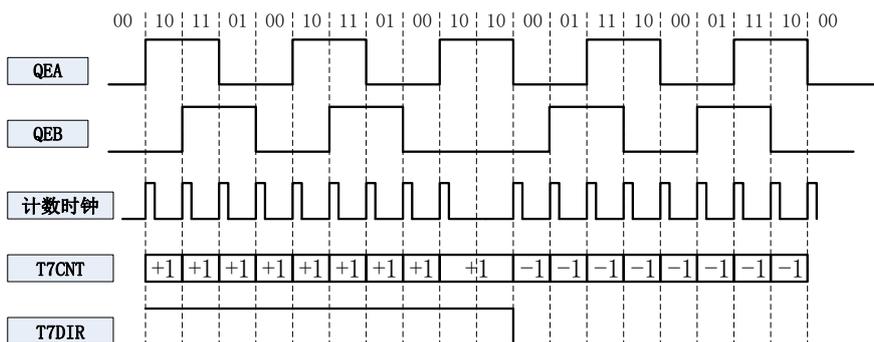
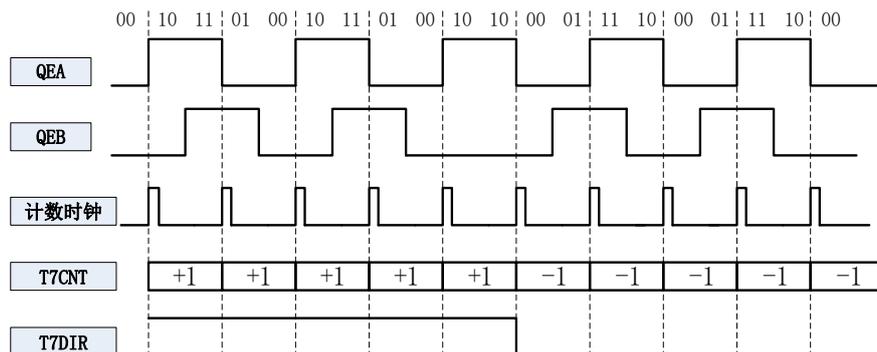


图 16-5 x4 模式下正交编码信号

选择 x4 测量模式时, QEI 逻辑在 A 相和 B 相输入信号的上升沿和下降沿都使位置计数器计数。


图 16-6 x2 模式下正交编码信号

选择 x2 测量模式，QEI 逻辑仅根据 A 相输入的上升沿和下降沿来确定位置计数器的递增速率。A 相的每个上升沿和下降沿都是位置计数器计数，B 相仍用于确定计数器方向，与 x4 模式下完全一样。

16.5.1 计数方向状态

QEI 逻辑根据 A 相和 B 相的时间关系产生 TXDIR 信号，如果 A 相超前 B 相，那么认为电机正向旋转。如果 A 相滞后于 B 相，那么认为电机反向旋转。使能 TXDIROUT，计数方向可以通过引脚输出。

16.5.2 编码器计数方向

正交计数的方向由数字滤波控制寄存器 QEIy_CTL1 中的输入交换选择位 SWPAB 决定。如果 SWPAB=0，A 相输入送到正交计数器的 A 输入，B 相输入送到正交计数器的 B 输入。因此，当 A 相信号超前于 B 相信号时，正交计数器在每个边沿都递增。这种情况（QEA 信号超前于 QEB 信号）被定义为运动的正方向。

将 SWPAB 位设置为逻辑 1，使 A 相输入送到正交计数器的 B 输入，B 相输入送到正交计数器的 A 输入。因此，如果引脚上的 A 相信号超前于 B 相信号，那么正交计数器的 A 相输入将滞后于 B 相输入。这种情况下，则为反方向旋转，且计数器在每个正交脉冲递减。

16.5.3 正交速率

位置控制系统的每分钟转数（RPM）各不相同。RPM 与正交编码器线计数一起决定了 QEA 和 QEB 的输入信号的频率。

以 x4 模式为例，一个 6,000RPM 的电机使用 8,192 线编码器产生的正交计数速率为：
 $(6000/60) * (8192*4) = 3.2768\text{MHZ}$ 。

16.6 16 位递增/递减位置计数器

16 位递增/递减计数器在每一个计数脉冲递增或递减计数，该脉冲信号由正交编码器逻辑产生。计数器的计数值与位置成正比。计数方向由 TXDIR 信号决定，该信号由正交编码器接口逻辑产生。

用户软件可以通过读取 TXCNT 寄存器来检查计数的内容。用户软件还可以通过写 TXCNT 寄存器来初始化计数值。

16.6.1 使用 PPX 复位位置计数器 TX

当 QEIMOD<2:0>位为 111 或者 101 时，在位置计数与预先确定的高、低值匹配时，位置计数器将会复位。不使用索引脉冲复位机制。

对于该模式，位置计数器的复位机制以如下方式工作

- 如果编码器正向旋转（QEA 超前于 QEB），并且 TXCNT 寄存器中的值与 PPX 寄存器中的值匹配时，TXCNT 将在下一个使 TXCNT 递增的正交脉冲边沿复位为 0。在此计满返回事件发生时将产生中断。

如果编码器反向旋转（QEB 超前于 QEA），并且 TXCNT 寄存器中的值递减计数至 0，那么在下一个使 TXCNT 递减的正交脉冲边沿 PPX 寄存器中的值会被装入 TXCNT 寄存器。在此下溢事件发生时将产生中断。

当将 PPX 用作位置极限时，位置计数器将以 x2 或 x4 编码器模式计数。对于标准的旋转编码器，写入 PPX 的适当值应该是 2N-1（x2 位置模式）或 4N-1（x4 位置模式），其中 N 为编码器每转一圈的计数数字。

16.6.2 使用索引脉冲复位位置计数器 Tx

在索引脉冲复位模式下，位置计数器复位使能位 INDRES（QEly_CTL0<4>）控制当检测到索引脉冲时是否对位置计数器进行复位。只有当 QEIMOD<2:0> = 110 或 100 时，该位才适用。

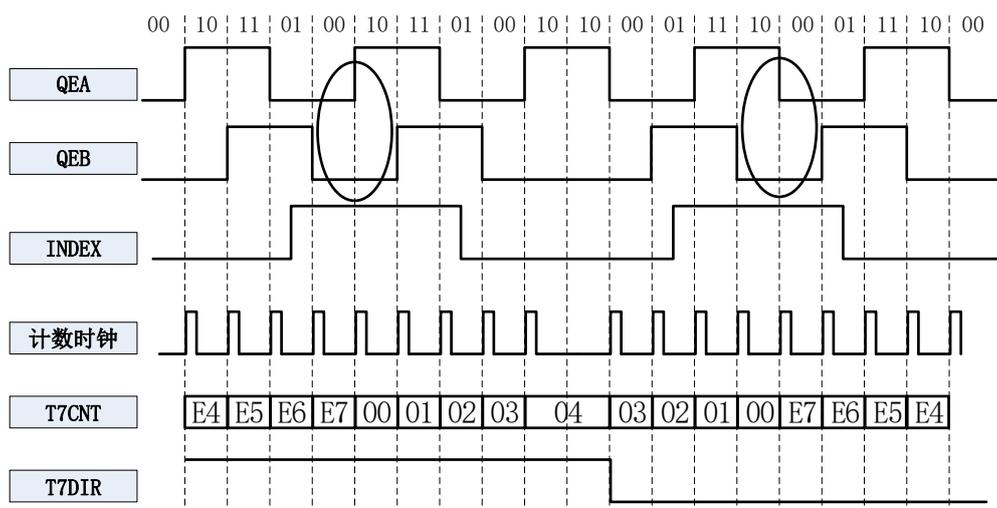


图 16-7 以 x4 模式为例的索引脉冲复位

如上图所示，如果 INDRES 位设置为 1，则位置计数器在检测到索引脉冲时将被复位。对于该模式，位置计数器的复位机制以如下方式工作：

- 如果编码器正向旋转（QEA 超前于 QEB），TXCNT 将复位为 0。
- 如果编码器反向旋转（QEB 超前于 QEA），PPX 寄存器中的值会被装入 TXCNT。

如果 INDRES 位设置为 0，则位置计数器在检测到索引脉冲时将不会被复位。位置计数器将继续进行递增或递减计数。

- 如果编码器正向旋转（QEA 超前于 QEB），TXCNT 一直计数到#0xFFFF，溢出到 0 继续向上计数。
- 如果编码器反向旋转（QEB 超前于 QEA），开始时 PPX 寄存器中的值会被装入 TXCNT，一直向下计数到 0 后溢出将 0xFFFF 装入 TXCNT，此时会产生计数错误

中断。

在索引脉冲复位模式下，无论 INDRES 是否置 1，只要检测到索引脉冲都将产生 TX 计数溢出中断将 TXIF 置 1，但如果检测不到索引脉冲在溢出时也不会产生溢出中断。

16.6.3 相位检测

在正交编码器接口模式下 (QEIMOD<2:0> = 111、110、101 或者 100)，QEI 模块会对正交编码脉冲 QEA/QEB 进行检测，当出现错误相位时，中断标志位 PHEIF 将被置 1，如果相位错误中断使能位 PHEIE 置 1，将产生中断触发信号至中断模块；相位错误中断清零位 PHEIC 用于清零 PHEIF 位。

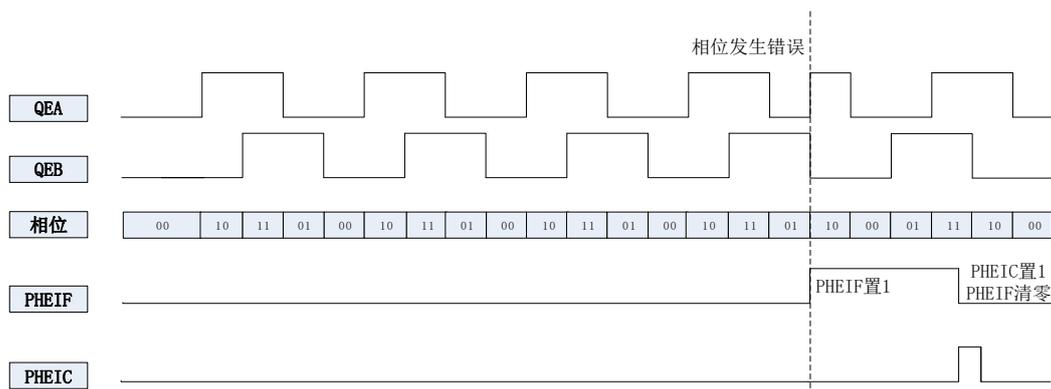


图 16-8 QEI 相位检测错误示意图

16.6.4 计数检测

当控制位 QEIMOD<2:0> = 0x111、0x110、0x101、0x100 或者 0x001 时，QEI 模块提供计数检测功能。在该模式下，正向计数时，计数寄存器 TXCNT 的值会一直与 PPX+1 比较；反向计数时，计数寄存器 TXCNT 的值会与 0xFFFF 比较；以检测计数器是否在 0-PPX 配置值之间向上向下计数。当周期寄存器 PPX 的配置值为 0xFFFF 时，该检测自动屏蔽。

当检测到计数器跑到 0-PPX 配置值以外时，计数错误中断标志位 CNTERRIF 将被置 1，如果计数错误中断标志使能位 CNTERRIE 置 1，将产生中断触发信号至中断模块；计数错误中断清零位 CNTERRIC 用于清零 CNTERRIF 位。

16.7 QEI 作为 16 位定时/计数器

当控制位 QEIMOD<2:0> = 001 时，QEI 作为 16 位定时/计数器使用。此模式下 TXDIR 位可写，用户可以通过该位将 Tx 配置为向上或者向下计数；该模式下还可通过 TXCS 位选择计数时钟。该模式下 Tx 计数溢出标志位为唯一可用标志位，在计数上溢（计数至 PPX 配置值）或者下溢（计数至 0）时 TXIF 置 1。

在配置为向下计数模式时，在配置控制位 QEIMOD<2:0> = 001 后，PPX 的配置值将会载入 TXCNT 寄存器；配置为向上计数模式时，在配置控制位 QEIMOD<2:0> = 001 后，Tx 将会从 TXCNT 的配置值开始向上计数。

若在计数过程中配置 PPX 寄存器，在 Tx 向上计数到 PPX 或向下计数到 0 时更新 PPX 寄存器，不会在写 PPX 寄存器后立即更新。

16.8 16 位预分频器

QEI 模块内包含一个 16 位的预分频器 TXPRSC，可将 QEI 的时基 Tx 的计数时钟信号进行 1~65536 倍分频。通过 TXPRSC 寄存器设置分频倍数。TXPRSC 设置为 0x01，则进行 1:2 分频，以此类推。

16.9 QEI 的使用说明

QEI 默认的计数方向为向下计数，使能 QEI 后由于初始的方向需要等待相位检测，造成初始的计数与实际需要有所偏差，为规避初始计数不可预测，建议用户按照下面方式：

- (1) 为了能够正确使用 QEI，务必在配置时，将周期值同时写入 Tx_CNT 寄存器和 Tx_PPX 寄存器。
- (2) 在第一个有效计数时钟沿，模块会固定将周期值 PPX 重载载入 Tx_CNT 中。之后开始按照预先设置的模式和预分频 PRSC 进行正常计数。
- (3) 对于 x2 模式，有效计数时钟沿指 QEA 通道的每个沿（无论上升下降）；对于 x4 模式，有效计数时钟沿指 QEA 通道和 QEB 通道的每个沿（无论上升下降）。

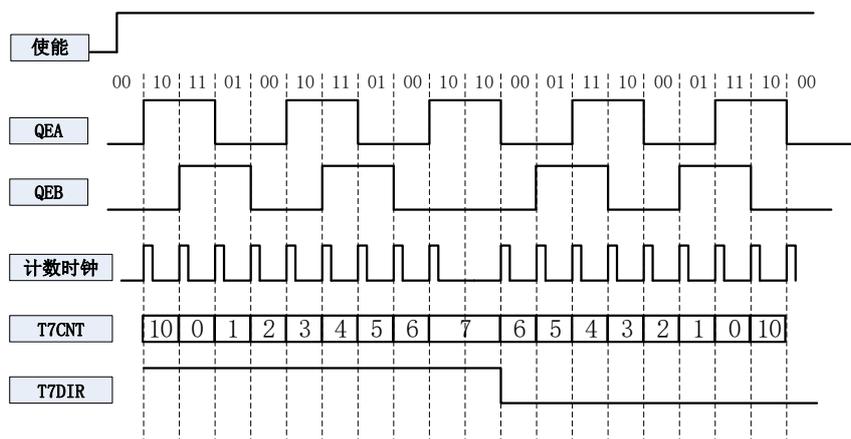


图 16-9 QEI 的启动说明图 1 (X4 模式，周期值为 10，1 分频)

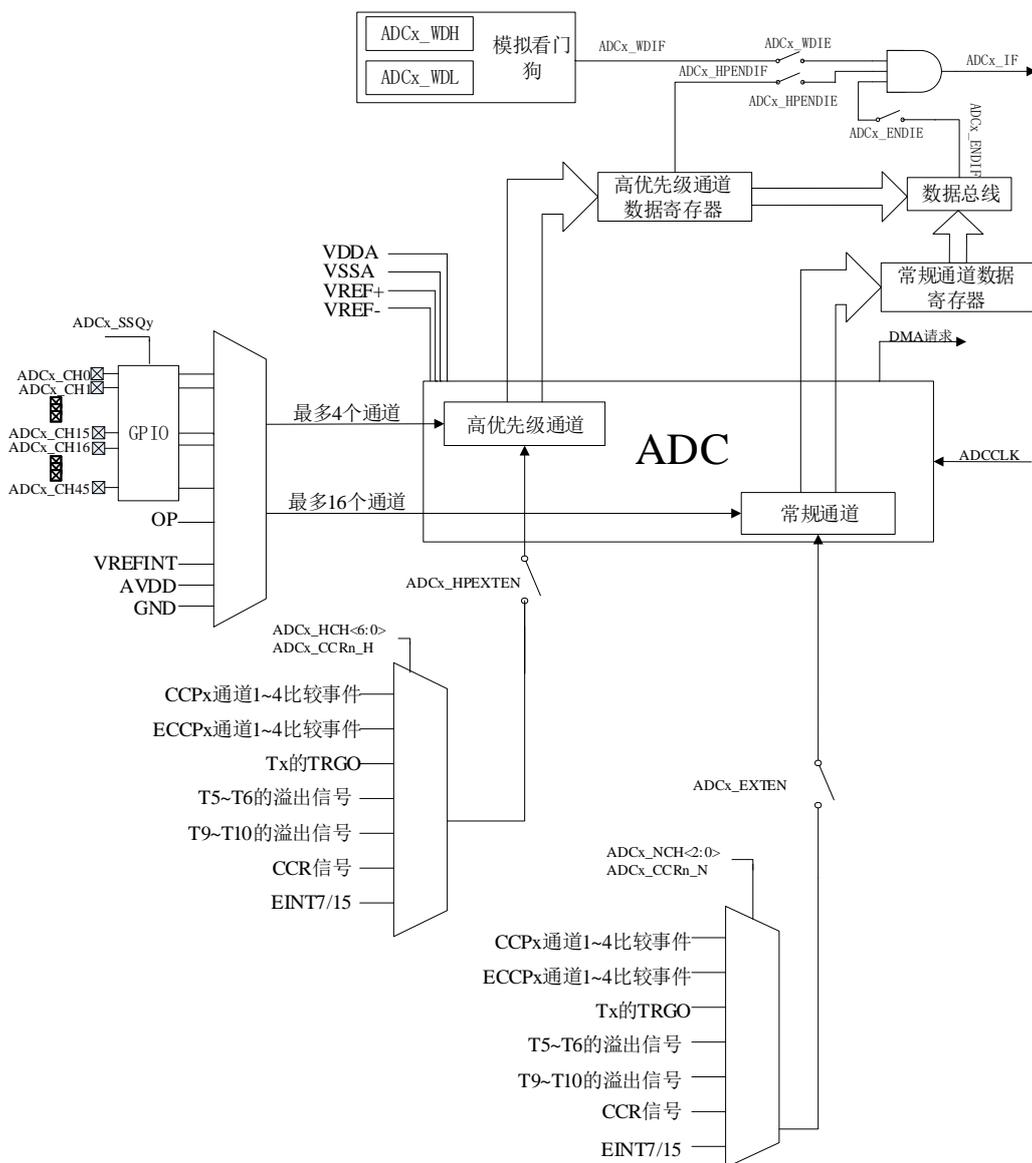
17 模数转换模块 (A/D)

17.1 概述

ADC 特性:

- 12 位分辨率
- 16 常规扫描通道+4 个高优先级通道
- 46 个外部通道+8 个内部信号通道
- 支持常规模式和高优先级模式
- 支持单次转换模式和连续转换模式
- 最高 20 个通道连续转换模式
- 数据左对齐或右对齐
- ADC 支持 DMA 触发
- 支持模拟看门狗事件
- 支持定时器触发 ADC
- 双 AD 模式
- ADC 转换时间: 14 个周期
- AD 电压: 2.4V 到 3.6V 或者 V_{ref+}
- ADC 输入范围 V_{ref-} 到 V_{ref+}

17.2 模数(A/D)转换模块原理


 图 17-1 模数(A/D)转换模块框图¹²³⁴
¹ ADCx_CH0-45 为外部模拟输入信号引脚,VREFINT、OP 为内部信号引脚,可参考内部信号通道小节。

² VDDA、VSSA、VREF+、VREF-为 ADC 电压引脚,详见 ADC 电压引脚小节。

³ CCR 信号参考高级定时器章 Tx/Tz 触发 AD 小节。

⁴ CCPx 通道 1-4 比较事件,其中 x=0/1/2/3/4/18/19/20/21/22/23;ECCPx 通道 1-4 比较事件,其中 x=5/9;Tx 的 TRGO,其中 x=1/2/3/4/18/19/20/21

17.3 ADCx 相关寄存器

 表 17-1 ADC 相关的寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x000	ADCx_CTL0	R/W	ADC 控制寄存器 0	0x0000 0000
0x004	ADCx_CTL1	R/W	ADC 控制寄存器 1	0x0000 0000
0x008	ADCx_SCANSQ0	R/W	ADC 常规通道扫描序列设置寄存器 0	0x0000 0000
0x00C	ADCx_SCANSQ1	R/W	ADC 常规通道扫描序列设置寄存器 1	0x0000 0000
0x010	ADCx_SCANSQ2	R/W	ADC 常规通道扫描序列设置寄存器 2	0x0000 0000
0x044	ADCx_SCANSQ3	R/W	ADC 常规通道扫描序列设置寄存器 3	0x0000 0000
0x014	ADCx_HSCANSQ	R/W	ADC 高优先级通道扫描序列设置寄存器	0x0000 0000
0x018	ADCx_WDH	R/W	ADC 模拟看门狗高阈值寄存器	0x0000 0000
0x01C	ADCx_WDL	R/W	ADC 模拟看门狗低阈值寄存器	0x0000 0000
0x020	ADCx_DATA	R	ADC 常规通道数据寄存器 ²	0xuuuuuuuu
0x024	ADCx_HPDATA0	R	ADC 高优先级通道数据寄存器 0	0xuuuuuuuu
0x028	ADCx_HPDATA1	R	ADC 高优先级通道数据寄存器 1	0xuuuuuuuu
0x02C	ADCx_HPDATA2	R	ADC 高优先级通道数据寄存器 2	0xuuuuuuuu
0x030	ADCx_HPDATA3	R	ADC 高优先级通道数据寄存器 3	0xuuuuuuuu
0x034	ADCx_HPOFF0	R/W	ADC 高优先级通道数据偏移寄存器 0	0x0000 0000
0x038	ADCx_HPOFF1	R/W	ADC 高优先级通道数据偏移寄存器 1	0x0000 0000
0x03C	ADCx_HPOFF2	R/W	ADC 高优先级通道数据偏移寄存器 2	0x0000 0000
0x040	ADCx_HPOFF3	R/W	ADC 高优先级通道数据偏移寄存器 3	0x0000 0000
0x050	ADCx_STATE	R/C	ADC 状态寄存器	0x0000 0000
0x054	ADC0_DELAY	R/W	ADC 快慢交叉延时寄存器 ³	0x0000 0000

ADC0 基址为 0x4000 0580;

ADC1 基址为 0x4000 0600;

ADC2 基址为 0x4000 0680;

17.3.1 ADCx_CTL0 ADC 控制寄存器 0

表 17-2 ADCx_CTL0 ADC 控制寄存器 0

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W					R/W	R/W	R/W	R/W
位名	WDSCH	HPWDEN	WDEN	DSNUM<2:0>			HP3DMAEN	HP2DMAEN	HP1DMAEN	HP0DMAEN	HPDSEN	DSEN	HPAUTO	CONT	SCANEN	BOSSSELCOMP	BOSSSEL	TRIMIBIAS	NDMAEN			ADCCLKS<2:0>							LR	CLKS<1:0>		EN

WDSCH: 扫描模式中在单一通道或所有通道使用模拟看门狗 (watchdog single Channelen)

0 = 在所有通道上都使用模拟看门狗

1 = 在单一通道上使用模拟看门狗, 通道由 WDCH<5:0>位选择

¹ 本章除特殊说明外, x=0,1,2

² ADCx_DATA 或者 ADCx_HPDATA0/1/2/3 的复位值不确定。

³ ADC0_DELAY 只在 ADC0 中有, ADC1 和 ADC2 中并没有

- HPWDEN: 高优先级通道上使能看门狗 (high priority channel watchdog en)
0 = 禁止高优先级通道上模拟看门狗
1 = 使能高优先级通道上模拟看门狗
- WDEN: 常规通道上使能看门狗 (regular channel watchdog en)
0 = 禁止常规通道上模拟看门狗
1 = 使能常规通道上模拟看门狗
- DSNUM<2:0>: 间隔模式通道计数。收到外部触发后已转换常规通道的数量。
000 = 1 个通道
001 = 2 个通道
.....
111 = 8 个通道
- HP3DMAEN: 高优先级通道 3 直接存储器访问模式使能
0 = 不使用 DMA 模式
1 = 使用 DMA 模式
- HP2DMAEN: 高优先级通道 2 直接存储器访问模式使能
0 = 不使用 DMA 模式
1 = 使用 DMA 模式
- HP1DMAEN: 高优先级通道 1 直接存储器访问模式使能
0 = 不使用 DMA 模式
1 = 使用 DMA 模式
- HP0DMAEN: 高优先级通道 0 直接存储器访问模式使能
0 = 不使用 DMA 模式
1 = 使用 DMA 模式
- HPDSEN: 高优先级通道上的间隔模式 (high priority channel discontinuous mode)
0 = 禁止高优先级通道上的间隔模式
1 = 使能高优先级通道上的间隔模式
- DSEN: 常规通道上的间隔模式 (regular channel discontinuous mode)
0 = 禁止常规通道上的间隔模式
1 = 使能常规通道上的间隔模式
- HPAUTO: 自动高优先级通道组转换 (high priority channel auto conversion)
0 = 禁止自动的高优先级通道转换
1 = 使能自动的高优先级通道转换, 当常规通道转换结束后自动转入高优先级通道转换。
- CONT: 连续转换使能
0 = 单次转换模式
1 = 连续转换模式
- SCANEN: 扫描模式使能位
0 = 禁止扫描模式
1 = 使能扫描模式
- BOSSSELCOMP: 比较器的运放输入参考电压二分之一校准
0 = 不校准
1 = 校准, 比较器输入端电压更准确
- BOSSSEL: 参考电压二分之一运放电压校准
0 = 校准, 运放输出电压更准确

- 1 = 不校准
- TRIMIBIAS: 偏置电流校准
 - 0 = 不校准
 - 1 = 校准, 偏置电流可达到 3.5~4uA
- NDMAEN: 常规通道直接存储器访问模式使能
 - 0 = 不使用 DMA 模式
 - 1 = 使用 DMA 模式
- ADCCLKS<2:0>: ADC 转换时钟选择位
 - 000 = T/2
 - 001 = T/4
 - 010 = T/8
 - 011 = T/16
 - 100 = T/32
 - 101 = T/64
 - 其它 = 保留

注: T 为 ADC 转换时钟源, 通过设置 CLKS<1:0>选择得到。
- LR: A/D 转换结果输出格式选择位
 - 0 = 转换结果存储时左对齐
 - 1 = 转换结果存储时右对齐
- CLKS<1:0>: A/D 工作时钟源选择位
 - 00 = 选用 SCLK 作为 A/D 数字模块工作时钟
 - 01 = 选用 HFCLK 作为 A/D 数字模块工作时钟
 - 1x = 选用 LFCLK 作为 A/D 数字模块工作时钟
- EN: A/D 工作使能位
 - 0 = A/D 转换器关闭且不消耗工作电流
 - 1 = 使能 A/D 转换模块工作

17.3.2 ADCx_CTL1 ADCx 控制寄存器 1

表 17-3 ADCx_CTL1 ADCx 控制寄存器 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	DOUMOD<3:0>			HPEXTEN	HCH<6:0>						REF2	REF1	REF0					EXTEN	NCH<6:0>						HPSTART	START						

- DOUMOD<3:0>: 双 ADC 模式选择
- 0000 = 独立模式
 - 0001 = 混合常规通道同步+高优先级通道同步模式
 - 0010 = 混合常规通道同步+交替触发模式
 - 0011 = 混合高优先级通道同步+快速交叉模式
 - 0100 = 高优先级同步模式
 - 0101 = 常规通道同步模式
 - 0110 = 快速交叉同步

0111 = 保留

1000 = 交替触发模式

其他 = 保留

注：仅在 ADC0 中该位有效，在 ADC1,2 中这些位为保留位

HPEXTEN: 高优先级通道外部触发转换模式使能。

0 = 禁止外部事件触发转换

1 = 使能外部事件触发转换

HCH<6:0>: 高优先级通道外部触发事件选择

0000000 = CCP1 通道 1 比较模式触发 ADCx 高优先级通道

0000001 = CCP1 通道 2 比较模式触发 ADCx 高优先级通道

0000010 = CCP1 通道 3 比较模式触发 ADCx 高优先级通道

0000011 = CCP1 通道 4 比较模式触发 ADCx 高优先级通道

0000100 = CCP2 通道 1 比较模式触发 ADCx 高优先级通道

0000101 = CCP2 通道 2 比较模式触发 ADCx 高优先级通道

0000110 = CCP2 通道 3 比较模式触发 ADCx 高优先级通道

0000111 = CCP2 通道 4 比较模式触发 ADCx 高优先级通道

0001000 = CCP3 通道 1 比较模式触发 ADCx 高优先级通道

0001001 = CCP3 通道 2 比较模式触发 ADCx 高优先级通道

0001010 = CCP3 通道 3 比较模式触发 ADCx 高优先级通道

0001011 = CCP3 通道 4 比较模式触发 ADCx 高优先级通道

0001100 = CCP4 通道 1 比较模式触发 ADCx 高优先级通道

0001101 = CCP4 通道 2 比较模式触发 ADCx 高优先级通道

0001110 = CCP4 通道 3 比较模式触发 ADCx 高优先级通道

0001111 = CCP4 通道 4 比较模式触发 ADCx 高优先级通道

0010000 = CCP18 通道 1 比较模式触发 ADCx 高优先级通道

0010001 = CCP18 通道 2 比较模式触发 ADCx 高优先级通道

0010010 = CCP18 通道 3 比较模式触发 ADCx 高优先级通道

0010011 = CCP18 通道 4 比较模式触发 ADCx 高优先级通道

0010100 = CCP19 通道 1 比较模式触发 ADCx 高优先级通道

0010101 = CCP19 通道 2 比较模式触发 ADCx 高优先级通道

0010110 = CCP19 通道 3 比较模式触发 ADCx 高优先级通道

0010111 = CCP19 通道 4 比较模式触发 ADCx 高优先级通道

0011000 = CCP20 通道 1 比较模式触发 ADCx 高优先级通道

0011001 = CCP20 通道 2 比较模式触发 ADCx 高优先级通道

0011010 = CCP20 通道 3 比较模式触发 ADCx 高优先级通道

0011011 = CCP20 通道 4 比较模式触发 ADCx 高优先级通道

0011100 = CCP21 通道 1 比较模式触发 ADCx 高优先级通道

0011101 = CCP21 通道 2 比较模式触发 ADCx 高优先级通道

0011110 = CCP21 通道 3 比较模式触发 ADCx 高优先级通道

0011111 = CCP21 通道 4 比较模式触发 ADCx 高优先级通道

0100000 = 保留
0100001 = 保留
0100010 = 保留
0100011 = 保留
0100100 = CCP22 通道 1 比较模式触发 ADCx 高优先级通道
0100101 = CCP22 通道 2 比较模式触发 ADCx 高优先级通道
0100110 = CCP22 通道 3 比较模式触发 ADCx 高优先级通道
0100111 = CCP22 通道 4 比较模式触发 ADCx 高优先级通道

0101000 = T1TRGO 触发 ADCx 高优先级通道
0101001 = T2TRGO 触发 ADCx 高优先级通道
0101010 = T3TRGO 触发 ADCx 高优先级通道
0101011 = T4TRGO 触发 ADCx 高优先级通道
0101100 = T18TRGO 触发 ADCx 高优先级通道
0101101 = T19TRGO 触发 ADCx 高优先级通道
0101110 = T20TRGO 触发 ADCx 高优先级通道
0101111 = T21TRGO 触发 ADCx 高优先级通道

0110000 = T5TRGO 触发 ADCx 高优先级通道
0110001 = T9TRGO 触发 ADCx 高优先级通道
0110010 = T14TRGO 触发 ADCx 高优先级通道
0110011 = T15TRGO 触发 ADCx 高优先级通道
0110100 = T5 溢出触发 ADCx 高优先级通道
0110101 = T6 溢出触发 ADCx 高优先级通道
0110110 = T9 溢出触发 ADCx 高优先级通道
0110111 = T10 溢出触发 ADCx 高优先级通道
0111000 = CCP9 通道 1 比较模式触发 ADCx 高优先级通道
0111001 = CCP9 通道 2 比较模式触发 ADCx 高优先级通道
0111010 = CCP9 通道 3 比较模式触发 ADCx 高优先级通道
0111011 = CCP9 通道 4 比较模式触发 ADCx 高优先级通道
0111100 = 保留
0111101 = 保留
0111110 = 外部中断线 EINT7 触发 ADCx 高优先级通道
0111111 = 外部中断线 EINT15 触发 ADCx 高优先级通道

1000000 = CCP0 通道 1 比较模式触发 ADCx 高优先级通道
1000001 = CCP0 通道 2 比较模式触发 ADCx 高优先级通道
1000010 = CCP0 通道 3 比较模式触发 ADCx 高优先级通道
1000011 = CCP0 通道 4 比较模式触发 ADCx 高优先级通道
1000100 = CCP23 通道 1 比较模式触发 ADCx 高优先级通道
1000101 = CCP23 通道 2 比较模式触发 ADCx 高优先级通道
1000110 = CCP23 通道 3 比较模式触发 ADCx 高优先级通道
1000111 = CCP23 通道 4 比较模式触发 ADCx 高优先级通道
其它 = 保留

- ADCx_REF2: 内部参考电压作为 AD 参考电压选择位
0 = 禁止选择内部参考电压作为 AD 参考电压
1 = 选择内部参考电压作为 AD 参考电压
- ADCx_REF1: VREF+作为 AD 参考电压选择位
0 = 禁止选择 VREF+作为 AD 参考电压
1 = 选择 VREF+作为 AD 参考电压
- ADCx_REF0: VDDA 作为 AD 参考电压选择位
0 = 禁止选择 VDDA 作为 AD 参考电压
1 = 选择 VDDA 作为 AD 参考电压
- EXTEN: 常规通道外部触发转换模式使能。(external trigger conversion mode)
0 = 禁止外部事件触发转换
1 = 使能外部事件触发转换
- NCH<6:0>: 常规通道外部触发事件选择
0000000 = CCP1 通道 1 比较模式触发 ADCx 普通优先级通道
0000001 = CCP1 通道 2 比较模式触发 ADCx 普通优先级通道
0000010 = CCP1 通道 3 比较模式触发 ADCx 普通优先级通道
0000011 = CCP1 通道 4 比较模式触发 ADCx 普通优先级通道
0000100 = CCP2 通道 1 比较模式触发 ADCx 普通优先级通道
0000101 = CCP2 通道 2 比较模式触发 ADCx 普通优先级通道
0000110 = CCP2 通道 3 比较模式触发 ADCx 普通优先级通道
0000111 = CCP2 通道 4 比较模式触发 ADCx 普通优先级通道

0001000 = CCP3 通道 1 比较模式触发 ADCx 普通优先级通道
0001001 = CCP3 通道 2 比较模式触发 ADCx 普通优先级通道
0001010 = CCP3 通道 3 比较模式触发 ADCx 普通优先级通道
0001011 = CCP3 通道 4 比较模式触发 ADCx 普通优先级通道
0001100 = CCP4 通道 1 比较模式触发 ADCx 普通优先级通道
0001101 = CCP4 通道 2 比较模式触发 ADCx 普通优先级通道
0001110 = CCP4 通道 3 比较模式触发 ADCx 普通优先级通道
0001111 = CCP4 通道 4 比较模式触发 ADCx 普通优先级通道

0010000 = CCP18 通道 1 比较模式触发 ADCx 普通优先级通道
0010001 = CCP18 通道 2 比较模式触发 ADCx 普通优先级通道
0010010 = CCP18 通道 3 比较模式触发 ADCx 普通优先级通道
0010011 = CCP18 通道 4 比较模式触发 ADCx 普通优先级通道
0010100 = CCP19 通道 1 比较模式触发 ADCx 普通优先级通道
0010101 = CCP19 通道 2 比较模式触发 ADCx 普通优先级通道
0010110 = CCP19 通道 3 比较模式触发 ADCx 普通优先级通道
0010111 = CCP19 通道 4 比较模式触发 ADCx 普通优先级通道

0011000 = CCP20 通道 1 比较模式触发 ADCx 普通优先级通道
0011001 = CCP20 通道 2 比较模式触发 ADCx 普通优先级通道
0011010 = CCP20 通道 3 比较模式触发 ADCx 普通优先级通道
0011011 = CCP20 通道 4 比较模式触发 ADCx 普通优先级通道

0011100 = CCP21 通道 1 比较模式触发 ADCx 普通优先级通道
0011101 = CCP21 通道 2 比较模式触发 ADCx 普通优先级通道
0011110 = CCP21 通道 3 比较模式触发 ADCx 普通优先级通道
0011111 = CCP21 通道 4 比较模式触发 ADCx 普通优先级通道

0100000 = 保留
0100001 = 保留
0100010 = 保留
0100011 = 保留
0100100 = CCP22 通道 1 比较模式触发 ADCx 普通优先级通道
0100101 = CCP22 通道 2 比较模式触发 ADCx 普通优先级通道
0100110 = CCP22 通道 3 比较模式触发 ADCx 普通优先级通道
0100111 = CCP22 通道 4 比较模式触发 ADCx 普通优先级通道

0101000 = T1TRGO 触发 ADCx 普通优先级通道
0101001 = T2TRGO 触发 ADCx 普通优先级通道
0101010 = T3TRGO 触发 ADCx 普通优先级通道
0101011 = T4TRGO 触发 ADCx 普通优先级通道
0101100 = T18TRGO 触发 ADCx 普通优先级通道
0101101 = T19TRGO 触发 ADCx 普通优先级通道
0101110 = T20TRGO 触发 ADCx 普通优先级通道
0101111 = T21TRGO 触发 ADCx 普通优先级通道

0110000 = T5TRGO 触发 ADCx 普通优先级通道
0110001 = T9TRGO 触发 ADCx 普通优先级通道
0110010 = T14TRGO 触发 ADCx 普通优先级通道
0110011 = T15TRGO 触发 ADCx 普通优先级通道
0110100 = T5 溢出触发 ADCx 普通优先级通道
0110101 = T6 溢出触发 ADCx 普通优先级通道
0110110 = T9 溢出触发 ADCx 普通优先级通道
0110111 = T10 溢出触发 ADCx 普通优先级通道
0111000 = CCP9 通道 1 比较模式触发 ADCx 普通优先级通道
0111001 = CCP9 通道 2 比较模式触发 ADCx 普通优先级通道
0111010 = CCP9 通道 3 比较模式触发 ADCx 普通优先级通道
0111011 = CCP9 通道 4 比较模式触发 ADCx 普通优先级通道
0111100 = 保留
0111101 = 保留
0111110 = 外部中断线 EINT7 触发 ADCx 普通优先级通道
0111111 = 外部中断线 EINT15 触发 ADCx 普通优先级通道

1000000 = CCP0 通道 1 比较模式触发 ADCx 普通优先级通道
1000001 = CCP0 通道 2 比较模式触发 ADCx 普通优先级通道
1000010 = CCP0 通道 3 比较模式触发 ADCx 普通优先级通道
1000011 = CCP0 通道 4 比较模式触发 ADCx 普通优先级通道

- 1000100 = CCP23 通道 1 比较模式触发 ADCx 普通优先级通道
- 1000101 = CCP23 通道 2 比较模式触发 ADCx 普通优先级通道
- 1000110 = CCP23 通道 3 比较模式触发 ADCx 普通优先级通道
- 1000111 = CCP23 通道 4 比较模式触发 ADCx 普通优先级通道
- 其它 = 保留

HPSTART: A/D 转换状态位, 由软件设置该位以启动转换, 转换开始后硬件马上清除此位。
该位可以用于启动一组高优先级通道的转换。

- 0 = 没有启动软件触发高优先级通道转换
- 1 = 开始转换高优先级通道

START: A/D 转换状态位, 由软件设置该位以启动转换, 转换开始后硬件马上清除此位。该位可以用于启动一组常规通道的转换。

- 0 = 没有启动软件触发常规通道转换
- 1 = 开始转换规则通道

17.3.3 ADCx_SCANSQ0 常规通道扫描序列设置寄存器

表 17-4 ADCx_SCANSQ0 常规通道扫描序列设置寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W																																
位名			SSQ4<5:0>					SSQ3<5:0>					SSQ2<5:0>					SSQ1<5:0>					SSQ0<5:0>									

SSQx<5:0>: 常规通道扫描的第 x 个转换, 可选择任意的 AD 输入。

- 000000 = 通道 ADCx_CH0
-
- 001000 = 通道 ADCx_CH8
- 001001 = 通道 ADCx_CH9
- 001010 = 通道 ADCx_CH10
-
- 111111 = 通道 ADCx_CH63

17.3.4 ADCx_SCANSQ1 常规通道扫描序列设置寄存器

表 17-5 ADCx_SCANSQ1 常规通道扫描序列设置寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W																																
位名			SSQ9<5:0>					SSQ8<5:0>					SSQ7<5:0>					SSQ6<5:0>					SSQ5<5:0>									

SSQx<5:0>: 常规通道扫描的第 x 个转换, 可选择任意的 AD 输入。

- 000000 = 通道 ADCx_CH0
-

.....
 111111 = 通道 ADC_x_CH63
 SSQ15<5:0>: 常规通道扫描的第 16 个转换, 可选择任意的 AD 输入
 000000 = 通道 ADC_x_CH0

 001000 = 通道 ADC_x_CH8
 001001 = 通道 ADC_x_CH9
 001010 = 通道 ADC_x_CH10

 111111 = 通道 ADC_x_CH63

17.3.7 ADC_x_HSCANSQ 高优先级通道扫描序列设置寄存器

表 17-8 ADC_x_HSCANSQ 高优先级通道扫描序列设置寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W			R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名			HSLLEN <1:0>						HSSQ4<5:0>				HSSQ3<5:0>				HSSQ2<5:0>				HSSQ1<5:0>											

HSLLEN<1:0>: 高优先级通道扫描长度。

- 00 = 扫描 1 个转换
- 01 = 扫描 2 个转换
- 10 = 扫描 3 个转换
- 11 = 扫描 4 个转换

HSSQ4<5:0>: 高优先级通道扫描的第 4 个转换, 可选择任意的 AD 输入。

- 000000 = 通道 ADC_x_CH0
-
- 001000 = 通道 ADC_x_CH8
- 001001 = 通道 ADC_x_CH9
- 001010 = 通道 ADC_x_CH10
-
- 111111 = 通道 ADC_x_CH63

HSSQ3<5:0>: 高优先级通道扫描的第 3 个转换, 可选择任意的 AD 输入, 设置同上。

HSSQ2<5:0>: 高优先级通道扫描的第 2 个转换, 可选择任意的 AD 输入, 设置同上。

HSSQ1<5:0>: 高优先级通道扫描的第 1 个转换, 可选择任意的 AD 输入, 设置同上。

17.3.8 ADCx_WDH 模拟看门狗高阈值寄存器

表 17-9 ADCx_WDH 模拟看门狗高阈值寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																		
R/W																					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																					WDH<11:0>																													

WDH<11:0>: 模拟看门狗高阈值设置值。

17.3.9 ADCx_WDL 模拟看门狗低阈值寄存器

表 17-10 ADCx_WDL 模拟看门狗低阈值寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																		
R/W																					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																					WDL<11:0>																													

WDL<11:0>: 模拟看门狗低阈值设置值。

17.3.10 ADCx_DATA 常规通道数据寄存器

表 17-11 ADCx_DATA 常规通道数据寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0															
R/W																	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u
位名																	DATA<15:0>																														

DATA<15:0>: 常规通道转换结果数据。数据对齐方式由 LR 决定。

17.3.11 ADCx_HPDATAy 高优先级通道数据寄存器

表 17-12 ADCx_HPDATAy 高优先级通道数据寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0															
R/W																	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u	R u
位名																	HPDATA<15:0>																														

HPDATA<15:0>: 高优先级通道转换结果数据。数据对齐方式由 LR 决定。

17. 3. 12 ADCx_HPDOFFy 高优先级通道数据失调寄存器

表 17-13 ADCx_HPDOFFy 高优先级通道数据失调寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W																						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																					HPDOFF<11:0>												

HPDOFF<11:0>: 高优先级通道 y 转换结果数据偏移。当转换高优先级通道时, 这些位定义了用于从原始转换数据减去的数值。校正后的数值可以从 HPDATA<15:0>读出。

17. 3. 13 ADCx_STATE ADC 状态寄存器

表 17-14 ADCx_STATE ADC 状态寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W	R/W	R/W			R/W		R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R													
位名	TRIMERROR <1:0>				TRIMBIASOPA		CCR5H	CCR5N	CCR4H	CCR4N	CCR3H	CCR3N	CCR2H	CCR2N	CCR1H	CCR1N	CCR0H	CCR0N	EOC1E	EOC1C	EOC1F	WDIE	HPENDIE	ENDIE	WDIC	HPENDIC	ENDIC	WDIF	HPENDIF	ENDIF	HPSTARTIF	STARTIF	

TRIMERROR<1:0>: 增加正 offset 调节位 (默认为 0)

TRIMBIASOPA: 模拟运放输出级驱动调节档

- 0 = 关闭
- 1 = 打开

CCR5H: 高优先级通道的 T10_CCR0 触发使能

- 0 = 高优先级通道触发使能关闭
- 1 = 高优先级通道触发使能打开

CCR5N: 常规优先级通道的 T10_CCR0 触发使能

- 0 = 常规优先级通道触发使能关闭
- 1 = 常规优先级通道触发使能打开

CCR4H: 高优先级通道的 T9_CCR1 触发使能

- 0 = 高优先级通道触发使能关闭
- 1 = 高优先级通道触发使能打开

CCR4N: 常规优先级通道的 T9_CCR1 触发使能

- 0 = 常规优先级通道触发使能关闭
- 1 = 常规优先级通道触发使能打开

CCR3H: 高优先级通道的 T9_CCR0 触发使能

- 0 = 高优先级通道触发使能关闭
- 1 = 高优先级通道触发使能打开

CCR3N: 常规优先级通道的 T9_CCR0 触发使能

- 0 = 常规优先级通道触发使能关闭

- 1 = 常规优先级通道触发使能打开
- CCR2H: 高优先级通道的 T6_CCR0 触发使能
- 0 = 高优先级通道触发使能关闭
- 1 = 高优先级通道触发使能打开
- CCR2N: 常规优先级通道的 T6_CCR0 触发使能
- 0 = 常规优先级通道触发使能关闭
- 1 = 常规优先级通道触发使能打开
- CCR1H: 高优先级通道的 T5_CCR1 触发使能
- 0 = 高优先级通道触发使能关闭
- 1 = 高优先级通道触发使能打开
- CCR1N: 常规优先级通道的 T5_CCR1 触发使能
- 0 = 常规优先级通道触发使能关闭
- 1 = 常规优先级通道触发使能打开
- CCR0H: 高优先级通道的 T5_CCR0 触发使能
- 0 = 高优先级通道触发使能关闭
- 1 = 高优先级通道触发使能打开
- CCR0N: 常规优先级通道的 T5_CCR0 触发使能
- 0 = 常规优先级通道触发使能关闭
- 1 = 常规优先级通道触发使能打开
- EOCIE: ADC 一次转换结束中断使能位
- 0 = 不使能 ADC 一次转换结束触发中断
- 1 = 使能 ADC 一次转换结束触发中断
- EOCIC: ADC 一次转换结束中断标志位清零位
- 0 = 不清零 EOC
- 1 = 清零 EOC
- EOCIF: ADC 一次转换结束中断标志位¹
- 0 = ADC 转换未开始或者转换未结束
- 1 = 至少一次 ADC 转换已结束²
- WDIE: ADC 模拟看门狗中断使能
- 0 = 禁止模拟看门狗中断
- 1 = 允许模拟看门狗中断
- HPENDIE: ADC 高优先级通道转换结束中断使能位
- 0 = 禁止高优先级通道转换结束中断
- 1 = 允许高优先级通道转换结束中断
- ENDIE: ADC 常规通道转换结束中断标志使能位
- 0 = 禁止 AD 转换结束中断
- 1 = 允许 AD 转换结束中断
- WDIC: ADC 模拟看门狗中断清零位
- 0 = 不清零模拟看门狗中断标志位
- 1 = 清零模拟看门狗中断标志位

¹ EOCIF 和 ENDIF, HPENDIF 是独立的, 不受后两者的影响。扫描模式下, 可以通过 EOC 判定每次 ADC 转换的情况。

² 每次转换结束后 (不区分常规通道或者高优先级通道), EOC 置 1, 读取常规通道或高优先级通道数据寄存器会清零该位, 也可以通过将 EOCIC 位置 1 清零。

- HPENDIC: ADC 高优先级通道转换结束中断清零位
 0 = 不清零高优先级通道转换结束中断标志位
 1 = 清零高优先级通道转换结束中断标志位
- ENDIC: ADC 常规通道转换结束中断清零位
 0 = 不清零 AD 常规通道转换结束中断标志位
 1 = 清零 AD 常规通道转换结束中断标志位
- WDIF: ADC 模拟看门狗中断标志位
 0 = 未发生模拟看门狗中断
 1 = 发生了模拟看门狗中断
- HPENDIF: ADC 高优先级通道转换结束中断标志位
 0 = 未发生高优先级通道转换结束中断
 1 = 发生了高优先级通道转换结束中断
- ENDIF: ADC 常规通道转换结束中断标志位
 0 = 未发生 AD 常规通道转换结束中断
 1 = 发生了 AD 常规通道转换结束中断
- HPSTARTIF: A/D 高优先级通道转换状态位。
 该位由硬件在注入通道组转换开始时设置，由软件清除。
 0 = 高优先级通道组转换未开始；
 1 = 高优先级通道组转换已开始。
- STARTIF: A/D 常规通道转换状态位。
 该位由硬件在注入通道组转换开始时设置，由软件清除。
 0 = 常规通道组转换未开始；
 1 = 常规通道组转换已开始。

17. 3. 14 ADC0_DELAY ADC 快速交叉延时寄存器

表 17-15 ADC0_DELAY ADC 快速交叉延时寄存器

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

FDELAY<5:0>: 快速交叉模式延时时间选择

- 000000 = 1 周期
- 000001 = 2 周期
- 000000 = 3 周期
-
- 000110 = 7 周期 (默认值)
-
- 001110 = 15 周期
- 001111 = 16 周期
- 010000 = 17 周期
-

111111 = 64 周期

17.4 ADC 电压引脚

为了提高 AD 转换精度,AD 有独立的电源供电,过滤和屏蔽外部电路板上的毛刺干扰。ADC 的独立电源引脚为 VDDA, 独立的电源地引脚为 VSSA。ADC 电源需要保证在 2.4V 到 3.6V 之间。

为了保证 ADC 工作的稳定,用户需要将 VREF-引脚接地,否则会出现 ADC 工作异常。

当转换参考电压选择 VREF+时,VREF+引脚上所接电压需要处于 2.4V 到 VDDA 之间。

在小封装的产品中, VREF+和 VREF-引脚会和通用 I/O 口复用, 在使用时, 同样须把 VREF-所在引脚接地。如无复用, 则代表 VREF-在内部和 VSSA 相连。

17.5 ADC 时钟和时序图

ADC 数字模块时钟与模拟模块时钟分别由 CLKS<1:0>和 ADCCLKS<2:0>控制, 通过 CLKS<1:0>可以选择 SCLK/HFCLK/LFCLK 作为 ADC 数字控制部分时钟, 通过 ADCCLKS<2:0>可以设置模拟工作时钟的频率, 该频率由 ADC 数字控制部分时钟分频得到, 范围为 $(1/2 \sim 1/64) T$, T 为数字模块工作时钟。

如下图 17-2, ADC 在使能后需要经过 5us 的时间以完成硬件初始化, 然后可以使能触发信号以启动转换(软件触发或外部触发)。触发后, ADC 需要等待 2 个 ADC 转换时钟, 以等待被采样信号稳定并完成采样。采样完成之后, 开始转换并在 14 个转换时钟周期后完成转换, 并将 EOC 标志位置 1, 转换的数据存放于 ADCx_DATA 或者 ADCx_HPDATAy 寄存器中。注意, 在两次转换的间隔最少为 17 个转换时钟, 因此两次触发的间隔至少需要大于 17 个转换时钟。

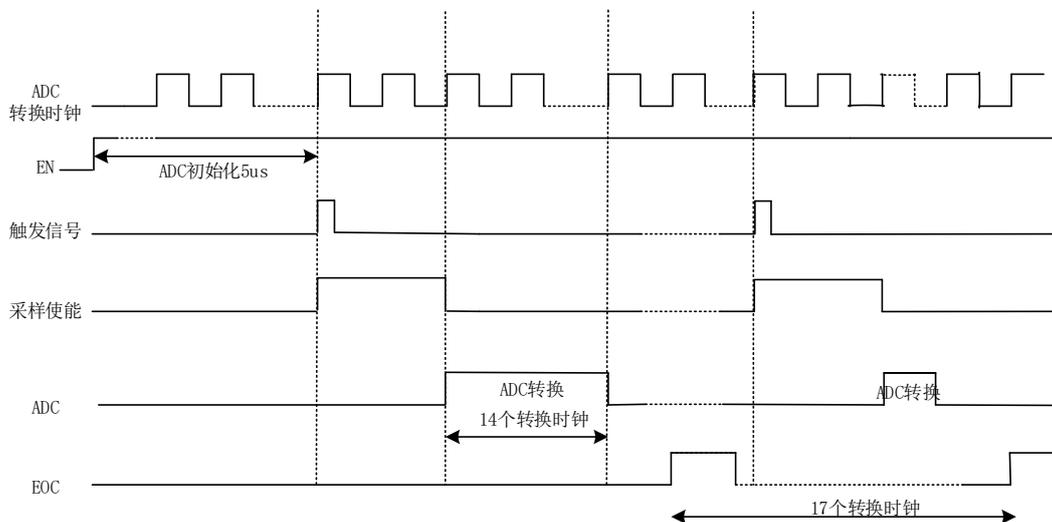


图 17-2 ADC 转换时序图

17.6 ADC 通道

ADC 支持两种通道模式: 常规通道和高优先级通道。这两种通道都支持任意 AD 通道顺序和可定义的通道数量配置。例如, 支持按顺序完成下列转换, ADCx_CH1, ADCx_CH3, ADCx_CH7, ADCx_CH6。

常规通道由最高 16 个通道组成。通过 ADCx_SCANSQy 可以选择需要的转换的通道数目和每个通道的输入源。通过 SLEN<3:0>选择通道的数量, 转换时按照通道 0、通道 1、通

道 2 的顺序转换,每个通道通过 SSQx<5:0>选择需要的输入 ADCx_CH 通道。如 SLEN<3:0>=0 时,只有通道 0 有效,SLEN<3:0>=1 时,通道 0 和通道 1 有效。

高优先级通道由 4 个通道组成,通过 ADCx_HSCANSQy 可以选择需要转换的通道数目和每个通道的输入源。通过 HSLEN<3:0>选择通道的数量,转换时按照通道 1、通道 2、通道 3、通道 4 的顺序转换,每个通道通过 HSSQx<5:0>选择需要的输入 ADCx_CH 通道。如 HSLEN<3:0>=0 时,只有通道 1 有效,HSLEN<3:0>=1 时,通道 1 和通道 2 有效。高优先级通道如抢占了常规通道的进程,高优先级转换结束之后,恢复之前被复位的常规通道转换。

ADC 还有 6 个内部通道,分别是:VREFINT、OPA0_OUT、OPA1_OUT、OPA3_OUT、AVDD 和 GND。内部通道和其他通道的使用方法相同。

17.7 ADC 结果

ADC 转换输出结果:

$$\text{ADC转换结果} = \frac{\text{通道输入电压}}{\text{参考电压}} \times 0xFFF \quad \text{公式 17-1}$$

常规通道转换结束后将结果保存在 ADCx_DATA 常规通道数据寄存器中。常规通道支持多个通道转换,但结果寄存器只有一个,如若上一次的结果没有及时被取出则会被下一次的结果覆盖。

高优先级通道转换结束后将结果保存在 ADCx_HPDATAy 高优先级通道数据寄存器中。每个高优先级通道都有独立的数据寄存器,可以保存对应的高优先级通道的转换结果。如若上一次的结果没有及时被取出则会被下一次的结果覆盖。

17.8 高优先级通道管理

当 HPAUTO 为 0,SCANEN 为 1 时,可以使用触发高优先级功能。

利用外部触发或者通过设置 ADCx_CTL0 寄存器的 START 位,启动一组常规通道的转换。如果在常规通道转换期间产生一个外部高优先级通道触发,当前转换被复位,高优先级通道序列被以单次扫描的方式进行转换。高优先级通道转换结束后恢复上次被中断的常规通道转换。如果在高优先级通道转换期间产生一个常规通道触发事件,高优先级通道不会被打断,而常规通道则会在高优先级通道转换结束后被执行。

(注意:在使用触发的高优先级通道转换时,必须保证触发事件的时间间隔长于高优先级通道序列。)

当 HPAUTO 为 1 时,SCANEN 为 1 时,在常规通道转换之后,高优先级通道被自动转换。这可以让转换序列多至 20 个。在此模式中,需要将高优先级通道的外部触发禁止。此时如果 CONT 为 1,常规通道至高优先级通道的转换序列被连续执行。(注意该模式下不能使用间隔模式)

17.9 间隔模式

通过 ADCx_CTL0 寄存器的 DSEN 和 HPDSEN 位可以设置常规通道和高优先级通道的间隔扫描模式。

- 常规通道

常规通道间隔扫描模式可以用来执行一个短序列的 n 次转换 (n<=8),此转换是

ADCx_SCANSQy (y=0,1,2,3) 序列的一部分。数值 n 由 ADCx_CTL0 寄存器的 DSNUM<2:0> 给出。一个外部触发信号可以启动 ADCx_SCANSQy (y=0,1,2,3) 寄存器中描述的下一轮 n 次的转换，直到此序列所有的转换完成为止。总的扫描序列长度由 ADCx_SCANSQ2 寄存器的 SLEN<3:0> 决定。

- 例：n=3，被转换的通道为 0,1,2,3,7,8,9,10
- 第一次触发：0,1,2 被转换
- 第二次触发：3,7,8 被转换
- 第三次触发：9,10 被转换，并产生结束信号
- 第四次触发：0,1,2 被转换

高优先级通道

高优先级通道间隔扫描模式可以按顺序逐个转换 ADCx_HSCANSQ 寄存器选中的序列。一个外部触发信号可以启动 ADCx_HSCANSQ 寄存器中描述的下一个通道的转换，直到此序列所有转换完成为止。总的扫描序列长度由 ADCx_HSCANSQ 寄存器的 HSLEN<1:0> 决定。

- 例：被转换的通道为 1,2,3
- 第一次触发：1 被转换
- 第二次触发：2 被转换
- 第三次触发：3 被转换，并产生结束信号
- 第四次触发：1 被转换

17.10 ADC 数据对齐

通过 ADCx_CTL0 寄存器的 LR 位可以设置转换后的数据储存的对齐方式。数据可以实现低 16 位空间中左对齐或右对齐。

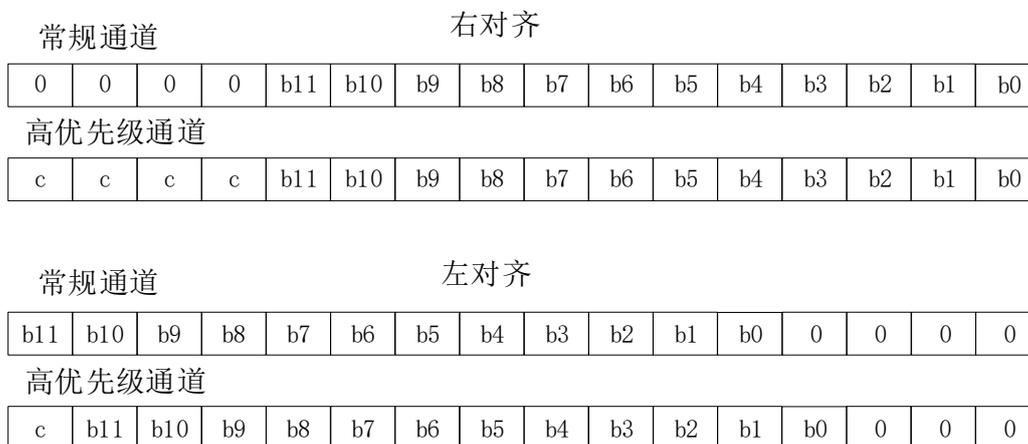


图 17-3 数据对齐方式

常规通道，右对齐时低 12 位存放转换数据，高 4 位补 0；左对齐时，高 12 位存放转换数据，低 4 位补 0。

高优先级通道，右对齐时低 12 位存放转换数据，高 4 位补进位标志位 C；左对齐时，最高位是进位标志位 C，然后依次填充转换数据，低 3 位补 0。

进位标志位 C 由高优先级通道 AD 转换的数值减高优先级通道数据偏移量 HPDOFF<11:0> 得到，如果 C=1，说明设置的偏移值大于转换的数值，建议用户重新调整高优先级通道数据偏移量 HPDOFF<11:0>。(x=0,1,2;y=0,1,2,3)

17.11 ADC 单次转换模式

当 ADC 处于单次转换模式时，ADC 只执行一次转换。该模式可以通过 ADCx_CTL0 控制寄存器的 CONT 位控制，0 为单次转换模式。EN 使能，软件将 START 或 ADCx_HSTART 置 1，或者外部触发使能且有外部触发信号时，开始单次转换。

转换结束后，常规通道和高优先级通道分别将转换结果保存在 ADCx_DATA 和 ADCx_HDATAy 中。当使能了 ENDIE 或 ENDIF 时，相应通道转换结束，则会产生中断。然后 ADC 停止工作。(y=0,1,2,3)

17.12 ADC 连续转换模式

当 ADC 处于连续转换模式时，当 AD 转换一结束马上就启动下一次 AD 转换。该模式可以通过 ADCx_CTL0 控制寄存器的 CONT 位控制，1 为连续转换模式。EN 使能，软件将 START 或 ADCx_HSTART 置 1，或者外部触发使能且有外部触发信号时，开始转换。

一个通道转换结束后，常规通道和高优先级通道分别将转换结果保存在 ADCx_DATA 和 ADCx_HDATAy 中。当使能了 ENDIE 或 ENDIF 时，相应通道转换结束，则会产生中断。ADC 会持续转换。(y=0,1,2,3)

17.13 ADC 扫描模式

通过 SCANEN 位可以设置扫描模式。一旦该位置 1，ADC 扫描所有 ADCx_SCANSQy (y=0,1,2) (常规通道) 或 ADCx_HSCANSQ (高优先级通道) 选中的所有通道。在每个组的每个通道上执行单次转换，在每次转换结束后，同一组的下一个通道被自动转换。如果 CONT 位置 1，转换不会在被选择的通道组的最后一个通道上停止，而是继续再从被选择的通道组的第一个通道继续转换。

如果设置了 DMA，在每次结束后，DMA 控制器把通道的转换数据传输到指定位置中。

17.14 ADC 模拟看门狗

模拟看门狗允许在一个或多个通道中工作。其中，当 WDSCH 为 0 时，表明所有通道都可以使用模拟看门狗功能，此时当 WDEN 为 1 时，常规通道所有通道都使用模拟看门狗功能；当 HPWDEN 为 1 时，高优先级通道所有通道都使用模拟看门狗功能。而当 WDSCH 为 1 时，表明单一通道使用模拟看门狗功能，这个通道由 WDCH<5:0>设置，需要注意的是，如果 WDCH<5:0>选择的是常规通道，需要将 WDEN 位值 1；如果 WDCH<5:0>选择的是高优先级通道，需要将 HPWDEN 位值 1。

当使能 ADC 的模拟看门狗功能时，如果输入 ADC 通道的转换电压低于模拟看门狗的低阈值或高于模拟看门狗的高阈值时，ADC 模拟看门狗状态位 WDIF 会被置 1，该位由软件清零。若 ADCx_STATE 寄存器的 WDIE 位置 1，则允许产生相应的中断。模拟看门狗的高低阈值分别保存在 ADCx_WDH 和 ADCx_WDL 中。

17.15 ADC 外部触发

AD 转换可以由外部事件触发。通过 EXTEN 和 HPEXTEN 可以使能常规通道和高优先级通道的外部触发功能。通过 ADCx_CTL1 寄存器的 NCH<6:0>和 HCH<6:0>位域选择触发

AD 的外部触发的事件。或者通过 ADCx_STATE 寄存器选择高级定时器触发 AD。

以高级定时器 Tx 触发 AD 为例，当使用 Tx，可以使用自动触发 AD 的功能。AD 触发功能有两种形式，一种是通过 Tx 溢出信号触发，一种是通过 Tx_CCRy (x=5;y=0,1) 寄存器与 Tx 匹配时产生触发，这种方法可在周期中的任意位置触发 AD。

以 Tx 触发 AD 为例，设置寄存器 Tx_CTL，选择合适的计数模式，通过 TXOFS<1:0> 选择 Tx 的上溢/下溢触发 AD，通过 TXADAUTO 使能 Tx 溢出触发 AD。TXADAUTO 信号在完成一次 AD 转换后自动清零。

还可以设置 Tx_CCRy (y=0,1) 寄存器，当 Tx_CCRy 与 Tx 匹配时，也会产生 AD 触发信号，将 ADSTART 位置 1。

注意：使能 AD 后，延迟大概 5us 后，再去给触发信号去触发 AD 转换。

17.16 DMA 请求

AD 带有 DMA 功能。通过 ADCx_CTL0 寄存器的 NDMAEN 位可以使能常规通道转换的 DMA 请求。使能常规通道上的 DMA 请求后，在 ADC 常规通道转换一次结束后，EOCIF 标志位置 1，就会产生一个 DMA 请求。

ADC 模块有 4 个可配置的高优先级通道，每个高优先级通道都带有一个单独的 DMA 请求功能。通过 ADCx_CTL0 寄存器的 HP0DMAEN/HP1DMAEN/HP2DMAEN/HP3DMAEN 位可以分别使能 4 个高优先通道的 DMA 请求。使能高优先级通道上的 DMA 请求后，在对应 ADC 高优先级通道转换一次结束后，EOCIF 标志位置 1，就会产生一个对应通道的 DMA 请求。换言之，ADCxHPDATA0/1/2/3 分别保存着 4 个高优先级通道的转换结果，如果 4 个高优先级通道都使能了 DMA 请求功能，则 ADCxHPDATA0/1/2/3 发生更新时，就会发生对应通道的 DMA 请求。

常规通道的存储只有一个数据寄存器，所以在连续转换多个常规通道时，应优先考虑使用 DMA 避免转换数据丢失。

17.17 内部参考电压

内部参考电压缓冲器可以提供 ADC、DAC 的参考电压，也可以通过 VREF+ 引脚提供给外部元器件使用。

内部参考电压缓冲器可以提供 1.5V、2.0V、2.5V、3V 四种参考电压选项¹。

当不使用内部参考电压缓冲器的时候，也可以通过 VREF+ 引脚引入外部参考电压。

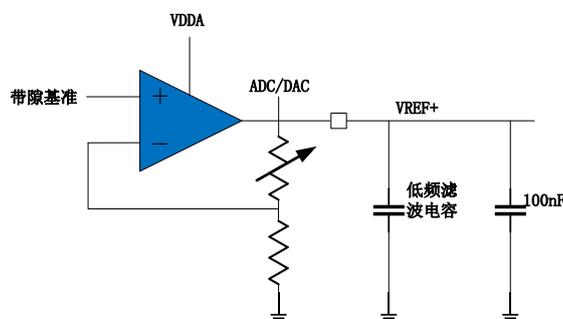


图 17-4 电压参考缓冲器

通过 ADCx_CTL1 寄存器的 ADCx_REF0、ADCx_REF1 和 ADCx_REF2 位，分别配置 VDDA、外部 VREF+ 或者内部参考电压作为 ADCx 的参考电压。

¹ 在使用内部电压参考缓冲器时，必须要有 VREF+ 引脚。

通过 PM_CTL1 寄存器的 VREFSEL<1:0>位, 可以选择内部参考电压为 1.5V、2V、2.5V 或 3V。

17.18 内部信号通道

ADC 模块有 3 个内部信号通道, 分别是: VREFINT、AVDD 和 GND。这些通道由扫描序列设置寄存器进行选择。

表 17-16 内部信号通道

内部信号	ADC 通道
AVDD	ADCx_CH50
GND	ADCx_CH51
VERFINT	ADCx_CH53

17.19 ADC 中断

常规通道和高优先级通道转换结束后能产生中断, 当模拟看门狗状态位被设置时也能产生中断, 每个中断都有独立的中断使能。并且, 有专门的 EOC_IF 指示每次 AD 转换是否结束。例如, 扫描模式下, 常规通道只有在所有设置的通道被扫描执行完毕后, 常规通道转换结束标志位才置 1, 但是, EOC_IF 在每次 AD 转换结束后都会置 1。

扫描模式下, 只有当所有设置的通道扫描完成之后, ADC 的 ENDIF 或者 HPENDIF 置 1。间隔模式下, 也是将所有通道扫描执行完成后 ADCENDIF 或者 HPENDIF 为才置 1。

ADC 中断逻辑如下图所示:

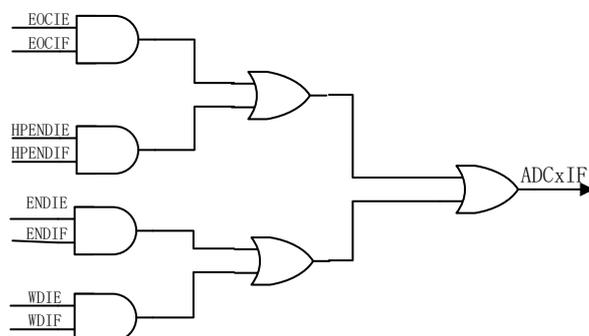


图 17-5 ADC 中断逻辑图

17.20 双 ADC 模式

双 AD 模式是相对于独立模式的, 在独立模式下各个 ADC 单独工作, 双 ADC 模式下, ADC0 和 ADC1 配合起来工作。

有两个以上 ADC 模块的产品中, 可以使用双 ADC 模式。通过 ADC0_CTL1 控制寄存器的 ADC0_DOUMOD<3:0>位可以选择双 ADC 工作模式, 启动双 ADC 模式后, 使用 ADC0 和 ADC1 工作。

双 ADC 共有 4 种工作模式:

- 高优先级通道同步模式
- 常规通道同步模式
- 快速交叉模式

- 交替触发模式

还可以使用下列方式组合使用上面的模式：

- 混合(高优先级通道同步模式+常规通道同步模式)
- 混合(常规通道同步模式+交替触发模式)
- 混合(高优先级通道同步模式+交叉模式)

注意：

- (1) 在双 ADC 模式中，当转换配置成由外部事件触发时，用户必须将其设置成仅触发主 ADC(ADC0)，同时将从 ADC (ADC1) 的外部触发屏蔽，这样可以防止意外的触发转换。
- (2) 混合模式是指：在常规通道转换执行时，插入高优先级通道转换，此时 ADC 被强制执行高优先级通道的转换，高优先级通道转换完成后，恢复之前被中断的常规通道转换。所有的混合模式必然包含常规通道和高优先级通道转换。

17. 20. 1 高优先级通道同步模式

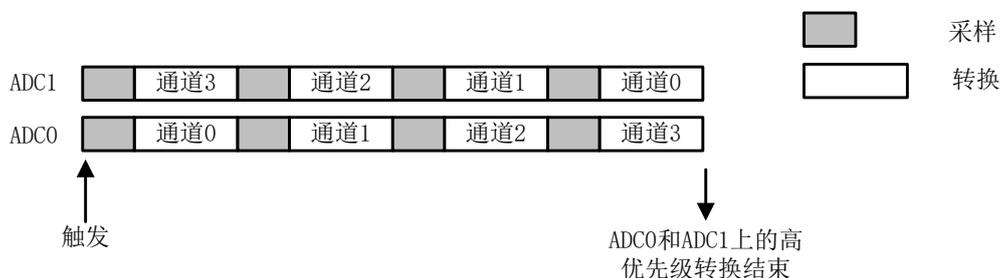


图 17-6 高优先级通道同步模式

ADC0 上的高优先级通道的触发信号，会同时触发 ADC0 和 ADC1 上的高优先级通道转换。

注意：

- (1) 为了避免在转换同一个通道时采样时间重叠，不要在两个 ADC 上转换两个相同通道
- (2) 同步模式必须转换具有相同长度的序列，或者保证触发间隔比两个中较长的序列长，否则当较长的序列还没有转换完成时，具有较短的序列的 ADC 可能会被重新启动。

17. 20. 2 常规通道同步模式

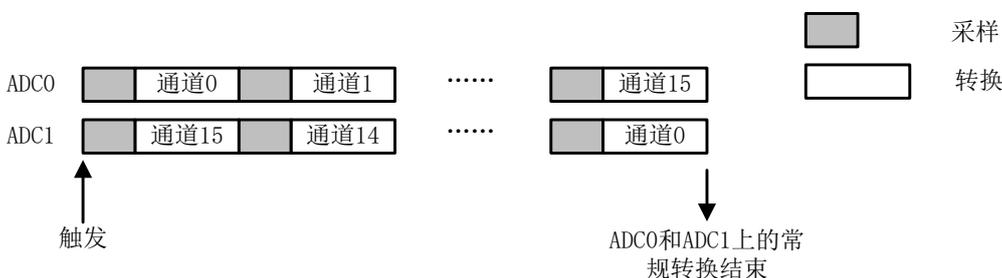


图 17-7 常规通道同步模式

ADC0 上的常规通道的触发信号，会同时触发 ADC0 和 ADC1 上的常规通道转换。

注意：

- (1) 为了避免在转换同一个通道时采样时间重叠，不要在两个 ADC 上转换两个相同通道
- (2) 同步模式必须转换具有相同长度的序列，或者保证触发间隔比两个中较长的序列长，

否则当较长的序列还没有转换完成时,具有较短的序列的 ADC 可能会被重新启动。

17. 20. 3 快速交叉模式

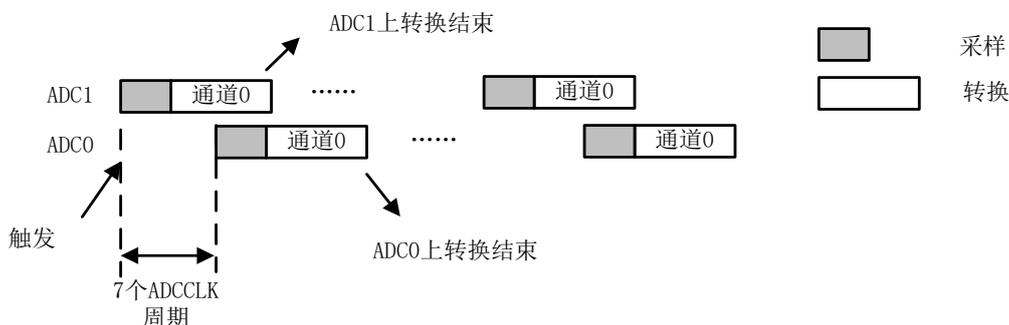


图 17-8 快速交叉模式

快速交叉只适用于一个常规通道,可设置单次触发和连续触发。通常不开启扫描模式。延时时间间隔可以通过 ADC0_DELAY 寄存器中的 FDELAY<5:0>来设定,默认延时时间为 7 个 ADCCLK。这里以 7 个 ADCCLK 做说明。ADCCLK 是 ADC 模拟部分的工作时钟。触发产生后,ADC1 立即开启通道转换,并且,ADC0 在延时 7 个 ADCCLK 后开启通道转换。

同样,要避免在转换同一个通道时采样时间重叠,因此要合理选择延时时间。

17. 20. 4 交替触发模式

交替触发只适用于高优先级通道,可以设置 HPDSEN 来实现无间隔交替和有间隔交替两种模式。

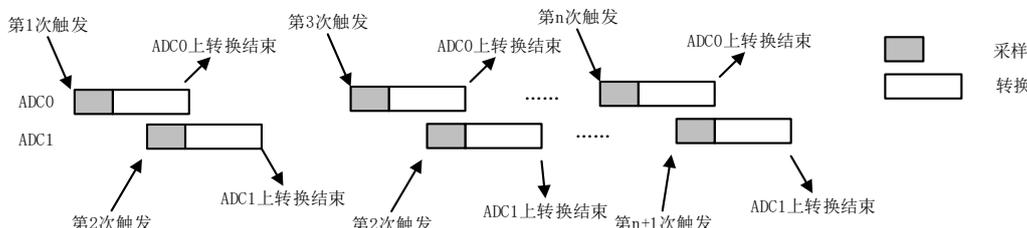


图 17-9 交替触发-无间隔交替模式

无间隔模式下,第一次触发后,ADC0 上设定的所有通道被依次执行 AD 转换,后 ADC0 上转换结束。第二次触发后,ADC1 上设定的所有通道被依次执行 AD 转换,后 ADC1 上转换结束。第三次触发后,ADC0 上设定的所有通道被再依次执行 AD 转换,后 ADC0 上转换结束,如此往复。

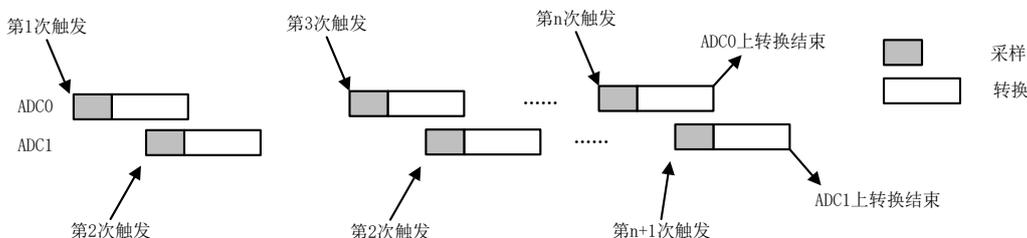


图 17-10 交替触发-有间隔交替模式

有间隔模式下,第一次触发后,ADC0 上设定的通道中第 1 个通道被执行 AD 转换,ADC0 上设定的所有通道并没有被全部转换。第二次触发后,ADC1 上设定的通道中第 1 个被执行 AD 转换,ADC1 上设定的所有通道并没有被全部转换。第三次触发后,ADC0 上设

定的通道中第 2 个通道被执行 AD 转换，如此按顺序执行，直到 ADC0 上所有设定的通道都执行完成 AD 转换，ADC0 上的转换才结束，ADC1 亦是如此。

交替触发只能采用外部触发。

17. 20. 5 混合(常规通道同步+高优先级通道同步模式)

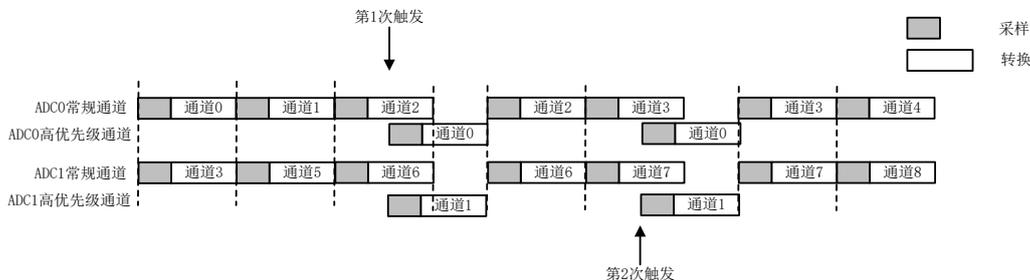


图 17-11 混合常规通道同步+高优先级通道同步模式

混合常规通道同步+高优先级通道同步模式必须转换具有相同长度的序列，或者保证触发间隔比两个中较长的序列长，否则当较长的序列还没有转换完成时，具有较短的序列的 ADC 可能会被重新启动。

常规通道 AD 转换被中断，转而执行高优先级通道 AD 转换，高优先级通道 AD 转换结束后，重新开始之前被中断的常规通道 AD 转换。

17. 20. 6 混合(常规通道同步+交替触发模式)

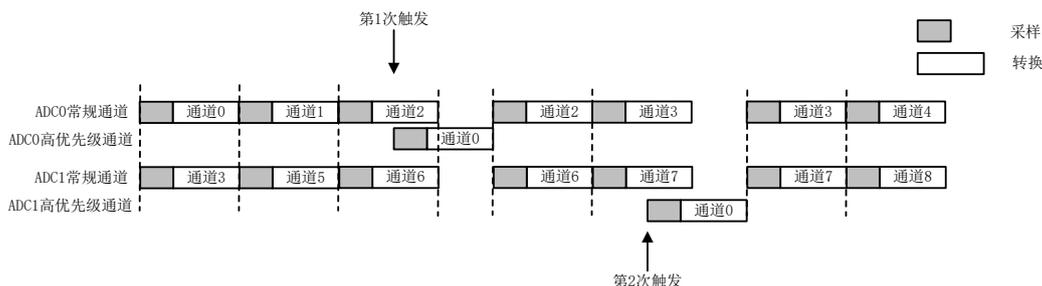


图 17-12 混合常规通道同步+交替触发模式 1

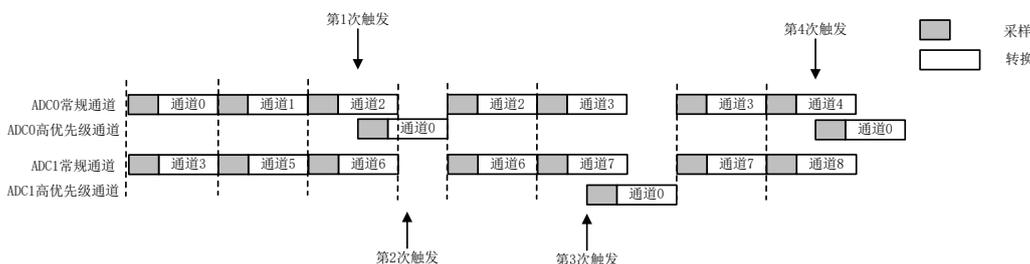


图 17-13 混合常规通道同步+交替触发模式 2

混合常规通道同步+交替触发模式必须转换具有相同长度的序列，或者保证触发间隔比两个中较长的序列长，否则当较长的序列还没有转换完成时，具有较短的序列的 ADC 可能会被重新启动。

常规通道 AD 转换被中断，转而执行高优先级通道 AD 转换，高优先级通道 AD 转换结束后，重新开始之前被中断的常规通道 AD 转换

如果在高优先级通道执行过程中产生高优先级通道触发事件，那么该事件会被忽略。

此时不能使能间隔模式。

17. 20. 7 混合(高优先级通道同步+交叉模式)

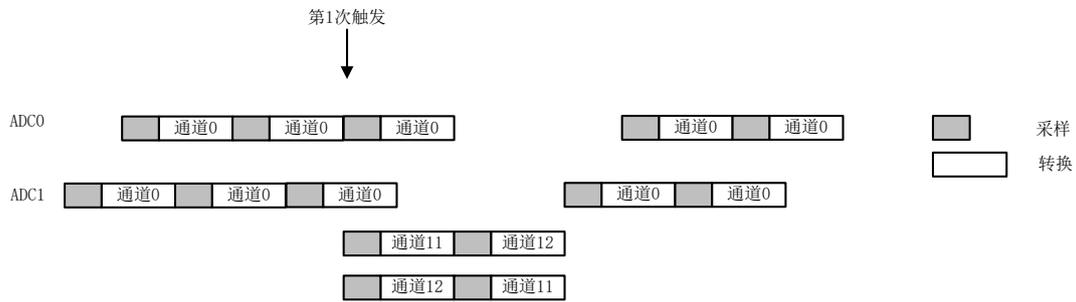


图 17-14 混合高优先级通道同步交叉模式

常规通道 AD 转换被中断，转而执行高优先级通道 AD 转换，高优先级通道 AD 转换结束后，重新开始之前被中断的常规通道 AD 转换。

这里注意：重新开始后的常规通道转换，ADC0 依旧会延时 ADC1 一定的周期数。

不能使能连续转换 CONT。

18 数模转换器模块 (D/A)

18.1 概述

DA 特性:

- 2 个 12 位 DAC
- 12 位 DAC 支持 DMA 功能
- 12 位 DAC 支持外部触发转换
- 12 位 DAC 支持噪声波发生器
- 12 位 DAC 支持三角波发生器
- 独立的外部参考电压源

18.2 数模 (D/A) 转换器模块原理框图

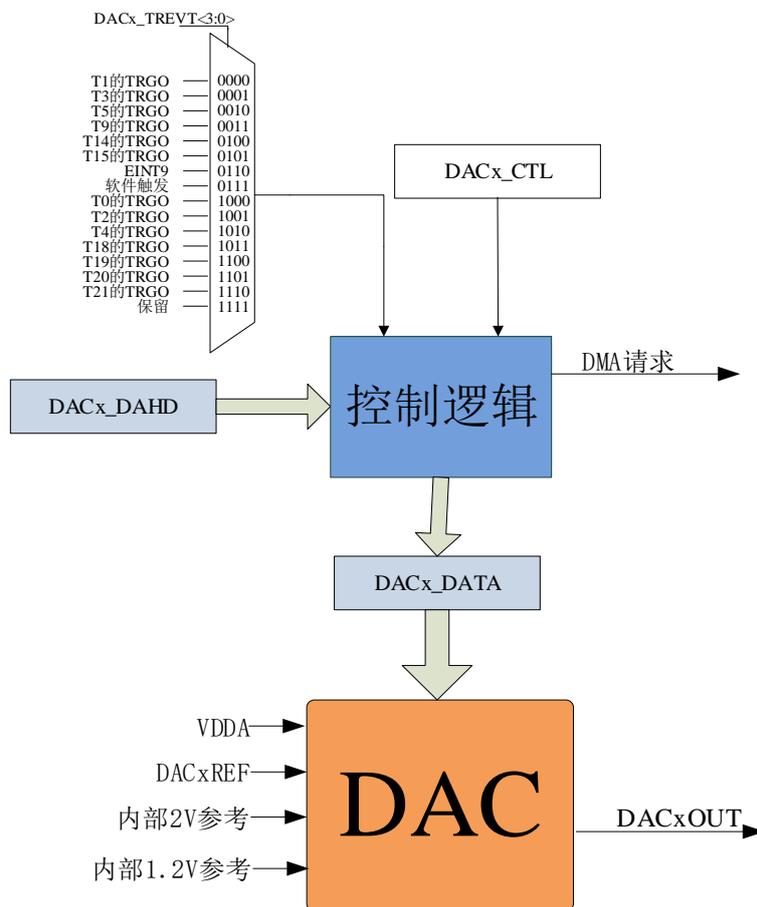


图 18-1 数模 DACx 转换器框图

注: 上图中 $x=0,1$, `VDDA`、`DACxREF`、内部 2V 参考, 内部 1.2V 参考都是参考电压, 详细说明在“18.4”小节, `DACxOUT` 为 DAC 的模拟输出。

18.3 DACx 相关寄存器

 表 18-1 DACx 相关寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x000	DACx_CTL	R/W	DACx 控制寄存器	0x0000 0000
0x004	DACx_DAHD	R/W	DACx 数据保持寄存器	0x0000 0000
0x008	DACx_DATA	R	DACx 输出数据寄存器	0x0000 0000
0x00C	DACx_CTL1	R/W	DACx 控制寄存器 1	0x0000 0000
0x014	DACx_CAL	R/W	DACx 校验寄存器	0x0000 0000

DAC0 基地址: 0x4000 0700;

DAC1 基地址: 0x4000 0780;

18.3.1 DACx_CTL DACx 控制寄存器

表 18-2 DACx_CTL DACx 控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R																R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	DF	SWTRIG	MAS <3:0>			WAVE <1:0>		TREV <3:0>			TRIGEN		DE	EN			

DF: DACx 通道触发 DMA 中断标志 (DMA 响应后硬件自动清零标志位)

0 = DACx 通道未触发 DMA

1 = DACx 通道触发 DMA

SWTRIG: DACx 软件触发 (该位由硬件自动清零)

0 = 禁止 DACx 软件触发

1 = 使能 DACx 软件触发

MAS<3:0>: DACx 屏蔽/幅值选择器

噪声生成模式:

0000 = 不屏蔽 LFSR 位<0>

0001 = 不屏蔽 LFSR 位<1:0>

0010 = 不屏蔽 LFSR 位<2:0>

0011 = 不屏蔽 LFSR 位<3:0>

0100 = 不屏蔽 LFSR 位<4:0>

0101 = 不屏蔽 LFSR 位<5:0>

0110 = 不屏蔽 LFSR 位<6:0>

0111 = 不屏蔽 LFSR 位<7:0>

1000 = 不屏蔽 LFSR 位<8:0>

1001 = 不屏蔽 LFSR 位<9:0>

1010 = 不屏蔽 LFSR 位<10:0>

1011 = 不屏蔽 LFSR 位<11:0>

11xx = 不屏蔽 LFSR 位<11:0>

¹ 本章除特殊说明外, x=0,1

注：LFSR(Linear Feedback Shift Register)线性反馈移位寄存器

三角波生成模式：

0000 = 三角波幅值等于 1

0001 = 三角波幅值等于 3

0010 = 三角波幅值等于 7

0011 = 三角波幅值等于 15

0100 = 三角波幅值等于 31

0101 = 三角波幅值等于 63

0110 = 三角波幅值等于 127

0111 = 三角波幅值等于 255

1000 = 三角波幅值等于 511

1001 = 三角波幅值等于 1023

1010 = 三角波幅值等于 2047

1011 = 三角波幅值等于 4095

11xx = 三角波幅值等于 4095

WAVE<1:0>: DACx 波形发生器使能

00 = 关闭波形发生器

01 = 使能噪声发生器

1x = 使能三角波发生器

TREVT<3:0>: DACx 通道触发事件选择

0000 = 定时器 1 的 TRGO 事件

0001 = 定时器 3 的 TRGO 事件

0010 = 定时器 5 的 TRGO 事件

0011 = 定时器 9 的 TRGO 事件

0100 = 定时器 14 的 TRGO 事件

0101 = 定时器 15 的 TRGO 事件

0110 = 外部中断线 EINT9 中断触发

0111 = 软件触发

1000 = 定时器 0 的 TRGO

1001 = 定时器 2 的 TRGO

1010 = 定时器 4 的 TRGO

1011 = 定时器 18 的 TRGO

1100 = 定时器 19 的 TRGO

1101 = 定时器 20 的 TRGO

1110 = 定时器 21 的 TRGO

1111 = 保留

TRIGEN: DACx 通道触发使能

0 = 禁止触发事件触发 DACx

1 = 使能触发事件触发 DACx

DE: DACx 通道 DMA 使能

0 = 禁止 DACx 的 DMA 模式

1 = 使能 DACx 的 DMA 模式

EN: DACx 使能。

0 = 禁止 DACx

1 = 使能 DACx

18.3.2 DACx_DAHD DACx 数据保持寄存器

表 18-3 DACx_DAHD DACx 数据保持寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																		
R/W																					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																					DAHD<11:0>																													

DAHD<11:0>: DACx 数据保持寄存器。

18.3.3 DACx_DATA DACx 输出数据寄存器

表 18-4 DACx_DATA DACx 输出数据寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																		
R/W																					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																					DATA<11:0>																													

DATA<11:0>: DACx 输出数据寄存器。

18.3.4 DACx_CTL1 DACx 控制寄存器 1

表 18-5 DACx_CTL1 DACx 控制寄存器 1

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																		
R/W																					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																					CLKDIV <2:0>				BUFEN	CLK <1:0>		RFS <1:0>	SEL																					

CLKDIV<2:0>: DAC 工作时钟分频控制

- 000 = 1 分频
- 001 = 1/2 分频
- 010 = 1/4 分频
- 011 = 1/8 分频
- 100 = 1/16 分频
- 101 = 1/32 分频
- 110 = 1/64 分频
- 111 = 1/128 分频

BUFEN: DACx 输出缓冲使能

0 = DACx 输出无缓冲

1 = DACx 输出带缓冲

CLK<1:0>: DACx 工作时钟选择位

00 = 选用 SCLK 时钟为 DACx 工作时钟

01 = 选用 HFCLK 时钟为 DACx 工作时钟

1x = 选用 LFCLK 时钟为 DACx 工作时钟

RFS<1:0>: DACx 参考电压选择位

00 = DACx 选择 VDDA 电压作为参考电压

01 = DACx 选择外部参考电压 DAC_REF 作为参考电压

10 = 内部 1.2V 参考

11 = 内部 2V 参考

SEL: DACx 输出 IO 口选择位

0 = DACx 输出到 IO 口 0

1 = DACx 输出到 IO 口 1

18.3.5 DACx_CAL DACx 校验寄存器

表 18-6 DACx_CAL DACx 校验寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W											R	R/W															
位名	PPOSITRIM <3:0>			PNEGATRIM <3:0>			NPOSITRIM <3:0>			NNEGATRIM <3:0>			BUFTRIM											TRIM_CMP_IF	TRIM_CMP_EN										

PPOSITRIM<3:0>: DAC 模块 BUFFER 输入 P 管正失调方向校准位, 8421 权重控制。

PNEGATRIM<3:0>: DAC 模块 BUFFER 输入 P 管负失调方向校准位, 8421 权重控制。

NPOSITRIM<3:0>: DAC 模块 BUFFER 输入 N 管正失调方向校准位, 8421 权重控制。

NNEGATRIM<3:0>: DAC 模块 BUFFER 输入 N 管负失调方向校准位, 8421 权重控制。

BUFTRIM: DAC 模块 BUFFER 失调校准使能位

0 = 失调校准禁止

1 = 失调校准使能

TRIM_CMP_IF: DAC 模块校准结果比较标志位

0 = 初始时需要正向校准

1 = 初始时需要负向校准

注:校准过程中发生标志位翻转着说明当前校准完成

TRIM_CMP_EN: DAC 模块比较校准使能位

0 = 校准禁止

1 = 校准使能

18.4 DACxREF 参考电压

DACx 有 4 个参考电压，VDDA/DACx_REF/内部 2V 参考/内部 1.2V 参考，和 2 个输出引脚 DACx_OUTy。其中，VDDA 为模拟外设供电电源过电阻后电压，DACx_REF 为外部参考电压。（DAC0_REF 为 (PA5)引脚，DAC1_REF 为(PA6)引脚，具体安排看 IO 口分配）

注：y=0,1

18.5 DACx 转换

DACx 输出电压计算公式如下：

$$V_{out} = V_{in} \times \frac{DATA <11:0>}{4095} \quad \text{公式 18-1}$$

任何输出到 DAC 通道 x 的数据都必须写入 DACx_DAHD 寄存器，而不能直接对寄存器 DACx_DATA 写入数据。如果没有选中 DAC 触发（即 TRIGEN 置 0），存入寄存器 DACx_DAHD 的数据会在一个工作周期后自动载入寄存器 DACx_DATA。如果选中 DAC 触发（即 TRIGEN 置 1），数据会在触发之后 3 个工作周期自动载入到寄存器 DACx_DATA。一旦数据从 DACx_DAHD 寄存器载入到 DACx_DATA 寄存器，在经过一段时间后，DAC 输出即有效，这段时间的长短依电源电压和模拟输出负载的不同会有所变化。

DACx 可选择带缓冲输出，用来减少输出阻抗，无需外部运放就可以直接驱动外部负载。通过 DACx_CTL1 的 SEL 使能输出到 IO 口，通过 BUFEN 可以使能是否带缓冲输出。

18.6 DACx 触发事件选择

如果 EN 位置 1，DACx 可以通过 TRIGEN 使能触发事件，通过 TREVT<3:0>可以选择触发事件。当选择软件触发时，通过 DACx_SWTRI 位可以用于软件触发的使能和禁止。

表 18-7 DAC 触发事件

DAC_TREVT<3:0>	触发源	类型
0000	定时器 1 的 TRGO 事件	内部信号
0001	定时器 3 的 TRGO 事件	
0010	定时器 5 的 TRGO 事件	
0011	定时器 9 的 TRGO 事件	
0100	定时器 14 的 TRGO 事件	
0101	定时器 15 的 TRGO 事件	
0110	外部中断线 EINT9 中断	外部引脚
0111	软件触发	软件控制
1000	定时器 0 的 TRGO 事件	内部信号
1001	定时器 2 的 TRGO 事件	
1010	定时器 4 的 TRGO 事件	
1011	定时器 18 的 TRGO 事件	
1100	定时器 19 的 TRGO 事件	
1101	定时器 20 的 TRGO 事件	
1110	定时器 21 的 TRGO 事件	
1111	保留	

以定时器 T14 的 TXEN 作为 TRGO 输出为例：下面是其主要配置步骤：

- 配置 T14 的主模式, TXMMS<1:0>=01, T14 使能信号 TXEN 作为触发输出 TRGO。
- 配置 T14 的触发输入源, TXTS<3:0>=1100, T14CK 的输入。
- 配置 T14 的从模式, TXSMS<2:0>=100, 触发模式。
- 配置 T14 的主从模式同步位, TXMSSYNC=1。
- 配置 T14_CTL1 的 TXEN=1 使能定时器 T14。

18.7 DMA 请求

DACx 带有 DMA 功能。通过 DACX_CTL 寄存器的 DE 位可以使能 DMA 请求。使能 DACx 的 DMA 请求后，一旦有外部触发（而不是软件触发），此时 DACX_CTL 寄存器的 DF 位置 1，产生一个 DMA 请求，如果 DMA 响应完成该请求后，硬件将自动清零 DF 标志位。

DMA 请求不会累计，如果第 2 个外部触发发生在第 1 个外部触发响应之前，则不会处理第 2 个 DMA 请求。

18.8 噪声发生器

通过内部噪声发生器可以产生幅度变化的伪噪声。通过 WAVE<1:0>位可以选择 DACx 噪声生成功能。寄存器 LFSR 的预载入值为 0xAAA，按照特定的算法，在每次触发事件后更新该寄存器的值。

通过 DACx_CTL 寄存器的 MAS<3:0>位可以屏蔽部分或者全部的噪声发生器 12 位数据寄存器的值，这样得到的值与 DACx_DAHD 的值相加，去掉溢出之后得到的值写入 DACx_DATA 寄存器中。将 WAVE<1:0>置 0 可以复位噪声发生器的寄存器。

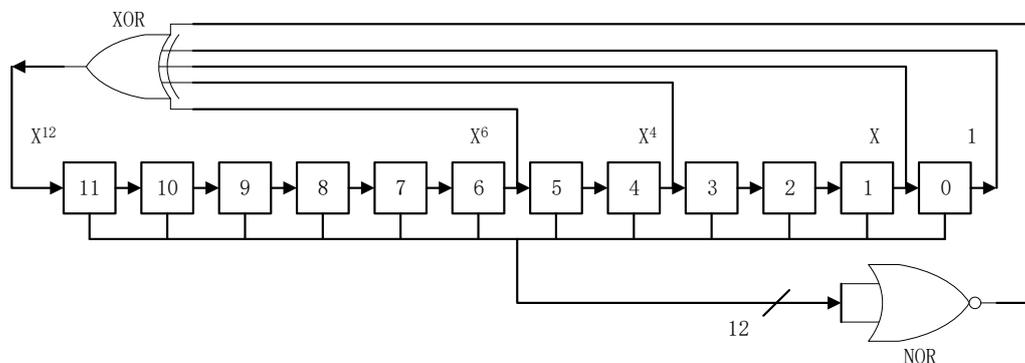


图 18-2 噪声发生器

如下图:

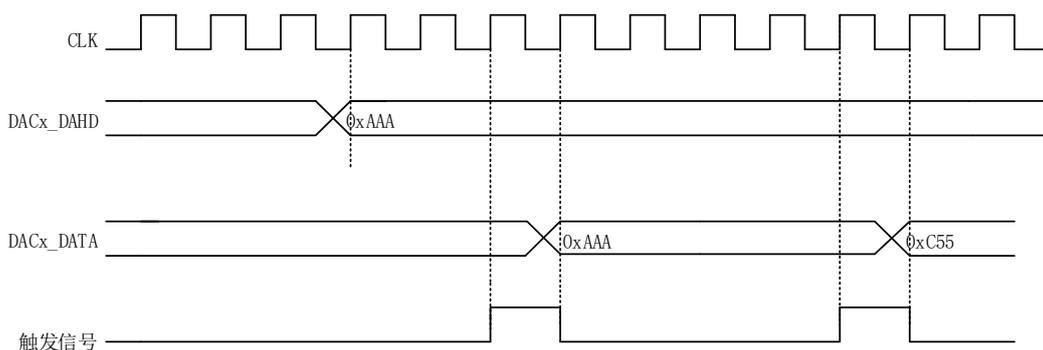


图 18-3 LFSR 产生波形的 DAC 转换

注：为了产生噪声，必须使能 DAC 触发，即设 TRIGEN 位为 1。

18.9 三角波发生器

三角波发生器允许在 DACx 缓慢变化的信号上加一个小幅度的三角波。通过 DACx_CTL 寄存器的 WAVE<1:0>位可以选择 DACx 三角波生成功能。通过 DACx_CTL 的 MAS<3:0>位可以设置三角波的幅度。内部的三角波计数器初始值为 0，在每次触发事件之后累加 1。计数器的值与 DACx_DAHD 的值相加，得到的和值如果没有溢出（超过 0xFFF），则直接写入 DACx_DATA 寄存器中，如果溢出了，则将和值减去 0xFFF 后再写入 DACx_DATA 寄存器中。

三角波计数器的值与 DACx_DAHD 的值相加，如果小于 DAC_MAS<3:0>最大幅值与 DAC_DAHD 基值之和，则三角波计数器逐步累加。等到相等时，三角波计数器开始递减，达到 0 后再开始累加，周而复始。将 WAVE<1:0>置 0 可以复位三角波发生器的寄存器。

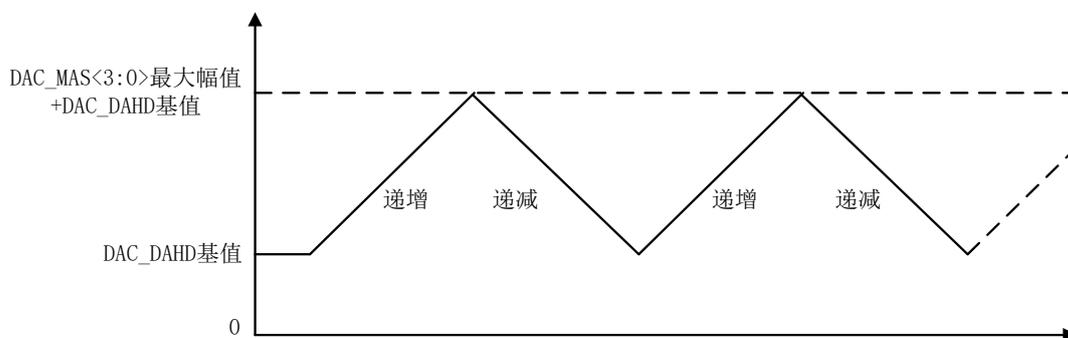


图 18-4 三角波发生器

如下图：为生成三角波波形的 DAC 转换。

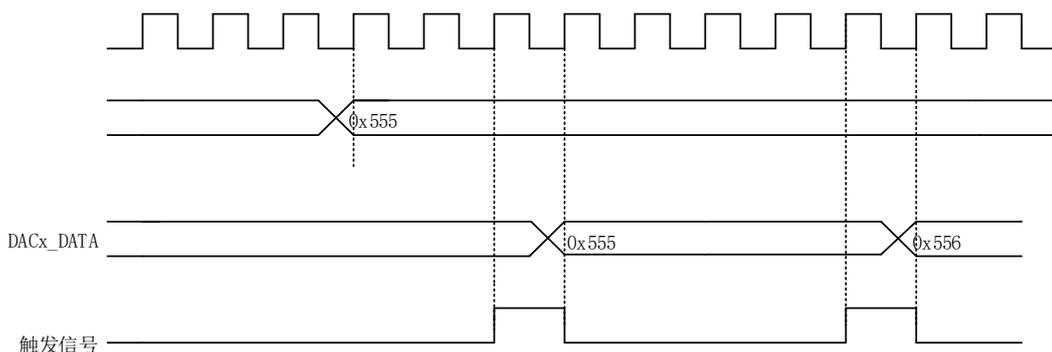


图 18-5 生成三角波波形的 DAC 转换

注：

- 为了产生三角波，必须使能 DAC 触发，即设 TRIGEN 位为 1
- MAS<3:0>位域必须在使能 DAC 之前进行配置。

18.10 DACx 校准功能

在 DACx 失调导致输出结果存在偏差时，可以开启 DAC 校准功能。

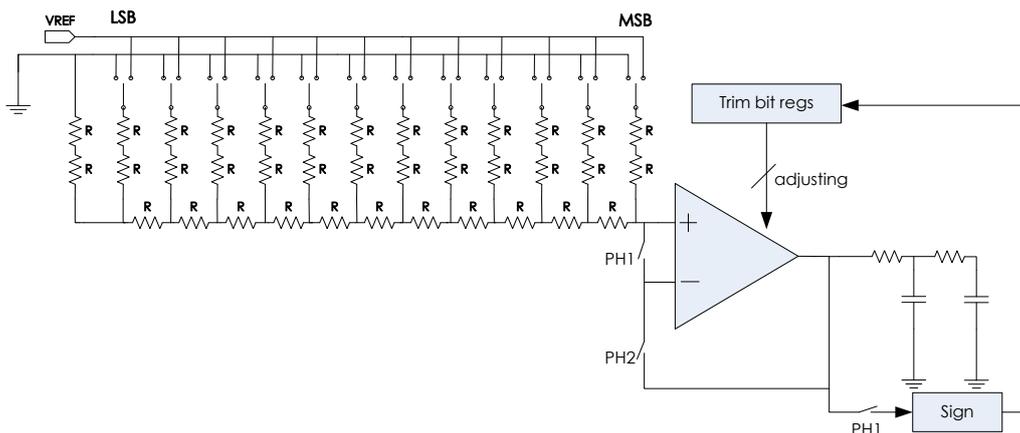


图 18-6 DAC 校准示意图

DAC 模块默认不开启校准功能，DAC 可以正常使用。当需要进行校准时，需要将 DACx_CAL 寄存器的 BUFTRIM 位置 1 以使能失调校准功能，并将 TRIM_CMP_EN 也置 1 以打开校准通路开关，校准过程分为小共模校准和大共模校准。

- **大共模 (VCOM=7/8 VDDA) 校准：**
 - (1) DAC12 的参考电压选择 VDDA，输出值选择：1110 0000 0000，也就是 VDDA 的八分之七分压点，送给缓冲器输入；
 - (2) 等待几个 NOP；
 - (3) 查看校准结果比较标志 TRIM_CMP_IF 的初始值，如果此时输出“1”，则使用 NPOSITRIM<3:0>，逐次增加 NPOSITRIM<3:0>的值，直到输出翻转位“0”，停止校准；如果此时输出“0”，则使用 NNEGATRIM<3:0>，逐次增加 NNEGATRIM<3:0>的值，直到输出翻转位“1”，停止校准。大共模校准完毕。
- **小共模 (VCOM=1/8 VDDA) 校准：**
 - (1) DAC12 的参考电压选择 VDDA，输出值选择：0010 0000 0000，也就是 VDDA 的八分之一分压点，送给缓冲器输入；

- (2) 等待几个 NOP，使比较器建立；
- (3) 采集比较器的输出值，如果此时输出“1”，则使用 PPOSITRIM<3:0>，逐次增加 PPOSITRIM<3:0>的值，直到输出翻转位“0”，停止校准；如果初始输出“0”，则使用 PNEGATRIM<3:0>，逐次增加 PNEGATRIM<3:0>的值，直到输出翻转位“1”，停止校准。小共模校准完毕。

将 BUFTRIM 位和 TRIM_CMP_EN 位清 0，DAC 恢复正常工作模式。

以上确定的校准值，不断电不会消失，将会在接下来的 DAC 使用中起作用。

19 模拟比较器模块 (CMP)

19.1 概述

单片机内置 4 个模拟比较器模块，其主要特点如下：

- 正负端多输入端口可选
- 电阻分压模块提供可选内部参考电压
- 输出极性可选
- 中断边沿可选
- 数字滤波功能
- 比较器输出可作为定时器捕捉输入、PWM 关断源或用于清零定时器
- 可配置为 BEMF（反向电动势）模式和 HALL（霍尔检测）模式

19.2 模拟比较器原理

比较器 x 通过寄存器 $CMPx_CTL$ 进行功能配置。 $PMOD<2:0>$ 位用于选择比较器正输入端口； $NMOD<2:0>$ 位用于选择比较器负输入端口； EN ($CMP3$) 位和 $CxEN$ ($CMP0/1/2$) 用于使能比较器。其中， $CMP3$ 的使能为 $CMP3_CTL$ 的 EN 位， $CMP0/1/2$ 的使能为 CMP_CTL4 的 $C0EN/C1EN/C2EN$ 位。

每个比较器都配备一个数字滤波器，可以通过配置 $FLTEN$ 位使能滤波器。 $FLTCS<1:0>$ 位用于选择滤波器的采样时钟； $FLTDIV<7:0>$ 位用于选择滤波器的工作时钟； $FLTCNT<2:0>$ 位用于设置滤波器的滤波脉宽。

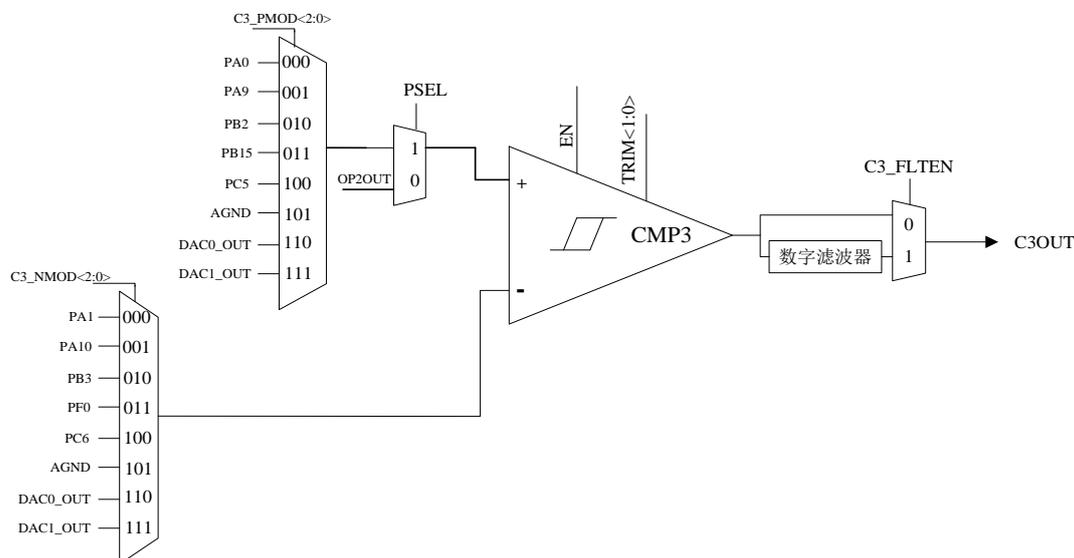


图 19-1 模拟比较器 3 框图

注：如果未开放 OP2 运放，请将该位(PSEL)配置为 1 使用。

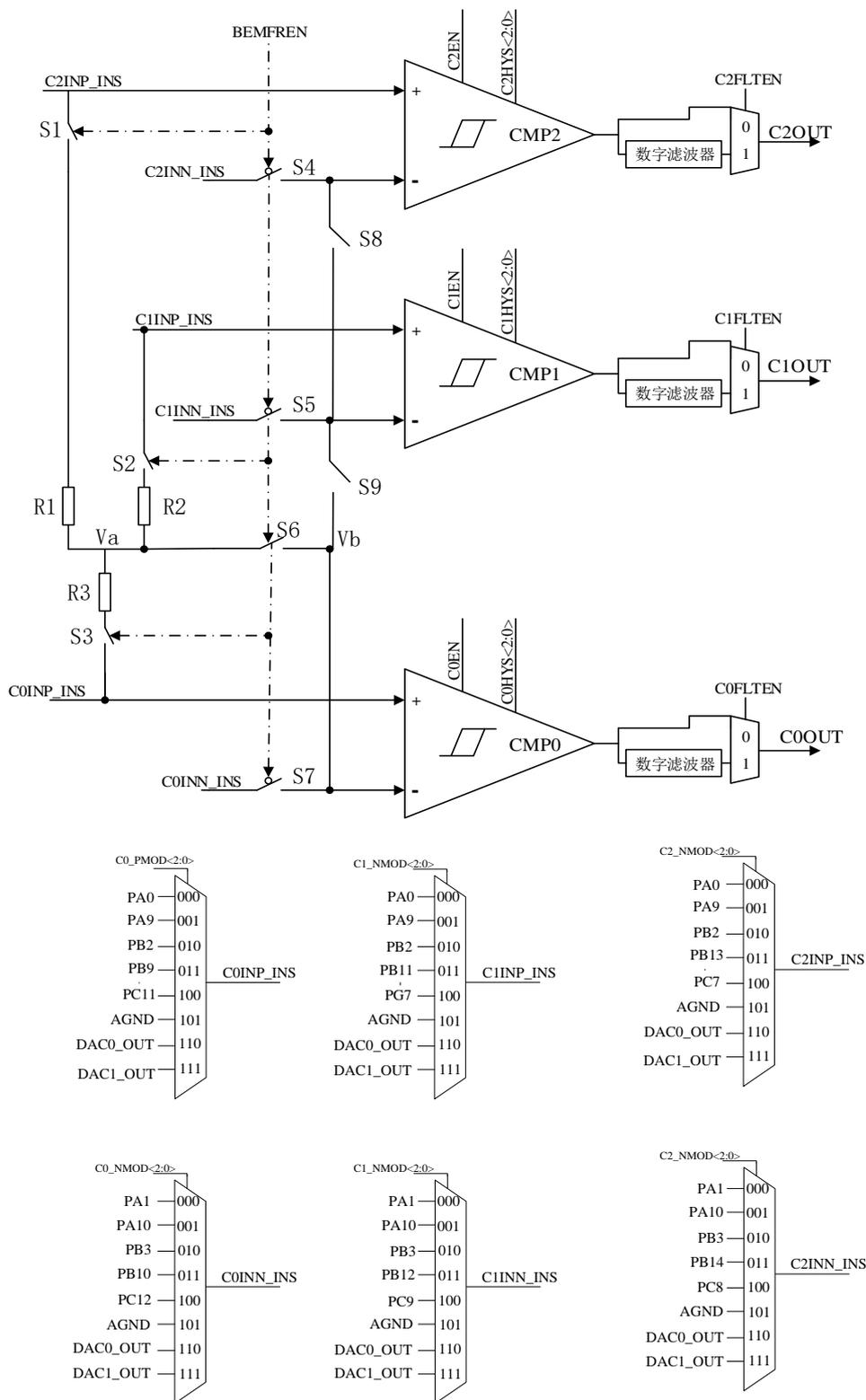


图 19-2 模拟比较器 0/1/2 框图

CMP0/1/2/3 的正端和负端输入见下表:

表 19-1 CMP0/1/2/3 正端和负端输入通道

配置位	配置值	CMP0	CMP1	CMP2	CMP3
PMOD<2:0>	000	PA0	PA0	PA0	PA0
	001	PA9	PA9	PA9	PA9
	010	PB2	PB2	PB2	PB2
	011	PB9	PB11	PB13	PB15
	100	PC11	PG7	PC7	PC5
	101	AGND	AGND	AGND	AGND
	110	DAC0_OUT	DAC0_OUT	DAC0_OUT	DAC0_OUT
	111	DAC1_OUT	DAC1_OUT	DAC1_OUT	DAC1_OUT
NMOD<2:0>	000	PA1	PA1	PA1	PA1
	001	PA10	PA10	PA10	PA10
	010	PB3	PB3	PB3	PB3
	011	PB10	PB12	PB14	PF0
	100	PC12	PC9	PC8	PC6
	101	AGND	AGND	AGND	AGND
	110	DAC0_OUT	DAC0_OUT	DAC0_OUT	DAC0_OUT
	111	DAC1_OUT	DAC1_OUT	DAC1_OUT	DAC1_OUT

19.3 模拟比较器相关的寄存器

表 19-2 模拟比较器相关寄存器

地址	寄存器	访问	功能描述	复位值
0x000	CMP_CTL0	R/W	模拟比较器控制寄存器 0	0x0000 0000
0x004	CMP_CTL1	R/W	模拟比较器控制寄存器 1	0x0000 0000
0x008	CMP_CTL2	R/W	模拟比较器控制寄存器 2	0x0000 0000
0x00C	CMP_CTL3	R/W	模拟比较器控制寄存器 3	0x0000 0000
0x010	CMP_CTL4	R/W	模拟比较器控制寄存器 4	0x0000 0000

比较器基地址为：0x4000 0800

19.3.1 CMP_CTLx 比较器控制寄存器 0/1/2

表 19-3 CMP_CTLx 比较器控制寄存器 0/1/2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W		R/W	R/W	R/W		R/W	R/W	R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W			
位名		PMOD<2:0>				NMOD<2:0>							IFFEN	IFREN	FLTCS<1:0>		FLTDIV<7:0>								FLTCNT<2:0>			FLTEN	WINEN	POL				

PMOD<2:0>：比较器 x 的正端输出选择位

见“表 19-1 CMP0/1/2/3 正端和负端输入通道”

NMOD<2:0>：比较器 x 的负端输出选择位

见“表 19-1 CMP0/1/2/3 正端和负端输入通道”

IFFEN：比较器 x 下降沿触发中断使能位

0 = 禁止比较器 x 下降沿触发中断

1 = 使能比较器 x 下降沿触发中断

IFREN：比较器 x 上升沿触发中断使能位

0 = 禁止比较器 x 上升沿触发中断

1 = 使能比较器 x 上升沿触发中断

FLTCS<1:0>：比较器 x 滤波器滤波时钟源选择位

00 = 滤波时钟源为 SCLK 系统时钟

01 = 滤波时钟源为 HFCLK 高频外设时钟

1x = 滤波时钟源为 LFCLK 低频外设时钟

FLTDIV<7:0>：比较器 x 滤波器滤波时钟分频

00H=1CLK

01H=2CLK

.....

09H=10CLK

0AH=11CLK

.....

FFH=256CLK

(CLK 为 FLTCS<1:0>选中的时钟，分频后的波形为脉宽为 0.5CLK 的方波)

FLTCNT<2:0>：比较器 x 滤波器取样数量选择位 filter sample count bits

- 00x = 滤波器不采样，滤波功能被禁止
- 001 = 滤除脉宽小于 1 个 FILT_CLK 周期的输入噪声
比较器输出延时 0~1 个 FILT_CLK 周期时间
- 010 = 滤除脉宽小于 2 个 FILT_CLK 周期的输入噪声
比较器输出延时 1~2 个 FILT_CLK 周期时间
- 011 = 滤除脉宽小于 3 个 FILT_CLK 周期的输入噪声
比较器输出延时 2~3 个 FILT_CLK 周期时间
- 100 = 滤除脉宽小于 4 个 FILT_CLK 周期的输入噪声
比较器输出延时 4~5 个 FILT_CLK 周期时间
- 101 = 滤除脉宽小于 5 个 FILT_CLK 周期的输入噪声
比较器输出延时 5~6 个 FILT_CLK 周期时间
- 110 = 滤除脉宽小于 6 个 FILT_CLK 周期的输入噪声
比较器输出延时 6~7 个 FILT_CLK 周期时间
- 111 = 滤除脉宽小于 7 个 FILT_CLK 周期的输入噪声
比较器输出延时 7~8 个 FILT_CLK 周期时间

FLTEN: 比较器 x 滤波器使能位

- 0 = 禁止滤波功能
- 1 = 使能滤波功能

WINEN: 比较器 x 范围控制使能位 window control enable bit

- 0 = 禁止范围控制功能
- 1 = 使能范围控制功能

POL: 模拟比较器 x 输出极性选择位

- 0 = 比较器正常输出: CIN+>CIN-=1, CIN+<CIN-=0
- 1 = 比较器输出极性转换: CIN+>CIN-=0, CIN+<CIN-=1

19.3.2 CMP_CTL3 比较器控制寄存器 3

表 19-4 CMP_CTL3 比较器控制寄存器 3

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	PSEL	PMOD<2:0>				NMOD<2:0>									FLTCS<1:0>		FLTDIV<7:0>								FLTCNT<2:0>			FLTEN	WINEN	POL	EN	

PSEL: 比较器 3 正端输入选择位

- 0 = 运算放大器 2 的输出 OP2OUT 作为比较器 3 的正端输入
- 1 = 由 PMOD<2:0>位选择

注: 如果未开放 OP2 运放, 请将该位配置为 1 使用。

PMOD<2:0>: 比较器 3 的正端输入选择位

见“表 19-1 CMP0/1/2/3 正端和负端输入通道”

NMOD<2:0>: 比较器 3 的负端输入选择位

见“表 19-1 CMP0/1/2/3 正端和负端输入通道”

FLTCS<1:0>: 比较器 3 滤波器滤波时钟源选择位

- 00 = 滤波时钟源为 SCLK 系统时钟

01 = 滤波时钟源为 HFCLK 高频外设时钟

13 = 滤波时钟源为 LFCLK 低频外设时钟

FLTDIV<7:0>: 比较器 3 滤波器滤波时钟分频

00H = 1CLK

01H = 2CLK

.....

09H = 10CLK

0AH = 11CLK

.....

FFH = 256CLK

(CLK 为 FLTCS<1:0>选中的时钟, 分频后的波形为脉宽为 0.5CLK 的方波)

FLTCNT<2:0>: 比较器 3 滤波器取样数量选择位 filter sample count bits

003 = 滤波器不采样, 滤波功能被禁止

001 = 滤除脉宽小于 1 个 FILT_CLK 周期的输入噪声
比较器输出延时 0~1 个 FILT_CLK 周期时间

010 = 滤除脉宽小于 2 个 FILT_CLK 周期的输入噪声
比较器输出延时 1~2 个 FILT_CLK 周期时间

011 = 滤除脉宽小于 3 个 FILT_CLK 周期的输入噪声
比较器输出延时 2~3 个 FILT_CLK 周期时间

100 = 滤除脉宽小于 4 个 FILT_CLK 周期的输入噪声
比较器输出延时 4~5 个 FILT_CLK 周期时间

101 = 滤除脉宽小于 5 个 FILT_CLK 周期的输入噪声
比较器输出延时 5~6 个 FILT_CLK 周期时间

110 = 滤除脉宽小于 6 个 FILT_CLK 周期的输入噪声
比较器输出延时 6~7 个 FILT_CLK 周期时间

111 = 滤除脉宽小于 7 个 FILT_CLK 周期的输入噪声
比较器输出延时 7~8 个 FILT_CLK 周期时间

FLTEN: 比较器 3 滤波器使能位

0 = 禁止滤波功能

1 = 使能滤波功能

WINEN: 比较器 3 范围控制使能位 window control enable bit

0 = 禁止范围控制功能

1 = 使能范围控制功能

POL: 模拟比较器 3 输出极性选择位

0 = 比较器正常输出: CIN+>CIN-=1, CIN+<CIN-=0

1 = 比较器输出极性转换: CIN+>CIN-=0, CIN+<CIN-=1

EN: 比较器 3 使能位

0 = 比较器 3 未使能

1 = 使能比较器 3

19.3.3 CMP_CTL4 比较器控制寄存器 4

表 19-5 CMP_CTL4 比较器控制寄存器 4

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R	R	R	R	R	R	R	R	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	C3OUT	C2OUT	C1OUT	C0OUT	C3IF	C2IF	C1IF	C0IF	CIFMS		CMPOUTSEL<1:0>		C3IC	C2IC	C1IC	C0IC	C3IE	C2IE	C1IE	C0IE	C3IFREN	C3IFREN			TRIM<1:0>		HALSEL	BEMFEN	FLTIMSEL	C2EN	C1EN	COEN

C3OUT: 比较器 3 输出状态位

- 0 = 比较器输出为 0
- 1 = 比较器输出为 1

C2OUT: 比较器 2 输出状态位

- 0 = 比较器输出为 0
- 1 = 比较器输出为 1

C1OUT: 比较器 1 输出状态位

- 0 = 比较器输出为 0
- 1 = 比较器输出为 1

C0OUT: 比较器 0 输出状态位

- 0 = 比较器输出为 0
- 1 = 比较器输出为 1

C3IF: 比较器 3 中断标志位

- 0 = 没有检测到比较器 3 中断
- 1 = 检测到比较器 3 中断

C2IF: 比较器 2 中断标志位

- 0 = 没有检测到比较器 2 中断
- 1 = 检测到比较器 2 中断

C1IF: 比较器 1 中断标志位

- 0 = 没有检测到比较器 1 中断
- 1 = 检测到比较器 1 中断

C0IF: 比较器 0 中断标志位

- 0 = 没有检测到比较器 0 中断
- 1 = 检测到比较器 0 中断

CIFMS: 比较器中断标志位触发模式选择位

- 0 = 通过比较器输出与锁存状态比较触发中断标志
- 1 = 通过检测比较器输出变化沿触发中断标志

CMPOUTSEL<1:0>: 比较器输出选择位, 将比较器结果直接通过 PC4 模拟通道输出

- 00 = 无输出
- 01 = 选择 CMP0 结果输出
- 10 = 选择 CMP1 结果输出
- 11 = 选择 CMP2 结果输出

注: CMP3 无法输出

C3IC: 比较器 3 中断标志清零位

- 0 = 未对比较器 3 中断标志位进行清零操作

- 1 = 对比较器 3 中断标志位进行清零操作
- C2IC: 比较器 2 中断标志清零位
- 0 = 未对比较器 2 中断标志位进行清零操作
- 1 = 对比较器 2 中断标志位进行清零操作
- C1IC: 比较器 1 中断标志清零位
- 0 = 未对比较器 1 中断标志位进行清零操作
- 1 = 对比较器 1 中断标志位进行清零操作
- C0IC: 比较器 0 中断标志清零位
- 0 = 未对比较器 0 中断标志位进行清零操作
- 1 = 对比较器 0 中断标志位进行清零操作
- C3IE: 比较器 3 中断使能位
- 0 = 禁止比较器 3 中断
- 1 = 使能比较器 3 中断
- C2IE: 比较器 2 中断使能位
- 0 = 禁止比较器 2 中断
- 1 = 使能比较器 2 中断
- C1IE: 比较器 1 中断使能位
- 0 = 禁止比较器 1 中断
- 1 = 使能比较器 1 中断
- C0IE: 比较器 0 中断使能位
- 0 = 禁止比较器 0 中断
- 1 = 使能比较器 0 中断
- C3IFFEN: 比较器 3 下降沿触发中断使能位
- 0 = 禁止比较器 3 下降沿触发中断
- 1 = 使能比较器 3 下降沿触发中断
- C3IFREN: 比较器 3 上升沿触发中断使能位
- 0 = 禁止比较器 3 上升沿触发中断
- 1 = 使能比较器 3 上升沿触发中断
- TRIM<1:0>: 比较器迟滞电压校准位
- 00 = 关闭校准
- 01 = 5mV
- 10 = 10mV
- 11 = 15mV
- HALSEL: HALL 模式选择位
- 0 = 单端选择
- 1 = 双端选择
- BEMFEN: 反向电动势模式使能位
- 0 = 比较器 0/1/2 为通用模式
- 1 = 比较器 0/1/2 为反向电动势模式
- FLTINSEL: 滤波器输入选择位
- 0 = 比较器输出作为滤波器的输入信号
- 1 = I/O 口作为滤波器的输入信号
- C2EN: 比较器 2 使能位
- 0 = 比较器 2 未使能

1 = 使能比较器 2

C1EN: 比较器 1 使能位

0 = 比较器 1 未使能

1 = 使能比较器 1

C0EN: 比较器 0 使能位

0 = 比较器 0 未使能

1 = 使能比较器 0

注: CMP1/2 工作时, 需要 C0EN 位置 1。

19.4 比较器引脚设置

比较器使用时，需要通过 GPIO 的相关寄存器将比较器的输入引脚设置成模拟引脚，通过 GPIO 的相关寄存器将比较器的输出引脚设置成重映射输出。具体可参照重映射部分。

19.5 输出极性选择

模拟比较器的输出状态可以通过 CMP_CTLx 寄存器的 POL 位结果选择，如下表所示。POL=0，比较器正常输出：C1IN+>C1IN-=1，C1IN+<C1IN-=0；POL=1，比较器输出极性转换：C1IN+>C1IN-=0，C1IN+<C1IN-=1。

表 19-6 比较器输出状态与极性关系

输入条件	极性选择	比较器输出
C1IN+ > C1IN-	INV1 = 0	C1OUT = 1
C1IN+ > C1IN-	INV1 = 1	C1OUT = 0
C1IN+ < C1IN-	INV1 = 0	C1OUT = 0
C1IN+ < C1IN-	INV1 = 1	C1OUT = 1

19.6 范围控制

范围控制功能是利用系统时钟作为筛选时钟，当时钟电平为高时，比较器结果输出；当时钟电平为低时，比较器结果保持。通过 CMP_CTLx 寄存器的 WINEN 位来使能/禁止范围控制功能。

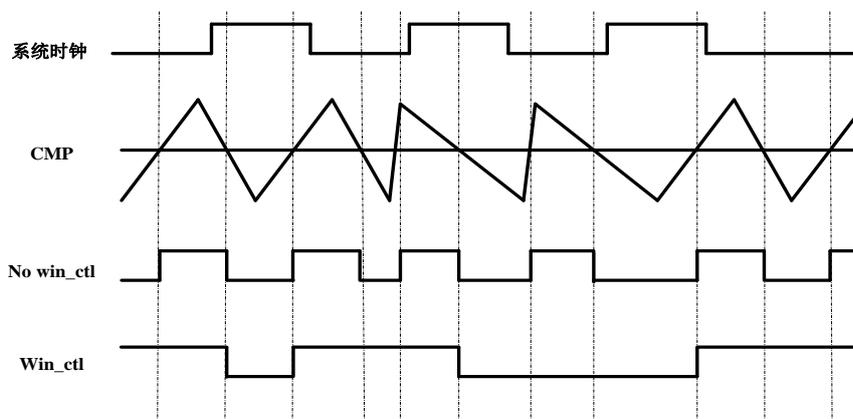


图 19-3 模拟比较器范围控制

19.7 数字滤波器

当比较器的输入电压的压差很小时，比较器的输出会发生震荡。内部和外部的寄生效应，和信号线、电源线与其它系统部分间的信号耦合造成了比较器的输出震荡。输出震荡降低了比较器的精度，影响了比较输出的结果。每个比较器都集成了一个数字滤波器，通过 CMP_CTLx 寄存器的 FLTEN 位选择将比较器的输出通过滤波器，可以减少比较器震荡带来的误差。

19.7.1 数字滤波器框图

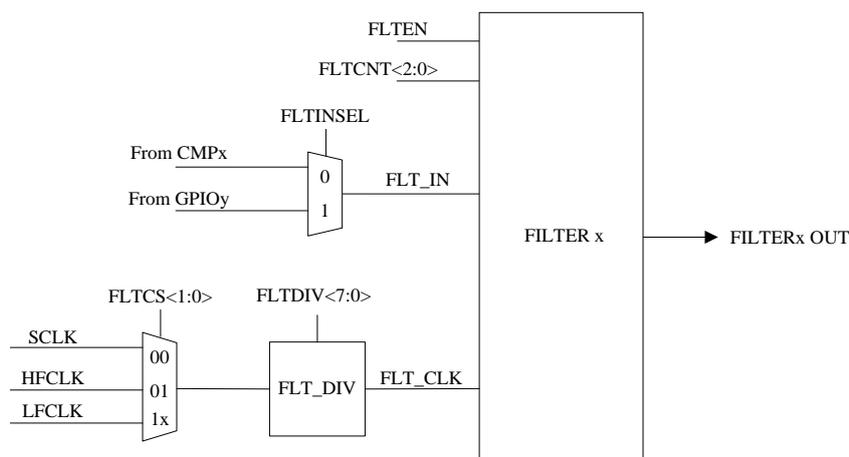


图 19-4 数字滤波器时序图

滤波器的输入可以通过 CMP_CTL4 寄存器的 FLTINSEL 位配置选择。当 FLTINSEL=0，滤波器选择比较器的输出作为滤波器的输入；当 FLTINSEL=1，滤波器选择 I/O 口输入信号作为滤波器的输入。

通过 GPIO 的重映射控制寄存器 GPIOx_RMPH 和 GPIOx_RMPL 可以将 GPIO 配置为数字滤波器的输入端口 FLT2/1/0。

注意，只有比较器 0/1/2 对应的 FILTER0/1/2 模块输入可以选择比较器 0/1/2 输出或 GPIO，比较器 3 对应的 FILTER3 只能以比较器 3 输出作为 FILTER3 的输入，无 GPIO 选项。当需要单独使用数字滤波器时，用户可通过将 CMP_CTL4 寄存器的 FLTINSEL 位置 1，选择 GPIO 作为数字滤波器的输入端。

滤波器通过 FLTCS<1:0>位和 FLTDIV<7:0>位配置得到滤波器的工作时钟(FILT_CLK)频率；FLTCNT<2:0>位配置滤波器输入噪声的滤波脉宽，FLTCNT<2:0>位=n，滤波器将滤除小于 n 个 FLT_CLK 周期宽度的输入噪声，比较器输出延时 n 到 n+1 个 FLT_CLK 周期时间。

19.7.2 数字滤波器时序

数字滤波器时序图如下所示：

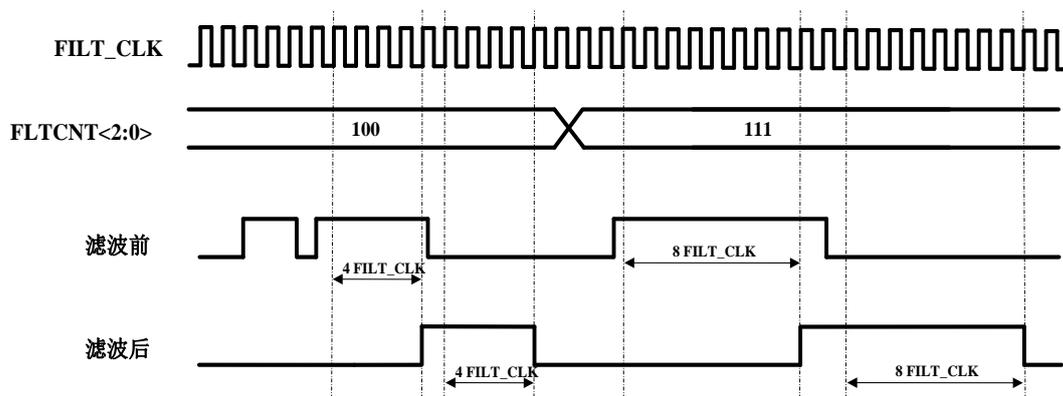


图 19-5 数字滤波器时序图

19.7.3 数字滤波器使用方法

数字滤波器使用方法：

- 通过 FLTCS<1:0>位选择滤波器的采样时钟；

- 通过 FLTDIV<7:0>位选择滤波器的工作时钟；
- 通过 FLTCNT<2:0>位设置滤波器的采样次数；
- 通过 CMP_CTLx 寄存器的 FLTEN 位来使能滤波功能。

19.8 比较器中断

CxIF 位是相应的比较器中断标志位，为只读位，不可写，必须由软件清零。

比较器中断提供不同的变化沿触发方式：

- 比较器输出上升沿触发
- 比较器输出下降沿触发
- 比较器输出上升沿和下降沿均触发

通过配置 IFREN 和 IFFEN 位分别使能比较器输出的上升沿触发和下降沿触发，当 IFREN 和 IFFEN 位同时置 1 时，比较器的上升沿和下降沿将都会触发比较器中断标志位置 1。

另外，比较器模块还提供了两种不同的变化沿检测方式，通过 CMP_CTL4 寄存器的 IFMS 位进行配置。当 IFMS=0 时，读写 CMP_CTL4 寄存器会对比较器输出结果进行锁存，通过比较器当前的输出状态与上次读写 CMP_CTL4 寄存器时比较器输出的锁存状态进行比较判断，不一致时，比较器中断标志位置 1。当 IFMS=1 时，比较器模块通过直接检测比较器输出的变化沿来判断，当对应变化沿来临时，比较器中断标志位置 1。两种检测方式均可通过 IFREN 和 IFFEN 位来配置触发方式。

如果需要系统响应比较器的中断，还需要使能 INT_EIE1 的比较器中断使能位和 INT_CTL0 全局中断使能位 AIE。

如下图所示，为 CMP 中断逻辑框图：

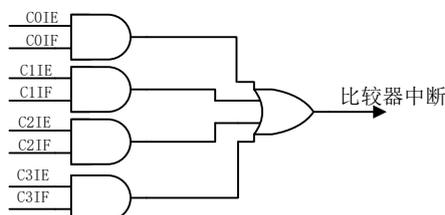


图 19-6 CMP 中断逻辑框图

19.9 比较器输出关断 PWM 和清零定时器

比较器 2/3 的输出可以作为 ECCP5/9 自动关断模式的触发关断信号。通过配置 ECCP5_PXASCTL0 寄存器的 PXASSx<2:0>位，可选择比较器 2/3 的输出作为 ECCP5/9 通道 4 自动关断的触发信号，可选择比较器 3 的输出作为 ECCP5/9 通道 1/2/3 自动关断的触发信号，触发信号为电平（高电平或低电平）有效，而非(沿)触发，即只要关断电平存在，自动关断状态将保持。

比较器 0/1/2/3 的输出可以用来清零定时器。

19.10 BEMF 模式和 HALL 模式

比较器 0/1/2 可以通过 BEMFEN 设置通用模式和反电动势模式切换。

当 BEMFEN=0 时，如上图 19-2 所示：此时开关 S4、S5、S7 闭合，其余开关断开，比

较器 0/1/2 输入通道上的内置电阻电路断开，作为通用比较器使用，通过配置可实现霍尔检测功能。

当 BEMFEN=1 时，比较器 0/1/2 将被配置为 BEMF 模式，该模式下，开关 S1、S2、S3、S6、S8、S9 闭合，其余开关断开，电阻电路接入，三个比较器的负端接在一起，并接入电阻电路（三个电阻阻值相同）。

三个电阻连线相交的点 Va 的电压为 CMP0/1/2 三个正端输入电压之和的三分之一，Vb 点的电压等于 Va 点的电压，比较器 CMP0/1/2 的负端电压一样大，也等于 Vb，为 CMP0/1/2 三个正端输入电压之和的三分之一，例如：CMP0、CMP1、CMP2 正端输入电压分别为 1V、2V、0V，则比较器 CMP0/1/2 的负端输入电压都为 $(1+2+0)/3=1v$ ，这样再与各自的正端电压去比较，实现比较器的功能。

20 通用全/半双工收发器 (USART)

20.1 概述

USART 是 Universal Synchronous /Asynchronous Receive & Transmit 的缩写，它的中文名称是通用同步/异步收发器，又称通用全双工/半双工收发器。这是一个串口通信的 I/O 外设，也可作为串行通信接口。它可被配置为与个人计算机等外设通信的全双工异步系统。也可以被配置为与外设或其它单片机通信的半双工同步系统，与之通信的单片机通常不具有产生波特率的内部时钟，它需要主控同步器件提供外部时钟信号。

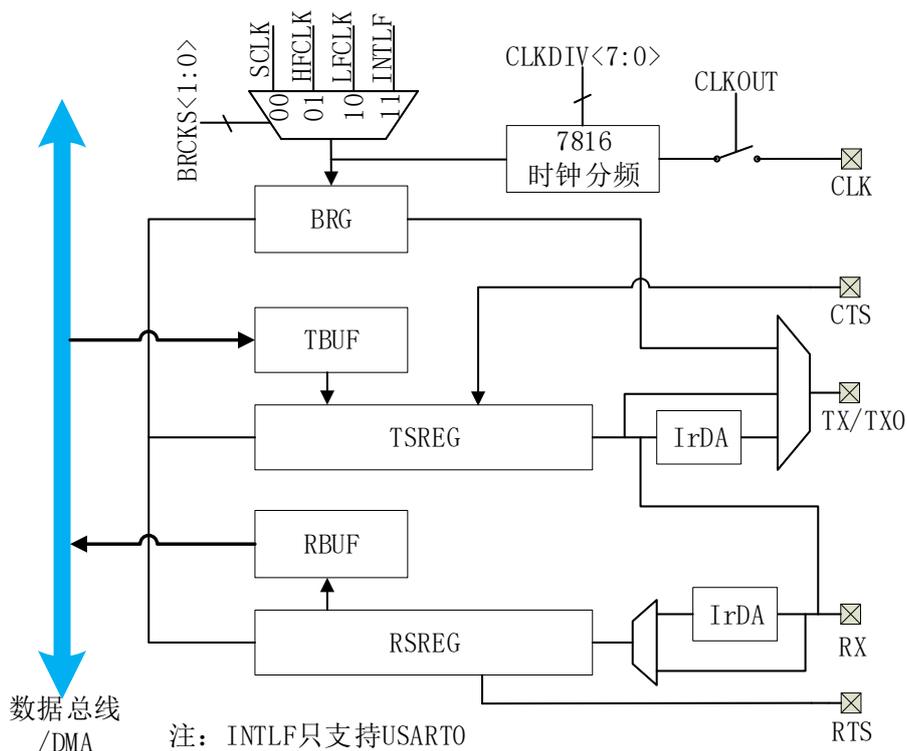


图 20-1 USART 结构框图

从上图中可以看出，通用全双工/半双工收发器 USART 模块包括波特率发生器(BRG)、数据发送和数据接收这三部分。

20.1.1 USART 引脚说明

RX: 全双工模式时为接收数据输入引脚，半双工模式时为接收或发送数据引脚。

TX: 全双工模式时为发送数据输出引脚，半双工模式时为时钟输入或输出引脚，7816 模式时为接收或发送数据引脚 TX0。

TX1: 7816 可选数据引脚。

RTS: (发送请求 Require To Send) --输出信号，用于指示本设备可接收数据，低电平有效，低电平表示本设备可以接收数据。

CTS: (接收允许 Clear To Send) --输入信号，用于判断是否可以向对方发送数据，低电平有效，低电平表示本设备可以向对方发送数据。

CLK: 7816 时钟输出引脚。

USART 特性:

- 全双工异步通信
- 半双工同步通信
- NRZ 标准格式
- 可编程数据长度（8 位或 9 位）
- 可配置停止位（1 位或 2 位）
- 单独发送器和接收器使能位
- 支持硬件奇偶校验
- 硬件自动波特率检测和校准
- 16 位整数波特率设置，8 位分数波特率设置
- 兼容 LIN 总线协议
- 支持 RS-485 模式
- 支持智能卡接口 7816 协议
- IrDA 红外调制解调
- DMA 读写
- 支持硬件流 RTS 和 CTS 控制
- 空闲帧检测

20.2 USART 相关寄存器

表 20-1 USART 相关的寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	USARTx_CTLR	R/W	USART 控制寄存器	0x0000 0000
0x004	USARTx_BRGR	R/W	USART 波特率寄存器	0x0010 0001
0x008	USARTx_STR	R/W	USART 状态寄存器	0x0000 3200
0x00C	USARTx_TBUFR	W	USART 发送数据寄存器	0x0000 0000
0x00C	USARTx_RBUFR	R	USART 接收数据寄存器	0xuuuu uuuu
0x010	USARTx_U7816R	R/W	7816 控制寄存器	0x0000 0000
0x014	USARTx_IER	R/W	USART 中断使能寄存器	0x0000 0000
0x018	USARTx_ADM	R/W	地址匹配设置寄存器	0x0000 0000

USART0 基地址: 0x4000 0980;

USART1 基地址: 0x4000 0A00;

USART2 基地址: 0x4000 0A80;

USART3 基地址: 0x4000 0B00;

USART4 基地址: 0x4000 0B80;

USART5 基地址: 0x4000 1E80;

USART6 基地址: 0x4000 1F00;

USART7 基地址: 0x4000 1F80;;

注 1: 其中 USART0 可以工作在掉电情况下, 此时它只能用来接收。

注 2: TBUFR 和 RBUFR 虽然对应相同的地址, 但是有独立的物理结构。

20.2.1 USARTx_CTLR USART 控制寄存器

表 20-2 USARTx_CTLR USART 控制寄存器

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W			R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	SLMEN	RESHD	IRDAEN	RES			RTSE	CTSE			DT9EN	PARM	TX9SEL	STPW	TIDLEEN	RXEN	TXEN	SYNC	SENDB	RIDLEIS	ABRDEN		ADREN	CSRS	WUEN	DINV	RCONV	TCONV	SCKPS	BRCKS<1:0>	USARTEN	

SLMEN: 单线模式使能位

0 = 禁止单线模式

1 = 使能单线模式

RESHD: 使用 USART 时, 请在 USARTEN 位使能之前将该位置 1

IRDAEN: 红外调制使能位

0 = 禁能 IrDA

1 = 使能 IrDA

RES: 备用位

RTSEN: RTS 使能位

0 = 禁能 RTS 功能

1 = 使能 RTS 功能

CTSEN: CTS 使能位

- 0 = 禁能 CTS 功能
- 1 = 使能 CTS 功能

DT9EN: 第 9 位数据使能位

- 0 = 未使能第 9 位数据发送和接收
- 1 = 使能第 9 位数据发送和接收

PARM: 奇偶校验选择

- 当 TX9SEL 为 0 时:
- 0 = 奇校验
 - 1 = 偶校验

当 TX9SEL 为 1 时:

- 0 = 发送/接收的为数据
- 1 = 发送/接收的为地址

TX9SEL: 发送数据第 9 位数据选择

- 0 = 选择奇偶校验
- 1 = RS-485 模式

STPW: 停止位长度选择位

- 0 = 停止位长度为 1bit
- 1 = 停止位长度为 2bits

TIDLEEN: 全双工模式时发送空闲帧使能位

- 0 = 禁止发送器发送空闲帧
- 1 = 使能发送器发送空闲帧

注: TIDLEEN=1 时, 在 TXEN&USARTEN 上升沿(高低电平最少持续一个工作时钟)时, 发送数据前自动插入一帧空闲帧(起始位、数据位、奇偶校验信和停止位都为 1)。

RXEN: 接收使能位

- 0 = 禁止接收
- 1 = 使能接收

TXEN: 发送使能位

- 0 = 禁止发送
- 1 = 使能发送

SYNC: 串行通信模式选择位

- 0 = 全双工异步模式
- 1 = 半双工同步模式

SENDB: 发送间隔字符位

- 0 = 未使能发送间隔字符
- 1 = 发送间隔字符使能(完成后由软件清零)

RIDLEIS: 接收空闲帧中断标志产生模式

- 0 = 空闲帧中断标志和 RDRIF 有关 (RDRIF=1 时才能置 1 IDLEIF)
- 1 = 空闲帧中断标志和 RDRIF 无关

ABRDEN: 自动波特率检测使能位

- 0 = 禁止自动波特率模式
- 1 = 使能自动波特率模式(完成自动波特率后清零)

ADREN: 地址检测使能位

RS-485 全双工异步模式(TX9SEL = 1):

0 = 禁止地址检测, 接收所有字节并且第 9 位可作为奇偶校验位

1 = 使能地址检测, 当地址位为 1 时数据装入 BUF

8 位异步模式(TX9SEL=0): 无关位

CSRS: 时钟源选择位

全双工异步模式: 无关位

半双工同步模式:

0 = 从模式(时钟源来自外部)

1 = 主模式(由 BRG 内部产生时钟)

WUEN: 唤醒使能位

全双工异步模式:

0 = 接收器正常工作

1 = 接收器正在等待上降沿(不会接收字符), 检测到上升沿后间隔符标志位

BRIF 将被置 1, WUEN 被自动清零。

半双工同步模式: 无关位

DINV: 红外调制高电平输出极性

0 = 发送和接收正向电平

1 = 发送和接收反向电平

RCONV: 接收次序选择

0 = 先接收 LSB

1 = 先接收 MSB

TCONV: 发送次序选择

0 = 先发送 LSB

1 = 先发送 MSB

SCKPS: 半双工同步时钟极性选择位

全双工异步模式无效

半双工同步模式:

0 = 数据在时钟上升沿同步

1 = 数据在时钟下降沿同步

BRCK<1:0>: USART 波特率发生器时钟选择位

00 = 选择 SCLK 主时钟作为 USART 波特率发生器时钟

01 = 选择 HFCLK 时钟作为 USART 波特率发生器时钟

1x = 选择 LFCLK 时钟作为 USART 波特率发生器时钟 (除 USART0 外)

10 = 选择 LFCLK 时钟作为 USART 波特率发生器时钟 (USART0)

11 = 选择 INTLF 时钟作为 USART 波特率发生器时钟 (USART0)

USARTEN: USART 使能位

0 = 禁止 USART

1 = 使能 USART

注: USART 使能位需要在其他配置好之后使能, 否则可能会导致不能运行。

20.2.2 USARTx_BRGR USART 波特率寄存器
表 20-3 USARTx_BRGR USART 波特率寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W																																
位名									BRGF2<3:0>			BRGF1<3:0>			BRGM<15:0>																	

BRGF2<3:0>: 小数波特率分母, 复位值为 1。

BRGF1<3:0>: 小数波特率分子。该位为 0 时, 小数波特率发生器不影响波特率。

BRGM<15:0>: USARTDIV 的整数部分。

这 16 位定义了 USART 分频器除法因子(USARTDIV)的整数部分。

20.2.3 USARTx_STR USART 状态寄存器
表 20-4 USARTx_STR USART 状态寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R
R/W																																
位名						IDLEIC	UADMIC	CTSIC	REIC	TEIC	WUIC	ABTOIC	BRIC	FREIC	PAREIC	OVFEIC	IDLFIF	UADMIF	TXEIF	TFEIF	RDRIF	TXFR	RCFR	CTSIF	REIF	TEIF	WUIF	ABTOIF	BRIF	FREIF	PAREIF	OVFEIF

IDLEIC: 接收空闲帧中断清零位

0 = 不清零 IDLEIF 位

1 = 清零 IDLEIF 位

UADMIC: UADM 中断清零位

0 = 不清零 UADMIF 位

1 = 清零 UADMIF 位

CTSIC: CTS 中断清零位

0 = 不清零 CTSIF 位

1 = 清零 CTSIF 位

REIC: 接收错误中断清零位

0 = 不清零 REIF 位

1 = 清零 REIF 位

TEIC: 发送错误中断清零位

0 = 不清零 TEIF 位

1 = 清零 TEIF 位

WUIC: 自动唤醒中断清零位

0 = 不清零 WUIF 位

1 = 清零 WUIF 位

ABTOIC: 自动波特率超时位清零位

0 = 不清零 ABTOIF 位

1 = 清零 ABTOIF 位

BRIC: 间隔符位清零位

0 = 不清零 BRIF 位

- 1 = 清零 BRIF 位
- FREIC: 帧错误位清零位
- 0 = 不清零 FREIF 位
- 1 = 清零 FREIF 位
- PAREIC: 奇偶校验错误位清零位
- 0 = 不清零 PAREIF 位
- 1 = 清零 PAREIF 位
- OVFEIC: 溢出错误清零位
- 0 = 不清零 OVFEIF 位
- 1 = 清零 OVFEIF 位
- IDLEIF: 接收空闲帧中断标志
- 0 = 接收器没有接收到空闲帧
- 1 = 接收器接收到空闲帧
- UADMIF: 串口地址匹配标志位
- 0 = 串口地址未匹配
- 1 = 串口地址匹配
- TXEIF: 发射器为空标志位(TBUFR 和 TSREG 均为空)
- 0 = 发射器不为空
- 1 = 发射器为空, 往发送 BUF 中写入数据可清空此位
- TFEIF: 发送 BUF 为空标志位
- 0 = 发送 BUF 不为空
- 1 = 发送 BUF 为空, 往发送 BUF 中写入数据可清空此位
- RDRIF: 数据就绪中断标志位 (通过读取接收 BUF 清空该位)
- 0 = 接收 BUF 中无数据可读
- 1 = 接收 BUF 中有数据可读
- TXFR: 发送 BUF 清零位
- 0 = 发送 BUF 正常工作
- 1 = 清空发送 BUF
- RCFR: 接收 BUF 清零位
- 0 = 接收 BUF 正常工作
- 1 = 清空接收 BUF
- CTSIF: CTS 中断位
- 0 = 未接收到有效 CTS 信号
- 1 = 接收到有效 CTS 信号
- REIF: 7816 接收错误中断标志位
- 0 = 未产生 7816 接收错误中断
- 1 = 产生了 7816 接收错误中断
- TEIF: 7816 发送错误中断标志位
- 0 = 未产生 7816 发送错误中断
- 1 = 产生了 7816 发送错误中断
- WUIF: 自动唤醒中断标志位
- 0 = 未发生自动唤醒事件
- 1 = 发生了自动唤醒事件 (RXEN、WUEN 为 1, RX 线上产生下降沿)
- ABTOIF: 自动波特率超时标志位

CLKDIV<7:0>: 7816 工作时钟和引脚输出时钟控制

当 CLKDIV 的值不为 0 时, 7816 工作时钟 f_{7816} 和系统时钟 f_{SCLK} 之间的关系为:

$$f_{7816} = f_{SCLK} / (2 * CLKDIV)$$

当 CLKDIV 的值为 0 时, f_{7816} 和 f_{SCLK} 关系为:

$$f_{7816} = f_{SCLK}$$

注: 7816 协议规定 7816 工作时钟频率范围为 1~5M。

TREPMS: 重发模式选择

0 = 只要接收到错误帧就启动重发

1 = 重发次数到设置次数后停止重发

RXRE<1:0>: 奇偶校验出错时重接收的最大次数

00 = 最大重接收次数为 0

01 = 最大重接收次数为 1

10 = 最大重接收次数为 2

11 = 最大重接收次数为 3

TXRE<1:0>: 奇偶校验出错时重发送的最大次数

00 = 最大发送收次数为 0

01 = 最大发送收次数为 1

10 = 最大发送收次数为 2

11 = 最大发送收次数为 3

RREPEN: 重收使能控制

0 = 奇偶校验错误后直接进入中断

1 = 奇偶校验错误后重收数据, 达到最大次数后进入中断

TREPEN: 重发使能控制

0 = 收到校验错误信号后直接进入中断

1 = 收到校验错误信号后重发, 达到最大次数后进入中断

BGTEN: BGT(block guard time)(块保护时间, 为两个相对方向发送的连续字符的起始沿之间的最短时间, 因此一个已接收块的最后一个字符与一个被传输块的第一个字符之间的延迟至少应为 BGT)控制位, 控制接收到发送之间是否插入 BGT

0 = 不插入 BGT

1 = 插入 BGT, 宽度为 22etu

PSEL: 数据通道选择

0 = 选择通道 0(TX0)

1 = 选择通道 1(TX1)

ERSW<1:0>: error signal 宽度选择

0x = 2etu

10 = 1.5etu

11 = 1etu

CLKOUT: 7816 时钟输出使能

0 = 禁止时钟输出

1 = 使能时钟输出

7816EN: 7816 模式使能控制

0 = 禁能 7816 模式

1 = 使能 7816 模式

20. 2. 7 USARTx_IER USART 中断使能寄存器
表 20-8 USARTx_IER USART 中断使能寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值																																	
R/W														R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名														UADMIE	UTXDE	URCDE				IDLEIE	TXEIE	TFEIE	RDRIE	CTSIE	REIE	TEIE	WUIE	ABTOIE	BRIE	FREIE	PAREIE	OVFEIE	

UADMIE: UADMIF 中断使能位

0 = 未使能 UADMIF 中断

1 = 使能 UADMIF 中断

UTXDE: DMA 写串口发送数据寄存器使能位

0 = 未使能 DMA 写串口发送数据寄存器

1 = 使能 DMA 写串口发送数据寄存器

URCDE: DMA 读串口接收数据寄存器使能位

0 = 未使能 DMA 读串口接收数据寄存器

1 = 使能 DMA 读串口接收数据寄存器

IDLEIE: IDLEIF 中断使能位

0 = 未使能 IDLEIF 中断

1 = 使能 IDLEIF 中断

TXEIE: TXEIF 中断使能位

0 = 未使能 TXEIF 中断

1 = 使能 TXEIF 中断

TFEIE: TFEIF 中断使能位

0 = 未使能 TFEIF 中断

1 = 使能 TFEIF 中断

RDRIE: RDRIF 中断使能位

0 = 未使能 RDRIF 中断

1 = 使能 RDRIF 中断

CTSIE: CTSIF 中断使能位

0 = 未使能 CTSIF 中断

1 = 使能 CTSIF 中断

REIE: 7816 接收错误中断使能位

0 = 未使能 7816 接收错误中断

1 = 使能 7816 接收错误中断

TEIE: 7816 发送错误中断使能位

0 = 未使能 7816 发送错误中断

1 = 使能 7816 发送错误中断

WUIE: 自动唤醒中断使能位

0 = 未使能自动唤醒中断位

1 = 使能自动唤醒中断位

ABTOIE: 自动波特率超时中断使能位

0 = 未使能自动波特率超时中断位

1 = 使能自动波特率超时中断位

BRIE: 间隔符位中断使能位

0 = 未使能间隔符位中断位

1 = 使能间隔符位中断位

FREIE: 帧错误中断使能位

0 = 未使能帧错误中断位

1 = 使能帧错误中断位

PAREIE: 奇偶校验错误中断使能位

0 = 未使能奇偶校验错误中断位

1 = 使能奇偶校验错误中断位

OVFEIE: 溢出错误中断使能位

0 = 未使能溢出错误中断位

1 = 使能溢出错误中断位

20. 2. 8 USARTx_ADM USART 地址匹配设置寄存器

表 20-9 USARTx_ADM USART 地址匹配设置寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
R/W																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																								ADM<7:0>																													

ADM<7:0>: USARTx 地址匹配功能设置位

20.3 功能描述

20.3.1 波特率发生器

波特率发生器 (BRG) 是 16 位定时器, 专用于支持 USART 操作。BRCK<1:0>位决定波特率发生器选用哪一种工作时钟, 其中包括 SCLK、HFCLK 和 LFCLK 三种时钟信号。BRGR 寄存器决定运行波特率定时器的周期。

20.3.1.1 波特率产生

接收器和发送器使用同一个波特率发生器 BRGA。

$$\text{Baud Rate} = \frac{F_{ck}}{16 \times \text{BRGM} \times \left(1 + \frac{\text{BRGF1}}{\text{BRGF2}}\right)} \quad \text{公式 20-1}$$

这里的 F_{ck} 是给外设的时钟; BRGM 为波特率整数部分, BRGF1 和 BRGF2 组成波特率分数部分。这 24 位的值设置在 BRGR 寄存器。

如何从 BRGR 寄存器值得到波特率 BR:

例:

如果 BRGM = 1BH, BRGF1 = 1H, BRGF2=4H (BRGR=0x41001B),

于是, 整数= 1BH

小数= 1H/4H = 0.25

所以 BR = $F_{ck}/540$

在异步模式中, 主机和从机必须配置成相同的波特率值。这要求主机和从机必须设置相同的系统时钟和工作时钟, 同时波特率寄存器 BRGR 也要设置成相同的数值。

20.3.1.2 波特率误差分析

异步模式中, 虽然主机和从机之间设置了相同的系统时钟和工作时钟, 同时波特率寄存器 BRGR 也要设置成相同的数值, 但是由于芯片间差异, 以及受温度, 电压等因素的影响, 主机和从机之间不可能保证波特率完全一致, 总是存在一定的误差。

异步模式要求在波特率误差允许的范围之内, 能够正常工作, 并保证接收数据的正确性。从设计原理分析, 可以给出一个最大的误差范围区间, 实际误差范围要根据实际的测试结构来定。

异步模式 8 位数据发送 (数据位最小), 停止位 1 位 (停止位最短) 的时候, 容许的误差范围最大。

异步模式数据采样方式: 中间值采样

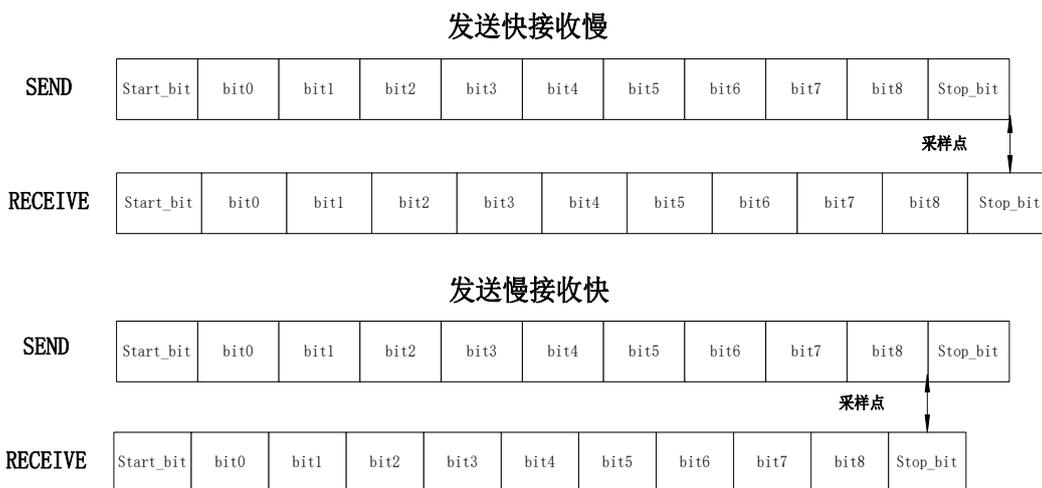


图 20-2 发送接收临界状态

Case1: 主机发送端波特率大于接收端波特率

为保证数据正确接收，箭头处对应发送数据第一位停止位的终止位置，接收端第一位停止位的中间值处

$$SEND = (8+1+1)*16$$

$$RECEIVE = (8+1)*16+8$$

注：8 表示数据发送/接收位数，1 表示停止位/起始位，每一位数据内部包含 16 个时间单位（线路内部 4 位格雷码状态机决定），每个时间单位这里称为一个字段，+8 是由于中间值采样方式决定

Case2: 主机发送端波特率小于接收端波特率

为保证数据正确接收，箭头处对应发送数据第一位停止位的起始位置，接收端对应第一位停止位的中间值处

$$SEND = (8+1)*16$$

$$RECEIVE = (8+1)*16+8$$

在相同的时间内，比较主机和从机的字段数目，可以得到发送端波特率和接收端波特率的比值 Rate

定义 $Rate = SEND / RECEIVE$

这样可以得到 Rate 在一个范围之内，实际上 Rate 要比这里公式计算得到的数值范围小。上述是在特定模式下的波特率误差分析，下面给出通用的波特率误差分析公式

Case1: 主机发送端波特率大于接收端波特率：

$$SEND = (D+1+1)*16$$

$$RECEIVE = (D+1)*16+8$$

注：D 表示数据发送/接收位数

Case2: 主机发送端波特率小于接收端波特率：

$$SEND = (D+1)*16$$

$$RECEIVE = (D+1)*16+8$$

$$Rate = SEND / RECEIVE$$

$$\text{Rate0} = ((D+1+1)*16)/((D+1)*16+8)$$

$$\text{Rate1} = ((D+1)*16)/((D+1)*16+8)$$

$$\text{Rate} \in (\text{Rate1}, \text{Rate0})$$

注：由于温度，电压，芯片间差异等影响，实际上测试可行的 Rate 要比(Rate1,Rate0)小的多，具体数值要根据测试结果定。

20.3.1.3 自动波特率检测

USART 模块支持自动波特率检测和校准。

在自动波特率检测 (Auto-Baud Rate Detect, ABRD) 模式下，由 RX 的输入信号为 BRG 定时。用于自动波特率检测发送的第一位数据必须为‘1’，第二位数据必须是 0，第三位数据必须为 1，如 0X55（低位先发）。包含起始位 0 在内，不管是低位先发还是高位先发，起始四位的序列必须满足 0_1_0_1 的要求，至于后续数据序列，不再要求。

注意：在使用自动波特率检测时，小数波特率不可用。（主机和从机都不能使用小数波特率，否则会引起计数偏差）。

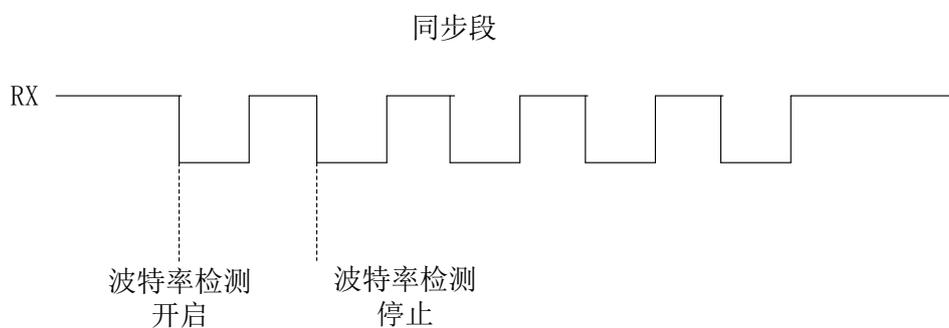


图 20-3 自动波特率检测示意图

20.3.2 间隔字符

20.3.2.1 间隔符时序

USART 模块能够发送符合 LIN 总线标准的特殊间隔字符序列。间隔符长度需由软件控制。

要发送间隔字符，只需要将 SENDB 位置 1，之后就会启动间隔字符的发送，此时 TBUFR 并不工作。在发送了间隔字符之后，需要软件将 SENDB 位清零，此时 USART 模块会自动发送一位停止位。用户可以在发送间隔字符之前将下一个要发送的字节预先装入发送 BUF 中，这样发送完间隔符后，USART 将自动开始发送新的数据。

从机接收到间隔符后，BRIF 位将会置 1。写 STR 寄存器中的 BRIC 位可清零 BRIF 位。

注意：如果一个数据的起始位、数据位、奇偶校验位和停止位均为 0，USART 模块就会判断该帧数据为间隔符并置高 BRIF 位。

20.3.2.2 间隔和同步发送序列

以下序列会发送一个报文帧头，包含一个间隔字符和其后的自动波特率同步字节。此序列适用于典型的 LIN 总线器件。

- (1) 将 USART 配置为发送模式；
- (2) 将 55H 写入 TBUFR 中，以便把同步字符装入发送 TBUFR 缓冲中；
- (3) 将 SENDB 位置 1，启动发送；
- (4) 间隔字符发送后，软件将 SENDB 位复位，同时使能 TXEN 位，开始发送同步字

符。

注：间隔字符长度由软件控制，要保证间隔字符长度大于等于 13 个 bit 长度。

20.3.3 USART 全双工模式

在全双工异步通信中，数据是一帧一帧传送的，每一帧的数据格式如下图所示。

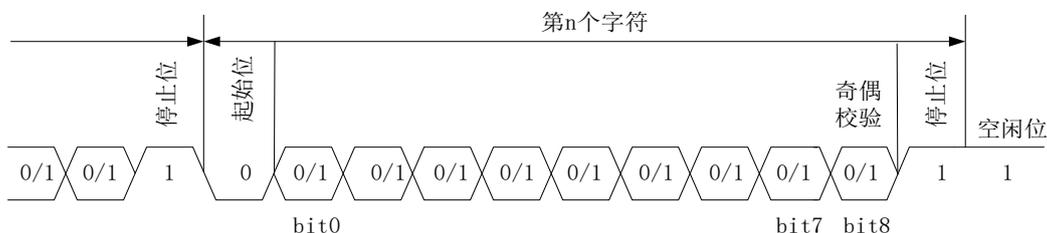


图 20-4 全双工异步通信数据结构图

在帧格式中，一个字符由 4 部分组成：起始位，数据位，奇偶校验位和停止位。

起始位：通常情况下是逻辑 0，占用一位，用来通知接收设备一个等待接收字符的开始。

数据位：8 位。

奇偶校验位：bit8，占用一位，但在字符中可以规定不用奇偶校验位，则这一位可以省去。

停止位：一定为逻辑 1，用来表征字符的结束。停止位可以是 1 位或 2 位。接收端收到停止位后，知道上一字符已经传送完毕，同时，也为接收下一字符作好准备——只要再接收到 0，就是新字符的起始位。若停止位以后不再紧接着传送下一个字符，则使线路电平保持为高电平（逻辑 1），处于空闲状态。这也是全双工异步通信的一大特点。

USART 的发送器和接收器在功能上是相互独立的，但采用相同的数据格式和波特率。硬件支持奇偶校验。

下图所示为两个串口全双工异步的连接框图。两个 USART 模块分别使用各自的 BRG 模块产生相同的波特率进行通信。

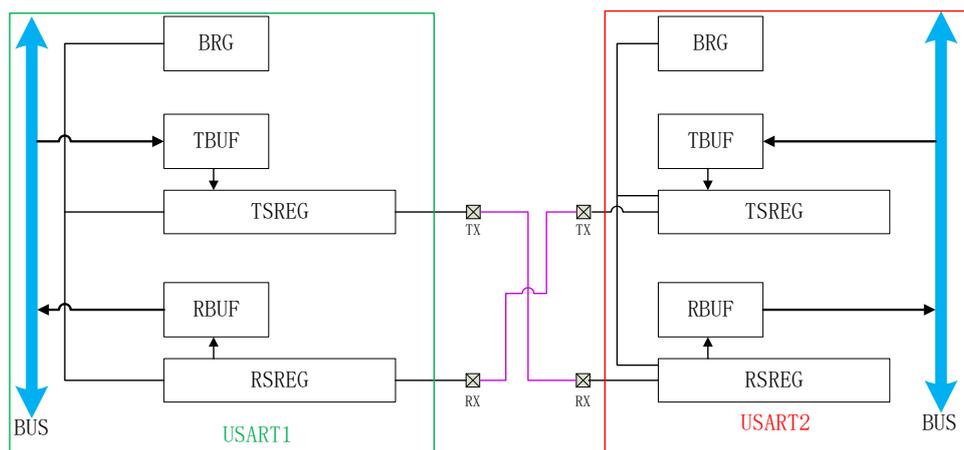


图 20-5 全双工异步连接框图

20.3.3.1 发送数据

当 TXEN 为 1 时，向 TBUF 寄存器写入一个字符，如果这是第一个字符，或者前一个字符已经完全从发送移位寄存器中移出，TBUF 中的数据会立即发送给发送移位寄存器。如果发送移位寄存器中仍保存全部或部分前一字符，新的字符数据将保存在 TBUF 中，直

到发送完前一字符的停止位为止。当数据从 TBUFR 传输至发送移位寄存器后，立即开始进行起始位、数据位和停止位序列的发送。

USTAR 寄存器的 TFE 位指示 TBUFR 的状态，当 TBUFR 为空时，TFE 置 1。当 TBUFR 和 TSREG 都为空时，TXE 置 1。TFE 和 TXE 为只读位。当有字符写入 TBUFR 时，TFE 和 TXE 被清零。用户可以查询该位或使用 TFEIF 和 TXEIF 来确定发送移位寄存器的状态。

注：TIDLEEN=1 时，在 TXEN&USARTEN 的上升沿（高低电平最少持续一个工作时钟）时，发送数据前自动插入一帧空闲帧（起始位、数据位、奇偶校验信和停止位都为 1）。

20.3.3.2 全双工发送的设置

- (1) 设置 BRGR 寄存器，对波特率等进行配置；
- (2) 设置 CTRL 寄存器，对 USART 模式等进行设置，使能发送位 TXEN；
- (3) 将需要发送的数据写入 TBUFR；
- (4) 使能 USARTEN

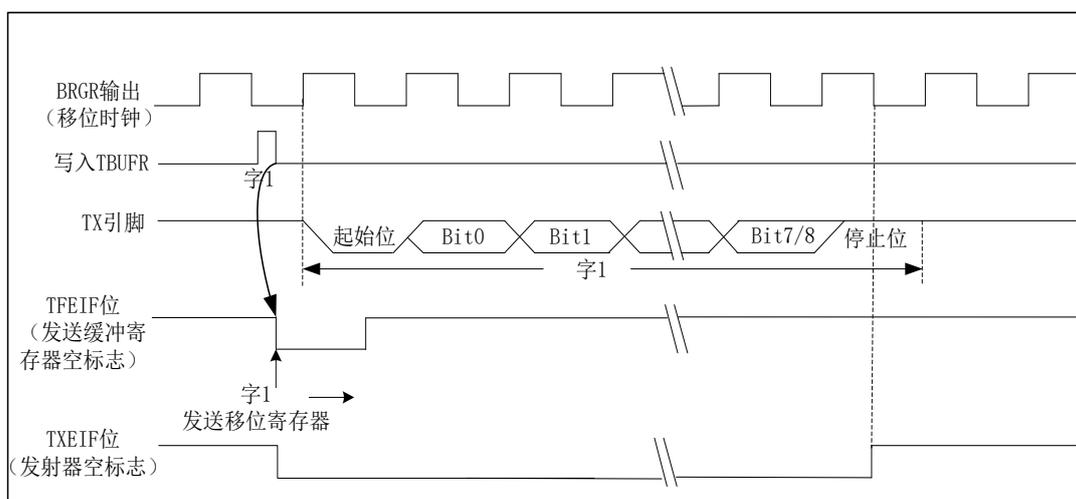


图 20-6 全双工异步发送（两字符间有空闲位）

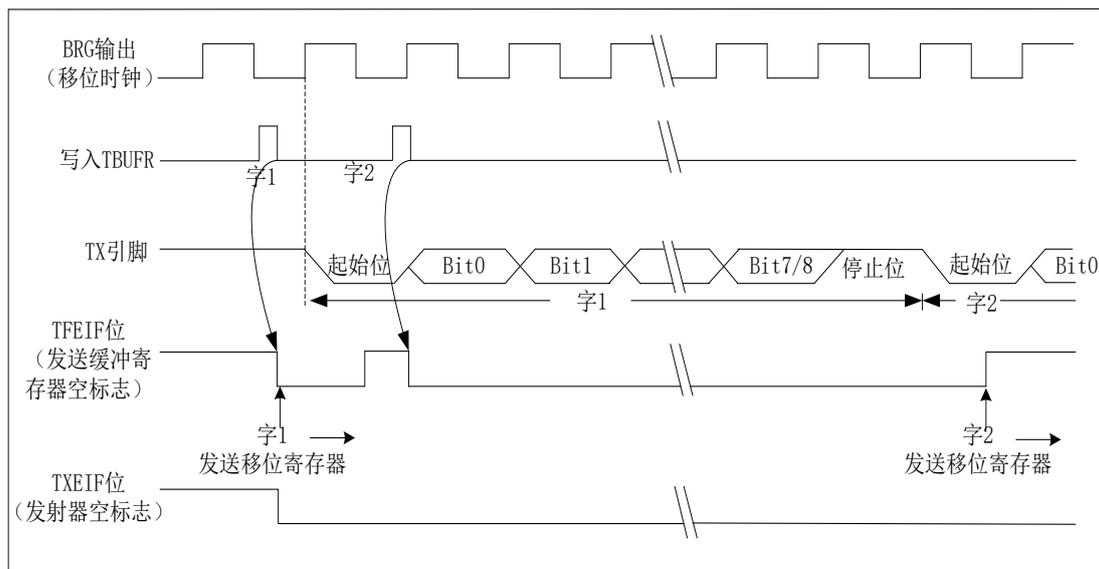


图 20-7 全双工异步发送（一字符接一字符）

20.3.3.3 接收数据

全双工异步接收模式通常用于 RS-232 系统。串行接收移位寄存器(ReceiveShift Register, RSREG)以波特率工作。当字符的全部 8 位或 9 位数据位被移入后,立即将它们传输到 RBUFR。如果 RBUFR 溢出,那么 OVFEIF 将置 1(参考接受错误章节),并且最新收到的数据将被抛弃。OVFEIF 位需要通过软件置高 OVFEIC 位清零。通过读 RBUFR 寄存器将 RBUFR 里的字符移出 RBUF 缓冲器。

只要在 RBUFR 中有未读数据,STR 寄存器中的 RDR 状态位就会置 1。RDR 状态位为只读,通过读取 RBUFR 清零该位。

注:当接收完数据后,总线空闲时间超过一帧数据时间时,接收器会将空闲中断标志 IDLEIF 置 1,如果 IDLEIE 为 1,将产生空闲中断。

20.3.3.4 接收错误

RBUFR 中的每个字符都有一个相应的帧错误状态位。帧错误指示未在预期的时间内接收到停止位。由 STR 寄存器的 FREIF 位代表 RBUFR 未读字符的状态。

FREIF 位为只读位。帧错误(FREIF = 1)并不会阻止接收更多的字符。需要由软件置高 FREIC 位清零 FREIF 位。

RBUFR 可以保存 1 个字符。但如果在读 RBUFR 之前,接收到完整的第 2 个字符,则会产生溢出错误。此时,STR 寄存器的 OVFEIF 位会置 1。可以读取 RBUFR 内的字符,但是在错误清除之前,不能再接收其它字符。可以通过置高 STR 寄存器的 OVFEIC 位来清除错误。

20.3.3.5 地址检测

当多个接收器共享同一传输线时(如在 RS-485 系统中),可使用特殊地址检测模式。将 CTRL 寄存器的 ADREN 位置 1,使能地址检测模式。地址检测要求接收 9 位字符。使能地址检测后,只有第 9 位数据位被置 1 的字符可以被传输到 RBUFR。被传输到 RBUFR 中的字符将与地址匹配设置寄存器 ADM 中的值进行比较,当匹配时,使 UADMIF 中断标志位置 1,同时若在休眠模式下,可唤醒 CPU。所有其它字符将被忽略。通过 UADMIC 可以清零 UADMIF 标志位。

20.3.3.6 全双工接收的设置

- (1) 设置 BRGR 寄存器,对波特率等进行配置;
- (2) 设置 CTRL 寄存器,对 USART 模式等进行设置,使能 RXEN 位;
- (3) 接收数据。

20.3.3.7 RS-485 发送/接收

USART 支持 RS-485 模式 9 位发送。当 CTRL 寄存器的 DT9EN 和 TX9SEL 位置 1 时,可通过 PARM 位来控制第 8 位数据(最低位为第 0 位)的值。详情可见 CTRL 寄存器说明。

USART 将接收到的每个字符的第 8 位(最低位为第 0 位)用于判断该位为数据还是地址。当读出 RBUFR 中的值第 8 位为 1 时,表示该帧数据为地址。

注意:在使用 9 位发送模式时,不能使用奇偶校验位。

20.3.4 USART 半双工模式

半双工同步串行通信通常用在具有一个主控器件和一个或多个从动器件的系统中。主控器件包含产生波特率时钟所必需的电路,并为系统中的所有器件提供时钟。从动器件可以使

用主控器件的时钟，因此无需内部时钟发生电路。

在半双工同步模式下，有 2 条信号线：双向数据线和时钟线。从动器件使用主控器件提供的时钟，将数据串行移入或移出相应的接收和发送移位寄存器。半双工是指：主控器件和从动器件都可以接收和发送数据，但是不能同时进行接收或发送。USART 既可以作为主控器件，也可以作为从动器件。

半双工同步模式发送无需使用起始位和停止位。

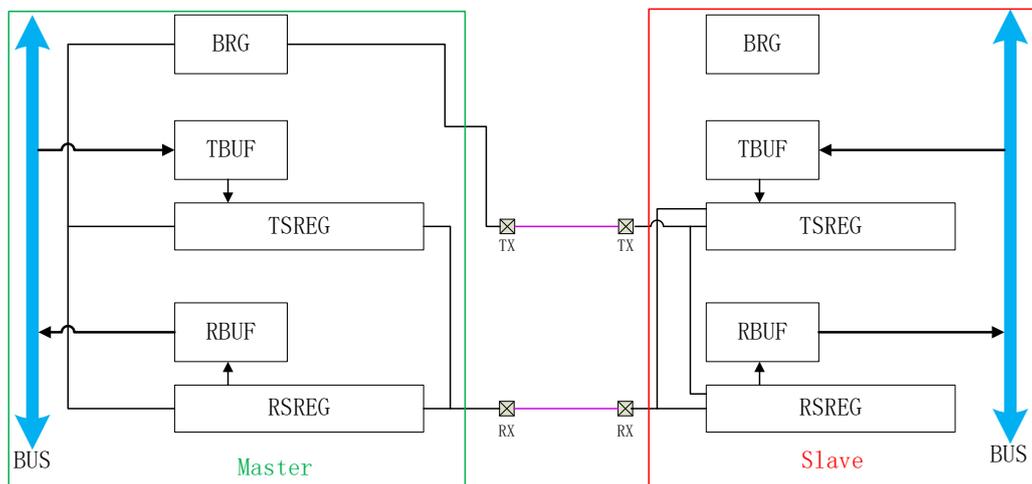


图 20-8 USART 半双工异步连接框图

上图展示了两个 USART 模块在半双工异步模式下的连接框图。其中，左边的模块作为主机，右边的模块做为从机。主机的时钟从 TX 口输出，从机从 TX 管脚接收时钟。主机的 RX 引脚连接从机的 RX 引脚用来交换数据；当主机发送的时候，从机接收；从机发送的时候，主机接收。

20.3.4.1 USART 半双工主控模式

下列位用来将 USART 配置为半双工同步主控操作：

- SYNC = 1
- RESHD = 1
- CSRS = 1
- RXEN = 0, TXEN = 1（用于发送）；RXEN = 1, TXEN = 0（用于接收）
- USARTEN = 1

将 CTLR 寄存器的 SYNC 位置 1，并将 RESHD 位置 1，可将 USART 配置用于半双工同步操作。将 CTLR 寄存器的 CSRS 位置 1，将器件配置为主控器件。将 CTLR 寄存器的 RXEN 位清零，TXEN 置 1 以确保器件处于发送模式，将 CTLR 寄存器的 RXEN 位置 1，TXEN 清零配置为接收模式。将 CTLR 寄存器的 USARTEN 位置 1，使能 USART。

半双工同步数据传输使用独立的时钟线半双工同步传输数据。配置为主控器件的器件在 TX/CK 引脚发送时钟信号。每个数据位的时间为一个时钟周期，有多少数据位就只能产生多少个时钟周期。

由 CTLR 寄存器的 SCKPS 位选择时钟极性。当 SCKPS 位置 1 时，数据在每个时钟的下降沿发生改变。当清零 SCKPS 位时，数据在每个时钟的上升沿发生改变。

20.3.4.2 半双工主控发送

由器件的 RX/DT 引脚输出数据。向 TBUF 寄存器写入一个字符并使能 USARTEN 和

TXEN 开始发送。如果发送移位寄存器中仍保存全部或部分前一字符，新的字符数据保存在 TBUFR 中，直到发送完前一字符的停止位为止。如果这是第一个字符，或者前一个字符已经完全从发送移位中移出，则 TBUFR 中的数据会被立即传输到发送移位寄存器。当字符从 TBUFR 传输到发送移位后会立即开始发送数据。

注意，发送移位寄存器并未映射到数据存储器中，因此用户不能直接访问它。

半双工主控发送设置：

- (1) 初始化 BRGR 寄存器以获得所需的波特率（详见章节“波特率发生器”）；
- (2) 将 SYNC、RESHD、TXEN 和 CSRS 位置 1，使能半双工同步主控串行端口；
- (3) 将 RXEN 位清零，禁止接收模式；
- (4) 如果需要发送 9 位字符，将 DT9EN 置 1；
- (5) 通过 SCKPS 选择极性；
- (6) 若需要中断，使能 IER 寄存器中的对应的位（*注）；
- (7) 将数据装入 TBUFR 寄存器；
- (8) 将 USARTEN 位置 1 使能 USART，启动发送。

注意：

- (1) 中断需要放在模式配置结束后，否则可能导致一开始由于 TXEIF 和 TFEIF 进入中断而无法退出。
- (2) 所有用到的 IO 及 USART 模块都需将时钟使能并退出复位。

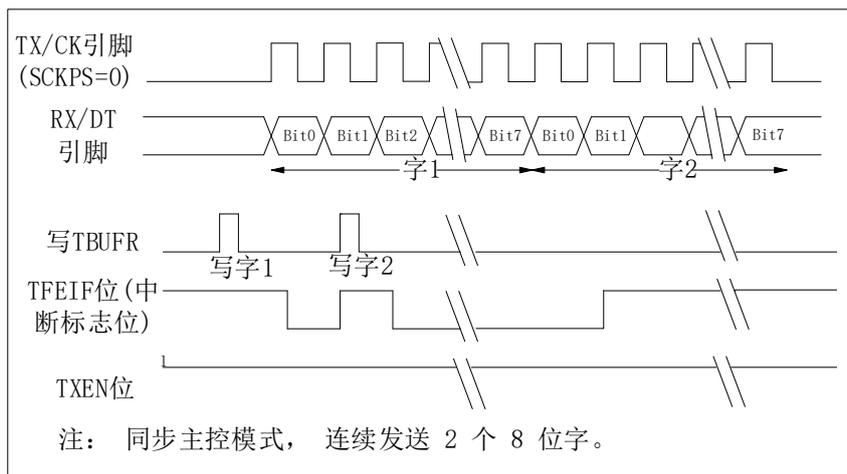
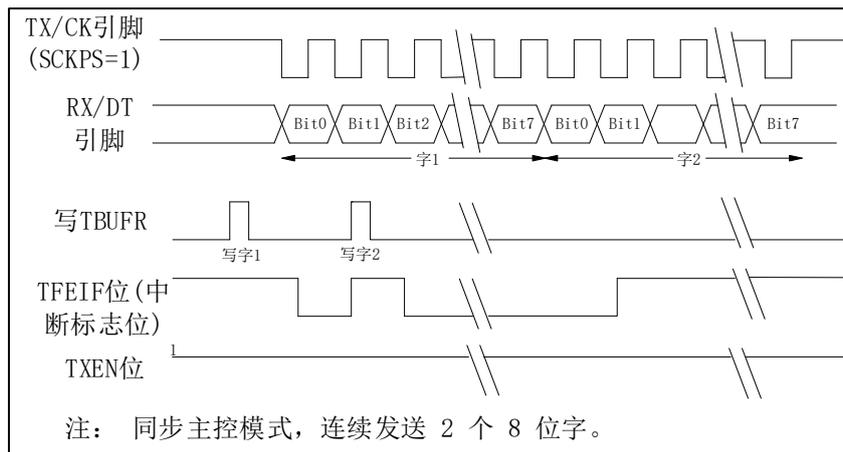


图 20-9 半双工同步发送 (SCKPS = 0)


图 20-10 半双工同步发送(SCKPS=1)

20.3.4.3 半双工主控接收

在 RX/DT 引脚接收数据。在半双工同步模式下，将接收使能位 (CTRL 寄存器的 RXEN 位) 置 1 使能接收。当 RXEN 置 1 时，将产生连续时钟，直到清零 RXEN 为止。如果 RXEN 在一个字符的传输过程中清零，则 CK 时钟立即停止，并丢弃该不完整的字符。

将 RXEN 位置 1，启动接收。在 TX/CK 时钟引脚信号的下降沿采样 RX/DT 引脚上的数据，并将采样到的数据移入接收移位寄存器 (RSR)。当 RSR 接收到一个完整字符时，将 RDR 位置 1，字符移入接收 BUF。只要接收 BUF 缓冲器中有未读字符，则 RDR 位保持置 1 状态。

半双工同步数据传输使用于数据线通读的独立时钟线。配置为从器件的器件接收 TX/CK 线上的时钟信号。串行数据位在时钟信号的沿改变，以确保其在每个时钟的后沿有效。每个时钟周期只能传输一位数据，因此有多少数据位要传输就必须接收多少个时钟。

接收 BUF 缓冲器可以保存 1 个字符。在读 RBUF 以访问 BUF 缓冲器之前，若完整地接收到第 2 个字符，则产生溢出错误。此时，STR 寄存器的 OVFEIF 位会置 1。BUF 缓冲器中先前的数据不会被改写。可以读取 BUF 缓冲器内的字符，但是在错误被清除前，不能再接收其它字符。通过置高 STR 的 OVFEIC 位，将 OVFEIF 位清零。

半双工主控接收设置:

- (1) 用正确的波特率初始化 BRGR 寄存器，以获得所需的波特率。
- (2) 将 SYNC、RESHD 和 CSRS 位置 1 使能半双工同步主控串行端口；
- (3) 将 RXEN 位置 1，TXEN 清零使能接收；
- (4) 通过 SCKPS 选择极性；
- (5) 若需要中断，使能 IER 寄存器中的对应的位；
- (6) 将 USARTEN 位置 1 使能 USART，启动接收；
- (7) 当字符接收完毕后，将 RDRIF 位置 1。如果 RDRIE 置 1，还会产生一个中断；
- (8) 读 RBUF 寄存器获取接收到的 8 位数据，硬件将自动清空 RDRIF；
- (9) 如果产生溢出错误，置高 OVFEIC 来清除溢出错误(清除溢出错误后，需要清零 OVFEIC)。

注：所有用到的 IO 及 USART 模块都需将时钟使能并退出复位。

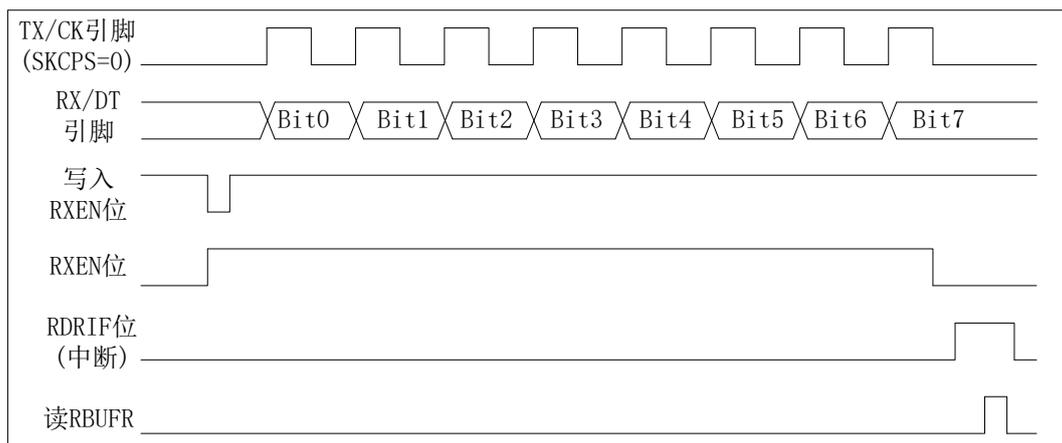


图 20-11 半双工同步接收（主控模式，RXEN=1，SCKPS=0）

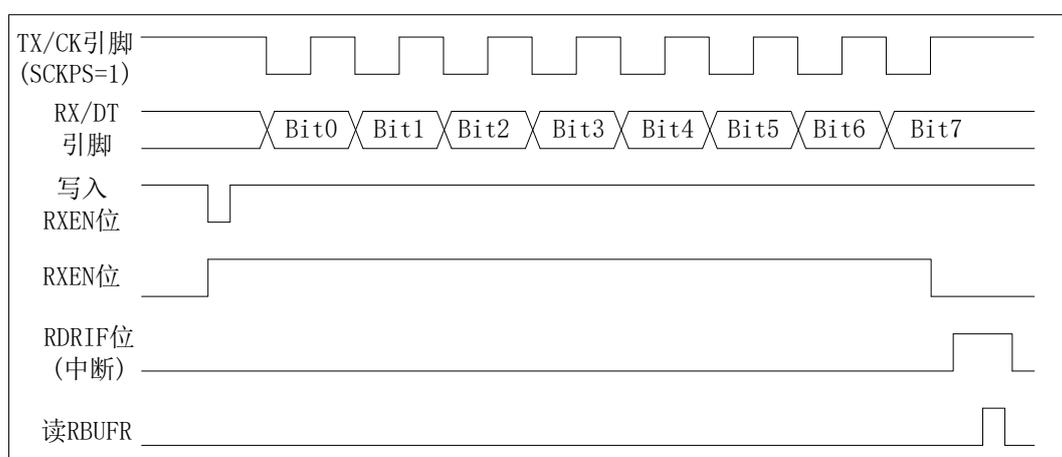


图 20-12 半双工同步接收（主控模式，RXEN=1，SCKPS=1）

20.3.4.4 USART 半双工从动模式

下列位用来将 USART 配置为半双工同步从动操作：

- SYNC=1
- RESHD=1
- CSRS=0
- RXEN = 0, TXEN = 1（用于发送）；RXEN = 1, TXEN = 0（用于接收）
- USARTEN = 1

将 CTLR 寄存器的 SYNC 位置 1，可将器件配置用于半双工同步操作。将 CTLR 寄存器的 CSRS 位置 0，将器件配置为从动器件。将 CTLR 寄存器的 RXEN 位清零，以确保器件处于发送模式，否则器件将被配置为接收模式。将 CTLR 寄存器的 USARTEN 位置 1，使能 USART。

20.3.4.5 USART 半双工从动发送

半双工从动发送设置：

- (1) 将 SYNC 位和 RESHD 位置 1 并将 CSRS 位清零；
- (2) 将 RXEN 位清零，TXEN 位置 1 使能发送；
- (3) 若需要中断，使能 IER 寄存器中的对应的位；
- (4) 根据需要配置其它 CTLR 寄存器；

- (5) 将低 8 位数据写入 TBUFR 寄存器；
 - (6) 等待 TX/CK 线上的时钟，开始数据传输。
- 注：所有用到的 IO 及 USART 模块都需将时钟使能并退出复位。

20.3.4.6 USART 半双工从动接收

半双工从动接收设置：

- (1) 将 SYNC 位和 RESHD 位置 1 并将 CSRS 位清零；
- (2) 若需要中断，使能 IER 寄存器中的对应的位；
- (3) 将 RXEN 位置 1，使能接收；
- (4) 当接收完成后，将 RDRIF 位置 1。如果 RDRIE 已置 1，还会产生一个中断；
- (5) 读 RBUFR 寄存器，从接收 BUF 缓冲器获取接收到的 8 个数据位；
- (6) 如果产生溢出错误，置高 OVFEIC 来清除溢出错误(清除溢出错误后，需要清零 OVFEIC)。

注：所有用到的 IO 及 USART 模块都需将时钟使能并退出复位。

20.3.4.7 USARTA 半双工 RS-485 模式

请参考“RS-485 发送/接收”

20.3.5 硬件流控制

RTS: (发送请求 require to send) --输出信号，用于指示本设备可接收数据，低电平有效，低电平表示本设备可以接收数据。

CTS: (接收允许 clear to send) --输入信号，用于判断是否可以向对方发送数据，低电平有效，低电平表示本设备可以向对方发送数据。

如果 CTS 流控制被使能(CTSEN=1),发送器在发送下一帧前检查 nCTS 输入。如果 nCTS 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 TXEN=0),否则下一帧数据不被发出去。若 nCTS 在传输期间被变成无效,当前的传输完成后停止发送。

如果 RTS 流控制被使能(RTSEN=1),只要 USART 接收器准备好接收新的数据, nRTS 就变成有效(接低电平)。当接收寄存器内有数据到达时, nRTS 被释放,由此表明希望在当前帧结束时停止数据传输。

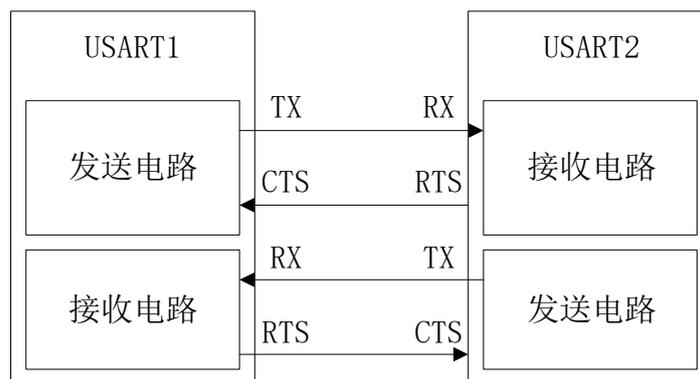


图 20-13 硬件流控制

20.3.6 IrDA 红外调制

通过设置 CTRL 寄存器的 IRDAEN 位选择 IrDA 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。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 (见图 20-14 IrDA 数据调制)。
- 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 输出处于低电平状态。
- 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 1.41us。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 BRGR 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，CTRL 寄存器上的 STPW 位必须配置成 1 个停止位。

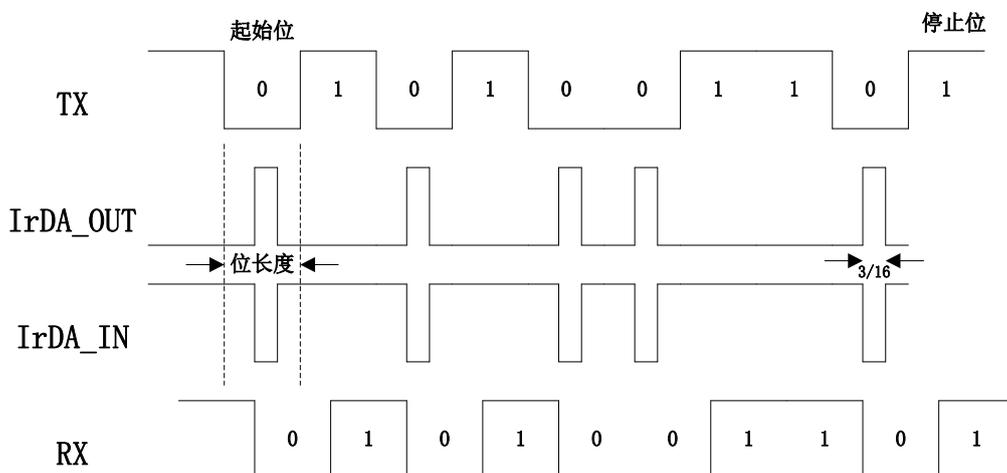


图 20-14 IrDA 数据调制

20.3.7 7816 模式

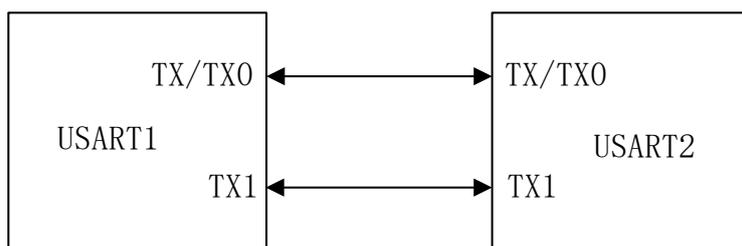


图 20-15 7816 连接方式

7816 模式时可通过 PSEL (USARTx_U7816R<4>) 选择 TX 或 TX1 进行数据通信, PSEL 为 0 时选择 TX, PSEL 为 1 时选择 TX1。

7816 模式基于 ISO/IEC 7816-3 标准, 利用此功能可与其他 7816 设备进行通信。按照 7816 协议标准, 7816 基本接口时序如图所示:

无校验错误



校验错误



图 20-16 7816 接口时序

- 一个起始位后跟 8 个数据位以及一个校验位, 以 2etu 的 guard time 结束。
- 第 10.5 个 etu 接收电路校验接收数据, 如果正确, 插入 2etu 的 guard time, 确保数据长度为 12etu, 完成数据发送; 若校验错误, 则在第 10.5etu 拉低 IO, 产生 error signal。error signal 长度可从 1etu、2etu、1.5etu 中选择。
- 第 11 个 etu 时发送电路未采样到 error signal, 则说明发送数据正确, 数据发送完成。
- 若第 11 个 etu 发送电路采样到 error signal, 则说明发送数据错误, 等待 2 个 etu 后重发数据。

20.3.7.1 7816 模式发送

发送设置

通过配置如下控制位使能 USART 发送器, 以用于 7816 模式发送操作:

- (1) 7816EN = 1, 使能 7816 模式;
- (2) TXEN = 1, 使能发送端;
- (3) SYNC = 0, 选择异步模式;
- (4) CLKOUT = 1, 使能 7816 时钟输出;
- (5) DT9EN = 1, 选择 9 位数据发送;
- (6) TX9SEL = 1, 选择自动生成奇偶校验码;
- (7) STPW = 1, 选择 2 位停止位;
- (8) PARM、TCONV、BGTEN、PSEL、TREPEN、TXRE、EGT 等控制寄存器视情况进行选择;

(9) USARTEN = 1, 使能串口。

注：所有用到的 IO 及 USART 模块都需将时钟使能并退出复位。

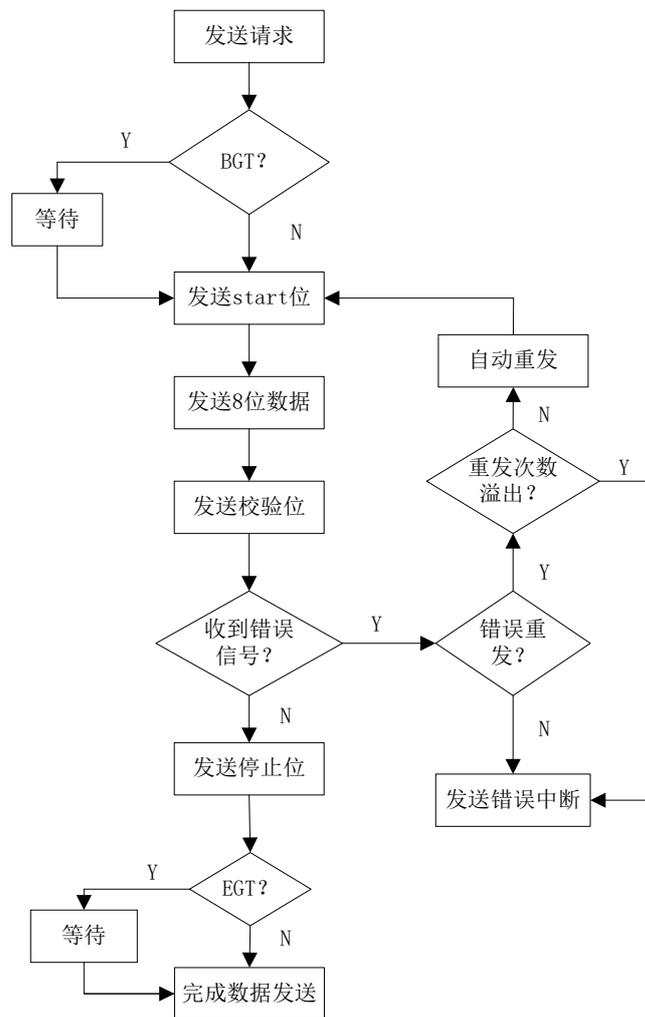


图 20-17 数据发送流程

20.3.7.2 7816 模式接收

通过配置如下控制位使能 USART 接收器，以用于 7816 接收操作：

- (1) RXEN = 1, 使能接收端；
- (2) 7816EN = 1, 使能 7816 模式；
- (3) SYNC = 0, 选择异步模式；
- (4) DT9EN = 1, 选择 9 位数据接收；
- (5) CLKOUT = 1, 使能 7816 时钟输出；
- (6) PARM、RCONV、BGTEN、PSEL、RREPEN、RXRE、EGT 等控制寄存器视情况进行选择；
- (7) USARTEN = 1 使能串口。

注：所有用到的 IO 及 USART 模块都需将时钟使能并退出复位。

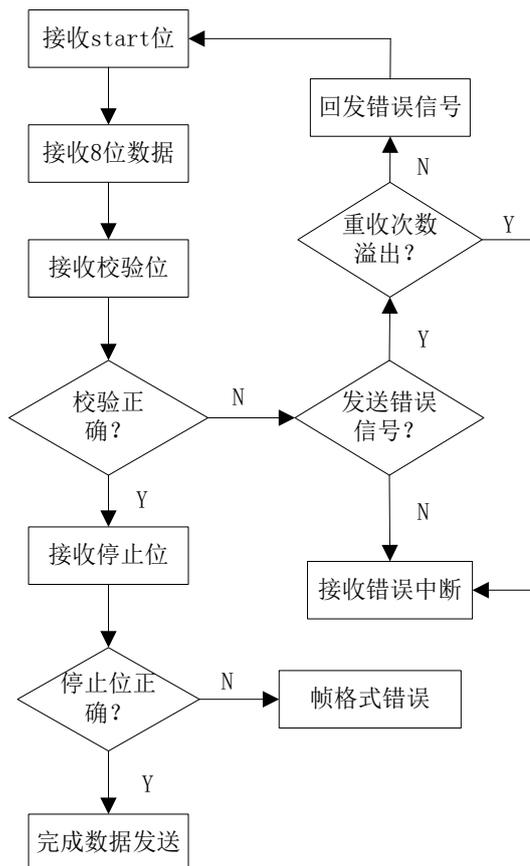


图 20-18 数据接收流程

20.3.8 USART 单线通信模式

USART 单线通信模式（半双工异步模式）通过 USARTx_CTLR 寄存器的 SLMEN 位置 1 使能。在使能 USART 单线通信模式之前，须确保 USARTx_CTLR 寄存器的 SYNC 位保持清零状态。

USART 单线通信模式下，USART 相关引脚特性如下：

- RX 引脚不再使用，即可作为通用 I/O 口使用；
- TX0 引脚作为数据通信脚；
- 当 USART 配置为单线通信接收时，TX0 引脚（重映射为 USART 功能）一直处于接收状态；
- 当 USART 配置为单线通信发送时：
 - 当有数据发送时，TX0 引脚（重映射为 USART 功能）处于数据发送状态；
 - 当无数据发送时（空闲状态），TX0 引脚被释放，表现为通用 I/O 口，因此在无数据发送时需要将 TX0 引脚配置为数字输出口，并输出高电平（根据 USART 通信协议，当 USART 处于空闲状态时，TX0 被拉高）。

用户可通过软件来管理线上冲突。

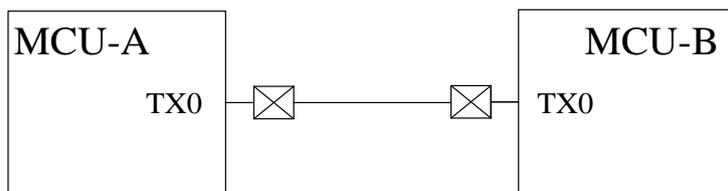


图 20-19 单线模式连接示意

20.3.8.1 单线发送模式

单线模式的发送和全双工的发送基本相同，可参考“发送数据”。

单线发送模式设置：

设置 BRGR 寄存器，对波特率等进行配置；

设置 CTLR 寄存器，对 USART 模式等进行设置，SYNC 设置为 0，SLMEN 设置为 1，使能发送位 TXEN；

将 TX0 的对应引脚重映射为 USART 功能；

将需要发送的数据写入 TBUFR；

使能 USARTEN；

在数据发送完成后，将 TX0 对应引脚设置为数字输出模式，并输出高电平（空闲状态）。

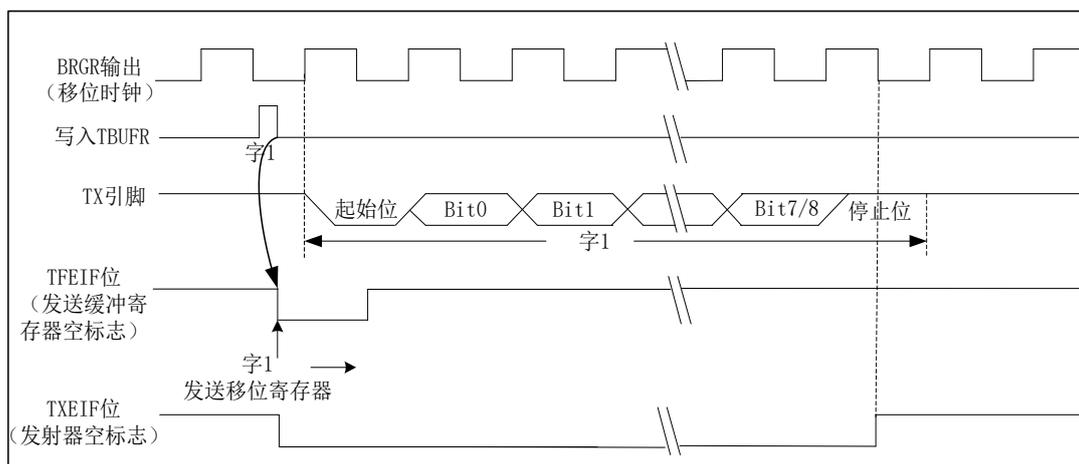


图 20-20 单线模式连接示意

20.3.8.2 单线接收模式

单线模式的接收和全双工的接收基本相同，可参考“接收数据”。

单线接收模式设置：

(1) 设置 BRGR 寄存器，对波特率等进行配置；

(2) 设置 CTLR 寄存器，对 USART 模式等进行设置，SYNC 设置为 0，SLMEN 设置为 1，使能接收位 RXEN；

(3) 将 TX0 的对应引脚重映射为 USART 功能；

(4) 使能 USARTEN。

注：所有用到的 IO 及 USART 模块都需将时钟使能并退出复位。

20.3.9 USART 的 DMA 读写

DMA 写：

(1) 配置 DMA 工作模式，DMA 的外设地址寄存器为 USARTx 的 TBUFR 地址，数据

方向 DDIR 位置 1（从存储器读取，写入外设）；

- (2) 配置 USARTx 的波特率和工作模式，UTXDE 位（USARTx_IER<17>）置 1，使能 DMA 写请求；
- (3) 使能 USARTx 的发送，在启动发送后，TFEIF 置 1，自动启动 DMA 写 TBUFR。

DMA 读：

- (1) 配置 DMA 工作模式，DMA 的外设地址寄存器为 USARTx 的 RBUFR 地址，数据方向 DDIR 位置 0（从外设读取，写入存储器）；
- (2) 配置 USARTx 的波特率和工作模式，URCDE 位（USARTx_IER<16>）置 1，使能 DMA 读请求；
- (3) 在 USARTx 接收到一个字节后，RDFIF 置 1，自动启动 DMA 读

20.3.10 USART 中断

USART 的中断逻辑如下图所示：

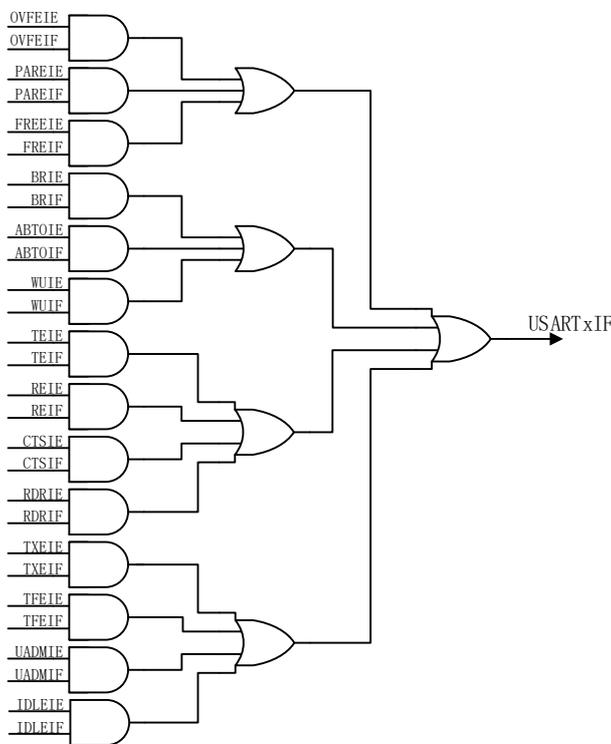


图 20-21 USART 中断框图

USART 外设模块到内核的中断标志 USARTxIF 由 USART 模块内部的所有中断逻辑组合产生，只要有一组中断标志及使能位都有效时（置 1），USARTxIF 将置 1，向内核申请中断，中断是否响应由内核的中断使能位控制。

20.3.11 低功耗 USART（LP-USART）

USART0 可作为低功耗 USART（LP-USASRT）使用。低功耗的 USART 可以在除待机和关断模式以外的模式中工作。当单片机处于停止模式下，内核已经掉电，此时低功耗 USART 仍可以保持在工作状态，并能通过中断信号唤醒 CPU。

为了让 USART0 进入低功耗模式，需要使能 PM_CTL2 寄存器的 USART0LPEN 位不让 USART0 复位，同时设置 USARTx_CTLR 寄存器的 BRCK<1:0>=11 即选用内部低频振荡器

INTLF 作为相应的时钟源，并将 PM_CTL2 寄存器的 USART0CLKLPEN 置 1 允许内部低频振荡器作为 USART0 工作时钟，同时保持时钟源处于工作状态，这样就可以在低功耗模式下运行。

USART0 处于低功耗模式工作时，只能使用 PA0~PA6 口作为引脚。

21 串行外设接口（SPI）

21.1 概述

SPI 模块可配置为支持 SPI 协议或者 I2S 协议。SPI 模块默认工作在 SPI 方式，可通过软件将其切换到 I2S 模式。在 I2S 模式下，原则上数据传输为全双工模式，主机和从机同时收发数据，但实际情况下通常只有一个方向上的数据是有意义的。

SPI 模式主要特征：

- 3 线或者 4 线数据传输
- 8/16/32 位传输帧格式
- MSB/LSB 先发送可选
- 主从模式
- 时钟频率可设
- 可编程的时钟极性和相位
- 可触发中断的发送和接收标志
- DMA 读写

I2S 主要特征：

- 单工通信
- 主从模式
- 数据长度可为 16/32 位
- 8 位线性可编程预分频器（音频采样频率 8KHz 到 96KHz）
- 可编程时钟极性
- 支持多种 I2S 协议：
 - I2S 飞利浦标准
 - LSB 对齐标准（右对齐）
 - MSB 对齐标准（左对齐）
 - PCM 标准
- DMA 读写
- 可输出的主时钟，频率可通过 I2CCKS<1:0>（SPIx_CTLR<26:25>）位配置

21.1.1 SPI 功能描述

SPI 模式允许同时同步发送和接收数据。通常使用以下三个引脚来完成通信：

- SDO：串行数据输出（Serial Data Out）
- SDI：串行数据输入（Serial Data In）
- SCK：串行时钟（Serial Clock）

此外，当工作在从动模式下时可以使用第 4 个引脚：

- \overline{SS} ：从动选择（Slave Select）

SPI 原理框图如下图所示：

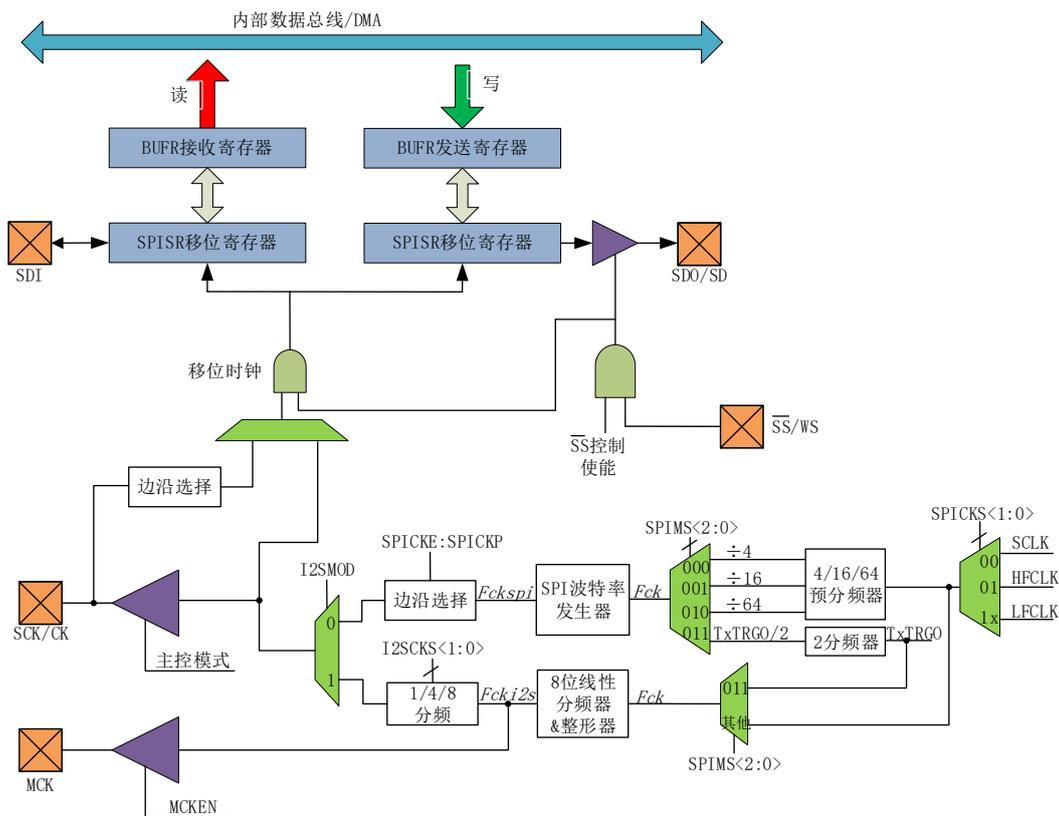


图 21-1 SPI 原理框图

注意：

当 SPI 处于从动模式，并且使能 \overline{SS} 引脚控制（SPIMS<2:0> = 100）时，如果 \overline{SS} 引脚设为 VDD，那么 SPI 模块将复位。

如果使用 SPI 从动模式，且 SPICKE = 1，则必须使能 \overline{SS} 引脚控制。

21.1.2 I2S 功能描述

I2S 与 SPI 共用 3 个引脚：

- SD：串行数据口（映射至 SDO 口），用来发送和接收数据
 - WS：字选口（映射至 \overline{SS} 口），主模式下作为数据控制输出，从模式下输入
 - CK：串行时钟口（映射至 SCK 口），主模式下作为时钟输出，从模式下输入
- I2S 也提供一个额外的时钟引脚，用于支持那些需要主时钟的外部音频设备：
- MCK：主时钟口，输出时钟频率可通过 I2SCKS<1:0>（SPIx_CTLR<26:25>）位配置

21.2 相关寄存器

表 21-1 SPIx 模块相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	SPIx_BRGR	R/W	SPI 波特率寄存器	0x0000 0000
0x004	SPIx_CTLR	R/W	SPI 控制寄存器	0x0000 0000
0x008	SPIx_BUFR	R/W	SPI 数据寄存器	0x0000 0000
0x00C	SPIx_STR	R/W	SPI 状态寄存器	0x0000 0000

SPI0 基地址: 0x4000 0C00

SPI1 基地址: 0x4000 0C80

SPI2 基地址: 0x4000 2080

SPI3 基地址: 0x4000 2100

21.2.1 SPIx_BRGR SPI 波特率寄存器

表 21-2 SPIx_BRGR SPI 波特率寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名							I2SODD	MCKEN	I2SDIV<7:0>							SPIBRG<15:0>																

I2SODD: I2S 时钟精度微调位

MCKEN: 主设备时钟输出使能

0 = 未使能 MCK 输出

1 = 使能 MCK 输出

I2SDIV<7:0>: I2S 预分频寄存器 (只对 I2S 有效)

$$F_{ck_{I2C}} = \frac{F_{ck}}{2 \times I2SDIV + I2SODD} \quad \text{公式 21-1}$$

注:

F_{ck} 可通过 SPIMS<2:0>选择不同时钟源;

I2SDIV 不能设置为小于 6 的值。

SPIBRG<15:0>: SPI 波特率寄存器 (只对 SPI 有效)

$$F_{ck_{SPI}} = \frac{F_{ck}}{2 \times (SPIBGR + 1)} \quad \text{公式 21-2}$$

21. 2. 2 SPIx_CTLR SPI 控制寄存器
表 21-3 SPIx_CTLR SPI 控制寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名					PCMCKP	I2SCKS<1:0>	PCMS	I2SSS<1:0>	I2SMC<1:0>		MCKOS	SMINS	I2SMOD		SPIBS<1:0>								SPICKE	SPICKP	SPIIOS	SPIDS	SPICKS<1:0>		SPTMS<2:0>		SPIEN	

PCMCKP: PCM 模式时钟极性选择

0 = PCM 数据在时钟上升沿发送

1 = PCM 数据在时钟下降沿发送

I2SCKS<1:0>: I2S 模式的时钟分频选择

0x = I2S 波特率 F_{ckI2S}

10 = I2S 波特率 $F_{ckI2S/4}$

11 = I2S 波特率 $F_{ckI2S/8}$

注: F_{ckI2S} 参考 SPIx_BRGR 寄存器说明

PCMS: PCM 帧同步

0 = 短帧同步

1 = 长帧同步

I2SSS<1:0>: I2S 标准选择

00 = I2S 飞利浦标准

01 = 低字节对齐标准 (右对齐)

10 = 高字节对齐标准 (左对齐)

11 = PCM 标准

I2SMC<1:0>: I2S 模式设置

00 = 从设备发送模式

01 = 从设备接收模式

10 = 主设备发送模式

11 = 主设备接收模式

MCKOS: I2S 空闲时 MCK 输出模式选择

0 = 连续输出模式, 空闲时 MCK 输出

1 = 禁止输出模式, 空闲时 MCK 禁止

I2SMOD: I2S 模式选择

0 = 选择 SPI 模式

1 = 选择 I2S 模式

SPIBS<1:0>: SPI 位模式选择

00 = SPI 使用 8 位模式 (不支持 I2S 模式)

01 = SPI/I2S 使用 16 位模式

10 = SPI/I2S 使用 32 位模式

11 = 保留

SPICKE: SPI 时钟边沿选择位

0 = 在第 1 个时钟沿 (包括上升, 下降) 发送数据

1 = 在第 2 个时钟沿 (包括上升, 下降) 发送数据

21. 2. 4 SPIx_STR SPI 状态寄存器
表 21-5 SPIx_STR SPI 状态寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值																																	
R/W											R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W					R	R	R	R	R	R	R	
位名											STXDE	SRCDE		UDRIC	TOVFIC	ROVFIC	TNEIE	RNEIE	TOVFIE	ROVFIE	UDRIE					UDRIF	TOVFIF	ROVFIF	CHSIDE	TNE	RNE	SPIBUSY	

STXDE: DMA 写 SPI 数据寄存器使能

0 = 未使能 DMA 写 SPI 数据寄存器

1 = 使能 DMA 写 SPI 数据寄存器

SRCDE: DMA 读 SPI 数据寄存器使能

0 = 未使能 DMA 读 SPI 数据寄存器

1 = 使能 DMA 读 SPI 数据寄存器

UDRIC: 从机发送下溢中断清零

0 = 不清零从机发送下溢中断

1 = 清零从机发送下溢中断

TOVFIC: 发送溢出中断清零

0 = 不清零发送溢出中断

1 = 清零发送溢出中断

ROVFIC: 接收溢出中断清零

0 = 不清零接收溢出中断

1 = 清零接收溢出中断

TNEIE: TBUF 为空中断使能

0 = 未使能 TBUF 为空中断

1 = 使能 TBUF 为空中断

RNEIE: RBUF 不为空中断使能

0 = 未使能 RBUF 不为空中断

1 = 使能 RBUF 不为空中断

TOVFIE: 发送溢出中断使能

0 = 未使能发送溢出中断

1 = 使能发送溢出中断

ROVFIE: 接收溢出中断使能

0 = 未使能接收溢出中断

1 = 使能接收溢出中断

UDRIE: 从机发送下溢中断使能

0 = 不使能从机发送下溢中断

1 = 使能从机发送下溢中断

UDRIF: 从机发送下溢中断标志

0 = 未产生从机发送下溢中断

1 = 产生从机发送下溢中断(从发送模式时第一个时钟到达时还没写发送 BUF)

TOVFIF: 发送溢出中断标志

0 = 未产生发送溢出中断

1 = 产生了发送溢出中断 (TBUF 不为空时写发送缓冲器)

ROVFIF: 接收溢出中断标志

0 = 未产生接收溢出中断

1 = 产生接收溢出中断 (RBUF 不为空时再接收到数据)

CHSIDE: 声道

0 = 正在发送的是左声道

1 = 正在发送的是右声道

TNE: 发送 BUF 未空, 启动发送时该位为 0, 写 SPIx_BUFRR 后该位为 1

0 = 发送 BUF 为空

1 = 发送 BUF 不为空

RNE: 接收 BUF 未空, 接收结束该位为 1, 读 SPIx_BUFRR 后该位为 0

0 = 接收 BUF 为空

1 = 接收 BUF 不为空

SPIBUSY: SPI 忙

0 = SPI 没有在发送数据

1 = SPI 正在发送数据

注: RNE 和 TNE 这两位引起的中断标志会随着 SPIx_BUFRR 状态的改变而改变, 并不需要像溢出中断这样需要专门的一位来清零。

21.3 SPI 操作

21.3.1 SPI 工作原理

当初始化 SPI 时，需要通过相应的控制位编程来指定几个选项。这些控制位用于设置以下选项：

- 主控模式（SCK 作为时钟输出）
- 从动模式（SCK 作为时钟输入）
- 时钟极性（SCK 的空闲状态）
- 时钟边沿（在 SCK 的上升沿/ 下降沿输出数据）
- 从动选择模式（仅用于从动模式）

SPI 模块由一个发送/ 接收移位寄存器 SPISR（SPISR 是内部寄存器，程序无法直接访问）和数据缓冲寄存器（SPIx_BUFR）组成。SPISR 对进出器件的数据进行移位。在新数据接收完毕前，SPIx_BUFR 保存上次写入 SPISR 的数据。一旦 8（16/32）位数据接收完毕，该数据就被移入 SPIx_BUFR 寄存器。然后，缓冲器非空检测位 RNE 被置 1。这种数据接收方式，允许在 CPU 读取刚接收的数据之前，就开始接收下一个字节。

为确保应用软件能有效地接收数据，应该在要发送的下一数据字节写入 SPIx_BUFR 之前，读取 SPIx_BUFR 中现有的数据。缓冲器非空检测位 RNE 指出将接收到的数据装入 SPIx_BUFR（发送完成）的时间。当 SPIx_BUFR 中的数据被读取后，RNE 位即被清零。如果 SPI 仅作为一个发送器，则不必理会接收的数据。通常，可用 SPIBUSY 和 RNE 来判断发送和接收完成的时间。此外，SPI 状态寄存器（SPIx_STR）指示各种状态条件。

注：SPI 主机不管是发送或接收，都需先写一个数据到数据缓冲寄存器（SPIx_BUFR），用于启动 SPI。

21.3.2 使能 SPI/IO 与外部链接

21.3.2.1 使能 SPI/IO

与 SPI 相关 IO 口需设置为重映射模式，根据主从模式，硬件将自动控制 IO 口方向。

21.3.2.2 典型连接

下图 SPI 主从控制器连接给出了两个单片机之间的典型连接。主控制器（处理器 1）通过拉低 \overline{SS} 线并发送 SCK 信号来启动数据传输。在两个处理器的移位寄存器之间，数据在编程设定的时钟边沿被传送，并在相反的时钟边沿被锁存。必须将两个处理器的时钟极性（SPICKP）设置为相同，这样两个处理器就可以同时收发数据。数据是否有效，取决于应用软件。这就导致以下三种数据传输情形：

- 主控制器发送数据—从控制器发送无效数据
- 主控制器发送数据—从控制器发送数据
- 主控制器发送无效数据—从控制器发送数据

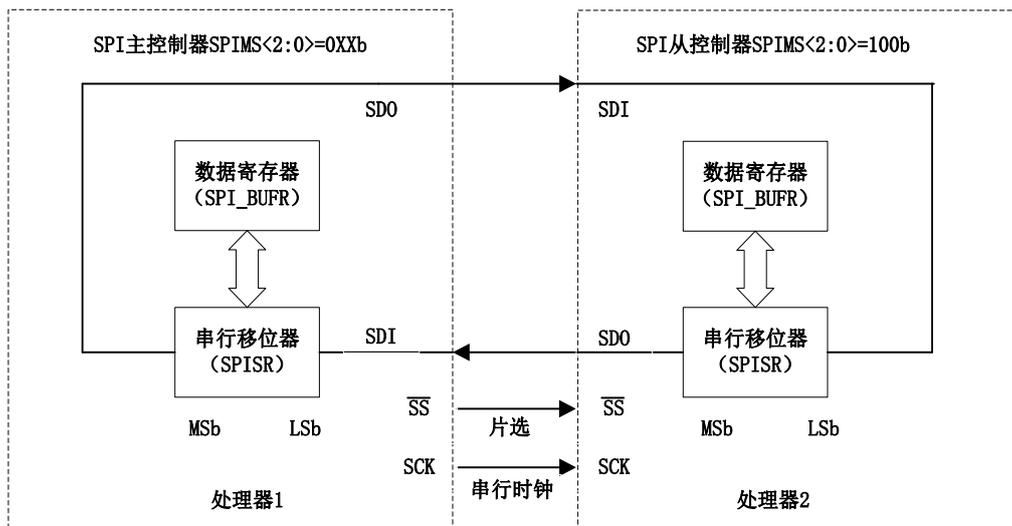


图 21-2 SPI 主从控制器连接 (SPIIOS=0)

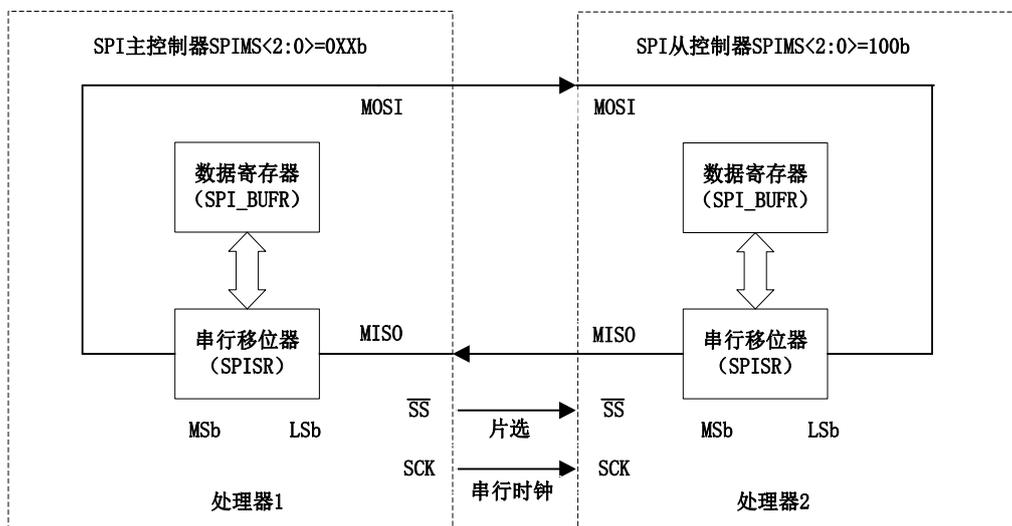


图 21-3 SPI 主从控制器连接 (SPIIOS=1)

注：MOSI 对应 SDO，MISO 对应 SDI

21.3.3 主模式

因为由主控制器控制 \overline{SS} 和 SCK 信号，所以它可以在任意时刻启动数据传输。主控制器根据软件协议确定从控制器（上图 SPI 主从控制器连接中的处理器 2）应在何时广播数据。在主控模式下，数据一旦写入 SPI_x_BUFR 寄存器就开始发送或接收。SPIISR 寄存器按设置的时钟速率，对 SDI 引脚上的信号进行连续移位输入。每收到一个完整的数据（8 位、16 位或 32 位），就将其装入 SPI_x_BUFR 寄存器（中断和状态位相应置 1）。

可通过对 SPICKP 位进行适当的编程来选择时钟极性。图 SPI 主控模式时序图将给出 SPI 通信的时序图，其中首先发送的是最高有效位。在主控模式下，SPI 时钟源可由用户编程设定为下面几种方式之一：

- TxTRGO/2
- 工作时钟/4

- 工作时钟/16
- 工作时钟/64

下图 SPI 主控模式时序图给出了主控模式的波形图。当 SPICKE 位置 1 时，SDO 数据在 SCK 出现时钟边沿前一直有效。图中给出了将接收到的数据装入 SPIx_BUF 的时间。

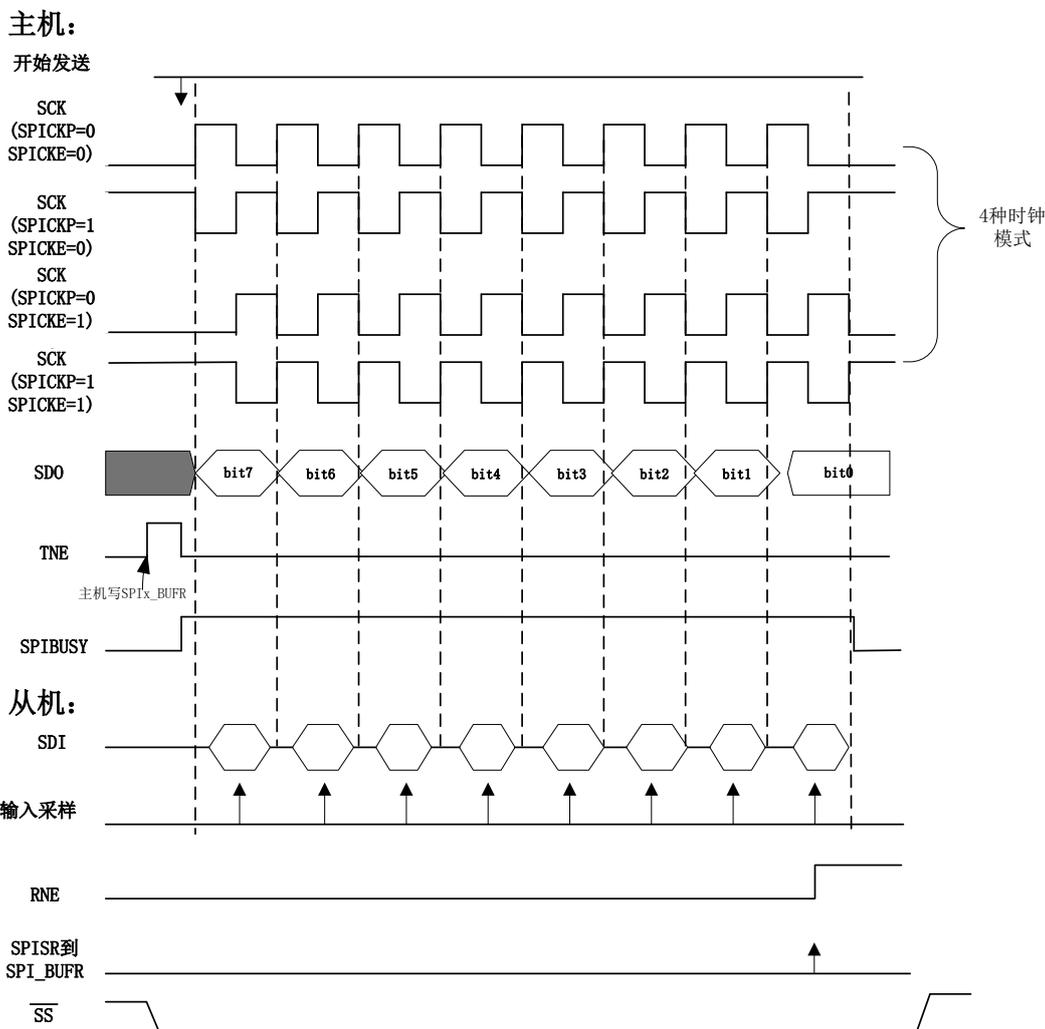


图 21-4 SPI 主控模式时序图（8 位数据）

21.3.4 从模式

21.3.4.1 从动模式

在从动模式下， \overline{SS} 为低时，当 SCK 引脚上出现外部时钟脉冲时发送和接收数据。当最后一位数据被锁存后，RNE 位置 1。在从动模式下，外部时钟由 SCK 引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。在休眠状态下，从控制器仍可发送/接收数据。当接收到一个字节时，器件从休眠状态唤醒。当读取 SPIx_BUF 后 RNE 会自动清零。

21.3.4.2 从动选择同步

\overline{SS} 引脚允许器件工作于同步从动模式。当 SPI 处于从动模式，如果 \overline{SS} 引脚置为 VDD 电平将使 SPI 模块复位。

当 SPI 模块复位时，位计数器被强制为 0。这可以通过强制将 \overline{SS} 引脚拉为高电平或将 SPIEN 位清零实现。

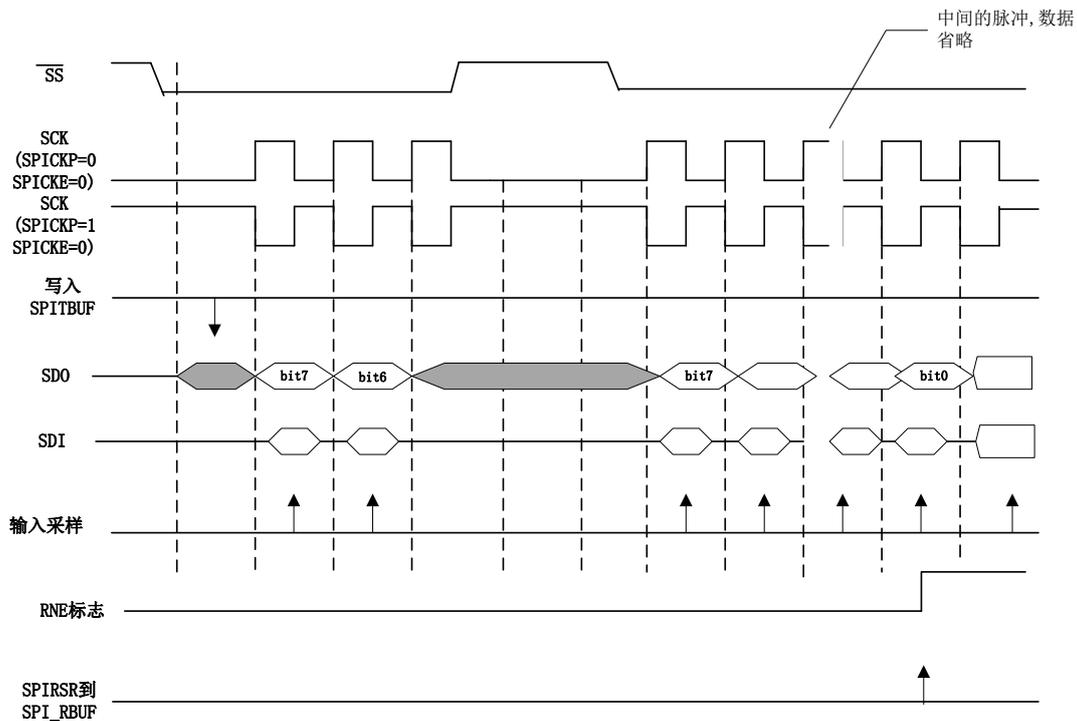


图 21-5 从动时序图 (8 位数据)

21.3.5 休眠模式和复位

21.3.5.1 休眠模式下的工作

在主动模式下，进入休眠模式后 SPI 工作时钟保持，在器件被唤醒前，将继续保持发送/接收状态。发送/接收完成后，SPI 中断标志位将置 1，如果此时该中断是允许的，还将唤醒器件。

在从动模式下，SPI 发送/接收移位寄存器与器件异步工作。这可以使器件在休眠状态时，仍可使数据被移入 SPI 发送/接收移位寄存器。当接收完一次完整的数据 (8 位、16 位或 32 位) 后，SPI 中断标志位将置 1，如果此时该中断是允许的，还将唤醒器件。

21.3.5.2 复位的影响

复位会禁止 SPI 模块并终止当前的数据传输。

21.3.6 SPI 工作模式设置

21.3.6.1 主动工作流程

主要步骤：

- (1) 通过 SPIx_CTLR 寄存器的 SPICKS<1:0>设置 SPI 工作时钟；
 - (2) 通过 SPIx_BRGR 设置工作时钟波特率；
 - (3) 通过 SPIMS<2:0>确定 SPI 主控方式
 - 000: SPI 主控方式, 时钟=工作时钟/4
 - 001: SPI 主控方式, 时钟=工作时钟/16
 - 010: SPI 主控方式, 时钟=工作时钟/64
 - 011: SPI 主控方式, 时钟= TxTRGO/2
 - (4) 通过寄存器 SPIx_CTLR 的 SPICKP 和 SPICKE 选择时钟与数据的时序关系；
 - (5) 将要发送的数据写到 SPIx_CTLR 寄存器；
 - (6) 如果需要中断, 则使能相应的中断使能位。
 - (7) 将 SPIEN 位置 1, 使能 SPI 模块；
 - (8) 如果同时在发送/接收数据, 在 RNE 位置 1 后可读取从机发送的数据。
- 注: 所有用到的 IO 及 SPI 模块都需将时钟使能并退出复位。

21.3.6.2 从动工作流程

主要步骤:

- (1) 通过 SPIMS<2:0>确定 SPI 为从动模式；
 - (2) 通过寄存器 SPIx_CTLR 的 SPICKP 和 SPICKE 选择时钟与数据的时序关系；与主控方的时钟时序要一致；
 - (3) 如果需要在从动模式下发送数据, 则将数据写入 SPIx_CTLR 寄存器；
 - (4) 如果需要中断, 则使能相应的中断使能位。
 - (5) 将 SPIEN 位置 1, 使能 SPI 模块
 - (6) 当接受到一个字节的数据后, RNE 自动置 1, 应立即读取 SPIx_BUFR 的值；
- 注: 所有用到的 IO 及 SPI 模块都需将时钟使能并退出复位。

21.4 I2S 操作

三线总线支持 2 个声道上音频数据的时分复用: 左声道和右声道。左声道总是先于右声道发送数据(CHSIDE 位在 PCM 协议下无意义)。

注: I2S 模式不管是发送还是接收, 都需先写一个数据到数据缓冲寄存器(SPIx_CTLR), 用于启动 I2S 通信。

21.4.1 I2S 飞利浦标准

在此标准下, 引脚 WS 用来指示正在发送的数据属于哪个声道。在发送第一位数据(MSB)前 1 个时钟周期, 该引脚即为有效。

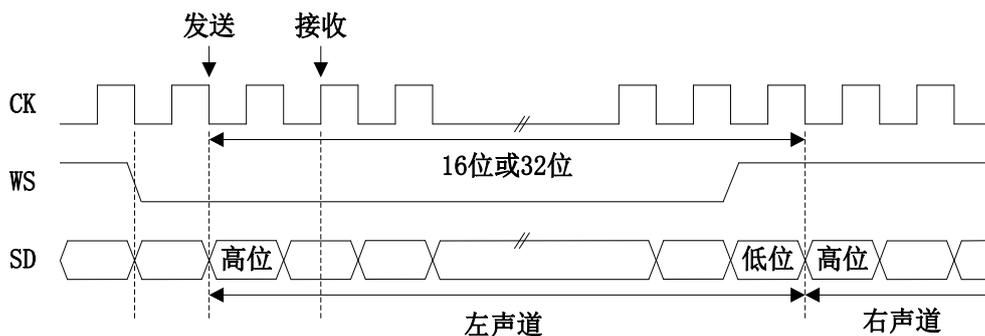


图 21-6 I2S 飞利浦协议波形(16/32 位全精度)

21.4.2 MSB 对齐标准

在此标准下，WS 信号和第一个数据位，即最高位(MSB)同时产生。

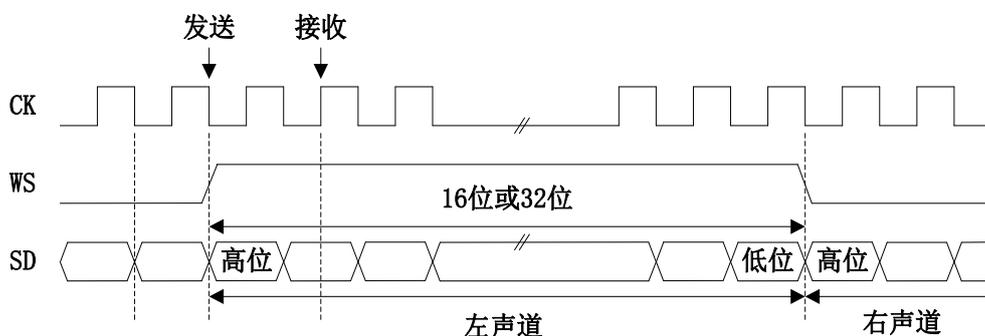


图 21-7 MSB 对齐 16 位或 32 位全精度

21.4.3 LSB 对齐标准

此标准与 MSB 对齐标准类似(在 16 位或 32 位全精度帧格式下无区别)。

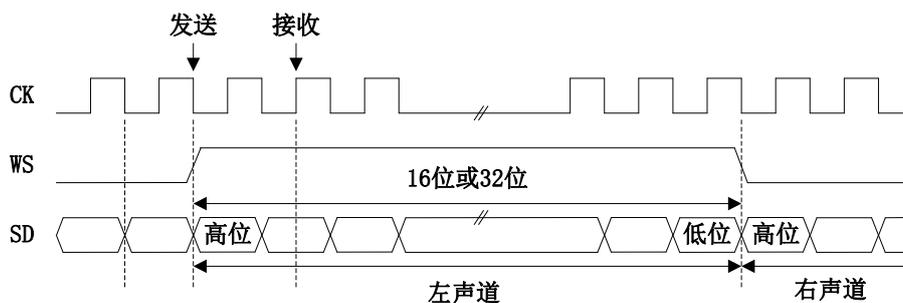
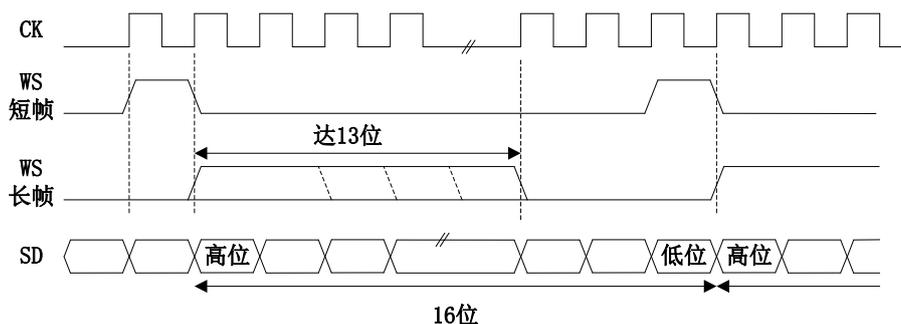


图 21-8 LSB 对齐 16 位或 32 位全精度

21.4.4 PCM 标准

在 PCM 标准下，不存在声道选择的信息。PCM 标准有 2 种可用的帧结构，短帧或者长帧，可以通过设置寄存器 SPIx_CTLR 的 PCMS 位来选择。


图 21-9 PCM 标准波形(16 位)

对于长帧，主模式下，用来同步的 WS 信号有效的固定为 13 位。

对于短帧，用来同步的 WS 信号长度只有 1 位。

21.4.5 时钟发生器

I2S 的比特率即确定了在 I2S 数据线上的数据流和 I2S 的时钟信号频率。

$$\text{I2S 比特率} = \text{每个声道的比特数} \times \text{声道数目} \times \text{音频采样频率} \quad \text{公式 21-3}$$

对于一个具有左右声道和 16 位音频信号，I2S 比特率计算如下：

$$\text{I2S 比特率} = 16 \times 2 \times F_s \quad \text{公式 21-4}$$

如果包长为 32 位，则有：

$$\text{I2S 比特率} = 32 \times 2 \times F_s \quad \text{公式 21-5}$$


图 21-10 音频采样频率定义

当 I2SCKS<1:0>=0x 时：

$$F_s = F_{ck} / [(16 \times 2) \times ((2 \times I2SDIV) + ODD)] \quad (\text{通道帧宽度为 16 位时}) \quad \text{公式 21-6}$$

$$F_s = F_{ck} / [(32 \times 2) \times ((2 \times I2SDIV) + ODD)] \quad (\text{通道帧宽度为 32 位时}) \quad \text{公式 21-7}$$

当 I2SCKS<1:0>=10 时：

$$F_s = F_{ck} / [(16 \times 2) \times ((2 \times I2SDIV) + ODD) \times 4] \quad (\text{通道帧宽度为 16 位时}) \quad \text{公式 21-8}$$

$$F_s = F_{ck} / [(32 \times 2) \times ((2 \times I2SDIV) + ODD) \times 4] \quad (\text{通道帧宽度为 32 位时}) \quad \text{公式 21-9}$$

当 I2SCKS<1:0>=11 时：

$$F_s = F_{ck} / [(16 \times 2) \times ((2 \times I2SDIV) + ODD) \times 8] \quad (\text{通道帧宽度为 16 位时}) \quad \text{公式 21-10}$$

$$F_s = F_{ck} / [(32 \times 2) \times ((2 \times I2SDIV) + ODD) \times 8] \quad (\text{通道帧宽度为 32 位时}) \quad \text{公式 21-11}$$

输出 MCK 时钟：

$$MCK = F_{ck} / ((2 \times I2SDIV) + ODD) \quad \text{公式 21-12}$$

21.4.6 状态标志位

有 4 个状态标志位供用户监控 I2S 总线的状态。

21.4.6.1 忙标志位(SPIBUSY)

SPIBUSY 标志由硬件设置与清除(写入此位无效果), 该标志位指示 I2S 通信层的状态。该位为 1 时表明 I2S 通讯正在进行中, 但有一个例外: 主接收模式(I2SMC<1:0>=11)下, 在接收期间 SPIBUSY 标志始终为低。

在软件要关闭 SPI 模块之前, 可以使用 SPIBUSY 标志检测传输是否结束, 这样可以避免破坏最后一次传输, 因此需要严格按照下述过程执行。

当传输开始时, SPIBUSY 标志被置为 1, 除非 I2S 模块处于主接收模式。

下述情况时, 该标志位被清除:

- 当传输结束时(除了主发送模式, 这种模式下通信是连续的);
- 当关闭 I2S 模块时。

当通信是连续的时候:

- 在主发送模式时, 整个传输期间, SPIBUSY 标志始终为高;
- 在从模式时, 每个数据项传输之间, SPIBUSY 标志在 1 个 I2S 时钟周期内变低。

注: 不要使用 SPIBUSY 标志处理每一个数据项的发送和接收, 最好使用 TNE 和 RNE 标志。

21.4.6.2 发送缓存非空标志位(TNE)

该标志位为 0 表示发送缓冲器为空, 可以对发送缓冲器写入新的待发送数据, 写 SPIx_BUFR 后该位为 1。在 I2S 被关闭时(SPIEN 位为 0), 该标志位清零。

21.4.6.3 接收缓存非空标志位(RNE)

该标志位置 1 表示在接收缓存里有接收到的有效数据。在读取寄存器 SPIx_BUFR 时, 该位清零。

21.4.6.4 声道标志位(CHSIDE)

在发送模式下, 该标志位在 TNE 为低时刷新, 指示从 SD 引脚上发送的数据所在的声道。如果在从发送模式下发生了下溢错误, 该标志位的值无效, 在重新开始通讯前需要把 I2S 关闭再打开。

在接收模式下, 该标志位在寄存器 SPIx_BUFR 接收到数据时刷新, 指示接收到的数据所在的声道。注意, 如果发生错误, 该标志位无意义, 需要将 I2S 关闭再打开(同时, 如果必要修改 I2S 的配置)。

在 PCM 标准下, 无论短帧格式还是长帧格式, 这个标志位都没有意义。

如果寄存器 SPIx_STR 的标志位 UDRIF 为 1, 且寄存器 SPIx_STR 的 UDRIE 位为 1, 则会产生中断。(中断源已经被清除后)可以通过 SPIx_STR 的 UDRIC 来清除中断标志。

21.4.7 错误标志位

I2S 单元有 1 个错误标志位, 下溢标志位 (UDRIF)。

在从发送模式下, 如果数据传输的第一个时钟边沿到达时, 新的数据仍然没有写入 SPIx_BUFR 寄存器, 该标志位会被置 1。在寄存器 SPI_CTLR 的 I2SMOD 位置 1 后, 该标志位才有效。如果寄存器 SPIx_STR 的 UDRIE 位为 1, 就会产生中断。

可以通过 SPIx_STR 的 UDRIC 来清除中断标志。

21.5 SPI 的 DMA 读写

DMA 写:

- (1) 配置 DMA 工作模式, DMA 的外设地址寄存器为 SPIx_BUFRR 地址, 数据方向 DDIR 位置 1 (从存储器读取, 写入外设);
- (2) 配置 SPIx 的工作模式, STXDE 位 (SPIx_STR<21>) 置 1, 使能 DMA 写请求;
- (3) 使能 SPIx 发送, 在启动发送后, TNE 清零, 自动启动 DMA 写 TBUFR。

DMA 读:

- (1) 配置 DMA 工作模式, DMA 的外设地址寄存器为 SPIx_BUFRR 地址, 数据方向 DDIR 位置清零 (从外设读取, 写入存储器);
- (2) 配置 SPI 工作模式, SRCDE 位 (SPIx_STR<20>) 置 1, 使能 DMA 读请求;
- (3) 在 SPIx 接收到一个数据后, RNE 置 1, 自动启动 DMA 读。

21.6 SPI 中断

SPI 的中断逻辑如下图所示:

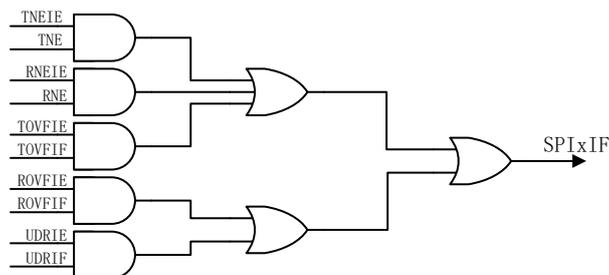


图 21-11 SPI 中断框图

SPI 外设模块到内核的中断标志 SPIxIF 由 SPI 模块内部的所有中断逻辑组合产生, 只要有一组中断标志及使能位都有效时 (置 1), SPIxIF 将置 1, 向内核申请中断, 中断是否响应由内核的中断使能位控制。

22 内部集成电路接口（I2C）

22.1 概述

I2C 特征：

- 多主机模式：可用作主设备或者从设备
- I²C 主设备产生时钟，起始和停止信号
- 检测 7 位和 10 位地址
- 支持 Fast Mode Plus 模式，最高速度可达 1Mbit/s
- 支持多地址识别
- 在监控模式下可观察所有的 I2C 总线通信量
- DMA 读写

I²C 模块能实现全部从动功能，且硬件支持启动位和停止位中断，以便于固件实现主控功能。I²C 模块实现标准模式规范以及 7 位和 10 位寻址。有两个引脚用于数据传输：时钟线（SCL）和数据线（SDA）。通过使能位 I2CEN 置 1 以使能 I2C 模块的功能。

22.2 I2C 模块结构框图

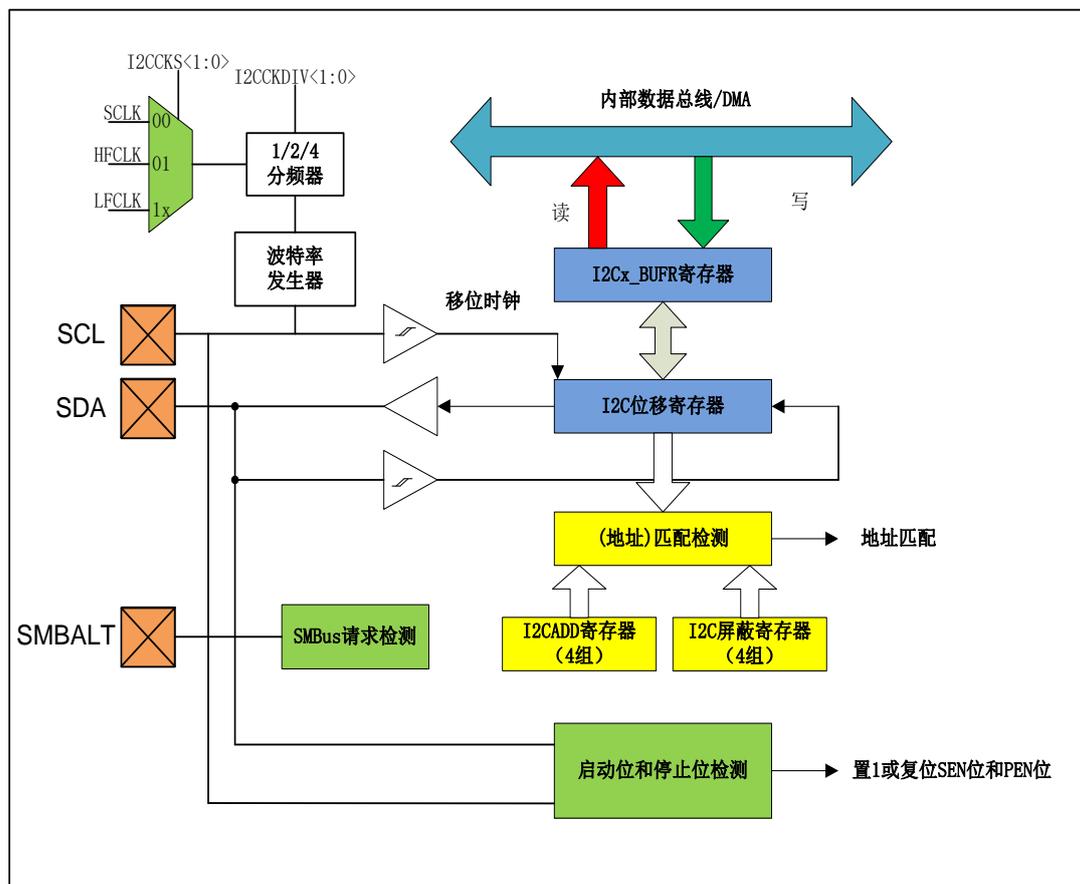


图 22-1 I2C 模块结构框图

22.2.1 I2C 引脚说明

SCL: I2C 的时钟引脚，由主机发送。

SDA: I2C 的数据引脚，可由主机或从机发送。

SMBALT: SMBUS 提醒线，为 SMBUS 可选引脚，由从机发送，提醒主机需要进行通信。

22.3 I2C 模块相关寄存器

 表 22-1 I2Cx 模块相关寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x000	I2Cx_CTLR	R/W	I2C 控制寄存器	0x0000_0000
0x004	I2Cx_SR	R/W	I2C 状态寄存器	0x0000_0000
0x008	I2Cx_BUFR	R/W	I2C 数据寄存器	0x0000_0000
0x00C	I2Cx_ADDR0	R/W	I2C 地址寄存器 0	0x0000_0000
0x010	I2Cx_BRGR	R/W	I2C 波特率寄存器	0x0004_0004
0x014	I2Cx_ADDR1	R/W	I2C 地址寄存器 1	0x0000_0000
0x018	I2Cx_ADDR2	R/W	I2C 地址寄存器 2	0x0000_0000
0x01C	I2Cx_ADDR3	R/W	I2C 地址寄存器 3	0x0000_0000
0x020	I2Cx_IER	R/W	I2C 中断使能寄存器	0x0000_0000

I2C0 基地址: 0x4000 0D00

I2C1 基地址: 0x4000 0D80

I2C2 基地址: 0x4000 0E00

I2C3 基地址: 0x4000 2380

22.3.1 I2Cx_CTLR I2C 控制寄存器

表 22-2 I2Cx_CTLR I2C 控制寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W	R				R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W		R/W		R/W				R/W	R/W	R/W	R/W	R/W	R/W
位名	ADRL				ALERT	ARPEN	SMBT	SMBUS		MENA	ENASCL	MTHALL	I2CCKDIV<1:0>	I2CCKS<1:0>			GCEN	ACKDT	ACKEN		PEN		SEN				DHCNT <3:0>			BADR10	I2CEN		

ADRL: 10 位地址的低位标志位。

0 = 无效

1 = 10 位地址低位有效

ALERT: SMBus 提醒。软件可以设置或清除该位(仅 SMBus 从机)。

0 = 释放 SMBALT 引脚使其变高

1 = 驱动 SMBALT 引脚使其变低

ARPEN: ARP 使能

0 = 禁止 ARP

1 = 使能 ARP

如果 SMBT=0, 使用 SMBus 设备的默认地址。

如果 SMBT=1, 使用 SMBus 的主地址。

SMBT: SMBUS 类型

0 = SMBus 设备

1 = SMBus 主机

SMBUS: SMBus 模式使能信号

¹ 本章除特殊说明, x=0,1,2,3

0 = I2C 模式

1 = SMBus 模式

NMENA: 监控模式使能位

0 = 不使能监控模式

1 = 使能监控模式

ENASCL: SCL 输出使能。当模块处于监控模式下，清零该位则 SCL 输出将被强制为高电平

0 = SCL 被强制为高

1 = 正常使用 SCL 线

MTHALL: 地址寄存器匹配位

0 = 只有在 I2CADDR 地址匹配时才会接收数据

1 = 可在任意接受的地址上产生中断，将使器件监控总线上的所有通信量

注：在监控模式下使能全地址匹配时，禁止将 I2C 设成十位地址模式，否则可能导致错误。

I2CCKDIV<1:0>: I2C 工作时钟分频比选择

00 = I2C 工作时钟分频比为 1:1

01 = I2C 工作时钟分频比为 1:2

1x = I2C 工作时钟分频比为 1:4

I2CCKS<1:0>: I2C 工作时钟选择寄存器

00 = 选用 SCLK 为 I2C 工作时钟

01 = 选用 HFCLK 为 I2C 工作时钟

1x = 选用 LFCLK 为 I2C 工作时钟

GCEN: 广播呼叫使能位（仅限 I2C 从动模式）

0 = 禁止广播呼叫地址

1 = 允许在 I2C 移位寄存器中接收到广播呼叫地址（0000h）时产生中断

注：需要注意的是，当 I2Cx_ADDRy 寄存器未被设置时，它们的复位值为 0x0000，也会使 I2C 模块对广播呼叫地址产生响应。

ACKDT: 应答数据位

0 = 应答

1 = 不应答

ACKEN: 应答序列使能位

0 = 应答序列空闲

1 = 在 SDA 和 SCL 引脚启动应答序列，发送 ACKDT 数据位。

PEN: 停止条件使能位（仅限 I²C 主控模式）

0 = 停止条件空闲

1 = 在 SDA 和 SCL 引脚启动停止条件。由硬件自动清零。

SEN: 启动条件使能位（仅限 I²C 主控模式）

0 = 启动条件空闲

1 = 在 SDA 和 SCL 引脚启动启动条件。由硬件自动清零。

DHCNT<3:0>: I2C 数据保持时间

0000 = 3 个 I2C 工作时钟

0001 = 4 个 I2C 工作时钟

.....

0111 = 10 个 I2C 工作时钟

1000 = 11 个 I2C 工作时钟

.....

1110 = 17 个 I2C 工作时钟

1111 = 18 个 I2C 工作时钟

BADR10: 10 位地址使能位

0 = I2C 模块使用 7 位地址

1 = I2C 模块使用 10 位地址

注：在十位地址模式下，高位地址匹配时，该地址不会存入 I2Cx_BUFR。而低 8 位地址或者七位地址模式下的地址将存入 I2Cx_BUFR。

I2CEN: I2C 使能位

0 = 禁能 I2C 模块

1 = 使能 I2C 模块

22. 3. 2 I2Cx_SR I2C 状态寄存器

表 22-3 I2Cx_SR I2C 状态寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W														R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W			
位名														BERRIF	ITXDF	IRCDF						TXBE	RCBF	ISIF	SMBDF	SMBHIF	SMBALF	RNW	ARBLIF	AFIF	DATA	ADD10	ADDR	PIF	SIF

BERRIF: I2C 总线错误标志位

0 = 未产生 I2C 总线错误

1 = 产生了 I2C 总线错误

ITXDF: 请求 DMA 写 I2C 缓存寄存器标志位

0 = 请求 DMA 写 I2C 缓存寄存器标志未产生

1 = 产生了请求 DMA 写 I2C 缓存寄存器标志

注：该位在 DMA 响应后自动清 0。

IRCDF: 请求 DMA 读 I2C 缓存寄存器标志位

0 = 请求 DMA 读 I2C 缓存寄存器标志未产生

1 = 产生了请求 DMA 读 I2C 缓存寄存器标志

注：该位在 DMA 响应后自动清 0

TXBE: 发送 BUFF 状态位，写 I2Cx_BUFR 清零该位

0 = 不需要写 I2Cx_BUFR

1 = 等待写 I2Cx_BUFR

该位只在发送情况下置 1，且需要使能发送或者应答成功。

RCBF: 接收 BUFF 为满，读 I2Cx_BUFR 清零该位

0 = 接收 BUFF 为空

1 = 接收 BUFF 为满

注：如果之前接收 BUFF 中有数据没有读出，而此时去写 I2Cx_BUFR 寄存器的话，将清零该位。

ISIF: I2C 中断信号标志位，该信号由硬件自动置 1，需要手动清零来继续 I2C 操作。（写零清零，写一无效）

0 = 未产生 I2C 中断信号标志位

1 = 产生了 I2C 中断信号标志位

注: 需要发送的数据需在 ISIF 置 1 之后写入 I2Cx_BUFRR, 否则可能导致 TXBE 信号错误。

SMBDIF: SMBus 设备默认地址(从模式) (SMBus device default address (Slave mode))。

0 = 未收到 SMBus 设备的默认地址;

1 = 当 ARPEN=1 时, 收到 SMBus 设备的默认地址。

SMBHIF: SMBus 主机头系列(从模式) (SMBus host header (Slave mode))。

0 = 未收到 SMBus 主机的地址;

1 = 当 SMBT=1 且 ARPEN=1 时, 收到 SMBus 主机地址。

SMBAIF: SMBus 提醒中断标志(SMBus alert), 软件可以设置或清除该位。

SMBus 主机模式下:

0 = 无 SMBus 提醒;

1 = 在引脚上产生 SMBus 提醒事件。

SMBus 从机模式下:

0 = 没有 SMBus ALERT 响应地址头序列;

1 = 收到 SMBus ALERT 响应地址头序列。

RNW: 读/ 写信息状态位 (仅从机)

该位用来保存在上次地址匹配后的 RNW 位信息。此位仅在地址匹配与遇到下一个启动位或停止位之间有效。

0 = 写

1 = 读

ARBLIF: 失去仲裁标志位

0 = 发送数据过程中未失去仲裁

1 = 发送数据过程中失去仲裁

AFIF: 应答错误标志位 (该位只对发送方有效)

0 = 未发生应答错误

1 = 发生了应答错误

DATA: 数据内容状态位, 总线上出现起始位将清零该状态位 (仅限 I2C 从动模式)

0 = 表示上次接收或发送的字节是地址

1 = 表示上次接收或发送的字节是数据

7 位地址下默认为起始位后接收的第 2 个字节开始。十位地址下默认为起始位后接收的第 3 个字节开始。

ADD10: 高位地址状态位

0 = 上次接收或发送的字节不是高位地址

1 = 上次接收或发送的字节是高位地址

ADDR: 地址匹配状态位, 总线上出现停止位将清零该状态位

0 = 从机未收到对应地址

1 = 从机收到匹配地址且应答

PIF: 停止信号标志位

0 = 总线上未出现停止位

1 = 总线上出现了停止位

SIF: 起始信号标志位

0 = 总线上未出现起始位

1 = 总线上出现了起始位

22. 3. 3 I2Cx_BUFR I2C 缓存寄存器

表 22-4 I2Cx_BUFR I2C 缓存寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																					
R/W																							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																							I2CBUF<9:0>																														

I2CBUF<9:0>: I2C 数据缓存寄存器

22. 3. 4 I2Cx_ADDRy I2C 地址寄存器 (y=0,1,2,3)

表 22-5 I2Cx_ADDRy I2C 地址寄存器 (y=0,1,2,3)

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名							I2CMSK<9:0>										I2CADD<9:0>																

I2CMSK<9:0>: I2C 地址屏蔽位

I2CMSK<n> = 0: 接收到的地址的第 n 位与 I2CADD<n>比较以检测 I²C 的地址匹配情况

I2CMSK<n> = 1: 接收到的地址的第 n 位不用于检测 I²C 的地址匹配情况

I2CADD<9:0>: I2C 地址位

注: 每一个 I2C 模块有 4 个 I2CADDR, 只需要和其中一个匹配就算匹配成功。

22. 3. 5 I2Cx_BRGR I2C 波特率寄存器

表 22-6 I2Cx_BRGR I2C 波特率寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	I2CBRGL<15:0>															I2CBRGH<15:0>																	

I2CBRGL<15:0>: SCL 低电平占用的时钟周期数

I2CBRGH<15:0>: SCL 高电平占用的时钟周期数

注: 由于有滤波电路在输入通道上, 将波特率寄存器设为比 0x0003_0003 更小的值时, 可能导致 I2C 模块出现错误。

22. 3. 6 I2Cx_IER I2C 中断使能寄存器

表 22-7 I2Cx_IER I2C 中断使能寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0			
R/W														R/W	R/W	R/W	R/W	R/W								R/W	R/W				R/W	R/W	R/W		
位名														BERRIE	ITXDE	IRCDE								ISIE	SMBDIE	SMBHIE	SMBAIE		ARBLIE	AFIE				PIE	SIE

BERRIE: 总线错误中断使能位

- 0 = 未使能总线错误中断
- 1 = 使能总线错误中断

ITXDE: DMA 写 I2C 缓存寄存器使能位

- 0 = 未使能 DMA 写 I2C 缓存寄存器
- 1 = 使能 DMA 写 I2C 缓存寄存器

IRCDE: DMA 读 I2C 缓存寄存器使能位

- 0 = 未使能 DMA 读 I2C 缓存寄存器
- 1 = 使能 DMA 读 I2C 缓存寄存器

ISIE: I2C 中断信号使能位

- 0 = 未使能 I2C 中断信号中断位
- 1 = 使能了 I2C 中断信号中断位

SMBDIE: SMBDIF 中断使能位

- 0 = 未使能 SMBDIF 中断
- 1 = 使能 SMBDIF 中断

SMBHIE: SMBHIF 中断使能位

- 0 = 未使能 SMBHIF 中断
- 1 = 使能 SMBHIF 中断

SMBAIE: SMBAIF 中断使能位

- 0 = 未使能 SMBAIF 中断
- 1 = 使能 SMBAIF 中断

ARBLIE: 失去仲裁中断使能位

- 0 = 未使能失去仲裁中断
- 1 = 使能失去仲裁中断

AFIE: 应答错误中断使能位

- 0 = 未使能应答错误中断
- 1 = 使能了应答错误中断

PIE: 停止信号中断使能

- 0 = 未使能停止信号中断
- 1 = 使能停止信号中断

SIE: 起始信号中断使能

- 0 = 未使能起始信号中断
- 1 = 使能起始信号中断

22.4 功能描述

22.4.1 从动模式

在从动模式下，SCL 引脚和 SDA 引脚必须被配置为重映射模式，设置为开漏输出并使能上拉功能。

当地址匹配或在地址匹配后发送的数据被接收时，如果设置了 ACKEN (I2Cx_CTLR<12>) 为 1，硬件会自动产生一个应答 (ACK) 脉冲，并把当时移位寄存器中接收到的值装入 I2Cx_BUFR 寄存器。应答脉冲的值由 ACKDT (I2Cx_CTLR<13>) 决定。

在从动模式下，地址匹配/在地址匹配后发送的数据被接收/总线上出现重新起始位时，ISIF (I2Cx_SR<11>) 信号将置高同时将拉低 SCL 线，直到软件清零该状态位。

BERRIF 状态标志

在传输地址或数据期间检测到外部停止位或起始位时，会出现此错误（同时将 ISIF 置 1）。从机检测到 BERRIF 总线错误标志时，会停止数据传输并释放总线。如果此时 BERRIF 为 1，将产生总线错误中断。

22.4.1.1 寻址

一旦 I2C 模块被使能，它就会等待启动条件发生。启动条件发生后，外部输入的 8 位数据被移入 I2C 移位寄存器。I2C 模块在时钟 (SCL) 线的上升沿采样所有的输入位。在第 8 个时钟 (SCL) 脉冲的下降沿 I2C 移位寄存器<7:1>的值会和 I2C 地址寄存器的值比较。如果地址匹配，会发生下列事件：

- (1) I2C 移位寄存器的值被装入 I2Cx_BUFR 寄存器。
- (2) 缓冲器满标志位 RCBF 被置 1。
- (3) 产生 ACK 脉冲。
- (4) 在第 9 个 SCL 脉冲的上升沿，I2C 中断标志位 ISIF 被置 1（如果允许中断，则产生中断）。

在 10 位地址模式下，从控制器需要收到两个地址字节。第一个地址字节的高 5 位将指定这是否是一个 10 位地址。第一个地址的最低位必须为 0 指定写操作，这样从机才能接收到第二个地址字节。对于 10 位地址，第一个字节等于“1111 0 A9 A8 0”，其中 A9 和 A8 是该地址的两个最高有效位。

10 位地址的工作步骤如下，其中 5-8 步是针对从动发送器而言的：

- (1) 接收地址的第一个（高）字节（ISIF 位置 1，RCBF 并不会置 1，因为这个字节并不会存入 I2Cx_BUFR）。
 - (2) 将标志位 ISIF 清零，释放 SCL 线。
 - (3) 接收地址的第二个（低）字节（ISIF 位和 RCBF 位置 1）。
 - (4) 读取 I2Cx_BUFR (RCBF 位清零)，清零标志位 ISIF，释放 SCL 线。
 - (5) 接收重复启动条件（ISIF 位置 1）。
 - (6) 将标志位 ISIF 清零，释放 SCL 线。
 - (7) 接收地址的第一个（高）字节（ISIF 位和 RCBF 位置 1）。
 - (8) 注意：这里的高字节的最低位为 1，表示读从机。
 - (9) 读 I2Cx_BUFR (RCBF 位清零)，并清零标志位 ISIF，释放 SCL 线。
- 注：所有用到的 IO 及 I2C 模块都需将时钟使能并退出复位。

22.4.1.2 接收

当地址字节的 R/W 位为零并发生地址匹配时，I2Cx_SR 寄存器中的 RNW 位清零。接收到的地址被装入 I2Cx_BUFR 寄存器。

每个数据传输字节都会产生 I2C 中断信号 ISIF。标志位 ISIF 必须用软件清零。通过 I2Cx_SR 寄存器可以确定该字节的状态。

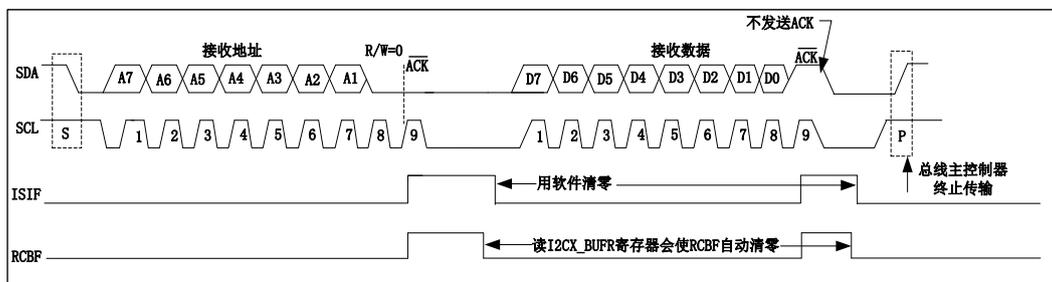


图 22-2 七位地址接收模式

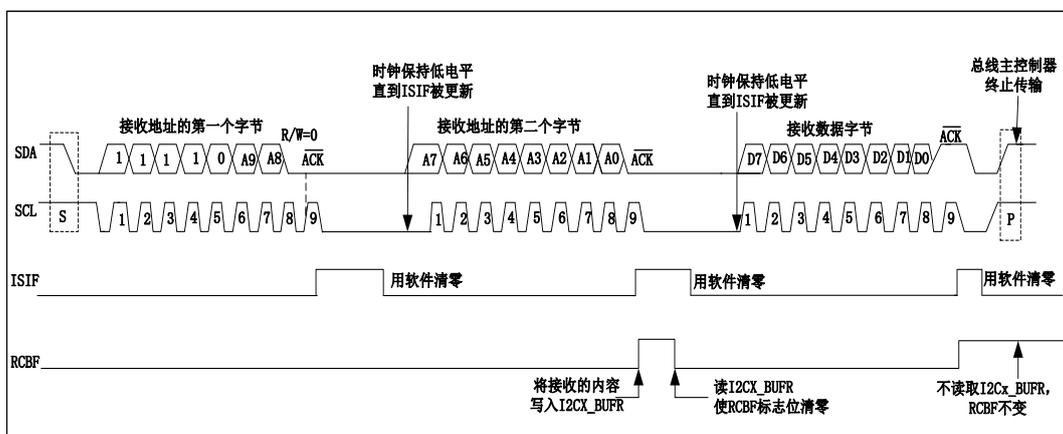


图 22-3 十位地址接收模式

特别注意，十位地址的第一个地址字节不会存入 I2Cx_BUFR 也不会置高 RCBF 位。

从动七位地址接收设置：

设置 SDA 和 SCL 引脚为重映射模式，开漏输出并使能上拉功能；

设置 I2Cx_CTLR 寄存器，通过 I2CCKS 选择工作时钟，BADR10 选择七位地址模式，使能 ACKEN；

设置 I2Cx_ADDRy 寄存器，设置从机地址；

如果需要中断打开各中断使能位；

使能 I2CEN，开始接收数据，等待地址匹配；如果地址匹配，则 ISIF 和 RCBF 置 1；

读取 I2Cx_BUFR 中的数据硬件将自动清零 RCBF 位，软件清除 ISIF 信号继续接收数据；

之后每次收到数据之后 ISIF 和 RCBF 都将置 1，读取数据之后 RCBF 自动清零，需要软件清除 ISIF 信号来释放 SCL 线主机才能继续发送数据；

注：所有用到的 IO 及 I2C 模块都需将时钟使能并退出复位。

从动十位地址接收设置：

设置 SDA 和 SCL 引脚为重映射模式，开漏输出并使能上拉功能；

设置 I2Cx_CTLR 寄存器，通过 I2CCKS 选择工作时钟，BADR10 选择十位地址模式，使能 ACKEN；

设置 I2Cx_ADDRy 寄存器，设置从机地址；

如果需要中断打开各中断使能位；

使能 I2CEN，开始接收数据，等待地址匹配；如果地址匹配，则 ISIF 置 1；

软件清除 ISIF 信号继续接收数据；

之后每次收到数据之后 ISIF 和 RCBF 都将置 1，读取数据之后 RCBF 自动清零，需要软件清除 ISIF 信号来释放 SCL 线主机才能继续发送数据；

注：所有用到的 IO 及 I2C 模块都需将时钟使能并退出复位。

22.4.1.3 发送

当输入地址字节的 R/W 位（最低位）置 1 并发生地址匹配时，I2Cx_SR 寄存器的 RNW 位被置 1。接收到的地址被装入 I2Cx_BUFR 寄存器。ACK 脉冲在第 9 位上发送，SCL 引脚保持低电平。

每个数据传输字节都会产生 I2C 中断，标志位 ISIF 必须用软件清零。标志位 ISIF 在第 9 个时钟脉冲被置 1。对于从发送器，来自接收器的 ACK 脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。若 SDA 线为高电平（无 ACK 应答信号），则表示数据传输已完成。如果 SDA 线为低电平（ACK），则必须将下一个要发送的数据装入 I2Cx_BUFR 寄存器。

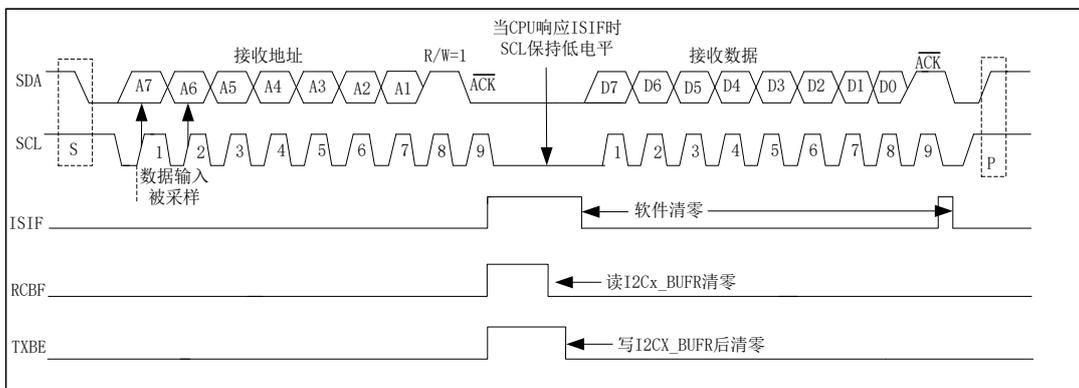


图 22-4 七位地址发送模式

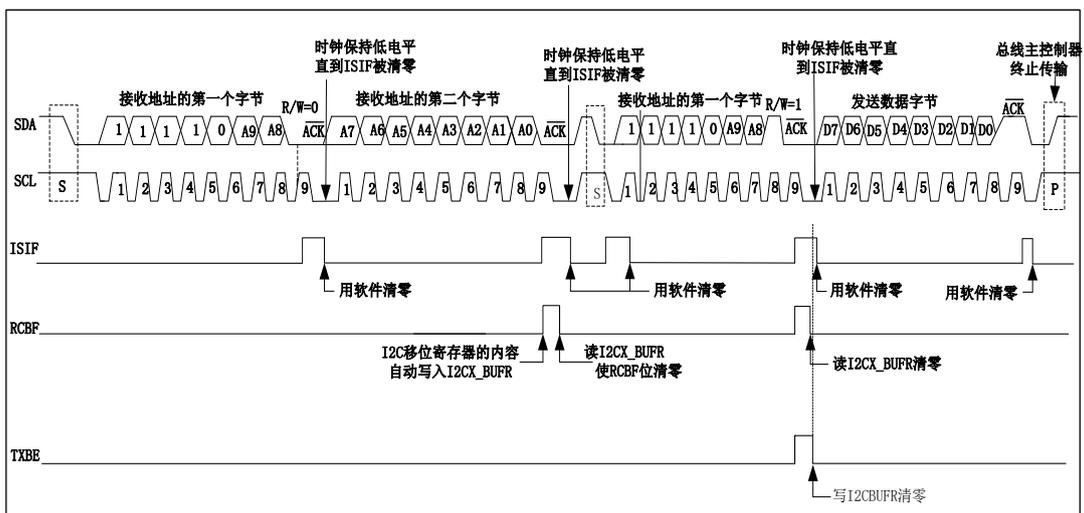


图 22-5 十位地址发送模式

注：从机接收到重复启动条件且当前的应答位为 0 时 ISIF 置 1

从动七位地址发送设置：

- (1) 设置 SDA 和 SCL 引脚为重映射模式，开漏输出并使能上拉功能；
- (2) 设置 I2Cx_CTLR 寄存器，通过 I2CCKS 选择工作时钟，BADR10 选择七位地址模

式，使能 ACKEN，根据需要选择 ACKDT 的值；

- (3) 设置 I2Cx_ADDRy 寄存器，设置从机地址；
- (4) 等待主机发送从机地址，并且读写位为 1，硬件将自动置高 ISIF；
- (5) 如果需要，先读取 I2Cx_BUFR 中接收到的地址值（可忽略）；此时 TXBE 状态位为 1，往 I2Cx_BUFR 中写入需要发送的数据，硬件将清除该位；
- (6) 软件清零 ISIF 位，开始发送数据；
- (7) 对于从发送器，来自主接收器的 ACK 脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。若 SDA 线为高电平（无 ACK 应答信号），则表示数据传输已完成。如果 SDA 线为低电平（ACK），则必须将下一个要发送的数据装入 I2Cx_BUFR 寄存器。

从动十位地址发送设置：

- (1) 设置 SDA 和 SCL 引脚为重映射模式，开漏输出并使能上拉功能；
- (2) 设置 I2Cx_CTLR 寄存器，通过 I2CCKS 选择工作时钟，BADR10 选择十位地址模式，使能 ACKEN，根据需要选择 ACKDT 的值，通过 I2Cx_ADDRy 设置从机地址；
- (3) 等待主机发送地址第一字节且读写位为 0，如果地址匹配，硬件将自动置高 ISIF；
- (4) 软件清零 ISIF。等待主机发送地址第二字节，如果地址匹配，硬件将自动置高 ISIF；
- (5) 软件清零 ISIF。等待检测到重复启动条件，硬件将自动置高 ISIF；
- (6) 软件清零 ISIF。等待主机发送地址第一字节且读写位为 1，如果地址匹配，硬件将自动置高 ISIF；
- (7) 如果需要，先读取 I2Cx_BUFR 中接收到的地址值（可忽略）；此时 TXBE 状态位为 1，往 I2Cx_BUFR 中写入需要发送的数据，硬件将清除该位；
- (8) 软件清零 ISIF 位，开始发送数据；
- (9) 对于从发送器，来自主接收器的 ACK 脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。若 SDA 线为高电平（无 ACK 应答信号），则表示数据传输已完成。如果 SDA 线为低电平（ACK），则必须将下一个要发送的数据装入 I2Cx_BUFR 寄存器。

注：所有用到的 IO 及 I2C 模块都需将时钟使能并退出复位。

22.4.1.4 广播呼叫地址支持

在 I²C 总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有的器件都应该发送一个应答响应。

广播呼叫地址是根据 I²C 协议为特定目的保留的八个地址之一。它由全 0 组成，且 R/W = 0。广播呼叫使能位 GCEN（I2Cx_CTLR<14>）使能时，即可识别广播呼叫地址。

如果与广播呼叫地址匹配，I2C 移位寄存器的值将传输到 I2Cx_BUFR，RCBF 标志位置 1，并且 ISIF 中断标志位在第 9 位（ACK 位）的置 1。

当响应中断时，可以通过读取 I2Cx_BUFR 的内容来判断中断源。该值可以用于判断地址是特定器件的还是一个广播呼叫地址。

如果 GCEN 位置 1 时采样到广播呼叫地址，同时从器件被配置为 10 位地址模式，不再需要地址的后半部分，从器件将在应答后开始接收数据如下图所示。

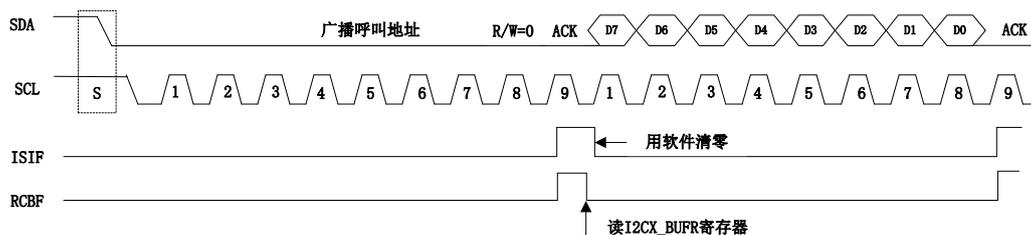


图 22-6 从动模式广播呼叫模式

如果使用 I2C 模块充当主机来发送一个广播呼叫，需要将 I2C 设置为七位地址模式。

22.4.2 主控模式

在主机模式下，SCL 引脚和 SDA 引脚必须被配置为重映射模式，设置为开漏输出并启用上拉功能。在以下几种情况下，标志位 ISIF 置 1（如果 ISIE 为 1，则产生中断）：

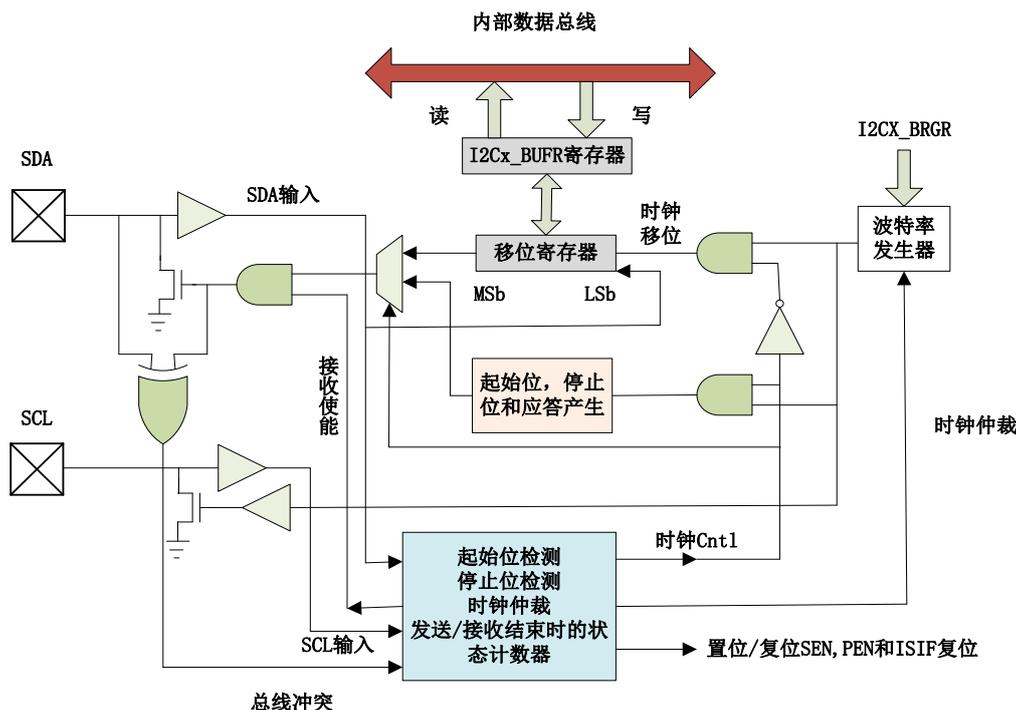
- (1) 启动条件
- (2) 停止条件
- (3) 发送/接收到数据传输字节
- (4) 重复启动条件

BERRIF 状态标志

在传输地址或数据期间检测到外部停止位或起始位时，会出现此错误（同时将 ISIF 置 1）。主机检测到 BERRIF 总线错误标志时，会停止数据传输并释放总线。如果此时 BERRIF 为 1，将产生总线错误中断。

22.4.2.1 主控模式进入

通过将 I2Cx_CTLR 中相应的 I2CEN 位置 1 并使能 SEN 发送起始位，即可占领总线进入主控模式。



注：I/O 引脚具有连接到 VDD 和 VSS 的保护二极管

图 22-7 主控模式结构框图

22.4.2.2 I2C 主模式操作

所有串行时钟脉冲和启动/停止条件均由主器件产生。停止条件或重复启动条件能结束传输。因为重复启动条件也是下一次串行传输的开始，因此不会释放 I²C 总线。在主控发送器模式下，串行数据通过 SDA 输出，而串行时钟由 SCL 输出。发送的第一个字节包括接收器件的地址（7 位）和读/写（R/W）位。串行数据每次发送 8 位。每发送一个字节，会收到一个应答位。启动和停止条件的输出表明串行传输的开始和结束。

在 I²C 模式下，波特率发生器将 SCL 时钟频率设置为 100kHz、400kHz 或 1MHz。波特率发生器的重载值位于 I2Cx_BRGR。当发生对 I2Cx_BUFRR 的写操作时，波特率发生器将自动开始计数。

下面是一个典型的发送事件序列：

- (1) 用户通过将启动使能位 SEN（I2Cx_CTLR 寄存器）置 1 产生启动条件。
- (2) 在进行任何其他操作前，I2C 模块将等待所需的启动时间，ISIF 位置 1。
- (3) 用户将从器件地址装入 I2Cx_BUFRR 进行发送（R/W=0），并清除 ISIF 位。
- (4) 地址从 SDA 引脚移出，直到发送完所有 8 位为止。
- (5) I2C 模块移入来自从器件的 ACK 位，AFIF 位将指示收到的是 ACK 还是 NACK。
- (6) I2C 模块在第 9 个时钟周期将 ISIF 位置 1。
- (7) 用户将 8 位数据装入 I2Cx_BUFRR，清除 ISIF 位。
- (8) 数据从 SDA 引脚移出，直到发送完所有 8 位为止。
- (9) I2C 模块移入来自从器件的 ACK 位。
- (10) I2C 模块在第 9 个时钟将 ISIF 位置 1。
- (11) 在 ISIF 置 1 的情况下，将 PEN 位（I2Cx_CTLR<10>）置 1，然后清除 ISIF 位，产生停止位。
- (12) 一旦停止条件完成，ISIF 和 PIF（I2Cx_SR<1>）将置 1。

如果在步骤(3)，主机在发送完从器件地址并且 R/W=1 时，表明主机准备接收数据。按上述完成步骤(6)后：

- (1) 清除 ISIF 位，等待从机释放 SCL（从机清 ISIF 位）。
- (2) 从机释放 SCL 后，主机开始接收数据。接收完 8 位数据后，主机根据需要发送或不发送应答位 ACK，ISIF 位置 1。
- (3) 如果应答位有效（ACK=0），主机继续接收数据，将 ISIF 位清零；如果应答位无效（ACK=1），将 PEN 位置 1，然后清除 ISIF 位，产生停止位。

注：所有用到的 IO 及 I2C 模块都需将时钟使能并退出复位。

22.4.2.3 波特率发生器

在 I2C 主控模式下，波特率发生器的重载值位于 I2Cx_BRGR。该寄存器由两部分组成：I2CBRGH 和 I2CBRGL。

$$TBRG = I2CBRGH + I2CBRGL \quad \text{公式 22-1}$$

当装载了该值后，波特率发生器将自动开始计数并递减至 0，然后停止直到下次重载为止。在 I2C 主控模式下，会自动重载 BRG。例如，在发生时钟仲裁时，BRG 将在 SCL 引脚采样到高电平时重载。如下图所示。

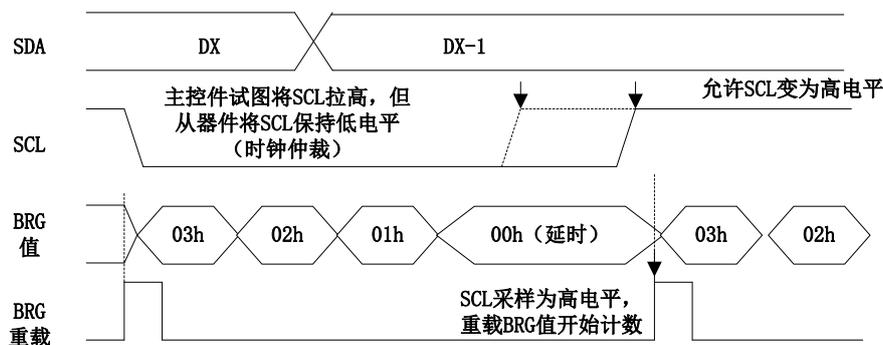


图 22-8 带有仲裁的波特率发生时序

22.4.2.4 I2C 主控模式启动条件时序

要发起启动条件，用户应将 I2Cx_CTLR 寄存器的启动条件使能位 SEN 置 1。

如果在启动条件开始时，SDA 和 SCL 引脚已经采样为低电平，或者在启动条件期间，SCL 在 SDA 线被驱动为低电平之前已经采样为低电平，则会发生总线冲突。总线冲突中断标志位 BCLIF 置 1，启动条件中止，I2C 模块复位到空闲状态。

在启动条件结束之前，不能对 I2Cx_CTLR 或其他寄存器进行写操作。

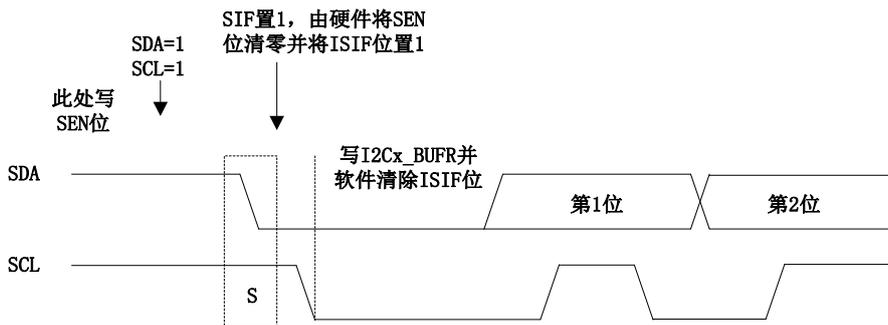


图 22-9 第一个启动位时序

22.4.2.5 I2C 主控模式重复启动条件时序

发送一个启动条件之后,在发送停止条件之前再次发送一个启动条件即为重复启动条件。和启动条件一样,只需要在 SDA 和 SCL 均为高电平时,使能 SEN 就可以发送。

注:

- (1) 有任何其他事件进行时,对 SEN 的编程无效。
- (2) 在重复启动条件期间,下列事件将会导致总线冲突:

当 SCL 由低电平变为高电平时,SDA 采样为低电平。

在 SDA 被拉低之前,SCL 变为低电平。这表示可能有另一个主器件正尝试发送数据 1。

- (3) 使用重复启动时,需要在 ISIF 置 1 的情况下(其他事件完成),再将 SEN 置 1,然后清除 ISIF 位,产生重复启动信号。

一旦 ISIF 位被置 1,用户便可以在 7 位地址模式下将 7 位地址写入 I2Cx_BUF,或者在 10 位地址模式下写入默认的 1 个地址字节。

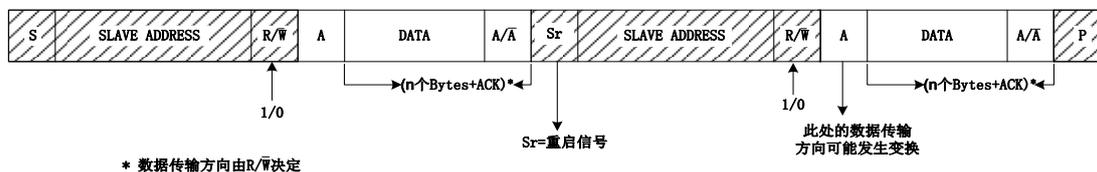


图 22-10 I2C 协议复合数据帧

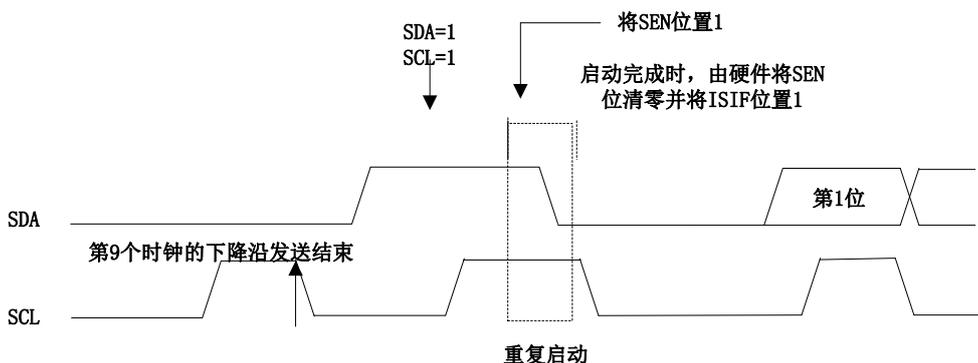


图 22-11 重复启动时序

重复启动后,七位地址模式主机还是要发送原先的地址,但是最低位的读/写从机位可以改变。十位地址模式只需要发送原先的高字节地址,同样最低位的读/写从机位可以改变。

22.4.2.6 I2C 主控模式发送

I2C 协议中 SDA 引脚上的数据不仅来源于 I2Cx_BUFRR 寄存器，还需要按 I2C 协议的规定发送起始位、停止位、应答位等信号。下图所示是 I2C 协议典型写数据帧格式。

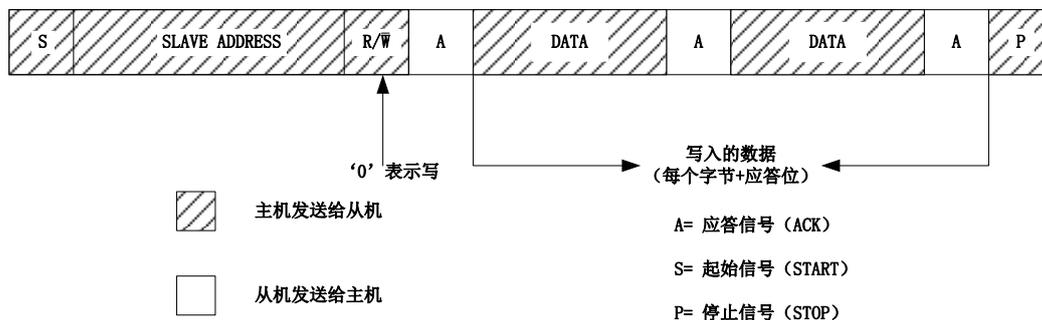


图 22-12 I2C 协议典型写数据帧格式

TXBE 状态标志

在发送模式下，TXBE 位 (I2Cx_SR 寄存器) 在 CPU 写 I2Cx_BUFRR 时清零，在所有 8 位数据移出后置 1。

AFIF 状态标志

在发送模式下，当从器件发送应答响应 (ACK=0) 时，AFIF 位 (I2Cx_SR 寄存器) 为 0；当从器件没有应答或应答错误 (ACK=1) 时，该位置 1。

注：若主机发送完一个字节后收到的应答标志位 AFIF=1，则应及时停止传输（通过发送停止信号 PEN 位来实现）。

下图所示为 I2C 协议的典型写数据帧格式在单片机硬件 I2C 上的实现过程时序图。

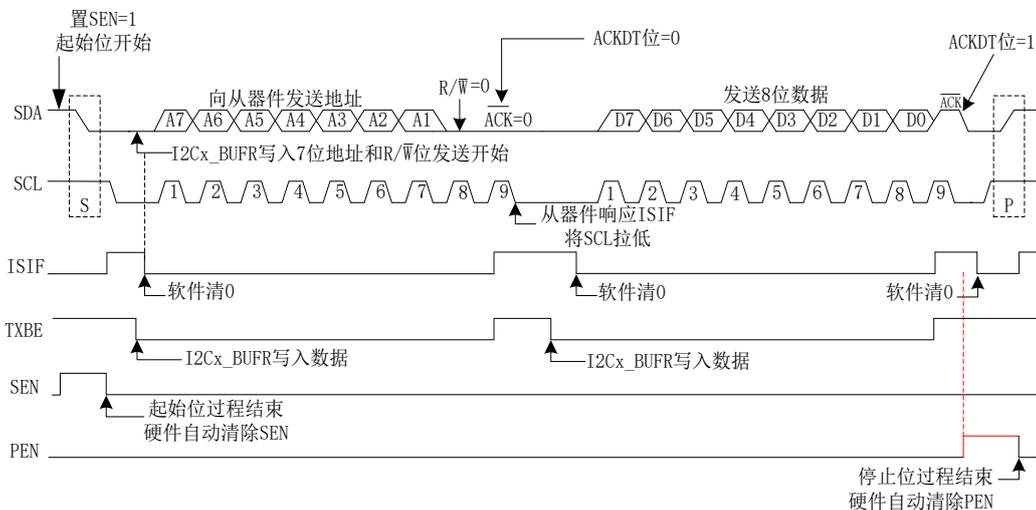


图 22-13 I2C 主模式七位地址发送时序

具体操作过程请参考“I2C 主模式操作”。

22.4.2.7 I2C 主控模式接收

I2C 主模式数据接受的系统结构与主模式数据发送系统结构相同，但是数据接收流程与数据发送流程不同。如下图所示为 I2C 协议典型读数据帧格式。

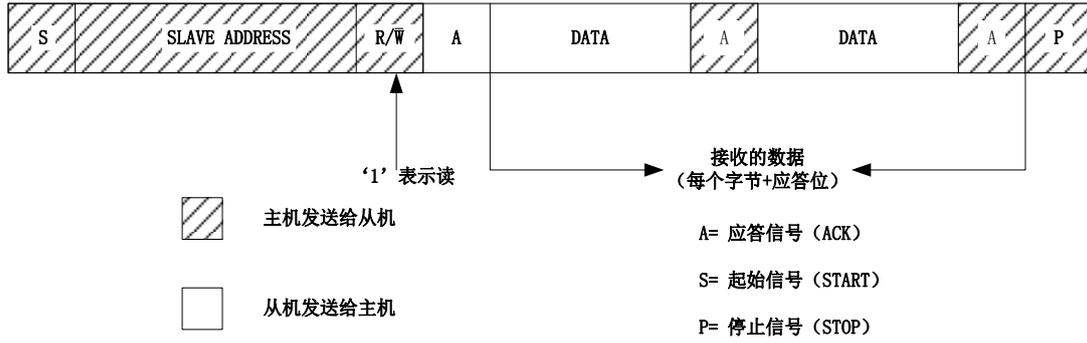


图 22-14 I2C 协议典型读数据帧时序

如果主机发送地址时，R/W 为 1，主机将自动使能接收。

ACKDT 和 ACKEN 位同时设置为 1，则数据接收完成之后，I2C 移位寄存器的内容装入 I2Cx_BUFER，RCBF 标志位置 1，ISIF 标志位置 1，SCL 保持为低电平。

RCBF 状态标志

接收时，当将地址或数据字节从 I2C 移位寄存器装入 I2Cx_BUFER 时，RCBF 位置 1；在读 I2Cx_BUFER 寄存器时 RCBF 位清零。

下图为典型读数据帧格式在单片机硬件 I2C 上的实现接收过程时序图。

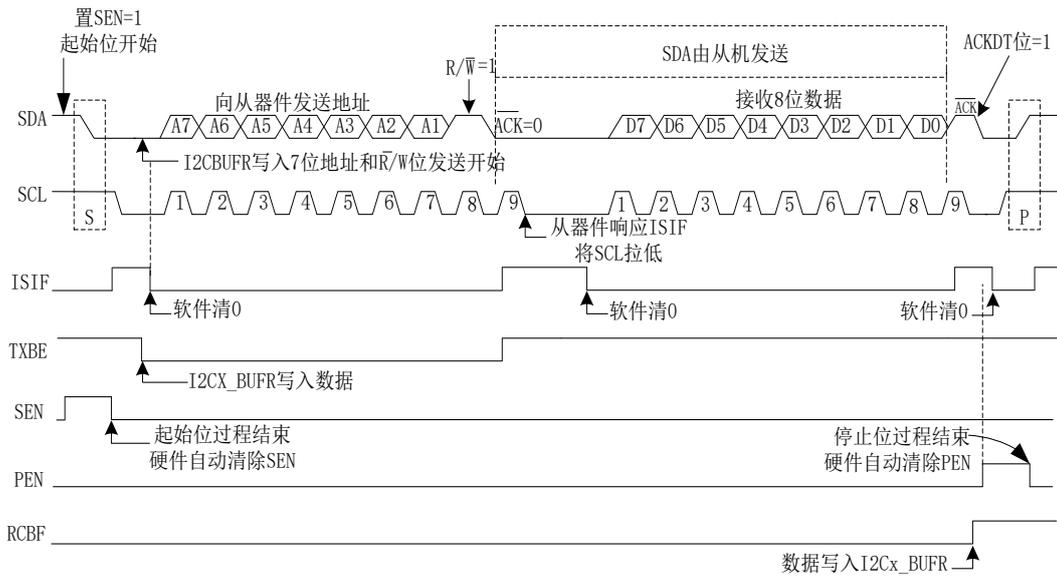


图 22-15 I2C 主模式接收数据时序图

具体操作过程请参考“I2C 主模式操作”。

22.4.2.8 停止条件序列

在接收/发送结束时，通过置高 PEN (I2Cx_CTLR 寄存器)，SDA 引脚将产生一个停止位。

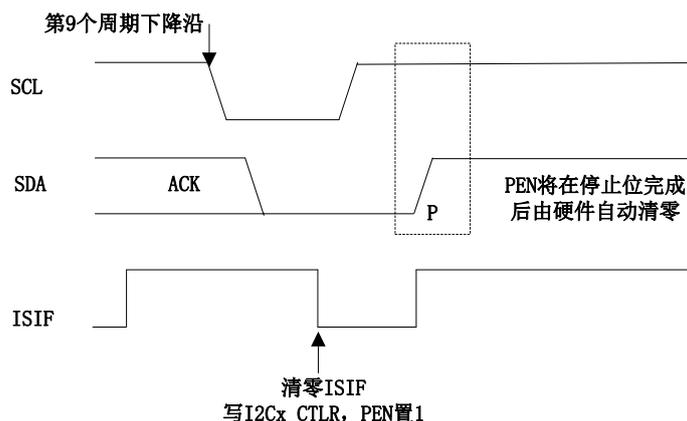


图 22-16 发送模式停止条件

22.4.3 时钟仲裁

如果在任何接收、发送或重复启动/停止条件期间，主器件拉高了 SCL 引脚（允许 SCL 引脚悬空为高电平），就会发生时钟仲裁。如果主器件 SCL 为高，而实际采样 SCL 为低时，波特率发生器（BRG）将暂停计数，直到实际采样到 SCL 引脚为高电平为止。当 SCL 引脚采样为高电平时，波特率发生器中将被重新装入 I2Cx_BRGR 的值（I2CBRGL 和 I2CBRGH 之和）并开始计数。这可以保证当外部器件将时钟拉低时，SCL 始终保持至少一个 I2Cx_BRGR 周期的高电平。

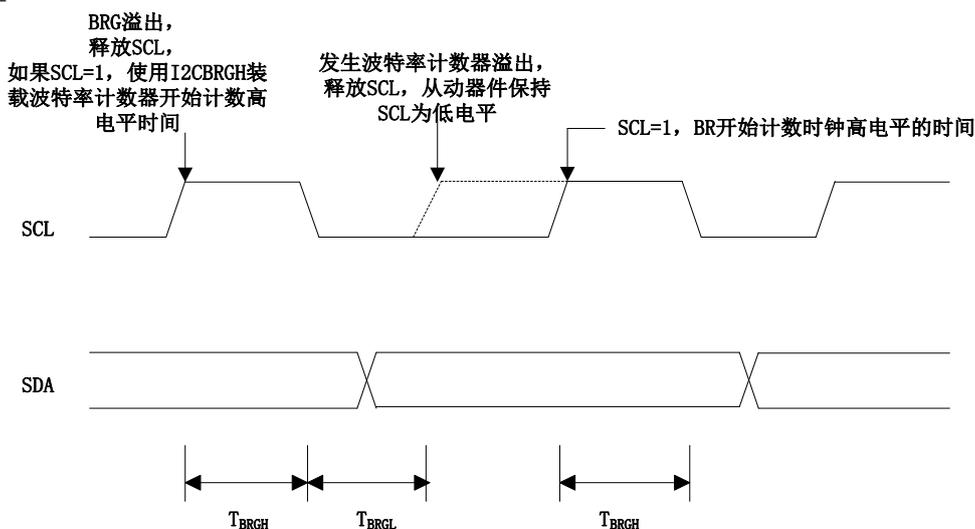


图 22-17 主控发送模式下的时钟仲裁时序

22.4.4 多主控制器模式

在多主控制器模式下，在检测到启动条件和停止条件时产生的中断可用于判断总线是否空闲。停止标志位（PIF）和启动标志位（SIF）需要软件清零。当 PIF 位（I2Cx_SR<1>）置 1 时，可以获得 I²C 总线的控制权；否则，PIF 位和 SIF 位都清零，总线处于空闲状态。当总线处于忙状态且允许 PIF 中断时，一旦发生停止条件便产生 PIF 中断。

在多主控制器操作中，必须监视 SDA 线以确定信号电平是否为所需的输出电平。此检查仅需在输出为高电平时进行。如果期望输出高电平，但检测到的是低电平，器件就需要释放 SDA 和 SCL 线。此仲裁在以下状态可能会失败：

- 地址传输

- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

当使能从动逻辑电路时，从控制器将继续接收数据。如果在地址传输阶段仲裁失败，可能表示与器件的通信正在进行中。如果寻址到器件，则将会产生一个 ACK 脉冲。如果在数据传输阶段仲裁失败，则器件需要在以后重新传输数据。

22.4.4.1 多主机通信，总线冲突与总线仲裁

多主机模式是通过总线仲裁来支持的。当主器件将地址/数据位输出到 SDA 引脚时，如果一个主器件通过将 SDA 引脚悬空为高电平以在 SDA 上输出 1，而另一个主器件输出 0，就会发生总线仲裁。如果 SDA 引脚上期望的数据是 1，而实际在 SDA 引脚上采样到的数据是 0，则发生了丢失仲裁。主器件将把总线冲突中断标志位 ARBLIF 置 1，并将 I2C 端口复位到空闲状态。如果在发送过程中发生总线冲突，则发送停止，SDA 线被拉高。当执行完总线冲突中断服务程序后，如果 I2C 总线空闲，用户可通过发出启动条件恢复通信。如果在启动、重复启动、停止或应答条件的进行过程中发生总线冲突，则该条件被中止，SDA 和 SCL 线被拉高，I2Cx_CTLR 寄存器中的对应控制位清零。当执行完总线冲突中断服务程序后，如果 I2C 总线空闲，用户可通过发出启动条件恢复通信。在多主机模式下，通过在检测到启动和停止条件时产生中断可以确定总线何时空闲。PIF 位置 1 时，可以获取 I2C 总线的控制权。

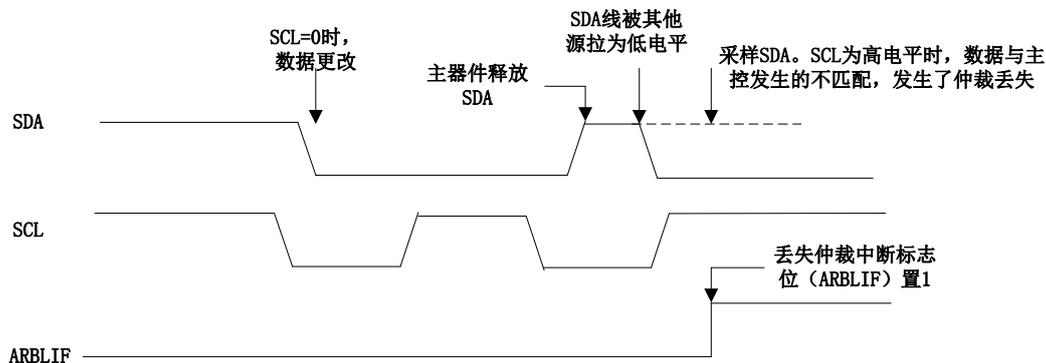


图 22-18 I2C 发送和应答时的总线冲突时序

22.4.5 I2C 屏蔽寄存器

在 I2C 从动模式下，I2C 屏蔽 (I2CMSK<9:0>) 寄存器用于在地址比较操作下屏蔽 I2CADD<9:0>中的值。I2CMSK<9:0>中某位为 1 会使 I2CADD<9:0>中相应的位成为“无关位”。

此寄存器在任何复位条件发生时均复位为全 0，因此，在写入屏蔽值前，它对标准 I2C 操作没有影响。

必须在通过设置 BADR10 位以选择 I2C 从动模式 (7 位或 10 位地址) 之前对此寄存器进行初始化。只有通过 I2Cx_CTLR 的 BADR10 位选择了适当的模式后才可访问此寄存器。I2C 屏蔽寄存器在以下情况下有效：

- 7 位地址模式：与 A<7:1>进行地址比较。
- 10 位地址模式：仅与 A<7:0>进行地址比较。I2C 屏蔽在接收到地址的第一个 (高) 字节期间无效。

22. 4. 6 监控模式

使能 NMENA 位 (I2Cx_CTLR 寄存器), I2C 模块将进入监控模式。在该模式下, 将 ACKEN 位设为 1 且 ACKDT 为 0 才能正常工作, SDA 输出将被强制为高电平。这可防止 I2C 模块向 I2C 数据总线输出任何类型的数据 (包括 ACK)。根据 ENASCL 位状态, 也可以将 SCL 强制为高电平, 以防止模块控制 I2C 时钟线。当 MTHALL 位置 1 且 I2C 处于监控模式时, 可在任意接收的地址上产生中断。这将使器件监控总线上的所有通信量。

监控模式下的中断:

当模块处于监控模式时所有 ISIF 中断将正常出现。这意味着检测到地址匹配时就会产生第一个中断 (如果 MTHALL 位置位, 则接收到任意地址都会产生中断, 否则只有在地址与地址寄存器 (MSK 可正常工作) 匹配时才会产生中断)。

检测地址匹配后, 从机每接收/发送到一个字节就会产生中断。在从机情况下, 数据寄存器实际上包含了总线上其它从机发送的数据, 这些从机实际上是被主机寻址的。需要注意的是, 监控模式下接收七位地址或者十位地址的低八位也会产生中断。

所有 ISIF 中断产生后, 处理器可读数据寄存器以查看总线上实际发送的数据。

22. 4. 7 SMBus

22. 4. 7. 1 介绍

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: 用来发送命令、产生时钟和终止发送的设备。主机: 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。一个系统里只允许有一个主机。

SMBus 和 I2C 之间的相似点:

2 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;

SMBus 和 I2C 之间的不同点

下表列出了 SMBus 和 I2C 的不同点。

表 22-8 SMBUS 与 I2C 的比较

SMBus	I2C
最大传输速度 100kHz	最大传输速度 400kHz
最小传输速度 10kHz	无最小传输速度
35ms 时钟低超时	无时钟超时
固定的逻辑电平	逻辑电平由 VDD 决定
不同的地址类型(保留的、动态的等)	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

22.4.7.2 SMBus 应用用途

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。

设备标识

在系统管理总线上，任何一个作为从模式的设备都有一个唯一的地址，叫做从地址。保留的从地址表请参考 2.0 版的 SMBus 规范(<http://smbus.org/specs/>)。

总线协议

SMBus 技术规范支持 9 个总线协议。有关这些协议的详细资料和 SMBus 地址类型，请参考 2.0 版的 SMBus 规范(<http://smbus.org/specs/>)。这些协议由用户的软件来执行。

地址解析协议(ARP)

通过给每个从设备动态地分配一个新的唯一地址，可以解决 SMBus 的从地址冲突。地址解析协议(ARP)具有以下特性：

- 使用标准 SMBus 物理层仲裁机制分配地址；
- 当设备维持供电期间，分配的地址仍保持不变，也允许设备在断电后保留其地址。
- 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。

唯一的设备标识符(UDID)

为了分配地址，需要一种区分每个设备的机制，每个设备必须拥有一个唯一的设备标识符。关于在 ARP 上 128 位的 UDID 的详细信息，参考 2.0 版的 SMBus 规范(<http://smbus.org/specs/>)。

22.4.7.3 SMBus 提醒模式

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。SMBALT 和 SCL、SDA 信号一样，是一种线与信号。SMBALT 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，可以通过设置 I2Cx_CTLR 寄存器上的 ALERT 位，使用 SMBALT 给主机发信号表示它希望进行通信。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 SMBALT 设备。只有那些将 SMBALT 拉低的设备能应答 ARA。此状态是由 I2Cx_SR 寄存器中的 SMBAIF 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，第八位可以是“0”或“1”。

如果多个设备把 SMBALT 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 SMBALT，如果当信息传输完成后，主机仍看到 SMBALT 低，就知道需要再次读 ARA。

没有实现 SMBALT 信号的主机可以定期访问 ARA。

有关 SMBus 提醒模式的更多详细资料，请参考 2.0 版的 SMBus 规范(<http://smbus.org/specs/>)。

超时错误：

在定时规范上 I2C 和 SMBus 之间有很多差别。

SMBus 定义了一个时钟低超时，35ms 的超时。SMBus 规定 TLOW: SEXT 为从设备的累积时钟低扩展时间。SMBus 规定 TLOW: 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/>)。

需通过软件程序来处理时钟低超时错误。

22.4.7.4 如何使用 SMBus 模式的接口

为了从 I2C 模式切换到 SMBus 模式，应该执行下列步骤：

- 设置 I2Cx_CTLR 寄存器中的 SMBUS 位；
- 按应用要求配置 I2Cx_CTLR 寄存器中的 SMBT 和 ARPEN 位。

如果要把设备配置成主设备，产生起始条件的步骤见 I2C 主模式。否则，参见 I2C 从模式。软件程序必须处理多种 SMBus 协议。

- 如果 ARPEN=1 且 SMBT=0，使用 SMB 设备默认地址。
- 如果 ARPEN=1 且 SMBT=1，使用 SMB 主设备头字段。
- 如果 ALERT=1，使用 SMB 提醒响应地址。

22.4.8 I2C 的 DMA 读写

DMA 写：

- (1) 配置 DMA 工作模式，DMA 的外设地址寄存器为 I2Cx_BUFRR 地址，数据方向 DDIR 位置 1（从存储器读取，写入外设）；
- (2) 配置 I2Cx 的工作模式，ITXDE 位（I2Cx_IER<17>）置 1，使能 DMA 写请求；
- (3) 使能 I2Cx，I2C 产生起始条件后（TXBE 置 1，需要 I2C 发送数据）时自动启动 DMA 写 I2Cx_BUFRR。
- (4) 在 DMA 完成写操作后，会自动将 I2C 的所有标志（SIF、PIF、AFIF、ARBLIF、SMBDIF、SMBHIF、SMBAIF 和 ISIF）全部清零。

DMA 读：

- (1) 配置 DMA 工作模式，DMA 的外设地址寄存器为 I2Cx_BUFRR 地址，数据方向 DDIR 位置 0（从外设读取，写入存储器）；
- (2) 配置 I2C 工作模式，IRCDE 位（I2Cx_IER<16>）置 1，使能 DMA 读请求；
- (3) 在 I2Cx 接收到一个数据后，RCBF 置 1，自动启动 DMA 读。
- (4) 在 DMA 完成读操作后，会自动将 I2C 的所有标志（SIF、PIF、AFIF、ARBLIF、SMBDIF、SMBHIF、SMBAIF 和 ISIF）全部清零。

22.4.9 I2C 中断

I2C 的中断逻辑如下图所示：

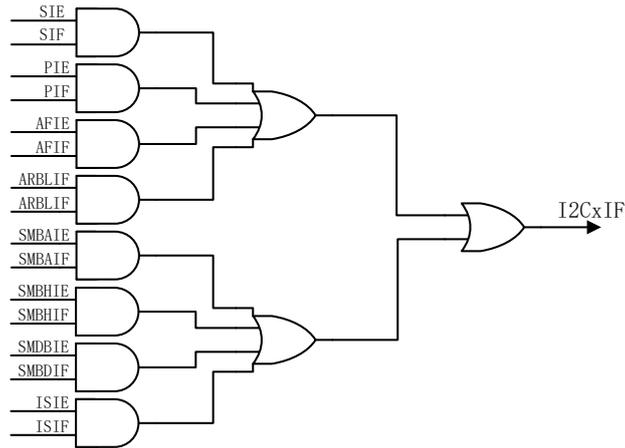


图 22-19 I2C 中断框图

I2C 外设模块到内核的中断标志 I2CxIF 由 I2C 模块内部的所有中断逻辑组合产生，只要有一组中断标志及使能位都有效时（置 1），I2CxIF 将置 1，向内核申请中断，中断是否响应由内核的中断使能位控制。

23 EXIC 接口

23.1 概述

EXIC 接口是用于并口通讯，最高支持 18bit 的并行数据输出，支持 DMA 读写支持 LCD8080 接口，可用于与专用的 LCD 驱动芯片进行通讯。

EXIC 模块框图如下：

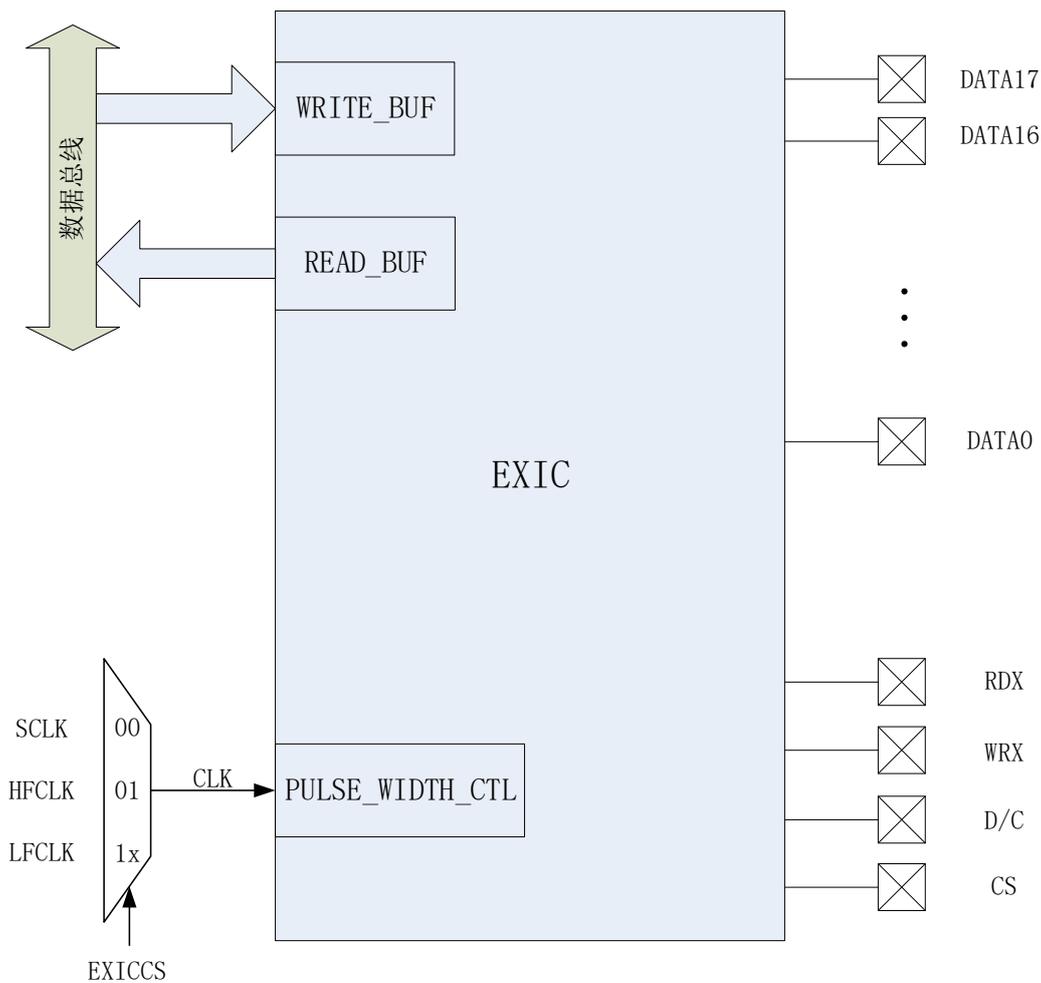


图 23-1 EXIC 模块框图

- DATA17~DATA0: 数据脚;
- RDX: 读使能;
- WRX: 写使能;
- D/C: 数据/命令使能;
- CS: 片选使能;

23.2 EXIC 相关寄存器

表 23-1 EXIC 相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	EXIC_CTL0	R/W	EXIC 控制寄存器 0	0x0000 0000
0x004	EXIC_CTL1	R/W	EXIC 控制寄存器 1	0x0000 0000
0x008	EXIC_STATE	R/W	EXIC 状态寄存器 1	0x0000 0000
0x00C	保留	R	保留	0x0000 0000
0x010	EXIC_WRITEBUF	R/W	EXIC 写数据缓存	0x0000 0000
0x014	EXIC_READBUF	R	EXIC 读数据缓存	0x0000 0000

基地址：0x4000 2800

23.2.1 EXIC_CTL0 EXIC 控制寄存器 0

表 23-2 EXIC_CTL0 EXIC 接口控制寄存器 0

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W	R/W	R/W	R/W	R/W											R/W	R/W							R/W	R/W	R/W	R/W	R/W	R/W					
位名	MODESEL	DATATR	READIEN	WRITEIEN											RENDIE	WENDIE							RDMAEN	TDMAEN				READEN	WRITEEN	DCSEL	CSEN	EXICEN	

EXICEN: EXIC 接口使能位

- 0 = 禁止 EXIC 模块
- 1 = 使能 EXIC 模块

D/CSEL: 数据/命令选择位

- 0 = 传输数据
- 1 = 传输命令

WRITEEN: 写数据使能¹

- 0 = 无动作
- 1 = 写一组数据到子机

READEN: 读数据使能

- 0 = 无动作
- 1 = 从子机读一组数据

TDMAEN: DMA 写使能

- 0 = 关闭 DMA 传输写数据
- 1 = 使能 DMA 传输写数据

RDMAEN: DMA 读使能

- 0 = 关闭 DMA 传输读数据
- 1 = 使能 DMA 传输读数据

WENDIE: 写完成中断使能

- 0 = 关闭写完成中断

¹ WRITEEN 与 READEN 在置 1 后会自动清零，无须手动清零；其读写时间（低电平期间）由脉宽控制寄存器控制，此外在数据传输期间（低电平期间）这两位不能被置 1。

- 1 = 使能写完成中断
- RENDIE: 读完成中断使能
 - 0 = 关闭读完成中断
 - 1 = 使能读完成中断
- WRITE1EN: 写数据使能¹
 - 0 = 无动作
 - 1 = 写一组数据到子机
- READ1EN: 读数据使能
 - 0 = 无动作
 - 1 = 从子机读一组数据
- DATATR: 数据方向选择
 - 0 = 数据输出
 - 1 = 数据输入
 - (仅在 MODE_SEL 为 1 时有效)
- MODESEL: 模式选择位
 - 0 = 使能 WRITE_EN 与 READ_EN 控制
 - 1 = 使能 WRITE1_EN 与 READ1_EN 控制

23. 2. 2 EXIC_CTL1 EXIC 控制寄存器 1

表 23-3 EXIC_CTL1 EXIC 接口控制寄存器 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W																								R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W
位名																								PULSEWIDTH <3:0>							EXICCS <1:0>			

- EXICCS<1:0>: EXIC 工作时钟源选择位
 - 00 = EXIC 工作时钟源为 SCLK
 - 01 = EXIC 工作时钟源为 HFCLK
 - 1x = EXIC 工作时钟源为 LFCLK
- PULSEWIDTH<3:0>: EXIC 读写脉宽 (Tpulse) 选择位
 - 0000 = 1Tclk
 - 0001 = 2Tclk
 - 0010 = 4Tclk
 - 0011 = 8Tclk
 - 0100 = 16Tclk
 - 0101 = 32clk
 - 0110 = 64clk
 - 0111 = 128clk
 - 1000 = 256Tclk
 - 1001 = 512Tclk

¹ WRITE1EN 与 READ1EN 在置 1 后不会自动清零，需要手动清零；其读写时间（低电平期间）由软件控制。

1010 = 1024Tclk
 1011 = 2048Tclk
 1100 = 4096Tclk
 1101 = 8192Tclk
 1110 = 16384Tclk
 1111 = 32768Tclk

23. 2. 3 EXIC_STAT 状态寄存器
表 23-4 EXIC_STAT EXIC 状态寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W																											R/W	R/W				
位名																											RENDF	WENDF				

WENFIF: 写完成标志位

0 = 写操作未完成或未进行写操作
 1 = 写操作完成

RENDF: 读完成标志位

0 = 读操作未完成或未进行写操作
 1 = 读操作完成

注:

- (1) 读/写完成是指读/写脉冲出现上升沿;
- (2) 该寄存器写“0”清零, 写 1 无效;
- (3) 在 MODE=1 时该寄存器无效;

23. 2. 4 EXIC_WRITEBUF 写数据缓存寄存器
表 23-5 EXIC_WRITEBUF 写数据缓存寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0												
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
R/W															R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名															WRITEBUF<17:0>																													

WRITEBUF<17:0>: 写数据缓存寄存器

23. 2. 5 EXIC_READBUF 读数据缓存寄存器

表 23-6 EXIC_READBUF 读数据缓存寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	R 0 17	R 0 16	R 0 15	R 0 14	R 0 13	R 0 12	R 0 11	R 0 10	R 0 9	R 0 8	R 0 7	R 0 6	R 0 5	R 0 4	R 0 3	R 0 2	R 0 1	R 0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W															R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位名															READBUF<17:0>																	

READBUF<17:0>: 读数据缓存寄存器

23.3 功能描述

EXIC 接口数据传输时序如下，首先使能片选使能（CSX），根据传输数据的类型（数据/命令）置位(或清零)D/CX，然后通过使能读写使能进行数据的读和写。

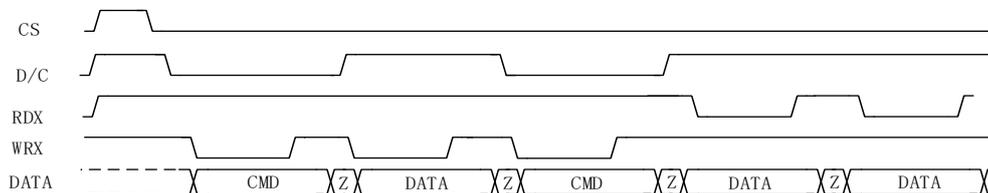


图 23-2 EXIC 数据传输时序

23.3.1 写功能

主机通过 EXIC 接口向 LCD 显示器写数据，包含 3 个控制信号（D/CX, RDX, WRX），和一组数据信号，数据在写信号为低电平（WRX）的时候进行传输，传输时序如下：

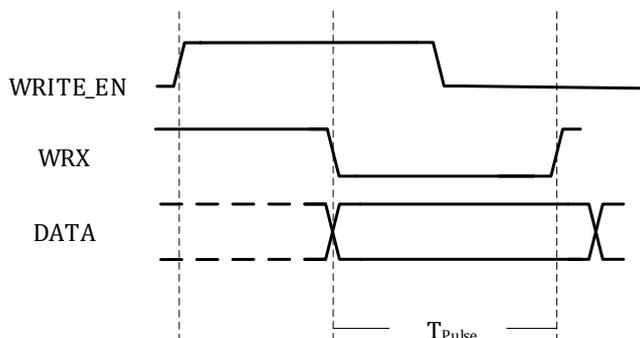


图 23-3 EXIC 写时序

写脉冲宽度为 T_{pulse} ，在使能主机写使能寄存器（WRITE_EN）后，约 $1/2T_{pulse}$ 后，写使能寄存器被硬件清零，此时写脉冲产生，数据在此时由写缓存寄存器传输至输出寄存器，开始数据传输，然后数据在写脉冲的上升沿写入下位机（LCD 驱动模块），完成数据传输。

（注在使用 DMA 传输时，脉冲宽度设置应不低于 4 个系统时钟）

23.3.2 读功能

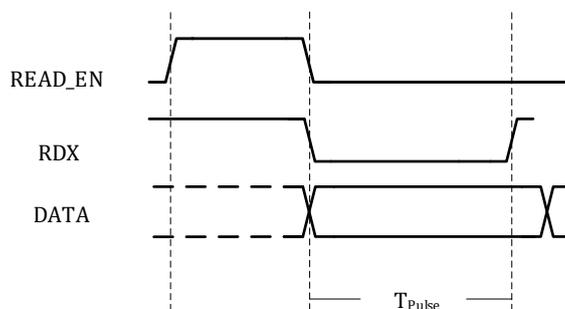


图 23-4 EXIC 读时序

如上图，读脉冲宽度为 T_{pulse} ，在使能主机读使能寄存器（READ_EN）后，约 $1/2T_{pulse}$ 后，写使能寄存器被硬件清零，此时写脉冲产生，数据在此时由 LCD 驱动模块传输数据线，

开始数据传输，然后数据在读脉冲的上升沿读入到 EXIC 的接口模块。

（注在使用 DMA 传输时，脉冲宽度设置应不低于 4 个系统时钟）

23.3.3 EXIC 接口 DMA 使用方法

EXIC 接口可以使用 DMA 缓存接收数据或发送数据，具体用法如下：

- (1) 使能设置 DMA 模块，使能 EXIC 对应的数据通道（详见章节 9 DMA 控制器）
- (2) 根据数据方向（发送/接收），使能 EXIC_CTL0 寄存器的 TDMAEN/RDMAEN；
- (3) 发送 DMA 缓存的数据，或读取 DMA 缓存的接收数据。

23.3.4 EXIC 接口使用方法

- (1) 通过 EXIC_CTL1 寄存器 EXICCS<1:0>位选择时钟源，并配置 PULSEWIDTH<3:0>，选择合适的读写脉冲宽度；
- (2) 使能 EXIC 接口，使能片选信号，根据数据类型使能数据/命令选择
- (3) 将数据写入写缓存（如果读数据，忽略此步）
- (4) 使能读/写使能
- (5) 通过读/写状态位判断是否完成读/写
- (6) 读/写缓存，准备进行下次的传输

24 实时时钟 (RTC)

24.1 概述

实时时钟 (Real Time Counting, RTC) 单元提供给用户实时时间以及日历信息。RTC 单元通过时间寄存器提供时间信息 (秒、分、时、星期、日、月、年)。数据信息由 BCD 码格式进行表示。修改计数器的值可以重新设置系统当前的时间和日期。

RTC 模块可以根据年、月份 (闰年、大小月), 自动补偿天数; 还可以进行夏令时、冬令时补偿。

RTC 的时钟源可以通过软件选择外部低频晶振 EXTLF、内部低频时钟 INTLF 和外部高频晶振的 128 分频。RTC 模块自带高精度的数字时钟校准功能。

RTC 提供两个可编程的闹钟功能及中断, 用户可预先在时间闹钟寄存器中设置闹钟日期进行闹铃设置。

RTC 模块位于备份域内, 因此所有对 RTC 模块的操作都将受到备份域保护, 操作 RTC 寄存器之前需要允许备份域可写; 使能 RTC 模块之后, 只要电源电压保持在工作范围内, RTC 将可正常工作在任何运行模式和休眠模式。

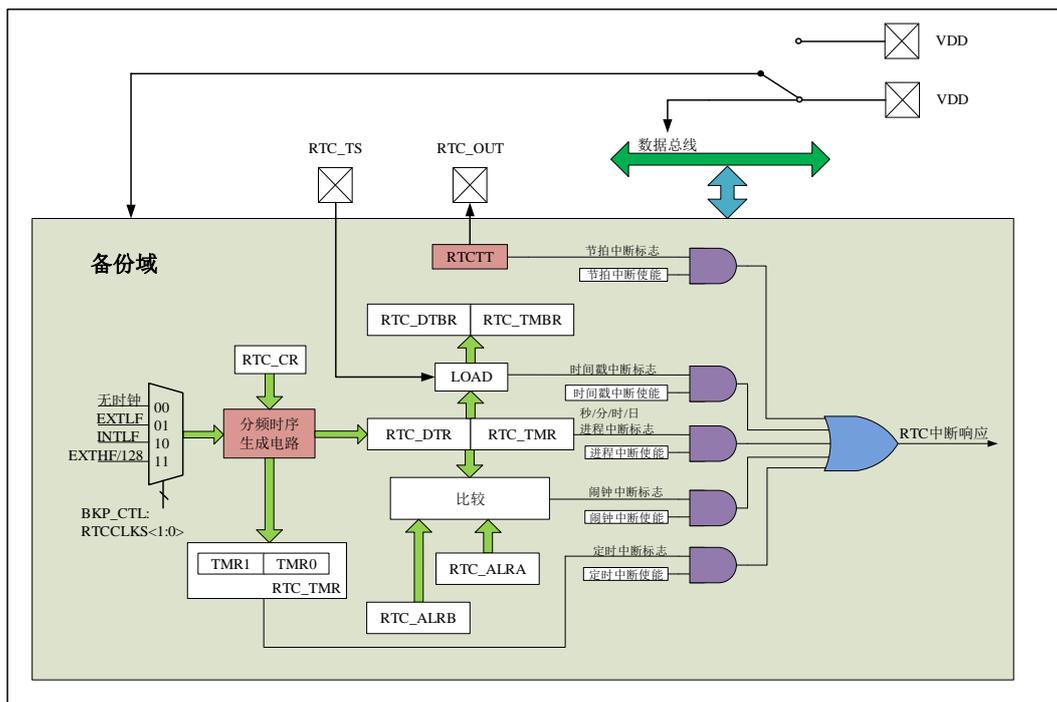


图 24-1 RTC 结构框图

RTC 相关引脚

- RTC_TS: RTC 时间戳功能外部输入引脚;
- RTC_OUT: RTC 的周期时间节拍 (TT) 输出引脚。

RTC 特性

- 提供实时时钟的日历功能, 自动解决闰年问题, 计时范围 100 年;
- 闰年、平年显示;
- 12/24 小时模式选择功能;
- 高精度的数字时钟校正功能;
- 外接备份电源 VBAT;

- 支持闹钟中断功能；
- 支持时间节拍中断功能，提供 8 个周期选项；
- 支持时分秒进程中中断功能；
- 内置 2 个 16 位定时器 TMR0 和 TMR1；
- 时间戳 timestamp function
- ADD1H、SUB1H 用于时间调整（冬令时/夏令时）
- 双闹钟
- 可工作在所有工作模式

24.2 RTC 相关寄存器

表 24-1 RTC 相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	RTC_CR	R/W	实时时钟控制寄存器	0x0000 0000
0x004	RTC_ALRA	R/W	实时时钟闹钟 A 控制寄存器	0x0000 0000
0x008	RTC_TMR	R/W	实时时钟时间寄存器	0x0000 0000
0x00C	RTC_DTR	R/W	实时时钟日期寄存器	0x0000 0000
0x010	RTC_ALRB	R/W	实时时钟闹钟 B 控制寄存器	0x0000 0000
0x014	RTC_TMER	R/W	实时时钟定时器寄存器	0x0000 0000
0x018	RTC_TCR	R/W	实时时钟定时器控制寄存器	0x0000 0000
0x01C	RTC_IER	R/W	实时时钟中断使能寄存器	0x0000 0000
0x020	RTC_IFR	R/W	实时时钟中断标志寄存器	0x0000 0000
0x024	RTC_TMBR	R/W	实时时钟时间备份寄存器	0x0000 0000
0x028	RTC_DTBR	R/W	实时时钟日期备份寄存器	0x0000 0000

基地址：0x4000 0F00

24.2.1 RTC_CR 实时时钟控制寄存器

表 24-2 RTC_CR 实时时钟控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R	R/W	
位名	RTCCAL<7:0>									TSEEDGE	TSEN	ADD1H	SUB1H	RTCTT<2:0>					RTCTSCH<5:0>						RESET	LIF	HT	CNF	RTOFF	LD	RTCEN	

RTCCAL<7:0>: RTC 时钟校正位

TSEEDGE: 时间戳触发沿

0 = 上升沿触发

1 = 下降沿触发

TSEN: 时间戳使能

0 = 未使能时间戳功能

1 = 使能时间戳功能

ADD1H: 增加 1 小时

0 = RTC 正常计时

1 = RTC 时间增加 1 小时(变为夏令时)

注：24 小时制时，不支持在 23 时将 ADD1H 置 1

12 小时制时，不支持在 11 时将 ADD1H 置 1

不支持在已经是夏令时的情况下将该位置 1 使用。

SUB1H: 减少 1 小时

0 = RTC 正常计时

1 = RTC 时间减少 1 小时 (变为冬令时)

注：24 小时制或 12 小时制时，均不支持在 0 时将 SUB1H 置 1

不支持在已经是冬令时的情况下将该位置 1 使用。

RTCTT<2:0>: 时间节拍设置位

- 000 = 时间节拍为 1 秒
- 001 = 时间节拍为 1/2 秒
- 010 = 时间节拍为 1/4 秒
- 011 = 时间节拍为 1/8 秒
- 100 = 时间节拍为 1/16 秒
- 101 = 时间节拍为 1/32 秒
- 110 = 时间节拍为 1/64 秒
- 111 = 时间节拍为 1/128 秒

RTCTSCH<5:0>: 时间戳通道使能位, 见“表 24-18 时间戳通道”

- 0 = 禁止 RTCTSCHx 通道的时间戳功能
- 1 = 使能 RTCTSCHx 通道的时间戳功能

RESET: 初始化位

通过在此位中写入"1", 可进行 IC 内部的初始化。此位为写入专用位, 读出时一直为"0"。另外, 在 IC 的电源电压接通时, 请务必在初始化位中写入 "1", 将电路初始化。

LIF: 闰年指示标志

- 0 = 当前年份为平年
- 1 = 当前年份为闰年

HT: 选择小时显示的类型

- 0 = 小时显示为 24 小时制
- 1 = 小时显示为 12 小时制

CNF: 配置标志

此位必须由软件置'1'以进入配置模式, 从而允许向实时时钟寄存器、闹钟中断时间寄存器写入数据。只有当此位在被置 1 并重新由软件清零后, 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

RTOFF: RTC 操作关闭

RTC 模块利用这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。此位为只读位。

- 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

RTCLD: RTC 活动状态 (只读)

- 0 = RTC 处在复位状态
- 1 = RTC 处在正常有效状态

RTCEN: RTC 使能位

- 0 = 禁止 RTC
- 1 = 使能 RTC

24. 2. 2 RTC_ALRA 实时时钟闹钟 A 控制寄存器
表 24-3 RTC_ALRA 实时时钟闹钟 A 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	ALRENA				AWENA	ALRWA<2:0>			AHENA	AMPMA	ALRHA<5:0>				AMENA	ALRMA<6:0>				ASENA	ALRSA<6:0>												

ALRENA: 闹钟 A 使能位

0 = 禁止闹钟 A 功能

1 = 使能闹钟 A 功能

AWENA: 闹钟 A 周使能

0 = 禁止闹钟 A 周位

1 = 使能闹钟 A 周位

ALRWA<2:0>: 设置闹钟中断星期时间（闹钟星期数值为 1~7），采用 BCD 编码

AHENA: 闹钟 A 时钟使能

0 = 禁止闹钟 A 时钟位

1 = 使能闹钟 A 时钟位

AMPMA: 12 小时制时闹钟 A 的上午下午选择（24 小时制时无效）

0 = 选择上午时间

1 = 选择下午时间

ALRHA<5:0>: 设置闹钟中断小时时间（闹钟时的数值为 00~23），采用 BCD 编码

AMENA: 闹钟 A 分钟使能

0 = 禁止闹钟 A 分钟位

1 = 使能闹钟 A 分钟位

ALRMA<6:0>: 设置闹钟中断分时间(闹钟分的数值为 00~59)，采用 BCD 编码

ASENA: 闹钟 A 秒钟使能

0 = 禁止闹钟 A 秒钟位

1 = 使能闹钟 A 秒钟位

ALRSA<6:0>: 设置闹钟中断秒时间(闹钟秒的数值为 00~59)，采用 BCD 编码

24. 2. 3 RTC_ALRB 实时时钟闹钟 B 控制寄存器
表 24-4 RTC_ALRB 实时时钟闹钟 B 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	ALRENB				AWENB	ALRWB<2:0>			AHENB	AMPMB	ALRHB<5:0>				AMENB	ALRMB<6:0>				ASENB	ALRSB<6:0>												

ALRENB: 闹钟 B 使能位

0 = 禁止闹钟 B 功能

1 = 使能闹钟 B 功能

AWENB: 闹钟 B 周使能

- 0 = 禁止闹钟 B 周位
- 1 = 使能闹钟 B 周位

ALRWB<2:0>: 设置闹钟中断星期时间 (闹钟星期数值为 1~7), 采用 BCD 编码

AHENB: 闹钟 B 时钟使能

- 0 = 禁止闹钟 B 时钟位
- 1 = 使能闹钟 B 时钟位

AMPMB: 12 小时制时闹钟 B 的上午下午选择 (24 小时制时无效)

- 0 = 选择上午时间
- 1 = 选择下午时间

ALRHB<5:0>: 设置闹钟中断小时时间 (闹钟时的数值为 00~23), 采用 BCD 编码

AMENB: 闹钟 B 分钟使能

- 0 = 禁止闹钟 B 分钟位
- 1 = 使能闹钟 B 分钟位

ALRMB<6:0>: 设置闹钟中断分时间(闹钟分的数值为 00~59), 采用 BCD 编码

ASENB: 闹钟 B 秒钟使能

- 0 = 禁止闹钟 B 秒钟位
- 1 = 使能闹钟 B 秒钟位

ALRSB<6:0>: 设置闹钟中断秒时间(闹钟秒的数值为 00~59), 采用 BCD 编码

24. 2. 4 RTC_TMR 实时时钟时间寄存器

表 24-5 RTC_TMR 实时时钟时间寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	u	u	u	0	u	u	u	u	u	u	u	0	u	u	u	u	u	u	u	0	u	u	u	u	u	u	u	0
R/W						R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W							
位名					WEK <2:0>			AMPM	HOUR<5:0>					MIN<6:0>					SEC<6:0>														

WEK<2:0>: RTC 时钟星期

AM/PM: 当 HT 置 1 选择 24 小时制时, 该位无效, 0,1 均可。

当 HT 置 0 选择 12 小时制时:

- 0 = 时间为 AM
- 1 = 时间为 PM

HOUR<5:0>: RTC 时钟时钟

MIN<6:0>: RTC 时钟分钟

SEC<6:0>: RTC 时钟秒钟

注 1: 可通过读此寄存器来获得 RTC 计数器当前值的时时间。小时数值为 00~23 或 00~12, 采用 BCD 编码。

注 2: 要对此寄存器进行写操作, 必须先进入配置模式(CNF = 1)。

24. 2. 5 RTC_DTR 实时时钟日期寄存器

表 24-6 RTC_DTR 实时时钟日期寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	u	u	u	u	u	u	u	u	0	0	0	u	u	u	u	u	0	0	u	u	u	u	u	0	
R/W									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	
位名									YEAR<7:0>												MTH<4:0>								DAY<5:0>				

YEAR<7:0>: RTC 时钟年份

MTH<4:0>: RTC 时钟月份

DAY<5:0>: RTC 时钟日期

注: 要对此寄存器进行写操作, 必须先进入配置模式(CNF = 1)。

24. 2. 6 RTC_TMER 实时时钟定时器寄存器

表 24-7 RTC_TMER 实时时钟定时器寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	TMR1<15:0>																TMR0<15:0>															

TMR1<15:0>: 定时器 1 计数值

TMR0<15:0>: 定时器 0 计数值

24. 2. 7 RTC_TCR 实时时钟定时器控制寄存器

表 24-8 RTC_TCR 实时时钟定时器控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W																							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																							TMRIEN		TMROEN		TMR1C <3:0>			TMROC <3:0>		

TMRIEN: 定时器 1 使能位

0 = 禁止定时器 1

1 = 使能定时器 1

TMROEN: 定时器 0 使能位

0 = 禁止定时器 0

1 = 使能定时器 0

TMR1C<3:0>: 定时器 1 时钟源选择位

0000 = RTC 时钟源/128 约 1/256s

0001 = RTC 时钟源/512 约 1/64s

0010 = RTC 时钟源/1024 约 1/32s

0 = 禁止闹钟中断

1 = 使能闹钟中断

DAYIE: 日进程中中断使能位

0 = 禁止日进程中中断

1 = 使能日进程中中断

HOURIE: 小时进程中中断使能位

0 = 禁止小时进程中中断

1 = 使能小时进程中中断

MINIE: 分进程中中断使能位

0 = 禁止分进程中中断

1 = 使能分进程中中断

SECIE: 秒进程中中断使能位

0 = 禁止秒进程中中断

1 = 使能秒进程中中断

24.2.9 RTC_IFR 实时时钟中断标志寄存器

表 24-10 RTC_IFR 实时时钟中断标志寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W					R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W						R	R	R	R		R	R	R	R	R	R	R	R
位名					TSIC	TSOVFIC	RTCT1IC	RTCT0IC		TTIC	ALRBIC	ALRAIC	DAVIC	HOURLC	MINIC	SECCIC						TSIF	TSOVFIF	RTCT1IF	RTCT0IF		TTIF	ALRBIF	ALRAIF	DAVIF	HOURLF	MINIF	SECCIF	

TSIC: 时间戳中断清零位

0 = 不清零时间戳中断

1 = 清零时间戳中断

TSOVFIC: 时间戳溢出中断清零位

0 = 不清零时间戳溢出中断

1 = 清零时间戳溢出中断

RTCT1IC: RTC 定时器 1 中断清零位

0 = 不清零 RTC 定时器 1 中断

1 = 清零 RTC 定时器 1 中断

RTCT0IC: RTC 定时器 0 中断清零位

0 = 不清零 RTC 定时器 0 中断

1 = 清零 RTC 定时器 0 中断

TTIC: 时间节拍中断清零位

0 = 不清零时间节拍中断

1 = 清零时间节拍中断

ALRBIC: 闹钟 B 中断清零位

0 = 不清零闹钟中断

1 = 清零闹钟中断

ALRAIC: 闹钟 A 中断清零位

0 = 不清零闹钟中断

- 1 = 清零闹钟中断
- DAYIC: 日进程中中断清零位
- 0 = 不清零日进程中中断
- 1 = 清零日进程中中断
- HOURIC: 小时进程中中断清零位
- 0 = 不清零小时进程中中断
- 1 = 清零小时进程中中断
- MINIC: 分进程中中断清零位
- 0 = 不清零分进程中中断
- 1 = 清零分进程中中断
- SECIC: 秒进程中中断清零位
- 0 = 不清零秒进程中中断
- 1 = 清零秒进程中中断
- TSIF: 时间戳中断标志位
- 0 = 未产生时间戳中断
- 1 = 产生了时间戳中断
- TSOVFIF: 时间戳溢出中断标志位
- 0 = 未产生时间戳溢出中断
- 1 = 产生了时间戳溢出中断
- RTCT1IF: RTC 定时器 1 中断标志位
- 0 = 未产生 RTC 定时器 1 中断
- 1 = 产生了 RTC 定时器 1 中断
- RTCT0IF: RTC 定时器 0 中断标志位
- 0 = 未产生 RTC 定时器 0 中断
- 1 = 产生了 RTC 定时器 0 中断
- TTIF: 时间节拍中断标志位
- 0 = 未产生时间节拍中断
- 1 = 产生了时间节拍中断
- ALRBIF: 闹钟 B 中断标志位
- 0 = 未产生闹钟中断
- 1 = 产生了闹钟中断
- ALRAIF: 闹钟 A 中断标志位
- 0 = 未产生闹钟中断
- 1 = 产生了闹钟中断
- DAYIF: 日进程中中断标志位
- 0 = 未产生日进程中中断
- 1 = 产生了日进程中中断
- HOURIF: 小时进程中中断标志位
- 0 = 未产生小时进程中中断
- 1 = 产生了小时进程中中断
- MINIF: 分进程中中断标志位
- 0 = 未产生分进程中中断
- 1 = 产生了分进程中中断
- SECIF: 秒进程中中断标志位

0 = 未产生秒进程中断

1 = 产生了秒进程中断

24. 2. 10 RTC_TMBR 实时时钟时间备份寄存器

表 24-11 RTC_TMBR 实时时钟时间备份寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名						WEKB<2:0>		AMPMB	HOURB<5:0>					MINB<6:0>					SECB<6:0>													

WEKB<2:0>: RTC 时钟星期备份

AM/PMB: AM/PM 备份

HOURB<5:0>: RTC 时钟时钟备份

MINB<6:0>: RTC 时钟分钟备份

SECB<6:0>: RTC 时钟秒钟备份

24. 2. 11 RTC_DTBR 实时时钟日期备份寄存器

表 24-12 RTC_DTBR 实时时钟日期备份寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名									YEARB<7:0>							MTHB<4:0>				DAYB<5:0>												

YEARB<7:0>: RTC 时钟年份备份

MTHB<4:0>: RTC 时钟月份备份

DAYB<5:0>: RTC 时钟日期备份

24.3 RTC 模块的复位和寄存器读写

RTC 模块位于备份域内，完成上电复位后，在 VBAT 有电条件下，RTC 模块不会因为 VDD 掉电而停止工作，VBAT 将继续维持 RTC 供电以维持 RTC 继续工作。

PM_CTL0 寄存器的 BKPREGCLR 位可用于 RTC 模块的软件复位，当 BKPREGCLR=0 时，整个 RTC 模块将被复位。注意，BKPREGCLR 位清零时将同时复位备份域数据寄存器组。

对 RTC 模块进行操作时，需要先对如下寄存器位进行配置：

- (1) OSCCTL0 寄存器的 PMWREN 位置 1，允许整个备份域的读写操作；
- (2) PM_CTL0 寄存器的 BKPREGCLR 位置 1，RTC 模块退出复位；
- (3) PM_CTL0 寄存器的 BKPWR 位置 1，允许备份域数据寄存器组和 RTC 模块内的寄存器读写操作。

24.4 功能描述

RTC 时钟可在任何模式下工作，并且可触发中断将 CPU 从休眠模式唤醒。实时时钟可使用的专用振荡器频率为 32.768kHz，通过 BKP_CTL 寄存器的 RTCCLKS<1:0>位可以选择实时时钟的时钟源。可选择无时钟输入、外部低频振荡器 (EXTLF)、内部低频振荡器 (INTLF) 和外部高频振荡器 128 分频后的时钟作为 RTC 时钟，默认为无时钟输入。

RTC 可显示在 2000/01/01 日 00:00:00 至 2099/12/31 日 23:59:59 之间的任意时间。时间显示格式可在 12 小时制和 24 小时制之间自由切换。时间的最小间隔为 1 秒。

24.4.1 时钟校正

时钟校正功能是为了实现高精度的时钟功能，校正因振荡频率的偏差而导致的时钟的提前或滞后的功能。进行校正时，并不调节石英振荡器的频率，而是利用分频电路的一部分对时钟脉冲进行调整。校正工作每 20 秒 (或 60 秒) 进行一次。最小分解能约为在大约 3 ppm (或约 1 ppm) 的条件下，可以在 -195.3 ppm ~ +192.2 ppm (或是 -65.1 ppm ~ +64.1 ppm) 的范围内进行校正。此设定可在时钟校正寄存器上进行。另外，不使用时钟校正功能时，请务必将其设定为 "00 h"。

表 24-13 时间校正参考

项目	RTCCAL7=0	RTCCAL7=1
校正工作	每20秒	每60秒
最小分解能	3.052ppm	1.017ppm
校正范围	-195.3ppm~+192.2ppm	-65.1ppm~+64.1ppm

24.4.1.1 计算方法

- (1) 当前振荡频率 > 目标频率时 (时间提前时)

$$\text{校正值} = 128 - \text{整数} \times \left(\frac{\text{当前振荡频率实测值} - \text{目标振荡频率}}{\text{当前振荡频率实测值} \times \text{最小分解能}} \right) \quad \text{公式 24-1}^1$$

- (2) 当前振荡频率 < 目标频率时 (时间滞后时)

¹ 计算时取括号内的整数值来计算，此计算值的“0~64”范围为可校正范围。

$$\text{校正值} = \text{整数} \times \left(\frac{(\text{目标振荡频率}) - (\text{当前振荡频率实测值})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right) + 1 \quad \text{公式 24-2}^1$$

24.4.1.2 校准原理

根据实际测出的频率，利用分频电路的一部分对时钟脉冲进行增加或者减少。

(1) 当前振荡频率 > 目标频率时 (时间提前时)

$$\text{校正值} = 128 - \text{整数} \times \left(\frac{(\text{当前振荡频率实测值}) - (\text{目标振荡频率})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right) \quad \text{公式 24-3}$$

例：校准寄存器值为 0111 1101 (t=1/32768)

表 24-14 校准例图 (校准周期为 20s, 延迟时间, 校准大小为 3)

	校准值为 0111 1101	无校准
分频计数器计数	01 00 0000 0	01 00 0000 0
	00 11 1101 1 +6t	01 00 0000 1
	00 11 1110 0 +6t	01 00 0001 0
	00 11 1110 1 +6t	01 00 0001 1

在 20s 校准周期中增加了 6 个 t (1/32768), 延迟时间。

20 秒校准 1 次增加的 t = 校准寄存器 <B6:B0> 补码值 * 2t

(2) 当前振荡频率 < 目标频率时 (时间滞后时)

$$\text{校正值} = \text{整数} \times \left(\frac{(\text{目标振荡频率}) - (\text{当前振荡频率实测值})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right) + 1 \quad \text{公式 24-4}$$

例：校准寄存器值为 0000 0011 (t=1/32768)

表 24-15 校准例图 (校准周期为 20s, 加速时间, 校准大小为 3)

	校准值为 0000 0011	无校准
分频计数器计数	01 00 0000 0	01 00 0000 0
	01 00 0011 1 -6t	01 00 0000 1
	01 00 0100 0 -6t	01 00 0001 0
	01 00 0100 1 -6t	01 00 0001 1

在 20s 校准周期中减去了 6 个 t (1/32768), 加速时间。

20 秒校准 1 次减去的 t = 校准寄存器值 <B6:B0> * 2t

¹ 计算时取括号内的整数值来计算，此计算值的“0~62”范围为可校正范围。

24. 4. 1. 3 校准寄存器设定值表
表 24-16 校准寄存器设定值表 1（最小分解能=3.052ppm）

RTCCAL7=0,每 20 秒校准一次最小分解能=3.052ppm									
RTCCAL7	RTCCAL6	RTCCAL5	RTCCAL4	RTCCAL3	RTCCAL2	RTCCAL1	RTCCAL0	校准值 (ppm)	精度 (sec/day)
0	0	1	1	1	1	1	1	192.3	16.61
0	0	1	1	1	1	1	0	189.2	16.35
0	0	1	1	1	1	0	1	186.2	16.09
...							
0	0	0	0	0	0	1	0	6.1	0.53
0	0	0	0	0	0	0	1	3.1	0.26
0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	-3.1	-0.26
0	1	1	1	1	1	1	0	-6.1	-0.53
...							
0	1	0	0	0	0	1	1	-186.2	-16.09
0	1	0	0	0	0	1	0	-189.2	-16.35
0	1	0	0	0	0	0	1	-192.3	-16.61
0	1	0	0	0	0	0	0	-195.3	-16.88

表 24-17 校准寄存器设定值表 2（最小分解能=1.017ppm）

RTCCAL7=1,每60秒校准一次最小分解能=1.017ppm									
RTCCAL7	RTCCAL6	RTCCAL5	RTCCAL4	RTCCAL3	RTCCAL2	RTCCAL1	RTCCAL0	校准值 (ppm)	精度 (sec/day)
1	0	1	1	1	1	1	1	64.1	5.54
1	0	1	1	1	1	1	0	63.1	5.45
1	0	1	1	1	1	0	1	62.0	5.36
...							
1	0	0	0	0	0	1	0	2.0	0.18
1	0	0	0	0	0	0	1	1.0	0.09
1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	-1.0	-0.09
1	1	1	1	1	1	1	0	-2.0	-0.18
...							
1	1	0	0	0	0	1	1	-62.0	-5.36
1	1	0	0	0	0	1	0	-63.1	-5.45
1	1	0	0	0	0	0	1	-64.1	-5.54
1	1	0	0	0	0	0	0	-65.1	-5.62

24. 4. 2 闹钟

使能闹钟功能后，当实时时钟寄存器等于闹钟中断寄存器中的设定值，闹钟标志位 ALRxIF(x=A,B)置 1，并发送 RTC_WKUP (EINT19) 闹钟中断请求。闹钟中断请求可以触发唤醒功能，将芯片从深度休眠模式唤醒到正常运行模式。ALRxIF(x=A,B)置 1 同时也可以

发送 RTC (EINT17) 中断。

通过 ALRAIC/ALRBIC 闹铃清零位可以清除 ALRxIF(x=A,B)闹铃中断标志位。通过清零 EINTIF17/19 位清零 RTC/RTC_WKUP 中断。

24.4.3 时间戳

时间戳可用于记录外部引脚的变化时刻。

通过 RTC_CR 寄存器的 RTCTSCHEN<5:0>位可以使能对应的 RTC 通道 (RTCTSCH0-5)。注意, 唤醒引脚 WKUPx 也可以用作时间戳引脚。

表 24-18 时间戳通道

RTCTSCHx	引脚
RTCTSCH0	RTC_TS 引脚
RTCTSCH1	WKUP1 引脚
RTCTSCH2	WKUP2 引脚
RTCTSCH3	WKUP3 引脚
RTCTSCH4	WKUP4 引脚
RTCTSCH5	WKUP5 引脚

时间戳有上升沿触发和下降沿触发两种。若使能上升沿触发, 则时间戳通道 RTCTSCHx 在由 0 变 1 时, 寄存器 RTC_TMR 和 RTC_DTR 的值分别被存入备份寄存器 RTC_TMBR 和 RTC_DTBR; 若使能下降沿触发, 则时间戳通道 RTCTSCHx 在由 1 变 0 时, 寄存器 RTC_TMR 和 RTC_DTR 的值分别被存入备份寄存器 RTC_TMBR 和 RTC_DTBR。当触发时间发生时, TSIF 位置 1, 并发出 RTC_ALRAM (EINT17) 的 RTC 中断。通过 TSIC 位可以清零 TSIF 标志位, 通过清零 EINTIF17 位清零 RTC_ALRAM 中断

当时间戳中断 TSIF 为 1 时, 继续接收到时间戳触发, 则会引起时间戳中断溢出, 将 TSOVFIF 位置 1, 并发送发出 RTC_ALRAM (EINT17) 的 RTC 中断。通过 TSOVFIC 位可以清零 TSIF 标志位, 通过清零 EINTIF17 位清零 RTC_ALRAM 中断

24.4.4 周期时间节拍中断

周期时间节拍中断有 8 个选项: 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 以及 1 秒, 通过设置 RTC_CR 寄存器中 RTCTT 位来进行选择。当 RTC 满足节拍中断时, 会将 TTIF 位置 1 时, 当时间节拍中断使能位 TTIE 置 1 时, 可产生周期时间节拍中断。

周期时间节拍中断是 RTC (EINT17) 中断的一种, 可通过中断控制寄存器和中断优先级寄存器控制配置 RTC 中断的优先级。通过 TTIC 可以清零 TTIF 中断标志位, 通过清零 EINTIF17 可以清零 RTC 中断。

表 24-19 节拍中断

RTCTT<2:0>	时间节拍(秒)
000	1
001	1/2
010	1/4
011	1/8
100	1/16
101	1/32
110	1/64
111	1/128

24.4.5 进程中中断

通过 RTC_IER 寄存器的 SECIE/MINIE/HURIE/DAYIE 这 4 个使能端分别控制秒、分、时或天的进程中中断。当 RTC 达到设置的进程时间时则将对应的 SECIF/MINIF/HURIF/DAYIF 位置 1，若进程中中断使能置位，则可以产生中断信号。

进程中中断是 RTC 中断 (EINT17) 的一种，可通过中断控制寄存器和中断优先级寄存器控制配置 RTC 中断的优先级通过 SECIC/ MINIC/ HURIC/ DAYIC 可以清零对应的进程中中断标志位，通过清零 EINTIF17 可以清零 RTC 中断。

24.4.6 实时时钟定时器

实时时钟模块内置 2 个可级联的 16 位定时器 TMR0 和 TMR1。RTC_TCR 寄存器的 TMR1EN 位和 TMR0EN 位为实时时钟定时器的使能位。

24.4.6.1 实时时钟定时器的使能

RTC_TCR 寄存器的 TMR1EN 位为定时器 1 的使能位，同时也是定时器 1 和定时器 0 的时钟使能位；RTC_TCR 寄存器的 TMR0EN 位为定时器 0 的使能位，使用定时器 0 需要同时置位 TMR1EN 位和 TMR0EN 位。

- 使能定时器 0：TMR1EN 和 TMR0EN 位需要同时置 1；
- 使能定时器 1：TMR1EN 位置 1。

24.4.6.2 实时时钟定时器的中断

TMR0 和 TMR1 为向下计数，当发生下溢时(计数值=00H)，RTC_IFR 寄存器的 RTCT1IF (或 RTCT0IF) 位将置 1，如果 RTC_IER 寄存器的 RTCT1IE (或 RTCT0IE) 位置 1，程序将进入中断行。RTC_IFR 寄存器的 RTCT1IC(或 RTCT0IC)位用于清零 RTCT1IF(或 RTCT0IF) 位。

24.4.6.3 实时时钟定时器的计数和重载

TMR0 和 TMR1 为向下计数，使能定时器后，定时器将载入 RTC_TMER 寄存器的值，开始递减计数；当定时器计数到 0 后，将会重新载入 RTC_TMER 寄存器的值。

如果在计数过程中对 RTC_TMER 寄存器进行修改，不会影响当前计数周期，定时器会在下一个下溢事件（计数到 0）发生后，载入 RTC_TMER 寄存器的新值并开始一个新的计数周期。

TMR0 和 TMR1 定时器的实时计数值不可读。

24.4.6.4 实时时钟定时器的级联

将 RTC_TCR 寄存器的 TMR1C<3:0>位配置为 1000H 时，将使能 TMR0 和 TMR1 的级联，TMR0 作为低 16 位，TMR1 作为高 16 位，组成一个 32 位定时器；通过 TMR0C<3:0>位选择计数时钟；TMR0EN 位和 TMR1EN 位需要同时置 1；两个中断标志位 RTCT0IF 和 RTCT1IF 均有效，用户可任选其一。

24.4.6.5 实时时钟定时器的配置步骤

定时器的配置步骤如下：

- (1) 通过配置 RTC_TCR 寄存器的 TMR1C<3:0>和 TMR0C<3:0>位，选择 TMR1 和 TMR0 的时钟源及是否级联；
- (2) 向 RTC_TMER 寄存器写入定时器初值；
- (3) 配置 RTC_TCR 寄存器的 TMR1EN 和 TMR0EN 位，使能定时器；
- (4) 配置 RTC_IFR 寄存器的 RTCT1IC 和 RTCT0IC 位用于清零 RTCT1IF 和 RTCT0IF 位；
- (5) 配置 RTC_IER 寄存器的 RTCT1IE 和 RTCT0IE 位，使能定时器中断。

注：使用实时时钟定时器前必须使能 RTC 模块。

24.4.7 复位

RTC 位于备份域中，使能后，在 VBAT 维持供电条件下，即使 VDD 掉电，RTC 依然正常工作。在初始上电和 VDD 和 VBAT 两者都掉电的时候，才会发生掉电复位。

备份域控制寄存器中的 BKPRST 位置 1，也会复位 RTC 模块。

25 控制器局域网总线（CAN0/1/2）

25.1 概要

控制器局域网（Controller Area Network，简称为 CAN）是一种用于连接电子控制设备（Electronic Control Unit，简称为 ECU）的多主共享型串行总线标准。CAN 总线针对抗电磁干扰进行了专门设计，适用于具有较强电磁干扰的环境，不但可以使用与 RS-485 类似的差分平衡传输线，也可以使用更加可靠的双绞线。CAN 总线最初是针对汽车应用而研发的，不过时至今日已经广泛应用于各种嵌入式控制领域（例如工业方面和医疗方面）。CAN 总线在总线长度小于 40 米时最高可达 1Mbps 位速率。位速率越低则有效通讯距离越远（例如 125kbps 时通讯距离可达 500 米）。

CAN 有如下特性：

- 支持 CAN2.0B 协议
- 同时支持 11 位和 29 位识别码
- 位速率可达 1Mbps
- 可读/写访问的错误计数器
- 可编程的错误报警限制
- 最近一次错误代码寄存器
- 对每一个 CAN 总线错误的中断
- 具体控制位控制的仲裁丢失中断
- 在标准和扩展格式中都有验收滤波器含屏蔽和代码寄存器
- 当错误或仲裁丢失时可配置是否重发
- 支持接收和发送 DMA

25.2 CAN 模块结构框图

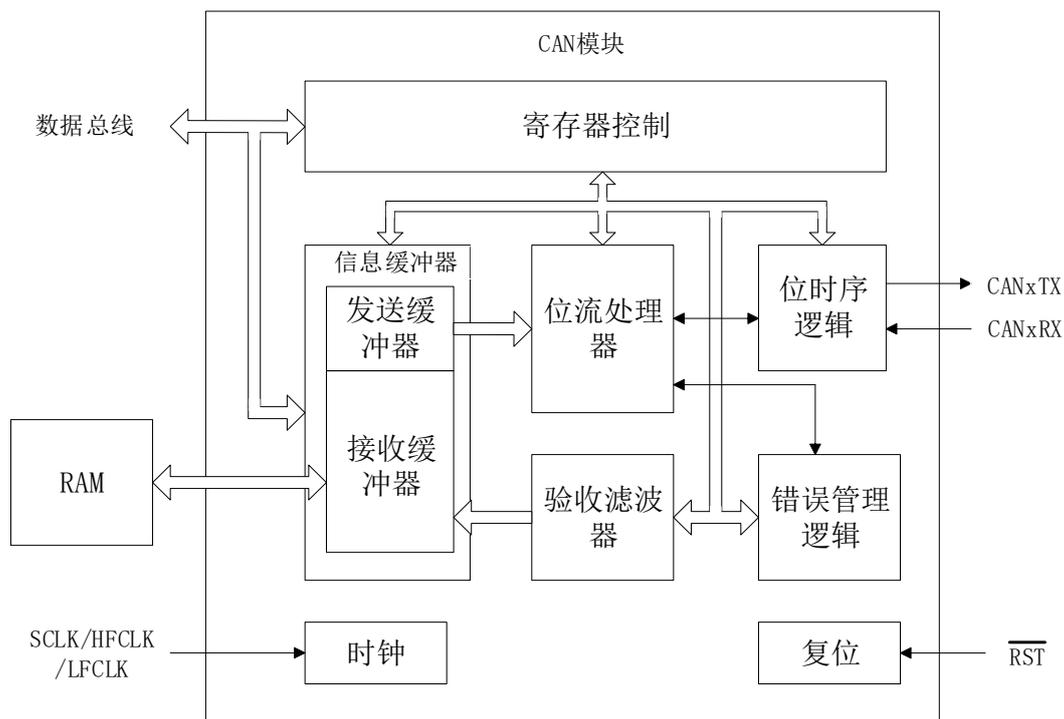


图 25-1 CAN 模块结构框图

- CANxTX 引脚：CAN 模块的通信发送引脚，通过 IO 口重映射功能映射到 IO 口。
- CANxRX 引脚：CAN 模块的通信接收引脚，通过 IO 口重映射功能映射到 IO 口。

25.3 CAN 模块相关寄存器

 表 25-1 CANx 模块相关寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x000	CANx_CTLR	R/W	CAN 控制寄存器	0x003C 0001
0x004	CANx_BRGR	R/W	CAN 波特率寄存器	0xuuuu uuuu
0x008	CANx_RCR	R	CAN 记录寄存器	0x0000 0000
0x00C	CANx_EROR	R/W	CAN 错误寄存器	0x0060 0000
0x010	CANx_ACRR	R/W	CAN 验收代码寄存器	0xuuuu uuuu
0x014	CANx_MSKR	R/W	CAN 验收屏蔽寄存器	0xuuuu uuuu
0x018	CANx_IER	R/W	CAN 中断使能寄存器	0x0000 0000
0x01C	CANx_IFR	R/W	CAN 中断标志寄存器	0x0000 0000
0x020	CANx_INFR	R/W	CAN 数据发送信息寄存器	0xuuuu uuuu
0x024	CANx_TX0R	R/W	CAN 数据发送缓冲寄存器 0	0xuuuu uuuu
0x028	CANx_TX1R	R/W	CAN 数据发送缓冲寄存器 1	0xuuuu uuuu
0x02C	CANx_TX2R	R/W	CAN 数据发送缓冲寄存器 2	0xuuuu uuuu
0x030	CANx_RXDATA0	R	CAN 数据接收缓冲寄存器 0	0xuuuu uuuu
0x034	CANx_RXDATA1	R	CAN 数据接收缓冲寄存器 1	0xuuuu uuuu
0x038	CANx_RXDATA2	R	CAN 数据接收缓冲寄存器 2	0xuuuu uuuu
0x03C	CANx_RXDATA3	R	CAN 数据接收缓冲寄存器 3	0xuuuu uuuu

表 25-2 寄存器不同模式下功能

偏移地址	工作模式		复位模式	
	读	写	读	写
0x000	CANx_CTLR	CANx_CTLR	CANx_CTLR	CANx_CTLR
0x004	CANx_BRGR	-	CANx_BRGR	CANx_BRGR
0x008	CANx_RCR	-	CANx_RCR	-
0x00C	CANx_EROR	-	CANx_EROR	CANx_EROR
0x010	CANx_ACRR	-	CANx_ACRR	CANx_ACRR
0x014	CANx_MSKR	-	CANx_MSKR	CANx_MSKR
0x018	CANx_IER	CANx_IER	CANx_IER	CANx_IER
0x01C	CANx_IFR	CANx_IFR	CANx_IFR	CANx_IFR
0x020	CANx_INFR	CANx_INFR	CANx_INFR	CANx_INFR
0x024	CANx_TX0R	CANx_TX0R	CANx_TX0R	CANx_TX0R
0x028	CANx_TX1R	CANx_TX1R	CANx_TX1R	CANx_TX1R
0x02C	CANx_TX2R	CANx_TX2R	CANx_TX2R	CANx_TX2R
0x030	CANx_RXDATA0	-	CANx_RXDATA0	-
0x034	CANx_RXDATA1	-	CANx_RXDATA1	-
0x038	CANx_RXDATA2	-	CANx_RXDATA2	-
0x03C	CANx_RXDATA3	-	CANx_RXDATA3	-

CAN0 基地址: 0x4000 1100

¹ 本章除特殊说明外, x=0,1,2,3,4,5

CAN1 基地址: 0x4000 1180

CAN2 基地址: 0x4000 1900

注: CAN0 支持低功耗 CAN 功能。

25. 3. 1 CANx_CTLR CAN 控制寄存器

表 25-3 CANx_CTLR CAN 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0	
复位值				R	R	R	R	R	R	R	R	R	R	R	R	R					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	CANRMC<6:0>								BOFF	CERROR	TXSTA	RXSTA	TCSTA	TXBSTA	DOSTA	RXBSTA						RELF	RELX	ATX	TXR	CANEN	CANCKS<1:0>		TRIEP	SLEEP	LBACK	SILENT	RSMOD

CANRMC<4:0>: RX 信息计数器

00000 = RAM 中无数据

00001 = RAM 中有 1 帧数据

.....

01111 = RAM 中有 15 帧数据

10000 = RAM 中有 16 帧数据

BOFF: 总线状态

0 = 总线开启

1 = 总线关闭

CERROR: 出错状态

0 = 无错

1 = 出错

TXSTA: 发送状态

0 = 空闲

1 = 正在发送信息

RXSTA: 接收状态

0 = 空闲

1 = 正在接收信息

TCSTA: 发送完毕状态

0 = 发送请求未处理完

1 = 最后一次发送已被成功处理

TXBSTA: 发送缓冲器状态

0 = CPU 不能访问发送缓冲器

1 = CPU 可以访问发送缓冲器

DOSTA: 数据满状态

0 = RAM 中未读数据未读

1 = RAM 中未读数据满

RXBSTA: 数据空状态

0 = RAM 中无未读信息

1 = RAM 中有未读信息

RELF: 清除 RAM 满标志 (置 1 后需要软件清零)

- 0 = 无动作
 - 1 = 清除 RAM 满标志
- RELRX: 释放接收缓冲器 (置 1 后需要软件清零)
- 0 = 无动作
 - 1 = 接收缓冲器中的信息被释放, 信息计数器减 1
- ATX: 中止重发送 (置 1 后需要软件清零)
- 0 = 无动作
 - 1 = 等待中的发送请求被取消
- TXR: 发送请求 (置 1 后需要软件清零)
- 0 = 无动作
 - 1 = 发送当前信息
- CANEN: CAN 使能选择
- 0 = CAN 禁止
 - 1 = CAN 使能
- CANCK<1:0>: CAN 工作时钟选择位
- 00 = 选择 SCLK 主时钟/2 作为 CAN 工作时钟
 - 01 = 选择 HFCLK 时钟/2 作为 CAN 工作时钟
 - 1x = 选择 LFCLK 时钟/2 作为 CAN 工作时钟 (非 CAN0)
 - 10 = 选择 LFCLK 时钟/2 作为 CAN 工作时钟 (CAN0)
 - 11 = 选择 INTLF 时钟/2 作为 CAN 工作时钟 (CAN0)¹
- SLEEP: 睡眠模式选择 (复位模式下不可写)
- 0 = 禁止睡眠模式
 - 1 = 使能睡眠模式
- LBACK: 回环模式选择
- 0 = 禁止回环模式
 - 1 = 使能回环模式
- SILENT: 安静模式选择
- 0 = 禁止安静模式
 - 1 = 使能安静模式 (进入安静模式后, 错误消极中断如果使能将置 1)
- RSMOD: 复位模式选择
- 0 = 工作模式
 - 1 = 复位模式

RX 信息计数寄存器反映了 RAM 中未读信息的数目。其值在每次成功接收数据后加 1, 每次释放接收缓冲器时减 1 (置 1RELRX 位)。

如果信息计数器数值为 0, 即 RAM 中没有未读的信息; 如果信息计数器数值为其他非 0 值, 即 RAM 中有未读的信息, 就置位接收中断; 如果信息计数器数值为 1111 后还有数据存入 RAM 中, 即 RAM 中未读的信息数量已满, 就会置位数据满中断。

¹ 选用内部低频振荡器 INTLF 作为工作时钟时, 需要使能 PM_CTL2 寄存器的 CAN0CLKLPEN 位

25. 3. 3 CANx_RCR CAN 记录寄存器

表 25-5 CANx_RCR CAN 记录寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值	0	0	0	0	0	0	0	0	0	0	R	R	R	R	R	R	R	R	R	R	R	R	R	R	0	0	0	R	R	R	R	R	R
R/W																																	
位名											RAMMB<5:0>					CANERRC<1:0>		CANDIR	CANSEG<4:0>					CANALC<4:0>									

RAMMB<3:0>: 邮箱接收指针

0000 = 下一个接收到的帧会被放在区域 0

0001 = 下一个接收到的帧会被放在区域 1

...

1111 = 下一个接收到的帧会被放在区域 15

注: 每成功接收一次, 该位域加 1。

CANERRC<1:0>: 错误代码

00 = 位错误

01 = 格式错误

10 = 填充错误

11 = 其他错误

CANDIR: 方向

0 = 发送时发生的错误

1 = 接收时发生的错误

CANSEG<4:0>: 错误发生的段

00011 = 帧开始

00010 = ID.28-ID.21

00110 = ID.20-ID.18

00100 = SRR 位

00101 = IDE 位

00111 = ID.17-ID.13

01111 = ID.12-ID.5

01110 = ID.4-ID.0

01100 = RTR 位

01101 = 保留位 1

01001 = 保留位 0

01011 = 数据长度代码

01010 = 数据区

01000 = CRC 序列

11000 = CRC 界定符

11001 = 应答位

11011 = 应答界定符

11010 = 帧结尾

10010 = 中止 (帧间域)

10001 = 主动错误标志

- 10110 = 消极错误标志
- 10011 = 支配（控制）位误差
- 10111 = 错误界定符/过载界定符
- 11100 = 溢出标志（过载标志）

CANALC<4:0>: 仲裁丢失位

- 00000 = 仲裁丢失在识别码 ID28
- 00001 = 仲裁丢失在识别码 ID27
- 00010 = 仲裁丢失在识别码 ID26
- 00011 = 仲裁丢失在识别码 ID25
- 00100 = 仲裁丢失在识别码 ID24
- 00101 = 仲裁丢失在识别码 ID23
- 00110 = 仲裁丢失在识别码 ID22
- 00111 = 仲裁丢失在识别码 ID21
- 01000 = 仲裁丢失在识别码 ID20
- 01001 = 仲裁丢失在识别码 ID19
- 01010 = 仲裁丢失在识别码 ID18
- 01011 = 仲裁丢失在 SRR 位（标准帧信息 RTR 位）
- 01100 = 仲裁丢失在 IDE 位
- 01101 = 仲裁丢失在识别码 ID17（用于扩展帧信息）
- 01110 = 仲裁丢失在识别码 ID16（用于扩展帧信息）
- 01111 = 仲裁丢失在识别码 ID15（用于扩展帧信息）
- 10000 = 仲裁丢失在识别码 ID14（用于扩展帧信息）
- 10001 = 仲裁丢失在识别码 ID13（用于扩展帧信息）
- 10010 = 仲裁丢失在识别码 ID12（用于扩展帧信息）
- 10011 = 仲裁丢失在识别码 ID11（用于扩展帧信息）
- 10100 = 仲裁丢失在识别码 ID10（用于扩展帧信息）
- 10101 = 仲裁丢失在识别码 ID9（用于扩展帧信息）
- 10110 = 仲裁丢失在识别码 ID8（用于扩展帧信息）
- 10111 = 仲裁丢失在识别码 ID7（用于扩展帧信息）
- 11000 = 仲裁丢失在识别码 ID6（用于扩展帧信息）
- 11001 = 仲裁丢失在识别码 ID5（用于扩展帧信息）
- 11010 = 仲裁丢失在识别码 ID4（用于扩展帧信息）
- 11011 = 仲裁丢失在识别码 ID3（用于扩展帧信息）
- 11100 = 仲裁丢失在识别码 ID2（用于扩展帧信息）
- 11101 = 仲裁丢失在识别码 ID1（用于扩展帧信息）
- 11110 = 仲裁丢失在识别码 ID0（用于扩展帧信息）
- 11111 = 仲裁丢失在 RTR 位（用于扩展帧信息）

这个寄存器包含了总线错误类型和位置信息、仲裁丢失信息。对 CPU 来说是只读寄存器。

仲裁丢失或总线发生错误时，会产生相应的中断（中断允许）。同时，位流处理器的当前位置被捕捉送入该寄存器。一直到用户通过软件读这个值或者进入复位模式之后，寄存器中的内容都不会变。读出后，捕捉机制又被激活了。新的总线中断直到捕捉寄存器被读一次才有可能有效。

25. 3. 4 CANx_EROR CAN 错误寄存器

表 25-6 CANx_EROR CAN 错误寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	1 22	1 21	1 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W																																
位名									CANEWL<7:0>							CANTXE<7:0>							CANRXE<7:0>									

CANEWL<7:0>: 错误报警限制寄存器, 默认值为 0x60。

CANTXE<7:0>: 发送错误计数器

CANRXE<7:0>: 接收错误计数器

这个寄存器定义了错误报警限制的值, 硬件复位时默认值是 0x60 (96)。发送和接收错误计数器硬件复位后被初始化为 0。工作模式中, 这个寄存器是只读的, 在复位模式中可以读/写此寄存器。

如果因为错误发生了总线关闭, CANRXE 错误计数器就被初始化为 0, CANTXE 会被复位成 0x7F。总线关闭期间写这个寄存器是无效的。注意, 只有进入复位模式, 才有可能由 CPU 写操作使 RXE 错误计数器发生改变。

CANTXE 反映了发送错误计数器的当前值, 硬件复位后寄存器被初始化为 0。

当发送错误计数器超过限制 (255), 总线状态位 BOFF 被置为 1 (总线关闭), CAN 控制器将设置复位模式位为 1, 而且产生错误报警中断 (相应的中断允许时)。如果发生总线关闭, CANTXE 被初始化为 127。CPU 将复位模式位 RSMOD 清除后, CAN 控制器将等待 128 个总线空闲信号, 之后总线状态位被清除 (总线开启), 错误计数器复位且产生一个错误报警中断和错误消极中断 (中断允许时)。这段时间里读发送错误计数器, 将反映出总线关闭恢复的状态信息。

当发送或接收错误计数器计数值大于错误报警限制寄存器值时, 会产生错误报警中断标志, 同时出错状态位 CERROR 置一。当发送或接收错误计数器计数值大于 127 时, 节点进入消极错误状态, 会产生错误消极中断标志。

如果总线关闭是激活的, 写访问 CANTXE 的 0-254 单元会清除总线关闭标志, 复位模式被清除后控制器会等待一个 11 位的连续隐藏位。

错误计数机制具体参考本章“错误处理”小节。

25. 3. 5 CAN 验收滤波器组

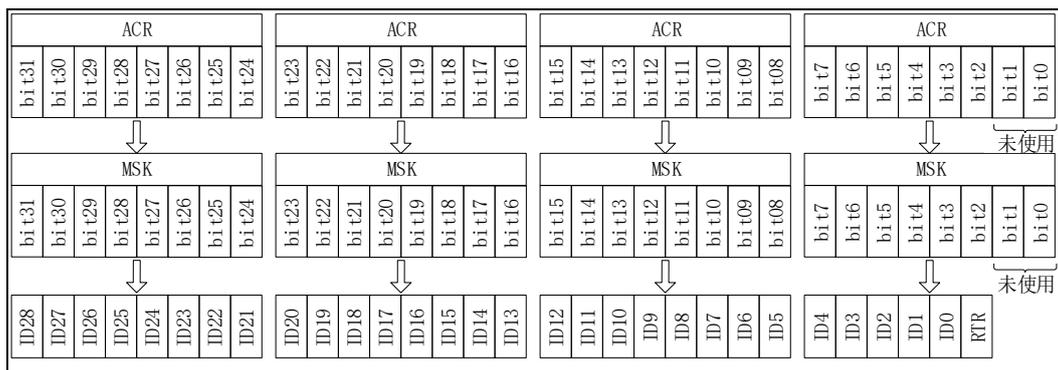
无论接收到的消息是否通过验收滤波器组寄存器的校验, CAN 控制寄存器都会将当前接收到的帧按顺序存入 RAM 中 (见本章“RAM 接收”小节)。

但是, 如果该帧消息通过了验收滤波器组的校验, 则 RAM 地址会加 0x10, 指向下一帧消息的接收地址。如果该帧消息没有通过验收滤波器组寄存器的校验, RAM 地址保持不变, 则下一次接收到的帧会将该帧覆盖。

当要接收的信息已经成功通过验收滤波器的时候, CAN 控制器需要在接收缓冲器中有足够的空间来存储信息描述符和每一个接收的数据字节。如果接收缓冲器没有足够的空间来存储信息, 数据满状态位置位, 触发相应中断。

验收滤波器由验收代码寄存器 (ACR) 和验收屏蔽寄存器 (CANMSK) 定义。要接收的信息的识别码在验收代码寄存器中定义, 相应的验收屏蔽寄存器置位允许定义某些位为“不影

图 25-4 扩展帧验收滤波示意图



25.3.6 CAN 发送缓冲器

发送缓冲器被分为描述符区和数据区。描述符区的第一个字节是帧信息字节，它说明了帧格式（标准帧或扩展帧）、远程或数据帧和数据长度。标准帧有两个字节的识别码，扩展帧有四个字节的识别码。数据区最多长 8 个数据字节。发送缓冲器长 13 个字节。

25.3.6.1 CANx_INFR CAN 数据信息寄存器

表 25-9 CANINFR CAN 数据信息寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W																									R/W	R/W			R/W	R/W	R/W	R/W
位名																								IDE	RTR	DLC<3:0>						

IDE: 帧格式选择位

0 =标准帧格式 SFF

1 =扩展帧格式 EFF

RTR: 远程发送请求位

0 =数据帧

1 =远程帧

DLC<3:0>: 数据长度代码

数据字节长度范围是 0-8，编码形式如下

数据字节数=8*DLC3+4*DLC2+2*DLC1+DLC0

大于 8 的数据长度代码是不可用的。如果大于 8，将以 8 个字节计。

25.3.6.2 CANx_TX0R CAN 数据寄存器 0

表 25-10 CANx_TX0R CAN 数据寄存器 0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	CANTX0<31:0>																															

在标准帧格式下，高 11 位是识别码，对应 ID28 到 ID18；中间有 5 位没有被使用；低

16 位是待发送的数据，CANTX0<15: 8>先发，然后发送 CANTX0<7:0>。

在扩展帧格式下，高 29 位是识别码，对应 ID28 到 ID0 位。ID28 将最先被发送。

TX0R 标准帧格式如下：

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18							CANTXDT0					CANTXDT1									

TX0R 扩展帧格式如下：

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0				

25.3.6.3 CANx_TX1R CAN 数据寄存器 1

表 25-11 CANx_TX1R CAN 数据寄存器 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CANTX1<31:0>																															

CANTX1: 待发送数据，从 CANTX1<31:24>开始发送，每次 8 位。

25.3.6.4 CANx_TX2R CAN 数据寄存器 2

表 25-12 CANx_TX2R CAN 数据寄存器 2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CANTX2<31:0>																															

CANTX2: 待发送数据，从 CANTX2<31:24>开始发送，每次 8 位。

25.3.7 CAN 接收缓冲寄存器组

CAN 会将当前接收到的帧存放在接收缓冲寄存器中，当接收标准帧和扩展帧时，接收缓冲寄存器组的存储格式分别如下所示。

25.3.7.1 标准帧格式 SFF

CANx_RXDATA0: 一个接收存储区域的第一个数据

表 25-16 CANx_RXDATA3 标准帧 CAN 接收数据寄存器 3

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R		R		R		R		R		R		R		R																	
位名	CANRXDT6								CANRXDT7																							

25.3.7.2 扩展帧格式 EFF

CANx_RXDATA0: 一个接收存储区域的第一个数据

表 25-17 CANx_RXDATA0 扩展帧 CAN 接收数据信息寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	u	0	0	R	u	R	u	R	u
R/W																									R		R		R		R		R	
位名																								IDE	RTR				DLC<3:0>					

IDE: 帧格式选择位

- 0 = 标准帧格式 SFF
- 1 = 扩展帧格式 EFF

RTR: 远程发送请求位

- 0 = 数据帧
- 1 = 远程帧

DLC<3:0>: 数据长度代码

数据字节长度范围是 0-8, 编码形式如下

$$\text{数据字节数} = 8 * \text{DLC3} + 4 * \text{DLC2} + 2 * \text{DLC1} + \text{DLC0}$$

大于 8 的数据长度代码是不可用的。如果大于 8, 将以 8 个字节计。

CANx_RXDATA1: 一个接收存储区域的第二个数据

表 25-18 CANx_RXDATA1 扩展帧 CAN 接收数据寄存器 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	0	0
R/W	R		R		R		R		R		R		R		R		R		R		R		R		R		R		R		R	
位名	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR		

CANx_RXDATA2: 一个接收存储区域的第三个数据

表 25-19 CANx_RXDATA2 扩展帧 CAN 接收数据寄存器 2

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u
R/W	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u
位名	CANRXDT0								CANRXDT1								CANRXDT2								CANRXDT3									

CANx_RXDATA3: 一个接收存储区域的第四个数据

表 25-20 CANx_RXDATA3 扩展帧 CAN 接收数据寄存器 3

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u
R/W	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u	R	u
位名	CANRXDT4								CANRXDT5								CANRXDT6								CANRXDT7									

CANRXDTx 表示接收到的第 x 个字节 DATA。

表 25-21 接收数据字节表

存储空间	名称
CANRXDT0	RX 数据字节 0
CANRXDT1	RX 数据字节 1
CANRXDT2	RX 数据字节 2
CANRXDT3	RX 数据字节 3
CANRXDT4	RX 数据字节 4
CANRXDT5	RX 数据字节 5
CANRXDT6	RX 数据字节 6
CANRXDT7	RX 数据字节 7

25.3.8 RAM 接收

接收缓冲器组中的帧消息会被转存至专用双端口 RAM 中，最多可以保存 16 个帧。每个 CAN 模块都对应一个专用的双端口 RAM，对应关系为：

表 25-22 CAN 和双端口 RAM 对应关系

CAN	DPRAM	地址
CAN0	DPRAM_A	0x1FFE F800 ~ 0x1FFE F8FF
CAN1	DPRAM_B	0x1FFE F900 ~ 0x1FFE F9FF
CAN2	DPRAM_C	0x1FFE FA00 ~ 0x1FFE FAFF
CAN3	DPRAM_D	0x1FFE FB00 ~ 0x1FFE FBFF
CAN4	DPRAM_E	0x1FFE F600 ~ 0x1FFE F6FF
CAN5	DPRAM_F	0x1FFE F700 ~ 0x1FFE F7FF

CPU 可以直接读取 RAM，以获得 CAN 接收到的数据。在每一块区域的 RAM 存储空间里，接收的信息按照帧格式的不同接收的顺序也不同。读取 RAM 数据之后，CPU 可以通

过把释放接收缓冲器位置 1 来释放接收缓冲器的存储空间，信息计数器就会自减 1。如果信息计数器数值为 0，即接收缓冲器中没有未读的信息，就复位接收中断。

通过滤波器的信息在图中所示的 0、1、2 至 15 的区域依次被接收。在 RAM 的 15 区域接收到报文以后，那么再次接收到报文后就回到 0 区域继续接收。如果信息计数器数值为 16，即接收缓冲器中未读的信息数量已满，就会置位数据满中断；最新接收到的数据将被抛弃。

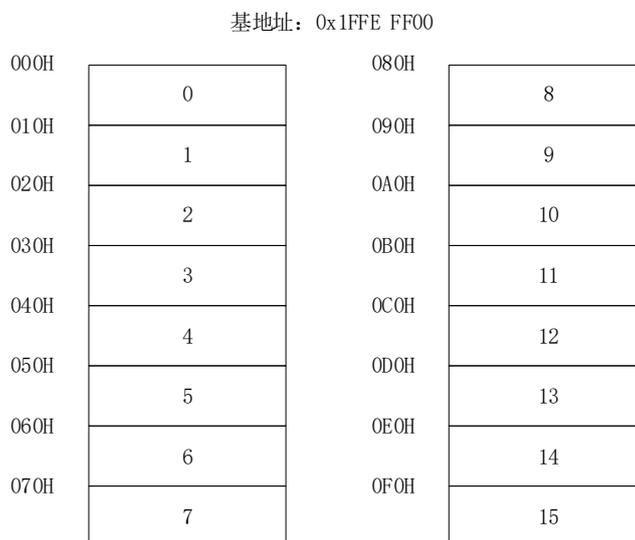


图 25-5 信息接收区域顺序

RAM 每个区域的信息排列顺序与前面讲述的发送缓冲器很相似。每块区域只接收一帧信息里的数据，接收完一帧数据后，自动跳转至下一个区域接收下一帧的信息。以区域 0 为例来说明接收的帧信息的顺序：

在接收标准帧格式 (SFF) 的信息中，第 0 个字节数据是接收帧信息 (RXINF)，接下来是 2 个字节识别码 (RXID0 和 RXID1)，再下面接收的数据字节的多少取决与接收帧信息 (RXINF) 的内容。若接收帧信息里 RTR 位值为 1，则无数据字节的接收，此帧信息到此结束，下帧信息从区域 1 开始接收；若 RTR 位为 0，则接收的数据字节数由 DLC<3:0>的值决定，最多接收 8 字节数据，此帧信息全部接收完成，下帧信息从区域 1 开始接收。

在接收扩展帧格式 (EFF) 的信息中，第 0 个字节数据是接收帧信息 (RXINF)，接下来是 4 个字节识别码 (RXID0、RXID1、RXID2 和 RXID3)，再下面接收的数据字节的多少取决与接收帧信息 (RXINF) 的内容。若接收帧信息里 RTR 位值为 1，则无数据字节的接收，此帧信息到此结束，下帧信息从区域 1 开始接收；若 RTR 位为 0，则接收的数据字节数由 DLC<3:0>的值决定，最多接收 8 字节数据，此帧信息全部接收完成，下帧信息从区域 1 开始接收。

RAM 每个区域可以存放 4 个字的信息 (128bit)，只用来接收一帧信息里的数据。每接收完一帧数据后，当前接收到的信息会被放在接收缓冲寄存器组。同时，接收缓冲寄存器组 CANx_RXDATA0/1/2/3 中内容会被依次存入 RAM 中当前接收区域偏移地址 00/04/08/0CH 的位置。如果本次接收通过了验收过滤，则下次接收时由下一个区域进行接收信息，否则下一帧信息仍由本区域接收 (覆盖上次信息)。

- 1 = 清除发送中断
- CANRXIC: 接收中断清除位
- 0 = 不清除接收中断
- 1 = 清除接收中断
- CANRFIE: 成功接收中断使能位
- 0 = 未使能成功接收中断
- 1 = 使能成功接收中断
- CANRXDE: CAN 接收 DMA 使能位
- 0 = 未使能接收 DMA
- 1 = 使能接收 DMA
- CANTXDE: CAN 发送 DMA 使能位
- 0 = 未使能发送 DMA
- 1 = 使能发送 DMA
- BEIE: 总线错误中断使能位
- 0 = 未使能总线错误中断
- 1 = 使能总线错误中断
- ALIE: 仲裁丢失中断使能位
- 0 = 未使能仲裁丢失中断
- 1 = 使能仲裁丢失中断
- ENIE: 错误消极中断使能位
- 0 = 未使能错误消极中断
- 1 = 使能错误消极中断
- WUIE: 唤醒中断使能位
- 0 = 未使能唤醒中断
- 1 = 使能唤醒中断
- DOVFIE: 数据溢出中断使能位
- 0 = 未使能数据溢出中断
- 1 = 使能数据溢出中断
- EAIE: 错误报警中断使能位
- 0 = 未使能错误报警中断
- 1 = 使能错误报警中断
- CANTXIE: 发送中断使能位
- 0 = 未使能发送中断
- 1 = 使能发送中断
- CANRXIE: 接收中断使能位
- 0 = 未使能接收中断
- 1 = 使能接收中断

25. 3. 10 CANx_IFR CAN 中断标志寄存器

表 25-24 CANx_IFR CAN 中断标志寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
R/W																						R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0
位名																						CANRFIF		CANRXDF		CANTXDF		BEIF		ALIF		ENIF		WUIF		DOVFIF		EAIF		CANTXIF		CANRXIF											

CANRFIF: 成功接收中断标志位

- 0 = 未产生成功接收中断
- 1 = 产生了成功接收中断

CANRXDF: CAN 接收 DMA 使能位

- 0 = 未产生接收触发 DMA 请求
- 1 = 产生接收触发 DMA 请求

CANTXDF: CAN 发送 DMA 使能位

- 0 = 未产生发送触发 DMA 请求
- 1 = 产生发送触发 DMA 请求

BEIF: 总线错误中断标志位

- 0 = 未产生总线错误中断
- 1 = 产生了总线错误中断

ALIF: 仲裁丢失中断标志位

- 0 = 未产生仲裁丢失中断
- 1 = 产生了仲裁丢失中断

ENIF: 错误消极中断标志位

- 0 = 未产生错误消极中断
- 1 = 产生了错误消极中断

WUIF: 唤醒中断标志位

- 0 = 未产生唤醒中断
- 1 = 产生了唤醒中断

DOVFIF: 数据溢出中断标志位

- 0 = 未产生数据溢出中断
- 1 = 产生了数据溢出中断

EAIF: 错误报警中断标志位

- 0 = 未产生错误报警中断
- 1 = 产生了错误报警中断

CANTXIF: 发送中断标志位

- 0 = 未产生发送中断
- 1 = 产生了发送中断

CANRXIF: 接收中断标志位

- 0 = 未产生接收中断
- 1 = 产生了接收中断

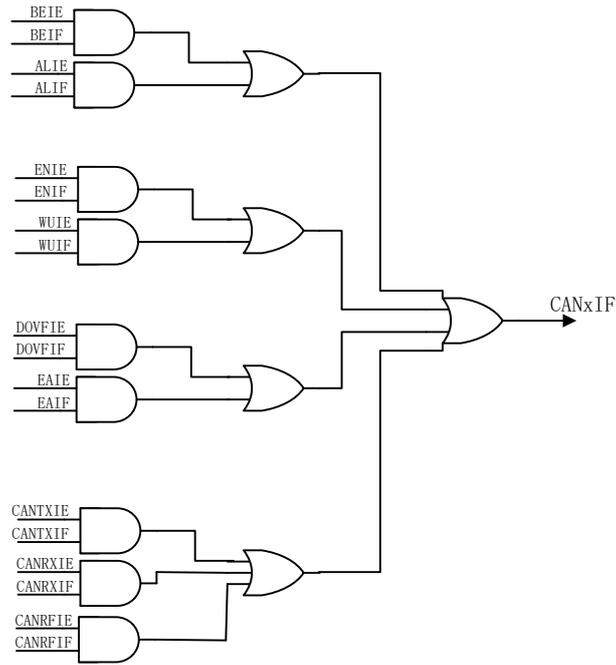


图 25-7 CAN 中断框图

25.4 CAN 模块功能描述

25.4.1 模块功能

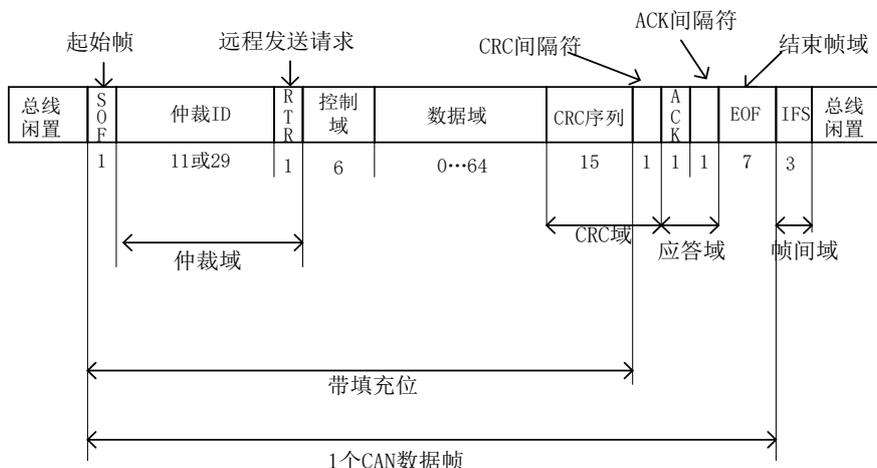


图 25-8 CAN 协议帧格式图

CAN 协议规定，数据帧或者远程帧的帧格式如上图所示：

帧起始由 1 位低电平组成。

仲裁域有两种不同的情况，在标准帧格式下，仲裁域由 11 位识别码和 RTR 位组成；在扩展帧格式下，仲裁域由 29 位识别码以及 SRR 位、IDE 位和 RTR 位组成。

在不同的帧格式下，控制域也有所不同。标准帧格式的控制域为 IDE 位、1 位保留位 r0 和 4 位 DLC 位组成；扩展帧格式的控制域为 2 位保留位 r1、r0 和 4 位 DLC 位组成。

在前面 RTR 位取不同的值，会影响到数据域。当 RTR 位为 0 时，此帧为数据帧，后面有数据域，数据域的字节数由 DLC 控制，从 0-8 字节长度均可；当 RTR 位为 1 时，此帧为远程帧，无数据域。

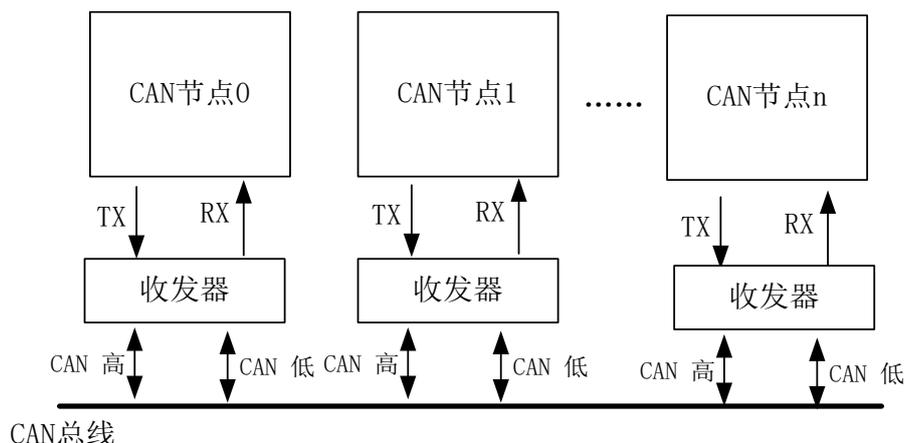
CRC 域为对前面所有数据进行 CRC 校验得出的 15 位长度的 CRC 校验码和 1 位 CRC 界定符组成，CRC 校验多项式为 $g(x)=x^{15}+x^{14}+x^{10}+x^8+x^7+x^4+x^3+1$ 。

应答域为 1 位应答位和 1 位应答界定符组成。

帧结尾由 7 位长度高电平组成。

帧间域由 3 位长度高电平组成。

更多关于 CAN 帧的详细资料请参考 CAN 协议。

25.4.1.1 CAN 系统拓扑结构

图 25-9 CAN 网络拓扑

在 CAN 网络内，各个 CAN 节点的 TX 和 RX 引脚通过收发器同时连接到 CAN 总线上。

25.4.1.2 初始化

要开始软件初始化，通过软件或以硬件复位的方式将 CAN 模式寄存器的 RSMOD 位置位，或进入总线关闭状态（当发送器的错误计数值超过 255 时即进入此状态）。将 RSMOD 位置位后，CAN 总线上的报文传输都将终止，而且 TX 信号将保持为高电平。进入复位模式并不会改变 CAN 控制器、报文对象或错误计数器的配置。不过，某些配置寄存器只能在复位模式下才能访问。

初始化完成后要退出复位模式，必须将 RSMOD 位清零。此后，首先等待一个连续 11 个隐性位序列的产生（表明总线处于空闲状态），依此序列实现与 CAN 总线数据传输的同步，之后才能参与总线活动、开始报文传输。

注：隐性位为“1”，显性位“0”。

25.4.1.3 消息报文

一旦 CAN 模块完成初始化并且 CANx_CTLR 寄存器中的 RSMOD 位被清零，CAN 模块将自动与 CAN 总线同步，然后开始传输报文。收到的每个报文都需要经过验收滤波器的滤波，如果能够通过滤波，将会保存到接收缓冲器中。整个报文（包括所有仲裁位、数据长度码以及 8 个数据字节）都将保存在接收缓冲器中。如果使用标识符屏蔽（验收屏蔽寄存器 CANx_MSKR），那么报文对象中被屏蔽为“不影响”的仲裁位可能会被覆盖。

25.4.1.4 回环模式

回环模式用于实现自检功能。在回环模式下，CAN 控制器在内部将自己发送的信息视为接收到的信息并存储到接收寄存器接收缓冲器中（假如这些信息能够顺利通过验收滤波器）。将 CANx_CTLR 寄存器的 LBACK 位置位即可让 CAN 控制器进入回环模式。

此模式下，CAN 模块的 TX 引脚和 RX 引脚需要同时连接到总线上。

为了彻底隔绝外部事件的影响，在回环模式下，CAN 控制器将会忽略应答错误（即在数据帧/远程帧的应答时间内采样到的隐性位）。接收端 RX 上的实际值会被 CAN 控制器忽略。不过，发送端 TX 上仍然能够监控到发送的信息。

25.4.1.5 安静模式

安静模式下 CAN 控制器不会发送显性位（确认位、错误帧等等），因此在分析 CAN 总线数据流的同时不会对 CAN 总线通讯造成任何影响。将 CANx_CTLR 寄存器的 SILENT 位置位即可进入安静模式。

在安静模式下，CAN 控制器能够接收有效的数据帧和远程帧，但本身只能在 CAN 总线上发送隐性位，并且无法发送任何报文。即使 CAN 控制器必须发送显性位（确认位、过载位或有效错误标志位），也仅在控制器内部连接。也就是说只有 CAN 控制器本身能够监控到发送的显性位，在 CAN 总线上仍然发送的是隐性位。

25.4.1.6 睡眠模式

睡眠模式位设为 1，CAN 将进入睡眠模式。至少破坏以下三种情况之一时将会产生唤醒中断：睡眠模式位电平设为低、总线进入活动状态、CAN 中断（需要使能），随后 CAN 将被唤醒。唤醒后，振荡器启动且产生一个唤醒中断。

由于总线活动唤醒的直到检测到 11 个连续的隐藏位（总线空闲序列）后才能接收这条信息。注意在复位模式中是不能设置睡眠模式位的。清除复位模式后，再一次检测到总线空闲时，睡眠模式位的设置才开始有效。

25.4.1.7 信息发送

CPU 在发送缓冲器中写入帧信息、识别码（标准帧和扩展帧的识别码寄存器有不同的长度）、数据等信息后，有如下几种发送模式：

- 置 1 命令位 TXR，会立即产生一次消息的发送，当发生错误或者仲裁丢失时会进行重发。
- 同时设置命令位 TXR 和 ATX，会立即产生一次消息的发送，当发生错误或者仲裁丢失时不会进行重发（单次发送）。
- 置 1 命令位 LBACK、TXR 后会立即产生一次自接收性质的信息发送，发生错误、仲裁丢失时会进行重发。
- 置 1 命令位 LBACK、TXR 和 ATX 后会立即产生一次自接收性质的信息发送，发生错误、仲裁丢失时不会进行重发的（单次发送）。
- 当发送错误或者仲裁丢失时，若没有使能 ATX，模块会进行重发。在重发过程中，置 1 命令位 ATX 会中断重发并回到空闲状态。

注意，如果需要使用单次发送功能，ATX 需要和 ATR 同时置 1。在置 1 命令位 ATR（ATX）前，请先执行一次 ATR（ATX）的清零操作，以保证启动正常。

25.4.1.8 信息接收

在配置完相应的波特率寄存器、验收过滤寄存器，并退出复位后，等待 11 个位时间，模块会进入空闲状态，在空闲状态时如果检测到总线上的帧，将会开始信息接收。

无论接收到的消息是否通过验收滤波器组寄存器的校验，CAN 控制寄存器都会将当前接收到的帧按顺序存入接收缓冲器并转存到 RAM 中。

但是，如果该帧消息通过了验收滤波器组的校验，则 RAM 地址会加 0x10，指向下一帧消息的接收地址。如果该帧消息没有通过验收滤波器组寄存器的校验，RAM 地址保持不变，则下一次接收到的帧会将该帧覆盖。

举例说明，假设现在模块依次连续接收到 3 个有效帧：帧 A、帧 B 和帧 C，其中仅仅帧 A 和帧 C 能够通过验收过滤。假设帧 A 成功接收后被存放在 RAM 地址 0x20。此时模块又接收到帧 B，帧 B 会被存放在 RAM 地址 0x30（没有通过验收过滤）。最后模块又接收到帧

C, 帧 C 仍然会被存放在 RAM 地址 0x30(将帧 B 覆盖)。

25.4.1.9 CAN 标志位说明

CAN 模块在收发时有以下常用的标志位。

TXSTA 发送状态标志位, 当节点处于空闲和接收状态时, TXSTA=0; 在节点发送帧起始位时, TXSTA 立即置 1, 当发送完帧间域并回到空闲状态时 TXSTA 自动清零。当出现发送错误且主动发送错误帧时, TXSTA 仍然保持 1。

RXSTA 接收状态标志位, 当节点处于空闲和发送状态时, RXSTA=0; 在节点开始接收帧消息时, RXSTA 立即置 1, 当接收完帧间域并回到空闲状态时 RXSTA 自动清零。当出现接收错误且接收错误帧时时, RXSTA 仍然保持 1。

TCSTA 发送完毕状态, 该位初始状态为 1, 置 1 发送请求 TXR 位后 TCSTA 标志会立即清零, 等到当前发送帧的帧结尾发送完毕后(或者遇到了错误或总裁丢失且没有使能重发送时), TCSTA 重新置 1。

TXBSTA 发送缓冲器状态, TXBSTA=0 时表示模块接收到了发送请求并处于发送状态中, 此时禁止 CPU 写发送缓冲器, 等到当前发送帧的帧结尾发送完毕后(或者遇到了错误或总裁丢失且没有使能重发送时)TXBSTA 重新置 1。当模块处于空闲且没有接收到发送请求, TXBSTA 会一直保持 1, 表示允许 CPU 写发送缓冲器。

DOSTA 数据满状态,当 CANRMC<4:0>=0B10000 时, 如果之后又接收到了新的帧, 当新帧接收完毕后(接收到的该帧不会被储存), DOSTA 位会立即置 1, 表示当前 RAM 已经接收到了 16 帧消息(上限)。通过 RELRX 位将 CANRMC<4:0>减至小于 0B10000, 然后置 1 命令位 RELF 可以清零标志位 DOSTA。

RXBSTA 数据空状态, 当 CANRMC<4:0>≠0B00000 时, RXBSTA 置 1 表示 RAM 中有未读信息。RXBSTA=0 表示 RAM 中无未读信息。

成功接收中断标志位 CANRFIF, 每次成功接收时(通过验收过滤), 该位置 1。该位可以通过 CANRXIC 位清零。

接收中断标志 CANRXIF, 该标志位置 1 和清零机制与 RXBSTA 位完全相同。

发送中断标志 CANTXIF, 当模块将一帧消息发送完成后(发送完帧结尾), CANTXIF 置 1, 该位可以通过 CANTXIC 位清零。注意, 使用单次发送时, 如果出现了仲裁丢失, 会在 ALIF 置 1 或的下一个 bit 位将 CANTXIF 置 1。如果在单次发送时出现了总线错误, 也会在发生错误后的下一个 bit 位将 CANTXIF 置 1。在使用重发功能时, CANTXIF 标志会在重发完成后置 1。

总线错误中断标志 BEIF, 当总线出错时模块会立即将 BEIF 标志置 1, 该位可以通过 BEIC 位清零。

仲裁丢失标志 ALIF, 当模块丢失发送仲裁时会立即将 ALIF 标志置 1, 该位可以通过 ALIC 位清零。当发生仲裁丢失后, 需要读 RCR 记录寄存器, 否则下次仲裁丢失时不会将 ALIF 标志置 1

数据溢出中断标志 DOVFIF, 置起时机与 DOSTA 位相同, 但是 DOVFIF 可以直接通过 DOVFIC 清零。

25.4.1.10 DMA 请求

CAN 模块支持发送和接收的 DMA 请求, 通过 CANx_IER 寄存器的 CANTXDE 和 CANRXDE 位可以分别控制发送和接收的 DMA 请求使能。

当 CANTXDE 位置 1 时, 将会使能发送触发 DMA 请求。当 DMA 发送一帧数据时, 会触发发送的 DMA 请求, 并将 CANTXDF 位置 1, 并触发相应的 DMA 事件(如果配置了相

应的 DMA 通道)。当 DMA 被触发后，CANTXDF 标志会自动清零。

当 CANRXDE 位置 1 时，将会使能接收触发 DMA 请求。当 DMA 接收一帧数据时，会触发接收的 DMA 请求，并将 CANRXDF 位置 1，并触发相应的 DMA 事件（如果配置了相应的 DMA 通道）。当 DMA 被触发后，CANRXDF 标志会自动清零。

25.4.1.11 错误处理

当发生总线通信错误时，会将总线中断标志错误 BEIF 置 1。发生错误的 CAN 模块始终处于 3 种状态之一：主动错误状态、消极错误状态和总线关闭状态。

主动错误状态：主动错误状态是可以正常参加总线通信的状态，处于主动错误状态的节点检测出错误时，输出主动错误标志。

消极错误状态：消极错误状态是容易引起错误的状态，处于消极错误状态的节点能够参加总线通信，但接收时即便检测到了错误也不能积极的发送错误通知，因此整个总线会被认为是没有错误的。处于消极错误状态的节点检测出错误时，会将消极错误标志 ENIF 置 1。此外，处于消极错误状态的节点在连续发送时，帧间隔期间需要插入 8 个隐性位。

总线关闭状态：总线关闭的节点（BOFF 置 1）是不能参加总线上的通信的，此时需要手动将 RSMOD 位清零，并等待总线出现 128 个空闲（1 个空闲即 11 个连续的隐性位）后，才能回到正常状态并恢复通信。

表 25-25 错误计数器与错误状态关系

节点错误状态	发送错误计数 TXE	接收错误计数 RXE
主动错误状态	0~127	且 0~127
消极错误状态	128~255	或 128~255
总线关闭状态	256~---	

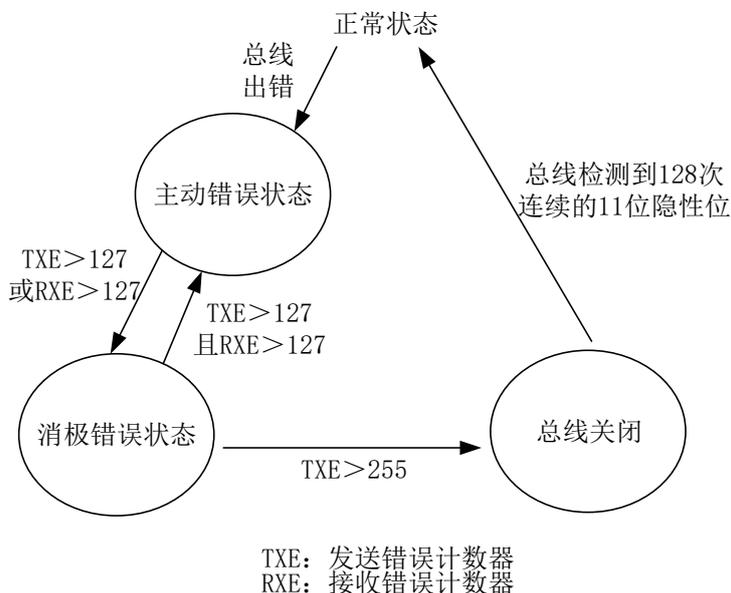


图 25-10 错误状态图

特别的，当总线关闭时，TEC<7:0>会被复位成 0x7F，并将报警中断标志 EAIF 置 1。处于总线关闭的节点需要手动清零 RSMOD 位，并等待总线 128 个连续的空闲位（1 个空闲即 11 个连续的隐性位）后才能重新恢复通信。当通过此操作回到正常状态时，ENIF 会再次置 1（提示已经回到正常状态）。

发送错误计数器 TEC<7:0>和接收错误计数器 REC<7:0>的计数机制如下所示：

表 25-26 错误计数器计数条件

错误条件	发送错误计数值 TEC	发送错误计数值 REC
接收节点检测出错误	--	+1
接收节点在发送完主动错误标志后检测到的第一个位为显性位	--	+8
发送节点在发送主动错误标志时	+8	--
发送节点在发送主动错误标志或过载标志时，检测出错误	+8	--
接收节点在接收主动错误标志或过载标志时，检测出错误	--	+8
各节点从主动错误标志、过载标志开始检测出连续 14 个位的显性位时，之后每检测出连续的 8 个显性位时。	发送时+8	接收时+8
检测出消极错误标志后出现 8 个连续的显性位时	发送时+8	接收时+8
发送节点正常发送数据结束时，CRC 正确且收到了 ACK	-1 TEC=0 时不变	--
接收节点正常接收数据结束时，CRC 未检测出错误且返回 ACK	--	-1
节点关闭，且重新清零 RSMOD 位后，检测到 128 次连续 11 个隐性位	TEC=0	REC=0

25.4.1.12 低功耗 CAN

CAN0 可以作为低功耗 CAN (LP-CAN) 使用。低功耗的 CAN 可以在除待机和关断模式以外的模式中工作。当单片机处于停止模式下，内核已经掉电，此时低功耗 CAN 仍可以保持在工作状态，并能通过中断信号唤醒 CPU。

为了让 CAN0 进入低功耗模式，需要使能 PM_CTL2 寄存器的 CAN0LPEN 位不让 CAN0 复位，同时设置 CAN0_CTLR 的 CANCLKS<1: 0>=11 即选用内部低频振荡器 INTLF 作为相应的时钟源，并将 CAN0CLKLPEN 置一允许内部低频振荡器作为 CAN00 工作时钟，同时保持时钟源处于工作状态，这样就可以在低功耗模式下运行。

CAN0 处于低功耗模式工作时，可以使用 PA13~PA14 口或者 PA5~PA6 作为引脚，可以通过 CAN0_IER 寄存器的 LPCANGPIOSEL1 位和 LPCANGPIOSEL0 位进行选择。

25.5 CAN 的使用

- (1) 先在复位状态下使能外设模块时钟，再退出复位（低功耗 CAN 的外设复位和时钟使能有 2 套控制）；正确设置 MCU 时钟、中断控制；正确设置 FlexCAN 模块 RX 和 TX 引脚的重映射功能；
- (2) 设置 CANCTLR 寄存器 CANEN 位为 1，使能 CAN 模块；设置 CANCTLR 寄存器的 RSMOD 位为 1，使 CAN 进入复位模式（硬件复位之后，CAN 模块将自动进入复位模式）；
- (3) 配置 CAN 波特率设置位速率；
- (4) 如果需要，可以修改错误报警限制寄存器（默认值为 96）、TX 错误计数器和 RX 错误计数器的值；一般情况下，这三个寄存器不需要设置；

- (5) 设置 CAN 验收代码寄存器和 CAN 屏蔽寄存器;
- (6) 设置 CANCTLR 寄存器的 RSMOD 位为 0, CAN 模块退出复位模式 (硬件会等待 11 个位时间后, 模块才会开启并进入空闲状态);
- (7) 读取状态寄存器的 TXBSTA 位, 如果为 1, 将需要发送的数据按照格式填入发送缓冲器;
- (8) 设置 CANCTLR 寄存器的 LBACK、ATX 和 TXR 位 (注意它们之间的配合使用, 不同组合将导致不同的发送情况, 具体参考“信息发送”小节), 使能发送。

注: 执行完步骤(6)后模块已经可以用于接收, 步骤(7)(8)为发送功能的使用。

- 0110 = 计数时钟为 INTLF/2048
- 0111 = 计数时钟为 INTLF/4096
- 1000 = 计数时钟为 INTLF/8192
- 1001 = 计数时钟为 INTLF/16384
- 1010 = 计数时钟为 INTLF/32768
- 1011 = 计数时钟为 INTLF/65536
- 1100 = 保留
- 1101 = 保留
- 1110 = 保留
- 1111 = 保留

26. 2. 2 IWD_T_FD 独立看门狗喂狗寄存器

表 26-3 IWD_T_FD 独立看门狗喂狗寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
位名	IWD _T FD<31:0>																															

往 IWD_T_FD 中写入 0X55AA_55AA 完成喂狗，清零看门狗计数器。喂狗完成之后，该寄存器自动清零。该寄存器为只写寄存器，读出永远为 0。

26.3 独立看门狗原理

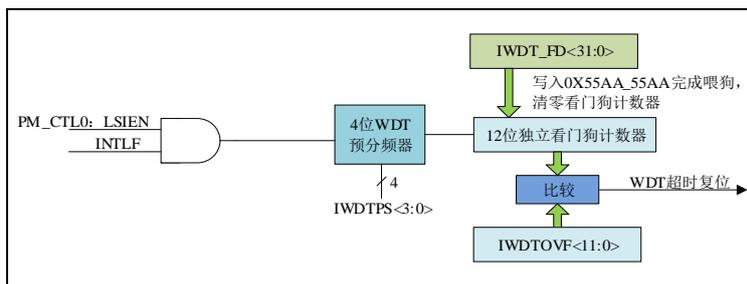


图 26-1 独立看门狗原理框图

26.3.1 独立看门狗的计数时钟

独立看门狗计数器的时钟源为内部低频振荡器，配备 4 位预分频位 IWDTPS<3:0>，可以为看门狗提供较宽的时钟频率选择。在使用独立看门狗时，必须通过 PM_CTL0 寄存器的 LSIEN 位使能内部低频振荡器；其他对内部低频振荡器的使能方式，可能导致独立看门狗无法正常运行从而引起芯片异常。

26.3.2 独立看门狗的喂狗方式

在独立看门狗定时器溢出之前，往 IWDTFD 寄存器中写入 0X55AA_55AA，清零独立看门狗计数器，以完成喂狗操作。喂狗完成之后，IWDTFD 寄存器自动清零；该寄存器为只写寄存器，读出永远为 0。

如未能及时喂狗，独立看门狗计数器发生溢出事件，将对整个芯片（除备份域外）进行复位操作，默认情况下也会对独立看门狗本身进行复位，可以通过配置 PM_CTL0 寄存器的 IWDTRMSEL 位，选择是否对独立看门狗本身进行复位操作。

26.3.3 独立看门狗的复位周期

独立看门狗的复位周期可根据如下公式进行计算：

$$T_{IWDT} = \frac{2^{IWDTPS+5}}{F_{\text{时钟源}}} \times IWDTOVF<11:0>$$

公式 26-1

26.4 IWDG 默认使能功能

除了 IWDG_CTL 寄存器的 IWDGTEN 位外，PM_CAL0 寄存器的 PM_IWDGTEN 位也可以用作使能 IWDG 模块。

用户可以在校准时，如果将 PM_CAL0 寄存器的 PM_IWDGTEN 写成 1，那么 IWDG 将处于“默认使能”模式，且在产生 IWDG 复位时，不会复位 IWDG 自身。将 PM_IWDGTEN 置 1 也会使能内部低频 INTLF，以作为 IWDG 工作时的时钟源，IWDG 的溢出值默认为 0xFFF。

26.5 独立看门狗的使用注意事项

在使用独立看门狗时，必须通过 PM_CTL0 寄存器的 LSIEN 位使能内部低频振荡器；其他对内部低频振荡器的使能方式，可能导致独立看门狗无法正常运行从而引起芯片异常。

独立看门狗位于备份域内，在配置独立看门狗前，需要：

- (1) 将 OSC_CTL0 寄存器的 PMWREN 位置 1，开放备份域的读写权限；
- (2) 将 PM_CTL0 寄存器的 IWDTCLR 位置 1，将 IWDT 退出复位。

27 窗口看门狗（WWDT）

27.1 概述

窗口看门狗通常被用来监测由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。

WWDT 最适合那些要求看门狗在精确计时窗口起作用的应用程序。通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

特点：

- 可编程自由运行计数；
- 可编程预分频；
- 提供中断；
- 避免复位（喂狗）：窗口内写计数器或利用中断写计数器。

27.2 窗口看门狗相关寄存器

表 27-1 窗口看门狗相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	WWDT_CTL	R/W	窗口看门狗控制寄存器	0x0000 0000
0x004	WWDT_CNT	R/W	窗口看门狗计数寄存器	0x0000 0000

基地址：0x4000 1280

27.2.1 WWDT_CTL 窗口看门狗控制寄存器

表 27-2 WWDT_CTL 窗口看门狗控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																	WWDTIC	WW<6:0>						WPS<4:0>				WWDTIF	WWDTIE	WWDTEN			

WWDTIC: 窗口看门狗中断标志位清零位，需要软件置位和清零。

0 = 对 WWDTIF 无操作

1 = 对 WWDTIF 进行清零操作

WW<6:0>: 窗口看门狗配置位

WPS<4:0>: 窗口看门狗预分频位

00000 = WWDTCLK 为 INTLF

00001 = WWDTCLK 为 INTLF/2

00010 = WWDTCLK 为 INTLF/4

00011 = WWDTCLK 为 INTLF/8

00100 = WWDTCLK 为 INTLF/16

00101 = WWDTCLK 为 INTLF/32

00110 = WWDTCLK 为 INTLF/64

00111 = WWDTCLK 为 INTLF/128

01000 = WWDTCLK 为 INTLF/256

01001 = WWDTCLK 为 INTLF/512

01010 = WWDTCLK 为 INTLF/1024

01011 = WWDTCLK 为 INTLF/2048

01100 = WWDTCLK 为 INTLF/4096

01101 = WWDTCLK 为 INTLF/8192

01110 = WWDTCLK 为 INTLF/16384

01111 = WWDTCLK 为 INTLF/32768

10000 = WWDTCLK 为 INTLF/65536

10001 = WWDTCLK 为 INTLF/131072

10010 = WWDTCLK 为 INTLF/262144

其他 = WWDTCLK 为 INTLF

WWDTIF: 窗口看门狗中断标志位

0 = 未产生窗口看门狗中断

27.3 窗口看门狗原理

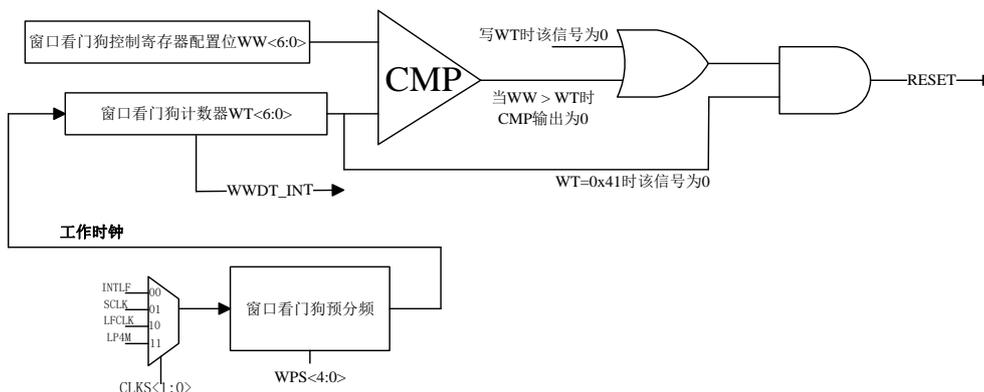


图 27-1 窗口看门狗原理框图 (1)

注：其中 SCLK/LFCLK/LP4M 只在可操作区域内（参照“图 27-2 窗口看门狗原理图 (2)”）可配置使用。

窗口看门狗的 7 位计数器在可操作区域内，多种时钟可选择作为计数时钟。在非可操作区，只能用内部低频（INTLF）时钟作为计数时钟，使用前应将 INTLF 打开。窗口看门狗配备 5 位预分频位，可以为窗口看门狗提供较宽的时钟频率选择。

窗口看门狗产生复位信号有两种情况：一种是 WT<6:0> 值为 41H 时；另一种是 WW<6:0> 大于 WT<6:0> 时对 WWDT_CNT 进行写操作。即当过早对 WWDT_CNT 寄存器进行操作（WT < WW）和过迟操作（WT > 40H），窗口看门狗 WWDT 都会产生复位信号。

避免窗口看门狗发生复位，应当在合适的时间段内（ $WW < WWDT_CNT < 41H$ ）对 WWDT_CNT 寄存器进行清零（写 0 操作）。

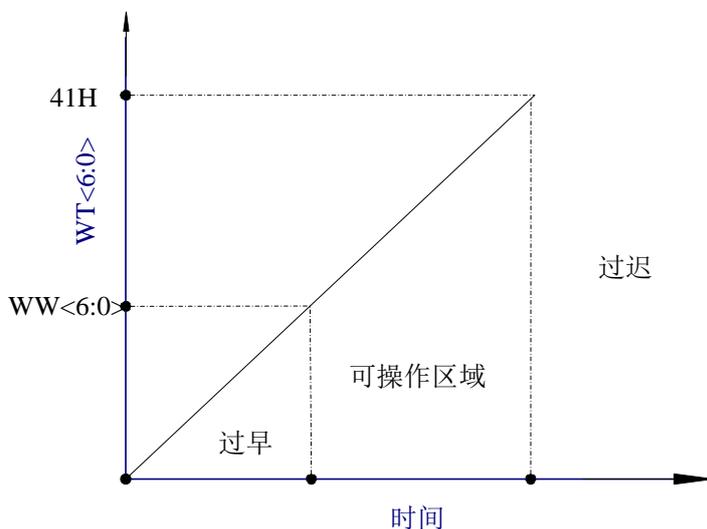


图 27-2 窗口看门狗原理图 (2)

27.4 窗口看门狗中断

窗口看门狗模块还提供一个 WWDT 中断，用户也可以利用此中断来避免窗口看门狗发生复位。窗口看门狗中断会在窗口看门狗计数器 WT<6:0> 计数到 40H 时产生，用户可以在 WT 计数到 41H 前，在中断程序中对 WT<6:0> 清零即可防止复位。通过 WWDTIC 位可以清

零 WWDTIF 标志位。

27.5 时钟选择位

只有在 WWDTEN 位为 0 时,才能够对 WWDT_CNT 寄存器中的 CLKS<1:0>位和 TPEN 位进行写操作,否则对这 3 位的写操作无效。因此,在“喂狗”操作时,不会影响 CLKS<1:0>位和 TPEN 位的配置。

28 USB 模块 (USB)

28.1 概述

该通用串行总线 (USB) 为一个带有单个连接器的串行接口, 可以连接所有 USB 外设到主机系统。下面是 USB 的一些特性。

- 兼容 USB 2.0 全速规范
- 支持控制/批量/中断/等时传输类型
- 支持 USB 挂起/恢复操作
- 提供 8 个可配置为控制/批量/中断/等时传输类型的端点
- 512 字节 SRAM 内置缓存
- 提供远程唤醒功能
- 所有端点均支持双缓冲模式
- 可通过软件来配置端点最大包的长度 (取决于 USB 最大规格)

28.2 USB 模块结构框图

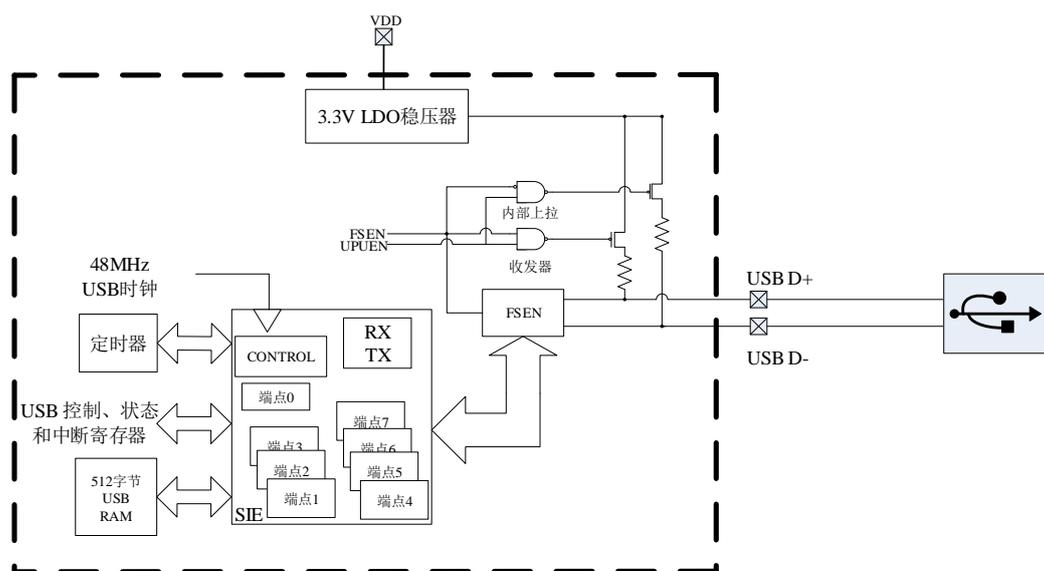


图 28-1 USB 模块结构框图

28.3 USB 模块相关寄存器

表 28-1 USB 模块相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	USB_UCON	R/W	USB 控制寄存器	0x0000 u000
0x004	USB_USTAT	R	USB 状态寄存器	0x0000 uuuu
0x008	USB_UADDR	R/W	USB 地址寄存器	0x0000 0000
0x00C	USB_UFRM	R/W	USB 帧号寄存器	0xuuuu uuuu
0x010	USB_UIR	R/W0	USB 中断标志寄存器	0x0000 0000
0x014	USB_UIE	R/W	USB 中断使能寄存器	0x0000 0000
0x018	USB_UEP0	R/W	USB 端点 0 控制寄存器	0x0000 0000
0x01C	USB_UEP1	R/W	USB 端点 1 控制寄存器	0x0000 0000
0x020	USB_UEP2	R/W	USB 端点 2 控制寄存器	0x0000 0000
0x024	USB_UEP3	R/W	USB 端点 3 控制寄存器	0x0000 0000
0x028	USB_UEP4	R/W	USB 端点 4 控制寄存器	0x0000 0000
0x02C	USB_UEP5	R/W	USB 端点 5 控制寄存器	0x0000 0000
0x030	USB_UEP6	R/W	USB 端点 6 控制寄存器	0x0000 0000
0x034	USB_UEP7	R/W	USB 端点 7 控制寄存器	0x0000 0000
0x038	USB_UBD	R/W	USB Debug 控制寄存器	0x0000 0000
- ¹	USB_BDnSTAT	R/W	缓存区描述符 n 的状态寄存器	0xuuuu uuuu

USB 基地址: 0x4000 0F80

28.3.1 USB_UCON USB 控制寄存器

表 28-2 USB_UCON USB 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W																		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																		DBRST	SE0	PKTDIS	SUSPND	RESUME	USBEN	EYETEST		SPEEDEN		UPUEN	DB<1:0>					

DBRST: 双缓冲区复位位

0 = 不复位双缓冲区指针

1 = 将所有双缓冲区指针复位为偶数缓冲区描述符 BD (Buffer Descriptor) 存储区

SE0: 有效单端 0 电平标志位

0 = 未检测到单端 0 电平

1 = USB 总线上的单端 0 电平有效

PKTDIS: 数据包传送标志位

0 = 允许 SIE 令牌和数据包处理

1 = 已禁止 SIE 令牌和数据包处理, 收到 SETUP 令牌时自动置 1

SUSPND: 暂停(挂起)USB 位

¹ 该寄存器由双端口 RAM 映射, 根据双缓冲配置, 地址会有所变化, 详见“28.5”。

0 = USB 模块和支持电路处于正常工作状态，SIE 时钟以配置的速率运行

1 = USB 模块和支持电路处于休眠模式，SIE 时钟不工作

RESUME: 恢复信号传输使能位

0 = 禁止恢复信号传输

1 = 激活恢复信号传输

USBEN: USB 模块使能位

0 = 禁止 USB 模块和支持电路（断开设备）

1 = 使能 USB 模块和支持电路（连接设备）

EYETEST: USB 眼图测试使能位

0 = 禁止自动眼图测试

1 = 使能自动眼图测试

SPEEDEN: USB 速度选择位

0 = 低速设备

1 = 全速设备

UPUEN: USB 片上上拉使能位¹

0 = 禁止片上上拉

1 = 使能片上上拉（SPEEDEN=1 时在 D+ 上拉，SPEEDEN=0 时在 D- 上拉）

DB<1:0>: 双缓冲区配置位

00 = 禁止偶数/奇数双缓冲区

01 = 使能 OUT 端点 0 的偶数/奇数双缓冲区

10 = 使能所有端点的偶数/奇数双缓冲区

11 = 使能除端点 0 外其它所有端点的偶数/奇数双缓冲区

USB 控制寄存器（USB_UCON）包含的位用于在传送期间控制模块行为。

当使用双重缓冲模式时，USB_UCON 寄存器的 DBRST 位用于控制复位状态。当 DBRST 位置 1 时，所有双缓冲区指针均设置为偶数缓冲区。DBRST 位必须由软件清零。该位在不使用双缓冲模式下将被忽略。

SE0 位用于指示总线上是否出现单端零。使能 USB 模块时，监视此位有助于区分初始上电状态与 USB 复位信号。

USBEN 位用于使能和禁止 USB 模块。将此位置 1 可激活模块，并将缓冲区描述符表中的所有 DBI 位复位为 0。如果使能，该位还将激活 USB 内部上拉电阻。因此，该位可用作与 USB 的软连接/断开。需要为 USB 模块提供有效时钟源，然后才能将 USBEN 位置 1。此外，在使能 USB 模块之前，需要预先完全配置好 USB 模块。

PKTDIS 位是一个标志位，用于指示 SIE 已禁止数据包发送和接收。当收到允许设置处理的 SETUP 令牌时，SIE 将该位置 1，以允许设置处理。单片机不能将该位置 1，只能清零。将该位清零允许 SIE 继续发送或接收令牌和数据包。缓冲区描述符表中的所有待处理事件仍将可用，通过 USB_USTAT 寄存器的 FIFO 缓冲区 PEP 位指示。

RESUME 允许外设通过执行恢复信令来进行远程唤醒。要生成有效的远程唤醒，软件必须将 RESUME 位置 1 并保持 10ms，然后软件将该位清零。有关“恢复信号传送”的详细信息，请参见 USB 2.0 规范。

SUSPND 将模块和支持电路置于低功耗模式（挂起状态）。SIE 的输入时钟也会被禁止。该位必须由软件置 1 来响应 IDLEIF 中断。在出现 ACTVIF 中断之后，应由单片机软件复位该位。当该位有效时，器件保持与总线的连接，但收发器输出保持空闲。在 IDLEIF 请求之

¹ 使能 USB 模块时绝不能更改 UPUEN 和 SPEEDEN 位，用户必须在使能模块前配置这些值

前将该位置 1 会导致不可预测的总线行为。

将 EYETEST 位置 1 可生成自动眼图测试。眼图输出取决于 USB 模块设置，这些设置必须在使用之前配置。模块必须针对眼图进行使能才能正常工作。

EYETEST 位置 1 后，模块将模拟从接收状态到发送状态的转换，然后开始发送 J-K-J-K 位序列（K-J-K-J 表示全速）。当使能眼图测试模式（Eye Pattern Test mode）时，将无限重复该序列。此测试模式用于电路板验证，以协助进行 USB 认证测试。该模式旨在向开发人员展示 USB 信号的噪声完整性，该特性可受到电路板走线、阻抗不匹配以及与其他系统元件的接近程度的影响。它通常不会测试从接收到发送状态的转换。尽管眼图的目的并不是替代更复杂的 USB 认证测试，但在首轮系统调试期间应该能提供帮助。

（注：当模块连接到实际 USB 系统时，绝不能将 EYETEST 位置 1。）

USB 外设内置了一个具有全速和低速 USB 2.0 功能的收发器，在 USB 内部与 SIE 连接。当 USB_UCON 寄存器的 USBEN 位置 1 时，该内部收发器使能。通过将 UCFG 寄存器的 SPEEDEN 位置 1 来选择全速操作。

片上 USB 上拉电阻由 USB_UCON 寄存器的 UPUEN 位控制。仅当 USB_UCON 寄存器的 USBEN 位置 1 并且模块已经过配置可供使用时，上拉电阻才有效。

28. 3. 2 USB_USTAT USB 状态寄存器

表 28-3 USB_USTAT USB 状态寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	u 15	u 14	u 13	u 12	u 11	u 10	u 9	u 8	u 7	u 6	u 5	u 4	u 3	u 2	u 1	u 0																					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u	u	u	u	u	u	u	u	u	R	R	R	R	R	R	u																					
R/W																								R	R	R	R	R	R																								
位名																								PEP<3:0>			PDIR	DBI																									

PEP<3:0>：上一个活动端点的编码编号（代表上次 USB 传送更新的的 BDT 编号）

0000 = 端点 0

0001 = 端点 1

...

0111 = 端点 7

...

1110 = 端点 14

1111 = 端点 15

PDIR：上一个 BD 方向指示位

0 = 上一个事务为 OUT 或 SETUP 令牌

1 = 上一个事务为 IN 令牌

DBI：双 BD 指针指示位（此位仅对于那些奇偶 BD 寄存器可用的端点有效）

0 = 上一个事务是针对在偶数 BD 存储区进行的

1 = 上一个事务是针对在奇数 BD 存储区进行的

USB 状态寄存器（USB_USTAT）报告 SIE 中的事务状态。当 SIE 发出 USB 事务完成中断（TRNSIF 位）时，应读取 USB_USTAT 以确定传送状态。USB_USTAT 包含传送端点编号、方向和双缓冲区指针值（如果使用）。

注：在 TRNSIF 位有效(置 1)后，USB 状态寄存器中的数据将持续两个 SIE 时钟有效。

在系统时钟工作频率为 48MHz 的全速工作模式下，接收事务完成中断与处理 USB_USTAT 寄存器中的数据之间可能需要延时。

USB_USTAT 寄存器实际上是四字节状态 FIFO 的读取窗口，由 SIE 维护。它允许单片机在 SIE 处理其他端点时处理一次传送(图 32.2)。SIE 使用缓冲区读取或写入数据完毕后，将更新 USB_USTAT 寄存器。如果在处理 TRNSIF 位之前执行了其他 USB 传送，SIE 会将下一个事务的状态存储到状态 FIFO 中。

将 TRNSIF 位清零可以将 FIFO 指针加 1。如果 FIFO 保持寄存器中的下一个数据有效，SIE 会在将 TRNSIF 位清零的 6 个指令周期内（或 3 个 SIE 周期内）重新使中断有效。如果不存在其他数据，TRNSIF 位将保持清零状态；USB_USTAT 数据将不再可靠。

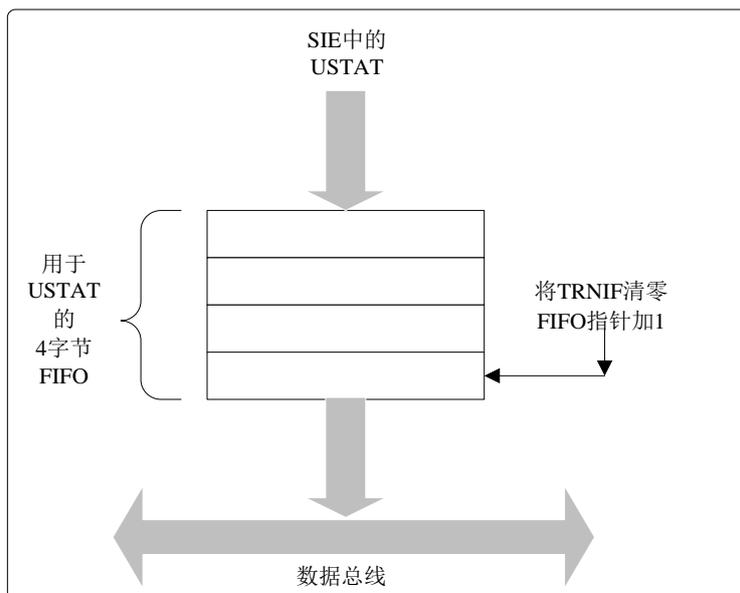


图 28-2 USB_USTAT FIFO

28. 3. 3 USB_UADDR USB 地址寄存器

表 28-4 USB_UADDR USB 地址寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
R/W																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																								UADDR<6:0>																													

USB 地址寄存器包含惟一 USB 地址，当此地址有效时，可被外设识别。当收到 USB 复位信号（通过 USB 复位中断位（RSTIF）指示）或从单片机收到复位信号时，USB_UADDR 寄存器复位为 00H。单片机必须在 USB 启动阶段（枚举）写入 USB 地址，以响应 USB SET_ADDRESS 请求。

28. 3. 4 USB_UFRM USB 帧编号寄存器

表 28-5 USB_UFRM USB 帧编号寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	10	9	8	7	6	5	4	3	2	1	0																			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u	u	u	u	u	u	u	u	u	u	u																			
R/W																						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																						UFRM<10:0>																													

帧编号寄存器包含 11 位帧编号。只要收到 SOF 令牌，就会用当前帧编号更新寄存器。对于单片机，这些寄存器为只读。帧编号寄存器主要用于同步传送。仅当全速模式，USB_UFRM 寄存器的内容才有效。

28.3.5 USB_UIR USB 中断标志寄存器

表 28-6 USB_UIR USB 中断标志寄存器¹

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W																			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																			BTSEIF	BTOEIF	PIDEIF	CRC16EIF	CRC4EIF	DFN8EIF			OSFIF	STLIF	IDLEIF	ACTVIF	TRNSIF	RSTIF	ERRIF		

BTSEIF: 位填充错误标志位

- 0 = 无位填充错误
- 1 = 检测到位填充错误

BTOEIF: 总线周转 (turnaround) 超时错误标志位

- 0 = 未发生总线周转超时
- 1 = 发生总线周转超时 (自上一个 EOP 结束后, 空闲时间超过 16 SIE 周期)

PIDEIF: PID 检查失败标志位

- 0 = PID 检查通过
- 1 = PID 检查失败

CRC16EIF: CRC16 失败标志位

- 0 = CRC16 通过
- 1 = CRC16 失败

CRC5EIF: CRC5 主机错误标志位

- 0 = 令牌包被接受
- 1 = 令牌包由 CRC5 错误而被拒绝

DFN8EIF: 数据字段大小错误标志位

- 0 = 数据字段的字节数是整数
- 1 = 数据字段的字节数不是整数

SOFIF: 帧起始令牌中断位

- 0 = SIE 未收到帧起始令牌
- 1 = SIE 收到帧起始令牌

STLIF: STALL 握手中断位

- 0 = STALL 握手尚未发送
- 1 = SIE 已发送 STALL 握手

IDLEIF: 空闲检测中断位²

- 0 = 未检测到空闲条件
- 1 = 检测到空闲条件 (3 ms 或更长时间的连续空闲状态)

ACTVIF: 总线活动检测中断位³

¹ W0 为写 0 清零, 写 1 无效。

² 检测到空闲状态后, 用户最好将 USB 模块置于暂停模式。

³ 该位通常只在检测到 IDLE 中断事件后才不被屏蔽。

- 1 = 允许 CRC5 主机错误中断
- DFN8EIE: 数据字段大小错误中断允许位
 - 0 = 禁止数据字段大小错误中断
 - 1 = 允许数据字段大小错误中断
- SOFIE: 帧起始令牌中断允许位
 - 0 = 禁止帧起始令牌中断
 - 1 = 允许帧起始令牌中断
- STLIE: STALL 握手中断允许位
 - 0 = 禁止 STALL 中断
 - 1 = 允许 STALL 中断
- IDLEIE: 空闲检测中断允许位
 - 0 = 禁止空闲检测中断
 - 1 = 允许空闲检测中断
- ACTVIE: 总线活动检测中断允许位
 - 0 = 禁止总线活动检测中断
 - 1 = 允许总线活动检测中断
- TRNIE: 事务完成中断允许位
 - 0 = 禁止事务中断
 - 1 = 允许事务中断
- RSTIE: USB 复位中断允许位
 - 0 = 禁止 USB 复位中断
 - 1 = 允许 USB 复位中断
- ERRIE: USB 错误中断允许位
 - 0 = 禁止 USB 错误中断
 - 1 = 允许 USB 错误中断

USB 中断控制寄存器包含 USB 状态中断源的允许位。将这些位中的任何一位置 1 将允许 USB_UIR 寄存器中的相应中断源。

此寄存器中的值仅影响中断条件向单片机中断逻辑的传播。标志位通过各自的中断条件进行设置，允许对它们进行轮询和处理，而不会实际产生中断。

28. 3. 7 USB_UEPn USB 端点 n 控制寄存器

表 28-8 USB_UEPn USB 端点 n 控制寄存器 (n=0,1,2,3,4,5,6,7)

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W																										R/W	R/W	R/W	R/W	R/W		
位名																										EPHSK	EPCONDIS	EPFOUTEN	EPINEN	EPSTL		

- EPHSK: 端点握手使能位
 - 0 = 禁止端点握手 (通常用于同步端点)
 - 1 = 使能端点握手
- EPCONDIS: 双向端点控制位

如果 EPOUTEN = 1 且 EPINEN = 1:

0 = 使能端点 n 控制 (SETUP) 传送; 同时允许 IN 和 OUT 传送

1 = 禁止端点 n 控制传送; 仅允许 IN 和 OUT 传送

EPOUTEN: 端点输出使能位

0 = 禁止端点 n 输出

1 = 使能端点 n 输出

EPINEN: 端点输入使能位

0 = 禁止端点 n 输入

1 = 使能端点 n 输入

EPSTL: 端点 STALL 指示位¹

0 = 端点上未发出 STALL 握手包

1 = 端点上发出 STALL 握手包 (设备无法执行这个请求, 或端点已挂起。)

每个双向端点对都有自己的独立控制寄存器 EPn (其中“n”代表端点编号)。每个寄存器相应控制位的功能都是相同的。

EPHSK 位用于配置端点的 USB 握手。除了使用同步端点时, 该位通常始终置 1。

EPCONDIS 位用于配置通过端点进行的 USB 控制操作。清零此位将使能 SETUP 事务。相应的 EPINEN 位和 EPOUTEN 位必须置 1 才能使能 IN 和 OUT 事务。

注: 对于端点 0, EPCONDIS 位应始终清零, EPOUTEN 与 EPINEN 位应始终置 1。因为 USB 规范将端点 0 标识为默认控制端点。

EPOUTEN 位用于配置来自主机的 USB OUT 事务。将此位置 1 可使能 OUT 事务。类似地, EPINEN 位用于配置来自主机的 USBIN 事务。

EPSTL 位表示端点的 STALL 条件。如果在特定端点上发出 STALL, 该端点对应的 EPSTL 位将由 SIE 置 1。该位将保持置 1 状态, 直到通过软件清零, 或者 SIE 复位。

28.3.8 USB_UDB USB DEBUG 控制寄存器

表 28-9 USB_UDB USB DEBUG 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W																									R/W			R/W	R/W	R/W	R/W	
位名																									DEBUGEN			DEBUG <2:0>				

DEBUGEN: Debug 功能使能位

0 = 禁止 Debug 信号输出

1 = 使能 Debug 信号输出

DEBUG<2:0>: Debug 输出选择位

000 = DebugPort0: 数字输出正端

DebugPort1: 数字输出负端

DebugPort2: 数字内部时钟

001 = DebugPort0: 数字模块处理后接收数据

DebugPort1: 模拟模块接收到的数据

¹ 仅当使能端点 n 时该位有效; 否则忽略该位; n 的范围为 0~7。

- DebugPort2: 数字模块同步时钟
- 010 = DebugPort0: 数字输入正端
- DebugPort1: 数字输入正端
- DebugPort2: 发送使能
- 011 = DebugPort0: RAM 数据 0
- DebugPort1: RAM 数据 2
- DebugPort2: RAM 数据 7
- 100 = DebugPort0: USB 状态机 0 位
- DebugPort1: USB 状态机 1 位
- DebugPort2: USB 状态机 2 位
- 101 = DebugPort0: USB RAM 状态机 0 位
- DebugPort1: USB RAM 状态机 1 位
- DebugPort2: USB RAM 状态机 2 位
- 110 = DebugPort0: RAM 地址 0
- DebugPort1: RAM 地址 1
- DebugPort2: RAM 地址 2
- 111 = DebugPort0: CRC5 运行控制
- DebugPort1: CRC16 运行控制
- DebugPort2: USB 接收控制

28.3.9 USB_BDnSTAT 缓存区描述符 n 的状态寄存器

表 28-10 USB_BDnSTAT 缓存区描述符 n 的状态寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
复位值	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
位名	BADDR/-																BC<7:0>										UOWN	DTS/-	PID3	PID2	DTSEN/PID1	BSTALL/PID0	BC9	BC8

UOWN: USB 拥有位¹

- 0 = 单片机内核拥有 BD 及其相应的缓冲区
- 1 = SIE 拥有 BD 及其相应的缓冲区

UOWN=1 时的位功能:

DTS: 数据翻转同步位²

- 0 = 数据 0 数据包
- 1 = 数据 1 数据包

DTSEN: 数据翻转同步使能位

- 0 = 不执行数据翻转同步
- 1 = 使能数据翻转同步; 同步值不正确的数据包将被忽略, 但 SETUP 事务除外, 即使数据翻转位不匹配, 该事务也将被接受

BSTALL: 缓冲区停止使能位

- 0 = 禁止缓冲区停止

¹ 用户必须在使能 USB 模块之前将该位初始化为所需值。

² 除非 DTSEN = 1, 否则该位将被忽略。

1 = 使能缓冲区停止；如果收到的令牌使用给定单元的 BD，则发出 STALL 握手（UOWN 位保持置 1，BD 值不变）

BADDR<15:0> (BUFFERADDR): RAM 缓存区地址¹（低 9 位有效，以字为单位）

UOWN=0 时的位功能：

PID<3:0>：数据包标识符位²

上一次传送接收到的令牌 PID 值（仅限 IN、OUT 或 SETUP 事务）

UOWN=0/1 时的公共位：

BC<9:8>：字节计数 bit9 和 bit8 位

这些位由 SIE 更新，反映在 IN 令牌期间发送的字节数或在 OUT 令牌期间接收的字节数。连同 BC<9:0>，有效字节计数为 0-1023。

当 UOWN = 1 时，寄存器用作控制位。

USB_BDnSTAT 寄存器的数据翻转同步使能（Data Toggle Sync Enable, DTSEN）位控制数据翻转奇偶校验检查，并且在该位置 1 时，通过 SIE 使能数据翻转同步。使能后，DTSEN 根据翻转同步（Data Toggle Synchronization, DTS）位的值检查数据包的奇偶性。同步不正确的数据包将被忽略，不会写入 USB RAM，并且 USB 事务完成中断标识 TRNSIF 位不会置 1。但是，SIE 将向主机发送 ACK 令牌，以确认收到。

USB_BDnSTAT 寄存器的缓冲区停止位 BSTALL 为控制传送提供支持，通常是在端点 0 停止一次。它还为 USB 规范第 9 章中指定的 SET_FEATURE/CLEAR_FEATURE 命令提供支持。通常，通过向除默认控制端点外的任何端点连续发送 STALL 来执行这些命令。

BSTALL 位用于使能缓冲区停止。将 BSTALL 置 1 会使 SIE 将 STALL 令牌返回到主机（如果收到的令牌会使用该单元中的 BD）。向主机发出 STALL 时，相应的 EPn 控制寄存器中的 EPSTL 位将置 1，并产生 STALL 中断。UOWN 位保持置 1 状态，且 BD 不会改变，除非收到 SETUP 令牌。在这种情况下，将清除 STALL 条件，并将 BD 的所有权归还给单片机内核。

USB_BDnSTAT 寄存器的 BD 位存储 SIE 字节计数的两个最高有效位；低 8 位存储在相应的 BDnCNT 寄存器中。更多信息，请参见“BD 字节计数”小节。

USB_BDnSTAT 寄存器由 SIE 更新令牌数据包标识符（Packet Identifier, PID），该标识符存储在 USB_BDnSTAT 寄存器的 PID 位中。相应 BDnCNT 寄存器中的传送计数将被更新。溢出 8 位寄存器的值将进位到计数的两个最高有效位，即 USB_BDnSTAT 寄存器的 BC8、BC9 位。

表 28-11 DTSEN 位对奇/偶（DATA0/DATA1）数据包接收的影响

来自主机的 OUT 数据包	USB_BDnSTAT 设置		收到数据包后的设备响应			
	DTSEN	DTS	握手	UOWN	TRNSIF	USB_BDnSTAT 和 USB_USTAT 状态
DATA0	1	0	ACK	0	1	已更新
DATA1	1	0	ACK	1	0	未更新
DATA0	1	1	ACK	1	0	未更新
DATA1	1	1	ACK	0	1	已更新
任一	0	x	ACK	0	1	已更新
任一，有错误	x	x	NAK	1	0	未更新

¹ 在传输完成后，该位会被 USB SIE 修改，故在下次传输开始前，需进行重新配置。

² 如果 PID2 和 PID3 位置 1，则 USB 可能无法进行通信。因此，这些位应始终保持为 0。

注： x=无关位

28.4 USB RAM

USB 数据通过双端口 USB RAM 在单片机内核与 SIE 之间传送。尽管 USB RAM 可供单片机用作数据存储，但 SIE 正在访问的段不应被单片机访问。这里采用了一种信号机制来确定任意时间内对特定缓冲区的访问权限。

28.5 缓冲区描述符和缓冲区描述符表 BDT

USB 使用 512 字节的双端口 RAM 作为 USB 的端点缓冲区，8 个端点共享该缓冲区。双端口通用存储器空间专用于在称为缓冲区描述符表（Buffer Descriptor Table, BDT）的结构中进行端点缓冲区控制，每个端点缓冲区的起始地址与块大小可自定义。这为用户提供了一种灵活的方法来构造和控制各种长度及配置的端点缓冲区。

8 个端点的每个端点的输出和/或输入端口均有各自的 USB_BDnSTAT 寄存器组。而每个寄存器包含以下信息：

- BDnSTAT: BD 状态
- BDnCNT: BD 字节计数
- BDnADR: BD 地址

注：1、只要在本文档中出现 BDn 标识，n 就代表其中一个可能的 BD。

USB_BDnSTAT 寄存器组并不象传统单片机 SFR 一样进行硬件映射。如果未使能与特定 BD 寄存器组相应的端点，则不使用其寄存器。但是，这些寄存器并不显示为未实现地址，而是显示为可用 RAM。仅当通过将 EPn 寄存器的 EPINEN/EPOUTEN 位置 1 来使能端点时，这些地址的存储器才会用作 BD 寄存器。BD 寄存器组在任何设备复位时都具有不确定的值。

图 32-4 显示了将端点 0 的输出缓冲区定义为 64 字节的 USB_BDnSTAT 寄存器组的示例。对于一组特定的 BD 寄存器，仅当 BD 寄存器组对端点的 EPINEN/EPOUTEN 使能时，这些寄存器才有效。所有 BD 寄存器均在 USB RAM 中提供。每个端点的 BD 都应在使能该端点前设置。

在 IN 传送期间从 USB RAM 发送到主机的数据长度或在 OUT 传送期间从主机接收到并存储在 USB RAM 中的实际数据长度由 BDnCNT 寄存器和 USB_BDnSTAT 寄存器的低两位决定。即最大的数据负载为 10 位，高 2 位在 USB_BDnSTAT 寄存器的第 0、1 位，低 8 位在 BDnCNT 寄存器中。这代表了发送到主机的数据长度或从主机接收到的实际数据长度可为 0 至 1023 字节。

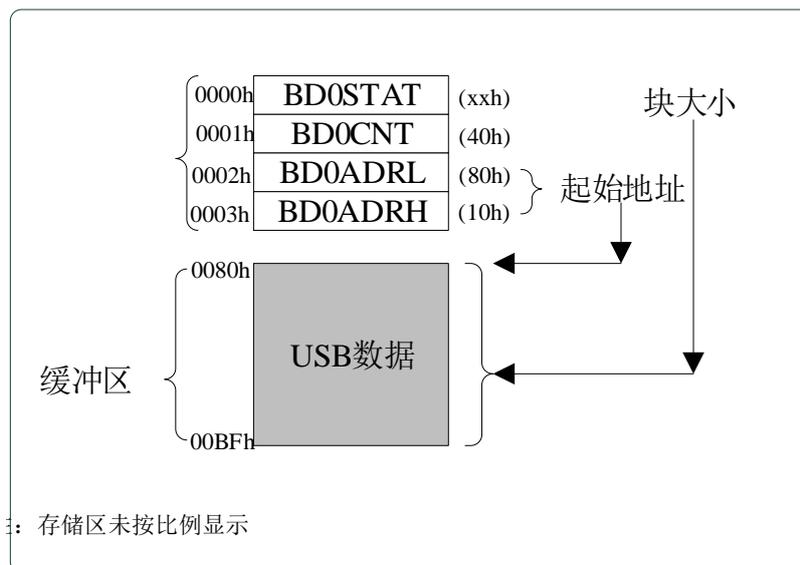


图 28-3 缓冲区描述符示例

28.5.1 BD 状态和配置

BD 的所有权和 USB_BDnSTAT 存储器中各个位的配置根据 UOWN 位电平更改功能。

如果 UOWN 位清零，则单片机能够修改 BD 及其相应的缓冲区。如果 UOWN 位置 1，则 USB SIE 能够修改 BD 及其相应的缓冲区。无论 UOWN 位值为何，UOWN、BC9 和 BC8 位均包含在 USB_BDnSTAT 寄存器中。

当 UOWN 位清零时，BD 条目和缓冲区存储器由单片机内核“所有”。当 UOWN 位置 1 时，则为 USB 外设“所有”。BD 和相应缓冲区只能由“所有者”修改。但是，即使单片机或 USB 不是“所有者”，也可以读取 USB_BDnSTAT 寄存器。

用户必须先通过 USB_BDnSTAT 寄存器配置 USB 外设的基本操作，然后再分配 USB 外设的所有权。即使单片机拥有所有权，也必须设置字节计数和缓冲区单元寄存器。

当 UOWN 位置 1 时，USB 外设拥有所有权，SIE 根据需要更新 BD，覆盖原始 BD 值。因此，用户写入 BD 的值不再可靠。SIE 将使用 PID 令牌更新 USB_BDnSTAT 寄存器，同时也将更新传送计数 (BDnCNT)，在此过程中 BDnADR 也将被修改，变的不在可靠。

在准备与端点通信之前，USB_BDnSTAT 的最低字节 (UOWN 所在的字节) 总是最后更新的一个字节。事务完成后，SIE 将 UOWN 位清零。

当 UOWN 位置 1 时，硬件上并没有任何机制来阻止单片机对存储区的访问。因此，如果单片机尝试在 SIE 拥有 USB RAM 的时候更改该存储区，可能会出现不可预料的后果。同样，在 USB 外设将 USB RAM 所有权归还给单片机之前，读此存储区可能会导致数据出错。

28.5.2 BD 字节计数

字节计数代表在 IN 传送期间将发送的字节总数。在 IN 传送之后，SIE 将发送的字节数返回给主机。

对于 OUT 传送，字节计数代表可接收并存储在 USB RAM 中的最大字节数。在 OUT 传送之后，SIE 将返回实际接收的字节数。如果接收的字节数超出相应的字节计数，将拒绝数据包，并生成一个 NAK 握手信号。在这种情况下，字节计数不会更新。

10 位字节计数分布在 2 个字节中。计数的低 8 位在 USB_BDnSTAT 寄存器的第二个字节中。高 2 位在 USB_BDnSTAT 寄存器的最低字节中。这代表了 0 至 1023 的有效字节范围。

28.5.3 BD 地址验证

BD 地址寄存器对包含相应端点缓冲区的起始 RAM 地址单元。没有可用的硬件机制来验证 BD 地址。

如果 BD 地址的值未指向 USB RAM 中的地址，或者指向另一个端点的缓冲区中的地址，则数据很可能丢失或被覆盖。类似地，将接收缓冲区（OUT 端点）与正在使用的 BD 单元重叠也可能得到意外结果。开发 USB 应用时，用户最好考虑在代码中包括基于软件的地址验证。

28.5.4 双缓冲（乒乓缓冲）

当端点被定义为具有双缓冲端点时，端点有两组 BD 项：一组用于偶数传送，一组用于奇数传送。这允许在 CPU 处理一个 BD 的同时，SIE 处理另一个 BD。这种方式的双重缓冲 BD 可实现 USB 输入/输出的最大吞吐量。

USB 模块的端点缓冲区支持 4 种工作模式：

- 不支持双缓冲区
- 仅支持端点 0 OUT 的双缓冲区
- 支持所有端点的双缓冲区
- 支持除了端点 0 外的所有其他端点的双缓冲区

使用 UCFG 寄存器中的 DB<1:0>位配置双缓冲区设置。USB 模块分别跟踪每个端点的双缓冲指针。当使能模块时，所有指针最初均复位为偶数 BD。事务完成后（UOWN 由 SIE 清零），指针翻转到奇数 BD。下一个事务完成后，指针翻转回偶数 BD，以此类推。

上一个事务的偶数/奇数状态存储在 USB_USTAT 寄存器的 DBI 位中。用户可以使用 DBRST 位将所有乒乓指针复位为偶数。图 32-4 缓冲模式的缓冲区描述符表映射显示了 4 种不同的工作模式以及 BD 如何填充 USB RAM。BD 与特定端点具有固定的关系，具体取决于缓冲配置。表 32-12 详细列出了 BD 到端点的映射。这种关系还意味着，如果端点没有连续使能，BDT 中可能出现空白区域。理论上，这意味着已禁止端点的 BD 可用作缓冲区空间。但实际上，用户应避免使用 BDT 中的此类空间，除非实现了验证 BD 地址的方法。

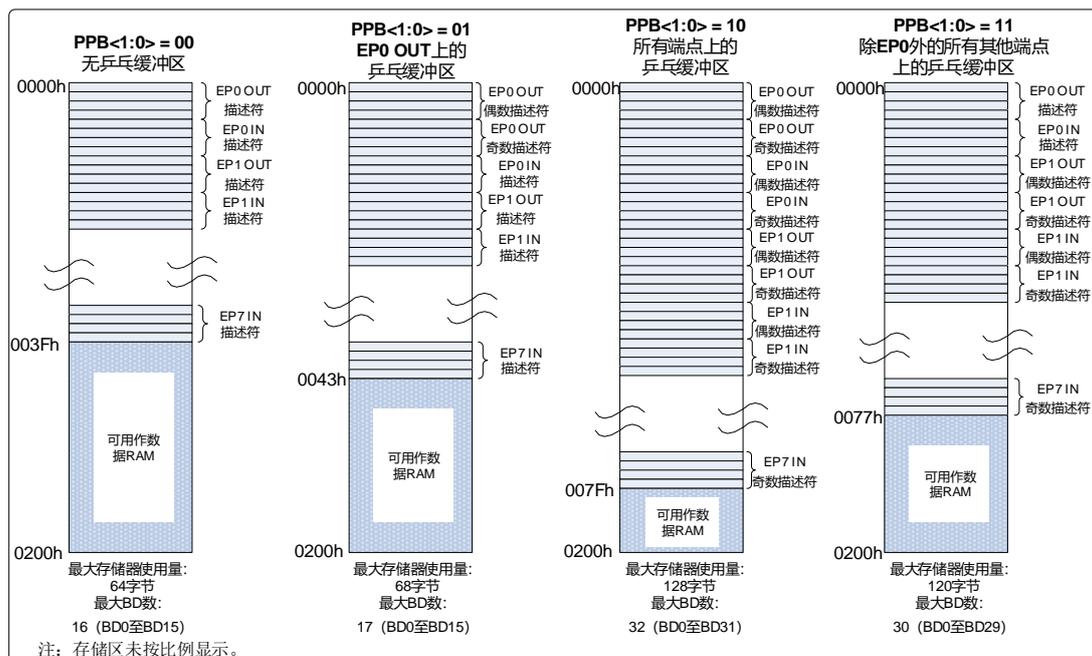


图 28-4 缓冲模式的缓冲区描述符表映射

表 28-12 不同缓冲模式的缓冲区描述符分配

端点	分配给端点的 BD							
	模式 0 (无乒乓缓冲区)		模式 1 (EP0 OUT 上的 乒乓缓冲区)		模式 2 (所有端点上的 乒乓缓冲区)		模式 3 (除 EP0 外的所有其它 端点上的乒乓缓冲区)	
	输出	输入	输出	输入	输出	输入	输出	输入
0	0	1	0(E),1(O)	2	0(E),1(O)	2(E),3(O)	0	1
1	2	3	3	4	4(E),5(O)	6(E),7(O)	2(E),3(O)	4(E),5(O)
2	4	5	5	6	8(E),9(O)	10(E),11(O)	6(E),7(O)	8(E),9(O)
3	6	7	7	8	12(E),13(O)	14(E),15(O)	10(E),11(O)	12(E),13(O)
4	8	9	9	10	16(E),17(O)	18(E),19(O)	14(E),15(O)	16(E),17(O)
5	10	11	11	12	20(E),21(O)	22(E),23(O)	18(E),19(O)	20(E),21(O)
6	12	13	13	14	24(E),25(O)	26(E),27(O)	22(E),23(O)	24(E),25(O)
7	14	15	15	16	28(E),29(O)	30(E),31(O)	26(E),27(O)	28(E),29(O)

注：(E) = 偶数事务缓冲区，(O) = 奇数事务缓冲区

表 32-13 USB 缓冲区描述符表寄存器汇总

名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
USB_BDnSTAT	UOWN	DTS	PID3	PID2	PID1 DTSEN	PID0 BSTALL	BC9	BC8
	字节计数 (BDnCNT)							
	缓冲区地址 (BDnADR)							

注：

- (1) 对于缓冲区描述符寄存器，n 的值可能为 0 至 31 之间的某个值。为了简洁起见，所有 32 个寄存器都显示为一个通用原型。所有寄存器都有不确定的复位值 (0xuuuuuuuu)。
- (2) USB_BDnSTAT 寄存器的 bit <5: 2> 在该寄存器受控于 SIE 后 (UOWN 位置 1)，由 SIE 用来返回 PID<3:0> 值。寄存器由 SIE 控制后，DTSEN 和 BSTALL 写入的值不再有效。
- (3) 在将缓冲区描述符移交给 SIE (UOWN 位清零) 之前，使用 USB_BDnSTAT 寄存器的 bit 5 至 2 配置 DTSEN 和 BSTALL 设置。
- (4) 除非 DTSEN = 1，否则 DTS 位将被忽略。
- (5) 缓存区首地址 0x0000，对应到 RAM 区的地址为 0x1FFE FC00

28.6 USB 中断

USB 模块可生成多个中断条件。为适应所有这些中断源，模块附带自己的中断逻辑结构，与单片机的结构类似。USB 中断通过一组控制寄存器允许，并通过一组独立的标志寄存器捕获。所有源均汇集为单个 USB 中断请求 (EIF6 的 USBIF 位)，来与单片机的中断逻辑配合使用。

下图显示了 USB 模块的中断逻辑，该中断逻辑在 USB 模块中分成两个寄存器。USB 状态中断被视为顶层，通过 USB_UIE 寄存器允许中断，同时通过 UIF 寄存器维护标志。USB 错误条件被视为第二层，通过 UERIE 寄存器允许中断，同时通过 UEIF 寄存器维护标志。任何 USB 中断条件都将触发 USB 错误中断标志，即 UIF 寄存器的 ERRIF 位。

中断可用于捕获 USB 事务中的程序事件。图 32-6 给出了 USB 帧中的一些常见事件及其相应的中断。

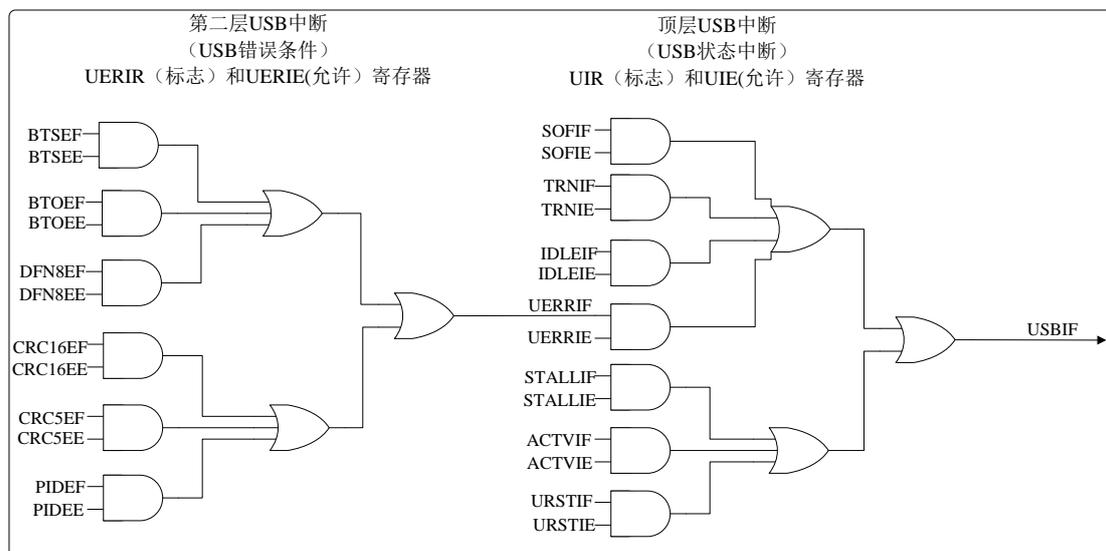


图 28-5 USB 中断逻辑汇集

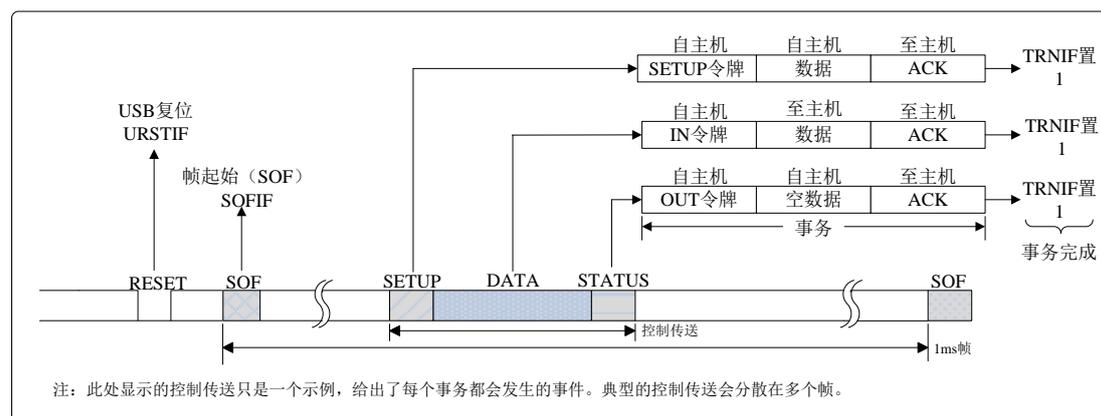


图 28-6 USB 事务和中断事件示例

28.7 SB 时钟源

USB 模块具有特定时钟要求。时钟源必须为 48 MHz。但是单片机内核和其他外设不一定要以该时钟频率运行。

USB 模块的工作时钟的设置，见“OSC_CTL1 振荡器控制寄存器 1”，USB 时钟通过 CK48MS 选择时钟源，利用 CK48DIV 将所选时钟源的时钟分频至 48MHz，然后使能 CK48MEN。

28.8 USB 操作概述

本节介绍一些基本的 USB 概念以及设计 USB 设备所需的有用信息。尽管本节提供了许多信息，但可根据需要参见 USB 2.0 规范，以了解更多详细信息。

28.8.1 分层框架

USB 设备的功能是以分层框架的形式实现的。如图 28-7 USB 层所示。每一层都和设备中的某个功能级别关联。除设备以外的最高层是配置层。一个设备可能有多个配置。例如，某个特定设备可能有多个基于“仅自供电”或“仅总线供电”模式的电源要求。

对于每个配置，可能有多个接口。每个接口可支持该配置的一个特定模式。

接口下面是端点。数据在此层直接移动。端点 0 始终是控制端点，默认情况下，当设备在总线上时，端点 0 必须可用于配置设备。

28.8.2 帧

在总线上通信的信息分成多个 1 ms 的时隙，称为帧。每个帧可包含许多到各个设备和端点的事务。图 32-6 USB 事务和中断事件示例显示了某个帧中的事务的示例。

28.8.3 传输

USB 规范中定义了 4 种传送类型：

同步传输：此类型提供了传送大量数据（最多 1023 字节）的方法，可确保及时传送，但不确保数据完整性。这种方法适合于丢失少量数据无关紧要的流应用，如音频。

批量传输：此类型的传送方法适合传送大量数据，并确保数据完整性，但不保证及时传送。

中断传输：此类型的传送方法可确保及时传送，适合传送少量数据，同时可确保数据完整性。

控制传输：此类型提供设备设置控制。尽管全速设备支持所有传送类型，但低速设备仅支持中断传送和控制传送。

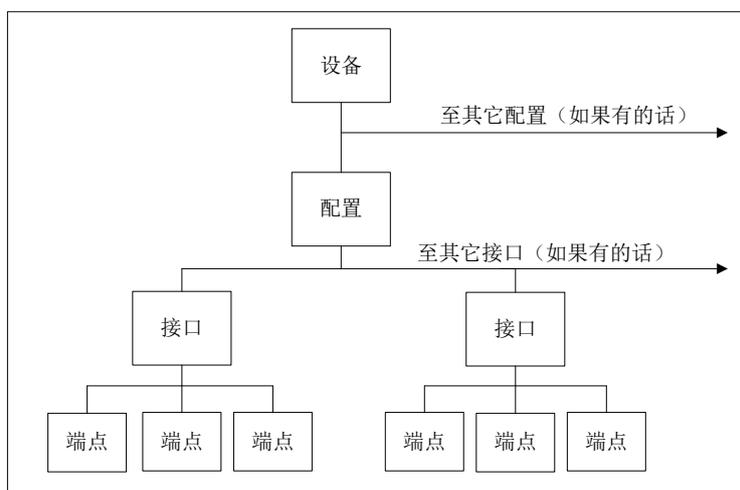


图 28-7 USB 层

28.8.4 枚举

设备最初连接到总线时，主机会进入枚举过程以尝试识别设备。基本上，主机会询问设备，收集功耗、数据速率和大小、协议等信息以及其他描述性信息；描述符包含这些信息。典型的枚举过程如下：

- (1) USB 复位：复位设备，这意味着设备未配置，且没有地址（地址 0）。
- (2) 获取设备描述符：主机请求设备描述符的一小部分。
- (3) USB 复位：再次复位设备。
- (4) 设置地址：主机为设备分配地址。

- (5) 获取设备描述符：主机检索设备描述符，收集制造商、设备类型和最大控制包大小等信息。
 - (6) 获取配置描述符。
 - (7) 获取任何其他描述符。
 - (8) 设置配置。
- 具体的枚举过程取决于主机。

28.8.5 描述符

有 8 种不同的标准描述符类型，其中 5 种对于设备最重要。

28.8.5.1 设备描述符

设备描述符提供常规信息，如制造商、产品编号、序列号、设备类别和配置数。只有一个设备描述符。

28.8.5.2 配置描述符

配置描述符提供有关设备的电源要求以及在此配置中支持多少不同接口的信息。一个设备可能有多个配置（即低功耗配置和高功耗配置）。

28.8.5.3 接口描述符

接口描述符详细说明此接口中使用的端点数，以及接口的类别。一个配置可能有多个接口。

28.8.5.4 端点描述符

端点描述符确定传输类型和方向，以及端点的一些其他定义。一个设备中可以有許多端点，端点可在不同配置间共用。

28.8.5.5 字符串描述符

许多之前的描述符引用了一个或多个字符串描述符。字符串描述符提供有关它们描述的层的可读信息。这些字符串通常显示在主机中，帮助用户识别设备。字符串描述符通常为可选，以节省存储器，并使用 unicode 格式进行编码。

28.8.6 总线速度

每个 USB 设备都必须向主机指示其总线情况和速度。这通过上拉电阻来完成，上拉电阻在发生连接事件时连接到总线。根据设备的速度，上拉电阻将 D+或 D-线连接到 3.3V。对于低速设备，上拉电阻连接到 D-线。对于全速设备，上拉电阻连接到 D+线。

28.8.7 分类规范和驱动程序速度

USB 规范包括多种分类规范，操作系统供应商可选择支持。类别示例包括音频、大容量存储、通信和人机界面（Human Interface, HID）。在大多数情况下，主机侧需要驱动程序，才能与 USB 设备“对话”。在定制应用中，可能需要开发驱动程序。幸运的是，对于最常用的设备类别，已经为最常用的主机系统提供了驱动程序。因此，这些驱动程序可重复使用。

28.9 USB 端点配置概述

在完成对 USB 工作时钟与速度等基础配置后，需要对传输使用的端点进行配置，主要配置寄存器 USB_UEPn 与 USB_BDnSTAT。

配置端点时首先确定需要配置端点的传输类型，比如端点 0，一般用于控制传输；其次确定端点的传输方向；在配置 USB_BDnSTAT 时，要根据双缓冲的设置，确定对应 USB_BDnSTAT 位置，并根据传输数据的大小与 USB RAM 的空余空间确定合适的传输字节数以及数据缓存地址。对于接收 SETUP 的端点，USB_UEPn 通常设置为 0x16，因为 SETUP 传输需要在同一个端点完成输入与输出，所以必须同时使能 EPOUTEN 与 EPINEN；USB_BDnSTAT 的设置则如上所述，对于 SETUP 包，一般使用端点 0，在不使能双缓冲的情况下，端点 0 OUT 的 USB_BDnSTAT 地址为 1FFFFC00H，端点 0 IN 的地址为 1FFFFC04H，详见图 32_4。在设置传输字节数与地址时，需要注意传输要使用到其他端点，防止与其他端点冲突。

对于同步传输的端点，USB_UEPn 通常设置为 0x0C (OUT) 或 0x0A (IN)，由于同步传输不需要握手包，故 EPHSHK 置 0。对于其他传输，只需要按需求进行设置即可。

在端点完成一次数据传输后，USB_BDnSTAT 会被 SIE 写入当次数据传输的信息 (PID，传输字节等)，同时之前的配置信息将被覆盖，因此该寄存器在完成一次数据传输后需要重新配置包括数据地址在内的所有有效位。

29 CFGL 模块 (CFGL)

29.1 概述

可配置逻辑单元 (CFGL_x) 提供可超越软件执行速度限制而工作的可编程逻辑。该逻辑单元最多可接收 16 个输入信号, 并通过使用可配置门将 16 个输入缩减为 4 条驱动 8 种可选单输出逻辑功能之一的逻辑线。

输入源是以下信号源的组合:

- I/O 引脚
- 内部时钟
- 外设
- 寄存器位

可能的配置包括:

- 组合逻辑
 - AND
 - NAND
 - AND-OR
 - AND-OR-INVERT
 - OR-XOR
 - OR-XNOR
- 锁存器
 - S-R
 - 带置 1 和复位功能的时钟控制 D 型锁存器
 - 带置 1 和复位功能的透明 D 型锁存器
 - 带复位功能的时钟控制 J-K 型锁存器

29.2 CFGL 模块结构框图

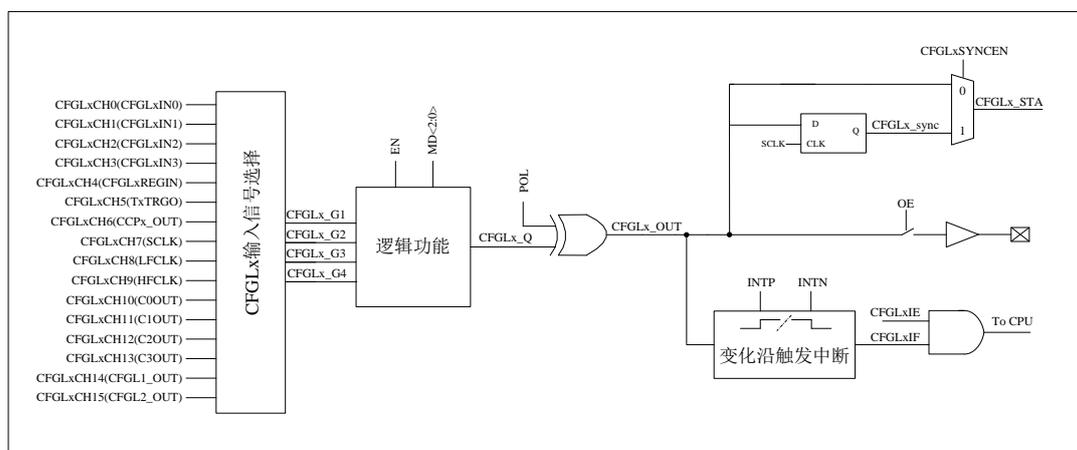


图 29-1 CFGL 模块结构框图

29.3 CFGL 相关寄存器

表 29-1 CFGL 相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	CFGL1_CTL0	R/W	CFGL1 控制寄存器 0	0x0000 0000
0x004	CFGL1_CTL1	R/W	CFGL1 控制寄存器 1	0x0000 0000
0x008	CFGL2_CTL0	R/W	CFGL2 控制寄存器 0	0x0000 0000
0x00c	CFGL2_CTL1	R/W	CFGL2 控制寄存器 1	0x0000 0000
0x010	CFGL_IFR	R/W	CFGL 中断控制寄存器	0x0000 0000

CFGL 基地址：0x4000 2980

29.3.1 CFGL1_CTL0 CFGL1 控制寄存器 0

表 29-2 CFGL1_CTL0 CFGL1 控制寄存器 0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	EN	OE	INTP	INTN	MD<2:0>			POL	G4POL	G3POL	G2POL	G1POL	D4S<2:0>			D3S<2:0>			D2S<2:0>			D1S<2:0>			REGIN		TPSEL<2:0>					

EN: 可配置逻辑单元 1 使能位

- 1 = 使能可配置逻辑单元 1，并混合输入信号
- 0 = 禁止可配置逻辑单元 1，并输出逻辑 0

OE: 可配置逻辑单元 1 输出使能位

- 1 = 使能可配置逻辑单元 1 端口引脚输出
- 0 = 禁止可配置逻辑单元 1 端口引脚输出

INTP: 可配置逻辑单元 1 上升边沿中断允许位

- 1 = CFGL1_IF 将在 CFGL1_OUT 出现上升沿时置 1
- 0 = CFGL1_IF 不会在 CFGL1_OUT 出现上升沿时置 1

INTN: 可配置逻辑单元 1 下降边沿中断允许位

- 1 = CFGL1_IF 将在 CFGL1_OUT 出现下降沿时置 1
- 0 = CFGL1_IF 不会在 CFGL1_OUT 出现下降沿时置 1

MD<2:0>: CFGL1 的可配置逻辑单元功能模式位

- 111 = 单元是带置 1 和复位功能的 1 输入透明锁存器
- 110 = 单元是带复位功能的 J-K 型单稳态触发器
- 101 = 单元是带复位功能的 2 输入 D 型单稳态触发器
- 100 = 单元是带置 1 和复位功能的 1 输入 D 型单稳态触发器
- 011 = 单元是 S-R 锁存器
- 010 = 单元是 4 输入 AND 逻辑
- 001 = 单元是 OR-XOR 逻辑
- 000 = 单元是 AND-OR 逻辑

POL: CFGL1_OUT 极性选择位

- 1 = CFGL1 的逻辑单元的输出反相
- 0 = CFGL1 的逻辑单元的输出同相

- G4POL: 门 4 输出(CFGL1_G4)极性选择位
1 = 门 4 的输出反向
0 = 门 4 的输出同相
- G3POL: 门 3 输出(CFGL1_G3)极性选择位
1 = 门 3 的输出反向
0 = 门 3 的输出同相
- G2POL: 门 2 输出(CFGL1_G2)极性选择位
1 = 门 2 的输出反向
0 = 门 2 的输出同相
- G1POL: 门 1 输出(CFGL1_G1)极性选择位
1 = 门 1 的输出反向
0 = 门 1 的输出同相
- D4S<2:0>: 门 4 输入数据通道选择位
111 = 门 4 输入选择 CFGL1CH3
110 = 门 4 输入选择 CFGL1CH2
101 = 门 4 输入选择 CFGL1CH1
100 = 门 4 输入选择 CFGL1CH0
011 = 门 4 输入选择 CFGL1CH15
010 = 门 4 输入选择 CFGL1CH14
001 = 门 4 输入选择 CFGL1CH13
000 = 门 4 输入选择 CFGL1CH12
- D3S<2:0>: 门 3 输入数据通道选择位
111 = 门 3 输入选择 CFGL1CH15
110 = 门 3 输入选择 CFGL1CH14
101 = 门 3 输入选择 CFGL1CH13
100 = 门 3 输入选择 CFGL1CH12
011 = 门 3 输入选择 CFGL1CH11
010 = 门 3 输入选择 CFGL1CH10
001 = 门 3 输入选择 CFGL1CH9
000 = 门 3 输入选择 CFGL1CH8
- D2S<2:0>: 门 2 输入数据通道选择位
111 = 门 2 输入选择 CFGL1CH11
110 = 门 2 输入选择 CFGL1CH10
101 = 门 2 输入选择 CFGL1CH9
100 = 门 2 输入选择 CFGL1CH8
011 = 门 2 输入选择 CFGL1CH7
010 = 门 2 输入选择 CFGL1CH6
001 = 门 2 输入选择 CFGL1CH5
000 = 门 2 输入选择 CFGL1CH4
- D1S<2:0>: 门 1 输入数据通道选择位
111 = 门 1 输入选择 CFGL1CH7
110 = 门 1 输入选择 CFGL1CH6
101 = 门 1 输入选择 CFGL1CH5
100 = 门 1 输入选择 CFGL1CH4

- 011 = 门 1 输入选择 CFGL1CH3
- 010 = 门 1 输入选择 CFGL1CH2
- 001 = 门 1 输入选择 CFGL1CH1
- 000 = 门 1 输入选择 CFGL1CH0

REGIN: 寄存器作为 CFGL1CH4 通道输入数据位

- 1 = CFGL1CH4 通道输入 1
- 0 = CFGL1CH4 通道输入 0

TPSEL<2:0>: CFGL1 的 CFGL1CH5 和 CFGL1CH6 通道选择位

- 111 = CFGL1CH5 为 T21TRGO, CFGL1CH6 为 CCP21CH1_OUT
- 110 = CFGL1CH5 为 T20TRGO, CFGL1CH6 为 CCP20CH1_OUT
- 101 = CFGL1CH5 为 T5TRGO, CFGL1CH6 为 ECCP5CH1_OUT
- 100 = CFGL1CH5 为 T4TRGO, CFGL1CH6 为 CCP4CH1_OUT
- 011 = CFGL1CH5 为 T3TRGO, CFGL1CH6 为 CCP3CH1_OUT
- 010 = CFGL1CH5 为 T2TRGO, CFGL1CH6 为 CCP2CH1_OUT
- 001 = CFGL1CH5 为 T1TRGO, CFGL1CH6 为 CCP1CH1_OUT
- 000 = CFGL1CH5 为 T0TRGO, CFGL1CH6 为 CCP0CH1_OUT

29. 3. 2 CFGL2_CTL0 CFGL2 控制寄存器 0

表 29-3 CFGL2_CTL0 CFGL2 控制寄存器 0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	EN	OE	INTP	INTN	MD<2:0>		POL	G4POL	G3POL	G2POL	G1POL		D4S<2:0>																			

EN: 可配置逻辑单元 2 使能位

- 1 = 使能可配置逻辑单元 2, 并混合输入信号
- 0 = 禁止可配置逻辑单元 2, 并输出逻辑 0

OE: 可配置逻辑单元 2 输出使能位

- 1 = 使能可配置逻辑单元 2 端口引脚输出
- 0 = 禁止可配置逻辑单元 2 端口引脚输出

INTP: 可配置逻辑单元 2 上升边沿中断允许位

- 1 = CFGL2_IF 将在 CFGL2_OUT 出现上升沿时置 1
- 0 = CFGL2_IF 不会在 CFGL2_OUT 出现上升沿时置 1

INTN: 可配置逻辑单元 2 下降边沿中断允许位

- 1 = CFGL2_IF 将在 CFGL2_OUT 出现下降沿时置 1
- 0 = CFGL2_IF 不会在 CFGL2_OUT 出现下降沿时置 1

MD<2:0>: CFGL2 的可配置逻辑单元功能模式位

- 111 = 单元是带置 1 和复位功能的 1 输入透明锁存器
- 110 = 单元是带复位功能的 J-K 型单稳态触发器
- 101 = 单元是带复位功能的 2 输入 D 型单稳态触发器
- 100 = 单元是带置 1 和复位功能的 1 输入 D 型单稳态触发器
- 011 = 单元是 S-R 锁存器

010 = 单元是 4 输入 AND 逻辑

001 = 单元是 OR-XOR 逻辑

000 = 单元是 AND-OR 逻辑

POL: CFGL2_OUT 极性选择位

1 = CFGL2 的逻辑单元的输出反相

0 = CFGL2 的逻辑单元的输出同相

G4POL: 门 4 输出(CFGL2_G4)极性选择位

1 = 门 4 的输出反向

0 = 门 4 的输出同相

G3POL: 门 3 输出(CFGL2_G3)极性选择位

1 = 门 3 的输出反向

0 = 门 3 的输出同相

G2POL: 门 2 输出(CFGL2_G2)极性选择位

1 = 门 2 的输出反向

0 = 门 2 的输出同相

G1POL: 门 1 输出(CFGL2_G1)极性选择位

1 = 门 1 的输出反向

0 = 门 1 的输出同相

D4S<2:0>: 门 4 输入数据通道选择位

111 = 门 4 输入选择 CFGL2CH3

110 = 门 4 输入选择 CFGL2CH2

101 = 门 4 输入选择 CFGL2CH1

100 = 门 4 输入选择 CFGL2CH0

011 = 门 4 输入选择 CFGL2CH15

010 = 门 4 输入选择 CFGL2CH14

001 = 门 4 输入选择 CFGL2CH13

000 = 门 4 输入选择 CFGL2CH12

D3S<2:0>: 门 3 输入数据通道选择位

111 = 门 3 输入选择 CFGL2CH15

110 = 门 3 输入选择 CFGL2CH14

101 = 门 3 输入选择 CFGL2CH13

100 = 门 3 输入选择 CFGL2CH12

011 = 门 3 输入选择 CFGL2CH11

010 = 门 3 输入选择 CFGL2CH10

001 = 门 3 输入选择 CFGL2CH9

000 = 门 3 输入选择 CFGL2CH8

D2S<2:0>: 门 2 输入数据通道选择位

111 = 门 2 输入选择 CFGL2CH11

110 = 门 2 输入选择 CFGL2CH10

101 = 门 2 输入选择 CFGL2CH9

100 = 门 2 输入选择 CFGL2CH8

011 = 门 2 输入选择 CFGL2CH7

010 = 门 2 输入选择 CFGL2CH6

001 = 门 2 输入选择 CFGL2CH5

000 = 门 2 输入选择 CFGL2CH4
 DIS<2:0>: 门 1 输入数据通道选择位
 111 = 门 1 输入选择 CFGL2CH7
 110 = 门 1 输入选择 CFGL2CH6
 101 = 门 1 输入选择 CFGL2CH5
 100 = 门 1 输入选择 CFGL2CH4
 011 = 门 1 输入选择 CFGL2CH3
 010 = 门 1 输入选择 CFGL2CH2
 001 = 门 1 输入选择 CFGL2CH1
 000 = 门 1 输入选择 CFGL2CH0

REGIN: 寄存器作为 CFGL2CH4 通道输入数据位

1 = CFGL2CH4 通道输入 1
 0 = CFGL2CH4 通道输入 0

TPSEL<2:0>: CFGL2 的 CFGL2CH5 和 CFGL2CH6 通道选择位

111 = CFGL2CH5 为 T21TRGO, CFGL2CH6 为 CCP21CH1_OUT
 110 = CFGL2CH5 为 T20TRGO, CFGL2CH6 为 CCP20CH1_OUT
 101 = CFGL2CH5 为 T5TRGO, CFGL2CH6 为 ECCP5CH1_OUT
 100 = CFGL2CH5 为 T4TRGO, CFGL2CH6 为 CCP4CH1_OUT
 011 = CFGL2CH5 为 T3TRGO, CFGL2CH6 为 CCP3CH1_OUT
 010 = CFGL2CH5 为 T2TRGO, CFGL2CH6 为 CCP2CH1_OUT
 001 = CFGL2CH5 为 T1TRGO, CFGL2CH6 为 CCP1CH1_OUT
 000 = CFGL2CH5 为 T0TRGO, CFGL2CH6 为 CCP0CH1_OUT

29. 3. 3 CFGL1_CTL1 CFGL1 控制寄存器 1

表 29-4 CFGL1_CTL1 CFGL1 控制寄存器 1

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N

G4D4T: CFGL1 门 4 数据 4 同向允许位

1 = CFGL1_D4T 允许通过门输入到 CFGL1_G4
 0 = CFGL1_D4T 禁止通过门输入到 CFGL1_G4

G4D4N: CFGL1 门 4 数据 4 反向允许位

1 = CFGL1_D4N 允许通过门输入到 CFGL1_G4
 0 = CFGL1_D4N 禁止通过门输入到 CFGL1_G4

G4D3T: CFGL1 门 4 数据 3 同向允许位

1 = CFGL1_D3T 允许通过门输入到 CFGL1_G4
 0 = CFGL1_D3T 禁止通过门输入到 CFGL1_G4

G4D3N: CFGL1 门 4 数据 3 反向允许位

1 = CFGL1_D3N 允许通过门输入到 CFGL1_G4
 0 = CFGL1_D3N 禁止通过门输入到 CFGL1_G4

- G4D2T: CFGL1 门 4 数据 2 同向允许位
1 = CFGL1_D2T 允许通过门输入到 CFGL1_G4
0 = CFGL1_D2T 禁止通过门输入到 CFGL1_G4
- G4D2N: CFGL1 门 4 数据 2 反向允许位
1 = CFGL1_D2N 允许通过门输入到 CFGL1_G4
0 = CFGL1_D2N 禁止通过门输入到 CFGL1_G4
- G4D1T: CFGL1 门 4 数据 1 同向允许位
1 = CFGL1_D1T 允许通过门输入到 CFGL1_G4
0 = CFGL1_D1T 禁止通过门输入到 CFGL1_G4
- G4D1N: CFGL1 门 4 数据 1 反向允许位
1 = CFGL1_D1N 允许通过门输入到 CFGL1_G4
0 = CFGL1_D1N 禁止通过门输入到 CFGL1_G4
- G3D4T: CFGL1 门 3 数据 4 同向允许位
1 = CFGL1_D4T 允许通过门输入到 CFGL1_G3
0 = CFGL1_D4T 禁止通过门输入到 CFGL1_G3
- G3D4N: CFGL1 门 3 数据 4 反向允许位
1 = CFGL1_D4N 允许通过门输入到 CFGL1_G3
0 = CFGL1_D4N 禁止通过门输入到 CFGL1_G3
- G3D3T: CFGL1 门 3 数据 3 同向允许位
1 = CFGL1_D3T 允许通过门输入到 CFGL1_G3
0 = CFGL1_D3T 禁止通过门输入到 CFGL1_G3
- G3D3N: CFGL1 门 3 数据 3 反向允许位
1 = CFGL1_D3N 允许通过门输入到 CFGL1_G3
0 = CFGL1_D3N 禁止通过门输入到 CFGL1_G3
- G3D2T: CFGL1 门 3 数据 2 同向允许位
1 = CFGL1_D2T 允许通过门输入到 CFGL1_G3
0 = CFGL1_D2T 禁止通过门输入到 CFGL1_G3
- G3D2N: CFGL1 门 3 数据 2 反向允许位
1 = CFGL1_D2N 允许通过门输入到 CFGL1_G3
0 = CFGL1_D2N 禁止通过门输入到 CFGL1_G3
- G3D1T: CFGL1 门 3 数据 1 同向允许位
1 = CFGL1_D1T 允许通过门输入到 CFGL1_G3
0 = CFGL1_D1T 禁止通过门输入到 CFGL1_G3
- G3D1N: CFGL1 门 3 数据 1 反向允许位
1 = CFGL1_D1N 允许通过门输入到 CFGL1_G3
0 = CFGL1_D1N 禁止通过门输入到 CFGL1_G3
- G2D4T: CFGL1 门 2 数据 4 同向允许位
1 = CFGL1_D4T 允许通过门输入到 CFGL1_G2
0 = CFGL1_D4T 禁止通过门输入到 CFGL1_G2
- G2D4N: CFGL1 门 2 数据 4 反向允许位
1 = CFGL1_D4N 允许通过门输入到 CFGL1_G2
0 = CFGL1_D4N 禁止通过门输入到 CFGL1_G2
- G2D3T: CFGL1 门 2 数据 3 同向允许位
1 = CFGL1_D3T 允许通过门输入到 CFGL1_G2

0 = CFGL1_D3T 禁止通过门输入到 CFGL1_G2
G2D3N: CFGL1 门 2 数据 3 反向允许位
1 = CFGL1_D3N 允许通过门输入到 CFGL1_G2
0 = CFGL1_D3N 禁止通过门输入到 CFGL1_G2
G2D2T: CFGL1 门 2 数据 2 同向允许位
1 = CFGL1_D2T 允许通过门输入到 CFGL1_G2
0 = CFGL1_D2T 禁止通过门输入到 CFGL1_G2
G2D2N: CFGL1 门 2 数据 2 反向允许位
1 = CFGL1_D2N 允许通过门输入到 CFGL1_G2
0 = CFGL1_D2N 禁止通过门输入到 CFGL1_G2
G2D1T: CFGL1 门 2 数据 1 同向允许位
1 = CFGL1_D1T 允许通过门输入到 CFGL1_G2
0 = CFGL1_D1T 禁止通过门输入到 CFGL1_G2
G2D1N: CFGL1 门 2 数据 1 反向允许位
1 = CFGL1_D1N 允许通过门输入到 CFGL1_G2
0 = CFGL1_D1N 禁止通过门输入到 CFGL1_G2
G1D4T: CFGL1 门 1 数据 4 同向允许位
1 = CFGL1_D4T 允许通过门输入到 CFGL1_G1
0 = CFGL1_D4T 禁止通过门输入到 CFGL1_G1
G1D4N: CFGL1 门 1 数据 4 反向允许位
1 = CFGL1_D4N 允许通过门输入到 CFGL1_G1
0 = CFGL1_D4N 禁止通过门输入到 CFGL1_G1
G1D3T: CFGL1 门 1 数据 3 同向允许位
1 = CFGL1_D3T 允许通过门输入到 CFGL1_G1
0 = CFGL1_D3T 禁止通过门输入到 CFGL1_G1
G1D3N: CFGL1 门 1 数据 3 反向允许位
1 = CFGL1_D3N 允许通过门输入到 CFGL1_G1
0 = CFGL1_D3N 禁止通过门输入到 CFGL1_G1
G1D2T: CFGL1 门 1 数据 2 同向允许位
1 = CFGL1_D2T 允许通过门输入到 CFGL1_G1
0 = CFGL1_D2T 禁止通过门输入到 CFGL1_G1
G1D2N: CFGL1 门 1 数据 2 反向允许位
1 = CFGL1_D2N 允许通过门输入到 CFGL1_G1
0 = CFGL1_D2N 禁止通过门输入到 CFGL1_G1
G1D1T: CFGL1 门 1 数据 1 同向允许位
1 = CFGL1_D1T 允许通过门输入到 CFGL1_G1
0 = CFGL1_D1T 禁止通过门输入到 CFGL1_G1
G1D1N: CFGL1 门 1 数据 1 反向允许位
1 = CFGL1_D1N 允许通过门输入到 CFGL1_G1
0 = CFGL1_D1N 禁止通过门输入到 CFGL1_G1

29. 3. 4 CFGL2_CTL1 CFGL2 控制寄存器 1

表 29-5 CFGL2_CTL1 CFGL2 控制寄存器 1

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N

G4D4T: CFGL2 门 4 数据 4 同向允许位

1 = CFGL2_D4T 允许通过门输入到 CFGL2_G4

0 = CFGL2_D4T 禁止通过门输入到 CFGL2_G4

G4D4N: CFGL2 门 4 数据 4 反向允许位

1 = CFGL2_D4N 允许通过门输入到 CFGL2_G4

0 = CFGL2_D4N 禁止通过门输入到 CFGL2_G4

G4D3T: CFGL2 门 4 数据 3 同向允许位

1 = CFGL2_D3T 允许通过门输入到 CFGL2_G4

0 = CFGL2_D3T 禁止通过门输入到 CFGL2_G4

G4D3N: CFGL2 门 4 数据 3 反向允许位

1 = CFGL2_D3N 允许通过门输入到 CFGL2_G4

0 = CFGL2_D3N 禁止通过门输入到 CFGL2_G4

G4D2T: CFGL2 门 4 数据 2 同向允许位

1 = CFGL2_D2T 允许通过门输入到 CFGL2_G4

0 = CFGL2_D2T 禁止通过门输入到 CFGL2_G4

G4D2N: CFGL2 门 4 数据 2 反向允许位

1 = CFGL2_D2N 允许通过门输入到 CFGL2_G4

0 = CFGL2_D2N 禁止通过门输入到 CFGL2_G4

G4D1T: CFGL2 门 4 数据 1 同向允许位

1 = CFGL2_D1T 允许通过门输入到 CFGL2_G4

0 = CFGL2_D1T 禁止通过门输入到 CFGL2_G4

G4D1N: CFGL2 门 4 数据 1 反向允许位

1 = CFGL2_D1N 允许通过门输入到 CFGL2_G4

0 = CFGL2_D1N 禁止通过门输入到 CFGL2_G4

G3D4T: CFGL2 门 3 数据 4 同向允许位

1 = CFGL2_D4T 允许通过门输入到 CFGL2_G3

0 = CFGL2_D4T 禁止通过门输入到 CFGL2_G3

G3D4N: CFGL2 门 3 数据 4 反向允许位

1 = CFGL2_D4N 允许通过门输入到 CFGL2_G3

0 = CFGL2_D4N 禁止通过门输入到 CFGL2_G3

G3D3T: CFGL2 门 3 数据 3 同向允许位

1 = CFGL2_D3T 允许通过门输入到 CFGL2_G3

0 = CFGL2_D3T 禁止通过门输入到 CFGL2_G3

G3D3N: CFGL2 门 3 数据 3 反向允许位

1 = CFGL2_D3N 允许通过门输入到 CFGL2_G3

0 = CFGL2_D3N 禁止通过门输入到 CFGL2_G3
G3D2T: CFGL2 门 3 数据 2 同向允许位
1 = CFGL2_D2T 允许通过门输入到 CFGL2_G3
0 = CFGL2_D2T 禁止通过门输入到 CFGL2_G3
G3D2N: CFGL2 门 3 数据 2 反向允许位
1 = CFGL2_D2N 允许通过门输入到 CFGL2_G3
0 = CFGL2_D2N 禁止通过门输入到 CFGL2_G3
G3D1T: CFGL2 门 3 数据 1 同向允许位
1 = CFGL2_D1T 允许通过门输入到 CFGL2_G3
0 = CFGL2_D1T 禁止通过门输入到 CFGL2_G3
G3D1N: CFGL2 门 3 数据 1 反向允许位
1 = CFGL2_D1N 允许通过门输入到 CFGL2_G3
0 = CFGL2_D1N 禁止通过门输入到 CFGL2_G3
G2D4T: CFGL2 门 2 数据 4 同向允许位
1 = CFGL2_D4T 允许通过门输入到 CFGL2_G2
0 = CFGL2_D4T 禁止通过门输入到 CFGL2_G2
G2D4N: CFGL2 门 2 数据 4 反向允许位
1 = CFGL2_D4N 允许通过门输入到 CFGL2_G2
0 = CFGL2_D4N 禁止通过门输入到 CFGL2_G2
G2D3T: CFGL2 门 2 数据 3 同向允许位
1 = CFGL2_D3T 允许通过门输入到 CFGL2_G2
0 = CFGL2_D3T 禁止通过门输入到 CFGL2_G2
G2D3N: CFGL2 门 2 数据 3 反向允许位
1 = CFGL2_D3N 允许通过门输入到 CFGL2_G2
0 = CFGL2_D3N 禁止通过门输入到 CFGL2_G2
G2D2T: CFGL2 门 2 数据 2 同向允许位
1 = CFGL2_D2T 允许通过门输入到 CFGL2_G2
0 = CFGL2_D2T 禁止通过门输入到 CFGL2_G2
G2D2N: CFGL2 门 2 数据 2 反向允许位
1 = CFGL2_D2N 允许通过门输入到 CFGL2_G2
0 = CFGL2_D2N 禁止通过门输入到 CFGL2_G2
G2D1T: CFGL2 门 2 数据 1 同向允许位
1 = CFGL2_D1T 允许通过门输入到 CFGL2_G2
0 = CFGL2_D1T 禁止通过门输入到 CFGL2_G2
G2D1N: CFGL2 门 2 数据 1 反向允许位
1 = CFGL2_D1N 允许通过门输入到 CFGL2_G2
0 = CFGL2_D1N 禁止通过门输入到 CFGL2_G2
G1D4T: CFGL2 门 1 数据 4 同向允许位
1 = CFGL2_D4T 允许通过门输入到 CFGL2_G1
0 = CFGL2_D4T 禁止通过门输入到 CFGL2_G1
G1D4N: CFGL2 门 1 数据 4 反向允许位
1 = CFGL2_D4N 允许通过门输入到 CFGL2_G1
0 = CFGL2_D4N 禁止通过门输入到 CFGL2_G1
G1D3T: CFGL2 门 1 数据 3 同向允许位

- 1 = CFGL2_D3T 允许通过门输入到 CFGL2_G1
 0 = CFGL2_D3T 禁止通过门输入到 CFGL2_G1
- G1D3N: CFGL2 门 1 数据 3 反向允许位
 1 = CFGL2_D3N 允许通过门输入到 CFGL2_G1
 0 = CFGL2_D3N 禁止通过门输入到 CFGL2_G1
- G1D2T: CFGL2 门 1 数据 2 同向允许位
 1 = CFGL2_D2T 允许通过门输入到 CFGL2_G1
 0 = CFGL2_D2T 禁止通过门输入到 CFGL2_G1
- G1D2N: CFGL2 门 1 数据 2 反向允许位
 1 = CFGL2_D2N 允许通过门输入到 CFGL2_G1
 0 = CFGL2_D2N 禁止通过门输入到 CFGL2_G1
- G1D1T: CFGL2 门 1 数据 1 同向允许位
 1 = CFGL2_D1T 允许通过门输入到 CFGL2_G1
 0 = CFGL2_D1T 禁止通过门输入到 CFGL2_G1
- G1D1N: CFGL2 门 1 数据 1 反向允许位
 1 = CFGL2_D1N 允许通过门输入到 CFGL2_G1
 0 = CFGL2_D1N 禁止通过门输入到 CFGL2_G1

29. 3. 5 CFGLIFR CFGL 中断控制寄存器

表 29-6 CFGLIFR CFGL 中断控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W																	R	R					R/W	R	R	R		
位名	CFGL2SYNCEN	CFGL1SYNCEN	CFGL2IE	CFGL1IE																	CFGL2IF	CFGL1IF					CFGLIC	CFGL2OUT	CFGL1OUT			

- CFGL2SYNCEN: CFGL2 输出同步使能位
 1 = CFGL2_OUT 被 SCLK 同步
 0 = CFGL2_OUT 未被 SCLK 同步
- CFGL1SYNCEN: CFGL1 输出同步使能位
 1 = CFGL1_OUT 被 SCLK 同步
 0 = CFGL1_OUT 未被 SCLK 同步
- CFGL2IE: CFGL2 中断使能位
 1 = 使能 CFGL2 中断
 0 = 未使能 CFGL2 中断
- CFGL1IE: CFGL1 中断使能位
 1 = 使能 CFGL1 中断
 0 = 未使能 CFGL1 中断
- CFGL2IF: CFGL2 中断标志位
 1 = CFGL2_OUT 发生翻转
 0 = CFGL2_OUT 未发生翻转
- CFGL1IF: CFGL1 中断标志位
 1 = CFGL1_OUT 发生翻转

0 = CFGL1_OUT 未发生翻转

CFGLIC: CFGL 中断标志清零位

1 = 将 CFGL1IF/CFGL2IF 标志位清 0

0 = 无作用

注:该位需要手动清 0

CFGL2OUT: CFGL2 数据状态位

CFGL1OUT: CFGL1 数据状态位

29.4 CFGL 模块的配置

CFGLx 模块的编程通过配置逻辑信号流中的 4 个阶段来实现。这 4 个阶段是：

- 数据选择
- 数据门控
- 逻辑功能选择
- 输出极性

每个阶段都可在运行时通过写入相应的 CFGLx 特殊功能寄存器来进行设置。这具有支持在程序执行期间即时执行逻辑重新配置的额外优点。

29.4.1 CFGL 数据输入信号

有 16 个信号可用作可配置逻辑的输入。使用 4 个 8 输入多路开关来选择要传递到下一阶段的输入。多路开关的 16 个输入以 4 个为一组进行分组。每组可供 4 个多路开关中的两个使用，在每种情形中，每组都与另一个组进行配对。这种分组安排使这些多路开关最多可以选择一个组两次，又不排除对另一个组的选择。

数据输入通过 CFGLx_CTL0 寄存器中的 DyS<2:0>位进行配置选择(x=1,2; y=1,2,3,4)。

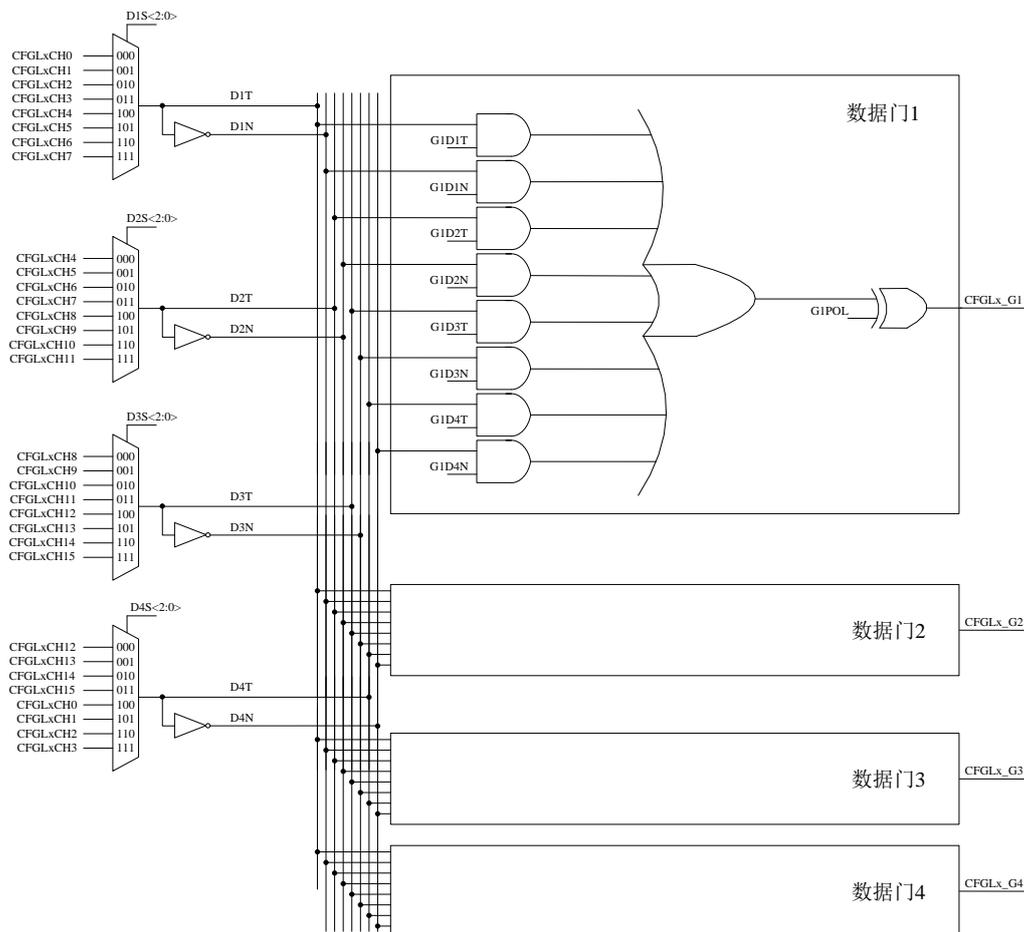


图 29-2 CFGL 数据输入结构框图

CFGLx 数据输入结构框图中的输入通道以顺序编码表示 (CFGLxCH<15:0>), 其对应输入源如下表所示。

表 29-7 CFGL 数据输入通道对应表

CFGL1	CFGL2	数据输入	CFGLx_D1S	CFGLx_D2S	CFGLx_D3S	CFGLx_D4S
CFGL1_IN0	CFGL2_IN0	CFGLxCH0	000			100
CFGL1_IN1	CFGL2_IN1	CFGLxCH1	001			101
CFGL1_IN2	CFGL2_IN2	CFGLxCH2	010			110
CFGL1_IN3	CFGL2_IN3	CFGLxCH3	011			111
CFGL1_REGIN	CFGL2_REGIN	CFGLxCH4	100	000		
TxTRGO	TxTRGO	CFGLxCH5	101	001		
CCPxCH1_OUT	CCPxCH1_OUT	CFGLxCH6	110	010		
SCLK	SCLK	CFGLxCH7	111	011		
LFCLK	LFCLK	CFGLxCH8		100	000	
HFCLK	HFCLK	CFGLxCH9		101	001	
C0OUT	C0OUT	CFGLxCH10		110	010	
C1OUT	C1OUT	CFGLxCH11		111	011	
C2OUT	C2OUT	CFGLxCH12			100	000
C3OUT	C3OUT	CFGLxCH13			101	001
CFGL1_OUT	CFGL1_OUT	CFGLxCH14			110	010
CFGL2_OUT	CFGL2_OUT	CFGLxCH15			111	011

29.4.2 CFGL 数据门控

来自输入多路开关的输出将通过数据门控阶段转送到所需的逻辑功能输入。每个数据门可以转送由 4 个选定输入组成的任意组合。门阶段不仅仅是信号方向。可将门配置为将每个输入信号指定为反相或同相数据。在每个门中，将定向后的信号进行与运算。每个门的输出可以先进行反相，然后再进入逻辑功能阶段。

门控实际上是一个 1 至 4 的输入 AND/NAND/OR/NOR 门。如果将每个输入和输出进行反相，则该门的作用是对所有已使能数据输入进行或运算。如果输入和输出不进行反相，则该门的作用是对所有已使能输入进行与运算。

用户可以（但建议不要）同时选择同一输入的正负值。如果这么做，则无论其他输入如何，门的输出都将为 0，但可能会出现逻辑故障（瞬态电流引起的脉冲）。如果通道的输出必须为 0 或 1，则建议的方法是将所有门位设置为 0，并使用门极性位来设置所需的电平。

数据门控使用 CFGLx_CTL1 寄存器进行配置。图 29-2 右侧给出了数据门控的图示。其中仅详细说明了一个门。其余三个门使用相同的配置，只是数据使能对应于该门的使能信号。

29.4.3 CFGL 逻辑功能

有 8 种可用的逻辑功能，包括：

- AND-OR
- OR-XOR
- AND
- S-R 锁存器
- 带置 1 和复位功能的 D 型单稳态触发器
- 带复位功能的 D 型单稳态触发器
- 带复位功能的 J-K 型单稳态触发器
- 带置 1 和复位功能的透明锁存器

每种逻辑功能具有 4 个输入和 1 个输出。4 个输入是上一阶段的 4 个数据门输出。输出送到反相阶段，接着送到其他外设、输出引脚，然后回到 CFGLx。

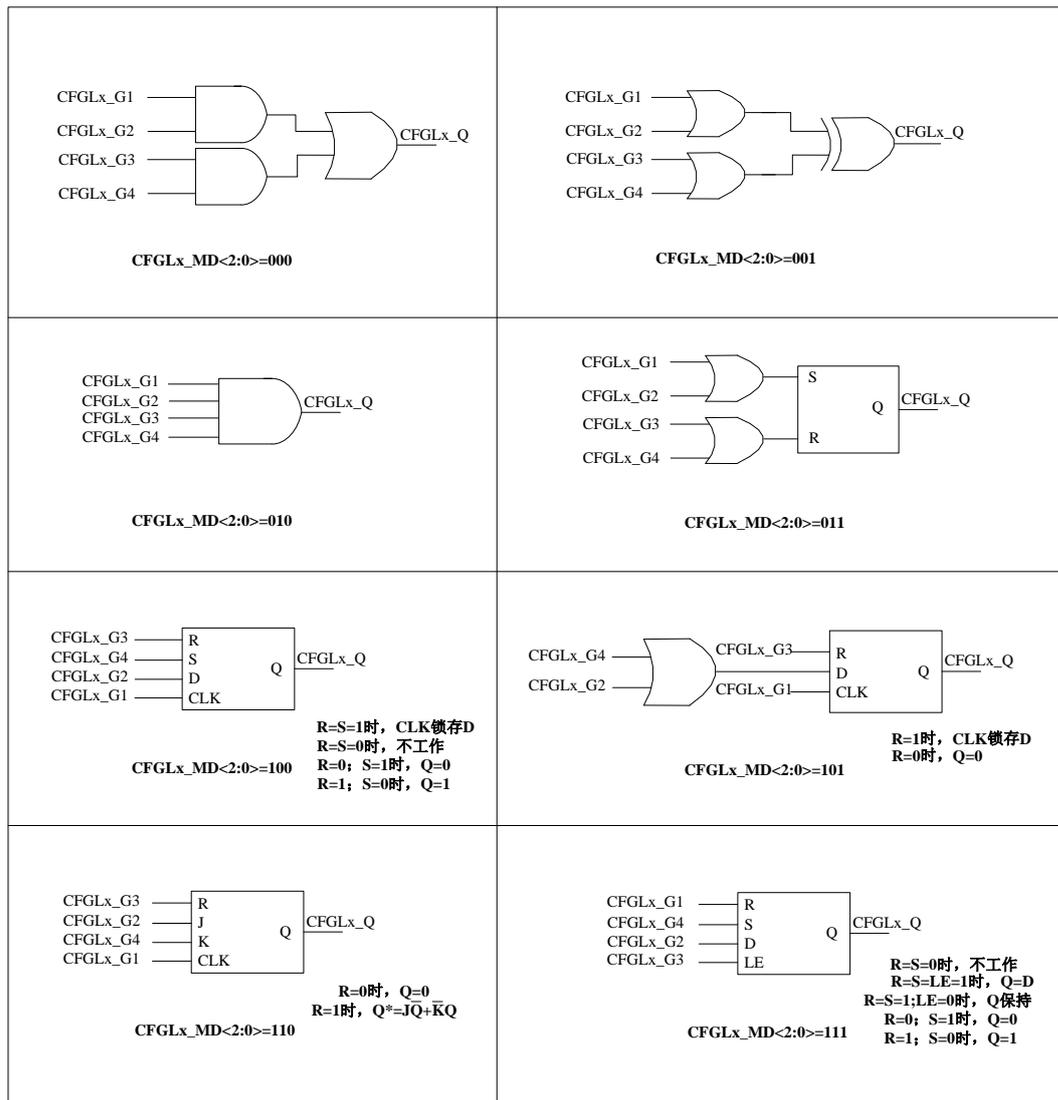


图 29-3 CFGL 逻辑功能结构框图

29.4.4 CFGL 输出同步

CFGLIFR 寄存器的 CFGLxSYNCEN 位可以选择是否将 CFGLx 的输出 CFGLxOUT 被 SCLK 同步。当 CFGLxSYNCEN=1，CFGLxOUT 将被同步至 SCLK 的上升沿。

29.4.5 CFGL 输出极性

可配置逻辑单元中的最后一个阶段是输出极性。将 CFGLxCTL0 寄存器的 POL 位置 1 时，来自逻辑阶段的输出信号会进行反相。如果在允许中断时改变极性会导致中断结果输出转换。

29.5 CFGL 中断

CFGL 中断内置 1 个边沿检测器，通过 CFGLxCTL0 寄存器的 INTP 和 INTN 进行配置。当 INTP=1 时，CFGLxOUT 的上升沿将产生一个 CFGL 中断，CFGLxIF 位置 1；当 INTN=1

时，CFGLxOUT 的下降沿将产生一个 CFGL 中断，CFGLxIF 位置 1；如果 INTP 和 INTN 位同时置 1，CFGLxOUT 的上升沿和下降沿都将会产生 CFGL 中断。

如果 CFGLxSYNCEN 位置 1，将使 CFGLxOUT 被 SCLK 同步，中断也将会在同步后产生。

通过将 CFGLIFR 寄存器的 CFGLIC 位置 1，可以清 0 中断标志位 CFGLxIF。在清 0 完中断标志位后，CFGLIC 需要手动清 0。

29.6 CFGL 输出

CFGLx 的输出可以输出至 IO 口，也可以返回作为 CFGLx 的输入源。将 CFGLx 输出至 IO 口时，需要将 CFGLxCTL0 寄存器的 OE 位置 1 使能输出逻辑。

CFGLIFR 寄存器的 CFGLxOUT 位为 CFGLx 结果的状态位，可以通过读取该位得到 CFGLx 当前数据。CFGLxOUT 受 CFGLxSYNCEN 位影响，当 CFGLxSYNCEN=1 时，CFGLxOUT 将被同步至 SCLK 的上升沿。

29.7 CFGL 工作在休眠模式

CFGL 模块可以在普通休眠模式下保持工作状态；当 CFGLxSYNCEN=0 时，CFGL 模块可在深度休眠模式下保持运行。

30 复位 (RESET)

系统复位源:

- POR 上电复位
- BOR 复位
- NRST 外部复位引脚复位
- 窗口看门狗复位
- 独立看门狗复位
- 软件复位

单片机具有: POR 上电复位、BOR 复位、NRST 复位、IWDT 复位、WWDT 复位、软件复位六种复位方式。

除复位方式以外,单片机还提供一个可编程的电压检测模块(PVD),对供电电源 VDD 的电压进行检测。

有些寄存器的状态在任何复位条件下都不会受到影响,其它大多数寄存器在复位事件发生时将被复位成“复位状态”。下图给出了片内复位电路的简化结构方框图。

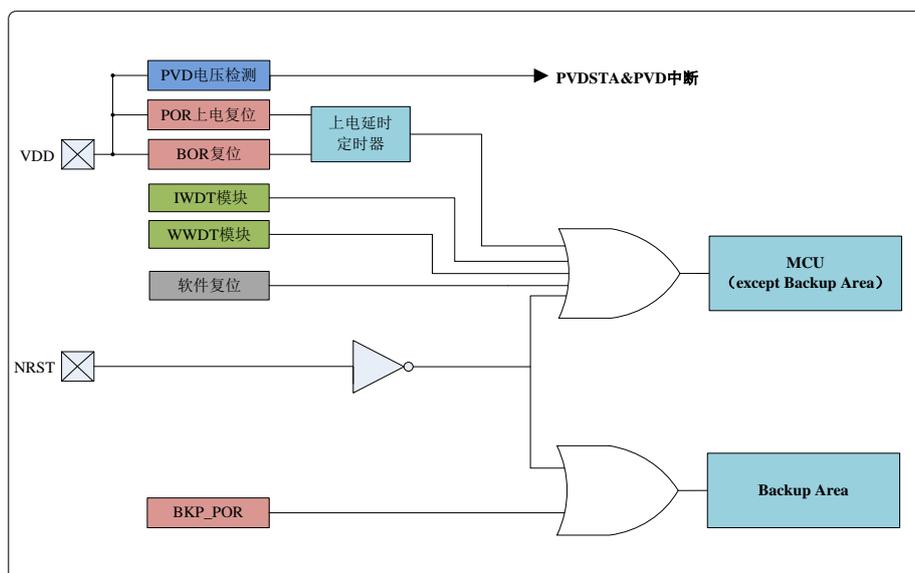


图 30-1 片内复位电路简化框图

注: 上电延时定时器只对 POR/BOR 复位有效

30.1 复位相关寄存器

表 30-1 复位相关的寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	RST_CTL0	R/W	复位控制寄存器 0	0x0000 0000
0x004	RST_CTL1	R/W	复位控制寄存器 1	0xFFFF FFFF
0x008	RST_CTL2	R/W	复位控制寄存器 2	0xFFFF FFFF
0x00C	RST_CTL3	R/W	复位控制寄存器 3	0xFFFF FFFF

基地址：0x4000 1500

30.1.1 RST_CTL0 复位控制寄存器 0

表 30-2 RST_CTL0 复位控制寄存器 0

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W																									R/W							
位名																									GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST

xRST: x 复位

0 = 无作用

1 = 复位

30.1.2 RST_CTL1 复位控制寄存器 1

表 30-3 RST_CTL1 复位控制寄存器 1

复位值	1 31	1 30	1 29	1 28	1 27	1 26	1 25	1 24	1 23	1 22	1 21	1 20	1 19	1 18	1 17	1 16	1 15	1 14	1 13	1 12	1 11	1 10	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0
R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	
位名	USERST			I2C2RST	I2C1RST	I2C0RST	SPI1RST	SPI0RST	USART4RST	USART3RST	USART2RST	USART1RST	USART0RST		TORST	CMPRST	DAC1RST	DAC0RST	ADC2RST	ADC1RST	ADC0RST	QE1ORST		T9T10RST		T5T6RST	T4RST	T3RST	T2RST	T1RST	QE11RST	

xRST: x 复位

0 = 无作用

1 = 复位

30. 1. 3 RST_CTL2 复位控制寄存器 2

表 30-4 RST_CTL2 复位控制寄存器 2

复位值	1 31	1 30	1 29	1 28	1 27	1 26	1 25	1 24	1 23	1 22	1 21	1 20	1 19	1 18	1 17	1 16	1 15	1 14	1 13	1 12	1 11	1 10	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	
R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W				R/W							R/W		R/W	R/W			
位名	USART7RST	USART6RST	USART5RST			T23RST	T22RST	T21RST	T20RST	T19RST	T18RST			CAN2RST	T15RST	T14RST			DMA1RST		DMA0RST						WWDTRST			CAN1RST	CAN0RST		

xRST: x 复位

0 = 无作用

1 = 复位

注：备份域中外设复位描述在电源结构以及备份域章节。

30. 1. 4 RST_CTL3 复位控制寄存器 3

表 30-5 RST_CTL3 复位控制寄存器 3

复位值	1 31	1 30	1 29	1 28	1 27	1 26	1 25	1 24	1 23	1 22	1 21	1 20	1 19	1 18	1 17	1 16	1 15	1 14	1 13	1 12	1 11	1 10	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0
R/W													R/W			R/W			R/W						R/W					R/W	R/W	
位名													CFGLRST			EXICRST			CRCRST						I2C3RST					SPI3RST	SPI2RST	

xRST: x 复位

0 = 无作用

1 = 复位

30.2 上电/掉电复位 POR/PDR

当 VDD/VDDA 高于 VPOR 值，系统退出复位状态；当 VDD/VDDA 低于 VPDR 值，系统进入复位状态。

- 芯片在 1.7V 以下保持复位
- 无需外部复位电路
- 回滞典型值 40mV

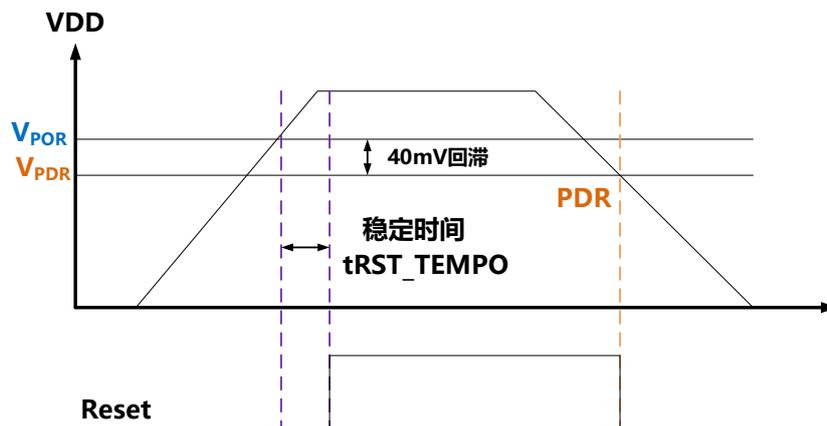


图 30-2 POR/PDR 电压说明简图

30.3 BrownOut 复位 BOR

使能 BOR 时，当 VDD/VDDA 高于 VBORH 值，系统退出复位状态；当 VDD/VDDA 低于 VBORL 值，系统进入复位状态。

- 芯片在 VBOR 以下保持复位
- 回滞典型值 250mV 左右
- BOR 值可由 PM_CTL0 寄存器的 BORSEL<1:0>位进行配置

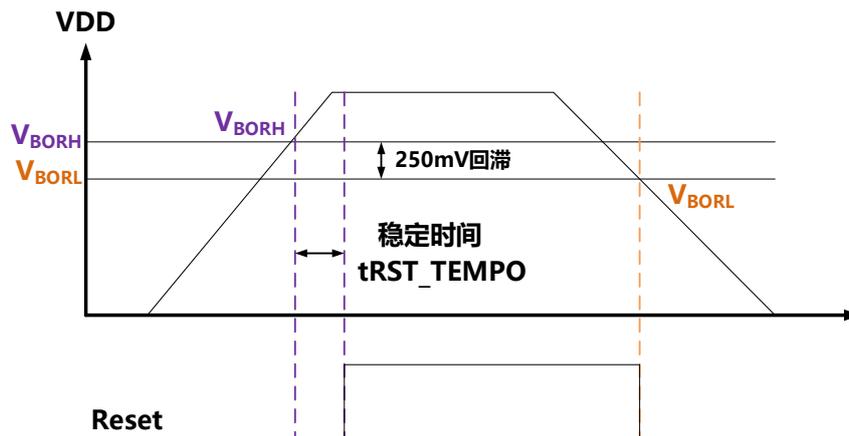


图 30-3 BOR 电压说明简图

30.4 IWDG 复位

使能独立看门狗后，当看门狗计数器计满后产生溢出，将使单片机复位。

在休眠模式下，IWDG 也可以正常工作，当 IWDG 定时器计满溢出后，将会使单片机从

休眠模式唤醒转入正常工作模式，在休眠模式不会对备份域寄存器复位。

30.5 WWDT 复位

在使能窗口看门狗后，没有在指定的时间窗口内喂狗时会产生系统复位，复位单片机。

30.6 NRST 复位

NRST 引脚上电默认为复位引脚功能，因此，请确保芯片在上电时，NRST 引脚无任何下拉到 VSS 通路，否则将引起芯片复位。

芯片上电正常工作后，可以通过配置 PM_CTL0 寄存器的 NRSTEN 位，将 NRST 引脚的外部复位功能关闭，此时，该引脚可作为通用 GPIO 引脚使用。

当外部引脚 NRST 输入复位信号，不管单片机工作在正常模式还是休眠模式，均会使单片机（包括备份域在内）复位。下图是建议 NRST 复位电路。

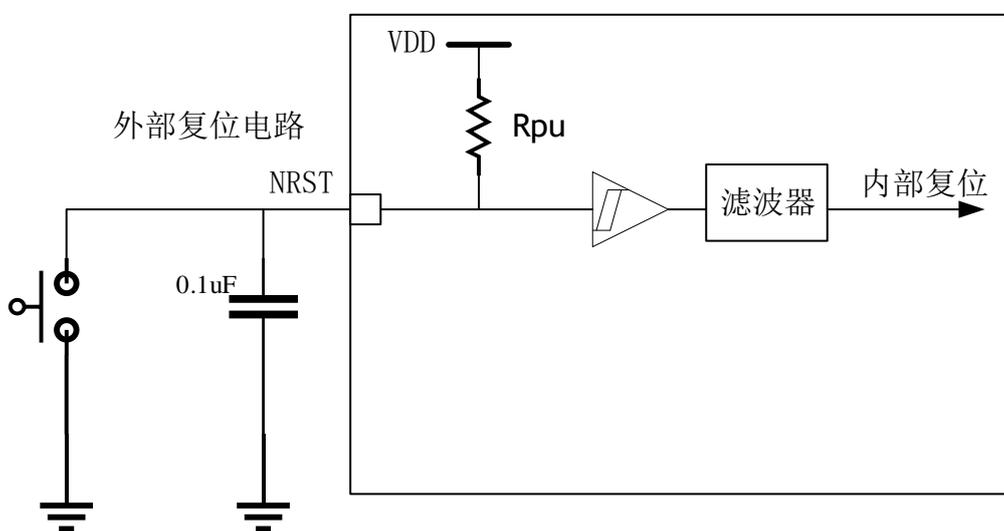


图 30-4 建议 NRST 复位电路

注：Rpu 的阻值约为 10KΩ；当 PM_CTL0 寄存器的 NRSTEN 位置 1 时，NRST 引脚的外部复位功能被关闭，同时 Rpu 电阻被断开。

30.7 电压检测模块（PVD）

单片机提供一个可编程的电压检测模块（PVD），对供电电源 VDD 的电压进行检测。PVD 模块相关寄存器为位于备份域中的 PM_CTL1 寄存器和 PM_STA1 寄存器。

- 监控 VDD/VDDA 电压
- 回滞典型值 250mV

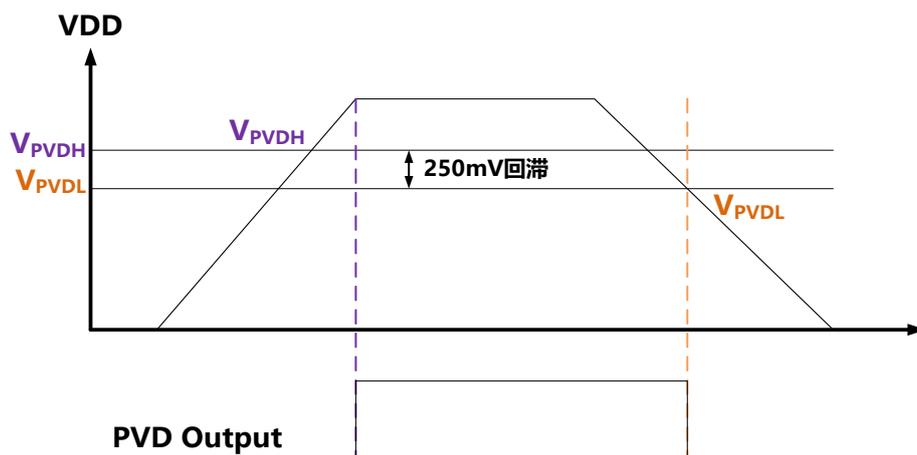


图 30-5 电压检测说明简图

通过将 PM_CTL1 寄存器的 PVD_EN 位置 1 使能电压检测模块；PVDVS<2:0>位可选择电压检测点。PM_STA1 寄存器的 PVDSTA 位实时更新 PVD 的检测结果，当 PVDSTA=0，VDD 电压高于 PVD 检测电压点；当 PVDSTA=1，VDD 电压低于 PVD 检测电压点。

PVD 检测结果可触发中断，PVD 使用外部中断线 16(EINT16)中断，将 INT_EINTMASK 寄存器的 EINTM16 位置 1 将使能 PVD 中断；PVD_EN=1，当 VDD 电压低于 PVD 检测电压点时，PVDSTA=1，当 VDD 电压高于 PVD 检测电压时，PVDSTA=0，通过 INT_EINTRISE/INT_EINTFALL 可以选择上升或下降沿触发 PVD 中断。

当 PVD_EN=0 即 PVD 模块不使能时，PVDSTA=0，PVD_EN 置 1 时可能会立即将 PVD 中断 EINT16 位置 1，启动 PVD 模块后，请对 EINT16 位进行清零操作。

30.8 备份域复位

备份域的复位源有两个：备份域专用 POR (BKP_POR) 和外部复位引脚 NRST。

备份域内的 RTC 模块和备份域数据寄存器组还配备软件复位；PM_CTL0 寄存器的 BKPREGCLR 位用于复位 RTC 模块和备份域数据寄存器组。

31 外设模块时钟使能模块(CLK_EN)

31.1 软件复位

软件复位有三种实现形式：

- RESET 指令复位

单片机自带系统复位指令 RESET，当使用 RESET 指令时，当 RESET 指令前面的指令执行完成后，就会进入系统复位操作。

- SYSTEMRST 位置 1 复位

可以通过 SYS_ARCTL 应用和复位控制寄存器的 SYSTEMRST 位置 1 实现发出一个系统复位操作。为了防止误操作，对 SYS_ARCTL 应用和复位控制寄存器执行写操作时，需要将 0x05FA 写入 ARKEY<31:16>中，否则写操作被忽略。

- 外设复位

通过 RST_CTLx(x=0,1,2,3)复位控制寄存器对相应的外设进行软件复位。默认复位打开，清零后才能正常使用相应的外设。需要注意的是，定时器和相应的 CCP/ECCP 为同一个软件清零。

31.2 概述

为了降低功耗，默认外设时钟就禁止。在使用外设模块时，需要使能该外设模块时钟控制信号，否则模块不工作。通过 PCLK_CTLx (x=0,1,2,3) 外设时钟控制寄存器控制相应的外设时钟。当外设时钟禁止时，CPU 无法对相应的模块寄存器进行写操作。

31.3 外设模块时钟使能模块相关寄存器

表 31-1 外设模块时钟使能模块相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	保留	R/W	振荡器控制寄存器 0	保留
0x040	PCLK_CTL0	R/W	外设时钟控制寄存器 0	0xFFFFFFFF
0x044	PCLK_CTL1	R/W	外设时钟控制寄存器 1	0x0000 0000
0x048	PCLK_CTL2	R/W	外设时钟控制寄存器 2	0x0000 0000
0x04C	PCLK_CTL3	R/W	外设时钟控制寄存器 3	0x0000 0000

基地址: 0x4000 2600

31.3.1 PCLK_CTL0 外设时钟控制寄存器 0

表 31-2 PCLK_CTL0 外设时钟控制寄存器 0

	131	130	129	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112	111	110	109	108	107	106	105	104	103	102	101	100																					
复位值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																						
R/W																									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																										GPI0HCLKEN	GPI0GCLKEN	GPI0FCLKEN	GPI0ECLKEN	GPI0DCLKEN	GPI0CCLKEN	GPI0BCLKEN	GPI0ACLKEN																				

xCLKEN: x 模块时钟使能

0 = 模块时钟禁止

1 = 模块时钟使能

31.3.2 PCLK_CTL1 外设时钟控制寄存器 1

表 31-3 PCLK_CTL1 外设时钟控制寄存器 1

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	USBCLKEN			I2C2CLKEN	I2C1CLKEN	I2C0CLKEN	SPI1CLKEN	SPI0CLKEN	USART4CLKEN	USART3CLKEN	USART2CLKEN	USART1CLKEN	USART0CLKEN		TOCLKEN	CMPCLKEN	DAC1CLKEN	DAC0CLKEN	ADC2CLKEN	ADC1CLKEN	ADC0CLKEN	QEI0CLKEN		T9TIOCLKEN		T5T6CLKEN	T4CLKEN	T3CLKEN	T2CLKEN	T1CLKEN	QEI1CLKEN	

xCLKEN: x 模块时钟使能

0 = 模块时钟禁止

1 = 模块时钟使能

31. 3. 3 PCLK_CTL2 外设时钟控制寄存器 2

表 31-4 PCLK_CTL2 外设时钟控制寄存器 2

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W				R/W							R/W		R/W	R/W			
位名	USART7CLKEN	USART6CLKEN	USART5CLKEN			T23CLKEN	T22CLKEN	T21CLKEN	T20CLKEN	T19CLKEN	T18CLKEN			CAN2CLKEN	T15CLKEN	T14CLKEN			DMA1CLKEN		DMA0CLKEN						WWDTCCLKEN			CAN1CLKEN	CAN0CLKEN		

xCLKEN: x 模块时钟使能

0 = 模块时钟禁止

1 = 模块时钟使能

31. 3. 4 PCLK_CTL3 外设时钟控制寄存器 3

表 31-5 PCLK_CTL3 外设时钟控制寄存器 3

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W													R/W			R/W			R/W						R/W					R/W	R/W	
位名													CFGCLKEN			EXICLKEN			CRCLKEN						I2C3CLKEN					SPI3CLKEN	SPI2CLKEN	

xCLKEN: x 模块时钟使能

0 = 模块时钟禁止

1 = 模块时钟使能

32 电源结构及功耗模式 (PM, power manage)

32.1 电源配备介绍

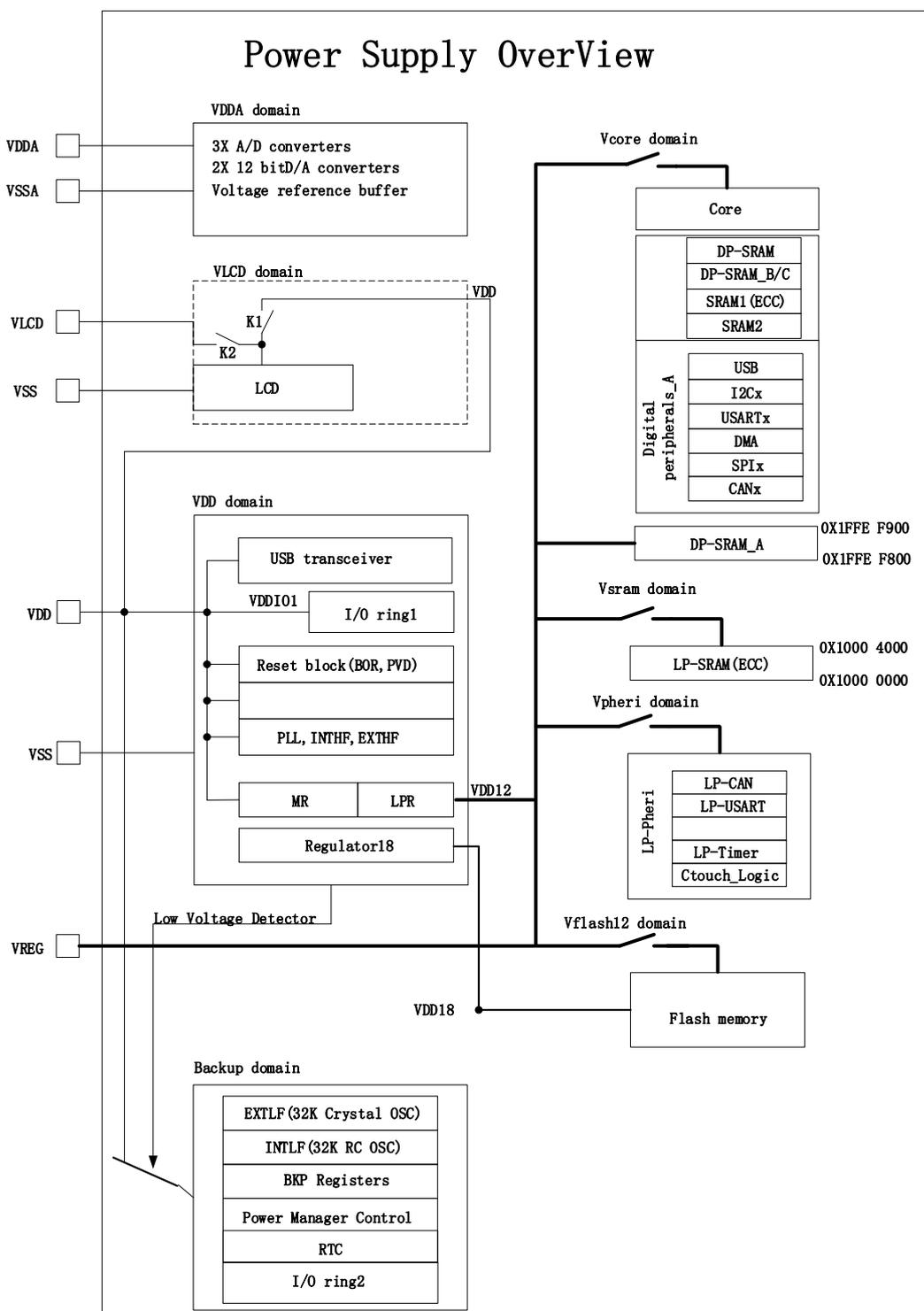


图 32-1 电源结构框图

32.2 电源地引脚说明

表 32-1 电源介绍

电源		
名称	电压范围	注释
VDD	1.67 ~3.6 V	供电电源
VSS	0	通用地线
VDDA	1.67~3.6 V	模拟外设供电电源
VSSA	0	模拟外设专用地线
VLCD	2.5~3.6 V	LCD 模块外部供电电源
VREG	0.9 ~1.32 V	逻辑部分供电电源外部接口

VDD 为芯片的主供电电源，VSS 为芯片的通用地线。VDD 为 I/O 口、内部电压调制器和部分系统（如复位、电源管理和内部时钟等）供电。使用时，要求所有的电源口 VDD 全部连接到同一供电电源，所有的地口 VSS 连接到同一地线。

VDDA/VSSA 为芯片内模拟外设（ADC、DAC12 和 Voltage reference buffer）的专用供电电源口和地线。

VLCD 为 LCD 模块的外部供电电源口。LCD 模块可以通过软件选择供电电源：VDD 或者 VLCD。

VREG 为芯片内部逻辑部分供电电源外部接口。当 VREG 悬空时，芯片内部逻辑部分由 MR 或者 LPR 供电；当 VREG 外接 1.25V 电压时，MR 或 LPR 自动关闭，转由 VREG 供电。

32.3 电压调制器

单片机内置两个电压调制器：主电压调制器 MR 和低功耗电压调制器 LPR。在不同工作模式下，MR 和 LPR 配合使用，为 VREG 供电。MR 和 LPR 在不同模式之间切换时，可通过 PM_CAL2 寄存器的 MR_DELAY<1:0>位和 LPR_DELAY<1:0>位分别配置稳定延长时间。

在不同功耗模式下 MR 和 LPR 的使用情况如下表所示：

表 32-2 功耗模式和电压调制器关系

功耗模式	MR	LPR
Run	on	off
Sleep	on	off
LP-run	off	on
LP-sleep	off	on
Stop	off	on
Standby (SRAM2 on)	off	on
Standby (SRAM2 off)	off	off
Shutdown	off	off

32.4 VREG 电源域分类

VREG 为芯片内部逻辑部分供电电源外部接口。当 VREG 悬空时，芯片内部逻辑部分由 MR 或者 LPR 供电。根据不同功耗模式规格，将 1.2V 域分为如下 5 类：

- Vcore domain: 包括内核、系统、IO 口（除 GPIO0 外）、大部分 SRAM 和大部分外设 PHERI_A（包括模拟外设控制逻辑部门）；
- Vflash12 domain: FLASH 模块 1.2V 域部分；
- Vsram domain: 地址段从 0X1000 0000 到 0X1000 4000 的 RAM(ECC)；
- Vpheri domain: 低功耗外设（LP-CAN、LP-USART、LP-Timer 和 LCD）和 GPIO0。

框图如下所示：

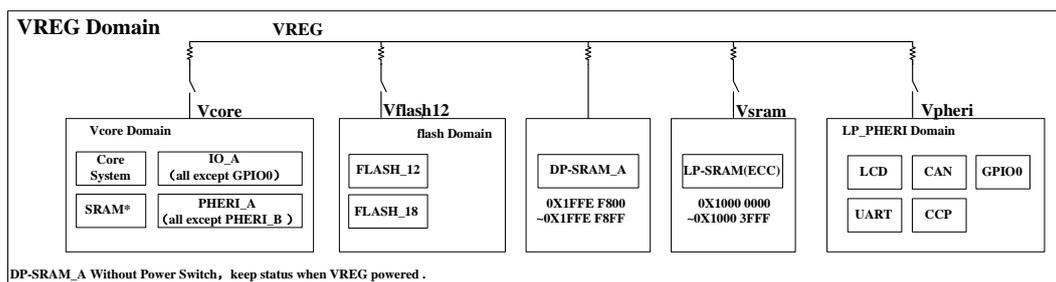


图 32-2 VREG 域框图

32.5 电源结构及功耗模式相关寄存器

 表 32-3 复位相关的寄存器¹

偏移地址	寄存器	访问	功能描述	复位值
0x00	PM_CTL0	R/W	功耗模式控制寄存器 0	0x0300 0000
0x04	PM_CTL1	R/W	功耗模式控制寄存器 1	0x0000 0400
0x08	PM_STA0	R	功耗模式状态寄存器 0	0x0000 0000
0x0C	PM_STA1	R	功耗模式状态寄存器 1	0x0000 0000
0x10	PM_STAC	R/W	功耗模式状态清零寄存器	0x0000 0000
0x14	PM_CTL2	R/W	功耗模式控制寄存器 2	0x0030 0800
0X18	PM_CAL0	R/W	校准寄存器 0	0x0000 0000
0X1C	PM_CAL1	R/W	校准寄存器 1	0x0000 0000

基地址：0x4000 1580

32.5.1 PM_CTL0 功耗模式控制寄存器 0

表 32-4 PM_CTL0 功耗模式控制寄存器 0

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	IOLATCH	VDDPORON	LSIEN	LSEEN	LSEEXEN	MRBGEN	LDO18EN	LPBGEN	BKPREGCLR	IWDTCR	DPRAMSEL	LPRAMSEL	LP4MEN	LSECONF	IWDTRMSEL	PORDELAYSEL	BKPPORDELAYSEL	PHERTIOSEL	PDMSEL	PDMDELAY	<1:0>	HSIEN	BKPWR	OCALLOCK	LPREN	NRSTEN	MEMSEL	LPMS<2:0>				

IOLATCH: IO 口状态锁存控制位

1 = IO 口状态被锁存

0 = IO 口状态未被锁存

VDDPORON: 保留

LSIEN: 内部低频振荡器 INTLF 软件使能位，如需掉电使用，需将该位置 1

1 = 软件使能内部低频振荡器

0 = 软件未使能内部低频振荡器

LSEEN: 外部低频振荡器 EXTLF 软件使能位，如需掉电使用，需将该位置 1

1 = 软件使能外部低频振荡器

0 = 软件未使能外部低频振荡器

LSEEXEN: 外部低频振荡器 EXTLF 时钟输入使能位

1 = 可直接从外部低频晶振输入接口外接时钟信号作为晶振输入

0 = 禁止外部时钟输入功能

MRBGEN: 主 BG 软件使能位

0 = 未软件使能主 BG 模块

1 = 软件使能主 BG 模块

LDO18EN: LDO18 软件使能位

0 = 禁止 LDO18 模块，Power18 (flash) 掉电

1 = 使能 LDO18 模块

¹ 表中所有寄存器位于 Vbkip 域内

- LPBGON: 保留
- BKPREGCLR: 备份区寄存器模块软件复位位
0 = 备份区寄存器模块处于复位状态
1 = 备份区寄存器模块退出复位
- IWDTCLR: IWDT 模块软件复位位
0 = IWDT 处于复位状态
1 = IWDT 退出复位
- DPRAMASEL: DPRAM_A 区在 standby 模式下工作配置位
0 = DPRAM_A 区在 standby 模式下掉电
1 = DPRAM_A 区在 standby 模式下保持数据
- LPRAMSEL: LPRAM 在 stop1 和 standby 模式下工作配置位
0 = LPRAM 在 stop1 和 standby 模式下掉电
1 = LPRAM 在 stop1 和 standby 模式下保持数据
- LP4MEN: LP4M OSC 软件使能位
0 = 禁止 LP4M OSC
1 = 使能 LP4M OSC
- LSECONF: LSE 启动时的 IO 配置位
0 = 未配置
1 = 下拉 LSE 晶振引脚
- IWDTRMSEL: IWDT 模块复位方式选择位
0 = 发生 IWDT 复位时, 将复位 IWDT 模块自身
1 = 发生 IWDT 复位时, IWDT 模块不复位, 继续保持工作状态
- PORDELAYSEL: 主 POR、POR12 和 POR18 延时时间选择位
0 = 2ms
1 = 32us
- BKPPORDELAYSEL: BKP_POR 延时时间选择位
0 = 1ms
1 = 32us
- PHERIIOSEL: USART0 低功耗 IO 口使能位
0 = 禁止 USART0 低功耗 IO 口
1 = 使能 USART0 低功耗 IO 口
- PDMSEL: 保留
- PDMDELAY<1:0>: 保留
- HSIEN: 内部高频振荡器 INTHF 软件使能位
1 = 软件使能内部高频振荡器
0 = 软件未使能内部高频振荡器
- BKPWR: 备份区读写操作允许位
1 = 允许 CPU 对备份区内寄存器进行读写操作
0 = 禁止 CPU 对备份区内寄存器进行读写操作
- OCCALLOCK: HSI 校准寄存器 0 锁存位
1 = 锁存 OSCCAL0 寄存器
0 = 未锁存 OSCCAL0 寄存器
- LPREN: 低功耗电压调制器 LPR 软件使能位
1 = 电压调制器从 MR 切换至 LPR

0 = 电压调制器使用 MR, LPR 被关闭

注: LPREN 位只用于低功耗运行模式的手动切换, LPMS 配置的低功耗模式还是通过硬件直接对 LDO 进行切换, 不会对该位进行配置。

NRSTEN: 外部复位引脚功能软件使能位

1 = 禁止外部复位功能

0 = 使能外部复位功能

MEMSEL: MEMM 数据保持使能。需要按照以下步骤执行:

(1) 将需要的值写入 MEMM<1:0> (保证上电后至少写过一次)

(2) 将 MEMSEL 置 1

LPMS<2:0>: 低功耗模式选择位

000 = 系统保留位

001 = 停止模式 0 (stop mode 0)

010 = 停止模式 1 (stop mode 1)

011 = 待机模式 (standby mode)

1xx = 关断模式 (shutdown mode)

32.5.2 PM_CTL1 功耗模式控制寄存器 1

表 32-5 PM_CTL1 功耗模式控制寄存器 1

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W											R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	BOREN	LPBOREN	TSEN	TSOE	P18OE	INTBUFEN	BUFCLKEN	BUFCLKDIV <2:0>													CAN0_LP_CLKS	DPRAM_ECCEN			VREFSEL<1:0>	VREFEN	PVDS <2:0>				PVDEN	

BOREN: BOR 使能位

0 = 禁止 BOR

1 = 使能 BOR

LPBOREN: LPBOR 使能位

0 = 禁止 LPBOR

1 = 使能 LPBOR

TSEN: TEMPESENSOR 使能位

0 = 禁止 TEMPESENSOR

1 = 使能 TEMPESENSOR

TSOE: TEMPESENSOR 输出到 BUFFER 使能位

0 = 禁止 TEMPESENSOR 输出到 BUFFER

1 = 使能 TEMPESENSOR 输出到 BUFFER

P18OE: P18 输出到 BUFFER 使能位

0 = 禁止 POWER18 输出到 BUFFER

1 = 使能 POWER18 输出到 BUFFER

INTBUFEN: 保留

BUFCLKEN: 保留

BUFCLKDIV<2:0>: 保留

CAN0_LP_CLKS: CAN0 低功耗时钟选择

- 0 = 选择内部 INTLF 时钟;
- 1 = 选择 LPM4M 时钟;
- DPRAM_ECCEN: DPRAM_ECC 使能
 - 0 = DPRAM_ECC 未使能;
 - 1 = DPRAM_ECC 使能;
- VREFSEL<1:0>: 参考电压选择位
 - 00 = 2V
 - 01 = 1.5V
 - 10 = 2.5V
 - 11 = 3V
- VREFEN: 参考电压模块使能位
 - 0 = 禁止参考电压模块
 - 1 = 使能参考电压模块
- PVDS<2:0>: 电压检测点选择位
 - 000 = 1.93V
 - 001 = 2.06V
 - 010 = 2.20V
 - 011 = 2.34V
 - 100 = 2.48V
 - 101 = 2.61V
 - 110 = 2.71V
 - 111 = 保留
- PVDEN: 电压检测功能使能位
 - 0 = 禁止电压检测功能
 - 1 = 使能电压检测功能

注: BUFFER 位于 PB3 脚, 使用 BUFFER 功能时, 请将 PB3 引脚配置为模拟输入状态; 请勿同时使能多个功能输出到 BUFFER, 否则将引起信号冲突, 可能导致芯片工作不正常。

32. 5. 3 PM_CTL2 功耗模式控制寄存器 2

表 32-6 PM_CTL2 功耗模式控制寄存器 2

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W
位名	WKP5EN	WKP4EN	WKP3EN	WKP2EN	WKP1EN	WKP5P	WKP4P	WKP3P	WKP2P	WKP1P	VBKPOREN	VBKPORSHD	PMCIE	PMCI			RTCWEN	RTCTPEN	RTCTSEL	<1:0>	VF12EN	VF12INF	<1:0>	POR18SHD	USARTOCLKLPEN		CCPOLKLPEN	CANOKLPEN	USARTOLPEN		CCPOLPEN	CANOLPEN

- WKP5EN: 外部唤醒引脚 WKP5 使能位
 - 0 = 禁止 WKP5 电平变化唤醒
 - 1 = 使能 WKP5 电平变化唤醒
- WKP4EN: 外部唤醒引脚 WKP4 使能位
 - 0 = 禁止 WKP4 电平变化唤醒
 - 1 = 使能 WKP4 电平变化唤醒

- WKP3EN: 外部唤醒引脚 WKP3 使能位
0 = 禁止 WKP3 电平变化唤醒
1 = 使能 WKP3 电平变化唤醒
- WKP2EN: 外部唤醒引脚 WKP2 使能位
0 = 禁止 WKP2 电平变化唤醒
1 = 使能 WKP2 电平变化唤醒
- WKP1EN: 外部唤醒引脚 WKP1 使能位
0 = 禁止 WKP1 电平变化唤醒
1 = 使能 WKP1 电平变化唤醒
- WKP5P: 外部唤醒引脚 WKP5 唤醒触发沿选择位
0 = WKP5 上升沿触发
1 = WKP5 下降沿触发
- WKP4P: 外部唤醒引脚 WKP4 唤醒触发沿选择位
0 = WKP4 上升沿触发
1 = WKP4 下降沿触发
- WKP3P: 外部唤醒引脚 WKP3 唤醒触发沿选择位
0 = WKP3 上升沿触发
1 = WKP3 下降沿触发
- WKP2P: 外部唤醒引脚 WKP2 唤醒触发沿选择位
0 = WKP2 上升沿触发
1 = WKP2 下降沿触发
- WKP1P: 外部唤醒引脚 WKP1 唤醒触发沿选择位
0 = WKP1 上升沿触发
1 = WKP1 下降沿触发
- VBKPOREN: VBKP 电源 RC_POR 模块使能位
0 = 关闭 VBKP_RC_POR
1 = 打开 VBKP_RC_POR
- VBKPORSHD: VBKP 电源 RC_POR 软件屏蔽位
0 = 未屏蔽 VBKP_RC_POR
1 = 屏蔽 VBKP_RC_POR
- PMCIE: PMC 错误状态中断使能位
0 = 禁止 PMC 错误状态中断
1 = 使能 PMC 错误状态中断
注: 使能后, STOPERROR 和 PMCERROR 都能产生 PMC 错误中断 (外部中断事件<19>)
- PMCIC: PMC 错误状态中断清零位
0 = 无作用
1 = 清零 STOPERROR、PMCERROR 和 DROPLPM 标志
- RTCTWEN: RTC 节拍输出唤醒功能使能位
1 = 允许 RTC 节拍输出将单片机从 stop/standby/shutdown 模式唤醒
0 = 禁止 RTC 节拍输出将单片机从 stop/standby/shutdown 模式唤醒
- RTCTPEN: RTC 节拍输出唤醒分频器使能位
1 = 使能 RTC 节拍输出分频器
0 = 禁止 RTC 节拍输出分频器

RTCTSEL<1:0>: RTC 节拍输出唤醒分频器选择位

00 = 2 分频

01 = 4 分频

10 = 8 分频

11 = 16 分频

VF12EN: FLASH_1.2V 域电源开关软件位

1 = 软件未关断 FLASH_1.2V 域电源开关

0 = 软件关断 FLASH_1.2V 域电源开关

VF12INF1: FLASH_1.2V 域状态信号接口控制位

1 = 接口下拉

0 = 接口不处理

VF12INF0: FLASH_1.2V 域数据信号接口控制位

1 = 接口下拉

0 = 接口不处理

POR18SHD: POR18 信号软件屏蔽位

1 = 屏蔽 POR18

0 = 未屏蔽 POR18

USART0CLKLPEN: USART0 工作在 Stop 模式下内部低频时钟使能位

0 = 禁止时钟

1 = 允许内部低频时钟作为 USART0 工作时钟

CCPCLKLPEN: CCP 工作在 Stop 模式下内部低频时钟使能位

0 = 禁止时钟

1 = 允许内部低频时钟作为 CCP 工作时钟

CAN0CLKLPEN: CAN0 工作在 Stop 模式下内部低频时钟使能位

0 = 禁止时钟

1 = 允许内部低频时钟作为 CAN0 工作时钟

USART0LPEN: USART0 复位控制位

0 = USART0 模块处于复位状态

1 = USART0 模块退出复位

CCP0LPEN: CCP0 复位控制位

0 = CCP0 模块处于复位状态

1 = CCP0 模块退出复位

CAN0LPEN: CAN0 复位控制位

0 = CAN0 模块处于复位状态

1 = CAN0 模块退出复位

32. 5. 5 PM_STA1 功耗模式状态寄存器 1

表 32-8 PM_STA1 功耗模式状态寄存器 1

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	R	R	R	R	R					R			R	R																		
R/W	R	R	R	R	R					R			R	R																		R
位名	WKP5STA	WKP4STA	WKP3STA	WKP2STA	WKP1STA					POR18STA			PMCERROR	STOPERROR																	PVDSTA	

- WKP5STA:** 外部唤醒引脚 WKP5 状态位
 0 = WKP5 引脚未发生唤醒事件
 1 = WKP5 引脚发生唤醒事件
- WKP4STA:** 外部唤醒引脚 WKP4 状态位
 0 = WKP4 引脚未发生唤醒事件
 1 = WKP4 引脚发生唤醒事件
- WKP3STA:** 外部唤醒引脚 WKP3 状态位
 0 = WKP3 引脚未发生唤醒事件
 1 = WKP3 引脚发生唤醒事件
- WKP2STA:** 外部唤醒引脚 WKP2 状态位
 0 = WKP2 引脚未发生唤醒事件
 1 = WKP2 引脚发生唤醒事件
- WKP1STA:** 外部唤醒引脚 WKP1 状态位
 0 = WKP1 引脚未发生唤醒事件
 1 = WKP1 引脚发生唤醒事件
- POR18STA:** 保留
- STOPERROR:** STOP 模式进入错误中断标志
 0 = 正常进入了 STOP 模式
 1 = 进入 STOP 模式前存在唤醒条件
- PMCERROR:** 掉电模式进入错误中断标志
 0 = 正常进入了掉电模式
 1 = 在掉电动作完成前发生了唤醒条件
- PVDSTA:** 电压检测状态位
 0 = VDD 电压高于 PVD 电压点
 1 = VDD 电压低于 PVD 电压点

32. 5. 6 PM_STAC 功耗模式状态清零寄存器

表 32-9 PM_STAC 功耗模式状态清零寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	R/W	R/W	R/W	R/W	R/W																											
R/W	R/W	R/W	R/W	R/W	R/W																											
位名	WKP5SC	WKP4SC	WKP3SC	WKP2SC	WKP1SC																											

10 = BOR 电压点为 2.25V

11 = BOR 电压点为 2.49V

PM_IWDTEN: IWDT 使能位

0 = 无作用

1 = 使能 IWDT

注:使能后也会自动使能内部低频 INTLF, 且产生 IWDT 复位时不复位 IWDT 自身。

BOR_VREF_SEL: BOR 参考电压选择位

0 = BOR 参考电压选择 LPBG

1 = BOR 参考电压选择 MRBG

MRTRIM<1:0>: MR 模块校准位

00 = 1.2V

01 = 0.9V

10 = 1V

11 = 1.32V

MRHVEN: MR_HV 模块软件使能位

0 = 禁止 MR_HV 模块

1 = 使能 MR_HV 模块

MRHVTRIM<1:0>: MR_HV 模块校准位

00 = 1.2V

01 = 0.9V

10 = 1V

11 = 1.32V

BGTRIMREBUF<4:0>: 参考 BUFFER 基准校准位

10000-10001...-11111-00000-00001...-01110-01111 以此增加 5mV

BGTRIMRLDO<3:0>: LDO12 基准校准位

1000-1001...-1111-0000-0001...-0110-0111 以此增加 10mV

BGTRIMFLDO<3:0>: LDO18 基准校准位

1000-1001...-1111-0000-0001...-0110-0111 以此增加 15mV

32.5.8 PM_CAL1 校准寄存器 1

表 32-11 PM_CAL1 校准寄存器 1

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
位名	LSITRIMBIAS <2:0>		LSITRIMCAP <4:0>				SPCPUR	LPTRIMBIAS <2:0>		LSESEL	LPBGTRIMPUMP <2:0>		HSE_SEL		LP4MTRIM<8:0>								LSETRIMBIAS <3:0>			LSETRIMBIAS <1:0>						

LSITRIMBIAS<2:0>: 内部低频振荡器电流校准位

000 = 10nA 偏置电流

001 = 12.5nA 偏置电流

010 = 15nA 偏置电流

011 = 17.5nA 偏置电流

100 = 0nA 偏置电流
101 = 2.5nA 偏置电流
110 = 5nA 偏置电流
111 = 7.5nA 偏置电流

LSITRIMCAP<4:0>: 内部低频振荡器电容校准位

01111-01110...00001-00000-11111-11110...10001-10000 以此增加测出频率的 3.3%

SPCPUR: SPCLK 口上拉信号锁存允许位

0 = SPCLK 口弱上拉配置允许被 PM_CTL0 寄存器的 IOLATCH 位锁存
1 = SPCLK 口弱上拉配置不会被 PM_CTL0 寄存器的 IOLATCH 位锁存

LPTRIMBIAS<2:0>:

001-010-011-000-100-101-110-111 以此增加测出电流的 10%

LSESEL: 外部低频晶振模块选择位

0 = 选择外部低频晶振 1, 端口引脚位于 PD11/PD12
1 = 选择外部低频晶振 2, 端口引脚位于 PH8/PH9

LPBGTRIMPUMP<2:0>:

100-101-110-111-000-001-010-011 以此增加 cp 电压 36Mv

HSE_SEL: 外部高频晶振模块选择位

0 = 选择外部高频晶振 1, 端口引脚位于 PD14/PD15
1 = 选择外部高频晶振 2, 端口引脚位于 PD9/PD10

LP4MTRIM<8:0>: LP4M 振荡器校准位

LSETRIMBIAS<3:0>: 外部低频振荡器偏置校准位

000 = 20nA 偏置电流
001 = 25nA 偏置电流
010 = 30nA 偏置电流
011 = 35nA 偏置电流
100 = 0nA 偏置电流
101 = 5nA 偏置电流
110 = 10nA 偏置电流
111 = 15nA 偏置电流

LSETRIMNBIAS<1:0>: 外部低频振荡器 N 管偏置校准位

00 = 没有增加支路电流
01 = 增加 50%支路电流
10 = 增加 150%支路电流
11 = 增加 200%支路电流

32.6 工作模式介绍

单片机提供多种工作模式供用户在不同工作场景下使用。

运行模式: 正常运行模式。正常运行模式允许用户使用任何资源以达到最佳性能。

运行模式

- 正常运行模式 (Run)

休眠模式: 普通休眠模式。休眠模式的内部逻辑由 MR 供电。

休眠模式

- 普通休眠模式 (Sleep)

三种低功耗模式：停止模式、待机模式和关断模式；低功耗模式下，系统时钟、高频时钟和大部分外设被禁止工作，提供不同程度下的区域断电以节省待机功耗。

低功耗模式

- 停止模式 (Stop)
- 待机模式 (Standby)
- 关断模式 (Shutdown)

32.6.1 工作模式切换关系图

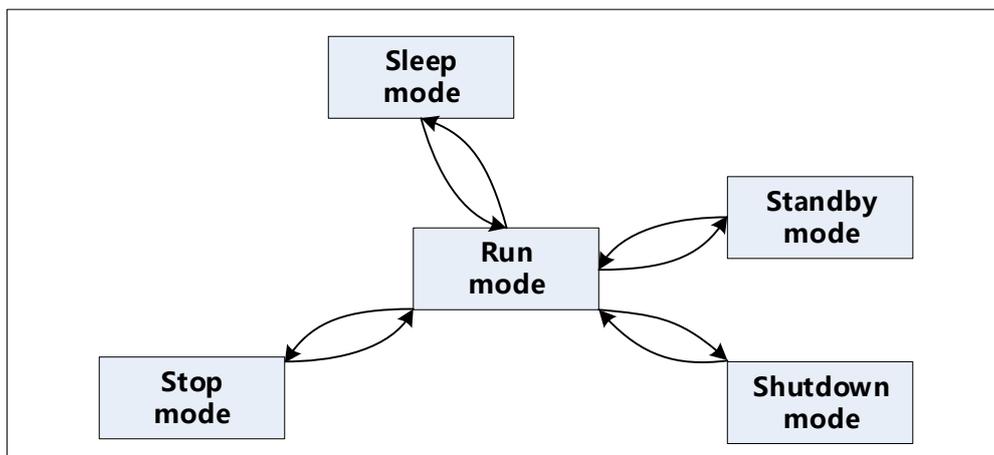


图 32-3 工作模式切换关系图

32.6.2 正常运行模式

正常运行模式下，允许用户使用任何资源（时钟、外设、存储等）。在正常运行模式下，MCU 正常执行指令工作；系统时钟工作，外设高频时钟、外设低频时钟均可正常使用。Flash/RAM 均可正常读写；所有外设均可正常使用；所有 I/O 口均可正常使用；用户可以按照需求对 MCU 进行设置以达到最佳性能。

在正常运行模式下，MCU 内部逻辑供电电压由 MR 提供。

在正常运行模式下，MCU 可以通过休眠指令进入普通休眠模式，也可以通过配置进入停止模式、待机模式和关断模式。

32.6.3 普通休眠模式

在正常运行模式下执行休眠指令，单片机将进入普通休眠模式。在普通休眠模式下，除 CPU 停止执行指令、FLASH/RAM 保持外，系统时钟仍旧正常工作，在进入休眠模式之前使能的外设功能也将继续工作，不会被屏蔽。因此在休眠模式下，所有外设功能均可使用；所有 I/O 口均可被分配给相关外设或者保持状态。

在普通休眠模式下，MCU 内部逻辑供电电压由 MR 提供。

普通休眠模式只能在正常运行模式下执行休眠指令进入。任何中断、IWDG/WWDT 复位、NRST 引脚复位都将会使单片机唤醒/复位至正常运行模式。

表 32-12 普通休眠模式配置表

普通休眠模式 Sleep mode	
进入	在正常运行模式且无任何复位/唤醒条件发生情况下，执行休眠指令
电压调制器	MR
I/O 口	保持正常运行模式时的状态
唤醒方式 (至正常运行模式)	NRST 引脚复位 IWDT/WWDT 复位 任何已使能的中断唤醒
唤醒时间	3~4 个 SCLK

32.6.4 休眠模式间关系表

表 32-13 休眠模式间关系表

功耗模式	普通休眠	
电压调制器	MR	MR
系统时钟	Y	N
唤醒至	RUN	RUN
CPU 指令运行	停止	

32.6.5 停止模式

单片机提供两种停止模式：Stop0 和 Stop1。

32.6.5.1 Stop0 模式

Stop0 模式可以从正常运行模式通过配置 PM_CTL0 寄存器的 LPMS<2:0>=001H 进入。在 Stop0 模式下，CPU 停止运行，RAM 保持，FLASH 掉电；外设被允许使用低频时钟工作在停止模式，高频时钟和 PLL 被禁止，只允许使用内部低频时钟和外部低频时钟；所有 I/O 口均可被分配给相关外设或者保持状态。

在停止模式下，MCU 内部逻辑供电电压由 LPR 提供。

IWDT 复位、NRST 引脚复位都将会使单片机复位至正常运行模式。允许工作在停止模式下的外设中断和外部引脚中断，可以将单片机唤醒至正常运行模式。

表 32-14 停止模式配置表

Stop0 mode	
进入	在正常运行模式： 无任何复位/唤醒条件发生条件下，将 PM_CTL0 寄存器的 LPMS<2:0>位配置为 001H；执行休眠指令；
电压调制器	LPR
I/O 口	保持正常运行模式时的状态
电源开关	FLASH 掉电
唤醒方式	NRST 引脚复位（备份域被复位） IWDT 复位（备份域不复位） 外部唤醒引脚唤醒 外设中断唤醒： LP-TIMER

	LP-UART LP-CAN RTC
唤醒时间	96us/2ms 可选

32.6.5.2 Stop1 模式

Stop1 模式可以从正常运行模式通过配置 PM_CTL0 寄存器的 LPMS<2:0>=010H 进入。在 Stop1 模式下，Vcore 域（包括内核、大部分外设和大部分 SRAM）和 FLASH 掉电；LP-SRAM 可以通过软件配置是否断电；小部分外设（BOR/PVD/ RTC/IWDT/ LCD/LP-TIMER/LP-UART/LP-CAN）被允许工作在 Stop1 模式，只允许使用内部低频时钟和外部低频时钟；部分 I/O 口可被分配给相关外设或者保持状态。

在 Stop1 模式下，MCU 内部逻辑供电电压由 LPR 提供。DPRAM_A 数据保持；LPRAM 可以通过软件配置是否保持数据。

IWDT 复位、NRST 引脚复位都将会使单片机复位至正常运行模式。允许工作在 Stop1 模式下的外设中断和外部引脚中断，可以将单片机唤醒至正常运行模式。

表 32-15 停止模式配置表

Stop1 mode	
进入	在正常运行模式下： 无任何复位/唤醒条件发生条件下，将 PM_CTL0 寄存器的 LPMS<2:0>位配置为 010H；执行休眠指令；
电压调制器	LPR
I/O 口	保持正常运行模式时的状态
电源开关	Vcore 域掉电 FLASH 掉电
唤醒方式	NRST 引脚复位（备份域被复位） IWDT 复位（备份域不复位） 外部唤醒引脚唤醒 RTC 闹钟中断唤醒 外设中断唤醒： LP-TIMER LP-UART LP-CAN RTC
唤醒时间	96us/2ms 可选

32.6.6 待机模式

在正常运行模式下，通过配置 PM_CTL0 寄存器的 LPMS<2:0>=011H 进入待机模式。在待机模式下，CPU 停止运行且进入掉电状态，同时 FLASH /SRAM 也进入掉电状态；DPRAM_A 和 LPRAM 可以通过软件配置选择保持数据或者进入掉电状态；高频时钟和 PLL 被禁止使用，内部低频时钟和外部低频时钟被允许使用；少部分外设（BOR/RTC/IWDT）被允许工作在待机模式，其余模拟外设被禁止使用，以及数字外设均进入掉电状态；所有 I/O 口均可通过 I/O 口状态锁存位锁存需要的状态。

在待机模式下，如 DPRAM_A 和 LPRAM 被配置为掉电，则 MR 和 LPR 均被关闭；如 DPRAM_A 或 LPRAM 被配置为数据保持，则 LPR 工作，提供供电电压。

IWDT 复位、NRST 引脚复位、外部唤醒引脚和 RTC 中断（只允许闹钟）可以将单片机从待机模式下唤醒/复位；单片机从待机模式下唤醒后，除 DPRAM_A 和 LP-SRAM（如配置为数据保持）以及备份区外，所有掉电的区域重新上电，并处于初始化状态（被复位），上电完成后，CPU 重新开始执行，单片机开始工作。

待机模式可以从正常运行模式进入，唤醒/复位发生后转入正常运行模式。

表 32-16 待机模式配置表

待机模式 Standby mode	
进入	在正常运行模式下： 无任何复位/唤醒条件发生条件下，将 PM_CTL0 寄存器的 LPMS<2:0>位配置为 011H；执行休眠指令；
电压调制器	DPRAM_A 或 LPRAM 保持：LPR
	DPRAM_A 和 LPRAM 掉电：无
I/O 口	保持锁存时的状态
电源开关	Vcore 域掉电； Flash 域掉电； Vpheri 域掉电； DPRAM_A 和 LPRAM 根据配置选择是否掉电；
唤醒方式	NRST 引脚复位 IWDT 复位 外部唤醒引脚唤醒 RTC 中断唤醒
唤醒时间	160us/4ms 可选

32.6.7 关断模式

在正常运行模式下，通过配置 PM_CTL0 寄存器的 LPMS<2:0>=1xxH 进入关断模式。在关断模式下，CPU、FLASH、EE 和 SRAM 均被掉电；备份区保持；高频时钟和 PLL 被禁止使用，内部低频时钟和外部低频时钟被允许使用；除 RTC 被允许工作在关断模式以外，模拟外设被禁止使用，数字外设（IWDT 被禁止使用）均进入掉电状态；所有 I/O 口均可通过 I/O 口状态锁存位锁存需要的状态。

在关断模式下，DPRAM_A 和 LPRAM 不再可软件选择是否保持数据，只能进入掉电状态，MR 和 LPR 均被关闭；在待机模式的基础上，关断模式下 BOR 和 IWDT 被禁止使用。

NRST 引脚复位、外部唤醒引脚和 RTC 中断（只允许闹钟）可以将单片机从关断模式下唤醒/复位；单片机从关断模式下唤醒后，除备份区外，所有掉电的区域重新上电，并处于初始化状态（被复位），上电完成后，CPU 重新开始执行，单片机开始工作。

表 32-17 关断模式配置表

关断模式 Shutdown mode	
进入	在正常运行模式下： 无任何复位/唤醒条件发生条件下，将 PM_CTL0 寄存器的 LPMS<2:0>位配置为 1xxH；执行休眠指令；
电压调制器	无
I/O 口	保持锁存时的状态
电源开关	整个 VREG 域掉电； FLASH 掉电；
唤醒方式	NRST 引脚复位

	外部唤醒引脚唤醒 RTC 中断唤醒
唤醒时间	256us/6ms 可选

32. 6. 8 工作模式与电源域关系表

表 32-18 工作模式与电源域关系表

Pow-Mode/Pow- domain	Regulator		Vcore	Vflash		Vsram		Vpheri	VDD33	VBKP
	MR	LPR	CPU&PHERI_A &SRAM_A&IO_A	FLASH_1.2V	FLASH_1.8V	DP-SRAM_A	LP-SRAM	LP-PHERI &GPIOA	Main_Power	Backup_Power
Run	ON	OFF	ON	ON	ON	ON	ON	ON	ON	ON
Sleep	ON	OFF	ON	ON	ON	ON	ON	ON	ON	ON
Run_SRAM	OFF/ON	OFF/ON	ON	OFF	OFF	ON	ON	ON	ON	ON
Stop0	OFF	ON	ON	OFF	OFF	ON	ON	OFF/ON	ON	ON
Stop1	OFF	ON	OFF	OFF	OFF	ON	OFF/ON	ON	ON	ON
Standby0(with LP-SRAM)	OFF	ON	OFF	OFF	OFF	ON	ON	OFF	ON	ON
Standby1(without LP-SRAM)	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
Shutdown	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON

32.7 低功耗模式外部唤醒引脚 Wakeup Pin

单片机配备 5 个外部唤醒引脚 WKP_x (x=1-5)，用于唤醒 Stop 模式、Standby 模式或 Shutdown 模式。

PM_CTL2 寄存器的 WKP_xEN 位用于使能 WKP_x 引脚功能；WKP_xP 位用于选择触发沿；PM_STA1 寄存器的 WKP_xSTA 位表示 WKP_x 引脚是否发生唤醒事件；PM_STAC 寄存器的 WKP_xSC 位用于对 WKP_xSTA 位清零，置 1 将清零对应的 WKP_xSTA 位，清零操作后，需要软件清零 WKP_xSC 位，否则 WKP_xSTA 位将无法置 1。

在使用外部唤醒引脚功能时，在进入 Stop 模式、Standby 模式或 Shutdown 模式之前，必须先执行清零 WKP_xSTA 位操作。如果在 WKP_xSTA 位为 1 的条件下执行休眠指令，将导致单片机无法进入 Stop 模式、Standby 模式或 Shutdown 模式，转而进入其他休眠模式，从而导致单片机无法正常唤醒。还需使能总中断。在唤醒中断服务程序中需要清除 WKPIF 标志位。

33 备份域 (BKP)

33.1 概述

备份域中的 32 个 32 位带有侵入检测功能的备份寄存器，可用于保存数据；寄存器组在 VDD 电源被切断时，仍然可以通过 VBAT 维持供电。备份域内寄存器只会在初始上电复位时被复位，不会因为 VDD 掉电上电而复位。

33.2 备份域数据寄存器组相关寄存器

表 33-1 备份域数据寄存器组相关的寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x00	BKP_CTL	R/W	备份域控制寄存器	0x0000 0000
0x04	BKP_INT	R/W	备份域中断控制寄存器	0x0000 0000
0x40~0xBC	BKP_DATAx	R/W	备份域数据寄存器 (x=0~31)	0x0000 0000

基地址: 0x4000 1400

33.2.1 BKP_CTL 备份域控制寄存器

表 33-2 BKP_CTL 备份域控制寄存器

	031	030	029	028	027	026	025	024	023	022	021	020	019	018	017	016	015	014	013	012	011	010	09	08	07	06	05	04	03	02	01	00
复位值																																
R/W	R/W									R/W	R/W	R/W		R/W	R/W	R/W							R/W	R/W				R/W	R/W	R/W	R/W	
位名	BKDRST									TAMP3LVL	TAMP2LVL	TAMP1LVL		TAMP3EN	TAMP2EN	TAMP1EN							RTCCLK <1:0>					SYNC	TAMPCLKS	EXTLFBYP	EXTHFBYP	

BKDRST: 备份域寄存器组软件复位。BKDRST 位软件置 1 后, 会在一个系统时钟周期后自动清零。

0 = 无作用

1 = 备份域寄存器组复位

TAMP3LVL: 侵入检测引脚 RTC_TAMP3 有效电平

0 = 侵入检测引脚上的高电平会清除所有的数据备份寄存器

1 = 侵入检测引脚上的低电平会清除所有的数据备份寄存器

TAMP2LVL: 侵入检测引脚 RTC_TAMP2 有效电平

0 = 侵入检测引脚上的高电平会清除所有的数据备份寄存器

1 = 侵入检测引脚上的低电平会清除所有的数据备份寄存器

TAMP1LVL: 侵入检测引脚 RTC_TAMP1 有效电平

0 = 侵入检测引脚上的高电平会清除所有的数据备份寄存器

1 = 侵入检测引脚上的低电平会清除所有的数据备份寄存器

TAMP3EN: 侵入检测引脚 RTC_TAMP3 使能

0 = 禁止侵入引脚 3 功能

1 = 侵入引脚 3 功能使能

TAMP2EN: 侵入检测引脚 RTC_TAMP2 使能

0 = 禁止侵入引脚 2 功能

1 = 侵入引脚 2 功能使能

TAMP1EN: 侵入检测引脚 RTC_TAMP1 使能

0 = 禁止侵入引脚 1 功能

1 = 侵入引脚 1 功能使能

RTCCLKS<1:0>: RTC 时钟源选择

00 = 无时钟

01 = EXTLF 作为 RTC 时钟

10 = INTLF 作为 RTC 时钟

11 = EXTHF 经过 128 分频后作为 RTC 时钟

SYNC: 侵入检测信号同步异步选择位

0 = 同步模式

1 = 异步模式

SAMPCLKS: 侵入检测信号同步采样时钟选择位

0 = 选择 SCLK 作为采样时钟

1 = 选择内部低频 INTLF 作为采样时钟

注: 只有 SYNC 为 0 时该位才有效

EXTLFBYP: 外部低频时钟旁路控制位

0 = 外部低频时钟未旁路

1 = 外部低频时钟被旁路

EXTHFBYP: 外部高频时钟旁路控制位

0 = 外部高频时钟未旁路

1 = 外部高频时钟被旁路

33. 2. 2 BKP_INT 备份域中断控制寄存器

表 33-3 BKP_INT 备份域中断控制寄存器

复位值	0	31	0	30	0	29	0	28	0	27	0	26	0	25	0	24	0	23	0	22	0	21	0	20	0	19	0	18	0	17	0	16	0	15	0	14	0	13	0	12	0	11	0	10	0	9	0	8	0	7	0	6	0	5	0	4	0	3	0	2	0	1	0	0
R/W																			R/W	R/W	R/W													R/W	R/W	R/W													R/W	R/W	R/W													
位名																			TAMP3IF	TAMP2IF	TAMP1IF													TAMP3IE	TAMP2IE	TAMP1IE													TAMP3IC	TAMP2IC	TAMP1IC													

TAMP3IF: 侵入中断标志位

0 = 无侵入事件发生

1 = 发生侵入事件

TAMP2IF: 侵入中断标志位

0 = 无侵入事件发生

1 = 发生侵入事件

TAMP1IF: 侵入中断标志位

0 = 无侵入事件发生

1 = 发生侵入事件

注: 当检测到有侵入事件且 TAMPxIE 位为 1 时, 此位由硬件置 1。通过向 TAMPxIC 位写 1 来清除此标志位(同时也清除了中断)。如果 TAMPxIE 位被清除, 则此位也会被清除。仅当系统复位后才复位该位。

TAMP3IE: 侵入引脚中断使能位。

0 = 禁止侵入检测中断

1 = 使能侵入检测中断

TAMP2IE: 侵入引脚中断使能位。

0 = 禁止侵入检测中断

1 = 使能侵入检测中断

TAMP1IE: 侵入引脚中断使能位。

0 = 禁止侵入检测中断

1 = 使能侵入检测中断

注：侵入中断无法将系统内核从低功耗模式中唤醒，仅当系统复位后才复位该位。

TAMP3IC：清除侵入检测中断

0 = 无效

1 = 清除侵入检测中断和 TAMPIF 中断标志位

TAMP2IC：清除侵入检测中断

0 = 无效

1 = 清除侵入检测中断和 TAMPIF 中断标志位

TAMP1IC：清除侵入检测中断

0 = 无效

1 = 清除侵入检测中断和 TAMPIF 中断标志位

33. 2. 3 BKP_DATAx 备份域数据寄存器 (x=0~31)

表 33-4 BKP_DATAx 备份域数据寄存器 (x=0~31)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	BKPDATAx<31:0>																																

BKPDATAx<31:0>：备份域备份数据寄存器 (x=0~31)

该寄存器不会被系统复位、电源复位。它们可由备份域复位或侵入事件复位。起始上电时，还会被 POR 复位。

33.3 寄存器组的复位

完成上电复位后，在 VBAT 有电条件下，备份域寄存器组数据不会因为 VDD 掉电而丢失，VBAT 将继续维持寄存器组供电以维持数据。

PM_CTL0 寄存器的 BKPREGCLR 位可用于备份域寄存器组的软件复位，当 BKPREGCLR=0 时，备份域寄存器组模块所有相关寄存器（包括控制和数据寄存器）被复位。注意，BKPREGCLR 位清零时将同时复位 RTC 模块，如果用户不希望在复位寄存器组的同时复位 RTC 模块，可以通过清零 BKP_CTL 寄存器的 BKDRST 位单独对寄存器组进行复位。

33.4 寄存器组的读写

对备份域数据寄存器组进行操作时，需要对如下寄存器位进行配置：

- (1) OSC_CTL0 寄存器的 PMWREN 位置 1，允许整个备份域的读写操作；
- (2) PM_CTL0 寄存器的 BKPREGCLR 位置 1，寄存器组退出复位；
- (3) PM_CTL0 寄存器的 BKPWR 位置 1，允许备份域数据寄存器组和 RTC 模块内的寄存器读写操作。

33.5 侵入检测

当 TAMPx 引脚上的检测到指定电平时（取决于备份控制寄存器 TAMPxLVL 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。

- 当 TAMPxLVL=0 时：如果在启动侵入检测 TAMPx 引脚前(TAMPxEN 位为 0)该引脚已经为高电平，一旦启动侵入检测功能，则会产生一个侵入事件。
- 当 TAMPxLVL=1 时：如果在启动侵入检测 TAMPx 引脚前(TAMPxEN 位为 0)该引脚已经为低电平，一旦启动侵入检测功能，则会产生一个的侵入事件。

设置 BKP_INT 寄存器的 TAMPxIE 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，侵入检测引脚 TAMPx 应该被禁止。然后，在再次写入备份数据寄存器前重新用 TAMPxEN 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 TAMPx 进行电平检测。

注：当 VDD 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，TAMPx 引脚应该在片外连接上拉/下拉电阻。

33.6 侵入检测信号同步/异步模式

通过 BKP_CTL 寄存器的 SYNC 位可以配置侵入检测信号的作用模式，当 SYNC 为 0 时，侵入检测信号工作在同步模式，侵入检测引脚上的信号需要经过时钟采样同步，同步后的信号才是真正的侵入检测信号。当 SYNC 为 1 时，侵入检测信号工作在异步模式，侵入检测引脚上的信号无需经过时钟采样，直接生效。

当侵入检测信号工作在同步模式时，可以通过 SAMPCLKS 位选择同步采样时钟，当 SAMPCLKS 位为 0 时，选择 SCLK 作为采样时钟，当 SAMPCLKS 位为 1 时，选择内部低频 INTLF 作为采样时钟。

注：在部分低功耗模式下，MCU 的系统时钟 SCLK 会停止工作，此时可以通过改变采

样时钟或者选择异步模式的方法，让侵入检测功能正常工作。

33.7 外部时钟旁路功能

可以通过 EXTLFBYP/EXTHFBYP 使能外部低频旁路功能/外部高频旁路功能。当使能旁路功能时，可以从振荡器的输入引脚输入需要的频率，振荡器的输出引脚可以用作通用 IO 口。

34 Flash 程序存储器

34.1 FLASH 概述

- 512 Kbyte 的空间大小
- 3K 的用户配置区（只能做数据存储区使用）
- 以 64bit 为单元进行读写访问
- 支持指令预取
- 加密模式可选
- 纠 1 位错的 ECC

34.2 FLASH 相关寄存器

表 34-1 振荡器相关寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x100	FLASH_ISPCON0	R/W	FLASH_ISPCON0 控制寄存器	0x0000 0040
0x104	FLASH_ISPCON1	R/W	控制寄存器	0x0000 0002
0x108	FLASH_ISPCMD	R/W	控制寄存器	0x0000 07E0
0x10C	FLASH_ISPTRG	R/W	控制寄存器	0x0000 0000
0x114	FLASH_CFG	R/W	控制寄存器	0x0000 00C7
0x11C	FLASH_ISPADDR	R/W	地址寄存器	0x0000 0000
0x120	FLASH_STATE	R/W	状态寄存器	0x0000 0000
0x128	FLASH_NVMUNLOCK	R/W	解锁 KEY 寄存器	0xuuuu uuuu
0x12C	FLASH_PROUNLOCK	R/W	解锁 KEY 寄存器	0xuuuu uuuu
0x130	FLASH_CFGUNLOCK	R/W	解锁 KEY 寄存器	0xuuuu uuuu
0x138	FLASH_CSSTART	R/W	寄存器	0x0000 0000
0x13C	FLASH_CSSTOP	R/W	寄存器	0x0000 0000
0x140	FLASH_CSRES0	R/W	寄存器	0x0000 0000
0x144	FLASH_CSRES1	R/W	寄存器	0x0000 0000
0x148	FLASH_CSRES2	R/W	寄存器	0x0000 0000
0x14C	FLASH_CSRES3	R/W	寄存器	0x0000 0000

基地址：0x4020 0000

34.2.1 FLASH_ISPCON0 控制寄存器

表 34-2 FLASH_ISPCON0 控制寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0																					
R/W																									R/W	R/W	R/W	R/W	R/W	R/W																							
位名																									STANDBY1	IFEN	DWEN	CFGLOCK	FLASHLOCK	NVMLOCK																							

STANDBY1: (高 16 位必须写 50AF 才能进行写入操作)

1 = FLASH 进入 Standby 模式

0 = FLASH 处于正常模式

IFEN: FLASH 信息区选择位, 不要试图在 FLASH 运行程序时, 改变该位的值。信息区和程序区无法并行访问。

0 = NVM 操作存储器选择 FLASH 程序区

1 = NVM 操作存储器选择 FLASH 信息区

DWEN: FLASH 写缓存使能位

0 = FLASH 写缓存禁止被写入数据

1 = FLASH 写缓存可以被写入数据

CFGLOCK: FLASH 配置区解锁状态位 (解锁擦写 CFG 权限)

0 = 上锁状态

- 1 = 解锁状态（解锁需要向 FLASH_CFGUNLOCK 寄存器写入特定序列）
（往该位写“1”清零此状态位，高 16 位必须写 50AF）

FLASHLOCK: FLASH 解锁状态位（解锁擦写 FLASH 权限）

- 0 = 上锁状态
- 1 = 解锁状态（解锁需要向 FLASH_PROUNLOCK 寄存器写入特定序列）
（往该位写“1”清零此状态位，高 16 位必须写 50AF）

NVMLOCK: FLASH 编程组件解锁状态位（解锁 FLASH 相关寄存器使能位）

- 0 = 上锁状态
- 1 = 解锁状态（解锁需要向 FLASH_NVMUNLOCK 寄存器写入特定序列）
（往该位写“1”清零此状态位，高 16 位必须写 50AF）

34. 2. 2 FLASH_ISPCON1 控制寄存器

表 34-3 FLASH_ISPCON1 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W																R/W	R/W										R/W	R/W	R/W	R/W	
位名																	CHIPONEN											CONFEN	RECALLEN	IPSEL		

CHIPONEN: 芯片信息区擦写解锁状态位（高 16 位必须写 A05F 才能对该位进行写操作）

- 0 = 上锁状态
- 1 = 解锁状态

CONFEN: FLASH 校准信息更新控制位（高 16 位必须写 A05F 才能对该位进行写操作）

- 0 = 不更新
- 1 = 将 FLASH_DATA 总线的数据更新到 FLASH-IP 内部的 Trim 寄存器

RECALLEN: 正常模式下，读 FLASH 最少需要 20ns；在 RECALL 模式下，读 FLASH 最少需要 80ns，RECALL 模式会更加稳定。

- 0 = FLASH 读被设置为正常模式
- 1 = FLASH 读被设置为 RECALL 模式（RECALL 模式比正常模式读 FLASH 需要更久的时间）

IPSEL: 选择其中某一个 FLASH-IP

- 0 = 某些操作时选择 FLASH-IP1
- 1 = 某些操作时选择 FLASH-IP2

34. 2. 3 FLASH_ISPCMD 控制寄存器

表 34-4 FLASH_ISPCMD 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	9	8	7	6	5	4	3	2	1	0																			
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0																			
R/W	R/W																					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名																						WSIZE<5:0>					CMD<4:0>																								

WSIZE<5:0>: 半页编程模式下, 需要写入的字数

CMD<4:0>: 命令选择位

10101 = 单字编程 (只能写 64bit)

01010 = 片擦

11011 = 页擦 (擦除当前地址所在的 PAGE)

00101 = 半页编程 (最多能写一行)

34. 2. 4 FLASH_ISPTRG 控制寄存器

表 34-5 FLASH_ISPTRG 控制寄存器

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

ISPGO: 命令触发位, 往该位写 1, 执行当前 FLASH_ISPCMD 寄存器设定的命令。高 16 位必须为 0x50AF 才能写该寄存器。

34. 2. 5 FLASH_CFG 控制寄存器

表 34-6 FLASH_CFG 控制寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																				
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																				
R/W	R/W																						R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
位名																							ECCREADEN		PREFETCHEN																										TCFG<3:0>	

ECCREADEN: 非易失性存储器 ECC 使能位

0 = 不使能 ECC 校验

1 = 使能 ECC 校验

PREFETCHEN: 线性预取使能位

0 = 不使能线性预取

1 = 使能线性预取

TCFG<3:0>: FLASH 访问周期控制位

0000 = 1 个系统周期

0001 = 2 个系统周期

...

1110 = 15 个系统周期

1111 = 16 个系统周期

34. 2. 6 FLASH_ISPADDR 地址寄存器

表 34-7 FLASH_ISPADDR 寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
复位值													R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W																																	
位名													SADDR<19:2>																				

SADDR<2:0>: 指定 FLASH 在 CPU 执行编程和行擦期间的地址信息

34. 2. 7 FLASH_STATE 状态寄存器

表 34-8 FLASH_STATE 状态寄存器

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																				
复位值																				R/W	R/W																															
R/W																																																				
位名																															BDBGREN	BDBG																	TBUSY	TBIT	SIGDONE	CFGERRR

TBIT: FLASH 擦写状态位

1=ISP 正在执行擦除或者写命令

0=ISP 处于空闲状态

TBUSY: FLASH 编程状态位

1=FLASH 正处于编程状态

0=FLASH 不在编程状态

SIGDONE:

1=计算完成

0=未计算或者计算中

写“1”清零

34. 2. 8 FLASH_NVMUNLOCK 解锁 KEY 寄存器

表 34-9 FLASH_NVMUNLOCK 寄存器

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
复位值	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
R/W	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
位名	NVMUNLOCK<31:0>																															

向该寄存器连续写入 KEY1 和 KEY2 的值，可以将 FLASH_ISPCON0 寄存器的 NVMLOCK 置 1。

KEY1: 0x8765_4321

KEY2: 0x0504_0302

34. 2. 13 FLASH_CSRES0 寄存器

表 34-14 FLASH_CSRES0 寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CKSPRES0<31:0>																																	

34. 2. 14 FLASH_CSRES1 寄存器

表 34-15 FLASH_CSRES1 寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CKSPRES1<31:0>																																	

34. 2. 15 FLASH_CSRES2 寄存器

表 34-16 FLASH_CSRES2 寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CKSPRES2<31:0>																																	

34. 2. 16 FLASH_CSRES3 寄存器

表 34-17 FLASH_CSRES3 寄存器

复位值	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CKSPRES3<31:0>																																	

34.3 FLASH 存储映射

FLASH 包含程序区和信息区。信息区需要解锁 ISPCON0 寄存器的 CFGLOCK 位以及配置 FLASH_ISPCON0 的 IFEN，才能信息区访问。当信息区被指向时，信息区被映射到地址 0000H~1FFFH。

注：FLASH 信息区仅仅作为数据存储器，严格禁止在程序区运行指令时切换到信息区。

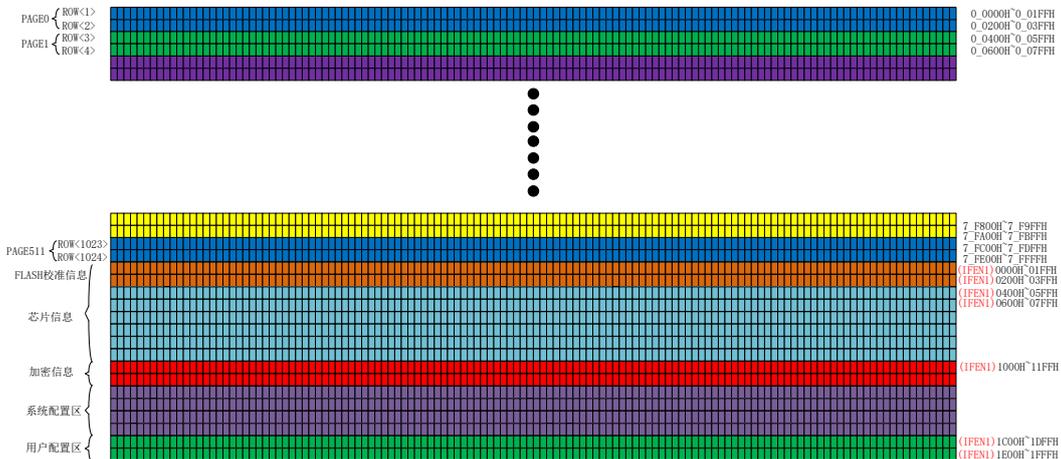


图 34-1 FLASH 的地址映射



注：用户只能使用用户配置区储存数据，写入用户配置区之外的区域，可能导致芯片失效

图 34-2 FLASH 信息区的地址映射

下图为具体的访问方式，IPSEL、IFEN、RECALLEN、CONFEN 为 FLASH_ISPCON0 和 FLASH_ISPCON1 的相关寄存器位。

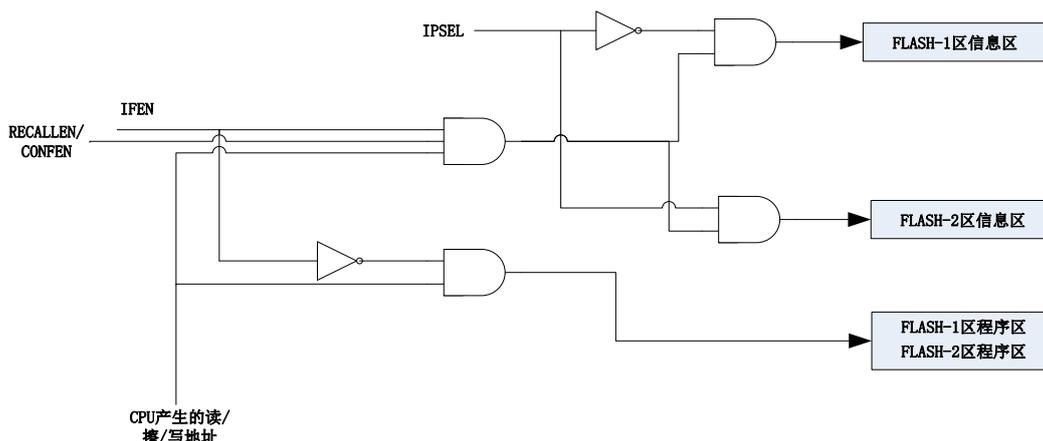


图 34-3 FLASH 访问方式

34.4 信息区内容介绍

FLASH 信息区作为数据存储器，可用于存储启动校准信息。通过将校准值写入对应的位置能够在启动阶段实现校准。校准值对应的寄存器关系如下。

表 34-18 信息区校准值信息

信息区区块	详细地址	校准内容
0x0F00 ~ 0x0FFF	0xF00	OSC_HFOSCCAL0
	0xF04	HFOSCCAL
	0xF08	PM_CAL0
	0xF0C	PM_CAL1
	0xF1C	STCALIB
	其他	保留

34.5 FLASH 读接口

34.5.1 CPU 时钟频率和 FLASH 读时间

FLASH 具有如下特点：

- 64bit 数据位宽
- 可选的预取功能
- 可选的 RECALL 功能
- 可选的 flash 等待时间

FLASH 一次性读出 64 位的数据，可以加速 CPU 的执行速度。FLASH 带可选的预取功能，打开预取后，在顺序执行的时候可以在空闲的时候自动读取下一个区间的数据，提高 flash 的读效率。可选的 RECALL 功能，RECALL 为 0 模式下，为普通模式，此时 flash 可以在较高的速度下操作，当 RECALL 为 1 时，FLASH 读时间会延长，使得 FLASH 读更稳定，通常只有在（上电操作或是其他极端条件下）对信息区操作时使用 RECALL=1 模式。FLASH 等待时间可以在设置读 FLASH 等待时间，当芯片运行在不同频率时，通过设置合适的 FLASH 等待时间，可以提高芯片的运行效率（不同频率下要注意不要低于推荐值，否则可能导致读 flash 异常）。

为使芯片运行达到最优效率，不同条件下 FLASH_CFG 控制寄存器的 TCFG<3:0>配置

如下。

表 34-19 开预取时 FLASH 读时间与系统时钟

SCLK 频率/MHz	开预取: TCFG<3:0>等待周期
0~50	1
50~90	2
90~120	3

表 34-20 关预取时 FLASH 读时间与系统时钟

SCLK 频率/MHz	关预取: TCFG<3:0>等待周期
0~35	0
35~72	1
72~110	2
110~120	3

注意:

- (1) 在 SCLK 频率升高之前, 先配置 FLASH_CFG 控制寄存器; 反之, 在 SCLK 频率降低之后, 再配置 FLASH_CFG 控制寄存器, 以保证可靠的读 FLASH。
- (2) 不同频率下要注意不要低于推荐值, 否则可能会导致 FLASH 读异常。
- (3) 以上均是 RECALL 为 0 的模式下推荐值, 当 RECALL 为 1 时确保 $T_{sclk} * (TCFG + 1) > 80$

34.6 FLASH 加密

为了限制加密模式下 ISP 的命令权限, 而 IAP 仍不受影响。首先通过软件的方式对 ISP 程序进行处理, 在 ISP 部分命令的入口处进行 FLASH 区的加密信息核对, 如果 FLASH 处于加密状态, 则会限制对 FLASH 的读写访问, 擦除命令仍然是有效的。同时出于安全考虑, 加密模式下 (不管何种加密模式) 读存储器的 ISP 代码区域被硬件禁止访问。

在“写加密控制信息”时, 硬件自动对加密字所在 PAGE 进行一次 SERA 擦除和编程。如果加密等级从高到低改变时 (对比 FLASH 总线上的数值和加密信息), 那么硬件自动在页擦和编程之前追加一次片擦, 擦除整个程序区内容。

加密重映射寄存器在一个电源周期内只能写一次, 每修改一次加密信息, 需要重新掉电并上电。

IP 加密 A 模式下, 不支持 ISP 和 IAP 功能对加密区域 (前 64KB) 的擦、写, 以及 ISP 的读; 如果 IP 程序需要修改配置, 可以通过调用 ROM-IAP 程序读、写用户配置区的配置信息, 这样可以不去占用二级用户的程序区资源。

位于 FLASH 的配置区的加密信息用来指示芯片的加密状态。

表 34-21 FLASH 加密状态表

名称	加密信息	ISP 描述	IAP 描述	DPI 描述
不加密	A5A5_5A5A	正常	正常	正常
加密 A	F1A5_0E5A	写 RAM 命令不能访问 0x10000200 以下的 RAM; 将 RAM 复制到 FLASH 命令不能写扇区 0 (可以计算 SIG 值验证写入是否正确); 只有在选择全擦除 MASE 时, 擦除命令才能擦除扇区 0; 禁止读存储器 (IP 加密禁止了全部区域的读, 只能通过 DPI 读出非 IP 加密区域的值); 比较命令禁能; 非扇区 0 的 LD 指令无法访问扇区 0;	正常	DPI 不能读、擦、写扇区 0
加密 C	FFFF_FFFF	只有扇区查空和擦除存储器命令有效	正常	DP 寄存器能正常访问, AP 寄存器无法访问, 无法擦读写 FLASH
加密 C	非 A5A5_5A5A 和 4312_BCDE 以外的值	只有扇区查空和擦除存储器命令有效	正常	DP 寄存器能正常访问, AP 寄存器无法访问, 无法擦读写 FLASH
加密 D	4321_BCDE	禁止 ISP 功能	正常	禁止 DPI 端口

34. 6. 1 调试引脚相关

芯片默认状态下, 如果用户代码占用了调试口, MCU 无法中途进入调试。为了避免用户在不知情的時候, 选择了 NO_ISP 模式, 并且用户代码里又占用了调试口, 芯片将不可通过 ISP 编程且不可调试。

当 MCU 应用于系统需要外部供电进行调试时 (内部供电无法提供足够大的电流), 可以提前 (在应用于目标板之前) 配置芯片配置区的调试信息, 上电后硬件会自动禁用 DPI 调试引脚的其他复用功能, 即使用户代码误占用调试通信口, 也能进行带电调试下载。

表 34-22 调制引脚配置状态表

名称	调试编程信息	ISP 描述	IAP 描述	DPI 描述
NO_ISP	8765_4321	阻止对 ISP_Entry 采样，不进入 ISP 模式；ISP_Entry 引脚做其他用途（用户程序不要占用调试通信口，或者进入超低功耗模式否则无法通过 DPI 调试和编程）	正常	正常
ODMOD	1234_5678	正常	正常	DPI 调试引脚的其他复用功能被禁，其他正常
ODMOD+ NO_ISP	4321_8765	阻止对 ISP_Entry 采样，不进入 ISP 模式；ISP_Entry 做其引脚他用途（用户程序不要进入超低功耗模式，否则无法通过 DPI 调试和编程）	正常	DPI 调试引脚的其他复用功能被禁，其他正常

34. 7 Check_Sum

34. 7. 1 功能说明

FLASH 内部集成一个计算 Check_Sum 的模块。该模块从指定的 FLASH 首尾地址产生一个 128-bit 的 Check_Sum（校验和）。典型的用途就是核对 FLASH 里的内容和计算好的值是否一致。

首尾地址必须和 FLASH-128bit 数据的边界地址对齐。一旦启动 Check_Sum 的计算，FLASH 不能作其他用途（比如编程等），任何读 FLASH 的访问会被延至 Check_Sum 计算完成。启动 Check_Sum 的代码应该放置在 RAM 里。

将首地址写入 FLASH_CSSTART 寄存器，尾地址写入 FLASH_CSSTOP 寄存器，同时将 FLASH_CSSTOP 寄存器的 SIGGO 置“1”以启动计算。计算完成后，128bit 的结果会存入 FLASH_CSRESx(x=0,1,2,3)寄存器里。用户可读取这些寄存器的值判断是否和期望值一致，可以节约时间和代码空间。

计算完成后，FLASH_STATE 的 SIG_DONE 硬件自动置 1，软件判断该位以结束 Check_Sum 过程。启动下一次 Check_Sum 之前，需要向 SIG_DONE 写“1”以清零该位。

34. 7. 2 Check_sum 算法

算法示例如下，flashword 以 128bit 为单元。

```

int128 Checksum= 0
int128 nextChecksum
for address = CheckSum_START 0 toaddress = CheckSum_STOP
{
    for i = 0 to126 {
        nextChecksum[i] = flashword[i] xorChecksum[i+1]}
    next Checksum[127] = flashword[127] xor Checksum[0] xor Checksum[2] xor Checksum[27]
    xor Checksum[29]
    Checksum = nextChecksum
}
return Checksum

```

35 循环冗余校验单元 (CRC)

35.1 概述

循环冗余校验单元 (Cyclic Redundancy Check, CRC) 可以通过生成多项式计算不同长度数据的 CRC 校验值。CRC 技术可应用于核实数据传输或者数据存储的正确性和完整性。

CRC 特性:

- 可编程的多项式，最高支持 33 项数的生成多项式
- 单周期 (最大 20ns) 计算时间
- 支持可编程的初始值
- 支持 8/16/24/32 位长度的输入数据格式
- 输入数据支持字节 (byte) 反序操作
- 计算结果支持可编程的异或操作
- 计算结果支持字 (word) 反序操作

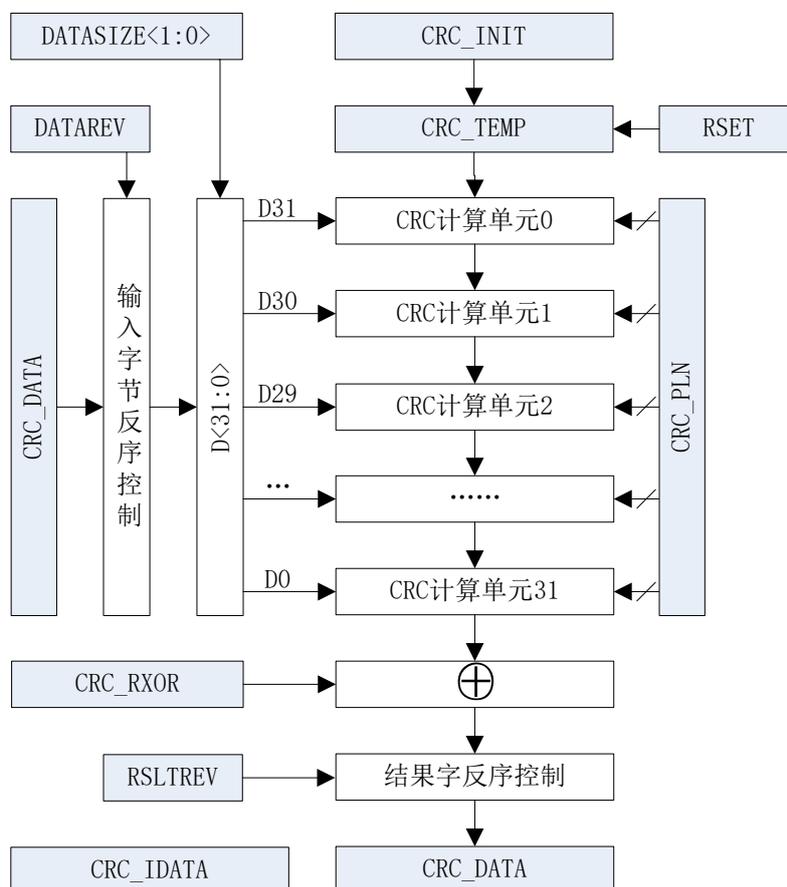


图 35-1 CRC 结构框图

35.2 CRC 相关寄存器

表 35-1 CRC 相关的寄存器

偏移地址	寄存器	访问	功能描述	复位值
0x000	CRC_CTL	R/W	CRC 控制寄存器	0x0000 00AC
0x004	CRC_DATA	R/W	CRC 输入数据寄存器	0x0000 0000
0x008	CRC_RSLT	R	CRC 结果寄存器	0x2144 DF1C
0x00C	CRC_INIT	R/W	CRC 初始值寄存器	0xFFFF FFFF
0x010	CRC_PLN	R/W	CRC 多项式寄存器	0x04C1 1DB7
0x014	CRC_RXOR	R/W	CRC 结果异或值寄存器	0xFFFF FFFF
0x018	CRC_IDATA	R/W	CRC 独立寄存器	0x0000 0000
0x01C	CRC_TEMP	R	CRC 缓存寄存器	0xC704 DD7B

基地址：0x4000 2680

35.2.1 CRC_CTL CRC 控制寄存器 0

表 35-2 CRC_CTL CRC 控制寄存器 0

	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0																						
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																						
R/W																								R/W			R/W																											
位名																								RSLTREV			DATAREV																											RSET

RSLTREV: CRC 结果反序控制位。

- 0 = 无影响
- 1 = CRC 结果位反序

DATAREV: 输入数据位反序控制位。

- 0 = 无位反序操作
- 1 = 按字节为单位进行位反序操作

DATASIZE<1:0>: 输入数据格式控制位。

- 00 = 8 位数据
- 01 = 16 位数据
- 10 = 24 位数据
- 11 = 32 位数据

RSET: CRC 计算单元复位控制位。该位软件置 1 后再对 CRCDATA<31:0>写一个值会复位 CRC 的计算单元，并在一个工作时钟后硬件自动清零。建议在完成其他操作之后最后操作该位。

- 0 = 无作用
- 1 = 复位状态

35.2.2 CRC_DATA CRC 数据寄存器

表 35-3 CRC_DATA CRC 数据寄存器

复位值	R/W	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	0 23	0 22	0 21	0 20	0 19	0 18	0 17	0 16	0 15	0 14	0 13	0 12	0 11	0 10	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
位名	CRCDATA<31:0>																																

CRCDATA<31:0>: CRC 输入数据寄存器。该寄存器用于向 CRC 计算器写入新数据。数据需要右对齐, 数据有效位由 CRC_CTL 寄存器的 DATASIZE<1:0>输入数据格式控制位控制。

35.2.3 CRC_RSLT CRC 结果寄存器

表 35-4 CRC_RSLT CRC 结果寄存器

复位值	R/W	0 31	0 30	1 29	0 28	0 27	0 26	0 25	1 24	0 23	1 22	0 21	0 20	0 19	1 18	0 17	0 16	1 15	1 14	0 13	1 12	1 11	1 10	1 9	1 8	0 7	0 6	0 5	1 4	1 3	1 2	0 1	0 0
位名	CRCRSLT<31:0>																																

CRCRSLT<31:0>: CRC 结果寄存器。读该寄存器可获得 CRC 计算结果。该寄存器为只读。CRC 结果有效位与使用的多项式相同。结果若未反序, 则结果有效位左对齐; 结果若反序, 则结果有效位右对齐。

35.2.4 CRC_INIT CRC 初始值寄存器

表 35-5 CRC_INIT CRC 初始值寄存器

复位值	R/W	1 31	1 30	1 29	1 28	1 27	1 26	1 25	1 24	1 23	1 22	1 21	1 20	1 19	1 18	1 17	1 16	1 15	1 14	1 13	1 12	1 11	1 10	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0
位名	CRCINIT<31:0>																																

CRCINIT<31:0>: CRC 初始值寄存器。该寄存器可设置 CRC 计算的初始值。CRC 计算过程中不会改变该值, 使用 CRC_CTL 的 RSET 复位后, 写 CRCDATA<31:0>操作会重新将该值加载到 CRC 计算单元中。该寄存器值需要左对齐, 有效位长度与 CRC_PLN 一致。常见多项式初始值可参考“常见多项式控制”章节。

35.2.5 CRC_PLN CRC 多项式寄存器

表 35-6 CRC_PLN CRC 多项式寄存器

复位值	R/W	0 31	0 30	0 29	0 28	0 27	0 26	0 25	0 24	1 23	1 22	0 21	0 20	0 19	0 18	0 17	1 16	0 15	0 14	0 13	1 12	1 11	1 10	0 9	1 8	1 7	0 6	1 5	1 4	0 3	1 2	1 1	1 0
位名	CRCPLN<31:0>																																

CRCPLN<31:0>: CRC 多项式寄存器。该寄存器可设置 CRC 多项式。该寄存器每一位都对应一个多项式的系数。多项式需要左对齐, 最高项忽略, 默认为 1, 当多项式少于 32 项时, 左对齐即可, 多项式最低项必须为 1。如 CRC-32 项多项式, 其中 bit31 对应 X^{31} 系数, bit0 对应 X^0 系数, 最高项忽略, 默认为 1, 以此类推。

默认多项式系数为 CRC-32 多项式：0x04C11DB7。

35.2.6 CRC_RXOR CRC 结果异或值寄存器

表 35-7 CRC_RXOR CRC 结果异或值寄存器

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CRCRXOR<31:0>																																

CRCRXOR<31:0>: CRC 结果异或值寄存器。该寄存器可设置 CRC 计算结果的异或运算值。
注意该寄存器为左对齐，有效位长度与 CRC_PLN 有效位长度一致。

35.2.7 CRC_IDATA CRC 独立数据寄存器

表 35-8 CRC_IDATA CRC 独立数据寄存器

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位名	CRCIDATA<31:0>																															

CRCIDATA<31:0>: CRC 独立数据寄存器。该寄存器可用于临时存放 32 位的数据。CRC_CTL 寄存器中的 RSET 位的复位不会影响该寄存器的值。

35.2.8 CRC_TEMP CRC 缓存寄存器

表 35-9 CRC_TEMP CRC 缓存寄存器

复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位名	CRCTEMP<31:0>																															

CRCTEMP<31:0>: CR 缓存寄存器。该寄存器为只读。该寄存器用于 CRC 计算的中间结果缓存，当需要超过 32 位数据进行 CRC 运算时，该寄存器可以确保只需要连续写入数据即可完成 CRC 运算。当 CRC_CTL 寄存器的 RSET 位置 1 时，CRC_INIT<31:0>的值会缓存到该寄存器中，当对 CRC_DATA<31:0>写操作时，该寄存器值会将 CRC 运算的中间值更新到其中。

35.3 CRC 操作

CRC 为单周期计算，需要时间约 20ns。通过 CRC_DATA 可以写入需要进行 CRC 计算的数据，通过 CRC_INIT 可以设置计算的初始值，通过 CRC_PLN 可以设置 CRC 的生成多项式，通过 CRC_RXOR 可以设置结果的异或值，通过 CRC_CTL 寄存器可以设置输入字节反序、输入数据有效位、输出结果反序和 CRC 计算单元复位控制。CRC 计算一个周期即可完成计算。

注意，将 RSET 置 1 后，再对 CRCDATA<31:0>写操作会更新 CRC 计算的初始值，以此可以复位 CRC 的计算，RSET 在对 CRCDATA<31:0>写操作后硬件自动清零。

当需要输入的数据多于 32 位时，只要在不重新置位 RSET 的条件下，继续向 CRC_DATA 输入数据写入值即可继续上次的运算结果继续计算，此时硬件会将上次的计算结果作为初始值参与到下一次的运算中。通过 CRC_TEMP<31:0>可以读出 CRC 运算的中间结果

通过 CRC_RSLT 寄存器可以读出结算结果。结果的有效位为多项式的有效位减 1 项，最后需要根据有效位来读取最后结果。

35.4 CRC 多项式操作

通过 CRC_PLN 多项式寄存器可以灵活配置生成多项式，默认为 CRC-32（以太网）多项式：0x04C11DB7。CRC_PLN 寄存器对应多项式中的系数，为 1 时表明存在该位对应权值的多项式项，其中 bit31 对应 X^{31} 系数，bit0 对应 X^0 系数，以此类推。

例如：CRC-4/ITU 多项式为 x^4+x+1 ，即 $1 \times x^4 + 0 \times x^3 + 0 \times x^2 + 1 \times x^1 + 1 \times x^0$ ，多项式系数为 10011，写入 CRC_PLN 寄存器时，最高位忽略（最高位默认为 1），即需要将 0011 写入 CRC_PLN 寄存器。该寄存器需左对齐，即将 0011 0000 0000 0000 0000 0000 0000（即 0x3000 0000）写入 CRC_PLN 寄存器。

支持任意项的多项式，当多项式少于 32 项时，左对齐，且多项式最低有效位必须为 1。在配置多项式中，最高位的 1 忽略掉。需要注意的是，由于 CRC 结果位数等于有效多项式项，因此选择不同的多项式时，最后的结果也应根据不同的数据长度读取数据。

表 35-10 常见多项式及对应的 CRC_PLN 寄存器值

CRC 类型	多项式	对应寄存器值
CRC-4/ITU	x^4+x+1	0x3000 0000
CRC-5/EPC	x^5+x^3+1	0x4800 0000
CRC-5/ITU	$x^5+x^4+x^2+1$	0xA800 0000
CRC-5/USB	x^5+x^2+1	0x2800 0000
CRC-6/ITU	x^6+x+1	0x0C00 0000
CRC-7/MMC	x^7+x^3+1	0x1200 0000
CRC-8 CRC-8/ITU CRC-8/ROHC	x^8+x^2+x+1	0x0700 0000
CRC-8/MAXIM	$x^8+x^5+x^4+1$	0x3100 0000
CRC-16/IBM CRC-16/ MAXIM CRC-16/ USB CRC-16/ MODBUS	$x^{16}+x^{15}+x^2+1$	0x8005 0000

CRC 类型	多项式	对应寄存器值
CRC-16/ CCITT CRC-16/ X25 CRC-16/ XMODEM	$x^{16}+x^{12}+x^5+1$	0x1021 0000
CRC-16/ DNP	$x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^8+x^6+x^5+x^2+1$	0x3D65 0000
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$	0x04C1 1DB7

35.5 CRC 输入数据

35.5.1 输入数据格式

CRC 支持 8/16/24/32 位数据输入，通过 CRC_CTL 的 DATASIZE<1:0>控制。支持输入数据的有效长度小于多项式的有效长度。

35.5.2 反序操作

输入数据可以根据不同的 CRC 计算需要，进行字节反序操作。通过 DATAREV 控制可以实现字节为单位进行位反序操作。

表 35-11 DATAREV 位反序操作示意

DATAREV	输入	反序结果
0	0x1A2B 3C4D	0x1A2B 3C4D
1	0x1A2B 3C4D	0x58D4 3CB2

35.6 CRC 运算结果反序操作

CRC 结果输出也可以通过 RSLTREV 实现字为单位进行反序或不反序。

表 35-12 RSLTREV 位反序操作示意

RSLTREV	输入	反序结果
0	0x1A2B 3C4D	0x1A2B 3C4D
1	0x1A2B 3C4D	0xB23C D458

35.7 CRC 运算初始值设置

通过 CRC_INIT 初始值寄存器可以设置 CRC 运算的初始值。该寄存器可用于下列情况：

- 用于不同的 CRC 需要不同的初始值的情况。如全 0、全 1 或者其他特殊值的情况。
- 用于 CRC 复位后仍可以继续之前的 CRC 运算结果运算的情况。此时需要手动将上一次的结果写入 CRC_INIT 寄存器中作为初始值。
- CRC 运算过程中，会将中间结果缓存下来，当需要进行多数据计算时直接更新数据即可，如需要重新开始计算，需要将 CRC_RSET 置 1 复位到初始状态，复位后会将初始值加载到 CRC 计算单元中。

CRC 计算过程中不会改变 CRC_INIT 的值，使用 CRC_CTL 的 RSET 复位后，会重新将该值加载到 CRC 计算单元中。CRC_INIT 需要左对齐，有效位长度与 CRC_PLN 一致。如多项式 CRC-5/EPC 的初始值为 0x9，由于该多项式有效位是 6 位，所以数据和初始值有效

位是 6-1=5 位，为 0x01001，初始值需要左对齐，即为 0x4800 0000。非有效位初始值可以为任意值。

35.8 CRC 独立寄存器

CRC_IDATA 为独立寄存器，可用于暂存 32 位的数据。当 CRC_RSET 置 1 复位 CRC 计算单元时，该位不受影响。该位受 CRC 模块的复位的影响清零。

35.9 常见多项式控制

这里列举了常见 CRC 运算的多项式及数据处理要求。

表 35-13 常见多项式控制¹

CRC 类型	多项式值	初始值 ²	输入反序	结果反序	结果异或值
CRC-4/ITU	0x3000 0000	0x0000 0000	T	T	0x0000 0000
CRC-5/EPC	0x4800 0000	0x4800 0000	F	F	0x0000 0000
CRC-5/ITU	0xA800 0000	0x0000 0000	T	T	0x0000 0000
CRC-5/USB	0x2800 0000	0xFFFF FFFF	T	T	0xFFFF FFFF
CRC-6/ITU	0x0C00 0000	0x0000 0000	T	T	0x0000 0000
CRC-7/MMC	0x1200 0000	0x0000 0000	F	F	0x0000 0000
CRC-8	0x0700 0000	0x0000 0000	F	F	0x0000 0000
CRC-8/ITU	0x0700 0000	0x0000 0000	F	F	0x5500 0000
CRC-8/ROHC	0x0700 0000	0xFFFF FFFF	T	T	0x0000 0000
CRC-8/MAXIM	0x3100 0000	0x0000 0000	T	T	0x0000 0000
CRC-16/IBM	0x8005 0000	0x0000 0000	T	T	0x0000 0000
CRC-16/MAXIM	0x8005 0000	0x0000 0000	T	T	0xFFFF FFFF
CRC-16/USB	0x8005 0000	0xFFFF FFFF	T	T	0xFFFF FFFF
CRC-16/MODBUS	0x8005 0000	0xFFFF FFFF	T	T	0x0000 0000
CRC-16/CCITT	0x1021 0000	0x0000 0000	T	T	0x0000 0000
CRC-16/CCITT	0x1021 0000	0xFFFF FFFF	F	F	0x0000 0000
CRC-16/X25	0x1021 0000	0xFFFF FFFF	T	T	0xFFFF FFFF
CRC-16/XMODEM	0x1021 0000	0x0000 0000	F	F	0x0000 0000
CRC-16/DNP	0x3D65 0000	0x0000 0000	T	T	0xFFFF FFFF
CRC-32	0x04C1 1DB7	0xFFFF FFFF	T	T	0xFFFF FFFF
CRC-32/MPEG-2	0x04C1 1DB7	0xFFFF FFFF	F	F	0x0000 0000

35.10 CRC 操作步骤

CRC 模块可按照下面的步骤进行操作：

- (1) 配置 CRC_PLN 寄存器，设置需要的多项式值，注意该值为左对齐，忽略最高位；
- (2) 配置 CRC_INIT 初始值（可参考常见“常见多项式控制”章节），如果需要一个中间值开始计算请将中间值写入 CRC_INIT，该寄存器为左对齐，有效位长度与

¹ 表中 T 表示 True，F 表示 False。

² 为方便操作，示例初始值能用全 0 或全 1 的这里尽可能使用同样的值。

CRC_PLN 一致:

- (3) 配置 CRC_RXOR 结果异或值, 该寄存器为左对齐, 有效位长度与 CRC_PLN 一致;
- (4) 配置 CRC_CTL 寄存器, 通过 CRC_DATAREV 设置输入数据反序操作, 通过 CRC_RSLTREV 设置输出数据反序操作, 通过 CRC_DATASIZE<1:0>设置输入数据的有效位, 将 RSET 置 1 初始化 CRC_TEMP; 需要注意的是如果不是进行长数据的计算, 需要在这个步骤时将 RSET 置 1。RSET 置 1 后写 CRC_DATA 寄存器操作会初始化 CRC_TEMP, 之后 RSET 位硬件自动清零;
- (5) 配置 CRC_DATA 寄存器, 写入需要进行计算的数据, 数据为右对齐, 在 RSET 为 1 时写该寄存器操作时会将初始值加载到 CRC_TEMP 中进行计算, 在 RSET 为 0 时会将当前 CRC 运算的中间结果更新到 CRC_TEMP 中;
- (6) 如果需要进行长数据的 CRC 计算, 只需要继续修改 CRC_DATA 寄存器即可。

CRC 为单周期计算, 最长需要 20ns, 如果计算出错可以插入 NOP 指令增加计算周期, 通过 CRC_RSLT 可以读出 CRC 的运算结果, 通过 CRC_TEMP 可以读出 CRC 计算的中间结果, 可以软件将 CRC_TEMP 保存下来, 在需要的时刻更新到 CRC_INIT 中继续当前数据的运算。

36 封装信息

36.1 QFN32 封装

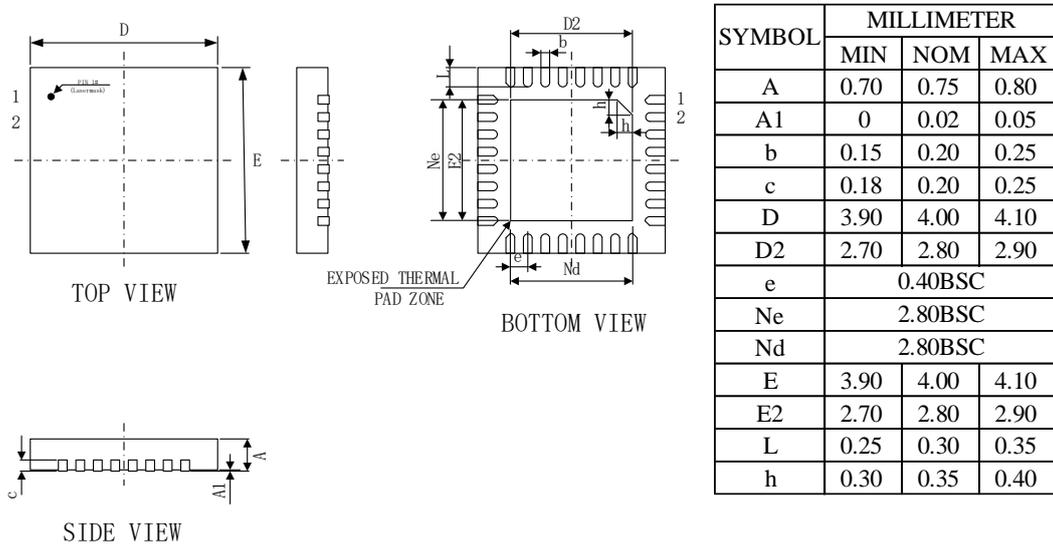


图 36-1 QFN32 封装图

36.2 48 脚 LQFP 封装

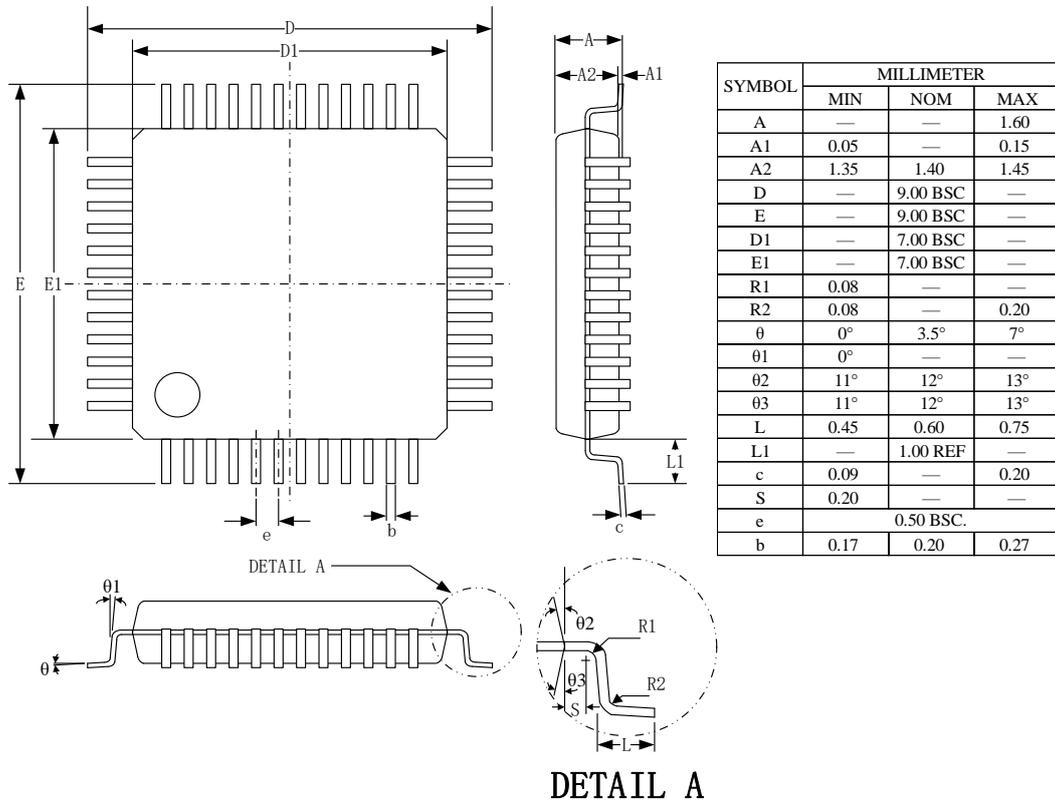


图 36-2 LQFP48 封装图

36.3 64 脚 LQFP 封装

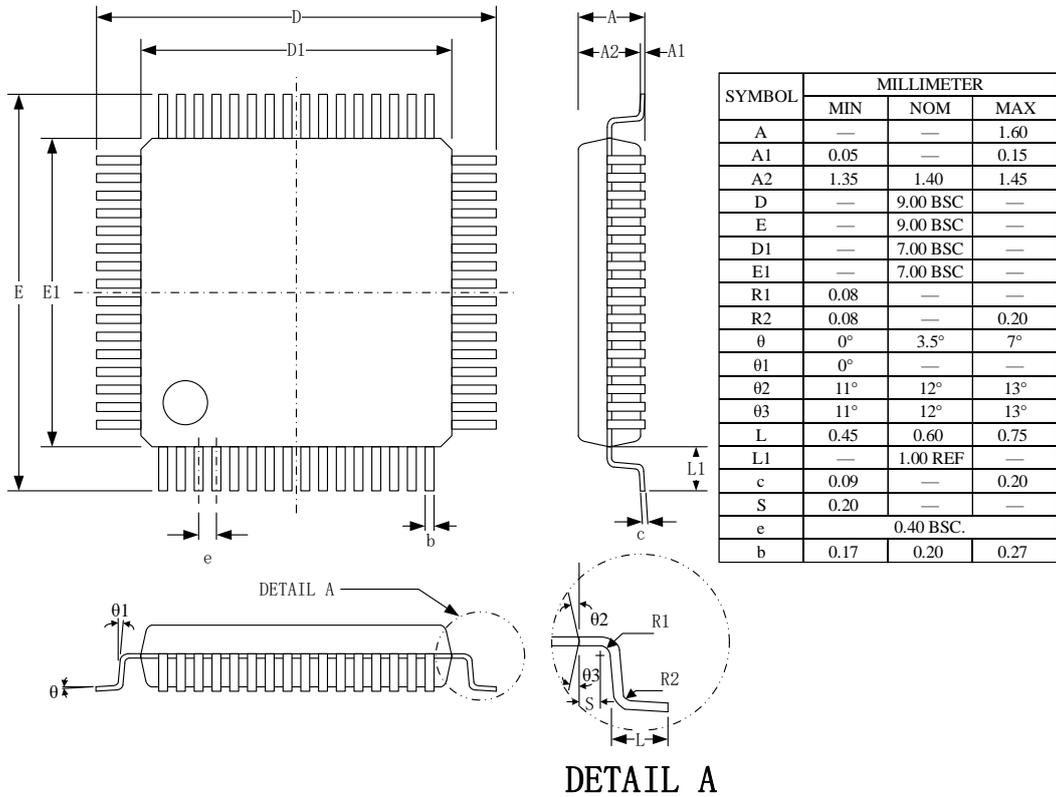


图 36-3 LQFP64 封装图 (7.0mm*7.0mm)

36.4 64 脚 LQFP64 封装 (10.0mm*10.0mm)

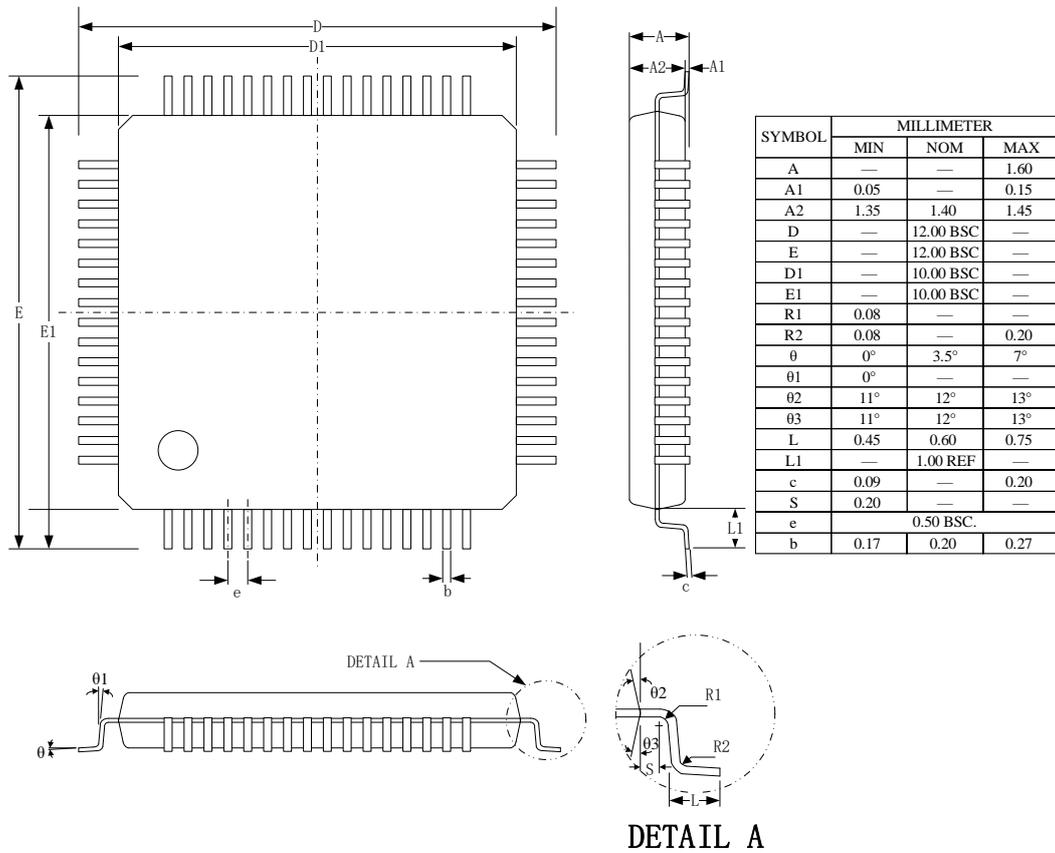


图 36-4 LQFP64 封装图 (10.0mm*10.0mm)

36.5 100 脚 LQFP 封装

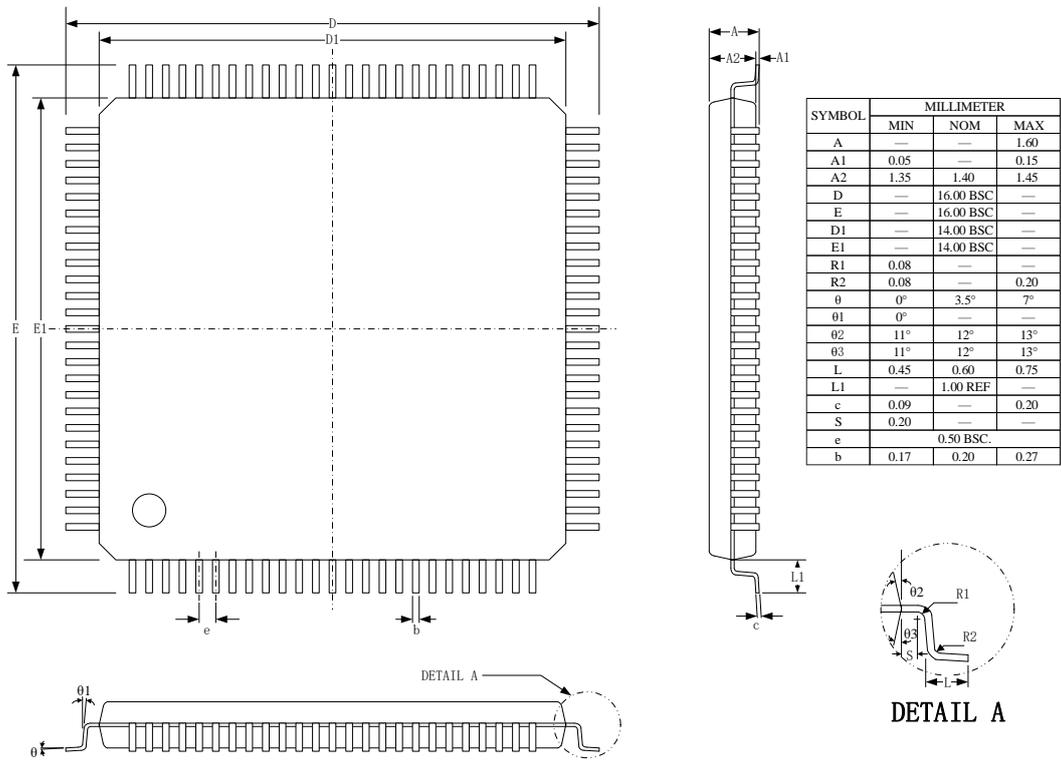
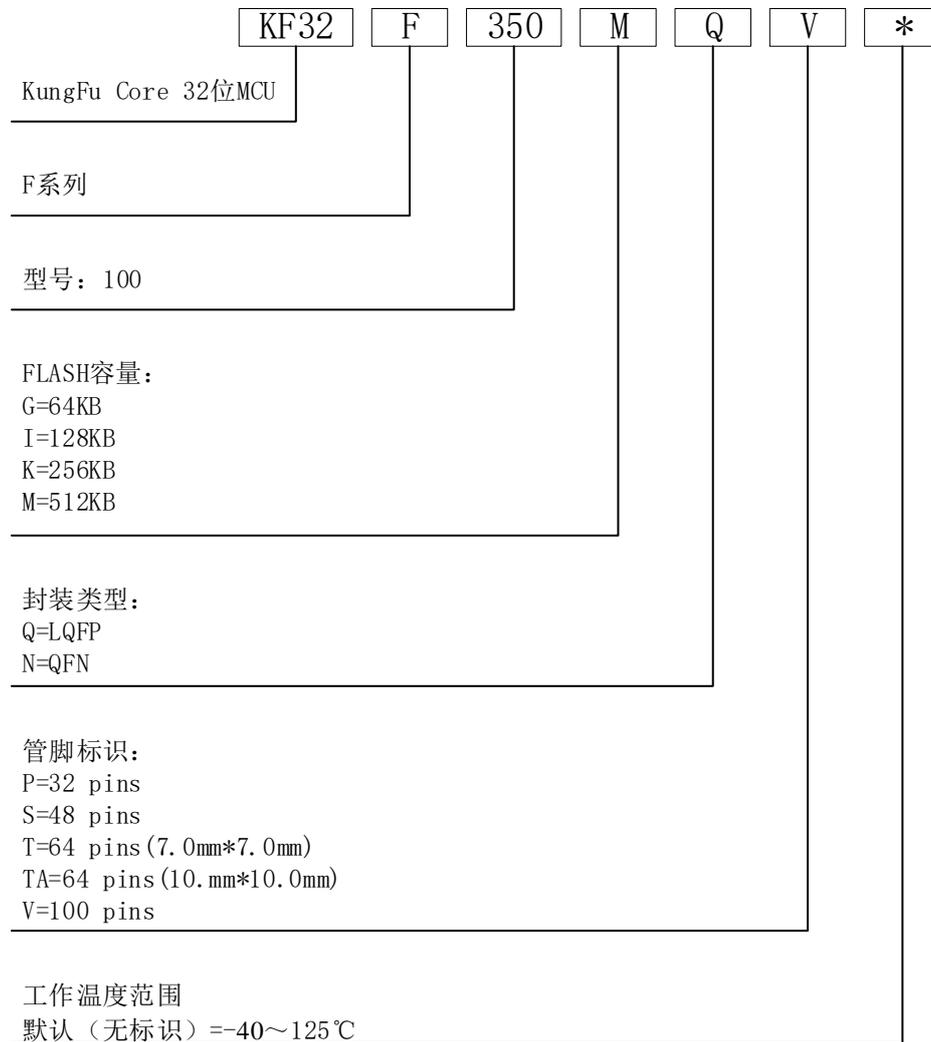


图 36-5 LQFP100 封装图

37 产品标识体系



38 声明及销售网络

销售及服务网点

上海 TEL: 021-50275927

地 址：上海浦东龙东大道 3000 号张江集电港 1 幢 906 号 B 座

附录 1 汇编指令集

汇编语法：描述汇编指令的格式。

指令说明：使用操作符号描述指令执行的操作，符号如下

= : 结果赋值

== : 逻辑等

!= : 不等于

+ : 加运算

- : 减运算

NOT: 按位取反操作

& : 与操作

| : 或操作

^ : 异或操作

<< : 逻辑左移

[] : 寄存器寻址

.H : 对高 16 位操作

.L : 对低 16 位操作

.B : 对低 4 位操作

ZeroExtend(imm, "numb", 32):

将 imm 低位补 numb, 并零拓展成 32 位, numb 可无。

SignExtend(imm, "numb", 32):

将 imm 低位补 numb, 并符号拓展成 32 位, numb 可无。

{list}: 分别对应 LR/PC,R12-R6, 当相应位置 1 时, 表示选中相应的 Rx。

<c>: cond, 条件执行编码, 按照 N/Z/C/V 标志位的值来判断是否执行。条件执行编码表见表 1.2.1 条件执行编码表

labelx: x 位相对 PC 的偏移地址

offsetx: x 位偏移地址

bit5: 5 位操作数, 指示第 0 到第 31 位。

immx: x 位立即数

逻辑左移: 左移, 低位补 0。

逻辑右移: 右移, 高位补 0

算术右移: 右移, 高位补符号位。

循环右移: 右移, 低位移出位补高位。

字对齐: 地址的低 2 位无效。

半字对齐: 地址的最低位无效。

字节对齐: 所有地址均有效。

受影响标志位：描述指令执行影响的标志位 (N/Z/C/V), 不涉及的标志位会保持。

影响标志位的具体操作如下:

N: 当结果最高位为 1 (即结果为负数时), 使 N 置 1, 否则清零。

Z: 当结果为 0, 使 Z 置 1, 否则清零。

C: 当结果的最高位仍有进位时, 使 C 置 1, 否则清零。

V: 当结果溢出时 (最高位和次高位进位异或为 1), 将 V 置 1, 否则清零。

指令不涉及的标志位会保持。

执行周期：描述指令执行阶段需要的周期数

条件执行编码

跳转指令 JMP<c>可以根据 N/Z/C/V 标志位来决定是否转型跳转，具体如下表所示：

附表 1 条件执行编码表

<c>	助记简写	助记全写	执行条件
0000	Z	Zero	Z==1
0001	NZ	Not Zero	Z==0
0010	C	Carry	C==1
0011	NC	No Carry	C==0
0100	MI	Minus,negative	N==1
0101	PL	Plus,positive or zero	N==0
0110	VS	Overflow	V==1
0111	VC	No Overflow	V==0
1000	HI	Unsigned Higher	C==1 & Z==0
1001	LS	Unsigned Lower or Same	C==0 Z==1
1010	GE	Signed Greater or Equal	N==V
1011	LT	Signed Less Than	N!=V
1100	GT	Signed Greater Than	Z==0 & N==V
1101	LE	Signed Less or Equal	Z==1 N!=V

传送指令

附表 2 传送指令

汇编语法	说明	影响标志位	周期数
MOV Rd,#imm8	将 8 位立即数送到 Rd	N/Z	1
MOV Rd,Rs	将 Rs 中内容送到 Rd		1
MOVL Rd,#imm16	将 16 位立即数送到 Rd, 高半字清零		2
MOVH Rd,#imm16	将 16 位立即数送到 Rd 高半字, 低半字内容不变		2
MOV SYS,Rs	将 Rs 中内容送到 MSP 或 PSP		1
MOV Rd,SYS	将 MSP 或 PSP 中内容送到 Rs		1
XCH Rd,Rs	将 Rd 和 Rs 中内容互换		2

存取指令

附表 3 存取指令

汇编语法	说明	影响标志位	周期数
LD.W Rd,[PC+#offset8]	将[PC+{#offset8,"00"}]指向的 32 位内容传送到 Rd		2
LD.B Rd,[sp+#offset5]	将[sp+#offset5]指向的 8 位内容零扩展成 32 位, 再传送到 Rd		2
LD.H Rd,[sp+#offset5]	将[sp+{#offset5,"0"}]指向的 16 位内容零扩展成 32 位, 再传送到 Rd		2
LD.W Rd,[sp+#offset8]	将[sp+{#offset8,"00"}]指向的 32 位内容传送到 Rd		2
LD.B Rd,[Rs]	将[Rs]指向的 8 位内容零扩展成 32 位, 再传送到 Rd		2
LD.H Rd,[Rs]	将[Rs]指向的 16 位内容零扩展成 32 位, 再传送到 Rd		2
LD.W Rd,[Rs]	将[Rs]指向的 32 位内容传送到 Rd		2
LD.B Rd,[Rs++]	将[Rs]指向的 8 位内容零扩展成 32 位, 再传送到 Rd, Rs 加 1		2
LD.H Rd,[Rs++]	将[Rs]指向的 16 位内容零扩展成 32 位, 再传送到 Rd, Rs 加 2		2
LD.W Rd,[Rs++]	将[Rs]指向的 32 位内容传送到 Rd, Rs 加 4		2
LD.B Rd,[Rs+#offset5]	将[Rs+#offset5]指向的 8 位内容零扩展成 32 位, 再传送到 Rd		2
LD.H Rd,[Rs+#offset5]	将[Rs+{#offset5,"0"}]指向的 16 位内容零扩展成 32 位, 再传送到 Rd		2
LD.W Rd,[Rs+#offset5]	将[Rs+{offset5,"00"}]指向的 32 位内容传送到 Rd		2
LD.B Rd,[Rt+Rs]	将[Rt+Rs]指向的 8 位内容零扩展成 32, 再传送到 Rd		2
LD.H Rd,[Rt+Rs]	将[Rt+Rs]指向的 16 位内容零扩展成 32 位, 再传送到 Rd		2
LD.W Rd,[Rt+Rs]	将[Rt+Rs]指向的 32 位内容传送到 Rd		2
LDS.B Rd,[Rs]	将[Rs]指向的 8 位内容符号扩展成 32 位, 再传送到 Rd		2
LDS.H Rd,[Rs]	将[Rs]指向的 16 位内容符号扩展成 32 位, 再传送到 Rd		2
LDP.B Rd,[Rs]<<#imm2	将[Rs]指向的 8 位内容零扩展成 32 位, 再左移 imm2 位后传送到 Rd		2
LDP.H Rd,[Rs]<<#imm2	将[Rs]指向的 16 位内容零扩展成 32 位, 再左移 imm2 位后传送到 Rd		2

汇编语法	说明	影响标志位	周期数
LDP.W Rd,[Rs]<<#imm2	将[Rs]指向的 32 位内容左移 imm2 位,再传送到 Rd		2
ST.B [sp+#offset5],Rs	将 Rs 的低 8 位内容传送到[sp+#offset5]对应地址,其他地址不变		2
ST.H [sp+#offset5],Rs	将 Rs 的低 16 位内容传送到[sp+{#offset5,"0"}]对应地址,其他地址不变		2
ST.W [sp+#offset8],Rs	将 Rs 的 32 位内容传送到[sp+{#offset8,"00"}]对应地址		2
ST.B [Rt],Rs	将 Rs 的低 8 位内容传送到[Rs]对应地址,其他地址不变		2
ST.H [Rt],Rs	将 Rs 的低 16 位内容传送到[Rs]对应地址,其他地址不变		2
ST.W [Rt],Rs	将 Rs 的 32 位内容传送到[Rt]对应地址		2
ST.B [Rt++],Rs	将 Rs 的低 8 位内容传送到[Rs]对应地址,其他地址不变,Rt 加 1		2
ST.H [Rt++],Rs	将 Rs 的低 16 位内容传送到[Rs]对应地址,其他地址不变,Rt 加 2		2
ST.W [Rt++],Rs	将 Rs 的 32 位内容传送到[Rs]对应地址,Rt 加 4		2
ST.B [Rt+#offset5],Rs	将 Rs 的低 8 位内容传送到[Rt+#offset5]对应地址,其他地址不变		2
ST.H [Rt+#offset5],Rt	将 Rs 的低 16 位内容传送到[Rt+{#offset5,"0"}]对应地址,其他地址不变		2
ST.W [Rt+#offset5],Rs	将 Rs 的 32 位内容传送到[Rt+{#offset5,"00"}]对应地址		2
ST.B [Rt+Ra],Rs	将 Rs 的低 8 位内容传送到[Rt+Ra]对应地址,其他地址不变		2
ST.H [Rt+Ra],Rs	将 Rs 的低 16 位内容传送到[Rt+Ra]对应地址,其他地址不变		2
ST.W [Rt+Ra],Rs	将 Rs 的 32 位内容传送到[Rt+Ra]对应地址		2
STP.B [Rt],Rs <<#imm2	将 Rs 左移 imm2 位的低 8 位内容传送到[Rs]对应地址,其他地址不变		2
STP.H [Rt],Rs<<#imm2	将 Rs 左移 imm2 的低 16 位内容传送到[Rs]对应地址,其他地址不变		2
STP.W [Rt],Rs<<#imm2	将 Rs 左移 imm2 的 32 位内容传送到[Rs]对应地址		2

数学指令

附表 4 数学指令

汇编语法	说明	影响标志位	周期数
ADD Rd,#imm7	$Rd = Rd + imm7$	N/Z/C/V	1
ADD Rd,Rs	$Rd = Rd + Rs$	N/Z/C/V	1
ADDC Rd,Rs	$Rd = Rd + Rs + C$	N/Z/C/V	1
ADD Rd,Rs,#imm3	$Rd = Rs + imm3$	N/Z/C/V	1
ADD Rd,Rt,Rs	$Rd = Rt + Rs$	N/Z/C/V	1
ADDC Rd,Rt,Rs	$Rd = Rt + Rs + C$	N/Z/C/V	1
DIVS Rd,Rt,Rs	有符号除法, $Rd = Rt / Rs$		2-12
DIVU Rd,Rt,Rs	无符号除法, $Rd = Rt / Rs$		2-12
MULS Rd,Rt,Rs	有符号乘法, $Rd = Rt * Rs$	N/Z	1
SUB Rd,#imm7	$Rd = Rd - imm7$	N/Z/C/V	1
SUB Rd, Rs	$Rd = Rd - Rs$	N/Z/C/V	1
SUBC Rd,Rs	$Rd = Rd + NOT(Rs) + C$	N/Z/C/V	1
SUB Rd,Rs,#imm3	$Rd = Rs - imm3$	N/Z/C/V	1
SUB Rd,Rt,Rs	$Rd = Rt - Rs$	N/Z/C/V	1
SUBC Rd,Rt,Rs	$Rd = Rt + NOT(Rs) + C$	N/Z/C/V	1
SXT.B Rd,Rs	将 Rs 的低 8 位内容符号位扩展成 32 位, 结果保存到 Rd 中		1
SXT.H Rd,Rs	将 Rs 的低 16 位内容符号扩展成 32 位, 结果保存到 Rd 中		1
ZXT.B Rd,Rs	将 Rs 的低 8 位内容零扩展成 32 位, 结果保存到 Rd 中		1
ZXT.H Rd,Rs	将 Rs 的低 16 位内容零位扩展成 32 位, 结果保存到 Rd 中		1

逻辑指令

附表 5 逻辑指令

汇编语法	说明	影响标志位	周期数
ANL Rd,Rs	$Rd = Rd \& Rs$	N/Z	1
ANL Rd,Rt,Rs	$Rd = Rt \& Rs$	N/Z	1
NOT Rd,Rs	$Rd = \sim Rs$	N/Z	1
ORL Rd,Rs	$Rd = Rd Rs$	N/Z	1
ORL Rd,Rt,Rs	$Rd = Rt Rs$	N/Z	1
XRL Rd,Rs	$Rd = Rd \wedge Rs$	N/Z	1
XRL Rd,Rt,Rs	$Rd = Rt \wedge Rs$	N/Z	1

位操作指令

附表 6 位操作指令

汇编语法	说明	影响标志位	周期数
CLR Rd,#bit5	将 Rd 的 bit5 位清零，其他位不变	N/Z	1
CLR [Rs],#bit5	将[Rd]的 bit5 位清零，其他位不变	N/Z	2
CLR PSW,#bit5	将 PSW 的 bit5 位清零，其他位不变	N/Z/C/V	1
DSI	关闭中断总使能，AIE = 0		1
ENI	使能中断总使能，AIE = 1		1
REV Rd, Rs	Rs 位反序后写入 Rd		1
SET Rd, #bit5	将 Rd 的 bit5 位置 1，其他位不变	N/Z	1
SET [Rt], #bit5	将[Rd]的 bit5 位置 1，其他位不变	N/Z	2
SET PSW, #bit5	将 PSW 的 bit5 位置 1，其他位不变	N/Z/C/V	1

循环/移位指令

附表 7 循环/位移指令

汇编语法	说明	影响标志位	周期数
ASR Rd,#imm5	Rd = Rd 算术右移 imm5 位	N/Z/C	1
ASR Rd,Rs	Rd = Rd 算术右移 Rs<3:0>位	N/Z/C	1
ASR Rd,Rt,Rs	Rd = Rt 算术右移 Rs<2:0>位	N/Z/C	1
LSL Rd,#imm5	Rd = Rd 逻辑左移 imm5 位	N/Z/C	1
LSL Rd,Rs	Rd = Rd 逻辑左移 Rs<3:0>位	N/Z/C	1
LSL Rd,Rt,Rs	Rd = Rt 逻辑左移 Rs<2:0>位	N/Z/C	1
LSR Rd,#imm5	Rd = Rd 逻辑右移 imm5 位	N/Z/C	1
LSR Rd,Rs	Rd = Rd 逻辑右移 Rs<3:0>位	N/Z/C	1
LSR Rd,Rt,Rs	Rd = Rt 逻辑右移 Rs<2:0>位	N/Z/C	1
ROR Rd, #imm5	Rd = Rd 循环右移 imm5 位	N/Z/C	1
ROR Rd, Rs	Rd = Rd 循环右移 Rs<3:0>位	N/Z/C	1

比较/跳过指令

附表 8 比较/跳过指令

汇编语法	说明	影响标志位	周期数
CMN Rt,Rs	比较 Rt + Rs	N/Z/C/V	1
CMP Rt,#imm7	比较 Rt - imm7	N/Z/C/V	1
CMP Rt,Rs	比较 Rt - Rs	N/Z/C/V	1
JB Rt,#bit5	Rt 的 bit5 位为 1 跳过，否则顺序执行		2/4
JB [Rt],#bit5	[Rt]的 bit5 位为 1 跳过，否则顺序执行		2/4
JNB Rt,#bit5	Rt 的 bit5 位为 0 跳过，否则顺序执行		2/4
JNB [Rt],#bit5	[Rt]的 bit5 位为 0 跳过，否则顺序执行		2/4
TST Rt,Rs	Rt & Rs	N/Z	1

注：如果没有跳过时条件跳过指令执行周期为 2，跳过时执行周期为 4

程序流指令

附表 9 程序流指令

汇编语法	说明	影响标志位	周期数
JZ label8	Z=1 跳转，否则顺序执行		1/3
JNZ label8	Z=0 跳转，否则顺序执行		1/3
JC label8	C=1 跳转，否则顺序执行		1/3
JNC label8	C=0 跳转，否则顺序执行		1/3
JMI label8	N=1 跳转，否则顺序执行		1/3
JPL label8	N=0 跳转，否则顺序执行		1/3
JVS label8	V=1 跳转，否则顺序执行		1/3
JVC label8	V=0 跳转，否则顺序执行		1/3
JHI label8	C=1 且 Z=0 跳转，否则顺序执行		1/3
JLS label8	C=0 或 Z=1 跳转，否则顺序执行		1/3
JGE label8	N=V 跳转，否则顺序执行		1/3
JLT label8	N!=V 跳转，否则顺序执行		1/3
JGT label8	Z=0 且 N=V 跳转，否则顺序执行		1/3
JLE label8	Z=1 或 N!=V 跳转，否则顺序执行		1/3
JMP label24	跳转到 PC+label24 的地址		3
JMP Rs	跳转到 Rs 指向的地址		3
LJMP Rs	跳转到 Rs 指向的地址并保存 PC		3
LJMP label8	跳转到 PC+label8 的地址并保存 PC		3
LJMP label21	跳转到 PC+label21 的地址并保存 PC		3
SJMP label10	跳转到 PC+label10 的地址		3

注：如果条件不满足未发生转移时，条件跳转指令执行周期为 1 周期，发生转移时执行周期为 3 周期

堆栈/出栈指令

附表 10 堆栈/出栈指令

汇编语法	说明	影响标志位	周期数
POP {list}	将堆栈内容压出到 list 列表中为 1 的对应寄存器		n+1/n+4
POP Rd	将堆栈内容压出到 Rd		2
PUSH {list}	将 list 列表中为 1 的对应寄存器压入堆栈		n+1
PUSH Rs	将 Rs 内容压入堆栈		2

注：n 为 list 列表中的 1 的个数

n+4 为 list 列表中包含 PC (R15) 的情况

控制指令

附表 11 控制指令

汇编语法	说明	影响标志位	周期数
BREAK	设置断点		1
NOP	空操作		1
RESET	复位内核系统（DEBUG 除外）		1
SLEEP	休眠		1
SVC #imm8	触发 SVC 中断		1
SYNC	流水线同步		2

附录 2 用户手册更新记录

版本号	更新说明	更新日期
V3.0		20201106
V3.1	规范格式;	20211123
	修改 GPIO_OMOD 寄存器章节的高速模式下的速度大小;	
	增加 ADC 章节的内部参考电压的 1.5V 档位, 并增加注释;	
	增加低功耗模式外部唤醒引脚章节的中断使用描述;	
	修改 NRST 外部复位电路的电阻标注;	
	修改 ST_CALI 的默认值, 第 30/31 位为 1;	
	修改 RTCTSCH 位的描述	
	修改基本定时器的计数模式下的计数方式: 下降沿计数;	
	修改定时器 (基本、通用、高级) 的 DMA 相关描述;	
	修改定时器 (通用、高级) 的 Tx_CNT 的注释;	
	删除 RTC_CR 寄存器的 TTOE 位及其描述;	
	修改 PM_CTL2 寄存器的 RTCTPEN、RTCCTSEL 位描述;	
	修改 PM_CTL0 寄存器的 LPMS<2:0>位描述;	
	修改通用定时器的 Tx 向下计数立即更新图;	
修改 PM_CTL1 寄存器 PVDS<2:0>位的电压描述;		
修改 PM_CAL0 寄存器 BORSEL<2:0>位的 BOR 电压描述;		