



南京凌鸥创芯电子有限公司

# ***LKS32MC05x User Manual***

© 2020, 版权归凌鸥创芯所有  
机密文件，未经许可不得扩散

# 目录

目录.....	II
表格目录.....	I
图片目录.....	I
<b>1 文档约定.....</b>	<b>1</b>
1.1 寄存器读写权限.....	1
1.2 缩略词汇表.....	1
<b>2 地址空间.....</b>	<b>2</b>
<b>3 中断.....</b>	<b>3</b>
<b>4 模拟电路.....</b>	<b>4</b>
4.1 简述.....	4
4.2 电源管理系统.....	5
4.3 时钟系统.....	6
4.4 基准电压源.....	6
4.5 ADC 模块.....	6
4.6 运算放大器.....	7
4.7 比较器.....	8
4.8 温度传感器.....	9
4.9 DAC 模块.....	10
<b>5 系统时钟复位.....</b>	<b>11</b>
5.1 时钟.....	11
5.1.1 时钟源.....	11
5.1.2 功耗管理及休眠唤醒.....	12
5.1.2.1 休眠.....	12
5.1.2.2 唤醒.....	13
5.1.2.3 外设时钟门控.....	13
5.1.2.4 外设时钟分频.....	13
5.2 复位.....	14



5.2.1	复位源.....	14
5.2.1.1	硬件复位.....	14
5.2.1.1.1	硬件复位架构.....	14
5.2.1.1.2	硬件复位记录.....	15
5.2.1.2	软件复位.....	15
5.2.2	复位作用域.....	15
5.3	系统寄存器.....	16
5.3.1	地址分配.....	16
5.3.2	SYS_WDT_PSW 看门狗密码寄存器.....	17
5.3.3	SYS_WDT_CLR 看门狗清零寄存器.....	17
5.3.4	SYS_WDT_TH 看门狗门限寄存器.....	18
5.3.5	SYS_WDT_CNT 看门狗计数值寄存器.....	19
5.3.6	SYS_OPA_SEL 运放通道选择寄存器.....	19
5.3.7	模拟寄存器概述.....	21
5.3.8	SYS_AFE_REG0 模拟配置寄存器 0.....	21
5.3.9	SYS_AFE_REG1 模拟配置寄存器 1.....	22
5.3.10	SYS_AFE_REG3 模拟配置寄存器 3.....	22
5.3.11	SYS_AFE_REG5 模拟配置寄存器 5.....	24
5.3.12	SYS_AFE_REG6 模拟配置寄存器 6.....	25
5.3.13	SYS_AFE_REG7 模拟配置寄存器 7.....	26
5.3.14	SYS_TMP_A 温度传感器系数 A 寄存器.....	26
5.3.15	SYS_TMP_B 温度传感器系数 B 寄存器.....	27
5.3.16	SYS_AFE_DAC DAC 数字量寄存器.....	27
5.3.17	SYS_AFE_DAC_AMC DAC 增益校正寄存器.....	28
5.3.18	SYS_AFE_DAC_DC DAC 直流偏置寄存器.....	28
5.3.19	SYS_CLK_CFG 时钟控制寄存器.....	29
5.3.20	SYS_RST_CFG 复位控制寄存器.....	29
5.3.21	SYS_RST_SRC 复位源记录寄存器.....	30
5.3.22	SYS_CLR_RST 复位源记录清除寄存器.....	31
5.3.23	SYS_CLK_DIV0 外设时钟分频寄存器 0.....	31



5.3.24	<i>SYS_CLK_DIV2</i> 外设时钟分频寄存器 2.....	32
5.3.25	<i>SYS_CLK_FEN</i> 外设时钟门控寄存器.....	32
5.3.26	<i>SYS_CLK_SLP</i> 休眠寄存器.....	33
5.3.27	<i>SYS_SFT_RST</i> 软复位寄存器.....	33
5.3.28	<i>SYS_PROTECT</i> 写保护寄存器.....	35
<b>6</b>	<b>FLASH.....</b>	<b>36</b>
6.1	概述.....	36
6.2	功能特点.....	36
6.2.1	功能描述.....	37
6.2.1.1	复位操作.....	37
6.2.1.2	休眠操作.....	37
6.2.1.3	FLASH 读取操作.....	38
6.2.1.4	FLASH 编程操作.....	38
6.2.1.5	FLASH 擦除操作.....	40
6.2.1.6	FLASH 预取操作.....	41
6.2.1.7	FLASH 加密保护.....	41
6.2.1.8	FLASH 在线升级(IAP).....	42
6.2.1.8.1	开启中断的在线升级.....	42
6.2.1.8.2	关闭中断的在线升级.....	43
6.2.1.8.3	在线升级函数的位置.....	43
6.3	寄存器.....	43
6.3.1	地址分配.....	43
6.3.2	<i>FLASH_CFG</i> 配置寄存器 (推荐先读回, 按或/与方式修改).....	43
6.3.3	<i>FLASH_ADDR</i> 地址寄存器.....	44
6.3.4	<i>FLASH_WDATA</i> 写数据寄存器.....	45
6.3.5	<i>FLASH_RDATA</i> 读数据寄存器.....	45
6.3.6	<i>FLASH_ERASE</i> 擦除控制寄存器.....	46
6.3.7	<i>FLASH_PROTECT</i> 加密状态寄存器.....	46
6.3.8	<i>FLASH_READY</i> 工作状态寄存器.....	47
<b>7</b>	<b>GPIO.....</b>	<b>47</b>



7.1	概述.....	50
7.1.1	功能框图.....	50
7.1.2	产品特点.....	51
7.2	寄存器.....	51
7.2.1	地址分配.....	51
7.2.2	GPIOx_PIE.....	52
7.2.3	GPIOx_POE.....	52
7.2.4	GPIOx_PDI.....	53
7.2.5	GPIOx_PDO.....	54
7.2.6	GPIOx_PUE.....	54
7.2.7	GPIOx_PODE.....	55
7.2.8	GPIOx_PFLT.....	55
7.2.9	GPIOx_F3210.....	56
7.2.10	GPIOx_F7654.....	57
7.2.11	GPIOx_FBA98.....	58
7.2.12	GPIOx_FFEDC.....	58
7.2.13	GPIOx_BSRR.....	58
7.2.14	GPIOx_BRR.....	60
7.2.15	外部中断、唤醒.....	60
7.2.15.1	EXTI_CR0.....	61
7.2.15.2	EXTI_CR1.....	62
7.2.15.3	EXTI_IF.....	63
7.2.15.4	WAKE_POL.....	64
7.2.15.5	WAKE_EN.....	64
7.3	实现说明.....	65
7.3.1	上拉实现.....	65
7.4	应用指南.....	65
7.4.1	外部中断.....	65
7.4.2	使用GPIO的模拟功能.....	66
<b>8</b>	<b>CRC.....</b>	<b>67</b>



8.1	概述.....	67
8.2	基本原理.....	67
8.3	基本概念.....	67
8.3.1	对应关系.....	67
8.3.2	生成多项式.....	68
8.3.3	校验码位数.....	68
8.3.4	生成步骤.....	68
8.4	寄存器.....	69
8.4.1	地址分配.....	69
8.4.2	寄存器描述.....	70
8.4.2.1	CRC 信息码寄存器.....	70
8.4.2.2	CRC 控制寄存器.....	70
8.4.2.3	CRC 初始码寄存器.....	71
8.4.2.4	CRC 生成码寄存器.....	72
<b>9</b>	<b>ADC.....</b>	<b>73</b>
9.1	概述.....	73
9.1.1	功能框图.....	73
9.1.2	ADC 触发方式.....	74
9.1.3	ADC 输出数制.....	74
9.1.4	ADC 量程.....	75
9.1.5	ADC 校正.....	76
9.1.6	ADC 信号来源.....	76
9.2	寄存器.....	77
9.2.1	地址分配.....	77
9.2.2	采样数据寄存器.....	78
9.2.2.1	ADC_DAT0.....	78
9.2.2.2	ADC_DAT1.....	79
9.2.2.3	ADC_DAT2.....	79
9.2.2.4	ADC_DAT3.....	80
9.2.2.5	ADC_DAT4.....	80



9.2.2.6	ADC_DAT5.....	80
9.2.2.7	ADC_DAT6.....	81
9.2.2.8	ADC_DAT7.....	81
9.2.2.9	ADC_DAT8.....	82
9.2.2.10	ADC_DAT9.....	82
9.2.2.11	ADC_DAT10.....	82
9.2.2.12	ADC_DAT11.....	83
9.2.2.13	ADC_DAT12.....	83
9.2.2.14	ADC_DAT13.....	84
9.2.2.15	ADC_DAT14.....	84
9.2.2.16	ADC_DAT15.....	84
9.2.3	信号来源寄存器.....	85
9.2.3.1	ADC_CHN0.....	85
9.2.3.2	ADC_CHN1.....	85
9.2.3.3	ADC_CHN2.....	86
9.2.3.4	ADC_CHN3.....	86
9.2.4	分段通道数寄存器.....	87
9.2.4.1	ADC_CHNT.....	87
9.2.5	配置寄存器.....	87
9.2.5.1	ADC_CFG.....	87
9.2.6	软件触发寄存器.....	88
9.2.6.1	ADC_SWT.....	88
9.2.7	直流偏置寄存器.....	89
9.2.7.1	ADC_DC.....	89
9.2.8	增益校正寄存器.....	90
9.2.8.1	ADC_AMC.....	90
9.2.9	中断寄存器.....	90
9.2.9.1	ADC_IE.....	90
9.2.9.2	ADC_IF.....	91
9.3	应用指南.....	92



9.3.1	ADC 采样触发模式.....	92
9.3.1.1	单段触发模式.....	93
9.3.1.2	两段触发模式.....	94
9.3.1.3	四段触发模式.....	94
9.3.2	中断.....	95
9.3.2.1	单段触发采样完成中断.....	95
9.3.2.2	两段触发采样完成中断.....	95
9.3.2.3	四段触发采样完成中断.....	95
9.3.3	配置修改.....	95
9.3.4	选择对应的模拟通道.....	96
<b>10</b>	<b>UTIMER 通用定时器.....</b>	<b>97</b>
10.1	概述.....	97
10.1.1	功能框图.....	97
10.1.1.1	总线接口模块.....	97
10.1.1.2	寄存器模块.....	98
10.1.1.3	IO 滤波模块.....	98
10.1.1.4	通用定时器模块.....	98
10.1.1.5	时钟分频模块.....	98
10.1.2	功能特点.....	98
10.2	实现说明.....	98
10.2.1	时钟分频.....	98
10.2.2	中断标志清零.....	99
10.2.3	滤波.....	99
10.2.4	模式.....	99
10.2.4.1	计数器.....	99
10.2.4.2	比较模式.....	100
10.2.4.3	捕获模式.....	100
10.3	寄存器.....	101
10.3.1	地址分配.....	101
10.3.2	系统控制寄存器.....	102



10.3.2.1	UTIMER_CFG .....	102
<b>10.3.3</b>	<b>滤波控制寄存器.....</b>	<b>103</b>
10.3.3.1	UTIMER_FLT_TH01 .....	103
10.3.3.2	UTIMER_FLT_TH23 .....	103
<b>10.3.4</b>	<b>定时器寄存器.....</b>	<b>105</b>
10.3.4.1	UTIMER_UNT0_CFG Timer0 配置寄存器 .....	105
10.3.4.2	UTIMER_UNT1_CFG Timer1 配置寄存器 .....	106
10.3.4.3	UTIMER_UNT2_CFG Timer2 配置寄存器 .....	107
10.3.4.4	UTIMER_UNT3_CFG Timer3 配置寄存器 .....	109
10.3.4.5	UTIMER_UNT0_TH Timer 0 门限寄存器 .....	110
10.3.4.6	UTIMER_UNT1_TH Timer 1 门限寄存器 .....	111
10.3.4.7	UTIMER_UNT2_TH Timer 2 门限寄存器 .....	111
10.3.4.8	UTIMER_UNT3_TH Timer 3 门限寄存器 .....	112
10.3.4.9	UTIMER_UNT0_CNT Timer 0 计数寄存器 .....	112
10.3.4.10	UTIMER_UNT1_CNT Timer 1 计数寄存器 .....	112
10.3.4.11	UTIMER_UNT2_CNT Timer 2 计数寄存器 .....	113
10.3.4.12	UTIMER_UNT3_CNT Timer 3 计数寄存器 .....	113
10.3.4.13	UTIMER_UNT0_CMP0 Timer 0 通道 0 比较捕获寄存器 .....	114
10.3.4.14	UTIMER_UNT0_CMP1 Timer 0 通道 1 比较捕获寄存器 .....	114
10.3.4.15	UTIMER_UNT1_CMP0 Timer 1 通道 0 比较捕获寄存器 .....	115
10.3.4.16	UTIMER_UNT1_CMP1 Timer 1 通道 1 比较捕获寄存器 .....	115
10.3.4.17	UTIMER_UNT2_CMP0 Timer 2 通道 0 比较捕获寄存器 .....	116
10.3.4.18	UTIMER_UNT2_CMP1 Timer 2 通道 1 比较捕获寄存器 .....	116
10.3.4.19	UTIMER_UNT3_CMP0 Timer 3 通道 0 比较捕获寄存器 .....	117
10.3.4.20	UTIMER_UNT3_CMP1 Timer 3 通道 1 比较捕获寄存器 .....	117
10.3.4.21	UTIMER_UNT0_EVT Timer0 外部事件选择寄存器 .....	118
10.3.4.22	UTIMER_UNT1_EVT Timer1 外部事件选择寄存器 .....	118
10.3.4.23	UTIMER_UNT2_EVT Timer2 外部事件选择寄存器 .....	119
10.3.4.24	UTIMER_UNT3_EVT Timer3 外部事件选择寄存器 .....	119
<b>10.3.5</b>	<b>中断管理寄存器.....</b>	<b>120</b>



10.3.5.1	UTIMER_IE 中断使能寄存器.....	120
10.3.5.2	UTIMER_IF 中断标志寄存器.....	121
<b>11</b>	<b>HALL 信号处理模块.....</b>	<b>123</b>
11.1	综述.....	123
11.2	实现说明.....	123
11.2.1	信号来源.....	123
11.2.2	工作时钟.....	123
11.2.3	信号滤波.....	123
11.2.4	捕获.....	124
11.2.5	中断.....	124
11.2.6	数据流程.....	124
11.3	寄存器.....	125
11.3.1	地址分配.....	125
11.3.2	HALL_CFG HALL 模块配置寄存器.....	125
11.3.3	HALL_INFO HALL 模块信息寄存器.....	126
11.3.4	HALL_WIDTH HALL 宽度计数值寄存器.....	127
11.3.5	HALL_TH HALL 模块计数器门限值寄存器.....	127
11.3.6	HALL_CNT HALL 计数寄存器.....	128
<b>12</b>	<b>MCPWM.....</b>	<b>129</b>
12.1	概述.....	129
12.1.1	Base Counter 模块.....	130
12.1.2	Fail 信号处理.....	131
12.1.3	MCPWM 特殊输出状态.....	132
12.1.4	IO DRIVER 模块.....	132
12.1.4.1	MCPWM 波形输出-中心对齐模式.....	133
12.1.4.2	MCPWM 波形控制-中心对齐推挽模式.....	134
12.1.4.3	MCPWM 波形输出-边沿对齐模式.....	135
12.1.4.4	MCPWM 波形控制-边沿对齐推挽模式.....	135
12.1.4.5	MCPWM IO 死区控制.....	136
12.1.4.6	MCPWM IO 极性设置.....	137



12.1.4.7	MCPWM IO 自动保护 .....	137
12.1.5	ADC Trigger Timer 模块.....	137
12.2	寄存器 .....	138
12.2.1	地址分配.....	138
12.2.2	MCPWM_TH00.....	140
12.2.3	MCPWM_TH01.....	140
12.2.4	MCPWM_TH10.....	140
12.2.5	MCPWM_TH11.....	141
12.2.6	MCPWM_TH20.....	141
12.2.7	MCPWM_TH21.....	142
12.2.8	MCPWM_TH30.....	142
12.2.9	MCPWM_TH31.....	143
12.2.10	MCPWM_TMR0.....	143
12.2.11	MCPWM_TMR1 .....	143
12.2.12	MCPWM_TMR2 .....	144
12.2.13	MCPWM_TMR3.....	144
12.2.14	MCPWM_TH .....	145
12.2.15	MCPWM_UPDATE .....	145
12.2.16	MCPWM_IE.....	146
12.2.17	MCPWM_IF.....	147
12.2.18	MCPWM_EIE.....	149
12.2.19	MCPWM{EIF.....	149
12.2.20	MCPWM_PP.....	150
12.2.21	MCPWM_IO01 .....	150
12.2.22	MCPWM_IO23 .....	151
12.2.23	MCPWM_SDCFG .....	152
12.2.24	MCPWM_AUEN .....	153
12.2.25	MCPWM_TCLK.....	154
12.2.26	MCPWM_FAIL.....	155
12.2.27	MCPWM_PRT.....	156



12.2.28	<i>MCPWM_CNT</i> .....	156
12.2.29	<i>MCPWM_SWAP</i> .....	157
12.2.30	<i>MCPWM_DTH00</i> .....	158
12.2.31	<i>MCPWM_DTH01</i> .....	158
12.2.32	<i>MCPWM_DTH10</i> .....	159
12.2.33	<i>MCPWM_DTH11</i> .....	159
12.2.34	<i>MCPWM_DTH20</i> .....	160
12.2.35	<i>MCPWM_DTH21</i> .....	160
12.2.36	<i>MCPWM_DTH30</i> .....	160
12.2.37	<i>MCPWM_DTH31</i> .....	161
<b>13</b>	<b>UART</b> .....	<b>162</b>
13.1	概述.....	162
13.2	功能说明.....	162
13.2.1	发送.....	162
13.2.2	接收.....	162
13.2.3	波特率配置.....	162
13.2.4	收发端口互换(TX/RX 互换).....	163
13.3	寄存器.....	163
13.3.1	地址分配.....	163
13.3.2	<i>UARTx_CTRL</i> <i>UARTx</i> 控制寄存器.....	164
13.3.3	<i>UARTx_DIVH</i> <i>UARTx</i> 波特率设置高字节寄存器.....	164
13.3.4	<i>UARTx_DIVL</i> <i>UARTx</i> 波特率设置低字节寄存器.....	165
13.3.5	<i>UARTx_BUFF</i> <i>UARTx</i> 收发缓冲寄存器.....	165
13.3.6	<i>UARTx_ADR</i> <i>UARTx</i> 地址匹配寄存器.....	166
13.3.7	<i>UARTx_STT</i> <i>UARTx</i> 状态寄存器.....	166
13.3.8	<i>UARTx_IE</i> <i>UARTx</i> 中断使能寄存器.....	167
13.3.9	<i>UARTx_IF</i> <i>UARTx</i> 中断标志寄存器.....	167
13.3.10	<i>UARTx_IOC</i> <i>UARTx</i> IO 控制寄存器.....	168
<b>14</b>	<b>数字信号协处理器</b> .....	<b>170</b>
14.1	概述.....	170



14.2	寄存器.....	170
14.2.1	地址分配.....	170
14.2.2	协处理器状态控制寄存器.....	170
14.2.3	协处理器 <i>sin/cos</i> 相关寄存器.....	171
14.2.3.1	DSP_THETA.....	171
14.2.3.2	DSP_SIN.....	171
14.2.3.3	DSP_COS.....	172
14.2.4	协处理器 <i>arctan</i> 相关寄存器.....	172
14.2.4.1	DSP_XY.....	172
14.2.4.2	DSP_MOD.....	173
14.2.4.3	DSP_ARCTAN.....	173
14.2.5	协处理器开方相关寄存器.....	174
14.2.5.1	DSP_RAD.....	174
14.2.5.2	DSP_SQRT.....	174
<b>15</b>	<b>I2C.....</b>	<b>176</b>
15.1	概述.....	176
15.2	主要特性.....	176
15.3	功能描述.....	176
15.3.1	功能框图.....	176
15.3.2	功能说明.....	177
15.3.2.1	模式选择.....	177
15.3.2.2	从模式.....	178
15.3.2.2.1	从模式传输.....	178
15.3.2.2.2	从模式发送.....	179
15.3.2.2.3	从模式单字节接收.....	179
15.3.2.3	主模式.....	179
15.3.2.3.1	主模式传输.....	180
15.3.2.3.2	主模式发送.....	180
15.3.2.3.3	主模式接收.....	181
15.3.2.4	I2C 总线异常处理.....	181



15.3.2.5	中断处理.....	181
15.3.2.6	通讯速度设置.....	182
15.4	寄存器.....	182
15.4.1	地址分配.....	182
15.4.2	I2C_ADDR 地址寄存器.....	182
15.4.3	I2C_CFG 系统控制寄存器.....	183
15.4.4	I2C_SCR 状态控制寄存器.....	183
15.4.5	I2C_DATA 数据寄存器.....	185
15.4.6	I2C_MSCR 主模式寄存器.....	185
15.4.7	I2C_BCR I2C 传输控制寄存器.....	186
<b>16</b>	<b>SPI.....</b>	<b>187</b>
16.1	概述.....	187
16.2	主要特性.....	187
16.3	功能描述.....	187
16.3.1	功能框图.....	187
16.3.2	功能说明.....	188
16.3.2.1	全双工模式.....	188
16.3.2.2	半双工模式.....	189
16.3.2.3	片选信号.....	190
16.3.2.4	通讯格式.....	191
16.3.2.5	数据格式及长度.....	191
16.3.2.6	传输.....	192
16.3.2.7	中断处理.....	192
16.3.2.8	波特率设置.....	192
16.4	寄存器.....	193
16.4.1	地址分配.....	193
16.4.2	SPI_CFG SPI 控制寄存器.....	193
16.4.3	SPI_IE SPI 中断寄存器.....	194
16.4.4	SPI_BAUD SPI 波特率寄存器.....	195
16.4.5	SPI_TXDATA SPI 数据发送寄存器.....	195



16.4.6	SPI_RXDATA SPI 数据接收寄存器.....	196
16.4.7	SPI_SIZE SPI 数据传输长度寄存器.....	196
<b>17</b>	<b>CMP .....</b>	<b>197</b>
17.1	概述.....	197
17.2	寄存器.....	198
17.2.1	地址分配.....	198
17.2.2	CMP_IE 中断使能寄存器.....	198
17.2.3	CMP_IF 中断标志寄存器.....	199
17.2.4	CMP_TCLK 分频时钟控制寄存器.....	199
17.2.5	CMP_CFG 控制寄存器.....	200
17.2.6	CMP_BLCWIN 开窗控制寄存器.....	203
17.2.7	CMP_DATA 输出数据寄存器.....	203
<b>18</b>	<b>版本历史.....</b>	<b>205</b>



# 表格目录

表 2-1 系统地址空间分配.....	2
表 3-1 中断号分布.....	3
表 5-1 系统时钟源.....	11
表 5-2 PLL 作为 MCLK 时钟时的分频配置 .....	11
表 5-3 系统复位源.....	14
表 5-4 复位作用域.....	15
表 5-5 系统控制寄存器.....	16
表 5-6 SYS_WDT_PSW 看门狗密码寄存器.....	17
表 5-7 SYS_WDT_CLR 看门狗清零寄存器.....	18
表 5-8 SYS_WDT_TH 看门狗门限寄存器.....	18
表 5-9 SYS_WDT_CNT 看门狗计数值寄存器.....	19
表 5-10 运放通道选择寄存器 SYS_OPA_SEL.....	19
表 5-11 ADC 采样信号来源与寄存器设置.....	20
表 5-12 ADC 涉及 OPA 复用的采样通道设置 .....	20
表 5-13 SYS_AFE_REG0 模拟配置寄存器 0.....	21
表 5-14 SYS_AFE_REG1 模拟配置寄存器 1.....	22
表 5-15 SYS_AFE_REG3 模拟配置寄存器 3.....	23
表 5-16 SYS_AFE_REG5 模拟配置寄存器 5.....	24
表 5-17 SYS_AFE_REG6 模拟配置寄存器 6.....	26
表 5-18 SYS_AFE_REG7 模拟配置寄存器 7.....	26
表 5-23 SYS_TMP_A 温度传感器系数 A 寄存器.....	27
表 5-24 SYS_TMP_B 温度传感器系数 B 寄存器.....	27
表 5-25 SYS_AFE_DACDAC 数字量寄存器.....	27
表 5-26 SYS_AFE_DAC_AMC DAC 增益校正寄存器 .....	28
表 5-27 SYS_AFE_DAC_DC DAC 直流偏置寄存器.....	28
表 5-28 SYS_CLK_CFG 时钟控制寄存器.....	29
表 5-29 SYS_RST_CFG 复位控制寄存器.....	29
表 5-30 SYS_RST_SRC 复位源记录寄存器.....	30



表 5-31 SYS_CLR_RST 复位源记录清除寄存器.....	31
表 5-32 SYS_CLK_DIV0 外设时钟分频寄存器 0.....	31
表 5-33 SYS_CLK_DIV2 外设时钟分频寄存器 2.....	32
表 5-34 SYS_CLK_FEN 外设时钟门控寄存器.....	32
表 5-35 SYS_CLK_SLP 休眠寄存器.....	33
表 5-36 SYS_SFT_RST 软复位寄存器.....	33
表 5-37 SYS_PROTECT 写保护寄存器.....	35
表 6-1 FLASH 访问空间分配表.....	38
表 6-2 FLASH Sector 地址分配表.....	40
表 6-3 IAP_VTOR 寄存器描述.....	42
表 6-4 FLASH 控制器模块寄存器列表.....	43
表 6-5 FLASH_CFG 配置寄存器.....	43
表 6-6 FLASH_ADDR 地址寄存器.....	44
表 6-7 FLASH_WDATA 写数据寄存器.....	45
表 6-8 FLASH_RDATA 读数据寄存器.....	45
表 6-9 FLASH_ERASE 擦除控制寄存器.....	46
表 6-10 FLASH_PROTECT 加密状态寄存器.....	46
表 6-11 FLASH_READY 工作状态寄存器.....	47
表 7-1 GPIOx 寄存器列表.....	51
表 7-2 GPIO 中断/唤醒/配置锁定模块寄存器列表.....	51
表 7-3 GPIOx_PIE GPIOx 输入使能寄存器.....	52
表 7-4 GPIOx_POE GPIOx 输出使能寄存器.....	52
表 7-5 GPIOx_PDI GPIOx 输入数据寄存器.....	53
表 7-6 GPIOx_PDO GPIOx 输出数据寄存器.....	54
表 7-7 GPIOx_PUE GPIOx 上拉使能寄存器.....	54
表 7-8 GPIOx_PODE GPIOx 开漏使能寄存器.....	55
表 7-9 GPIOx_PFLT GPIOx 配置锁定寄存器.....	56
表 7-10 GPIOx_F3210 GPIOx 功能选择寄存器.....	56
表 7-11 GPIO 功能复用.....	57
表 7-12 GPIOx_F7654 GPIOx 功能选择寄存器.....	57



表 7-13 GPIOx_FBA98 GPIOx 功能选择寄存器.....	58
表 7-14 GPIOx_FFEDC GPIOx 功能选择寄存器.....	58
表 7-15 GPIOx_BSRR GPIOx 位操作寄存器.....	59
表 7-16 GPIOx_BRR GPIOx 位清零寄存器.....	60
表 7-17 EXTI_CR0 外部中断配置寄存器.....	61
表 7-18 EXTI_CR1 外部中断配置寄存器.....	62
表 7-19 EXTI_IF 外部中断标志寄存器.....	63
表 7-20 WAKE_POL 外部唤醒源极性配置寄存器.....	64
表 7-21 WAKE_EN 外部唤醒源使能寄存器.....	64
表 7-22 GPIO 上拉资源分布表.....	65
表 8-1 CRC 寄存器列表.....	69
表 8-2 CRC_DR CRC 数据寄存器.....	70
表 8-3 CRC_CR CRC 控制寄存器.....	70
表 8-4 CRC_INIT CRC 初始码寄存器.....	71
表 8-5 CRC_POL CRC 生成码寄存器.....	72
表 9-1 ADC 输出数字量数制转换.....	75
表 9-2 ADC 采样信号来源.....	76
表 9-3 ADC0 寄存器列表.....	78
表 9-4 ADC_DAT0 采样数据寄存器.....	78
表 9-5 ADC_DAT1 采样数据寄存器.....	79
表 9-6 ADC_DAT2 采样数据寄存器.....	79
表 9-7 ADC_DAT3 采样数据寄存器.....	80
表 9-8 ADC_DAT4 采样数据寄存器.....	80
表 9-9 ADC_DAT5 采样数据寄存器.....	80
表 9-10 ADC_DAT6 采样数据寄存器.....	81
表 9-11 ADC_DAT7 采样数据寄存器.....	81
表 9-12 ADC_DAT8 采样数据寄存器.....	82
表 9-13 ADC_DAT9 采样数据寄存器.....	82
表 9-14 ADC_DAT10 采样数据寄存器.....	82
表 9-15 ADC_DAT11 采样数据寄存器.....	83



表 9-16 ADC_DAT12 采样数据寄存器 .....	83
表 9-17 ADC_DAT13 采样数据寄存器 .....	84
表 9-18 ADC_DAT14 采样数据寄存器 .....	84
表 9-19 ADC_DAT15 采样数据寄存器 .....	84
表 9-20 ADC_CHN0 信号来源寄存器 .....	85
表 9-21 ADC_CHN1 信号来源寄存器 .....	85
表 9-22 ADC_CHN2 信号来源寄存器 .....	86
表 9-23 ADC_CHN3 信号来源寄存器 .....	86
表 9-24 ADC_CHNT 分段通道数寄存器 .....	87
表 9-25 ADC_CFG 模式配置寄存器 .....	87
表 9-26 ADC_SWT 软件触发寄存器 .....	89
表 9-27 ADC_DC 直流偏置寄存器 .....	89
表 9-28 ADC_AMC 增益校正寄存器 .....	90
表 9-29 ADC_IE 中断使能寄存器 .....	90
表 9-30 ADC_IF 中断标志寄存器 .....	91
表 9-31 ADC 采样触发模式 .....	93
表 10-1 通用定时器配置寄存器地址分配 .....	101
表 10-2 UTIMER_CFG UTIMER 配置寄存器 .....	102
表 10-3 UTIMER_FLT_TH01 滤波控制寄存器 .....	103
表 10-4 UTIMER_FLT_TH23 滤波控制寄存器 .....	104
表 10-5 UTIMER_UNT0_CFG Timer 0 配置寄存器 .....	105
表 10-6 UTIMER_UNT1_CFG Timer 1 配置寄存器 .....	106
表 10-7 UTIMER_UNT2_CFG Timer 2 配置寄存器 .....	108
表 10-8 UTIMER_UNT3_CFG Timer 3 配置寄存器 .....	109
表 10-9 UTIMER_UNT0_TH Timer 0 门限寄存器 .....	110
表 10-10 UTIMER_UNT1_TH Timer 1 门限寄存器 .....	111
表 10-11 UTIMER_UNT2_TH Timer 2 门限寄存器 .....	111
表 10-12 UTIMER_UNT3_TH Timer 3 门限寄存器 .....	112
表 10-13 UTIMER_UNT0_CNT Timer 0 计数寄存器 .....	112
表 10-14 UTIMER_UNT1_CNT Timer 1 计数寄存器 .....	113



表 10-15 UTIMER_UNT2_CNT Timer 2 计数寄存器.....	113
表 10-16 UTIMER_UNT3_CNT Timer 3 计数寄存器.....	113
表 10-17 UTIMER_UNT0_CMP0 Timer 0 通道 0 比较捕获寄存器.....	114
表 10-18 UTIMER_UNT0_CMP1 Timer 0 通道 1 比较捕获寄存器.....	114
表 10-19 UTIMER_UNT1_CMP0 Timer 1 通道 0 比较捕获寄存器.....	115
表 10-20 UTIMER_UNT1_CMP1 Timer 1 通道 1 比较捕获寄存器.....	115
表 10-21 UTIMER_UNT2_CMP0 Timer 2 通道 0 比较捕获寄存器.....	116
表 10-22 UTIMER_UNT2_CMP1 Timer 2 通道 1 比较捕获寄存器.....	116
表 10-23 UTIMER_UNT3_CMP0 Timer 3 通道 0 比较捕获寄存器.....	117
表 10-24 UTIMER_UNT3_CMP1 Timer 3 通道 1 比较捕获寄存器.....	117
表 10-25 UTIMER_UNT0_EVT Timer 0 外部事件选择寄存器.....	118
表 10-26 UTIMER_UNT1_EVT Timer 1 外部事件选择寄存器.....	118
表 10-27 UTIMER_UNT2_EVT Timer 2 外部事件选择寄存器.....	119
表 10-28 UTIMER_UNT3_EVT Timer 3 外部事件选择寄存器.....	120
表 10-29 UTIMER_IE 中断使能寄存器.....	120
表 10-30 UTIMER_IF 中断标志寄存器.....	121
表 11-1 HALL 模块寄存器地址分配.....	125
表 11-2 HALL_CFG HALL 模块配置寄存器.....	125
表 11-3 HALL_INFO HALL 模块信息寄存器.....	126
表 11-4 HALL_WIDTH HALL 宽度计数值寄存器.....	127
表 11-5 HALL_TH HALL 模块计数器门限值寄存器.....	128
表 11-6 HALL_CNT HALL 计数寄存器.....	128
表 12-1 MCPWM 计数器阈值与事件对应表.....	137
表 12-2 MCPWM 模块寄存器列表.....	138
表 12-3 受 MCPWM_PRT 保护的寄存器.....	139
表 12-4 存在影子寄存器的寄存器.....	139
表 12-5 MCPWM_TH00 配置寄存器.....	140
表 12-6 MCPWM_TH01 配置寄存器.....	140
表 12-7 MCPWM_TH10 配置寄存器.....	141
表 12-8 MCPWM_TH11 配置寄存器.....	141



表 12-9 MCPWM_TH20 配置寄存器.....	141
表 12-10 MCPWM_TH21 配置寄存器.....	142
表 12-11 MCPWM_TH30 配置寄存器.....	142
表 12-12 MCPWM_TH31 配置寄存器.....	143
表 12-13 MCPWM_TMR0 配置寄存器.....	143
表 12-14 MCPWM_TMR1 配置寄存器.....	144
表 12-15 MCPWM_TMR2 配置寄存器.....	144
表 12-16 MCPWM_TMR3 配置寄存器.....	144
表 12-17 MCPWM_TH 配置寄存器.....	145
表 12-18 MCPWM_UPDATE MCPWM 手动更新寄存器.....	145
表 12-19 MCPWM_IE 配置寄存器.....	146
表 12-20 MCPWM_IF 配置寄存器.....	148
表 12-21 MCPWM_EIE 配置寄存器.....	149
表 12-22 MCPWM{EIF 配置寄存器.....	149
表 12-23 MCPWM_PP 配置寄存器.....	150
表 12-24 MCPWM_IO01 配置寄存器.....	151
表 12-25 MCPWM_IO23 配置寄存器.....	152
表 12-26 MCPWM_SDCFG 配置寄存器.....	153
表 12-27 MCPWM_AUEN MCPWM 自动更新使能寄存器.....	153
表 12-28 MCPWM_TCLK 配置寄存器.....	154
表 12-29 MCPWM_FAIL 配置寄存器.....	155
表 12-30 MCPWM_PRT 寄存器.....	156
表 12-31 MCPWM_CNT 寄存器.....	156
表 12-32 MCPWM_SWAP 寄存器.....	157
表 12-33 MCPWM 默认输出表.....	157
表 12-34 MCPWM 修改后的输出表.....	158
表 12-35 MCPWM_DTH00 配置寄存器.....	158
表 12-36 MCPWM_DTH01 配置寄存器.....	159
表 12-37 MCPWM_DTH10 配置寄存器.....	159
表 12-38 MCPWM_DTH11 配置寄存器.....	159



表 12-39 MCPWM_DTH20 配置寄存器.....	160
表 12-40 MCPWM_DTH21 配置寄存器.....	160
表 12-41 MCPWM_DTH30 配置寄存器.....	161
表 12-42 MCPWM_DTH31 配置寄存器.....	161
表 13-1 UART 波特率配置示例 .....	163
表 13-2 UARTx 地址分配列表.....	163
表 13-3 UARTx_CTRL UARTx 控制寄存器.....	164
表 13-4 UARTx_DIVH UARTx 波特率设置高字节寄存器.....	165
表 13-5 UARTx_DIVL UARTx 波特率设置低字节寄存器.....	165
表 13-6 UARTx_BUFF UARTx 收发缓冲寄存器.....	165
表 13-7 UARTx_ADR UARTx 地址匹配寄存器.....	166
表 13-8 UARTx_STT UARTx 状态寄存器.....	166
表 13-9 UARTx_IE UARTx 中断使能寄存器.....	167
表 13-10 UARTx_IF UARTx 中断使能寄存器.....	168
表 13-11 UARTx_IOC UARTxIO 控制寄存器.....	168
表 14-1 协处理器寄存器列表.....	170
表 14-2 DSP_SC 协处理器状态控制寄存器.....	170
表 14-3 协处理器 sin/cos 角度输入寄存器.....	171
表 14-4 协处理器 sin/cos 正弦结果寄存器.....	172
表 14-5 协处理器 sin/cos 余弦结果寄存器.....	172
表 14-6 协处理器 arctan/module 坐标输入寄存器.....	172
表 14-7 协处理器 arctan 角度结果 arctan(Y/X) 角度寄存器.....	173
表 14-8 协处理器 arctan 角度结果 arctan(Y/X) 角度寄存器.....	173
表 14-9 协处理器被开方数寄存器.....	174
表 14-10 协处理器平方根寄存器.....	174
表 15-1 I2C 寄存器地址分配表.....	182
表 15-2 I2C_ADDR 地址寄存器.....	182
表 15-3 I2C_CFG 系统控制寄存器.....	183
表 15-4 I2C_SCR 状态控制寄存器.....	183
表 15-5 I2C_DATA 数据寄存器.....	185



表 15-6 I2C_MSCR 主模式寄存器.....	185
表 15-7 I2C_BCR 传输控制寄存器.....	186
表 16-1 SPI 模块控制寄存器列表.....	193
表 16-2 SPI_CFG 系统控制寄存器.....	193
表 16-3 SPI_IE 中断寄存器.....	194
表 16-4 SPI_BAUD 控制寄存器.....	195
表 16-5 SPI_TXDATA 数据发送寄存器.....	195
表 16-6 SPI_RXDATA 数据接收寄存器.....	196
表 16-7 SPI_SIZE 数据传输长度寄存器.....	196
表 17-1 比较器寄存器列表.....	198
表 17-2 CMP_IE 比较器中断使能寄存器.....	198
表 17-3 CMP_IF 比较器中断标志寄存器.....	199
表 17-4 CMP_TCLK 比较器分频时钟控制寄存器.....	199
表 17-5 CMP_CFG 比较器控制寄存器.....	200
表 17-6 CMP_BLCWIN 比较器开窗控制寄存器.....	203
表 17-7 CMP_DATA 比较器输出数据寄存器.....	204
表 18-1 文档版本历史.....	205



# 图片目录

图 4-1 模拟电路功能框图.....	5
图 4-2 BEMF <sub>x</sub> MID 信号.....	8
图 4-3 温度传感器曲线.....	9
图 5-1 时钟架构.....	12
图 5-2 复位架构.....	15
图 6-1 FLASH 存储体空间划分框图.....	36
图 6-2 FLASH 控制状态转换图.....	37
图 6-3 FLASH 间接读取操作流程图中.....	38
图 6-4 FLASH 模块编程操作流程图中.....	39
图 6-5 FLASH 模块编程操作流程图中.....	40
图 6-6 FLASH 模块擦除操作流程图中.....	41
图 7-1 GPIO 功能框图.....	50
图 9-1 ADC 采集模块功能框图.....	74
图 9-2 一倍增益设置下 ADC 模数转换数制量程.....	75
图 9-3 ADC 单段采样状态转移图.....	94
图 9-4 ADC 两段采样状态转移图.....	94
图 9-5 ADC 四段采样状态转移图.....	95
图 10-1 模块顶层功能框图.....	97
图 10-2 滤波示意图.....	99
图 10-3 通用计数器.....	99
图 10-4 比较模式.....	100
图 10-5 捕获模式.....	101
图 11-1 7/5 滤波模块框图.....	124
图 11-2 数据流程框图.....	125
图 12-1 MCPWM 模块框图.....	129
图 12-2 Base Counter t <sub>0</sub> /t <sub>1</sub> 时序.....	130
图 12-3 MCPWM 更新机制.....	131



图 12-4 MCPWM FAIL 逻辑示意图 .....	131
图 12-5 MCPWM Fail 信号滤波时钟生成逻辑 .....	132
图 12-6 IO Driver 模块数据流程图 .....	133
图 12-7 MCPWM 时序 TH<n>0 和 TH<n>1-中心对齐模式 .....	134
图 12-8 MCPWM 时序 TH<n>0 和 TH<n>1-中心对齐推挽模式 .....	134
图 12-9 MCPWM 时序边沿对齐模式 .....	135
图 12-10 MCPWM 时序 TH<n>0 和 TH<n>1 边沿对齐推挽模式 .....	136
图 12-11 MCPWM IO 控制示意图 .....	137
图 15-1 I2C 模块顶层功能框图 .....	176
图 15-2 基本 I2C 传输时序图 .....	177
图 15-3 从模式传输示意图 .....	179
图 15-4 主模式下单字节传输示意图 .....	180
图 16-1 SPI 模块结构框图 .....	187
图 16-2 SPI 接口全双工模式互连框图 .....	189
图 16-3 SPI 接口半双工模式互连框图 .....	190
图 16-4 SPI 模块 Slave 模式片选信号选择 .....	191
图 16-5 SPI 模块 Master 模式片选信号选择 .....	191
图 16-6 SPI 模块中断选信号产生图 .....	192
图 17-1 比较器滤波时钟产生 .....	200
图 17-2 比较器控制及中断产生逻辑 .....	201
图 17-3 CMP 与 MCPWM 的联动 .....	202
图 17-4 比较器开窗功能图示 .....	202



# 1 文档约定

## 1.1 寄存器读写权限

RW	读/写，软件可以读写这些位。
RO	只读，软件只能读取这些位。
WO	只写，软件只能写入该位。读取该位时将返回默认值。
RW1C(Read and Write 1 to Clear)	可读，写 1 清零。

## 1.2 缩略词汇表

字：32 位数据/指令。

半字：16 位数据/指令。

字节：8 位数据。

双字：64 位数据。

ADC: Analog-Digital Converter, 模数转换器

DAC: Digital-Analog Converter, 数模转换器

BGP: Bandgap, 带隙基准

WDT: Watch dog, 看门狗

LSI: Low Speed Internal Clock, 即 64kHz RC 时钟

HSI: High Speed Internal Clock, 即 4MHz RC 时钟

HSE: High Speed External Clock, 即 4~8MHz 外部晶振时钟

PLL: Phase Lock Loop Clock, 即 96MHz 锁相环时钟, 通常用作系统高速时钟

POR: Power-On Reset, 即上电复位, 芯片系统上电时产生的复位信号

NVR: Non-Volatile Register, flash 中区别于 main 区域之外的一块存储区域

IAP (在应用中编程) : IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。

ICP (在线编程) : ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。

CW: Clock wise, 顺时针

CCW: Counter clock wise, 逆时针

Option bytes: 选项字节, 保存在 Flash 中的 MCU 配置字节



## 2 地址空间

数据字节以小端格式存放在存储器中。一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。其他所有没有分配给片上存储器和外设的存储器空间都是保留的地址空间。

表 2-1 系统地址空间分配

外设	工作时钟/软复位	开始地址	结束地址	空间大小	说明
FLASH	同总线	0x0000_0000	0x0000_7FFF	32KB	FLASH 存储空间
RAM	同总线	0x2000_0000	0x2000_09FF	2560B	RAM
SYS	同总线	0x4000_0000	0x4000_03FF	1kB	SYSTEM control, Clock / Reset Management
FLSCR	同总线	0x4000_0400	0x4000_07FF	1kB	FLASH control registers
SPI	外设分频时钟[0] 软复位[0]	0x4001_0000	0x4001_03FF	1kB	SPI interface
I2C	外设分频时钟[0] 软复位[0]	0x4001_0400	0x4001_07FF	1kB	I2C interface
IO MASK	配置	0x4001_0800	0x4001_0BFF	1kB	IO MASK
CMP	同总线	0x4001_0C00	0x4001_0FFF	1kB	Comparator
HALL	外设分频时钟[1] 软复位[1]	0x4001_1000	0x4001_13FF	1kB	HALL interface
ADC	ACLK	0x4001_1400	0x4001_17FF	1kB	ADC interface
TIMER	外设分频时钟[2] 软复位[2]	0x4001_1800	0x4001_1BFF	1kB	General Purpose Timer
MCPWM	外设分频时钟[3] 软复位[3]	0x4001_1C00	0x4001_1FFF	1kB	Motor Control Pulse Width Modulation
GPIO	同总线	0x4001_2000	0x4001_23FF	1kB	General Purpose Input / Output
CRC	同总线	0x4001_2400	0x4001_27FF	1kB	Cyclic redundancy check
UART0	外设分频时钟[4] 软复位[4]	0x4001_2800	0x4001_2BFF	1kB	
UART1	外设分频时钟[5] 软复位[5]	0x4001_2C00	0x4001_2FFF	1kB	
协处理器	同总线	0x4001_5000	0x4001_5FFF	4kB	

### 3 中断

嵌套向量中断控制器位于 Cortex-M0 内部。当中断事件发生时，通知 M0 暂停主程序执行，按照优先级设定跳转进入中断服务函数。

最多可以支持 32 个独立的中断源及中断向量，其中 LKS32MC05X 系列芯片共使用了其中的 18 个中断源，后 14 个保留未使用。

最多支持 4 个中断优先级可供编程选择。

表 3-1 中断号分布

中断号	说明	中断号	说明
-14	NMI		
-13	HardFault		
-12	保留		
-11			
-10			
-9			
-8			
-7			
-6			
-5	SVCall		
-4	保留		
-3			
-2	PendSV		
-1	SysTick		
0	TIMER0	16	Reserved
1	TIMER1	17	Reserved
2	TIMER2	18	Reserved
3	TIMER3	19	Reserved
4	I2C	20	Reserved
5	SPI	21	Reserved
6	GPIO	22	Reserved
7	HALL	23	Reserved
8	UART0	24	Reserved
9	UART1	25	Reserved
10	ADC	26	Reserved
11	MCPWM	27	Reserved
12	CMP	28	Reserved
13	WAKEUP, 唤醒中断	29	Reserved
14	Reserved	30	Reserved
15	Reserved	31	Reserved

## 4 模拟电路

### 4.1 简述

模拟电路包含以下模块：

- 集成 1 路 12BIT SAR ADC，采样率 2MHz，16 通道
- 集成 2 路运算放大器，可设置为 PGA 模式
- 集成两路比较器，可设置迟滞模式
- 集成 12BIT 数模转换器
- 内置±2°C温度传感器
- 内置高精度基准源

各个模块之间的相互关系、以及各模块的控制寄存器（寄存器的说明见下文“模拟寄存器表”）如下图所示。

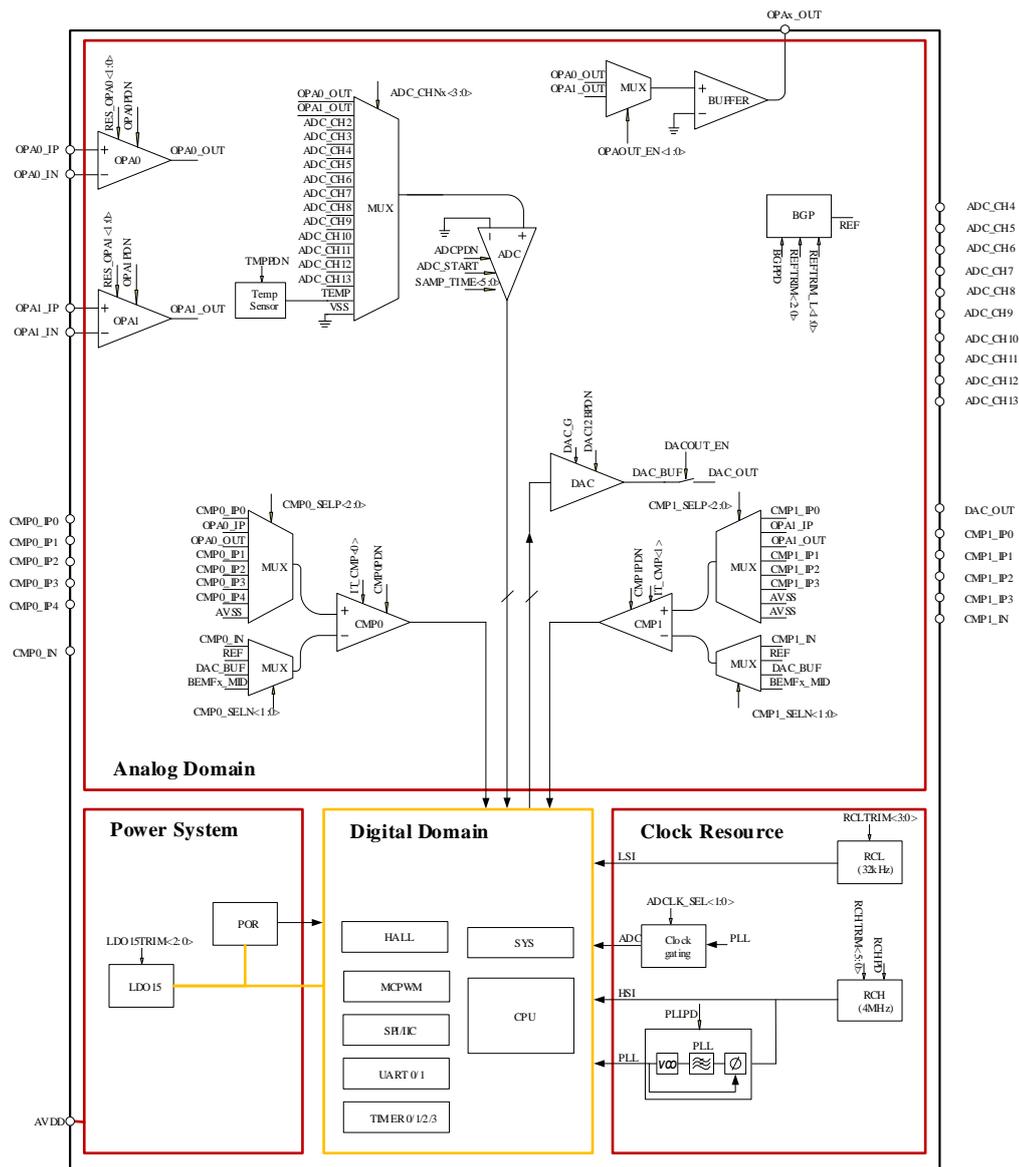


图 4-1 模拟电路功能框图

## 4.2 电源管理系统

电源管理系统由 LDO15 模块、上电/掉电复位模块（POR）组成。

该芯片由 3.3~5V 单电源供电，以节省芯片外的电源成本。芯片内部集成一路 LDO15 给内部所有数字电路、PLL 模块供电。

LDO 上电后自动开启，无需软件配置，但 LDO 输出电压可通过软件实现微调。

LPOR 模块监测 LDO15 的电压，在 LDO15 电压低于 1.25V 时（例如上电之初，或者掉电时），为数字电路提供复位信号以避免数字电路工作产生异常。



HPOR 模块监测 AVDD 的电压，在 AVDD 电压低于 2.2V 时（例如上电之初，或者掉电时），为数字电路提供复位信号以避免数字电路工作产生异常。

### 4.3 时钟系统

时钟系统包括内部 64kHz RC 时钟、内部 4MHz RC 时钟、PLL 电路组成。

64kHz RC 时钟 LSI 主要用于系统内的看门狗模块以及复位信号滤波等。4MHz RC 时钟可作为 MCU 主时钟使用，MCU 也可以使用 PLL 时钟，PLL 最高可提供 96MHz 的时钟。

64kHz 和 4MHz RC 时钟均带有出厂校正，64kHz RC 时钟在-40~105°C范围内的精度为±50%，4MHz RC 时钟在该温度范围的精度为±1%。

4MHz RC 时钟通过设置 RCHPD='0'打开（默认打开，写 1 关闭），RC 时钟需要 BGP 电压基准源模块提供基准电压和电流，因此开启 RC 时钟需要先开启 BGP 模块（BGPPD='0'）。芯片上电的默认状态下，4MHz RC 时钟和 BGP 模块都是开启的。64kHz RC 时钟始终开启，不能关闭。

PLL 对 4MHz RC 时钟进行倍频，给 MCU、ADC 等模块提供更高速的工作时钟。MCU 和 PWM 模块的最高时钟为 96MHz，ADC 模块最高时钟 48MHz，通过寄存器 ADCLKSEL[1:0]可设置不同的 ADC 工作频率。

PLL 通过设置 PLLPDN='1'打开（默认关闭，设 1 打开），开启 PLL 模块之前，同样也需要开启 BGP 模块。开启 PLL 之后，PLL 需要 6us 的稳定时间来输出稳定时钟。芯片上电的默认状态下，RCH 时钟和 BGP 模块都是开启的，但 PLL 默认是关闭的，需要软件来开启。

ADCLKSEL<1:0>的说明见模拟寄存器 [SYS AFE REG7](#)

BGPPD/RCHPD/PLLPDN 的说明见模拟寄存器 [SYS AFE REG5](#)

### 4.4 基准电压源

基准源电路(BGP REF: Bandgap reference)为 ADC、DAC、RC 时钟、PLL、温度传感器、运算放大器、比较器和 FLASH 提供基准电压和电流，使用上述任何一个模块之前，都需要开启 BGP 基准电压源。

芯片上电的默认状态下，BGP 模块是开启的。通过设置 BGPPD='0'将基准源打开，从关闭到开启，BGP 需要约 6us 达到稳定。BGP 输出电压约 1.2V，精度为±0.8%

BGPPD 的说明见模拟寄存器 [SYS AFE REG5](#)

### 4.5 ADC 模块

请参加第 9 章 ADC



## 4.6 运算放大器

芯片集成 2 路输入输出轨到轨 (rail-to-rail) 运算放大器 OPA\_A 和 OPA\_B，内置反馈电阻，外部引脚上需串联一个电阻  $R_0$  到信号源。反馈电阻  $R_2$ : $R_1$  的阻值可通过寄存器 RES\_OPAx[1:0] 设置，以实现不同的放大倍数。

RES\_OPAx<1:0>的说明见模拟寄存器 [SYS AFE REG0](#)

放大器的结构示意图如下所示：

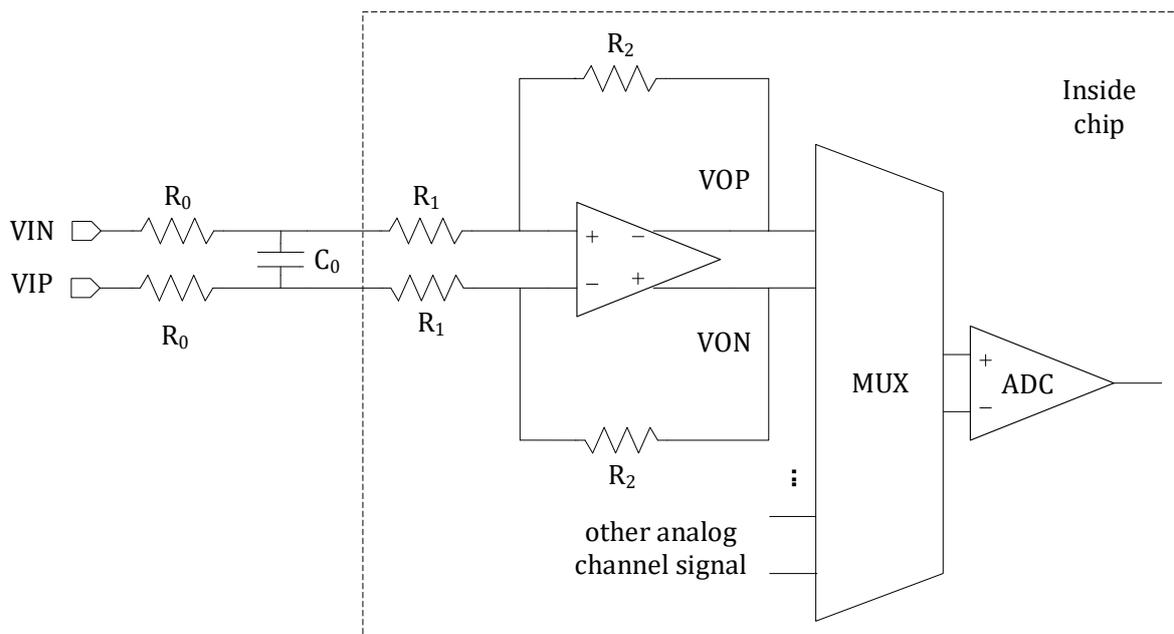


图 4-2 放大器框图

图中两个  $R_0$  是片外需放置的电阻，阻值必须相等，最终的放大倍数为  $R_2/(R_1+R_0)$ 。

对于 MOS 管电阻直接采样的应用，由于 MOS 下管关断、上管导通时信号会升高到数十 V 的电源电压，为减小此时往芯片引脚里流入的电流，建议接  $>20k\Omega$  的外部电阻。

对于分流电阻采样的应用，建议接  $100\sim 2K\Omega$  的外部电阻。 $C_0$  为信号滤波电容，和  $R_0$  形成一阶 RC 滤波电路。 $R_0$  的具体阻值可根据  $R_0 * C_0$  的滤波常数而定。如果信号上噪声较小不需要滤波、或者信号需要很大的带宽（较快的响应速度），则  $C_0$  可以不加。

放大器可通过设置 OPAOUT\_EN<1:0> 选择将 2 路放大器中的某一路输出信号通过 BUFFER 送至 P2.7 管脚口进行测量和应用（对应关系见 datasheet 芯片管脚说明）。因为有 BUFFER 存在，在运放正常工作模式下也可以选择送一路运放输出信号出来。

OPAOUT\_EN<1:0>的说明见模拟寄存器 [SYS AFE REG0](#)

芯片上电的默认状态下，放大器模块是关闭的。放大器可通过设置 OPAxPDN=1 (x=0,1,2,3) 打开。开启放大器之前，需要先开启 BGP 模块。

OPAxPDN 的说明见模拟寄存器 [SYS AFE REG5](#)



运放输入正负端内置钳位二极管，电机相线通过一个匹配电阻后直接接入输入端，从而简化了 MOSFET 电流采样的外置电路。

芯片引脚有 4 组 OPA 输入差分信号，OPA0\_IP/OPA0\_IN 和 OPA2\_IP/OPA2\_IN 分时复用运放 OPA\_A，OPA1\_IP/OPA1\_IN 和 OPA3\_IP/OPA3\_IN 分时复用运放 OPA\_B。

## 4.7 比较器

内置 2 路比较器，比较器比较速度可编程、迟滞电压可编程、信号源可编程。

比较器的比较延时为 0.15us，还可通过寄存器 CMP\_FT 设置为小于 30ns。迟滞电压通过 CMP\_HYS 设置为 20mV/0mV。

比较器正端输入信号来源可以通过寄存器 CMPx\_SELN[2:0] 进行设置；负端输入信号来源可以通过寄存器 CMPx\_SELN[1:0] 进行设置 (x=0/1，代表 CMP0/CMP1 两个比较器)。

需说明的是，两个比较器负输入端的 BEMFx\_MID 信号，是对比较器正输入端信号 CMPx\_IP1/ CMPx\_IP2/ CMPx\_IP3 信号的平均。其中电阻 R=8.2k 欧，图中的开关只有在比较器负输入端信号选择为 BEMFx\_MID 之后才会导通，否则开关都处于断开状态。

BEMFx\_MID 主要用于 BLDC 方波模式控制时，虚拟电机相线中心点电压，用于反电势过零点检测。三个相线分压后，分别接 CMPx\_IP1、CMPx\_IP2、CMPx\_IP3，MCU 控制比较器负端选择 BEMFx\_MID，比较器正端的多路选择器以分时复用的方式分别选择 CMPx\_IP1、CMPx\_IP2、CMPx\_IP3，就可以比较出反电势过零点。

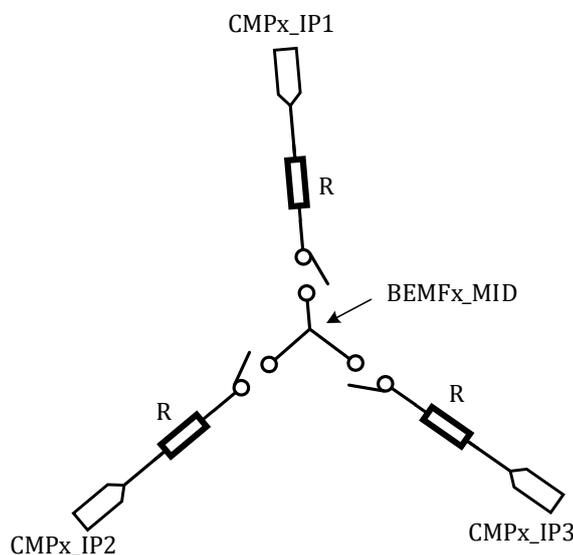


图 4-2 BEMFx\_MID 信号

比较器输出结果，可以通过 CMP\_DATA 寄存器读出。

CMPx\_SELN<1:0>/ CMPx\_SELN<2:0>/ CMP\_HYS 的说明见模拟寄存器 [SYS\\_AFE\\_REG3](#)

比较器的输出的说明见 17.2.7CMP\_DATA 输出数据寄存器

芯片上电的默认状态下，比较器模块是关闭的。比较器通过设置  $CMPxPDN=1$  ( $x=0,1$ ) 打开，开启比较器之前，需要先开启 BGP 模块。

CMPxPDN 的说明见模拟寄存器 [SYS AFE REG5](#)

## 4.8 温度传感器

芯片内置温度传感器，在  $-40\sim 85^{\circ}\text{C}$  范围内精度为  $2^{\circ}\text{C}$ 。  $85\sim 105^{\circ}\text{C}$  范围内精度为  $3^{\circ}\text{C}$ 。

芯片出厂前会经温度校正，校正值保存在 flash info 区。

芯片上电的默认状态下，温度传感器模块是关闭的。开启传感器之前，需要先开启 BGP 模块。

温度传感器通过设置  $TMPPDN=1$  打开，开启到稳定需要约  $2\mu\text{s}$ ，因此需在 ADC 测量传感器之前  $2\mu\text{s}$  打开。

温度传感器信号连至 ADC 的通道 14。

ADC 部分的设置参考[模数转换器\(ADC\)章节](#)

TMPPDN 的说明见模拟寄存器 [SYS AFE REG5](#)

温度传感器的典型曲线如下图所示：

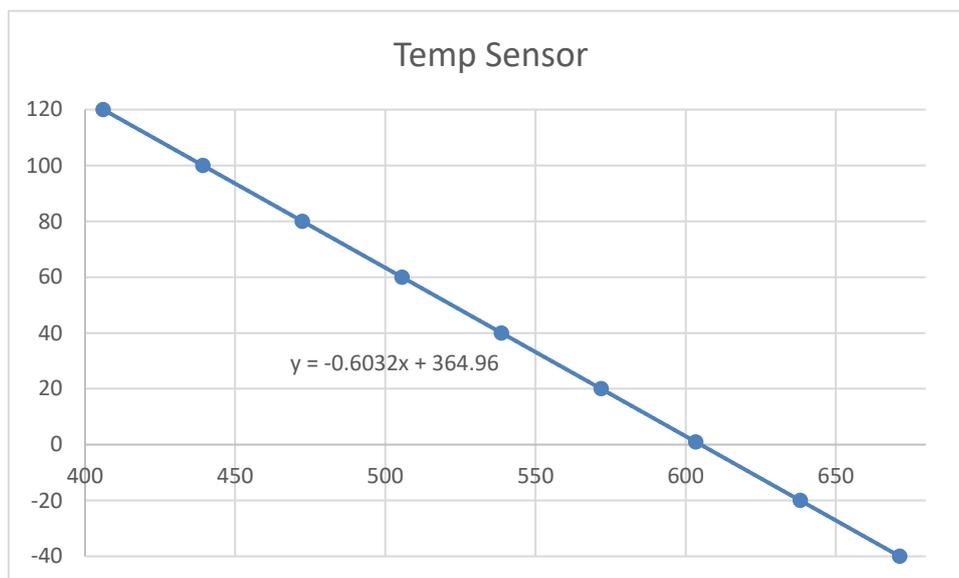


图 4-3 温度传感器曲线

图中 X 轴为温度传感器的温度信号所对应的 ADC 值，Y 轴为传感器所处的温度。测温时，按照如上要求配置传感器相关寄存器，并得到 ADC 值后，将 ADC 值作为 X 代入公式：

$$y = -0.6110x + 364.96$$

求得的 Y 值即为此时的温度。

公式中有两个系数， $a = -0.6032$ ,  $b = 364.96$ 。对于不同的芯片，b 系数的值是不一样的。芯片出厂

前会经过温度标定，将每颗芯片所对应的系数 **b** 写入 flash 的 info 区，地址为 0x0000039C。存储时，会将 **b** 系数小数点右移一位（乘 10）存入 info 区，小数点后第二位不进行保存。

同时为方便客户操作，系数 **a** 也会存入 flash info 区，地址为 0x00000398。存储时，将 **a** 系数小数点右移四位（乘 10000）存入 info 区。

实际使用中，应从 flash info 区对应地址读出 **a/b** 系数，同时将读取到的 ADC 测到的当下温度传感器值代入公式，即可计算得到当下温度值，单位为摄氏度。计算时，需注意系数 **a/b** 在保存时小数点的位移数，即 **a** 系数应除以 10000，**b** 系数除以 10。

注意，上述计算公式，基于 ADC 右对齐实现。若换成左对齐，ADC 采样值需右移 4 位后，才能代入上述公式。

## 4.9 DAC 模块

芯片内置一路 12bit DAC，输出信号的最大量程可通过寄存器 DAC\_G 设置为 1.2V/4.8V。

12bit DAC 可通过配置寄存器 DACOUT\_EN=1，将 DAC 输出送至 P0.0 管脚，可驱动 >50kΩ 的负载电阻和 50pF 的负载电容。

DAC 最大输出码率为 1MHz。

芯片上电的默认状态下，DAC 模块是关闭的。DAC 可通过设置 DAC12BPDN =1 打开，开启 DAC 模块之前，需要先开启 BGP 模块。

DAC 的输入数字信号寄存器为 SYS\_AFE\_DAC，低 12BIT 有效。信号范围是 0x000~0xFFF。0x000 对应零模拟量输出 0V，0xFFF 对应满量程模拟量输出为  $DAC_{fs}$ ，如上文所述， $DAC_{fs}$  的值可由 DAC12B\_FS 寄存器进行设置。每一档信号(LSB)所对应的模拟信号幅度为  $\frac{DAC_{fs}}{4096}$ 。若 SYS\_AFE\_DAC 的

数字值为  $Din$ ，则该数字信号所对应的 DAC 输出模拟信号为  $\frac{DAC_{fs}}{4096} * Din$

不同芯片，DAC 存在制造偏差，为抵消偏差，DAC 自带校准硬件模块。DAC 输出遵循公式  $y=ax+b$ 。 $x$  为 SYS\_AFE\_DAC 填入值（理想值对应数字量）。 $a$  来自 SYS\_AFE\_DAC\_AMC 寄存器， $b$  来自 SYS\_AFE\_DAC\_DC。硬件根据 SYS\_AFE\_DAC、SYS\_AFE\_DAC\_AMC 和 SYS\_AFE\_DAC\_DC 进行乘加从而求得校正后的数字量，送至 DAC 输入端，使得 DAC 最终输出的模拟值就是填入的理想数字量对应的值。系统上电后，默认加载 3V 的校准值，若换成其它量程，软件需读取 flash info 区域，重新加载到相应寄存器。

地址 0x00000330，为 1.2V 量程的  $a$  参数，地址 0x00000340，为 1.2V 量程的  $b$  参数。

地址 0x00000334，为 4.8V 量程的  $a$  参数，地址 0x00000344，为 4.8V 量程的  $b$  参数。

DAC 输出的模拟信号，除了可以送至 IO 口供外部模块使用外，还可通过配置寄存器连至芯片内部的 2 路比较器负端，作为比较器的基准信号使用。详见比较器章节。

DACOUT\_EN、DAC\_G 的说明见模拟寄存器 [SYS\\_AFE\\_REG3](#)

DAC12BPDN 的说明见模拟寄存器 [SYS\\_AFE\\_REG5](#)

SYS\_AFE\_DAC 的说明见寄存器 [SYS\\_AFE\\_DAC](#)



## 5 系统时钟复位

### 5.1 时钟

#### 5.1.1 时钟源

如下表所示，系统包括 5 个时钟源。其中内部低速 RC 振荡时钟 LSI/内部高速 RC 振荡时钟 HSI 不会停振。

表 5-1 系统时钟源

时钟源	频率	来源	误差	说明
LSI	64kHz	内部 RC 振荡器	全温度范围误差<50%	内部系统管理时钟，用于 WDT，复位信号的滤波和展宽
HSI	4MHz	内部 RC 振荡器	全温度范围误差<1%	可作为 PLL 源时钟
PLL	96MHz	PLL 时钟	0	PLL 输出时钟，以 HSI 作为输入，输出是 HSI 时钟的 24 倍频，作为系统主时钟。
SWD	1MHz	调试器		SWD 的 JTAG 时钟

\*SWD 时钟速率典型值，实际大小与硬件环境有关。

系统可以使用内部高速 RC 时钟 HSI 作为 PLL 的参考时钟。PLL 将 4MHz 的参考时钟 HSI 倍频 24 倍至 96MHz。

PLL 经过  $n/8$  分频后，可以得到  $96\text{MHz} \times n/8$  的高速时钟，SYS\_CLK\_CFG.CLK\_SEL 在此分频后的高速时钟与 4MHz 的 HSI 之间进行二选一，作为系统主时钟 MCLK。系统复位时，PLL 默认关闭，HSI 默认开启，系统选择 HSI 时钟，即 4MHz 作为系统主时钟进行工作，从而保证系统上电之初功耗处于较低水平。

MCLK 是系统主时钟。可以通过 [SYS\\_CLK\\_CFG](#) 寄存器 CLK\_DIV 位域控制进行  $n/8$  分频，可以产生 12,24, 48,96MHz 等频率值。SYS\_CLK\_CFG.CLK\_SEL 表示选择 PLL 或 CLK\_HS 作为系统主时钟。当 SYS\_CLK\_CFG.CLK\_SEL 为 1 时，SYS\_CLK\_CFG.CLK\_DIV 作为 PLL 的分频系数。当 SYS\_CLK\_CFG.CLK\_SEL 为 0 时，SYS\_CLK\_CFG.CLK\_DIV 不起任何作用。

表 5-2 PLL 作为 MCLK 时钟时的分频配置

SYS_CLK_CFG	分频系数	频率/MHz	是否均匀
0x0101	1/8	12	是
0x0111	2/8	24	是
0x0155	4/8	48	是
0x01FF	8/8	96	是

MCLK 时钟经过 SYS\_CLK\_FEN 寄存器控制的开关之后供给外设。I2C 时钟由 SYS\_CLK\_DIV0 寄存器控制可以进一步分频，UART 时钟由 SYS\_CLK\_DIV2 寄存器控制可以进一步分频。



PLL 输出的时钟经过 SYS\_AFE\_REG7.ADCLKSEL 控制的 2/4/8 分频后送至 ADC（典型工作频率 48MHz），即 ACLK。

内部 64kHz RC 产生一路 LSI 时钟 LCLK，主要用于 WDT 工作时钟，以及部分系统控制，复位的滤波展宽等。

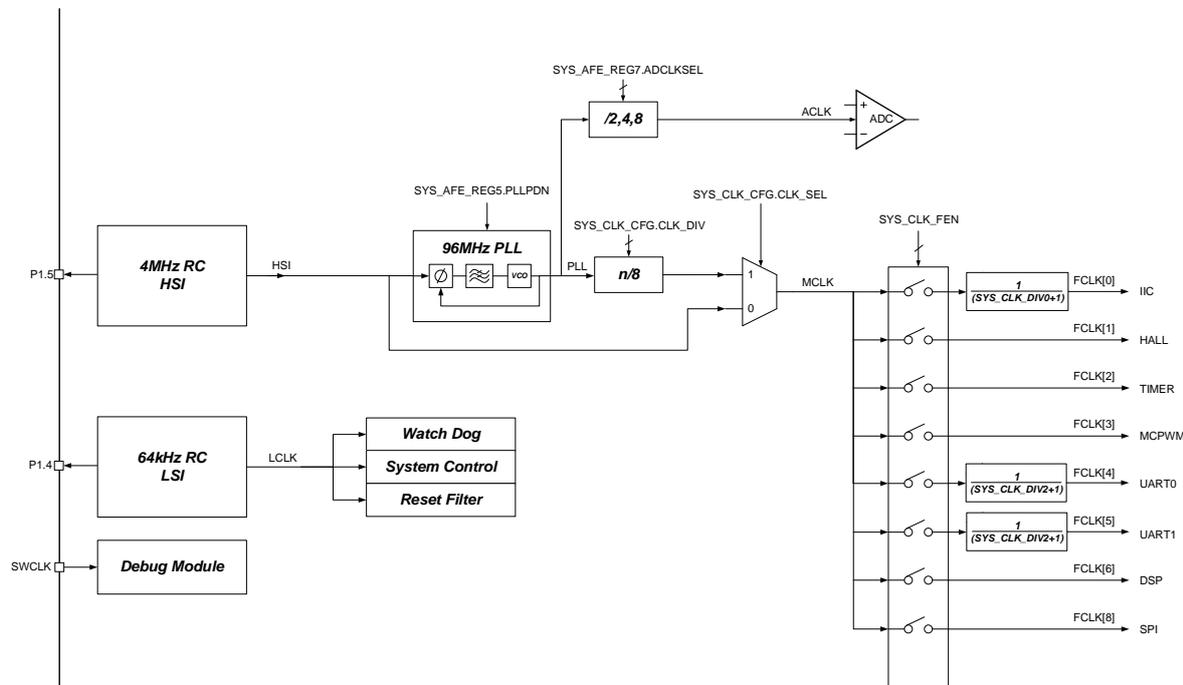


图 5-1 时钟架构

为了保证系统可靠工作，时钟系统有防止时钟被误操作关闭的机制。如当 PLL 被用作系统工作主时钟时，PLL 本身无法被关闭，作为 PLL 参考时钟的 HSI 无法被软件关闭；64kHz LSI 时钟上电即工作，且无法关闭。SWCLK 由调试器提供，频率可在调试界面进行选择。

为便于调试和出厂校正，高速 RC 时钟 HSI 和低速时钟 LSI 可以通过配置 GPIO 的第二功能通过芯片管脚输出

### 5.1.2 功耗管理及休眠唤醒

#### 5.1.2.1 休眠

MCLK 可以通过配置进行门控，从而令包括 Cortex-M0 和所有外设在内的大部分数字电路处于休眠状态。门控时，PMU 状态机依次关闭 PLL, HIS, BGP 等模拟模块，以降低功耗。

可参考休眠例程，进入休眠前关闭数字部分各个模块的时钟，关闭模拟 ADC/OPA/CMP/DAC 等模块。

需要注意的是，系统的休眠模式仅仅关闭 PLL, HSI 等高速时钟，LSI 时钟仍然存在。工作于 LSI 时钟的看门狗如果被使能，看门狗复位作为全局复位可以令系统回到初始状态重新开始工作。

向 SYS\_CLK\_SLP 寄存器写入 0xDEAD 可以令芯片准备进入休眠状态，之后立刻执行\_WFI()宏



指令使得 Cortex-M0 停止取指。

在应用程序编写中请事先配置好唤醒条件。

### 5.1.2.2 唤醒

休眠后，外部 IO 事件、内部唤醒 Timer 均可以作为唤醒源。

内部唤醒 Timer 为独立于 UTimer 模块的独立 Timer，使用 LSI 时钟，不同于系统中的通用 Timer 工作于系统主时钟。唤醒 Timer 可以使用 SYS\_RST\_CFG.WK\_INTV 配置 0.25s,0.5s,1s,2s,4s,8s,16s,32s 共 8 档唤醒时间间隔，具体请参考 SYS\_RST\_CFG 复位控制寄存器章节。

仅有 P0[1:0]、P1[1:0]四个 IO 可以作为外部唤醒 IO 使用，可以配置独立的使能和极性。具体寄存器配置请参考 7.2.15.4 WAKE\_POL 和 7.2.15.5 WAKE\_EN 章节。需要注意的是，由于外部 IO 唤醒属于电平触发，如果外部 IO 在芯片休眠之前处于唤醒电平，会导致芯片休眠后立刻唤醒。

在应用程序编写中请尽量避免上电即进入休眠状态，如果使用内部唤醒 Timer 作为唤醒源，且唤醒后立即再次睡眠，会导致普通下载器无法进行连接调试，此时需要使用芯片供应商提供的离线下载器进行应用程序擦除改写。

唤醒事件发生后，系统会先后开启 Bandgap，4MHz RC 时钟，PLL 时钟等，并且会产生 WAKEUP 中断通知 CPU，通常这个中断将 CPU 从\_WFI()指令下唤醒。

### 5.1.2.3 外设时钟门控

外设时钟由系统高速时钟 MCLK 分频而来；当外设不需要使用时可以通过配置 SYS\_CLK\_FEN 寄存器门控关闭相应的外设时钟。对于每一个外设的工作时钟，均有一个时钟门控，共有 7 路可关闭的外设时钟分别供给不同的外设模块，详见 5.3.25。门控时钟上电后默认是关闭的，使用相应外设模块之前需要由软件来开启。

I2C 使用 FCLK[0]

Hall 模块使用 FCLK [1]

Timer 模块使用 FCLK [2]

MCPWM 模块使用 FCLK [3]

UART0/UART1 分别使用 FCLK [4]/ FCLK [5]

协处理器使用 FCLK [6]

SPI 使用 FCLK [8]

### 5.1.2.4 外设时钟分频

部分外设拥有独立的时钟分频模块使得该模块可以工作在合适的时钟频率上。

其中 I2C 使用 SYS\_CLK\_DIV[0]作为分频系数，UART0/1 共享 SYS\_CLK\_DIV[2]作为分频系数。UART 的波特率在 UART 模块内部还有一个额外的分频器，详见 13.3.4 和 13.3.3。



## 5.2 复位

### 5.2.1 复位源

芯片的复位来源包括硬件复位与软件复位。

#### 5.2.1.1 硬件复位

如表 5-3 硬件复位源所示，系统包括 4 个硬件复位源，产生的复位均为芯片全局复位，复位产生后芯片程序计数器回到 0 地址，所有寄存器恢复到默认值。4 个硬件复位均为低电平有效。

表 5-3 系统复位源

名称	来源	说明
LPORn	内部 1.5V 电源管理	监控 1.5V 数字电源，低于 1.25V 时产生复位
HPORn	内部 3.3V 电源管理	监控 3.3V 电源，低于 2.2V 时产生复位
RESETn	外部按键	外部 RC 组成按键复位电路
WDTn	硬件看门狗	如果不进行软件喂狗则定时产生复位，复位间隔可配置

##### 5.2.1.1.1 硬件复位架构

如下图所示，LPORn/HPORn 来自内部模拟电路，RSTn 来自外部按键。

WDTn 为 1 个 LSI 时钟周期宽度信号，是内部数字信号。WDTn 信号在 Debug 模式下会被屏蔽。

经过滤波展宽预处理的复位信号进行与运算得到一个复位信号。

P0.2 引脚低于 16us 宽度的外部复位会被复位滤波滤除，要求可靠外部复位宽度大于 200us。

4 个复位信号复位等级和作用域一致，均为全局复位。

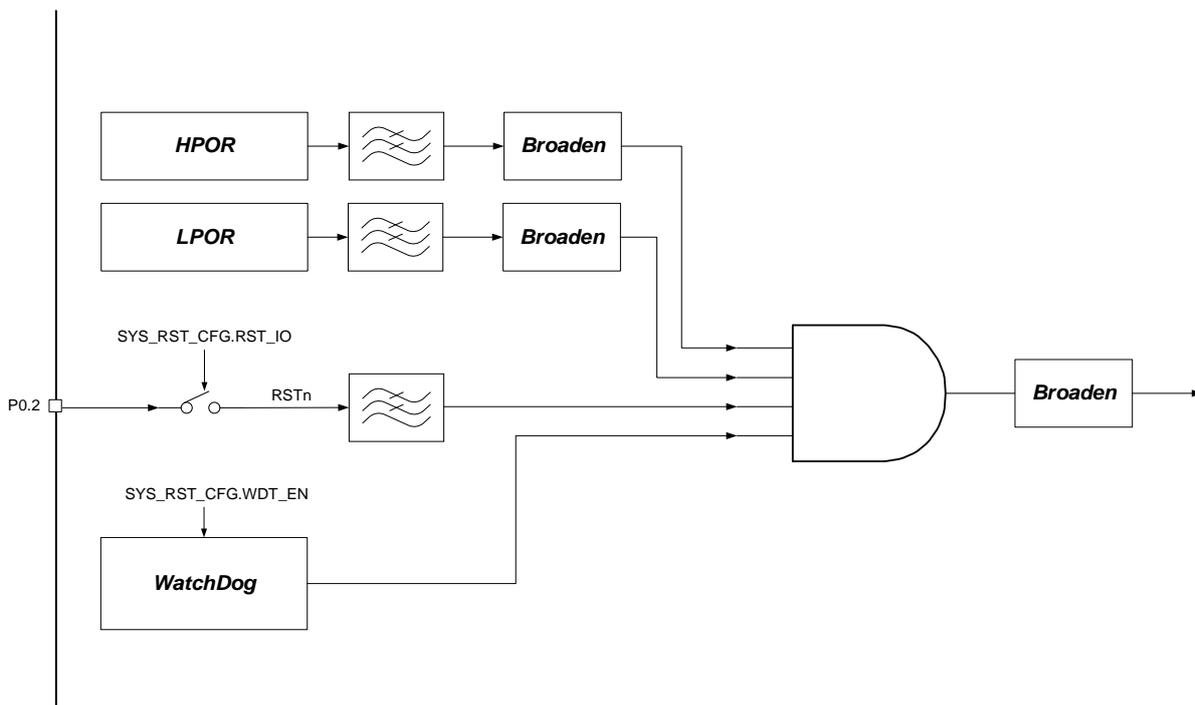


图 5-2 复位架构

5.2.1.1.2 硬件复位记录

**SYS\_RST\_SRC** 寄存器用于保存硬件复位事件，当某个硬件复位发生后，**SYS\_RST\_SRC** 对应位置位。**SYS\_RST\_SRC** 寄存器本身无法被复位信号复位，只能通过向 **SYS\_CLR\_RST** 寄存器写入 **0xCA40** 清空记录，复位记录可以方便地了解是否发生以及发生过何种复位。

5.2.1.2 软件复位

CPU 的软复位操作可以使程序计数器(PC: Program Counter)回到 0 地址，但对所有外设中的寄存器没有影响。

在集成开发环境(IDE: Integrated Development Environment)中的调试模式下，点击 **Reset** 与 CPU 软复位操作作用相同，仅仅使得 PC 回到 0 地址，对外设中的寄存器没有影响。但如果在 bootloader 中进行了外设模块的软复位，则会使得外设寄存器被复位为默认值。具体 bootloader 实现请咨询芯片供应商。

部分外设模块有模块级软复位，可以使用 **SYS\_SFT\_RST** 寄存器进行复位，写入对应的位，可以将模块状态机恢复到初始状态，同时将模块的寄存器恢复到默认值，详见 5.3.27。

5.2.2 复位作用域

表 5-4 复位作用域

复位源	作用域
LPORn	内部 1.5V 电源管理，全局复位
HPOR	内部 3.3V 电源管理，全局复位，除极少数寄存器
RSTn	外部按键，全局复位，除极少数寄存器



WDTn	硬件看门狗，全局复位，除极少数寄存器
SYS_SFT_RST.ADC_SFT_RST	ADC 模块
SYS_SFT_RST.SPI_SFT_RST	SPI 模块
SYS_SFT_RST.DSP_SFT_RST	协处理器模块
SYS_SFT_RST.UART1_SFT_RST	UART1 模块
SYS_SFT_RST.UART0_SFT_RST	UART0 模块
SYS_SFT_RST.MCPWM_SFT_RST	MCPWM 模块
SYS_SFT_RST.UTIMER_SFT_RST	UTIMER 模块
SYS_SFT_RST.HALL_SFT_RST	HALL 模块
SYS_SFT_RST.I2C_SFT_RST	I2C 模块
NVIC_SystemReset();	CPU 软复位，仅复位 CPU 内核，将 PC 重置为 0，所有外设寄存器值仍然维持。

SYS\_RST\_CFG.RST\_IO（用于控制 P0[2] 作为 GPIO 使用还是作为外部复位脚使用），SYS\_RST\_CFG.WDT\_EN(用于控制看门狗使能),复位记录寄存器 SYS\_RST\_SRC, 只受 LPOR 复位影响, 不被其他复位信号复位。

通常，全局复位会复位全芯片寄存器，包括 CPU 内核寄存器以及所有外设寄存器，除上述极少数寄存器。

由于 CPU 软复位仅仅复位 CPU 内核，而不复位外设寄存器，因此建议重新烧录程序后使用掉电重新上电或外部复位的方式重置外设寄存器。

Flash 存储内容，SRAM 存储内容不受复位影响。

## 5.3 系统寄存器

### 5.3.1 地址分配

系统模块寄存器基地址为 0x40000000。

表 5-5 系统控制寄存器

名称	偏移	说明
SYS_WDT_PSW	0x00	看门狗密码寄存器
SYS_WDT_CLR	0x04	看门狗清零寄存器
SYS_WDT_TH	0x08	看门狗计数器门限值
SYS_WDT_CNT	0x0C	看门狗计数器当前计数值
SYS_OPA_SEL	0x14	运放通道选择寄存器
SYS_AFE_REG0	0x20	模拟配置寄存器 0
SYS_AFE_REG1	0x24	模拟配置寄存器 1
--	--	--
SYS_AFE_REG3	0x2C	模拟配置寄存器 3
--	--	--
SYS_AFE_REG5	0x34	模拟配置寄存器 5



SYS_AFE_REG6	0x38	模拟配置寄存器 6
SYS_AFE_REG7	0x3C	模拟配置寄存器 7
--	--	--
SYS_TMP_A	0x54	温度传感器系数 A
SYS_TMP_B	0x58	温度传感器系数 B
SYS_AFE_DAC	0x60	DAC 数字量寄存器
SYS_DAC_AMC	0x64	DAC 增益校正寄存器
SYS_DAC_DC	0x68	DAC 直流偏置寄存器
SYS_CLK_CFG	0x80	时钟控制寄存器
SYS_RST_CFG	0x84	复位控制寄存器
SYS_RST_SRC	0x88	复位源记录寄存器
SYS_CLR_RST	0x8C	复位源记录清除寄存器
SYS_CLK_DIV0	0x90	外设时钟分频寄存器 0
SYS_CLK_DIV2	0x98	外设时钟分频寄存器 2
SYS_CLK_FEN	0x9C	外设时钟门控寄存器
SYS_CLK_SLP	0xA0	休眠寄存器
SYS_SFT_RST	0xAC	软复位寄存器
SYS_PROTECT	0xB0	写保护寄存器

### 5.3.2 SYS\_WDT\_PSW 看门狗密码寄存器

地址：0x4000\_0000

复位值：0x0

表 5-6 SYS\_WDT\_PSW 看门狗密码寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_WDT_PSW															
WO															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	SYS_WDT_PSW	写入 0xA6B4，才能对 WDT_CLR/WDT_TH 等进行写操作，对 SYS_WDT_CLR 或 SYS_WDT_TH 的写操作会将密码清空，因此每次对看门狗进行写操作前都需要写入密码

### 5.3.3 SYS\_WDT\_CLR 看门狗清零寄存器

地址：0x4000\_0004

复位值：0x0



表 5-7 SYS\_WDT\_CLR 看门狗清零寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_WDT_CLR															
WO															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	SYS_WDT_CLR	写入字节 16'b0111_1001_1000_110B <sub>0</sub> ，高 15 位为密码，密码正确时，B[0]才能写入。 B[0]为 CLR，写入 1，则重置 WDT 计数器为 TH，且该 bit 写入后自动清零。

### 5.3.4 SYS\_WDT\_TH 看门狗门限寄存器

地址：0x4000\_0008

复位值：0x001F\_F000

表 5-8 SYS\_WDT\_TH 看门狗门限寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											SYS_WDT_TH				
											RW				
											0x1F				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_WDT_TH															
RW															
0xF															

位置	位名称	说明
[31:21]		未使用
[20:12]	SYS_WDT_TH	看门狗门限值，看门狗以 LRC 时钟从 TH 开始计数，计数至 0 产生复位。向该寄存器段写入 9'h000，会被硬件强制改写为 9'h001。先向 SYS_WDT_PSW 写入正确密码才能改写 SYS_WDT_TH 寄存器。改写 SYS_WDT_TH 同样具有重置看门狗计数器的作用，看门狗会从新的 TH 开始计数。
[11:0]		未使用

为防止 SYS\_WDT\_TH 被写入为 0，当软件写入值为全 0 时，硬件会强制改写为 0x1000，且寄存器低 12 位恒为 0，折算最小复位时间间隔为  $4096/64\text{kHz} = 0.064\text{s}$ 。复位时间范围 0.064~32s，以 0.064s 为最小步长连续可配置。



在全温度范围内，64k RC 时钟还会存在±50%左右的偏差，所以保险起见，建议看门狗喂狗时间要比 64k 计算的理论值再快 20~30%。

### 5.3.5 SYS\_WDT\_CNT 看门狗计数值寄存器

地址：0x4000\_000C

复位值：0x0

表 5-9 SYS\_WDT\_CNT 看门狗计数值寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											SYS_WDT_CNT				
											RO				
											0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_WDT_CNT															
RO															
0															

位置	位名称	说明
[31:21]		未使用
[20:0]	SYS_WDT_CNT	看门狗当前计数值，此数值≤SYS_WDT_TH。

### 5.3.6 SYS\_OPA\_SEL 运放通道选择寄存器

地址：0x4000\_0014

复位值：0x0

表 5-10 运放通道选择寄存器 SYS\_OPA\_SEL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
							OPA_SW_SEL_EN								OPA_SEL_EN		
								RW								RO	
								0								0	

位置	位名称	说明
[31:9]		未使用
[8]	OPA_SW_SEL_EN	ADC_CH8/9 信号来源 0: 对应 OPA2/3 输出



		1: 对应 IO 口的 ADC_CH8/9
[7:1]		未使用
[0]	OPA_SEL_EN	OPA2 可使用 OPA_A, OPA3 可使用 OPA_B, 使能开关, 高电平有效, 此 BIT 寄存器只读

当 OPA\_SEL\_EN=1 时, 允许使用 OPA2/3 的运放通道。

表 5-11 ADC 采样信号来源与寄存器设置

其中 CH8/CH9 是支持复用的。

ADC_CHNx<3:0>	ADC 通道正端选择	0000: OPA0 Output; 0001: OPA1 Output; 0010: ADC_CH2; 0011: ADC_CH3; 0100: ADC_CH4; 0101: ADC_CH5; 0110: ADC_CH6; 0111: ADC_CH7; 1000: ADC_CH8/OPA2 Output; 1001: ADC_CH9/OPA3 Output ; 1010: ADC_CH10; 1011: ADC_CH11; 1100: ADC_CH12; 1101: ADC_CH13; 1110: Temperature Sensor; 1111: GND;
---------------	------------	--

表 5-12 ADC 涉及 OPA 复用的采样通道设置

OPA_SEL_EN	OPA_SW_SEL_EN	ADC 设置采样通道	ADC 实际采样通道
0	X	8	ADC_CH8
		9	ADC_CH9
		8	ADC_CH8
		9	ADC_CH9
1	0	8	OPA2_OUT
	0	9	OPA3_OUT
	1	8	ADC_CH8
	1	9	ADC_CH9

当 OPA\_SEL\_EN=0, 或者 OPA\_SEL\_EN=1 且 OPA\_SW\_SEL\_EN=1 时, ADC 在设置采样通道号为 8/9 时, 实际采样 IO 口的 ADC\_CH8/9。

当 OPA\_SEL\_EN=1 且 OPA\_SW\_SEL\_EN=0 时, ADC 在设置采样通道号为 8/9 时, 实际采样 OPA2/3



的输出。

### 5.3.7 模拟寄存器概述

模拟寄存器 SYS\_AFE\_REG0~ SYS\_AFE\_REG7，对应地址为 0x40000020~0x4000003C。其中保留寄存器(Res)必须全部配置为 0（芯片上电后会被复位为 0）。其他寄存器根据应用场合需要进行配置。

下面是各个模拟寄存器的详细说明。

### 5.3.8 SYS\_AFE\_REG0 模拟配置寄存器 0

地址：0x4000\_0020

复位值：0x0

表 5-13 SYS\_AFE\_REG0 模拟配置寄存器 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resv.	Resv.	Resv.	Resv.	Resv.	Resv.	GA_AD	OPAOUT_EN	RES_OPA_A	RES_OPA_B						
RW	RW	RW													
0	0	0	0	0	0	0	0	0	0						

位置	位名称	说明
[31:16]		未使用
[15:14]	RESERVED	必须为 0
[13:12]	RESERVED	必须为 0
[11]	RESERVED	必须为 0
[10]	RESERVED	必须为 0
[9:7]	RESERVED	必须为 0
[6]	GA_AD	ADC 增益选择 0: 2/3 倍 1: 1 倍
[5:4]	OPAOUT_EN	使能 OPAx 输出信号送至 IO 口 P2.7 00: 不输出; 01: 输出 OPA0 信号到 IO 口 P2.7; 10: 输出 OPA1 信号到 IO 口 P2.7; 11: 不允许此配置
[3:2]	RES_OPA_B	运放 B 反馈电阻 00: 200k:10.6k 01: 190k:20.6k



		10: 180k:30.6k 11: 170k:40.6k 运放 B 根据配置,可以在不同时刻对 OPA1/OPA3 通道的输入进行放大
[1:0]	RES_OPA_A	运放 A 反馈电阻 00: 200k:10.6k 01: 190k:20.6k 10: 180k:30.6k 11: 170k:40.6k 运放 A 根据配置,可以在不同时刻对 OPA0/OPA2 通道的输入进行放大

### 5.3.9 SYS\_AFE\_REG1 模拟配置寄存器 1

地址: 0x4000\_0024

复位值: 0x0

表 5-14 SYS\_AFE\_REG1 模拟配置寄存器 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resv.	Resv.		Resv.												CMP_FT
RW	RW		RW												RW
0	0		0												0

位置	位名称	说明
[31:16]		未使用
[15]	RESERVED	必须为 0
[14]	RESERVED	必须为 0
[13]		未使用
[12]	RESERVED	必须为 0
[11:1]		未使用
[0]	CMP_FT	比较器快速比较使能开关 1: 使能比较器比较速度小于 30ns 0: 不使能

### 5.3.10 SYS\_AFE\_REG3 模拟配置寄存器 3

地址: 0x4000\_002C

复位值: 0x0



表 5-15 SYS\_AFE\_REG3 模拟配置寄存器 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC_G	CMP1_SEL_P		DACOUT_EN	CMP0_SEL_P		CMP_HYS	Resv.	CMP1_SEL_N		CMP0_SEL_N				LDOOUT_EN	
RW	RW		RW	RW		RW	RW	RW		RW				RW	
0	0		0	0		0	0	0		0				0	

位置	位名称	说明
[31:16]		未使用
[15]	DAC_G	DAC 输出档位配置 0: DAC 输出档位, 满量程为 1.2V 1: DAC 输出档位, 满量程为 4.8V
[14:12]	CMP1_SEL_P	比较器 1 信号正端选择 000: 连 CMP1_IP0 001: 连 OPA1_IP 010: 连 OPA1_OUT 011: 连 CMP1_IP1 100: 连 CMP1_IP2 101: 连 CMP1_IP3 110: 连 AVSS 111: 连 AVSS 说明:上述除 AVSS/OPA1_OUT 外都为管脚名称, 请参看 datasheet 里管脚定义章节。
[11]	DACOUT_EN	DAC 输出到 IO 使能 0: 不使能 1: 输出 DAC 结果到 IO 口 P0.0
[10:8]	CMP0_SEL_P	比较器 0 信号正端选择 000: 连 CMP0_IP0 001: 连 OPA0_IP 010: 连 OPA0_OUT 011: 连 CMP0_IP1 100: 连 CMP0_IP2 101: 连 CMP0_IP3 110: 连 CMP0_IP4 111: 连 AVSS 说明:上述除 OPA0_OUT/OPA1_OUT 外都为管脚名称, 请参看 datasheet 里管脚定义章节。
[7]	CMP_HYS	比较器回差选择, 默认配置为 0 0: 20mv; 1: 0mv
[6]	RESERVED	必须为 0



[5:4]	CMP1_SELN	比较器 1 信号负端选择 00: 连 CMP1_IN 01: 连 REF 10: 连 DAC 输出 11: 连 HALL1_MID 说明:上述 CMP1_IN 为管脚名称, 请参看 datasheet 里管脚定义章节; REF 为芯片内部 1.2V BANDGAP 基准源; DAC 输出即为芯片内部 DAC 模块输出模拟信号; HALL1_MID 为 CMP1_IP1, CMP1_IP2, CMP1_IP3 信号经电阻星形连接后得到的平均值。
[3:2]	CMP0_SELN	比较器 0 信号负端选择 00: 连 CMP0_IN 01: 连 REF 10: 连 DAC 输出 11: 连 HALL0_MID 说明:上述 CMP0_IN 为管脚名称, 请参看 datasheet 里管脚定义章节; REF 为芯片内部 1.2V BANDGAP 基准源; DAC 输出即为芯片内部 DAC 模块输出模拟信号; HALL0_MID 为 CMP0_IP1, CMP0_IP2, CMP0_IP3 信号经电阻星形连接后得到的平均值。
[1]		未使用
[0]	LDOOUT_EN	LDO 输出到 IO 使能, 默认配置为不输出 0: 不输出 1: 输出 LDO 到 IO 口 P2.7

### 5.3.11 SYS\_AFE\_REG5 模拟配置寄存器 5

地址: 0x4000\_0034

复位值: 0x0

表 5-16 SYS\_AFE\_REG5 模拟配置寄存器 5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLDPN		TMPPDN	DAC12BPDN		RCHPD		BGPPD	CMP1PDN	CMP0PDN			OPA1PDN	OPA0PDN		ADCPDN
RW		RW	RW		RW		RW	RW	RW			RW	RW		RW
0		0	0		0		0	0	0			0	0		0

位置	位名称	说明
[31:16]		未使用
[15]	PLLDPN	PLL 开关控制, 默认配置为关闭 PLL 0: 关闭 PLL 1: 开启 PLL
[14]		未使用



[13]	TMPPDN	温度传感器开关控制，默认配置为关闭温度传感器 0: 关闭温度传感器 1: 开启温度传感器
[12]	DAC12BPDN	12BIT DAC 开关控制，默认配置为关闭 DAC 0: 关闭 DAC 1: 开启 DAC
[11]		未使用
[10]	RCHPD	RCH 时钟开关控制，默认配置为开启 RCH 0: 开启 RCH 1: 关闭 RCH
[9]		未使用
[8]	BGPPD	BGP 开关控制，默认配置为开启 BGP 0: 开启 BGP 1: 关闭 BGP
[7]	CMP1PDN	CMP1 开关控制，默认配置为关闭 CMP1 0: 关闭 CMP1 1: 开启 CMP1
[6]	CMP0PDN	CMP0 开关控制，默认配置为关闭 CMP0 0: 关闭 CMP0 1: 开启 CMP0
[5:4]		未使用
[3]	OPA1PDN	OPA1 开关控制，默认配置为关闭 OPA1 0: 关闭 OPA1 1: 开启 OPA1
[2]	OPA0PDN	OPA0 开关控制，默认配置为关闭 OPA0 0: 关闭 OPA0 1: 开启 OPA0
[1]		未使用
[0]	ADCPDN	ADC 开关控制，默认配置为关闭 ADC 0: 关闭 ADC 1: 开启 ADC

如果 SYS\_CLK\_CFG 选择 PLL 时钟，则 PLLPDN 是被硬件控制的，软件配置 PLLPDN 关闭 PLL 无效。关闭 PLL 需要 PLLPDN=0，且 SYS\_CLK\_CFG 不选择 PLL 作为芯片主时钟，这两个条件都须满足。同理如果 SYS\_CLK\_CFG 选择 HRC 时钟，则 RCHPD 是被硬件控制的，软件直接配置 RCHPD 关闭 RCH 无效。关闭 PLL 需要 RCHPD=1，且芯片进入休眠。如果芯片主时钟为 PLL 时钟，且 HRC 为 PLL 参考时钟，则 RCH 也是被硬件控制的。由于 RCH 和 PLL 依赖 BGP，所以 BGPPD 也是硬件控制的，在芯片使用了 RCH 或 PLL 时，软件直接配置 BGPPD 关闭 BGP 无效。关闭 BGP 需要先顺序关闭 PLL 和 RCH，且芯片进入休眠。

### 5.3.12 SYS\_AFE\_REG6 模拟配置寄存器 6

地址：0x4000\_0038

复位值：0x0



表 5-17 SYS\_AFE\_REG6 模拟配置寄存器 6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
									CH16EN							
									RW							
									0							

位置	位名称	说明
[31:7]		未使用
[6]	CH15EN	ADC CH15 通道信号使能 0: ADC CH15 对应内部 GND 1: ADC CH15 对应 SWCK IO
[5:0]		未使用

### 5.3.13 SYS\_AFE\_REG7 模拟配置寄存器 7

地址: 0x4000\_003C

复位值: 0x0

表 5-18 SYS\_AFE\_REG7 模拟配置寄存器 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						SAMP_TIME						ADCLKSEL			
						RW						RW			
						0						0			

位置	位名称	说明
[31:14]		未使用
[13:8]	SAMP_TIME	000000->111111:对应采样时间从 4->68 个 ADC 时钟周期, 逐次增加
[7:6]		未使用
[5:4]	ADCLKSEL	ADC 时钟频率选择 00: 48MHz 01: 不允许此配置 10: 12MHz 11: 24MHz
[3:0]		未使用

### 5.3.14 SYS\_TMP\_A 温度传感器系数 A 寄存器

地址: 0x4000\_0054



复位值：0x0

表 5-19 SYS\_TMP\_A 温度传感器系数 A 寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMP_GAIN_A															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TMP_GAIN_A	温度传感器增益校正系数 A

### 5.3.15 SYS\_TMP\_B 温度传感器系数 B 寄存器

地址：0x4000\_0058

复位值：0x0

表 5-20 SYS\_TMP\_B 温度传感器系数 B 寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMP_SIGN				TMP_OFFSET_B											
R				RW											
0				0											

位置	位名称	说明
[31:16]		未使用
[15:12]	TMP_SIGN	温度传感器 offset 校正系数 B 符号扩展位
[11:0]	TMP_OFFSET_B	温度传感器 offset 校正系数 B

### 5.3.16 SYS\_AFE\_DAC DAC 数字量寄存器

地址：0x4000\_0060

复位值：0x0

表 5-21 SYS\_AFE\_DACDAC 数字量寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC_IN															
RW															
0															

位置	位名称	说明
----	-----	----



[31:12]		未使用
[11:0]	DAC_IN	DAC 待转换的数字量输入

### 5.3.17 SYS\_AFE\_DAC\_AMC DAC 增益校正寄存器

地址：0x4000\_0064

复位值：0x0

表 5-22 SYS\_AFE\_DAC\_AMC DAC 增益校正寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											DAC_AMC				
											RW				
											0x200				
位置	位名称		说明												
[31:10]			未使用												
[9:0]	DAC_AMC		DAC 增益校正，10bit 无符号定点数，B[9]为整数部分，B[8:0]为小数部分												

### 5.3.18 SYS\_AFE\_DAC\_DC DAC 直流偏置寄存器

地址：0x4000\_0068

复位值：0x0

表 5-23 SYS\_AFE\_DAC\_DC DAC 直流偏置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											DAC_DC				
											RW				
											0				

位置	位名称		说明												
[31:8]			未使用												
[7:0]	DAC_DC		DAC 直流偏置，8bit 有符号数，B[7]为符号位												

DAC 增益校正，记模拟输出的 12bitDAC 数值为 DAC\_raw，经过校正后的 12bitDAC 数值为 DAC\_cali

$$\text{DAC\_cali} = \text{DAC\_raw} * \text{DAC\_AMC} - \text{DAC\_DC};$$

其中 DAC\_AMC 为 DAC 增益校正系数，为 10bit 定点无符号数，B[9]为整数，B[8:0]为小数，大小为 1 左右，例如  $\text{DAC\_AMC} = 10'b10_0001_0000 = 1+1/32$ ，或

$$\text{DAC\_AMC} = 10'b01_1110_1100 = 1-5/128$$

DAC\_DC 位 DAC 直流偏置，为 8bit 有符号整数。



增益校正计算结果进行截断保留，最终 DAC\_cal 仍为 12bit 整数。

且增益校正加直流偏置校正后的数值会进行饱和处理，最大为 0xff，最小为 0x000。

需要注意的是，DAC 有两个输出档位，系统上电后，加载默认档位的 DAC 校准值，若切换到其它档位，请使用原厂提供的库函数。

### 5.3.19 SYS\_CLK\_CFG 时钟控制寄存器

地址：0x4000\_0080

复位值：0x0

表 5-24 SYS\_CLK\_CFG 时钟控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								CLK_SEL	CLK_DIV							
								RW	RW							
								0	0							

位置	位名称	说明
[31:9]		未使用
[8]	CLK_SEL	系统时钟 MCLK 的来源选择信号，默认选择 CLK_HS 0: CLK_HS 1: PLL 注意，PLL 在上电后默认关闭，需要软件来开启。
[7:0]	CLK_DIV	PLL 输出分频控制，默认为 0x00: 1/8 分频 0x01: 1/8 分频 0x11: 1/4 分频 0x55: 1/2 分频 0xFF: 1/1 分频 不推荐其它配置值。

当 CLK\_SEL = 0 时，MCLK 选择 CLK\_HS 时钟（4MHz），此时 SYS\_CLK\_CFG[7:0]的分频系数无效。最终输出的系统时钟频率即为 4MHz。

### 5.3.20 SYS\_RST\_CFG 复位控制寄存器

地址：0x4000\_0084

复位值：0x40

表 5-25 SYS\_RST\_CFG 复位控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SWDMUX	RST_IO	WK_INTV		WDT_EN	
										RW	RW	RW		RW	
										1	0	0		0	



位置	位名称	说明
[31:7]		未使用
[6]	SWDMUX	SWD 复用控制信号，默认配置为 SWD 0: P2.13 和 P2.0 作为正常 GPIO 使用 1: P2.13 复用为 SWCLK, P2.0 复用为 SWDIO 用作 SWD 信号时，默认开启上拉，不受软件控制 用作 GPIO 时，上拉受 GPIO2_PUE[0]和 GPIO2_PUE[13]控制
[5]	RST_IO	RSTn/P0.2 复用控制信号，默认配置为 RSTn 0: RSTn 1: P0.2 注意，上电后默认是 RSTn，后续软件可使能此位，RSTn 功能失效。 用作 RSTn 时，默认开启上拉，不受软件控制 用作 GPIO 时，上拉受 GPIO0_PUE[2]控制。
[4:2]	WK_INTV	休眠唤醒间隔设置 000: 0.25S                      100: 4S 001: 0.5S                        101: 8S 010: 1S                            110: 16S 011: 2S                            111: 32S
[1]		未使用
[0]	WDT_EN	看门狗使能控制信号，默认为关闭看门狗 0: 关闭看门狗 1: 开启看门狗

为了安全起见，上电后 16ms 内，SWD 的 IO 不能切换为 GPIO 功能，只能作为 SWD 使用。即使软件改写了 SYS\_IO\_CFG [6]将不起作用，需要在 16ms 之后再次对 SYS\_IO\_CFG [6]改写。现象为：在 16ms 内，软件可以向 SYS\_IO\_CFG[6] 1'b1，但读回该 bit 仍为 1'b0，如果经过 16ms 后再次写入 1'b1，则可读回 1'b1。即在上电后的 16ms 内，软件 SYS\_IO\_CFG[6]写入 1'b1 无效，上电 16ms 后，软件对 SYS\_IO\_CFG[6]写入 1'b1 才有效。这是为了防止应用上电即切换 SWD 功能为 GPIO，导致芯片后续无法通过 SWD 进行擦写烧录 flash。

注意，P2.13 和 P2.0 切换为 GPIO 功能后，除非应用软件特殊设计，否则只能使用原厂下载器擦除下载程序。

### 5.3.21 SYS\_RST\_SRC 复位源记录寄存器

地址：0x4000\_0088

复位值：0x0

表 5-26 SYS\_RST\_SRC 复位源记录寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												WDT_RST_RCD	KEY_RST_RCD	HPOR_RST_RCD	LPOR_RST_RCD
												RO	RO	RO	RO
												0	0	0	0



位置	位名称	说明
[31:4]		未使用
[3]	WDT_RST_RCD	看门狗复位发生标志，高有效
[2]	KEY_RST_RCD	按键复位发生标志，高有效
[1]	HPOR_RST_RCD	HPOR 复位发生标志，高有效
[0]	LPOR_RST_RCD	LPOR 复位发生标志，高有效

### 5.3.22 SYS\_CLR\_RST 复位源记录清除寄存器

地址：0x4000\_008C

复位值：0x0

表 5-27 SYS\_CLR\_RST 复位源记录清除寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSW															
WO															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	PSW	写入 0xCA40，清除复位标志记录 注意，由于复位记录工作于低速时钟域，清除执行完成需要一定时间，不应清除后立即读记录状态。

### 5.3.23 SYS\_CLK\_DIV0 外设时钟分频寄存器 0

地址：0x4000\_0090

复位值：0x0

表 5-28 SYS\_CLK\_DIV0 外设时钟分频寄存器 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV0															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DIV0	I2C 工作时钟=MCLK/(CLK_DIV0+1) 其中 MCLK 由 SYS_CLK_CFG 分频系数决定。



## 5.3.24 SYS\_CLK\_DIV2 外设时钟分频寄存器 2

地址：0x4000\_0098

复位值：0x0

表 5-29 SYS\_CLK\_DIV2 外设时钟分频寄存器 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV2															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DIV2	UART 工作时钟= $MCLK/(CLK\_DIV2+1)$ UART0/UART1 共享此分频配置，波特率根据 UART 波特率寄存器进一步分频，其中 MCLK 由 SYS_CLK_CFG 分频系数决定。

## 5.3.25 SYS\_CLK\_FEN 外设时钟门控寄存器

地址：0x4000\_009C

复位值：0x0

表 5-30 SYS\_CLK\_FEN 外设时钟门控寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
							SPI_CLK_EN									
							RW	DSP_CLK_EN	UART1_CLK_EN	UART0_CLK_EN	MCPWM_CLK_EN	UTTIMER_CLK_EN	HALL_CLK_EN	I2C_CLK_EN		
							0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:9]		未使用
[8]	SPI_CKL_EN	SPI 模块时钟控制信号，默认关闭 SPI 模块时钟 1: 使能 SPI 模块时钟 0: 关闭 SPI 模块时钟
[7]	--	未使用
[6]	DSP_CLK_EN	协处理器模块时钟控制信号，默认关闭协处理器模块时钟 1: 使能协处理器模块时钟 0: 关闭协处理器模块时钟
[5]	UART1_CLK_EN	UART1 模块时钟控制信号，默认关闭 UART1 模块时钟



		1: 使能 UART1 模块时钟 0: 关闭 UART1 模块时钟
[4]	UART0_CLK_EN	UART0 模块时钟控制信号，默认关闭 UART0 模块时钟 1: 使能 UART0 模块时钟 0: 关闭 UART0 模块时钟
[3]	MCPWM_CLK_EN	MCPWM 模块时钟控制信号，默认关闭 MCPWM 模块时钟 1: 使能 MCPWM 模块时钟 0: 关闭 MCPWM 模块时钟
[2]	UTIMER_CLK_EN	UTIMER 模块时钟控制信号，默认关闭 UTIMER 模块时钟 1: 使能 UTIMER 模块时钟 0: 关闭 UTIMER 模块时钟
[1]	HALL_CLK_EN	HALL 模块时钟控制信号，默认关闭 HALL 模块时钟 1: 使能 HALL 模块时钟 0: 关闭 HALL 模块时钟
[0]	I2C_CLK_EN	I2C 模块时钟控制信号，默认关闭 I2C 模块时钟 1: 使能 I2C 模块时钟 0: 关闭 I2C 模块时钟

注意，上述每个模块的时钟为各自模块内部电路的工作时钟，即使不开启各自模块的时钟，也不影响软件访问各自模块的寄存器。

### 5.3.26 SYS\_CLK\_SLP 休眠寄存器

地址：0x4000\_00A0

复位值：0x0

表 5-31 SYS\_CLK\_SLP 休眠寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSW															
WO															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	PSW	写入密码 0xDEAD，系统关闭高速时钟，进入休眠状态

### 5.3.27 SYS\_SFT\_RST 软复位寄存器

地址：0x4000\_00AC

复位值：0x0

表 5-32 SYS\_SFT\_RST 软复位寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---



	ADC_SFT_RST	SPI_SFT_RST		DSP_SFT_RST	UART1_SFT_RST	UART0_SFT_RST	MCPWM_SFT_RST	UTIMER_SFT_RST	HALL_SFT_RST	I2C_SFT_RST
	WO	WO		WO	WO	WO	WO	WO	WO	WO
	0	0		0	0	0	0	0	0	0

位置	位名称	说明
[31:10]		未使用
[9]	ADC_SFT_RST	ADC 模块软复位控制信号，默认配置是不复位 ADC 模块 1: 复位 ADC 模块 0: 释放 ADC 模块
[8]	SPI_SFT_RST	SPI 模块软复位控制信号，默认配置是不复位 SPI 模块 1: 复位 SPI 模块 0: 释放 SPI 模块
[7]		未使用
[6]	DSP_SFT_RST	协处理器模块软复位控制信号，默认配置是不复位协处理器模块 1: 复位协处理器模块 0: 释放协处理器模块
[5]	UART1_SFT_RST	UART1 模块软复位控制信号，默认配置是不复位 UART1 模块 1: 复位 UART1 模块 0: 释放 UART1 模块
[4]	UART0_SFT_RST	UART0 模块软复位控制信号，默认配置是不复位 UART0 模块 1: 复位 UART0 模块 0: 释放 UART0 模块
[3]	MCPWM_SFT_RST	MCPWM 模块软复位控制信号，默认配置是不复位 MCPWM 模块 1: 复位 MCPWM 模块 0: 释放 MCPWM 模块
[2]	UTIMER_SFT_RST	UTIMER 模块软复位控制信号，默认配置是不复位 UTIMER 模块 1: 复位 UTIMER 模块 0: 释放 UTIMER 模块
[1]	HALL_SFT_RST	HALL 模块软复位控制信号，默认配置是不复位 HALL 模块 1: 复位 HALL 模块 0: 释放 HALL 模块
[0]	I2C_SFT_RST	I2C 模块软复位控制信号，默认配置是不复位 I2C 模块 1: 复位 I2C 模块 0: 释放 I2C 模块

注意，模块软复位在 SYS\_SFT\_RST 对应位写入 1 后会保持在复位状态，需要再次写入 0 才能解除复位状态。



### 5.3.28 SYS\_PROTECT 写保护寄存器

地址：0x4000\_00B0

复位值：0x0

表 5-33 SYS\_PROTECT 写保护寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSW															
W0															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	PSW	除 SYS_AFE_DAC、SYS_AFE_DAC_AMC、SYS_AFE_DAC_DC 以及 SYS_WDT_PSW、SYS_WDT_CLR/SYS_WDT_TH/SYS_WDT_CNT 等看门狗寄存器以外,其他系统寄存器受 SYS_PROTECT 保护,写入前需要先写入 0x7A83,使能寄存器写操作 写入其它值,禁止寄存器写操作



## 6 FLASH

### 6.1 概述

FLASH 存储体包含两个部分：NVR 和 MAIN。NVR 大小为 1KB，MAIN 为 32KB。

主闪存存储区（MAIN），包括应用程序和用户数据区

信息存储区（info 区/NVR），其包含三个部分：

- 选项字节(Option bytes) – 内含硬件校准值及产品 ID（不占下图 NVR 空间）
- 系统内存(System memory) – 其包含 boot loader 代码（不占下图 NVR 空间）
- 用户数据区 – 预留给用户使用，大小为 1KB

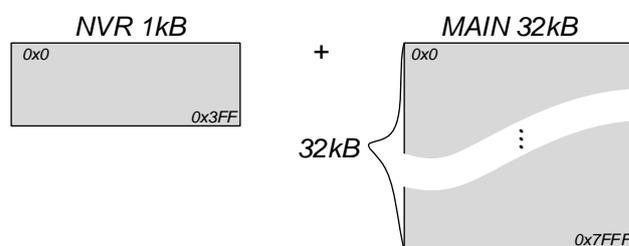


图 6-1 FLASH 存储体空间划分框图

- 可反复擦除写入不低于 2 万次
- 室温数据保持长达 100 年
- 单字节编程时间最长 7.5us，Sector 擦除时间最长 5ms。
- Sector 大小 512 字节，可按 Sector 擦除写入，支持运行时编程，擦写一个 Sector 的同时读取访问另一个 Sector
- Flash 数据防窃取（最后一个 word 须写入非 0xFFFFFFFF 的任意值）

### 6.2 功能特点

FLASH 控制器模块，主要实现对 FLASH 存储体的相关操作。包括：

- FLASH 读取数据的操作，包括对 NVR 部分的读取和对 MAIN 部分的读取。
- FLASH 写入数据的操作，包括对 NVR 部分的写入和对 MAIN 部分的写入。
- FLASH 擦除操作，包括 CHIP 擦除和 SECTOR 擦除。NVR 部分仅支持 SECTOR 擦除，MAIN 部



分支支持 CHIP 擦除和 SECTOR 擦除。

- FLASH 深度休眠的操作，以降低芯片的休眠功耗。
- FLASH 存储体内容的加密操作。
- FLASH 的读取加速操作，以提升芯片整体运行效率。
- FLASH 控制寄存器的访问。

### 6.2.1 功能描述

控制模块，实现了对 FLASH 存储体的复位/读出/写入/擦除/休眠等操作。如下为控制状态转换图：

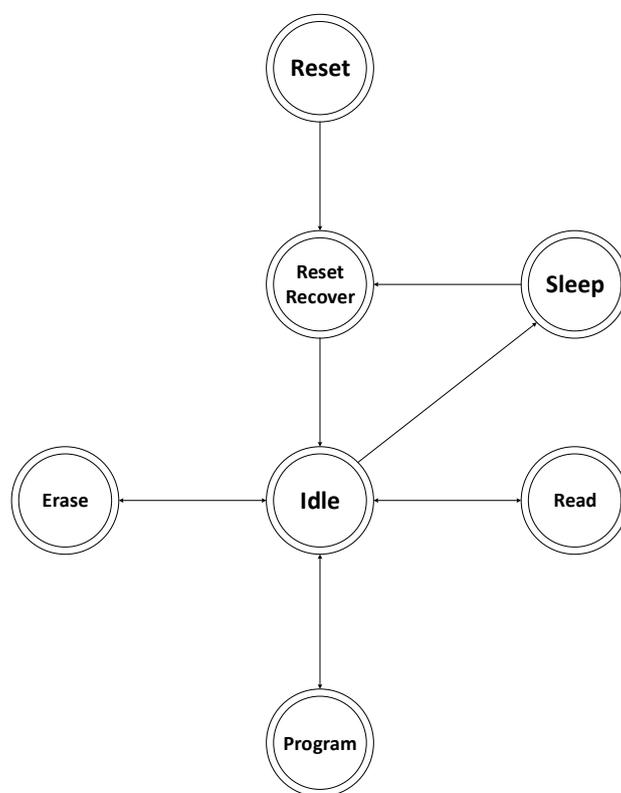


图 6-2 FLASH 控制状态转换图

#### 6.2.1.1 复位操作

系统完成复位后，FLASH 需要一段时间恢复。其目的是保证 FLASH 存储体内部的电路稳定。待内部稳定后，才可对 FLASH 执行操作。此恢复操作由硬件自动实现，无需软件干预。

#### 6.2.1.2 休眠操作

FLASH 的休眠操作分成两个部分：**Standby** 和 **Deep Sleep**。当系统不执行对 FLASH 的操作时，FLASH 可自动进入 **StandBy** 状态（若开启预取，此功能失效）。当系统执行 **Deep Sleep** 操作时，将触发 FLASH 也进入 **Deep Sleep**，实现降低功耗的目的。FLASH 进入 **Deep Sleep** 的操作，硬件自动完成，无需软件介入。



当外界唤醒系统，同时将唤醒 FLASH。经过一段时间恢复后，FLASH 可执行正常操作。此唤醒恢复操作，硬件自动完成，无需软件介入。

### 6.2.1.3 FLASH 读取操作

读操作为 FLASH 的基本操作。系统可通过两条路径访问 FLASH 内部的数据。

- MCU 通过 AHB 总线，直接对 FLASH 执行取指取数操作。取指宽度为 32bit，且只能访问 MAIN 空间的数据。为了加快 MCU 的取指取数数据的速度，硬件提供了加速的功能。
- MCU 通过 AHB 总线，访问控制器的寄存器，间接实现读取 FLASH 内部数据的操作。可以访问 MAIN 和 NVR 空间的数据；若执行连续读取操作，硬件可自动完成地址累加，无需每次都更新地址寄存器的值。

FLASH\_CFG.REGION 位指示当前访问的是哪个空间。具体表格如下：

表 6-1 FLASH 访问空间分配表

NVR (FLASH_CFG.REGION)	访问区域
0	MAIN 区域
1	NVR 区域

访问本控制器的寄存器，实现间接读取 FLASH 内部数据的操作的执行流程如下：

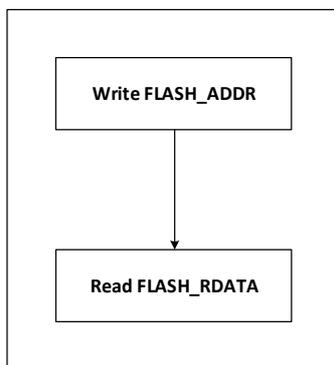


图 6-3 FLASH 间接读取操作流程图

### 6.2.1.4 FLASH 编程操作

执行对 FLASH 存储体的编程操作。一般而言，我们先执行擦除操作，然后才能执行数据编程操作。同时，只能通过访问 FLASH 控制器的寄存器，实现编程操作。具体流程为：

- 控制寄存器 CFG，开启编程使能
- 地址寄存器 ADDR，写入编程地址
- 写数据寄存器 WDATA，写入编程数据

访问本控制器的寄存器，实现 FLASH 编程操作的执行流程如下：

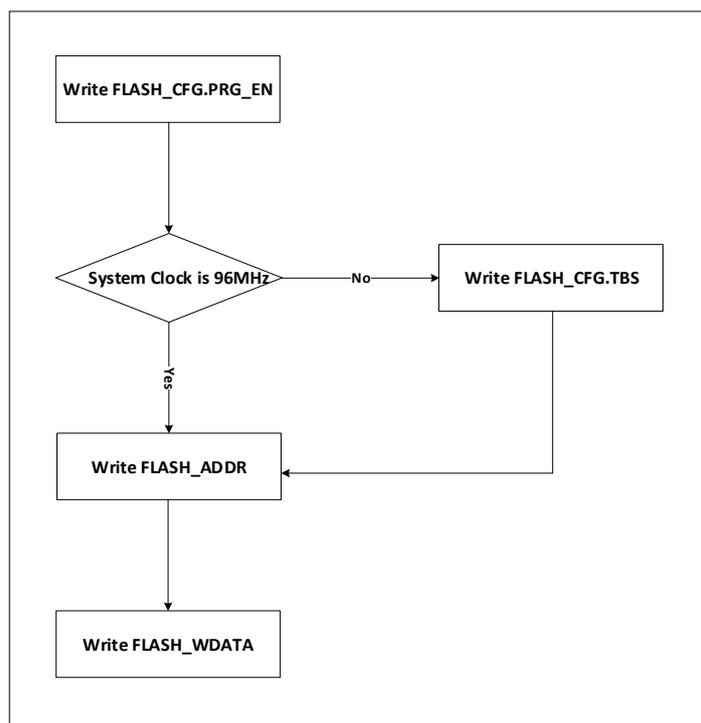


图 6-4 FLASH 模块编程操作流程

系统工作频率的判断，需要参考 `SYS_CLK_CFG` 的配置。`FLASH` 写入/擦除操作的绝对时间是固定的，`FLASH` 控制器需要保存这些绝对时间对应的计数值。`FLASH_CFG.TBS` 默认值是 `96MHz` 时钟频率下的计数值；当芯片工作在其他频率下时，需要配置 `FLASH_CFG.TBS` 的值，以实现 `48MHz/24MHz` 和 `12MHz` 的计数值（其它频率暂不支持）。最终保证计数值的值 $\times$ 时钟频率等于恒定的时间。不同频率下对应的的 `FLASH_CFG.TBS` 值请参考 6.3.2。切记，`FLASH_CFG.TBS` 仅能配置寄存器说明中提供的几组值，不能被写入其它值，否则可能导致 `FLASH` 编程/擦除失败。建议对 `FLASH_CFG` 的操作，执行先读回，然后按照或/与的方法操作。另外，在执行 `FLASH` 的编程/擦除操作时，`MCU` 将暂停工作直至 `FLASH` 的编程/擦除操作完毕。

图 6-4 仅展示了一次编程的流程。若执行连续编程时，可以在写入 `FLASH_ADDR` 寄存器前，配置 `FLASH_CFG.ADR_INC`，开启地址自动递增模式，后续只需要反复写 `FLASH_WDATA` 寄存器即可，`FLASH_ADDR` 每次写入一次数据会自动增加 `0x4`。连续读操作类似。连续编程的流程如下：

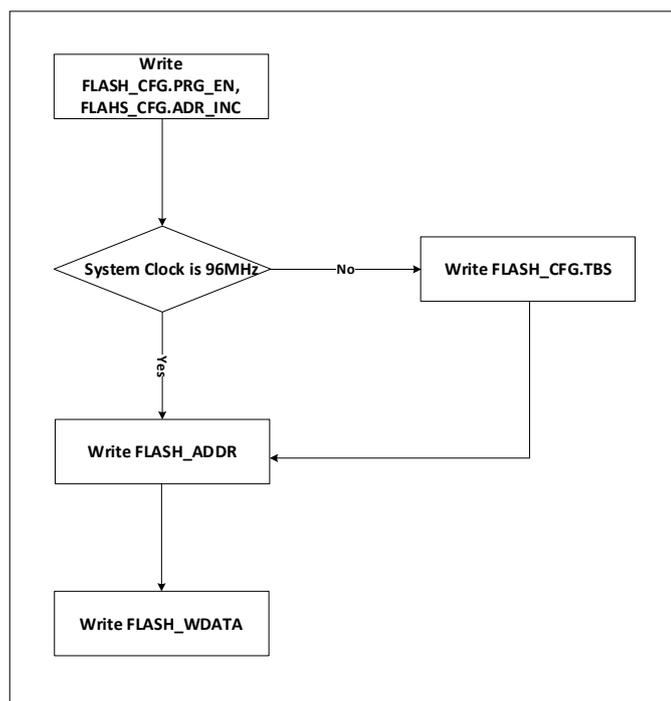


图 6-5 FLASH 模块编程操作流程图

### 6.2.1.5 FLASH 擦除操作

擦除操作为 FLASH 的基本操作。系统只能通过访问 FLASH 控制器的寄存器实现。具体流程为：

- FLASH 擦除操作使能
- 地址寄存器 ADDR，写入擦除地址
- 写擦除寄存器 ERASE，触发擦除操作

执行对 FLASH 存储体的擦除操作。擦除分成 Sector 和 FullChip。分别对应，512Byte 的擦除和 32kB 的擦除。通过配置 FLASH 控制寄存器决定执行哪一种类型的擦除操作。

下表为 Block 和 Sector 地址分配空间。

表 6-2 FLASH Sector 地址分配表

Name	Addresses	Size(Bytes)
Sector 0	0x0000 0000 - 0x0000 01FF	512
Sector1	0x0000 0200 - 0x0000 03FF	512
Sector2	0x0000 0400 - 0x0000 05FF	512
...	...	...
Sector63	0x0000 7E00 - 0x0000 7FFF	512

NVR 区域只能实现 Sector 擦除；MAIN 区域可以实现 Sector 擦除和 FULL 擦除。具体表格如下：

NVR (CFG B[11])	Sector Erase	FULL Erase
0	Main 区域	Main 区域



1	NVR 区域	Main 区域
---	--------	---------

FLASH 擦除操作流程如下所示。

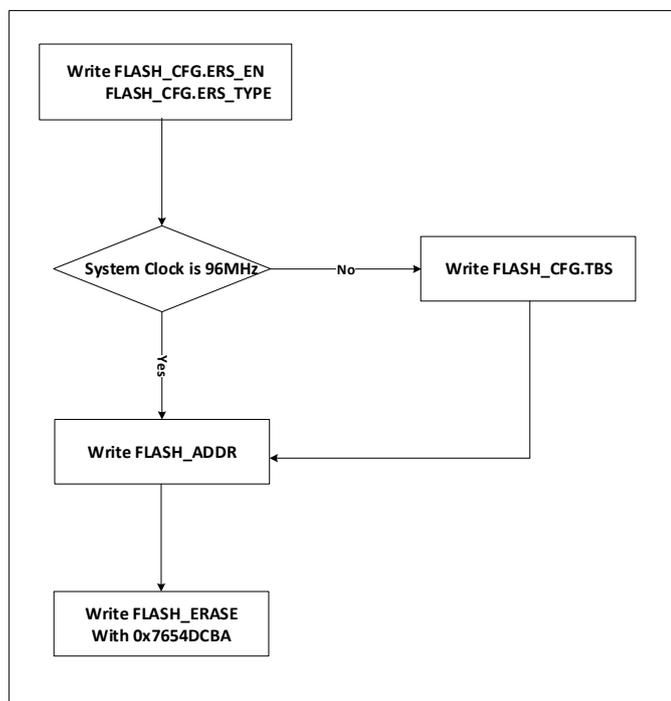


图 6-6 FLASH 模块擦除操作流程

若选择 Sector 擦除，需要通过 FLASH\_ADDR 确定哪个 Sector 被擦除，若是 Full Chip 模式的话，FLASH\_ADDR 的值将失效。FLASH\_ERASE 写入 0x7654DCBA 触发擦除操作。

#### 6.2.1.6 FLASH 预取操作

因 FLASH 存储体的速度限制，无法达到 96MHz 的速度。当对 FLASH 进行读取操作时，需要大于 1 个时钟周期才能完成数据的读出。为了加快数据的读出，FLASH 控制器增加了预取功能。当 FLASH 控制器执行完当前读取操作后，在不影响正常程序执行的前提下，顺序预取下一个 WORD 的数据。预取操作的开启和关闭，只需要设置 FLASH\_CFG.PREF 即可。

#### 6.2.1.7 FLASH 加密保护

若 FLASH 存储体内的数据处于加密状态，用户可执行解密操作，可对 FLASH 存储体内的数据进行解密。相反，若 FLASH 存储体内的数据处于解密状态，用户可执行加密操作，对 FLASH 存储体内的数据进行加密。默认情况下，FLASH 存储体内的数据处于加密状态。芯片上电复位后，硬件自动执行一次加密状态更新操作，是加密状态的话仍然加密状态，是解密的话状态变成解密状态。

FLASH 存储体共有 32KB，最后一个 WORD 设计为加密字。当这个 WORD 内容为全 1 时，表明此时 FLASH 处于解密状态；当这个 WORD 的内容被写为非全 1 时，表明此时 FLASH 处于加密状态。若需要加密，执行最后一个 WORD 的编程，写入非全 1 的值，读取 FLASH\_PROTECT 寄存器，即触发一次加密状态更新，完成加密（读取 FLASH\_PROTECT 返回值无意义）。

对应的解密流程，分成两种情况。若最后一个 WORD 没有执行过编程写入非全 1 的操作，读取



FLASH\_PROTECT 寄存器，即完成解密更新（无需考虑此时返回值）。若已经执行过编程写入非全 1 的操作，那么只能执行擦除操作才能解除。先对 FLASH 执行擦除操作，将最后一个 WORD 恢复为全 1 值，然后读取 FLASH\_PROTECT 寄存器，即触发一次加密状态更新，完成解密（读取 FLASH\_PROTECT 返回值无意义）。

### 6.2.1.8 FLASH 在线升级(IAP)

IAP 模式，实现中断向量表的重映射。在 LKS32MC05x 系列芯片中，包含了系统寄存器 VTOR，其地址为 0xE000\_ED08。用于重新映射中断向量表入口地址。

表 6-3 IAP VTOR 寄存器描述

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

位置	位名称	说明
[31:7]	VTOR	执行写入操作，写入中断向量表入口地址

默认值为 0x0，此时中断向量表入口地址为 0x0。当写入非 0 值时，中断向量表入口地址将映射到写入值对应的地址上，立即生效。

在 LKS32MC05x 系列芯片中，因为有 VTOR 寄存器。用户可根据自己需求，更新整个 FLASH 的内容。在线升级过程中可以使用中断，也可以关闭中断。

#### 6.2.1.8.1 开启中断的在线升级

推荐软件配置流程：

关闭 MCU 的中断控制器，暂时不接收新的中断响应；

在新的中断入口地址处，放置中断处理函数代码；

将新的中断入口地址写入 VTOR 寄存器；

开启 MCU 的中断控制器，使能中断；

用户跳转至在线升级函数，开始在线升级功能；

完成升级，关闭 MCU 的中断控制器，配置 VTOR 为默认值 0；

执行 MCU 软复位，系统 PC 重新从 0 地址开始执行升级后的程序；



## 6.2.1.8.2 关闭中断的在线升级

关闭 MCU 的中断控制器，暂时不接收新的中断响应；

用户跳转至在线升级函数，开始在线升级功能；如果在线升级使用了类似 UART 的外设通讯，需要 MCU 轮询处理 UART 的中断标志位。

执行 MCU 软复位，系统 PC 重新从 0 地址开始执行升级后的程序；

## 6.2.1.8.3 在线升级函数的位置

如果需要将 FLASH 全部擦除，则需要将在线升级函数放置在 RAM 中，如果需要使用中断则新的中断向量入口地址也需要位于 RAM 地址空间。

如果只需要擦除应用程序占用的部分 FLASH 区域，则可以将在线升级函数放置在 FLASH 高段地址的空闲区域，使用块擦除 FLASH 旧的应用程序，写入新的应用程序。

## 6.3 寄存器

## 6.3.1 地址分配

FLASH 控制器模块寄存器的基地址是 0x40000400。

表 6-4 FLASH 控制器模块寄存器列表

名称	偏移	说明
FLASH_CFG	0x00	FLASH 配置寄存器
FLASH_ADDR	0x04	地址寄存器
FLASH_WDATA	0x08	写数据寄存器
FLASH_RDATA	0x0C	读数据寄存器
FLASH_ERASE	0x10	擦除使能寄存器
FLASH_PROTECT	0x14	FLASH 保护状态寄存器
FLASH_READY	0x18	FLASH 闲忙状态寄存器

## 6.3.2 FLASH\_CFG 配置寄存器（推荐先读回，按或/与方式修改）

地址：0x4000\_0400

复位值：0x60

表 6-5 FLASH\_CFG 配置寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERS_EN				PRG_EN				ADR_INC				PREF			
RW				RW				RW				RW			
0				0				0				0			



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERS_TYPE				REGION				TBS							
RW				RW				RW							
0				0				60							

位置	位名称	说明
[31]	ERS_EN	FLASH 擦除使能。默认为 0。 0: 关闭擦除 1: 开启擦除
[27]	PRG_EN	FLASH 编程使能。默认为 0。 0: 关闭编程 1: 开启编程
[23]	ADR_INC	FLASH 地址递增使能。默认为 0。 0: 关闭递增使能 1: 开启递增使能 当执行 FLASH 连续读写访问时, 可以开启此功能减少对地址的操作。
[19]	PREF	FLASH 预取加速使能。默认为 0。 0: 关闭加速 1: 开启加速
[15]	ERS_TYPE	FLASH 擦除类型选择。默认为 0。 0: Sector 1: FULL
[11]	REGION	访问 FLASH 区域选择。默认为 0。 0: MAIN 1: NVR
[6:0]	TBS	编程/擦除时间基数寄存器, 默认值为 0x60。只能配成如下几个值 0x60: 96Mhz 系统频率下, FLASH 编程/擦除时间基数配置值。 0x2F: 48Mhz 系统频率下, FLASH 编程/擦除时间基数配置值。 0x17: 24Mhz 系统频率下, FLASH 编程/擦除时间基数配置值。 0x0B: 12Mhz 系统频率下, FLASH 编程/擦除时间基数配置值。

### 6.3.3 FLASH\_ADDR 地址寄存器

地址: 0x4000\_0404

复位值: 0x0

表 6-6 FLASH\_ADDR 地址寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
RW															
0															



位置	位名称	说明
[31:16]		未使用
[15:0]	ADDR	地址寄存器。读/写/擦除操作对应的地址寄存器。因按照 WORD 操作，最低两位会被 FLASH 控制器忽略。 执行擦除操作时，需要根据擦除类型，地址需要对齐。一个 Sector 是 512-Byte。若执行 Sector 擦除，地址需要是 512 的整数倍（若带偏移，偏移量会被忽略）。全芯片擦除，不会参考这个寄存器的值。

### 6.3.4 FLASH\_WDATA 写数据寄存器

地址：0x4000\_0408

复位值：0x0

表 6-7 FLASH\_WDATA 写数据寄存器

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

位置	位名称	说明
[31:0]	WDATA	执行写入操作，写入 FLASH 的值

### 6.3.5 FLASH\_RDATA 读数据寄存器

地址：0x4000\_040C

复位值：0x0

表 6-8 FLASH\_RDATA 读数据寄存器

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



位置	位名称	说明
[31:0]	RDATA	执行读取操作，读出 FLASH 的值

### 6.3.6 FLASH\_ERASE 擦除控制寄存器

地址：0x4000\_0410

复位值：0x0

表 6-9 FLASH\_ERASE 擦除控制寄存器

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

位置	位名称	说明
[31:0]	ERASE	写入 0x7654DCBA，触发擦除操作

### 6.3.7 FLASH\_PROTECT 加密状态寄存器

地址：0x4000\_0414

复位值：0x0

表 6-10 FLASH\_PROTECT 加密状态寄存器

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

位置	位名称	说明
[31:0]	PROTECT	读取该寄存器，更新加密/解密状态读取返回值无参考意义



### 6.3.8 FLASH\_READY 工作状态寄存器

地址：0x4000\_0418

复位值：0x0

表 6-11 FLASH\_READY 工作状态寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															READY
															RO
															0

位置	位名称	说明
[31:1]		未使用
[0]	READY	1: FLASH 处于闲状态 0: FLASH 处于忙状态

### 6.3.9 NVR 校准参数地址信息

LKS32MC5x NVR 区域使用说明：

LKS32MC05x 产品 NVR 区域特性：

NVR 区域，大小为 1KB；一个 sector 大小为 512B，因此，NVR 区域共计 2 个 sector；

NVR 区域，仅用于数据存储，仅能通过 FLASH 寄存器访问；

NVR 区域支持的操作包括，读取、编程和擦除；

NVR 区域仅支持 sector 擦除，在全芯片擦除情况下，NVR 区域的数据不会被擦除掉；

NVR 区域，推荐关闭系统中断后，执行访问操作；尤其是执行编程和擦除操作；

LKS 公司提供库函数，包含上述操作：

lks32MC05x\_Flash.c 文件包含了编程和擦除操作函数

擦除函数：void EraseSector(u32 adr, u16 nvr);

编程函数：int ProgramPage(u32 adr, u32 sz, u8 \*buf, u16 nvr);

lks32mc05x\_nvr.o 文件包含了读取函数：

读取函数：uint32\_t Read\_NVR(uint32\_t adr);



LKS32MC05x 产品存储芯片校准参数：

校准参数，每颗产品均独立校准，每颗产品不支持校准参数混用；

校准参数，在出厂前进行写入，出厂后不支持编程和擦除，仅支持读取；

校准参数，通过 LKS 公司提供的库函数进行读取访问；

校准参数，推荐关闭系统中断后，执行访问操作；

lks32mc05x\_nvr.o 文件包含了校准参数的读取函数：

读取函数：uint32\_t Read\_Trim(uint32\_t adr);

表 6-12 校准参数列表

地址	内容
0x00000310	ADC0_DC0 校准值
0x00000314	ADC0_DC1 校准值
0x00000318	ADC0_AMC0 校准值
0x0000031C	ADC0_AMC1 校准值
0x00000330	DAC 选择 1.20V 档位, SYS_AFE_DAC_AMC 校准值 (扩大 512 倍结果)
0x00000334	DAC 选择 4.80V 档位, SYS_AFE_DAC_AMC 校准值 (扩大 512 倍结果)
0x00000340	DAC 选择 1.20V 档位, SYS_AFE_DAC_DC 校准值
0x00000344	DAC 选择 4.80V 档位, SYS_AFE_DAC_DC 校准值
0x00000348	DAC 选择 1.20V 档位, DAC 二次校准值
0x0000034C	DAC 选择 4.80V 档位, DAC 二次校准值
0x00000350	OPA0 , 200K 欧姆 VS 10.6K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆
0x00000354	OPA0 , 190K 欧姆 VS 20.6K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆
0x00000358	OPA0 , 180K 欧姆 VS 30.6K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆
0x0000035C	OPA0 , 170K 欧姆 VS 40.6K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆
0x00000360	OPA1 , 200K 欧姆 VS 10.6K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆
0x00000364	OPA1 , 190K 欧姆 VS 20.6K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆
0x00000368	OPA1 , 180K 欧姆 VS 30.6K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆
0x0000036C	OPA1 , 170K 欧姆 VS 40.6K 欧姆, GAIN 校准值 (扩大 1000 倍结果), R0 为 0 欧姆
0x000002D0	OPA0, 200K 欧姆 VS 10.6K 欧姆, 高 16-bit 存放 R2 实际电阻值(注意, 放大 100 倍存入) 低 16-bit 存放 R1 电阻值(注意, 放大 100 倍存入)
0x000002D4	OPA0, 190K 欧姆 VS 20.6K 欧姆, 高 16-bit 存放 R2 实际电阻值(注意, 放大 100 倍存入) 低 16-bit 存放 R1 电阻值(注意, 放大 100 倍存入)
0x000002D8	OPA0, 180K 欧姆 VS 30.6K 欧姆, 高 16-bit 存放 R2 实际电阻值(注意, 放大 100 倍存入) 低 16-bit 存放 R1 电阻值(注意, 放大 100 倍存入)
0x000002DC	OPA0, 170K 欧姆 VS 40.6K 欧姆, 高 16-bit 存放 R2 实际电阻值(注意, 放大 100 倍存入) 低 16-bit 存放 R1 电阻值(注意, 放大 100 倍存入)



0x000002E0	OPA1, 200K 欧姆 VS 10.6K 欧姆, 高 16-bit 存放 R2 实际电阻值(注意, 放大 100 倍存入) 低 16-bit 存放 R1 电阻值(注意, 放大 100 倍存入)
0x000002E4	OPA1, 190K 欧姆 VS 20.6K 欧姆, 高 16-bit 存放 R2 实际电阻值(注意, 放大 100 倍存入) 低 16-bit 存放 R1 电阻值(注意, 放大 100 倍存入)
0x000002E8	OPA1, 180K 欧姆 VS 30.6K 欧姆, 高 16-bit 存放 R2 实际电阻值(注意, 放大 100 倍存入) 低 16-bit 存放 R1 电阻值(注意, 放大 100 倍存入)
0x000002EC	OPA1, 170K 欧姆 VS 40.6K 欧姆, 高 16-bit 存放 R2 实际电阻值(注意, 放大 100 倍存入) 低 16-bit 存放 R1 电阻值(注意, 放大 100 倍存入)
0x000002F0	OPA0 四个挡位下 Offset 值(有符号类型, 单位 V, 放大 10000 倍) Bit[31:16] 4 倍放大倍数 3 档 Bit[15:0] 8 倍放大倍数 2 档
0x000002F4	OPA0 四个挡位下 Offset 值(有符号类型, 单位 V, 放大 10000 倍) Bit[31:16] 16 倍放大倍数 1 档 Bit[15:0] 32 倍放大倍数 0 档
0x000002F8	OPA1 四个挡位下 Offset 值(有符号类型, 单位 V, 放大 10000 倍) Bit[31:16] 4 倍放大倍数 3 档 Bit[15:0] 8 倍放大倍数 2 档
0x000002FC	OPA1 四个挡位下 Offset 值(有符号类型, 单位 V, 放大 10000 倍) Bit[31:16] 16 倍放大倍数 1 档 Bit[15:0] 32 倍放大倍数 0 档
0x00000398	温度传感器, 斜率校准值
0x0000039C	温度传感器, 偏置校准值
0x000003B0	高 16-bit 存放 OPA1 共模电压, 低 16-bit 存放 OPA0 共模电压 (扩大 10000 倍结果)
0x000003B4	高 16-bit 存放 OPA3 共模电压, 低 16-bit 存放 OPA2 共模电压 (扩大 10000 倍结果)

## 7 GPIO

### 7.1 概述

LKS32MC05x 系列芯片共集成了 3 组 16bit 位宽 GPIO。P0.0/P0.1/P1.0/P1.1 4 个 GPIO 可以作为系统的唤醒源。P0.0 ~ P0.15 共 16 个 GPIO 可以用作外部中断源输入。外部中断和唤醒功能使用的是 IO 的 GPIO 功能，相应 IO 的第二功能可以配置为 0。

#### 7.1.1 功能框图

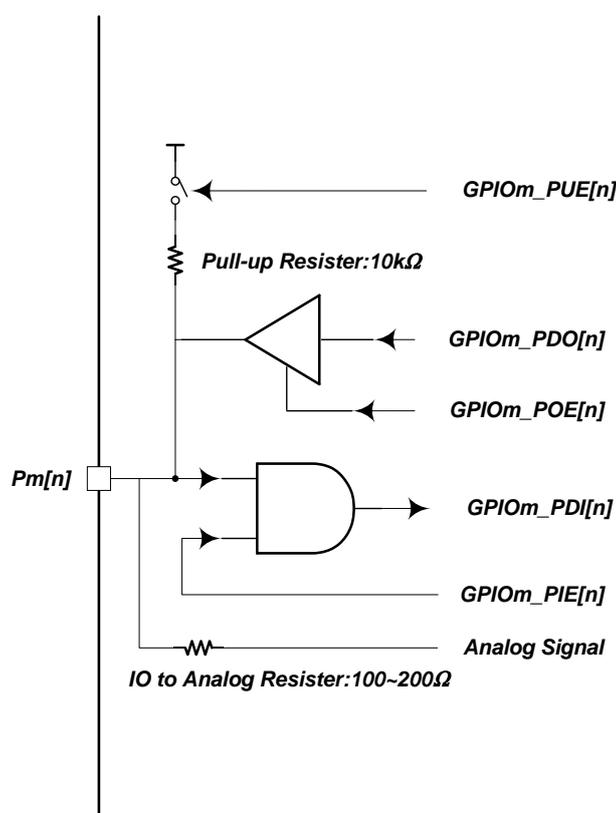


图 7-1 GPIO 功能框图

如上图所示,  $Pm[n]$  为芯片 PAD,  $m$  可以是 0~2, 表示 3 组 GPIO 中的任意一组,  $n$  可以是 0~15, 表示一组 16bit GPIO 中的一个 IO。模拟信号通过一个电阻串联直接连接到 PAD, 电阻阻值为 100~200Ω。数字信号经过一个三态门输出, 当输出使能  $GPIOm\_POE[n]=0$  时, buffer 输出高阻态, 否则 buffer 输出与  $GPIOm\_PDO[n]$  电平相同。数字信号输入通过一个与门进入芯片内部, 当  $GPIOm\_PIE[n]=0$  时,  $GPIOm\_PDI[n]$  恒为 0, 当  $GPIOm\_PIE[n]=1$ , 即输入使能打开时,  $GPIOm\_PDI[n]$  的电平与  $Pm[n]$  电平相同。芯片 PAD 可以配置上拉, P0[2] 引脚因为复用为外部复位脚 RSTN 上拉电阻为 100kΩ, 其余上拉电阻为 10kΩ, 注意, 并非所有 PAD 都配备上拉电阻, 具体哪些 PAD 具有上拉电阻资源, 请参考 7.3.1 章节。没有上拉电阻的 PAD 也可以配置  $GPIOm\_PUE[n]$  寄存器, 但无实际作用。

## 7.1.2 产品特点

- 3 组 16bit GPIO
- 支持开漏
- 部分 IO 支持内部上拉
- 支持配置锁定保护
- 支持外部中断
- 支持 GPIO 唤醒

## 7.2 寄存器

### 7.2.1 地址分配

GPIO 0 模块在芯片中的基地址是 0x40012000。

GPIO 1 模块在芯片中的基地址是 0x40012040。

GPIO 2 模块在芯片中的基地址是 0x40012080。

GPIO 0/1/2 的寄存器定义完全相同，仅基地址不同。

表 7-1 GPIOx 寄存器列表

名称	偏移地址	说明
GPIOx_PIE	0x00	GPIO x 输入使能
GPIOx_POE	0x04	GPIO x 输出使能
GPIOx_PDI	0x08	GPIO x 输入数据
GPIOx_PDO	0x0C	GPIO x 输出数据
GPIOx_PUE	0x10	GPIO x 上拉使能
GPIOx_PODE	0x18	GPIO x 开漏使能
GPIOx_F3210	0x20	GPIO x [3:0]功能选择
GPIOx_F7654	0x24	GPIO x [7:4]功能选择
GPIOx_FBA98	0x28	GPIO x [11:8]功能选择
GPIOx_FFEDC	0x2C	GPIO x [15:12]功能选择
GPIOx_BSRR	0x30	GPIO x 位操作寄存器
GPIOx_BRR	0x34	GPIO x 位清零寄存器

GPIO 中断/唤醒模块的基地址是 0x40012100。

表 7-2 GPIO 中断/唤醒模块寄存器列表

名称	偏移地址	说明
EXTI_CR0	0x00	GPIO 0[7:0] 中断触发类型
EXTI_CR1	0x04	GPIO 0[15:8]中断触发类型



EXTI_IF	0x08	GPIO 中断标志
WAKE_POL	0x10	GPIO 唤醒信号极性
WAKE_EN	0x14	GPIO 唤醒使能

### 7.2.2 GPIOx\_PIE

地址分别是：0x4001\_2000，0x4001\_2040，0x4001\_2080。

复位值：0x0

表 7-3 GPIOx\_PIE GPIOx 输入使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIE15	PIE14	PIE13	PIE12	PIE11	PIE10	PIE9	PIE8	PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	PIE15	GPIO x[15] / Px[15] 输入使能
[14]	PIE14	GPIO x[14] / Px[14] 输入使能
[13]	PIE13	GPIO x[13] / Px[13] 输入使能
[12]	PIE12	GPIO x[12] / Px[12] 输入使能
[11]	PIE11	GPIO x[11] / Px[11] 输入使能
[10]	PIE10	GPIO x[10] / Px[10] 输入使能
[9]	PIE9	GPIO x[9] / Px[9] 输入使能
[8]	PIE8	GPIO x[8] / Px[8] 输入使能
[7]	PIE7	GPIO x[7] / Px[7] 输入使能
[6]	PIE6	GPIO x[6] / Px[6] 输入使能
[5]	PIE5	GPIO x[5] / Px[5] 输入使能
[4]	PIE4	GPIO x[4] / Px[4] 输入使能
[3]	PIE3	GPIO x[3] / Px[3] 输入使能
[2]	PIE2	GPIO x[2] / Px[2] 输入使能
[1]	PIE1	GPIO x[1] / Px[1] 输入使能
[0]	PIE0	GPIO x[0] / Px[0] 输入使能

### 7.2.3 GPIOx\_POE

地址分别是：0x4001\_2004，0x4001\_2044，0x4001\_2084

复位值：0x0

表 7-4 GPIOx\_POE GPIOx 输出使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---



POE15	POE14	POE13	POE12	POE11	POE10	POE9	POE8	POE7	POE6	POE5	POE4	POE3	POE2	POE1	POE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	POE15	GPIO x[15] / Px[15] 输出使能
[14]	POE14	GPIO x[14] / Px[14] 输出使能
[13]	POE13	GPIO x[13] / Px[13] 输出使能
[12]	POE12	GPIO x[12] / Px[12] 输出使能
[11]	POE11	GPIO x[11] / Px[11] 输出使能
[10]	POE10	GPIO x[10] / Px[10] 输出使能
[9]	POE9	GPIO x[9] / Px[9] 输出使能
[8]	POE8	GPIO x[8] / Px[8] 输出使能
[7]	POE7	GPIO x[7] / Px[7] 输出使能
[6]	POE6	GPIO x[6] / Px[6] 输出使能
[5]	POE5	GPIO x[5] / Px[5] 输出使能
[4]	POE4	GPIO x[4] / Px[4] 输出使能
[3]	POE3	GPIO x[3] / Px[3] 输出使能
[2]	POE2	GPIO x[2] / Px[2] 输出使能
[1]	POE1	GPIO x[1] / Px[1] 输出使能
[0]	POE0	GPIO x[0] / Px[0] 输出使能

#### 7.2.4 GPIOx\_PDI

地址分别是：0x4001\_2008，0x4001\_2048，0x4001\_2088

复位值：0x0

表 7-5 GPIOx\_PDI GPIOx 输入数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDI															
RO															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	PDI	GPIO x 输入数据



### 7.2.5 GPIOx\_PDO

地址分别是：0x4001\_200C, 0x4001\_204C, 0x4001\_208C

复位值：0x0

表 7-6 GPIOx\_PDO GPIOx 输出数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDO															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	PDO	GPIO x 输出数据

### 7.2.6 GPIOx\_PUE

地址分别是：0x4001\_2010, 0x4001\_2050, 0x4001\_2090

复位值：0x0

表 7-7 GPIOx\_PUE GPIOx 上拉使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUE15	PUE14	PUE13	PUE12	PUE11	PUE10	PUE9	PUE8	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	PUE15	GPIO x[15] / Px[15] 上拉使能
[14]	PUE14	GPIO x[14] / Px[14] 上拉使能
[13]	PUE13	GPIO x[13] / Px[13] 上拉使能
[12]	PUE12	GPIO x[12] / Px[12] 上拉使能
[11]	PUE11	GPIO x[11] / Px[11] 上拉使能
[10]	PUE10	GPIO x[10] / Px[10] 上拉使能
[9]	PUE9	GPIO x[9] / Px[9] 上拉使能
[8]	PUE8	GPIO x[8] / Px[8] 上拉使能
[7]	PUE7	GPIO x[7] / Px[7] 上拉使能
[6]	PUE6	GPIO x[6] / Px[6] 上拉使能
[5]	PUE5	GPIO x[5] / Px[5] 上拉使能
[4]	PUE4	GPIO x[4] / Px[4] 上拉使能



[3]	PUE3	GPIO x[3] / Px[3] 上拉使能
[2]	PUE2	GPIO x[2] / Px[2] 上拉使能
[1]	PUE1	GPIO x[1] / Px[1] 上拉使能
[0]	PUE0	GPIO x[0] / Px[0] 上拉使能

注意：并非所有 IO 都具有上拉功能，具体哪些 IO 具有上拉功能请参考 7.3.1。没有上拉功能的 IO 对应的 PUE 寄存器也没有实现，因此向这些位置写入 1 无效，读回恒为 0。

## 7.2.7 GPIOx\_PODE

地址分别是：0x4001\_2018, 0x4001\_2058, 0x4001\_2098

复位值：0x0

表 7-8 GPIOx\_PODE GPIOx 开漏使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PODE15	PODE14	PODE13	PODE12	PODE11	PODE10	PODE9	PODE8	PODE7	PODE6	PODE5	PODE4	PODE3	PODE2	PODE1	PODE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	PODE15	GPIO x[15] / Px[15] 开漏使能
[14]	PODE14	GPIO x[14] / Px[14] 开漏使能
[13]	PODE13	GPIO x[13] / Px[13] 开漏使能
[12]	PODE12	GPIO x[12] / Px[12] 开漏使能
[11]	PODE11	GPIO x[11] / Px[11] 开漏使能
[10]	PODE10	GPIO x[10] / Px[10] 开漏使能
[9]	PODE9	GPIO x[9] / Px[9] 开漏使能
[8]	PODE8	GPIO x[8] / Px[8] 开漏使能
[7]	PODE7	GPIO x[7] / Px[7] 开漏使能
[6]	PODE6	GPIO x[6] / Px[6] 开漏使能
[5]	PODE5	GPIO x[5] / Px[5] 开漏使能
[4]	PODE4	GPIO x[4] / Px[4] 开漏使能
[3]	PODE3	GPIO x[3] / Px[3] 开漏使能
[2]	PODE2	GPIO x[2] / Px[2] 开漏使能
[1]	PODE1	GPIO x[1] / Px[1] 开漏使能
[0]	PODE0	GPIO x[0] / Px[0] 开漏使能

## 7.2.8 GPIOx\_PFLT

地址分别是：0x4001\_201C, 0x4001\_205C, 0x4001\_209C



复位值: 0x0

表 7-9 GPIOx\_PFLT GPIOx 滤波寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFLT15	PFLT14	PFLT13	PFLT12	PFLT11	PFLT10	PFLT9	PFLT8	PFLT7	PFLT6	PFLT5	PFLT4	PFLT3	PFLT2	PFLT1	PFLT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	PFLT15	GPIO x[15] / Px[15] 配置滤波
[14]	PFLT14	GPIO x[14] / Px[14] 配置滤波
[13]	PFLT13	GPIO x[13] / Px[13] 配置滤波
[12]	PFLT12	GPIO x[12] / Px[12] 配置滤波
[11]	PFLT11	GPIO x[11] / Px[11] 配置滤波
[10]	PFLT10	GPIO x[10] / Px[10] 配置滤波
[9]	PFLT9	GPIO x[9] / Px[9] 配置滤波
[8]	PFLT8	GPIO x[8] / Px[8] 配置滤波
[7]	PFLT7	GPIO x[7] / Px[7] 配置滤波
[6]	PFLT6	GPIO x[6] / Px[6] 配置滤波
[5]	PFLT 5	GPIO x[5] / Px[5] 配置滤波
[4]	PFLT 4	GPIO x[4] / Px[4] 配置滤波
[3]	PFLT 3	GPIO x[3] / Px[3] 配置滤波
[2]	PFLT 2	GPIO x[2] / Px[2] 配置滤波
[1]	PFLT1	GPIO x[1] / Px[1] 配置滤波
[0]	PFLT0	GPIO x[0] / Px[0] 配置滤波

### 7.2.9 GPIOx\_F3210

地址分别是: 0x4001\_2020, 0x4001\_2060, 0x4001\_20A0

复位值: 0x0

表 7-10 GPIOx\_F3210 GPIOx 功能选择寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F3				F2				F1				F0			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
----	-----	----



[31:16]		未使用
[15:12]	F3	GPIO x[3] / Px[3] 功能选择
[11:8]	F2	GPIO x[2] / Px[2] 功能选择
[7:4]	F1	GPIO x[1] / Px[1] 功能选择
[3:0]	F0	GPIO x[0] / Px[0] 功能选择

GPIO 的功能复用选择如表 7-11 所示

表 7-11 GPIO 功能复用

GPIOx_Fxxxx 配置值	第二功能代号	功能
0x0	AF0	模拟功能
0x1	AF1	SYS_AF, 比较器及时钟等数字信号输出功能
0x2	AF2	HALL
0x3	AF3	MCPWM
0x4	AF4	UART
0x5	AF5	SPI
0x6	AF6	IIC
0x7	AF7	Timer0/Time1
0x8	AF8	Timer2/Timer3
0x9	AF9	ADC trigger debug

### 7.2.10 GPIOx\_F7654

地址分别是：0x4001\_2024，0x4001\_2064，0x4001\_20A4

复位值：0x0

表 7-12 GPIOx\_F7654 GPIOx 功能选择寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F7				F6				F5				F4			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	F7	GPIO x[7] / Px[7] 功能选择
[11:8]	F6	GPIO x[6] / Px[6] 功能选择
[7:4]	F5	GPIO x[5] / Px[5] 功能选择
[3:0]	F4	GPIO x[4] / Px[4] 功能选择



### 7.2.11 GPIOx\_FBA98

地址分别是：0x4001\_2028, 0x4001\_2068, 0x4001\_20A8

复位值：0x0

表 7-13 GPIOx\_FBA98 GPIOx 功能选择寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F11				F10				F9				F8			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	F11	GPIO x[11] / Px[11] 功能选择
[11:8]	F10	GPIO x[10] / Px[10] 功能选择
[7:4]	F9	GPIO x[9] / Px[9] 功能选择
[3:0]	F8	GPIO x[8] / Px[8] 功能选择

### 7.2.12 GPIOx\_FFEDC

地址分别是：0x4001\_202C, 0x4001\_206C, 0x4001\_20AC

复位值：0x0

表 7-14 GPIOx\_FFEDC GPIOx 功能选择寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F15				F14				F13				F12			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	F15	GPIO x[15] / Px[15] 功能选择
[11:8]	F14	GPIO x[14] / Px[14] 功能选择
[7:4]	F13	GPIO x[13] / Px[13] 功能选择
[3:0]	F12	GPIO x[12] / Px[12] 功能选择

GPIO 的功能复用详细列表请见对应产品 DATASHEET 管脚分布章节。

### 7.2.13 GPIOx\_BSRR

地址分别是：0x4001\_2030, 0x4001\_2070, 0x4001\_20B0



复位值: 0x0

表 7-15 GPIOx\_BSRR GPIOx 位操作寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SET15	SET14	SET13	SET12	SET11	SET10	SET9	SET8	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31]	CLR15	写 1 将 GPIO x[15]清零, 写 0 无作用
[30]	CLR14	写 1 将 GPIO x[14]清零, 写 0 无作用
[29]	CLR13	写 1 将 GPIO x[13]清零, 写 0 无作用
[28]	CLR12	写 1 将 GPIO x[12]清零, 写 0 无作用
[27]	CLR11	写 1 将 GPIO x[11]清零, 写 0 无作用
[26]	CLR10	写 1 将 GPIO x[10]清零, 写 0 无作用
[25]	CLR9	写 1 将 GPIO x[9]清零, 写 0 无作用
[24]	CLR8	写 1 将 GPIO x[8]清零, 写 0 无作用
[23]	CLR7	写 1 将 GPIO x[7]清零, 写 0 无作用
[22]	CLR6	写 1 将 GPIO x[6]清零, 写 0 无作用
[21]	CLR5	写 1 将 GPIO x[5]清零, 写 0 无作用
[20]	CLR4	写 1 将 GPIO x[4]清零, 写 0 无作用
[19]	CLR3	写 1 将 GPIO x[3]清零, 写 0 无作用
[18]	CLR2	写 1 将 GPIO x[2]清零, 写 0 无作用
[17]	CLR1	写 1 将 GPIO x[1]清零, 写 0 无作用
[16]	CLR0	写 1 将 GPIO x[0]清零, 写 0 无作用
[15]	SET15	写 1 将 GPIO x[15]置 1, 写 0 无作用
[14]	SET14	写 1 将 GPIO x[14]置 1, 写 0 无作用
[13]	SET13	写 1 将 GPIO x[13]置 1, 写 0 无作用
[12]	SET12	写 1 将 GPIO x[12]置 1, 写 0 无作用
[11]	SET11	写 1 将 GPIO x[11]置 1, 写 0 无作用
[10]	SET10	写 1 将 GPIO x[10]置 1, 写 0 无作用
[9]	SET9	写 1 将 GPIO x[9]置 1, 写 0 无作用
[8]	SET8	写 1 将 GPIO x[8]置 1, 写 0 无作用
[7]	SET7	写 1 将 GPIO x[7]置 1, 写 0 无作用
[6]	SET6	写 1 将 GPIO x[6]置 1, 写 0 无作用
[5]	SET5	写 1 将 GPIO x[5]置 1, 写 0 无作用



[4]	SET4	写 1 将 GPIO x[4]置 1, 写 0 无作用
[3]	SET3	写 1 将 GPIO x[3]置 1, 写 0 无作用
[2]	SET2	写 1 将 GPIO x[2]置 1, 写 0 无作用
[1]	SET1	写 1 将 GPIO x[1]置 1, 写 0 无作用
[0]	SET0	写 1 将 GPIO x[0]置 1, 写 0 无作用

若用 BSRR 的高 16 位与低 16 位同时对 GPIO 同一位置既置 1 又清零, 则该位被清零。

#### 7.2.14 GPIOx\_BRR

地址分别是: 0x4001\_2034, 0x4001\_2074, 0x4001\_20B4

复位值: 0x0

表 7-16 GPIOx\_BRR GPIOx 位清零寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	CLR15	写 1 将 GPIO x[15]清零, 写 0 无作用
[14]	CLR14	写 1 将 GPIO x[14]清零, 写 0 无作用
[13]	CLR13	写 1 将 GPIO x[13]清零, 写 0 无作用
[12]	CLR12	写 1 将 GPIO x[12]清零, 写 0 无作用
[11]	CLR11	写 1 将 GPIO x[11]清零, 写 0 无作用
[10]	CLR10	写 1 将 GPIO x[10]清零, 写 0 无作用
[9]	CLR9	写 1 将 GPIO x[9]清零, 写 0 无作用
[8]	CLR8	写 1 将 GPIO x[8]清零, 写 0 无作用
[7]	CLR7	写 1 将 GPIO x[7]清零, 写 0 无作用
[6]	CLR6	写 1 将 GPIO x[6]清零, 写 0 无作用
[5]	CLR5	写 1 将 GPIO x[5]清零, 写 0 无作用
[4]	CLR4	写 1 将 GPIO x[4]清零, 写 0 无作用
[3]	CLR3	写 1 将 GPIO x[3]清零, 写 0 无作用
[2]	CLR2	写 1 将 GPIO x[2]清零, 写 0 无作用
[1]	CLR1	写 1 将 GPIO x[1]清零, 写 0 无作用
[0]	CLR0	写 1 将 GPIO x[0]清零, 写 0 无作用

#### 7.2.15 外部中断、唤醒

P0.0/P0.1/P1.0/P1.1 4 个 GPIO 可以作为系统的唤醒源。P0.0 ~ P0.15 共 16 个 GPIO 可以用作



外部中断源输入。

### 7.2.15.1 EXTI\_CR0

地址：0x4001\_2100

复位值：0x0

表 7-17 EXTI\_CR0 外部中断配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T7	T6	T5	T4	T3	T2	T1	T0								
RW															
0	0	0	0	0	0	0	0								

位置	位名称	说明
[31:16]		未使用
[15:14]	T7	GPIO 0[7]/ P0[7]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[13:12]	T6	GPIO 0[6]/ P0[6]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[11:10]	T5	GPIO 0[5]/ P0[5]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[9:8]	T4	GPIO 0[4]/ P0[4]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[7:6]	T3	GPIO 0[3]/ P0[3]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[5:4]	T2	GPIO 0[2]/ P0[2]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发



		11: 上升沿、下降沿均触发
[3:2]	T1	GPIO 0[1]/ P0[1]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[1:0]	T0	GPIO 0[0]/ P0[0]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发

## 7.2.15.2 EXTI\_CR1

地址: 0x4001\_2104

复位值: 0x0

表 7-18 EXTI\_CR1 外部中断配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T15	T14	T13	T12	T11	T10	T9	T8								
RW	RW	RW	RW	RW	RW	RW	RW								
0	0	0	0	0	0	0	0								

位置	位名称	说明
[31:16]		未使用
[15:14]	T15	GPIO 0[15]/P0[15]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[13:12]	T14	GPIO 0[14]/P0[14]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[11:10]	T13	GPIO 0[13]/P0[13]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[9:8]	T12	GPIO 0[12]/P0[12]外部中断触发类型选择 00: 不触发 01: 下降沿触发



		10: 上升沿触发 11: 上升沿、下降沿均触发
[7:6]	T11	GPIO 0[11]/P0[11]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[5:4]	T10	GPIO 0[10]/P0[10]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[3:2]	T9	GPIO 0[9]/P0[9]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发
[1:0]	T8	GPIO 0[8]/P0[8]外部中断触发类型选择 00: 不触发 01: 下降沿触发 10: 上升沿触发 11: 上升沿、下降沿均触发

## 7.2.15.3 EXTI\_IF

地址: 0x4001\_2108

复位值: 0x0

表 7-19 EXTI\_IF 外部中断标志寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	IF3	IF2	IF1	IF0
RW1C															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	IF15	GPIO 0[15] / P0[15] 外部中断标志。中断标志高有效, 写 1 清零
[14]	IF14	GPIO 0[14] / P0[14] 外部中断标志。中断标志高有效, 写 1 清零
[13]	IF13	GPIO 0[13] / P0[13] 外部中断标志。中断标志高有效, 写 1 清零
[12]	IF12	GPIO 0[12] / P0[12] 外部中断标志。中断标志高有效, 写 1 清零
[11]	IF11	GPIO 0[11] / P0[11] 外部中断标志。中断标志高有效, 写 1 清零



[10]	IF10	GPIO 0[10] / P0[10] 外部中断标志。中断标志高有效，写 1 清零
[9]	IF9	GPIO 0[9] / P0[9] 外部中断标志。中断标志高有效，写 1 清零
[8]	IF8	GPIO 0[8] / P0[8] 外部中断标志。中断标志高有效，写 1 清零
[7]	IF7	GPIO 0[7] / P0[7] 外部中断标志。中断标志高有效，写 1 清零
[6]	IF6	GPIO 0[6] / P0[6] 外部中断标志。中断标志高有效，写 1 清零
[5]	IF5	GPIO 0[5] / P0[5] 外部中断标志。中断标志高有效，写 1 清零
[4]	IF4	GPIO 0[4] / P0[4] 外部中断标志。中断标志高有效，写 1 清零
[3]	IF3	GPIO 0[3] / P0[3] 外部中断标志。中断标志高有效，写 1 清零
[2]	IF2	GPIO 0[2] / P0[2] 外部中断标志。中断标志高有效，写 1 清零
[1]	IF1	GPIO 0[1] / P0[1] 外部中断标志。中断标志高有效，写 1 清零
[0]	IF0	GPIO 0[0] / P0[0] 外部中断标志。中断标志高有效，写 1 清零

## 7.2.15.4 WAKE\_POL

地址: 0x4001\_2110

复位值: 0x0

表 7-20 WAKE\_POL 外部唤醒源极性配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												GPIO1_1_POL	GPIO1_0_POL	GPIO0_1_POL	GPIO0_0_POL
												RW	RW	RW	RW
												0	0	0	0

位置	位名称	说明
[31:4]		未使用
[3]	GPIO1_1_POL	GPIO 1[1] / P1[1]外部唤醒触发电平选择。1: 高电平; 0: 低电平
[2]	GPIO1_0_POL	GPIO 1[0] / P1[0]外部唤醒触发电平选择。1: 高电平; 0: 低电平
[1]	GPIO0_1_POL	GPIO 0[1] / P0[1]外部唤醒触发电平选择。1: 高电平; 0: 低电平
[0]	GPIO0_0_POL	GPIO 0[0] / P0[0]外部唤醒触发电平选择。1: 高电平; 0: 低电平

## 7.2.15.5 WAKE\_EN

地址: 0x4001\_2114

复位值: 0x0

表 7-21 WAKE\_EN 外部唤醒源使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												GPIO1_1_WKEN	GPIO1_0_WKEN	GPIO0_1_WKEN	GPIO0_0_WKEN
												RW	RW	RW	RW
												0	0	0	0



位置	位名称	说明
[31:4]		未使用
[3]	GPIO1_1_WKEN	GPIO 1[1] / P1[1] 外部唤醒使能。1: 使能; 0: 禁用。
[2]	GPIO1_0_WKEN	GPIO 1[0] / P1[0] 外部唤醒使能。1: 使能; 0: 禁用。
[1]	GPIO0_1_WKEN	GPIO 0[1] / P0[1] 外部唤醒使能。1: 使能; 0: 禁用。
[0]	GPIO0_0_WKEN	GPIO 0[0] / P0[0] 外部唤醒使能。1: 使能; 0: 禁用。

## 7.3 实现说明

### 7.3.1 上拉实现

LKS32MC05x 系列芯片，通过内部模拟电路进行上拉功能实现。所有 GPIO 都有上拉控制寄存器 PUE，但不是所有 GPIO 都有上拉电路。配备上拉功能的 GPIO 如下：

表 7-22 GPIO 上拉资源分布表

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0	√	√		√	√				√	√	√	√	√	√		√
P1					√								√			√
P2	√		√			√	√		√	√	√	√			√	√

## 7.4 应用指南

### 7.4.1 外部中断

示例如下：

```

GPIO0_PIE = 0x0080;    // 使能 P0[7]输入
NVIC_EnableIRQ(GPIO_IRQn); //使能 GPIO 中断
_enable_irq();        //使能中断
i = 1000;
while(i--);
// P0[7] IO 上外接方波信号
EXTI_CR0 = 0x8000;    // 使能 p0[7]上升沿触发，产生外部中断
while(irq_flag != 2); // 外部信号翻转两次，产生两次中断，irq_flag 在 GPIO 中断处理
                        程序中递增两次
EXTI_CR0 = 0x4000;    // 使能 p0[7]下降沿触发，产生外部中断
while(irq_flag != 4);

```



```
EXTI_CR0 = 0xC000; // 同时使能 P0[7]上升沿、下降沿触发, 产生外部中断
while(irq_flag != 8);
EXTI_CR0 = 0x0000; // 同时禁用 P[7]上下沿触发, 将无法产生外部中断
i = 1000;
while(i--);
if(irq_flag != 8)FAIL;
i = 1000;
while(i--);
PASS;
}
```

#### 7.4.2 使用 GPIO 的模拟功能

将 GPIO 的 IE 和 OE 关闭, 即可使用模拟功能。此时, PAD 通过内部电阻直接与模拟模块相连。

## 8 CRC

### 8.1 概述

CRC 即循环冗余校验码 (Cyclic Redundancy Check)：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查 (CRC) 是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

利用 CRC 进行检错的过程可简单描述为：在发送端根据要传送的  $K$  位二进制码序列，以一定的规则产生一个校验用的  $R$  位监督码 (CRC 码)，附在原始信息后边，构成一个新的二进制码序列数共  $K+R$  位，然后发送出去。在接收端，根据信息码和 CRC 码之间所遵循的规则进行检验，以确定传送中是否出错。这个规则，在差错控制理论中称为“生成多项式”。

### 8.2 基本原理

循环冗余校验码 (CRC) 的基本原理是：在  $K$  位信息码后再拼接  $R$  位的校验码，整个编码长度为  $N$  位，因此，这种编码也叫  $(N, K)$  码。对于一个给定的  $(N, K)$  码，可以证明存在一个最高次幂为  $N-K=R$  的多项式  $G(x)$ 。根据  $G(x)$  可以生成  $K$  位信息的校验码，而  $G(x)$  叫做这个 CRC 码的生成多项式。校验码的具体生成过程为：假设要发送的信息用多项式  $C(x)$  表示，将  $C(x)$  左移  $R$  位 (可表示成  $C(x)*2^R$ )，这样  $C(x)$  的右边就会空出  $R$  位，这就是校验码的位置。用  $C(x)*2^R$  除以生成多项式  $G(x)$  得到的余数就是校验码。

任意一个由二进制位串组成的代码都可以和一个系数仅为‘0’和‘1’取值的多项式一一对应。例如：代码 1010111 对应的多项式为  $x^6+x^4+x^2+x+1$ ，而多项式为  $x^5+x^3+x^2+x+1$  对应的代码 101111。

### 8.3 基本概念

#### 8.3.1 对应关系

多项式和二进制数有直接对应关系： $X$  的最高幂次对应二进制数的最高位，以下各位对应多项式的各幂次，有此幂次项对应 1，无此幂次项对应 0。可以看出： $X$  的最高幂次为  $R$ ，转换成对应的二进制数有  $R+1$  位。

多项式包括生成多项式  $G(X)$  和信息多项式  $C(X)$ 。

如生成多项式为  $G(X)=X^4+X^3+X+1$ ，可转换为二进制数码 11011。

而发送信息为 101111，可转换为数据多项式为  $C(X)=X^5+X^3+X^2+X+1$ 。



### 8.3.2 生成多项式

生成多项式是接受方和发送方的一个约定，也就是一个二进制数，在整个传输过程中，这个数始终保持不变。

在发送方，利用生成多项式对信息多项式做模 2 除生成校验码。在接收方利用生成多项式对收到的编码多项式做模 2 除检测和确定错误位置。

应满足以下条件：

- A、生成多项式的最高位和最低位必须为 1。
- B、当被传送信息（CRC 码）任何一位发生错误时，被生成多项式做除后应该使余数不为 0。
- C、不同位发生错误时，应该使余数不同。
- D、对余数继续做除，应使余数循环。

### 8.3.3 校验码位数

CRC 校验码位数 = 生成多项式位数 - 1。注意有些生成多项式的简记式中将生成多项式的最高位 1 省略了。

### 8.3.4 生成步骤

- 1、将 X 的最高次幂为 R 的生成多项式  $G(X)$  转换成对应的 R+1 位二进制数。
- 2、将信息码左移 R 位，相当于对应的信息多项式  $C(X)*2^R$ 。
- 3、用生成多项式（二进制数）对信息码做除，得到 R 位的余数(注意：这里的二进制做除法得到的余数其实是模 2 除法得到的余数，并不等于其对应十进制数做除法得到的余数。)
- 4、将余数拼到信息码左移后空出的位置，得到完整的 CRC 码。

**【例】**假设使用的生成多项式是  $G(X)=X^3+X+1$ 。4 位的原始报文为 1010，求编码后的报文。

解：

- 1、将生成多项式  $G(X)=X^3+X+1$  转换成对应的二进制除数 1011。
- 2、此题生成多项式有 4 位 (R+1) (注意：4 位的生成多项式计算所得的校验码为 3 位，R 为校验码位数)，要把原始报文  $C(X)$  左移 3 (R) 位变成 1010 000
- 3、用生成多项式对应的二进制数对左移 3 位后的原始报文进行模 2 除（高位对齐），相当于按位异或：

1010000

1011



-----  
0001000

0001011  
-----

0000011

得到的余位 011，所以最终编码为：1010 011

POL=0x13, data=0x77

011101110000000

10010011

01111101000000

10010011

0110100100000

10010011

010000010000

10010011

00010001000

10010011

00011011

## 8.4 寄存器

### 8.4.1 地址分配

CRC 的基地址是 0x4001\_2400。

表 8-1 CRC 寄存器列表

名称	偏移地址	说明
CRC_DR	0x00	CRC 数据（输入信息码/输出编码）寄存器
CRC_CR	0x04	CRC 控制寄存器
CRC_INIT	0x08	CRC 初始码寄存器
CRC_POL	0x0C	CRC 生成多项式对应的二进制码寄存器



## 8.4.2 寄存器描述

### 8.4.2.1 CRC 信息码寄存器

地址：0x4001\_2400

复位值：0x0

表 8-2 CRC\_DR CRC 数据寄存器

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

位置	位名称	说明
[31:0]	DR	存放待编码的信息码和经 CRC 校验后的编码

CRC\_DR 寄存器既用于放入待校验数据，也用于返回校验结果。写入 CRC\_DR 寄存器即触发一次 CRC 计算。待编码数据应在 CR 等寄存器配置完成后最后写入，以触发 CRC 计算开始。

### 8.4.2.2 CRC 控制寄存器

地址：0x4001\_2404

复位值：0x0

表 8-3 CRC\_CR CRC 控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			REV_OUT_TYPE				REV_IN_TYPE				POLY_SIZE				RESET
			RW				RW				RW				WO
			0				0				0				0

位置	位名称	说明
[31:13]		未使用
[12]	REV_OUT_TYPE	是否将 CRC 校验后的编码反转后输出, 即 $b[31]=b[0]$ , $b[30]=b[1]$ ,... $[b0]=b[31]$
[11:10]		未使用
[9:8]	REV_IN_TYPE	待编码数据反转类型 00:不反转



		01:按字节反转,即 b[31]=b[24], b[30]=b[25], ..., b[24]=b[31], ..., b[7]=b[0], b[6]=b[1], ..., b[0]=b[7] 10:按半字(16bit)反转,即 b[31]=b[16], b[30]=b[17], ..., b[16]=b[31], ..., b[15]=b[0], b[14]=b[1], ..., b[0]=b[15] 11:按字反转,即 b[31]=b[0], b[30]=b[1],... [b0]=b[31]
[7:6]		未使用
[5:4]	POLY_SIZE	输出编码(多项式)位宽 00: 32bits 01: 16bits 10: 8bits 11: 7bits
[3:1]		未使用
[0]	RESET	与输入信息码进行CRC计算的数据来源 0:来自于上一次的计算结果 1:来自于CRC_INIT 写入1实现CRC数据重置并自动清零,读回恒为0.

同时需要注意的是,向CRC\_CR.RESET写入1会将CRC\_INIT寄存器复位为0xFFFFFFFF。

如果需要清除CRC的计算结果,应向CRC\_CR.RESET写入1,否则后续CRC计算会以之前的计算结果为初值进行。

#### 8.4.2.3 CRC 初始码寄存器

地址: 0x4001\_2408

复位值: 0x0

表 8-4 CRC\_INIT CRC 初始码寄存器

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

位置	位名称	说明
[31:0]	INIT	存放初始码

初CRC\_DR与CRC\_INIT相异或后开始进行CRC校验计算。



8.4.2.4 CRC 生成码寄存器

地址：0x4001\_240C

复位值：0x0

表 8-5 CRC\_POL CRC 生成码寄存器

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

位置	位名称	说明
[31:0]	POL	存放生成多项式对应的生成码



## 9 ADC

### 9.1 概述

LKS32MC05x 系列芯片集成了 1 个 12bit SAR ADC，16 路输入通道。主要有以下特性：

- 最高 3Msps 采样率，48MHz 工作频率。
- 支持 16 通道选择。
- 支持软件、硬件触发功能。
- 可以与 MCPWM、Timer 单元联动，触发指示信号可通过 GPIO 送出调试
- 支持单段、双段、四段自定义采样序列采样，序列次数和通道号可灵活配置。
- 支持左对齐、右对齐模式。

ADC 采样的量词含义约定：

1 次采样：完成对应的一个通道的模拟信号量到数据信号量的采样转换并存储数字量至 ADC\_DATx 寄存器；

1 段采样：可能包含 1 次或若干次采样，若干次采样可以是相同的模拟量通道，也可以是不同的模拟量通道。采样开始通常由 MCPWM、UTimer 或软件进行触发，一个触发信号完成一段采样，采样完成后产生相应的段采样完成中断；以 MCPWM 触发的四段采样为例，每段采样 3 次（即完成 3 个模拟量的采样），TADC[0]触发 ADC 开始第一段采样，第一段采样完成后 ADC 进入等待状态，等待 TADC[1]触发事件发生；TADC[1]发生后，触发 ADC 开始第二段采样；同理，TADC[2]/TADC[3] 分别触发第三段和第四段采样。

1 轮采样：可能包含 1 段、2 段或 4 段采样，每段分别由特定触发信号触发；ADC 完成一轮采样后回归空闲状态等待下次触发

ADC 最高工作时钟为 48MHz，采样时间 4-68 个 ADC 时钟周期 (Cycle) 可配置，数据转换时间固定为 12Cycle。以采样时间 4Cycle 为例，采样+转换最少 16Cycle，即 ADC 最高转换速率是  $48/16=3\text{Msps}$ 。但考虑到采样建立时间，建议设置采样时间 12Cycle 或以上，对应转换速度率  $48/(12+12)=2\text{Msps}$ 。

#### 9.1.1 功能框图

ADC 接口包括 16 个数据寄存器（ADC 16 次采样各个通道模拟量对应的数字量），以及若干控制寄存器。

数据寄存器 ADC\_DATx 用于存储 ADC 第 x 次采样得到的数据量。被转换的模拟信号来源由寄存器 ADC\_CHNx 中的某 4bit 进行选择（详见 9.2.3 信号来源寄存器章节）。以 ADC\_CHN0 为例，位[3:0]选择第 0 次采样的模拟通道号，通道号 CH0~CH15 任选，若 ADC\_CHN0[3:0]=0, ADC\_CHN0[7:4]=3，则第 0 个采样的模拟量对应通道 CH0，第 1 个采样的模拟量对应通道 CH3，以此类推。

分段采样次数寄存器 ADC\_CHNT 控制每段采样的次数，1~15 对应 1~15 次。



控制逻辑根据配置寄存器 ADC\_CFG 选择来自 MCPWM 或 UTimer 的触发信号启动一段采样或者软件触发启动。MCPWM/UTimer 会送出定时触发信号 TADC[0]/TADC[1]/TADC[2]/TADC[3]，可选择 TADC[0]/TADC[1]/TADC[2]/TADC[3]作为触发信号。触发信号的选择保存在控制寄存器中。

一段转换（一段内的所有通道采样转换完毕）完成，触发 ADC 转换完成中断。多段触发模式下，每一段转换完成可触发产生一个转换完成中断。

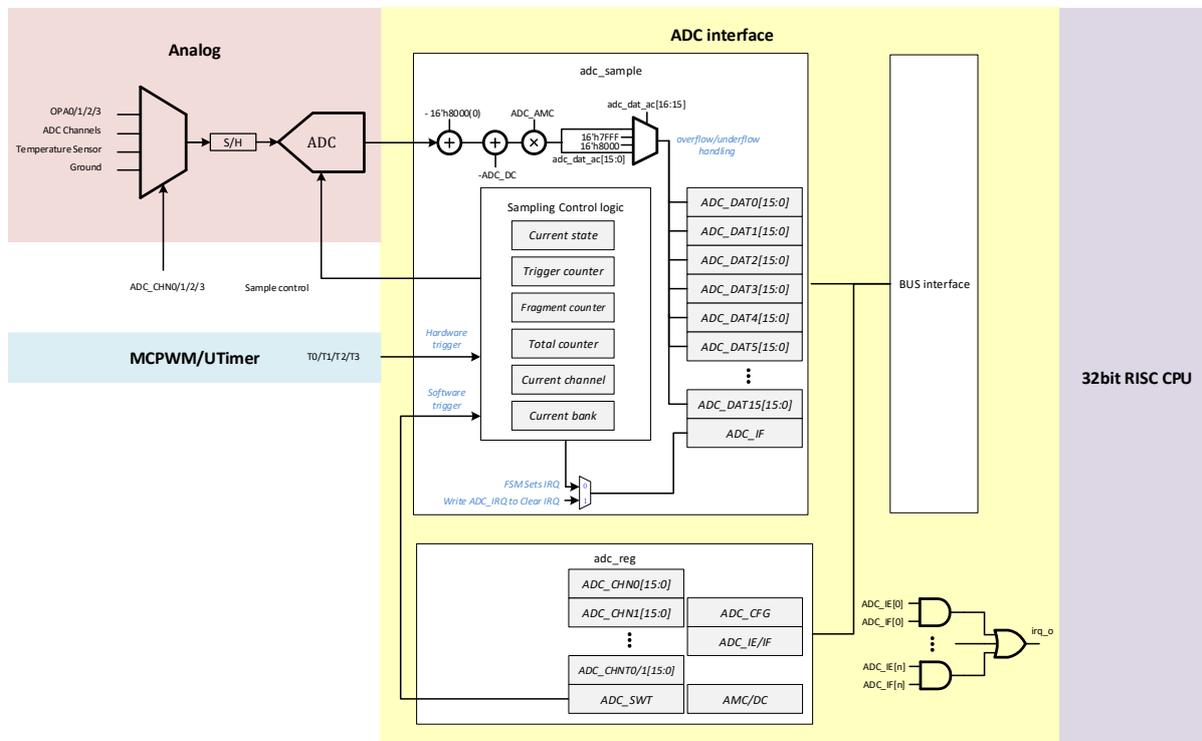


图 9-1 ADC 采集模块功能框图

来源可选使得用户可以灵活配置采样顺序、以及采样信号来源，甚至实现对单个信号多次采样的目的。控制寄存器使得用户可以配置采样个数，提高采样频率/降低采样功耗。

### 9.1.2 ADC 触发方式

- 支持单段触发、两段触发、四段触发完成采样
- 单段触发可以设置触发事件发生次数
- 两段触发的触发源只能为 MCPWM/UTimer 的定时信号 TADC[0]+TADC[1]，或两次软件触发
- 四段触发的触发源只能为 MCPWM/UTimer 的定时信号 TADC[0]+TADC[1]+TADC[2]+TADC[3]，或四次软件触发
- 每段触发完成均可产生中断
- 触发指示信号可以通过 GPIO 送出用于调试

### 9.1.3 ADC 输出数制

ADC 输出数据为 12bit 补码，输入信号 0 对应 12h'0000\_0000\_0000，以 1 倍增益配置为例，输入信号-2.4V 对应 12h'1000\_0000\_0000，输入信号+2.4V 对应 12h'0111\_1111\_1111。ADC 转换后的



12bit 补码需扩展为 16BIT 存入 16bit 位宽的采样数据寄存器，左对齐/右对齐可根据配置寄存器进行设置。以 12'h1000\_0000\_1101 为例，如果配置为左对齐，右侧补 4 个 0，存入 ADCx\_DAT 的值为 16'h1000\_0000\_1101\_0000；如果配置为右对齐，左侧进行符号扩展，存入 ADCx\_DAT 的值为 16'h1111\_1000\_0000\_1101。推荐统一使用左对齐方式。

需要注意的是，由于存在增益校正和直流偏置校正，ADC 最终数据可能会超过 12bit 有符号数的表示范围，比如在右对齐的模式下，ADC 某次转换的数字量可能为 0xF745，此时直接进行低 12bit 的截取取出 0x745，会使得负数被作为正数处理，即发生溢出错误。亦或者 ADC 某次转换的数字量可能为 0x0810，此时直接进行低 12bit 的截取取出 0x810，会使得正数被错误地当做负数处理。因此需要将 ADC 数据作为 16bit 有符号数进行处理。

表 9-1 ADC 输出数字量数制转换

ADC 一倍增益输入模拟量数值/V	ADC 2/3 倍增益输入模拟量数值/V	转为有符号数后的数值
2.4	3.6	12'h0111_1111_1111
0	0	12'h0000_0000_0000
-2.4	-3.6	12'h1000_0000_0000

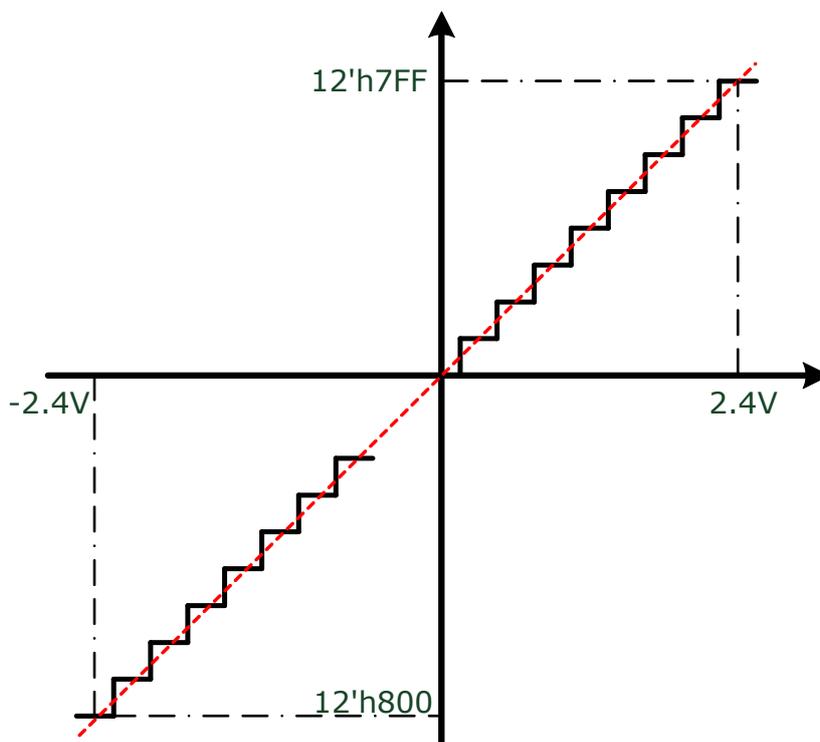


图 9-2 一倍增益设置下 ADC 模数转换数制量程

### 9.1.4 ADC 量程

ADC 有两种增益模式：高增益（1 倍）和低增益（2/3 倍），针对这两种增益，ADC 的量程也相应有所区别。1 倍增益模式下，对应最大±2.4V 的输入信号幅度，2/3 倍增益模式下，对应最大±3.6V 的输入信号幅度。

在 ADC 采样通道配置为运放的输出信号时（即 OPA0~OPA1），应选择合适的运放增益，使得



具体应用上的最大信号可被放大到接近 $\pm 3.3V$ 的水平,同时将 ADC 配置为  $2/3$  倍增益。举例来说,相线电流最大  $100A$  (正弦波有效值), MOS 内阻 (假设为 MOS 内阻采样) 为  $5m\Omega$ , 则运放的最大输入信号幅值为 $\pm 707mV$ 。此时应该选择运放的放大倍数为  $4.5$  倍(放大倍数选择方式详见 5.3.8), 则放大后的信号约为 $\pm 3.18V$ 。

如果因为客观原因, 运放的输出信号经放大后, 最大信号仍然小于 $\pm 2.4V$ , 则应将 ADC 的增益配置为  $1$  倍。

在 ADC 采样通道配置为 GPIO 复用口输入的信号时, 同样根据信号的最大幅度来选择 ADC 增益。由于 IO 口的限制, GPIO 复用口输入的信号范围只能在 $-0.3V \sim AVDD+0.3V$  之间。

### 9.1.5 ADC 校正

ADC 硬件接口模块可以进行直流偏置校正与增益校正。

ADC\_AMC 存储的是增益校正系数  $AMP_{correction}$ , 为  $10bit$  无符号定点数, ADC\_AMC[9]为整数部分, ADC\_AMC[8:0]为小数部分。可以表示数值在  $1$  附近的定点数。

ADC\_DC 存储的是 ADC 的直流偏置, 通常在校正阶段通过测量通道  $15$  (从  $0$  开始计数) 的 AVSS (内部地)得到 ADC 直流偏置数值并存入 flash 中, 并在系统加载阶段由软件将直流偏置写入 ADC\_DC 寄存器中。

记 ADC 输出的数字量为  $D_{ADC}$ ,  $D_{ADC}$  对应的真实值为  $D$ ,  $D_0$  为编码数制的  $0$ , 则

$$D = \text{Saturation}((D_{ADC} - D_0) * AMP_{correction} - DC_{offset})$$

最终硬件会将进行校正后的  $D$  存入相应的采样数据寄存器。

### 9.1.6 ADC 信号来源

表 9-2 ADC 采样信号来源

ADC_CHNx	采样信号
0	OPA0 输出
1	OPA1 输出
2	ADC_CH2
3	ADC_CH3
4	ADC_CH4
5	ADC_CH5
6	ADC_CH6
7	ADC_CH7
8	ADC_CH8
9	ADC_CH9
A	ADC_CH10
B	ADC_CH11
C	ADC_CH12
D	ADC_CH13
E	温度传感器
F	内部地/SWCLK

当 SYS\_AFE\_REG6.CH15EN=1 时, ADC 通道  $15$  选通 SWCLK, 而不再是内部地。



## 9.1.7 ADC 配置流程

推荐配置流程：

1. 打开 ADC 模拟开关，选择 ADC 工作频率

通过配置寄存器 [SYS AFE REG5.ADCPDN](#) 可以开启 ADC，ADC 开启前，需要先开启 BGP、4M RC 时钟和 PLL 模块。通过配置寄存器 [SYS AFE REG7.ADCCLKSEL\[5:4\]](#) 设置 ADC 工作频率，00 为 48MHz，10 为 12MHz，11 为 24MHz。

2. 配置 ADC 数据输出格式

ADC 的输出格式可配置为左对齐或者右对齐，配置的是 [ADC\\_CFG.DATA\\_ALIGN](#) 寄存器，0 为左对齐，1 为右对齐。

3. 配置 ADC 采样模式

ADC 的采样模式可配置为单段、双段、四段采样模式，配置的是 [ADC\\_CFG.TRG\\_MODE\[9:8\]](#) 寄存器，00 为单段采样模式，01 为两段采样模式，11 为四段采样模式。

4. 配置 ADC 触发事件

ADC 采样的触发事件选择配置的是 [ADC\\_CFG.SEL](#) 寄存器。单段采样模式下可以设置触发一次采样所需的事件数，配置的是 [ADC\\_CFG.SINGLE\\_TCNT\[7:4\]](#) 寄存器，设置范围是 0~15，0 表示一次事件即触发，15 表示 16 次事件才触发。

5. 配置 ADC 量程

ADC 的两种增益模式可以通过配置 [SYS AFE REG0.GA\\_AD](#) 增益寄存器进行选择，0 为低增益（2/3 倍），1 为高增益（1 倍）。

6. 配置 ADC 通道数，选择采样信号源

配置各段采样模式下，采样的通道个数，配置的是 [ADC\\_CHNT](#) 寄存器，设置范围是 1~15，1 代表一个通道。ADC 的采样信号源配置，通过配置 [ADC\\_CHN0](#)、[ADC\\_CHN1](#) 等寄存器选择。

7. 配置 ADC 中断

ADC 一共有六种中断：第一段~第四段采样完成中断、软件触发发生在非空闲状态下的中断、硬件触发发生在非空闲状态下的中断。通过配置 [ADC\\_IE](#) 寄存器可以使能以上中断。即使未开启中断，中断事件仍能置位 [ADC\\_IF](#)，但不会提出中断请求。

## 9.2 寄存器

### 9.2.1 地址分配

ADC 的基地址是 0x40011400。



表 9-3 ADC0 寄存器列表

名称	偏移地址	说明
ADC0_DAT0	0x00	ADC 第 0 次采样数据
ADC0_DAT1	0x04	ADC 第 1 次采样数据
ADC0_DAT2	0x08	ADC 第 2 次采样数据
ADC0_DAT3	0x0C	ADC 第 3 次采样数据
ADC0_DAT4	0x10	ADC 第 4 次采样数据
ADC0_DAT5	0x14	ADC 第 5 次采样数据
ADC0_DAT6	0x18	ADC 第 6 次采样数据
ADC0_DAT7	0x1C	ADC 第 7 次采样数据
ADC0_DAT8	0x20	ADC 第 8 次采样数据
ADC0_DAT9	0x24	ADC 第 9 次采样数据
ADC0_DAT10	0x28	ADC 第 10 次采样数据
ADC0_DAT11	0x2C	ADC 第 11 次采样数据
ADC0_DAT12	0x30	ADC 第 12 次采样数据
ADC0_DAT13	0x34	ADC 第 13 次采样数据
ADC0_DAT14	0x38	ADC 第 14 次采样数据
ADC0_DAT15	0x3C	ADC 第 15 次采样数据
ADC0_CHN0	0x40	ADC 第 0~3 次采样信号选择
ADC0_CHN1	0x44	ADC 第 4~7 次采样信号选择
ADC0_CHN2	0x48	ADC 第 8~11 次采样信号选择
ADC0_CHN3	0x4C	ADC 第 12~15 次采样信号选择
ADC0_CHNT	0x50	ADC 各种触发模式下采样次数
ADC0_CFG	0x54	ADC 对齐模式配置
ADC0_SWT	0x58	ADC 软件触发
ADC0_DC	0x60	ADC DC offset
ADC0_AMC	0x64	ADC 增益校正
ADC0_IE	0x68	ADC 中断使能
ADC0_IF	0x6C	ADC 中断标志

## 9.2.2 采样数据寄存器

### 9.2.2.1 ADC0\_DAT0

地址：0x4001\_1400

复位值：0x0

表 9-4 ADC0\_DAT0 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT0															
RW															



0
---

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT0	ADC 第 0 次采样数据

### 9.2.2.2 ADC0\_DAT1

地址：0x4001\_1404

复位值：0x0

表 9-5 ADC0\_DAT1 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT1															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT1	ADC 第 1 次采样数据

### 9.2.2.3 ADC0\_DAT2

地址：0x4001\_1408

复位值：0x0

表 9-6 ADC0\_DAT2 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT2															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT2	ADC 第 2 次采样数据



### 9.2.2.4 ADC0\_DAT3

地址：0x4001\_140C

复位值：0x0

表 9-7 ADC0\_DAT3 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT3															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT3	ADC 第 3 次采样数据

### 9.2.2.5 ADC0\_DAT4

地址：0x4001\_1410

复位值：0x0

表 9-8 ADC0\_DAT4 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT4															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT4	ADC 第 4 次采样数据

### 9.2.2.6 ADC0\_DAT5

地址：0x4001\_1414

复位值：0x0

表 9-9 ADC0\_DAT5 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT5															
RW															



0
---

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT5	ADC 第 5 次采样数据

### 9.2.2.7 ADC0\_DAT6

地址：0x4001\_1418

复位值：0x0

表 9-10 ADC0\_DAT6 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT6															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT6	ADC 第 6 次采样数据

### 9.2.2.8 ADC0\_DAT7

地址：0x4001\_141C

复位值：0x0

表 9-11 ADC0\_DAT7 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT7															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT7	ADC 第 7 次采样数据



### 9.2.2.9 ADC0\_DAT8

地址：0x4001\_1420

复位值：0x0

表 9-12 ADC0\_DAT8 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT8															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT8	ADC 第 8 次采样数据

### 9.2.2.10 ADC0\_DAT9

地址：0x4001\_1424

复位值：0x0

表 9-13 ADC0\_DAT9 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT9															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT9	ADC 第 9 次采样数据

### 9.2.2.11 ADC0\_DAT10

地址：0x4001\_1428

复位值：0x0

表 9-14 ADC0\_DAT10 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT10															
RW															



0
---

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT10	ADC 第 10 次采样数据

### 9.2.2.12 ADC0\_DAT11

地址：0x4001\_142C

复位值：0x0

表 9-15 ADC0\_DAT11 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT11															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT11	ADC 第 11 次采样数据

### 9.2.2.13 ADC0\_DAT12

地址：0x4001\_1430

复位值：0x0

表 9-16 ADC0\_DAT12 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT12															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT12	ADC 第 12 次采样数据



### 9.2.2.14 ADC0\_DAT13

地址：0x4001\_1434

复位值：0x0

表 9-17 ADC0\_DAT13 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT13															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT13	ADC 第 13 次采样数据

### 9.2.2.15 ADC0\_DAT14

地址：0x4001\_1438

复位值：0x0

表 9-18 ADC0\_DAT14 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT14															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT14	ADC 第 14 次采样数据

### 9.2.2.16 ADC0\_DAT15

地址：0x4001\_143C

复位值：0x0

表 9-19 ADC0\_DAT15 采样数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT15															
RW															



0

位置	位名称	说明
[31:16]		未使用
[15:0]	DAT15	ADC 第 15 次采样数据

## 9.2.3 信号来源寄存器

### 9.2.3.1 ADC0\_CHN0

地址: 0x4001\_1440

复位值: 0x0

表 9-20 ADC0\_CHN0 信号来源寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS3				DS2				DS1				DS0			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	DS3	ADC 第 3 次采样信号选择
[11:8]	DS2	ADC 第 2 次采样信号选择
[7:4]	DS1	ADC 第 1 次采样信号选择
[3:0]	DS0	ADC 第 0 次采样信号选择

### 9.2.3.2 ADC0\_CHN1

地址: 0x4001\_1444

复位值: 0x0

表 9-21 ADC0\_CHN1 信号来源寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS7				DS6				DS5				DS4			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
----	-----	----



[31:16]		未使用
[15:12]	DS7	ADC 第 7 次采样信号选择
[11:8]	DS6	ADC 第 6 次采样信号选择
[7:4]	DS5	ADC 第 5 次采样信号选择
[3:0]	DS4	ADC 第 4 次采样信号选择

## 9.2.3.3 ADC0\_CHN2

地址：0x4001\_1448

复位值：0x0

表 9-22 ADC0\_CHN2 信号来源寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS11				DS10				DS9				DS8			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	DS11	ADC 第 11 次采样信号选择
[11:8]	DS10	ADC 第 10 次采样信号选择
[7:4]	DS9	ADC 第 9 次采样信号选择
[3:0]	DS8	ADC 第 8 次采样信号选择

## 9.2.3.4 ADC0\_CHN3

地址：0x4001\_144C

复位值：0x0

表 9-23 ADC0\_CHN3 信号来源寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS15				DS14				DS13				DS12			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	DS15	ADC 第 15 次采样信号选择
[11:8]	DS14	ADC 第 14 次采样信号选择



[7:4]	DS13	ADC 第 13 次采样信号选择
[3:0]	DS12	ADC 第 12 次采样信号选择

## 9.2.4 分段通道数寄存器

### 9.2.4.1 ADC0\_CHNT

地址：0x4001\_1450

复位值：0x0

表 9-24 ADC0\_CHNT 分段通道数寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S4				S3				S2				S1			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	S4	四段采样模式下第四段采样次数
[11:8]	S3	四段采样模式下第三段采样次数
[7:4]	S2	两段或四段采样模式下第二段采样次数
[3:0]	S1	单段、两段或四段采样模式下第一段采样次数

注意：每段采样次数不允许配置为 0。1 表示 1 次采样，2 表示 2 次采样，.....，12 表示 12 次采样，.....，15 表示 15 次采样。

## 9.2.5 配置寄存器

### 9.2.5.1 ADC0\_CFG

地址：0x4001\_1454

复位值：0x0

表 9-25 ADC0\_CFG 模式配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSEL		FSM_RESET		DATA_ALIGN		TRG_MODE		SINGLE_TCNT				TRG_EN			
RW		WO		RW		RW		RW				RW			
0		0		0		0		0				0			



位置	位名称	说明
[31:13]		未使用
[12]	SEL	TADC 触发来源选择。0: MCPWM, 1: UTimer
[11]	FSM_RESET	状态机复位控制信号。软件写入 1 产生复位, 将 ADC 内部状态机回到初始状态, 完成后自动回到 0。该复位控制, 不影响 ADC 其它寄存器的配置值。
[10]	DATA_ALIGN	ADC_DAT 对齐方式 0: 左对齐, 右端补 4'h0, 1: 右对齐, 左端补 4bit 符号位
[9:8]	TRG_MODE	触发模式选择 0: 单段触发; 1: 两段触发; 2: 保留; 3: 四段触发
[7:4]	SINGLE_TCNT	单段触发模式下触发一次采样所需的事件数。 0: 表示需要发生 1 次事件才能触发一段采样 1: 表示需要发生 2 次事件才能触发一段采样 ..... 15: 表示需要发生 16 次事件才能触发一段采样
[3:0]	TRG_EN	TADC 触发 ADC 采样使能信号。高电平有效, 默认为低。 TRG_EN[0]: TADC[0]使能开关 TRG_EN[1]: TADC[1]使能开关 TRG_EN[2]: TADC[2]使能开关 TRG_EN[3]: TADC[3]使能开关 对应位置高表示使能, 拉低表示关闭。

ADC 的触发来源为 MCPWM 或 UTimer, 不可以同时使用 MCPWM 和 UTimer 进行触发, 只能在二者之中选一。触发事件共计四个--TADC[3:0]。

若 ADC 的触发来源为 UTimer, ADC 采样触发事件为 UTimer\_T0/ UTimer\_T1/ UTimer\_T2/ UTimer\_T3; 其依次对应 Timer2 通道 0/1、Timer3 通道 0/1 的比较事件。

若 ADC 的触发来源为 MCPWM。ADC 采样触发事件为 MCPWM 产生的 MCPWM\_T0/ MCPWM\_T1/ MCPWM\_T2/ MCPWM\_T3。

UTimer 对 ADC 的触发信号可以通过配置 GPIO 为第 8 功能, 即 Timer2/3 功能送出用于捕捉调试。

MCPWM 对 ADC 的触发信号可以通过配置 GPIO 为第 9 功能, 即 ADC\_TRIGGER 功能送出用于捕捉调试。每发生一次 ADC 触发, ADC\_TRIGGER 信号翻转一次。

## 9.2.6 软件触发寄存器

### 9.2.6.1 ADC0\_SWT

地址: 0x4001\_1458

复位值: 0x0



表 9-26 ADC0\_SWT 软件触发寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWT															
WO															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	SWT	写入数据为 0x5AA5 时，产生一次软件触发

注意，软件触发采集寄存器为只写寄存器，且只有写入数据为 0x5AA5 时产生软件触发事件，一次总线的写入产生一次软件触发，数据写入产生一个软件触发后寄存器自动清零，等待后续的软件触发到来。

### 9.2.7 直流偏置寄存器

ADC 的一个信号源为 GND，通过对这个通道的测量可以得到系统的 DC 偏置。

通常系统初始化后会进行一次 GND 信号的测量，并将测量值存入 DC offset 寄存器，后续每次其他 channel 的信号的采样值会自动减去 DC offset，再存入转换后的数字量寄存器。

考虑到信号误差，去除 DC offset 的信号可能会发生溢出，对于溢出的数据会做饱和处理，以使信号范围在 -2.4V~2.4V 或 -3.6V~3.6 之间。芯片出厂时已经过工厂标定，标定数据存放在 Flash info 中，芯片上电自动完成校准参数的加载。ADC 模块在初始化的时候，需要根据数据左右对齐模式配置 DC offset，可以参看芯片供应商提供的库函数。使用 SYS\_SFT\_RST 软复位 ADC 模块后，ADC 内部的寄存器会被复位。需要重新使用 ADC 初始化函数从 NVR 中读取 DC/AMC 等校准参数。

#### 9.2.7.1 ADC0\_DC

地址：0x4001\_1460

复位值：0x0

表 9-27 ADC0\_DC 直流偏置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DC_OFFSET															
RW															
0															

位置	位名称	说明
[31:16]		未使用



[15:0]	DC_OFFSET	采样电路增益 ADC DC offset
--------	-----------	----------------------

考虑到 ADC 的 DC offset 是接近 0 的数值，因此此处寄存器仅有低 10bit 是有实际实现的，高 6bit 仅仅是 ADC\_DC[9]的符号扩展，同时 ADC\_DC 寄存器参与 ADC 数据校正时也会进行符号扩展。即如果向 ADC\_DC 写入 0x12B0；实际写入的是 0x2B0，而读出时会读出 0xFFB0。

此处存入的 ADC\_DC 值应该对应的是右对齐的 offset 数值，当 ADC\_CFG 寄存器配置为左对齐时，硬件会根据对齐的设置，自动调整 ADC\_DC 参与 ADC 校正。具体来说，右对齐时，ADC\_DC[15:0] 直接参与校正运算，左对齐时，ADC\_DC[15:0]会先左移 4 位然后参与 ADC 校正运算。

## 9.2.8 增益校正寄存器

### 9.2.8.1 ADC0\_AMC

地址：0x4001\_1464

复位值：0x200

表 9-28 ADC0\_AMC 增益校正寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										AM_CALI					
										RW					
										0					

位置	位名称	说明
[31:10]		未使用
[9:0]	AM_CALI	采样电路 ADC 增益校正寄存器

## 9.2.9 中断寄存器

### 9.2.9.1 ADC0\_IE

地址：0x4001\_1468

复位值：0x0

表 9-29 ADC0\_IE 中断使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										HERR_IE	SERR_IE	S4_IE	S3_IE	S2_IE	S1_IE
										RW	RW	RW	RW	RW	RW



	0	0	0	0	0	0
--	---	---	---	---	---	---

位置	位名称	说明
[31:6]		未使用
[5]	HERR_IE	硬件触发发生在非空闲状态中断使能
[4]	SERR_IE	软件触发发生在非空闲状态中断使能
[3]	S4_IE	第四段采样完成中断使能
[2]	S3_IE	第三段采样完成中断使能
[1]	S2_IE	第二段采样完成中断使能
[0]	S1_IE	第一段采样完成中断使能

## 9.2.9.2 ADC0\_IF

地址: 0x4001\_146C

复位值: 0x0

表 9-30 ADC0\_IF 中断标志寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
											HERR_IF	SERR_IF	S4_IF	S3_IF	S2_IF	S1_IF
											RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
											0	0	0	0	0	0

位置	位名称	说明
[31:6]		未使用
[5]	HERR_IF	硬件触发发生在非空闲状态中断标志
[4]	SERR_IF	软件触发发生在非空闲状态中断标志
[3]	S4_IF	第四段采样完成中断标志
[2]	S3_IF	第三段采样完成中断标志
[1]	S2_IF	第二段采样完成中断标志
[0]	S1_IF	第一段采样完成中断标志

以上 ADC0\_IF 标志位, 0: 表示未发生中断, 1: 表示发生过中断, 写 1 清零。



## 9.3 应用指南

### 9.3.1 ADC 采样触发模式

ADC 支持一段、两段、四段采样模式，每段采样需要特定的外部事件来触发开始，每段采样支持不同采样次数和采样信号通道配置。ADC 内部的状态转移描述如下，共有 8 个状态分别为采样状态 0~3，空闲状态 0~3。

#### 第一次触发

来自 MCPWM/UTimer 的定时事件 TADC[0]/TADC[1]/TADC[2]/TADC[3]可以触发 ADC 采样。可以选择四个触发源的任何一个或者几个触发采样。也可以通过向 ADC\_SWT 写入命令字的方式 16'h5AA5 软件触发 ADC 采样。

#### 第一段采样

判断是否为一段采样。

是：采样次数达到预设值 ADC\_CHNT[3:0]，ADC 回到空闲状态 0；采样次数未达到预设值，继续采样。

否：采样次数达到预设值 ADC\_CHNT[3:0]，ADC 进入空闲状态 1（两段或四段采样第一段完成，等待触发第二段）；采样次数未达到预设值，继续第一段采样。

#### 第二段触发

#### 第二段采样

第二段采样次数到达预设值 ADC\_CHNT[7:4]，判断是否为两段采样。

是：结束本次采样，回到空闲状态 0。

否：进入空闲状态 2，等待第三次触发及第四次触发完成采样。

#### 第三段触发

#### 第三段采样

第三段采样次数到达预设值 ADC\_CHNT[11:8]，进入空闲状态 3。

#### 第四段触发

#### 第四段采样

第四段采样通道数到达预设值 ADC\_CHNT[15:12]，回到空闲状态 0。

各种硬件触发模式的触发条件汇总如表 9-30 ADC 采样触发模式所示。其中单段采样模式较为特殊，可以通过 ADC\_CFG 寄存器设置，一次 TADC 事件即触发采样，还是多次 TADC 事件才触发采样；而两段、四段采样模式仅支持一次相应的 TADC 事件即触发一段采样。

此外 ADC 模块也支持通过软件写入特殊数值的方式触发采样，软件触发也仅支持写入一次即触发。



表 9-31 ADC 采样触发模式

	单段触发	两段触发	四段触发
TADC 触发	None(TADC 触发使能未打开)	第一段 TADC[0] 第二段 TADC[1]	第一段 TADC[0] 第二段 TADC[1] 第三段 TADC[2] 第四段 TADC[3]
	C 次 TADC[0]		
	C 次 TADC[1]		
	C 次 TADC[2]		
	C 次 TADC[3]		
C 次* TADC[0]/TADC[1]/ TADC[2]/TADC[3]			
软件触发	向 ADC_SWT 写入 16'h5aa5	第一段向 ADC_SWT 写入 16'h5aa5 第二段向 ADC_SWT 写入 16'h5aa5	第一段向 ADC_SWT 写入 16'h5aa5 第二段向 ADC_SWT 写入 16'h5aa5 第三段向 ADC_SWT 写入 16'h5aa5 第四段向 ADC_SWT 写入 16'h5aa5

\*C 次通过 ADC\_CFG.SINGLE\_TCNT 设置。ADC\_CFG.SINGLE\_TCNT 只在单段触发下使用，如果同时使能了 TADC[3:0]，则 4 个触发源都会被计数，到 SINGLE\_TCNT 次触发一次 ADC 采样转换。

### 9.3.1.1 单段触发模式

单段触发收到一次触发完成一段采样动作，一段采样可能包含多次对模拟信号的采样，次数由分段采样次数寄存器配 ADC\_CHNT 进行配置，寄存器数值为 1~16 时，对应的采样次数为 1~16。

假设单段采样配置通道数目为 4，则采样转换后的数据会依次填充到 ADC\_DAT0、ADC\_DAT1、ADC\_DAT2、ADC\_DAT3。

触发事件可以是来自外部的 MCPWM/UTimer 定时信号 TADC[0]、TADC[1]、TADC[2]、TADC[3]、发生到预设次数、或者为软件触发。

每个采样的信号源通过信号来源寄存器 ADC\_CHN0/1/2/3 进行配置选定，信号源的选定需在触发前完成，且在一次采样过程完成前不应该改变。

完成一段采样动作后，进入空闲状态，并产生采样完成中断。

以 MCPWM 触发单段采样为例，设置 TADC[2]发生 4 次才进行触发，状态转移如图 9-3 ADC 单段采样状态转移图所示。



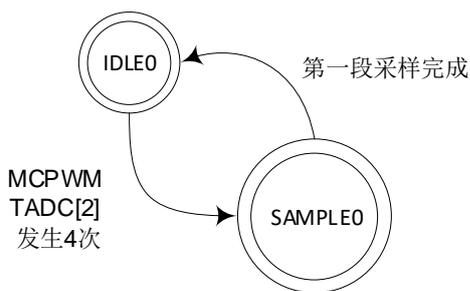


图 9-3 ADC 单段采样状态转移图

### 9.3.1.2 两段触发模式

两段触发需要两次触发才能完成完整的一轮采样。第一个触发到达时进行第一段采样，第二个触发到达时进行第二段采样。

假设两段采样配置通道数目分别为 2 和 3,则第一段采样转换后的数据会依次填充到 ADC\_DAT0、ADC\_DAT1，第二段采样转换后的数据会依次填充到 ADC\_DAT2、ADC\_DAT3、ADC\_DAT4。

触发事件可以是来自外部的 MCPWM 定时信号 TADC[0]和 TADC[1]或两次软件触发。

TADC[0]或软件触发发生后，先进行 ADC\_CHNT[3:0]次采样，完成后进入空闲状态并等待下一个触发信号的到来；TADC[1]或软件触发作为第二个触发信号发生后，再进行 ADC\_CHNT[7:4]次采样。采样次数均通过分段采样次数寄存器 ADC\_CHNT 进行配置。

每个采样的信号源通过寄存器配置选定，信号源的选定需在触发前完成，且在一次采样过程完成前不应该改变。

软件触发较硬件触发的优先级低，在硬件触发采样的过程中发生软件触发，状态机不予处理，而产生一个错误中断。即只有状态机处于空闲状态时才会处理软件触发的采样请求。如果需要使用软件触发采样，需要确保硬件触发已经关闭。然后通过向 ADC\_SWT 寄存器写入 0x5AA5 以产生一次软件触发。

以两次软件触发两段采样为例，状态转移如图 9-4 ADC 两段采样状态转移图所示。

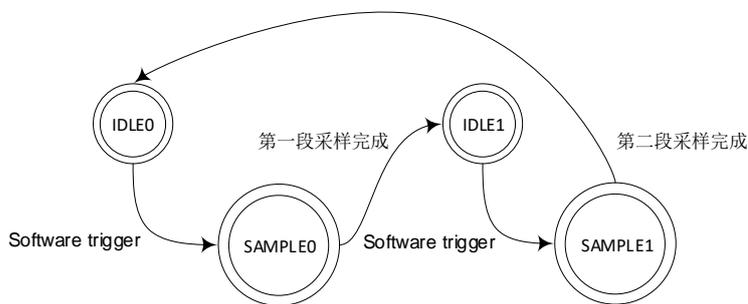


图 9-4 ADC 两段采样状态转移图

### 9.3.1.3 四段触发模式

与两段触发类似。四段的触发源分别为 TADC[0]、TADC[1]、TADC[2]、TADC[3],且必须为 MCPWM



TADC[0]/TADC[1]/TADC[2]/TADC[3]顺序触发 ADC 的四段采样；或者也可以是 4 次软件触发采样。四段采样的采样次数分别为 ADC\_CHNT1[12:8]、ADC\_CHNT1[4:0]、ADC\_CHNT0[12:8]、ADC\_CHNT0[4:0]。以 MCPWM TADC[0]/TADC[1]/TADC[2]/TADC[3]触发四段采样为例的状态转移如图 9-5 ADC 四段采样状态转移图所示。

假设四段采样配置通道数目分别为 2、3、1、5，则第一段采样转换后的数据会依次填充到 ADC\_DAT0、ADC\_DAT1，第二段采样转换后的数据会依次填充到 ADC\_DAT2、ADC\_DAT3、ADC\_DAT4，第三段采样转换后的数据会填充到 ADC\_DAT5，第四段采样转换后的数据会依次填充到 ADC\_DAT6、ADC\_DAT7、ADC\_DAT8、ADC\_DAT9、ADC\_DAT10

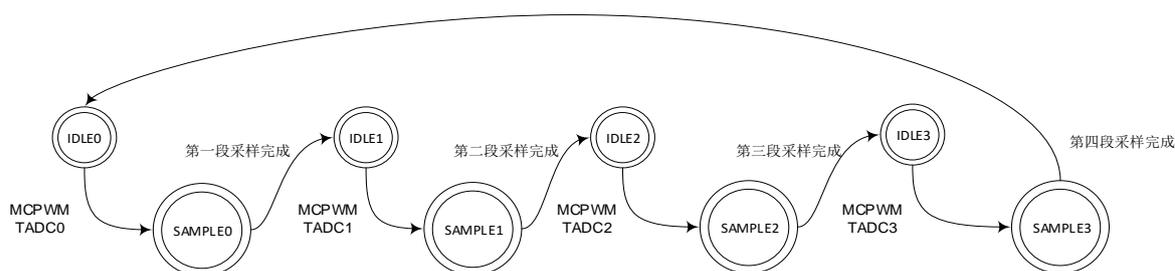


图 9-5 ADC 四段采样状态转移图

为使用 MCPWM 定时器产生 ADC 采样触发信号，需要配置 MCPWM\_TMR0/ MCPWM\_TMR1/ MCPWM\_TMR2/ MCPWM\_TMR3 等寄存器，对应 TADC0/1/2/3 发生时的 MCPWM 计数值，此外需要配置 MCPWM\_TH 设置计数器计数范围以及 MCPWM\_TCLK 设置计数时钟频率并使能时钟。

## 9.3.2 中断

### 9.3.2.1 单段触发采样完成中断

采样完成产生一个中断。

### 9.3.2.2 两段触发采样完成中断

第一段采样完成产生一个中断，第二段采样完成产生一个中断。

### 9.3.2.3 四段触发采样完成中断

第一段采样完成产生一个中断，第二段采样完成产生一个中断，第三段采样完成产生一个中断，第四段采样完成产生一个中断。

### 9.3.2.4 ADC\_IF 触发所需时间

CPU 软件运行于 PLL 时钟域，ADC 模块运行于 ADC 时钟域。典型地，ADC 时钟周期为 PLL 时钟周期（总线时钟周期）2 倍。ADC\_IF 的清除需要操作 ADC 时钟域内的寄存器，最多需要 20 个总线周期。建议 ADC\_IF 标志在中断函数入口处进行读取清除。如果清除语句在中断函数最后，可能导致中断函数退出后标志仍未清除，导致再次进入中断。

同理，如果软件在清除 ADC\_IF 后立刻查询 ADC\_IF，可能 ADC\_IF 仍为原始值



### 9.3.3 配置修改

建议在 ADC 中断中进行 ADC\_CHNx 的配置和修改，因为进入 ADC 中断后说明 ADC 此时已完成一段采样且处于空闲状态。而在主程序中，无法确认 ADC 运行状态，因此在主程序中如需修改 ADC\_CHNx 和 ADC\_CHNT 等寄存器，需要先关闭 ADC 触发，并向 ADC\_CFG[11]写入 1,以复位 ADC 接口电路状态机，确保 ADC 不在工作状态。如果 ADC 在运行中配置发生变化会发生不可预判的行为。

示例程序如下

```
ADCx_CFG_temp = ADCx_CFG;
```

```
ADCx_CFG = 0x0000;
```

```
ADCx_CFG = 0x0800;
```

```
/*
```

Add your code below, like:

```
ADCx_CHNT = 0x0005
```

```
ADCx_CHN0 = 0x3210;
```

```
ADCx_CHN1 = 0x7654;
```

```
*/
```

```
ADCx_CFG = ADCx_CFG_temp;
```

### 9.3.4 选择对应的模拟通道

ADC 所采用信号对应的通道，请查阅 DATASHEET 中表 2.2 引脚功能选择。关闭对应 IO 的 IE 和 OE，即可使用其模拟功能。



## 10 UTimer 通用定时器

### 10.1 概述

#### 10.1.1 功能框图

通用定时器 UTIMER 主要包括 4 个独立的 Timer，分别可以独立配置运行计数时钟和滤波常数。每个 Timer 可以用于输出特定周期占空比的波形，也可以捕获外部波形进行周期占空比的检测。

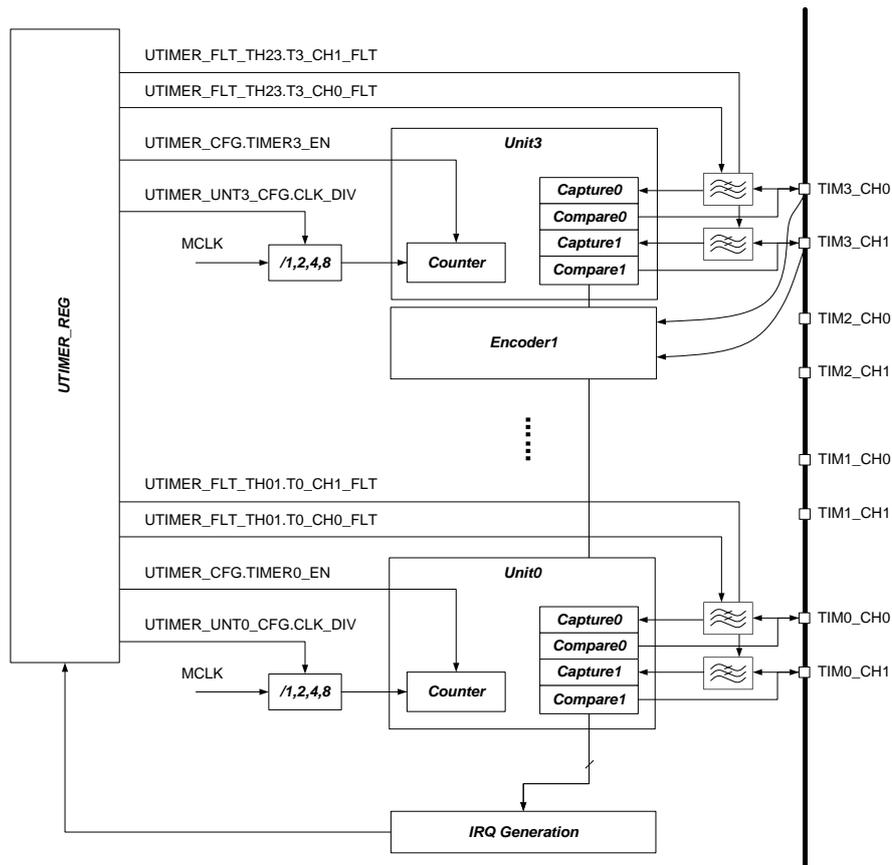


图 10-1 模块顶层功能框图

#### 10.1.1.1 总线接口模块

总线接口模块包括:

将来自 AHB 总线的访问信号翻译为寄存器读写信号，控制寄存器模块的时钟，并对寄存器模块发起读写。

CG 时钟门控模块，在 AHB 总线无访问时，将寄存器模块时钟关闭以降低功耗。



### 10.1.1.2 寄存器模块

`utimer_reg` 寄存器模块，实现对各个子模块控制寄存器的读写。  
对各个子模块状态、结果寄存器的访问。  
对各个子模块中断信号的处理和中断产生。

### 10.1.1.3 IO 滤波模块

IO 滤波模块对来自芯片外部的输入信号进行滤波，降低毛刺对定时器功能的影响。

### 10.1.1.4 通用定时器模块

`utimer_unt` 模块实现了通用的定时器功能，包括比较和捕获工作模式，可以处理两个外部输入信号或者产生两个脉冲信号送到芯片外部。定时器模块中一共包括 4 个独立工作的通用定时器，`Timer0/Unit0`、`Timer1/Unit1` 为 16bit 位宽，`Timer2/Unit2`、`Timer3/Unit3` 为 32bit 位宽。每个定时器包含两个通道。

`utimer_unt` 模块，支持外部事件触发开始计数。外部事件源头可配。当外部事件触发后，`utimer_unt` 定时器开始自增。

### 10.1.1.5 时钟分频模块

时钟分频模块用于产生时钟分频的各种信号。

## 10.1.2 功能特点

定时器模块有以下特点：

- 独立工作，可工作在不同频率下
- `Timer0` 和 `Timer1` 为 16bit 通用定时器
- `Timer2` 和 `Timer3` 为 32bit 通用定时器
- 每个通用定时器处理 2 个外部输入信号（捕获模式），或者产生 2 个输出信号（比较模式）
- 对每个输入信号可以进行最大 120 个系统主时钟的滤波，即，当芯片工作在 96MHz 时钟频率下时，可以滤除 1.25uS 宽度以下毛刺

## 10.2 实现说明

### 10.2.1 时钟分频

为了实现各个 `timer` 独立分频，且可以方便对中断/计数值进行写操作，采取了各个 `timer` 均在系统主频，但使用分频计数器来降低计数器计数频率。



## 10.2.2 中断标志清零

采用了通过对每个中断标志位写 1 来清除标志位的设计。

## 10.2.3 滤波

定时器模块共有 8 个/4 对通道输入，定时器可以对每个输入进行不同程度的滤波。

通过配置滤波寄存器可以调整滤波宽度，0~120 个系统时钟宽度。

原始输入信号在 t1~t6 几个时刻发生了翻转，滤波器宽度配置成 T。可以看到只有 t3 和 t6 时刻发生的翻转维持了大于 T 的时间，因此从滤波器的输出看，信号仅发生了两次翻转。。

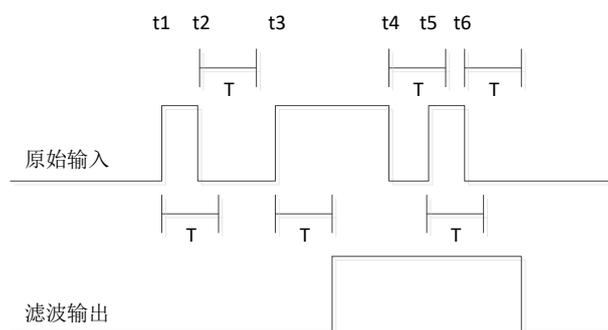


图 10-2 滤波示意图

## 10.2.4 模式

### 10.2.4.1 计数器

Timer 中的计数器采用递增方向计数。

计数器从 0 计数到 TH 值，再回到 0 重新开始计数，计数器回到 0 时，产生回零中断。实际计数周期为  $\text{clk\_freq} * (\text{TH} + 1)$ 。

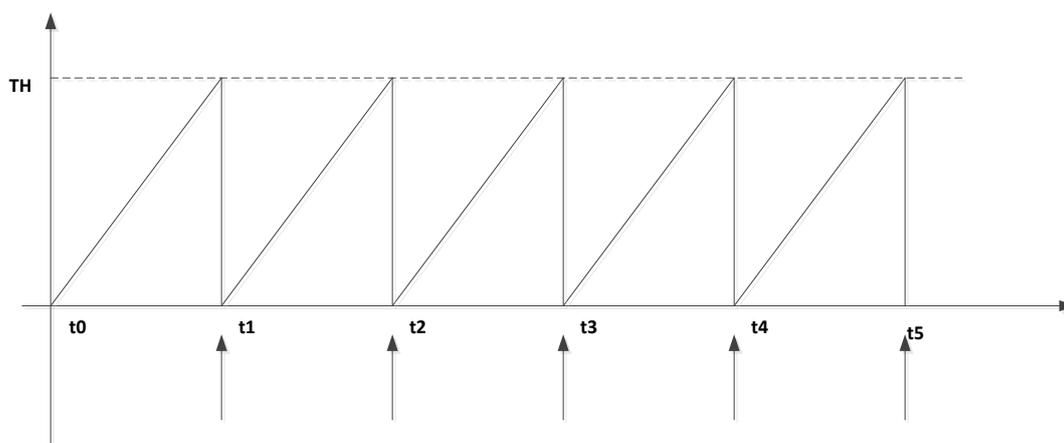


图 10-3 通用计数器

### 10.2.4.2 比较模式

比较模式下，计数器计数到 **UTIMER\_UNTx\_CMP** 值时，产生比较中断。比较模式可以驱动一个比较脉冲发生，在回零时，向 IO 口输出一个电平（极性可配置），在比较事件发生时，电平翻转，向 IO 口输出另一个电平。计数器回零时，仍然会产生回零中断。设置 **UTIMER\_UNTx\_CMP0=0**，可使得 **Timer X 通道 0** 为恒 1，设置 **UTIMER\_UNTx\_CMP0=UTIMER\_UNTx\_TH+1**，可使得 **Timer X 通道 0** 为恒 0。

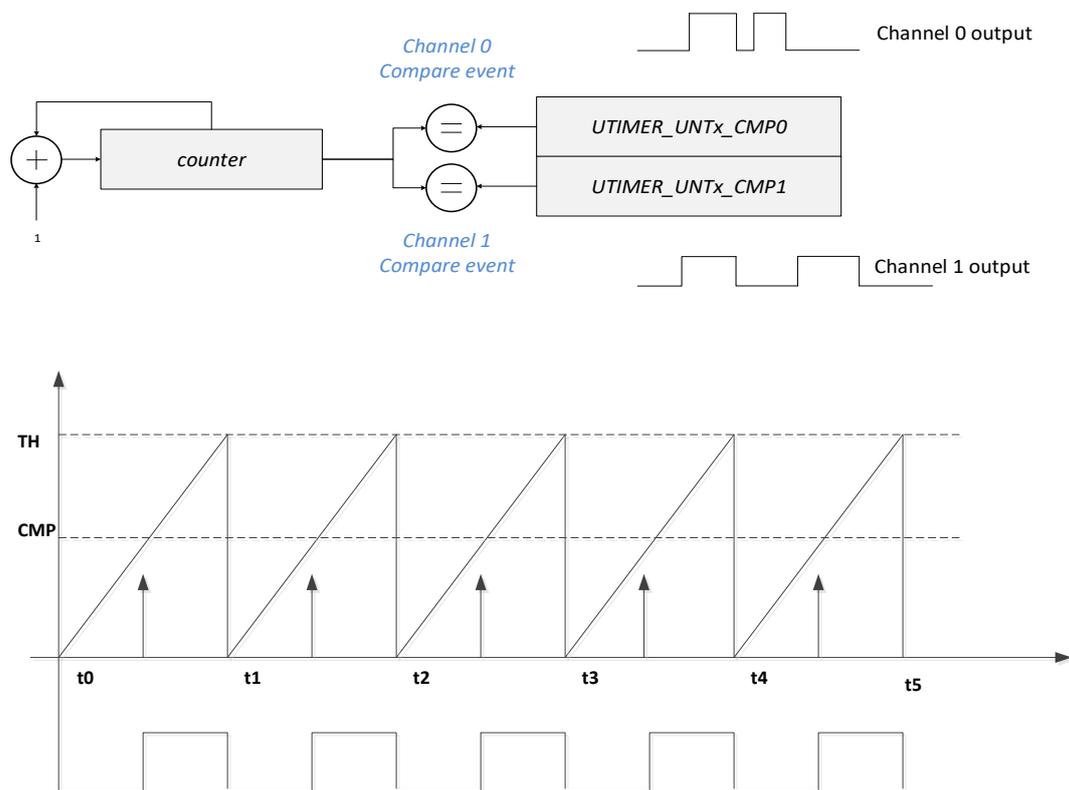
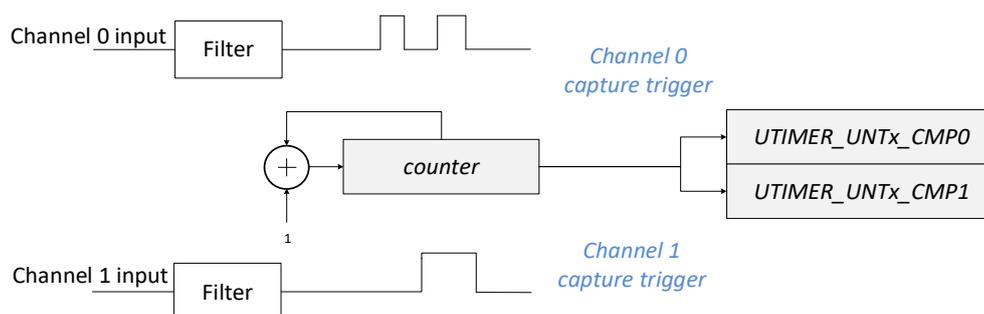


图 10-4 比较模式

### 10.2.4.3 捕获模式

捕获模式下，可以使用 **Timer** 来检测输入信号的上升/下降或者双沿，发生捕获事件（即输入信号电平变化）时，定时器计数值存入 **UTIMER\_UNTx\_CMP** 寄存器，并产生捕获中断。计数器回零时，仍然会产生回零中断。



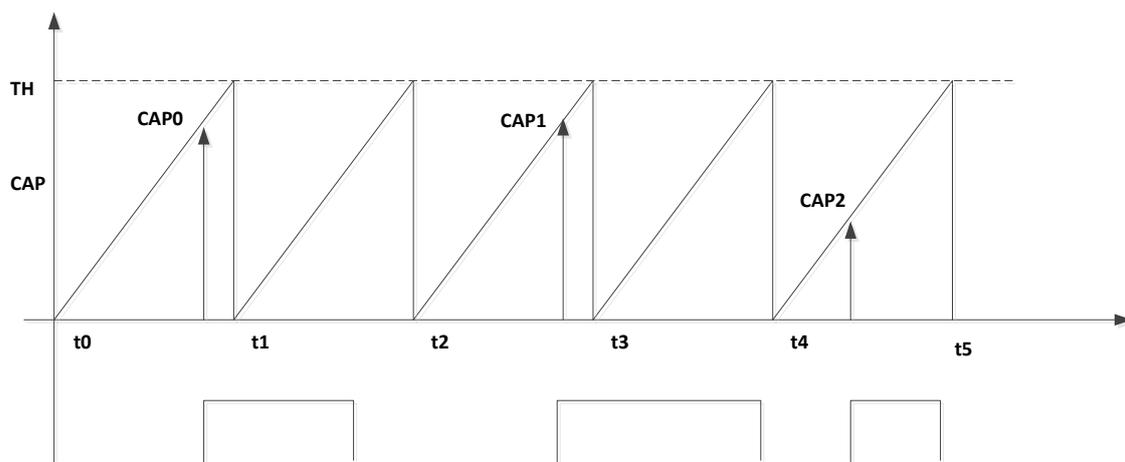


图 10-5 捕获模式

如图 10-5 所示，定时器设置为上升沿捕获。在 CAP0/CAP1/CAP2 三个时刻点，捕获到输入信号发生上升沿变化，对应时刻点的定时器计数值将存入 UTIMER\_UNTx\_CMP 寄存器中。

## 10.3 寄存器

### 10.3.1 地址分配

通用定时器模块的基地址是 0x40011800。

表 10-1 通用定时器配置寄存器地址分配

名称	偏移	描述
UTIMER_UNT0_CFG	0x00	Timer0 配置寄存器
UTIMER_UNT0_TH	0x04	Timer0 计数门限寄存器
UTIMER_UNT0_CNT	0x08	Timer0 计数值寄存器
UTIMER_UNT0_CMP0	0x0C	Timer0 比较/捕获寄存器 0
UTIMER_UNT0_CMP1	0x10	Timer0 比较/捕获寄存器 1
UTIMER_UNT0_EVT	0x14	Timer0 外部事件选择寄存器
UTIMER_UNT1_CFG	0x20	Timer1 配置寄存器
UTIMER_UNT1_TH	0x24	Timer1 计数门限寄存器
UTIMER_UNT1_CNT	0x28	Timer1 计数值寄存器
UTIMER_UNT1_CMP0	0x2C	Timer1 比较/捕获寄存器 0
UTIMER_UNT1_CMP1	0x30	Timer1 比较/捕获寄存器 1
UTIMER_UNT1_EVT	0x34	Timer1 外部事件选择寄存器
UTIMER_UNT2_CFG	0x40	Timer2 配置寄存器
UTIMER_UNT2_TH	0x44	Timer2 计数门限寄存器
UTIMER_UNT2_CNT	0x48	Timer2 计数值寄存器
UTIMER_UNT2_CMP0	0x4C	Timer2 比较/捕获寄存器 0
UTIMER_UNT2_CMP1	0x50	Timer2 比较/捕获寄存器 1



UTIMER_UNT2_EVT	0x54	Timer2 外部事件选择寄存器
UTIMER_UNT3_CFG	0x60	Timer3 配置寄存器
UTIMER_UNT3_TH	0x64	Timer3 计数门限寄存器
UTIMER_UNT3_CNT	0x68	Timer3 计数值寄存器
UTIMER_UNT3_CMP0	0x6C	Timer3 比较/捕获寄存器 0
UTIMER_UNT3_CMP1	0x70	Timer3 比较/捕获寄存器 1
UTIMER_UNT3_EVT	0x74	Timer3 外部事件选择寄存器
UTIMER_FLT_TH01	0xA0	滤波门限寄存器 01
UTIMER_FLT_TH23	0xA4	滤波门限寄存器 23
UTIMER_CFG	0xF0	通用定时器配置寄存器
UTIMER_IE	0xF4	中断使能寄存器
UTIMER_IF	0xF8	中断标志寄存器

### 10.3.2 系统控制寄存器

#### 10.3.2.1 UTIMER\_CFG

地址：0x4001\_18F0

复位值：0x0

表 10-2 UTIMER\_CFG UTIMER 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TIMER3_EN	TIMER2_EN	TIMER1_EN	TIMER0_EN				
								RW	RW	RW	RW				
								0	0	0	0				

位置	位名称	说明
[31:8]		未使用
[7]	TIMER3_EN	Timer3 使能, 当 TIMER3_EN 为 0 时, timer3 停止计数, 同时所有中断标志位输出 0。
[6]	TIMER2_EN	Timer2 使能, 当 TIMER2_EN 为 0 时, timer2 停止计数, 同时所有中断标志位输出 0。
[5]	TIMER1_EN	Timer1 使能, 当 TIMER1_EN 为 0 时, timer1 停止计数, 同时所有中断标志位输出 0。
[4]	TIMER0_EN	Timer0 使能, 当 TIMER0_EN 为 0 时, timer0 停止计数, 同时所有中断标志位输出 0。
[3:0]	RESERVED	系统保留, 推荐写入 0



## 10.3.3 滤波控制寄存器

## 10.3.3.1 UTIMER\_FLT\_TH01

地址：0x4001\_18A0

复位值：0x0

表 10-3 UTIMER\_FLT\_TH01 滤波控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T1_CH1_FLT				T1_CH0_FLT				T0_CH1_FLT				T0_CH0_FLT			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	T1_CH1_FLT	TIM1_CH1 信号滤波宽度选择，取值范围 0~15。 [15:12]为 0 时，对 TIM1_CH1 不进行滤波。 [15:12]不为 0 时，对 TIM1_CH1 信号进行滤波：滤波宽度为 T1_CH1_FLT×8。当 TIM1_CH1 电平稳定超过 T1_CH1_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM1_CH1 信号值；否则，滤波器保持当前的输出不变。
[11:8]	T1_CH0_FLT	TIM1_CH0 信号滤波宽度选择。取值范围 0~15。 [11:08]为 0 时，对 TIM1_CH0 不进行滤波。 [11:08]不为 0 时，对 TIM1_CH0 信号进行滤波：滤波宽度为 T1_CH0_FLT×8。当 TIM1_CH0 电平稳定超过 T1_CH0_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM1_CH0 信号值；否则，滤波器保持当前的输出不变。
[7:4]	T0_CH1_FLT	TIM0_CH1 信号滤波宽度选择。取值范围 0~15。 [07:04]为 0 时，对 TIM0_CH1 不进行滤波。 [07:04]不为 0 时，对 TIM0_CH1 信号进行滤波：滤波宽度为 T0_CH1_FLT×8。当 TIM0_CH1 电平稳定超过 T0_CH1_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM0_CH1 信号值；否则，滤波器保持当前的输出不变。
[3:0]	T0_CH0_FLT	TIM0_CH0 信号滤波宽度选择。取值范围 0~15。 [03:00]为 0 时，对 TIM0_CH0 不进行滤波。 [03:00]不为 0 时，对 TIM0_CH0 信号进行滤波：滤波宽度为 T0_CH0_FLT×8。当 TIM0_CH0 电平稳定超过 T0_CH0_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM0_CH0 信号值；否则，滤波器保持当前的输出不变。

## 10.3.3.2 UTIMER\_FLT\_TH23

地址：0x4001\_18A4



复位值：0x0

表 10-4 UTIMER\_FLT\_TH23 滤波控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T3_CH1_FLT				T3_CH0_FLT				T2_CH1_FLT				T2_CH0_FLT			
RW				RW				RW				RW			
0				0				0				0			

位置	位名称	说明
[31:16]		未使用
[15:12]	T3_CH1_FLT	TIM3_CH1 信号滤波宽度选择，取值范围 0~15。 [15:12]为 0 时，对 TIM3_CH1 不进行滤波。 [15:12]不为 0 时，对 TIM3_CH1 信号进行滤波：滤波宽度为 T3_CH1_FLT×8。当 TIM3_CH1 电平稳定超过 T3_CH1_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM3_CH1 信号值；否则，滤波器保持当前的输出不变。
[11:8]	T3_CH0_FLT	TIM3_CH0 信号滤波宽度选择。取值范围 0~15。 [11:08]为 0 时，对 TIM3_CH0 不进行滤波。 [11:08]不为 0 时，对 TIM3_CH0 信号进行滤波：滤波宽度为 T3_CH0_FLT×8。当 TIM3_CH0 电平稳定超过 T3_CH0_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM3_CH0 信号值；否则，滤波器保持当前的输出不变。
[7:4]	T2_CH1_FLT	TIM2_CH1 信号滤波宽度选择。取值范围 0~15。 [07:04]为 0 时，对 TIM2_CH1 不进行滤波。 [07:04]不为 0 时，对 TIM2_CH1 信号进行滤波：滤波宽度为 T2_CH1_FLT×8。当 TIM2_CH1 电平稳定超过 T2_CH1_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM2_CH1 信号值；否则，滤波器保持当前的输出不变。
[3:0]	T2_CH0_FLT	TIM2_CH0 信号滤波宽度选择。取值范围 0~15。 [03:00]为 0 时，对 TIM2_CH0 不进行滤波。 [03:00]不为 0 时，对 TIM2_CH0 信号进行滤波：滤波宽度为 T2_CH0_FLT×8。当 TIM2_CH0 电平稳定超过 T2_CH0_FLT×8 个系统时钟周期宽度时，滤波器输出更新到 TIM2_CH0 信号值；否则，滤波器保持当前的输出不变。

注意，以上滤波器的工作时钟与对应的 Timer 的分频后的工作时钟为相同时钟，受 UTIMER\_UNTx\_CFG[9:8]的分频系数控制。

假设 UTIMER\_FLT\_TH23[11:8] = 0x6；UTIMER\_UNT3\_CFG[9:8] = 0x2；则 Timer3 的运行时钟相对系统时钟进行了 4 分频。Timer3 的通道 0 输入信号需要经过 8×6 倍 Timer3 的运行时钟的滤波，或者说 8×6×4 倍系统时钟的滤波。



### 10.3.4 定时器寄存器

Timer0/1 完全相同，此处仅说明 Timer0 寄存器。

Timer2/3 完全相同，与 Timer0/1 不同之处在于 Timer2/3 计数器相关寄存器为 32 位宽，而 Timer0/1 计数器相关寄存器为 16 位宽。

#### 10.3.4.1 UTIMER\_UNT0\_CFG Timer0 配置寄存器

地址：0x4001\_1800

复位值：0x0

表 10-5 UTIMER\_UNT0\_CFG Timer 0 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV	CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN	
	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:15]		未使用
[14]	ONE_TRIG	在比较模式下，且 UTIMER_CFG[4]为 0 时，写 1 触发 Timer0 发送一个周期的特定占空比的脉冲，此位在脉冲发送期间内为 1，一个 Timer 周期后，自动清零。
[13]	SRC1	Timer0 捕获模式通道 1 信号来源。默认为 0。 0: 芯片 GPIO (参见 Datasheet 及应用配置) 1: 比较器 1 的输出
[12]	SRC0	Timer0 捕获模式通道 0 信号来源。默认为 0。 0: 芯片 GPIO (参见 Datasheet 及应用配置) 1: 比较器 0 的输出
[11]	ETON	Timer0 计数器计数使能配置。默认为 0。 0, 自动运行，持续计数; 1, 等待外部事件触发计数一个 Timer 周期后停止。外部事件通过 UTIMER_UNT0_EVT 寄存器进行配置。
[10]		未使用
[9:8]	CLK_DIV	Timer0 计数器频率配置，计数器计数频率是系统主时钟频率的 1/2/4/8 分频。默认值为 0，不分频。 00: 1 分频， 01: 2 分频， 10: 4 分频， 11: 8 分频
[7]	CH1_POL	Timer0 通道 1 在比较模式下的输出极性控制，当计数器计数值回零时的输出值。
[6]	CH1_MODE	Timer0 通道 1 的工作模式选择，默认值为 0。 0: 比较模式。输出方波，在 Timer0 通道 1 计数器计数值等于 0 或



		等于 Timer0 比较捕获寄存器值时，IO 发生翻转。 1: 捕获模式。当 Timer0 通道 1 输入信号发生捕获事件时，将计数器计数值存入 Timer0 通道 1 比较捕获寄存器。
[5]	CH1_FE_CAP_EN	Timer0 通道 1 下降沿捕获事件使能。1: 使能; 0: 关闭。 Timer0 通道 1 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。
[4]	CH1_RE_CAP_EN	Timer0 通道 1 上升沿捕获事件使能。1: 使能; 0: 关闭。 Timer0 通道 1 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。
[3]	CH0_POL	Timer0 通道 0 在比较模式下的输出极性控制: 当计数器计数值回零时的输出值。
[2]	CH0_MODE	Timer0 通道 0 的工作模式选择, 默认值为 0。 0: 比较模式。输出方波, 在 Timer0 通道 0 计数器计数值等于 0 或等于 Timer0 比较捕获寄存器值时, IO 发生翻转。 1: 捕获模式。当 Timer0 通道 0 输入信号发生捕获事件时, 将计数器计数值存入 Timer0 通道 0 比较捕获寄存器。
[1]	CH0_FE_CAP_EN	Timer0 通道 0 下降沿捕获事件使能。1: 使能; 0: 关闭。 Timer0 通道 0 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。
[0]	CH0_RE_CAP_EN	Timer0 通道 0 上升沿捕获事件使能。1: 使能; 0: 关闭。 Timer0 通道 0 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。

10.3.4.2 UTIMER\_UNT1\_CFG Timer1 配置寄存器

地址: 0x4001\_1820

复位值: 0x0

表 10-6 UTIMER\_UNT1\_CFG Timer 1 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV	CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN	
	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:15]		未使用
[14]	ONE_TRIG	在比较模式下, 且 UTIMER_CFG[5]为 0 时, 写 1 触发 Timer1 发送一个周期的特定占空比的脉冲, 此位在脉冲发送期间内为 1, 一个 Timer 周期后, 自动清零。
[13]	SRC1	Timer1 捕获模式通道 1 信号来源。默认为 0。



		0: 芯片 GPIO (参见 Datasheet 及应用配置) 1: 比较器 1 的输出
[12]	SRC0	Timer1 捕获模式通道 0 信号来源。默认为 0。 0: 芯片 GPIO (参见 Datasheet 及应用配置) 1: 比较器 0 的输出
[11]	ETON	Timer1 计数器计数使能配置。默认为 0。 0, 自动运行, 持续计数; 1, 等待外部事件触发计数一个 Timer 周期后停止。外部事件通过 UTIMER_UNT1_EVT 寄存器进行配置。
[10]		未使用
[9:8]	CLK_DIV	Timer1 计数器频率配置, 计数器计数频率是系统主时钟频率的 1/2/4/8 分频。默认值为 0, 不分频。 00: 1 分频, 01: 2 分频, 10: 4 分频, 11: 8 分频
[7]	CH1_POL	Timer1 通道 1 在比较模式下的输出极性控制, 当计数器计数值回零时的输出值。
[6]	CH1_MODE	Timer1 通道 1 的工作模式选择, 默认值为 0。 0: 比较模式。输出方波, 在 Timer1 通道 1 计数器计数值等于 0 或等于 Timer1 比较捕获寄存器值时, IO 发生翻转。 1: 捕获模式。当 Timer1 通道 1 输入信号发生捕获事件时, 将计数器计数值存入 Timer1 通道 1 比较捕获寄存器。
[5]	CH1_FE_CAP_EN	Timer1 通道 1 下降沿捕获事件使能。1: 使能; 0: 关闭。 Timer1 通道 1 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。
[4]	CH1_RE_CAP_EN	Timer1 通道 1 上升沿捕获事件使能。1: 使能; 0: 关闭。 Timer1 通道 1 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。
[3]	CH0_POL	Timer1 通道 0 在比较模式下的输出极性控制: 当计数器计数值回零时的输出值。
[2]	CH0_MODE	Timer1 通道 0 的工作模式选择, 默认值为 0。 0: 比较模式。输出方波, 在 Timer1 通道 0 计数器计数值等于 0 或等于 Timer1 比较捕获寄存器值时, IO 发生翻转。 1: 捕获模式。当 Timer1 通道 0 输入信号发生捕获事件时, 将计数器计数值存入 Timer1 通道 0 比较捕获寄存器。
[1]	CH0_FE_CAP_EN	Timer1 通道 0 下降沿捕获事件使能。1: 使能; 0: 关闭。 Timer1 通道 0 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。
[0]	CH0_RE_CAP_EN	Timer1 通道 0 上升沿捕获事件使能。1: 使能; 0: 关闭。 Timer1 通道 0 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。

#### 10.3.4.3 UTIMER\_UNT2\_CFG Timer2 配置寄存器

地址: 0x4001\_1840



复位值: 0x0

表 10-7 UTIMER\_UNT2\_CFG Timer 2 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV	CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN	
	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:15]		未使用
[14]	ONE_TRIG	在比较模式下, 且 UTIMER_CFG[6]为 0 时, 写 1 触发 Timer2 发送一个周期的特定占空比的脉冲, 此位在脉冲发送期间内为 1, 一个 Timer 周期后, 自动清零。
[13]	SRC1	Timer2 捕获模式通道 1 信号来源。默认为 0。 0: 芯片 GPIO (参见 Datasheet 及应用配置) 1: 比较器 1 的输出
[12]	SRC0	Timer2 捕获模式通道 0 信号来源。默认为 0。 0: 芯片 GPIO (参见 Datasheet 及应用配置) 1: 比较器 0 的输出
[11]	ETON	Timer2 计数器计数使能配置。默认为 0。 0, 自动运行, 持续计数; 1, 等待外部事件触发计数一个 Timer 周期后停止。外部事件通过 UTIMER_UNT2_EVT 寄存器进行配置。
[10]		未使用
[9:8]	CLK_DIV	Timer2 计数器频率配置, 计数器计数频率是系统主时钟频率的 1/2/4/8 分频。默认值为 0, 不分频。 00: 1 分频, 01: 2 分频, 10: 4 分频, 11: 8 分频
[7]	CH1_POL	Timer2 通道 1 在比较模式下的输出极性控制, 当计数器计数值回零时的输出值。
[6]	CH1_MODE	Timer2 通道 1 的工作模式选择, 默认值为 0。 0: 比较模式。输出方波, 在 Timer2 通道 1 计数器计数值等于 0 或等于 Timer2 比较捕获寄存器值时, IO 发生翻转。 1: 捕获模式。当 Timer2 通道 1 输入信号发生捕获事件时, 将计数器计数值存入 Timer2 通道 1 比较捕获寄存器。
[5]	CH1_FE_CAP_EN	Timer2 通道 1 下降沿捕获事件使能。1: 使能; 0: 关闭。 Timer2 通道 1 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。
[4]	CH1_RE_CAP_EN	Timer2 通道 1 上升沿捕获事件使能。1: 使能; 0: 关闭。 Timer2 通道 1 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。



[3]	CH0_POL	Timer2 通道 0 在比较模式下的输出极性控制：当计数器计数值回零时的输出值。
[2]	CH0_MODE	Timer2 通道 0 的工作模式选择，默认值为 0。 0: 比较模式。输出方波，在 Timer2 通道 0 计数器计数值等于 0 或等于 Timer2 比较捕获寄存器值时，IO 发生翻转。 1: 捕获模式。当 Timer2 通道 0 输入信号发生捕获事件时，将计数器计数值存入 Timer2 通道 0 比较捕获寄存器。
[1]	CH0_FE_CAP_EN	Timer2 通道 0 下降沿捕获事件使能。1: 使能；0: 关闭。 Timer2 通道 0 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。
[0]	CH0_RE_CAP_EN	Timer2 通道 0 上升沿捕获事件使能。1: 使能；0: 关闭。 Timer2 通道 0 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。

10.3.4.4 UTIMER\_UNT3\_CFG Timer3 配置寄存器

地址：0x4001\_1860

复位值：0x0

表 10-8 UTIMER\_UNT3\_CFG Timer 3 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV	CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN	
	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:15]		未使用
[14]	ONE_TRIG	在比较模式下，且 UTIMER_CFG[7]为 0 时，写 1 触发 Timer3 发送一个周期的特定占空比的脉冲，此位在脉冲发送期间内为 1，一个 Timer 周期后，自动清零。
[13]	SRC1	Timer3 捕获模式通道 1 信号来源。默认为 0。 0: 芯片 GPIO（参见 Datasheet 及应用配置） 1: 比较器 1 的输出
[12]	SRC0	Timer3 捕获模式通道 0 信号来源。默认为 0。 0: 芯片 GPIO（参见 Datasheet 及应用配置） 1: 比较器 0 的输出
[11]	ETON	Timer3 计数器计数使能配置。默认为 0。 0, 自动运行，持续计数； 1, 等待外部事件触发计数一个 Timer 周期后停止。外部事件通过 UTIMER_UNT3_EVT 寄存器进行配置。



[10]		未使用
[9:8]	CLK_DIV	Timer3 计数器频率配置，计数器计数频率是系统主时钟频率的 1/2/4/8 分频。默认值为 0，不分频。 00: 1 分频， 01: 2 分频， 10: 4 分频， 11: 8 分频
[7]	CH1_POL	Timer3 通道 1 在比较模式下的输出极性控制，当计数器计数值回零时的输出值。
[6]	CH1_MODE	Timer3 通道 1 的工作模式选择，默认值为 0。 0: 比较模式。输出方波，在 Timer3 通道 1 计数器计数值等于 0 或等于 Timer3 比较捕获寄存器值时，IO 发生翻转。 1: 捕获模式。当 Timer3 通道 1 输入信号发生捕获事件时，将计数器计数值存入 Timer3 通道 1 比较捕获寄存器。
[5]	CH1_FE_CAP_EN	Timer3 通道 1 下降沿捕获事件使能。1: 使能; 0: 关闭。 Timer3 通道 1 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。
[4]	CH1_RE_CAP_EN	Timer3 通道 1 上升沿捕获事件使能。1: 使能; 0: 关闭。 Timer3 通道 1 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。
[3]	CH0_POL	Timer3 通道 0 在比较模式下的输出极性控制：当计数器计数值回零时的输出值。
[2]	CH0_MODE	Timer3 通道 0 的工作模式选择，默认值为 0。 0: 比较模式。输出方波，在 Timer3 通道 0 计数器计数值等于 0 或等于 Timer3 比较捕获寄存器值时，IO 发生翻转。 1: 捕获模式。当 Timer3 通道 0 输入信号发生捕获事件时，将计数器计数值存入 Timer3 通道 0 比较捕获寄存器。
[1]	CH0_FE_CAP_EN	Timer3 通道 0 下降沿捕获事件使能。1: 使能; 0: 关闭。 Timer3 通道 0 输入信号发生 1→0 跳变被视为捕获事件。下降沿事件使能可以与上升沿事件使能并存。
[0]	CH0_RE_CAP_EN	Timer3 通道 0 上升沿捕获事件使能。1: 使能; 0: 关闭。 Timer3 通道 0 输入信号发生 0→1 跳变被视为捕获事件。上升沿事件使能可以与下降沿事件使能并存。

在 LKS32MC05x 系列中，Timer0/1 为 16bit 位宽；Timer2/3 为 32bit 位宽，因此 Timer2/3 下列相关寄存器为 32bit 位宽。

#### 10.3.4.5 UTIMER\_UNT0\_TH Timer 0 门限寄存器

地址：0x4001\_1804

复位值：0x0

表 10-9 UTIMER\_UNT0\_TH Timer 0 门限寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT0_TH															
RW															
0															



位置	位名称	说明
[31:16]		未使用
[15:0]	UNT0_TH	Timer 0 计数器计数门限。计数器从 0 计数到 UTIMER_UNT0_TH 值后再次回 0 开始计数。

#### 10.3.4.6 UTIMER\_UNT1\_TH Timer 1 门限寄存器

地址: 0x4001\_1824

复位值: 0x0

表 10-10 UTIMER\_UNT1\_TH Timer 1 门限寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT1_TH															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	UNT1_TH	Timer 1 计数器计数门限。计数器从 0 计数到 UTIMER_UNT1_TH 值后再次回 0 开始计数。

#### 10.3.4.7 UTIMER\_UNT2\_TH Timer 2 门限寄存器

地址: 0x4001\_1844

复位值: 0x0

表 10-11 UTIMER\_UNT2\_TH Timer 2 门限寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT2_TH															
RW															
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT2_TH															
RW															
0															

位置	位名称	说明
[31:0]	UNT2_TH	Timer 2 计数器计数门限。计数器从 0 计数到 UTIMER_UNT2_TH 值后再次回 0 开始计数。



## 10.3.4.8 UTIMER\_UNT3\_TH Timer 3 门限寄存器

地址：0x4001\_1864

复位值：0x0

表 10-12 UTIMER\_UNT3\_TH Timer 3 门限寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT3_TH															
RW															
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT3_TH															
RW															
0															

位置	位名称	说明
[31:0]	UNT3_TH	Timer 3 计数器计数门限。计数器从 0 计数到 UTIMER_UNT3_TH 值后再次回 0 开始计数。

## 10.3.4.9 UTIMER\_UNT0\_CNT Timer 0 计数寄存器

地址：0x4001\_1808

复位值：0x0

表 10-13 UTIMER\_UNT0\_CNT Timer 0 计数寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT0_CNT															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	UNT0_CNT	Timer 0 计数器当前计数值。写操作可以写入新的计数值。

注意：写入 UTIMER\_UNT0\_CNT 前需要通过 SYS\_CLK\_FEN 开启 timer 时钟。

## 10.3.4.10 UTIMER\_UNT1\_CNT Timer 1 计数寄存器

地址：0x4001\_1828

复位值：0x0



表 10-14 UTIMER\_UNT1\_CNT Timer 1 计数寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT1_CNT															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	UNT1_CNT	Timer 1 计数器当前计数值。写操作可以写入新的计数值。

注意：写入 UTIMER\_UNT1\_CNT 前需要通过 SYS\_CLK\_FEN 开启 timer 时钟。

#### 10.3.4.11 UTIMER\_UNT2\_CNT Timer 2 计数寄存器

地址：0x4001\_1848

复位值：0x0

表 10-15 UTIMER\_UNT2\_CNT Timer 2 计数寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT2_CNT															
RW															
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT2_CNT															
RW															
0															

位置	位名称	说明
[31:0]	UNT2_CNT	Timer 2 计数器当前计数值。写操作可以写入新的计数值。

注意：写入 UTIMER\_UNT2\_CNT 前需要通过 SYS\_CLK\_FEN 开启 timer 时钟。

#### 10.3.4.12 UTIMER\_UNT3\_CNT Timer 3 计数寄存器

地址：0x4001\_1868

复位值：0x0

表 10-16 UTIMER\_UNT3\_CNT Timer 3 计数寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT3_CNT															
RW															



0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT3_CNT															
RW															
0															

位置	位名称	说明
[31:0]	UNT3_CNT	Timer3 计数器当前计数值。写操作可以写入新的计数值。

注意：写入 UTIMER\_UNT3\_CNT 前需要通过 SYS\_CLK\_FEN 开启 timer 时钟。

### 10.3.4.13 UTIMER\_UNT0\_CMP0 Timer 0 通道 0 比较捕获寄存器

地址：0x4001\_180C

复位值：0x0

表 10-17 UTIMER\_UNT0\_CMP0 Timer 0 通道 0 比较捕获寄存器

UNT0_CMP0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	UNT0_CMP0	Timer 0 通道 0 工作在比较模式时，当计数器计数值等于 UTIMER_UNT0_CMP0 时，发生比较事件。 Timer 0 通道 0 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER_UNT0_CMP0 寄存器。

### 10.3.4.14 UTIMER\_UNT0\_CMP1 Timer 0 通道 1 比较捕获寄存器

地址：0x4001\_1810

复位值：0x0

表 10-18 UTIMER\_UNT0\_CMP1 Timer 0 通道 1 比较捕获寄存器

UNT0_CMP1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW															
0															

位置	位名称	说明
----	-----	----



[31:16]		未使用
[15:0]	UNT0_CMP1	Timer 0 通道 1 工作在比较模式时，当计数器计数值等于 UTIMER_UNT0_CMP1 时，发生比较事件。 Timer 0 通道 1 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER_UNT0_CMP1 寄存器。

#### 10.3.4.15 UTIMER\_UNT1\_CMP0 Timer 1 通道 0 比较捕获寄存器

地址：0x4001\_182C

复位值：0x0

表 10-19 UTIMER\_UNT1\_CMP0 Timer 1 通道 0 比较捕获寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT1_CMP0															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	UNT1_CMP0	Timer 1 通道 0 工作在比较模式时，当计数器计数值等于 UTIMER_UNT1_CMP0 时，发生比较事件。 Timer 1 通道 0 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER_UNT1_CMP0 寄存器。

#### 10.3.4.16 UTIMER\_UNT1\_CMP1 Timer 1 通道 1 比较捕获寄存器

地址：0x4001\_1830

复位值：0x0

表 10-20 UTIMER\_UNT1\_CMP1 Timer 1 通道 1 比较捕获寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT1_CMP1															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	UNT1_CMP1	Timer 1 通道 1 工作在比较模式时，当计数器计数值等于 UTIMER_UNT1_CMP1 时，发生比较事件。 Timer 1 通道 1 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER_UNT1_CMP1 寄存器。



## 10.3.4.17 UTIMER\_UNT2\_CMP0 Timer 2 通道 0 比较捕获寄存器

地址：0x4001\_184C

复位值：0x0

表 10-21 UTIMER\_UNT2\_CMP0 Timer 2 通道 0 比较捕获寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT2_CMP0															
RW															
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT2_CMP0															
RW															
0															

位置	位名称	说明
[31:0]	UNT2_CMP0	Timer 2 通道 0 工作在比较模式时，当计数器计数值等于 UTIMER_UNT2_CMP0 时，发生比较事件。 Timer 2 通道 0 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER_UNT2_CMP0 寄存器。

## 10.3.4.18 UTIMER\_UNT2\_CMP1 Timer 2 通道 1 比较捕获寄存器

地址：0x4001\_1850

复位值：0x0

表 10-22 UTIMER\_UNT2\_CMP1 Timer 2 通道 1 比较捕获寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT2_CMP1															
RW															
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT2_CMP1															
RW															
0															

位置	位名称	说明
[31:0]	UNT2_CMP1	Timer 2 通道 1 工作在比较模式时，当计数器计数值等于 UTIMER_UNT2_CMP1 时，发生比较事件。



		Timer 2 通道 1 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER_UNT2_CMP1 寄存器。
--	--	--

### 10.3.4.19 UTIMER\_UNT3\_CMP0 Timer 3 通道 0 比较捕获寄存器

地址：0x4001\_186C

复位值：0x0

表 10-23 UTIMER\_UNT3\_CMP0 Timer 3 通道 0 比较捕获寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT3_CMP0															
RW															
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT3_CMP0															
RW															
0															

位置	位名称	说明
[31:0]	UNT3_CMP0	Timer 3 通道 0 工作在比较模式时，当计数器计数值等于 UTIMER_UNT3_CMP0 时，发生比较事件。 Timer 3 通道 0 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER_UNT3_CMP0 寄存器。

### 10.3.4.20 UTIMER\_UNT3\_CMP1 Timer 3 通道 1 比较捕获寄存器

地址：0x4001\_1870

复位值：0x0

表 10-24 UTIMER\_UNT3\_CMP1 Timer 3 通道 1 比较捕获寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT3_CMP1															
RW															
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT3_CMP1															
RW															
0															

位置	位名称	说明
----	-----	----



[31:0]	UNT3_CMP1	Timer 3 通道 1 工作在比较模式时，当计数器计数值等于 UTIMER_UNT3_CMP1 时，发生比较事件。 Timer 3 通道 1 工作在捕获模式时，发生捕获事件时的计数器计数值存入 UTIMER_UNT3_CMP1 寄存器。
--------	-----------	--

## 10.3.4.21 UTIMER\_UNT0\_EVT Timer0 外部事件选择寄存器

地址：0x4001\_1814

复位值：0x0

表 10-25 UTIMER\_UNT0\_EVT Timer 0 外部事件选择寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														EVT_SRC	
														RW	
														0	

位置	位名称	说明
[31:3]		未使用
[2:0]	EVT_SRC	Timer0 外部事件选择寄存器，本寄存器需要配合 UTIMER_UNT0_CFG[11]使用。UOCFG[11]为高时，根据本寄存器选择触发 Timer0 计数的事件。 0: MCPWM TADC[2]比较事件 1: MCPWM TADC[3]比较事件 2: TIMER1 通道 0 比较事件 3: TIMER1 通道 1 比较事件 4: TIMER2 通道 0 比较事件 5: TIMER2 通道 1 比较事件 6: TIMER3 通道 0 比较事件 7: TIMER3 通道 1 比较事件

## 10.3.4.22 UTIMER\_UNT1\_EVT Timer1 外部事件选择寄存器

地址：0x4001\_1834

复位值：0x0

表 10-26 UTIMER\_UNT1\_EVT Timer 1 外部事件选择寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														EVT_SRC	
														RW	
														0	



位置	位名称	说明
[31:3]		未使用
[2:0]	EVT_SRC	<p>Timer1 外部事件选择寄存器，本寄存器需要配合 UTIMER_UNT1_CFG[11]使用。UTIMER_UNT1_CFG[11]为高时，根据本寄存器选择触发 Timer1 计数的事件。</p> <p>0: TIMER0 通道 0 比较事件            1: TIMER0 通道 1 比较事件            2: MCPWM TADC[2]比较事件            3: MCPWM TADC[3]比较事件            4: TIMER2 通道 0 比较事件            5: TIMER2 通道 1 比较事件            6: TIMER3 通道 0 比较事件            7: TIMER3 通道 1 比较事件</p>

## 10.3.4.23 UTIMER\_UNT2\_EVT Timer2 外部事件选择寄存器

地址: 0x4001\_1854

复位值: 0x0

表 10-27 UTIMER\_UNT2\_EVT Timer 2 外部事件选择寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														EVT_SRC	
														RW	
														0	

位置	位名称	说明
[31:3]		未使用
[2:0]	EVT_SRC	<p>Timer2 外部事件选择寄存器，本寄存器需要配合 UTIMER_UNT2_CFG[11]使用。UTIMER_UNT2_CFG[11]为高时，根据本寄存器选择触发 Timer2 计数的事件。</p> <p>0: TIMER0 通道 0 比较事件            1: TIMER0 通道 1 比较事件            2: TIMER1 通道 0 比较事件            3: TIMER1 通道 1 比较事件            4: MCPWM TADC[2]比较事件            5: MCPWM TADC[3]比较事件            6: TIMER3 通道 0 比较事件            7: TIMER3 通道 1 比较事件</p>

## 10.3.4.24 UTIMER\_UNT3\_EVT Timer3 外部事件选择寄存器

地址: 0x4001\_1874



复位值: 0x0

表 10-28 UTIMER\_UNT3\_EVT Timer 3 外部事件选择寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													EVT_SRC		
													RW		
													0		

位置	位名称	说明
[31:3]		未使用
[2:0]	EVT_SRC	<p>Timer3 外部事件选择寄存器，本寄存器需要配合 UTIMER_UNT3_CFG[11]使用。UTIMER_UNT3_CFG[11]为高时，根据本寄存器选择触发 Timer3 计数的事件。</p> <p>0: TIMER0 通道 0 比较事件            1: TIMER0 通道 1 比较事件            2: TIMER1 通道 0 比较事件            3: TIMER1 通道 1 比较事件            4: TIMER2 通道 0 比较事件            5: TIMER2 通道 1 比较事件            6: MCPWM TADC[2]比较事件            7: MCPWM TADC[3]比较事件</p>

### 10.3.5 中断管理寄存器

中断管理寄存器包括中断标志寄存器 UTIMER\_IF 和中断使能寄存器 UTIMER\_IE。两个寄存器各个比特对应相同的中断。

#### 10.3.5.1 UTIMER\_IE 中断使能寄存器

地址: 0x4001\_18F4

复位值: 0x0

表 10-29 UTIMER\_IE 中断使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				T3_CH1_IE	T3_CH0_IE	T3_ZC_IE	T2_CH1_IE	T2_CH0_IE	T2_ZC_IE	T1_CH1_IE	T1_CH0_IE	T1_ZC_IE	T0_CH1_IE	T0_CH0_IE	T0_ZC_IE
				RW	RW	RW									
				0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:12]		未使用



[11]	T3_CH1_IE	Timer3 通道 1 比较/捕获中断使能，高电平有效。
[10]	T3_CH0_IE	Timer3 通道 0 比较/捕获中断使能，高电平有效。
[9]	T3_ZC_IE	Timer3 计数器过 0 中断使能，高电平有效。
[8]	T2_CH1_IE	Timer2 通道 1 比较/捕获中断使能，高电平有效。
[7]	T2_CH0_IE	Timer2 通道 0 比较/捕获中断使能，高电平有效。
[6]	T2_ZC_IE	Timer2 计数器过 0 中断使能，高电平有效。
[5]	T1_CH1_IE	Timer1 通道 1 比较/捕获中断使能，高电平有效。
[4]	T1_CH0_IE	Timer1 通道 0 比较/捕获中断使能，高电平有效。
[3]	T1_ZC_IE	Timer1 计数器过 0 中断使能，高电平有效。
[2]	T0_CH1_IE	Timer0 通道 1 比较/捕获中断使能，高电平有效。
[1]	T0_CH0_IE	Timer0 通道 0 比较/捕获中断使能，高电平有效。
[0]	T0_ZC_IE	Timer0 计数器过 0 中断使能，高电平有效。

## 10.3.5.2 UTIMER\_IF 中断标志寄存器

地址：0x4001\_18F8

复位值：0x0

表 10-30 UTIMER\_IF 中断标志寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				T3_CH1_IF	T3_CH0_IF	T3_ZC_IF	T2_CH1_IF	T2_CH0_IF	T2_ZC_IF	T1_CH1_IF	T1_CH0_IF	T1_ZC_IF	T0_CH1_IF	T0_CH0_IF	T0_ZC_IF
				RW1C	RW1C	RW1C									
				0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:12]		未使用
[11]	T3_CH1_IF	Timer3 通道 1 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[10]	T3_CH0_IF	Timer3 通道 0 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[9]	T3_ZC_IF	Timer3 计数器过 0 中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[8]	T2_CH1_IF	Timer2 通道 1 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。



[7]	T2_CH0_IF	Timer2 通道 0 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[6]	T2_ZC_IF	Timer2 计数器过 0 中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[5]	T1_CH1_IF	Timer1 通道 1 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[4]	T1_CH0_IF	Timer1 通道 0 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[3]	T1_ZC_IF	Timer1 计数器过 0 中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[2]	T0_CH1_IF	Timer0 通道 1 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[1]	T0_CH0_IF	Timer0 通道 0 比较/捕获中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。
[0]	T0_ZC_IF	Timer0 计数器过 0 中断标志。高电平有效，对此 bit 写 1 可清 0 此 bit。

其中 Timer2 通道 0/1、Timer3 通道 0/1 的比较事件可作为 ADC 采样触发事件 UTimer\_T0/ UTimer\_T1/ UTimer\_T2/ UTimer\_T3;

此 4 个事件与 MCPWM 产生的 MCPWM\_T0/ MCPWM\_T1/ MCPWM\_T2/ MCPWM\_T3 在 ADC 内部经过使能控制后分别相或得到 4 个 ADC 采样触发事件 TADC[0]/TADC[1]/TADC[2]/TADC[3]。

中断标志写 1 清零，一般不建议用如下|=方式清零，因为|=是先读取中断标志，将对应位改为 1 再写入清零，如果同时有其他中断标志置位，会被一起清零，而这通常不是软件所期望的。例如，如下写法本意是清零 T0\_ZC\_IF，但如果同时 T0\_CH0\_IF 在写入执行前置 1 了，则软件先读取回 UTIMER\_IF 值为 0x2，然后执行或操作 0x2|0x1=0x3，然后写入，同时对 T0\_CH0\_IF 和 T0\_ZC\_IF 进行了清零，可能导致 Timer 少进入一次因捕获而产生的中断。

*UTIMER\_IF|=0x1;*

如果希望清零 T0\_ZC\_IF 标志位，应以如下方式，直接对 BIT0 写 1.

*UTIMER\_IF=0x1;*

## 11 HALL 信号处理模块

### 11.1 综述

芯片共支持 3 路 HALL 信号输入。

对于输入的 HALL 传感器信号，所进行的处理包括：

滤波，消除 HALL 信号毛刺的影响

捕获，当 HALL 输入有变化时，记录当前的定时器值，并输出中断

溢出，当 HALL 信号长时间不发生变化导致计数器溢出时，输出中断

### 11.2 实现说明

#### 11.2.1 信号来源

HALL 信号来源于 GPIO，对于每一路 HALL 信号，芯片有两个 IO 可以作为该信号的来源。通过配置 GPIO 寄存器，用户可以选择将其中一个 GPIO 的输入信号做为 HALL 信号使用。

详细管脚位置说明见芯片器件 datasheet。

#### 11.2.2 工作时钟

HALL 模块工作频率可调。通过配置 HALL\_CFG.CLK\_DIV 寄存器，可以选择系统主时钟的 1/2/4/8 分频作为 HALL 模块工作频率，滤波和计数均采用该频率工作。

#### 11.2.3 信号滤波

滤波模块主要用于去除 HALL 信号上的毛刺。

滤波包括两级滤波器，两级滤波电路可单独开启，也可全部开启：

第一级采用 7 判 5 进行滤波，即连续 7 个采样点中，如果达到超过 5 个 1 则输出 1，如果达到或超过 5 个 0 则输出 0，否则输出保持上一次的滤波结果。通过配置 HALL\_CFG.FIL\_75 可以选择是否使能第一级滤波器。具体如下图所示：

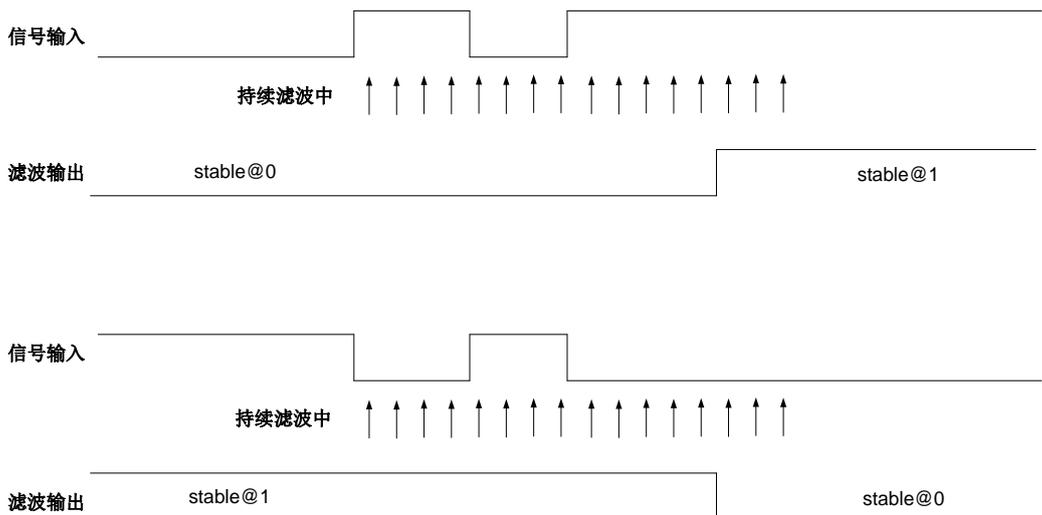


图 11-1 7/5 滤波模块框图

第二级采用连续滤波，在连续 N 个采样点中，如全为 0 则输出 0，如全为 1 则输出 1，否则输出保持上一次的滤波结果。

通过配置 HALL\_CFG.FIL\_LEN 可以配置第二级滤波器滤波深度，即连续采样个数。连续采样个数最大为 2<sup>15</sup>，滤波时间常数计算公式如下：

$$T_{fit} = T_{clk} * (HALL\_CFG.FIL\_LEN[14:0] + 1)$$

举例，在 96MHz 工作频率下，周期 T<sub>clk</sub> 是 10.4ns，寄存器配置最大为 32767，最长滤波宽度为约 10.4ns×32768≈340us。

通过访问 HALL\_INFO.FIL\_DATA[2:0]可以捕捉滤波后的 HALL 信号；HALL\_INFO.RAW\_DATA[2:0]则是滤波前原始 HALL 输入信号，详见 11.3.3。

### 11.2.4 捕获

捕获模块用于测量两次 HALL 信号变化之间的时间，其核心为一个 24 位计数器，在 96MHz 工作频率下，最大可以记录约 1.39 秒的时间宽度，达到 10.4ns 的时间分辨率。

HALL\_CNT 从 0 开始计数，当发生 HALL 信号变化时，将此时此刻的 HALL\_CNT 值保存到 HALL\_WIDTH 寄存器，将此时此刻的 HALL 信号保存到 HALL\_INFO.FIL\_DATA，输出 HALL 信号变化中断，HALL\_CNT 重新从 0 开始计数。

当计数器计数值达到 HALL\_TH 时，输出 HALL 计数器溢出中断，计数器重新从 0 开始计数。

### 11.2.5 中断

捕获、溢出事件触发中断，中断使能控制位位于 HALL\_CFG.CHG\_IE 和 HALL\_CFG.OV\_IE，中断标志位位于 HALL\_INFO.CHG\_IF 和 HALL\_INFO.OV\_IF。终端标志可以通过对 HALL\_INFO.CHG\_IF 和 HALL\_INFO.OV\_IF 写 1 清空。

### 11.2.6 数据流程

HALL 模块的数据流程如下图所示，FCLK[1]为受 SYS\_CLK\_FEN 门控控制的系统主时钟，通常为



96MHz 的 PLL 时钟。

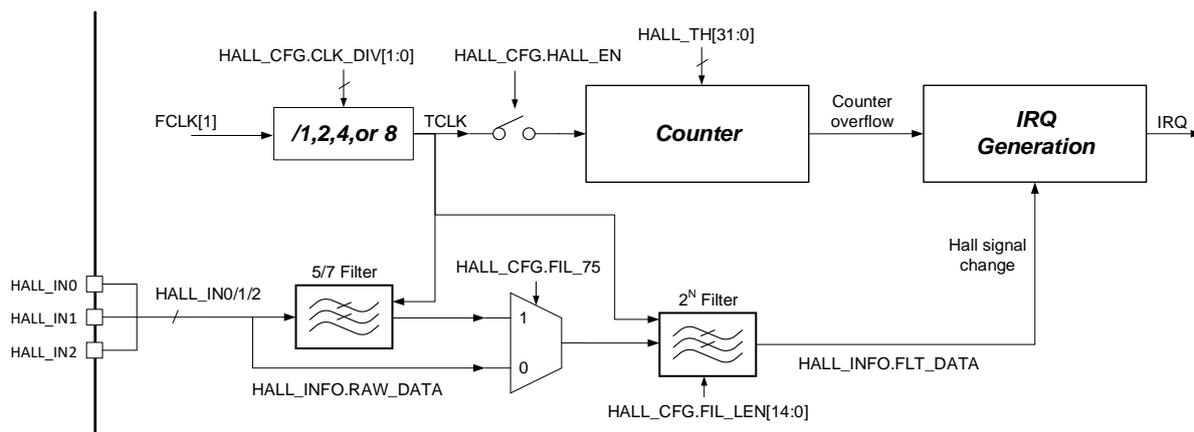


图 11-2 数据流程框图

### 11.3 寄存器

#### 11.3.1 地址分配

HALL 模块寄存器的基地址是 0x40011000。

表 11-1 HALL 模块寄存器地址分配

名称	偏移	描述
HALL_CFG	0x00	HALL 模块配置寄存器
HALL_INFO	0x04	HALL 模块信息寄存器
HALL_WIDTH	0x08	HALL 宽度计数值寄存器
HALL_TH	0x0C	HALL 模块计数器门限值寄存器
HALL_CNT	0x10	HALL 计数寄存器

#### 11.3.2 HALL\_CFG HALL 模块配置寄存器

地址: 0x4001\_1000

复位值: 0x0

表 11-2 HALL\_CFG HALL 模块配置寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SW_IE	OV_IE	CHG_IE			HALL_EN				FIL_75				CLK_DIV	
	RW	RW	RW			RW				RW				RW	
	0	0	0			0				0				0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIL_LEN_														
	RW														



	0
--	---

位置	位名称	说明
[31]		未使用
[30]	SW_IE	软件触发 HALL 信号变化中断使能, 高电平有效。当 SW_IE=1 时, 向 HALL_INFO.SW_IF 写 1, 将手动产生 HALL 信号变化标志, 即令 HALL_INFO.CHG_IF 置 1。
[29]	OV_IE	HALL 计数器溢出中断使能开关。默认关闭。 0: 关闭 1: 使能
[28]	CHG_IE	HALL 信号变化中断使能开关。默认关闭。 0: 关闭 1: 使能
[27:25]		未使用
[24]	HALL_EN	HALL 模块使能开关。默认关闭。 0: 关闭 1: 使能
[23:21]		未使用
[20]	FIL_75	7/5 滤波开关 (连续采样 7 次, 5 次值一致)。默认关闭。 0: 关闭 1: 使能
[19:18]		未使用
[17:16]	CLK_DIV	HALL 时钟分频系数。默认不分频。 00: 不分频 01: 2 分频 10: 4 分频 11: 8 分频
[15]		未使用
[14:0]	FIL_LEN	滤波宽度, 低于对应脉冲宽度的信号将被硬件自动过滤掉。滤波宽度的计算公式为 [14:0] + 1。

### 11.3.3 HALL\_INFO HALL 模块信息寄存器

地址: 0x4001\_1004

复位值: 0x0

表 11-3 HALL\_INFO HALL 模块信息寄存器

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



	RAW_DATA		FLT_DATA
	RO		RO
	0		0

位置	位名称	说明
[31:19]		未使用
[18]	SW_IF	软件触发 HALL 信号变化中断。写 1 触发，自动清零。当 HALL_CFG.SW_IE=1 时，向 SW_IF 写 1 可以触发 CFG_IF 置 1。当 HALL_CFG.CHG_IE=1 时，可以同时产生 HALL 中断。
[17]	OV_IF	HALL 计数器溢出事件标志，写 1 清空
[16]	CHG_IF	HALL 信号变化事件标志，写 1 清空
[15:11]	RESERVED	系统保留，必须写入 0，读出 0
[10:8]	RAW_DATA	HALL 值，未滤波结果
[7:3]	RESERVED	系统保留，必须写入 0，读出 0
[2:0]	FLT_DATA	HALL 值，滤波结果

#### 11.3.4 HALL\_WIDTH HALL 宽度计数值寄存器

地址：0x4001\_1008

复位值：0x0

表 11-4 HALL\_WIDTH HALL 宽度计数值寄存器

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

位置	位名称	说明
[31:24]		未使用
[23:0]	CAP_CNT	HALL 宽度计数器值

#### 11.3.5 HALL\_TH HALL 模块计数器门限值寄存器

地址：0x4001\_100C

复位值：0x0



表 11-5 HALL\_TH HALL 模块计数器门限值寄存器

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

位置	位名称	说明
[31:24]		未使用
[23:0]	TH	HALL 计数器门限值

### 11.3.6 HALL\_CNT HALL 计数寄存器

地址: 0x4001\_1010

复位值: 0x0

表 11-6 HALL\_CNT HALL 计数寄存器

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

位置	位名称	说明
[31:24]		未使用
[23:0]	CNT	HALL 计数值, 写入任意值可清零

## 12 MCPWM

### 12.1 概述

MCPWM 模块，是一个精确控制电机驱动波形输出的模块。

包含一个 16 位递增计数器，用于提供一个基础周期。计数器的时钟分频有 1/2/4/8 四种选项，产生的分频时钟频率分别为 96MHz、48MHz、24MHz 和 12MHz。

包含四组 PWM 生成模块。

- 可以产生 4 对（互补信号）或 8 路独立（边沿模式）不交叠的 PWM 信号；
- 支持边沿对齐 PWM
- 中心对齐 PWM
- 移相 PWM

同时可以产生 4 路与 MCPWM 同时基的定时信息，用于触发 ADC 模块同步采样，进行与 MCPWM 的联动。

包含一组急停保护模块，用于不依赖 MCU 软件的处理快速关断 MCPWM 模块输出。MCPWM 模块可输入 4 路急停信号，其中两路来自芯片 IO，两路来自片内比较器的输出。当急停事件发生时（支持有效电平极性选择），把所有 MCPWM 输出信号复位到规定状态，以避免短路发生。

对急停信号有独立滤波模块。

MCPWM 的每个输出 IO 支持两种控制模式：PWM 硬件控制或者软件直接控制（用于 EABS 软刹车，或 BLDC 方波换相控制）。

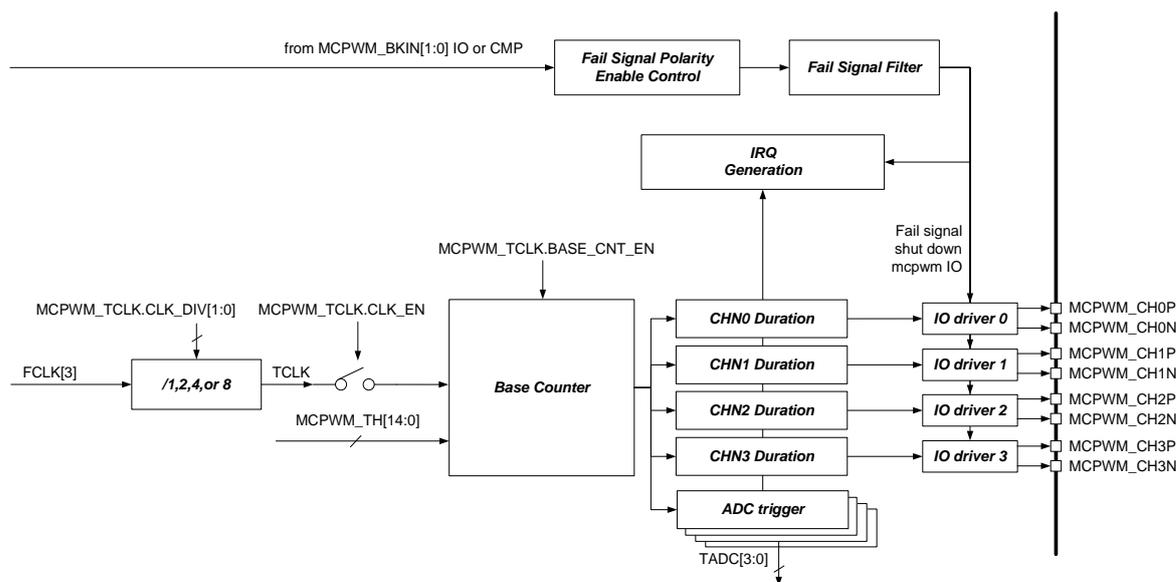


图 12-1 MCPWM 模块框图

为了保证定时精度，通常采用 96MHz 的时钟作为 MCPWM 模块工作频率。

### 12.1.1 Base Counter 模块

该模块主要是由一个递增计数器组成，其计数门限值为 MCPWM\_TH，计数器从 t0 时刻开始从 -TH 递增计数，在 t1 时刻过 0 点，在 t2 时刻计数到 TH 完成一次计数循环，回到 -TH，重新开始计数。计数周期为  $(TH \times 2 + 1) \times$  计数时钟周期。

在 t0/t1(本次 t0 即上一次 t2)可产生定时事件中断，MCPWM\_IF.T0\_IF 和 MCPWM\_IF.T1\_IF 将被置位。

可通过寄存器配置 MCPWM\_TCLK.BASE\_CNT\_EN 控制 Base Counter 的启动和停止。

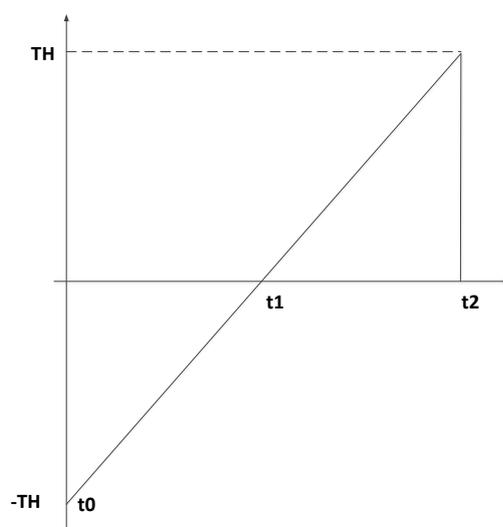


图 12-2 Base Counter t0/t1 时序

在运行 MCPWM 模块前，用户一般需将对应的比较门限值 MCPWM\_TH00~MCPWM\_TH31，死区寄存器 MCPWM\_DTH00~MCPWM\_DTH31 以及 PWM 周期寄存器 MCPWM\_TH 配置好。在实际运行过程中，也可动态改变比较门限值和 PWM 周期值，可通过写 MCPWM\_UPDATE 寄存器实现手动更新，也可以通过配置 MCPWM\_SDCFG.T1\_UPDATE\_EN 及 MCPWM\_SDCFG.T0\_UPDATE\_EN 进行硬件自动更新。硬件更新，仅在 t0 及 t1 时刻（可配置 t0 或 t1 更新和 t0 t1 时刻都更新）才能产生更新事件，硬件把加载寄存器的值载入到实际运行的寄存器中。而更新事件的发生频率可以配置，即每间隔 N 个 t0 及 t1 时刻才发生更新。无论是否发生更新，t0/t1 时刻均可产生相应的中断。若硬件把加载寄存器的值到载入实际运行的寄存器后，产生装载中断。

通过配置 MCPWM\_SDCFG 寄存器选择更新发生在 t0 或者 t1 或者二者皆可，配置更新间隔数，间隔数为 1~16。最频繁的更新配置为更新发生在 t0 和 t1，连续发生。最低速的更新配置为更新发生在 t1，每 16 个 t1 更新一次。

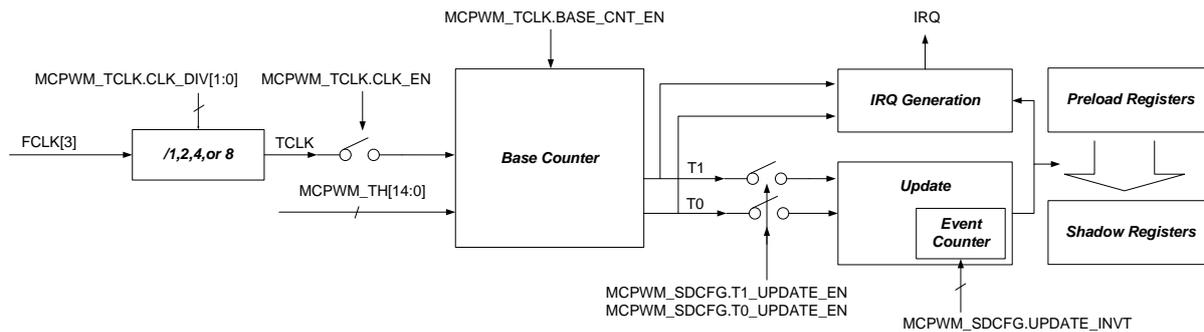


图 12-3 MCPWM 更新机制

### 12.1.2 Fail 信号处理

Fail 信号即急停信号，主要用于在出现异常时迅速关断功率管，以免造成不可逆的硬件损毁。该信号处理模块主要是根据实际情况设置急停事件，实现快速关断 PWM 的输出。有两路 fail 信号输入 MCPWM，即 FAIL0 和 FAIL1，分别可以来自芯片 IO MCPWM\_BKIN[1:0]或芯片内部比较器的输出 CMP[1:0]。

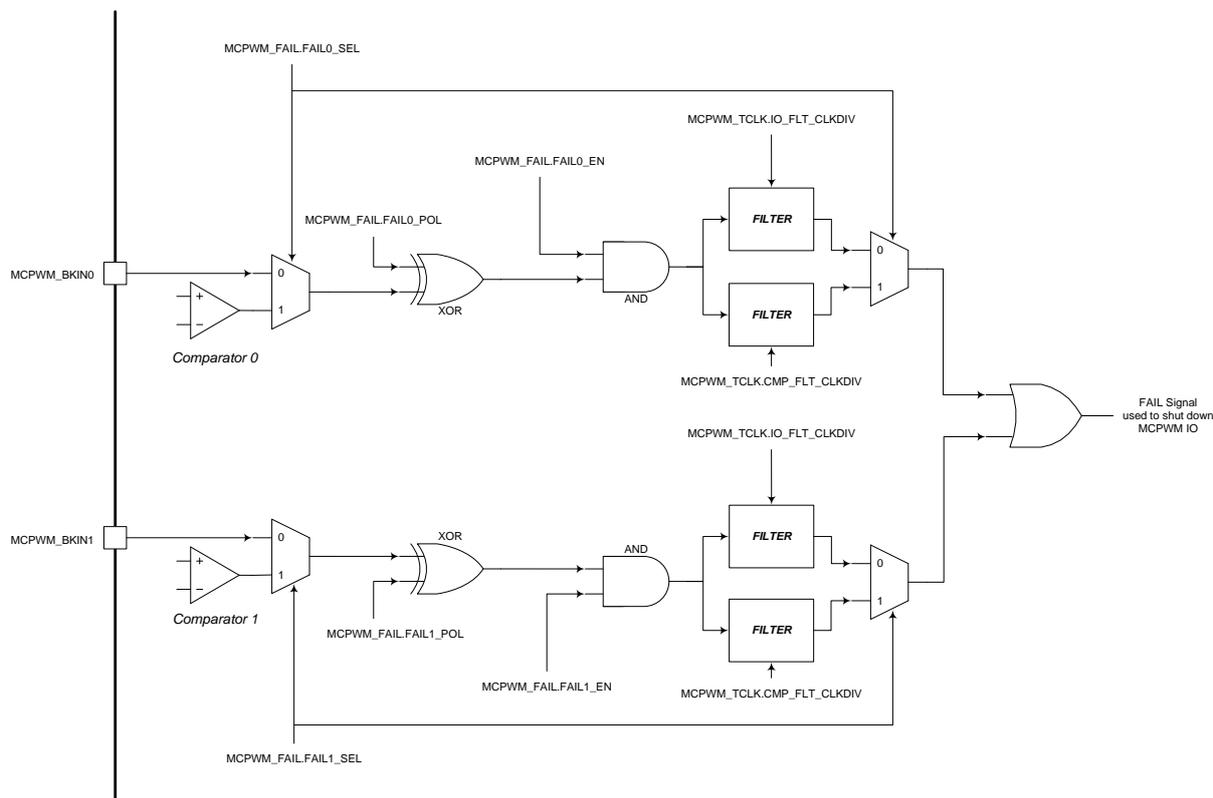


图 12-4 MCPWM FAIL 逻辑示意图

Filter 滤波模块的时钟，来自系统主时钟 MCLK 的门控时钟 FCLK[3]，并经过两级分频，第一级分频由 MCPWM\_TCLK.CLK\_DIV 控制，进行 1/2/4/8 倍分频。第二级分频可实现 1~16 倍的分频，



如果 Fail 信号来自 MCPWM\_BKIN[1:0]则使用 MCPWM\_TCLK.IO\_FLT\_CLKDIV[3:0]作为第二级的分频系数；如果 Fail 信号来自芯片内部比较器输出，则使用 MCPWM\_TCLK.CMP\_FLT\_CLKDIV[3:0]作为第二级的分频系数，如图 12-5 所示。

MCPWM 模块使用分频后的时钟进行 Fail 信号滤波，滤波宽度固定为 16 个周期，即输入信号必须保持至少 16 个时钟（两级分频后的时钟）周期稳定后，硬件才判定其为有效输入信号。滤波时间常数的公式为，其中  $T_{MCLK}$  为  $MCLK/FCLK[3]$  的时钟周期，96MHz 对应 10.4ns。MCPWM\_TCLK.FLT\_CLKDIV 根据配置情况可能是 MCPWM\_TCLK.IO\_FLT\_CLKDIV 或 MCPWM\_TCLK.CMP\_FLT\_CLKDIV。

$$T = T_{MCLK} \times (MCPWM\_TCLK\_CLK\_DIV) \times (MCPWM\_TCLK\_FLT\_CLKDIV + 1) \times 16$$

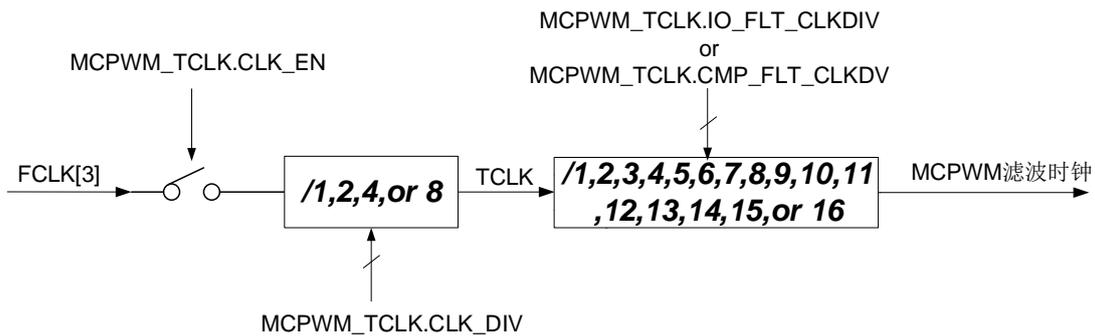


图 12-5 MCPWM Fail 信号滤波时钟生成逻辑

一旦发生 Fail 事件，硬件将 IO 输出强制变为 MCPWM\_FAIL.CHxN\_DEFAULT 和 MCPWM\_FAIL.CHxP\_DEFAULT 寄存器所指定的故障缺省值，此时 MCPWM\_FAIL.CHxN\_DEFAULT 和 MCPWM\_FAIL.CHxP\_DEFAULT 的值直接输出到 IO 口，不再受到 MCPWM\_FAIL.FAIL\_POL 等极性控制的影响。

来自比较器的 MCPWM FAIL 信号为模拟比较器输出的原始信号，未经过比较器数字接口模块的滤波处理，但是可以被 MCPWM 的通道信号进行开窗控制，开窗控制设置见比较器数字接口模块。FAIL 信号进入 MCPWM 模块后，可以通过设置 MCPWM\_TCLK 进行滤波。

### 12.1.3 MCPWM 特殊输出状态

电机控制中经常会用到全零和全 1 输出状态，以下互补模式设置可以得到期望的输出。

1. 如果  $THn0 \geq THn1$ ，芯片处于恒 0 状态（CH<n>P 关闭，CH<n>N 开启），无死区
2. 如果  $THn0 = -TH$ ， $THn1 = TH$ ，芯片处于恒 1 状态（CH<n>P 开启，CH<n>N 关闭），无死区

### 12.1.4 IO DRIVER 模块

该模块根据实际 MCPWM 的寄存器配置情况，将 IO 设置到相应电平。IO Driver 模块的整体数据流程图如下：





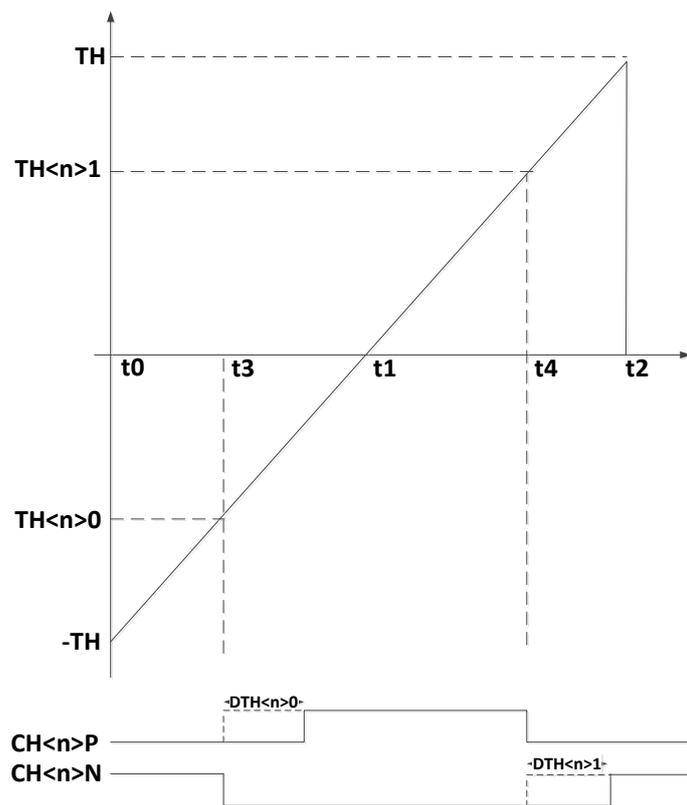


图 12-7 MCPWM 时序 TH<n>0 和 TH<n>1-中心对齐模式

#### 12.1.4.2 MCPWM 波形控制-中心对齐推挽模式

中心对齐推挽模式。第一个周期内，在  $t_3$  时刻  $CH<n>P$  打开，在  $t_4$  时刻， $CH<n>P$  关闭。第二个周期内，在  $t_3$  时刻  $CH<n>N$  打开，在  $t_4$  时刻， $CH<n>N$  关闭。

$t_3/t_4$  时刻均会产生相应中断。

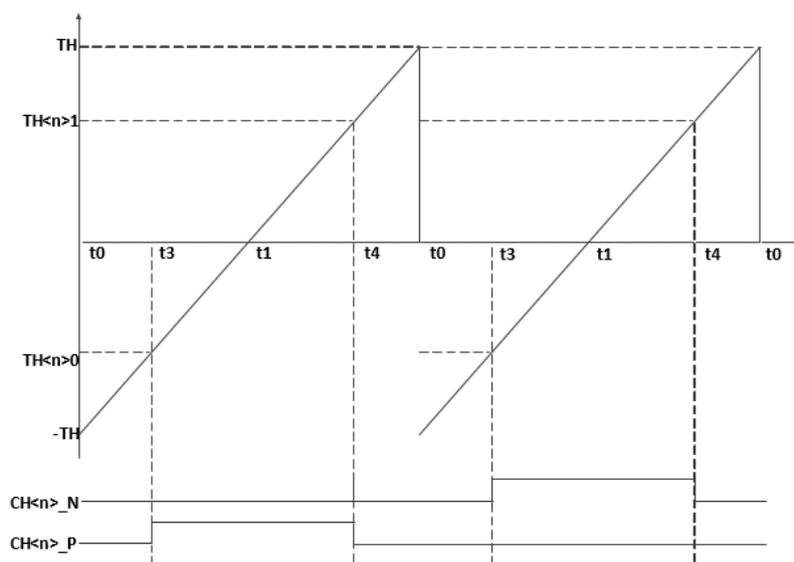


图 12-8 MCPWM 时序 TH<n>0 和 TH<n>1-中心对齐推挽模式



## 12.1.4.3 MCPWM 波形输出-边沿对齐模式

边沿对齐模式。在  $t_0$  时刻  $CH\langle n \rangle P/CH\langle n \rangle N$  同时打开，在  $t_3$  时刻， $CH\langle n \rangle P$  关闭；在  $t_4$  时刻， $CH\langle n \rangle N$  关闭。

$t_3/t_4$  时刻均会产生相应中断。

边沿对齐模式下， $CH\langle n \rangle P/CH\langle n \rangle N$  无需死区保护。

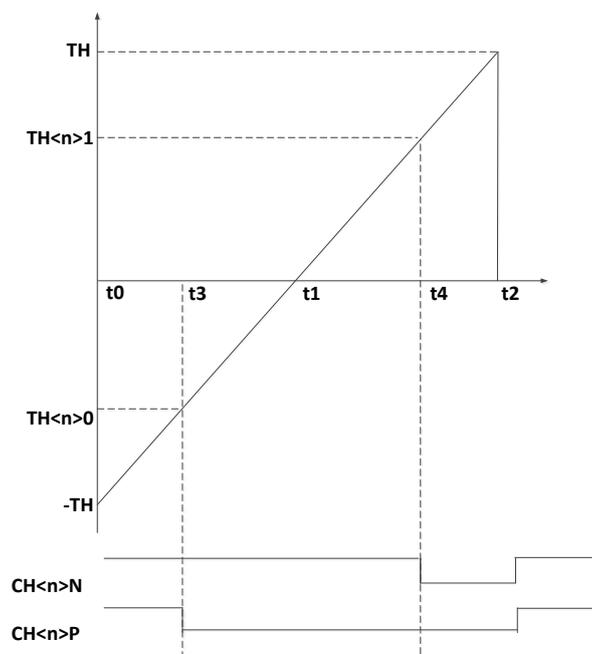


图 12-9 MCPWM 时序边沿对齐模式

## 12.1.4.4 MCPWM 波形控制-边沿对齐推挽模式

边沿对齐推挽模式。第一个周期内，在  $t_0$  时刻  $CH\langle n \rangle P$  打开，在  $t_3$  时刻， $CH\langle n \rangle P$  关闭。第二个周期内，在  $t_0$  时刻  $CH\langle n \rangle N$  打开，在  $t_3$  时刻， $CH\langle n \rangle N$  关闭。

$t_0/t_3$  时刻均会产生相应中断。

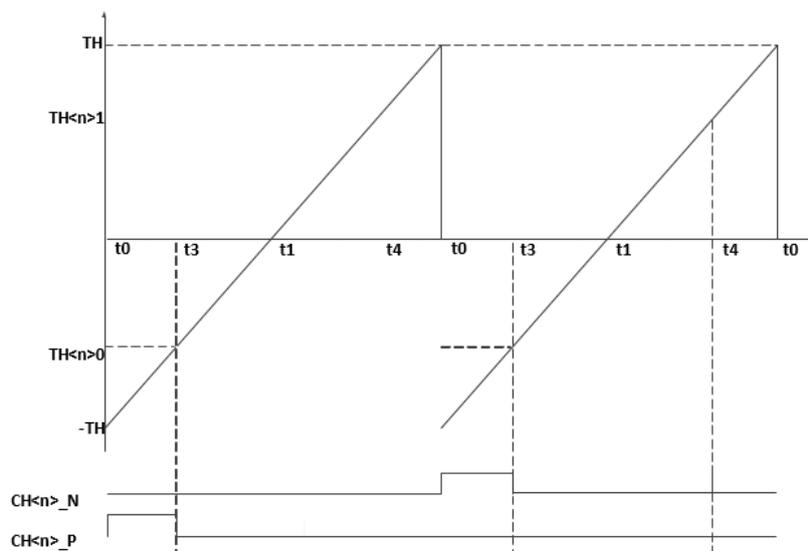


图 12-10 MCPWM 时序 TH&lt;n&gt;0 和 TH&lt;n&gt;1 边沿对齐推挽模式

#### 12.1.4.5 MCPWM IO 死区控制

MCPWM IO 是一对互斥控制信号 CH<n>P/CH<n>N，控制如下图所示的电路，

当 CH<n>P 为高/CH<n>N 为低时，Vout 输出高（VDD）；

当 CH<n>P 为低/CH<n>N 为高时，Vout 输出低（VSS）；

当 CH<n>P 为高/CH<n>N 为高时，Vout 输出不确定，但是会产生 VDD 到 VSS 的短路；

当 CH<n>P 为低/CH<n>N 为低时，Vout 输出不确定。

必须避免 CH<n>P/CH<n>N 同时打开的情况。死区的引入，可以有效避免 VDD 到 VSS 的短路。

四组 MCPWM IO 的死区宽度可独立调整。

对于互补模式 MCPWM IO 自动插入死区。

对于边沿对齐模式，MCPWM IO 无死区。

在 IO Driver 模块中增加 CH<n>P/CH<n>N 冲突检测，发生冲突时，自动将 IO 拉低，同时给出错误中断（错误中断标志位可软件清除或者硬件自动清除）。

MCPWM IO 也可通过软件配置的方式输出，此时，死区控制通过软件实现，如果 MCPWM 模块配置为中心对齐模式，硬件短路保护机制仍有效，保证 P 和 N 不同时打开。

CH<n>P/CH<n>N，在 IO 上可以互换。

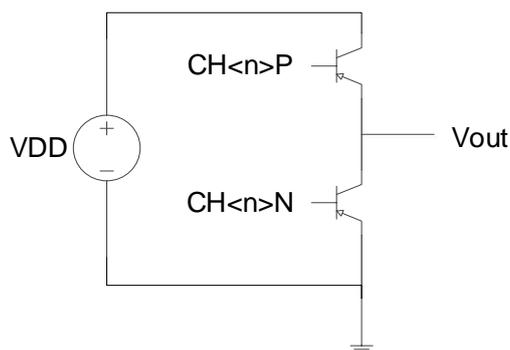


图 12-11 MCPWM IO 控制示意图

#### 12.1.4.6 MCPWM IO 极性设置

CH<n>P/CH<n>N 的有效电平可以配置为高有效/低有效，每个 IO 的有效电平单独可配。CH<n>P/CH<n>N 输出到 IO 的位置通过软件配置可以互换。

#### 12.1.4.7 MCPWM IO 自动保护

当发生急停事件（Fail 事件），应立刻将 CH<n>P/CH<n>N 自动切换到关闭状态。需要注意关闭电平配置（MCPWM\_FAIL.CHxN\_DEFAULT 和 MCPWM\_FAIL.CHxP\_DEFAULT 控制默认电平）。

- 芯片正常工作后，IO 默认输出的电平是寄存器 MCPWM\_FAIL.CHxN\_DEFAULT 和 MCPWM\_FAIL.CHxP\_DEFAULT 指定值，当用户配置完毕，MCPWM 正常工作后，配置 MCPWM\_FAIL.MCPWM\_OE（即 MOE）为 1，IO 输出电平受到 MCPWM IO 模块控制。
- 当发生 FAIL 短路状况时，硬件立即切换到 IO 默认输出电平。
- 当芯片调试中，发生 MCU Halt 时，可配置实现停止 MCPWM 正常输出，转而输出 FAIL[15:8] 寄存器的值。

### 12.1.5 ADC Trigger Timer 模块

MCPWM 可以提供 ADC 采样控制。当计数器计数到 MCPWM\_TMR0/ MCPWM\_TMR1/ MCPWM\_TMR2/ MCPWM\_TMR3 时，可产生定时事件，触发 ADC 采样。该触发信号可同时输出到 GPIO，便于调试之用。输出的具体 GPIO，参见对应器件的 datasheet。

表 12-1 MCPWM 计数器阈值与事件对应表

t0	-th
t1	0
tio0[0]	th00
tio0[1]	th01
TADC[0]	tmr0
TADC [1]	tmr1
TADC [2]	tmr2
TADC [3]	tmr3



## 12.2 寄存器

### 12.2.1 地址分配

MCPWM 模块寄存器的基地址是 0x40011C00。

表 12-2 MCPWM 模块寄存器列表

名称	偏移地址	说明
MCPWM_TH00	0x00	MCPWM CH0_P 比较门限值寄存器
MCPWM_TH01	0x04	MCPWM CH0_N 比较门限值寄存器
MCPWM_TH10	0x08	MCPWM CH1_P 比较门限值寄存器
MCPWM_TH11	0x0C	MCPWM CH1_N 比较门限值寄存器
MCPWM_TH20	0x10	MCPWM CH2_P 比较门限值寄存器
MCPWM_TH21	0x14	MCPWM CH2_N 比较门限值寄存器
MCPWM_TH30	0x18	MCPWM CH3_P 比较门限值寄存器
MCPWM_TH31	0x1C	MCPWM CH3_N 比较门限值寄存器
MCPWM_TMR0	0x20	ADC 采样定时器比较门限 0 寄存器
MCPWM_TMR1	0x24	ADC 采样定时器比较门限 1 寄存器
MCPWM_TMR2	0x28	ADC 采样定时器比较门限 2 寄存器
MCPWM_TMR3	0x2C	ADC 采样定时器比较门限 3 寄存器
MCPWM_TH	0x30	MCPWM 门限值寄存器
MCPWM_UPDATE	0x34	MCPWM 加载控制寄存器
MCPWM_IE	0x38	MCPWM 中断控制寄存器
MCPWM_IF	0x3C	MCPWM 中断标志位寄存器
MCPWM_EIE	0x40	MCPWM 异常中断控制寄存器
MCPWM{EIF	0x44	MCPWM 异常中断标志位寄存器
	0x48	保留
MCPWM_PP	0x4C	MCPWM 推挽模式使能寄存器
MCPWM_IO01	0x50	MCPWM IO01 控制寄存器
MCPWM_IO23	0x54	MCPWM IO23 控制寄存器
MCPWM_SDCFG	0x58	MCPWM 加载配置寄存器
MCPWM_AUEN	0x5C	MCPWM 自动更新使能寄存器
MCPWM_TCLK	0x60	MCPWM 时钟分频控制寄存器
MCPWM_FAIL	0x64	MCPWM 短路控制寄存器
	0x68	保留
	0x6C	保留
MCPWM_PRT	0x74	MCPWM 保护寄存器
MCPWM_CNT	0x78	MCPWM 计数器寄存器
MCPWM_SWAP	0x7C	MCPWM 通道映射寄存器
MCPWM_DTH00	0x80	MCPWM CH0 P 通道死区宽度控制寄存器
MCPWM_DTH01	0x84	MCPWM CH0 N 通道死区宽度控制寄存器
MCPWM_DTH10	0x88	MCPWM CH1 P 通道死区宽度控制寄存器
MCPWM_DTH11	0x8C	MCPWM CH1 N 通道死区宽度控制寄存器



MCPWM_DTH20	0x90	MCPWM CH2 P 通道死区宽度控制寄存器
MCPWM_DTH21	0x94	MCPWM CH2 N 通道死区宽度控制寄存器
MCPWM_DTH30	0x98	MCPWM CH3 P 通道死区宽度控制寄存器
MCPWM_DTH31	0x9C	MCPWM CH3 N 通道死区宽度控制寄存器

表 12-3 受 MCPWM\_PRT 保护的寄存器

名称	偏移地址	说明
MCPWM_TH	0x30	MCPWM 门限值寄存器
MCPWM_IE	0x38	MCPWM 中断控制寄存器
MCPWM_EIE	0x40	MCPWM 异常中断控制寄存器
MCPWM_PP	0x4C	MCPWM 推挽模式使能寄存器
MCPWM_IO01	0x50	MCPWM IO01 控制寄存器
MCPWM_IO23	0x54	MCPWM IO23 控制寄存器
MCPWM_SDCFG	0x58	MCPWM 加载配置寄存器
MCPWM_AUEN	0x5C	MCPWM 自动加载使能寄存器
MCPWM_TCLK	0x60	MCPWM 时钟分频控制寄存器
MCPWM_FAIL	0x64	MCPWM 短路控制寄存器
MCPWM_DTH00	0x80	MCPWM CH0 P 通道死区宽度控制寄存器
MCPWM_DTH01	0x84	MCPWM CH0 N 通道死区宽度控制寄存器
MCPWM_DTH10	0x88	MCPWM CH1 P 通道死区宽度控制寄存器
MCPWM_DTH11	0x8C	MCPWM CH1 N 通道死区宽度控制寄存器
MCPWM_DTH20	0x90	MCPWM CH2 P 通道死区宽度控制寄存器
MCPWM_DTH21	0x94	MCPWM CH2 N 通道死区宽度控制寄存器
MCPWM_DTH30	0x98	MCPWM CH3 P 通道死区宽度控制寄存器
MCPWM_DTH31	0x9C	MCPWM CH3 N 通道死区宽度控制寄存器

表 12-4 存在影子寄存器的寄存器

名称	偏移地址	说明
MCPWM_TH00	0x00	MCPWM CH0_P 比较门限值寄存器
MCPWM_TH01	0x04	MCPWM CH0_N 比较门限值寄存器
MCPWM_TH10	0x08	MCPWM CH1_P 比较门限值寄存器
MCPWM_TH11	0x0C	MCPWM CH1_N 比较门限值寄存器
MCPWM_TH20	0x10	MCPWM CH2_P 比较门限值寄存器
MCPWM_TH21	0x14	MCPWM CH2_N 比较门限值寄存器
MCPWM_TH30	0x18	MCPWM CH3_P 比较门限值寄存器
MCPWM_TH31	0x1C	MCPWM CH3_N 比较门限值寄存器
MCPWM_TMR0	0x20	ADC 采样定时器比较门限 0 寄存器
MCPWM_TMR1	0x24	ADC 采样定时器比较门限 1 寄存器
MCPWM_TMR2	0x28	ADC 采样定时器比较门限 2 寄存器
MCPWM_TMR3	0x2C	ADC 采样定时器比较门限 3 寄存器
MCPWM_TH	0x30	MCPWM 门限值寄存器
MCPWM_CNT	0x78	MCPWM 计数器寄存器



### 12.2.2 MCPWM\_TH00

无写保护的寄存器

地址：0x4001\_1C00

复位值：0x0

表 12-5 MCPWM\_TH00 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH00															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TH00	MCPWM CH0_P 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.3 MCPWM\_TH01

无写保护的寄存器

地址：0x4001\_1C04

复位值：0x0

表 12-6 MCPWM\_TH01 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH01															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TH01	MCPWM CH0_N 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.4 MCPWM\_TH10

无写保护的寄存器

地址：0x4001\_1C08

复位值：0x0



表 12-7 MCPWM\_TH10 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH10															
RW															
0															

位置	位名称	说明
[31:16]		未使用
15:0]	TH10	MCPWM CH1_P 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.5 MCPWM\_TH11

无写保护的寄存器

地址：0x4001\_1C0C

复位值：0x0

表 12-8 MCPWM\_TH11 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH11															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TH11	MCPWM CH1_N 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.6 MCPWM\_TH20

无写保护的寄存器

地址：0x4001\_1C10

复位值：0x0

表 12-9 MCPWM\_TH20 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH20															
RW															
0															



位置	位名称	说明
[31:16]		未使用
[15:0]	TH20	MCPWM CH2_P 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.7 MCPWM\_TH21

无写保护的寄存器

地址：0x4001\_1C14

复位值：0x0

表 12-10 MCPWM\_TH21 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH21															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TH21	MCPWM CH2_N 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.8 MCPWM\_TH30

无写保护的寄存器

地址：0x4001\_1C18

复位值：0x0

表 12-11 MCPWM\_TH30 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH30															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TH30	MCPWM CH3_P 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。



### 12.2.9 MCPWM\_TH31

无写保护的寄存器

地址：0x4001\_1C1C

复位值：0x0

表 12-12 MCPWM\_TH31 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH31															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TH31	MCPWM CH3_N 比较门限值，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.10 MCPWM\_TMR0

无写保护的寄存器

地址：0x4001\_1C20

复位值：0x0

表 12-13 MCPWM\_TMR0 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TMR0	ADC 采样定时器比较门限 0 寄存器，16 位有符号数；发生更新事件后，本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.11 MCPWM\_TMR1

无写保护的寄存器

地址：0x4001\_1C24

复位值：0x0



表 12-14 MCPWM\_TMR1 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR1															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TMR1	ADC 采样定时器比较门限 1 寄存器, 16 位有符号数; 发生更新事件后, 本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.12 MCPWM\_TMR2

无写保护的寄存器

地址: 0x4001\_1C28

复位值: 0x0

表 12-15 MCPWM\_TMR2 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR2															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	TMR2	ADC 采样定时器比较门限 2 寄存器, 16 位有符号数; 发生更新事件后, 本寄存器加载到 MCPWM 实际运行系统中。

### 12.2.13 MCPWM\_TMR3

无写保护的寄存器

地址: 0x4001\_1C2C

复位值: 0x0

表 12-16 MCPWM\_TMR3 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR3															
RW															
0															



位置	位名称	说明
[31:16]		未使用
[15:0]	TMR3	ADC 采样定时器比较门限 3 寄存器, 16 位有符号数; 发生更新事件后, 本寄存器加载到 MCPWM 实际运行系统中。。

### 12.2.14 MCPWM\_TH

写保护的寄存器

地址: 0x4001\_1C30

复位值: 0x0

表 12-17 MCPWM\_TH 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH															
RW															
0															

位置	位名称	说明
[31:15]		未使用
[14:0]	TH	MCPWM 计数器门限值, 15 位无符号数, MCPWM 实际运行系统中的计数器从 -TH 计数到 TH; 发生更新事件后, 本寄存器加载到 MCPWM 实际运行系统中。。

### 12.2.15 MCPWM\_UPDATE

无写保护的寄存器

地址: 0x4001\_1C34

复位值: 0x0

表 12-18 MCPWM\_UPDATE MCPWM 手动更新寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT_UPDATE	TH_UPDATE	TMR3_UPDATE	TMR2_UPDATE	TMR1_UPDATE	TMR0_UPDATE	TH31_UPDATE	TH30_UPDATE	TH21_UPDATE	TH20_UPDATE	TH11_UPDATE	TH10_UPDATE	TH01_UPDATE	TH00_UPDATE	
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
----	-----	----



[31:14]		未使用
[13]	CNT_UPDATE	手动将加载 MCPWM_CNT 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[12]	TH_UPDATE	手动将加载 MCPWM_TH 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[11]	TMR3_UPDATE	手动将加载 MCPWM_TMR3 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[10]	TMR2_UPDATE	手动将加载 MCPWM_TMR2 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[9]	TMR1_UPDATE	手动将加载 MCPWM_TMR1 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[8]	TMRO_UPDATE	手动将加载 MCPWM_TMRO 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[7]	TH31_UPDATE	手动将加载 MCPWM_TH31 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[6]	TH30_UPDATE	手动将加载 MCPWM_TH30 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[5]	TH21_UPDATE	手动将加载 MCPWM_TH21 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[4]	TH20_UPDATE	手动将加载 MCPWM_TH20 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[3]	TH11_UPDATE	手动将加载 MCPWM_TH11 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[2]	TH10_UPDATE	手动将加载 MCPWM_TH10 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[1]	TH01_UPDATE	手动将加载 MCPWM_TH01 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。
[0]	TH00_UPDATE	手动将加载 MCPWM_TH00 寄存器的内容到 MCPWM 运行系统中。 1: 加载; 0: 不加载。

向 MCPWM\_UPDATE 对应位写 1 可以触发寄存器值从预装载寄存器写入影子寄存器，MCPWM\_UPDATE 写入后自动清零。每写 1 一次，进行一次软件/手动触发。向 MCPWM\_UPDATE[13] 写 1 会导致 MCPWM\_CNT 更新为预装载值，使用请留意，应用上是否需要更新 CNT 值。

## 12.2.16 MCPWM\_IE

写保护的寄存器

地址: 0x4001\_1C38

复位值: 0x0

表 12-19 MCPWM\_IE 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---



	SHADE_IE	TMR3_IE	TMR2_IE	TMR1_IE	TMR0_IE	TH31_IE	TH30_IE	TH21_IE	TH20_IE	TH11_IE	TH10_IE	TH01_IE	TH00_IE	T1_IE	T0_IE
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位置	位名称	说明													
[31:15]		未使用													
[14]	SHADE_IE	MCPWM_TH/MCPWM_TH00~MCPWM_TH31/MCPWM_TMR0~MCPWM_TMR3 等寄存器更新到 MCPWM 实际运行系统的中断源使能。 1, 使能; 0, 关闭。													
[13]	TMR3_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TMR3 中断源使能。 1, 使能; 0, 关闭。													
[12]	TMR2_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TMR2 中断源使能。 1, 使能; 0, 关闭。													
[11]	TMR1_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TMR1 中断源使能。 1, 使能; 0, 关闭。													
[10]	TMR0_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TMR0 中断源使能。 1, 使能; 0, 关闭。													
[9]	TH31_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH31 中断源使能。 1, 使能; 0, 关闭。													
[8]	TH30_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH30 中断源使能。 1, 使能; 0, 关闭。													
[7]	TH21_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH21 中断源使能。 1, 使能; 0, 关闭。													
[6]	TH20_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH20 中断源使能。 1, 使能; 0, 关闭。													
[5]	TH11_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH11 中断源使能。 1, 使能; 0, 关闭。													
[4]	TH10_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH10 中断源使能。 1, 使能; 0, 关闭。													
[3]	TH01_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH01 中断源使能。 1, 使能; 0, 关闭。													
[2]	TH00_IE	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH00 中断源使能。 1, 使能; 0, 关闭。													
[1]	T1_IE	t1 事件, 计数器的计数值到达 0 中断源使能。 1, 使能; 0, 关闭。													
[0]	T0_IE	t0 事件, 计数器的计数值回到 MCPWM_TH 中断源使能。 1, 使能; 0, 关闭。													

### 12.2.17 MCPWM\_IF

无写保护的寄存器



地址：0x4001\_1C3C

复位值：0x0

表 12-20 MCPWM\_IF 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SHADE_IF	TMR3_IF	TMR2_IF	TMR1_IF	TMR0_IF	TH31_IF	TH30_IF	TH21_IF	TH20_IF	TH11_IF	TH10_IF	TH01_IF	TH00_IF	T1_IF	T0_IF
	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:15]		未使用
[14]	SHADE_IF	MCPWM_TH/MCPWM_TH00~MCPWM_TH31/MCPWM_TMR0~MCPWM_TMR3 等寄存器更新到 MCPWM 实际运行系统的中断源事件。 1, 发生; 0, 没发生。写 1 清零。
[13]	TMR3_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TMR3 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[12]	TMR2_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TMR2 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[11]	TMR1_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TMR1 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[10]	TMR0_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TMR0 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[9]	TH31_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH31 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[8]	TH30_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH30 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[7]	TH21_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH21 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[6]	TH20_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH20 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[5]	TH11_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH11 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[4]	TH10_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH10 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[3]	TH01_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH01 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[2]	TH00_IF	MCPWM 实际运行系统中计数器的计数值等于 MCPWM_TH00 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[1]	T1_IF	t1 事件, 计数器的计数值到达 0 中断源事件。 1, 发生; 0, 没发生。写 1 清零。



[0]	T0_IF	t0 事件，计数器的计数值回到 MCPWM_TH 中断源事件。 1，发生；0，没发生。写 1 清零。
-----	-------	---

### 12.2.18 MCPWM\_EIE

写保护的寄存器

地址：0x4001\_1C40

复位值：0x0

表 12-21 MCPWM\_EIE 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
											FAIL1_IE	FAIL0_IE	CH3_SHORT_IE	CH2_SHORT_IE	CH1_SHORT_IE	CH0_SHORT_IE
											RW	RW	RW	RW	RW	RW
											0	0	0	0	0	0

位置	位名称	说明
[31:6]		未使用
[5]	FAIL1_IE	FAIL1 中断源使能。1，使能；0，关闭。
[4]	FAIL0_IE	FAIL0 中断源使能。1，使能；0，关闭。
[3]	CH3_SHORT_IE	MCPWM CH3_P 和 CH3_N 同时有效，中断源使能。1，使能；0，关闭。
[2]	CH2_SHORT_IE	MCPWM CH2_P 和 CH2_N 同时有效，中断源使能。1，使能；0，关闭。
[1]	CH1_SHORT_IE	MCPWM CH1_P 和 CH1_N 同时有效，中断源使能。1，使能；0，关闭。
[0]	CH0_SHORT_IE	MCPWM CH0_P 和 CH0_N 同时有效，中断源使能。1，使能；0，关闭。

### 12.2.19 MCPWM{EIF

无写保护的寄存器

地址：0x4001\_1C44

复位值：0x0

表 12-22 MCPWM{EIF 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
											FAIL1_IF	FAIL0_IF	CH3_SHORT_IF	CH2_SHORT_IF	CH1_SHORT_IF	CH0_SHORT_IF
											RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
											0	0	0	0	0	0



位置	位名称	说明
[31:6]		未使用
[5]	FAIL1_IF	FAIL1 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[4]	FAIL0_IF	FAIL0 中断源事件。1, 发生; 0, 没发生。写 1 清零。
[3]	CH3_SHORT_IF	MCPWM CH3_P 和 CH3_N 同时有效, 中断源事件。 1, 发生; 0, 没发生。写 1 清零。
[2]	CH2_SHORT_IF	MCPWM CH2_P 和 CH2_N 同时有效, 中断源事件。 1, 发生; 0, 没发生。写 1 清零。
[1]	CH1_SHORT_IF	MCPWM CH1_P 和 CH1_N 同时有效, 中断源事件。 1, 发生; 0, 没发生。写 1 清零。
[0]	CH0_SHORT_IF	MCPWM CH0_P 和 CH0_N 同时有效, 中断源事件。 1, 发生; 0, 没发生。写 1 清零。

### 12.2.20 MCPWM\_PP

写保护的寄存器

地址: 0x4001\_1C4C

复位值: 0x0

表 12-23 MCPWM\_PP 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												IO3_PPE	IO2_PPE	IO1_PPE	IO0_PPE
												RW	RW	RW	RW
												0	0	0	0

位置	位名称	说明
[31:4]		未使用
[3]	IO3_PPE	IO3 推挽模式使能信号。写 1, 使能; 写 0, 关闭。
[2]	IO2_PPE	IO2 推挽模式使能信号。写 1, 使能; 写 0, 关闭。
[1]	IO1_PPE	IO1 推挽模式使能信号。写 1, 使能; 写 0, 关闭。
[0]	IO0_PPE	IO0 推挽模式使能信号。写 1, 使能; 写 0, 关闭。

推挽模式使能信号。根据工作模式不同而不同。边沿模式, 开启边沿模式的推挽模式; 中心对齐, 开启中心对齐的推挽模式。

### 12.2.21 MCPWM\_IO01

写保护的寄存器

地址: 0x4001\_1C50



复位值: 0x0

表 12-24 MCPWM\_IO01 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1_WM	CH1_PN_SW	CH1_SCTRLP	CH1_SCTRLN	CH1_PS	CH1_NS	CH1_PP	CH1_NP	CH0_WM	CH0_PN_SW	CH0_SCTRLP	CH0_SCTRLN	CH0_PS	CH0_NS	CH0_PP	CH0_NP
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	CH1_WM	CH1 工作模式选择。1: 边沿模式; 0: 互补模式。
[14]	CH1_PN_SW	CH1 的 P 和 N 通道输出互换选择。即 P 通道信号最后从 N 通道输出, N 通道的信号最后从 P 通道输出。1: 互换; 0: 不互换。
[13]	CH1_SCTRLP	当 CH1_PS=1 时, 输出到 CH1 P 通道的值。
[12]	CH1_SCTRLN	当 CH1_NS=1 时, 输出到 CH1 N 通道的值。
[11]	CH1_PS	CH1 P 来源。1: 来自 CH1_SCTRLP; 0: MCPWM 内部计数器产生。
[10]	CH1_NS	CH1 N 来源。1: 来自 CH1_SCTRLN; 0: MCPWM 内部计数器产生。
[9]	CH1_PP	CH1 P 极性选择。1: CH1 P 信号取反输出; 0: CH1 P 信号正常输出。
[8]	CH1_NP	CH1 N 极性选择。1: CH1 N 信号取反输出; 0: CH1 N 信号正常输出。
[7]	CH0_WM	CH0 工作模式选择。1: 边沿模式; 0: 互补模式。
[6]	CH0_PN_SW	CH0 的 P 和 N 通道输出互换选择。即 P 通道信号最后从 N 通道输出, N 通道的信号最后从 P 通道输出。1: 互换; 0: 不互换。
[5]	CH0_SCTRLP	当 CH0_PS=1 时, 输出到 CH0 P 通道的值。
[4]	CH0_SCTRLN	当 CH0_NS=1 时, 输出到 CH0 N 通道的值。
[3]	CH0_PS	CH0 P 来源。1: 来自 CH0_SCTRLP; 0: MCPWM 实际运行系统中计数器产生。
[2]	CH0_NS	CH0 N 来源。1: 来自 CH0_SCTRLN; 0: MCPWM 实际运行系统中计数器产生。
[1]	CH0_PP	CH0 P 极性选择。1: CH0 P 信号取反输出; 0: CH0 P 信号正常输出。
[0]	CH0_NP	CH0 N 极性选择。1: CH0 N 信号取反输出; 0: CH0 N 信号正常输出。 极性选择跟随通道交换, 例如 CH0 N 选择取反输出, 同时选择了通道交换, 则交换后的 CH0 N 仍是取反输出。

### 12.2.22 MCPWM\_IO23

写保护的寄存器

地址: 0x4001\_1C54

复位值: 0x0



表 12-25 MCPWM\_IO23 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3_WM	CH3_PN_SW	CH3_SCTRLP	CH3_SCTRLN	CH3_PS	CH3_NS	CH3_PP	CH3_NP	CH2_WM	CH2_PN_SW	CH2_SCTRLP	CH2_SCTRLN	CH2_PS	CH2_NS	CH2_PP	CH2_NP
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	位名称	说明
[31:16]		未使用
[15]	CH3_WM	CH3 工作模式选择。1: Edge 模式; 0: 互补模式。
[14]	CH3_PN_SW	CH3 的 P 和 N 通道输出互换选择。即 P 通道信号最后从 N 通道输出, N 通道的信号最后从 P 通道输出。1: 互换; 0: 不互换。
[13]	CH3_SCTRLP	当 CH3_PS =1 时, 输出到 CH3 P 通道的值。
[12]	CH3_SCTRLN	当 CH3_NS =1 时, 输出到 CH3 N 通道的值。
[11]	CH3_PS	CH3 P 来源。1: 来自 CH3_SCTRLP; 0: MCPWM 实际运行系统中计数器产生。
[10]	CH3_NS	CH3 N 来源。1: 来自 CH3_SCTRLN; 0: MCPWM 实际运行系统中计数器产生。
[9]	CH3_PP	CH3 P 极性选择。1: CH3 P 信号取反输出; 0: CH3 P 信号正常输出。
[8]	CH3_NP	CH3 N 极性选择。1: CH3 N 信号取反输出; 0: CH3 N 信号正常输出。
[7]	CH2_WM	CH2 工作模式选择。1: Edge 模式; 0: 互补模式。
[6]	CH2_PN_SW	CH2 的 P 和 N 通道输出互换选择。即 P 通道信号最后从 N 通道输出, N 通道的信号最后从 P 通道输出。1: 互换; 0: 不互换。
[5]	CH2_SCTRLP	当 CH2_PS =1 时, 输出到 CH2 P 通道的值。
[4]	CH2_SCTRLN	当 CH2_NS =1 时, 输出到 CH2 N 通道的值。
[3]	CH2_PS	CH2 P 来源。1: 来自 CH2_SCTRLP; 0: MCPWM 实际运行系统中计数器产生。
[2]	CH2_NS	CH2 N 来源。1: 来自 CH2_SCTRLN; 0: MCPWM 实际运行系统中计数器产生。
[1]	CH2_PP	CH2 P 极性选择。1: CH2 P 信号取反输出; 0: CH2 P 信号正常输出。
[0]	CH2_NP	CH2 N 极性选择。1: CH2 N 信号取反输出; 0: CH2 N 信号正常输出。 极性选择跟随通道交换, 例如 CH0 N 选择取反输出, 同时选择了通道交换, 则交换后的 CH0 N 仍是取反输出。

### 12.2.23 MCPWM\_SDCFG

写保护的寄存器

地址: 0x4001\_1C58



复位值: 0x0

表 12-26 MCPWM\_SDCFG 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										AUTO_ERR_CLR	T1_UPDATE_EN	T0_UPDATE_EN	UPDATE_INTV		
										RW	RW	RW	RW		
										0	0	0	0		

位置	位名称	说明
[31:8]		未使用
[7]	RESERVED	保留位
[6]	AUTO_ERR_CLR	AUTO_ERR_CLR 更新事件是否自动清除 MCPWM_EIF[5:4]并置位 MOE, 恢复 MCPWM 信号输出。 1: 使能自动故障清除功能; 0: 关闭自动故障清除功能。
[5]	T1_UPDATE_EN	t1 (过零) 事件更新使能。1: 使能; 0, 关闭。
[4]	T0_UPDATE_EN	t0 (起点) 事件更新使能。1: 使能; 0, 关闭。
[3:0]	UPDATE_INTV	更新间隔。一旦 t0/t1 发生次数同 UPDATE_INTV+1 相等, MCPWM 系统自动触发 MCPWM_TH (包括 THxx) 和 MCPWM_TMR 寄存器加载到 MCPWM 运行系统的操作。若 B[5]和 B[4]均关闭, 将不会触发此类型加载, 只能手动触发加载。

### 12.2.24 MCPWM\_AUEN

写保护的寄存器

地址: 0x4001\_1C5C

复位值: 0x0

表 12-27 MCPWM\_AUEN MCPWM 自动更新使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		CNT_AUEN	TH_AUEN	TMR3_AUEN	TMR2_AUEN	TMR1_AUEN	TMR0_AUEN	TH31_AUEN	TH30_AUEN	TH21_AUEN	TH20_AUEN	TH11_AUEN	TH10_AUEN	TH01_AUEN	TH00_AUEN
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
		0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:14]		未使用
[13]	CNT_AUEN	MCPWM_CNT 自动加载使能。1: 加载; 0: 不加载。
[12]	TH_AUEN	MCPWM_TH 自动加载使能。1: 加载; 0: 不加载。
[11]	TMR3_AUEN	MCPWM_TMR3 自动加载使能。1: 加载; 0: 不加载。



[10]	TMR2_AUEN	MCPWM_TMR2 自动加载使能。1: 加载; 0: 不加载。
[9]	TMR1_AUEN	MCPWM_TMR1 自动加载使能。1: 加载; 0: 不加载。
[8]	TMRO_AUEN	MCPWM_TMRO 自动加载使能。1: 加载; 0: 不加载。
[7]	TH31_AUEN	MCPWM_TH31 自动加载使能。1: 加载; 0: 不加载。
[6]	TH30_AUEN	MCPWM_TH30 自动加载使能。1: 加载; 0: 不加载。
[5]	TH21_AUEN	MCPWM_TH21 自动加载使能。1: 加载; 0: 不加载。
[4]	TH20_AUEN	MCPWM_TH20 自动加载使能。1: 加载; 0: 不加载。
[3]	TH11_AUEN	MCPWM_TH11 自动加载使能。1: 加载; 0: 不加载。
[2]	TH10_AUEN	MCPWM_TH10 自动加载使能。1: 加载; 0: 不加载。
[1]	TH01_AUEN	MCPWM_TH01 自动加载使能。1: 加载; 0: 不加载。
[0]	TH00_AUEN	MCPWM_TH00 自动加载使能。1: 加载; 0: 不加载。

### 12.2.25 MCPWM\_TCLK

写保护的寄存器

地址: 0x4001\_1C60

复位值: 0x0

表 12-28 MCPWM\_TCLK 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP_FLT_CLKDIV				IO_FLT_CLKDIV								BASE_CNT_EN	CLK_EN	CLK_DIV	
RW				RW								RW	RW	RW	
0				0								0	0	0	

位置	位名称	说明
[31:16]		未使用
[15:12]	CMP_FLT_CLKDIV	比较器输出的滤波时钟分频寄存器, 基于系统时钟分频, 影响 MCPWM_FAIL[1:0]。计算公式如下: 系统时钟 / (B[15:12] + 1)。分频范围是 1-16。
[11:8]	IO_FLT_CLKDIV	GPIO 输入的滤波时钟分频寄存器, 基于系统时钟分频, 影响 MCPWM_FAIL[1:0]。计算公式如下: 系统时钟 / (B[11:8] + 1)。分频范围是 1-16。
[7:4]		未使用
[3]	BASE_CNT_EN	MCPWM 实际运行计数器使能开关。1: 使能; 0: 关闭。
[2]	CLK_EN	MCPWM 工作时钟使能。1: 使能; 0: 关闭。
[1:0]	CLK_DIV	MCPWM 工作时钟分频寄存器。 0: 系统时钟 1: 系统时钟/2 2: 系统时钟/4 3: 系统时钟/8



## 12.2.26 MCPWM\_FAIL

写保护的寄存器

地址：0x4001\_1C64

复位值：0x0

表 12-29 MCPWM\_FAIL 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3P_DEFAULT	CH3N_DEFAULT	CH2P_DEFAULT	CH2N_DEFAULT	CH1P_DEFAULT	CH1N_DEFAULT	CH0P_DEFAULT	CH0N_DEFAULT	HALT_PRT	MCPWM_OE	FAIL1_EN	FAIL0_EN	FAIL1_POL	FAIL0_POL	FAIL1_SEL	FAIL0_SEL
RW	RW	RW	RW	RW	RW	RW	RW	RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:16]		未使用
[15]	CH3N_DEFAULT	CH3 N 通道默认值
[14]	CH3P_DEFAULT	CH3 P 通道默认值
[13]	CH2N_DEFAULT	CH2 N 通道默认值
[12]	CH2P_DEFAULT	CH2 P 通道默认值
[11]	CH1N_DEFAULT	CH1 N 通道默认值
[10]	CH1P_DEFAULT	CH1 P 通道默认值
[9]	CH0N_DEFAULT	CH0 N 通道默认值
[8]	CH0P_DEFAULT	CH0 P 通道默认值。当发生 FAIL 事件或 MOE 为 0 时，相应通道输出默认电平。 <b>默认电平输出不受 MCPWM_IO01 和 MCPWM_IO23 的 BIT0、BIT1、BIT8、BIT9、BIT6、BIT14 通道交换和极性控制的影响，直接控制通道输出。</b>
[7]	HALT_PRT	MCU 进入 HALT 状态，MCPWM 输出值选择。 1：正常输出；0：强制 MCPWM 输出保护值。
[6]	MCPWM_OE	MOE 控制 MCPWM CH P 和 N 输出值。 1：输出 MCPWM 产生的正常信号 0：输出 CHxN_DEFAULT 和 CHxP_DEFAULT 默认值，此默认值不受极性/通道选择等控制。 MCPWM_EIF.FAIL1_IF 和 MCPWM_EIF.FAIL0_IF 任意一位变 1 将触发 MCPWM_OE 变成 0，输出默认值。
[5]	FAIL1_EN	FAIL1 输入使能。1：使能；0：关闭。
[4]	FAIL0_EN	FAIL0 输入使能。1：使能；0：关闭。
[3]	FAIL1_POL	FAIL1 极性选择。1：信号极性取反输入，信号输入低为有效电平； 0：信号极性正常输入，信号输入高为有效电平。
[2]	FAIL0_POL	FAIL0 极性选择。1：信号极性取反输入，信号输入低为有效电平； 0：信号极性正常输入，信号输入高为有效电平。



[1]	FAIL1_SEL	FAIL1 来源选择。1: 比较器 1 的结果; 0: 来自 GPIO 第 1 路。
[0]	FAIL0_SEL	FAIL0 来源选择。1: 比较器 0 的结果; 0: 来自 GPIO 第 0 路。

MCPWM\_FAIL 可以用来设置紧急停车事件, 封锁 MCPWM 的信号输出。急停事件主要有两个 FAIL0 和 FAIL1。共有 4 个信号来源, 比较器 0 输出和比较器 1 输出以及 MCPWM\_BKIN0 和 MCPWM\_BKIN1。FAIL0 可以来自比较器 0 输出或芯片 IO MCPWM\_BKIN0, FAIL1 可以来自比较器 1 输出或芯片 IO MCPWM\_BKIN1。

FAIL 的输入信号可以使用数字滤波, 滤波时钟的第一级分频由 MCPWM\_TCLK.CLK\_DIV 寄存器设置。信号来源比较器 0 输出和比较器 1 输出的滤波时钟分频由 MCPWM\_TCLK.CMP\_FLT\_CLKDIV 设置; 信号来源 MCPWM\_BKIN0 和 MCPWM\_BKIN1 的滤波时钟分频由 MCPWM\_TCLK.IO\_FLT\_CLKDIV 设置。

最后 FAIL0,FAIL1 会对信号进行 16 个滤波时钟的滤波, 即只有信号稳定时间超过 16 个滤波周期才能通过滤波器。即滤波宽度=滤波时钟周期\*16。

更多信息可以参考 FAIL 信号处理。

### 12.2.27 MCPWM\_PRT

无写保护的寄存器

地址: 0x4001\_1C74

复位值: 0x0

表 12-30 MCPWM\_PRT 寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRT															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	PRT	写入 0xDEAD, 解除 MCPWM 寄存器写保护; 写入其它值, MCPWM 寄存器进入写保护。此寄存器读出恒为 0。

### 12.2.28 MCPWM\_CNT

无写保护的寄存器

地址: 0x4001\_1C78

复位值: 0x0

表 12-31 MCPWM\_CNT 寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---



CNT
RW
0

位置	位名称	说明
[31:16]		未使用
[15:0]	CNT	向此寄存器写入，更改 MCPWM_CNT 的设定值，发生更新事件后，本寄存器加载到 MCPWM 实际运行系统的 CNT 中。 读出的数据为 MCPWM 实际运行系统中计数器的值。实际读出的计数范围为 0x8000-TH ~ 0x8000+TH

### 12.2.29 MCPWM\_SWAP

写保护的寄存器

地址：0x4001\_1C7C

复位值：0x0

表 12-32 MCPWM\_SWAP 寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															MCPWM_SWAP
															RW
															0

位置	位名称	说明
[31:1]		未使用
[0]	MCPWM_SWAP	向此寄存器写入 0x67 可将 BIT[0]写为 1，写其他值则将 BIT[0]写为 0

MCPWM\_SWAP 的值为 0 时，MCPWM 通道输出与 GPIO 关系如下：

表 12-33 MCPWM 默认输出表

MCPWM 输出顺序	GPIO 对应顺序
MCPWM_CH0P	P1.4
MCPWM_CH0N	P1.5
MCPWM_CH1P	P1.6
MCPWM_CH1N	P1.7
MCPWM_CH2P	P1.8



MCPWM_CH2N	P1.9
MCPWM_CH3P	P1.10
MCPWM_CH3N	P1.11

MCPWM\_SWAP 的值为 1 时，用于包含预驱芯片应用环境。在逻辑内部转换顺序，方便芯片与驱动芯片互连，一般应用上只需要三组 MCPWM 通道，因此仅转换三组的顺序。关系如下：

表 12-34 MCPWM 修改后的输出表

MCPWM 输出顺序	GPIO 对应顺序
MCPWM_CH0N	P1.4
MCPWM_CH1N	P1.5
MCPWM_CH2N	P1.6
MCPWM_CH0P	P1.7
MCPWM_CH1P	P1.8
MCPWM_CH2P	P1.9
MCPWM_CH3P	P1.10
MCPWM_CH3N	P1.11

### 12.2.30 MCPWM\_DTH00

写保护的寄存器

地址：0x4001\_1C80

复位值：0x0

表 12-35 MCPWM\_DTH00 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTH00															
RW															
0															

位置	位名称	说明
[31:10]		未使用
[9:0]	DTH00	MCPWM CH0 P 通道死区宽度控制寄存器，10bit 无符号数

### 12.2.31 MCPWM\_DTH01

写保护的寄存器

地址：0x4001\_1C84

复位值：0x0



表 12-36 MCPWM\_DTH01 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DTH01					
										RW					
										0					

位置	位名称	说明
[31:10]		未使用
[9:0]	DTH01	MCPWM CH0 N 通道死区宽度控制寄存器, 10bit 无符号数

### 12.2.32 MCPWM\_DTH10

写保护的寄存器

地址: 0x4001\_1C88

复位值: 0x0

表 12-37 MCPWM\_DTH10 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DTH10					
										RW					
										0					

位置	位名称	说明
[31:10]		未使用
[9:0]	DTH10	MCPWM CH1 P 通道死区宽度控制寄存器, 10bit 无符号数

### 12.2.33 MCPWM\_DTH11

写保护的寄存器

地址: 0x4001\_1C8C

复位值: 0x0

表 12-38 MCPWM\_DTH11 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DTH11					
										RW					
										0					

位置	位名称	说明
[31:10]		未使用



[9:0]	DTH11	MCPWM CH1 N 通道死区宽度控制寄存器，10bit 无符号数
-------	-------	------------------------------------

### 12.2.34 MCPWM\_DTH20

写保护的寄存器

地址：0x4001\_1C90

复位值：0x0

表 12-39 MCPWM\_DTH20 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTH20															
RW															
0															

位置	位名称	说明
[31:10]		未使用
[9:0]	DTH20	MCPWM CH2 P 通道死区宽度控制寄存器，10bit 无符号数

### 12.2.35 MCPWM\_DTH21

写保护的寄存器

地址：0x4001\_1C94

复位值：0x0

表 12-40 MCPWM\_DTH21 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTH21															
RW															
0															

位置	位名称	说明
[31:10]		未使用
[9:0]	DTH21	MCPWM CH2 N 通道死区宽度控制寄存器，10bit 无符号数

### 12.2.36 MCPWM\_DTH30

写保护的寄存器

地址：0x4001\_1C98



复位值: 0x0

表 12-41 MCPWM\_DTH30 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTH30															
RW															
0															

位置	位名称	说明
[31:10]		未使用
[9:0]	DTH30	MCPWM CH3 P 通道死区宽度控制寄存器, 10bit 无符号数

### 12.2.37 MCPWM\_DTH31

写保护的寄存器

地址: 0x4001\_1C9C

复位值: 0x0

表 12-42 MCPWM\_DTH31 配置寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTH31															
RW															
0															

位置	位名称	说明
[31:10]		未使用
[9:0]	DTH31	MCPWM CH3 N 通道死区宽度控制寄存器, 10bit 无符号数

## 13 UART

### 13.1 概述

通用异步收发传输器（Universal Asynchronous Receiver/Transmitter），通常称作 UART，是一种异步收发传输器。

UART 主要特征如下：

- 全双工工作
- 支持 7/8 位数据位
- 支持 1/2 停止位
- 支持奇/偶/无校验模式
- 带 1 字节发送缓存
- 带 1 字节接收缓存
- 支持 Multi-drop Slave/Master 模式

### 13.2 功能说明

#### 13.2.1 发送

UART 包括一个字节发送缓冲区，当发送缓冲区有数据时，UART 将发送缓冲区的数据加载，并通过 TX 发送出去。

完成加载后，产生发送缓冲区空中断，此时，用户可以往发送缓冲区填入下一个需要发送的字节，这样，发送完成后，UART 将加载这个字节进行发送。

完成发送后，会产生发送完成中断。

#### 13.2.2 接收

UART 包括一个字节的接收缓冲区，当完成一个字节的接收后，会产生接收中断，并将接收到字节存储到接收缓冲区，用户应当在 UART 接收完成下一个字节前完成此字节的读取，否则缓冲区会被写入新接收的字节。

#### 13.2.3 波特率配置

UART 输入时钟为系统主时钟，波特率通过两级分频实现。

波特率=UART 模块时钟/ (256\*DIVH+DIVL+1)

可以通过 SYS\_CLK\_DIV2 对 UART 模块时钟进行分频.

UART 模块时钟 =系统主时钟/(1+SYS\_CLK\_DIV2)



表 13-1 UART 波特率配置示例

UART 波特率	SYS_CLK_DIV2	UART_DIVH	UART_DIVL
300	0x0007	0x9C	0x3F
600	0x0003	0x9C	0x3F
1200	0x0001	0x9C	0x3F
2400	0x0000	0x9C	0x3F
4800	0x0000	0x4E	0x1F
9600	0x0000	0x27	0x0F
19200	0x0000	0x13	0x87
38400	0x0000	0x09	0xC3
43000	0x0000	0x08	0xB8
56000	0x0000	0x06	0xB1
57600	0x0000	0x06	0x82
115200	0x0000	0x03	0x40

注意，同一波特率，其配置系数可能不唯一。

#### 13.2.4 收发端口互换(TX/RX 互换)

UART 模块支持 Tx 与 Rx 端口互换。通过将 Tx 对应的 GPIO 配置为输入使能，Rx 对应的 GPIO 配置为输出使能，即可实现 Tx 与 Rx 端口的互换。此时，GPIO 第二功能仍选择为 UART，UART 本身配置无需修改。

此外，如果要使用一个 GPIO 同时作为 Tx 和 Rx，需要将 IO 分时复用为输入或输出，对应 Rx 或 Tx，即可实现单口半双工逻辑。

### 13.3 寄存器

#### 13.3.1 地址分配

UART0 与 UART1 实现完全相同。

UART0 基地址 0x40012800。

UART1 基地址 0x40012C00。

表 13-2 UARTx 地址分配列表

名称	偏移地址	说明
UARTx_CTRL	0x00	UART 控制寄存器
UARTx_DIVH	0x04	UART 波特率设置高字节寄存器
UARTx_DIVL	0x08	UART 波特率设置低字节寄存器
UARTx_BUFF	0x0C	UART 收发缓冲寄存器
UARTx_ADR	0x10	485 通信地址匹配寄存器
UARTx_STT	0x14	UART 状态寄存器
UARTx_IE	0x18	UART 中断使能寄存器



UARTx_IF	0x1C	UART 中断标志寄存器
UARTx_IOC	0x20	UART IO 控制

### 13.3.2 UARTx\_CTRL UARTx 控制寄存器

UART0\_CTRL 地址: 0x4001\_2800

UART1\_CTRL 地址: 0x4001\_2C00

复位值: 0x0

表 13-3 UARTx\_CTRL UARTx 控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									MDMASTER_BIT9	MD_EN	CK_EN	CK_TYPE	BIT_ORDER	STOP_LEN	DAT_LEN
									RW	RW	RW	RW	RW	RW	RW
									0	0	0	0	0	0	0

位置	位名称	说明
[31:8]		未使用
[6]	MDMASTER_BIT9	Multi-drop Master 模式时, 配置第 9-Bit 数据位的值。
[5]	MD_EN	使能 Multi-drop, 默认值为 0。 0: 关闭; 1: 开启
[4]	CK_EN	数据校验开关, 默认值为 0。 0: 关闭; 1: 开启
[3]	CK_TYPE	奇偶校验配置, 默认值为 0。 0: 偶校验 (EVEN); 1: 奇校验 (ODD)
[2]	BIT_ORDER	数据发送顺序配置, 默认值为 0。 0: LSB; 1: MSB
[1]	STOP_LEN	停止位长度配置, 默认值为 0。 0: 1-Bit; 1: 2-Bit
[0]	DAT_LEN	数据长度配置, 默认值为 0。 0: 8-Bit; 1: 7-Bit

### 13.3.3 UARTx\_DIVH UARTx 波特率设置高字节寄存器

UART0\_CTRL 地址: 0x4001\_2804

UART1\_CTRL 地址: 0x4001\_2C04

复位值: 0x0



表 13-4 UARTx\_DIVH UARTx 波特率设置高字节寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DIVH							
								RW							
								0							

位置	位名称	说明
[31:8]		未使用
[7:0]	DIVH	波特率设置高字节 BAUDRATE = 主时钟 / (1 + 256 * UARTx_DIVH + UARTx_DIVL)

### 13.3.4 UARTx\_DIVL UARTx 波特率设置低字节寄存器

UART0\_CTRL 地址: 0x4001\_2808

UART1\_CTRL 地址: 0x4001\_2C08

复位值: 0x0

表 13-5 UARTx\_DIVL UARTx 波特率设置低字节寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DIVL							
								RW							
								0							

位置	位名称	说明
[31:8]		未使用
[7:0]	DIVL	波特率设置低字节 BAUDRATE = 主时钟 / (1 + 256 * UARTx_DIVH + UARTx_DIVL)

### 13.3.5 UARTx\_BUFF UARTx 收发缓冲寄存器

表 13-6 UARTx\_BUFF UARTx 收发缓冲寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								BUFF							
								RW							
								0							

位置	位名称	说明
[31:8]		未使用
[7:0]	BUFF	写: 发送数据缓存; 读: 接收数据寄存器



UART 的 Tx\_buffer 和 Rx\_buffer 共享 UARTx\_BUFF。其中，Tx\_buffer 是只写的，Rx\_buffer 是只读的。因此，读 UARTx\_BUFF 寄存器是访问 UARTx\_RX\_BUFF，写 UARTx\_BUFF 寄存器是访问 UARTx\_TX\_BUFF。

### 13.3.6 UARTx\_ADR UARTx 地址匹配寄存器

UART0\_CTRL 地址：0x4001\_2810

UART1\_CTRL 地址：0x4001\_2C10

复位值：0x0

表 13-7 UARTx\_ADR UARTx 地址匹配寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ADR				
											RW				
											0				

位置	位名称	说明
[31:8]		未使用
[7:0]	ADR	用作 485 通信时的匹配地址

### 13.3.7 UARTx\_STT UARTx 状态寄存器

UART0\_CTRL 地址：0x4001\_2814

UART1\_CTRL 地址：0x4001\_2C14

复位值：0x0

表 13-8 UARTx\_STT UARTx 状态寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ADR_MATCH	TX_DONE	TX_BUF_EMPTY
													R	R	R
													0	1	1

位置	位名称	说明
[31:3]		未使用
[2]	ADR_MATCH	Multi-drop 模式下，地址匹配标志位。 1: 匹配; 0: 未匹配。



[1]	TX_DONE	发送完成标志位。 1: 完成; 0: 未完成。
[0]	TX_BUF_EMPTY	发送缓存状态位。 1: 空; 0: 非空。

### 13.3.8 UARTx\_IE UARTx 中断使能寄存器

UART0\_CTRL 地址: 0x4001\_2818

UART1\_CTRL 地址: 0x4001\_2C18

复位值: 0x0

表 13-9 UARTx\_IE UARTx 中断使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											CK_ERR_IE	STOP_ERR_IE	TX_BUF_EMPTY_IE	RX_DONE_IE	TX_DONE_IE
											RW	RW	RW	RW	RW
											0	0	0	0	0

位置	位名称	说明
[31:5]		未使用
[4]	CK_ERR_IE	校验错误中断开关, 默认值为 0。 0: 关闭; 1: 开启。
[3]	STOP_ERR_IE	停止位错误中断开关, 默认值为 0。 0: 关闭; 1: 开启。
[2]	TX_BUF_EMPTY_IE	发送缓冲区空中断开关, 默认值为 0。 0: 关闭; 1: 开启。
[1]	RX_DONE_IE	接收完成中断开关, 默认值为 0。 0: 关闭; 1: 开启。
[0]	TX_DONE_IE	发送完成中断开关, 默认值为 0。 0: 关闭; 1: 开启。

### 13.3.9 UARTx\_IF UARTx 中断标志寄存器

UART0\_CTRL 地址: 0x4001\_281C

UART1\_CTRL 地址: 0x4001\_2C1C

复位值: 0x0



表 13-10 UARTx\_IF UARTx 中断使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
												CK_ERR_IF	STOP_ERR_IF	TX_BUF_EMPTY_IF	RX_DONE_IF	TX_DONE_IF
												RW1C	RW1C	RW1C	RW1C	RW1C
												0	0	1	0	1

位置	位名称	说明
[31:5]		未使用
[4]	CK_ERR_IF	校验错误中断标志，高有效，写 1 清零。
[3]	STOP_ERR_IF	停止位错误中断标志，高有效，写 1 清零。
[2]	TX_BUF_EMPTY_IF	发送缓冲区空中断标志，高有效，写 1 清零。
[1]	RX_DONE_IF	接收完成中断标志，高有效，写 1 清零。
[0]	TX_DONE_IF	发送完成中断标志，高有效，写 1 清零。

上电后发送缓冲区空标志 TX\_BUF\_EMPTY\_IF 和发送完成中断标志 TX\_DONE\_IF 默认为 1。

中断标志写 1 清零，一般不建议用如下方式清零，因为|=是先读取中断标志，将对应位改为 1 再写入清零，如果同时有其他中断标志置位，会被一起清零，而这通常不是软件所期望的。例如，如下写法本意是清零 TX\_DONE\_IF，但如果同时 RX\_DONE\_IF 在写入执行前置 1 了，则软件先读取回 UARTx\_IF 值为 0x2，然后执行或操作 0x2|0x1=0x3，然后写入，同时对 RX\_DONE\_IF 和 TX\_DONE\_IF 进行了清零，可能导致 UART 少进入一次因接收数据产生的中断，从而少接收到一字节数据。

```
UARTx_IF|=0x1;
```

如果希望清零 TX\_DONE\_IF 标志位，应以如下方式，直接对 BIT0 写 1。

```
UARTx_IF=0x1;
```

### 13.3.10 UARTx\_IOC UARTx IO 控制寄存器

UART0\_CTRL 地址：0x4001\_2820

UART1\_CTRL 地址：0x4001\_2C20

复位值：0x0

表 13-11 UARTx\_IOC UARTxIO 控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TXD_INV	RXD_INV	
													RW	RW	
													0	0	



位置	位名称	说明
[31:2]		未使用
[1]	TXD_INV	TXD 输出极性使能开关，默认值为 0。 0: 正常输出; 1: 取反输出。 正常输出极性，意思是应用发送 1，硬件发送的是 1; 取反输出极性，意思是应用发送 1，硬件发送的是 0。
[0]	RXD_INV	RXD 输入极性使能开关，默认值为 0。 0: 正常输入; 1: 取反输入。 正常输入极性，意思是硬件接收 1，应用接收的是 1; 取反输入极性，意思是硬件接收 1，应用接收的是 0。

## 14 数字信号协处理器

### 14.1 概述

被开方数为 32 位无符号数，平方根为 16 位无符号数。

三角函数 Cordic 模块位宽为 16 位，Q15 定点数格式。

开方 8 个总线周期（96MHz）完成。

三角函数 16 个总线周期（96MHz）完成。

### 14.2 寄存器

#### 14.2.1 地址分配

数字信号协处理器寄存器在芯片中的基地址是 0x4001\_5000。

表 14-1 协处理器寄存器列表

名称	偏移	说明
DSP_SC	0x00	协处理器状态控制寄存器
DSP_THETA	0x04	协处理器 sin/cos 输入角度寄存器
DSP_XY	0x08	协处理器 arctan/module 计算输入坐标 XY 寄存器
	0x0C	保留
DSP_SIN	0x10	协处理器 sin/cos 计算结果 sin 寄存器
DSP_COS	0x14	协处理器 sin/cos 计算结果 cos 寄存器
DSP_MOD	0x18	协处理器 arctan 计算结果 $\sqrt{X^2+Y^2}$ 寄存器
DSP_ARCTAN	0x1C	协处理器 arctan 计算结果 $\arctan(Y/X)$ 角度寄存器
	0x20	保留
	0x24	保留
	0x28	保留
	0x2C	保留
DSP_RAD	0x30	协处理器 开方操作被开方数
DSP_SQRT	0x34	协处理器 开方操作平方根

#### 14.2.2 协处理器状态控制寄存器

地址：0x4001\_5000

复位值：0x0

表 14-2 DSP\_SC 协处理器状态控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---



	CORDIC_MODE	
	RW	
	0	

位置	位名称	说明
[31:3]	RESERVED	保留
[2]	CORDIC_MODE	CORDIC mode, 0: arctan, 1: sin/cos
[1:0]	RESERVED	保留

CORDIC\_MODE 位，MCU 通过总线访问 CORDIC 模块，sin/cos mode 和 arctan mode 的选择，CORDIC 模块计算 sin/cos 或 arctan 使用的是相同的硬件电路。因此在进行某一种计算前，应通过配置 DSP\_SC 计算器进行适当模式选择。CORDIC 模块在计算 sin/cos 时以角度 theta 为输入，计算并输出 sin/cos 结果；计算 arctan 时以坐标 x/y 为输入，计算并输出角度  $\theta = \arctan(y/x)$  和  $module = \sqrt{x^2+y^2}$ 。

### 14.2.3 协处理器 sin/cos 相关寄存器

Cordic 模块计算 sin/cos 和 arctan 使用的是相同的数据通路，因此通过 MCU 使用 cordic 模块进行 sin/cos 计算，需要先将 DSP\_SC[2] 写为 1，使 cordic 进入 sin/cos 模式。

#### 14.2.3.1 DSP\_THETA

地址：0x4001\_5004

复位值：0x0

表 14-3 协处理器 sin/cos 角度输入寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THETA															
RW															
0															

位置	位名称	说明
[31:16]		保留，读出时为符号扩展，即{16{DSP_THETA[15]}}
[15:0]	THETA	协处理器 sin/cos 输入角度寄存器

DSP\_THETA 为 16 位有符号定点数，表示范围  $(-32768 \sim 32767)$  对应  $(-\pi \sim \pi)$ 。

#### 14.2.3.2 DSP\_SIN

地址：0x4001\_5010

复位值：0x0



表 14-4 协处理器 sin/cos 正弦结果寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIN															
RO															
0															

位置	位名称	说明
[31:16]		保留，读出时为符号扩展，即{16{DSP_SIN[15]}}
[15:0]	SIN	协处理器 sin/cos 计算结果 sin 寄存器

DSP\_SIN 为 16 位有符号定点数，其中 1bit 符号位，1bit 整数位，14bit 小数位；表示范围 ( -1 ~ 1 ) 。

#### 14.2.3.3 DSP\_COS

地址：0x4001\_5014

复位值：0x0

表 14-5 协处理器 sin/cos 余弦结果寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COS															
RO															
0															

位置	位名称	说明
[31:16]		保留，读出时为符号扩展，即{16{DSP_COS[15]}}
[15:0]	COS	协处理器 sin/cos 计算结果 cos 寄存器

DSP\_COS 为 16 位有符号定点数，其中 1bit 符号位，1bit 整数位，14bit 小数位；表示范围 ( -1 ~ 1 ) 。

#### 14.2.4 协处理器 arctan 相关寄存器

##### 14.2.4.1 DSP\_XY

地址：0x4001\_5008

复位值：0x0

表 14-6 协处理器 arctan/module 坐标输入寄存器

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



W0															
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y															
W0															
0															

位置	位名称	说明
[31:16]	X	协处理器 arctan/module 计算输入坐标 X 寄存器
[15:0]	Y	协处理器 arctan/module 计算输入坐标 Y 寄存器

DSP\_XY 为 16 位有符号定点数, 其中 1bit 符号位, 15bit 整数位, 表示范围(-32768 ~ 32767)。

#### 14.2.4.2 DSP\_MOD

地址: 0x4001\_5018

复位值: 0x0

表 14-7 协处理器 arctan 角度结果 arctan(Y/X) 角度寄存器

MOD															
RO															
0															

位置	位名称	说明
[31:16]		保留, 读恒为 0
[15:0]	MOD	协处理器 arctan 计算结果 $\sqrt{X^2+Y^2}$ 寄存器

DSP\_MOD 为 16 位有符号定点数, 其中 1bit 符号位, 15bit 整数位, 表示范围(-32768 ~ 32767)。

需要注意的是, 当计算 arctan(Y/X)时, 需要保证  $\sqrt{X^2+Y^2}$  也不超过 32767, 因此一般建议 abs(DSP\_X)和 abs(DSP\_Y)不超过  $2^{14}$  为宜。

#### 14.2.4.3 DSP\_ARCTAN

地址: 0x4001\_501C

复位值: 0x0

表 14-8 协处理器 arctan 角度结果 arctan(Y/X) 角度寄存器

ARCTAN															
RO															



0
---

位置	位名称	说明
[31:16]		保留，读出时为符号扩展，即{16{DSP_ARCTAN[15]}}
[15:0]	ARCTAN	协处理器 arctan 计算结果 arctan(Y/X) 角度寄存器

DSP\_ARCTAN 为 16 位有符号定点数，表示范围 (-32768 ~ 32767) 对应 (- $\pi$  ~  $\pi$ )。

需要注意的是，当计算 arctan(Y/X)时，需要保证 sqrt(X<sup>2</sup>+Y<sup>2</sup>)也不超过 32767，因此一般建议 abs(X)和 abs(Y)不超过 2<sup>14</sup> 为宜。

### 14.2.5 协处理器开方相关寄存器

#### 14.2.5.1 DSP\_RAD

地址：0x4001\_5030

复位值：0x0

表 14-9 协处理器被开方数寄存器

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

位置	位名称	说明
[31:0]	RAD	协处理器被开方数寄存器

#### 14.2.5.2 DSP\_SQRT

地址：0x4001\_5024

复位值：0x0

表 14-10 协处理器平方根寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQRT															
RO															
0															



位置	位名称	说明
[31:16]		保留，读出时恒为 0
[15:0]	SQRT	协处理器平方根寄存器

向协处理器写入被开方数；写入被开方数可以触发一次开方操作，32 位开方需要 8 周期完成，期间读取开方结果 DSP\_SQRT 会使 MCU 进入等待状态，等待开方计算完成并通过总线返回计算结果。

## 15 I2C

### 15.1 概述

I2C 总线接口连接微控制器和串行 I2C 总线。它提供多主机功能，控制所有 I2C 总线特定的时序、协议、仲裁和定时。支持标准和快速两种模式。

### 15.2 主要特性

- 多主机功能：该模块既可做主设备也可做从设备。
- I2C 主设备功能：产生时钟、START 和 STOP 事件。
- I2C 从设备功能：可编程的 I2C 硬件地址比较（仅支持 7 位硬件地址）、停止位检测。
- 根据系统分频，实现不同的通讯速度。
- 状态标志：发送器/接收器模式标志、字节发送结束标志、I2C 总线忙标志。
- 错误标志：主模式时的仲裁丢失、地址/数据传输后的应答(ACK)错误、检测到错位的起始或停止条件。
- 一个中断向量，包含五个中断源：总线错误中断源、完成中断源、NACK 中断源、硬件地址匹配中断源和传输完成中断源。

### 15.3 功能描述

#### 15.3.1 功能框图

本接口采用同步串行设计，实现 MCU 同外部设备之间的 I2C 传输。支持轮询和中断方式获得传输状态信息。本接口的主要功能模块如下图所示。

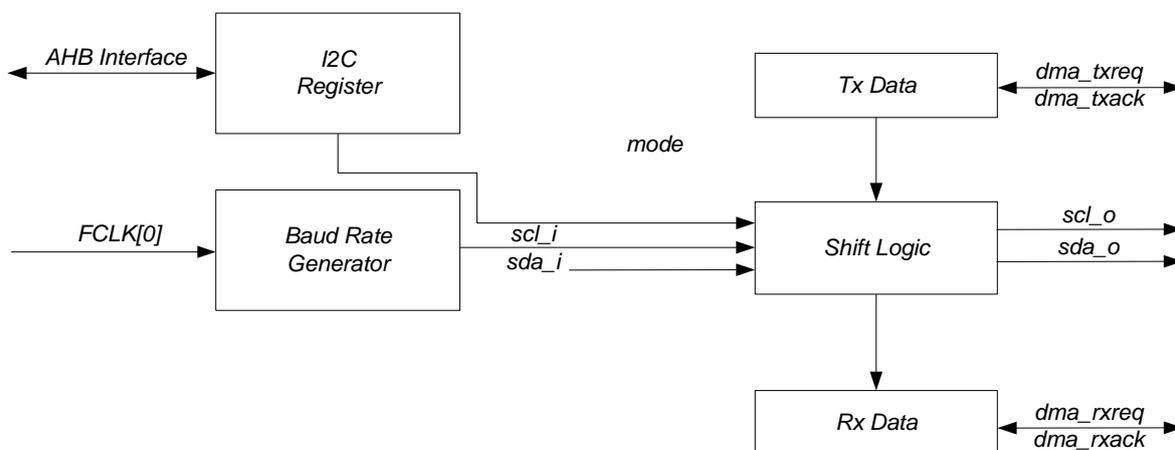


图 15-1 I2C 模块顶层功能框图

I2C 接口同外界通讯只有 SCL 和 SDA 两根信号线。SDA 为双向复用信号线，受到 sda\_oe 控制。模块级，I2C 接口信号包括，scl\_i, sda\_i, scl\_o, sda\_o 和 sda\_oe。

scl\_i: 时钟信号。当 I2C 接口配置为从模式时，此为 I2C 总线的时钟输入信号。

sda\_i: 数据信号。当 I2C 接口接收数据时（无论主模式还是从模式），此为 I2C 总线的数据输入信号。

scl\_o: 时钟信号。当 I2C 接口配置为主模式时，此为 I2C 总线的时钟输出信号。

sda\_o: 数据信号。当 I2C 接口发送数据时（无论主模式还是从模式），此为 I2C 总线的数据输出信号。

sda\_oe: 数据使能信号。当 sda\_o 输出时，sda\_oe 有效；当 sda\_i 输入时，sda\_oe 无效。

### 15.3.2 功能说明

I2C 模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。可以开启或禁止中断。接口通过数据引脚(SDA)和时钟引脚(SCL)连接到 I2C 总线。

#### 15.3.2.1 模式选择

接口可以下述 4 种模式中的一种运行：

- 从发送模式
- 从接收模式
- 主发送模式
- 主接收模式

I2C 接口默认主从均不使能。接口根据配置情况，进入主模式或者从模式。当仲裁丢失或产生停止信号时，主模式自动释放总线并产生相应异常中断。允许多主机功能。

主模式时，I2C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。起始条件和停止条件都是在主模式下由软件控制产生。

从模式时，I2C 接口能识别它自己的地址(7 位)。软件能够控制开启或禁止硬件地址比较功能，硬件地址比较功能可降低 MCU 的负担。只有地址匹配才通知 MCU 进行相关处理。

数据和地址按 8 位/字节进行传输，高位在前。跟在起始条件后的 1 个字节是地址。地址只在主模式发送。

在一个字节传输的 8 个时钟后的第 9 个时钟期间,接收器必须回送一个应答位(ACK)给发送器。

软件可以开启或禁止应答(ACK)，并可以设置 I2C 接口的地址。

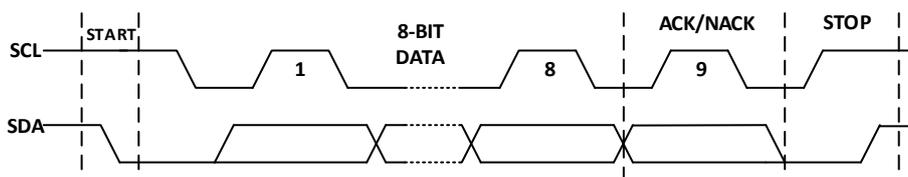


图 15-2 基本 I2C 传输时序图



一般情况下，一次传输一个字节（可反复单次传输，需软件介入提供数据）。上述所有模式，遵循如下基本原则：

- 单字节发送，中断将在 8-bit 数据发送完毕且收到响应后（ACK/NACK 均可）产生。
- 单字节接收，中断将在 8-bit 数据接收完毕后产生。
- 当 I2C 接口配置为主模式时，检测到错误后，I2C 接口会主动释放总线，恢复到起始状态并产生中断信号。

### 15.3.2.2 从模式

默认情况下，I2C 接口主模式和从模式均关闭。若工作在从模式，需使能从模式。为了产生正确的时序，必须在系统寄存器 CLK\_DIV0 中设定 I2C 接口的工作时钟。

- 从模式下，I2C 接口时刻在监控总线上的信号。一旦检测到起始条件，其将保存地址位数据和读写位数据。
- 从模式下，若硬件地址匹配功能开启，只有地址匹配的情况下，才会产生中断，通知 MCU 进行后续处理。若没有开启，每次收到地址及读写位数据，都将产生中断。
- 从模式接收。每次收到一个字节的的数据后，产生中断，此时 I2C 接口可拉低 SCL，直至中断完成，继续后续操作。
- 从模式发送。每次发送一个字节完毕后且收到响应（ACK/NACK），产生中断，此时 I2C 接口可拉低 SCL，直至中断完成，继续后续操作。

#### 15.3.2.2.1 从模式传输

每次传输完一个字节的的数据后，硬件将产生中断，软件判断是否还要继续传输。图 15-3 为从模式传输的总线示意图。从图可知，流程如下：

- 地址匹配，产生地址匹配中断，准备开始传输。
- 若是从接收，一个字节接收完毕后，产生中断，软件判断是否继续接收，返回 ACK/NACK 响应。
- 若是从发送，一个字节发送完毕后，等待响应（ACK/NACK），产生中断，根据响应判断后续操作。
- 获得总线 STOP 事件，本次传输完成。

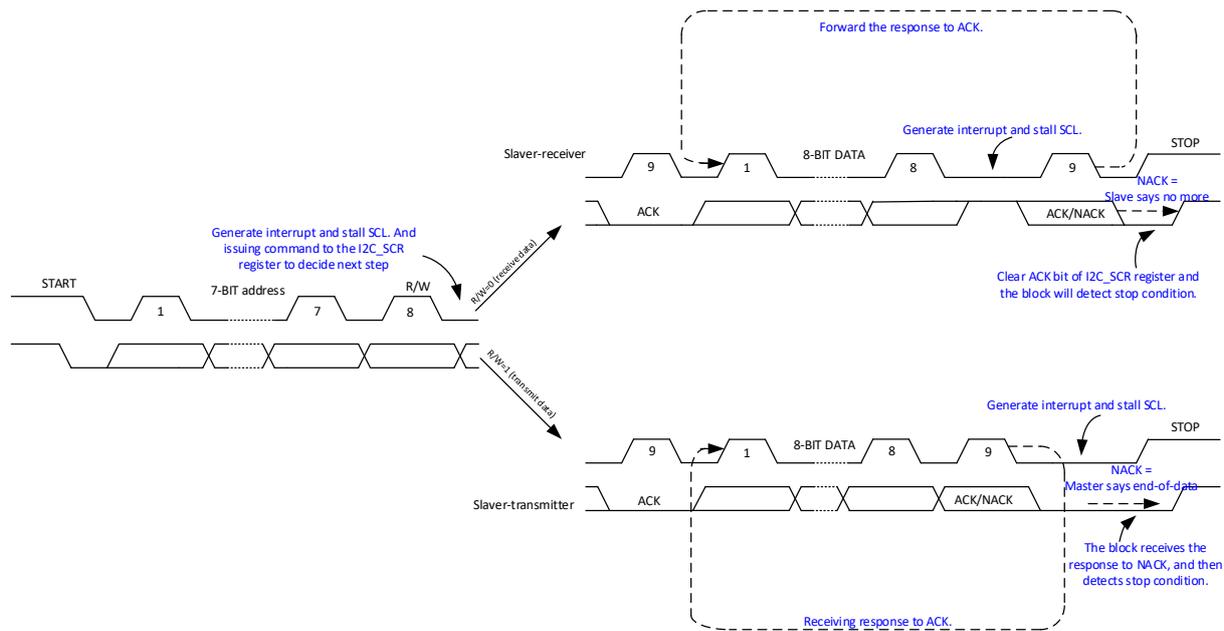


图 15-3 从模式传输示意图

### 15.3.2.2.2 从模式发送

地址匹配后，将数据从 I2C\_DATA 寄存器经由内部移位寄存器发送到 SDA 线上。在 I2C\_DATA 数据没有准备好之前，从设备可拉底 SCL，直到待发送数据已写入 I2C\_DATA 寄存器。I2C 接口在发送完毕每个字节后都执行下列操作：

- 如果接收到 ACK 位，装载下一个字节数据，继续传输。装载过程中，可拉底 SCL。
- 如果接收到 NACK 位，停止下一个字节的装载。
- 等待 STOP 事件，停止本次传输。

### 15.3.2.2.3 从模式单字节接收

地址匹配后，将通过内部移位寄存器从 SDA 线接收到的数据存入 I2C\_DATA 寄存器。I2C 接口在接收到每个字节后都执行下列操作：

- 如果设置了 ACK 位，在一个字节接收完毕后，产生一个 ACK 应答脉冲。
- 如果清除了 ACK 位，在一个字节接收完毕后，产生一个 NACK 应答脉冲。
- 等待 STOP 事件，结束本次传输。

### 15.3.2.3 主模式

默认情况下，I2C 接口主模式和从模式均关闭。若工作在主模式，需使能主模式。为了产生正确的时序，必须在系统寄存器 CLK\_DIV0 中设定 I2C 接口的工作时钟。



I2C 接口执行主模式传输之前，需要判断总线是否空闲。可读取 I2C\_MSCR 寄存器的 BIT3，查询当前总线状态。若总线处于忙的状态，可以开启 I2C 中断，通过收到 STOP 中断事件判断总线是否空闲下来。只有空闲状态下，才能正常发送 START 状态，以及后续的数据。

### 15.3.2.3.1 主模式传输

每次传输完一个字节的的数据后，将产生中断判断是否还要继续传输。下图为主模式传输的总线示意图。从图可知，流程如下：

- 判断总线是否空闲，若空闲，准备开始传输。
- 首先，发送从地址，若地址匹配，才继续后续传输，否则停止。
- 若是接收模式，一个字节接收完毕后，产生中断，软件判断是否继续接收，返回 ACK/NACK 响应。
- 若是发送模式，一个字节发送完毕后，等待响应 (ACK/NACK)，产生中断，根据响应判断后续操作。
- 发送总线 STOP 事件，本次传输完成。

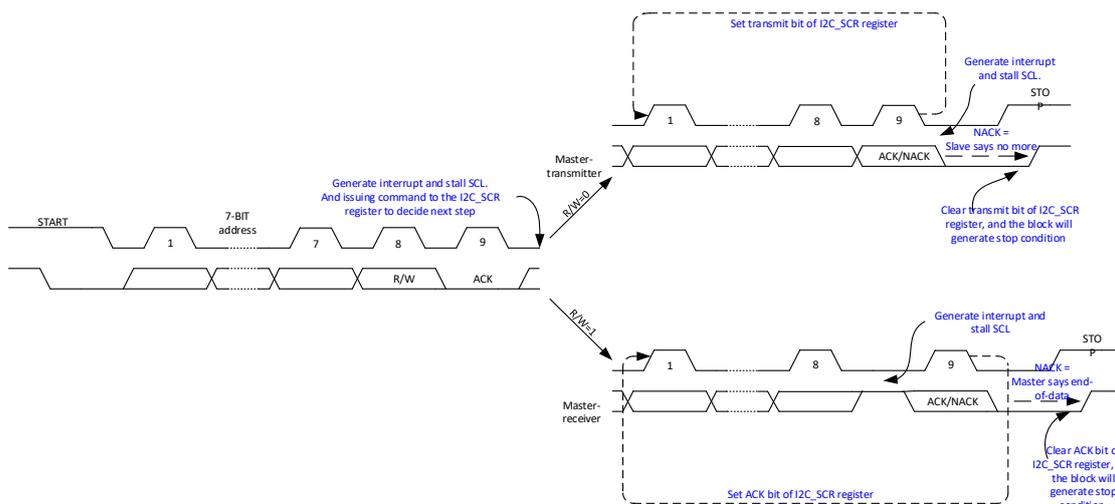


图 15-4 主模式下单字节传输示意图

### 15.3.2.3.2 主模式发送

开始传输后，I2C 接口将字节从 I2C\_DATA 寄存器经由内部移位寄存器发送到 SDA 线上，从地址也通过 I2C\_DATA 发送出去。在 I2C\_DATA 数据没有准备好之前，主设备可不产生 SCL 时钟信号，直到待发送数据已写入 I2C\_DATA 寄存器。I2C 接口在发送完毕每个字节后都执行下列操作：

- 如果接收到 ACK 位，装载下一个字节数据，继续传输。装载过程中，可拉底 SCL。
- 如果接收到 NACK 位，停止下一个字节的装载。



- 若主设备，完成传输。无论接收的是 ACK/NACK，都停止后续传输。
- 产生 STOP 事件，结束本次传输。

### 15.3.2.3.3 主模式接收

开始传输后，I2C 接口将通过内部移位寄存器从 SDA 线接收到的数据存入 I2C\_DATA 寄存器，从地址也通过 I2C\_DATA 发送出去。I2C 接口在接收到每个字节后都执行下列操作：

- 如果设置了 ACK 位，在一个字节接收完毕后，产生一个 ACK 应答脉冲。
- 如果清除了 ACK 位，在一个字节接收完毕后，产生一个 NACK 应答脉冲。
- 若主设备，完成传输。无论发送的是 ACK/NACK，都停止后续传输。
- 产生 STOP 事件，结束本次传输。

### 15.3.2.4 I2C 总线异常处理

在一个地址或数据字节传输期间，当 I2C 接口检测到一个外部的停止或起始条件则产生总线错误。一般而言，产生总线错误是由于总线上有干扰、某些 I2C 设备没有同步于本 I2C 网络自行发送了 START 事件/STOP 事件。根据 I2C 协议规定，发生总线错误的时候，在收到 START 事件/STOP 事件后要重置本 I2C 设备的接口逻辑。对于从设备而言，这个操作是没有问题的；对于主设备而言，总线错误强行要求其释放总线并重置其 I2C 接口逻辑。因为主设备是不响应外部 START 和 STOP 事件的，发生总线错误后，需要中断处理函数处理本次异常，并指导主设备继续监视总线情况，以便后续执行 I2C 总线传输。

本 I2C 接口。主模式下，总线错误可被检测到同时总线错误中断也会产生；从模式下，总线错误将触发地址数据被接收，同时让 I2C 接口恢复空闲状态并产生中断。

### 15.3.2.5 中断处理

I2C 接口包含三种类型的中断事件，分别是：数据传输完成事件，总线错误事件、STOP 事件、NACK 事件和硬件地址匹配事件。

- 数据完成事件。当前数据传输完成，高电平有效，对 I2C\_SCR.Done 写 0 清除。
- 总线错误事件。传输过程中，总线产生错误的 START 事件/STOP 事件，高电平有效，对 I2C\_SCR.STT\_ERR 写 0 清除。
- STOP 事件。当前数据传输完成，主设备发送 STOP 事件，从设备收到 STOP 事件并产生相应中断。高电平有效，对 I2C\_SCR.STOP\_EVT 写 0 清除。
- NACK 事件。发送端接收到 NACK 响应，表明接收端无法继续后续传输。高电平有效，对 I2C\_SCR.RX\_ACK 写 0 清除。
- 硬件地址匹配事件。从模式下接收到的地址同本设备地址匹配，产生相应中断。高电平有效，对 I2C\_SCR.ADDR\_DATA 写 0 清除。

## 15.3.2.6 通讯速度设置

I2C 接口的工作时钟来自系统时钟的分频，分频寄存器为 SYS 模块的 CLK\_DIV0。

I2C 接口采用同步设计，需要对外部设备的信号进行同步采样，同步时钟为 I2C 接口工作时钟。

- I2C 模块工作时钟频率 = 系统频率 / (CLK\_DIV0 + 1)
- I2C 模块数据信号(SDA)和时钟信号(SCL)的时钟频率= I2C 模块工作时钟频率 / 17。
- I2C 波特率 = I2C 模块工作时钟频率 / 17

## 15.4 寄存器

## 15.4.1 地址分配

I2C 模块寄存器的基地址是 0x4001\_0400。

表 15-1 I2C 寄存器地址分配表

名称	偏移	说明
I2C_ADDR	0x00	I2C 地址寄存器
I2C_CFG	0x04	I2C 配置寄存器
I2C_SCR	0x08	I2C 状态寄存器
I2C_DATA	0x0C	I2C 数据寄存器
I2C_MSCR	0x10	I2C 主模式寄存器
I2C_BCR	0x14	I2C 传输控制寄存器

## 15.4.2 I2C\_ADDR 地址寄存器

地址：0x4001\_0400

复位值：0x0

表 15-2 I2C\_ADDR 地址寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ADDR_CMP	ADDR						
								RW	RW						
								0	0						

位置	位名称	说明
[31:8]		未使用
[7]	ADDR_CMP	I2C 硬件地址比较使能开关，默认值为 0。 0：关闭 1：开启
[6:0]	ADDR	仅用于从模式下，I2C 设备硬件地址。主模式下，从设备地址写入 I2C_DATA



		寄存器。
--	--	------

### 15.4.3 I2C\_CFG 系统控制寄存器

地址：0x4001\_0404

复位值：0x0

表 15-3 I2C\_CFG 系统控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								IE	TC_IE	BUS_ERR_IE	STOP_IE				MST_MODE	SLV_MODE
								RW	RW	RW	RW				RW	RW
								0	0	0	0				0	0

位置	位名称	说明
[31:8]		未使用
[7]	IE	I2C 中断使能信号。默认值为 0。 1: 使能 I2C 中断 0: 关闭 I2C 中断
[6]	TC_IE	I2C 数据传输完成中断使能信号。默认值为 0。 1: 使能此中断源 0: 屏蔽此中断源
[5]	BUS_ERR_IE	I2C 总线错误事件中断使能信号。默认值为 0。 1: 使能此中断源 0: 屏蔽此中断源
[4]	STOP_IE	I2CSTOP 事件中断使能信号。默认值为 0。 1: 使能此中断源 0: 屏蔽此中断源
[3:2]		未使用
[1]	MST_MODE	I2C 主模式使能信号。默认值为 0。 1: 使能主模式 0: 关闭主模式
[0]	SLV_MODE	I2C 从模式使能信号。默认值为 0。 1: 使能从模式 0: 关闭从模式

### 15.4.4 I2C\_SCR 状态控制寄存器

地址：0x4001\_0408

复位值：0x0

表 15-4 I2C\_SCR 状态控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---



	STT_ERR	LOST_ARB	STOP_EVT	BYTE_CMPLT	ADDR_DATA	DATA_DIR	RX_ACK	DONE
	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0	0	0	0	0

位置	位名称	说明
[31:8]		未使用
[7]	STT_ERR	总线错误状态标志位，用于主模式发送/主模式接收，写 0 清除。 0：无 START/STOP 总线错误 1：有 START/STOP 总线错误
[6]	LOST_ARB	总线仲裁丢失状态标志位，用于主模式发送/主模式接收，发生总线仲裁丢失事件将此位置 1，无中断事件产生，在字节完成中断中需查此位。总线上任何 START 事件将导致硬件清除此位。 0：无总线仲裁丢失错误发生 1：有总线仲裁丢失错误发生
[5]	STOP_EVT	STOP 事件状态标志位，用于主模式发送/从模式发送/主模式接收/从模式接收。写 0 清除。 0：无 STOP 事件 1：有 STOP 事件
[4]	BYTE_CMPLT	ACK 控制位，用于主模式接收/从模式接收。发送方发送完毕当前字节，接收方对此的响应。若是发送方，此位保留 0 值。接收方，根据实际情况配置。 0：字节发送完成，返回 NACK 回应，表示接收方不能接收更多数据 1：字节发送完成，返回 ACK 回应，表示接收方可以继续接收数据
[3]	ADDR_DATA	地址数据标志位，用于主模式发送/从模式发送/主模式接收/从模式接收。START 后，第一个字节为地址数据，此位是一个提示位。写 0 清除。 0：当前传输的数据非地址数据。 1：当前传输的数据是地址数据。
[2]	DATA_DIR	发送或接收控制位，主模式发送/从模式发送，此位置 1，触发发送，硬件自动清零；主模式接收/从模式接收，此位置 0，等待接收。 0：接收 1：触发发送
[1]	RX_ACK	接收响应标志位，用于主模式发送/从模式发送，告知发送方，接收方的反馈。发送方收到反馈后，对该位执行清零操作。 0：本 I2C 接口发送数据，接收到 ACK 响应。 1：本 I2C 接口发送数据，接收到 NACK 响应。
[0]	Done	传输完成状态标志位，用于主模式发送/从模式发送/主模式接收/从模式接收。写 0 清除。 0：传输未完成 1：传输已完成

一般, 进入中断后, 需读取 I2C\_SCR 寄存器, 获得当前 I2C 总线状态及当前传输处于什么阶段; 然后, 对 I2C\_SCR 进行写操作, 写入不同的值, 软件通知硬件下一步如何处理。

#### 15.4.5 I2C\_DATA 数据寄存器

地址: 0x4001\_040C

复位值: 0x0

表 15-5 I2C\_DATA 数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
RW															
0															

位置	位名称	说明
[31:8]		未使用
[7:0]	DATA	数据寄存器, 用于主模式发送/从模式发送/主模式接收/从模式接收。发送方, 写入发送数据; 接收方, 读取接收数据。注意, 地址数据也是数据, 主模式只能将要发送地址数据写入此寄存器。

#### 15.4.6 I2C\_MSCR 主模式寄存器

地址: 0x4001\_0410

复位值: 0x0

表 15-6 I2C\_MSCR 主模式寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												BUSY	MST_CHECK	RESTART	START
												RW	RW	RW	RW
												0	0	0	0

位置	位名称	说明
[31:4]		未使用
[3]	BUSY	I2C 总线, 闲忙状态。 0: 检测到 STOP 事件, 空闲。 1: 检测到 START 事件, 忙碌。
[2]	MST_CHECK	主模式争抢总线标志位。争抢到总线, 置 1; STOP 事件或者发生总线冲突本模块释放总线, 置 0。
[1]	RESTART	再次触发 START 事件, 写 1 有效。发送 START 完毕, 硬件清 0。I2C_CFG[1]置 1, 才能实现写 1 操作。
[0]	START	触发 START 事件并发送地址数据至总线, 写 1 有效。I2C_CFG[1]置 1,



	才能实现写 1 操作。
--	-------------

#### 15.4.7 I2C\_BCR I2C 传输控制寄存器

地址：0x4001\_0414

复位值：0x0

表 15-7 I2C\_BCR 传输控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								NACK	ADDR_CMP						
								RW	RW						
								0	0						

位置	位名称	说明
[31:8]		未使用
[7]	NACK_IE	I2C 传输，NACK 事件中断使能信号。 0：屏蔽此中断源 1：使能此中断源
[6]	ADDR_CMP_IE	I2C 传输，硬件地址匹配中断使能信号。 0：屏蔽此中断源 1：使能此中断源
[5:0]		保留

## 16 SPI

### 16.1 概述

SPI 接口主要使用在，外部设计采用 SPI 协议的应用场景下。SPI 的工作模式软件可选，默认为 SPI Motorola 模式。SPI 接口支持全双工传输和半双工传输。当接口配置为 Master 模式时，可发送时钟信号供外部 Slave 设备使用。

### 16.2 主要特性

- 支持 Master 和 Slave 操作
- 全双工传输，可根据应用情况，使用 3 根或者 4 根信号线
- 支持半双工传输，可根据应用情况，使用 2 根信号线
- 可编程的时钟极性和相位
- 可编程的数据顺序：MSB 或 LSB
- 最快传输速度为系统最高时钟频率的 1/8
- 片选信号均可选。Master 模式下，片选信号可以软件控制也可以硬件产生；Slave 模式下，片选信号可以恒定有效，也可以来自外界设备
- 无本地 FIFO。包含溢出检测和片选信号异常检测

### 16.3 功能描述

#### 16.3.1 功能框图

本接口采用同步串行设计，实现 MCU 同外部设备之间的 SPI 传输。支持轮询和中断方式获得传输状态信息。本接口的主要功能模块如下图所示。

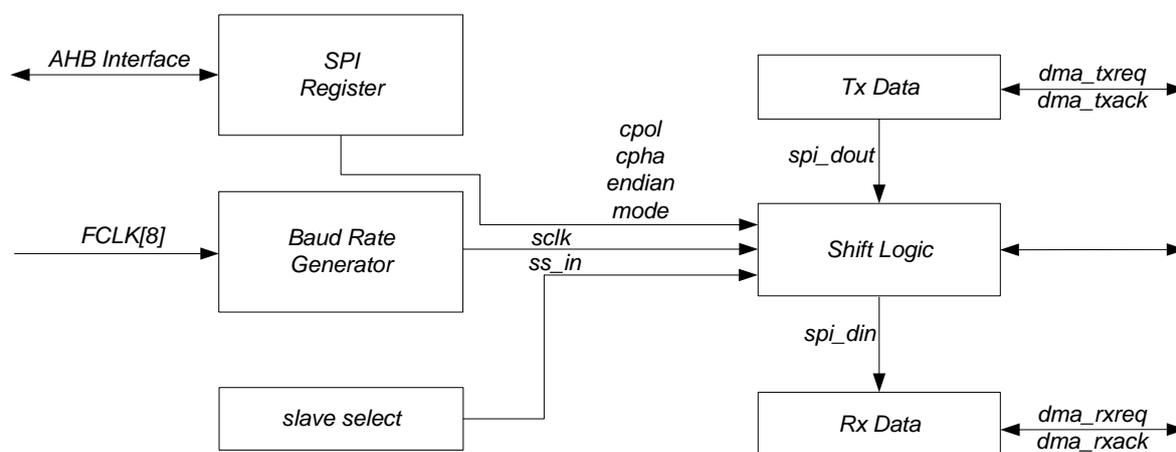


图 16-1 SPI 模块结构框图



接口信号包括, spi\_din, spi\_dout, sclk\_in, sclk\_out, ss\_in 和 ss\_out。

**spi\_din**: 接口接收的数据信号。同 SPI 协议比较, 当接口配置为 Master 模式时, 其等效为 MISO; 当接口配置为 Slave 模式时, 其等效为 MOSI。

**spi\_dout**: 接口发送的数据信号。同 SPI 协议比较, 当接口配置为 Master 模式时, 其等效为 MOSI; 当接口配置为 Slave 模式时, 其等效为 MISO。

**sclk\_in**: 接口接收的时钟信号。此时, 接口的工作模式为 Slave。非 Slave 模式下, 此信号输入无效。

**sclk\_out**: 接口发送的时钟信号。此时, 接口的工作模式为 Master, 非 Master 模式下, 此信号输出恒定为 0。

**ss\_in**: 接口接收的片选信号。此时, 接口的工作模式为 Slave。非 Slave 模式下, 此信号输入无效。

**ss\_out**: 接口发送的片选信号。此时, 接口的工作模式为 Master。非 Master 模式下, 此信号输出恒定为 1。

## 16.3.2 功能说明

### 16.3.2.1 全双工模式

默认情况下, SPI 接口配置为全双工模式。此时, 数据传输需要两根数据线。数据信号的变化, 发生在时钟信号的边沿, 即同步于时钟信号。

接口为 Master 模式时:

- spi\_din 为数据输入, 接外部 Slave 设备的 MISO
- spi\_dout 为数据输出, 接外部 Slave 设备的 MOSI
- spi\_ss\_out 为片选信号, 根据应用情况选择是使用该信号还是软件控制其它 GPIO 实现

接口为 Slave 模式时:

- spi\_din 为数据输入, 接外部 Master 设备的 MOSI
- spi\_dout 为数据输出, 接外部 Master 设备的 MISO
- spi\_ss\_in 为片选信号, 根据应用情况是使用该信号还是片选恒有效

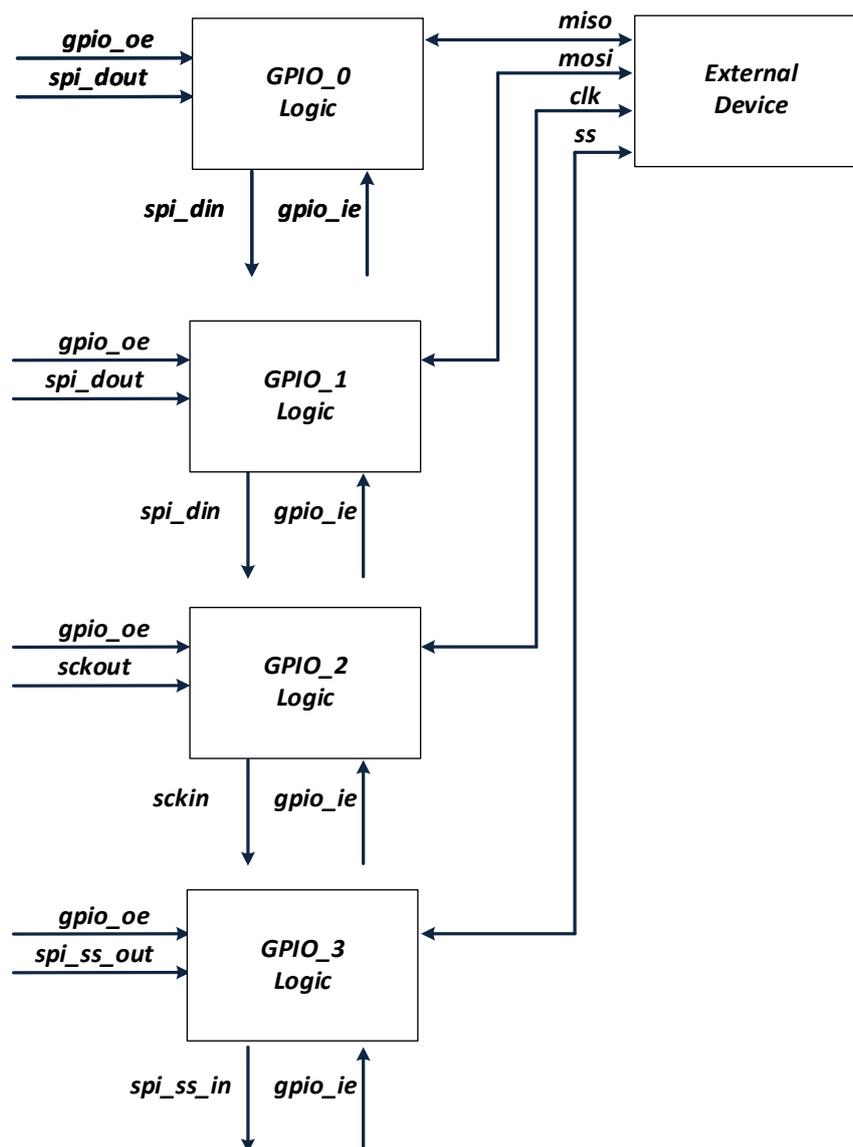


图 16-2 SPI 接口全双工模式互连框图

从上图可知，GPIO 若配置为输出，则 SPI 接口可发送数据；GPIO 若配置为输入，则 SPI 接口可接收数据。SPI\_DI 和 SPI\_DO 支持互换。

### 16.3.2.2 半双工模式

SPI 接口可配置为半双工模式。此时，数据传输只需要一根数据线。数据信号的变化，发生在时钟信号的边沿，即同步于时钟信号。一次传输只能是一个方向的，要不就是发送，要不就是接收。

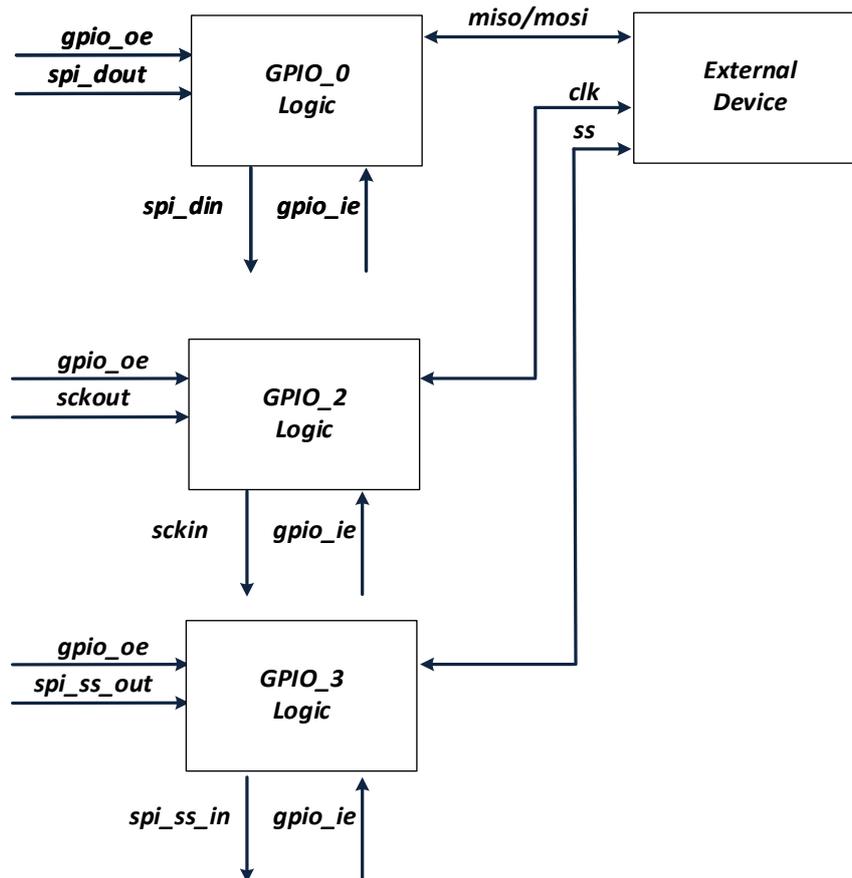


图 16-3 SPI 接口半双工模式互连框图

注意，上图中若本接口做 Master，则 clk 为本接口的输出信号；若本接口做 Slave，则 clk 为本接口的输入信号。

#### 仅发送

SPI\_CFG.DUPLEX 配置为 2，半双工发送模式有效。此时，本接口只能发送数据。GPIO\_0 的 oe 使能，发送 spi\_dout 数据到外界；GPIO\_0 的 ie 关闭，spi\_din 恒定输入为 0。此模式下，支持 Master/Slave 模式下的发送。

#### 仅接收

SPI\_CFG.DUPLEX 配置为 3，半双工接收模式有效。此时，本接口只能接收数据。GPIO\_0 的 oe 关闭，spi\_dout 无法发送数据到外界；GPIO\_0 的 ie 开启，spi\_din 接收来自外部的数据。此模式下，支持 Master/Slave 模式下的接收。

注意，全双工下是两个 GPIO 用于数据传输，半双工下可从中任意选一个 GPIO 用于数据传输。

### 16.3.2.3 片选信号

本接口做 Slave 模式时，片选信号可选，SPI\_CFG.CS 决定片选来源。ss 为 Master 设备发出的选通使能信号，低电平有效。



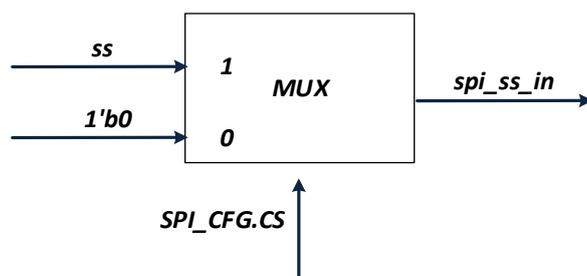


图 16-4 SPI 模块 Slave 模式片选信号选择

本接口做 Master 模式时，片选信号亦可选。模块硬件产生了标准的片选信号，实际应用可屏蔽此信号通过软件操作额外的 GPIO 实现。

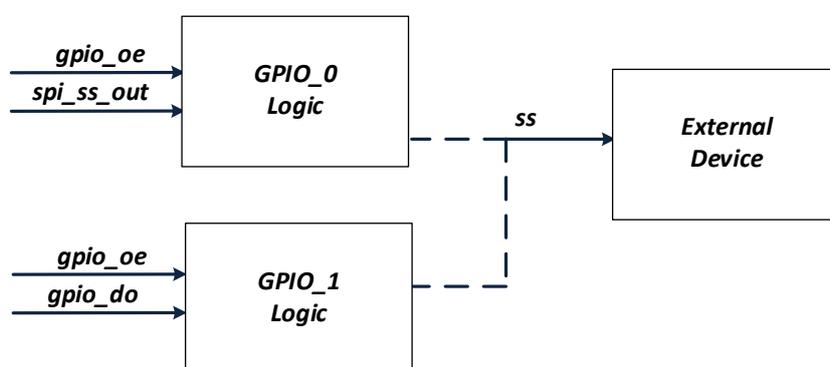


图 16-5 SPI 模块 Master 模式片选信号选择

注意，图 16-5 虚线仅表示不确定。若使用 `spi_ss_out` 为 `ss` 的源头，那么将 GPIO\_0 同外界设备互连；若使用软件操作 GPIO 的方式，那么可将 GPIO\_1 同外界设备互连。

#### 16.3.2.4 通讯格式

在 SPI 通讯过程中，发送或者接收操作均是基于 SPI 时钟的。通讯格式受到 `SPI_CFG.SAMPLE` 和 `SPI_CLK_POL` 控制。`SPI_CFG.SAMPLE` 为 Phase 控制位，`SPI_CLK_POL` 为 Polarity 控制位。

Polarity 控制了 SPI 时钟信号在默认情况下的电平状态。Polarity 为 0 时，默认时钟电平为低电平；Polarity 为 1 时，默认电平为高电平。

Phase 控制了 SPI 数据的发送/接收时刻。Phase 为 0 时，时钟从默认电平到第一个跳变边沿为采样数据时刻，Phase 为 1 时，时钟从默认电平到第一个跳变边沿为发送数据时刻。

#### 16.3.2.5 数据格式及长度

SPI 数据传输格式分成两种：MSB 和 LSB。数据传输格式受到 `SPI_CFG.ENDIAN` 控制。注意，在数据传输过程中硬件自动实现传输格式的转换，无需软件转换。

SPI 数据长度可配，范围是从 8-Bit 到 16-Bit。`SPI_SIZE.BITSIZE` 控制长度。



### 16.3.2.6 传输

一次只能发送/接收一个 `SPI_SIZE.BITSIZE` 长度的数据,每次完成后需要通过中断或者轮询的方式判断传输是否完成。无论是主模式还是从模式,写 `SPI_TX_DATA` 寄存器,才能触发传输。主模式为主动发送,从模式为加载数据到发送队列等待主模式发出时钟信号,开始传输。推荐软件配置流程如下:

- 初始化 GPIO 模块,将 SPI 复用的 GPIO 配置完毕。
- 初始化 SPI 接口, `SPI_IE/SPI_CFG/SPI_BAUD/SPI_SIZE` 等寄存器配置完毕。
- MCU 对 `SPI_TX_DATA` 寄存器执行写操作,触发 SPI 接口进入发送流程。从模式下,数据加载到内部状态机等待主设备发起读取操作;主模式下,触发发送。

支持连续发送,由 `SPI_BAUD.TRANS_MODE` 控制。主要针对主模式。

非连续模式下,一次完整的输出传输为:片选信号有效, `SPI_TX_DATA` 写入发送值,触发一个 `SPI_SIZE.BITSIZE` 长度的数据传输,完毕后,片选信号失效。

连续模式下,一次完整的输出传输为:片选信号有效, `SPI_TX_DATA` 写入发送值,一个 `SPI_SIZE.BITSIZE` 长度的数据传输完毕, `SPI_TX_DATA` 写入新值,触发下一个 `SPI_SIZE.BITSIZE` 长度的数据传输,片选信号保持有效,直至应用将本批次数据发送完毕,片选信号失效。

### 16.3.2.7 中断处理

SPI 接口包含三种类型的中断事件,分别是:数据传输完成事件,异常事件和溢出事件。

- 数据完成事件,当前数据传输完成。高电平有效,对 `SPI_IE.CMPLT_IF` 写 1 清除。
- 异常事件, SPI 接口为 Slave 模式,若在传输过程中片选信号受到干扰,被拉高,将产生片选异常事件。高电平有效,对 `SPI_IE.AB_IF` 写 1 清除。
- 溢出事件, `SPI_RX_DATA` 寄存器数据没有及时被读走,将产生溢出事件。高电平有效,对 `SPI_IE.OV_IF` 写 1 清除。

上述事件,默认是不触发 SPI 中断,可以通过配置 `SPI_IE[7:4]`使能事件产生中断。

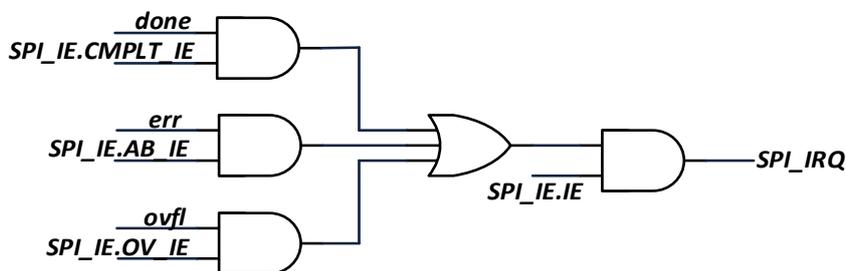


图 16-6 SPI 模块中断选信号产生图

### 16.3.2.8 波特率设置

SPI 接口时钟通过对系统时钟分频获得,分频系数来自 `SPI_BAUD.BAUD`。SPI 传输波特率配置计算



公式为:

$$\text{SPI 传输波特率} = \text{系统时钟} / (2 * (\text{BAUD} + 1))$$

SPI 协议为半拍协议，上升沿发送数据，下降沿采集数据；或者下降沿发送数据，上升沿采集数据。

SPI 接口采用同步设计，需要对外部设备的信号进行同步采样，同步时钟为系统时钟。数据和时钟信号（此时为 Slave 模式）的同步，需要两拍系统时钟。考虑到时钟相位的偏移，此时需要一拍系统时钟的冗余，由此推导出最快的 SPI 传输波特率为系统时钟的 1/8，高电平周期为四拍系统时钟，低电平周期为四拍系统时钟。因此，SPI\_BAUD.BAUD 的配置值不能小于 3。

## 16.4 寄存器

### 16.4.1 地址分配

SPI 模块寄存器的基地址是 0x40010000。

表 16-1 SPI 模块控制寄存器列表

名称	偏移	说明
SPI_CFG	0x00	SPI 配置寄存器
SPI_IE	0x04	SPI 中断寄存器
SPI_BAUD	0x08	SPI 波特率寄存器
SPI_TXDATA	0x0C	SPI 发送数据寄存器
SPI_RXDATA	0x10	SPI 接收数据寄存器
SPI_SIZE	0x14	SPI 传输数据长度寄存器

### 16.4.2 SPI\_CFG SPI 控制寄存器

地址：0x4001\_0000

复位值：0x0

表 16-2 SPI\_CFG 系统控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DUPLEX	CS	MS	CPHA	CPOL	ENDIAN	EN	
								RW	RW	RW	RW	RW	RW	RW	
								0	1	0	0	2	0	0	

位置	位名称	说明
[31:8]		未使用
[7:6]	DUPLEX	半双工模式设置 0X: 关闭半双工模式 10: 开启半双工模式，仅发送 11: 开启半双工模式，仅接收
[5]	CS	SPI 从设备下，片选信号来源。默认值为 1。



		0: Slave 模式下, 片选信号恒为有效值--0 1: Slave 模式下, 片选信号来自 Master 设备
[4]	MS	SPI 主从模式选择。默认值为 0。 0: Slave 模式 1: Master 模式
[3]	CPHA	SPI 相位选择。默认值为 0。 0: Phase 为 0 1: Phase 为 1
[2]	CPOL	SPI 极性选择。默认值为 0。 0: Polarity 为 0 1: Polarity 为 1
[1]	ENDIAN	SPI 模块传输顺序。默认值为 0。 0: MSB, 高位先传输 1: LSB, 低位先传输
[0]	EN	SPI 模块使能信号。默认值为 0。 0: 关闭 SPI 模块 1: 开启 SPI 模块

### 16.4.3 SPI\_IE SPI 中断寄存器

地址: 0x4001\_0004

复位值: 0x0

表 16-3 SPI\_IE 中断寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IE	CMPLT_IE	AB_IE	OV_IE					CMPLT_IF	AB_IF	OV_IF	
				RW	RW	RW	RW					RW	RW	RW	
				0	0	0	0					0	0	0	

位置	位名称	说明
[31:8]		未使用
[7]	IE	SPI 中断使能开关。默认值为 0。 0: 关闭 SPI 中断 1: 使能 SPI 中断
[6]	CMPLT_IE	SPI 传输, 完成事件中断使能信号。 0: 屏蔽此中断源 1: 使能此中断源
[5]	AB_IE	SPI 传输, 异常事件中断使能信号。 0: 屏蔽此中断源 1: 使能此中断源
[4]	OV_IE	SPI 传输, 溢出事件中断使能信号。默认值为 0。 0: 屏蔽此中断源 1: 使能此中断源



[3]		
[2]	CMPLT_IF	SPI 传输，完成事件。高电平有效，写 1 清除。
[1]	AB_IF	SPI 传输，异常事件。 <b>Slave</b> 模式下，传输未完成，发生片选信号无效事件。高电平有效，写 1 清除。
[0]	OV_IF	SPI 传输，溢出事件。前次接收的旧数据没有被取得走，本次接收的新数据已经到达。 高电平有效，写 1 清除。

#### 16.4.4 SPI\_BAUD SPI 波特率寄存器

地址：0x4001\_0008

复位值：0x0

表 16-4 SPI\_BAUD 控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TRANS_MODE	BAUD						
								RW	RW						
								0	0						

位置	位名称	说明
[31:8]		未使用
[7]	TRANS_MODE	SPI 连续发送，仅用于主模式。默认为 0。 0：非连续发送。 1：连续发送。
[6:0]	BAUD	SPI 传输波特率配置，SPI 实际传输速度计算公式为： SPI 传输速度 = 系统时钟 / (2*(BAUD + 1)) 切记，BAUD 的配置值不能小于 3。

#### 16.4.5 SPI\_TXDATA SPI 数据发送寄存器

地址：0x4001\_000C

复位值：0x0

表 16-5 SPI\_TXDATA 数据发送寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_DATA															
RW															
0															

位置	位名称	说明
[31:16]		未使用



[15:0]	TX_DATA	SPI 数据发送寄存器
--------	---------	-------------

#### 16.4.6 SPI\_RXDATA SPI 数据接收寄存器

地址：0x4001\_0010

复位值：0x0

表 16-6 SPI\_RXDATA 数据接收寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_DATA															
RW															
0															

位置	位名称	说明
[31:16]		未使用
[15:0]	RX_DATA	SPI 数据接收寄存器

#### 16.4.7 SPI\_SIZE SPI 数据传输长度寄存器

地址：0x4001\_0014

复位值：0x0

表 16-7 SPI\_SIZE 数据传输长度寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												BITSIZE			
												RW			
												0			

位置	位名称	说明
[31:8]		未使用
[4:0]	BITSIZE	字节长度寄存器。 0x00: 非法值 0x07: 非法值 0x08: 8-Bit 0x09: 9-Bit ... 0x0E: 14-Bit 0x0F: 15-Bit 0x10: 16-Bit



## 17 CMP

### 17.1 概述

比较器信号处理模块(以下简称 CMP 模块, 为便于区分后续图示中模拟比较器使用 Comparator 表示, 数字 CMP 模块使用 CMP 表示), 用于处理两个模拟轨到轨比较器产生的输出信号, 由一系列使能、极性控制、滤波等数字电路组成。信号处理时钟由系统主时钟分频得到, 此模块也用于向 MCU 产生比较器中断。

比较器可用于以下功能:

1. 反电势过零点检测
2. 硬件过流检测
3. 作为 MCPWM 的 fail 信号来源

比较器的主要特性如下:

1. 每个比较器都具备多种的信号来源可供选择:
  - 多路 GPIO 口输入信号
  - 运放输出信号
  - 运放正端输出信号
  - 1.2V BANDGAP 基准源
  - DAC 输出信号
2. 比较速度可编程, 迟滞电压可编程
3. 输出信号可滤波, 滤波深度可选
4. 可产生 CMP 中断

CMP 模块寄存器基地址 为 0x40010C00。

模拟比较器未经滤波的原始输出值, 可以通过读取 CMP\_DATA 值获得。同时模拟比较器未经滤波的原始输出值也可以通过配置 GPIO 的第二功能通过 P0.14 和 P2.3 送出, 具体 GPIO 第二功能配置及引进位置, 请参考器件 datasheet。

关于模拟比较器的更多说明, 包括其输入端信号的选择, 迟滞的配置, 请参考 4.7 章节。

推荐配置流程:

1. 配置 CMP 的模拟开关

芯片上电的默认状态下, 比较器模块是关闭的, 通过配置寄存器 [SYS\\_AFE\\_REG5.CMPxPDN](#) 可以打开比较器(x=0/1, 代表 CMP0/CMP1 两个比较器)。开启比较器之前, 需要先开启 BGP 模块。



## 2. 配置 CMP 的数字时钟开关、时钟滤波和信号输入开关

通过配置寄存器 [CMP\\_TCLK.CLK\\_EN](#) 可以打开 CMP 的数字时钟开关，1 表示打开，0 表示关闭；通过配置寄存器 [CMP\\_TCLK.FIL\\_CLK\\_DIV16\[7:4\]](#) 可以配置滤波时钟，数值设置范围是 0~15，0 表示 1 分频滤波，15 表示 16 分频滤波；通过配置寄存器 [CMP\\_CFG.CMPx\\_IN\\_EN](#) 可以打开信号输入开关，1 表示打开，0 表示关闭。

## 3. 配置 CMP 的比较速度和迟滞电压

比较器的比较延时为 0.15us，还可通过寄存器 [SYS\\_AFE\\_REG1.CMP\\_FT](#) 设置为小于 30ns。迟滞电压可通过配置寄存器 [SYS\\_AFE\\_REG3.CMP\\_HYS](#) 设置为 20mV/0mV。

## 4. 选择 CMP 的正负端信号来源

CMP 的正端信号有 8 种信号来源可选择，可以通过配置寄存器 [SYS\\_AFE\\_REG3.CMPx\\_SELp](#) 进行设置，负端有 4 种信号来源可选择，可以通过配置寄存器 [SYS\\_AFE\\_REG3.CMPx\\_SELn](#) 进行设置。

## 5. 配置 CMP 的中断

通过配置寄存器 [CMP\\_IE.CMPx\\_IE](#) 可以打开 CMP 中断，1 表示打开，0 表示关闭。

## 17.2 寄存器

## 17.2.1 地址分配

CMP 模块寄存器的基地址是 0x40010000。

表 17-1 比较器寄存器列表

名称	偏移地址	说明
CMP_IE	0x00	比较器中断使能寄存器
CMP_IF	0x04	比较器中断标志寄存器
CMP_TCLK	0x08	比较器分频时钟控制寄存器
CMP_CFG	0x0C	比较器控制寄存器
CMP_BLCWIN	0x10	比较器开窗控制寄存器
CMP_DATA	0x14	比较器输出数值寄存器

## 17.2.2 CMP\_IE 中断使能寄存器

地址：0x4001\_0C00

复位值：0x0

表 17-2 CMP\_IE 比较器中断使能寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														CMP1_IE	CMP0_IE



	RW	RW
	0	0

位置	位名称	说明
[31:2]		未使用
[1]	CMP1_IE	比较器 1 中断使能, 高有效
[0]	CMP0_IE	比较器 0 中断使能, 高有效

### 17.2.3 CMP\_IF 中断标志寄存器

地址: 0x4001\_0C04

复位值: 0x0

表 17-3 CMP\_IF 比较器中断标志寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														CMP1_IF	CMP0_IF
														RW1C	RW1C
														0	0

位置	位名称	说明
[31:2]		未使用
[1]	CMP1_IF	比较器 1 中断标志, 高有效, 写 1 清零
[0]	CMP0_IF	比较器 0 中断标志, 高有效, 写 1 清零

### 17.2.4 CMP\_TCLK 分频时钟控制寄存器

地址: 0x4001\_0C08

复位值: 0x0

表 17-4 CMP\_TCLK 比较器分频时钟控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
										FIL_CLK_DIV16		CLK_EN					FIL_CLK_DIV1248
										RW		RW					RW
										0		0					0

位置	位名称	说明
[31:8]		未使用
[7:4]	FIL_CLK_DIV16	比较器滤波时钟分频, 基于 MCLK 进行 1~16 分频, 影响进入比较器中断的时间
[3]	CLK_EN	时钟使能, 高有效



[2]		未使用
[1:0]	FIL_CLK_DIV1248	比较器滤波时钟分频, 2'b00:1 分频, 2'b01:2 分频, 2'b10:4 分频, 2'b11:8 分频

CMP 滤波时钟频率

$\text{Freq}(\text{CMP\_Filter}) = \text{Freq}(\text{MCLK}) / 2^{\text{CMP\_TCLK.FIL\_CLK\_DIV1248}} / (\text{CMP\_TCLK.FIL\_CLK\_DIV16} + 1)$ , 其中 MCLK 为系统的主时钟, 通常为 96MHz 全速时钟。需要注意的是, 产生 CMP 滤波时钟需要使能 CMP\_TCLK.CLK\_EN 位。

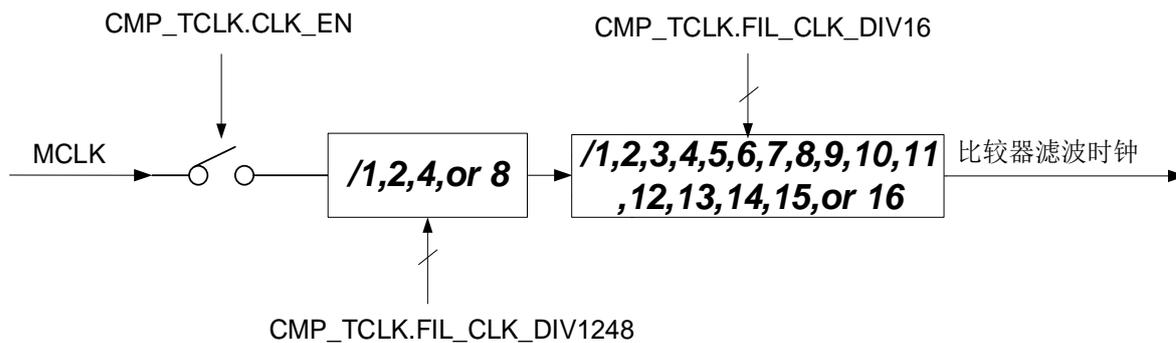


图 17-1 比较器滤波时钟产生

CMP 模块使用此滤波时钟对模拟比较器的输出信号进行 16 时钟周期长的滤波, 即只有信号稳定时间超过 16 个滤波时钟周期才能通过滤波器, CMP 模块输出的滤波后的信号才会发生变化, 如果输入信号稳定时间不足 16 个滤波时钟周期即发生变化, 则 CMP 模块输出的滤波后的信号维持原值不变。即滤波宽度=滤波时钟周期\*16。

### 17.2.5 CMP\_CFG 控制寄存器

地址: 0x4001\_0C0C

复位值: 0x0

表 17-5 CMP\_CFG 比较器控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CMP1_OW_PWM_POL	CMP1_IRQ_TRIG	CMP1_IN_EN	CMP1_POL	CMP0_OW_PWM_POL	CMP0_IRQ_TRIG	CMP0_IN_EN	CMP0_POL
								RW	RW	RW	RW	RW	RW	RW	RW
								0	0	0	0	0	0	0	0

位置	位名称	说明
[31:8]		未使用
[7]	CMP1_OW_PWM_POL	比较器 1 开窗 PWM 信号极性选择, 在 CMP_BLCWIN 使能情况



		下使用
[6]	CMP1_IRQ_TRIG	比较器 1 中断触发类型, 0:电平触发, 1:边沿触发
[5]	CMP1_IN_EN	比较器 1 信号输入使能
[4]	CMP1_POL	比较器 1 极性选择, 0:高电平有效; 1:低电平有效
[3]	CMP0_OW_PWM_POL	比较器 0 开窗 PWM 信号极性选择, 在 CMP_BLCWIN 使能情况下使用
[2]	CMP0_IRQ_TRIG	比较器 0 中断触发类型, 0:电平触发, 1:边沿触发
[1]	CMP0_IN_EN	比较器 0 信号输入使能
[0]	CMP0_POL	比较器 0 极性选择, 0:高电平有效; 1:低电平有效

比较器的极性及使能控制如图 17-2 所示。

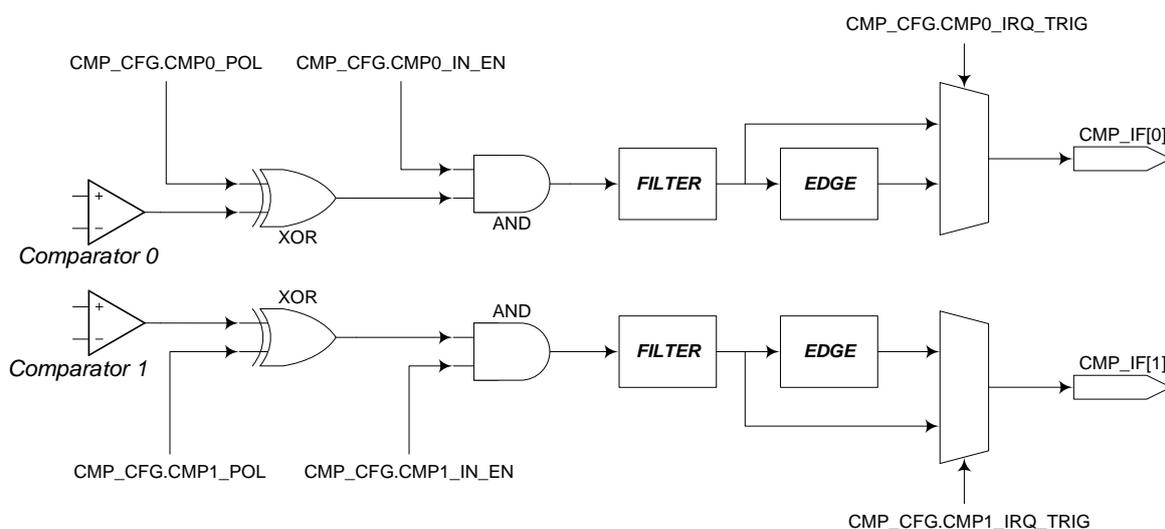


图 17-2 比较器控制及中断产生逻辑

比较器模块与 MCPWM 模块可以联合动作, 其中 MCPWM 模块的 P 管控制信号可以作为比较器开窗的控制信号。但比较器自身的中断信号产生于开窗控制无关, 仅仅受 CMP\_CFG 寄存器影响。

MCPWM 的 fail 信号可以来自 GPIO, 也可以来自比较器模块, 使用 MCPWM\_FAIL 寄存器进行控制。如果 MCPWM 的 fail 信号来自比较器, 则是经过比较器模块内部的开窗控制的。fail 信号进入 MCPWM 后也会进行极性使能以及滤波等处理, 与比较器模块类似但完全独立, 由 MCPWM 内部的寄存器进行控制。MCPWM 内部与 fail 相关的错误中断信号产生收到 MCPWM 内部有关极性使能滤波控制寄存器的影响。具体请参考 MCPWM 章节。

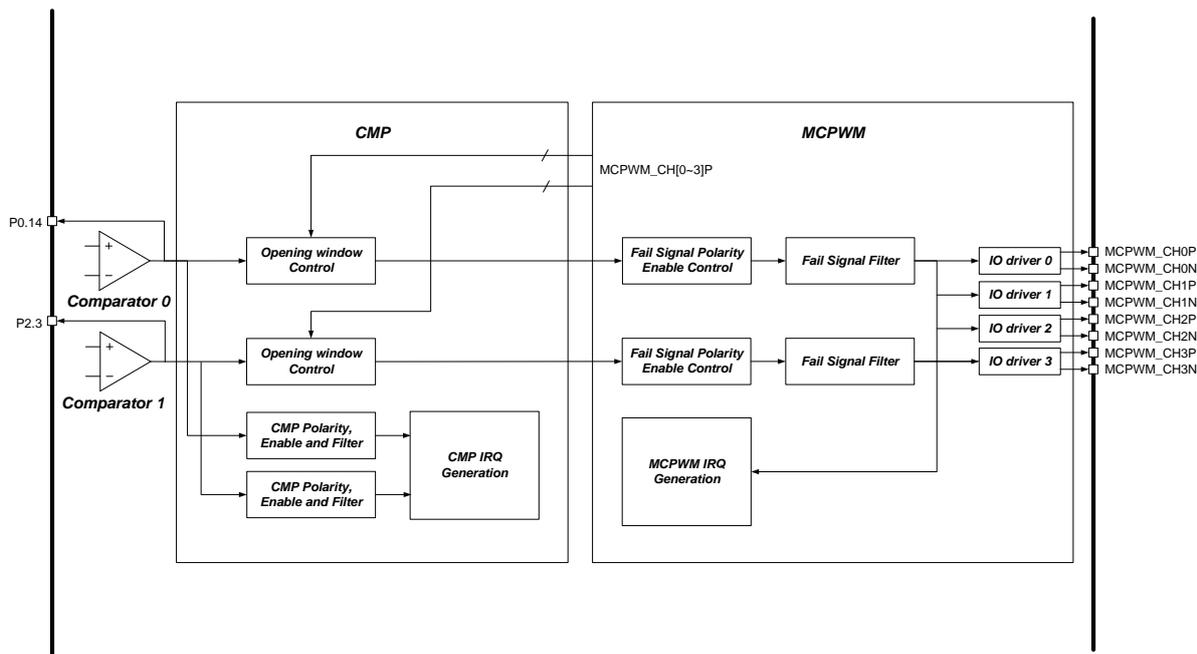


图 17-3 CMP 与 MCPWM 的联动

对于比较器的开窗功能，若  $CMP\_CFG.CMP0\_PWM\_POL=1$ ，则在对应  $MCPWM\ CHN_x\_P$  信号为 1 时，比较器 0 可以产生比较信号输出，其他时刻比较信号为 0；反之，若  $CMP\_CFG.CMP0\_PWM\_POL=0$ ，则在对应  $MCPWM\ CHN_x\_P$  信号为 0 时，比较器 0 可以产生比较信号输出，其他时刻比较信号为 0。比较器 1 的开窗控制信号极性由  $CMP\_CFG.CMP1\_PWM\_POL$  位进行控制，逻辑相同。

注意： $CMP\_CFG.CMP0\_PWM\_POL$  和  $CMP\_CFG.CMP1\_PWM\_POL$  同时会影响送入 MCPWM 模块作为 FAIL 信号的比较器信号，如图 17-4 所示。来自比较器的 MCPWM FAIL 信号为模拟比较器输出的原始信号，未经过比较器数字接口模块的滤波处理，但是可以被 MCPWM 的通道信号进行开窗控制，开窗控制设置见比较器数字接口模块。FAIL 信号进入 MCPWM 模块后，可以通过设置  $MCPWM\_TCLK$  进行滤波。

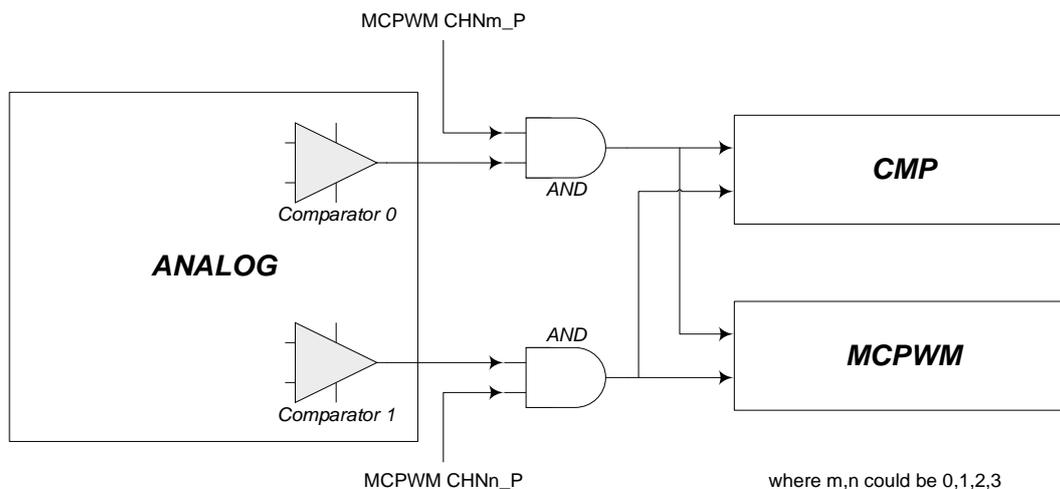


图 17-4 比较器开窗功能图示

## 17.2.6 CMP\_BLCWIN 开窗控制寄存器

地址：0x4001\_0C10

复位值：0x0

表 17-6 CMP\_BLCWIN 比较器开窗控制寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CMP1_CHN3P_WIN_EN	CMP1_CHN2P_WIN_EN	CMP1_CHN1P_WIN_EN	CMP1_CHN0P_WIN_EN	CMP0_CHN3P_WIN_EN	CMP0_CHN2P_WIN_EN	CMP0_CHN1P_WIN_EN	CMP0_CHN0P_WIN_EN
								RW							
								0	0	0	0	0	0	0	0

位置	位名称	说明
[31:8]		保留
[7]	CMP1_CHN3P_WIN_EN	使用 MCPWM 模块 CHN3_P 通道输出的 P 管开关控制信号作为比较器 1 开窗使能
[6]	CMP1_CHN2P_WIN_EN	使用 MCPWM 模块 CHN2_P 通道输出的 P 管开关控制信号作为比较器 1 开窗使能
[5]	CMP1_CHN1P_WIN_EN	使用 MCPWM 模块 CHN1_P 通道输出的 P 管开关控制信号作为比较器 1 开窗使能
[4]	CMP1_CHN0P_WIN_EN	使用 MCPWM 模块 CHN0_P 通道输出的 P 管开关控制信号作为比较器 1 开窗使能
[3]	CMP0_CHN3P_WIN_EN	使用 MCPWM 模块 CHN3_P 通道输出的 P 管开关控制信号作为比较器 0 开窗使能
[2]	CMP0_CHN2P_WIN_EN	使用 MCPWM 模块 CHN2_P 通道输出的 P 管开关控制信号作为比较器 0 开窗使能
[1]	CMP0_CHN1P_WIN_EN	使用 MCPWM 模块 CHN1_P 通道输出的 P 管开关控制信号作为比较器 0 开窗使能
[0]	CMP0_CHN0P_WIN_EN	使用 MCPWM 模块 CHN0_P 通道输出的 P 管开关控制信号作为比较器 0 开窗使能

通常 CMP\_BLCWIN[3:0]或 CMP\_BLCWIN[7:4]中有 1bit 为 1，表明使用对应的 CHNx\_P 对比较器 0/1 的信号产生进行控制。如果 CMP\_BLCWIN[3:0]或 CMP\_BLCWIN[7:4]为 4'b0000，则表示比较器 0/1 比较信号的产生与 PWM 信号无关。

## 17.2.7 CMP\_DATA 输出数据寄存器

地址：0x4001\_0C14

复位值：0x0



表 17-7 CMP\_DATA 比较器输出数据寄存器

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												CMP1_FLT_DATA	CMP0_FLT_DATA	CMP1_RAW_DATA	CMP0_RAW_DATA
												R	R	R	R
												0	0	0	0

位置	位名称	说明
[31:4]		保留
[3]	CMP1_FLT_DATA	比较器 1 经过滤波后的信号
[2]	CMP0_FLT_DATA	比较器 0 经过滤波后的信号
[1]	CMP1_RAW_DATA	比较器 1 原始输出信号，直接来自模拟比较器 1
[0]	CMP0_RAW_DATA	比较器 0 原始输出信号，直接来自模拟比较器 0

## 18 版本历史

表 18-1 文档版本历史

时间	版本号	说明
2025.01.02	1.24	增加 FLASH NVR 关于 OPA Offset 的地址
2024.12.02	1.23	增加 ADC_IF 触发所需时间说明
2024.06.17	1.22	添加 NVR 校准参数地址信息
2023.09.28	1.21	修改 ADC 模块寄存器命名, 和器件库中头文件中命名保持一致
2023.08.10	1.19	修改了 GPIO 部分的笔误
2023.04.07	1.18	增加 ADC 转换占用时间描述
2023.03.23	1.17	修改 LSI 时钟精度范围
2023.02.18	1.16	修订 MCPWM_SDCFG.UPDATE_INTV 描述
2023.02.10	1.15	补充关于 PLLPDN、BGPPD、BGPPD 的描述
2022.12.30	1.14	修订开方运算周期数的说明
2022.11.23	1.13	增加软复位 ADC 模块描述
2022.11.17	1.12	修订 OPA0 和 OPA2 分时复用, OPA1 和 OPA3 分时复用
2022.11.10	1.11	增加 IO 与内部模拟电路间连接电阻描述
2022.09.15	1.10	增加 ADC/OPA/CMP 配置流程描述
2022.08.02	1.09	简化 SYS_AFE_REG 描述
2021.10.18	1.08	修改 FLASH 擦写次数
2021.08.04	1.07	修订 OPA 反馈电阻比例
2021.05.20	1.06	增加 SYS_OPA_SEL 描述
2020.11.15	1.05	修改 Timer 滤波时钟的描述
2020.09.17	1.04	LRC 时钟由 32kHz 改为 64kHz, 影响看门狗复位时间
2020.06.11	1.03	修订对 ADC_DC 的描述
2020.06.06	1.02	修订 MCPWM_DTHxx 寄存器描述
2020.05.01	1.01	增加复位源的描述, 修订 Timer ETON 的描述
2020.01.18	1.0	修订格式, 修订描述有问题的章节
2019.05.08	0.1	初始版本, 包含测试相关说明的内部版本

## 免责声明

LKS 和 LKO 为凌鸥创芯注册商标。

南京凌鸥创芯电子有限公司（以下简称：“Linko”）尽力确保本文档内容的准确和可靠，但是保留随时更改、更正、增强、修改产品和/或 文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。

客户应针对应用需求选择合适的 Linko 产品，详细设计、验证和测试您的应用，以确保满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。

Linko 在此确认未以明示或暗示方式授予 Linko 或第三方的任何知识产权许可。

Linko 产品的转售，若其条款与此处规定不同，Linko 对此类产品的任何保修承诺无效。

Linko 产品禁止用于军事用途或生命监护、维持系统。

如有更早期版本文档，一切信息以此文档为准。

