



南京凌鸥创芯电子有限公司

LKS32MC06x User Manual

© 2019, 版权归凌鸥创芯所有
机密文件，未经许可不得扩散

目录

1	文档约定	1
1.1	寄存器读写权限	1
1.2	缩略词汇表	1
2	存储器和总线构架	2
2.1	系统架构	2
2.2	地址空间分配	2
2.3	中断号分配	2
2.4	嵌入式闪存	3
3	模拟电路	4
3.1	简述	4
3.2	电源管理系统	5
3.3	时钟系统	6
3.4	基准电压源	7
3.5	ADC 模块	7
3.6	运算放大器	8
3.7	比较器	9
3.8	温度传感器	10
3.9	DAC 模块	11
4	时钟和复位	12
4.1	时钟	12
4.1.1	时钟源	12
4.1.2	时钟域	13
4.1.2.1	MCLK	13
4.1.2.1.1	MCLK 整体门控	14
4.1.2.1.2	外设时钟门控	14
4.1.2.1.3	外设时钟分频	15
4.1.2.2	ACLK	15



4.1.2.3	JCLK	15
4.1.2.4	LCLK	15
4.2	复位	16
4.2.1	复位源	16
4.2.2	复位结构	16
4.2.3	复位记录	16
4.3	寄存器	16
4.3.1	地址分配	16
4.3.2	看门狗密码寄存器 <i>SYS_WDT_PSW</i>	17
4.3.3	看门狗清零寄存器 <i>SYS_WDT_CLR</i>	18
4.3.4	比较器输出寄存器 <i>SYS_AFE_CMP</i>	18
4.3.5	模拟寄存器概述	18
4.3.6	模拟配置寄存器 0 <i>SYS_AFE_REG0</i>	18
4.3.7	模拟配置寄存器 1 <i>SYS_AFE_REG1</i>	20
4.3.8	模拟配置寄存器 2 <i>SYS_AFE_REG2</i>	20
4.3.9	模拟配置寄存器 3 <i>SYS_AFE_REG3</i>	21
4.3.10	模拟配置寄存器 4 <i>SYS_AFE_REG4</i>	23
4.3.11	模拟配置寄存器 5 <i>SYS_AFE_REG5</i>	23
4.3.12	模拟配置寄存器 6 <i>SYS_AFE_REG6</i>	24
4.3.13	模拟配置寄存器 7 <i>SYS_AFE_REG7</i>	25
4.3.14	模拟配置寄存器 8 <i>SYS_AFE_REG8</i>	25
4.3.15	模拟配置寄存器 9 <i>SYS_AFE_REG9</i>	26
4.3.16	模拟配置寄存器 10 <i>SYS_AFE_REGA</i>	27
4.3.17	DAC 数字量寄存器 <i>SYS_AFE_DAC</i>	27
4.3.18	时钟控制寄存器 <i>SYS_CLK_CFG</i>	28
4.3.19	复位控制寄存器 <i>SYS_RST_CFG</i>	28
4.3.20	复位源记录寄存器 <i>SYS_RST_SRC</i>	28
4.3.21	复位源记录清除寄存器 <i>SYS_CLR_RST</i>	29
4.3.22	外设时钟分频寄存器 0 <i>SYS_CLK_DIV0</i>	29



4.3.23	外设时钟分频寄存器 2 SYS_CLK_DIV2.....	29
4.3.24	外设时钟门控寄存器 SYS_CLK_FEN.....	29
4.3.25	休眠寄存器 SYS_CLK_SLP.....	30
4.3.26	在线编程模式寄存器 SYS_IAP.....	30
4.3.27	校正模式寄存器 SYS_TRIM.....	30
4.3.28	软复位寄存器 SYS_SFT_RST.....	30
4.3.29	写保护寄存器 SYS_PROTECT.....	31
5	FLASH.....	32
5.1	概述.....	32
5.2	功能特点.....	32
5.2.1	访问操作.....	32
5.2.1.1	FLASH 读取操作.....	32
5.2.1.2	FLASH 写入操作.....	33
5.2.1.3	FLASH 擦除操作.....	35
5.2.1.4	FLASH 预取操作.....	37
5.2.1.5	FLASH 加密保护.....	37
5.2.1.6	FLASH 在线升级(IAP).....	39
5.3	寄存器.....	40
5.3.1	地址分配.....	40
5.3.2	擦除/写入时间参数配置寄存器 FLASH_TH.....	41
5.3.3	地址寄存器 FLASH_ADDR.....	41
5.3.4	写数据寄存器 FLASH_WDATA.....	42
5.3.5	写数据寄存器 FLASH_RDATA.....	42
5.3.6	控制寄存器 FLASH_CFG.....	42
5.3.7	写入控制寄存器 FLASH_PROG.....	42
5.3.8	写入保护寄存器 FLASH_PASS.....	43
5.3.9	擦除控制寄存器 FLASH_ERASE.....	43
5.3.10	擦除类型寄存器 FLASH_ERASE_OP.....	43
5.3.11	加密状态寄存器 FLASH_PROTECT.....	43



5.3.12	加密状态更新寄存器 FLASH_PROTECT_LD.....	43
5.3.13	工作状态寄存器 FLASH_READY.....	44
6	通用 IO (GPIO)	45
6.1	概述.....	45
6.1.1	功能框图.....	45
6.1.2	产品特点.....	45
6.2	寄存器.....	46
6.2.1	地址分配.....	46
6.2.2	GPIOx_PIE.....	47
6.2.3	GPIOx_POE.....	47
6.2.4	GPIOx_PDI.....	48
6.2.5	GPIOx_PDO.....	48
6.2.6	GPIOx_PUE.....	48
6.2.7	GPIOx_PDE.....	49
6.2.8	GPIOx_PODE.....	49
6.2.9	GPIOx_LCKR.....	50
6.2.10	GPIOx_F3210.....	50
6.2.11	GPIOx_F7654.....	51
6.2.12	GPIOx_FBA98.....	51
6.2.13	GPIOx_FFEDC.....	51
6.2.14	外部中断、唤醒、锁定保护.....	51
6.2.14.1	EXTI_CR0.....	51
6.2.14.2	EXTI_CR1.....	52
6.2.14.3	EXTI_IF.....	52
6.2.14.4	LCKR_PRT.....	53
6.2.14.5	WAKE_POL.....	53
6.2.14.6	WAKE_EN.....	54
6.3	应用指南.....	54
6.3.1	配置锁定.....	54



6.3.2	外部中断.....	56
7	模数转换器(ADC).....	57
7.1	概述.....	57
7.1.1	功能框图.....	57
7.1.2	ADC 触发方式.....	58
7.1.3	ADC 输出数制.....	59
7.1.4	ADC 量程.....	59
7.1.5	ADC 校正.....	60
7.2	寄存器.....	60
7.2.1	地址分配.....	60
7.2.2	采样数据寄存器.....	62
7.2.2.1	ADCx_DAT0.....	62
7.2.2.2	ADCx_DAT1.....	62
7.2.2.3	ADCx_DAT2.....	62
7.2.2.4	ADCx_DAT3.....	62
7.2.2.5	ADCx_DAT4.....	62
7.2.2.6	ADCx_DAT5.....	63
7.2.2.7	ADCx_DAT6.....	63
7.2.2.8	ADCx_DAT7.....	63
7.2.2.9	ADCx_DAT8.....	63
7.2.2.10	ADCx_DAT9.....	63
7.2.2.11	ADCx_DAT10.....	64
7.2.2.12	ADCx_DAT11.....	64
7.2.3	信号来源寄存器.....	64
7.2.3.1	ADCx_CHN0.....	64
7.2.3.2	ADCx_CHN1.....	64
7.2.3.3	ADCx_CHN2.....	65
7.2.4	分段采样次数寄存器.....	65
7.2.4.1	ADCx_CHNT.....	65



7.2.5	中断使能寄存器.....	66
7.2.5.1	ADCx_IE.....	66
7.2.6	配置寄存器.....	66
7.2.6.1	ADCx_CFG.....	66
7.2.7	增益选择寄存器.....	67
7.2.7.1	ADCx_GAIN.....	67
7.2.8	中断标志寄存器.....	68
7.2.8.1	ADCx_IF.....	68
7.2.9	软件触发寄存器.....	68
7.2.9.1	ADCx_SWT.....	68
7.2.10	直流偏置寄存器.....	68
7.2.10.1	ADCx_DC0.....	69
7.2.10.2	ADCx_DC1.....	69
7.2.11	增益校正寄存器.....	69
7.2.11.1	ADCx_AMC0.....	69
7.2.11.2	ADCx_AMC1.....	69
7.3	应用指南.....	70
7.3.1	ADC 采样触发模式.....	70
7.3.1.1	单段触发模式.....	71
7.3.1.2	两段触发模式.....	72
7.3.1.3	四段触发模式.....	72
7.3.2	中断.....	73
7.3.2.1	单段触发采样完成中断.....	73
7.3.2.2	两段触发采样完成中断.....	73
7.3.2.3	四段触发采样完成中断.....	73
7.3.3	配置修改.....	73
8	通用定时器.....	75
8.1	概述.....	75
8.1.1	功能框图.....	75



8.1.1.1	总线接口模块.....	75
8.1.1.2	寄存器模块.....	75
8.1.1.3	IO 滤波模块.....	75
8.1.1.4	通用定时器模块.....	76
8.1.1.5	编码器模块.....	76
8.1.1.6	时钟分频模块.....	76
8.1.2	功能特点.....	76
8.2	实现说明.....	76
8.2.1	时钟分频.....	76
8.2.2	中断标志清零.....	76
8.2.3	滤波.....	76
8.2.4	模式.....	77
8.2.4.1	计数器.....	77
8.2.4.2	比较模式.....	77
8.2.4.3	捕获模式.....	78
8.2.5	编码器.....	78
8.2.5.1	正交编码信号.....	79
8.2.5.2	符号加脉冲信号.....	80
8.2.5.3	CCW/CW 双脉冲信号.....	81
8.3	寄存器.....	82
8.3.1	地址分配.....	82
8.3.2	Time 寄存器.....	83
8.3.2.1	Timer x 配置寄存器 UTIMER_UNTx_CFG.....	83
8.3.2.2	Timer x 门限寄存器 UTIMER_UNTx_TH.....	84
8.3.2.3	Timer x 计数寄存器 UTIMER_UNTx_CNT.....	84
8.3.2.4	Timer x 通道 0 比较捕获寄存器 UTIMER_UNTx_CMP0.....	85
8.3.2.5	Timer x 通道 1 比较捕获寄存器 UTIMER_UNTx_CMP1.....	85
8.3.3	Encoder x 寄存器.....	85
8.3.3.1	Encoder x 配置寄存器 UTIMER_ECDx_CFG.....	85



8.3.3.2	Encoder x 计数门限寄存器 UTIMER_ECDx_TH.....	86
8.3.3.3	Encoder x 计数值寄存器 UTIMER_ECDx_CNT.....	86
8.3.4	滤波控制寄存器.....	86
8.3.4.1	UTIMER_FLT_TH01.....	86
8.3.4.2	UTIMER_FLT_TH23.....	87
8.3.5	系统控制寄存器.....	87
8.3.5.1	UTIMER_CFG.....	87
8.3.6	中断管理寄存器.....	87
8.3.6.1	中断使能寄存器 UTIMER_IE.....	88
8.3.6.2	中断标志寄存器 UTIMER_IF.....	88
9	HALL 信号处理模块.....	90
9.1	综述.....	90
9.2	寄存器.....	90
9.2.1	地址分配.....	90
9.2.2	HALL 模块配置寄存器 HALL_CFG.....	90
9.2.3	HALL 模块信息寄存器 HALL_INFO.....	91
9.2.4	HALL 宽度计数值寄存器 HALL_WIDTH.....	91
9.2.5	HALL 模块计数器门限值寄存器 HALL_TH.....	91
9.2.6	HALL 计数寄存器 HALL_CNT.....	91
9.3	实现说明.....	91
9.3.1	信号来源.....	91
9.3.2	工作时钟.....	92
9.3.3	信号滤波.....	92
9.3.4	捕获.....	93
9.3.5	中断.....	93
9.3.6	数据流程.....	93
10	MCPWM.....	94
10.1	概述.....	94
10.1.1	Base Counter 模块.....	95



10.1.2	Fail Check 模块.....	96
10.1.3	MCPWM 特殊输出状态.....	97
10.1.4	IO DRIVER 模块.....	97
10.1.4.1	MCPWM 波形输出-中心对齐模式.....	98
10.1.4.2	MCPWM 波形输出-边沿对齐模式.....	99
10.1.4.3	MCPWM IO 死区控制.....	100
10.1.4.4	MCPWM IO 极性设置.....	101
10.1.4.5	MCPWM IO 自动保护.....	101
10.1.5	ADC Trigger Timer 模块.....	101
10.2	寄存器.....	102
10.2.1	地址分配.....	102
10.2.2	MCPWM_TH00.....	103
10.2.3	MCPWM_TH01.....	103
10.2.4	MCPWM_TH10.....	103
10.2.5	MCPWM_TH11.....	103
10.2.6	MCPWM_TH20.....	104
10.2.7	MCPWM_TH21.....	104
10.2.8	MCPWM_TH30.....	104
10.2.9	MCPWM_TH31.....	105
10.2.10	MCPWM_TMR0.....	105
10.2.11	MCPWM_TMR1.....	105
10.2.12	MCPWM_TMR2.....	105
10.2.13	MCPWM_TMR3.....	106
10.2.14	MCPWM_IE.....	106
10.2.15	MCPWM_IF.....	107
10.2.16	MCPWM_EIE.....	108
10.2.17	MCPWM{EIF.....	109
10.2.18	MCPWM_IO01.....	110
10.2.19	MCPWM_IO23.....	111



10.2.20	MCPWM_SDCFG	112
10.2.21	MCPWM_UPDATE.....	112
10.2.22	MCPWM_TCLK.....	113
10.2.23	MCPWM_FAIL.....	113
10.2.24	MCPWM_TH	114
10.2.25	MCPWM_PRT	115
10.2.26	MCPWM_CNT	115
10.2.27	MCPWM_DTH00.....	115
10.2.28	MCPWM_DTH01.....	115
10.2.29	MCPWM_DTH10.....	116
10.2.30	MCPWM_DTH11.....	116
10.2.31	MCPWM_DTH20.....	116
10.2.32	MCPWM_DTH21.....	116
10.2.33	MCPWM_DTH30.....	117
10.2.34	MCPWM_DTH01.....	117
11	UART	118
11.1	概述.....	118
11.2	功能说明.....	118
11.2.1	发送.....	118
11.2.2	接收.....	118
11.2.3	波特率配置.....	118
11.3	寄存器.....	119
11.3.1	地址分配.....	119
11.3.2	UARTx 控制寄存器 UARTx_CTRL.....	119
11.3.3	UARTx 波特率设置高字节寄存器 UARTx_DIVH	119
11.3.4	UARTx 波特率设置低字节寄存器 UARTx_DIVL	120
11.3.5	UARTx 收发缓冲寄存器 UARTx_BUFF	120
11.3.6	UARTx 地址匹配寄存器 UARTx_ADR.....	120
11.3.7	UARTx 状态寄存器 UARTx_STT.....	120



11.3.8	UARTx 中断使能寄存器 UARTx_IE	120
11.3.9	UARTx 中断标志寄存器 UARTx_IF.....	121
11.4	应用指南.....	121
12	信号协处理器模块.....	122
12.1	概述.....	122
12.1.1	功能框图.....	122
12.1.2	特点.....	122
12.2	寄存器.....	123
12.2.1	地址分配.....	123
12.2.2	除法器.....	123
12.2.2.1	被除数寄存器 DSP_DID	123
12.2.2.2	除数寄存器 DSP_DIS.....	123
12.2.2.3	商寄存器 DSP_QUO	123
12.2.2.4	余数寄存器 DSP_REM	124
12.2.3	开方器.....	124
12.2.3.1	被开方数寄存器 DSP_RAD	124
12.2.3.2	平方根寄存器 DSP_SQRT	124
12.2.3.3	控制状态寄存器 DSP_SC	124
12.3	实现说明.....	125
12.3.1	时钟门控时序.....	125
13	I2C.....	126
13.1	概述.....	126
13.2	寄存器说明.....	127
13.2.1	地址分配.....	127
13.2.2	I2C_ADDR.....	128
13.2.3	I2C0_CFG	128
13.2.4	I2C0_SCR.....	129
13.2.5	I2C0_DATA	129
13.2.6	I2C0_MSCR.....	130



13.2.7	<i>I2C0_BUF_CTRL</i>	130
13.2.8	<i>I2C0_BUF_ADDR</i>	131
13.3	应用指南.....	131
14	SPI	134
14.1	概述.....	134
14.2	寄存器说明.....	134
14.2.1	<i>地址分配</i>	134
14.2.2	<i>SPI0_SHIFTER</i>	135
14.2.3	<i>SPI0_DATA</i>	135
14.2.4	<i>SPI0_CR0</i>	135
14.2.5	<i>SPI0_CR1</i>	137
14.2.6	<i>SPI0_CR2</i>	138
14.2.7	<i>SPI0_BUF_ADDR</i>	138
14.3	应用说明.....	138
15	版本历史	141



表格目录

表 2-1 系统地址空间分配.....	2
表 2-2 中断号分布.....	2
表 4-1 系统时钟源.....	12
表 4-2 MCLK 时钟分频.....	13
表 4-3 系统复位源.....	16
表 4-4 系统控制寄存器.....	17
表 4-5 看门狗密码寄存器 SYS_WDT_PSW.....	17
表 4-6 看门狗清零寄存器 SYS_WDT_CLR.....	18
表 4-7 模拟配置寄存器 0 SYS_AFE_REG0.....	18
表 4-8 模拟配置寄存器 1 SYS_AFE_REG1.....	20
表 4-9 模拟配置寄存器 2 SYS_AFE_REG2.....	20
表 4-10 模拟配置寄存器 3 SYS_AFE_REG3.....	21
表 4-11 模拟配置寄存器 4 SYS_AFE_REG4.....	23
表 4-12 模拟配置寄存器 5 SYS_AFE_REG5.....	23
表 4-13 模拟配置寄存器 6 SYS_AFE_REG6.....	24
表 4-14 模拟配置寄存器 7 SYS_AFE_REG7.....	25
表 4-15 模拟配置寄存器 8 SYS_AFE_REG8.....	25
表 4-16 模拟配置寄存器 9 SYS_AFE_REG9.....	26
表 4-17 模拟配置寄存器 10 SYS_AFE_REGA.....	27
表 4-18 DAC 数字量寄存器 SYS_AFE_DAC.....	27
表 4-19 时钟控制寄存器 SYS_CLK_CFG.....	28
表 4-20 复位控制寄存器 SYS_RST_CFG.....	28
表 4-21 复位源记录寄存器 SYS_RST_SRC.....	28
表 4-22 复位源记录清除寄存器 SYS_CLR_RST.....	29
表 4-23 外设时钟分频寄存器 0SYS_CLK_DIV0.....	29
表 4-24 外设时钟分频寄存器 2SYS_CLK_DIV2.....	29
表 4-25 外设时钟门控寄存器 SYS_CLK_FEN.....	29
表 4-26 休眠寄存器 SYS_CLK_SLP.....	30



表 4-27 在线编程模式寄存器 SYS_IAP	30
表 4-28 校正模式寄存器 SYS_TRIM	30
表 4-29 软复位寄存器 SYS_SFT_RST	30
表 4-30 保护寄存器 SYS_PROTECT	31
表 5-1 FLASH 地址分配表	35
表 5-2 FLASH 擦除类型表	36
表 5-3 FLASH 控制寄存器	40
表 5-4 擦除/写入时间参数配置寄存器 FLASH_TH	41
表 5-5 地址寄存器 FLASH_ADDR	41
表 5-6 写数据寄存器 FLASH_WDATA	42
表 5-7 写数据寄存器 FLASH_RDATA	42
表 5-8 控制寄存器 FLASH_CFG	42
表 5-9 写入控制寄存器 FLASH_PROG	42
表 5-10 写入保护寄存器 FLASH_PASS	43
表 5-11 擦除控制寄存器 FLASH_ERASE	43
表 5-12 擦除类型寄存器 FLASH_ERASE_OP	43
表 5-13 加密状态寄存器 FLASH_PROTECT	43
表 5-14 加密状态更新寄存器 FLASH_PROTECT_LD	43
表 5-15 工作状态寄存器 FLASH_READY	44
表 6-1 GPIOx 寄存器列表	46
表 6-2 GPIO 中断/唤醒/配置锁定模块寄存器列表	46
表 6-3 GPIOx 输入使能寄存器 GPIOx_PIE	47
表 6-4 GPIOx 输出使能寄存器 GPIOx_POE	47
表 6-5 GPIOx 输入数据寄存器 GPIOx_PDI	48
表 6-6 GPIOx 输出数据寄存器 GPIOx_PDO	48
表 6-7 GPIOx 上拉使能寄存器 GPIOx_PUE	48
表 6-8 GPIOx 下拉使能寄存器 GPIOx_PDE	49
表 6-9 GPIOx 开漏使能寄存器 GPIOx_PODE	49
表 6-10 GPIOx 配置锁定寄存器 GPIOx_LCKR	50



表 6-11GPIOx 功能选择寄存器 GPIOx_F3210	50
表 6-12GPIOx 功能选择寄存器 GPIOx_F7654	51
表 6-13GPIOx 功能选择寄存器 GPIOx_FBA98.....	51
表 6-14GPIOx 功能选择寄存器 GPIOx_FFEDC.....	51
表 6-15 外部中断配置寄存器 EXTI_CR0	51
表 6-16 外部中断配置寄存器 EXTI_CR1	52
表 6-17 外部中断标志寄存器 EXTI_IF	52
表 6-18 锁定保护寄存器 LCKR_PRT	53
表 6-19 外部唤醒源极性配置寄存器 WAKE_POL	53
表 6-20 外部唤醒源使能寄存器 WAKE_EN	54
表 7-1 ADC 输出数字量数制转换	59
表 7-2 ADC0 寄存器列表.....	60
表 7-3 ADC1 寄存器列表.....	61
表 7-4 采样数据寄存器 ADCx_DAT0	62
表 7-5 采样数据寄存器 ADCx_DAT1	62
表 7-6 采样数据寄存器 ADCx_DAT2	62
表 7-7 采样数据寄存器 ADCx_DAT3	62
表 7-8 采样数据寄存器 ADCx_DAT4	62
表 7-9 采样数据寄存器 ADCx_DAT5	63
表 7-10 采样数据寄存器 ADCx_DAT6	63
表 7-11 采样数据寄存器 ADCx_DAT7	63
表 7-12 采样数据寄存器 ADCx_DAT8	63
表 7-13 采样数据寄存器 ADCx_DAT9	63
表 7-14 采样数据寄存器 ADCx_DAT10.....	64
表 7-15 采样数据寄存器 ADCx_DAT11.....	64
表 7-16 信号来源寄存器 ADCx_CHN0.....	64
表 7-17 信号来源寄存器 ADCx_CHN1.....	64
表 7-18 信号来源寄存器 ADCx_CHN2.....	65
表 7-19 ADC 采样信号通道选择	65



表 7-20 分段采样次数寄存器 ADCx_CHNT.....	65
表 7-21 中断使能寄存器 ADCx_IE.....	66
表 7-22 配置寄存器 ADCx_CFG.....	66
表 7-23 增益选择寄存器 ADCx_GAIN.....	67
表 7-24 中断标志寄存器 ADCx_IF.....	68
表 7-25 软件触发寄存器 ADCx_SWT.....	68
表 7-26 直流偏置寄存器 ADCx_DC0.....	69
表 7-27 直流偏置寄存器 ADCx_DC1.....	69
表 7-28 增益校正寄存器 ADCx_AMC0.....	69
表 7-29 增益校正寄存器 ADCx_AMC1.....	69
表 7-30 ADC 采样触发模式.....	71
表 8-1 编码器正交编码工作模式.....	79
表 8-2 编码器符号加脉冲工作模式.....	80
表 8-3 编码器 CCW/CW 双脉冲工作模式.....	81
表 8-4 通用定时器配置寄存器地址分配.....	82
表 8-5Timer x 配置寄存器 UTIMER_UNTx_CFG.....	83
表 8-6Timer x 门限寄存器 UTIMER_UNTx_TH.....	84
表 8-7Timer x 计数寄存器 UTIMER_UNTx_CNT.....	84
表 8-8Timer x 通道 0 比较捕获寄存器 UTIMER_UNTx_CMP0.....	85
表 8-9Timer x 通道 1 比较捕获寄存器 UTIMER_UNTx_CMP1.....	85
表 8-10 Encoder x 配置寄存器 UTIMER_ECDx_CFG.....	85
表 8-11 Encoder x 计数门限寄存器 UTIMER_ECDx_TH.....	86
表 8-12 Encoder x 计数值寄存器 UTIMER_ECDx_CNT.....	86
表 8-13 滤波控制寄存器 UTIMER_FLT_TH01.....	86
表 8-14 滤波控制寄存器 UTIMER_FLT_TH23.....	87
表 8-15 UTIMER 配置寄存器 UTIMER_CFG.....	87
表 8-16 中断使能寄存器 UTIMER_IE.....	88
表 8-17 中断标志寄存器 UTIMER_IF.....	88
表 9-1HALL 模块寄存器地址分配.....	90



表 9-2 HALL 模块配置寄存器 HALL_CFG	90
表 9-3 HALL 模块信息寄存器 HALL_INFO	91
表 9-4 HALL 宽度计数值寄存器 HALL_WIDTH	91
表 9-5 HALL 模块计数器门限值寄存器 HALL_TH	91
表 9-6 HALL 计数寄存器 HALL_CNT	91
表 10-1 MCPWM 计数器阈值与事件对应表	101
表 10-2 MCPWM 模块寄存器列表	102
表 10-3 MCPWM_TH00 配置寄存器	103
表 10-4 MCPWM_TH01 配置寄存器	103
表 10-5 MCPWM_TH10 配置寄存器	103
表 10-6 MCPWM_TH11 配置寄存器	104
表 10-7 MCPWM_TH20 配置寄存器	104
表 10-8 MCPWM_TH21 配置寄存器	104
表 10-9 MCPWM_TH30 配置寄存器	104
表 10-10 MCPWM_TH31 配置寄存器	105
表 10-11 MCPWM_TMR0 配置寄存器	105
表 10-12 MCPWM_TMR1 配置寄存器	105
表 10-13 MCPWM_TMR2 配置寄存器	105
表 10-14 MCPWM_TMR3 配置寄存器	106
表 10-15 MCPWM_IE 配置寄存器	106
表 10-16 MCPWM_IF 配置寄存器	107
表 10-17 MCPWM_EIE 配置寄存器	108
表 10-18 MCPWM{EIF 配置寄存器	109
表 10-19 MCPWM_IO01 配置寄存器	110
表 10-20 MCPWM_IO23 配置寄存器	111
表 10-21 MCPWM_SDCFG 配置寄存器	112
表 10-22 MCPWM_UPDATE 配置寄存器	112
表 10-23 MCPWM_TCLK 配置寄存器	113
表 10-24 MCPWM_FAIL 配置寄存器	114



表 10-25 MCPWM_TH 配置寄存器.....	114
表 10-26 MCPWM_PRT 配置寄存器.....	115
表 10-27 MCPWM_CNT 配置寄存器.....	115
表 10-28 MCPWM_DTH00 配置寄存器.....	115
表 10-29 MCPWM_DTH01 配置寄存器.....	115
表 10-30 MCPWM_DTH10 配置寄存器.....	116
表 10-31 MCPWM_DTH11 配置寄存器.....	116
表 10-32 MCPWM_DTH20 配置寄存器.....	116
表 10-33 MCPWM_DTH21 配置寄存器.....	116
表 10-34 MCPWM_DTH30 配置寄存器.....	117
表 10-35 MCPWM_DTH31 配置寄存器.....	117
表 11-1 UARTx 地址分配列表.....	119
表 11-2 UARTx 控制寄存器 UARTx_CTRL.....	119
表 11-3 UARTx 波特率设置高字节寄存器 UARTx_DIVH.....	119
表 11-4 UARTx 波特率设置低字节寄存器 UARTx_DIVL.....	120
表 11-5 UARTx 收发缓冲寄存器 UARTx_BUFF.....	120
表 11-6 UARTx 地址匹配寄存器 UARTx_ADR.....	120
表 11-7 UARTx 状态寄存器 UARTx_STT.....	120
表 11-8 UARTx 中断使能寄存器 UARTx_IE.....	120
表 11-9 UARTx 中断标志寄存器 UARTx_IF.....	121
表 12-1 DSP 寄存器列表.....	123
表 12-2 被除数寄存器 DSP_DID.....	123
表 12-3 除数寄存器 DSP_DIS.....	123
表 12-4 商寄存器 DSP_QUO.....	123
表 12-5 余数寄存器 DSP_REM.....	124
表 12-6 被开放数寄存器 DSP_RAD.....	124
表 12-7 平方根寄存器 DSP_SQRT.....	124
表 12-8 DSP 控制状态寄存器 DSP_SC.....	124
表 13-1 I2C 模块控制寄存器列表.....	127



表 13-2 I2C0_ADDR 地址寄存器.....	128
表 13-3 I2C0_CFG 配置寄存器.....	128
表 13-4 I2C0_SCR 状态和控制寄存器.....	129
表 13-5 I2C 数据寄存器.....	129
表 13-6 I2C 主机状态和控制寄存器.....	130
表 13-7 I2C0_BUF_CTRL 控制寄存器.....	130
表 13-8 I2C0 buffer 地址寄存器.....	131
表 14-1 SPI 模块控制寄存器列表.....	135
表 14-2 SPI0_SHIFTER 移位寄存器.....	135
表 14-3 SPI0_DATA 数据寄存器.....	135
表 14-4 SPI0_CR0 控制寄存器 0.....	135
表 14-5 SPI0_CR1 控制寄存器 1.....	137
表 14-6 SPI0_CR2 控制寄存器 2.....	138
表 14-7 SPI0_BUF_ADDR 地址寄存器.....	138
表 15-1 文档版本历史.....	141



图片目录

图 3-1 模拟电路功能框图	5
图 4-1 时钟架构	13
图 4-2 MCLK 架构	14
图 4-3 外设时钟门控分频	15
图 4-4 复位架构	16
图 5-1 Flash 模块结构框图	32
图 5-2 Flash 模块读操作流程	33
图 5-3 Flash 模块写操作流程	34
图 5-4 Flash 模块连续写操作流程	35
图 5-5 Flash 模块擦除操作流程	37
图 5-6 Flash 模块加密操作流程	38
图 5-7 Flash 模块解密操作流程	39
图 5-8 在线升级空间映射关系	40
图 5-9 在线升级流程转换图	40
图 6-1 GPIO 功能框图	45
图 7-1 ADC 采集模块功能框图	58
图 7-2 ADC 单段采样状态转移图	72
图 7-3 ADC 两段采样状态转移图	72
图 7-4 ADC 四段采样状态转移图	73
图 8-1 模块顶层功能框图	75
图 8-2 滤波示意图	77
图 8-3 通用计数器	77
图 8-4 比较模式	78
图 8-5 捕获模式	78
图 8-6 编码器只在 T1 时刻计数的正交编码信号计数情况	79
图 8-7 编码器在 T1 或 T2 时刻计数的正交编码信号计数情况	80
图 8-8 编码器在 T1 上升下降沿都计数的符号加脉冲信号计数情况	80
图 8-9 编码器在仅 T1 上升沿计数的符号加脉冲信号计数情况	81



图 8-10 编码器仅在 T1/T2 上升沿计数的 CCW/CW 双脉冲信号计数情况	81
图 8-11 编码器在 T1/T2 上升下降沿计数的 CCW/CW 双脉冲信号计数情况	82
图 9-1 7/5 滤波模块框图	92
图 9-2 数据流程框图	93
图 10-1 MCPWM 模块框图	95
图 10-2 Base Counter t0/t1 时序	95
图 10-3 Base Counter 数据流程图	96
图 12-1 DSP 模块功能框图	122
图 12-2 时钟门控	125
图 13-1 I2C 模块结构框图	127
图 13-2 I2C 模块基本的一次传输时序	131
图 14-1 SPI 模块结构框图	134
图 16-6 SPI 通讯信号极性相位(Polarity=0, Phase=0)	136
图 16-7 SPI 通讯信号极性相位(Polarity=0, Phase=1)	137
图 16-8 SPI 通讯信号极性相位(Polarity=1, Phase=0)	137
图 16-9 SPI 通讯信号极性相位(Polarity=1, Phase=1)	137



1 文档约定

1.1 寄存器读写权限

R/W	读/写，软件可以读写这些位。
R(RO)	只读，软件只能读取这些位。
W(WO)	只写，软件只能写入该位。读取该位时将返回默认值。

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。

WDT: Watch dog, 看门狗

NVR: Non-volatile register

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 存储器 and 总线构架

2.1 系统架构

LKS06x 系列使用 Cortex-M0 内核，32bit AHB-lite 总线。

2.2 地址空间分配

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，请参考相应器件的数据手册中的存储器映像图。

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_0FFF	4KB	RAM
		0xF000_0000	0xF000_1FFF	8KB	N/A
SYS	同总线	0x4000_0000	0x4000_00FF	256B	SYSTEM control, Clock / Reset Management
FLSCR	同总线	0x4000_0100	0x4000_013F	64B	FLASH control registers
		0x4000_0200	0x4000_021F	32B	N/A
SPI	FCLK[0]/sft_rst[0]	0x4000_3080	0x4000_30BF	64B	SPI interface
I2C	FCLK[0]/sft_rst[0]	0x4000_30C0	0x4000_30FF	64B	I2C interface
HALL	FCLK[1]/sft_rst[1]	0x4000_3200	0x4000_32FF	256B	HALL interface
ADC0	ACLK	0x4000_3300	0x4000_33FF	256B	ADC1 interface
ADC1	ACLK	0x4000_3400	0x4000_34FF	256B	ADC2 interface
TIMER	FCLK[2]/sft_rst[2]	0x4000_3500	0x4000_35FF	256B	General Purpose Timer
MCPWM	FCLK[3]/sft_rst[3]	0x4000_3600	0x4000_36FF	256B	Motor Control Pulse Width Modulation
GPIO	同总线	0x4000_3700	0x4000_37FF	256B	General Purpose Input / Output
DSP	同总线	0x4000_3800	0x4000_38FF	256B	DSP with 32-cycle DIV and 8-cycle SQRT
UART0	FCLK[4]/sft_rst[4]	0x4000_3900	0x4000_39FF	256B	UART0
UART1	FCLK[5]/sft_rst[5]	0x4000_3A00	0x4000_3AFF	256B	UART1

2.3 中断号分配

表 2-2 中断号分布

中断号	说明	中断号	说明
0	TIMER0	16	WAKEUP, 系统唤醒中断



1	TIMER1	17	电源电压过低
2	TIMER2	18	Reserved
3	TIMER3	19	Reserved
4	ENCODER0	20	Reserved
5	ENCODER1	21	Reserved
6	I2C	22	Reserved
7	GPIO	23	Reserved
8	UART0	24	Reserved
9	HALL	25	Reserved
10	SPI	26	Reserved
11	ADC0	27	Reserved
12	ADC1	28	Reserved
13	MCPWM	29	Reserved
14	UART1	30	Reserved
15	CMP	31	Reserved

2.4 嵌入式闪存

闪存的大小: 32k

闪存存储器有两个不同存储区域:

主闪存存储块 (main), 它包括应用程序和用户数据区

信息块 (info 区/NVR), 其包含两个部分:

- 选项字节(Option bytes) – 内含硬件及存储保护用户配置选项。
- 系统内存(System memory) – 其包含 boot loader 代码



3 模拟电路

3.1 简述

模拟电路包含以下模块：

- 集成 2 路 12BIT SAR ADC，最高采样率 3MHz。每路 12 通道
- 集成 4 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 $\pm 2^{\circ}\text{C}$ 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（图中红色线表示电源线，其他线代表信号线）。



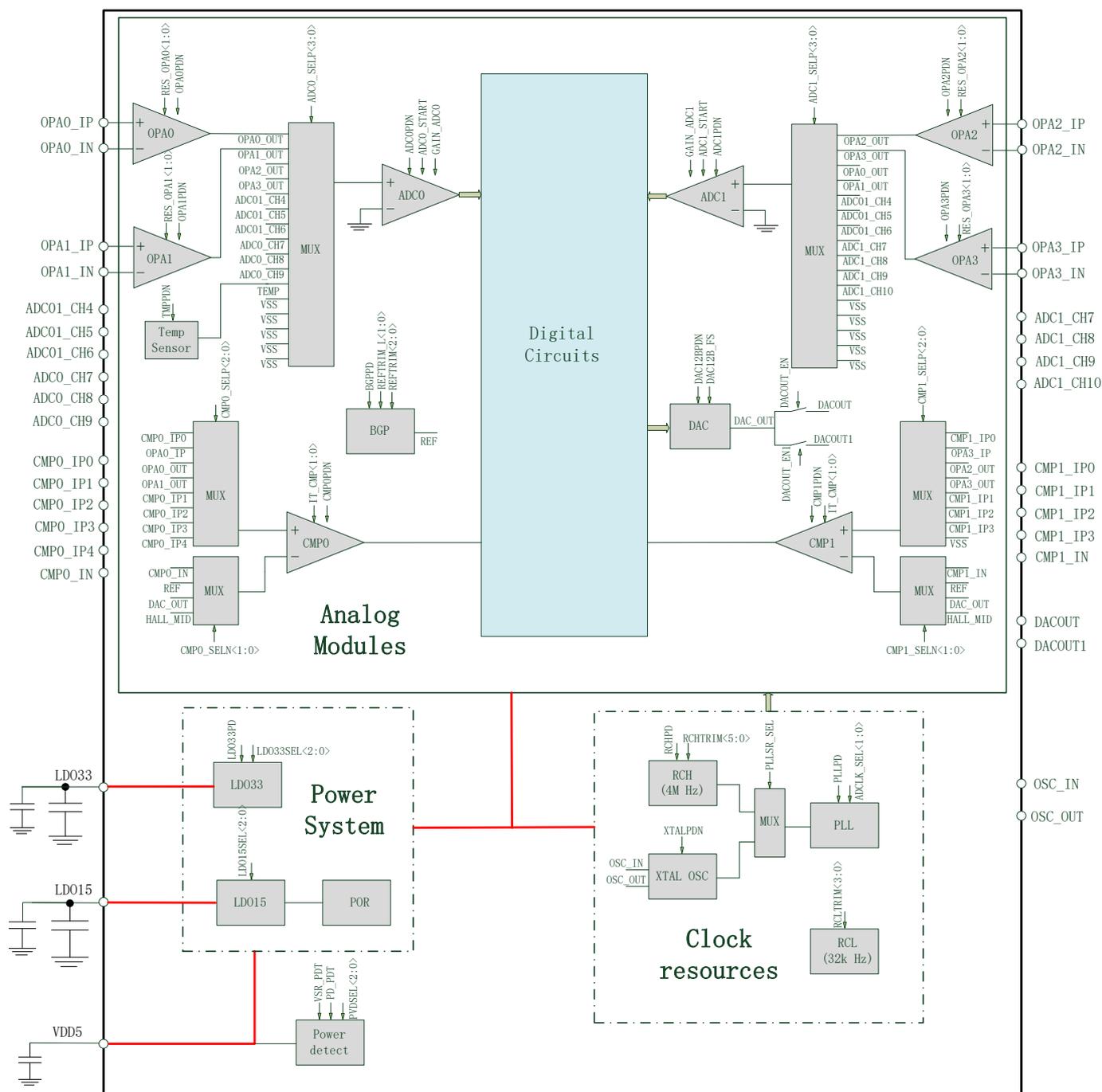


图 3-1 模拟电路功能框图

3.2 电源管理系统

电源管理系统由 LDO33 模块 (3.3V LDO)、LD015 模块 (1.5V LDO)、电源检测模块 (PVD)、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 1.5V LDO、3.3V LDO,



其中 3.3V LDO 给内部 ADC、DAC、基准电压源、运算放大器、比较器、温度传感器、RC 时钟、晶体时钟等模拟电路供电。1.5V LDO 为内部所有数字电路、PLL 模块供电。

两个 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。LDO33/LDO15 输出引脚上均需接 10 μ F 和 0.1 μ F 并联的去耦电容到地。去耦电容应尽可能靠近芯片引脚，且去耦电容接地点和芯片模拟地之间在 PCB 上应该有充分的敷地相连。LDO33 的负载驱动电流为 40mA，LDO15 为 30mA。

LDO33 的输出电压可通过设置寄存器 LDO33TRIM[2:0]来调节，LDO15 的输出电压可通过设置寄存器 LDO15TRIM[2:0]来调节。LDO33 和 LDO15 在芯片出厂前已经过校正，一般情况下，用户不需要额外配置这些寄存器。如需微调 LDO 的输出电压，需要读取原配置值，在此基础上加上微调量，再将对应的配置值填入模拟控制寄存器。

LDO33TRIM[2:0]和 LDO15TRIM[2:0]的说明见[模拟寄存器 SYS_AFE_REG9](#)

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 LDO33 的电压，在 LDO33 电压低于 2.5V 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。

PVD 模块对 5V 输入电源进行检测，如低于某一设定阈值，则产生报警(中断)信号以提醒 MCU。中断提醒阈值可通过寄存器 PVDSEL[1:0]设置为不同的电压。PVD 模块可通过设置 PD_PDT=1 关闭。

PVDSEL[1:0]/ PD_PDT 的说明见[模拟寄存器 SYS_AFE_REG6](#)

3.3 时钟系统

时钟系统包括内部 32KHz RC 时钟、内部 4MHz RC 时钟、外部 4~8MHz 晶体起振电路、PLL 电路组成。

32K RC 时钟作为 MCU 系统慢时钟使用，作为低功耗状态下的 MCU 时钟使用。4MHz RC 时钟作为 MCU 主时钟使用，配合 PLL 可提供最高到 96MHz 的时钟。外部 4~8MHz 晶体起振电路作为备用时钟使用。

32k 和 4M RC 时钟均带有出厂校正，可在常温下实现 32K RC 时钟 $\pm 5\%$ 的精度，4M RC 时钟 $\pm 1\%$ 的精度。其中 4M RC 时钟还开放有用户校正寄存器，可进一步将精度校正到 $\pm 0.5\%$ 范围。32K RC 时钟在-40~105 $^{\circ}$ C范围内的精度为 $\pm 20\%$ ，4M RC 时钟在该温度范围的精度为 $\pm 1\%$ 。

32K RC 时钟频率可通过寄存器 RCLTRIM[3:0]进行设置，4M RC 时钟频率可通过寄存器 RCHTRIM[5:0]进行设置。

芯片出厂前时钟已经过校正，一般情况下，用户不需要额外配置这些寄存器。如需微调频率，需要读取原配置值，在此基础上加上微调量，再将对应的配置值填入相应的寄存器。

RCLTRIM_L[3:0]的说明见[模拟寄存器 SYS_AFE_REG8](#)



RCHTRIM<5:0>的说明见[模拟寄存器 SYS_AFE_REG9](#)

4M RC 时钟通过设置 RCHPD = '0' 打开（默认打开，设 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（保证 BGPPD='0'）。芯片上电的默认状态下，4M RC 时钟和 BGP 模块都是开启的。32K RC 时钟始终开启，不能关闭。

PLL 对 4M RC 时钟进行倍频，以提供给 MCU、ADC 等模块更高速的时钟。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] 可设置为不同的 ADC 工作频率。

PLL 通过设置 PLLPD = '0' 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，RC 时钟、PLL 和 BGP 模块都是开启的。

晶体起振电路内置了放大器和起振电容，仅需在 IO OSC_IN/OSC_OUT 之间接入一个晶体，并设置 XTALPDN=1 即可起振。

ADCLKSEL<1:0>的说明见[模拟寄存器 SYS_AFE_REG7](#)

BGPPD/RCHPD/XTALPDN/PLLPD 的说明见[模拟寄存器 SYS_AFE_REG5](#)

3.4 基准电压源

基准源电路(BGP REF)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 2us 达到稳定。BGP 输出电压约 1.2V，精度为±0.8%

基准电压源的电压大小可通过寄存器 REFTRIM_L<1:0>、REF_LTRIM、REFTRIM <2:0> 进行设置，芯片出厂前基准源已经过校正，一般情况下，用户不需要额外配置这些寄存器。如需微调电压，需要读取原配置值，在此基础上加上微调量，再将对应的配置值填入相应的寄存器。

基准源可通过设置 REF_BGP_EN=1，将基准电压送至 IO 'CMP1_IP2' 进行测量。正常工作模式下

BGPPD 的说明见[模拟寄存器 SYS_AFE_REG5](#)

REF_BGP_EN 的说明见[模拟寄存器 SYS_AFE_REG3](#)

REFTRIM_L<1:0>的说明见[模拟寄存器 SYS_AFE_REG9](#)

REFTRIM<2:0>/ REF_LTRIM 的说明见[模拟寄存器 SYS_AFE_REGA](#)

3.5 ADC 模块

参见模数转换器(ADC)章节。



3.6 运算放大器

4路输入输出 rail-to-rail 运算放大器, 内置反馈电阻, 外部引脚上还需接一个电阻 R_0 到信号源。反馈电阻 $R_2:R_1$ 的阻值可通过寄存器 $RES_OPA_x[1:0]$ 设置, 以实现不同的放大倍数。

$RES_OPA_x<1:0>$ 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:

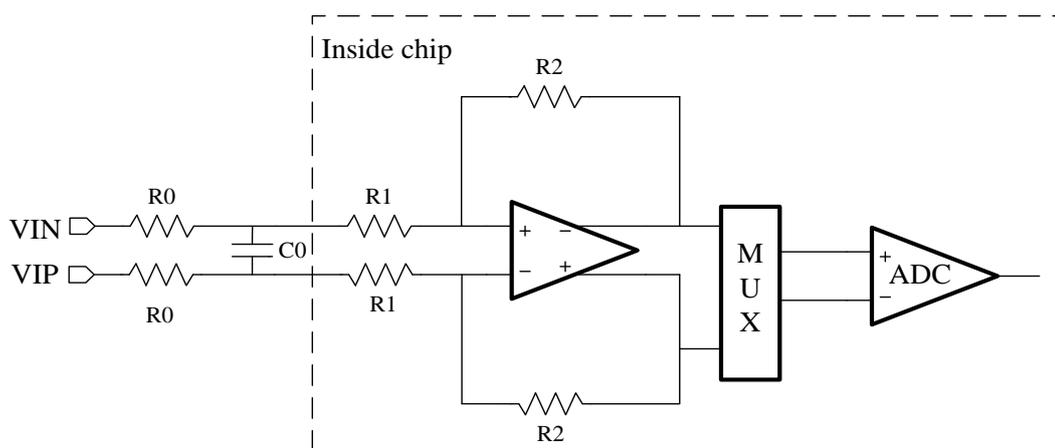


图 3-2 放大器框图

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, 一般建议接 $>20k$ 欧的外部电阻。

对于分流电阻采样的应用, 建议接 $100\sim 2K$ 欧的外部电阻。C0 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 $R_0 \cdot C_0$ 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 C0 可以不加。

放大器可通过设置 $OPAOUT_EN=1$ 将 4 路放大器负向的信号送至相应 IO 口 ($OPA0_OUT\sim OPA3_OUT$) 进行测量 (对应关系见 datasheet 芯片管脚说明), 但是在正常工作模式下不可将信号送出, 以免受到干扰。

$OPAOUT_EN$ 的说明见[模拟寄存器 SYS_AFE_REG2](#)

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 $OPA_xPDN(x=0,1,2,3) = 1$ 打开, 开启放大器之前, 需要先开启 BGP 模块。

OPA_xPDN 的说明见[模拟寄存器 SYS_AFE_REG5](#)

运放输入正负端内置限压二极管, 电机相线通过一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。



3.7 比较器

内置 2 路输入 rail-to-rail 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。

比较器的比较延时可通过寄存器 `IT_CMP` 设置为 0.15uS/0.6uS。迟滞电压通过 `CMP_HYS` 设置为 20mV/0mV。

比较器正负两个输入端的信号来源都可通过寄存器 `CMPx_SELP[2:0]`和 `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 `HALLx_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，具体连接方式见下图 3-3。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 `HALLx_MID` 之后才会导通，否则开关都处于断开状态。当
`CMPx_IP1/` `CMPx_IP2/` `CMPx_IP3` 管脚连的是 HALL 信号时，通过将 HALL 信号与 `HALLx_MID` 信号
进行比较，可快速得到 HALL 信号的状态。

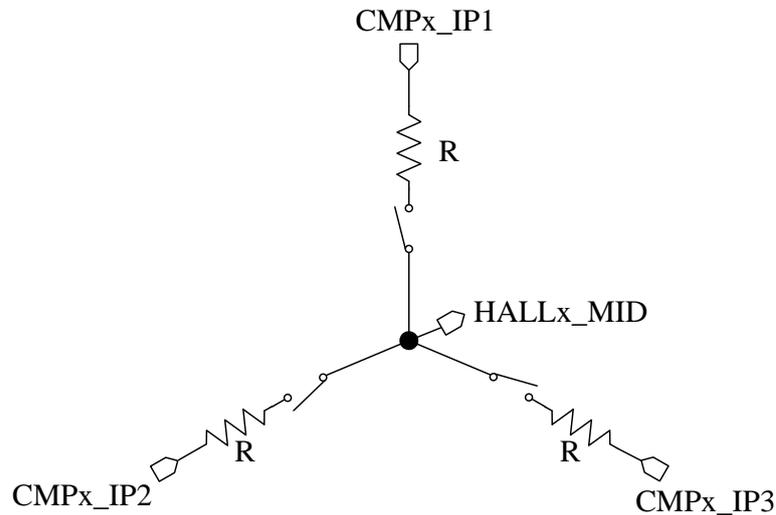


图 3-3 HALLx_MID 信号

比较器输出结果，可以通过 `SYS_AFE_CMP` 寄存器读出。

`IT_CMP<1:0>`的说明见[模拟寄存器 SYS_AFE_REG4](#)

`CMPx_SELN<1:0>/` `CMPx_SELP<2:0>/` `CMP_HYS` 的说明见[模拟寄存器 SYS_AFE_REG3](#)

比较器的输出 `CMPx_RESULT` 的说明见[比较器输出寄存器 SYS_AFE_CMP](#)

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(x=0,1) = 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](#)



3.8 温度传感器

芯片内置温度传感器,大批量时,在-40~85°C范围内精度为 3°C。-40~105°C范围内精度为 4°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 2us,因此需在 ADC 测量传感器之前 2us 打开。另外,开启温度传感器前需要设置工作状态寄存器,将 $TMPCKOFF$ 和 $TEMP_MODE$ 寄存器都设置为'1'。

温度传感器信号仅连至 ADC0 进行测量。需要将温度传感器配置到 ADC0 中的一个输入信号通道,转换完成后到该设置通道的相应地址上读取转换后的 ADC 值。

ADC 部分的设置参考第七章[模数转换器\(ADC\)](#)

$TEMP_MODE$ 的说明见[模拟寄存器 SYS_AFE_REG2](#)

$TMPPDN$ 的说明见[模拟寄存器 SYS_AFE_REG5](#)

$TMPCKOFF$ 的说明见[模拟寄存器 SYS_AFE_REG6](#)

温度传感器的典型曲线如下图所示:

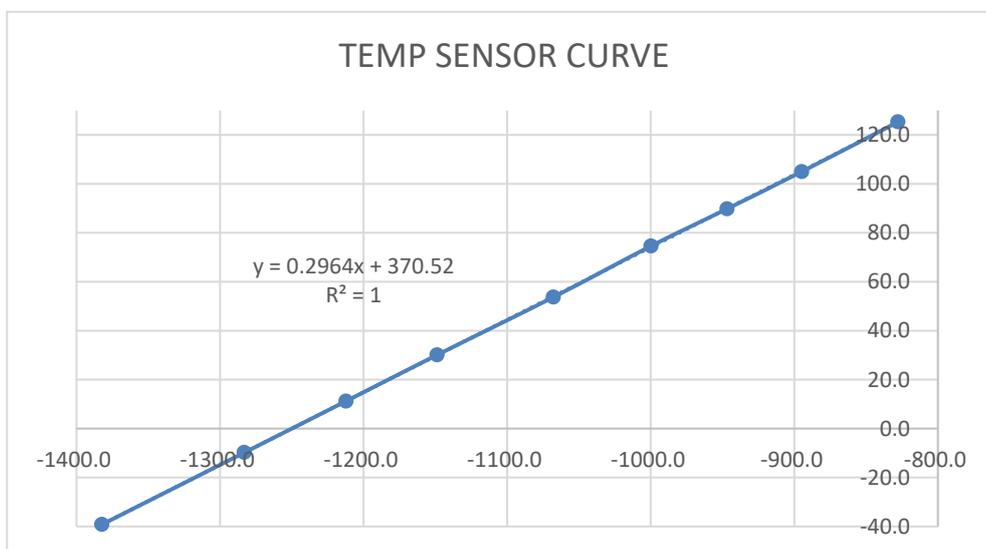


图 3-4 温度传感器曲线

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:

$$y = 0.2964x + 370.52$$

求得的 Y 值即为此时的温度。

公式中有两个系数, $a=0.2964$, $b=370.52$ 。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 0x0000028C。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 0x00000288。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 区相应地址读出 **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。

3.9 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 DAC12B_FS 设置为 1V 或 3V。

12bit DAC 可通过配置寄存器 DACOUT_EN/DACOUT_EN1=1, 分别将 DAC 输出送至 IO 口 P0.0/P0.3, 可驱动 >5kΩ 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 $\frac{DAC_{fs}}{4096} * Din$

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/ DACOUT_EN1/ DAC12B_FS 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](#)



4 时钟和复位

4.1 时钟

4.1.1 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。HSE 可能失效，仅部分应用会使用外部晶振时钟 HSE。

表 4-1 系统时钟源

时钟源	频率	来源	误差	说明
LSI	32KHz	内部 RC 振荡器	23KHz~42KHz	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 HIS/HSE 作为输入，输出是 HSI/HSE 时钟的 24 倍频，作为系统主时钟。
HSE	4MHz	外部晶体振荡器	0	外部晶体，在对时钟精度有严格要求(例如 ppm 级别的精度要求)的应用下，可使用 HSE 作为 PLL 输入时钟来产生 96M 的系统主时钟
SWD	1MHz	调试器		SWD 的 JTAG 时钟

如下图，MUX0 用于在 HSI / HSE 中进行选择，切换电路由模拟电路完成。

PLL 经过分频 (1/8~8/8) 后路作为系统主时钟 MCLK。系统复位时，MCLK 门控，系统自动打开 HSI/PLL，等待 PLL 稳定后，切换到 PLL /8/4，即 3MHz 上工作，保证系统安全。

PLL 时钟经过分频送到 ADC (最高工作频率 48MHz)，即 ACLK。

内部低频 RC 产生一路 LSI 时钟 LCLK，主要用于 WDT 工作时钟。

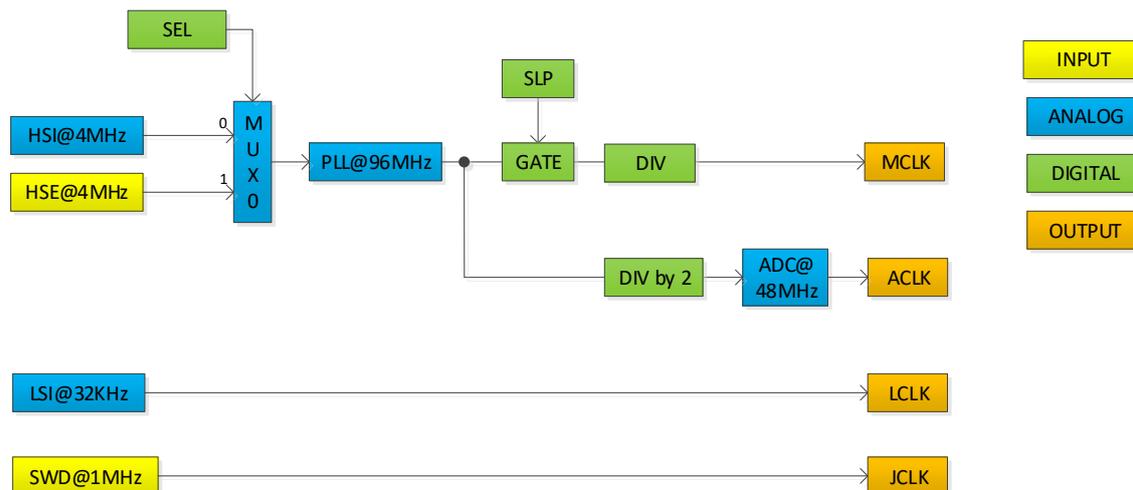


图 4-1 时钟架构

4.1.2 时钟域

系统主要包括 4 个时钟域，MCLK/ACLK/LCLK/JCLK。

4.1.2.1 MCLK

MCLK 是系统主时钟。支持以 1/8 为最小粒度的分频配置，可以覆盖 12MHz~96MHz 的频率范围。

表 4-2 MCLK 时钟分频

分频系数	频率/MHz	是否均匀
1/8	12	是
2/8	24	是
3/8	36	否
4/8	48	是
5/8	60	否
6/8	72	否
7/8	84	否
8/8	96	是

PMU 模块控制处理器核的时钟，当系统进入低功耗模式时，会关闭 PLL、HRC 等高速时钟，只保留 LRC 时钟供最小系统工作使用。

外设模块通常会包含至少两个时钟域：总线时钟 bclk/主时钟 mclk:

bclk 来自 ahb_clk，该时钟在总线访问本模块时打开，平时关闭以降低功耗。

mclk 来自时钟 MCLK，模块功能时钟可以是 MCLK 被门控或者进一步分频以获得所需功能，并降低功耗。



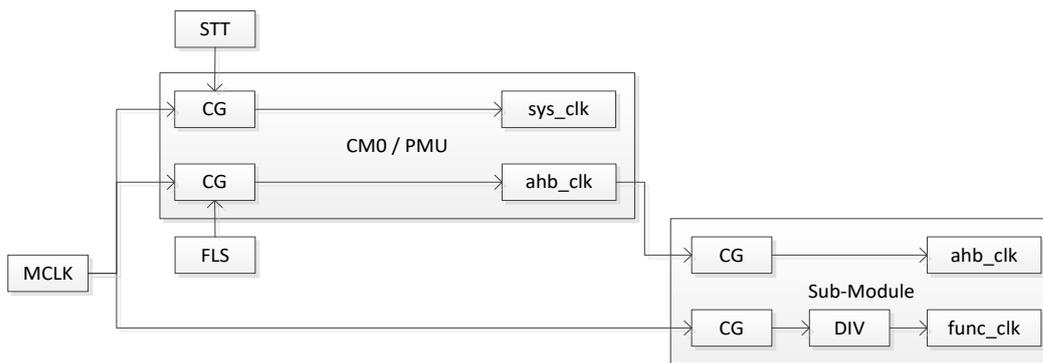


图 4-2 MCLK 架构

4.1.2.1.1 MCLK 整体门控

MCLK 时钟可以通过配置进行关闭，从而令包括 CPU 和所有外设在内的大部分数字电路处于休眠状态。MCLK 时钟门控关闭时，功耗管理单元(Power Management Unit, PMU)状态机依次关闭 PLL, HSI/HSE, BGP 等模拟模块，以降低功耗。

系统的休眠模式仅仅关闭 PLL, HSI, HSE 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 CPU 停止取指。

在进入休眠之前，需要设置 SYS_AFE_REG5 寄存器的 PLLPD、RCHPD 和 BGPPD 关闭 PLL 时钟，HSI 时钟和 BandGap；并根据应用实际情况，设置 SYS_AFE_REG5 关闭 OPA, CMP, ADC。如果芯片当前工作于 PLL 时钟上，且 HSI 时钟作为了 PLL 的参考时钟，则设置 SYS_AFE_REG5 关闭 PLL 时钟、HSI 时钟和 BandGap 的动作不会立即生效，需要等到休眠指令发出才会由 PMU 控制依次关闭各个时钟。芯片可以在工作于 PLL 时钟的情况下直接写入休眠指令进入休眠状态。

在应用程序编写中请事先配置好唤醒条件。

由于休眠仅仅关闭了数字电路时钟，而没有关闭电路电源，所以寄存器值会保持为休眠前的配置。

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 配置 0.125s,0.25s,0.5s,1s,2s,4s,8s,16s 共 8 档唤醒时间间隔，具体请参考 4.3.19 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 6.2.14.5 WAKE_POL 和 6.2.14.6 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。



在唤醒后，需要根据应用实际情况，设置 `SYS_AFE_REG5` 开启 OPA, CMP, ADC。如果芯片是在工作于 PLL 时钟的状态下直接写入休眠指令进入休眠状态，则唤醒后芯片仍工作于 PLL 时钟下。当芯片工作于 PLL 时钟且 HSI 作为 PLL 参考时钟时，无论 `SYS_AFE_REG5` 寄存器的 PLLPD、RCHPD 和 BGPPD 是怎样设置，PLL 时钟、HSI 时钟和 BandGap 都会开启。

4.1.2.1.2 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。共设计 8 路可控时钟。

其中，SPI/I2C 有共享 `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]`。

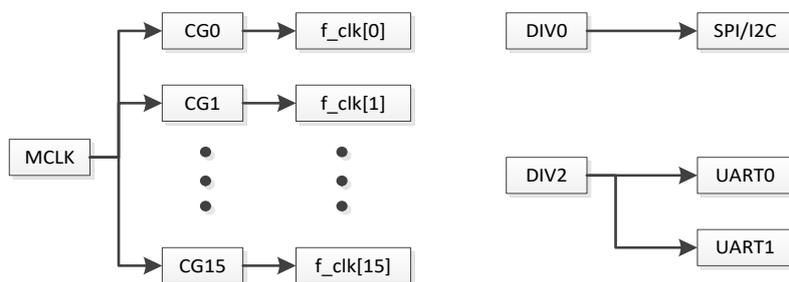


图 4-3 外设时钟门控分频

4.1.2.1.3 外设时钟分频

所有外设均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。

SPI/I2C/UART 的工作时钟分频在 SYS 时钟管理模块中完成。

如图 4-3 外设时钟门控分频所示，模块对应的 `f_clk` 可以进一步分频出模块的工作时钟。

其中 SPI/I2C 共享 `CLK_DIV[0]`，UART0/1 共享 `CLK_DIV[2]`。UART 的波特率在 UART 模块内部还有一个额外的分频器。

4.1.2.2 ACLK

ACLK 是模拟电路对 PLL 时钟的一个分频时钟，供 ADC 使用。

4.1.2.3 JCLK

JCLK 独立于其余的时钟，供 SWD 接口模块 DAP 使用，实现 JTAG 时序。

4.1.2.4 LCLK

LCLK 独立于其余的时钟，供 WDT 模块使用，该模块需要处理 `ahb_clk` 和 LCLK 的异步通信问题。



4.2 复位

4.2.1 复位源

如表 4-3 系统复位源所示，系统包括 3 个复位源。

表 4-3 系统复位源

名称	来源	说明
LPOR	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
HPOR	内部 3.3V 电源管理	监控 3.3V 电源，低于 2.5V 时产生复位
RESET	外部按键	外部 RC 组成按键复位电路
WDT	内部软件狗	0.5S 产生复位

4.2.2 复位结构

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

WDTn 为 1 个 LRC 时钟周期宽度信号,是内部数字信号。但是在 Debug/Sleep 或者其它模式下，WDT 能否禁用根据芯片不同型号可选。

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。

在本芯片中，4 个复位信号复位等级和作用域一致。

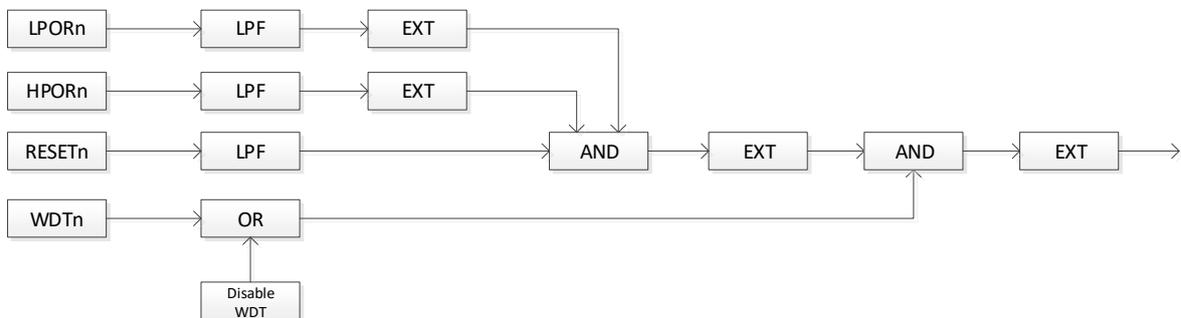


图 4-4 复位架构

4.2.3 复位记录

用于保存复位事件，该寄存器只能通过写入清空，无法被复位清除，可以方便的了解是否发生以及发生过何种复位。

4.3 寄存器

4.3.1 地址分配

系统模块寄存器基地址为 0x40000000。



表 4-4 系统控制寄存器

名称	偏移	说明
SYS_WDT_PSW	0x2C	看门狗密码寄存器
SYS_WDT_CLR	0x38	看门狗清零寄存器
Reserved	0x40~0x44	保留
SYS_AFE_CMP	0x48	比较器输出寄存器
Reserved	0x4C	保留
SYS_AFE_REG0	0x50	模拟配置寄存器 0
SYS_AFE_REG1	0x54	模拟配置寄存器 1
SYS_AFE_REG2	0x58	模拟配置寄存器 2
SYS_AFE_REG3	0x5C	模拟配置寄存器 3
SYS_AFE_REG4	0x60	模拟配置寄存器 4
SYS_AFE_REG5	0x64	模拟配置寄存器 5
SYS_AFE_REG6	0x68	模拟配置寄存器 6
SYS_AFE_REG7	0x6C	模拟配置寄存器 7
SYS_AFE_REG8	0x70	模拟配置寄存器 8
SYS_AFE_REG9	0x74	模拟配置寄存器 9
SYS_AFE_REGA	0x78	模拟配置寄存器 10
SYS_AFE_DAC	0x7C	DAC 数字量寄存器
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_IAP	0xA4	在线编程模式寄存器
SYS_TRIM	0xA8	校正模式寄存器
SYS_SFT_RST	0xAC	软复位寄存器
SYS_PROTECT	0xB0	写保护寄存器

4.3.2 看门狗密码寄存器 SYS_WDT_PSW

表 4-5 看门狗密码寄存器 SYS_WDT_PSW

名称	复位值	偏移	位置	权限	说明
SYS_WDT_PSW	0x0	0x2C	[31:16]	NA	未使用
			[15:0]	WO	写入 0xA6B4，才能对 WDT_CLR 进行写操作 对 WDT_CLR 进行写操作，将清除 WDT_PSW， 下一次写 WDT_CLR 需要再次写 WDT_PSW



4.3.3 看门狗清零寄存器 SYS_WDT_CLR

表 4-6 看门狗清零寄存器 SYS_WDT_CLR

名称	复位值	偏移	位置	权限	说明
SYS_WDT_CLR	0x0	0x38	[31:16]	NA	未使用
			[15:0]	WO	写入字节 16'b0111_1001_1000_11B ₁ B ₀ , 高 14 位为密码, 密码正确时, B[1:0]才能写入其中, B[1]为 MODE, 1: 16 秒复位 0: 0.5 秒复位 B[0]为 CLR, 写入 1, 则清空 WDT 计数器

4.3.4 比较器输出寄存器 SYS_AFE_CMP

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_CMP	0x48	[31:16]	NA		未使用
		[15:14]	NA		未使用
		[13]	R	CMP1_RESULT	CMP1 输出结果寄存器
		[12]	R	CMP0_RESULT	CMP0 输出结果寄存器
		[11:0]	NA		未使用

4.3.5 模拟寄存器概述

模拟寄存器的名称为 SYS_AFE_REG0~SYS_AFE_REGA, 对应地址为 0x40000050~0x40000078。其中地址 0x40000070~0x40000078 是模拟各个模块的校正寄存器, 这些寄存器在出厂之前都会将各自的校正值填入 Flash info 区, 并在上电后自动加载到 SYS_AFE_REG8~SYS_AFE_REGA。一般情况下用户不要去配置或改变这些值。如果需要某个模拟参数进行微调, 需要读取原校正值, 并以此为基础进行微调。

地址 0x40000050~0x4000006c 是开放给用户的寄存器, 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。

4.3.6 模拟配置寄存器 0 SYS_AFE_REG0

表 4-7 模拟配置寄存器 0 SYS_AFE_REG0

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG0	0x50	[31:16]	NA		未使用
		[15:14]	RW	IT_RBUF <1:0>	ADC 基准缓冲器偏置电流调节, 采用默认配置 00:×1; 01:×1.2;



					10:×1.5; 11:×2;
		[13:12]	RW	IT_ADCMP <1:0>	ADC CMP 偏置电流调节, 采用默认配置 00:×1; 01:×2; 10:×0.66; 11:×1;
		[11:10]	RW	IT_AMP<1:0>	ADC AMP 偏置电流调节, 采用默认配置 00:×1; 01:×1.5; 10:×0.75; 11:×1;
		[9:8]	RW	IT_OPA<1:0>	OPA 偏置电流调节, 采用默认配置 00:×1; 01:×1.2; 10:×1.5; 11:×2;
		[7:6]	RW	REF_OPA3<1:0>	运放 3 反馈电阻 00: 200k:10k; 01: 190k:20k; 10: 180k:30k; 11:170k:40k
		[5:4]	RW	RES_OPA2<1:0>	运放 2 反馈电阻 00: 200k:10k; 01: 190k:20k; 10: 180k:30k; 11:170k:40k
		[3:2]	RW	RES_OPA1<1:0>	运放 1 反馈电阻 00: 200k:10k; 01: 190k:20k; 10: 180k:30k; 11:170k:40k
		[1:0]	RW	RES_OPA0<1:0>	运放 0 反馈电阻 00: 200k:10k; 01: 190k:20k; 10: 180k:30k; 11:170k:40k



4.3.7 模拟配置寄存器 1 SYS_AFE_REG1

表 4-8 模拟配置寄存器 1 SYS_AFE_REG1

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG1	0x54	[31:16]	NA		未使用
		[15:4]	RW	Reserved	保留位, 需全部为'0'
		[3]	RW	GAIN_REF	ADC 基准电压调节, 采用默认配置 0:×1; 1:×2;
		[2:0]	RW	Reserved	保留位, 需全部为'0'

4.3.8 模拟配置寄存器 2 SYS_AFE_REG2

表 4-9 模拟配置寄存器 2 SYS_AFE_REG2

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG2	0x58	[31:16]	NA		未使用
		[15:14]	RW	CURRIT<1:0>	ADC 全局偏置电流调整 01:1; 00:-33%; 11:-66%; 10:-75%
		[13]	RW	OPAOUT_EN	使能运放负端信号输出, 用于测试使用, 正常工作时配置为 0 0: 不使能 1:使能
		[12]	RW	CUR_EN	使能共模调节, 采用默认配置 0: 不使能 1: 使能
		[11:10]	RW	CSEL<1:0>	晶体起振电容调节, 采用默认配置 10:+0pf; 11: +2pf; 00: +4pf; 01:+6pf;
		[9:8]	RW	XTRSEL<1:0>	晶体起振电路电阻调节, 采用默认配置 XTRSEL<1>=1: N 端阻值增加 XTRSEL<0>=1: P 端电阻增加一倍
		[7]	RW	TEMP_MODE	1:配置温度传感器模式, 开启测温



					时应配置为'1'
		[6:0]	RW	Reserved	保留位, 需全部为'0'

4.3.9 模拟配置寄存器 3 SYS_AFE_REG3

表 4-10 模拟配置寄存器 3 SYS_AFE_REG3

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG3	0x5C	[31:16]	NA		未使用
		[15]	RW	Reserved	保留位, 需全部为'0'
		[14:12]	RW	CMP1_SEL<2:0>	比较器 1 信号正端选择 000: 连 CMP1_IP0 001: 连 OPA3_IP 010: 连 OPA2_OUT 011: 连 OPA3_OUT 100: 连 CMP1_IP1 101: 连 CMP1_IP2 110: 连 CMP1_IP3 111: 连 AVSS 说明: 上述除 AVSS/OPA2_OUT/OPA3_OUT 外都为管脚名称, 请参看 datasheet 里管脚定义章节
		[11]	RW	DACOUT_EN1	DAC 输出到 IO 使能 1:使能输出到 IO ADC0_CH7/P0.3
		[10:8]	RW	CMP0_SEL<2:0>	比较器 0 信号正端选择 000: 连 CMP0_IP0 001: 连 OPA0_IP 010: 连 OPA0_OUT 011: 连 OPA1_OUT 100: 连 CMP0_IP1 101: 连 CMP0_IP2 110: 连 CMP0_IP3 111: 连 CMP0_IP4 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]	RW	CMP_HYS	比较器回差选择, 采用默认配置 0: 20mv;



				1: 0mv
		[6]	RW	REF_BGP_EN REF BGP 输出使能, 测试 REF BGP 的时候使用, 正常情况采用默认配置 0: 不输出 1: 使能 REF BGP 输出到 CMP1_IP2 管脚
		[5:4]	RW	CMP1_SELN<1:0> 比较器 1 信号负端选择 00: 连 CMP1_IN 01: 连 REF BGP 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 BGP 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]	RW	CMP0_SELN<1:0> 比较器 0 信号负端选择 00: 连 CMP0_IN 01: 连 REF BGP 10: 连 DAC 输出 11: 连 HALLO_MID 说明: 上述 CMP0_IN 为管脚名称, 请参看 datasheet 里管脚定义章节; REF BGP 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; HALLO_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]	RW	DAC12B_FS 12BIT DAC 量程选择 0: 满量程 0~3.1V; 1: 满量程 0~1V
		[0]	RW	DACOUT_EN DAC 输出到 IO 使能 0: 不输出 1: 使能输出到 IO P2.3



4.3.10 模拟配置寄存器 4 SYS_AFE_REG4

表 4-11 模拟配置寄存器 4 SYS_AFE_REG4

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG4	0x60	[31:16]	NA		未使用
		[15:2]	RW	Reserved	保留位, 需全部为'0'
		[1]	RW	IT_CMP<1>	比较器 1 延时选择 0: 0.15us; 1: 0.6us
		[0]	RW	IT_CMP<0>	比较器 0 延时选择 0: 0.15us; 1: 0.6us

4.3.11 模拟配置寄存器 5 SYS_AFE_REG5

表 4-12 模拟配置寄存器 5 SYS_AFE_REG5

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG5	0x64	[31:16]	NA		未使用
		[15]	RW	PLLPD	PLL 关闭使能 0:开启 1:关闭
		[14]	RW	XTALPDN	晶体起振电路开启使能 0:关闭 1:开启
		[13]	RW	TMPPDN	温度传感器开启使能 0:关闭 1:开启
		[12]	RW	DAC12BPDN	12BIT DAC 开启使能 0:关闭 1:开启
		[11]	RW	Reserved	保留位, 需全部为'0'
		[10]	RW	RCHPD	RCH 时钟关闭使能 0:开启 1:关闭
		[9]	RW	Reserved	保留位, 需全部为'0'
		[8]	RW	BGPPD	BGP 关闭使能 0:开启 1:关闭
		[7]	RW	CMP1PDN	CMP1 开启使能 0:关闭 1:开启



		[6]	RW	CMP0PDN	CMP0 开启使能 0:关闭 1:开启
		[5]	RW	OPA3PDN	OPA3 开启使能 0:关闭 1:开启
		[4]	RW	OPA2PDN	OPA2 开启使能 0:关闭 1:开启
		[3]	RW	OPA1PDN	OPA1 开启使能 0:关闭 1:开启
		[2]	RW	OPA0PDN	OPA0 开启使能 0:关闭 1:开启
		[1]	RW	ADC1PDN	ADC1 开启使能 0:关闭 1:开启
		[0]	RW	ADC0PDN	ADC0 开启使能 0:关闭 1:开启

4.3.12 模拟配置寄存器 6 SYS_AFE_REG6

表 4-13 模拟配置寄存器 6 SYS_AFE_REG6

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG6	0x68	[31:16]	NA		未使用
		[15]	RW	PLLSR_SEL	PLL 时钟源选择 0:使用 RCH 作为输入时钟源; 1:使用 XTAL OSC 作为输入时钟源
		[14:13]	RW	Reserved	保留位, 需全部为'0'
		[12]	RW	TMPCOFF	1:关闭温度传感器的时钟, 开启测温时需配置为'1'
		[11:10]	RW	Reserved	保留位, 需全部为'0'
		[9:8]	RW	PVDSEL<1:0>	电源掉电监测阈值选择 00: 4.5V; 01: 4.2V; 10: 3.9V; 11: 3.6V
		[7]	RW	LDO33PD	关闭 LDO33, 使其进入跟随模式, 即 LDO33 的输出跟随 AVDD 管脚输入电压



					0: 开启 1: 关闭, 进入跟随模式
		[6:5]	RW	Reserved	保留位, 需全部为'0'
		[4]	RW	LDO3IT	LDO33 偏置电流调节 0: 保持不变 1: x2, 增加一倍
		[3]	RW	Reserved	保留位, 需全部为'0'
		[2]	RW	PDT_HYS	掉电检测回差, 采用默认配置 0: 打开回差功能 1: 关闭回差功能
		[1]	RW	VSR_PDT	掉电检测基准源选择, 其中低功耗基准源为 1.3V 左右, DAC 输出则可以通过软件配置 0: 选择低功耗基准源; 1: 选择 DAC 输出
		[0]	RW	PD_PDT	关闭掉电检测电路 0: 开启 1: 关闭

4.3.13 模拟配置寄存器 7 SYS_AFE_REG7

表 4-14 模拟配置寄存器 7 SYS_AFE_REG7

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG7	0x6C	[31:16]	NA		未使用
		[15:6]	RW	Reserved	保留位, 需全部为'0'
		[5:4]	RW	ADCLKSEL<1:0>	ADC 时钟频率选择 00: 48MHz; 01: 24MHz ; 10: 12MHz ; 11: 6MHz
		[3:0]	RW	Reserved	保留位, 需全部为'0'

4.3.14 模拟配置寄存器 8 SYS_AFE_REG8

表 4-15 模拟配置寄存器 8 SYS_AFE_REG8

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG8	0x70	[31:16]	NA		未使用
		[15:12]	RW	RCLTRIM_L<3:0>	32kHz RC 时钟输出频率调节 0000: +0%; 0001: +7.5%;



					0100:-6.5%; 0010:+15%; 1000:-13%; 0011:+22%; 1100:-19%;
		[11]	RW	Reserved	保留位, 需全部为'0'
		[10]	RW	FDER	减小 RCH 频率 0:不变 1:减小约 12%
		[9:8]	RW	VREFTRIM<1:0>	VREF 输出电压调节, 将等比例影响 LDO33/15 的输出值 00: 0; 01: +3.2%; 10: -7.2%; 11: -3.6%
		[7:0]	RW	Reserved	保留位, 需全部为'0'

4.3.15 模拟配置寄存器 9 SYS_AFE_REG9

表 4-16 模拟配置寄存器 9 SYS_AFE_REG9

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REG9	0x74	[31:16]	NA		未使用
		[15:14]	RW	REFTRIM_L<1:0>	REF BGP 输出电压粗调 00:0mV; 01:-45mV; 10:+90mV; 11:+45mV;
		[13:11]	RW	LDO33TRIM<2:0>	LDO33 输出电压调节 011:+9%; 010:+6%; 001: +3%; 000: +0%; 111: -3%; 110: -6%; 101: +9%; 100: +9% ;
		[10:8]	RW	LDO15TRIM<2:0>	LDO15 输出电压调节 011:+20%; 010:+13.4%; 001: +6.7%; 000: +0%; 111: -6.7%;



					110: -13.4%; 101: +20%; 100: +20%;
		[7:6]	RW	Reserved	保留位, 需全部为'0'
		[5:0]	RW	RCHTRIM<5:0>	4MHz RC 时钟输出频率调节 000000~011111: 频率对应+0~+38.75%每加一档频率增加 1.25%; 100000~111111: 频率对应-40~-1%每加一档频率增加 1.25%;

4.3.16 模拟配置寄存器 10 SYS_AFE_REGA

表 4-17 模拟配置寄存器 10 SYS_AFE_REGA

名称	偏移	位置	权限	寄存器名	说明
SYS_AFE_REGA	0x78	[31:16]	NA		未使用
		[15:7]	RW	Reserved	保留位, 需全部为'0'
		[6:4]	RW	FREFTRIM<2:0>	Flash REF 输出电压细调 芯片出厂已做校正, 请勿修改校正
		[3]	RW	REF_LTRIM	REF BGP 输出电压粗调 0: 0mV; 1: -32mV;
		[2:0]	RW	REFTRIM<2:0>	REF BGP 输出电压细调 000:0mV; 001:+6.6mV; 010:+13.2mV; 011:+19.8mV; 100:-26.4mV; 101:-19.8mV; 110:-13.2mV; 111:-6.6mV;

4.3.17 DAC 数字量寄存器 SYS_AFE_DAC

表 4-18 DAC 数字量寄存器 SYS_AFE_DAC

名称	复位值	偏移	位置	权限	说明
----	-----	----	----	----	----



SYS_AFE_DAC	0x0	0x7C	[31:12]	NA	未使用
			[11:0]	RW	DAC 待转换的数字量输入

4.3.18 时钟控制寄存器 SYS_CLK_CFG

表 4-19 时钟控制寄存器 SYS_CLK_CFG

名称	复位值	偏移	位置	权限	说明
SYS_CLK_CFG	0x0	0x80	[31:16]	NA	未使用
			[15:8]	RW	当 B[15:8]为全零时系统在 1/8 分频的基础上再进行 1/4 分频,复位后系统使用此默认配置,即 96MHz/8/4=3MHz 时钟。 如果要选择 PLL 时钟, B[8]应为 1
			[7:0]	RW	PLL 输出分频控制,选择 8 个时钟周期中,哪些周期输出时钟,例如 8'b00000001 表示 1/8 分频, 8'b00010001 表示 2/8,即 1/4 分频, 8'b00100101 表示 3/8 分频,但不均匀 8'b01010101 表示 4/8 分频,即 1/2 分频 8'b01010111 表示 5/8 分频 8'b01011111 表示 6/8 分频,即 3/4 分频 8'b01111111 表示 7/8 分频, 一般地,8bit 中只要存在 n 个 1,即是 n/8 分频,1 的位置没有要求

4.3.19 复位控制寄存器 SYS_RST_CFG

表 4-20 复位控制寄存器 SYS_RST_CFG

名称	复位值	偏移	位置	权限	说明
SYS_RST_CFG	0x0	0x84	[31:5]	NA	未使用
			[4:2]	RW	休眠唤醒间隔设置 000: 0.125S 100: 2S 001: 0.25S 101: 4S 010: 0.5S 110: 8S 011: 1S 111: 16S
			[1]	NA	未使用
			[0]	RW	看门狗使能,高有效。

4.3.20 复位源记录寄存器 SYS_RST_SRC

表 4-21 复位源记录寄存器 SYS_RST_SRC

名称	复位值	偏移	位置	权限	说明
SYS_RST_SRC	0x0	0x88	[31:4]	NA	未使用



			[3]	RO	看门狗复位发生标志，高有效
			[2]	RO	按键复位发生标志，高有效
			[1]	RO	HPOR 复位发生标志，高有效
			[0]	RO	LPOR 复位发生标志，高有效

4.3.21 复位源记录清除寄存器 SYS_CLR_RST

表 4-22 复位源记录清除寄存器 SYS_CLR_RST

名称	复位值	偏移	位置	权限	说明
SYS_CLR_RST	0x0	0x8C	[31:16]	NA	未使用
			[15:0]	WO	写入 0xDEAD，清除复位标志记录 请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

4.3.22 外设时钟分频寄存器 0 SYS_CLK_DIV0

表 4-23 外设时钟分频寄存器 0SYS_CLK_DIV0

名称	复位值	偏移	位置	权限	说明
SYS_CLK_DIV0	0x0	0x90	[31:16]	NA	未使用
			[15:0]	RW	SPI/I2C 工作时钟=MCLK/(CLK_DIV0+1)，其中 MCLK 由 SYS_CLK_CFG 分频系数决定

4.3.23 外设时钟分频寄存器 2 SYS_CLK_DIV2

表 4-24 外设时钟分频寄存器 2SYS_CLK_DIV2

名称	复位值	偏移	位置	权限	说明
SYS_CLK_DIV2	0x0	0x98	[31:16]	NA	未使用
			[15:0]	RW	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

4.3.24 外设时钟门控寄存器 SYS_CLK_FEN

表 4-25 外设时钟门控寄存器 SYS_CLK_FEN

名称	复位值	偏移	位置	权限	说明
SYS_CLK_FEN	0x0	0x9C	[31:6]	NA	未使用
			[5]	RW	UART1 时钟门控，1:使能；0:禁用
			[4]	RW	UART0 时钟门控，1:使能；0:禁用
			[3]	RW	MCPWM 时钟门控，1:使能；0:禁用



			[2]	RW	通用定时器(UTIMER)时钟门控, 1:使能; 0:禁用
			[1]	RW	霍尔模块(HALL)时钟门控, 1:使能; 0:禁用
			[0]	RW	SPI/I2C 模块时钟门控, 1:使能; 0:禁用

4.3.25 休眠寄存器 SYS_CLK_SLP

表 4-26 休眠寄存器 SYS_CLK_SLP

名称	复位值	偏移	位置	权限	说明
SYS_CLK_SLP	0x0	0xA0	[31:16]	NA	未使用
			[15:0]	WO	写入密码 0xDEAD, 系统关闭高速时钟, 进入休眠状态

4.3.26 在线编程模式寄存器 SYS_IAP

表 4-27 在线编程模式寄存器 SYS_IAP

名称	复位值	偏移	位置	权限	说明
SYS_IAP	0x0	0xA4	[31:16]	NA	未使用
			[15:0]	RW	写入 0x3721, 软复位后, 芯片进入 IAP 模式; 写入 0x3720, 软复位后, 芯片进入正常工作模式。

4.3.27 校正模式寄存器 SYS_TRIM

表 4-28 校正模式寄存器 SYS_TRIM

名称	复位值	偏移	位置	权限	说明
SYS_TRIM	0x0	0xA8	[31:1]	NA	未使用
			[0]	RO	芯片复位后, 进入 TRIM 模式 TRIM 结束后, 通过软复位退出 TRIM 模式

4.3.28 软复位寄存器 SYS_SFT_RST

表 4-29 软复位寄存器 SYS_SFT_RST

名称	复位值	偏移	位置	权限	说明
SYS_SFT_RST	0x0	0xAC	[31:6]	NA	未使用
			[5]	WO	UART1 软复位, 写 1 触发, 再写 0 释放
			[4]	WO	UART0 软复位, 写 1 触发, 再写 0 释放
			[3]	WO	MCPWM 软复位, 写 1 触发, 再写 0 释放
			[2]	WO	UTIMER 软复位, 写 1 触发, 再写 0 释放
			[1]	WO	HALL 软复位, 写 1 触发, 再写 0 释放
			[0]	WO	SPI/I2C 软复位, 写 1 触发, 再写 0 释放



4.3.29 写保护寄存器 SYS_PROTECT

表 4-30 保护寄存器 SYS_PROTECT

名称	复位值	偏移	位置	权限	说明
SYS_PROTECT	0x0	0xB0	[31:16]	NA	未使用
			[15:0]	WO	除看门狗相关、SYS_AFE_REG3、SYS_AFE_DAC 外，其他系统寄存器受写保护，写入前需先写入密码解除写保护 写入 0x7a83，使能寄存器写操作 写入其它值，禁止寄存器写操作



5 FLASH

5.1 概述

FLASH 控制器模块，主要实现对容量为 32KB 的 FLASH 进行读/写/擦除操作。

FLASH 模块总体结构如下：

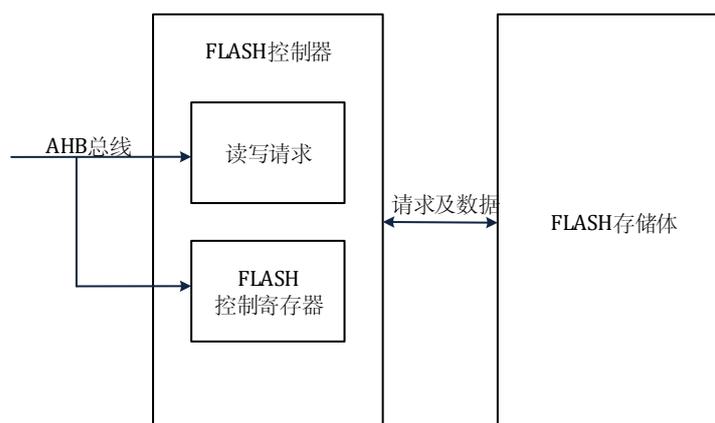


图 5-1 Flash 模块结构框图

FLASH 模块由两个部分组成：FLASH 控制器和 FLASH 存储体。本文档主要描述 FLASH 控制器。

FLASH 控制器，实现了芯片内部资源通过 AHB 总线对 FLASH 存储体的访问。

5.2 功能特点

- WORD (32-BIT) 对齐方式，进行读取
- WORD (32-BIT) 对齐方式，进行写入
- SECTOR/BLOCK/FULL CHIP 三种方式，进行擦除操作
- FLASH 控制寄存器的访问
- FLASH 预取操作
- FLASH 加密保护
- FLASH 在线升级 (IAP)

5.2.1 访问操作

5.2.1.1 FLASH 读取操作

执行对 FLASH 存储体的读取操作。FLASH 一次性读取出 32-BIT 宽的数据。系统有两种方法读取 FLASH 数据。第一，MCU 执行取值或者取数据的指令；第二，MCU 通过访问 FLASH 寄存器 (FLASH_ADDR 和 FLASH_RDATA) 间接读取 FLASH 数据。第二种访问流程如下：



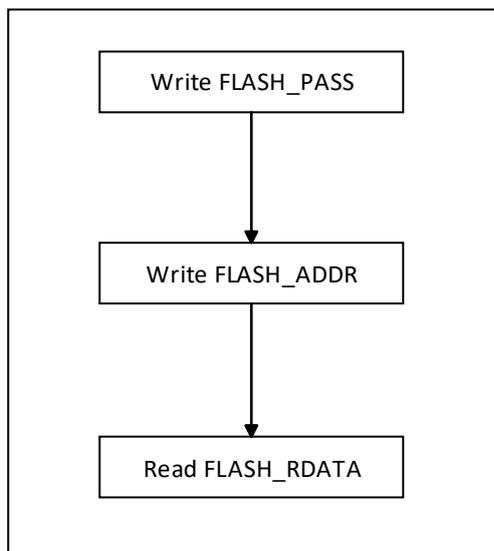


图 5-2 Flash 模块读操作流程图

5.2.1.2 FLASH 写入操作

执行对 FLASH 存储体的写入操作。一般而言，我们先执行擦除操作，然后才能执行数据写入操作。FLASH 写入流程如下。

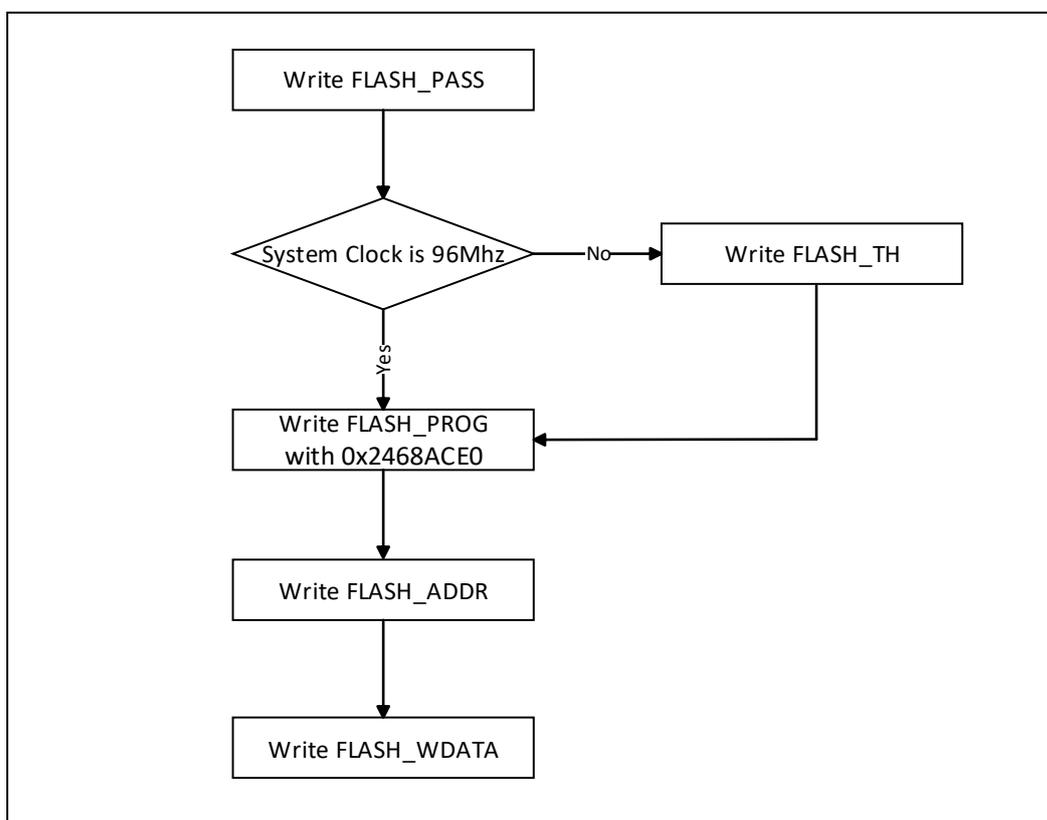


图 5-3 Flash 模块写操作流程图

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_TH` 寄存器默认值是 96MHz 时钟频率下的计数值；其它时钟频率时，`FLASH_TH` 寄存器的值需要等比例变化。另外，在执行 `FLASH` 的写入/擦除操作时，MCU 将暂停工作直至 `FLASH` 的写入/擦除操作完毕。

图 5-3 仅展示了一次写入的流程。若执行连续写入时，上述流程将比较冗长，可以在写 `FLASH_ADDR` 寄存器前，配置 `FLASH_CFG` 的 `BIT8`，开启地址自动递增模式（每次增加 `0x4`），后续只需要反复写 `FLASH_WDATA` 寄存器即可。对于连续读，操作类似。连续写的流程如下



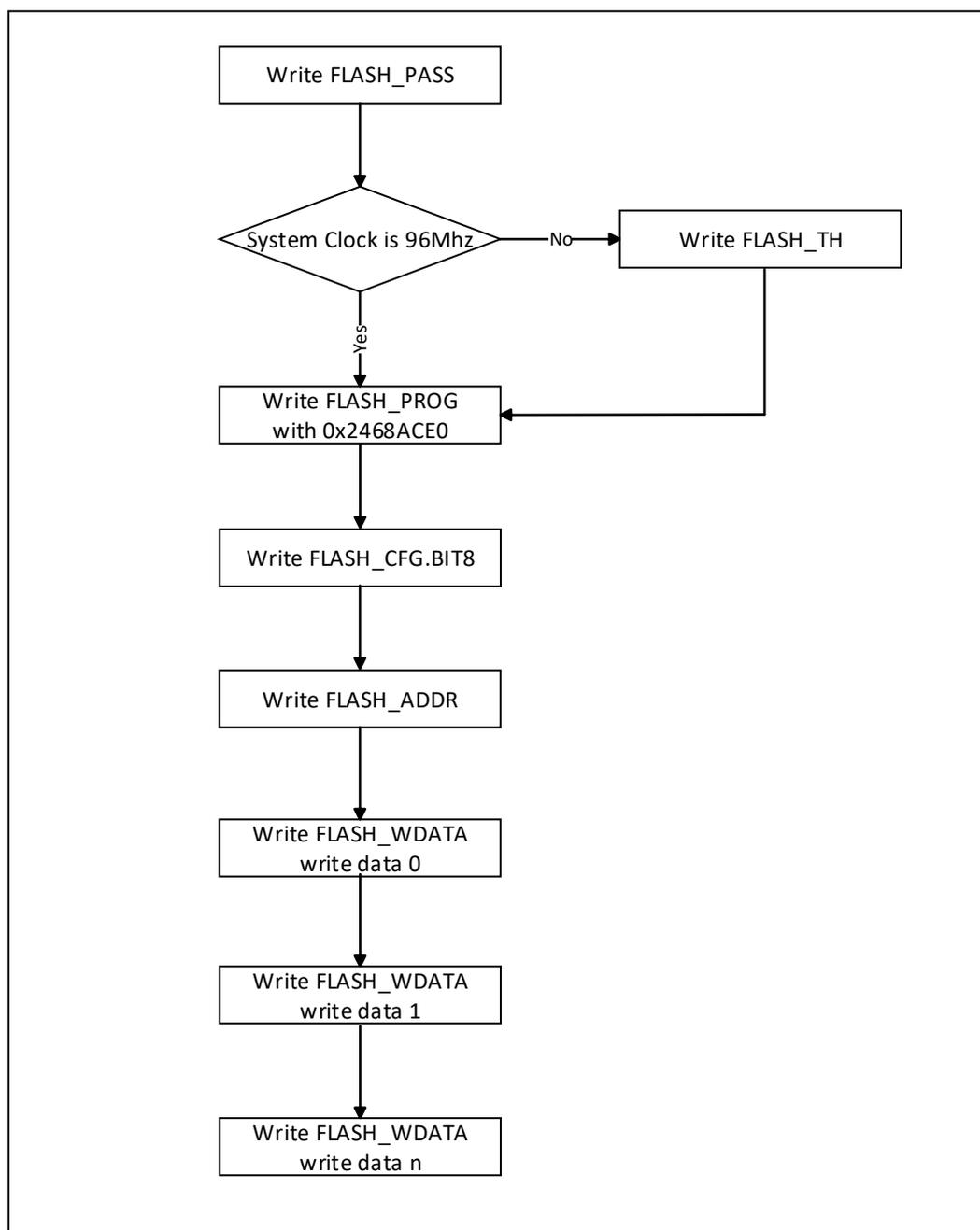


图 5-4 Flash 模块连续写操作流程图

5.2.1.3 FLASH 擦除操作

执行对 FLASH 存储体的擦除操作。擦除分成 Sector/Block 和 FullChip。分别对应，256Byte 的擦除，2KB 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 5-1FLASH 地址分配表

Name	Addresses	Size(Bytes)
------	-----------	-------------



Block 0	0x0000 0000 - 0x0000 07FF	2KB
Block 1	0x0000 0800 - 0x0000 0FFF	2KB
Block 2	0x0000 1000 - 0x0000 17FF	2KB
...
Block 15	0x0000 7800 - 0x0000 7FFF	2KB

Name	Addresses	Size(Bytes)
Sector 0	0x0000 0000 - 0x0000 00FF	256
Sector 1	0x0000 0100 - 0x0000 01FF	256
Sector 2	0x0000 0200 - 0x0000 02FF	256
...
Sector 127	0x0000 7F00 - 0x0000 7FFF	256

FLASH_CFG 寄存器的 BIT0 和 FLASH_ERASE_OP 的 BIT4，共同决定本次擦除的类型。

表 5-2FLASH 擦除类型表

	FLASH_ERASE_OP.BIT4	FLASH_CFG.BIT0
Secotor	0	0
Block	0	1
Full Chip	1	X

FLASH 擦除操作流程如下所示。



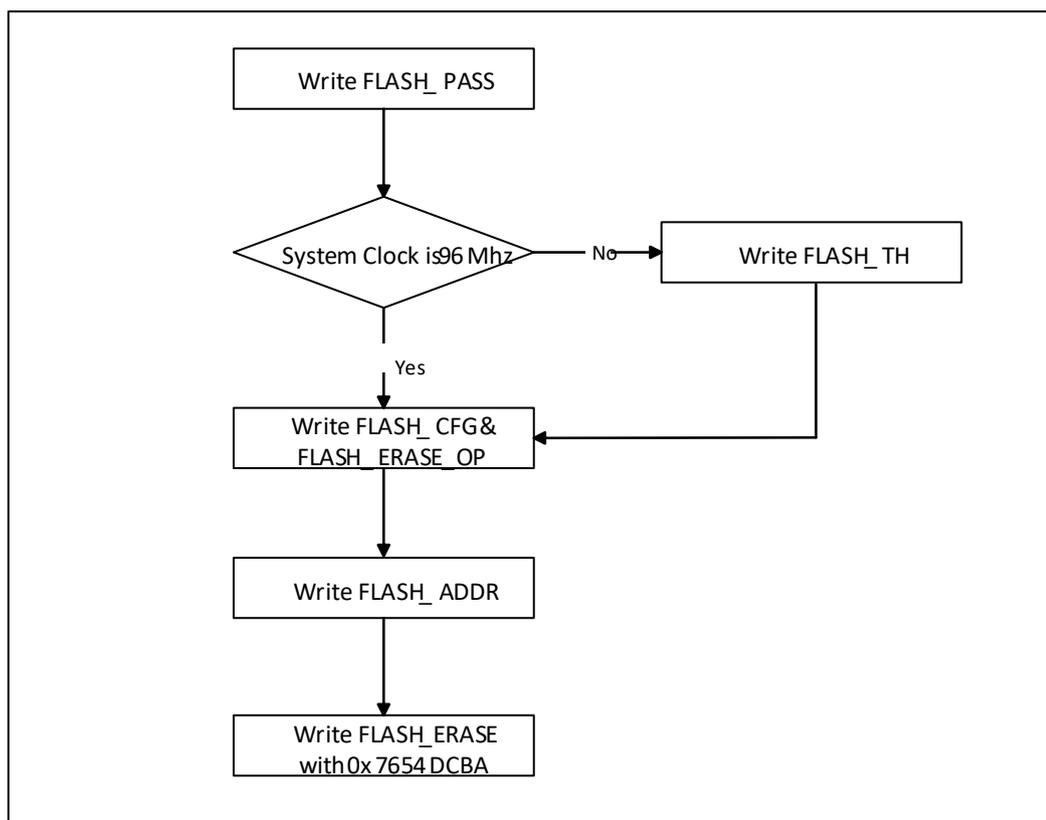


图 5-5 Flash 模块擦除操作流程

若选择 Block 或者 Sector 擦除,需要通过 FLASH_ADDR 确定哪个 Block 或者 Sector 被擦除,若是 Full Chip 模式的话,FLASH_ADDR 的值将失效。FLASH_ERASE 只有写入 0x7654DCBA 才能正常触发擦除操作,写入其它值将无法触发擦除操作。

5.2.1.4 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 的 BIT16 即可。

5.2.1.5 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。

FLASH 存储体共有 32KB 大小,其中最后一个 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。具体加密执行的流程如下。



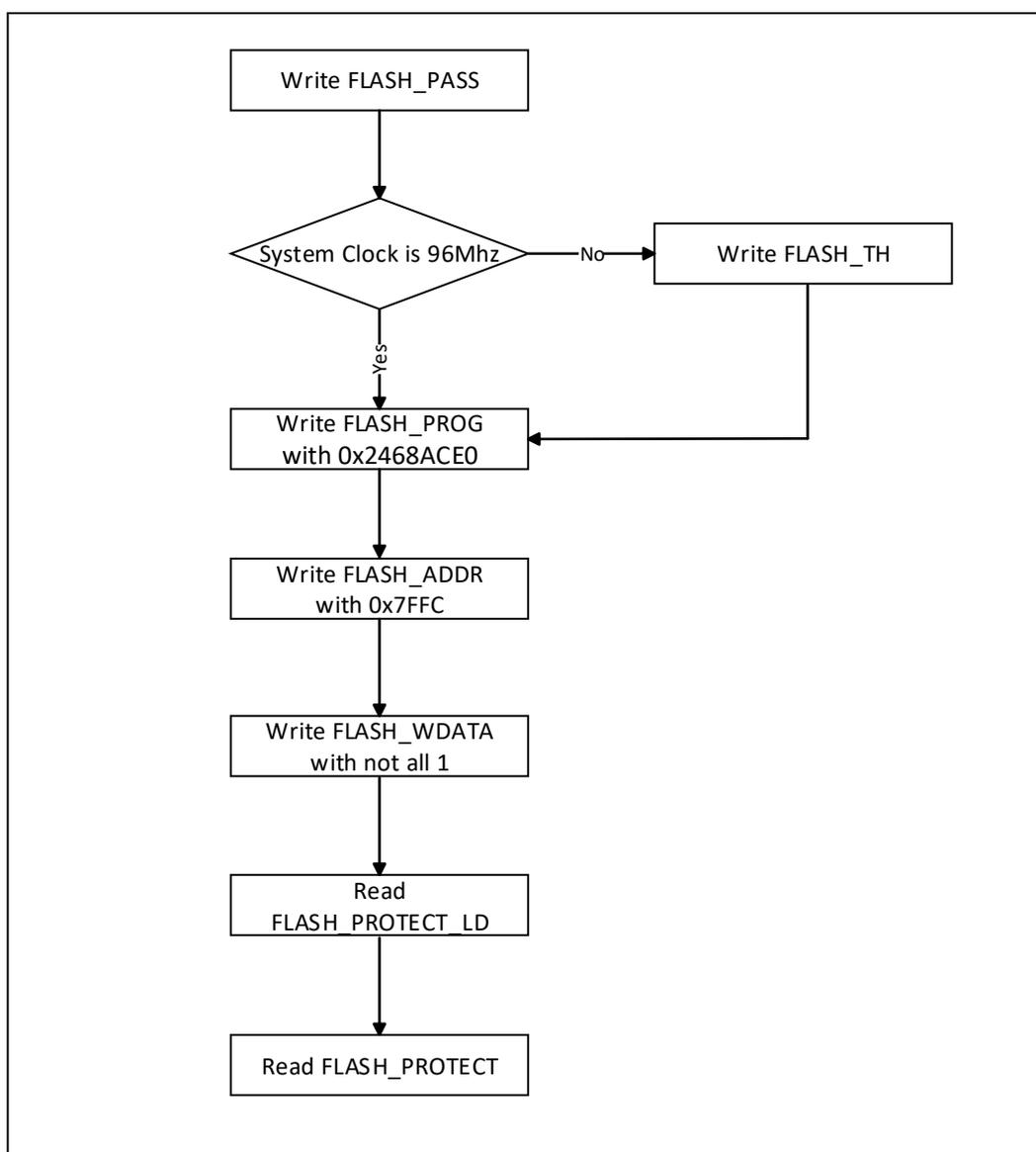


图 5-6 Flash 模块加密操作流程

将 FLASH 存储体的最后一个 WORD 写入非全 1 值，然后读取 FLASH_PROTECT_LD 寄存器，更新加密状态位，最后可以通过读取 FLASH_PROTECT 寄存器检查是否加密。若是加密状态，FLASH_PROTECT 返回值为 1。

对应的解密流程，只能将 FLASH 执行 Full Chip 擦除才能解除。具体如下

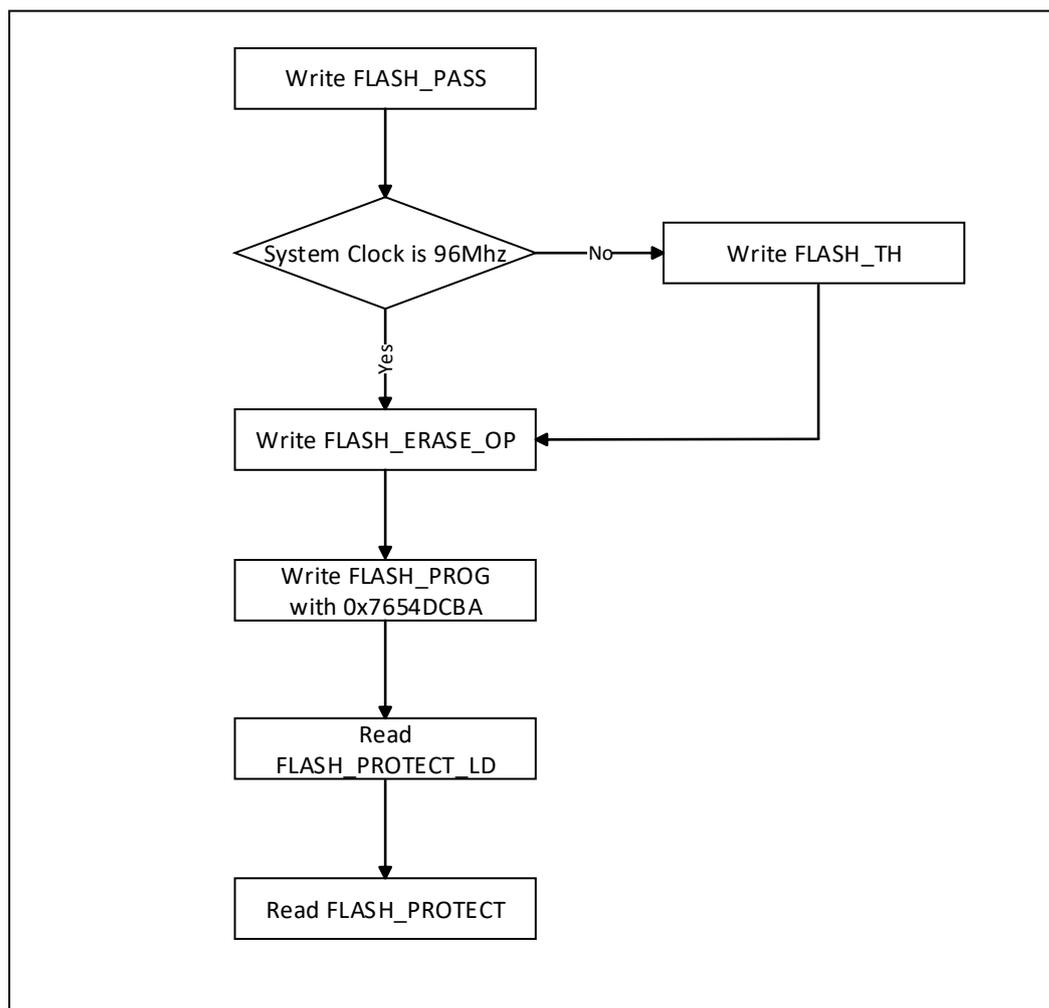


图 5-7 Flash 模块解密操作流程

先将 FLASH 执行 Full Chip 擦除，然后读取 FLASH_PROTECT_LD 寄存器，更新加密状态位，最后可以通过读取 FLASH_PROTECT 寄存器检查是否解密。若是解密状态，FLASH_PROTECT 返回值为 0。

切记，要获得最新的加密/解密状态，均需先读取 FLASH_PROTECT_LD 寄存器，完成状态位的更新操作。

5.2.1.6 FLASH 在线升级(IAP)

FLASH 的大小为 32KB。FLASH 划分成两个区域：2KB 区域和 30KB 区域。

2KB 区域从实际物理空间的 0x0000_0000 开始至 0x0000_07FF

30KB 区域从实际物理空间的 0x0000_0800 开始至 0x0000_7FFF

若用户关闭 IAP 功能，则 2KB + 30KB，全部是用户代码空间。

若用户使能 IAP 功能，则 2KB 为 IAP 空间，30KB 为用户代码空间。系统发送的访问 FLASH 的



地址，经过 FLASH 控制器后，将被自动加上 2KB 偏移。意味着，系统实际是从 2KB 开始执行的。简单说，在线升级实现了--是否重映射 FLASH 的 32KB 空间。



图 5-8 在线升级空间映射关系

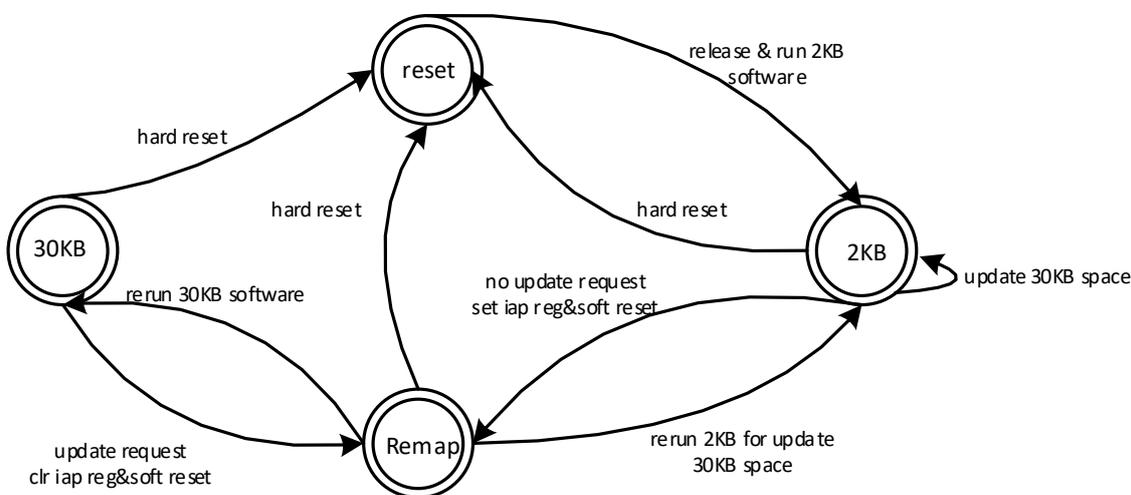


图 5-9 在线升级流程转换图

若用户使用 IAP 功能。整个 32KB 空间有两段程序，IAP 的 2KB 空间的程序。后面 30KB 存放用户正常使用的程序。此时需要注意的是，用户正常使用的程序在执行过程中，若需要通过 FLASH 寄存器间接访问 FLASH 数据，在配置 FLASH_ADDR 时，需要考虑 2KB 的偏移。例如，若访问 0x0000_1000，实际访问的是 0x0000_1800。受此影响，加密操作原本是改写 0x0000_7FFC 地址，此时只能写成 0x0000_77FC。

5.3 寄存器

5.3.1 地址分配

FLASH 控制器模块寄存器的基地址是 0x4000_0100 寄存器列表

表 5-3FLASH 控制寄存器

名称	偏移	说明
FLASH_TH	0x00	不同频率下，FLASH 擦除/写入的时间参数配置寄存器



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_CFG	0x10	配置寄存器
FLASH_PROG	0x14	编程寄存器
FLASH_PASS	0x20	FLASH 寄存器配置使能寄存器
FLASH_ERASE	0x24	擦除使能寄存器
FLASH_ERASE_OP	0x28	擦除类型寄存器
FLASH_PROTECT	0x2C	FLASH 保护状态寄存器
FLASH_PROTECT_LD	0x30	FLASH 保护状态更新寄存器
FLASH_READY	0x34	FLASH 闲忙状态寄存器

5.3.2 擦除/写入时间参数配置寄存器 FLASH_TH

表 5-4 擦除/写入时间参数配置寄存器 FLASH_TH

名称	复位值	偏移	位置	权限	说明
FLASH_TH	0x40F8F078	0x00	[31:24]	RW	写入时间, 默认为 96Mhz 配置值, 其它频率等比例调整
			[23:16]	RW	写入建立时间, 默认为 96Mhz 配置值, 其它频率等比例调整
			[15:8]	RW	擦除时间, 默认为 96Mhz 配置值, 其它频率等比例调整
			[7:0]	RW	擦除建立时间, 默认为 96Mhz 配置值, 其它频率等比例调整

5.3.3 地址寄存器 FLASH_ADDR

表 5-5 地址寄存器 FLASH_ADDR

名称	复位值	偏移	位置	权限	说明
FLASH_ADDR	0x0	0x04	[31:15]	NA	未使用, 默认写 0
			[14:0]	RW	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 256-Byte, 一个 Block 是 2048-Byte。若执行 Sector 擦除, 地址需要是 256 的整数倍 (若带偏移, 偏移量会被忽略)。同理, 适用于 Block 擦除。全芯片擦除, 不会参考这个寄存器的值



5.3.4 写数据寄存器 FLASH_WDATA

表 5-6 写数据寄存器 FLASH_WDATA

名称	复位值	偏移	位置	权限	说明
FLASH_WDATA	0x0	0x08	[31:0]	RW	执行写入操作，写入 FLASH 的值

5.3.5 写数据寄存器 FLASH_RDATA

表 5-7 写数据寄存器 FLASH_RDATA

名称	复位值	偏移	位置	权限	说明
FLASH_RDATA	0x0	0x0C	[31:0]	R	执行读取操作，读出 FLASH 的值

5.3.6 控制寄存器 FLASH_CFG

表 5-8 控制寄存器 FLASH_CFG

名称	复位值	偏移	位置	权限	说明
FLASH_CFG	0x0	0x10	[31:17]	NA	未使用，默认写 0
			[16]	RW	预取使能。1，使能预取；0，关闭预取。
			[15:9]	NA	未使用，默认写 0
			[8]	RW	地址自动递增。1，自动递增；0，不自动递增。
			[7:1]	NA	未使用，默认写 0
			[0]	RW	块擦除使能。1，使能；0，关闭。

5.3.7 写入控制寄存器 FLASH_PROG

表 5-9 写入控制寄存器 FLASH_PROG

名称	复位值	偏移	位置	权限	说明
FLASH_PROG	0x0	0x14	[31:0]	RW	写入 0x2468ACE0，使能写入操作；写入其它值，关闭写入操作。 若写入操作使能，读取该寄存器，返回 1；否则，返回 0。



5.3.8 写入保护寄存器 FLASH_PASS

表 5-10 写入保护寄存器 FLASH_PASS

名称	复位值	偏移	位置	权限	说明
FLASH_PASS	0x0	0x20	[31:1]	NA	未使用，默认写 0
			[0]	RW	FLASH 寄存器写入保护。1，使能配置 FLASH 寄存器；0，关闭配置 FLASH 寄存器。

5.3.9 擦除控制寄存器 FLASH_ERASE

表 5-11 擦除控制寄存器 FLASH_ERASE

名称	复位值	偏移	位置	权限	说明
FLASH_ERASE	0x0	0x24	[31:0]	RW	写入 0x7654DCBA，触发擦除操作；写入其它值，无效。

5.3.10 擦除类型寄存器 FLASH_ERASE_OP

表 5-12 擦除类型寄存器 FLASH_ERASE_OP

名称	复位值	偏移	位置	权限	说明
FLASH_ERASE_OP	0x0	0x28	[31:5]	NA	未使用，默认写 0
			[4]	RW	1，全芯片擦除模式；0，非全芯片擦除模式。
			[3:0]	NA	未使用，默认写 0

5.3.11 加密状态寄存器 FLASH_PROTECT

表 5-13 加密状态寄存器 FLASH_PROTECT

名称	复位值	偏移	位置	权限	说明
FLASH_PROTECT	0x0	0x2C	[31:1]	NA	未使用
			[0]	R	FLASH 加密状态。1，加密；0，解密。

5.3.12 加密状态更新寄存器 FLASH_PROTECT_LD

表 5-14 加密状态更新寄存器 FLASH_PROTECT_LD

名称	复位值	偏移	位置	权限	说明
----	-----	----	----	----	----



FLASH_PROTECT_LD	0x0	0x30	[31:0]	R	读该寄存器更新加密状态，返回值无效
------------------	-----	------	--------	---	-------------------

5.3.13 工作状态寄存器 FLASH_READY

表 5-15 工作状态寄存器 FLASH_READY

名称	复位值	偏移	位置	权限	说明
FLASH_READY	0x0	0x34	[31]	R	1:FLASH 处于闲状态; 0:FLASH 处于忙状态
			[30:0]	NA	未使用



6 通用 IO (GPIO)

6.1 概述

LSK32MC061C6T8 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。P0.15 ~ P0.16 个 GPIO 可以用作外部中断源输入。

6.1.1 功能框图

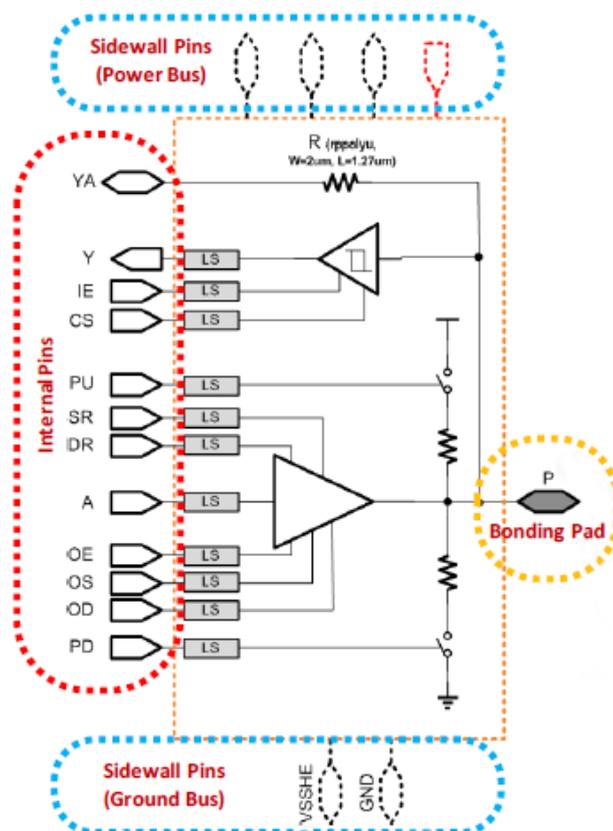


图 6-1 GPIO 功能框图

6.1.2 产品特点

- 3 组 16bit GPIO
- 支持上拉、下拉、开漏
- 支持配置锁定保护
- 支持外部中断
- 支持 GPIO 唤醒



6.2 寄存器

6.2.1 地址分配

GPIO 0 模块在芯片中的基地址是 0x4000_3700。

GPIO 1 模块在芯片中的基地址是 0x4000_3740。

GPIO 2 模块在芯片中的基地址是 0x4000_3780。

GPIO 0/1/2 的寄存器定义完全相同，仅基地址不同。

表 6-1 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_PDE	0x14	GPIO x 下拉使能
GPIOx_PODE	0x18	GPIO x 开漏使能
GPIOx_LCKR	0x1C	GPIO x 配置锁定
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

GPIO 中断/唤醒/配置锁定模块在 LSK32MC061C6T8 中的基地址是 0x4000_37C0。

表 6-2 GPIO 中断/唤醒/配置锁定模块寄存器列表

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
LCKR_PRT	0x0C	GPIO 保护锁定配置
WAKE_POL	0x10	GPIO 唤醒信号极性
WAKE_EN	0x14	GPIO 唤醒使能



6.2.2 GPIOx_PIE

表 6-3GPIOx 输入使能寄存器 GPIOx_PIE

名称	复位值	偏移	位置	权限	说明
GPIOx_PIE	0x0	0x00	[31:16]	NA	未使用
			[15]	RW	GPIO x[15] / Px[15] 输入使能
			[14]	RW	GPIO x[14] / Px[14] 输入使能
			[13]	RW	GPIO x[13] / Px[13] 输入使能
			[12]	RW	GPIO x[12] / Px[12] 输入使能
			[11]	RW	GPIO x[11] / Px[11] 输入使能
			[10]	RW	GPIO x[10] / Px[10] 输入使能
			[9]	RW	GPIO x[9] / Px[9] 输入使能
			[8]	RW	GPIO x[8] / Px[8] 输入使能
			[7]	RW	GPIO x[7] / Px[7] 输入使能
			[6]	RW	GPIO x[6] / Px[6] 输入使能
			[5]	RW	GPIO x[5] / Px[5] 输入使能
			[4]	RW	GPIO x[4] / Px[4] 输入使能
			[3]	RW	GPIO x[3] / Px[3] 输入使能
			[2]	RW	GPIO x[2] / Px[2] 输入使能
			[1]	RW	GPIO x[1] / Px[1] 输入使能
[0]	RW	GPIO x[0] / Px[0] 输入使能			

6.2.3 GPIOx_POE

表 6-4GPIOx 输出使能寄存器 GPIOx_POE

名称	复位值	偏移	位置	权限	说明
GPIOx_POE	0x0	0x04	[31:16]	NA	未使用
			[15]	RW	GPIO x[15] / Px[15] 输出使能
			[14]	RW	GPIO x[14] / Px[14] 输出使能
			[13]	RW	GPIO x[13] / Px[13] 输出使能
			[12]	RW	GPIO x[12] / Px[12] 输出使能
			[11]	RW	GPIO x[11] / Px[11] 输出使能
			[10]	RW	GPIO x[10] / Px[10] 输出使能
			[9]	RW	GPIO x[9] / Px[9] 输出使能
			[8]	RW	GPIO x[8] / Px[8] 输出使能
			[7]	RW	GPIO x[7] / Px[7] 输出使能
			[6]	RW	GPIO x[6] / Px[6] 输出使能
			[5]	RW	GPIO x[5] / Px[5] 输出使能
			[4]	RW	GPIO x[4] / Px[4] 输出使能
			[3]	RW	GPIO x[3] / Px[3] 输出使能
			[2]	RW	GPIO x[2] / Px[2] 输出使能



			[1]	RW	GPIO x[1] / Px[1] 输出使能
			[0]	RW	GPIO x[0] / Px[0] 输出使能

6.2.4 GPIOx_PDI

表 6-5GPIOx 输入数据寄存器 GPIOx_PDI

名称	复位值	偏移	位置	权限	说明
GPIOx_PDI	0x0	0x08	[31:16]	NA	未使用
			[15:0]	R	GPIO x 输入数据

6.2.5 GPIOx_PDO

表 6-6GPIOx 输出数据寄存器 GPIOx_PDO

名称	复位值	偏移	位置	权限	说明
GPIOx_PDO	0x00000000	0x0C	[31:16]	NA	未使用
			[15:0]	RW	GPIO x 输出数据

6.2.6 GPIOx_PUE

表 6-7GPIOx 上拉使能寄存器 GPIOx_PUE

名称	复位值	偏移	位置	权限	说明
GPIOx_PUE	0x0	0x10	[31:16]	NA	未使用
			[15]	RW	GPIO x[15] / Px[15] 上拉使能
			[14]	RW	GPIO x[14] / Px[14] 上拉使能
			[13]	RW	GPIO x[13] / Px[13] 上拉使能
			[12]	RW	GPIO x[12] / Px[12] 上拉使能
			[11]	RW	GPIO x[11] / Px[11] 上拉使能
			[10]	RW	GPIO x[10] / Px[10] 上拉使能
			[9]	RW	GPIO x[9] / Px[9] 上拉使能
			[8]	RW	GPIO x[8] / Px[8] 上拉使能
			[7]	RW	GPIO x[7] / Px[7] 上拉使能
			[6]	RW	GPIO x[6] / Px[6] 上拉使能
			[5]	RW	GPIO x[5] / Px[5] 上拉使能
			[4]	RW	GPIO x[4] / Px[4] 上拉使能
			[3]	RW	GPIO x[3] / Px[3] 上拉使能
			[2]	RW	GPIO x[2] / Px[2] 上拉使能
			[1]	RW	GPIO x[1] / Px[1] 上拉使能
[0]	RW	GPIO x[0] / Px[0] 上拉使能			



6.2.7 GPIOx_PDE

表 6-8GPIOx 下拉使能寄存器 GPIOx_PDE

名称	复位值	偏移	位置	权限	说明
GPIOx_PDE	0x0	0x14	[31:16]	NA	未使用
			[15]	RW	GPIO x[15] / Px[15] 下拉使能
			[14]	RW	GPIO x[14] / Px[14] 下拉使能
			[13]	RW	GPIO x[13] / Px[13] 下拉使能
			[12]	RW	GPIO x[12] / Px[12] 下拉使能
			[11]	RW	GPIO x[11] / Px[11] 下拉使能
			[10]	RW	GPIO x[10] / Px[10] 下拉使能
			[9]	RW	GPIO x[9] / Px[9] 下拉使能
			[8]	RW	GPIO x[8] / Px[8] 下拉使能
			[7]	RW	GPIO x[7] / Px[7] 下拉使能
			[6]	RW	GPIO x[6] / Px[6] 下拉使能
			[5]	RW	GPIO x[5] / Px[5] 下拉使能
			[4]	RW	GPIO x[4] / Px[4] 下拉使能
			[3]	RW	GPIO x[3] / Px[3] 下拉使能
			[2]	RW	GPIO x[2] / Px[2] 下拉使能
			[1]	RW	GPIO x[1] / Px[1] 下拉使能
[0]	RW	GPIO x[0] / Px[0] 下拉使能			

6.2.8 GPIOx_PODE

表 6-9GPIOx 开漏使能寄存器 GPIOx_PODE

名称	复位值	偏移	位置	权限	说明
GPIOx_PODE	0x0	0x18	[31:16]	NA	未使用
			[15]	RW	GPIO x[15] / Px[15] 开漏使能
			[14]	RW	GPIO x[14] / Px[14] 开漏使能
			[13]	RW	GPIO x[13] / Px[13] 开漏使能
			[12]	RW	GPIO x[12] / Px[12] 开漏使能
			[11]	RW	GPIO x[11] / Px[11] 开漏使能
			[10]	RW	GPIO x[10] / Px[10] 开漏使能
			[9]	RW	GPIO x[9] / Px[9] 开漏使能
			[8]	RW	GPIO x[8] / Px[8] 开漏使能
			[7]	RW	GPIO x[7] / Px[7] 开漏使能
			[6]	RW	GPIO x[6] / Px[6] 开漏使能
			[5]	RW	GPIO x[5] / Px[5] 开漏使能
			[4]	RW	GPIO x[4] / Px[4] 开漏使能
			[3]	RW	GPIO x[3] / Px[3] 开漏使能
			[2]	RW	GPIO x[2] / Px[2] 开漏使能



			[1]	RW	GPIO x[1] / Px[1] 开漏使能
			[0]	RW	GPIO x[0] / Px[0] 开漏使能

6.2.9 GPIOx_LCKR

表 6-10GPIOx 配置锁定寄存器 GPIOx_LCKR

名称	复位值	偏移	位置	权限	说明
GPIOx_LCKR	0x0	0x1C	[31:16]	NA	未使用
			[15]	RW	GPIO x[15] / Px[15] 配置锁定
			[14]	RW	GPIO x[14] / Px[14] 配置锁定
			[13]	RW	GPIO x[13] / Px[13] 配置锁定
			[12]	RW	GPIO x[12] / Px[12] 配置锁定
			[11]	RW	GPIO x[11] / Px[11] 配置锁定
			[10]	RW	GPIO x[10] / Px[10] 配置锁定
			[9]	RW	GPIO x[9] / Px[9] 配置锁定
			[8]	RW	GPIO x[8] / Px[8] 配置锁定
			[7]	RW	GPIO x[7] / Px[7] 配置锁定
			[6]	RW	GPIO x[6] / Px[6] 配置锁定
			[5]	RW	GPIO x[5] / Px[5] 配置锁定
			[4]	RW	GPIO x[4] / Px[4] 配置锁定
			[3]	RW	GPIO x[3] / Px[3] 配置锁定
			[2]	RW	GPIO x[2] / Px[2] 配置锁定
			[1]	RW	GPIO x[1] / Px[1] 配置锁定
[0]	RW	GPIO x[0] / Px[0] 配置锁定			

配置保护，高有效；有效时 GPIO 输入/输出/上下拉/开漏/功能选择不能被修改；需要注意，只有在 LCKR_PRT 写保护打开时才能改写 LCKR。

6.2.10 GPIOx_F3210

表 6-11GPIOx 功能选择寄存器 GPIOx_F3210

名称	复位值	偏移	位置	权限	说明
GPIOx_F3210	0x0	0x20	[31:16]	NA	未使用
			[15:12]	RW	GPIO x[3] / Px[3] 功能选择
			[11:8]	RW	GPIO x[2] / Px[2] 功能选择
			[7:4]	RW	GPIO x[1] / Px[1] 功能选择
			[3:0]	RW	GPIO x[0] / Px[0] 功能选择



6.2.11 GPIOx_F7654

表 6-12GPIOx 功能选择寄存器 GPIOx_F7654

名称	复位值	偏移	位置	权限	说明
GPIOx_F7654	0x0	0x24	[31:16]	NA	未使用
			[15:12]	RW	GPIO x[7] / Px[7] 功能选择
			[11:8]	RW	GPIO x[6] / Px[6] 功能选择
			[7:4]	RW	GPIO x[5] / Px[5] 功能选择
			[3:0]	RW	GPIO x[4] / Px[4] 功能选择

6.2.12 GPIOx_FBA98

表 6-13GPIOx 功能选择寄存器 GPIOx_FBA98

名称	复位值	偏移	位置	权限	说明
GPIOx_FBA98	0x0	0x28	[31:16]	NA	未使用
			[15:12]	RW	GPIO x[11] / Px[11] 功能选择
			[11:8]	RW	GPIO x[10] / Px[10] 功能选择
			[7:4]	RW	GPIO x[9] / Px[9] 功能选择
			[3:0]	RW	GPIO x[8] / Px[8] 功能选择

6.2.13 GPIOx_FFEDC

表 6-14GPIOx 功能选择寄存器 GPIOx_FFEDC

名称	复位值	偏移	位置	权限	说明
GPIOx_FFEDC	0x0	0x2C	[31:16]	NA	未使用
			[15:12]	RW	GPIO x[15] / Px[15] 功能选择
			[11:8]	RW	GPIO x[14] / Px[14] 功能选择
			[7:4]	RW	GPIO x[13] / Px[13] 功能选择
			[3:0]	RW	GPIO x[12] / Px[12] 功能选择

6.2.14 外部中断、唤醒、锁定保护

6.2.14.1 EXTI_CR0

表 6-15 外部中断配置寄存器 EXTI_CR0

名称	复位值	偏移	位置	权限	说明
EXTI_CR0	0x0	0x00	[31:16]	NA	未使用
			[15:14]	RW	GPIO 0[7]/ P0[7]外部中断触发类型选择
			[13:12]	RW	GPIO 0[6]/ P0[6]外部中断触发类型选择
			[11:10]	RW	GPIO 0[5]/ P0[5]外部中断触发类型选择



			[9:8]	RW	GPIO 0[4]/ P0[4]外部中断触发类型选择
			[7:6]	RW	GPIO 0[3]/ P0[3]外部中断触发类型选择
			[5:4]	RW	GPIO 0[2]/ P0[2]外部中断触发类型选择
			[3:2]	RW	GPIO 0[1]/ P0[1]外部中断触发类型选择
			[1:0]	RW	GPIO 0[0]/ P0[0]外部中断触发类型选择

以[0:1]为例:

0x00: P0[0] 不触发,

0x01: P0[0] 下降沿触发,

0x10: P0[0] 上升沿触发,

0x11: P0[0] 上升沿、下降沿都触发。

6.2.14.2 EXTI_CR1

表 6-16 外部中断配置寄存器 EXTI_CR1

名称	复位值	偏移	位置	权限	说明
EXTI_CR1	0x0	0x04	[31:16]	NA	未使用
			[15:14]	RW	GPIO 0[15]/P0[15]外部中断触发类型选择
			[13:12]	RW	GPIO 0[14]/P0[14]外部中断触发类型选择
			[11:10]	RW	GPIO 0[13]/P0[13]外部中断触发类型选择
			[9:8]	RW	GPIO 0[12]/P0[12]外部中断触发类型选择
			[7:6]	RW	GPIO 0[11]/P0[11]外部中断触发类型选择
			[5:4]	RW	GPIO 0[10]/P0[10]外部中断触发类型选择
			[3:2]	RW	GPIO 0[9]/P0[9]外部中断触发类型选择
			[1:0]	RW	GPIO 0[8]/P0[8]外部中断触发类型选择

以[0:1]为例:

0x00: P0[8] 不触发,

0x01: P0[8] 下降沿触发,

0x10: P0[8] 上升沿触发,

0x11: P0[8] 上升沿、下降沿都触发。

6.2.14.3 EXTI_IF

表 6-17 外部中断标志寄存器 EXTI_IF

名称	复位值	偏移	位置	权限	说明
EXTI_IF	0x0	0x08	[31:16]	NA	未使用



			[15]	RW	GPIO 0[15] / P0[15] 外部中断标志
			[14]	RW	GPIO 0[14] / P0[14] 外部中断标志
			[13]	RW	GPIO 0[13] / P0[13] 外部中断标志
			[12]	RW	GPIO 0[12] / P0[12] 外部中断标志
			[11]	RW	GPIO 0[11] / P0[11] 外部中断标志
			[10]	RW	GPIO 0[10] / P0[10] 外部中断标志
			[9]	RW	GPIO 0[9] / P0[9] 外部中断标志
			[8]	RW	GPIO 0[8] / P0[8] 外部中断标志
			[7]	RW	GPIO 0[7] / P0[7] 外部中断标志
			[6]	RW	GPIO 0[6] / P0[6] 外部中断标志
			[5]	RW	GPIO 0[5] / P0[5] 外部中断标志
			[4]	RW	GPIO 0[4] / P0[4] 外部中断标志
			[3]	RW	GPIO 0[3] / P0[3] 外部中断标志
			[2]	RW	GPIO 0[2] / P0[2] 外部中断标志
			[1]	RW	GPIO 0[1] / P0[1] 外部中断标志
			[0]	RW	GPIO 0[0] / P0[0] 外部中断标志

中断标志高有效，写 1 清零。

6.2.14.4 LCKR_PRT

表 6-18 锁定保护寄存器 LCKR_PRT

名称	复位值	偏移	位置	权限	说明
LCKR_PRT	0x0	0x0C	[31:16]	NA	未使用
			[15:0]	W	配置锁定写保护；写入 0x5AC4 关闭写保护，然后才能修改 GPIO_LCKR；写入任意其他数据开启写保护；1bit 状态指示当前写保护与否，高有效

6.2.14.5 WAKE_POL

表 6-19 外部唤醒源极性配置寄存器 WAKE_POL

名称	复位值	偏移	位置	权限	说明
WAKE_POL	0x0	0x10	[31:16]	NA	未使用
			[15:4]	RW	保留
			[3]	RW	GPIO 1[1] / P1[1]外部唤醒触发电平选择
			[2]	RW	GPIO 1[0] / P1[0]外部唤醒触发电平选择
			[1]	RW	GPIO 0[1] / P0[1]外部唤醒触发电平选择
			[0]	RW	GPIO 0[0] / P0[0]外部唤醒触发电平选择



1: 高电平, 0: 低电平。

6.2.14.6 WAKE_EN

表 6-20 外部唤醒源使能寄存器 WAKE_EN

名称	复位值	偏移	位置	权限	说明
WAKE_EN	0x0	0x14	[31:16]	NA	未使用
			[15:4]	RW	保留
			[3]	RW	GPIO 1[1] / P1[1] 外部唤醒使能
			[2]	RW	GPIO 1[0] / P1[0] 外部唤醒使能
			[1]	RW	GPIO 0[1] / P0[1] 外部唤醒使能
			[0]	RW	GPIO 0[0] / P0[0] 外部唤醒使能

1: 使能, 0: 禁用。

6.3 应用指南

6.3.1 配置锁定

芯片提供对 GPIO 配置的保护功能。当 LCKR_PRT 写保护使能时, 3 组 GPIO 的 GPIO_LCKR 不能修改, GPIO_PIE / GPIO_POE / GPIO_PUE / GPIO_PDE / GPIO_ODE / GPIO_F3210 / GPIO_F7654 / GPIO_FBA98 / GPIO_FFEDC 不能修改;

若需要修改 GPIO 配置, 应先解除 LCKR_PRT 写保护, 然后将对应 GPIO 的 GPIO_LCKR 写 0, 解除配置锁定, 然后修改 GPIO 配置。

示例如下:

```

GPIO0_PIE   = 0x1234;

GPIO1_PIE   = 0x7777;

GPIO2_PIE   = 0xF000;

//-----

// lock specific gpio

GPIO0_LCKR  = 0x0100;    //lock gpio0 here

GPIO1_LCKR  = 0xFFFF;    //lock gpio1 here

GPIO2_LCKR  = 0x8000;    //lock gpio2 here

//-----

// modify to test if gpio config is locked

GPIO0_PIE   = 0x3333;

```



```

GPIO1_PIE    = 0x0000;

GPIO2_PIE    = 0x0000;

//-----
// read gpio config to flag PASS or FAIL
if(GPIO0_PIE  != 0x3233)FAIL;
if(GPIO1_PIE  != 0x7777)FAIL;
if(GPIO2_PIE  != 0x8000)FAIL;

LCKR_PRT     = 0x0000; // write any value other than 0x5AC4 to enable lock protect
GPIO0_LCKR   = 0x0000;
GPIO1_LCKR   = 0x0000;
GPIO2_LCKR   = 0x0000;

if(GPIO0_LCKR != 0x0100)FAIL;
if(GPIO1_LCKR != 0xFFFF)FAIL;
if(GPIO2_LCKR != 0x8000)FAIL;

LCKR_PRT     = 0x5AC4; // disable protect
GPIO0_LCKR   = 0x0000;
GPIO1_LCKR   = 0x0000;
GPIO2_LCKR   = 0x0000;

if(GPIO0_LCKR != 0x0000)FAIL;
if(GPIO1_LCKR != 0x0000)FAIL;
if(GPIO2_LCKR != 0x0000)FAIL;

i=1000;

while(i--);

PASS;

```



6.3.2 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 模数转换器(ADC)

7.1 概述

芯片集成 2 个 12BIT SARADC，每个 ADC 有 12 路输入通道。芯片上电的默认状态下，ADC 模块是关闭的。通过将 ADCxPDN 设置为 1 来开启 ADC，ADC 开启前，需要先开启 BGP、4M RC 时钟和 PLL 模块，并通过配置 ADCCLKSEL<1:0>选择 ADC 工作频率。

ADCxPDN 的说明见[模拟寄存器 SYS_AFE_REG5](#)

ADCCLKSEL<1:0>的说明见[模拟寄存器 SYS_AFE_REG7](#)

ADC 完成一次转换需要 16 个 ADC 时钟周期，其中 13 个为转换周期，3 个为采样周期。在 ADC 时钟设为 48M 时，转换速率是 3MHz。

ADC 在降频应用时，可通过模拟寄存器 CURRIT[1:0]降低 ADC 的功耗水平。

CURRIT[1:0]的说明见[模拟寄存器 SYS_AFE_REG2](#)

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存储值 ADCx_DAT 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 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]为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。

7.1.1 功能框图

如下图所示，芯片集成两路 SAR ADC，每路 ADC 可以选择配置 12 个通道的输入。

每个 ADC 接口包括 12 个数据寄存器（ADC 12 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 7.2.3 信号来源寄存器章节）。以 ADCx_CHN0 (x=0,1, 分别对应 ADC0 和 ADC1) 为例，位[3:0]选择第 0 次采样的模拟通道号，通道号 CH0~CH11 任选，若 ADCx_CHN0[3:0]=0，则第 0 次采样会采样 CH0；若 ADCx_CHN0[15:12]=3，则第 3 次采样会采样 CH3，以此类推。



分段采样次数寄存器 $ADCx_CHNT$ 控制每轮采样的次数，0~15 对应 1~16 次。

控制逻辑根据配置寄存器 $ADCx_CFG$ 选择来自 MCPWM 定时器的触发信号启动一轮采样或者软件触发启动。MCPWM 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。

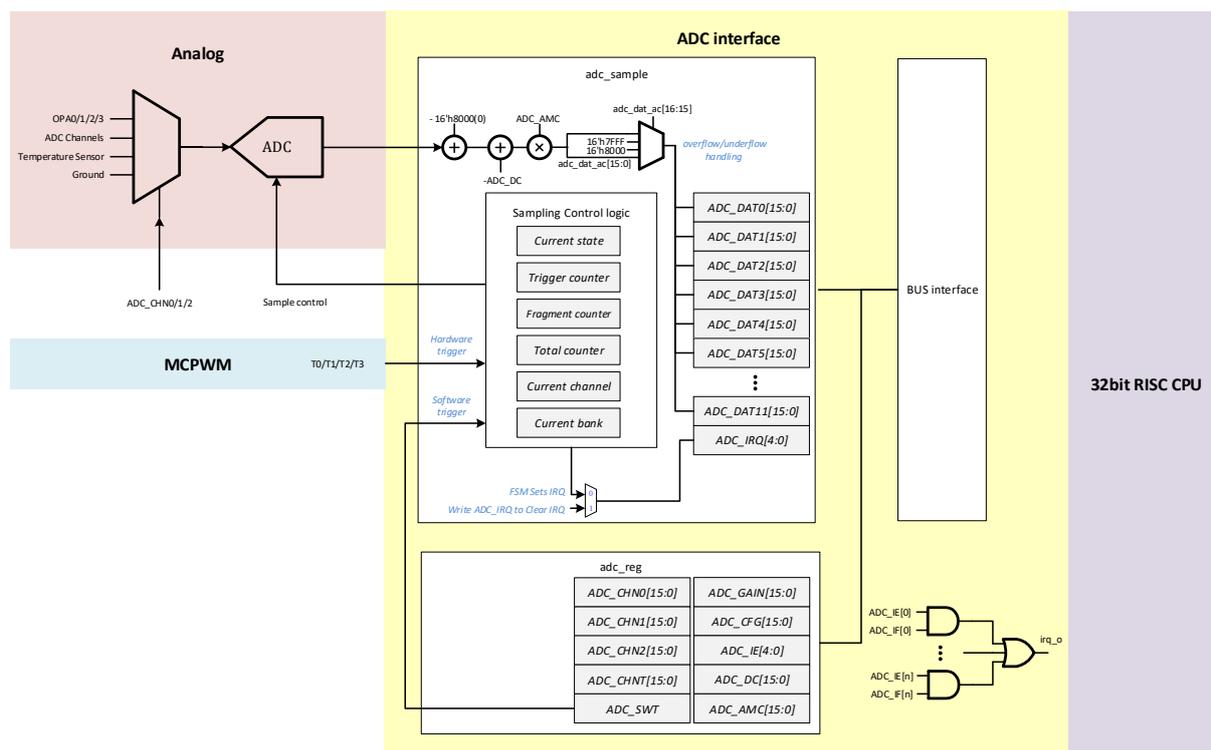


图 7-1 ADC 采集模块功能框图

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。同时每个通道的 ADC 增益也可以通过寄存器配置（两档增益可选）。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。

7.1.2 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 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 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

7.1.3 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-1.2V 对应 12h'1000_0000_0000，输入信号+1.2V 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。

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 7-1 ADC 输出数字量数制转换

ADC 一倍增益输入模拟量数值/V	ADC1/3 倍增益输入模拟量数值/V	转为有符号数后的数值
1.2	3.6	12'h0111_1111_1111
0	0	12'h0000_0000_0000
-1.2	-3.6	12'h1000_0000_0000

7.1.4 ADC 量程

ADC 有两种增益模式：高增益（1 倍）和低增益（1/3 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±1.2V 的输入信号幅度，1/3 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~OPA3，配置方式见 7.2.3 信号来源寄存器），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+/-3.3V 的水平，同时将 ADC 配置为 1/3 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 5mR，则运放的最大输入信号幅值为+/-707mV。此时应该选择运放的放大倍数为 4.5 倍（放大倍数选择方式见 3.6 运算放大器），则放大后的信号约为+/-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+/-1.2V，则应将 ADC 的增益配置为 1 倍。

在 ADC 采样通道配置为 GPIO 复用口输入的的信号时（即 ADC0_CH4~ADC0_CH9 及 ADC1_CH4~ADC1_CH10，配置方式见 7.2.3 信号来源寄存器），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~AVDD+0.3V 之间。

高低增益选择由 ADCx_GAIN 增益寄存器进行控制。



7.1.5 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。

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

ADCx_DC 存储的是 ADC 的直流偏置，通常在校正阶段通过测量通道 11 的 VSS 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 ADCx_DC 寄存器中。

需要注意的是，ADC 有高增益和低增益两档配置，两种配置对应两套校正参数，每套校正数据分别包含一个 DC offset(以下记为 DC_{offset})和一个增益校正系数 $AMP_{correction}$ 。高增益对应的校正系数为 $ADCx_DC1/ADCx_AMC1$ ，低增益对应的校正系数为 $ADCx_DC0/ADCx_AMC0$ 。

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

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

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。ADC 接口硬件电路会根据每个通道的增益配置(ADCx_GAIN)来自动选择 $AMP_{correction}$ 与 DC_{offset} 。

7.2 寄存器

7.2.1 地址分配

ADC0 在芯片中的基地址是 0x4000_3300;

表 7-2 ADC0 寄存器列表

名称	偏移地址	说明
ADC0_DAT0	0x00	ADC0 第 0 次采样数据
ADC0_DAT1	0x04	ADC0 第 1 次采样数据
ADC0_DAT2	0x08	ADC0 第 2 次采样数据
ADC0_DAT3	0x0C	ADC0 第 3 次采样数据
ADC0_DAT4	0x10	ADC0 第 4 次采样数据
ADC0_DAT5	0x14	ADC0 第 5 次采样数据
ADC0_DAT6	0x18	ADC0 第 6 次采样数据
ADC0_DAT7	0x1C	ADC0 第 7 次采样数据
ADC0_DAT8	0x20	ADC0 第 8 次采样数据
ADC0_DAT9	0x24	ADC0 第 9 次采样数据
ADC0_DAT10	0x28	ADC0 第 10 次采样数据
ADC0_DAT11	0x2C	ADC0 第 11 次采样数据
ADC0_CHN0	0x40	ADC0 第 0~3 次采样信号选择
ADC0_CHN1	0x44	ADC0 第 4~7 次采样信号选择
ADC0_CHN2	0x48	ADC0 第 8~11 次采样信号选择
ADC0_CHNT	0x50	ADC0 各段采样通道数
ADC0_IE	0x54	ADC0 中断使能



ADC0_CFG	0x60	ADC0 配置
ADC0_GAIN	0x64	ADC0 增益控制
ADC0_IF	0x68	ADC0 中断标志
ADC0_SWT	0x6C	ADC0 软件触发
ADC0_DC0	0x70	ADC0 非 1 倍增益 DC offset
ADC0_DC1	0x74	ADC0 1 倍增益 DC offset
ADC0_AMC0	0x78	ADC0 非 1 倍增益增益校正
ADC0_AMC1	0x7C	ADC0 1 倍增益增益校正

ADC1 在芯片中的基地址是 0x4000_3400;

表 7-3 ADC1 寄存器列表

名称	偏移地址	说明
ADC1_DAT0	0x00	ADC1 第 0 次采样数据
ADC1_DAT1	0x04	ADC1 第 1 次采样数据
ADC1_DAT2	0x08	ADC1 第 2 次采样数据
ADC1_DAT3	0x0C	ADC1 第 3 次采样数据
ADC1_DAT4	0x10	ADC1 第 4 次采样数据
ADC1_DAT5	0x14	ADC1 第 5 次采样数据
ADC1_DAT6	0x18	ADC1 第 6 次采样数据
ADC1_DAT7	0x1C	ADC1 第 7 次采样数据
ADC1_DAT8	0x20	ADC1 第 8 次采样数据
ADC1_DAT9	0x24	ADC1 第 9 次采样数据
ADC1_DAT10	0x28	ADC1 第 10 次采样数据
ADC1_DAT11	0x2C	ADC1 第 11 次采样数据
ADC1_CHN0	0x40	ADC1 第 0~3 次采样信号选择
ADC1_CHN1	0x44	ADC1 第 4~7 次采样信号选择
ADC1_CHN2	0x48	ADC1 第 8~11 次采样信号选择
ADC1_CHNT	0x50	ADC1 各段采样通道数
ADC1_IE	0x54	ADC1 中断使能
ADC1_CFG	0x60	ADC1 触发控制
ADC1_GAIN	0x64	ADC1 增益控制
ADC1_IF	0x68	ADC1 中断标志
ADC1_SWT	0x6C	ADC1 软件触发
ADC1_DC0	0x70	ADC1 低增益 DC offset
ADC1_DC1	0x74	ADC1 高增益 DC offset
ADC1_AMC0	0x78	ADC1 低增益增益校正
ADC1_AMC1	0x7C	ADC1 高增益增益校正

由于 ADC0、ADC1 实现及地址偏移完全相同，以下寄存器说明不作区分。



7.2.2 采样数据寄存器

7.2.2.1 ADCx_DAT0

表 7-4 采样数据寄存器 ADCx_DAT0

名称	复位值	偏移	位置	权限	说明
ADCx_DAT0	0x0	0x00	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 0 次采样数据

7.2.2.2 ADCx_DAT1

表 7-5 采样数据寄存器 ADCx_DAT1

名称	复位值	偏移	位置	权限	说明
ADCx_DAT1	0x0	0x04	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 1 次采样数据

7.2.2.3 ADCx_DAT2

表 7-6 采样数据寄存器 ADCx_DAT2

名称	复位值	偏移	位置	权限	说明
ADCx_DAT2	0x0	0x08	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 2 次采样数据

7.2.2.4 ADCx_DAT3

表 7-7 采样数据寄存器 ADCx_DAT3

名称	复位值	偏移	位置	权限	说明
ADCx_DAT3	0x0	0x0C	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 3 次采样数据

7.2.2.5 ADCx_DAT4

表 7-8 采样数据寄存器 ADCx_DAT4

名称	复位值	偏移	位置	权限	说明
ADCx_DAT4	0x0	0x10	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 4 次采样数据



7.2.2.6 ADCx_DAT5

表 7-9 采样数据寄存器 ADCx_DAT5

名称	复位值	偏移	位置	权限	说明
ADCx_DAT5	0x0	0x14	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 5 次采样数据

7.2.2.7 ADCx_DAT6

表 7-10 采样数据寄存器 ADCx_DAT6

名称	复位值	偏移	位置	权限	说明
ADCx_DAT6	0x0	0x18	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 6 次采样数据

7.2.2.8 ADCx_DAT7

表 7-11 采样数据寄存器 ADCx_DAT7

名称	复位值	偏移	位置	权限	说明
ADCx_DAT7	0x0	0x1C	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 7 次采样数据

7.2.2.9 ADCx_DAT8

表 7-12 采样数据寄存器 ADCx_DAT8

名称	复位值	偏移	位置	权限	说明
ADCx_DAT8	0x0	0x20	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 8 次采样数据

7.2.2.10 ADCx_DAT9

表 7-13 采样数据寄存器 ADCx_DAT9

名称	复位值	偏移	位置	权限	说明
ADCx_DAT9	0x0	0x24	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 9 次采样数据



7.2.2.11 ADCx_DAT10

表 7-14 采样数据寄存器 ADCx_DAT10

名称	复位值	偏移	位置	权限	说明
ADCx_DAT10	0x0	0x28	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 10 次采样数据

7.2.2.12 ADCx_DAT11

表 7-15 采样数据寄存器 ADCx_DAT11

名称	复位值	偏移	位置	权限	说明
ADCx_DAT11	0x0	0x2C	[31:16]	NA	未使用
			[15:0]	RW	ADCx 第 11 次采样数据

ADC 可以在一轮采样中进行若干次采样，若不足 12 次采样（第 0 次~第 11 次）则后续的采样数据寄存器保持上一次的采样值不变。

7.2.3 信号来源寄存器

7.2.3.1 ADCx_CHN0

表 7-16 信号来源寄存器 ADCx_CHN0

名称	复位值	偏移	位置	权限	说明
ADCx_CHN0	0x0	0x40	[31:16]	NA	未使用
			[15:12]	RW	ADCx 第 3 次采样信号选择
			[11:8]	RW	ADCx 第 2 次采样信号选择
			[7:4]	RW	ADCx 第 1 次采样信号选择
			[3:0]	RW	ADCx 第 0 次采样信号选择

7.2.3.2 ADCx_CHN1

表 7-17 信号来源寄存器 ADCx_CHN1

名称	复位值	偏移	位置	权限	说明
ADCx_CHN1	0x0	0x44	[31:16]	NA	未使用
			[15:12]	RW	ADCx 第 7 次采样信号选择
			[11:8]	RW	ADCx 第 6 次采样信号选择
			[7:4]	RW	ADCx 第 5 次采样信号选择
			[3:0]	RW	ADCx 第 4 次采样信号选择



7.2.3.3 ADCx_CHN2

表 7-18 信号来源寄存器 ADCx_CHN2

名称	复位值	偏移	位置	权限	说明
ADCx_CHN2	0x0	0x48	[31:16]	NA	未使用
			[15:12]	RW	ADCx 第 11 次采样信号选择
			[11:8]	RW	ADCx 第 10 次采样信号选择
			[7:4]	RW	ADCx 第 9 次采样信号选择
			[3:0]	RW	ADCx 第 8 次采样信号选择

上述每 4 个 BIT 一组的信号，用来选择具体的模拟输入通道。这 4 个 BIT 所对应的信号通道说明如下（OPA1~OPA3 输出是 OPA 输入信号经 OPA 放大后的内部信号，ADC0_CH4~ADC0_CH9 及 ADC1_CH4~ADC1_CH10 在芯片管脚上的位置，参见数据手册引脚说明部分）：

表 7-19 ADC 采样信号通道选择

ADC0 输入信号正端选择	4'b0000:OPA0 输出; 4'b0010:OPA2 输出; 4'b0100:ADC01_CH4; 4'b0110:ADC01_CH6; 4'b1000:ADC0_CH8; 4'b1010:温度传感器;	4'b0001:OPA1 输出; 4'b0011:OPA3 输出; 4'b0101:ADC01_CH5; 4'b0111: ADC0_CH7; 4'b1001:ADC0_CH9; 4'b1011~1111:内部地
ADC1 输入信号正端选择	4'b0000:OPA2 输出; 4'b0010:OPA0 输出; 4'b0100:ADC01_CH4; 4'b0110: ADC01_CH6; 4'b1000:ADC1_CH8; 4'b1010: ADC1_CH10;	4'b0001:OPA3 输出; 4'b0011:OPA1 输出; 4'b0101:ADC01_CH5; 4'b0111: ADC1_CH7; 4'b1001:ADC1_CH9; 4'b1011~1111:内部地

ADC 输入信号的负端统一接地。

7.2.4 分段采样次数寄存器

7.2.4.1 ADCx_CHNT

表 7-20 分段采样次数寄存器 ADCx_CHNT

名称	复位值	偏移	位置	权限	说明
ADCx_CHNT	0x0	0x50	[31:16]	NA	未使用
			[15:12]	RW	四段采样模式下第四段采样次数
			[11:8]	RW	四段采样模式下第三段采样次数
			[7:4]	RW	两段或四段采样模式下第二段采样次数
			[3:0]	RW	单段、两段或四段采样模式下第一段采样次数



0 表示 1 次采样，1 表示 2 次采样，以此类推，11 表示 12 次采样。

假配置 ADC 进行四段采样工作模式，第一段采样 3 次，第二段采样 2 次，第三段采样 3 次，第四段采样 1 次；则配置寄存器值为：

$ADCx_CHNT = 0x0212$ 。

假配置 ADC 进行两段采样工作模式，第一段采样 3 次，第二段采样 2 次；则配置寄存器值为：

$ADCx_CHNT = 0x0012$ ；

$ADCx_CHNT[15:8]$ 的值不起作用。

完成采样后 $ADCx_DAT0$ 、 $ADCx_DAT1$ 、 $ADCx_DAT2$ 、 $ADCx_DAT3$ 、 $ADCx_DAT4$ 会更新； $ADCx_DAT5$ 、 $ADCx_DAT6$ 、 $ADCx_DAT7$ 、 $ADCx_DAT8$ 、 $ADCx_DAT9$ 、 $ADCx_DAT10$ 、 $ADCx_DAT11$ 维持原值不变。

四段采样次数的数目之和最大为 12。若采样次数数目之和 $y < 12$ ，则 $ADCx_DAT0 \sim ADCx_DAT(y-1)$ 的数据寄存器数据会相应发生更新，而之后的几个 $ADCx_DAT$ 寄存器保持原值不变。

7.2.5 中断使能寄存器

7.2.5.1 $ADCx_IE$

表 7-21 中断使能寄存器 $ADCx_IE$

名称	复位值	偏移	位置	权限	说明
$ADCx_IE$	0x0	0x54	[31:5]	NA	未使用
			[4]	RW	软件触发发生在非空闲状态中断使能
			[3]	RW	第四段采样完成中断使能
			[2]	RW	第三段采样完成中断使能
			[1]	RW	第二段采样完成中断使能
			[0]	RW	第一段采样完成中断使能

7.2.6 配置寄存器

7.2.6.1 $ADCx_CFG$

表 7-22 配置寄存器 $ADCx_CFG$

名称	复位值	偏移	位置	权限	说明
$ADCx_CFG$	0x0	0x60	[31:12]	NA	未使用
			[11]	RW	状态机复位，软件写入后状态机回到 idle 状态，完成后自动清零
			[10]	RW	$ADCx_DAT$ 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]	RW	触发模式 0:单段触发; 1:两段触发; 2:保留; 3:四段触发
			[7:4]	RW	单段触发模式下触发一次采样所需的事件数 (0 表示 1 次事件即触发, 15 表示 16 次事件才触发)
			[3:0]	RW	MCPWM 触发 ADC 采样使能 4'b0000: 全部禁用, 状态机始终处于空闲状态 4'bxxx1: TADC[0]被使能 4'bxx1x: TADC[1]被使能 4'bx1xx: TADC[2]被使能 4'b1xxx: TADC[3]被使能 4'b1111: TADC[3]/TADC[2]/TADC[1]/TADC[0]都被使能

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。

7.2.7 增益选择寄存器

7.2.7.1 ADCx_GAIN

表 7-23 增益选择寄存器 ADCx_GAIN

名称	复位值	偏移	位置	权限	说明
ADCx_GAIN	0x0	0x64	[31:12]	NA	未使用
			[11]	RW	ADCx_DAT11 增益选择
			[10]	RW	ADCx_DAT10 增益选择
			[9]	RW	ADCx_DAT9 增益选择
			[8]	RW	ADCx_DAT8 增益选择
			[7]	RW	ADCx_DAT7 增益选择
			[6]	RW	ADCx_DAT6 增益选择
			[5]	RW	ADCx_DAT5 增益选择
			[4]	RW	ADCx_DAT4 增益选择
			[3]	RW	ADCx_DAT3 增益选择
			[2]	RW	ADCx_DAT2 增益选择
			[1]	RW	ADCx_DAT1 增益选择
			[0]	RW	ADCx_DAT0 增益选择

0:低增益, 1:高增益。



当 ADCx_GAIN 为 1 时，ADC 增益=1；当 ADCx_GAIN 为 0 时，ADC 增益=1/3；。具体请参考下表，模拟寄存器位置请参考[错误!未找到引用源](#)。模拟寄存器表或[错误!未找到引用源](#)。其他 ADC 相关系统寄存器说明。

7.2.8 中断标志寄存器

7.2.8.1 ADCx_IF

表 7-24 中断标志寄存器 ADCx_IF

名称	复位值	偏移	位置	权限	说明
ADCx_IF	0x0	0x68	[31:5]	NA	未使用
			[4]	RW	1: 软件触发发生在非空闲状态, 0: 未发生中断
			[3]	RW	1: 第四段采样完成, 0: 未发生中断
			[2]	RW	1: 第三段采样完成, 0: 未发生中断
			[1]	RW	1: 第二段采样完成, 0: 未发生中断
			[0]	RW	1: 第一段采样完成, 0: 未发生中断

中断标志位通过写入 1 清空。

7.2.9 软件触发寄存器

7.2.9.1 ADCx_SWT

表 7-25 软件触发寄存器 ADCx_SWT

名称	复位值	偏移	位置	权限	说明
ADCx_SWT	0x0	0x6C	[31:16]	NA	未使用
			[15:0]	W	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。

7.2.10 直流偏置寄存器

ADC 的通道 11 为内部模拟地，通过 ADCx_DAT11 可以得到系统的 DC 偏置。

通常系统初始化后会进行一次内部模拟地的测量，并使用软件将测量值 ADCx_DAT11 存入 DC offset 寄存器。硬件电路会在后续采样转换其他通道信号（通道 0~通道 10）时，从转换后的数字量中减去 DC offset 然后再存入相应的采样数据寄存器（ADCx_DAT0~ADCx_DAT10）。由于 ADC 有两种增益设置，高增益和低增益时各需要测量一次 DC offset，分别存入 DC0 和 DC1，后续每个通道采样时会根据增益的设置自动选择相应的 DC offset 进行偏置减除。



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，防止因减除 DC offset 而发生上溢或下溢。

7.2.10.1 ADCx_DC0

表 7-26 直流偏置寄存器 ADCx_DC0

名称	复位值	偏移	位置	权限	说明
ADCx_DC0	0x0	0x70	[31:16]	NA	未使用
			[15:0]	RW	低增益 ADC DC offset

7.2.10.2 ADCx_DC1

表 7-27 直流偏置寄存器 ADCx_DC1

名称	复位值	偏移	位置	权限	说明
ADCx_DC1	0x0	0x74	[31:16]	NA	未使用
			[15:0]	RW	高增益 ADC DC offset

7.2.11 增益校正寄存器

7.2.11.1 ADCx_AMC0

表 7-28 增益校正寄存器 ADCx_AMC0

名称	复位值	偏移	位置	权限	说明
ADCx_AMC0	0x0	0x78	[31:10]	NA	未使用
			[9:0]	RW	低增益 ADC 增益校正寄存器

7.2.11.2 ADCx_AMC1

表 7-29 增益校正寄存器 ADCx_AMC1

名称	复位值	偏移	位置	权限	说明
ADCx_AMC1	0x0	0x7C	[15:10]	NA	未使用
			[9:0]	RW	高增益 ADC 增益校正寄存器

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

ADC 有高增益和低增益两档配置，两种配置对应两套校正参数，每套校正数据分别包含一个 DC offset(以下记为 DC_{offset})和一个增益校正值 $AMP_{correction}$ 。

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



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

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。

7.3 应用指南

7.3.1 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 的比较事件 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 ADCx_SWT 写如命令字的方式 16'h5AA5 软件触发 ADC 采样。

第一轮采样

判断是否为一段采样。

是：采样次数达到预设值 ADCx_CHNT[3:0]+1，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 ADCx_CHNT[3:0]+1，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

第二轮采样

第二轮采样次数到达预设值 ADCx_CHNT[7:4]+1，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

第三轮采样

第三轮采样次数到达预设值 ADCx_CHNT[11:8]+1，进入空闲状态 3。

第四段触发

第四轮采样

第四段采样通道数到达预设值 ADCx_CHNT[15:12]+1，回到空闲状态 0。



各种硬件触发模式的触发条件汇总如表 7-30 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 ADCx_CFG 寄存器设置，一次 MCPWM 事件即触发采样，还是多次 MCPWM 事件才触发采样；而两段、四段采样模式仅支持一次相应的 MCPWM 事件即触发采样。

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 7-30 ADC 采样触发模式

	单段触发	两段触发	四段触发
Timer 触发	None (Timer trigger 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 ADCx_SWT 写入 16'h5aa5 第二段向 ADCx_SWT 写入 16'h5aa5	第一段向 ADCx_SWT 写入 16'h5aa5 第二段向 ADCx_SWT 写入 16'h5aa5 第三段向 ADCx_SWT 写入 16'h5aa5 第四段向 ADCx_SWT 写入 16'h5aa5

7.3.1.1 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 ADCx_CHNT 进行配置，寄存器数值为 0~15 时，对应的采样次数为 1~16。

触发事件可以是来自外部的 MCPWM 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 ADCx_CHN0/1/2 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 7-2 ADC 单段采样状态转移图所示。



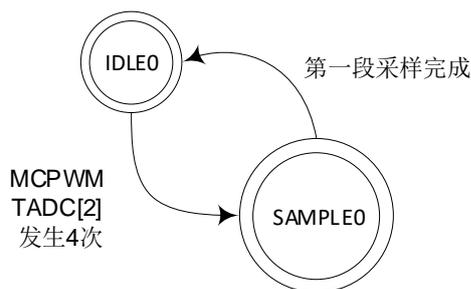


图 7-2 ADC 单段采样状态转移图

7.3.1.2 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。

触发事件可以是来自外部的定时器信号 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 ADCx_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 ADCx_CHNT[7:4]次采样。采样次数均通过分段采样次数寄存器 ADCx_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，即 ADCx_CFG [1:0]=2'b00。然后通过向偏移地址 0x30 的寄存器写入 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 7-3 ADC 两段采样状态转移图所示。

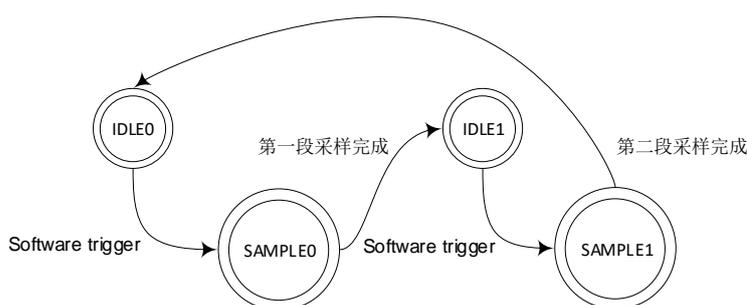


图 7-3 ADC 两段采样状态转移图

7.3.1.3 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 MCPWMTADC[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 ADCx_CHNT[15:12]、ADCx_CHNT[11:8]、ADCx_CHNT[7:4]、ADCx_CHNT[3:0]。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 7-4 ADC 四段采样状态转移图所示。



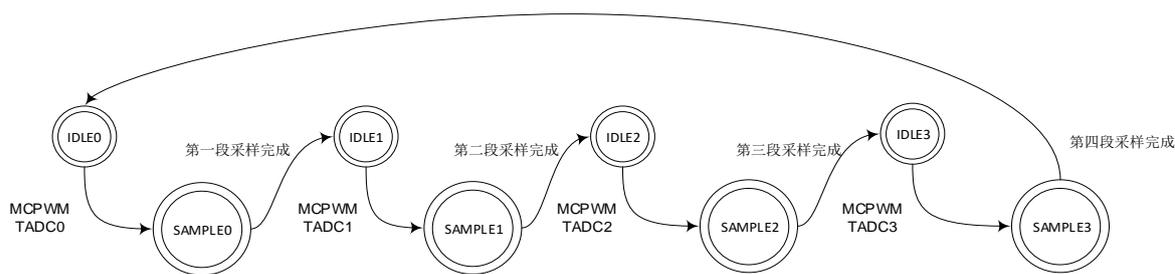


图 7-4 ADC 四段采样状态转移图

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 计数器值，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。

7.3.2 中断

7.3.2.1 单段触发采样完成中断

采样完成产生一个中断。

7.3.2.2 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。

7.3.2.3 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。

7.3.3 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;
```



8 通用定时器

8.1 概述

8.1.1 功能框图

如图 8-1 模块顶层功能框图所示，通用定时器 **UTIMER** 主要包括下面功能模块。

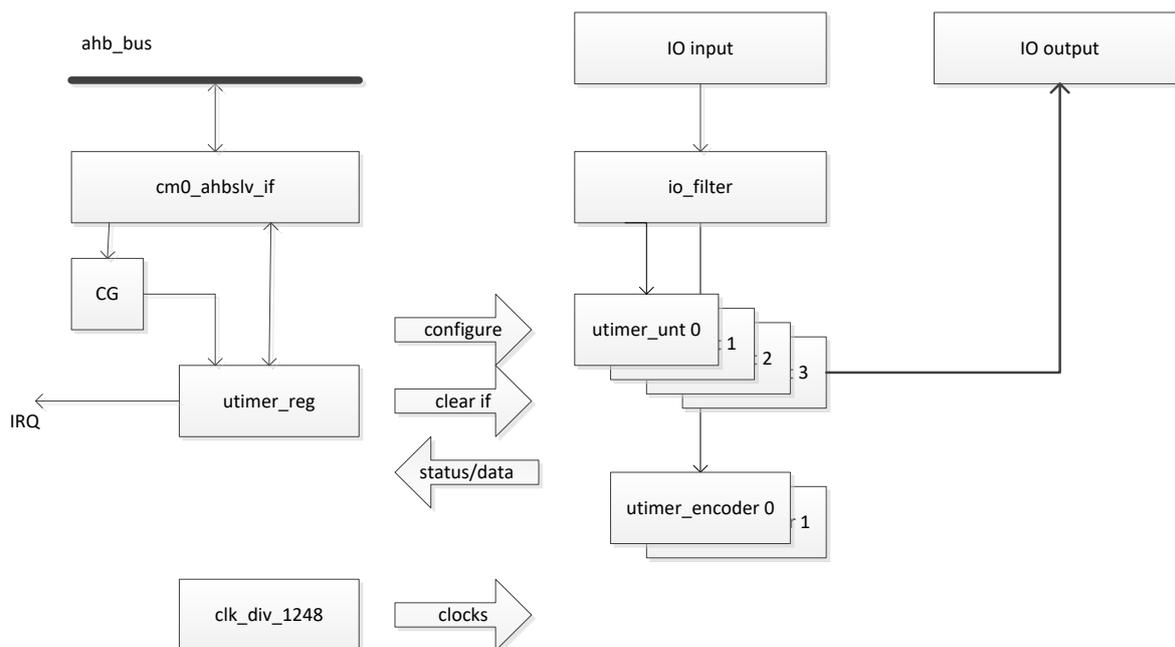


图 8-1 模块顶层功能框图

8.1.1.1 总线接口模块

总线接口模块包括：

cm0_ahbslv_if，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。

8.1.1.2 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。

8.1.1.3 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。



8.1.1.4 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，每个定时器包含两个通道。

8.1.1.5 编码器模块

编码器模块用于对芯片外部送入的编码器编码信号进行计数。定时器模块中集成了 2 个编码器模块。

8.1.1.6 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。

8.1.2 功能特点

定时器模块有以下特点：

4 个独立工作，可工作在不同频率下的 16bit 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）

2 个独立工作计数器

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 宽度一下毛刺

8.2 实现说明

8.2.1 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 均工作在系统主频，但采用分频计数器来降低计数器计数频率的方案。

8.2.2 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。

8.2.3 滤波

定时器模块共有 8 个输入，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。

如下图，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。



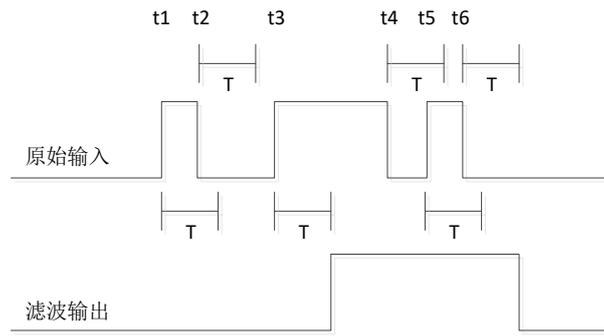


图 8-2 滤波示意图

8.2.4 模式

8.2.4.1 计数器

Timer 中的计数器采用 up 模式技术。

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。

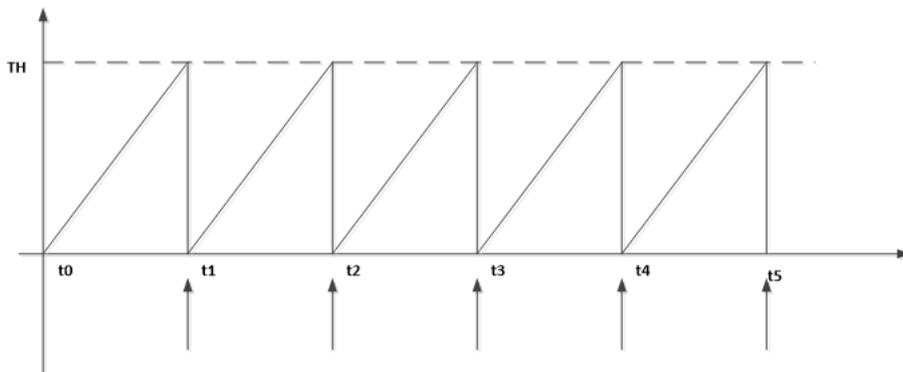
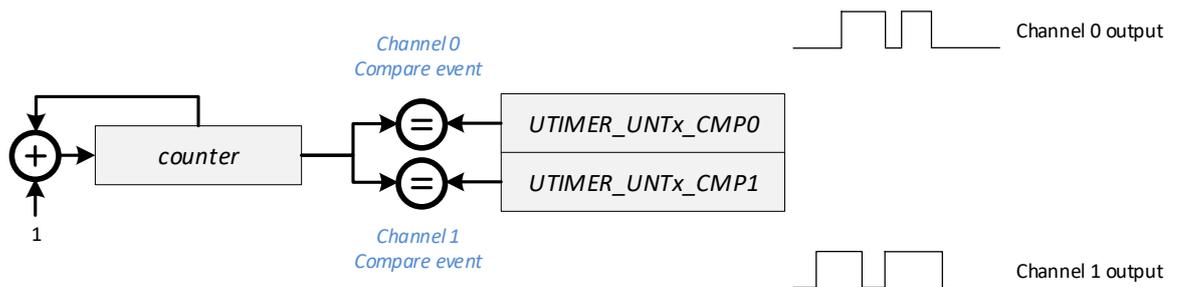


图 8-3 通用计数器

8.2.4.2 比较模式

比较模式下，计数器计数到 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，输出一个电平（可配置极性），在比较事件发生时，电平翻转。定时器回零时，任然会产生回零中断。



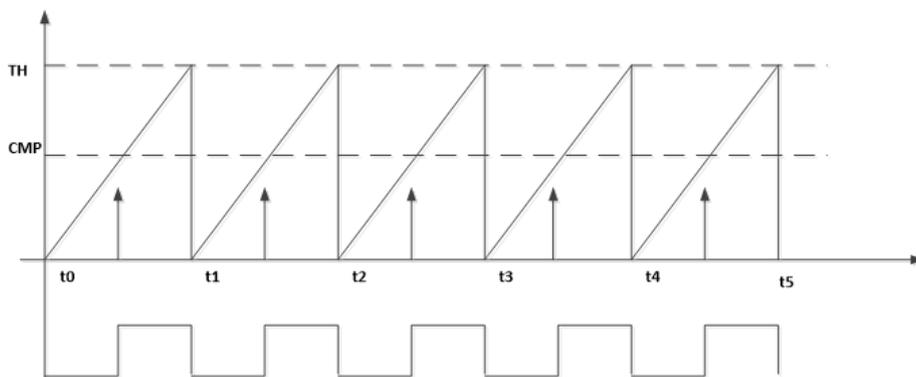


图 8-4 比较模式

8.2.4.3 捕获模式

捕获模式下，可以捕获输入信号的上升/下降或者双沿，发生捕获事件时，定时器计数值存入 CMP 寄存器，并产生捕获中断。定时器回零时，仍然会产生回零中断。

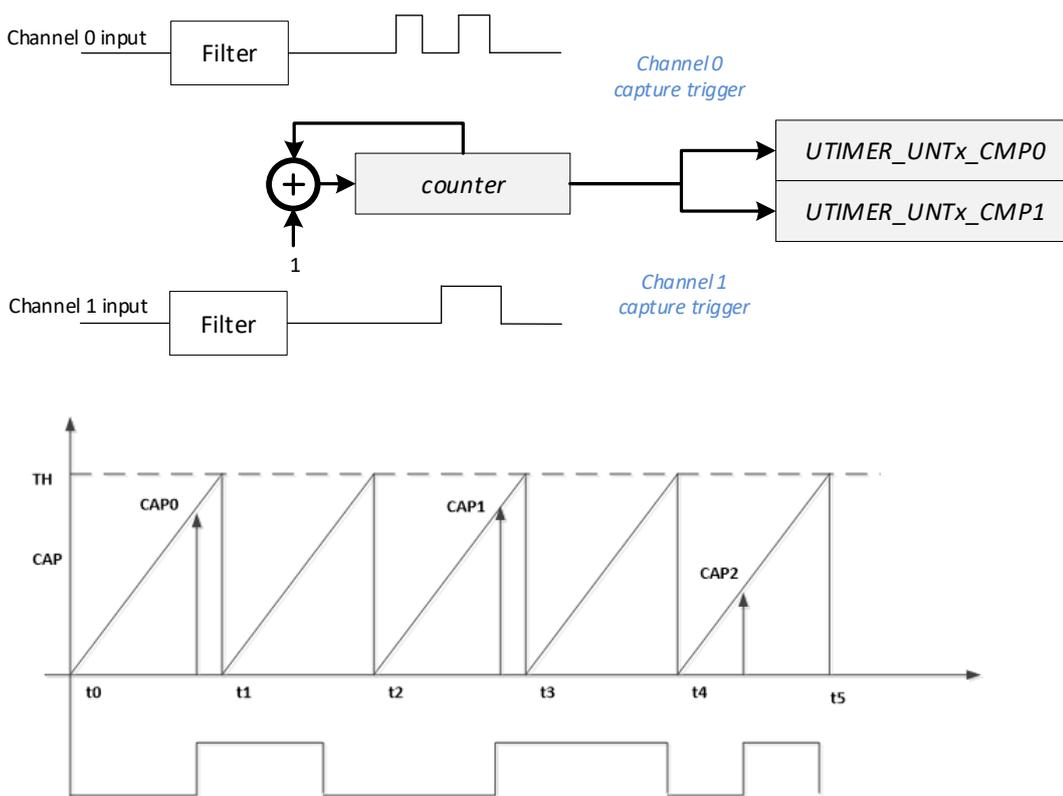


图 8-5 捕获模式

如图 8-5 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 CMP 寄存器中。

8.2.5 编码器

编码器接口支持正交编码信号、符号加脉冲信号、CW/CCW 双脉冲信号三种模式。每个编码器



有两个输入信号。

其中 Encoder0 的输入信号 T1/T2 分别来自 Timer2 Channel0/1 对应的 GPIO 输入；Encoder1 的输入信号 T1/T2 分别来自 Timer3 Channel0/1 对应的 GPIO 输入。开启编码器功能时并不影响 Timer 功能的正常使用。

8.2.5.1 正交编码信号

正交编码信号多用于计数编码器圈数，输入为 T1/T2 两个信号，支持下表中两个模式。

概括来讲，T1/T2 的跳变沿会导致计数器递增或递减。而计数器计数方向（递增或递减）由跳变信号之外的另一个稳态信号的电平高低决定。

如果 T1 发生了上升沿跳变，则看 T2 是高电平还是低电平，如果是高电平则计数器递减，如果是低电平计数器递增，T1 下降沿计数器变化相反。

如果 T2 发生了上升沿跳变，则看 T1 是高电平还是低电平，如果是高电平则计数器递增，如果是低电平计数器递减，T2 下降沿计数器变化相反。

以下式子表示

$$\text{Counter Up} = (T1 \neq T2) @ (T1 \text{ triggering edges}) | (T1 == T2) @ (T2 \text{ triggering edges})$$

$$\text{Counter Down} = (T1 == T2) @ (T1 \text{ triggering edges}) | (T1 \neq T2) @ (T2 \text{ triggering edges})$$

表 8-1 编码器正交编码工作模式

计数模式	T1/T2 电平状态(稳态信号)	T1 变化边沿状态		T2 变化边沿状态	
		上升沿	下降沿	上升沿	下降沿
仅 T1 计数	T2 高	递减	递增	不计数	不计数
	T2 低	递增	递减	不计数	不计数
	T2 高	递减	递增	不计数	不计数
T1/T2 都计数	T2 低	递增	递减	不计数	不计数
	T1 高	不计数	不计数	递增	递减
	T1 低	不计数	不计数	递减	递增

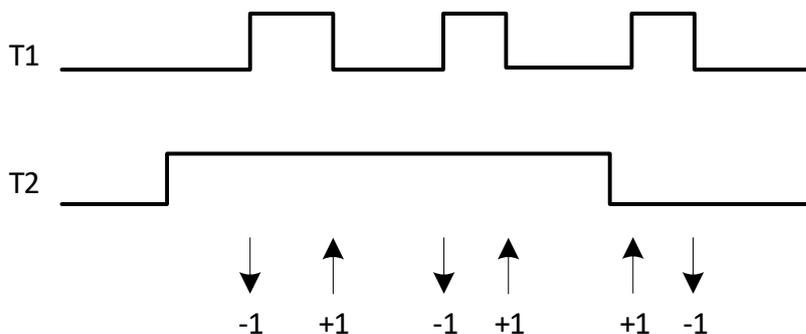


图 8-6 编码器只在 T1 时刻计数的正交编码信号计数情况



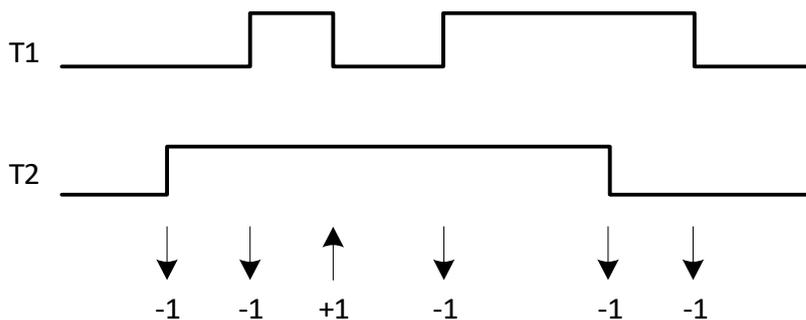


图 8-7 编码器在 T1 或 T2 时刻计数的正交编码信号计数情况

8.2.5.2 符号加脉冲信号

这种工作模式下，T1 为脉冲信号，T2 为符号信号。T1 的边沿触发计数，T2 电平控制计数方向，高则递增，低则递减。可以配置仅 T1 上升沿计数还是 T1 上升下降沿都计数。

Counter Up = (T2==1) @ (T1 triggering edges)

Counter Down = (T2==0) @ (T1 triggering edges)

表 8-2 编码器符号加脉冲工作模式

计数模式	T2 电平状态 (稳态信号)	T1 变化边沿状态	
		上升沿	下降沿
仅 T1 上升沿	高	递增	不计数
	低	递减	不计数
T1 上升下降沿	高	递增	递增
	低	递减	递减

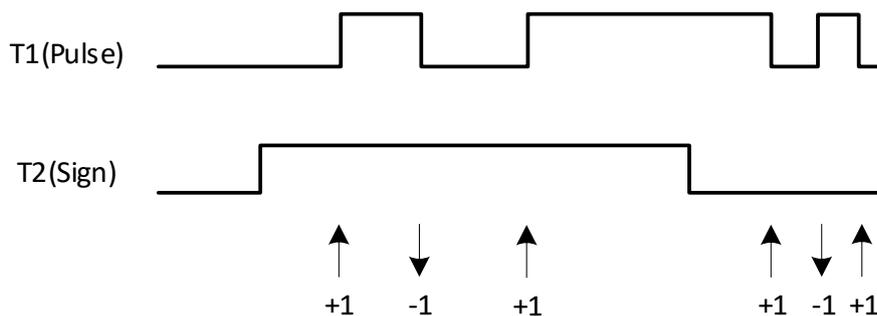


图 8-8 编码器在 T1 上升下降沿都计数的符号加脉冲信号计数情况



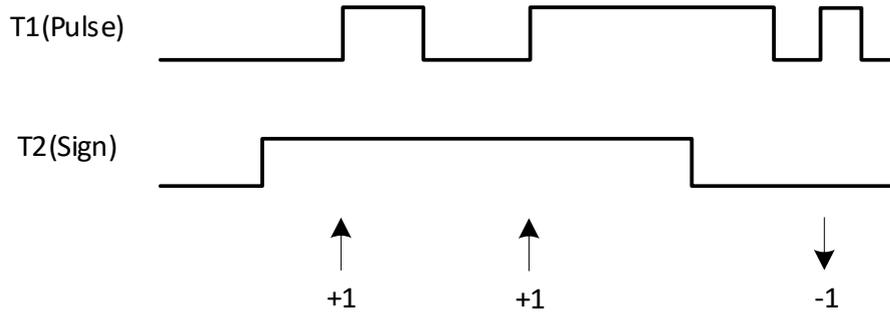


图 8-9 编码器在仅 T1 上升沿计数的符号加脉冲信号计数情况

8.2.5.3 CCW/CW 双脉冲信号

在 T1 跳变时计数器递增，在 T2 跳变时计数器递减。可以配置计数器仅在上升沿变化或者在上升下降沿都变化。以下式表示

$$\text{Counter Up} = 1 @ (\text{T1 triggering edges})$$

$$\text{Counter Down} = 1 @ (\text{T2 triggering edges})$$

表 8-3 编码器 CCW/CW 双脉冲工作模式

计数模式	变化边沿状态			
	T1 上升沿	T1 下降沿	T2 上升沿	T2 下降沿
T1/T2 上升沿	递增	不计数	递减	不计数
T1/T2 上升下降沿	递增	递增	递减	递减

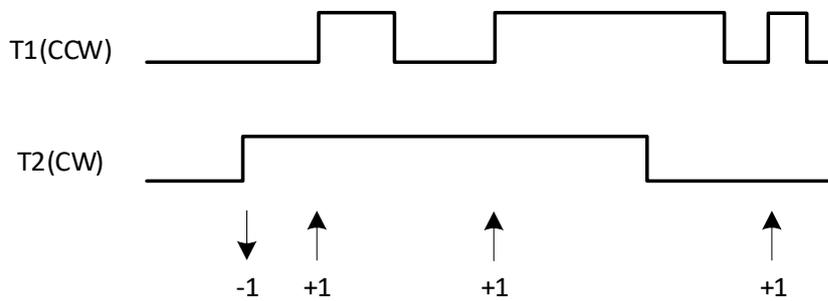


图 8-10 编码器仅在 T1/T2 上升沿计数的 CCW/CW 双脉冲信号计数情况



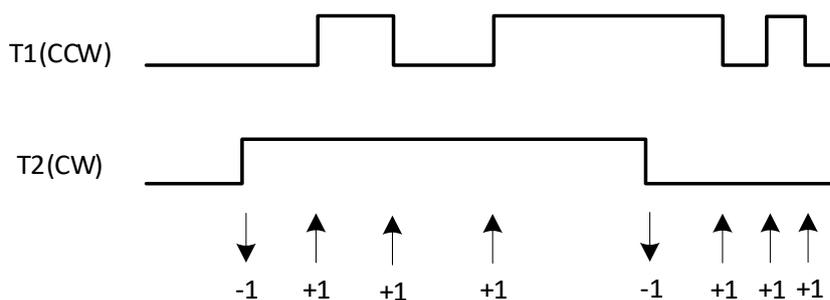


图 8-11 编码器在 T1/T2 上升下降沿计数的 CCW/CW 双脉冲信号计数情况

8.3 寄存器

8.3.1 地址分配

通用定时器模块在芯片中的基地址是 0x4000_3500

表 8-4 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_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_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_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_ECD0_CFG	0x80	Encoder0 配置寄存器
UTIMER_ECD0_TH	0x84	Encoder0 计数门限寄存器



UTIMER_ECD0_CNT	0x88	Encoder0 计数值寄存器
UTIMER_ECD1_CFG	0x90	Encoder1 配置寄存器
UTIMER_ECD1_TH	0x94	Encoder1 计数门限寄存器
UTIMER_ECD1_CNT	0x98	Encoder1 计数值寄存器
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

8.3.2 Time 寄存器

Timer x, 其中 x 可以为 0,1,2,3。注意, Encoder0 复用了 Timer2 的输入端口, Encoder1 复用了 Timer3 的输入端口; 开启 Encoder 功能时, 并不影响对应 Timer 的正常使用。

8.3.2.1 Timer x 配置寄存器 UTIMER_UNTx_CFG

表 8-5 Timer x 配置寄存器 UTIMER_UNTx_CFG

名称	复位值	偏移	位置	RW	说明
UTIMER_UNT0_CFG	0x0	0x00	[31:11]	NA	未使用
			[10]	RW	Timer x 使能配置寄存器 TON, 当 TON 为 0 时, Timer x 停止计数, 同时所有中断标志位输出 0。
			[9:8]	RW	Timer x 计数器频率配置 CLK_DIV[1:0], 计数器计数频率是系统主频率的 1~8 分频 00: 1 分频, 01: 2 分频, 10: 4 分频, 11: 8 分频
			[7]	RW	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]	RW	通道 1 的工作模式, 0, 比较模式, 在计数器计数值为 0 值和通道 1 比较捕获寄存器值时分别将两个不同的电平送出通道 1。 1, 捕获模式, 当通道 1 输入信号发生捕获事件时, 将计数器计数值存入通道 1 比较捕获寄存器。
			[5]	RW	通道 1 下降沿捕获事件使能。值为 1 时, 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]	RW	通道 1 上升沿捕获事件使能。值为 1 时,



					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]	RW	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]	RW	通道 0 的工作模式， 0，比较模式，输出方波，在通道 0 计数器计数值等于 0 或等于比较捕获寄存器值时发生翻转。 1，捕获模式，当通道 0 输入信号发生捕获事件时，将计数器计数值存入通道 0 比较捕获寄存器。
			[1]	RW	通道 0 下降沿捕获事件使能。当此位值为 1 时，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]	RW	通道 0 上升沿捕获事件使能。当此位值为 1 时，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。

8.3.2.2 Timer x 门限寄存器 UTIMER_UNTx_TH

表 8-6 Timer x 门限寄存器 UTIMER_UNTx_TH

名称	复位值	偏移	位置	RW	说明
UTIMER_UNTx_TH	0x0	0x04	[31:16]	NA	未使用
			[15:0]	RW	Timer x 计数器计数门限。计数器从 0 计数到 UTIMER_UNTx_TH 值后再次回 0 开始计数。

8.3.2.3 Timer x 计数寄存器 UTIMER_UNTx_CNT

表 8-7 Timer x 计数寄存器 UTIMER_UNTx_CNT

名称	复位值	偏移	位置	RW	说明
UTIMER_UNTx_CNT	0x0	0x08	[31:16]	NA	未使用
			[15:0]	RW	Timer x 计数器当前计数值。写操作可以写入新的计数值。



8.3.2.4 Timer x 通道 0 比较捕获寄存器 UTIMER_UNTx_CMP0

表 8-8 Timer x 通道 0 比较捕获寄存器 UTIMER_UNTx_CMP0

名称	复位值	偏移	位置	RW	说明
UTIMER_UNTx_CMP0	0x0	0x0C	[31:16]	NA	未使用
			[15:0]	RW	Timer x 通道 0 工作在比较模式时, 当计数器计数值等于 UTIMER_UNTx_CMP0 时, 发生比较事件。 Timer x 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_UNTx_CMP0 寄存器。

8.3.2.5 Timer x 通道 1 比较捕获寄存器 UTIMER_UNTx_CMP1

表 8-9 Timer x 通道 1 比较捕获寄存器 UTIMER_UNTx_CMP1

名称	复位值	偏移	位置	RW	说明
UTIMER_UNTx_CMP1	0x0	0x10	[31:16]	NA	未使用
			[15:0]	RW	Timer x 通道 1 工作在比较模式时, 当计数器计数值等于 UTIMER_UNTx_CMP1 时, 发生比较事件。 Timer x 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_UNTx_CMP1 寄存器。

8.3.3 Encoder x 寄存器

Encoder x, 其中 x 可以为 0,1。

8.3.3.1 Encoder x 配置寄存器 UTIMER_ECDx_CFG

表 8-10 Encoder x 配置寄存器 UTIMER_ECDx_CFG

名称	复位值	偏移	位置	RW	说明
UTIMER_ECDx_CFG	0x0	0x80	[31:11]	NA	未使用
			[10]	RW	CCW+SIGN/CCW+CW 两种模式下, 是否在下降沿进行计数 (上升沿总是计数)
			[9:8]	RW	Encoder x 编码器模式选择 00: counting on T1, 01: counting on T1 & T2 以上两种模式都为正交编码信号计数模式 10: CCW+SIGN, 符号加脉冲信号计数模



					式 11: CCW+CW, CCW+CW 双脉冲信号计数模式
			[7:0]	RW	系统保留, 必须写入 0.

8.3.3.2 Encoder x 计数门限寄存器 UTIMER_ECDx_TH

表 8-11 Encoder x 计数门限寄存器 UTIMER_ECDx_TH

名称	复位值	偏移	位置	RW	说明
UTIMER_ECDx_TH	0x0	0x84	[31:16]	NA	未使用
			[15:0]	RW	Encoder x 计数门限 TH。编码器向上计数 (增) 到 TH 值后, 再次向上计数会导致计数器回到 0。编码器向下计数 (减) 到 0 值后, 再次向下计数会导致计数器回到 TH。

8.3.3.3 Encoder x 计数值寄存器 UTIMER_ECDx_CNT

表 8-12 Encoder x 计数值寄存器 UTIMER_ECDx_CNT

名称	复位值	偏移	位置	RW	说明
UTIMER_ECDx_CNT	0x0	0x88	[31:16]	NA	未使用
			[15:0]	R	Encoder0 计数值。

8.3.4 滤波控制寄存器

8.3.4.1 UTIMER_FLT_TH01

表 8-13 滤波控制寄存器 UTIMER_FLT_TH01

名称	复位值	偏移	位置	RW	说明
UTIMER_FLT_TH01	0x0	0xA0	[31:16]	NA	未使用
			[15:12]	RW	TIM1_CH11 信号滤波宽度选择 FTH3 取值范围 0~15。。 FTH3 为 0 时, 对 TIM1_CH1 不进行滤波。 FTH3 不为 0 时, 对 TIM1_CH1 信号进行滤波: 滤波宽度为 8 倍 FTH3 寄存器值。 当 TIM1_CH1 电平稳定超过 FTH3x8 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]	RW	TIM1_CH0 信号滤波信号 FTH2。含义同



					FTH3。
			[7:4]	RW	TIM0_CH1 信号滤波信号 FTH1。含义同 FTH3。
			[3:0]	RW	TIM0_CH0 信号滤波信号 FTH0。含义同 FTH3。

8.3.4.2 UTIMER_FLT_TH23

表 8-14 滤波控制寄存器 UTIMER_FLT_TH23

名称	复位值	偏移	位置	RW	说明
UTIMER_FLT_TH23	0x0	0xA4	[31:16]	NA	未使用
			[15:12]	RW	TIM3_CH1 信号滤波宽度选择 FTH7,取值范围 0~15。 FTH7 为 0 时,对 TIM3_CH1 不进行滤波。 FTH7 不为 0 时,对 TIM3_CH1 信号进行滤波: 滤波宽度为 8 倍 FTH7 寄存器值。 当 TIM3_CH1 电平稳定超过 FTH7x8 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]	RW	TIM3_CH0 信号滤波信号 FTH6。含义同 FTH7。
			[7:4]	RW	TIM2_CH1 信号滤波信号 FTH5。含义同 FTH7。
			[3:0]	RW	TIM2_CH0 信号滤波信号 FTH4。含义同 FTH7。

8.3.5 系统控制寄存器

8.3.5.1 UTIMER_CFG

表 8-15 UTIMER 配置寄存器 UTIMER_CFG

名称	复位值	偏移	位置	RW	描述
UTIMER_CFG	0x0	0xF0	[31:10]	NA	未使用
			[9]	RW	1:启动编码器 1, 0:停止编码器 1
			[8]	RW	1:启动编码器 0, 0:停止编码器 0
			[7:0]	RW	系统保留, 推荐写入 0

8.3.6 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。

8.3.6.1 中断使能寄存器 UTIMER_IE

表 8-16 中断使能寄存器 UTIMER_IE

名称	复位值	偏移	位置	RW	说明
UTIMER_IE	0x0	0xF4	[31:16]	NA	未使用
			[15]	RW	Encoder1 上溢出中断使能, 高电平有效 (下同)。当 Encoder1 计数器计数达到计数门限时, 上计数事件触发上溢出中断。
			[14]	RW	Encoder1 下溢出中断使能。当 Encoder1 计数器计数达到 0 时, 下计数事件触发下溢出中断。
			[13]	RW	Encoder0 上溢出中断使能。
			[12]	RW	Encoder0 下溢出中断使能。
			[11]	RW	Timer3 通道 1 比较/捕获中断使能。
			[10]	RW	Timer3 通道 0 比较/捕获中断使能。
			[9]	RW	Timer3 计数器过 0 中断使能。
			[8]	RW	Timer2 通道 1 比较/捕获中断使能。
			[7]	RW	Timer2 通道 0 比较/捕获中断使能。
			[6]	RW	Timer2 计数器过 0 中断使能。
			[5]	RW	Timer1 通道 1 比较/捕获中断使能。
			[4]	RW	Timer1 通道 0 比较/捕获中断使能。
			[3]	RW	Timer1 计数器过 0 中断使能。
			[2]	RW	Timer0 通道 1 比较/捕获中断使能。
[1]	RW	Timer0 通道 0 比较/捕获中断使能。			
[0]	RW	Timer0 计数器过 0 中断使能。			

8.3.6.2 中断标志寄存器 UTIMER_IF

表 8-17 中断标志寄存器 UTIMER_IF

名称	复位值	偏移	位置	RW	说明
UTIMER_IF	0x0	0xF8	[31:16]	NA	未使用
			[15]	RW	Encoder1 上溢出中断标志, 高电平有效, 对此 bit 写 1 可清 0 此 bit (下同)。当 Encoder1 计数器计数达到计数门限时, 上计数事件触发上溢出中断。
			[14]	RW	Encoder1 下溢出中断标志。当 Encoder1 计数器计数达到 0 时, 下计数事件触发下溢出中断。
			[13]	RW	Encoder0 上溢出中断标志。
			[12]	RW	Encoder0 下溢出中断标志。
			[11]	RW	Timer3 通道 1 比较/捕获中断标志
			[10]	RW	Timer3 通道 0 比较/捕获中断标志



		[9]	RW	Timer3 计数器过 0 中断标志
		[8]	RW	Timer2 通道 1 比较/捕获中断标志
		[7]	RW	Timer2 通道 0 比较/捕获中断标志
		[6]	RW	Timer2 计数器过 0 中断标志
		[5]	RW	Timer1 通道 1 比较/捕获中断标志
		[4]	RW	Timer1 通道 0 比较/捕获中断标志
		[3]	RW	Timer1 计数器过 0 中断标志
		[2]	RW	Timer0 通道 1 比较/捕获中断标志
		[1]	RW	Timer0 通道 0 比较/捕获中断标志
		[0]	RW	Timer0 计数器过 0 中断标志



9 HALL 信号处理模块

9.1 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

9.2 寄存器

9.2.1 地址分配

表 9-1HALL 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

9.2.2 HALL 模块配置寄存器 HALL_CFG

表 9-2 HALL 模块配置寄存器 HALL_CFG

名称	复位值	偏移	位置	RW	说明
HALL_CFG	0x0	0x00	[31:30]	NA	未使用
			[29]	RW	HALL 计数器溢出中断使能开关。1，使能；0，关闭。
			[28]	RW	HALL 信号变化中断使能开关，1，使能；0，关闭。
			[27:25]	NA	未使用
			[24]	RW	HALL 模块使能开关。1，使能；0，关闭。
			[23:21]	NA	未使用
			[20]	RW	7/5 滤波开关。1，使能；0，关闭。
			[19:18]	NA	未使用
			[17:16]	RW	HALL 时钟分频系数 00：不分频 01：2 分频 10：4 分频



					11: 8 分频
			[15]	NA	未使用
			[14:0]	RW	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。

9.2.3 HALL 模块信息寄存器 HALL_INFO

表 9-3 HALL 模块信息寄存器 HALL_INFO

名称	复位值	偏移	位置	RW	说明
HALL_INFO	0x0	0x04	[31:18]	NA	未使用
			[17]	RW	HALL 计数器溢出事件标志，写 1 清空
			[16]	RW	HALL 信号变化事件标志，写 1 清空
			[15:11]	RW	系统保留，必须写入 0，读出 0
			[10:8]	R	从 GPIO 输入的原始 HALL 信号值
			[7:3]	RW	系统保留，必须写入 0，读出 0
			[2:0]	R	捕获到的 HALL 值

9.2.4 HALL 宽度计数值寄存器 HALL_WIDTH

表 9-4 HALL 宽度计数值寄存器 HALL_WIDTH

名称	复位值	偏移	位置	RW	说明
HALL_WIDTH	0x0	0x08	[31:0]	R	HALL 宽度计数器值

9.2.5 HALL 模块计数器门限值寄存器 HALL_TH

表 9-5 HALL 模块计数器门限值寄存器 HALL_TH

名称	复位值	偏移	位置	RW	说明
HALL_TH	0x0	0x0C	[31:0]	RW	HALL 计数器门限值

9.2.6 HALL 计数寄存器 HALL_CNT

表 9-6 HALL 计数寄存器 HALL_CNT

名称	复位值	偏移	位置	RW	说明
HALL_CNT	0x0	0x10	[31:0]	RW	HALL 计数值，写入任意值可清零

9.3 实现说明

9.3.1 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。



详细说明见 GPIO 的章节。

9.3.2 工作时钟

HALL 模块工作频率可调。通过配置 CFG 寄存器 B[17:16]，可以选择系统主频的 1/2/4/8 分频作为模块工作频率，滤波和计数均采用该频率工作。

9.3.3 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：

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，否则输出保持上一轮的滤波结果。具体如下图所示

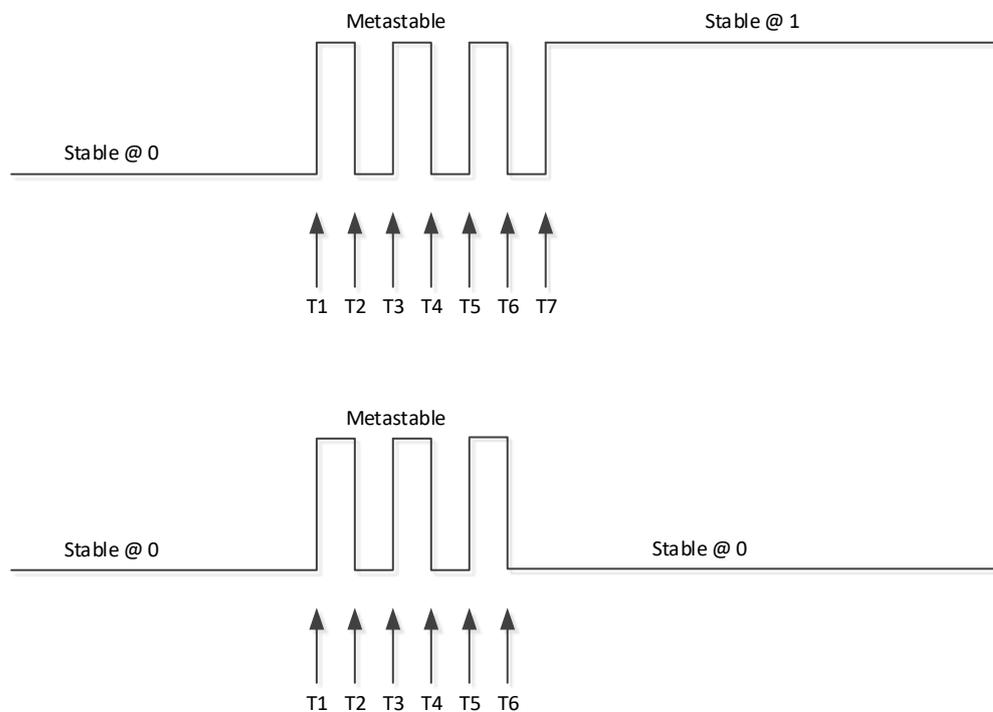


图 9-1 7/5 滤波模块框图

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，否则输出保持上一轮的滤波结果。

通过配置 HALL_CFG 寄存器 B[24] 可以选择是否使能第一级滤波器。

通过配置 HALL_CFG 寄存器 B[14:0] 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^{15} ，在 96MHz 工作频率下，最长滤波宽度为约 340us。

通过访问 HALL_INFO 寄存器 B[2:0] 可以捕捉后的 HALL 信号；B[10:8] 则是滤波前原始 HALL 输入信号。



9.3.4 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 32 位计数器，在 96MHz 工作频率下，最大可以记录约 44.7 秒的时间宽度，达到 10ns 的时间分辨率。

HALL_CNT 从 0 开始计数，当发生 HALL 信号变化时，将此刻的 HALL_CNT 值保存到 HALL_WIDTH 寄存器，将此刻的 HALL 信号保存到 HALL_INFO 寄存器 B[2:0]，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。

9.3.5 中断

捕获、溢出事件触发中断，中断使能控制位位于 HALL_CFG 寄存器 B[29:28]，中断标志位位于 HALL_INFO 寄存器 B[17:16]。终端标志可以通过对 HALL_INFO 寄存器的写操作清空。

9.3.6 数据流程

HALL 模块的数据流程如下图所示，fclk 为系统时钟。

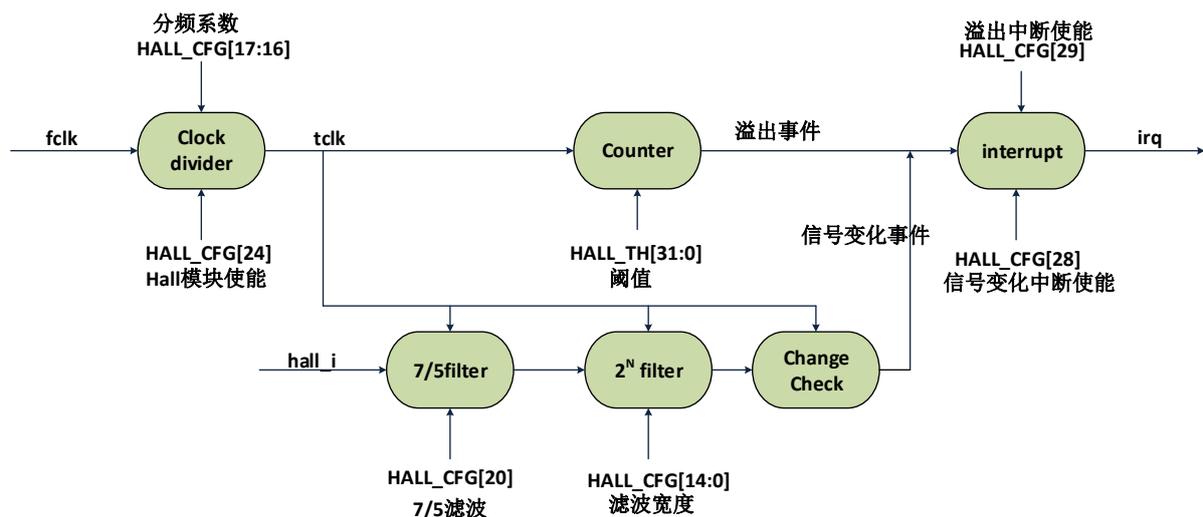


图 9-2 数据流程框图

10 MCPWM

10.1 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 UP 计数器，用于提供一个基础周期。计数器的时钟频率有四种选项，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

包含四组 Timer 定时模块。产生 4 路和 MCPWM 同时基的定时信息，用于触发两组 ADC 模块同步采样。

包含一组急停保护模块，用于快速关断 MCPWM 模块输出而不依赖 MCU 的处理。MCPWM 模块可输入 4 路急停信号，其中两路来自外部 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。

对比较器模块的输出信号有独立滤波模块，可以用做急停保护模块的输出，也可以产生单独的比较器中断事件。

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 10-1 MCPWM 模块框图。



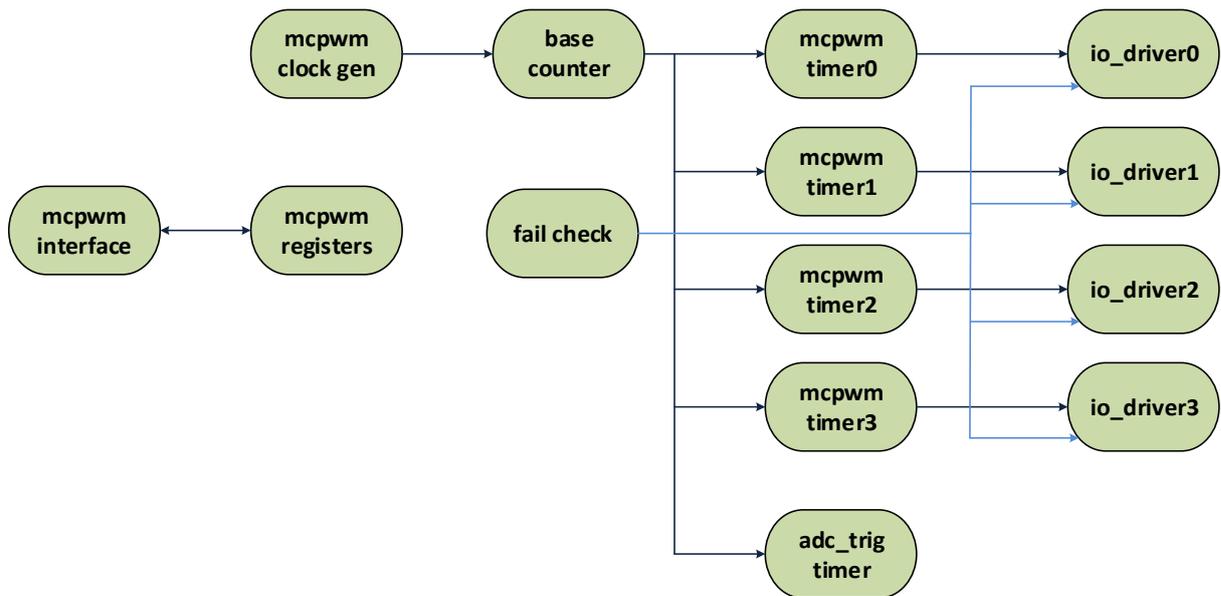


图 10-1 MCPWM 模块框图

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

10.1.1 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 TH，计数器从 t0 开始从 -TH 递增计数 (UP)，在 t1 过 0，在 t2 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 $(2 \times TH + 1) / f_{clk}$ 。fclk 是计数时钟频率。

在 t0/t1(本次 t0 即上一次 t2)可产生定时事件中断，IF[0]和 IF[1]将被置位。

可通过寄存器配置该定时器启动和停止。

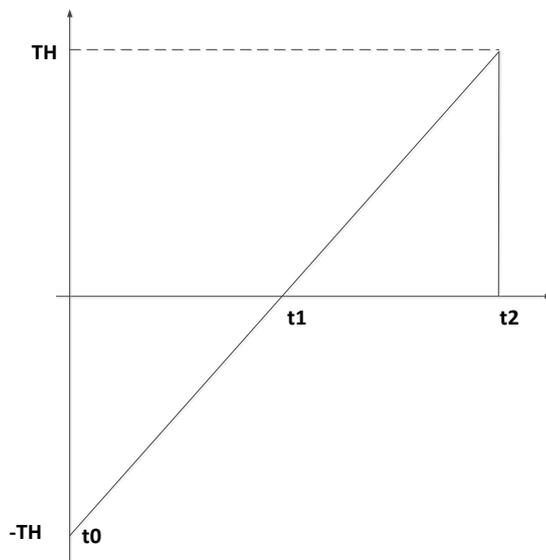


图 10-2 Base Counter t0/t1 时序

在运行 MCPWM 模块前，用户一般需将对应的比较门限值，死区寄存器配置好。在实际运行过



程中，也可动态改变比较门限值和 PWM 周期值，可手动更新，也可以硬件自动更新。硬件更新，仅在 t0 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 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，产生装载完毕中断。

通过配置选择更新发生在 t0 或者 t1 或者二者皆可，配置更新间隔数，间隔数为 1~16。最快的配置为更新发生在 t0 和 t1，连续发生。最慢的配置为更新发生在 t1，每 16 个 t1 更新一次。图 10-3。

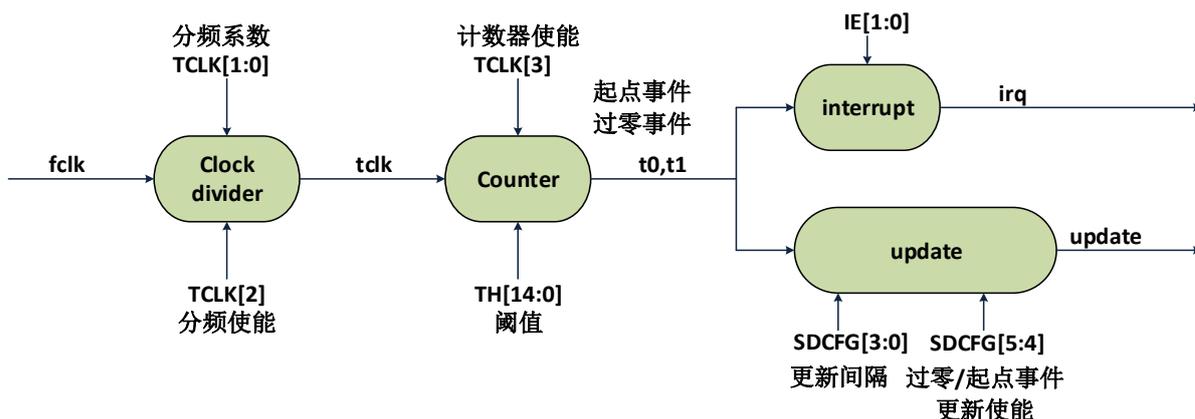


图 10-3 Base Counter 数据流程图

10.1.2 Fail Check 模块

该模块主要是检测电机反馈回的实际短路情况，实现快速关断 PWM 的输出。有两个通道 FAIL0 和 FAIL1，共有 4 个源头 BK[1:0]和 CMP[1:0]。BK 来自 IO，CMP 来自芯片内部的比较器模块。同时，CMP 还可单独送往 MCPWM 模块中断模块，产生对应中断信号。

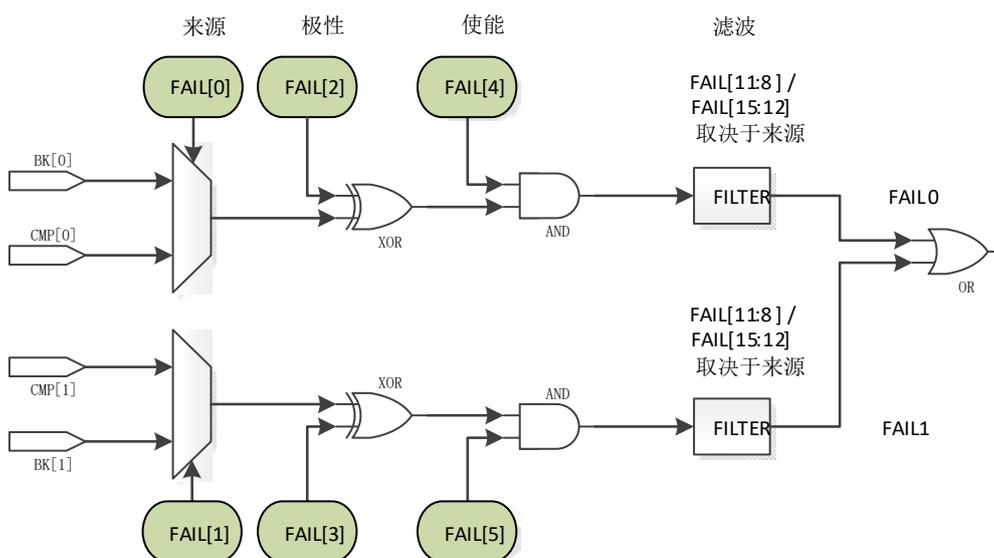


图 10-4 MCPWM FAIL 逻辑示意图

Filter 滤波模块的时钟，来自 MCLK，可实现 1--16 倍的分频，分频后的时钟用于采样 Filter 的输入信号，滤波宽度为 16，即输入信号必须稳定至少 16 个分频后的时钟，硬件才判定其为有效输入信号。滤波宽度的公式为，其中 T_{MCLK} 为 MCLK 的时钟周期，96MHz 对应 10.4ns。

$$T = T_{MCLK} \times (TCLK[15:12]+1) \times 16$$

一旦发生 Fail，硬件强制将 IO 输出的 FAIL[15:8]寄存器的值，此时 FAIL[15:8]的值直接输出，不受极性控制等影响。

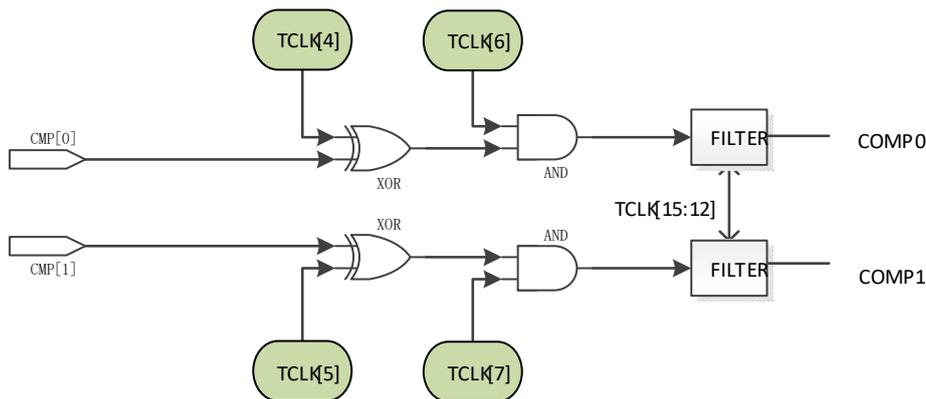


图 10-5 MCPWM COMP 逻辑示意图

10.1.3 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>_P 关闭），无死区

10.1.4 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：



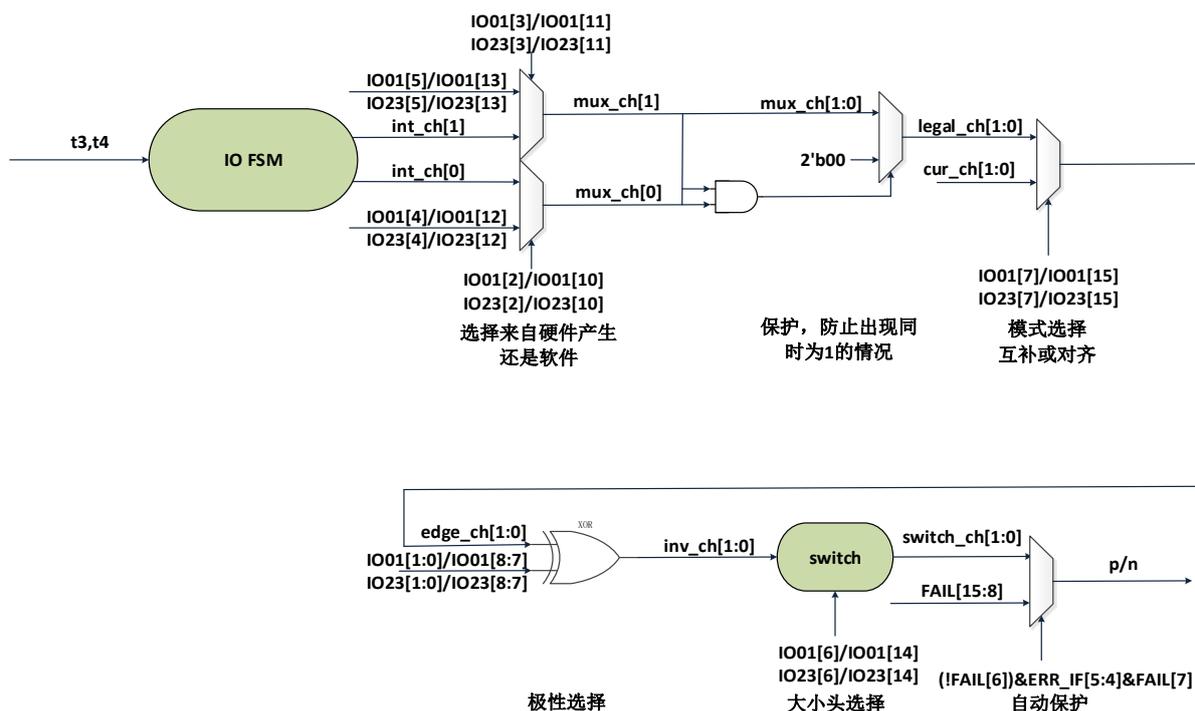


图 10-6 IO Driver 模块数据流程图

10.1.4.1 MCPWM 波形输出-中心对齐模式

4 个 MCPWM IO Driver 采用独立的控制门限，独立死区宽度（每一对互补 IO 的死区需要独立配置，即 4 个死区配置寄存器），共享数据更新事件。

采用 TH<n>0 和 TH<n>1 控制第<n>个 MCPWM IO 的启动、关闭动作，n 为 1/2/3/4。

当计数器 CNT 值向上计数达到 TH<n>0 时，在 t3 时刻关闭 CH<n>_N，经过死区延时 Tdead，打开 CH<n>_P。

当计数器 CNT 值向上计数达到 TH<n>1 时，在 t4 时刻关闭 CH<n>_P，经过死区延时 Tdead，打开 CH<n>_N。

采用独立的启动和关闭时间控制，可以提供相位控制的能力。

死区延时保证 CH<n>_P/CH<n>_N 不会同时为高，避免短路发生。

t3/t4 时刻均会产生相应中断。



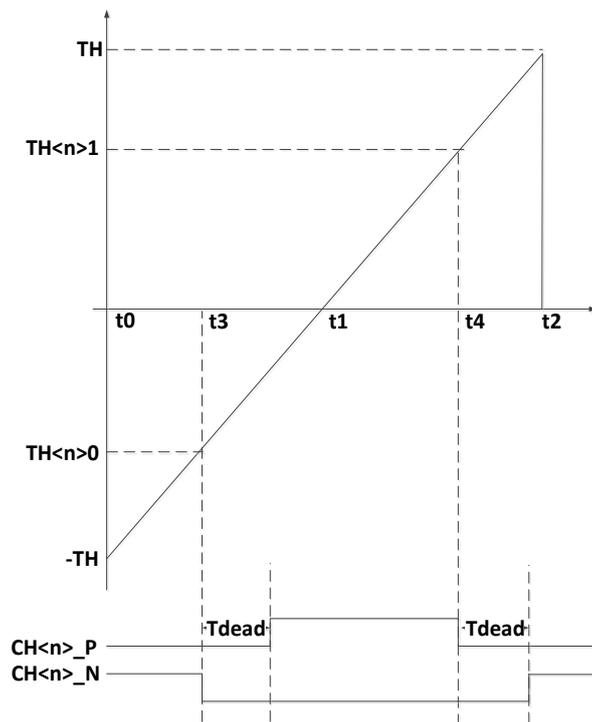


图 10-7 MCPWM 时序 TH<n>0 和 TH<n>1-互补模式

10.1.4.2 MCPWM 波形输出-边沿对齐模式

边沿对齐模式中，在 t_0 时刻 $CH<n>_P/CH<n>_N$ 同时置 1，在 t_3 时刻， $CH<n>_P$ 变低，在 t_4 时刻， $CH<n>_N$ 变低。

t_3/t_4 均会产生相应中断。

边沿对齐模式下， $CH<n>_P/CH<n>_N$ 无需死区保护。

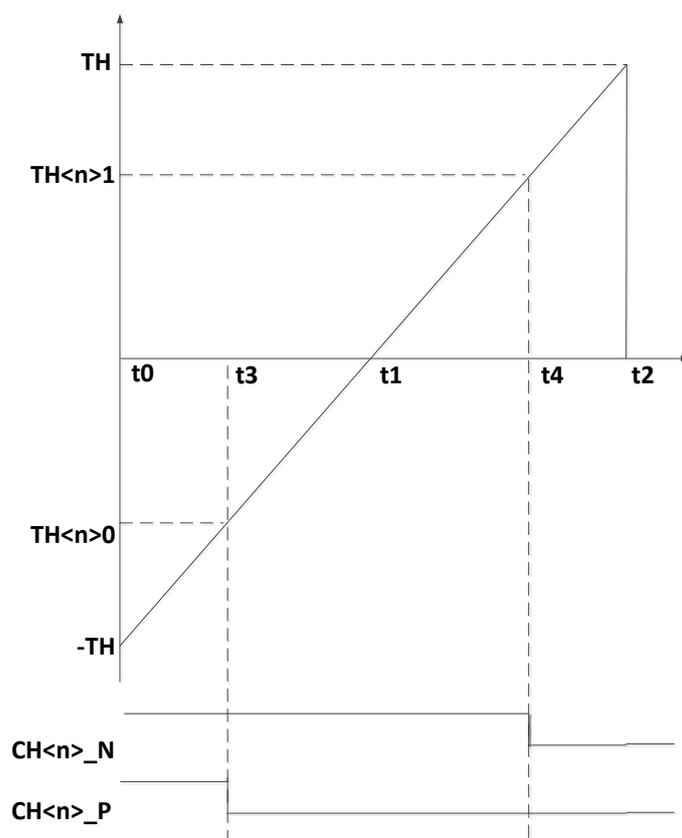


图 10-8 MCPWM 时序边沿对齐模式

10.1.4.3 MCPWM IO 死区控制

MCPWM IO 是一对互斥控制信号 $CH\langle n \rangle_P/CH\langle n \rangle_N$ ，控制如下图所示的电路，

当 $CH\langle n \rangle_P$ 为高/ $CH\langle n \rangle_N$ 为低时， V_{out} 输出高 (VDD)；

当 $CH\langle n \rangle_P$ 为低/ $CH\langle n \rangle_N$ 为高时， V_{out} 输出低 (VSS)；

当 $CH\langle n \rangle_P$ 为高/ $CH\langle n \rangle_N$ 为高时， V_{out} 输出不确定，但是会产生 VDD 到 VSS 的短路；

当 $CH\langle n \rangle_P$ 为低/ $CH\langle n \rangle_N$ 为低时， V_{out} 输出不确定。

必须避免 $CH\langle n \rangle_P/CH\langle n \rangle_N$ 同时为高的情况，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 $CH\langle n \rangle_P/CH\langle n \rangle_N$ 冲突检测，发生冲突时，自动将 IO 拉低，同时给出错误中断（中断保持，直到 MCU 写 0）。

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 PWM 模式为互补，仍然由硬件保证不同时为高或者为低。



CH<n>_P/CH<n>_N，在 IO 上可以互换。

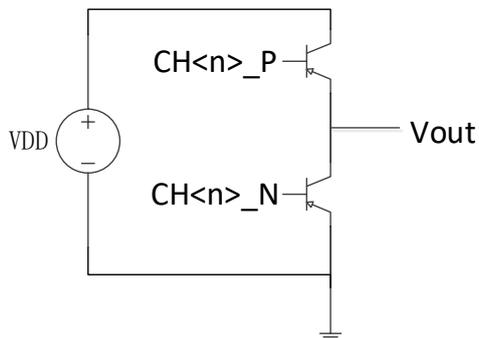


图 10-9 MCPWM IO 控制示意图

10.1.4.4 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。

10.1.4.5 MCPWM IO 自动保护

当发生短路事件（来自 Fail Check 模块），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（FAIL[15:8]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 FAIL[15:8]指定值，当用户配置完毕，MCPWM 正常工作后，配置 FAIL[6]（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，MCU Halt 时，PWM 停止输出，输出 FAIL[15:8]的值。
- IO Driver 发现的由于比较寄存器配置带来的 CH<n>P/CH<n>N 冲突保护不采用本方案实现。

10.1.5 ADC Trigger Timer 模块

Timer0/1/2/3 提供 ADC 采样控制。当计数器计数到 TMR0/TMR1/TMR2/TMR3，产生定时事件驱动 ADC 采样动作。该输出信号应该也能同时输出到 IO，便于调试之用。

表 10-1 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
----------	------

10.2 寄存器

10.2.1 地址分配

MCPWM 模块寄存器的基地址是 0x4000_3600 寄存器列表

表 10-2 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_IE	0x30	MCPWM 中断控制寄存器
MCPWM_IF	0x34	MCPWM 中断标志位寄存器
MCPWM_EIE	0x38	MCPWM 异常中断控制寄存器
MCPWM{EIF	0x3C	MCPWM 异常中断标志位寄存器
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_UPDATE	0x5C	MCPWM 加载控制寄存器
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_TH	0x70	MCPWM 门限值寄存器
MCPWM_PRT	0x74	MCPWM 保护寄存器
MCPWM_CNT	0x78	MCPWM 计数器寄存器
MCPWM_DTH00	0x80	MCPWM CH0 N 通道死区宽度控制寄存器
MCPWM_DTH01	0x84	MCPWM CH0 P 通道死区宽度控制寄存器
MCPWM_DTH10	0x88	MCPWM CH1 N 通道死区宽度控制寄存器
MCPWM_DTH11	0x8C	MCPWM CH1 P 通道死区宽度控制寄存器
MCPWM_DTH20	0x90	MCPWM CH2 N 通道死区宽度控制寄存器
MCPWM_DTH21	0x94	MCPWM CH2 P 通道死区宽度控制寄存器



MCPWM_DTH30	0x98	MCPWM CH3 N 通道死区宽度控制寄存器
MCPWM_DTH31	0x9C	MCPWM CH3 P 通道死区宽度控制寄存器

10.2.2 MCPWM_TH00

无写保护的寄存器

表 10-3 MCPWM_TH00 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH00	0x0	0x00	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.3 MCPWM_TH01

无写保护的寄存器

表 10-4 MCPWM_TH01 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH01	0x0	0x04	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.4 MCPWM_TH10

无写保护的寄存器

表 10-5 MCPWM_TH10 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH10	0x0	0x08	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.5 MCPWM_TH11

无写保护的寄存器



表 10-6 MCPWM_TH11 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH11	0x0	0x0C	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.6 MCPWM_TH20

无写保护的寄存器

表 10-7 MCPWM_TH20 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH20	0x0	0x10	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.7 MCPWM_TH21

无写保护的寄存器

表 10-8 MCPWM_TH21 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH21	0x0	0x14	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.8 MCPWM_TH30

无写保护的寄存器

表 10-9 MCPWM_TH30 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH30	0x0	0x18	[31:16]	NA	未使用
			[15:0]	RW	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。



10.2.9 MCPWM_TH31

无写保护的寄存器

表 10-10 MCPWM_TH31 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH31	0x0	0x1C	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.10 MCPWM_TMR0

无写保护的寄存器

表 10-11 MCPWM_TMR0 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TMR0	0x0	0x20	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.11 MCPWM_TMR1

无写保护的寄存器

表 10-12 MCPWM_TMR1 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TMR1	0x0	0x24	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.12 MCPWM_TMR2

无写保护的寄存器

表 10-13 MCPWM_TMR2 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TMR2	0x0	0x28	[31:16]	NA	未使用
			[15:0]	RW	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。
--	--	--	--	--	----------------

10.2.13 MCPWM_TMR3

无写保护的寄存器

表 10-14 MCPWM_TMR3 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TMR3	0x0	0x2C	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.14 MCPWM_IE

写保护的寄存器

表 10-15 MCPWM_IE 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_IE	0x0	0x30	[14]	RW	TH (THxx) /TMR 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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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。

10.2.15 MCPWM_IF

写保护的寄存器

表 10-16 MCPWM_IF 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_IF	0x0	0x34	[14]	RW	TH (THxx) /TMR 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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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。

10.2.16 MCPWM_EIE

无写保护的寄存器

表 10-17 MCPWM_EIE 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_EIE	0x0	0x30	[7]	RW	比较器 1 中断源使能。1, 使能; 0, 关闭。



			[6]	RW	比较器 0 中断源使能。1, 使能; 0, 关闭。
			[5]	RW	FAIL1 中断源使能。1, 使能; 0, 关闭。
			[4]	RW	FAIL0 中断源使能。1, 使能; 0, 关闭。
			[3]	RW	MCPWM CH3_P 和 CH3_N 同时有效, 中断源使能。1, 使能; 0, 关闭。
			[2]	RW	MCPWM CH2_P 和 CH2_N 同时有效, 中断源使能。1, 使能; 0, 关闭。
			[1]	RW	MCPWM CH1_P 和 CH1_N 同时有效, 中断源使能。1, 使能; 0, 关闭。
			[0]	RW	MCPWM CH0_P 和 CH0_N 同时有效, 中断源使能。1, 使能; 0, 关闭。

10.2.17 MCPWM{EIF

无写保护的寄存器

表 10-18 MCPWM{EIF 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM{EIF	0x0	0x30	[7]	RW	比较器 1 中断源事件。1, 发生; 0, 没发生。写 1 清零。
			[6]	RW	比较器 0 中断源事件。1, 发生; 0, 没发生。写 1 清零。
			[5]	RW	FAIL1 中断源事件。1, 发生; 0, 没发生。可写 1 清零。
			[4]	RW	FAIL0 中断源事件。1, 发生; 0, 没发生。可写 1 清零。
			[3]	RW	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]	RW	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]	RW	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]	RW	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。

MCPWM{EIF[7:6]对应的是 CMP 中断, 而与 MCPWM 中断无关。当 MCPWM{EIF[7:6]使能, 同时 MCPWM{EIF[7:6]置位时, 处理器会收到 CMP 中断事件。而 MCPWM{EIF[5:0]则属于 MCPWM 中断。



10.2.18 MCPWM_IO01

写保护的寄存器

表 10-19 MCPWM_IO01 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_IO01	0x0	0x50	[15]	RW	CH1 工作模式选择。1: Edge 模式; 0: 互补模式。
			[14]	RW	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]	RW	当 B[11]为 1 时, B[13]的值输出到 CH1 P。
			[12]	RW	当 B[10]为 1 时, B[12]的值输出到 CH1 N。
			[11]	RW	CH1 P 来源。1: 来自 B[13]; 0: MCPWM 内部计数器产生。
			[10]	RW	CH1 N 来源。1: 来自 B[12]; 0: MCPWM 内部计数器产生。
			[9]	RW	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]	RW	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]	RW	CH0 工作模式选择。1: Edge 模式; 0: 互补模式。
			[6]	RW	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]	RW	当 B[3]为 1 时, B[5]的值输出到 CH0 P。
			[4]	RW	当 B[2]为 1 时, B[4]的值输出到 CH0 N。
			[3]	RW	CH0 P 来源。1: 来自 B[5]; 0: MCPWM 实际运行系统中计数器产生。
			[2]	RW	CH0 N 来源。1: 来自 B[4]; 0: MCPWM 实际运行系统中计数器产生。
			[1]	RW	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]	RW	CH0 N 极性选择。1: CH0 N 信号取反输出; 0: CH0 N 信号正常输出。



10.2.19 MCPWM_IO23

写保护的寄存器

表 10-20 MCPWM_IO23 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_IO23	0x0	0x54	[15]	RW	CH3 工作模式选择。1: Edge 模式; 0: 互补模式。
			[14]	RW	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]	RW	当 B[11]为 1 时, B[13]的值输出到 CH3 P。
			[12]	RW	当 B[10]为 1 时, B[12]的值输出到 CH3 N。
			[11]	RW	CH3 P 来源。1: 来自 B[13]; 0: MCPWM 实际运行系统中计数器产生。
			[10]	RW	CH3 N 来源。1: 来自 B[12]; 0: MCPWM 实际运行系统中计数器产生。
			[9]	RW	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]	RW	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]	RW	CH2 工作模式选择。1: Edge 模式; 0: 互补模式。
			[6]	RW	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]	RW	当 B[3]为 1 时, B[5]的值输出到 CH2 P。
			[4]	RW	当 B[2]为 1 时, B[4]的值输出到 CH2 N。
			[3]	RW	CH2 P 来源。1: 来自 B[5]; 0: MCPWM 实际运行系统中计数器产生。
			[2]	RW	CH2 N 来源。1: 来自 B[4]; 0: MCPWM 实际运行系统中计数器产生。
			[1]	RW	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]	RW	CH2 N 极性选择。1: CH2 N 信号取反输出; 0: CH2 N 信号正常输出。



10.2.20 MCPWM_SDCFG

写保护的寄存器

表 10-21 MCPWM_SDCFG 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_SDCFG	0x0	0x58	[5]	RW	t1 (过零) 事件更新使能。1: 使能; 0, 关闭。
			[4]	RW	t0 (起点) 事件更新使能。1: 使能; 0, 关闭。
			[3:0]	RW	更新间隔。一旦 t0 和 t1 事件发生次数同 B[3:0] 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。

10.2.21 MCPWM_UPDATE

无写保护的寄存器

表 10-22 MCPWM_UPDATE 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_UPDATE	0x0	0x5C	[12]	RW	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]	RW	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]	RW	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]	RW	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]	RW	手动将加载 MCPWM_TMR0 寄存器的内容到 MCPWM 运行系统中。1: 加载; 0: 不加载。
			[7]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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]	RW	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。

10.2.22 MCPWM_TCLK

写保护的寄存器

表 10-23 MCPWM_TCLK 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TCLK	0x0	0x60	[15:12]	RW	来自比较器结果的滤波时钟分频寄存器, 基于系统时钟分频。计算公式如下: 系统时钟 / (B[15:12] + 1)。分频范围是 1-16。
			[11:8]	RW	来自 GPIO 输入的滤波时钟分频寄存器, 基于系统时钟分频。计算公式如下: 系统时钟 / (B[11:8] + 1)。分频范围是 1-16。
			[7]	RW	比较器 1 结果, 输入使能。1: 使能; 0: 关闭。
			[6]	RW	比较器 0 结果, 输入使能。1: 使能; 0: 关闭。
			[5]	RW	比较器 1 结果, 极性选择开关。1: 结果取反, 即低电平有效; 0: 结果不取反, 即高电平有效。
			[4]	RW	比较器 0 结果, 极性选择开关。1: 结果取反, 即低电平有效; 0: 结果不取反, 即高电平有效。
			[3]	RW	MCPWM 实际运行计数器使能开关。1: 使能; 0: 关闭。
			[2]	RW	MCPWM 工作时钟使能。1: 使能; 0: 关闭。
			[1:0]	RW	MCPWM 工作时钟分频寄存器。计算公式: 系统时钟 / (B[1:0] + 1)。范围 1-4。

10.2.23 MCPWM_FAIL

写保护的寄存器



表 10-24 MCPWM_FAIL 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_FAIL	0x0	0x64	[15]	RW	CH3 N 通道默认值
			[14]	RW	CH3 P 通道默认值
			[13]	RW	CH2 N 通道默认值
			[12]	RW	CH2 P 通道默认值
			[11]	RW	CH1 N 通道默认值
			[10]	RW	CH1 P 通道默认值
			[9]	RW	CH0 N 通道默认值
			[8]	RW	CH0 P 通道默认值
			[7]	RW	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]	RW	控制 MCPWM CH P 和 N 输出值。 1: 输出 MCPWM 产生的正常信号 0: 输出 B[15:8]默认值，此默认值不受极性/ 通道选择等控制。MCPWM_EIF[5:4]任意一位 变 1 将触发 B[6]变成 0，输出默认值。
			[5]	RW	FAIL1 输入使能。1: 使能；0: 关闭。
			[4]	RW	FAIL0 输入使能。1: 使能；0: 关闭。
			[3]	RW	FAIL1 极性选择。1: 信号取反输入；0: 信号 正常输入。
			[2]	RW	FAIL0 极性选择。1: 信号取反输入；0: 信号 正常输入。
[1]	RW	FAIL1 来源选择。1: 比较器 1 的结果；0: 来 自 GPIO 第 1 路。			
[0]	RW	FAIL0 来源选择。1: 比较器 0 的结果；0: 来 自 GPIO 第 0 路。			

10.2.24 MCPWM_TH

写保护的寄存器

表 10-25 MCPWM_TH 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_TH	0x0	0x70	[31:15]	NA	未使用
			[14:0]	RW	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。
--	--	--	--	--	----------------

10.2.25 MCPWM_PRT

无写保护的寄存器

表 10-26 MCPWM_PRT 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_PRT	0x0	0x74	[31:16]	NA	未使用
			[15:0]	RW	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。

10.2.26 MCPWM_CNT

无写保护的寄存器

表 10-27 MCPWM_CNT 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_CNT	0x0	0x78	[31:16]	NA	未使用
			[15:0]	RW	MCPWM 实际运行系统中计数器的值。

10.2.27 MCPWM_DTH00

写保护的寄存器

表 10-28 MCPWM_DTH00 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_DTH00	0x0	0x80	[31:10]	NA	未使用
			[9:0]	RW	MCPWM CH0 N 通道死区宽度控制寄存器， 10bit 无符号数

10.2.28 MCPWM_DTH01

写保护的寄存器

表 10-29 MCPWM_DTH01 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_DTH01	0x0	0x84	[31:10]	NA	未使用
			[9:0]	RW	MCPWM CH0 P 通道死区宽度控制寄存器， 10bit 无符号数



10.2.29 MCPWM_DTH10

写保护的寄存器

表 10-30 MCPWM_DTH10 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_DTH10	0x0	0x88	[31:10]	NA	未使用
			[9:0]	RW	MCPWM CH1 N 通道死区宽度控制寄存器，10bit 无符号数

10.2.30 MCPWM_DTH11

写保护的寄存器

表 10-31 MCPWM_DTH11 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_DTH11	0x0	0x8C	[31:10]	NA	未使用
			[9:0]	RW	MCPWM CH1 P 通道死区宽度控制寄存器，10bit 无符号数

10.2.31 MCPWM_DTH20

写保护的寄存器

表 10-32 MCPWM_DTH20 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_DTH20	0x0	0x90	[31:10]	NA	未使用
			[9:0]	RW	MCPWM CH2 N 通道死区宽度控制寄存器，10bit 无符号数

10.2.32 MCPWM_DTH21

写保护的寄存器

表 10-33 MCPWM_DTH21 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_DTH21	0x0	0x94	[31:10]	NA	未使用
			[9:0]	RW	MCPWM CH2 P 通道死区宽度控制寄存器，10bit 无符号数



10.2.33 MCPWM_DTH30

写保护的寄存器

表 10-34 MCPWM_DTH30 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_DTH30	0x0	0x98	[31:10]	NA	未使用
			[9:0]	RW	MCPWM CH3 N 通道死区宽度控制寄存器，10bit 无符号数

10.2.34 MCPWM_DTH01

写保护的寄存器

表 10-35 MCPWM_DTH31 配置寄存器

名称	复位值	偏移	位置	权限	说明
MCPWM_DTH31	0x0	0x9C	[31:10]	NA	未使用
			[9:0]	RW	MCPWM CH3 P 通道死区宽度控制寄存器，10bit 无符号数



11 UART

11.1 概述

UART 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

11.2 功能说明

11.2.1 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。

11.2.2 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。

11.2.3 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=主时钟/ (256*DIVH+DIVL+1)

系统主时钟最大为 96MHz 时,波特率最低为 96MHz/ (256*255+255+1) =1464Hz。



11.3 寄存器

11.3.1 地址分配

UART0 与 UART1 实现完全相同。

UART0 基地址 0x40003900。

UART1 基地址 0x40003A00。

表 11-1 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

11.3.2 UARTx 控制寄存器 UARTx_CTRL

表 11-2 UARTx 控制寄存器 UARTx_CTRL

名称	复位值	偏移	位置	权限	说明
UARTx_CTRL	0x0	0x00	[31:8]	NA	未使用
			[7]	RW	使能 IO 翻转, 0:禁用, 1:使能
			[6]	RW	Multi-drop Master 模式时, 第 9 个数据位值
			[5]	RW	使能 Multi-drop, 0:禁用, 1:使能
			[4]	RW	使能校验, 0:禁用, 1:使能
			[3]	RW	奇偶校验, 0:EVEN 1:ODD
			[2]	RW	先发送的比特, 0:LSB, 1:MSB
			[1]	RW	停止位长度, 0:1bit, 1:2bit
[0]	RW	数据长度, 0:8bit, 1:7bit			

11.3.3 UARTx 波特率设置高字节寄存器 UARTx_DIVH

表 11-3 UARTx 波特率设置高字节寄存器 UARTx_DIVH

名称	复位值	偏移	位置	权限	说明
UARTx_DIVH	0x0	0x04	[31:8]	NA	未使用
			[7:0]	RW	波特率设置高字节 BAUDRATE = 主时钟 / (1 + DIVL + 256 * DIVH)



11.3.4 UARTx 波特率设置低字节寄存器 UARTx_DIVL

表 11-4UARTx 波特率设置低字节寄存器 UARTx_DIVL

名称	复位值	偏移	位置	权限	说明
UARTx_DIVL	0x0	0x08	[31:8]	NA	未使用
			[7:0]	RW	波特率设置低字节 BAUDRATE = 模块时钟 / (1 + DIVL + 256 * DIVH)

11.3.5 UARTx 收发缓冲寄存器 UARTx_BUFF

表 11-5UARTx 收发缓冲寄存器 UARTx_BUFF

名称	复位值	偏移	位置	权限	说明
UARTx_BUFF	0x0	0x0C	[31:8]	NA	未使用
			[7:0]	RW	写:发送数据缓存读:接收数据寄存器

UART 的 Tx_buffer 和 Rx_buffer 共享地址 0x0C 地址。其中, Tx_buffer 是只写的, Rx_buffer 是只读的。因此读访问 UARTx_BUFF 是访问 UARTx_RX_BUFF, 写访问 UARTx_BUFF 是访问 UARTx_TX_BUFF。

11.3.6 UARTx 地址匹配寄存器 UARTx_ADR

表 11-6UARTx 地址匹配寄存器 UARTx_ADR

名称	复位值	偏移	位置	权限	说明
UARTx_ADR	0x0	0x10	[31:8]	NA	未使用
			[7:0]	RW	用作 485 通信时的匹配地址

11.3.7 UARTx 状态寄存器 UARTx_STT

表 11-7UARTx 状态寄存器 UARTx_STT

名称	复位值	偏移	位置	权限	说明
UARTx_STT	0x0	0x14	[31:3]	NA	未使用
			[2]	RW	Multi-drop 模式下, 地址匹配上
			[1]	RW	发送缓存空
			[0]	RW	发送完成 (此时发送缓存如不为空, 则可以继续发送缓存中的数据)

11.3.8 UARTx 中断使能寄存器 UARTx_IE

表 11-8UARTx 中断使能寄存器 UARTx_IE

名称	复位值	偏移	位置	权限	说明
UARTx_IE	0x0	0x18	[31:5]	NA	未使用
			[4]	RW	校验错误中断使能



			[3]	RW	停止位错误中断使能
			[2]	RW	发送缓冲区空中断使能
			[1]	RW	接收完成中断使能
			[0]	RW	发送完成中断使能

11.3.9 UARTx 中断标志寄存器 UARTx_IF

表 11-9 UARTx 中断标志寄存器 UARTx_IF

名称	复位值	偏移	位置	权限	说明
UARTx_IF	0x0	0x1C	[31:5]	NA	未使用
			[4]	RW	校验错误中断标志
			[3]	RW	停止位错误中断标志
			[2]	RW	发送缓冲区空中断标志
			[1]	RW	接收完成中断标志
			[0]	RW	发送完成中断标志

11.4 应用指南

SYS_CLK_DIV2 可对 UART 模块时钟进行分频，以系统主时钟为 96MHz 为例，如果需要设置为较低的波特率，可以使用 SYS_CLK_DIV2 和 UART_DIVL、UART_DIVH 联合分频

$$\text{UART 模块工作时钟} = 96\text{MHz}/(1+\text{SYS_CLK_DIV2})$$

$$\text{UART 波特率} = \text{UART 模块工作时钟}/(1+\text{DIVL}+256*\text{DIVH})$$



12 信号协处理器模块

12.1 概述

协处理器模块主要完成除法和开方两种运算，被除数、除数、商、余数位宽均为 32 位，被开方数为 32 位，平方根为 16 位，另外有 1 位除 0 错误指示。

除法 32 个总线周期（96MHz），经过 32 次移位加减完成。

开方 8 个总线周期（96MHz）完成。

12.1.1 功能框图

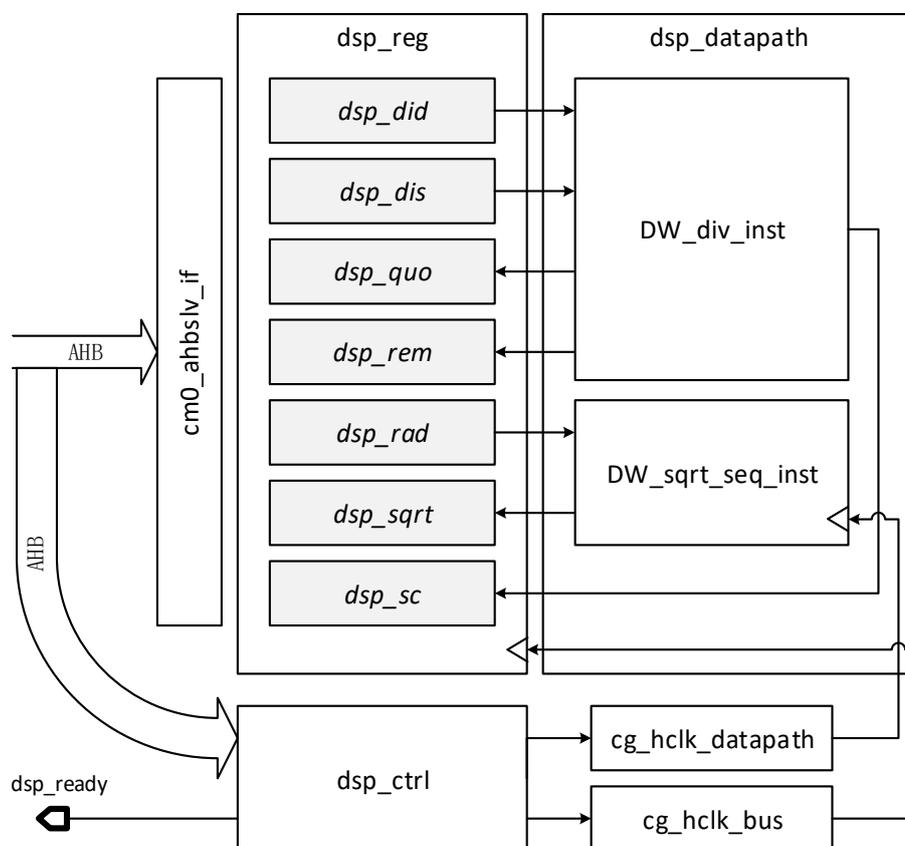


图 12-1 DSP 模块功能框图

12.1.2 特点

- 32bit 除法器，32 周期完成一次除法，支持除 0 检测
- 32bit 被开方数，平方根 16bit



12.2 寄存器

12.2.1 地址分配

信号协处理器模块在芯片中的基地址是 0x4000_3800。

表 12-1 DSP 寄存器列表

名称	偏移	说明
DSP_DID	0x00	DSP 除法器被除数寄存器
DSP_DIS	0x04	DSP 除法器除数寄存器
DSP_QUO	0x08	DSP 除法器商寄存器
DSP_REM	0x0C	DSP 除法器余数寄存器
DSP_RAD	0x10	DSP 开方器被开方数寄存器
DSP_SQRT	0x14	DSP 开方器平方根寄存器
DSP_SC	0x18	DSP 控制状态寄存器

12.2.2 除法器

12.2.2.1 被除数寄存器 DSP_DID

表 12-2 被除数寄存器 DSP_DID

名称	复位值	偏移	位置	权限	说明
DSP_DID	0x0	0x00	[31:0]	RW	DSP 除法器 32bit 被除数

12.2.2.2 除数寄存器 DSP_DIS

表 12-3 除数寄存器 DSP_DIS

名称	复位值	偏移	位置	权限	说明
DSP_DIS	0x0	0x04	[31:0]	RW	DSP 除法器 32bit 除数

除数不应为 0，否则会发生除 0 错误。

写入除数可以触发一次除法开始进行。

12.2.2.3 商寄存器 DSP_QUO

表 12-4 商寄存器 DSP_QUO

名称	复位值	偏移	位置	权限	说明
DSP_QUO	0x0	0x08	[31:0]	R	DSP 除法器 32bit 商



12.2.2.4 余数寄存器 DSP_REM

表 12-5 余数寄存器 DSP_REM

名称	复位值	偏移	位置	权限	说明
DSP_REM	0x0	0x0C	[31:0]	R	DSP 除法器 32bit 余数

12.2.3 开方器

12.2.3.1 被开方数寄存器 DSP_RAD

表 12-6 被开放数寄存器 DSP_RAD

名称	复位值	偏移	位置	权限	说明
DSP_RAD	0x0	0x10	[31:0]	RW	DSP 开放器 32bit 被开方数

32bit 被开方数被当做无符号数处理。

写入被开方数可以触发一次开放运算开始进行。

12.2.3.2 平方根寄存器 DSP_SQRT

表 12-7 平方根寄存器 DSP_SQRT

名称	复位值	偏移	位置	权限	说明
DSP_SQRT	0x0	0x14	[31:16]	NA	未使用
			[15:0]	R	DSP 开方器输出的平方根

12.2.3.3 控制状态寄存器 DSP_SC

表 12-8 DSP 控制状态寄存器 DSP_SC

名称	复位值	偏移	位置	权限	说明
DSP_SC	0x0	0x18	[31:5]	NA	未使用
			[4]	R	除 0 错误标志位, 高有效
			[3:1]	NA	未使用
			[0]	RW	DSP 使能, 高有效



12.3 实现说明

12.3.1 时钟门控时序

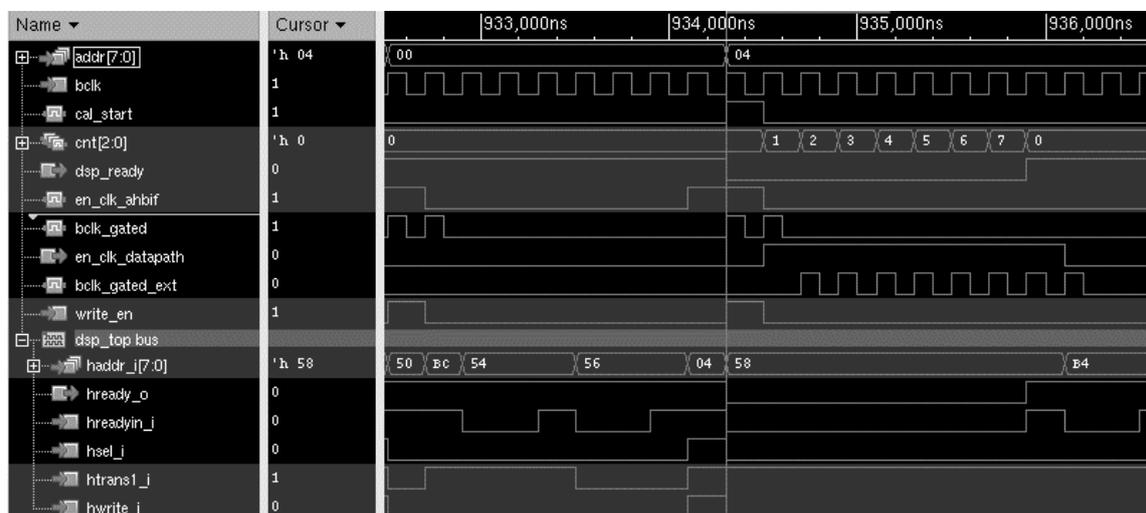


图 12-2 时钟门控

其中 `bclk` 为总线时钟, `bclk_gated` 为寄存器访问的门控时钟; `bclk_gated_ext` 为开方运算所需的门控时钟。

除法 32 个总线周期 (96MHz), 经过 32 次移位加减完成。写入除数寄存器触发一次新的除法计算, 32 个周期后可以从商、余数、除 0 错误标志寄存器读回计算结果。除法计算过程中 `dsp` 模块会拉低 `ahb` 总线 `ready` 信号, 中断整个总线上的所有操作, `cpu` 如果在此阶段内发起总线操作也会等待总线响应。

开方 8 个总线周期 (96MHz) 完成。写入被开方数寄存器触发一次新的开方计算, 8 个周期后可以从平方根寄存器读回计算结果。开方计算过程中 `dsp` 模块会拉低 `ahb` 总线 `ready` 信号, 中断整个总线上的所有操作, `cpu` 如果在此阶段内发起总线操作也会等待总线响应。



13 I2C

13.1 概述

本芯片的 SPI 模块和 I2C 模块，共享 FIFO。若一个使用了 FIFO，另外一个就只能单字节传输。本芯片的 I2C 模块可工作在主模式、从模式或主从模式。此时，I2C 对应为主设备、从设备或主从设备。I2C 总线网络是支持多主多从的，所以，I2C 模块工作在何种模式下，应考虑实际系统的组网情况。

当模块工作在从模式时，I2C 模块仅是一个从设备，无需考虑 I2C 总线网络是否多主设备。从模式下，I2C 模块只需要实现的功能为：监听总线状态--捕捉 START 信号、接收数据或发送数据。

当模块工作在主模式时，I2C 模块仅是一个主设备，此时 I2C 总线网络只有一个主设备。在主模式下，I2C 模块只需要实现的功能为：产生 START 信号和 STOP 信号、接收数据或发送数据。

当模块工作在主从模式时，I2C 模块是一个主从设备，此时 I2C 总线网络存在多个主设备。主从模式下，I2C 模块除开实现主模式所有功能和从模式所有功能外，还需要处理多主机争抢 I2C 总线资源的问题。

本芯片的 I2C 模块主要实现如下功能：

- 支持主/从模式，接收/发送操作
- 支持中断或轮询两种方式
- 主模式时钟频率：50K，100K，400K
- 支持多主机时钟同步
- 支持多主机模式判断
- 7-Bit 寻址
- 硬件支持自动地址比较(仅在 7-Bit 地址和从模式下)
- 主从模式下突发传输(Burst Transfer Mode)
- 支持的中断触发条件包括：总线错误、停止、NACK、硬件地址匹配、传输完成

I2C 模块整体结构框图如下：



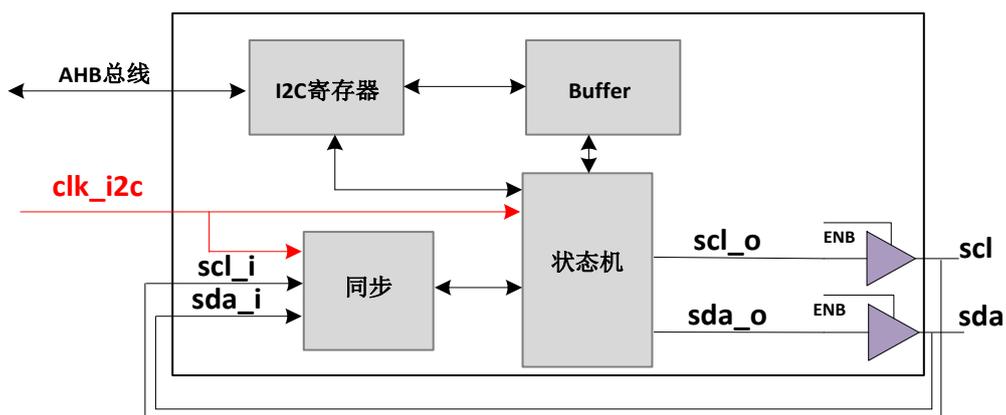


图 13-1 I2C 模块结构框图

I2C 模块的时钟信号来自于系统时钟。因为 I2C 总线传输速度比较低，我们须在系统端对输送给 I2C 模式使用的 I2C 时钟进行预先分频（见时钟模块相关寄存器）。SCL 时钟频率和 I2C 时钟频率关系，如下：

当 I2C 模块配置成主模式时，I2C 时钟频率是 SCL 时钟频率的 17 倍。SCL 的高电平时间为 8 个 I2C 时钟周期，低电平时间为 9 个 I2C 时钟周期。

当 I2C 模块配置成从模式时，I2C 时钟频率是 SCL 时钟频率的至少 17 倍。两者相差越大，对于整个系统而言，处理时间越宽裕。

为了减少外界干扰对 I2C 模块的影响，我们使用同步模块对 SCL 和 SDA 信号进行采样滤波处理，可以将低于 3 个 I2C 时钟周期的短脉冲过滤掉。

13.2 寄存器说明

13.2.1 地址分配

I2C 模块寄存器的基地址是 0x4000_30C0 寄存器列表。

表 13-1 I2C 模块控制寄存器列表

名称	偏移地址	说明
I2C0_ADDR	0x00	I2C 模块硬件地址寄存器
I2C0_CFG	0x04	I2C 配置寄存器
I2C0_SCR	0x08	I2C 状态和控制寄存器
I2C0_DATA	0x0C	I2C 数据寄存器
I2C0_MSCR	0x10	I2C 主机状态和控制寄存器
I2C0_BUF_CTR	0x14	I2C Buffer 控制寄存器
I2C0_BUF_ADDR	0x20	I2C Buffer 地址寄存器



13.2.2 I2C_ADDR

表 13-2 I2C0_ADDR 地址寄存器

名称	复位值	偏移	位置	权限	说明
I2C0_ADDR	0x0	0x00	[31:8]	NA	未使用
			[7]	RW	硬件地址自动比较使能开关。仅支持 7-bit 地址，且仅从模式有效。 置 1, 接收到的 7-bit 地址和本寄存器中[6:0]的内容进行比较。地址比较成功，产生中断，在中断服务程序中，I2C_SCR 寄存器的 Address 位须清零。地址比较失败，则是非针对本 I2C 模块的访问，无需软件处理，硬件直接抛弃本次操作。 置 0, 不比较。只要接收到地址数据，交给软件处理。
			[6:0]	RW	存储地址。I2C 为主模式且是 Burst 传输时，存储将要发送的地址；I2C 为从模式时，存储自身地址。

13.2.3 I2C0_CFG

表 13-3 I2C0_CFG 配置寄存器

名称	复位值	偏移	位置	权限	说明
I2C0_CFG	0x0	0x04	[31:8]	NA	未使用
			[7]	RW	I2C 中断使能总开关。1, 使能; 0, 关闭。
			[6]	NA	未使用
			[5]	RW	I2C 总线错误中断使能开关。1, 使能; 0, 关闭。仅在主模式下使用，控制检测到总线错误时是否产生中断。总线错误通常是一个错误的启动或者停止条件，在主机操作的情况下这是一个重要的中断。当产生总线错误时，所有从机设备要根据该信号重新设定总线接口并同步。但，主机模式进行数据传输过程中，硬件检测到总线错误时，主设备将释放总线成为空闲状态。
			[4]	RW	检测到总线产生停止信号，是否产生中断。置 1, 允许; 置 0, 不允许。主模式和从模式均可使用。
			[3:2]	NA	未使用
			[1]	RW	主机模式。置 1, I2C 模块使能主模式功能。置 0, 屏蔽 I2C 模块主模式功能。
			[0]	RW	从机模式。置 1, I2C 模块使能从模式功能。置 0, 屏蔽 I2C 模块从模式功能。



13.2.4 I2C0_SCR

表 13-4 I2C0_SCR 状态和控制寄存器

名称	复位值	偏移	位置	权限	说明
I2C0_SCR	0x0	0x08	[31:8]	NA	未使用
			[7]	RW	检测到一个错误的启动或者停止信号。仅主机模式使用，软件写 0 清除。
			[6]	RW	丢失仲裁。主机模式使用。主机丢失仲裁之后这位即被置位，无中断产生。在字节传输完成后，这个状态可以被检测出来。任何初始检测将会自动清除该位。
			[5]	RW	检测到停止信号。主模式和从模式均可使用。软件写 0 清除。
			[4]	RW	1，接收完字节之后发送 ACK；0，接收完字节之后发送 NACK。主模式和从模式均可使用。 在非 Burst 传输模式下，在字节传输完成事件之后这位被硬件自动清除。 在 Burst 传输模式下，在接收模式下开始传输数据之前这位必须是 1，在接下来的字节传输完成事件之后这位被硬件自动清除。
			[3]	RW	发送或者接收字节是一个地址数据。主模式和从模式均可使用。软件写 0 清除。
			[2]	RW	1，发送模式；0，接收模式。主模式和从模式均可使用。
			[1]	RW	1，传输字节后接收端反馈是 NACK；0，传输字节后接收端反馈是 ACK。主模式和从模式均可使用。
[0]	RW	1，传输完成信号。软件写 0 清除。 发送模式：预定义字节数的数据接收成功发送成功且 ACK 或者 NACK 已经收到。 接收模式：预定义字节数的数据接收成功。			

13.2.5 I2C0_DATA

表 13-5 I2C 数据寄存器

名称	复位值	偏移	位置	权限	说明
I2C0_DATA	0x0	0x0C	[31:8]	NA	未使用
			[7:0]	RW	数据寄存器。 非 Burst 模式： 1. 主模式，发送模式。填入地址，操作 I2C_MSCR 寄存器的 start/restart 位，触发地址的发送。 2. 主模式/从模式，发送模式。填入发送的数据，



					<p>写操作 I2C_SCR 寄存器的 transmit 位，触发数据的发送。</p> <p>3. 主模式/从模式，接收数据。字节接收完成，读取该寄存器获得接收数据，写操作 I2C_SCR 寄存器，继续后续流程。</p> <p>Burst 模式：</p> <p>1.主模式，发送模式。填入地址到 I2C_ADDR 寄存器，填入数据到 I2C_DR 寄存器，操作 I2C_MSCR 寄存器的 start/restart 位，触发地址的发送。</p> <p>2. 主模式/从模式，发送模式。填入发送的数据，写操作 I2C_SCR 寄存器，触发数据的继续发送。</p> <p>3. 主模式/从模式，接收数据。字节接收完成，读取该寄存器获得接收数据，写操作 I2C_SCR 寄存器，继续后续流程。</p>
--	--	--	--	--	---

13.2.6 I2C0_MSCR

表 13-6 I2C 主机状态和控制寄存器

名称	复位值	偏移	位置	权限	说明
I2C0_MSCR	0x0	0x10	[31:4]	NA	未使用
			[3]	R	闲忙标识信号。检测到 START 信号，该位被硬件自动置 1；当检测到 STOP 信号，该位被硬件自动清 0。
			[2]	R	主从标识信号。当本 I2C 模块发出 START 信号，该位硬件自动置 1；当检测到 STOP 信号，该位硬件自动清 0。
			[1]	RW	产生一个 RESTART 信号。产生结束后，硬件自动清 0。
			[0]	RW	产生一个 START 信号。产生结束后，硬件自动清 0。

13.2.7 I2C0_BUF_CTRL

表 13-7 I2C0_BUF_CTRL 控制寄存器

名称	复位值	偏移	位置	权限	说明
I2C0_BUF_CTRL	0x0	0x14	[31:8]	NA	未使用
			[7]	RW	Burst 模式下，发送方对接收方返回的 NACK 的处理方式。1，允许在发送方产生 NACK 中断，此时中断来临时，会执行拉底 SCL 总线的操作；0，在发送方不产生 NACK 中断，此时若发送方是主设备--直接 STOP 返回，若发送方是从设备--直接进入 IDLE。
			[6]	RW	Burst 模式下，是否允许 HwAddr 中断产生。1，允许；0，不允许。在从机突发模式下接收地址，



					该位和 I2C_ADDR[7]配合使用。
			[5]	RW	Burst 模式开关。1, 开启 Burst 模式; 0, 关闭 Burst 模式。
			[4:0]	RW	Burst 模式开启下, 一次 Burst 传输字节数。实际传输数等于[4:0] + 1。

13.2.8 I2C0_BUF_ADDR

表 13-8 I2C0 buffer 地址寄存器

名称	复位值	偏移	位置	权限	说明
I2C0_BUF_ADDR	0x0	0x18	[31:5]	NA	未使用
			[4:0]	RW	Buffer 地址指针。指向下一个 I2C_DR 操作位置。一般在使用前, 软件把地址写 0, 每操作一次 I2C_DR, 该寄存器自动累加 1.

13.3 应用指南

I2C 模块基本的一次传输, 如下举例:

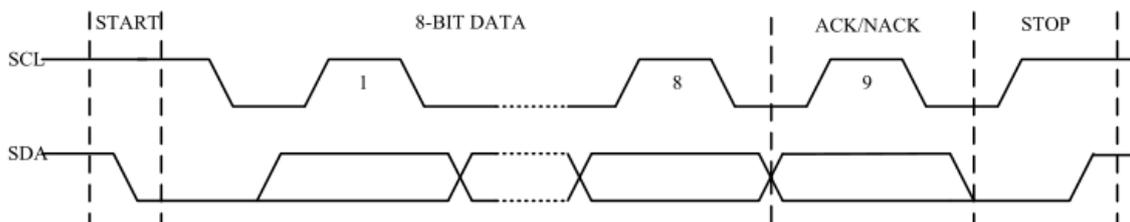


图 13-2 I2C 模块基本的一次传输时序

主设备, 发起 START 信号; 发送第一个 Byte 数据 (该数据一般是 I2C 从设备地址和读写控制信号); 若有从设备地址匹配, 且从设备可以完成数据交换, 则回应 ACK。从设备无法完成数据交换, 则回应 NACK; 主设备发送 STOP 信号, 表明此次传输完成 (中间可根据软件协议传输 N 个字节的数据)。I2C 执行读/写, 按照单字节发送还是 Burst 模式发送, 每个发送字节的含义, 均由上层软件协议规定, 同 I2C 模块本身没有关系。

从设备, Burst 接收模式:

```

//配置 I2C_CFG 寄存器
I2C0_CFG = 0x91; //enable stop ie & i2c interrupt enable, slave mode enabled
//配置系统中断使能
_enable_irq();
    
```



```

NVIC_Enable(I2C_IRQn);
//配置硬件地址比较

I2C0_ADDR = 0x99; //enable hardware address compare, slave address = 0x19
//配置 Buffer 地址和参数

I2C0_BUF_CTRL = 0x3f; //burst mode, buffer size = 32bytes

I2C0_BUF_ADDR= 0x00; //buffer start point is 0x0
//使能 I2C Slave Burst 发送

I2C0_SCR = 0x10; //set ack for slave that responds master data requirement.

```

使能 I2C 从设备的硬件地址比较；配置好从设备 I2C 地址；开启 I2C 中断源和中断在系统级别的使能；配置好 Buffer 参数（本次 Burst 长度为 32 字节），使能 Burst 传输模式。完成上述配置后，MCU 可以执行其它任务。I2C 模块就会根据总线发送的地址，自行判断是否相应；同时，只有完成 32 字节数据的接收操作后，I2C 才会产生完成中断。

主设备，Burst 发送模式：

```

//配置 I2C_CFG 寄存器

I2C0_CFG = 0x92; //enable stop ie & i2c interrupt enable, master mode enabled
//配置系统中断使能

__enable_irq();
NVIC_Enable(I2C_IRQn);
//配置硬件地址比较

I2C0_ADDR = 0x19; // store target slave address = 0x19
//配置 Buffer 地址和参数。

I2C0_BUF_CTRL = 0x3f; //burst mode, buffer size = 32bytes

I2C0_BUF_ADDR= 0x00; //buffer start point is 0x0
//加载数据至 Buffer

I2C0_DATA = data; //load transmit data to buffer
//传输数据。

I2C0_MSCR = 0x01;

```

使能 I2C 主模式；配置好目标从设备 I2C 地址；开启 I2C 中断源和中断在系统级别的使能；配置好 Buffer 参数（本次 Burst 长度为 32 字节），使能 Burst 传输模式；把将要发送的数据，逐一写入 Buffer 中。完成上述配置后，触发 I2C 发送，MCU 可以执行其它任务。完成 32 字节数据的发送操作后，I2C 会产生完成中断。





14 SPI

14.1 概述

本芯片的 SPI 模块和 I2C 模块，共享 FIFO。若一个使用了 FIFO，另外一个就只能单字节传输。SPI 模块主要实现如下功能：

- 主模式和从模式
- SPI 时钟信号极性和相位可配
- 32-Byte 的 FIFO，用于发送和接收数据

SPI 总线由四根信号线构成，分别是 SPI_CLK，SPI_DI，SPI_DO 和 SPI_SS。

- 在主模式下，SPI_CLK 为时钟输出，SPI_DI 为数据输入，SPI_DO 为数据输出，SPI_SS 为片选输出
- 在从模式下，SPI_CLK 为时钟输入，SPI_DI 为数据输入，SPI_DO 为数据输出，SPI_SS 为片选输入

一般应用，对方的数据输出接本模块的 DI；对方的数据输入接本模块的 DO；SS 则看谁是主则是输出，从则是输入；CLK 接法同 SS。在一种应用中，本模块只能固定为一种工作模式--主模式或者从模式。

SPI 总体框图，如下：

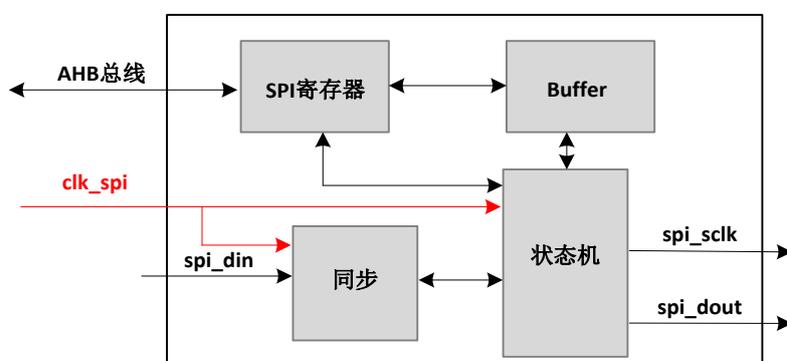


图 14-1 SPI 模块结构框图

注意，clk_spi 和 spi_sclk 之间为 8 倍关系，前者是后者频率的 8 倍。

14.2 寄存器说明

14.2.1 地址分配

SPI 模块寄存器的基地址是 0x4000_3080 寄存器列表。



表 14-1 SPI 模块控制寄存器列表

名称	偏移地址	说明
SPIO_SHIFTER	0x00	SPI 移位寄存器
SPIO_DATA	0x04	SPI 数据寄存器
SPIO_CR0	0x08	SPI 控制寄存器 0，控制主/从功能。
SPIO_CR1	0x0C	SPI 控制寄存器 1
SPIO_CR2	0x10	SPI 控制寄存器 2
SPIO_BUF_ADDR	0x14	SPI Buffer 地址寄存器

14.2.2 SPIO_SHIFTER

表 14-2 SPIO_SHIFTER 移位寄存器

名称	复位值	偏移	位置	权限	说明
SPIO_SHIFTER	0x0	0x00	[31:8]	NA	未使用
			[7:0]	R	SPI 移位寄存器。从 Tx Buffer 加载数据到该寄存器中，发送出去；接收 SPI 串行输入数据，转存入 Rx Buffer 中。

14.2.3 SPIO_DATA

表 14-3 SPIO_DATA 数据寄存器

名称	复位值	偏移	位置	权限	说明
SPIO_DATA	0x0	0x04	[31:8]	NA	未使用
			[7:0]	RW	SPI 数据寄存器。发送模式，可通过写该寄存器，把将发送的数据写入 Tx Buffer；接收模式，可通过读该寄存器，获得 Rx Buffer 中接收到的数据。

14.2.4 SPIO_CR0

表 14-4 SPIO_CR0 控制寄存器 0

名称	复位值	偏移	位置	权限	说明
SPIO_CR0	0x0	0x08	[31:8]	NA	未使用
			[7]	RW	SPI 中断使能信号。1，使能；0，关闭。
			[6]	NA	未使用
			[5]	RW	SPI 片选信号来源。1，输入 spi_ss_n 控制；0，一直被选中。



			[4]	RW	SPI 主从模式选择信号。1，主模式；0，从模式。
			[3]	RW	数据采样，更新选择。 0，上升沿采样数据，下降沿数据更新 1，上升沿数据更新，下降沿采样数据
			[2]	RW	时钟极性选择。 0，非反转，时钟默认为低 1，反转，时钟默认为高
			[1]	RW	数据大小端传输格式选择。 0，按大端发送，先发高位数据 1，按小端发送，先发低位数据
			[0]	RW	SPI 模块使能信号。1，使能；0，关闭。

SPI_CR0[2] 控制了 SPI 时钟信号在默认情况下的电平状态。SPI_CR0[2]为 0 时，默认时钟电平为低电平；SPI_CR0[2]为 1 时，默认电平为高电平。SPI_CR0[3] 控制了 SPI 数据的发送/接收时刻。SPI_CR0[3]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 SPI_CR0[3]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。

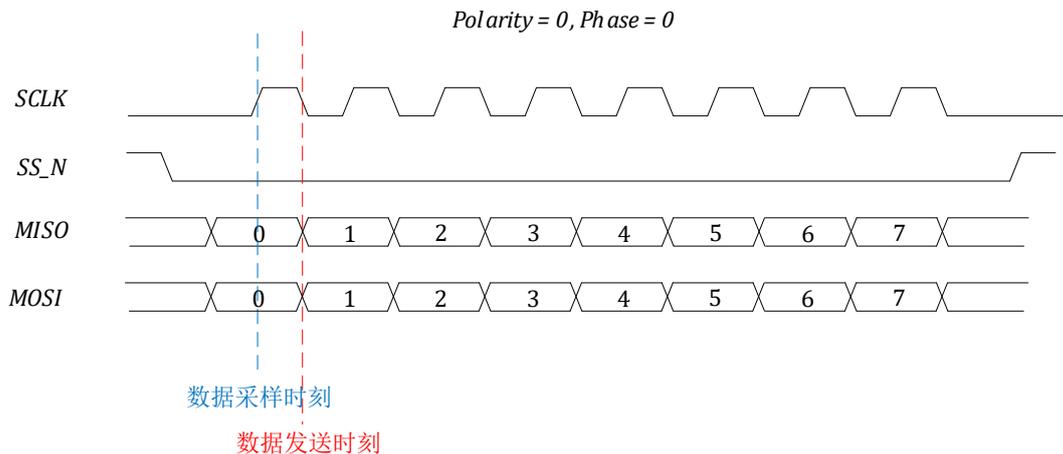


图 14-2 SPI 通讯信号极性相位(Polarity=0, Phase=0)

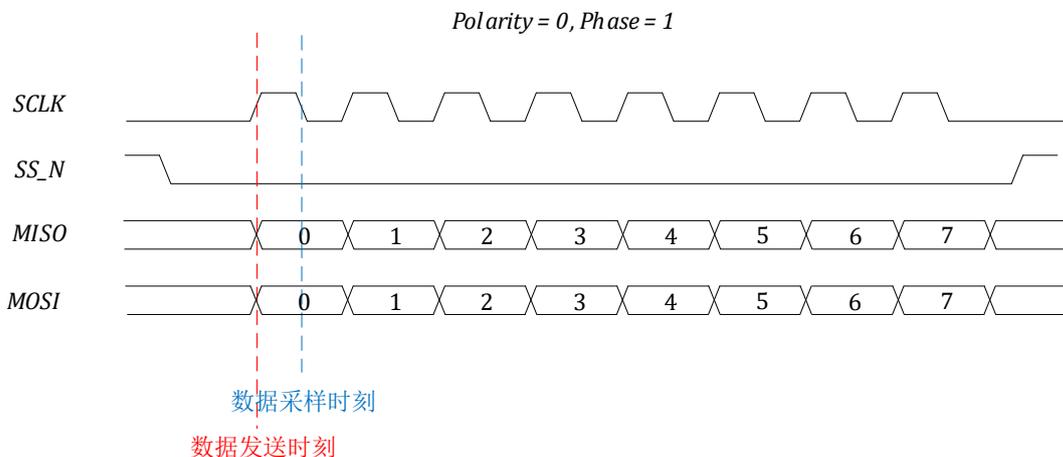


图 14-3 SPI 通讯信号极性相位(Polarity=0, Phase=1)

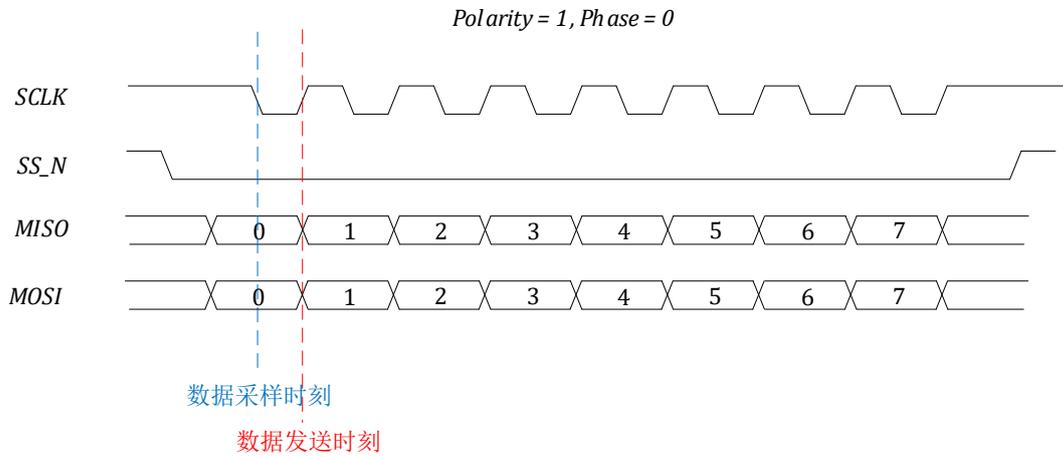


图 14-4 SPI 通讯信号极性相位(Polarity=1, Phase=0)

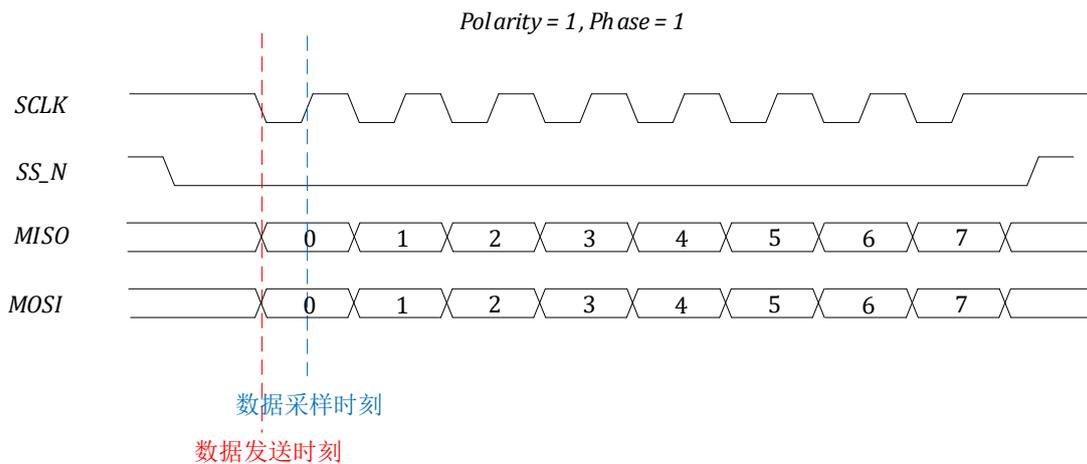


图 14-5 SPI 通讯信号极性相位(Polarity=1, Phase=1)

14.2.5 SPI0_CR1

表 14-5 SPI0_CR1 控制寄存器 1

名称	复位值	偏移	位置	权限	说明
SPI0_CR1	0x0	0x0C	[31:1]	NA	未使用
			[0]	RW	接收 Rx Buffer 空满信号。 0, Rx Buffer 没满 1, Rx Buffer 已满, 产生相应中断。读 Rx Buffer



					将自动清除该位。
--	--	--	--	--	----------

14.2.6 SPI0_CR2

表 14-6 SPI0_CR2 控制寄存器 2

名称	复位值	偏移	位置	权限	说明
SPI0_CR2	0x0	0x10	[31:5]	NA	未使用
			[4:0]	RW	一次数据传输的字节数。实际传输数等于[4:0] + 1。

14.2.7 SPI0_BUF_ADDR

表 14-7 SPI0_BUF_ADDR 地址寄存器

名称	复位值	偏移	位置	权限	说明
SPI0_BUF_ADDR	0x0	0x14	[31:5]	NA	未使用
			[4:0]	RW	Buffer 地址指针。指向下一个 SPI_DATA 操作位置。一般在使用前，软件把地址写 0，每操作一次 SPI_DATA，该寄存器自动累加 1。

14.3 应用说明

SPI 模块基本的一次传输，如下举例。注意，从模式下，推荐将片选使能连接到 GPIO。

主模式，数据传输的发起者。选择传输模式（CPOL 和 CPHA），选择好 SPI 的传输频率。按照单字节发送还是 Burst 模式发送，每个发送字节的含义，均由上层软件协议规定，同 SPI 模块本身没有关系。

主模式：

```

_enable_irq();           //登记 SPI 中断

NVIC_EnableIRQ (SPI0_IRQn);

GPIO0_PIE = 0x0000; //GPIO 复用配置

GPIO0_POE = 0xC000;

GPIO1_PIE = 0x0001;

GPIO1_POE = 0x0002;

GPIO0_FFEDC = 0x5500;
```



```

GPIO1_F3210 = 0x0005;
//配置 SPI 传输相关寄存器

SPI0_CR1=0x00;

SPI0_BUF_ADDR=0x00;

SPI0_CR2=0x1f; //发送数据量

SPI0_CR0=0x90; //使能 SPI 中断

SPI0_CR0=0x91;

//写入发送数据到 SPI FIFO，根据发送数量写多少次该寄存器

SPI0_DATA = DATA; //写完毕，将自动开始传输

```

SPI 本次传输完毕后，自动产生完成中断。

从模式下，一旦时钟上有毛刺（超过 30ns 宽度的毛刺），可能导致整个传输错误。因为，片选信号不会清除 SPI 内部状态机，会出现死锁状态。推荐从模式下，片选信号接 GPIO。通过 GPIO 中断处理程序强制对 SPI 执行软复位。

```

__enable_irq(); //登记 SPI 中断

NVIC_EnableIRQ (SPI0_IRQn);

GPIO0_PIE = 0x4000; //GPIO 复用配置

GPIO0_POE = 0x8000;

GPIO1_PIE = 0x0003;

GPIO1_POE = 0x0000;

GPIO0_FFEDC = 0x5500;

GPIO1_F3210 = 0x0055;

SPI0_CR1=0x00; //配置 SPI 传输相关寄存器，片选永远有效

SPI0_BUF_ADDR=0x00;

SPI0_CR2=0x1f;

```



```
SPI0_CR0=0x86;
```

```
SPI0_CR0=0x87;
```

```
//写入发送数据到 SPI FIFO，根据发送数量写多少次该寄存器
```

```
SPI0_DATA = DATA; //写完毕，若主设备开始传输，从将自动开始传输
```

SPI 本次传输完毕后，自动产生完成中断。



15 版本历史

表 15-1 文档版本历史

时间	版本号	作者	说明
2017.05.10	0.1	张威龙、邓廷、钟书鹏	初始版本
2017.05.15	0.2	钟书鹏	增加 3.10 模拟寄存器说明
2017.06.01	0.3	邓廷	增加 5. Flash 模块说明 增加 12. I2C 模块说明 增加 13. SPI 模块说明
2017.06.12	0.4	李鹏	更新 MCPWM 说明、Encoder 说明
2017.07.29	0.5	徐蓉	格式更新
2017.08.18	0.6	张威龙	Minor revision
2017.08.20	0.8	张威龙	更新寄存器定义
2017.08.28	0.91	张威龙	增加 HALL 模块部分
2017.08.31	0.92	张威龙	修改 UART、UTIMER 寄存器描述
2017.08.31	0.93	刘虎、张威龙	增加 GPIO 引脚功能复用说明
2017.09.02	0.94	邓廷	修订 SPI 寄存器描述错误
2017.09.03	0.95	张威龙	更新时钟复位寄存器描述，更改 UTIMER/UART/SYS 寄存器名，移除 GPIO 引进复用说明
2017.09.05	0.96	钟书鹏	晚上运算放大器部分的修改
2017.09.07	0.97	张威龙	为模拟寄存器表增加寄存器名
2017.09.07	0.98	钟书鹏	在各个模拟电路模块里，增加与该模块相关寄存器的地址和说明。同时将 ADC 部分说明和 ADC 接口部分的说明整合在一起
2017.09.07	0.99	张威龙	增加寄存器偏移，ADC 接口模块修改 ADC_DAT 寄存器名为 ADCx_DAT；ADCx_DAT0 描述由通道 0 采样数据修改为 ADC 第 0 次采样数据；ADCx_CHN0 描述由控制第 0-3 通道修改为控制第 0-3 次采样；增加对于模拟信号通道实际含义的描述，与模拟寄存器表相同。 统一了表格中各列名称
2017.10.29	1.00	张威龙	修改 ADC、UTimer 模块系统功能框图，增加 Encoder 计数模式示意图，修改 UTimer、Encoder 寄存器描述 整合 ADC 接口模块与模拟寄存器表，将模拟寄存器 ADC 相关描述移动至 ADC 接口模块中，去除模拟寄存器表中硬件相关而用户不需使用的接口。
2017.11.03	1.01	邓廷	修改 MCPWM 模块，增加相应功能框图，修正寄存器描述。
2017.11.12	1.01	张威龙	修改 ADC 接口模块，统一各处变量名与寄存器名保持一致，增加触发模式的详细说明。
2017.12.21	1.04	邓廷	修订 FLASH_READY 寄存器描述。
2017.12.27	1.05	邓廷	修订 FLASH 模块描述



2018.01.03	1.06	钟书鹏	修改 ADC 接口部分，增加量程和增益使用，修改触发部分描述。统一全部章节的排序符号
2018.01.20	1.07	邓廷	修改 HALL 模块描述，MCPWM 模块描述
2018.01.31	1.08	邓廷	增加 SYS_AFE_CMP 寄存器，修改 MCPWM 描述
2018.02.01	1.09	张威龙	更新模拟寄存器表格格式，修改图表目录错误
2018.02.03	1.10	钟书鹏	修改模拟寄存器说明格式，将各个模拟模块设计到模拟寄存器的部分通过超链接连至具体的模拟寄存器说明处
2018.03.27	1.11	邓廷	更新对 TIMER 模块捕获模式的说明
2018.04.05	1.12	钟书鹏	更新温度传感器和比较器的描述
2018.05.29	1.13	邓廷	修正 HALL 模块寄存器的描述
2018.08.06	1.14	张威龙	增加 ADC 校正说明
2018.08.08	1.15	邓廷	修正看门狗喂狗时间，休眠唤醒间隔时间
2018.11.16	1.16	邓廷，张威龙	补充部分应用说明
2018.12.16	1.17	邓廷	修改 MCPWM 滤波宽度描述及 TCLK 寄存器描述
2018.12.17	1.18	张威龙	更新部分表格格式，增加 UART 低波特率设置的应用说明
2018.12.20	1.19	邓廷	修改 MCPWM 部分描述
2019.01.04	1.20	邓廷	修改 MCPWM IF/EIF 描述
2019.03.18	1.21	张威龙	针对发布的修订
2019.03.21	1.22	邓廷	修订 UTIMER 部分，Encoder 复用 Timer 部分的描述
2019.03.26	1.23	张威龙	8.2.5 增加 Encoder T1/T2 信号来源说明

