

G32F0xx 系列 MCU

参考手册

目录

1 简介	29
1.1 概述	29
1.2 主要特点	29
1.3 器件功能表	0
2 封装及管脚配置	1
2.1 概述	1
2.2 LQFP 32/QFN 32 封装	1
2.3 SSOP 20 封装	2
2.4 管脚功能描述	3
2.5 外设管脚描述	5
3 系统框图	7
4 系统描述	8
4.1 ARM Cortex™-M0 内核	8
4.2 内存映射	9
4.3 Cortex™-M0 内核控制模块	12
4.3.1 概述	12
4.3.2 CPUID 寄存器	12
4.3.3 中断控制和状态寄存器	12
4.3.4 应用程序中断控制和复位寄存器	14
4.3.5 系统控制寄存器	14
4.3.6 配置和控制寄存器	14
4.3.7 系统中断处理优先级寄存器	15

4.4	嵌套向量中断控制器(NVIC)	15
4.4.1	NVIC 概述.....	15
4.4.2	嵌套向量中断控制器(NVIC)寄存器列表	17
4.4.3	中断允许寄存器	17
4.4.4	中断禁止寄存器	18
4.4.5	中断挂起寄存器	18
4.4.6	清除中断挂起寄存器.....	18
4.4.7	中断优先级寄存器	18
4.5	系统定时器.....	19
4.5.1	系统定时器控制和状态寄存器	20
4.5.2	系统定时器重载寄存器	20
4.5.3	系统定时器当前值寄存器	20
4.5.4	使用系统定时器	21
4.6	系统控制	22
4.6.1	系统复位.....	23
4.6.1.1	系统内存重映射寄存器	24
4.6.1.1	系统复位状态寄存器.....	24
4.6.1.2	软件复位	24
4.6.1.3	POR.....	24
4.6.1.4	BOD.....	25
4.6.2	时钟控制	25
4.6.2.1	概述	25
4.6.2.2	看门狗振荡器控制寄存器.....	26

4.6.2.1	内部晶体振荡器控制寄存器	27
4.6.2.1	RTC 配置寄存器	27
4.6.2.1	系统主时钟	28
4.6.2.2	AHB 时钟控制	28
4.6.2.1	UART 时钟控制	30
4.6.2.1	CLKOUT 输出时钟控制	31
4.6.2.2	可编程抗干扰滤波器	31
4.6.3	功耗管理	32
4.6.3.1	电源控制寄存器	33
4.6.3.1	深度睡眠配置寄存器	33
4.6.3.1	唤醒配置寄存器	34
4.6.3.2	电源配置寄存器	34
4.6.3.3	正常工作模式	35
4.6.3.4	睡眠模式	35
4.6.3.5	深度睡眠模式	36
4.6.3.6	掉电模式	37
4.6.4	深度睡眠模式唤醒控制	38
4.6.4.1	深度睡眠唤醒控制寄存器	38
4.6.4.2	深度睡眠唤醒信号使能寄存器	39
4.6.4.3	深度睡眠唤醒信号复位寄存器	41
4.6.4.4	深度睡眠唤醒信号状态寄存器	42
4.6.5	其它	43
4.6.5.1	NMI 中断源配置寄存器	43

4.6.5.1	外设复位控制寄存器	44
4.7	I/O 配置	45
4.7.1	IOCON 寄存器描述	45
4.7.1.1	管脚功能	46
4.7.1.2	管脚模式	46
4.7.1.3	管脚驱动	46
4.7.1.4	开漏模式	46
4.7.1.5	可编程抗干扰滤波器	47
4.7.2	IOCON 寄存器列表	47
4.7.2.1	PA0 IOCON 寄存器	48
4.7.2.2	PA1 IOCON 寄存器	48
4.7.2.3	PA2 IOCON 寄存器	48
4.7.2.4	PA3 IOCON 寄存器	48
4.7.2.5	PA4 IOCON 寄存器	48
4.7.2.6	PA5 IOCON 寄存器	49
4.7.2.7	PA6 IOCON 寄存器	49
4.7.2.8	PA7 IOCON 寄存器	49
4.7.2.9	PA8 IOCON 寄存器	50
4.7.2.10	PA9 IOCON 寄存器	50
4.7.2.11	PA10 IOCON 寄存器	50
4.7.2.12	PA11 IOCON 寄存器	50
4.7.2.13	PA12 IOCON 寄存器	51
4.7.2.14	PA13 IOCON 寄存器	51

4.7.2.15	PA14 IOCON 寄存器	51
4.7.2.16	PA15 IOCON 寄存器	51
4.7.2.17	PB0 IOCON 寄存器	51
4.7.2.18	PB1 IOCON 寄存器	52
4.7.2.19	PB3 IOCON 寄存器	52
4.7.2.20	PB4 IOCON 寄存器	52
4.7.2.21	PB5 IOCON 寄存器	52
4.7.2.22	PB6 IOCON 寄存器	53
4.7.2.23	PB7 IOCON 寄存器	53
4.7.2.24	PC0 IOCON 寄存器	53
4.7.2.25	PC1 IOCON 寄存器	53
4.7.2.26	PC2 IOCON 寄存器	54
4.7.2.27	PC3 IOCON 寄存器	54
5	GPIO	55
5.1	概述	55
5.2	管脚描述	55
5.3	GPIO 控制寄存器	55
5.3.1	GPIO 屏蔽寄存器	56
5.3.2	GPIO 管脚值寄存器	56
5.3.3	GPIO 管脚输出寄存器	57
5.3.4	GPIO 管脚输出置位寄存器	57
5.3.5	GPIO 管脚输出清除寄存器	57
5.3.6	GPIO 取反寄存器	58

5.3.7	GPIO 数据方向寄存器	58
5.3.8	GPIO 中断感应寄存器	58
5.3.9	GPIO 中断配置寄存器	58
5.3.10	GPIO 中断事件寄存器	58
5.3.11	GPIO 中断屏蔽寄存器	59
5.3.12	GPIO 原始中断状态寄存器	59
5.3.13	GPIO 中断状态寄存器	59
5.3.14	GPIO 中断清除寄存器	59
6	增强型串口 (UART)	60
6.1	概述	60
6.2	管脚描述	61
6.3	UART 寄存器描述	61
6.3.1	接收缓冲寄存器	61
6.3.2	发送保持寄存器	61
6.3.3	UART 状态寄存器	62
6.3.4	UART 控制寄存器	62
6.3.5	UART 中断状态寄存器	64
6.3.6	UART 波特率分频器寄存器	64
6.3.7	UART TX/RX FIFO 数据清除寄存器	65
6.4	操作描述	65
6.4.1	UART 通讯约定	65
6.4.2	IrDA 红外接收发送功能	65
7	基本型 16 位定时器/计数器 TIM0/TIM1	66
7.1	概述	66

7.2	寄存器描述.....	67
7.2.1	中断寄存器	67
7.2.2	定时器控制寄存器	67
7.2.3	定时器计数寄存器	68
7.2.4	预分频寄存器.....	68
7.2.5	预分频计数寄存器	68
7.2.6	匹配控制寄存器	68
7.2.7	匹配寄存器	70
8	16 位定时器/计数器 TIM2/TIM3.....	71
8.1	概述	71
8.2	管脚描述	72
8.3	寄存器描述.....	73
8.3.1	中断寄存器	74
8.3.2	定时器控制寄存器	74
8.3.3	定时器计数寄存器	74
8.3.4	预分频寄存器.....	75
8.3.5	预分频计数寄存器	75
8.3.6	匹配控制寄存器	75
8.3.7	匹配寄存器组.....	76
8.3.8	捕捉控制寄存器	76
8.3.9	捕捉寄存器组.....	78
8.3.10	外部匹配寄存器	78
8.3.11	计数控制寄存器	79

8.3.11.1	边沿计数模式	81
8.3.11.2	正交编码计数模式	81
8.3.11.3	触发计数模式	82
8.3.11.4	符号计数模式	82
8.3.11.5	门控计数模式	82
9	看门狗定时器 (WDT)	83
9.1	概述	83
9.2	寄存器描述	83
9.2.1	看门狗模式寄存器	84
9.2.2	看门狗定时器常数寄存器	85
9.2.3	看门狗喂狗命令寄存器	86
9.2.4	看门狗定时器寄存器	86
9.2.5	看门狗警告中断比较值寄存器	86
9.2.6	看门狗窗口比较值寄存器	86
9.2.7	看门狗时钟及低功耗控制	87
9.2.8	看门狗锁定功能	87
10	CRC	88
10.1	概述	88
10.2	CRC 接口寄存器描述	88
10.2.1	CRC 模式寄存器	88
10.2.2	CRC Seed 寄存器	89
10.2.3	CRC 校验和寄存器	89
10.2.4	CRC 数据寄存器	89
10.3	功能描述	89

10.3.1	CRC 计算	89
11	SPI	91
11.1	概述	91
11.2	管脚描述	91
11.3	寄存器描述	92
11.3.1	SPI 控制寄存器 0	92
11.3.2	SPI 控制寄存器 1	93
11.3.3	SPI 数据寄存器	93
11.3.4	SPI 状态寄存器	94
11.3.5	SPI 时钟寄存器	94
11.3.6	SPI 中断屏蔽控制寄存器	94
11.3.7	SPI 原始中断状态寄存器	95
11.3.8	SPI 中断状态寄存器	95
11.3.9	SPI 中断清除寄存器	95
11.4	操作	96
11.4.1	SPI 帧数据格式	96
11.4.2	SSI 帧数据格式	99
12	I2C	101
12.1	概述	101
12.2	管脚描述	101
12.3	寄存器描述	102
12.3.1	I2C 控制置位寄存器	102
12.3.2	I2C 状态寄存器	104
12.3.3	I2C 数据寄存器	104

12.3.4	I2C 从机地址寄存器 0~3.....	104
12.3.5	I2C 高电平占空比寄存器	104
12.3.6	I2C 低电平占空比寄存器	105
12.3.7	I2C 控制清除寄存器	105
12.3.8	I2C 数据缓冲寄存器	105
12.3.9	I2C 屏蔽寄存器 0~3.....	106
12.4	I2C 操作	106
12.4.1	主机发送模式.....	106
12.4.2	主机接收模式.....	107
12.4.3	从机接收模式.....	109
12.4.4	从机发送模式.....	110
12.4.5	从机接收发送简化流程	111
12.4.6	状态明细表	112
12.4.7	I2C 状态服务程序	114
12.4.7.1	初始化程序	114
12.4.7.2	启动主机发送功能.....	114
12.4.7.3	启动主机接收功能.....	114
12.4.7.4	I2C 中断服务程序	115
12.4.7.5	无指定模式下的状态服务程序.....	115
12.4.7.6	主机发送状态	115
12.4.7.7	主机接收状态	116
12.4.7.8	从机接收状态	117
12.4.7.9	从机发送状态	118

13	RTC	120
13.1	概述	120
13.2	管脚描述	120
13.3	RTC 寄存器 描述	120
13.3.1	RTC 数据寄存器	120
13.3.2	RTC 匹配寄存器	121
13.3.3	RTC 装载寄存器	121
13.3.4	RTC 控制寄存器	121
13.3.5	RTC 中断控制寄存器	121
13.3.6	RTC 原始中断状态寄存器	121
13.3.7	RTC 中断状态寄存器	122
13.3.8	RTC 中断清除寄存器	122
13.4	功能描述	122
14	ADC	123
14.1	概述	123
14.2	管脚描述	124
14.3	寄存器描述	124
14.3.1	ADC 控制寄存器	124
14.3.2	ADC 全局数据寄存器	125
14.3.3	ADC 通道选择寄存器	126
14.3.4	ADC 中断使能寄存器	126
14.3.5	ADC 数据寄存器组	126
14.3.6	ADC 状态寄存器	127
14.3.7	上限控制寄存器	127

14.3.8	下限控制寄存器	128
14.3.9	软件触发寄存器	128
14.4	操作	128
14.4.1	选择需要进行 ADC 转换的 A/D 通道	128
14.4.2	ADC 触发转换模式	129
14.4.3	中断	129
15	PWM 脉宽调制	130
15.1	概述	130
15.1.1	特点	130
15.1.2	Block Diagram	131
15.2	管脚描述	131
15.3	寄存器描述	132
15.3.1	PWM 控制寄存器	133
15.3.2	PWM 故障输入控制寄存器	135
15.3.3	PWM 故障状态寄存器	136
15.3.4	PWM 输出控制寄存器	137
15.3.5	PWM 计数器寄存器	138
15.3.6	PWM 模数计数器寄存器	138
15.3.7	PWM 数值寄存器组	138
15.3.8	PWM 死区寄存器	139
15.3.9	PWM 映射失效控制寄存器	139
15.3.10	PWM 配置寄存器	140
15.3.11	PWM 通道控制寄存器	141

15.3.12	PWM 故障端口寄存器	142
15.3.13	PWM 死区效应补偿控制寄存器	143
15.3.14	PWM 比较输出极性控制寄存器	144
15.3.15	PWM 计数器初始值寄存器	145
15.4	功能描述	145
15.4.1	时钟预分频	145
15.4.2	PWM 信号发生器	145
15.4.2.1	输出对齐和极性控制	145
15.4.2.2	载波周期	146
15.4.2.3	脉宽占空比	147
15.4.3	独立和互补通道输出	148
15.4.4	Deadtime Generators	149
15.4.4.1	死区补偿效应	150
15.4.5	非对称 PWM 输出	152
15.4.6	PWM 输出极性控制	152
15.4.7	软件输出控制	153
15.4.8	PWM 发生器设置重载	154
15.4.8.1	装载使能	154
15.4.8.2	装载频率	154
15.4.8.3	重载标志	155
15.4.8.4	同步输出控制	156
15.4.8.5	PWM 初始化	157
15.4.9	故障保护	158

15.4.9.1	外部故障管脚信号采样滤波	158
15.4.9.2	故障的自动清除	158
15.4.9.3	故障的手动清除	158
15.4.10	中断	159
16	Flash/SRAM 存储器和器件信息	160
16.1	User Flash/Boot Flash 结构	160
16.2	Flash 操作	161
16.2.1	Flash 控制寄存器列表	161
16.2.2	Flash 访问周期寄存器	162
16.3	芯片硬件信息寄存器	错误!未定义书签。
16.4	ISP 协议和命令	错误!未定义书签。
16.4.1	ISP 命令列表	错误!未定义书签。
16.4.2	ISP 命令返回代码	错误!未定义书签。
16.4.3	删除扇区	错误!未定义书签。
16.4.4	扇区空白检查	错误!未定义书签。
16.4.5	写数据到 SRAM 存储器	错误!未定义书签。
16.4.6	复制 RAM 数据到 Flash	错误!未定义书签。
16.4.7	比较存储器数据	错误!未定义书签。
16.4.8	读存储器数据	错误!未定义书签。
16.4.9	芯片擦除	错误!未定义书签。
16.4.10	读取芯片 ID	错误!未定义书签。
16.4.11	读取 BSL 版本	错误!未定义书签。
16.4.12	跳转到指定地址	错误!未定义书签。

16.5	IAP 命令及入口地址	错误!未定义书签。
17	SWD	163
17.1	概述	163
17.2	管脚描述	163
17.3	调试操作	163
18	Revision History	164

插图

Figure 3-1: G32F0xx 模块框图	7
Figure 5-1: Cortex M0 内核功能模块图	8
Figure 4-2: 内存分配图	11
Figure 5-3: IPR 寄存器	19
Figure 4-4: Clock System block diagram	26
Figure 6-1: UART/IrDA 模块框图	60
Figure 6-2: UART 功能模块图	60
Figure 6-3: UART 和 IrDA 信号转换	65
Figure 7-1: 16 位基本型定时器/计数器模块框图	66
Figure 8-1: 16 位定时器/计数器模块框图	72
Figure 8-2: 正交编码计数	82
Figure 8-3: 触发计数模式	82
Figure 8-4: 符号计数模式	82
Figure 8-5: 门控计数模	82
Figure 11-1: 看门狗功能模块图	83
Figure 13-1: CPOL=0 且 CPHA=0 时的 SPI 帧格式 (a) 单帧和 b) 连续帧传输)	96
Figure 13-2: SPI 在 CPOL = 0 和 CPHA = 1 时的数据格式	97
Figure 13-3: SPI 在 CPOL = 1 和 CPHA = 0 下 (a) 单帧和(b) 连续帧传输	98
Figure 13-4: SPI 在 CPOL = 1 和 CPHA = 1 时的数据格式	99
Figure 13-5: 4 线同步串行帧格式: a) 单帧和 b) 连续/ 背靠背 2 帧传输	100
Figure 15-1: I2C 总线应用框图	101
Figure 15-2: 主机发送模式的格式	107

Figure 15-3:主机发送模式流程图	107
Figure 15-4: 主机接收模式的格式	108
Figure 15-5: 主机接收模式流程图	108
Figure 15-6: 从机接收模式下的格式	109
Figure 15-7: 从机接收模式使用状态码控制流程图	110
Figure 15-8: 从机发送模式下的格式	110
Figure 15-9: 从机发送模式使用状态码控制流程图	111
Figure 15-10: 从机接收发送模式中断服务流程图	111
Figure 16-1: ADC 功能模块框图	123
Figure 17-1: PWM 功能模块框图	131
Figure 17-2: 中心对齐 PWM 输出	146
Figure 17-3: 中心对齐 PWM 周期	146
Figure 17-4: 边沿对齐 PWM 周期	146
Figure 17-5: 中心对齐方式下脉冲宽度	147
Figure 17-6: 边沿对齐方式下脉冲宽度	147
Figure 17-7: 互补通道对	148
Figure 17-8: 三相交流逆变器	148
Figure 17-9: 死区发生器	149
Figure 17-10: 互补输出模式下的死区插入	150
Figure 17-11: 临界占空比下的死区插入	150
Figure 17-12: 短脉冲下的死区插入	150
Figure 17-13: Deadtime Distortion	150
Figure 17-14: 正电流下的死区补偿	151

Figure 17-15: 负电流下的死区补偿	152
Figure 17-16: 非对称波形 – 相位移位的 PWM 输出	152
Figure 17-17: PWM Output Polarity	153
Figure 17-18: 互补模式下的软件输出控制	154
Figure 17-19: 全周期重载频率变化	154
Figure 17-20: 半周期重载频率变化	155
Figure 17-21: 中心对齐的 PWM 数值寄存器全周期重载	155
Figure 17-22: 中心对齐的 PWM 模数寄存器全周期重载	155
Figure 17-23: 中心对齐的 PWM 数值寄存器半周期重载	156
Figure 17-24: 中心对齐的 PWM 模数寄存器半周期重载	156
Figure 17-25: 边沿对齐的 PWM 数值寄存器周期重载	156
Figure 17-26: 边沿对齐的 PWM 模数寄存器半周期重载	156
Figure 17-27: PWMEN 和 PWM 管脚在独立模式下工作(OUTCTL0-5 = 0)	157
Figure 17-28: PWMEN 和 PWM 管脚在互补模式下工作(OUTCTL0,2,4 = 0)	157
Figure 17-29: 故障自动清除	158
Figure 18-1: 用户程序和引导程序地址映射空间分布	160
Figure 20-4: ISP 通信命令数据包结构	错误!未定义书签。
Figure 18-2: IAP 参数传递	错误!未定义书签。
Figure 19-1: SWD 连接器与 SWD 管脚连接示意图	164

表

Table 2-1: 管脚功能描述	3
Table 2-2: 外设管脚选择	5
Table 4-1: 内存地址分配表	11
Table 5-2: Cortex M0 内核 SCB 寄存器一览 (基地址: 0xE000-ED00).....	12
Table 5-3: CPUID 寄存器 (CPUID, 地址: 0xE000-ED00) 描述	12
Table 5-4: 中断控制和状态寄存器(ICSR, 地址: 0xE000-ED04) 描述	12
Table 5-5: 应用程序中断控制和重启寄存器(AIRCR, 地址: 0xE000-ED0C) 描述	14
Table 5-6: 系统控制寄存器(SCR, 地址: 0xE000-ED10) 描述	14
Table 5-7: 配置和控制寄存器(CCR, 地址: 0xE000-ED14) 描述	15
Table 5-8: SHPR2 寄存器 (地址: 0xE000-ED1C) 描述	15
Table 5-9: SHPR3 寄存器 (地址: 0xE000-ED20) 描述	15
Table 5-10: 中断向量源和向量表	16
Table 5-11: NVIC 寄存器列表	17
Table 5-12: 中断允许寄存器(ISER, 地址: 0xE000 E100) 描述	17
Table 5-13: 中断禁止寄存器(ICER, 地址: 0xE000 E180)描述	18
Table 5-14: 中断挂起寄存器(ISPR, 地址: 0xE000 E200) 描述	18
Table 5-15: 清除中断挂起寄存器(ICPR, 地址: 0xE000 E280) 描述	18
Table 5-16: IPRn 寄存器 (IPR0~7, 地址: 0xE000 E400~0xE000 E41C) 描述.....	19
Table 5-18: 系统定时器控制和状态寄存器 (SYST_CSR - 0xE000 E010) 描述	20
Table 5-19: 系统定时器重载寄存器(SYST_RVR - 0xE000 E014) 描述	20
Table 5-20: 系统定时器当前值寄存器(SYST_CVR - 0xE000 E018) 描述	20
Table 5-21: 系统控制寄存器列表	22

Table 5-22: 系统内存重映射寄存器(SYSMEMREMAP, 地址: 0x4004 8000) 描述	24
Table 5-23: 系统复位状态寄存器(SYSRESSTAT, 地址: 0x4004 8030) 描述	24
Table 5-27: BOD 控制寄存器 (BODCTRL, 地址: 0x4004 8150)描述.....	25
Table 5-29: 看门狗振荡器控制寄存器(WDTOSCCTRL, 地址 0x4004 8024) 描述	27
Table 5-30: 内部晶体振荡器控制寄存器(IRCCTRL, 地址 0x4004 8028) 描述	27
Table 5-49: 系统配置寄存器(SYSCFG, 地址 0x4003 8014) 描述	27
Table 5-37: 主时钟源选择寄存器 (MAINCLKSEL, 地址 0x4004 8070) 描述	28
Table 5-38: 主时钟源更新寄存器(MAINCLKUEN, 地址 0x4004 8074) 描述	28
Table 5-39: AHB 接口时钟分频器寄存器(SYSAHBCLKDIV, 地址 0x4004 8078) 描述	28
Table 5-40: AHB 接口时钟控制寄存器(SYSAHBCLKCTRL, 地址 0x4004 8080) 描述	29
Table 5-41: UARTn 时钟分频器控制寄存器 (UART0CLKDIV, 地址 0x4004 8098; UART1CLKDIV, 地址 0x4004 809C;) 描述.....	30
Table 5-42: CLKOUT 输出时钟源选择寄存器 (CLKOUTCLKSEL, 地址 0x4004 80E0) 描述	31
Table 5-43: CLKOUT 时钟源更新使能寄存器(CLKOUTUEN, 地址 0x4004 80E4) 描述	31
Table 5-44: CLKOUT 输出时钟分频器寄存器 (CLKOUTDIV, 地址 0x4004 80E8) 描述	31
Table 5-45: IOCONFIG 滤波器时钟寄存器 0 (IOCONFIGCLKDIV0 地址 4004 8014C) 描述.....	32
Table 5-46: 不同工作模式下系统模块电源和时钟状态	32
Table 5-47: 电源控制寄存器 (PCON, 地址 0x4003 8000) 描述	33
Table 5-50: 深度睡眠配置寄存器(PDSLEEP_CFG, 地址 0x4004 8230) 描述.....	33
Table 5-51: 唤醒配置寄存器(PDAWAKECFG, 地址 0x4004 8234) 描述	34
Table 5-52: 电源配置寄存器(PDRUNCFG, 地址 0x4004 8238) 描述.....	35
Table 5-53: 深度睡眠唤醒控制寄存器(DSWAKECTL, 地址 0x4004 8200) 描述	38
Table 5-54: 深度睡眠唤醒信号使能寄存器(DSWAKEEN, 地址 0x4004 8204) 描述.....	39
Table 5-55: 深度睡眠唤醒信号复位寄存器(DSWAKECLR, 地址 0x4004 8208) 描述.....	41

Table 5-56: 深度睡眠唤醒信号状态寄存器(DSWAKE, 地址 0x4004 820C) 描述	42
Table 5-17: NMI 中断源配置寄存器(INTNMI, 地址 0x4004 8174)描述	43
Table 5-57: 外设复位控制寄存器(PRESETCTRL, 地址 0x4004 8004) 描述	44
Table 5-58: IOCON 寄存器 (I2C 管脚除外)	45
Table 5-59: I/O 配置寄存器列表 (基地址 0x4004 4000)	47
Table 4-50: PA0 寄存器功能位选择	48
Table 4-51: PA1 寄存器功能位选择	48
Table 4-52: PA2 寄存器功能位选择	48
Table 4-53: PA3 寄存器功能位选择	48
Table 4-54: PA4 寄存器功能位选择	48
Table 4-55: PA5 寄存器功能位选择	49
Table 4-56: PA6 寄存器功能位选择	49
Table 4-57: PA7 寄存器功能位选择	49
Table 4-58: PA8 寄存器功能位选择	50
Table 4-59: PA9 寄存器功能位选择	50
Table 4-60: PA10 寄存器功能位选择	50
Table 4-61: PA11 寄存器功能位选择	50
Table 4-62: PA12 寄存器功能位选择	51
Table 4-63: PA13 寄存器功能位选择	51
Table 4-64: PA14 寄存器功能位选择	51
Table 4-65: PA15 寄存器功能位选择	51
Table 4-66: PB0 寄存器功能位选择	51
Table 4-67: PB1 寄存器功能位选择	52

Table 4-69: PB3 寄存器功能位选择	52
Table 4-70: PB4 寄存器功能位选择	52
Table 4-71: PB5 寄存器功能位选择	52
Table 4-72: PB6 寄存器功能位选择	53
Table 4-73: PB7 寄存器功能位选择	53
Table 4-82: PC0 寄存器功能位选择	53
Table 4-83: PC1 寄存器功能位选择	53
Table 4-84: PC2 寄存器功能位选择	54
Table 4-85: PC10 寄存器功能位选择	54
Table 5-1: GPIO 管脚及端口.....	55
Table 5-2: 寄存器概述: GPIO (基地址 port 0: 0x5000 0000; port 1: 0x5001 0000, port 2: 0x5002 0000).....	55
Table 7-3: GPIO 屏蔽寄存器 (MASK - 地址 0x5000 0000 (GPIOA), 0x5001 0000 (GPIOB), 0x5002 0000 (GPIOC)) 描述	56
Table 7-4: GPIO 管脚值寄存器 (PIN - 地址 0x5000 0004 (GPIOA), 0x5001 0004 (GPIOB);0x5002 0004 (GPIOC)) 描述	56
Table 7-5: GPIO 管脚输出寄存器 (OUT - 地址 0x5000 0008 (GPIOA), 0x5001 0008 (GPIOB), 0x5002 0008 (GPIOC)) 描述	57
Table 7-6: GPIO 管脚输出置位寄存器 (SET - 地址 0x5000 000C (GPIOA), 0x5001 000C (GPIOB), 0x5002 000C (GPIOC)) 描述	57
Table 7-7: GPIO 管脚输出清除寄存器 (CLR - 地址 0x5000 0010 (GPIOA), 0x5001 0010 (GPIOB), 0x5002 0010 (GPIOC)) 描述	57
Table 7-8: GPIO 取反寄存器 (NOT - 地址 0x5000 0014 (GPIOA), 0x5001 0014 (GPIOB), 0x5002 0014 (GPIOC)) 描述	58
Table 7-9: GPIO 数据方向寄存器 (DIR - 地址 0x5000 0020 (GPIOA), 0x5001 0020 (GPIOB), 0x5002 0020 (GPIOC)) 描述	58

Table 7-10: GPIO 中断感应寄存器 (IS - 地址 0x5000 0024 (GPIOA), 0x5001 0024 (GPIOB), 0x5002 0024 (GPIOC))	
描述	58
Table 7-11: GPIO 边沿中断配置寄存器 (IBE - 地址 0x5000 0028 (GPIOA), 0x5001 0028 (GPIOB), 0x5002 0028 (GPIOC))	
描述	58
Table 7-12: GPIO 中断事件寄存器 (IEV - 地址 0x5000 002C (GPIOA), 0x5001 002C (GPIOB), 0x5002 002C (GPIOC))	
描述	58
Table 7-13: GPIO 中断屏蔽寄存器 (IE - 地址 0x5000 0030, 0x5001 0030 (GPIOB), 0x5002 0030 (GPIOC))	描述.....
	59
Table 7-14: GPIO 原始中断状态寄存器 (RIS - 地址 0x5000 0034 (GPIOA), 0x5001 0034 (GPIOB), 0x5002 0034 (GPIOC))	
描述	59
Table 7-15: GPIO 被屏蔽中断状态寄存器 (MIS - 地址 0x5000 0038 (GPIOA), 0x5001 0038 (GPIOB), 0x5002 0038 (GPIOC))	
描述	59
Table 7-16: GPIO 中断清除寄存器 (IC - 地址 0x5000 003C, 0x5001 003C (GPIOB), 0x5002 003C (GPIOC))	描述..
	59
Table 6-1: 管脚描述	61
Table 6-2: 寄存器一览 (UART0: 基地址-0x4000 8000; UART1 基地址-0x4000 c000).....	61
Table 12-3: UART 接收缓冲寄存器 (RDR) 描述	61
Table 12-4: UART 发送保持寄存器 (TDR) 描述	61
Table 6-5: UART 状态寄存器 (STATE) 描述	62
Table 6-6: UART 控制寄存器 (CTRL) 描述	62
Table 12-7: UART 中断状态寄存器 (INTSTATUS) 描述	64
Table 12-8: UART 波特率分频器寄存器 (BAUDDIV) 描述	64
Table 6-9: UART FIFO 清除寄存器 (FIFOCLR) 描述	65
Table 9-1: 定时器/计数器管脚描述	72
Table 9-2: 16 位定时器/计数器的寄存器一览(TIM2 基地址 0x4001 0000; TIM3 基地址 0x4001 4000)	73
Table 9-3: 中断寄存器 (IR, 地址: 0x4001 0000 (TIM2)和 0x4001 4000 (TIM3)) 描述	74
Table 9-4: 定时器控制寄存器(TCR, 地址 : 0x4001 0004 (TIM2), 0x4001 4004 (TIM3)) 描述	74

Table 9-5: 定时器计数寄存器(TC, address 0x4001 0008 (TIM2) and 0x4001 4008 (TIM3)) 描述	74
Table 9-6: 预分频寄存器 (PR, 地址: 0x4001 000C (TIM2), 0x4001 400C (TIM3)) 描述	75
Table 9-7: 预分频计数寄存器 (PC, 地址: 0x4001 0010 (TIM2), 0x4001 4010 (TIM3)) 描述	75
Table 9-8: 匹配控制寄存器寄存器 (MCR, 地址: 0x4001 0014 (TIM2) , 0x4001 4014 (TIM3)) 描述	75
Table 9-9: 匹配寄存器组 (MR0 ~ 3, 起始地址: 0x4001 0018~ 24 (TIM2) , 0x4001 4018 ~ 24 (TIM3)) 描述	76
Table 9-10: 捕捉控制寄存器 (CCR, 地址: 0x4001 0028 (TIM2) , 0x4001 4028 (TIM3)) 描述	76
Table 9-11: 捕捉寄存器组(CR0 ~ 3, 起始地址: 0x4001 002C~ 38 (TIM2), 0x4001 402C~ 38 (TIM3)) 描述	78
Table 9-12: 外部匹配寄存器 (EMR, 地址: 0x4001 003C (TIM2), 0x4001 403C (TIM3)) 描述	78
Table 9-13: 计数寄存器 (CTCR, 地址: 0x4001 0070 (TIM2) , 0x4001 4070 (TIM3)) 描述	80
Table 10-1: 看门狗寄存器一览(基地址 0x4000 4000)	83
Table 10-2: 看门狗模式寄存器 (MOD - 0x4000 4000) 描述	84
Table 10-3: 看门狗运行模式选择	85
Table 10-4: 看门狗定时器常数寄存器寄存器 (TC - 0x4000 4004) 描述	86
Table 10-5: 看门狗喂狗命令寄存器 (FEED - 0x4000 4008) 描述	86
Table 10-6: 看门狗定时器值寄存器 (TV - 0x4000 400C) 描述	86
Table 10-8: 看门狗警告中断寄存器(WARNINT - 0x4000 4014) 描述	86
Table 10-9: 看门狗窗口比较值寄存器 (WINDOW - 0x4000 4018) 描述	86
Table 11-1: 寄存器一览 (基地址 0x5007 0000)	88
Table 11-2: CRC 模式寄存器 (CRC_MODE, 地址 0x5007 0000) 描述	88
Table 11-3: CRC 种子寄存器 (CRC_SEED, 地址 0x5007 0004) 描述	89
Table 11-4: CRC 校验和寄存器 (CRC_SUM, 地址 0x5007 0008) 描述	89
Table 11-5: CRC Data 寄存器 (CRC_DATA, 地址 0x5007 0008) 描述	89
Table 13-1: SPI 管脚描述	91

Table 13-2: 寄存器一览 (基地址 0x4004 0000)	92
Table 13-3: SPI 控制寄存器 0 (CR0 - 地址 0x4004 0000) 描述	92
Table 13-4: SPI 控制寄存器 1 (CR1 - 地址 0x4004 0004) 描述	93
Table 13-5: SPI 数据寄存器 (DR - 地址 0x4004 0008) 描述	94
Table 13-6: SPI 状态寄存器 (SR - 地址 0x4004 000C) 描述	94
Table 13-7: SPI 时钟预分频寄存器 (CPSR - 地址 0x4004 0010) 描述	94
Table 13-8: SPI 中断屏蔽控制寄存器 (IMSC - 地址 0x4004 0014) 描述	94
Table 13-9: SPI 原始中断状态寄存器 (RIS - 地址 0x4004 0018) 描述	95
Table 13-10: SPI 中断状态寄存器 (MIS - 地址 0x4004 001C) 描述	95
Table 13-11: SPI 中断清除寄存器 (ICR - 地址 0x4004 0020) 描述	95
Table 15-1: I2C 总线管脚 描述	101
Table 15-2: 寄存器一览 (基地址 0x4000 0000)	102
Table 15-3: I2C 控制置位寄存器 (CONSET - 地址 0x4000 0000) 描述	102
Table 15-4: I2C 状态寄存器 (STAT - 0x4000 0004) 描述	104
Table 15-5: I2C 数据寄存器 (DAT - 0x4000 0008) 描述	104
Table 15-6: I2C 从机地址 寄存器 0 (ADR0- 0x4000 000C, ADR1 - 0x4000 0020, ADR2 - 0x4000 0024, ADR3 - 0x4000 0028)) 描述	104
Table 15-7: I2C 高电平占空比寄存器 (SCLH - 地址 0x4000 0010) 描述	104
Table 15-8: I2C 低电平占空比寄存器 (SCLL - 地址 0x4000 0014) 描述	105
Table 15-9: I2C 控制清除寄存器 (CONCLR - 0x4000 0018) 描述	105
Table 15-10: I2C 数据缓冲寄存器 (DATA_BUFFER - 0x4000 002C) 描述	105
Table 15-11: I2C 屏蔽寄存器 (MASK0 - 0x4000 0030, MASK1 - 0x4000 0034, MASK2 - 0x4000 0038, MASK3 - 0x4000 003C) 描述	106
Table 15-12: I2C CONSET 寄存器配置为主机发送模式	106

Table 15-13: I2C CONSET 寄存器配置为从机接收模式	109
Table 15-14: 主机发送模式	112
Table 15-15: 主机接收模式	112
Table 15-16: 从机接收模式	113
Table 15-17: 从机发送模式	113
Table 16-1: RTC 管脚描述	120
Table 16-2: RTC 寄存器一览 (基地址 0x4005 0000)	120
Table 16-3: RTC 数据寄存器 (DR - 地址 0x4005 0000) 描述	120
Table 16-4: RTC 匹配寄存器 (MR - 地址 0x4005 0004) 描述	121
Table 16-5: RTC 装载寄存器 (LR - 地址 0x4005 0008) 描述	121
Table 16-6: RTC 控制寄存器 (CR - 地址 0x4005 000C) 描述	121
Table 16-7: RTC 中断 Mask 寄存器 (ICSC - 地址 0x4005 0010) 描述	121
Table 16-8: RTC 中断状态寄存器 (RIS - 地址 0x4005 0014) 描述	121
Table 16-9: RTC 中断状态寄存器 (MIS - 地址 0x4005 0018) 描述	122
Table 12-16-10: RTC 中断清除寄存器 (ICR - 地址 0x4005 001C) 描述	122
Table 17-1: ADC 管脚描述	124
Table 17-2: ADC 寄存器一览 (ADC0: 基地址 0x4002 0000;)	124
Table 17-4: A/D 控制寄存器 (CR) 描述	124
Table 17-5: A/D 全局数据寄存器 (GDR) 描述	125
Table 17-6: A/D 通道选择寄存器 (CHSEL) 描述	126
Table 17-7:: A/D 中断使能寄存器 (INTEN) 描述	126
Table 17-8: A/D 数据寄存器组(DR0 ~ DR7) 描述	126
Table 17-9: A/D 状态寄存器 (STAT) 描述	127

Table 17-10: 上限控制寄存器(HILMT) 描述	127
Table 17-11: 下限控制寄存器(LOLMT) 描述	128
Table 17-12: 软件触发寄存器(SSCR) 描述	128
Table 6-1: PWM 管脚描述	131
Table 6-2: PWM 寄存器一览(基地址 0x4004 C000).....	132
Table 17-3 PWM 控制(CTRL) 寄存器	133
Table 17-4 PWM 故障输入控制住(FCTRL) 寄存器	135
Table 17-5 PWM 故障状态 (FLTACK) 寄存器	136
Table 17-6 PWM 输出控制(OUT) 寄存器	137
Table 17-7 PWM 计数器(CNTR) 寄存器	138
Table 17-8 PWM 模数计数器(CMOD) 寄存器	138
Table 17-9 PWM 数值 (VALm, m=0~5) 寄存器	138
Table 17-10 PWM 死区 (DTIMm, m=0,1) 寄存器	139
Table 17-11: 映射失效 (DMAP0) 寄存器 0.....	139
Table 17-12: 映射失效 (DMAP1) 寄存器 1	139
Table 17-13: PWM 配置(CNFG) 寄存器.....	140
Table 17-14: PWM 通道(CCTRL) 寄存器	141
Table 17-15: PWM 故障端口 (FPORTCTRL) 寄存器	142
Table 17-16: PWM 死区效应补偿控制(ICCTRL) 寄存器	143
Table 17-17: PWM 比较输出极性控制(PSCR) 寄存器	144
Table 17-18: PWM 计数器初始值 (CNTRINI) 寄存器	145
Table 17-19: 上/下边输出手动补偿	151
Table 17-20: ICCn 位控制的上/下边输出补偿	152

Table 18-1: Flash 寄存器一览 (基地址: 0x5006 0000)	161
Table 18-2: Flash 访问周期寄存器(FLASH_RDCYC, 地址 0x5006 0028) 描述.....	162
Table 18-3: 器件硬件信息寄存器组(基地址: 0x1FFF 1400).....	错误!未定义书签。
Table 17-4: ISP 命令代码	错误!未定义书签。
Table 20-9: ISP 命令执行成功时返回代码	错误!未定义书签。
Table 17-6: ISP 错误代码	错误!未定义书签。
Table 18-4: IAP 命令.....	错误!未定义书签。
Table 19-1: SWD 管脚 描述.....	163

1 简介

1.1 概述

G32F0xx 系列是内嵌 ARM Cortex™ M0 核的 32 位低成本微控制器。主要应用是智能家居消费类市场，如电饭煲，微波炉，豆浆机等等。 G32F0xx 系列控制器具备有常用外设和功能，如高速 12 位的 ADC 转换器，UART 串口，SPI 接口，I2C 总线接口，看门狗定时器（WDT），4 个通用计数器/定时器。除此之外，G32F0xx 还集成人机界面控制器和马达控制功能，如直流无刷电机控制 PWM 模块。

1.2 主要特点

•高性能 32位 ARM CPU 内核

- 速度高达48Mhz 的ARM Cortex™-M0 处理器
- 可嵌套中断向量控制器(NVIC)
- 单周期32位硬件乘法器
- 24位系统嘀嗒定时器

•Flash 内存

- 8/16/32/64K字节的用户程序空间
- 4K 字节启动程序
- 32位Flash数据总线，支持高性能应用要求

•SRAM 内存

- 内嵌2/4K 字节高速SRAM

•16位PWM发生器

- 多达6通道PWM输出
- 自动死区插入
- 基于硬件的保护系统

•内嵌系统引导程序（bootloader）

- 支持Flash存储器In-System-Program (ISP) 和 In-Application-Program (IAP)
- 支持用户程序保护和加密

•灵活的时钟单元

- 多个时钟源

- 内置40/48 MHz精度 1% 的内部高速振荡器

- 内置32KHz 看门狗振荡器

•增强的定时器/计数器

- 2个基础定时器/计数器和2个增强型定时器/计数器
- 基础定时器/计数器支持匹配中断功能
- 增强型定时器/计数器支持边沿计数，门控计数，AB 相正交计数，触发计数，符号计数

•模拟外设

- 12位 1MHz SAR ADC 转换器
- 最多8个通道
- 支持软件和硬件AD转换触发模式

•丰富的通信接口和通用输入/输出端口（GPIO）

- 2个具备16字节FIFO 的串口通信模块，同时支持 IrDA 协议
- 扩展的 SPI接口，支持多种协议
- I2C接口
- 多达 27个高速GPIO接口

•电源管理

- 三种省电模式：睡眠模式，停止模式，待机模式
- 集成的电源管理单元（PMU）

•工作温度范围

- 工业级 (-40°C ~ +85°C)

•2.2V~5.5 V 宽压工作范围

•LQFP32, QFN32, SSOP20 多种封装

1.3 器件功能表

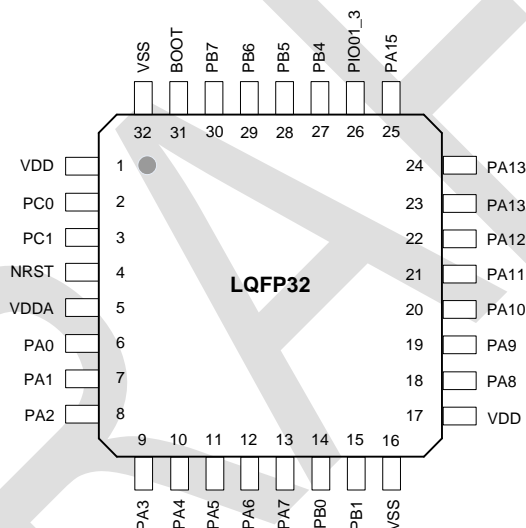
型号	Flash (KB)	SRAM (KB)	PMU	Timer	CRC	RTC	PWM	ADC	UART	I2C	SPI	WDT	GPIO	封装
G32F0324	32	4	√	4	√	-	6ch	8ch/12bit	2	1	1	√	27	LQFP32/QFN32
G32F0084	16	4	√	4	√	-	3ch	8ch/12bit	1	1	1	√	17	SSOP-20

2 封装及管脚配置

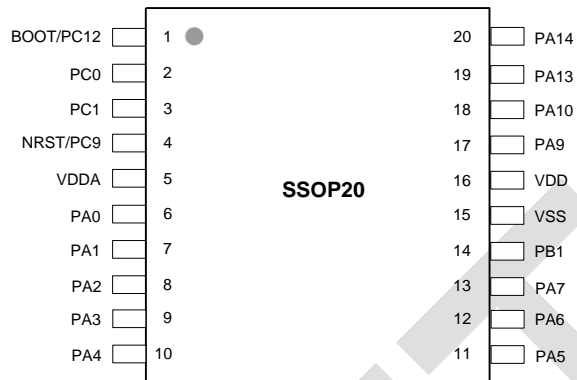
2.1 概述

G32F0xx 有 LQFP32, LQFN 32 和 SSOP20 多种封装。管脚功能由 IO 控制寄存器配置，除电源管脚，其余管脚均可复用。系统复位后，管脚功能将被设置成默认值。

2.2 LQFP 32/QFN 32 封装



2.3 SSOP 20 封装



2.4 管脚功能描述

Table 2-1: 管脚功能描述

管脚名称	LQFPN 32	SSOP 20	唤醒 输入	管脚 类型	初始 设置	描述
PC0	2	2		I/O	I; PU	PC0 — General purpose digital input/output pin
PC1	3	3		I/O	I; PU	PC1 — General purpose digital input/output pin
NRST	4	4		I	I; PU	NRST— Device reset input (active low)
				I/O		PC2 — General purpose digital input/output pin
VDDA	5	5		S		Analog power supply
PA0	6	6		I/O	I; PU	PA0 — General purpose digital input/output pin
				I		ADC_IN0 — ADC input channel 0
PA1	7	7		I/O	I; PU	PA1 — General purpose digital input/output pin
				I		ADC_IN1 — ADC input channel 1
PA2	8	8		I/O	I; PU	PA2 — General purpose digital input/output pin
				I		TIM3_CAP0 — Capture input, channel 0 for 16-bit timer 3
				O		TIM3_MAT0 — Match output, channel 0 for 16-bit timer 3
				O		TXD0 — Transmitter output for UART0
				I		ADC_IN2 — ADC input channel 2
PA3	9	9		I/O	I; PU	PA17 — General purpose digital input/output pin
				I		TIM3_CAP1 — Capture input, channel 1 for 16-bit timer 3
				O		TIM3_MAT1 — Match output, channel 1 for 16-bit timer 3
				I		RXD0 — Receiver input for UART0
				I		ADC_IN3 — ADC input channel 3
PA4	10	10		I/O	I; PU	PA4 — General purpose digital input/output pin
				I		TIM3_CAP2 — Capture input, channel 2 for 16-bit timer 3
				O		TIM3_MAT2 — Match output, channel 2 for 16-bit timer 3
				I/O		SPI_SSEL — SPI chip select
				I		ADC_IN4 — ADC input channel 4
PA5	11	11		I/O	I; PU	PA5 — General purpose digital input/output pin
				I/O		SPI_SCK — SPI serial clock
				I		ADC_IN5 — ADC input channel 5
PA6	12	12		I/O	I; PU	PA6 — General purpose digital input/output pin
				I		PWM_FAULT— PWM Fault in
				I		TIM2_CAP0 — Capture input, channel 0 for 16-bit timer 2
				O		TIM2_MAT0 — Match output, channel 0 for 16-bit timer 2
				I/O		SPI_MISO — SPI master in slave out

管脚名称	LQFP/N 32	SSOP 20	唤醒 输入	管脚 类型	初始 设置	描述
				I		ADC_IN6 — ADC input channel 6
PA7	13	13		I/O	I; PU	PA7 — General purpose digital input/output pin
				O		PWM_OUT1 — PWM output channel 1
				I		TIM2_CAP1 — Capture input, channel 1 for 16-bit timer 2
				O		TIM2_MAT1 — Match output, channel 1 for 16-bit timer 2
				I/O		SPI_MOSI — SPI master out slave in
				I		ADC_IN7 — ADC input channel 7
PB0	14	-		I/O	I; PU	PB0 — General purpose digital input/output pin
				O		PWM_OUT3 — PWM output channel 3
				I		TIM2_CAP2 — Capture input, channel 2 for 16-bit timer 2
				O		TIM2_MAT2 — Match output, channel 2 for 16-bit timer 2
PB1	15	14		I/O	I; PU	PB1 — General purpose digital input/output pin.
				O		PWM_OUT5 — PWM output channel 5
				I		TIM2_CAP3 — Capture input, channel 3 for 16-bit timer 2
				O		TIM2_MAT3 — Match output, channel 3 for 16-bit timer 2
VSS	16	-		S		Ground
VDD	17	16		S		Digital power supply
PA8	18	-		I/O	I; PU	PA8 — General purpose digital input/output pin
				O		PWM_OUT0 — PWM output channel 0
				I		TIM3_CAP3 — Capture input, channel 3 for 16-bit timer 3
				O		TIM3_MAT3 — Match output, channel 3 for 16-bit timer 3
				O		CLKOUT — clock output pin
PA9	19	17		I/O	I; PU	PA9 — General purpose digital input/output pin
				O		PWM_OUT2 — PWM output channel 2
				O		TXD0 — UART0 TXD
				I/O		I2C_SCL — I2C serial clock
PA10	20	18		I/O	I; PU	PA10 — General purpose digital input/output pin
				O		PWM_OUT4 — PWM output channel 4
				I		RXD0 — UART0 RXD
				I/O		I2C_SDA — I2C serial data
PA11	21	-		I/O	I; PU	PA11 — General purpose digital input/output pin
				I		TIM2_CAP2 — Capture input, channel 2 for 16-bit timer 2
				O		TIM2_MAT2 — Match output, channel 2 for 16-bit timer 2
PA12	22	-		I/O	I; PU	PA12 — General purpose digital input/output pin
SWDIO	23	19		I/O	I; PU	SWD debug port data
				I/O		PA13 — General purpose digital input/output pin

管脚名称	LQFP/N 32	SSOP 20	唤醒 输入	管脚 类型	初始 设置	描述
SWCLK	24	20		I/O	I; PU	SWD debug port clock
				I/O		PA14 — General purpose digital input/output pin
				O		TXD1 — Transmitter output for UART1
PA15	25	-		I/O	I; PU	PA15 — General purpose digital input/output pin
				I/O		SPI_SSEL — SPI chip select
				I		RXD1 — Receiver input for UART1
PB3	26	-		I/O	I; PU	PB3 — General purpose digital input/output pin
				I/O		SPI_SCK — SPI serial clock
PB4	27	-		I/O	I; PU	PB4 — General purpose digital input/output pin
				I/O		SPI_MISO — SPI master in slave out
				I		TIM2_CAP0 — Capture input, channel 0 for 16-bit timer 2
				O		TIM2_MAT0 — Match output, channel 0 for 16-bit timer 2
PB5	28	-		I/O	I; PU	PB5 — General purpose digital input/output pin
				I/O		SPI_MOSI — SPI master out slave in
				I		TIM2_CAP1 — Capture input, channel 1 for 16-bit timer 2
				O		TIM2_MAT1 — Match output, channel 1 for 16-bit timer 2
PB6	29	-		I/O	I; PU	PB6 — General purpose digital input/output pin
				I/O		I2C_SCL — I2C serial clock
				O		TXD1 — Transmitter output for UART1
				I		TIM3_CAP0 — Capture input, channel 0 for 16-bit timer 3
				O		TIM3_MAT0 — Match output, channel 0 for 16-bit timer 3
PB7	30	-		I/O	I; PU	PB7 — General purpose digital input/output pin
				I/O		I2C_SDA—I2C serial data
				I		RXD1 — Receiver input for UART1
				I		TIM3_CAP1 — Capture input, channel 1 for 16-bit timer 3
				O		TIM3_MAT1 — Match output, channel 1 for 16-bit timer 3
BOOT	31	1		I	I; PU	BOOT—Boot memory selection
				I/O		PC3— General purpose digital input/output pin
VSS	32	15		S		Ground
VDD	1	16		S		Digital power supply

2.5 外设管脚描述

使用 MCU 外设，需找到相应的管脚并通过 IOCONFIG 寄存器设置管脚的功能。系统复位时，主 SWD 调试端口和 RESET 复位功能是默认设置，所有其他的管脚被设定为数字 GPIO。

Table 2-2: 外设管脚选择

外设	功能名称	LQFP32	SOP28	SOP20	类型	可选管脚		
ADC	AD0	√	√	√	I	PA0		
	AD1	√	√	√	I	PA1		
	AD2	√	√	√	I	PA2		
	AD3	√	√	√	I	PA3		
	AD4	√	√	√	I	PA4		
	AD5	√	√	√	I	PA5		
	AD6	√	√	√	I	PA6		
	AD7	√	√	√	I	PA7		
TIM2	TIM2_CAP0	√	√	√	I	PA6	PB4	
	TIM2_CAP1		√	√	I	PA7	PB5	
	TIM2_CAP2	√	√		I	PB0	PA11	
	TIM2_CAP3		√		I	PB1		
	TIM2_MAT0	√	√	√	O	PA6	PB4	
	TIM2_MAT1		√	√	O	PA7	PB5	
	TIM2_MAT2	√	√		O	PB0	PA11	
	TIM2_MAT3		√		O	PB1		
TIM3	TIM3_CAP0	√	√	√	I	PA2	PB6	
	TIM3_CAP1	√	√	√	I	PA3	PB7	
	TIM3_CAP2		√	√	I	PA4	PB8	
	TIM3_CAP3	√	√		I		PA8	
	TIM3_MAT0	√	√	√	O	PA2	PB6	
	TIM3_MAT1	√	√	√	O	PA3	PB7	
	TIM3_MAT2		√	√	O	PA4	PB8	
	TIM3_MAT3	√	√		O		PA8	
PWM	PWM_OUT0	√	√		O	PA8		
	PWM_OUT1	√	√	√	O	PA7		
	PWM_OUT2	√	√	√	O	PA9		
	PWM_OUT3	√	√		O	PB0		
	PWM_OUT4	√	√	√	O	PA10		
	PWM_OUT5	√	√	√	O	PB1		
	PWM_FAULT	√	√	√	I	PA6		
UART0	RXD0	√	√	√	I	PA3	PA10	
	TXD0	√	√	√	O	PA2	PA9	
UART1	RXD1	√	√		I	PA15	PB7	
	TXD1	√	√		O	PA14	PB6	
SPI	SSEL	√	√	√	I/O	PA4	PA15	
	SCK	√	√	√	I/O	PA5	PB3	
	MISO	√	√	√	I/O	PA6	PB4	
	MOSI	√	√	√	I/O	PA7	PB5	

I2C	SCL	√	√	√	I/O	PA9	PB6	
	SDA	√	√	√	I/O	PA10	PB7	
SWD	SWCLK	√	√	√	I	PA14		
	SWDIO	√	√	√	I/O	PA13		

3 系统框图

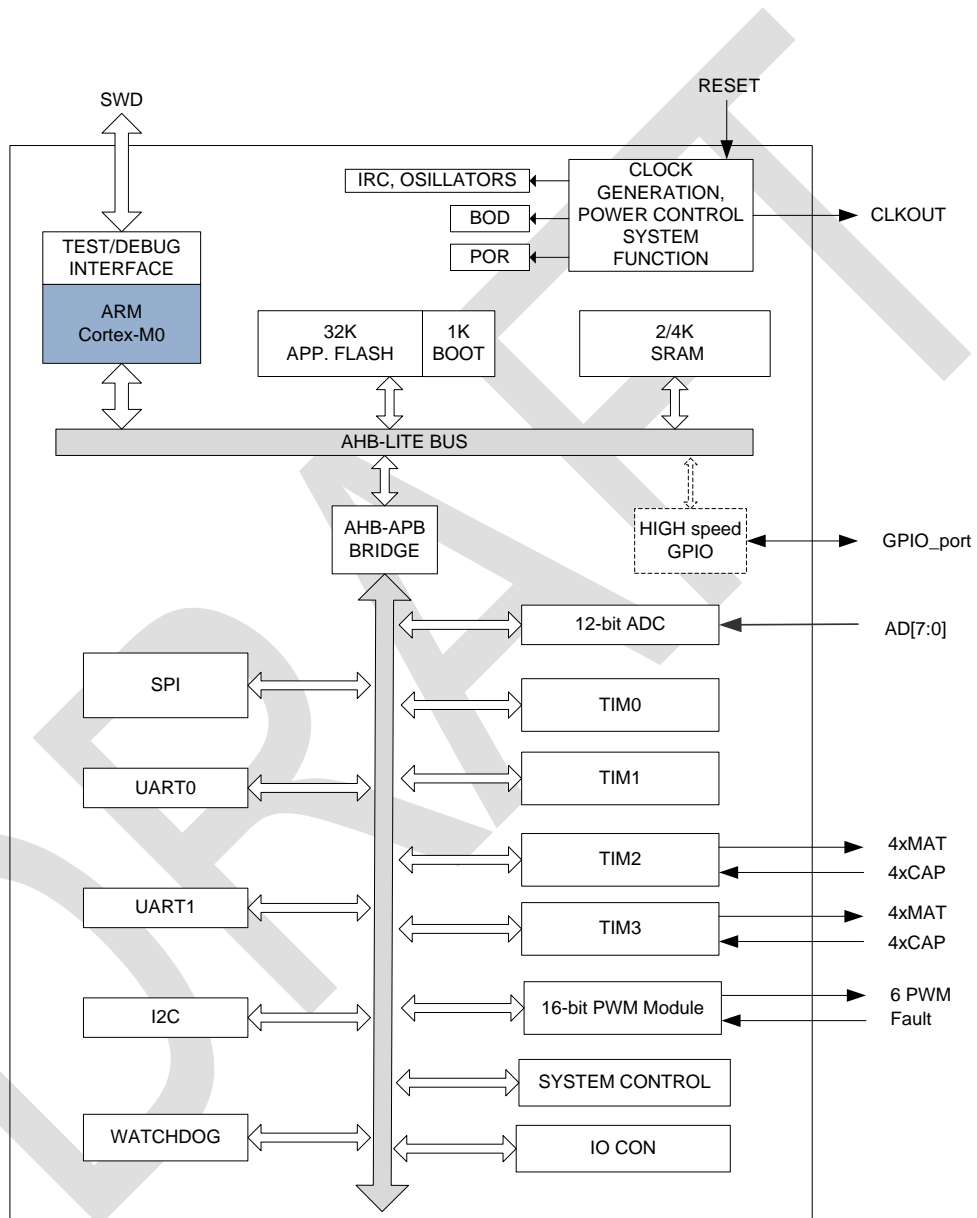


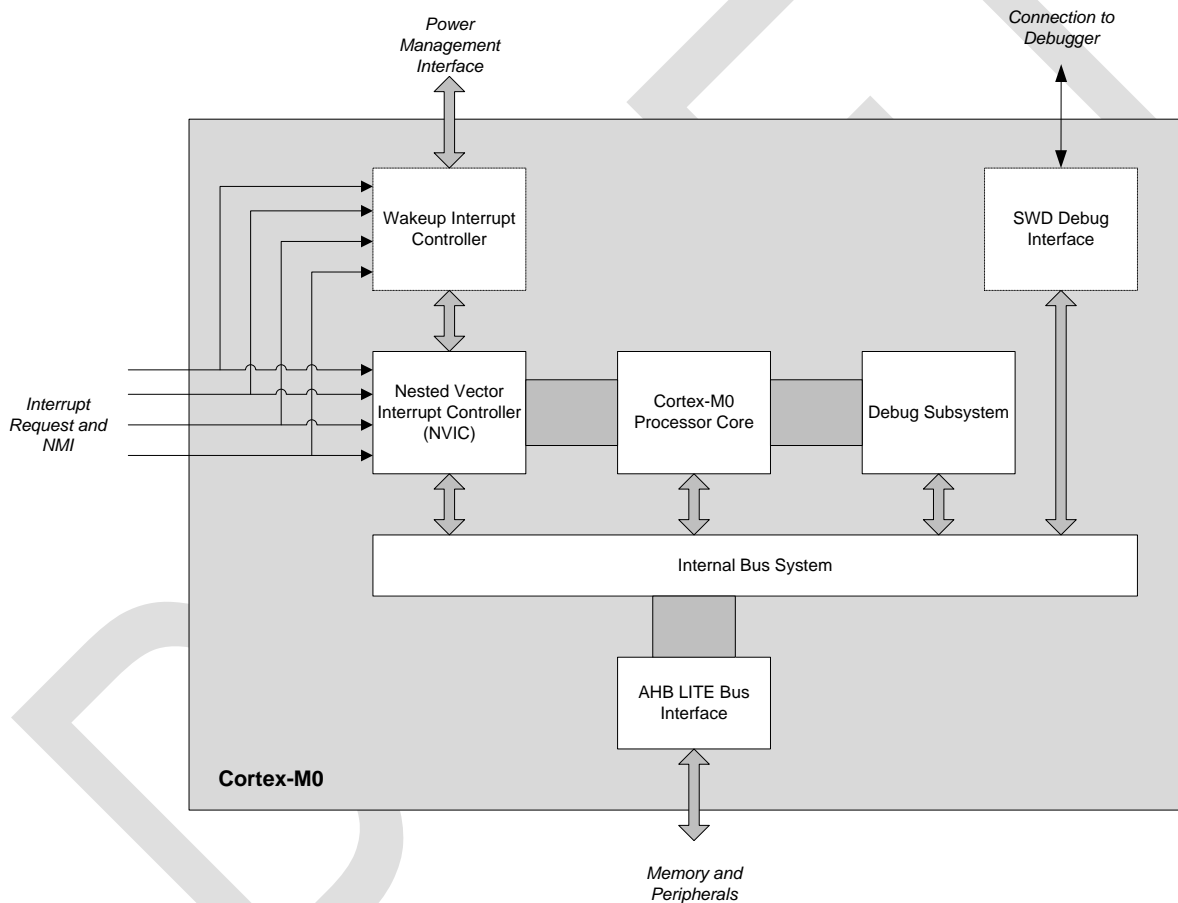
Figure 3-1: G32F0xx 模块框图

4 系统描述

4.1 ARM Cortex™-M0 内核

Cortex™ M0 处理器是 32 位可配置的多级流水线 RISC 处理器，内嵌 AMBA-Lite 接口和嵌套向量中断控制器（NVIC）。具有可选的硬件调试功能，可以执行 Thumb 指令，并与其它 Cortex-M 系列兼容。它工作在两种模式：Thread 模式和 Handler 模式。当系统异常发生时，就进入 Handler 模式。从 Handler 模式返回时，执行异常返回。复位时系统进入 Thread 模式。Thread 模式也可由异常返回时进入。

下图为M0功能模块图：



ARM Cortex™-M0 处理器功能特征：

- ARMv6-M Thumb
- Thumb-2 技术
- ARMv6-M 兼容 24-位 SysTick定时器
- 32-位硬件乘法器

- 系统接口支持小端（little-endian）数据访问
- 准确而及时的中断处理能力
- 加载、存储多个数据和多周期乘法指令可被终止然后重新开始从而实现快速中断处理
- C应用程序二进制接口的异常兼容模式（C-ABI）。ARMv6-M的模式允许用户使用纯C函数实现中断处理
- 使用中断唤醒（WFI）与事件唤醒（WFE）指令进入低功耗的休眠模式，或者从中断退出休眠模式
- NVIC特征：
 - 32 个外部中断，每个中断具有4级优先级
 - 专用的不可屏蔽中断（NMI）
 - 同时支持电平和脉冲中断触发
 - 中断唤醒控制器(WIC), 支持极低功耗休眠模式
- 调试支持
 - 四个硬件断点
 - 两个观察点
 - 用于非侵入式代码分析的程序计数采样寄存器（PCSR）
 - 单步和向量捕获能力
- 总线接口：
 - 提供简单的集成到所有系统外设和存储器的单一32位AMBA-3 ABH-Lite系统接口。
 - 支持DAP(Debug Access Port)的单一32位的从机端口。

4.2 内存映射

G32F0XX 内存地址空间支持 4GB 大小。分为：引导区，Flash 存储器区，SRAM 区，系统自有外设区，APB 外设区和 AHB 外设区。系统自有外设区保留为 M0 内核所用。

AHB 外设区占据 2M 大小空间，可以支持到 128 个外设。其中，GPIO 端口属 AHB 外设。APB 外设区是一片 512K 大小空间。每一个 APB 外设都分配 16K 空间。所有外设控制寄存器读写都采用字（word）地址寻址，用户不可以单独访问寄存器高位字节或低位字节。

引导区有 1KB 大小空间用于存储引导程序，ISP 和 IAP 函数。G32F0xx 允许的用户 Flash 空间可多达 32KB，SRAM 空间达 4KB。Flash 和 SRAM 数据总线都采用 32 位操作方式。

下图是 G32F0xx 内存地址空间分配图：

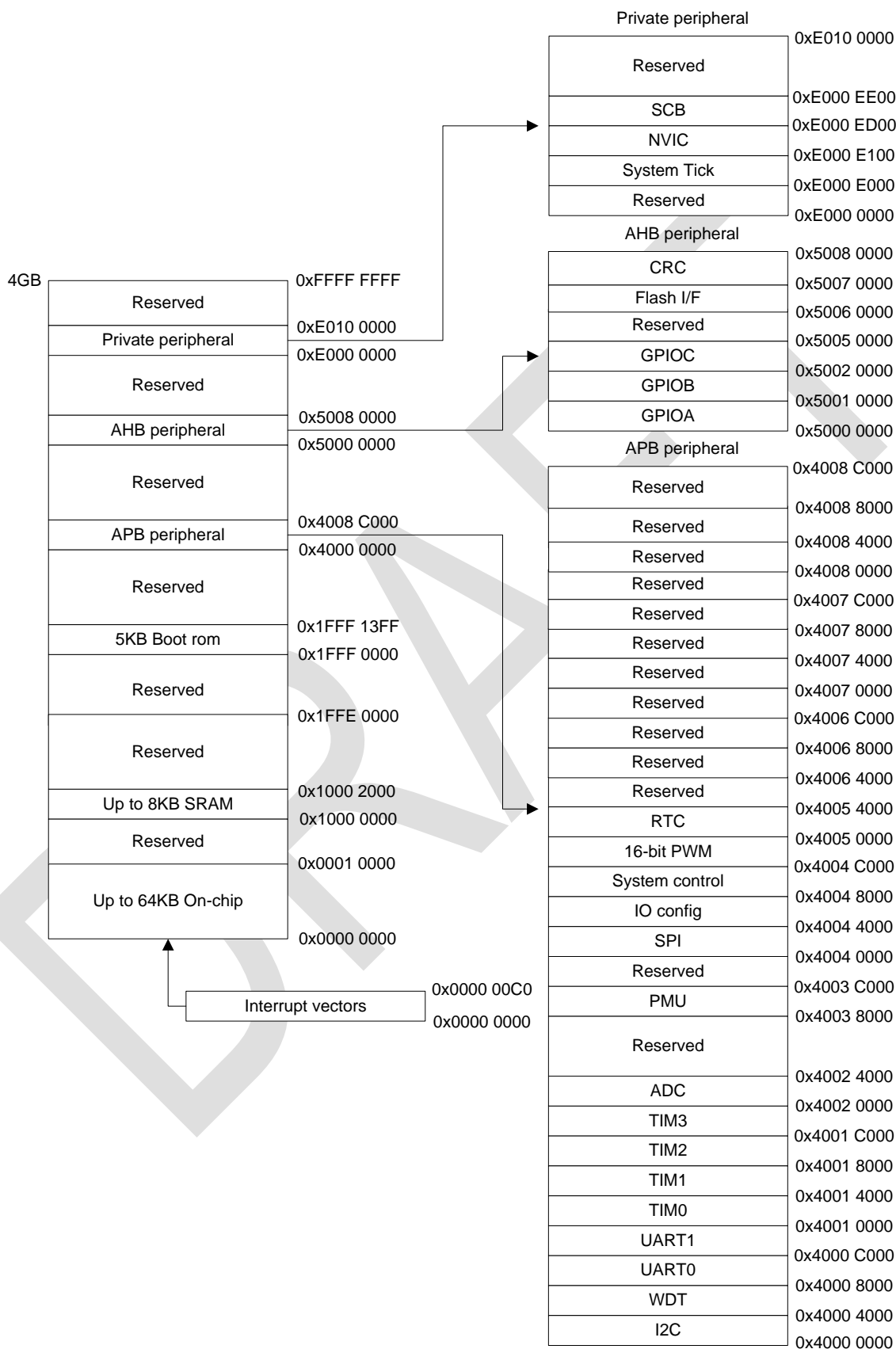


Figure 4-2: 内存分配图

Table 4-1: 内存地址分配表

地址	功能	描述
0x00000000 ~ 0x00015FFF	Flash 存储器	
0x10000000 ~ 0x10001FFF	SRAM 存储器	
0x10002000 ~ 0x1FFEFFFF	保留	
0x1FFF0000 ~ 0x1FFF1FFF	引导程序 (Bootloader)	
0x40000000 ~ 0x40003FFF	I2C	
0x40004000 ~ 0x40007FFF	看门狗	
0x40008000 ~ 0x4000BFFF	串口 (UART) 0	
0x4000C000 ~ 0x4000FFFF	串口 (UART) 1	
0x40010000 ~ 0x40013FFF	16-位定时器 0	
0x40014000 ~ 0x40017FFF	16-位定时器 1	
0x40018000 ~ 0x4001BFFF	32-位定时器 0	
0x4001C000 ~ 0x4001FFFF	32-位定时器 1	
0x40020000 ~ 0x40023FFF	ADC0	
0x40024000 ~ 0x40037FFF	保留	
0x40038000 ~ 0x4003BFFF	电源管理 (PMU)	
0x4003C000 ~ 0x4003FFFF	保留	
0x40040000 ~ 0x40043FFF	SPI	
0x40044000 ~ 0x40047FFF	管脚配置 (IOCONFIG)	
0x40048000 ~ 0x4004BFFF	系统控制	
0x4004C000 ~ 0x4004FFFF	16-bit PWM	
0x40050000 ~ 0x40053FFF	实时时钟 (Real Timer Clock)	
0x40054000 ~ 0x40063FFF	保留	
0x40064000 ~ 0x40067FFF	保留	
0x40068000 ~ 0x4006BFFF	保留	
0x4006C000 ~ 0x4007BFFF	保留	
0x4007C000 ~ 0x4FFFFFFF	保留	
0x50000000 ~ 0x5000FFFF	GPIOA	
0x50010000 ~ 0x5001FFFF	GPIOB	
0x50020000 ~ 0x5004FFFF	GPIOC	
0x50050000 ~ 0x5005FFFF	保留	
0x50060000 ~ 0x5006FFFF	Flash I/F	
0x50070000 ~ 0x5007FFFF	保留	
0x50080000 ~ 0xFFFFFFFF	保留	
0xE000E000 ~ 0xE000E0FF	系统定时器 (System Tick Timer)	
0xE000E100 ~ 0xE000E4FF	中断控制 (NVIC)	
0xE000ED00 ~ 0xE000EE00	M0 内核 SCB 控制	

4.3 Cortex™-M0 内核控制模块

4.3.1 概述

Cortex-M0 内核控制模块 (SCB) 包括了 M0 内核相关信息和系统控制，如系统配置、控制、异常报告。下表列出 SCB 模块寄存器：

Table 4-2: Cortex M0 内核 SCB 寄存器一览 (基地址： 0xE000-ED00)

名称	读写	偏移地址	描述	初始值
CPUID	RO	0x00	M0 内核信息	0x410CC200
ICSR	RW	0x04	NMI 中断控制和状态寄存器	0x00000000
AIRCR	RW	0x0C	应用程序中断控制和复位寄存器	0xFA050000
SCR	RW	0x10	系统控制寄存器	0x00000000
CCR	RO	0x14	配置和控制寄存器	0x00000204
SHPR2	RW	0x1C	系统中断处理优先级寄存器 2	0x00000000
SHPR3	RW	0x20	系统中断处理优先级寄存器 3	0x00000000

4.3.2 CPUID 寄存器

CPUID 寄存器包含 M0 内核处理器型号、版本和相关信息。

Table 4-3: CPUID 寄存器 (CPUID, 地址： 0xE000-ED00) 描述

位	名称	描述
3:0	REVISION	版本
15:4	PARTNO	处理器型号: 0xC20 = Cortex™-M0
19:16	CONSTANT	处理器架构: 0xC = ARMv6-M 结构
23:20	-	保留
31:24	IMPLEMENTER	开发商代码: 0x41 = ARM

4.3.3 中断控制和状态寄存器

中断控制和状态寄存器(ICSR) 用于标记和屏蔽可以屏蔽的中断

Table 4-4: 中断控制和状态寄存器(ICSR, 地址： 0xE000-ED04) 描述

位	名称	读写	描述
5:0	VECTACTIVE	RO	当前活动异常代码: 0 = Thread 模式 非 0 = 当前活动异常代码. 注: 当前值减 16 便是 CMSIS IRQ 中断号, 对应到中断允许寄存器, 中断禁止寄存器, 中断清除挂起寄存器, 中断挂起寄存器和中断优先级寄存器
11:6	-	-	保留.

17:12	VECTPENDING	RO	最高优先级挂起中断异常代码: 0 = 无挂起中断异常 非 0 =最高优先级挂起中断异常代码.
21:18	-	-	保留.
22	ISRPENDING	RO	中断挂起标识, 不包含 NMI 和故障: 0 = 没有挂起的中断 1 = 有中断正在挂起.
24:23	-	-	保留.
25	PENDSTCLR	WO	系统时钟 (SysTick) 异常清除挂起控制位, 写: 0 = 无效 1 =清除系统时钟中断挂起。 只写位。
26	PENDSTSET	RW	系统时钟 (SysTick) 异常允许挂起控制位. 写: 0 =无效 1 = 修改系统时钟 (SysTick) 异常到挂起状态 读: 0 =系统时钟 (SysTick) 异常不在挂起状态 1 =系统时钟 (SysTick) 异常在挂起状态.
27	PENDSVCLR	WO	PendSV 异常清除挂起控制位, 写: 0 =无效 1 =清除 PendSV 中断挂起.
28	PENDSVSET	RW	PendSV 异常允许挂起控制位. 写: 0 =无效 1 = 强制 PendSV 异常到挂起状态 读: 0 = PendSV 异常不在挂起状态 1 = PendSV 异常在挂起状态.
30:29	-	-	保留.
31	NMIPENDSET	RW	NMI 异常允许挂起控制位. 写: 0 =无效 1 = 强制 NMI 异常到挂起状态 读: 0 = NMI 异常不在挂起状态 1 = NMI 异常在挂起状态 因为 NMI 是最高优先级中断, 在正常情况下当此位变成 1 时, 处理器会立即进入中断处理程序并清除此位。如在中断处理程序中检测到此位为 1, 即意味着 NMI 中断异常再次发生。

4.3.4 应用程序中断控制和复位寄存器

应用程序中断控制和复位寄存器(AIRCR) 包含数据读写大小端控制和系统重启控制。

Table 4-5: 应用程序中断控制和重启寄存器(AIRCR, 地址: 0xE000-ED0C) 描述

位	名称	读写	描述
0	-	-	保留。
1	VECTCLRACTIVE	WO	保留为调试用。此位只允许写 0。否则 CPU 行为不可预知。
2	SYSRESETREQ	WO	系统复位请求: 0 = 无效 1 = 触发系统级别复位。
14:3	-	-	保留
15	ENDIANESS	RO	数据访问大小端模式设定: 0 = 小端模式 (Little-endian) 1 = 大端模式 (Big-endian)。
31:16	VECTKEY	WO	寄存器启动钥匙: 只有写入 0x05FA 到 VECTKEY 时, 寄存器修改才有效。否则, 写入无效。

4.3.5 系统控制寄存器

系统控制寄存器(SCR) 控制系统进入/退出省电模式。

Table 4-6: 系统控制寄存器(SCR, 地址: 0xE000-ED10) 描述

位	名称	描述
0	-	保留
1	SLEEPONEXIT	标志当系统从 Handler 模式返回到 Thread 模式是否进入睡眠模式: 0 = 不进入睡眠状态当系统返回到 Thread 模式。 1 = 进入睡眠或深度睡眠当系统从中断处理程序(ISR)返回到 Thread 模式。设置此位为 1 可以防止中断驱动的应用返回到一个空的主程序中。
2	SLEEPDEEP	控制处理器在低功耗模式下进入睡眠或深度睡眠: 0 = 睡眠 1 = 深度睡眠。
3	-	保留。
4	SEVONPEND	挂起时发送事件位: 0 = 只允许中断或事件唤醒处理器, 禁止中断被排除。 1 = 允许所有中断或事件唤醒处理器, 包括禁止中断。当一个事件或中断进入挂起状态, 事件信号会把处理器从 WFE 状态中唤醒。如处理器不在等待事件唤醒, 那么这个事件将会被记录并影响下一个 WFE。 SEV 命令也会唤醒处理器。
31:5	-	保留

4.3.6 配置和控制寄存器

配置和控制寄存器(CCR) 是只读寄存器, 用于标记 Cortex™-M0 处理器特殊行为。

Table 4-7: 配置和控制寄存器(CCR, 地址: 0xE000-ED14) 描述

位	名称	描述
2:0	-	保留.
3	UNALIGN_TRP	读为 1。指示所有非对齐的数据访问将导致 HardFault 故障。
8:4	-	保留.
9	STKALIGN	读为 1。指示进入异常时堆栈按 8 字节对齐。进入异常时, 处理器使用入栈的 PSR 的位[9] 来指示栈对齐。从异常中返回时, 处理器使用这个入栈的位来恢复正确的栈对齐。
31:10	-	保留.

4.3.7 系统中断处理优先级寄存器

为提高软件效率, Cortex Microcontroller Software Interface Standard (CMSIS)简化 SCB 寄存器表示方式。在 CMSIS 中, 系统中断处理优先级数组对应到 SHPR2-SHPR3 寄存器。

Table 4-8: SHPR2 寄存器 (地址: 0xE000-ED1C) 描述

位	名称	描述
29:0	-	保留.
31:30	PRI_11	中断 11 – SVCALL 的优先级: “0” 表示最高优先级, “3” 表示最低优先级

Table 4-9: SHPR3 寄存器 (地址: 0xE000-ED20) 描述

位	名称	描述
21:0	-	保留.
23:22	PRI_14	中断 14 – PendSV 的优先级: “0” 表示最高优先级, “3” 表示最低优先级
29:24	-	保留.
31:30	PRI_15	中断 15 – SysTick 的优先级: “0” 表示最高优先级, “3” 表示最低优先级

4.4 嵌套向量中断控制器(NVIC)

4.4.1 NVIC 概述

嵌套向量中断控制器(NVIC)是 Cortex™-M0 内核的一部分。它可以让 CPU 以最短的时间对中断作出反应。主要的特征有:

- 较短的中断响应延迟.
- 处理系统异常和外设中断.
- 支持 32 个中断向量.
- 四种可编程的中断响应优先级.
- 产生软件中断.
- 可配置的不可屏蔽中断源(NMI).

下表列出所有系统和外设中断源。每一个外设可产生 1 到多个中断到中断向量控制器。每一个中断可能有多个中断源。

Table 4-10: 中断向量源和向量表

异常代码	IRQ 号	偏移地址	异常类型	优先级	描述
		0x00	SP 值		
1		0x04	Reset	-3 最高级	
2	-14	0x08	NMI	-2	
3	-13	0x0C	HardFault	-1	
10~4		0x10~0x28	保留		
11	-5	0x2C	SVCall	可配置 ⁽¹⁾	
13~12		0x30~0x34	保留		
14	-2	0x38	PendSV	可配置 ⁽¹⁾	
15	-1	0x3C	SysTick	可配置 ⁽¹⁾	
16	0	0x40	唤醒中断	可配置 ⁽²⁾	在深度睡眠模式下，PA0~PA11 产生的唤醒中断
17	1	0x44	PWM 故障中断	可配置 ⁽²⁾	
18	2	0x48	I2C	可配置 ⁽²⁾	SI 状态更改
19	3	0x4C	TIM0	可配置 ⁽²⁾	匹配 1~0
20	4	0x50	TIM1	可配置 ⁽²⁾	匹配 1 ~0
21	5	0x54	TIM2	可配置 ⁽²⁾	匹配 0~3 捕捉 0~1
22	6	0x58	TIM3	可配置 ⁽²⁾	匹配 0~3 捕捉 0~1
23	7	0x5C	UART0	可配置 ⁽²⁾	Transmit Holding Register Empty (THRE) Transmit Holding Register overrun(THROE) Receive Buffer Full(RBRS) Receive Buffer Overrun(RBROE)
24	8	0x60	UART1	可配置 ⁽²⁾	Transmit Holding Register Empty (THRE) Transmit Holding Register overrun(THROE) Receive Buffer Full(RBRS) Receive Buffer Overrun(RBROE)
25	9	0x64	ADC	可配置 ⁽²⁾	
26	10	0x68	WDT	可配置 ⁽²⁾	看门狗中断 (WDINT)
27	11	0x6C	BOD	可配置 ⁽²⁾	掉电检测中断
28	12	0x70	PIO0	可配置 ⁽²⁾	GPIO 端口 0 中断状态
29	13	0x74	PIO1	可配置 ⁽²⁾	GPIO 端口 1 中断状态
30	14	0x78	PIO2	可配置 ⁽²⁾	GPIO 端口 2 中断状态
31	15	0x7C	RTC	可配置 ⁽²⁾	RTC interrupt
32	16	0x80	SPI	可配置 ⁽²⁾	Tx FIFO 半空 Rx FIFO 半满 Rx 超时

					Rx 溢出
33	17	0x84	PWM_Reload	可配置 ⁽²⁾	

注:

(1) 请参照 SCB 控制寄存器 SHPR2-SHPR3.

(2) 请参照 NVIC 的 IPR0~IPR7.

4.4.2 嵌套向量中断控制器(NVIC)寄存器列表

下列 NVIC 寄存器可用于中断 IRQ0~IRQ31, 包括中断使能, 等待和优先级。

Table 4-11: NVIC 寄存器列表

名称	读写	地址	描述	初始值
INTNMI	R/W	0x4004 8174	非屏蔽中断源 (NMI)配置	0x00000000
ISER	R/W	0xE000 E100	中断允许寄存器	0x00000000
ICER	R/W	0xE000 E180	中断禁止寄存器	0x00000000
ISPR	R/W	0xE000 E200	中断挂起寄存器	0x00000000
ICPR	R/W	0xE000 E280	中断清除挂起寄存器	0x00000000
IPR0	R/W	0xE000 E400	中断优先级寄存器 IPR0	0x00000000
IPR1	R/W	0xE000 E404	中断优先级寄存器 IPR1	0x00000000
IPR2	R/W	0xE000 E408	中断优先级寄存器 IPR2	0x00000000
IPR3	R/W	0xE000 E40C	中断优先级寄存器 IPR3	0x00000000
IPR4	R/W	0xE000 E410	中断优先级寄存器 IPR4	0x00000000
IPR5	R/W	0xE000 E414	中断优先级寄存器 IPR5	0x00000000
IPR6	R/W	0xE000 E418	中断优先级寄存器 IPR6	0x00000000
IPR7	R/W	0xE000 E41C	中断优先级寄存器 IPR7	0x00000000

4.4.3 中断允许寄存器

中断允许寄存器(ISER)用于允许中断设置, 同时可返回当前允许中断设置。

Table 4-12: 中断允许寄存器(ISER, 地址: 0xE000 E100) 描述

位	名称	描述	初始值
31:0	SETENA	IRQ0~IRQ31 允许中断位。 写: 0 = 无效 1 = 允许中断 读: 0 = 中断禁止 1 = 中断允许	0x00000000

如果中断被允许, 并且相应中断挂起被设置, NVIC 将会根据中断优先级触发中断。反之, 中断被禁止, 中断源只会改变中断挂起状态, 而 NVIC 不会对中断源信号采取任何动作, 不论中断优先级任何。

4.4.4 中断禁止寄存器

中断禁止寄存器(ICER) 用于禁止中断，同时可返回当前允许中断设置。

Table 4-13: 中断禁止寄存器(ICER, 地址: 0xE000 E180)描述

位	名称	描述	初始值
31:0	CLRENA	IRQ0~IRQ31 中断禁止位 写: 0 = 无效 1 = 禁止中断 读: 0 = 中断禁止 1 = 中断允许	0x00000000

4.4.5 中断挂起寄存器

中断挂起寄存器(ISPR) 强制中断进入挂起状态，同时可返回当前中断挂起设置。

Table 4-14: 中断挂起寄存器(ISPR, 地址: 0xE000 E200) 描述

位	名称	描述	初始值
31:0	SETPEND	IRQ0~IRQ31 中断挂起位。 写: 0 = 无效 1 =强制中断进入挂起状态 读: 0 = 没有中断挂起 1 = 中断挂起	0x00000000

4.4.6 清除中断挂起寄存器

清除中断挂起寄存器(ICPR)用于清除中断的挂起状态或返回中断的挂起状态

Table 4-15: 清除中断挂起寄存器(ICPR, 地址: 0xE000 E280) 描述

位	名称	描述	初始值
31:0	CLRPEND	中断 IRQ0~IRQ31 清除挂起控制位。 写: 0 = 无效 1 = 清除该中断挂起状态 读: 0 = 该中断不在挂起状态 1 = 该中断处于挂起状态	0x00000000

4.4.7 中断优先级寄存器

中断优先级寄存器 IPR0-IPR7 提供给每个中断 (IRQ0~IRQ31)2 位 4 种优先级设置。每一个寄存器包含 4 个中断的优先级。

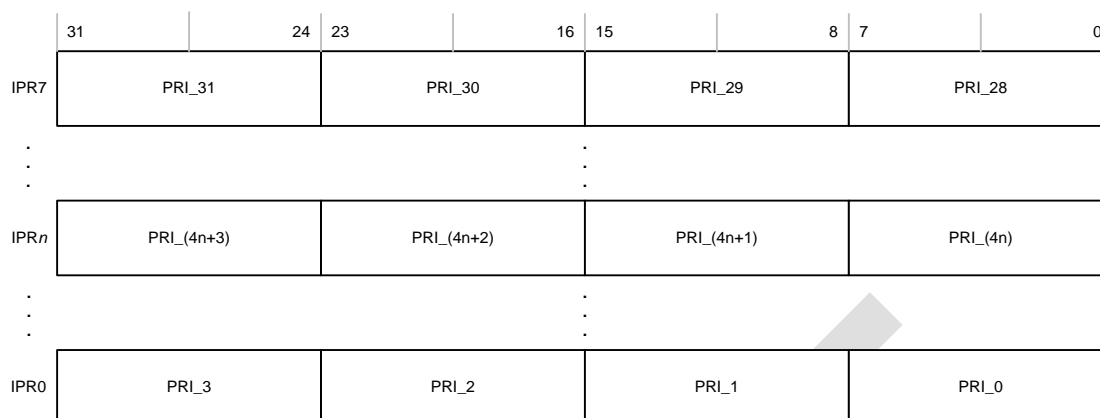


Figure 4-3: IPR 寄存器

Table 4-16: IPRn 寄存器 (IPR0~7, 地址: 0xE000 E400~0xE000 E41C) 描述

位	名称	描述	初始值
7:0	优先级, 字节偏移量 0	每个中断的优先级设定放在字节的高两位: 位 7 和位 6。 每个中断的优先级值为: 0-3。 越低值, 表示越高的优先级。 位 5~0 读为 0, 写无效	0x00
15:8	优先级, 字节偏移量 1		0x00
23:16	优先级, 字节偏移量 2		0x00
31:24	优先级, 字节偏移量 3		0x00

用下面的方法计算中断 **M** 的 IPR 寄存器号:

- 计算对应的 IPR 寄存器号, $N, N = M \text{ DIV } 4$
- 计算 IPR 寄存器内的字节偏移量 $M \text{ MOD } 4$, 其中:
 - 字节偏移量 0 对应寄存器的位 7:0
 - 字节偏移量 1 对应寄存器的位 15:8
 - 字节偏移量 2 对应寄存器的位 23:16
 - 字节偏移量 3 对应寄存器的位 31:24

4.5 系统定时器

系统定时器(SysTick)属于 Cortex™-M0 内核的一部分。它被用来产生 10 毫秒间隔中断。这个中断可以用于操作系统或系统管理软件。由于系统定时器是 Cortex™-M0 内核的一部分, 它可以作为 Cortex™-M0 器件软件的标准定时器。它有以下用途:

- 产生可编程的实时时钟 (RTOS) 触发频率 (如 100Hz) 去调用系统定时器(SysTick)处理程序
- 用内核时钟产生高速报警定时器。
- 计数器。软件可以用来测量时间
- 基于丢失/命中期限的内部时钟源。控制和状态寄存器的 COUNTFLAG 位域, 可用于动态时钟控制环的一部分, 去检测某个动作是否在设定的期限内完成。

该定时器特征为:

- 24位定时器

- 使用专用的中断向量
- 使用内部系统时钟

Table 4-18: 系统定时器寄存器列表 (基地址 0xE000 E000)

名称	读写	偏移地址	描述	初始值
SYST_CSR	R/W	0x010	系统定时器控制和状态寄存器	0x0000 0000
SYST_RVR	R/W	0x014	系统定时器重载寄存器	0x0000 0000
SYST_CVR	R/W	0x018	系统定时器当前值寄存器	0x0000 0000

4.5.1 系统定时器控制和状态寄存器

系统定时器控制和状态寄存器包含定时器控制信息和状态标识。本寄存器属于 Cortex™-M0 内核寄存器的一部分。

Table 4-17: 系统定时器控制和状态寄存器 (SYST_CSR - 0xE000 E010) 描述

位	名称	值	描述	初始值
0	ENABLE		定时器使能	0
		0	禁止定时器	
		1	使能定时器	
1	TICKINT		定时器中断使能	0
		0	定时器中断被禁止	
		1	使能定时器中断	
15:2	-		保留。	NA
16	COUNTFLAG	-	从上次读该寄存器后，如定时器计数值已为 0，则返回 1。	0
31:17	-	-	保留	NA

4.5.2 系统定时器重载寄存器

系统定时器重载寄存器用于存放系统寄存器为 0 时的重载值。该寄存器值由软件在做系统定时器初始化时设定。

Table 4-18: 系统定时器重载寄存器(SYST_RVR - 0xE000 E014) 描述

位	名称	描述	初始值
23:0	RELOAD	该值被装载到系统定时器当定时器计数到 0 时。	0
31:24	-	保留	NA

4.5.3 系统定时器当前值寄存器

系统定时器当前值寄存器返回系统定时器当前计数值。

Table 4-19: 系统定时器当前值寄存器(SYST_CVR - 0xE000 E018) 描述

位	名称	描述	初始值
23:0	CURRENT	读返回当前定时器计数值。 写任何数都会清除定时器计数值和 STCTRL 的 COUNTFLAG 位。	0
31:24		保留	NA

4.5.4 使用系统定时器

系统定时器是一个 24 位计数器。它采用向下计数并产生一个计数到 0 时的中断。一般使用固定 10ms 中断间隔。它的计数时钟信号可由 CPU 时钟（系统时钟）提供。为了产生指定间隔的循环中断，SYST_RVR 必须初始化并设置相应的值。下面给出一个 CPU 主频 20Mhz，10ms 系统定时器的例子。

举例：

1. 设置 SYST_RVR 寄存器的 RELOAD 值

$$\text{RELOAD} = (\text{系统时钟频率} \times 10 \text{ ms}) - 1$$

$$= (20\text{MHz} \times 10 \text{ ms}) - 1$$

$$= 200000 - 1$$

$$= 199999$$

$$= 0x00030D3F$$

2. 清除 SYST_CVR 寄存器值或设定一个当定时器开始工作时的起始值
3. 设置 SYST_SCR 寄存器值为 0x7，使能系统定时器和定时器中断

4.6 系统控制

G32F0xx 系统控制包括：

- 系统复位
- 系统内存的重映射
- 时钟控制
- 电源管理
- 系统级的外设控制

Table 4-20: 系统控制寄存器列表

名称	读写	偏移地址	描述	Reset value
Base:0x4003 8000				
PCON	R/W	0x000	电源控制寄存器	0x0000 0000
-	-	0x004 - 0x010	Reserved	-
RTCCFG	R/W	0x014	RTC 时钟控制	0x0000 0000
Base:0x4004 8000				
SYSMEMREMAP	R/W	0x000	内存重映射寄存器	0x0000 0000
PRESETCTRL	R/W	0x004	外设复位控制寄存器	0x000F FFFF
-	-	0x008 - 0x01C	保留	-
WDTOSCCTRL	R/W	0x024	看门狗振荡器控制寄存器	0x0000 0000
IRCCTRL	R/W	0x028	内部振荡器（IRC）控制寄存器	0x0000 0080
-	-	0x02C	保留	-
SYSRESSTAT	R/W	0x030	系统复位状态寄存器	0x0000 0000
-	-	0x034 - 0x06C	保留	-
MAINCLKSEL	R/W	0x070	主时钟源选择寄存器	0x0000 0000
MAINCLKUEN	R/W	0x074	主时钟源更新使能寄存器	0x0000 0000
SysAHBCLKDIV	R/W	0x078	AHB 时钟分频器	0x0000 0001
-	-	0x07C	保留	-
SysAHBCLKCTRL	R/W	0x080	AHB 时钟控制寄存器	0xFFFF FFFF
-	-	0x084 - 0x090	Reserved	-
SSPCLKDIV	R/W	0x094	SPI clock divider	0x0000 0000
UART0CLKDIV	R/W	0x098	UART0 时钟分频器	0x0000 0000
UART1CLKDIV	R/W	0x09C	UART1 时钟分频器	0x0000 0000
-	-	0x0A0 - 0x0DC	保留	-
CLKOUTCLKSEL	R/W	0x0E0	CLKOUT 时钟源选择寄存器	0x0000 0000

CLKOUTUEN	R/W	0x0E4	CLKOUT 时钟源更新使能寄存器	0x0000 0000
CLKOUTDIV	R/W	0x0E8	CLKOUT 时钟分频器	0x0000 0000
-	-	0x0EC - 0x148	保留	-
IOCONFIGCLKDIV0	R/W	0x14C	IO 输入滤波器寄存器	0x0000 0000
BODCTRL	R/W	0x150	BOD 掉电检测控制寄存器	0x0000 0000
-	-	0x154 - 0x170	保留	-
INTNMI	R/W	0x174	NMI interrupt source configuration control	0x0000 003F
-	-	0x178 - 0x1FC	保留	-
DSWAKECTL	R/W	0x200	深度睡眠唤醒控制寄存器	0x0000 0000
DSWAKEEN	R/W	0x204	深度睡眠唤醒信号使能控制寄存器	0x0000 0000
DSWAKECLR	W	0x208	深度睡眠唤醒信号复位控制寄存器	0x0000 0000
DSWAKE	R	0x20C	深度睡眠唤醒信号状态寄存器	0x0000 0000
-	-	0x210 - 0x22C	保留	-
PDSLEEPCFG	R/W	0x230	深度睡眠下的电源控制寄存器	0x0000 0003
PDWAKECFG	R/W	0x234	深度睡眠唤醒后的电源控制寄存器	0x0000 EFF0
PDRUNCFG	R/W	0x238	电源控制寄存器	0x0000 0000
-	-	0x23C	保留	

4.6.1 系统复位

下列事件可以触发系统复位:

- 上电复位 (POR)
- RESET#管脚复位信号 (低电平)
- 看门狗超时复位
- 掉电检测复位 (BOD)
- 软件复位
- 掉电模式唤醒复位

RESET#管脚是施密特触发 (Schmitt trigger) 输入管脚。一个有效的复位信号要求不低于 15us 的信号宽度。

复位可以由任意一个复位源引起, 只要工作电压处于可用电平, 就会启动 IRC 并保持有效。当外部复位无效后, 此时振荡器运行, 同时闪存控制器完成初始化。

当 POR, BOD 复位, 外部复位和看门狗复位发生后, 下列初始化会被执行:

- 启动 IRC。经过 IRC 启动时间, IRC 提供系统稳定时钟输出。
- 执行在 ROM 区的引导程序。引导程序会对系统初始化然后跳入用户程序。

当外部复位无效后, 处理器执行映射到地址 0 的引导区程序, 同时所有的处理器和外设寄存器都赋予初始值。

4.6.1.1 系统内存重映射寄存器

系统内存重映射寄存器选择从引导区、Flash 区或 SRAM 区读取 ARM 中断向量表。

Table 4-21: 系统内存重映射寄存器(SYSMEMREMAP, 地址: 0x4004 8000) 描述

位	名称	值	描述	初始值
1:0	MAP		内存重映射寄存器	00
		0x0	引导区模式。 中断向量表被映射到引导区。	
		0x1	用户 SRAM 模式。 中断向量表被映射到 SRAM 起始地址。	
		0x2	用户 Flash 模式。 中断向量表被映射到 Flash 0 地址。	
31:2	-	-	保留	0x00

4.6.1.1 系统复位状态寄存器

系统复位状态寄存器 (SYSRSTSTAT) 显示最近一次的系统复位源。写 1 清除状态位。当 POR 复位发生时, 其它状态位被清 0。如果有其它复位信号 (如 EXTRST) 保持在 POR 发生时, 这个信号会被检测到并显示在寄存器中。

Table 4-22: 系统复位状态寄存器(SYSRESSTAT, 地址: 0x4004 8030) 描述

位	名称	值	描述	初始值
0	POR		POR 复位状态	0
		0	没有 POR 复位被检测到	
		1	POR 复位检测到	
1	EXTRST		外部复位状态	0
		0	没有 RESET 复位被检测到	
		1	RESET 复位检测到	
2	WDT		看门狗复位状态	0
		0	没有 WDT 复位被检测到	
		1	WDT 复位检测到	
3	BOD		BOD 掉电复位状态	0
		0	没有 BOD 复位被检测到	
		1	BOD 复位检测到	
4	SYSRST		软件复位状态。 软件复位有同 RESET#管脚相同效果。	0
		0	没有检测到软件复位	
		1	检测到软件复位	
31:5	-	-	保留	0x00

4.6.1.2 软件复位

软件可以设置内核 AIRCR 寄存器的 SYSRESETREQ 位复位系统。参照 SCB 章节。

4.6.1.3 POR

内嵌上电复位 (POR) 模块监视电源上电状态并触发 POR 信号对 Cortex™-M0 内核和外设进行复位。

4.6.1.4 BOD

掉电检测（BOD）模块用于对器件的供电状态进行监测。它有如下两个功能：

1. 复位并保持复位当电源低于 2.624V 时。迟滞电压范围 0.135V（典型值）。
2. 触发 BOD 中断当电压低于 2.828V 时。迟滞电压范围 0.101V（典型值）。

Table 4-23: BOD 控制寄存器 (BODCTRL, 地址: 0x4004 8150)描述

位	名称	值	描述	位
3:0	-	-	保留	3:0
4	BODRSTEN		BOD 复位使能	4
		0	禁止 BOD 复位功能	
		1	使能 BOD 复位功能	
5	-	-	保留	5
6	BODINTCLR	-	写 1 清除 BOD 中断信号	6
31:5	-	-	保留	31:5

4.6.2 时钟控制

4.6.2.1 概述

与其它 MCU 比较，G32F0 具有非常灵活的时钟控制系统。用户可根据不同应用要求来配置时钟从而取得最高的性能及优化的能耗管理。下图为 G32F0 的时钟系统概要图。

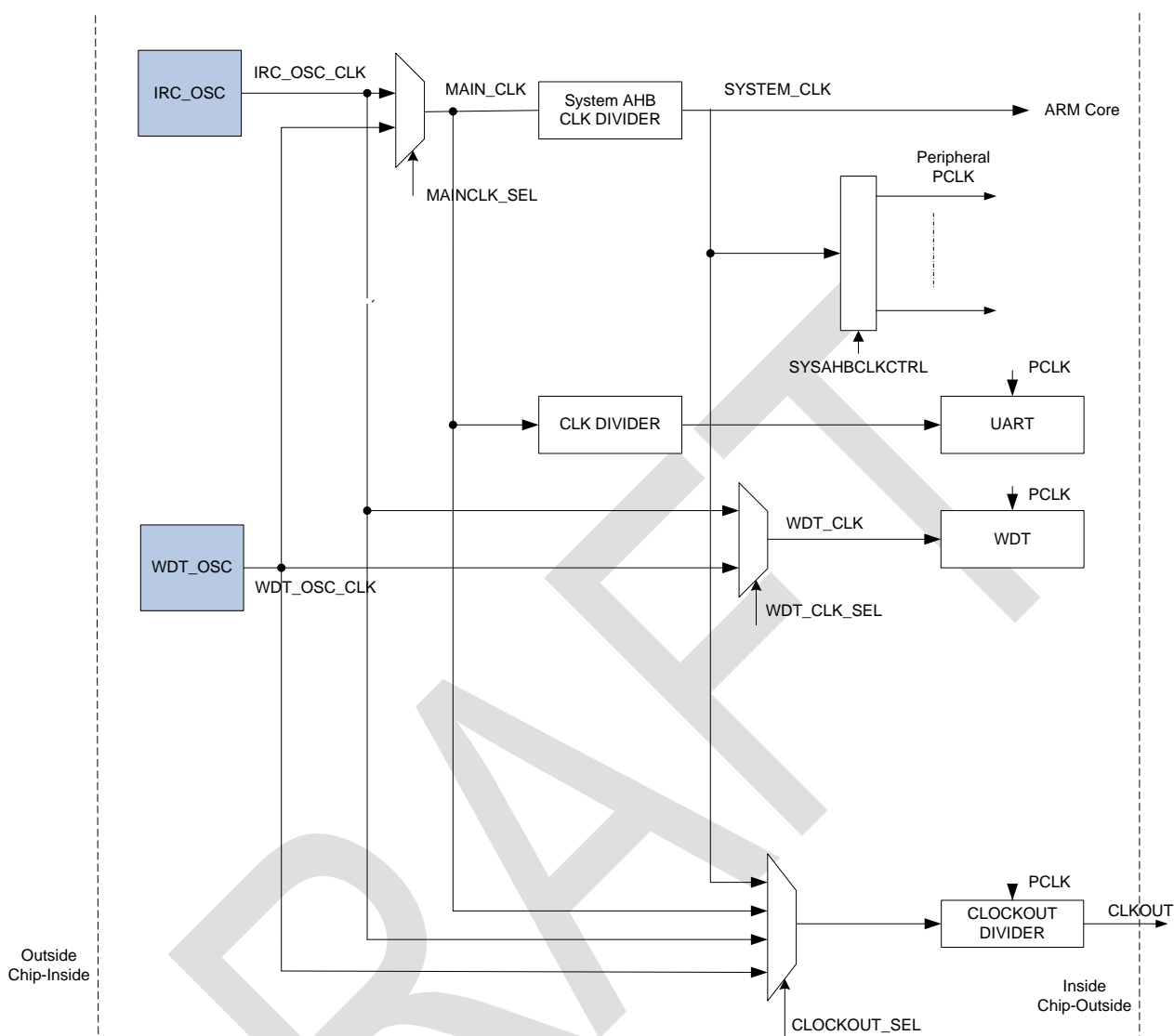


Figure 4-4: Clock System block diagram

系统复位后，G32F0 会一直采用 IRC 时钟直到用户使用软件切换到其它的时钟源。这样，系统可以在没有外部晶振时以已知的频率执行引导程序。

SYSAHBCLKCTRL 寄存器用于内存及外设时钟供给控制。UART0/1 有自己独立的时钟分频器从主时钟（main clock）取得操作时钟。看门狗操作时钟可来自于 WDT 振荡器或主时钟（main clock）。主时钟（main clock），IRC，外部晶振及看门狗振荡器时钟都可从 CLKOUT 管脚输出。

4.6.2.2 看门狗振荡器控制寄存器

此寄存器用于配置看门狗振荡器控制。振荡器包涵模拟和数字两部分。振荡器的模拟部分用于产生模拟时钟(F_{clkana})。在振荡器数字部分，模拟时钟(F_{clkana})会被重新整形分频到 WDT_CLK 输出。模拟时钟(F_{clkana})输出一个 32KHz 频率时钟。将模拟时钟(F_{clkana})进行 4 分频，然后再被 DIVSEL 控制的分频器分频输出到 WDT_CLK，作为看门狗时钟源选择（参照 WDT 章节）。

看门狗振荡器输出频率可用下列公式推算：

$WDT_CLK = F_{clkana} / (4 \times DIVSEL) = 8K\ Hz \sim 250\ Hz$ (标称值)。

注:在任何 FREQSEL 设置下, F_{clkana} 频率都有可能存在 $\pm 20\%$ 的误差。虽然看门狗振荡器为低功耗时钟源,但如果要求时钟精度的话,必须使用 IRC 或系统时钟。

注:在使用前必须对看门狗振荡器控制寄存器进行配置。

Table 4-24: 看门狗振荡器控制寄存器(WDTOSCCTRL, 地址 0x4004 8024) 描述

位	名称	值	描述	初始值
4:0	DIVSEL		针对看门狗时钟 F_{clkana} 频率的分频器。置 0 停止看门狗时钟。	0x1
7:5	CTRIM		电容调制值	0x0
11:8	RTRIM		电阻调制值	0x0
12	WDTCLKSRC		看门狗时钟源选择: 0: 内部时钟振荡器 IRC 1: 看门狗 32K 时钟	0x0
31:13	-	-	保留	0x0

4.6.2.1 内部晶体振荡器控制寄存器

该寄存器用于对 40/48MHz 内部晶振频率校准从而达到更高精度。缺省值是工厂出厂预设并由引导程序写入的。

Table 4-25: 内部晶体振荡器控制寄存器(IRCCTRL, 地址 0x4004 8028) 描述

位	名称	描述	初始值
1:0	CTRIM	电容调制值	00
4:2	RTRIM	电阻调制值	110
5	SW	40MHz/48MHz 开关, 0:40MHz, 1:48MHz	0
31:10	-	保留	0x00

4.6.2.1 RTC 配置寄存器

此寄存器控制 RTC 的时钟输入。32 kHz RTC 振荡器中有两个时钟可供选择: 1 Hz 时钟(默认值)及 1kHz 时钟。此外, 可以选择由 RTC 时钟分频器分频的主时钟派生的外围设备 RTC 时钟作为 RTC 时钟源。必须先选择 RTC 时钟源, 才能在 SYSAHBCLKCTRL 寄存器中启用 RTC。当 RTC 正在运行时, 不得更改时钟源。

Table 4-26: 系统配置寄存器(SYSCFG, 地址 0x4003 8014) 描述

位	名称	值	描述	初始值
10:0	-	-	保留。	0x0
12:11	RTCCLK		RTC 时钟源选择	0000
		00	1 Hz 时钟	
		01	IRC 时钟	
		10	1 kHz 时钟	
		11	RTC PCLK	
31:15	-	-	保留。	0x0

4.6.2.1 系统主时钟

4.6.2.1.1 主时钟源选择寄存器

该寄存器用于选择系统主时钟。系统主时钟可以是 PLL 的输入或输出时钟、看门狗振荡器时钟、内部晶振时钟。主时钟可作为内核、外设和内存的时钟源。该寄存器时钟选择生效依赖于 MAINCLKUEN 寄存器从 0 到 1 触发过程。时钟在切换之前必须稳定工作。

Table 4-27: 主时钟源选择寄存器 (MAINCLKSEL, 地址 0x4004 8070) 描述

位	名称	值	描述	初始值
1:0	SEL		主时钟源	00
		0x0	IRC	
		0x1	保留	
		0x2	WDT	
		0x3	保留	
31:2	-	-	保留	0x00

4.6.2.1.1 主时钟源更新使能寄存器

该寄存器用来当 MAINCLKSEL 寄存器时钟源被修改时使新的时钟源生效。为了让时钟源生效, MAINCLKUEN 寄存器必须先写 0, 然后写 1。

Table 4-28: 主时钟源更新寄存器(MAINCLKUEN, 地址 0x4004 8074) 描述

位	名称	值	描述	初始值
0	ENA		系统主时钟源使能	0
		0	无效	
		1	更新主时钟源	
31:1	-	-	保留	0x00

4.6.2.2 AHB 时钟控制

4.6.2.2.1 AHB 接口时钟分频器寄存器

该寄存器把主时钟分频到系统时钟供内核、外设和内存使用。寄存器 DIV 值设为 0 将导致系统时钟关闭。

Table 4-29: AHB 接口时钟分频器寄存器(SYSAHBCLKDIV, 地址 0x4004 8078) 描述

位	名称	描述	初始值
7:0	DIV	AHB 接口时钟分频器寄存器值为 0 时, 系统时钟被关闭。 1: 主时钟除 1. 到 255: 主时钟除 255.	0x01
31:8		保留	0x00

4.6.2.2.2 AHB 接口时钟控制寄存器

AHBCLKCTRL 寄存器可以对单个的系统数字模块和外设数字模块进行时钟控制。寄存器位 0 控制着下列模块所需时钟: AHB 到 APB 桥, AHB 矩阵, ARM Cortex™-M0 内核, SYSCON 模块和电源控制 (PMU) 模块。该寄存器初始值设定在 bootloader 中可能改变, 建议用户程序在使用相关外设时先打开外设时钟。

Table 4-30: AHB 接口时钟控制寄存器(SYSAHBCLKCTRL, 地址 0x4004 8080) 描述

位	名称	值	描述	初始值
0	SYS		对 AHB 到 APB 桥, AHB 矩阵, ARM Cortex™-M0 内核, SYSCON 模块和电源控制 (PMU) 模块时钟控制。	1
		0	保留	
		1	打开	
1	-		保留	NA
2	RAM		RAM 的时钟控制。	1
		0	关闭	
		1	打开	
3	-	-	保留	NA
4	FLASH		Flash 编程和擦除功能使能, Flash 读取不受影响	1
		0	关闭	
		1	打开	
5	I2C		I2C 模块时钟控制	1
		0	关闭	
		1	打开	
6	-		CRC 模块时钟控制	1
		0	关闭	
		1	打开	
7	TIM0		16 位的定时器 0 模块时钟控制	1
		0	关闭	
		1	打开	
8	TIM1		16 位的定时器 1 模块时钟控制	1
		0	关闭	
		1	打开	
9	TIM2		32 位的定时器 0 模块时钟控制	1
		0	关闭	
		1	打开	
10	TIM3		32 位的定时器 1 模块时钟控制	1
		0	关闭	
		1	打开	
11	SPI		SPI 模块时钟控制	1
		0	关闭	
		1	打开	
12	UART0		UART0 模块时钟控制	1
		0	关闭	
		1	打开	
13	UART1		UART1 模块时钟控制	1
		0	关闭	

		1	打开	
14	ADC		ADC 模块时钟控制	1
		0	关闭	
		1	打开	
15	WDT		看门狗（WDT）模块时钟控制	1
		0	关闭	
		1	打开	
16	IOCON		IO 配置模块时钟控制	1
		0	关闭	
		1	打开	
17	PWM		PWM 模块时钟控制	1
		0	关闭	
		1	打开	
18	-	-	保留	-
19	RTC		RTC 模块时钟控制	1
		0	关闭	
		1	打开	
28:20	-	-	保留	-
29	GPIOA		GPIOA 模块时钟控制	1
		0	关闭	
		1	打开	
30	GPIOB		GPIOB 模块时钟控制	1
		0	关闭	
		1	打开	
31	GPIOC		GPIOC 模块时钟控制	1
		0	关闭	
		1	打开	

4.6.2.1 UART 时钟控制

UART 时钟控制寄存器组用于对 UART0/1 外设的时钟（UARTn_PCLK）配置。当 DIV 值设为 0 时，该时钟被关闭。

Table 4-31: UARTn 时钟分频器控制寄存器 (UART0CLKDIV, 地址 0x4004 8098; UART1CLKDIV, 地址 0x4004 809C;) 描述

位	名称	值	描述	初始值
7:0	DIV		UARTn 时钟分频器值	0x0
		0x00	UARTn 时钟被关闭	
		0x01	除 1	
		~	~	
		0xFF	除 255.	
31:8		-	保留	0x00

4.6.2.1 CLKOUT 输出时钟控制

4.6.2.1.1 CLKOUT 输出时钟源选择寄存器

该寄存器用于配置输出到 CLKOUT 管脚的 `clkout_clk` 时钟信号。内部晶振（IRC），外部时钟，看门狗（WDT）振荡器，系统主时钟和 RTC 晶振都可作 `clkout_clk` 时钟。CLKOUTCLKUEN 寄存器必须触发一次低到高的变化才能使时钟源选择生效。

Table 4-32: CLKOUT 输出时钟源选择寄存器 (CLKOUTCLKSEL, 地址 0x4004 80E0) 描述

位	名称	值	描述	初始值
1:0	SEL		CLKOUT 时钟源	00
		0x0	IRC 晶振	
		0x1	系统时钟	
		0x2	看门狗振荡器输出	
		0x3	主时钟	
		0x4	RTC 晶振	
31:2	-	-	保留	0x00

4.6.2.1.1 CLKOUT 时钟源更新使能寄存器

该寄存器用来当 CLKOUTCLKSEL 寄存器时钟源被修改时使新的时钟源生效。为了让时钟源生效，CLKCLKUEN 寄存器必须先写 0，然后写 1。

Table 4-33: CLKOUT 时钟源更新使能寄存器 (CLKOUTUEN, 地址 0x4004 80E4) 描述

位	名称	值	描述	初始值
0	ENA		CLKOUT 时钟源更新使能	0
		0	无效	
		1	时钟源更新使能	
31:1	-	-	保留	0x00

4.6.2.1.2 CLKOUT 输出时钟分频器

该寄存器用于对输出到 CLKOUT 管脚的 `clkout_clk` 时钟信号进行分频。

Table 4-34: CLKOUT 输出时钟分频器寄存器 (CLKOUTDIV, 地址 0x4004 80E8) 描述

位	名称	值	描述	初始值
7:0	DIV		时钟分频除数	0
		0x00	输出时钟关闭	
		0x01	除 1	
		~	~	
		0xFF	除 255.	
31:8	-	-	保留	0x00

4.6.2.2 可编程抗干扰滤波器

可编程抗干扰滤波器可用于 IO 输入信号滤波。此数字滤波是基于主时钟的。当 DIV 设为 0 时，滤波时钟关闭。

Table 4-35: IOCONFIG 滤波器时钟寄存器 0 (IOCONFIGCLKDIV0 地址 4004 8014C) 描述

位	名称	值	描述	初始值
7:0	DIV		时钟分频除数，该设置必须为 1~255 之间的奇数。	0
		0x00	时钟关闭	
		0x01	除 1	
		0x03	除 3	
		~ 0xFF	~ 除 255	
31:8	-	-	保留	0x00

4.6.3 功耗管理

G32F0 支持多种功耗控制功能。在处理器正常工作模式下，可以对所选外设的电源和时钟进行优化，从而降低系统功耗。此外，处理器有三种特殊的低功耗模式：睡眠模式、深度睡眠模式和掉电模式。在睡眠、深度睡眠和掉电模式下，不支持 SWD 系统调试。

Table 4-36: 不同工作模式下系统模块电源和时钟状态

功能组	功能模块	工作模式		睡眠模式		深度睡眠模式		掉电模式	
		时钟	电源	时钟	电源	时钟	电源	时钟	电源
内核	M0	-	上电	停止	上电	停止	上电	停止	掉电
存储器	SRAM	-	上电	停止	上电	停止	上电	停止	掉电
	FLASH	-	上电	停止	上电	停止	上电	停止	掉电
模拟模块	IRC	-	PDRUNCFG	-	PDRUNCFG	-	OFF	-	OFF
	BOD	-	PDRUNCFG	-	PDRUNCFG	-	PDSLEEP_CFG	-	OFF
	ADC	-	PDRUNCFG	-	PDRUNCFG	-	OFF	-	OFF
	WDTOSC	-	PDRUNCFG	-	PDRUNCFG	-	PDSLEEP_CFG	-	OFF
	RTCOSC	-	PDRUNCFG	-	PDRUNCFG	-	PDSLEEP_CFG	-	PDRUNCFG
数字模块	I2C	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	CRC	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	TIM0	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	TIM1	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	TIM2	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	TIM3	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	PWM	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	SPI	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	UART0,1	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	ADC	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	WDT	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	IOCON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	RTC	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	ON
	GPIOA	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF

	GPIOB	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
	GPIOC	AHBCLKCTRL	ON	AHBCLKCTRL	ON	AHBCLKCTRL	ON	-	OFF
掉电唤醒	WAKEUP	-	-	-	-	-	-	-	ON

4.6.3.1 电源控制寄存器

电源控制寄存器选择是否进入其中一种 ARM Cortex-M0 控制的低功耗模式（睡眠模式、深度睡眠模式或掉电模式），并分别提供睡眠或深度睡眠模式、掉电模式的标志。

Table 4-37: 电源控制寄存器 (PCON, 地址 0x4003 8000) 描述

位	名称	值	描述	初始值
0	-	-	保留.	0
1	DPDEN		掉电模式使能	0
		0	ARM WFI 将进入睡眠或深度睡眠模式（ARM Cortex™-M0 内核时钟关闭）。	
		1	ARM WFI 将进入掉电模式	
7:2	-	-	保留.	0x0
8	SLEEPFLAG		睡眠模式标志	0
		0	读: 系统没有进入过睡眠、深度睡眠或掉电模式 写: 无效.	
		1	读: 系统有进入过睡眠、深度睡眠或掉电模式 写: 写 1 清除 SLEEPFLAG 标志位	
10:9	-	-	保留.	00
11	DPDFLAG		掉电标志	0
		0	读: 系统没有进入过掉电模式 写: 无效.	
		1	读: 系统有进入过掉电模式 写: 清除掉电标志	
31:12	-	-	保留.	0x0

4.6.3.1 深度睡眠配置寄存器

当系统在深度睡眠模式下，所有的模拟模块电源都被关闭。但 RTC 晶振，看门狗振荡器和 BOD 检测可以按系统要求上电工作。此寄存器用于控制看门狗(WDT) 振荡器和 BOD 电路在器件进入深度睡眠模式后的行为。此外，WDT 振荡器的行为受到 WDMODE 寄存器中 WDLOCKCLK 位的影响。所有其它模拟模块在深度睡眠和掉电模式下电源都被关闭。

Table 4-38: 深度睡眠配置寄存器(PDSLEEPCFG, 地址 0x4004 8230) 描述

位	名称	值	描述	初始值
2:0	-		保留.	011
3	BOD_PD		深度睡眠模式下的 BOD 电源控制	0
		0	上电	
		1	掉电	
4	-		保留.	0
5	RTCOSC_PD		深度睡眠模式下的 RTC 晶振电源控制	0

		0	上电	
		1	掉电	
6	WDTOSC_PD		深度睡眠模式下的看门狗振荡器电源控制。必须在 WDMODE 寄存器中的 WDLOCKCLK 位设置成 1 之前修改。	0
		0	上电	
		1	掉电	
31:7	-	-	保留	00

4.6.3.1 唤醒配置寄存器

可通过该寄存器中的位进行编程来决定当系统从深度睡眠模式中唤醒后各个模块的上电状态。

Table 4-39: 唤醒配置寄存器(PDAWAKECFG, 地址 0x4004 8234) 描述

位	名称	值	描述	初始值
0	-		保留	0
1	IRC_PD		IRC 振荡器上电控制	0
		0	上电	
		1	掉电	
2	-	-	保留	0
3	BOD_PD		BOD 上电控制	0
		0	上电	
		1	掉电	
4	ADC_PD		ADC 上电控制	1
		0	上电	
		1	掉电	
5	RTC_PD		RTC 晶振上电控制	1
		0	上电	
		1	掉电	
6	WDTOSC_PD		看门狗振荡器上电控制	1
		0	上电	
		1	掉电	
31:7	-	Reserved	-	0

4.6.3.2 电源配置寄存器

PDRUNCFG 寄存器中的位控制各个模拟模块的电源。微控制器运行时的任何时刻都可以写入该寄存器，而且该写入操作将会立即生效。该寄存器初始值受 **bootloader** 影响。

注:下面 2 位设置效果会受到 WDT 锁定状态的影响:

- 如果选择看门狗振荡器作为 WDT 的时钟源，如果同时在 MOD 寄存器中设置位 5，则将会忽略对 PDRUNCFG 寄存器中位 6 的写入操作。

- 如果选择 IRC 作为 WDT 的时钟源, 如果同时在 MOD 寄存器中设置位 5, 则将会忽略对 PDRUNCFG 寄存器中位 0 和位 1 的写入操作。

Table 4-40: 电源配置寄存器(PDRUNCFG, 地址 0x4004 8238) 描述

Bit	Symbol	Value	Description	Reset value
0	-		保留	0
1	IRC_PD		IRC 振荡器上电控制	0
		0	上电	
		1	掉电	
2	-	-	保留	0
3	BOD_PD		BOD 上电控制	0
		0	上电	
		1	掉电	
4	ADC_PD		ADC 上电控制	1
		0	上电	
		1	掉电	
5	RTC_PD		RTC 晶振上电控制	1
		0	上电	
		1	掉电	
6	WDTOSC_PD		看门狗振荡器上电控制	1
		0	上电	
		1	掉电	
31:7	-	Reserved	-	0

4.6.3.3 正常工作模式

在正常工作模式下, ARM Cortex™-M0 内核, 内存采用系统时钟(system clock), 外设则可以是系统时钟, 也可以是专用的外设时钟。整个微处理器的电源及功耗在系统复位后由 PDRUNCFG 和 SYSAHBCLKCTRL 寄存器初始值决定。PDRUNCFG 和 SYSAHBCLKCTRL 寄存器的配置可以在工作模式下实时修改。

微处理器功耗状态可以由以下几个方面来决定:

- SYSAHBCLKCTRL 寄存器控制内存和外设是否工作。
- PDRUNCFG 寄存器控制单个模拟模块的掉电(如振荡器, ADC, BOD)从而达到省电目的。,
- 系统时钟(system clock)源可以是 IRC(初始值设定), 外部晶振和看门狗振荡器。.
- 系统时钟(system clock)频率控制由 SYSAHBCLKDIV 寄存器决定。
- 特定外设(如 UART0/1, WDT)使用单独的外设时钟。它们有自己的时钟分频器。外设时钟可以用对应的分频器设定去关闭。

4.6.3.4 睡眠模式

在睡眠模式下, ARM Cortex™-M0 内核的时钟被关闭, 所有的指令执行被挂起直到系统被重启或中断唤醒。

对于外设来说，它的时钟此时由 SYSAHBCLKCTRL 寄存器控制并可以产生中断，从而唤醒处理器重启执行指令。睡眠模式可以减少由处理器，内存和内部总线产生的系统动态功耗。处理器的状态和寄存器，外设寄存器，内存数据以及管脚的逻辑状态在睡眠模式下都保持不变。

睡眠模式下的功耗配置

睡眠模式下的功耗控制配置与工作模式下一样：

- 时钟系统保持工作。
- 系统时钟（system clock）频率不变，但内核时钟被关闭。
- 模拟和数字外设状态不变。

进入睡眠模式

通过下列步骤进入睡眠模式：

1. 电源控制寄存器 PCON 中的 DPDEN 位清 0。
2. ARM Cortex™-M0 SCR 寄存器中的 SLEEPDEEP 位清 0。
3. 调用 ARM Cortex™-M0 Wait-For-Interrupt (WFI) 指令。

4. 睡眠模式唤醒

5. 当中断发生或系统复位时，系统会自动从睡眠模式中唤醒。系统被中断唤醒后，系统会恢复到由 PDRUNCFG 和 SYSAHBCLKDIV 寄存器控制的电源配置状态。如系统被复位唤醒，它的电源配置会变为初始值。

4.6.3.5 深度睡眠模式

在深度睡眠模式下，处理器系统时钟（system clock）被关闭，同时所有的模拟模块（BOD，看门狗振荡器和 RTC 晶振除外）也被掉电。BOD，看门狗振荡器和 RTC 晶振的上电状态由 PDSLEEPCFG 寄存器决定。RTC 模块及 RTC 晶振在深度睡眠模式下正常工作，除非 RTC 被掉电。在深度睡眠模式下，系统可以最大限度减少由处理器，内存，外设，相关控制器及总线带来的动态功耗。当然，处理器的状态和寄存器，外设寄存器，内存数据以及管脚的逻辑状态都保持不变。

深度睡眠模式下的电源配置

深度睡眠模式下的电源配置由 PDSLEEPCFG 寄存器决定：

- 除 RTC 晶振外，看门狗振荡器是另一个可使用的时钟源。如系统需要外设来唤醒处理器，看门狗时钟需保持工作，因为它时钟源（如 IRC，外部晶振）此时均被关闭。看门狗振荡器时钟输出频率可配置至最低。
- BOD 检测电路使能可以按应用需求决定。
- 如看门狗振荡器工作，SYSAHBCLKCTRL 寄存器应关闭掉除 WDT 外的外设接口时钟，这样才能最大限度地减少功耗。
- RTC 和 RTC 晶振可以在深度睡眠模式下保持工作。

进入深度睡眠模式

通过下列步骤进入睡眠模式：

1. 电源控制寄存器 PCON 中的 DPDEN 位清 0。
2. 配置 PDSLEEPCFG 寄存器。
3. 配置唤醒后电源配置寄存器 PDAWAKECFG。

4. 配置深度掉电模式唤醒管脚:

- 如采用外部管脚唤醒系统，必须使能管脚唤醒功能，同时打开 NVIC 中的唤醒中断。
- 如使用 RTC 唤醒系统，NVIC 中 RTC 中断使能也要打开。

5. 如需外设来唤醒系统，看门狗振荡器必须上电并用 MAINCLKSEL 寄存器把主时钟切换到看门狗时钟。

6. 在 SYSAHBCLKCTRL 寄存器中，关掉除 RTC 和 WDT 时钟以外的时钟。

7. ARM Cortex™-M0 SCR 寄存器中的 SLEEPDEEP 位置 1。

8. 调用 ARM Cortex™-M0 Wait-For-Interrupt (WFI) 指令。

深度睡眠模式唤醒

下列方法可以唤醒处于深度睡眠模式下的处理器:

- 外部管脚信号。PA0 ~ PA11 可以作为深度睡眠模式唤醒管脚。
- RTC 匹配中断信号。
- 看门狗时钟中断唤醒。
- BOD 复位信号。要使 BOD 复位信号工作，PDSLEEP_CFG 寄存器中 BOD 使能和 BODCTRL 寄存器中复位使能必须打开。
- 看门狗时钟复位。在这种情况下，看门狗振荡器需保持工作（参照 PDSLEEP_CFG 寄存器），并且 SYSAHBCLKCTRL 寄存器中的 WDT 时钟使能也要打开。
- 外部复位（Reset）管脚。

使用外部管脚唤醒系统

在 ARM 内核中，深度睡眠退出是由中断促成的。在该系统中，PA0 ~ PA11 被连接到唤醒控制器上并当作唤醒管脚使用。用户需定义唤醒管脚工作时的沿和极性，相应的唤醒中断也要打开。更进一步，NVIC 唤醒中断使能也需打开并有相应中断处理程序。唤醒中断 0 对应 PA0 到 PA11 唤醒管脚。

深度唤醒管脚中断在正常模式下也可以工作。

使用 RTC 时钟唤醒系统

RTC 使用独立的 RTC 时钟源，可以深度睡眠下继续工作。NVIC RTC 中断使能也必须打开。

4.6.3.6 掉电模式

在掉电模式下，除了 WAKEUP 管脚之外，整个微控制器的电源和时钟都被关闭。当 WDMODE 寄存器中的 WDLOCKDP 位设置为 1 时，可以阻止微控制器进入深度掉电模式。

如果在进入掉电模式之前启用 RTC，则 RTC 和 RTC 振荡器将继续在掉电模式下运行。如果在掉电模式下不需要 RTC，可禁用 RTC，最大限度降低功耗。

在掉电模式下，SRAM 和寄存器的内容将不会被保留。

在掉电模式下，除了 WAKEUP 管脚之外，所有功能管脚维持原态。

掉电模式下的电源配置

深度掉电模式没有配置选项。除了 RTC 和 RTC 振荡器之外，所有时钟、内核和所有外设都已掉电。只有 WAKEUP 管脚和备份寄存器上电。低功耗 RTC 和 RTC 振荡器可以继续运行。

进入掉电模式

如果WDMODE 寄存器中的WDLOCKDP 位设置为0，则微控制器可以进入掉电模式。如果WDLOCKDP = 1，则微控制器必须先系统复位，然后才能进入掉电模式。

通过下列步骤进入进入掉电模式：

1. 电源控制寄存器PCON中的DPDEN位置1。
2. ARM Cortex™-M0 SCR寄存器中的SLEEPDEEP位置1。
3. 调用ARM Cortex™-M0 Wait-For-Interrupt (WFI) 指令。

掉电模式唤醒

从掉电模式唤醒会引起微控制器复位。但是，RTC 寄存器和备份寄存器的内容得以保留。G32F0 从掉电模式唤醒有以下一种方式：

- 将WAKEUP 管脚下拉到低电平去唤醒。

使用 WAKEUP 管脚唤醒掉电模式

WAKEUP 管脚拉到低电平会唤醒掉电模式，同时处理器进入系统复位处理。

下列步骤是 WAKEUP 管脚的唤醒处理过程：

1. 在WAKEUP管脚上给予一个低电平脉冲
 - 系统PMU会打开内部稳压电源。当内核电压达到正常工作电压时，系统就会复位，微控制器也将重新启动。
 - 所有寄存器都将处于复位状态。
 - 如果RTC被启用，则RTC 寄存器的内容将会被保留。
2. 启动微控制器之后，可以读取PCON 寄存器中的掉电标志，以证明复位是由掉电模式的唤醒事件引起的。
3. 清除PCON 寄存器中的深度掉电标志。
4. （可选）读取通用寄存器中储存的数据。
5. 为下一个深度掉电循环设置PMU。

4.6.4 深度睡眠模式唤醒控制

G32F0 提供一组控制寄存器对外部管脚功能进行配置，这些外部管脚可以用来唤醒处于深度睡眠的器件。

4.6.4.1 深度睡眠唤醒控制寄存器

DSWAKECTL 寄存器控制 P0 端口的 PA0 ~ PA11 管脚的唤醒功能。此寄存器用于选择管脚输入的上升或下降沿信号作相应的唤醒信号。DSWAKECTL 寄存器的 0 到 11 位对应 PA0 ~ PA11 管脚。

Table 4-41:深度睡眠唤醒控制寄存器(DSWAKECTL, 地址 0x4004 8200) 描述

位	名称	描述	初始值
0	CTLPA0	PA0 信号沿触发唤醒中断设置。 0 =下降沿 1 =上升沿	0

1	CTLPA1	PA1 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
2	CTLPA2	PA2 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
3	CTLPA3	PA3 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
4	CTLPA4	PA4 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
5	CTLPA5	PA5 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
6	CTLPA6	PA6 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
7	CTLPA7	PA7 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
8	CTLPA8	PA8 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
9	CTLPA9	PA9 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
10	CTLPA10	PA10 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
11	CTLPA11	PA11 信号沿触发唤醒中断设置。 0 = 下降沿 1 = 上升沿	0
31:12	-	保留 -	0

4.6.4.2 深度睡眠唤醒信号使能寄存器

DSWAKEEN寄存器控制管脚深度睡眠唤醒使能。

Table 4-42: 深度睡眠唤醒信号使能寄存器(DSWAKEEN, 地址 0x4004 8204) 描述

位	名称	描述	初始值
---	----	----	-----

0	ERPA0	允许 PA0 唤醒功能 0 = 禁止 1 = 允许	0
1	ERPA1	允许 PA1 唤醒功能 0 = 禁止 1 = 允许	0
2	ERPA2	允许 PA2 唤醒功能 0 = 禁止 1 = 允许	0
3	ERPA3	允许 PA3 唤醒功能 0 = 禁止 1 = 允许	0
4	ERPA4	允许 PA4 唤醒功能 0 = 禁止 1 = 允许	0
5	ERPA5	允许 PA5 唤醒功能 0 = 禁止 1 = 允许	0
6	ERPA6	允许 PA6 唤醒功能 0 = 禁止 1 = 允许	0
7	ERPA7	允许 PA7 唤醒功能 0 = 禁止 1 = 允许	0
8	ERPA8	允许 PA8 唤醒功能 0 = 禁止 1 = 允许	0
9	ERPA9	允许 PA9 唤醒功能 0 = 禁止 1 = 允许	0
10	ERPA10	允许 PA10 唤醒功能 0 = 禁止 1 = 允许	0
11	ERPA11	允许 PA11 唤醒功能 0 = 禁止 1 = 允许	0
31:12		保留	

4.6.4.3 深度睡眠唤醒信号复位寄存器

在 DSWAKECLR 寄存器中, 对一个位写入 1 可复位启动逻辑状态。为了记录启动信号, 启动逻辑会利用输入信号产生一个时钟边沿来实现。时钟边沿 (下降沿或上升沿) 可以设置从深度睡眠模式唤醒的中断。所以, 启动逻辑的状态必须在它使用之前清除。

Table 4-43: 深度睡眠唤醒信号复位寄存器(DSWAKECLR, 地址 0x4004 8208) 描述

位	名称	描述	初始值
0	RSRPA0	PA0 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
1	RSRPA1	PA1 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
2	RSRPA2	PA2 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
3	RSRPA3	PA3 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
4	RSRPA4	PA4 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
5	RSRPA5	PA5 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
6	RSRPA6	PA6 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
7	RSRPA7	PA7 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
8	RSRPA8	PA8 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
9	RSRPA9	PA9 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
10	RSRPA10	PA10 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0

11	RSRPA11	PA11 唤醒信号复位 0 = 无效. 1 = 写 1 复位唤醒信号	0
31:12	-	保留	-

4.6.4.4 深度睡眠唤醒信号状态寄存器

该寄存器反映了已启用启动信号位的状态。每一位（如果启用）均能反映启动逻辑的状态，即给定管脚是否已接收到唤醒信号。

Table 4-44: 深度睡眠唤醒信号状态寄存器(DSWAKE, 地址 0x4004 820C) 描述

位	名称	描述	初始值
0	SRPA0	PA0 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
1	SRPA1	PA1 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
2	SRPA2	PA2 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
3	SRPA3	PA3 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
4	SRPA4	PA4 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
5	SRPA5	PA5 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
6	SRPA6	PA6 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
7	SRPA7	PA7 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
8	SRPA8	PA8 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
9	SRPA9	PA9 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0

10	SRPA10	PA10 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
11	SRPA11	PA11 管脚唤醒信号状态 0 = 未接收到唤醒信号 1 = 唤醒信号挂起	0
31:12	-	保留	-

4.6.5 其它

4.6.5.1 NMI 中断源配置寄存器

该寄存器用于配置 ARM Cortex™-M0 非屏蔽中断源(NMI).

Table 4-45: NMI 中断源配置寄存器(INTNMI, 地址 0x4004 8174)描述

位	名称	值	描述	初始值
5:0	NMISRC		NMI 中断源选择.	-
		0	唤醒中断 (wake-up	
		1	PWM 故障	
		2	I2C	
		3	TIM0	
		4	TIM1	
		5	TIM2	
		6	TIM3	
		7	UART0	
		8	UART1	
		9	ADC	
		10	WDT	
		11	BOD	
		12	GPIOA	
		13	GPIOB	
		14	GPIOC	
		15	RTC	
		16	SPI	
		17	PWM Reload	
		18	保留	
		19	保留	
		20	保留	
		62~21	保留	
		63	NMI 禁止	
31:6	-	-	保留	0x0

4.6.5.1 外设复位控制寄存器

该寄存器提供片上外设复位功能，可以允许软件对每一个外设单独进行复位。如寄存器位的值为 0，外设便处于复位状态。寄存器位的值置 1，外设复位解除。

Table 4-46: 外设复位控制寄存器(PRESETCTRL, 地址 0x4004 8004) 描述

位	名称	值	描述	初始值
0	SPI_RST_N		SPI 复位控制	1
		0	SPI 处于复位状态	
		1	SPI 复位解除	
1	I2C_RST_N		I2C 复位控制	1
		0	I2C 处于复位状态	
		1	I2C 复位解除	
2	UART0_RST_N		UART0 复位控制	1
		0	UART0 处于复位状态	
		1	UART0 复位解除	
3	UART1_RST_N		UART1 复位控制	1
		0	UART1 处于复位状态	
		1	UART1 复位解除	
4	UART1_RST_N		UART2 reset control	1
		0	UART2 reset enabled	
		1	UART2 reset de-asserted	
5	TIM0_RST_N		16 位计数器/定时器 0 (TIM0) 复位控制	1
		0	TIM0 处于复位状态	
		1	TIM0 复位解除	
6	TIM1_RST_N		16 位计数器/定时器 1 (TIM1) 复位控制	1
		0	TIM1 处于复位状态	
		1	TIM1 复位解除	
7	TIM2_RST_N		16 位计数器/定时器 0 (TIM2) 复位控制	1
		0	TIM2 处于复位状态	
		1	TIM2 复位解除	
8	TIM3_RST_N		16 位计数器/定时器 1 (TIM3) 复位控制	1
		0	TIM3 处于复位状态	
		1	TIM3 复位解除	
9	PWM_RST_N		PWM 复位控制	1
		0	PWM 处于复位状态	
		1	PWM 复位解除	
10	CRC_RST_N		CRC 复位控制	1
		0	CRC 处于复位状态	
		1	CRC 复位解除	
11	-	-	-	-

12	ADC_RST_N		ADC 复位控制	1
		0	ADC 处于复位状态	
		1	ADC 复位解除	
13	-		-	-
31:14	-	-	保留	NA

4.7 I/O 配置

为实现管脚多路复用，G32F0xx 为每一个管脚设计了一个 IOCON 配置寄存器用于对管脚功能定义。I/O 配置寄存器控制管脚的电气特性。可配置下列功能：

- 管脚功能
- 管脚模式：内部上拉电阻使能/禁止
- 管脚驱动
- ADC 管脚的模拟输入或数字模式切换
- IO 管脚抗干扰滤波

4.7.1 IOCON 寄存器描述

下表列出该寄存器所有位控制分配（PA10 和 PA11 除外）。

Table 4-47: IOCON 寄存器（I2C 管脚除外）

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		000	选择功能 0 (初始值).	
		001	选择功能 1.	
		010	选择功能 2.	
		011	选择功能 3.	
		100	选择功能 4.	
		101	选择功能 5.	
		110	选择功能 6.	
		111	保留.	
3	PDE		选择管脚模式 (下拉电阻控制).	0
		0	无下拉电阻	
		1	使能下拉电阻	
4	PUE		选择管脚模式 (上拉电阻控制).	1
		0	无上拉电阻	
		1	使能上拉电阻	
5	CSE		Schmitt 触发控制.	1
		0	禁止	

		1	使能	
6	INV		输入取反	0
		0	无输入取反	
		1	输入取反	
7	SRM		Slew rate mode	0
		0	fast slew rate mode.	
		1	slow slew rate mode.	
8	ADM		模拟/数字功能选择	0
		0	选择数字功能	
		1	选择模拟功能	
9	DRV		驱动电流	1
		0	选择高电流模式	
		1	选择低电流模式	
10	OD		开漏模式	0
		0	禁止	
		1	已使能开漏模式。 注：这不是真正的开漏模式。输入不能上拉至超过 $V_{DD(I/O)}$ 。	
11	S_MODE		采样模式	0
		0	绕过输入滤波器	
		1	采样 1 个滤波器时钟周期。不足 1 个滤波器时钟的输入脉冲将被抑制。	
12	IEN		输入控制	0
		0	使能	
		1	禁止	
31:13	-	-	保留	0

4.7.1.1 管脚功能

IOCON 寄存器中的 FUNC 位可设为 GPIO (FUNC = 000) 或外设功能。如果将管脚配置为 GPIO 管脚，则 DIR 寄存器决定管脚是配置为输入还是输出。对于任何外设功能，会根据管脚的功能自动控制管脚方向。GPIO_nDIR 寄存器对外设功能无效。

4.7.1.2 管脚模式

IOCON 寄存器的 MODE 位允许为每个管脚使能或禁止片内上拉电阻。默认情况下，所有管脚的上拉电阻都被使能。

4.7.1.3 管脚驱动

对于每个正常驱动管脚，可以选择两种电流的输出驱动，即低电流模式和高电流模式。

4.7.1.4 开漏模式

所有数字 I/O 管脚都可为开漏模式。该模式不是真正的开漏模式。输入不能上拉至超过 $V_{DD(I/O)}$..

4.7.1.5 可编程抗干扰滤波器

所有GPIO 管脚都配有可编程数字抗干扰滤波器。滤波器在一个可选择的时间段内抑制输入脉冲，这个时间段可短于一个滤波器时钟周期。滤波器时钟是使用IOCONFIGCLKDIV寄存器从主时钟推导得出。也可以完全绕过滤波器。

满足以下条件时， t_{pulse} 期间内任意极性的输入脉冲将被抑制：

$$t_{\text{pulse}} < t_{\text{PCLKn}}$$

4.7.2 IOCON 寄存器列表

Table 4-48: I/O 配置寄存器列表 (基地址 0x4004 4000)

名称	读写	偏移地址	描述	初始值
PA0	R/W	0x000	配置管脚 PA0	0x0000 0230
PA1	R/W	0x004	配置管脚 PA1	0x0000 0230
PA2	R/W	0x008	配置管脚 PA2	0x0000 0230
PA3	R/W	0x00C	配置管脚 PA3	0x0000 0230
PA4	R/W	0x010	配置管脚 PA4	0x0000 0230
PA5	R/W	0x014	配置管脚 PA5	0x0000 0230
PA6	R/W	0x018	配置管脚 PA6	0x0000 0230
PA7	R/W	0x01C	配置管脚 PA7	0x0000 0230
PA8	R/W	0x020	配置管脚 PA8	0x0000 0230
PA9	R/W	0x024	配置管脚 PA9	0x0000 0230
PA10	R/W	0x028	配置管脚 PA10	0x0000 0230
PA11	R/W	0x02C	配置管脚 PA11	0x0000 0230
PA12	R/W	0x030	配置管脚 PA12	0x0000 0230
PA13	R/W	0x034	配置管脚 PA13	0x0000 0230
PA14	R/W	0x038	配置管脚 PA14	0x0000 0230
PA15	R/W	0x03C	配置管脚 PA15	0x0000 0230
PB0	R/W	0x040	配置管脚 PB0	0x0000 0230
PB1	R/W	0x044	配置管脚 PB1	0x0000 0230
-	-	0x048	保留	-
PB3	R/W	0x04C	配置管脚 PB3	0x0000 0230
PB4	R/W	0x050	配置管脚 PB4	0x0000 0230
PB5	R/W	0x054	配置管脚 PB5	0x0000 0230
PB6	R/W	0x058	配置管脚 PB6	0x0000 0230
PB7	R/W	0x05C	配置管脚 PB7	0x0000 0230
-	-	0x60-0x7C	保留	-
PC0	R/W	0x080	配置管脚 PC0	0x0000 0230
PC1	R/W	0x084	配置管脚 PC1	0x0000 0230
PC2	R/W	0x088	配置管脚 PC2	0x0000 0230
PC3	R/W	0x08C	配置管脚 PC3	0x0000 0230

4.7.2.1 PA0 IOCON 寄存器

Table 4-49: PA0 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA0.	
		0x1	-	
		0x2	选择功能 ADC_IN0.	

4.7.2.2 PA1 IOCON 寄存器

Table 4-50: PA1 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA1.	
		0x1	-	
		0x2	选择功能 ADC_IN1.	

4.7.2.3 PA2 IOCON 寄存器

Table 4-51: PA2 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA2.	
		0x1	选择功能 TIM3_CAP0	
		0x2	选择功能 TIM3_MAT0	
		0x3	选择功能 TXD0	
		0x4	-	
		0x5	选择功能 ADC_IN2	

4.7.2.4 PA3 IOCON 寄存器

Table 4-52: PA3 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA3.	
		0x1	选择功能 TIM3_CAP1	
		0x2	选择功能 TIM3_MAT1	
		0x3	选择功能 RXD0	
		0x4	-	
		0x5	选择功能 ADC_IN3	

4.7.2.5 PA4 IOCON 寄存器

Table 4-53: PA4 寄存器功能位选择

位	名称	值	描述	初始值
---	----	---	----	-----

2:0	FUNC		功能选择	000
		0x0	选择功能 PA4.	
		0x1	选择功能 TIM3_CAP2.	
		0x2	选择功能 TIM3_MAT2.	
		0x3	选择功能 SPI_SSEL	
		0x4	-	
		0x5	选择功能 ADC_IN4.	

4.7.2.6 PA5 IOCON 寄存器

Table 4-54: PA5 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA5.	
		0x1	选择功能 SPI_SCK	
		0x2	-	
		0x3	选择功能 ADC_IN5.	

4.7.2.7 PA6 IOCON 寄存器

Table 4-55: PA6 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA6.	
		0x1	选择功能 PWM_FAULT	
		0x2	选择功能 TIM2_CAP0.	
		0x3	选择功能 TIM2_MAT0.	
		0x4	选择功能 SPI_MISO	
		0x5	-	
		0x6	选择功能 ADC_IN6.	

4.7.2.8 PA7 IOCON 寄存器

Table 4-56: PA7 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA7.	
		0x1	选择功能 PWM_OUT1	
		0x2	选择功能 TIM2_CAP1.	
		0x3	选择功能 TIM2_MAT1.	
		0x4	选择功能 SPI_MOSI	
		0x5	-	
		0x6	选择功能 ADC_IN7.	

4.7.2.9 PA8 IOCON 寄存器

Table 4-57: PA8 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA8.	
		0x1	选择功能 PWM_OUT0	
		0x2	选择功能 TIM3_CAP3.	
		0x3	选择功能 TIM3_MAT3.	
		0x4	-	
		0x5	选择功能 CLKOUT	

4.7.2.10 PA9 IOCON 寄存器

Table 4-58: PA9 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA9.	
		0x1	选择功能 PWM_OUT2	
		0x2	选择功能 TXD0	
		0x3	-	
		0x4	选择功能 I2C_SCL	
		0x5	-	

4.7.2.11 PA10 IOCON 寄存器

Table 4-59: PA10 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA10.	
		0x1	选择功能 PWM_OUT4	
		0x2	选择功能 RXD0	
		0x3	-	
		0x4	选择功能 I2C_SDA	
		0x5	-	

4.7.2.12 PA11 IOCON 寄存器

Table 4-60: PA11 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA11.	
		0x1	-	
		0x2	选择功能 TIM2_CAP2.	

		0x3	选择功能 TIM2_MAT2.	
		0x4	-	

4.7.2.13 PA12 IOCON 寄存器

Table 4-61: PA12 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA12.	
		0x1	-	
		0x2	-	

4.7.2.14 PA13 IOCON 寄存器

Table 4-62: PA13 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 SWDIO.	
		0x1	选择功能 PA13.	
		0x2	-	

4.7.2.15 PA14 IOCON 寄存器

Table 4-63: PA14 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 SWDCLK	
		0x1	选择功能 PA14.	
		0x2	选择功能 TXD1.	
		0x3	-	

4.7.2.16 PA15 IOCON 寄存器

Table 4-64: PA15 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PA15.	
		0x1	选择功能 SPI_SSEL	
		0x2	选择功能 RXD1.	
		0x3	-	

4.7.2.17 PB0 IOCON 寄存器

Table 4-65: PB0 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000

		0x0	选择功能 PB0.	
		0x1	选择功能 PWM_OUT3	
		0x2	选择功能 TIM2_CAP2	
		0x3	选择功能 TIM2_MAT2	
		0x4	-	

4.7.2.18 PB1 IOCON 寄存器

Table 4-66: PB1 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PB1.	
		0x1	选择功能 PWM_OUT5	
		0x2	选择功能 TIM2_CAP3	
		0x3	选择功能 TIM2_MAT3	
		0x4	-	

4.7.2.19 PB3 IOCON 寄存器

Table 4-67: PB3 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择 This pin functions as WAKEUP pin if the part is in Power-down mode regardless of the value of FUNC.	000
		0x0	选择功能 PB3.	
		0x1	选择功能 SPI_SCK.	
		0x2	-	

4.7.2.20 PB4 IOCON 寄存器

Table 4-68: PB4 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PB4.	
		0x1	选择功能 SPI_MISO.	
		0x2	选择功能 TIM2_CAP0	
		0x3	选择功能 TIM2_MAT0	
		0x4	-	

4.7.2.21 PB5 IOCON 寄存器

Table 4-69: PB5 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PB5.	
		0x1	选择功能 SPI_MOSI.	

		0x2	选择功能 TIM2_CAP1	
		0x3	选择功能 TIM2_MAT1	
		0x4	-	

4.7.2.22 PB6 IOCON 寄存器

Table 4-70: PB6 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PB6.	
		0x1	选择功能 I2C_SCL.	
		0x2	选择功能 TXD1	
		0x3	Reserved	
		0x4	选择功能 TIM3_CAP0	
		0x5	选择功能 TIM3_MAT0	
		0x6	-	

4.7.2.23 PB7 IOCON 寄存器

Table 4-71: PB7 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PB7.	
		0x1	选择功能 I2C_SDA.	
		0x2	选择功能 RXD1	
		0x3	Reserved	
		0x4	选择功能 TIM3_CAP1	
		0x5	选择功能 TIM3_MAT1	
		0x6	-	

4.7.2.24 PC0 IOCON 寄存器

Table 4-72: PC0 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PC0.	
		0x1	-	
		0x2	-	

4.7.2.25 PC1 IOCON 寄存器

Table 4-73: PC1 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000

		0x0	选择功能 PC1.	
		0x1		
		0x2	-	

4.7.2.26 PC2 IOCON 寄存器

Table 4-74: PC2 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 NRST	
		0x1	选择功能 PC2.	
		0x2	-	

4.7.2.27 PC3 IOCON 寄存器

Table 4-75: PC10 寄存器功能位选择

位	名称	值	描述	初始值
2:0	FUNC		功能选择	000
		0x0	选择功能 PC3.	
		0x1		
		0x2	-	
		0x3	-	

5 GPIO

5.1 概述

G32F0xx 提供多达 41 GPIO 管脚。主要的特点有：

- 数字管脚可以用软件定义为输入或输出
- 管脚读写可以被屏蔽
- 多个管脚的置位、清零位用一条指令实现
- 管脚的输出取反
- 每一个管脚可作为外部中断信号
- 可编程的中断触发条件及中断优先级
- 所有GPIO管脚在复位后被配置成带上拉电阻的输入管脚。

5.2 管脚描述

Table 5-1: GPIO 管脚及端口

端口	管脚	GPIO 寄存器位
GPIOA	PA0 ~PA15	15:0
GPIOB	PB0~ PB7	7:0
GPIOC	PC0~PC3	3:0

5.3 GPIO 控制寄存器

所有 GPIO 被分布到 3 个端口: Port0, Port1 和 Port2。每个端口拥有自己的控制寄存器去管理 GPIO 的功能。

Table 5-2: 寄存器概述: GPIO (基地址 port 0: 0x5000 0000; port 1: 0x5001 0000, port 2: 0x5002 0000)

名称	读写	偏移地址	描述	初始值
MASK	R/W	0x000	管脚屏蔽寄存器。受影响的寄存器有: PIN, OUT, SET, CLR, NOT。	0x0000 0000
PIN	R	0x004	管脚状态寄存器。	根据配置
OUT	R/W	0x008	管脚输出值寄存器。	0x0000 0000
SET	W	0x00C	管脚输出置位寄存器。	NA
CLR	W	0x010	管脚输出清除寄存器。	NA
NOT	W	0x014	管脚输出取反寄存器。	0x0000 0000
DIR	R/W	0x020	数据方向寄存器。	0x0000 0000
IS	R/W	0x024	中断感应寄存器。	0x0000 0000
IBE	R/W	0x028	中断边沿触发控制寄存器。	0x0000 0000
IEV	R/W	0x02C	中断事件寄存器。	0x0000 0000

IE	R/W	0x030	中断屏蔽寄存器.	0x0000 0000
RIS	R	0x034	原始中断状态寄存器.	0x0000 0000
MIS	R	0x038	中断状态寄存器.	0x0000 0000
IC	W	0x03C	中断清除寄存器.	0x0000 0000

5.3.1 GPIO 屏蔽寄存器

该寄存器可屏蔽下列寄存器的读和/或写访问：PIN、OUT、SET、CLR 和 NOT。只有相应的 MASK 寄存器中被置 0，被屏蔽寄存器中相应位的才能更改或作读值操作。将任何屏蔽位置 0 将允许对管脚的 OUT、SET、CLR 和 NOT 寄存器的写操作来更改管脚输出。管脚的当前状态可从 PIN 寄存器读取，OUT 寄存器的当前值也可被读取。将任何屏蔽位置 1 将使对管脚的 OUT、SET、CLR 和 NOT 寄存器的写操作对管脚输出影响无效。此时读操作返回 0，无论管脚的电平或 OUT 寄存器的值如何。

Table 5-3: GPIO 屏蔽寄存器 (MASK - 地址 0x5000 0000 (GPIOA), 0x5001 0000 (GPIOB), 0x5002 0000 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	MASKx		GPIO 管脚 PION_x 访问控制	0x0
		0	不屏蔽读写	
		1	屏蔽读写	

5.3.2 GPIO 管脚值寄存器

该寄存器提供已配置为执行数字功能的端口管脚的当前逻辑状态。对该寄存器的读操作将返回管脚的逻辑值，不管管脚是配置为输入还是输出，也不管它是配置为 GPIO 还是任何其它适用的备用数字功能。例如，某个特定端口管脚可能具有 GPIO 输入、GPIO 输出以及计数器/定时器匹配输出和捕获输入作为可选功能。通过 PIN 寄存器，不管其配置如何，都可读出管脚的当前逻辑状态，如可读出捕获输入状态。但有个例外：如果选择了管脚的模拟功能（如适用），则不能读取管脚状态，因为将管脚选作 ADC 输入会断开管脚的数字功能。在这种情况下，PIN 寄存器中读出的管脚值无效。读操作可通过 MASK 位屏蔽。对被屏蔽位的读操作总是返回 0，不管管脚的当前电平如何。

Table 5-4: GPIO 管脚值寄存器 (PIN - 地址 0x5000 0004 (GPIOA), 0x5001 0004 (GPIOB);0x5002 0004 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	PINx		GPIO PION_x 管脚值.	0
		0	低电平	
		1	高电平	

5.3.3 GPIO 管脚输出寄存器

向该寄存器写 0 或 1 将在相应端口管脚产生低电平或高电平。如果端口管脚配置为 GPIO 输出，将设为此值。对于所有其他配置（输入、非 GPIO 功能），OUT 寄存器位的值对管脚输出电平无效。写操作被 MASK 寄存器屏蔽。读取该寄存器将返回 GPIO 输出寄存器的内容，不管数字管脚配置和方向如何。读操作被 MASK 寄存器屏蔽。SET、CLR 和 NOT 寄存器对 OUT 寄存器执行写操作，以允许按位对单个端口管脚进行设置、清除和取反。端口输出状态只由 OUT 寄存器的内容决定。

Table 5-5: GPIO 管脚输出寄存器 (OUT - 地址 0x5000 0008 (GPIOA), 0x5001 0008 (GPIOB), 0x5002 0008 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	OUTx		GPIO PION_x 管脚输出值。	0
		0	写: GPIO 输出设为低电平。读: GPIO 输出值为低电平。	
		1	写: GPIO 输出设为高电平。读: GPIO 输出值为高电平。	

5.3.4 GPIO 管脚输出置位寄存器

该寄存器用于在通过 DIR 寄存器配置为 GPIO 输出并通过相应 IOCONFIG 寄存器配置为 GPIO 的端口管脚处产生高电平输出。写 1 会将相应端口管脚设为高电平。写 0 对 GPIO 输出电平无效。如果管脚未配置为 GPIO 和输出，SET 寄存器对管脚电平无效。该寄存器为只写寄存器。对 SET 寄存器的写操作可通过 MASK 寄存器屏蔽。

Table 5-6: GPIO 管脚输出置位寄存器 (SET - 地址 0x5000 000C (GPIOA), 0x5001 000C (GPIOB), 0x5002 000C (GPIOC)) 描述

位	名称	描述	初始值
x (31~0)	SETx	GPIO PION_x 管脚输出置位。写: 0 = 无效 1 = GPIO 管脚输出为高电平	0

5.3.5 GPIO 管脚输出清除寄存器

该寄存器用于在通过 DIR 寄存器配置为 GPIO 输出并通过相应 IOCONFIG 寄存器配置为 GPIO 的端口管脚处产生低电平输出。写 1 会将相应端口管脚设为低电平。写 0 对 GPIO 输出电平无效。如果管脚未配置为 GPIO 和输出，CLR 寄存器对管脚电平无效。该寄存器为只写寄存器。对 CLR 寄存器的写操作可通过 MASK 寄存器屏蔽。

Table 5-7: GPIO 管脚输出清除寄存器 (CLR - 地址 0x5000 0010 (GPIOA), 0x5000 1010 (GPIOB), 0x5002 0010 (GPIOC)) 描述

位	名称	描述	初始值
x (31~0)	CLEARx	GPIO PION_x 管脚输出清零。写: 0 = 无效 1 = GPIO 管脚输出为低电平	0

5.3.6 GPIO 取反寄存器

该寄存器用于将通过 DIR 寄存器配置为 GPIO 输出并通过相应 IOCONFIG 寄存器配置为 GPIO 的端口管脚的输出电平倒相。写 1 对相应端口管脚倒相。写 0 对 GPIO 输出电平无效。如果管脚未配置为 GPIO 和输出，NOT 寄存器对管脚电平无效。该寄存器为只写寄存器。对 NOT 寄存器的写操作可通过 MASK 寄存器屏蔽。

Table 5-8: GPIO 取反寄存器 (NOT - 地址 0x5000 0014 (GPIOA), 0x5001 0014 (GPIOB), 0x5002 0014 (GPIOC)) 描述

位	名称	描述	初始值
x (31~0)	NOT	GPIO PION_x 输出值倒相。写: 0 = 无效 1 = GPIO 管脚输出值倒相	0

5.3.7 GPIO 数据方向寄存器

Table 5-9: GPIO 数据方向寄存器 (DIR - 地址 0x5000 0020 (GPIOA), 0x5001 0020 (GPIOB), 0x5002 0020 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	IOx		选择 GPIO PION_x 管脚作为输入或输出	0
		0	GPIO PION_x 管脚作为输入	
		1	GPIO PION_x 管脚作为输出	

5.3.8 GPIO 中断感应寄存器

Table 5-10: GPIO 中断感应寄存器 (IS - 地址 0x5000 0024 (GPIOA), 0x5001 0024 (GPIOB), 0x5002 0024 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	ISENSEx		选择 PION_x 管脚中断感应方式	0
		0	PION_x 管脚触发沿中断	
		1	PION_x 管脚触发电平中断	

5.3.9 GPIO 中断配置寄存器

Table 5-11: GPIO 边沿中断配置寄存器 (IBE - 地址 0x5000 0028 (GPIOA), 0x5001 0028 (GPIOB), 0x5002 0028 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	IBEx		选择 PION_x 管脚上升/下降沿触发中断	0
		0	PION_x 管脚中断触发由寄存器 IEV 控制	
		1	PION_x 管脚上升和下降沿都触发中断	

5.3.10 GPIO 中断事件寄存器

Table 5-12: GPIO 中断事件寄存器 (IEV - 地址 0x5000 002C (GPIOA), 0x5001 002C (GPIOB), 0x5002 002C (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	IEVx		选择 PION_x 管脚的中断触发事件	0
		0	下降沿可触发中断。	

		1	上升沿可触发中断.	
--	--	---	-----------	--

5.3.11 GPIO 中断屏蔽寄存器

如果 IE 寄存器中的位设为高，对应的管脚就会触发各自的中断。清除该位会禁止对应管脚的中断触发。

Table 5-13: GPIO 中断屏蔽寄存器 (IE - 地址 0x5000 0030, 0x5001 0030 (GPIOB), 0x5002 0030 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	MASKx		选择是否屏蔽 PION_x 管脚触发的中断	0
		0	PION_x 中断触发被屏蔽	
		1	允许 PION_x 中断触发	

5.3.12 GPIO 原始中断状态寄存器

IRS 寄存器的位读出为高时反映了对应管脚上的原始（屏蔽之前）中断状态，表示在触发 IE 之前所有的要求都满足。位读出为 0 时表示对应的输入管脚还未启动中断。该寄存器为只读。

Table 5-14: GPIO 原始中断状态寄存器 (RIS - 地址 0x5000 0034 (GPIOA), 0x5001 0034 (GPIOB), 0x5002 0034 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	RAWSTx		原始中断中断状态	0
		0	没有中断在管脚 PION_x 上	
		1	管脚 PION_x 上存在中断条件	

5.3.13 GPIO 中断状态寄存器

MIS 寄存器中的位读为高反映了输入触发中断的状态。读出为低则表示对应的输入管脚没有中断产生，或者中断被屏蔽。MIS 是屏蔽后的中断状态。该寄存器为只读。

Table 5-15: GPIO 被屏蔽中断状态寄存器 (MIS - 地址 0x5000 0038 (GPIOA), 0x5001 0038 (GPIOB), 0x5002 0038 (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	INTSx		PION_x 引起的中断状态	0
		0	PION_x 没有触发中断	
		1	PION_x 触发中断	

5.3.14 GPIO 中断清除寄存器

Table 5-16: GPIO 中断清除寄存器 (IC - 地址 0x5000 003C, 0x5001 003C (GPIOB), 0x5002 003C (GPIOC)) 描述

位	名称	值	描述	初始值
x (31~0)	CLR _x		清除 PION_x 触发的中断	0
		0	写 0 无效	
		1	写 1 清除中断	

6 增强型串口（UART）

6.1 概述

G32F0xx 提供 2 个带 16 字节 FIFO 缓存器的 UART 外设：UART0, UART1。串行接口都支持红外传输（IrDA）协议功能。时钟都受 SYSAHBCLKCTRL 寄存器控制。同时每个 UART 有独立的时钟分频器使之不受系统时钟影响（参照 UART0CLKDIV / UART1CLKDIV）。

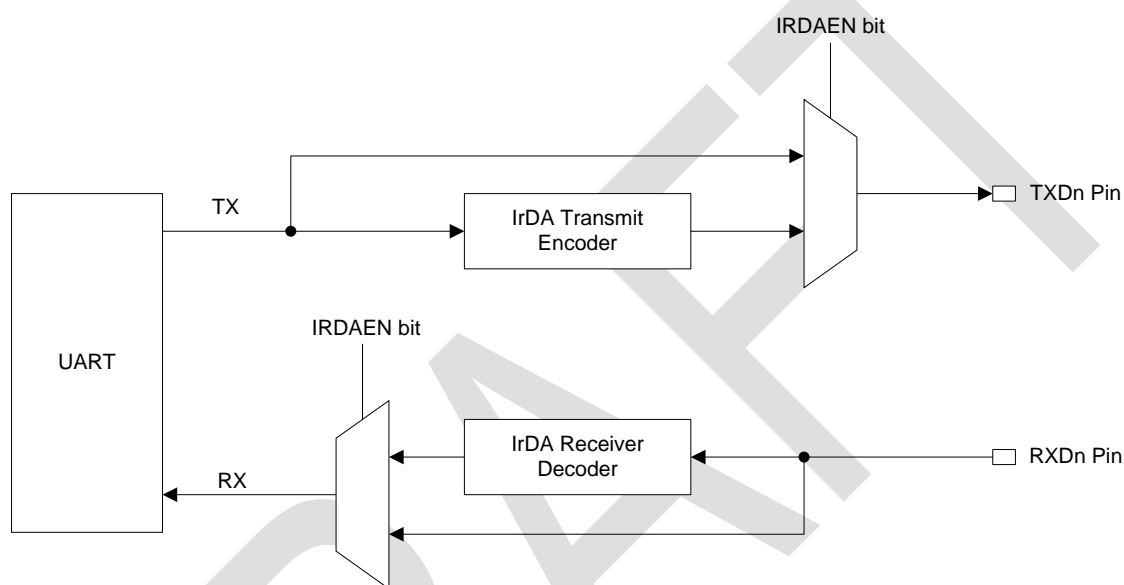


Figure 6-1: UART/IrDA 模块框图

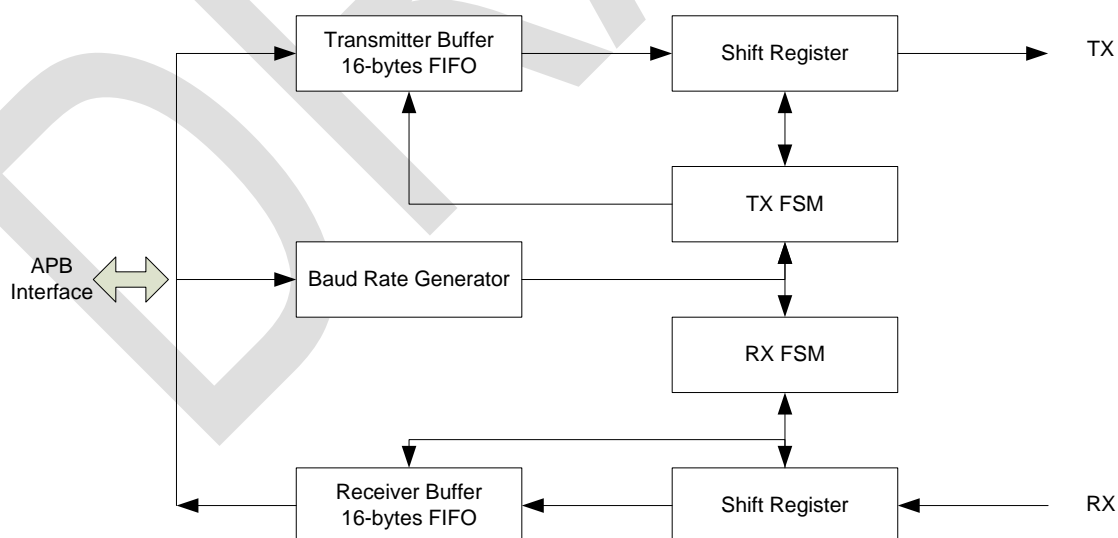


Figure 6-2: UART 功能模块图

6.2 管脚描述

Table 6-1: 管脚描述

管脚	类型	描述
RXD0	输入	UART0 输入管脚，数据接收端口。
TXD0	输出	UART0 输出管脚，数据发送端口。
RXD1	输入	UART1 输入管脚，数据接收端口。
TXD1	输出	UART1 输出管脚，数据发送端口。

6.3 UART 寄存器描述

UART 包涵的寄存器如下表：

Table 6-2: 寄存器一览 (UART0: 基地址-0x4000 8000; UART1 基地址-0x4000 c000)

名称	读写	偏移地址	描述	初始值
RDR	RO	0x000	接收缓冲寄存器。包含下一个要读取的已接收字符。	NA
TDR	WO	0x000	发送保持寄存器。在此写入下一个要发送的字符。	NA
STATE	R/W	0x004	RX 和 TX FIFO 状态	0x00
CTRL	R/W	0x008	UART 中断使能控制寄存器	0x00
INTSTATUS	R/W	0x00C	UART RX/TX 中断状态寄存器	0x0
BAUDDIV	R/W	0x010	UART 波特率控制寄存器	0x10
FIFOCLR	WO	0x014	TX 和 RX FIFO 清除寄存器	

6.3.1 接收缓冲寄存器

RDR 是 UART 接收缓冲器。它包含接收到的最后一个可以通过总线读取的字符数据。

Table 6-3: UART 接收缓冲寄存器 (RDR) 描述

位	名称	描述	初始值
7:0	RDR	UART 接收器缓冲寄存器包含的 UART 接收到的字节。	NA
31:8	-	保留	-

6.3.2 发送保持寄存器

TDR是发送保持器。只写。。

Table 6-4: UART 发送保持寄存器 (TDR) 描述

位	名称	描述	初始值
7:0	TDR	任何写入发送保持寄存器的数据都会进入 UART 的移位寄存器，然后从 UART 的 TX 数据线自动发出。	NA

31:8	-	保留	-
------	---	----	---

6.3.3 UART 状态寄存器

STATE 寄存器用于提供 UART 接收发送缓存器状态。

Table 6-5: UART 状态寄存器 (STATE) 描述

位	名称	值	描述	初始值
0	TXNE		发送 FIFO 空状态	0
		0	发送 FIFO 空，可以触发一个中断	
		1	发送 FIFO 非空	
1	RXNE		接收 FIFO 状态	0
		0	接收 FIFO 空，接收数据寄存器不可以读	
		1	接收 FIFO 非空，可以读取接收数据寄存器	
2	TXF		发送 FIFO 满状态	0
		0	发送 FIFO 非满	
		1	发送 FIFO 满	
3	RXF		接收 FIFO 满状态	0
		0	接收 FIFO 非满	
		1	接收 FIFO 满	
4	TXHLF		发送 FIFO 半满状态	0
		0	发送 FIFO 低于半满	
		1	发送 FIFO 半满	
5	RXHLF		接收 FIFO 半满状态	0
		0	接收 FIFO 低于半满	
		1	接收 FIFO 半满	
6	PARIERR		奇偶校验错误状态	0
		0	没有奇偶校验错误	
		1	接收检测到奇偶错误，写 1 清除错误标志	
7	OVERRUN		接收缓存器溢出标志	0
		0	没有缓存器溢出	
		1	缓存器溢出	

6.3.4 UART 控制寄存器

CTRL 寄存器用于 UART 使能及中断控制

Table 6-6: UART 控制寄存器 (CTRL) 描述

位	名称	值	描述	初始值
0	TXEIE		发送 FIFO 空中断允许	0
		0	禁止发送 FIFO 空中断	
		1	使能发送 FIFO 空中断	
1	RXEIE		接收 FIFO 非空中断允许	0

		0	禁止接收 FIFO 非空中断	
		1	使能接收 FIFO 非空中断	
2	TXFIE		发送 FIFO 满中断允许	0
		0	禁止发送 FIFO 满中断	
		1	使能发送 FIFO 满中断	
3	RXFIE		接收 FIFO 满中断允许	0
		0	禁止接收 FIFO 满中断	
		1	使能接收 FIFO 满中断	
4	TXHLFIE		发送 FIFO 半满中断允许	0
		0	禁止发送 FIFO 半满中断	
		1	使能发送 FIFO 半满中断	
5	RXHLFIE		接收 FIFO 半满中断允许	0
		0	禁止接收 FIFO 半满中断	
		1	使能接收 FIFO 半满中断	
6	PARIERRIE		奇偶校验错误中断使能	0
		0	禁止	
		1	允许	
7	OVERRUNIE		接收溢出中断使能	0
		0	禁止	
		1	允许	
8	PARISEL		奇偶校验选择	0
		0	选择偶校验	
		1	选择奇校验	
9	PARIEN		奇偶校验使能	0
		0	禁止	
		1	允许	
10	IRDEN		IRDEN 使能	0
		0	禁止	
		1	允许	
21-11			保留	0
22	RXEN		RX 使能	0
		0	禁止	
		1	允许	
23	TXEN		TX 使能	0
		0	禁止	
		1	允许	

31:24			保留	0
-------	--	--	----	---

6.3.5 UART 中断状态寄存器

INTSTATUS 寄存器提供 UART 当前处于等待状态的中断源。

Table 6-7: UART 中断状态寄存器 (INTSTATUS) 描述

位	名称	值	描述	初始值
0	TXEINT		发送结束中断状态。数据发送结束时产生的中断。	0
		0	中断没有挂起	
		1	中断正在挂起。写 1 清除中断挂起。	
1	RXEINT		接收中断状态。数据接收时产生的中断。	0
		0	中断没有挂起	
		1	中断正在挂起。写 1 清除中断挂起。	
2	TXFINT		发送 FIFO 满中断	0
		0	中断没有挂起	
		1	中断正在挂起。写 1 清除中断挂起。	
3	RXFINT		接收 FIFO 满中断	0
		0	中断没有挂起	
		1	中断正在挂起。写 1 清除中断挂起。	
4	TXHLFINT		发送 FIFO 半满中断	
		0	中断没有挂起	
		1	中断正在挂起。写 1 清除中断挂起。	
5	RXHLFINT		接收 FIFO 半满中断	
		0	中断没有挂起	
		1	中断正在挂起。写 1 清除中断挂起。	
6	PARIERRINT		奇偶校验错误中断	
		0	中断没有挂起	
		1	中断正在挂起。写 1 清除中断挂起。	
7	OVERRUNINT		接收溢出中断	
		0	中断没有挂起	
		1	中断正在挂起。写 1 清除中断挂起。	
31:8	-	-	保留	NA

6.3.6 UART 波特率分频器寄存器

UART 波特率分频器寄存器 (BAUDDIV) 用于时钟分频从而产生相应的波特率。该寄存器可读写。该分频器的时钟源是由 UART CLKDIV 控制 UART 的外设时钟 (PCLK)。

Table 6-8: UART 波特率分频器寄存器 (BAUDDIV) 描述

位	名称	描述	初始值
19:0	BAUDDIV	波特率分频值。最小值为 16。	0x10
31:20	-	保留。	0

波特率计算公式:

$$UART\ baudrate = UARTn_PCLK / BAUDDIV$$

UARTn_PCLK 是 UART 外设时钟。

6.3.7 UART TX/RX FIFO 数据清除寄存器

Table 6-9: UART FIFO 清除寄存器 (FIFOCLR) 描述

位	名称	描述	初始值
0	TXFIFOCLR	写 1 清除 TX FIFO 数据	0x0
1	RXFIFOCLR	写 1 清除 RX FIFO 数据	0x0
31:2	-	保留。	0

6.4 操作描述

6.4.1 UART 通讯约定

数据格式

为简化 UART 通讯控制，UART 通讯采用的是固定的数据通讯格式：1 个起始位，8 个数据位，1 个停止位，奇偶效验可选及，无硬件握手。。

波特率

用户可以设置 BAUDDIV 寄存器的分频值去产生适用不同应用的波特率。

6.4.2 IrDA 红外接收发送功能

UART 控制寄存器的 IRDAEN 位是 IrDA 的功能使能位。当 IrDA 被使能后，UART 的 RXDn/TXDn 管脚就能够以设定波特率的 3/16 脉冲宽度接收/发送红外信号。红外脉冲宽度必须是符合 IrDA 标准的超过 1.63us 信号。下图表示 UART 和 IrDA 信号的相互转换关系。

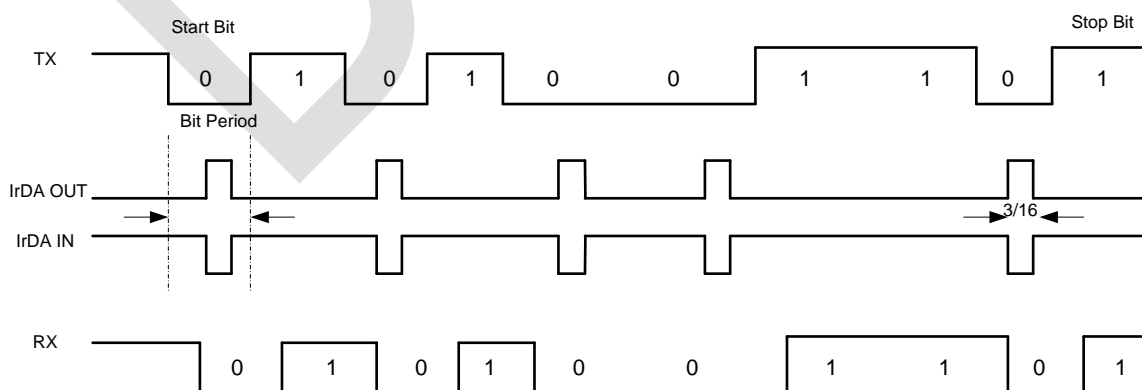


Figure 6-3: UART 和 IrDA 信号转换

7 基本型 16 位定时器/计数器 TIM0/TIM1

7.1 概述

G32F0xx 内置两个多功能的 16 位定时器/计数器。定时器/计数器工作时钟由 SYSAHBCLKDIV 寄存器控制。关闭 SYSAHBCLKDIV 寄存器中定时器/计数器的时钟供给可节省系统功耗。主要功能如下：

- 可预置分频的 16 位定时器/计数器
- 4 个 16 位匹配寄存器：
 - 可产生中断
 - 停止定时器
 - 对定时器复位

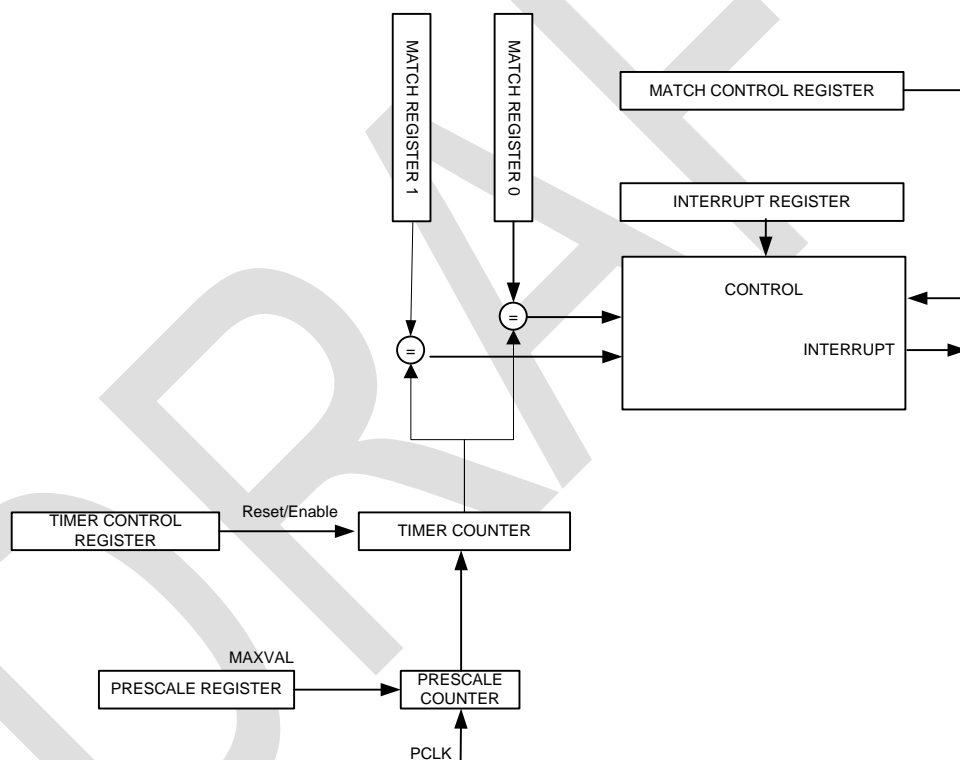


Figure 7-1: 16 位基本型定时器/计数器模块框图

7.2 寄存器描述

Table 7-2: 16 位定时器/计数器的寄存器概貌 (TIM0 基地址 0x4001 0000; TIM1 基地址 0x4001 4000)

名称	读写	偏移地址	描述	初始值
IR	R/W	0x000	中断寄存器。中断寄存器用来清除中断或读取计数器/定时器中断状态。	0
TCR	R/W	0x004	定时器控制寄存器。TCR 用于定义定时器功能，也可以通过它复位或终止定时器。	0
TC	R/W	0x008	定时器计数寄存器。计数值在每个 PCLK 时钟都会加 1。计数功能由 TCR 寄存器控制。	0
PR	R/W	0x00C	预分频寄存器，当 PC 值等于该数据时，下个时钟 TC 增加，PC 值清零	0
PC	R/W	0x010	预分频计数寄存器。16 位计数器，最高到分频值。当 PC 计数到 PR 值时，TC 计数器加 1，PC 也会被清零。	0
MCR	R/W	0x014	匹配控制寄存器。MCR 用于控制触发中断或清除 TC 寄存器。	0
MR0	R/W	0x018	匹配寄存器 0。当 TC 计数器值等于 MR0 时，清零 TC 寄存器，终止 TC 计数，或触发中断。	0
MR1	R/W	0x01C	匹配寄存器 1。参照 MR0 描述。	0
MR2	R/W	0x020	匹配寄存器 2。参照 MR0 描述。	0
MR3	R/W	0x024	匹配寄存器 3。参照 MR0 描述。	0

7.2.1 中断寄存器

中断寄存器包含 4 个用于匹配中断的位及 2 个用于捕获中断的位。如果有中断产生，IR 中的相应位为高电平。否则，该位为低电平。向对应的 IR 位写逻辑 1 会使中断复位。写 0 无效。清除定时器匹配中断会同时清除任何相应的 DMA 请求。

Table 7-3: 中断寄存器 (IR, 地址: 0x4001 0000 (TIM0) ; 0x4001 4000 (TIM1)) 描述

位	名称	描述	初始值
0	MR0INT	匹配通道 0 中断标志	0
1	MR1INT	匹配通道 1 中断标志	0
31:2	-	保留	-

7.2.2 定时器控制寄存器

定时器控制寄存器用于控制计数器/定时器的操作。

Table 7-4: 定时器控制寄存器 (TCR, 地址: 0x4001 0004 (TIM0) ; 0x4001 4004 (TIM1)) 描述

位	名称	值	描述	初始值
0	CEN		计数器使能	0
		0	计数器被禁止	
		1	定时器计数和预分频计数使能	
1	CRST		计数器复位	0

		0	无效操作.	
		1	定时器计数器和预分频计数器在下一个 PCLK 时钟上升沿复位。计数器保持复位直到 TCR 被清零。	
31:2	-	-	保留.	NA

7.2.3 定时器计数寄存器

当预分频器计数器达到其 PC 数值时，16 位定时器计数器会递增计数。如果 TC 在到达计数器上限之前没有复位，它将一直计数到 0x0000 FFFF 然后翻转到 0x0000 0000。该事件不会产生中断，如果需要，可使用匹配寄存器检测溢出。

Table 7-5: 定时器计数寄存器 (TC, 地址: 0x4001 0008 (TIM0) ; 0x4001 4008 (TIM1)) 描述

位	名称	描述	初始值
15:0	TC	定时器计数值.	0
31:16	-	保留.-	NA

7.2.4 预分频寄存器

16 位预分频寄存器指定分频计数器的最大值。

Table 7-6: 预分频寄存器 (PR, 地址: 0x4001 000C (TIM0) ; 0x4001 400C (TIM1)) 描述

位	名称	描述	初始值
15:0	PCVAL	预分频值.	0
31:16	-	保留.	-

7.2.5 预分频计数寄存器

16 位预分频计数器用某个常量来控制 PCLK 的分频，再使其输入到定时器计数器。它所控制的是定时器分辨率与最大时间之间的关系，从而能防止定时器溢流。预分频计数器会在每个 PCLK 时钟上递增计数。当预分频计数器的计数达到预分频寄存器中存储的值时，定时器计数器将递增计数，并且在下一个 PCLK 时钟上对预分频计数器复位。这将使得 TC 当 PR = 0 时在每个 PCLK 上递增计数，当 PR = 1 时，在每 2 个 PCLK 上递增计数，依次类推。.

Table 7-7: 预分频计数寄存器 (PC, 地址: 0x4001 0010 (TIM0) ; 0x4001 4010 (TIM1)) 描述

位	名称	描述	初始值
15:0	PC	预分频当前计数值.	0
31:16	-	保留.	-

7.2.6 匹配控制寄存器

匹配控制寄存器用于控制当其中一个匹配寄存器的值与定时器计数器的值匹配时应执行的操作。功能如下所示。

Table 7-8: 匹配控制寄存器 (MCR, 地址: 0x4001 0014 (TIM0) ; 0x4001 4014 (TIM1)) 描述

位	名称	值	描述	初始值
0	MR0I		当 MR0 与 TC 值匹配时中断使能。	0

		1	使能	
		0	禁止	
1	MR0R		MR0 与 TC 匹配时 TC 复位使能。	0
		1	使能	
		0	禁止	
2	MR0S		MR0 与 TC 匹配时 TC 和 PC 停止， TCR 的 CEN 置 0 使能	0
		1	使能	
		0	禁止	
3	MR1I		当 MR1 与 TC 值匹配时中断使能。	0
		1	使能	
		0	禁止	
4	MR1R		MR1 与 TC 匹配时 TC 复位使能。	0
		1	使能	
		0	禁止	
5	MR1S		MR1 与 TC 匹配时 TC 和 PC 停止， TCR 的 CEN 置 0 使能	0
		1	使能	
		0	禁止	
6	MR2I		当 MR2 与 TC 值匹配时中断使能。	0
		1	使能	
		0	禁止	
7	MR2R		MR2 与 TC 匹配时 TC 复位使能。	0
		1	使能	
		0	禁止	
8	MR2S		MR2 与 TC 匹配时 TC 和 PC 停止， TCR 的 CEN 置 0 使能	0
		1	使能	
		0	禁止	
9	MR3I		当 MR3 与 TC 值匹配时中断使能。	0
		1	使能	
		0	禁止	
10	MR3R		MR3 与 TC 匹配时 TC 复位使能。	0
		1	使能	
		0	禁止	
11	MR3S		MR3 与 TC 匹配时 TC 和 PC 停止， TCR 的 CEN 置 0 使能	0
		1	使能	
		0	禁止	
31:12			保留。	NA

7.2.7 匹配寄存器

匹配寄存器组的值会不断地与定时器计数器值进行比较。当两个值相等时，自动触发相应操作。这些操作包括产生中断、复位定时器计数器或停止定时器。所有操作均由 **MCR** 寄存器中的设置控制。

Table 7-9: 匹配寄存器 s (MR0 ~ 3, 地址: 0x4001 0018 ~ 24 (TIM0) 和 0x4001 4018~24 (TIM1)) 描述

位	名称	描述	初始值
15:0	MATCH	匹配比较值.	0
31:16	-	保留.	NA

8 16 位定时器/计数器 TIM2/TIM3

8.1 概述

G32F0xx 内置两个多功能的 16 位定时器/计数器。定时器/计数器工作时钟由 SYSAHBCLKDIV 寄存器控制。关闭 SYSAHBCLKDIV 寄存器中定时器/计数器的时钟供给可节省系统功耗。主要功能如下：

- 可预置分频的 16 位定时器/计数器
- 增强的定时器/计数器：
 - 沿计数
 - 门控计数
 - 正交计数
 - 触发计数
 - 带符号计数
- 16 位信号捕捉定时器，可触发中断和信号测量
- 4 个 16 位匹配寄存器：
 - 可产生中断
 - 停止定时器
 - 对定时器复位
- 有两个外部输出对应匹配事件发生时：
 - 当匹配发生时，输出低电平
 - 当匹配发生时，输出高电平
 - 当匹配发生时，触发事件

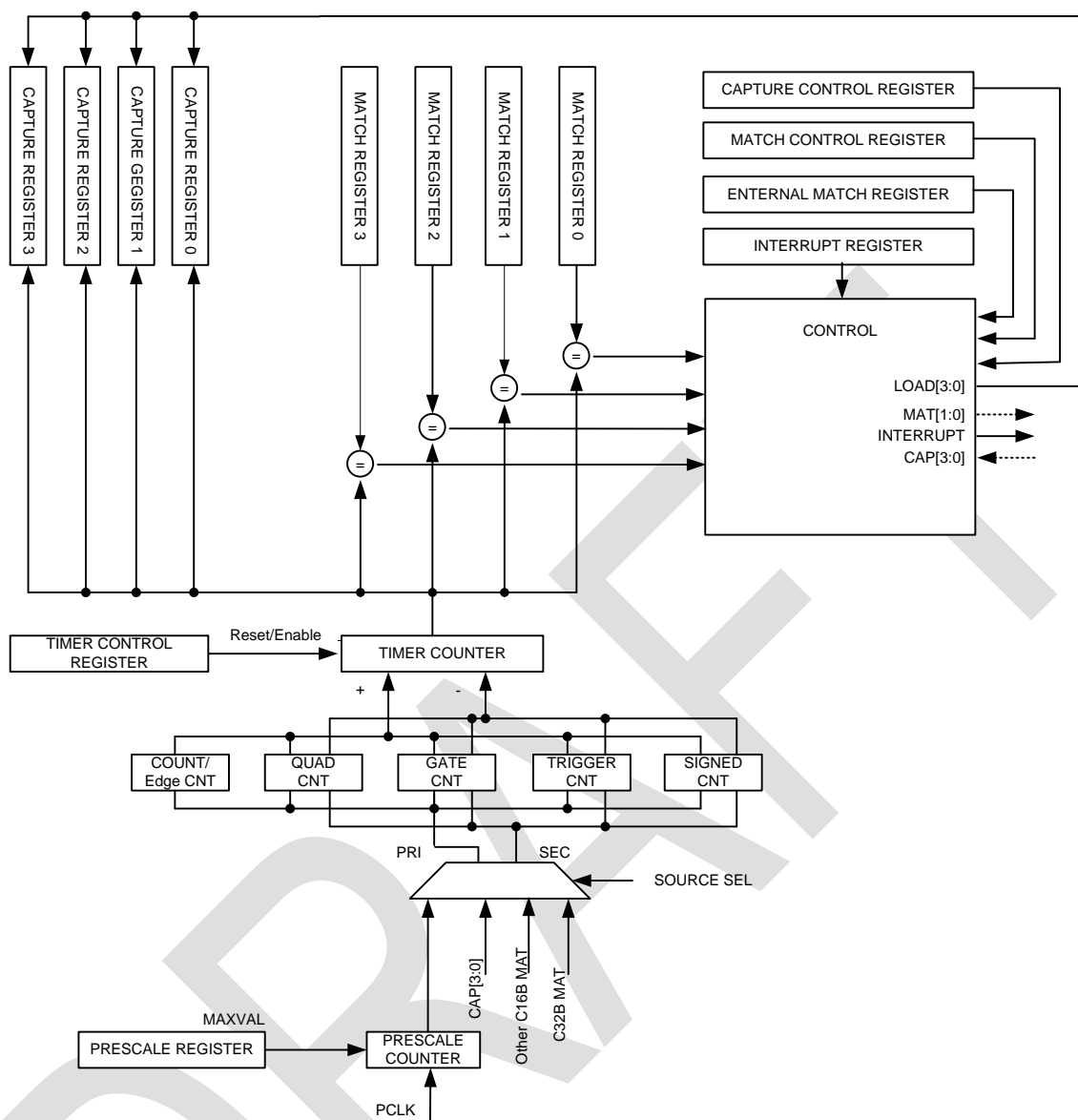


Figure 8-1: 16 位定时器/计数器模块框图

8.2 管脚描述

Table 8-1: 定时器/计数器管脚描述

管脚	类型	描述
TIM2_CAP[3:0] TIM3_CAP[3:0]	输入	信号捕捉: 信号捕捉管脚捕捉信号变化进而触发中断处理。此管脚信号可以作为计数器的时钟源。
TIM2_MAT[3:0] TIM3_MAT[3:0]	输出	TIM2/3 匹配外部输出: 当 TIM2/3 的匹配寄存器 (MR1:0) 等于当前 TC 计数值时, 可配置在管脚输出一个高电平、低电平或电平切换。寄存器 EMR 和 PWMCON 会影响到匹配输出功能。

8.3 寄存器描述

如下所示 16 位定时器/计数器控制寄存器列表。

Table 8-2: 16 位定时器/计数器的寄存器一览(TIM2 基地址 0x4001 0000; TIM3 基地址 0x4001 4000)

名称	读写	偏移地址	描述	初始值
IR	R/W	0x000	中断寄存器。中断寄存器用来清除中断或读取计数器/定时器中断状态。	0
TCR	R/W	0x004	定时器控制寄存器。TCR 用于定义定时器功能，也可以通过它复位或终止定时器。	0
TC	R/W	0x008	定时器计数寄存器。计数值在每个 PCLK 时钟都会加 1。计数功能由 TCR 寄存器控制。	0
PR	R/W	0x00C	预分频计数寄存器 16 位计数器，最高到分频值。当 PC 计数到 PR 值时，TC 计数器加 1，PC 也会被清零。	0
PC	R/W	0x010	预分频计数寄存器。16 位计数器，最高到分频值。当 PC 计数到 PR 值时，TC 计数器加 1，PC 也会被清零。	0
MCR	R/W	0x014	匹配控制寄存器。MCR 用于控制触发中断或清除 TC 寄存器。	0
MR0	R/W	0x018	匹配寄存器 0。当 TC 计数器值等于 MR0 时，清零 TC 寄存器，终止 TC 计数，或触发中断。	0
MR1	R/W	0x01C	匹配寄存器 1。参照 MR0 描述。	0
MR2	R/W	0x020	匹配寄存器 2。参照 MR0 描述。	0
MR3	R/W	0x024	匹配寄存器 3。参照 MR0 描述。	0
CCR	R/W	0x028	信号捕捉控制寄存器。在捕捉的信号变化时，CCR 控制锁存 TC 值到捕捉寄存器，触发中断。	0
CR0	RO	0x02C	捕捉寄存器 0。CR0 用于当 CT16Bn_CAP0 信号有效时去锁存 TC 值。	0
CR1	RO	0x030	捕捉寄存器 1。CR0 用于当 CT16Bn_CAP1 信号有效时去锁存 TC 值。	0
CR2	RO	0x034	捕捉寄存器 2。CR0 用于当 CT16Bn_CAP2 信号有效时去锁存 TC 值。	0
CR3	RO	0x038	捕捉寄存器 3。CR0 用于当 CT16Bn_CAP3 信号有效时去锁存 TC 值。	0
EMR	R/W	0x03C	外部匹配寄存器。EMR 用于控制匹配功能以及外部匹配管脚输出 CT16Bn_MAT[3:0]。	0
-	-	0x040 - 0x06C	保留	-
CTCR	R/W	0x070	计数器控制寄存器。CTCR 用于选择计数器类型和计数信号源。	0

8.3.1 中断寄存器

中断寄存器包含 4 个用于匹配中断的位及 2 个用于捕获中断的位。如果有中断产生，IR 中的相应位为高电平。否则，该位为低电平。向对应的 IR 位写逻辑 1 会使中断复位。写 0 无效。清除定时器匹配中断会同时清除任何相应的 DMA 请求。

Table 8-3: 中断寄存器 (IR, 地址: 0x4001 0000 (TIM2)和 0x4001 4000 (TIM3)) 描述

位	名称	描述	初始值
0	MR0INT	匹配通道 0 中断标志	0
1	MR1INT	匹配通道 1 中断标志	0
2	MR2INT	匹配通道 2 中断标志	0
3	MR3INT	匹配通道 3 中断标志	0
4	CR0INT	捕捉通道 0 事件中断标志	0
5	CR1INT	捕捉通道 1 事件中断标志	0
6	CR2INT	捕捉通道 2 事件中断标志	0
7	CR3INT	捕捉通道 3 事件中断标志	0
31:8	-	保留 -	-

8.3.2 定时器控制寄存器

定时器控制寄存器用于控制计数器/定时器的操作。

Table 8-4: 定时器控制寄存器(TCR, 地址: 0x4001 0004 (TIM2), 0x4001 4004 (TIM3)) 描述

位	名称	值	描述	初始值
0	CEN		计数器使能	0
		0	计数器被禁止	
		1	定时器计数和预分频计数使能	
1	CRST		计数器复位	0
		0	无效操作。	
		1	定时器计数器和预分频计数器在下一个 PCLK 时钟上升沿复位。计数器保持复位直到 TCR 被清零。	
31: 14	-	-	保留。	N/A

8.3.3 定时器计数寄存器

当预分频器计数器达到其最终计数时，16 位定时器计数器会递增计数。如果 TC 在到达计数器上限之前没有复位，它将一直计数到 0xFFFF 然后翻转到 0x0000。该事件不会产生中断，如果需要，可使用匹配寄存器检测溢出。

Table 8-5: 定时器计数寄存器(TC, address 0x4001 0008 (TIM2) and 0x4001 4008 (TIM3)) 描述

位	名称	描述	初始值
15:0	TC	定时器计数值	0
31:16		保留	0

8.3.4 预分频寄存器

16 位预分频寄存器指定分频计数器的最大值。

Table 8-6: 预分频寄存器 (PR, 地址: 0x4001 000C (TIM2), 0x4001 400C (TIM3)) 描述

位	名称	描述	初始值
15:0	PCVAL	预分频值.	0
31:16	-	保留	-

8.3.5 预分频计数寄存器

16 位预分频计数器用某个常量来控制 PCLK 的分频, 再使其输入到定时器计数器。它所控制的是定时器分辨率与最大时间之间的关系, 从而能防止定时器溢流。预分频计数器会在每个 PCLK 时钟上递增计数。当预分频计数器的计数达到预分频寄存器中存储的值时, 定时器计数器将递增计数, 并且在下一个 PCLK 时钟上进行预分频计数器复位。这将使得 TC 当 PR = 0 时在每个 PCLK 上递增计数, 当 PR = 1 时, 在每 2 个 PCLK 上递增计数, 依次类推。

Table 8-7: 预分频计数寄存器 (PC, 地址: 0x4001 0010 (TIM2), 0x4001 4010 (TIM3)) 描述

位	名称	描述	初始值
15:0	PC	预分频当前计数值	0
31:16	-	保留	-

8.3.6 匹配控制寄存器

匹配控制寄存器用于控制当其中一个匹配寄存器的值与定时器计数器的值匹配时应执行的操作。功能如下所示。

Table 8-8: 匹配控制寄存器寄存器 (MCR, 地址: 0x4001 0014 (TIM2), 0x4001 4014 (TIM3)) 描述

位	名称	值	描述	初始值
0	MR0I		当 MR0 与 TC 值匹配时中断使能。	0
		1	使能	
		0	禁止	
1	MR0R		MR0 与 TC 匹配 TC 复位使能。	0
		1	使能	
		0	禁止	
2	MR0S		MR0 与 TC 匹配时 TC 和 PC 停止, TCR 的 CEN 置 0 使能	0
		1	使能	
		0	禁止	
3	MR1I		当 MR1 与 TC 值匹配时中断使能。	0
		1	使能	
		0	禁止	
4	MR1R		MR1 与 TC 匹配 TC 复位使能。	0
		1	使能	

		0	禁止	
5	MR1S		MR1 与 TC 匹配时 TC 和 PC 停止， TCR 的 CEN 置 0 使能	0
		1	使能	
		0	禁止	
6	MR2I		当 MR2 与 TC 值匹配时中断使能。	0
		1	使能	
		0	禁止	
7	MR2R		MR2 与 TC 匹配 TC 复位使能。	0
		1	使能	
		0	禁止	
8	MR2S		MR2 与 TC 匹配时 TC 和 PC 停止， TCR 的 CEN 置 0 使能	0
		1	使能	
		0	禁止	
9	MR3I		当 MR3 与 TC 值匹配时中断使能。	0
		1	使能	
		0	禁止	
10	MR3R		MR3 与 TC 匹配 TC 复位使能。	0
		1	使能	
		0	禁止	
11	MR3S		MR3 与 TC 匹配时 TC 和 PC 停止， TCR 的 CEN 置 0 使能	0
		1	使能	
		0	禁止	
31:12			保留	NA

8.3.7 匹配寄存器组

匹配寄存器组的值会不断地与定时器计数器值进行比较。当两个值相等时，自动触发相应操作。这些操作包括产生中断、复位定时器计数器或停止定时器。所有操作均由 MCR 寄存器中的设置控制。

Table 8-9: 匹配寄存器组 (MR0 ~ 3, 起始地址: 0x4001 0018~ 24 (TIM2) , 0x4001 4018 ~ 24 (TIM3)) 描述

位	名称	描述	初始值
15:0	MATCH	匹配比较值	0
31:16	-	保留.-	NA

8.3.8 捕捉控制寄存器

捕捉控制寄存器用于控制当捕捉事件发生时，是否将计数器/ 定时器中的值装入捕捉寄存器，以及捕捉事件是否产生中断。同时将上升沿位和下降沿位置位是有效配置，会使两个边沿都产生捕捉事件。在下面描述中，“n”表示定时器编号，2 或 3。

Table 8-10: 捕捉控制寄存器 (CCR, 地址: 0x4001 0028 (TIM2) , 0x4001 4028 (TIM3)) 描述

位	名称	值	描述	初始值
---	----	---	----	-----

0	CAP0RE		CT16Bn_CAP0 上升沿捕捉设定: CT16Bn_CAP0 的从 0 至 1 的序列, 将使 CR0 锁存 TC 内容。	0
		1	使能	
		0	禁止	
1	CAP0FE		CT16Bn_CAP0 下降沿捕捉设定: CT16Bn_CAP0 的从 1 至 0 的序列, 将使 CR0 锁存 TC 内容。	0
		1	使能	
		0	禁止	
2	CAP0I		CT16Bn_CAP0 捕捉事件中断: CT16Bn_CAP0 捕捉事件所导致的 CR0 装载将产生一个中断。	0
		1	使能	
		0	禁止	
3	CAP1RE		CT16Bn_CAP1 上升沿捕捉设定: CT16Bn_CAP1 的从 0 至 1 的序列, 将使 CR0 锁存 TC 内容。	0
		1	使能	
		0	禁止	
4	CAP1FE		CT16Bn_CAP1 下降沿捕捉设定: CT16Bn_CAP1 的从 1 至 0 的序列, 将使 CR0 锁存 TC 内容。	0
		1	使能	
		0	禁止	
5	CAP1I		CT16Bn_CAP1 捕捉事件中断: CT16Bn_CAP1 捕捉事件所导致的 CR0 装载将产生一个中断。	0
		1	使能	
		0	禁止	
6	CAP2RE		CT16Bn_CAP2 上升沿捕捉设定: CT16Bn_CAP2 的从 0 至 1 的序列, 将使 CR0 锁存 TC 内容。	0
		1	使能	
		0	禁止	
7	CAP2FE		CT16Bn_CAP2 下降沿捕捉设定: CT16Bn_CAP2 的从 1 至 0 的序列, 将使 CR0 锁存 TC 内容。	0
		1	使能	
		0	禁止	
8	CAP2I		CT16Bn_CAP2 捕捉事件中断: CT16Bn_CAP2 捕捉事件所导致的 CR0 装载将产生一个中断。	0
		1	使能	
		0	禁止	
9	CAP3RE		CT16Bn_CAP3 上升沿捕捉设定: CT16Bn_CAP3 的从 0 至 1 的序列, 将使 CR0 锁存 TC 内容。	0
		1	使能	
		0	禁止	

10	CAP3FE		CT16Bn_CAP3 下降沿捕捉设定: CT16Bn_CAP3 的从 1 至 0 的序列, 将使 CR0 锁存 TC 内容。	0
		1	使能	
		0	禁止	
11	CAP3I		CT16Bn_CAP3 捕捉事件中断: CT16Bn_CAP3 捕捉事件所导致的 CR0 装载将产生一个中断。	0
		1	使能	
		0	禁止	
31:12		-	保留	NA

8.3.9 捕捉寄存器组

每个捕捉寄存器都与器件管脚相关联, 当管脚发生特定的事件时, 可将计数器/ 定时器的值装入该捕捉寄存器。捕捉控制寄存器中的设置决定是否使能捕捉功能, 及在相关管脚的上升沿、下降沿或上升沿和下降沿上是否产生捕捉事件。在下面描述中, “n” 表示定时器编号, 2 或 3。

Table 8-11: 捕捉寄存器组(CR0 ~ 3, 起始地址: 0x4001 002C~ 38 (TIM2), 0x4001 402C~ 38 (TIM3)) 描述

位	名称	描述	初始值
15:0	CAP	捕捉到计数器值	0
31:16	-	保留	NA

8.3.10 外部匹配寄存器

外部匹配寄存器为外部匹配管脚 TIM2_MAT[3:0]/ TIM3_MAT[3:0] 提供控制和状态。

Table 8-12: 外部匹配寄存器 (EMR, 地址: 0x4001 003C (TIM2), 0x4001 403C (TIM3)) 描述

位	名称	值	描述	初始值
0	EM0		外部匹配 0。该位反映输出 TIM2_MAT0/TIM3_MAT0 的状态, 不管该输出是否连接到此管脚。当 TC 与 MR0 相匹配时, 该位可以进行切换、转入低电平、转入高电平或不执行任何操作。位 EMR[5:4] 控制该输出的功能。如果选用了 IOCON 寄存器的匹配功能 (0=低电平, 1= 高电平), 该位就会被驱动到 TIM2_MAT0/TIM3_MAT0 管脚上。	0
1	EM1		外部匹配 1。该位反映输出 TIM2_MAT1/TIM3_MAT1 的状态, 不管该输出是否连接到此管脚。当 TC 与 MR1 相匹配时, 该位可以进行切换、转入低电平、转入高电平或不执行任何操作。位 EMR[7:6] 控制该输出的功能。如果选用了 IOCON 寄存器的匹配功能 (0=低电平, 1= 高电平), 该位就会被驱动到 TIM2_MAT1/TIM3_MAT1 管脚上。	0
2	EM2		外部匹配 2。该位反映输出 TIM2_MAT2/TIM3_MAT2 的状态, 不管该输出是否连接到此管脚。当 TC 与 MR2 相匹配时, 该位可以进行切换、转入低电平、转入高电平或不执行任何操作。位 EMR[9:8] 控制该输出的功能。如果选用了 IOCON 寄存器的匹配功能 (0=低电平, 1= 高电平), 该位就会被驱动到 TIM2_MAT2/TIM3_MAT2 管脚上。	0

3	EM3		外部匹配 3。该位反映输出 TIM2_MAT3/TIM3_MAT3 的状态，不管该输出是否连接到此管脚。当 TC 与 MR3 相匹配时，该位可以进行切换、转入低电平、转入高电平或不执行任何操作。位 EMR[11:10] 控制该输出的功能。如果选用了 IOCON 寄存器的匹配功能（0=低电平，1= 高电平），该位就会被驱动到 TIM2_MAT3/TIM3_MAT3 管脚上。	0
5:4	EMC0		外部匹配控制 0。决定外部匹配 0 的功能。	00
		00	无效	
		01	将相应的外部匹配位/ 输出清零（如果管脚处于输出状态，则 CT16Bn_MAT0 管脚为低电平）。	
		10	将相应的外部匹配位/ 输出置 1（如果管脚处于输出状态，则 CT16Bn_MAT0 管脚为高电平）。	
		11	切换相应的外部匹配位/ 输出。	
7:6	EMC1		外部匹配控制 1。决定外部匹配 1 的功能。	00
		00	无效	
		01	将相应的外部匹配位/ 输出清零（如果管脚处于输出状态，则 CT16Bn_MAT1 管脚为低电平）。	
		10	将相应的外部匹配位/ 输出置 1（如果管脚处于输出状态，则 CT16Bn_MAT1 管脚为高电平）。	
		11	切换相应的外部匹配位/ 输出。	
9:8	EMC2		外部匹配控制 2。决定外部匹配 2 的功能。	00
		00	无效	
		01	将相应的外部匹配位/ 输出清零（如果管脚处于输出状态，则 CT16Bn_MAT2 管脚为低电平）。	
		10	将相应的外部匹配位/ 输出置 1（如果管脚处于输出状态，则 CT16Bn_MAT2 管脚为高电平）。	
		11	切换相应的外部匹配位/ 输出。	
11: 10	EMC3		外部匹配控制 3。决定外部匹配 3 的功能。	00
		00	无效	
		01	将相应的外部匹配位/ 输出清零（如果管脚处于输出状态，则 CT16Bn_MAT3 管脚为低电平）。	
		10	将相应的外部匹配位/ 输出置 1（如果管脚处于输出状态，则 CT16Bn_MAT3 管脚为高电平）。	
		11	切换相应的外部匹配位/ 输出。	
31: 12	-	-	保留	NA

8.3.11 计数控制寄存器

计数控制寄存器(CTCR) 用于在定时器模式和计数器模式之间进行选择，且在处于计数器模式时选择进行计数的管脚和边沿。

如果将计数器模式选为操作模式，则在 PCLK 时钟的每个上升沿上会对（CTCR 位 3:2 所选的）CAP 输入进行采样。在比较该 CAP 输入的两个连续样本后，将会确认下列四个事件之一：所选 CAP 输入电平处于上升沿、下降沿，或上升下降沿或无变化。仅当所标识的事件与 CTCR 寄存器中的位 1:0 所选的内容相匹配时，定时器计数器寄存器才会递增计数。

要有效地处理计数器的外部源时钟会有一些限制。由于 PCLK 时钟的两个连续上升沿仅用于标识 CAP 所选输入的一个边缘，因此 CAP 输入的频率不能超过 PCLK 时钟的一半。因此在这种情况下，相同的 CAP 输入上的高电平/低电平的持续时间不能少于 1/PCLK。

该寄存器的位 7:3 还用于使能和配置捕获-清除-定时器特性。该特性允许特定 CAP 输入的指定边沿将定时器全部清零。使用该机制在输入脉冲前沿清除定时器然后在后沿执行捕获，可以使用单捕获输入来直接测量脉冲宽度，而无需在软件中执行减法操作。

Table 8-13: 计数寄存器 (CTCR, 地址: 0x4001 0070 (TIM2), 0x4001 4070 (TIM3)) 描述

位	名称	值	描述	初始值
2:0	CTM		计数器/定时器模式。该字段用于选择可使用哪个 PCLK 上升沿，使定时器预分频计数器(PC) 递增计数，或清除 PC 并使定时器计数器(TC) 递增计数。如果在 CTCR 中选择定时器模式，必须将捕获控制寄存器(CCR) 中的位 2:0 设置为 000。	00
		000	定时器模式：每个 PCLK 上升沿	
		001	计数器模式：TC 在位 11:8 选择的 CAP 输入的上升沿时递增。	
		010	计数器模式：TC 在位 11:8 选择的 CAP 输入的下降沿时递增。	
		011	计数器模式：TC 在位 11:8 选择的 CAP 输入的上升和下降沿时递增。	
		100	正交编码计数模式	
		101	触发计数模式	
		110	符号计数模式	
		111	门控计数模式	
3	ENCC		将此位置 1 可在发生位 7:4 指定的捕获-边沿事件时清零定时器和预分频器。	0
7:4	SELCC		当位 3 为 1 时，这两位选择哪个捕获输入边沿将导致定时器和预分频器被清零。当位 3 为低电平时，这两位无效。	000
		0000	CAP0 的上升沿清零定时器（如果位 3 被置位）	
		0001	CAP0 的下降沿清零定时器（如果位 3 被置位）	
		0010	CAP1 的上升沿清零定时器（如果位 3 被置位）	
		0011	CAP1 的下降沿清零定时器（如果位 3 被置位）	
		0100	CAP2 的上升沿清零定时器（如果位 3 被置位）	
		0101	CAP2 的下降沿清零定时器（如果位 3 被置位）	
		0110	CAP3 的上升沿清零定时器（如果位 3 被置位）	
		0111	CAP3 的下降沿清零定时器（如果位 3 被置位）	
		1000	另一个计数器 CAP0 的上升沿清零定时器（如果位 3 被置位）	
		1001	另一个计数器 CAP0 的下降沿清零定时器（如果位 3 被置位）	
		1010	另一个计数器 CAP1 的上升沿清零定时器（如果位 3 被置位）	
		1011	另一个计数器 CAP1 的下降沿清零定时器（如果位 3 被置位）	
		1100	另一个计数器 CAP2 的上升沿清零定时器（如果位 3 被置位）	

		1101	另一个计数器 CAP2 的下降沿清零定时器（如果位 3 被置位）	
		1110	另一个计数器 CAP3 的上升沿清零定时器（如果位 3 被置位）	
		1111	另一个计数器 CAP3 的下降沿清零定时器（如果位 3 被置位）	
11:8	PRISEL		计数主时钟源选择	0000
		0000	捕捉管脚 CAP0.	
		0001	捕捉管脚 CAP1.	
		0010	捕捉管脚 CAP2	
		0011	捕捉管脚 CAP3.	
		0100	PWM 重载	
		0101	TIM2_MAT0./ TIM3_MAT0 输出触发信号	
		0110	保留	
		0111	保留	
		1xxx	预分频计数器	
15:12	SECSEL		计数从时钟源选择	0000
		0000	捕捉管脚 CAP0.	
		0001	捕捉管脚 CAP1.	
		0010	捕捉管脚 CAP2	
		0011	捕捉管脚 CAP3.	
		0100	PWM 重载	
		0101	TIM2_MAT0./ TIM3_MAT0 输出触发信号	
		0110	保留	
		0111	保留	
		1xxx	预分频计数器	
16	IPS		从时钟源输入极性选择	0
		0	无反转	
		1	输入信号极性反转	
31: 17	-	-	保留.	NA

8.3.11.1 边沿计数模式

当计数器选择边沿计数（上升沿，下降沿）时，计数器会对所选择的时钟信号沿进行计数。此计数模式广泛应用于外部输入变化计数中。

8.3.11.2 正交编码计数模式

当正交编码计数模式被选择后，计数器会对选定的主，次输入信号进行正交译码。正交编码信号通常用作马达控制中旋转编码器和光栅直尺测量的输出信号。正交信号是由两路相位相差 90° 方波组成。正交译码可以提供计数数量和计数方向两方面的信息。下面的时序图是对正交编码的一个简单说明。

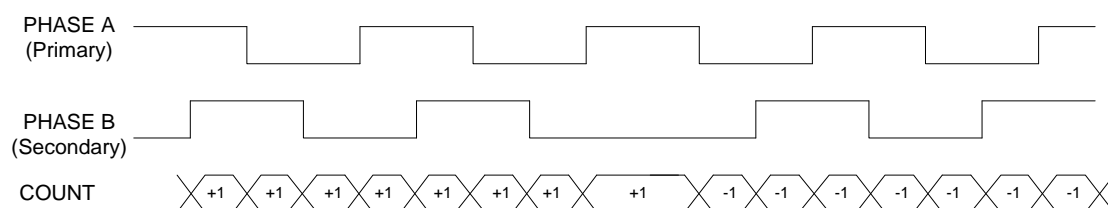


Figure 8-2: 正交编码计数

8.3.11.3 触发计数模式

当计数器被设置为触发计数模式后，如从时钟信号检测到上升沿，计数器开始对主信号源时钟进行计数，直到匹配事件发生或另一个从时钟信号上升沿被检测到。在计数器到达计数终点，从时钟信号可以不断地启动、停止、启动控制计数器计数。

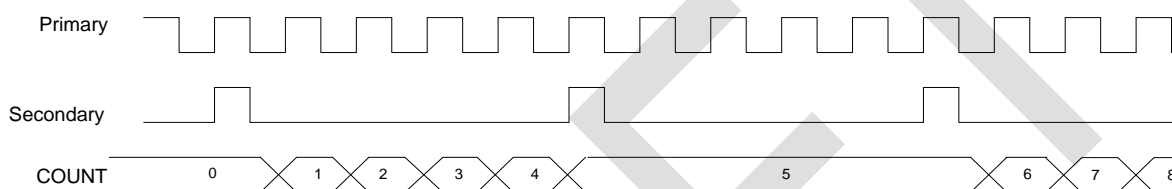


Figure 8-3: 触发计数模式

8.3.11.4 符号计数模式

当计数器被设置为符号计数模式后，主信号源时钟导致计数器作加计数，从时钟信号作减计数。

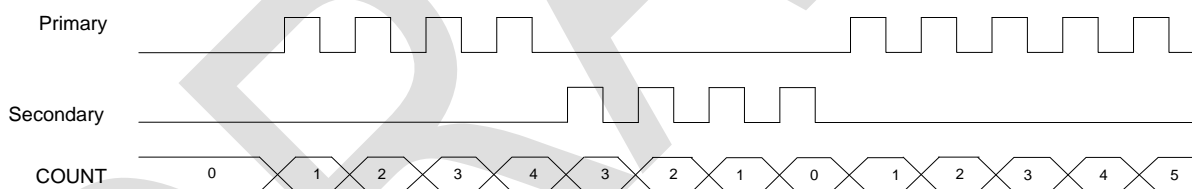


Figure 8-4: 符号计数模式

8.3.11.5 门控计数模式

在此模式下，计数器仅在从时钟信号处于高电平时才对主时钟信号进行计数。此计数方法可用于测量从时钟信号长度。如输入极性选择（IPS）反转，从时钟信号低电平信号会使能计数器计数。

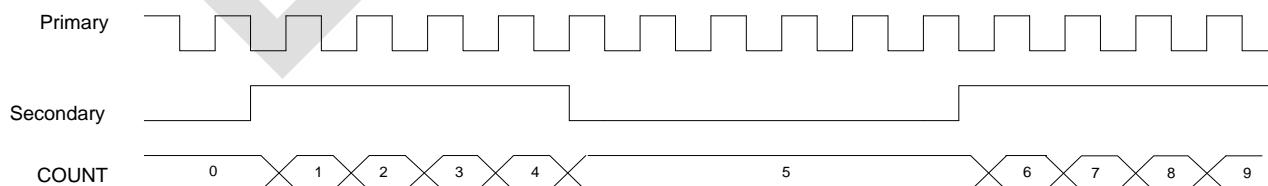


Figure 8-5: 门控计数模

9 看门狗定时器 (WDT)

9.1 概述

看门狗定时器用于在用户程序出错并无法喂狗后对系统进行中断和复位处理。使用可编程的看门狗定时器，用户可改变定时器时间去应对不同的应用程序。该看门狗定时器有如下主要功能：

- 独立的频率可以设定的看门狗时钟振荡器
- 看门狗定时器可触发中断或复位
- 支持低功耗模式

下图是看门狗功能模块图。

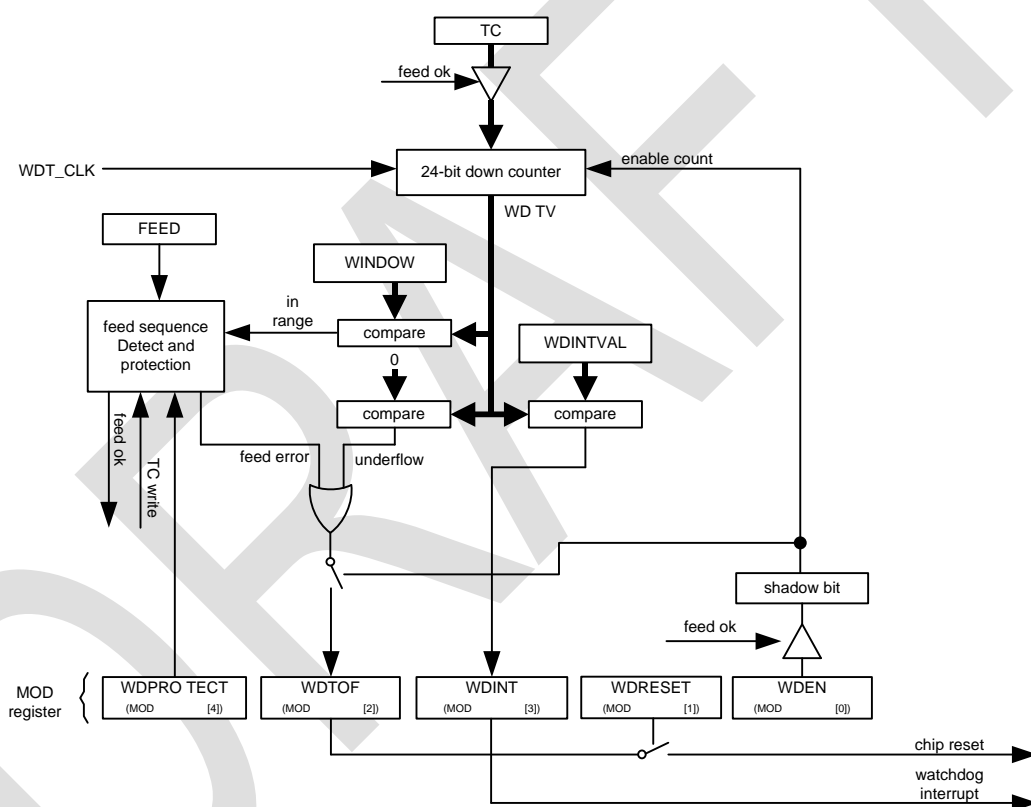


Figure 9-1: 看门狗功能模块图

9.2 寄存器描述

Table 9-1: 看门狗寄存器一览(基地址 0x4000 4000)

名称	读写	偏移地址	描述	初始值
----	----	------	----	-----

MOD	R/W	0x000	看门狗模式寄存器。此寄存器包涵基本的看门狗模式和状态信息。	0x0000 0003
TC	R/W	0x004	看门狗定时器常数寄存器。此寄存器决定看门狗超时值。	0x0000 FFFF
FEED	WO	0x008	看门狗喂狗命令寄存器。向该寄存器写入0xAA 和0x55 使看门狗定时器重新载入TC 中的值。	NA
TV	RO	0x00C	看门狗定时器寄存器。该寄存器读出看门狗定时器的当前值。	0xFF
-	-	0x010	-	-
WARNINT	R/W	0x014	看门狗警告中断比较值寄存器	0
WINDOW	R/W	0x018	看门狗窗口寄存器	0xFF FFFF

9.2.1 看门狗模式寄存器

MOD 寄存器通过 WDEN 和 RESET 位的组合控制看门狗的操作。看门狗超时可以产生看门狗复位或中断。如果在睡眠模式中发生看门狗中断，则看门狗中断会唤醒系统。

Table 9-2: 看门狗模式寄存器 (MOD - 0x4000 4000) 描述

位	名称	值	描述	初始值
0	WDEN		看门狗启用位。通过 WDLOCKEN 位后续写入可锁定 WDEN 位。	1
		0	看门狗定时器停止。	
		1	看门狗定时器运行。	
1	WDRESET		看门狗复位使能位。通过 WDLOCKEN 位后续写入可锁定 WDRESET 位。	0
		0	看门狗超时会引起系统复位。	
		1	看门狗超时会引起中断。	
2	WDTOF		看门狗超时标志。在看门狗超时、发生喂狗错误或当 WDPROTECT=1 且尝试向 WDTC 寄存器写入时，便会设置看门狗超时标志。通过软件向此位写入 0 可将该标志清零。WDTOF 位置 1，都可以触发中断。在任何情况下，非看门狗超时引起的看门狗超时标志置位都不会触发系统复位。。	0 (仅在外部复位后)
3	WDINT		当看门狗计数器计数到 WDWARNINT 指定的值时，便会设置看门狗中断标志。此标志可通过任何复位或通过软件向此位写入 1 清零。	0
4	WDPROTECT		看门狗保护模式。此位只能置位。WDPROTECT 位一旦置位，便无法通过软件清零。WDPROTECT 位可通过外部复位或看门狗定时器复位清零。	0
		0	看门狗定时器常量值(WDTC) 可随时更改。	
		1	只有计数器小于 WARNINT 和 WINDOW 的值后，才能更改看门狗定时器常量值(WDTC)。	
5	WDLOCKCLK		看门狗时钟锁定位。此位可在复位时清零，且随后只能写入一次以进行设置。此位一旦设置，便只能通过芯片复位清零。	0
		0	看门狗时钟(WDCLK) 没有锁定。	

		1	<p>设置此位将禁止向控制电源配置寄存器 PDRUNCFG、PDSLEEPCFG 和 PDAWAKECFG 中当前所选看门狗时钟源电源的位执行任何写入操作。电源配置寄存器中的其他位将不受影响。设置 WDLOCKCLK 位确保只要看门狗振荡器和/或 IRC 上电，WDT 便始终具有有效时钟源。</p> <p>注：在设置 WDLOCKCLK 位前，用户必须在所有三个电源配置寄存器中启用看门狗振荡器或 IRC 或两者，以确保所选的时钟源在活动、睡眠或深度睡眠模式下运行。一旦 WDLOCKCLK 位设置，看门狗时钟源便无法关闭或开启。如果 WDT 将在深度睡眠模式下运行，则必须先在 PDSLEEPCFG 寄存器中启用看门狗振荡器，再设置 WDLOCKCLK 位。</p>	
6	WDLOCKDP		掉电禁用位。此位可在复位时清零，且随后只能写入一次以进行设置。此位一旦设置，便只能通过芯片复位清零。	0
		0	允许进入掉电模式。	
		1	掉电模式被禁止。PMU 中的 DPDEN 位不能置 1。	
7	WDLOCKEN		看门狗启用和复位锁定位。此位可在复位时清零，且随后只能写入一次以进行设置。此位一旦设置，便只能通过芯片复位清零。	0
		0	允许写入 WDEN 和 WDRESET 位以启用或禁用看门狗操作。	
		1	<p>如果此位设为 1，则将阻止 WDEN 和 WDRESET 位的任何后续写入。当设置 WDLOCK 位时，将根据 WDEN 位的状态永久禁用或启用看门狗。复位行为受影响如下：</p> <ul style="list-style-type: none"> 如果设置 WDLOCKEN 位之前，看门狗启用且 WDRESET 位设为 1，则看门狗触发器会始终引起复位，且此行为无法由软件覆盖。 如果设置 WDLOCKEN 位之前，看门狗启用且 WDRESET 位设为 0，则看门狗触发器会始终引起中断，且此行为无法由软件覆盖。 	
31: 8			保留	

Table 9-3: 看门狗运行模式选择

WDEN	WDRESET	运行模式
0	X (0 or 1)	无看门狗模式
1	0	看门狗中断模式：将产生看门狗警告中断，但不会产生看门狗复位。 当选择这种模式时，看门狗计数器计数到 WDWARNINT 指定的值时便会设置 WDINT 标志，并产生看门狗中断请求。
1	1	看门狗复位模式：启用看门狗中断和看门狗复位。 当选择这种模式时，看门狗计数器计数到 WDWARNINT 指定的值时便会设置 WDINT 标志，并产生看门狗中断请求。看门狗计数器为零将复位微控制器。

9.2.2 看门狗定时器常数寄存器

寄存器 TC 决定超时值。每当喂狗序列产生时，TC 的内容就会重新载入看门狗定时器。复位时，值 0x00 FFFF 会预载。写入小于 0xFF 的值会使 0xFF 载入 TC。因此，最小超时时间为 $TWDCLK \times 256 \times 4$ 。如果 MOD 中的 WDPROTECT 位为 1，则在看门狗计数器小于 WARNINT 和 WINDOW 的值之前，尝试更改 TC 值将引起看门狗设置 WDTOF 标志。

Table 9-4: 看门狗定时器常数寄存器 (TC - 0x4000 4004) 描述

位	名称	描述	初始值
23:0	WDTC	看门狗超时时间间隔	0x00 FFFF
31:24	-	保留	

9.2.3 看门狗喂狗命令寄存器

向该寄存器依序写入 0xAA 和 0x55 将使 WDTC 的值重新载入看门狗定时器。非法的喂狗序列，可产生 WDTOF 标志置位。

Table 9-5: 看门狗喂狗命令寄存器 (FEED - 0x4000 4008) 描述

位	名称	描述	初始值
7:0	WDFEED	喂狗值应依序为 0xAA 和 0x55。	-
31:8		保留	

9.2.4 看门狗定时器寄存器

WDTV 寄存器用于读取看门狗定时器的当前值。

Table 9-6: 看门狗定时器值寄存器 (TV - 0x4000 400C) 描述

位	名称	描述	初始值
23:0	WDTV	看门狗定时器值	0x00 00FF
31:24	-	保留	NA

9.2.5 看门狗警告中断比较值寄存器

WDWARNINT 寄存器决定将产生看门狗中断的看门狗定时器值。当看门狗定时器与 WDWARNINT 定义的值匹配时，将在后续 WDCLK 后产生中断。当计数器的低 10 位与 WARNINT 的 10 位具有相同值，且计数器其余高位均为 0 时，便触发看门狗警告中断。在看门狗超时之前会有最长 1,023 个看门狗定时器计数（4,096 个看门狗时钟）时间。如果 WARNINT 设为 0，则将与看门狗事件同时产生中断。

Table 9-7: 看门狗警告中断寄存器(WARNINT - 0x4000 4014) 描述

位	名称	描述	初始值
9:0	WARNINT	看门狗警告中断比较值	0
31:10	-	保留	NA

9.2.6 看门狗窗口比较值寄存器

WDWINDOW 寄存器决定在执行看门狗喂狗时允许的 WDTV 最大值。如果喂狗有效序列在 WDTV 达到 WDWINDOW 中的值之前完成，则将发生看门狗喂狗错误事件。WDWINDOW 会复位为 WDTV 最大的可能值，因此窗口不会生效。如果 WDWINDOW 的值小于 0x100，将无法成功进行看门狗喂狗。

Table 9-8: 看门狗窗口比较值寄存器 (WINDOW - 0x4000 4018) 描述

位	名称	描述	初始值
---	----	----	-----

23:0	WDWINDOW	看门狗窗口比较值.	0xFF FFFF
31:24	-	保留	NA

9.2.7 看门狗时钟及低功耗控制

看门狗定时器模块使用两个时钟：PCLK 和 WDCLK。PCLK 由系统时钟生成，供 APB 访问看门狗寄存器使用。WDCLK 由看门狗时钟振荡器中的 wdt_clk 生成，供看门狗定时器计数使用。以下两个时钟可用作 wdt_clk 时钟的时钟源：IRC 和看门狗振荡器。时钟源在 WDCLKSEL 寄存器中选择，但请注意，时钟源可以由软件通过 MODE 寄存器锁定。

这两个时钟域之间存在一些同步逻辑。当 MOD 和 TC 寄存器通过 APB 操作更新时，新的值将由 WDCLK 时钟设定并生效。当看门狗定时器处于 WDCLK 时钟周期时，同步逻辑会先锁定 WDCLK 上计数器的值，然后使其与 PCLK 同步，以作为 TV 寄存器的值供 CPU 读取。

如果没有使用看门狗振荡器，则可在 PDRUNCFG 寄存器中将其关闭，除非设置了 MOD 寄存器中的位 5。为了节能，可在 AHBCLKCTRL 寄存器将输入到看门狗寄存器模块的时钟(PCLK) 禁用。

9.2.8 看门狗锁定功能

可以通过多种方式锁定看门狗定时器操作，以确保 WDT 始终处于运行状态。锁定功能可以通过一次写入相应的锁定寄存器位来启用，只有通过芯片复位才能取消。可应用以下锁定机制：

- 锁定 WDT 的启用/ 禁用状态，并同时锁定看门狗是否触发中断或复位
- 锁定时钟源的切换。此锁定机制可防止切换至已掉电的时钟源
- 锁定 PDRUNCFG、PDSLEEPCFG、PDAWAKECFG 寄存器中任何 WDT 时钟源的电源控制
- 锁定更新 WDT 重载值
- 锁定进入掉电模式

注：使用锁定功能需注意下列事项

- 确保在锁定电源控制和时钟源选择前，所有三个电源配置寄存器 PDSLEEPCFG、PDRUNCFG 和 PDAWAKECFG 中所选的 WDT 时钟源已上电。
- 如果在深度睡眠模式下使用 WDT，则必须在锁定电源控制前开启看门狗振荡器。

10 CRC

10.1 概述

G32F0 集成一个微型 CRC 协处理器,它用来处理控制器的复杂计算。软件可以通过调用 CRC 寄存器去完成预设的计算功能。CRC 系统时钟 CRC_PCLK 由控制器系统时钟提供并由 SYSAHBCLKCTRL 控制。类似控制器其它模块, CRC 可以通过关闭时钟从而节能。xDSP 可以完成下列任务:

- CRC-CCITT
- CRC-16
- CRC-32

10.2 CRC 接口寄存器描述

Table 10-1: 寄存器一览 (基地址 0x5007 0000)

名称	读写	偏移地址	描述	初始值
CRC_MODE	R/W	0x00	CRC 模式寄存器	0x0000 0000
CRC_SEED	R/W	0x04	CRC 种子寄存器	0x0000 FFFF
CRC_SUM	R/W	0x08	读: CRC 校验和寄存器 写: CRC 数据寄存器	0x0000 FFFF

10.2.1 CRC 模式寄存器

Table 10-2: CRC 模式寄存器 (CRC_MODE, 地址 0x5007 0000) 描述

位	名称	值	描述	初始值
1:0	CRC_POLY		CRC 多项式选择	00
		00	选择 CRC-CCITT 多项式	
		01	选择 CRC-16 多项式	
		1X	选择 CRC-32 多项式	
2	BIT_RVS_WR		数据位顺序	0
		0	不对 CRC_WR_DATA 进行位顺序取反 (每字节)	
		1	对 CRC_WR_DATA 进行位顺序取反 (每字节)	
3	CMPL_WR		数据补码	0
		0	不针对 CRC_WR_DATA 采用 1 的补码	
		1	1 的补码 (针对 CRC_WR_DATA)	
4	BIT_RVS_SUM		CRC 校验和位顺序	0
		0	不对 CRC_SUM 进行位顺序取反	
		1	对 CRC_SUM 进行位顺序取反	
5	CMPL_SUM		CRC 校验和补码	0

		0	不针对 CRC_SUM 采用 1 的补码	
		1	1 的补码 (针对 CRC_SUM)	
6	SEED_OP		CRC 种子设定选择	0
		0	使用默认值	
		1	使用种子寄存器作为 CRC 运算种子	
7	SEED_SET	-	写 1 装载种子数据到 CRC	0
31:8	-	-	保留	NA

10.2.2 CRC Seed 寄存器

Table 10-3: CRC 种子寄存器 (CRC_SEED, 地址 0x5007 0004) 描述

位	名称	描述	初始值
31:0	CRC_SEED	CRC 种子值	0x0000 FFFF

10.2.3 CRC 校验和寄存器

只读寄存器。保存最近一次校验和。

Table 10-4: CRC 校验和寄存器 (CRC_SUM, 地址 0x5007 0008) 描述

位	名称	描述	初始值
31:0	CRC_SUM	可通过该寄存器按选定位顺序和1的补码后处理读取最近一次的CRC 校验和。	0x0000 FFFF

10.2.4 CRC 数据寄存器

该寄存器是只写寄存器，用于输入将计算 CRC 和的数据块。

Table 10-5: CRC Data 寄存器 (CRC_DATA, 地址 0x5007 0008) 描述

位	名称	描述	初始值
31:0	CRC_WR_DATA	将使用写入该寄存器的数据按选定位顺序和1 的补码预处理进行CRC 计算。允许8、16 或32 位的任意写大小，并接受连续计算。	-

10.3 功能描述

10.3.1 CRC 计算

下面给出了适用于每个支持的CRC 标准的寄存器设置例子：

CRC-CCITT 设置

$$\text{多项式} = x^{16} + x^{12} + x^5 + 1$$

$$\text{种子值} = 0xFFFF$$

对数据输入进行位顺序取反：否

对数据输入采用1 的补码：否

对CRC 和进行位顺序取反：否

对CRC 和采用1 的补码：否

CRC_MODE = 0x0000 0000

CRC_SEED = 0x0000 FFFF

CRC-16 设置

多项式= $x^{16} + x^{15} + x^2 + 1$

种子值 = 0x0000

对数据输入进行位顺序取反：是

对数据输入采用1 的补码：否

对CRC 和进行位顺序取反：是

对CRC 和采用1 的补码：否

CRC_MODE = 0x0000 0015

CRC_SEED = 0x0000 0000

CRC-32 设置

多项式= $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

种子值 = 0xFFFF FFFF

对数据输入进行位顺序取反：是

对数据输入采用1 的补码：否

对CRC 和进行位顺序取反：是

对CRC 和采用1 的补码：是

CRC_MODE = 0x0000 0036

CRC_SEED = 0xFFFF FFFF

11 SPI

11.1 概述

G32F0XX 支持扩展的 SPI (Serial Peripheral interface) 接口, 它可以支持标准的 SPI 操作, 兼容 4 线的 SSI (Synchronous Serial Interface)。该接口允许多主、从机总线模式, 但同一时刻, 只有一个主/从机在发送数据。数据传输支持 4 位 到 16 位全双工模式。外设时钟 (SPI_PCLK) 由系统时钟 (system clock) 提供, 并受控于 SYSAHBCLKDIV 寄存器。主要功能如下:

- 支持标准的 SPI, 兼容 4 线的 SSI
- 同步串行通信
- 支持主/从机模式
- 深度为 8 帧的先进先出 FIFO (First In First Out) 缓存器
- 每帧数据长度 4-位到 16-位

11.2 管脚描述

Table 11-1: SPI 管脚描述

管脚名称	类型	接口功能名称		描述
		SPI	SSI	
SCK	I/O	SCK	CLK	串行时钟。SCK/CLK 是用于同步数据传输的时钟信号。由主机驱动, 从机接收。当使用 SPI 接口时, 可将时钟编程为高电平有效或低电平有效, 否则, 它一直是高电平有效。SCK 只在数据传输期间跳变。在其它时间, SPI 接口使其保持非工作状态或不驱动它。
SSEL	I/O	SSEL	FS	帧同步/ 从机选择。当 SPI 接口为总线主机时, 它在串行数据发起前将该信号驱动到工作状态, 再在发送数据后将信号释放到非工作状态。该信号为高电平有效还是低电平有效取决于所选择的总线和模式。当 SPI 接口为总线从机时, 该信号根据使用的协议限定从主机发出的数据。当只有一个总线主机和一个总线从机时, 来自主机的帧同步或从机选择信号可直接连接到从机的相应输入。当总线上有多个从机时, 通常必需进一步限制其帧选择/ 从机选择输入, 以避免多个从机对传输作出响应。
MISO	I/O	MISO	DR(M) DX(S)	主机输入从机输出。MISO 信号将串行数据由从机传输到主机。当 SPI 是从机时, 从该信号上输出串行数据。当 SPI 为主机时, 它记录从该信号发出的串行数据。当 SPI 为从机, 且未被 FS/SSEL 选择时, 它不会驱动该信号。
MOSI	I/O	MOSI	DX(M) DR(S)	主机输出从机输入。MOSI 信号将串行数据从主机传输到从机。当 SPI 为主机时, 从该信号上输出串行数据。当 SPI 为从机时, 它记录从该信号发出的串行数据。

11.3 寄存器描述

Table 11-2: 寄存器一览 (基地址 0x4004 0000)

名称	读写	偏移地址	描述	初始值
CR0	R/W	0x000	控制寄存器 0。选择串行时钟频率、总线类型和数据大小。	0x0
CR1	R/W	0x004	控制寄存器 1。选择主机/从机及其他模式。	0x0
DR	R/W	0x008	数据寄存器。写满发送 FIFO，读空接收 FIFO。	0x0
SR	RO	0x00C	状态寄存器	0x0000 0003
CPSR	R/W	0x010	SPI 时钟分频器寄存器	0x0
IMSC	R/W	0x014	中断屏蔽控制寄存器	0x0
RIS	RO	0x018	原始中断状态 寄存器	-
MIS	RO	0x01C	中断寄存器	0x0000 0008
ICR	WO	0x020	中断清除寄存器	NA

11.3.1 SPI 控制寄存器 0

此寄存器控制 SPI 的基本操作。

Table 11-3: SPI 控制寄存器 0 (CR0 - 地址 0x4004 0000) 描述

位	名称	值	描述	初始值
3:0	DSS		数据大小选择。该字段控制每帧中传输的位数。	0000
		0x0~0x2	保留	
		0x3	4-位传输	
		0x4	5-位传输	
		0x5	6-位传输	
		0x6	7-位传输	
		0x7	8-位传输	
		0x8	9-位传输	
		0x9	10-位传输	
		0xA	11-位传输	
		0xB	12-位传输	
		0xC	13-位传输	
		0xD	14-位传输	
		0xE	15-位传输	
		0xF	16-位传输	
5:4	FRF		帧格式	00
		0x0	SPI	
		0x1	SSI	
		0x2	保留	

		0x3	保留	
6	CPOL		时钟极性控制。该位只用于 SPI 模式。	0
		0	SPI 控制器使帧之间的总线时钟保持为低电平。	
		1	SPI 控制器使帧之间的总线时钟保持为高电平。	
7	CPHA		时钟相位控制。该位只用于 SPI 模式。	0
		0	SPI 控制器在帧传输的第一次时钟跳变时捕获串行数据。	
		1	SPI 控制器在帧传输的第二次时钟跳变时捕获串行数据。	
31:8		-	保留。	

11.3.2 SPI 控制寄存器 1

此寄存器控制 SPI 控制器操作的某些方面。

Table 11-4: SPI 控制寄存器 1 (CR1 - 地址 0x4004 0004) 描述

位	名称	值	描述	初始值
0	LBM		测试模式设置。	0
		0	正常操作模式。	
		1	测试模式。串行输入取自串行输出 (MISO)，而不是串行输入管脚 (MOSI)。	
1	SSE		SPI 使能	0
		0	SPI 禁止	
		1	SPI 控制器可与串行总线上的其他设备相互通信。设置此位前，软件应向 SPI 其他寄存器和中断控制器寄存器写入适当的控制信息。	
2	MS		主/从机选择。 SSE 位置位后有效。	0
		0	SPI 控制器作为总线上的主机，驱动 SCLK、MOSI 和 SSEL 线并接收 MISO 线。	
		1	SPI 控制器作为总线上的从机，驱动 MISO 线并接收 SCLK、MOSI 及 SSEL 线。	
3	SOD		从机输出禁用。只有在从机模式下才与此位有关 (MS = 1)。如果值为 1，则禁止此 SPI 控制器驱动发送数据线 (MISO)。	0
4	CSFL		SPI SSEL 控制	0
		0	SSEL 在帧传输间歇被强制为高电平。	
		1	SSEL 在帧传输间歇被强制为低电平当 SSE=1 时。	
5	RSFR		清除接收 FIFO。	0
		0	停止清除 FIFO。	
		1	清除 FIFO。	
6	FILTEN		SPI 数据线滤波使能。	0
		0	不使用滤波。	
		1	滤波使能。	
31:7			保留	NA

11.3.3 SPI 数据寄存器

软件可向该寄存器写入要发送的数据，并读取已接收的数据。

Table 11-5: SPI 数据寄存器 (DR - 地址 0x4004 0008) 描述

位	名称	描述	初始值
15:0	DATA	<p>写：当状态寄存器中的 TNF 位为 1（指示 Tx FIFO 未空）时，软件就可以将要在后续帧中发送的数据写入该寄存器。如果 Tx FIFO 先前为空且总线上的 SPI 控制器空闲，则将立即开始发送数据。否则，一旦先前所有的数据都已发送（和接收），即会发送写入该寄存器的数据。如果数据长度小于 16 位，则软件必须使写入该寄存器的数据向右对齐。</p> <p>读：只要状态寄存器中的 RNE 位为 1（指示 Rx FIFO 未空），软件就可以从该寄存器读取数据。当软件读取该寄存器时，SPI 控制器返回 Rx FIFO 中最早接收到的帧数据。如果数据长度小于 16 位，则使此字段的数据向右对齐，更高阶位用 0 填充。</p>	0x0000
31:16	-	保留	-

11.3.4 SPI 状态寄存器

该只读寄存器反映 SPI 控制器的当前状态。

Table 11-6: SPI 状态寄存器 (SR - 地址 0x4004 000C) 描述

位	名称	描述	初始值
0	TFE	发送 FIFO 为空。如果发送 FIFO 为空，则该位为 1；反之为 0。	1
1	TNF	发送 FIFO 未空。如果 Tx FIFO 已满，则该位为 0；反之为 1。	1
2	RNE	接收 FIFO 未空。如果接收 FIFO 为空，则该位为 0；反之为 1。	0
3	RFF	接收 FIFO 满。如果接收 FIFO 已满，则该位为 1；反之为 0。	0
4	BSY	忙。如果 SPI 控制器空闲，则该位为 0；或如果当前正在发送/接收一个帧和/或 Tx FIFO 未空，则该位为 1。	0
31:5	-	保留	NA

11.3.5 SPI 时钟寄存器

该寄存器用于产生 SPI 时钟频率。SPI 时钟频率通过对 SPI 外设时钟（SPI_PCLK）进行分频而取得。

Table 11-7: SPI 时钟预分频寄存器 (CPSR - 地址 0x4004 0010) 描述

位	名称	描述	初始值
7:0	CPSDVSR	<p>此偶数值介于 3 至 255 之间，SPI_PCLK 通过该值进行分频以产生预分频器输出时钟。位 0 始终读取为 0。</p> <p>如 CR1 寄存器的 FILTEN 位置位（使能数据滤波），CPSDVSR 值在主机模式下则必须大于 7。</p>	0
31:8		保留	

注: CPSR 值必须适当地进行初始化，否则控制器不能正确发送数据。

11.3.6 SPI 中断屏蔽控制寄存器

该寄存器控制是否启用 SPI 控制器中 4 个可能的中断条件。

Table 11-8: SPI 中断屏蔽控制寄存器 (IMSC - 地址 0x4004 0014) 描述

位	名称	描述	初始值
0	RORIM	当 Rx FIFO 已满时，此位将产生中断	0
1	RTIM	当 Rx FIFO 不为空时，会产生中断	0
2	RXIM	Rx FIFO 至少有一半为满时，如此位为 1 将产生中断。	0
3	TXIM	Tx FIFO 至少有一半为空时，如此位为 1 将产生中断。	0
31:4	-	保留	NA

11.3.7 SPI 原始中断状态寄存器

不论 IMSC 中是否产生中断，只要出现有效的中断条件，该只读寄存器便会在相应的位置包含 1。

Table 11-9: SPI 原始中断状态寄存器 (RIS - 地址 0x4004 0018) 描述

位	名称	描述	初始值
0	RORRIS	如果在 Rx FIFO 已满时，则该位为 1。	0
1	RTRIS	当 Rx FIFO 不为空时，则该位为 1	0
2	RXRIS	如果 Rx FIFO 至少有一半为满时，则该位为 1。当 CPU 读取后，少于一半时，该位自动变为 0	0
3	TXRIS	如果 Tx FIFO 至少有一半为空时，则该位为 1。当 CPU 写入后，多于一半时，该位自动变为 0	1
31:4	-	保留	NA

11.3.8 SPI 中断状态寄存器

该只读寄存器。当中断条件出现且相应的中断在 IMSC 中使能时，该寄存器中对应的位为 1。当出现 SPI 中断时，中断服务程序可读取此寄存器以确定中断的原因。

Table 11-10: SPI 中断状态寄存器 (MIS - 地址 0x4004 001C) 描述

位	名称	描述	初始值
0	RORMIS	如果在 Rx FIFO 已满时，则该位为 1。	0
1	RTMIS	当 Rx FIFO 不为空时，则该位为 1	0
2	RXMIS	如果 Rx FIFO 至少有一半为满，且此中断启用，则此位为 1。	0
3	TXMIS	如果 Tx FIFO 至少有一半为空，且此中断启用，则此位为 1。	0
31:4	-	保留	NA

11.3.9 SPI 中断清除寄存器

软件可向该只写寄存器相应位写入 1 以清除 SPI 控制器的中断。对应 FIFO 空/满中断，可通过写入或读取相应的 FIFO 清除，或通过清除 IMSC 中的相应位将其禁用。

Table 11-11: SPI 中断清除寄存器 (ICR - 地址 0x4004 0020) 描述

位	名称	描述	初始值
---	----	----	-----

0	RORIC	向该位写入 1 可清除 Rx FIFO 为满时接收帧中断。	NA
1	RTIC	向该位写入 1 可清除 Rx FIFO 不为空中断。	NA
31:2	-	保留	NA

11.4 操作

11.4.1 SPI 帧数据格式

SPI 接口是 4 线接口，其中 SSEL 信号用作从机选择。SPI 格式的主要特性是 SCK 信号的非工作状态和相位可通过对 SPICR0 控制寄存器内的 CPOL 和 CPHA 位编程设定。

时钟极性 (CPOL) 和相位 (CPHA) 控制

当没有传输数据时，如 CPOL 时钟极性控制位为 0，它会在 SCK 管脚产生一个稳态的低电平。如 CPOL 时钟极性控制位为 1，则它会在 CLK 管脚上产生一个稳态高电平。

CPHA 控制位用于设定捕获数据及允许数据更改状态的时钟沿。当 CPHA 相位控制位为 0 时，则在第一次出现时钟沿跳变时捕获数据。如果 CPHA 时钟相位控制位为 1 时，则在第二次出现时钟沿跳变时捕获数据。

CPOL=0, CPHA=0 时的 SPI 格式

CPOL = 0 且 CPHA = 0 时 SPI 格式的单帧和连续帧传输信号序列如下图所示。

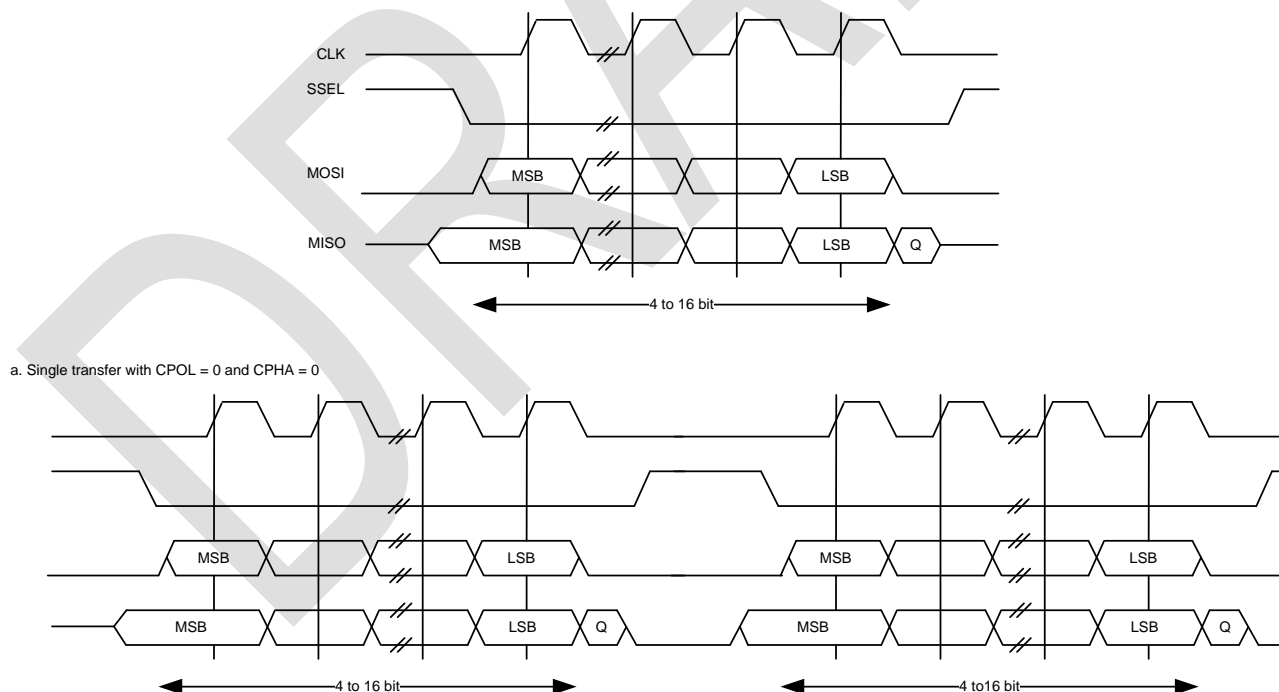


Figure 11-1: CPOL=0 且 CPHA=0 时的 SPI 帧格式 (a) 单帧和 b) 连续帧传输

该配置中，在空闲期间：

- CLK 信号被强制为低电平。

- SSEL被强制为高电平（CSFL=0）。
- MOSI/MISO 空闲时为高阻状态。

如果启用 SPI，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。这样，从机数据就可以进入到主机的 MISO 输入线中。启用主机的 MOSI，1/2 个 SCK 周期后，主机有效的数据将被传输到 MOSI 管脚。由于主机和从机数据均已设定，因此再过 1/2 个 SCK 周期，SCK 主时钟管脚就会变为高电平。此时，在 SCK 信号的上升沿捕获数据，并在 SCK 信号的下降沿传播数据。发送单个字时，在传输完数据字的所有位后，在捕获到最后一位后的一个 SCK 周期内，SSEL 线将返回到其空闲高电平状态如果 CSFL=0。在 CSFL=1 时，SSEL 线将保持低电平。

CPOL=0, CPHA=1时的SPI格式

CPOL = 0 且 CPHA = 1 时 SPI 格式的传输信号序列如下图所示，包含单帧和连续帧传输。

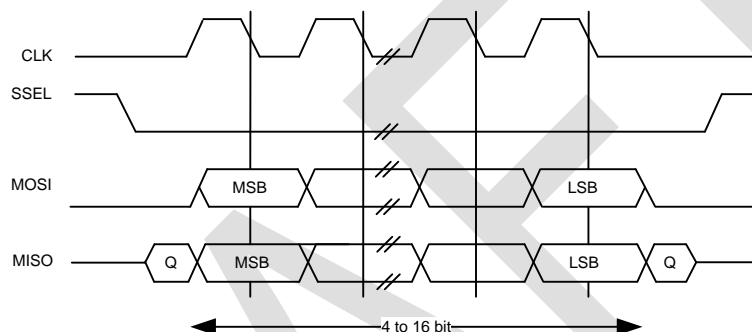


Figure 11-2: SPI 在 CPOL = 0 和 CPHA = 1 时的数据格式

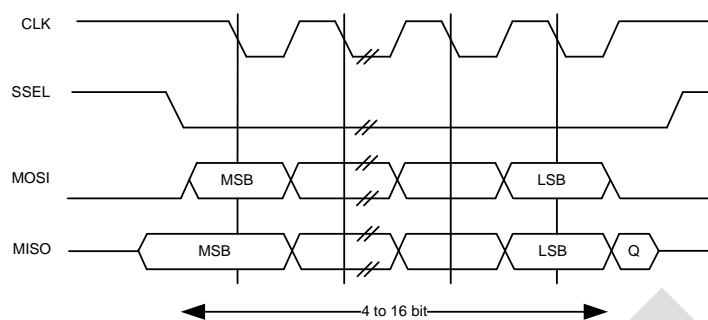
该配置中，在空闲期间：

- CLK 信号被强制为低电平。
- SSEL被强制为高电平（CSFL=0）。
- MOSI/MISO 空闲时为高阻状态。

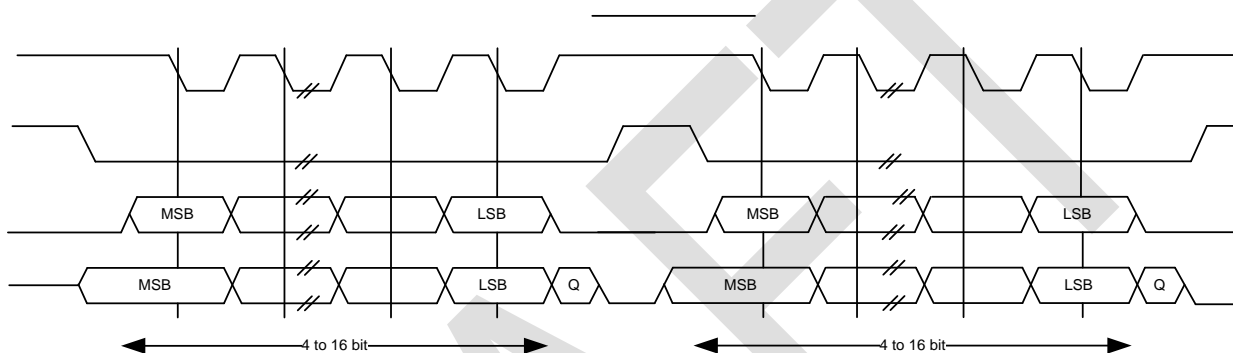
如果启用 SPI，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。启用主机的 MOSI 管脚。再过 1/2 个 SCK 周期后，主机和从机的有效数据都将在其各自的传输线上启用。同时，SCK 在上升沿跳变时启用。然后，在 SCK 信号的下降沿捕获数据，并在 SCK 信号的上升沿传播数据。发送单个字时，在传输完所有位后，在捕获到最后一位后的一个 SCK 周期内，SSEL 线返回其空闲高电平状态如果 CSFL=0。在 CSFL=1 时，SSEL 线将保持低电平。

CPOL = 1, CPHA = 0时的SPI格式

CPOL=1 且 CPHA=0 时，SPI 格式的单帧和连续帧传输信号序列如下图所示。



a. Single transfer with CPOL = 1 and CPHA = 0



b. Continuous transfer with CPOL = 1 and CPHA = 0

Figure 11-3: SPI 在 CPOL = 1 和 CPHA = 0 下 (a) 单帧和(b) 连续帧传输

该配置中，在空闲期间：

- CLK 信号被强制为高电平。
- SSEL被强制为高电平（CSFL=0）。
- MOSI/MISO 空闲时为高阻状态。

如果启用 SPI，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始，这会使从机数据立即传输到主机 MISO 线上。启用主机的 MOSI 管脚。1/2 个 SCK 周期后，有效的主机数据将被传输到 MOSI 线。由于主机和从机数据均已设定，因此再过 1/2 个 SCK 周期，SCK 主时钟管脚就会变为低电平。这意味着，在 SCK 信号的下降沿捕获数据并在 SCK 信号的上升沿传播数据。发送单个字时，在传输完所有位后，在捕获最后一位后的一个 SCK 周期内，SSEL 线将返回到其空闲高电平状态如果 CSFL=0。在 CSFL=1 时，SSEL 线将保持低电平。

CPOL = 1, CPHA = 1时的SPI格式

CPOL = 1 且 CPHA = 1 时 SPI 格式的传输信号序列如下图所示，包含单帧和连续帧传输。

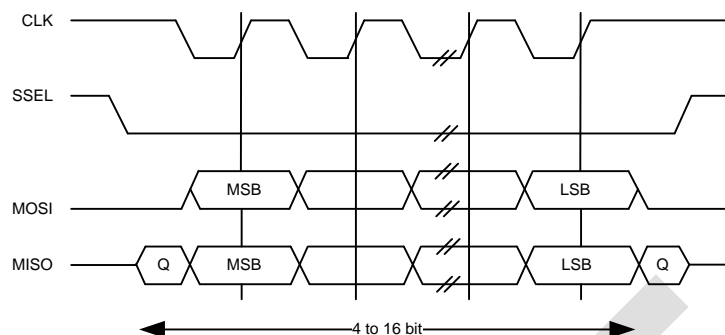


Figure 11-4: SPI 在 CPOL = 1 和 CPHA = 1 时的数据格式

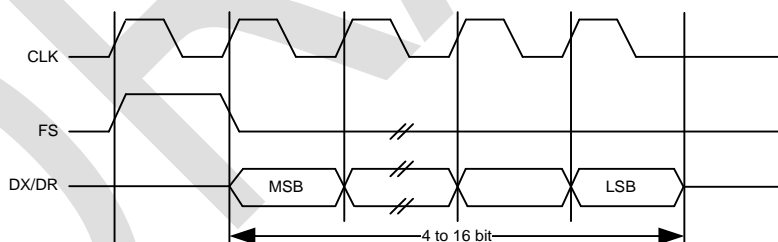
该配置中，在空闲期间：

- CLK 信号被强制为高电平。
- SSEL被强制为高电平（CSFL=0）。
- MOSI/MISO 空闲时为高阻状态。

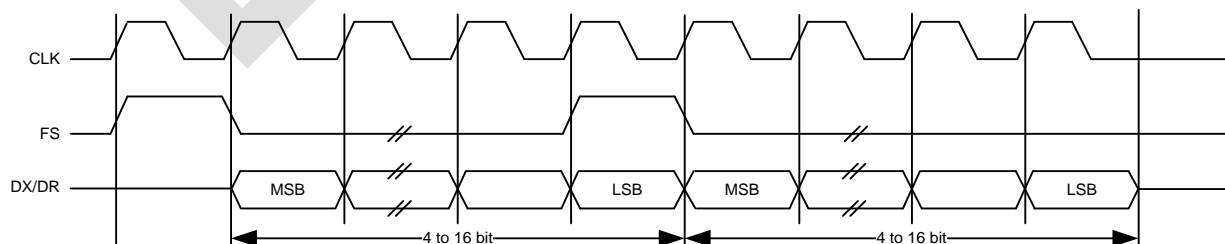
如果启用 SPI，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。启用主机的 MOSI。再过 1/2 个 SCK 周期后，主机和从机数据都将在其各自的传输线上启用。同时，SCK 在下降沿跳变时启用。然后，在 SCK 信号的上升沿捕获数据，并在 SCK 信号的下降沿传播数据。发送单个字时，在传输完所有位后，在捕获最后一位后的一个 SCK 周期内，SSEL 线返回到其空闲高电平状态如果 CSFL=0。在 CSFL=1 时，SSEL 线将保持低电平。

11.4.2 SSI 帧数据格式

下图显示了扩展SPI模块支持的4线同步串行帧格式。



a. Single frame transfer



b. Continuous/back-to-back frames transfer

Figure 11-5: 4线同步串行帧格式: a) 单帧和b) 连续/ 背靠背2 帧传输

对于在该模式下配置为主机的设备，CLK 和 FS 被强制为低电平，且只要扩展 SPI 空闲，发送数据线 DX 便会处于高阻。一旦发送 FIFO 的底端含有数据，FS 就会变为高电平，并持续一个 CLK 周期。要发送的值也会从发送 FIFO 传输到发送逻辑的串行移位寄存器。在下一个 CLK 上升沿上，4 位到 16 位数据帧的 MSB 输出到 DX 管脚。同样，接收数据的 MSB 由片外串行从器件传送到 DR 管脚。在每个 CLK 的下降沿，SPI 和片外串行从器件将各个数据位放入其串行移位器。LSB 被锁存后，在 CLK 的第一个上升沿，接收的数据从串行移位器传输到接收 FIFO。

12 I2C

12.1 概述

I2C 是两线串行通信接口，它与 I²C 总线兼容。可以支持主、从机两种模式的 I²C 通信方式。主要功能如下：

- 兼容标准的 I²C 主从机工作模式
- 可编程到时钟频率支持不同的 I2C 数据传输速率
- 支持从机工作模式下双向数据传输
- 串行时钟同步使得传输速率不同的器件能够通过一条串行总线进行通信
- 支持达 3MHz 速率
- 可设定为多达 4 个不同的从机地址
- I2C 总线可用于测试和诊断

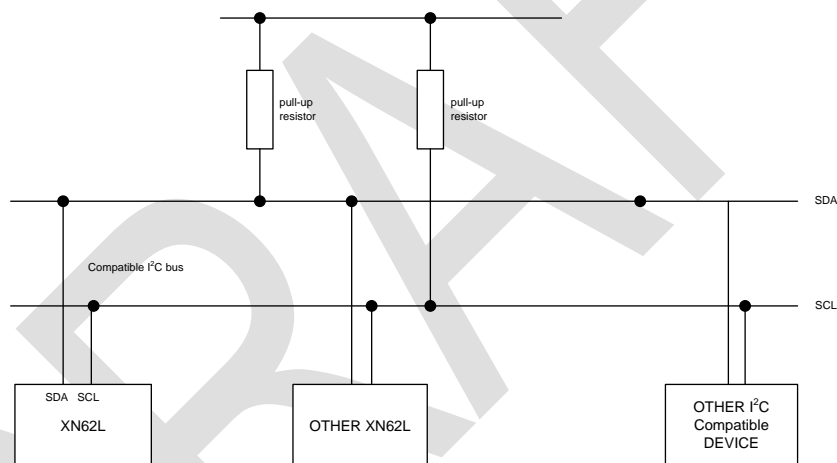


Figure 12-1: I2C 总线应用框图

12.2 管脚描述

Table 12-1: I2C 总线管脚 描述

管脚	类型	描述
SDA	输入/输出	I2C 串行数据
SCL	输入/输出	I2C 串行时钟

12.3 寄存器描述

Table 12-2: 寄存器一览 (基地址 0x4000 0000)

名称	读写	偏移地址	描述	初始值
CONSET	R/W	0x000	I2C 控制置位寄存器。当向该寄存器的位写 1 时，I2C 控制寄存器中的相应位置位。写 0 无效。	0x00
STAT	RO/WO	0x004	I2C 状态寄存器。在 I2C 工作期间，该寄存器提供详细的状态码，允许软件决定需要执行的下一步操作。	0x1F
DAT	R/W	0x008	I2C 数据寄存器。在主/从发送模式期间，要发送的数据写入该寄存器。在主/从接收模式期间，可从该寄存器读出已接收的数据。	0x00
ADR0	R/W	0x00C	I2C 从机地址寄存器 0。包含 7 位从属地址，用于从机模式下 I2C 接口操作，不用于主机模式下。	0x00
SCLH	R/W	0x010	占空比寄存器高半字。决定 I2C 时钟的高电平时间。	0x04
SCLL	R/W	0x014	占空比寄存器低半字。决定 I2C 时钟的低电平时间。	0x04
CONCLR	WO	0x018	I2C 控制清零寄存器。当向该寄存器的位写 1 时，I2C 控制寄存器中的相应位清零。写 0 时对 I2C 控制寄存器的相应位没有影响。	NA
-	-	0x01C	保留	0x00
ADR1	R/W	0x020	I2C 从机地址寄存器 1。包含 7 位从属地址，用于从机模式下 I2C 接口操作，不用于主机模式下。	0x00
ADR2	R/W	0x024	I2C 从机地址寄存器 2。包含 7 位从属地址，用于从机模式下 I2C 接口操作，不用于主机模式下。	0x00
ADR3	R/W	0x028	I2C 从机地址寄存器 3。包含 7 位从属地址，用于从机模式下 I2C 接口操作，不用于主机模式下。	0x00
DATA_BUFFER	RO	0x02C	数据缓冲寄存器。每次从总线接收到 9 个位(8 位数据和 ACK 或 NACK)后，DAT 移位寄存器的高 8 位的内容将自动传输到 DATA_BUFFER。	0x00
MASK0	R/W	0x030	I2C 从属地址屏蔽寄存器 0。该屏蔽寄存器与 ADR0 一起决定地址匹配。	0x00
MASK1	R/W	0x034	I2C 从属地址屏蔽寄存器 1。该屏蔽寄存器与 ADR0 一起决定地址匹配。	0x00
MASK2	R/W	0x038	I2C 从属地址屏蔽寄存器 2。该屏蔽寄存器与 ADR0 一起决定地址匹配。	0x00
MASK3	R/W	0x03C	I2C 从属地址屏蔽寄存器 3。该屏蔽寄存器与 ADR0 一起决定地址匹配。	0x00

12.3.1 I2C 控制置位寄存器

CONSET 寄存器控制 I2C 接口的操作。向该寄存器的位写 1 会使 I2C 控制寄存器中的相应位置位。写 0 无效。

Table 12-3: I2C 控制置位寄存器 (CONSET - 地址 0x4000 0000) 描述

位	名称	描述	初始值
0	TXRX	发送/接受标志	0x0
1	MASL	主/从机标志	0x0
2	AA	声明应答标志	0x0
3	SI	I2C 中断标志	0x0

4	STO	停止标志	0x0
5	STA	起始标志	0x0
6	I2CEN	I2C 接口使能	0x0
31:7	-	保留	0x0

TXRX: 发送/接受标志

- 当 TXRX 置 1 时, I2C 接口工作在发送方式
- 当 TXRX 为 0 时, I2C 接口工作在接收方式

MASL: 主/从机标志

- 当 MASL 置 1 时, I2C 接口工作在主机模式
- 当 MASL 为 0 时, I2C 接口工作在从机模式

AA: 应答标志位

- 当 AA 置 1 时, 在 SCL 线的应答时钟脉冲内, 出现下面的任意情况时都将返回一个应答信号 (SDA 线为低电平):
 1. 接收到从属地址寄存器中的地址。
 2. 当 I2C 接口处于主机接收模式时, 接收到一个数据字节。
 3. 当 I2C 接口处于从机接收模式时, 接收到一个数据字节。
- 可通过向 CONCLR 寄存器中的 AAC 位写 1 来清除 AA 位。当 AA 位为 0 时, SCL 线上的应答时钟脉冲内出现下列情况时将返回一个无应答信号 (SDA 为高电平):
 1. 当 I2C 接口处于主机接收模式时, 接收到一个数据字节。
 2. 当 I2C 接口处于从机接收模式时, 接收到一个数据字节。

SI: 中断标志

- 当 I2C 状态改变时该位置位。当 SI 置位时, SCL 线上的串行时钟低电平持续时间扩展, 且串行传输被中止。当 SCL 为高时, 它不受 SI 标志的状态影响。SI 必须通过软件复位, 通过向 CONCLR 寄存器的 SIC 位写入 1 来实现。

STO: 停止标志

- 在主机模式下, 该位置位会使 I2C 接口发送一个停止条件, 或在从机模式下从错误状态中恢复。当主机模式下 STO=1 时, 向 I2C 总线发送停止条件。当总线检测到停止条件时, STO 自动清零。从机模式下, 置位 STO 位可从错误状态中恢复。这种情况下不向总线发送停止条件。硬件的表现就好像是接收到一个停止条件并切换到“不可寻址”的从接收模式。STO 标志由硬件自动清零。

STA: 起始标志

- 该位置位时, I2C 接口进入主机模式并发送一个起始条件, 如果已经处于主机模式, 则发送一个重复起始条件。当 STA 为 1 且 I2C 接口没有处于主机模式时, 它将进入主机模式, 检测总线, 并在总线空闲时产生一个起始条件。如果总线忙, 则等待一个停止条件 (将释放总线) 并在延迟半个内部时钟发生器周期后发送一个起始条件。当 I2C 接口已经处于主机模式且已发送或接收了数据时, 它会发送一个重复起始条件。STA 可在任意时间置位, 包括 I2C 接口处于可寻址的从机模式时也可置位。可通过向 CONCLR 寄存器中的 STAC 位写 1 来清零 STA。当 STA 为 0 时, 不会产生起始条件或重复起始条件。STA 和 STO 都置位时, 如果接口处于主机模式下, 则向 I2C 总线发送一个停止条件, 然后再发送一个起始条件。如果 I2C 接口处于从模式, 则产生内部停止条件, 但不发送到总线上。

I2CEN: I2C 接口使能

- 当I2CEN 为1 时， I2C接口使能。可通过向CONCLR 寄存器中的I2CENC 位写1 来清零I2CEN 位。当I2CEN 为0 时， I2C接口禁能。
- 当I2CEN 为0 时，忽略SDA 和SCL 输入信号，I2C 块处于“不可寻址”的从状态，STO位强制为0。I2CEN 不应用于暂时释放I2C 总线，因为当I2CEN 复位时，I2C 总线状态丢失。应使用AA 标志代替。

12.3.2 I2C 状态寄存器

每个 I2C 状态寄存器反映相应 I2C 接口的情况。只读寄存器

Table 12-4: I2C 状态寄存器 (STAT - 0x4000 0004) 描述

位	名称	描述	初始值
6:0	STATUS	该域通过 I2C 的实际状态信息。参照 Table 12-14~Table 12-17.	0x1F
7	-	保留	0
8	SLVADDMATCH	从机地址已匹配。	0
9	SLVRXBUFFULL	从机接收缓冲器满。读取数据寄存器(偏移地址: 0x08)自动清除该位。	0
10	SLVTXBUFEMPTY	从机发送缓冲器空。写数据寄存器(偏移地址: 0x08)自动清除该位。	0
31: 11	-	保留	NA

12.3.3 I2C 数据寄存器

该寄存器包含要发送的数据或刚接收的数据。SI 位置位时，只有在该寄存器没有进行字节移位时，CPU 才可以对其进行读/ 写操作。只要 SI 位置位，DAT 中的数据就保持不变。DAT 中的数据总是从右向左移位：要发送的第一位是 MSB（位 7），接收到一个字节后，接收到数据的第一位放在 DAT 的 MSB 位。

Table 12-5: I2C 数据寄存器 (DAT - 0x4000 0008) 描述

位	名称	描述	初始值
7:0	DATA	该寄存器保存已接收或将要发送的数据值	0
31: 8	-	保留	-

12.3.4 I2C 从机地址寄存器 0~3

4 个寄存器可读/ 写，只有在 I2C 接口设置为从机模式时才可用。在主机模式下，该寄存器无效。

Table 12-6: I2C 从机地址 寄存器 0 (ADR0- 0x4000 000C, ADR1 - 0x4000 0020, ADR2 - 0x4000 0024, ADR3 -0x4000 0028)) 描述

位	名称	描述	初始值
0		保留	0
7:1	ADDR	从机模式的 I2C 器件地址	0x00
31: 8	-	保留	-

12.3.5 I2C 高电平占空比寄存器

Table 12-7: I2C 高电平占空比寄存器 (SCLH - 地址 0x4000 0010) 描述

位	名称	描述	初始值
15:0	SCLH	SCL 高电平周期选择	0x0004
31: 16	-	保留	-

12.3.6 I2C 低电平占空比寄存器

Table 12-8: I2C 低电平占空比寄存器 (SCLL - 地址 0x4000 0014) 描述

位	名称	描述	初始值
15:0	SCLL	SCL 低电平周期选择	0x0004
31: 16	-	保留	-

注:

软件必须设定寄存器 SCLH 和 SCLL 的值以选择适当的数据速率和占空比。SCLH 定义了 SCL 高电平期间 I2C_PCLK 的周期数, SCLL 定义了 SCL 低电平期间 I2C_PCLK 的周期数。频率由下面公式得出 (I2C_PCLK 是外设总线 APB 的频率):

$$I2C \text{ 位速率} = I2C_PCLK / (SCLH + SCLL)$$

12.3.7 I2C 控制清除寄存器

CONCLR 寄存器控制对 CON 寄存器中的位清零, 该寄存器控制 I2C 接口的操作。向该寄存器的位写 1 会使 I2C 控制寄存器中的相应位清零。写 0 无效。

Table 12-9: I2C 控制清除寄存器 (CONCLR - 0x4000 0018) 描述

位	名称	描述	初始值
0	TXRX	发送/接收选择清除位。写 1 清除 I2CONSET 的 TXRX 位。写 0 无效。.	NA
1	MASL	主/从机选择清除位。写 1 清除 I2CONSET 的 MASL 位。写 0 无效。.	0
2	AAC	应答声明清除位。写 1 清除 I2CONSET 的 AA 位。写 0 无效。	0
3	SIC	中断清除位。写 1 清除 I2CONSET 的 SI 位。写 0 无效。	0
4	-	保留。	NA
5	STAC	起始标志清除位。写 1 清除 I2CONSET 的 STA 位。写 0 无效。	0
6	I2CENC	I2C 接口使能清除位。写 1 清除 I2CONSET 的 I2CEN 位。写 0 无效。	0
7	-	保留	NA
8	ADRMATC	从机地址匹配标志清除。写 1 清除 STAT 寄存器的 SLVADDMATCH 位。写 0 无效	0
31: 9	-	保留	-

12.3.8 I2C 数据缓冲寄存器

Table 12-10: I2C 数据缓冲寄存器 (DATA_BUFFER - 0x4000 002C) 描述

位	名称	描述	初始值
7:0	DATA	该寄存器保存 DAT 数据寄存器中高 8 位的内容	0
31: 8	-	保留	-

12.3.9 I2C 屏蔽寄存器 0~3

4 个屏蔽寄存器各包含 7 个有效位（7:1）。当与屏蔽寄存器关联的 ADR_n 寄存器比较时，这些寄存器中的任一位置‘1’都会使接收地址的相应位自动比较。也就是说，决定地址匹配时不考虑 ADR_n 寄存器中被屏蔽的位。复位时，所有屏蔽寄存器位清零。

Table 12-11: I2C 屏蔽寄存器 (MASK0 - 0x4000 0030, MASK1 - 0x4000 0034, MASK2 - 0x4000 0038, MASK3 - 0x4000 003C) 描述

位	名称	描述	初始值
0	-	保留.	0
7:1	MASK	屏蔽位	0x00
31: 8	-	保留.	0

12.4 I2C 操作

与 I^2C 应用一样，I2C 总线可以工作在主机和从机模式。在从机模式下，I2C 硬件时刻监视总线上是否出现 4 个已定义的从机地址。如地址被检测到，就会发出中断请求。如控制器设置为主机模式，I2C 硬件检测到总线空闲时才会进入到主机模式，从机通讯不会因此而中断。

12.4.1 主机发送模式

该模式下，数据由主机发送到从机。在进入主发送模式前，CONSET 寄存器必须被初始化。MASL 置 1 选择为主机，TXRX 置 1 选择发送模式。I2CEN 必须置 1 以使能 I2C 功能。如果 AA 位为 0，则当另一个器件为总线上的主机时，I2C 接口不会对任何地址作出应答，因此不能进入从机模式。STA、STO 和 SI 位必须为 0。通过向 CONCLR 寄存器中的 SIC 位写 1 来清零 SI 位。写从机地址后应清零 STA 位。STA、STO 和 SI 位必须为 0。SI 位可用写 1 到 CONCLR 寄存器的 SIC 来清除。写从机地址后应清零 STA 位。

Table 12-12: I2C CONSET 寄存器配置为主机发送模式

位	7	6	5	4	3	2	1	0
名称	-	I2CEN	STA	STO	SI	AA	MASL	TXRX
值	-	1	0	0	0	0	1	1

发送的第一个字节包含接收器件的从机地址（7 位）和数据方向位。在该模式下，数据方向位(R/W) 应为 0，表示执行写操作。发送的第一个字节包含从机地址和写操作位。一次发送 8 位数据。每发送完一个字节后，接收到一个应答位。输出起始和停止条件指示串行传输的起始和结束。转件置位 STA 位时，I2C 接口将进入主发送模式。一旦总线空闲，I2C 逻辑就会发送起始条件。发送起始条件后，SI 位置位，STAT 寄存器中的状态代码为 0x01。该状态代码引导状态服务程序，将从机地址和写操作位装入 DAT 寄存器，然后清零 SI 位。通过向 CONCLR 寄存器中的 SIC 位写入 1 清零 SI。

当已发送从机地址和 R/W 位并接收到应答位后，SI 位再次置位，此时，主机模式下可能的状态为 0x0b, 0x4b 或 0x14。参照 Table 12-14 到 Table 12-17。

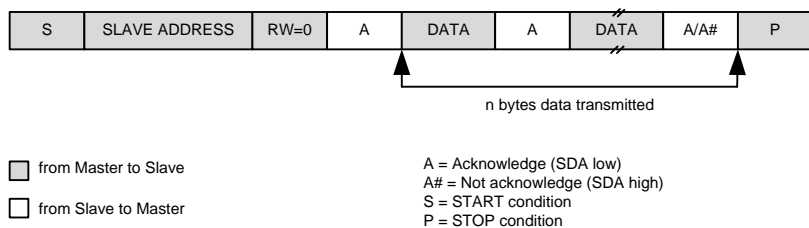


Figure 12-2:主机发送模式的格式

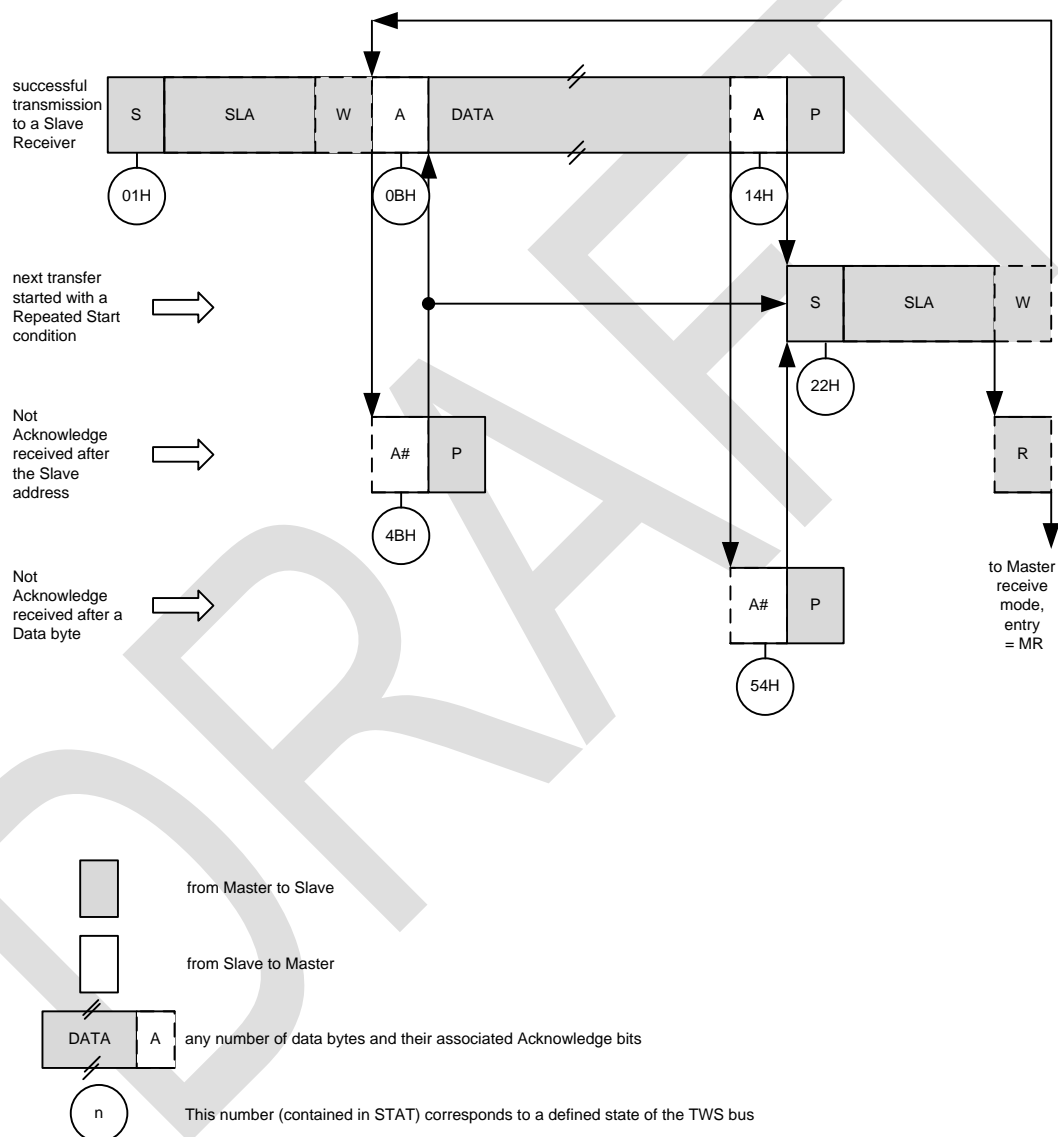


Figure 12-3:主机发送模式流程图

12.4.2 主机接收模式

在主接收模式中，主机所接收的数据字节来自从机。传输按照主接收模式中的情况初始化。当发送完起始条件后，中断服务程序必须把从机地址和数据方向位(SLA+R) 装入 DAT。必须先清零 CON 中的 SI 位，串行传输才能继续。当发送完从机地址和数据方向位且接收到一个应答位时，串行中断标志(SI) 再次置位，STAT 中可能是一系列不同的状态代码。主机模式下为

0x0b, 0x4b, 或 0x1d., 从机模式(AA = 1) 为 0x68、0x78 或 0xB0。表 Table 12-15 中详细介绍了每个状态代码对应的操作。当 I2C 需要应答一个接收到的字节时, 需在清零 SI 位并启动读取的字节之前先将 AA 位相应置位。当 I2C 需要不应答一个接收到的字节时, 需在清零 SI 位并启动读取的字节之前先将 AA 位清零。最后一个收到的字节后总是跟有 I2C 的“非应答”位, 以便主机能指示从机读取序列已完成而它需要发出停止或重复起始命令。发送“非应答”位且 SI 位置位后, 可发送停止 (STO 位置位) 或重复起始条件 (STA 位置位)。然后 SI 位清零以启动请求操作。

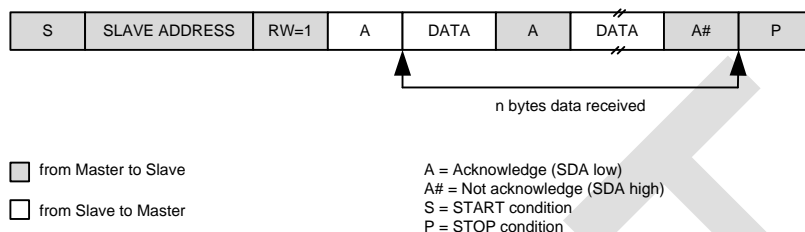


Figure 12-4: 主机接收模式的格式

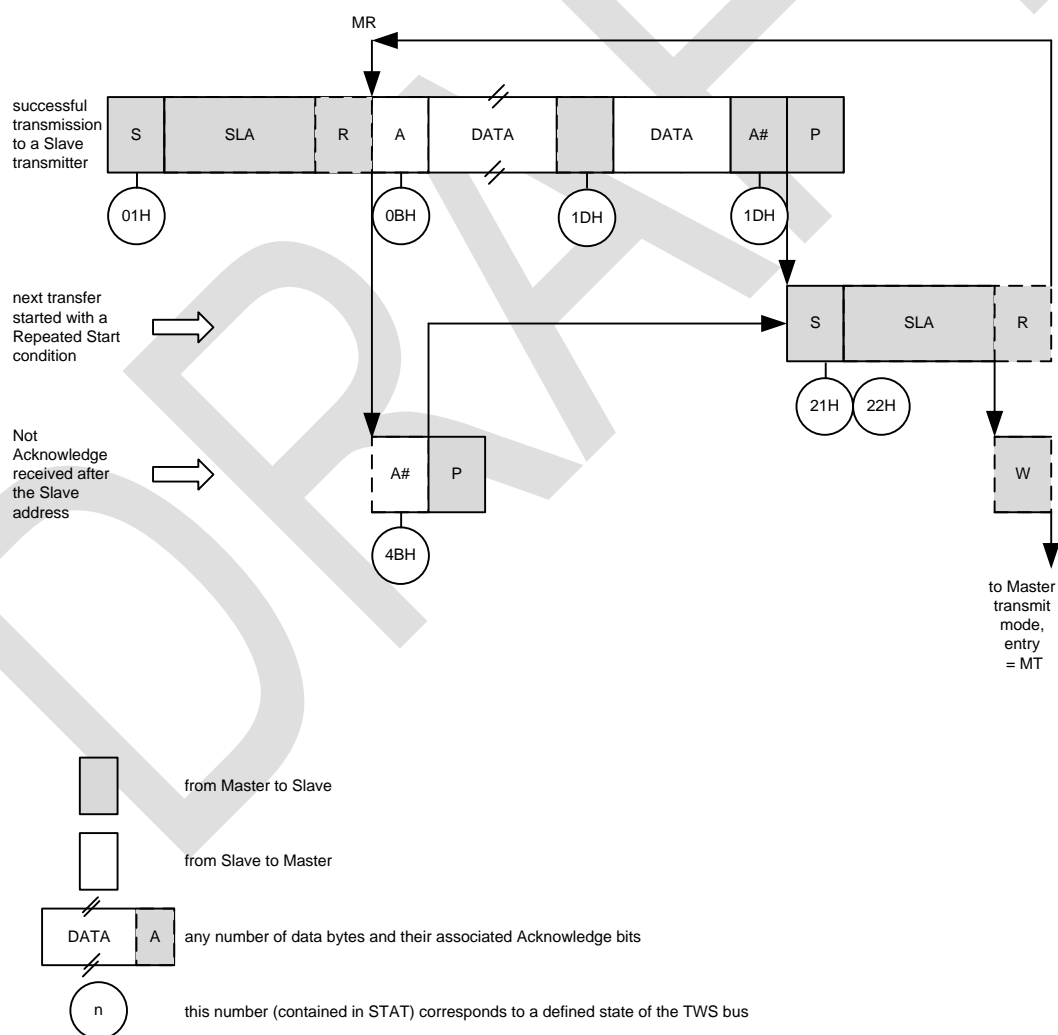


Figure 12-5: 主机接收模式流程图

12.4.3 从机接收模式

从机接收模式下, 从主机接收数据字节。要初始化从接收模式, 对任一从机地址寄存器(ADR0-3) 和从屏蔽寄存器(MASK0-3) 进行写操作并按下表所示写 I2C 控制置位寄存器(CONSET)。

Table 12-13: I2C CONSET 寄存器配置为从机接收模式

位	7	6	5	4	3	2	1	0
名称	-	I2CEN	STA	STO	SI	AA	MASL	TXRX
值	-	1	0	0	0	1	0	0

MASL 位必须为 0, 并且 I2CEN 需置位去使能 I2C 功能。AA 位必须置 1 以应答其自身的从机地址。STA、STO 和 SI 位清 0。初始化 ADR 和 CONSET 后, I2C 接口开始等待, 直到被其自身从机地址。如果方向位为 0(W), 则进入从接收模式。如果方向位为 1(R), 则进入从发送模式。接收到地址和方向位后, SI 位置位, 可从状态寄存器(STAT) 读取一个有效状态代码。关于状态低码和操作参照 Table 12-16。

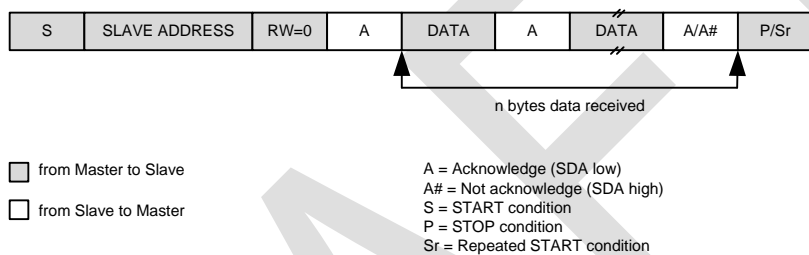


Figure 12-6: 从机接收模式下的格式

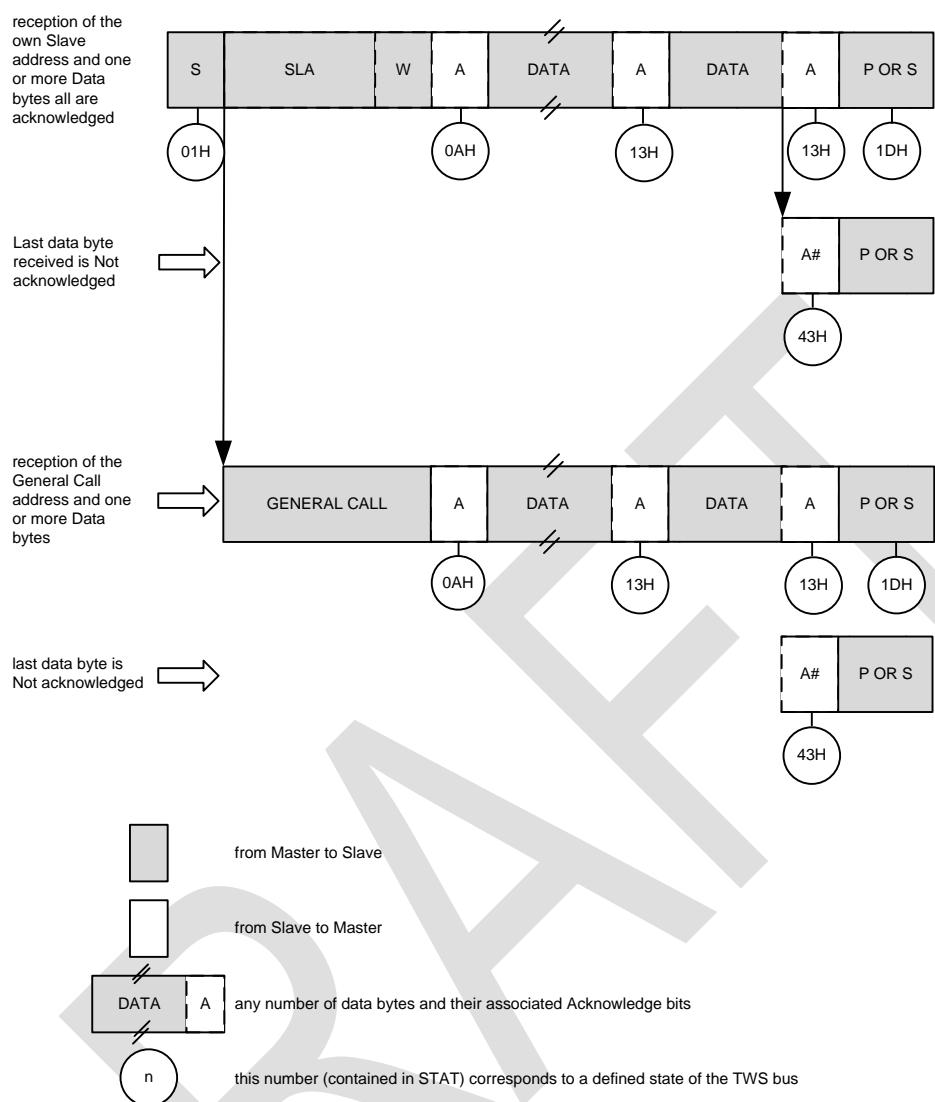


Figure 12-7: 从机接收模式使用状态码控制流程图

12.4.4 从机发送模式

接收和处理第一个字节的方式与从接收模式下相同。但是，在该模式下，方向位为 1，指示读操作。通过 SDA 发送串行数据，通过 SCL 输入串行时钟。起始和停止条件分别看作串行传输的开始和结束。在特定应用中，I2C 可作为主机/从机。在从机模式，I2C 硬件查找自身从机地址。如果检测到其中一个地址，则请求中断。如果微控制器想成为总线主机，则在进入主机模式前，硬件将一直等待，直到总线空闲，这样就不会中断可能存在的从机操作。

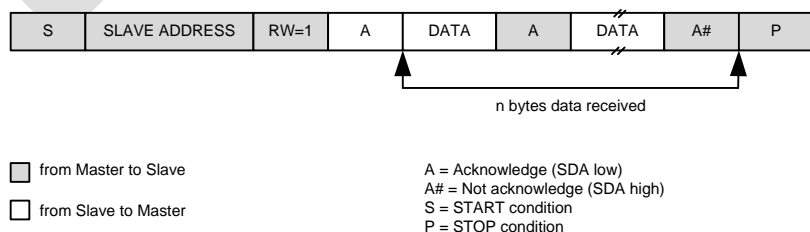


Figure 12-8: 从机发送模式下的格式

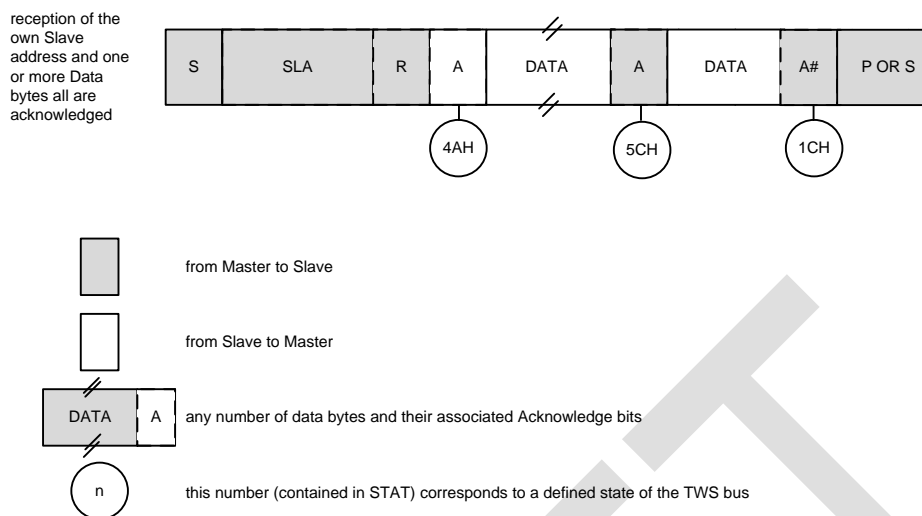


Figure 12-9: 从机发送模式使用状态码控制流程图

12.4.5 从机接收发送简化流程

对于 I2C 从机接收发送，G32F0XX 还提供一套简化的状态标志来实现 I2C 的接收和发送。该方法无需查看 I2C 的状态代码，只需查看状态寄存器的 SLVADDMATCH，SLVRXBUFFULL 和 SLVTXBUFEMPTY 标志位。I2C 从机初始化同上。I2C 从机启动后，等待主机通讯请求。当从机接收到与自己匹配地址，SLVADDMATCH 状态置 1 并引发中断。当从主机接收到数据时，SLVRXBUFFULL 标志位会被置 1 并产生中断，缓冲区数据读走后自动清零。如要发送数据，用户需在启动 I2C 后，存入发送数据，当数据被主机读走后，SLVTXBUFEMPTY 标志位被置 1 并产生中断，用户需填入下一个将发送的数据。中断服务流程图如下：

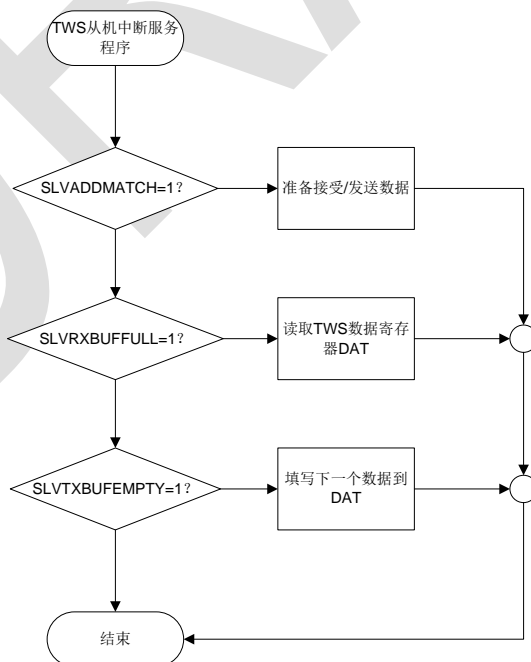


Figure 12-10: 从机接收发送模式中断服务流程图

12.4.6 状态明细表

下面列出了 I2C 在四种操作模式下的状态明细。

Table 12-14: 主机发送模式

状态代码	I2C 总线及硬件状态	应用软件响应						I2C 硬件工作的下一个步骤
		写/读 DAT	写 CON					
			STA	TXRX	STO	SI	AA	
0x01	已发送起始条件。	装入 SLA+W; 清除 STA	X	1	0	0	X	将发送 SLA+W ；接收 ACK 位。
0x22	已发送重复起始条件。	装入 SLA+W 或	X	1	0	0	X	同上。 .
		装入 SLA+R; 清除 STA	X	1	0	0	X	将发送 SLA+R; I2C 将切换为主机接收模式。
0x0b	已发送 SLA+W; 已接收 ACK。	装入 数据字节	0	1	0	0	X	将发送数据字节, 接收 ACK 位。
		无 DAT 操作	1	1	0	0	X	将发送重复的起始条件。
		无 DAT 操作	0	1	1	0	X	将发送停止条件; STO 标志将复位。
		无 DAT 操作	1	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将复位。
0x4b	已发送 SLA+W; 已接收非 ACK。	装入 数据字节	0	1	0	0	X	将发送数据字节, 接收 ACK 位。
		无 DAT 操作	1	1	0	0	X	将发送重复的起始条件。
		无 DAT 操作	0	1	1	0	X	将发送停止条件; STO 标志将复位。
		无 DAT 操作	1	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将复位。
0x14	已发送 DAT 中的数据字节; 已接收 ACK。	装入 数据字节	0	1	0	0	X	将发送数据字节, 接收 ACK 位。
		无 DAT 操作	1	1	0	0	X	将发送重复的起始条件。
		无 DAT 操作	0	1	1	0	X	将发送停止条件; STO 标志将复位。
		无 DAT 操作	1	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将复位。
0x54	已发送 DAT 中的数据字节; 已接收非 ACK。	装入 数据字节	0	1	0	0	X	将发送数据字节, 接收 ACK 位。
		无 DAT 操作	1	1	0	0	X	将发送重复的起始条件。
		无 DAT 操作	0	1	1	0	X	将发送停止条件; STO 标志将复位。
		无 DAT 操作	1	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将复位。

Table 12-15: 主机接收模式

状态代码	I2C 总线及硬件状态	应用软件响应						I2C 硬件工作的下一个步骤
		写/读 DAT	写 CON					
			STA	TXRX	STO	SI	AA	
0x01	已发送起始条件。	装入 SLA+R	X	1	0	0	X	将发送 SLA+R ； 接收 ACK 位。
0x21	已发送重复起始条件。	装入 SLA+R	X	1	0	0	X	同上。
		装入 SLA+R	X	1	0	0	X	将发送 SLA+W； I2C 将切换为主机发送模式。

		无 DAT 操作	1	1	0	0	X	当总线空闲时发送起始条件。
0x0b	已发送 SLA+R; 已接收 ACK。	无 DAT 操作	0	0	0	0	0	将接收数据字节, 返回非 ACK 位。
		无 DAT 操作	0	0	0	0	1	将接收数据字节, 返回 ACK 位。
0x4b	已发送 SLA+R; 已接收非 ACK。	无 DAT 操作	1	1	0	0	X	将发送重复的起始条件。
		无 DAT 操作	0	1	1	0	X	将发送停止条件; STO 标志将复位。
		无 DAT 操作	1	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将复位。
0x1d		读数据字节	0	0	0	0	0	将接收数据字节, 返回非 ACK 位。
		读数据字节	0	0	0	0	1	将接收数据字节, 返回 ACK 位。
0x5d		读数据字节	1	1	0	0	X	将发送重复的起始条件。
		读数据字节	0	X	1	0	X	将发送停止条件; STO 标志将复位。
		读数据字节	1	X	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将复位。

Table 12-16: 从机接收模式

状态代码	I2C 总线及硬件状态	应用软件响应						I2C 硬件工作的下一个步骤	
		写/读 DAT	写 CON						
			STA	TXRX	STO	SI	AA		
0x0a	已接收自身的 SLA+W， 已返回 ACK。	无 DAT 操作	X	0	0	0	0	将接收数据字节，返回非 ACK 位。	
		无 DAT 操作	X	0	0	0	1	将接收数据字节，返回 ACK 位。	
0x13	前一次寻址使用自身从 属地址；已接收数据字 节；已返回 ACK。	读数据字节	X	0	0	0	0	将接收数据字节，返回非 ACK	
		读数据字节	X	0	0	0	0	1	将接收数据字节，返回 ACK
0x43	前一次寻址使用自身从 属地址；已接收数据字 节；已返回非 ACK。	读数据字节	0	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA。	
		读数据字节	0	0	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；
		读数据字节	1	0	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA。当 总线空闲时发送起始条件。
		读数据字节	1	0	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；当总 线空闲后发送起始条件。
0x1d	当使用从接收或从发送 模式静态寻址时，接收到 停止条件或重复的起始 条件。	无 DAT 操作	0	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA。	
		无 DAT 操作	1	0	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA。当 总线空闲时发送起始条件。

Table 12-17: 从机发送模式

状态代码	I2C 总线及硬件状态	应用软件响应						I2C 硬件工作的下一个步骤
		写/读 DAT	写 CON					
			STA	TXRX	STO	SI	AA	
0x4a	已接收自身的 SLA+W， 已返回 ACK。	装入数据字节	X	0	0	0	X	将发送数据字节，返回 ACK 位。

0x5C	已发送 DAT 中的数据字节；已接收 ACK。	装入数据字节	X	1	0	0	0	将发送最后一个数据字节，返回 ACK 位。
		装入数据字节	X	1	0	0	1	将发送数据字节，返回 ACK 位。
0x4C	已发送 DAT 中的数据字节；已接收非 ACK。	无 DAT 操作	0	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA。
		无 DAT 操作	0	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；
		无 DAT 操作	1	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA。当总线空闲时发送起始条件。
		无 DAT 操作	1	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；当总线空闲时发送起始条件。
0x1C	已发送 DAT 中的数据字节 (AA = 0)；已接收 ACK。	无 DAT 操作	0	X	0	0	X	切换到不可寻址 SLV 模式；不识别自身 SLA。
		无 DAT 操作	1	X	0	0	X	切换到不可寻址 SLV 模式；不识别自身 SLA。当总线空闲时发送起始条件。

12.4.7 I2C 状态服务程序

此节给出各种情况下 I2C 需要完成的服务程序例子。包括：

- 系统复位后的 I2C 初始化
- I2C 中断服务程序
- 支持 I2C 在四种工作模式下的 13 种状态服务程序

12.4.7.1 初始化程序

初始化 I2C 成主机或从机例子。

1. 将自身的从属地址装入 ADR。
2. 使能 I2C 中断。
3. 向寄存器 CONSET 写入 0x42 来置位 EN 和 MASL 位，并使能主机功能。对于从机功能，可向寄存器 CONSET 写入 0x40。

12.4.7.2 启动主机发送功能

通过建立缓冲区、指针和数据计数、然后启动起始条件来执行主机发送操作。

1. 初始化主机数据计数器。
2. 建立数据将被发送到的从属地址，并且添加写位。
3. 向 CONSET 写入 0x20 来置位 STA 位。
4. 在主发送缓冲区内建立要发送的数据。
5. 退出。

12.4.7.3 启动主机接收功能

通过建立缓冲区、指针和数据计数、然后启动起始条件来执行主机接收操作。

1. 初始化主机数据计数器。
2. 建立数据将被发送到的从属地址，并且添加写位。

3. 向CONSET 写入0x20 来置位STA 位。
4. 在主发送缓冲区内建立要发送的数据。
5. 建立主接收缓冲区。
6. 退出。

12.4.7.4 I2C 中断服务程序

确定 I2C 的状态和处理该状态的状态程序。

- 从STA 中读出I2C 的状态。
- 使用状态值跳转到13个可能状态服务程序中的一个。

12.4.7.5 无指定模式下的状态服务程序

状态代码: 0x01

已发送起始条件。将发送 SLA+R/W 和接收 ACK。

1. 向DAT 写入从机地址和R/W 位。
2. 向CONSET 写入0x04 来置位AA 位。
3. 向CONCLR 写入0x08 来清除SI 标志。
4. 建立主发送模式数据缓冲区。
5. 建立主接收模式数据缓冲区。
6. 初始化主机数据计数器。
7. 退出。

状态代码: 0x22

已发送重复起始条件。即将发送从机地址+ R/W 位和接收 ACK 位。

1. 向DAT 写入从机地址和R/W 位。
2. 向CONSET 写入0x04 来置位AA 位和TXRX 位。
3. 向CONCLR 写入0x08 来清除SI 标志。
4. 建立主发送模式数据缓冲区。
5. 建立主接收模式数据缓冲区。
6. 初始化主机数据计数器。
7. 退出。

12.4.7.6 主机发送状态

状态代码: 0x0b

之前状态为 0x01，已发送从机地址和写操作位，并接收了 ACK。即将发送第一个数据字节和接收 ACK 位。

1. 将主发送缓冲区的第一个数据字节装入DAT。

2. 向CONCLR 写入0x08 来清除SI 标志。
3. 主发送缓冲区指针加1。
4. 退出。

状态代码: 0x4b

已发送从机地址和写操作位，并接收了非 ACK。即将发送停止条件。

1. 向CONSET 写入0x14 来置位STO 和AA 位。
2. 向CONCLR 写入0x08 来清除SI 标志。
3. 退出。

状态代码: 0x14

已发送数据并接收了 ACK。如果发送的数据是最后一个数据字节则发送一个停止条件，否则发送下一个数据字节。

1. 主机数据计数器减1，如果发送的不是最后一个数据字节就跳至第5 步。
2. 如果要接收数据，跳至第11步。
3. 向CONSET 写入0x14 来置位STO 和AA 位。
4. 向CONCLR 写入0x08 来清除SI 标志。
5. 退出。
6. 从主发数据缓冲区中装入下一个数据字节到DAT
7. 向CONSET 写入0x04 来置位AA 位。
8. 向CONCLR 写入0x08 来清除SI 标志。
9. 主发送缓冲区指针加1。
10. 退出。
11. CONSET 的STA 和 MASL 位置位
12. 发送从机地址+R
13. 向CONCLR 写入0x08 来清除SI 标志。
14. 退出。

状态代码: 0x54

已发送数据，并接收了非 ACK。即将发送停止条件。

1. 向CONSET 写入0x14 来置位STO 和AA 位。
2. 向CONCLR 写入0x08 来清除SI 标志。
3. 退出。

12.4.7.7 主机接收状态**状态代码: 0x0b**

之前状态为 0x01，已发送从机地址和写操作位，并接收了 ACK。即将接收数据字节和返回 ACK 位。

1. 向 CONCLR 写入 0x09 来清除 SI 和 TXRX 标志。
2. 退出。

状态代码: 0x4b

已发送从机地址和读操作位，并接收了非 ACK。即将发送停止条件。

1. 向 CONSET 写入 0x14 来置位 STO 和 AA 位。
2. 向 CONCLR 写入 0x08 来清除 SI 标志。
3. 退出。

状态代码: 0x1d

数据已接收，ACK 已发送。即将从 DAT 读取数据，下一个数据也即将接收。如果是最后一个数据，发送非 ACK，否则发送 ACK。

1. 从 DAT 读取数据，存入主机接收缓冲区。
2. 主机缓冲区指针加1。如果是最后一个数据，跳转到5。
3. 向 CONCLR 写入 0x0C 来清除 SI 和 AA 标志。
4. 向 CONSET 写入 0x10 来置位 STO。
4. 退出。
5. 向 CONCLR 写入 0x08 来清除 SI 标志。
6. 主发送缓冲区指针加1。
7. 退出。

状态代码: 0x5d

数据已接收，非 ACK 已发送。即将从 DAT 读取数据，并发送停止条件。

1. 从 DAT 读取数据，存入主机接收缓冲区。
2. 向 CONSET 写入 0x14 来置位 STO 和 AA 位。
3. 向 CONCLR 写入 0x08 来清除 SI 标志。
4. 退出。

12.4.7.8 从机接收状态

状态代码: 0x0a

已接收自身的 SLA+W，已返回 ACK。即将接收数据，并返回 ACK。

1. 向 CONCLR 写入 0x08 来清除 SI 标志。
2. 初始化从机数据接收缓冲区。
3. 初始化从机数据接收缓冲区指针。
4. 退出。

状态代码: 0x13

前一次寻址使用自身从机地址；已接收数据字节；已返回 ACK。即将接收更多数据。

1. 从DAT读取数据字节存入从机数据接收缓冲区。
2. 缓冲区指针加1。如果是最后一个数据，跳转到5。
3. 向CONCLR 写入0x0C 来清除SI 和AA标志。
4. 退出。.
5. 向CONCLR 写入0x08 来清除SI 标志。
6. 缓冲区指针加1。
7. 退出。

状态代码: 0x43

前一次寻址使用自身从机地址；已接收数据字节；已返回非 ACK。

1. 向CONCLR 写入0x08 来清除SI 标志。
2. 退出。

状态代码: 0x1d

已接收停止条件或重复起始条件，但仍作为从机寻址。不会保存数据。

1. 向CONCLR 写入0x08 来清除SI 标志。
2. 退出。

12.4.7.9 从机发送状态**状态代码: 0x4a**

已接收到自身从机地址和读操作位，已返回 ACK。将发送数据和接收 ACK 位。

1. 将从机发送缓冲区的第一个数据字节装入DAT。
2. 向CONCLR 写入0x09 来清除SI 和TXRX标志。
3. 初始化从机发送缓冲区。
4. 从机发送缓冲区指针加1。
5. 退出。

状态代码: 0x5c

已发送数据并接收了 ACK。将发送数据和接收 ACK 位。

1. 将从机发送缓冲区的第一个数据字节装入DAT。.
2. 向CONCLR 写入0x08 来清除SI 标志。
3. 从机发送缓冲区指针加1。
4. 退出。

状态代码: 0x1c

已发送数据并接收了非 ACK。

1. 向CONCLR 写入0x08 来清除SI 标志。
2. 退出。.

13 RTC

13.1 概述

通过对 1 Hz/1KHz 时钟计数来作参照时间，RTC 定时器可用于完成实时时钟功能。此 RTC 主要功能如下：

- 专用超低功耗的32 kHz晶振
- 使用1 Hz 时钟作为秒和1KHz 时钟作为毫秒时间间隔
- 32位RTC计数器
- 报警和系统唤醒功能

13.2 管脚描述

Table 13-1: RTC 管脚描述

管脚	类型	描述
RTCXIN	输入	32 kHz 晶振输入
RTCXOUT	输出	32 kHz 晶振输出

13.3 RTC 寄存器 描述

Table 13-2: RTC 寄存器一览 (基地址 0x4005 0000)

名称	读写	偏移地址	描述	初始值
DR	R	0x000	数据寄存器	0x00
MR	R/W	0x004	匹配寄存器	0x00
LR	R/W	0x008	装载寄存器	0x00
CR	R/W	0x00C	控制寄存器	0x00
ICSC	R/W	0x010	中断控制寄存器	0x00
RIS	R	0x014	原始中断状态寄存器	0x00
MIS	R	0x018	中断状态寄存器	0x00
ICR	W	0x01C	中断清除寄存器	0x00

13.3.1 RTC 数据寄存器

Table 13-3: RTC 数据寄存器 (DR - 地址 0x4005 0000) 描述

位	名称	描述	初始值
31:0	DATA	返回当前计数值.	0x00

13.3.2 RTC 匹配寄存器

Table 13-4: RTC 匹配寄存器 (MR - 地址 0x4005 0004) 描述

位	名称	描述	初始值
31:0	MATCH	RTC 匹配寄存器值。	0x00

13.3.3 RTC 装载寄存器

Table 13-5: RTC 装载寄存器 (LR - 地址 0x4005 0008) 描述

位	名称	描述	初始值
31:0	装入	RTC 装载寄存器值。	0x00

13.3.4 RTC 控制寄存器

此寄存器为读/写(R/W) 寄存器。读取操作会返回 RTC 的状态。写入操作会启用或禁用 RTC。启用 RTC 后，对此寄存器位 0 的任何写入操作只有在系统复位后才会生效。

Table 13-6: RTC 控制寄存器 (CR - 地址 0x4005 000C) 描述

位	名称	值	描述	初始值
0	RTCSTART		RTC 使能。通过此位启用 RTC 后，对此位的任何写入操作只有在上电复位 (POR) 后才会生效。	0x0
		0	RTC 禁止。	
		1	RTC 使能	
31:1	-	-	保留。	-

13.3.5 RTC 中断控制寄存器

此寄存器为 R/W 寄存器，可控制 RTC 产生的中断屏蔽。写入操作可设置或清除屏蔽。读取此寄存器可返回 RTC 中断上屏蔽的当前值。

Table 13-7: RTC 中断 Mask 寄存器 (ICSC - 地址 0x4005 0010) 描述

位	名称	值	描述	初始值
0	RTCIC		中断控制寄存器。读取操作将返回 RTC 控制寄存器的当前值。	0x0
		0	写入 0 将屏蔽中断	
		1	写入 1 将启用中断	
31:1	-	-	保留。	0x0

13.3.6 RTC 原始中断状态寄存器

此寄存器为只读(RO) 寄存器。读取此寄存器将提供屏蔽前相应中断的当前原始状态值。写入操作无效。

Table 13-8: RTC 中断状态寄存器 (RIS - 地址 0x4005 0014) 描述

位	名称	描述	初始值
0	RTC RIS	原始中断事件标志寄存器。读取操作将返回原始中断事件标志的状态。	0x0

31:1	-	保留.	0x0
------	---	-----	-----

13.3.7 RTC 中断状态寄存器

只读寄存器。读取此寄存器将提供相应中断的当前屏蔽状态值。写入操作无效。

Table 13-9: RTC 中断状态寄存器 (MIS - 地址 0x4005 0018) 描述

位	名称	描述	初始值
0	RTCMIS	屏蔽中断寄存器状态。读取操作将返回由 ICR 寄存器控制的屏蔽中断状态。	0x0
31:1		保留. Read as zero.	0x0

13.3.8 RTC 中断清除寄存器

此寄存器为只写寄存器。写入 1 会清除相应的中断。写入 0 无效。

Table 12-13-10: RTC 中断清除寄存器 (ICR - 地址 0x4005 001C) 描述

位	名称	描述	初始值
0	RTICR	原始中断事件标志清除寄存器。写入 1 将清除中断事件标志。写入 0 无效。	0x0
31:1		保留.	0x0

13.4 功能描述

使用RTC从深度睡眠（Deep-sleep）和掉电（Power-down）模式下唤醒系统

RTC 可以配置来在深度睡眠和掉电模式下产生 RTC 中信号从而唤醒系统。在深度睡眠和掉电模式下使用 RTC 必须选择 RTC 晶振作为时钟源。

注: 为了在从深度掉电模式中唤醒后获取有效的 RTC 值，应先对 RTC 执行“虚拟”读取。下一次读取将包含更新的 RTC 值。.

14 ADC

14.1 概述

G32F0 提供 12 位的 ADC 转换器，主要功能如下：

- 1M Hz 转换率，12 位的 A/D 转换器
- 支持 8 个外部 AD 通道采样转换
- 模块支持低功耗掉电
- ADC 测量范围 $0 \sim V_{DDA}$
- 支持突发模式 ADC 转换
- 可配置 ADC 转换触发源-输入管脚电平转换或定时器匹配信号。
- 每个 ADC 转换器有 8 个寄存器存储转换结果，从而减少中断负担 d.

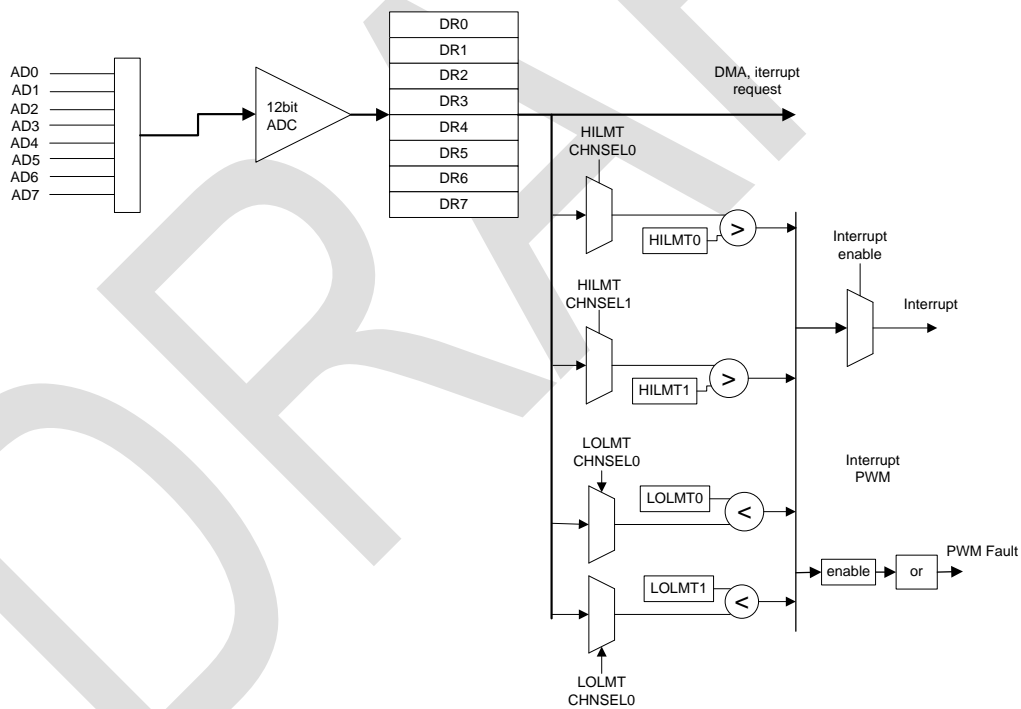


Figure 14-1: ADC 功能模块框图

14.2 管脚描述

Table 14-1: ADC 管脚描述

管脚	类型	描述
AD0~AD7	输入	模拟输入。ADC 转换器元件可测量这些输入信号的电压。输入信号不得超过 V_{DDA} 。

14.3 寄存器描述

Table 14-2: ADC 寄存器一览 (ADC0: 基地址 0x4002 0000;)

名称	读写	偏移地址	描述	初始值
CR	R/W	0x000	ADC 控制寄存器。CR 寄存器必须在转换之前进行配置。	0x0000 0000
GDR	R/W	0x004	ADC 全局数据寄存器。包含 ADC 转换器最近的转换结果。	NA
CHSEL	R/W	0x008	通道选择控制寄存器	0x0000 0000
INTEN	R/W	0x00C	ADC 中断使能寄存器。该寄存器控制 ADC 转换结束标志 DONE 是否触发 ADC 中断。	0x0000 0100
DR0	R/W	0x010	A/D 转换结果寄存器 0。该寄存器对应 CHSEL 的 SEL0 定义的通道最近转换值。	NA
DR1	R/W	0x014	A/D 转换结果寄存器 1。该寄存器对应 CHSEL 的 SEL1 定义的通道最近转换值。	NA
DR2	R/W	0x018	A/D 转换结果寄存器 2。该寄存器对应 CHSEL 的 SEL2 定义的通道最近转换值。	NA
DR3	R/W	0x01C	A/D 转换结果寄存器 3。该寄存器对应 CHSEL 的 SEL3 定义的通道最近转换值。	NA
DR4	R/W	0x020	A/D 转换结果寄存器 4。该寄存器对应 CHSEL 的 SEL4 定义的通道最近转换值。	NA
DR5	R/W	0x024	A/D 转换结果寄存器 5。该寄存器对应 CHSEL 的 SEL5 定义的通道最近转换值。	NA
DR6	R/W	0x028	A/D 转换结果寄存器 6。该寄存器对应 CHSEL 的 SEL6 定义的通道最近转换值。	NA
DR7	R/W	0x02C	A/D 转换结果寄存器 7。该寄存器对应 CHSEL 的 SEL7 定义的通道最近转换值。 注意: ADC1 DR7 保留为芯片内部温度传感器转换。	NA
INTSTAT	RO	0x030	ADC 状态寄存器。该寄存器包含 DONE 和 OVERRUN 标志以及中断标志。	0
HILMT	R/W	0x034	ADC 上限控制寄存器。寄存器控制上限检测功能。	0
LOLMT	R/W	0x038	ADC 下限控制寄存器。寄存器控制下限检测功能。	0
-	-	0x03C	保留	0
SSCR	R/W	0x040	软件触发转换控制	0

14.3.1 ADC 控制寄存器

ADC 控制寄存器用于对 A/D 通道选择, 采样率, 转换模式以及转换条件的设置。

Table 14-3: A/D 控制寄存器 (CR) 描述

位	名称	值	描述	初始值
7:0	CNVEN		ADC 通道使能	0x0

15:8	CLKDIV		A/D 转换时钟分频值。ADC 转换时钟由 ADC 外设时钟(PCLK)除以该分频取得。ADC 转换时钟最大不超过 16MHz。每一次 ADC 转换需要 16 个时钟，ADC 最大采样率为 1MHz。	0x0
16	BURST		Burst mode control.	0
		0	触发模式。ADC 转换由 START 域定义的信号触发。每次触发，ADC 将对通道选择寄存器（CHSEL）选择的有效通道进行一次遍历，转换结果存入相应的 DR 寄存器。	
		1	突发模式。当该位置为 1 且 START 域设置为 0 时，该 ADC 转换器进入突发模式。在该模式下，ADC 会对通道选择寄存器选定的通道(DR0SEL~DR7SEL)循环进行转换，每次转换值都会存入相应 DR 寄存器。清除该位才能停止 ADC 转换。	
23:17	-	-	保留	0x0
27:24	START		ADC 转换启动控制。在触发模式下，该域控制 ADC 什么时候进行 ADC 转换。	0x0
		0x0	停止触发采样	
		0x1	采用软件触发	
		0x2	采用 TIM2_CAP0 输入信号触发。触发边沿由位 28 决定。	
		0x3	采用 TIM2_CAP1 输入信号触发。触发边沿由位 28 决定。	
		0x4	采用 TIM2_MAT0 输入信号触发。触发边沿由位 28 决定。	
		0x5	采用 TIM2_MAT1 输入信号触发。触发边沿由位 28 决定。	
		0x6	采用 TIM3_MAT0 输入信号触发。触发边沿由位 28 决定。	
		0x7	采用 TIM3_MAT1 输入信号触发。触发边沿由位 28 决定。	
		0x8	采用 PWM 重载事件信号触发。	
28	EDGE		边沿控制。该边沿控制只对 START 域 2~8 设置有效。	0
		0	上升沿有效	
		1	下降沿有效	
29	SCMODE		ADC 转换采样时钟选择，必须为 1	0
		0	ADC 内部时钟作为采样时钟	
		1	外部时钟作为采样时钟，当为触发模式时，必须选为外部时钟模式	
31:30	-	-	保留	0x0

14.3.2 ADC 全局数据寄存器

该寄存器包含当前 ADC 转换器最近的转换结果。它包括通道选择，转换结果及相关标志。

Table 14-4: A/D 全局数据寄存器 (GDR) 描述

位	名称	描述	初始值
11:0	RESULT	当 DONE 为 1 时，该域包含了 CHN 指定的通道选择号所选择的通道的 A/D 转换值。 0 表示所选择的通道的电压小于等于或非常接近 V_{SS} 。 0xFF 表示所选择的通道的电压大于等于或非常接近 V_{DDA} 。	X
14:12	CHN	AD 通道号	000
15	OVERRUN	此位为 1 表示在突发模式下，有一到多个 A/D 转换结果在软件读出前被覆盖。	0
16	DONE	此位为 1 表示一次 A/D 转换结束。该标志位在数据读出后自动清零。此位是全局的 done。有任何通道采样完成，都会出此标志位	0
31:17	-	保留	NA

14.3.3 ADC 通道选择寄存器

该寄存器用于选择在单次触发或突发模式下需要进行 AD 转换的通道。

Table 14-5: A/D 通道选择寄存器 (CHSEL) 描述

位	名称	值	描述	初始值
0:2	DR0CHSEL		选择的 AD 通道号。该通道的转换值将存放于 DR0 寄存器	0
3	-		保留	-
6:4	DR1CHSEL		选择的 AD 通道号。该通道的转换值将存放于 DR1 寄存器	0
7	-		保留	-
10:8	DR2CHSEL		选择的 AD 通道号。该通道的转换值将存放于 DR2 寄存器	0
11	-		保留	-
14:12	DR3CHSEL		选择的 AD 通道号。该通道的转换值将存放于 DR3 寄存器	0
15	-		保留	-
18:16	DR4CHSEL		选择的 AD 通道号。该通道的转换值将存放于 DR4 寄存器	0
19	-		保留	-
22:20	DR5CHSEL		选择的 AD 通道号。该通道的转换值将存放于 DR5 寄存器	0
23	-		保留	-
26:24	DR6CHSEL		选择的 AD 通道号。该通道的转换值将存放于 DR6 寄存器	0
27	-		保留	-
30:28	DR7CHSEL		选择的 AD 通道号。该通道的转换值将存放于 DR7 寄存器	0
31	-		保留	-

14.3.4 ADC 中断使能寄存器

寄存器控制是否允许每次 A/D 转换结束时产生中断请求。例如，可能需要通过不断在某些 A/D 上执行转换来监视传感器。最近的转换结果可根据需要随时由应用程序读出。在这种情况下，某些 A/D 转换结束时都不使用中断方式。

Table 14-6: A/D 中断使能寄存器 (INTEN) 描述

位	名称	描述	初始值
7:0	INTEN	0 到 7 位对应通道选择寄存器 SEL0 到 SEL7 选择的通道 A/D 转换后是否产生中断。 清 0 则禁止该位对应的通道产生中断，置 1 则允许 A/D 转换结束后产生中断。	0x00
8	GINTEN	为 1 时，使能 ADC 中的全局 DONE 标志产生中断。为 0 时，只有个别由 INTEN 7:0 使能的 A/D 通道产生中断。	1
31:9	-	保留。	NA

14.3.5 ADC 数据寄存器组

ADC 数据寄存器用于存放 A/D 转换结果。它包括转换结束和溢出标志。

Table 14-7: A/D 数据寄存器组(DR0 ~ DR7) 描述

位	名称	描述	初始值
---	----	----	-----

11:0	RESULT	当 DONE 为 1 时, 该字段包含的是一个二进制数, 表示的是 SELn 字段所选定的 ADn 管脚的电压除以 V_{REF_ADC} 管脚上的电压: V/V_{REF_ADC} 。该字段为 0 表示 ADn 管脚处的电压小于、等于或接近于 V_{SS} , 而 0xFFFF 表明 ADn 管脚处的电压接近于、等于或大于 V_{REF_ADC} 。	X
29:12	-	保留。	0x0
30	OVERRUN	在突发模式下, 如果在产生 RESULT 位结果转换之前一个或多个转换结果丢失或被覆盖, 则该位置 1。读取该寄存器可清除此位。	0
31	DONE	A/D 转换结束时该位置 1。读取该寄存器时会清除此位。此位为全局的 done。	0

14.3.6 ADC 状态寄存器

A/D 状态寄存器允许同时检查所有 A/D 转换的状态。每个 A/D 转换的 DRn 寄存器中的 DONE 和 OVERRUN 标志都反映在 STAT 中。在 STAT 中还可以找到中断标志（所有 DONE 标志的逻辑“或”）和 A/D 值超出设定范围状态。

Table 14-8: A/D 状态寄存器 (STAT) 描述

位	名称	描述	初始值
7:0	DONE	0 到 7 位对应通道选择寄存器 DR0SEL 到 DR7SEL 选择的通道 A/D 转换后 DONE 的状态标志。	0
15:8	OVERRUN	0 到 7 位对应通道选择寄存器 DR0SEL 到 DR7SEL 选择的通道 A/D 转换后 OVERRRUN 的状态标志。	0
16	ADINT	该位为 A/D 中断标志。当任何一次 A/D 转换的 Done 标志置 1 且使能 A/D 产生中断（通过 INTEN 寄存器设置）时, 该位置 1。	0
17	HILMTFLAG0	上限 0 溢出标志。当选择的转换值大于上限 0 时, 该位为 1。写 1 清除该标志位。	0
18	HILMTFLAG1	上限 1 溢出标志。当选择的转换值大于上限 1 时, 该位为 1。写 1 清除该标志位。	0
19	LOLMTFLAG0	下限 0 溢出标志。当选择的转换值小于上限 0 时, 该位为 1。写 1 清除该标志位。	0
20	LOLMTFLAG1	下限 1 溢出标志。当选择的转换值小于上限 1 时, 该位为 1。写 1 清除该标志位。	0
21	ADCRDY	该位值为 1 时标志 ADC 转换器可以使用。	0
31:22	-	保留。	0x0

14.3.7 上限控制寄存器

上限控制寄存器用于设定与 A/D 转换值比较的最大值。当选择的 A/D 转换值大于上限值时, 可生成一个标志并产生中断。每个 ADC 转换器有两个上限比较器。

Table 14-9: 上限控制寄存器(HILMT) 描述

位	名称	值	描述	初始值
11:0	HILMT0	0~0xFFFF	上限 0 值。	0
14:12	CHNSELO	0~7	上限 0 通道选择号。0~7 对应通道选择器 SEL0~SEL7。该域选择的转换值与 HILMT0 进行比较。	0
15	INTEN0		上限 0 溢出中断使能。当选择的转换值大于设定上限时使能中断。	0
		0	中断禁止	
		1	中断使能	
27:16	HILMT1	0~0xFFFF	上限 1 值。	0

30:28	CHNSEL1	0~7	上限 1 通道选择号。0~7 对应通道选择器 SEL0~SEL7。该域选择的转换值与 HILMT1 进行比较。	0
31	INTEN1		上限 1 溢出中断使能。当选择的转换值大于设定上限时使能中断。	0
		0	中断禁止	
		1	中断使能	

14.3.8 下限控制寄存器

下限控制寄存器用于设定与 A/D 转换值比较的最小值。当选择的 A/D 转换值小于下限时，可生成一个标志并产生中断。每个 ADC 转换器有两个下限比较器。

Table 14-10: 下限控制寄存器(LOLMT) 描述

位	名称	值	描述	初始值
11:0	LOLMT0	0~0xFFFF	下限 0 值。	0
14:12	CHNSEL0	0~7	下限 0 通道选择号。0~7 对应通道选择器 SEL0~SEL7。该域选择的转换值与 HILMT0 进行比较。	0
15	INTEN0		下限 0 溢出中断使能。当选择的转换值小于设定下限时使能中断。	0
		0	中断禁止	
		1	中断使能	
27:16	LOLMT1	0~0xFFFF	下限 1 值。	0
30:28	CHNSEL1	0~7	下限 1 通道选择号。0~7 对应通道选择器 SEL0~SEL7。该域选择的转换值与 HILMT1 进行比较。	0
31	INTEN1		下限 1 溢出中断使能。当选择的转换值小于设定下限时使能中断。	0
		0	中断禁止	
		1	中断使能	

14.3.9 软件触发寄存器

该寄存器用于软件去触发 ADC 转换

Table 14-11: 软件触发寄存器(SSCR) 描述

位	名称	值	描述	初始值
0	ADCTRIG	-	置 1 触发 ADC 转换。	0
31:2	-	-	保留。	0

14.4 操作

14.4.1 选择需要进行 ADC 转换的 A/D 通道

每个 ADC 可以同时选择 8 个需要 A/D 转换的通道进行转换并把 12 位的转换结果分别保存在 8 个寄存器中。用户需在通道选择控制寄存器 (CHSEL) 定义将进行 A/D 转换的采样通道。A/D 转换通道扫描按照从 DR0SEL 到 DR7SEL 次序。在 DR0SEL 到 DR7SEL 中可设置相同通道选择，其结果是相同通道重复采样转换并存入相应的结果寄存器中。

14.4.2 ADC 触发转换模式

如 ADC 的触发转换被触发，所有选中的 A/D 通道都会被转换一次。在完成对所有通道转换之前，不能被中断。如此时发出软件命令或有事件触发 ADC 转换，那么该次 ADC 转换请求会被忽略掉。

ADC 转换触发事件可以是软件请求，定时器捕捉(CAP)或匹配(MAT)信号，还可以是 PWM 事件。在定时器捕捉(CAP)或匹配(MAT)作触发信号时，信号边沿控制由 CR 寄存器的 EDGE 位决定。

14.4.3 中断

当任何 A/D 通道转换完成时，DONE 标志位会置 1，并触发中断。软件可通过控制中断使能位来决定在转换结束时是否进入中断。同时由于 DONE 标志位在 DRn 寄存器数据读取自动清零，用户需查询 ADC 寄存器来作正确处理。

15 PWM 脉宽调制

15.1 概述

G32F0 可提供独立的事件驱动的脉宽调制模块 PWM。PWM 模块可配置成 3 对互补输出，6 个独立输出或互补和独立混合的 PWM 信号输出（如 2 对互补输出，2 个独立输出）。同时支持 0~100% 占空比的边沿对齐和中心对齐模式。

PWM 模块使用 16 位计数器，它的精度在边沿对齐时是一个时钟周期，中心对齐时是 2 个时钟周期。时钟周期由 PWM 时钟源 PWM_PCLK(系统时钟)、预分频寄存器以及模数值决定。

当 PWM 信号配置成互补输出时，PWM 具有自动死区插入功能。每一个 PWM 输出可被 PWM 发生器、系统定时器、ADC 转换结果、GPIO 输入以及软件控制。非对称的 PWM 还可以允许在每半个周期改变 PWM 占空比而不需要软件参与。

1 路的故障信号输入，可以有效的对外部环境进行实时响应。

PWM 重载，输出翻转事件都可触发 A/D 转换，定时器工作，适用于各种复杂应用。

15.1.1 特点

- 独立的脉宽调制模块
- 工作在系统时钟（System Clock）
- 多达 6 路 PWM 信号输出
 - 单路独立输出模式
 - 互补输出模式
 - 混合输出模式
- 互补输出功能
 - 单独的死区上升沿/下降沿插入
 - 单独的高低脉冲宽度软件补偿
 - 中心对齐下的非对称 PWM 输出
 - 单独的高低脉冲输出极性控制
- 边沿对齐 PWM 输出
- 24 位调制精度
- 半周期重载机制
- 完善的 1 到 16 的重载频率控制
- 独立、可软件控制的 PWM 输出
- 1 路故障输入保护
- 输出极性控制
- PWM 控制寄存器写保护功能

•可配置的 PWM 互补输出信号源

- PWM 信号发生器
- 外部 GPIO 管脚
- 定时器
- ADC 转换值

15.1.2 Block Diagram

Figure 18-1.表示 PWM 功能模块框图:

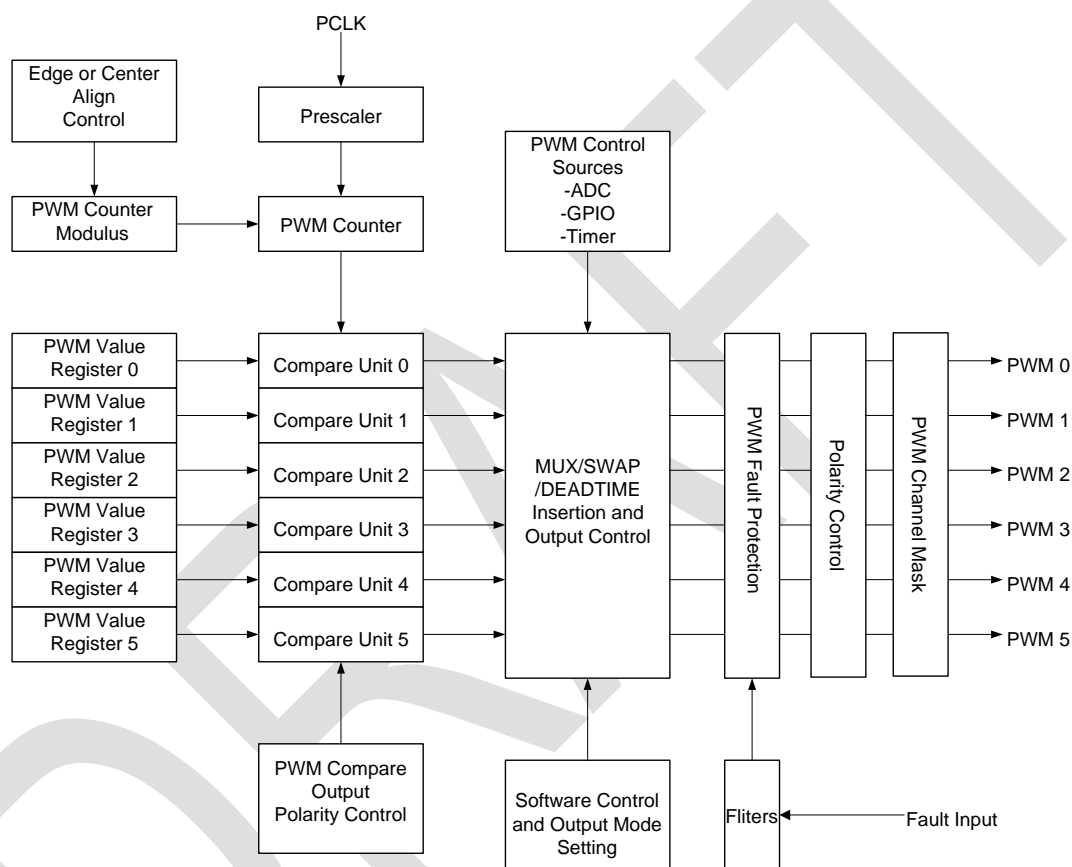


Figure 15-1: PWM 功能模块框图

15.2 管脚描述

PWM 脉宽调制模块涉及下列外部管脚。

Table 15-1: PWM 管脚描述

管脚	类型	G32F064x	G32F032x	G32F016x	G32F008x	描述
PWM_0	输出	Yes	Yes	Yes	No	PWM 输出通道 0
PWM_1	输出	Yes	Yes	Yes	Yes	PWM 输出通道 1
PWM_2	输出	Yes	Yes	Yes	Yes	PWM 输出通道 2
PWM_3	输出	Yes	Yes	Yes	No	PWM 输出通道 3

PWM_4	输出	Yes	Yes	Yes	Yes	PWM 输出通道 4
PWM_5	输出	Yes	Yes	Yes	Yes	PWM 输出通道 5
PWM_FAULT	输入		Yes	Yes	Yes	故障输入，可用于禁止相应 PWM 输出

15.3 寄存器描述

下列寄存器列表包涵寄存器基地址和偏移地址。基地址是基于系统内核的，偏移地址是每个模块内的相对地址。

Table 15-2: PWM 寄存器一览(基地址 0x4004 C000)

名称	读写	偏移地址	描述	初始值
CTRL	R/W	0x000	控制寄存器	0x0
FCTRL	R/W	0x004	故障输入控制寄存器	0x0
FLTACK	R/W	0x008	故障输入状态寄存器	0x0
OUT	R/W	0x00C	输出控制寄存器	0x0
CNTR	RO	0x010	计数器寄存器	0x0
CMOD	R/W	0x014	模数计数器寄存器	0x0
VAL0	R/W	0x018	数值寄存器 0	0x0
VAL1	R/W	0x01C	数值寄存器 1	0x0
VAL2	R/W	0x020	数值寄存器 2	0x0
VAL3	R/W	0x024	数值寄存器 3	0x0
VAL4	R/W	0x028	数值寄存器 4	0x0
VAL5	R/W	0x02C	数值寄存器 5	0x0
-	-	0x030	数值寄存器 6	
-	-	0x034	数值寄存器 7	
DTIM0	R/W	0x038	死区寄存器 0	0xFFFF
DTIM1	R/W	0x03C	死区寄存器 1	0xFFFF
DMAPO	R/W	0x040	映射失效控制寄存器 1	0x0
DMAPI	R/W	0x044	映射失效控制寄存器 2	0x0
CNFG	R/W	0x048	配置寄存器	0x0
CCTRL	R/W	0x04C	输出通道控制寄存器	0x0
PORT	R/W	0x050	端口控制寄存器	0x0
ICCTRL	R/W	0x054	Internal Correction Control Register.	0x0
PSCR	R/W	0x060	Polarity Invert control Register	0x0
CNTRINI	WO	0x064	Counter Init Register	0x0

15.3.1 PWM 控制寄存器

Table 15-3 PWM 控制(CTRL) 寄存器

位	名称	值	描述	初始值
0	PWMEN		使能 PWM 发生器。可读写。	0
		0	PWM 发生器输出被禁止。此时 PWM 输出处于非工作状态，除非寄存器 OUTCTLm=1。	
		1	PWM 发生器输出使能。	
1	LDOK		此位用于装载 CTRL 预分频值 (PRSC)，CMOD 模数寄存器值和所有数值寄存器 VALm 值到 PWM 运行模块。CTRL 预分频值，CMOD 模数寄存器值和所有数值寄存器 VALm 值会在下一个 PWM 重载时生效。置 LDOK 位遵循先读然后再写逻辑 1。当新的值重载入 PWM 模块后，LDOK 位会被自动清 0。也可在值重载之前，手动写 0 清除此位。	0
		0	无效操作	
		1	装载 CTRL 预分频值，CMOD 模数寄存器值和所有数值寄存器 VALm 值到 PWM 运行模块。	
2	PWMF		在每一个重载周期开始阶段，此标识位被置 1。清除 PWMF 位应先读然后写 0。如果另一次重载在清 0 前刚发生，则此次写 0 无效。 注： 清除 PWMF 信号会清除等待的 PWMF 中断请求。	0
		0	自从上次 PWMF 清除后没有新的重载发生	
		1	自从上次 PWMF 清除后有新的重载发生	
3	PWMRIE		此位使能 PWMF 标识产生中断请求	0
		0	PWMF 中断禁止	
		1	PWMF 中断使能	
4	-	-	保留	0
6:5	PRSC	-	PWM 时钟预分频率设置 注： 读取 PRSC 位只是返回缓冲器值，此值可能不是当前生效的值。PRSC 位值只有在 LDOK 置 1 后的下一个 PWM 周期才开始生效。	00
		00	PWM 工作时钟频率	
		01	PWM 工作时钟频率/2	
		10	PWM 工作时钟频率/4	
		11	PWM 工作时钟频率/8	
7	IPOLO		选择 PWMn_0 和 PWMn_1 在互补输出下 PWM 数值寄存器的值	0
		0	在 PWM 下一个周期，取 VAL0 寄存器的值	
		1	在 PWM 下一个周期，取 VAL1 寄存器的值	
8	IPOL1		选择 PWMn_2 和 PWMn_3 在互补输出下 PWM 数值寄存器的值	0
		0	在 PWM 下一个周期，取 VAL2 寄存器的值	
		1	在 PWM 下一个周期，取 VAL3 寄存器的值	
9	IPOL2		选择 PWMn_4 和 PWMn_5 在互补输出下 PWM 数值寄存器的值	0
		0	在 PWM 下一个周期，取 VAL4 寄存器的值	
		1	在 PWM 下一个周期，取 VAL5 寄存器的值	
10	IPOL3		选择 PWMn_6 和 PWMn_7 在互补输出下 PWM 数值寄存器的值	0
		0	在 PWM 下一个周期，取 VAL6 寄存器的值	

		1	在 PWM 下一个周期, 取 VAL7 寄存器的值	
11	HALF		中心对齐下模式半周期重载使能。此位在边沿对齐时无效。	0
		0	半周期重载禁止	
		1	半周期重载使能	
15:12	LDFQ	-	PWM 重载频率控制。清除 LDFQ 域导致在每个/半个 PWM 周期重载。参照 HALF 位描述。 注: 无论 LDOK 位状态如何, LDFQ 域值只会在当前装载的周期已结束后才生效。读取的 LDFQ 域值不一定表示当前的工作状态。	0000
		0000	每个 PWM 重载时机	
		0001	每 2 个 PWM 重载时机	
		0010	每 3 个 PWM 重载时机	
		0011	每 4 个 PWM 重载时机	
		0100	每 5 个 PWM 重载时机	
		0101	每 6 个 PWM 重载时机	
		0110	每 7 个 PWM 重载时机	
		0111	每 8 个 PWM 重载时机	
		1000	每 9 个 PWM 重载时机	
		1001	每 10 个 PWM 重载时机	
		1010	每 11 个 PWM 重载时机	
		1011	每 12 个 PWM 重载时机	
		1100	每 13 个 PWM 重载时机	
		1101	每 14 个 PWM 重载时机	
		1110	每 15 个 PWM 重载时机	
		1111	每 16 个 PWM 重载时机	
16	SOFTFAULT		此位可触发一个软件故障输入	0
		0	无效	
		1	产生一个故障输入	
17	INIDIR		选择中心对齐下计数器初始计数方向	0
		0	PWM 开始时计数器向上计数	
		1	PWM 开始时计数器向下计数	
18~19	-	-	保留	
21: 20	TSCNT		触发信号输出计数寄存器。寄存器的配置值决定了每次输出信号的间隔。	00
		00	触发一次输出信号	
		01	触发两次输出信号	
		10	触发三次输出信号	
		11	触发四次输出信号	
23:22	TSSEL		触发信号选择寄存器: 触发输出信号被送给 ADC 和 timer	00
		00	当 PWM counter=1 时触发输出	
		01	当 PWM counter 累加到 VALn 时触发输出	
		10	当 PWM counter = CMOD 时触发输出	
		11	当 PWM counter 累减到 VALn 时触发输出	

24	CH0OUTEN		通道 0 的数值寄存器的值等于计数寄存器时产生的信号输出使能	0
		0	禁止	
		1	使能	
25	CH1OUTEN		通道 1 的数值寄存器的值等于计数寄存器时产生的信号输出使能	0
		0	禁止	
		1	使能	
26	CH2OUTEN		通道 2 的数值寄存器的值等于计数寄存器时产生的信号输出使能	0
		0	禁止	
		1	使能	
27	CH3OUTEN		通道 3 的数值寄存器的值等于计数寄存器时产生的信号输出使能	0
		0	禁止	
		1	使能	
28	CH4OUTEN		通道 4 的数值寄存器的值等于计数寄存器时产生的信号输出使能	0
		0	禁止	
		1	使能	
29	CH5OUTEN		通道 5 的数值寄存器的值等于计数寄存器时产生的信号输出使能	0
		0	禁止	
		1	使能	
30:31			保留	

15.3.2 PWM 故障输入控制寄存器

Table 15-4 PWM 故障输入控制住(FCTRL) 寄存器

位	名称	值	描述	初始值
0	FMODE0		选择手动/自动清除 PWM_FAULT0 故障输入状态.	0
		0	手动清除	
		1	自动清除	
1	FIE0		PWM_FAULT0 故障输入中断使能控制	0
		0	中断禁止	
		1	中断使能	
2	FMODE1		选择手动/自动清除 PWM_FAULT1 故障输入状态.	0
		0	手动清除	
		1	自动清除	
3	FIE1		PWM_FAULT1 故障输入中断使能控制	0
		0	中断禁止	
		1	中断使能	
4	FMODE2		选择手动/自动清除 PWM_FAULT2 故障输入状态.	0
		0	手动清除	
		1	自动清除	

5	FIE2		PWM_FAULT2 故障输入中断使能控制	0
		0	中断禁止	
		1	中断使能	
31:6	-	-	保留	0

15.3.3 PWM 故障状态寄存器

Table 15-5 PWM 故障状态 (FLTACK) 寄存器

位	名称	值	描述	初始值
0	FTACK0	-	此位写 1 清除 FFLAG0。写 0 无效。读此位返回 0。系统复位 (RESET) 会清除 FTACK0 位。 PWM 故障保护在 PWM 没有工作时也会生效。如 PWM 没有工作, 但故障输入信号锁存在系统中, 这时需要清除此位防止当 PWM 启动时进入中断。	0
1	FTACK6	-	此位写 1 清除 FFLAG6。写 0 无效。读此位返回 0。系统复位 (RESET) 会清除 FTACK6 位。 PWM 故障保护在 PWM 没有工作时也会生效。如 PWM 没有工作, 但故障输入信号锁存在系统中, 这时需要清除此位防止当 PWM 启动时进入中断。	0
2	FTACK7	-	此位写 1 清除 FFLAG7。写 0 无效。读此位返回 0。系统复位 (RESET) 会清除 FTACK7 位。 PWM 故障保护在 PWM 没有工作时也会生效。如 PWM 没有工作, 但故障输入信号锁存在系统中, 这时需要清除此位防止当 PWM 启动时进入中断。	0
15:3	-		保留	0
16	FFLAG0		只读标识位。此标识位在检测到 PWM_FAULT0 故障输入上升沿后 2 个 PWM 周期内保持为 1。 FTACK0 位写 1 可清除 FFLAG0 标识。	0
		0	没有检测到 PWM_FAULT0 故障输入信号	
		1	检测到 PWM_FAULT0 故障输入信号	
17	FPIN0		只读标识位。经过滤波后的 PWM_FAULT0 管脚状态	0
		0	逻辑 0 在 PWM_FAULT0 管脚	
		1	逻辑 1 在 PWM_FAULT0 管脚	
18	FFLAG6		只读标识位。此标识位在软件故障(CTRL 寄存器的 SOFTFAULT)设置后 2 个 PWM 周期内保持为 1。FTACK5 位写 1 可清除 FFLAG5 标识。	0
		0	无软件故障设置	
		1	有软件故障设置	
19			保留	
20	FFLAG7		只读标识位, 当 ADC 限制电流故障发生后的两个 PWM 操作时钟周期被刷新, 在 FLTACK 寄存器的 FTACK7 为写 1 将 FFLAG7 状态清除。	0
		0	无 ADC 限制电流故障	
		1	有 ADC 限制电流故障	
21	FPIN7		只读标识位, 反应了 ADC 限制电流状态	0
		0	没有 ADC 限制电流故障	
		1	有 ADC 限制电流故障	
31: 22	-	-	保留	0

15.3.4 PWM 输出控制寄存器

此可读写寄存器用于软件控制 PWMn 的 0~5 通道输出。当软件控制 (OUTCTL m) 被使能, PWMn_m 采用 OUTm 值来输出。系统复位会清除 OUTCTL 寄存器。

Table 15-6 PWM 输出控制(OUT) 寄存器

位	名称	值	描述	初始值
0	OUT0		PWMn_0 输出	0
		0	逻辑 0	
		1	逻辑 1	
1	OUT1		PWMn_1 输出	0
		0	逻辑 0	
		1	逻辑 1。如处于互补输出模式, PWM n_1 变为 PWM n_0 互补输出。	
2	OUT2		PWMn_2 输出	0
		0	逻辑 0	
		1	逻辑 1	
3	OUT3		PWMn_3 输出	0
		0	输出逻辑 0	
		1	输出逻辑 1。如处于互补输出模式, PWM n_3 变为 PWM n_2 互补输出。	
4	OUT4		PWMn_4 输出	0
		0	逻辑 0	
		1	逻辑 1	
5	OUT5		PWMn_5 输出	0
		0	逻辑 0	
		1	逻辑 1。如处于互补输出模式, PWM n_5 变为 PWM n_4 互补输出。	
7: 6			保留	
8	OUTCTL 0		PWMn_0 的软件控制使能	0
		0	软件控制禁止	
		1	软件控制使能	
9	OUTCTL 1		PWMn_1 的软件控制使能	0
		0	软件控制禁止	
		1	软件控制使能	
10	OUTCTL 2		PWMn_2 的软件控制使能	0
		0	软件控制禁止	
		1	软件控制使能	
11	OUTCTL 3		PWMn_3 的软件控制使能	0
		0	软件控制禁止	
		1	软件控制使能	

12	OUTCTL 4		PWMn_4 的软件控制使能	0
		0	软件控制禁止	
		1	软件控制使能	
13	OUTCTL 5		PWMn_5 的软件控制使能	0
		0	软件控制禁止	
		1	软件控制使能	
31:14	-	-	保留	0

15.3.5 PWM 计数器寄存器

Table 15-7 PWM 计数器(CNTR) 寄存器

位	名称	值	描述	初始值
15:0	CNT	-	此寄存器表示 PWM 计数器当前值。此寄存器在 PWM 启动之前可写。当 PWMEN 置 1 后，寄存器变为只读。	0
31:16	-	-	保留	-

15.3.6 PWM 模数计数器寄存器

PWM 模数寄存器用于控制 PWM 载波周期。此值只有当 LDOK 位置 1 和下一个 PWM 重载开始时才生效。读 CMOD 寄存器返回缓存器值。但它不一定是 PWM 发生器当前使用的值。

Table 15-8 PWM 模数计数器(CMOD) 寄存器

位	名称	值	描述	初始值
15:0	CMOD	-	此 32 位寄存器值用于定义 PWM 输出载波周期。	0
31:16	-	-	保留	-

15.3.7 PWM 数值寄存器组

16 位寄存器组值用于定义每路 PWM 输出的脉冲占空比。如寄存器值小于等于 0，PWM 输出在整个 PWM 周期内输出为逻辑 0。如寄存器值大于等于 PWM 模数计数器模数值（CMOD），PWM 输出在整个 PWM 周期内输出为逻辑 1。

Table 15-9 PWM 数值 (VALm, m=0~5) 寄存器

位	名称	值	描述	初始值
15:0	VAL	-	PWM 数值寄存器 s	0
31:16	-	-	保留	-

15.3.8 PWM 死区寄存器

PWMDT0 用于在 PWM 偶数输出通道从 0 变为 1 时插入的死区时间。PWMDT1 则用于在 PWM 奇数输出通道从 1 变为 0 时插入的死区时间。

死区时间会受到 PWM 预分频 (PRSC) 影响。死区时间计算公式如下：

$DT = P \times DTIMm - 1$ 个 PWM 工作时钟；其中 P 预分频 (PRSC) 值，DTIMm 是死区寄存器设定值。

例如 PWM 预分频 (PRSC) 为 2，并死区寄存器 DTIMm 为 5，那么死区时间就等于：

$DT = 2 \times 5 - 1 = 9$ PWM 工作时钟

另一个特殊情况是：P=1，DT= DTIMm。

Table 15-10 PWM 死区 (DTIMm, m=0,1) 寄存器

位	名称	值	描述	初始值
15:0	DTIMm	-	带写保护控制的 16 位寄存器，定义 PWM 互补输出下的死区时间。	0xFFFF
31:16	-	-	保留	-

15.3.9 PWM 映射失效控制寄存器

此寄存器带写保护。它用于定义当故障输入发生时需禁止的 PWM 输出通道。系统复位时，DMAP1-2 寄存器所有位都置 1。当 CNFG 寄存器的 WP 位置 1 时，DMAP1-2 寄存器被写保护。

Table 15-11: 映射失效 (DMAPO) 寄存器 0

位	名称	值	描述	初始值
0	P0FLTMAP0		当故障输入 0 (PWM_FAULT0) 发生时禁止 PWM 通道 0 输出。	1
		0	无效	
		1	使能	
1	P0FLTMAP1		当故障输入 1 (PWM_FAULT1) 发生时禁止 PWM 通道 0 输出。	1
		0	无效	
		1	使能	
2	P0FLTMAP2		当故障输入 2 (PWM_FAULT2) 发生时禁止 PWM 通道 0 输出。	1
		0	无效	
		1	使能	
7: 3	-	-	保留	
15:8	P1FLTMAPn		PWM 通道 1 故障保护，参照 PWM 通道 0 (F0FLTMAPn)	0xFF
23:16	P2FLTMAPn		PWM 通道 2 故障保护，参照 PWM 通道 0 (F0FLTMAPn)	0xFF
31:24	P3FLTMAPn		PWM 通道 3 故障保护，参照 PWM 通道 0 (F0FLTMAPn)	0xFF

Table 15-12: 映射失效 (DMAP1) 寄存器 1

位	名称	值	描述	初始值
---	----	---	----	-----

7:0	P4FLTMAn		PWM 通道 4 故障保护, 参照 PWM 通道 0 (F0FLTMAn)	0xFF
15:8	P5FLTMAn		PWM 通道 5 故障保护, 参照 PWM 通道 0 (F0FLTMAn)	0xFF
31:16	-	-	保留	

15.3.10 PWM 配置寄存器

此寄存器带写保护。它用于定义 PWM 工作模式。当 CNFG 寄存器的 WP 位置 1 时, 该寄存器被写保护。

Table 15-13: PWM 配置(CNFG) 寄存器

位	名称	值	描述	初始值
0	WP		此位为 PWM 模块中的写保护寄存器使能位。当此位置位后, 所有的保护寄存器 (包括 CNFG 本身) 的写操作被禁止。一旦此位被置位, 只有系统复位才能清除。被该位保护的寄存器包括: DMAP1-2, DTIM, CNFG, 和 CCTRL 寄存器的 ENHA 位。 而 ENHA 位清零时, VLMODE, SWP67, SWP45, SWP23, 和 CCTRL 寄存器的 SWP01 也会被保护。 注: 写寄存器 CNFG 并置 WP 位为 1 将是到下一次系统复位前最后一次修改保护寄存器。	0
		0	允许写保护寄存器	
		1	保护寄存器修改被禁止	
1	NDEP01		此位为写保护位。用于定义 PWMn 通道 0-1 为互补输出模式或独立输出模式	0
		0	互补输出模式	
		1	独立输出模式	
2	NDEP23		此位为写保护位。用于定义 PWMn 通道 2-3 为互补输出模式或独立输出模式	0
		0	互补输出模式	
		1	独立输出模式	
3	NDEP45		此位为写保护位。用于定义 PWMn 通道 4-5 为互补输出模式或独立输出模式	0
		0	互补输出模式	
		1	独立输出模式	
4	-	-	保留	0
5	BOTNEG01		此位为写保护位。用于定义互补输出模式 PWMn 通道 1 输出极性控制。	0
		0	无	
		1	取反	
6	BOTNEG23		此位为写保护位。用于定义互补输出模式 PWMn 通道 3 输出极性控制。	0
		0	无	
		1	取反	
7	BOTNEG45		此位为写保护位。用于定义互补输出模式 PWMn 通道 5 输出极性控制。	0
		0	无	
		1	取反	
8	-	-	保留	0

9	TOPNEG01		此位为写保护位。用于定义互补输出模式 PWMn 通道 0 输出极性控制。	0
		0	无	
		1	取反	
10	TOPNEG23		此位为写保护位。用于定义互补输出模式 PWMn 通道 2 输出极性控制。	0
		0	无	
		1	取反	
11	TOPNEG45		此位为写保护位。用于定义互补输出模式 PWMn 通道 4 输出极性控制。	0
		0	无	
		1	取反	
12	-	-	保留	0
13	EDG		此位为写保护位。用于定义 PWMn 通道采用中心对齐或边沿对齐方式计数方式产生波形。	0
		0	中心对齐	
		1	边沿对齐	
31:14	-	-	保留	0

15.3.11 PWM 通道控制寄存器

此寄存器带写保护。用于 PWM 通道互换控制。同时，ENHA 位提供对 VLMODE, SWP45, SWP23 和 SWP01 寄存器位的写控制。ENHA 位自身写控制由 CNFG 寄存器的 WP 位决定。

Table 15-14: PWM 通道(CCTRL) 寄存器

位	名称	值	描述	初始值
0	SWP01		当 ENHA 为 0 时，该位被写保护。PWM 通道 0-1 互换使能。	0
		0	无效	
		1	PWM 通道 0-1 互换	
1	SWP23		当 ENHA 为 0 时，该位被写保护。PWM 通道 2-3 互换使能。	0
		0	无效	
		1	PWM 通道 2-3 互换	
2	SWP45		当 ENHA 为 0 时，该位被写保护。PWM 通道 4-5 互换使能。	0
		0	无效	
		1	PWM 通道 4-5 互换	
3	-	-	保留	0
5:4	VLMODE		此域决定 PWM 重载时如何装载数值寄存器。当 ENHA 为 0 时，该位被写保护。	00
		00	每个 VAL0~7 寄存器被独立装载到相应 PMW0~7 通道	
		01	VAL0 寄存器被装载到 PMW0~5 通道	
		10	VAL0 寄存器被装载到 PMW0~3 通道	
		11	VAL0 寄存器被装载到 PMW0~7 通道	
6	MSK0		屏蔽 PWM 通道 0 输出使能	0

		0	不屏蔽	
		1	通道 0 输出被屏蔽，输出低电平	
7	MSK1		屏蔽 PWM 通道 1 输出使能	0
		0	不屏蔽	
		1	通道 1 输出被屏蔽，输出低电平	
8	MSK2		屏蔽 PWM 通道 2 输出使能	0
		0	不屏蔽	
		1	通道 2 输出被屏蔽，输出低电平	
9	MSK3		屏蔽 PWM 通道 3 输出使能	0
		0	不屏蔽	
		1	通道 3 输出被屏蔽，输出低电平	
10	MSK4		屏蔽 PWM 通道 4 输出使能	0
		0	不屏蔽	
		1	通道 4 输出被屏蔽，输出低电平	
11	MSK5		屏蔽 PWM 通道 5 输出使能	0
		0	不屏蔽	
		1	通道 5 输出被屏蔽，输出低电平	
14: 12	-	-	保留	0
15	ENHA		硬件加速使能。该位用于对寄存器 VLMODE, SWP67, SWP45, SWP23, 和 SWP01 写保护控制。该位不可修改当 CNFG 寄存器的 WP 位置 1 时。.	0
		0	不允许写 VLMODE, SWP67, SWP45, SWP23 和 SWP01 寄存器位。	
		1	允许写 VLMODE, SWP67, SWP45, SWP23 和 SWP01 寄存器位。	
31:16	-	-	保留	0

15.3.12 PWM 故障端口寄存器

Table 15-15: PWM 故障端口 (FPORTCTRL) 寄存器

位	名称	值	描述	初始值
0	FAULTPORT		此域包涵 PWM 的故障输入状态。1 为有故障发生，0 则没有。该域为只读。	未知
7:1	-	-	保留	
8	FAULTVAL0		PWM 故障并且 FPSEL=1 时 PWM 通道 0 输出逻辑值。	0
		0	输出逻辑 0	
		1	输出逻辑 1	
9	FAULTVAL1		PWM 故障并且 FPSEL=1 时 PWM 通道 1 输出逻辑值。	0
		0	输出逻辑 0	
		1	输出逻辑 1	
10	FAULTVAL2		PWM 故障并且 FPSEL=1 时 PWM 通道 2 输出逻辑值。	0
		0	输出逻辑 0	
		1	输出逻辑 1	

11	FAULTVAL3		PWM 故障并且 FPSEL=1 时 PWM 通道 3 输出逻辑值。	0
		0	输出逻辑 0	
		1	输出逻辑 1	
12	FAULTVAL4		PWM 故障并且 FPSEL=1 时 PWM 通道 4 输出逻辑值。	0
		0	输出逻辑 0	
		1	输出逻辑 1	
13	FAULTVAL5		PWM 故障并且 FPSEL=1 时 PWM 通道 5 输出逻辑值。	0
		0	输出逻辑 0	
		1	输出逻辑 1	
15: 14	-	-	保留	0
16	FPSEL0		PWM 故障时，允许输出预设值在 FAULTVAL0 位值到 PWM 通道 0.	0
		0	PWM 通道变为输入管脚。	
		1	输出 FAULTVAL0 值到 PWM 通道	
17	FPSEL1		PWM 故障时，允许输出预设值在 FAULTVAL1 位值到 PWM 通道 1.	0
		0	PWM 通道变为输入管脚。	
		1	输出 FAULTVAL1 值到 PWM 通道	
18	FPSEL2		PWM 故障时，允许输出预设值在 FAULTVAL2 位值到 PWM 通道 2.	0
		0	PWM 通道变为输入管脚。	
		1	输出 FAULTVAL2 值到 PWM 通道	
19	FPSEL3		PWM 故障时，允许输出预设值在 FAULTVAL3 位值到 PWM 通道 3.	0
		0	PWM 通道变为输入管脚。	
		1	输出 FAULTVAL3 值到 PWM 通道	
20	FPSEL4		PWM 故障时，允许输出预设值在 FAULTVAL4 位值到 PWM 通道 4.	0
		0	PWM 通道变为输入管脚。	
		1	输出 FAULTVAL4 值到 PWM 通道	
21	FPSEL5		PWM 故障时，允许输出预设值在 FAULTVAL5 位值到 PWM 通道 5.	0
		0	PWM 通道变为输入管脚。	
		1	输出 FAULTVAL5 值到 PWM 通道	
31:22	-	-	保留.	NA

15.3.13 PWM 死区效应补偿控制寄存器

该寄存器只适用于互补输出且中心对齐模式，它决定使用哪一个 IPOL_m 位控制的 Val_m 寄存器数值用来产生 PWM 波形。

Table 15-16: PWM 死区效应补偿控制(ICCTRL) 寄存器

位	名称	值	描述	初始值
0	ICC0		该位控制 PWM 通道 0-1 互补输出对.	0
		0	根据 IPOL0 设置采用 VAL0 或 VAL1 寄存器数值	

		1	当 PWM 计数器处于上升计数时, 使用 VAL0 寄存器值。 当 PWM 计数器处于下降计数时, 使用 VAL1 寄存器值。	
1	ICC1		该位控制 PWM 通道 2-3 互补输出对。	0
		0	根据 IPOL1 设置采用 VAL2 或 VAL3 寄存器数值	
		1	当 PWM 计数器处于上升计数时, 使用 VAL2 寄存器值。 当 PWM 计数器处于下降计数时, 使用 VAL3 寄存器值。	
2	ICC2		该位控制 PWM 通道 4-5 互补输出对。	0
		0	根据 IPOL2 设置采用 VAL4 或 VAL5 寄存器数值	
		1	当 PWM 计数器处于上升计数时, 使用 VAL4 寄存器值。 当 PWM 计数器处于下降计数时, 使用 VAL5 寄存器值。	
3	-	-	保留	0
4	PAD_EN		PWM 通道管脚输出使能。上电复位时, PWM 通道管脚输出被禁止。管脚输出控制不会影响 PWM 模块内部功能运行。上电复位时管脚输出禁止, 可以保护电力驱动模块安全。	0
		0	管脚输出禁止	
		1	管脚输出使能	
31:5	-	-	保留。	0

15.3.14 PWM 比较输出极性控制寄存器

Table 15-17: PWM 比较输出极性控制(PSCR) 寄存器

位	名称	值	描述	初始值
0	CINV0		PWM 比较输出 0 极性控制	0
		0	当 CNTR 寄存器大于 VAL0, 输出为高电平	
		1	当 CNTR 寄存器小于 VAL0, 输出为高电平	
1	CINV1		PWM 比较输出 1 极性控制	0
		0	当 CNTR 寄存器大于 VAL1, 输出为高电平	
		1	当 CNTR 寄存器小于 VAL1, 输出为高电平	
2	CINV2		PWM 比较输出 2 极性控制	0
		0	当 CNTR 寄存器大于 VAL2, 输出为高电平	
		1	当 CNTR 寄存器小于 VAL2, 输出为高电平	
3	CINV3		PWM 比较输出 3 极性控制	0
		0	当 CNTR 寄存器大于 VAL3, 输出为高电平	
		1	当 CNTR 寄存器小于 VAL3, 输出为高电平	
4	CINV4		PWM 比较输出 4 极性控制	0
		0	当 CNTR 寄存器大于 VAL4, 输出为高电平	
		1	当 CNTR 寄存器小于 VAL4, 输出为高电平	
5	CINV5		PWM 比较输出 5 极性控制	0
		0	当 CNTR 寄存器大于 VAL5, 输出为高电平	
		1	当 CNTR 寄存器小于 VAL5, 输出为高电平	

31:6	-	-	保留	0
------	---	---	----	---

15.3.15 PWM 计数器初始值寄存器

Table 15-18: PWM 计数器初始值 (CNTRINI) 寄存器

位	名称	值	描述	初始值
15:0	INITVAL		在 PWM 计数器启动前, 可通过直接写该寄存器给 PWM 计数器赋初值。这样, PWM 输出可以从任何相位开始。寄存器只可在 PWM 启动前赋值。	0
31:16	-	-	保留	-

15.4 功能描述

15.4.1 时钟预分频

为了降低 PWM 时钟工作频率, PWM 模块允许 PWMn_PCLK (同 System Clock) 按 1, 2, 4, 8 进行分频。CTRL 寄存器中的 PRSC 用来选择不同的分频。分频器选择设定只有在 LDOK 位置 1 后的下一个 PWM 周期或下一个 PWM 重载周期开始时才会生效。

15.4.2 PWM 信号发生器

PWM 发生器包涵了一个 16 位可按软件设定对齐方式上下进行计数的计数器。它可以产生设定的 PWM 载波周期, 占空比和反转的 PWM 信号。

15.4.2.1 输出对齐和极性控制

PWM 模块支持边沿对齐 (Edge-Align) 和中心对齐 (Center-Aligned) 方式产生 PWM 输出波形。CNFG 可以用来选择边沿对齐或中心对齐模式。比较输出极性控制 PWMSCTRL 寄存器的 CINVM 定义 PWM 发生器输出极性。

- 当 VALm 寄存器大于 PWM 计数器当前值并向下计数时, 如对应 CINVM=0, PWM 输出为高; 如对应 CINVM=1, PWM 输出为低。
- 当 VALm 寄存器等于小于 PWM 计数器当前值并向上计数时, 如对应 CINVM=0, PWM 输出为低; 如对应 CINVM=1, PWM 输出为高。

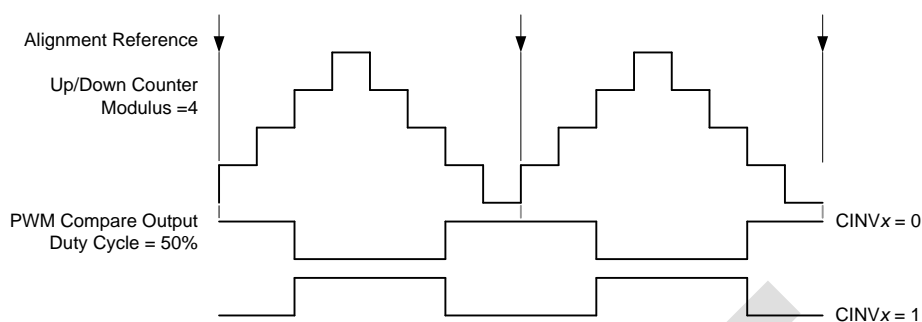


Figure 15-2: 中心对齐 PWM 输出

Note: 由于采用数字比较结构，PWM 模数寄存器（modulu）等于 0 被认为不合法。但在此情况下，死区限制和故障输入条件仍然会得到保证。

15.4.2.2 载波周期

PWM 载波周期由模数寄存器 CMOD 决定。在中心对齐模式下，PWM 计数器以升降方式进行计数。在此模式下，PWM 输出最高分辨率就是 2 个系统时钟如 PWM 选择系统时钟作为工作时钟的话。模数寄存器 CMOD 的时钟数正好是一半 PWM 载波周期时钟。

PWM 载波周期 = (PWM 模数) × (PWM 工作时钟周期) × 2

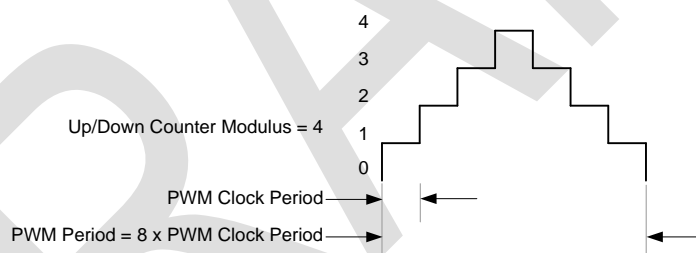


Figure 15-3: 中心对齐 PWM 周期

在边沿对齐模式下，PWM 计数器以升降方式进行计数。PWM 输出最高分辨率就是 1 个系统时钟如 PWM 选择系统时钟作为工作时钟的话。模数寄存器 CMOD 的时钟数正好是 PWM 载波周期时钟。

PWM 载波周期 = (PWM 模数) × (PWM 工作时钟周期)

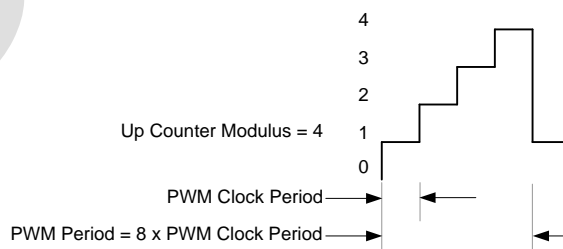


Figure 15-4: 边沿对齐 PWM 周期

15.4.2.3 脉宽占空比

PWM 周期中脉冲宽度由 16 位 PWM 数值寄存器设定的时钟（该时钟为预分频后的时钟）个数决定（如极性设置 $CINVx=1$ ，脉冲宽度则是 PWM 周期时间减 PWM 数值寄存器设定的时钟个数）。

注：在 PWM 数值寄存器值大于等于 PWM 模数值，如 $CINVx=0$ ，PWM 输出会被禁止在整个 PWM 周期；反之如 $CINVx=0$ ，PWM 在整个 PWM 周期输出为高。在 PWM 数值寄存器值等于 0，如 $CINVx=0$ ，PWM 在整个 PWM 周期输出为高；如 $CINVx=1$ ，则 PWM 输出禁止在整个 PWM 周期。

中心对齐方式如图 Figure 15-5 所示。PWM 脉冲宽度等于 2 倍的 PWM 数值寄存器值再乘以 PWM 工作时钟周期。

$$\text{PWM 脉冲宽度} = (\text{PWM 数值寄存器值}) \times (\text{PWM 工作时钟周期}) \times 2$$

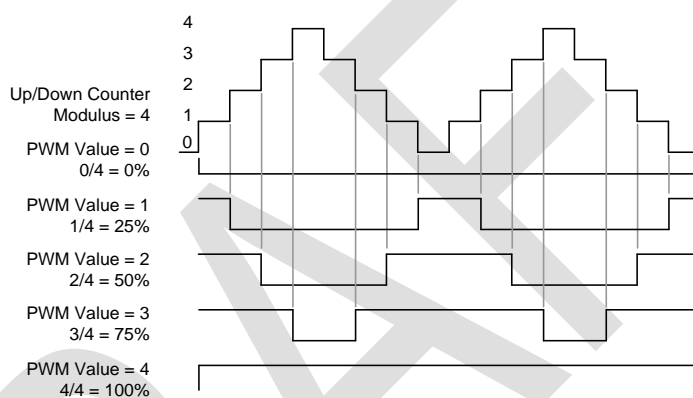


Figure 15-5: 中心对齐方式下脉冲宽度

边沿对齐方式如图 Figure 15-6 所示。PWM 脉冲宽度等于 PWM 通道数值寄存器值再乘以 PWM 工作时钟周期。

$$\text{PWM 脉冲宽度} = (\text{PWM 数值寄存器值}) \times (\text{PWM 工作时钟周期})$$

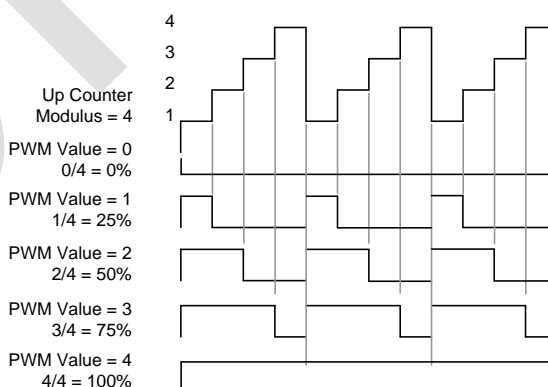


Figure 15-6: 边沿对齐方式下脉冲宽度

15.4.3 独立和互补通道输出

CNFG 寄存器的 INDEPm 位控制 PWM 通道是以独立或互补模式输出。当 INDEPm 为 1 时，PWM 输出对 (0-1,2-3,4-5,6-7) 定义成两个独立的 PWM 输出。每个输出都有自己的 PWM 数字寄存器控制输出的脉宽占空比。

当 INDEPm 为 0 时，PWM 输出对 (0-1,2-3,4-5,6-7) 定义成互补的 PWM 输出通道。如图 Figure 15-7 所示。

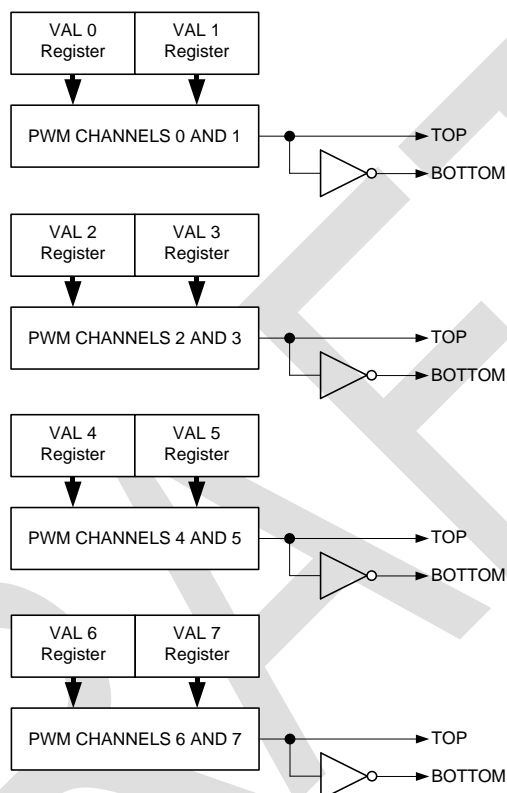


Figure 15-7: 互补通道对

在互补输出模式下，PWM 用于驱动如图 Figure 15-8 所示的类似的逆变器负载。

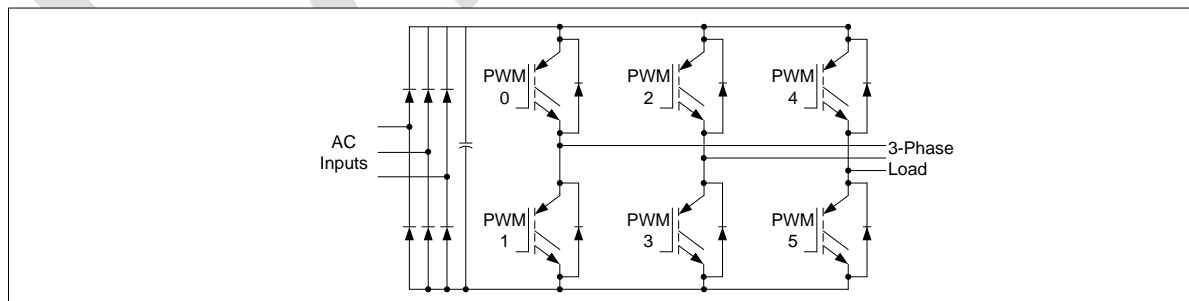


Figure 15-8: 三相交流逆变器

在互补输出模式下，PWM 还具备下列三个功能：

1. 死区插入功能

2. 单独的上下桥 PWM 输出脉宽补偿功能。此功能用来补偿由死区插入带来的波形失真。
3. 单独的上下桥 PWM 输出极性控制

15.4.4 Deadtime Generators

在互补输出模式下，PWM 每一对互补输出分别去驱动逆变器的一个桥臂上下桥晶体管关断，如 Figure 15-9 所示。在理想情况下，PWM 输出通道信号是互补的。即当 PWM 上桥通道输出有效时，下桥通道输出是无效的，反之亦然。为了避免上下桥晶体管短路，上下桥晶体管任何时间都决不可以同时处于打开状态。但是晶体管到关断电气特性可能造成关断时间大于打开时间，从而使上下桥晶体管瞬间有同时打开的现象。为避免这种情况发生，死区时间被插入到上下桥晶体管开关转换期间。

软件可控的死区发生器用来在每对互补 PWM 输出进行转换时自动产生开关延迟。PWM 死区时间寄存器（DTIMx）用来指示死区延迟时间（通过对 PWM 工作时钟的计数）。每次 PWM 输出状态变化时，死区时间都会自动插入。DTIM0 指示 PWM 通道输出从低到高的死区插入时间。DTIM1 指示 PWM 通道输出从高到低的死区插入时间。在死区插入时间内，互补的 PWM 上下桥通道被强制输出无效。死区时间通过增大/减少死区时间寄存器设定来实现。

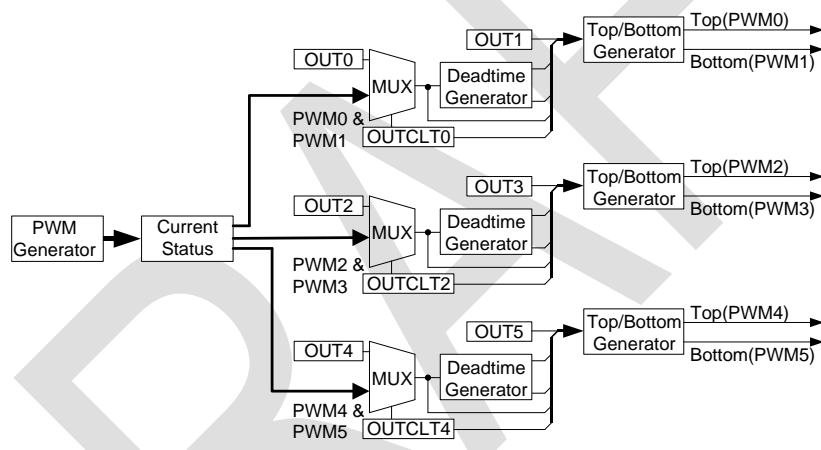


Figure 15-9: 死区发生器

Figure 15-10, Figure 15-11, and Figure 15-12 表示不同情况下的死区插入。

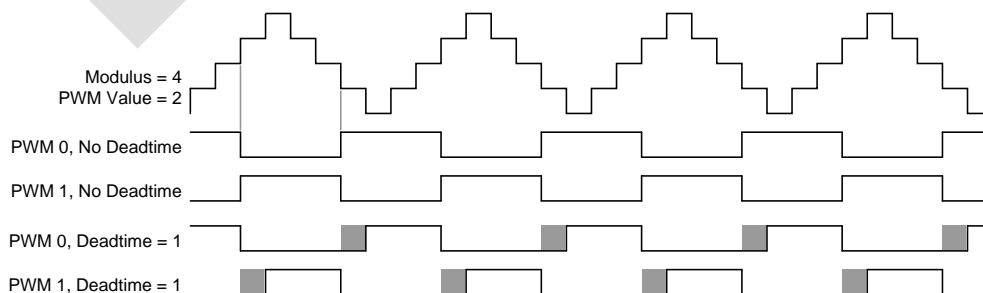


Figure 15-10: 互补输出模式下的死区插入

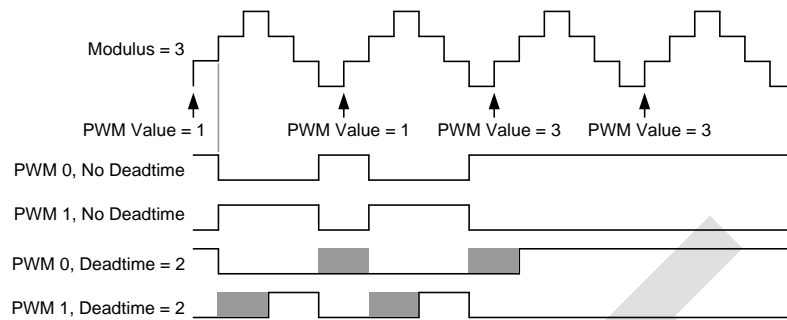


Figure 15-11: 临界占空比下的死区插入

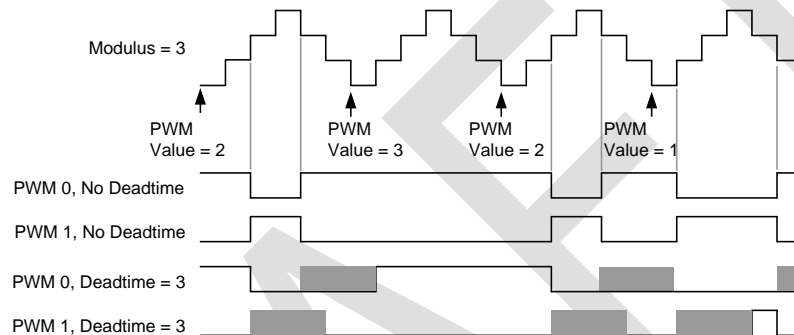


Figure 15-12: 短脉冲下的死区插入

15.4.4.1 死区补偿效应

I 在互补输出模式下控制逆变器时，要么上桥晶体管控制输出电压，要么下桥晶体管控制输出电压。死区时间的插入避免了上下桥开关转换可能引起的晶体管同时打开。但在死区时间内，上下桥晶体管都被关闭，这样会对由负载电流决定的输出电压造成畸变。参照图 Figure 15-13。比较典型的表现是电机负载电流的波形畸变和可见的干扰。

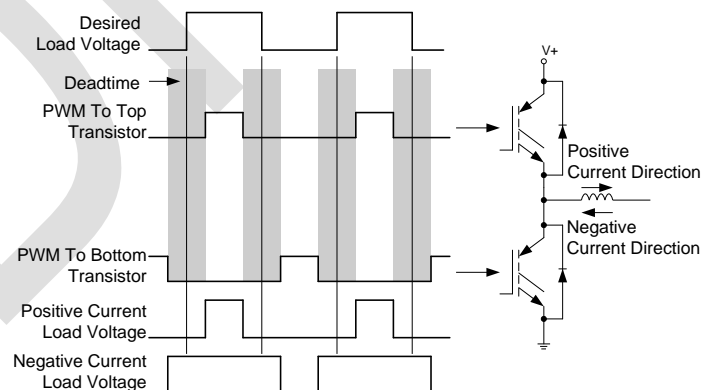


Figure 15-13: Deadtime Distortion

在死区时间内，由于是感性负载，电流通过续流二极管得以保持电感电流的通路，进而对输出电压产生畸变。死区电流产生的输出电压会因电流的流向变化而变化。当电流是正电流流向时，输出电压被钳位在下桥供电电压；当电流是负电流流向时，输

出电压被钳位在上桥供电电压。死区插入，将导致原始的波形的缺失，输出平均值也低于期望值。当死区插入时，负载电流波形发生畸变，这都是由于晶体管开关延迟不一致造成的。通过控制晶体管开关时间，畸变可以得到补偿。

在典型的互补输出控制电路中，任何时候只会有一个晶体管影响并控制输出电压。它跟电流在上下桥的流向有关。为了补偿由于死区插入带来的误差，PWM 数值寄存器需加上误差值从而达到所期望输出效果。因此，需要软件对所在奇偶通道数值寄存器进行补偿值计算，从而去控制脉冲宽度。对于指定的数值寄存器（VALm）对，由下列中条件决定是使用奇数数值寄存器还是偶数数值寄存器：

- 如 ICCTRL 寄存器中的 ICC 置 0，IPOLn 决定使用奇数数值寄存器还是偶数数值寄存器进行补偿。
- 如 ICCTRL 寄存器中的 ICC 置 1，需用软件计算出计数器上升和下降补偿值并放在相应的 VALn 寄存器。
- 在边沿对齐方式下，采用死区时间增加或减少 VALn 寄存器来作补偿。
- 在中心对齐方式下，采用死区时间的一半去增加或减少 VALn 寄存器来作补偿。

注：如果用户可以提供电流流向的检测电路的话，并相应的 ICCn 位清 0，那么 CTRL 寄存器的 IPOL0–IPOL2 位可以根据电流方向决定选择奇/偶 PWM 数值寄存器去作补偿。

Table 15-19：上/下边输出手动补偿

位	逻辑值	输出控制
IPOL0	0	PWMnVAL0 控制 PWMn_0/ PWMn_1 对
	1	PWMnVAL1 控制 PWMn_0/ PWMn_1 对
IPOL1	0	PWMnVAL2 控制 PWMn_2/ PWMn_3 对
	1	PWMnVAL3 控制 PWMn_2/ PWMn_3 对
IPOL2	0	PWMnVAL4 控制 PWMn_4/ PWMn_5 对
	1	PWMnVAL5 控制 PWMn_4/ PWMn_5 对

注：IPOLn 位选择哪一个 PWM 数值寄存器在 PWM 周期生效。如 IPOLn 位在 PWM 周期内被修改，它的值不会立刻生效。无论 LDOK 位设置如何，IPOLn 位都会在下一个周期有效。

电压输出补偿如图 Figure 15-14 和 Figure 15-15 所示。

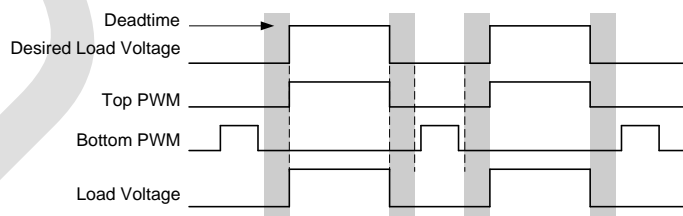


Figure 15-14: 正电流下的死区补偿

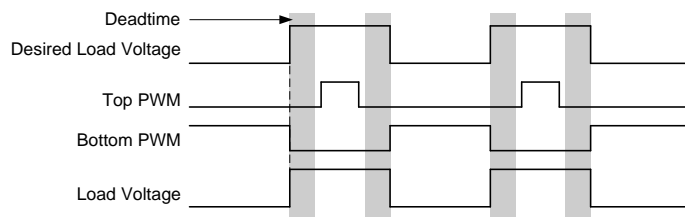


Figure 15-15: 负电流下的死区补偿

15.4.5 非对称 PWM 输出

在中心对齐互补模式下，PWM 占空比可以在每半周期内改变。PWM 计数器在上升或下降计数时选择奇数或偶数 PWM 数值寄存器来产生 PWM 波形。如 PWM 上升计数时，采用偶数 PWM 数值寄存器产生 PWM 波形；在 PWM 下降计数时，采用奇数 PWM 数值寄存器产生 PWM 波形。

Table 15-20: ICCn 位控制的上/下边输出补偿

位	逻辑值	输出控制
ICC0	0	IPOL0 控制 PWMn_0/ PWMn_1 对
	1	PWM 计数方向控制 PWMn_0/ PWMn_1 对
ICC1	0	IPOL1 控制 PWMn_2/ PWMn_3 对
	1	PWM 计数方向控制 PWMn_2/ PWMn_3 对
ICC2	0	IPOL2 控制 PWMn_4/ PWMn_5 对
	1	PWM 计数方向控制 PWMn_4/ PWMn_5 对

注：如 ICCTRL 寄存器的 ICCn 位在 PWM 周期内被改变，这个新的设定会在下一个 PWM 周期生效。

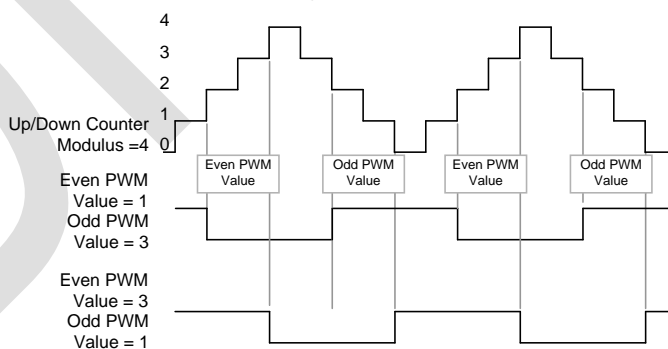


Figure 15-16: 非对称波形 – 相位移位的 PWM 输出

15.4.6 PWM 输出极性控制

PWM 输出极性由下两方面选择控制：

1. **TOPNEG n** 控制 PWM 通道 PWMn_0, PWMn_2, PWMn_4 和 PWMn_6。互补输出模式下用于驱动逆变器的上桥臂晶体管。当 TOPNEG n 置位时，输出低有效。
2. **BOTNEG n** 控制 PWM 通道 PWMn_1, PWMn_3, PWMn_5 和 PWMn_7。互补输出模式下用于驱动逆变器的下桥臂晶体管。当 BOTNEG n 置位时，输出低有效。

所有 TOPNEG n 和 BOTNEG n 位都在寄存器 CNFG 中配置。参照图 Figure 15-17。.

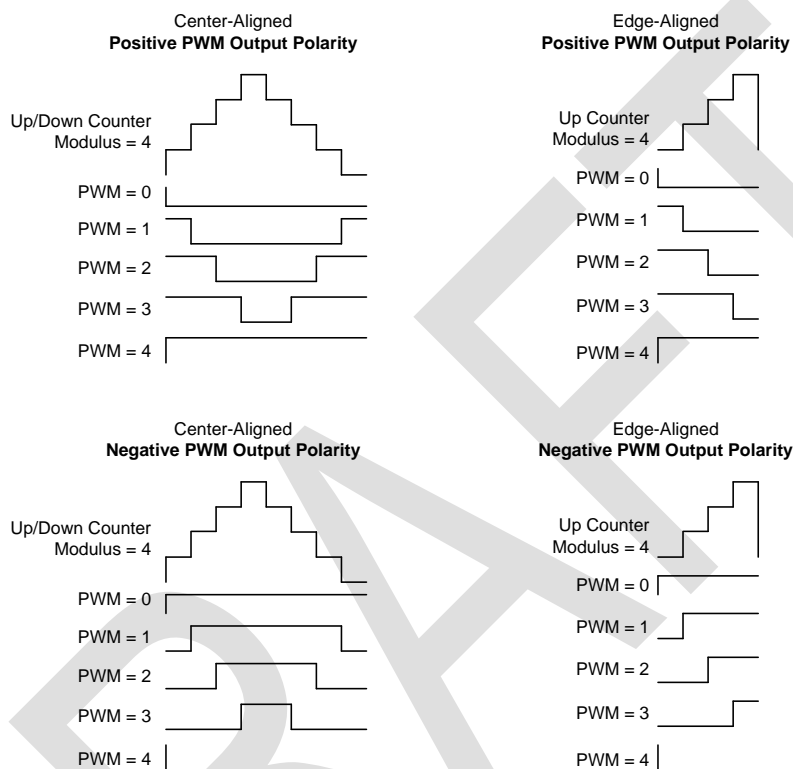


Figure 15-17: PWM Output Polarity

15.4.7 软件输出控制

设置输出控制使能(OUTCTRL n) 位，可以让 PWM 的输出由软件控制，而不是 PWM 信号发生器。在独立输出模式下，当 OUTCTRL $m=1$ 时，寄存器 OUT m 值会输出到 PWMn_ m 通道。在此时，置位/复位 OUT m 位就会导致对应的 PWM 通道输出有效无效变化。OUTCTRL m 和 OUT m 控制位都位于 PWM 输出控制寄存器 OUT 中。在使用软件输出控制时，PWM 的输出极性仍然由 TOPNEG mm 和 BOTNEG mm 控制。

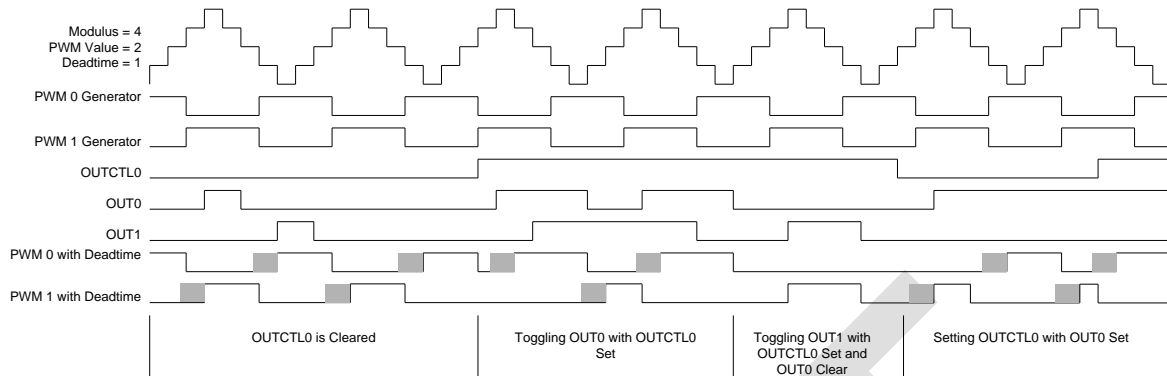


Figure 15-18: 互补模式下的软件输出控制

15.4.8 PWM 发生器设置重载

15.4.8.1 装载使能

LDOK 使能位将修改 PWM 发生器的下列参数：

- 使用 PWM 控制(CTRL)寄存器的 PRSC 设置的预分频。
- 使用模数(CMOD) 寄存器控制 PWM 载波周期
- 使用 VALm 寄存器值控制所有 PWM 通道脉冲宽度

LDOK 允许软件不用同时修改 PWM 所有参数。当 LDOK 置位后，工作频率的预分频，模数值，通道占空比设置都会载入 PWM 缓存器中，并在下一个 PWM 周期开始时正式生效。在新的参数生效后 LDOK 位会被自动清 0。

15.4.8.2 装载频率

CTRL 寄存器的 LFQ 可以用来选择 PWM 在每 1 到 16 个重载机会时，重载 PWM 发生器的寄存器值。不论 LDOK 设置如何，LDFQ 在每个重载机会都会有效。CTRL 寄存器的 HALF 位控制是否允许中心对齐的半周期重载。如 HALF 位被置位，重载可以在周期开始或半周期时发生。如 HALF 位清零，重载只可能在周期开始时发生。当边沿对齐来说，重载只可能在周期开始时发生。

注：在半周期时，重载一个新的模数值（modulu）会强制 PWM 计数器采用模数值减一在接下来的周期计数过程。使能半周期重载只对中心对齐有效，对边沿对齐无效。

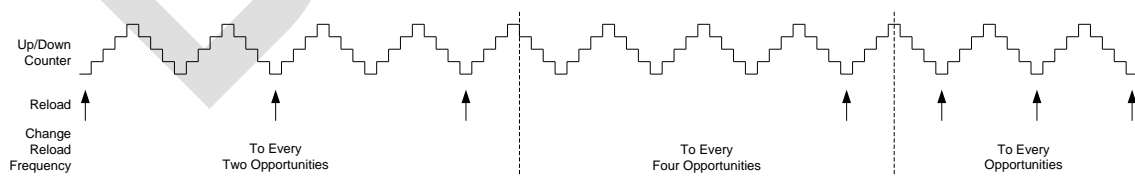


Figure 15-19: 全周期重载频率变化

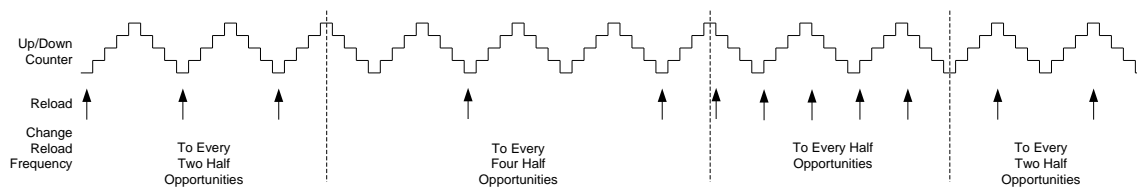


Figure 15-20: 半周期重载频率变化

15.4.8.3 重载标志

在 PWM 每一次重载，不论 LDOK 位状态如何，CTRL 寄存器中的 PWM 重载标志 (PWMF) 位都会被置位。如重载中断使能(PWMRIE) 位置位， PWMF 标志会触发中断，从而软件可以实时计算控制参数。如重载中断使能(PWMRIE) 位没有置位， PWMF 标志仍然会置位，但不会触发中断。可以向 PWMF 位写 0 清除 PWMF 标志。

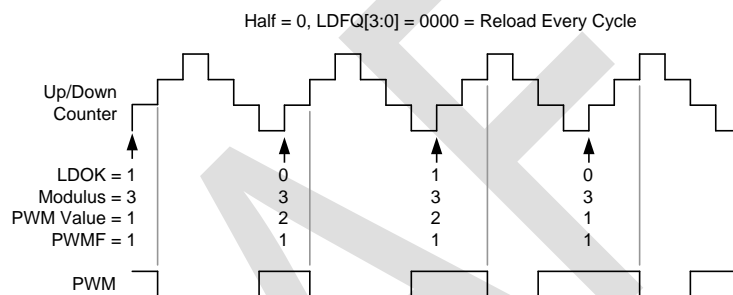


Figure 15-21: 中心对齐的 PWM 数值寄存器全周期重载

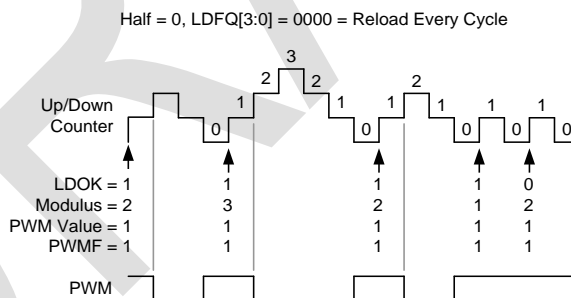


Figure 15-22: 中心对齐的 PWM 模数寄存器全周期重载

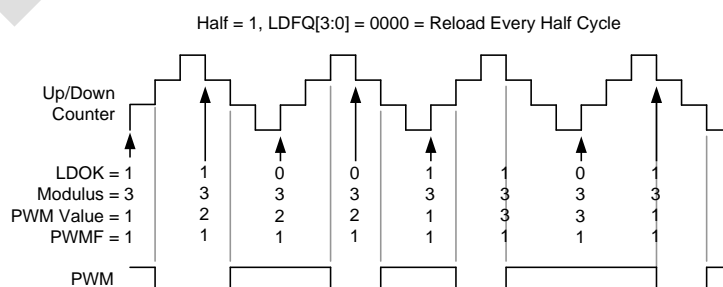


Figure 15-23: 中心对齐的 PWM 数值寄存器半周期重载

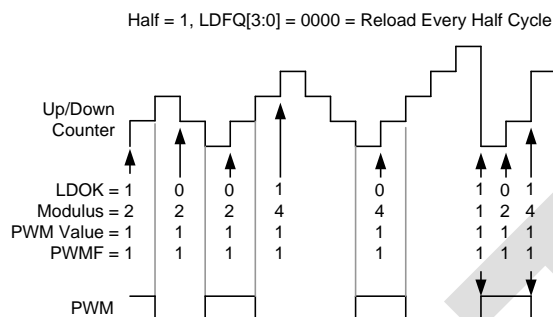


Figure 15-24: 中心对齐的 PWM 模数寄存器半周期重载

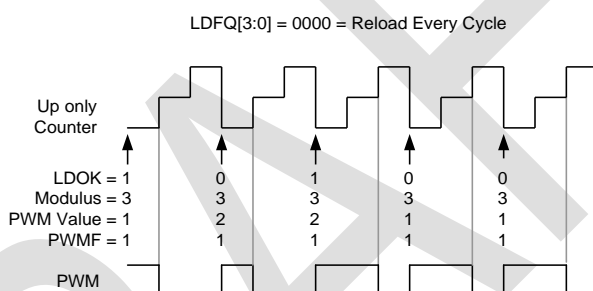


Figure 15-25: 边沿对齐的 PWM 数值寄存器周期重载

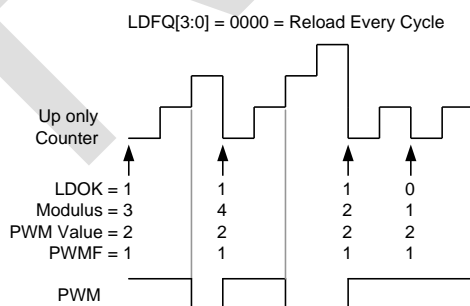


Figure 15-26: 边沿对齐的 PWM 模数寄存器半周期重载

15.4.8.4 同步输出控制

PWM 使用重载事件去同步输出脉冲。它可以用来作为定时器输入信号。无论 LDOK 位状态和重载频率怎样，在每一个 PWM 周期都会有波形输出。

15.4.8.5 PWM 初始化

在 **PWMEN** 置位之前，应初始化所有寄存器和对 **LDOK** 置位。当 **LDOK=1**，**PWMEN** 位置位后，PWM 装载会立即发生，**PWMF** 位也会被置位。**PWMF** 置位导致中断请求发生如果 **PWMRIE** 中断使能的话。在互补模式下，**IPOLn** 位 **s** 和 **ICCn** 位共同决定使用奇或偶数的 PWM 数值寄存器值去控制 PWM 在第一个周期的输出。

注:启动 PWM (**PWMEN** 置位) 会使 **PWMF** 置位甚至 **LDOK** 没有置 1。为防止不必要的中断请求，应在启动 PWM 前清除 **PWMRIE** 位。

如没有设置 **LDOK** 装载使能就启动 PWM 会导致 PWM 预分频为 1，PWM 数值为 1 和未知的模数值。如 **PWMEN** 位置位后再启动之前没有设置 **LDOK** 位，PWM 会使用上一次装载的值去产生 PWM 波形。如 **PWMEN** 或 **OUTCTLm** 位置位后死区设定改变，会发生不正确的死区插入。

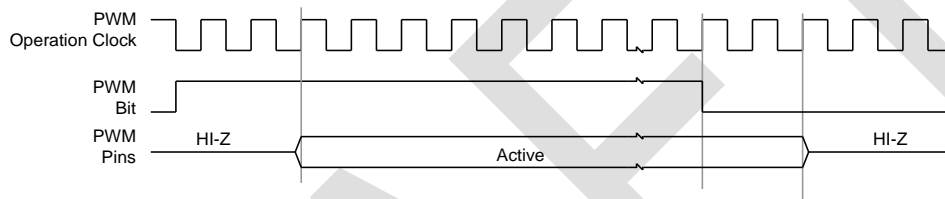


Figure 15-27: PWMEN 和 PWM 管脚在独立模式下工作(**OUTCTL0-5 = 0**)

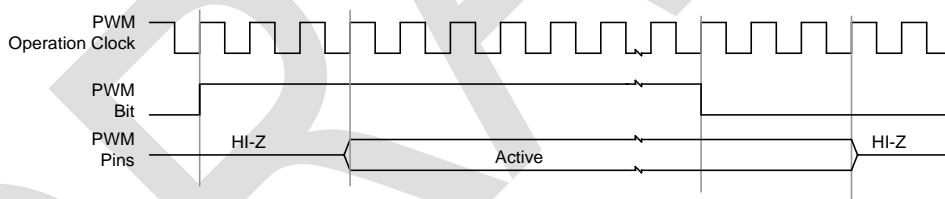


Figure 15-28: PWMEN 和 PWM 管脚在互补模式下工作(**OUTCTL0,2,4 = 0**)

当 **PWMEN** 位置位时:

- **PWMn** 管脚输出失效除非使用软件输出 **OUTCTLn=1**
- PWM 计数器清零并停止计数
- PWM 发生器强制它的输出为 0
- **PWMF** 和挂起中断请求不会被清除
- 所有故障机制仍然工作
- 软件输出仍然保持如果 **OUTCTLn=1**
- 死区插入机制在软件输出时仍然工作

15.4.9 故障保护

PWM 故障保护功能可使任何 PWM 输出无效。故障由 FCTRL 寄存器控制的故障输入管脚的有效极性 0 或 1 产生，或由其它模块信号产生。每个故障输入都可以映射到任何 PWM 输出管脚并使之无效。当故障输入使 PWM 输出无效时，除输出管脚无效外，PWM 模块继续工作。导致 PWM 输出无效的故障译码由故障逻辑和映射寄存器组成。寄存器 DMAP1–2 定义 7 个故障信号的映射到 PWM 的 0~7 的输出无效控制。故障保护甚至在 PWM 没有启动时就已经工作。所以，如果故障信号被锁定，用户必须在启动 PWM 之前清除故障锁定信号，以免产生意外中断。

15.4.9.1 外部故障管脚信号采样滤波

ST32L103xB 有 4 个外部故障信号管脚：PWM_FAULT0 ~ PWM_FAULT3。除管脚滤波外，PWM 故障信号有效遵循下列故障条件：一个有效故障信号必须是在 PWM 操作时钟下连续两次采样都是高电平。此时 FFLAGm 和 FPINm 标志会被置位。FPINm 信号会保持直到在连续的两个 PWM 操作时钟都检测到低电平。写 1 到 FTACKm 位清除 FFLAGm 标志。如 FIE_m，PWM_FAULT_n 管脚中断使能打开，FFLAGm 标志会产生中断请求。中断请求会保持直到下列事件中的一个发生：

- 软件写 1 到 FTACK_m 位去清除 FFLAG_m 标志
- 软件写 0 到 FIE_m 位去屏蔽故障中断
- 系统复位

15.4.9.2 故障的自动清除

在自动清除模式下，当 FMODE_m 被置位，PWM_FAULT_n 管脚恢复到低电平并且新的 PWM 半周期开始，这时失效的 PWM 的输出恢复有效。参照 Figure 15-29。在 FMODE_m 置位时，清除 FFLAG_m 标志不会影响到失效的 PWM 输出。

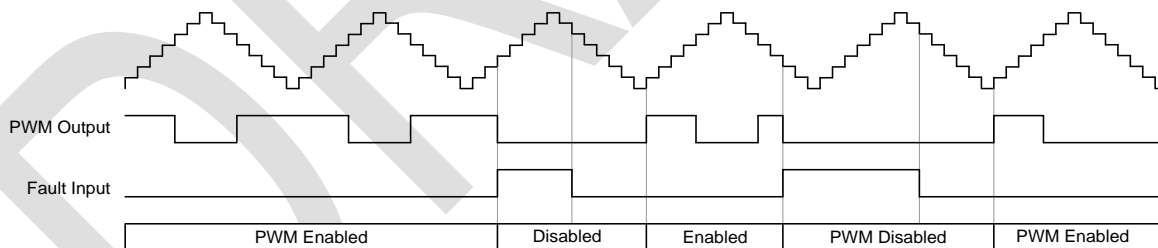


Figure 15-29: 故障自动清除

15.4.9.3 故障的手动清除

在手动模式下，故障输入被分成两组，每组有相同功能。故障输入 0, 2, 4, 5, 6 一组。该组故障输入可以通过软件对 FFLAG 位清零来消除。在这种情况下，PWM 输出可在下个半周期恢复而不论故障输入是否处于高电平。故障输入只有在重新从低电平变为高电平才会使 PWM 失效。参照 [错误!未找到引用源。](#)。

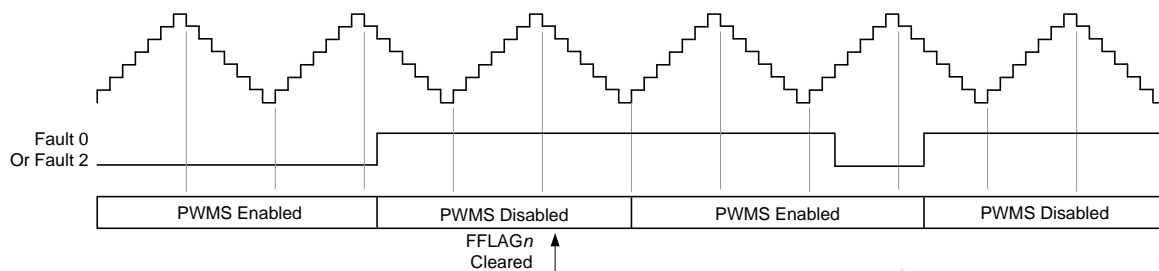


Figure 15-31: 故障的手动清除 0

- PWM 半周期边界是指 PWM 周期开始和计数器值等于 PWM 模数值。对于边沿对齐来说，全周期和半周期是一样的。
- 故障保护对于软件输出模式（OUTCTL m 位置 1）时也是有效的。在 PWM 波形发生器工作时（PWMEN=1），故障清除会在下一个半周期边界发生。如 PWM 波形发生器没有工作（PWMEN=0），那么 OUT m 位会控制 PWM 输出，故障清除会在 PWM 发生器开始工作的第一个周期发生。

15.4.10 中断

有两个 PWM 信号源可以产生不同中断：

- 重载标志 (PWMF)—PWMF 在每一次 PWM 重载周期开始时会被置位。如重载中断使能 (PWMRIE) 位置位，PWMF 就会触发中断。PWMF 和 PWMRIE 都位于 PWM 控制寄存器(CTRL)。
- 故障标志(FFLAG0–FFLAG6)—故障标志 FFLAG m 位在 PWM_FAULT n 管脚和其他故障输入为逻辑 1 时被置位。故障输入中断使能 FIE0–FIE6 决定是否允许 FFLAG m 标志触发中断。FFLAG0–FFLAG6 位于故障状态寄存器(FLTACK)，而 FIE0–FIE6 位于故障输入控制 (FCTRL)。

16 Flash/SRAM 存储器和器件信息

G32F0 有多达 64KB Flash 和 8 KB SRAM 用户存储器空间。Flash 支持在系统编程 (ISP)和在应用编程(IAP)两种编程方式。用户可以通过 G32F0 bootloader 接口来调用 ISP/IAP 功能。

16.1 User Flash/Boot Flash 结构

G32F0 采用统一的 512B 大小扇区架构，共 128 个扇区。用户 Flash 起始于从地址 0 开始的连续空间。

系统引导 ROM 有 5KB 大小用于存放 MCU 的 bootloader。bootloader 地址映射在 0x1FFF0000。

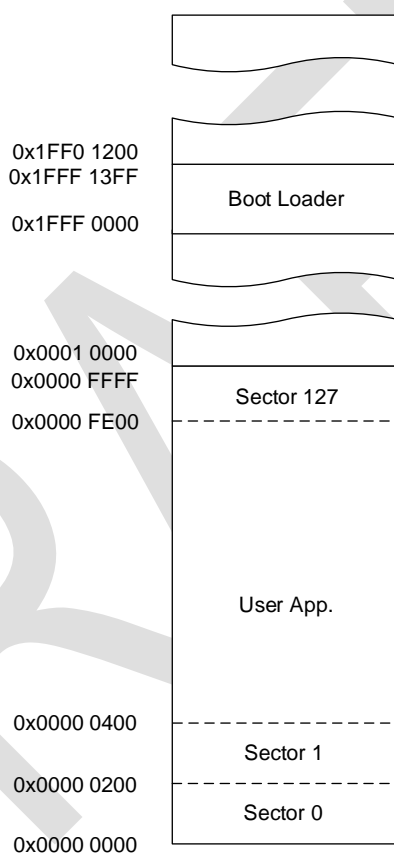


Figure 16-1: 用户程序和引导程序地址映射空间分布

16.2 Flash 操作

16.2.1 Flash 寄存器列表

Table 16-1: Flash 寄存器一览 (基地址: 0x5006 0000)

名称	偏移地址	读写	描述	初始值
FLASH_CMD	0x00	R/W	Flash 操作命令	0x0000 0000
FLASH_DATA	0x04	R/W	Flash 编程数据	0x0000 0000
FLASH_ADDR	0x08	R/W	Flash 编程/擦除地址	0x0000 0000
FLASH_RDCYC	0x0C	R/W	Flash 访问周期	0x0000 0000

16.2.2 Flash 操作命令寄存器

需要对 Flash 区域进行擦除或编程，可以配置 FLASH_CMD 寄存器来实现。

Table 16-2: Flash 操作命令寄存器(FLASH_CMD, 地址 0x5006 0000) 描述

位	名称	值	描述	初始值
7:0	CMD		Flash 访问周期	00
		0x02	编程命令	
		0x04	擦除命令	
		其他值	保留	
8	BUSY		Flash 编程或擦除状态	0
		0	Flash 处于空闲，可以进行擦除或者编程操作	
		1	Flash 操作 busy，需等该 bit 变为 0，才可以进行后面操作	
31:9	-	-	-	0x0

需要对 Flash 进行编程操作时，操作流程如下：

第一步：写入数据 **FLASH_DATA = 0x12345678;**

第二步：写入目标地址 **FLASH_ADDR = 0x00004000;**

第三步：写入命令（0x02 为编程） **FLASH_CMD = 0x02;**

第四步：循环查 Busy 状态， **while((FLASH_CMD&0x100) == 0x100)**，只有当 Busy 变为 0，方可进行下一地址单元操作。

第五步：地址加 0x04，（编程为 word 操作，故地址要加 0x04），返回第一步

需要对 Flash 进行擦除操作时，操作流程如下：

第一步：写入目标地址 `FLASH_ADDR = 0x00004000;`

第二步：写入命令（0x04 为擦除） `FLASH_CMD = 0x04;`

第三步：循环查 Busy 状态， `while((FLASH_CMD&0x100) == 0x100)`，只有当 Busy 变为 0，方可进行下一地址单元操作。

第四步：擦除地址加 0x200 (一条擦除命令可擦除一页数据，一个页大小为 512K Bytes)，返回第一步。

16.2.3 Flash 操作数据寄存器

Flash 编程的数据，FLASH_DATA 寄存器来实现。

Table 16-3: Flash 编程数据寄存器(FLASH_DATA, 地址 0x5006 0004) 描述

位	名称	值	描述	初始值
31:0	DATA		编程数据	0x0000 0000

16.2.4 Flash 操作地址寄存器

Flash 编程或擦除地址寄存器 FLASH_ADDR。

Table 16-4: Flash 地址寄存器(FLASH_ADDR, 地址 0x5006 0008) 描述

位	名称	值	描述	初始值
31:0	ADDR		编程或擦除地址	0x0000 0000

16.2.5 Flash 访问周期寄存器

根据系统时钟频率不同，Flash 访问速度可以配置 FLASH_RDCYC 寄存器来实现。

Table 16-5: Flash 访问周期寄存器(FLASH_RDCYC, 地址 0x5006 0028) 描述

位	名称	值	描述	初始值
1:0	CYCLES		Flash 访问周期	00
		00	1 个时钟周期。推荐在系统时钟（system clock）在 30MHz 下使用。	
		01	2 个时钟周期。推荐在系统时钟（system clock）在 60MHz 下使用。	
31:2	-	-	-	0x0

17 SWD

17.1 概述

G32F0 提供 SWD/JTAG 调试接口支持标准 ARM Serial Wire Debug mode。功能有：

- 可直接对所有存储器、寄存器和外设进行调试。
- 调试阶段不需要占用目标资源。
- 4 个断点。4 个指令断点，可以用于重映射代码补丁的指令地址。2 个数据比较器，可用于将补丁的地址重映射到字面值。
- 2 个数据观察点，可用作跟踪触发器。

17.2 管脚描述

Table 17-1: SWD 管脚 描述

名称	类型	描述
SWCLK	输入	串行接口时钟。此管脚在串行调试接口模式下是调试逻辑的时钟(SWDCLK)。
SWDIO	输入/输出	串行调试接口数据输入/ 输出。外部调试工具可通过 SWDIO 管脚与 ARM Cortex™-M3 CPU 通信，并对其进行控制。

17.3 调试操作

SWD 管脚是复用管脚。系统上电复位缺省设置是 SWD 功能。如用户要将 SWD 管脚作它用，需对管脚功能进行配置。

SWD 调试功能在 A 区保护下不能工作。用户需取消保护才能进行调试。

SWD 在省电模式（睡眠，停止模式，掉电）下不能唤醒系统，无法工作。

下图是 SWD 管脚连接图：

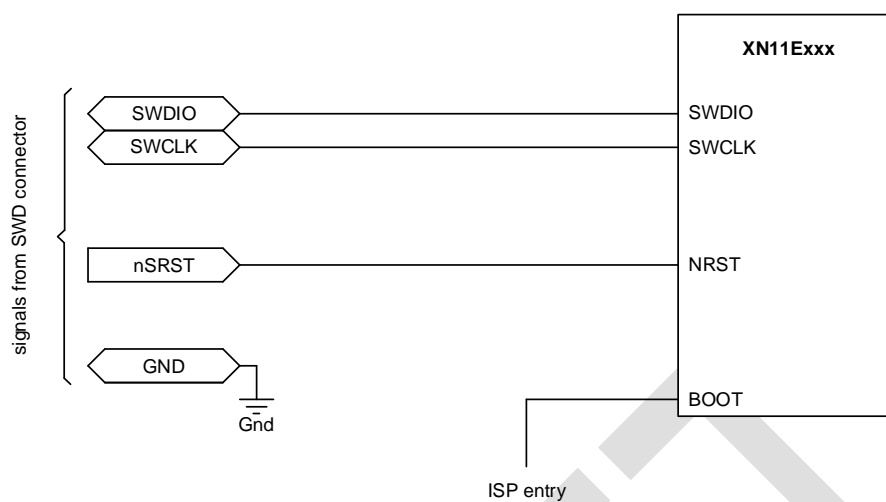


Figure 17-1: SWD 连接器与 SWD 管脚连接示意图

18 Revision History

Revision	Description	Date
1.0	Initial Release.	May 2016