

32 位 MCU  
FS030

# 数 据 手 册

- ☐ 产品简介
- ☒ 数据手册
- ☐ 产品规格

2024 年 12 月 17 日

## FS030 系列 MCU 芯片使用注意事项

### 关于芯片的上/下电

FS030 系列 MCU 芯片具有独立电源管脚。当 MCU 芯片应用在多电源供电系统时，应先对 MCU 芯片上电，再对系统其他部件上电；反之，下电时，先对系统其他部件下电，再对 MCU 芯片下电。若操作顺序相反则可能导致芯片内部元件过压或过流，从而导致芯片故障或元件退化。具体可参照芯片的数据手册说明。

### 关于芯片的复位

FS030 系列 MCU 芯片具有内部上电复位。对于不同的快速上/下电或慢速上/下电系统，内部上电复位电路可能失效，建议用户使用外部复位、下电复位、看门狗复位等，确保复位电路正常工作。在系统设计时，若使用外部复位电路，建议采用三极管复位电路、RC 复位电路。若不使用外部复位电路，建议采用复位管脚接电阻到电源，或采取必要的电源抖动处理电路或其他保护电路。具体可参照芯片的数据手册说明。

### 关于芯片的时钟

FS030 系列 MCU 芯片具有内部和外部时钟源。内部时钟源会随着温度、电压变化而偏移，可能会影响时钟源精度；外部时钟源采用陶瓷、晶体振荡器电路时，建议使能起振延时；使用 RC 振荡电路时，需考虑电容、电阻匹配；采用外部有源晶振或时钟输入时，需考虑输入高/低电平电压。具体可参照芯片的数据手册说明。

### 关于芯片的初始化

FS030 系列 MCU 芯片具有各种内部和外部复位。对于不同的应用系统，有必要对芯片寄存器、内存、功能模块等进行初始化，尤其是 I/O 管脚复用功能进行初始化，避免由于芯片上电以后，I/O 管脚状态的不确定情况发生。

### 关于芯片的管脚

FS030 系列 MCU 芯片具有宽范围的输入管脚电平，建议用户输入高电平应在  $V_{IHMIN}$  之上，低电平应在  $V_{ILMAX}$  之下。避免输入电压介于  $V_{IHMIN}$  和  $V_{ILMAX}$  之间，以免波动噪声进入芯片。对于未使用的输入/输出管脚，建议用户设为输入状态，并通过电阻上拉至电源或下拉至地，或设置为输出管脚，输出固定电平并浮空。对未使用的管脚处理因应用系统而异，具体遵循应用系统的相关规定和说明。

### 关于芯片的 ESD 防护措施

FS030 系列 MCU 芯片具有满足工业级 ESD 标准保护电路。建议用户根据芯片存储/应用的环境采取适当静电防护措施。应注意应用环境的湿度；建议避免使用容易产生静电的绝缘体；存放和运输应在抗静电容器、抗静电屏蔽袋或导电材料容器中；包括工作台在内的所有测试和测量工具必须保证接地；操作者应该佩戴静电消除手腕环手套，不能用手直接接触芯片等。

### 关于芯片的 EFT 防护措施

FS030 系列 MCU 芯片具有满足工业级 EFT 标准的保护电路。当 MCU 芯片应用在 PCB 系统时，需要遵守 PCB 相关设计要求，包括电源、地走线（包括数字/模拟电源分离，单/多点接地等）、复位管脚保护电路、电源和地之间的去耦电容、高低频电路单独分别处理以及单/多层板选择等。

注：在产品开发时，如遇到不清楚的地方，请与销售人员联系。

产品订购信息

型 号	FLASH	RAM	I/O	Timer	UART	SPI	I2C	ADC	LVD	封装类型
FS030LK	64KB	4KB	30	4+1	3	1	1	12bit×19	1	LQFP32
FS030NK										QFN32

CONFIDENTIAL

修订历史

版本	修改日期	更改概要
V1.0	2024-12-17	初版发布

CONFIDENTIAL

# 目 录

## 内容目录

<b>第 1 章</b>	<b>芯片简介 .....</b>	<b>18</b>
1.1	概述 .....	18
1.2	应用领域 .....	20
1.3	结构框图 .....	21
1.4	管脚分配图 .....	22
1.4.1	LQFP32 封装图 .....	22
1.4.2	QFN32 封装图 .....	23
1.5	管脚说明 .....	23
1.5.1	管脚说明 .....	23
1.5.2	管脚对照表 .....	25
<b>第 2 章</b>	<b>系统控制及操作特性 .....</b>	<b>27</b>
2.1	系统控制保护 .....	27
2.1.1	概述 .....	27
2.1.2	特殊功能寄存器 .....	27
2.1.2.1	系统设置保护寄存器 (SCU_PROT) .....	27
2.2	系统电源 .....	27
2.2.1	结构框图 .....	27
2.2.2	芯片供电电源 .....	28
2.3	系统复位 .....	28
2.3.1	概述 .....	28
2.3.2	结构框图 .....	28
2.3.3	复位时序图 .....	28
2.3.4	外部复位 MRSTN 参考 .....	30
2.3.5	特殊功能寄存器 .....	31
2.3.5.1	复位寄存器 (SCU_PWRC) .....	31
2.4	低电压监测 (LVD) .....	32
2.4.1	概述 .....	32
2.4.2	特殊功能寄存器 .....	33
2.4.2.1	低电压监测控制寄存器 (SCU_LVDCON) .....	33
2.5	系统低功耗操作模式 .....	35
2.5.1	概述 .....	35
2.5.2	浅睡眠模式 .....	35
2.5.3	深度睡眠模式 .....	35
2.5.4	睡眠模式的唤醒 .....	36
2.5.5	睡眠模式的唤醒时间 .....	36
2.5.6	FLASH 存储器等待功能 .....	36
2.5.7	特殊功能寄存器 .....	37
2.5.7.1	FLASH 访问等待时间寄存器 (SCU_FLASHWAIT) .....	37
2.6	系统时钟 .....	38
2.6.1	概述 .....	38
2.6.2	结构框图 .....	39

2.6.3	功能说明 .....	39
2.6.3.1	外部时钟 XTAL .....	39
2.6.3.2	内部高速时钟 HRC .....	40
2.6.3.3	内部低速时钟 LRC .....	41
2.6.3.4	外部时钟停振检测 CCM .....	41
2.6.3.5	时钟滤波 CLKFLT .....	41
2.6.3.6	睡眠模式系统状态 .....	41
2.6.3.7	浅睡眠模式 .....	42
2.6.3.8	深度睡眠模式 .....	42
2.6.4	特殊功能寄存器 .....	42
2.6.4.1	系统时钟控制寄存器 0 (SCU_SCLKEN0) .....	42
2.6.4.2	系统时钟控制寄存器 1 (SCU_SCLKEN1) .....	43
2.6.4.3	外设时钟控制寄存器 (SCU_PCLKEN) .....	44
2.6.4.4	系统唤醒时间控制寄存器 (SCU_WAKEUPTIME) .....	46
2.6.4.5	外部时钟检测控制寄存器 (SCU_CCM) .....	47
2.6.5	系统时钟应用说明 .....	48
2.6.5.1	外部时钟 XTAL .....	48
2.6.5.2	内部高速时钟 HRC .....	49
2.6.5.3	内部低速时钟 LRC .....	49
2.7	中断和异常处理 .....	51
2.7.1	中断和异常 .....	51
2.7.2	中断和异常向量的分配 .....	52
2.7.3	中断向量表的重映射 .....	53
2.7.4	特殊功能寄存器 .....	53
2.7.4.1	不可屏蔽中断制寄存器 (SCU_NMICON) .....	53
2.7.4.2	中断向量表重映射使能寄存器 (SCU_TBLREMAPEN) .....	54
2.7.4.3	中断向量表偏移寄存器 (SCU_TBLOFF) .....	54
2.7.4.4	硬件错误标志寄存器 (SCU_FAULTFLAG) .....	55
2.7.4.5	IRQ0~31 置中断请求使能寄存器 (NVIC_ISER) .....	55
2.7.4.6	IRQ0~31 清中断请求使能寄存器 (NVIC_ICER) .....	56
2.7.4.7	IRQ0~31 置中断挂起寄存器 (NVIC_ISPR) .....	56
2.7.4.8	IRQ0~31 清中断挂起寄存器 (NVIC_ICPR) .....	57
2.7.4.9	IRQ0~3 优先级控制寄存器 (NVIC_PR0) .....	57
2.7.4.10	IRQ4~7 优先级控制寄存器 (NVIC_PR1) .....	58
2.7.4.11	IRQ8~11 优先级控制寄存器 (NVIC_PR2) .....	58
2.7.4.12	IRQ12~15 优先级控制寄存器 (NVIC_PR3) .....	59
2.7.4.13	IRQ16~19 优先级控制寄存器 (NVIC_PR4) .....	60
2.7.4.14	IRQ20~23 优先级控制寄存器 (NVIC_PR5) .....	60
2.7.4.15	IRQ24~27 优先级控制寄存器 (NVIC_PR6) .....	61
2.7.4.16	IRQ28~31 优先级控制寄存器 (NVIC_PR7) .....	62
2.8	系统控制块 (SCB) .....	63
2.8.1	概述 .....	63
2.8.2	特殊功能寄存器 .....	63
2.8.2.1	SCB_CPUID 寄存器 (SCB_CPUID) .....	63

2.8.2.2	中断控制和状态寄存器 (SCB_ICSR)	63
2.8.2.3	应用中断和复位控制寄存器 (SCB_AIRCR)	64
2.8.2.4	系统控制寄存器 (SCB_SCR)	65
2.8.2.5	配置和控制寄存器 (SCB_CCR)	65
2.8.2.6	系统处理程序优先级寄存器 2 (SCB_SHPR2)	66
2.8.2.7	系统处理程序优先级寄存器 3 (SCB_SHPR3)	66
2.9	系统定时器 (SYSTICK)	67
2.9.1	概述	67
2.9.2	特殊功能寄存器	68
2.9.2.1	SYSTICK 控制和状态寄存器 (SYST_CSR)	68
2.9.2.2	SYSTICK 重装值寄存器 (SYST_RVR)	68
2.9.2.3	SYSTICK 当前值寄存器 (SYST_CVR)	69
2.9.2.4	SYSTICK 校准值寄存器 (SYST_CALIB)	69
2.10	定时器 (T16N/T32N) 同步启动关停控制	70
2.10.1	概述	70
2.10.2	特殊功能寄存器	70
2.10.2.1	SCU_TIMEREN 使能控制寄存器 (SCU_TIMEREN)	70
2.10.2.2	SCU_TIMERDIS 关停控制寄存器 (SCU_TIMERDIS)	71
<b>第 3 章</b>	<b>存储器资源</b>	<b>72</b>
3.1	内部存储器地址映射	72
3.2	FLASH 存储器	72
3.2.1	信息区 FLASH	72
3.2.1.1	芯片配置字	72
3.2.1.2	芯片唯一识别码 UID	76
3.2.2	程序区 FLASH	77
3.3	功能描述	77
3.3.1	IAP 概述	77
3.3.2	Flash 保护	77
3.3.2.1	IAP 操作保护 KEY	77
3.3.2.2	Flash 写保护区	77
3.3.2.3	Flash 私有读保护区	78
3.3.2.4	数据 Flash 区	78
3.3.2.5	Flash 全局读保护	78
3.3.3	Flash 程序区全擦除	79
3.3.4	Flash 非私有读保护区全擦除	79
3.3.5	Flash 页擦除	79
3.3.6	Flash 字编程	80
3.3.7	Flash 编程数据 FIFO	80
3.3.8	IAP 自编程硬件固化模块	80
3.3.8.1	CODE 区单页擦函数	80
3.3.8.2	CODE 区单字编程函数	80
3.3.8.3	CODE 区多字编程	81
3.3.8.4	DATA 区单页擦函数	81
3.3.8.5	DATA 区单字编程函数	81

3.3.8.6	DATA 区多字编程 .....	81
3.3.9	特殊功能寄存器.....	82
3.3.9.1	FLASH 程序区关键码寄存器 (IAP_FLASHKEY) .....	82
3.3.9.2	FLASH 擦除编程地址寄存器 (IAP_FLASHADDR) .....	82
3.3.9.3	FLASH 编程数据 FIFO 寄存器 (IAP_FLASHFIFO) .....	83
3.3.9.4	FLASH 编程数据寄存器 (IAP_FLASHDR) .....	83
3.3.9.5	FLASH 操作命令寄存器 (IAP_FLASHCMD) .....	83
3.3.9.6	FLASH 控制寄存器 (IAP_FLASHCR) .....	84
3.3.9.7	FLASH 状态寄存器 (IAP_FLASHSR) .....	85
3.3.9.8	FLASH 擦除编程地址反码寄存器 (IAP_FLASHADDINV) .....	86
3.4	数据存储 (SRAM) .....	87
3.4.1	SRAM 地址映射 .....	87
3.4.2	SRAM 位带扩展 .....	87
3.5	外设寄存器 .....	88
3.5.1	外设寄存器映射 .....	88
3.5.2	外设寄存器位带扩展 .....	88
3.5.3	系统控制单元 (SCU) 寄存器列表 .....	89
3.5.4	GPIO 寄存器列表 .....	89
3.5.5	IAP 寄存器列表 .....	91
3.5.6	ADC 寄存器列表 .....	91
3.5.7	IWDT 寄存器列表 .....	91
3.5.8	WWDT 寄存器列表 .....	92
3.5.9	T16N0/T16N1/T16N2/T16N3 寄存器列表 .....	92
3.5.10	T32N0 寄存器列表 .....	93
3.5.11	UART0/UART1/UART2 寄存器列表 .....	93
3.5.12	SPI1 寄存器列表 .....	93
3.5.13	I2C0 寄存器列表 .....	94
3.6	内核寄存器 .....	94
3.6.1	系统定时器 (SYSTICK) 寄存器列表 .....	94
3.6.2	中断控制器 (NVIC) 寄存器列表 .....	94
3.6.3	系统控制块 (SCB) 寄存器列表 .....	95
第 4 章	输入输出端口 (GPIO) .....	96
4.1	概述 .....	96
4.2	结构框图 .....	97
4.3	外部端口中断 .....	97
4.4	外部按键中断 .....	98
4.5	Buzz 输出 .....	99
4.6	特殊功能寄存器 .....	100
4.6.1	PA 端口状态寄存器 (GPIO_PAPORT) .....	100
4.6.2	PA 端口数据寄存器 (GPIO_PADATA) .....	100
4.6.3	PA 端口输出置位寄存器 (GPIO_PADATABSR) .....	100
4.6.4	PA 端口输出清零寄存器 (GPIO_PADATABCR) .....	101
4.6.5	PA 端口输出翻转寄存器 (GPIO_PADATABRR) .....	101
4.6.6	PA 端口方向控制寄存器 (GPIO_PADIR) .....	101



4. 6. 7	PA 端口方向置位寄存器 (GPIO_PADIRBSR) .....	102
4. 6. 8	PA 端口方向清零寄存器 (GPIO_PADIRBCR) .....	102
4. 6. 9	PA 端口方向翻转寄存器 (GPIO_PADIRBRR) .....	102
4. 6. 10	PA<7:0>端口复用选择寄存器 (GPIO_PAFUNC0) .....	103
4. 6. 11	PA<15:8>端口复用选择寄存器 (GPIO_PAFUNC1) .....	104
4. 6. 12	PA<23:16>端口复用选择寄存器 (GPIO_PAFUNC2) .....	105
4. 6. 13	PA<31:24>端口复用选择寄存器 (GPIO_PAFUNC3) .....	105
4. 6. 14	PA 端口输入控制寄存器 (GPIO_PAINEB) .....	106
4. 6. 15	PA 端口开漏控制寄存器 (GPIO_PAODE) .....	106
4. 6. 16	PA 端口弱上拉使能寄存器 (GPIO_PAPUE) .....	107
4. 6. 17	PA 端口弱下拉使能寄存器 (GPIO_PAPDE) .....	107
4. 6. 18	PA 端口驱动电流控制寄存器 (GPIO_PADS) .....	107
4. 6. 19	PA 端口类型选择寄存器 (GPIO_PATYP) .....	108
4. 6. 20	PA 端口滤波控制寄存器 (GPIO_PAFLT) .....	108
4. 6. 21	PB 端口状态寄存器 (GPIO_PBPORT) .....	108
4. 6. 22	PB 端口数据寄存器 (GPIO_PBDATA) .....	109
4. 6. 23	PB 端口输出置位寄存器 (GPIO_PBDATABSR) .....	109
4. 6. 24	PB 端口输出清零寄存器 (GPIO_PBDATABCR) .....	109
4. 6. 25	PB 端口输出翻转寄存器 (GPIO_PBDATABRR) .....	110
4. 6. 26	PB 端口方向控制寄存器 (GPIO_PBDIR) .....	110
4. 6. 27	PB 端口方向置位寄存器 (GPIO_PBDIRBSR) .....	111
4. 6. 28	PB 端口方向清零寄存器 (GPIO_PBDIRBCR) .....	111
4. 6. 29	PB 端口方向翻转寄存器 (GPIO_PBDIRBRR) .....	111
4. 6. 30	PB<7:0>端口复用选择寄存器 (GPIO_PBFUNC0) .....	112
4. 6. 31	PB<15:8>端口复用选择寄存器 (GPIO_PBFUNC1) .....	112
4. 6. 32	PB 端口输入控制寄存器 (GPIO_PBINEB) .....	113
4. 6. 33	PB 端口开漏控制寄存器 (GPIO_PBODE) .....	113
4. 6. 34	PB 端口弱上拉使能寄存器 (GPIO_PBPUE) .....	114
4. 6. 35	PB 端口弱下拉使能寄存器 (GPIO_PBPDE) .....	114
4. 6. 36	PB 端口驱动电流控制寄存器 (GPIO_PBDS) .....	114
4. 6. 37	PB 端口类型选择寄存器 (GPIO_PBTYP) .....	115
4. 6. 38	PB 端口滤波控制寄存器 (GPIO_PBFLT) .....	115
4. 6. 39	PINT 中断使能寄存器 (GPIO_PINTIE) .....	115
4. 6. 40	PINT 中断标志寄存器 (GPIO_PINTIF) .....	116
4. 6. 41	PINT0~7 中断源选择寄存器 (GPIO_PINTSEL) .....	116
4. 6. 42	PINT 中断配置寄存器 (GPIO_PINTCFG) .....	117
4. 6. 43	KINT 中断使能寄存器 (GPIO_KINTIE) .....	118
4. 6. 44	KINT 中断标志寄存器 (GPIO_KINTIF) .....	119
4. 6. 45	KINT0~7 中断源选择寄存器 (GPIO_KINTSEL) .....	119
4. 6. 46	KINT 中断配置寄存器 (GPIO_KINTCFG) .....	120
4. 6. 47	脉宽调制寄存器 (GPIO_TXPWM) .....	121
4. 6. 48	BUZ 控制寄存器 (GPIO_BUZC) .....	122
<b>第 5 章</b>	<b>外设 .....</b>	<b>123</b>
5. 1	定时器/计数器 .....	123

5.1.1	16 位定时器/计数器 T16N	123
5.1.1.1	概述	123
5.1.1.2	结构框图	124
5.1.1.3	T16N 定时/计数功能	124
5.1.1.4	T16N 输入捕捉功能	126
5.1.1.5	T16N 输出调制功能	127
5.1.1.6	特殊功能寄存器	131
5.1.1.7	T16N 计数值寄存器 0 (T16N_CNT0)	131
5.1.1.8	T16N 计数值寄存器 1 (T16N_CNT1)	131
5.1.1.9	T16N 预分频器计数值寄存器 (T16N_PRECNT)	132
5.1.1.10	T16N 预分频器计数匹配寄存器 (T16N_PREMAT)	132
5.1.1.11	T16N 控制寄存器 0 (T16N_CON0)	132
5.1.1.12	T16N 控制寄存器 1 (T16N_CON1)	134
5.1.1.13	T16N 控制寄存器 2 (T16N_CON2)	135
5.1.1.14	T16N 中断使能寄存器 (T16N_IE)	137
5.1.1.15	T16N 中断标志寄存器 (T16N_IF)	138
5.1.1.16	PWM 调制模式死区宽度寄存器 (T16N_PDZ)	140
5.1.1.17	PWM 调制模式 ADC 触发寄存器 (T16N_PTR)	140
5.1.1.18	T16N 计数匹配寄存器 0 (T16N_MAT0)	141
5.1.1.19	T16N 计数匹配寄存器 1 (T16N_MAT1)	141
5.1.1.20	T16N 计数匹配寄存器 2 (T16N_MAT2)	141
5.1.1.21	T16N 计数匹配寄存器 3 (T16N_MAT3)	142
5.1.1.22	T16N_CNT0 计数峰值寄存器 0 (T16N_TOP0)	142
5.1.1.23	T16N_CNT1 计数峰值寄存器 1 (T16N_TOP1)	142
5.1.1.24	T16N 应用说明	142
5.1.2	32 位定时器/计数器 T32N (T32N0)	143
5.1.2.1	概述	143
5.1.2.2	结构框图	143
5.1.2.3	T32N 定时/计数功能	143
5.1.2.4	T32N 输入捕捉功能	145
5.1.2.5	T32N 输出调制功能	146
5.1.2.6	特殊功能寄存器	147
5.1.2.7	T32N 计数值寄存器 (T32N_CNT)	147
5.1.2.8	T32N 控制寄存器 0 (T32N_CON0)	148
5.1.2.9	T32N 控制寄存器 1 (T16N_CON1)	149
5.1.2.10	T32N 预分频器计数值寄存器 (T32N_PRECNT)	151
5.1.2.11	T32N 预分频器计数匹配寄存器 (T32N_PREMAT)	151
5.1.2.12	T32N 中断使能寄存器 (T32N_IE)	151
5.1.2.13	T32N 中断标志寄存器 (T32N_IF)	152
5.1.2.14	T32N 计数匹配寄存器 0 (T32N_MAT0)	153
5.1.2.15	T32N 计数匹配寄存器 1 (T32N_MAT1)	154
5.1.2.16	T32N 计数匹配寄存器 2 (T32N_MAT2)	154
5.1.2.17	T32N 计数匹配寄存器 3 (T32N_MAT3)	154
5.1.2.18	T32N 应用说明	154

5.2	通用异步接收/发送器 (UART0/ UART1/UART2)	155
5.2.1	概述	155
5.2.2	结构框图	155
5.2.3	UART 数据格式	155
5.2.4	UART 异步发送器	156
5.2.5	UART 异步接收器	158
5.2.6	UART 发送调制功能	160
5.2.7	UART 红外唤醒功能	161
5.2.8	UART 端口极性	161
5.2.9	UART 自动波特率检测	161
5.2.10	UART 空闲帧检测	162
5.2.11	UART 发送暂停和接收暂停	163
5.2.12	特殊功能寄存器	163
5.2.12.1	UART 控制寄存器 0 (UART_CON0)	163
5.2.12.2	UART 控制寄存器 1 (UART_CON1)	165
5.2.12.3	UART 波特率寄存器 (UART_BRR)	166
5.2.12.4	UART 状态寄存器 (UART_STA)	167
5.2.12.5	UART 中断使能寄存器 (UART_IE)	168
5.2.12.6	UART 中断标志寄存器 (UART_IF)	169
5.2.12.7	UART 发送数据写入寄存器 (UART_TBW)	171
5.2.12.8	UART 接收数据读取寄存器 (UART_RBR)	171
5.2.12.9	UART 发送缓冲寄存器 0 (UART_TB0)	171
5.2.12.10	UART 接收缓冲寄存器 0 (UART_RB0)	172
5.2.13	UART 应用说明	172
5.3	SPI 同步串口通讯控制器 (SPI1)	173
5.3.1	概述	173
5.3.2	结构框图	173
5.3.3	SPI 通讯模式	173
5.3.4	SPI 数据格式	173
5.3.5	SPI 帧位宽	175
5.3.6	SPI 同步发送器	175
5.3.7	SPI 同步接收器	176
5.3.8	SPI 通讯控制	177
5.3.9	SPI 延迟接收功能	177
5.3.10	SPI 数据帧发送间隔功能	178
5.3.11	特殊功能寄存器	178
5.3.11.1	SPI 控制寄存器 (SPI_CON)	178
5.3.11.2	SPI 发送数据写入寄存器 (SPI_TBW)	179
5.3.11.3	SPI 接收数据读取寄存器 (SPI_RBR)	180
5.3.11.4	SPI 中断使能寄存器 (SPI_IE)	180
5.3.11.5	SPI 中断标志寄存器 (SPI_IF)	181
5.3.11.6	SPI 发送缓冲寄存器 (SPI_TB)	182
5.3.11.7	SPI 接收缓冲寄存器 (SPI_RB)	183
5.3.11.8	SPI 状态寄存器 (SPI_STA)	183

5.3.11.9	SPI 波特率设置寄存器 (SPI_CKS)	184
5.3.12	SPI 应用说明	184
5.4	I2C 总线串口通讯控制器 (I2C0)	186
5.4.1	概述	186
5.4.2	结构框图	186
5.4.3	I2C 总线基本原理	186
5.4.3.1	I2C 通讯协议	186
5.4.3.2	I2C 数据传输格式	187
5.4.4	I2C 通讯端口配置	188
5.4.5	I2C 时基定时器与 16 倍速采样器	189
5.4.6	I2C 通讯发送器	190
5.4.7	I2C 通讯接收器	191
5.4.8	I2C 通讯控制	191
5.4.8.1	I2C 起始位	192
5.4.8.2	I2C 停止位	192
5.4.8.3	I2C 应答延迟功能	193
5.4.8.4	I2C 数据帧传输间隔功能	193
5.4.8.5	I2C 时钟线自动下拉等待请求功能	193
5.4.8.6	I2C 自动发送未应答功能	194
5.4.9	特殊功能寄存器	195
5.4.9.1	I2C 控制寄存器 (I2C_CON)	195
5.4.9.2	I2C 工作模式寄存器 (I2C_MOD)	196
5.4.9.3	I2C 中断使能寄存器 (I2C_IE)	198
5.4.9.4	I2C 中断标志寄存器 (I2C_IF)	199
5.4.9.5	I2C 发送数据写入寄存器 (I2C_TBW)	200
5.4.9.6	I2C 接收数据读取寄存器 (I2C_RBR)	201
5.4.9.7	I2C 发送缓冲寄存器 (I2C_TB)	201
5.4.9.8	I2C 接收缓冲寄存器 (I2C_RB)	201
5.4.9.9	I2C 状态寄存器 (I2C_STA)	202
5.4.10	I2C 应用说明	203
5.5	模数转换器 (ADC)	204
5.5.1	概述	204
5.5.2	结构框图	204
5.5.3	ADC 基本配置	204
5.5.4	ADC 高精度参考电压	204
5.5.5	ADC 数据转换	204
5.5.6	自动转换比较功能	207
5.5.7	特殊功能寄存器	208
5.5.7.1	ADC 参考控制寄存器 (ADC_VREFCON)	208
5.5.7.2	ADC 转换值寄存器 (ADC_DR)	209
5.5.7.3	ADC 控制寄存器 0 (ADC_CON0)	209
5.5.7.4	ADC 控制寄存器 1 (ADC_CON1)	210
5.5.7.5	ADC 通道选择寄存器 (ADC_CHS)	212
5.5.7.6	ADC 中断使能寄存器 (ADC_IE)	212

5.5.7.7	ADC 中断标志寄存器 (ADC_IF)	213
5.5.7.8	ADC 自动转换比较控制寄存器 (ADC_ACPC)	214
5.5.7.9	ADC 自动转换比较阈值寄存器 (ADC_ACPCMP)	215
5.5.7.10	ADC 自动转换均值数据寄存器 (ADC_ACPMEAN)	215
5.5.7.11	ADC 结果数据偏置寄存器 (ADC_OFFDR)	215
5.6	独立看门狗定时器 (IWDG)	216
5.6.1	概述	216
5.6.2	特殊功能寄存器	216
5.6.2.1	IWDG 计数器装载值寄存器 (IWDG_LOAD)	216
5.6.2.2	IWDG 计数器当前值寄存器 (IWDG_VALUE)	217
5.6.2.3	IWDG 控制寄存器 (IWDG_CON)	217
5.6.2.4	IWDG 中断标志清除寄存器 (IWDG_INTCLR)	218
5.6.2.5	IWDG 中断标志寄存器 (IWDG_RIS)	218
5.6.2.6	IWDG 访问使能寄存器 (IWDG_LOCK)	218
5.7	窗口看门狗 (WWDG)	220
5.7.1	概述	220
5.7.2	特殊功能寄存器	222
5.7.2.1	WWDG 计数器装载值寄存器 (WWDG_LOAD)	222
5.7.2.2	WWDG 计数器当前值寄存器 (WWDG_VALUE)	222
5.7.2.3	WWDG 控制寄存器 (WWDG_CON)	222
5.7.2.4	WWDG 中断标志清除寄存器 (WWDG_INTCLR)	223
5.7.2.5	WWDG 中断标志寄存器 (WWDG_RIS)	223
5.7.2.6	WWDG 锁定寄存器 (WWDG_LOCK)	224
第 6 章	芯片封装外观尺寸图	225
6.1	LQFP 32-pin 封装外观尺寸图	225
6.2	QFN 32-pin 封装外观尺寸图	226
附录 1	Cortex-M0 内核描述	227
附录 1.1	Cortex-M0 指令集	227
附录 1.2	Cortex-M0 内核寄存器	229
附录 1.2.1	通用寄存器 R0~R12	229
附录 1.2.2	堆栈指针寄存器 SP (R13)	229
附录 1.2.3	链接寄存器 LR (R14)	230
附录 1.2.4	程序计数器 PC (R15)	230
附录 1.2.5	程序状态寄存器 xPSR	230
附录 1.2.6	异常/中断屏蔽寄存器 PRIMASK	231
附录 1.2.7	控制寄存器 CONTROL	231
附录 2	电气特性	233
附录 2.1	参数特性表	233
附录 2.1.1	芯片工作条件	233
附录 2.1.2	芯片特性参数测量方法	234
附录 2.1.3	芯片功耗特性	234
附录 2.1.4	芯片 IO 端口特性	236
附录 2.1.5	芯片系统时钟特性	238
附录 2.1.6	芯片 ESD 特性	238

附录 2.1.7 芯片功能模块特性 .....	238
附录 2.2 参数特性图 .....	242
附录 2.2.1 芯片功耗特性 .....	242
附录 2.2.2 芯片 IO 端口输入特性 .....	244
附录 2.2.3 芯片 IO 端口输出特性（普通驱动） .....	245
附录 2.2.4 芯片 IO 端口输出特性（增强驱动，PB8~PB9 端口除外） .....	248
附录 2.2.5 芯片 IO 端口输出特性（增强驱动，PB8~PB9 端口） .....	251
<b>附录 3 编程调试接口 .....</b>	<b>253</b>
附录 3.1 概述 .....	253
附录 3.2 ISP 编程接口 .....	253
附录 3.2.1 通信协议 .....	253
附录 3.2.2 操作流程 .....	254
附录 3.3 SWD 调试接口 .....	255
附录 3.3.1 概述 .....	255
附录 3.3.2 SWD 特性 .....	255

## 图目录

图 1-1	FS030 结构框图 .....	21
图 1-2	LQFP32 封装顶视图 (FS030LK) .....	22
图 1-3	QFN32 封装顶视图(FS030NK) .....	23
图 2-1	系统电源结构框图 .....	27
图 2-2	系统复位电路结构框图 .....	28
图 2-3	上电复位时序示意图 .....	29
图 2-4	掉电复位时序示意图 .....	29
图 2-5	外部复位时序示意图 .....	29
图 2-6	MRSTN 复位参考电路图 1 .....	30
图 2-7	MRSTN 复位参考电路图 2 .....	30
图 2-8	MRSTN 复位参考电路图 3 .....	30
图 2-9	MRSTN 复位参考电路图 4 .....	31
图 2-10	LVD 低电压监测示意图 .....	33
图 2-11	系统时钟电路结构框图 .....	39
图 2-12	XTAL 振荡器电路结构示意图 .....	40
图 2-13	晶体振荡器效电路示意图 .....	40
图 2-14	系统定时器框图 .....	67
图 3-1	内部存储系统分配示意图 .....	72
图 3-2	SRAM 映射图 .....	87
图 3-3	外设存储器分配示意图 .....	88
图 4-1	IO 端口电路结构图 .....	97
图 4-2	外部端口中断 PINT0 电路结构示意图 .....	97
图 4-3	外部按键中断 KINT0 电路结构示意图 .....	98
图 4-4	Buzz 高电平调制输出波形图 .....	99
图 4-5	Buzz 低电平调制输出波形图 .....	99
图 5-1	T16N0 电路结构框图 .....	124
图 5-2	T16N0 计数匹配功能示意图 .....	126
图 5-3	T16N0 捕捉功能示意图 .....	127
图 5-4	T16N0 独立 PWM 模式输出调制功能示意图 .....	129
图 5-5	T16N0 互补 PWM 输出功能示意图 .....	130
图 5-6	带死区的 T16N0 互补 PWM 输出功能示意图 .....	131
图 5-7	T32N0 电路结构框图 .....	143
图 5-8	T32N0 计数匹配功能示意图 .....	145
图 5-9	T32N0 捕捉功能示意图 .....	146
图 5-10	T32N0 输出调制功能示意图 .....	147
图 5-11	UART 电路结构图 .....	155
图 5-12	UART 7 位数据格式 .....	156
图 5-13	UART 8 位数据格式 .....	156
图 5-14	UART 9 位数据格式 .....	156
图 5-15	UART0 发送数据流示意图 .....	157
图 5-16	UART0 发送数据操作流程示意图 .....	158
图 5-17	UART0 接收数据流示意图 .....	159
图 5-18	UART0 接收数据操作流程示意图 .....	160

图 5-19	TX0 高电平调制输出波形图 .....	160
图 5-20	TX0 低电平调制输出波形图 .....	161
图 5-21	自动波特率检测时序示意图 .....	162
图 5-22	自动波特率检测错误时序示意图 .....	162
图 5-23	空闲帧检测时序示意图 .....	163
图 5-24	SPI 电路结构框图 .....	173
图 5-25	SPI 时钟上升沿发送，下降沿接收波形示意图 .....	174
图 5-26	SPI 时钟下降沿发送，上升沿接收波形示意图 .....	174
图 5-27	SPI 时钟上升沿接收，下降沿发送波形示意图 .....	174
图 5-28	SPI 时钟下降沿接收，上升沿发送波形示意图 .....	175
图 5-29	SPI 发送数据流示意图 .....	175
图 5-30	SPI 接收数据流示意图 .....	176
图 5-31	SPI 延迟接收功能波形示意图 .....	178
图 5-32	I2C 电路结构框图 .....	186
图 5-33	I2C 总线通讯协议示意图 .....	187
图 5-34	I2C 主控器写入从动器数据示意图 .....	188
图 5-35	I2C 主控器读取从动器数据示意图 .....	188
图 5-36	开漏输出端口示意图 .....	189
图 5-37	I2C 总线端口信号的波形示意图 .....	190
图 5-38	I2C 发送数据流示意图 .....	190
图 5-39	I2C 接收数据流示意图 .....	191
图 5-40	I2C 起始位波形图 .....	192
图 5-41	I2C 自动寻呼波形图 .....	192
图 5-42	I2C 停止位波形图 .....	192
图 5-43	I2C 应答延迟功能波形示意图 .....	193
图 5-44	I2C 数据帧传输间隔功能波形示意图 .....	193
图 5-45	I2C 时钟线下拉等待波形示意图 .....	194
图 5-46	ADC 内部结构图 .....	204
图 5-47	ADC 数据转换时序示意图 (ADC_CON1 寄存器的 SMPS=0, 软件控制采样) .....	205
图 5-48	ADC 数据转换时序示意图 (ADC_CON1 寄存器的 SMPS=1, 硬件控制采样) .....	206
图 5-49	窗口看门狗中断和溢出复位产生时序图 (WWDTWIN 设定为 25%) .....	221
图 5-50	错误的喂狗时序图 (WWDTWIN 设定为 25%) .....	221



## 表目录

表 1-1	管脚说明.....	24
表 1-2	FS030 管脚对照表 .....	25
表 2-1	低功耗模式时钟状态表.....	35
表 2-2	异常/中断优先级操作类型说明说明表.....	51
表 2-3	异常/中断优先级列表 .....	52
表 2-4	IRQ 分配列表 .....	53
表 4-1	PINT 选择对应列表 .....	98
表 4-2	KINT 选择对应列表 .....	99
表 5-1	I2C 总线端口信号的时序参数列表 .....	189
表 5-2	ADC 精度与转换速率的对应关系列表 .....	205

## 第1章 芯片简介

### 1.1 概述

FS030 系列产品是一款高集成度的通用 MCU 芯片，内部集成 32 位 ARM Cortex-M0 CPU 内核。集成多个 16 位和 32 位定时器/计数器，UART 模块，SPI 和 I2C 通信模块，12 位 ADC，以及用于系统电源监测的 LVD 模块等外设。

#### ◆ 工作条件

- ◇ 工作电压范围：2.2V ~ 5.5V
- ◇ 工作温度范围：-40 ~ 85℃（工业级）
- ◇ 工作主时钟频率：32KHz~48MHz
- ◇ 工作电流：I<sub>vdd</sub> = 2.8mA（@内部 HRC 16MHz，5V，典型值）
- ◇ 待机电流：I<sub>vdd</sub> = 2.5uA（常温，5V，典型值）

#### ◆ 封装

- ◇ LQFP32/QFN32 封装（最多支持 30 个 I/O 端口）

#### ◆ 电源

- ◇ 系统电源输入 VDD，支持工作电压为 5V 或 3.3V 的应用系统
- ◇ 低功耗 LVD 用于监测系统电源掉电和上电，可选择产生掉电或上电中断

#### ◆ 复位

- ◇ 内嵌上电复位电路 POR
- ◇ 内嵌掉电复位电路 BOR
- ◇ 支持外部复位

#### ◆ 时钟

- ◇ 外部晶体振荡器可配置，支持低速振荡器 32KHz 和高速振荡器 1~20MHz，可配置为系统时钟源
- ◇ 内部 2/16/32/48MHz RC 振荡器（HRC）可配置为系统时钟源，出厂前已校准（常温频率精度为±1%）
- ◇ 内部 32KHz RC 振荡器（LRC）作为 IWDG 和 WWDG 时钟源，可配置为系统时钟源

#### ◆ 内核

- ◇ ARM Cortex-M0 32 位嵌入式处理器内核
- ◇ 支持 SWD 串行调试接口，支持 2 个监视点（watchpoint）和 4 个断点（breakpoint）
- ◇ 支持 1 组 SWD 调试接口
- ◇ 内嵌向量中断控制器 NVIC
- ◇ 支持唤醒中断控制器 WIC
- ◇ NVIC 包含一个不可屏蔽中断 NMI
- ◇ 内置 1 个 SysTick 系统定时器

- ◇ 支持单周期 32 位乘法器
- ◆ 独立看门狗 IWDG
  - ◇ 时钟源可选择
  - ◇ 支持低功耗模式下唤醒
  - ◇ 超时计数溢出可选择触发中断或复位
- ◆ 窗口看门狗 WWDG
  - ◇ 时钟源可选择，可用于检测软件的过早或过晚喂狗异常
  - ◇ 安全可靠，一旦使能，只能通过复位关断
  - ◇ 可设定喂狗窗口，喂狗窗口外喂狗将产生复位
- ◆ 存储器
  - ◇ 64K 字节 FLASH 存储器
    - 支持 ISP 在线串行编程
    - 支持 IAP 在应用中编程，可选取部分区域作为数据存储使用
    - 支持 FLASH 编程代码加密保护
  - ◇ 4K 字节 SRAM 存储器
    - SRAM 存储空间及外设寄存器地址空间支持位带（Bit band）扩展
- ◆ I/O 端口
  - ◇ 支持最多 30 个双向 I/O 端口
    - PA 端口（PA1~PA16, PA22~PA25, PA27, PA28）
    - PB 端口（PB0, PB1, PB8~PB13）
  - ◇ 支持 8 路外部中断输入，触发方式可配置，每个 I/O 端口均可作为外部中断输入源
  - ◇ 支持 1 路按键中断输入，触发方式可配置，每个 I/O 端口均可作为按键中断输入源
  - ◇ PB8、PB9 支持大电流口
- ◆ 定时器/计数器
  - ◇ T16N0: 16 位定时器/计数器，带预分频器，缓存机制，输入捕捉/输出调制功能
  - ◇ T16N1: 16 位定时器/计数器，带预分频器，缓存机制，输入捕捉/输出调制功能
  - ◇ T16N2: 16 位定时器/计数器，带预分频器，缓存机制，输入捕捉/输出调制功能
  - ◇ T16N3: 16 位定时器/计数器，带预分频器，缓存机制，输入捕捉/输出调制功能
  - ◇ T32N0: 32 位定时器/计数器，带预分频器，扩展输入捕捉/输出调制功能
- ◆ UART 通信接口
  - ◇ 支持三路 UART 通信接口 UART0, UART1, UART2
  - ◇ 支持全/半双工异步通信模式
  - ◇ 支持传输波特率可配置
  - ◇ 支持 7/8/9 位数据格式可配
  - ◇ 支持奇偶校验功能可配，支持硬件自动奇偶校验位判断
  - ◇ 支持空闲帧检测
  - ◇ 支持接收帧错误标志、溢出标志、奇偶校验错误标志

- ◇ 支持数据接收和发送中断
- ◇ 支持 PWM 调制输出，且 PWM 占空比线性可调
- ◇ 支持接收端口红外唤醒功能
- ◇ 支持 UART 输入输出通讯端口极性可配置

#### ◆ I2C 通信接口

- ◇ 支持一路通信接口 I2C0
- ◇ 支持主控和从动模式
- ◇ 支持标准 I2C 总线协议，最高传输速率 400K bit/s
- ◇ 支持 7 位寻址方式
- ◇ 约定数据从最高位开始接收/发送
- ◇ 支持数据接收和发送中断
- ◇ SCL/SDA 端口支持推挽/开漏模式，开漏时必须使能内部弱上拉或使用外部上拉电阻
- ◇ SCL 端口支持时钟线自动下拉等待请求功能

#### ◆ SPI 通信接口

- ◇ 支持一路通信接口 SPI1
- ◇ 支持主控模式和从动模式
- ◇ 支持 4 种通信数据格式
- ◇ 支持 4 级接收/发送缓冲器
- ◇ 支持数据接收和发送中断

#### ◆ ADC 模拟数字转换器

- ◇ 支持 12 位转换结果，有效精度为 11 位
- ◇ 支持 19 通道模拟输入端
- ◇ 支持参考电压源可选择，支持内部参考 2.048V。
- ◇ 支持中断产生
- ◇ 支持转换结果自动比较
- ◇ 支持定时触发 ADC 转换

## 1.2 应用领域

本芯片可用于家电，小家电以及工业控制仪表等领域。

### 1.3 结构框图

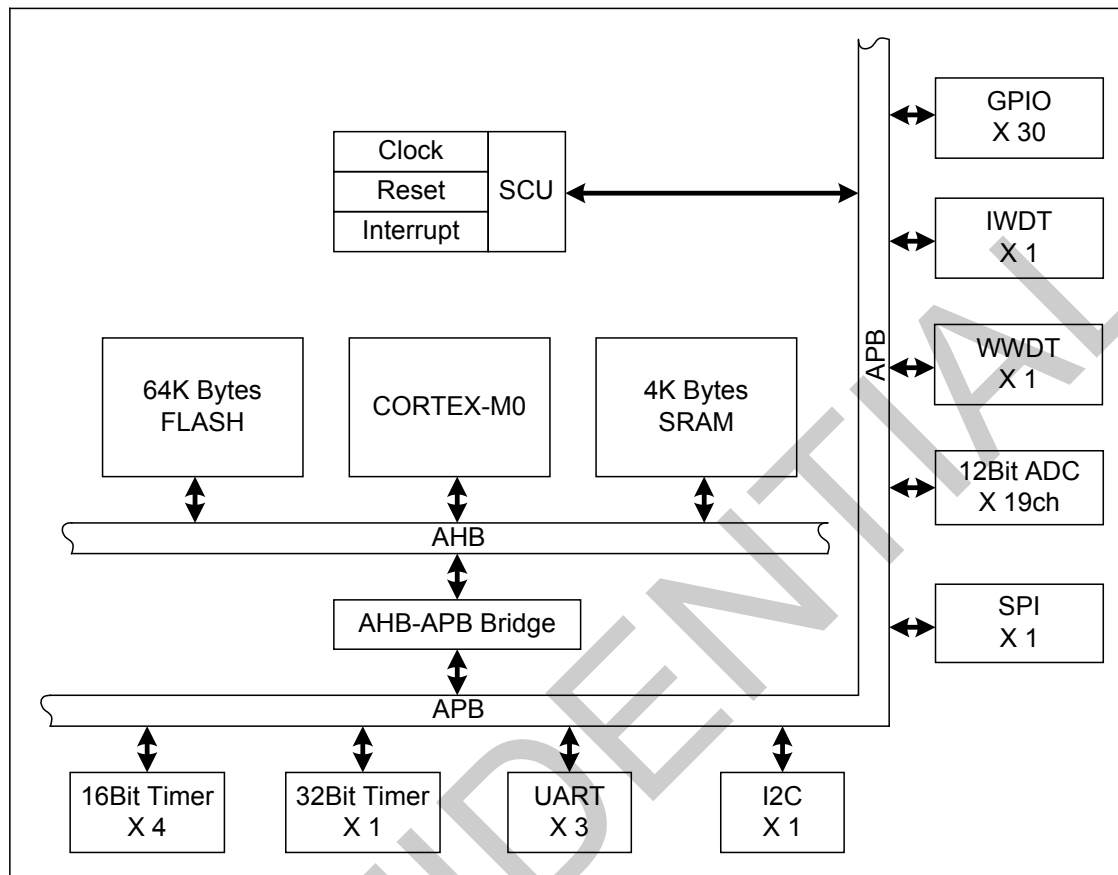


图 1-1 FS030 结构框图

1.4 管脚分配图

1.4.1 LQFP32 封装图

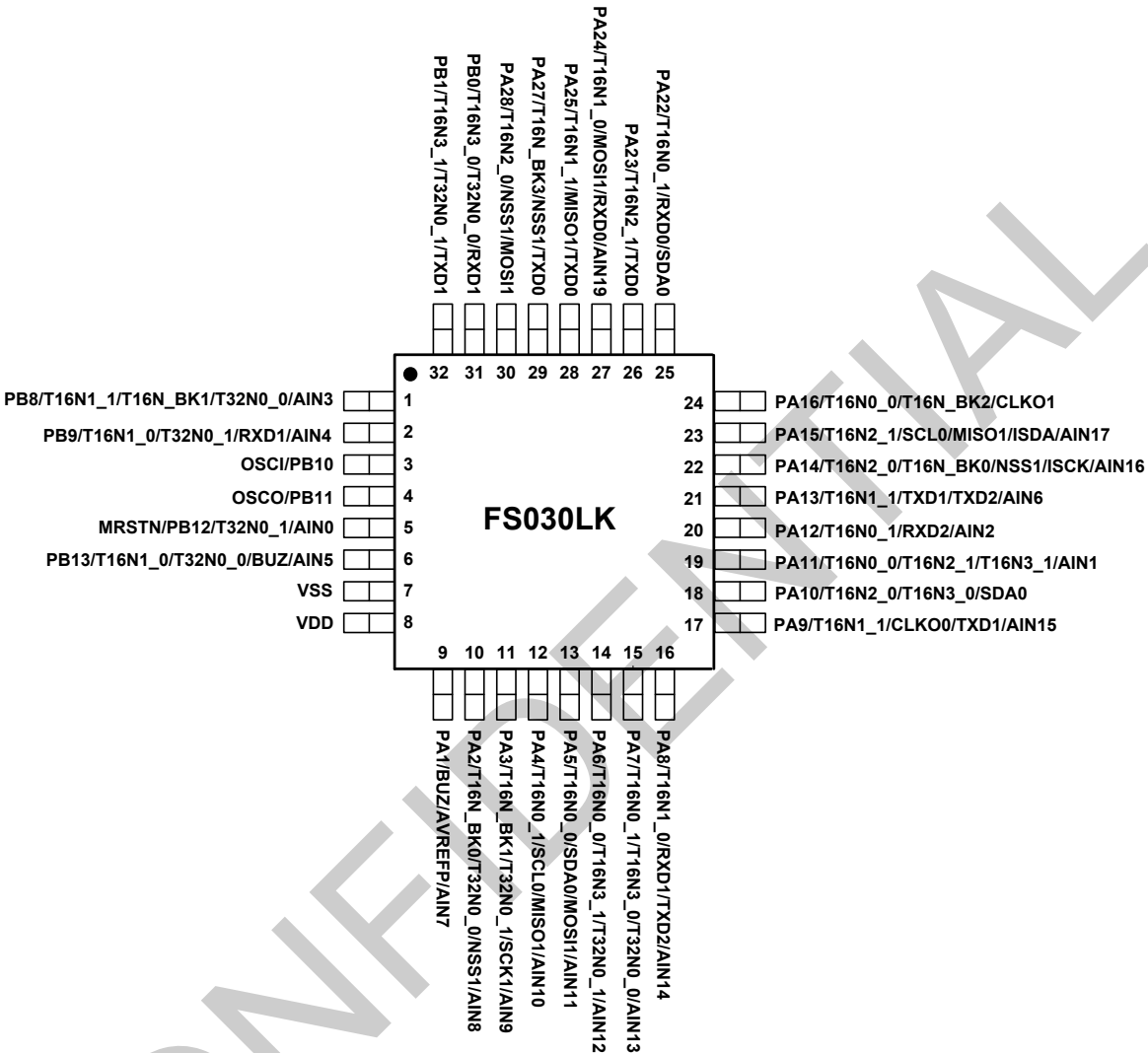


图 1-2 LQFP32 封装顶视图 (FS030LK)

### 1.4.2 QFN32 封装图

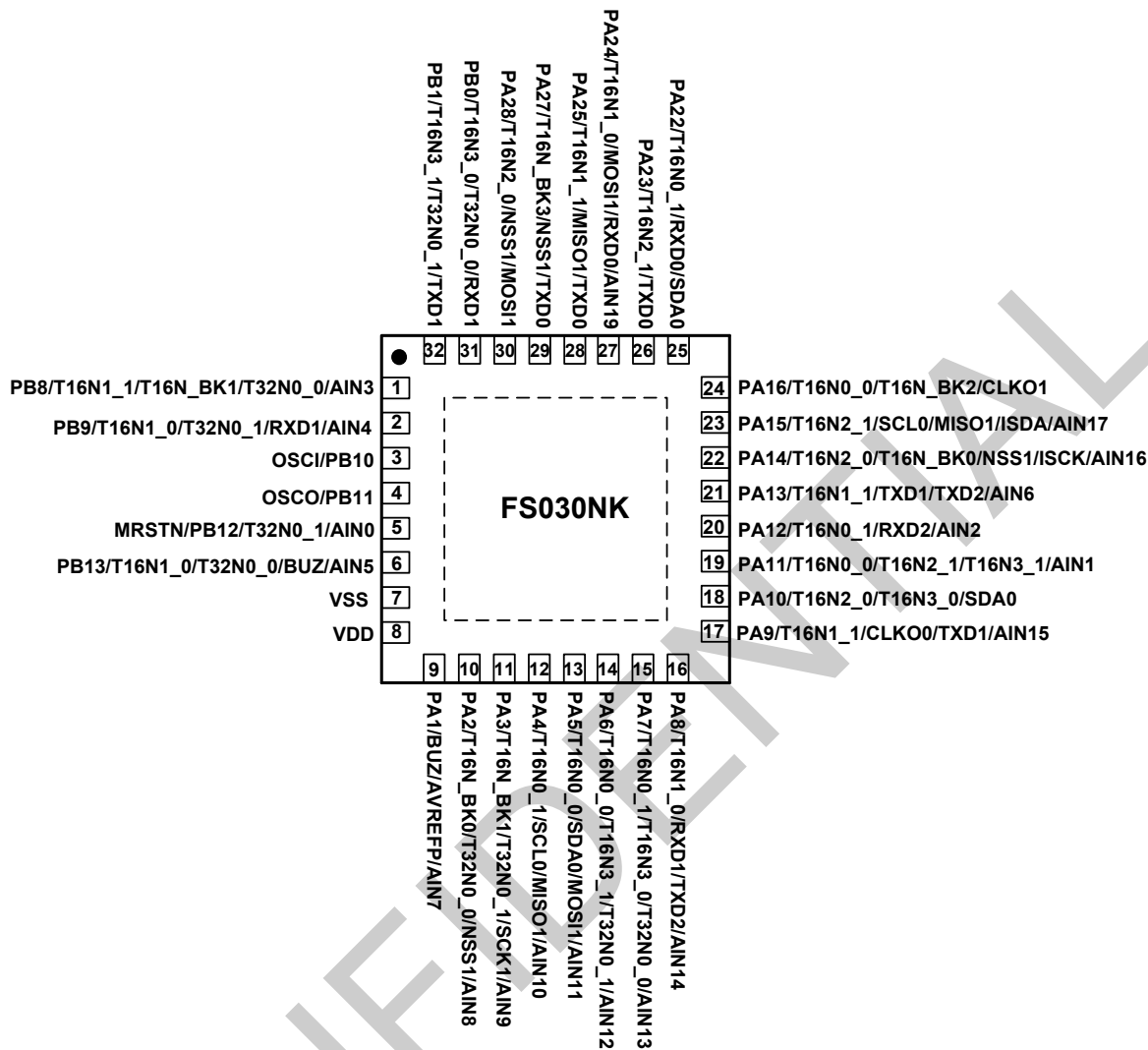


图 1-3 QFN32 封装顶视图(FS030NK)

注 1: QFN32 封装底部接地。如图虚线框内为接地 VSS。

注 2: 编程/调试接口共用 5 线接口配置, 即电源线 VDD、地线 VSS、复位线 MRSTN、时钟线 ISCK 和数据线 ISDA。

注 3: 在应用系统中, 未使用的 I/O 管脚, 以及由于产品封装引脚数小于最大封装管脚数而未引出的 I/O 管脚也需软件设置为输出低电平, 否则芯片功耗可能会出现异常, 芯片工作稳定性也容易因外界干扰而降低。

FS030LK 和 FS030NK 为最大引脚数封装, 所有 I/O 管脚均已引出。

注 4: PB12 端口在芯片 VDD 上电过程中, 内部弱上拉自动使能, 端口电平跟随 VDD 上升, 直到 VDD 上升到芯片开始工作的电压后, 该 IO 端口的内部弱上拉恢复为默认的禁止状态。

## 1.5 管脚说明

### 1.5.1 管脚说明

管脚名称	输入类型	输出类型	A/D	管脚说明
PA1~PA16 PA22~PA25,	CMOS	CMOS	D	通用 I/O 端口

管脚名称	输入类型	输出类型	A/D	管脚说明
PA27, PA28				
PB0, PB1, PB8~PB13	CMOS	CMOS	D	通用 I/O 端口
ISCK	CMOS	—	D	编程/调试串行时钟端口
ISDA	CMOS	CMOS	D	编程/调试串行数据端口
AIN0~AIN17, AIN19	—	—	A	ADC 模拟通道 0~17, 19
TXD0~TXD2	—	CMOS	D	UART0~UART2 发送输出端口
RXD0~RXD2	CMOS	—	D	UART0~UART2 接收输入端口
SCK1	CMOS	CMOS	D	SPI1 时钟输入/输出端口
NSS1	CMOS	—	D	SPI1 片选端口
MISO1	CMOS	CMOS	D	SPI1 主控输入/从机输出端口
MOSI1	CMOS	CMOS	D	SPI1 主控输出/从机输入端口
SCL0	CMOS	CMOS	D	I2C 时钟输入/输出端口
SDA0	CMOS	CMOS	D	I2C 数据输入/输出端口
T16N0_0, T16N0_1 T16N1_0, T16N1_1 T16N2_0, T16N2_1 T16N3_0, T16N3_1	CMOS	CMOS	D	T16N0/T16N1/T16N2/T16N3 外部时钟输入/捕捉输入/调制输出端口
T32N0_0, T32N0_1	CMOS	CMOS	D	T32N0 外部时钟输入/捕捉输入/调制输出端口
T16N_BK0 T16N_BK1 T16N_BK2 T16N_BK3	CMOS	—	D	T16N0/T16N1/T16N2/T16N3 刹车输入端口（只能设置一个有效）
AVREFP	—	—	A	ADC 外部正向参考电压
MRSTN	CMOS	—	D	芯片主复位，低电平有效
OSCI	—	—	A	外部晶体振荡器端口
OSCO	—	—	A	
VDD	—	—	P	系统主电源
VSS	—	—	P	系统地

表 1-1 管脚说明

注 1: A = 模拟端口, D = 数字端口, P = 电源/地;

注 2: 表中 T16N0\_0 表示 T16N0 定时器/计数器的 T16N0CK0/T16N0IN0/T16N0OUT0 三个复用输入或输出。下文若未特别说明, T16N0\_1/T16N1\_0/T16N1\_1/T16N2\_0/T16N2\_1/T16N3\_0/T16N3\_1 和 T32N0\_0/T32N0\_1 都指其对应的三个复用输入或输出信号。



## 1.5.2 管脚对照表

PIN NAME (FUN0(D))	FUN1(D)	FUN2(D)	FUN3(D)	FUN4(A)
PA1	BUZ	—	—	AVREFP/AIN7
PA2	T16N_BK0	T32N0_0	NSS1	AIN8
PA3	T16N_BK1	T32N0_1	SCK1	AIN9
PA4	SCL0	T16N0_1	MISO1	AIN10
PA5	SDA0	T16N0_0	MOSI1	AIN11
PA6	T32N0_1	T16N0_0	T16N3_1	AIN12
PA7	T32N0_0	T16N0_1	T16N3_0	AIN13
PA8	RXD1	T16N1_0	TXD2	AIN14
PA9	TXD1	T16N1_1	CLKO0	AIN15
PA10	T16N2_0	T16N3_0	SDA0	—
PA11	T16N2_1	T16N3_1	T16N0_0	AIN1
PA12	—	RXD2	T16N0_1	AIN2
PA13	TXD1	TXD2	T16N1_1	AIN6
PA14(ISCK)	T16N_BK0	T16N2_0	NSS1	AIN16
PA15(ISDA)	SCL0	T16N2_1	MISO1	AIN17
PA16	T16N0_0	T16N_BK2	CLKO1	—
PA22	RXD0	T16N0_1	SDA0	—
PA23	TXD0	T16N2_1	—	—
PA24	MOSI1	RXD0	T16N1_0	AIN19
PA25	MISO1	TXD0	T16N1_1	—
PA27	NSS1	T16N_BK3	TXD0	—
PA28	MOSI1	NSS1	T16N2_0	—
PB0	T32N0_0	RXD1	T16N3_0	—
PB1	T32N0_1	TXD1	T16N3_1	—
PB8	T16N_BK1	T16N1_1	T32N0_0	AIN3
PB9	RXD1	T16N1_0	T32N0_1	AIN4
PB10	—	—	—	OSCI
PB11	—	—	—	OSCO
PB12(MRSTN)	—	—	T32N0_1	AIN0
PB13	T16N1_0	BUZ	T32N0_0	AIN5

表 1-2 FS030 管脚对照表

注 1: FUN0(D)/FUN1(D)/FUN2(D)/FUN3(D)表示数字端口; FUN4(A)表示模拟端口。

注 2: 表中的 FUN4(A)为模拟端口功能, 不通过 GPIO\_PAFUNC/GPIO\_PBFUNC 控制寄存器选取。

注 3: 支持一组编程/调试接口 ISCK (PA14), ISDA (PA15)。

注 4: 对通信功能模块 UART, SPI 和 I2C, 每个通信管脚支持多个 IO 端口可复用, 并可独立设置, 例如 PB0 端口复用为 RXD1 时, 可以设置 PA9, PA13 和 PB1 端口中的任意一个复用为 TXD1, 作为 UART1 模块的通信管脚。

注 5: 当多个 IO 端口复用为 T16N0\_0 时, 由硬件电路自动选择其中一个作为 T16N0 定时器/计数器的输入端口 T16N0CK0/T16N0IN0, 优先级由高到低依次为 PA16, PA5, PA6, PA11 (例如当 PA16, PA5, PA6 和 PA11 都被复用为 T16N0\_0 时, 则只有 PA16 会被作为输入功能 T16N0CK0/T16N0IN0; 而 PA16, PA5, PA6 和 PA11 这四个端口均可被作为 T16N0OUT0, 通过软件设置对应的端口方向控制位 DIR 为输出; 为避免端口电平冲突, 如果应用中使用了 T16N0\_0 的输入端口功能, 则需避免将该端口的方向控制位 DIR 设置为输出)。

同样:

当 T16N0\_1 作为输入端口 T16N0CK1/T16N0IN1 时, 优先级由高到低依次为 PA4, PA7, PA22, PA12;

当 T16N1\_0 作为输入端口 T16N1CK0/T16N1IN0 时, 优先级由高到低依次为 PB13, PA8, PB9, PA24;

当 T16N1\_1 作为输入端口 T16N1CK1/T16N1IN1 时, 优先级由高到低依次为 PA9, PB8, PA25, PA13;

当 T16N2\_0 作为输入端口 T16N2CK0/T16N2IN0 时, 优先级由高到低依次为 PA10, PA14, PA28;

当 T16N2\_1 作为输入端口 T16N2CK1/T16N2IN1 时, 优先级由高到低依次为 PA11, PA15, PA23;

当 T16N3\_0 作为输入端口 T16N3CK0/T16N3IN0 时, 优先级由高到低依次为 PA10, PA7, PB0;

当 T16N3\_1 作为输入端口 T16N3CK1/T16N3IN1 时, 优先级由高到低依次为 PA11, PA6, PB1;

当 T32N0\_0 作为输入端口 T32N0CK0/T32N0IN0 时, 优先级由高到低依次为 PA7, PB0, PA2, PB8, PB13;

当 T32N0\_1 作为输入端口 T32N0CK1/T32N0IN1 时, 优先级由高到低依次为 PA6, PB1, PA3, PB9, PB12。

## 第2章 系统控制及操作特性

### 2.1 系统控制保护

#### 2.1.1 概述

由于系统控制寄存器的访问操作会影响整个芯片的运行状态，为避免误操作导致芯片运行不正常，芯片提供系统设置保护寄存器。修改系统控制单元前，必须先关闭写保护，操作完成后应当重新使能写保护，使芯片安全运行。

寄存器列表和基址参见章节：系统控制单元（SCU）寄存器列表。

#### 2.1.2 特殊功能寄存器

##### 2.1.2.1 系统设置保护寄存器（SCU\_PROT）

系统设置保护寄存器（SCU_PROT）																																
偏移地址：00 <sub>H</sub>																																
复位值：00000000_00000000_00000000_00000001 <sub>B</sub>																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	保留															PROT	
—		bit31-1				W		对 SCU_PROT<31:0>写 0x55AA6996 时，位 PROT 为 0； 写其它值时位 PROT 为 1																								
PROT		bit0				R/W		<b>SCU 写保护位</b> 0：写保护关闭 1：写保护使能																								

注 1：只有以字方式对 SCU\_PROT 寄存器写入 0x55AA6996 才能关闭写保护，其他任何对 SCU\_PROT 寄存器的写操作都将使能写保护功能。

注 2：SCU\_PROT 保护的寄存器为 SCU\_NMICON，SCU\_PWRC，SCU\_FAULTFLAG，SCU\_FLASHWAIT，SCU\_LVDCON，SCU\_CCM，SCU\_TIMEREN，SCU\_TIMERDIS，SCU\_SCLKEN0，SCU\_SCLKEN1，SCU\_PCLKEN，SCU\_WAKEUPTIME，SCU\_TBLREMAPEN。

### 2.2 系统电源

#### 2.2.1 结构框图

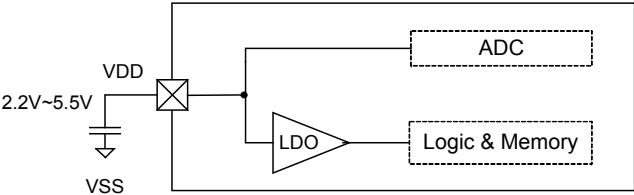


图 2-1 系统电源结构框图

### 2.2.2 芯片供电电源

芯片供电电源为 VDD，与其对应的是芯片的参考地 VSS。

VDD 给 GPIO 端口、ADC 供电，内部 LDO 输出电压给数字逻辑、Flash、SRAM 等供电。

## 2.3 系统复位

### 2.3.1 概述

- ◇ 支持 POR 上电复位
- ◇ 支持 BOR 低电压监测复位
- ◇ 支持 MRSTN 外部端口复位
- ◇ 支持 IWDT 和 WWDT 看门狗复位
- ◇ 支持 Cortex-M0 软件复位
- ◇ 支持 LOCKUP 复位

当上述 POR，BOR 或 MRSTN 复位发生时，芯片内部各模块的特殊功能寄存器均会被复位；当发生 IWDT，WWDT，软件复位或 LOCKUP 复位时，系统时钟控制寄存器 SCU\_SCLKEN0 和 SCU\_SCLKEN1，系统唤醒时间控制寄存器 SCU\_WAKEUPTIME 不受影响，其他寄存器会被复位。

### 2.3.2 结构框图

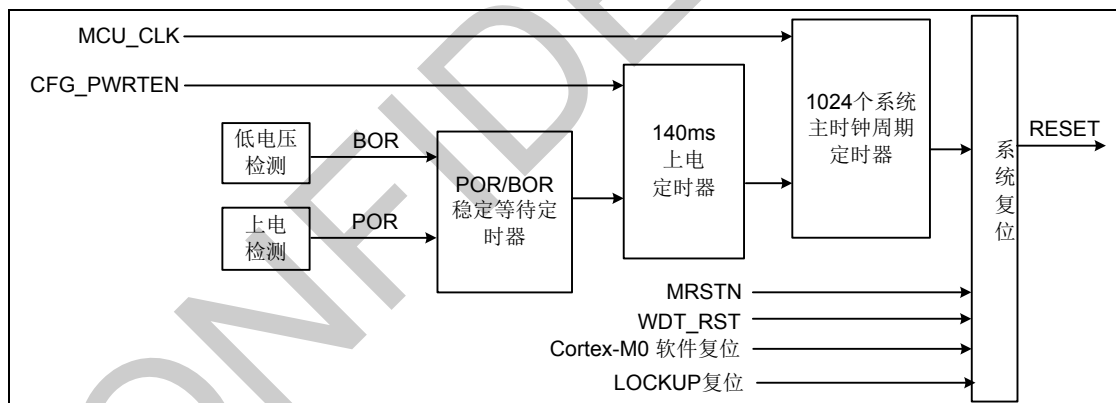


图 2-2 系统复位电路结构框图

注 1: 对 140ms 上电延时定时器，在 MRSTN 管脚复用为 GPIO 功能时，该延时固定为使能，与配置位 CFG\_PWRTEN 无关。

注 2: 芯片上电稳定后，在工作过程中，如果发生外部复位，IWDT 和 WWDT 看门狗复位 WDT\_RST，或软件复位，则在复位条件撤除后，芯片会立即退出复位状态，恢复正常运行状态，与上图中的各定时器无关。

### 2.3.3 复位时序图

以下分别对芯片上电复位 POR，掉电复位 BOR，外部复位时序进行说明，时序图中的 RESET 信号，为芯片内部产生的复位信号，高电平有效。

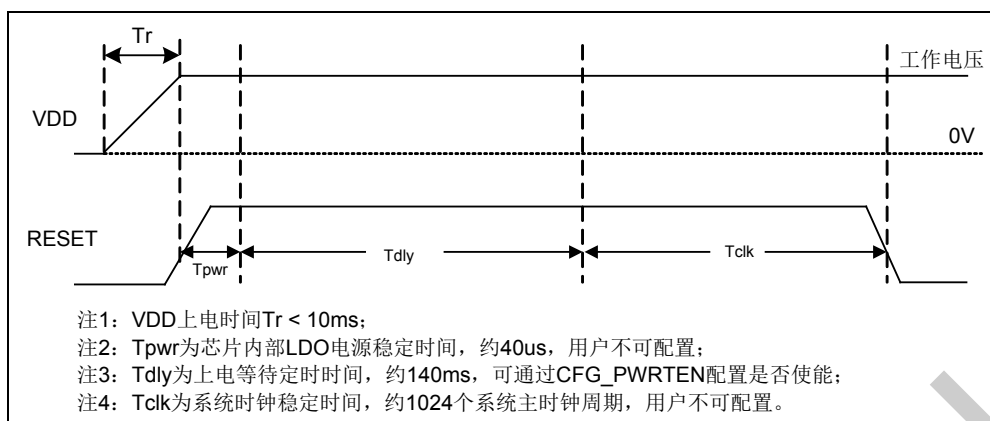


图 2-3 上电复位时序示意图

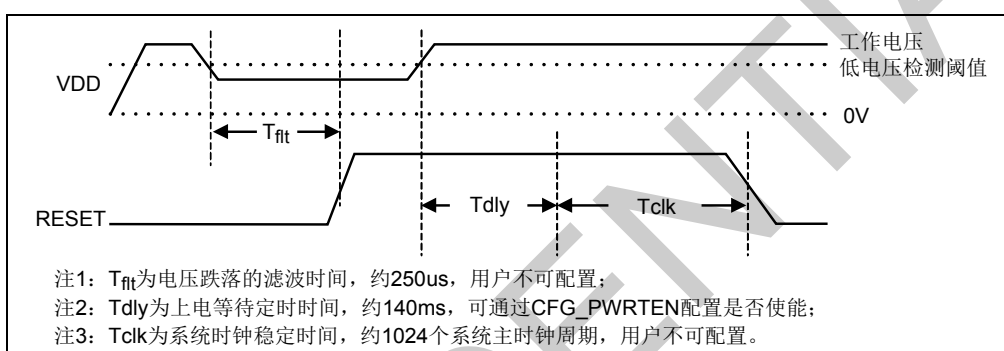


图 2-4 掉电复位时序示意图

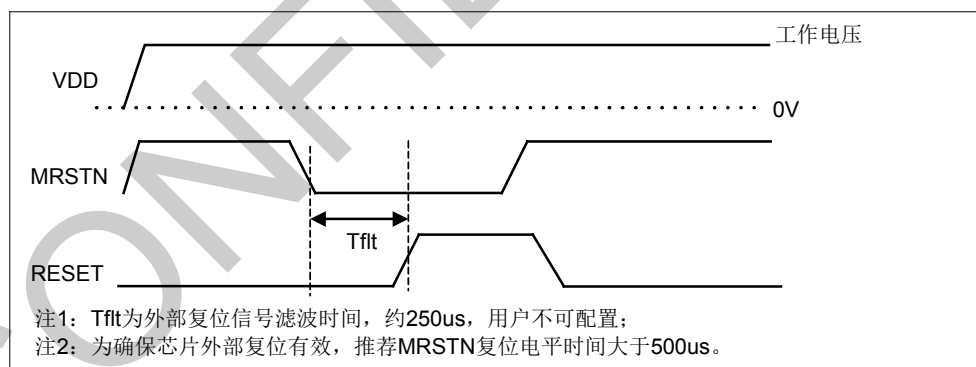


图 2-5 外部复位时序示意图

### 2.3.4 外部复位MRSTN参考

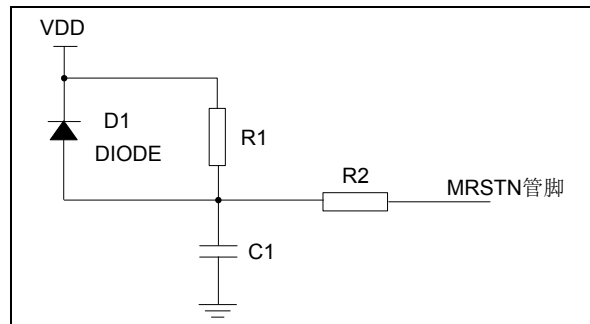


图 2-6 MRSTN 复位参考电路图 1

注 1: 采用 RC 复位, 其中  $47\text{K}\Omega \leq R1 \leq 100\text{K}\Omega$ , 电容  $C1 = (0.1\mu\text{F})$ ,  $R2$  为限流电阻,  $0.1\text{K}\Omega \leq R2 \leq 1\text{K}\Omega$ 。

注 2: 当 MRSTN 用作外部复位管脚时, 芯片内部固定集成了约 53K 欧姆的上拉电阻。

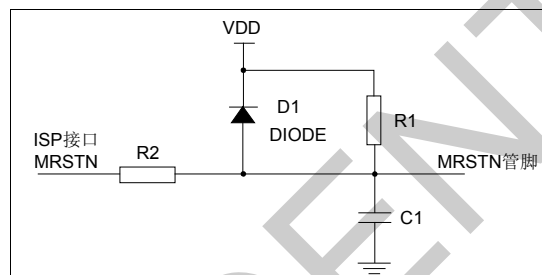


图 2-7 MRSTN 复位参考电路图 2

注 1: 采用 RC 复位, 其中  $47\text{K}\Omega \leq R1 \leq 100\text{K}\Omega$ , 电容  $C1 = (0.1\mu\text{F})$ ,  $R2$  为限流电阻,  $0.1\text{K}\Omega \leq R2 \leq 1\text{K}\Omega$ 。

注 2: 如果应用系统中的 ISP 编程接口与芯片管脚之间连线较长, 且应用环境中存在较强干扰, 则推荐按上图方式在 ISP 接口 MRSTN 和芯片的 MRSTN 管脚之间添加电阻  $R2$ 。

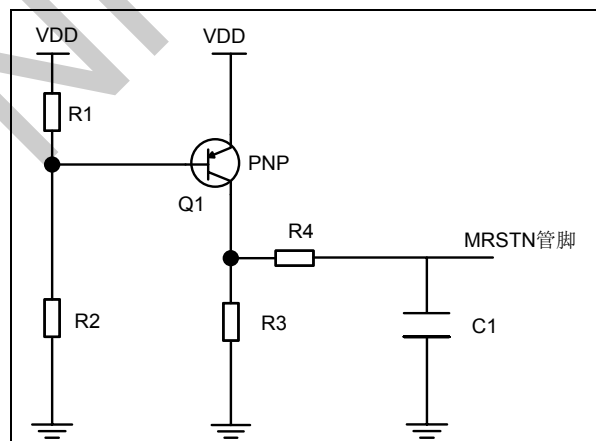


图 2-8 MRSTN 复位参考电路图 3

注: 采用 PNP 三极管复位, 通过  $R1$  ( $2\text{K}\Omega$ ) 和  $R2$  ( $10\text{K}\Omega$ ) 分压作为基极输入, 发射极接 VDD, 集电极一路通过  $R3$  ( $20\text{K}\Omega$ ) 接地, 另一路通过  $R4$  ( $1\text{K}\Omega$ ) 和  $C1$  ( $0.1\mu\text{F}$ ) 接地,  $C1$  另一端作为 MRSTN 输入。

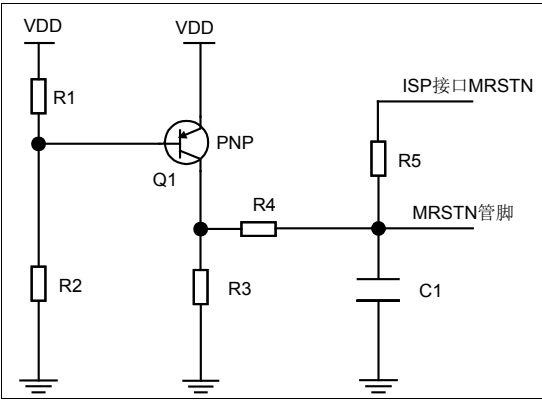


图 2-9 MRSTN 复位参考电路图 4

注 1：采用 PNP 三极管复位，通过 R1（2KΩ）和 R2（10KΩ）分压作为基极输入，发射极接 VDD，集电极一路通过 R3（20KΩ）接地，另一路通过 R4 和 C1（0.1μF）接地，C1 另一端作为 MRSTN 输入。

注 2：如果应用系统中的 ISP 编程接口与芯片管脚之间连线较长，且应用环境中存在较强干扰，则推荐按上图方式添加电阻 R5， $0.1K\Omega \leq R5 \leq 1K\Omega$ ，同时选择电阻 R4 的阻值为  $47K\Omega \leq R4 \leq 100K\Omega$ 。

### 2.3.5 特殊功能寄存器

#### 2.3.5.1 复位寄存器（SCU\_PWRC）

复位寄存器（SCU\_PWRC）

偏移地址：08H

复位值：00000000\_00000000\_00000000\_xxxxxxxx B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						LKR	CFG_	POR_L	SOFT_	MR	WDTR	BOR	PORRST	POR	POR
						STF	RST	OST	RSTF	STF	STF	F	F	RCF	F

—	bit31-10	—	—
LKRSTF	bit9	R/W	<b>LOCKUP 复位标志位</b> 0：无 LOCKUP 复位发生 1：LOCKUP 复位发生
CFG_RST	bit8	R/W	<b>配置字读取标志位</b> （内部测试用，用户无需关心此位） 0：无读取配置字发生 1：读取配置字发生
POR_LOST	bit7	R/W	<b>POR 丢失标志位</b> （内部测试用，用户无需关心此位） 0：无 POR 丢失 1：有 POR 丢失
SOFT_RSTF	bit6	R/W	<b>软件复位标志位</b> 0：无软件复位

			1: 有软件复位
MRSTF	bit5	R/W	<b>MRSTN 复位标志位</b> 0: 无 MRSTN 复位 1: 有 MRSTN 复位
WDTRSTF	bit4	R/W	<b>WDT 复位标志位</b> 0: 无 WDT 复位 1: 有 WDT 复位
BORF	bit3	R/W	<b>BOR 掉电复位标志位</b> 0: 无 BOR 复位 1: 有 BOR 复位
PORRSTF	bit2	R/W	<b>PORRST 上电复位标志位</b> （内部测试用，使用 注意事项参见注 2） 0: 无 PORRST 复位 1: 有 PORRST 复位
PORRCF	bit1	R/W	<b>PORRC 复位标志位</b> 0: 无 PORRC 复位 1: 有 PORRC 复位
PORF	bit0	R/W	<b>POR 复位标志位</b> （内部测试用，用户无需关心 此位） 0: 无 POR 复位 1: 有 POR 复位

注 1: 电源供电异常时，PORRCF 标志可能会失效。

注 2: 上电后用户必须先对 PORRSTF 作清零操作，否则即使发生 bit3~bit6, bit9 对应的复位事件，也无法置起 bit3~bit6, bit9 标志位。

注 3: SCU\_PWRC 寄存器中的各标志位在发生对应复位事件时由硬件置 1, 并只能由软件写 0 清零, 对 SCU\_PWRC 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。

注 4: 当发生 IWDT 或 WWDT 复位时, 均会置起 WDTRSTF 复位标志。

## 2.4 低电压监测（LVD）

### 2.4.1 概述

LVD 可用于监视 VDD 电源, 通过设置 LVDCON.EN 使能 LVD, 将 VDD 电压和 LVDCON.VS 所选择的电压阈值进行比较, 可粗略判断当前电源 VDD 的电压值。

LVD 提供了一个状态标志位 LVDO, 用于指示 VDD 是大于还是小于 LVD 电压阈值。通过使能 LVDCON.IE 可使能 LVD 中断, 通过设置 LVDCON.IFS 可选择 LVD 中断类型（针对 LVDO 信号的变化）。当 VDD 降至 LVD 电压阈值以下, 或者当 VDD 升至 LVD 电压阈值以上时, 可以产生 LVD 中断, 具体取决于 LVDCON.IFS 的中断类型配置。该功能的作用之一就是可以在 VDD 发生跌落时, 立即进入中断服务程序执行紧急关闭系统的任务。



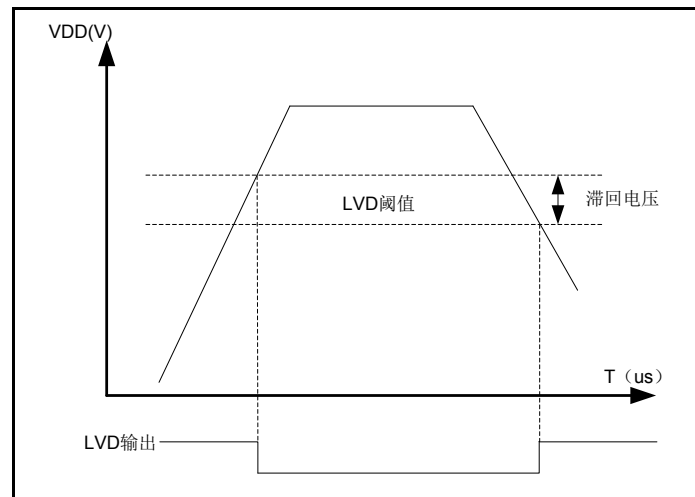


图 2-10 LVD 低电压监测示意图

## 2.4.2 特殊功能寄存器

### 2.4.2.1 低电压监测控制寄存器 (SCU\_LVDCON)

低电压监测控制寄存器 (SCU\_LVDCON)

偏移地址: 28<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDO	保留	IFS<2:0>			IE	IF	VS<3:0>			保留	FLTEN	EN			

—	bit31-16	—	—
LVDO	bit15	R	<b>LVD 输出状态位</b> 0: 被监测电压高于电压阈值 1: 被监测电压低于电压阈值
—	bit14-13	—	—
IFS<2:0>	bit12-10	R/W	<b>LVD 中断标志产生模式选择位</b> 000: LVDO 上升沿产生中断 001: LVDO 下降沿产生中断 010: LVDO 高电平产生中断 011: LVDO 低电平产生中断 1xx: LVDO 变化 (上升或下降沿) 产生中断
IE	bit9	R/W	<b>LVD 中断使能位</b> 0: 禁止 1: 使能
IF	bit8	R/W	<b>LVD 中断标志位</b> 0: 未发生 LVD 触发事件 1: 发生 LVD 触发事件 边沿模式产生中断标志时, 可以写 1 清除标志;

			电平模式产生中断标志时，该标志只读，触发电平消失后，中断标志自动清零。
VS<3:0>	bit7-4	R/W	<b>LVD 触发电压</b> 0000: 2.2V 0001: 2.4V 0010: 2.6V 0011: 2.8V 0100: 3.0V 0101: 3.6V 0110: 4.1V 0111: 4.7V 1xxx: 保留
—	bit3-2	—	—
FLTEN	bit1	R/W	<b>LVD 滤波使能位</b> 0: 禁止 1: 使能
EN	bit0	R/W	<b>LVD 使能位</b> 0: 禁止 1: 使能

注 1：对 SCU\_LVDCON 寄存器进行写操作前，需要设置 SCU\_PROT 寄存器，关闭写保护。

注 2：对 LVD 滤波使能位 FLTEN，需根据实际芯片的工作电源和环境，及应用系统的具体要求进行设置，当 LVD 滤波使能时，会滤除短暂的电源电压抖动，但也会降低 LVD 电路对电源波动的反应敏感度。

## 2.5 系统低功耗操作模式

### 2.5.1 概述

配置外设时钟控制寄存器 `SCU_PCLKEN`, 可分别关闭芯片各个外设功能模块电路的时钟, 使该部分电路功耗降到最低。

通过 `WFI` 指令, 可使芯片进入休眠状态, 配置 `SCB_SCR` 寄存器的 `SLEEPDEEP` 位, 可选择休眠状态为浅睡眠模式或深度睡眠模式。

芯片进入休眠状态后, 所有 I/O 端口将保持进入休眠前的状态。为了降低功耗, 所有 I/O 端口都应保持为高电平或低电平, 同时避免输入端口悬空而产生漏电流, 可通过弱上拉或弱下拉将悬空的输入端口固定为高电平或低电平。

芯片进入休眠状态后, 时钟工作状态参考下表:

时钟源	浅睡眠模式	深度睡眠模式
XTAL	工作 (若 <code>XTAL_EN=1</code> )	工作 (若 <code>XTAL_EN=1</code> 且 <code>MOSC_EN=1</code> )
HRC	工作 (若 <code>HRC_EN=1</code> )	工作 (若 <code>HRC_EN=1</code> 且 <code>MOSC_EN=1</code> )
LRC	工作	工作

表 2-1 低功耗模式时钟状态表

### 2.5.2 浅睡眠模式

在浅睡眠模式下, 芯片内核时钟停止, 指令停止运行。可通过复位或中断唤醒浅睡眠模式。

芯片进入浅睡眠模式的步骤如下:

- 1) 配置休眠状态选择位 `SLEEPDEEP=0`;
- 2) 运行等待中断 (`WFI`) 指令, 进入浅睡眠模式。

在浅睡眠模式下外设功能模块继续运行, 并可能产生中断使内核处理器恢复运行。浅睡眠模式下不访问存储器系统, 相关控制器和内部总线。

在浅睡眠模式下, 内核处理器的状态和寄存器, 外设寄存器和内部 `SRAM` 的值都会保持, 端口的逻辑电平也会保持睡眠前的状态。

### 2.5.3 深度睡眠模式

在深度睡眠模式下, 芯片内核时钟停止, 指令停止运行。可通过复位或中断唤醒深度睡眠模式。

芯片进入深度睡眠模式的步骤如下:

- 1) 配置休眠状态选择位 `SLEEPDEEP=1`;
- 2) 运行等待中断 (`WFI`) 指令, 进入深度睡眠模式。

在深度睡眠模式下, 外设时钟 `PCLK` 停止, 使用 `PCLK` 或其分频时钟作为时钟源的外设功能模块都停止工作, 其它使用内部低速时钟 `LRC` 或外部时钟 `XTAL` 作为时钟源的外设功能模块可正常工作。深度睡眠模式下不访问存储器系统, 相关控制器和内部总线。

在深度睡眠模式下，内核处理器的状态和寄存器，外设寄存器和内部 SRAM 的值都会保持，端口的逻辑电平也会保持深度睡眠前的状态。

在进入深度睡眠模式前，通过系统唤醒时间控制寄存器（SCU\_WAKEUPTIME）的深度睡眠模式时钟控制位（MOSC\_EN）来选择主晶振 XTAL、HRC 和时钟滤波器 CLKFLT 等时钟模块是否关闭。选择时钟模块关闭时（即 MOSC\_EN=0），可降低深度睡眠模式下系统的功耗，但同时也增大了唤醒时所需要的时间。

注：进入睡眠模式前需先清除所有中断挂起标志位，否则任意已经置起的中断挂起标志位，都会导致芯片无法进入睡眠模式，在清除中断挂起标志位的指令和进睡眠模式的指令之间，需延时至少一个 NOP 指令周期，确保清除标志位操作执行完毕。

#### 2.5.4 睡眠模式的唤醒

芯片可通过以下事件从睡眠状态唤醒，并执行下一条指令或进入中断处理程序。如果是中断唤醒且该中断已使能，则唤醒后立即进入中断处理程序。

- ◇ 浅睡眠模式唤醒：
  - 所有中断均可以唤醒浅睡眠模式
  - 芯片复位唤醒浅睡眠模式
- ◇ 深度睡眠模式唤醒：
  - 外部端口中断 PINT 可以唤醒深度睡眠模式
  - 外部端口中断 KINT 可以唤醒深度睡眠模式
  - LVD 中断可以唤醒深度睡眠模式
  - WDT 中断可以唤醒深度睡眠模式（工作于 LRC 时钟源）
  - ADC 中断可以唤醒深度睡眠模式（工作于 LRC 时钟源）
  - 芯片复位唤醒深度睡眠模式

#### 2.5.5 睡眠模式的唤醒时间

芯片深度睡眠模式的唤醒时间，包括系统时钟稳定时间和内部 LDO 电压稳定时间，具体的唤醒时间，与系统时钟源和深度睡眠模式下是否使能系统时钟有关。

内部 HRC 时钟的起振稳定时间约为 10us，外部时钟 XTAL 16MHz 振荡器的起振稳定时间约为 5ms，外部 XTAL 32KHz 的起振稳定时间约为 1.2 秒。内部 LDO 电压稳定时间约为 15us。

内部 HRC 时钟的稳定时间可软件设置：Tpclk\*WAKEUPTIME（其中 Tpclk 为系统时钟周期，WAKEUPTIME 为唤醒时间控制位 WAKEUPTIME<11:0>），推荐 HRC 时钟的稳定时间需设置为大于 10us，否则芯片唤醒后有可能工作异常。

浅睡眠模式无唤醒时间，与 MOSC\_EN 和 WAKEUPTIME 的设置无关，有唤醒事件时，芯片立即被唤醒并开始执行程序。

#### 2.5.6 FLASH 存储器等待功能

FLASH 存储器的访问频率对芯片功耗影响较大，降低其访问频率，可降低芯片功耗。可以通过降低系统时钟频率来降低 FLASH 存储器的访问频率，但这同时也会降低芯片外设模块的工作速率。

芯片支持增加 FLASH 存储器等待时间的设置，在不降低系统时钟频率的前提下，降低

FLASH 存储器取指令或数据的频率，从而降低芯片整体功耗。同时 FLASH 存储器支持最高 24MHz 的访问频率，如果系统时钟的频率超过 24MHz，则也需要设置 FLASH 存储器等待时间，否则会导致 FLASH 访问错误。

配置 SCU\_FLASHWAIT 寄存器的 ACCT<3:0>，可设定 FLASH 访问的等待时间。FLASH 访问的等待时间，与芯片支持的系统时钟最高频率的对应关系描述如下：

- ACCT<3:0>=0 时，芯片系统时钟频率最高可为 24MHz；
- ACCT<3:0>=1~F 时，芯片系统时钟频率最高可为 48MHz。

2. 5. 7 特殊功能寄存器

2. 5. 7. 1 FLASH访问等待时间寄存器（SCU\_FLASHWAIT）

FLASH 访问等待时间寄存器（SCU_FLASHWAIT）															
偏移地址：20 <sub>H</sub>															
复位值：00000000_00000000_00000000_00000010 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												ACCT<3:0>			
—	bit31-4				—	—									
ACCT<3:0>	bit3-0				R/W	<b>FLASH 读取访问等待时间设置位</b> 0: 1T <sub>CLK</sub> 完成 FLASH 读取 1: 2T <sub>CLK</sub> 2: 3T <sub>CLK</sub> ... F: 16T <sub>CLK</sub>									

注 1：对 SCU\_FLASHWAIT 寄存器进行写操作前，需要设置 SCU\_PROT 寄存器，关闭写保护。

注 2：TCLK 时钟周期与芯片系统时钟周期相同。

## 2.6 系统时钟

### 2.6.1 概述

芯片系统有四个可选时钟源。

- ◇ 外部时钟源支持两种模式，即高速模式 HS/XT（又称为 HOSC，1~20MHz）和低速模式 LP（又称为 LOSC，32KHz）。
- ◇ 内部高频 RC 时钟源 HRC，支持 2/16/32/48MHz 时钟频率。
- ◇ 内部低频 RC 时钟源 LRC，支持约 32KHz 时钟频率。
- ◇ 支持 2 组 IO 端口可输出系统时钟频率。
- ◇ 外部时钟停振检测，支持停振后自动切换至 LRC 时钟并产生中断。
- ◇ 支持系统时钟滤波，提高系统工作稳定性。

芯片系统时钟源的选择方式，见如下描述：

- 1) 系统时钟为外部时钟源 XTAL：在编程界面中设置配置字，选择高速 HS/XT 模式或低速 LP 模式；在程序软件中设置 SCU\_SCLKEN0 寄存器的 CLK\_SEL=2，设置 SCU\_SCLKEN1 寄存器的 XTAL\_EN=1，选择并使能外部振荡器时钟。
- 2) 系统时钟为内部时钟源 HRC 16MHz：在程序软件中设置 SCU\_SCLKEN0 寄存器的 CLK\_SEL=0，设置 SCU\_SCLKEN1 寄存器的 HRC\_FRE=1。
- 3) 系统时钟为内部时钟源 LRC 32KHz：设置 SCU\_SCLKEN0 寄存器的 CLK\_SEL=1。

芯片支持 2 路 IO 端口输出时钟信号。其中 CLK00 端口支持高频时钟直接输出，CLK01 端口支持高频时钟 512 分频后输出。使用时需配置相应的端口复用选择寄存器 GPIO\_PAFUNC/GPIO\_PBFUNC，使能管脚的时钟输出功能。当使用高频时钟直接输出时，需使能管脚大电流驱动模式，以免输出时钟波形严重失真。

## 2.6.2 结构框图

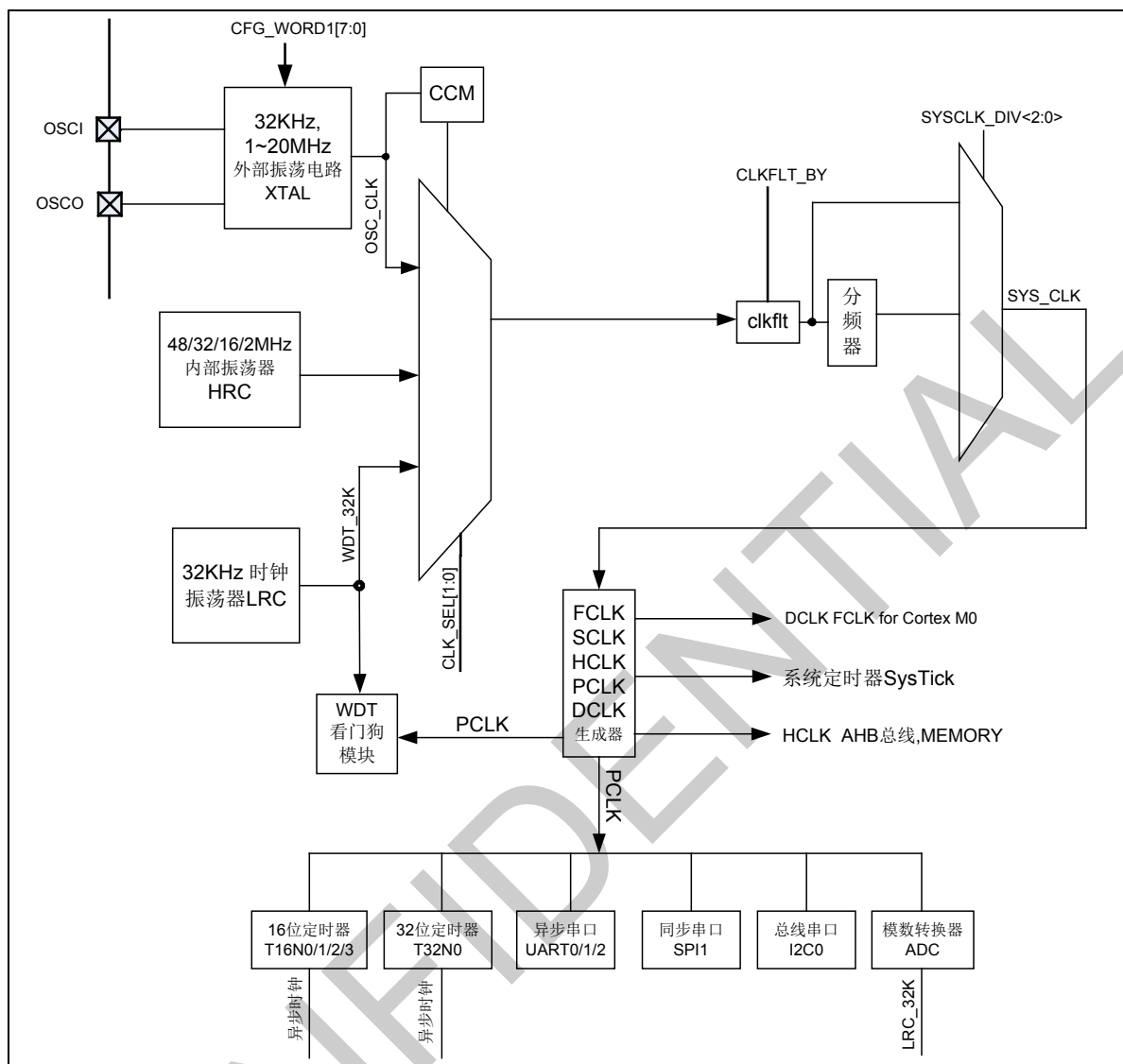


图 2-11 系统时钟电路结构框图

## 2.6.3 功能说明

### 2.6.3.1 外部时钟XTAL

外部振荡器可通过 PB10、PB11 端口串接晶振工作。只要外部振荡器模块使能后 (XTAL\_EN=1)，对应的 IO 端口即被用作模拟端口，禁止其数字输入输出功能。外部振荡器模块使能后，振荡器便可正常起振工作，起振稳定后会置起 SCU\_SCLKEN1 的 XTAL\_RDY 位，振荡器的稳定时间可通过寄存器位 WAKEUPTIME<11:0>进行设置，推荐将其设置为 0xFFFF。

外部时钟源支持两种模式，即高速模式 HS/XT（又称为 HOSC，频率范围为 1~20MHz）和低速模式 LP（又称为 LOSC，频率约为 32KHz）。可在编程界面中设置芯片配置字进行选择，工作在低速模式时，建议使用 32.768KHz 晶振，工作在高速模式 HS 时，建议使用 5~20MHz 晶振，工作在高速模式 XT 时，建议使用 1~4MHz 晶振。

当使用外部振荡器时，需外接匹配电容。XTAL 振荡器电路示意图如下：

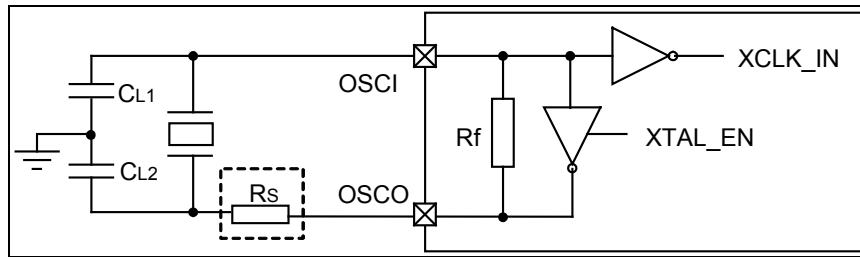


图 2-12 XTAL 振荡器电路结构示意图

注 1: 电阻 RS 为可选配置。

注 2: CL1 和 CL2 为晶振匹配电容，根据所使用的晶振，电容参考取值范围为 10~20pF，建议 1~20MHz 晶振匹配 15pf 电容，32.768KHz 晶振匹配 12pf 电容，具体电容值需根据外接晶振的参数需求确定。

晶体振荡器的等效电路如下图所示：

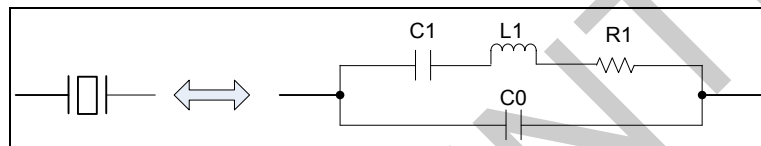


图 2-13 晶体振荡器等效电路示意图

注 1: C1: 动态电容; L1: 动态电感; R1 动态电阻; C0: 静电容; 等效串联电阻  $ESR=R1 \times (1+C0/CL)^2$ ,  $CL$  为负载电容。

注 2: 晶体振荡器与芯片配合使用时，为达到理想的晶振起振和稳定工作状态，对 1~8MHz 晶振参数选型时，推荐  $ESR \leq 200\Omega$ ,  $CL \leq 16pF$  (晶振的负载电容参数  $\leq 16pF$ ); 对 9~20MHz 晶振参数选型时，推荐  $ESR \leq 50\Omega$ ,  $CL \leq 16pF$  (晶振的负载电容参数  $\leq 16pF$ ); 对 32KHz 晶振参数选型时，推荐  $ESR \leq 40K\Omega$ 。

芯片上电后系统时钟默认为内部高速时钟 HRC，需软件配置方可使用外部时钟 XTAL。详情可参考外部时钟操作例程。当 MOSC\_EN=0，芯片进入深度睡眠模式时，XTAL 时钟振荡器会自动关闭，被唤醒后，XTAL 时钟振荡器会自动打开；当 MOSC\_EN=1，芯片进入深度睡眠模式时，XTAL 时钟振荡器不会关闭。

当系统时钟选择为外部时钟 XTAL，在芯片正常工作时，不建议关闭 XTAL 时钟振荡器 (XTAL\_EN=0)，否则系统时钟会自动切换至内部低速 LRC 时钟。

### 2.6.3.2 内部高速时钟HRC

内部高速时钟 HRC 频率可配置为 2MHz、16MHz、32MHz 或 48MHz。常温条件下 HRC 频率精度为  $\pm 1\%$ 。芯片上电后系统时钟默认为内部高速时钟 16MHz HRC，可通过寄存器位 HRC\_EN 关闭。

当系统时钟选择为内部高速时钟 HRC 时，不建议关闭 HRC 时钟 (HRC\_EN=0)，否则系统时钟会自动切换至内部低速 LRC 时钟。

在 HRC\_EN=1 的情况下，当 MOSC\_EN=0，芯片进入深度睡眠模式时，HRC 时钟会自动关断，被唤醒后，HRC 时钟会自动使能；当 MOSC\_EN=1，芯片进入深度睡眠模式



时，HRC 时钟不会关闭。

### 2.6.3.3 内部低速时钟LRC

芯片支持内部低速时钟 LRC（频率约为 32KHz），且无法关闭，始终保持工作。全温度范围内，LRC 时钟频率精度约为±6%。内部低速时钟可供芯片主系统、IWDG、WWDT、ADC 等模块使用。对于时钟频率精度要求高的模块不建议使用 LRC 作为时钟源。

### 2.6.3.4 外部时钟停振检测CCM

外部时钟停振检测模块使能必须同时满足以下条件：

1. SCU\_CCM 寄存器的外部时钟停振检测软件使能位 EN=1。缺省为使能。
2. 时钟源必须选择为外部时钟，即 SCU\_SCLKEN0 寄存器的 CLK\_SEL=2。

外部时钟停振检测开始工作后，当检测到外部时钟停振时，系统时钟会自动切换至 LRC 时钟，同时会置起 SCU\_CCM 寄存器的外部时钟停振标志 FLAG 和 CCM 中断标志位 IF，通过 SCU\_CCM 寄存器的 IFS 位，可配置 CCM 中断标志 IF 的产生方式，通过停振中断使能位 IE，可设置中断标志位 IF 是否触发 CCM 中断请求 IRQ。需注意，当 MOSC\_EN=0 时，因芯片进入深度睡眠模式关闭外部时钟源导致外部时钟停振的情况不会产生时钟停振中断标志位。当芯片进入停振中断后，可根据应用需要进行处理操作，需注意在外部时钟停振期间无法软件进行系统时钟切换，需保持 SCU\_SCLKEN0 寄存器的 CLK\_SEL=2，当外部时钟恢复振荡后，硬件自动将系统时钟切换回外部时钟。

### 2.6.3.5 时钟滤波CLKFLT

芯片支持系统时钟滤波。

系统时钟滤波使能操作时，必须先设置 SCU\_WAKEUPTIME 寄存器的 CLKFLT\_EN 位为 1 以使能系统时钟滤波器，然后设置 SCU\_SCLKEN0 寄存器的 CLKFLT\_BY≠0x55 以选用经滤波的系统时钟。

当需要关闭时钟滤波时，必须先旁路 CLKFLT，即设置 CLKFLT\_BY=0x55，然后再设置 CLKFLT\_EN=0 来关闭时钟滤波器。详情可见时钟滤波例程。

在 CLKFLT\_EN=1 的情况下，当 MOSC\_EN=0，芯片进入深度睡眠模式时，CLKFLT 会自动关断，而当深度睡眠唤醒后，CLKFLT 会自动打开；当 MOSC\_EN=1，芯片进入深度睡眠模式时，CLKFLT 不会关断。

当系统时钟为 48MHz 时钟时，需要设置 CLKFLT\_BY<7:0>=0x55，旁路时钟滤波器；当系统时钟为其它时钟源时，则需设置 CLKFLT\_BY<7:0>=0x00（或其它非 0x55 的值），不要旁路时钟滤波器。

为保证系统工作可靠性，除系统时钟为 HRC 输出 48MHz 时钟外，不建议关闭 CLKFLT。

### 2.6.3.6 睡眠模式系统状态

在睡眠模式下，芯片内核处理器的状态和寄存器，外设寄存器和内部 SRAM 的值都会保持，端口的逻辑电平也会保持睡眠前的状态。

### 2.6.3.7 浅睡眠模式

在浅睡眠模式下，芯片内核时钟停止工作，外设时钟 PCLK 正常运行，芯片时钟源正常工作。

### 2.6.3.8 深度睡眠模式

在深度睡眠模式下，芯片内核时钟停止工作，外设时钟 PCLK 停止运行。

- 1) MOSC\_EN=0 时：除 LRC 时钟源正常运行外，XTAL、HRC 和 CLKFLT 模块全部关断。外设模块只有选择使用 LRC 作为时钟源的能正常工作，其它全部停止工作（异步唤醒功能正常工作）。当芯片唤醒后，XTAL、HRC、CLKFLT 自动恢复到睡眠前的状态。
- 2) MOSC\_EN=1 时：时钟源正常运行，HRC 模块、CLKFLT 正常工作。外设模块只有选择使用 LRC、XTAL 作为时钟源的能正常工作，其它使用 PCLK 或其分频时钟作为时钟源的全部停止工作（异步唤醒功能正常工作）。

## 2.6.4 特殊功能寄存器

### 2.6.4.1 系统时钟控制寄存器 0 (SCU\_SCLKEN0)

系统时钟控制寄存器 0 (SCU\_SCLKEN0)

偏移地址：40<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CLKOUT1_SEL<1:0>		CLKOUT0_SEL<1:0>		CLKFLT_BY<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SYSCLK_DIV			保留										CLK_SEL<1:0>	

—	bit 31-28	—	—
CLKOUT1_SEL<1:0>	bit27-26	R/W	<b>CLKO1 管脚输出选择位</b> 00: 禁止时钟输出 01: 系统时钟输出（512 分频） 10: LRC 时钟输出 11: HRC 时钟输出（512 分频）
CLKOUT0_SEL<1:0>	bit 25-24	R/W	<b>CLKO0 管脚输出选择位</b> 00: 禁止时钟输出 01: 系统时钟输出 10: LRC 时钟输出 11: HRC 时钟输出
CLKFLT_BY<7:0>	bit 23-16	R/W	<b>CLKFLT 旁路控制位</b> 0x55: CLKFLT 旁路 其它: 不旁路 CLKFLT CLKFLT 为系统时钟滤波器。当系统时钟为 48MHz 时，需旁路 CLKFLT，否则可能会造成系统时钟有时失效；当系统时钟为其它时钟

			源时，则不建议旁路 CLKFLT，可进一步提升系统工作稳定性。
—	bit15	—	—
SYSCLK_DIV<2:0>	bit14-12	R/W	<b>系统时钟后分频选择位</b> 000: 1:1 其它: 保留（仅用于内部测试用）
—	bit11-2	—	—
CLK_SEL<1:0>	bit1-0	R/W	<b>系统时钟源选择位</b> 00: HRC 时钟 01: LRC 时钟 10: XTAL 时钟 （可由配置字设置为 HS, XT 或 LP 模式） 11: HRC 时钟

注 1: 对 SCU\_SCLKEN0 寄存器进行写操作前，需要设置 SCU\_PROT 寄存器，关闭写保护。

注 2: 当系统时钟选择为 HRC 32MHz 或 48MHz 时，需要先设置 SCU\_FLASHWAIT 寄存器的 ACCT<3:0>，选择合适的 FLASH 读取时间，再将系统时钟切换到 32MHz 或 48MHz，否则会导致芯片指令运行错误。具体参见“FLASH 存储器等待功能”章节的描述。

## 2.6.4.2 系统时钟控制寄存器 1 (SCU\_SCLKEN1)

系统时钟控制寄存器 1 (SCU\_SCLKEN1)

偏移地址: 44<sub>H</sub>

复位值: 00000000\_00000010\_00000000\_00000110<sub>B</sub>

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

—	bit 31-18	—	—
HRC_RDY	bit17	R	<b>内部高速时钟振荡模式稳定标志位</b> 0: 不稳定 1: 稳定
XTAL_RDY	bit16	R	<b>外部时钟振荡模式稳定标志位</b> 0: 不稳定 1: 稳定 该标志位仅在寄存器位 XTAL_EN=1 时有效，作为 XTAL 振荡器工作稳定的标志位，推荐将寄存器位 WAKEUPTIME<11:0>设置为 0xFFFF
—	bit15-4	—	—
HRC_FRE	bit3-2	R/W	<b>HRC 时钟频率选择位</b> 00: 2MHz 01: 16MHz（默认）

			10: 32MHz 11: 48MHz
HRC_EN	bit1	R/W	内部高速时钟振荡电路使能位 0: 禁止 1: 使能
XTAL_EN	bit0	R/W	外部时钟振荡电路使能位 0: 禁止 1: 使能

注 1: 对 SCU\_SCLKEN1 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。

注 2: 在时钟管脚无外接晶振或存在外部干扰的情况下, 外部时钟振荡稳定标志位 XTAL\_RDY 可能会被误置 1。

### 2.6.4.3 外设时钟控制寄存器 (SCU\_PCLKEN)

外设时钟控制寄存器 (SCU\_PCLKEN)

偏移地址: 48<sub>H</sub>

复位值: 00010010\_00000111\_00011111\_11010111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	I2C0_EN	保留	SPI1_EN	保留	保留	保留	保留	保留	保留	保留	保留	保留	UART2_EN	UART1_EN	UART0_EN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	T32N0_EN	T16N3_EN	T16N2_EN	T16N1_EN	T16N0_EN	IWDT_EN	WWDT_EN	保留	ADC_EN	保留	IAP_EN	GPIO_EN	SCU_EN		

—	bit 31-29	—	—
I2C0_EN	bit28	R/W	<b>I2C0 时钟使能位</b> 0: 禁止 1: 使能
—	bit 27-26	—	—
SPI1_EN	bit 25	R/W	<b>SPI1 时钟使能位</b> 0: 禁止 1: 使能
—	bit 24-19	—	—
UART2_EN	bit 18	R/W	<b>UART2 时钟使能位</b> 0: 禁止 1: 使能
UART1_EN	bit 17	R/W	<b>UART1 时钟使能位</b> 0: 禁止 1: 使能
UART0_EN	bit 16	R/W	<b>UART0 时钟使能位</b> 0: 禁止 1: 使能
—	bit 15-13	—	—

T32N0_EN	bit 12	R/W	<b>T32N0 时钟使能位</b> 0: 禁止 1: 使能
T16N3_EN	bit 11	R/W	<b>T16N3 时钟使能位</b> 0: 禁止 1: 使能
T16N2_EN	bit 10	R/W	<b>T16N2 时钟使能位</b> 0: 禁止 1: 使能
T16N1_EN	bit 9	R/W	<b>T16N1 时钟使能位</b> 0: 禁止 1: 使能
T16N0_EN	bit 8	R/W	<b>T16N0 时钟使能位</b> 0: 禁止 1: 使能
IWDT_EN	bit 7	R/W	<b>IWDT 时钟使能位</b> 0: 禁止 1: 使能
WWDT_EN	bit 6	R/W	<b>WWDT 时钟使能位</b> 0: 禁止 1: 使能
—	bit 5	—	—
ADC_EN	bit 4	R/W	<b>ADC 时钟使能位</b> 0: 禁止 1: 使能
—	bit 3	—	—
IAP_EN	bit 2	R/W	<b>FLASH_IAP 时钟使能位</b> 0: 禁止 1: 使能
GPIO_EN	bit 1	R/W	<b>GPIO 时钟使能位</b> 0: 禁止 1: 使能
SCU_EN	bit 0	R/W	<b>SCU 时钟使能位</b> 0: 禁止 1: 使能

注 1: 对 SCU\_PCLKEN 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。

注 2: 使能某外设时钟之前需要先使能 SCU 时钟, 即 SCU\_PCLKEN 寄存器 SCU\_EN 位设置为 1。

以 IWDT 时钟配置为例:

```
LDR R0, =SCU_PCLKEN
```

```
LDR R1, =0X00000001
```

```
STR R1, [R0] ; 首先使能 SCU 时钟
```

```
LDR R1, =0X00000081
```

STR R1, [R0] ; 使能 SCU 和 IWDT 时钟

注 3: IWDT 和 WWDT 使用 LRC 时钟计数时, 其 PCLK 时钟使能位 IWDT\_EN=0, WWDT\_EN=0 时, 对 IWDT 和 WWDT 模块的寄存器读写操作被禁止, 但 IWDT 和 WWDT 计数器仍保持工作状态, 看门狗定时器功能仍有效。

注 4: 外设模块时钟关闭后, 与该外设模块对应的所有特殊功能寄存器均保持时钟关闭前的状态, 并且无法进行读写操作。

#### 2.6.4.4 系统唤醒时间控制寄存器 (SCU\_WAKEUPTIME)

系统唤醒时间控制寄存器 (SCU\_WAKEUPTIME)

偏移地址: 4C<sub>H</sub>

复位值: 00000000\_00000100\_01110000\_01111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								FLS_ST OP	保留	BG_STO P	LP_STO P	保留	LDOLP_VOSE L<1:0>	保留	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STPR TNEN	VROSC EN	CLKFLT_E N	MOSC_E N	WAKEUPTIME<11:0>											

—	bit31-24	—	—
FLS_STOP	bit 23	R/W	深度睡眠模式下, <b>Flash STOP</b> 使能位 0: 禁止 1: 使能 (推荐设置为使能, 以降低功耗)
—	bit 22	—	—
BG_STOP	bit 21	R/W	深度睡眠模式下, <b>BG</b> 低功耗使能位 0: 禁止 1: 使能 (设置为使能可以降低深睡眠模式功耗, 使能时会影响芯片深睡眠模式下的 BOR 和 LVD 档位电压, 见表格下方备注)
LP_STOP	bit 20	R/W	深度睡眠模式下, <b>LDO</b> 低功耗使能位 0: 禁止 1: 使能 (推荐设置为使能, 以降低功耗)
—	bit 19	—	—
LDOLP_VOSEL<1:0>	bit18-17	W/R	在深度睡眠模式下, <b>LDO</b> 电压输出选择位 01: 1.4V (需软件固定设置为 01) 00, 10, 11: 保留 (仅用于内部测试用)
—	bit 16	—	—
STPRTNEN	bit15	R/W	深度睡眠模式下, <b>SRAM</b> 低功耗使能位 0: 禁止 1: 使能 (推荐设置为使能, 以降低功耗)
VROSCEN	bit14	R/W	<b>VR</b> 工作时钟使能位 0: 禁止 (推荐设置为禁止, 以降低功耗) 1: 使能

CLKFLT_EN	bit13	RW	<b>CLKFLT 系统时钟滤波器使能位</b> 0: 禁止 1: 使能 CLKFLT 为系统时钟滤波器，当系统时钟为 HRC 48MHz 时，需禁止 CLKFLT；当系统时钟为其它时钟源时，则建议使能 CLKFLT，可进一步提升系统工作稳定性，在深度睡眠模式下，可禁止 CLKFLT，降低芯片功耗
MOSC_EN	bit12	RW	<b>深度睡眠模式下，时钟控制位</b> 0: 深度睡眠模式下，自动关闭 HRC、XTAL 和时钟滤波器 CLKFLT 1: 深度睡眠模式下，使能 HRC、XTAL 和时钟滤波器 CLKFLT（需同时设置 FLS_STOP=0）
WAKEUPTIME<11:0>	bit11-0	RW	<b>唤醒时间控制位</b> $T_{PCLK} * WAKEUPTIME$

注 1: 对 SCU\_WAKEUPTIME 寄存器进行写操作前，需要设置 SCU\_PROT 寄存器，关闭写保护。

注 2: 对 LDOLP\_VOSEL<1:0>寄存器位，需软件在芯片初始化时固定设置为 01，以降低深度睡眠模式下的芯片功耗。

注 3: 深度睡眠模式下，当 MOSC\_EN 为 1 时，HRC、XTAL 和时钟滤波器还必须各自的控制位 HRC\_EN、XTAL\_EN 和 CLKFLT\_EN 为 1 时，才实际被使能，如果使能了 HRC、PLL 或 XTAL，则还必须设置 FLS\_STOP 为 0，禁止 Flash STOP 模式，同时将 WAKEUPTIME<11:0>设置为大于等于 0x3FF。

注 4: 唤醒时间控制位 WAKEUPTIME<11:0>，用于设定在深度睡眠模式下，被关闭的 HRC 和 XTAL 时钟模块，在芯片被唤醒后，能够恢复稳定工作的等待时间，通常需要设置唤醒时间控制位 WAKEUPTIME 的值大于等于 0x120（使用 XTAL 时钟时推荐设置为 0xFFFF），具体根据芯片在应用系统中的实际工作状况进行调整。

注 5: 芯片在深睡眠模式下，如果 BG 电压模块为低功耗模式（寄存器 SCU\_WAKEUPTIME 的 BG\_STOP=1），则 BOR 和 LVD 档位电压相对于芯片工作模式下的档位电压均会有约 ±10% 范围的偏差；如果应用中需要 BOR 和 LVD 档位电压在芯片深睡眠模式与工作模式下保持一致，则需要禁止 BG 低功耗模式（设置寄存器 SCU\_WAKEUPTIME 的 BG\_STOP=0），此时芯片深睡眠模式功耗会增大约 0.5uA。

注 6: 唤醒时间控制位 WAKEUPTIME<11:0>，还用于设置外部时钟 XTAL 的稳定时间，在使用外部时钟 XTAL 时，推荐将其设置为 0xFFFF，以增强振荡器工作稳定标志位 XTAL\_RDY 的可靠性。

#### 2.6.4.5 外部时钟检测控制寄存器（SCU\_CCM）

外部时钟检测控制寄存器（SCU\_CCM）

偏移地址：2CH

复位值：00000000\_00000000\_00000000\_00000001<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															FLAG
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							IF	IFS<2:0>			IE	保留		EN	
—			bit31-17			—			—						

FLAG	bit16	R	<b>CCM 外部时钟停振检测标志位</b> 0: 晶振未停振 1: 晶振停振
—	bit15-9	—	—
IF	bit8	R/W	<b>CCM 中断标志位</b> 0: 未发生 CCM 触发事件 1: 发生 CCM 触发事件 边沿模式产生中断标志时，可以写 1 清除标志； 电平模式产生中断标志时，该标志只读，触发电平消失后，中断标志自动清零。
IFS<2:0>	bit7-5	R/W	<b>CCM 中断标志产生模式选择位</b> 000: CCM_FLAG 上升沿产生中断，晶振停振 001: CCM_FLAG 下降沿产生中断，晶振恢复振荡 010: CCM_FLAG 高电平产生中断，晶振停振 011: CCM_FLAG 低电平产生中断，晶振恢复振荡 1xx: CCM_FLAG 变化（上升或下降沿）产生中断
IE	bit 4	R/W	<b>外部时钟停振中断使能位</b> 0: 禁止 1: 使能
—	bit3-1	—	—
EN	bit0	R/W	<b>外部时钟停振检测使能位</b> 0: 禁止 1: 使能

注 1: 对 SCU\_CCM 寄存器进行写操作前，需要设置 SCU\_PROT 寄存器，关闭写保护。

注 2: 在时钟管脚无外接晶振或存在外部干扰的情况下，晶振停振检测标志可能会失效。

## 2.6.5 系统时钟应用说明

以下操作都已关闭系统保护寄存器 SCU\_PROT，且使能了 SCU\_PCLKEN 寄存器的 SCU 时钟使能位 SCU\_EN。

### 2.6.5.1 外部时钟XTAL

使用外部时钟 XTAL:

```
SWITCH_XTAL PROC
```

```
PUSH    {LR}
```

```
LDR     R0, =SCU_SCLKEN1
```

```
LDR     R1, [R0]
```

```
LDR     R2, =0X01
```

```
ORRS    R1, R1, R2
```

```
STR     R1, [R0]
```

```
;使能 XTAL_EN
```

```
WAIT_XTAL_FLAG
```

```
LDR     R0, =SCU_SCLKEN1
```

```
LDR     R1, [R0]
```

```
LDR     R2, =0X010000
```



```

TST    R1, R2
BEQ     WAIT_XTAL_FLAG           ;等待 XTAL_RDY

LDR     R0, =SCU_SCLKEN0
LDR     R1, =0X02
STRB    R1, [R0]                 ;系统时钟选用 XTAL

POP     {PC}
ALIGN
LTORG
ENDP

```

### 2.6.5.2 内部高速时钟HRC

使用内部高速时钟 HRC:

```

SWITCH_HRC PROC
PUSH    {LR}
LDR     R0, =SCU_SCLKEN1
LDR     R1, [R0]
LDR     R2, =0X02
ORRS    R1, R1, R2
STR     R1, [R0]                 ;使能 HRC_EN
WAIT_HRC_FLAG
LDR     R0, =SCU_SCLKEN1
LDR     R1, [R0]
LDR     R2, =0X020000
TST     R1, R2
BEQ     WAIT_HRC_FLAG           ;等待 HRC_RDY

LDR     R0, =SCU_SCLKEN0
LDRB    R1, [R0]
LDR     R2, =0XFC
ANDS    R1, R1, R2
STRB    R1, [R0]                 ;系统时钟选用 HRC
POP     {PC}
ALIGN
LTORG
ENDP

```

### 2.6.5.3 内部低速时钟LRC

使用内部低速时钟 LRC:

```

SWITCH_LRC PROC
PUSH    {LR}
LDR     R0, =SCU_SCLKEN0
LDRB    R1, [R0]

```

```
LDR    R2, =0XFC
ANDS   R1, R1, R2
LDR    R2, =0X01
ORRS   R1, R1, R2
STRB   R1, [R0]           ;系统时钟选用 LRC
POP    {PC}
ALIGN
LTORG
ENDP
```

## 2.7 中断和异常处理

### 2.7.1 中断和异常

Cortex-M0 内核支持嵌套向量中断控制器 NVIC(Nested Vectored Interrupt Controller)，具体功能如下：

- ◇ 支持中断嵌套
- ◇ 支持中断向量
- ◇ 支持中断优先级动态调整
- ◇ 支持中断可屏蔽

对 Cortex-M0 内核来说，打断程序正常执行流程的事件均称之为异常，中断也是其中一种异常。为便于理解，本文档将内核的中断等事件称为异常，将外设模块的中断称为中断。

异常/中断优先级操作说明：

操作类型	描述
抢占	产生条件：ISR 或线程正在执行时，出现新的优先级更高的异常/中断。 操作结果：如果当前处于线程状态，则产生异常/中断挂起中断；如果当前处于 ISR 状态，则产生中断嵌套，处理器自动保存工作状态并压栈。
末尾连锁	产生条件：当前 ISR 执行结束，正在返回时，出现新的优先级更高的异常/中断。 操作结果：跳过出栈操作，处理新的异常/中断。
返回	产生条件：当前 ISR 执行结束，正在返回时，没有出现新的优先级更高的异常/中断。 操作结果：执行出栈操作，并将处理器状态恢复为进入 ISR 之前的状态。
迟来	产生条件：当前 ISR 执行开始，正在保存时，出现新的优先级更高的异常/中断。 操作结果：处理器转去处理优先级更高的异常/中断。

表 2-2 异常/中断优先级操作类型说明说明表

注 1：ISR – Interrupt Service Routine，中断服务程序。

异常/中断优先级：

编号	类型	优先级	简介
0	N/A	N/A	没有异常在运行
1	复位	-3（最高）	复位
2	NMI	-2	不可屏蔽中断（来自外设 NMI 中断输入）
3	Hard Fault	-1	所有被禁用的 Fault，都将升级为 Hard Fault
4~10	保留	NA	—
11	SVC	可编程控制	系统服务调用
12~13	保留	NA	—
14	PendSV	可编程控制	为系统设备而设的“可悬挂请求”
15	SysTick	可编程控制	系统定时计数器
16	IRQ0	可编程控制	外设中断 0

编号	类型	优先级	简介
17	IRQ1	可编程控制	外设中断 1
...	...	...	...
47	IRQ31	可编程控制	外设中断 31

表 2-3 异常/中断优先级列表

Cortex-M0 支持如下异常/中断：

NMI 中断、Hard Fault 异常、SVC 异常、PendSV 异常、SysTick 异常和 32 个外设中断请求 IRQ0~IRQ31。

其中 Hard Fault 异常、SVC 异常、PendSV 异常、SysTick 异常为 Cortex-M0 内核异常源，只受 Cortex-M0 内核控制，而 NMI 中断与 32 个 IRQ 可由芯片配置控制。

虽然 Cortex-M0 对 NMI 不支持中断使能位，但为了防止芯片上电初始化完成前，误产生 NMI 中断源，而误进中断，芯片提供了 NMI 使能位 NMIEN，可在 NMI 中断源配置完成后，再设置 NMIEN=1。

对于 32 个 IRQ，Cortex-M0 内核提供 32 个 IRQ 使能位，可对每个中断请求独立控制。配置 NVIC\_ISER 和 NVIC\_ICER 中断控制寄存器可使能或禁止 IRQ。

配置 NVIC\_PR0~NVIC\_PR7 优先级控制寄存器，可设置 IRQ0~IRQ31 的中断优先级。如果同时产生多个 IRQ 请求，则最先响应优先级最高的 IRQ；如果同时产生多个相同最高优先级的 IRQ 请求，则按照中断向量分配表，最先响应向量表编号最低的 IRQ，即如果同时产生中断优先级相同的 IRQ0 与 IRQ1，则先响应 IRQ0。

## 2.7.2 中断和异常向量的分配

编号	类型	功能	说明
0~15	异常	—	Cortex-M0 内核异常，包括 NMI 不可屏蔽中断
16	IRQ0	PINT0 中断	外部端口中断 0
17	IRQ1	PINT1 中断	外部端口中断 1
18	IRQ2	PINT2 中断	外部端口中断 2
19	IRQ3	PINT3 中断	外部端口中断 3
20	IRQ4	PINT4 中断	外部端口中断 4
21	IRQ5	PINT5 中断	外部端口中断 5
22	IRQ6	PINT6 中断	外部端口中断 6
23	IRQ7	PINT7 中断	外部端口中断 7
24	IRQ8	T16N0 中断	16 位定时器/计数器 0 中断
25	IRQ9	T16N1 中断	16 位定时器/计数器 1 中断
26	IRQ10	T16N2 中断	16 位定时器/计数器 2 中断
27	IRQ11	T16N3 中断	16 位定时器/计数器 3 中断
28	IRQ12	T32N0 中断	32 位定时器/计数器 0 中断
29	IRQ13	Reserved	预留
30	IRQ14	Reserved	预留
31	IRQ15	WWDT 中断	窗口看门狗中断

编号	类型	功能	说明
32	IRQ16	IWDT 中断	独立看门狗中断
33	IRQ17	Reserved	预留
34	IRQ18	KINT 中断	外部按键输入中断
35	IRQ19	ADC 中断	模数转换中断
36	IRQ20	Reserved	预留
37	IRQ21	LVD 中断	低电压检测中断
38	IRQ22	Reserved	预留
39	IRQ23	UART0 中断	UART0 中断
40	IRQ24	UART1 中断	UART1 中断
41	IRQ25	UART2 中断	UART2 中断
42	IRQ26	Reserved	预留
43	IRQ27	Reserved	预留
44	IRQ28	SPI1 中断	SPI1 中断
45	IRQ29	I2C0 中断	I2C0 中断
46	IRQ30	Reserved	预留
47	IRQ31	CCM 中断	外部振荡器停振检测中断

表 2-4 IRQ 分配列表

### 2.7.3 中断向量的重映射

Cortex-M0 内核本身并不支持中断向量的重映射，芯片中有两个特殊功能寄存器“中断向量表重映射使能寄存器”和“中断向量表偏移寄存器”，可以支持中断向量的重映射。具体的使用方式可参考 Flash 自编程（IAP）相关章节的描述。

### 2.7.4 特殊功能寄存器

#### 2.7.4.1 不可屏蔽中断控制寄存器（SCU\_NMICON）

不可屏蔽中断控制寄存器（SCU\_NMICON）

偏移地址：04<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-6	—	—
NMICS<4:0>	bit5-1	R/W	<b>NMI 不可屏蔽中断选择位</b> 00000: IRQ0 00001: IRQ1 ... 11111: IRQ31
NMIEN	bit0	R/W	<b>NMI 不可屏蔽中断使能位</b>

			0: 禁止 1: 使能
--	--	--	----------------

注 1: 对 SCU\_NMICON 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。

#### 2.7.4.2 中断向量表重映射使能寄存器 (SCU\_TBLREMAPEN)

中断向量表重映射使能寄存器 (SCU\_TBLREMAPEN)

偏移地址: 60<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-1	—	—
EN	bit0	R/W	<p><b>中断向量表重映射使能</b></p> <p>0: 中断向量表位于 Flash Memory 的 “0” 地址开始的一段空间 (默认状态); 目前共支持 48 个向量, 因此, 这段空间的大小为 192 字节;</p> <p>1: 中断向量表位于 “中断向量表偏移寄存器” 指定的地址开始的 192 字节空间。</p>

注: 对 SCU\_TBLREMAPEN 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。

#### 2.7.4.3 中断向量表偏移寄存器 (SCU\_TBLOFF)

中断向量表偏移寄存器 (SCU\_TBLOFF)

偏移地址: 64<sub>H</sub>

复位值: 00100000\_00000000\_00000000\_00000000<sub>B</sub>

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

TBLOFF<31:0>	bit31-0	R/W	<p><b>中断向量表偏移地址</b></p> <p>该寄存器存放重映射后的中断向量表所在的起始地址, “中断向量表重映射使能寄存器” 为 “1” 时有效。</p> <p>高 24 位 TBLOFF&lt;31:8&gt;可读可写, 但低 8 位 TBLOFF&lt;7:0&gt;只读, 不可写, 且读取时返回全零。</p>
--------------	---------	-----	---------------------------------------------------------------------------------------------------------------------------------------------------------------------

注: 该地址为起始地址是有要求的: 必须先求出系统中共有多少个向量, 再把这个数字向上增大到是 2 的整次幂, 而起始地址必须对齐到后者的边界上。如果一共有 32 个中断, 则共有 32+16 (系统异常)=48 个向量, 向上

增大到 2 的整次幂后值为 64，因此地址值必须能被  $64 \times 4 = 256$  整除，从而合法的起始地址可以是：0x000，0x100，0x200 等。

#### 2.7.4.4 硬件错误标志寄存器 (SCU\_FAULTFLAG)

##### 硬件错误标志寄存器 (SCU\_FAULTFLAG)

偏移地址：0C<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													FLAG2	FLAG1	FLAG0

—	bit31-3	—	—
FLAG2	bit2	R/W	<b>硬件错误 2 标志位</b> 0：未发生在异常区域进行写入操作 1：发生在异常区域进行写入操作（硬件自动置 1，软件写 1 清除）
FLAG1	bit1	R/W	<b>硬件错误 1 标志位</b> 0：未发生在异常区域进行取指操作 1：发生在异常区域进行取指操作（硬件自动置 1，软件写 1 清除）
FLAG0	bit0	R/W	<b>硬件错误 0 标志位</b> 0：未发生读指令代码为空 1：发生读指令代码为空（硬件自动置 1，软件写 1 清除）

注 1：读指令代码为空表示 Cortex-M0 内核读 Flash 程序存储器的指令时，读到的值为 FFFFFFFF<sub>H</sub>。

注 2：清除硬件错误标志位时，需要设置 SCU\_PROT 寄存器，关闭写保护。

#### 2.7.4.5 IRQ0~31 置中断请求使能寄存器 (NVIC\_ISER)

##### IRQ0~31 置中断请求使能寄存器 (NVIC\_ISER)

偏移地址：00<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

SETENA<31:0>	bit31-0	R/W	<b>IRQx 使能位</b> 0：中断使能无效 1：中断使能有效
--------------	---------	-----	-----------------------------------------

			软件写 1 使能中断请求，写 0 无效
--	--	--	---------------------

注：对 NVIC\_ISER 寄存器中的各 IRQx 使能位，写 0 无效，写 1 才使能中断请求；读操作时，实际是读取 IRQx 中断使能的状态，读取的值为 1 表示中断使能有效，为 0 表示中断使能无效。

#### 2.7.4.6 IRQ0~31 清中断请求使能寄存器 (NVIC\_ICER)

##### IRQ0~31 清中断请求使能寄存器 (NVIC\_ICER)

偏移地址：80<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

CLRENA<31:0>	bit31-0	R/W	<b>IRQx 禁止位</b> 0: 中断禁止无效 1: 中断禁止有效 软件写 1 禁止中断请求，写 0 无效
--------------	---------	-----	------------------------------------------------------------------

注：对 NVIC\_ICER 寄存器中的各 IRQx 禁止位，写 0 无效，写 1 才禁止中断请求；读操作时，实际是读取 IRQx 中断禁止的状态，读取的值为 1 表示中断禁止有效，为 0 表示中断禁止无效。

#### 2.7.4.7 IRQ0~31 置中断挂起寄存器 (NVIC\_ISPR)

##### IRQ0~31 置中断挂起寄存器 (NVIC\_ISPR)

偏移地址：100<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

SETPEND<31:0>	bit31-0	R/W	<b>置 IRQx 挂起位</b> 0: 中断未挂起 1: 中断挂起 软件写 1 挂起中断，写 0 无效
---------------	---------	-----	---------------------------------------------------------------

注：对 NVIC\_ISPR 寄存器中的各 IRQx 挂起位，写 0 无效，写 1 才挂起中断；读操作时，实际是读取 IRQx 中断挂起的状态，读取的值为 1 表示中断挂起，为 0 表示中断未挂起。



### 2.7.4.8 IRQ0~31 清中断挂起寄存器 (NVIC\_ICPR)

#### IRQ0~31 清中断挂起寄存器 (NVIC\_ICPR)

偏移地址: 180<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

CLRPEND<31:0>	bit31-0	R/W	<b>清 IRQx 挂起位</b> 0: 中断未挂起 1: 中断挂起 软件写 1 清除中断挂起, 写 0 无效
---------------	---------	-----	------------------------------------------------------------------

注: 对 NVIC\_ICPR 寄存器中的各 IRQx 清挂起位, 写 0 无效, 写 1 才清除中断挂起; 读操作时, 实际是读取 IRQx 中断挂起的状态, 读取的值为 1 表示中断挂起, 为 0 表示中断未挂起。

### 2.7.4.9 IRQ0~3 优先级控制寄存器 (NVIC\_PR0)

#### IRQ0~3 优先级控制寄存器 (NVIC\_PR0)

偏移地址: 300<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_3<1:0>		保留						PRI_2<1:0>		保留					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_1<1:0>		保留						PRI_0<1:0>		保留					

PRI_3<1:0>	bit31-30	R/W	<b>IRQ3 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_2<1:0>	bit23-22	R/W	<b>IRQ2 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_1<1:0>	bit15-14	R/W	<b>IRQ1 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_0<1:0>	bit7-6	R/W	<b>IRQ0 优先级设置位</b> 00: 最高优先级 11: 最低优先级

—	bit5-0	—	—
---	--------	---	---

## 2.7.4.10 IRQ4~7 优先级控制寄存器 (NVIC\_PR1)

### IRQ4~7 优先级控制寄存器 (NVIC\_PR1)

偏移地址: 304<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_7<1:0>		保留						PRI_6<1:0>		保留					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_5<1:0>		保留						PRI_4<1:0>		保留					

PRI_7<1:0>	bit31-30	R/W	<b>IRQ7 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_6<1:0>	bit23-22	R/W	<b>IRQ6 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_5<1:0>	bit15-14	R/W	<b>IRQ5 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_4<1:0>	bit7-6	R/W	<b>IRQ4 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

## 2.7.4.11 IRQ8~11 优先级控制寄存器 (NVIC\_PR2)

### IRQ8~11 优先级控制寄存器 (NVIC\_PR2)

偏移地址: 308<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_11<1:0>		保留						PRI_10<1:0>		保留					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_9<1:0>		保留						PRI_8<1:0>		保留					

PRI_11<1:0>	bit31-30	R/W	<b>IRQ11 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—

PRI_10<1:0>	bit23-22	R/W	<b>IRQ10 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_9<1:0>	bit15-14	R/W	<b>IRQ9 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_8<1:0>	bit7-6	R/W	<b>IRQ8 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

#### 2.7.4.12 IRQ12~15 优先级控制寄存器 (NVIC\_PR3)

IRQ12~15 优先级控制寄存器 (NVIC\_PR3)

偏移地址: 30C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_15<1:0>	保留							PRI_14<1:0>	保留						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_13<1:0>	保留							PRI_12<1:0>	保留						

PRI_15<1:0>	bit31-30	R/W	<b>IRQ15 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_14<1:0>	bit23-22	R/W	<b>IRQ14 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_13<1:0>	bit15-14	R/W	<b>IRQ13 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_12<1:0>	bit7-6	R/W	<b>IRQ12 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

## 2.7.4.13 IRQ16~19 优先级控制寄存器 (NVIC\_PR4)

IRQ16~19 优先级控制寄存器 (NVIC\_PR4)

偏移地址: 310<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_19<1:0>				保留				PRI_18<1:0>				保留			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_17<1:0>				保留				PRI_16<1:0>				保留			

PRI_19<1:0>	bit31-30	R/W	<b>IRQ19 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_18<1:0>	bit23-22	R/W	<b>IRQ18 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_17<1:0>	bit15-14	R/W	<b>IRQ17 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_16<1:0>	bit7-6	R/W	<b>IRQ16 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

## 2.7.4.14 IRQ20~23 优先级控制寄存器 (NVIC\_PR5)

IRQ20~23 优先级控制寄存器 (NVIC\_PR5)

偏移地址: 314<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_23<1:0>				保留				PRI_22<1:0>				保留			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_21<1:0>				保留				PRI_20<1:0>				保留			

PRI_23<1:0>	bit31-30	R/W	<b>IRQ23 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_22<1:0>	bit23-22	R/W	<b>IRQ22 优先级设置位</b> 00: 最高优先级

			11: 最低优先级
—	bit21-16	—	—
PRI_21<1:0>	bit15-14	R/W	<b>IRQ21 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_20<1:0>	bit7-6	R/W	<b>IRQ20 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

#### 2.7.4.15 IRQ24~27 优先级控制寄存器 (NVIC\_PR6)

IRQ24~27 优先级控制寄存器 (NVIC\_PR6)

偏移地址: 318<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_27<1:0>	保留							PRI_26<1:0>	保留						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_25<1:0>	保留							PRI_24<1:0>	保留						

PRI_27<1:0>	bit31-30	R/W	<b>IRQ27 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_26<1:0>	bit23-22	R/W	<b>IRQ26 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_25<1:0>	bit15-14	R/W	<b>IRQ25 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_24<1:0>	bit7-6	R/W	<b>IRQ24 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

## 2.7.4.16 IRQ28~31 优先级控制寄存器 (NVIC\_PR7)

IRQ28~31 优先级控制寄存器 (NVIC\_PR7)

偏移地址: 31C<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_31<1:0>		保留						PRI_30<1:0>		保留					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_29<1:0>		保留						PRI_28<1:0>		保留					

PRI_31<1:0>	bit31-30	R/W	<b>IRQ31 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_30<1:0>	bit23-22	R/W	<b>IRQ30 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_29<1:0>	bit15-14	R/W	<b>IRQ29 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_28<1:0>	bit7-6	R/W	<b>IRQ28 优先级设置位</b> 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

## 2.8 系统控制块（SCB）

### 2.8.1 概述

系统控制块提供芯片内核系统实现的状态信息，并对内核系统工作进行控制。

SCB 寄存器列表和基址参见章节：系统控制块（SCB）寄存器列表

### 2.8.2 特殊功能寄存器

#### 2.8.2.1 SCB\_CPUID 寄存器（SCB\_CPUID）

SCB\_CPUID 寄存器（SCB\_CPUID）

偏移地址：00<sub>H</sub>

复位值：01000001\_00001100\_11000010\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMPLEMENTER<7:0>								VARIANT<3:0>				CONSTANT<3:0>			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARTNO<11:0>												REVISION<3:0>			

IMPLEMENTER<7:0>	bit31-24	R	处理器实现者编号 0x41, ARM
VARIANT<3:0>	bit23-20	R	主版本号 R=0x0, 作为 rmpn 版本编号格式中的主要编号
CONSTANT<3:0>	bit19-16	R	处理器构架 0xC, ARMv6-M
PARTNO<11:0>	bit15-4	R	处理器分类号 0xC20, Cortex-M0
REVISION<3:0>	bit3-0	R	次版本号 P=0x0, 作为 rmpn 版本编号格式中的次要编号

#### 2.8.2.2 中断控制和状态寄存器（SCB\_ICSR）

中断控制和状态寄存器（SCB\_ICSR）

偏移地址：04<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NMIPENDSET	保留				PENDSTSET	PENDSTCLR			ISRPPENDING	保留				VECTPENDING<5:4>	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VECTPENDING<3:0>					保留					VECTACTIVE<5:0>					

NMIPENDSET	bit31	R/W	<b>NMI 中断挂起控制位</b> 0: 不置 NMI 中断挂起 1: 置 NMI 中断挂起
—	bit30-27	—	—

PENDSTSET	bit26	R/W	置 SysTick 异常挂起位 0: 无效 1: 置 SysTick 异常挂起
PENDSTCLR	bit25	W	清 SysTick 异常挂起位 0: 无效 1: 清除 SysTick 异常挂起
—	bit24-23	—	—
ISRPENDING	bit22	R	中断挂起标志位 0: 无中断挂起 1: 有中断挂起
—	bit21-18	—	—
VECTPENDING	bit17-12	R	当前的挂起中，优先级最高的异常/中断号 0x0: 无挂起异常/中断 非 0: 当前被挂起的异常/中断中，优先级最高的异常/中断号
—	bit11-6	—	—
VECTACTIVE	bit5-0	R	当前被处理的异常/中断号 0x0: 线程 (Thread) 模式 非 0: 当前被处理的异常/中断号

### 2.8.2.3 应用中断和复位控制寄存器 (SCB\_AIRCR)

应用中断和复位控制寄存器 (SCB\_AIRCR)

偏移地址: 0C<sub>H</sub>复位值: 11111010\_00000101\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEY<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANNESS	保留												SYSRESET REQ	VECTCLR ACTIVE	保留

VECTKEY<15:0>	bit31-16	W	向量关键码位 只能写 0x05FA，其它无效
ENDIANNESS	bit15	R	存储器数据格式选择位 0: 小端格式 1: 大端格式
—	bit14-3	—	—
SYSRESETREQ	bit2	W	系统复位请求位 0: 无效 1: 请求系统复位，复位后自动清零
VECTCLRACTIVE	bit1	W	异常/中断状态清除位 该位只能写 0；写 1 会产生 HardFault 异常
—	bit0	—	—



注：寄存器 SCB\_AIRCR 只能进行字写入，且高半字只能写入 0x05FA，否则对该寄存器的写入操作无效。

#### 2.8.2.4 系统控制寄存器 (SCB\_SCR)

##### 系统控制寄存器 (SCB\_SCR)

偏移地址：10<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-5	—	—
SEVONPEND	bit4	R/W	中断被挂起时，是否作为唤醒事件的选择位 0：中断被挂起时，不作为唤醒事件 1：中断被挂起时，作为唤醒事件
—	bit3	—	—
SLEEPDEEP	bit2	R/W	休眠模式选择位 0：浅睡眠模式 1：深度睡眠模式
SLEEPONEXIT	bit1	R/W	从 ISR 中断处理程序返回到线程模式时，是否进入休眠状态的选择位 0：不进入休眠状态 1：进入休眠状态
—	bit0	—	—

#### 2.8.2.5 配置和控制寄存器 (SCB\_CCR)

##### 配置和控制寄存器 (SCB\_CCR)

偏移地址：14<sub>H</sub>

复位值：00000000\_00000000\_00000010\_00001000<sub>B</sub>

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

—	bit31-10	—	—
STKALIGN	bit9	R	非堆栈对齐标志位 读取始终为 1，指示异常入口 8 字节堆栈对齐
—	bit8-4	—	—
UNALIGN_TRP	bit3	R	字或半字访问操作的非对齐故障标志位

			读取始终为 1，指示非对齐访问产生硬故障
—	bit2-0	—	—

### 2.8.2.6 系统处理程序优先级寄存器 2 (SCB\_SHPR2)

系统处理程序优先级寄存器 2 (SCB\_SHPR2)

偏移地址: 1C<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_11<1:0>		保留													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															

PRI_11<1:0>	bit31-30	R/W	SVCall (异常编号 11) 的优先级设置位
—	bit29-0	—	—

### 2.8.2.7 系统处理程序优先级寄存器 3 (SCB\_SHPR3)

系统处理程序优先级寄存器 3 (SCB\_SHPR3)

偏移地址: 20<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_15<1:0>		保留						PRI_14<1:0>		保留					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															

PRI_15<1:0>	bit31-30	R/W	SysTick (异常编号 15) 的优先级设置位
—	bit29-24	—	—
PRI_14<1:0>	bit23-22	R/W	PendSV (异常编号 14) 的优先级设置位
—	bit21-0	—	—

## 2.9 系统定时器（SYSTICK）

### 2.9.1 概述

- ◇ 24 位系统递减计数器，递减至零可自动重载计数初值
- ◇ 可产生周期性 SysTick 异常，用作嵌入式操作系统的多任务调度计数器；或对于无嵌入式操作系统的运用，可用于调用需周期性执行的任务
- ◇ SysTick 亦可用作普通定时器，如用于延时计数
- ◇ SysTick 异常优先级可由系统处理优先级寄存器 SHPR3 的 PRI\_15<1:0>设定
- ◇ SysTick 异常处理的挂起可由中断控制和状态寄存器 SCB\_ICSR 的 PENDSTSET 位设置
- ◇ 工作时钟可为系统时钟 HCLK 或其三分频

SysTick 是一个系统递减计数器，配置 SYST\_RVR 寄存器，可设定计数初值。当 SysTick 计数为 0 时，COUNTFLAG 状态位置 1，并重载 SYST\_RVR 中的计数初值。在处理器调试停机时，SysTick 停止计数。在计数过程中，如果将 SYST\_RVR 寄存器设置为 0，则计数器递减计数到 0 后，停止计数。

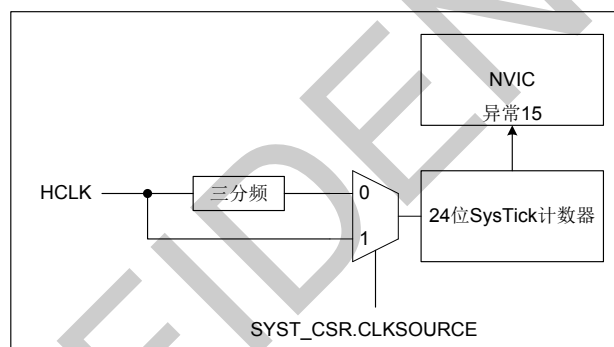


图 2-14 系统定时器框图

SysTick 的当前计数值可以通过读 SYST\_CVR 寄存器获得。如果写 SYST\_CVR 寄存器，则将该寄存器清零，并且将 COUNTFLAG 位清零，写操作不会触发 SysTick 异常事件。

访问 SysTick 寄存器时，需使用字操作方式。配置 SysTick 计数器的步骤如下：

- 1) 设置计数器重装值寄存器 SYST\_RVR。
- 2) 清除计数器当前值寄存器 SYST\_CVR。
- 3) 设置控制和状态寄存器 SYST\_CSR。

SysTick 寄存器列表和基址参见：3.5.1 系统定时器（SYSTICK）寄存器列表

## 2.9.2 特殊功能寄存器

### 2.9.2.1 SYSTICK控制和状态寄存器 (SYST\_CSR)

SYSTICK 控制和状态寄存器 (SYST\_CSR)

偏移地址: 10<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-17	—	—
COUNTFLAG	bit16	R	<b>SYSTICK 递减计数到零的标志位</b> 0: 未计数到 0 1: 计数到 0 该位读操作后清零, 或写 SYST_CVR 寄存器清零
—	bit15-3	—	—
CLKSOURCE	bit2	R/W	<b>SYSTICK 时钟源选择位</b> 0: 基准时钟 1: 处理器时钟
TICKINT	bit1	R/W	<b>SYSTICK 异常挂起使能位</b> 0: 计数到 0 时, 不产生异常挂起 1: 计数到 0 时, 产生异常挂起
ENABLE	bit0	R/W	<b>SYSTICK 计数器使能位</b> 0: 禁止 1: 使能

注 1: 处理器时钟为芯片内核工作时钟 HCLK, 时钟频率与系统时钟频率相同。

注 2: SYSTICK 基准时钟, 实际是处理器时钟 3 分频后的时钟, 频率为  $F_{HCLK}/3$ 。

### 2.9.2.2 SYSTICK重装值寄存器 (SYST\_RVR)

SYSTICK 重装值寄存器 (SYST\_RVR)

偏移地址: 14<sub>H</sub>复位值: 00000000\_11111111\_11111111\_11111111<sub>B</sub>

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

—	bit31-24	—	—
RELOAD<23:0>	bit23-0	R/W	<b>SYSTICK 计数器重载值</b> 计数范围 0x00_0001~0xFF_FFFF。如果为 0,

			SysTick 不计数。
--	--	--	--------------

### 2.9.2.3 SYSTICK当前值寄存器 (SYST\_CVR)

#### SYSTICK 当前值寄存器 (SYST\_CVR)

偏移地址: 18<sub>H</sub>

复位值: 00000000\_11111111\_11111111\_11111111<sub>B</sub>

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

—	bit31-24	—	—
CURRENT<23:0>	bit23-0	R/W	<b>SYSTICK 计数器当前值</b> 读取时返回 SysTick 计数器的当前值。 写入任何值都会将该寄存器清零，同时还会清零 COUNTFLAG 标志位。

### 2.9.2.4 SYSTICK校准值寄存器 (SYST\_CALIB)

#### SYSTICK 校准值寄存器 (SYST\_CALIB)

偏移地址: 1C<sub>H</sub>

复位值: 01000000\_00000010\_10001011\_00001010<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOREF	SKEW	保留						TENMS<23:16>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TENMS<15:0>															

NOREF	bit31	R	<b>基准时钟标志位</b> 0: 不提供外部基准时钟; 提供内部基准时钟, 其频率为 $F_{HCLK}/3$ 1: 提供外部基准时钟
SKEW	bit30	R	<b>TENMS 校准值是否准确的标志位</b> 0: TENMS 校准值准确 1: TENMS 校准值不准确
—	bit29-24	—	—
TENMS<23:0>	bit23-0	R/W	<b>SYSTICK 校准值</b> 读取为 0 时, 表示校准值未知

注: 本产品只提供内部基准时钟, 其频率为  $F_{HCLK}/3$ 。

## 2. 10 定时器（T16N/T32N）同步启动关停控制

### 2. 10. 1 概述

通过 SCU\_TIMEREN 和 SCU\_TIMERDIS 控制寄存器，可以选择性同时启动或关停多个 T16N/T32N 定时器。可用于对多个 TIMER 同时启动或关停，对于其它应用，仍然可使用各个 TIMER 自身的 T16N\_CON0 或 T32N\_CON0 寄存器的 EN 控制位来使能或关停 TIMER。

对各 TIMER 工作的控制，SCU\_TIMEREN 和 SCU\_TIMERDIS 控制寄存器的优先级高于 T16N\_CON0 和 T32N\_CON0 寄存器的 EN 控制位，并且 SCU\_TIMEREN 控制寄存器的优先级高于 SCU\_TIMERDIS。

### 2. 10. 2 特殊功能寄存器

#### 2. 10. 2. 1 SCU\_TIMEREN使能控制寄存器（SCU\_TIMEREN）

SCU\_TIMEREN 使能控制寄存器（SCU\_TIMEREN）

偏移地址：34<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							T32N0EN	保留				T16N3EN	T16N2EN	T16N1EN	T16N0EN

—	bit31-9	—	—
T32N0EN	bit8	R/W	<b>T32N0使能位</b> 0: — 1: 使能
—	bit7-4	—	—
T16N3EN	bit3	R/W	<b>T16N3使能位</b> 0: — 1: 使能
T16N2EN	bit2	R/W	<b>T16N2使能位</b> 0: — 1: 使能
T16N1EN	bit1	R/W	<b>T16N1使能位</b> 0: — 1: 使能
T16N0EN	bit0	R/W	<b>T16N0使能位</b> 0: — 1: 使能

注 1：对 SCU\_TIMEREN 寄存器的各位写 0 无效，写 1 使能后，硬件自动清零。

注 2：对 SCU\_TIMEREN 寄存器进行写操作前，需要设置 SCU\_PROT 寄存器，关闭写保护。

## 2.10.2.2 SCU\_TIMERDIS 关停控制寄存器 (SCU\_TIMERDIS)

### SCU\_TIMERDIS 使能控制寄存器 (SCU\_TIMERDIS)

偏移地址: 38<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							T32N0DIS	保留			T16N3DIS	T16N2DIS	T16N1DIS	T16N0DIS	

—	bit31-9	—	—
T32N0DIS	bit8	R/W	<b>T32N0</b> 关停位 0: — 1: 关停
—	bit7-4	—	—
T16N3DIS	bit3	R/W	<b>T16N3</b> 关停位 0: — 1: 关停
T16N2DIS	bit2	R/W	<b>T16N2</b> 关停位 0: — 1: 关停
T16N1DIS	bit1	R/W	<b>T16N1</b> 关停位 0: — 1: 关停
T16N0DIS	bit0	R/W	<b>T16N0</b> 关停位 0: — 1: 关停

注 1: 对 SCU\_TIMERDIS 寄存器的各位写 0 无效, 写 1 关停后, 硬件自动清零。

注 2: 对 SCU\_TIMERDIS 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。

## 第3章 存储器资源

### 3.1 内部存储器地址映射

芯片内部存储器包括程序存储器，数据存储器，外设寄存器和系统内核寄存器，各存储器区域的地址映射关系如下图所示，图中对系统内核寄存器区域的地址映射进行了详细描述。

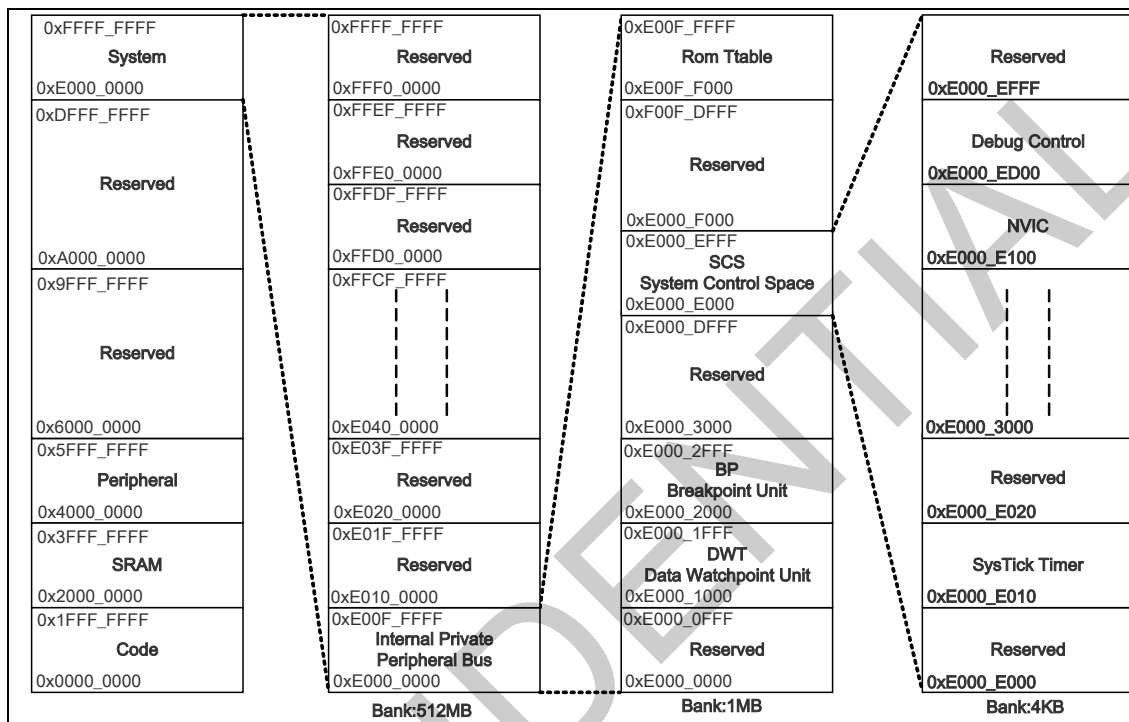


图 3-1 内部存储系统分配示意图

### 3.2 FLASH存储器

#### 3.2.1 信息区FLASH

芯片内部的信息区 FLASH 分为 3 个分区：INFO0 区、INFO1 区和 INFO4 区。

- ◇ INFO0 信息区用于存储芯片配置字 CFG\_WORD0, CFG\_WORD1, CFG\_WRP0, CFG\_WRP1, CFG\_DAFLS, CFG\_GBRDP
- ◇ INFO1 信息区用于存储芯片配置字 CFG\_PCROP0, CFG\_PCROP1
- ◇ INFO4 信息区包含 96 位芯片唯一识别码 UID
  - 芯片唯一识别码 UID 出厂时已固定，无法更改，程序只读
- ◇ 信息区基地址为 0x10000，用户程序只可读信息区数据，不能擦写。

##### 3.2.1.1 芯片配置字

芯片配置字位于 FLASH 存储器的 INFO0 信息区，用户可在 ISP 编程时进行设置。芯片的各种功能配置由芯片配置字和各功能相关寄存器共同设置完成。芯片配置字包括外部 XTAL 振荡器工作模式的选择、IWDW/WWDT 使能控制、BOR 电压选择等。

芯片配置字默认值仅表示为编程器界面的缺省设置，配置字地址单元在出厂时可能为非空，在 FLASH 编程之前必须先擦除，才能正确写入所设置的配置字。



寄存器名称	芯片配置字 0 (CFG_WORD0)	
地址偏移	0000 <sub>H</sub>	
低 16 位复位值	0001_1010_1000_1101 <sub>B</sub> (1A8D <sub>H</sub> )	
—	bit 31-16	<b>bit 15-0 取反值</b>
WWDTEN	bit15	<b>WWDT 窗口看门狗使能</b> 0: 软件使能后可再关闭 (默认) 1: 软件使能后无法再关闭
IWDTEN	bit 14	<b>IWDT 独立看门狗使能位</b> 0: 由软件使能 (默认) 1: 硬件强制使能 注: 硬件强制使能后, 软件无法关闭; 中断强制使能, 软件无法关闭; 复位强制使能, 软件无法关闭; 时钟源固定为 LRC, 软件无法切换。
IWDTINTEN	bit 13	<b>IWDT 独立看门狗中断使能位</b> (仅在 IWDTEN=1 时有效) 0: 禁止 (默认) 1: 使能
DEBUGEN	bit 12	<b>SWD 调试模式使能位</b> 0: 禁止 (调试完毕后需禁止, 避免管脚输入悬空) 1: DEBUG 自动识别 (默认)
BORVS	bit 11-10	<b>BOR 电压点选择位</b> 00: 保留 01: 2.5V 10: 2.1V (默认) 11: 3.1V
PWRTEN	bit 9	<b>上电 140ms 延时使能位</b> 0: 禁止 1: 使能 (默认)
XTAL	bit 8	<b>外部振荡器模式选择位</b> 0: 1~20MHz (高速、默认, 为 HS 或 XT 模式) 1: 32KHz (低速, 为 LP 模式) 该位应与 CFG_WORD1 中的 OSCMD 位对外部振荡器 HS, XT 和 LP 模式的选择保持一致。
CFG_MRSTN	bit 7	<b>MRSTN 管脚复用选择位</b> 0: GPIO 功能 1: MRSTN 功能 (默认)
—	bit 6-3	保留 (固定写 0001)
IWDTRL	bit2-0	<b>上电复位 IWDT 重载值选择位</b> (仅在 CFG_IWDTEN=1 时有效) 000: 0x0000_0200 (IWDT 计数溢出时间约 16ms) 001: 0x0000_0400 (IWDT 计数溢出时间约 32ms) 010: 0x0000_1000 (IWDT 计数溢出时间约 128ms) 011: 0x0000_4000 (IWDT 计数溢出时间约 512ms) 100: 0x0000_8000 (IWDT 计数溢出时间约 1s)

		101: 0x0001_0000 (IWDT 计数溢出时间约 2s) (默认)
		110: 0x0002_0000 (IWDT 计数溢出时间约 4s)
		111: 0x0004_0000 (IWDT 计数溢出时间约 8s)

注 1: 对配置位 CFG\_PWRSTEN, 仅在将 MRSTN 管脚作为外部复位 (CFG\_MRSTN=1) 时, 该配置位才有效, 推荐用户设置为上电延时使能 (CFG\_PWRSTEN=1), 只有在应用系统特别要求芯片上电后快速进入工作状态, 并且系统供电电源稳定可靠的条件下, 才可考虑禁止上电延时。

注 2: 当 MRSTN 管脚用作外部复位管脚时, 芯片内部固定集成了约 53K 欧姆的弱上拉电阻。

注 3: 对上电 140ms 延时, 在 MRSTN 管脚复用为 GPIO 功能时, 该延时固定为使能, 与配置位 CFG\_PWRSTEN 无关。

注 4: 芯片上电后系统时钟默认为内部 HRC 16MHz, 如果需要切换到外部 XTAL 振荡器, 则需要软件设置外部时钟振荡器使能位 XTAL\_EN=1 (SCU\_SCLKEN1<0>), 等外部时钟振荡稳定后, 再设置时钟源选择位 CLK\_SEL<1:0>=10 (SCU\_SCLKEN0<1:0>), 将 XTAL 作为时钟源。

注 5: 在对 Flash 程序加密编程时, 必须要禁止 CFG\_DEBUGEN 位, 否则加密无效; 调试完毕后, 也需禁止 CFG\_DEBUGEN 位, 并避免调试管脚输入悬空而产生漏电流, 影响芯片抗干扰性能等隐患。

注 6: SWD 调试模式使能后, 需避免在程序软件中设置对应的 IO 端口控制寄存器, 否则会导致调试异常。

注 7: 在 SWD 调试模式下, 需要禁止 IWDT 和 WWDT, 否则在调试过程中, IWDT 和 WWDT 会始终保持工作, 可能会产生计数溢出复位, 导致芯片调试异常。

寄存器名称	芯片配置字 1 (CFG_WORD1)	
地址偏移	0008 <sub>H</sub>	
低 16 位复位值	0000_0000_1100_0010 <sub>B</sub> (C2 <sub>H</sub> )	
—	bit 31-16	bit 15-0 取反值
—	bit 15-8	保留未用 (固定写 0)
OSCMD	bit 7-0	<b>XTAL 振荡器工作模式选择位</b> 0xC2: 高速 HS 模式 (5~20MHz) (默认) 0x84: 高速 XT 模式 (1~4MHz) 0x1F: 低速 LP 模式 (32KHz)

寄存器名称	写保护区域 x 配置字 (CFG_WRPx) (x=0..1)	
地址偏移	0020 <sub>H</sub> ~ 0028 <sub>H</sub>	
低 16 位复位值	0000_0000_0000_0001 <sub>B</sub> (0001 <sub>H</sub> )	
—	bit 31-16	bit 15-0 取反值
END	bit 15-11	<b>保护结束页配置位</b> 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 ..... 0x1F: Flash Page 127 注: 保护结束页数必须配置为大于或等于起始页数, 否则保护配置失效
—	bit 10-8	保留未用 (固定写 0)
START	bit 7-3	<b>保护起始页配置位</b>

		0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 ..... 0x1F: Flash Page 124
—	bit 2-1	保留未用 (固定写 0)
ENB	bit 0	保护使能位 0: 使能 1: 禁止 (默认)

注 1: Flash 每页 (page) 大小为 512 字节 (byte)。

注 2: 写保护区使能后, 位于写保护区的地址单元, 不支持 IAP 擦除和编程操作。

寄存器名称	数据 Flash 配置字 (CFG_DAFLS)	
地址偏移	0030 <sub>H</sub>	
低 16 位复位值	0000_0000_0000_0001 <sub>B</sub> (0001 <sub>H</sub> )	
—	bit 31-16	bit 15-0 取反值
END	bit 15-11	数据 Flash 结束页配置位 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 ..... 0x1F: Flash Page 127 注: 数据 Flash 结束页数必须配置为大于或等于起始页数, 否则数据 Flash 配置失效
—	bit 10-8	保留未用 (固定写 0)
START	bit 7-3	数据 Flash 起始页配置位 0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 ..... 0x1F: Flash Page 124
—	bit 2-1	保留未用 (固定写 0)
ENB	bit 0	数据 Flash 使能位 0: 使能 1: 禁止 (默认)

寄存器名称	全局读保护配置字 (CFG_GBRDP)	
地址偏移	0040 <sub>H</sub>	
复位值	0101_0101_1010_1010_0101_0101_1010_1010 <sub>B</sub> (55AA_55AA <sub>H</sub> )	
GBRDP	bit 31-0	全部读保护配置位 0xFFFF_FFFF: 读保护等级 Level 0 0xFFFF_XXXX: 读保护等级 Level 1 (XXXX 不为 FFFF)

		0xYYYY_XXXX: 读保护等级为 Level 2 (YYYY 不为 FFFF) (默认)
--	--	-------------------------------------------------

寄存器名称	私有代码读出保护区域 x 配置字 (CFG_PCROPx) (x=0、1)	
地址偏移	0200 <sub>H</sub> , 0208 <sub>H</sub>	
低 16 位复位值	0000_0000_0000_0001 <sub>B</sub> (0001 <sub>H</sub> )	
—	bit 31-16	bit 15-0 取反值 (不满足取反时 bit15-0 强制为默认值)
END	bit 15-11	保护结束页配置位 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 ..... 0x1F: Flash Page 127 注: 保护结束页数必须配置为大于或等于起始页数, 否则整个 Flash 程序区域都将处于非保护状态
—	bit 10-8	保留未用 (固定写 0)
START	bit 7-3	保护起始页配置位 0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 ..... 0x1F: Flash Page 124
—	bit 2-1	保留未用 (固定写 0)
ENB	bit 0	保护使能位 0: 使能 1: 禁止 (默认)

### 3.2.1.2 芯片唯一识别码UID

96 位芯片唯一识别码 UID 位于 FLASH 存储器的 INFO4 信息区, 以 word 为单位存放, 分为 3 个 words, 用户程序可读。UID2~UID0 各字节描述如下。

芯片唯一识别码 (UID11~UID0)		
地址	000109F0 <sub>H</sub> (UID2) 、 000109E8 <sub>H</sub> (UID1), 000109E0 <sub>H</sub> (UID0)	
UID2	bit 95~64	芯片唯一识别码 UID2
UID1	bit 63~32	芯片唯一识别码 UID1
UID0	bit 31~0	芯片唯一识别码 UID0

### 3.2.2 程序区FLASH

芯片内部的程序存储器 FLASH 总容量为 64K 字节，地址范围为 0000\_0000<sub>H</sub>~0000\_FFFF<sub>H</sub>，共分 128 页，每页 512 字节。FLASH 存储器支持至少 10 万次擦写次数，10 年以上的数据保持时间。

芯片支持通过 IAP 模块对程序存储器 FLASH 进行编程、页擦除操作，其中字地址单元编程时间约为 25us，页擦除时间约为 2ms。

芯片支持在 SWD 调试模式下对 FLASH 进行编程、擦除、读取等操作。

在进行 FLASH 编程时，无论是否编写相同的数据，在 FLASH 编程前均必须先进行擦除。

## 3.3 功能描述

### 3.3.1 IAP概述

- ◇ 支持 FLASH 数据保护，进行 IAP 操作前需先进行解锁，去除相关寄存器的写保护。
- ◇ 支持程序存储器 FLASH 全擦除模式（仅在 SWD 调试时有效）和页擦除模式。
- ◇ 支持字编程模式，每个字包含 4 个字节。
- ◇ IAP 操作过程中可软件禁止全局中断；也可使能中断，将中断向量表和中断服务程序（ISR）复制到 SRAM，通过设置中断向量表重映射使能寄存器 SCU\_TBLREMAPEN 和中断向量表偏移寄存器 SCU\_TBLOFF 可调用 SRAM 中的中断服务程序（ISR）来响应中断。
- ◇ IAP 自编程操作程序需放在芯片的 SRAM 中执行，并在程序中对 FLASH 擦除或编程结果进行校验。
- ◇ 芯片内置 IAP 自编程硬件固化模块，在 IAP 自编程操作程序中可以调用这些自编程固化模块，以减少 SRAM 中的 IAP 操作代码量。

### 3.3.2 Flash保护

#### 3.3.2.1 IAP操作保护KEY

软件通过写 IAP\_FLASHKEY 寄存器，可解除对程序区的保护，处于保护状态时，无法进行擦除和编程的操作。通过检查 IAP\_FLASHKEY.STATUS 是否为 0，判断 Flash 是否处于保护状态。

#### 3.3.2.2 Flash写保护区

Flash 存储器可以通过配置字 CFG\_WRP0 和 CFG\_WRP1 的 START、END 位配置两段写保护区域，通过 ENB 位配置两段写保护区域使能。

Flash 页擦除和 Flash 字编程，无法对写保护区擦除和写入，Flash 全擦时，可以将写保护区数据清除。

写保护区域	使能	起始页号	结束页号
区域 1	CFG_WRP0.ENB	CFG_WRP0.START	CFG_WRP0.END
区域 2	CFG_WRP1.ENB	CFG_WRP1.START	CFG_WRP1.END

### 3.3.2.3 Flash私有读保护区

Flash 存储器可以通过配置字 CFG\_PCROP0 和 CFG\_PCROP1 的 START、END 位配置两段私有读保护区，通过 ENB 位配置使能。

Flash 对私有读保护区进行任何非法的读取或擦写，均会置位对应的错误标识；但是可以进行 Flash 全擦操作，且 Flash 全擦时，可以将读保护区数据清除。

私有读保护区	使能	起始页号	结束页号
区域 1	CFG_PCROP0.ENB	CFG_PCROP0.START	CFG_PCROP0.END
区域 2	CFG_PCROP1.ENB	CFG_PCROP1.START	CFG_PCROP1.END

### 3.3.2.4 数据Flash区

Flash 区域可以通过配置字 CFG\_DAFLS 划分数据 Flash 区，通过 CFG\_DAFLS 的 ENB 位配置数据 Flash 的使能。

Data Flash	使能	起始页号	结束页号
区域 1	CFG_DAFLS.ENB	CFG_DAFLS.START	CFG_DAFLS.END

### 3.3.2.5 Flash全局读保护

Flash 存储器可以进行全局读保护，保护等级分为 Level0，Level1，Level2。

当全局保护字为 32 位全 1 时，全局保护级别即为 Level0。

当全局保护字高 16 位为全 1 且低 16 位为非全 1 时，全局保护级别即为 Level1。

当全局保护字高 16 位为非全 1 且低 16 位也为非全 1 时，全局保护级别即为 Level2。

不同全局加密保护级别下的访问限制如下表：

存储区		全局保护级别	调试模式 运行程序			用户模式					
						在 FLASH 中运行			在 SRAM 中运行		
			擦	写	读	擦	写	读	擦	写	读
Flash Code 区	非私有读保护区	Level0	全擦/页擦	是	是	NA	NA	是	页擦	是	是
		Level1	全擦/页擦	否	否	NA	NA	是	页擦	是	否
		Level2	否	否	否	NA	NA	是	页擦	是	否
	私有区保护区	Level0	全擦	否	是	NA	NA	否	否	否	否
		Level1	全擦	否	否	NA	NA	否	否	否	否
		Level2	否	否	否	NA	NA	否	否	否	否
	写保护区	Level0	全擦	是	是	NA	NA	是	否	否	是
		Level1	全擦	是	是	NA	NA	是	否	否	是
		Level2	否	否	否	NA	NA	是	否	否	是

注 1：调试模式下，在 SRAM 中运行程序时，若全局读保护等级为 Level1 或 Level2 时，不能读取 Flash Code 区。

注 2：用户模式下，在 Flash 中运行程序时，只有私有读保护区不能被读出，其他均可以被读出。

注 3：在 Flash 中运行程序时，禁止对 Flash 本身进行擦写操作，见上表标识 NA。

注 4：用户模式下，在 SRAM 或 IAPROM 中运行程序时，可以擦写 Flash Code 区中的非私有读保护区。

注 5：用户模式下，不支持对 Flash 的全擦和非私有读保护区全擦命令。

注 6：info 区在所有全局读保护等级下都为只读。

### 3.3.3 Flash程序区全擦除

程序区全擦除可擦除全部程序区空间，一次全擦除耗时约 8ms。具体步骤如下：

1. 查看 IAP\_FLASHSR.BUSY 标志是否处于空闲状态；
2. 通过 IAP\_FLASHKEY 解除 Flash 程序区保护状态；
3. 设置 Flash 操作请求使能；
4. 写入 IAP\_FLASHCMD.CMD 命令触发全擦除；
5. 等待 IAP\_FLASHSR.BUSY 标志再次变为空闲状态；
6. 判断 IAP\_FLASHSR.MASE 标志位是否置起；
7. 设置 Flash 操作请求禁止。

### 3.3.4 Flash非私有读保护区全擦除

擦除私有读保护区以外的程序区空间。具体步骤如下：

1. 查 IAP\_FLASHSR.BUSY 标志是否处于空闲状态；
2. 通过 IAP\_FLASHKEY 解除 Flash 程序区保护状态；
3. 设置 Flash 操作请求使能；
4. 写入程序区的首地址；
5. 写入 IAP\_FLASHCMD.CMD 命令触发全擦除；
6. 等待 IAP\_FLASHSR.BUSY 标志再次变为空闲状态；
7. 判定 IAP\_FLASHSR.UPCEBUSY 是否为空闲状态；
8. 判断 IAP\_FLASHSR.UPCEDONE 标志位是否置起；
9. 设置 Flash 操作请求禁止。

### 3.3.5 Flash页擦除

页擦除可擦除固定一页空间（512 Bytes），一次页擦除耗时约 2ms。具体步骤如下：

1. 检查 IAP\_FLASHSR.BUSY 标志是否处于空闲状态；
2. 通过 IAP\_FLASHKEY 解除 Flash 程序区保护状态；
3. 设置 Flash 操作请求使能；
4. 写入需擦除页的首地址；
5. 写入 IAP\_FLASHCMD.CMD 命令触发页擦除；
6. 等待 IAP\_FLASHSR.BUSY 标志再次变为空闲状态；
7. 判断 IAP\_FLASHSR.SERA 标志位是否置起；
8. 设置 Flash 操作请求禁止。

注：数据 Flash 页擦除流程与普通 Flash 页擦除流程一致，仅触发命令不同。

### 3.3.6 Flash字编程

程序区字编程可一次编程 4 Bytes 空间，一次字编程耗时约 25us。具体步骤如下：

1. 检查 IAP\_FLASHSR.BUSY 标志是否处于空闲状态；
2. 通过 IAP\_FLASHKEY 解除 Flash 程序区保护状态；
3. 设置 Flash 操作请求使能；
4. 写入需编程地址；
5. 写入需编程数据 IAP\_FLASHDR.DATA；
6. 写入 IAP\_FLASHCMD.CMD 命令触发字编程；
7. 等待 IAP\_FLASHSR.BUSY 标志再次变为空闲状态；
8. 判断 IAP\_FLASHSR.PROG 标志位是否置起；
9. 设置 Flash 操作请求禁止。

注：数据 Flash 字编程流程与普通 Flash 字编程流程一致，仅触发命令不同。

### 3.3.7 Flash编程数据FIFO

FLASH 编程数据 FIFO 可通过 FIFOEN 使能，该 FIFO 为写入 FIFO，读取无效。当数据写入 FIFO 后，可在 IAP\_FLASHDR 寄存器中体现。在 FIFO 中写入一次数据时，可触发一次编程。

### 3.3.8 IAP自编程硬件固化模块

芯片内置 IAP 自编程固化模块，由硬件电路实现，在 IAP 自编程操作程序中可以调用这些自编程固化模块，以减少 SRAM 中的 IAP 操作代码量。

IAP 自编程硬件固化模块支持页擦，单字编程，双字编程和多字编程，每次调用 IAP 操作函数之前，需要进行解锁操作。分别由如下 IAP 操作函数来实现：

#### 3.3.8.1 CODE区单页擦函数

- ◆ 函数功能：擦除 CODE 区指定的页
- ◆ 入口地址：0x10000004
- ◆ 输入参数：R0-擦除页的首地址，R1-擦除页首地址的反码，R2-固定写 0
- ◆ 返回值：R0-函数执行状态（R0=1 为成功，R0=0 为失败）

#### 3.3.8.2 CODE区单字编程函数

- ◆ 函数功能：向 FLASH CODE 区指定地址写入一个字(32-bits)
- ◆ 入口地址：0x10000008
- ◆ 输入参数：R0-待编程的 FLASH 地址，R1-待编程的 FLASH 地址的反码，R2-待编程数据



- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

#### 3.3.8.3 CODE区多字编程

- ◆ 函数功能：向 FLASH CODE 区指定地址写入多个字
- ◆ 入口地址：0x10000000
- ◆ 输入参数：R0-待编程的 FLASH 首地址，R1-待编程的 FLASH 首地址的反码，R2-放在 SRAM 空间的编程数据首地址，R3-编程数据长度（以字节为单位，必须为 4 的整数倍），R4-当编程到页首时是否先进行页擦除（R4≠0 为擦除，R4=0 为不擦除）
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

#### 3.3.8.4 DATA区单页擦函数

- ◆ 函数功能：擦除 DATA 区指定的页
- ◆ 入口地址：0x10000014
- ◆ 输入参数：R0-擦除页的首地址，R1-擦除页首地址的反码，R2-固定写 0
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

#### 3.3.8.5 DATA区单字编程函数

- ◆ 函数功能：向 FLASH DATA 区指定地址写入一个字(32-bits)
- ◆ 入口地址：0x10000018
- ◆ 输入参数：R0-待编程的 FLASH 地址，R1-待编程的 FLASH 地址的反码，R2-待编程数据
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

#### 3.3.8.6 DATA区多字编程

- ◆ 函数功能：向 FLASH DATA 区指定地址写入多个字
- ◆ 入口地址：0x10000010
- ◆ 输入参数：R0-待编程的 FLASH 首地址，R1-待编程的 FLASH 首地址的反码，R2-放在 SRAM 空间的编程数据首地址，R3-编程数据长度（以字节为单位，必须为 4 的整数倍），R4-当编程到页首时是否先进行页擦除（R4≠0 为擦除，R4=0 为不擦除）
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

注：在使用自编程硬件固化模块操作 IAP 前，需要解锁和关闭写保护。

### 3.3.9 特殊功能寄存器

#### 3.3.9.1 FLASH程序区关键码寄存器 (IAP\_FLASHKEY)

FLASH 程序区关键码寄存器 (IAP\_FLASHKEY)

偏移地址: 00<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000011<sub>B</sub>

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

Reserved	bit 31-2	—	保留
STATUS	bit 1-0	R	<b>FLASH 程序区状态位</b> 00: 可擦除或编程 其他: 被保护, 不可擦除或编程 IAP复位可将该寄存器复位

注: 对上述该寄存器连续写入 0x8ACE0246 和 0x9BDF1357 可去除保护, 写入其他值或中间插入其他操作将失效。

#### 3.3.9.2 FLASH擦除编程地址寄存器 (IAP\_FLASHADDR)

FLASH 擦除编程地址寄存器 (IAP\_FLASHADDR)

偏移地址: 08<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

Reserved	bit 31-16	—	保留
ADDR	bit 15-0	R/W	<b>FLASH 地址</b>

注: 关于上述寄存器中的 ADDR 位: 低 2 位写入无效, 读出始终为 0; 页擦除完成后, 地址自动加 0x200; 字编程完成后, 地址自动加 4。

### 3.3.9.3 FLASH编程数据FIFO寄存器 (IAP\_FLASHFIFO)

#### FLASH 编程数据 FIFO 寄存器 (IAP\_FLASHFIFO)

偏移地址: 0C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

FIFO<31:16>

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

FIFO<15:0>

FIFO	bit 31-0	W	FLASH编程FIFO
------	----------	---	-------------

注 1: 当写入相应个数数据后, 将自动触发字编程

### 3.3.9.4 FLASH编程数据寄存器 (IAP\_FLASHDR)

#### FLASH 编程数据寄存器 (IAP\_FLASHDR)

偏移地址: 10<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

DATA<31:16>

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

DATA<15:0>

DATA	bit 31-0	R/W	FLASH编程数据
------	----------	-----	-----------

### 3.3.9.5 FLASH操作命令寄存器 (IAP\_FLASHCMD)

#### FLASH 操作命令寄存器 (IAP\_FLASHCMD)

偏移地址: 18<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

CMD<31:16>

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

CMD<15:0>

CMD	bit 31-0	W	<b>FLASH编程命令</b> 0x000051AE: Flash Code区全擦除 0x00005EA1: 普通Flash页擦除 0x00005DA2: 普通Flash字编程 0x00005BA4: 数据Flash页擦除 0x00005AA5: 数据Flash字编程 0x000050AF: 非私有读保护区全擦除
-----	----------	---	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

			其他：保留
--	--	--	-------

注：私有保护区通过私有读出保护区配置字（CFG\_PCROPx）设置，如果未设置私有保护区，建议使用全擦除命令。

### 3.3.9.6 FLASH控制寄存器（IAP\_FLASHCR）

#### FLASH 控制寄存器（IAP\_FLASHCR）

偏移地址：1C<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>b</sub>

3	3	2	28	27	26	25	24	23	22	21	20	19	18	17	16
1	0	9													
保留															
1	1	1	12	11	10	9	8	7	6	5	4	3	2	1	0
5	4	3													
保留			PGSZ	FIFODF	保留	FIFOEN	FLASHREQ	保留	OTFEN	IAPRST	IAPEN				

Reserved	bit 31-9	—	保留
PGSZ	bit 8	R/W	页擦除区域大小选择 0：512B（需软件固定写0） 1：保留，仅作内部测试用
FIFODF	bit 7	R/W	<b>FIFO 编程数据 Flash 使能</b> 0：禁止 1：使能
Reserved	bit 6	—	保留
FIFOEN	bit 5	R/W	<b>FIFO 使能</b> 0：禁止 1：使能
FLASHREQ	bit 4	R/W	<b>FLASH 操作请求使能</b> 0：禁止 1：使能
Reserved	bit 3	—	保留
OTFEN	bit 2	R/W	<b>自编程操作 FLASH 运行使能</b> 0：禁止 1：使能
IAPRST	bit 1	W1	<b>自编程复位</b> 0：无操作 1：自编程复位
IAPEN	bit 0	R/W	<b>自编程使能</b> 0：禁止 1：使能

## 3.3.9.7 FLASH状态寄存器 (IAP\_FLASHSR)

FLASH 状态寄存器 (IAP\_FLASHSR)

偏移地址: 20<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留						UPC EUL	MAS EUL	ADDE RR	保留						UPCE DONE	UPCEBU SY
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留								TIME OUT	PROG	SERA	MASE	WAE	WPE	BUSY	FLASHA CK	

Reserved	bit 31-26	—	保留
UPCEUL	bit 25	R	非私有读保护区全擦保护解除位 0: 未解除 1: 已解除 解除保护后, 允许对私有读保护配置区进行擦除
MASEUL	bit 24	R	程序区全擦保护解除位 0: 未解除 1: 已解除 解除保护后, 允许对对用户配置区和私有读保护配置区进行擦除
ADDERR	bit 23	R	地址反码错误标识位 0: 反码正确 1: 反码错误
Reserved	bit 22-18	—	保留
UPCEDONE	bit 17	R	非私有读保护区全擦完成标志 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
UPCEBUSY	bit 16	R	非私有读保护区全擦状态位 0: 空闲 1: 正在进行
Reserved	bit 15-8	—	保留
TIMEOUT	bit 7	R	超时错误标志 0: 无错误 1: 发生错误 未在规定时间内完成相应擦除或编程动作时产生错误标志, 可能硬件发生了故障, 需软件触发一次 IAP 复位
PROG	bit 6	R	字编程完成标志 0: 未进行或正在进行中 1: 已完成

			重新启动新的擦除或编程操作时自动清除
SERA	bit 5	R	<b>页擦除完成标志</b> 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
MASE	bit 4	R	<b>程序区全擦除完成标志</b> 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
WAE	bit 3	R	<b>擦写地址错误标志</b> 0: 无错误 1: 发生错误 可能是在IAP操作在非法的FLASH地址, 或是在擦除和编程时使用了错误的命令, 需软件触发一次IAP复位
WPE	bit 2	R	<b>擦写保护错误标志</b> 0: 无错误 1: 发生错误 触发了保护区的擦除或编程, 操作失败, 需软件触发一次IAP复位
BUSY	bit 1	R	<b>自编程状态复位</b> 0: 空闲 1: 正在进行
FLASHACK	bit 0	R	<b>FLASH 操作许可状态</b> 0: 禁止操作 1: 允许操作

### 3.3.9.8 FLASH擦除编程地址反码寄存器 (IAP\_FLASHADDINV)

FLASH 擦除编程地址反码寄存器 (IAP\_FLASHADDINV)

偏移地址: 2C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

Reserved	bit 31-16	—	保留
ADDRINV	bit 15-0	R/W	FLASH 地址反码

3.4 数据存储器（SRAM）

芯片内部集成 4K 字节数据存储器 SRAM，地址范围为 2000\_0000<sub>H</sub>~2000\_0FFF<sub>H</sub>。

3.4.1 SRAM地址映射

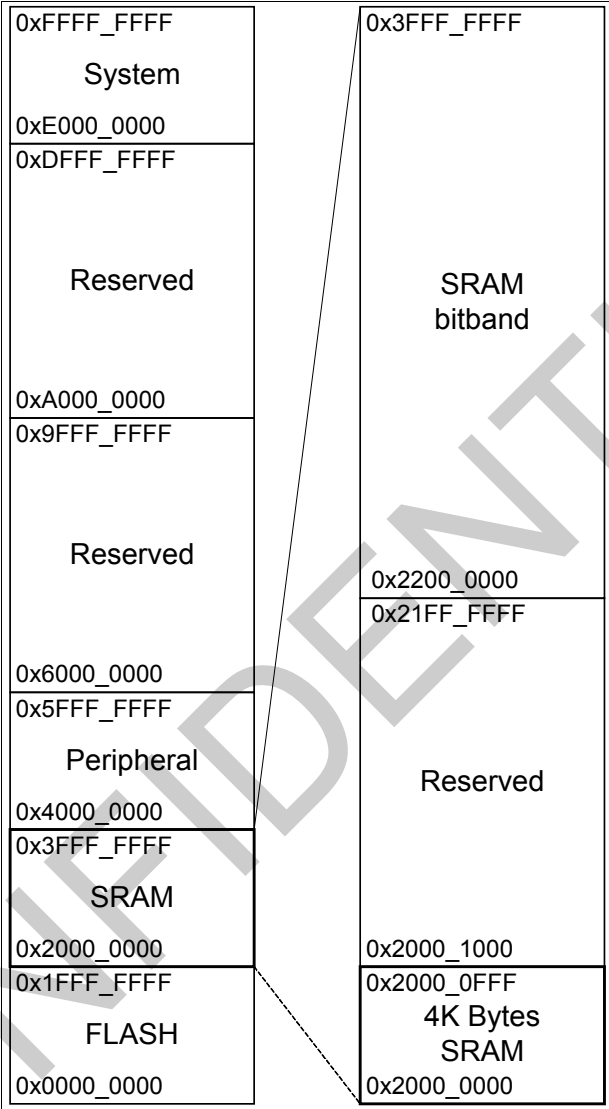


图 3-2 SRAM 映射图

3.4.2 SRAM位带扩展

SRAM 支持位带扩展，可使用普通的加载和存储指令对单比特进行读写操作。通过位带扩展，除可在起始地址为 0x2000\_0000 的空间访问 SRAM 外，还可在起始地址为 0x2200\_0000 的位带扩展区以单比特方式访问 SRAM。

位带扩展区把每个比特扩展为一个 32-bit 的字，通过访问该区域的字可达到访问原始比特的目的。对于 SRAM 的某个 bit，如果它所在字节地址为 A，位序号为 N（0≤N≤7），则该 bit 在 SRAM 位带扩展后的地址为：

$$\text{AliasAddress\_A\_N} = 0x2200\_0000 + (A - 0x2000\_0000) \times 32 + N \times 4$$

3.5 外设寄存器

3.5.1 外设寄存器映射

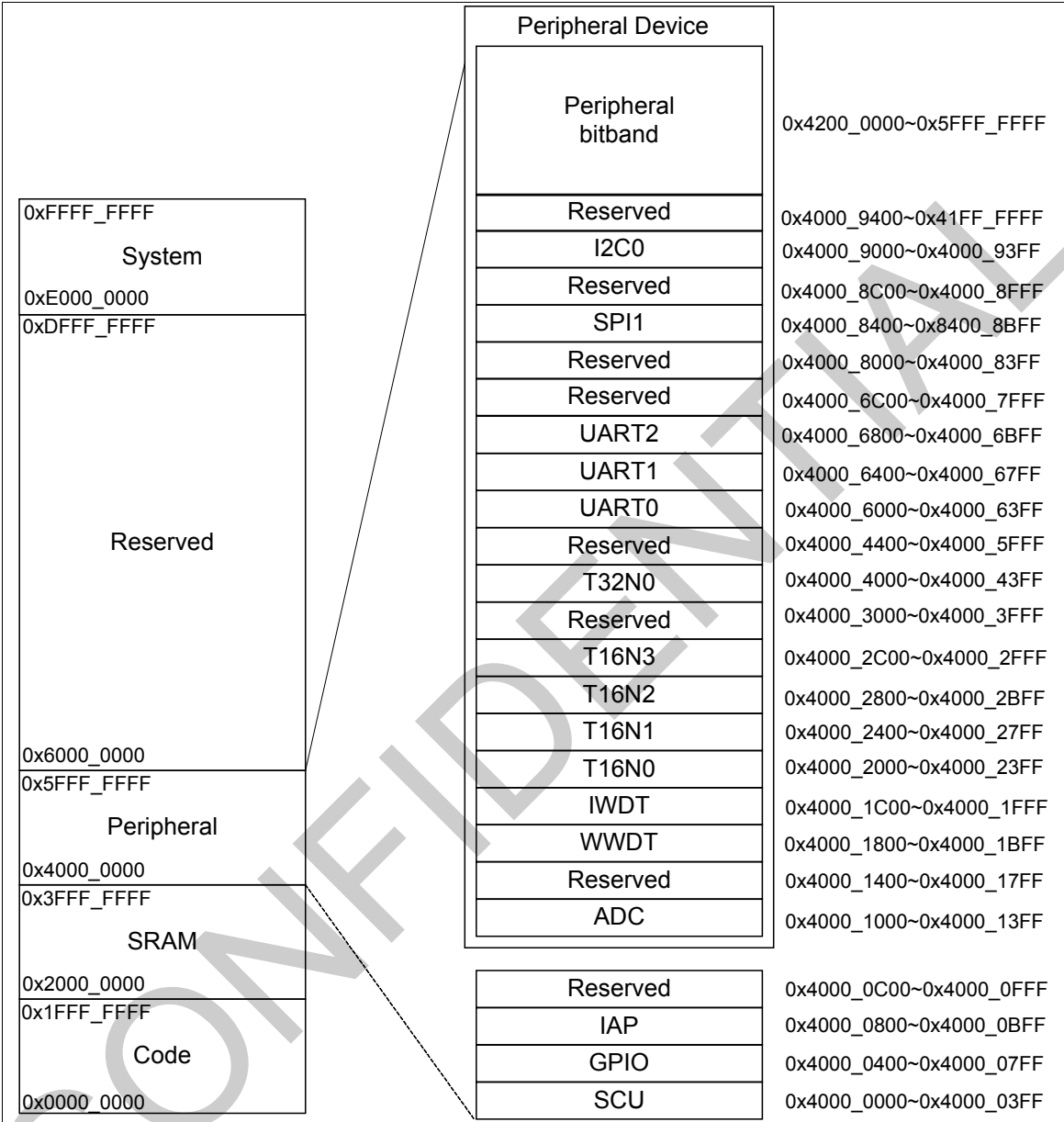


图 3-3 外设存储器分配示意图

注：Reserved 保留寄存器区域为只读，读出值为 0000\_0000<sub>H</sub>。

3.5.2 外设寄存器位带扩展

外设寄存器支持位带扩展，可使用普通的加载和存储指令对单比特进行读写操作。通过位带扩展，除可在 0x4000\_0000 起始的空间访问外设外，还可在起始地址为 0x4200\_0000 的位带扩展区以单比特方式访问外设寄存器。

位带扩展区把每个比特扩展为一个 32-bit 的字，通过访问这些字可达到访问原始比特的目的。对于外设寄存器的某一位，如果它所在字节地址为 A，位序号为 N (0≤N≤7)，则该位在外设寄存器位带扩展后的地址为：



$$\text{AliasAddress\_A\_N} = 0x4200\_0000 + (A - 0x4000\_0000) \times 32 + N \times 4$$

GPIO 端口寄存器 GPIO\_PADATABSR, GPIO\_PADATABCR, GPIO\_PADATABRR, GPIO\_PADIRBSR, GPIO\_PADIRBCR, GPIO\_PADIRBRR, GPIO\_PBDATABSR, GPIO\_PBDATABCR, GPIO\_PBDATABRR, GPIO\_PBDIRBSR, GPIO\_PBDIRBCR, GPIO\_PBDIRBRR, 用于实现对 GPIO 端口数据寄存器和方向控制寄存器的位操作, 这些端口位操作寄存器不再支持位带扩展访问; ADC 控制寄存器 ADC\_CON0 不支持位带扩展访问。除上述寄存器外, 其它外设寄存器均支持位带扩展访问操作。

### 3.5.3 系统控制单元 (SCU) 寄存器列表

系统控制单元 (SCU)		
寄存器名称	寄存器偏移地址	寄存器描述
SCU 基地址: 4000_0000 <sub>H</sub>		
SCU_PROT	0000 <sub>H</sub>	系统设置保护寄存器
SCU_NMICON	0004 <sub>H</sub>	不可屏蔽中断控制寄存器
SCU_PWRC	0008 <sub>H</sub>	复位寄存器
SCU_FAULTFLAG	000C <sub>H</sub>	硬件错误标志寄存器
SCU_FLASHWAIT	0020 <sub>H</sub>	FLASH 访问等待时间寄存器
SCU_LVDCON	0028 <sub>H</sub>	LVD 控制寄存器
SCU_CCM	002C <sub>H</sub>	外部时钟检测控制寄存器
SCU_TIMEREN	0034 <sub>H</sub>	TIMER 使能控制寄存器
SCU_TIMERDIS	0038 <sub>H</sub>	TIMER 关停控制寄存器
SCU_SCLKEN0	0040 <sub>H</sub>	系统时钟控制寄存器 0
SCU_SCLKEN1	0044 <sub>H</sub>	系统时钟控制寄存器 1
SCU_PCLKEN	0048 <sub>H</sub>	外设时钟控制寄存器
SCU_WAKEUPTIME	004C <sub>H</sub>	系统唤醒时间控制寄存器
SCU_TBLREMAPEN	0060 <sub>H</sub>	中断向量表重映射使能寄存器
SCU_TBLOFF	0064 <sub>H</sub>	中断向量表偏移寄存器

### 3.5.4 GPIO寄存器列表

GPIO 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
GPIO 基地址: 4000_0400 <sub>H</sub>		
GPIO_PAPORT	0000 <sub>H</sub>	PA 端口状态寄存器
GPIO_PADATA	0004 <sub>H</sub>	PA 端口数据寄存器
GPIO_PADATABSR	0008 <sub>H</sub>	PA 端口输出置位寄存器
GPIO_PADATABCR	000C <sub>H</sub>	PA 端口输出清零寄存器
GPIO_PADATABRR	0010 <sub>H</sub>	PA 端口输出翻转寄存器
GPIO_PADIR	0014 <sub>H</sub>	PA 端口方向控制寄存器
GPIO_PADIRBSR	0018 <sub>H</sub>	PA 端口方向置位寄存器.
GPIO_PADIRBCR	001C <sub>H</sub>	PA 端口方向清零寄存器
GPIO_PADIRBRR	0020 <sub>H</sub>	PA 端口方向翻转寄存器
GPIO_PAFUNC0	0024 <sub>H</sub>	PA<7:0>端口复用选择寄存器

GPIO 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
GPIO_PAFUNC1	0028 <sub>H</sub>	PA<15:8>端口复用选择寄存器
GPIO_PAFUNC2	002C <sub>H</sub>	PA<23:16>端口复用选择寄存器
GPIO_PAFUNC3	0030 <sub>H</sub>	PA<31:24>端口复用选择寄存器
GPIO_PAINEB	0034 <sub>H</sub>	PA 端口输入控制寄存器
GPIO_PAODE	0038 <sub>H</sub>	PA 端口开漏控制寄存器
GPIO_PAPUE	003C <sub>H</sub>	PA 端口弱上拉使能寄存器
GPIO_PAPDE	0040 <sub>H</sub>	PA 端口弱下拉使能寄存器
GPIO_PADS	0044 <sub>H</sub>	PA 端口驱动电流控制寄存器
GPIO_PATYP	0048 <sub>H</sub>	PA 端口类型选择寄存器
GPIO_PAFLT	004C <sub>H</sub>	PA 端口滤波控制寄存器
GPIO_PBPORT	0080 <sub>H</sub>	PB 端口状态寄存器
GPIO_PBDATA	0084 <sub>H</sub>	PB 端口数据寄存器
GPIO_PBDATABSR	0088 <sub>H</sub>	PB 端口输出置位寄存器
GPIO_PBDATABCR	008C <sub>H</sub>	PB 端口输出清零寄存器
GPIO_PBDATABRR	0090 <sub>H</sub>	PB 端口输出翻转寄存器
GPIO_PBDIR	0094 <sub>H</sub>	PB 端口方向控制寄存器
GPIO_PBDIRBSR	0098 <sub>H</sub>	PB 端口方向置位寄存器
GPIO_PBDIRBCR	009C <sub>H</sub>	PB 端口方向清零寄存器
GPIO_PBDIRBRR	00A0 <sub>H</sub>	PB 端口方向翻转寄存器
GPIO_PBFUNC0	00A4 <sub>H</sub>	PB<7:0>端口复用选择寄存器
GPIO_PBFUNC1	00A8 <sub>H</sub>	PB<13:8>端口复用选择寄存器
GPIO_PBINEB	00B4 <sub>H</sub>	PB 端口输入控制寄存器
GPIO_PBODE	00B8 <sub>H</sub>	PB 端口开漏控制寄存器
GPIO_PBPUE	00BC <sub>H</sub>	PB 端口弱上拉使能寄存器
GPIO_PBPDE	00C0 <sub>H</sub>	PB 端口弱下拉使能寄存器
GPIO_PBDS	00C4 <sub>H</sub>	PB 端口驱动电流控制寄存器
GPIO_PBTYP	00C8 <sub>H</sub>	PB 端口类型选择寄存器
GPIO_PBFLT	00CC <sub>H</sub>	PB 端口滤波控制寄存器
GPIO_PINTIE	0300 <sub>H</sub>	PINT 中断使能寄存器
GPIO_PINTIF	0304 <sub>H</sub>	PINT 中断标志寄存器
GPIO_PINTSEL	0308 <sub>H</sub>	PINT 中断源选择寄存器
GPIO_PINTCFG	030C <sub>H</sub>	PINT 中断配置寄存器
GPIO_KINTIE	0310 <sub>H</sub>	KINT 中断使能寄存器
GPIO_KINTIF	0314 <sub>H</sub>	KINT 中断标志寄存器
GPIO_KINTSEL	0318 <sub>H</sub>	KINT 中断源选择寄存器
GPIO_KINTCFG	031C <sub>H</sub>	KINT 中断配置寄存器
GPIO_TXPWM	0380 <sub>H</sub>	脉宽调制寄存器
GPIO_BUZC	0384 <sub>H</sub>	蜂鸣器计数控制寄存器

### 3.5.5 IAP寄存器列表

IAP 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
IAP 基地址: 4000_0800 <sub>H</sub>		
IAP_FLASHKEY	000 <sub>H</sub>	FLASH 程序区操作关键码寄存器
—	004 <sub>H</sub>	保留
IAP_FLASHADDR	008 <sub>H</sub>	FLASH 擦除编程地址寄存器
IAP_FLASHFIFO	00C <sub>H</sub>	FLASH 编程数据 FIFO 寄存器
IAP_FLASHDR	010 <sub>H</sub>	FLASH 编程数据寄存器
—	014 <sub>H</sub>	保留
IAP_FLASHCMD	018 <sub>H</sub>	FLASH 操作命令寄存器
IAP_FLASHCR	01C <sub>H</sub>	FLASH 控制寄存器
IAP_FLASHSR	020 <sub>H</sub>	FLASH 状态寄存器
—	024 <sub>H</sub>	保留
—	028 <sub>H</sub>	保留
IAP_FLASHADDINV	02C <sub>H</sub>	FLASH 擦除编程地址反码寄存器

### 3.5.6 ADC寄存器列表

ADC 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
ADC 基地址: 4000_1000 <sub>H</sub>		
ADC_DR	0000 <sub>H</sub>	ADC 转换值寄存器
ADC_CON0	0004 <sub>H</sub>	ADC 控制寄存器 0
ADC_CON1	0008 <sub>H</sub>	ADC 控制寄存器 1
ADC_CHS	000C <sub>H</sub>	ADC 通道选择寄存器
ADC_IE	0010 <sub>H</sub>	ADC 中断使能寄存器
ADC_IF	0014 <sub>H</sub>	ADC 中断标志寄存器
ADC_ACPC	0028 <sub>H</sub>	ADC 自动转换比较控制寄存器
ADC_ACPCMP	0030 <sub>H</sub>	ADC 自动转换比较阈值寄存器
ADC_ACPMEAN	0034 <sub>H</sub>	ADC 自动转换均值数据寄存器
	0038 <sub>H</sub>	
ADC_VREFCON	0040 <sub>H</sub>	ADC 参考控制寄存器

### 3.5.7 IWDt寄存器列表

IWDt 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
IWDt 基地址: 4000_1C00 <sub>H</sub>		
IWDt_LOAD	0000 <sub>H</sub>	IWDt 计数器装载值寄存器
IWDt_VALUE	0004 <sub>H</sub>	IWDt 计数器当前值寄存器
IWDt_CON	0008 <sub>H</sub>	IWDt 控制寄存器
IWDt_INTCLR	000C <sub>H</sub>	IWDt 中断标志清除寄存器

IWDT 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
IWDT_RIS	0010 <sub>H</sub>	IWDT 中断标志寄存器
IWDT_LOCK	0100 <sub>H</sub>	IWDT 访问使能寄存器

### 3.5.8 WWDT寄存器列表

WWDT 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
WWDT 基地址: 4000_1800 <sub>H</sub>		
WWDT_LOAD	0000 <sub>H</sub>	WWDT 计数器装载值寄存器
WWDT_VALUE	0004 <sub>H</sub>	WWDT 计数器当前值寄存器
WWDT_CON	0008 <sub>H</sub>	WWDT 控制寄存器
WWDT_INTCLR	000C <sub>H</sub>	WWDT 中断标志清除寄存器
WWDT_RIS	0010 <sub>H</sub>	WWDT 中断标志寄存器
WWDT_LOCK	0100 <sub>H</sub>	WWDT 锁定寄存器

### 3.5.9 T16N0/T16N1/T16N2/T16N3 寄存器列表

T16N 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
T16N0 基地址: 4000_2000 <sub>H</sub>		
T16N1 基地址: 4000_2400 <sub>H</sub>		
T16N2 基地址: 4000_2800 <sub>H</sub>		
T16N3 基地址: 4000_2C00 <sub>H</sub>		
T16N_CNT0	0000 <sub>H</sub>	T16N 计数值寄存器 0
T16N_CNT1	0004 <sub>H</sub>	T16N 计数值寄存器 1
T16N_PRECNT	0008 <sub>H</sub>	T16N 预分频器计数值寄存器
T16N_PREMAT	000C <sub>H</sub>	T16N 预分频器计数匹配寄存器
T16N_CON0	0010 <sub>H</sub>	T16N 控制寄存器 0
T16N_CON1	0014 <sub>H</sub>	T16N 控制寄存器 1
T16N_CON2	0018 <sub>H</sub>	T16N 控制寄存器 2
T16N_IE	0020 <sub>H</sub>	T16N 中断使能寄存器
T16N_IF	0024 <sub>H</sub>	T16N 中断标志寄存器
T16N_PDZ	0028 <sub>H</sub>	T16N 调制模式死区宽度寄存器
T16N_PTR	002C <sub>H</sub>	T16N 调制模式 ADC 触发寄存器
T16N_MAT0	0030 <sub>H</sub>	T16N 计数匹配寄存器 0
T16N_MAT1	0034 <sub>H</sub>	T16N 计数匹配寄存器 1
T16N_MAT2	0038 <sub>H</sub>	T16N 计数匹配寄存器 2
T16N_MAT3	003C <sub>H</sub>	T16N 计数匹配寄存器 3
T16N_TOP0	0040 <sub>H</sub>	T16NCNT0 计数峰值寄存器 0
T16N_TOP1	0044 <sub>H</sub>	T16NCNT1 计数峰值寄存器 1

## 3.5.10 T32N0 寄存器列表

T32N 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
T32N0 基地址: 4000_4000 <sub>H</sub>		
T32N_CNT	0000 <sub>H</sub>	T32N 计数值寄存器
T32N_CON0	0004 <sub>H</sub>	T32N 控制寄存器 0
T32N_CON1	0008 <sub>H</sub>	T32N 控制寄存器 1
T32N_PRECNT	0010 <sub>H</sub>	T32N 预分频器计数值寄存器
T32N_PREMAT	0014 <sub>H</sub>	T32N 预分频器计数匹配寄存器
T32N_IE	0018 <sub>H</sub>	T32N 中断使能寄存器
T32N_IF	001C <sub>H</sub>	T32N 中断标志寄存器
T32N_MAT0	0020 <sub>H</sub>	T32N 计数匹配寄存器 0
T32N_MAT1	0024 <sub>H</sub>	T32N 计数匹配寄存器 1
T32N_MAT2	0028 <sub>H</sub>	T32N 计数匹配寄存器 2
T32N_MAT3	002C <sub>H</sub>	T32N 计数匹配寄存器 3

## 3.5.11 UART0/UART1/UART2 寄存器列表

UART 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
UART0 基地址: 4000_6000 <sub>H</sub>		
UART1 基地址: 4000_6400 <sub>H</sub>		
UART2 基地址: 4000_6800 <sub>H</sub>		
UART_CON0	0000 <sub>H</sub>	UART 控制寄存器 0
UART_CON1	0004 <sub>H</sub>	UART 控制寄存器 1
UART_BRR	0010 <sub>H</sub>	UART 波特率寄存器
UART_STA	0014 <sub>H</sub>	UART 状态寄存器
UART_IE	0018 <sub>H</sub>	UART 中断使能寄存器
UART_IF	001C <sub>H</sub>	UART 中断标志寄存器
UART_TBW	0020 <sub>H</sub>	UART 发送数据写入寄存器
UART_RBR	0024 <sub>H</sub>	UART 接收数据读取寄存器
UART_TB0	0040 <sub>H</sub>	UART 发送缓冲寄存器 0
UART_RB0	0060 <sub>H</sub>	UART 接收缓冲寄存器 0

## 3.5.12 SPI1 寄存器列表

SPI 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
SPI1 基地址: 4000_8400 <sub>H</sub>		
SPI_CON	0000 <sub>H</sub>	SPI 控制寄存器
SPI_TBW	0008 <sub>H</sub>	SPI 发送数据写入寄存器
SPI_RBR	000C <sub>H</sub>	SPI 接收数据读取寄存器
SPI_IE	0010 <sub>H</sub>	SPI 中断使能寄存器

SPI 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
SPI_IF	0014 <sub>H</sub>	SPI 中断标志寄存器
SPI_TB	0018 <sub>H</sub>	SPI 发送缓冲寄存器
SPI_RB	001C <sub>H</sub>	SPI 接收缓冲寄存器
SPI_STA	0020 <sub>H</sub>	SPI 状态寄存器
SPI_CKS	0024 <sub>H</sub>	SPI 波特率设置寄存器

### 3.5.13 I2C0 寄存器列表

I2C 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
I2C0 基地址: 4000_9000 <sub>H</sub>		
I2C_CON	0000 <sub>H</sub>	I2C 控制寄存器
I2C_MOD	0004 <sub>H</sub>	I2C 工作模式寄存器
I2C_IE	0008 <sub>H</sub>	I2C 中断使能寄存器
I2C_IF	000C <sub>H</sub>	I2C 中断标志寄存器
I2C_TBW	0010 <sub>H</sub>	I2C 发送数据写入寄存器
I2C_RBR	0014 <sub>H</sub>	I2C 接收数据读取寄存器
I2C_TB	0018 <sub>H</sub>	I2C 发送缓冲寄存器
I2C_RB	001C <sub>H</sub>	I2C 接收缓冲寄存器
I2C_STA	0020 <sub>H</sub>	I2C 状态寄存器

## 3.6 内核寄存器

### 3.6.1 系统定时器（SYSTICK）寄存器列表

系统定时器（SYSTICK）		
寄存器名称	寄存器偏移地址	寄存器描述
SYSTICK 基地址: E000_E000 <sub>H</sub>		
SYST_CSR	0010 <sub>H</sub>	SYSTICK 控制/状态寄存器
SYST_RVR	0014 <sub>H</sub>	SYSTICK 重装值寄存器
SYST_CVR	0018 <sub>H</sub>	SYSTICK 当前值寄存器
SYST_CALIB	001C <sub>H</sub>	SYSTICK 校准值寄存器

### 3.6.2 中断控制器（NVIC）寄存器列表

中断控制器（NVIC）		
寄存器名称	寄存器偏移地址	寄存器描述
NVIC 基地址: E000_E100 <sub>H</sub>		
NVIC_ISER	0000 <sub>H</sub>	IRQ0~31 置中断请求使能寄存器
NVIC_ICER	0080 <sub>H</sub>	IRQ0~31 清中断请求使能寄存器
NVIC_ISPR	0100 <sub>H</sub>	IRQ0~31 置中断挂起寄存器
NVIC_ICPR	0180 <sub>H</sub>	IRQ0~31 清中断挂起寄存器

中断控制器 (NVIC)		
寄存器名称	寄存器偏移地址	寄存器描述
NVIC_PR0	0300 <sub>H</sub>	IRQ0~3 优先级控制寄存器
NVIC_PR1	0304 <sub>H</sub>	IRQ4~7 优先级控制寄存器
NVIC_PR2	0308 <sub>H</sub>	IRQ8~11 优先级控制寄存器
NVIC_PR3	030C <sub>H</sub>	IRQ12~15 优先级控制寄存器
NVIC_PR4	0310 <sub>H</sub>	IRQ16~19 优先级控制寄存器
NVIC_PR5	0314 <sub>H</sub>	IRQ20~23 优先级控制寄存器
NVIC_PR6	0318 <sub>H</sub>	IRQ24~27 优先级控制寄存器
NVIC_PR7	031C <sub>H</sub>	IRQ28~31 优先级控制寄存器

### 3.6.3 系统控制块 (SCB) 寄存器列表

系统控制块 (SCB)		
寄存器名称	寄存器偏移地址	寄存器描述
SCB 基地址: E000_ED00 <sub>H</sub>		
SCB_CPUID	0000 <sub>H</sub>	SCB_CPUID 寄存器
SCB_ICSR	0004 <sub>H</sub>	中断控制和状态寄存器
SCB_AIRCR	000C <sub>H</sub>	应用中断和复位控制寄存器
SCB_SCR	0010 <sub>H</sub>	系统控制寄存器
SCB_CCR	0014 <sub>H</sub>	配置和控制寄存器
SCB_SHPR2	001C <sub>H</sub>	系统处理程序优先级寄存器 2
SCB_SHPR3	0020 <sub>H</sub>	系统处理程序优先级寄存器 3

## 第4章 输入输出端口（GPIO）

### 4.1 概述

本芯片支持两组 GPIO 端口，最多支持共 30 个 I/O 管脚。

所有 I/O 端口都支持 CMOS 或 TLL 输入类型可选择，CMOS 输出驱动（可配置为开漏输出），每个 I/O 端口的复用功能和工作模式由端口复用选择寄存器 GPIO\_PAFUNC/GPIO\_PBFUNC 配置。

当 I/O 端口配置为通用数字 I/O 功能时，其输出状态由端口方向控制寄存器 GPIO\_PADIR/GPIO\_PBDIR 配置，输入状态由相应的端口输入控制寄存器 GPIO\_PAINEB/GPIO\_PBINEB 配置。当 I/O 端口处于输出状态时，其电平由端口数据寄存器 GPIO\_PADATA/GPIO\_PBDATA 决定，1 为高电平，0 为低电平；当 I/O 端口处于输入状态时，其电平状态可通过读取端口状态寄存器 GPIO\_PAPORT/GPIO\_PBPORT 获得。

端口输出电平支持位操作。将 GPIO 输出置位寄存器 GPIO\_PADATABSR/GPIO\_PBATABSR 相应位写 1，可将相应位的 GPIO 端口设置为高电平；将 GPIO 端口输出清零寄存器 GPIO\_PADATABCR / GPIO\_PBATABCR 相应位写 1，可将相应位的 GPIO 端口设置为低电平；将 GPIO 端口输出翻转寄存器 GPIO\_PADATABRR/GPIO\_PBATABRR 相应位写 1，可将相应位的 GPIO 端口电平取反。

端口方向控制支持位操作。将 GPIO 端口方向置位寄存器 GPIO\_PADIRBSR/GPIO\_PBDIRBSR 相应位写 1，可将相应位的 GPIO 端口设置为输入；将 GPIO 端口方向清零寄存器 GPIO\_PADIRBCR/ GPIO\_PBDIRBCR 相应位写 1，可将相应位的 GPIO 端口设置为输出；将 GPIO 端口方向翻转寄存器 GPIO\_PADIRBRR/GPIO\_PBDIRBRR 相应位写 1，可将相应位的 GPIO 端口方向取反。

当 I/O 端口配置为复用功能时，作为芯片外设功能模块的复用端口，其输出状态仍需通过端口方向控制寄存器 GPIO\_PADIR/GPIO\_PBDIR 进行配置，输入状态仍需通过相应的端口输入控制寄存器 GPIO\_PAINEB/GPIO\_PBINEB 进行配置。

每个 I/O 端口均支持 CMOS 或 TLL 输入类型可选择，由相应的端口类型选择寄存器 GPIO\_PATYP/ GPIO\_PBTYP 进行配置；每个 I/O 端口均支持输入滤波，由相应的端口滤波控制寄存器 GPIO\_PAFILT/ GPIO\_PBFILT 控制滤波是否使能。

每个 I/O 端口均支持开漏输出，由相应的端口开漏控制寄存器 GPIO\_PAODE/ GPIO\_PBODE 控制开漏输出是否使能。

每个 I/O 端口均支持弱上拉或弱下拉，由相应的端口弱上拉使能寄存器 GPIO\_PAPUE/ GPIO\_PBPUE 控制其弱上拉功能是否使能，由相应的端口弱下拉使能寄存器 GPIO\_PAPDE/ GPIO\_PBPDE 控制其弱下拉功能是否使能。需注意不可同时使能弱上拉和弱下拉功能。

每个 I/O 端口均支持电流驱动能力可配置，由相应的端口驱动电流控制寄存器 GPIO\_PADS/ GPIO\_PBDS 选择 I/O 端口的输出驱动能力，可选择为强电流驱动 I/O 端口，或者普通驱动 I/O 端口。

通过软件方式操作 I/O 端口输出脉冲信号时，能够得到的最快脉冲频率为外设时钟 PCLK 频率的 4 分频。



4.2 结构框图

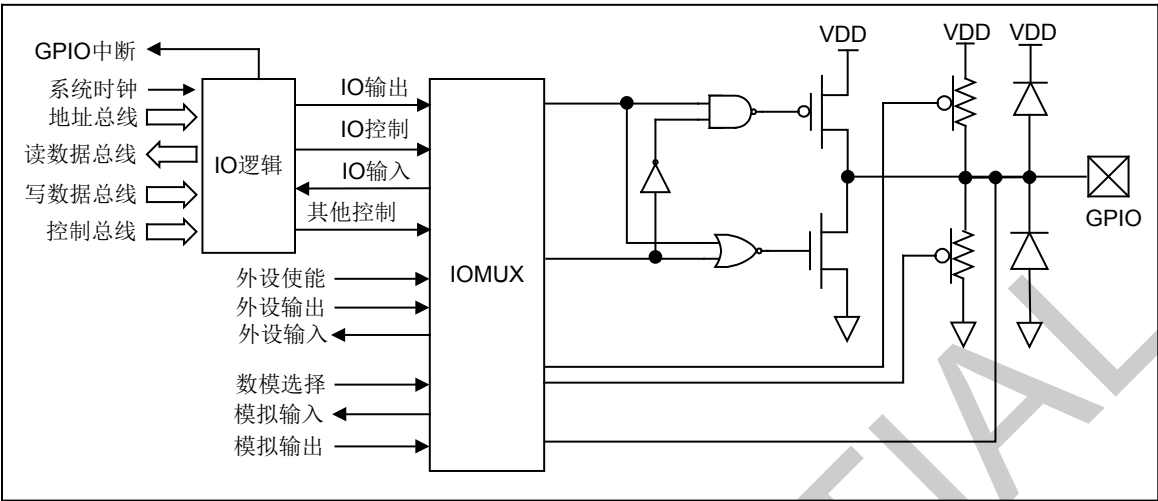


图 4-1 IO 端口电路结构图

4.3 外部端口中断

支持全部 IO 口外部端口中断，分成 8 组，每组对应一个 IRQ，并可通过 GPIO\_PINTCFG 寄存器的 PINT7CFG~PINT0CFG 位配置各外部中断触发方式。触发方式配置为高电平或低电平时，中断标志只有在触发电平翻转后，才可通过对 GPIO\_PINTIF 寄存器的相应位写 1 来清除。

GPIO\_PINTIE 寄存器的输入屏蔽使能位 PMASK，可对外部端口中断输入源进行屏蔽，屏蔽位使能时，对应的中断输入源就被屏蔽了，不会产生中断和标志位。

GPIO\_PINTIE 寄存器的外部端口中断使能位 PINTIE，可对每个外部端口中断标志 PINTIF 是否触发外部端口中断请求，进行配置。

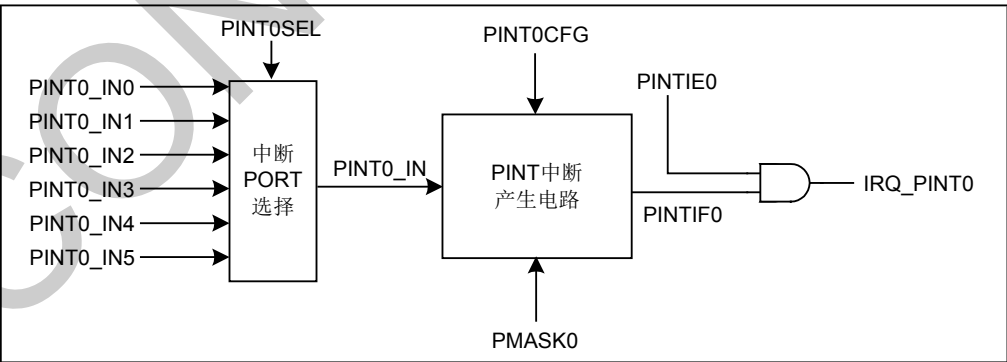


图 4-2 外部端口中断 PINT0 电路结构示意图

上图以外部端口中断 PINT0 电路结构示意图为例，可见其有 5 个可选外部中断源，分别来自 PA8、PA16、PA24、PB0 和 PB8。PINT0 ~PINT7 的外部中断源如下表所示：

PINT	SEL0	SEL1	SEL2	SEL3	SEL4	SEL5
PINT0	-	PA8	PA16	PA24	PB0	PB8
PINT1	PA1	PA9	-	PA25	PB1	PB9
PINT2	PA2	PA10	-	-	-	PB10
PINT3	PA3	PA11	-	PA27	-	PB11
PINT4	PA4	PA12	-	PA28	-	PB12
PINT5	PA5	PA13	-	-	-	PB13
PINT6	PA6	PA14	PA22	-	-	-
PINT7	PA7	PA15	PA23	-	-	-

表 4-1 PINT 选择对应列表

#### 4.4 外部按键中断

支持 1 个 8 输入外部按键中断 (KINT)，8 个按键输入为 KINT0 ~KINT7，其中任意一个按键输入都可以触发按键中断，每个按键输入可以从 6 个 I/O 端口中选择一个作为其输入源。

每个输入屏蔽使能位 KMASK，可对相应的输入源 KINT 进行屏蔽。屏蔽使能时，不论 KINT\_IN 输入如何变化，中断标志 KINTIF 均保持不变；屏蔽使能时，可通过端口中断配置寄存器 GPIO\_KINTCFG，对各个按键输入 KINT 进行配置，选择中断的有效触发边沿或电平。

每个按键中断使能位 KINTIE，可对相应的按键标志 KINTIF 是否触发按键中断请求，进行配置。

GPIO\_KINTCFG 寄存器的按键中断配置位 KINT7CFG~KINT0CFG，可对每个输入源的触发方式进行配置，根据端口电平变化的不同状态，触发中断。切换输入源触发方式时，需先屏蔽输入源，避免误产生中断；或先禁止按键中断，并在切换完成后，对中断标志进行清零，然后再使能按键中断。

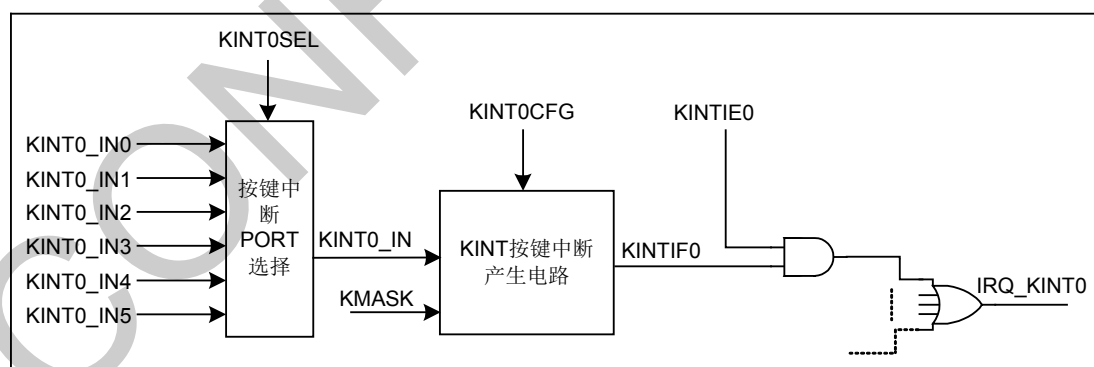


图 4-3 外部按键中断 KINT0 电路结构示意图

上图以外外部按键中断 KINT0 电路结构示意图为例，可见其有 5 个可选外部中断源，分别来自 PA8、PA16、PA24、PB0 和 PB8。KINT0 ~KINT7 的外部中断源如下表所示：

KINT	SEL0	SEL1	SEL2	SLE3	SEL4	SEL5
KINT0	-	PA8	PA16	PA24	PB0	PB8
KINT1	PA1	PA9	-	PA25	PB1	PB9
KINT2	PA2	PA10	-	-	-	PB10
KINT3	PA3	PA11	-	PA27	-	PB11
KINT4	PA4	PA12	-	PA28	-	PB12
KINT5	PA5	PA13	-	-	-	PB13
KINT6	PA6	PA14	PA22	-	-	-
KINT7	PA7	PA15	PA23	-	-	-

表 4-2 KINT 选择对应列表

## 4.5 Buzz输出

Buzz 输出可用于驱动蜂鸣器等音频发声器件。

GPIO\_BUZC 控制寄存器用于使能 Buzz 和设定其输出信号的频率。Buzz 信号频率为：

$$F_{BUZ} = \frac{F_{pclk}}{2 \times (BUZ\_LOAD + 1)}$$

由 GPIO\_PAFUNC0 和 GPIO\_PBFUNC1 寄存器来设定 Buzz 信号直接输出至 PA1 或 PB13。

Buzz 的固定频率输出信号还可被 UART0 的输出信号 TXD0 调制为 TXPWM0 信号，并送至相应输出端口上。可通过 TX0PLV 控制位设定为高电平调制还是低电平调制。

当 GPIO\_TXPWM 寄存器的 TX0PS 控制位为 2'b11 时，TXPWM0 调制信号可输出至 TXD0、T16N0OUT0、T16N0OUT1 或 BUZ 管脚（由 GPIO\_TXPWM 寄存器的 TX0\_S3~TX0\_S0 选取）。

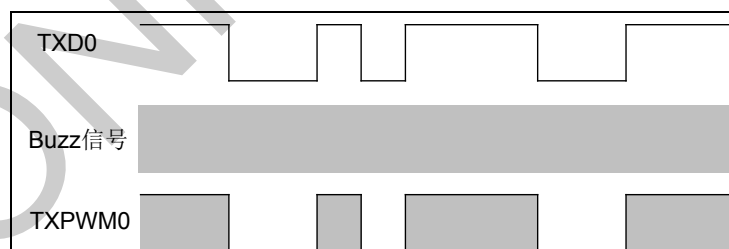


图 4-4 Buzz 高电平调制输出波形图

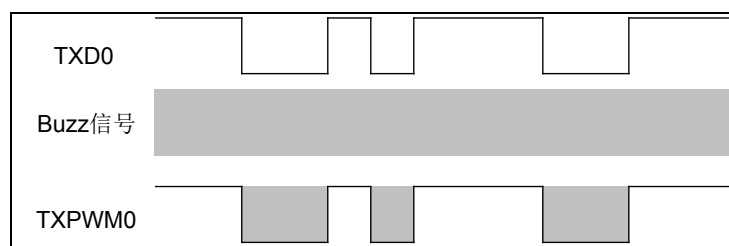


图 4-5 Buzz 低电平调制输出波形图

## 4.6 特殊功能寄存器

### 4.6.1 PA端口状态寄存器（GPIO\_PAPORT）

PA 端口状态寄存器（GPIO\_PAPORT）

偏移地址：00<sub>H</sub>

复位值：XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

PORT<31:0>	bit 31-0	R	PA 端口输入/输出电平 0: 低电平 1: 高电平
------------	----------	---	----------------------------------

### 4.6.2 PA端口数据寄存器（GPIO\_PADATA）

PA 端口数据寄存器（GPIO\_PADATA）

偏移地址：04<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

DATA<31:0>	bit 31-0	R/W	PA 端口输出寄存器 0: 输出低电平 1: 输出高电平
------------	----------	-----	------------------------------------

### 4.6.3 PA端口输出置位寄存器（GPIO\_PADATABSR）

PA 端口输出置位寄存器（GPIO\_PADATABSR）

偏移地址：08<sub>H</sub>

复位值：XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

DATABSR<31:0>	bit 31-0	W	PA 输出置位选择 0: 不改变输出电平 1: 相应端口输出高电平
---------------	----------	---	-----------------------------------------

#### 4.6.4 PA端口输出清零寄存器（GPIO\_PADATABCR）

##### PA 端口输出清零寄存器（GPIO\_PADATABCR）

偏移地址：0C<sub>H</sub>复位值：XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

DATABCR<31:0>	bit 31-0	W	<b>PA 输出清零选择</b> 0: 不改变输出电平 1: 相应端口输出低电平
---------------	----------	---	------------------------------------------------

#### 4.6.5 PA端口输出翻转寄存器（GPIO\_PADATABRR）

##### PA 端口输出翻转寄存器（GPIO\_PADATABRR）

偏移地址：10<sub>H</sub>复位值：XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

DATABRR<31:0>	bit 31-0	W	<b>PA 输出翻转选择</b> 0: 不改变输出电平 1: 相应端口输出翻转
---------------	----------	---	-----------------------------------------------

#### 4.6.6 PA端口方向控制寄存器（GPIO\_PADIR）

##### PA 端口方向控制寄存器（GPIO\_PADIR）

偏移地址：14<sub>H</sub>复位值：11111111\_11111111\_11111111\_11111111<sub>B</sub>

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

DIR<31:0>	bit31-0	R/W	<b>PA 端口方向控制位</b> 0: 输出 1: 非输出（若 GPIO_PAINEB 对应位为 0，则可作为数字输入端口使用。若需使能模拟通道功能，GPIO_PAINEB 和 GPIO_PADIR 的对应位都应设置
-----------	---------	-----	---------------------------------------------------------------------------------------------------------------------

			为 1，关闭数字输入和输出功能)
--	--	--	------------------

#### 4.6.7 PA端口方向置位寄存器 (GPIO\_PADIRBSR)

PA 端口方向置位寄存器 (GPIO\_PADIRBSR)

偏移地址: 18<sub>H</sub>

复位值: XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

DIRBSR<31:0>	bit 31-0	W	<b>PA 端口方向置位选择</b> 0: 不改变 GPIO_PADIR 值 1: 对应的 GPIO_PADIR 位设置为 1
--------------	----------	---	-----------------------------------------------------------------------

#### 4.6.8 PA端口方向清零寄存器 (GPIO\_PADIRBCR)

PA 端口方向清零寄存器 (GPIO\_PADIRBCR)

偏移地址: 1C<sub>H</sub>

复位值: XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

DIRBCR<31:0>	bit 31-0	W	<b>PA 端口方向清零选择</b> 0: 不改变 GPIO_PADIR 的值 1: 对应的 GPIO_PADIR 位设置为 0
--------------	----------	---	------------------------------------------------------------------------

#### 4.6.9 PA端口方向翻转寄存器 (GPIO\_PADIRBRR)

PA 端口方向翻转寄存器 (GPIO\_PADIRBRR)

偏移地址: 20<sub>H</sub>

复位值: XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

DIRBRR<31:0>	bit 31-0	W	<b>PA 端口方向翻转选择</b> 0: 不改变 GPIO_PADIR 的值 1: 对应的 GPIO_PADIR 位值翻转
--------------	----------	---	----------------------------------------------------------------------

#### 4.6.10 PA<7:0>端口复用选择寄存器 (GPIO\_PAFUNC0)

PA<7:0>端口复用选择寄存器 (GPIO\_PAFUNC0)

偏移地址: 24<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	保留	PA7<1:0>	保留	保留	保留	PA6<1:0>	保留	保留	保留	PA5<1:0>	保留	保留	保留	PA4<1:0>	保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	PA3<1:0>	保留	保留	保留	PA2<1:0>	保留	保留	保留	PA1<1:0>	保留	保留	保留	PA0<1:0>	保留

—	bit31-30	—	—
PA7<1:0>	bit29-28	R/W	<b>PA7 复用选择位</b> 00: FUN0 01: FUN1 10: FUN2 11: FUN3
—	bit27-26	—	—
PA6<1:0>	bit25-24	R/W	<b>PA6 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PA5<1:0>	bit21-20	R/W	<b>PA5 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PA4<1:0>	bit17-16	R/W	<b>PA4 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PA3<1:0>	bit13-12	R/W	<b>PA3 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PA2<1:0>	bit9-8	R/W	<b>PA2 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PA1<1:0>	bit5-4	R/W	<b>PA1 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-0	—	—

#### 4.6.11 PA<15:8>端口复用选择寄存器 (GPIO\_PAFUNC1)

PA<15:8>端口复用选择寄存器 (GPIO\_PAFUNC1)

偏移地址: 28<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	保留	PA15<1:0>	保留	保留	保留	PA14<1:0>	保留	保留	保留	PA13<1:0>	保留	保留	保留	PA12<1:0>	保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	PA11<1:0>	保留	保留	保留	PA10<1:0>	保留	保留	保留	PA9<1:0>	保留	保留	PA8<1:0>

—	bit31-30	—	—
PA15<1:0>	bit29-28	R/W	<b>PA15 复用选择位</b> 00: FUN0 01: FUN1 10: FUN2 11: FUN3
—	bit27-26	—	—
PA14<1:0>	bit25-24	R/W	<b>PA14 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PA13<1:0>	bit21-20	R/W	<b>PA13 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PA12<1:0>	bit17-16	R/W	<b>PA12 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PA11<1:0>	bit13-12	R/W	<b>PA11 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PA10<1:0>	bit9-8	R/W	<b>PA10 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PA9<1:0>	bit5-4	R/W	<b>PA9 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PA8<1:0>	bit1-0	R/W	<b>PA8 复用选择位</b> 00~11: FUN0~FUN3



#### 4.6.12 PA<23:16>端口复用选择寄存器 (GPIO\_PAFUNC2)

PA<23:16>端口复用选择寄存器 (GPIO\_PAFUNC2)

偏移地址: 2C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	保留	PA23<1:0>	保留	保留	保留	PA22<1:0>	保留	保留	保留	PA21<1:0>	保留	保留	保留	保留	PA20<1:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	PA19<1:0>	保留	保留	保留	PA18<1:0>	保留	保留	保留	保留	PA17<1:0>

—	bit31-30	—	—
PA23<1:0>	bit29-28	R/W	<b>PA23 复用选择位</b> 00: FUN0 01: FUN1 10: FUN2 11: FUN3
—	bit27-26	—	—
PA22<1:0>	bit25-24	R/W	<b>PA22 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-2	—	—
PA16<1:0>	bit1-0	R/W	<b>PA16 复用选择位</b> 00~11: FUN0~FUN3

#### 4.6.13 PA<31:24>端口复用选择寄存器 (GPIO\_PAFUNC3)

PA<31:24>端口复用选择寄存器 (GPIO\_PAFUNC3)

偏移地址: 30<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	保留	PA31<1:0>	保留	保留	保留	PA30<1:0>	保留	保留	保留	PA29<1:0>	保留	保留	保留	保留	PA28<1:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	PA27<1:0>	保留	保留	保留	PA26<1:0>	保留	保留	保留	保留	PA25<1:0>

—	bit31-18	—	—
PA28<1:0>	bit17-16	R/W	<b>PA28 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PA27<1:0>	bit13-12	R/W	<b>PA27 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-6	—	—
PA25<1:0>	bit5-4	R/W	<b>PA25 复用选择位</b> 00~11: FUN0~FUN3

—	bit3-2	—	—
PA24<1:0>	bit1-0	R/W	<b>PA24 复用选择位</b> 00~11: FUN0~FUN3

注 1: PA2/PA3/PA27/PA16 中只能有一个可配置为 PWM 输出的刹车控制信号, 优先级为: PA2>PA3>PA27>PA16,

例如一旦 PA3 设置为 PWM 输出刹车控制信号, 则 PA27 和 PA16 不可再配置为 PWM 输出刹车控制信号。

注 2: GPIO\_PAFUNC 寄存器仅用于端口的数字输入输出功能选择。使用端口的模拟功能时, 需要设置 GPIO\_PADIR 和 GPIO\_PAINEB 寄存器的对应位为 1, 关闭端口的数字输入和输出功能。

#### 4.6.14 PA端口输入控制寄存器 (GPIO\_PAINEB)

##### PA 端口输入控制寄存器 (GPIO\_PAINEB)

偏移地址: 34<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

INEB<31:0>	bit31-0	R/W	<b>端口数字输入功能使能位</b> 0: 使能 1: 禁止
------------	---------	-----	--------------------------------------

#### 4.6.15 PA端口开漏控制寄存器 (GPIO\_PAODE)

##### PA 端口开漏控制寄存器 (GPIO\_PAODE)

偏移地址: 38<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

ODE<31:0>	bit31-0	R/W	<b>端口输出开漏使能位</b> 0: 禁止, 端口为推挽输出 1: 使能, 端口为开漏输出
-----------	---------	-----	------------------------------------------------------

#### 4.6.16 PA端口弱上拉使能寄存器 (GPIO\_PAPUE)

##### PA 端口弱上拉使能寄存器 (GPIO\_PAPUE)

偏移地址: 3C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

PUE<31:0>	bit31-0	R/W	端口弱上拉使能位 0: 禁止 1: 使能
-----------	---------	-----	----------------------------

#### 4.6.17 PA端口弱下拉使能寄存器 (GPIO\_PAPDE)

##### PA 端口弱下拉使能寄存器 (GPIO\_PAPDE)

偏移地址: 40<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

PDE<31:0>	bit31-0	R/W	端口弱下拉使能位 0: 禁止 1: 使能
-----------	---------	-----	----------------------------

#### 4.6.18 PA端口驱动电流控制寄存器 (GPIO\_PADS)

##### PA 端口驱动电流控制寄存器 (GPIO\_PADS)

偏移地址: 44<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

DS<31:0>	bit31-0	R/W	端口输出驱动能力选择位 0: 普通电流驱动 1: 强电流驱动
----------	---------	-----	--------------------------------------

#### 4.6.19 PA端口类型选择寄存器 (GPIO\_PATYP)

##### PA 端口类型选择寄存器 (GPIO\_PATYP)

偏移地址: 48<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

TYP<31:0>	bit31-0	R/W	PA 端口类型选择位 0: CMOS 1: TTL
-----------	---------	-----	---------------------------------

#### 4.6.20 PA端口滤波控制寄存器 (GPIO\_PAFLT)

##### PA 端口滤波控制寄存器 (GPIO\_PAFLT)

偏移地址: 4C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

FLT<31:0>	bit31-0	R/W	PA 端口 20ns 滤波使能位 0: 禁止 1: 使能
-----------	---------	-----	------------------------------------

#### 4.6.21 PB端口状态寄存器 (GPIO\_PBPORT)

##### PB 端口状态寄存器 (GPIO\_PBPORT)

偏移地址: 80<sub>H</sub>

复位值: 00000000\_00000000\_00xxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-14	—	—
PORT<13:0>	bit13-0	R	PB 端口输入/输出电平 0: 低电平 1: 高电平

#### 4. 6. 22 PB端口数据寄存器 (GPIO\_PBDATA)

##### PB 端口数据寄存器 (GPIO\_PBDATA)

偏移地址: 84<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-14	—	—
DATA<13:0>	bit13-0	R/W	<b>PB 端口输出电平</b> 0: 输出低电平 1: 输出高电平

#### 4. 6. 23 PB端口输出置位寄存器 (GPIO\_PBDATABSR)

##### PB 端口输出置位寄存器 (GPIO\_PBDATABSR)

偏移地址: 88<sub>H</sub>

复位值: XXXXXXXX\_XXXXXXX\_XXXXXXX\_XXXXXXX<sub>B</sub>

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

—	bit31-14	—	—
DATABSR<13:0>	bit13-0	W	<b>PB 输出置位选择</b> 0: 不改变输出电平 1: 相应端口输出高电平

#### 4. 6. 24 PB端口输出清零寄存器 (GPIO\_PBDATABCR)

##### PB 端口输出清零寄存器 (GPIO\_PBDATABCR)

偏移地址: 8C<sub>H</sub>

复位值: XXXXXXXX\_XXXXXXX\_XXXXXXX\_XXXXXXX<sub>B</sub>

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

—	bit31-14	—	—
DATABCR<13:0>	bit13-0	W	<b>PB 输出清零选择</b> 0: 不改变输出电平

			1: 相应端口输出低电平
--	--	--	--------------

#### 4. 6. 25 PB端口输出翻转寄存器 (GPIO\_PBDATABRR)

##### PB 端口输出翻转寄存器 (GPIO\_PBDATABRR)

偏移地址: 90<sub>H</sub>

复位值: XXXXXXXX\_XXXXXXX\_XXXXXXX\_XXXXXXX<sub>B</sub>

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

—	bit31-14	—	—
DATABRR<13:0>	bit13-0	W	<b>PB 输出翻转选择</b> 0: 不改变输出电平 1: 相应端口输出翻转

#### 4. 6. 26 PB端口方向控制寄存器 (GPIO\_PBDIR)

##### PB 端口方向控制寄存器 (GPIO\_PBDIR)

偏移地址: 94<sub>H</sub>

复位值: 11111111\_11111111\_11111111\_11111111<sub>B</sub>

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

—	bit31-14	—	—
DIR<13:0>	bit13-0	RW	<b>PB 端口方向控制位</b> 0: 输出 1: 输入 (若 GPIO_PBINEB 对应位为 0, 则可作为数字输入端口使用。若需使能模拟通道功能, GPIO_PBINEB 和 GPIO_PBDIR 的对应位都应设置为 1, 关闭数字输入和输出功能)

#### 4.6.27 PB端口方向置位寄存器 (GPIO\_PBDIRBSR)

PB 端口方向置位寄存器 (GPIO\_PBDIRBSR)

偏移地址: 98<sub>H</sub>复位值: XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

—	bit31-14	—	—
DIRBSR<13:0>	bit13-0	W	<b>PB 端口方向置位选择</b> 0: 不改变 GPIO_PBDIR 的值 1: 对应的 GPIO_PBDIR 位设置为 1

#### 4.6.28 PB端口方向清零寄存器 (GPIO\_PBDIRBCR)

PB 端口方向清零寄存器 (GPIO\_PBDIRBCR)

偏移地址: 9C<sub>H</sub>复位值: XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

—	bit31-14	—	—
DIRBCR<13:0>	bit13-0	W	<b>PB 端口方向清零选择</b> 0: 不改变 GPIO_PBDIR 的值 1: 对应的 GPIO_PBDIR 位设置为 0

#### 4.6.29 PB端口方向翻转寄存器 (GPIO\_PBDIRBRR)

PB 端口方向翻转寄存器 (GPIO\_PBDIRBRR)

偏移地址: A0<sub>H</sub>复位值: XXXXXXXX\_XXXXXXXX\_XXXXXXXX\_XXXXXXXX<sub>B</sub>

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

—	bit31-14	—	—
DIRBRR<13:0>	bit13-0	W	<b>PB 端口方向翻转选择</b> 0: 不改变 GPIO_PBDIR 的值

			1: 对应的 GPIO_PBDIR 位值翻转
--	--	--	------------------------

#### 4. 6. 30 PB<7:0>端口复用选择寄存器 (GPIO\_PBFUNC0)

PB<7:0>端口复用选择寄存器 (GPIO\_PBFUNC0)

偏移地址: A4<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PB7<1:0>		保留		PB6<1:0>		保留		PB5<1:0>		保留		PB4<1:0>	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		PB3<1:0>		保留		PB2<1:0>		保留		PB <1:0>		保留		PB0<1:0>	

—	bit31-6	—	—
PB1<1:0>	bit5-4	R/W	<b>PB1 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PB0<1:0>	bit1-0	R/W	<b>PB0 复用选择位</b> 00~11: FUN0~FUN3

#### 4. 6. 31 PB<15:8>端口复用选择寄存器 (GPIO\_PBFUNC1)

PB<15:8>端口复用选择寄存器 (GPIO\_PBFUNC1)

偏移地址: A8<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PB13<1:0>		保留		PB12<1:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		PB11<1:0>		保留		PB10<1:0>		保留		PB9<1:0>		保留		PB8<1:0>	

—	bit31-22	—	—
PB13<1:0>	bit21-20	R/W	<b>PB13 复用选择位</b> 00: FUN0 01: FUN1 10: FUN2 11: FUN3
—	bit19-18	—	—
PB12<1:0>	bit17-16	R/W	<b>PB12 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PB11<1:0>	bit13-12	R/W	<b>PB11 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—



PB10<1:0>	bit9-8	R/W	<b>PB10 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PB9<1:0>	bit5-4	R/W	<b>PB9 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PB8<1:0>	bit1-0	R/W	<b>PB8 复用选择位</b> 00~11: FUN0~FUN3

注：GPIO\_PBFUNC 仅用于端口的数字输入输出功能选择。使用 PB 端口的模拟功能时，需要设置 GPIO\_PBDIR 和 GPIO\_PBINEB 寄存器的对应位为 1，关闭端口的数字输入和输出功能。

#### 4.6.32 PB端口输入控制寄存器（GPIO\_PBINEB）

**PB 端口输入控制寄存器（GPIO\_PBINEB）**

偏移地址：B4<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-14	—	—
INEB<13:0>	bit13-0	R/W	端口数字输入功能使能位 0: 使能 1: 禁止

#### 4.6.33 PB端口开漏控制寄存器（GPIO\_PBODE）

**PB 端口开漏控制寄存器（GPIO\_PBODE）**

偏移地址：B8<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-14	—	—
ODE<13:0>	bit13-0	R/W	端口输出开漏使能位 0: 禁止，端口为推挽输出 1: 使能，端口为开漏输出

#### 4.6.34 PB端口弱上拉使能寄存器 (GPIO\_PBPUE)

##### PB 端口弱上拉使能寄存器 (GPIO\_PBPUE)

偏移地址: BC<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-14	—	—
PUEN<13:0>	bit13-0	R/W	端口弱上拉使能位 0: 禁止 1: 使能

注: 当 MRSTN 管脚用作外部复位管脚时, 芯片内部固定集成了约 53K 欧姆的弱上拉电阻, 不受端口弱上拉使能寄存器的控制。

#### 4.6.35 PB端口弱下拉使能寄存器 (GPIO\_PBPDE)

##### PB 端口弱下拉使能寄存器 (GPIO\_PBPDE)

偏移地址: C0<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-14	—	—
PDEN<13:0>	bit13-0	R/W	端口弱下拉使能位 0: 禁止 1: 使能

#### 4.6.36 PB端口驱动电流控制寄存器 (GPIO\_PBDS)

##### PB 端口驱动电流控制寄存器 (GPIO\_PBDS)

偏移地址: C4<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-14	—	—
DS<13:0>	bit13-0	R/W	端口输出驱动能力选择位 0: 普通电流驱动 1: 强电流驱动

#### 4.6.37 PB端口类型选择寄存器 (GPIO\_PBTYP)

PB 端口类型选择寄存器 (GPIO\_PBTYP)

偏移地址: C8<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-14	—	—
TYP<13:0>	bit13-0	R/W	PB 端口类型选择位 0: CMOS 1: TTL

#### 4.6.38 PB端口滤波控制寄存器 (GPIO\_PBFLT)

PB 端口滤波控制寄存器 (GPIO\_PBFLT)

偏移地址: CC<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-14	—	—
FLT<13:0>	bit13-0	R/W	PB 端口 20ns 滤波使能位 0: 禁止 1: 使能

#### 4.6.39 PINT中断使能寄存器 (GPIO\_PINTIE)

PINT 中断使能寄存器 (GPIO\_PINTIE)

偏移地址: 00<sub>H</sub>

复位值: 00000000\_00000000\_11111111\_00000000<sub>B</sub>

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

—	bit31-16	—	—
PMASK<7:0>	bit15-8	R/W	<b>PINT 中断源屏蔽控制位</b> 0: 不屏蔽 1: 屏蔽
PINTIE<7:0>	bit7-0	R/W	<b>PINT 使能位</b> 0: 禁止 1: 使能

#### 4.6.40 PINT中断标志寄存器（GPIO\_PINTIF）

##### PINT 中断标志寄存器（GPIO\_PINTIF）

偏移地址：04<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit 31-8	—	—
PINTIF<7:0>	bit7-0	R/W	<b>GPIO 外部中断标志位</b> 0: 无中断 1: 有中断 软件写 1 清除中断标志位，写 0 无效

注：对 GPIO\_PINTIF 寄存器中的各中断标志位，写 0 无效，写 1 才能清除标志位；读操作时，读取的值为 1 表示有中断发生。

#### 4.6.41 PINT0~7 中断源选择寄存器（GPIO\_PINTSEL）

##### PINT0~7 中断源选择寄存器（GPIO\_PINTSEL）

偏移地址：08<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	PINT7SEL<2:0>		保留	PINT6SEL<2:0>		保留	PINT5SEL<2:0>		保留	PINT4SEL<2:0>		保留	PINT3SEL<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PINT3SEL<2:0>		保留	PINT2SEL<2:0>		保留	PINT1SEL<2:0>		保留	PINT0SEL<2:0>		保留	PINT0SEL<2:0>		保留

—	bit31	—	—
PINT7SEL<2:0>	bit30-28	R/W	<b>PINT7 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit27	—	—

PINT6SEL<2:0>	bit26-24	R/W	<b>PINT6 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit23	—	—
PINT5SEL<2:0>	bit22-20	R/W	<b>PINT5 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit19	—	—
PINT4SEL<2:0>	bit18-16	R/W	<b>PINT4 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit15	—	—
PINT3SEL<2:0>	bit14-12	R/W	<b>PINT3 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit11	—	—
PINT2SEL<2:0>	bit10-8	R/W	<b>PINT2 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit7	—	—
PINT1SEL<2:0>	bit6-4	R/W	<b>PINT1 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit3	—	—
PINT0SEL<2:0>	bit2-0	R/W	<b>PINT0 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0

#### 4.6.42 PINT中断配置寄存器（GPIO\_PINTCFG）

PINT 中断配置寄存器（GPIO\_PINTCFG）

偏移地址: 0C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	PINT7CFG<2:0>		保留	PINT6CFG<2:0>		保留	PINT5CFG<2:0>		保留	PINT4CFG<2:0>		保留	PINT3CFG<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PINT3CFG<2:0>		保留	PINT2CFG<2:0>		保留	PINT1CFG<2:0>		保留	PINT0CFG<2:0>		保留			

—	bit31	—	—
PINT7CFG<2:0>	bit30-28	R/W	<b>PINT7 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit27	—	—
PINT6CFG<2:0>	bit26-24	R/W	<b>PINT6 配置位</b>

			参见 GPIO_PINTCFG 详情
—	bit23	—	—
PINT5CFG<2:0>	bit22-20	R/W	<b>PINT5 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit19	—	—
PINT4CFG<2:0>	bit18-16	R/W	<b>PINT4 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit15	—	—
PINT3CFG<2:0>	bit14-12	R/W	<b>PINT3 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit11	—	—
PINT2CFG<2:0>	bit10-8	R/W	<b>PINT2 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit7	—	—
PINT1CFG<2:0>	bit6-4	R/W	<b>PINT1 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit3	—	—
PINT0CFG<2:0>	bit2-0	R/W	<b>PINT0 配置位</b> 参见 GPIO_PINTCFG 详情

寄存器名称	GPIO_PINTCFG 详情		
PINTCFG<2:0>	bit 2-0	R/W	<b>GPIO_PINTCFG 配置位</b> 000: 上升沿触发中断 001: 下降沿触发中断 010: 高电平触发中断 011: 低电平触发中断 1xx: 上升沿和下降沿均触发中断

#### 4. 6. 43 KINT中断使能寄存器（GPIO\_KINTIE）

##### KINT 中断使能寄存器（GPIO\_KINTIE）

偏移地址：10<sub>H</sub>

复位值：00000000\_00000000\_11111111\_00000000<sub>B</sub>

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

—	bit31-16	—	—
KMASK<7:0>	bit15-8	R/W	<b>KIN 按键输入屏蔽控制位</b> 0: 不屏蔽 1: 屏蔽
KINTIE<7:0>	bit7-0	R/W	<b>KINT 中断使能位</b>

			0: 禁止 1: 使能
--	--	--	----------------

#### 4. 6. 44 KINT中断标志寄存器 (GPIO\_KINTIF)

##### KINT 中断标志寄存器 (GPIO\_KINTIF)

偏移地址: 14<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit 31-8	—	—
KINTIF<7:0>	bit7-0	R/W	<b>GPIO 按键中断标志位</b> 0: 无中断 1: 有中断 软件写 1 清除中断标志位, 写 0 无效

注: 对 GPIO\_KINTIF 寄存器中的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

#### 4. 6. 45 KINT0~7 中断源选择寄存器 (GPIO\_KINTSEL)

##### KINT0~7 中断源选择寄存器 (GPIO\_KINTSEL)

偏移地址: 18<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	KINT7SEL<2:0>		保留	KINT6SEL<2:0>		保留	KINT5SEL<2:0>		保留	KINT4SEL<2:0>		保留	KINT3SEL<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	KINT7SEL<2:0>		保留	KINT6SEL<2:0>		保留	KINT5SEL<2:0>		保留	KINT4SEL<2:0>		保留	KINT3SEL<2:0>		保留

—	bit31	—	—
KINT7SEL<2:0>	bit30-28	R/W	<b>KINT7 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit27	—	—
KINT6SEL<2:0>	bit26-24	R/W	<b>KINT6 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit23	—	—
KINT5SEL<2:0>	bit22-20	R/W	<b>KINT5 输入选择位</b> 000~101: SEL0~SEL5

			其余: SEL0
—	bit19	—	—
KINT4SEL<2:0>	bit18-16	R/W	<b>KINT4 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit15	—	—
KINT3SEL<2:0>	bit14-12	R/W	<b>KINT3 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit11	—	—
KINT2SEL<2:0>	bit10-8	R/W	<b>KINT2 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit7	—	—
KINT1SEL<2:0>	bit6-4	R/W	<b>KINT1 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit3	—	—
KINT0SEL<2:0>	bit2-0	R/W	<b>KINT0 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0

#### 4.6.46 KINT中断配置寄存器 (GPIO\_KINTCFG)

KINT 中断配置寄存器 (GPIO\_KINTCFG)

偏移地址: 1C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	KINT7CFG<2:0>			保留	KINT6CFG<2:0>			保留	KINT5CFG<2:0>			保留	KINT4CFG<2:0>		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	KINT3CFG<2:0>			保留	KINT2CFG<2:0>			保留	KINT1CFG<2:0>			保留	KINT0CFG<2:0>		

—	bit31	—	—
KINT7CFG<2:0>	bit30-28	R/W	<b>KINT7 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit27	—	—
KINT6CFG<2:0>	bit26-24	R/W	<b>KINT6 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit23	—	—
KINT5CFG<2:0>	bit22-20	R/W	<b>KINT5 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit19	—	—
KINT4CFG<2:0>	bit18-16	R/W	<b>KINT4 配置位</b>



			参见 GPIO_KINTCFG 详情
—	bit15	—	—
KINT3CFG<2:0>	bit14-12	R/W	<b>KINT3 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit11	—	—
KINT2CFG<2:0>	bit10-8	R/W	<b>KINT2 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit7	—	—
KINT1CFG<2:0>	bit6-4	R/W	<b>KINT1 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit3	—	—
KINT0CFG<2:0>	bit2-0	R/W	<b>KINT0 配置位</b> 参见 GPIO_KINTCFG 详情

寄存器名称	GPIO_KINTCFG 详情		
GPIO_KINTCFG	bit 2-0	R/W	<b>GPIO_KINTCFG 配置位</b> 000: 上升沿触发中断 001: 下降沿触发中断 010: 高电平触发中断 011: 低电平触发中断 1xx: 上升沿和下降沿均触发中断

#### 4.6.47 脉宽调制寄存器 (GPIO\_TXPWM)

脉宽调制寄存器 (GPIO\_TXPWM)

偏移地址: 80<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TX0 _S3	TX0 _S2	TX0 _S1	TX0 S0	保留	TX0PL V	TX0PS<1:0>	

—	bit31-8	—	—
TX0_S3	bit7	R/W	<b>TXPWM0 调制输出使能端</b> 0: BUZ 管脚正常输出 1: BUZ 管脚为 TXPWM0 输出
TX0_S2	bit6	R/W	<b>TXPWM0 调制输出使能端</b> 0: T16N0OUT1 管脚正常输出 1: T16N0OUT1 管脚为 TXPWM0 输出
TX0_S1	bit5	R/W	<b>TXPWM0 调制输出使能端</b> 0: T16N0OUT0 管脚正常输出 1: T16N0OUT0 管脚为 TXPWM0 输出

TX0_S0	bit4	R/W	<b>TXPWM0 调制输出使能端</b> 0: TXD0 管脚正常输出 1: TXD0 管脚为 TXPWM0 输出
—	bit3	—	—
TX0PLV	bit2	R/W	<b>TXPWM0 调制电平选择位</b> 0: 低电平调制 (TXD0 与 TX0PS 所选取的脉冲信号进行硬件或操作) 1: 高电平调制 (TXD0 与 TX0PS 所选取的脉冲信号进行硬件与操作)
TX0PS<1:0>	bit1-0	R/W	<b>TXPWM0 调制 PWM 脉冲选择位</b> 00: 调制禁止 01: 与 T16N0OUT0 进行调制 10: 与 T16N0OUT1 进行调制 11: 与 BUZ 进行调制

#### 4. 6. 48 BUZ控制寄存器 (GPIO\_BUZC)

BUZ 控制寄存器 (GPIO\_BUZC)

偏移地址: 84<sub>H</sub>

复位值: 00001111\_11111111\_11111111\_00000000<sub>B</sub>

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

—	bit31-28	—	—
BUZ_LOAD<19:0>	bit27-8	R/W	<b>BUZ 计数装载值寄存器</b> BUZ 信号频率的计算公式为: $F_{BUZ} = \frac{F_{pclk}}{2 \times (BUZ\_LOAD + 1)}$
—	bit7-1	—	—
BUZEN	bit0	R/W	<b>BUZ 使能位</b> 0: 禁止 1: 使能

## 第5章 外设

### 5.1 定时器/计数器

#### 5.1.1 16 位定时器/计数器T16N

以 T16N0 为例，T16N1/T16N2/T16N3 同 T16N0。

##### 5.1.1.1 概述

- ◆ 1 个 8 位可配置预分频器，分频时钟作为 T16N\_CNT0/1 的定时/计数时钟
  - ◇ 预分频时钟源可选：PCLK 或 T16N0CK0/T16N0CK1
  - ◇ 预分频计数器可由 T16N\_PRECNT 寄存器设定计数初值
  - ◇ 分频比由寄存器 T16N\_PREMAT 设定
  - ◇ 支持预分频比缓冲器 T16N\_PREBUF(调制模式下寄存器位 REGBUF\_EN=1 使能)
- ◆ 2 个 16 位可配置定时/计数寄存器 T16N\_CNT0/T16N\_CNT1
  - ◇ T16N\_CNT1 仅在独立调制工作模式下可用
- ◆ 2 个 16 位峰值寄存器 T16N\_TOP0/T16N\_TOP1
  - ◇ T16N\_TOP1 仅在独立调制工作模式下可用
  - ◇ T16N\_CNT0/ T16N\_CNT1 计数值达到峰值时被清零
  - ◇ 在调制工作模式下，若使能 ADC 触发功能，T16N\_CNT0/T16N\_CNT1 计数值达到峰值时可分别产生 ADC 转换触发信号
  - ◇ 支持 2 个 16 位峰值缓冲器 T16N\_TOPBUF0/T16N\_TOPBUF1(调制模式下寄存器位 REGBUF\_EN=1 使能)
- ◆ 支持定时/计数工作模式
  - ◇ 支持 4 组 16 位计数匹配寄存器 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3，计数匹配后支持下列操作（需设置寄存器位 REGBUF\_EN=0）：
    - 产生中断
    - 支持 T16N\_CNT0 计数寄存器三种操作：保持，清零或继续计数（在独立调制工作模式下也支持 T16N\_CNT1 计数寄存器的上述三种操作）
    - 支持 T16N0OUT0 端口四种操作：保持，清零，置 1 或取反（在独立调制工作模式下也支持 T16N0OUT1 端口的上述四种操作，在同步和互补调制工作模式下支持 T16N0OUT1 端口输出与 T16N0OUT0 端口相同和互补的 PWM 波形）
- ◆ 支持输入捕捉工作模式
  - ◇ 捕捉边沿可配置
  - ◇ 捕捉次数可配置
- ◆ 支持调制工作模式
  - ◇ 通过对匹配寄存器进行配置，同时设置匹配后端口输出特性，可得到相应的 PWM 输出
  - ◇ 支持调制过程触发 ADC 转换
    - T16N\_MAT0、T16N\_MAT1、T16N\_MAT2、T16N\_MAT3 与 T16N\_CNT0 计数值

匹配并产生中断时触发（当寄存器位 REGBUF\_EN=1 时，由缓冲器 T16N\_MATBUF0、T16N\_MATBUF1、T16N\_MATBUF2、T16N\_MATBUF3 与 T16N\_CNT0 计数值匹配）

- T16N\_TOP0 与 T16N\_CNT0 计数值匹配触发（当寄存器位 REGBUF\_EN=1 时，由缓冲器 T16N\_TOPBUF0 与 T16N\_CNT0 计数值匹配）
- T16N\_TOP1 与 T16N\_CNT1 计数值匹配触发（当寄存器位 REGBUF\_EN=1 时，由缓冲器 T16N\_TOPBUF1 与 T16N\_CNT1 计数值匹配）

◇ 两路 PWM 可配置为独立/同步/互补三种模式

- 独立模式，T16N0OUT0/T16N0OUT1 输出不同的 PWM 波形
- 同步模式，T16N0OUT0/T16N0OUT1 输出相同 PWM 波形
- 互补模式，T16N0OUT0/T16N0OUT1 输出互补 PWM 波形，互补模式下死区时间宽度可配置

◇ 支持刹车控制

- 可通过 GPIO\_PAFUNC 寄存器来选取 PA2 (T16N\_BK0), PA3 (T16N\_BK1), PA16 (T16N\_BK2) 或 PA27 (T16N\_BK3)，其中任意一个作为外部刹车信号，有效刹车极性可配置
- 刹车输出电平可配置

### 5.1.1.2 结构框图

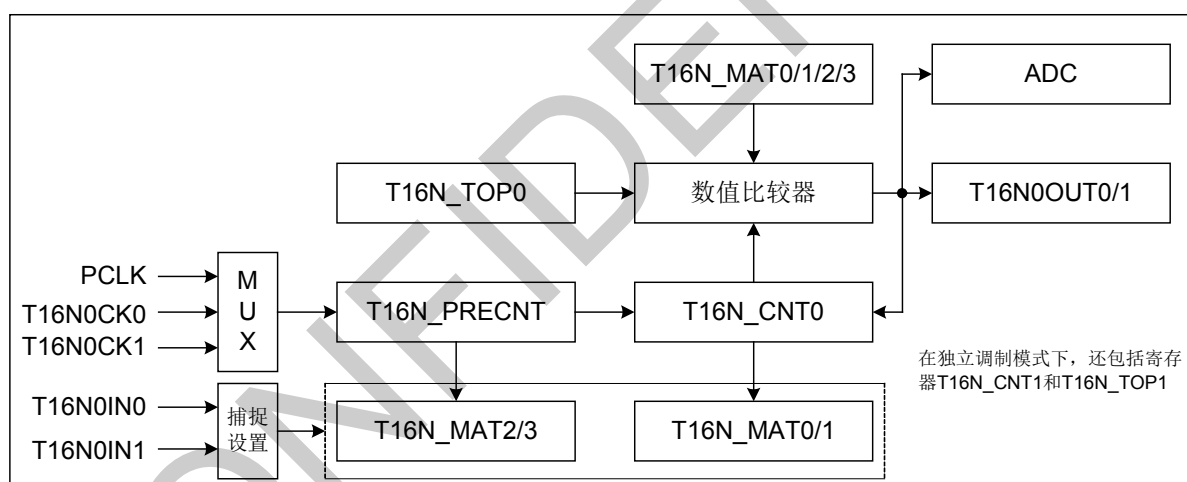


图 5-1 T16N0 电路结构框图

注：调制模式下，当寄存器位 REGBUF\_EN=1 时，由缓冲器 T16N\_MATBUF0/1/2/3，T16N\_TOPBUF0 与 T16N\_CNT0 计数值进行比较匹配，T16N\_TOPBUF1 与 T16N\_CNT1 计数值进行比较匹配，预分频计数匹配由预分频比缓冲器 T16N\_PREBUF 设置。

### 5.1.1.3 T16N定时/计数功能

设置 T16N\_CON0 寄存器的 MOD<1:0>=00 或 01，使 T16N 工作在定时/计数模式。在定时/计数模式下 T16N\_CNT1 和 T16N\_TOP1 不可使用。

设置 T16N\_CON0 寄存器的 EN=1，使能 T16N，计数值寄存器 T16N\_CNT0 从预设值开始累加计数。

设置 T16N\_CON0 寄存器的 CS，选择计数时钟源。时钟源为内部时钟 PCLK 时，为定

时模式；时钟源为外部时钟 T16N0CK0/T16N0CK1 端口输入时，为计数模式。

设置 T16N\_CON0 寄存器的 SYNC，选择外部时钟 T16N0CK0/T16N0CK1 是否被内部时钟 PCLK 同步。当选择外部时钟被同步时，为同步计数模式，否则为异步计数模式。同步计数模式时，T16N0CK0/T16N0CK1 端口输入的高/低电平脉宽均必须大于 2 个 PCLK 时钟周期。

设置 T16N\_CON0 寄存器的 EDGE，选择外部时钟计数方式：上升沿计数，下降沿计数，或上升/下降沿均计数，其中上升/下降沿均计数只适用于同步计数模式。

设置 T16N\_CON0 寄存器的 MAT0S/MAT1S/MAT2S/MAT3S<1:0>，选择计数匹配后 T16N\_CNT0 计数值寄存器的工作状态。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=00：当 T16N\_CNT0 计数值匹配 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 时，继续累加计数，不产生中断，当计数到 0xFFFF 后，下一次累加计数溢出，T16N\_CNT0 的值为 0x0000，并产生中断，重新开始累加计数。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=01：当 T16N\_CNT0 计数值匹配 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 时，计数值将保持，即在下一个计数时钟（经过预分频之后的时钟）到来时，T16N\_CNT0 不再累加计数，只产生中断。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=10：当 T16N\_CNT0 计数值匹配 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 时，计数值在下一个计数时钟（经过预分频之后的时钟）到来时被清零，并产生中断，重新开始累加计数。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=11：当 T16N\_CNT0 计数值匹配 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 时，继续累加计数，并在下一个计数时钟（经过预分频之后的时钟）到来时，产生中断，当计数到 0xFFFF<sub>H</sub>，下一次累加计数溢出，T16N\_CNT0 的值为 0x0000，并产生中断，重新开始累加计数。

对设置的多个不同匹配值 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3，当计数值匹配到每一个匹配值时，会产生相应的中断。由于中断产生后，T16N 继续累加计数，若未及时读取匹配中断标志位并软件清除，则有可能后续的匹配中断也已产生，从而会同时读到多个有效的匹配中断标志。

如果 T16N\_CNT0 的计数值匹配 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3，当 T16N\_CON2 寄存器的 MOE0=1 时，T16N0OUT0 端口输出状态由 MOM0/MOM1<1:0>位控制；当 MOE1=1 时，T16N0OUT1 端口输出状态由 MOM2/MOM3<1:0>位控制。可选择 T16N0OUT0/T16N0OUT1 端口同时输出。

注：定时/计数模式下，需设置寄存器位 REGBUF\_EN=0，不使用预分频比缓冲器和计数匹配缓冲器。

举例说明：T16N\_CNT0 计数值匹配 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2 后的工作方式。

T16N\_MAT0<15:0>=0x0002，T16N\_CON0 寄存器的 MAT0S<1:0>=00，继续计数，不产生中断

T16N\_MAT1<15:0>=0x0004，T16N\_CON0 寄存器的 MAT1S<1:0>=11，继续计数，产

生中断；

T16N\_MAT2<15:0>=0x0006，T16N\_CON0 寄存器的 MAT2S<1:0>=10，清零，产生中断，重新计数。

预分频设置为 1:1，采用内部 PCLK 时钟源。计数匹配功能示意图如下所示：

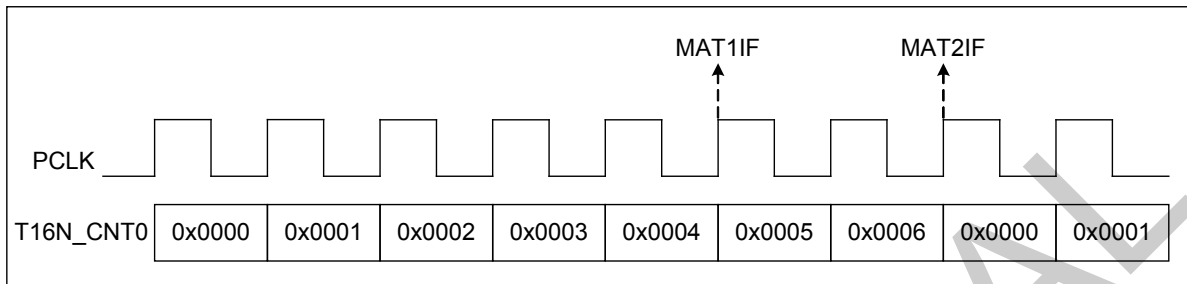


图 5-2 T16N0 计数匹配功能示意图

#### 5.1.1.4 T16N输入捕捉功能

设置 T16N\_CON0 寄存器的 MOD<1:0>=10，使 T16N 工作在捕捉模式。在捕捉模式下 T16N\_CNT1 不可使用。

在捕捉工作模式下，需设置 T16N\_CON0 寄存器的 CS<1:0>=00，使 T16N\_CNT0 采用内部 PCLK 时钟源计数；并且设置 T16N\_CON0 寄存器的 MAT0S/MAT1S/MAT2S/MAT3S <1:0>=00，计数匹配不影响 T16N\_CNT0 的工作。

在捕捉工作模式下，对端口 T16N0IN0 和 T16N0IN1 的状态进行检测。

当 T16N0IN0 端口的状态变化符合所设定的捕捉事件时，将 T16N\_CNT0 和 T16N\_PRECNT 的当前值分别装载到 T16N\_MAT0 和 T16N\_MAT2 寄存器中，产生 T16N 的 CAP0IF 中断。通过使能 T16N\_CON1 寄存器的 CAPL0，可在 CAP0IF 中断时将 T16N\_CNT0 和 T16N\_PRECNT 清零；若 CAPL0 为零，则计数器继续累加。

当 T16N0IN1 端口的状态变化符合所设定的捕捉事件时，将 T16N\_CNT0 和 T16N\_PRECNT 的当前值分别装载到 T16N\_MAT1 和 T16N\_MAT3 寄存器中，产生 T16N 的 CAP1IF 中断。通过使能 T16N\_CON1 寄存器的 CAPL1，可在 CAP1IF 中断时将 T16N\_CNT0 和 T16N\_PRECNT 清零；若 CAPL1 为零，则计数器继续累加。

当 T16N\_CNT0 计数直到溢出时，仍未检测到设定的捕捉事件，T16N\_CNT0 的值被清零，并重新开始累加计数。

设置 T16N\_CON1 寄存器的 CAPPE 和 CAPNE，可选择 T16N0IN0 和 T16N0IN1 端口信号的捕捉事件。

选择捕捉上升沿：只设置 CAPPE 为 1；

选择捕捉下降沿：只设置 CAPNE 为 1；

选择捕捉上升沿/下降沿：设置 CAPPE 为 1，CAPNE 为 1。

设置 T16N\_CON1 寄存器的 CAPIS0，选择 T16N0IN0 是否作为捕捉输入端口；设置 CAPIS1，选择 T16N0IN1 是否作为捕捉输入端口；可同时选择两个端口作为捕捉输入端

口。

设置 T16N\_CON1 寄存器的 CAPT，可选择捕捉事件发生的次数。

举例说明：捕捉 T16N0IN0 端口上升沿/下降沿，捕捉 8 次；预分频设置为 1:1。

T16N\_CON0 寄存器的 MOD<1:0>=10，CS<1:0>=00，MAT0S<1:0>=00，

T16N\_CON1 寄存器的 CAPPE=1，CAPNE=1，CAPL0=1，CAPIS0=1，

CAPT<3:0>=0111。

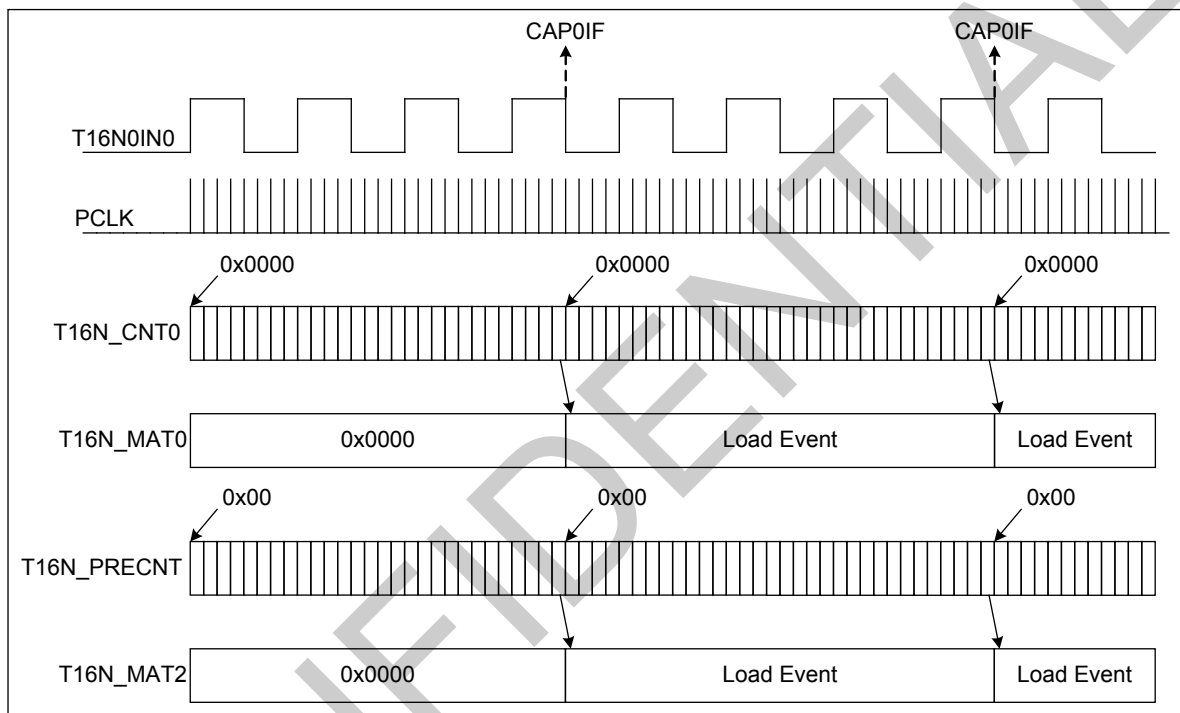


图 5-3 T16N0 捕捉功能示意图

在捕捉工作模式下，修改 T16N 预分频器计数匹配寄存器 T16N\_PREMAT 时，预分频器计数不会被清零。因此，首次捕捉可以从一个非零预分频器计数开始。当捕捉事件匹配发生时，产生的中断标志位必须通过软件清除，并及时读取捕捉到的 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 寄存器的值，在下次捕捉事件发生时，T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 寄存器会装载为新的计数值。

#### 5.1.1.5 T16N 输出调制功能

设置 T16N\_CON0 寄存器的 MOD<1:0>=11，使 T16N 工作在调制模式。

在调制工作模式下，需设置 T16N\_CON0 寄存器的 CS<1:0>=00，使 T16N\_CNT0/T16N\_CNT1 采用内部 PCLK 时钟源计数。

设置 T16N\_CON2 寄存器的 MOE0，选择 T16N0OUT0 是否使能为匹配输出端口，端口的输出状态通过 MOM0<1:0>和 MOM1<1:0>进行控制，由 T16N\_MAT0/T16N\_MAT1 匹配寄存器和 T16N\_CNT0 进行计数匹配。

设置 T16N\_CON2 寄存器的 MOE1, 选择 T16N0OUT1 是否使能为匹配输出端口, 端口的输出状态通过 PWMMOD<1:0>进行控制。在同步 PWM 模式下输出与 T16N0OUT0 相同的波形; 在互补 PWM 模式下输出与 T16N0OUT0 互补的波形; 在独立 PWM 模式下通过 MOM2<1:0>和 MOM3<1:0>进行控制, 由 T16N\_MAT2/T16N\_MAT3 匹配寄存器和 T16N\_CNT1 进行计数匹配。

设置 T16N\_CON2 寄存器的 MOM0/MOM1/MOM2/MOM3, 选择计数匹配发生时, 对 T16N0OUT0/T16N0OUT1 端口的影响: 保持, 清零, 置 1, 取反。

设置 T16N\_CON2 寄存器的 PWMMOD<1:0>, 两路 PWM 可配置为独立/同步/互补三种模式:

独立模式, T16N0OUT0/T16N0OUT1 输出不同的 PWM 波形。其中 T16N\_CNT0 匹配 T16N\_MAT0/T16N\_MAT1, 控制 T16N0OUT0 输出, PWM 周期由 T16N\_TOP0 设定, 为 T16N\_TOP0+1; 而 T16N\_CNT1 匹配 T16N\_MAT2/T16N\_MAT3, 控制 T16N0OUT1 输出, PWM 周期由 T16N\_TOP1 设定, 为 T16N\_TOP1+1。

同步模式, T16N0OUT0/T16N0OUT1 输出相同 PWM 波形, T16N\_CNT0 匹配 T16N\_MAT0/T16N\_MAT1, 控制 T16N0OUT0/T16N0OUT1 输出, PWM 周期由 T16N\_TOP0 设定, 为 T16N\_TOP0+1, 与 T16N\_CNT1 和 T16N\_TOP1 无关。

互补模式, T16N0OUT0/T16N0OUT1 输出为互补 PWM 波形, 死区时间宽度可配置, T16N\_CNT0 匹配 T16N\_MAT0/T16N\_MAT1, 控制 T16N0OUT0/T16N0OUT1 输出, PWM 周期由 T16N\_TOP0 设定, 为 T16N\_TOP0+1, 与 T16N\_CNT1 和 T16N\_TOP1 无关。

调制工作模式下支持匹配 0、匹配 1、匹配 2、匹配 3、峰值 0 及峰值 1 触发 ADC:

T16N\_MAT0、T16N\_MAT1、T16N\_MAT2、T16N\_MAT3、T16N\_TOP0 与 T16N\_CNT0 计数值匹配, 匹配时产生 ADC 触发信号;

T16N\_TOP1 与 T16N\_CNT1 计数值匹配, 匹配时产生 ADC 触发信号。

注 1: 调制模式下, 当寄存器位 REGBUF\_EN=1 时, T16N\_CNT0 计数值与缓冲器 T16N\_MATBUF0/1 进行比较匹配, T16N\_CNT1 计数值与缓冲器 T16N\_MATBUF2/3 进行比较匹配, T16N\_PRECNT 计数值与缓冲器 T16N\_PREBUF 进行比较匹配, PWM 周期由缓冲器 T16N\_TOPBUF0/1 设定。

注 2: 调制模式下, 当寄存器位 REGBUF\_EN=1 时, 则在 PWM 周期开始时, 预分频比缓冲器 T16N\_PREBUF, 计数匹配缓冲器 T16N\_MATBUF0/1/2/3 和峰值缓冲器 T16N\_TOPBUF0/1, 分别加载为预分频计数匹配寄存器 T16N\_PREMAT, 计数匹配寄存器 T16N\_MAT0/1/2/3 和峰值寄存器 T16N\_TOP0/1 的初始值; 在 PWM 波形输出过程中, 写入到寄存器 T16N\_PREMAT, T16N\_MAT0/1/2/3 或 T16N\_TOP0/1 中的值, 直到当前 PWM 周期结束时才会更新到对应的缓冲器中并生效。

举例说明:

1) 独立模式, 在 T16N0OUT0 和 T16N0OUT1 端口, 产生双边 PWM 波形。

T16N\_CON2 寄存器中:

MOE0=1, MOE1=1; T16N0OUT0 和 T16N0OUT1 匹配输出端口使能;

MOM0<1:0>=10; T16N\_MAT0 匹配, T16N0OUT0 输出高电平;



MOM1<1:0>=01; T16N\_MAT1 匹配, T16N0OUT0 输出低电平;  
MOM2<1:0>=10; T16N\_MAT2 匹配, T16N0OUT1 输出高电平;  
MOM3<1:0>=01; T16N\_MAT3 匹配, T16N0OUT1 输出低电平;  
PWMMOD<1:0>=00; 独立模式。

T16N\_CON0 寄存器中:  
MOD<1:0>=11; T16N 设置为调制输出;  
MAT0S<1:0>=11; T16N\_CNT0 继续计数, 并产生中断;  
MAT1S<1:0>=11; T16N\_CNT0 继续计数, 并产生中断;  
MAT2S<1:0>=00; T16N\_CNT1 继续计数, 不产生中断;  
MAT3S<1:0>=10; T16N\_CNT1 清零, 并重新计数。

设置其它寄存器:  
T16N\_MAT0 = 0x0002; T16N\_MAT1 = 0x0004;  
T16N\_MAT2 = 0x0006; T16N\_MAT3 = 0x0008;  
设置合适的 T16N\_TOP0/T16N\_TOP1 值。

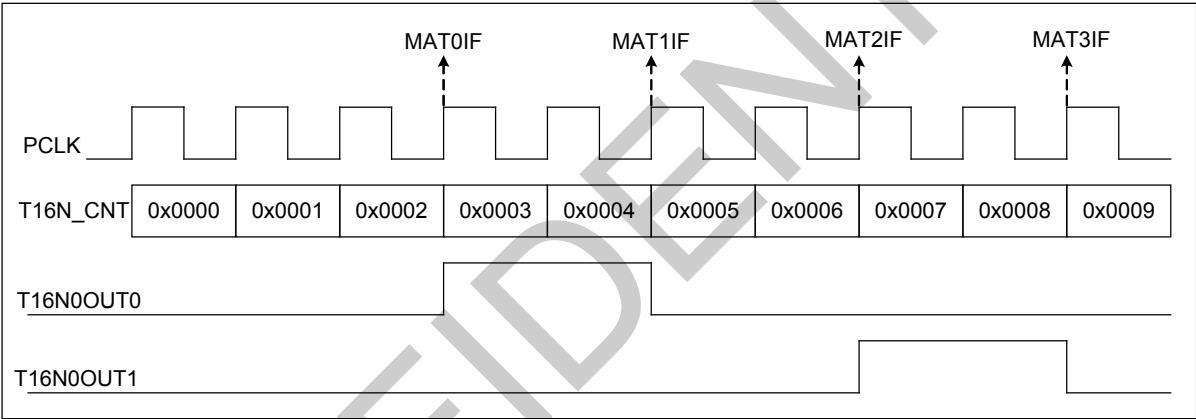


图 5-4 T16N0 独立 PWM 模式输出调制功能示意图

注: 若 T16N\_MAT0 和 T16N\_MAT1 设置了相同的值, 因 T16N\_MAT0 匹配优先级高于 T16N\_MAT1, T16N0OUT0 的输出电平取决于 MOM0 的设定。若 T16N\_MAT2 和 T16N\_MAT3 设置了相同的值, 因 T16N\_MAT2 匹配优先级高于 T16N\_MAT3, T16N0OUT1 的输出电平取决于 MOM2 的设定。

2) 互补模式, PCLK=48MHz, 在 T16N0OUT0 和 T16N0OUT1 端口, 产生 24MHz 互补 PWM 波形。

T16N\_CON2 寄存器中:  
MOE0=1, MOE1=1, 即 T16N0OUT0 和 T16N0OUT1 匹配输出端口使能;  
MOM0<1:0>=10, 即 T16N\_MAT0 匹配, T16N0OUT0 输出高电平, T16N0OUT1 输出低电平;  
MOM1<1:0>=01, 即 T16N\_MAT1 匹配, T16N0OUT0 输出低电平, T16N0OUT1 输出高电平;  
PWMMOD<1:0>=11, 即互补 PWM 模式; PWMDZE=0, 禁止死区时间

T16N\_CON0 寄存器中:

MOD<1:0>=11; T16N 设置为调制输出;  
 MAT0S<1:0>=11; T16N\_CNT0 继续计数, 并产生中断;  
 MAT1S<1:0>=11; T16N\_CNT0 继续计数, 并产生中断。

设置其它寄存器:

T16N\_MAT0 = 0x0000; T16N\_MAT1 = 0x0001; T16N\_TOP0=0x0001;  
 T16N\_PREMAT=0, 即不预分频。

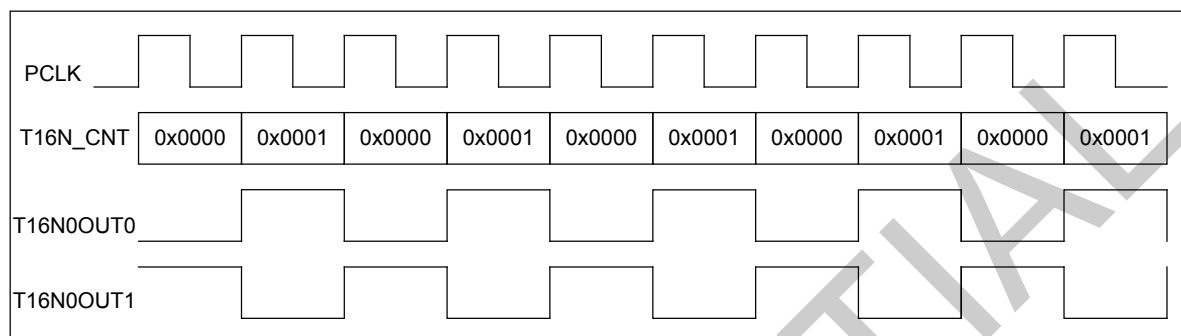


图 5-5 T16N0 互补 PWM 输出功能示意图

3) 带死区的互补模式, PCLK=48MHz, 在 T16N0OUT0 和 T16N0OUT1 端口, 产生带死区的互补 PWM 波形。

T16N\_CON2 寄存器中:

MOE0=1, MOE1=1, 即 T16N0OUT0 和 T16N0OUT1 匹配输出端口使能;

MOM0<1:0>=10, 即 T16N\_MAT0 匹配, T16N0OUT0 输出高电平, T16N0OUT1 输出低电平;

MOM1<1:0>=01, 即 T16N\_MAT1 匹配, T16N0OUT0 输出低电平, T16N0OUT1 输出高电平;

PWMMOD<1:0>=11, 即互补 PWM 模式; PWMDZE=1, 使能死区时间。

T16N\_CON0 寄存器中:

MOD<1:0>=11; T16N 设置为调制输出;

MAT0S<1:0>=11; T16N\_CNT0 继续计数, 并产生中断;

MAT1S<1:0>=11; T16N\_CNT0 继续计数, 并产生中断。

设置其它寄存器:

T16N\_MAT0, T16N\_MAT1 和 T16N\_TOP0 寄存器设定合适的值, 以设定 PWM 周期, 占空比等; T16N\_PREMAT 可设定为较小值, 如 0 值, 即不预分频, 使得死区时间和 PWM 占空比都有较高的精度。

如下图, 作为对比, 分别给出了无死区时间和带死区时间的 PWM 互补输出波形。需要注意的是, 死区时间会对 T16N0OUT0 和 T16N0OUT1 的高电平宽度作等量压缩。

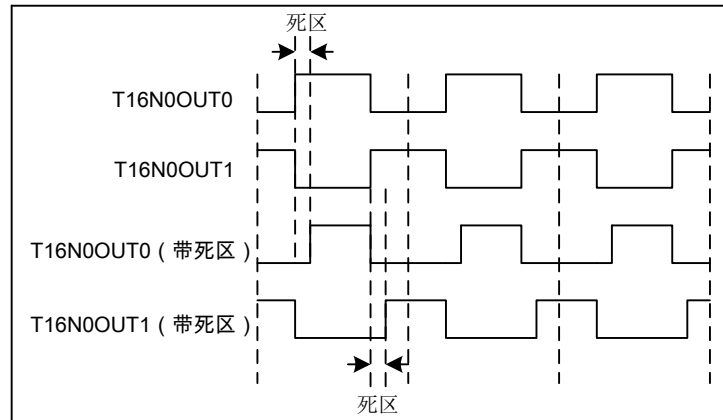


图 5-6 带死区的 T16N0 互补 PWM 输出功能示意图

#### 5.1.1.6 特殊功能寄存器

##### 5.1.1.7 T16N计数值寄存器 0 (T16N\_CNT0)

**T16N 计数值寄存器 0 (T16N\_CNT0)**

偏移地址: 00<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT0<15:0>															

—	bit31-16	—	—
CNT0<15:0>	bit 15-0	R/W	T16N_CNT0计数值

##### 5.1.1.8 T16N计数值寄存器 1 (T16N\_CNT1)

**T16N 计数值寄存器 1 (T16N\_CNT1)**

偏移地址: 04<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT1<15:0>															

—	bit31-16	—	—
CNT1<15:0>	bit 15-0	R/W	T16N_CNT1计数值

## 5.1.1.9 T16N预分频器计数值寄存器 (T16N\_PRECNT)

T16N 预分频器计数值寄存器 (T16N\_PRECNT)

偏移地址: 08<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PRECNT<7:0>							
—				bit31-8				—				—			
PRECNT<7:0>				bit7-0				R/W				T16N预分频器计数值			

## 5.1.1.10 T16N预分频器计数匹配寄存器 (T16N\_PREMAT)

T16N 预分频器计数匹配寄存器 (T16N\_PREMAT)

偏移地址: 0C<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PREMAT<7:0>							
—				bit31-8				—				—			
PREMAT<7:0>				bit7-0				R/W				预分频比例设置位 00: 预分频1: 1 01: 预分频1: 2 02: 预分频1: 3 ..... FE: 预分频1: 255 FF: 预分频1: 256			

## 5.1.1.11 T16N控制寄存器 0 (T16N\_CON0)

T16N 控制寄存器 0 (T16N\_CON0)

偏移地址: 10<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															ASYWEN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT3S<1:0>			MAT2S<1:0>			MAT1S<1:0>			MAT0S<1:0>			MOD<1:0>			EDGE<1:0>
—			bit 31~17			—			—			—			EN

ASYWEN	bit 16	R/W	<p><b>计数器写使能位</b></p> <p>0: 使能写 T16N_CNT1; 在同步计数模式, 定时模式和调制模式下, 使能写 T16N_CNT0 和 T16N_PRECNT, 在异步计数模式下, 不要在 ASYWEN=0 时, 写 T16N_CNT0 和 T16N_PRECNT, 否则有可能写操作失败</p> <p>1: 禁止写 T16N_CNT1; 在同步计数模式, 定时模式和调制模式下, 禁止写 T16N_CNT0 和 T16N_PRECNT, 在异步计数模式下, 使能写 T16N_CNT0 和 T16N_PRECNT</p>
MAT3S<1:0>	bit 15~14	R/W	<p><b>T16N_CNT0/1 匹配 T16N_MAT3 后的工作模式选择位</b></p> <p>00: T16N_CNT0/1 继续计数, 不产生中断</p> <p>01: T16N_CNT0/1 保持, 产生中断</p> <p>10: T16N_CNT0/1 清零并重新计数, 产生中断</p> <p>11: T16N_CNT0/1 继续计数, 产生中断 (T16N_CNT1 只在独立 PWM 调制模式下计数和匹配)</p>
MAT2S<1:0>	bit 13~12	R/W	<p><b>T16N_CNT0/1 匹配 T16N_MAT2 后的工作模式选择位</b></p> <p>00: T16N_CNT0/1 继续计数, 不产生中断</p> <p>01: T16N_CNT0/1 保持, 产生中断</p> <p>10: T16N_CNT0/1 清零并重新计数, 产生中断</p> <p>11: T16N_CNT0/1 继续计数, 产生中断 (T16N_CNT1 只在独立 PWM 调制模式下计数和匹配)</p>
MAT1S<1:0>	bit 11~10	R/W	<p><b>T16N_CNT0 匹配 T16N_MAT1 后的工作模式选择位</b></p> <p>00: T16N_CNT0 继续计数, 不产生中断</p> <p>01: T16N_CNT0 保持, 产生中断</p> <p>10: T16N_CNT0 清零并重新计数, 产生中断</p> <p>11: T16N_CNT0 继续计数, 产生中断</p>
MAT0S<1:0>	bit 9~8	R/W	<p><b>T16N_CNT0 匹配 T16N_MAT0 后的工作模式选择位</b></p> <p>00: T16N_CNT0 继续计数, 不产生中断</p> <p>01: T16N_CNT0 保持, 产生中断</p> <p>10: T16N_CNT0 清零并重新计数, 产生中断</p> <p>11: T16N_CNT0 继续计数, 产生中断</p>
MOD<1:0>	bit 7~6	R/W	<p><b>工作模式选择位</b></p> <p>00: 定时/计数模式</p> <p>01: 定时/计数模式</p> <p>10: 捕捉模式</p> <p>11: 调制模式</p>
EDGE<1:0>	bit 5~4	R/W	<p><b>外部时钟计数边沿选择位</b></p> <p>00: 上升沿计数</p> <p>01: 下降沿计数</p> <p>10: 上升沿/下降沿均计数 (仅同步计数模式)</p> <p>11: 上升沿/下降沿均计数 (仅同步计数模式)</p>

SYNC	bit 3	R/W	<b>外部时钟同步使能位</b> 0: 不同步外部时钟 T16N0CK0/T16N0CK1, 为异步计数模式 1: 通过 PCLK 对外部时钟 T16N0CK0/T16N0CK1 同步, 为同步计数模式, 外部时钟的高/低电平平均至少保持 2 个 PCLK 时钟周期
CS<1:0>	bit 2~1	R/W	<b>T16N 计数时钟源选择位</b> 00: 内部时钟 PCLK 01: 外部时钟 T16N0CK0 10: 外部时钟 T16N0CK1 11: 内部时钟 PCLK
EN	bit 0	R/W	<b>T16N 使能位</b> 0: 禁止 1: 使能

注 1: PCLK 为芯片内部外设模块时钟源, 时钟频率与芯片系统时钟频率相同。

注 2: 在定时/计数、PWM 调制模式(独立 PWM 模式除外)下, 计数器 T16N\_CNT0 的值与匹配寄存器 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 进行匹配, 计数器 T16N\_CNT1 不进行计数和匹配操作; 在独立 PWM 调制模式下, 即 PWMMOD=2'b0x, 则计数器 T16N\_CNT0 的值与 T16N\_MAT0 和 T16N\_MAT1 进行匹配, 计数器 T16N\_CNT1 的值与 T16N\_MAT2 和 T16N\_MAT3 进行匹配。

注 3: 在同步/互补 PWM 调制模式下, 即 PWMMOD=2'b1x 时, 建议设置 MAT3S=0, MAT2S=0, 以免影响计数器 T16N\_CNT0 的计数。

#### 5.1.1.12 T16N控制寄存器 1 (T16N\_CON1)

**T16N 控制寄存器 1 (T16N\_CON1)**

偏移地址: 14<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CAPL1	CAPL0	CAPT<3:0>				CAPIS1	CAPIS0	CAPNE	CAPPE

—	bit31-10	—	—
CAPL1	bit9	R/W	<b>捕捉 1 重载计数器使能位</b> 0: 禁止 1: 使能, T16N0IN1 端口的捕捉事件清零计数器 T16N_CNT0 和 T16N_PRECNT
CAPL0	bit8	R/W	<b>捕捉 0 重载计数器使能位</b> 0: 禁止 1: 使能, T16N0IN0 端口的捕捉事件清零计数器 T16N_CNT0 和 T16N_PRECNT
CAPT<3:0>	bit7-4	R/W	<b>捕捉次数控制位</b>

			0: 捕捉 1 次后, 产生装载动作 1: 捕捉 2 次后, 产生装载动作 2: 捕捉 3 次后, 产生装载动作 ..... F: 捕捉 16 次后, 产生装载动作
CAPIS1	bit3	R/W	捕捉输入端口 <b>T16N0IN1</b> 使能位 0: 禁止 1: 使能
CAPIS0	bit2	R/W	捕捉输入端口 <b>T16N0IN0</b> 使能位 0: 禁止 1: 使能
CAPNE	bit1	R/W	下降沿捕捉使能位 0: 禁止 1: 使能
CAPPE	bit0	R/W	上升沿捕捉使能位 0: 禁止 1: 使能

### 5.1.1.13 T16N控制寄存器 2 (T16N\_CON2)

T16N 控制寄存器 2 (T16N\_CON2)

偏移地址: 18<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							PWMBK	PWMB	PWMB	保留			PWMBK	PWMBK	PWMBK
							F	KP1	KP0				L1	L0	E1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOM3<1:	MOM2<1:	MOM1<1:	MOM0<1:0>				REGB	PWMD	PWMMOD<1:0>			POL1	POL0	MOE1	MOE0
0>	0>	0>					UF_EN	ZE							

—	31-25	—	—
PWMBKF	bit24	R/W	<b>PWM 刹车事件标志位</b> 0: 未发生刹车事件 1: 发生刹车事件, PWM 端口输出刹车电平; 软件写 1 清零, 清零后 PWM 端口输出恢复正常
PWMBKP1	bit23	R/W	<b>PWM 通道 1 刹车信号极性选择位</b> 0: 高电平刹车 1: 低电平刹车
PWMBKP0	bit22	R/W	<b>PWM 通道 0 刹车信号极性选择位</b> 0: 高电平刹车 1: 低电平刹车
—	bit21-20	—	—
PWMBKL1	bit19	R/W	<b>PWM 通道 1 刹车输出电平选择位</b>

			POL1=0 时: 0: 输出低电平 1: 输出高电平 POL1=1 时: 0: 输出高电平 1: 输出低电平
PWMBKL0	bit18	R/W	<b>PWM 通道 0 刹车输出电平选择位</b> POL0=0 时: 0: 输出低电平 1: 输出高电平 POL0=1 时: 0: 输出高电平 1: 输出低电平
PWMBKE1	bit17	R/W	<b>PWM 通道 1 刹车使能位</b> 0: 禁止 1: 使能
PWMBKE0	bit16	R/W	<b>PWM 通道 0 刹车使能位</b> 0: 禁止 1: 使能
MOM3<1:0>	bit15-14	R/W	<b>T16N_MAT3 匹配后的端口 1 工作模式选择位</b> （仅在独立调制模式下有效） 00: 匹配端口保持 01: 匹配端口清零 10: 匹配端口置 1 11: 匹配端口取反
MOM2<1:0>	bit13-12	R/W	<b>T16N_MAT2 匹配后的端口 1 工作模式选择位</b> （仅在独立调制模式下有效） 00: 匹配端口保持 01: 匹配端口清零 10: 匹配端口置 1 11: 匹配端口取反
MOM1<1:0>	bit11-10	R/W	<b>T16N_MAT1 匹配后的端口 0 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清零 10: 匹配端口置 1 11: 匹配端口取反
MOM0<1:0>	bit9-8	R/W	<b>T16N_MAT0 匹配后的端口 0 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清零 10: 匹配端口置 1 11: 匹配端口取反
REGBUF_EN	bit7	R/W	<b>缓冲寄存器使能位</b> 0: 禁止（在定时/计数模式下，需软件固定清 0） 1: 使能（仅在调制模式下，才可以软件置 1）



PWMDZE	bit6	R/W	<b>PWM 互补模式死区使能位</b> 0: 禁止 1: 使能
PWMMOD<1:0>	bit5-4	R/W	<b>PWM 模式选择位</b> 0x: 独立 10: 同步 11: 互补
POL1	bit3	R/W	<b>T16N0OUT1 输出极性选择位</b> 0: 正极性 1: 反极性
POL0	bit2	R/W	<b>T16N0OUT0 输出极性选择位</b> 0: 正极性 1: 反极性
MOE1	bit1	R/W	<b>输出端口 1 使能位</b> 0: 禁止 1: 使能
MOE0	bit0	R/W	<b>输出端口 0 使能位</b> 0: 禁止 1: 使能

注 1: 刹车信号可通过 GPIO\_PAFUNC 寄存器配置为 PA2/PA3/PA27/PA16 中的其中一个, 而且只能选一个。

注 2: 在刹车使能的情况下, PWMBKF 标志位置 1 时, 会硬件自动清零对应的输出端口使能位 MOE, 该标志位清零后, 还需软件重新设置使能位 MOE 为 1, PWM 端口输出才会恢复正常。

#### 5.1.1.14 T16N中断使能寄存器 (T16N\_IE)

##### T16N 中断使能寄存器 (T16N\_IE)

偏移地址: 20<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						PBK1IE	PBK0IE	CAP1IE	CAP0IE	TOP1IE	TOP0IE	MAT3IE	MAT2IE	MAT1IE	MAT0IE

—	bit31-10	—	—
PBK1IE	bit9	R/W	<b>PWM 通道 1 刹车中断使能位</b> 0: 禁止 1: 使能
PBK0IE	bit8	R/W	<b>PWM 通道 0 刹车中断使能位</b> 0: 禁止 1: 使能
CAP1IE	bit7	R/W	<b>输入端口 T16N0IN1 捕捉中断使能位</b> 0: 禁止

			1: 使能
CAP0IE	bit6	R/W	输入端口 T16N0IN0 捕捉中断使能位 0: 禁止 1: 使能
TOP1IE	bit5	R/W	T16N_CNT1 匹配计数峰值中断使能位 0: 禁止 1: 使能
TOP0IE	bit4	R/W	T16N_CNT0 匹配计数峰值中断使能位 0: 禁止 1: 使能
MAT3IE	bit3	R/W	匹配 3 中断使能位 0: 禁止 1: 使能
MAT2IE	bit2	R/W	匹配 2 中断使能位 0: 禁止 1: 使能
MAT1IE	bit1	R/W	匹配 1 中断使能位 0: 禁止 1: 使能
MAT0IE	bit0	R/W	匹配 0 中断使能位 0: 禁止 1: 使能

### 5.1.1.15 T16N中断标志寄存器 (T16N\_IF)

**T16N 中断标志寄存器 (T16N\_IF)**

偏移地址: 24<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						PBK1IF	PBK0IF	CAP1IF	CAP0IF	TOP1IF	TOP0IF	MAT3IF	MAT2IF	MAT1IF	MAT0IF

—	bit31-10	—	—
PBK1IF	bit9	R/W	<b>PWM 通道 1 刹车中断标志位</b> 0: 无刹车中断 1: 通道 1 刹车事件产生 软件写 1 清除标志位, 写 0 无效
PBK0IF	bit8	R/W	<b>PWM 通道 0 刹车中断标志位</b> 0: 无刹车中断 1: 通道 0 刹车事件产生 软件写 1 清除标志位, 写 0 无效
CAP1IF	bit7	R/W	输入端口 1 捕捉成功中断标志位

			0: 输入端口 1 捕捉未成功 1: 输入端口 1 捕捉成功 软件写 1 清除标志位, 写 0 无效
CAP0IF	bit6	R/W	<b>输入端口 0 捕捉成功中断标志位</b> 0: 输入端口 0 捕捉未成功 1: 输入端口 0 捕捉成功 软件写 1 清除标志位, 写 0 无效
TOP1IF	bit5	R/W	<b>T16N_CNT1 匹配计数峰值中断标志位</b> 0: 未匹配 1: 匹配峰值 软件写 1 清除标志位, 写 0 无效
TOP0IF	bit4	R/W	<b>T16N_CNT0 匹配计数峰值中断标志位</b> 0: 未匹配 1: 匹配峰值 软件写 1 清除标志位, 写 0 无效
MAT3IF	bit3	R/W	<b>匹配 3 中断标志位</b> 0: 计数器值与匹配寄存器 3 不相等 1: 计数器值与匹配寄存器 3 相等 软件写 1 清除标志位, 写 0 无效
MAT2IF	bit2	R/W	<b>匹配 2 中断标志位</b> 0: 计数器值与匹配寄存器 2 不相等 1: 计数器值与匹配寄存器 2 相等 软件写 1 清除标志位, 写 0 无效
MAT1IF	bit1	R/W	<b>匹配 1 中断标志位</b> 0: 计数器值与匹配寄存器 1 不相等 1: 计数器值与匹配寄存器 1 相等 软件写 1 清除标志位, 写 0 无效
MAT0IF	bit0	R/W	<b>匹配 0 中断标志位</b> 0: 计数器值与匹配寄存器 0 不相等 1: 计数器值与匹配寄存器 0 相等 软件写 1 清除标志位, 写 0 无效

注 1: 在定时/计数、PWM 调制模式(独立 PWM 模式除外)下, 计数器 T16N\_CNT0 的值与匹配寄存器 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 进行匹配, 计数器 T16N\_CNT1 不进行计数和匹配操作; 在独立 PWM 调制模式下, 即 PWMMOD=2'b0x, 则计数器 T16N\_CNT0 的值与 T16N\_MAT0 和 T16N\_MAT1 进行匹配, 计数器 T16N\_CNT1 的值与 T16N\_MAT2 和 T16N\_MAT3 进行匹配。

注 2: T16N 中断禁止时, 如果满足条件仍会置起对应的中断标志位, 只是不会产生中断请求。

注 3: 对 T16N\_IF 寄存器的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

注 4: 刹车使能时, 即 PWMBKE0 或 PWMBKE1 置 1, T16N\_IF 寄存器中的 PBK0IF 或 PBK1IF 中断标志位在刹车事件产生时才被置 1。

### 5.1.1.16 PWM调制模式死区宽度寄存器 (T16N\_PDZ)

PWM 调制模式死区宽度寄存器 (T16N\_PDZ)

偏移地址: 28<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-8	—	—
PDZ<7:0>	bit7-0	R/W	<b>PWM调制模式死区宽度寄存器</b> 0x00: 死区为1个计数周期 0x01: 死区为2个计数周期 ..... 0xFF: 死区为 256 个计数周期

### 5.1.1.17 PWM调制模式ADC触发寄存器 (T16N\_PTR)

PWM 调制模式 ADC 触发寄存器 (T16N\_PTR)

偏移地址: 2C<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								P1TOP1TRE	P1MAT3TRE	P1MAT2TRE	保留	P0TOP0TRE	P0MAT1TRE	P0MAT0TRE	保留

—	bit31-8	—	—
P1TOP1TRE	bit7	R/W	<b>PWM 通道 1 峰值 1 触发使能</b> 0: 禁止 1: 使能触发 ADC 转换
P1MAT3TRE	bit6	R/W	<b>PWM 通道 1 匹配 3 触发使能</b> 0: 禁止 1: 使能触发 ADC 转换 (仅 MAT3S#00 时有效)
P1MAT2TRE	bit5	R/W	<b>PWM 通道 1 匹配 2 触发使能</b> 0: 禁止 1: 使能触发 ADC 转换 (仅 MAT3S#00 时有效)
—	bit4	—	—
P0TOP0TRE	bit3	R/W	<b>PWM 通道 0 峰值 0 触发使能</b> 0: 禁止 1: 使能触发 ADC 转换
P0MAT1TRE	bit2	R/W	<b>PWM 通道 0 匹配 1 触发使能</b> 0: 禁止 1: 使能触发 ADC 转换 (仅 MAT3S#00 时有效)

P0MAT0TRE	bit1	R/W	<b>PWM 通道 0 匹配 0 触发使能</b> 0: 禁止 1: 使能触发 ADC 转换 (仅 MAT3S#00 时有效)
—	bit0	—	—

#### 5.1.1.18 T16N计数匹配寄存器 0 (T16N\_MAT0)

##### T16N 计数匹配寄存器 0 (T16N\_MAT0)

偏移地址: 30<sub>H</sub>

复位值: 00000000\_00000000\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT0<15:0>															

—	bit31-16	—	—
MAT0<15:0>	bit15-0	R/W	T16N计数匹配值0

#### 5.1.1.19 T16N计数匹配寄存器 1 (T16N\_MAT1)

##### T16N 计数匹配寄存器 1 (T16N\_MAT1)

偏移地址: 34<sub>H</sub>

复位值: 00000000\_00000000\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT1<15:0>															

—	bit31-16	—	—
MAT1<15:0>	bit15-0	R/W	T16N计数匹配值1

#### 5.1.1.20 T16N计数匹配寄存器 2 (T16N\_MAT2)

##### T16N 计数匹配寄存器 2 (T16N\_MAT2)

偏移地址: 38<sub>H</sub>

复位值: 00000000\_00000000\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT2<15:0>															

—	bit31-16	—	—
MAT2<15:0>	bit15-0	R/W	T16N计数匹配值2

### 5.1.1.21 T16N计数匹配寄存器 3 (T16N\_MAT3)

#### T16N 计数匹配寄存器 3 (T16N\_MAT3)

偏移地址: 3C<sub>H</sub>

复位值: 00000000\_00000000\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT3<15:0>															

—	bit31-16	—	—
MAT3<15:0>	bit15-0	R/W	T16N计数匹配值3

### 5.1.1.22 T16N\_CNT0 计数峰值寄存器 0 (T16N\_TOP0)

#### T16N\_CNT0 计数峰值寄存器 0 (T16N\_TOP0)

偏移地址: 40<sub>H</sub>

复位值: 00000000\_00000000\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOP0<15:0>															

—	bit31-16	—	—
TOP0<15:0>	bit15-0	R/W	T16N_CNT0计数峰值寄存器0

### 5.1.1.23 T16N\_CNT1 计数峰值寄存器 1 (T16N\_TOP1)

#### T16N\_CNT1 计数峰值寄存器 1 (T16N\_TOP1)

偏移地址: 44<sub>H</sub>

复位值: 00000000\_00000000\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOP1<15:0>															

—	bit31-16	—	—
TOP1<15:0>	bit15-0	R/W	T16N_CNT1计数峰值寄存器1

### 5.1.1.24 T16N应用说明

芯片支持 4 个 16 位定时器/计数器, 分别为 T16N0, T16N1, T16N2 和 T16N3。

## 5.1.2 32 位定时器/计数器T32N（T32N0）

### 5.1.2.1 概述

- ◆ 1 个 8 位可配置预分频计数器，所产生分频时钟作为 T32N\_CNT 计数器的定时或计数时钟
  - ◇ 预分频时钟源可选：PCLK 或 T32N0CK0/T32N0CK1
  - ◇ 预分频计数器可由 T32N\_PRECNT 寄存器设定计数初值
  - ◇ 分频比由 T32N\_PREMAT 寄存器设定
- ◆ 1 个 32 位可配置定时/计数寄存器 T32N\_CNT
- ◆ 可配置定时/计数工作模式
  - ◇ 支持 4 组 32 位计数匹配寄存器 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3，计数匹配后支持下列操作：
    - 产生中断
    - 支持 T32N\_CNT 计数寄存器三种操作：保持，清零，或继续计数
    - 支持 T32N0OUT0/T32N0OUT1 端口四种操作：保持，清零，置 1 或取反
- ◆ 支持输入捕捉功能
  - ◇ 支持捕捉边沿可配置
  - ◇ 支持捕捉次数可配置
- ◆ 支持输出调制功能 PWM

### 5.1.2.2 结构框图

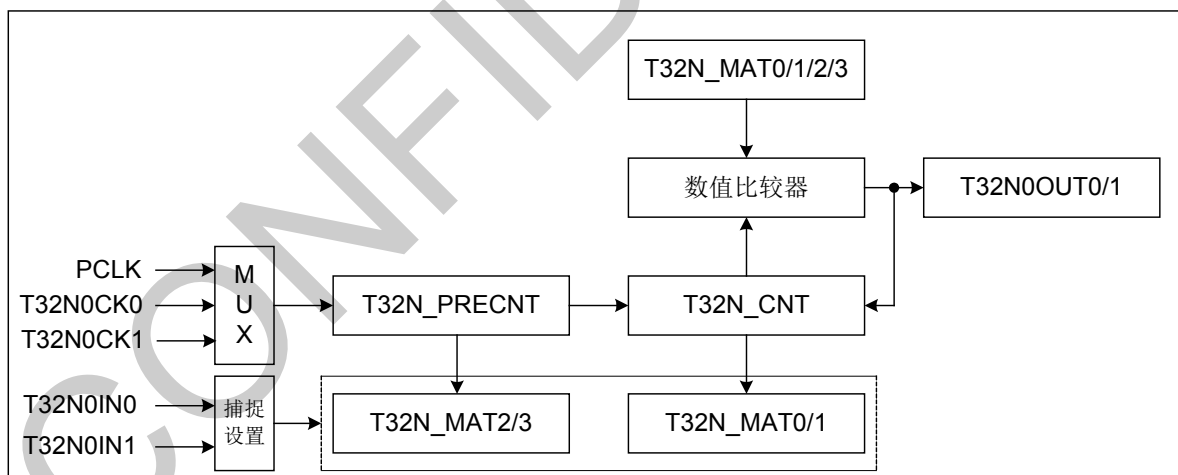


图 5-7 T32N0 电路结构框图

### 5.1.2.3 T32N定时/计数功能

设置 T32N\_CON0 寄存器的 MOD<1:0> =00 或 01，T32N 工作在定时/计数模式。

设置 T32N\_CON0 寄存器的 EN=1，使能 T32N，计数值寄存器 T32N\_CNT 从预设值开始累加计数。

设置 T32N\_CON0 寄存器的 CS，选择计数时钟源。时钟源为内部时钟 PCLK 时，为定时模式；时钟源为外部时钟 T32N0CK0/T32N0CK1 端口输入时，为计数模式。

设置 T32N\_CON0 寄存器的 SYNC，选择外部时钟 T32N0CK0/T32N0CK1 是否被内部时钟 PCLK 同步。当选择外部时钟被同步时，为同步计数模式，否则为异步计数模式。同步计数模式时，T32N0CK0/T32N0CK1 端口输入的高/低电平脉宽均必须大于 2 个 PCLK 时钟周期。

设置 T32N\_CON0 寄存器的 EDGE，选择外部时钟计数方式：上升沿计数，下降沿计数，或上升/下降沿均计数，其中上升/下降沿均计数只适用于同步计数模式。

设置 T32N\_CON0 寄存器的 MAT0S/MAT1S/MAT2S/MAT3S，选择计数匹配后 T32N\_CNT 计数值寄存器的工作状态。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=00：当 T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 时，继续累加计数，不产生中断，当计数到 0xFFFFFFFF 后，下一次累加计数溢出，T32N\_CNT 的值为 0x00000000，并产生中断，重新开始累加计数。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=01：当 T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 时，计数值将保持，即在下一个计数时钟（经过预分频之后的时钟）到来时，T32N\_CNT 不再累加计数，只产生中断。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=10：当 T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 时，计数值在下一个计数时钟（经过预分频之后的时钟）到来时被清零，并产生中断，重新开始累加计数。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=11：当 T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 时，继续累加计数，并在下一个计数时钟（经过预分频之后的时钟）到来时，产生中断，当计数到 0xFFFFFFFF 后，下一次累加计数溢出，T32N\_CNT 的值为 0x00000000，并产生中断，重新开始累加计数。对设置的多个不同匹配值 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3，当计数值匹配到每一个匹配值时，会产生相应的中断。由于中断产生后，T32N 继续累加计数，若未及时读取匹配中断标志位并软件清除，则有可能后续的匹配中断也已产生，从而会同时读到多个有效的匹配中断标志。

如果 T32N\_CNT 的计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3，当 T32N\_CON1 寄存器的 MOE0=1 时，T32N0OUT0 端口输出状态由 MOM0/MOM1<1:0>位控制；MOE1=1 时，T32N0OUT1 端口输出状态由 MOM2/MOM3<1:0>位控制。可选择 T32N0OUT0/T32N0OUT1 端口同时输出。

举例说明：T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 后的工作方式

T32N\_CON0 寄存器中

MAT0<31:0>=0x00000002， MAT0S<1:0>=00，继续计数，不产生中断；

MAT1<31:0>=0x00000004， MAT1S<1:0>=11，继续计数，产生中断；

MAT2<31:0>=0x00000006， MAT2S<1:0>=10，清零，产生中断，重新计数。

预分频设置为 1:1，采用内部 PCLK 时钟源。计数匹配功能示意图如下所示：



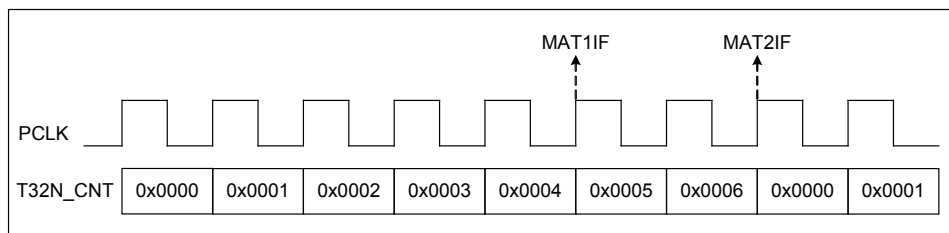


图 5-8 T32N0 计数匹配功能示意图

#### 5.1.2.4 T32N输入捕捉功能

设置 T32N\_CON0 寄存器的 MOD<1:0>=10，使 T32N 工作在捕捉模式。

在捕捉工作模式下，需设置 T32N\_CON0 寄存器的 CS<1:0>=00，使 T32N\_CNT 采用内部 PCLK 时钟源计数；并且设置 T32N\_CON0 寄存器的 MAT0S/MAT1S/MAT2S/MAT3S <1:0>=00，计数匹配不影响 T32N\_CNT 的工作。

在捕捉工作模式下，对端口 T32N0IN0 和 T32N0IN1 的状态进行检测。

当 T32N0IN0 端口的状态变化符合所设定的捕捉事件时，将 T32N\_CNT 和 T32N\_PRECNT 的当前值分别装载到 T32N\_MAT0 和 T32N\_MAT2 寄存器中，产生 T32N 的 CAP0IF 中断。通过使能 T32N\_CON1 寄存器的 CAPL0，可在 CAP0IF 中断时将 T32N\_CNT 和 T32N\_PRECNT 清零；若 CAPL0 为零，则计数器继续累加计数。

当 T32N0IN1 端口的状态变化符合所设定的捕捉事件时，将 T32N\_CNT 和 T32N\_PRECNT 的当前值分别装载到 T32N\_MAT1 和 T32N\_MAT3 寄存器中，产生 T32N 的 CAP1IF 中断，通过使能 T32N\_CON1 寄存器的 CAPL1，可在 CAP1IF 中断时将 T32N\_CNT 和 T32N\_PRECNT 清零；若 CAPL1 为零，则计数器继续累加计数。

当 T32N\_CNT 计数直到溢出时，仍未检测到设定的捕捉事件，T32N\_CNT 的值被清零，并重新开始累加计数。

设置 T32N\_CON1 寄存器的 CAPPE 和 CAPNE，可选择 T32N0IN0 和 T32N0IN1 端口信号的捕捉事件：捕捉上升沿，捕捉下降沿，捕捉上升沿/下降沿。

设置 T32N\_CON1 寄存器的 CAPIS0，选择 T32N0IN0 是否作为捕捉输入端口；设置 CAPIS1，选择 T32N0IN1 是否作为捕捉输入端口；可同时选择两个端口作为捕捉输入端口。

设置 T32N\_CON1 寄存器中的 CAPT，可选择捕捉事件发生的次数。

当 MOE0=1 时，如果 T32N0IN0 端口捕捉事件发生，T32N0OUT0 端口输出翻转；

当 MOE1=1 时，如果 T32N0IN1 端口捕捉事件发生，T32N0OUT1 端口输出翻转。

举例说明：捕捉 T32N0IN0 端口上升沿/下降沿，捕捉 8 次；预分频设置为 1:1。

T32N\_CON0 寄存器中

MOD<1:0>=10, CS<1:0>=00, MAT0S<1:0>=00;

T32N\_CON1 寄存器中

CAPPE=1, CAPNE=1, CAPL0=1, CAPIS0=1, CAPT<3:0>=0111。

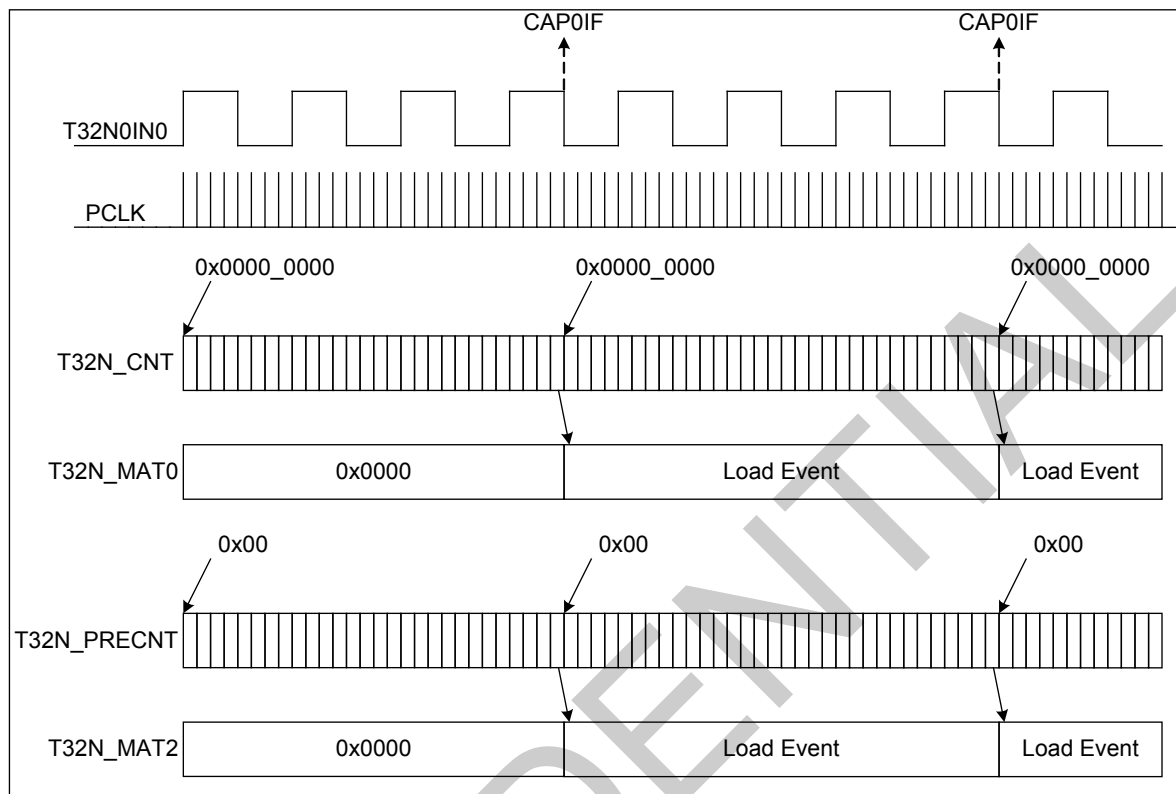


图 5-9 T32N0 捕捉功能示意图

在捕捉工作模式下，修改 T32N 预分频器计数匹配寄存器 T32N\_PREMAT 时，预分频器计数不会被清零。因此，首次捕捉可以从一个非零预分频器计数开始。当捕捉事件匹配发生时，产生的中断标志位必须通过软件清除，并及时读取捕捉到的 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 寄存器的值，在下一次捕捉事件发生时，T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 寄存器会装载为新的计数值。

#### 5.1.2.5 T32N 输出调制功能

设置 T32N\_CON0 寄存器中的 MOD<1:0>=11，使 T32N 工作在调制模式。

在调制工作模式下，需设置 T32N\_CON0 寄存器的 CS<1:0>=00，使 T32N\_CNT 采用内部 PCLK 时钟源计数。

设置 T32N\_CON1 寄存器的 MOE0，选择 T32N0OUT0 是否使能为匹配输出端口，端口的输出状态通过 MOM0<1:0>和 MOM1<1:0>进行控制，由 T32N\_MAT0/T32N\_MAT1 匹配寄存器和 T32N\_CNT 进行计数匹配；设置 MOE1，选择 T32N0OUT1 是否使能为匹配输出端口，端口的输出状态通过 MOM2<1:0>和 MOM3<1:0>进行控制，由 T32N\_MAT2/T32N\_MAT3 匹配寄存器和 T32N\_CNT 进行计数匹配。

设置 MOM0/MOM1/MOM2/MOM3，选择计数匹配发生时，对 T32N0OUT0/T32N0OUT1 端口的影响：保持，清零，置 1，取反。

举例说明：在 T32N0OUT0 和 T32N0OUT1 端口，产生双边 PWM 波形。

MOE0=1, MOE1=1; T32N0OUT0 和 T32N0OUT1 匹配输出端口使能;

MOM0<1:0>=10; T32N\_MAT0 匹配, T32N0OUT0 输出高电平;

MOM1<1:0>=01; T32N\_MAT1 匹配, T32N0OUT0 输出低电平;

MOM2<1:0>=10; T32N\_MAT2 匹配, T32N0OUT1 输出高电平;

MOM3<1:0>=01; T32N\_MAT3 匹配, T32N0OUT1 输出低电平;

MAT0 = 0x00000002; T32N\_MAT1 = 0x00000004;

MAT2 = 0x00000006; T32N\_MAT3 = 0x00000008;

MOD<1:0>=11; T32N 设置为调制输出

MAT0S<1:0>=11; T32N\_CNT 继续计数, 并产生中断

MAT1S<1:0>=11; T32N\_CNT 继续计数, 并产生中断

MAT2S<1:0>=11; T32N\_CNT 继续计数, 并产生中断

MAT3S<1:0>=10; T32N\_CNT 清零, 并产生中断

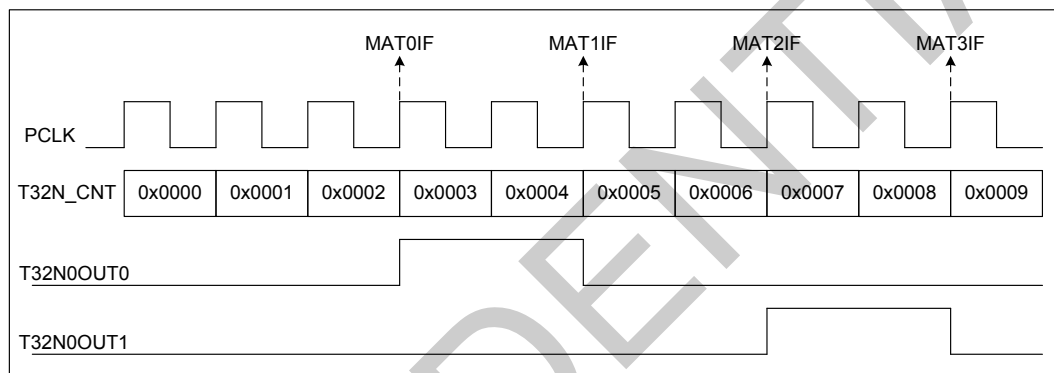


图 5-10 T32N0 输出调制功能示意图

注：若 T32N\_MAT0 和 T32N\_MAT1 设置了相同的值，因 T32N\_MAT0 匹配优先级高于 T32N\_MAT1，T32N0OUT0 的输出电平取决于 MOM0 的设定。若 T32N\_MAT2 和 T32N\_MAT3 设置了相同的值，因 T32N\_MAT2 匹配优先级高于 T32N\_MAT3，T32N0OUT1 的输出电平取决于 MOM2 的设定。

#### 5.1.2.6 特殊功能寄存器

#### 5.1.2.7 T32N计数值寄存器 (T32N\_CNT)

T32N 计数值寄存器 (T32N\_CNT)

偏移地址: 00H

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT<15:0>															
CNT<31:0>				bit 31-0				R/W				T32N计数值			

## 5.1.2.8 T32N控制寄存器 0 (T32N\_CON0)

T32N 控制寄存器 0 (T32N\_CON0)

偏移地址: 04<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															ASYNCWREN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT3S<1:0>	MAT2S<1:0>	MAT1S<1:0>	MAT0S<1:0>	MOD<1:0>	EDGE<1:0>	SYNC	CS<1:0>	EN							

—	bit31-17	—	—
ASYNC_WREN	bit16	R/W	外部时钟异步计数模式下, 对计数器的写使能位 0: 禁止写 T32N_CNT 和 T32N_PRECNT, 如果强制写, 有可能写操作不成功 (为避免对计数器的写操作错误, 不建议用户对该位写 0) 1: 使能写 T32N_CNT 和 T32N_PRECNT
MAT3S<1:0>	bit15-14	R/W	<b>T32N_CNT 匹配 T32N_MAT3 后的工作模式选择位</b> 00: T32N_CNT 继续计数, 不产生中断 01: T32N_CNT 保持, 产生中断 10: T32N_CNT 清零并重新计数, 产生中断 11: T32N_CNT 继续计数, 产生中断
MAT2S<1:0>	bit13-12	R/W	<b>T32N_CNT 匹配 T32N_MAT2 后的工作模式选择位</b> 00: T32N_CNT 继续计数, 不产生中断 01: T32N_CNT 保持, 产生中断 10: T32N_CNT 清零并重新计数, 产生中断 11: T32N_CNT 继续计数, 产生中断
MAT1S<1:0>	bit11-10	R/W	<b>T32N_CNT 匹配 T32N_MAT1 后的工作模式选择位</b> 00: T32N_CNT 继续计数, 不产生中断 01: T32N_CNT 保持, 产生中断 10: T32N_CNT 清零并重新计数, 产生中断 11: T32N_CNT 继续计数, 产生中断
MAT0S<1:0>	bit9-8	R/W	<b>T32N_CNT 匹配 T32N_MAT0 后的工作模式选择位</b> 00: T32N_CNT 继续计数, 不产生中断 01: T32N_CNT 保持, 产生中断 10: T32N_CNT 清零并重新计数, 产生中断 11: T32N_CNT 继续计数, 产生中断
MOD<1:0>	bit7-6	R/W	工作模式选择位 00: 定时/计数模式 01: 定时/计数模式 10: 捕捉模式 11: 调制模式
EDGE<1:0>	bit5-4	R/W	外部时钟计数边沿选择位 00: 上升沿计数 01: 下降沿计数

			10: 上升沿/下降沿均计数 (仅同步计数模式) 11: 上升沿/下降沿均计数 (仅同步计数模式)
SYNC	bit3	R/W	<b>外部时钟同步使能位</b> 0: 不同步外部时钟 T32N0CK0/T32N0CK1, 为异步计数模式 1: 通过 PCLK 对外部时钟 T32N0CK0/T32N0CK1 同步, 为同步计数模式, 外部时钟的高/低电平至少保持 2 个 PCLK 时钟周期
CS<1:0>	bit2-1	R/W	<b>T32N 计数时钟源选择位</b> 00: 内部时钟 PCLK 01: 外部时钟 T32N0CK0 时钟输入 10: 外部时钟 T32N0CK1 时钟输入 11: 内部时钟 PCLK
EN	bit0	R/W	<b>T32N使能位</b> 0: 禁止 1: 使能

### 5.1.2.9 T32N控制寄存器 1 (T16N\_CON1)

**T32N 控制寄存器 1 (T32N\_CON1)**

偏移地址: 08<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MOM3<1:0>	MOM2<1:0>	MOM1<1:0>	MOM0<1:0>	保留										MOE1	MOE0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CAPL1	CAPL0	CAPT<3:0>				CAPIS1	CAPIS0	CAPNE	CAPPE

MOM3<1:0>	bit31-30	R/W	<b>T32N_MAT3 匹配后的端口 1 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清零 10: 匹配端口置 1 11: 匹配端口取反
MOM2<1:0>	bit29-28	R/W	<b>T32N_MAT2 匹配后的端口 1 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清零 10: 匹配端口置 1 11: 匹配端口取反
MOM1<1:0>	bit27-26	R/W	<b>T32N_MAT1 匹配后的端口 0 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清零 10: 匹配端口置 1 11: 匹配端口取反
MOM0<1:0>	bit25-24	R/W	<b>T32N_MAT0 匹配后的端口 0 工作模式选择位</b>

			00: 匹配端口保持 01: 匹配端口清零 10: 匹配端口置 1 11: 匹配端口取反
—	bit23-18	—	—
MOE1	bit17	R/W	输出端口 1 使能位 0: 禁止 1: 使能
MOE0	bit16	R/W	输出端口 0 使能位 0: 禁止 1: 使能
—	bit15-10	—	—
CAPL1	bit9	R/W	捕捉 1 重载计数器使能位 0: 禁止 1: 使能, T32N0IN1 端口的捕捉事件清零计数器 T32N_CNT 和 T32N_PRECNT
CAPL0	bit8	R/W	捕捉 0 重载计数器使能位 0: 禁止 1: 使能, T32N0IN0 端口的捕捉事件清零计数器 T32N_CNT 和 T32N_PRECNT
CAPT<3:0>	bit7-4	R/W	捕捉次数控制位 0: 捕捉 1 次后, 产生装载动作 1: 捕捉 2 次后, 产生装载动作 2: 捕捉 3 次后, 产生装载动作 ..... F: 捕捉 16 次后, 产生装载动作
CAPIS1	bit3	R/W	捕捉输入端口 1 使能位 0: 禁止 1: 使能
CAPIS0	bit2	R/W	捕捉输入端口 0 使能位 0: 禁止 1: 使能
CAPNE	bit1	R/W	下降沿捕捉使能位 0: 禁止 1: 使能
CAPPE	bit0	R/W	上升沿捕捉使能位 0: 禁止 1: 使能

## 5.1.2.10 T32N预分频器计数值寄存器 (T32N\_PRECNT)

T32N 预分频器计数值寄存器 (T32N\_PRECNT)

偏移地址: 10<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-8	—	—
PRECNT<7:0>	bit7-0	R/W	T32N预分频器计数值

## 5.1.2.11 T32N预分频器计数匹配寄存器 (T32N\_PREMAT)

T32N 预分频器计数匹配寄存器 (T32N\_PREMAT)

偏移地址: 14<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-8	—	—
PREMAT<7:0>	bit7-0	R/W	预分频比例设置 00: 预分频1: 1 01: 预分频1: 2 02: 预分频1: 3 ..... FE: 预分频1: 255 FF: 预分频1: 256

## 5.1.2.12 T32N中断使能寄存器 (T32N\_IE)

T32N 中断使能寄存器 (T32N\_IE)

偏移地址: 18<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CAP1IE	CAP0IE	IE	MAT3IE	MAT2IE	MAT1IE	MAT0IE	

—	bit31-7	—	—
---	---------	---	---

CAP1IE	bit6	R/W	输入端口 1 捕捉中断使能位 0: 禁止 1: 使能
CAP0IE	bit5	R/W	输入端口 0 捕捉中断使能位 0: 禁止 1: 使能
IE	bit4	R/W	匹配 0xFFFFFFFF 中断使能位 0: 禁止 1: 使能
MAT3IE	bit3	R/W	匹配 3 中断使能位 0: 禁止 1: 使能
MAT2IE	bit2	R/W	匹配 2 中断使能位 0: 禁止 1: 使能
MAT1IE	bit1	R/W	匹配 1 中断使能位 0: 禁止 1: 使能
MAT0IE	bit0	R/W	匹配 0 中断使能位 0: 禁止 1: 使能

### 5.1.2.13 T32N中断标志寄存器 (T32N\_IF)

T32N 中断标志寄存器 (T32N\_IF)

偏移地址: 1C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									CAP1IF	CAP0IF	IF	MAT3IF	MAT2IF	MAT1IF	MAT0IF

—	bit31-7	—	—
CAP1IF	bit6	R/W	输入端口 1 捕捉成功中断标志位 0: 输入端口 1 捕捉未成功 1: 输入端口 1 捕捉成功 软件写 1 清除标志位, 写 0 无效
CAP0IF	bit5	R/W	输入端口 0 捕捉成功中断标志位 0: 输入端口 0 捕捉未成功 1: 输入端口 0 捕捉成功 软件写 1 清除标志位, 写 0 无效
IF	bit4	R/W	匹配 0xFFFFFFFF 中断标志位 0: 计数器值不等于 0xFFFFFFFF



			1: 计数器值等于 0xFFFFFFFF 软件写 1 清除标志位, 写 0 无效
MAT3IF	bit3	R/W	<b>匹配 3 中断标志位</b> 0: 计数器值与匹配寄存器 3 不相等 1: 计数器值与匹配寄存器 3 相等 软件写 1 清除标志位, 写 0 无效
MAT2IF	bit2	R/W	<b>匹配 2 中断标志位</b> 0: 计数器值与匹配寄存器 2 不相等 1: 计数器值与匹配寄存器 2 相等 软件写 1 清除标志位, 写 0 无效
MAT1IF	bit1	R/W	<b>匹配 1 中断标志位</b> 0: 计数器值与匹配寄存器 1 不相等 1: 计数器值与匹配寄存器 1 相等 软件写 1 清除标志位, 写 0 无效
MAT0IF	bit0	R/W	<b>匹配 0 中断标志位</b> 0: 计数器值与匹配寄存器 0 不相等 1: 计数器值与匹配寄存器 0 相等 软件写 1 清除标志位, 写 0 无效

注 1: 在定时/计数, 调制模式下, 计数器 T32N\_CNT 的值与匹配寄存器 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 进行匹配。

注 2: T32N 中断禁止时, 如果满足条件仍会置起对应的中断标志位, 只是不会产生中断请求。

注 3: 对 T32N\_IF 寄存器的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

#### 5.1.2.14 T32N计数匹配寄存器 0 (T32N\_MAT0)

**T32N 计数匹配寄存器 0 (T32N\_MAT0)**

偏移地址: 20<sub>H</sub>

复位值: 11111111\_11111111\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAT0<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT0<15:0>															
MAT0<31:0>		bit31-0		R/W		T32N计数匹配值0									

### 5.1.2.15 T32N计数匹配寄存器 1 (T32N\_MAT1)

#### T32N 计数匹配寄存器 1 (T32N\_MAT1)

偏移地址: 24<sub>H</sub>

复位值: 11111111\_11111111\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAT1<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT1<15:0>															

MAT1<31:0>	bit31-0	R/W	T32N计数匹配值1
------------	---------	-----	------------

### 5.1.2.16 T32N计数匹配寄存器 2 (T32N\_MAT2)

#### T32N 计数匹配寄存器 2 (T32N\_MAT2)

偏移地址: 28<sub>H</sub>

复位值: 11111111\_11111111\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAT2<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT2<15:0>															

MAT2<31:0>	bit31-0	R/W	T32N计数匹配值2
------------	---------	-----	------------

### 5.1.2.17 T32N计数匹配寄存器 3 (T32N\_MAT3)

#### T32N 计数匹配寄存器 3 (T32N\_MAT3)

偏移地址: 2C<sub>H</sub>

复位值: 11111111\_11111111\_11111111\_11111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAT3<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT3<15:0>															

MAT3<31:0>	bit31-0	R/W	T32N计数匹配值3
------------	---------	-----	------------

### 5.1.2.18 T32N应用说明

芯片支持 1 个 32 位定时器/计数器 T32N0。

## 5.2 通用异步接收/发送器（UART0/ UART1/UART2）

以 UART0 为例，UART2/UART1 参考 UART0

### 5.2.1 概述

- ◇ 支持异步接收和异步发送
- ◇ 支持内置波特率发生器，支持 4 位小数波特率和 12 位整数波特率
- ◇ 兼容 RS-232/RS-442/RS-485 的通讯接口
- ◇ 支持全/半双工通讯模式
- ◇ 异步接收器
  - 支持独立的接收移位寄存器
  - 支持硬件自动波特率检测
  - 支持 7/8/9 位数据格式可配，支持奇偶校验功能可配
  - 支持硬件自动奇偶校验位判断
  - 支持空闲帧检测
  - 支持 3 类接收错误中断：接收 FIFO 溢出错误、奇偶校验错误、帧结束错误
- ◇ 异步发送器
  - 支持独立的发送移位寄存器
  - 支持 7/8/9 位数据格式可配，支持奇偶校验功能可配
  - 支持 1/2 位停止位可配
  - 支持硬件自动产生发送奇偶校验位
  - 支持发送 FIFO 写入错误中断
- ◇ 支持 PWM 调制输出，且 PWM 占空比线性可调
- ◇ 支持 UART 输入输出通讯端口极性可配置
- ◇ UART 接收端口支持红外唤醒功能

### 5.2.2 结构框图

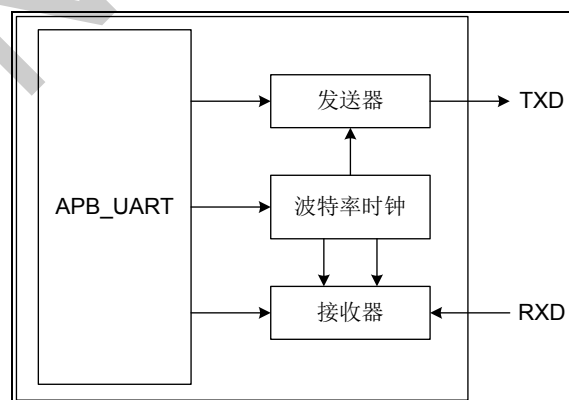


图 5-11 UART 电路结构图

### 5.2.3 UART 数据格式

UART 通讯每帧数据由 1 位起始位，7/8/9 位数据位、可配置奇偶校验位和停止位组成。UART\_CON0 寄存器中配置 TXMOD 和 RXMOD 选择发送和接收的数据格式，配置 TXFS

选择发送 1 位或 2 位停止位。接收数据时，只判断第 1 位停止位，若不为高电平则产生“帧错误”中断标志。在没有数据传输时，通讯端口处于高电平状态。

帧数据格式如下图所示：

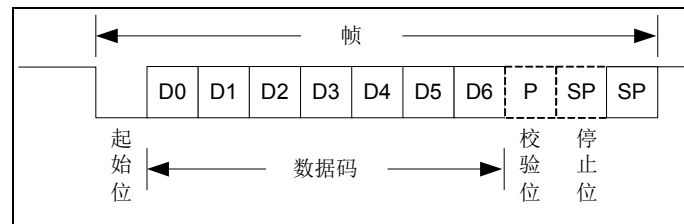


图 5-12 UART 7 位数据格式

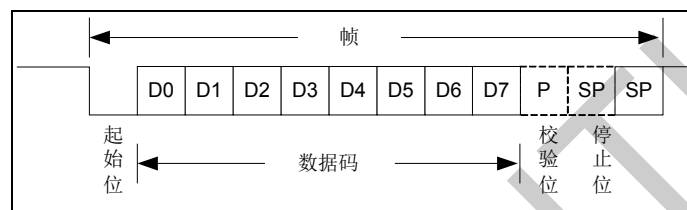


图 5-13 UART 8 位数据格式

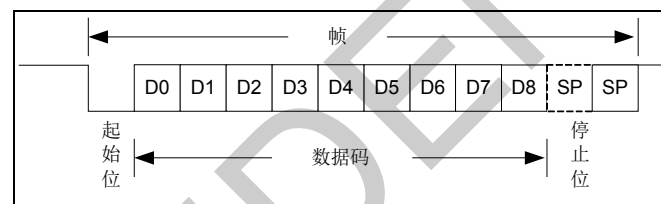


图 5-14 UART 9 位数据格式

数据发送和接收时，均是低位在前，高位在后，即先发送或接收数据的 LSB。通过发送数据寄存器 `UART_TBW` 写入发送的数据，通过接收数据寄存器 `UART_RBR` 读取接收的数据。

#### 5.2.4 UART 异步发送器

发送数据时，起始位 `START` 和停止位 `STOP` 由芯片硬件电路自动产生，用户只需要配置相应的 I/O 端口复用功能；配置 `UART_BRR` 和 `UART_CON0` 寄存器的 `BCS`，设定传输波特率；`UART_CON0` 寄存器中配置 `TXMOD`，选择发送的数据格式；配置 `TXFS`，选择发送的停止位数；配置 `TXEN`，使能数据发送；将要发送的数据写入发送数据寄存器 `UART_TBW`，就可以开始数据的异步发送。如果数据格式支持奇偶校验位，硬件电路会根据相应的数据位产生校验位，在数据位后自动发送校验位。

配置 `UART_CON0` 寄存器的 `TXP`，可选择发送端口极性。选择为正极性时，发送端口的数据与被发送数据一致；选择为负极性时，发送端口的数据与被发送数据相反，即被发送数据位为 1 时，发送端口的数据位为 0。

支持 1 级发送缓冲器 `TB0` 和 1 级发送移位寄存器，可进行数据的连续发送，直到发送缓冲器和移位寄存器全空，最多可连续写入和发送 2 帧数据，配置 `UART_CON0` 寄存器的 `TXFS` 可选择相邻两帧数据发送的时间间隔。发送缓冲器 `TB0` 为只读寄存器，只能通过发送数据寄存器 `UART_TBW` 写入。

发送数据寄存器 `UART_TBW` 为一个虚拟地址单元，物理上不存在实际的寄存器电路，写该寄存器地址单元时，实际上是将发送数据写入到发送缓冲器 `TB0` 中，再传输到发送移位寄存器，通过发送端口 `TX0` 进行数据发送。

发送数据寄存器 `UART_TBW` 支持 3 种写入方式：字节写入，半字写入和字写入。

当发送数据为 7 位和 8 位数据格式时：只能以字节方式写入 `UART_TBW`，发送数据被写入到发送缓冲器 `TB0`。

当发送数据为 9 位数据格式时：只能以半字或字方式写入 `UART_TBW`，发送数据均被写入到发送缓冲器 `TB0`。

发送数据从写入到发送到端口的数据流示意图如下所示：



图 5-15 UART0 发送数据流示意图

当发送缓冲器 `TB0` 被写入数据时，硬件自动置起 `UART_TB0` 寄存器的空满标志 `TBFF0` =1，当发送缓冲器 `TB0` 的数据移到发送移位寄存器后，硬件自动清除空满标志 `TBFF0`。当缓冲器 `TB0` 的发送空满标志 `TBFF0` 为 1 时，表示发送缓冲器和发送移位寄存器均满，此时如果仍继续写入 `UART_TBW`，则会置起写发送缓冲溢出中断标志 `TBWOIF`，同时新写入的数据无效，缓冲器数据仍保持。当字节写操作地址非最低字节，或半字写操作地址非低半字时，则会置起写发送缓冲错误中断标志 `TBWEIF`，同时新写入的数据无效，缓冲器数据仍保持。

当发送端口正在发送数据时，会置起发送状态位 `TXBUSY`=1，当发送缓冲器和发送移位寄存器均空时，会置起 `UART_IF` 寄存器的发送完成中断标志 `TCIF`，并清除发送状态位 `TXBUSY`=0，表示当前的数据发送进程结束。

支持发送缓冲器空中断，配置 `UART_CON1` 寄存器中 `TBIM`，可选择中断模式。

`TBIM<1:0>= 11`，为全空产生中断，即发送缓冲器 `TB0` 为空时，会置起 `UART_IF` 寄存器的发送缓冲器空中断标志 `TBIF`。

配置 `UART_CON0` 寄存器中 `TBCLR`，可将写入发送缓冲器数据清空，同时清除 `UART_TB0` 寄存器的各发送缓冲器 `TB0` 的空满标志 `TBFF0` =0。被清空的发送缓冲器中的数据将不会被发送，但移位寄存器中的数据仍将会被发送。

`UART_CON0` 寄存器中配置 `TRST`，可将异步发送器软件复位，复位后：禁止数据发送 `TXEN`=0；禁止 `UART_IF` 寄存器中发送相关中断 `TBIE`=0，`TBWEIE`=0；复位 `UART_IF` 寄存器的相关中断标志为默认值 `TBIF`=1，`TBWEIF`=0；清除发送忙标志 `TXBUSY`=0；清除发送缓冲器空满标志 `TBFF0` =0。

在发送数据时，可以先写发送数据寄存器 `UART_TBW`，再使能发送 `TXEN`，开始数据发送；也可以先使能发送 `TXEN`，再写发送数据寄存器 `UART_TBW`，开始数据发送。`UART` 发送缓冲器共有 1 级（即 1 个字节），发送缓冲器空中断标志位 `TBIF` 初始值为 1，该中断标志位可通过写缓冲器 `UART_TBW` 进行清除，当发送 `TB0` 缓冲器数据的起始位时（`TB0` 发送缓冲器的数据传输到发送移位寄存器，变为空），`TBIF` 位会再次被置起。

发送数据的操作流程图示例如下（以先使能 `TXEN`，再写 `UART_TBW` 为例）：

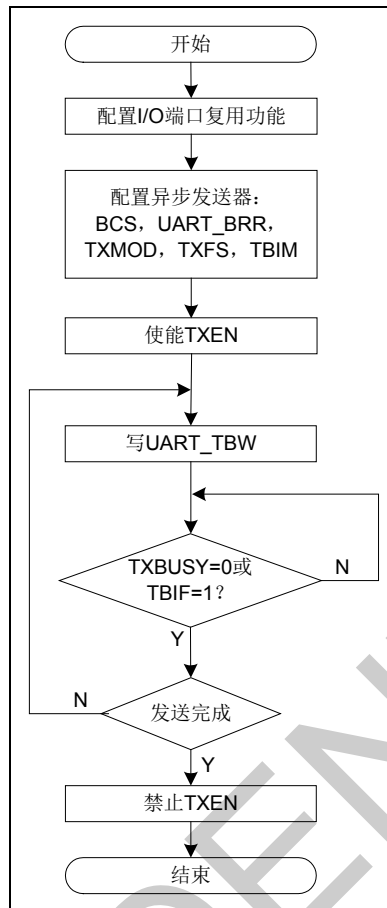


图 5-16 UART0 发送数据操作流程图示例

注：通过对 TBIF 标志的查询或中断方式，可实现数据连续无延迟发送；通过对 TXBUSY 标志的查询也可实现数据的连续发送，但需要在 TXBUSY=0 后等待至少 2 个波特率时钟周期，才能向寄存器 UART\_TBW 写入新的数据，否则数据写入操作无效，新数据写入后，再经过约 2 个波特率时钟周期，开始新一帧数据的发送。

### 5.2.5 UART异步接收器

接收数据时，配置相应的 I/O 端口复用功能；配置 UART\_BRR 寄存器和 UART\_CON1 寄存器的 BCS，设定传输波特率；配置 RXMOD，选择接收的数据格式；配置 RXEN，使能数据接收，就可以开始数据的异步接收。如果数据格式支持奇偶校验位，硬件电路会自动判断奇偶校验位是否正确，若不正确则会置起 UART\_IF 寄存器的奇偶校验错误中断标志 PEIF。如果接收到的第 1 位停止位不为高电平，则会置起 UART\_IF 寄存器的帧错误中断标志 FEIF。

UART\_CON0 寄存器中配置 RXP，可选择接收端口极性。选择为正极性时，接收端口的数据即作为接收数据；选择为负极性时，接收端口的数据取反后，作为接收数据，即接收端口的数据位为 1 时，接收的数据位为 0。

支持 1 级接收缓冲器 RB0 和 1 级接收移位寄存器，可进行数据的连续接收，直到接收缓冲器和移位寄存器全满，最多可连续接收 2 帧数据，再执行数据读取操作。读取接收数据寄存器 UART\_RBR，可得到接收的数据，对应的接收缓冲器清除 UART\_RB0 寄存器的满标志 RBFF0；也可以读取接收缓冲器 RB0 得到接收的数据，但不会清除满标志 RBFF0。

接收数据寄存器 `UART_RBR` 为一个虚拟地址单元，物理上不存在实际的寄存器电路，读该寄存器地址单元时，实际上是读取接收缓冲器 `RB0` 中的数据。

接收数据寄存器 `UART_RBR` 支持 3 种读取方式：字节读取，半字读取和字读取。

当接收数据为 7 位和 8 位数据格式时：只能以字节方式读取 `UART_RBR`，实际是读取接收缓冲器 `RB0` 的数据。

当接收数据为 9 位数据格式时：只能以半字或字方式读取 `UART_RBR`，均是读取接收缓冲器 `RB0` 的数据。

接收数据从接收端口到各级缓冲器的数据流示意图如下所示：

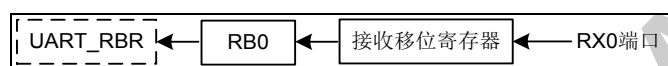


图 5-17 UART0 接收数据流示意图

接收缓冲器 `RB0` 的数据被读取后，会清除其接收满标志 `RBFF0`。

当接收缓冲器和接收移位寄存器均满时，如果再次接收到数据起始位，会置起接收数据溢出中断标志 `ROIF`，同时不会接收新数据，缓冲器数据仍保持。

当接收缓冲器和接收移位寄存器均空时，会清除接收忙标志 `RxBUSY`，表示当前无数据接收。

支持接收缓冲器满中断，配置 `UART_CON1` 寄存器的 `RBIM`，可选择中断模式。

`RBIM<1:0>=00`，为字节满产生中断，即接收缓冲器 `RB0` 为满时，会置起 `UART_IF` 寄存器的中断标志 `RBIF`；

配置 `UART_CON0` 寄存器 `RBCLR`，可将接收缓冲器接收到的数据清空，同时清除接收缓冲器满标志 `RBFF0=0`，已经在接收过程中的数据将不受影响。

配置 `UART_CON0` 寄存器 `RRST`，可将异步接收器软件复位，复位后：禁止数据接收 `RXEN=0`；`UART_IE` 寄存器中禁止接收相关中断 `RBIE=0`，`ROIE=0`，`FEIE=0`，`PEIE=0`；`UART_IF` 寄存器中复位相关中断标志为默认值 `RBIF=0`，`ROIF=0`，`FEIF=0`，`PEIF=0`；清除接收忙标志 `RxBUSY=0`；清除接收缓冲器满标志 `RBFF0=0`；清除接收缓冲器错误标志 `FE0=0`，`PE0=0`。

接收数据的操作流程图示例如下：

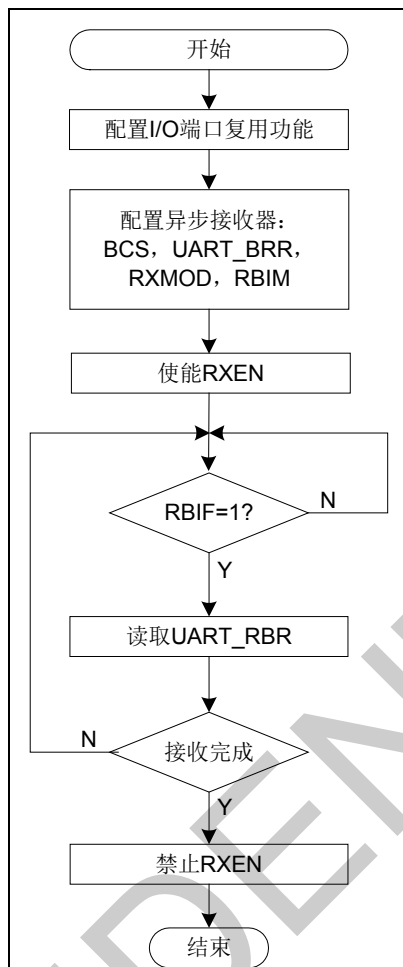


图 5-18 UART0 接收数据操作流程图示例

### 5.2.6 UART发送调制功能

发送调制模式是将 UART 传输到发送端口的信号电平，由 T16N 产生的 PWM 信号源或 BUZ 信号进行调制后，再从发送端口 TX0 输出。配置 TX0PS，可配置 UART 调制模式。配置 GPIO\_TXPWM 寄存器的 TX0PLV，选择发送端口 TX0 被调制的信号电平。配置 TX0\_S0、TX0\_S1、TX0\_S2 和 TX0\_S3，使能调制信号输出到端口。

UART 调制功能使用的 PWM 信号源，由 T16N 或 BUZ 提供。配置 TX0PS，可选择与 TX0 输出端口进行调制的 PWM 信号源。

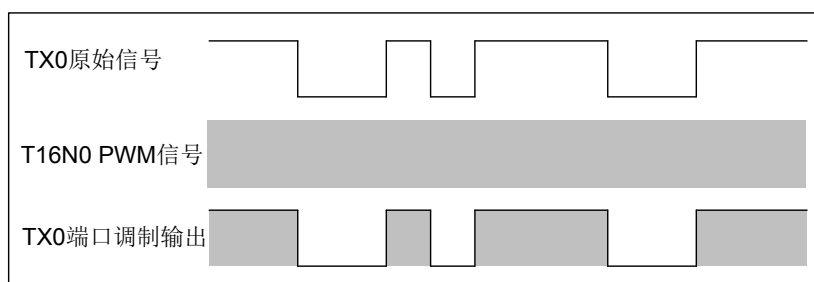


图 5-19 TX0 高电平调制输出波形图



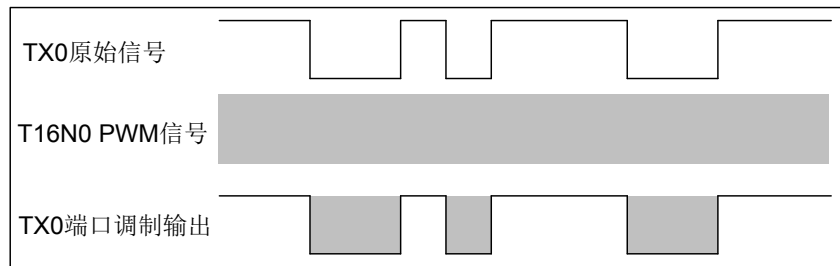


图 5-20 TX0 低电平调制输出波形图

### 5.2.7 UART红外唤醒功能

芯片支持 UART 接收端口的红外唤醒功能，需要通过软件控制实现。使能 UART 接收复用端口 RX0 的外部端口中断 PINT 功能，通过端口的 PINT 中断实现芯片的红外唤醒功能。对中断和唤醒的具体控制操作，可参考外部端口中断和浅睡眠模式，唤醒模式的相关章节描述。

### 5.2.8 UART端口极性

配置 UART\_CON0 寄存器 TXP 和 RXP，可分别选择发送端口 TX0 和接收端口 RX0 的正负极性。选择为正极性时，UART 端口数据与传输的数据一致；选择为负极性时，UART 端口数据与传输的数据相反。

### 5.2.9 UART自动波特率检测

接收数据时，配置 UART\_CON0 寄存器的 BDEN 可使能自动波特率检测功能，配置 UART\_CON1 寄存器的 BDM 可选择自动波特率检测的模式。UART\_CON1 寄存器中

BDM<1:0>=00，选择模式 1，接收数据流需以二进制 1 开始（即接收数据的最低位为 1B），UART 检测起始位波特率；

BDM<1:0>=01，选择模式 2，接收数据流需以二进制 10 开始（即接收数据的最低两位为 01B），UART 检测起始位和数据第一位波特率；

BDM<1:0>=10，选择模式 3，接收数据流需以二进制 1111\_1110 开始（即接收第一帧数据为 7FH），UART 检测起始位及前 7 个数据位波特率；

BDM<1:0>=11，选择模式 4，接收数据流需以二进制 1010\_1010 开始（即接收第一帧数据为 55H），UART 检测起始位及前 7 个数据位波特率。

检测完成后，硬件自动清除 BDEN 关闭自动波特率检测功能。若检测成功将更新波特率寄存器 UART\_BRR，并将这一帧数据写入接收缓冲器中，用户可从接收缓冲器中读取该数据；若检测超时，将置起 UART\_IF 寄存器的波特率检测错误中断标志 BDEIF=1，但波特率寄存器将不会被更新，接收缓冲器不会写入新的数据。

若接收的数据格式为 7 位数据或者 7 位数据之后的校验位不固定为 0 时，模式 3 和模式 4 将不适用，必须选择模式 1 和模式 2。其它接收的数据格式只要满足各模式起始字符，即可任意配置。

自动波特率检测不受初始配置的波特率值影响，但需要根据应用条件，配置合适的波特率时钟预分频选择位 BCS。

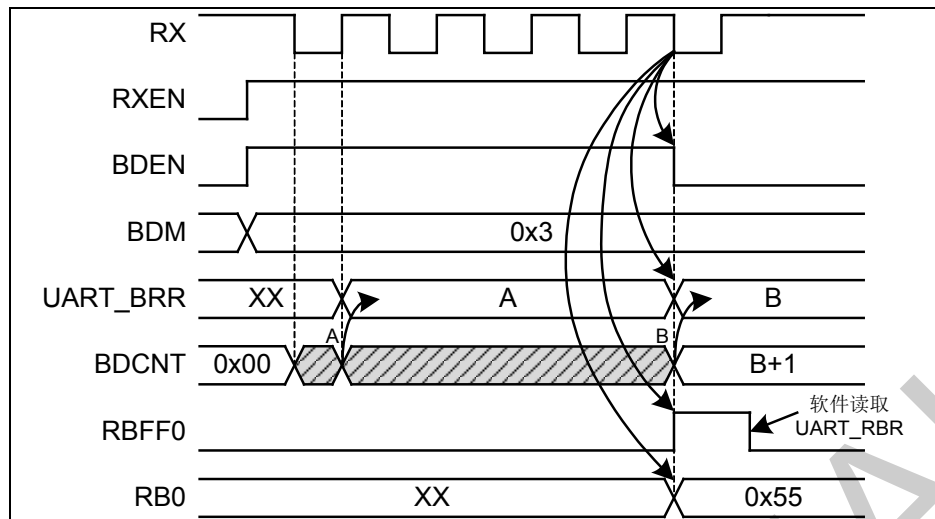


图 5-21 自动波特率检测时序示意图

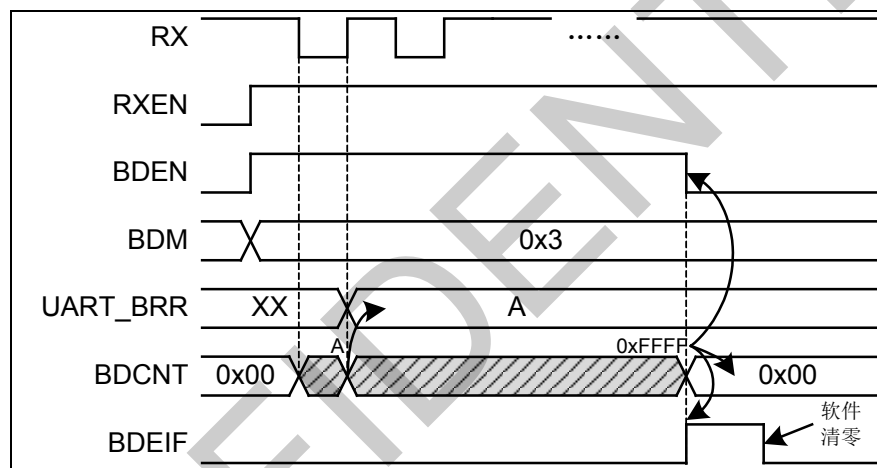


图 5-22 自动波特率检测错误时序示意图

### 5.2.10 UART空闲帧检测

接收数据时，配置 UART\_CON0 寄存器的 IDEN 可使能空闲帧检测功能，配置 UART\_CON1 寄存器的 IDM 可选择空闲帧检测的模式。

IDM<1:0>=00，将检测连续 10 个比特位空闲；

IDM<1:0>=01，将检测连续 11 个比特位空闲；

IDM<1:0>=10，将检测连续 12 个比特位空闲；

IDM<1:0>=11，将检测连续 13 个比特位空闲。

当检测到空闲帧后，将置起 UART\_IF 寄存器的接收空闲帧中断标志位 IDIF=1。硬件在接收到数据之后才会触发空闲帧检测，使能空闲帧检测功能后若接收数据线始终为空闲，则不会影响接收空闲帧中断标志位。

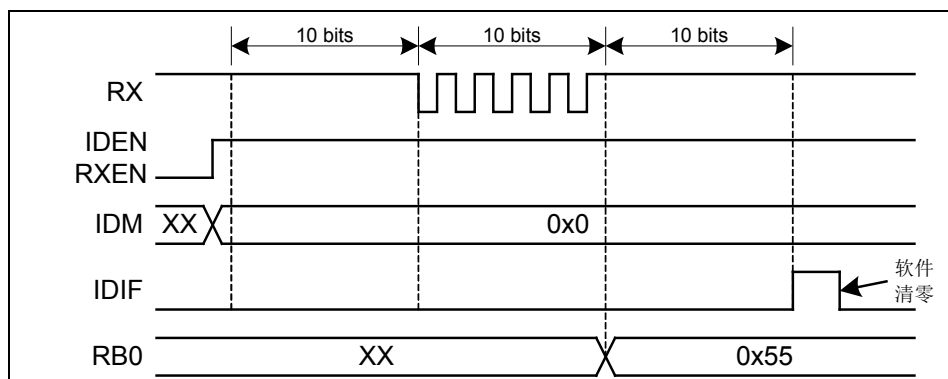


图 5-23 空闲帧检测时序示意图

### 5.2.11 UART发送暂停和接收暂停

配置 UART\_CON0 寄存器的 TXI 可暂停发送数据，TX0 端口将相应为空闲电平，发送缓冲器中的数据将保持不变。若此时正在发送数据，则待当前帧数据发送完成后才会被暂停。

配置 UART\_CON0 寄存器的 RXI 可暂停接收数据，接收缓冲器中的数据将保持不变。若此时正在接收数据，则该一帧数据将被丢弃，不会写入接收缓冲器中。接收被暂停后，不会影响接收数据的校验位判断和接收空闲帧检测。

### 5.2.12 特殊功能寄存器

#### 5.2.12.1 UART控制寄存器 0 (UART\_CON0)

UART 控制寄存器 0 (UART\_CON0)

偏移地址: 00H

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	RXP	RXMOD<3:0>			保留	IDEN	BDEN	RXI	RBCLR	RRST	RXEN				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TXFS	TXP	TXMOD<3:0>			保留			TXI	TBCLR	TRST	TXEN			

—	bit31-29	—	—
RXP	bit28	R/W	接收端口极性选择位 0: 正极性 (标准 UART 极性) 1: 负极性 (反向 UART 极性)
RXMOD<3:0>	bit27-24	R/W	接收数据格式选择位 0000: 8 位数据 0010: 9 位数据 0100: 7 位数据 1000: 8 位数据+偶校验位 1001: 8 位数据+奇校验位 1010: 8 位数据+固定 0 1011: 8 位数据+固定 1 1100: 7 位数据+偶校验位 1101: 7 位数据+奇校验位

			1110: 7 位数据+固定 0 1111: 7 位数据+固定 1 其他: 未用
—	bit23-22	—	—
IDEN	bit21	R/W	空闲帧检测使能位 0: 禁止 1: 使能
BDEN	bit20	R/W	自动波特率检测使能位 0: 禁止 1: 使能
RXI	bit19	R/W	接收暂停功能使能位 0: 禁止 1: 使能
RBCLR	bit18	W	接收缓冲器清除 0: 读取时始终为 0 1: 清除接收缓冲器
RRST	bit17	W	接收器软件复位 0: 读取时始终为 0 1: 软件复位
RXEN	bit16	R/W	接收使能位 0: 禁止 1: 使能
—	bit15-14	—	—
TXFS	bit13	R/W	发送帧停止位选择位 0: 1 位停止位 1: 2 位停止位
TXP	bit12	R/W	发送端口极性选择位 0: 正极性 (标准 UART 极性) 1: 负极性 (反向 UART 极性)
TXMOD<3:0>	bit11-8	R/W	发送数据格式选择位 0000: 8 位数据 0010: 9 位数据 0100: 7 位数据 1000: 8 位数据+偶校验位 1001: 8 位数据+奇校验位 1010: 8 位数据+固定 0 1011: 8 位数据+固定 1 1100: 7 位数据+偶校验位 1101: 7 位数据+奇校验位 1110: 7 位数据+固定 0 1111: 7 位数据+固定 1 其他: 未用
—	bit7-4	—	—

TXI	bit3	R/W	发送暂停功能使能位 0: 禁止 1: 使能
TBCLR	bit2	W	发送缓冲器清除 0: 读取时始终为 0 1: 清除发送缓冲器
TRST	bit1	W	发送器软件复位 0: 读取时始终为 0 1: 软件复位
TXEN	bit0	R/W	发送使能位 0: 禁止 1: 使能

### 5.2.12.2 UART控制寄存器 1 (UART\_CON1)

UART 控制寄存器 1 (UART\_CON1)

偏移地址: 04<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														IDM<1:0>	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		BDM<1:0>		保留	BCS<2:0>			保留		RBIM<1:0>		保留		TBIM<1:0>	

—	bit31-18	—	—
IDM<1:0>	bit17-16	R/W	空闲帧检测模式选择位 00: 连续 10 个比特空闲 01: 连续 11 个比特空闲 10: 连续 12 个比特空闲 11: 连续 13 个比特空闲
—	bit15-14	—	—
BDM<1:0>	bit13-12	R/W	自动波特率模式选择位 00: 模式 1 01: 模式 2 10: 模式 3 11: 模式 4
—	bit11	—	—
BCS<2:0>	bit10-8	R/W	波特率时钟预分频选择位 000: 波特率时钟关闭 001: PCLK 010: PCLK/2 011: PCLK/4 1xx: PCLK/8
—	bit7-6	—	—

RBIM<1:0>	bit5-4	R/W	接收缓冲满中断模式选择位 00: 字节满产生中断（缓冲器数据多于 1 个字节） 其他: 保留
—	bit3-2	—	—
TBIM<1:0>	bit1-0	R/W	发送缓冲空中断模式选择位 11: 全空产生中断（缓冲器全部为空） 其他: 保留

注 1: UART\_CON1 寄存器中 BDM 模式操作方式如下:

模式 1: 数据流以二进制 1 开始（即接收数据的最低位为 1B），检测起始位波特率;

模式 2: 数据流以二进制 10 开始（即接收数据的最低两位为 01B），检测起始位及数据第一位波特率;

模式 3: 数据流以二进制 1111\_1110 开始（即接收数据 7FH），检测起始位及前 7 个数据位;

模式 4: 数据流以二进制 1010\_1010 开始（即接收数据 55H），检测起始位及前 7 个数据位。

注 2: 若接收的数据的格式为 7 位数据或者 7 位数据之后的校验位不固定为 0 时，模式 3 和模式 4 将不适用，必须选择模式 1 和模式 2。其他接收数据格式只要满足各模式起始字符，可任意配置。

### 5.2.12.3 UART 波特率寄存器 (UART\_BRR)

#### UART 波特率寄存器 (UART\_BRR)

偏移地址: 10<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-16	—	—
BRINT<11:0>	bit15-4	R/W	传输波特率整数位
BRFRA<3:0>	bit3-0	R/W	传输波特率小数位

注 1: 当 UART\_BRR 表示为 16 位无符号数，其中 4 位小数位，12 位整数位，其数值表示的是 UART 波特率分频数。例如：0x0104（十进制为 260）则表示为分频数  $BRRDIV=260/16=16.25$ ；0x0156（十进制为 342）则表示为分频数  $BRRDIV=342/16=21.375$ 。

注 2: 当 UART\_BRR 表示的波特率分频数小于 1.0 时，则硬件固定为  $BRRDIV=1.0$ 。

注 3: UART 传输波特率计算公式如下:

$$BAUD = \frac{F_{pclk}}{16 \times n \times BRRDIV}$$

其中  $F_{pclk}$  为系统时钟频率， $n$  为波特率时钟预分频数，由 UART\_CON1 寄存器中 BCS<2:0>决定

BCS<2:0>=001 时:  $n = 1$ ;

BCS<2:0>=010 时:  $n = 2$ ;

BCS<2:0>=011 时:  $n = 4$ ;

BCS<2:0>=1xx 时:  $n = 8$ 。

## 5.2.12.4 UART状态寄存器 (UART\_STA)

## UART 状态寄存器 (UART\_STA)

偏移地址: 14<sub>H</sub>复位值: 00000000\_00000000\_00000001\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								PER3	FER3	PER2	FER2	PER1	FER1	PER0	FER0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RXBUSY		RBOV	保留						TXBUSY		TBOV	保留		

—	bit31-24	—	—
PER3	bit23	R	当前读取的 <b>BYTE3</b> 校验错误位 0: 正确 1: 错误
FER3	bit22	R	当前读取的 <b>BYTE3</b> 帧格式错误位 0: 正确 1: 错误
PER2	bit21	R	当前读取的 <b>BYTE2</b> 校验错误位 0: 正确 1: 错误
FER2	bit20	R	当前读取的 <b>BYTE2</b> 帧格式错误位 0: 正确 1: 错误
PER1	bit19	R	当前读取的 <b>BYTE1</b> 校验错误位 0: 正确 1: 错误
FER1	bit18	R	当前读取的 <b>BYTE1</b> 帧格式错误位 0: 正确 1: 错误
PER0	bit17	R	当前读取的 <b>BYTE0</b> 校验错误位 0: 正确 1: 错误
FER0	bit16	R	当前读取的 <b>BYTE0</b> 帧格式错误位 0: 正确 1: 错误
—	bit15-14	—	—
RXBUSY	bit13	R	接收状态位 0: 接收空闲 1: 正在接收
RBOV	bit12	R	接收缓冲器溢出状态位 0: 未溢出 1: 溢出
—	bit11-6	—	—

TXBUSY	bit5	R	发送状态位 0: 发送空闲 1: 正在发送
TBOV	bit4	R	发送缓冲器溢出状态位 0: 未溢出 1: 溢出
—	bit3-0	—	—

### 5.2.12.5 UART中断使能寄存器 (UART\_IE)

#### UART 中断使能寄存器 (UART\_IE)

偏移地址: 18<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	RBROIE	RBREIE	BDEIE	PEIE	FEIE	ROIE	保留	保留	保留	保留	保留	保留	保留	IDIE	RBIE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	TBWOIE	TBWEIE	保留	保留	保留	保留	保留	保留	TCIE	TBIE

—	bit31-30	—	—
RBROIE	bit29	R/W	读接收缓冲溢出中断使能位 0: 禁止 1: 使能
RBREIE	bit28	R/W	读接收缓冲错误中断使能位 0: 禁止 1: 使能
BDEIE	bit27	R/W	波特率检测错误中断使能位 0: 禁止 1: 使能
PEIE	bit26	R/W	接收校验错误中断使能位 0: 禁止 1: 使能
FEIE	bit25	R/W	接收帧错误中断使能位 0: 禁止 1: 使能
ROIE	bit24	R/W	接收数据溢出中断使能位 0: 禁止 1: 使能
—	bit23-18	—	—
IDIE	bit17	R/W	接收空闲帧中断使能位 0: 禁止 1: 使能
RBIE	bit16	R/W	接收缓冲器满中断使能位 0: 禁止



			1: 使能
—	bit15-10	—	—
TBWOIE	bit9	R/W	写发送缓冲溢出中断使能位 0: 禁止 1: 使能
TBWEIE	bit8	R/W	写发送缓冲错误中断使能位 0: 禁止 1: 使能
—	bit7-2	—	—
TCIE	bit1	R/W	发送完成中断使能位 0: 禁止 1: 使能
TBIE	bit0	R/W	发送缓冲器空中断使能位 0: 禁止 1: 使能

### 5.2.12.6 UART中断标志寄存器 (UART\_IF)

#### UART 中断标志寄存器 (UART\_IF)

偏移地址: 1C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000001<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	RBROIF	RBREIF	BDEIF	PEIF	FEIF	ROIF	保留						IDIF	RBIF	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						TBWOIF	TBWEIF	保留						TCIF	TBIF

—	bit31-30	—	—
RBROIF	bit29	R/W	读接收缓冲溢出中断标志位 0: 读接收缓冲未溢出 1: 读接收缓冲溢出 软件写 1 清除标志位, 写 0 无效
RBREIF	bit28	R/W	读接收缓冲错误中断标志位 0: 读接收缓冲未错误 1: 读接收缓冲错误 软件写 1 清除标志位, 写 0 无效
BDEIF	bit27	R/W	波特率检测错误中断标志位 0: 波特率检测未错误 1: 波特率检测错误 软件写 1 清除标志位, 写 0 无效
PEIF	bit26	R/W	接收校验错误中断标志位 0: 接收校验正常 1: 接收校验错误 软件写 1 清除标志位, 写 0 无效

FEIF	bit25	R/W	接收帧错误中断标志位 0: 接收帧正常 1: 接收帧错误 软件写 1 清除标志位, 写 0 无效
ROIF	bit24	R/W	接收数据溢出中断标志位 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
—	bit23-18	—	—
IDIF	bit17	R/W	接收空闲帧中断标志位 0: 未接收到空闲帧 1: 接收到空闲帧 软件写 1 清除标志位, 写 0 无效
RBIF	bit16	R	接收缓冲器满中断标志位 0: 非满 1: 满 读 UART_RBR 寄存器可清除标志位
—	bit15-10	—	—
TBWOIF	bit9	R/W	写发送缓冲溢出中断标志位 0: 写发送缓冲未溢出 1: 写发送缓冲溢出 软件写 1 清除标志位, 写 0 无效
TBWEIF	bit8	R/W	写发送缓冲错误中断标志位 0: 写发送缓冲未错误 1: 写发送缓冲错误 软件写 1 清除标志位, 写 0 无效
—	bit7-2	—	—
TCIF	bit1	R/W	发送完成中断标志位 0: 发送未完成 1: 发送已完成 软件写 1 清除标志位, 写 0 无效
TBIF	bit0	R	发送缓冲器空中断标志位 0: 非空 1: 空 写 UART_TBW 寄存器可清除标志位

注 1: UART 中断禁止时, 如果满足条件仍会置起对应的中断标志位, 只是不会产生中断请求。

注 2: 对 UART\_IF 寄存器中的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

## 5.2.12.7 UART发送数据写入寄存器 (UART\_TBW)

## UART 发送数据写入寄存器 (UART\_TBW)

偏移地址: 20<sub>H</sub>复位值: XXXXXXXX\_XXXXXXX\_XXXXXXX\_XXXXXXX<sub>B</sub>

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

TBW<31:0>	bit31-0	W	写入的发送数据 字节写入时: 仅允许对 UART_TBW<7:0>写入 半字写入时: 仅允许对 UART_TBW<15:0>写入 字写入时: 对 UART_TBW<31:0>写入
-----------	---------	---	----------------------------------------------------------------------------------------------------

## 5.2.12.8 UART接收数据读取寄存器 (UART\_RBR)

## UART 接收数据读取寄存器 (UART\_RBR)

偏移地址: 24<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

RBR<31:0>	bit31-0	R	读取的接收数据 字节读取时: 仅允许对 UART_RBR<7:0>读取 半字读取时: 仅允许对 UART_RBR<15:0>读取 字读取时: 对 UART_RBR<31:0>读取
-----------	---------	---	----------------------------------------------------------------------------------------------------

## 5.2.12.9 UART发送缓冲寄存器 0 (UART\_TB0)

## UART 发送缓冲寄存器 0 (UART\_TB0)

偏移地址: 40<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TBFF0	TP0	保留	TB0<8:0>											

—	bit31-14	—	—
TBFF0	bit13	R	发送缓冲器 0 空满标志位 0: 空

			1: 满
TP0	bit12	R	发送的奇偶校验位 发送缓冲器 0 对应的奇偶校验位
—	bit11-9	—	—
TB0<8:0>	bit8-0	R	发送缓冲器 0 数据

### 5.2.12.10 UART接收缓冲寄存器 0 (UART\_RB0)

#### UART 接收缓冲寄存器 0 (UART\_RB0)

偏移地址: 60<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PE0	FE0	RBFF0	RP0	保留				RB0<8:0>							

—	bit31-16	—	—
PE0	bit15	R	接收缓冲器 0 数据校验错误标志位 0: 正确 1: 错误
FE0	bit14	R	接收缓冲器 0 数据帧错误标志位 0: 正确 1: 错误
RBFF0	bit13	R	接收缓冲器 0 空满标志位 0: 空 1: 满
RP0	bit12	R	接收的奇偶校验位 接收缓冲器 0 对应的奇偶校验位
—	bit11-9	—	—
RB0<8:0>	bit8-0	R	接收缓冲器 0 数据

### 5.2.13 UART应用说明

芯片支持 3 个通用异步接收/发送器为 UART0, UART1, UART2。

UART 的发送脉宽调制模式, 使用 T16N 或 BUZ 产生的 PWM 脉冲进行调制, 然后通过 TX0/TX1/TX2 端口输出调制后的信号, 与 PWM 脉冲本身是否通过 T16N 输出端口 0 和 T16N 输出端口 1 或 BUZ 端口输出无关, 所以对 T16N 或 BUZ 的输出调制模式, 即使芯片不支持其 I/O 端口复用功能, 仍可以在正确配置后, 产生 PWM 脉冲分别对 TX0, TX1, TX2 输出端口的信号进行调制。

## 5.3 SPI同步串口通讯控制器（SPI1）

### 5.3.1 概述

- ◇ 支持主控模式、从动模式
- ◇ 支持 4 种数据传输格式
- ◇ 支持主控模式通讯时钟速率可配置
- ◇ 支持 1 到 8 位帧位宽选择
- ◇ 支持 4 级发送缓冲器和 4 级接收缓冲器
- ◇ 支持发送和接收缓冲器空/满中断
- ◇ 支持接收数据溢出中断、发送数据写错误中断、从动模式的发送数据错误中断
- ◇ 支持从动模式的片选变化中断、主控模式的空闲状态中断
- ◇ 支持主控模式延迟接收
- ◇ 支持主控模式发送间隔

### 5.3.2 结构框图

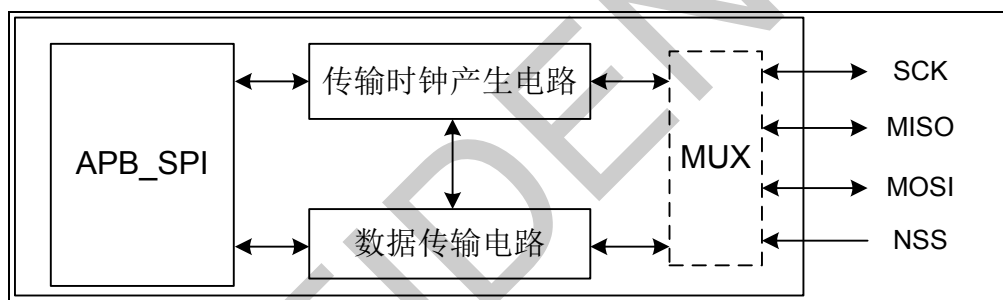


图 5-24 SPI 电路结构框图

### 5.3.3 SPI通讯模式

SPI 支持主控和从动两种通讯模式，配置 SPI\_CON 寄存器的 MS 位，可选择通讯模式。

通讯时钟端口为 SCK，从动模式下的片选信号端口为 NSS，数据输出端口 MISO，数据输入端口 MOSI，主控模式下可使用普通 I/O 端口作为片外从设备的片选信号端口，数据输出端口 MOSI，数据输入端口 MISO。具体见下表所示：

SPI 通讯端口	SPI 主控模式	SPI 从动模式
SCK	支持	支持
MOSI	支持	支持
MISO	支持	支持
NSS	—	支持

### 5.3.4 SPI数据格式

配置 SPI\_CON 寄存器的 DFS，可选择 SPI 通讯数据格式，数据发送和接收时，均是高位在前，低位在后。如果是发送数据在先，接收数据在后，输出端口 MOSI(或 MISO) 会在第一个 SCK 时钟边沿，输出数据的 MSB 位；反之输出端口 MOSI（或 MISO）在第一个

SCK 时钟边沿之前，输出数据的 MSB 位。

以下以 SPI 从动通讯模式为例，对数据通讯时序进行说明。

SPI\_CON 寄存器中 DFS<1:0> = 00，上升沿发送（先），下降沿接收（后）：

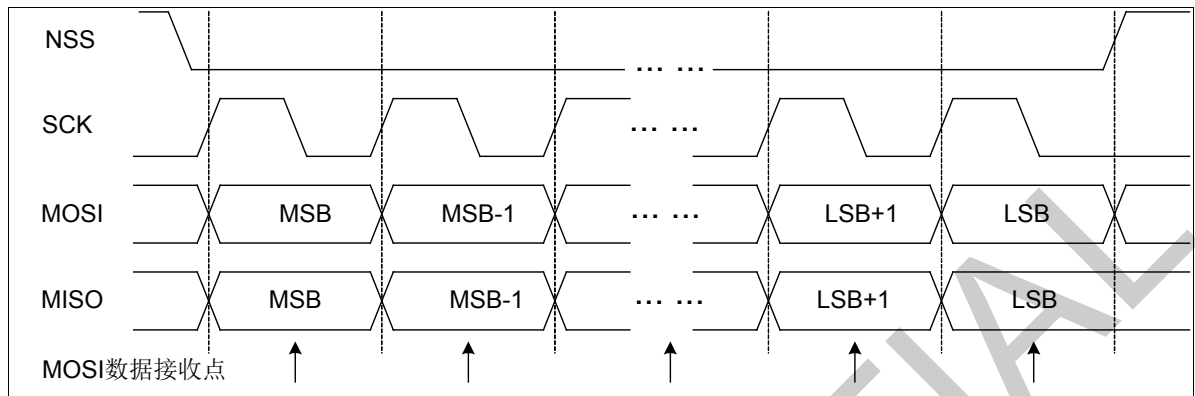


图 5-25 SPI 时钟上升沿发送，下降沿接收波形示意图

DFS<1:0> = 01，下降沿发送（先），上升沿接收（后）：

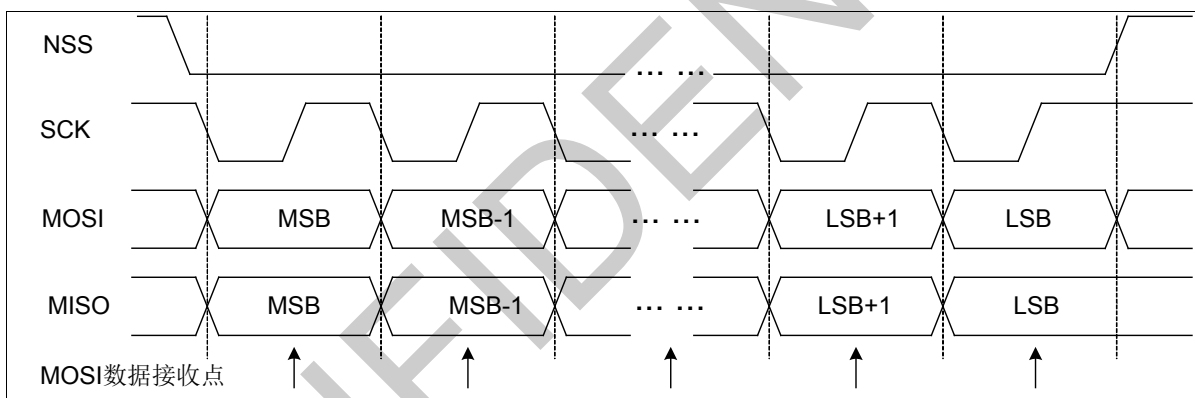


图 5-26 SPI 时钟下降沿发送，上升沿接收波形示意图

DFS<1:0> = 10，上升沿接收（先），下降沿发送（后）：

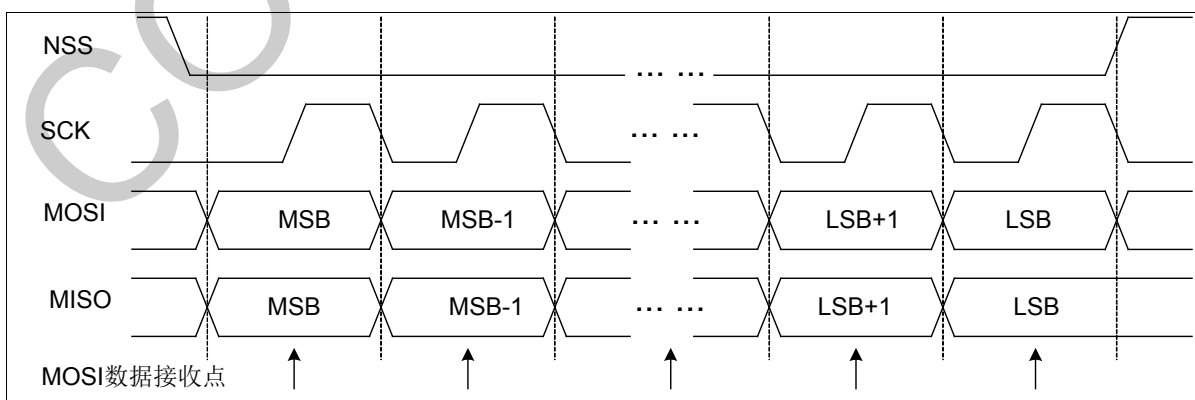


图 5-27 SPI 时钟上升沿接收，下降沿发送波形示意图

DFS<1:0> = 11，下降沿接收（先），上升沿发送（后）：

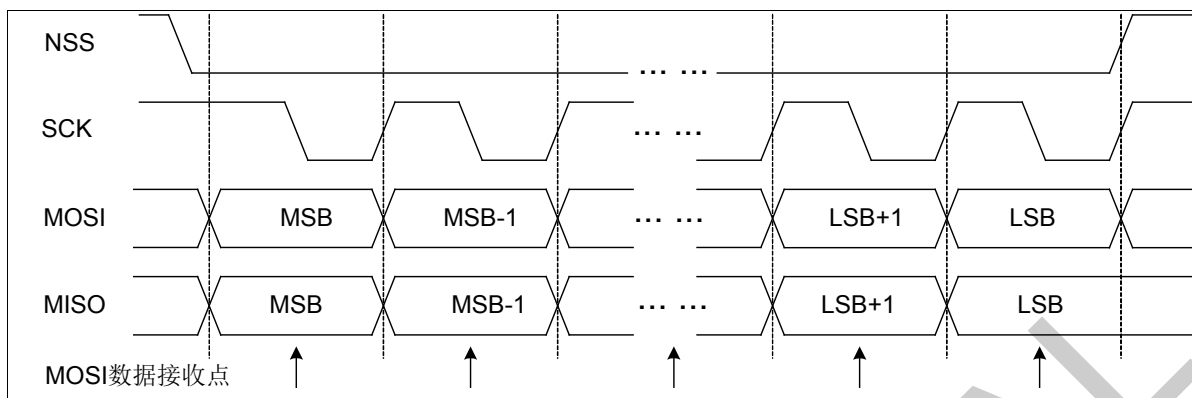


图 5-28 SPI 时钟下降沿接收，上升沿发送波形示意图

### 5.3.5 SPI 帧位宽

SPI 传输帧位宽可变，配置 SPI\_CON 寄存器的 DW<2:0>，可以设置传输帧位宽为 1~8 位。

SPI 模块支持同步发送器与同步接收器，容量均为 4 个字节，采用以下对齐方式：

SPI 帧位宽 1~8 时，发送/接收缓冲器采用字节对齐方式，每一级缓冲器中存储一帧数据，SPI 同步发送器或者同步接收器中最多可缓冲 4+1 帧数据；

以上对齐方式均采用低位对齐，以上“+1”帧数据为移位寄存器中存储的一帧数据，下文中将做详细介绍。

### 5.3.6 SPI 同步发送器

支持 4 级发送缓冲器 TB0, TB1, TB2, TB3 和 1 级发送移位寄存器，可进行数据的连续发送，直到发送缓冲器和移位寄存器全空，最多可连续写入和发送 5 帧数据。发送缓冲器 TB0~TB3 为只读寄存器，只能通过发送数据寄存器 SPI\_TBW 写入。

发送数据寄存器 SPI\_TBW 为一个虚拟地址单元，物理上不存在实际的寄存器电路，写该寄存器地址单元时，实际上是将发送数据写入到发送缓冲器 TB0~TB3 中，再传输到发送移位寄存器，通过发送数据端口 MOSI（或 MISO）进行数据发送。

发送数据寄存器 SPI\_TBW 支持 3 种写入方式：字节写入，半字写入和字写入。

字节方式写入 SPI\_TBW 时，发送数据被写入到发送缓冲器 TB0；半字方式写入 SPI\_TBW 时，发送数据被同时写入到发送缓冲器 TB0 和 TB1，其中低字节存放在 TB1 中；字方式写入 SPI\_TBW 时，发送数据被同时写入到发送缓冲器 TB0~TB3，其中低字节存放在 TB3 中。

发送数据从写入到发送到端口的数据流示意图如下所示（以主控模式为例）：

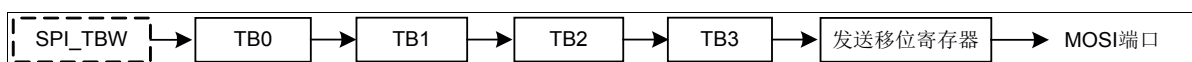


图 5-29 SPI 发送数据流示意图

支持发送缓冲器空中断，配置 SPI\_IE 寄存器的 TBIM，可选择中断模式。

TBIM<1:0>=00, 为字节空产生中断, 即发送缓冲器 TB0 为空时, SPI\_IF 寄存器会置起中断标志 TBIF;

TBIM<1:0>=01, 为半字空产生中断, 即发送缓冲器 TB0 和 TB1 均为空时, SPI\_IF 寄存器会置起中断标志 TBIF;

TBIM<1:0>=10, 为字空产生中断, 即发送缓冲器 TB0~TB3 均为空时, SPI\_IF 寄存器会置起中断标志 TBIF。

支持发送数据寄存器 SPI\_TBW 误写中断, 当对 SPI\_TBW 的写入方式, 与发送缓冲器 TB0~TB3 的空闲状态冲突时, 或写入操作访问错误时, SPI\_IF 寄存器会置起误写中断标志 TBWEIF。

### 5.3.7 SPI同步接收器

支持 4 级接收缓冲器 RB0, RB1, RB2, RB3 和 1 级接收移位寄存器, 可进行数据的连续接收, 直到接收缓冲器和移位寄存器全满, 最多可连续接收 5 帧数据, 再执行数据读取操作。读取接收数据寄存器 SPI\_RBR, 可得到接收的数据, SPI\_STA 寄存器中对应的接收缓冲器清除满标志 RBFF0~RBFF3; 也可以读取接收缓冲器 RB0~RB3 得到接收的数据, 但不会清除满标志 RBFF0~RBFF3。

接收数据寄存器 SPI\_RBR 为一个虚拟地址单元, 物理上不存在实际的寄存器电路, 读该寄存器地址单元时, 实际上是读取接收缓冲器 RB0~RB3 中的数据。

接收数据寄存器 SPI\_RBR 支持 3 种读取方式: 字节读取, 半字读取和字读取。

字节方式读取 SPI\_RBR 时, 实际是读取接收缓冲器 RB0 的数据; 半字方式读取 SPI\_RBR 时, 实际是同时读取接收缓冲器 RB0 和 RB1 的数据, 其中 RB0 中的数据为低字节; 字方式读取 SPI\_RBR 时, 实际是同时读取接收缓冲器 RB0~RB3, 其中 RB0 中的数据为低字节。

接收数据从接收端口到各级缓冲器的数据流示意图如下所示 (以主控模式为例):

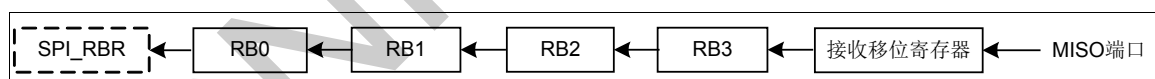


图 5-30 SPI 接收数据流示意图

同步接收器的接收顺序如下:

当同步接收器全空时, 接收移位寄存器的数据自动移入 RB0;

仅 RB1~RB3 空时, 接收移位寄存器的数据自动移入 RB1;

仅 RB2~RB3 空时, 接收移位寄存器的数据自动移入 RB2;

仅 RB3 空时, 接收移位寄存器的数据自动移入 RB3。

接收缓冲器 RB0~RB3 的数据移到下一级缓冲器后, 会清除其接收满标志 RBFF0~RBFF3。

当 4 级接收缓冲器和 1 级接收移位寄存器均满时, 如果再次接收到数据位, SPI\_IF 寄存器中会置起接收数据溢出中断标志 ROIF, 同时不会接收新数据, 缓冲器数据仍保持。

支持接收缓冲器满中断, 配置 SPI\_IE 寄存器的 RBIM, 可选择中断模式。



RBIM<1:0>=00，为字节满产生中断，即接收缓冲器 RB0 为满时，SPI\_IF 寄存器中会置起中断标志 RBIF；

RBIM<1:0>=01，为半字满产生中断，即接收缓冲器 RB0 和 RB1 均为满时，SPI\_IF 寄存器中会置起中断标志 RBIF；

RBIM<1:0>=10，为字满产生中断，即接收缓冲器 RB0~RB3 均为满时，SPI\_IF 寄存器中会置起中断标志 RBIF。

### 5.3.8 SPI通讯控制

配置 SPI 通讯模式，数据格式；对主控制模式，还需配置 SPI\_CKS，设定传输时钟速率，并选择是否使能接收延迟和发送间隔，对从动模式，传输时钟由主机方提供；SPI\_CON 寄存器中配置 EN 和 REN，使能数据发送和接收；将要发送的数据写入发送数据寄存器 SPI\_TBW，就可以开始数据的发送，读取接收数据寄存器 SPI\_RBR，可以获得接收到的数据。

SPI 主控模式下，发送缓冲器和发送移位寄存器中的数据发送完毕后，进入空闲状态，SPI\_STA 寄存器中会置起空闲标志 IDLE，并且 SPI\_IF 寄存器中产生空闲中断标志 IDIF。

SPI 从动模式下，如果发送缓冲器 TB0~TB3 和发送移位寄存器均为空时，又收到主机提供的通讯时钟，则 SPI\_IF 寄存器中会置起发送错误中断标志 TEIF。

SPI 从动模式，支持片选信号变化中断，配置 SPI\_IE 寄存器的 NSSIE，可使能该中断。

配置 SPI\_CON 寄存器的 RST 位，可将 SPI 通讯模块软件复位，复位后：禁止数据通讯 EN=0；SPI\_IE 寄存器中禁止相关中断 TBIE=0，TBWEIE=0，RBIE=0，TEIE=0，ROIE=0，IDIE=0，NSSIE=0；SPI\_IF 寄存器中复位相关中断标志为默认值 TBIF=1，TBWEIF=0，RBIF=0，TEIF=0，ROIF=0，IDIF=0，NSSIF=0；SPI\_STA 寄存器中置起空闲标志 IDLE=1；置起各发送缓冲器空标志 TBEF0~TBEF3=1；清除各接收缓冲器满标志 RBFF0~RBFF3=0。

### 5.3.9 SPI延迟接收功能

SPI 通讯时，是利用时钟的上升/下降沿分别对数据的发送和接收进行同步。正常通讯时，对主机接收数据来说，从机送出的数据应在半个时钟周期内，到达主机接收端口，否则会造成主机接收数据的丢失。

SPI 主控模式支持延迟接收功能，配置 SPI\_CON 寄存器的 DRE，可使能该功能，主机可以再延迟半个时钟周期，在下一个发送时钟边沿处，进行数据的接收采集。所以延迟接收功能使能后，从机发送端口和主机接收端口之间的线路延时，最大可接近 1 个通讯时钟周期。

举例说明 SPI 延迟接收功能：SPI\_CON 寄存器中 DFS<1:0>=00，上升沿发送（先），下降沿接收（后）。

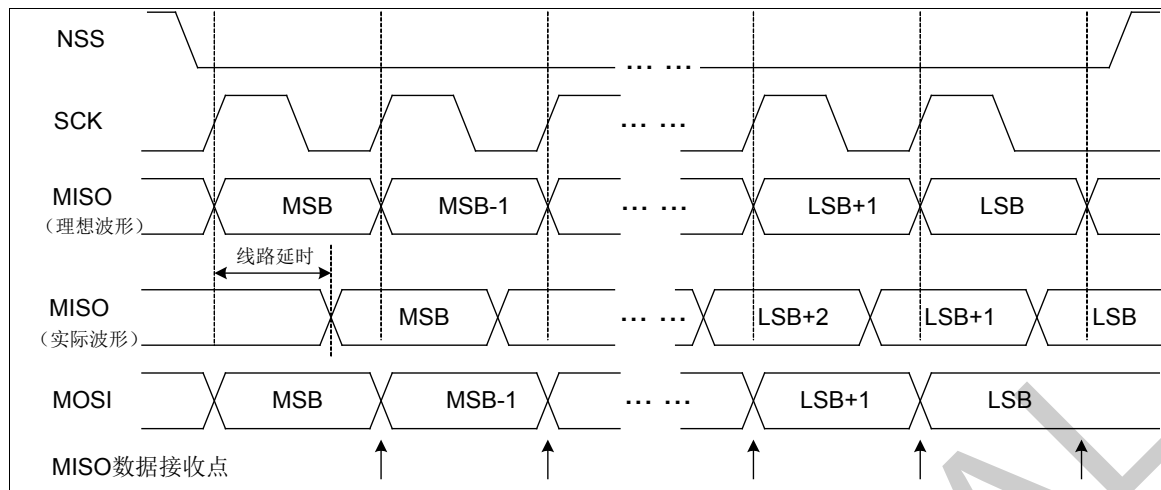


图 5-31 SPI 延迟接收功能波形示意图

### 5.3.10 SPI数据帧发送间隔功能

SPI 主控模式支持数据帧发送间隔功能，配置 SPI\_CON 寄存器的 TME，可启用该功能，配置 TMP，可设定发送间隔周期。当使能 SPI 数据帧发送间隔功能时，每帧数据发送完成后，会等待预先设定的发送间隔时间，再发送下一帧数据。

### 5.3.11 特殊功能寄存器

#### 5.3.11.1 SPI控制寄存器（SPI\_CON）

SPI 控制寄存器（SPI\_CON）

偏移地址：00<sub>H</sub>

复位值：00000111\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXCLR	TXCLR	保留			DW<2:0>			TMP<5:0>					TMS	TME	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								DFS<1:0>		DRE	保留	REN	MS	RST	EN

RXCLR	bit31	W	<b>SPI 接收缓冲器清空控制位</b> 0: 无效 1: 清空接收缓冲器
TXCLR	bit30	W	<b>SPI 发送缓冲器清空控制位</b> 0: 无效 1: 清空发送缓冲器
—	bit29-27	—	—
DW<2:0>	bit26-24	R/W	<b>SPI 发送帧位宽（1~8 位）</b> 一帧数据位宽为 DW<2:0>+1 位
TMP<5:0>	bit23-18	R/W	<b>SPI 帧发送间隔周期设置位（仅主控模式支持）</b> 详细说明见备注
TMS	bit17	R	<b>SPI 帧发送间隔状态标志位（仅主控模式支持）</b> 0: 非发送间隔状态

			1: 发送间隔状态
TME	bit16	R/W	<b>SPI 帧发送间隔使能位（仅主控模式支持）</b> 0: 禁止 1: 使能
—	bit15-8	—	—
DFS<1:0>	bit7-6	R/W	<b>SPI 通讯数据格式</b> 00: 上升沿发送（先），下降沿接收（后） 01: 下降沿发送（先），上升沿接收（后） 10: 上升沿接收（先），下降沿发送（后） 11: 下降沿接收（先），上升沿发送（后）
DRE	bit5	R/W	<b>SPI 延迟接收使能位（仅主控模式支持）</b> 0: 禁止 1: 使能
—	bit4	—	—
REN	bit3	R/W	<b>SPI 接收使能位</b> 0: 禁止 1: 使能（需 EN 同时使能）
MS	bit2	R/W	<b>SPI 通讯模式选择位</b> 0: 主控模式 1: 从动模式
RST	bit1	W	<b>SPI 软件复位</b> 0: 读取时始终为 0 1: 软件复位，自动清零
EN	bit0	R/W	<b>SPI 通讯使能位</b> 0: 禁止 1: 使能（SPI 通讯使能，但仅使能数据发送）

注 1: SPI 帧发送间隔周期计算公式如下:

$T_{SCK} * (1 + TMP)$ , 即时间间隔为 1~64 个通讯时钟周期  $T_{SCK}$ 。

注 2: 由于不同的通讯数据格式对端口的初始电平要求是不同的（参见上面各通讯波形示意图），因此若无法确定使能 SPI 之前的 SPI 端口的初始值，必须先配置通讯数据格式控制位，对 SPI 端口初始电平进行自动设置；然后再通过对 SPI\_CON 寄存器的 EN 和 REN 置 1 来使能 SPI 发送和接收。即 SPI\_CON 寄存器需要分两次写入，否则易产生通信错误。

### 5.3.11.2 SPI 发送数据写入寄存器（SPI\_TBW）

**SPI 发送数据写入寄存器（SPI\_TBW）**

偏移地址: 08<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

TBW<31:0>	bit31-0	W	<p>写入的发送数据</p> <p>字节写入时：仅允许对 TBW&lt;7:0&gt;写入</p> <p>半字写入时：仅允许对 TBW&lt;15:0&gt;写入</p> <p>字写入时：对 TBW&lt;31:0&gt;写入</p>
-----------	---------	---	-----------------------------------------------------------------------------------------------------------------------

### 5.3.11.3 SPI接收数据读取寄存器 (SPI\_RBR)

#### SPI 接收数据读取寄存器 (SPI\_RBR)

偏移地址：0C<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

RBR<31:0>	bit31-0	R	<p>读取的接收数据</p> <p>字节读取时：仅允许对 RBR&lt;7:0&gt;读取</p> <p>半字读取时：仅允许对 RBR&lt;15:0&gt;读取</p> <p>字读取时：对 RBR&lt;31:0&gt;读取</p>
-----------	---------	---	-----------------------------------------------------------------------------------------------------------------------

### 5.3.11.4 SPI中断使能寄存器 (SPI\_IE)

#### SPI 中断使能寄存器 (SPI\_IE)

偏移地址：10<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RBIM<1:0>		TBIM<1:0>		保留	TBWEIE	NSSIE	IDIE	ROIE	TEIE	RBIE	TBIE

—	bit31-12	—	—
RBIM<1:0>	bit11-10	R/W	<p><b>SPI 接收缓冲器满中断模式选择位</b></p> <p>00: RB0 满产生中断</p> <p>01: RB0 与 RB1 满产生中断</p> <p>10: RB0~ RB3 全满产生中断</p> <p>11: 保留</p>
TBIM<1:0>	bit9-8	R/W	<p><b>SPI 发送缓冲器空中断模式选择位</b></p> <p>00: TB0 空产生中断</p> <p>01: TB0 与 TB1 空产生中断</p> <p>10: TB0~TB3 全空产生中断</p> <p>11: 保留</p>
—	bit7	—	—

TBWEIE	bit6	R/W	<b>SPI 发送数据写错误中断使能位</b> 0: 禁止 1: 使能
NSSIE	bit5	R/W	<b>SPI 片选变化中断使能位（仅从动模式支持）</b> 0: 禁止 1: 使能
IDIE	bit4	R/W	<b>SPI 空闲状态中断使能位（仅主控模式支持）</b> 0: 禁止 1: 使能
ROIE	bit3	R/W	<b>SPI 接收数据溢出中断使能位</b> 0: 禁止 1: 使能
TEIE	bit2	R/W	<b>SPI 发送数据错误中断使能位（仅从动模式支持）</b> 0: 禁止 1: 使能
RBIE	bit1	R/W	<b>SPI 接收缓冲器满中断使能位</b> 0: 禁止 1: 使能
TBIE	bit0	R/W	<b>SPI 发送缓冲器空中断使能位</b> 0: 禁止 1: 使能

### 5.3.11.5 SPI中断标志寄存器（SPI\_IF）

**SPI 中断标志寄存器（SPI\_IF）**

偏移地址：14<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000001<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TBWEIF	NSSIF	IDIF	ROIF	TEIF	RBIF	TBIF

—	bit31-7	—	—
TBWEIF	bit6	R/W	<b>SPI 发送数据写错误中断标志位</b> 0: 未发生写错误 1: 发生写错误，可能会出现下列错误： 对 SPI_TBW 字写入时，TB0~TB3 未全空； 对 SPI_TBW 半字写入时，TB0~TB3 未半空； 对 SPI_TBW 字节写入时，TB0~TB3 全满； 对 SPI_TBW<31:16>进行半字写入； 对 SPI_TBW<31:8>进行字节写入。 软件写 1 清除标志位，写 0 无效
NSSIF	bit5	R/W	<b>SPI 片选变化中断标志位（仅从动模式支持）</b>

			0: 片选信号未发生变化 1: 片选信号发生变化 软件写 1 清除标志位, 写 0 无效
IDIF	bit4	R/W	<b>SPI 空闲中断标志位 (仅主控模式支持)</b> 0: 未进入空闲状态 1: 进入空闲状态 软件写 1 清除标志位, 写 0 无效; 或软件写寄存器 SPI_TBW 清除标志位
ROIF	bit3	R/W	<b>SPI 接收数据溢出中断标志位</b> 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
TEIF	bit2	R/W	<b>SPI 发送错误中断标志位 (仅从动模式支持)</b> 0: 未发生发送错误 1: 发生发送错误: 发送缓冲器和发送移位寄存器全空时, 又收到主控方提供的通讯时钟 软件写 1 清除标志位, 写 0 无效
RBIF	bit1	R	<b>SPI 接收缓冲器满中断标志位</b> 读 SPI_RBR 寄存器可清除中断标志位
TBIF	bit0	R	<b>SPI 发送缓冲器空中断标志位</b> 写 SPI_TBW 寄存器可清除中断标志位

注 1: SPI 中断禁止时, 如果满足条件仍会置起对应的中断标志位, 只是不会产生中断请求。

注 2: 对 SPI\_IF 寄存器中的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

### 5.3.11.6 SPI 发送缓冲寄存器 (SPI\_TB)

#### SPI 发送缓冲寄存器 (SPI\_TB)

偏移地址: 18<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB3<7:0>								TB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB1<7:0>								TB0<7:0>							

TB3<7:0>	bit31-24	R	发送数据缓冲器 3
TB2<7:0>	bit23-16	R	发送数据缓冲器 2
TB1<7:0>	bit15-8	R	发送数据缓冲器 1
TB0<7:0>	bit7-0	R	发送数据缓冲器 0

## 5.3.11.7 SPI接收缓冲寄存器 (SPI\_RB)

## SPI 接收缓冲寄存器 (SPI\_RB)

偏移地址: 1C<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB3<7:0>								RB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1<7:0>								RB0<7:0>							

RB3	bit31-24	R	接收数据缓冲器 3
RB2	bit23-16	R	接收数据缓冲器 2
RB1	bit15-8	R	接收数据缓冲器 1
RB0	bit7-0	R	接收数据缓冲器 0

## 5.3.11.8 SPI状态寄存器 (SPI\_STA)

## SPI 状态寄存器 (SPI\_STA)

偏移地址: 20<sub>H</sub>复位值: 00000000\_00000001\_00001111\_10000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															IDLE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBFF3	RBFF2	RBFF1	RBFF0	TBEF3	TBEF2	TBEF1	TBEF0	NSS	保留						

—	bit31-17	—	—
IDLE	bit16	R	<b>SPI 空闲标志位</b> (仅主控模式支持) 0: 非空闲状态 1: 空闲状态
RBFF3	bit15	R	<b>RB3 满标志位</b> 0: 空 1: 满
RBFF2	bit14	R	<b>RB2 满标志位</b> 0: 空 1: 满
RBFF1	bit13	R	<b>RB1 满标志位</b> 0: 空 1: 满
RBFF0	bit12	R	<b>RB0 满标志位</b> 0: 空 1: 满
TBEF3	bit11	R	<b>TB3 空标志位</b> 0: 满

			1: 空
TBEF2	bit10	R	<b>TB2 空标志位</b> 0: 满 1: 空
TBEF1	bit9	R	<b>TB1 空标志位</b> 0: 满 1: 空
TBEF0	bit8	R	<b>TB0 空标志位</b> 0: 满 1: 空
NSS	bit7	R	<b>SPI 片选标志位（仅从动模式支持）</b> 0: 选中 1: 未选中
—	bit6-0	—	—

### 5.3.11.9 SPI波特率设置寄存器（SPI\_CKS）

**SPI 波特率设置寄存器（SPI\_CKS）**

偏移地址：24<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00001000

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

—	bit31-8	—	—
CKS<7:0>	bit7-0	R/W	<b>SPI 通讯波特率设置位（仅主控模式支持）</b> SPI 通讯波特率计算公式如下： CKS<7:0>=0x00 时：F <sub>PCLK</sub> ； CKS<7:0>=0x01~0xFF 时：F <sub>PCLK</sub> /(CKS * 2)

注 1：SPI 通讯波特率计算公式如下，SPI\_CKS 寄存器中：

CKS<7:0>=0x00 时：F<sub>PCLK</sub>；

CKS<7:0>=0x01~0xFF 时：F<sub>PCLK</sub>/(CKS \* 2)。

### 5.3.12 SPI应用说明

芯片支持 1 个 SPI 同步串口通讯控制器 SPI1。

为保证 SPI 的正常通信，SPI 的配置需遵循如下要求：

- 1) 选择 SPI 通讯端口使用滤波器时，SPI 的通讯时钟频率必须小于 20Mz，为确保通讯稳定性，建议在 SPI 通讯过程中，使能 SPI 通讯端口滤波器。
- 2) SPI 的帧位宽与 SPI\_CKS 寄存器 CKS 的配置存在以下关系：



当 SPI 帧位宽选择 5~8 位时,  $CKS \geq 0$ ;

当 SPI 帧位宽选择 2~4 位时,  $CKS \geq 1$ ;

当 SPI 帧位宽选择 1 位时,  $CKS > 2$ 。

- 3) 由于不同的通讯数据格式对端口的初始电平要求是不同的(参见上面各通讯波形示意图), 因此若无法确定使能 SPI 之前的 SPI 端口的初始值, 必须先配置通讯数据格式控制位, 对 SPI 端口初始电平进行自动设置; 然后再通过对 SPI\_CON 寄存器的 EN 和 REN 置 1 来使能 SPI 发送和接收, 即对 SPI\_CON 寄存器分两步写入。

## 5.4 I2C总线串口通讯控制器（I2C0）

### 5.4.1 概述

- ◇ 支持单主控模式
  - 支持自动重复寻呼功能
  - 支持自动发送“停止位”功能
  - 支持数据应答延迟功能
  - 支持数据帧传输间隔功能
  - 支持软件触发“起始位”
  - 支持软件触发“停止位”
  - 支持软件触发数据接收，接收模式可配
- ◇ 支持从动模式
  - 支持7位从机地址可配
  - 支持从机地址匹配中断标志
  - 支持接收“停止位”中断标志
  - 支持时钟线自动下拉等待请求功能
  - 支持自动发送“未应答”功能
- ◇ 支持4级发送缓冲器和4级接收缓冲器
- ◇ 通讯端口 SCL0 和 SDA0，均支持输出模式可配置：推挽输出或开漏输出
- ◇ 通讯端口 SCL0 和 SDA0 支持16倍速采样器可配置
- ◇ 支持发送和接收缓冲器空/满中断
- ◇ 支持起始位中断、停止位中断
- ◇ 支持接收数据溢出中断、发送数据写错误中断

### 5.4.2 结构框图

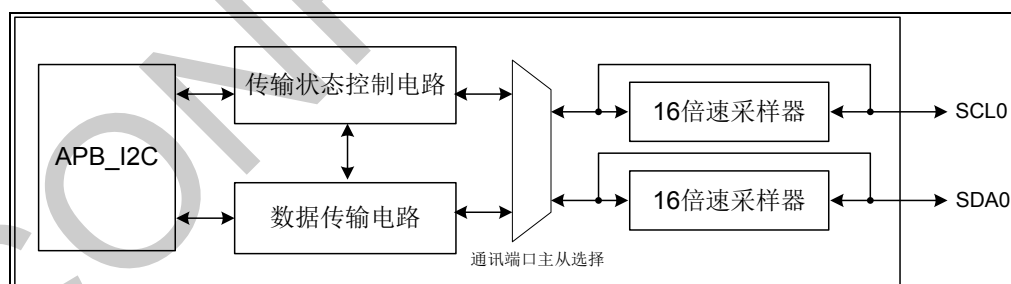


图 5-32 I2C 电路结构框图

### 5.4.3 I2C总线基本原理

#### 5.4.3.1 I2C通讯协议

I2C 总线通讯协议，读写操作示意图如下所示：

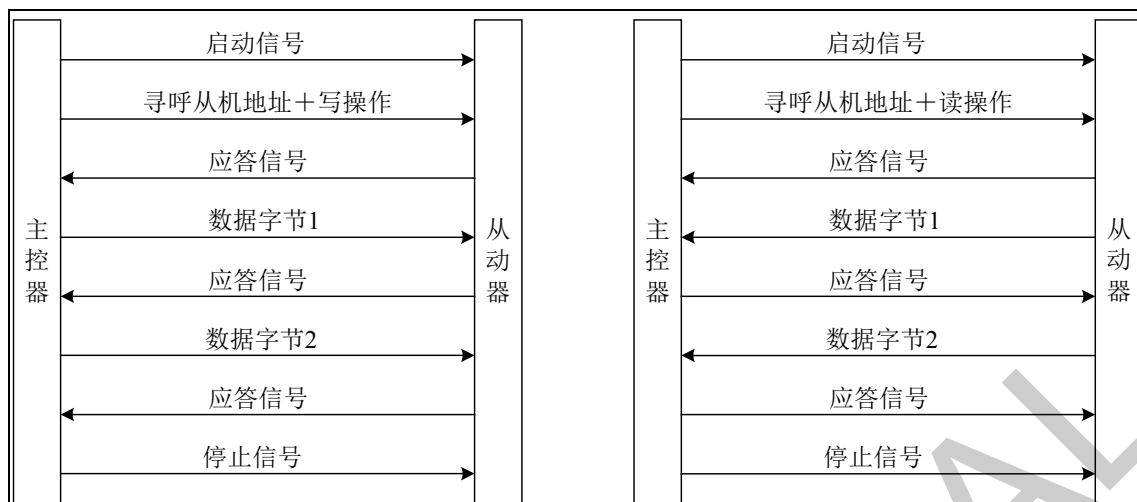


图 5-33 I2C 总线通讯协议示意图

I2C 通讯由主控器发起，发送启动信号 **S** 控制总线，发送停止信号 **P** 释放总线。

I2C 总线上可以同时有多个主控器（前提是每个主控器都支持多主机仲裁机制），并至少需要一个从动器，且每一个从动器都必须有一个独立且唯一的寻呼地址。

主控器在发送启动信号后，紧接着发送寻呼从机地址和读写控制位。

读写控制位  $R/\overline{W}$ ，用于通知从动器数据传送的方向，“0”表示由主控器向从动器“写”数据，“1”表示由主控器向从动器“读”数据。

I2C 通讯协议支持应答机制，即发送方每传送一个字节的的数据（包括寻呼地址），接收方必须反馈一个应答信号（**ACK** 或 **NACK**），发送方再根据应答信号进行下一步的操作。

如果主控器和从动器的时钟端口（**SCL**）都使用输出开漏设计，且主控器支持时钟线等待请求操作，那么从动器可以在时钟线为低电平时下拉时钟线，使主控器等待从动器，直到从动器释放时钟线。

I2C 通讯时，每个数据字节在传输时都是高位在前，低位在后。

I2C 通讯时，数据线 **SDA** 的数据信号电平，只在时钟线 **SCL** 的低电平期间变化，在 **SCL** 高电平期间应保持稳定。如果 **SDA** 电平在 **SCL** 高电平期间变化，则会触发起始位或停止位，由高到低变化触发起始位，由低到高变化触发停止位。

#### 5.4.3.2 I2C数据传输格式

I2C 通讯时，根据从动器的具体设计规格，确定实际的数据传输格式。以下只介绍一种常用的 I2C 通讯数据传输格式：

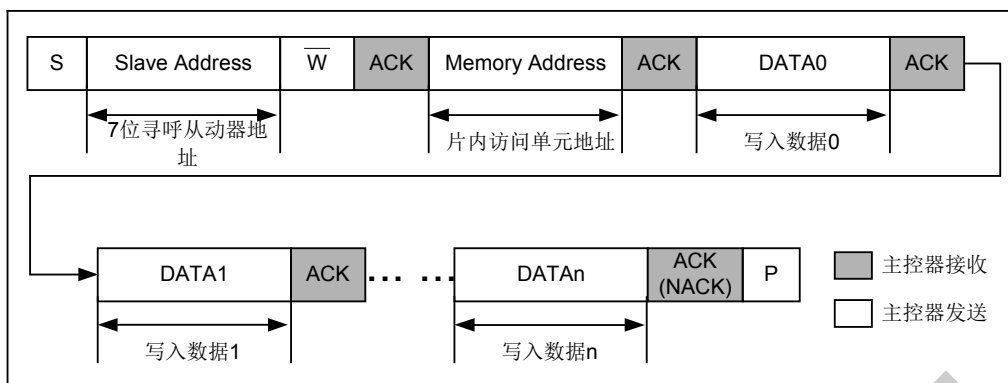


图 5-34 I2C 主控器写入从动器数据示意图

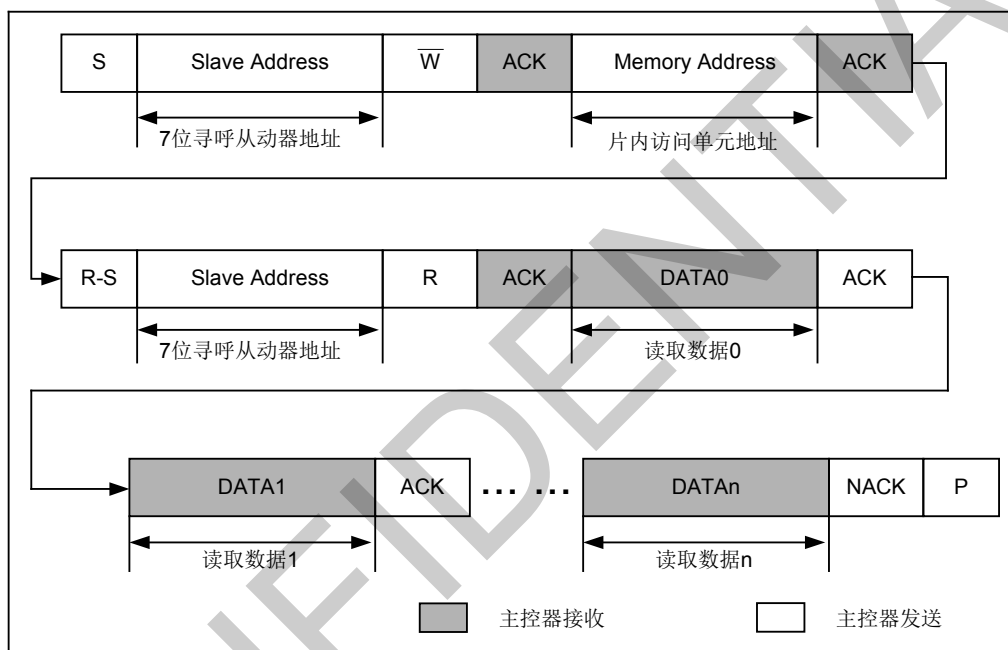


图 5-35 I2C 主控器读取从动器数据示意图

#### 5.4.4 I2C 通讯端口配置

I2C 通讯端口 SCL0 和 SDA0，均支持推挽输出和开漏输出两种模式，配置 I2C\_CON 寄存器的 SCLOD 和 SDAOD，可分别进行选择。

推挽输出是 I/O 端口的标准输出，输出数据 0 和 1 时，I/O 端口电平也分别为 0 和 1。

对推挽输出模式，存在端口电平冲突的风险。例如：当主控器输出 0，而从动器输出 1 时，会发生端口信号电平冲突，导致端口状态不确定。

开漏输出是 I2C 总线协议中的标准模式，可以避免端口电平冲突问题。开漏输出端口的示意图如下所示：

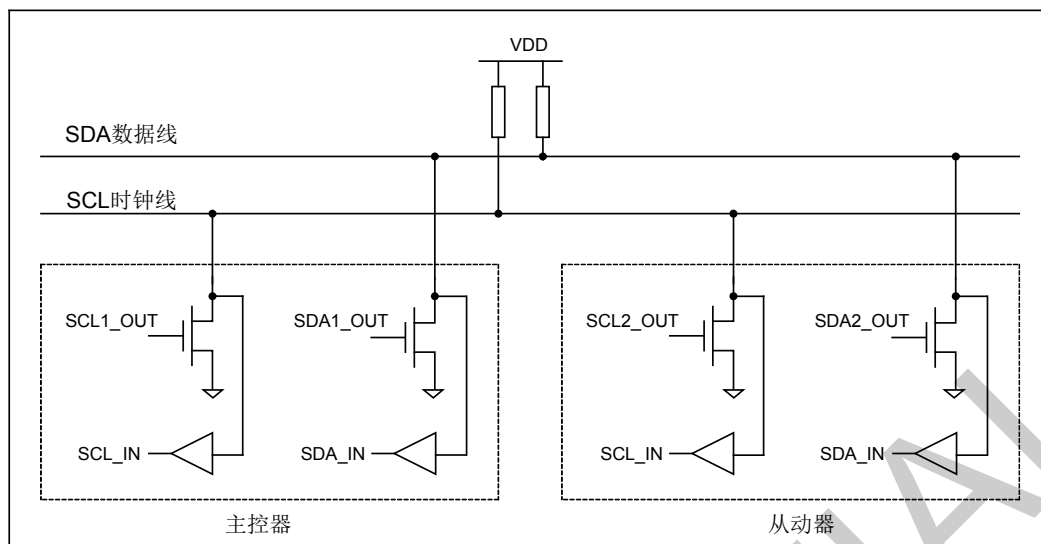


图 5-36 开漏输出端口示意图

开漏输出端口的高电平由 I2C 总线的上拉电阻提供，而低电平由主控器与从动器共同决定。任何一方都可以将总线电平下拉到 0，但只有当双方都释放总线后，总线电平才能被上拉到 1。

#### 5.4.5 I2C 时基定时器与 16 倍速采样器

通讯端口 SCL0 和 SDA0 支持 16 倍采样器，配置 I2C\_CON 寄存器的 SCLSE, SDASE 可使能两个端口的采样器。I2C 时基定时器的计数周期，作为 16 倍速采样器的采样周期。

在 I2C 主控模式下，I2C 时基定时器还用于提供通讯传输波特率。

如果需要使用 16 倍速采样器，或使用 I2C 主控模式，均必须使能 I2C 时基定时器。配置 I2C\_CON 寄存器的 TJE，可使能该时基定时器，配 TJP，可设定时基定时周期。

I2C 主控模式下，总线端口信号的时序参数如下：

参数名称	参数符号	使能 16 倍速采样器	禁止 16 速倍采样器
启动/重启动位建立时间	$T_{SU:S}$	$> T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
启动/重启动位保持时间	$T_{HD:S}$	$> T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
停止位建立时间	$T_{SU:P}$	$> T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
停止位保持时间	$T_{HD:P}$	$> T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
数据/应答位建立时间	$T_{SU:DA}$	$> T_{osc} \times (TJP+1) \times 4$	$T_{osc} \times (TJP+1) \times 4$
数据/应答位保持时间	$T_{HD:DA}$	$> T_{osc} \times (TJP+1) \times 8$	$T_{osc} \times (TJP+1) \times 4$
通讯时钟高电平脉宽	$T_{HIGH}$	$T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
通讯时钟低电平脉宽	$T_{LOW}$	$T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$

表 5-1 I2C 总线端口信号的时序参数列表

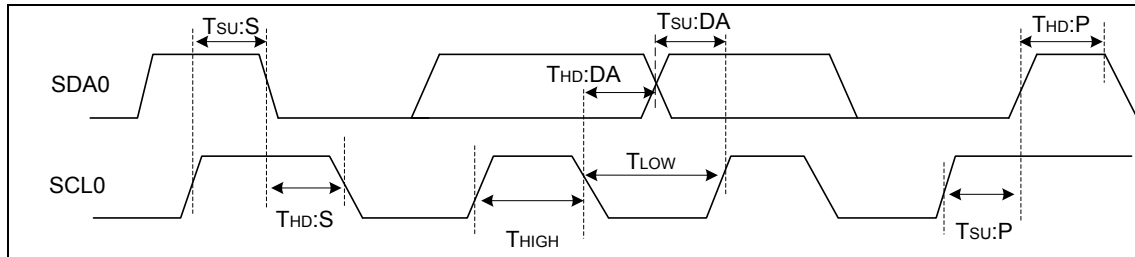


图 5-37 I2C 总线端口信号的波形示意图

使能 16 倍速采样器后，采样器对 I2C 总线进行采样，而 I2C 总线由于上拉电阻的作用，开始时的电平是不稳定的，当 I2C 总线电平稳定后，采样器的输出才能稳定。因此，这段不稳定的时间将造成 I2C 传输波特率的下降，下降程度取决于总线电平上升边沿的时间。

I2C 主控模式下，传输波特率计算公式如下（F<sub>osc</sub> 为系统时钟频率）：

使能通讯端口的 16 倍速采样器时：F<sub>SCL</sub>=F<sub>osc</sub> / ((TJP+1) x 24)；

禁止通讯端口的 16 倍速采样器时：F<sub>SCL</sub>=F<sub>osc</sub> / ((TJP+1) x 16)。

#### 5.4.6 I2C 通讯发送器

支持 4 级发送缓冲器 TB0, TB1, TB2, TB3 和 1 级发送移位寄存器，可进行数据的连续发送，直到发送缓冲器和移位寄存器全空，最多可连续写入和发送 5 帧数据。发送缓冲器 TB0~TB3 为只读寄存器，只能通过发送数据寄存器 I2C\_TBW 写入。

发送数据寄存器 I2C\_TBW 为一个虚拟地址单元，物理上不存在实际的寄存器电路，写该寄存器地址单元时，实际上是将发送数据写入到发送缓冲器 TB0~TB3 中，再传输到发送移位寄存器，通过数据端口 SDA0 进行数据发送。

发送数据寄存器 I2C\_TBW 支持 3 种写入方式：字节写入，半字写入和字写入。

字节方式写入 I2C\_TBW 时，发送数据被写入到发送缓冲器 TB0；半字方式写入 I2C\_TBW 时，发送数据被同时写入到发送缓冲器 TB0 和 TB1，其中低字节存放在 TB1 中；字方式写入 I2C\_TBW 时，发送数据被同时写入到发送缓冲器 TB0~TB3，其中低字节存放在 TB3 中。

发送数据从写入到发送到端口的数据流示意图如下所示：

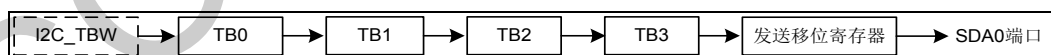


图 5-38 I2C 发送数据流示意图

支持发送缓冲器空中断，配置 I2C\_IE 寄存器的 TBIM，可选择中断模式。

TBIM<1:0>=00，为字节空产生中断，即发送缓冲器 TB0 为空时，会置起 I2C\_IF 寄存器的中断标志 TBIF；

TBIM<1:0>=01，为半字空产生中断，即发送缓冲器 TB0 和 TB1 均为空时，会置起 I2C\_IF 寄存器的中断标志 TBIF；

TBIM<1:0>=10，为字空产生中断，即发送缓冲器 TB0, TB1, TB2 和 TB3 均为空时，会置起 I2C\_IF 寄存器的中断标志 TBIF。

支持发送数据寄存器 I2C\_TBW 误写中断，当对 I2C\_TBW 的写入方式，与发送缓冲器 TB0~TB3 的空闲状态冲突时，或写入操作访问错误时，会置起 I2C\_IF 寄存器误写中断标志 TBWEIF。

#### 5.4.7 I2C通讯接收器

支持 4 级接收缓冲器 RB0, RB1, RB2, RB3 和 1 级接收移位寄存器，可进行数据的连续接收，直到接收缓冲器和移位寄存器全满，最多可连续接收 5 帧数据，再执行数据读取操作。读取接收数据寄存器 I2C\_RBR，可得到接收的数据，I2C\_STA 寄存器中对应的接收缓冲器清除满标志 RBFF0~RBFF3；也可以读取接收缓冲器 RB0~RB3 得到接收的数据，但不会清除满标志 RBFF0~RBFF3。

接收数据寄存器 I2C\_RBR 为一个虚拟地址单元，物理上不存在实际的寄存器电路，读该寄存器地址单元时，实际上是读取接收缓冲器 RB0~RB3 中的数据。

接收数据寄存器 I2C\_RBR 支持 3 种读取方式：字节读取，半字读取和字读取。

字节方式读取 I2C\_RBR 时，实际是读取接收缓冲器 RB0 的数据；半字方式读取 I2C\_RBR 时，实际是同时读取接收缓冲器 RB0 和 RB1 的数据，其中 RB0 中的数据为低字节；字方式读取 I2C\_RBR 时，实际是同时读取接收缓冲器 RB0~RB3，其中 RB0 中的数据为低字节。

接收数据从数据端口到各级缓冲器的数据流示意图如下所示：

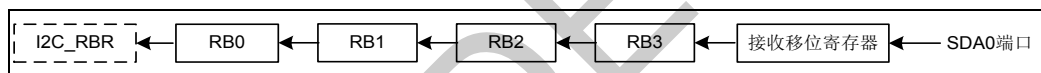


图 5-39 I2C 接收数据流示意图

接收缓冲器 RB0~RB3 的数据移到下一级缓冲器后，会清除其接收满标志 RBFF0~RBFF3。

当 4 级接收缓冲器和 1 级接收移位寄存器均满时，会立即置起接收数据溢出中断标志 I2C\_IF 寄存器中 ROIF，同时不会接收新数据。

支持接收缓冲器满中断，I2C\_IE 寄存器种配置 RBIM，可选择中断模式。

RBIM<1:0>=00，为字节满产生中断，即接收缓冲器 RB0 为满时，会置起 I2C\_IF 寄存器的中断标志 RBIF；

RBIM<1:0>=01，为半字满产生中断，即接收缓冲器 RB0 和 RB1 均为满时，会置起 I2C\_IF 寄存器的中断标志 RBIF；

RBIM<1:0>=10，为字满产生中断，即接收缓冲器 RB0, RB1, RB2 和 RB3 均为满时，会置起 I2C\_IF 寄存器的中断标志 RBIF。

#### 5.4.8 I2C通讯控制

配置 I2C\_CON 寄存器的 RST，可将 I2C 通讯模块软件复位，复位后：禁止数据通讯 EN=0；I2C\_IE 寄存器中禁止相关中断 SRIE=0, SPIE=0, TBIE=0, TBWEIE=0, RBIE=0, TEIE=0, ROIE=0, NAIE=0；I2C\_IF 寄存器中复位相关中断标志为默认值 SRIF=0, SPIF=0, TBIF=1, TBWEIF=0, RBIF=0, TEIF=0, ROIF=0, NAIF=0；置起空闲标志 IDLE=1；置起各发送缓冲器空标志 TBEF0~TBEF3=1；清除各接收缓冲器满标志 RBFF0~RBFF3=0。

#### 5.4.8.1 I2C起始位

配置 SRT，可触发 I2C 发送起始位，启动或重新启动一次传输操作，发送寻呼从动器地址。起始位波形图如下：

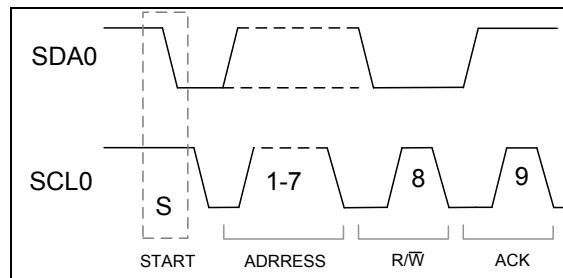


图 5-40 I2C 起始位波形图

I2C 主控模式，支持自动寻呼功能。配置 I2C\_MOD 寄存器的 SRAE，可启用自动寻呼功能，I2C 主控器会自动判断“地址应答”位，若该“地址应答”位为未应答 NACK，则自动发送起始位，重新启动本次地址寻呼操作，并且直到接收到应答 ACK 才停止继续重新启动。在芯片应用时，使能 I2C 自动寻呼功能前，需确保被寻呼的地址是真实存在的，否则会造成主控器芯片持续重新启动地址寻呼。

举例说明自动寻呼功能：芯片通过 I2C 向 EEPROM 器件写入数据时，存在写等待时间。在 EEPROM 存储器本身写数据期间，主控器芯片寻呼该器件时，将收到未应答 NACK。可有两种方法解决 EEPROM 写数据期间的地址寻呼问题：一种是主控器芯片设定寻呼间隔，在 EEPROM 数据写操作完成后，再寻呼该器件；另一种是主控器芯片启动自动寻呼功能，持续寻呼该器件，直到收到应答 ACK 为止。

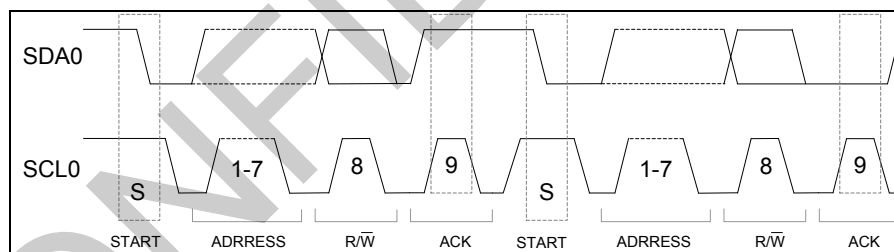


图 5-41 I2C 自动寻呼波形图

#### 5.4.8.2 I2C停止位

配置 I2C\_MOD 寄存器的 SPT，可触发 I2C 发送停止位，结束本次传输操作。停止位波形图如下：

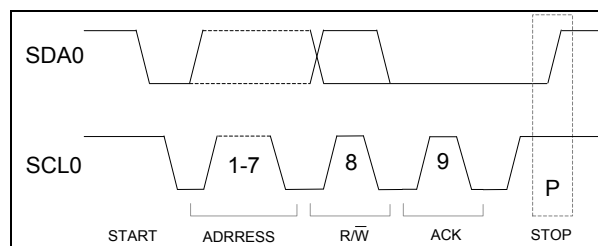


图 5-42 I2C 停止位波形图



I2C 主控模式，支持自动结束功能。配置 I2C\_MOD 寄存器的 SPAE，可使能自动结束功能，在发送 NACK 或接收 NACK 后，自动发送停止位，结束本次不成功的传输操作。自动结束功能的优先级小于自动寻呼功能。

#### 5.4.8.3 I2C 应答延迟功能

I2C 的主控模式，支持应答延迟功能，配置 I2C\_MOD 寄存器的 ADE，可使能该功能，配置 ADLY，可设定应答延迟的时间。应答延迟功能使能后，I2C 主控器将延迟发送通讯时钟 SCL0 的应答位脉冲。

当从动器不能按照正常数据的通讯速率，对应答位进行接收和发送时，主控器可使能应答延迟功能，并根据从动器的具体设计规格，设定应答延迟时间。

举例说明 I2C 应答延迟功能：I2C\_MOD 寄存器中 ADLY<2:0>=001，延迟时间为 1 个  $T_{SCL0}$ ，通讯波形示意图如下：

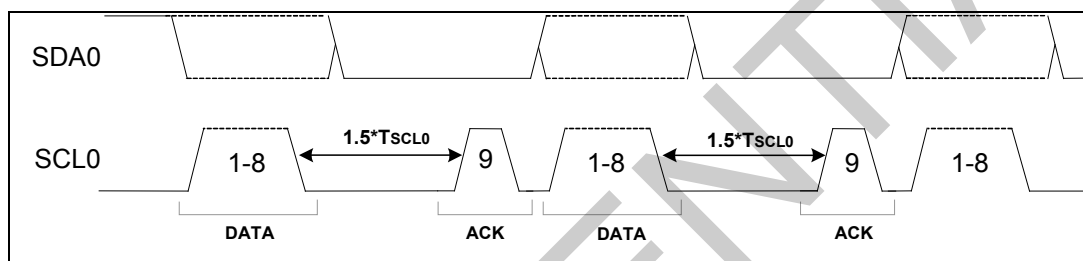


图 5-43 I2C 应答延迟功能波形示意图

#### 5.4.8.4 I2C 数据帧传输间隔功能

I2C 主控模式，支持数据帧传输间隔功能，配置 I2C\_MOD 寄存器的 TIS，可使能该功能，并设定间隔的时间。数据帧传输间隔时间设定后，在当前数据帧的应答位脉冲之后，I2C 主控器将延迟一段设定的时间，再发送下一个数据帧的通讯脉冲。

当从动器不能及时读取接收到的数据，或准备好发送的数据时，主控器可根据从动器的具体设计规格，设定数据帧的传输间隔。

举例说明 I2C 数据帧传输间隔功能：I2C\_MOD 寄存器中 TIS<3:0>=0001，间隔时间为 1 个  $T_{SCL0}$ ，通讯波形示意图如下：

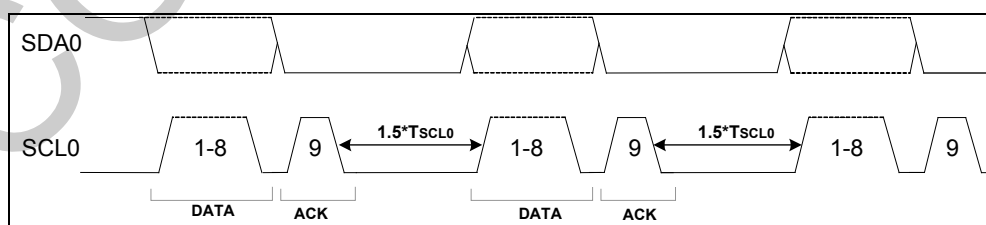


图 5-44 I2C 数据帧传输间隔功能波形示意图

#### 5.4.8.5 I2C 时钟线自动下拉等待请求功能

I2C 从动模式，支持时钟线自动下拉等待请求功能，配置 I2C\_MOD 寄存器的 CSE，可使能该功能。

为实现 I2C 时钟线的下拉等待请求功能，还需配置 I2C\_CON 寄存器的 SCLOD，将通讯端口 SCL0 选择为开漏输出模式，通过上拉电阻提供高电平，使从动器可对时钟线下拉控制，使主控器等待。

在通常情况下，从动器处于释放时钟线的状态，时钟线 SCL0 完全由主控器控制。但当从动器出现异常情况，短时间内无法继续进行数据传输时，从动器可以在时钟线 SCL0 为低电平时输出 0（不可以高电平时输出 0，否则会破坏数据传输过程），强行使 SCL0 保持低电平，使主控器进入通讯等待状态，直到从动器释放时钟线。时钟线下拉等待请求波形示意图如下：

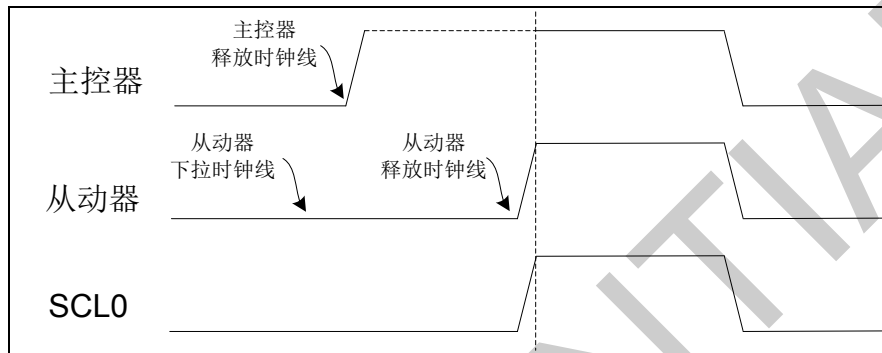


图 5-45 I2C 时钟线下拉等待波形示意图

I2C 从动模式下，时钟自动下拉等待请求功能使能后，当接收到本芯片寻呼地址和读操作位时，如果 I2C 的发送缓冲器和发送移位寄存器全空，且自动发送未应答使能位 ANAE=0，则会自动将时钟线下拉；当接收到本芯片寻呼地址和写操作位时，如果 I2C 的接收缓冲器和接收移位寄存器全满，则会自动将时钟线下拉。

#### 5.4.8.6 I2C自动发送未应答功能

I2C 从动模式，支持自动发送未应答 NACK 功能，配置 I2C\_MOD 寄存器的 ANAE，可使能该功能。从动器使能自动发送未应答功能时，不会强制控制时钟线（即使时钟线自动下拉使能位 CSE=1），适用于通讯端口 SCL0 选择为推挽输出或开漏输出模式两种情况。

I2C 从动模式下，自动发送未应答 NACK 功能使能后，当接收到本芯片寻呼地址和读操作位时，如果 I2C 的发送缓冲器和发送移位寄存器全空，则会自动发送未应答 NACK；当接收到本芯片寻呼地址和写操作位时，如果 I2C 的接收缓冲器和接收移位寄存器全满，则会自动发送未应答 NACK，通知主控器重新通讯。

## 5.4.9 特殊功能寄存器

### 5.4.9.1 I2C控制寄存器 (I2C\_CON)

I2C 控制寄存器 (I2C\_CON)

偏移地址: 00<sub>H</sub>复位值: 00000000\_00000000\_11111111\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								SA<6:0>							RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TJP<7:0>								TJE	保留	SDASE	SCLSE	SDAOD	SCLOD	RST	EN

—	bit31-24	—	—
SA<6:0>	bit23-17	R/W	从机地址位 主控模式: 触发“启动/重新启动”操作时, 自动发送从动模式: 接收到“启动/重新启动”后用于匹配比较
RW	bit16	R/W	<b>I2C 读写控制位</b> 0: 写操作 1: 读操作 主控模式: 该位可读可写, 触发“启动/重新启动”操作时, 自动发送该位 从动模式: 该位只可读, 从机地址匹配后, 硬件自动根据接收到的控制位值, 更新该位
TJP<7:0>	bit15-8	R/W	<b>I2C 时基定时周期设置位</b> 00~FF: 分别为 1~256 个 TPCLK
TJE	bit7	R/W	<b>I2C 时基定时器使能位</b> 0: 禁止 1: 使能
—	bit6	—	—
SDASE	bit5	R/W	<b>SDA 端口 16 倍速采样使能位</b> 0: 禁止 1: 使能
SCLSE	bit4	R/W	<b>SCL 端口 16 倍速采样使能位</b> 0: 禁止 1: 使能
SDAOD	bit3	R/W	<b>SDA 端口输出模式选择位</b> 0: 推挽输出 1: 开漏输出
SCLOD	bit2	R/W	<b>SCL 端口输出模式选择位</b> 0: 推挽输出 1: 开漏输出
RST	bit1	W	<b>I2C 软件复位</b> 0: 读取时始终为 0 1: 软件复位, 硬件自动清零

EN	bit0	R/W	<b>I2C 通讯使能位</b> 0: 禁止 1: 使能
----	------	-----	------------------------------------

#### 5.4.9.2 I2C工作模式寄存器 (I2C\_MOD)

I2C 工作模式寄存器 (I2C\_MOD)

偏移地址: 04<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							TAS	保留				BLD	RDT	SPT	SRT
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIS<3:0>				ADE	ADLY<2:0>			SPAE	SRAE	ANAE	CSE	RDM<2:0>			MS

—	bit31-25	—	—
TAS	bit24	R/W	<b>I2C 发送应答设置位 (仅从动模式支持)</b> 0: 发送 ACK 1: 发送 NACK
—	bit23-20	—	—
BLD	bit19	R/W	<b>I2C 总线释放功能控制位 (仅主控模式支持)</b> 0: 无效 1: SDA 端口为高电平, SCL 端口发送 8 个时钟。 该功能需配合 I2C 起始位和停止位使用, 达到释放时钟线和数据线的目的。
RDT	bit18	R/W	<b>I2C 接收数据触发位 (仅主控模式支持)</b> 0: 无效 1: 开始发送通讯时钟, 接收数据, 由 RDM<2:0> 配置接收数据模式
SPT	bit17	R/W	<b>I2C 停止位触发位 (仅主控模式支持)</b> 0: 无效 1: 触发停止位
SRT	bit16	R/W	<b>I2C 起始位触发位 (仅主控模式支持)</b> 0: 无效 1: 触发起始位, 产生起始位发送完成中断标志
TIS<3:0>	bit15-12	R/W	<b>I2C 数据帧传输间隔设置位 (仅主控模式支持)</b> 0000: 禁止 0001~1111: 分别为 1~15 个 I2C 通讯时钟周期
ADE	bit11	R/W	<b>I2C 应答延迟使能位 (仅主控模式支持)</b> 0: 禁止 1: 使能
ADLY<2:0>	bit10-8	R/W	<b>I2C 应答延迟时间设置位 (仅主控模式支持)</b> 000: 0.5 个 I2C 通讯时钟周期

			001: 1 个 I2C 通讯时钟周期 010: 1.5 个 I2C 通讯时钟周期 011: 2 个 I2C 通讯时钟周期 100: 2.5 个 I2C 通讯时钟周期 101: 3 个 I2C 通讯时钟周期 110: 3.5 个 I2C 通讯时钟周期 111: 4 个 I2C 通讯时钟周期
SPAE	bit7	R/W	<b>I2C 自动结束使能位（仅主控模式支持）</b> 0: 禁止 1: 使能（当发送或接收 NACK 后，自动发送停止位，优先级小于 SRAE）
SRAE	bit6	R/W	<b>I2C 自动寻呼使能位（仅主控模式支持）</b> 0: 禁止 1: 使能（若寻呼地址的应答位为 NACK，则自动重启本次寻呼操作）
ANAE	bit5	R/W	<b>I2C 自动发送未应答使能位（仅从动模式支持）</b> 0: 禁止 1: 使能
CSE	bit4	R/W	<b>I2C 时钟线自动下拉等待请求使能位（仅从动模式支持）</b> 0: 禁止 1: 使能
RDM<2:0>	bit3-1	R/W	<b>I2C 接收模式选择位（仅主控模式支持）</b> 000: 接收 1 字节，发送 ACK 001: 接收 1 字节，发送 NACK 010: 连续接收 2 字节，每个字节发送 ACK 011: 连续接收 2 字节，前 1 字节发送 ACK，后 1 字节，发送 NACK 100: 连续接收 4 字节，每个字节发送 ACK 101: 连续接收 4 字节，前 3 字节发送 ACK，后 1 字节，发送 NACK 110: 连续接收，每个字节发送 ACK 111: 完成该字节接收，发送 NACK
MS	bit0	R/W	<b>I2C 通讯模式选择位</b> 0: 主控模式 1: 从动模式

注：如果 ANAE 与 CSE 同时为 1，则 ANAE 优先级高，即发送未应答位后，不会将 I2C 时钟线自动下拉。

## 5.4.9.3 I2C中断使能寄存器 (I2C\_IE)

I2C 中断使能寄存器 (I2C\_IE)

偏移地址: 08<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIDLEIE	RBIM<1:0>	TBIM<1:0>	TBWEIE	NAIE	ROIE	TEIE	RBIE	TBIE	SPIE	SRIE				

—	bit31-13	—	—
TIDLEIE	bit12	R/W	<b>I2C 发送空闲中断使能位</b> 0: 禁止 1: 使能
RBIM<1:0>	bit11-10	R/W	<b>I2C 接收缓冲器满中断模式选择位</b> 00: 字节满产生中断 01: 半字满产生中断 10: 字满产生中断 11: 保留
TBIM<1:0>	bit9-8	R/W	<b>I2C 发送缓冲器空中断模式选择位</b> 00: 字节空产生中断 01: 半字空产生中断 10: 字空产生中断 11: 保留
TBWEIE	bit7	R/W	<b>I2C 发送数据写错误中断使能位</b> 0: 禁止 1: 使能
NAIE	bit6	R/W	<b>I2C 未应答 NACK 中断使能位</b> 0: 禁止 1: 使能
ROIE	bit5	R/W	<b>I2C 接收数据溢出中断使能位</b> 0: 禁止 1: 使能
TEIE	bit4	R/W	<b>I2C 发送数据错误中断使能位</b> 0: 禁止 1: 使能
RBIE	bit3	R/W	<b>I2C 接收缓冲器满中断使能位</b> 0: 禁止 1: 使能
TBIE	bit2	R/W	<b>I2C 发送缓冲器空中断使能位</b> 0: 禁止 1: 使能
SPIE	bit1	R/W	<b>I2C 停止位中断使能位</b>

			0: 禁止 1: 使能
SRIE	bit0	R/W	<b>I2C 起始位中断使能位</b> 0: 禁止 1: 使能

#### 5.4.9.4 I2C中断标志寄存器 (I2C\_IF)

##### I2C 中断标志寄存器 (I2C\_IF)

偏移地址: 0C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000100<sub>b</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIDLEIF		保留				TBWEIF	NAIF	ROIF	TEIF	RBIF	TBIF	SPIF	SRIF	

—	bit31-13	—	—
TIDLEIF	bit12	R/W	<b>I2C 发送空闲中断标志位</b> 0: 未产生空闲中断 1: 空闲中断标志 软件写 1 清除标志位, 写 0 无效
—	bit11-8	—	—
TBWEIF	bit7	R/W	<b>I2C 发送数据写错误中断标志位</b> 0: 未发生写错误 1: 发生写错误, 可能会出现下列错误: 对 I2C_TBW 字写入时, TB0~TB3 未全空; 对 I2C_TBW 半字写入时, TB0~TB3 未半空; 对 I2C_TBW 字节写入时, TB0~TB3 全满; 对 I2C_TBW<31:16>进行半字写入; 对 I2C_TBW<31:8>进行字节写入。 软件写 1 清除标志位, 写 0 无效
NAIF	bit6	R/W	<b>I2C 未应答中断标志位</b> 0: 未产生未应答 NACK 1: 产生未应答 NACK I2C 接收或发送 NACK 位后, 产生中断标志 软件写 1 清除标志位, 写 0 无效
ROIF	bit5	R/W	<b>I2C 接收数据溢出中断标志位</b> 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
TEIF	bit4	R/W	<b>I2C 发送错误中断标志位</b> 0: 未发生发送错误 1: 发生发送错误: 发送缓冲器和发送移位寄存器全空

			时，又收到主控方提供的通讯时钟 软件写 1 清除标志位，写 0 无效
RBIF	bit3	R	<b>I2C 接收缓冲器满中断标志位</b> 0: 非满 1: 满 读 I2C_RBR 寄存器可清除中断标志位
TBIF	bit2	R	<b>I2C 发送缓冲器空中断标志位</b> 0: 非空 1: 空 写 I2C_TBW 寄存器可清除中断标志位
SPIF	bit1	R/W	<b>I2C 停止位中断标志位</b> 0: 未产生停止位 1: 产生停止位 主控模式：发送停止位后产生中断标志。 从动模式：接收停止位后产生中断标志。 软件写 1 清除标志位，写 0 无效
SRIF	bit0	R/W	<b>I2C 起始位中断标志位</b> 0: 未产生起始位 1: 产生起始位 主控模式：如果禁止自动寻呼，发送完“起始位+地址+接收应答位后”产生中断标志。如果使能自动寻呼，发送完“起始位+地址+接受应答位”，并且应答位为 ACK 时，产生中断标志。 从动模式：接收到“起始位+地址位+读写控制位”，且地址匹配时，在发送应答位或未应答位之前，产生中断标志。 软件写 1 清除标志位，写 0 无效

注：对 I2C\_IF 寄存器中的各中断标志位，写 0 无效，写 1 才能清除标志位；读操作时，读取的值为 1 表示有中断发生。

#### 5.4.9.5 I2C 发送数据写入寄存器 (I2C\_TBW)

##### I2C 发送数据写入寄存器 (I2C\_TBW)

偏移地址：10<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

TBW<31:0>	bit31-0	W	<b>写入的发送数据</b> 字节写入时：仅允许对 TBW<7:0>写入 半字写入时：仅允许对 TBW<15:0>写入
-----------	---------	---	-------------------------------------------------------------------



			字写入时：对 TBW<31:0>写入
--	--	--	--------------------

#### 5.4.9.6 I2C接收数据读取寄存器 (I2C\_RBR)

##### I2C 接收数据读取寄存器 (I2C\_RBR)

偏移地址：14<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

RBR<31:0>	bit31-0	R	<p>读取的接收数据</p> <p>字节读取时：仅允许对 RBR&lt;7:0&gt;读取</p> <p>半字读取时：仅允许对 RBR&lt;15:0&gt;读取</p> <p>字读取时：对 RBR&lt;31:0&gt;读取</p>
-----------	---------	---	-----------------------------------------------------------------------------------------------------------------------

#### 5.4.9.7 I2C发送缓冲寄存器 (I2C\_TB)

##### I2C 发送缓冲寄存器 (I2C\_TB)

偏移地址：18<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB3<7:0>								TB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB1<7:0>								TB0<7:0>							

TB3<7:0>	bit31-24	R	发送数据缓冲器 3
TB2<7:0>	bit23-16	R	发送数据缓冲器 2
TB1<7:0>	bit15-8	R	发送数据缓冲器 1
TB0<7:0>	bit7-0	R	发送数据缓冲器 0

#### 5.4.9.8 I2C接收缓冲寄存器 (I2C\_RB)

##### I2C 接收缓冲寄存器 (I2C\_RB)

偏移地址：1C<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB3<7:0>								RB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1<7:0>								RB0<7:0>							

RB3<7:0>	bit31-24	R	接收数据缓冲器 3
----------	----------	---	-----------

RB2<7:0>	bit23-16	R	接收数据缓冲器 2
RB1<7:0>	bit15-8	R	接收数据缓冲器 1
RB0<7:0>	bit7-0	R	接收数据缓冲器 0

#### 5.4.9.9 I2C 状态寄存器 (I2C\_STA)

##### I2C 状态寄存器 (I2C\_STA)

偏移地址: 20<sub>H</sub>

复位值: 00000000\_00000010\_00001111\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														IDLE	ACK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBFF3	RBFF2	RBFF1	RBFF0	TBEF3	TBEF2	TBEF1	TBEF0	保留							

—	bit31-18	—	—
IDLE	bit17	R	<b>I2C 空闲标志位</b> 0: 非空闲状态 1: 空闲状态
ACK	bit16	R	<b>I2C 应答位</b> 0: 应答 ACK 1: 未应答 NACK
RBFF3	bit15	R	<b>RB3 满标志位</b> 0: 空 1: 满
RBFF2	bit14	R	<b>RB2 满标志位</b> 0: 空 1: 满
RBFF1	bit13	R	<b>RB1 满标志位</b> 0: 空 1: 满
RBFF0	bit12	R	<b>RB0 满标志位</b> 0: 空 1: 满
TBEF3	bit11	R	<b>TB3 空标志位</b> 0: 满 1: 空
TBEF2	bit10	R	<b>TB2 空标志位</b> 0: 满 1: 空
TBEF1	bit9	R	<b>TB1 空标志位</b> 0: 满 1: 空
TBEF0	bit8	R	<b>TB0 空标志位</b>

			0: 满 1: 空
—	bit7-0	—	—

#### 5.4.10 I2C应用说明

芯片支持 1 个 I2C 总线串口通讯控制器 I2C0。

I2C 总线在连续发送数据的应用中，利用发送空闲标志（TIDLEIF）进入中断的方式进行数据发送时，要注意以下 2 点：

- 1) 主控模式下，在主程序中设置 Memory Address 之后，发送空闲标志（TIDLEIF）置 1 并触发中断，在中断服务程序中，将数据写入 I2C\_TBW 中，并开始发送数据；
- 2) 从动模式下，必须在主程序中等待 I2C\_STA 寄存器的 IDLE 置 1 之后，将以 Memory Address 为起始地址的第一个数据写入 I2C\_TBW 中。当主机发送读命令后，发送空闲标志（TIDLEIF）置 1 并触发中断，在中断服务程序中读取接收的数据，在主机读完数据后，必须发送 STOP 位，否则，若要直接再次启动读操作，可能导致读取命令发送错误。

I2C\_IE 寄存器的 TBIM（发送缓冲器中断模式选择位）会影响 TIDLEIF，因此，对于字节发送或半字发送，在 I2C 开始发送数据时最好使用字空产生中断模式（TBIM=2'b10），否则会连续产生多个字节空中断，容易导致发送数据出错。如果使用字节空（TBIM=2'b00）或者半字空（TBIM=2'b01）中断模式，则发送开始时，必须向 I2C\_TBW 写满四个待发送的数据字节，否则也会连续产生多个字节空或半字空中断。

使用发送空闲标志（TIDLEIF）的好处在于，在 I2C 发送空闲中断使能位（TIDLEIE）有效的情况下，只需清除发送空闲标志位（TIDLEIF），即可达到连续发送数据的目的。

## 5.5 模数转换器（ADC）

### 5.5.1 概述

- ◇ 支持 12 位转换结果，有效精度为 11 位
- ◇ 采样速率最高支持 125ksps (kilo-samples per second)
- ◇ 支持 19 个模拟输入通道
- ◇ 支持 ADC 中断，可唤醒睡眠模式（仅在时钟源为 LRC 时唤醒）
- ◇ 支持正负向参考电压可配置
- ◇ 支持转换时钟可配置
- ◇ 支持自动转换比较功能

### 5.5.2 结构框图

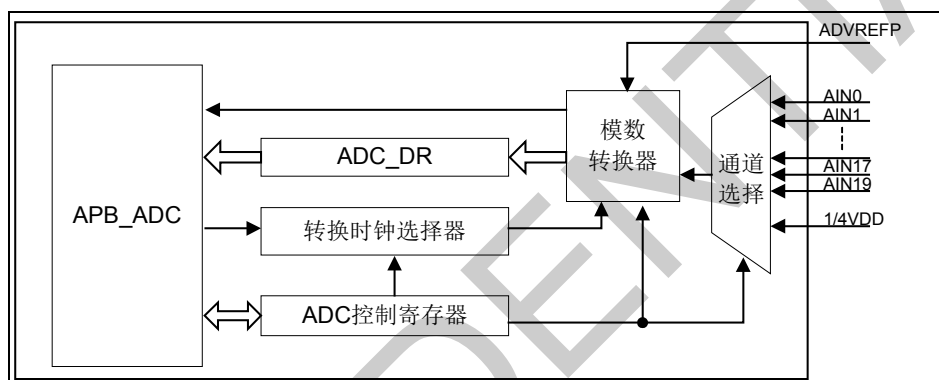


图 5-46 ADC 内部结构图

### 5.5.3 ADC 基本配置

将端口配置为 ADC 模拟输入通道的方式如下：

配置 GPIO\_PAINEB/GPIO\_PBINEB 寄存器，关断该端口的数字输入；配置 GPIO\_PADIR/GPIO\_PBDIR 寄存器，关断该端口的数字输出。

配置 ADC\_CHS 寄存器的 CHS<4:0>，选择相应的 ADC 模拟通道。

ADC 正常工作时必须使能的控制位：ADC\_VREFCON 寄存器的 IREF\_EN，ADC\_CON0 寄存器的 EN。

### 5.5.4 ADC 高精度参考电压

ADC 提供一个高精度内部 2.048V 的参考源，用作 ADC 的参考电压，可通过 ADC\_VREFCON 寄存器的 VREF\_EN 位使能。为了提高内部参考电压的稳定性，需先设置 ADC\_VREFCON 寄存器的 VREF\_EN 位使能内部参考，等待 300us 以后，再设置 CHOP\_EN 位使能内部参考电压斩波器，并等待 1ms 以后再触发 ADC 转换。

### 5.5.5 ADC 数据转换

配置 ADC\_CHS 寄存器的 CHS<4:0>，可选择 ADC 模拟通道；配置 ADC\_CON1 寄存器的 CLKS，可选择工作时钟源；配置 CLKDIV<2:0>，可选择时钟源预分频；配置 VREFP<1:0>，可选择正向参考电压，配置 VREFN，可选择负向参考电压（需固定

VREFN=1，选择内部地 VSS)，当 VREFP<1:0>为 1x 时，必须先设置 ADC\_VREFCON 寄存器的 VREF\_EN 位使能内部参考，并设置 IREF\_EN 和 ADC\_CON0 寄存器的 EN 位使能 ADC；等待 300us 以后，再设置 CHOP\_EN 位使能内部参考电压斩波器；等待 1ms 以后配置 TRIG，启动 A/D 转换，转换完成后，硬件电路自动将 TRIG 清零。

ADC 在每次转换完成后，会产生 ADC\_IF 寄存器的中断标志 IF，需软件清零；启动下一次 A/D 转换时，需重新配置 TRIG。

AD 采样支持硬件或软件控制，可通过 ADC\_CON1 寄存器的 SMPS 位进行选择，默认为硬件控制，通过配置 ADC\_CON0 寄存器的 TRIG 位，启动 A/D 采样和转换，采样时间最快为 1 个 ADC 时钟（取决于芯片的实际应用条件和 ADC 时钟源频率），转换时间为 14 个 ADC 时钟。AD 采样选择为软件控制时，通过配置 ADC\_CON1 寄存器的 SMPON 位，启动 A/D 采样和转换。

在使用 VDD 作参考电压，ADC 精度为 11 位时，最快采样速率为 125Ksps，即每秒可输出 125K 个高精度 ADC 转换值。通过配置 ADC\_CON1 寄存器的 CLKS 和 CLKDIV 寄存器来产生合适的 ADC 时钟。

ADC 时钟源选取系统时钟，采用内部参考电压 VREFP 时的建议配置如下表所示。

系统时钟	ADC 时钟分频比	ADC 精度	转换速率
48MHz	32	10.5 位	70Ksps

表 5-2 ADC 精度与转换速率的对应关系列表

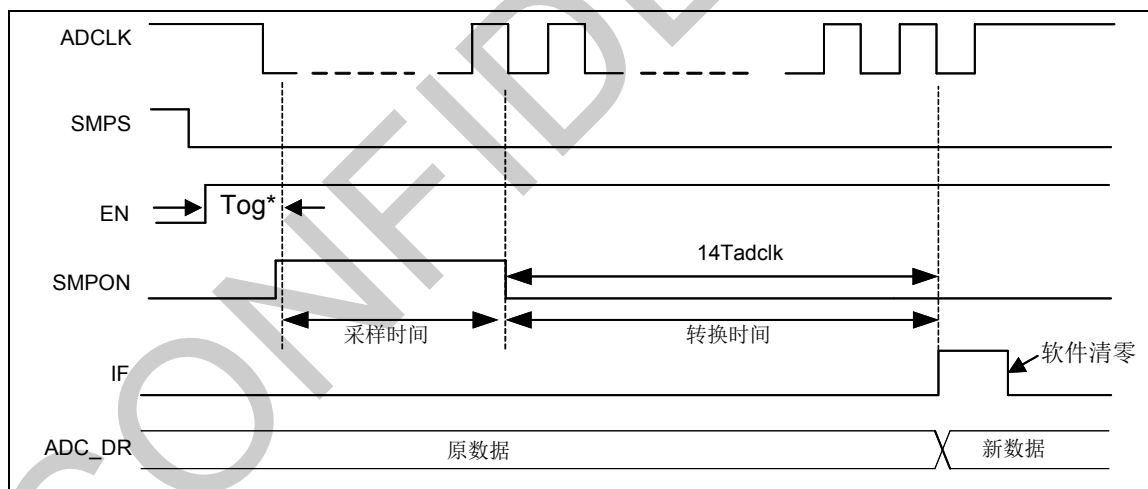


图 5-47 ADC 数据转换时序示意图（ADC\_CON1 寄存器的 SMPS=0，软件控制采样）

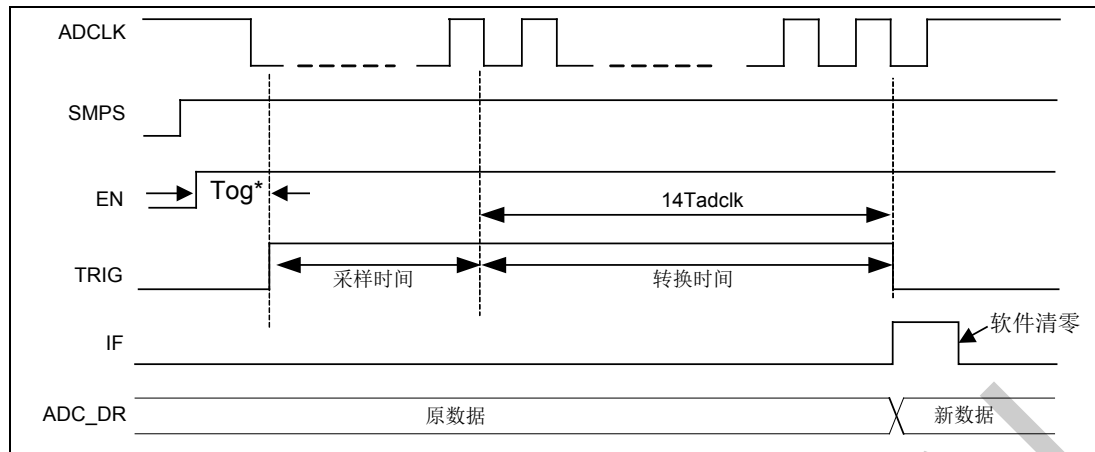


图 5-48 ADC 数据转换时序示意图 (ADC\_CON1 寄存器的 SMPS=1, 硬件控制采样)

注 1:  $Tog > 100\mu s$ ;

注 2: AD 转换时钟周期  $Tadclk$ , 可通过 ADC\_CON1 寄存器的 CLKS 和  $CLKDIV<2:0>$  配置不同的频率。

注 3: 实际应用中还需要考虑 ADC 自身工作的建立 (参见寄存器 ADC\_CON0 后面的备注描述内容)

#### 启动一次 A/D 转换例程

```

LDR    R0, =ADC_VREFCON    ;使能 VREF_EN 与 IREF_EN, 选择 2.048V
LDR    R1, =0X05
STR    R1, [R0]
LDR    R0, =ADC_CON1        ;A/D 时钟源选择 PCLK 的 32 分频,选择内部
                             ;参考电压 VREF 2.048V 为正向参考电压,
                             ;选择硬件控制采样, AD 转换高速使能

LDR    R1, =0X00021605
STR    R1, [R0]
LDR    R0, =ADC_CHS          ;选择 AIN3
LDR    R1, =0X03
STR    R1, [R0]
LDR    R0, =ADC_CON0          ;选择 12 位分辨率, 使能 ADC
LDR    R1, =0X31
STR    R1, [R0]
.....                          ;延时约 300us
LDR    R0, =ADC_VREFCON      ;使能 CHOP_EN
LDR    R1, =0X0D
STR    R1, [R0]
.....                          ;延时约 1ms
LDR    R0, =ADC_CON0          ;启动 A/D 转换
LDR    R1, =0X33
STR    R1, [R0]
WAIT4IF
LDR    R0, =ADC_IF            ;等待 ADC 中断
LDR    R1, =0X01
TST    R0, R1

```

```

BEQ      WAIT4IF
STR      R1, [R0]          ;清零 ADC 中断
.....

```

### 5.5.6 自动转换比较功能

芯片提供自动转换比较功能，可自动完成多次 AD 转换并计算出平均值，并根据所设定的阈值进行比较产生相应的中断，平均值和每次转换的结果均可读。

配置 ADC\_CON0 寄存器的 ACP\_EN 为 1 时，对 TRIG 写 1 则启动连续自动转换比较功能，并固定为硬件控制采样时间，对 SMPS 写 0 无效。启动此功能前必须先完成下面的配置：

配置 ADC\_CON1 寄存器的 ST 可设置采样时间，建议采样时间大于 1us。

配置 ADC\_ACPC 寄存器的 TIMES，可设置每个溢出时间周期内的 ADC 采样转换次数，ADC 转换达到设置的次数后，自动计算 ADC 转换平均值，保存在自动转换均值数据寄存器 ADC\_ACPMEAN 中，并进行自动转换结果比较，产生对应的阈值超出中断标志。

配置 OVFL\_TIMES，可设置自动转换比较溢出时间，每次计数溢出后，产生 ADC 自动转换溢出中断标志 ACPOVIF，如果此时采样转换次数（由 TIMES<1:0>设定）已满，则自动启动下一个自动转换比较过程；如果此时采样转换次数未满足，则溢出中断标志 ACPOVIF 无法被软件清除，当前 ADC 自动转换仍继续进行，直到完成所有 ADC 转换次数后，溢出中断标志才可被软件清除，才自动启动下一个 ADC 自动转换过程和溢出时间计算。

配置 ADC\_ACPC 寄存器的 CLKS 可选择溢出计数的时钟源，为 PCLK 或者 LRC(32KHz) 256 分频。如果需要在芯片浅睡眠或深睡眠模式下，自动转换比较模块仍保持工作，则在进入睡眠模式之前，需要将计数时钟源设置为 LRC 的 256 分频，并配置 ADC\_CON1 寄存器的 CLKS 选择 A/D 时钟源为 LRC。

配置 ADC\_ACPCMP 寄存器的 CMP\_MIN，设置自动比较低阈值，如果 ADC\_ACPMEAN 寄存器的 MEAN\_DATA 小于等于此阈值，则中断标志 ACPMINIF 置 1。

配置 ADC\_ACPCMP 寄存器的 CMP\_MAX，设置自动比较高阈值，如果 ADC\_ACPMEAN 寄存器的 MEAN\_DATA 大于等于此阈值，则中断标志 ACPMAXIF 置 1。

在 ADC 自动转换过程中，每次 ADC 转换完成后，均会产生 ADC 中断标志 IF(ADC\_IF<0>)，需软件清除。

在 ADC 自动转换过程中，如果关闭 ADC 使能位 EN (ADC\_CON0<0>)，则硬件立即自动将溢出计数，转换次数计数和自动转换均值数据寄存器清零；如果关闭自动转换比较使能位 ACP\_EN，则硬件会在当前的那次 ADC 转换完成后，自动将溢出计数，转换次数计数和自动转换均值数据寄存器清零，当前的 ADC 转换值寄存器 ADC\_DR 仍保留。

#### 启动一次 A/D 自动转换例程

```

LDR      R0, =ADC_VREFCON      ;使能 VREF_EN 与 IREF_EN，选择 2.048V
LDR      R1, =0X05
STR      R1, [R0]
LDR      R0, =ADC_CON1         ;A/D 时钟源选择 PCLK 的 32 分频,选择内部
                                ;参考电压 VREF 2.048V 为正向参考电压，

```

```

;选择硬件控制采样，AD 转换高速使能
LDR    R1, =0X00021605
STR     R1, [R0]
LDR     R0, =ADC_CHS          ;选择 AIN3
LDR     R1, =0X03
STR     R1, [R0]
LDR     ADC_IE, =0X07
LDR     ADC_ACPCMP, =0X00010001 ;设置自动比较的高/低阈值
LDR     ADC_ACPC, =0x0013001F   ;ACP 模块工作时钟为 LRC，每次溢出
;时间内自动转换 8 次，溢出时间为 32x Tacp
LDR     R0, =ADC_CON0          ;使能 ADC 与自动转换比较功能
LDR     R1, =0X35
STR     R1, [R0]
.....                          ;延时约 300us
LDR     R0, =ADC_VREFCON       ;使能 CHOP_EN
LDR     R1, =0X0D
STR     R1, [R0]
.....                          ;延时约 1ms
LDR     R0, =ADC_CON0          ;启动 A/D 转换
LDR     R1, =0X37
STR     R1, [R0]
.....

```

## 5.5.7 特殊功能寄存器

### 5.5.7.1 ADC参考控制寄存器（ADC\_VREFCON）

ADC 参考控制寄存器（ADC\_VREFCON）

偏移地址：40<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-4	—	—
CHOP_EN	bit3	R/W	<b>VREF CHOP 使能控制位</b> 0: 禁止 1: 使能（ADC 使用内部 VREFP 时，需使能 CHOP）
IREF_EN	bit2	R/W	<b>IREF 使能控制位</b> 0: 禁止 1: 使能（ADC 工作时需使能 IREF）
VREF_SEL	bit1	R/W	<b>内部 VREFP 电压选择控制位</b> 0: 2.048V



			1: 保留, 仅作测试使用 (禁止设置为 1)
VREF_EN	bit0	R/W	内部 VREFP 使能控制位 0: 禁止 1: 使能

注 1: ADC 正常工作时, 无论正向参考电压选择为 VDD, 内部 VREFP 或外部 AVREFP, 均必须开启 IREF\_EN, 否则会导致 ADC 工作异常。

注 2: 选择内部参考电压 VREF 2.048V 作为 ADC 正向参考电压时, 需先设置 VREF\_EN 位使能内部参考, 并设置 IREF\_EN 和 ADC\_CON0 寄存器的 EN 位使能 ADC, 然后等待 300us 以后, 再设置 CHOP\_EN 位使能参考电压斩波器, 否则内部参考电压可能不稳定。

### 5.5.7.2 ADC转换值寄存器 (ADC\_DR)

#### ADC 转换值寄存器 (ADC\_DR)

偏移地址: 00<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-12	—	—
DR<11:0>	bit11-0	R	A/D 转换结果 (对不同分辨率均高位对齐)

### 5.5.7.3 ADC控制寄存器 0 (ADC\_CON0)

#### ADC 控制寄存器 0 (ADC\_CON0)

偏移地址: 04<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-6	—	—
BIT_SEL<1:0>	bit5-4	R/W	A/D 分辨率选择位 00: 6 位分辨率, 结果存放在 ADC_DR<11:6> 01: 8 位分辨率, 结果存放在 ADC_DR<11:4> 10: 10 位分辨率, 结果存放在 ADC_DR<11:2> 11: 12 位分辨率, 结果存放在 ADC_DR<11:0>
—	bit3	—	—
ACP_EN	bit2	R/W	A/D 自动转换比较功能使能位 0: 关闭

			1: 启动
TRIG	bit1	R/W	<b>A/D 转换状态位</b> 0: A/D 未进行转换, 或 A/D 转换已完成 (硬件清零, 且硬件清零优先) 1: A/D 转换正在进行, 该位置 1 启动 A/D 转换 SMPS 为 0 时, TRIG 的写操作失效 (由硬件根据 SMPON 软件采样和 ADC 转换过程进行控制), 且不能将其读出作为 ADC 转换完成的标志
EN	bit0	R/W	<b>A/D 转换使能位 (ACP_EN 为 1 时, 该位无效)</b> 0: 禁止 1: 使能

注 1: TRIG 软件只能写 1, 且写 1 后硬件自动清零。

注 2: 关闭 SMPON 时, TRIG 与 ADC\_IF 寄存器的 IF 位均可作为转换完成标志。而开启 SMPON 时, 仅 ADC\_IF 寄存器的 IF 位可作为转换完成标志。建议无论 SMPON 是否开启, 均通过 ADC\_IF 寄存器的 IF 位来判断 ADC 是否转换完成。

注 3: 选择内部参考电压 VREF 2.048V 作为 ADC 正向参考电压时, 需先设置 ADC\_VREFCON 寄存器的 VREF\_EN 位使能内部参考, 并设置 IREF\_EN 和 ADC\_CON0 寄存器的 EN 位使能 ADC, 然后等待 300us 以后, 再设置 CHOP\_EN 位使能参考电压斩波器, 否则内部参考电压可能不稳定, 然后延时 1ms 以上, ADC 工作建立完成 (否则有可能导致 ADC 转换异常), 再启动 ADC 转换 (TRIG=1), 可得到正确的转换结果。

注 4: 因每次 IREF\_EN, VREF\_EN, CHOP\_EN, A/D 转换使能位 EN 重新使能后, 均需要执行上述 ADC 工作建立过程, 所以应用中, 在芯片正常运行时不建议关闭上述 4 个使能控制信号, 保持为 1, 只在进入深睡眠模式前, 可以关闭 ADC。

注 5: ADC\_CON0 寄存器不支持位操作写入, 必须按字方式进行写操作, 否则会使该寄存器设置错误, 导致 ADC 工作异常。

#### 5.5.7.4 ADC控制寄存器 1 (ADC\_CON1)

##### ADC 控制寄存器 1 (ADC\_CON1)

偏移地址: 08<sub>H</sub>

复位值: 00000000\_00000100\_00010000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CALZERO	保留		ST<4:0>				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	HSEN	SMPON	SMPS	保留	VREFN	VREFP<1:0>		保留			CLKS	CLKDIV<2:0>			

—	bit31-24	—	—
CALZERO	bit 23	R/W	<b>ADC 调零使能位</b> 0: 禁止 1: 保留, 仅作测试使用 (禁止设置为 1)
—	bit22-21	—	—
ST<4:0>	bit20-16	R/W	<b>A/D 采样时间选择位 (硬件控制有效)</b> 采样时间: ST*2+1 个 Tadcclk

—	bit15	—	—
HSEN	bit14	R/W	<b>AD 转换速度控制位</b> 0: 高速 1: 保留, 仅作测试使用 (禁止设置为 1)
SMPON	bit13	R/W	<b>A/D 采样软件控制位 (ACP_EN 为 1 时, 该位无效)</b> 0: 关闭 AD 采样 1: 启动 AD 采样
SMPS	bit12	R/W	<b>A/D 采样模式选择位 (ACP_EN 为 1 时, 固定为 1)</b> 0: 软件控制 1: 硬件控制
—	bit11	—	—
VREFN	bit10	R/W	<b>A/D 负向参考电压选择位</b> 0: 保留, 仅作测试使用 (禁止设置为 0) 1: 内部地电压 VSS
VREFP<1:0>	bit9-8	R/W	<b>A/D 正向参考电压选择位</b> 00: 选择芯片工作电压 VDD 01: 外部参考电压 AVREFP (该电压不能高于 VDD, 不能低于 1.0V) 1x: 选择内部参考电压 VREFP (2.048V)
—	bit7-4	—	—
CLKS	bit3	R/W	<b>A/D 时钟源选择位</b> 0: PCLK 1: LRC (32KHz)
CLKDIV<2:0>	bit2-0	R/W	<b>A/D 时钟源预分频选择位</b> 000 = 1:1 001 = 1:2 010 = 1:4 011 = 1:8 100 = 1:16 101 = 1:32 110 = 1:64 111 = 1:256

注 1: 选择内部参考电压 VREF 2.048V 作为 ADC 正向参考电压时, 需先设置 VREF\_EN 位使能内部参考, 并设置 IREF\_EN 和 ADC\_CON0 寄存器的 EN 位使能 ADC, 然后等待 300us 以后, 再设置 CHOP\_EN 位使能参考电压斩波器, 否则内部参考电压可能不稳定;

注 2: ADC 工作时, ADC\_CON1 寄存器的 HSEN 必须为 0, 设置为高速模式, VREFN 必须为 1, 负参考设置为内部地 VSS, 否则会导致 ADC 转换异常;

注 3: ADC 使用 VDD (VREFP=2'b00), 或外部参考电压 (VREFP=2'b01), 作为参考电压时, 最大转换时钟频率为 2MHz, 使用内部参考电压作为参考电压时 (VREFP=2'b10 或 2'b11), 最大转换时钟频率为 1MHz;

注 4: 当使用外部参考电压时, 参考电压不能低于 1.0V, 否则会导致 ADC 工作异常;

注 5：设置足够的采样时间有利于 ADC 采样稳定，从而提高 ADC 转换结果精确度，需根据应用系统中的模拟输入信号情况，通过寄存器位 ST<4:0>设置合适的采样时间。

### 5.5.7.5 ADC通道选择寄存器（ADC\_CHS）

#### ADC 通道选择寄存器（ADC\_CHS）

偏移地址：0C<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-5	—	—
CHS<4:0>	bit4-0	R/W	<b>A/D 模拟通道选择位</b> 00000: 通道 0 (AIN0) 00001: 通道 1 (AIN1) 00010: 通道 2 (AIN2) 00011: 通道 3 (AIN3) 00100: 通道 4 (AIN4) 00101: 通道 5 (AIN5) 00110: 通道 6 (AIN6) 00111: 通道 7 (AIN7) 01000: 通道 8 (AIN8) 01001: 通道 9 (AIN9) 01010: 通道 10 (AIN10) 01011: 通道 11 (AIN11) 01100: 通道 12 (AIN12) 01101: 通道 13 (AIN13) 01110: 通道 14 (AIN14) 01111: 通道 15 (AIN15) 10000: 通道 16 (AIN16) 10001: 通道 17 (AIN17) 10010: 通道 18 (1/4VDD) 10011: 通道 19 (AIN19) 其它: 通道关闭

### 5.5.7.6 ADC中断使能寄存器（ADC\_IE）

#### ADC 中断使能寄存器（ADC\_IE）

偏移地址：10<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												ACPOVIE	ACPMAXIE	ACPMINIE	IE

—	bit31-4	—	—
ACPOVIE	bit3	R/W	<b>ADC 自动转换溢出中断使能位</b> 0: 禁止 1: 使能
ACPMAXIE	bit2	R/W	<b>ADC 自动转换高阈值超出中断使能位</b> 0: 禁止 1: 使能
ACPMINIE	bit1	R/W	<b>ADC 自动转换低阈值超出中断使能位</b> 0: 禁止 1: 使能
IE	bit0	R/W	<b>ADC 中断使能位</b> 0: 禁止 1: 使能

### 5.5.7.7 ADC中断标志寄存器 (ADC\_IF)

ADC 中断标志寄存器 (ADC\_IF)

偏移地址: 14<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												ACPOVIF	ACPMAXIF	ACPMINIF	IF

—	bit31-4	—	—
ACPOVIF	bit3	R/W	<b>ADC 自动转换溢出中断标志位</b> 0: 自动转换的溢出时间未到 1: 自动转换的溢出时间已到 (由硬件置 1, 软件清除) 软件写 1 清除标志位, 写 0 无效
ACPMAXIF	bit2	R/W	<b>ADC 自动转换高阈值超出中断标志位</b> 0: 均值结果没有达到高阈值 1: 均值结果大于或等于高阈值 (由硬件置 1, 软件清除) 软件写 1 清除标志位, 写 0 无效
ACPMINIF	bit1	R/W	<b>ADC 自动转换低阈值超出中断标志位</b> 0: 均值结果没有到达最低阈值 1: 均值结果小于或等于低阈值 (由硬件置 1, 软件清除) 软件写 1 清除标志位, 写 0 无效

IF	bit0	R/W	<b>ADC 中断标志位</b> 0: 正在进行转换 1: A/D 转换完成（由硬件置 1，软件清除） 软件写 1 清除标志位，写 0 无效
----	------	-----	---------------------------------------------------------------------------------

注 1: ADC 中断禁止时，如果满足条件仍会置起对应的中断标志位，只是不会产生中断请求。

注 2: 对 ADC\_IF 寄存器中的各中断标志位，写 0 无效，写 1 才能清除标志位；读操作时，读取的值为 1 表示有中断发生。

### 5.5.7.8 ADC 自动转换比较控制寄存器 (ADC\_ACPC)

ADC 自动转换比较控制寄存器 (ADC\_ACPC)

偏移地址: 28<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											CLKS			TIMES<1:0>	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				OVFL_TIME<11:0>											

—	bit31-21	—	—
CLKS	bit20	R/W	<b>ACP 溢出计数的时钟源选择位</b> 0: FCLK/256 (PCLK 的 256 分频) 1: FLRC/256 (LRC 时钟的 256 分频)
—	bit19-18	—	—
TIMES<1:0>	bit17-16	R/W	自动转换比较次数选择位（基于由 <b>OVFL_TIME</b> 设定的时间段内） 00: 1 次 01: 2 次 10: 4 次 11: 8 次
—	bit15-12	—	—
OVFL_TIME<11:0>	bit11-0	R/W	每次自动转换比较的溢出时间，可配置范围 为 0~9C3 <sub>H</sub> ，计数时钟周期为 Tacp，分 别对应如下溢出时间： 0: 1 x Tacp 1: 2 x Tacp 2: 3 x Tacp ... 9C3 <sub>H</sub> : 2500 x Tacp Tacp 为由 CLKS 选择的 ACP 溢出计数的 时钟源周期

注：配置 OVFL\_TIME 的值必须大于一次 A/D 采样与转换时间。

### 5.5.7.9 ADC自动转换比较阈值寄存器 (ADC\_ACPCMP)

ADC 自动转换比较阈值寄存器 (ADC\_ACPCMP)

偏移地址: 30<sub>H</sub>复位值: 00001111\_11111111\_00000000\_00000000<sub>B</sub>

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

—	bit31-28	—	—
CMP_MAX<11:0>	bit 27-16	R/W	自动比较高阈值
—	bit15-12	—	—
CMP_MIN<11:0>	bit 11-0	R/W	自动比较低阈值

### 5.5.7.10 ADC自动转换均值数据寄存器 (ADC\_ACPMEAN)

ADC 自动转换均值数据寄存器 (ADC\_ACPMEAN)

偏移地址: 34<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-12	—	—
MEAN_DATA<11:0>	bit 11-0	R	自动转换结果的均值

### 5.5.7.11 ADC结果数据偏置寄存器 (ADC\_OFFDR)

ADC 结果数据偏置寄存器 (ADC\_OFFDR)

偏移地址: 38<sub>H</sub>复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-12	—	—
OFFDATA<11:0>	bit 11-0	W/R	转换结果偏置值

注: 本芯片暂不支持使用 ADC\_OFFDR 寄存器。

## 5.6 独立看门狗定时器 (IWDG)

### 5.6.1 概述

当配置字 `CFG_WORD0` 配置为“独立看门狗使能”时，即 `CFG_IWDTEN` 位配置为 1，软件不可配置 `IWDG_LOAD` 寄存器。上电复位后看门狗立即启动（IWDG 使用 32KHz LRC 时钟计数），IWDG 载入 `CFG_WORD0` 配置字中 `IWDTRL` 参数对应的计数初值的 1/4，并进行递减计数，计数到 0 时，窗口计数器加“1”，并在下一计数时钟到来时，计数器再次载入 `IWDTRL` 参数对应的计数初值的 1/4，并继续递减计数。当窗口计数器计数到 2 时（IWDG 累计计数为 `IWDTRL` 的一半），IWDG 产生中断标志。当窗口计数器计数到 4（IWDG 累计计数等于 `IWDTRL`）之前，没有在相应的喂狗窗口期进行喂狗动作，IWDG 模块将产生复位信号。软件只能读写 `IWDG_LOCK`、写 `IWDG_INTCLR`，其它 IWDG 相关特殊功能寄存器不再可写，只能读取，读 `IWDG_LOAD` 得到的值为“IWDTRL 参数对应的计数初值”，读 `IWDG_CON` 将返回“0x0000\_000F”。

当配置字 `CFG_WORD0` 配置为“独立看门狗禁止”时，即 `CFG_IWDTEN` 位配置为 0，上电看门狗不启动，但软件可配置看门狗使能，且软件可配置 `IWDG_LOAD` 寄存器。当配置为“软件看门狗使能”时，即软件配置 `IWDG_CON` 寄存器的 `EN` 为 1，IWDG 计数器载入 `IWDG_LOAD` 寄存器值的 1/4，开始递减计数，当计数到 0 时，窗口计数器加“1”，并在下一个计数时钟到来时，计数器再次载入 `IWDG_LOAD` 寄存器值的 1/4，并继续递减计数。当窗口计数器计数到 2 时（IWDG 累计计数为 `IWDG_LOAD` 的一半），IWDG 产生中断标志。当窗口计数器计数到 4（IWDG 累计计数等于 `IWDG_LOAD`）之前，没有在相应的喂狗窗口期进行喂狗动作，则 IWDG 模块将产生复位信号。配置 `CLKS`，可以选择计数时钟源；配置 `IWDG_LOAD` 寄存器，可设置计数初值；读取 `IWDG_VALUE`，可得到 IWDG 当前计数值。写入装载值寄存器 `IWDG_LOAD` 时，计数器当前值寄存器将被载入 `IWDG_LOAD` 寄存器值的 1/4（前提条件是看门狗要使能）。

IWDG 模块支持寄存器写保护，配置寄存器 `IWDG_LOCK=0x1ACCE551`，可去除 IWDG 寄存器的写保护状态，进行被保护寄存器的写操作，否则无法对被保护寄存器写入。

注：在 SWD 调试模式下，需要禁止 IWDG，否则在调试过程中，IWDG 会始终保持工作，可能会产生计数溢出复位，导致芯片调试异常。

### 5.6.2 特殊功能寄存器

#### 5.6.2.1 IWDG 计数器装载值寄存器 (IWDG\_LOAD)

IWDG 计数器装载值寄存器 (IWDG\_LOAD)

偏移地址：00H

复位值：11111111\_11111111\_11111111\_11111111<sub>B</sub>

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

LOAD<31:0>	bit31-0	W	IWDG 计数器重载值
------------	---------	---	-------------



			计数范围 0x0000_0001~0xFFFF_FFFF。如果为 0，IWDT 不计 数。
--	--	--	--------------------------------------------------

### 5.6.2.2 IWDT计数器当前值寄存器（IWDT\_VALUE）

#### IWDT 计数器当前值寄存器（IWDT\_VALUE）

偏移地址：04<sub>H</sub>

复位值：11111111\_11111111\_11111111\_11111111<sub>B</sub>

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

VALUE<31:0>	bit31-0	R	<b>IWDT 计数器当前值</b> 读取时返回 IWDT 计数器的当前计数值
-------------	---------	---	--------------------------------------------

### 5.6.2.3 IWDT控制寄存器（IWDT\_CON）

#### IWDT 控制寄存器（IWDT\_CON）

偏移地址：08<sub>H</sub>

复位值：00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-4	—	—
CLKS	bit3	R/W	<b>IWDT 计数时钟选择位</b> 0: PCLK 1: LRC 时钟（约 32KHz）
RSTEN	bit2	R/W	<b>IWDT 复位使能位</b> 0: 禁止 1: 使能，IWDT 计数到 0 时，产生复位信号，将芯片复位
IE	bit1	R/W	<b>IWDT 中断使能位</b> 0: 禁止 1: 使能，IWDT 计数到 0 时，产生中断标志
EN	bit0	R/W	<b>IWDT 模块使能位</b> 0: 禁止 1: 使能

注：IWDT\_CON 寄存器中的各个控制位，仅在配置字 CFG\_WORD0 的配置位 CFG\_IWDTEN=0 时才有效。

### 5.6.2.4 IWDt中断标志清除寄存器 (IWDt\_INTCLR)

#### IWDt 中断标志清除寄存器 (IWDt\_INTCLR)

偏移地址: 0C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

INTCLR<31:0>	bit31-0	W	<b>IWDt 中断标志清零位</b> 对 IWDt_INTCLR 寄存器进行任意写操作, IWDt 中断标志位均被清零, 计数器重载 IWDt_LOAD 寄存器值, 继续递减计数
--------------	---------	---	-----------------------------------------------------------------------------------------------

### 5.6.2.5 IWDt中断标志寄存器 (IWDt\_RIS)

#### IWDt 中断标志寄存器 (IWDt\_RIS)

偏移地址: 10<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-1	—	—
IWDtIF	bit0	R	<b>IWDt 中断标志位</b> 0: 未产生中断 1: IWDt 计数器计数到 0, 产生中断 写寄存器 IWDt_INTCLR, 可清除 IWDt 中断标志位

### 5.6.2.6 IWDt访问使能寄存器 (IWDt\_LOCK)

#### IWDt 访问使能寄存器 (IWDt\_LOCK)

偏移地址: 00<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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

—	bit31-1	W	对 IWDt_LOCK<31:0>写 0x1ACCE551 时, 位 LOCK 为 0; 写其它值时位 LOCK 为 1
---	---------	---	-----------------------------------------------------------------

LOCK	bit0	R/W	<b>IWDT 寄存器保护状态位</b> 0: IWDT 寄存器处于未保护状态 1: IWDT 寄存器处于保护状态 对 IWDT_LOCK 寄存器写入 0x1ACCE551，被保护的寄存器处于未保护状态；写入其它值，处于保护状态
------	------	-----	-----------------------------------------------------------------------------------------------------------------------------

注 1: IWDT\_LOCK 寄存器为只写 32 位寄存器，其中仅 LOCK 位可读，该寄存器必须使用字操作访问方式。

注 2: IWDT\_LOCK 保护的寄存器为 IWDT\_LOAD，IWDT\_CON，IWDT\_INTCLR。

## 5.7 窗口看门狗 (WWDT)

### 5.7.1 概述

- ◆ 支持设定喂狗禁止区
  - ◇ 通过寄存器位 `WWDTWIN` 设置喂狗禁止区
    - 寄存器位 `WWDTWIN` 设定为 11 时，任何时刻喂狗都不产生复位，也不产生中断
  - ◇ 在喂狗禁止区内喂狗，产生 `WWDT` 复位
  - ◇ 在喂狗禁止区外喂狗，产生 `WWDT` 中断
    - `WWDT` 中断可用作喂狗请求
- ◆ 安全可靠
  - ◇ 当配置字控制位 `CFG_WWDTEN` 为 1 时，一旦 `WWDT` 模块软件使能位 (`EN`) 使能，则只能通过复位关闭 `WWDT`
- ◆ `WWDT` 溢出长度可设定
  - ◇ 可通过 `WWDT_LOAD` 寄存器设定
  - ◇ 溢出时产生 `WWDT` 复位

对于 `WWDT` 看门狗，过早或过晚喂狗都将产生 `WWDT` 复位，可用于检测软件的过晚或过早喂狗行为，防止程序跑至不可控状态。例如发生中断异常，程序不断进入一个带喂狗指令的子程序的情况，可通过 `WWDT` 复位使程序执行恢复正常。

用户可根据程序正常执行的时间设定喂狗窗口，可检测程序未按正常次序执行，跳过某些程序段的异常情况。当选用 `PCLK` 作为 `WWDT` 时钟时，可精确地检测异常情况。

上电复位后，窗口看门狗不启动，需通过软件设置使能窗口看门狗。软件配置 `WWDT_CON` 寄存器的 `EN` 为 1 使能窗口看门狗后，`WWDT` 计数器载入 `WWDT_LOAD` 寄存器值的 1/4，开始递减计数，当计数到 0 时，窗口计数器加“1”，并在下一个计数时钟到来时，计数器再次载入 `WWDT_LOAD` 寄存器值的 1/4，并继续递减计数。若 `WWDTWIN` 设置为 25%，则窗口计数器为 1 时，`WWDT` 产生中断标志；若 `WWDTWIN` 设置为 50%，则窗口计数器为 2 时，`WWDT` 产生中断标志；若 `WWDTWIN` 设置为 75%，则窗口计数器为 3 时，`WWDT` 产生中断标志。`WWDT` 产生中断后，直至窗口计数器计数到 4（即 `WWDT` 累计计数等于 `WWDT_LOAD`）之前，没有在相应的喂狗窗口期进行喂狗动作，则 `WWDT` 模块将产生复位信号，如下图所示。配置 `WWDT_CON` 寄存器的 `CLKS`，可以选择计数时钟源；配置 `WWDT_LOAD` 寄存器，可设置计数初值；读取 `WWDT_VALUE`，可得到 `WWDT` 当前计数值。写入装载值寄存器 `WWDT_LOAD` 时，计数器当前值寄存器将被载入 `WWDT_LOAD` 寄存器值的 1/4（前提条件是看门狗要使能）。

若配置字 `CFG_WORD0` 的 `CFG_WWDTEN` 位配置为 1，则软件使能窗口看门狗之后，不可再通过软件关闭窗口看门狗，只能通过芯片复位关闭窗口看门狗。

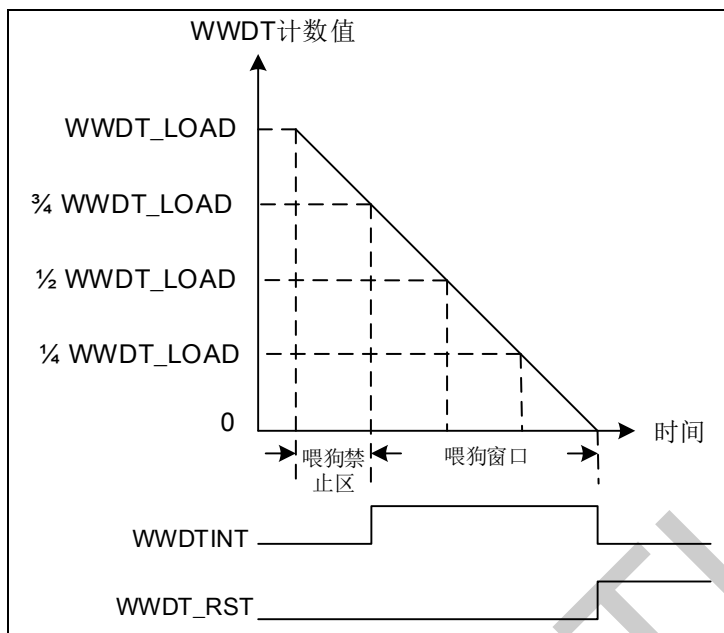


图 5-49 窗口看门狗中断和溢出复位产生时序图 (WWDTWIN 设定为 25%)

在喂狗窗口期写寄存器 `WWDT_INTCLR`, WWDT 将重新载入计数初值, 并重新进行递减计数。在喂狗窗口期外 (如时序图中的喂狗禁止区), 写寄存器 `WWDT_INTCLR` 会导致产生 WWDT 复位, 如下图所示。喂狗窗口期的定义参考寄存器 `WWDT_CON` 的 `WWDTWIN` 控制位。

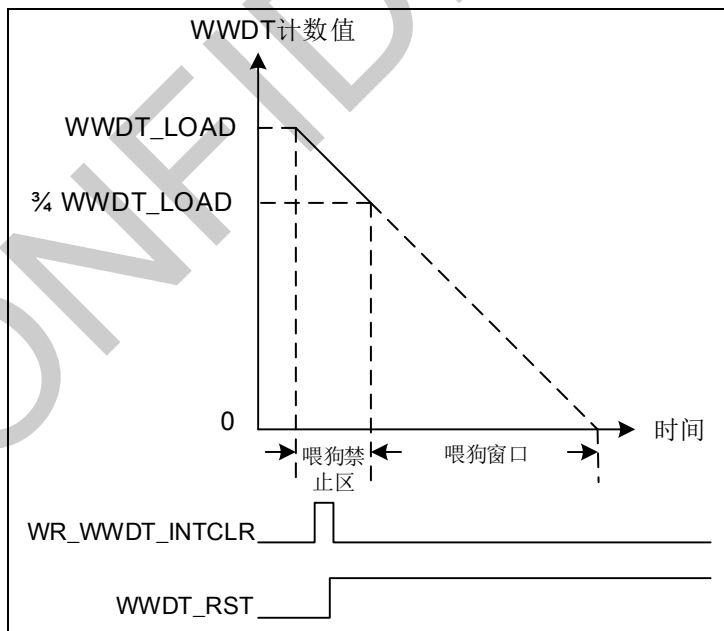


图 5-50 错误的喂狗时序图 (WWDTWIN 设定为 25%)

WWDT 模块支持寄存器写保护, 设置寄存器 `WWDT_LOCK=0x1ACCE551`, 可去除 WWDT 寄存器的写保护状态, 进行被保护寄存器的写操作, 否则无法对被保护寄存器写入。

## 5.7.2 特殊功能寄存器

### 5.7.2.1 WWDT计数器装载值寄存器 (WWDT\_LOAD)

WWDT 计数器装载值寄存器 (WWDT_LOAD)															
偏移地址: 00 <sub>H</sub>															
复位值: 00000000_00000010_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOAD<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD<15:0>															
LOAD<31:0>	bit31-0	W	<b>WWDT 计数器重载值</b> 计数范围 0x0000_0001~0xFFFF_FFFF。如果为 0，WWDT 不计数。												

### 5.7.2.2 WWDT计数器当前值寄存器 (WWDT\_VALUE)

WWDT 计数器当前值寄存器 (WWDT_VALUE)															
偏移地址: 04 <sub>H</sub>															
复位值: 00111111_11111111_11111111_11111111 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VALUE<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE<15:0>															
VALUE<31:0> >	bit31-0	R	<b>WWDT 计数器当前值</b> 读取时返回 WWDT 计数器的当前计数值，其中高两位为窗口计数器当前值												

### 5.7.2.3 WWDT控制寄存器 (WWDT\_CON)

WWDT 控制寄存器 (WWDT_CON)															
偏移地址: 08 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										WWDTWIN<1:0>		CLKS	RSTEN	IE	EN

—	bit31-6	—	—
WWDTWIN<1:0>	bit5-4	R/W	<b>WWDT 禁止喂狗窗口选择位</b> 00: 25%窗口内禁止喂狗，窗口内喂狗产生复位 01: 50%窗口内禁止喂狗，窗口内喂狗产生复位

			10: 75%窗口内禁止喂狗，窗口内喂狗产生复位 11: 不禁止喂狗，喂狗将使看门狗计数器重载
CLKS	bit3	R/W	<b>WWDT 计数时钟选择位</b> 0: PCLK 1: LRC 时钟（约 32KHz）
RSTEN	bit2	R/W	<b>WWDT 复位使能位</b> 0: 禁止 1: 使能，WWDT 计数到 0 时，产生复位信号，将芯片复位
IE	bit1	R/W	<b>WWDT 中断使能位</b> 0: 禁止 1: 使能，WWDT 计数到 0 时，产生中断标志
EN	bit0	R/W	<b>WWDT 模块使能位</b> 0: 禁止 1: 使能

注：如果 WWDT 使用 LRC 时钟计数，则程序不能在 3 个 LRC 时钟周期内连续两次或多次喂狗，否则可能会导致后续无法正常喂狗，推荐在 WWDT 中断服务程序中喂狗。

#### 5.7.2.4 WWDT中断标志清除寄存器（WWDT\_INTCLR）

WWDT 中断标志清除寄存器（WWDT_INTCLR）															
偏移地址：0C <sub>H</sub>															
复位值：00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTCLR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTCLR<15:0>															

INTCLR<31:0>	bit31-0	W	<b>WWDT 中断标志清 0 位</b> 对 WWDT_INTCLR 寄存器进行任意写操作，WWDT 中断标志位均被清零，计数器重载 WWDT_LOAD 寄存器值，继续递减计数
--------------	---------	---	----------------------------------------------------------------------------------------------

#### 5.7.2.5 WWDT中断标志寄存器（WWDT\_RIS）

WWDT 中断标志寄存器（WWDT_RIS）															
偏移地址：10 <sub>H</sub>															
复位值：00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														WWDTIF	

—	bit31-1	—	—
WWDTIF	bit0	R	<b>WWDT 中断标志位</b> 0: 未产生中断 1: WWDT 计数器计数到 0, 产生中断 写寄存器 WWDT_INTCLR, 可清除 WWDT 中断标志位

#### 5.7.2.6 WWDT 锁定寄存器 (WWDT\_LOCK)

WWDT 锁定寄存器 (WWDT_LOCK)																
偏移地址: 100 <sub>H</sub>																
复位值: 00000000_00000000_00000000_00000001 <sub>B</sub>																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留															LOCK	

—	bit31-1	W	对 WWDT_LOCK<31:0>写 0x1ACCE551 时, 位 LOCK 为 0; 写其它值时位 LOCK 为 1
LOCK	bit0	R	<b>WWDT 寄存器保护状态位</b> 0: WWDT 寄存器处于未保护状态 1: WWDT 寄存器处于保护状态 对 WWDT_LOCK 寄存器写入 0x1ACCE551, 被保护的寄存器处于未保护状态; 写入其它值, 处于保护状态

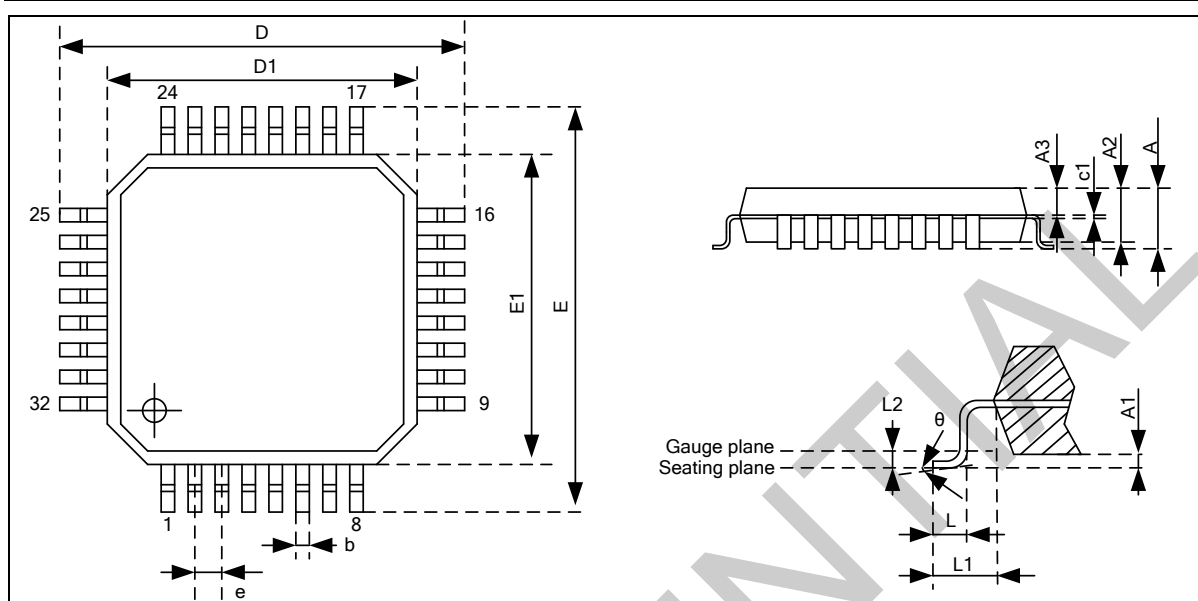
注 1: WWDT\_LOCK 寄存器为只写 32 位寄存器, 其中仅 LOCK 位可读, 该寄存器必须使用字操作访问方式。

注 2: WWDT\_LOCK 保护的寄存器为 WWDT\_LOAD, WWDT\_CON, WWDT\_INTCLR。



## 第6章 芯片封装外观尺寸图

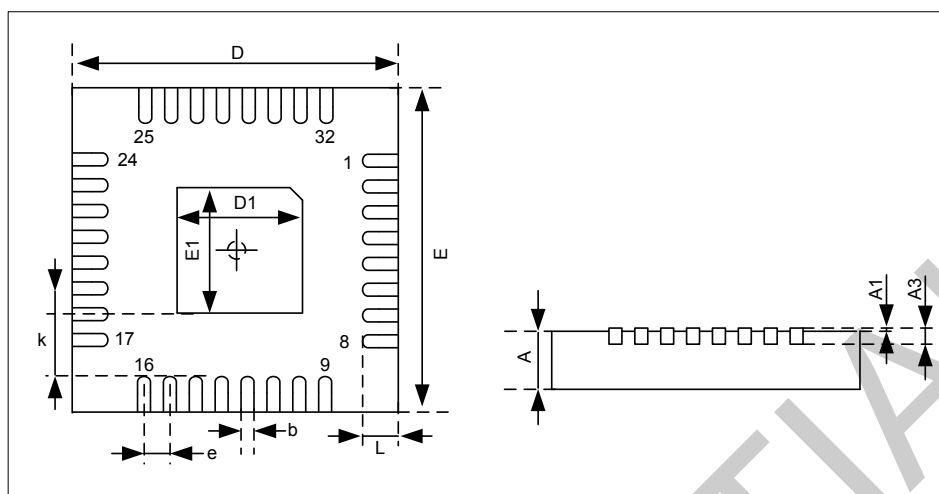
### 6.1 LQFP 32-pin封装外观尺寸图



标号	公制 (mm)		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
c1	0.13	—	0.18
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
b	0.32	—	0.43
e	—	0.80 BSC	—
L	0.40	0.60	0.75
L1	—	1.00 REF	—
$\theta$	0°	3.5°	7°

Note: Dimension "D" does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs shall not exceed 0.15mm per side.

## 6.2 QFN 32-pin封装外观尺寸图



标号	公制 (mm)		
	MIN	NOM	MAX
A	0.70	0.80	0.90
A1	0.00	—	0.05
A3	0.203REF.		
D	4.90	—	5.10
E	4.90	—	5.10
D1	3.30	—	3.95
E1	3.30	—	3.95
k	0.20MIN.		
b	0.18	—	0.30
e	0.50TYP.		
L	0.30	—	0.50

Note: Dimension "D" does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs shall not exceed 0.15mm per side.

## 附录1 Cortex-M0 内核描述

### 附录1.1 Cortex-M0 指令集

Cortex-M0 指令集，基本指令共 56 条，其中 50 条指令为 16 位，6 条指令为 32 位，并有多条指令可以进行助记符扩展，完成不同功能，如指令运行结果是否影响条件标志位等。

32 位指令是：BL，DSB，DMB，ISB，MRS 和 MSR。

指令中的符号说明：

- 1) 方括号< >，表示括号内的任一种格式的操作数，均可作为指令操作数。  
如<Rm | #imm>表示操作数可以是寄存器 Rm，或者立即数#imm。
- 2) 大括号{ }，表示括号内的操作数或符号可选。  
如：MOV{S}，表示指令助记符可以是 MOV 或 MOVS，区分该指令是否影响条件标志位。  
{Rd,}，表示指令中的目标操作数 Rd 可有可没有，没有时根据不同指令直接确定。

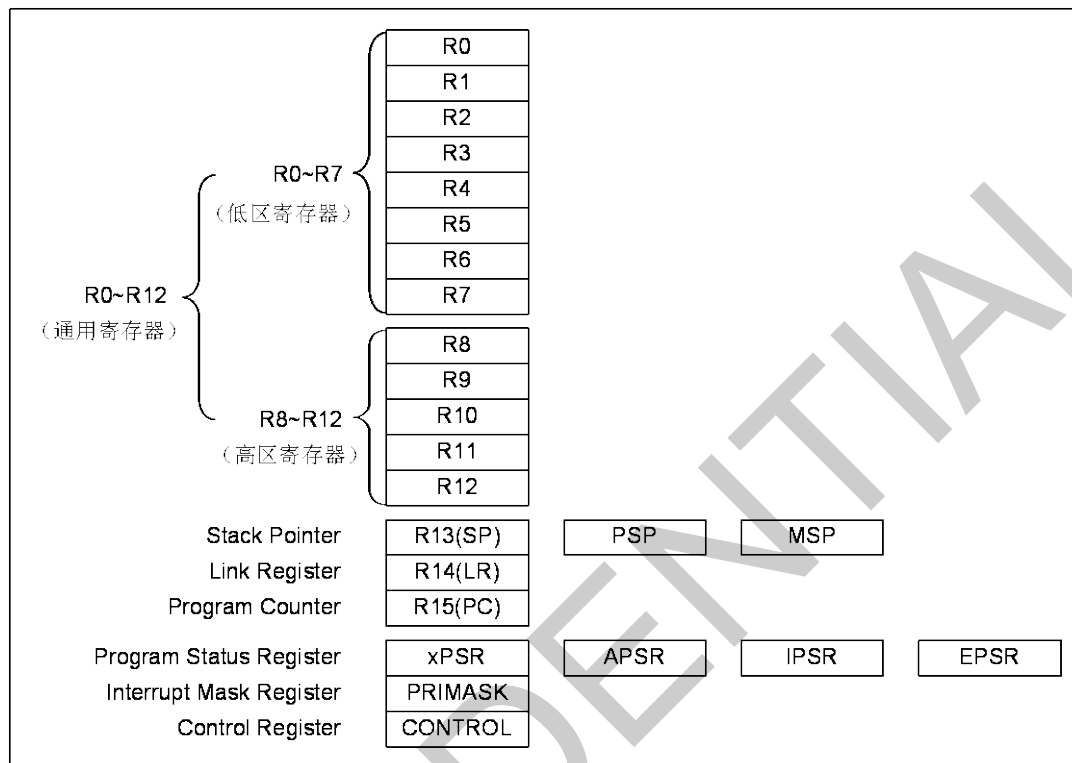
助记符	操作数	描述	影响标志位
ADR	Rd,Label	取 Label 地址到寄存器	-
LDR	Rt,Label	按字读 memory 到 Rt，立即数寻址	-
LDR	Rt,[Rn,<Rm   #imm>]	按字读 memory 到 Rt，带地址偏移寻址	-
LDRB	Rt,[Rn,<Rm   #imm>]	按字节读 memory 到 Rt，0 扩展为 32 位	-
LDRH	Rt,[Rn,<Rm   #imm>]	按半字读 memory 到 Rt，0 扩展为 32 位	-
LDRSB	Rt,[Rn,Rm]	按字节读 memory 到 Rt，符号位扩展	-
LDRSH	Rt,[Rn,Rm]	按半字读 memory 到 Rt，符号位扩展	-
LDM	Rn{!},reglist	批量读 memory 到 reglist，Rn 递增	-
STR	Rt,[Rn,<Rm   #imm>]	按字写 memory，带地址偏移寻址	-
STRB	Rt,[Rn,<Rm   #imm>]	按字节写 memory，0 扩展为 32 位	-
STRH	Rt,[Rn,<Rm   #imm>]	按半字写 memory，0 扩展为 32 位	-
STM	Rn!,reglist	批量写 memory，Rn 递增	-
PUSH	Reglist	寄存器压栈	-
POP	Reglist	寄存器出栈	-
MOV{S}	Rd, <Rm   #imm>	数据传送 Rd= <Rm   #imm>	N,Z 或-
MVNS	Rd,Rm	Rm 按位求反之后传送到 Rd	N,Z
MRS	Rd,spec_reg	读特殊功能寄存器，Rd=spec_reg	-
MSR	Spec_reg,Rm	写特殊功能寄存器，spec_reg=Rm	N,Z,C,V 或-
ADCS	{Rd,}Rn,Rm	带进位加法	N,Z,C,V
ADD{S}	{Rd,}Rn,<Rm   #imm>	加法	N,Z,C,V 或-
RSBS	{Rd,}Rn,#0	算术取反，Rd = 0-Rn	N,Z,C,V
SBCS	{Rd,}Rn,Rm	带借位减法，Rd = Rn-Rm-C	N,Z,C,V
SUB{S}	{Rt,}Rn,<Rm   #imm>	不带借位减法	N,Z,C,V 或-
ANDS	{Rd,}Rn,Rm	按位逻辑与，Rd = Rn&Rm	N,Z
ORRS	{Rd,}Rn,Rm	按位逻辑或，Rd = Rn Rm	N,Z

助记符	操作数	描述	影响标志位
EORS	{Rd,}Rn,Rm	按位逻辑异或, $Rd = Rn \wedge Rm$	N,Z
BICS	{Rd,}Rn,Rm	位清除, Rm 为掩码	N,Z
ASRS	{Rd,}Rn,<Rm   #imm>	算术右移	N,Z,C
LSLS	{Rd,}Rn,<Rm   #imm>	逻辑左移	N,Z,C
LSRS	{Rd,}Rn,<Rm   #imm>	逻辑右移	N,Z,C
RORS	{Rd,}Rn,Rm	循环右移	N,Z,C
CMP	{Rn,}<Rm   #imm>	比较	N,Z,C,V
CMN	Rn,Rm	负比较, 先将 Rm 取反, 再比较	N,Z,C,V
MULS	Rd,Rn,Rm	乘法, 结果为 32 位	N,Z
REV	Rd,Rm	按字节反转 (32 位大小端数据转换)	-
REV16	Rd,Rm	按半字反转 (2 个 16 位大小端数据转换)	-
REVSH	Rd,Rm	低半字反转, 按有符号数扩展为 32 位	-
SXTB	Rd,Rm	低字节, 按有符号数扩展到 32 位	-
SXTH	Rd,Rm	低半字, 按有符号数扩展到 32 位	-
UXTB	Rd,Rm	低字节, 零扩展到 32 位	-
UXTH	Rd,Rm	低半字, 零扩展到 32 位	-
TST	Rd,Rm	位测试	N,Z
B{cond}	Label	(条件) 分支短跳转到 Label 所指处	-
BL	Label	带链接的分支跳转, 跳转到 Label 所指处	-
BX	Rm	分支长跳转	-
BLX	Rm	带链接分支长跳转, 跳转到 Rm 所指处	-
CPSID	i	屏蔽中断响应, PRIMASK.PM=1	-
CPSIE	i	允许中断响应, PRIMASK.PM=0	-
SVC	#imm	管理调用, 产生 SVC 异常	-
DMB	-	数据存储器访问隔离	-
DSB	-	数据同步隔离	-
ISB	-	指令同步隔离	-
SEV	-	触发事件	-
WFE	-	等待事件	-
WFI	-	等待中断	-
BKPT	#imm	断点	-
NOP	-	空操作	-

注: 指令 CPSID 和 CPSIE, 分别用于禁止和允许中断请求, 指令操作码相同, 只是操作数不同, 实际为一条指令。

## 附录1.2 Cortex-M0 内核寄存器

Cortex-M0 内核寄存器如下图所示：



### 附录1.2.1 通用寄存器 R0~R12

R0~R12 为 32 位通用寄存器，用于数据操作。

### 附录1.2.2 堆栈指针寄存器 SP (R13)

Cortex-M0 内核有两个堆栈指针 MSP 和 PSP，但两者不能同时使用，具体使用的堆栈指针与进程模式有关。在线程模式下，配置 CONTROL 寄存器的 SPSEL 位，可选择当前使用的堆栈指针。编写指令时，两个堆栈指针均可通过 R13 或 SP 调用，访问当前正在使用的堆栈指针，也可通过 MRS/MSR 指令访问指定的堆栈指针。

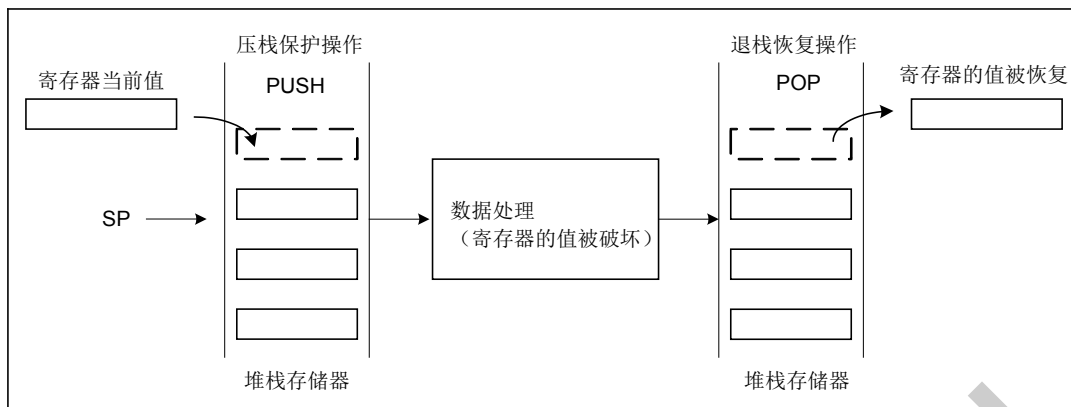
**主堆栈指针 (MSP)：**或写作 SP\_main，主要由操作系统内核，异常/中断服务程序，以及其它被授权访问的应用程序来使用，芯片复位后缺省使用主堆栈指针。

**进程堆栈指针 (PSP)：**或写作 SP\_process，在线程模式下，用户可选用进程堆栈指针；但在异常/中断服务程序中不能选用进程堆栈指针。

堆栈指针的最低两位始终是 0，即堆栈总是字（4 个字节）对齐的。

对应用程序，通常只需使用主堆栈指针 MSP，并且 PUSH 和 POP 指令也默认使用 MSP。

堆栈由一块地址连续的存储器空间，和一个栈顶指针组成，实现“先进后出”操作的缓冲区，常用于在异常/中断处理前后，保存和恢复一些关键寄存器的值。堆栈操作示意图如下：



### 附录1.2.3 链接寄存器 LR (R14)

链接寄存器 LR，也称为寄存器 R14，用于在调用子程序时存储返回地址。例如，当执行 BL 指令时，硬件电路会自动将下一条指令的地址保存到寄存器 LR。

### 附录1.2.4 程序计数器 PC (R15)

程序计数器 PC，也称为寄存器 R15。Cortex-M0 内核使用了指令流水线，所以读 PC 时，得到的值是当前指令的地址加 4。

如果对 PC 进行写操作，会产生程序跳转（但不更新 LR 寄存器），新写入的值即为程序跳转目的地址。Cortex-M0 中的指令至少是半字对齐的，所以 PC 的 LSB 位始终读取为 0。但无论是直接写 PC 还是使用跳转指令，都必须保证加载到 PC 的值 LSB 位为 1，用于表示这是在 Thumb 模式下执行指令，否则会被视为企图转入 ARM 模式，Cortex-M0 内核将产生一个 Fault 异常。

### 附录1.2.5 程序状态寄存器 xPSR

程序状态寄存器 xPSR，根据其各个状态位的功能，又划分为三个子状态寄存器：应用程序状态寄存器 APSR，中断服务程序状态寄存器 IPSR，执行程序状态寄存器 EPSR。

通过 MRS/MSR 指令，可对 3 个子状态寄存器进行单独访问，也可以同时访问其中 2 个或 3 个子状态寄存器。寄存器名称 IAPSR 表示同时访问 IPSR 和 APSR；寄存器名称 EAPSR 表示同时访问 EPSR 和 APSR；寄存器名称 IEPSR 表示同时访问 IPSR 和 EPSR；寄存器名称 XPSR 表示同时访问 3 个子状态寄存器。

程序状态寄存器 xPSR 及其 3 个子状态寄存器的状态位划分如下表所示：

	31	30	29	28	27:25	24	23:6	5:0
xPSR	N	Z	C	V	Reserved	T	Reserved	Exception Number
APSR	N	Z	C	V	Reserved			
IPSR	Reserved							Exception Number
EPSR	Reserved					T	Reserved	

应用程序状态寄存器 APSR 中的各个状态位，用于说明指令执行结果，各状态位描述如下：

**N:** 负数标志。指令执行结果为负数时，标志位 **N=1**，否则 **N=0**。

**Z:** 零标志。指令执行结果为零时，标志位 **Z=1**，否则 **Z=0**。对比较指令，如果被比较的两个数相等，则 **Z=1**。

**C:** 进位或借位标志。

对加法指令，如果执行结果有进位（结果 $\geq 232$ ），则 **C=1**，否则 **C=0**；  
对减法指令，如果执行结果无借位（结果 $\geq 0$ ），则 **C=1**，否则 **C=0**；  
对移位循环指令，取决于移位到 **C** 标志的数据位。

**V:** 溢出标志。

两个负数相加，结果为正数（**bit<31>=0**）时溢出，则 **V=1**，否则 **V=0**；  
两个正数相加，结果为负数（**bit<31>=1**）时溢出，则 **V=1**，否则 **V=0**；  
负数减去正数，结果为正数（**bit<31>=0**）时溢出，则 **V=1**，否则 **V=0**；  
正数减去负数，结果为负数（**bit<31>=1**）时溢出，则 **V=1**，否则 **V=0**。

中断服务程序状态寄存器 **IPSR** 中的状态位，用于表示正在处理的异常/中断号，目前正在执行异常/中断服务程序；如果 **IPSR<5:0>=0**，则表示目前是线程模式，未进行异常/中断处理。

执行程序状态寄存器 **ESPR** 中的 **T** 状态位，用于表示处理器是否处于 **Thumb** 模式。由于 **Cortex-M0** 处理器只支持 **Thumb** 模式，**T** 状态位应该始终为 **1**，如果将 **T** 位写为 **0**，会产生 **HardFault** 异常。使用 **MRS** 指令读取寄存器 **EPSR** 时，返回值为 **0**；如果使用 **MSR** 指令向 **EPSR** 写数据，写操作会被忽略。

附录1. 2. 6 异常/中断屏蔽寄存器 **PRIMASK**

异常/中断屏蔽寄存器 **PRIMASK**，可用于处理器屏蔽所有的异常/中断处理。

	31:1	0
PRIMASK	Reserved	PM

当屏蔽位 **PM=1** 时，禁止处理器响应所有可屏蔽异常/中断，不可屏蔽中断 **NMI** 除外。当 **PM=0** 时，不影响处理器对异常/中断的正常响应。

可以使用 **MRS** 和 **MSR** 指令访问 **PRIMASK** 寄存器，还可以使用专用的 **CPSID** 和 **CPSIE** 指令来设置寄存器中的 **PM** 位。

附录1. 2. 7 控制寄存器 **CONTROL**

控制寄存器 **CONTROL**，可用于在线程模式下，选择所使用的堆栈指针。

	31:2	1	0
CONTROL	Reserved	SPSEL	Reserved

当堆栈指针选择位 **SPSEL=0** 时，选择 **MSP** (**SP\_main**) 作为当前堆栈指针；当 **SPSEL=1** 时，选择 **PSP** (**SP\_process**) 作为当前堆栈指针。

在异常/中断处理模式下，总是使用 **MSP** 作为堆栈指针，**SPSEL=0**，且只读，不可写；处

处理器硬件电路会在异常/中断处理程序入口和返回时，对 **SPSEL** 位进行更新，确保进入异常/中断处理程序后，使用 **MSP** 作为堆栈指针，并在返回时恢复线程模式下的选择。在线程模式下，可配置 **SPSEL**，选择当前使用的堆栈指针。

通过 **MRS/MSR** 指令可访问两个的堆栈指针。在修改 **SPSEL** 位的指令后，需立即执行 **ISB**（指令同步隔离）指令，确保在 **SPSEL** 位修改完成，新的堆栈指针生效后，才会执行后续其它指令。

CONFIDENTIAL



## 附录2 电气特性

### 附录2.1 参数特性表

#### 附录2.1.1 芯片工作条件

##### ◆ 最大标称值

参数	符号	条件	标称值	单位
电源电压	VDD	VSS=0V	-0.3 ~ 7.5	V
管脚输入电压	V <sub>IN</sub>	VSS=0V	-0.3 ~ VDD + 0.3	V
管脚输出电压	V <sub>OUT</sub>	VSS=0V	-0.3 ~ VDD + 0.3	V
VDD 管脚最大输入电流	I <sub>MAXVDD</sub>	VDD=5.0V, 25℃	100	mA
VSS 管脚最大输出电流	I <sub>MAXVSS</sub>	VDD=5.0V, 25℃	120	mA
芯片存储温度	T <sub>STG</sub>	—	-55 ~ 125	℃
芯片最高结温	T <sub>JUNC</sub>	—	125	℃

注 1: 上述最大标称值参数为芯片工作条件的极限参数范围, 超出该范围, 可能会导致芯片永久性物理损坏;

注 2: 芯片需在正常工作条件下, 才能保证持续稳定运行, 对芯片的正常工作条件, 参见下面的表格所述。

##### ◆ 芯片工作条件表

参数	符号	工作条件	最小值	最大值	单位
芯片工作温度	T <sub>OPR</sub>	—	-40	85	℃
芯片结温	T <sub>JUNC</sub>	—	-40	105	℃
芯片工作电压	VDD	—	2.2	5.5	V
AHB 总线频率	F <sub>HCLK</sub>	—	0	48	MHz
APB 总线频率	F <sub>PCLK</sub>	—	0	48	MHz

##### ◆ 芯片功能模块工作电压范围表

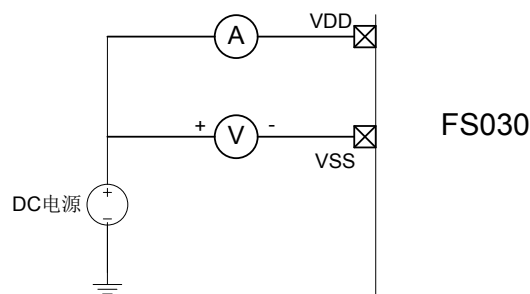
参数	符号	工作温度	VDD 电压	备注
ADC 工作电压	V <sub>ADC1</sub>	-40 ~ 85℃	2.5~5.5V	参考电压为 VDD 或外部 AVREFP, 当 VDD 电压低于 3.3V 时, 采样时间需大于 15Tadclk, 工作时钟速率需小于 100KHz
	V <sub>ADC2</sub>	-40 ~ 85℃	2.8~5.5V	参考电压为内部 VREFP, 当 VDD 电压低于 3.3V 时, 采样时间需大于 15Tadclk, 工作时钟速率需小于 50KHz

##### ◆ 芯片上电和下电工作条件表 (-40 ~ 85℃)

参数	符号	最小值	最大值	单位
VDD 上电初始电压	V <sub>start</sub>	0	0.2	V
VDD 上升速率	T <sub>VDD</sub>	10	—	us/V
VDD 下降速率		20	—	us/V

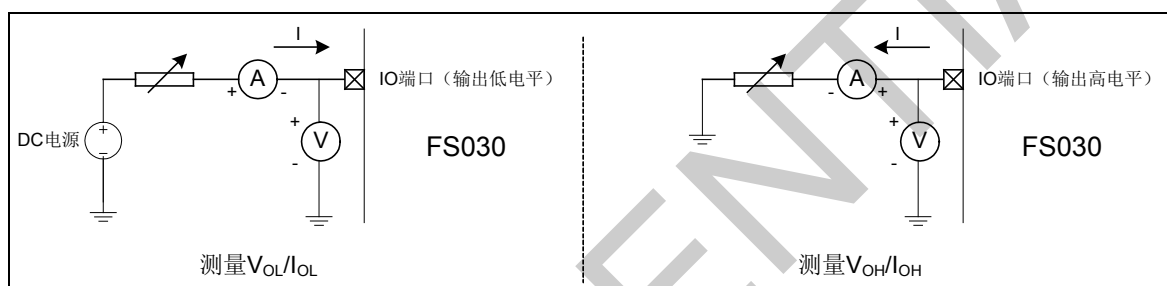
## 附录2.1.2 芯片特性参数测量方法

## ◆ 芯片功耗参数测量方法



芯片功耗测量连接示意图

## ◆ 芯片 IO 端口参数测量方法



芯片 IO 端口输出特性参数测量连接示意图

## 附录2.1.3 芯片功耗特性

## ◆ 芯片功耗特性参数表

参数	符号	最小值	典型值	最大值	单位	工作条件
芯片供电电压	VDD	2.2	—	5.5	V	-40℃ ~ 85℃
芯片静态电流	$I_{DD}$	—	2	—	mA	25℃, 上电复位, VDD = 5V, 所有的 I/O 端口输入低电平, MRSTN=0。
深度睡眠模式下芯片电流	$I_{PD1}$	—	2.5	—	μA	25℃, VDD = 5V, IWDTC 不使能, 所有 I/O 端口输出固定电平, 无负载。
浅睡眠模式下芯片电流	$I_{PD2}$	—	1.2	—	mA	25℃, VDD = 5V, IWDTC 不使能, 所有 I/O 端口输出固定电平, 无负载, 外设时钟关闭; 系统主时钟为内部 16MHz RC 时钟。
	$I_{PD3}$	—	1.5	—	mA	25℃, VDD = 5V, IWDTC 不使能, 所有 I/O 端口输出固定电平, 无负载, 外设时钟关闭; 系统主时钟为内部 48MHz RC 时钟。

参数	符号	最小值	典型值	最大值	单位	工作条件
正常运行模式 芯片电流	$I_{OP1}$	—	1.2	—	mA	25℃, VDD = 5V, IWDTC 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, ADC 使用内部 VREFP 作为参考电压; 系统主时钟为内部 2MHz HRC 时钟。
正常运行模式 芯片电流	$I_{OP2}$	—	2.8	—	mA	25℃, VDD = 5V, IWDTC 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, ADC 使用内部 VREFP 作为参考电压; 系统主时钟为内部 16MHz HRC 时钟。
正常运行模式 芯片电流	$I_{OP3}$	—	4.5	—	mA	25℃, VDD = 5V, IWDTC 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, ADC 使用内部 VREFP 作为参考电压; 系统主时钟为内部 32MHz HRC 时钟。
正常运行模式 芯片电流	$I_{OP4}$	—	6.0	—	mA	25℃, VDD = 5V, IWDTC 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, ADC 使用内部 VREFP 作为参考电压; 系统主时钟为内部 48MHz HRC 时钟。

◆ 芯片功能模块功耗特性参数表

参数	符号	最小值	典型值	最大值	单位	工作条件
外部振荡器 XTAL 16MHz 电流	$I_{XTAL}$	—	1	—	mA	25℃, VDD = 5V
内部高速时钟 HRC 48MHz 电流	$I_{HRC1}$	—	1	—	mA	25℃, VDD = 5V
内部高速时钟 HRC 32MHz 电流	$I_{HRC2}$	—	0.7	—	mA	25℃, VDD = 5V
内部高速时钟 HRC 16MHz 电流	$I_{HRC3}$	—	0.6	—	mA	25℃, VDD = 5V
内部高速时钟 HRC 2MHz 电流	$I_{HRC4}$	—	0.5	—	mA	25℃, VDD = 5V
LVD 模块电流	$I_{LVD}$	—	0.3	—	μA	25℃, VDD = 5V
ADC 模块电流	$I_{ADC1}$	—	0.9	—	mA	25℃, VDD = 5V, ADC 转换时钟频率为 1MHz, 内部

参数	符号	最小值	典型值	最大值	单位	工作条件
						VREFP 作为正向参考电压 (VREF_EN=1, CHOP_EN=1)
	I <sub>ADC2</sub>	—	0.5	—	mA	25℃, VDD = 5V, ADC 转换时钟频率为 2MHz, VDD 作为正向参考电压
IWDT 模块电流	I <sub>IWDT</sub>	—	0.3	—	μA	25℃, VDD = 5V, 计数时钟为 LRC
WWDT 模块电流	I <sub>WWDT</sub>	—	0.3	—	μA	25℃, VDD = 5V, 计数时钟为 LRC
T16N 模块电流	I <sub>T16N</sub>	—	0.3	—	mA	25℃, VDD = 5V, PWM 模式, 输出 200KHz 互补波形
T32N 模块电流	I <sub>T32N</sub>	—	1	—	mA	25℃, VDD = 5V, PWM 模式, 输出 200KHz 互补波形
UART 模块电流	I <sub>UART</sub>	—	0.7	—	mA	25℃, VDD = 5V, 通讯波特率为 115200bps
SPI 模块电流	I <sub>SPI</sub>	—	1	—	mA	25℃, VDD = 5V, 主控模式, 通讯速率为 2MHz
I2C 模块电流	I <sub>I2C</sub>	—	0.9	—	mA	25℃, VDD = 5V, 主控模式, 通讯速率为 400KHz
FLASH 编程电流	I <sub>PROG</sub>	—	—	4	mA	-40~85℃, VDD = 5V
FLASH 擦除电流	I <sub>ERAS</sub>	—	—	4	mA	

## 附录2.1.4 芯片 IO 端口特性

### ◆ 芯片输入端口特性表

芯片工作温度范围: -40℃ ~ 85℃						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输入高电平 (TTL 输入)	V <sub>IH1</sub>	0.6VDD	—	VDD	V	2.2V ≤ VDD ≤ 5.5V
I/O 端口输入低电平 (TTL 输入)	V <sub>IL1</sub>	VSS	—	0.1VDD	V	
IO 端口施密特迟滞窗口 (TTL 输入)	V <sub>HYST1</sub>	—	0.3	—	V	25℃, VDD=5.0V
I/O 端口输入高电平 (CMOS 输入)	V <sub>IH2</sub>	0.8VDD	—	VDD	V	2.2V ≤ VDD ≤ 5.5V

I/O 端口输入低电平 (CMOS 输入)	$V_{IL2}$	VSS	—	0.2VDD	V	
外部复位输入高电平 (CMOS 输入)	$V_{IH\_MRST}$	0.8VDD	—	VDD	V	$2.2V \leq VDD \leq 5.5V$
外部复位输入低电平 (CMOS 输入)	$V_{IL\_MRST}$	VSS	—	0.2VDD	V	$2.2V \leq VDD \leq 5.5V$
IO 端口施密特迟滞窗口 (CMOS 输入)	$V_{HYST2}$	—	0.7	—	V	25°C, VDD=5.0V
I/O 端口输入漏电流	$I_{IL}$	—	—	$\pm 1$	$\mu A$	$2.2V \leq VDD \leq 5.5V$ $V_{pin}=VSS$ 或 VDD
主复位端口漏电流		—	—	5	$\mu A$	$V_{pin}=VSS$ 或 VDD
I/O 端口弱上拉电阻	$R_{WPU}$	—	50	—	$k\Omega$	25°C, VDD=5.0V $V_{pin} = VSS$
I/O 端口弱下拉电阻	$R_{WPD}$	—	50	—	$k\Omega$	25°C, VDD=5.0V $V_{pin} = VDD$
I/O 输入端口 VDD/2 输出精度	$V_{VDD/2}$	—	$\pm 3\%$	—		25°C, VDD=5V, 弱上拉和弱下拉同时使能

注：I/O 端口弱上拉和弱下拉电阻，在全温度范围内（-40~85°C），相对于常温的变化量在 $\pm 10\%$ 以内。

◆ 芯片输出端口特性表

芯片工作温度范围：-40°C ~ 85°C						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输出高电平	$V_{OH}$	0.8VDD	—	VDD	V	$2.2V \leq VDD \leq 5.5V$ , 端口无负载
I/O 端口输出低电平	$V_{OL}$	VSS	—	0.2VDD	V	
I/O 端口灌电流 (PB8、PB9)	$I_{OL1}$	—	12	—	mA	25°C, VDD = 5V $V_{OL} = 0.6V$ , 普通驱动
	$I_{OL20}$	—	34	—		25°C, VDD = 5V $V_{OL} = 0.6V$ , 增强驱动
	$I_{OL3}$	—	60	—		25°C, VDD = 5V $V_{OL} = 1.4V$ , 增强驱动
I/O 端口灌电流 (PB8、PB9 除外)	$I_{OL1}$	—	12	—	mA	25°C, VDD = 5V $V_{OL} = 0.6V$ , 普通驱动
	$I_{OL21}$	—	22	—		25°C, VDD = 5V $V_{OL} = 0.6V$ , 增强驱动
I/O 端口拉电流	$I_{OH1}$	—	7	—	mA	25°C, VDD = 5V $V_{OH} = 4.4V$ , 普通驱动
	$I_{OH2}$	—	14	—		25°C, VDD = 5V $V_{OH} = 4.4V$ , 增强驱动

注：I/O 端口拉电流  $I_{OH}$  和灌电流  $I_{OL}$  的驱动能力随温度降低而增大。

## 附录2.1.5 芯片系统时钟特性

## ◆ 系统时钟规格表

参数	符号	最小值	典型值	最大值	单位	测试条件
系统时钟频率	$F_{OSC}$	—	—	48M	Hz	$2.2V \leq VDD \leq 5.5V$
系统时钟周期	$T_{OSC}$	20.8	—	—	ns	
机器周期	$T_{inst}$	—	$T_{OSC} * 1$	—	ns	
外部时钟高电平和低电平时间	$T_{OSL}, T_{OSH}$	20	—	—	ns	
外部时钟边沿上升和下降时间	$T_{OSR}, T_{OSF}$	—	—	8	ns	

## 附录2.1.6 芯片 ESD 特性

## ◆ 静电放电特性参数表

参数	符号	等级	典型值	单位	测试条件
ESD 电压（人体模型）	$V_{ESDHBM}$	3A	6000	V	25℃，遵循标准 MIL-STD-883H
ESD 电压（机器模型）	$V_{ESDMM}$	2	300	V	25℃，遵循标准 JESD22-A115
ESD 电压（充电器件模型）	$V_{ESDCDM}$	C3	1000	V	25℃，遵循标准 JEDEC JS-002

注：上述静电放电特性参数值是基于理论设计值和被测样品的测试值，不是批量产品测试值，仅供芯片应用时参考。

## ◆ 静态闩锁特性参数表

参数	符号	等级	测试条件
Latchup 电流	$I_{LU}$	IA	25℃，遵循标准 JESD78

注：上述静态闩锁特性参数值是基于理论设计值和被测样品的测试值，不是批量产品测试值，仅供芯片应用时参考。

## 附录2.1.7 芯片功能模块特性

## ◆ ADC 模块特性表

参数名称	符号	最小值	典型值	最大值	单位	测试条件
分辨率	RES	—	—	12	bit	见备注 1
参考电压范围	$V_{ADVREF}$	1.0	—	VDD	V	
模拟电压输入范围	$V_{IN}$	VSS	—	$V_{ADVREF}$	V	
输入电容	$C_{IN}$	—	40	—	pF	
模拟通道推荐输入电阻	$R_{IN}$	—	—	2k	$\Omega$	
AD 转换时钟周期	$T_{AD1}$	1	—	—	$\mu s$	
	$T_{AD2}$	0.5	—	—	$\mu s$	
AD 转换时间	$T_{CONV}$	—	14	—	$T_{AD}$	不包括采样时间

差分线性度	DNL	—	$\pm 1$	$\pm 2$	LSB	ADC 时钟频率为 1MHz
失调误差	$V_{\text{OFFSET}}$	—	2.5	4	mV	

注 1:  $T_{\text{AD1}}$  为 ADC 使用内部 VREFP 作为参考电压时的 AD 转换时钟周期;

$T_{\text{AD2}}$  为 ADC 使用 VDD 电压或外部 AVREFP 作为参考电压时的 AD 转换时钟周期。

◆ ADC 转换时钟源选择表

A/D 时钟源 选择	系统时钟工作频率 (Hz) (VREFP=2'b10 或 2'b11, 使用内部 VREFP 作为正向参考电压)			
	48M	32M	16M	2M
FPCLK	不推荐使用	不推荐使用	不推荐使用	不推荐使用
FPCLK /2	不推荐使用	不推荐使用	不推荐使用	$T_{\text{ADCLK}} = 1\mu\text{s}$
FPCLK /4	不推荐使用	不推荐使用	不推荐使用	$T_{\text{ADCLK}} = 2\mu\text{s}$
FPCLK /8	不推荐使用	不推荐使用	不推荐使用	$T_{\text{ADCLK}} = 4\mu\text{s}$
FPCLK /16	不推荐使用	不推荐使用	$T_{\text{ADCLK}} = 1\mu\text{s}$	$T_{\text{ADCLK}} = 8\mu\text{s}$
FPCLK /32	$T_{\text{ADCLK}} = 0.67\mu\text{s}$	$T_{\text{ADCLK}} = 1\mu\text{s}$	$T_{\text{ADCLK}} = 2\mu\text{s}$	$T_{\text{ADCLK}} = 16\mu\text{s}$
FPCLK /64	$T_{\text{ADCLK}} = 1.3\mu\text{s}$	$T_{\text{ADCLK}} = 2\mu\text{s}$	$T_{\text{ADCLK}} = 4\mu\text{s}$	$T_{\text{ADCLK}} = 32\mu\text{s}$
FPCLK /256	$T_{\text{ADCLK}} = 5.3\mu\text{s}$	$T_{\text{ADCLK}} = 8\mu\text{s}$	$T_{\text{ADCLK}} = 16\mu\text{s}$	$T_{\text{ADCLK}} = 128\mu\text{s}$
FLRC	$T_{\text{ADCLK}} = 31\mu\text{s}$	$T_{\text{ADCLK}} = 31\mu\text{s}$	$T_{\text{ADCLK}} = 31\mu\text{s}$	$T_{\text{ADCLK}} = 31\mu\text{s}$

A/D 时钟源 选择	系统时钟工作频率 (Hz) (VREFP=2'b00 或 2'b01, 使用 VDD 或外部 AVREFP 作为正向参考电压)			
	48M	32M	16M	2M
FPCLK	不推荐使用	不推荐使用	不推荐使用	$T_{\text{ADCLK}} = 0.5\mu\text{s}$
FPCLK /2	不推荐使用	不推荐使用	不推荐使用	$T_{\text{ADCLK}} = 1\mu\text{s}$
FPCLK /4	不推荐使用	不推荐使用	不推荐使用	$T_{\text{ADCLK}} = 2\mu\text{s}$
FPCLK /8	不推荐使用	不推荐使用	$T_{\text{ADCLK}} = 0.5\mu\text{s}$	$T_{\text{ADCLK}} = 4\mu\text{s}$
FPCLK /16	不推荐使用	$T_{\text{ADCLK}} = 0.5\mu\text{s}$	$T_{\text{ADCLK}} = 1\mu\text{s}$	$T_{\text{ADCLK}} = 8\mu\text{s}$
FPCLK /32	$T_{\text{ADCLK}} = 0.67\mu\text{s}$	$T_{\text{ADCLK}} = 1\mu\text{s}$	$T_{\text{ADCLK}} = 2\mu\text{s}$	$T_{\text{ADCLK}} = 16\mu\text{s}$
FPCLK /64	$T_{\text{ADCLK}} = 1.3\mu\text{s}$	$T_{\text{ADCLK}} = 2\mu\text{s}$	$T_{\text{ADCLK}} = 4\mu\text{s}$	$T_{\text{ADCLK}} = 32\mu\text{s}$
FPCLK /256	$T_{\text{ADCLK}} = 5.3\mu\text{s}$	$T_{\text{ADCLK}} = 8\mu\text{s}$	$T_{\text{ADCLK}} = 16\mu\text{s}$	$T_{\text{ADCLK}} = 128\mu\text{s}$
FLRC	$T_{\text{ADCLK}} = 31\mu\text{s}$	$T_{\text{ADCLK}} = 31\mu\text{s}$	$T_{\text{ADCLK}} = 31\mu\text{s}$	$T_{\text{ADCLK}} = 31\mu\text{s}$

◆ ADC 内部参考和 1/4VDD 电源分压电压特性表

参数	符号	最小值	典型值	最大值	单位	测试条件
ADC 内部参考电压	$V_{REF}$	2.028	2.048	2.068	V	25°C, VDD=5V, CHOP_EN=1
1/4VDD 电源分压	$V_{1/4VDD}$	1.23	1.25	1.27	V	25°C, VDD=5.0V
		1.225	1.25	1.275	V	-40°C~85°C, VDD=5.0V

注：ADC 内部参考电压在全温度范围内（-40°C~85°C）相对于常温条件下的偏差范围约±1%。

#### ◆ 内部时钟源特性表

参数	符号	最小值	典型值	最大值	单位	测试条件
HRC 时钟频率	$F_{HRC1}$	47.52	48	48.48	MHz	25°C, VDD=2.2V~5.5V
	$F_{HRC2}$	31.68	32	32.32	MHz	
	$F_{HRC3}$	15.84	16	16.16	MHz	
	$F_{HRC4}$	1.98	2	2.02	MHz	
	$F_{HRC1}$	47.18	48	48.96	MHz	-20°C~85°C, VDD=2.2V~5.5V
	$F_{HRC2}$	31.36	32	32.64	MHz	
	$F_{HRC3}$	15.68	16	16.32	MHz	
	$F_{HRC4}$	1.954	2	2.046	MHz	
	$F_{HRC1}$	47.04	48	48.96	MHz	-40°C~85°C, VDD=2.2V~5.5V
	$F_{HRC2}$	31.2	32	32.64	MHz	
	$F_{HRC3}$	15.6	16	16.32	MHz	
	$F_{HRC4}$	1.944	2	2.046	MHz	
HRC 起振时间	$T_{HRC}$	—	10	—	us	-40°C~85°C, VDD=2.2V~5.5V
LRC 时钟频率	$F_{LRC}$	31	32	33	KHz	25°C, VDD=2.2V~5.5V
		30	32	34	KHz	-40°C~85°C, VDD=2.2V~5.5V

注：在全温度范围内，如果应用系统对 HRC 时钟频率的精度要求较高，则推荐使用 HRC 48MHz。

#### ◆ BOR 模块特性表（BOR 档位由配置字位 CFG\_BORVS（CFG\_WORD0<11:10>）设置）

CFG_BORVS<1:0>	最小值	典型值	最大值	单位	测试条件
00	—	—	—	V	-40~85°C
01	2.35	2.5	2.65	V	
10	1.9	2.1	2.3	V	
11	2.95	3.1	3.25	V	

#### ◆ LVD 模块特性表



LVD_VS<3:0>		最小值	典型值	最大值	单位	测试条件
VDD 下降, LVDO 低电压 状态标志置 1	0000	2.05	2.2	2.4	V	-40~85℃
	0001	2.25	2.4	2.6	V	
	0010	2.45	2.6	2.8	V	
	0011	2.65	2.8	3.0	V	
	0100	2.85	3.0	3.2	V	
	0101	3.45	3.6	3.8	V	
	0110	3.95	4.1	4.3	V	
	0111	4.55	4.7	4.9	V	
LVD 电压检测迟滞窗口		—	50	130	mV	-40~85℃

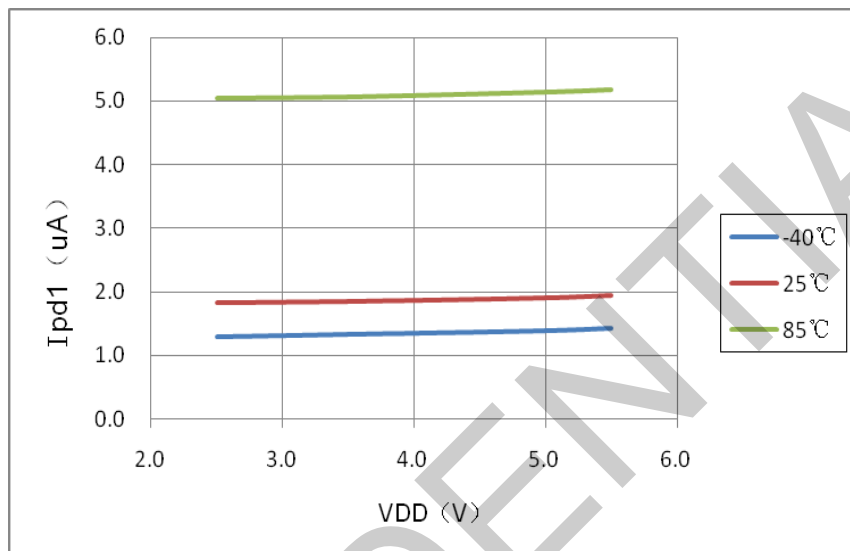
注：芯片在深睡眠模式下，如果 BG 电压模块为低功耗模式（寄存器 SCU\_WAKEUPTIME 的 BG\_STOP=1），则 BOR 和 LVD 档位电压相对于芯片工作模式下的档位电压均会有约±10%范围的偏差；如果应用中需要 BOR 和 LVD 档位电压在芯片深睡眠模式与工作模式下保持一致，则需要禁止 BG 低功耗模式（设置寄存器 SCU\_WAKEUPTIME 的 BG\_STOP=0），此时芯片深睡眠模式功耗会增大约 0.5uA。上表中的 BOR 和 LVD 档位电压描述，为芯片在工作模式（包括浅睡眠模式）或深睡眠模式（寄存器 SCU\_WAKEUPTIME 的 BG\_STOP=0）条件下的档位电压。

## 附录2.2 参数特性图

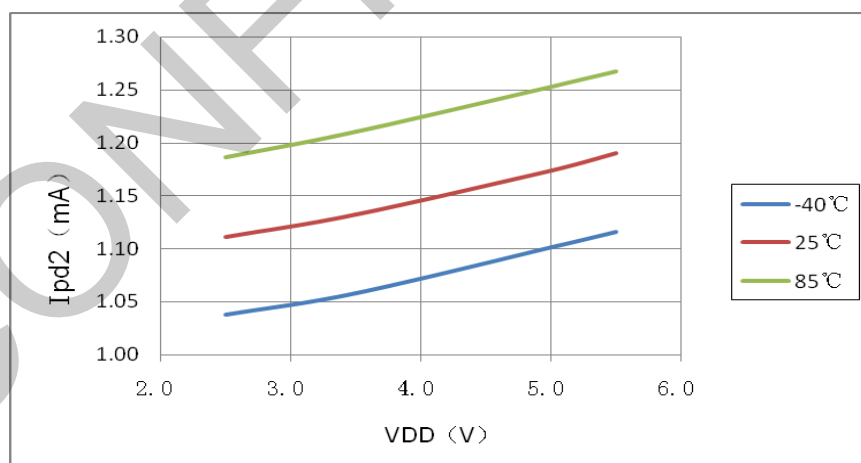
本节中所列图示均为抽样测试，仅作为设计参考之用。其中部分图示中所列的数据已超出指定的操作范围，此类信息也仅供参考，芯片只保证在指定的范围内正常工作。

### 附录2.2.1 芯片功耗特性

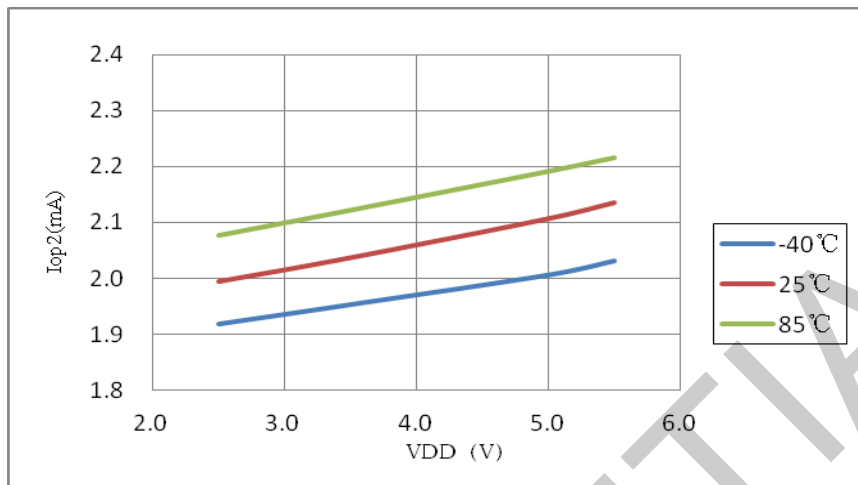
- ◆ 芯片深度睡眠模式电流随电压-温度变化特性图（IWDT 不使能，所有 I/O 端口输出固定电平，无负载）



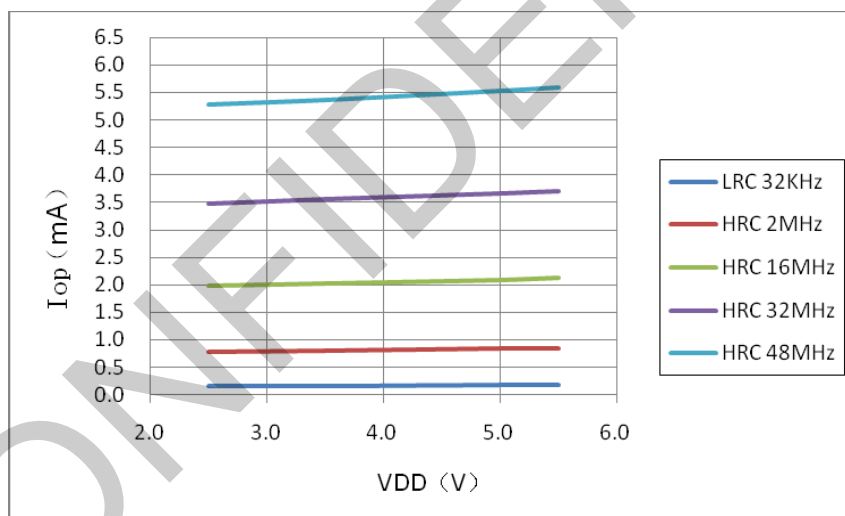
- ◆ 芯片浅睡眠模式电流随电压-温度变化特性图（IWDT 不使能，所有 I/O 端口输出固定电平，无负载；系统主时钟为内部 HRC 16MHz 时钟）



- ◆ 芯片运行模式电流随电压-温度变化特性图 (IWDT 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, ADC 使用 VDD 作为正向参考电压; 系统主时钟为内部 HRC 16MHz 时钟)

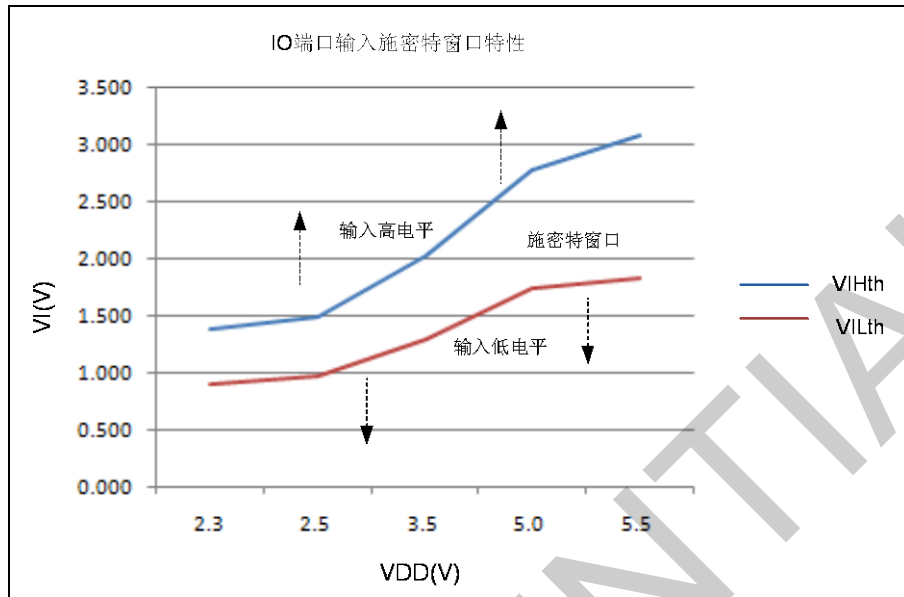


- ◆ 芯片运行模式电流随电压-系统时钟频率变化特性图 (IWDT 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, ADC 使用 VDD 作为正向参考电压; 室温 25°C)

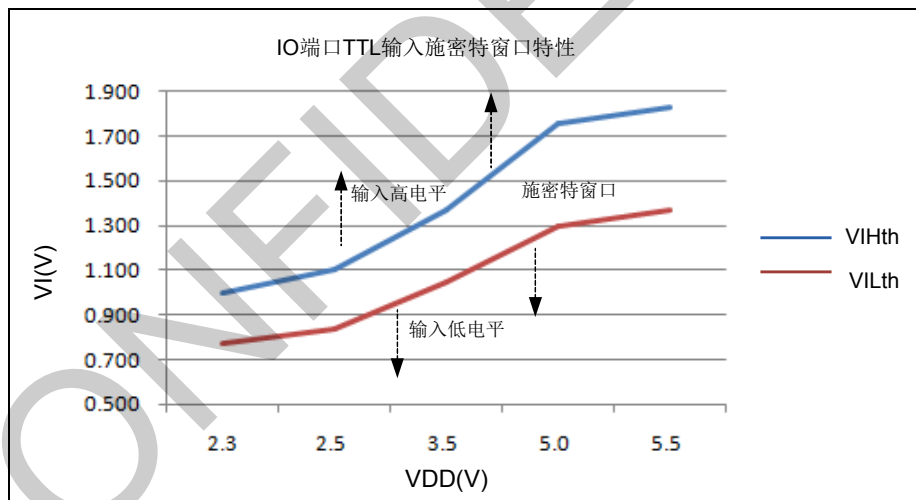


## 附录2.2.2 芯片 IO 端口输入特性

### ◆ I/O 端口 CMOS 输入特性图（室温 25℃）



### ◆ I/O 端口 TTL 输入特性图（室温 25℃）



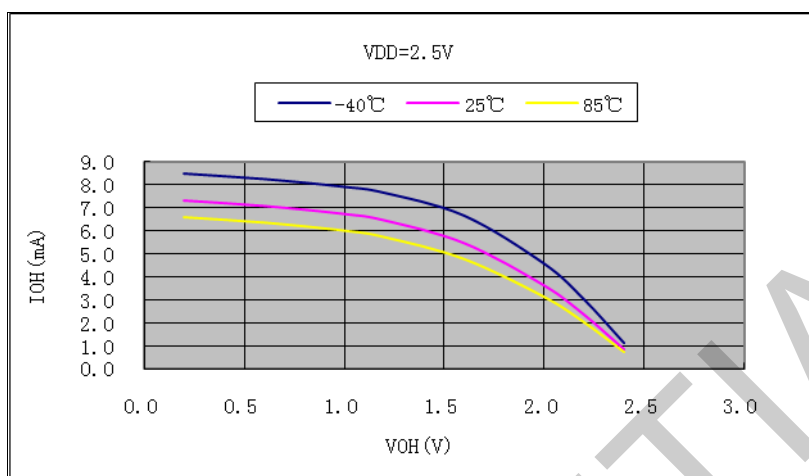
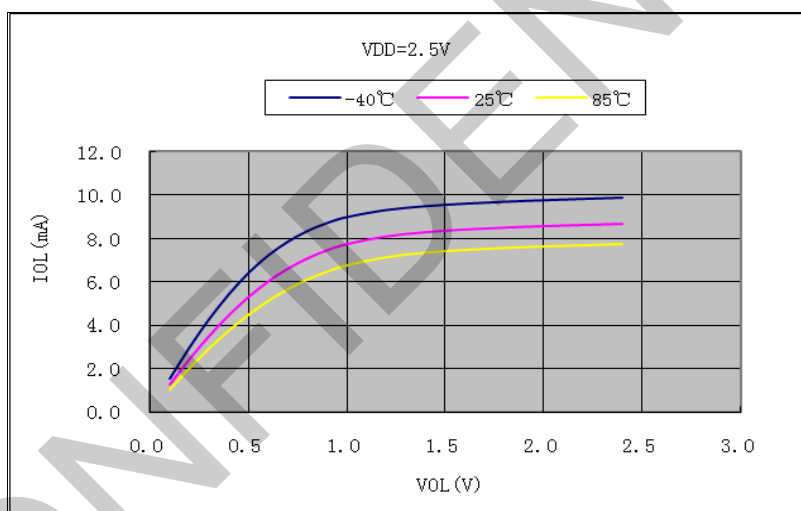
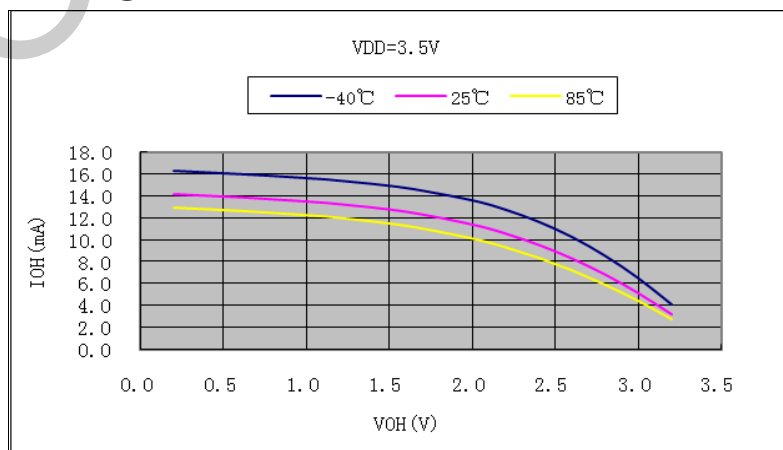
注 1:  $V_{IHth}$  为施密特窗口的上阈值电平，大于该阈值的输入电平为高；

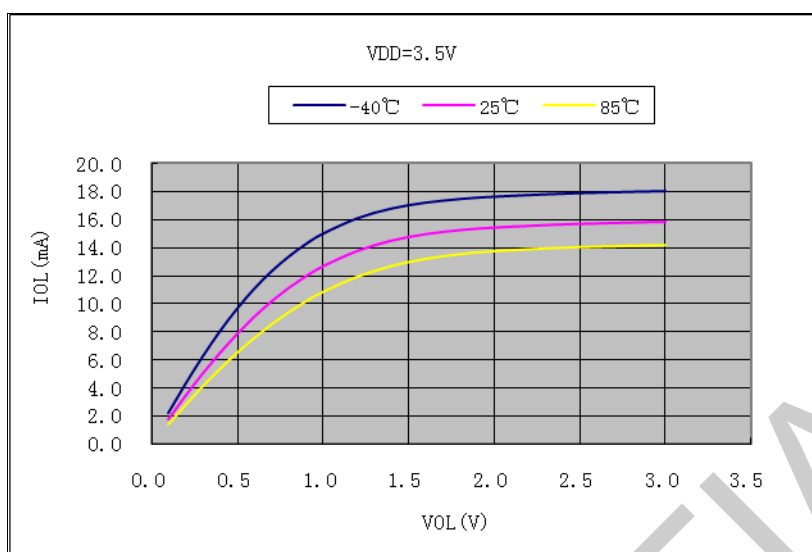
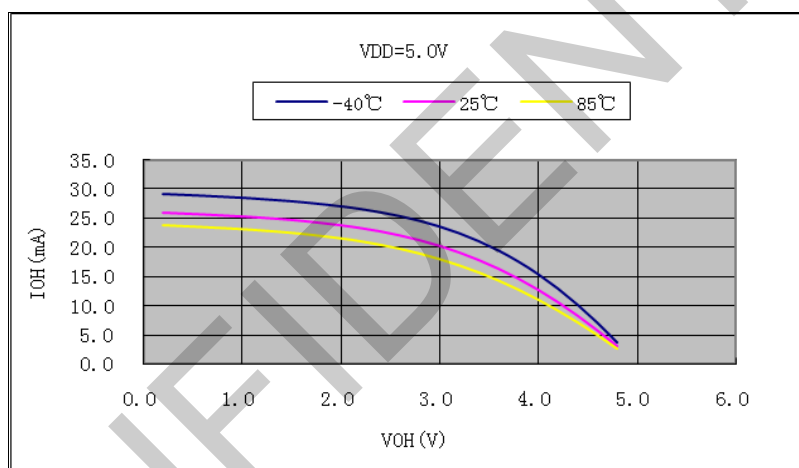
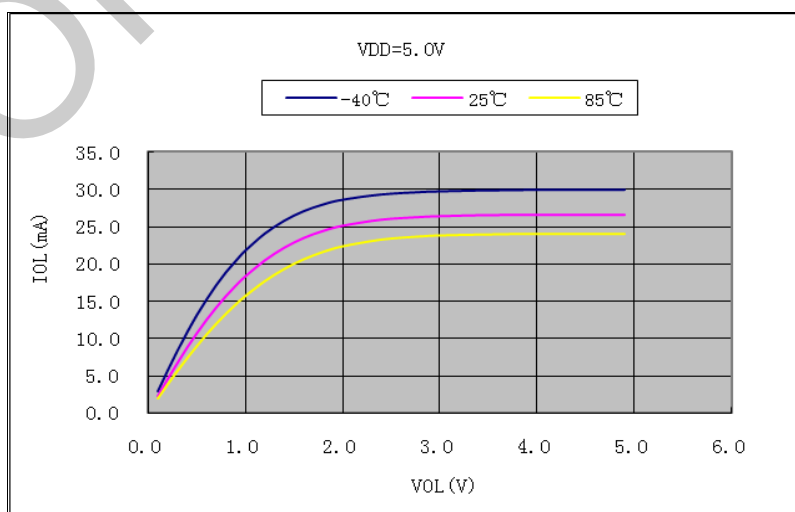
注 2:  $V_{ILth}$  为施密特窗口的下阈值电平，小于该阈值的输入电平为低；

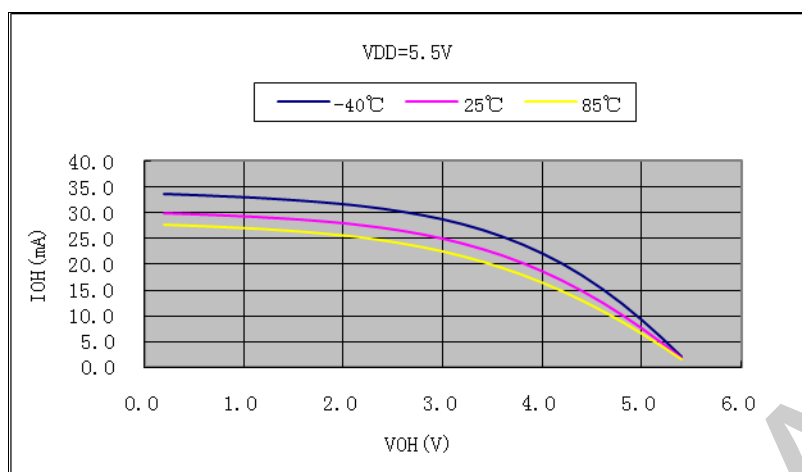
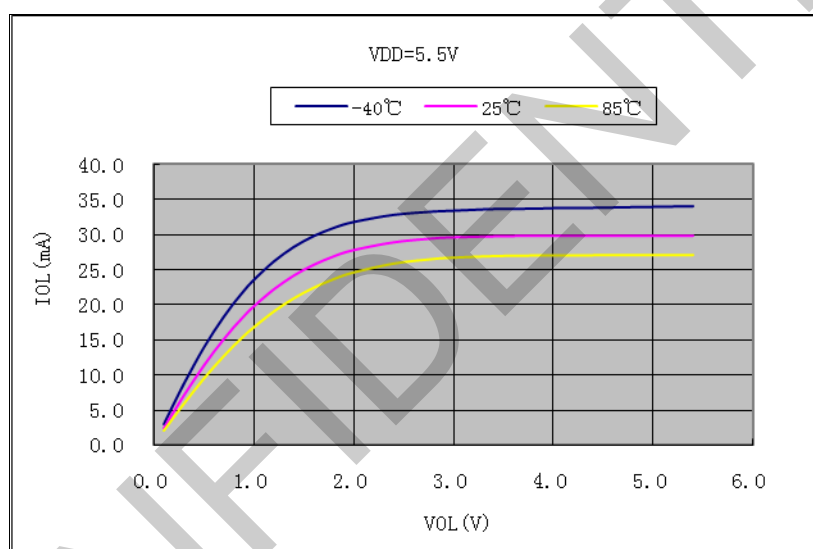
注 3:  $V_{IHth}$  和  $V_{ILth}$  之间为施密特窗口，在窗口内的输入电平不确定，可能为高或低。

## 附录2.2.3 芯片 IO 端口输出特性（普通驱动）

## ◆ I/O 端口信号输出特性图

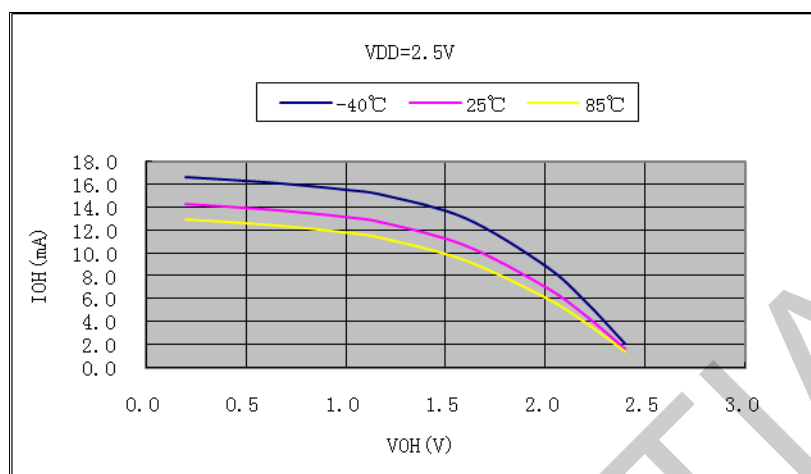
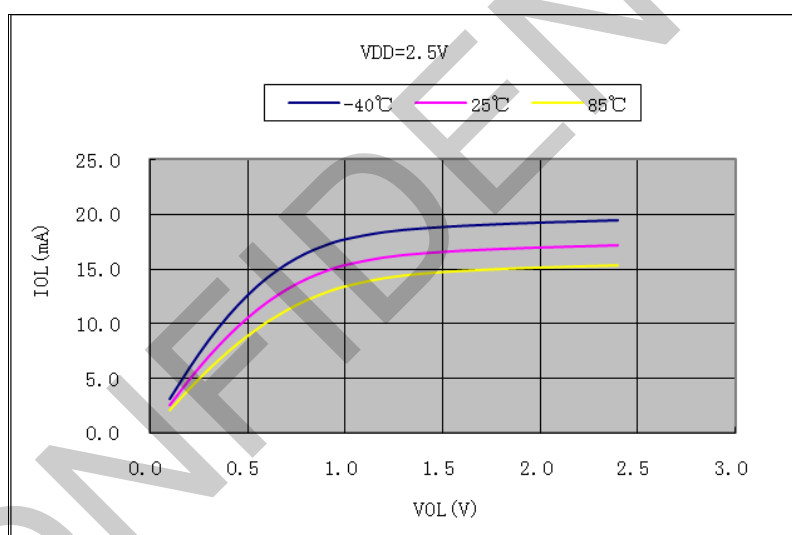
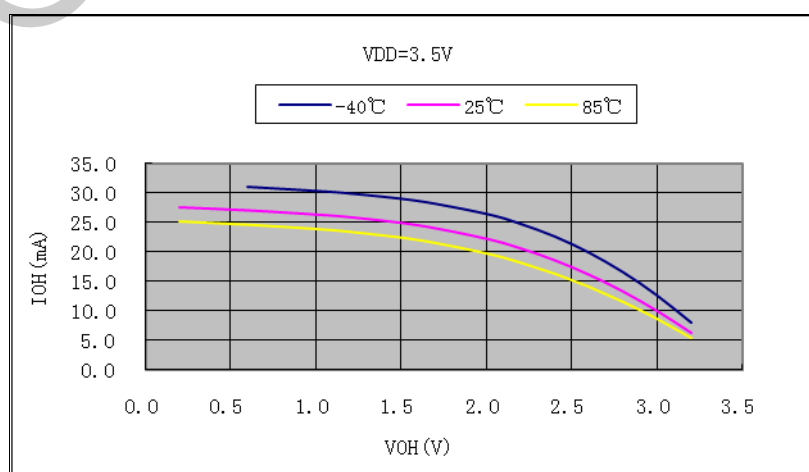
A:  $V_{OH}$  vs  $I_{OH}$ @VDD=2.5VB:  $V_{OL}$  vs  $I_{OL}$ @VDD=2.5VC:  $V_{OH}$  vs  $I_{OH}$ @VDD=3.5V

D:  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=3.5V$ E:  $V_{OH}$  vs  $I_{OH}$ @ $V_{DD}=5.0V$ F:  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=5.0V$ 

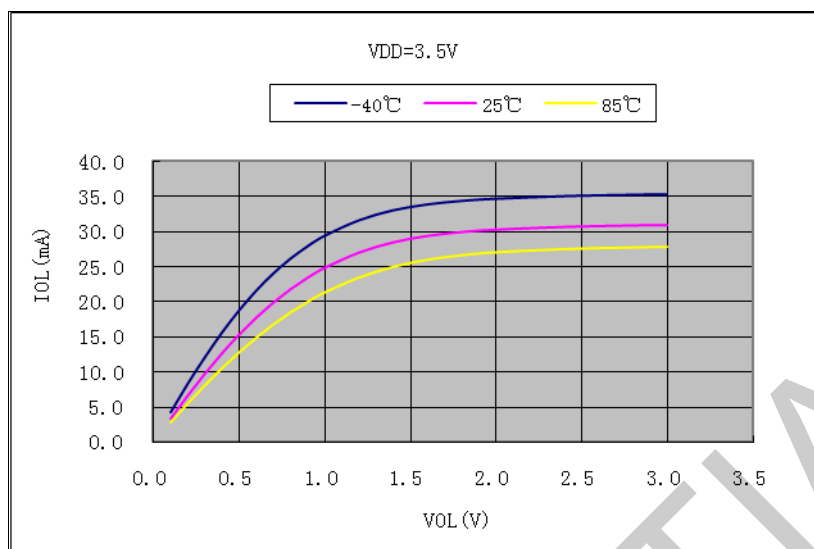
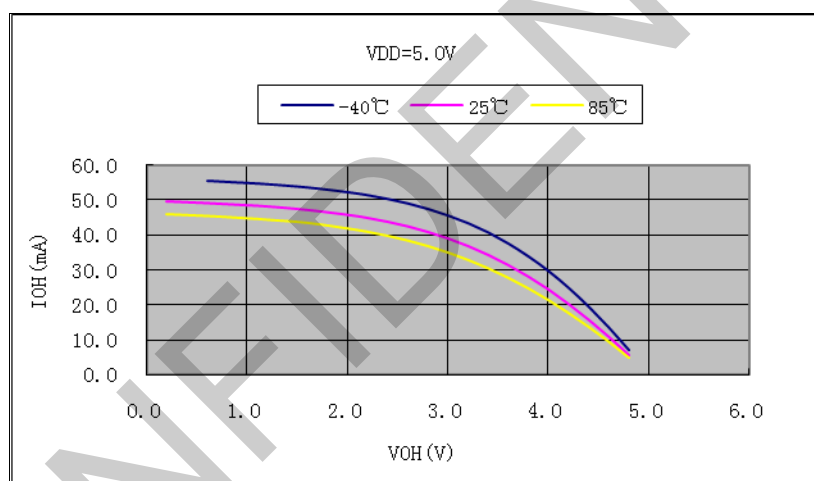
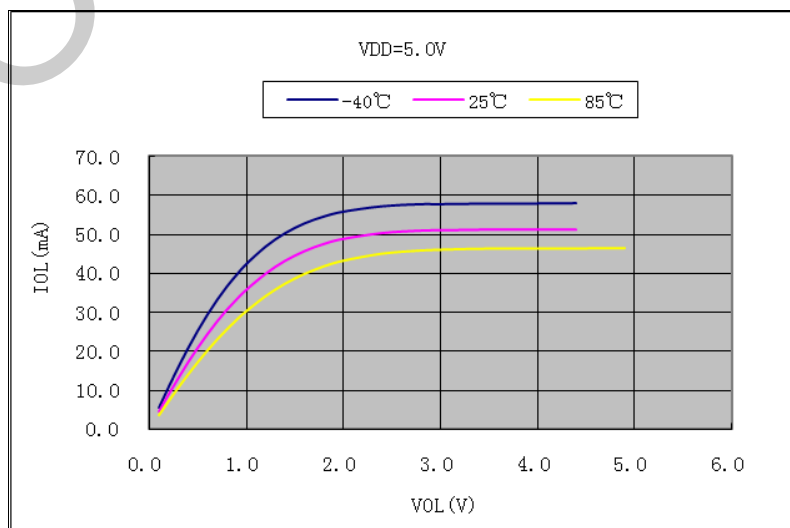
G:  $V_{OH}$  vs  $I_{OH}$ @VDD=5.5VH:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.5V

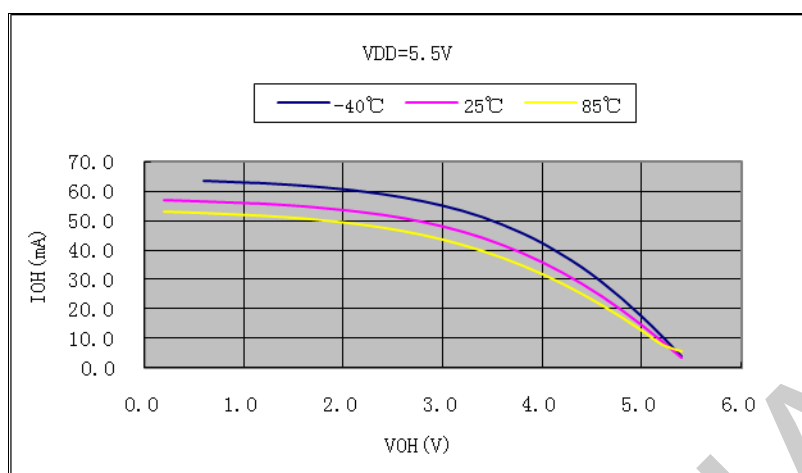
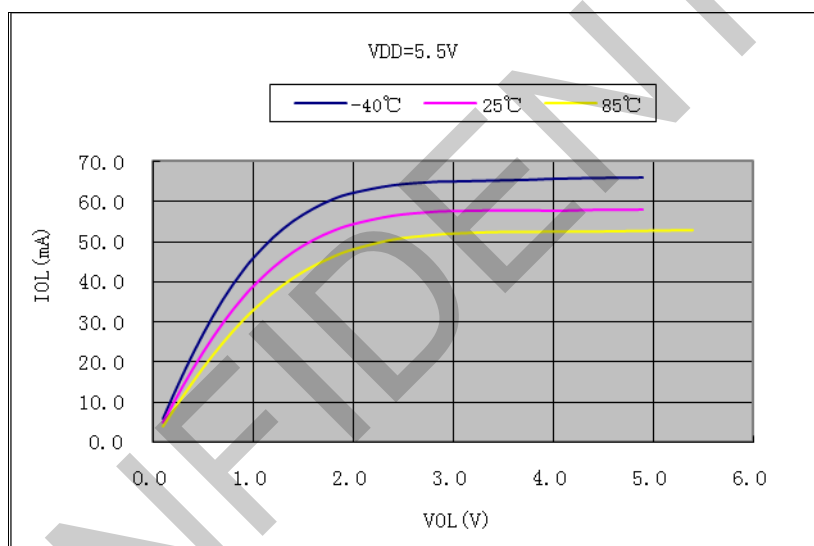
## 附录2.2.4 芯片 IO 端口输出特性（增强驱动，PB8~PB9 端口除外）

## ◆ I/O 端口信号输出特性图

A:  $V_{OH}$  vs  $I_{OH}$ @VDD=2.5VB:  $V_{OL}$  vs  $I_{OL}$ @VDD=2.5VC:  $V_{OH}$  vs  $I_{OH}$ @VDD=3.5V



D:  $V_{OL}$  vs  $I_{OL}$ @VDD=3.5VE:  $V_{OH}$  vs  $I_{OH}$ @VDD=5.0VF:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.0V

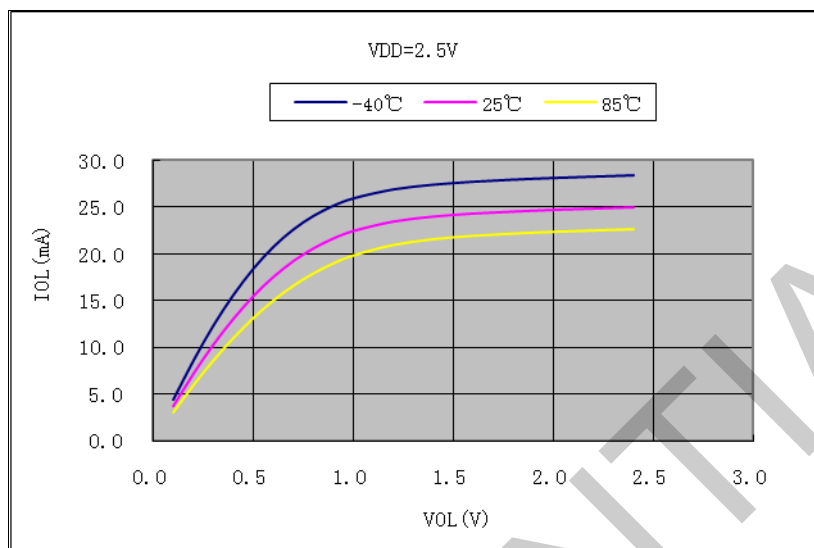
G:  $V_{OH}$  vs  $I_{OH}$ @VDD=5.5VH:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.5V

注：本章节中描述的  $V_{OL}$  vs  $I_{OL}$  增强驱动特性参数图适用于除 PB8~PB9 以外的 IO 端口， $V_{OH}$  vs  $I_{OH}$  增强驱动特性参数图适用于所有 IO 端口。

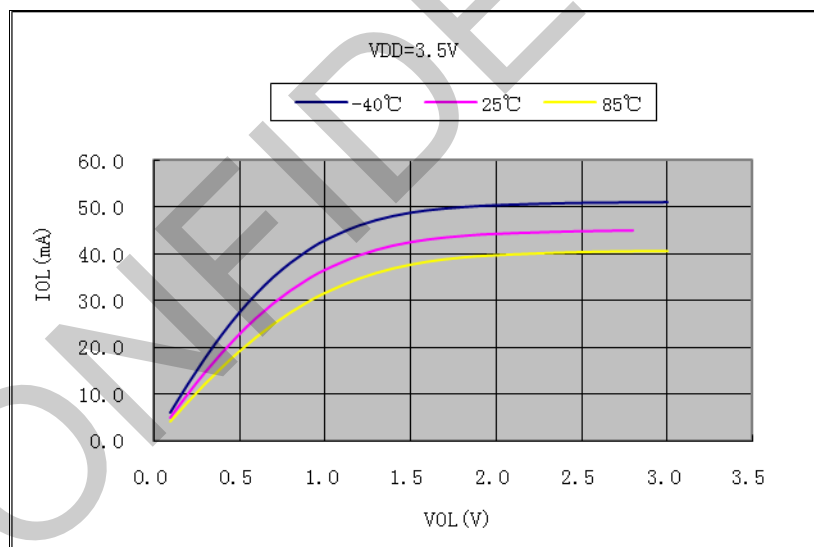
## 附录2.2.5 芯片 IO 端口输出特性（增强驱动，PB8~PB9 端口）

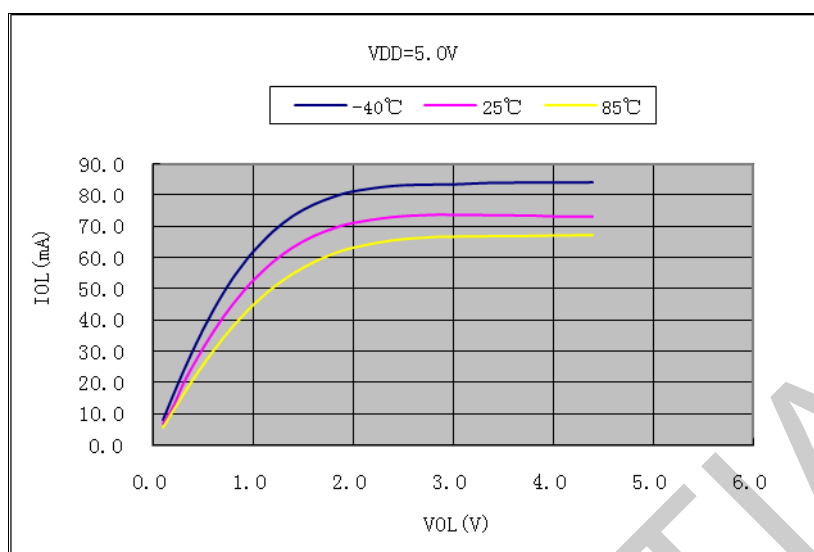
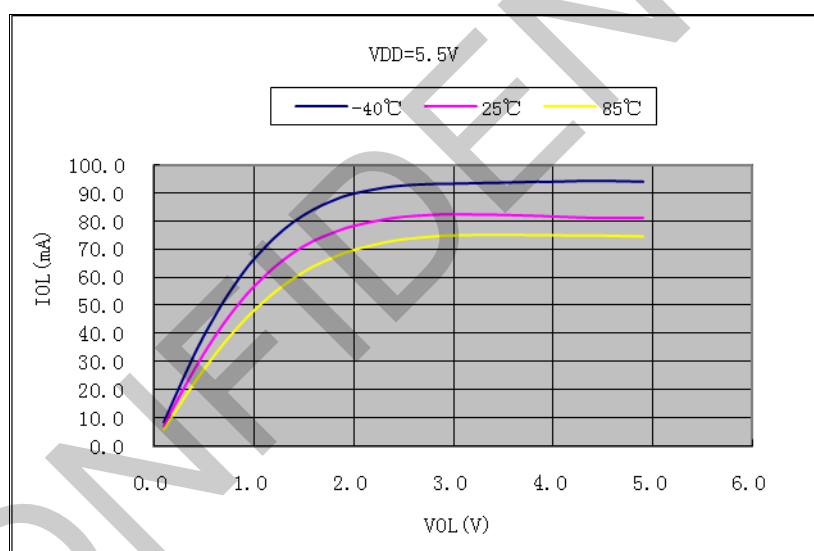
### ◆ I/O 端口信号输出特性图

A:  $V_{OL}$  vs  $I_{OL}$ @VDD=2.5V



B:  $V_{OL}$  vs  $I_{OL}$ @VDD=3.5V



C:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.0VD:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.5V

注：本章节中描述的  $V_{OL}$  vs  $I_{OL}$  增强驱动参数特性图仅适用于 PB8~PB9 端口。

### 附录3 编程调试接口

#### 附录3.1 概述

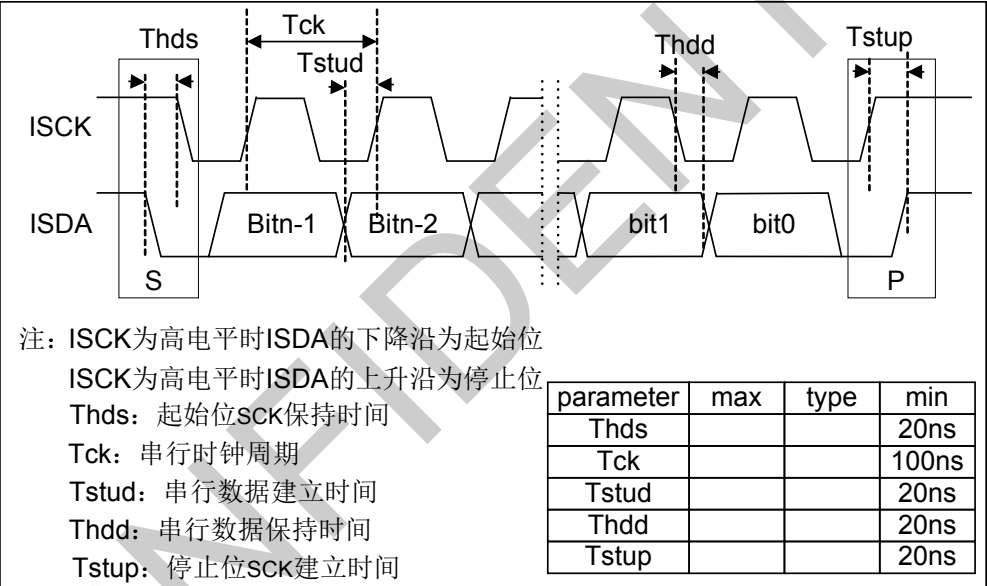
为方便应用程序及实际系统调试，芯片内部集成 ISP 在线编程接口和 SWD 串行调试接口，可实现芯片在线编程、仿真调试功能。

芯片 ISP 和 SWD 功能模块共用 5 线接口配置，即电源线 VDD、地线 VSS、复位线 MRSTN、时钟线 ISCK 和数据线 ISDA。

#### 附录3.2 ISP 编程接口

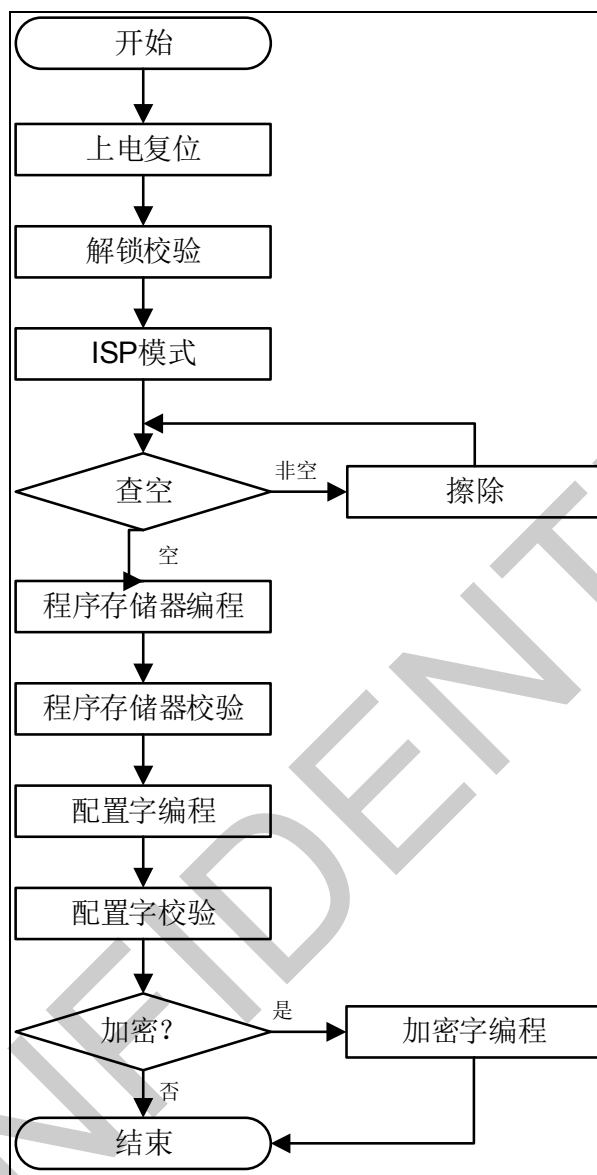
##### 附录3.2.1 通信协议

ISP 接口协议采用两线制半双工通信协议，每个信息块包含 n 个数据信息位，以及起始位和停止位。烧录器为主控设备，ISCK 时钟由烧录器发送；芯片为受控设备。如下图示：



ISP 在线编程接口为 PA14/PA15。

## 附录3.2.2 操作流程



## 附录3.3 SWD 调试接口

### 附录3.3.1 概述

SWD 是 Cortex-M0 内核自带的串行调试接口，与 ARM 的 CoreSight 调试技术兼容。

芯片通过 SWD 调试器完成调试程序(需通过配置字控制位 CFG\_DEBUG, 使能调试模式)下载，然后重新上电，芯片的 SWDIO（复用为 ISDA）和 SWCLK（复用为 ISCK）端口功能可用。

SWCLK：串行时钟输入端口，提供 SWD 串行通讯时钟。

SWDIO：串行数据输入/输出端口。

SWD 调试接口、ISP 通讯接口与 GPIO 复用：PA14/ISCK、PA15/ISDA。

在 SWD 使用过程中，软件不能将 SWDIO、SWCLK 对应的 GPIO 设置为输出，否则会导致 SWD 通讯失效，即仅当 SWDIO、SWCLK 对应的 GPIO 为输入口时 SWD 才可正常使用。用户在程序发布时应注意 SWDIO、SWCLK 对应的 GPIO 端口的处理方式，避免输入悬空。

建议用户在产品量产时将配置字控制位 CFG\_DEBUGEN 置 0 禁止调试模式，并避免调试管脚输入悬空，以提升产品可靠性和安全性。

### 附录3.3.2 SWD 特性

SWD 调试功能可分为侵入式调试和非侵入式调试两部分。

#### 侵入式调试

- ◇ 停机
- ◇ 单步执行
- ◇ 硬件断点（支持 4 个硬件断点）
- ◇ 软件断点（支持 BKPT 指令）
- ◇ 修改程序指针 PC 值
- ◇ 数据观察点 DWT
  - （Data Watchpoint and Trace，只支持 Watchpoint 功能，不支持 Trace 功能）
- ◇ 内部寄存器和 RAM 存储器的读写访问操作
- ◇ 矢量捕捉（包括 Reset 和 HardFault 异常的捕捉）

#### 非侵入式调试

- ◇ 程序指针 PC 值采样