

32 位 MCU  
ES8H0394  
ES8H0354

# 数 据 手 册

- 产品简介
- 数据手册
- 产品规格

上海东软载波微电子有限公司

2023 年 7 月 18 日

## 东软载波 MCU 芯片使用注意事项

### 关于芯片的上/下电

东软载波 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

东软载波 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

东软载波 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

东软载波 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

东软载波 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

东软载波 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

东软载波 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。

### 关于芯片的开发环境

东软载波 MCU 芯片具有完整的软/硬件开发环境，并受知识产权保护。选择上海东软载波微电子有限公司或其指定的第三方公司的汇编器、编译器、编程器、硬件仿真器开发环境，必须遵循与芯片相关的规定和说明。

注：在产品开发时，如遇到不清楚的地方，请通过销售或其它方式与上海东软载波微电子有限公司联系

## 产品订购信息

型号	FLASH	RAM	EEPROM	I/O	Timer	UART	SPI	I2C	ADC	LVD	LCD	封装类型
ES8H0394ELLT	256KB	24KB	1KB	60	16-bit X 6, 32-bit X 1	6	1	1	12bit×27	√	√	LQFP64
ES8H0354ELLR	256KB	24KB	1KB	48	16-bit X 6, 32-bit X 1	6	1	1	12bit×25	√	-	LQFP52
ES8H0394ELLP	256KB	24KB	1KB	42	16-bit X 6, 32-bit X 1	6	1	1	12bit×16	√	√	LQFP44
ES8H0354ELLK	256KB	24KB	1KB	30	16-bit X 6, 32-bit X 1	6	1	1	12bit×18	√	-	LQFP32

地 址：中国上海市徐汇区古美路 1515 号凤凰园 12 号楼 3 楼

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：http://www.essemi.com

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不承担或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系。

## 修订历史

版本	修改日期	更改概要
Preliminary	2023-7-18	Initial
Preliminary	2024-3-1	<ol style="list-style-type: none"><li>1. 在 ADC 模块特性表中，完善关于模拟通道推荐输入电阻的描述，更新输入采样电容参数值。</li><li>2. 添加关于MRSTN管脚复用为GPIO功能的描述，及当MRSTN复用为GPIO管脚时，上电延时固定为使能的备注描述。</li><li>3. LRC 频率从 32K 改为 31.25KHz。</li><li>4. 更新内部时钟源特性表中 HRC 2MHz 的参数描述。</li><li>5. 更新 BOR 模块特性表参数描述，更新配置字中的 BOR 电压点描述。</li><li>6. 添加应用注意事项，在使用外部时钟 XTAL 时，推荐将寄存器位 WAKEUPTIME&lt;11:0&gt;设置为 0xFFF。</li><li>7. 添加 SCU_HRCBCON 寄存器的描述，在使用 HRC 2MHz, 16MHz, 32MHz 时，需进行 HRC 时钟偏置系数设置。</li></ol>
Preliminary	2024-4-29	添加 ES8H0354ELLK 型号描述。

## 目 录

### 内容目录

<b>第 1 章</b>	<b>芯片简介</b> .....	<b>20</b>
1.1	概述 .....	20
1.2	应用领域 .....	23
1.3	结构框图 .....	24
1.4	管脚分配图 .....	25
1.4.1	64-pin 管脚封装图 .....	25
1.4.2	52-pin 管脚封装图 .....	26
1.4.3	44-pin 管脚封装图 .....	27
1.4.4	32-pin 管脚封装图 .....	28
1.5	管脚说明 .....	29
1.5.1	管脚说明 .....	29
1.5.2	管脚对照表 .....	30
1.5.2.1	ES8H0394ELLT 管脚对照表 .....	30
1.5.2.2	ES8H0354ELLR 管脚对照表 .....	32
1.5.2.3	ES8H0394ELLP 管脚对照表 .....	34
1.5.2.4	ES8H0354ELLK 管脚对照表 .....	36
1.5.2.5	ISP0/SWD 管脚对照表 .....	37
1.5.2.6	MCU 和内置 EEPROM 的内部连接关系 .....	37
<b>第 2 章</b>	<b>系统控制及操作特性</b> .....	<b>38</b>
2.1	系统控制保护 .....	38
2.1.1	概述 .....	38
2.1.2	特殊功能寄存器 .....	38
2.2	系统电源 .....	39
2.2.1	结构框图 .....	39
2.2.2	芯片供电电源 .....	39
2.3	系统复位 .....	39
2.3.1	概述 .....	39
2.3.2	结构框图 .....	40
2.3.3	复位时序图 .....	40
2.3.4	外部复位 MRSTN 参考 .....	41
2.3.5	外设模块复位控制 .....	42
2.3.6	特殊功能寄存器 .....	43
2.3.6.1	复位寄存器 (SCU_PWRC) .....	43
2.3.6.2	外设复位控制寄存器 0 (SCU_PRSTEN0) .....	44
2.3.6.3	外设复位控制寄存器 1 (SCU_PRSTEN1) .....	45
2.4	低电压监测 (LVD) .....	47
2.4.1	概述 .....	47
2.4.2	特殊功能寄存器 .....	47
2.4.2.1	低电压监测控制寄存器 (SCU_LVDCON) .....	47
2.5	系统低功耗操作模式 .....	49
2.5.1	概述 .....	49

2.5.2	浅睡眠模式 .....	49
2.5.3	深度睡眠模式 .....	49
2.5.4	睡眠模式的唤醒 .....	50
2.5.5	睡眠模式的唤醒时间 .....	50
2.5.6	FLASH 存储器等待功能 .....	51
2.5.7	特殊功能寄存器 .....	51
2.5.7.1	FLASH 访问等待时间寄存器 (SCU_FLASHWAIT) .....	51
2.6	系统时钟 .....	52
2.6.1	概述 .....	52
2.6.2	结构框图 .....	53
2.6.3	功能说明 .....	53
2.6.3.1	外部时钟 XTAL .....	53
2.6.3.2	内部高速时钟 HRC .....	54
2.6.3.3	内部低速时钟 LRC .....	55
2.6.3.4	锁相环 PLL .....	55
2.6.3.5	时钟输出 CLKO .....	56
2.6.3.6	外部时钟停振检测 CCM .....	56
2.6.3.7	时钟滤波 CLKFLT .....	57
2.6.3.8	睡眠模式系统状态 .....	57
2.6.3.9	浅睡眠模式 .....	57
2.6.3.10	深度睡眠模式 .....	57
2.6.4	特殊功能寄存器 .....	58
2.6.4.1	系统唤醒时间控制寄存器 (SCU_WAKEUPTIME) .....	58
2.6.4.2	系统时钟控制寄存器 0 (SCU_SCLKEN0) .....	59
2.6.4.3	系统时钟控制寄存器 1 (SCU_SCLKEN1) .....	60
2.6.4.4	外设时钟控制寄存器 (SCU_PCLKEN0) .....	62
2.6.4.5	外设时钟控制寄存器 (SCU_PCLKEN1) .....	63
2.6.4.6	外设时钟检测控制寄存器 (SCU_CCM) .....	65
2.6.4.7	PLL 锁定中断控制寄存器 (SCU_PLLLKCON) .....	66
2.6.4.8	HRC 时钟偏置控制寄存器 (SCU_HRCBCON) .....	67
2.6.4.9	LCD 时钟配置寄存器 (SCU_LCDCLK) .....	67
2.6.5	系统时钟应用说明 .....	68
2.7	中断和异常处理 .....	69
2.7.1	中断和异常 .....	69
2.7.2	中断和异常向量的分配 .....	70
2.7.3	中断向量表的重映射 .....	71
2.7.4	特殊功能寄存器 .....	72
2.7.4.1	不可屏蔽中断控制寄存器 (SCU_NMICON) .....	72
2.7.4.2	中断向量表重映射使能寄存器 (SCU_TBLREMAPEN) .....	72
2.7.4.3	中断向量表偏移寄存器 (SCU_TBLOFF) .....	73
2.7.4.4	硬件错误标志寄存器 (SCU_FAULTFLAG) .....	73
2.7.4.5	IRQ0~31 置中断请求使能寄存器 (NVIC_ISER) .....	74
2.7.4.6	IRQ0~31 清中断请求使能寄存器 (NVIC_ICER) .....	74
2.7.4.7	IRQ0~31 置中断挂起寄存器 (NVIC_ISPR) .....	75
2.7.4.8	IRQ0~31 清中断挂起寄存器 (NVIC_ICPR) .....	75

2.7.4.9	IRQ0~3 优先级控制寄存器 (NVIC_PR0)	76
2.7.4.10	IRQ4~7 优先级控制寄存器 (NVIC_PR1)	76
2.7.4.11	IRQ8~11 优先级控制寄存器 (NVIC_PR2)	77
2.7.4.12	IRQ12~15 优先级控制寄存器 (NVIC_PR3)	77
2.7.4.13	IRQ16~19 优先级控制寄存器 (NVIC_PR4)	78
2.7.4.14	IRQ20~23 优先级控制寄存器 (NVIC_PR5)	79
2.7.4.15	IRQ24~27 优先级控制寄存器 (NVIC_PR6)	79
2.7.4.16	IRQ28~31 优先级控制寄存器 (NVIC_PR7)	80
2.8	系统控制块 (SCB)	81
2.8.1	概述	81
2.8.2	特殊功能寄存器	81
2.8.2.1	SCB_CPUID 寄存器 (SCB_CPUID)	81
2.8.2.2	中断控制和状态寄存器 (SCB_ICSR)	81
2.8.2.3	应用中断和复位控制寄存器 (SCB_AIRCR)	82
2.8.2.4	系统控制寄存器 (SCB_SCR)	83
2.8.2.5	配置和控制寄存器 (SCB_CCR)	83
2.8.2.6	系统处理程序优先级寄存器 2 (SCB_SHPR2)	84
2.8.2.7	系统处理程序优先级寄存器 3 (SCB_SHPR3)	84
2.9	系统定时器 (SYSTICK)	85
2.9.1	概述	85
2.9.2	特殊功能寄存器	86
2.9.2.1	SYSTICK 控制和状态寄存器 (SYST_CSR)	86
2.9.2.2	SYSTICK 重装值寄存器 (SYST_RVR)	86
2.9.2.3	SYSTICK 当前值寄存器 (SYST_CVR)	87
2.9.2.4	SYSTICK 校准值寄存器 (SYST_CALIB)	87
2.10	配置字软件控制	88
2.10.1.1	系统配置软件控制寄存器 (SCU_SOFTCFG)	88
2.11	定时器 (T16N/T32N) 同步启动关停控制	89
2.11.1	概述	89
2.11.2	特殊功能寄存器	89
2.11.2.1	SCU_TIMEREN 使能控制寄存器 (SCU_TIMEREN)	89
2.11.2.2	SCU_TIMERDIS 关停控制寄存器 (SCU_TIMERDIS)	90
2.12	模块调试相关配置	91
2.12.1	概述	91
2.12.2	特殊功能寄存器	91
2.12.2.1	调试模式模块暂停控制寄存器 (SCU_DBGHALT)	91
<b>第 3 章</b>	<b>存储器资源</b>	<b>93</b>
3.1	内部存储器地址映射	93
3.2	FLASH 存储器	93
3.2.1	信息区 FLASH	93
3.2.1.1	芯片配置字	94
3.2.1.2	用户程序校验码和用户识别码	97
3.2.1.3	芯片唯一识别码 UID	97
3.2.2	程序区 FLASH	98
3.3	功能描述	99

3.3.1	IAP 概述 .....	99
3.3.2	Flash 保护 .....	99
3.3.2.1	IAP 操作保护 KEY .....	99
3.3.2.2	Flash 写保护区 .....	99
3.3.2.3	数据 Flash 区（程序存储器中划分独立 DATA 区） .....	99
3.3.2.4	Flash 全局读保护 .....	99
3.3.3	Flash 程序区全擦除 .....	100
3.3.4	Flash 页擦除 .....	101
3.3.5	Flash 字编程 .....	101
3.3.6	Flash 编程数据 FIFO .....	102
3.3.7	IAP 自编程硬件固化模块 .....	102
3.3.7.1	CODE 区单页擦函数 .....	102
3.3.7.2	CODE 区单字编程函数 .....	102
3.3.7.3	CODE 区多字编程 .....	102
3.3.7.4	DATA 区单页擦函数 .....	103
3.3.7.5	DATA 区单字编程函数 .....	103
3.3.7.6	DATA 区多字编程 .....	103
3.3.8	特殊功能寄存器 .....	104
3.3.8.1	FLASH 程序区密钥寄存器（IAP_FLASHKEY） .....	104
3.3.8.2	FLASH 擦除编程地址寄存器（IAP_FLASHADDR） .....	104
3.3.8.3	FLASH 编程数据 FIFO 寄存器（IAP_FLASHFIFO） .....	104
3.3.8.4	FLASH 编程数据寄存器（IAP_FLASHHDR） .....	105
3.3.8.5	FLASH 操作命令寄存器（IAP_FLASHCMD） .....	105
3.3.8.6	FLASH 控制寄存器（IAP_FLASHCR） .....	106
3.3.8.7	FLASH 状态寄存器（IAP_FLASHSR） .....	106
3.3.8.8	FLASH 擦除编程地址反码寄存器（IAP_FLASHADDINV） .....	108
3.4	数据存储器（SRAM） .....	109
3.4.1	SRAM 地址映射 .....	109
3.4.2	SRAM 位带扩展 .....	109
3.5	外设寄存器 .....	110
3.5.1	外设寄存器映射 .....	110
3.5.2	外设寄存器位带扩展 .....	110
3.5.3	系统控制单元（SCU）寄存器列表 .....	111
3.5.4	CRC 寄存器列表 .....	112
3.5.5	GPIO 寄存器列表 .....	112
3.5.6	IAP 寄存器列表 .....	113
3.5.7	ADC 寄存器列表 .....	114
3.5.8	LCD 寄存器列表 .....	114
3.5.9	WWDT 寄存器列表 .....	115
3.5.10	IWDT 寄存器列表 .....	115
3.5.11	T16N0/T16N1/T16N2/T16N3/T16N4/T16N5 寄存器列表 .....	115
3.5.12	T32N0 寄存器列表 .....	116
3.5.13	UART0/UART1/UART2/UART3/UART4/UART5 寄存器列表 .....	116
3.5.14	SPI0 寄存器列表 .....	117
3.5.15	I2C0 寄存器列表 .....	117



3.6	内核寄存器 .....	118
3.6.1	系统定时器 (SYSTICK) 寄存器列表 .....	118
3.6.2	中断控制器 (NVIC) 寄存器列表 .....	118
3.6.3	系统控制块 (SCB) 寄存器列表 .....	118
<b>第4章</b>	<b>输入输出端口 (GPIO) .....</b>	<b>119</b>
4.1	概述 .....	119
4.2	结构框图 .....	120
4.3	外部端口中断 .....	120
4.4	外部按键中断 .....	122
4.5	Buzz 输出 .....	123
4.6	特殊功能寄存器 .....	125
4.6.1	PA 端口状态寄存器 (GPIO_PAPORT) .....	125
4.6.2	PA 端口数据寄存器 (GPIO_PADATA) .....	125
4.6.3	PA 端口输出置位寄存器 (GPIO_PADATABSR) .....	125
4.6.4	PA 端口输出清零寄存器 (GPIO_PADATABCR) .....	126
4.6.5	PA 端口输出翻转寄存器 (GPIO_PADATABRR) .....	126
4.6.6	PA 端口方向控制寄存器 (GPIO_PADIR) .....	126
4.6.7	PA 端口方向置位寄存器 (GPIO_PADIRBSR) .....	127
4.6.8	PA 端口方向清零寄存器 (GPIO_PADIRBCR) .....	127
4.6.9	PA 端口方向翻转寄存器 (GPIO_PADIRBRR) .....	128
4.6.10	PA<7:0>端口复用选择寄存器 (GPIO_PAFUNC0) .....	128
4.6.11	PA<15:8>端口复用选择寄存器 (GPIO_PAFUNC1) .....	129
4.6.12	PA<23:16>端口复用选择寄存器 (GPIO_PAFUNC2) .....	130
4.6.13	PA<31:24>端口复用选择寄存器 (GPIO_PAFUNC3) .....	130
4.6.14	PA 端口输入控制寄存器 (GPIO_PAINEB) .....	131
4.6.15	PA 端口开漏控制寄存器 (GPIO_PAODE) .....	132
4.6.16	PA 端口弱上拉使能寄存器 (GPIO_PAPUE) .....	132
4.6.17	PA 端口弱下拉使能寄存器 (GPIO_PAPDE) .....	132
4.6.18	PA 端口驱动电流控制寄存器 (GPIO_PADS) .....	133
4.6.19	PA 端口类型选择寄存器 (GPIO_PATYP) .....	133
4.6.20	PA 端口滤波控制寄存器 (GPIO_PAFLT) .....	133
4.6.21	PB 端口状态寄存器 (GPIO_PBPORT) .....	134
4.6.22	PB 端口数据寄存器 (GPIO_PBDATA) .....	134
4.6.23	PB 端口输出置位寄存器 (GPIO_PBDATABSR) .....	134
4.6.24	PB 端口输出清零寄存器 (GPIO_PBDATABCR) .....	135
4.6.25	PB 端口输出翻转寄存器 (GPIO_PBDATABRR) .....	135
4.6.26	PB 端口方向控制寄存器 (GPIO_PBDIR) .....	136
4.6.27	PB 端口方向置位寄存器 (GPIO_PBDIRBSR) .....	136
4.6.28	PB 端口方向清零寄存器 (GPIO_PBDIRBCR) .....	136
4.6.29	PB 端口方向翻转寄存器 (GPIO_PBDIRBRR) .....	137
4.6.30	PB<7:0>端口复用选择寄存器 (GPIO_PBFUNC0) .....	137
4.6.31	PB<15:8>端口复用选择寄存器 (GPIO_PBFUNC1) .....	138
4.6.32	PB<23:16>端口复用选择寄存器 (GPIO_PBFUNC2) .....	139
4.6.33	PB<27:24>端口复用选择寄存器 (GPIO_PBFUNC3) .....	140
4.6.34	PB 端口输入控制寄存器 (GPIO_PBINEB) .....	140

4.6.35	PB 端口开漏控制寄存器 (GPIO_PBODE)	141
4.6.36	PB 端口弱上拉使能寄存器 (GPIO_PBPUE)	141
4.6.37	PB 端口弱下拉使能寄存器 (GPIO_PBPDE)	142
4.6.38	PB 端口驱动电流控制寄存器 (GPIO_PBDS)	142
4.6.39	PB 端口类型选择寄存器 (GPIO_PBTYP)	142
4.6.40	PB 端口滤波控制寄存器 (GPIO_PBFILT)	143
4.6.41	PINT 中断使能寄存器 (GPIO_PINTIE)	143
4.6.42	PINT 中断标志寄存器 (GPIO_PINTIF)	144
4.6.43	PINT0~7 中断源选择寄存器 (GPIO_PINTSEL0)	144
4.6.44	PINT8~11 中断源选择寄存器 (GPIO_PINTSEL1)	145
4.6.45	PINT0~7 中断配置寄存器 (GPIO_PINTCFG0)	146
4.6.46	PINT8~11 中断配置寄存器 (GPIO_PINTCFG1)	147
4.6.47	KINT 中断使能寄存器 (GPIO_KINTIE)	147
4.6.48	KINT 中断标志寄存器 (GPIO_KINTIF)	148
4.6.49	KINT0~7 中断源选择寄存器 (GPIO_KINTSEL0)	148
4.6.50	KINT8~11 中断源选择寄存器 (GPIO_KINTSEL1)	149
4.6.51	KINT0~7 中断配置寄存器 (GPIO_KINTCFG0)	150
4.6.52	KINT8~11 中断配置寄存器 (GPIO_KINTCFG1)	151
4.6.53	PINT 中断滤波控制寄存器 (GPIO_FLTCR)	151
4.6.54	脉宽调制寄存器 (GPIO_TXPWM)	152
4.6.55	BUZ 控制寄存器 (GPIO_BUZC)	154
<b>第 5 章</b>	<b>外设</b>	<b>156</b>
5.1	定时器/计数器	156
5.1.1	16 位定时器/计数器 T16N (T16N0~5)	156
5.1.1.1	概述	156
5.1.1.2	结构框图	157
5.1.1.3	T16N 定时/计数功能	157
5.1.1.4	T16N 输入捕捉功能	159
5.1.1.5	T16N 输出调制功能	160
5.1.1.6	特殊功能寄存器	163
5.1.1.7	T16N 计数值寄存器 (T16N_CNT)	163
5.1.1.8	T16N 预分频器计数值寄存器 (T16N_PRECNT)	163
5.1.1.9	T16N 预分频器计数匹配寄存器 (T16N_PREMAT)	163
5.1.1.10	T16N 控制寄存器 0 (T16N_CON0)	164
5.1.1.11	T16N 控制寄存器 1 (T16N_CON1)	165
5.1.1.12	T16N 控制寄存器 2 (T16N_CON2)	166
5.1.1.13	T16N 中断使能寄存器 (T16N_IE)	168
5.1.1.14	T16N 中断标志寄存器 (T16N_IF)	169
5.1.1.15	T16N 触发寄存器 (T16N_TRG)	170
5.1.1.16	T16N 计数匹配寄存器 0 (T16N_MAT0)	171
5.1.1.17	T16N 计数匹配寄存器 1 (T16N_MAT1)	171
5.1.1.18	T16N 计数匹配寄存器 2 (T16N_MAT2)	171
5.1.1.19	T16N 计数匹配寄存器 3 (T16N_MAT3)	172
5.1.1.20	T16N 计数峰值寄存器 (T16N_TOP)	172
5.1.1.21	T16N 应用说明	172

5.1.2	32 位定时器/计数器 T32N (T32N0)	173
5.1.2.1	概述	173
5.1.2.2	结构框图	173
5.1.2.3	T32N 定时/计数功能	174
5.1.2.4	T32N 输入捕捉功能	175
5.1.2.5	T32N 输出调制功能	177
5.1.2.6	特殊功能寄存器	179
5.1.2.7	T32N 计数值寄存器 (T32N_CNT)	179
5.1.2.8	T32N 预分频器计数值寄存器 (T32N_PRECNT)	179
5.1.2.9	T32N 预分频器计数匹配寄存器 (T32N_PREMAT)	179
5.1.2.10	T32N 控制寄存器 0 (T32N_CON0)	180
5.1.2.11	T32N 控制寄存器 1 (T32N_CON1)	181
5.1.2.12	T32N 控制寄存器 2 (T32N_CON2)	182
5.1.2.13	T32N 中断使能寄存器 (T32N_IE)	183
5.1.2.14	T32N 中断标志寄存器 (T32N_IF)	184
5.1.2.15	T32N 触发寄存器 (T32N_TRG)	185
5.1.2.16	T32N 计数匹配寄存器 0 (T32N_MAT0)	185
5.1.2.17	T32N 计数匹配寄存器 1 (T32N_MAT1)	186
5.1.2.18	T32N 计数匹配寄存器 2 (T32N_MAT2)	186
5.1.2.19	T32N 计数匹配寄存器 3 (T32N_MAT3)	186
5.1.2.20	T32N 应用说明	187
5.2	通用异步接收/发送器 (UART0~5)	188
5.2.1	概述	188
5.2.2	结构框图	188
5.2.3	UART 数据格式	189
5.2.4	UART 异步发送器	189
5.2.5	UART 异步接收器	192
5.2.6	UART 发送调制功能	195
5.2.7	UART 红外唤醒功能	196
5.2.8	UART 端口极性	196
5.2.9	UART 单线半双工通信	196
5.2.10	特殊功能寄存器	197
5.2.10.1	UART 控制寄存器 (UART_CON)	197
5.2.10.2	UART 波特率寄存器 (UART_BRR)	198
5.2.10.3	UART 发送数据写入寄存器 (UART_TBW)	199
5.2.10.4	UART 接收数据读取寄存器 (UART_RBR)	199
5.2.10.5	UART 发送缓冲 0/1 寄存器 (UART_TB01)	199
5.2.10.6	UART 发送缓冲 2/3 寄存器 (UART_TB23)	200
5.2.10.7	UART 接收缓冲 0/1 寄存器 (UART_RB01)	201
5.2.10.8	UART 接收缓冲 2/3 寄存器 (UART_RB23)	202
5.2.10.9	UART 中断使能寄存器 (UART_IE)	202
5.2.10.10	UART 中断标志寄存器 (UART_IF)	203
5.2.11	UART 应用说明	205
5.3	SPI 同步串口通讯控制器 (SPI0)	206
5.3.1	概述	206

5.3.2	结构框图.....	206
5.3.3	SPI 通讯模式.....	206
5.3.4	SPI 数据格式.....	206
5.3.5	SPI 帧位宽.....	208
5.3.6	SPI 同步发送器.....	208
5.3.7	SPI 同步接收器.....	210
5.3.8	SPI 通讯控制.....	211
5.3.9	SPI 延迟接收功能.....	212
5.3.10	SPI 数据帧发送间隔功能.....	212
5.3.11	特殊功能寄存器.....	213
5.3.11.1	SPI 控制寄存器 (SPI_CON).....	213
5.3.11.2	SPI 发送数据写入寄存器 (SPI_TBW).....	214
5.3.11.3	SPI 接收数据读取寄存器 (SPI_RBR).....	214
5.3.11.4	SPI 中断使能寄存器 (SPI_IE).....	215
5.3.11.5	SPI 中断标志寄存器 (SPI_IF).....	216
5.3.11.6	SPI 发送缓冲寄存器 (SPI_TB).....	217
5.3.11.7	SPI 接收缓冲寄存器 (SPI_RB).....	217
5.3.11.8	SPI 状态寄存器 (SPI_STA).....	218
5.3.11.9	SPI 波特率设置寄存器 (SPI_CKS).....	219
5.3.12	SPI 应用说明.....	219
5.4	I2C 总线串口通讯控制器 (I2C0).....	220
5.4.1	概述.....	220
5.4.2	结构框图.....	220
5.4.3	I2C 总线基本原理.....	220
5.4.3.1	I2C 通讯协议.....	220
5.4.3.2	I2C 数据传输格式.....	221
5.4.4	I2C 通讯端口配置.....	222
5.4.5	I2C 时基定时器与 16 倍速采样器.....	223
5.4.6	I2C 通讯发送器.....	224
5.4.7	I2C 通讯接收器.....	226
5.4.8	I2C 通讯控制.....	227
5.4.8.1	I2C 起始位.....	228
5.4.8.2	I2C 停止位.....	229
5.4.8.3	I2C 应答延迟功能.....	229
5.4.8.4	I2C 数据帧传输间隔功能.....	230
5.4.8.5	I2C 时钟线自动下拉等待请求功能.....	230
5.4.8.6	I2C 自动发送未应答功能.....	231
5.4.9	特殊功能寄存器.....	232
5.4.9.1	I2C 控制寄存器 (I2C_CON).....	232
5.4.9.2	I2C 工作模式寄存器 (I2C_MOD).....	233
5.4.9.3	I2C 中断使能寄存器 (I2C_IE).....	235
5.4.9.4	I2C 中断标志寄存器 (I2C_IF).....	236
5.4.9.5	I2C 发送数据写入寄存器 (I2C_TBW).....	237
5.4.9.6	I2C 接收数据读取寄存器 (I2C_RBR).....	238
5.4.9.7	I2C 发送缓冲寄存器 (I2C_TB).....	238

5.4.9.8	I2C 接收缓冲寄存器 (I2C_RB)	238
5.4.9.9	I2C 状态寄存器 (I2C_STA)	239
5.4.9.10	I2C 控制寄存器 2 (I2C_CON2)	240
5.4.10	I2C 应用说明	240
5.5	模数转换器 (ADC)	241
5.5.1	概述	241
5.5.2	结构框图	241
5.5.3	ADC 基本配置	241
5.5.4	ADC 高精度参考电压	241
5.5.5	ADC 数据转换	242
5.5.6	自动转换比较功能	244
5.5.7	特殊功能寄存器	246
5.5.7.1	ADC 参考控制寄存器 (ADC_VREFCON)	246
5.5.7.2	ADC 转换值寄存器 (ADC_DR)	246
5.5.7.3	ADC 控制寄存器 0 (ADC_CON0)	247
5.5.7.4	ADC 控制寄存器 1 (ADC_CON1)	248
5.5.7.5	ADC 通道选择寄存器 (ADC_CHS)	249
5.5.7.6	ADC 中断使能寄存器 (ADC_IE)	250
5.5.7.7	ADC 中断标志寄存器 (ADC_IF)	251
5.5.7.8	ADC 自动转换比较控制寄存器 (ADC_ACPC)	251
5.5.7.9	ADC 自动转换比较阈值寄存器 (ADC_ACPCMP)	252
5.5.7.10	ADC 自动转换均值数据寄存器 (ADC_ACPMEAN)	253
5.5.7.11	ADC 结果数据偏置寄存器 (ADC_OFFDR)	253
5.6	硬件独立看门狗 (IWDT)	254
5.6.1	概述	254
5.6.2	特殊功能寄存器	255
5.6.2.1	IWDT 计数器装载值寄存器 (IWDT_LOAD)	255
5.6.2.2	IWDT 计数器当前值寄存器 (IWDT_VALUE)	255
5.6.2.3	IWDT 控制寄存器 (IWDT_CON)	256
5.6.2.4	IWDT 中断标志清除寄存器 (IWDT_INTCLR)	256
5.6.2.5	IWDT 中断标志寄存器 (IWDT_RIS)	257
5.6.2.6	IWDT 锁定寄存器 (IWDT_LOCK)	257
5.7	窗口看门狗 (WWDT)	259
5.7.1	概述	259
5.7.2	特殊功能寄存器	261
5.7.2.1	WWDT 计数器装载值寄存器 (WWDT_LOAD)	261
5.7.2.2	WWDT 计数器当前值寄存器 (WWDT_VALUE)	261
5.7.2.3	WWDT 控制寄存器 (WWDT_CON)	261
5.7.2.4	WWDT 中断标志清除寄存器 (WWDT_INTCLR)	262
5.7.2.5	WWDT 中断标志寄存器 (WWDT_RIS)	262
5.7.2.6	WWDT 锁定寄存器 (WWDT_LOCK)	263
5.8	循环冗余校验 (CRC)	264
5.8.1	概述	264
5.8.2	运用说明	264
5.8.3	特殊功能寄存器	266

5.8.3.1	CRC 解锁寄存器 (CRC_UL)	266
5.8.3.2	CRC 控制寄存器 (CRC_CON)	266
5.8.3.3	CRC 触发寄存器 (CRC_TRIG)	267
5.8.3.4	CRC FLASH 校验起始地址寄存器 (CRC_ADDR)	268
5.8.3.5	CRC FLASH 校验数据块大小寄存器 (CRC_SIZE)	268
5.8.3.6	CRC 用户模式输入数据寄存器 (CRC_DI)	269
5.8.3.7	CRC 校验结果寄存器 (CRC_DO)	269
5.8.3.8	CRC 状态寄存器 (CRC_STA)	269
5.8.3.9	CRC 当前 Flash 地址寄存器 (CRC_FA)	270
5.9	液晶驱动控制器 (LCD)	271
5.9.1	概述	271
5.9.2	特性	271
5.9.3	结构框图	271
5.9.4	LCD 工作时钟	272
5.9.5	LCD 显示帧频率	272
5.9.6	LCD 闪烁功能	272
5.9.7	LCD 工作模式	272
5.9.7.1	占空比	272
5.9.7.2	偏压	272
5.9.7.3	偏压电阻	273
5.9.7.4	显示灰度	273
5.9.7.5	波形类型	273
5.9.8	COM/SEG 端口说明	273
5.9.9	LCD 输出波形	274
5.9.10	LCD 基本设置	280
5.9.11	特殊功能寄存器	281
5.9.11.1	LCD 模式控制寄存器 (LCD_MODE)	281
5.9.11.2	LCD 控制寄存器 (LCD_CTRL)	283
5.9.11.3	LCD 闪烁控制寄存器 1 (LCD_BLINK1)	285
5.9.11.4	LCD 闪烁控制寄存器 2 (LCD_BLINK2)	286
5.9.11.5	LCD 更新状态寄存器 (LCD_RIF)	287
5.9.11.6	LCD 状态标志清除寄存器 (LCD_ICR)	288
5.9.11.7	LCD 运行状态寄存器 (LCD_BUSY)	288
5.9.11.8	LCD 显示数据寄存器 0 (LCD_RAM0)	289
5.9.11.9	LCD 显示数据寄存器 1 (LCD_RAM1)	289
5.9.11.10	LCD 显示数据寄存器 2 (LCD_RAM2)	290
5.9.11.11	LCD 显示数据寄存器 3 (LCD_RAM3)	290
5.9.11.12	LCD 显示数据寄存器 4 (LCD_RAM4)	291
5.9.11.13	LCD 显示数据寄存器 5 (LCD_RAM5)	291
5.9.11.14	LCD 显示数据寄存器 6 (LCD_RAM6)	292
5.9.11.15	LCD 显示数据寄存器 7 (LCD_RAM7)	292
5.9.11.16	LCD 显示数据寄存器 8 (LCD_RAM8)	293
5.9.11.17	LCD 显示数据寄存器 9 (LCD_RAM9)	293
5.9.11.18	LCD 显示数据寄存器 10 (LCD_RAM10)	294
5.9.11.19	LCD 显示数据寄存器 11 (LCD_RAM11)	294

5.9.11.20	LCD 显示数据寄存器 12 (LCD_RAM12)	295
5.9.11.21	LCD 显示数据寄存器 13 (LCD_RAM13)	295
5.9.11.22	LCD 显示数据寄存器 14 (LCD_RAM14)	296
5.9.11.23	LCD 显示数据寄存器 15 (LCD_RAM15)	296
5.9.11.24	LCD SEG 输出使能寄存器 0 (LCD_SEGOE0)	297
5.9.11.25	LCD SEG 输出使能寄存器 1 (LCD_SEGOE1)	297
<b>第 6 章</b>	<b>芯片封装外观尺寸图</b>	<b>298</b>
6.1	64-PIN 封装外观尺寸图	298
6.2	52-PIN 封装外观尺寸图	299
6.3	44-PIN 封装外观尺寸图	300
6.4	32-pin 封装外观尺寸图	301
<b>附录 1</b>	<b>Cortex-M0 内核描述</b>	<b>302</b>
附录 1.1	Cortex-M0 指令集	302
附录 1.2	Cortex-M0 内核寄存器	304
附录 1.2.1	通用寄存器 R0~R12	304
附录 1.2.2	堆栈指针寄存器 SP (R13)	304
附录 1.2.3	链接寄存器 LR (R14)	305
附录 1.2.4	程序计数器 PC (R15)	305
附录 1.2.5	程序状态寄存器 xPSR	305
附录 1.2.6	异常/中断屏蔽寄存器 PRIMASK	306
附录 1.2.7	控制寄存器 CONTROL	306
<b>附录 2</b>	<b>MCU 电气特性</b>	<b>307</b>
附录 2.1	参数特性表	307
附录 2.1.1	芯片工作条件	307
附录 2.1.2	芯片特性参数测量方法	308
附录 2.1.3	芯片功耗特性	308
附录 2.1.4	芯片 IO 端口特性	310
附录 2.1.5	芯片系统时钟特性	311
附录 2.1.6	芯片 ESD 特性	312
附录 2.1.7	芯片功能模块特性	312
附录 2.2	参数特性图	316
附录 2.2.1	芯片功耗特性	316
附录 2.2.2	芯片 IO 端口输入特性	318
附录 2.2.3	芯片 IO 端口输出特性 (普通驱动, PA0~PA3 端口除外)	320
附录 2.2.4	芯片 IO 端口输出特性 (增强驱动, PA0~PA3 端口除外)	323
附录 2.2.5	芯片 IO 端口输出特性 (普通驱动, PA0~PA3 端口)	326
附录 2.2.6	芯片 IO 端口输出特性 (增强驱动, PA0~PA3 端口)	329
<b>附录 3</b>	<b>编程调试接口</b>	<b>332</b>
附录 3.1	概述	332
附录 3.2	ISP 编程接口	332
附录 3.2.1	通信协议	332
附录 3.2.2	操作流程	333
附录 3.3	SWD 调试接口	333
附录 3.3.1	概述	333
附录 3.3.2	SWD 特性	334

<b>附录 4</b>	<b>内置 EEPROM 芯片 .....</b>	<b>335</b>
附录 4.1	概述 .....	335
附录 4.2	管脚说明 .....	335
附录 4.3	结构框图 .....	335
附录 4.4	数据传输协议说明 .....	336
附录 4.5	低功耗待机模式 (Standby) .....	336
附录 4.6	存储器软复位 .....	336
附录 4.7	存储器上电复位 .....	337
附录 4.8	器件寻址 .....	337
附录 4.9	EEPROM 写操作 .....	337
附录 4.10	EEPROM 读操作 .....	338
附录 4.11	内置 EEPROM 电气特性 .....	340
附录 4.11.1	芯片工作条件 .....	340
附录 4.11.2	直流电气特性表 .....	340
附录 4.11.3	交流电气特性表 .....	340
附录 4.11.4	可靠性参数表 .....	341



**图目录**

图 1-1	芯片结构框图 .....	24
图 1-2	ES8H0394ELLT LQFP64 封装顶视图 .....	25
图 1-3	ES8H0354ELLR LQFP52 封装顶视图 .....	26
图 1-4	ES8H0394ELLP LQFP44 封装顶视图 .....	27
图 1-5	ES8H0354ELLK LQFP32 封装顶视图 .....	28
图 2-1	系统电源结构框图 .....	39
图 2-2	系统复位电路结构框图 .....	40
图 2-3	上电复位时序示意图 .....	40
图 2-4	掉电复位时序示意图 .....	41
图 2-5	外部复位时序示意图 .....	41
图 2-6	MRSTN 复位参考电路图 1 .....	41
图 2-7	MRSTN 复位参考电路图 2 .....	41
图 2-8	MRSTN 复位参考电路图 3 .....	42
图 2-9	MRSTN 复位参考电路图 4 .....	42
图 2-10	LVD 低电压监测示意图 .....	47
图 2-11	系统时钟电路结构框图 .....	53
图 2-12	XTAL 振荡器电路结构示意图 .....	54
图 2-13	晶体振荡器等效电路示意图 .....	54
图 2-14	系统定时器框图 .....	85
图 3-1	内部存储系统分配示意图 .....	93
图 3-2	Boot Flash 启动地址分配示意图 .....	98
图 3-3	SRAM 映射图 .....	109
图 3-4	外设存储器分配示意图 .....	110
图 4-1	IO 端口电流结构图 .....	120
图 4-2	外部端口中断 PINT0 电路结构示意图 .....	121
图 4-3	外部按键中断 KINT 电路结构示意图 .....	122
图 4-4	Buzz 高电平调制输出波形图 .....	124
图 4-5	Buzz 低电平调制输出波形图 .....	124
图 5-1	T16N0 电路结构框图 .....	157
图 5-2	T16N 计数匹配功能示意图 .....	158
图 5-3	T16N0 捕捉功能示意图 .....	160
图 5-4	T16N 输出调制功能示意图 .....	162
图 5-5	T32N 电路结构框图 .....	173
图 5-6	T32N 计数匹配功能示意图 .....	175
图 5-7	T32N 捕捉功能示意图 .....	176
图 5-8	T32N 输出调制功能示意图 .....	178
图 5-9	UART 电路结构图 .....	188
图 5-10	UART 7 位数据格式 .....	189
图 5-11	UART 8 位数据格式 .....	189
图 5-12	UART 9 位数据格式 .....	189
图 5-13	UART 发送数据流示意图 .....	190
图 5-14	UART 发送中断标志示意图 .....	191
图 5-15	UART 发送数据操作流程图示例 .....	192

图 5-16	UART 接收数据流示意图 .....	193
图 5-17	UART 接收中断标志示意图 .....	194
图 5-18	UART 接收数据操作流程示意图 .....	195
图 5-19	高电平调制输出波形图 .....	196
图 5-20	低电平调制输出波形图 .....	196
图 5-21	SPI 电路结构框图 .....	206
图 5-22	SPI 时钟上升沿发送，下降沿接收波形示意图 .....	207
图 5-23	SPI 时钟下降沿发送，上升沿接收波形示意图 .....	207
图 5-24	SPI 时钟上升沿接收，下降沿发送波形示意图 .....	207
图 5-25	SPI 时钟下降沿接收，上升沿发送波形示意图 .....	208
图 5-26	SPI 发送数据流示意图 .....	209
图 5-27	SPI 发送中断标志示意图 .....	209
图 5-28	SPI 接收数据流示意图 .....	210
图 5-29	SPI 接收中断标志示意图 .....	211
图 5-30	SPI 延迟接收功能波形示意图 .....	212
图 5-31	I2C 电路结构框图 .....	220
图 5-32	I2C 总线通讯协议示意图 .....	221
图 5-33	I2C 主控器写入从动器数据示意图 .....	222
图 5-34	I2C 主控器读取从动器数据示意图 .....	222
图 5-35	开漏输出端口示意图 .....	223
图 5-36	I2C 总线端口信号的波形示意图 .....	224
图 5-37	I2C 发送数据流示意图 .....	224
图 5-38	I2C 发送中断标志示意图 .....	225
图 5-39	I2C 接收数据流示意图 .....	226
图 5-40	I2C 接收中断标志示意图 .....	227
图 5-41	I2C 起始位波形图 .....	228
图 5-42	I2C 自动寻呼波形图 .....	228
图 5-43	I2C 停止位波形图 .....	229
图 5-44	I2C 应答延迟功能波形示意图 .....	230
图 5-45	I2C 数据帧传输间隔功能波形示意图 .....	230
图 5-46	I2C 时钟线下拉等待波形示意图 .....	231
图 5-47	ADC 内部结构图 .....	241
图 5-48	ADC 数据转换时序示意图 (ADC_CON1 寄存器的 SMPS=0, 软件控制采样) .....	242
图 5-49	ADC 数据转换时序示意图 (ADC_CON1 寄存器的 SMPS=1, 硬件控制采样) .....	243
图 5-50	看门狗时序图 .....	255
图 5-51	窗口看门狗中断和溢出复位产生时序图 (WWDTWIN 设定为 25%) .....	260
图 5-52	错误的喂狗时序图 (WWDTWIN 设定为 25%) .....	260
图 5-53	LCD 驱动结构图 .....	271
图 6-1	内核寄存器 .....	304
图 6-2	堆栈操作示意图 .....	305

## 表目录

表 1-1	管脚说明.....	29
表 1-2	ES8H0394ELLT 管脚对照表.....	31
表 1-3	ES8H0354ELLR 管脚对照表.....	33
表 1-4	ES8H0394ELLP 管脚对照表.....	35
表 1-5	ES8H0354ELLK 管脚对照表.....	36
表 1-6	ISP0/SWD 管脚对照表.....	37
表 2-1	低功耗模式时钟状态表.....	49
表 2-2	异常/中断优先级操作类型说明表.....	69
表 2-3	异常/中断优先级列表.....	70
表 2-4	IRQ 分配列表.....	71
表 4-1	PINT 选择分组 0 对应列表.....	121
表 4-2	PINT 选择分组 1 对应列表.....	121
表 4-3	KINT 选择分组 0 对应列表.....	123
表 4-4	KINT 选择分组 1 对应列表.....	123
表 5-1	I2C 总线端口信号的时序参数列表.....	223
表 5-2	ADC 精度与转换速率的对应关系列表.....	242

## 第1章 芯片简介

### 1.1 概述

ES8H0394/ES8H0354 系列是基于 ARM Cortex-M0 CPU 内核的高性能低功耗 32 位通用微控制器。内部集成多个 16 位和 32 位定时器/计数器，带红外发送调制功能的 UART 模块，SPI 和 I2C 通信模块，液晶驱动控制器 LCD 模块，用于系统电源监测的 LVD 模块，分辨率最高为 12 位的可配置 ADC 模块等资源和外设。

#### ◆ 工作条件

- ◇ 工作电压范围：2.2V ~ 5.5V
- ◇ 工作温度范围：-40 ~ 85°C（工业级）
- ◇ 工作主时钟频率：32KHz，400KHz~48MHz
- ◇ 工作电流：Ivdd = 4mA（@内部 HRC 16MHz，典型值）
- ◇ 待机电流：Ivdd = 3.5uA（常温，典型值）

#### ◆ 封装

- ◇ LQFP64 封装
- ◇ LQFP52 封装
- ◇ LQFP44 封装
- ◇ LQFP32 封装

#### ◆ 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 MRSTN 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%；48MHz 常温为±0.6%，-40°C~85°C 范围内为±1%）
- ◇ 内部 31.25KHz RC 振荡器（LRC）作为 WDT 时钟源，可配置为系统时钟源，出厂前已校准（常温频率精度为±4%）
- ◇ 支持 PLL 倍频，时钟源可选择，最大可倍频至 48MHz，可配置为系统时钟源
- ◇ 系统上电默认主时钟为 16MHz HRC 时钟

#### ◆ 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）

- ◇ 支持一组 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
- ◆ 硬件独立看门狗 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
  - ◇ 256K 字节 FLASH 存储器
    - 支持 ISP 在线串行编程
    - 支持两组 ISP 编程接口
    - 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 全加密的编程代码加密保护
  - ◇ 支持 8K 字节 Boot Flash
    - 通过芯片配置字设置从 Boot Flash 或主程序区启动
  - ◇ 1K 字节 EERPOM 存储器（为内置芯片）
  - ◇ 24K 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
- ◆ 通用 CRC16/32
  - ◇ 支持 Flash 数据完整性检查
  - ◇ 支持数据通信 CRC 校验
  - ◇ 可设定需进行 CRC 校验的 Flash 数据块的起始地址和大小
- ◆ I/O 端口
  - ◇ 最多 60 个双向 I/O 端口
    - PA 端口（PA0~PA31，其中 PA0~PA3 这 4 个端口支持 100mA 大电流驱动）
    - PB 端口（PB0~PB27）
  - ◇ 支持 12 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
- ◆ 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
- ◇ T16N4: 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
- ◇ T16N5: 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, 输入捕捉/输出调制功能
- ◆ **UART 通信接口**
  - ◇ 支持 6 路 UART 通信接口 UART0/UART1/UART2/UART3/UART4/UART5
  - ◇ 支持全/半双工异步通信模式
  - ◇ 支持单线半双工异步通信模式
  - ◇ 支持小数分频波特率可配置
  - ◇ 支持 4 级发送/接收缓冲器
  - ◇ 支持 7/8/9 位数据格式可配
  - ◇ 支持奇偶校验功能可配, 支持硬件自动奇偶校验位判断
  - ◇ 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
  - ◇ 接收端口支持红外唤醒功能
  - ◇ 支持 UART 输入输出通讯端口极性可配置
- ◆ **I2C 通信接口**
  - ◇ 支持 1 路通信接口 I2C
  - ◇ 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 1 路通信接口 SPI
  - ◇ 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 8/10/12 位分辨率，有效精度为 11 位
- ◇ 支持外部最多 27 通道模拟输入端
- ◇ 支持一路内部 1/4 VDD 通道输入
- ◇ 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
- ◆ LCD 液晶显示控制器（仅 ES8H0394ELLT 和 ES8H0394ELLP 支持）
  - ◇ 最大支持 8×40、6×42、4×44 的显示段数
  - ◇ 支持 A、B 两种驱动波形
  - ◇ 支持四种偏置电压模式：静态偏压、1/2 偏压、1/3 偏压、1/4 偏压
  - ◇ 支持输出波形 1/4、1/6、1/8 占空比配置
  - ◇ 支持显示灰度调节功能
  - ◇ 支持 3 种显示闪烁模式
  - ◇ 支持休眠模式下运行

## 1.2 应用领域

本芯片可用于白色家电、厨卫电器、智能硬件、工业仪表等领域。

### 1.3 结构框图

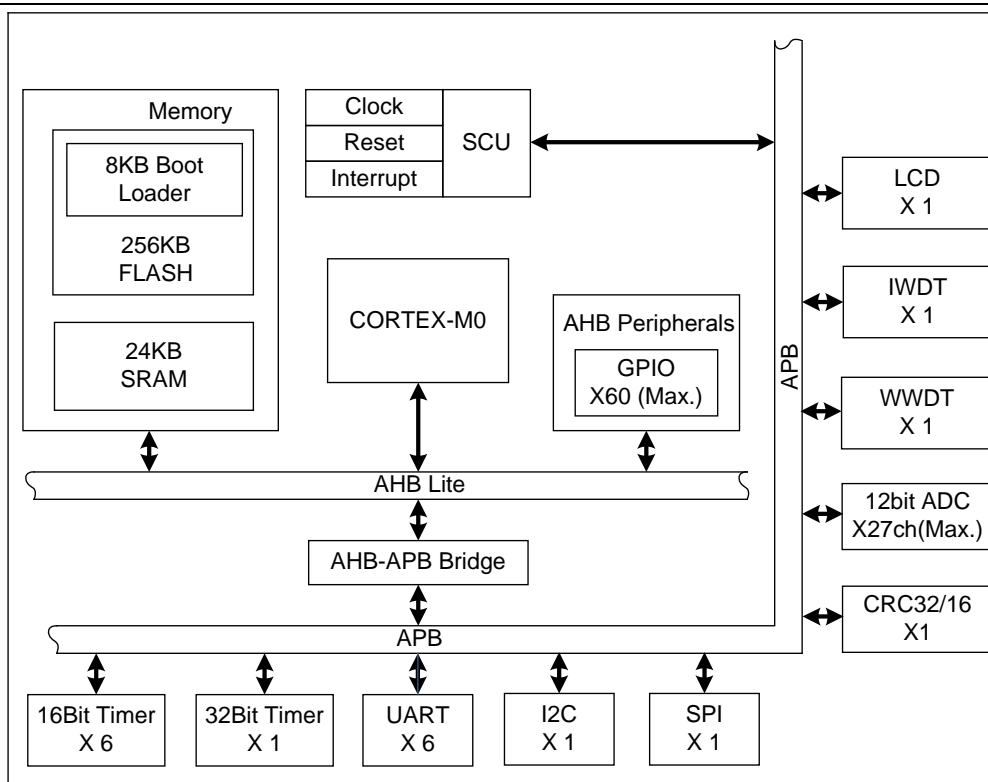


图 1-1 芯片结构框图



## 1.4 管脚分配图

### 1.4.1 64-pin管脚封装图

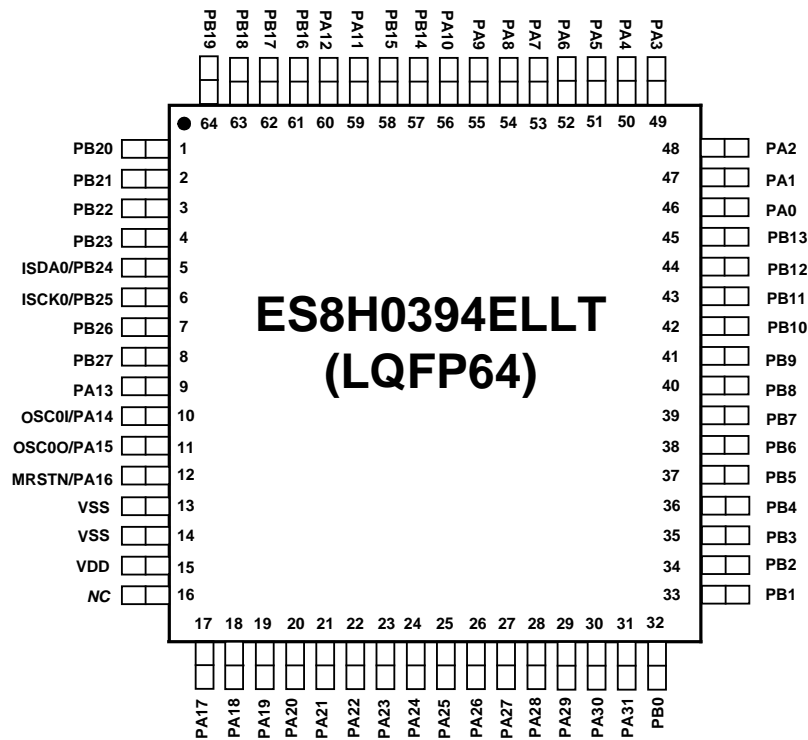


图 1-2 ES8H0394ELLT LQFP64 封装顶视图

### 1.4.2 52-pin管脚封装图

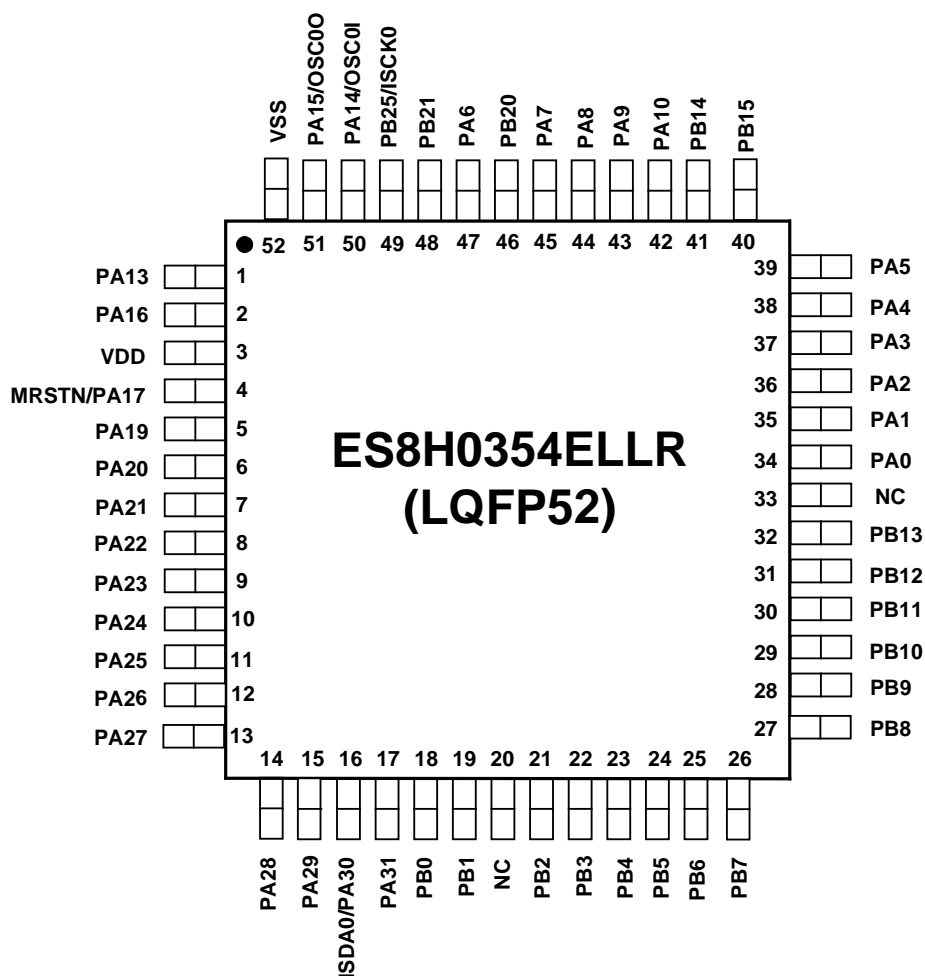


图 1-3 ES8H0354ELLR LQFP52 封装顶视图

1.4.3 44-pin管脚封装图

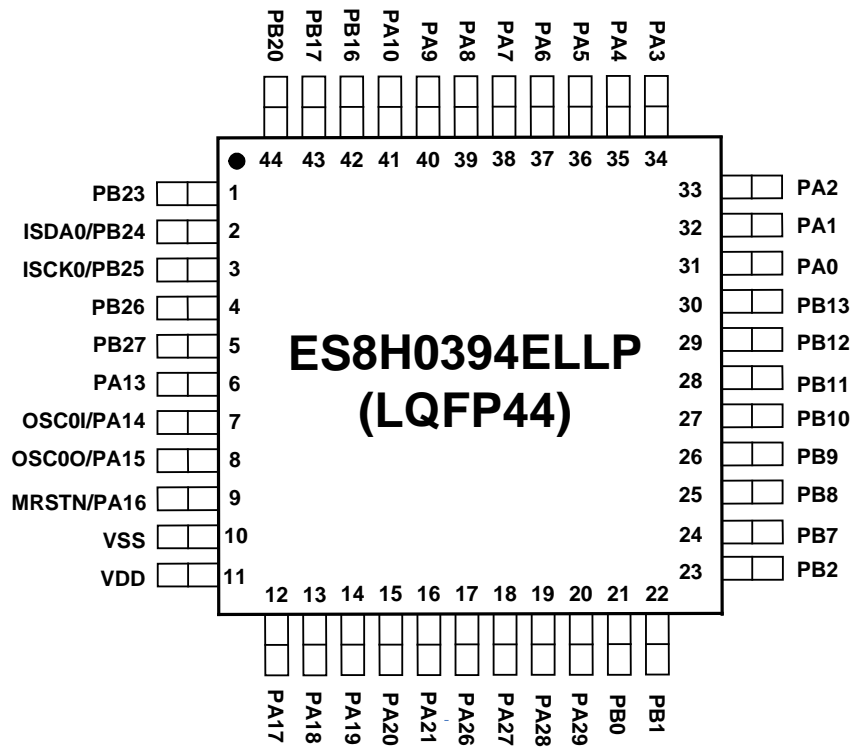


图 1-4 ES8H0394ELLP LQFP44 封装顶视图

### 1.4.4 32-pin管脚封装图

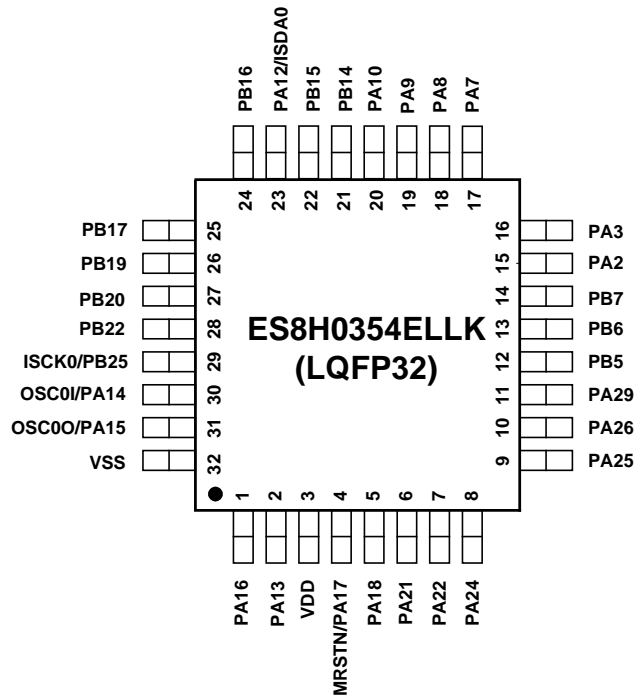


图 1-5 ES8H0354ELLK LQFP32 封装顶视图

- 注 1: 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。芯片共支持 2 组独立 ISP 编程接口 (ISP0 和 ISP1, 其中仅上述封装顶视图中标出的一组 ISP 接口可复用为 SWD 调试接口, 推荐使用该组 ISP 接口。ISP0/SWD 管脚分配详见“ISP0/SWD 管脚对照表”; ISP1 管脚固定为 PA7 (ISCK1) 和 PA8 (ISDA1)。
- 注 2: 若需支持 4 线 ISP 使用, 需将用户配置字 CFG\_PWRTEB 设置为 140ms 复位延时使能, 此时 4 线 ISP 接口配置包括电源线 VDD、地线 VSS、时钟线 ISCK 和数据线 ISDA (即省去 MRSTN 复位线)。
- 注 3: 未使用的 I/O 管脚 (包括产品封装引脚数小于最大引脚数时, 未引出的 I/O 管脚) 都需设置为输出低电平并禁止内部弱上下拉, 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。  
PB28 和 PB29 用于连接内部 EEPROM, 所有芯片型号均未封装引出, 如果使用 EEPROM 功能, 则可通过这两个端口与 EEPROM 进行数据传输, 否则需要将其作为未引出的 I/O 进行设置;  
ES8H0394ELLT 未引出的 I/O 包括 PB28~29。  
ES8H0394ELLP 未引出的 I/O 包括 PA11~12, PA22~25, PA30~31, PB3~6, PB14~15, PB18~19, PB21~22, PB28~29。  
ES8H0354ELLR 未引出的 I/O 包括 PA11~12, PA18, PB16~19, PB22~24, PB26~29。  
ES8H0354ELLK 未引出的 I/O 包括 PA0~1, PA4~6, PA11, PA19~20, PA23, PA27~28, PA30~31, PB0~4, PB8~13, PB18, PB21, PB23~24, PB26~29。
- 注 4: 外部复位与 GPIO 复用同一个管脚, 默认为外部复位 MRSTN 功能, 内部集成了约 55K 欧姆的上拉电阻, 用户可通过芯片配置字 CFG\_WORD0 的 CFG\_MRSTN 位将其设为 GPIO 功能, 当用作 GPIO 功能时, 对芯片复位可靠性没有影响, 该管脚在由复位功能配置切换到 GPIO 之前, 管脚内部上拉电阻为硬件自动使能状态 (在芯片开始工作前, 该管脚会被上拉出现 1~2 个高电平脉冲, 可通过外部输入源将该管脚设置为任意电平, 不会影响 GPIO 功能切换过程), GPIO 功能切换完成后, 内部上拉电阻自动禁止, 对该管脚按与其他 GPIO 管脚一样的方式使用即可。
- 注 5: 本数据手册所述资源为本产品所属产品系列资源最大化描述。文中所述 (包括管脚、寄存器、芯片配置字等) 不为本产品所有的资源, 请将其保持在正确设置值状态。如有其它需求, 请与我司 FAE 部门联系!

## 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
PA0~ PA31	CMOS	CMOS	D	通用 I/O 端口
PB0~PB29	CMOS	CMOS	D	通用 I/O 端口， PB28/PB29 连接内部 EEPROM, 未引出管脚
ISCK0~1	CMOS	—	D	编程/调试串行时钟端口，ISCK0 管脚分配与 芯片型号相关，ISCK1 固定位于 PA7 端口
ISDA0~1	CMOS	CMOS	D	编程/调试串行数据端口，ISDA0 管脚分配与 芯片型号相关，ISDA1 固定位于 PA8 端口
AIN0~AIN17, AIN19~AIN27	—	—	A	ADC 模拟通道 0~17,19~27
TXD0~5	—	CMOS	D	UART0~5 发送输出端口
RXD0~5	CMOS	—	D	UART0~5 接收输入端口
SCK0	CMOS	—	D	SPI0 时钟输入端口
NSS0	CMOS	—	D	SPI0 片选端口
MISO0	CMOS	CMOS	D	SPI0 主控输入/从机输出端口
MOSI0	CMOS	CMOS	D	SPI0 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
T16N0~5_0, T16N0~5_1	CMOS	CMOS	D	T16N0~5 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
AVREFP	—	—	A	ADC 外部正向参考电压（位于 PA10 端口）
SEG0~SEG43	—	—	A	LCD Segment 端口
COM0~COM7	—	—	A	LCD Common 端口
MRSTN	CMOS	—	D	芯片主复位，低电平有效
OSC0I、OSC1I	—	—	A	外部晶体振荡器端口
OSC0O、OSC1O	—	—	A	（每个芯片型号仅支持一组晶振口）
CLKO0	—	CMOS	D	内部时钟输出
CLKO1	—	CMOS	D	内部时钟分频输出
BUZ	—	CMOS	D	内部 BUZ 输出
VDD	—	—	P	系统主电源
VSS	—	—	P	系统地

表 1-1 管脚说明

注：A = 模拟端口，D = 数字端口，P = 电源/地；

## 1.5.2 管脚对照表

### 1.5.2.1 ES8H0394ELLT管脚对照表

PIN No.	PIN NAME	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7	FUNC
0394ELLT	(FUNC0(D))	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(A)
1	PB20	RXD0	T16N0_1	RXD1	RXD4	T16N5_0	CLKO0	SDA0	AIN16/SEG24
2	PB21	RXD2	T16N0_0	TXD1	RXD3	T16N4_1	CLKO1	T16N1_1	AIN17/SEG25
3	PB22	RXD1	MISO0	NSS0	RXD5	T16N4_0	T16N0_0	TXD5	SEG26
4	PB23	T16N1_1	RXD5	SCK0	TXD5	T16N5_0	T16N4_0	TXD2	SEG27
5	PB24	BUZ	T16N5_0	MISO0	RXD1	T16N3_0	T32N0_0	TXD5	—
6	PB25	T16N1_0	T16N5_1	MOSI0	TXD5	SCK0	T16N0_1	RXD1	AIN19
7	PB26	T16N1_1	MISO0	RXD5	T32N0_1	RXD1	T16N4_1	SDA0	SEG30
8	PB27	T16N2_1	MOSI0	TXD5	T32N0_0	RXD3	T16N1_1	SCL0	SEG31
9	PA13	T32N0_1	TXD3	MOSI0	TXD5	BUZ	T16N4_0	SCL0	—
10	PA14	T16N1_1	RXD3	T16N2_0	RXD5	TXD2	T16N0_0	SDA0	OSC0I
11	PA15	TXD5	TXD2	T16N2_1	SDA0	T16N1_0	T16N0_1	RXD2	OSC0O/AIN14
12	MRSTN/PA16	RXD5	RXD2	TXD1	CLKO0	T16N4_1	T32N0_0	SCL0	AIN15
13	VSS	—	—	—	—	—	—	—	—
14	VSS	—	—	—	—	—	—	—	—
15	VDD	—	—	—	—	—	—	—	—
16	NC	—	—	—	—	—	—	—	—
17	PA17	TXD1	T16N0_0	T16N1_0	SCL0	T32N0_0	RXD1	RXD0	COM0
18	PA18	RXD1	SDA0	T16N1_1	SCL0	T16N5_1	TXD0	T32N0_1	COM1
19	PA19	T16N0_0	SCL0	TXD0	RXD3	T16N4_0	NSS0	BUZ	AIN22/COM2
20	PA20	T16N0_1	CLKO1	RXD0	TXD3	T16N4_1	SCK0	BUZ	AIN20/COM3
21	PA21	T32N0_0	TXD4	T16N3_0	T16N0_0	BUZ	RXD5	RXD0	COM4/SEG0
22	PA22	T32N0_1	TXD0	T16N3_1	T16N1_1	TXD3	TXD5	MOSI0	AIN23/SEG33
23	PA23	BUZ	RXD0	T16N1_0	TXD0	RXD3	T16N1_1	SCL0	AIN24/TS/SEG34
24	PA24	T16N1_0	TXD1	MOSI0	TXD0	T16N0_0	RXD2	MISO0	AIN21/SEG35
25	PA25	T16N1_1	RXD1	MISO0	TXD3	T16N3_1	T16N2_0	SCK0	SEG36
26	PA26	T16N2_1	T16N3_0	SCK0	TXD1	TXD3	BUZ	RXD0	COM5/SEG1
27	PA27	TXD2	T16N3_1	NSS0	RXD3	SDA0	T16N2_0	MOSI0	COM6/SEG2
28	PA28	RXD2	TXD2	T16N2_0	T32N0_0	T16N1_1	RXD3	SDA0	COM7/SEG3
29	PA29	NSS0	TXD3	T16N2_1	RXD2	MOSI0	T32N0_1	SCL0	SEG40
30	PA30	T16N0_0	RXD3	SDA0	T16N2_0	TXD5	T32N0_0	TXD4	AIN25/SEG37
31	PA31	T16N0_1	BUZ	SCL0	T16N1_0	TXD3	T16N5_0	MOSI0	SEG29
32	PB0	TXD0	RXD0	T32N0_0	SDA0	T16N4_1	T16N5_1	RXD1	SEG41
33	PB1	RXD4	TXD0	T32N0_1	SCK0	T16N2_1	TXD5	SCL0	SEG42

PIN No.	PIN NAME	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7	FUNC
0394ELLT	(FUNC0(D))	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(A)
34	PB2	T32N0_0	RXD1	T16N3_0	T16N4_1	T16N5_0	RXD0	TXD1	SEG43
35	PB3	T32N0_1	TXD1	T16N3_1	TXD2	T16N0_1	BUZ	RXD1	SEG38
36	PB4	T16N0_0	NSS0	SCL0	RXD2	T16N5_1	T32N0_0	RXD1	SEG39
37	PB5	T16N0_1	SCK0	SDA0	T16N5_0	TXD2	T32N0_1	TXD1	AIN0/SEG32
38	PB6	T16N1_0	MISO0	RXD2	T32N0_1	TXD4	BUZ	SCL0	AIN1/SEG13
39	PB7	T16N1_1	MOSI0	TXD2	T32N0_0	TXD3	T16N3_1	SDA0	AIN2/SEG4
40	PB8	BUZ	RXD3	T16N2_0	TXD4	RXD2	T16N3_0	T16N0_0	SEG5
41	PB9	BUZ	TXD3	T16N2_1	MOSI0	T32N0_1	T16N4_0	RXD4	SEG6
42	PB10	RXD4	NSS0	T16N5_0	TXD1	T16N0_1	SCL0	TXD4	SEG7
43	PB11	TXD4	SCK0	T16N5_1	RXD1	T16N4_1	SDA0	RXD5	SEG8
44	PB12	RXD5	MISO0	T32N0_1	T16N0_0	T16N1_1	TXD4	TXD5	SEG9
45	PB13	TXD5	MOSI0	SCL0	T16N2_1	T32N0_0	RXD5	TXD3	SEG10
46	PA0	RXD4	T16N3_0	BUZ	T16N5_1	T32N0_0	T16N4_1	TXD4	SEG11
47	PA1	TXD4	T16N3_1	BUZ	TXD5	T32N0_1	T16N4_0	MOSI0	SEG12
48	PA2	RXD5	T32N0_0	RXD3	TXD0	T16N0_1	CLKO0	RXD1	AIN3
49	PA3	TXD5	T32N0_1	TXD3	RXD2	T16N2_0	TXD0	SCK0	AIN4
50	PA4	SCL0	RXD2	TXD5	RXD3	T16N2_0	RXD0	T32N0_0	AIN5/SEG15
51	PA5	SDA0	TXD2	MISO0	TXD0	T16N5_1	T32N0_1	RXD5	AIN6/SEG16
52	PA6	SDA0	T16N0_0	RXD1	T16N5_0	TXD1	TXD3	T16N1_1	AIN7/SEG17
53	PA7	T16N2_0	T16N0_1	TXD1	T16N1_1	RXD4	RXD3	NSS0	AIN8/SEG18
54	PA8	RXD0	T16N1_0	BUZ	MISO0	T32N0_1	TXD3	T16N2_0	AIN9/SEG19
55	PA9	TXD0	T16N1_1	CLKO0	TXD2	T16N5_0	MISO0	T16N2_1	AIN10/SEG20
56	PA10	T16N2_0	RXD4	NSS0	RXD2	T16N4_0	SCK0	TXD4	AIN11/AVREFF/ SEG21
57	PB14	T16N2_1	RXD3	TXD4	T16N3_0	RXD1	NSS0	RXD4	AIN26/SEG22
58	PB15	MOSI0	TXD3	CLKO0	T16N3_1	TXD1	T16N1_0	SDA0	AIN27/SEG23
59	PA11	T16N2_1	TXD4	SCK0	T16N1_0	RXD3	SCL0	MISO0	AIN12
60	PA12	T32N0_0	BUZ	MISO0	T16N1_1	TXD3	RXD0	T16N0_0	AIN13
61	PB16	T32N0_1	TXD5	NSS0	RXD0	T16N0_0	T16N3_0	SDA0	SEG14
62	PB17	T32N0_0	RXD5	SCK0	TXD0	T16N4_1	T16N3_1	SCL0	—
63	PB18	SDA0	T16N0_1	MISO0	BUZ	RXD4	T32N0_0	TXD4	—
64	PB19	SCL0	T16N0_0	MOSI0	BUZ	TXD0	T16N5_0	RXD4	SEG28
/	PB28	SCL0	SDA0	MOSI0	BUZ	TXD0	T16N5_0	RXD4	—
/	PB29	SCL0	SDA0	MOSI0	BUZ	TXD0	T16N5_0	RXD4	—

表 1-2 ES8H0394ELLT 管脚对照表

注：PB28 和 PB29 未封装引出，此处仅为对所有端口进行完整描述。

**1.5.2.2 ES8H0354ELLR管脚对照表**

PIN No.	PIN NAME	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7	FUNC
0354ELLR	(FUNC0(D))	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(A)
1	PA13	T32N0_1	TXD3	MOSI0	TXD5	BUZ	T16N4_0	SCL0	—
2	PA16	RXD5	RXD2	TXD1	CLKO0	T16N4_1	T32N0_0	SCL0	AIN15
3	VDD	—	—	—	—	—	—	—	—
4	MRSTN/PA17	TXD1	T16N0_0	T16N1_0	SCL0	T32N0_0	RXD1	RXD0	—
5	PA19	T16N0_0	SCL0	TXD0	RXD3	T16N4_0	NSS0	BUZ	AIN22
6	PA20	T16N0_1	CLKO1	RXD0	TXD3	T16N4_1	SCK0	BUZ	AIN20
7	PA21	T32N0_0	TXD4	T16N3_0	T16N0_0	BUZ	RXD5	RXD0	—
8	PA22	T32N0_1	TXD0	T16N3_1	T16N1_1	TXD3	TXD5	MOSI0	AIN23
9	PA23	BUZ	RXD0	T16N1_0	TXD0	RXD3	T16N1_1	SCL0	AIN24
10	PA24	T16N1_0	TXD1	MOSI0	TXD0	T16N0_0	RXD2	MISO0	AIN21
11	PA25	T16N1_1	RXD1	MISO0	TXD3	T16N3_1	T16N2_0	SCK0	—
12	PA26	T16N2_1	T16N3_0	SCK0	TXD1	TXD3	BUZ	RXD0	—
13	PA27	TXD2	T16N3_1	NSS0	RXD3	SDA0	T16N2_0	MOSI0	—
14	PA28	RXD2	TXD2	T16N2_0	T32N0_0	T16N1_1	RXD3	SDA0	—
15	PA29	NSS0	TXD3	T16N2_1	RXD2	MOSI0	T32N0_1	SCL0	—
16	PA30	T16N0_0	RXD3	SDA0	T16N2_0	TXD5	T32N0_0	TXD4	AIN25
17	PA31	T16N0_1	BUZ	SCL0	T16N1_0	TXD3	T16N5_0	MOSI0	—
18	PB0	TXD0	RXD0	T32N0_0	SDA0	T16N4_1	T16N5_1	RXD1	—
19	PB1	RXD4	TXD0	T32N0_1	SCK0	T16N2_1	TXD5	SCL0	—
20	NC	—	—	—	—	—	—	—	—
21	PB2	T32N0_0	RXD1	T16N3_0	T16N4_1	T16N5_0	RXD0	TXD1	—
22	PB3	T32N0_1	TXD1	T16N3_1	TXD2	T16N0_1	BUZ	RXD1	—
23	PB4	T16N0_0	NSS0	SCL0	RXD2	T16N5_1	T32N0_0	RXD1	—
24	PB5	T16N0_1	SCK0	SDA0	T16N5_0	TXD2	T32N0_1	TXD1	AIN0
25	PB6	T16N1_0	MISO0	RXD2	T32N0_1	TXD4	BUZ	SCL0	AIN1
26	PB7	T16N1_1	MOSI0	TXD2	T32N0_0	TXD3	T16N3_1	SDA0	AIN2
27	PB8	BUZ	RXD3	T16N2_0	TXD4	RXD2	T16N3_0	T16N0_0	—
28	PB9	BUZ	TXD3	T16N2_1	MOSI0	T32N0_1	T16N4_0	RXD4	—
29	PB10	RXD4	NSS0	T16N5_0	TXD1	T16N0_1	SCL0	TXD4	—
30	PB11	TXD4	SCK0	T16N5_1	RXD1	T16N4_1	SDA0	RXD5	—
31	PB12	RXD5	MISO0	T32N0_1	T16N0_0	T16N1_1	TXD4	TXD5	—
32	PB13	TXD5	MOSI0	SCL0	T16N2_1	T32N0_0	RXD5	TXD3	—
33	NC	—	—	—	—	—	—	—	—
34	PA0	RXD4	T16N3_0	BUZ	T16N5_1	T32N0_0	T16N4_1	TXD4	—
35	PA1	TXD4	T16N3_1	BUZ	TXD5	T32N0_1	T16N4_0	MOSI0	—
36	PA2	RXD5	T32N0_0	RXD3	TXD0	T16N0_1	CLKO0	RXD1	AIN3



PIN No.	PIN NAME	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7	FUNC
0354ELLR	(FUNC0(D))	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(A)
37	PA3	TXD5	T32N0_1	TXD3	RXD2	T16N2_0	TXD0	SCK0	AIN4
38	PA4	SCL0	RXD2	TXD5	RXD3	T16N2_0	RXD0	T32N0_0	AIN5
39	PA5	SDA0	TXD2	MISO0	TXD0	T16N5_1	T32N0_1	RXD5	AIN6
40	PB15	MOSI0	TXD3	CLK00	T16N3_1	TXD1	T16N1_0	SDA0	AIN27
41	PB14	T16N2_1	RXD3	TXD4	T16N3_0	RXD1	NSS0	RXD4	AIN26
42	PA10	T16N2_0	RXD4	NSS0	RXD2	T16N4_0	SCK0	TXD4	AIN11/AVREFP
43	PA9	TXD0	T16N1_1	CLK00	TXD2	T16N5_0	MISO0	T16N2_1	AIN10
44	PA8	RXD0	T16N1_0	BUZ	MISO0	T32N0_1	TXD3	T16N2_0	AIN9
45	PA7	T16N2_0	T16N0_1	TXD1	T16N1_1	RXD4	RXD3	NSS0	AIN8
46	PB20	RXD0	T16N0_1	RXD1	RXD4	T16N5_0	CLK00	SDA0	AIN16
47	PA6	SDA0	T16N0_0	RXD1	T16N5_0	TXD1	TXD3	T16N1_1	AIN7
48	PB21	RXD2	T16N0_0	TXD1	RXD3	T16N4_1	CLK01	T16N1_1	AIN17
49	PB25	T16N1_0	T16N5_1	MOSI0	TXD5	SCK0	T16N0_1	RXD1	AIN19
50	PA14	T16N1_1	RXD3	T16N2_0	RXD5	TXD2	T16N0_0	SDA0	OSC0I
51	PA15	TXD5	TXD2	T16N2_1	SDA0	T16N1_0	T16N0_1	RXD2	OSC0O/AIN14
52	VSS	—	—	—	—	—	—	—	—

表 1-3 ES8H0354ELLR 管脚对照表

**1.5.2.3 ES8H0394ELLP管脚对照表**

PIN No.	PIN NAME	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7	FUNC
0394ELLP	(FUNC0(D))	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(A)
1	PB23	T16N1_1	RXD5	SCK0	TXD5	T16N5_0	T16N4_0	TXD2	SEG27
2	PB24	BUZ	T16N5_0	MISO0	RXD1	T16N3_0	T32N0_0	TXD5	—
3	PB25	T16N1_0	T16N5_1	MOSI0	TXD5	SCK0	T16N0_1	RXD1	AIN19
4	PB26	T16N1_1	MISO0	RXD5	T32N0_1	RXD1	T16N4_1	SDA0	SEG30
5	PB27	T16N2_1	MOSI0	TXD5	T32N0_0	RXD3	T16N1_1	SCL0	SEG31
6	PA13	T32N0_1	TXD3	MOSI0	TXD5	BUZ	T16N4_0	SCL0	—
7	PA14	T16N1_1	RXD3	T16N2_0	RXD5	TXD2	T16N0_0	SDA0	OSC0I
8	PA15	TXD5	TXD2	T16N2_1	SDA0	T16N1_0	T16N0_1	RXD2	OSC0O/AIN14
9	MRSTN/PA16	RXD5	RXD2	TXD1	CLKO0	T16N4_1	T32N0_0	SCL0	AIN15
10	VSS	—	—	—	—	—	—	—	—
11	VDD	—	—	—	—	—	—	—	—
12	PA17	TXD1	T16N0_0	T16N1_0	SCL0	T32N0_0	RXD1	RXD0	COM0
13	PA18	RXD1	SDA0	T16N1_1	SCL0	T16N5_1	TXD0	T32N0_1	COM1
14	PA19	T16N0_0	SCL0	TXD0	RXD3	T16N4_0	NSS0	BUZ	AIN22/COM2
15	PA20	T16N0_1	CLKO1	RXD0	TXD3	T16N4_1	SCK0	BUZ	AIN20/COM3
16	PA21	T32N0_0	TXD4	T16N3_0	T16N0_0	BUZ	RXD5	RXD0	COM4/SEG0
17	PA26	T16N2_1	T16N3_0	SCK0	TXD1	TXD3	BUZ	RXD0	COM5/SEG1
18	PA27	TXD2	T16N3_1	NSS0	RXD3	SDA0	T16N2_0	MOSI0	COM6/SEG2
19	PA28	RXD2	TXD2	T16N2_0	T32N0_0	T16N1_1	RXD3	SDA0	COM7/SEG3
20	PA29	NSS0	TXD3	T16N2_1	RXD2	MOSI0	T32N0_1	SCL0	SEG40
21	PB0	TXD0	RXD0	T32N0_0	SDA0	T16N4_1	T16N5_1	RXD1	SEG41
22	PB1	RXD4	TXD0	T32N0_1	SCK0	T16N2_1	TXD5	SCL0	SEG42
23	PB2	T32N0_0	RXD1	T16N3_0	T16N4_1	T16N5_0	RXD0	TXD1	SEG43
24	PB7	T16N1_1	MOSI0	TXD2	T32N0_0	TXD3	T16N3_1	SDA0	AIN2/SEG4
25	PB8	BUZ	RXD3	T16N2_0	TXD4	RXD2	T16N3_0	T16N0_0	SEG5
26	PB9	BUZ	TXD3	T16N2_1	MOSI0	T32N0_1	T16N4_0	RXD4	SEG6
27	PB10	RXD4	NSS0	T16N5_0	TXD1	T16N0_1	SCL0	TXD4	SEG7
28	PB11	TXD4	SCK0	T16N5_1	RXD1	T16N4_1	SDA0	RXD5	SEG8
29	PB12	RXD5	MISO0	T32N0_1	T16N0_0	T16N1_1	TXD4	TXD5	SEG9
30	PB13	TXD5	MOSI0	SCL0	T16N2_1	T32N0_0	RXD5	TXD3	SEG10
31	PA0	RXD4	T16N3_0	BUZ	T16N5_1	T32N0_0	T16N4_1	TXD4	SEG11
32	PA1	TXD4	T16N3_1	BUZ	TXD5	T32N0_1	T16N4_0	MOSI0	SEG12
33	PA2	RXD5	T32N0_0	RXD3	TXD0	T16N0_1	CLKO0	RXD1	AIN3
34	PA3	TXD5	T32N0_1	TXD3	RXD2	T16N2_0	TXD0	SCK0	AIN4
35	PA4	SCL0	RXD2	TXD5	RXD3	T16N2_0	RXD0	T32N0_0	AIN5/SEG15

PIN No.	PIN NAME	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7	FUNC
0394ELLP	(FUNC0(D))	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(A)
<b>36</b>	<b>PA5</b>	SDA0	TXD2	MISO0	TXD0	T16N5_1	T32N0_1	RXD5	AIN6/SEG16
<b>37</b>	<b>PA6</b>	SDA0	T16N0_0	RXD1	T16N5_0	TXD1	TXD3	T16N1_1	AIN7/SEG17
<b>38</b>	<b>PA7</b>	T16N2_0	T16N0_1	TXD1	T16N1_1	RXD4	RXD3	NSS0	AIN8/SEG18
<b>39</b>	<b>PA8</b>	RXD0	T16N1_0	BUZ	MISO0	T32N0_1	TXD3	T16N2_0	AIN9/SEG19
<b>40</b>	<b>PA9</b>	TXD0	T16N1_1	CLK00	TXD2	T16N5_0	MISO0	T16N2_1	AIN10/SEG20
<b>41</b>	<b>PA10</b>	T16N2_0	RXD4	NSS0	RXD2	T16N4_0	SCK0	TXD4	AIN11/AVREFP/ SEG21
<b>42</b>	<b>PB16</b>	T32N0_1	TXD5	NSS0	RXD0	T16N0_0	T16N3_0	SDA0	SEG14
<b>43</b>	<b>PB17</b>	T32N0_0	RXD5	SCK0	TXD0	T16N4_1	T16N3_1	SCL0	—
<b>44</b>	<b>PB20</b>	RXD0	T16N0_1	RXD1	RXD4	T16N5_0	CLK00	SDA0	AIN16/SEG24

表 1-4 ES8H0394ELLP 管脚对照表

1.5.2.4 ES8H0354ELLK管脚对照表

PIN No.	PIN NAME	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7	FUNC
0354ELLK	(FUNC0(D))	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(A)
1	PA16	RXD5	RXD2	TXD1	CLKO0	T16N4_1	T32N0_0	SCL0	AIN15
2	PA13	T32N0_1	TXD3	MOSI0	TXD5	BUZ	T16N4_0	SCL0	—
3	VDD	—	—	—	—	—	—	—	—
4	MRSTN/PA17	TXD1	T16N0_0	T16N1_0	SCL0	T32N0_0	RXD1	RXD0	—
5	PA18	RXD1	SDA0	T16N1_1	SCL0	T16N5_1	TXD0	T32N0_1	—
6	PA21	T32N0_0	TXD4	T16N3_0	T16N0_0	BUZ	RXD5	RXD0	—
7	PA22	T32N0_1	TXD0	T16N3_1	T16N1_1	TXD3	TXD5	MOSI0	AIN23
8	PA24	T16N1_0	TXD1	MOSI0	TXD0	T16N0_0	RXD2	MISO0	AIN21
9	PA25	T16N1_1	RXD1	MISO0	TXD3	T16N3_1	T16N2_0	SCK0	—
10	PA26	T16N2_1	T16N3_0	SCK0	TXD1	TXD3	BUZ	RXD0	—
11	PA29	NSS0	TXD3	T16N2_1	RXD2	MOSI0	T32N0_1	SCL0	—
12	PB5	T16N0_1	SCK0	SDA0	T16N5_0	TXD2	T32N0_1	TXD1	AIN0
13	PB6	T16N1_0	MISO0	RXD2	T32N0_1	TXD4	BUZ	SCL0	AIN1
14	PB7	T16N1_1	MOSI0	TXD2	T32N0_0	TXD3	T16N3_1	SDA0	AIN2
15	PA2	RXD5	T32N0_0	RXD3	TXD0	T16N0_1	CLKO0	RXD1	AIN3
16	PA3	TXD5	T32N0_1	TXD3	RXD2	T16N2_0	TXD0	SCK0	AIN4
17	PA7	T16N2_0	T16N0_1	TXD1	T16N1_1	RXD4	RXD3	NSS0	AIN8
18	PA8	RXD0	T16N1_0	BUZ	MISO0	T32N0_1	TXD3	T16N2_0	AIN9
19	PA9	TXD0	T16N1_1	CLKO0	TXD2	T16N5_0	MISO0	T16N2_1	AIN10
20	PA10	T16N2_0	RXD4	NSS0	RXD2	T16N4_0	SCK0	TXD4	AIN11/AVREFF
21	PB14	T16N2_1	RXD3	TXD4	T16N3_0	RXD1	NSS0	RXD4	AIN26
22	PB15	MOSI0	TXD3	CLKO0	T16N3_1	TXD1	T16N1_0	SDA0	AIN27
23	PA12	T32N0_0	BUZ	MISO0	T16N1_1	TXD3	RXD0	T16N0_0	AIN13
24	PB16	T32N0_1	TXD5	NSS0	RXD0	T16N0_0	T16N3_0	SDA0	—
25	PB17	T32N0_0	RXD5	SCK0	TXD0	T16N4_1	T16N3_1	SCL0	—
26	PB19	SCL0	T16N0_0	MOSI0	BUZ	TXD0	T16N5_0	RXD4	—
27	PB20	RXD0	T16N0_1	RXD1	RXD4	T16N5_0	CLKO0	SDA0	AIN16
28	PB22	RXD1	MISO0	NSS0	RXD5	T16N4_0	T16N0_0	TXD5	—
29	PB25	T16N1_0	T16N5_1	MOSI0	TXD5	SCK0	T16N0_1	RXD1	AIN19
30	PA14	T16N1_1	RXD3	T16N2_0	RXD5	TXD2	T16N0_0	SDA0	OSC0I
31	PA15	TXD5	TXD2	T16N2_1	SDA0	T16N1_0	T16N0_1	RXD2	OSC0O/AIN14
32	VSS	—	—	—	—	—	—	—	—

表 1-5 ES8H0354ELLK 管脚对照表

### 1.5.2.5 ISP0/SWD管脚对照表

型号	ISCK0	ISDA0	SWCLK	SWDIO
ES8H0394ELLT	PB25	PB24	PB25	PB24
ES8H0354ELLR	PB25	PA30	PB25	PA30
ES8H0394ELLP	PB25	PB24	PB25	PB24
ES8H0354ELLK	PB25	PA12	PB25	PA12

表 1-6 ISP0/SWD 管脚对照表

注 1: FUN0~7(D)表示数字端口; FUN(A)表示模拟端口。

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/T16N4\_0/T16N4\_1/T16N5\_0/T16N5\_1 和 T32N0\_0/T32N0\_1 都指其对应的三个复用输入或输出信号。

注 3: 表中的 FUN(A)为模拟端口功能, 不通过 GPIO\_PAFUNCx/GPIO\_PBFUNCx 控制寄存器选取。

注 4: 对于某一确定的芯片型号, 只有一个 MRSTN 端口可用(部分芯片型号可通过用户配置字将 PB25 配置为 MRSTN), 同时可通过芯片配置字 CFG\_WORD0 将用作 MRSTN 的管脚配置为 GPIO 功能, 当用作 GPIO 功能时, 对芯片复位可靠性没有影响, 按与其他 GPIO 管脚一样的方式使用即可。

注 5: 支持 2 组编程接口, 1 组调试接口: ISCK0/ISDA0 根据不同芯片型号位于不同的 IO 端口, ISCK1/ISDA1 固定位于 PA7/PA8 端口。

注 6: 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 PA28 端口复用为 RXD2 时, 可以设置 PA27, PA15, PA5 和 PB7 端口中的任意一个复用为 TXD2, 作为 UART2 模块的通信管脚。

注 7: 支持 1 组外部晶振端口, 采用 PA14、PA15 做外部晶振端口。

### 1.5.2.6 MCU和内置EEPROM的内部连接关系

内置 EEPROM 是一款带 I2C 通信接口的数据存储器芯片, MCU 可按照 I2C 通信协议访问该芯片, 二者的连接关系如下表所示:

内置 EEPROM 引脚	MCU 内部引脚	说明
SDA (I/O)	PB29 (I/O)	I2C 数据线, 需软件使能 PB29 的内部上拉电阻
SCL (I)	PB28 (O)	I2C 时钟线
VDD	VDD	与 MCU 共用电源端口
VSS	VSS	与 MCU 共用地端口

注 1: I = 数字输入, O = 数字输出, I/O = 数字输入/输出;

注 2: MCU 与内部 EEPROM 通信时, PB28 和 PB29 可分别复用为 I2C 模块的 SCL0 和 SDA0 功能, EEPROM 的 SDA 引脚为开漏输出, 所以在 I2C 通信时必须使能 PB29 引脚的内部上拉电阻, 并建议通信速率不超过 400KHz, 以保证 MCU 与 EEPROM 之间通信可靠。

## 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

系统设置保护寄存器 (SCU_PROT)															
偏移地址: 00H															
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	—	对 SCU_PROT<31:0>写 0x55AA6996 时，位 PROT 为 0； 写其它值时位 PROT 为 1
PROT	bit0	R	<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\_WAKEUPTIME, SCU\_DBGHALT, SCU\_FLASHWAIT, SCU\_SOFTCFG, SCU\_LVDCON, SCU\_CCM, SCU\_PLCLKCON, SCU\_HRCBCON, SCU\_SCLKEN0, SCU\_SCLKEN1, SCU\_PCLKEN0, SCU\_PCLKEN1, SCU\_LCDCLK, SCU\_PRSTEN0, SCU\_PRSTEN1, SCU\_TIMEREN, SCU\_TIMERDIS, SCU\_TBLREMAPEN, SCU\_TBLOFF。

## 2.2 系统电源

### 2.2.1 结构框图

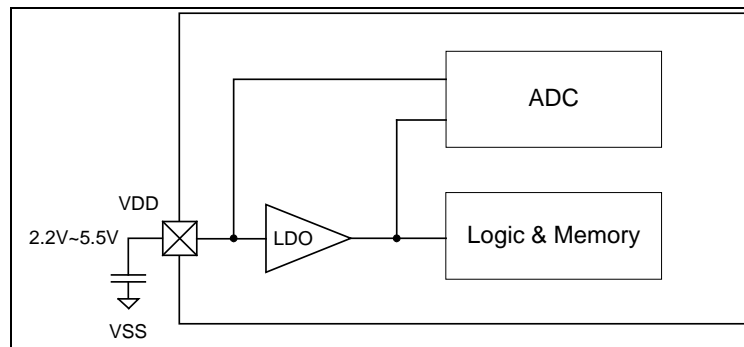


图 2-1 系统电源结构框图

### 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，芯片上电后，LDO 正常工作，无需继续给 LDO 模块（VR）提供时钟，推荐将 SCU\_WAKEUPTIME 寄存器的 VROSCEN 位禁止，可降低功耗。

## 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、WWDT 看门狗复位
- ◇ 支持 Cortex-M0 软件复位
- ◇ 支持 LOCKUP 复位（内核异常复位）

上述 POR 和 BOR 复位会对芯片所有逻辑电路均进行复位；MRSTN 复位对除内核调试电路（包括寄存器 SCU\_DBGHALT）以外的其他逻辑电路进行复位；IWDT/WWDT 复位，Cortex-M0 软件复位和 LOCKUP 复位对除内核调试电路（包括寄存器 SCU\_DBGHALT）和系统时钟控制电路（包括寄存器 SCU\_SCLKEN0, SCU\_SCLKEN1 和 SCU\_WAKEUPTIME）以外的其他逻辑电路进行复位。

通过复位寄存器 SCU\_PWRC，可读取判断各个复位源产生的复位标志，对 Cortex-M0 软件复位，可通过寄存器 SCB\_AIRCR 的 SYSRESETREQ 位置 1 触发，软件复位产生后置起寄存器 SCU\_PWRC 的软件复位标志 SOFT\_RSTF。

### 2.3.2 结构框图

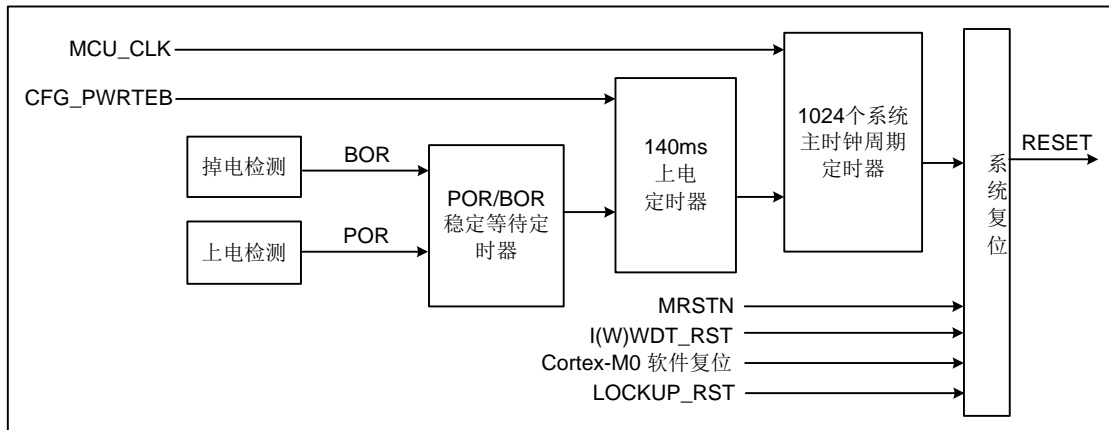


图 2-2 系统复位电路结构框图

- 注 1: 对 140ms 上电延时定时器, 可通过配置位 CFG\_PWRTEB 来配置是否使能。推荐用户设置为上电延时使能 (CFG\_PWRTEB=0), 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。当 MRSTN 管脚复用为 GPIO 功能时, 该延时固定为使能, 与配置位 CFG\_PWRTEB 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, I(W)WDT 复位, LOCKUP 复位或软件复位, 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。
- 注 3: 当不可恢复的异常导致内核锁死时, 将产生 LOCKUP 复位信号来重新启动内核及系统。详细说明可参考 Cortex-M0 技术手册。

### 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。

芯片在 VDD 电压上升过程中, 产生内部 POR 复位, 上升到约 1.8V 释放 POR 复位, 并直到 VDD 电压高于所配置的 BOR 电压点, 芯片退出内部复位状态。

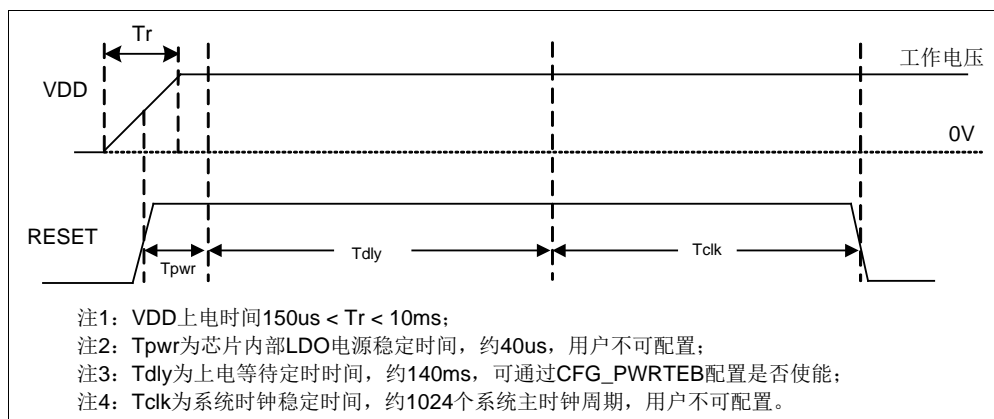


图 2-3 上电复位时序示意图



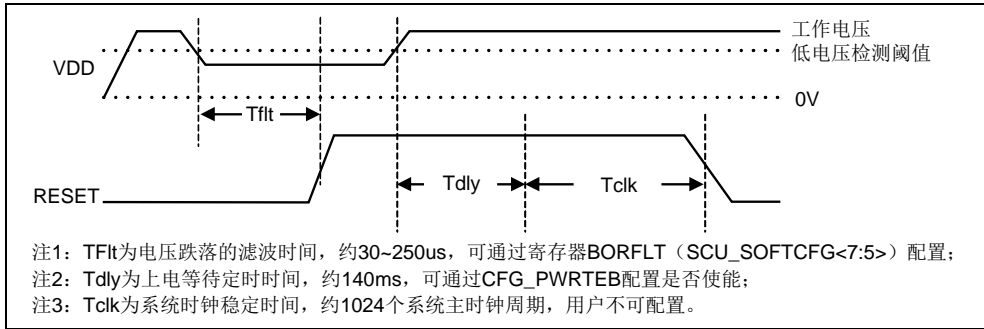


图 2-4 掉电复位时序示意图

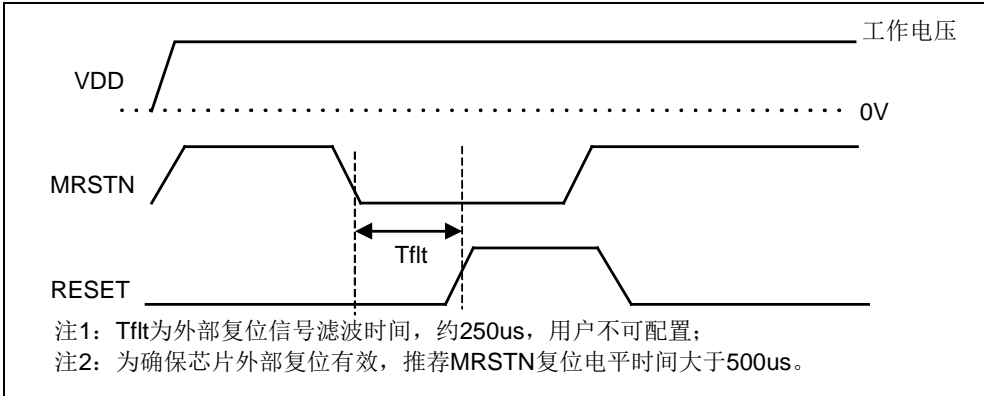


图 2-5 外部复位时序示意图

### 2.3.4 外部复位MRSTN参考

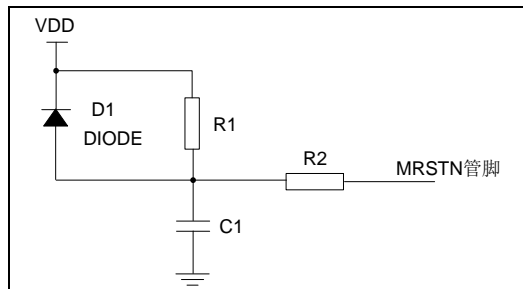


图 2-6 MRSTN 复位参考电路图 1

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

注 2: 对 MRSTN 复位管脚, 芯片内部固定集成了约 55K 欧姆的上拉电阻。

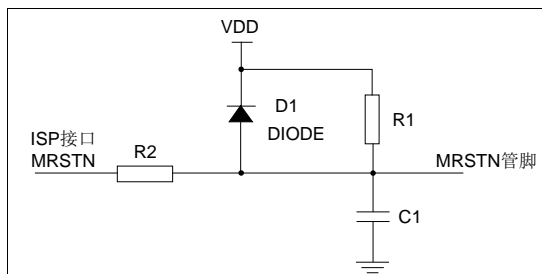


图 2-7 MRSTN 复位参考电路图 2

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

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  $R2$ 。

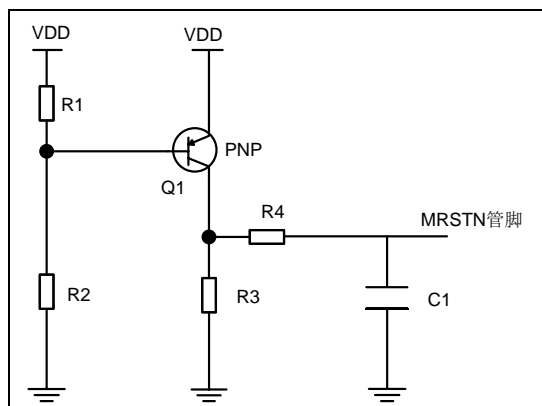


图 2-8 MRSTN 复位参考电路图 3

注：采用 PNP 三极管复位，通过 R1（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（1KΩ）和 C1（0.1μF）接地，C1 另一端作为 MRSTN 输入。

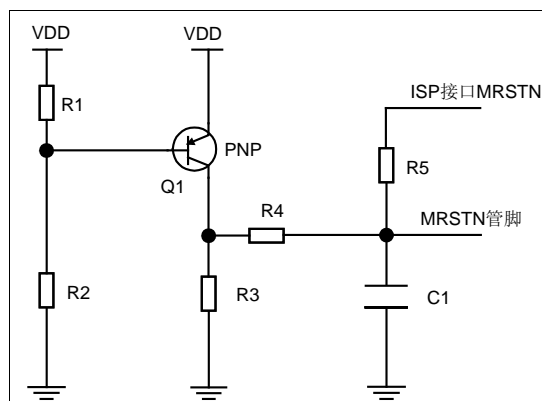


图 2-9 MRSTN 复位参考电路图 4

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 外设模块复位控制

每个外设模块可单独复位，对外设复位控制寄存器 SCU\_PRSTEN0/SCU\_PRSTEN1 的某外设的对应位写入 1，会产生两个 PCLK 周期长度的该外设复位。用户可在使用某一外设模块之前，使用外设复位控制寄存器对其复位初始化，防止状态不确定引起外设工作错误。

## 2.3.6 特殊功能寄存器

### 2.3.6.1 复位寄存器 (SCU\_PWRC)

复位寄存器 (SCU_PWRC)															
偏移地址: 08 <sub>H</sub>															
复位值: 00000000_00000000_0000xxx_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				LKUPRS TF		保留		SOFT_ RSTF	MRST F	IWDT RSTF	WWDT RSTF	BOR F	POR RSTF	POR RCF	POR F

—	bit31-8	—	—
LKUPRSTF	bit10	R/W	<b>LOCKUP 复位标志位</b> 0: 无软件复位 1: 有软件复位
—	bit9-8	—	—
SOFT_RSTF	bit7	R/W	<b>软件复位标志位</b> 0: 无软件复位 1: 有软件复位
MRSTF	bit6	R/W	<b>MRSTN 复位标志位</b> 0: 无 MRSTN 复位 1: 有 MRSTN 复位
IWDTRSTF	bit5	R/W	<b>IWDT 复位标志位</b> 0: 无 IWDT 复位 1: 有 IWDT 复位
WWDTRSTF	bit4	R/W	<b>WWDT 复位标志位</b> 0: 无 WWDT 复位 1: 有 WWDT 复位
BORF	bit3	R/W	<b>BOR 复位标志位</b> 0: 无 BOR 复位 1: 有 BOR 复位
PORRSTF	bit2	R/W	<b>PORRST 复位标志位</b> (内部测试用, 用户无需关心此位) 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> 0: 无 POR 复位 1: 有 POR 复位

- 注 1: 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 2: 标志位 PORRCF 和 PORF 任意一个为 1 时, 均表示芯片发生了上电复位。  
 注 3: 上电后用户必须先对标志位 PORRSTF 作清零操作, 否则即使发生 bit3~bit7, bit10 对应的复位事件, 也无法置起 bit3~bit7, bit10 标志。  
 注 4: 上述复位标志位可通过程序写 0 来清除, 其中 IWDTRSTF 需通过对 bit4 写 0 来清除, WWDTRSTF 需通过对 bit5 写 0 来清除。

### 2.3.6.2 外设复位控制寄存器 0 (SCU\_PRSTEN0)

外设复位控制寄存器 0 (SCU_PRSTEN0)															
偏移地址: 50 <sub>H</sub>															
复位值: 00000000_00000000_00000001_11011110 <sub>B</sub>															
3	3	2	2	2	2	2	24	23	22	21	20	19	18	17	1
1	0	9	8	7	6	5									6
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							WWDT_RS	IWDT_RS	LCD_RS	保	ADC_RS	CRC_RS	IAP_RS	GPIO_RS	保
							T	T	T	留	T	T	T	T	留

—	bit 31-9	—	—
WWDT_RST	bit 8	R/W	<b>WWDT 复位使能位</b> 0: 无复位 1: 使能复位
IWDT_RST	bit 7	R/W	<b>IWDT 复位使能位</b> 0: 无复位 1: 使能复位
LCD_RST	bit 6	R/W	<b>LCD 复位使能位</b> 0: 无复位 1: 使能复位
—	bit 5	—	—
ADC_RST	bit 4	R/W	<b>ADC 复位使能位</b> 0: 无复位 1: 使能复位
CRC_RST	bit 3	R/W	<b>CRC 复位使能位</b> 0: 无复位 1: 使能复位
IAP_RST	bit 2	R/W	<b>FLASH_IAP 复位使能位</b> 0: 无复位 1: 使能复位
GPIO_RST	bit 1	R/W	<b>GPIO 复位使能位</b> 0: 无复位 1: 使能复位
—	bit 0	—	—

注 1: 对 SCU\_PRSTEN0 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: SCU\_PRSTEN0 寄存器中的各个复位使能位, 写 0 时不会产生复位动作, 每次写 1 时会触发一次复位, 读取时该使能位仍保持为 1, 但不会继续触发复位, 直到该使能位被再次写 1。

### 2.3.6.3 外设复位控制寄存器 1 (SCU\_PRSTEN1)

外设复位控制寄存器 1 (SCU_PRSTEN1)															
偏移地址: 54 <sub>H</sub>															
复位值: 00010001_00111111_00000001_00111111 <sub>B</sub>															
3	3	2	28	2	26	2	24	23	22	21	20	19	18	17	16
1	0	9		7		5									
保留	I2C0_		保留	SPI0_R		保留	UART5_	UART4_RS	UART3_RST	UART2_R	UART1	UART0			
	RST			ST			RST	T		ST	_RST	_RST			
1	1	1	12	11	10	9	8	7	6	5	4	3	2	1	0
5	4	3													
			保留				T32N0_	保留	T16N5	T16N4_	T16N3_	T16N2	T16N1	T16N0	
							RST		_RST	RST	RST	_RST	_RST	_RST	

—	bit 31-29	—	—
I2C0_RST	bit28	R/W	<b>I2C0 复位使能位</b> 0: 无复位 1: 使能复位
—	bit 27-25	—	—
SPI0_RST	bit 24	R/W	<b>SPI0 复位使能位</b> 0: 无复位 1: 使能复位
—	bit 23-22	—	—
UART5_RST	bit 21	R/W	<b>UART5 复位使能位</b> 0: 无复位 1: 使能复位
UART4_RST	bit 20	R/W	<b>UART4 复位使能位</b> 0: 无复位 1: 使能复位
UART3_RST	bit 19	R/W	<b>UART3 复位使能位</b> 0: 无复位 1: 使能复位
UART2_RST	bit 18	R/W	<b>UART2 复位使能位</b> 0: 无复位 1: 使能复位
UART1_RST	bit 17	R/W	<b>UART1 复位使能位</b> 0: 无复位 1: 使能复位

UART0_RST	bit 16	R/W	<b>UART0 复位使能位</b> 0: 无复位 1: 使能复位
—	bit 15-9	—	—
T32N0_RST	bit 8	R/W	<b>T32N0 复位使能位</b> 0: 无复位 1: 使能复位
—	bit 7-6	—	—
T16N5_RST	bit 5	R/W	<b>T16N5 复位使能位</b> 0: 无复位 1: 使能复位
T16N4_RST	bit 4	R/W	<b>T16N4 复位使能位</b> 0: 无复位 1: 使能复位
T16N3_RST	bit 3	R/W	<b>T16N3 复位使能位</b> 0: 无复位 1: 使能复位
T16N2_RST	bit 2	R/W	<b>T16N2 复位使能位</b> 0: 无复位 1: 使能复位
T16N1_RST	bit 1	R/W	<b>T16N1 复位使能位</b> 0: 无复位 1: 使能复位
T16N0_RST	bit 0	R/W	<b>T16N0 复位使能位</b> 0: 无复位 1: 使能复位

注 1: 对 SCU\_PRSTEN1 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: SCU\_PRSTEN1 寄存器中的各个复位使能位, 写 0 时不会产生复位动作, 每次写 1 时会触发一次复位, 读取时该使能位仍保持为 1, 但不会继续触发复位, 直到该使能位被再次写 1。

## 2.4 低电压监测 (LVD)

### 2.4.1 概述

LVD 可用于监视 VDD 电源，通过 SCU\_LVDCON 寄存器进行 LVD 电压监测的设置和判断，设置 EN 位使能 LVD，将 VDD 电压和 VS 位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 SCU\_LVDCON 的 IE 位可使能 LVD 中断，当中断标志 IF 位置起时，可触发 LVD 中断请求 IRQ，通过设置 IFS 位可选择 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，置起中断标志位 IF，具体取决于 IFS 位的中断类型配置。该功能的用处之一就是可以在 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。

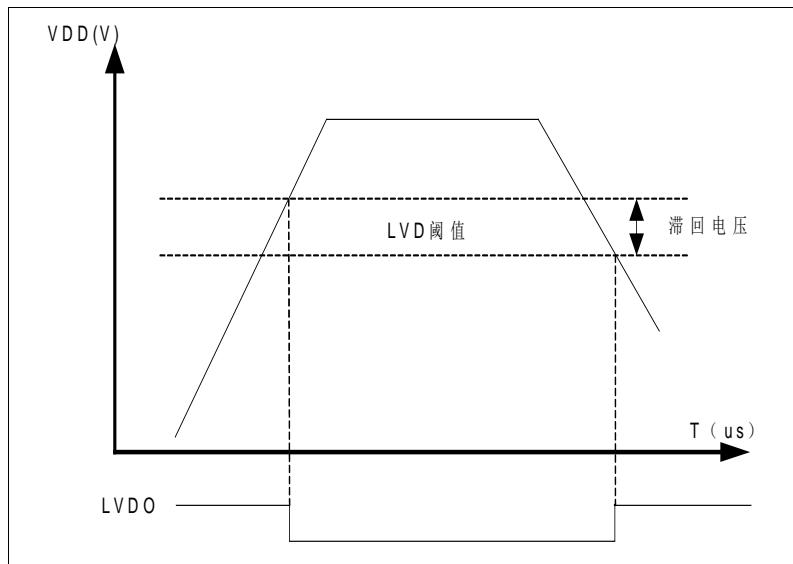


图 2-10 LVD 低电压监测示意图

### 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		LVD 输出状态位 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: 2.3V 0001: 2.4V 0010: 2.7V 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。

注 3: SCU\_LVDCON 寄存器使用时需先使能 EN，再使能 IE；若使能了 FLTEN，且 IFS 设置为 011，则建议在 EN 使能后，等待至少 500us 再使能 IE，避免中断误触发。



## 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\_PCLKEN0/ SCU\_PCLKEN1, 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, 芯片上电后, LDO 正常工作, 无需继续给 LDO 模块(VR)提供时钟, 推荐将 SCU\_WAKEUPTIME 寄存器的 VROSCEN 位禁止, 可降低功耗。

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, 可通过弱上拉或下拉将悬空的输入端口固定为高电平或低电平。如果产品封装引脚数小于最大引脚数, 则未引出的和未使用的 I/O 管脚都需设置为输出低电平并禁止内部弱上下拉。

芯片进入休眠状态后, 时钟工作状态参考下表:

时钟	浅睡眠模式	深度睡眠模式
XTAL	工作 (若 XTAL_EN=1)	工作 (若 XTAL_EN=1 且 MOSC_EN=1)
HRC	工作 (若 HRC_EN=1)	工作 (若 HRC_EN=1 且 MOSC_EN=1)
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，需设置系统唤醒时间控制寄存器 SCU\_WAKEUPTIME 的 FLS\_STOP=1, BG\_STOP=1, LP\_STOP=1, LP\_SEL=01, STPRTNEN=1, VROSCEN=0, 降低深睡眠时 Flash, BG (LDO 的参考基准), LDO 和 SRAM 的功耗，设置深度睡眠模式时钟控制位 MOSC\_EN 来选择主晶振 XTAL、PLL、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 PINTx 可唤醒深度睡眠模式
  - 外部端口中断 KINT 可唤醒深度睡眠模式
  - LVD 中断可唤醒深度睡眠模式
  - IWDT、WWDT 中断可唤醒深度睡眠模式（工作于 LRC 时钟源）
  - ADC 中断可唤醒深度睡眠模式（工作于 LRC 时钟源）
  - 芯片复位可唤醒深度睡眠模式

#### 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 80us，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 HRC 时钟的稳定时间可软件设置： $80us + T_{pclk} * WAKEUPTIME$ （其中  $T_{pclk}$  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 90us，否则芯片唤醒后有可能工作异常。

内部 LDO 电压稳定时间，芯片硬件固定约为 120us，不可软件设置。

例如使用内部 HRC 时钟作为系统时钟时：

时钟控制位 MOSC\_EN=0 时，深度睡眠模式的唤醒时间最短约为  $90us + 120us = 210us$ ；

时钟控制位 MOSC\_EN=1 时，深度睡眠模式的唤醒时间最短约为 120us。

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。

配置 SCU\_FLASHWAIT 寄存器的 ACCT<3:0>，可设定 FLASH 访问的等待时间。

当配置 HS=0 时，为 FLASH 普通访问模式，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：20H															
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											HS	ACCT<3:0>			

—	bit31-4	—	—
HS	bit4	R/W	<b>FLASH 访问模式</b> 0: 普通模式 1: 快速模式（仅供测试用，禁止用户设置为 1）
ACCT<3:0>	bit3-0	R/W	<b>FLASH 读取访问等待时间设置位</b> 0x0: 1T <sub>CLK</sub> 完成 FLASH 读取 0x1: 2T <sub>CLK</sub> 0x2: 3T <sub>CLK</sub> ... 0xF: 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：T<sub>CLK</sub> 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，16MHz 为系统上电默认时钟。
- ◇ 内部低频 RC 时钟源 LRC，支持约 31.25KHz 时钟频率。
- ◇ 内部集成锁相环电路 PLL，其时钟源输入可选，PLL 输出时钟频率支持 32MHz 或 48MHz。
- ◇ 系统时钟支持 1~128 分频。
- ◇ 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。
- ◇ PLL 失锁检测，支持 PLL 失锁后自动切换至使用 PLL 前的时钟源并产生中断。

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\_SCLKEN1 寄存器的 XTAL\_EN=1，等待稳定后设置 SCU\_SCLKEN0 寄存器的 CLK\_SEL=2，选择外部振荡器时钟；设置 SCU\_SCLKEN1 寄存器的 PLL\_EN=0。
2. 系统时钟为内部时钟源 HRC 16MHz：为上电默认系统时钟。设置 SCU\_SCLKEN0 寄存器的 CLK\_SEL=0；设置 SCU\_SCLKEN1 寄存器的 PLL\_EN=0。
3. 系统时钟为内部时钟源 LRC 31.25KHz：设置 SCU\_SCLKEN0 寄存器的 CLK\_SEL=1；设置 SCU\_SCLKEN1 寄存器的 PLL\_EN=0。
4. 系统时钟为内部时钟源 PLL 倍频时钟：设置 SCU\_SCLKEN0 寄存器的 CLK\_SEL=0；设置 SCU\_SCLKEN1 寄存器的 PLL\_REF\_SEL，选择 PLL 输入时钟源，设置 PLL\_48M\_SEL，选择 PLL 输出时钟频率，设置 PLL\_EN=1；设置 SCU\_SCLKEN0 寄存器的 PLL\_MUX=1。

## 2.6.2 结构框图

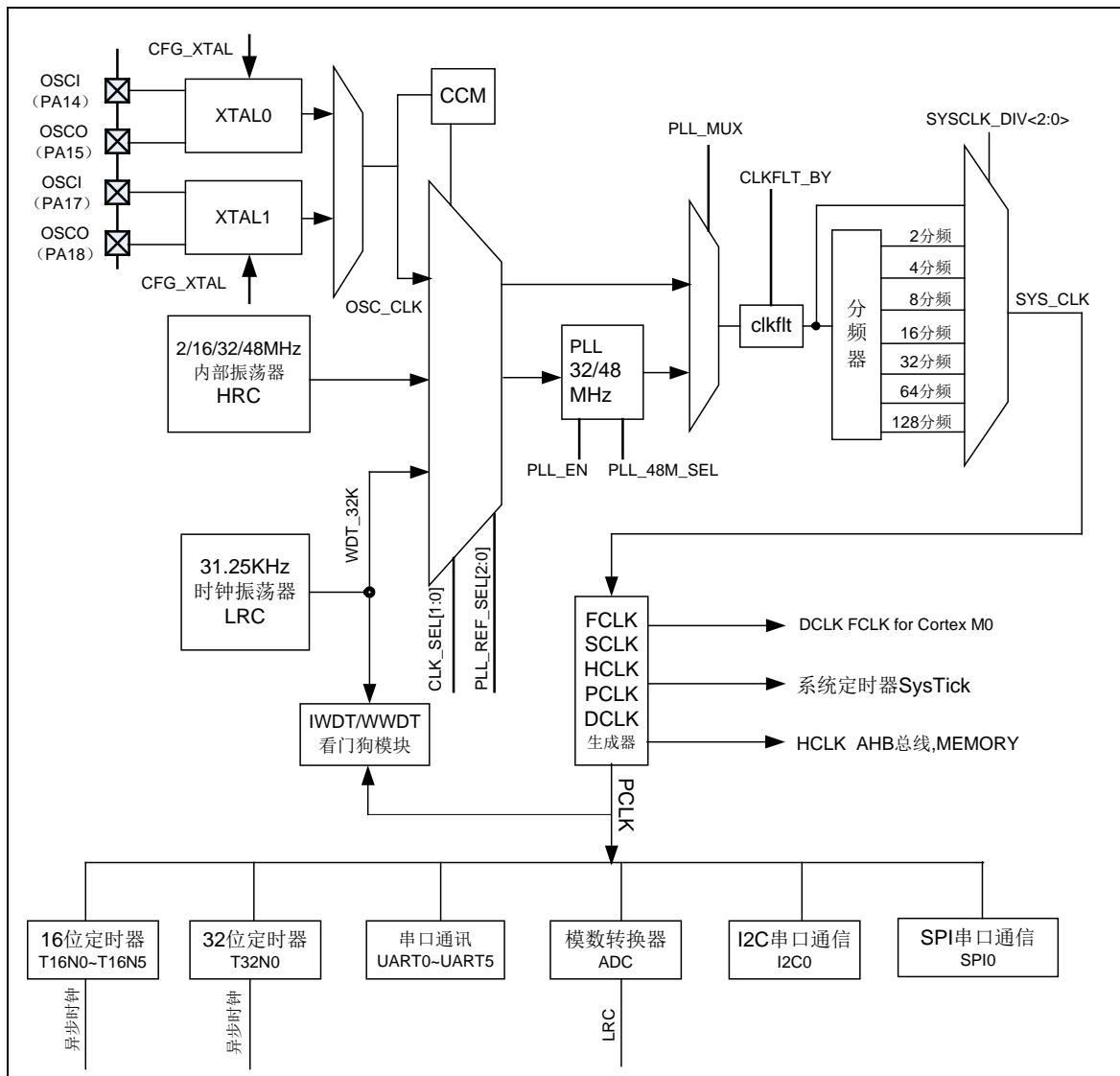


图 2-11 系统时钟电路结构框图

注：每款芯片支持 1 组外部晶振端口，ES8H0394ELLT、ES8H0354ELLR、ES8H0394ELLP 和 ES8H0354ELLK 均采用 PA14、PA15 做外部晶振端口。

## 2.6.3 功能说明

### 2.6.3.1 外部时钟XTAL

可通过 PA14、PA15 管脚外接时钟振荡器。振荡器对应的 IO 需设置为模拟端口（关闭端口的数字输入和输出功能），外部振荡器模块使能后（XTAL\_EN=1），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 振荡器电路结构示意图如下：

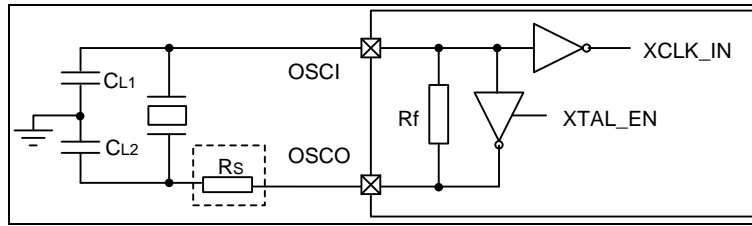


图 2-12 XTAL 振荡器电路结构示意图

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：

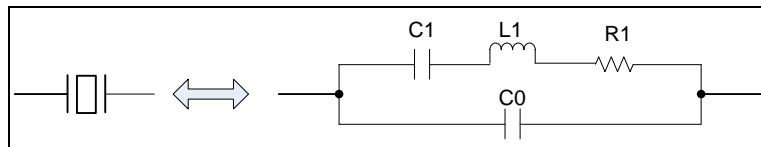


图 2-13 晶体振荡器等效电路示意图

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

芯片使用 HRC 作为内部高速时钟。常温条件下 HRC 频率精度为  $\pm 1\%$ （48MHz 频率精度为  $\pm 0.6\%$ ）。支持 4 种 HRC 时钟频率，分别是 2MHz，16 MHz，32 MHz 和 48 MHz，可通过寄存器 SCU\_SCLKEN1 的 HRC\_SEL 位进行频率选择，当使用 HRC 作为 PLL 时钟源时，只能选择使用 HRC 16MHz，芯片上电后系统时钟默认为内部高速时钟 HRC 16MHz，可通过寄存器 SCU\_SCLKEN1 的 HRC\_RDY 位判断 HRC 时钟是否稳定，可通过寄存器 SCU\_SCLKEN1 的 HRC\_EN 位关闭。

当选择使用 HRC 2MHz, 16MHz 和 32MHz 时, 需通过寄存器 SCU\_HRCBCON 的 HRCBSET<2:0>位将 HRC 时钟偏置系数设置为 111。

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 31.25KHz), 且无法关闭, 始终保持工作。常温条件下 LRC 时钟频率精度约为  $\pm 4\%$ 。内部低速时钟可供芯片主系统、WDT 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 锁相环PLL

PLL 时钟输入可选择 XTAL (32.768KHz、4MHz、8MHz、16MHz、20MHz), HRC (16MHz), LRC (31.25KHz)。

PLL 模块工作时, 通过寄存器位 PLL\_REF\_SEL<2:0>选择其输入时钟源, 同时需对所选用的 HRC, LRC 或 XTAL 时钟振荡器进行对应的正确设置; 通过寄存器位 PLL\_48M\_SEL 选择 PLL 模块的倍频系数, 得到对应的时钟输出频率, 具体见下面的详细描述:

当 PLL\_REF\_SEL= 1 时: PLL 输入时钟源即为 16MHz HRC 时钟的 4 分频时钟。当 PLL\_48M\_SEL=0 时: PLL 倍频系数为 8 倍, PLL 输出时钟为 32MHz。当 PLL\_48M\_SEL=1 时: PLL 倍频系数为 12 倍, PLL 输出时钟为 48MHz。

当 PLL\_REF\_SEL=2 时: PLL 输入时钟源为 LRC。当 PLL\_48M\_SEL=0 时: PLL 倍频系数为 1024 倍, PLL 输出时钟约为 32MHz。当 PLL\_48M\_SEL=1 时: PLL 倍频系数为 1536 倍, PLL 输出时钟约为 48MHz。

当 PLL\_REF\_SEL=3 时: PLL 输入时钟源为 XTAL 32.768KHz。当 PLL\_48M\_SEL=0 时: PLL 倍频系数为 1024 倍, PLL 输出时钟为 33.5544MHz。当 PLL\_48M\_SEL=1 时: PLL 倍频系数为 1536 倍, PLL 输出时钟为 50.3316MHz。

当 PLL\_REF\_SEL=4 时: XTAL 时钟振荡器频率必须为 4MHz, PLL 输入时钟源为 XTAL 时钟。当 PLL\_48M\_SEL=0 时: PLL 倍频系数为 8 倍, PLL 输出时钟为 32MHz。当 PLL\_48M\_SEL=1 时: PLL 倍频系数为 12 倍, PLL 输出时钟为 48MHz。

当 PLL\_REF\_SEL=5 时: XTAL 时钟振荡器频率必须为 8MHz, PLL 输入时钟源为 XTAL 的 2 分频。当 PLL\_48M\_SEL=0 时: PLL 倍频系数为 8 倍, PLL 输出时钟为 32MHz。当 PLL\_48M\_SEL=1 时: PLL 倍频系数为 12 倍, PLL 输出时钟为 48MHz。

当 PLL\_REF\_SEL=6 时: XTAL 时钟振荡器频率必须为 16MHz, PLL 输入时钟源为 XTAL 的 4 分频。当 PLL\_48M\_SEL=0 时: PLL 倍频系数为 8 倍, PLL 输出时钟为 32MHz。当 PLL\_48M\_SEL=1 时: PLL 倍频系数为 12 倍, PLL 输出时钟为 48MHz。

当 PLL\_REF\_SEL=7 时: XTAL 时钟振荡器频率必须为 20MHz, PLL 输入时钟源为 XTAL 的 5 分频。当 PLL\_48M\_SEL=0 时: PLL 倍频系数为 8 倍, PLL 输出时钟为 32MHz。

当 PLL\_48M\_SEL=1 时：PLL 倍频系数为 12 倍，PLL 输出时钟为 48MHz。

PLL 模块使用时，建议关闭 PLL\_BYLOCK，不旁路 PLL 锁频信号，当 PLL 失锁后，系统会自动切换至使用 PLL 前的时钟源（由 CLK\_SEL 决定）并产生中断。如果使能 PLL\_BYLOCK，当 PLL 失锁后，系统仍使用 PLL 时钟，可能会导致系统错误。在芯片进入深度睡眠模式前，如果系统时钟为 PLL 倍频时钟，则必须设置 PLL\_BYLOCK=0，不旁路 PLL 锁频信号，否则会导致睡眠模式唤醒异常。

对 PLL 模块，必须等其所选择的输入时钟源稳定后，才可以使能 PLL 工作(PLL\_EN=1)，当时钟源为 HRC 16MHz 时钟时，可判断寄存器 SCU\_SCLKEN1 的 HRC\_RDY 位，当时钟源为外部时钟时，可判断寄存器 SCU\_SCLKEN1 的 XTAL\_RDY 位，PLL 锁频稳定后，会置起寄存器 SCU\_SCLKEN1 的稳定标志 PLL\_RDY 和寄存器 SCU\_PLLLKCON 的锁频成功标志 LK\_FLAG，配置寄存器 SCU\_PLLLKCON 的 LK\_IFS 位，可选择 PLL 锁频成功或失败中断标志位 IF 的产生方式，通过 PLL 锁频中断使能位 IE，可对 PLL 中断标志 IF 是否触发 PLL 中断请求 IRQ，进行设置。

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

### 2.6.3.5 时钟输出CLKO

芯片支持 2 路 IO 端口输出时钟信号。其中 CLKO0 端口支持内部时钟直接输出，CLKO1 端口支持内部系统时钟 128 分频和 HRC 时钟 512 分频输出。通过 SCU\_SCLKEN0 寄存器的 CLKOUT0\_SEL 和 CLKOUT1\_SEL 控制位选取输出的时钟。使用时需配置相应端口的复用选择寄存器 GPIO\_PAFUNCx/GPIO\_PBFUNCx，使能管脚的时钟输出功能。当使用高频时钟输出时，需使能管脚大电流驱动模式，以免输出时钟波形严重失真。

### 2.6.3.6 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 寄存器的外部停振检测软件使能位 EN=1，缺省为使能；
- (2) 时钟源选择为外部时钟。即 SCU\_SCLKEN0 寄存器的 CLK\_SEL=2。

可通过寄存器 SCU\_CCM 的 CLKS 位，选择外部时钟停振后的备用时钟为 LRC 或 HRC 时钟，当选择为 HRC 时钟时，还需同时使能寄存器 SCU\_SCLKEN1 的 HRC\_EN 位。可通过寄存器 SCU\_CCM 的 MD 位，选择停振检测模式，当选择为慢速时，检测时间长；当选择为快速时，检测时间短，检测灵敏度高，该模式下还需同时使能寄存器 SCU\_SCLKEN1 的 HRC\_EN 位。

外部时钟停振检测开始工作后，当检测到外部时钟停振，系统时钟会自动切换至备用时钟，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。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，并清除 SCU\_CCM 寄存器的外部时钟停振标志 FLAG。

注 1: 使能 CCM 检测后，如果外部时钟停振，在停振期间，不要通过软件进行系统时钟切换，硬件自动使用由寄存器 SCU\_CCM 的 CLKS 位所选择的备用时钟作为系统时钟，直到外部时钟恢复振荡，再硬件自动重新切回外



部时钟。

注 2: 当 PLL 时钟源输入使用的是外部时钟, 但上述 2 个条件有任何一个不满足时, 外部时钟停振检测模块不工作。

注 3: 外部时钟停振检测开始工作后, 当 MOSC\_EN=0 时, 芯片进入深睡眠模式会关闭外部时钟源, 导致外部时钟停振, 此类情况不会触发时钟停振标志。

### 2.6.3.7 时钟滤波CLKFLT

芯片支持系统时钟分频前滤波。

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

当需要关闭滤波时, 必须先旁路 CLKFLT, 即设置 CLKFLT\_BY=0x55, 然后再通过 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。

应用系统中不关闭且不旁路 CLKFLT, 可进一步提升系统抗干扰能力。

### 2.6.3.8 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。

### 2.6.3.9 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。

### 2.6.3.10 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、PLL 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、PLL、CLKFLT 自动恢复到睡眠前的状态。

2. MOSC\_EN=1 时: 时钟源正常运行, PLL 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)。

在芯片进入深度睡眠模式前, 如果系统时钟为 PLL 倍频时钟, 则必须设置 PLL\_BYLOCK=0, 不旁路 PLL 锁频信号, 否则会导致睡眠模式唤醒异常。

## 2.6.4 特殊功能寄存器

### 2.6.4.1 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: 10 <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_STO	保留	BG_STO	LP_STO	保留	LP_SEL<1:0>	保留	
								P		P	P		>		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STPRTNE	VROSCEN	CLKFLT_E	MOSC_E	WAKEUPTIME<11:0>											
N	N	N	N												

—	bit31-24	—	—
FLS_STOP	bit 23	R/W	深度睡眠模式 Flash STOP 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 BG 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 LDO 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	—	—
LP_SEL<1:0>	bit18-17	W/R	在深度睡眠模式下, LDO 电压输出选择位 01: 1.5V (必须设置为本档位) 00, 10, 11: 保留 (仅用于内部测试用)
—	bit16	—	— (该位必须保持默认 0)
STPRTNEN	bit15	R/W	深度睡眠模式 SRAM 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	VR 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	R/W	CLKFLT 系统时钟滤波器使能位 0: 禁止 1: 使能 CLKFLT 为系统时钟滤波器, 使能 CLKFLT, 可进一步提升系统抗干扰能力, 在深度睡眠模式下, 可禁止 CLKFLT, 降低芯片功耗

MOSC_EN	bit12	R/W	<b>深度睡眠模式时钟控制位</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	R/W	<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: 深度睡眠模式下, 当 MOSC\_EN 为 1 时, HRC、PLL、XTAL 和时钟滤波器还必须各自的控制位 HRC\_EN、PLL\_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 3: 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、PLL 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, 设置值需大于等于 0x3FF (使用 XTAL 时钟时推荐设置为 0xFFF), 具体可根据芯片在应用系统中的实际工作状况向上调整, 确保唤醒稳定。
- 注 4: 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 5: 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 0xFFF, 以增强振荡器工作稳定标志位 XTAL\_RDY 的可靠性。

### 2.6.4.2 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: 40H															
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			PLL_MUX		保留					CLK_SEL<1:0>		

—	bit 31-28	—	—
CLKOUT1_SEL<1:0>	bit27-26	R/W	<b>CLKO1 选择位</b> 00: 禁止时钟输出 01: 系统时钟输出 (128 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 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, 不旁路 CLKFLT, 可进一步提升系统抗干扰能力
—	bit15	—	—
SYSCLK_DIV<2:0>	bit14-12	R/W	<b>系统时钟后分频选择位</b> 000: 1:1 001: 1:2 010: 1:4 011: 1:8 100: 1:16 101: 1:32 110: 1:64 111: 1:128 推荐设置为 1:1, 否则会影响 HRC 时钟精度
—	bit11-9	—	—
PLL_MUX	bit8	R/W	<b>时钟倍频选择位</b> 0: 使用原始时钟 (由 CLK_SEL 选择的时钟) 1: 使用倍频时钟
—	bit7-2	—	—
CLK_SEL<1:0>	bit1-0	R/W	<b>原始时钟源选择位</b> 00: HRC 时钟 01: LRC 时钟 10: XTAL 时钟 (由配置位 CFG_XTAL 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 PLL 倍频时钟 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。

注 3: 推荐设置 SYSCLK\_DIV<2:0>=000, 系统时钟后分频为 1:1, 否则可能会影响 HRC 时钟精度, 偏差增大 0.5% 左右。

### 2.6.4.3 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_00001010 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													PLL_RDY	HRC_RDY	XTAL_RDY
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL_BYLOCK	PLL_EN	PLL_48M_SEL	PLL_REF_SEL<2:0>		保留			HRC_SEL<1:0>		保留	HRC_EN	XTAL_EN		

—	bit 31-19	—	—
PLL_RDY	bit18	R	<b>PLL 时钟振荡模式稳定标志位</b> 0: 不稳定 1: 稳定
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
—	bit15-14	—	—
PLL_BYLOCK	bit13	R/W	<b>PLL 锁频信号旁路控制位</b> 0: 不旁路 1: 旁路
PLL_EN	bit12	R/W	<b>PLL 倍频电路工作使能位</b> 0: 禁止 1: 使能 (使能前需确认 PLL_REF_SEL 所选择的时钟源是否稳定)
PLL_48M_SEL	bit11	R/W	<b>PLL 输出时钟选择位</b> 0: 输出频率约 32MHz 时钟 1: 输出频率约 48MHz 时钟
PLL_REF_SEL<2:0>	bit10-8	R/W	<b>PLL 输入时钟源选择位 (仅在 PLL_EN=0 时可写入)</b> 00x: 选择 HRC16MHz 时钟 (其他 HRC 时钟无效) 010: 选择 LRC 时钟 011: 选择 XTAL 时钟 (外挂 32768Hz 晶振) 100: 选择 XTAL 时钟 (外挂 4MHz 晶振) 101: 选择 XTAL 时钟 (外挂 8MHz 晶振) 110: 选择 XTAL 时钟 (外挂 16MHz 晶振) 111: 选择 XTAL 时钟 (外挂 20MHz 晶振) (硬件会根据 PLL 输入时钟设置自动分频或预处理相关时钟, 软件无需再做处理)
—	bit7-5	—	—
HRC_SEL<1:0>	bit4-3	R/W	<b>HRC 频率选择位</b> 00: 2MHz 01: 16MHz 10: 32MHz 11: 48MHz 当选择使用 HRC 2MHz, 16MHz 和 32MHz 时, 需通过寄存器 SCU_HRCBCON 的 HRCBSET<2:0>位将 HRC 时钟偏置系数设置为 111

—	bit2	—	—
HRC_EN	bit1	R/W	内部高速时钟振荡电路控制位 0: 禁止 1: 使能
XTAL_EN	bit0	R/W	外部时钟振荡电路控制位 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: 因 LRC 时钟频率偏差大, 所以如果对 PLL 输出时钟频率精度要求高时, 请勿设置 PLL\_REF\_SEL<2:0>=010, 即不要将 PLL 输入时钟源选择为 LRC 时钟。

注 3: 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。

注 4: 在芯片进入深度睡眠模式前, 如果系统时钟为 PLL 倍频时钟, 则必须设置 PLL\_BYLOCK=0, 不旁路 PLL 锁频信号, 否则会导致睡眠模式唤醒异常。

#### 2.6.4.4 外设时钟控制寄存器 (SCU\_PCLKEN0)

外设时钟控制寄存器 (SCU\_PCLKEN0)

偏移地址: 48H

复位值: 00000000\_00000000\_00000001\_11011111<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							WWDT_EN	IWDT_EN	LCD_EN	保留	ADC_EN	CRC_EN	IAP_EN	GPIO_EN	SCU_EN

—	bit 31-9	—	—
WWDT_EN	bit 8	R/W	WWDT 时钟使能位 0: 禁止 1: 使能
IWDT_EN	bit 7	R/W	IWDT 时钟使能位 0: 禁止 1: 使能
LCD_EN	bit 6	R/W	LCD 时钟使能位 0: 禁止 1: 使能
—	bit 5	—	—
ADC_EN	bit 4	R/W	ADC 时钟使能位 0: 禁止 1: 使能
CRC_EN	bit 3	R/W	CRC 时钟使能位 0: 禁止 1: 使能

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\_PCLKEN0 寄存器 SCU\_EN 位设置为 1。

注 3: IWDT 时钟使能位 IWDT\_EN=0 时, 对 IWDT 模块的寄存器读写操作被禁止, 如果 IWDT 时钟源为 PCLK, 则 IWDT 计数器也停止计数, 如果 IWDT 时钟源为 LRC, 则 IWDT 计数器仍保持工作状态, 看门狗定时器功能仍有效。

注 4: WWDT 时钟使能位 WWDT\_EN=0 时, 对 WWDT 模块的寄存器读写操作被禁止, 如果 WWDT 时钟源为 PCLK, 则 WWDT 计数器也停止计数, 如果 WWDT 时钟源为 LRC, 则 WWDT 计数器仍保持工作状态, 看门狗定时器功能仍有效。

注 5: 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。

#### 2.6.4.5 外设时钟控制寄存器 (SCU\_PCLKEN1)

外设时钟控制寄存器 (SCU\_PCLKEN1)

偏移地址: 4C<sub>H</sub>

复位值: 00010001\_00111111\_00000001\_00111111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	I2C0	保留	SPI0	保留	UART5	UART4	UART3	UART2	UART1	UART0					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			保留				T32N0	保留	T16N5	T16N4	T16N3	T16N2	T16N1	T16N0	

—	bit 31-29	—	—
I2C0_EN	bit28	R/W	<b>I2C0 时钟使能位</b> 0: 禁止 1: 使能
—	bit 27-25	—	—
SPI0_EN	bit 24	R/W	<b>SPI0 时钟使能位</b> 0: 禁止 1: 使能
—	bit 23-22	—	—
UART5_EN	bit 21	R/W	<b>UART5 时钟使能位</b> 0: 禁止 1: 使能

UART4_EN	bit 20	R/W	<b>UART4 时钟使能位</b> 0: 禁止 1: 使能
UART3_EN	bit 19	R/W	<b>UART3 时钟使能位</b> 0: 禁止 1: 使能
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 15-9	—	—
T32N0_EN	bit 8	R/W	<b>T32N0 时钟使能位</b> 0: 禁止 1: 使能
—	bit 7-6	—	—
T16N5_EN	bit 5	R/W	<b>T16N5 时钟使能位</b> 0: 禁止 1: 使能
T16N4_EN	bit 4	R/W	<b>T16N4 时钟使能位</b> 0: 禁止 1: 使能
T16N3_EN	bit 3	R/W	<b>T16N3 时钟使能位</b> 0: 禁止 1: 使能
T16N2_EN	bit 2	R/W	<b>T16N2 时钟使能位</b> 0: 禁止 1: 使能
T16N1_EN	bit 1	R/W	<b>T16N1 时钟使能位</b> 0: 禁止 1: 使能
T16N0_EN	bit 0	R/W	<b>T16N0 时钟使能位</b> 0: 禁止 1: 使能

注 1: 对 SCU\_PCLKEN1 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: 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。



### 2.6.4.6 外设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: 2C <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															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		MD	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: 停振检测 FLAG 上升沿产生中断, 晶振停振 001: 停振检测 FLAG 下降沿产生中断, 晶振恢复振荡 010: 停振检测 FLAG 高电平产生中断, 晶振停振 011: 停振检测 FLAG 低电平产生中断, 晶振恢复振荡 1xx: 停振检测 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: 中断禁止 1: 中断使能
—	bit3	—	—
CLKS	bit2	R/W	<b>外部时钟停振后的备用时钟选择位</b> 0: LRC 1: HRC (HRC_EN 位需使能)
MD	bit1	R/W	<b>外部时钟停振检测模式位</b> 0: 慢速, 检测时间长 (对 32KHz 外部时钟停振检测, 只能设置为慢速) 1: 快速, 检测时间短 (HRC_EN 位需使能)
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;

注 3: 当 MD=0 停振检测为慢速时, 在约 2.5ms 时间内, 如果未检测到外部时钟脉冲, 则判定为外部时钟停振, 当 MD=1 停振检测为快速时, 在约 35us 时间内, 如果未检测到外部时钟脉冲, 则判定为外部时钟停振; 当外部时钟为 32KHz 晶振时, 则只能设置 MD=0, 采用慢速模式进行外部时钟的停振检测; 当外部时钟为 1~20MHz 晶振时, 可以设置为慢速或快速, 但推荐设置 MD=1, 采用快速模式进行外部时钟的停振检测, 可以提升检测灵敏度。

### 2.6.4.7 PLL锁定中断控制寄存器 (SCU\_PLLKCON)

PLL 锁定中断控制寄存器 (SCU_PLLKCON)															
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															LK_FLAG
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							IF	保留	LK_IFS<2:0>			保留		IE	

—	bit31-17	—	—
LK_FLAG	bit16	R	<b>PLL 锁定检测标志位</b> 0: PLL 未锁定 1: PLL 锁定, 锁频成功
—	bit15-9	—	—
IF	bit8	R/W	<b>PLL 中断标志位</b> 0: 未发生 PLL 锁定标志触发事件 1: 发生 PLL 锁定标志触发事件 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。
—	bit7	—	—
LK_IFS<2:0>	bit6-4	R/W	<b>PLL 锁定标志产生模式选择位</b> 000: PLL 锁定标志上升沿产生中断, 锁频成功 001: PLL 锁定标志下降沿产生中断, 锁频失败 010: PLL 锁定标志高电平产生中断, 锁频成功 011: PLL 锁定标志低电平产生中断, 锁频失败 1xx: PLL 锁定标志变化 (上升或下降沿) 产生中断
—	bit3-1	—	—
IE	bit0	R/W	<b>PLL 锁定中断使能位</b> 0: 禁止 1: 使能

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

### 2.6.4.8 HRC时钟偏置控制寄存器 (SCU\_HRCBCON)

HRC 时钟偏置控制寄存器 (SCU_HRCBCON)															
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				HRCBSET<2:0>				保留							

—	bit31-12	W	保留, 需固定设置为 0
HRCBSET<2:0>	bit11-9	R/W	HRC 时钟偏置系数: 111: 当 HRC 频率为 32MHz, 16MHz 或 2MHz 时, 需软件固定设置为 111
—	Bit8-0	W	保留, 需固定设置为 0

注 1: 对 SCU\_HRCBCON 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: SCU\_HRCBCON 寄存器用于设置 HRC 时钟偏置系数, 需按照上述寄存器描述进行设置, 否则会导致 HRC 时钟频率偏差增大, 当 HRC 时钟为 48MHz 时, 该寄存器保持为复位默认值即可。

### 2.6.4.9 LCD时钟配置寄存器 (SCU\_LCDCLK)

LCD 时钟配置寄存器 (SCU_LCDCLK)															
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							CLKDIV<3:0>			保留		CLKSEL	CLKEN		

—	bit31-8	—	—
CLKDIV<3:0>	bit7-4	R/W	<b>LCD 时钟源分频选择位</b> 0000: 2 分频 0001: 4 分频 0010: 6 分频 0011: 8 分频 0100: 10 分频 0101: 12 分频 ..... 1110: 30 分频 1111: 32 分频
—	bit3-2	—	—
CLKSEL	bit1	R/W	<b>LCD 时钟源选择位</b> 0: 内部 LRC (约 31.25KHz)

			1: 外部 LOSC (32KHz)
CLKEN	bit0	R/W	<b>LCD 时钟源使能位</b> 0: 禁止 1: 使能

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

### 2.6.5 系统时钟应用说明

系统时钟使用方式请参考《应用笔记》。

## 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表

注：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 不可屏蔽中断、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 不可屏蔽中断与 32 个外设中断请求 IRQ 可由芯片配置控制，可通过寄存器 SCU\_NMICON 的 NMICS 位选择设置 NMI 不可屏蔽中断，可选择 IRQ0~IRQ31 中的一个作为 NMI 不可屏蔽中断。

虽然 Cortex-M0 对 NMI 不可屏蔽中断不支持中断使能位，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寄存器 SCU\_NMICON 的 NMIEN=1。

对于 32 个 IRQ，Cortex-M0 内核提供 32 个 IRQ 使能位，可对每个中断挂起请求独立控制是否响应。配置寄存器 NVIC\_ISER 的 SETENA 位和寄存器 NVIC\_ICER 的 CLRENA 位可使能或禁止对 IRQ 中断挂起请求的响应，使能时，则对应的中断挂起会触发进入中断向量服务程序，可查询触发中断挂起的中断源，进行对应的中断响应处理，中断响应后硬件会自动清除中断挂起；中断请求 IRQ0~IRQ31 分别对应于不同的端口和外设中断源，当中断源的中断标志 IF 和中断使能 IE 均置起时，则产生该中断源对应的中断挂起；除了通过实际的中断源触发中断挂起外，芯片还支持通过寄存器 NVIC\_ISPR 软件写 1 置中断挂起，也支持通过寄存器 NVIC\_ICPR 软件写 1 清除中断挂起。

除了 32 个 IRQ 使能位外，对所有可屏蔽中断，还支持一个总的中断屏蔽控制位，该屏蔽位为 1 时，将不再响应所有可屏蔽中断挂起请求，可通过寄存器 PRIMASK 的 PM 位设置，具体见附录章节“附录 1.2.6 异常/中断屏蔽寄存器 PRIMASK”的描述。

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	外部端口中断 0
17	IRQ1	PINT1	外部端口中断 1
18	IRQ2	PINT2	外部端口中断 2
19	IRQ3	PINT3	外部端口中断 3
20	IRQ4	PINT4	外部端口中断 4

编号	类型	功能	说明
21	IRQ5	PINT5	外部端口中断 5
22	IRQ6	PINT6	外部端口中断 6
23	IRQ7	PINT7	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	T16N4 中断	16 位定时器/计数器 4 中断
30	IRQ14	IWDT 中断	独立看门狗中断
31	IRQ15	WWDT 中断	窗口看门狗中断
32	IRQ16	CCM/LVD/PLLLK 中断	停振检测中断/低电压检测中断/PLL 失锁中断
33	IRQ17	PINT8	外部端口中断 8
34	IRQ18	PINT9	外部端口中断 9
35	IRQ19	KINT	外部按键输入中断
36	IRQ20	PINT10	外部端口中断 10
37	IRQ21	ADC 中断	模数转换中断
38	IRQ22	PINT11	外部端口中断 11
39	IRQ23	T16N5 中断	16 位定时器/计数器 5 中断
40	IRQ24	UART0 中断	UART0 中断
41	IRQ25	UART1 中断	UART1 中断
42	IRQ26	UART2 中断	UART2 中断
43	IRQ27	UART3 中断	UART3 中断
44	IRQ28	UART4 中断	UART4 中断
45	IRQ29	UART5 中断	UART5 中断
46	IRQ30	SPI0 中断	SPI0 中断
47	IRQ31	I2C0 中断	I2C0 中断

表 2-4 IRQ 分配列表

### 2.7.3 中断向量表的重映射

Cortex-M0 内核本身并不支持中断向量表的重映射，在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 SCU\_TBLREMAPEN”和“中断向量表偏移寄存器 SCU\_TBLOFF”，可以支持中断向量表的重映射，可用于将程序迁移至 SRAM 中运行，在 Flash IAP 操作过程中能够正常执行程序进行中断处理。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：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	<b>中断向量表重映射使能</b> 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； 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。								

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 1: 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, 因此地址必须能被 64×4=256 整除, 从而合法的起始地址可以是: 0x000, 0x100, 0x200 等。

注 2: 该寄存器只支持字写入, 不支持字节和半字写入。

注 3: 对 SCU\_TBLOFF 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。

### 2.7.4.4 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	<p><b>硬件错误 2 标志位</b></p> <p>0: 未发生在异常区域进行写入操作</p> <p>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)</p>
FLAG1	bit1	R/W	<p><b>硬件错误 1 标志位</b></p> <p>0: 未发生在异常区域进行取指操作</p> <p>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)</p>
FLAG0	bit0	R/W	<p><b>硬件错误 0 标志位</b></p> <p>0: 未发生读指令代码为空</p>

			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 内核读取程序存储区的指令时，读到的值为 0xFFFFFFFF；
- 注 2: 在异常区域进行取指操作，表示 Cortex-M0 内核运行程序时，在非程序存储区读取指令；
- 注 3: 在异常区域进行写入操作，表示程序执行写入操作时，对 Flash 程序区，Flash 信息区，IAP 自编程硬件固化代码区进行了写操作；
- 注 4: 清除硬件错误标志位时，需要设置 SCU\_PROT 寄存器，关闭写保护；
- 注 5: 在应用中一般不需要设置和使用硬件错误标志寄存器 SCU\_FAULTFLAG。

以下 NVIC 寄存器列表及基址参见章节：中断控制器（NVIC）寄存器列表。

### 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：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
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>IRQ 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 IRQ 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：80H															
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>IRQ 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 IRQ 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	置 IRQ 挂起位 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 IRQ 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	清 IRQ 挂起位 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

—	bit21-16	—	11: 最低优先级
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)

<b>IRQ16~19 优先级控制寄存器 (NVIC_PR4)</b>															
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)

<b>IRQ20~23 优先级控制寄存器 (NVIC_PR5)</b>															
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)

<b>IRQ24~27 优先级控制寄存器 (NVIC_PR6)</b>															
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 rnpn 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 rnpn 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			ISRPENDDING	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	<b>置 SysTick 异常挂起位</b>

			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	—	—
ISRPENDDING	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

—	bit2-0	—	—
读取始终为 1，指示非对齐访问产生硬故障			

### 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務
- ◇ 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，SYST\_CSR 寄存器的 COUNTFLAG 状态位置 1，并重载 SYST\_RVR 寄存器中的计数初值。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。

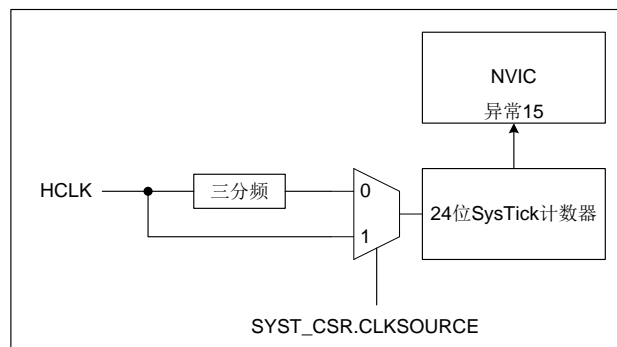


图 2-14 系统定时器框图

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 0，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，通过 CLKSOURCE 位选择计数时钟源，通过 TICKINT 位设置是否使能计数到 0 时产生异常挂起，再通过 ENABLE 位使能计数。

SysTick 寄存器列表和基址参见章节：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 0, 或写 SYST_CVR 寄存器清 0										
—	bit15-3				—	—										
CLKSOURCE	bit2				R/W	<b>SYSTICK 时钟源选择位</b> 0: 基准时钟 (HCLK 的 3 分频) 1: 处理器时钟 HCLK										
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 FHCLK/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 (本产品不支持)
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 1: 本产品只提供内部基准时钟, 其频率为  $F_{HCLK}/3$ ;

注 2: 在应用中一般不需要设置和使用 SYSTICK 校准值寄存器。

## 2.10 配置字软件控制

芯片支持对 BOR 复位电路检测的 VDD 电压信号进行滤波，可通过寄存器 SCU\_SOFTCFG 的 BORFLT<2:0>来设置滤波时间。

### 2.10.1.1 系统配置软件控制寄存器 (SCU\_SOFTCFG)

系统配置软件控制寄存器 (SCU_SOFTCFG)															
偏移地址: 24 <sub>H</sub>															
复位值: 00000000_00000000_00000000_11100001 <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								BORFLT<2:0>		保留					

—	bit31-8	—	—
BORFLT	bit7-5	R/W	<b>BOR 复位信号滤波长度</b> 00x: 1 个 LRC 时钟周期 010~111: 2~7 个 LRC 时钟周期
—	bit4-0	—	保留

注 1: 为防止电压瞬间跌落造成芯片工作异常, 可将 BORFLT 值适当减小, 在电压跌落时及时产生 BOR 复位。

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



## 2.11 定时器（T16N/T32N）同步启动关停控制

### 2.11.1 概述

通过 SCU\_TIMEREN 和 SCU\_TTIMERDIS 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\_TTIMERDIS 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\_TTIMERDIS。

### 2.11.2 特殊功能寄存器

#### 2.11.2.1 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：58 <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	T16N5EN	T16N4EN	T16N3EN	T16N2EN	T16N1EN	T16N0EN	

—	bit31-9	—	—
T32N0EN	bit8	W	<b>T32N0使能位</b> 0: 无动作 1: 触发使能
—	bit7-6	—	—
T16N5EN	bit5	W	<b>T16N5使能位</b> 0: 无动作 1: 触发使能
T16N4EN	bit4	W	<b>T16N4使能位</b> 0: 无动作 1: 触发使能
T16N3EN	bit3	W	<b>T16N3使能位</b> 0: 无动作 1: 触发使能
T16N2EN	bit2	W	<b>T16N2使能位</b> 0: 无动作 1: 触发使能
T16N1EN	bit1	W	<b>T16N1使能位</b> 0: 无动作 1: 触发使能
T16N0EN	bit0	W	<b>T16N0使能位</b>

			0: 无动作 1: 触发使能
--	--	--	-------------------

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。

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

SCU_TIMERDIS 关停控制寄存器 (SCU_TIMERDIS)															
偏移地址: 5C <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		T16N5DIS	T16N4DIS	T16N3DIS	T16N2DIS	T16N1DIS	T16N0DIS

—	bit31-9	—	—
T32N0DIS	bit8	R/W	<b>T32N0 关停位</b> 0: 无动作 1: 触发关停
—	bit7-6	—	—
T16N5DIS	bit5	R/W	<b>T16N5 关停位</b> 0: 无动作 1: 触发关停
T16N4DIS	bit4	R/W	<b>T16N4 关停位</b> 0: 无动作 1: 触发关停
T16N3DIS	bit3	R/W	<b>T16N3 关停位</b> 0: 无动作 1: 触发关停
T16N2DIS	bit2	R/W	<b>T16N2 关停位</b> 0: 无动作 1: 触发关停
T16N1DIS	bit1	R/W	<b>T16N1 关停位</b> 0: 无动作 1: 触发关停
T16N0DIS	bit0	R/W	<b>T16N0 关停位</b> 0: 无动作 1: 触发关停

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。

## 2.12 模块调试相关配置

### 2.12.1 概述

通过 SCU\_DBGHALT 控制寄存器，可以选择在调试模式下，Timer 或 WDT 是否跟随调试的状态计数或暂停，选择暂停时，则在调试暂停（HALT）时，计数也会暂停，否则仍保持计数。在非调试模式下或在调试模式全速运行时，该寄存器不会影响 Timer 和 WDT 的计数状态。

由于调试模式和非调试模式的不同，可能两种模式下 Timer 或 WDT 计数快慢会稍有差异。对于 Timer 计数精度要求较高的应用场景，建议关闭相应模块的调试暂停使能位，使得调试过程不影响 Timer 正常计数，例如 Timer 用作 PWM 输出等。

### 2.12.2 特殊功能寄存器

#### 2.12.2.1 调试模式模块暂停控制寄存器（SCU\_DBGHALT）

SCU_DBGHALT 调试模式模块暂停控制寄存器（SCU_DBGHALT）															
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													WWDT	IWDTH	
													HT		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						T32N0HT	保留	T16N5H	T16N4H	T16N3H	T16N2H	T16N1	T16N0H		
								T	T	T	T	HT	T		

—	bit31-18	—	—
WWDTHT	bit17	R/W	<b>WWDT调试暂停使能位</b> 0: 禁止（调试不影响计数） 1: 使能（随调试运行状态计数或暂停）
IWDTH	bit16	R/W	<b>IWDT调试暂停使能位</b> 0: 禁止（调试不影响计数） 1: 使能（随调试运行状态计数或暂停）
—	bit15-9	—	—
T32N0HT	bit8	R/W	<b>T32N0调试暂停使能位</b> 0: 禁止（调试不影响计数） 1: 使能（随调试运行状态计数或暂停）
—	bit7-6	—	—
T16N5HT	bit5	R/W	<b>T16N5调试暂停使能位</b> 0: 禁止（调试不影响计数） 1: 使能（随调试运行状态计数或暂停）
T16N4HT	bit4	R/W	<b>T16N4调试暂停使能位</b> 0: 禁止（调试不影响计数）

			1: 使能 (随调试运行状态计数或暂停)
T16N3HT	bit3	R/W	<b>T16N3调试暂停使能位</b> 0: 禁止 (调试不影响计数) 1: 使能 (随调试运行状态计数或暂停)
T16N2HT	bit2	R/W	<b>T16N2调试暂停使能位</b> 0: 禁止 (调试不影响计数) 1: 使能 (随调试运行状态计数或暂停)
T16N1HT	bit1	R/W	<b>T16N1调试暂停使能位</b> 0: 禁止 (调试不影响计数) 1: 使能 (随调试运行状态计数或暂停)
T16N0HT	bit0	R/W	<b>T16N0调试暂停使能位</b> 0: 禁止 (调试不影响计数) 1: 使能 (随调试运行状态计数或暂停)

注: 对 SCU\_DBGHALT 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。

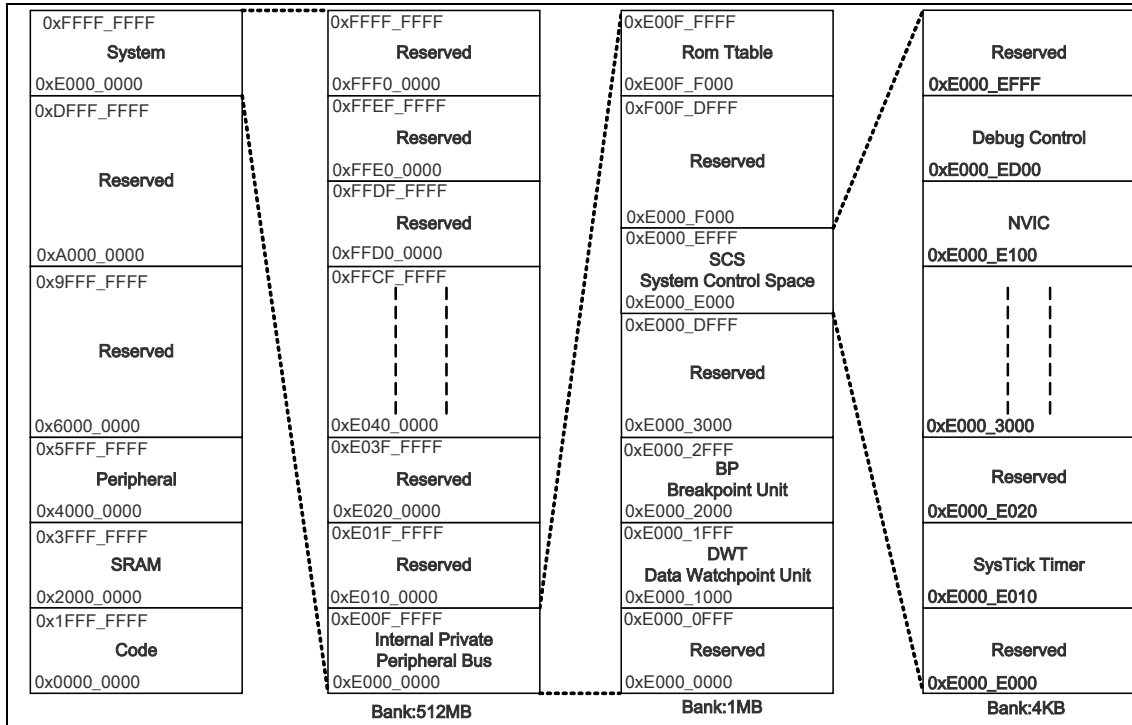


图 3-1 内部存储系统分配示意图

### 3.2 FLASH存储器

#### 3.2.1 信息区FLASH

芯片内部的信息区 FLASH 分为 2 个分区：INFO0 区和 INFO4 区。（可针对特定用户开发 INFO2~3 的擦写）

- ◇ 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, 以及用户程序校验码 CHKSUM, 用户识别码 UserID0~UserID3
- ◇ INFO4 信息区包含 96 位芯片唯一识别码 UID, 分为 UID0~UID2 共 3 个 words
  - 芯片唯一识别码 UID 出厂时已固定, 无法更改, 程序只读
- ◇ 用户程序只可读信息区数据, 不能擦写
- ◇ 信息区基地址为 0x40000。芯片配置字 CFG\_WORD0, CFG\_WORD1, CFG\_WRP0, CFG\_WRP1, CFG\_DAFLS 和 CFG\_GBRDP 的地址分别为 0x40000, 0x40008, 0x40020, 0x40030, 0x40040 和 0x40050; 用户程序校验码 CHKSUM 的地址为 0x401C0, 用户识别码 UserID0~UserID3 的地址分别为 0x401D0, 0x401D8, 0x401E0, 0x401E8; 芯片唯一识别码 UID0~UID2 的地址分别为 0x409E0, 0x409E8, 0x409F0。

### 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 Boot Flash 启动地址设置、WDT 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，芯片配置字地址单元在出厂时可能为非空，所以在 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 16Bits 复位值	1000_0100_1001_0000 <sub>B</sub> (8490 <sub>H</sub> )	
—	bit 63-32	保留未用
—	bit 31-16	<b>Bits 15-0 取反值（不满足取反时 Bit15-0 强制为默认值）</b>
CFG_SWD	bit 15-14	<b>SWD 调试模式使能位</b> 0x: 禁止（调试完毕后需禁止，避免管脚输入悬空） 10: SWD 调试自动识别（默认） 11: SWD 调试强制使能（软件对调试端口的设置无效）
CFG_PB25R	bit 13	<b>PB25 用于复位使能位</b> 0: 禁止（默认） 1: PB25 可用作 MRSTN
—	bit 12	固定为 0
CFG_RUNMD	bit 11	<b>Flash 指令运行效率选择位</b> 0: 正常（默认，推荐设置为 0） 1: 低速（向下兼容其他系列产品）
CFG_START	bit 10	<b>程序起始区域选择位</b> 0: BootFlash 1: AppFlash（默认）
—	bit 9	固定为 0
CFG_PWRTEB	bit 8	<b>上电 140ms 延时使能位</b> 0: 使能（默认） 1: 禁止 仅在将 MRSTN 管脚作为外部复位(CFG_MRSTN=1)时，该配置位才有效，推荐设置为使能。
CFG_MRSTN	bit 7	<b>MRSTN 管脚复用选择位</b> 0: GPIO 功能 1: MRSTN 功能（默认）
CFG_BORV	bit 6-4	<b>BOR 电压点选择位</b> 000: 保留 001: 2.1V（默认） 010: 2.5V 011: 2.8V 100: 3.1V 101: 3.6V 110: 4.0V

		111: 保留
CFG_WWDTEN	bit 3	<b>WWDT 硬件看门狗使能</b> 0: 软件使能后可再关闭 (默认) 1: 软件使能后无法再关闭
CFG_IWDTEN	bit 2	<b>IWDT 硬件看门狗使能</b> 0: 由软件控制 (默认) 1: 硬件看门狗使能 (建议使能硬狗可提升系统可靠性) IWDT 硬件看门狗使能后, 软件无法关闭; IWDT 中断强制使能, 软件无法关闭; IWDT 复位强制使能, 软件无法关闭; IWDT 时钟源固定为 LRC, 软件无法切换。
—	bit 1-0	固定为 0

注 1: 对配置位 CFG\_PWRTEB, 推荐用户设置为上电延时使能 (CFG\_PWRTEB=0), 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, 在使用 4 线烧录接口时, 必须使能该上电延时。当 MRSTN 管脚复用为 GPIO 功能 (CFG\_MRSTN=0) 时, 该上电延时固定为使能, 与配置位 CFG\_PWRTEB 无关。

注 2: 在对 Flash 程序加密编程时, 必须要禁止 CFG\_SWD 位, 否则加密无效; 调试完毕后, 也需禁止 CFG\_SWD 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 3: 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>), 将外部 XTAL 振荡器作为时钟源。

注 4: 如果将 PB25 配为复位端口 MRSTN, 则不能同时用作 SWD 调试端口, 此时该端口的内部上拉电阻固定为使能, 约 55K 欧姆, 只有在使用 4 线烧录接口时, 才建议根据实际需求决定是否将 PB25 配为 MRSTN 端口。

注 5: 在 SWD 调试自动识别模式下, 进入调试模式后, 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; 在 SWD 调试强制使能模式下, 对应的 IO 端口固定为调试端口, 程序软件对其设置无效。

注 6: 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	<b>bit 15-0 取反值</b>
—	bit 15-8	保留未用
CFG_XTAL	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) 0x0F: 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> , 0030 <sub>H</sub>

低 16 位复位值	0000_0000_0000_0001 <sub>B</sub> (0001 <sub>H</sub> )	
—	bit 31-16	<b>bit 15-0 取反值</b>
END	bit 15-9	<b>保护结束页配置位</b> 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 ..... 0x7F: Flash Page 511 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 8	保留未用
START	bit 7-1	<b>保护起始页配置位</b> 0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 ..... 0x7F: Flash Page 508
ENB	bit 0	<b>保护使能位</b> 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	0040 <sub>H</sub>	
低 16 位复位值	0000_0000_0000_0001 <sub>B</sub> (0001 <sub>H</sub> )	
—	bit 31-16	<b>bit 15-0 取反值</b>
END	bit 15-9	<b>数据 Flash 结束页配置位</b> 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 ..... 0x7F: Flash Page 511 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 8	保留未用
START	bit 7-1	<b>数据 Flash 起始页配置位</b> 0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 ..... 0x7F: Flash Page 508



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	0050 <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) 0/YYYYY_XXXX: 读保护等级为 Level 2 (YYYY 不为 FFFF) (默认)

### 3.2.1.2 用户程序校验码和用户识别码

用户程序校验码和用户识别码位于 FLASH 存储器的 INFO0 信息区，均可在 ISP 编程时烧录。

寄存器名称	用户程序校验码 (CHKSUM)	
地址偏移	01C0 <sub>H</sub>	
CHKSUM	bit 31-0	用户程序校验码 校验码计算方法包括 CRC 校验、字节累加校验，用户可以通过编程界面进行选择；校验码计算范围包括程序空间和用户配置字 (CFG_WORD0、CFG_WORD1、CFG_WRP0、CFG_WRP1、CFG_DAFLS)

寄存器名称	用户识别码 (UserID0~ UserID3)	
地址偏移	UserID0~ UserID3 分别为 01D0 <sub>H</sub> 、01D8 <sub>H</sub> 、01E0 <sub>H</sub> 、01E8 <sub>H</sub>	
UserID0~UserID3	bit 31-0	用户识别码 该识别码可以在编程器界面中设置，共包括 4 个 words (128bits)

注：用户识别码 UserID，可以作为用户产品序列码 UPSN 使用，不同厂家的编程器界面中对该信息区域的名称描述有所不同，用户识别码 UserID 和用户产品序列码 UPSN 均是指该信息区域。

### 3.2.1.3 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，共包括 3 个 words，用户程序可读。UID2~UID0 各字描述如下。

芯片唯一识别码 (UID2~UID0)	
地址	000209F0 <sub>H</sub> (UID2) 、000209E8 <sub>H</sub> (UID1), 000209E0 <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

ES8H0394/0354 系列芯片内部的程序存储器 FLASH 容量最大为 256K 字节，地址范围为 0000\_0000<sub>H</sub>~0003\_FFFF<sub>H</sub>，共分为 512 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。

型号	程序区 FLASH	寻址空间	页数
ES8H0394	256K Bytes	0x0000_0000 <sub>H</sub> ~0x0003_FFFF <sub>H</sub>	512

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。将芯片配置字的 CFG\_SWD 位配置为 SWD 调试自动识别时，SWD 端口不可再在程序软件中用作 GPIO 或其他外设端口，否则会导致调试异常；配置为 SWD 调试强制使能时，SWD 端口固定为调试端口，程序软件对其设置无效。

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。

Boot Flash 启动地址为高 8K 字节空间的首地址：0x0003\_E000(ES8H0394/0354)。

系统启动地址可选择为主程序空间首地址：0x0000\_0000；也可选择为 Boot Flash 空间首地址，通过编程界面选取。复位向量的存放位置由所选择的系统启动地址决定。系统启动地址选择为 Boot Flash 空间首地址时，boot 引导结束后，用户程序可通过配置 SCU\_TBLREMAPEN 和 SCU\_TBLOFF 切换回主程序空间。

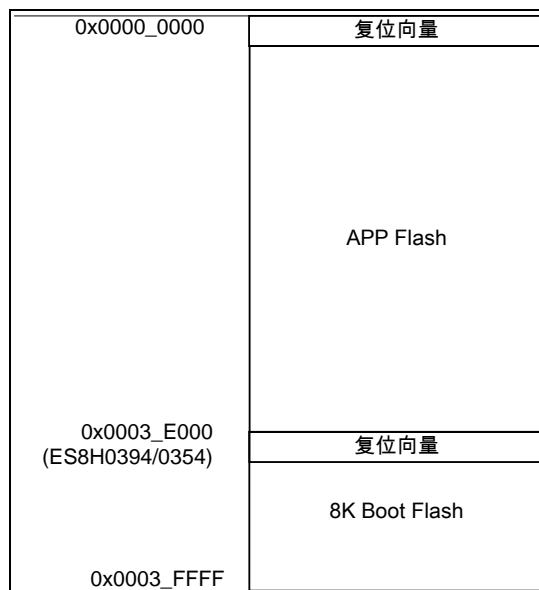


图 3-2 Boot Flash 启动地址分配示意图

### 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, 通过 ENB 位配置两段写保护区使能。

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	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 数据Flash区 (程序存储器中划分独立DATA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

##### 3.3.2.4 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					
			擦	写	读	在 FLASH 中运行			在 SRAM 中运行		
						擦	写	读	擦	写	读
Flash Code 区	非写保 护区	Level0	全擦/页擦	是	是	NA	NA	是	页擦	是	是
		Level1	全擦/页擦	否	是	NA	NA	是	页擦	是	是
		Level2	否	否	是	NA	NA	是	页擦	是	是
	写保护 区	Level0	全擦	否	是	NA	NA	是	否	否	是
		Level1	全擦	否	是	NA	NA	是	否	否	是
		Level2	否	否	是	NA	NA	是	否	否	是

注 1：若全局读保护等级为 Level1 或 Level2 时，编程和调试工具不能直接读取 Flash Code 区。  
注 2：调试模式和用户模式下，在 Flash、SRAM 中运行程序时，Flash Code 区均可以被程序正常读取，与全局保护等级无关，推荐在程序调试完毕，系统量产时，通过配置字 CFG\_SWID 禁止调试模式，加强对 Code 区的保护。  
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，不支持对 Flash 的全擦。  
注 5：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 IAP\_FLASHCR.FLASHREQ，查询 Flash 操作许可状态 IAP\_FLASHSR.FLASHACK；
4. 设置 IAP 自编程使能 IAP\_FLASHCR.IAPEN；
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. 判断 IAP\_FLASHSR.MASE 标志位是否置起；
8. 设置 Flash 操作请求禁止 IAP\_FLASHCR.FLASHREQ。

注 1：Flash 程序区全操作，仅在芯片调试模式下支持，芯片配置字的调试模式使能位 CFG\_SWID 需为 1，并且芯片进入 SWD 调试模式后，才可进行 Flash 程序区全擦除，程序区全擦除完毕后，解除了对配置区的擦除保护，也可以继续执行页擦除命令，对用户配置信息区 INFO0 进行页擦除，可查询寄存器 IAP\_FLASHSR 的 MASEUL 位，判断用户配置区擦除是否已解锁；  
注 2：当全擦除时间超过 15ms 仍未结束，会置起 IAP\_FLASHSR 寄存器的超时错误标志 TIMEOUT，并退出全擦除状态。

### 3.3.4 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 IAP\_FLASHCR.FLASHREQ，查询 Flash 操作许可状态 IAP\_FLASHSR.FLASHACK；
4. 设置 IAP 自编程使能 IAP\_FLASHCR.IAPEN；
5. 写入需擦除页的首地址 IAP\_FLASHADDR.ADDR 及地址反码 IAP\_FLASHADDR.ADDRINV，页擦除区域大小 IAP\_FLASHCR.PGSZ 设为 0，固定为 512 字节；
6. 写入 IAP\_FLASHCMD.CMD 命令触发页擦除；
7. 等待 IAP\_FLASHSR.BUSY 标志再次变为空闲状态；
8. 判断 IAP\_FLASHSR.SERA 标志位是否置起；
9. 设置 Flash 操作请求禁止 IAP\_FLASHCR.FLASHREQ。

注 1：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；

注 2：对擦除页首地址和地址反码，需先写地址寄存器 IAP\_FLASHADDR，再写地址反码寄存器 IAP\_FLASHADDRINV，两者的反码逻辑正确才能执行页擦除命令，否则无法执行页擦除命令，并会置起 IAP\_FLASHSR 寄存器的地址反码错误标识 ADDERR；

注 3：当页擦除时间超过 3ms 仍未结束，会置起 IAP\_FLASHSR 寄存器的超时错误标志 TIMEOUT，并退出页擦除状态；

注 4：芯片支持对擦除页地址是否超出 FLASH 地址范围，及是否触发对保护区域的擦除进行检测，可通过 IAP\_FLASHSR 寄存器的擦写地址错误标志 WAE 和擦写保护错误标志 WPE 进行判断，发生错误时，擦除操作无效，可通过 IAP\_FLASHCR 寄存器的 IAPRST 复位来清除错误标志。

### 3.3.5 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 IAP\_FLASHCR.FLASHREQ，查询 Flash 操作许可状态 IAP\_FLASHSR.FLASHACK；
4. 设置 IAP 自编程使能 IAP\_FLASHCR.IAPEN；
5. 写入需编程地址 IAP\_FLASHADDR.ADDR 及地址反码 IAP\_FLASHADDR.ADDRINV；
6. 写入需编程数据 IAP\_FLASHDR.DATA；
7. 写入 IAP\_FLASHCMD.CMD 命令触发字编程；

8. 等待 IAP\_FLASHSR.BUSY 标志再次变为空闲状态;
9. 判断 IAP\_FLASHSR.PROG 标志位是否置起;
10. 设置 Flash 操作请求禁止 IAP\_FLASHCR.FLASHREQ。

注 1: 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;

注 2: 对编程地址和地址反码, 需先写地址寄存器 IAP\_FLASHADDR, 再写地址反码寄存器 IAP\_FLASHADDINV, 两者的反码逻辑正确才能执行编程命令, 否则无法执行编程命令, 并会置起 IAP\_FLASHSR 寄存器的地址反码错误标识 ADDERR;

注 3: 当编程时间超过 45us 仍未结束, 会置起 IAP\_FLASHSR 寄存器的超时错误标志 TIMEOUT, 并退出编程状态。

注 4: 芯片支持对编程地址是否超出 FLASH 地址范围, 及是否触发对保护区域的编程进行检测, 可通过 IAP\_FLASHSR 寄存器的擦写地址错误标志 WAE 和擦写保护错误标志 WPE 进行判断, 发生错误时, 编程操作无效, 可通过 IAP\_FLASHCR 寄存器的 IAPRST 复位来清除错误标志。

### 3.3.6 Flash编程数据FIFO

芯片支持通过写 FIFO 数据直接触发字编程操作, 而无需通过 IAP\_FLASHCMD 寄存器触发字编程命令, 可提高指令执行效率。Flash 编程 FIFO 可通过 IAP\_FLASHCR 寄存器的 FIFOEN 使能, 该 FIFO 寄存器 IAP\_FLASHFIFO 为只写, 当数据写入 FIFO 后, 可在 IAP\_FLASHDR 寄存器中体现, 在 FIFO 中每写入一次数据, 均可直接触发一次 Flash 字编程。FIFOEN 使能后, 还可以通过 FIFODF 使能对数据 Flash 区进行字编程。

### 3.3.7 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:

#### 3.3.7.1 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)

#### 3.3.7.2 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)

#### 3.3.7.3 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）

#### 3.3.7.4 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）

#### 3.3.7.5 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）

#### 3.3.7.6 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。

### 3.3.8 特殊功能寄存器

#### 3.3.8.1 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>	

—	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。

#### 3.3.8.2 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														ADDR<17:16>	

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

—	bit 31-18	—	保留
ADDR	bit 17-0	R/W	FLASH 地址

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。

#### 3.3.8.3 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：当 FIFO 写入数据后，将自动触发字编程操作，无需通过 IAP\_FLASHCMD 寄存器进行编程命令触发。

### 3.3.8.4 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
------	----------	-----	-----------

### 3.3.8.5 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（仅在调试模式下支持） 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 其他：保留
-----	----------	---	--

### 3.3.8.6 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>

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							PGSZ	FIFODF	保留	FIFOEN	FLASHREQ	保留		IAPRST	IAPEN

—	bit 31-9	—	保留
PGSZ	bit 8	R/W	页擦除区域大小选择 0: 512Byte (软件需固定写 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
—	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
—	bit 3-2	—	保留
IAPRST	bit 1	W	自编程复位 0: 无操作 1: 复位 IAP 操作逻辑电路, 复位后硬件自动清 0
IAPEN	bit 0	R/W	自编程使能 0: 禁止 1: 使能

### 3.3.8.7 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							MAS EUL	ADDE RR	保留						
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	

—	bit 31-25	—	保留
MASEUL	bit 24	R	<b>程序区全擦保护解除位</b> 0: 未解除 1: 已解除 在芯片调试模式下，程序区全擦完毕后，解除了配置区擦除保护，允许对用户配置信息区INFO0进行页擦除
ADDERR	bit 23	R	<b>地址反码错误标识位</b> 0: 反码正确 1: 反码错误
—	bit 22-8	—	保留
TIMEOUT	bit 7	R	<b>超时错误标志</b> 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	<b>字编程完成标志</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
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IAP操作了非法的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 触发了保护区域的擦除或编程，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
--	--	--	--------------------

### 3.3.8.8 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														ADDRINV<17:16>	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRINV<15:0>															

—	bit 31-18	—	保留
ADDRINV	bit 17-0	R/W	FLASH 地址反码

### 3.4 数据存储器 (SRAM)

ES8H0394/0354 系列芯片内部集成最大 24K 字节数据存储器 SRAM，地址范围为 2000\_0000<sub>H</sub>~2000\_5FFF<sub>H</sub>。

型号	SRAM 数据存储器	地址范围
ES8H0394/0354	24K Bytes	2000_0000 <sub>H</sub> ~2000_5FFF <sub>H</sub>

#### 3.4.1 SRAM地址映射

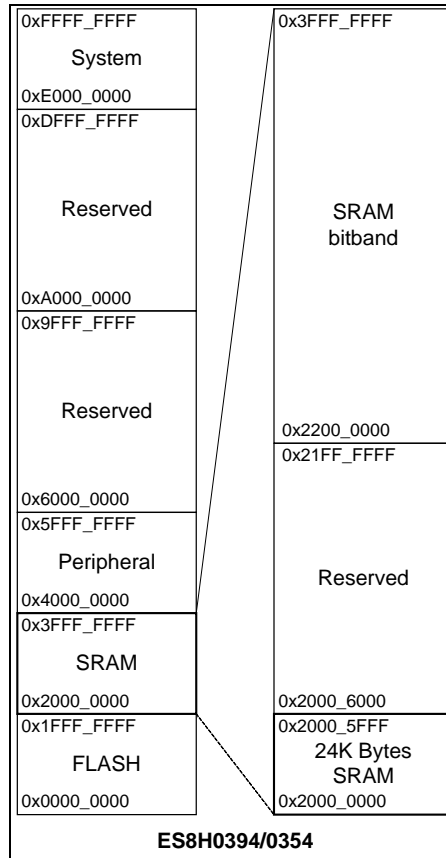


图 3-3 SRAM 映射图

#### 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，即占用 4 个字节地址；一个 byte 占用 8x4=32 个地址。通过访问这些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：

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

例如，字节地址 A 为 0x2000\_0001，访问该地址的 bit1，地址为：

$$\text{AliasAddress\_A\_N} = 0x2200\_0000 + 1 \times 32 + 1 \times 4 = 0x2200\_0024$$

### 3.5 外设寄存器

#### 3.5.1 外设寄存器映射

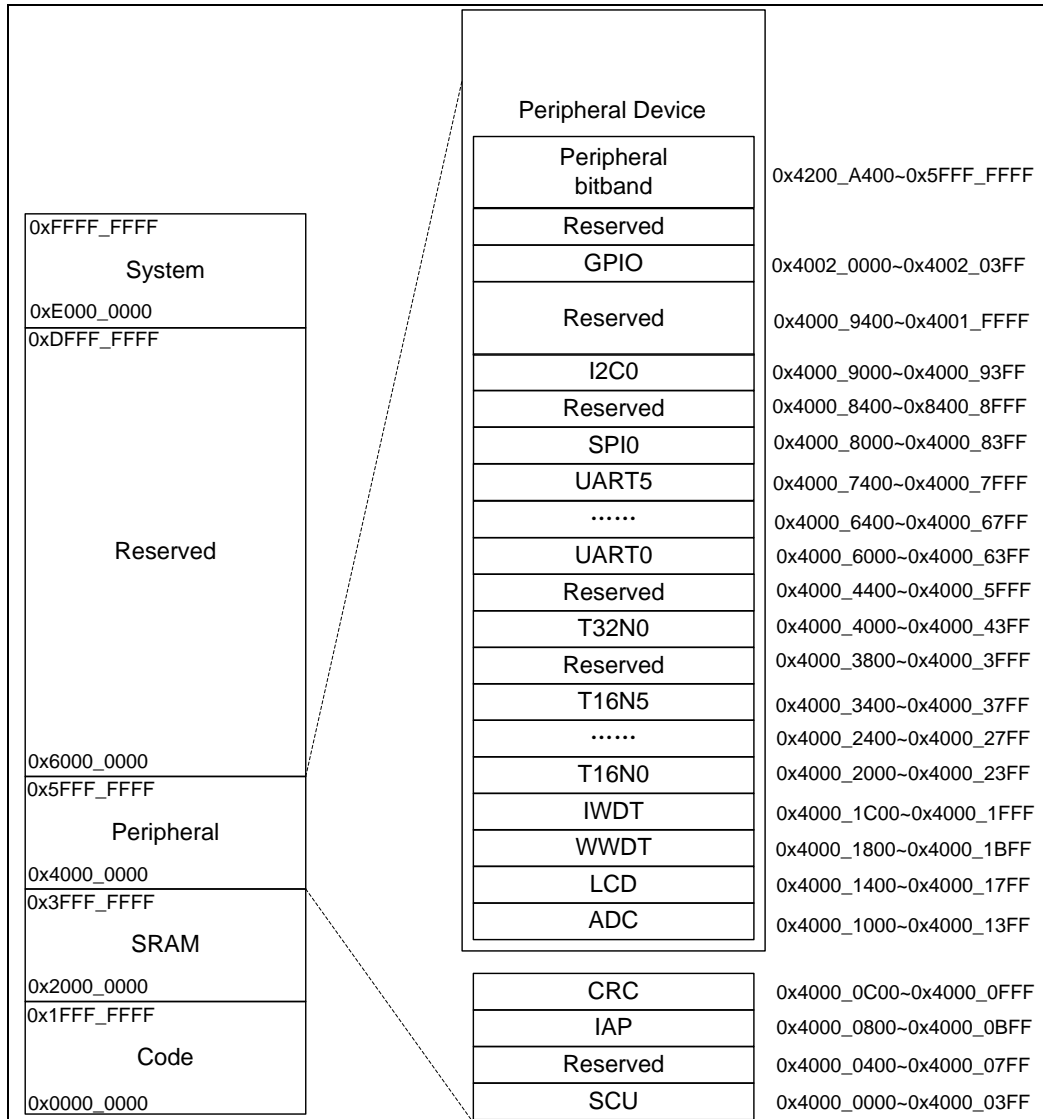


图 3-4 外设存储器分配示意图

注: Reserved 保留寄存器区域为只读, 读出值为 00000000H。

#### 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。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。

利用外设寄存器位带访问对寄存器位置 1 和清 0 的示例如下:

```
LDR    R0, = AliasAddress_A_N
MOVS   R1, #1
STR    R1, [R0]                ; 对该位置 1
```

```
LDR    R0, = AliasAddress_A_N
MOVS   R1, #0
STR    R1, [R0]                ; 对该位清 0
```

### 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_WAKEUPTIME	0010 <sub>H</sub>	系统唤醒时间控制寄存器
SCU_DBGHALT	0018 <sub>H</sub>	调试模式模块暂停控制寄存器
SCU_FLASHWAIT	0020 <sub>H</sub>	FLASH 访问等待时间寄存器
SCU_SOFTCFG	0024 <sub>H</sub>	系统配置软件控制寄存器
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_PLLLKCON	0030 <sub>H</sub>	PLL 锁定中断控制寄存器
SCU_HRCBCON	0034 <sub>H</sub>	HRC 时钟偏置控制寄存器
SCU_LCDCLK	0038 <sub>H</sub>	LCD 时钟配置寄存器
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_PCLKEN0	0048 <sub>H</sub>	外设时钟控制寄存器 0
SCU_PCLKEN1	004C <sub>H</sub>	外设时钟控制寄存器 1
SCU_PRSTEN0	0050 <sub>H</sub>	外设复位控制寄存器 0
SCU_PRSTEN1	0054 <sub>H</sub>	外设复位控制寄存器 1
SCU_TIMEREN	0058 <sub>H</sub>	TIMER 使能控制寄存器
SCU_TIMERDIS	005C <sub>H</sub>	TIMER 关停控制寄存器
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 CRC寄存器列表

CRC 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
CRC 基地址: 4000_0C00 <sub>H</sub>		
CRC_UL	0000 <sub>H</sub>	CRC 解锁寄存器
CRC_CON	0004 <sub>H</sub>	CRC 控制寄存器
CRC_TRIG	0008 <sub>H</sub>	CRC 触发寄存器
CRC_ADDR	000C <sub>H</sub>	CRC FLASH 校验起始地址寄存器
CRC_SIZE	0010 <sub>H</sub>	CRC FLASH 校验数据块大小寄存器
CRC_DI	0014 <sub>H</sub>	CRC 用户模式输入数据寄存器
CRC_DO	0018 <sub>H</sub>	CRC 校验结果寄存器
CRC_STA	001C <sub>H</sub>	CRC 状态寄存器
CRC_FA	0020 <sub>H</sub>	CRC 当前 FLASH 地址寄存器

### 3.5.5 GPIO寄存器列表

GPIO 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
GPIO 基地址: 4002_0000 <sub>H</sub>		
GPIO_PAPORT	0000 <sub>H</sub>	PA 端口状态寄存器
GPIO_PADATA	0010 <sub>H</sub>	PA 端口数据寄存器
GPIO_PADATABSR	0014 <sub>H</sub>	PA 输出置位寄存器
GPIO_PADATABCR	0018 <sub>H</sub>	PA 输出清零寄存器
GPIO_PADATABRR	001C <sub>H</sub>	PA 输出翻转寄存器
GPIO_PADIR	0020 <sub>H</sub>	PA 端口方向控制寄存器
GPIO_PADIRBSR	0024 <sub>H</sub>	PA 端口方向置位寄存器
GPIO_PADIRBCR	0028 <sub>H</sub>	PA 端口方向清零寄存器
GPIO_PADIRBRR	002C <sub>H</sub>	PA 端口方向翻转寄存器
GPIO_PAFUNC0	0030 <sub>H</sub>	PA[7:0]端口复用选择寄存器
GPIO_PAFUNC1	0034 <sub>H</sub>	PA[15:8]端口复用选择寄存器
GPIO_PAFUNC2	0038 <sub>H</sub>	PA[23:16]端口复用选择寄存器
GPIO_PAFUNC3	003C <sub>H</sub>	PA[31:24]端口复用选择寄存器
GPIO_PAINEB	0040 <sub>H</sub>	PA 端口输入控制寄存器
GPIO_PAODE	0044 <sub>H</sub>	PA 端口开漏控制寄存器
GPIO_PAPUE	0048 <sub>H</sub>	PA 端口弱上拉使能寄存器
GPIO_PAPDE	004C <sub>H</sub>	PA 端口弱下拉使能寄存器
GPIO_PADS	0050 <sub>H</sub>	PA 端口驱动电流控制寄存器
GPIO_PATYP	0054 <sub>H</sub>	PA 端口类型选择寄存器
GPIO_PAFLT	0058 <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	0090 <sub>H</sub>	PB 端口数据寄存器
GPIO_PBDATABSR	0094 <sub>H</sub>	PB 端口输出置位寄存器
GPIO_PBDATABCR	0098 <sub>H</sub>	PB 端口输出清零寄存器



GPIO 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
GPIO_PBDATABRR	009C <sub>H</sub>	PB 端口输出翻转寄存器
GPIO_PBDIR	00A0 <sub>H</sub>	PB 端口方向控制寄存器
GPIO_PBDIRBSR	00A4 <sub>H</sub>	PB 端口方向置位寄存器
GPIO_PBDIRBCR	00A8 <sub>H</sub>	PB 端口方向清零寄存器
GPIO_PBDIRBRR	00AC <sub>H</sub>	PB 端口方向翻转寄存器
GPIO_PBFUNC0	00B0 <sub>H</sub>	PB[7:0]端口复用选择寄存器
GPIO_PBFUNC1	00B4 <sub>H</sub>	PB[15:8]端口复用选择寄存器
GPIO_PBFUNC2	00B8 <sub>H</sub>	PB[23:16]端口复用选择寄存器
GPIO_PBFUNC3	00BC <sub>H</sub>	PB[27:24]端口复用选择寄存器
GPIO_PBINEB	00C0 <sub>H</sub>	PB 端口输入控制寄存器
GPIO_PBODE	00C4 <sub>H</sub>	PB 端口开漏控制寄存器
GPIO_PBPUE	00C8 <sub>H</sub>	PB 端口弱上拉使能寄存器
GPIO_PBPDE	00CC <sub>H</sub>	PB 端口弱下拉使能寄存器
GPIO_PBDS	00D0 <sub>H</sub>	PB 端口驱动电流控制寄存器
GPIO_PBTYP	00D4 <sub>H</sub>	PB 端口类型选择寄存器
GPIO_PBFLT	00D8 <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_PINTSEL0	0308 <sub>H</sub>	PINT0~7 中断源选择寄存器
GPIO_PINTSEL1	030C <sub>H</sub>	PINT8~11 中断源选择寄存器
GPIO_PINTCFG0	0310 <sub>H</sub>	PINT0~7 中断配置寄存器
GPIO_PINTCFG1	0314 <sub>H</sub>	PINT8~11 中断配置寄存器
GPIO_KINTIE	0318 <sub>H</sub>	KINT 中断使能寄存器
GPIO_KINTIF	031C <sub>H</sub>	KINT 中断标志寄存器
GPIO_KINTSEL0	0320 <sub>H</sub>	KINT0~7 中断源选择寄存器
GPIO_KINTSEL1	0324 <sub>H</sub>	KINT8~11 中断源选择寄存器
GPIO_KINTCFG0	0328 <sub>H</sub>	KINT0~7 中断配置寄存器
GPIO_KINTCFG1	032C <sub>H</sub>	KINT8~11 中断配置寄存器
GPIO_FLTCR	0330 <sub>H</sub>	PINT 中断滤波控制寄存器
GPIO_TXPWM	0380 <sub>H</sub>	脉宽调制寄存器
GPIO_BUZC	0390 <sub>H</sub>	蜂鸣器计数控制寄存器

### 3.5.6 IAP寄存器列表

IAP 寄存器列表		
寄存器名称	寄存器地址	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

### 3.5.7 ADC寄存器列表

ADC 寄存器列表		
寄存器名称	寄存器地址	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
ADC_VREFCON	0040 <sub>H</sub>	ADC 参考控制寄存器

### 3.5.8 LCD寄存器列表

LCD 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
LCD 基地址: 4000_1400 <sub>H</sub>		
LCD_MODE	0000 <sub>H</sub>	LCD 模式控制寄存器
LCD_CTRL	0004 <sub>H</sub>	LCD 控制寄存器
LCD_BLINK1	0008 <sub>H</sub>	LCD 闪烁控制寄存器 1
LCD_BLINK2	000C <sub>H</sub>	LCD 闪烁控制寄存器 2
LCD_RIF	001C <sub>H</sub>	LCD 更新状态寄存器
LCD_ICR	0024 <sub>H</sub>	LCD 状态标志清除寄存器
LCD_BUSY	0028 <sub>H</sub>	LCD 运行状态寄存器
LCD_RAM0	0030 <sub>H</sub>	LCD 显示数据寄存器 0
LCD_RAM1	0034 <sub>H</sub>	LCD 显示数据寄存器 1
LCD_RAM2	0038 <sub>H</sub>	LCD 显示数据寄存器 2
LCD_RAM3	003C <sub>H</sub>	LCD 显示数据寄存器 3
LCD_RAM4	0040 <sub>H</sub>	LCD 显示数据寄存器 4
LCD_RAM5	0044 <sub>H</sub>	LCD 显示数据寄存器 5
LCD_RAM6	0048 <sub>H</sub>	LCD 显示数据寄存器 6

LCD 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
LCD_RAM7	004C <sub>H</sub>	LCD 显示数据寄存器 7
LCD_RAM8	0050 <sub>H</sub>	LCD 显示数据寄存器 8
LCD_RAM9	0054 <sub>H</sub>	LCD 显示数据寄存器 9
LCD_RAM10	0058 <sub>H</sub>	LCD 显示数据寄存器 10
LCD_RAM11	005C <sub>H</sub>	LCD 显示数据寄存器 11
LCD_RAM12	0060 <sub>H</sub>	LCD 显示数据寄存器 12
LCD_RAM13	0064 <sub>H</sub>	LCD 显示数据寄存器 13
LCD_RAM14	0068 <sub>H</sub>	LCD 显示数据寄存器 14
LCD_RAM15	006C <sub>H</sub>	LCD 显示数据寄存器 15
LCD_SEGOE0	0070 <sub>H</sub>	LCD SEG 输出使能寄存器 0
LCD_SEGOE1	0074 <sub>H</sub>	LCD SEG 输出使能寄存器 1

### 3.5.9 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

### 3.5.10 IWDT寄存器列表

IWDT 寄存器列表		
寄存器名称	寄存器地址	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_RIS	0010 <sub>H</sub>	IWDT 中断标志寄存器
IWDT_LOCK	0100 <sub>H</sub>	IWDT 锁定寄存器

### 3.5.11 T16N0/T16N1/T16N2/T16N3/T16N4/T16N5 寄存器列表

T16N 寄存器列表		
寄存器名称	寄存器地址	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>		
T16N4 基地址: 4000_3000 <sub>H</sub>		
T16N5 基地址: 4000_3400 <sub>H</sub>		
T16N_CNT	0000 <sub>H</sub>	T16N 计数值寄存器
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_TRG	0028 <sub>H</sub>	T16N 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_TOP	0040 <sub>H</sub>	T16N 计数峰值寄存器

### 3.5.12 T32N0 寄存器列表

T32N 寄存器列表		
寄存器名称	寄存器地址	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_PRECNT	0008 <sub>H</sub>	T32N 预分频器计数值寄存器
T32N_PREMAT	000C <sub>H</sub>	T32N 预分频器计数匹配寄存器
T32N_CON0	0010 <sub>H</sub>	T32N 控制寄存器 0
T32N_CON1	0014 <sub>H</sub>	T32N 控制寄存器 1
T32N_CON2	0018 <sub>H</sub>	T32N 控制寄存器 2
T32N_IE	0020 <sub>H</sub>	T32N 中断使能寄存器
T32N_IF	0024 <sub>H</sub>	T32N 中断标志寄存器
T32N_TRG	0028 <sub>H</sub>	T32N 触发寄存器
T32N_MAT0	0030 <sub>H</sub>	T32N 计数匹配寄存器 0
T32N_MAT1	0034 <sub>H</sub>	T32N 计数匹配寄存器 1
T32N_MAT2	0038 <sub>H</sub>	T32N 计数匹配寄存器 2
T32N_MAT3	003C <sub>H</sub>	T32N 计数匹配寄存器 3

### 3.5.13 UART0/UART1/UART2/UART3/UART4/UART5 寄存器列表

UART 寄存器列表		
寄存器名称	寄存器地址	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>		
UART3 基地址: 4000_6C00 <sub>H</sub>		
UART4 基地址: 4000_7000 <sub>H</sub>		
UART5 基地址: 4000_7400 <sub>H</sub>		
UART_CON	0000 <sub>H</sub>	UART 控制寄存器
UART_BRR	0004 <sub>H</sub>	UART 波特率寄存器
UART_TBW	0008 <sub>H</sub>	UART 发送数据写入寄存器
UART_RBR	000C <sub>H</sub>	UART 接收数据读取寄存器
UART_TB01	0010 <sub>H</sub>	UART 发送缓冲 0/1 寄存器
UART_TB23	0014 <sub>H</sub>	UART 发送缓冲 2/3 寄存器
UART_RB01	0018 <sub>H</sub>	UART 接收缓冲 0/1 寄存器
UART_RB23	001C <sub>H</sub>	UART 接收缓冲 2/3 寄存器
UART_IE	0020 <sub>H</sub>	UART 中断使能寄存器
UART_IF	0024 <sub>H</sub>	UART 中断标志寄存器

### 3.5.14 SPI0 寄存器列表

SPI 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
SPI0 基地址: 4000_8000 <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_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

### 3.5.15 I2C0 寄存器列表

I2C 寄存器列表		
寄存器名称	寄存器地址	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

I2C_CON2	0024 <sub>H</sub>	I2C 控制寄存器 2
----------	-------------------	-------------

### 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 控制和状态寄存器
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
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 60 个 I/O 端口。

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 端口的数字复用功能由端口复用选择寄存器 GPIO\_PAFUNCx/GPIO\_PBFUNCx 配置, 具体复用功能参见《管脚对照表》章节的描述, 当复用为模拟端口功能时, 则不需要进行复用功能选择配置, 而是通过端口方向控制寄存器 GPIO\_PADIR/GPIO\_PBDIR 和端口输入控制寄存器 GPIO\_PAINEB/GPIO\_PBINEB 进行配置, 将端口设置为输入状态, 并且关断数字输入。

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\_PBDATABSR 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\_PBDATABCR 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\_PBDATABRR 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, 部分复用功能的端口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 端口均支持开漏输出, 由相应的端口开漏输出使能寄存器 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。

每个 I/O 端口均支持输入滤波功能, 滤波时间约 20ns, 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, 当滤波使能时, 输入信号的高低电平宽度都必须大于 20ns (推荐至少 50ns, 以提高信号输入的可靠性), 否则会被作为无效信号滤除。使能端口滤波功能, 可提高 I/O 输入信号的抗干扰能力, 但输入信号的生效时间会有约 20ns 的延迟, 可根据具体应用需求进行设置。

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。具体参见《电气特性》章节的描述。

GPIO 端口的特殊功能寄存器可通过 AHB 总线访问,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。

注 1: 因 GPIO 默认状态为输入悬空,在实际应用中,需对不使用的 GPIO 设置为输出 0,并关闭内部上下拉电阻。  
注 2: 本系列芯片总共有 62 个 GPIO,有两个 GPIO (PB28 和 PB29) 用作访问内部 EEPROM,所以芯片实际能支持的 GPIO 最多 60 个,当把其中一个管脚配置为 MRSTN 复位功能时,则能支持的 GPIO 最多 59 个。

## 4.2 结构框图

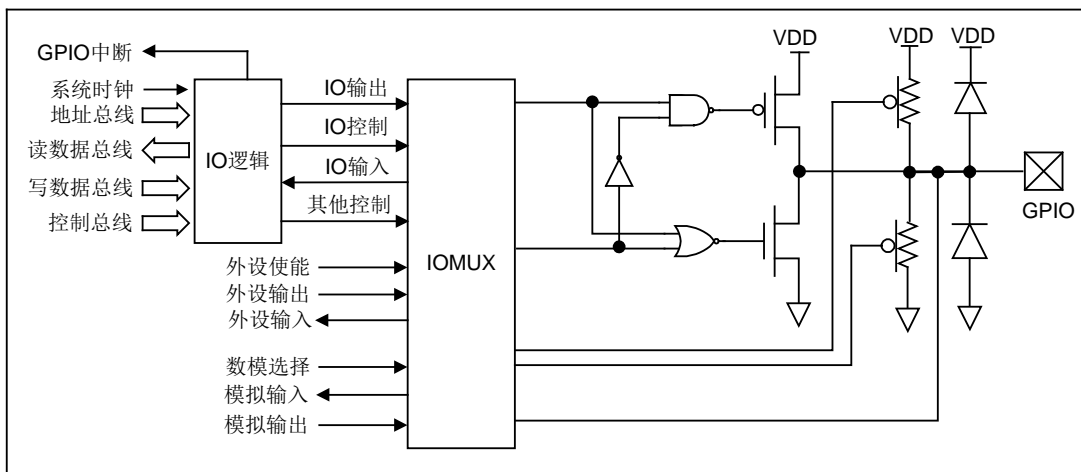


图 4-1 IO 端口电流结构图

## 4.3 外部端口中断

支持全部 IO 口外部端口中断,分成 12 组,每组对应一个外部端口 IRQ 中断请求,并可通过 GPIO\_PINTCFG0、GPIO\_PINTCFG1 寄存器的 PINT11CFG~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。

不同芯片型号对应的 PINT 中断源分组可能不同,共有两种分组,分别为分组 0 和分组 1。

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 IRQ 中断请求,进行配置。



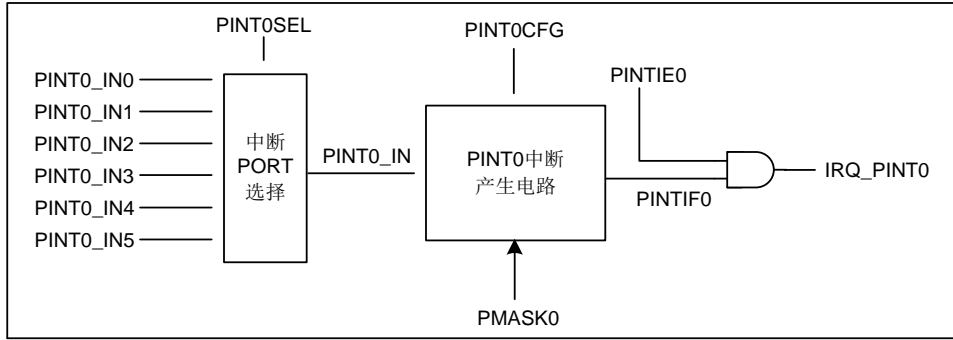


图 4-2 外部端口中断 PINT0 电路结构示意图

上图以外端口中断 PINT0 电路结构示意图为例，可见其有 6 个可选外部中断源，分别来自 PA0, PA8, PA16, PA24, PB0 和 PB8。PINT0 ~PINT11 的外部中断源参见下表：

PINTx	SEL0	SEL1	SEL2	SLE3	SEL4	SEL5
PINT0	PA0	PA8	PA16	PA24	PB0	PB8
PINT1	PA1	PA9	PA17	PA25	PB1	PB9
PINT2	PA2	PA10	PA18	PA26	PB2	PB10
PINT3	PA3	PA11	PA19	PA27	PB3	PB11
PINT4	PA4	PA12	PA20	PA28	PB4	PB12
PINT5	PA5	PA13	PA21	PA29	PB5	PB13
PINT6	PA6	PA14	PA22	PA30	PB6	PB14
PINT7	PA7	PA15	PA23	PA31	PB7	PB15
PINT8	PB16	PB20	PB24	—	—	—
PINT9	PB17	PB21	PB25	—	—	—
PINT10	PB18	PB22	PB26	—	—	—
PINT11	PB19	PB23	PB27	—	—	—

表 4-1 PINT 选择分组 0 对应列表

PINTx	SEL0	SEL1	SEL2	SLE3	SEL4	SEL5
PINT0	PA0	PA8	PA16	PA24	PB16	PB10
PINT1	PA1	PA9	PA17	PA25	PB17	PB20
PINT2	PA2	PA6	PA18	PA26	PB2	PB22
PINT3	PA3	PA11	PA19	PA27	PB8	PB12
PINT4	PA4	PA13	PA20	PA28	PB5	PB14
PINT5	PA5	PA12	PA21	PA29	PB9	PB15
PINT6	PA10	PA14	PA22	PA30	PB4	PB11
PINT7	PA7	PA15	PA23	PA31	PB3	PB13
PINT8	PB0	PB6	PB24	PB28	—	—
PINT9	PB1	PB7	PB25	PB29	—	—
PINT10	PB18	PB21	PB26	—	—	—
PINT11	PB19	PB23	PB27	—	—	—

表 4-2 PINT 选择分组 1 对应列表

使用 PINT 选择分组 1 的芯片型号有 ES8H0394ELLT、ES8H0354ELLR、ES8H0394ELLP、ES8H0354ELLK。

#### 4.4 外部按键中断

支持 1 个 12 输入外部按键中断 (KINT)，12 个按键输入为 KINT0~KINT11，其中任意一个按键输入都可以触发外部按键 IRQ 中断请求，每个按键输入可以从 6 个或 3 个 I/O 端口中选择一个作为其输入源。

不同芯片型号对应的 KINT 中断源分组可能不同，共有两种分组，分别为分组 0 和分组 1。

GPIO\_KINTIE 寄存器的输入屏蔽使能位 KMASK，可对相应的每个输入源 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；屏蔽不使能时，可通过端口中断配置寄存器 GPIO\_KINTCFG0、GPIO\_KINTCFG1，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。

GPIO\_KINTIE 寄存器的按键中断使能位 KINTIE，可对相应的按键标志 KINTIF 是否触发按键 IRQ 中断请求，进行配置。

GPIO\_KINTCFG0 寄存器的按键中断配置位 KINT7CFG~KINT0CFG，GPIO\_KINTCFG1 寄存器的按键中断配置位 KINT11CFG~KINT8CFG，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。

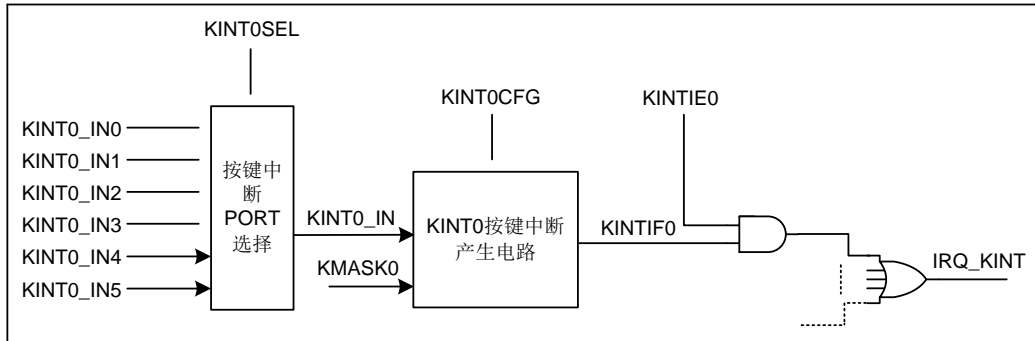


图 4-3 外部按键中断 KINT 电路结构示意图

上图以外部按键中断 KINT0 电路结构示意图为例，可见其有 6 个可选外部中断源，分别来自 PA0、PA8、PA16、PA24、PB0 和 PB8。KINT0 ~KINT11 的按键中断源如下表所示：

KINT <sub>x</sub>	SEL0	SEL1	SEL2	SLE3	SEL4	SEL5
KINT0	PA0	PA8	PA16	PA24	PB0	PB8
KINT1	PA1	PA9	PA17	PA25	PB1	PB9
KINT2	PA2	PA10	PA18	PA26	PB2	PB10
KINT3	PA3	PA11	PA19	PA27	PB3	PB11
KINT4	PA4	PA12	PA20	PA28	PB4	PB12
KINT5	PA5	PA13	PA21	PA29	PB5	PB13
KINT6	PA6	PA14	PA22	PA30	PB6	PB14

KINT7	PA7	PA15	PA23	PA31	PB7	PB15
KINT8	PB16	PB20	PB24	PB28	—	—
KINT9	PB17	PB21	PB25	PB29	—	—
KINT10	PB18	PB22	PB26	—	—	—
KINT11	PB19	PB23	PB27	—	—	—

表 4-3 KINT 选择分组 0 对应列表

KINTx	SEL0	SEL1	SEL2	SLE3	SEL4	SEL5
KINT0	PA0	PA8	PA16	PA24	PB16	PB10
KINT1	PA1	PA9	PA17	PA25	PB17	PB20
KINT2	PA2	PA6	PA18	PA26	PB2	PB22
KINT3	PA3	PA11	PA19	PA27	PB8	PB12
KINT4	PA4	PA13	PA20	PA28	PB5	PB14
KINT5	PA5	PA12	PA21	PA29	PB9	PB15
KINT6	PA10	PA14	PA22	PA30	PB4	PB11
KINT7	PA7	PA15	PA23	PA31	PB3	PB13
KINT8	PB0	PB6	PB24	PB28	—	—
KINT9	PB1	PB7	PB25	PB29	—	—
KINT10	PB18	PB21	PB26	—	—	—
KINT11	PB19	PB23	PB27	—	—	—

表 4-4 KINT 选择分组 1 对应列表

使用 KINT 选择分组 1 的芯片型号有 ES8H0394ELLT、ES8H0354ELLR、ES8H0394ELLP、ES8H0354ELLK。

## 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。可通过 BUZEN 位使能 Buzz，由外设时钟 PCLK 分频产生，并通过 BUZ\_LOAD 计数装载值位设置其分频比，当分频计数器计数 BUZ\_LOAD+1 个 PCLK 时钟周期时，Buzz 输出翻转，同时分频计数器清零并重新递增计数。BUZ\_LOAD 计数装载值不支持缓冲器，其写入值即时生效。Buzz 信号频率计算公式为：

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

可通过 GPIO\_PAFUNC 和 GPIO\_PBFUNC 寄存器来设定 Buzz 信号输出至某个 IO 端口，具体参见《管脚对照表》章节的描述。使用 Buzz 功能时，对应 IO 端口复用为 Buzz 功能后，固定为输出，与该 IO 端口的方向控制寄存器无关。Buzz 启动时的 IO 端口初始电平为 0；Buzz 停止时，关闭 Buzz 使能（BUZEN=0），分频计数器被清 0，IO 端口保持停止前的状态。

GPIO\_TXPWM 寄存器的 TX0PS~TX3PS 控制位设置为 01 时，Buzz 的固定频率输出信号还

可被 UART0/1/2/3 的输出信号 TXD0/1/2/3 调制为 TXPWM0/1/2/3 信号，并送至 GPIO\_TXPWM 寄存器的 TX0\_S~TX3\_S 位选择的相应输出端口上。可通过 GPIO\_TXPWM 寄存器的 TX0PLV~TX3PLV 控制位设定为高电平调制还是低电平调制。

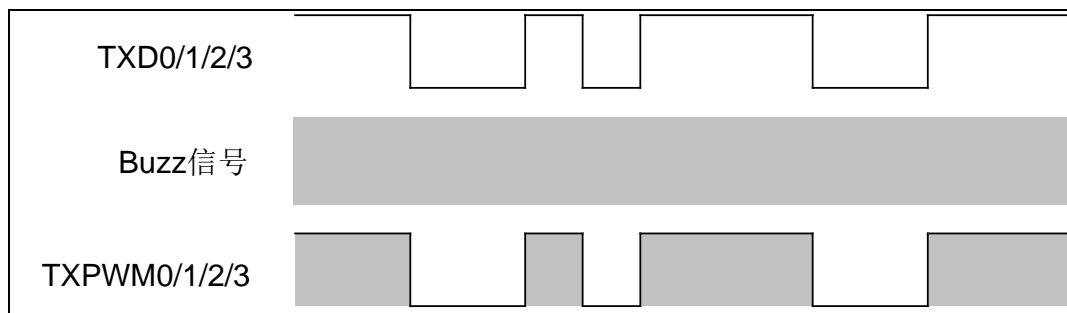


图 4-4 Buzz 高电平调制输出波形图

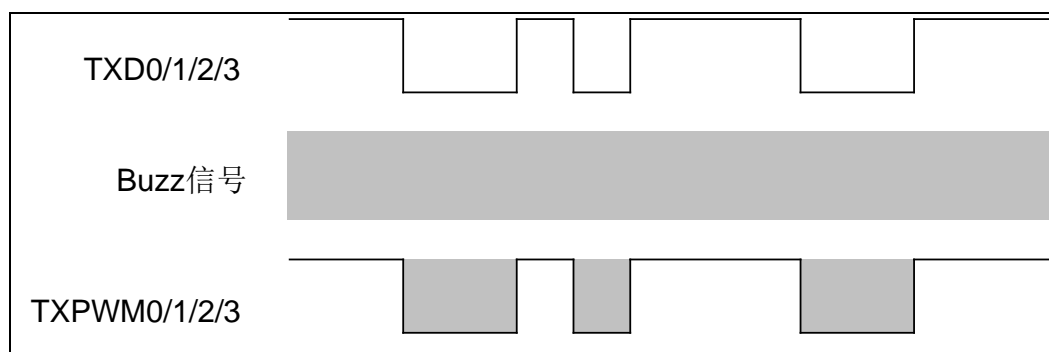


图 4-5 Buzz 低电平调制输出波形图

## 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 端口电平状态 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: 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<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 输出置位寄存器 (GPIO_PADATABSR)															
偏移地址: 14 <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											

注：GPIO\_PADATABSR 寄存器仅支持 Word 写入的方式，读数为 0，对其进行 Bitband 操作无效。

#### 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：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
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
---------------	----------	---	--

注：GPIO\_PADATABCR 寄存器仅支持 Word 写入的方式，读数为 0，对其进行 Bitband 操作无效。

#### 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：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
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
---------------	----------	---	---

注：GPIO\_PADATABRR 寄存器仅支持 Word 写入的方式，读数为 0，对其进行 Bitband 操作无效。

#### 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：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
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>	bit 31-0	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 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）

<b>PA 端口方向置位寄存器（GPIO_PADIRBSR）</b>															
偏移地址： 24 <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 GPIO_PADIR 位设置为 1
--------------	----------	---	--

注： GPIO\_PADIRBSR 寄存器仅支持 Word 写入的方式，读出为 0，对其进行 Bitband 操作无效。

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

<b>PA 端口方向清零寄存器（GPIO_PADIRBCR）</b>															
偏移地址： 28 <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 GPIO_PADIR 位设置为 0
--------------	----------	---	---

注： GPIO\_PADIRBCR 寄存器仅支持 Word 写入的方式，读出为 0，对其进行 Bitband 操作无效。

#### 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: 2C <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
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 GPIO_PADIR 位值翻转
--------------	----------	---	---

注: GPIO\_PADIRBRR 寄存器仅支持 Word 写入的方式, 读出为 0, 对其进行 Bitband 操作无效。

#### 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: 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	PA7<2:0>		保留	PA6<2:0>		保留	PA5<2:0>		保留	PA4<2:0>		保留	PA3<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PA3<2:0>		保留	PA2<2:0>		保留	PA1<2:0>		保留	PA0<2:0>		保留	PA0<2:0>		保留

—	bit31	—	—
PA7<2:0>	bit30-28	R/W	<b>PA7 复用选择位</b> 000~111: FUN0~FUN7
—	bit27	—	—
PA6<2:0>	bit26-24	R/W	<b>PA6 复用选择位</b> 000~111: FUN0~FUN7
—	bit23	—	—
PA5<2:0>	bit22-20	R/W	<b>PA5 复用选择位</b> 000~111: FUN0~FUN7
—	bit19	—	—
PA4<2:0>	bit18-16	R/W	<b>PA4 复用选择位</b> 000~111: FUN0~FUN7
—	bit15	—	—
PA3<2:0>	bit14-12	R/W	<b>PA3 复用选择位</b> 000~111: FUN0~FUN7
—	bit11	—	—
PA2<2:0>	bit10-8	R/W	<b>PA2 复用选择位</b>



			000~111: FUN0~FUN7
—	bit7	—	—
PA1<2:0>	bit6-4	R/W	<b>PA1 复用选择位</b> 000~111: FUN0~FUN7
—	bit3	—	—
PA0<2:0>	bit2-0	R/W	<b>PA0 复用选择位</b> 000~111: FUN0~FUN7

#### 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: 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	PA15<2:0>		保留	PA14<2:0>		保留	PA13<2:0>		保留	PA12<2:0>		保留	PA11<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PA11<2:0>		保留	PA10<2:0>		保留	PA9<2:0>		保留	PA8<2:0>		保留	PA7<2:0>		保留

—	bit31	—	—
PA15<2:0>	bit30-28	R/W	<b>PA15 复用选择位</b> 000~111: FUN0~FUN7
—	bit27	—	—
PA14<2:0>	bit26-24	R/W	<b>PA14 复用选择位</b> 000~111: FUN0~FUN7
—	bit23	—	—
PA13<2:0>	bit22-20	R/W	<b>PA13 复用选择位</b> 000~111: FUN0~FUN7
—	bit19	—	—
PA12<2:0>	bit18-16	R/W	<b>PA12 复用选择位</b> 000~111: FUN0~FUN7
—	bit15	—	—
PA11<2:0>	bit14-12	R/W	<b>PA11 复用选择位</b> 000~111: FUN0~FUN7
—	bit11	—	—
PA10<2:0>	bit10-8	R/W	<b>PA10 复用选择位</b> 000~111: FUN0~FUN7
—	bit7	—	—
PA9<2:0>	bit6-4	R/W	<b>PA9 复用选择位</b> 000~111: FUN0~FUN7
—	bit3	—	—
PA8<2:0>	bit2-0	R/W	<b>PA8 复用选择位</b> 000~111: FUN0~FUN7

#### 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: 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	PA23<2:0>		保留	PA22<2:0>		保留	PA21<2:0>		保留	PA20<2:0>		保留	PA19<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PA19<2:0>		保留	PA18<2:0>		保留	PA17<2:0>		保留	PA16<2:0>		保留	PA16<2:0>		保留

—	bit31	—	—
PA23<2:0>	bit30-28	R/W	<b>PA23 复用选择位</b> 000~111: FUN0~FUN7
—	bit27	—	—
PA22<2:0>	bit26-24	R/W	<b>PA22 复用选择位</b> 000~111: FUN0~FUN7
—	bit23	—	—
PA21<2:0>	bit22-20	R/W	<b>PA21 复用选择位</b> 000~111: FUN0~FUN7
—	bit19	—	—
PA20<2:0>	bit18-16	R/W	<b>PA20 复用选择位</b> 000~111: FUN0~FUN7
—	bit15	—	—
PA19<2:0>	bit14-12	R/W	<b>PA19 复用选择位</b> 000~111: FUN0~FUN7
—	bit11	—	—
PA18<2:0>	bit10-8	R/W	<b>PA18 复用选择位</b> 000~111: FUN0~FUN7
—	bit7	—	—
PA17<2:0>	bit6-4	R/W	<b>PA17 复用选择位</b> 000~111: FUN0~FUN7
—	bit3	—	—
PA16<2:0>	bit2-0	R/W	<b>PA16 复用选择位</b> 000~111: FUN0~FUN7

#### 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: 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
保留	PA31<2:0>		保留	PA30<2:0>		保留	PA29<2:0>		保留	PA28<2:0>		保留	PA28<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

保留	PA27<2:0>	保留	PA26<2:0>	保留	PA25<2:0>	保留	PA24<2:0>
----	-----------	----	-----------	----	-----------	----	-----------

—	bit31	—	—
PA31<2:0>	bit30-28	R/W	<b>PA31 复用选择位</b> 000~111: FUN0~FUN7
—	bit27	—	—
PA30<2:0>	bit26-24	R/W	<b>PA30 复用选择位</b> 000~111: FUN0~FUN7
—	bit23	—	—
PA29<2:0>	bit22-20	R/W	<b>PA29 复用选择位</b> 000~111: FUN0~FUN7
—	bit19	—	—
PA28<2:0>	bit18-16	R/W	<b>PA28 复用选择位</b> 000~111: FUN0~FUN7
—	bit15	—	—
PA27<2:0>	bit14-12	R/W	<b>PA27 复用选择位</b> 000~111: FUN0~FUN7
—	bit11	—	—
PA26<2:0>	bit10-8	R/W	<b>PA26 复用选择位</b> 000~111: FUN0~FUN7
—	bit7	—	—
PA25<2:0>	bit6-4	R/W	<b>PA25 复用选择位</b> 000~111: FUN0~FUN7
—	bit3	—	—
PA24<2:0>	bit2-0	R/W	<b>PA24 复用选择位</b> 000~111: FUN0~FUN7

注：GPIO\_PAFUNC 寄存器仅用于端口的数字输入输出功能选择，具体功能选择参见《管脚对照表》章节的描述。  
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：40H															
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		端口数字输入功能使能位									

			0: 开启 1: 关断
--	--	--	----------------

#### 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: 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
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	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
-----------	---------	-----	---

#### 4.6.16 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: 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
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: 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
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: 50<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	<b>端口输出驱动能力选择位</b> 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 端口类型寄存器 (GPIO\_PATYP)**

偏移地址: 54<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	<b>PA 端口类型选择位</b> 0: CMOS 输入 1: TTL 输入
-----------	---------	-----	--

#### 4.6.20 PA端口滤波控制寄存器 (GPIO\_PAFLT)

**PA 端口 20ns 滤波寄存器 (GPIO\_PAFLT)**

偏移地址: 58<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	<b>PA 端口 20ns 滤波使能位</b> 0: 禁止 1: 使能
-----------	---------	-----	---

注：端口输入滤波使能时，推荐输入信号的高低电平宽度均至少为 50ns，以提高信号输入的可靠性，避免被误滤除。

#### 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_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PORT<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORT<15:0>															
—		bit31-30		—		—									
PORT<29:0>				bit29-0		R		PB 端口电平状态 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: 90 <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		DATA<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA<15:0>															
—		bit31-30		—		—									
DATA<29:0>				bit29-0		R/W		PB 端口输出电平 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: 94 <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		DATABSR<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATABSR<15:0>															
—		bit31-30		—		—									

DATABSR<29:0>	bit29-0	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
---------------	---------	---	--

注：GPIO\_PBDATABSR 寄存器仅支持 Word 写入的方式，读出为 0，对其进行 Bitband 操作无效。

#### 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: 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		DATABCR<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATABCR<15:0>															

—	bit31-30	—	—
DATABCR<29:0>	bit29-0	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

注：GPIO\_PBDATABCR 寄存器仅支持 Word 写入的方式，读出为 0，对其进行 Bitband 操作无效。

#### 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: 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		DATABRR<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATABRR<15:0>															

—	bit31-30	—	—
DATABRR<29:0>	bit29-0	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

注：GPIO\_PBDATABRR 寄存器仅支持 Word 写入的方式，读出为 0，对其进行 Bitband 操作无效。

#### 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: A0 <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		DIR<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR<15:0>															

—	bit31-30	—	—
DIR<29:0>	bit29-0	R/W	<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 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: A4 <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		DIRBSR<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRBSR<15:0>															

—	bit31-30	—	—
DIRBSR<29:0>	bit29-0	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

注: GPIO\_PBDIRBSR 寄存器仅支持 Word 写入的方式, 读出为 0, 对其进行 Bitband 操作无效。

#### 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: A8 <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		DIRBCR<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



DIRBCR<15:0>			
--------------	--	--	--

—	bit31-30	—	—
DIRBCR<29:0>	bit29-0	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

注： GPIO\_PBDIRBCR 寄存器仅支持 Word 写入的方式，读出为 0，对其进行 Bitband 操作无效。

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

—	bit31-30	—	—
DIRBRR<29:0>	bit29-0	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

注： GPIO\_PBDIRBRR 寄存器仅支持 Word 写入的方式，读出为 0，对其进行 Bitband 操作无效。

#### 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：B0 <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<2:0>		保留		PB6<2:0>		保留		PB5<2:0>		保留		PB4<2:0>	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		PB3<2:0>		保留		PB2<2:0>		保留		PB <2:0>		保留		PB0<2:0>	

—	bit31	—	—
PB7<2:0>	bit30-28	R/W	<b>PB7 复用选择位</b> 000~111: FUN0~FUN7
—	bit27	—	—
PB6<2:0>	bit26-24	R/W	<b>PB6 复用选择位</b> 000~111: FUN0~FUN7

—	bit23	—	—
PB5<2:0>	bit22-20	R/W	<b>PB5 复用选择位</b> 000~111: FUN0~FUN7
—	bit19	—	—
PB4<2:0>	bit18-16	R/W	<b>PB4 复用选择位</b> 000~111: FUN0~FUN7
—	bit15	—	—
PB3<2:0>	bit14-12	R/W	<b>PB3 复用选择位</b> 000~111: FUN0~FUN7
—	bit11	—	—
PB2<2:0>	bit10-8	R/W	<b>PB2 复用选择位</b> 000~111: FUN0~FUN7
—	bit7	—	—
PB1<2:0>	bit6-4	R/W	<b>PB1 复用选择位</b> 000~111: FUN0~FUN7
—	bit3	—	—
PB0<2:0>	bit2-0	R/W	<b>PB0 复用选择位</b> 000~111: FUN0~FUN7

#### 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: 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	PB15<2:0>		保留	PB14<2:0>		保留	PB13<2:0>		保留	PB12<2:0>					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PB11<2:0>		保留	PB10<2:0>		保留	PB9<2:0>		保留	PB8<2:0>					

—	bit31	—	—
PB15<2:0>	bit30-28	R/W	<b>PB15 复用选择位</b> 000~111: FUN0~FUN7
—	bit27	—	—
PB14<2:0>	bit26-24	R/W	<b>PB14 复用选择位</b> 000~111: FUN0~FUN7
—	bit23	—	—
PB13<2:0>	bit22-20	R/W	<b>PB13 复用选择位</b> 000~111: FUN0~FUN7
—	bit19	—	—
PB12<2:0>	bit18-16	R/W	<b>PB12 复用选择位</b> 000~111: FUN0~FUN7
—	bit15	—	—

PB11<2:0>	bit14-12	R/W	<b>PB11 复用选择位</b> 000~111: FUN0~FUN7
—	bit11	—	—
PB10<2:0>	bit10-8	R/W	<b>PB10 复用选择位</b> 000~111: FUN0~FUN7
—	bit7	—	—
PB9<2:0>	bit6-4	R/W	<b>PB9 复用选择位</b> 000~111: FUN0~FUN7
—	bit3	—	—
PB8<2:0>	bit2-0	R/W	<b>PB8 复用选择位</b> 000~111: FUN0~FUN7

#### 4.6.32 PB<23:16>端口复用选择寄存器 (GPIO\_PBFUNC2)

PB<23:16>端口复用选择寄存器 (GPIO_PBFUNC2)															
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	PB23<2:0>			保留	PB22<2:0>			保留	PB21<2:0>			保留	PB20<2:0>		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PB19<2:0>			保留	PB18<2:0>			保留	PB17<2:0>			保留	PB16<2:0>		

—	bit31	—	—
PB23<2:0>	bit30-28	R/W	<b>PB23 复用选择位</b> 000~111: FUN0~FUN7
—	bit27	—	—
PB22<2:0>	bit26-24	R/W	<b>PB22 复用选择位</b> 000~111: FUN0~FUN7
—	bit23	—	—
PB21<2:0>	bit22-20	R/W	<b>PB21 复用选择位</b> 000~111: FUN0~FUN7
—	bit19	—	—
PB20<2:0>	bit18-16	R/W	<b>PB20 复用选择位</b> 000~111: FUN0~FUN7
—	bit15	—	—
PB19<2:0>	bit14-12	R/W	<b>PB19 复用选择位</b> 000~111: FUN0~FUN7
—	bit11	—	—
PB18<2:0>	bit10-8	R/W	<b>PB18 复用选择位</b> 000~111: FUN0~FUN7
—	bit7	—	—
PB17<2:0>	bit6-4	R/W	<b>PB17 复用选择位</b> 000~111: FUN0~FUN7

—	bit3	—	—
PB16<2:0>	bit2-0	R/W	<b>PB16 复用选择位</b> 000~111: FUN0~FUN7

#### 4.6.33 PB<27:24>端口复用选择寄存器 (GPIO\_PBFUNC3)

PB<27:24>端口复用选择寄存器 (GPIO_PBFUNC3)															
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	PB27<2:0>		保留	PB26<2:0>		保留	PB25<2:0>		保留	PB24<2:0>					

—	Bit31-27	—	—
PB30<2:0>	bit26-24	R/W	<b>PB30 复用选择位</b> 000~111: FUN0~FUN7
—	bit23	—	—
PB29<2:0>	bit22-20	R/W	<b>PB29 复用选择位</b> 000~111: FUN0~FUN7
—	bit19	—	—
PB28<2:0>	bit18-16	R/W	<b>PB28 复用选择位</b> 000~111: FUN0~FUN7
—	bit15	—	—
PB27<2:0>	bit14-12	R/W	<b>PB27 复用选择位</b> 000~111: FUN0~FUN7
—	bit11	—	—
PB26<2:0>	bit10-8	R/W	<b>PB26 复用选择位</b> 000~111: FUN0~FUN7
—	bit7	—	—
PB25<2:0>	bit6-4	R/W	<b>PB25 复用选择位</b> 000~111: FUN0~FUN7
—	bit3	—	—
PB24<2:0>	bit2-0	R/W	<b>PB24 复用选择位</b> 000~111: FUN0~FUN7

注: GPIO\_PBFUNC 仅用于端口的数字输入输出功能选择, 具体功能选择参见《管脚对照表》章节的描述。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。

#### 4.6.34 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: 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		INEB<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEB<15:0>															

—	bit31-30	—	—
INEB<29:0>	bit29-0	R/W	端口数字输入功能使能位 0: 使能 1: 禁止

#### 4.6.35 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: 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		ODE<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODE<15:0>															

—	bit31-30	—	—
ODE<29:0>	bit29-0	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

#### 4.6.36 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: 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		PUEN<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUEN<15:0>															

—	bit31-30	—	—
PUEN<29:0>	bit29-0	R/W	端口弱上拉使能位 0: 禁止 1: 使能

#### 4.6.37 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: 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		PDEN<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDEN<15:0>															

—	bit31-30	—	—
PDEN<29:0>	bit29-0	R/W	端口弱下拉使能位 0: 禁止 1: 使能

#### 4.6.38 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: D0 <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		DS<29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS<15:0>															

—	bit31-30	—	—
DS<29:0>	bit29-0	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

#### 4.6.39 PB端口类型选择寄存器 (GPIO\_PBTYP)

PB 端口类型寄存器 (GPIO_PBTYP)															
偏移地址: D4 <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		TYP <29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYP <15:0>															

—	bit31-30	—	—
TYP <29:0>	bit29-0	R/W	PB 端口类型选择位 0: CMOS 输入 1: TTL 输入

#### 4.6.40 PB端口滤波控制寄存器 (GPIO\_PBFLT)

PB 端口 20ns 滤波寄存器 (GPIO\_PBFLT)

偏移地址: D8<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		FLT <29:16>													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT <15:0>															

—	bit31-30	—	—
FLT <29:0>	bit29-0	R/W	<b>PB 端口 20ns 滤波使能位</b> 0: 禁止 1: 使能

注: 端口输入滤波使能时, 推荐输入信号的高低电平宽度均至少为 50ns, 以提高信号输入的可靠性, 避免被误滤除。

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

PINT 中断使能寄存器 (GPIO\_PINTIE)

偏移地址: 300<sub>H</sub>

复位值: 00000000\_11110000\_11111111\_00000000<sub>B</sub>

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

—	bit31-24	—	—
PMASK<11:8>	bit23-20	R/W	<b>PINT11~8 中断源屏蔽控制位</b> 0: 不屏蔽 1: 屏蔽
PINTIE<11:8>	bit19-16	R/W	<b>PINT11~8 使能位</b> 0: 禁止 1: 使能
PMASK<7:0>	bit15-8	R/W	<b>PINT7~0 中断源屏蔽控制位</b> 0: 不屏蔽 1: 屏蔽
PINTIE<7:0>	bit7-0	R/W	<b>PINT7~0 使能位</b> 0: 禁止 1: 使能

#### 4.6.42 PINT中断标志寄存器 (GPIO\_PINTIF)

PINT 中断标志寄存器 (GPIO_PINTIF)															
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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					PINTIF<11:0>										

—	bit 31-12	—	—
PINTIF<11:0>	bit11-0	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。

#### 4.6.43 PINT0~7 中断源选择寄存器 (GPIO\_PINTSEL0)

PINT0~7 中断源选择寄存器 (GPIO_PINTSEL0)															
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
保留	PINT7<2:0>		保留	PINT6<2:0>		保留	PINT5<2:0>		保留	PINT4<2:0>					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PINT3<2:0>		保留	PINT2<2:0>		保留	PINT1<2:0>		保留	PINT0<2:0>					

—	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

#### 4.6.44 PINT8~11 中断源选择寄存器 (GPIO\_PINTSEL1)

PINT8~11 中断源选择寄存器 (GPIO_PINTSEL1)															
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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PINT11<2:0>			保留	PINT10<2:0>			保留	PINT9<2:0>			保留	PINT8<2:0>		

—	bit31-15	—	—
PINT11SEL<2:0>	bit14-12	R/W	<b>PINT11 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit11	—	—
PINT10SEL<2:0>	bit10-8	R/W	<b>PINT10 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit7	—	—
PINT9SEL<2:0>	bit6-4	R/W	<b>PINT9 输入选择位</b> 000~101: SEL0~SEL5 其余: SEL0
—	bit3	—	—
PINT8SEL<2:0>	bit2-0	R/W	<b>PINT8 输入选择位</b>

			000~101: SEL0~SEL5 其余: SEL0
--	--	--	--------------------------------

#### 4.6.45 PINT0~7 中断配置寄存器 (GPIO\_PINTCFG0)

PINT 中断配置寄存器 0 (GPIO_PINTCFG0)															
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
保留	PINT7<2:0>			保留	PINT6<2:0>			保留	PINT5<2:0>			保留	PINT4<2:0>		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PINT3<2:0>			保留	PINT2<2:0>			保留	PINT1<2:0>			保留	PINT0<2:0>		

—	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

#### 4.6.46 PINT8~11 中断配置寄存器 (GPIO\_PINTCFG1)

PINT 中断配置寄存器 1 (GPIO_PINTCFG1)															
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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PINT11<2:0>		保留	PINT10<2:0>		保留	PINT9<2:0>		保留	PINT8<2:0>					

—	bit31-15	—	—
PINT11CFG<2:0>	bit14-12	R/W	<b>PINT11 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit11	—	—
PINT10CFG<2:0>	bit10-8	R/W	<b>PINT10 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit7	—	—
PINT9CFG<2:0>	bit6-4	R/W	<b>PINT9 配置位</b> 参见 GPIO_PINTCFG 详情
—	bit3	—	—
PINT8CFG<2:0>	bit2-0	R/W	<b>PINT8 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

#### 4.6.47 KINT中断使能寄存器 (GPIO\_KINTIE)

KINT 中断使能寄存器 (GPIO_KINTIE)															
偏移地址: 318 <sub>H</sub>															
复位值: 00000000_11110000_11111111_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								KMASK<11:8>				KINTIE<11:8>			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KMASK<7:0>								KINTIE<7:0>							

—	bit31-24	—	—
KMASK<11:8>	bit23-20	R/W	<b>KINT11~8 中断源屏蔽控制位</b>

			0: 不屏蔽 1: 屏蔽
KINTIE<11:8>	bit19-16	R/W	<b>KINT11~8 使能位</b> 0: 禁止 1: 使能
KMASK<7:0>	bit15-8	R/W	<b>KINT7~0 中断源屏蔽控制位</b> 0: 不屏蔽 1: 屏蔽
KINTIE<7:0>	bit7-0	R/W	<b>KINT7~0 使能位</b> 0: 禁止 1: 使能

#### 4.6.48 KINT中断标志寄存器 (GPIO\_KINTIF)

<b>KINT 中断标志寄存器 (GPIO_KINTIF)</b>															
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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				KINTIF<11:0>											
—		bit 31-12		—		—									
KINTIF<11:0>		bit11-0		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。

#### 4.6.49 KINT0~7 中断源选择寄存器 (GPIO\_KINTSEL0)

<b>KINT0~7 中断源选择寄存器 (GPIO_KINTSEL0)</b>															
偏移地址: 320 <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	KINT7<2:0>		保留	KINT6<2:0>		保留	KINT5<2:0>		保留	KINT4<2:0>		保留	KINT3<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	KINT3<2:0>		保留	KINT2<2:0>		保留	KINT1<2:0>		保留	KINT0<2:0>		保留	KINT0<2:0>		保留
—		bit31		—		—									
KINT7SEL<2:0>		bit30-28		R/W		<b>KINT7 输入选择位</b>									

—	bit27	—	000~101: SEL0~SEL5
KINT6SEL<2:0>	bit26-24	R/W	<b>KINT6</b> 输入选择位 000~101: SEL0~SEL5
—	bit23	—	—
KINT5SEL<2:0>	bit22-20	R/W	<b>KINT5</b> 输入选择位 000~101: SEL0~SEL5
—	bit19	—	—
KINT4SEL<2:0>	bit18-16	R/W	<b>KINT4</b> 输入选择位 000~101: SEL0~SEL5
—	bit15	—	—
KINT3SEL<2:0>	bit14-12	R/W	<b>KINT3</b> 输入选择位 000~101: SEL0~SEL5
—	bit11	—	—
KINT2SEL<2:0>	bit10-8	R/W	<b>KINT2</b> 输入选择位 000~101: SEL0~SEL5
—	bit7	—	—
KINT1SEL<2:0>	bit6-4	R/W	<b>KINT1</b> 输入选择位 000~101: SEL0~SEL5
—	bit3	—	—
KINT0SEL<2:0>	bit2-0	R/W	<b>KINT0</b> 输入选择位 000~101: SEL0~SEL5

#### 4.6.50 KINT8~11 中断源选择寄存器 (GPIO\_KINTSEL1)

KINT8~11 中断源选择寄存器 (GPIO_KINTSEL1)															
偏移地址: 324 <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	KINT11<2:0>		保留	KINT10<2:0>		保留	KINT9<2:0>		保留	KINT8<2:0>					

—	bit31-15	—	—
KINT11SEL<2:0>	bit14-12	R/W	<b>KINT11</b> 输入选择位 000~101: SEL0~SEL5
—	bit11	—	—
KINT10SEL<2:0>	bit10-8	R/W	<b>KINT10</b> 输入选择位 000~101: SEL0~SEL5
—	bit7	—	—
KINT9SEL<2:0>	bit6-4	R/W	<b>KINT9</b> 输入选择位 000~101: SEL0~SEL5
—	bit3	—	—

KINT8SEL<2:0>	bit2-0	R/W	<b>KINT8 输入选择位</b> 000~101: SEL0~SEL5
---------------	--------	-----	--

#### 4.6.51 KINT0~7 中断配置寄存器 (GPIO\_KINTCFG0)

KINT0~7 中断配置寄存器 (GPIO_KINTCFG0)															
偏移地址: 328 <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	KINT7<2:0>		保留	KINT6<2:0>		保留	KINT5<2:0>		保留	KINT4<2:0>		保留	KINT3<2:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	KINT3<2:0>		保留	KINT2<2:0>		保留	KINT1<2:0>		保留	KINT0<2:0>		保留	KINT0<2:0>		保留

—	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

#### 4.6.52 KINT8~11 中断配置寄存器 (GPIO\_KINTCFG1)

KINT8~11 中断配置寄存器 (GPIO_KINTCFG1)															
偏移地址: 32C <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	KINT11<2:0>		保留	KINT10<2:0>		保留	KINT9<2:0>		保留	KINT8<2:0>					

—	bit31-15	—	—
KINT11CFG<2:0>	bit14-12	R/W	<b>KINT11 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit11	—	—
KINT10CFG<2:0>	bit10-8	R/W	<b>KINT10 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit7	—	—
KINT9CFG<2:0>	bit6-4	R/W	<b>KINT9 配置位</b> 参见 GPIO_KINTCFG 详情
—	bit3	—	—
KINT8CFG<2:0>	bit2-0	R/W	<b>KINT8 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

#### 4.6.53 PINT中断滤波控制寄存器 (GPIO\_FLTCR)

PINT 中断滤波控制寄存器 (GPIO_FLTCR)															
偏移地址: 330 <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										FLTCLK<2:0>			FLTCNT<2:0>		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				FLTEN <11:0>											

—	bit31-22	—	—
FLTCLK<2:0>	bit21-19	R/W	<b>PINT 滤波时钟选择位</b>

			000: LRC 001: HCLK/2 010: HCLK/4 011: HCLK/8 100: HCLK/16 101: HCLK/32 110: HCLK/64 111: HCLK/128
FLTCNT<2:0>	bit18-16	R/W	<b>PINT 滤波时间选择位</b> 000: 1 个滤波时钟周期 001: 2 个滤波时钟周期 010: 3 个滤波时钟周期 011: 4 个滤波时钟周期 100: 5 个滤波时钟周期 101: 6 个滤波时钟周期 110: 7 个滤波时钟周期 111: 8 个滤波时钟周期 注: 因 PINT 中断为异步信号, 实际滤波时间可能会比设定值多 1 个滤波时钟周期, 滤波时间建议选择 3 个滤波时钟周期以上, 可根据实际应用需求进行调整
—	bit15-12	—	—
FLTEN<11:0>	bit11-0	R/W	<b>PINT11~PINT0 滤波使能位</b> 0: 禁止 1: 使能 注: 可对 PINT11~PINT0 分别设置滤波使能与否

注: 深度休眠后若使能 PINT 中断唤醒, 则不能使用 HCLK 作为滤波时钟, 因该时钟在深度休眠时会自动关闭。

#### 4.6.54 脉宽调制寄存器 (GPIO\_TXPWM)

脉宽调制寄存器 (GPIO\_TXPWM)

偏移地址: 380<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	TX3_S<1:0>	TX3PLV	保留	TX3PS<1:0>	保留	TX2_S<1:0>	TX2PLV	保留	TX2PS<1:0>						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TX1_S<1:0>	TX1PLV	保留	TX1PS<1:0>	保留	TX0_S<1:0>	TX0PLV	保留	TX0PS<1:0>						



—	bit31	—	—
TX3_S<1:0>	bit30-29	R/W	<b>TXPWM3 调制输出选择</b> 00: TXD3 管脚为 TXPWM3 输出 01: BUZ 管脚为 TXPWM3 输出 10: T16N3OUT0 管脚为 TXPWM3 输出 11: T16N3OUT1 管脚为 TXPWM3 输出
TX3PLV	bit28	R/W	<b>TXPWM3 调制方式选择位</b> 0: 低电平调制 (TXD3 与 TX3PS 所选取的脉冲信号进行硬件或操作) 1: 高电平调制 (TXD3 与 TX3PS 所选取的脉冲信号进行硬件与操作)
—	bit27-26	—	—
TX3PS<1:0>	bit25-24	R/W	<b>TXPWM3 调制 PWM 脉冲选择位</b> 00: 调制禁止 01: 与 BUZ 进行调制 10: 与 T16N3OUT0 进行调制 11: 与 T16N3OUT1 进行调制
—	bit23	—	—
TX2_S<1:0>	bit22-21	R/W	<b>TXPWM2 调制输出选择</b> 00: TXD2 管脚为 TXPWM2 输出 01: BUZ 管脚为 TXPWM2 输出 10: T16N2OUT0 管脚为 TXPWM2 输出 11: T16N2OUT1 管脚为 TXPWM2 输出
TX2PLV	bit20	R/W	<b>TXPWM2 调制方式选择位</b> 0: 低电平调制 (TXD2 与 TX2PS 所选取的脉冲信号进行硬件或操作) 1: 高电平调制 (TXD2 与 TX2PS 所选取的脉冲信号进行硬件与操作)
—	bit19-18	—	—
TX2PS<1:0>	bit17-16	R/W	<b>TXPWM2 调制 PWM 脉冲选择位</b> 00: 调制禁止 01: 与 BUZ 进行调制 10: 与 T16N2OUT0 进行调制 11: 与 T16N2OUT1 进行调制
—	bit15	—	—
TX1_S<1:0>	bit14-13	R/W	<b>TXPWM1 调制输出选择</b> 00: TXD1 管脚为 TXPWM1 输出 01: BUZ 管脚为 TXPWM1 输出 10: T16N1OUT0 管脚为 TXPWM1 输出 11: T16N1OUT1 管脚为 TXPWM1 输出
TX1PLV	bit12	R/W	<b>TXPWM1 调制方式选择位</b> 0: 低电平调制 (TXD1 与 TX1PS 所选取的脉冲信号进行硬件或操作)

			1: 高电平调制 (TXD1 与 TX1PS 所选取的脉冲信号进行硬件与操作)
—	bit11-10	—	—
TX1PS<1:0>	bit9-8	R/W	<b>TXPWM1 调制 PWM 脉冲选择位</b> 00: 调制禁止 01: 与 BUZ 进行调制 10: 与 T16N1OUT0 进行调制 11: 与 T16N1OUT1 进行调制
—	bit7	—	—
TX0_S<1:0>	bit6-5	R/W	<b>TXPWM0 调制输出选择</b> 00: TXD0 管脚为 TXPWM0 输出 01: BUZ 管脚为 TXPWM0 输出 10: T16N0OUT0 管脚为 TXPWM0 输出 11: T16N0OUT1 管脚为 TXPWM0 输出
TX0PLV	bit4	R/W	<b>TXPWM0 调制方式选择位</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)
—	bit3-2	—	—
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 BUZ 进行调制 10: 与 T16N0OUT0 进行调制 11: 与 T16N0OUT1 进行调制

注 1: TXPWM0/1/2/3 输出至 BUZ 管脚的优先级依次为: TXPWM0 > TXPWM1 > TXPWM2 > TXPWM3;

注 2: 调制输出信号 TXPWM 是 UART 的发送信号 TXD 被 BUZ、T16N<sub>x</sub>OUT0、T16N<sub>x</sub>OUT1 调制后的信号。以 TXPWM0 为例, UART0 的 TXD0 可通过 TX0PS 来选择被 BUZ、T16N0OUT 或 T16N1OUT 调制, 调制后的 TXPWM0 输出可通过 TX0\_S 来选择输出至 TXD0 管脚、BUZ 管脚、T16N0OUT 管脚或 T16N0OUT1 管脚。要在 TX0\_S 选择的管脚上观察到 TXPWM0 输出: 必须使能 UART0 TXD 发送数据; TX0PS 为非零值, 其所选取的 BUZ、T16N0OUT0 或 T16N0OUT1 必须有输出波形。

#### 4.6.55 BUZ控制寄存器 (GPIO\_BUZC)

BUZ 控制寄存器 (GPIO_BUZC)															
偏移地址: 390 <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				BUZCFG<1:0>		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： $F_{PCLK} / (2 * (BUZ\_LOAD + 1))$
—	bit7-3	—	—
BUZCFG	bit2-1	R/W	<b>BUZ 禁止时的端口输出配置位</b> 00: 禁止时保持不变 01: 保留 10: 禁止时输出 0 11: 禁止时输出 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 (T16N0~5)

以 T16N0 为例, T16N1/T16N2/T16N3/T16N4/T16N5 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\_CNT 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 寄存器设定预设值
  - ◇ 预分频比由寄存器 T16N\_PREMAT 设定
  - ◇ 支持预分频比缓冲器 T16N\_PREBUF (T16N\_CON2.MATBUFUP=1 使能, 仅调制模式下使用)
- ◆ 1 个 16 位可配置定时/计数寄存器 T16N\_CNT
- ◆ 1 个 16 位峰值寄存器 T16N\_TOP
  - ◇ T16N\_CNT 计数值达到峰值后溢出清零
  - ◇ 支持峰值缓冲器 T16N\_TOPBUF (T16N\_CON2.MATBUFUP=1 使能, 仅调制模式下使用)
- ◆ 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:
    - 产生中断
    - 支持 T16N\_CNT 计数寄存器三种操作: 保持, 清 0 或继续计数
    - 支持 T16N0OUT0/T16N0OUT1 端口四种操作: 保持, 清 0, 置 1 或取反
- ◆ 支持输入捕捉工作模式
  - ◇ 对 T16N0IN0/T16N0IN1 端口输入信号进行捕捉
  - ◇ 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, T16N0OUT0/T16N0OUT1 为输出端口
  - ◇ 支持匹配缓冲器 T16N\_MATBUF0~3 (T16N\_CON2.MATBUFUP=1 使能)
  - ◇ 对匹配寄存器的写操作, 可以设置为写入后立即生效 (寄存器位 T16N\_CON2.MATBUFUP=0 禁止缓冲) 或在写入后的当前计数周期结束后才生效 (寄存器位 T16N\_CON2.MATBUFUP=1 使能缓冲)。
  - ◇ 支持刹车控制
    - 刹车信号源可选择为 PINT0~11 外部端口中断源中的任意一个端口
    - 刹车信号电平极性可配置

- 刹车后端口输出电平可配置
- ◆ 支持计数中触发功能
  - ◇ 在 ADC 使能且硬件控制采样模式下, T16N 计数匹配中断标志 MAT0IF~MAT3IF 和计数溢出中断标志 OVIF 可触发 ADC 转换, 可通过寄存器 T16N\_TRG 的匹配触发使能位 MAT0TAR~MAT3TAR 和溢出触发使能位 OVTAR 设置使能

### 5.1.1.2 结构框图

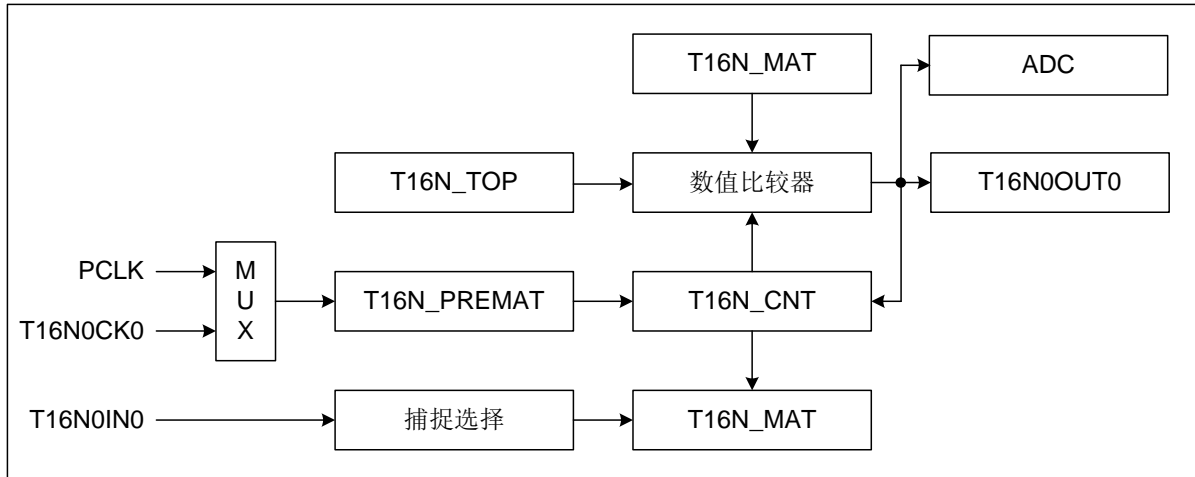


图 5-1 T16N0 电路结构框图

### 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。

设置 T16N\_CON0 寄存器的 EN=1, 使能 T16N, 计数值寄存器 T16N\_CNT 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\_CNT 计数值寄存器的工作状态和是否产生匹配中断。

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

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=01: 当 T16N\_CNT 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\_CNT 不再累加计数，只产生匹配中断 MAT0IF~MAT3IF。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=10：当 T16N\_CNT 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 0，并产生匹配中断 MAT0IF~MAT3IF，重新开始累加计数。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=11：当 T16N\_CNT 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，产生匹配中断 MAT0IF~MAT3IF，当计数到 T16N\_TOP，下一次累加计数溢出，T16N\_CNT 的值为 0x0000，并产生溢出中断 OVIF，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，会产生相应的匹配中断。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，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。

通过 T16N\_IE 寄存器的匹配中断使能位 MAT0IE~MAT3IE 和溢出中断使能位 OVIE，可对匹配中断标志 MAT0IF~MAT3IF 和溢出中断标志 OVIF 是否触发 T16N 中断请求 IRQ，进行设置。

使用外部计数功能时，需要将对应的 IO 端口复用功能设置为 T16N 端口，并且将该 IO 端口的方向控制寄存器设置为输入，作为外部时钟 T16N0CK0/T16N0CK1 输入端口。

**举例说明：T16N\_CNT 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 0，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：

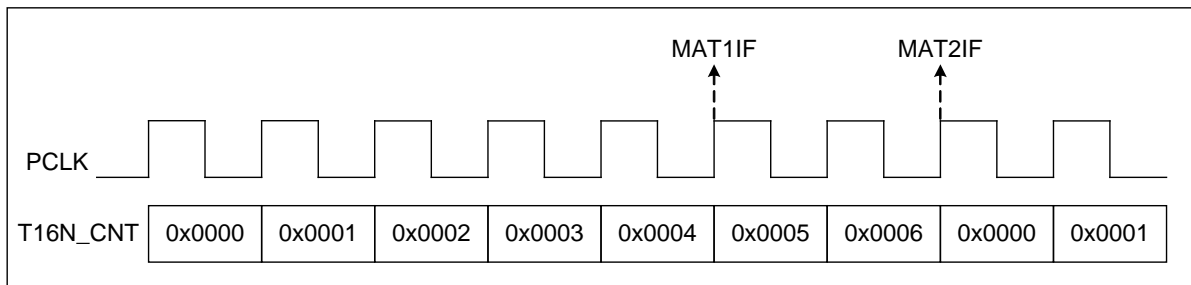


图 5-2 T16N 计数匹配功能示意图

#### 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。

在捕捉工作模式下，需设置 T16N\_CON0 寄存器的 CS<1:0>=00，使 T16N\_CNT 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\_CNT 的工作，也无需产生中断。

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\_CNT 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，可在产生 CAP0IF 中断时，同时将 T16N\_CNT 和 T16N\_PRECNT 清零，计数器重新开始计数；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\_CNT 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，可在产生 CAP1IF 中断时，同时将 T16N\_CNT 和 T16N\_PRECNT 清零，计数器重新开始计数；若 CAPL1 为零，则计数器继续累加。

通过 T16N\_IE 寄存器的捕捉中断使能位 CAP0IE~CAP1IE，可对捕捉中断标志 CAP0IF~CAP1IF 是否触发 T16N 中断请求 IRQ，进行设置。

当 T16N\_CNT 计数到 T16N\_TOP，并在下一个计数时钟（经过预分频之后的时钟）到来时，仍未检测到设定的捕捉事件，T16N\_CNT 的值因计数溢出被清零，产生溢出中断 OVIF，并重新开始累加计数。

设置 T16N\_CON1 寄存器的 CAPPE, CAPNE 和 CAPT，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；

选择捕捉次数：设置 CAPT，可选择捕捉边沿发生的次数，达到所设定的次数后，产生捕捉装载动作。

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。

**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~MAT3S <1:0>=00，

T16N\_CON1 寄存器的 CAPPE=1，CAPNE=1，CAPIS0=1，CAPT<3:0>=0111。

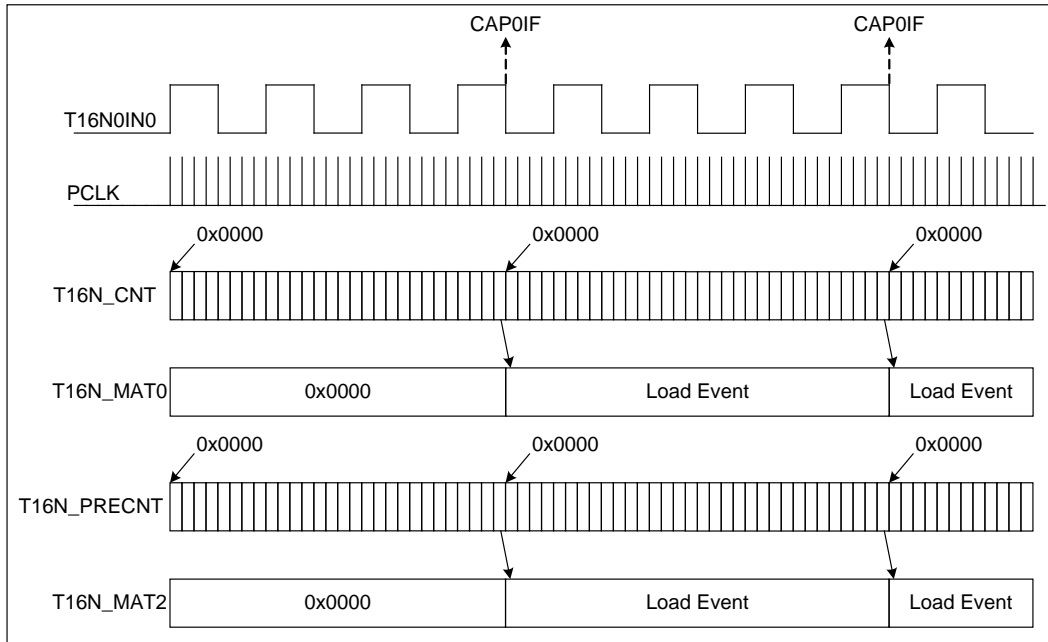


图 5-3 T16N0 捕捉功能示意图

使用捕捉功能时，需要将对应的 IO 端口复用功能设置为 T16N 端口，并且将该 IO 端口的方向控制寄存器设置为输入，作为 T16N0IN0/T16N0IN1 捕捉输入端口。

在捕捉工作过程中，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，因此首次捕捉可能会从一个非零预分频器计数开始，导致首次捕捉的 T16N\_PRECNT 和 T16N\_CNT 值不精确。

当捕捉事件发生时，产生的中断标志位必须通过软件清除，并及时读取捕捉到 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\_CNT 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，使能时启用 T16N\_MAT0/1 匹配寄存器和 T16N\_CNT 进行匹配；设置 T16N\_CON2 寄存器的 MOE1，选择 T16N0OUT1 是否使能为匹配输出端口，使能时启用 T16N\_MAT2/3 匹配寄存器和 T16N\_CNT 进行匹配。

设置 T16N\_CON2 寄存器的 MOM0/MOM1/MOM2/MOM3，选择计数匹配发生时，对 T16N0OUT0/1 端口的影响：保持，清 0，置 1，取反。

设置 T16N\_CON2 寄存器的 POL0/1，选择 T16N0OUT0/1 端口输出的极性。

通过寄存器 T16N\_TOP 可设置 PWM 的波形周期，在计数器计到 T16N\_TOP 后，下一次累加计数溢出，T16N\_CNT 的值为 0x0000，并产生溢出中断 OVIF，重新开始累加计数。



设置寄存器 T16N\_CON2 的 MATBUFUP=1，可使能缓存机制，此时计数器每次计到 T16N\_TOP 后，才将当前 PWM 周期写入的 T16N\_PREMAT、T16N\_MAT0~3 和 T16N\_TOP 重新加载到缓存寄存器，后续 T16N 按照缓存寄存器中的数据产生 PWM 波形，使能缓存机制，可方便实现 PWM 脉冲占空比和周期连续可调。若设置寄存器 T16N\_CON2 的 MATBUFUP=0，将禁止缓存机制，此时写入的 T16N\_PREMAT、T16N\_MAT0~3 和 T16N\_TOP 将实时生效，改变 PWM 波形。

使用 PWM 功能时，需要将对应的 IO 端口复用功能设置为 T16N 端口，并且将该 IO 端口的方向控制寄存器设置为输出，作为 T16N0OUT0/T16N0OUT1 输出端口。

PWM 启动时的 IO 端口默认电平由寄存器 T16N\_CON2 的输出极性选择位 POL0 和 POL1 控制（无论 PWM 输出端口使能位 MOEx 的值为 0 或 1），正极性时（POLx=0）IO 端口默认电平为 0，负极性时（POLx=1）IO 端口默认电平为 1。PWM 停止时，如果只关闭 T16N 使能（T16N\_CON0.EN=0），则计数器停止计数，IO 端口保持停止前的状态；如果需要 IO 端口保持确定的电平状态，则需先关闭 PWM 端口输出使能（MOEx=0），则 IO 端口的电平取决于输出极性选择位 POLx，正极性时 IO 端口电平为 0，负极性时 IO 端口电平为 1，然后再关闭 T16N 使能（T16N\_CON0.EN=0），计数器停止计数（此处需要注意 PWM 端口输出使能和 T16N 使能的关闭顺序）；如果只关闭 PWM 端口输出使能，不关闭 T16N 使能，则计数器仍继续计数，只是 IO 端口不再输出 PWM 波形，保持为由 POLx 设定的固定电平。

注 1: T16N\_PREMAT, T16N\_MAT0~3 和 T16N\_TOP 寄存器支持缓冲机制，仅在 PWM 调制模式下，可根据应用需求设置寄存器位 T16N\_CON2.MATBUFUP 使能缓存，在其他模式下需保持 T16N\_CON2.MATBUFUP=0，不使能缓存。

注 2: 在 PWM 调制模式下使能缓存后，当前周期写入的寄存器值在计数器计到 T16N\_TOP 后才加载到缓冲寄存器并生效。当改写多个（≥2）具备缓冲机制的寄存器时，可以在计数溢出中断中进行改写，从而确保改写操作在当前周期内全部完成，下一周期全部同时生效；如果不在计数溢出中断中改写，则需将 T16N\_CNT 设置为匹配 T16N\_MATx 后继续计数，并且该方式因无法确保所有寄存器的写操作在当前周期内全部完成，下一周期的 PWM 波形可能会因只更新了一部分缓冲寄存器而出现异常，直到再过一个周期更新完毕所有缓冲寄存器后才恢复正常。当只改写 1 个具备缓冲机制的寄存器时，则无需考虑上述注意事项。

#### 举例说明：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；

T16N\_MAT0 = 0x0002; T16N\_MAT1 = 0x0004;

T16N\_MAT2 = 0x0006; T16N\_MAT3 = 0x0008; T16N\_TOP = 0x000A;

T16N\_CON0 寄存器中:

MOD<1:0>=11; T16N 设置为调制输出

MAT0S<1:0>=11; T16N\_CNT 继续计数, 并产生中断

MAT1S<1:0>=11; T16N\_CNT 继续计数, 并产生中断

MAT2S<1:0>=11; T16N\_CNT 继续计数, 并产生中断

MAT3S<1:0>=11; T16N\_CNT 继续计数, 并产生中断

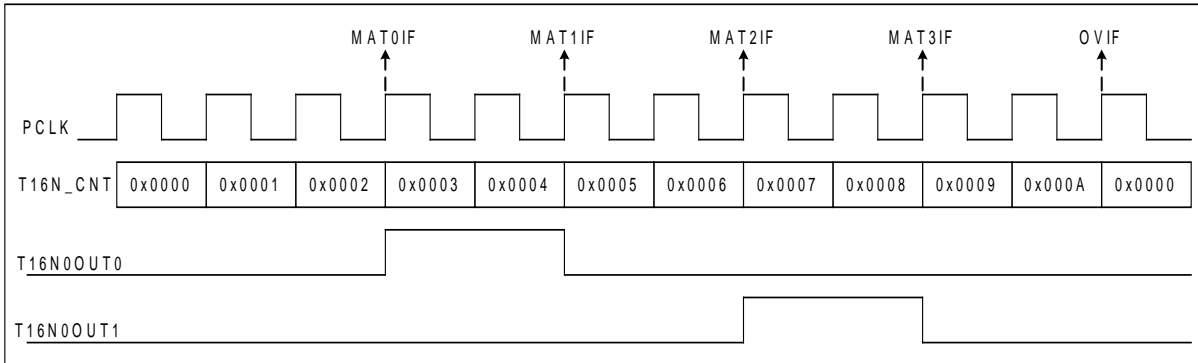


图 5-4 T16N 输出调制功能示意图

调制工作模式支持 PWM 通道输出刹车, 可通过 T16N\_CON2 寄存器的控制位进行刹车功能配置, 设置 PBKEN 可选择是否使能刹车功能, 设置 PBKPS 可选择刹车信号的极性, 设置 PBKS 可选择刹车信号输入源, 在 PINT0~PINT11 外部端口中断源中选择其中一个端口作为刹车信号输入端口, 设置 PBKL0/1 可选择刹车事件发生后 T16N0OUT0/1 端口输出的电平。外部端口中断源 PINT0~PINT11 作为刹车信号输入时, 只与被选择的端口信号电平有关, 与中断源屏蔽控制位 PMASK 和使能位 PINTIE 无关。

当外部有效刹车信号满足一定时间宽度后 (至少 11 个 PCLK 时钟周期), 即发生刹车事件, 此时 T16N\_CON2 寄存器的刹车事件标志位 PBKF 置 1, T16N\_IF 寄存器的刹车中断标志位 PBKIF 也置 1, 同时 T16N0OUT0/1 端口停止输出 PWM 信号波形, 而是输出由 PBKL0/1 所设置的电平, T16N\_CNT 保持继续计数。刹车中断标志位 PBKIF 需由软件写 1 清除, 当外部刹车信号撤销后, 刹车事件标志位 PBKF 在下一个 PWM 计数周期开始时自动清除, 并同时恢复正常的调制输出功能, T16N0OUT0/1 端口恢复输出 PWM 信号。

通过 T16N\_IE 寄存器的刹车中断使能位 PBKIE, 可对刹车中断标志 PBKIF 是否触发 T16N 中断请求 IRQ, 进行设置。

### 5.1.1.6 特殊功能寄存器

#### 5.1.1.7 T16N计数值寄存器 (T16N\_CNT)

T16N 计数值寄存器 (T16N_CNT)															
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
CNT<15:0>															
—				bit31-16				—				—			
CNT<15:0>				bit 15-0				R/W				T16N_CNT计数值			

#### 5.1.1.8 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			

#### 5.1.1.9 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>																
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
1	保留															
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
5	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
--	--	--	---

5.1.1.10 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>		SYNC	CS<1:0>		EN

—	bit 31~17	—	—
ASYWEN	bit 16	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 T16N_CNT 和 T16N_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 T16N_CNT 和 T16N_PRECNT
MAT3S<1:0>	bit 15~14	R/W	<b>T16N_CNT 匹配 T16N_MAT3 后的工作模式选择位</b> 00: T16N_CNT 继续计数，不产生匹配中断 01: T16N_CNT 保持，产生匹配中断 10: T16N_CNT 清 0 并重新计数，产生匹配中断 11: T16N_CNT 继续计数，产生匹配中断
MAT2S<1:0>	bit 13~12	R/W	<b>T16N_CNT 匹配 T16N_MAT2 后的工作模式选择位</b> 00: T16N_CNT 继续计数，不产生匹配中断 01: T16N_CNT 保持，产生匹配中断 10: T16N_CNT 清 0 并重新计数，产生匹配中断 11: T16N_CNT 继续计数，产生匹配中断
MAT1S<1:0>	bit 11~10	R/W	<b>T16N_CNT 匹配 T16N_MAT1 后的工作模式选择位</b> 00: T16N_CNT 继续计数，不产生匹配中断 01: T16N_CNT 保持，产生匹配中断 10: T16N_CNT 清 0 并重新计数，产生匹配中断 11: T16N_CNT 继续计数，产生匹配中断
MAT0S<1:0>	bit 9~8	R/W	<b>T16N_CNT 匹配 T16N_MAT0 后的工作模式选择位</b> 00: T16N_CNT 继续计数，不产生匹配中断 01: T16N_CNT 保持，产生匹配中断 10: T16N_CNT 清 0 并重新计数，产生匹配中断 11: T16N_CNT 继续计数，产生匹配中断

MOD<1:0>	bit 7~6	R/W	<b>工作模式选择位</b> 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>	bit 5~4	R/W	<b>外部时钟计数边沿选择位</b> 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	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: 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。

### 5.1.1.11 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>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留														
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	保留			CAP L1	CAP L0	CAPT<3:0>			CAPI S1	CAPI S0	CAP NE	CAP PE			
—		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									

CAPL0	bit8	R/W	捕捉0重载计数器使能位 0: 禁止 1: 使能
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 T16N0IN1 使能位 0: 禁止 1: 使能
CAPIS0	bit2	R/W	捕捉输入端口 T16N0IN0 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

### 5.1.1.12 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						MATBUF UP	保留		PBKL1	PBKL0	PBKS<3:0>				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PB KF	PB KP S	PB KE N	MOM3<1: 0>		MOM2<1:0>		MOM1<1:0>		MOM0<1:0>		POL1	POL0	MOE1	MOE0

—	bit 31-25	—	—
MATBUFUP	bit24	R/W	PWM 模式时 MAT/PREMAT/TOP 缓冲更新方式选择位 0: 即时更新 1: 当前 PWM 周期结束后才更新
—	bit 23-22	—	—
PBKL1	bit21	R/W	PWM 通道 1 刹车输出电平选择位 0: 输出低电平 1: 输出高电平

PBKLO	bit20	R/W	<b>PWM 通道 0 刹车输出电平选择位</b> 0: 输出低电平 1: 输出高电平
PBKS<3:0>	bit19-16	R/W	<b>PWM 刹车信号源选择位</b> 0000: PINT0 中断源端口 0001: PINT1 中断源端口 0010: PINT2 中断源端口 0011: PINT3 中断源端口 0100: PINT4 中断源端口 0101: PINT5 中断源端口 0110: PINT6 中断源端口 0111: PINT7 中断源端口 1000: PINT8 中断源端口 1001: PINT9 中断源端口 1010: PINT10 中断源端口 1011: PINT11 中断源端口 其他: 保留 (不建议设置其他)
—	bit15	—	—
PBKF	bit14	R	<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; 当外部刹车信号撤销后, 在下一个 PWM 计数周期开始时自动清除, 并同时恢复正常的调制输出功能。
PBKPS	bit13	R/W	<b>PWM 刹车信号极性选择位</b> 0: 高电平刹车 1: 低电平刹车
PBKEN	bit12	R/W	<b>PWM 刹车使能位</b> 0: 禁止 1: 使能
MOM3<1:0>	bit11-10	R/W	<b>T16N_MAT3 匹配后的输出端口 1 工作模式选择位</b> 00: 端口保持 01: 端口清 0 10: 端口置 1 11: 端口取反
MOM2<1:0>	bit9-8	R/W	<b>T16N_MAT2 匹配后的输出端口 1 工作模式选择位</b> 00: 端口保持 01: 端口清 0 10: 端口置 1 11: 端口取反
MOM1<1:0>	bit7-6	R/W	<b>T16N_MAT1 匹配后的输出端口 0 工作模式选择位</b> 00: 端口保持 01: 端口清 0 10: 端口置 1 11: 端口取反

MOM0<1:0>	bit5-4	R/W	<b>T16N_MAT0 匹配后的输出端口 0 工作模式选择位</b> 00: 端口保持 01: 端口清 0 10: 端口置 1 11: 端口取反
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: T16N\_PREMAT, T16N\_MAT0~3 和 T16N\_TOP 寄存器支持缓冲机制, 仅在 PWM 调制模式下, 可根据应用需求设置寄存器位 T16N\_CON2.MATBUFUP 使能缓存, 在其他模式下需保持 T16N\_CON2.MATBUFUP=0, 不使能缓存。

注 2: 在 PWM 调制模式下使能缓存后, 当前周期写入的寄存器值在计数器计到 T16N\_TOP 后才加载到缓冲寄存器并生效。当改写多个 (≥2) 具备缓冲机制的寄存器时, 可以在计数溢出中断中进行改写, 从而确保改写操作在当前周期内全部完成, 下一周期全部同时生效; 如果不在计数溢出中断中改写, 则需将 T16N\_CNT 设置为匹配 T16N\_MATx 后继续计数, 并且该方式因无法确保所有寄存器的写操作在当前周期内全部完成, 下一周期的 PWM 波形可能会因只更新了一部分缓冲寄存器而出现异常, 直到再过一个周期更新完毕所有缓冲寄存器后才恢复正常。当只改写 1 个具备缓冲机制的寄存器时, 则无需考虑上述注意事项。

### 5.1.1.13 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>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留														
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	保留							PBKI	CAP1I	CAP0I	OVIE	MAT3I	MAT2I	MAT1I	MAT0I
							E	E	E		E	E	E	E	

—	bit31-8	—	—
PBKIE	bit7	R/W	<b>PWM 刹车中断使能位</b> 0: 禁止



			1: 使能
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
OVIE	bit4	R/W	T16N_CNT 计数溢出中断使能位 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

#### 5.1.1.14 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>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留														
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	保留							PBKIF	CAP1I	CAP0I	OVIF	MAT3I	MAT2I	MAT1I	MAT0I
								F	F		F	F	F	F	

—	bit31-8	—	—
PBKIF	bit7	R/W	PWM 刹车中断标志位 0: 无刹车中断 1: 刹车事件产生
CAP1IF	bit6	R/W	输入端口 1 捕捉中断标志位 0: 输入端口 1 捕捉未成功 1: 输入端口 1 捕捉成功
CAP0IF	bit5	R/W	输入端口 0 捕捉中断标志位

			0: 输入端口 0 捕捉未成功 1: 输入端口 0 捕捉成功
OVIF	bit4	R/W	<b>T16N_CNT 计数溢出中断标志位</b> 0: 未溢出 1: 溢出
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
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
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
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

注 1: 在定时/计数, 捕捉, 调制模式下, 均可对计数器 T16N\_CNT 值与匹配寄存器 T16N\_MAT0/1/2/3 值比较是否相等。

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 PBKEN 置 1, T16N\_IF 寄存器的 PBKIF 在刹车事件产生时才被置 1。

### 5.1.1.15 T16N 触发寄存器 (T16N\_TRG)

T16N 触发寄存器 (T16N_TRG)																
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											OVT AR	MAT3TAR	MAT2TA R	MAT1TAR	MAT0T AR	

—	bit31-5	—	—
OVTAR	bit4	R/W	<b>T16N_CNT 计数溢出触发 ADC 使能位</b> 0: 禁止 1: 使能
MAT3TAR	bit3	R/W	<b>匹配 3 触发 ADC 使能位</b> 0: 禁止 1: 使能
MAT2TAR	bit2	R/W	<b>匹配 2 触发 ADC 使能位</b>

			0: 禁止 1: 使能
MAT1TAR	bit1	R/W	匹配 1 触发 ADC 使能位 0: 禁止 1: 使能
MAT0TAR	bit0	R/W	匹配 0 触发 ADC 使能位 0: 禁止 1: 使能

注: 仅在 ADC 硬件采样模式下, 支持 T16N 计数匹配中断标志 MAT0IF~MAT3IF 和计数溢出中断标志 OVIF 可触发 ADC 转换。

### 5.1.1.16 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

### 5.1.1.17 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

### 5.1.1.18 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

### 5.1.1.19 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

### 5.1.1.20 T16N计数峰值寄存器 (T16N\_TOP)

T16N_CNT 计数峰值寄存器 (T16N_TOP)															
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
TOP<15:0>															

—	bit31-16	—	—
TOP<15:0>	bit15-0	R/W	T16N_CNT计数峰值寄存器

### 5.1.1.21 T16N应用说明

芯片支持 6 个 16 位定时器/计数器，分别为 T16N0、T16N1、T16N2、T16N3、T16N4 和 T16N5。

## 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
- ◆ 支持定时/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 0，或继续计数
    - 支持 T32N0OUT0/T32N0OUT1 端口四种操作：保持，清 0，置 1，或取反
- ◆ 支持输入捕捉功能
  - ◇ 对 T32N0IN0/T32N0IN1 端口输入信号进行捕捉
  - ◇ 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
  - ◇ 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，T32N0OUT0/T32N0OUT1 为输出端口
- ◆ 支持计数中触发功能
  - ◇ 在 ADC 使能且硬件控制采样模式下，T32N 计数匹配中断标志 MAT0IF~MAT3IF 和计数溢出中断标志 OVIF 可触发 ADC 转换，可通过寄存器 T32N\_TRG 的匹配触发使能位 MAT0TAR~MAT3TAR 和溢出触发使能位 OVTAR 设置使能

### 5.1.2.2 结构框图

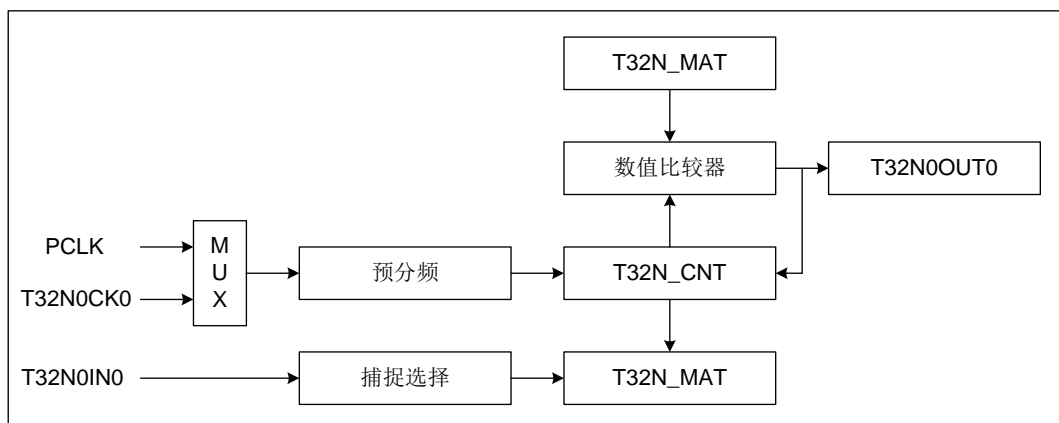


图 5-5 T32N 电路结构框图

### 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，使 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和是否产生匹配中断。

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，不产生匹配中断，当计数到 0xFFFFFFFF 后，下一次累加计数溢出，T32N\_CNT 的值为 0x00000000，并产生计数溢出中断 OVIF，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，只产生匹配中断 MAT0IF~MAT3IF。

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 0，并产生匹配中断 MAT0IF~MAT3IF，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，产生匹配中断 MAT0IF~MAT3IF，当计数到 0xFFFFFFFF 后，下一次累加计数溢出，T32N\_CNT 的值为 0x00000000，并产生计数溢出中断 OVIF，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，会产生相应的匹配中断。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，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。

通过 T32N\_IE 寄存器的匹配中断使能位 MAT0IE~MAT3IE 和溢出中断使能位 OVIE，可对匹配中断标志 MAT0IF~MAT3IF 和溢出中断标志 OVIF 是否触发 T32N 中断请求 IRQ，进行设置。

使用外部计数功能时，需要将对应的 IO 端口复用功能设置为 T32N 端口，并且将该 IO

端口的方向控制寄存器设置为输入，作为外部时钟 T32N0CK0/T32N0CK1 输入端口。

**举例说明：T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2 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 0，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：

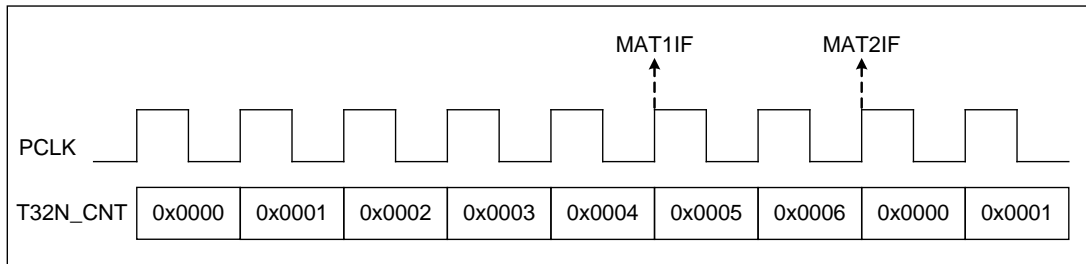


图 5-6 T32N 计数匹配功能示意图

**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，也无需产生中断。

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\_IF 的 CAP0IF 中断，通过使能 T32N\_CON1 寄存器的 CAPL0，可在产生 CAP0IF 中断时，同时将 T32N\_CNT 和 T32N\_PRECNT 清零，计数器重新开始计数；若 CAPL0 为零，则计数器继续累加。

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\_IF 的 CAP1IF 中断，通过使能 T32N\_CON1 寄存器的 CAPL1，可在产生 CAP1IF 中断时，同时将 T32N\_CNT 和 T32N\_PRECNT 清零，计数器重新开始计数；若 CAPL1 为零，则计数器继续累加。

通过 T32N\_IE 寄存器的捕捉中断使能位 CAP0IE~CAP1IE，可对捕捉中断标志 CAP0IF~CAP1IF 是否触发 T32N 中断请求 IRQ，进行设置。

当 T32N\_CNT 计数到 0xFFFFFFFF，并在下一个计数时钟（经过预分频之后的时钟）到来时，仍未检测到设定的捕捉事件，T32N\_CNT 的值因计数溢出被清零，产生溢出中断 OVIF，并重新开始累加计数。

设置 T32N\_CON1 寄存器的 CAPPE, CAPNE 和 CAPT, 可选择 T32N0IN0 和 T32N0IN1 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;

选择捕捉次数: 设置 CAPT, 可选择捕捉边沿发生的次数, 达到所设定的次数后, 产生捕捉装载动作。

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。

**举例说明: 当捕捉 T32N0IN0 端口上升沿/下降沿, 捕捉 8 次; 预分频设置为 1:1。**

T32N\_CON0 寄存器的 MOD<1:0>=10, CS<1:0>=00, MAT0S~MAT3S <1:0>=00,

T32N\_CON1 寄存器的 CAPPE=1; CAPNE=1, CAPIS=1, CAPT<3:0>=0111。

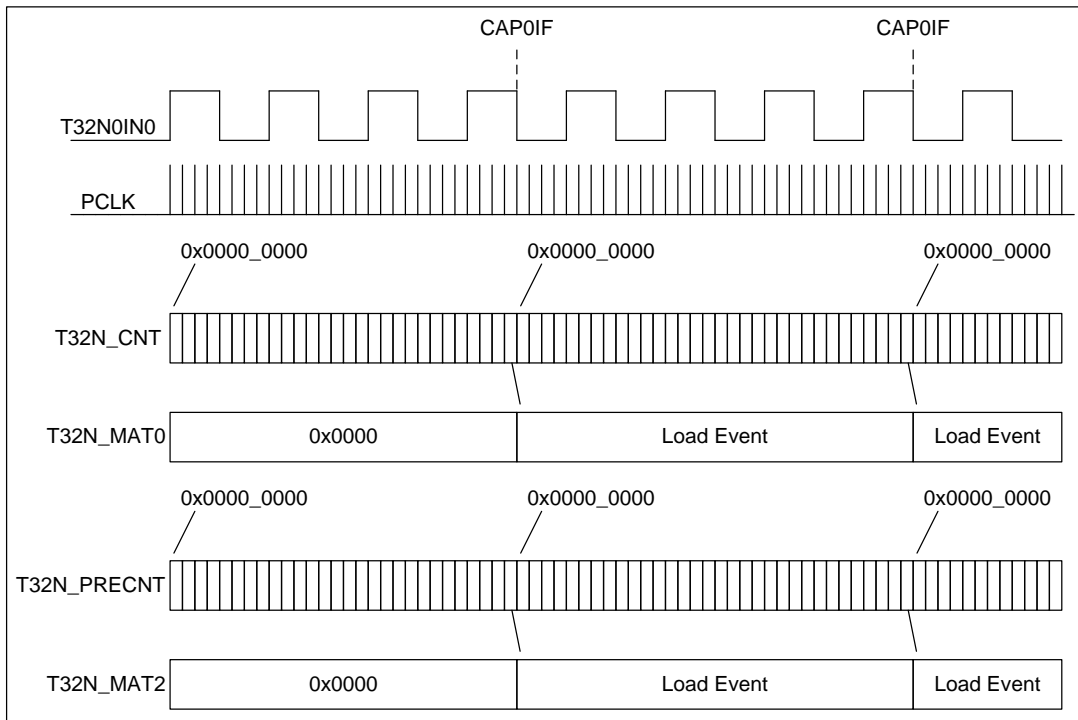


图 5-7 T32N 捕捉功能示意图

使用捕捉功能时, 需要将对应的 IO 端口复用功能设置为 T32N 端口, 并且将该 IO 端口的方向控制寄存器设置为输入, 作为 T32N0IN0/T32N0IN1 捕捉输入端口。

在捕捉工作过程中, 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, 因此首次捕捉可能会从一个非零预分频器计数开始, 导致首次捕捉的 T32N\_PRECNT 和 T32N\_CNT 值不精确。

当捕捉事件发生时, 产生的中断标志位必须通过软件清除, 并及时读取捕捉到



T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 寄存器的值，在下次捕捉事件发生时，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

T32N\_CON0 寄存器中设置 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 寄存器中 CS<1:0>=00，使 T32N\_CNT 采用内部 PCLK 时钟源计数。

设置 T32N\_CON2 寄存器的 MOE0，选择 T32N0OUT0 是否使能为匹配输出端口，使能时启用 T32N\_MAT0/T32N\_MAT1 匹配寄存器和 T32N\_CNT 进行匹配；设置 MOE1，选择 T32N0OUT1 是否使能为匹配输出端口，使能时启用 T32N\_MAT2/T32N\_MAT3 匹配寄存器和 T32N\_CNT 进行匹配。

设置 T32N\_CON2 寄存器的 MOM0/MOM1/MOM2/MOM3，选择计数匹配发生时，对 T32N0OUT0/1 端口的影响：保持，清 0，置 1，取反。

设置 T32N\_CON2 寄存器的 POL0/1，选择 T32N0OUT0/1 端口输出的极性。

使用 PWM 功能时，需要将对应的 IO 端口复用功能设置为 T32N 端口，并且将该 IO 端口的方向控制寄存器设置为输出，作为 T32N0OUT0/T32N0OUT1 输出端口。

PWM 启动时的 IO 端口默认电平由寄存器 T32N\_CON2 的输出极性选择位 POL0 和 POL1 控制（无论 PWM 输出端口使能位 MOE<sub>x</sub> 的值为 0 或 1），正极性时（POL<sub>x</sub>=0）IO 端口默认电平为 0，负极性时（POL<sub>x</sub>=1）IO 端口默认电平为 1。PWM 停止时，如果只关闭 T32N 使能（T32N\_CON0.EN=0），则计数器停止计数，IO 端口保持停止前的状态；如果需要 IO 端口保持确定的电平状态，则需先关闭 PWM 端口输出使能（MOE<sub>x</sub>=0），则 IO 端口的电平取决于输出极性选择位 POL<sub>x</sub>，正极性时 IO 端口电平为 0，负极性时 IO 端口电平为 1，然后再关闭 T32N 使能（T32N\_CON0.EN=0），计数器停止计数（此处需要注意 PWM 端口输出使能和 T32N 使能的关闭顺序）；如果只关闭 PWM 端口输出使能，不关闭 T32N 使能，则计数器仍继续计数，只是 IO 端口不再输出 PWM 波形，保持为由 POL<sub>x</sub> 设定的固定电平。

#### 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 0, 并产生中断

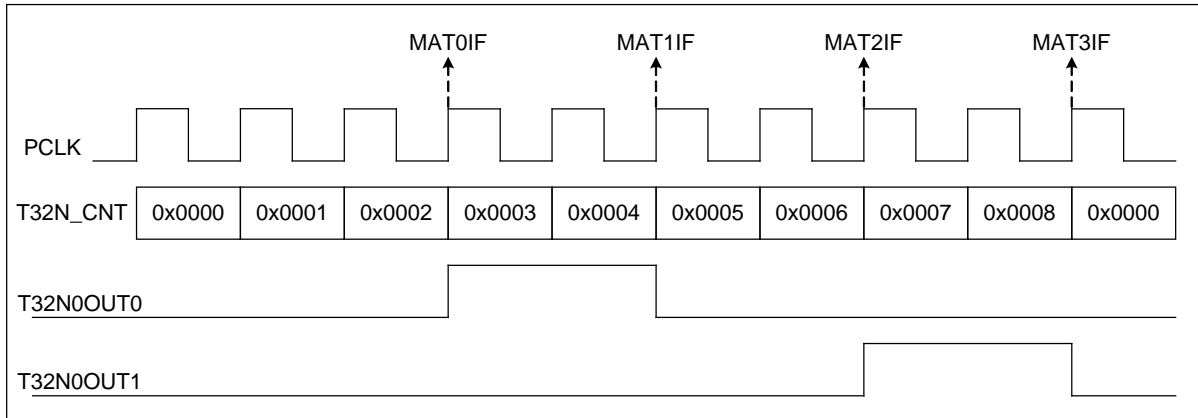


图 5-8 T32N 输出调制功能示意图

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: 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
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: 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		T32N预分频器计数值							

5.1.2.9 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: 0C <sub>H</sub>																
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>																
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
1	保留															
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
5	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
--	--	--	--------------------------------

### 5.1.2.10 T32N控制寄存器 0 (T32N\_CON0)

T32N 控制寄存器 0 (T32N_CON0)															
偏移地址: 10 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留													ASYNC WREN	
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	MAT3S<1:0>	MAT2S<1:0>	MAT1S<1:0>	MAT0S<1:0>	MOD<1:0>	EDGE<1:0>	SYN C	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, 不产生匹配中断 01: T32N_CNT 保持, 产生匹配中断 10: T32N_CNT 清 0 并重新计数, 产生匹配中断 11: T32N_CNT 继续计数, 产生匹配中断
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, 不产生匹配中断 01: T32N_CNT 保持, 产生匹配中断 10: T32N_CNT 清 0 并重新计数, 产生匹配中断 11: T32N_CNT 继续计数, 产生匹配中断
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, 不产生匹配中断 01: T32N_CNT 保持, 产生匹配中断 10: T32N_CNT 清 0 并重新计数, 产生匹配中断 11: T32N_CNT 继续计数, 产生匹配中断
MAT0S<1:0>	bit9-8	R/W	<b>T32N 匹配 T32N_MAT0 后的工作模式选择位</b> 00: T32N_CNT 继续计数, 不产生匹配中断 01: T32N_CNT 保持, 产生匹配中断 10: T32N_CNT 清 0 并重新计数, 产生匹配中断 11: T32N_CNT 继续计数, 产生匹配中断
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	<b>外部时钟计数边沿选择位</b> 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/低电平均至少保持 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

### 5.1.2.11 T32N控制寄存器 1 (T32N\_CON1)

T32N 控制寄存器 1 (T32N_CON1)															
偏移地址: 14 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留														
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	保留					CAP L1	CAP L0	CAPT<3:0>				CAPI S1	CAPI S0	CAP NE	CAP PE

—	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
CAPL0	bit8	R/W	<b>捕捉 0 重载计数器使能位</b> 0: 禁止 1: 使能

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

### 5.1.2.12 T32N控制寄存器 2 (T32N\_CON2)

T32N 控制寄存器 2 (T32N_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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		MOM3<1:0> 0>		MOM2<1:0>		MOM1<1:0>		MOM0<1:0>		POL1	POL0	MOE1	MOE0		

—	bit 31-12	—	—
MOM3<1:0>	bit11-10	R/W	<b>T32N_MAT3</b> 匹配后的输出端口 1 工作模式选择位 00: 端口保持 01: 端口清 0 10: 端口置 1 11: 端口取反
MOM2<1:0>	bit9-8	R/W	<b>T32N_MAT2</b> 匹配后的输出端口 1 工作模式选择位 00: 端口保持 01: 端口清 0 10: 端口置 1 11: 端口取反
MOM1<1:0>	bit7-6	R/W	<b>T32N_MAT1</b> 匹配后的输出端口 0 工作模式选择位 00: 端口保持 01: 端口清 0

			10: 端口置 1 11: 端口取反
MOM0<1:0>	bit5-4	R/W	<b>T32N_MAT0</b> 匹配后的输出端口 0 工作模式选择位 00: 端口保持 01: 端口清 0 10: 端口置 1 11: 端口取反
POL1	bit3	R/W	<b>T32N0OUT1</b> 输出极性选择位 0: 正极性 1: 反极性
POL0	bit2	R/W	<b>T32N0OUT0</b> 输出极性选择位 0: 正极性 1: 反极性
MOE1	bit1	R/W	输出端口 1 使能位 0: 禁止 1: 使能
MOE0	bit0	R/W	输出端口 0 使能位 0: 禁止 1: 使能

### 5.1.2.13 T32N中断使能寄存器 (T32N\_IE)

T32N 中断使能寄存器 (T32N_IE)																
偏移地址: 20 <sub>H</sub>																
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>																
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
1	保留															
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
5	保留															
										CAP	CAP	OVIE	MAT	MAT	MAT	MAT
										1IE	0IE		3IE	2IE	1IE	0IE

—	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
OVIE	bit4	R/W	计数溢出中断使能位 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

### 5.1.2.14 T32N中断标志寄存器 (T32N\_IF)

T32N 中断标志寄存器 (T32N_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									CAP 1IF	CAP 0IF	OVIF	MAT 3IF	MAT 2IF	MAT 1IF	MAT 0IF

—	bit31-7	—	—
CAP1IF	bit6	R/W	输入端口 1 捕捉中断标志位 0: 输入端口 1 捕捉未成功 1: 输入端口 1 捕捉成功
CAP0IF	bit5	R/W	输入端口 0 捕捉中断标志位 0: 输入端口 0 捕捉未成功 1: 输入端口 0 捕捉成功
OVIF	bit4	R/W	计数溢出中断标志位 0: 计数器值未溢出 1: 计数器值溢出
MAT3IF	bit3	R/W	匹配 3 中断标志位 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
MAT2IF	bit2	R/W	匹配 2 中断标志位 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
MAT1IF	bit1	R/W	匹配 1 中断标志位 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
MAT0IF	bit0	R/W	匹配 0 中断标志位



			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
--	--	--	---

注 1: 在定时/计数, 捕捉, 调制模式下, 均可对计数器 T32N\_CNT 值与匹配寄存器 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 值比较是否相等。

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。

### 5.1.2.15 T32N 触发寄存器 (T32N\_TRG)

T32N 触发寄存器 (T32N_TRG)															
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											OVT AR	MAT3TA R	MAT2TAR	MAT1TAR	MAT0T AR

—	bit31-5	—	—
OVTAR	bit4	R/W	<b>T32N_CNT 计数溢出触发 ADC 使能位</b> 0: 禁止 1: 使能
MAT3TAR	bit3	R/W	<b>匹配 3 触发 ADC 使能位</b> 0: 禁止 1: 使能
MAT2TAR	bit2	R/W	<b>匹配 2 触发 ADC 使能位</b> 0: 禁止 1: 使能
MAT1TAR	bit1	R/W	<b>匹配 1 触发 ADC 使能位</b> 0: 禁止 1: 使能
MAT0TAR	bit0	R/W	<b>匹配 0 触发 ADC 使能位</b> 0: 禁止 1: 使能

注: 仅在 ADC 硬件采样模式下, 支持 T32N 计数匹配中断标志 MAT0IF~MAT3IF 和计数溢出中断标志 OVIF 可触发 ADC 转换。

### 5.1.2.16 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: 30 <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							

### 5.1.2.17 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: 34 <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							

### 5.1.2.18 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: 38 <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							

### 5.1.2.19 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: 3C <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. 20 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 (UART0~5)

以 UART0 为例，UART1/UART2/UART3/UART4/UART5 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 11 位整数、4 位小数分频波特率
- ◇ 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 4 级接收缓冲器
  - 支持 7/8/9 位数据格式可配，支持奇偶校验功能可配
  - 支持硬件自动奇偶校验位判断
  - 支持接收缓冲中断，字节满中断、半字满中断、字满中断
  - 支持 3 类接收错误中断，接收缓冲溢出错误、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 4 级发送缓冲器
  - 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 位结束位可配
  - 支持硬件自动产生发送奇偶校验位
  - 支持发送缓冲中断，字节空中断、半字空中断、字空中断
- ◇ 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
- ◇ 支持单线半双工异步通信模式
  - 单线模式下只使用 RXD 端口，通过 GPIO\_PAFUNC/GPIO\_PBFUNC 端口复用选择寄存器选择 RXD 功能。
  - 单线接收和发送状态由用户软件协议决定

### 5.2.2 结构框图

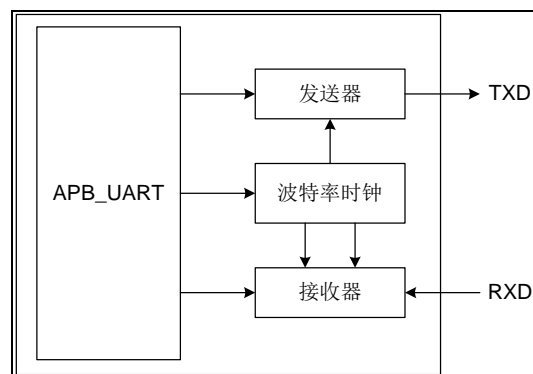


图 5-9 UART 电路结构图

### 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。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：

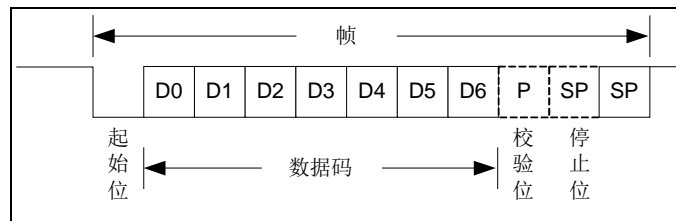


图 5-10 UART 7 位数据格式

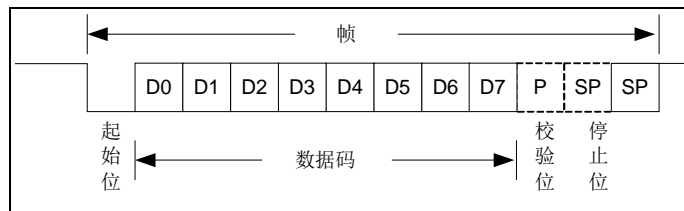


图 5-11 UART 8 位数据格式

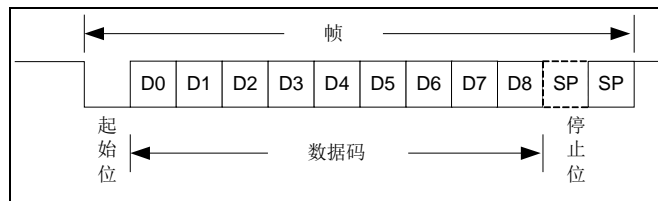


图 5-12 UART 9 位数据格式

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，复用为 TXD 端口后，固定为输出，与该 I/O 端口的方向控制寄存器无关，端口初始状态为高电平（正极性 TXP=0）或低电平（负极性 TXP=1）。

可通过 UART\_CON 寄存器的各控制位配置发送工作模式，配置 UART\_BRR 寄存器和 BCS，设定传输波特率；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，可通过寄存器 UART\_TB01/UART\_TB23 的 TP0~TP3 位读取校验位。

配置 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 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，配置 TXFS 可选择相邻两帧数据发送的时间间隔。发送缓冲器 TB0~TB3 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~TB3 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，其中字节写入时只能写低字节 UART\_TBW<7:0>，半字写入时只能写低半字 UART\_TBW<15:0>，否则会置起发送缓冲错误中断标志 TBEIF，写入的数据无效。

当发送数据为 7 位和 8 位数据格式时：字节方式写入 UART\_TBW 时，发送数据按先后顺序被依次写入到发送缓冲器 TB3~TB0，第一个数据保存在 TB3 中，如果只写一个字节数据，则该数据保存在 TB3 中，其他缓冲器为空；半字方式写入 UART\_TBW 时，发送数据被同时写入到两级发送缓冲器中，先写入的半字数据保存在 TB2 和 TB3，其中低字节存放在 TB3 中，后写入的半字数据保存在 TB0 和 TB1，其中低字节存放在 TB1 中，如果只写一个半字数据，则该半字数据保存在 TB2 和 TB3 中，其他缓冲器为空；字方式写入 UART\_TBW 时，发送数据被同时写入到发送缓冲器 TB0, TB1, TB2 和 TB3，其中低字节存放在 TB3 中。

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，半字方式写入 UART\_TBW 时，发送数据按先后顺序被依次写入到发送缓冲器 TB3~TB0，第一个数据保存在 TB3 中，如果只写一个 9 位数据，则该数据保存在 TB3 中，其他缓冲器为空；字方式写入 UART\_TBW 时，高半字被忽略，只有低半字有效，写操作等同于半字方式。

发送数据从写入到发送到端口的数据流示意图如下所示：

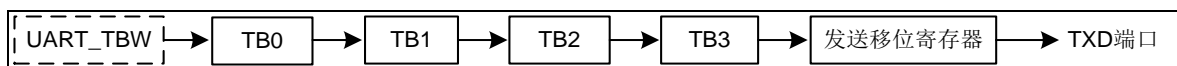


图 5-13 UART 发送数据流示意图

发送缓冲器 TB0~TB3 的数据移到下一级缓冲器或发送移位寄存器后，会置起发送空标志 TBEF0~TBEF3。当缓冲器 TB0 的发送空标志 TBEF0 为 0 时，表示 4 级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，则会置起发送缓冲错误中断标志 TBEIF，同时新写入的数据无效，缓冲器数据仍保持。

以下情况均会导致发送缓冲错误中断：当以字节方式写入 UART\_TBW 时，发送缓冲器 TB0~TB3 全满；当以半字或字方式写入 UART\_TBW 时，发送缓冲器只有一级为空或全满；当以字方式写入 UART\_TBW 时，发送缓冲器未全空；以字节方式写入非最低字节 UART\_TBW<31:8>；以半字方式写入高半字 UART\_TBW<31:16>。发生上述写错误时，会置起发送缓冲错误中断标志 TBEIF，同时新写入的数据无效，缓冲器数据仍保持。

当 4 级发送缓冲器和发送移位寄存器均空时，会置起发送空闲标志 TIDIF，表示当前的数

据发送进程结束，如果后续不再进行 UART 数据发送，可以禁止 TXEN。

每级发送缓冲器空标志 TBEF0~TBEF3 只可用于查询判断，不能产生中断请求；发送缓冲器空中断标志 TBIF 即可用于查询判断，也可用于产生中断请求，配置寄存器 UART\_IE 的 TBIM，可选择中断模式。

TBIM<1:0>=00，为字节空产生中断，有 1 级发送缓冲器（例如 TB0）为空时，会置起中断标志 TBIF；

TBIM<1:0>=01，为半字空产生中断，有 2 级发送缓冲器（例如 TB0 和 TB1）为空时，会置起中断标志 TBIF；

TBIM<1:0>=10 或 11，为字空产生中断，即发送缓冲器 TB0，TB1，TB2 和 TB3 均为空时，会置起中断标志 TBIF。

以字节空产生中断方式（TBIM<1:0>=00），发送四个 8 位数据为例，对中断标志 TBIF 和 TIDIF 的置起说明如下图：

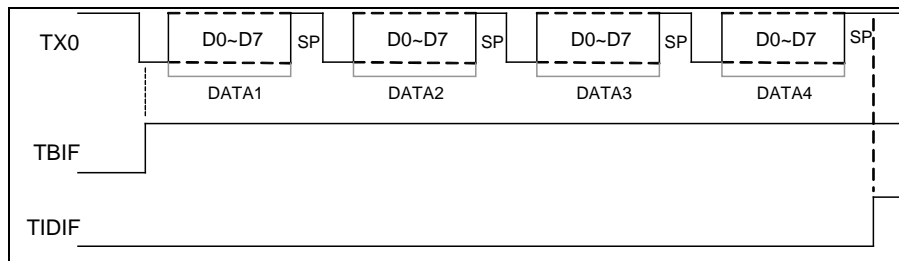


图 5-14 UART 发送中断标志示意图

通过 UART\_IE 寄存器的发送空闲中断使能位 TIDIE，发送缓冲器空中断使能位 TBIE 和发送缓冲错误中断使能位 TBEIE，可对发送空闲中断标志 TIDIF，发送缓冲器空中断标志 TBIF 和发送缓冲错误中断标志 TBEIF 是否触发 UART 中断请求 IRQ，进行设置。

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；禁止发送相关中断 TBIE=0，TBEIE=0；复位相关中断标志为默认值 TBIF=1，TBEIF=0；置起发送空闲标志 TIDIF=1；置起各发送缓冲器空标志 TBEF0~TBEF3=1。

发送数据的操作流程图示例如下：

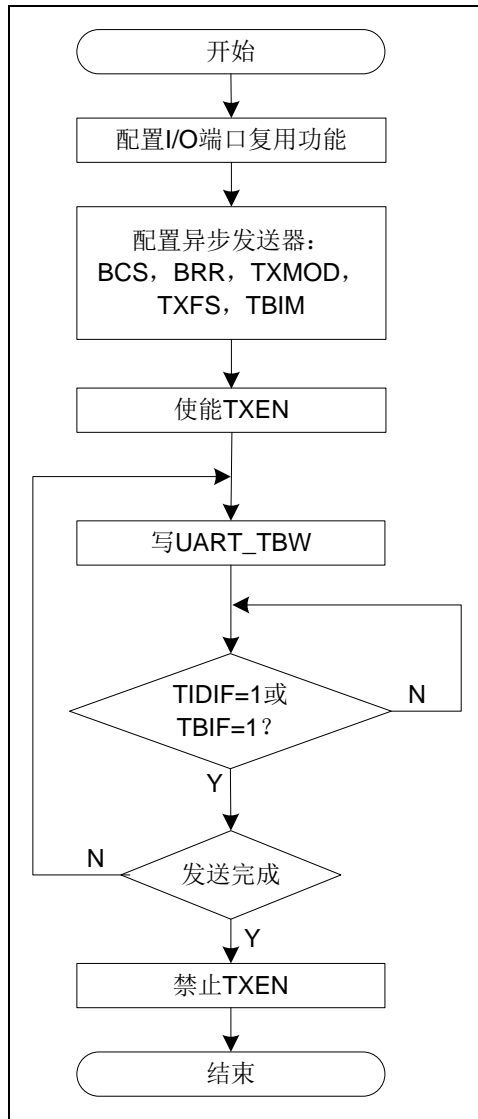


图 5-15 UART 发送数据操作流程图示例

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 TIDIF 标志的查询或中断方式也可实现数据的连续发送，但在数据帧间会有约 2 个 UART 波特率时钟周期的时间间隔。

### 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，复用为 RXD 端口后，固定为输入（单线模式除外），与该 I/O 端口的方向控制寄存器无关。

芯片内置硬件采样电路对接收端口的数据进行采样，可通过 UART\_CON 寄存器的各控制位配置接收工作模式，配置 UART\_BRR 寄存器和 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，若不正确则会置起该数据接收缓冲器数据校验错误标志位 PEx，同时也会置起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，则会置起该数据接收缓冲器数据帧错误标志位 FEx，同时也会置起帧错误中断标志 FEIF，可通过 FERSTDIS 位设置发生接收帧错误时，是否自动复位数据接收采样电路，使采样电路从复位状态重新开始运行，通常情况下保持



该位为 0，使能接收帧错误时自动复位采样电路。

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 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，接收的第一个数据保存在 RB0 中。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 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，读取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 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：字节方式读取 UART\_RBR 时，实际是读取接收缓冲器 RB0 的数据，其他缓冲器的有效数据按顺序前移；半字方式读取 UART\_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，缓冲器 RB2 和 RB3 的有效数据按顺序前移；字方式读取 UART\_RBR 时，实际是同时读取接收缓冲器 RB0, RB1, RB2 和 RB3，其中 RB0 中的数据为低字节，是本次接收的第一个数据。

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，半字方式读取 UART\_RBR 时，实际是读取接收缓冲器 RB0 的数据，其他缓冲器的有效数据按顺序前移；字方式读取 UART\_RBR 时，高半字被忽略，只有低半字有效，读操作等同于半字方式，也只是读取接收缓冲器 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：

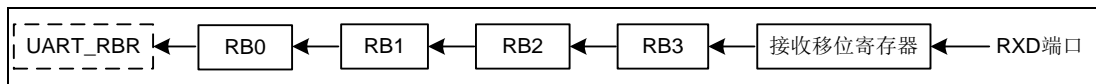


图 5-16 UART 接收数据流示意图

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，该缓冲器的数据不再有效，直到接收新的数据。

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。

当接收到起始位时，会清除接收空闲标志 RIDIF，表示正在接收数据；当接收到停止位时，会置起接收空闲标志 RIDIF，表示当前数据接收完毕。

每级接收缓冲器满标志 RBFF0~RBFF3 只可用于查询判断，不能产生中断请求；接收缓冲器满中断标志 RBIF 即可用于查询判断，也可用于产生中断请求，配置寄存器 UART\_IE 的 RBIM，可选择中断模式。

RBIM<1:0>=00，为字节满产生中断，有 1 级接收缓冲器（例如 RB0）为满时，会置起中断标志 RBIF；

RBIM<1:0>=01，为半字满产生中断，有 2 级接收缓冲器（例如 RB0 和 RB1）为满时，

会置起中断标志 RBIF;

RBIM<1:0>=10 或 11, 为字满产生中断, 即接收缓冲器 RB0, RB1, RB2 和 RB3 均为满时, 会置起中断标志 RBIF。

以字节满产生中断方式 (RBIM<1:0>=00), 接收 1 个 8 位数据为例, 对中断标志 RBIF 和 RIDIF 的置起说明如下图:

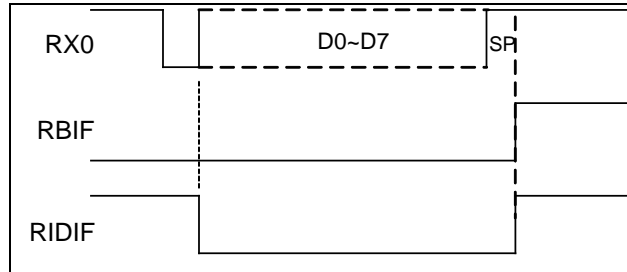


图 5-17 UART 接收中断标志示意图

通过 UART\_IE 寄存器的接收空闲中断使能位 RIDIE, 接收缓冲器满中断使能位 RBIE, 接收数据溢出中断使能位 ROIE, 接收校验错误中断使能位 PEIE 和接收帧错误中断使能位 FEIE, 可对接收空闲中断标志 RIDIF, 接收缓冲器满中断标志 RBIF, 接收数据溢出中断标志 ROIF, 接收校验错误中断标志 PEIF 和接收帧错误中断标志 FEIF 是否触发 UART 中断请求 IRQ, 进行设置。

配置 RRST, 可将异步接收器软件复位, 复位后: 禁止数据接收 RXEN=0; 禁止接收相关中断 RBIE=0, ROIE=0, FEIE=0, PEIE=0; 复位相关中断标志为默认值 RBIF=0, ROIF=0, FEIF=0, PEIF=0; 置起接收空闲标志 RIDIF=1; 清除各接收缓冲器满标志 RBFF0~RBFF3=0; 清除各接收缓冲器错误标志 FE0~FE3=0, PE0~PE3=0。

接收数据的操作流程图示例如下:

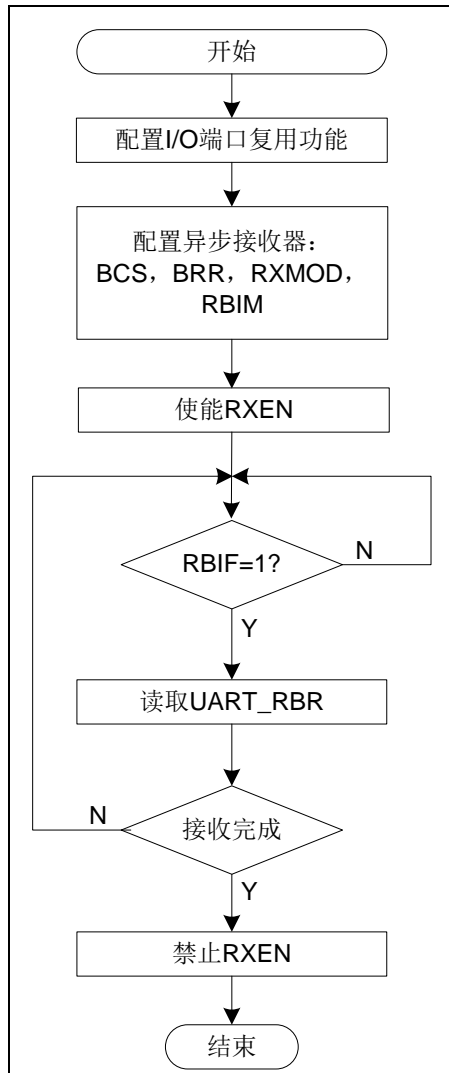


图 5-18 UART 接收数据操作流程图示例

### 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 TXD 的信号电平，由 T16N 产生的 PWM 信号源或 BUZ 信号进行调制后，再从端口输出。通过 GPIO\_TXPWM 寄存器的各个控制位可配置发送调制工作模式，通过 TXnPS 位可设置与发送端口 TXD 调制的信号源；通过 TXnPLV 位可选择发送端口 TXD 被调制的信号电平；通过 TXn\_S 位可选择调制信号输出的端口。

发送调制模式下，需使能 UART 的 TXD 端口发送数据，由 TXnPS 位所选取的调制信号源端口也需有对应的信号波形输出，才能得到正确的调制信号输出波形。

以 T16N 的 PWM 信号源，分别对 UART 发送端口 TXD 信号的高电平和低电平进行调制后，再从 TXD 端口输出的波形图如下所示：

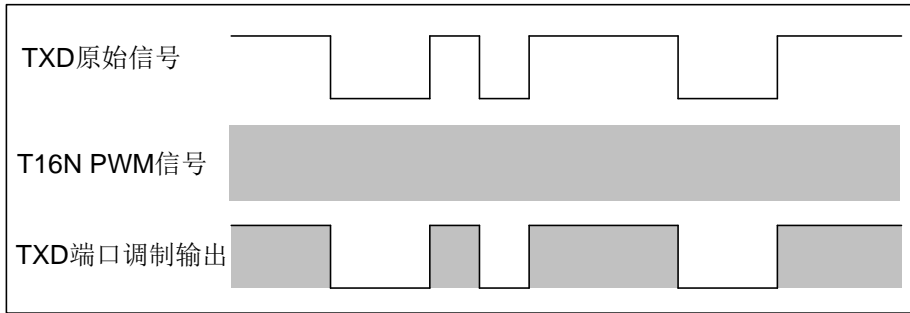


图 5-19 高电平调制输出波形图

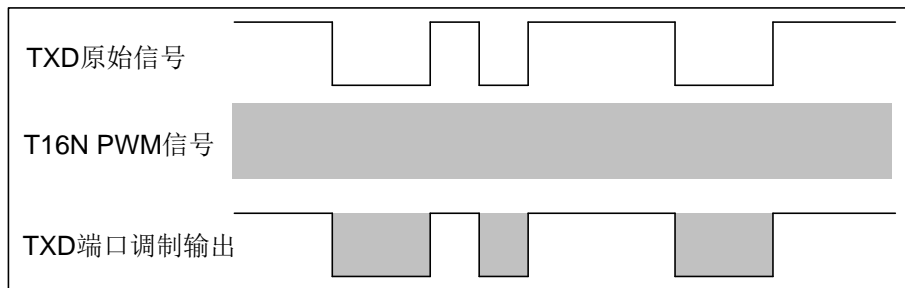


图 5-20 低电平调制输出波形图

### 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，需要通过软件控制的方式实现。使能 UART 接收复用端口 RXD 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\_CON 寄存器 TXP 和 RXP，可分别选择发送端口 TXD 和接收端口 RXD 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，起始位为低电平，停止位为高电平；选择为负极性时，UART 端口数据与传输的数据相反，起始位为高电平，停止位为低电平。

### 5.2.9 UART单线半双工通信

UART 支持通过 RXD 接口实现单线半双工模式，可配置 UART\_CON 寄存器的 SWHF=1 选择单线模式，并通过 GPIO\_PAODE/GPIO\_PBODE 寄存器配置 RXD 端口为开漏输出，可通过 GPIO\_PAPUE/GPIO\_PBPUE 寄存器使能 RXD 端口的内部弱上拉或在 RXD 端口外接上拉电阻。

单线半双工模式下只使用 RXD 端口，只需选择复用为 RXD 功能的端口，无需选择复用为 TXD 功能的 IO 端口；该模式下 RXD 端口极性必须设置为正极性，TXD 端口虽不输出数据，但仍需设置为正极性，使得通过 RXD 端口输出的数据为正极性。

单线半双工模式下 RXD 端口的输入和输出方向由硬件电路自动控制，与该端口的方向控制寄存器无关，进行数据接收时，RXD 端口为输入，数据发送时硬件自动切换为输出。

单线半双工通信过程中的接收和发送状态由用户软件实现。

## 5.2.10 特殊功能寄存器

### 5.2.10.1 UART控制寄存器 (UART\_CON)

UART 控制寄存器 (UART_CON)															
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	BCS<2:0>			保留										FERST	SWH
														DIS	F
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXP	RXMOD<2:0>			保留	RRS	RXE	TXP	TXMOD<2:	保留	TXFS	TRST	TXEN			
						T	N	0>							

—	bit 31	—	—
BCS<2:0>	bit30-28	R/W	<b>传输波特率发生器时钟选择位</b> 000: 禁止, 波特率发生器停止工作 001: PCLK 010: PCLK/2 011: PCLK/4 1xx: PCLK/8
—	bit27-18	—	—
FERSTDIS	bit 17	R/W	<b>接收帧错误后自动复位禁止位</b> 0: 使能帧错误后自动复位接收采样电路 1: 禁止帧错误后自动复位接收采样电路
SWHF	bit 16	R/W	<b>单线半双工选择位</b> 0: 未选择单线半双工模式 1: 选择单线半双工模式
RXP	bit15	R/W	<b>接收端口极性选择位</b> 0: 正极性 (标准 UART 极性) 1: 负极性 (反向 UART 极性)
RXMOD<2:0>	bit14-12	R/W	<b>接收数据格式选择位</b> 000: 7 位数据 001: 8 位数据 01x: 9 位数据 100: 7 位数据+奇校验位 101: 7 位数据+偶校验位 110: 8 位数据+奇校验位 111: 8 位数据+偶校验位
—	bit11-10	—	—
RRST	bit9	W	<b>接收器软件复位</b> 0: 读取时始终为 0 1: 软件复位
RXEN	bit8	R/W	<b>接收使能位</b>

			0: 禁止 1: 使能
TXP	bit7	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<2:0>	bit6-4	R/W	发送数据格式选择位 000: 7 位数据 001: 8 位数据 01x: 9 位数据 100: 7 位数据+奇校验位 101: 7 位数据+偶校验位 110: 8 位数据+奇校验位 111: 8 位数据+偶校验位
—	bit3	—	—
TXFS	bit2	R/W	发送帧停止位选择位 0: 1 位停止位 1: 2 位停止位
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

### 5.2.10.2 UART波特率寄存器 (UART\_BRR)

UART 波特率寄存器 (UART_BRR)															
偏移地址: 04 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留														
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	保留														
保	BRFRA<3:0>					BRR<10:0>									
留															

—	bit31-15	—	—
BRFRA<3:0>	bit14-11	R/W	传输波特率小数位
BRR<10:0>	bit10-0	R/W	传输波特率整数位

注 1: UART\_BRR 表示为 15 位无符号数, 其中 4 位小数位, 11 位整数位, 其数值表示的是 UART 波特率分频数。注意小数部分在高位。例如: 在 PCLK 为 48MHz 时, 设置 UART\_BRR 为 0x819、BCS 为 1, 则对应波特率约是 115200bps。

注 2: UART 传输波特率计算公式如下:

$$BAUD = \frac{Fpclk}{16 \times n \times (BRRDIV + 1)}$$

其中 Fpclk 为系统时钟频率, BRRDIV 为波特率分频数, 由 UART\_BRR 寄存器的 BRR<10:0>和 BRFR<3:0>决定, n 为波特率发生器时钟预分频数, 由 UART\_CON 寄存器的 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。

### 5.2.10.3 UART发送数据写 寄存器 (UART\_TBW)

UART 发送数据写入寄存器 (UART_TBW)															
偏移地址: 08 <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	<p><b>写入的发送数据</b></p> <p>字节写入时: 仅允许对 UART_TBW&lt;7:0&gt;写入</p> <p>半字写入时: 仅允许对 UART_TBW&lt;15:0&gt;写入</p> <p>字写入时: 对 UART_TBW&lt;31:0&gt;写入</p>
-----------	---------	---	--

### 5.2.10.4 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: 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><b>读取的接收数据</b></p> <p>字节读取时: 仅允许对 UART_RBR&lt;7:0&gt;读取</p> <p>半字读取时: 仅允许对 UART_RBR&lt;15:0&gt;读取</p> <p>字读取时: 对 UART_RBR&lt;31:0&gt;读取</p>
-----------	---------	---	--

### 5.2.10.5 UART发送缓冲 0/1 寄存器 (UART\_TB01)

UART 发送缓冲 0/1 寄存器 (UART_TB01)															
偏移地址: 10 <sub>H</sub>															

复位值: 00100000\_00000000\_00100000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TBEF1	TP1	保留	TB1<8:0>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TBEF0	TP0	保留	TB0<8:0>											

—	bit31-30	—	—
TBEF1	bit29	R	发送缓冲器 1 空标志位 0: 满 1: 空
TP1	bit28	R	发送的奇偶校验位 发送缓冲器 1 对应的奇偶校验位
—	bit27-25	—	—
TB1<8:0>	bit24-16	R	发送缓冲器 1 数据
—	bit15-14	—	—
TBEF0	bit13	R	发送缓冲器 0 空标志位 0: 满 1: 空
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

### 5.2.10.6 UART发送缓冲 2/3 寄存器 (UART\_TB23)

UART 发送缓冲 2/3 寄存器 (UART\_TB23)

偏移地址: 14<sub>H</sub>

复位值: 00100000\_00000000\_00100000\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TBEF3	TP3	保留	TB3<8:0>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TBEF2	TP2	保留	TB2<8:0>											

—	bit31-30	—	—
TBEF3	bit29	R	发送缓冲器 3 空标志位 0: 满 1: 空
TP3	bit28	R	发送的奇偶校验位 发送缓冲器 3 对应的奇偶校验位
—	bit27-25	—	—
TB3<8:0>	bit24-16	R	发送缓冲器 3 数据
—	bit15-14	—	—
TBEF2	bit13	R	发送缓冲器 2 空标志位



			0: 满 1: 空
TP2	bit12	R	发送的奇偶校验位 发送缓冲器 2 对应的奇偶校验位
—	bit11-9	—	—
TB2<8:0>	bit8-0	R	发送缓冲器 2 数据

### 5.2.10.7 UART接收缓冲 0/1 寄存器 (UART\_RB01)

UART 接收缓冲 0/1 寄存器 (UART_RB01)															
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
PE1	FE1	RBFF1	RP1	保留			RB1<8:0>								
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>								

PE1	bit31	R	接收缓冲器 1 数据校验错误标志位 0: 正确 1: 错误
FE1	bit30	R	接收缓冲器 1 数据帧错误标志位 0: 正确 1: 错误
RBFF1	bit29	R	接收缓冲器 1 空满标志位 0: 空 1: 满
RP1	bit28	R	接收的奇偶校验位 接收缓冲器 1 对应的奇偶校验位
—	bit27-25	—	—
RB1<8:0>	bit24-16	R	接收缓冲器 1 数据
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

### 5.2.10.8 UART接收缓冲 2/3 寄存器 (UART\_RB23)

UART 接收缓冲 2/3 寄存器 (UART_RB23)															
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
PE3	FE3	RBFF3	RP3	保留			RB3<8:0>								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PE2	FE2	RBFF2	RP2	保留			RB2<8:0>								

PE3	bit31	R	接收缓冲器 3 数据校验错误标志位 0: 正确 1: 错误
FE3	bit30	R	接收缓冲器 3 数据帧错误标志位 0: 正确 1: 错误
RBFF3	bit29	R	接收缓冲器 3 空满标志位 0: 空 1: 满
RP3	bit28	R	接收的奇偶校验位 接收缓冲器 3 对应的奇偶校验位
—	bit27-25	—	—
RB3<8:0>	bit24-16	R	接收缓冲器 3 数据
PE2	bit15	R	接收缓冲器 2 数据校验错误标志位 0: 正确 1: 错误
FE2	bit14	R	接收缓冲器 2 数据帧错误标志位 0: 正确 1: 错误
RBFF2	bit13	R	接收缓冲器 2 空满标志位 0: 空 1: 满
RP2	bit12	R	接收的奇偶校验位 接收缓冲器 2 对应的奇偶校验位
—	bit11-9	—	—
RB2<8:0>	bit8-0	R	接收缓冲器 2 数据

### 5.2.10.9 UART中断使能寄存器 (UART\_IE)

UART 中断使能寄存器 (UART_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	RIDIE	TIDIE	RBIM<1:0>	TBIM<1:0>		保留		TBEIE	PEIE	FEIE	ROIE	RBIE	TBIE		

—	bit31-14	—	—
RIDIE	bit13	R/W	接收空闲标志中断使能位 0: 禁止 1: 使能
TIDIE	bit12	R/W	发送空闲标志中断使能位 0: 禁止 1: 使能
RBIM<1:0>	bit11-10	R/W	接收缓冲器满中断模式选择位 00: 字节满产生中断 01: 半字满产生中断 1x: 字满产生中断
TBIM<1:0>	bit9-8	R/W	发送缓冲器空中断模式选择位 00: 字节空产生中断 01: 半字空产生中断 1x: 字空产生中断
—	bit7-6	—	—
TBEIE	bit5	R/W	发送缓冲错误中断使能位 0: 禁止 1: 使能
PEIE	bit4	R/W	接收校验错误中断使能位 0: 禁止 1: 使能
FEIE	bit3	R/W	接收帧错误中断使能位 0: 禁止 1: 使能
ROIE	bit2	R/W	接收数据溢出中断使能位 0: 禁止 1: 使能
RBIE	bit1	R/W	接收缓冲器满中断使能位 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

### 5.2.10.10 UART中断标志寄存器 (UART\_IF)

UART 中断标志寄存器 (UART\_IF)

偏移地址: 24<sub>H</sub>

复位值: 00000000\_00000000\_XX110000\_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		RIDIF	TIDIF	保留						TBEIF	PEIF	FEIF	ROIF	RBIF	TBIF

—	bit31-14	—	—
RIDIF	bit13	R/W	<b>接收空闲标志中断标志位</b> 0: 忙碌 1: 接收空闲 初始复位为 1, 软件写 1 清除标志位, 写 0 无效
TIDIF	bit12	R/W	<b>发送空闲标志中断标志位</b> 0: 忙碌 1: 发送空闲 初始复位为 1, 软件写 1 清除标志位, 写 0 无效
—	bit11-6	—	—
TBEIF	bit5	R/W	<b>发送缓冲错误中断标志位</b> 0: 发送缓冲未错误 1: 发送缓冲错误 软件写 1 清除标志位, 写 0 无效
PEIF	bit4	R/W	<b>接收校验错误中断标志位</b> 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	bit3	R/W	<b>接收帧错误中断标志位</b> 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	bit2	R/W	<b>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
RBIF	bit1	R	<b>接收缓冲器满中断标志位</b> 0: 非满 1: 满 (满足 RBIM 所选择的条件)
TBIF	bit0	R	<b>发送缓冲器空中断标志位</b> 0: 非空 1: 空 (满足 TBIM 所选择的条件)

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。

### 5.2.11 UART应用说明

芯片支持 6 个通用异步接收/发送器为 UART0, UART1, UART2, UART3, UART4, UART5。

UART0~3 支持发送调制功能，可通过 GPIO\_TXPWM 寄存器的相应控制位进行设置。

UART0 的发送调制功能由 TXD0 信号与 BUZ 或 T16N0 的 PWM 信号进行调制，可通过 TX0PS, TX0PLV, TX0\_S 位进行设置。

UART1 的发送调制功能由 TXD1 信号与 BUZ 或 T16N1 的 PWM 信号进行调制，可通过 TX1PS, TX1PLV, TX1\_S 位进行设置。

UART2 的发送调制功能由 TXD2 信号与 BUZ 或 T16N2 的 PWM 信号进行调制，可通过 TX2PS, TX2PLV, TX2\_S 位进行设置。

UART3 的发送调制功能由 TXD3 信号与 BUZ 或 T16N3 的 PWM 信号进行调制，可通过 TX3PS, TX3PLV, TX3\_S 位进行设置。

## 5.3 SPI同步串口通讯控制器 (SPI0)

### 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

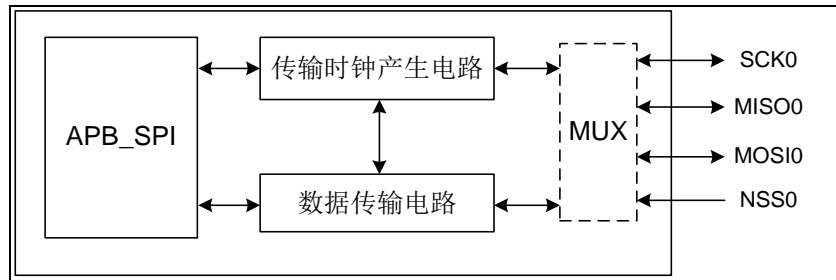


图 5-21 SPI 电路结构框图

### 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 SCK0，从动模式下的片选信号端口为 NSS0，数据输出端口 MISO0，数据输入端口 MOSI0，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 MOSI0，数据输入端口 MISO0。具体见下表所示：

SPI 通讯端口	SPI 主控模式	SPI 从动模式
SCK0	支持	支持
MOSI0	支持	支持
MISO0	支持	支持
NSS0	—	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 MOSI0(或 MISO0) 会在第一个 SCK0 时钟边沿，输出数据的 MSB 位；反之输出端口 MOSI0 (或 MISO0) 在第一个 SCK0 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）：

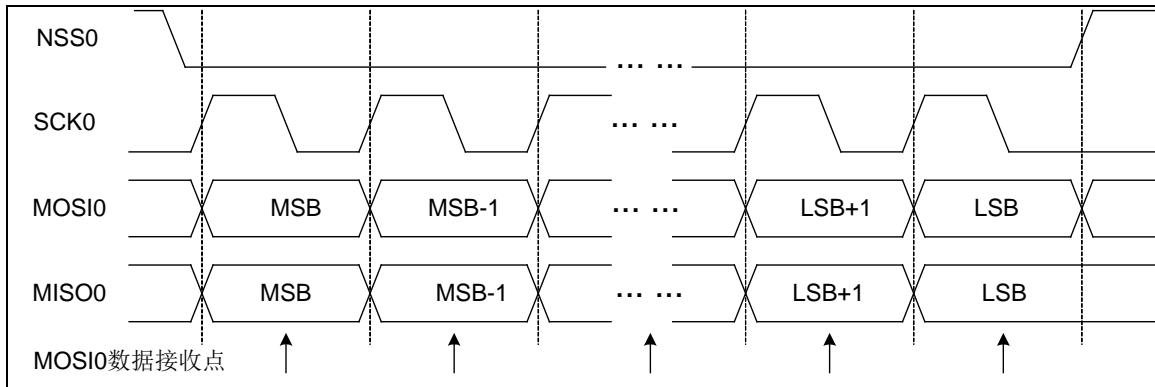


图 5-22 SPI 时钟上升沿发送，下降沿接收波形示意图

DFS<1:0> = 01，下降沿发送（先），上升沿接收（后）：

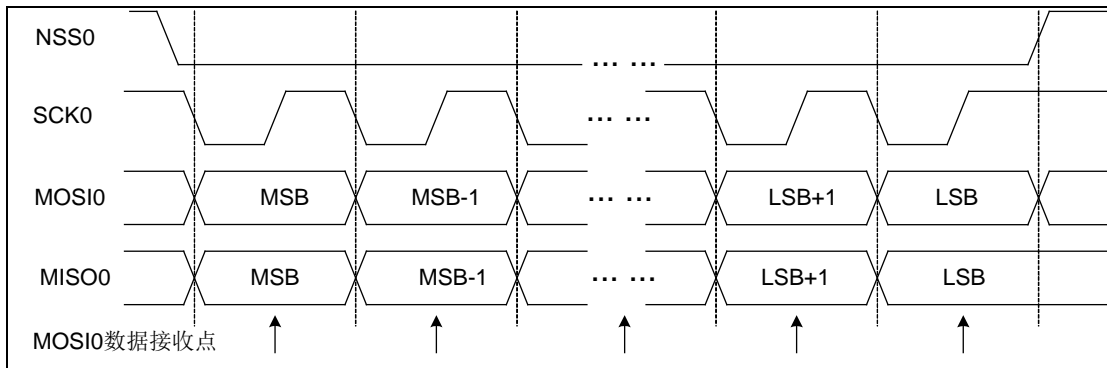


图 5-23 SPI 时钟下降沿发送，上升沿接收波形示意图

DFS<1:0> = 10，上升沿接收（先），下降沿发送（后）：

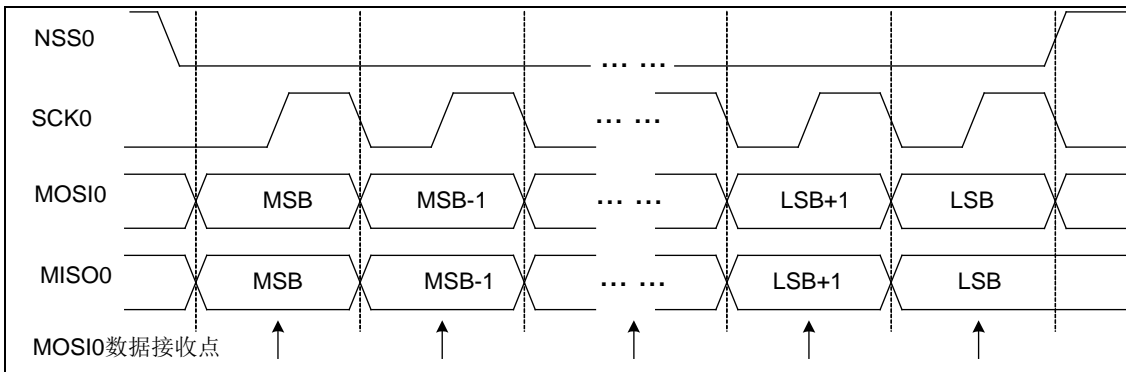


图 5-24 SPI 时钟上升沿接收，下降沿发送波形示意图

DFS<1:0> = 11, 下降沿接收 (先), 上升沿发送 (后):

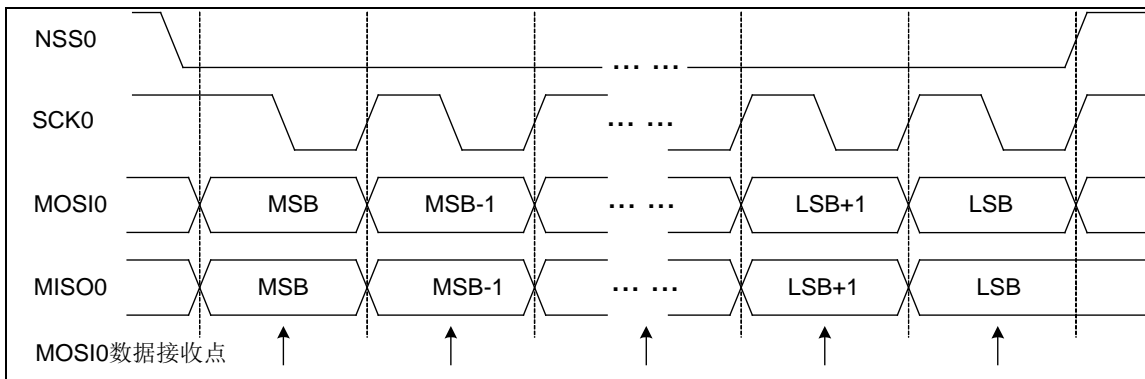


图 5-25 SPI 时钟下降沿接收, 上升沿发送波形示意图

### 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, 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 MOSI0 (或 MISO0) 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, 其中字节写入时只能写低字节 SPI\_TBW<7:0>, 半字写入时只能写低半字 SPI\_TBW<15:0>, 否则会置起写错误中断标志 TBWEIF, 写入的数据无效。

字节方式写入 SPI\_TBW 时, 发送数据按先后顺序被依次写入到发送缓冲器 TB3~TB0, 第一个数据保存在 TB3 中, 如果只写一个字节数据, 则该数据保存在 TB3 中, 其他缓冲器为空; 半字方式写入 SPI\_TBW 时, 发送数据被同时写入到两级发送缓冲器中, 先写入的半字数据保存在 TB2 和 TB3, 其中低字节存放在 TB3 中, 后写入的半字数据保存在 TB0 和 TB1, 其中低字节存放在 TB1 中, 如果只写一个半字数据, 则该半字数据保存在 TB2 和 TB3 中, 其他缓冲器为空; 字方式写入 SPI\_TBW 时, 发送数据被同时写入到发送缓冲器 TB0, TB1, TB2 和 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):



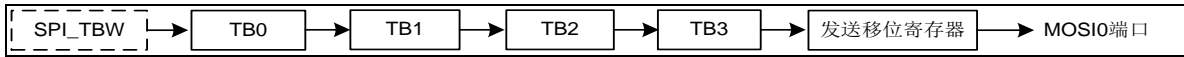


图 5-26 SPI 发送数据流示意图

发送缓冲器 TB0~TB3 的数据移到下一级缓冲器或发送移位寄存器后，会置起发送空标志 TBEF0~TBEF3。当缓冲器 TB0 的发送空标志 TBEF0 为 0 时，表示 4 级发送缓冲器和发送移位寄存器均满。

每级发送缓冲器空标志 TBEF0~TBEF3 只可用于查询判断，不能产生中断请求；发送缓冲器空中断标志 TBIF 即可用于查询判断，也可用于产生中断请求，配置 SPI\_IE 寄存器的 TBIM，可选择中断模式。

TBIM<1:0>=00，为 TB0 字节空产生中断，即有 1 级发送缓冲器为空时，SPI\_IF 寄存器会置起中断标志 TBIF；

TBIM<1:0>=01，为 TB0~TB1 半字空产生中断，即有 2 级发送缓冲器为空时，SPI\_IF 寄存器会置起中断标志 TBIF；

TBIM<1:0>=10，为 TB0~TB3 字全空产生中断，即 4 级发送缓冲器均为空时，SPI\_IF 寄存器会置起中断标志 TBIF。

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，并且数据接收也完毕后，则进入空闲状态，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。

以 SPI 主控模式，DFS<1:0> = 00，上升沿发送（先），下降沿接收（后），TB0 字节空产生中断方式（TBIM<1:0>=00），发送四个 8 位数据为例，对中断标志 TBIF 和 IDIF 的置起说明如下图：

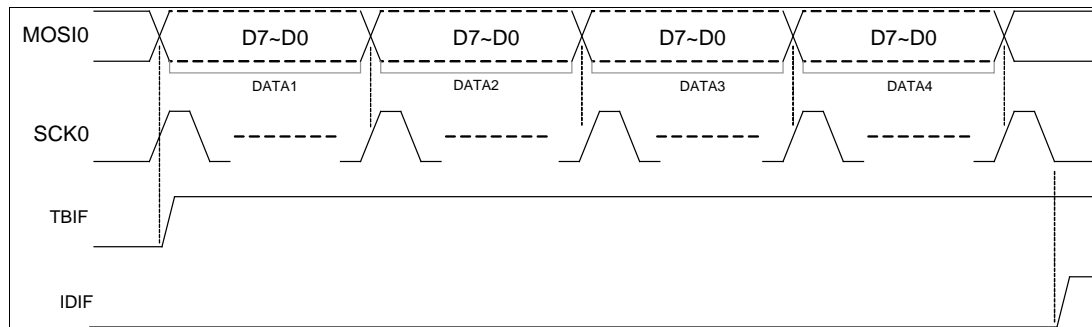


图 5-27 SPI 发送中断标志示意图

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/TB1/TB2/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。以下情况会导致写错误中断：当以字节方式写入 SPI\_TBW 时，发送缓冲器 TB0~TB3 全满；当以半字或字方式写入 SPI\_TBW 时，发送缓冲器只有一级为空或全满；当以字方式写入 SPI\_TBW 时，发送缓冲器未全空；以字节方式写入非最低字节 SPI\_TBW<31:8>；以半字方式写入高半字 SPI\_TBW<31:16>。发生上述写错误时，会置起写错误中断标志 TBWEIF，同时新写入的数据无效，缓冲器数据仍保持。

通过 SPI\_IE 寄存器的发送缓冲器空中断使能位 TBIE 和发送数据写错误中断使能位

TBWEIE，可对发送缓冲器空中断标志 TBIF 和发送数据写错误中断标志 TBWEIF 是否触发 SPI 中断请求 IRQ，进行设置。

### 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，其他缓冲器的有效数据按顺序前移；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，缓冲器 RB2 和 RB3 的有效数据按顺序前移；字方式读取 SPI\_RBR 时，实际是同时读取接收缓冲器 RB0, RB1, RB2 和 RB3，其中 RB0 中的数据为低字节，是本次接收的第一个数据。

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（以主控模式为例）：



图 5-28 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，该缓冲器的数据不再有效，直到接收新的数据。

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。

每级接收缓冲器满标志 RBFF0~RBFF3 只可用于查询判断，不能产生中断请求；接收缓冲器满中断标志 RBIF 即可用于查询判断，也可用于产生中断请求，配置 SPI\_IE 寄存器的 RBIM，可选择中断模式。

RBIM<1:0>=00，为 RB0 字节满产生中断，即有 1 级接收缓冲器为满时，SPI\_IF 寄存器中会置起中断标志 RBIF；

RBIM<1:0>=01，为 RB0~RB1 半字满产生中断，即有 2 级接收缓冲器为满时，SPI\_IF 寄

寄存器中会置起中断标志 RBIF;

RBIM<1:0>=10, 为 RB0~RB3 字全满产生中断, 即 4 级接收缓冲器均为满时, SPI\_IF 寄存器中会置起中断标志 RBIF。

以 SPI 主控模式, DFS<1:0> = 00, 上升沿发送 (先), 下降沿接收 (后), RB0 字节满产生中断方式 (RBIM<1:0>=00), 接收一个 8 位数据为例, 对中断标志 RBIF 的置起说明如下图:

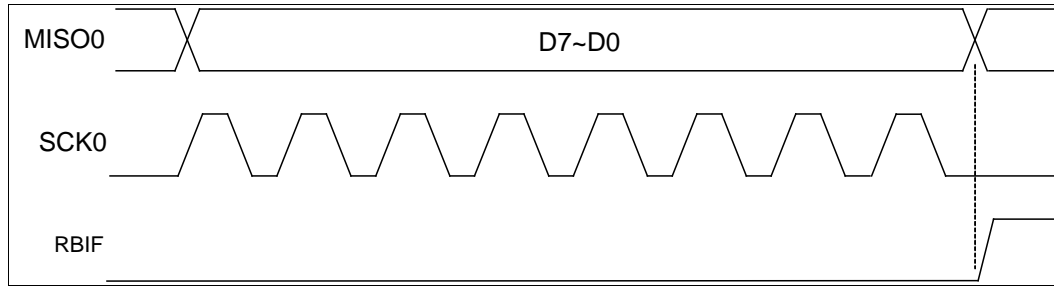


图 5-29 SPI 接收中断标志示意图

通过 SPI\_IE 寄存器的接收缓冲器满中断使能位 RBIE 和接收数据溢出中断使能位 ROIE, 可对接收缓冲器满中断标志 RBIF 和接收数据溢出中断标志 ROIF 是否触发 SPI 中断请求 IRQ, 进行设置。

### 5.3.8 SPI 通讯控制

进行 SPI 通讯时, 需通过寄存器 GPIO\_PAFUNCx/GPIO\_PBFUNCx, 将对应 I/O 端口复用功能设置为 SPI 通讯端口 MOSIO, MISO0, SCK0 和 NSS0, 由 SPI 硬件电路控制通讯端口的输入和输出, 与该 I/O 端口的方向控制寄存器无关。

通过寄存器 SPI\_CON 可进行 SPI 通讯配置, 通过 MS 位配置 SPI 通讯模式, 通过 DFS 位配置通讯数据格式, 通过 DW 位配置发送帧位宽; 对主控制模式, 还需通过 DRE 位设置是否使能接收延迟, 通过 TME 位设置是否使能帧发送间隔, 并通过 TMP 位设置帧发送间隔周期, 配置寄存器 SPI\_CKS 的 CKS 位, 设定传输时钟速率; 对从动模式, 传输时钟由主机方提供; 配置 SPI\_CON 寄存器的 EN 和 REN 位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, 注意需先使能 SPI\_CON 寄存器的 EN 和 REN 位, 再写发送数据寄存器 SPI\_TBW, 才能正确启动 SPI 数据的发送和接收。

需要注意 SPI 通讯数据格式默认为上升沿发送 (先), 下降沿接收 (后), 如果需要更改数据格式, 则需先修改完毕寄存器 SPI\_CON 的 DFS 位, 再使能 SPI 通讯使能位 EN 和接收使能位 REN, 即至少要分两次写寄存器 SPI\_CON。

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数据错误中断标志 TEIF。

SPI 从动模式, 支持片选信号变化中断, 当片选信号发生变化时, 会置起寄存器 SPI\_IF 的中断标志位 NSSIF。

通过 SPI\_IE 寄存器的空闲中断使能位 IDIE，发送数据错误中断使能位 TEIE，片选变化中断使能位 NSSIE，可对空闲中断标志 IDIF，发送数据错误中断标志 TEIF，片选变化中断标志 NSSIF 是否触发 SPI 中断请求 IRQ，进行设置。

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）。

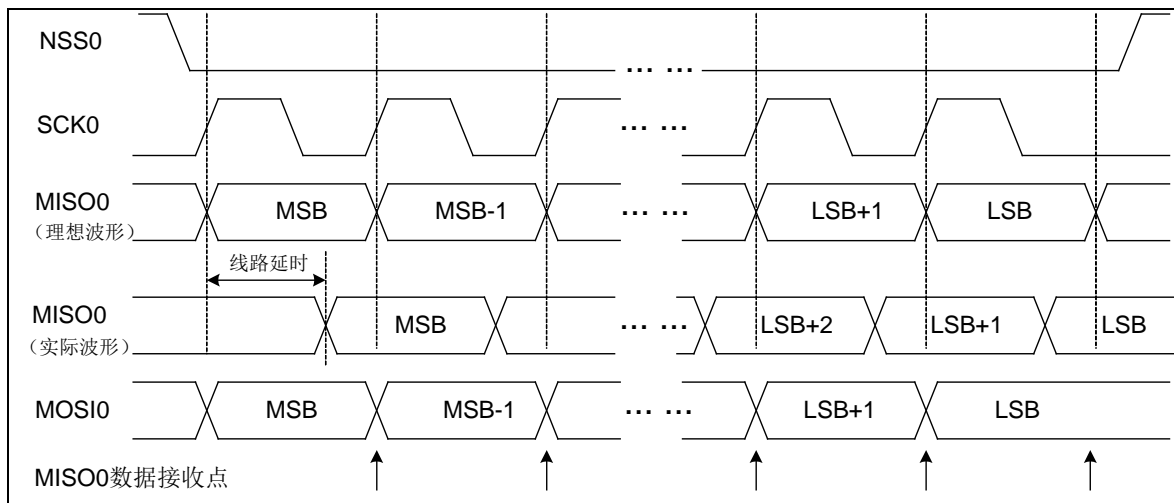


图 5-30 SPI 延迟接收功能波形示意图

### 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，可使能该功能，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		
RX	TXC	保留		DW<2:0>		TMP<5:0>					TMS	TME					
CL	LR																
R																	
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 SPIDW+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:

$T_{SCK0} * (1 + TMP)$ , 即时间间隔为 1~64 个通讯时钟周期  $T_{SCK0}$ 。

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 DFS, 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: 08H

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
-----------	---------	---	--

### 5.3.11.3 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: 0CH

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	<b>读取的接收数据</b>
-----------	---------	---	----------------

			字节读取时：仅允许对 RBR<7:0>读取 半字读取时：仅允许对 RBR<15:0>读取 字读取时：对 RBR<31:0>读取
--	--	--	---

### 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	<b>SPI 接收缓冲器满中断模式选择位</b> 00: RB0 字节满产生中断 01: RB0~RB1 半字满产生中断 10: RB0~RB3 字全满产生中断 11: 保留
TBIM<1:0>	bit9-8	R/W	<b>SPI 发送缓冲器空中断模式选择位</b> 00: TB0 字节空产生中断 01: TB0~TB1 半字空产生中断 10: TB0~TB3 字全空产生中断 11: 保留
—	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: 14H															
复位值: 00000000_00000000_00000000_00000001B															
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>



			0: 非满 1: 满 (满足 RBIM 所选择的条件) 读 SPI_RBR 可清除中断标志
TBIF	bit0	R	<b>SPI 发送缓冲器空中断标志位</b> 0: 非空 1: 空 (满足 TBIM 所选择的条件) 写 SPI_TBW 可清除中断标志

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 (仅主控模式支持)</b> 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: 24H															
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>							

CKS<7:0>	bit7-0	R/W	SPI 通讯时钟设置位 (仅主控模式支持) SPI 通讯波特率计算公式如下: CKS<7:0>=0x00 时: FPCLK; CKS<7:0>=0x01~0xFF 时: FPCLK/(CKS * 2)
—	bit31-8	—	—

注: SPI 通讯时钟特率计算公式如下, SPI\_CKS 寄存器中:

CKS<7:0>=0x00 时: FPCLK;

CKS<7:0>=0x01~0xFF 时: FPCLK/(CKS \* 2)。

### 5.3.12 SPI应用说明

为保证 SPI 的正常通信, SPI 的配置需遵循如下要求:

1. SPI 使用了 20ns 滤波器时, SPI 的通讯时钟频率需小于 10MHz。
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 >= 0;  
当 SPI 帧位宽选择 2~4 位时, CKS >= 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, 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

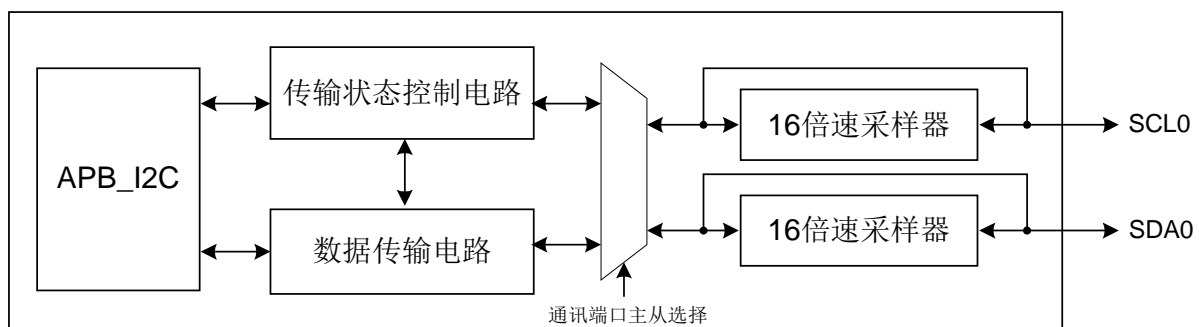


图 5-31 I2C 电路结构框图

### 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：

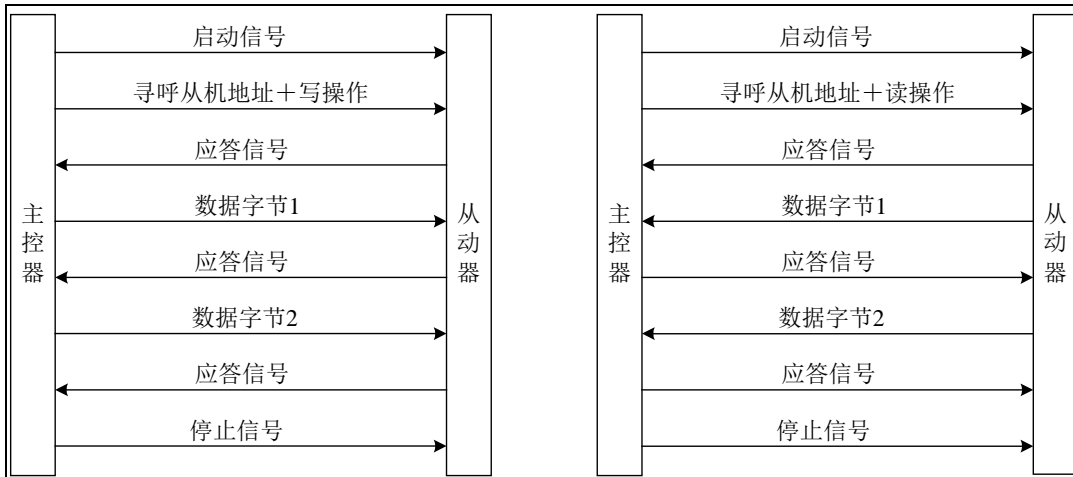


图 5-32 I2C 总线通讯协议示意图

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  $\overline{R/\overline{W}}$ ，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：

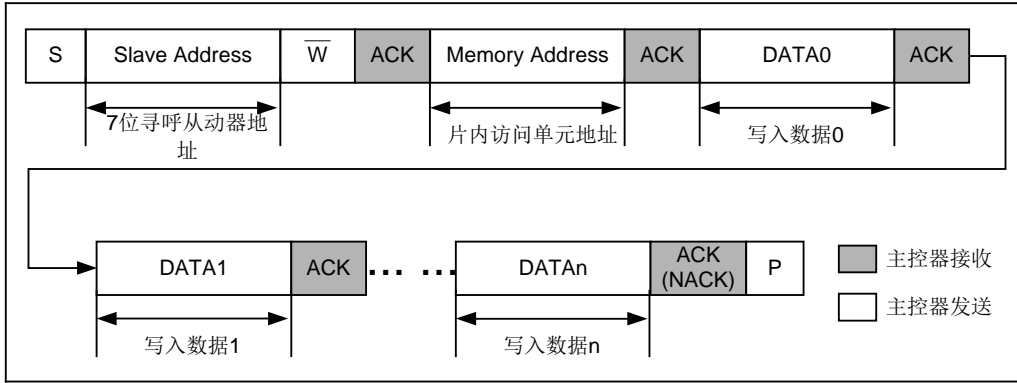


图 5-33 I2C 主控器写入从动器数据示意图

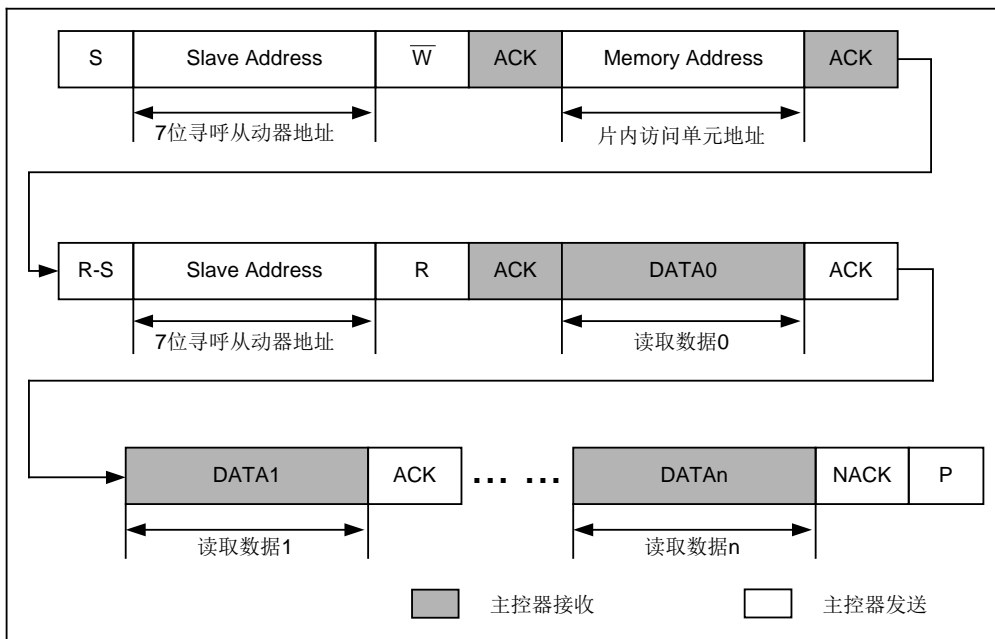


图 5-34 I2C 主控器读取从动器数据示意图

#### 5.4.4 I2C通讯端口配置

进行 I2C 通讯时，需通过寄存器 GPIO\_PAFUNCx/GPIO\_PBFUNCx，将对应 I/O 端口复用功能设置为 I2C 通讯端口 SCL0 和 SDA0，由 I2C 硬件电路控制通讯端口的输入和输出，与该 I/O 端口的方向控制寄存器无关，SCL0 和 SDA0 端口初始为高电平。

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 SCKOD 和 SDAOD，可分别进行选择，该配置仅在 I/O 端口开漏控制寄存器 GPIO\_PAODE/GPIO\_PBODE 保持为默认值（推挽输出）时有效，当 I/O 端口开漏控制寄存器将端口设置为开漏输出时，则该端口固定为开漏输出，与 I2C\_CON 寄存器无关。I2C 通讯端口 SCL0 和 SDA0 的内部弱上拉电阻，需通过对应 I/O 端口的弱上拉使能寄存器 GPIO\_PAPUE/GPIO\_PBPUE 进行设置。

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：

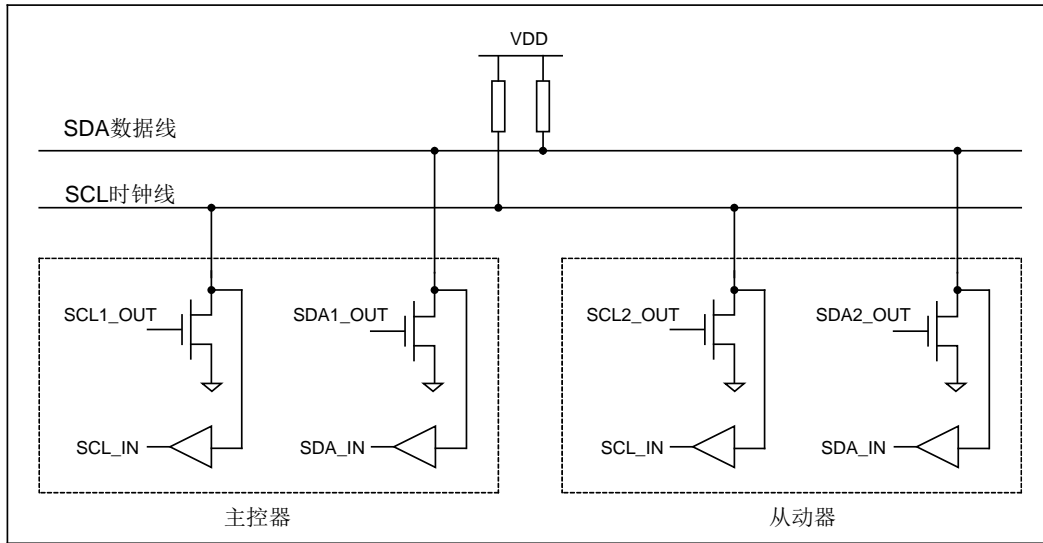


图 5-35 开漏输出端口示意图

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，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 SCKSE, 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，配置 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/重启位建立时间	T <sub>SU</sub> :S	$> T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
启动/重启位保持时间	T <sub>HD</sub> :S	$> T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
停止位建立时间	T <sub>SU</sub> :P	$> T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
停止位保持时间	T <sub>HD</sub> :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 <sub>SU</sub> :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 <sub>HD</sub> :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 <sub>HIGH</sub>	$T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$
通讯时钟低电平脉宽	T <sub>LOW</sub>	$T_{osc} \times (TJP+1) \times 12$	$T_{osc} \times (TJP+1) \times 8$

表 5-1 I2C 总线端口信号的时序参数列表

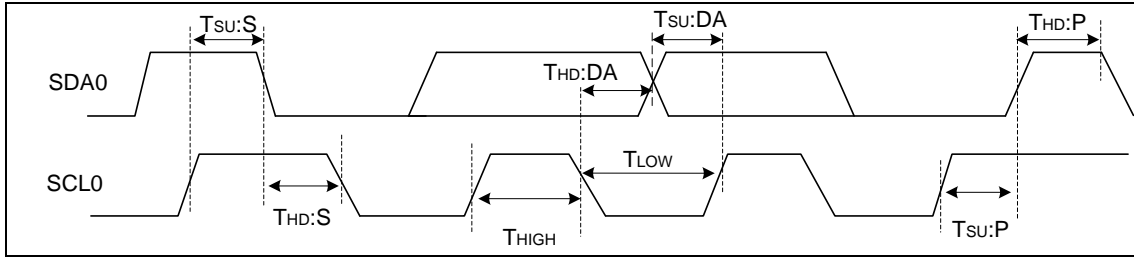


图 5-36 I2C 总线端口信号的波形示意图

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（Fosc 为系统时钟频率）：

使能通讯端口的 16 倍速采样器时： $F_{SCL} = F_{osc} / ((T_{JP} + 1) \times 24)$ ；

禁止通讯端口的 16 倍速采样器时： $F_{SCL} = F_{osc} / ((T_{JP} + 1) \times 16)$ 。

#### 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 TBy 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，其中字节写入时只能写低字节 I2C\_TBW<7:0>，半字写入时只能写低半字 I2C\_TBW<15:0>，否则会置起写错误中断标志 TBWEIF，写入的数据无效。

字节方式写入 I2C\_TBW 时，发送数据按先后顺序被依次写入到发送缓冲器 TB3~TB0，第一个数据保存在 TB3 中，如果只写一个字节数据，则该数据保存在 TB3 中，其他缓冲器为空；半字方式写入 I2C\_TBW 时，发送数据被同时写入到两级发送缓冲器中，先写入的半字数据保存在 TB2 和 TB3，其中低字节存放在 TB3 中，后写入的半字数据保存在 TB0 和 TB1，其中低字节存放在 TB1 中，如果只写一个半字数据，则该半字数据保存在 TB2 和 TB3 中，其他缓冲器为空；字方式写入 I2C\_TBW 时，发送数据被同时写入到发送缓冲器 TB0, TB1, TB2 和 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：

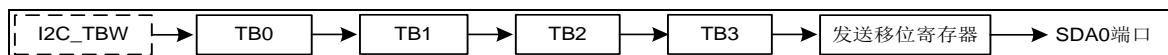


图 5-37 I2C 发送数据流示意图

发送缓冲器 TB0~TB3 的数据移到下一级缓冲器或发送移位寄存器后，会置起发送空标志 TBEF0~TBEF3。当缓冲器 TB0 的发送空标志 TBEF0 为 0 时，表示 4 级发送缓冲器和发送移位寄存器均满。



每级发送缓冲器空标志 TBEF0~TBEF3 只可用于查询判断，不能产生中断请求；发送缓冲器空中断标志 TBIF 即可用于查询判断，也可用于产生中断请求，配置寄存器 I2C\_IE 的 TBIM，可选择中断模式。

TBIM<1:0>=00，为字节空产生中断，有 1 级发送缓冲器（例如 TB0）为空时，会置起 I2C\_IF 寄存器的中断标志 TBIF，当前字节发送完毕并接收完应答信号（ACK 或 NACK）后，会置起中断标志 TIDLEIF；

TBIM<1:0>=01，为半字空产生中断，有 2 级发送缓冲器（例如 TB0 和 TB1）为空时，会置起 I2C\_IF 寄存器的中断标志 TBIF，当前半字发送完毕并接收完应答信号（ACK 或 NACK）后，会置起中断标志 TIDLEIF；

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，当前字发送完毕并接收完应答信号（ACK 或 NACK）后，会置起中断标志 TIDLEIF。

通常情况下选择使用字空产生中断方式（TBIM<1:0>=10），通过发送缓冲器空中断 TBIF，即可实现数据的连续发送，当全部缓冲器中的最后一个字节数据传输到发送移位寄存器，开始数据发送时，会置起发送缓冲器空中断标志 TBIF，即可向发送缓冲器中写入新的数据；如果发送的数据之间允许存在短暂的时间间隔，还可通过发送空闲中断 TIDLEIF，等发送缓冲器和发送移位寄存器中的数据全部发送完毕后，再向发送缓冲器中写入新的数据。

以字空产生中断方式（TBIM<1:0>=10），发送一个字节数据为例，对中断标志 TBIF 和 TIDLEIF 的置起说明如下图：

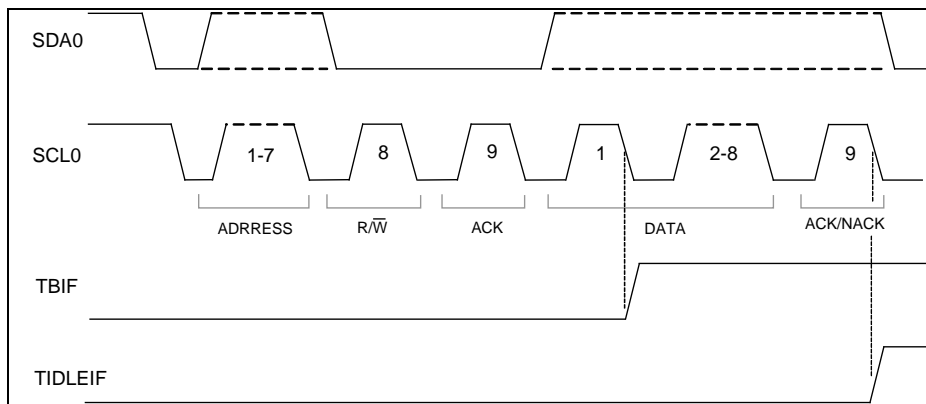


图 5-38 I2C 发送中断标志示意图

发送缓冲器空中断标志 TBIF 置起后，不能软件直接清 0，需要通过 I2C\_TBW 写入新的数据直到不满足由 TBIM 设置的发送缓冲器空中断条件，才能被硬件清 0；发送空闲中断标志 TIDLEIF 置起后，需由软件写 1 清 0。

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 寄存器写错误中断标志 TBWEIF。以下情况会导致写错误中断：当以字节方式写入 I2C\_TBW 时，发送缓冲器 TB0~TB3 全满；当以半字或字方式写入 I2C\_TBW 时，发送缓冲器只有一级为空或全满；当以字方式写入 I2C\_TBW 时，发送缓冲器未全空；以字节方式写入非最低字节 I2C\_TBW<31:8>；以半字方式写入高半字 I2C\_TBW<31:16>。发生上述写错误时，会置

起写错误中断标志 TBWEIF，同时新写入的数据无效，缓冲器数据仍保持。

I2C 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 I2C\_IF 寄存器中会置起发送错误中断标志 TEIF。

通过 I2C\_IE 寄存器的发送空闲中断使能位 TIDLEIE，发送缓冲器空中断使能位 TBIE，发送数据写错误中断使能位 TBWEIE 和发送错误中断使能位 TEIE，可对发送空闲中断标志 TIDLEIF，发送缓冲器空中断标志 TBIF，发送数据写错误中断标志 TBWEIF 和发送错误中断标志位 TEIF 是否触发 I2C 中断请求 IRQ，进行设置。

### 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，其他缓冲器的有效数据按顺序前移；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，缓冲器 RB2 和 RB3 的有效数据按顺序前移；字方式读取 I2C\_RBR 时，实际是同时读取接收缓冲器 RB0, RB1, RB2 和 RB3，其中 RB0 中的数据为低字节，是本次接收的第一个数据。

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：

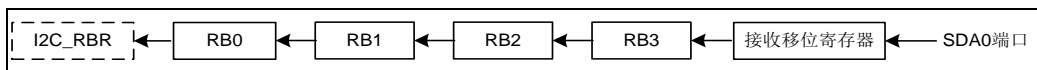


图 5-39 I2C 接收数据流示意图

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，该缓冲器的数据不再有效，直到接收新的数据。

当 4 级接收缓冲器和 1 级接收移位寄存器均满时，在发送完应答信号（ACK 或 NACK）后，会置起 I2C\_IF 寄存器中的接收数据溢出中断标志 ROIF，此后如果仍继续接收数据，则新数据会覆盖接收移位寄存器中的数据，4 级接收缓冲器中的数据保持不变，读取数据时，可读到 4 级接收缓冲器中的数据，以及接收移位寄存器中的数据。

每级接收缓冲器满标志 RBFF0~RBFF3 只可用于查询判断，不能产生中断请求；接收缓冲器满中断标志 RBIF 即可用于查询判断，也可用于产生中断请求，配置寄存器 I2C\_IE 的 RBIM，可选择中断模式。

RBIM<1:0>=00，为字节满产生中断，有 1 级接收缓冲器（例如 RB0）为满时，在发送完应答信号（ACK 或 NACK）后，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，有 2 级接收缓冲器（例如 RB0 和 RB1）为满时，

在发送完应答信号（ACK 或 NACK）后，会置起 I2C\_IF 寄存器的中断标 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，在发送完应答信号（ACK 或 NACK）后，会置起 I2C\_IF 寄存器的中断标志 RBIF。

通常情况下选择使用字节满产生中断方式(RBIM<1:0>=00)，通过接收缓冲器满中断 RBIF，即可实现数据的连续接收。

以字节满产生中断方式（RBIM<1:0>=00），接收一个字节数据为例，对中断标志 RBIF 的置起说明如下图：

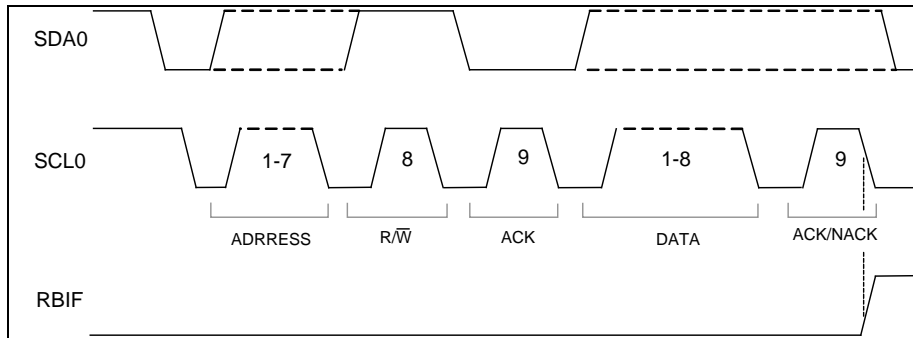


图 5-40 I2C 接收中断标志示意图

接收缓冲器满中断标志 RBIF 置起后，不能软件直接清 0，需要通过 I2C\_RBR 读取接收到的数据直到不满足由 RBIM 设置的接收缓冲器满中断条件，才能被硬件清 0。

通过 I2C\_IE 寄存器的接收缓冲器满中断使能位 RBIE 和接收数据溢出中断使能位 ROIE，可对接收缓冲器满中断标志 RBIF 和接收数据溢出中断标志 ROIF 是否触发 I2C 中断请求 IRQ，进行设置。

#### 5.4.8 I2C 通讯控制

配置 I2C\_CON 寄存器的 MS 位，可选择主控或从动通讯模式，配置 EN 位，可启用 I2C 通讯。

在主控通讯模式下，由芯片提供 I2C 通讯时钟，需配置 I2C\_CON 寄存器的 THE 位，使能 I2C 时基定时器，配置 TJP 位，设定时基定时周期，配置 I2C\_CON 寄存器的 SA 位，设置需寻呼的从机地址，配置 RW 位，设置读写操作位，设置 I2C\_MOD 寄存器的 SRT 位触发 I2C 起始位，设置 SPT 位触发 I2C 停止位；写操作时，通过读取 I2C\_STA 寄存器的 ACK 位，判断从机反馈的 ACK/NACK；读操作时，通过 I2C\_MOD 寄存器的 RDM 位，设置数据接收模式以及 ACK/NACK 的发送，通过设置 RDT 位为 1，使能接收数据时的通讯时钟发送。

在从动通讯模式下，芯片根据主机提供的 I2C 通讯时钟进行数据传输，配置 I2C\_CON 寄存器的 SA 位，设置本从机地址，读取 RW 位，判断主机的读写操作；主机进行读操作时，从机通过读取 I2C\_STA 寄存器的 ACK 位，判断主机反馈的 ACK/NACK，主机进行写操作时，从机通过 I2C\_MOD 寄存器的 TAS 位，设置 ACK/NACK 的发送。

I2C 主控模式下，收到从机的 NACK 后，会置起 I2C\_IF 寄存器的未应答中断标志 NAIF；I2C 从机模式下，收到主机的 NACK 后，会置起 I2C\_IF 寄存器的未应答中断标志 NAIF；通过 I2C\_IE 寄存器的 I2C 未应答中断使能位 NAIE，可对未应答中断标志 NAIF 是否触发

I2C 中断请求 IRQ，进行设置。

I2C 主控模式下，支持通过发送空白时钟来使从机释放数据线的的方式，可在主机空闲状态下 (I2C\_STA.IDLE=1)，设置 I2C\_MOD 寄存器的 BLD 位，触发 SCL0 端口发送 8 个时钟，此时 SDA0 端口保持为高电平，发送完毕 8 个时钟后，硬件自动将 BLD 位清 0，该方式可用于有此类需求的从动器件释放数据总线。

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, TIDLEIF=0；I2C\_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。

### 5.4.8.1 I2C起始位

I2C 主控模式下，配置 I2C\_MOD 寄存器的 SRT，可触发 I2C 发送起始位，启动或重新启动一次传输操作，自动发送读写操作位和寻呼从动器地址，其中读写操作位和需寻呼的从动器地址可分别通过寄存器 I2C\_CON 的 RW 和 SA 位进行设置。起始位波形图如下：

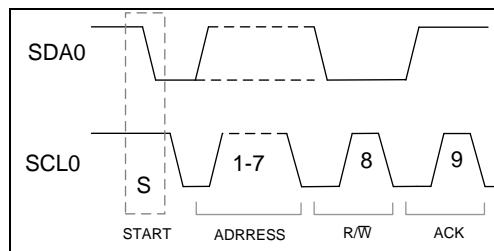


图 5-41 I2C 起始位波形图

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。

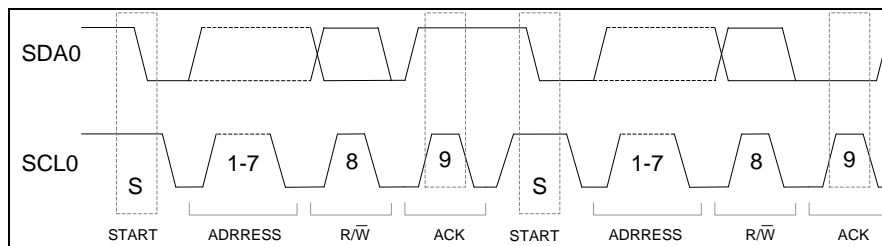


图 5-42 I2C 自动寻呼波形图

I2C 主控模式下，如果禁止自动寻呼，则发送完“起始位+地址+接收应答位后”产生中断标志 SRIF；如果使能自动寻呼，则发送完“起始位+地址+接受应答位”，并且应答位为 ACK 时，产生中断标志 SRIF。主控模式下 I2C\_STA 寄存器的空闲标志位 IDLE 在起始位发出后清 0。

I2C 从动模式下，本从动器地址可通过寄存器 I2C\_CON 的 SA 位进行设置，当接收到“起始位+地址位+读写控制位”，且地址匹配时，在发送应答位 ACK/NACK 之前，产生中断标志 SRIF。从动模式下 I2C\_STA 寄存器的空闲标志位 IDLE 在收到起始位后清 0。

通过 I2C\_IE 寄存器的 I2C 起始位中断使能位 SRIE，可对起始位中断标志 SRIF 是否触发 I2C 中断请求 IRQ，进行设置。

#### 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。

主机发送过程中配置 SPT=1 时，会在发送完当前字节数据，并接收完毕应答信号（ACK 或 NACK）后产生停止位，如果缓冲器中仍有未发送数据，则会被清空，可通过对发送空闲中断标志位 TIDLEIF 的查询或在其中断服务程序中配置 SPT=1，触发 I2C 发送停止位；主机接收过程中配置 SPT=1 时，会在接收完需要发送 NACK 的字节数据，并发送完毕应答信号（NACK）后产生停止位，可通过对接收缓冲器满中断标志位 RBIF 的查询或在其中断服务程序中配置 SPT=1，触发 I2C 发送停止位。

停止位波形图如下：

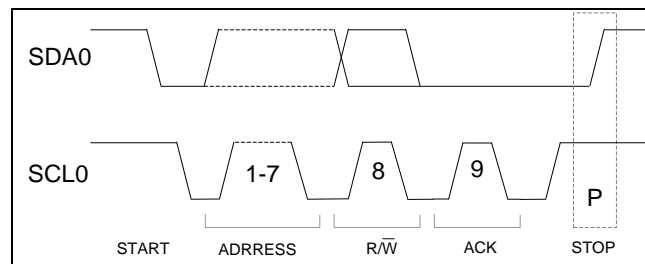


图 5-43 I2C 停止位波形图

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，结束本次传输操作。自动结束功能的优先级小于自动寻呼功能。

I2C 主控模式下，发送停止位后将 I2C\_STA 寄存器的空闲标志位 IDLE 置 1，然后产生中断标志 SPIF；I2C 从动模式下，接收到停止位后将 I2C\_STA 寄存器的空闲标志位 IDLE 置 1，并产生中断标志 SPIF。通过 I2C\_IE 寄存器的 I2C 停止位中断使能位 SPIE，可对停止位中断标志 SPIF 是否触发 I2C 中断请求 IRQ，进行设置。

#### 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<sub>SCL0</sub>，通讯波形示意图如下：

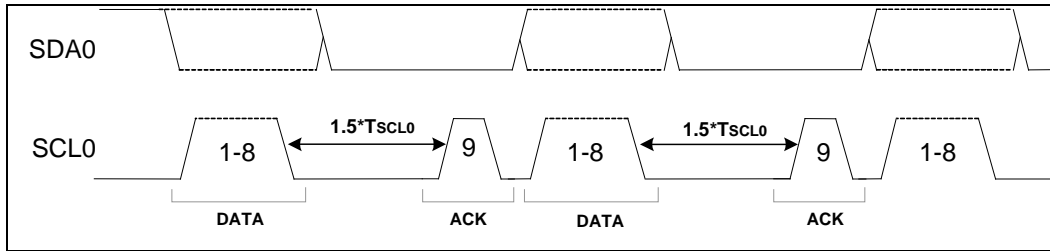


图 5-44 I2C 应答延迟功能波形示意图

#### 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，可启用该功能，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<sub>SCL0</sub>，通讯波形示意图如下：

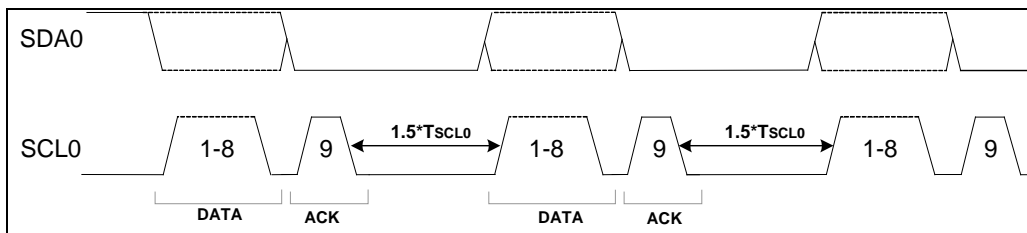


图 5-45 I2C 数据帧传输间隔功能波形示意图

#### 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，可启用该功能。

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 SCKOD，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：

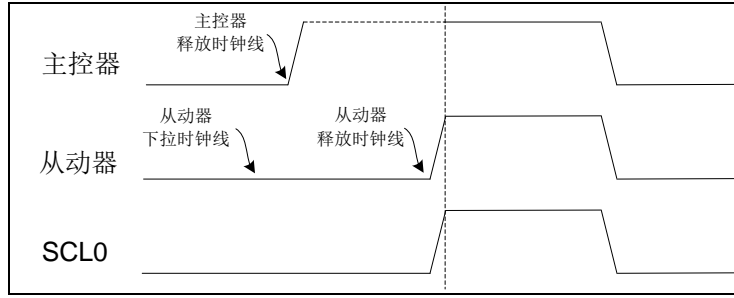


图 5-46 I2C 时钟线下拉等待波形示意图

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 4 级发送缓冲器全空，且自动发送未应答使能位 ANAE=0，则会自动将时钟线下拉，从机在数据发送过程中，如果发送缓冲器和发送移位寄存器全空，且仍收到主机的应答 ACK 信号时，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 4 级接收缓冲器全满，则会自动将时钟线下拉，从机在数据接收过程中，如果接收缓冲器和接收移位寄存器全满，且发送应答设置位 TAS=0 时（发送 ACK 信号），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，在接收到本芯片寻呼地址和读/写操作位后，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 4 级发送缓冲器全空，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 4 级接收缓冲器全满，则会自动发送未应答 NACK，通知主控器重新通讯，从机在数据接收过程中，如果接收缓冲器和接收移位寄存器全满，且时钟线自动下拉使能位 CSE=0 时（禁止自动下拉），则会自动发送未应答 NACK，其他情况下，从机发送的应答位由 TAS 位设置。

## 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>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留							SA<6:0>							RW
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	TJP<7:0>							TJE	保留	SDA SE	SCK SE	SDA OD	SCK OD	RST	EN

—	bit31-24	—	—
SA<6:0>	bit23-17	R/W	<b>从机地址位</b> 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 (仅主控模式支持)</b> 00~FF: 分别为 1~256 个 T <sub>PCLK</sub>
TJE	bit7	R/W	<b>I2C 时基定时器使能位 (仅主控模式支持)</b> 0: 禁止 1: 使能
—	bit6	—	—
SDASE	bit5	R/W	<b>SDA0 端口 16 倍速采样使能位</b> 0: 禁止 1: 使能
SCKSE	bit4	R/W	<b>SCL0 端口 16 倍速采样使能位</b> 0: 禁止 1: 使能
SDAOD	bit3	R/W	<b>SDA0 端口输出模式选择位</b> 0: 推挽输出 1: 开漏输出
SCKOD	bit2	R/W	<b>SCL0 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, 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>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留					TAS	保留					BLD	RDT	SPT	SRT
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	TIS<3:0>			ADE	ADLY<2:0>			SPA E	SRA E	ANA E	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: SDA0 端口为高电平, SCL0 端口发送 8 个时钟, 完成后由硬件自动清 0 该功能需配合 I2C 起始位和停止位使用, 达到从机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，则自动重启动本次寻呼操作）
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 1: 如果 ANAE 与 CSE 同时为 1，则 ANAE 优先级高，即当接收到本芯片寻呼地址和读/写操作位时，发送未应答位 NACK 后，不会将 I2C 时钟线自动下拉；

注 2: ANAE 为 1，使能自动发送未应答位，仅适用于通讯端口 SDA0 选择为开漏输出模式的情况。

### 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>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	保留	TIDL EIE	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>																
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
1	保留															
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
5	保留															
	保留	TIDL EIF	保留				TBW EIF	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 (满足 TBIM 所选择的条件) 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 (满足 RBIM 所选择的条件)
TBIF	bit2	R	<b>I2C 发送缓冲器空中断标志位</b> 0: 非空 1: 空 (满足 TBIM 所选择的条件)
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	<p><b>写入的发送数据</b></p> <p>字节写入时：仅允许对 I2C_TBW&lt;7:0&gt;写入 半字写入时：仅允许对 I2C_TBW&lt;15:0&gt;写入 字写入时：对 I2C_TBW&lt;31:0&gt;写入</p>
-----------	---------	---	--

#### 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><b>读取的接收数据</b></p> <p>字节读取时：仅允许对 I2C_RBR&lt;7:0&gt;读取 半字读取时：仅允许对 I2C_RBR&lt;15:0&gt;读取 字读取时：对 I2C_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: 20H																
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	—	—

#### 5.4.9.10 I2C控制寄存器 2 (I2C\_CON2)

I2C 控制寄存器 2 (I2C_CON2)															
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							FLTEN	DLY<7:0>							

—	bit31-9	—	—
FLTEN	bit8	R/W	<b>SCL0、SDA0 端口输入 50ns 模拟滤波控制位</b> 0: 关闭输入模拟滤波功能 1: 使能输入模拟滤波功能, 滤波时间约 50ns
DLY<7:0>	bit7-0	R/W	<b>SDA0 端口输出延时控制位 (仅从动模式支持)</b> 00: 关闭延时功能 01~FF: 延时分别约为 3~258 个 T <sub>PCLK</sub>

#### 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, 在中断服务程序中发送数据。主机读完数据后, 必须发送 STOP 位, 否则, 若要直接再次 START 读操作, 可能导致数据发送错误。

I2C\_IE 寄存器的 TBIM (发送缓冲器空中断模式选择位) 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) 产生中断模式, 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 27 个外部模拟输入通道
- ◇ 支持 1 路内部 1/4VDD 通道，对应于通道 18
- ◇ 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

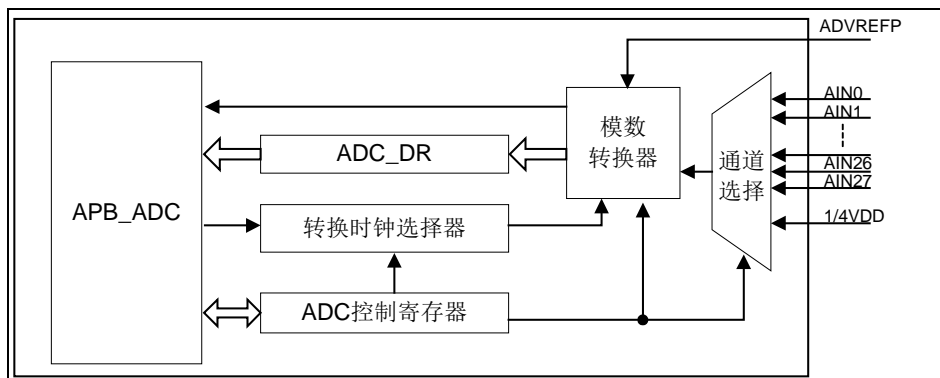


图 5-47 ADC 内部结构图

### 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 IO 端口输入控制寄存器 GPIO\_PAINEB/GPIO\_PBINEB，关断该端口的数字输入；  
配置 IO 端口方向控制寄存器 GPIO\_PADIR/ GPIO\_PBDIR，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，等待至少 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), 设置 ADC\_VREFCON 寄存器的 IREF\_EN 位使能 IREF, 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 ADC\_CON0 寄存器的 EN 位使能 ADC; 等待至少 300us 以后, 再设置 CHOP\_EN 位使能内部参考电压斩波器; 等待至少 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, 将结果保存在 ADC\_DR 寄存器中, 并会产生 ADC\_IF 寄存器的中断标志 IF, 需软件清零, 通过 ADC\_IE 寄存器的 ADC 中断使能位 IE, 可对 ADC 中断标志 IF 是否触发 ADC 中断请求 IRQ, 进行设置; 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 15 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, 当 SMPON=1 时进行采样, SMPON=0 时结束采样并启动 A/D 转换。通常情况下使用硬件控制即可满足 AD 采样需求, 当应用系统需要特别加长采样时间时, 可使用软件控制 AD 采样。

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

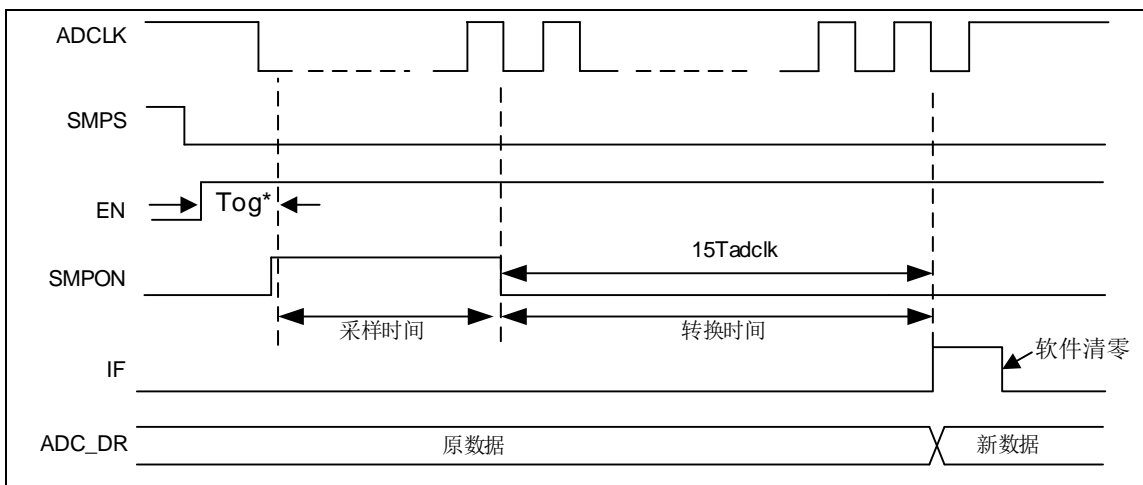


图 5-48 ADC 数据转换时序示意图 (ADC\_CON1 寄存器的 SMPS=0, 软件控制采样)

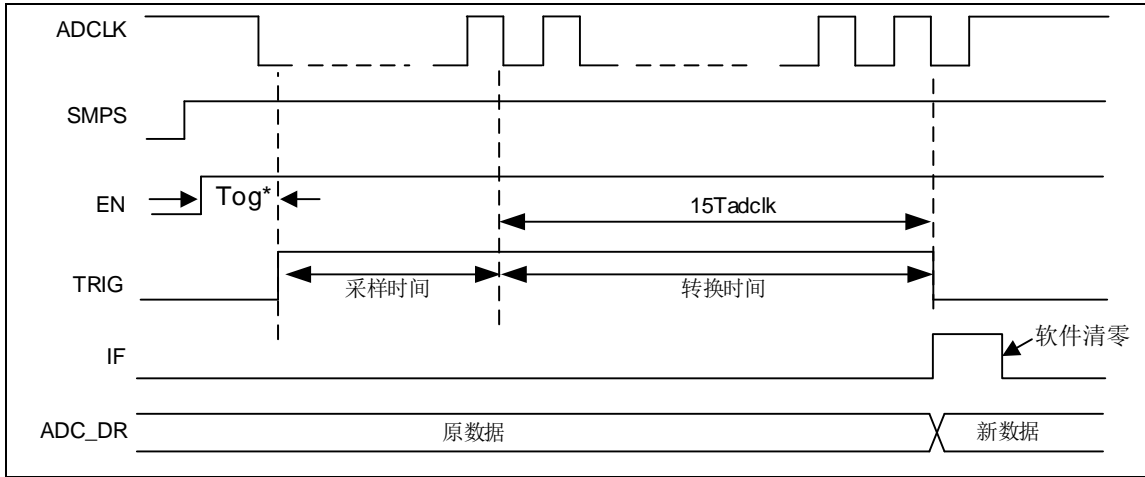


图 5-49 ADC 数据转换时序示意图 (ADC\_CON1 寄存器的 SMPS=1, 硬件控制采样)

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
LDR    R1, =0X00021605    ;参考电压 VREF 2.048V 为正向参考电压,
STR    R1, [R0]           ;选择硬件控制采样, AD 转换高速使能

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]
.....
LDR    R0, =ADC_VREFCON   ;使能 CHOP_EN
LDR    R1, =0X0D
STR    R1, [R0]
.....
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\_TIME，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滿，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（31.25KHz）时钟的 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。

通过 ADC\_IE 寄存器的 ADC 转换溢出中断使能位 ACPOVIE，转换高阈值超出中断使能位 ACPMAXIE，转换低阈值超出中断使能位 ACPMINIE，可对 ADC 转换溢出中断标志位 ACPOVIF，转换高阈值超出中断标志位 ACPMAXIF，转换低阈值超出中断标志位 ACPMINIF 是否触发 ADC 中断请求 IRQ，进行设置。

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 32xTacp

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: 40H

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	<b>内部 VREFP 使能控制位</b> 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, 然后等待至少 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: 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															
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	<b>A/D 转换结果 (对不同分辨率均高位对齐)</b>

### 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>

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

—	bit31-6	—	—
BIT_SEL<1:0>	bit5-4	R/W	<b>A/D 分辨率选择位</b> 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	<b>A/D 自动转换比较功能使能位</b> 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 SMPON 位进行软件采样和启动 A/D 转换的控制), 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: 硬件采样模式下, TRIG 与 ADC\_IF 寄存器的 IF 位均可作为转换完成标志。软件采样模式下, 仅 ADC\_IF 寄存器的 IF 位可作为转换完成标志。建议无论哪种采样模式下, 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, 然后等待至少 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: 设置不同的 BIT\_SEL, ADC 转换时间 (不含采样时间) 会有差异, 12 位对应 15T, 10 位对应 13T, 8 位对应 11T, 6 位对应 9T。

### 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 Tadclk
—	bit15	R/W	<b>需软件固定为 0, 禁止设置为 1</b>
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 (31.25KHz)
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, 然后等待至少 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 3MHz, 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) 10100: 通道 20 (AIN20) 10101: 通道 21 (AIN21) 10110: 通道 22 (AIN22) 10111: 通道 23 (AIN23) 11000: 通道 24 (AIN24) 11001: 通道 25 (AIN25) 11010: 通道 26 (AIN26) 11011: 通道 27 (AIN27) 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 <b>0~9C3<sub>H</sub></b> , 计数时钟周期为 <b>Tacp</b> , 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 硬件独立看门狗 (IWDT)

### 5.6.1 概述

- ◇ 支持硬件使能和关闭看门狗
  - 芯片配置位 CFG\_IWDTEN 位配置为 1 或 0 来使能或关闭 IWDT
  - 硬件使能后不可通过软件关停
  - 硬件使能时 IWDT 时钟强制为 31.25KHz LRC 时钟
- ◇ IWDT 中断可唤醒深度睡眠
- ◇ IWDT 溢出时间可设定
  - 写入 IWDT\_LOAD 寄存器将重新加载看门狗
  - 溢出时产生 IWDT 复位

IWDT 看门狗可用于检测软件和硬件异常。可由 CFG\_IWDTEN 配置位硬件强制使能，提高系统可靠性。硬件使能时，时钟强制为独立的 31.25KHz LRC 时钟，保证即使系统时钟失效，IWDT 仍然工作，IWDT 中断和复位也均强制使能。

当配置字 CFG\_WORD 配置为“硬件看门狗使能”时，即 CFG\_IWDTEN 位配置为 1，上电复位后看门狗立即启动（时钟固定为 31.25KHz LRC 时钟），IWDT 载入 IWDT\_LOAD 值（默认值为 0x0000\_4000，约 0.5s），并进行递减计数，计数到 0 时，IWDT 产生 IWDT\_RIS 寄存器的中断标志 IWDTIF，并在下一计数时钟到来时，计数器再次载入 IWDT\_LOAD 参数对应的计数初值，并继续递减计数。当计数器再次计数到 0 时，如果 IWDT 中断标志 IWDTIF 仍没有被软件清零，则 IWDT 模块将产生复位信号。软件只能读写 IWDT\_LOCK、写 IWDT\_INTCLR，其他 IWDT 相关特殊功能寄存器不再可写，只能读取，读 IWDT\_CON 将返回“0x0000\_000F”。

当配置字 CFG\_WORD 配置为“硬件看门狗由软件控制”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 IWDT\_LOAD 寄存器。当配置为“软件看门狗使能”时，即软件配置 IWDT\_CON 寄存器的 EN 为 1，IWDT 计数器载入 IWDT\_LOAD 寄存器值，开始递减计数，当计数到 0 时，IWDT 产生 IWDT\_RIS 寄存器的中断标志 IWDTIF，并在下一个计数时钟到来时，计数器再次载入 IWDT\_LOAD 寄存器值，并继续递减计数。当计数器再次计数到 0 时，如果 IWDT 中断标志 IWDTIF 仍没有被软件清零，并且 IWDT\_CON 寄存器的 IWDT 复位使能位 RSTEN 使能，则 IWDT 模块将产生复位信号。配置 CLKS，可以选择计数时钟源；配置 IWDT\_LOAD 寄存器，可设置计数初值；读取 IWDT\_VALUE，可得到 IWDT 当前计数值。写入装载值寄存器 IWDT\_LOAD 时，计数器当前值寄存器将被载入 IWDT\_LOAD 寄存器值。通过 IWDT\_CON 寄存器的 IWDT 中断使能位 IE，可对 IWDT 中断标志 IWDTIF 是否触发 IWDT 中断请求 IRQ，进行设置。

程序可通过读取芯片配置字（CFG\_WORD0，地址 0x2\_0000）来判断硬件看门狗是否使能，若 CFG\_WORD0.CFG\_IWDTEN 是 1，则表明硬件看门狗已经使能，否则硬件看门狗未使能。

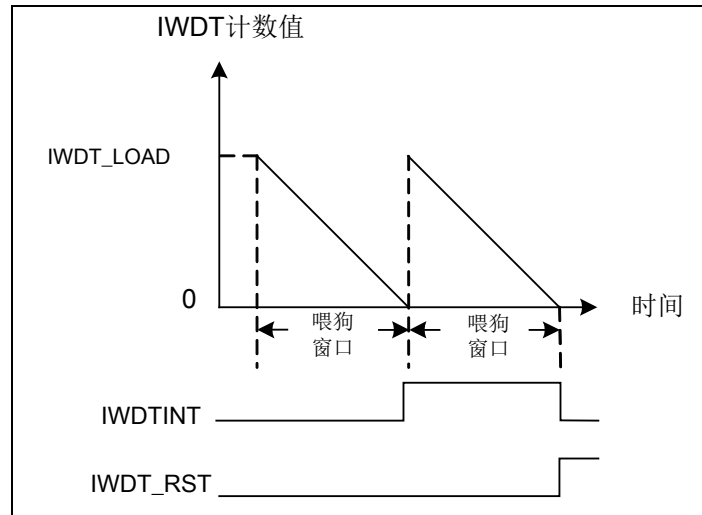


图 5-50 看门狗时序图

写 IWDT\_INTCLR，则 IWDT 重新载入计数初值，并重新进行递减计数。

IWDT 模块支持寄存器写保护，配置寄存器 IWDT\_LOCK=0x1ACCE551，可去除 IWDT 寄存器的写保护状态，此后可进行被保护的 IWDT 寄存器的写操作，否则无法写 IWDT 寄存器。

## 5.6.2 特殊功能寄存器

### 5.6.2.1 IWDT计数器装载值寄存器 (IWDT\_LOAD)

IWDT 计数器装载值寄存器 (IWDT_LOAD)																
偏移地址: 00 <sub>H</sub>																
复位值: 00000000_00000000_01000000_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	IWDT 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>																
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
1	VALUE<31:16>															
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
5																

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												CLK	RST	IE	EN
保留												S	EN		

—	bit31-4	—	—
CLKS	bit3	R/W	<b>IWDT 计数时钟选择位</b> 0: PCLK 1: LRC 时钟 (约 31.25KHz)
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\_WORD 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>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	INTCLR<31:16>														
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	INTCLR<15:0>														



INTCLR<31:0>	bit31-0	W	<b>IWDT 中断标志清 0 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, 即对应喂狗操作。
--------------	---------	---	---

### 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															IWD TIF

—	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锁定寄存器 (IWDT\_LOCK)

IWDT 锁定寄存器 (IWDT_LOCK)																
偏移地址: 100 <sub>H</sub>																
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>																
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
1	保留															
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
5	保留															LOC K

—	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	<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
  - ◇ 在喂狗禁止区内喂狗，产生 WWDT 复位
  - ◇ 在喂狗窗口内产生 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早或过晚行为，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，可在精确的时间内检测异常情况。

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 WWDT\_RIS 寄存器的中断标志 WWDTIF；若 WWDTWIN 设置为 50%，则窗口计数器为 2 时，WWDT 产生中断标志 WWDTIF；若 WWDTWIN 设置为 75%，则窗口计数器为 3 时，WWDT 产生中断标志 WWDTIF。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，并且 WWDT\_CON 寄存器的 WWDT 复位使能位 RSTEN 使能，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 WWDT 看门狗要使能）。通过 WWDT\_CON 寄存器的 WWDT 中断使能位 IE，可对 WWDT 中断标志 WWDTIF 是否触发 WWDT 中断请求 IRQ，进行设置。

若配置字 CFG\_WORD 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。

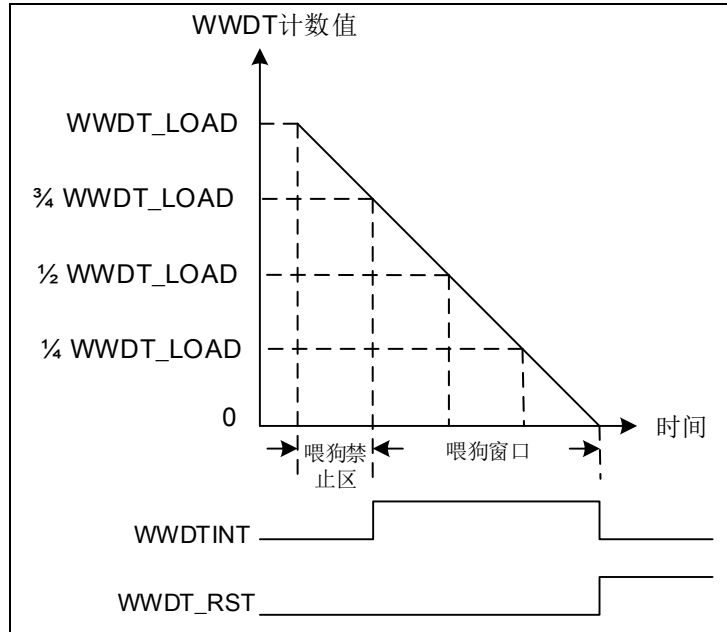


图 5-51 窗口看门狗中断和溢出复位产生时序图 (WWDTWIN 设定为 25%)

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。

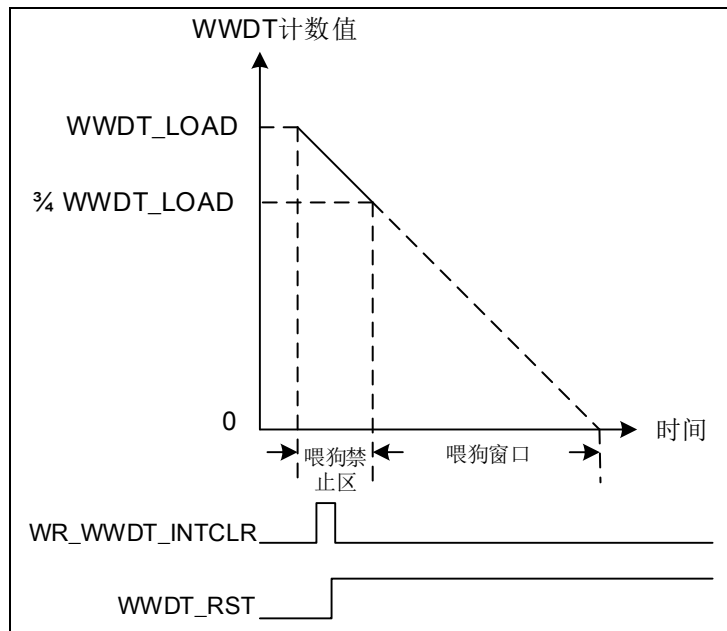


图 5-52 错误的喂狗时序图 (WWDTWIN 设定为 25%)

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>															
3	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	保留														
1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	保留										WWDTWIN<1	CLK	RST	IE	EN
											:0>	S	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, 在禁止窗口内喂狗产生复位												

			01: 50%窗口内禁止喂狗，在禁止窗口内喂狗产生复位 10: 75%窗口内禁止喂狗，在禁止窗口内喂狗产生复位 11: 保留
CLKS	bit3	R/W	<b>WWDT 计数时钟选择位</b> 0: PCLK 1: LRC 时钟 (约 31.25KHz)
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

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

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

—	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。

## 5.8 循环冗余校验 (CRC)

### 5.8.1 概述

- ◇ 支持 CRC-16 和 CRC-32
  - CRC-16:  $X^{16} + X^{15} + X^2 + 1$
  - CRC-16-CCITT:  $X^{16} + X^{12} + X^5 + 1$
  - CRC-32:  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- ◇ 支持 8/16/32 位宽数据
  - 通过 CRC\_CON 寄存器的 BYTE<1:0>选择位宽
  - 单系统时钟周期完成一个字节 CRC 计算, 对 32 位宽数据只需 4 个系统时钟周期
- ◇ 支持对 Flash 数据块的 CRC 校验
  - 以起始地址开始, 按设定的块大小, 自动对数据块进行 CRC 校验
  - 可用于 IAP 编程数据校验
  - 支持对已加密空间的 CRC 校验
- ◇ 可作为通用 CRC 模块
  - 用户可将数据写入用户模式输入数据寄存器 CRC\_DI, 计算结果从校验结果寄存器 CRC\_DO 读取

### 5.8.2 运用说明

CRC 模块支持寄存器写保护, 配置寄存器 CRC\_UL=0x43524355, 可去除 CRC 寄存器的写保护状态, 此后可进行被保护的 CRC 寄存器的写操作, 否则无法写 CRC 寄存器。

FLASH 查空或编程校验步骤:

1. 通过 CRC Flash 校验起始地址寄存器 CRC\_ADDR 可设置起始地址, 通过校验数据块大小寄存器 CRC\_SIZE 可设置数据块长度, 若长度设置小于 8, 则硬件自动固定为 8, 防止在 Flash 加密时被破解。
2. 通过 CRC 控制寄存器 CRC\_CON 的各个控制位进行 CRC 工作配置: 由 DS 位设置初始数据格式, 由 REFIN 位设置数据输入顺序, 由 REFOUT 位设置数据输出顺序, 由 XOROUT 设置数据输出是否取反, 由 BYTE 位设置位宽, 由 MOD 位设置 CRC 模式。
3. 在对 FLASH 进行 CRC 操作时需避免与 IAP 访问 FLASH 操作冲突, 必须先清除 IAP\_FLASHCR 寄存器的 FLASH 操作请求使能位 FLASHREQ=0, 然后置位 CRC\_CON 寄存器的 CRC 访问 FLASH 请求位 REQ=1, 查询 CRC 访问 FLASH 应答位 ACK=1 后, 再开始后续的 CRC 操作。
4. 通过寄存器 CRC\_TRIG, 可选择 CRC 工作模式为 Flash 校验或查空, 并触发 CRC 计算, 在 CRC 计算过程中, 状态寄存器 CRC\_STA 的 BUSY 变为 1, 通过 Flash 地址寄存器 CRC\_FA 可查询当前正在进行 CRC 计算的 Flash 单元地址, CRC 计算完成后, 寄存器 CRC\_TRIG 清 0, BUSY 变为 0, 用户可根据 BUSY 位判断操作是否完成 (对用户定义的数据进行 CRC 校验, 会立即完成, 无需判断), 可通过寄存器 CRC\_DO 读取校验结果。因 FLASH 被 CRC 模块占用, 对 CRC 操作的查空由 ROM



table 中的 CRC 函数完成，用户亦可将 CRC 状态查询程序复制到 SRAM 中执行。

5. 校验时，若单次校验的数据为半字或字格式，则先校验的是低字节。
6. 当执行 FLash 查空时，若发生查空错误，则停止在当前错误地址，BUSY 变为 0，并且置起寄存器 CRC\_STA 的 EMPTY\_ERR 标志位，软件可通过读取 CRC\_FA 寄存器来查询查空错误的地址。

当 CRC 工作模式为用户校验时，只需通过 CRC 控制寄存器 CRC\_CON 的各个控制位进行 CRC 工作配置：由 DS 位设置初始数据格式，由 REFIN 位设置数据输入顺序，由 REFOUT 位设置数据输出顺序，由 XOROUT 设置数据输出是否取反，由 BYTE 位设置位宽，由 MOD 位设置 CRC 模式，无需进行 FLASH 校验/查空相关的其他寄存器设置；将数据写入寄存器 CRC\_DI，然后通过寄存器 CRC\_TRIG，选择 CRC 工作模式为用户校验，并触发 CRC 计算，通过寄存器 CRC\_DO 读取校验结果。

配置 CRC\_CON 寄存器的 RST，可将 CRC 模块软件复位，复位后：CRC 上锁，各寄存器不可写；CRC\_STA 寄存器的 RSTF=1，EMPTY\_ERR=0；CRC\_FA 寄存器清 0；CRC 计算逻辑电路被复位。

### 5.8.3 特殊功能寄存器

#### 5.8.3.1 CRC解锁寄存器 (CRC\_UL)

CRC 解锁寄存器 (CRC_UL)															
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
UL<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UL<15:0>															

UL<31:0>	bit31-0	R/W	<b>CRC 解锁:</b> 写入 0x4352_4355, CRC 解锁; <b>CRC 上锁:</b> 进行如下任一操作均可上锁 写入其它值, CRC 上锁; CRC 软件复位后, CRC 上锁。
----------	---------	-----	---

注: CRC 上锁后, 处于写保护状态的寄存器为 CRC\_CON, CRC\_TRIG, CRC\_ADDR, CRC\_SIZE, CRC\_DI, CRC\_DO, CRC\_STA。

#### 5.8.3.2 CRC控制寄存器 (CRC\_CON)

CRC 控制寄存器 (CRC_CON)															
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	XO RO UT	REF OU T	REF IN	BYTE	MOD<1:0>	保留	ACK	RE Q	DS	HS	RST	EN			

—	bit31-15	—	—
XOROUT	bit14	R/W	<b>数据输出取反选择位</b> 0: 不取反 1: 取反 (对数据按位取反)
REFOUT	bit13	R/W	<b>数据输出顺序选择位</b> 0: 正序 (原始数据格式, 高位在前) 1: 倒序 (对数据位反序, 低位在前)
REFIN	bit12	R/W	<b>数据输入顺序选择位</b> 0: 正序 (原始数据格式, 高位在前) 1: 倒序 (对数据位反序, 低位在前)
BYTE<1:0>	bit11-10	R/W	<b>CRC 位宽选择位</b> 00: 字节

			01: 半字 1x: 字
MOD<1:0>	bit9-8	R/W	<b>CRC 模式选择位</b> 0x: CRC32 10: CRC16 11: CRC16-CCITT
—	bit7-6	—	—
ACK	bit5	R	<b>CRC 访问 FLASH 应答位</b> 0: 不允许操作 Flash 1: 允许操作 Flash
REQ	bit4	R/W	<b>CRC 访问 FLASH 请求位</b> 0: 无操作 1: 发起请求操作 Flash(仅在 IAP 访问 FLASH 请求信号 FLASH_REQ=0 时有效)
DS	bit3	R/W	<b>CRC 初始数据格式选择</b> 0: 初始为全 0 1: 初始为全 1
HS	bit2	R/W	<b>CRC 高速模式使能位</b> 0: 禁止 1: 使能
RST	bit1	W	<b>CRC 复位</b> 0: 无任何操作 1: 复位
EN	bit0	R/W	<b>CRC 使能位</b> 0: 禁止 1: 使能

注 1: 当 CRC 时钟小于 24MHz 时, CRC 高速模式使能位 HS 才可使能, 该模式下读 FLASH 等待一个时钟周期, 否则等待两个时钟周期。

注 2: 在对 FLASH 进行 CRC 访问操作前, 必须先清除 IAP\_FLASHCR 寄存器的 FLASH 操作请求使能位 FLASH\_REQ=0, 然后置位 CRC 访问 FLASH 请求位 REQ=1, 查询 CRC 访问 FLASH 应答位 ACK=1 后, 再开始 CRC 操作。

### 5.8.3.3 CRC 触发寄存器 (CRC\_TRIG)

#### CRC 触发寄存器 (CRC\_TRIG)

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
TRIG<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIG<15:0>															

TRIG<31:0>	bit31-0	R/W	<b>CRC 工作模式选择</b> 0x00006E91: Flash 校验 0x00006D92: Flash 查空 0x00006C93: 用户校验 其他: 保留未用 CRC 计算完成后自动硬件清 0
------------	---------	-----	---

### 5.8.3.4 CRC FLASH 校验起始地址寄存器 (CRC\_ADDR)

#### CRC FLASH 校验起始地址寄存器 (CRC\_ADDR)

偏移地址: 0CH

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											IFRE N	保留	ADDR<17:1 6>		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR<15:0>															

—	bit31-21	—	—
IFREN	bit20	R/W	<b>Flash 区域选择位</b> 0: 选择 Code 区 1: 选择 Info 区
—	bit19-18	—	—
ADDR<17:0>	bit17-0	R/W	<b>CRC 起始地址 (字地址)</b> 最低 2 位必须为 0

注: 寄存器 CRC\_ADDR 用于设定 FLASH 编程校验或擦除查空校验的起始地址。

### 5.8.3.5 CRC FLASH 校验数据块大小寄存器 (CRC\_SIZE)

#### CRC FLASH 校验数据块大小寄存器 (CRC\_SIZE)

偏移地址: 10H

复位值: 00000000\_00000000\_00000000\_00000111<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
CRC_SIZE<15:0>															

—	bit31-16	—	—
CRC_SIZE<15:0>	bit15-0	R/W	<b>CRC 数据块长度 (以字为单位)</b> CRC 数据块长度为 CRC_SIZE + 1, 当小于 8 个字时, 固定为 8

注：寄存器 CRC\_SIZE 用于设定 FLASH 校验或查空的数据块的大小，以 WORD（4 个字节）为单位。

### 5.8.3.6 CRC用户模式输入数据寄存器（CRC\_DI）

#### CRC 用户模式输入数据寄存器（CRC\_DI）

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

DI<31:16>

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

DI<15:0>

DI<31:0>	bit31-0	R/W	<b>CRC 用户数据</b> CRC_TRIG 寄存器设置为用户校验模式时，该寄存器为用户写入的需进行 CRC 校验计算的数据
----------	---------	-----	---

### 5.8.3.7 CRC校验结果寄存器（CRC\_DO）

#### CRC 校验结果寄存器（CRC\_DO）

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

DO<31:16>

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

DO<15:0>

DO<31:0>	bit31-0	R	<b>CRC 校验结果</b>
----------	---------	---	-----------------

### 5.8.3.8 CRC状态寄存器（CRC\_STA）

#### CRC 状态寄存器（CRC\_STA）

偏移地址：1C<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			EMP TY_ ERR	RST F	BUS Y
----	--	--	-------------------	----------	----------

—	bit31-3	—	—
EMPTY_ERR	bit2	R/W	查空错误标志位

			0: 无错误 1: 错误 (写 1 或再次触发 Flash 校验/查空清除)
RSTF	bit1	R/W	<b>CRC 复位标志位</b> 0: 未发生复位 1: 发生复位 (写 1 清除)
BUSY	bit0	R	<b>CRC 工作状态位</b> 0: 已完成 CRC 计算 1: 正在进行 CRC 计算

### 5.8.3.9 CRC当前Flash地址寄存器 (CRC\_FA)

#### CRC 当前 Flash 地址寄存器 (CRC\_FA)

偏移地址: 20<sub>H</sub>

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



—	bit31-21	—	—
IFREN	bit20	R	<b>CRC 对应的当前 Flash 区域</b> 0: Code 区 1: Info 区
—	bit19-18	—	—
FADR<17:0>	bit17-0	R	<b>CRC 对应的当前 Flash 地址 (字对齐)</b>

## 5.9 液晶驱动控制器 (LCD)

### 5.9.1 概述

LCD 控制器根据用户设定的帧频率与类型控制输出波形，支持三种闪烁模式供用户选择，闪烁周期可设置。

### 5.9.2 特性

- ◆ 支持 4x44、6x42 与 8x40 的 COM/SEG 组合
- ◆ 支持 A、B 两种类型输出驱动波形
- ◆ 支持显示灰度调节功能
- ◆ 支持内部偏压电阻可调
- ◆ 支持四段可调式偏置电压
  - ◇ 静态偏压
  - ◇ 1/2 偏压
  - ◇ 1/3 偏压
  - ◇ 1/4 偏压
- ◆ 支持三种 LCD 输出波形占空比
  - ◇ 1/4 占空比
  - ◇ 1/6 占空比
  - ◇ 1/8 占空比
- ◆ 支持 LCD 显示帧频率可配置
- ◆ 支持三种闪烁模式

### 5.9.3 结构框图

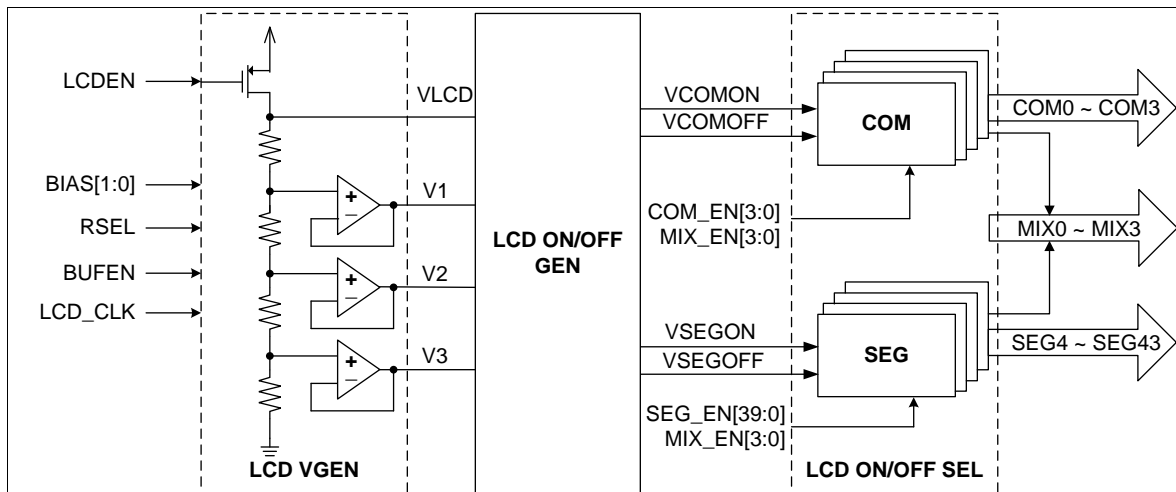


图 5-53 LCD 驱动结构图

注：仅 ES8H0394ELLT 和 ES8H0394ELLP 支持 LCD 控制器。

### 5.9.4 LCD工作时钟

LCD 控制器支持两种时钟源，可以通过寄存器 SCU\_LCDCLK 的 CLKSEL 位选择时钟源为 LOSC 或 LRC（约 31.25KHz），可通过 CLKDIV 位设置时钟源分频，分频后的时钟作为 LCD 控制器工作时钟 LCD\_CLK。

### 5.9.5 LCD显示帧频率

LCD 的工作时钟为 LCD\_CLK，通过寄存器 LCD\_CTRL 的 FRDIV 位，可设置 LCD 显示帧频率，需要依据外接 LCD 显示器的规格设置正确的 LCD 显示帧频率和波形类型。产生帧频率的时钟为 LCD\_PHY\_CLK，其频率计算公式如下：

$$f_{\text{LCD\_PHY\_CLK}} = f_{\text{LCD\_CLK}} / (2 * (\text{FRDIV} + 1)), \text{ 其中 } \text{FRDIV} = 0 \sim 127。$$

帧频率计算公式如下：

$$f_{\text{LCD\_PHY}} = f_{\text{LCD\_PHY\_CLK}} / n, \text{ 其中 } n \text{ 由 COM 占空比决定。}$$

当 COM 占空比为 1/4 时，n=4，占空比为 1/6 时，n=6，占空比为 1/8 时，n=8，COM 占空比可通过 LCD\_MODE 寄存器的 DUTY 位设置。

### 5.9.6 LCD闪烁功能

LCD 控制器支持三种闪烁模式，可以通过寄存器 LCD\_CTRL 的 WSBLINK 位进行选择。

- ◆ WSBLINK=01，为闪烁模式 0，LCD 显示在像素点设置数值和其反码数值之间切换。
- ◆ WSBLINK=10，为闪烁模式 1，LCD 显示在像素点设置数值和全部熄灭之间切换。
- ◆ WSBLINK=11，为闪烁模式 2，由寄存器 LCD\_BLINK1 和 LCD\_BLINK2 进行闪烁控制，其中每个 COM 有 8 个位，分别控制低 8 位 SEG 端口 SEG7~0 的输出，当设置为 1 时闪烁，否则不闪。

通过寄存器 LCD\_CTRL 的 FCVALUE 位可进行闪烁周期的设置，闪烁频率计算公式如下：

$$f_{\text{LCD\_BLINK}} = f_{\text{LCD\_PHY\_CLK}} / (2 * (\text{FCVALUE} + 1)), \text{ 其中 } \text{FCVALUE} = 0 \sim 127。$$

如果关闭帧计数器（LCD\_CTRL 的 FCCTRL=0），则每帧显示完后均会闪烁。

### 5.9.7 LCD工作模式

#### 5.9.7.1 占空比

LCD 控制器使用动态扫描的驱动方式进行 LCD 显示，每个 COM 的驱动有效时间与扫描周期的比值称为占空比。通过设定寄存器 LCD\_MODE 的 DUTY 位，可选择 1/4、1/6 和 1/8 占空比，对应的 COM 数量分别为 4、6 和 8。

#### 5.9.7.2 偏压

通常 LCD 的 SEG/COM 端驱动波形为模拟信号，端口波形电压相对于 LCD 输出最高电压的比例称为偏压。通过设定寄存器 LCD\_MODE 的 BIAS 位，可进行偏压选择。各种偏压条件下，



驱动信号波形所可能出现的电压值如下：

- ◆ 1/4 偏压时，驱动信号波形电压可为 0，1/4 VLCD，2/4 VLCD，3/4 VLCD 或 VLCD。
- ◆ 1/3 偏压时，驱动信号波形电压可为 0，1/3 VLCD，2/3 VLCD 或 VLCD。
- ◆ 1/2 偏压时，驱动信号波形电压可为 0，1/2 VLCD 或 VLCD。

用户只需设置所需的偏压类型，驱动信号波形中的各个电压由内部电路自动产生。

### 5.9.7.3 偏压电阻

通过寄存器 LCD\_MODE 的 RSEL 位，可选择 LCD 内部偏压电阻。偏压电阻越小，LCD 显示清晰度越高，但功耗越大。为了降低功耗同时保证显示效果，提供了 LCD 内部偏压电阻自动切换功能，用户可以根据实际 LCD 显示效果选择合适的偏压电阻。如果偏压电阻已设为最小，但 LCD 显示清晰度仍低时，则可以通过寄存器 LCD\_MODE 的 BUFEN 位使能输出缓冲器来提高显示清晰度，当使能输出缓冲器时，可将偏压电阻设为最大，以尽可能降低 LCD 显示功耗。

### 5.9.7.4 显示灰度

通过寄存器 LCD\_MODE 的 VLCD\_SEL 位，可控制 LCD 显示灰度，灰度电压越高，显示效果越亮，但功耗越大。用户可根据实际的 LCD 显示效果和功耗需求，设置合适的显示灰度。

### 5.9.7.5 波形类型

支持 A 和 B 两种 LCD 驱动波形类型，可通过寄存器 LCD\_MODE 的 TYPE 位进行选择。

## 5.9.8 COM/SEG 端口说明

LCD 输出端口包括：COM[3:0]、SEG[43:4]、MIX[3:0]。

在使用 COMx/MIXx/SEGx 端口作为 LCD 驱动信号输出时，需先设置对应的 GPIO 管脚为数字输入/输出禁止。

SEG[43:4] 端口所对应的数据，可通过寄存器 LCD\_RAMx 的 SEG43~SEG4 设置。MIX[3:0] 端口会根据占空比 1/4、1/6 或 1/8 的配置，来决定是用作 COM 还是 SEG 功能，可通过寄存器 LCD\_MODE 的 DUTY 位设置占空比。

- 占空比为 1/4 时，MIX[3:0] 端口为 SEG 功能，MIX3~MIX0 分别对应为 SEG3~SEG0。此时 LCD 可以提供 4 x 44 的 COM/SEG 组合。
- 占空比为 1/6 时，MIX[3:0] 端口为 COM 及 SEG 功能，其中 MIX1~MIX0 分别对应为 COM5~COM4，MIX3~MIX2 分别对应为 SEG3~SEG2。此时 LCD 可以提供 6x42 的 COM/SEG 组合。
- 占空比为 1/8 时，MIX[3:0] 端口为 COM 功能，MIX3~MIX0 分别对应为 COM7~COM4。此时 LCD 可以提供 8 x 40 的 COM/SEG 组合。

SEG4~SEG43 端口以及用作 SEG 功能的 MIX0~MIX3 端口，需要通过寄存器 LCD\_SEGOE1

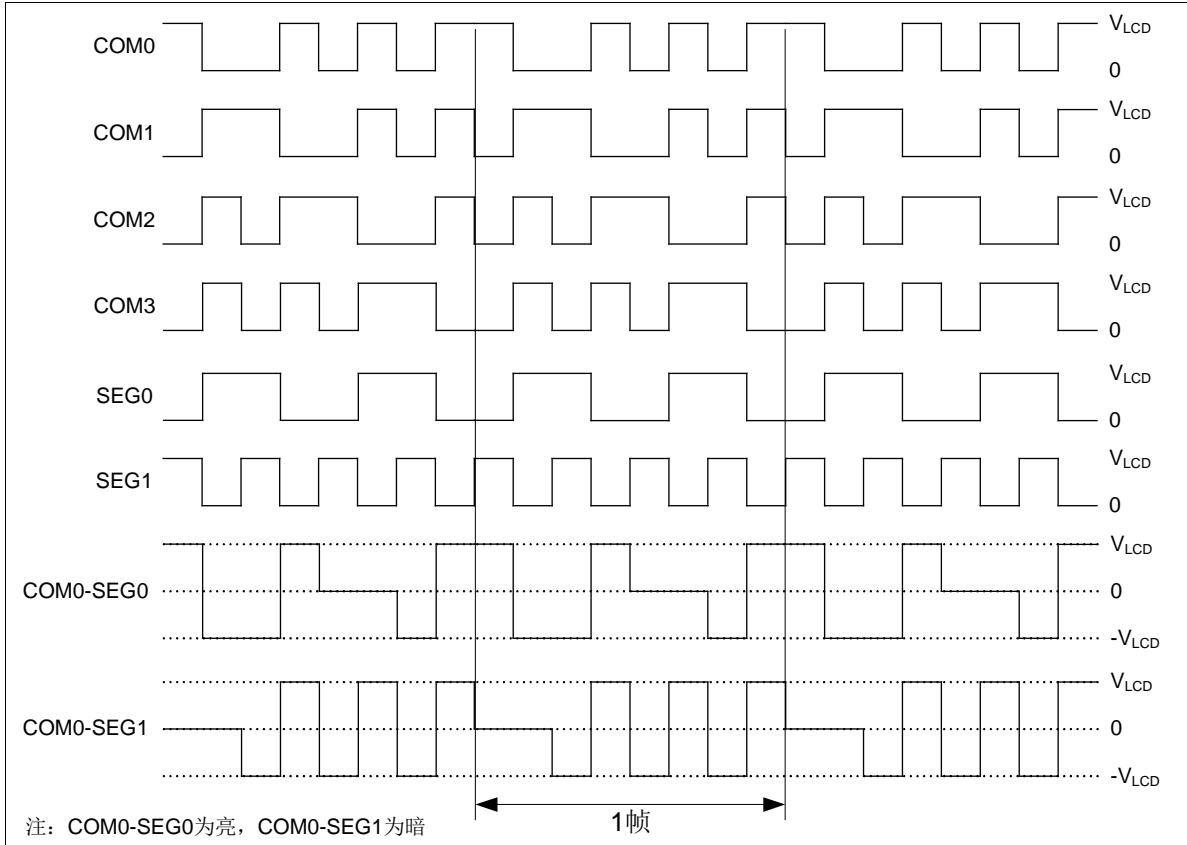
和 LCD\_SEGOE0 来设置为输出使能，才能输出驱动波形。

COM0-COM3 端口以及用作 COM 功能的 MIX0~MIX3 端口，在 LCD 使能时会自动开启输出，通过寄存器 LCD\_CTRL 的 LCDEN 位可使能 LCD。

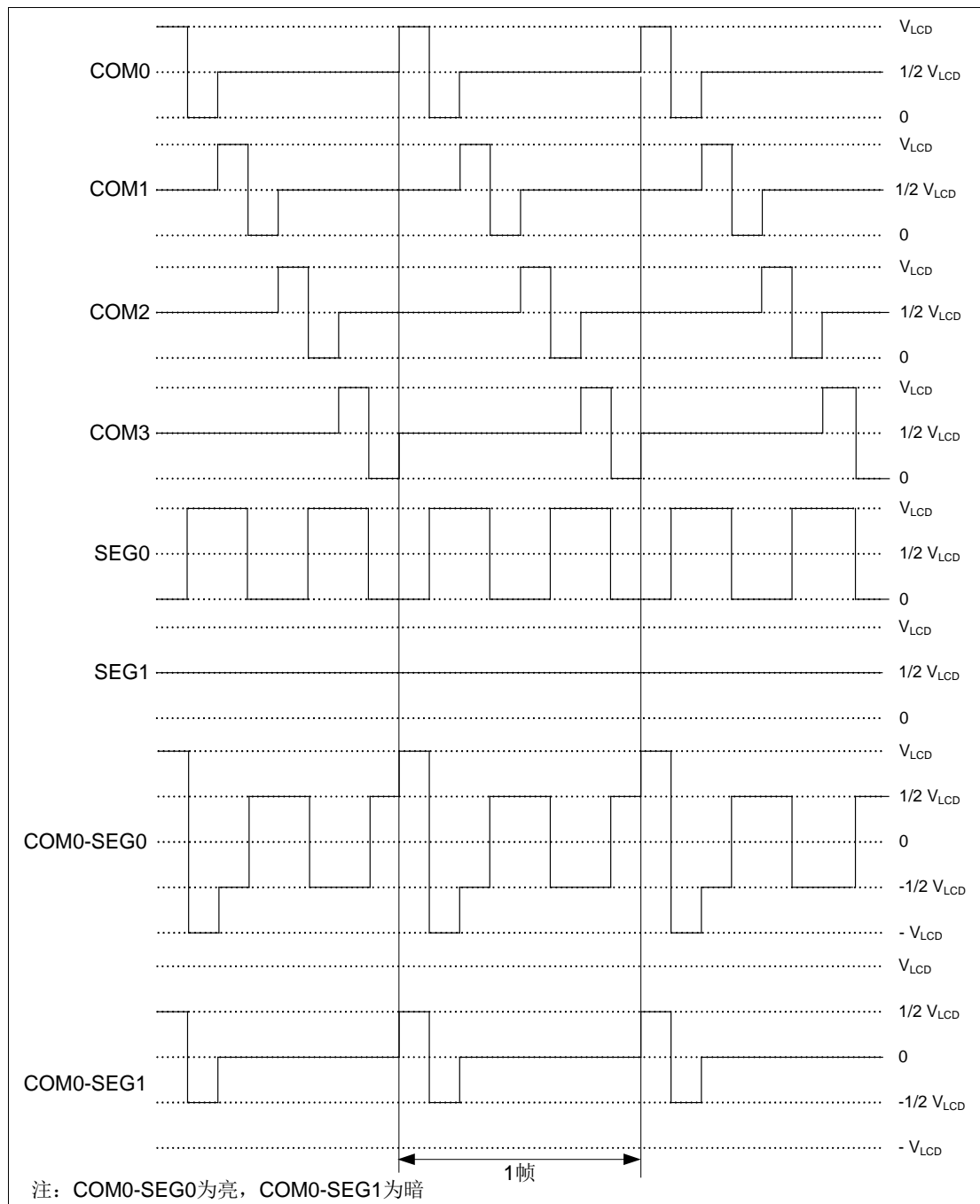
### 5.9.9 LCD输出波形

以下对不同占空比、不同偏压时的 LCD 驱动波形图进行示例说明。

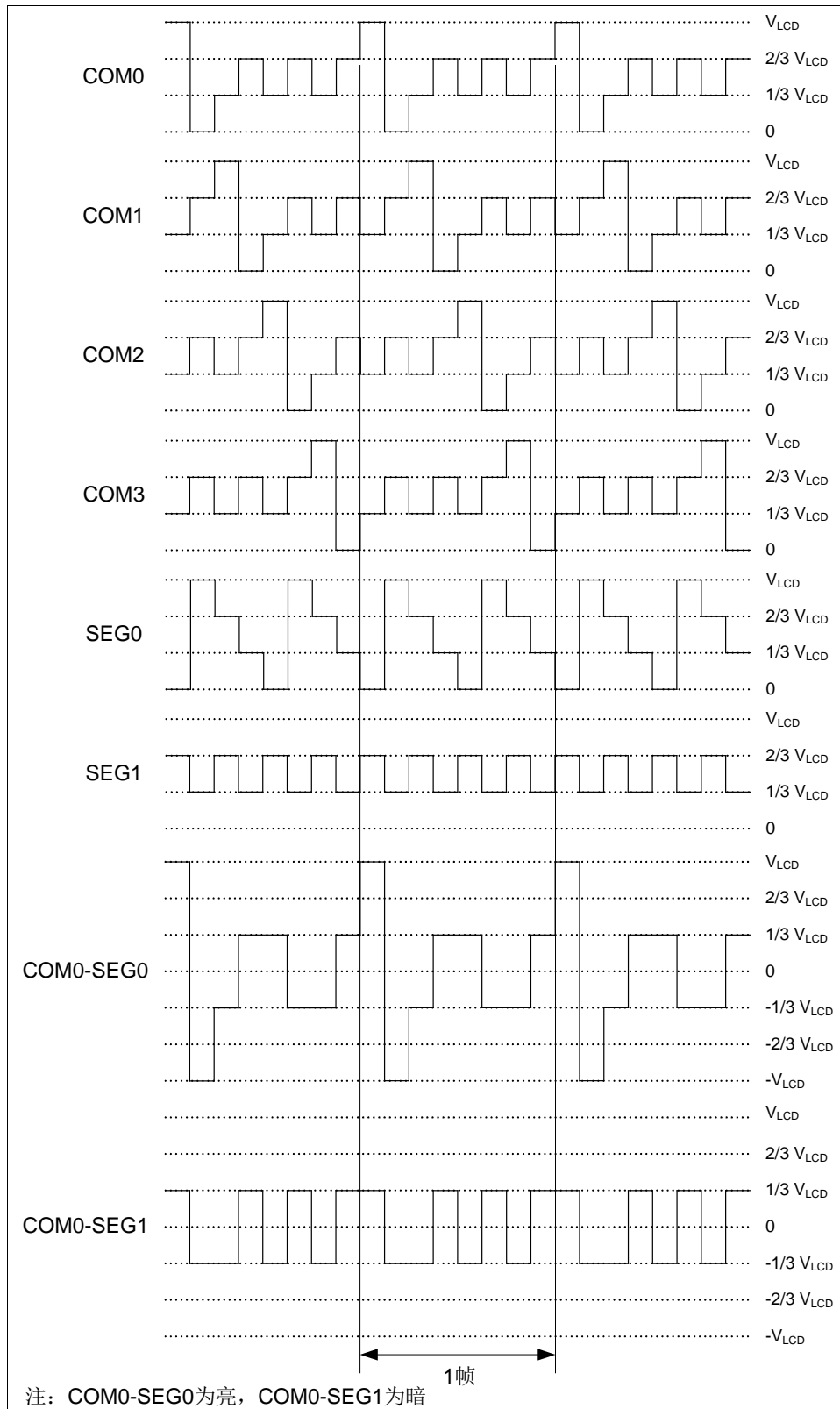
◆ 1/4 占空比、静态驱动时的 A 型波形图



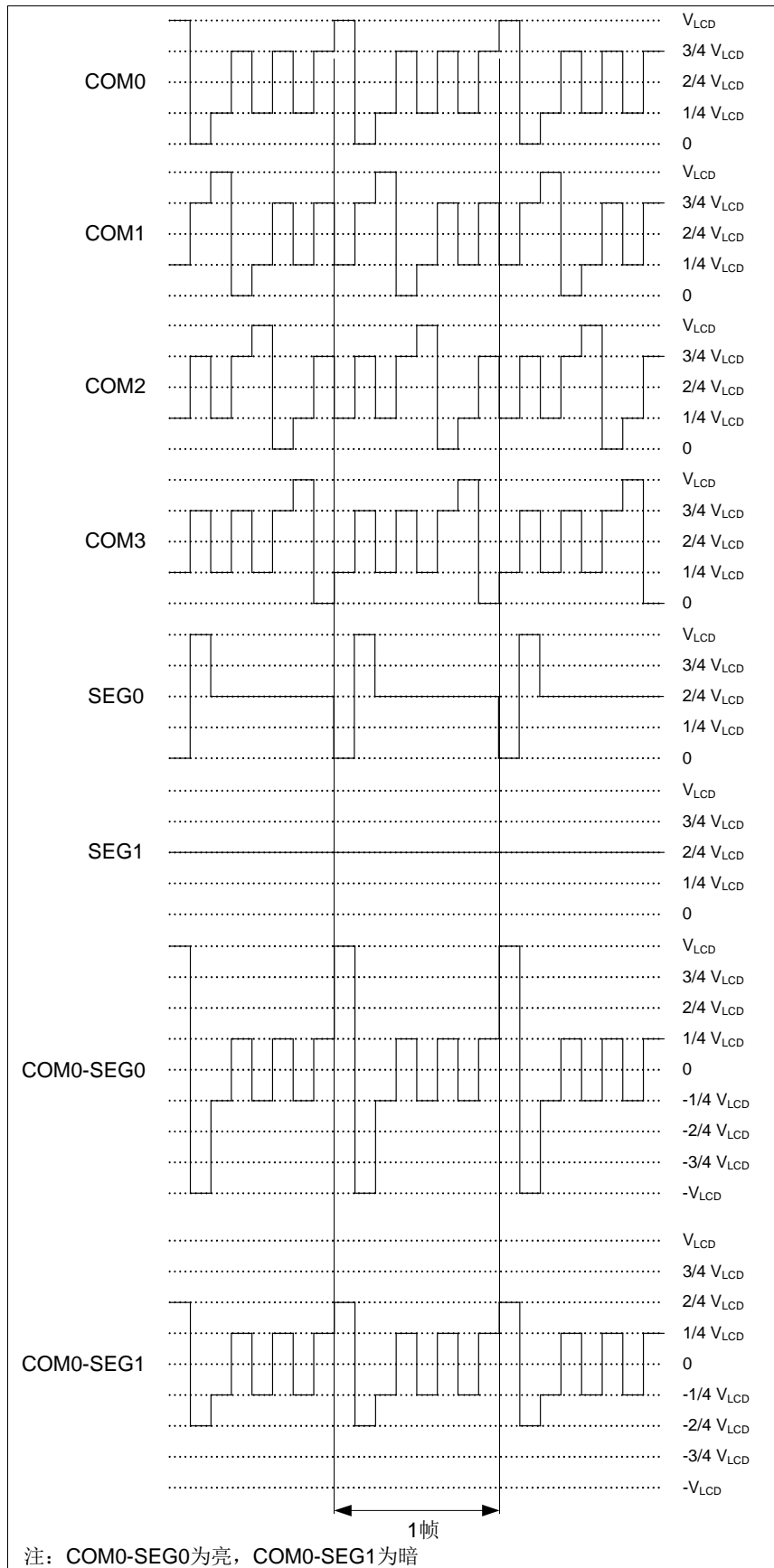
◆ 1/4 占空比、1/2 偏压驱动时的 A 型波形图



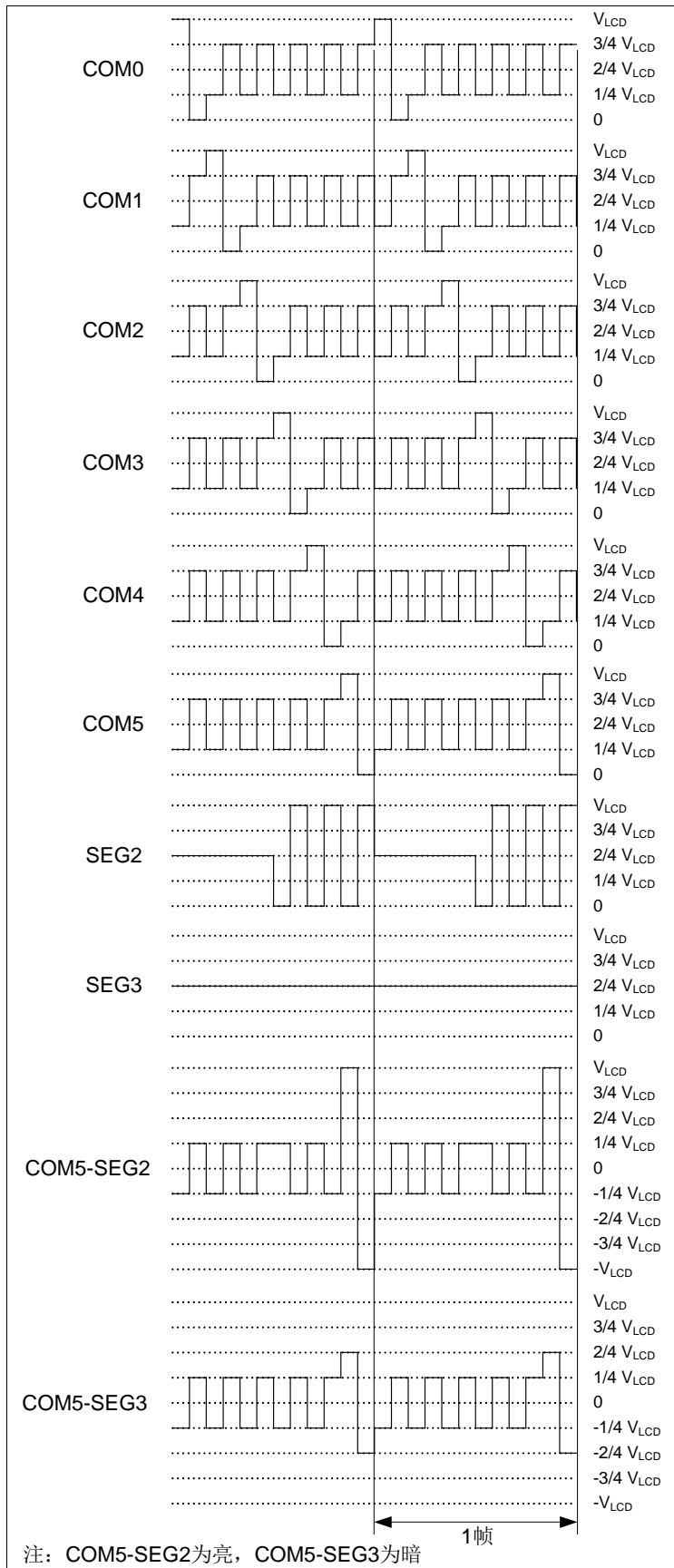
◆ 1/4 占空比、1/3 偏压驱动时的 A 型波形图



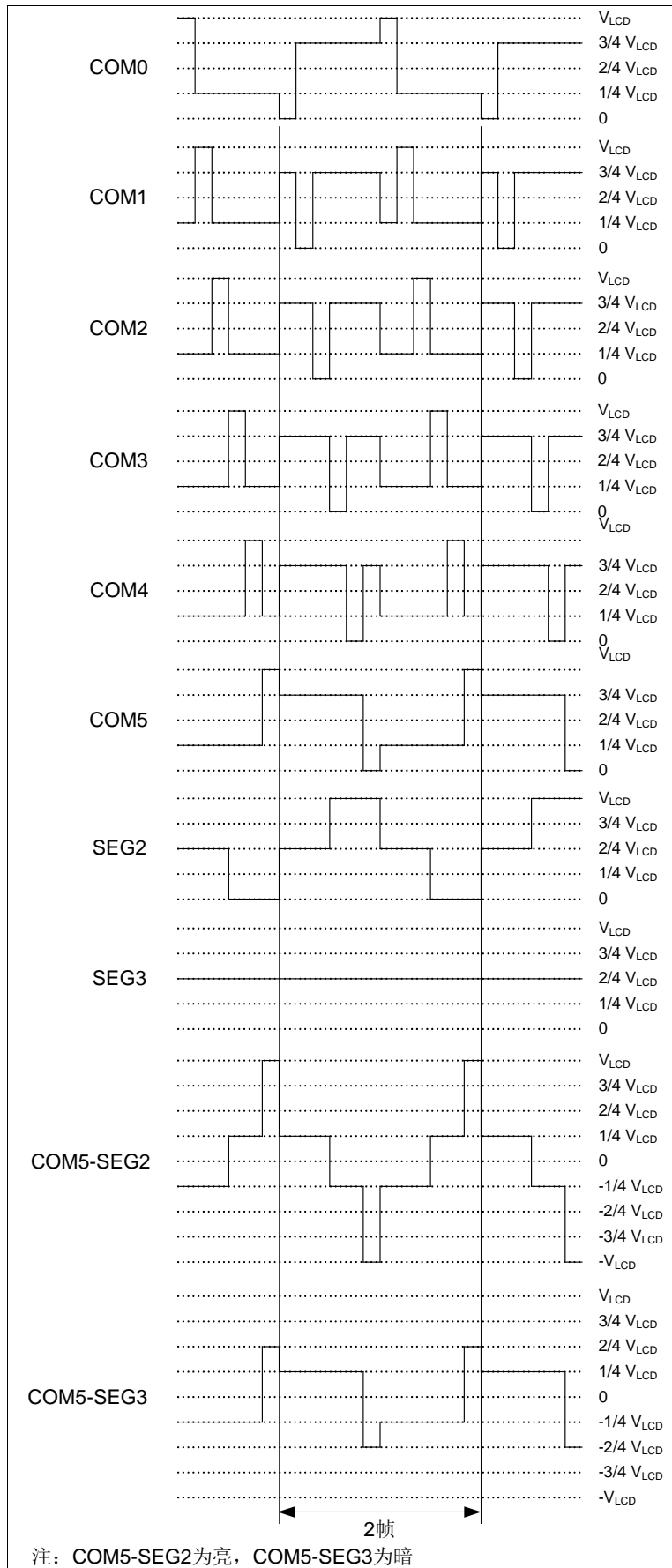
◆ 1/4 占空比、1/4 偏压驱动时的 A 型波形图



◆ 1/6 占空比、1/4 偏压驱动时的 A 型波形图



◆ 1/6 占空比、1/4 偏压驱动时的 B 型波形图



### 5.9.10 LCD基本设置

配置 LCD 驱动模块的操作步骤如下：

- ◆ 初始设置
  - 设置寄存器 SCU\_LCDCLK，对 LCDCLK 工作时钟进行配置。
- ◆ LCD 像素数值以及模式设置
  - 设置 COM、SEG 对应管脚的复用功能，开启 SEG 通道输出。
  - 将初始像素数据写入 LCD\_RAMx。
  - 设置寄存器 LCD\_MODE 的 DUTY、TYPE 以及 BIAS 位，对 COM 数和 LCD 驱动波形进行配置。
- ◆ LCD 帧频率设置
  - 设置寄存器 LCD\_CTRL 的 FRDIV 位，对帧时钟频率 LCD\_PHY\_CLK 进行配置。
  - 设置寄存器 LCD\_CTRL 的 UPDCTRL 位，对帧更新模式进行配置。
- ◆ LCD 开启
  - 设置寄存器 LCD\_CTRL 的 LCDEN，开启 LCD。
- ◆ LCD 显示更新
  - 更新显示数据，重新设置寄存器 LCD\_RAMx。
  - 设置寄存器 LCD\_CTRL 的 WSBLINK 位，更改闪烁模式。
  - 设置寄存器 LCD\_CTRL 的 SWUPDATE 位，下一帧完成更新（UPDCTRL=0）。

LCD 支持在休眠模式下运行，在 LCD 使能（LCDEN=1）后，经过 FRDIV+4 个 LCD\_CLK 时钟周期，对 LCD 的工作模式配置生效，然后可以设置芯片进入休眠模式，LCD 仍保持运行状态。



## 5.9.11 特殊功能寄存器

### 5.9.11.1 LCD模式控制寄存器 (LCD\_MODE)

LCD 模式控制寄存器 (LCD_MODE)															
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													QTSEL		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RSEL		VLCD_SEL				保留	BUFEN	TYPE	BIAS		DUTY			

—	bit 31-19	—	—
QTSEL	bit 18-16	R/W	<b>LCD快速充电时间选择位</b> 000: $1 \times t_{LCD\_CLK}$ 001: $2 \times t_{LCD\_CLK}$ 010: $3 \times t_{LCD\_CLK}$ 011: $4 \times t_{LCD\_CLK}$ 100: $5 \times t_{LCD\_CLK}$ 101: $6 \times t_{LCD\_CLK}$ 110: $7 \times t_{LCD\_CLK}$ 111: $8 \times t_{LCD\_CLK}$
—	bit 15	—	—
RSEL	bit 14-12	R/W	<b>LCD偏压电阻选择位</b> 000: 225K $\Omega$ 001: 900K $\Omega$ 010: 60K $\Omega$ 011: 60K $\Omega$ 100: 225K $\Omega$ 与 60K $\Omega$ 自动切换 101: 900K $\Omega$ 与 60K $\Omega$ 自动切换 11x: 保留 注: 只有在LCDEN=0时才能对该位进行设置。
VLCD_SEL	bit 11-8	R/W	<b>VLCD电压选择位</b> 0000 : VLCD=VDD/2 0001 : VLCD=VDD*(16/30) 0010 : VLCD=VDD*(17/30) 0011 : VLCD=VDD*(18/30) 0100 : VLCD=VDD*(19/30) 0101 : VLCD=VDD*(20/30) 0110 : VLCD=VDD*(21/30)

			<p>0111 : VLCD=VDD*(22/30)          1000 : VLCD=VDD*(23/30)          1001 : VLCD=VDD*(24/30)          1010 : VLCD=VDD*(25/30)          1011 : VLCD=VDD*(26/30)          1100 : VLCD=VDD*(27/30)          1101 : VLCD=VDD*(28/30)          1110 : VLCD=VDD*(29/30)          1111 : VLCD=VDD</p> <p>注：只有在LCDEN=0时才能对该位进行设置。</p>
—	bit 7-6	—	—
BUFEN	bit 5	R/W	<p><b>输出缓冲器使能位</b>          0: 关闭输出缓冲器          1: 使能输出缓冲器</p> <p>注：只有在LCDEN=0时才能对该位进行设置。</p>
TYPE	bit 4	R/W	<p><b>LCD驱动输出波形类型选择位</b>          0: A 类型（在每一个COM边界改变相位）          1: B 类型（在每一帧边界改变相位）</p> <p>注：只有在LCDEN=0时才能对该位进行设置。</p>
BIAS	bit 3-2	R/W	<p><b>LCD偏压选择位</b>          00: 静态          01: 1/2偏压          10: 1/3偏压          11: 1/4偏压</p> <p>注：只有在LCDEN=0时才能对该位进行设置。</p>
DUTY	bit 1-0	R/W	<p><b>COM占空比选择位</b>          00: 1/4 (COM0~COM3)          01: 1/6 (COM0~COM5)          1x: 1/8 (COM0~COM7)</p> <p>注：只有在LCDEN=0时才能对该位进行设置。</p>

注：当通过 BUFEN 位使能输出缓冲器来提高 LCD 显示清晰度时，可将偏压电阻设为最大（RSEL=001），以尽可能降低 LCD 显示功耗。

### 5.9.11.2 LCD控制寄存器 (LCD\_CTRL)

**LCD 控制寄存器 (LCD\_CTRL)**

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	
VLCD_RDY	保留										WSBLINK	FCVALUE				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
FCVALUE				FCCTRL	SWUPDATE	UPDCTRL			FRDIV				LCDEN			
				L	DATE											

VLCD_RDY	bit 31	R	<b>LCD电压稳定状态标志位</b> 0: LCD电压未稳定 1: LCD电压已稳定
—	bit 30-21	—	—
WSBLINK	bit 20-19	R/W	<b>闪烁模式选择位</b> 00: 关闭闪烁功能 01: 闪烁模式0, LCD显示在像素点设置数值和其反码数值之间切换 10: 闪烁模式1, LCD显示在像素点设置数值和全部熄灭之间切换 11: 闪烁模式2, 由寄存器LCD_BLINK1和LCD_BLINK2进行闪烁控制, 并且只控制SEGx端低8位对应的显示像素进行闪烁
FCVALUE	bit 18-12	R/W	<b>帧计数器数值</b> 计数显示帧的个数, 可用于设置闪烁周期, 以及LCD像素数据更新周期 闪烁频率等于 $f_{LCD\_PHY\_CLK}/(2*(FCVALUE+1))$ , 其中 $f_{LCD\_PHY\_CLK}$ 为帧时钟频率
FCCTRL	bit 11	R/W	<b>帧计数器开启位</b> 0: 关闭帧计数器 1: 开启帧计数器, 递减计数 (每显示完一帧就减1)
SWUPDATE	bit 10	C_W1	<b>软件生成更新事件</b> 该位由软件置1, 由硬件在更新完成后自动清零。 该位置1时, 产生更新事件, 并在下一个帧周期完成像素数据更新。 该位仅在UPDCTRL=0时有效。

UPDCTRL	bit 9-8	R/W	<p><b>更新控制选择位</b></p> <p>00: 当SWUPDATE为1时，下个帧周期更新</p> <p>01: 在每个帧周期更新</p> <p>10: 当帧计数器递减计数至0时的下个帧周期更新</p> <p>11: 保留</p>
FRDIV	bit 7-1	R/W	<p><b>LCD帧时钟频率<math>f_{\text{LCD\_PHY\_CLK}}</math>设置位</b></p> <p>帧频率由帧时钟频率<math>f_{\text{LCD\_PHY\_CLK}}</math>和COM占空比来设置，<math>f_{\text{LCD\_PHY\_CLK}}</math>频率等于<math>f_{\text{LCD\_CLK}}/(2*(\text{FRDIV}+1))</math>，其中<math>f_{\text{LCD\_CLK}}</math>为LCD控制器工作时钟，COM占空比由DUTY位设置</p>
LCDEN	bit 0	R/W	<p><b>LCD 控制器使能位</b></p> <p>0: 关闭LCD控制器</p> <p>1: 使能LCD控制器</p>

注：LCD 支持在休眠模式下运行，在 LCD 使能（LCDEN=1）后，经过 FRDIV+4 个 LCD\_CLK 时钟周期，对 LCD 的工作模式配置生效，然后可以设置芯片进入休眠模式，LCD 仍保持运行状态。

### 5.9.11.3 LCD闪烁控制寄存器 1 (LCD\_BLINK1)

**LCD 闪烁控制寄存器 1 (LCD\_BLINK1)**

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
COM3								COM2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COM1								COM0							

COM3	bit 31-24	R/W	<p><b>COM3闪烁选择位</b></p> <p>扫描选中COM3时，分别控制SEGx端低8位对应的显示像素进行闪烁，控制位为1时，对应的SEGx端像素闪烁，否则不闪</p> <p>注：此位仅在闪烁模式2时有效</p>
COM2	bit 23-16	R/W	<p><b>COM2闪烁选择位</b></p> <p>扫描选中COM2时，分别控制SEGx端低8位对应的显示像素进行闪烁，控制位为1时，对应的SEGx端像素闪烁，否则不闪</p> <p>注：此位仅在闪烁模式2时有效</p>
COM1	bit 15-8	R/W	<p><b>COM1闪烁选择位</b></p> <p>扫描选中COM1时，分别控制SEGx端低8位对应的显示像素进行闪烁，控制位为1时，对应的SEGx端像素闪烁，否则不闪</p> <p>注：此位仅在闪烁模式2时有效</p>
COM0	bit 7-0	R/W	<p><b>COM0闪烁选择位</b></p> <p>扫描选中COM0时，分别控制SEGx端低8位对应的显示像素进行闪烁，控制位为1时，对应的SEGx端像素闪烁，否则不闪</p> <p>注：此位仅在闪烁模式2时有效</p>

### 5.9.11.4 LCD闪烁控制寄存器 2 (LCD\_BLINK2)

#### LCD 闪烁控制寄存器 2 (LCD\_BLINK2)

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
COM7								COM6							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COM5								COM4							

COM7	bit31-24	R/W	<p><b>COM7闪烁选择位</b></p> <p>扫描选中COM7时, 分别控制SEGx端低8位对应的显示像素进行闪烁, 控制位为1时, 对应的SEGx端像素闪烁, 否则不闪</p> <p>注: 此位仅在闪烁模式2时有效</p>
COM6	bit23-16	R/W	<p><b>COM6闪烁选择位</b></p> <p>扫描选中COM6时, 分别控制SEGx端低8位对应的显示像素进行闪烁, 控制位为1时, 对应的SEGx端像素闪烁, 否则不闪</p> <p>注: 此位仅在闪烁模式2时有效</p>
COM5	bit 15-8	R/W	<p><b>COM5闪烁选择位</b></p> <p>扫描选中COM5时, 分别控制SEGx端低8位对应的显示像素进行闪烁, 控制位为1时, 对应的SEGx端像素闪烁, 否则不闪</p> <p>注: 此位仅在闪烁模式2时有效</p>
COM4	bit 7-0	R/W	<p><b>COM4闪烁选择位</b></p> <p>扫描选中COM4时, 分别控制SEGx端低8位对应的显示像素进行闪烁, 控制位为1时, 对应的SEGx端像素闪烁, 否则不闪</p> <p>注: 此位仅在闪烁模式2时有效</p>

### 5.9.11.5 LCD更新状态寄存器 (LCD\_RIF)

#### LCD 更新状态寄存器 (LCD\_RIF)

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														UPD	
保留														ATE	

—	bit 31-1	—	—
UPDATE	bit 0	R	<b>更新完成状态标志位</b> 当像素显示数据更新完成时，此标志由硬件自动置 1。写入 ICR 寄存器可清除该状态标志

### 5.9.11.6 LCD状态标志清除寄存器 (LCD\_ICR)

**LCD 状态标志清除寄存器 (LCD\_ICR)**

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														UPDAT	
														E	

—	bit 31-1	—	—
UPDATE	bit 0	C_W1	<b>更新完成状态标志清除</b> 0: 写入 0 无效 1: 清除更新完成状态标志 RIF

### 5.9.11.7 LCD运行状态寄存器 (LCD\_BUSY)

**LCD 运行状态寄存器 (LCD\_BUSY)**

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														UPDAT	
														E	

—	bit 31-1	—	—
UPDATE	bit 0	R	<b>更新忙状态标志位</b> 当像素显示数据正在更新时由硬件置1, 更新完成后自动清零



### 5.9.11.8 LCD显示数据寄存器 0 (LCD\_RAM0)

**LCD 显示数据寄存器 0 (LCD\_RAM0)**

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
LCD_RAM0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCD_RAM0															

LCD_RAM0	bit 31-0	R/W	<b>COM0 SEG[31:0]数值</b> 选择扫描至COM0时，输出SEG31~SEG0的状态 0: 不点亮（透明） 1: 点亮（不透明）
----------	----------	-----	---

### 5.9.11.9 LCD显示数据寄存器 1 (LCD\_RAM1)

**LCD 显示数据寄存器 1 (LCD\_RAM1)**

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				LCD_RAM1											

—	bit 31-12	—	—
LCD_RAM1	bit 11-0	R/W	<b>COM0 SEG[43:32]数值</b> 选择扫描至COM0时，输出SEG43~SEG32的状态 0: 不点亮（透明） 1: 点亮（不透明）

### 5.9.11.10 LCD显示数据寄存器 2 (LCD\_RAM2)

#### LCD 显示数据寄存器 2 (LCD\_RAM2)

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
LCD_RAM2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCD_RAM2															

LCD_RAM2	bit 31-0	R/W	<b>COM1 SEG[31:0]数值</b> 选择扫描至COM1时, 输出SEG31~SEG0的状态 0: 不点亮 (透明) 1: 点亮 (不透明)
----------	----------	-----	--

### 5.9.11.11 LCD显示数据寄存器 3 (LCD\_RAM3)

#### LCD 显示数据寄存器 3 (LCD\_RAM3)

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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				LCD_RAM3											

—	bit 31-12	—	—
LCD_RAM3	bit 11-0	R/W	<b>COM1 SEG[43:32]数值</b> 选择扫描至COM1时, 输出SEG43~SEG32的状态 0: 不点亮 (透明) 1: 点亮 (不透明)

### 5.9.11.12 LCD显示数据寄存器 4 (LCD\_RAM4)

#### LCD 显示数据寄存器 4 (LCD\_RAM4)

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
LCD_RAM4															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCD_RAM4															

LCD_RAM4	bit 31-0	R/W	<b>COM2 SEG[31:0]数值</b> 选择扫描至COM2时，输出SEG31~SEG0的状态 0: 不点亮（透明） 1: 点亮（不透明）
----------	----------	-----	---

### 5.9.11.13 LCD显示数据寄存器 5 (LCD\_RAM5)

#### LCD 显示数据寄存器 5 (LCD\_RAM5)

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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				LCD_RAM5											

—	bit 31-12	—	—
LCD_RAM5	bit 11-0	R/W	<b>COM2 SEG[43:32]数值</b> 选择扫描至COM2时，输出SEG43~SEG32的状态 0: 不点亮（透明） 1: 点亮（不透明）

### 5.9.11.14 LCD显示数据寄存器 6 (LCD\_RAM6)

**LCD 显示数据寄存器 6 (LCD\_RAM6)**

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
LCD_RAM6															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCD_RAM6															

LCD_RAM6	bit 31-0	R/W	<b>COM3 SEG[31:0]数值</b> 选择扫描至COM3时, 输出SEG31~SEG0的状态 0: 不点亮 (透明) 1: 点亮 (不透明)
----------	----------	-----	--

### 5.9.11.15 LCD显示数据寄存器 7 (LCD\_RAM7)

**LCD 显示数据寄存器 7 (LCD\_RAM7)**

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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				LCD_RAM7											

—	bit 31-12	—	—
LCD_RAM7	bit 11-0	R/W	<b>COM3 SEG[43:32]数值</b> 选择扫描至COM3时, 输出SEG43~SEG32的状态 0: 不点亮 (透明) 1: 点亮 (不透明)

### 5.9.11.16 LCD显示数据寄存器 8 (LCD\_RAM8)

#### LCD 显示数据寄存器 8 (LCD\_RAM8)

偏移地址: 50<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
LCD_RAM8															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCD_RAM8															

LCD_RAM8	bit 31-0	R/W	<b>COM4 SEG[31:0]数值</b> 选择扫描至COM4时, 输出SEG31~SEG0的状态 0: 不点亮 (透明) 1: 点亮 (不透明)
----------	----------	-----	--

### 5.9.11.17 LCD显示数据寄存器 9 (LCD\_RAM9)

#### LCD 显示数据寄存器 9 (LCD\_RAM9)

偏移地址: 54<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				LCD_RAM9											

—	bit 31-12	—	—
LCD_RAM9	bit 11-0	R/W	<b>COM4 SEG[43:32]数值</b> 选择扫描至COM4时, 输出SEG43~SEG32的状态 0: 不点亮 (透明) 1: 点亮 (不透明)

### 5.9.11.18 LCD显示数据寄存器 10 (LCD\_RAM10)

**LCD 显示数据寄存器 10 (LCD\_RAM10)**

偏移地址: 58<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
LCD_RAM10															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCD_RAM10															

LCD_RAM10	bit 31-0	R/W	<b>COM5 SEG[31:0]数值</b> 选择扫描至COM5时, 输出SEG31~SEG0的状态 0: 不点亮 (透明) 1: 点亮 (不透明)
-----------	----------	-----	--

### 5.9.11.19 LCD显示数据寄存器 11 (LCD\_RAM11)

**LCD 显示数据寄存器 11 (LCD\_RAM11)**

偏移地址: 5C<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				LCD_RAM11											

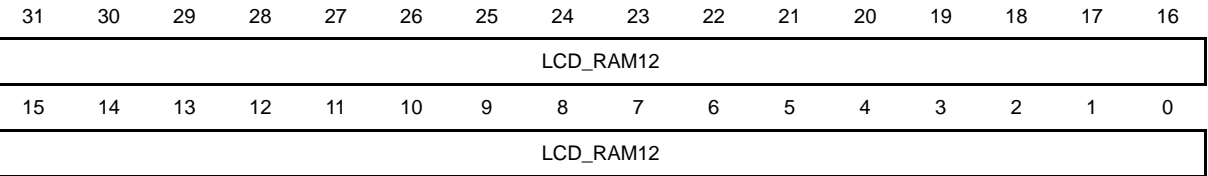
—	bit 31-12	—	—
LCD_RAM11	bit 11-0	R/W	<b>COM5 SEG[43:32]数值</b> 选择扫描至COM5时, 输出SEG43~SEG32的状态 0: 不点亮 (透明) 1: 点亮 (不透明)

### 5.9.11.20 LCD显示数据寄存器 12 (LCD\_RAM12)

**LCD 显示数据寄存器 12 (LCD\_RAM12)**

偏移地址: 60<sub>H</sub>

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



LCD_RAM12	bit 31-0	R/W	<b>COM6 SEG[31:0]数值</b> 选择扫描至COM6时, 输出SEG31~SEG0的状态 0: 不点亮 (透明) 1: 点亮 (不透明)
-----------	----------	-----	--

### 5.9.11.21 LCD显示数据寄存器 13 (LCD\_RAM13)

**LCD 显示数据寄存器 13 (LCD\_RAM13)**

偏移地址: 64<sub>H</sub>

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



—	bit 31-12	—	—
LCD_RAM13	bit 11-0	R/W	<b>COM6 SEG[43:32]数值</b> 选择扫描至COM6时, 输出SEG43~SEG32的状态 0: 不点亮 (透明) 1: 点亮 (不透明)

### 5.9.11.22 LCD显示数据寄存器 14 (LCD\_RAM14)

#### LCD 显示数据寄存器 14 (LCD\_RAM14)

偏移地址: 68<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

LCD\_RAM14

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

LCD\_RAM14

LCD_RAM14	bit 31-0	R/W	<b>COM7 SEG[31:0]数值</b> 选择扫描至COM7时, 输出SEG31~SEG0的状态 0: 不点亮 (透明) 1: 点亮 (不透明)
-----------	----------	-----	--

### 5.9.11.23 LCD显示数据寄存器 15 (LCD\_RAM15)

#### LCD 显示数据寄存器 15 (LCD\_RAM15)

偏移地址: 6C<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

LCD\_RAM15

—	bit 31-12	—	—
LCD_RAM15	bit 11-0	R/W	<b>COM7 SEG[43:32]数值</b> 选择扫描至COM7时, 输出SEG43~SEG32的状态 0: 不点亮 (透明) 1: 点亮 (不透明)



### 5.9.11.24 LCD SEG 输出使能寄存器 0 (LCD\_SEGOE0)

LCD SEG 输出使能寄存器 0 (LCD\_SEGOE0)

偏移地址: 70<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
SEGOE0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEGOE0															

SEGOE0	bit 31-0	R/W	<b>SEG[31:0] 输出使能位</b> SEG31~SEG0 输出 0: 禁止 1: 使能
--------	----------	-----	---

### 5.9.11.25 LCD SEG 输出使能寄存器 1 (LCD\_SEGOE1)

LCD SEG 输出使能寄存器 1 (LCD\_SEGOE1)

偏移地址: 74<sub>H</sub>

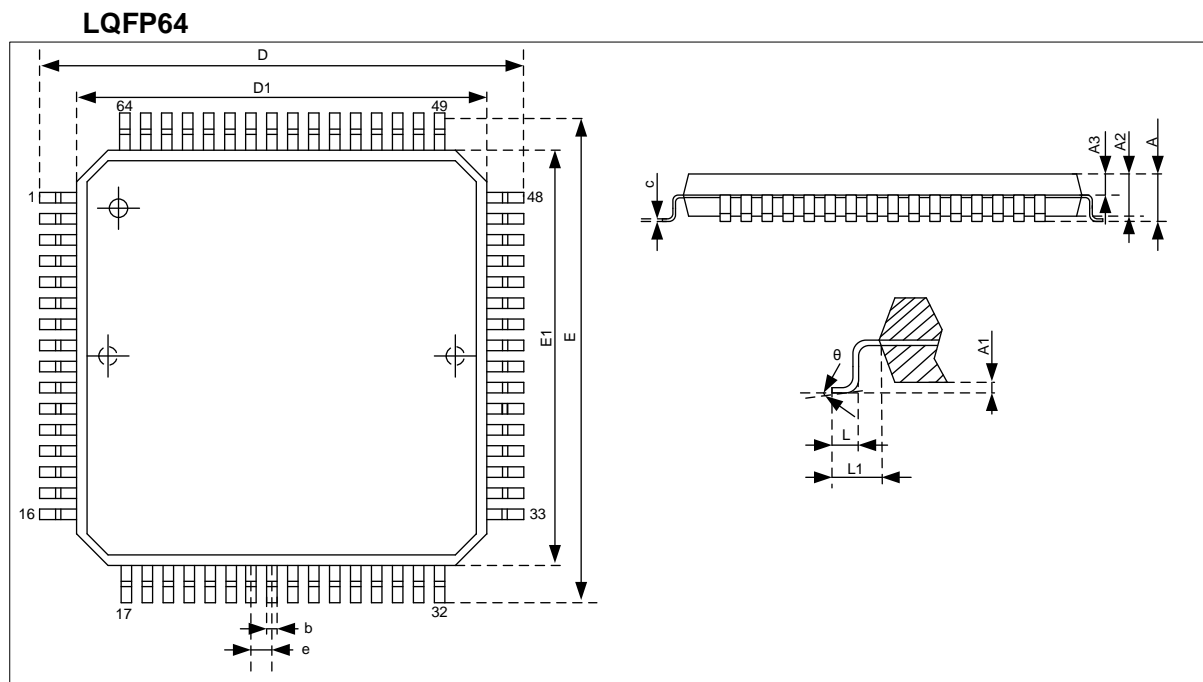
复位值: 00000000\_00000000\_00000000\_00000000<sub>B</sub>

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				SEGOE1											

—	bit 31-12	—	—
SEGOE1	bit 11-0	R/W	<b>SEG[43:32] 输出使能位</b> SEG43~SEG32 输出 0: 禁止 1: 使能

## 第6章 芯片封装外观尺寸图

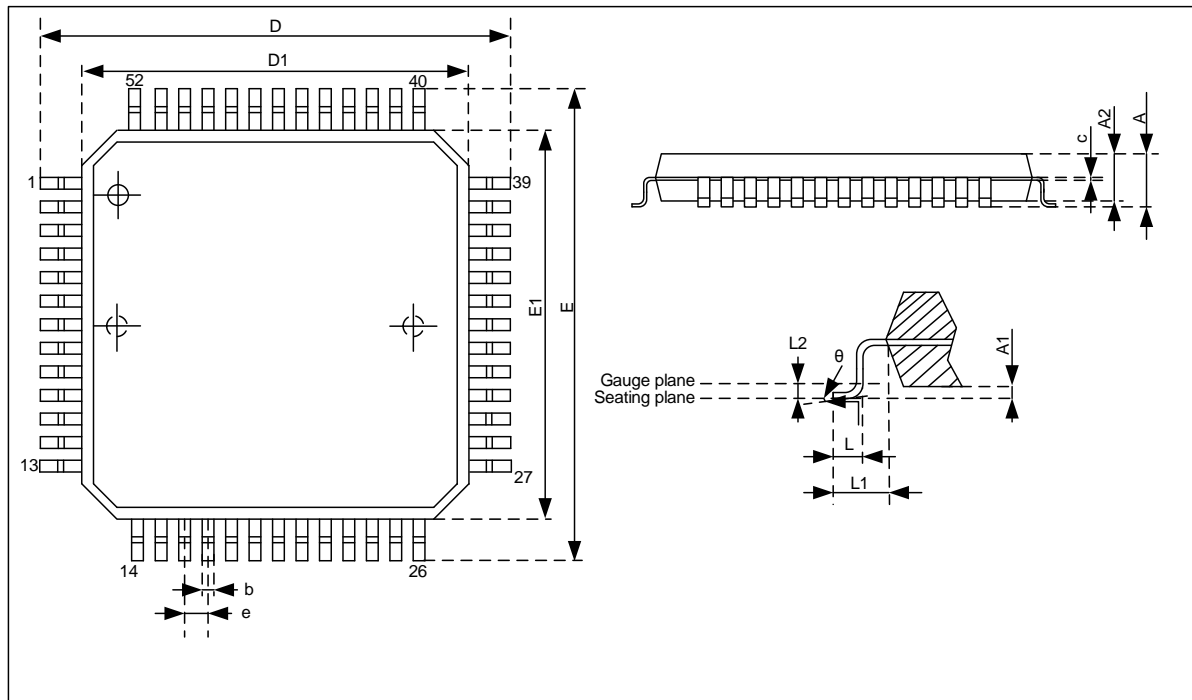
### 6.1 64-PIN封装外观尺寸图



标号	公制 (mm)		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.25
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.16	—	0.27
c	0.13	—	0.18
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e	0.50BSC		
L	0.45	—	0.75
L1	1.00BSC		
$\theta$	0	—	7°

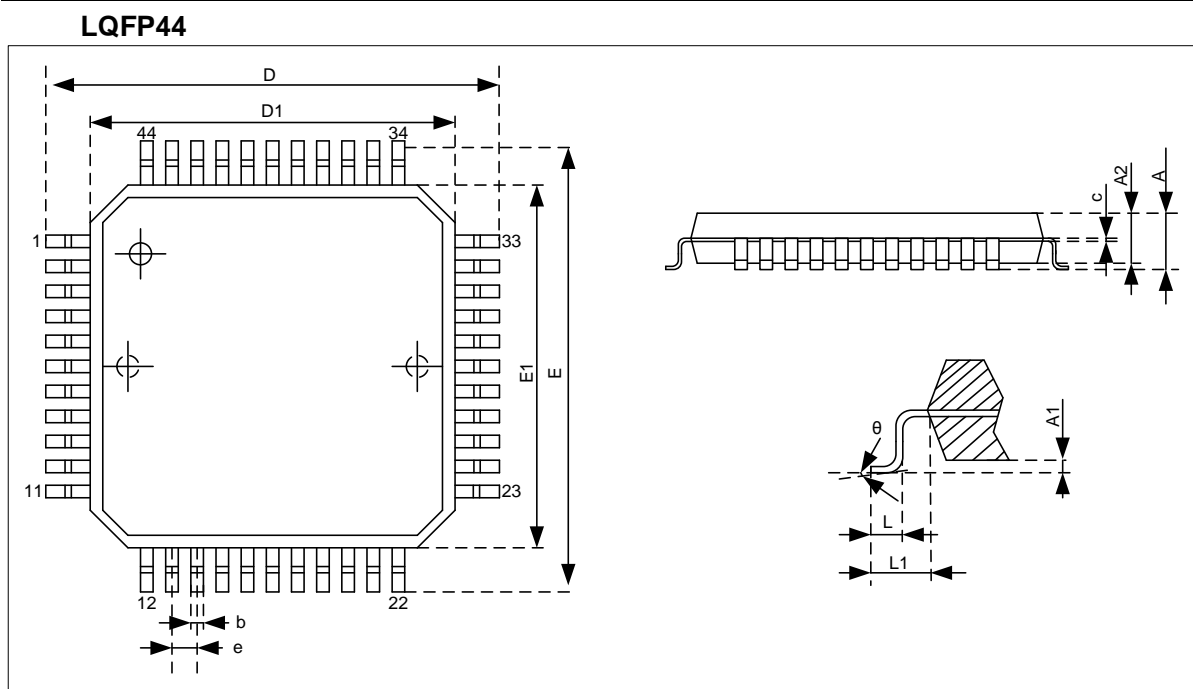
## 6.2 52-PIN封装外观尺寸图

LQFP52



标号	公制 (mm)		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.20
A2	1.30	1.40	1.50
c	0.13	—	0.18
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
b	0.28	—	0.37
e	0.55	0.65	0.75
L	0.45	0.60	0.75
L1	1.00REF		
$\theta$	0°	—	7°

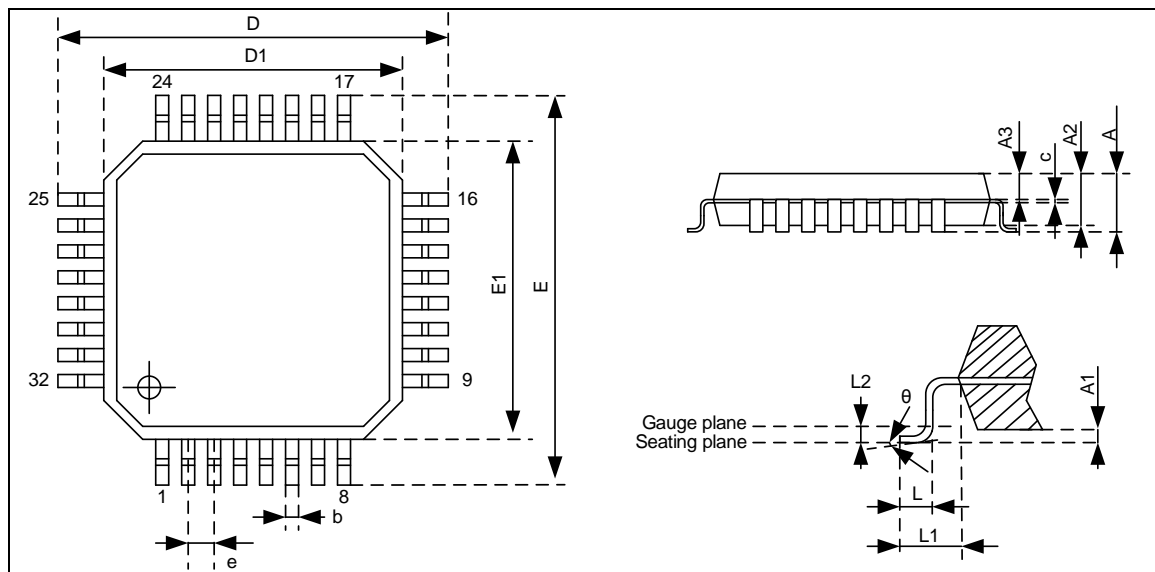
### 6.3 44-PIN封装外观尺寸图



标号	公制 (mm)		
	MIN	NOM	MAX
A	-	-	1.60
A1	0.05	-	0.20
A2	1.35	1.40	1.45
c	0.13	-	0.18
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
b	0.28	-	0.44
e	0.80BSC		
L	0.45	-	0.75
L1	1.00BSC		
$\theta$	0°	-	7°

## 6.4 32-pin封装外观尺寸图

LQFP32



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
c	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	—
θ	0°	3.5°	7°

## 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，并由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 MOV<sub>S</sub>，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
EORS	{Rd,}Rn,Rm	按位逻辑异或，Rd = Rn^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

助记符	操作数	描述	影响标志位
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：

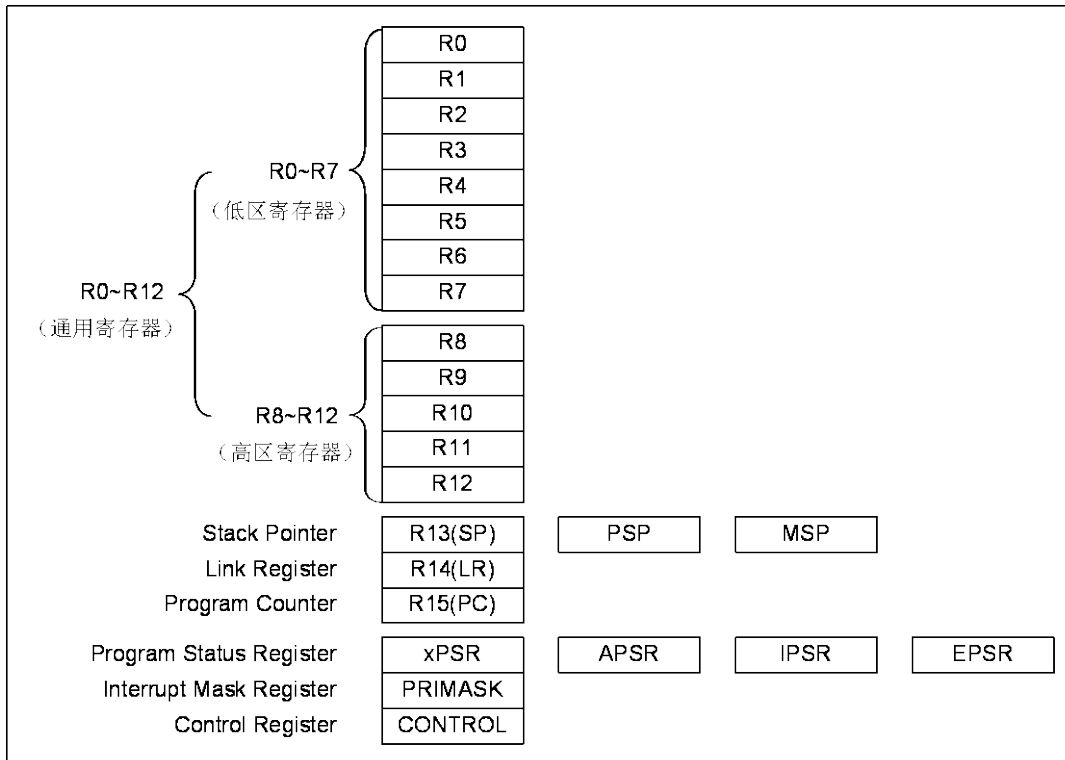


图 6-1 内核寄存器

### 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：



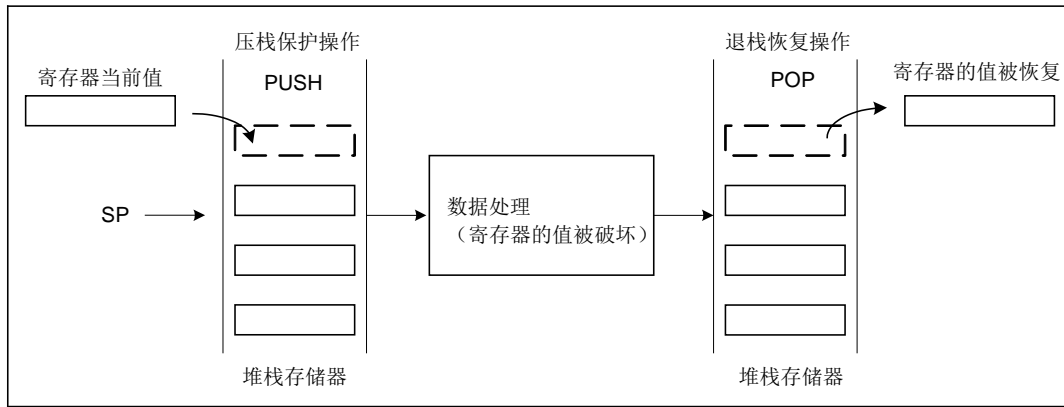


图 6-2 堆栈操作示意图

### 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 IAP\_STAR 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（ $\text{bit}\langle 31 \rangle = 0$ ）时溢出，则  $V=1$ ，否则  $V=0$ ；

两个正数相加，结果为负数（ $\text{bit}\langle 31 \rangle = 1$ ）时溢出，则  $V=1$ ，否则  $V=0$ ；

负数减去正数，结果为正数（ $\text{bit}\langle 31 \rangle = 0$ ）时溢出，则  $V=1$ ，否则  $V=0$ ；

正数减去负数，结果为负数（ $\text{bit}\langle 31 \rangle = 1$ ）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  $\text{IPSR}\langle 5:0 \rangle = 0$ ，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  $\text{PM}=1$  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  $\text{PM}=0$  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  $\text{SPSEL}=0$  时，选择 MSP (SP\_main) 作为当前堆栈指针；当  $\text{SPSEL}=1$  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， $\text{SPSEL}=0$ ，且只读，不可写；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。

## 附录2 MCU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 ~ 6.25	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°C	100	mA
VSS 管脚最大输出电流	I <sub>MAXVSS</sub>	VDD=5.0V, 25°C	150	mA
芯片存储温度	T <sub>STG</sub>	—	-55 ~ 150	°C
芯片最高结温	T <sub>JUNC</sub>	—	150	°C

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	°C
芯片工作结温	T <sub>JUNC</sub>	—	-40	105	°C
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

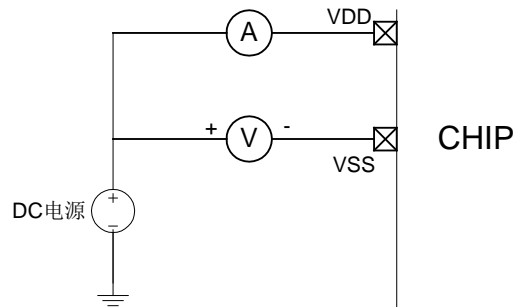
参数	符号	工作温度	VDD 电压	备注
ADC 工作电压	V <sub>ADC1</sub>	-40 ~ 85°C	2.5~5.5V	参考电压为 VDD 或外部 AVREFP
	V <sub>ADC2</sub>	-40 ~ 85°C	2.8~5.5V	参考电压为内部 VREFP

◆ 芯片上电和下电工作条件表 (-40 ~ 85°C)

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>	20	—	us/V
VDD 下降速率		50	—	us/V

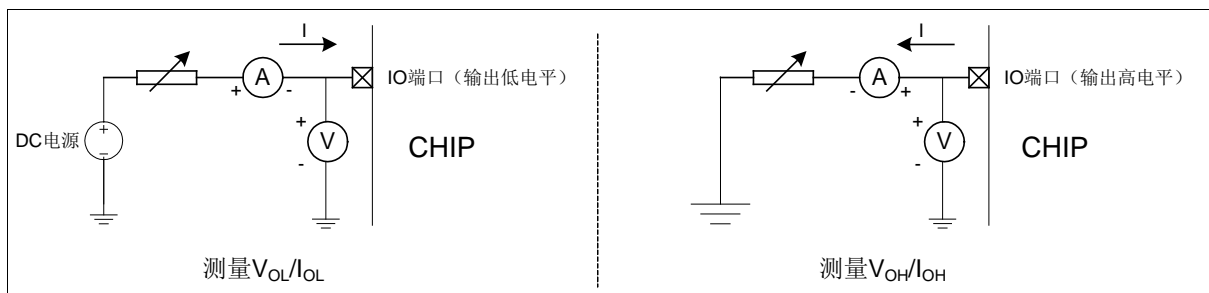
### 附录2.1.2 芯片特性参数测量方法

#### ◆ 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℃
深度睡眠模式下芯片电流	I <sub>PD1</sub>	—	3.5	—	μA	25℃, VDD = 5V, IWDT 不使能, 所有 I/O 端口输出固定电平, 无负载。(寄存器 SCU_WAKEUPTIME 设置为推荐值)
浅睡眠模式下芯片电流	I <sub>PD2</sub>	—	2.2	—	mA	25℃, VDD = 5V, IWDT 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 <sub>PD3</sub>	—	6.0	—	mA	25℃, VDD = 5V, IWDT 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。
正常运行模式芯片电流	I <sub>OP1</sub>	—	1.5	—	mA	25℃, VDD = 5V, 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 使用内部 VREFP

参数	符号	最小值	典型值	最大值	单位	工作条件
						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	$I_{OP2}$	—	4.0	—	mA	25°C, VDD = 5V, 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	$I_{OP3}$	—	6.5	—	mA	25°C, VDD = 5V, 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	$I_{OP4}$	—	9.2	—	mA	25°C, VDD = 5V, 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	$I_{XTAL}$	—	1.2	—	mA	25°C, VDD = 5V
内部高速时钟 HRC 48MHz 电流	$I_{HRC1}$	—	1	—	mA	25°C, VDD = 5V
内部高速时钟 HRC 32MHz 电流	$I_{HRC2}$	—	0.7	—	mA	25°C, VDD = 5V
内部高速时钟 HRC 16MHz 电流	$I_{HRC3}$	—	0.6	—	mA	25°C, VDD = 5V
内部高速时钟 HRC 2MHz 电流	$I_{HRC4}$	—	0.5	—	mA	25°C, VDD = 5V
LVD 模块电流	$I_{LVD}$	—	0.3	—	μA	25°C, VDD = 5V
ADC 模块电流	$I_{ADC1}$	—	1.2	—	mA	25°C, VDD = 5V, ADC 转换时钟频率为 1MHz, 内部 VREFP 作为正向参考电压 (VREF_EN=1, CHOP_EN=1)
	$I_{ADC2}$	—	1.0	—	mA	25°C, 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	$I_{IWDT}$	—	0.3	—	μA	25°C, VDD = 5V, 计数时钟为 LRC
WWDT 模块电流	$I_{WWDT}$	—	0.3	—	μA	25°C, VDD = 5V, 计数时钟为 LRC

参数	符号	最小值	典型值	最大值	单位	工作条件
T16N 模块电流	$I_{T16N}$	—	0.3	—	mA	25°C, VDD = 5V, PWM 模式, 输出 200KHz 互补波形
T32N 模块电流	$I_{T32N}$	—	1	—	mA	25°C, VDD = 5V, PWM 模式, 输出 200KHz 互补波形
UART 模块电流	$I_{UART}$	—	0.7	—	mA	25°C, VDD = 5V, 通讯波特率为 115200bps
SPI 模块电流	$I_{SPI}$	—	1	—	mA	25°C, VDD = 5V, 主控模式, 通讯速率为 2MHz
I2C 模块电流	$I_{I2C}$	—	0.9	—	mA	25°C, VDD = 5V, 主控模式, 通讯速率为 400KHz
FLASH 编程电流	$I_{PROG}$	—	—	4	mA	-40~85°C, VDD = 5V
FLASH 擦除电流	$I_{ERAS}$	—	—	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°C ~ 85°C						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输入高电平 (TTL 输入)	$V_{IH1}$	0.6VDD	—	VDD	V	2.2V ≤ VDD ≤ 5.5V
I/O 端口输入低电平 (TTL 输入)	$V_{IL1}$	VSS	—	0.1VDD	V	
IO 端口施密特迟滞窗口 (TTL 输入)	$V_{HYST1}$	—	0.3	—	V	25°C, VDD=5.0V
I/O 端口输入高电平 (CMOS 输入)	$V_{IH2}$	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)	$V_{IH\_MRSTN}$	0.8VDD	—	VDD	V	2.2V ≤ VDD ≤ 5.5V
外部复位输入低电平 (CMOS 输入)	$V_{IL\_MRSTN}$	VSS	—	0.2VDD	V	2.2V ≤ VDD ≤ 5.5V
IO 端口施密特迟滞窗口 (CMOS 输入)	$V_{HYST2}$	—	1.3	—	V	25°C, VDD=5.0V
I/O 端口输入漏电流	$I_{IL}$	—	—	±1	μA	2.2V ≤ VDD ≤ 5.5V Vpin=VSS 或 VDD
主复位端口漏电流		—	—	5	μA	Vpin=VDD
I/O 端口弱上拉电阻	$R_{WPU}$	—	55	—	kΩ	25°C, VDD=5.0V Vpin = VSS
I/O 端口弱下拉电阻	$R_{WPD}$	—	55	—	kΩ	25°C, VDD=5.0V Vpin = VDD
I/O 输入端口 VDD/2 输	$V_{VDD/2}$	—	±3%	—		25°C, VDD=5V, 弱

出精度						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℃），相对于常温的变化量在±10%以内。

◆ 芯片输出端口特性表

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 端口输出高电平	V <sub>OH</sub>	0.8VDD	—	VDD	V	2.2V ≤ VDD ≤ 5.5V, 端口无负载
I/O 端口输出低电平	V <sub>OL</sub>	VSS	—	0.2VDD	V	
I/O 端口灌电流 (PA0~PA3 除外)	I <sub>OL1</sub>	—	10	—	mA	25℃, VDD = 5V V <sub>OL</sub> = 0.6V, 普通驱动
	I <sub>OL2</sub>	—	22	—		25℃, VDD = 5V V <sub>OL</sub> = 0.6V, 增强驱动
	I <sub>OL3</sub>	—	40	—		25℃, VDD = 5V V <sub>OL</sub> = 1.4V, 增强驱动
I/O 端口灌电流 (PA0~PA3)	I <sub>OL4</sub>	—	15	—	mA	25℃, VDD = 5V V <sub>OL</sub> = 0.6V, 普通驱动
	I <sub>OL5</sub>	—	100	—		25℃, VDD = 5V V <sub>OL</sub> = 0.6V, 增强驱动
I/O 端口拉电流 (PA0~PA3 除外)	I <sub>OH1</sub>	—	7	—	mA	25℃, VDD = 5V V <sub>OH</sub> = 4.4V, 普通驱动
	I <sub>OH2</sub>	—	14	—		25℃, VDD = 5V V <sub>OH</sub> = 4.4V, 增强驱动
I/O 端口拉电流 (PA0~PA3)	I <sub>OH3</sub>	—	16	—	mA	25℃, VDD = 5V V <sub>OH</sub> = 4.4V, 普通驱动
	I <sub>OH4</sub>	—	30	—		25℃, VDD = 5V V <sub>OH</sub> = 4.4V, 增强驱动

注：I/O 端口拉电流 I<sub>OH</sub> 和灌电流 I<sub>OL</sub> 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 <sub>OSC</sub>	—	—	48M	Hz	2.2V ≤ VDD ≤ 5.5V
系统时钟周期	T <sub>OSC</sub>	20.8	—	—	ns	
机器周期	T <sub>inst</sub>	—	T <sub>OSC</sub> * 1	—	ns	
外部时钟高电平和低电平时间	T <sub>OSL</sub> , T <sub>OSH</sub>	20	—	—	ns	
外部时钟边沿上升和下降时间	T <sub>OSR</sub> , T <sub>OSF</sub>	—	—	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	4000	V	25°C, 遵循标准 MIL-STD-883H
ESD 电压 (充电器件模型)	$V_{ESDCDM}$	C3	1000	V	25°C, 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。

### ◆ 静态门锁特性参数表

参数	符号	等级	测试条件
Latchup 电流	$I_{LU}$	IA	25°C, 遵循标准 JESD78

注: 上述静态门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	
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	
输入采样电容	$C_{IN}$	—	11	—	pF	
模拟通道推荐输入电阻	$R_{IN}$	—	—	10	K $\Omega$	采样时间 2us
		—	—	20	K $\Omega$	采样时间 5us
		—	—	45	K $\Omega$	采样时间 10us
		—	—	95	K $\Omega$	采样时间 20us
AD 转换时钟周期	$T_{AD1}$	1	—	—	$\mu$ s	参考电压为内部 VREFP
	$T_{AD2}$	0.5	—	—	$\mu$ s	参考电压为 VDD 或外部 AVREFP
AD 转换时间 (不包括采样时间)	$T_{CONV}$	—	15	—	$T_{AD}$	12bit 分辨率
		—	13	—	$T_{AD}$	10bit 分辨率
		—	11	—	$T_{AD}$	8bit 分辨率
		—	9	—	$T_{AD}$	6bit 分辨率
差分线性度	DNL	—	$\pm 1$	$\pm 2$	LSB	ADC 时钟频率为 1MHz, 参考电压为 VDD=5V, 常温
失调误差	$V_{OFFSET}$	—	2.5	4	mV	

注 1:  $T_{AD1}$  为 ADC 使用内部 VREFP 作为参考电压时的 AD 转换时钟周期;

$T_{AD2}$  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。

注 2: 为了保证 ADC 转换结果稳定可靠、避免噪声干扰, 建议在模拟输入通道接外部电容 (100nF 或 10nF) 进行滤波。

注 3: 模拟通道输入电阻  $R_{IN}$  最大值除了与采样时间有关外, 还与通道是否外接电容有关, 需根据实际应用条件来测试确定合适的电阻值 (即外部模拟信号的输出阻抗), 上表中的  $R_{IN}$  推荐值在测试时通道无外接电容, 仅供参考。



◆ 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 <sub>ADCLK</sub> = 1us
FPCLK /4	不推荐使用	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 2us
FPCLK /8	不推荐使用	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 4us
FPCLK /16	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 1us	T <sub>ADCLK</sub> = 8us
FPCLK /32	不推荐使用	T <sub>ADCLK</sub> = 1us	T <sub>ADCLK</sub> = 2us	T <sub>ADCLK</sub> = 16us
FPCLK /64	T <sub>ADCLK</sub> = 1.3us	T <sub>ADCLK</sub> = 2us	T <sub>ADCLK</sub> = 4us	T <sub>ADCLK</sub> = 32us
FPCLK /256	T <sub>ADCLK</sub> = 5.3us	T <sub>ADCLK</sub> = 8us	T <sub>ADCLK</sub> = 16us	T <sub>ADCLK</sub> = 128us
FLRC	T <sub>ADCLK</sub> = 31us	T <sub>ADCLK</sub> = 31us	T <sub>ADCLK</sub> = 31us	T <sub>ADCLK</sub> = 31us

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 <sub>ADCLK</sub> = 0.5us
FPCLK /2	不推荐使用	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 1us
FPCLK /4	不推荐使用	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 2us
FPCLK /8	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 0.5us	T <sub>ADCLK</sub> = 4us
FPCLK /16	不推荐使用	T <sub>ADCLK</sub> = 0.5us	T <sub>ADCLK</sub> = 1us	T <sub>ADCLK</sub> = 8us
FPCLK /32	T <sub>ADCLK</sub> = 0.67us	T <sub>ADCLK</sub> = 1us	T <sub>ADCLK</sub> = 2us	T <sub>ADCLK</sub> = 16us
FPCLK /64	T <sub>ADCLK</sub> = 1.3us	T <sub>ADCLK</sub> = 2us	T <sub>ADCLK</sub> = 4us	T <sub>ADCLK</sub> = 32us
FPCLK /256	T <sub>ADCLK</sub> = 5.3us	T <sub>ADCLK</sub> = 8us	T <sub>ADCLK</sub> = 16us	T <sub>ADCLK</sub> = 128us
FLRC	T <sub>ADCLK</sub> = 31us	T <sub>ADCLK</sub> = 31us	T <sub>ADCLK</sub> = 31us	T <sub>ADCLK</sub> = 31us

◆ 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 <sub>REF</sub>	2.028	2.048	2.068	V	25°C, VDD=5V, CHOP_EN=1
1/4VDD 电源分压	V <sub>1/4VDD</sub>	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 <sub>HRC1</sub>	47.71	48	48.29	MHZ	25°C, VDD=5.0V
	F <sub>HRC2</sub>	31.68	32	32.32	MHZ	
	F <sub>HRC3</sub>	15.84	16	16.16	MHZ	

	F <sub>HRC4</sub>	1.98	2	2.02	MHz	
	F <sub>HRC1</sub>	47.52	48	48.48	MHz	-40℃~85℃, VDD=2.6V~5.5V
	F <sub>HRC2</sub>	31.2	32	32.64	MHz	-40℃~85℃, VDD=2.2V~5.5V
	F <sub>HRC3</sub>	15.6	16	16.32	MHz	
	F <sub>HRC4</sub>	1.93	2	2.07	MHz	
HRC 起振时间	T <sub>HRC</sub>	—	10	—	us	-40℃~85℃, VDD=2.2V~5.5V
LRC 时钟频率	F <sub>LRC</sub>	30	31.25	32.5	KHz	25℃, VDD=5.0V
		28.75	31.25	33.25	KHz	-40℃~85℃, 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\_BORV（CFG\_WORD0<6:4>）设置）

CFG_BORV<2:0>	最小值	典型值	最大值	单位	测试条件
000	—	—	—	—	-40~85℃
001	1.9	2.1	2.35	V	
010	2.35	2.5	2.95	V	
011	2.65	2.8	2.95	V	
100	2.95	3.1	3.75	V	
101	3.45	3.6	3.75	V	
110	3.85	4.0	4.75	V	
111	—	—	—	—	
BOR 电压档位迟滞窗口	60	90	140	mV	-40~85℃

注：2.1V, 2.8V, 3.6V 这三个 BOR 档位电压精度高，推荐优先使用这三个档位进行配置。

◆ 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	2.15	2.3	2.45	V	-40~85℃
	0001	2.25	2.4	2.6	V	
	0010	2.55	2.7	2.85	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

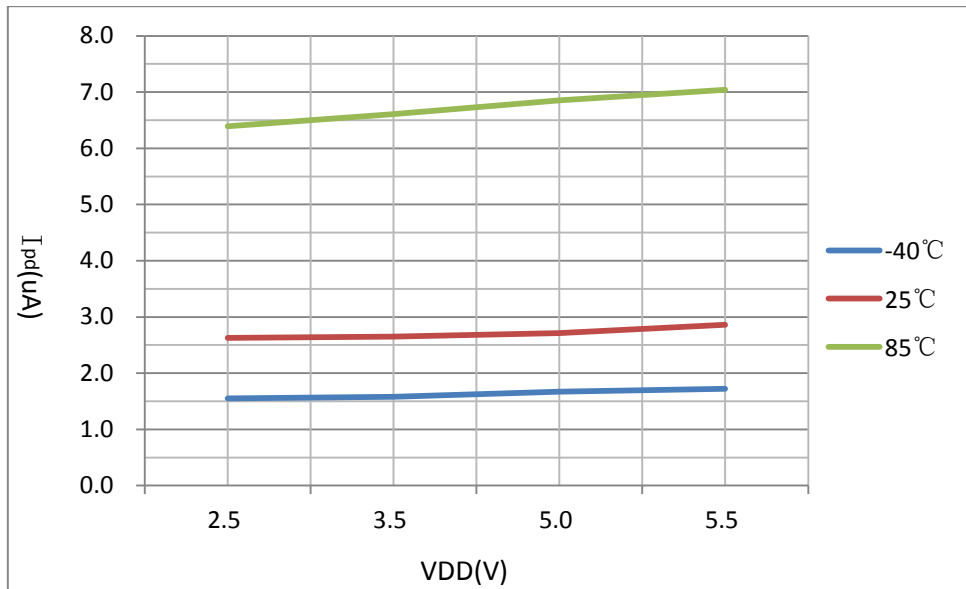
SCU\_WAKEUPTIME 的 BG\_STOP=0)，此时芯片深睡眠模式功耗会增大约 0.5 $\mu$ A。上表中的 BOR 和 LVD 档位电压描述，为芯片在工作模式(包括浅睡眠模式)或深睡眠模式(寄存器 SCU\_WAKEUPTIME 的 BG\_STOP=0)条件下的档位电压。

## 附录2.2 参数特性图

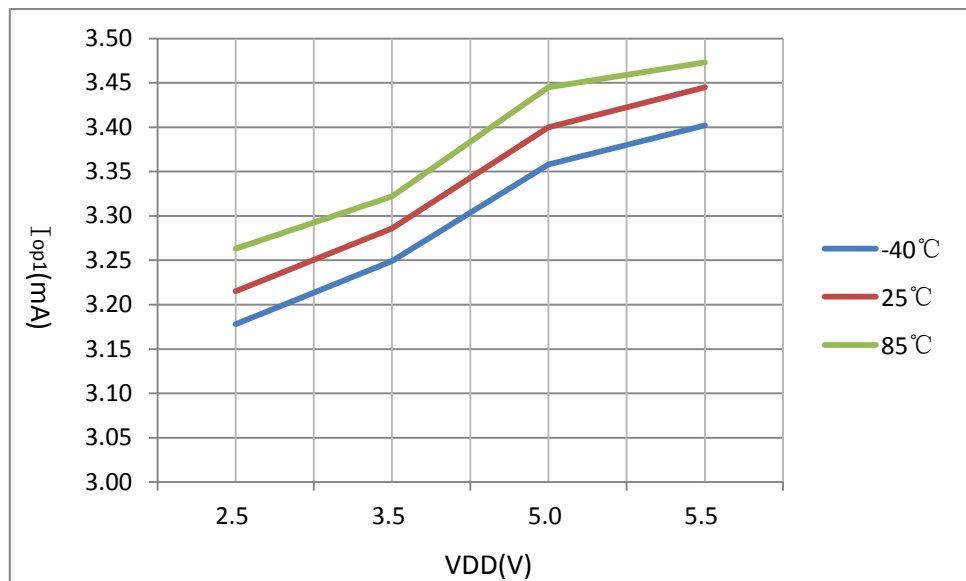
本节中所列图示均为抽样测试，仅作为设计参考之用。其中部分图示中所列的数据已超出指定的操作范围，此类信息也仅供参考，芯片只保证在指定的范围内正常工作。

### 附录2.2.1 芯片功耗特性

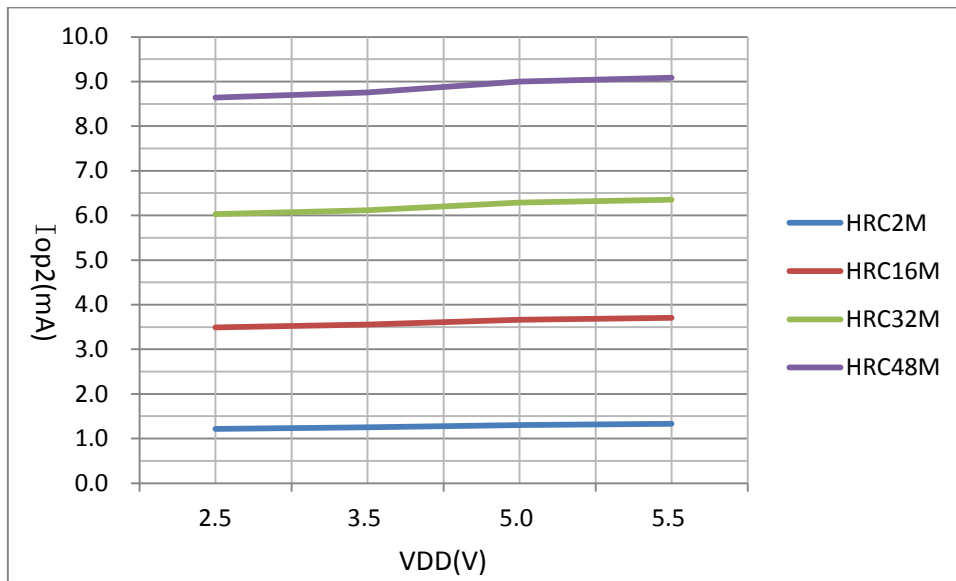
- ◆ 芯片深度睡眠模式电流随电压-温度变化特性图（IWDT 不使能，所有 I/O 端口输出固定电平，无负载）



- ◆ 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）

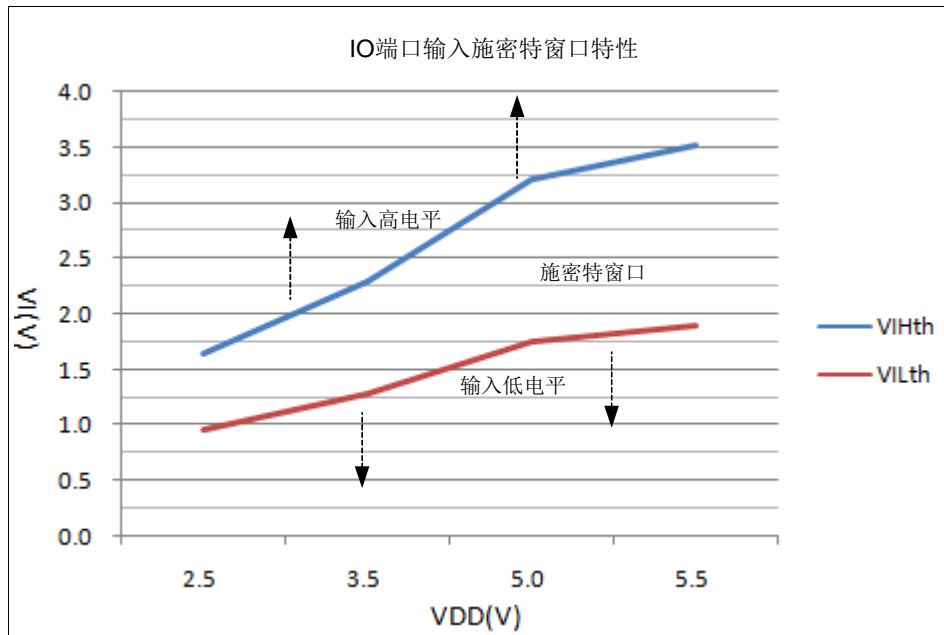


- ◆ 芯片运行模式电流随电压-系统时钟频率变化特性图（IWDT 使能，外设模块均工作，所有 I/O 端口输出固定电平，无负载，ADC 使用 VDD 作为正向参考电压；室温 25℃）

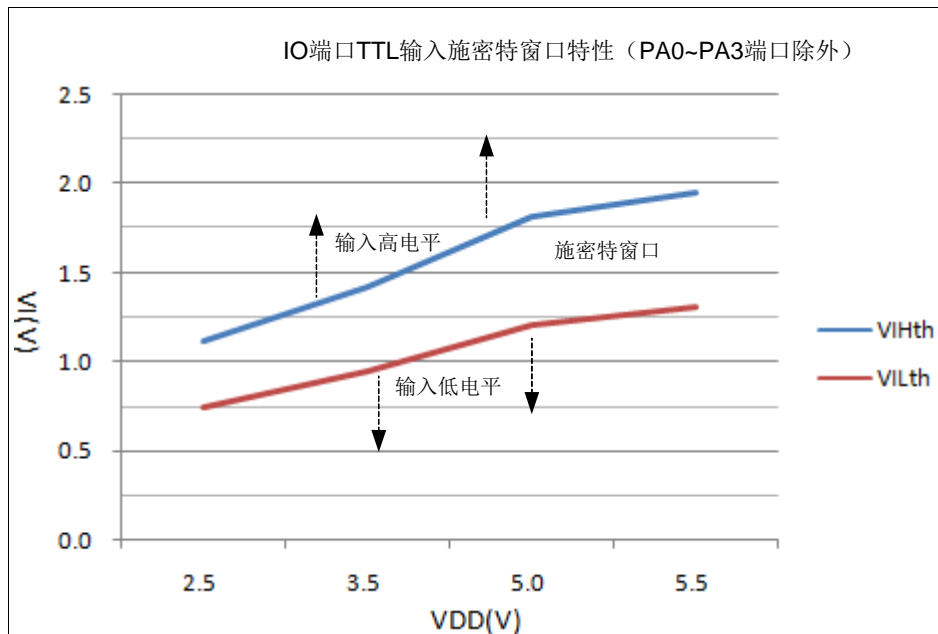


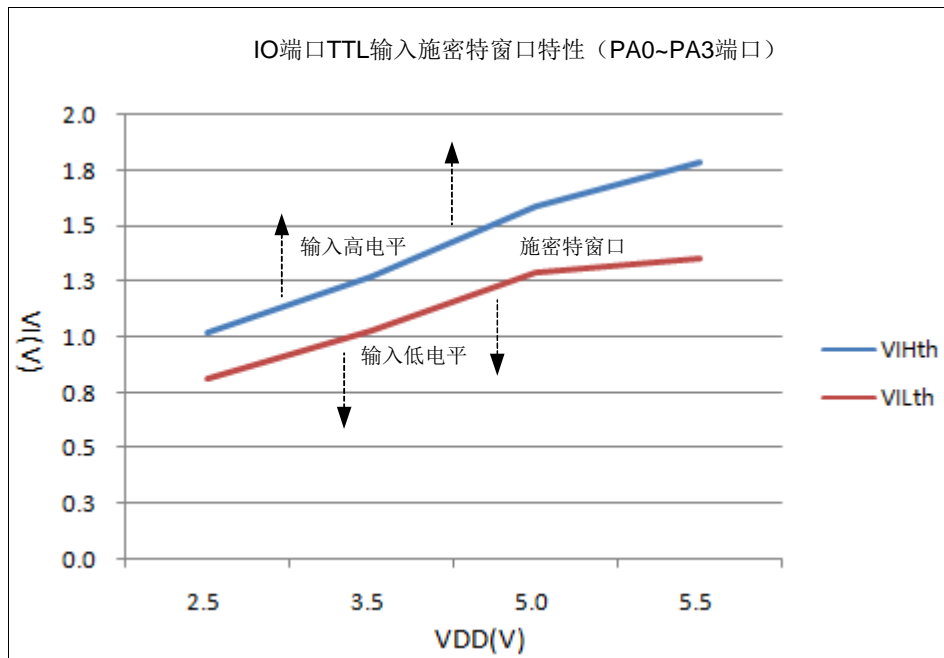
## 附录2.2.2 芯片IO端口输入特性

### ◆ 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℃）



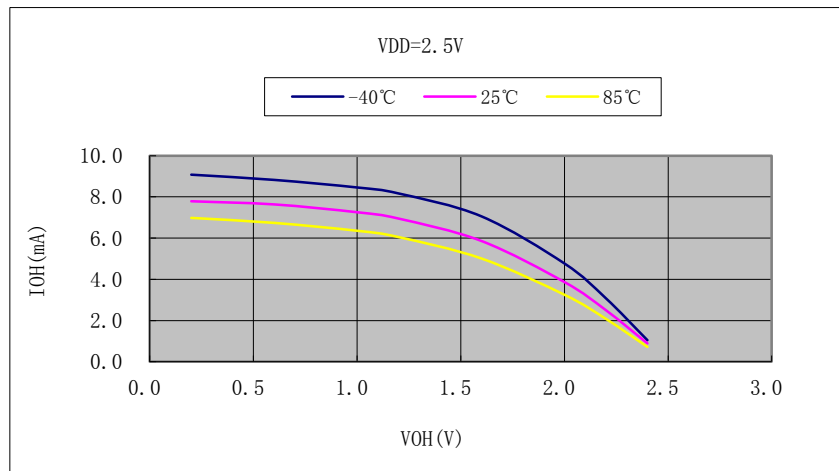


- 注 1:  $V_{IHth}$  为施密特窗口的上阈值电平, 大于该阈值的输入电平为高;
- 注 2:  $V_{ILth}$  为施密特窗口的下阈值电平, 小于该阈值的输入电平为低;
- 注 3:  $V_{IHth}$  和  $V_{ILth}$  之间为施密特窗口, 在窗口内的输入电平不确定, 可能为高或低。

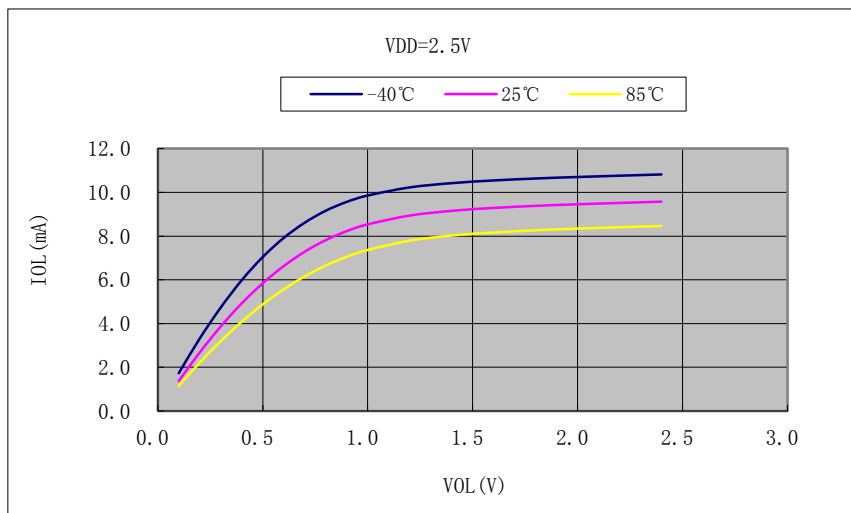
附录2.2.3 芯片IO端口输出特性（普通驱动，PA0~PA3 端口除外）

◆ I/O 端口信号输出特性图

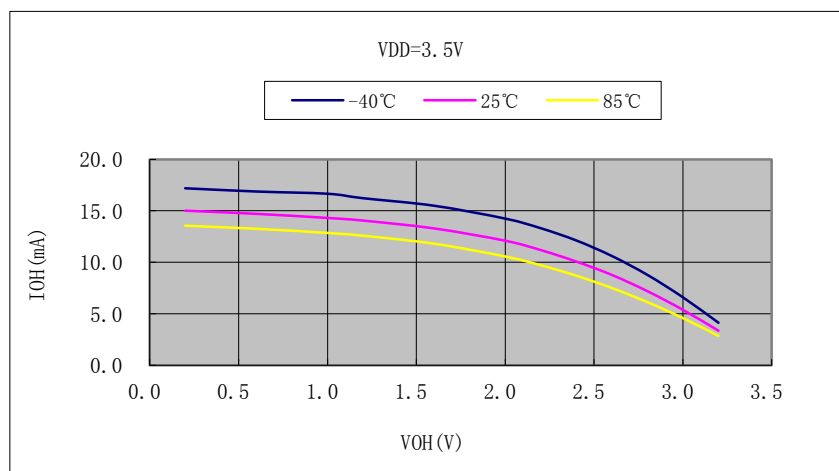
A:  $V_{OH}$  vs  $I_{OH}$ @VDD=2.5V



B:  $V_{OL}$  vs  $I_{OL}$ @VDD=2.5V

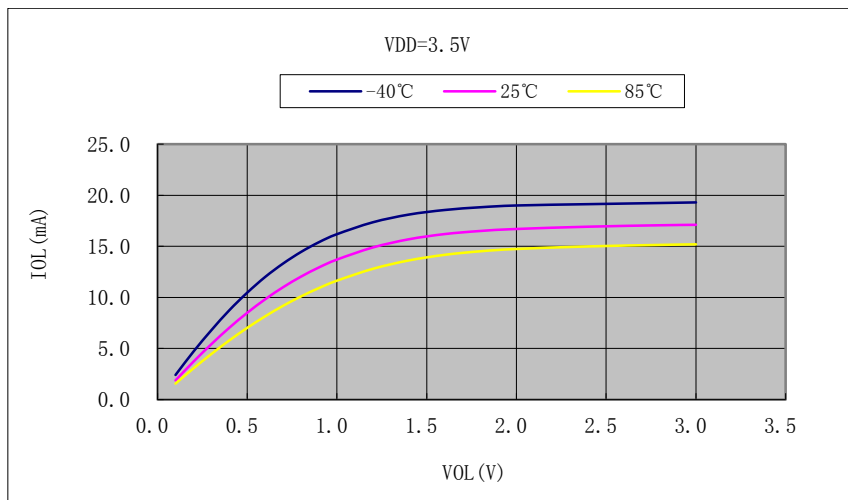


C:  $V_{OH}$  vs  $I_{OH}$ @VDD=3.5V

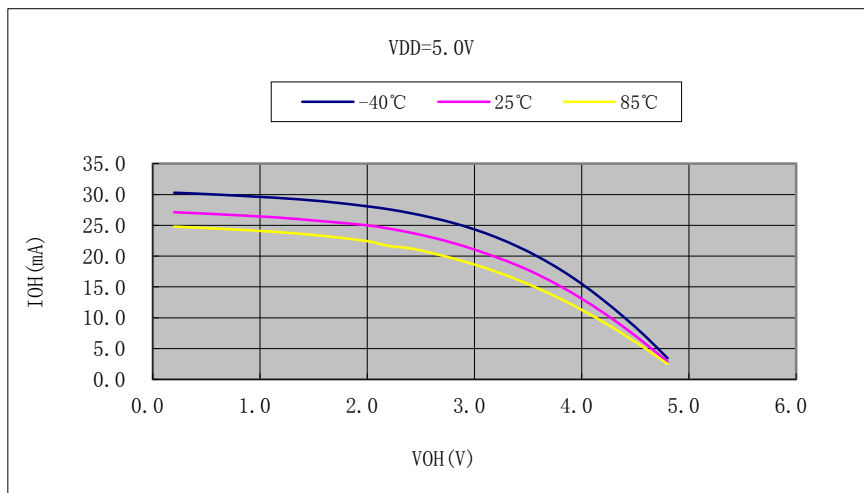




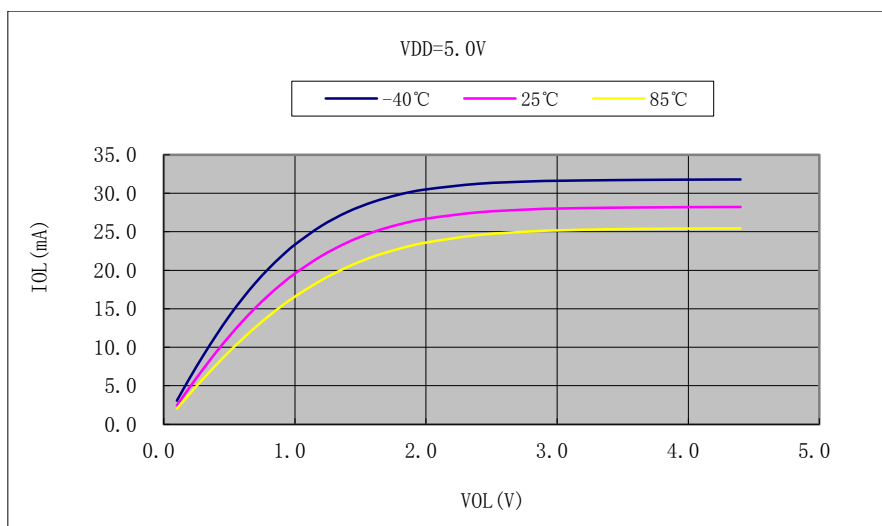
D:  $V_{OL}$  vs  $I_{OL}$  @  $V_{DD}=3.5V$



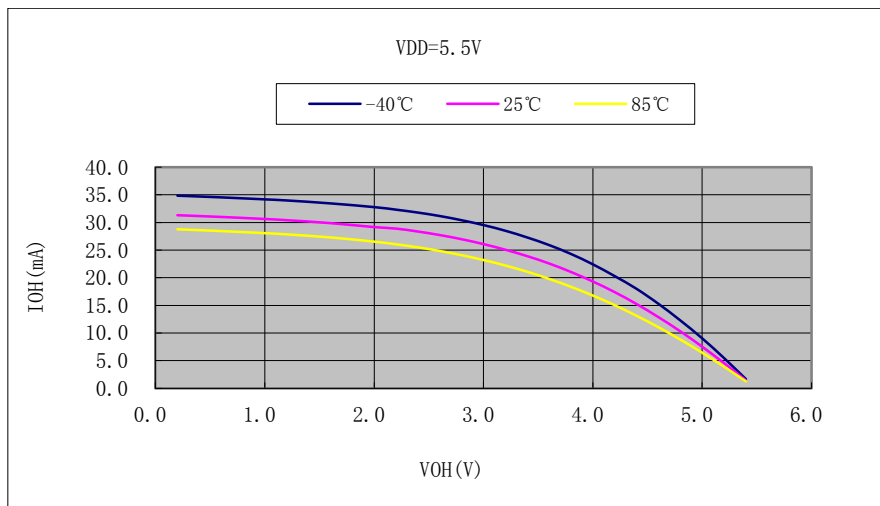
E:  $V_{OH}$  vs  $I_{OH}$  @  $V_{DD}=5.0V$



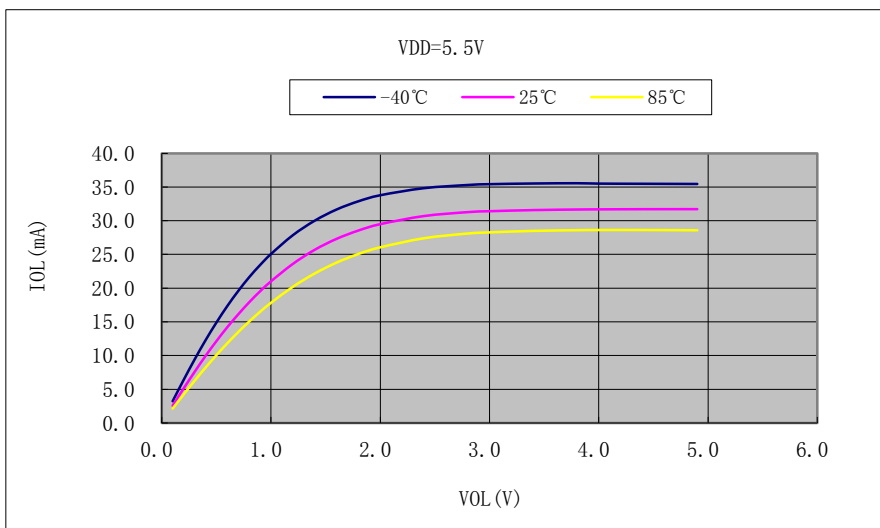
F:  $V_{OL}$  vs  $I_{OL}$  @  $V_{DD}=5.0V$



G:  $V_{OH}$  vs  $I_{OH}$ @VDD=5.5V



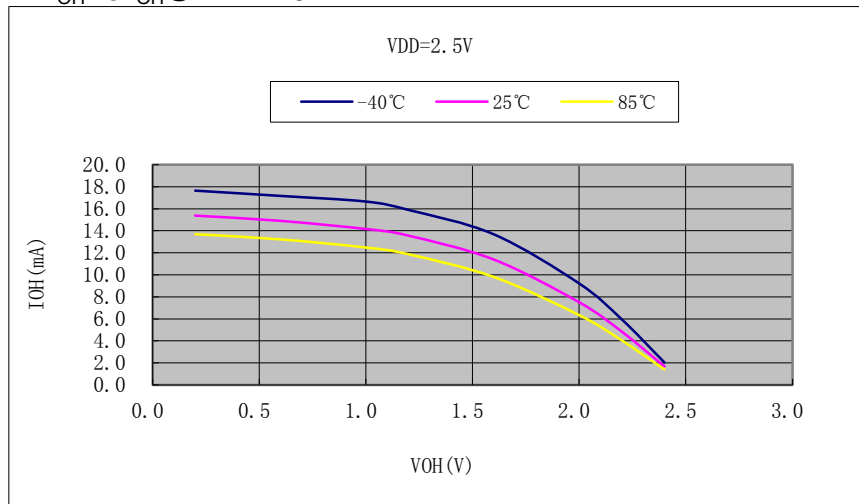
H:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.5V



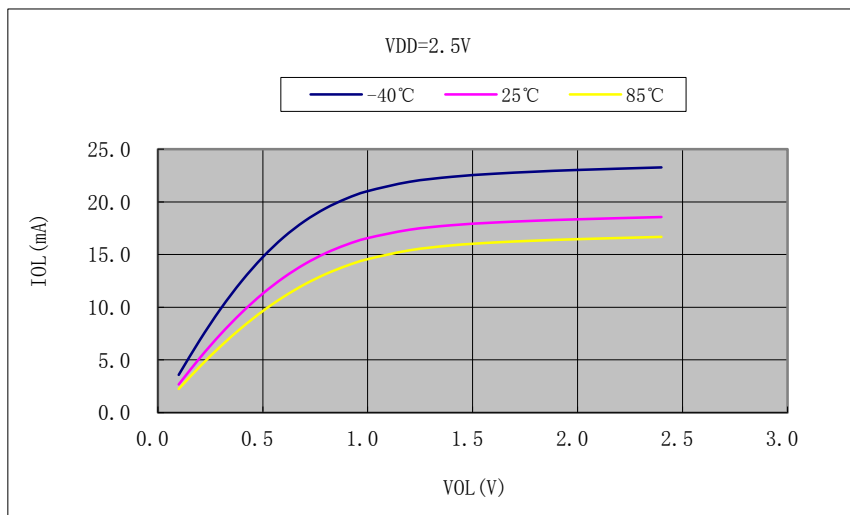
附录2.2.4 芯片IO端口输出特性（增强驱动，PA0~PA3 端口除外）

◆ I/O 端口信号输出特性图

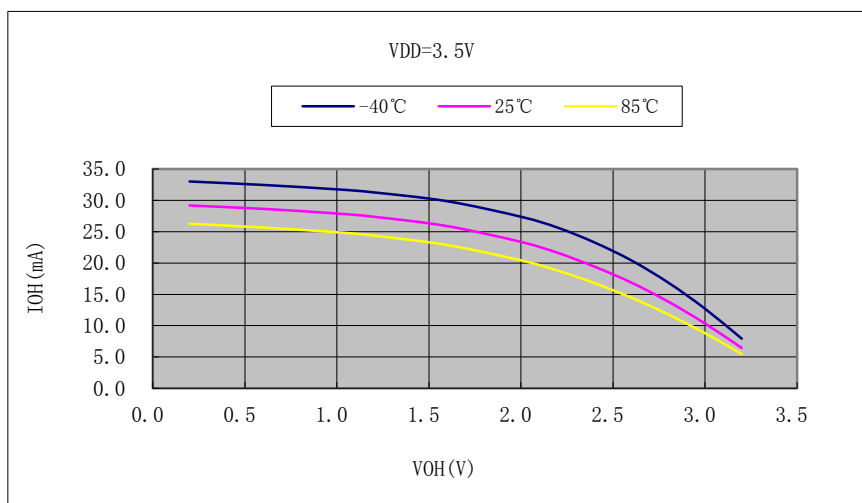
A:  $V_{OH}$  vs  $I_{OH}$ @VDD=2.5V



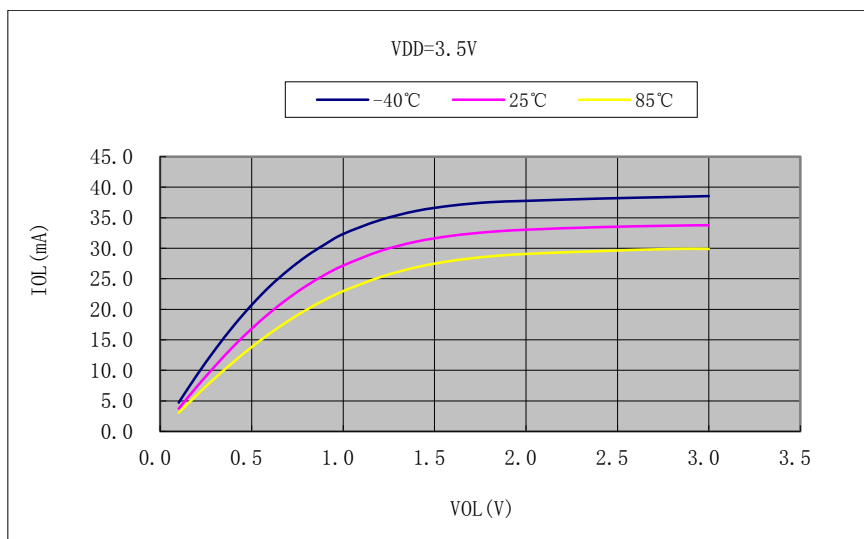
B:  $V_{OL}$  vs  $I_{OL}$ @VDD=2.5V



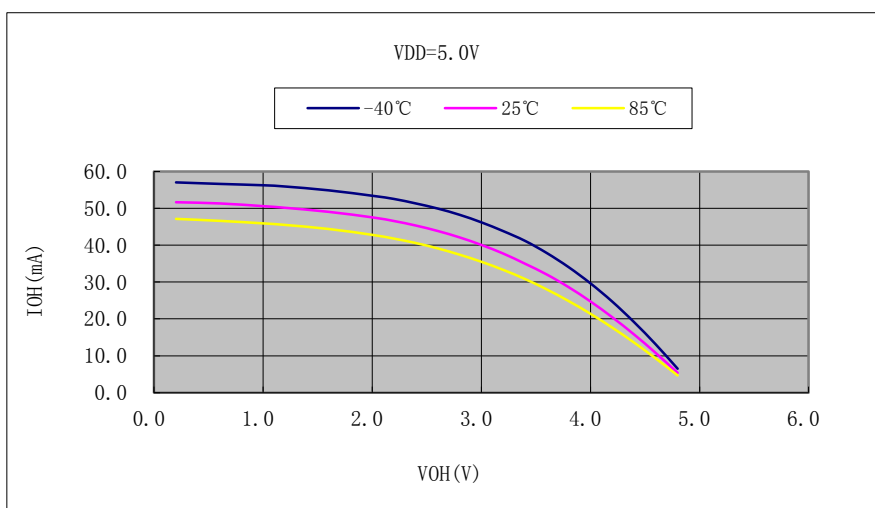
C:  $V_{OH}$  vs  $I_{OH}$ @VDD=3.5V



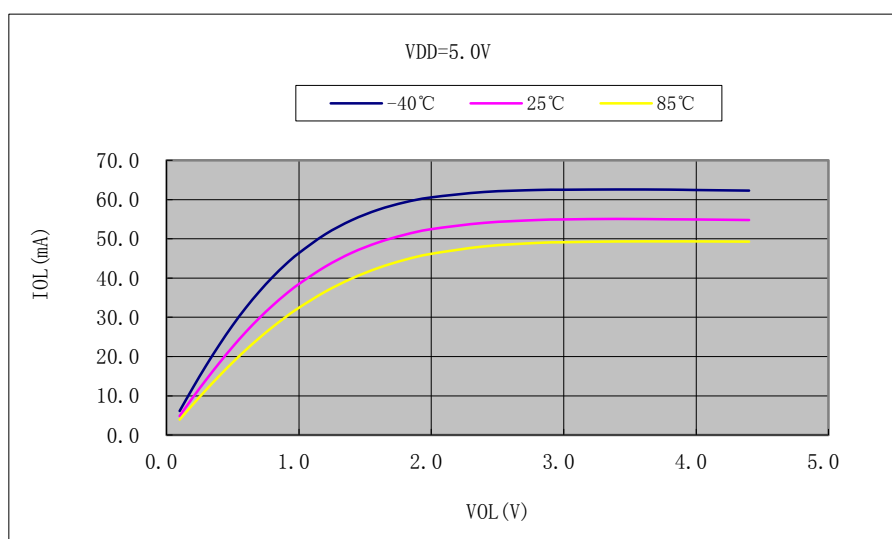
D:  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=3.5V$



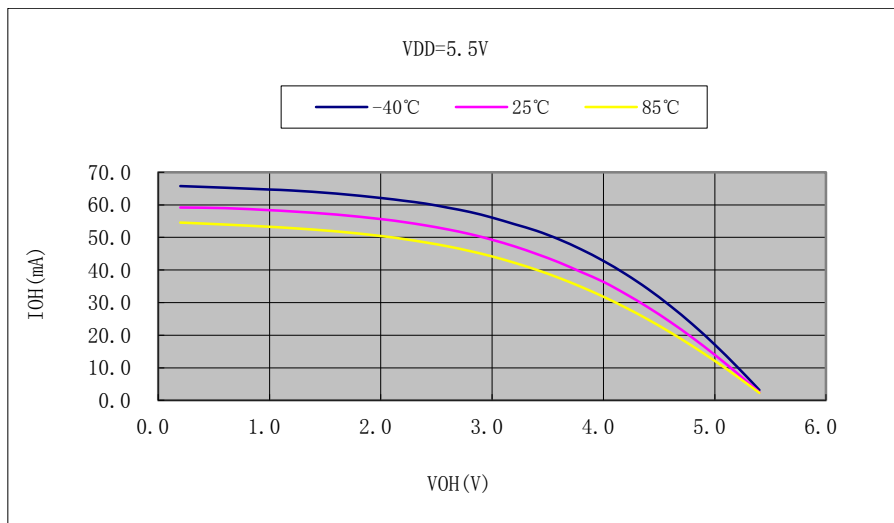
E:  $V_{OH}$  vs  $I_{OH}$ @ $V_{DD}=5.0V$



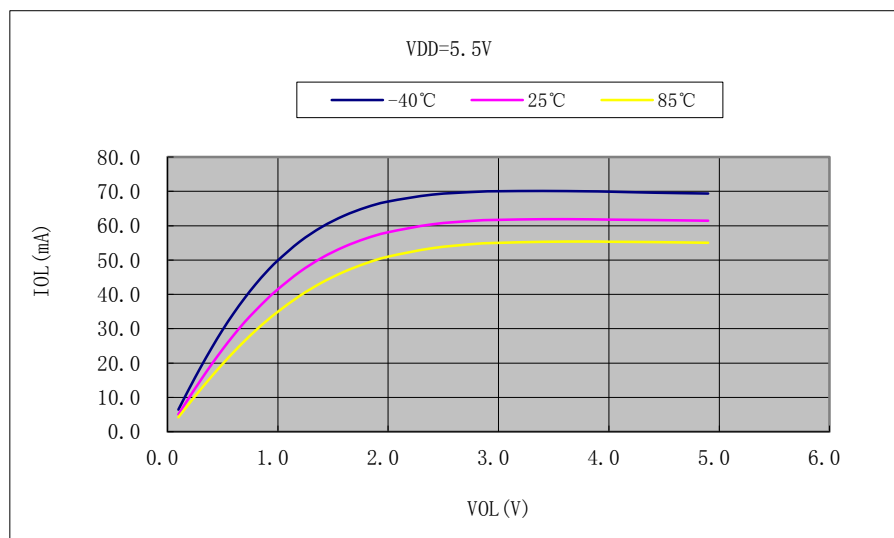
F:  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=5.0V$



G:  $V_{OH}$  vs  $I_{OH}$ @VDD=5.5V



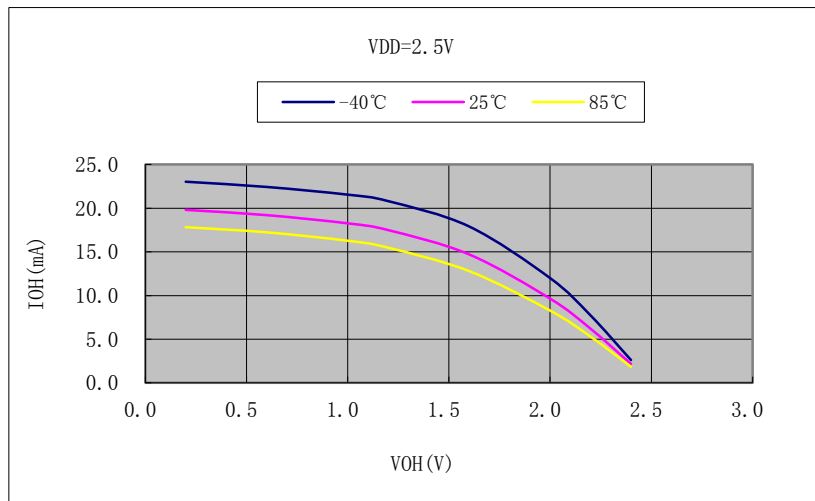
H:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.5V



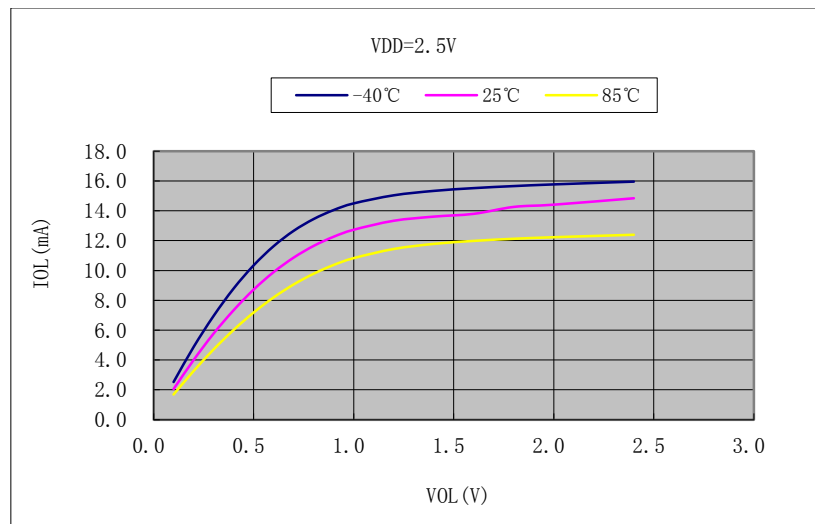
附录2.2.5 芯片IO端口输出特性（普通驱动，PA0~PA3 端口）

◆ I/O 端口信号输出特性图

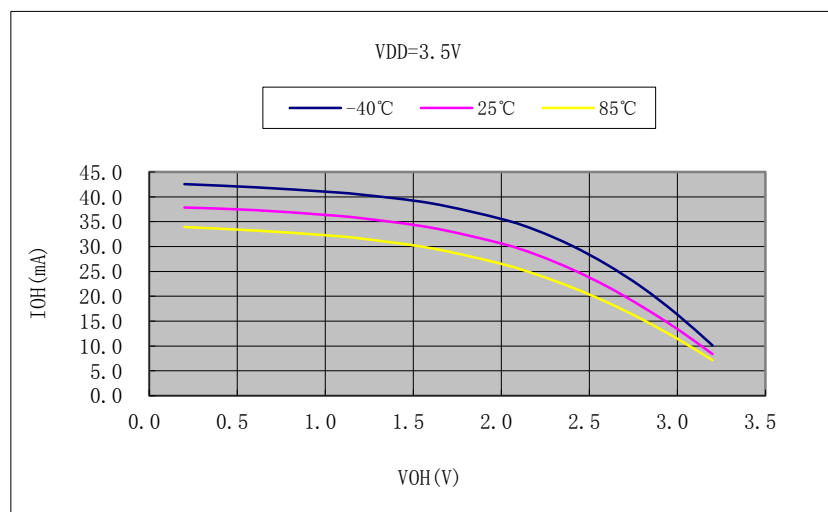
A:  $V_{OH}$  vs  $I_{OH}$ @VDD=2.5V



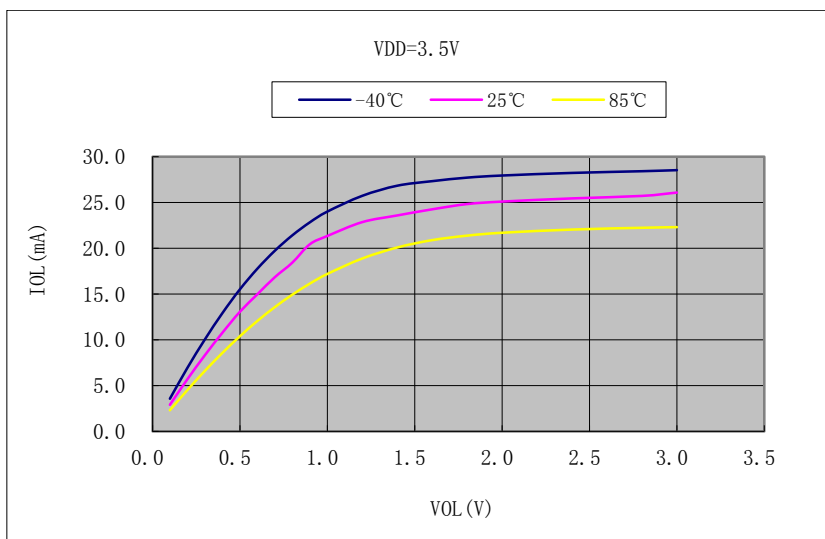
B:  $V_{OL}$  vs  $I_{OL}$ @VDD=2.5V



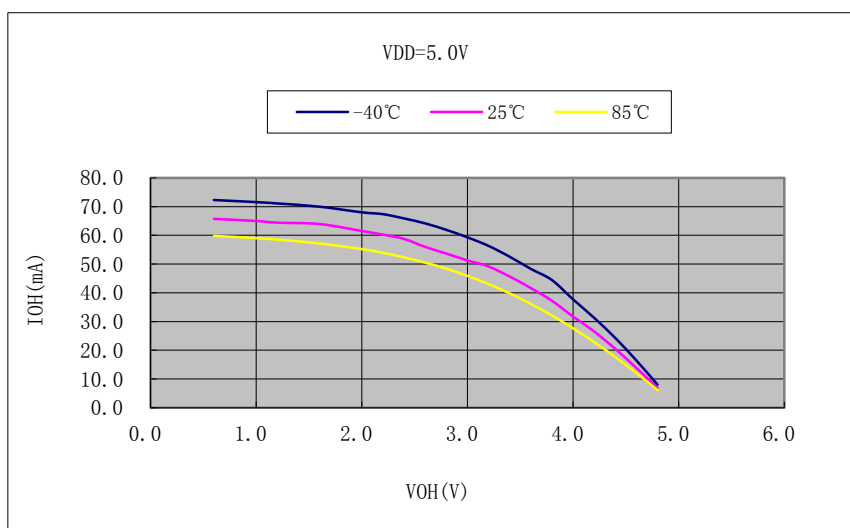
C:  $V_{OH}$  vs  $I_{OH}$ @VDD=3.5V



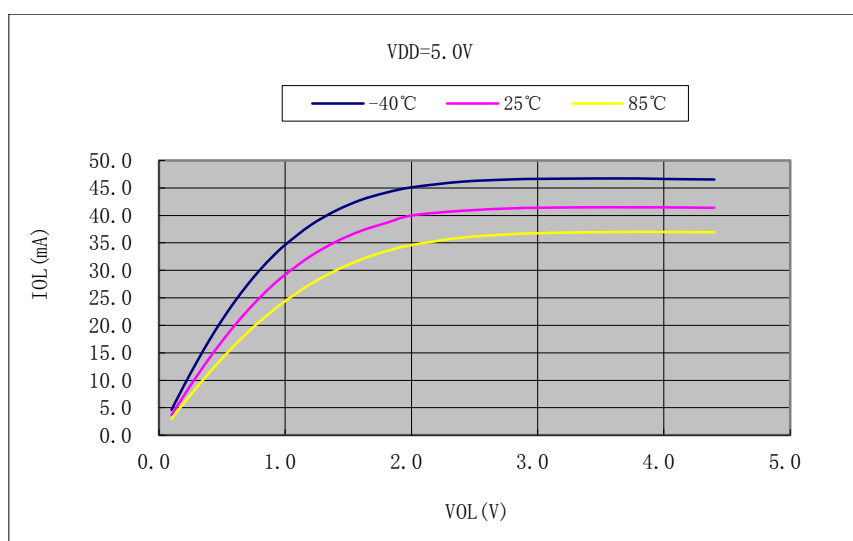
D:  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=3.5V$



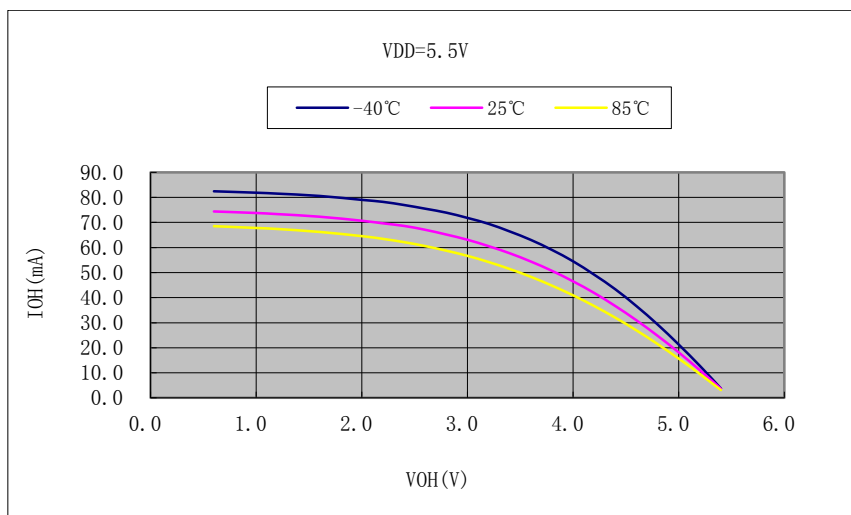
E:  $V_{OH}$  vs  $I_{OH}$ @ $V_{DD}=5.0V$



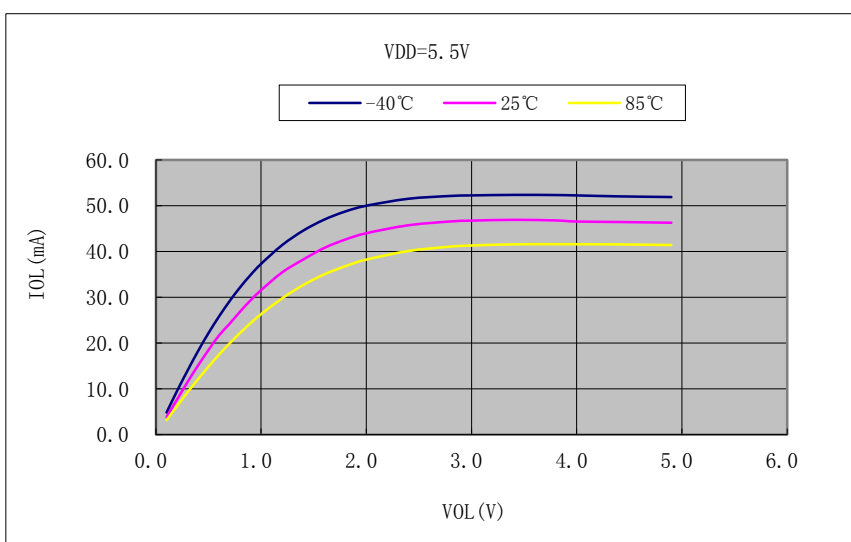
F:  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=5.0V$



G:  $V_{OH}$  vs  $I_{OH}$ @VDD=5.5V



H:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.5V

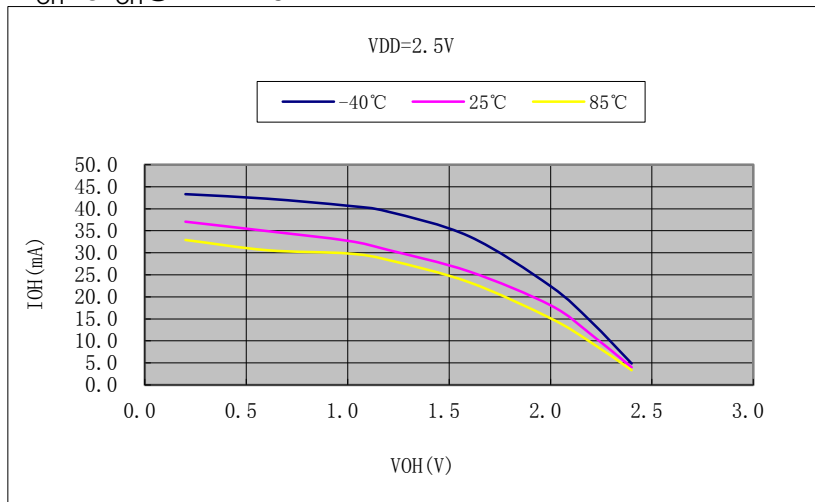




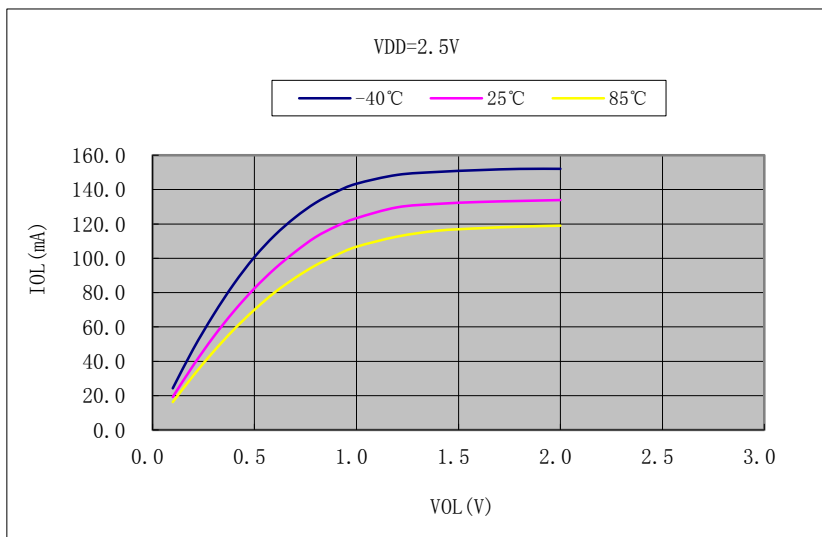
附录2.2.6 芯片IO端口输出特性（增强驱动，PA0~PA3 端口）

◆ I/O 端口信号输出特性图

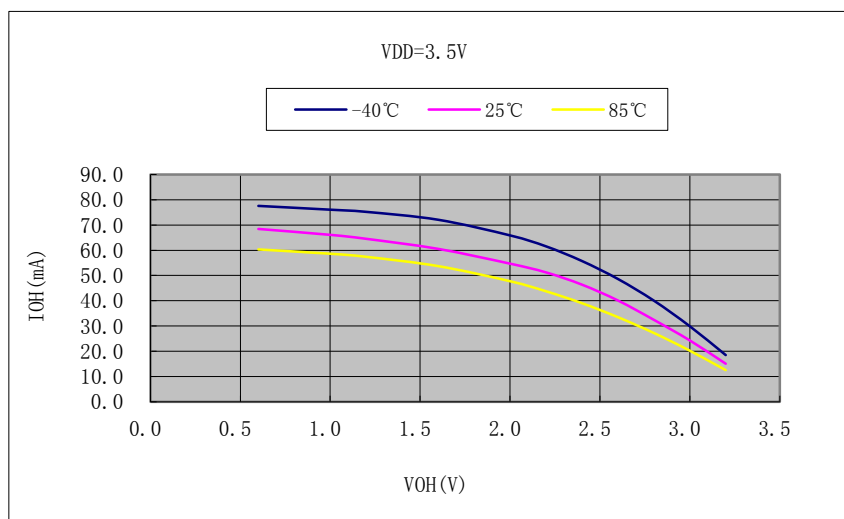
A:  $V_{OH}$  vs  $I_{OH}$ @VDD=2.5V



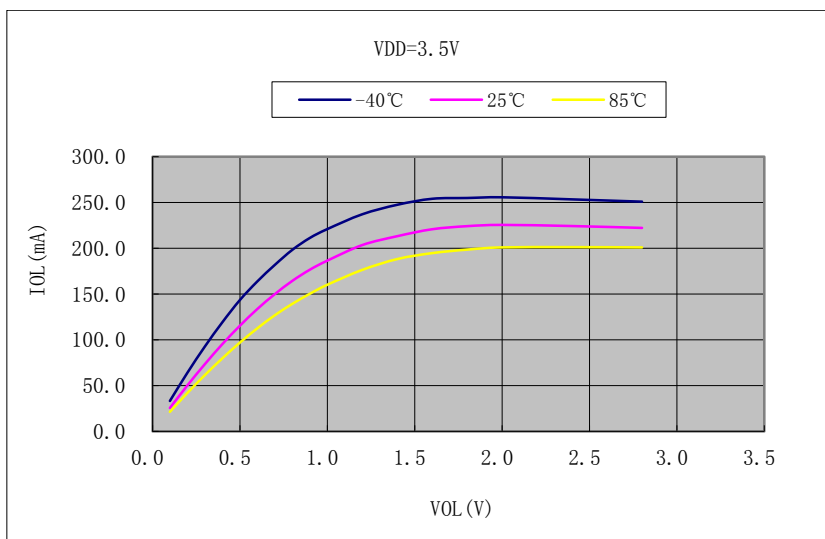
B:  $V_{OL}$  vs  $I_{OL}$ @VDD=2.5V



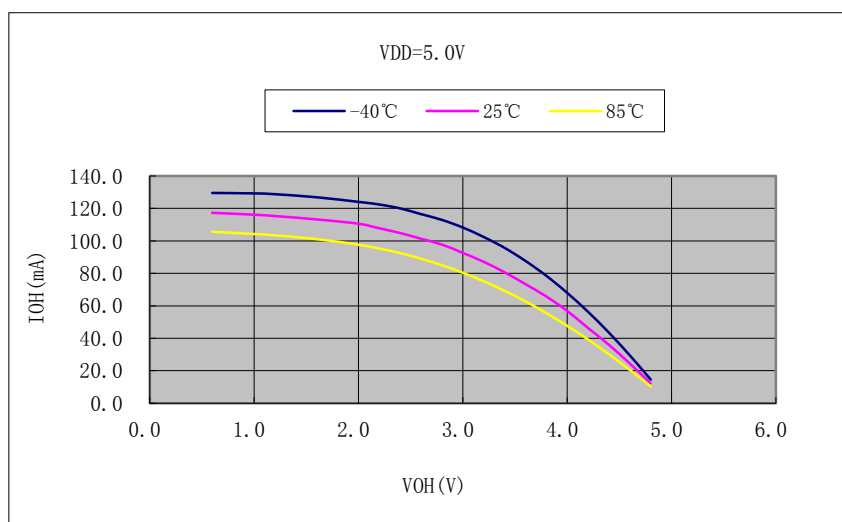
C:  $V_{OH}$  vs  $I_{OH}$ @VDD=3.5V



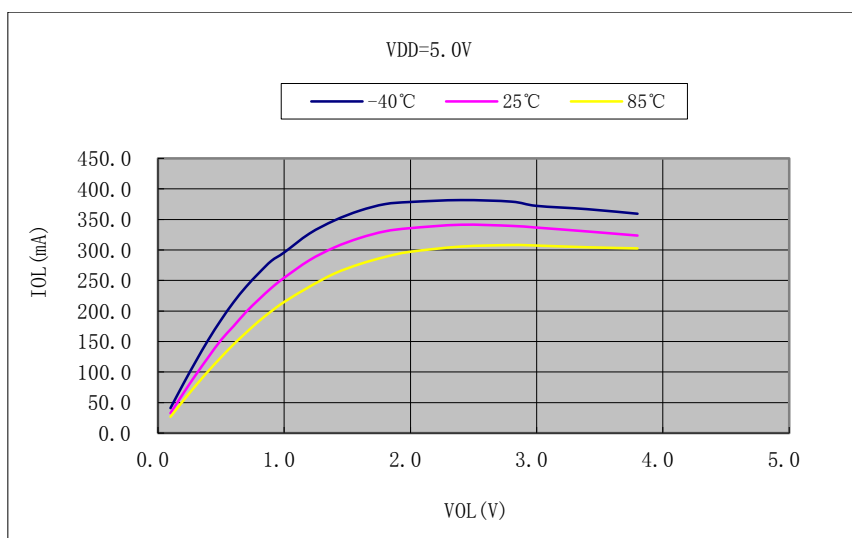
D:  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=3.5V$



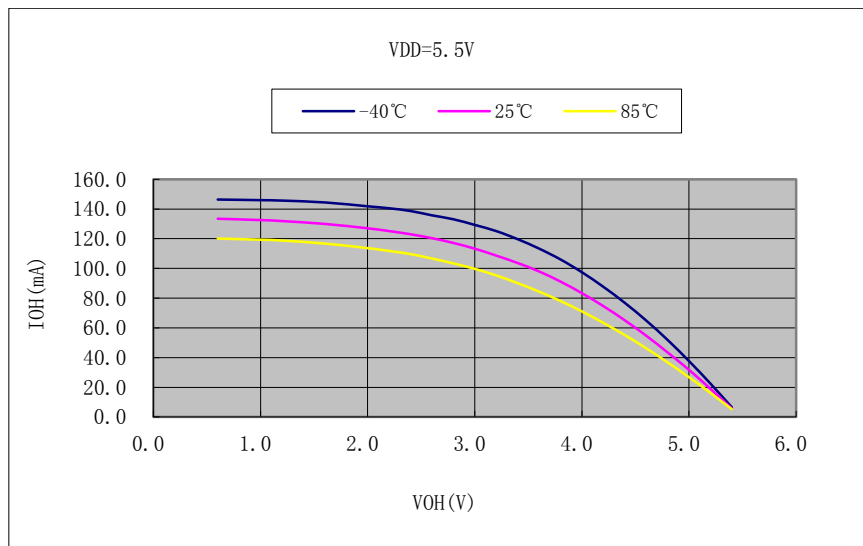
E:  $V_{OH}$  vs  $I_{OH}$ @ $V_{DD}=5.0V$



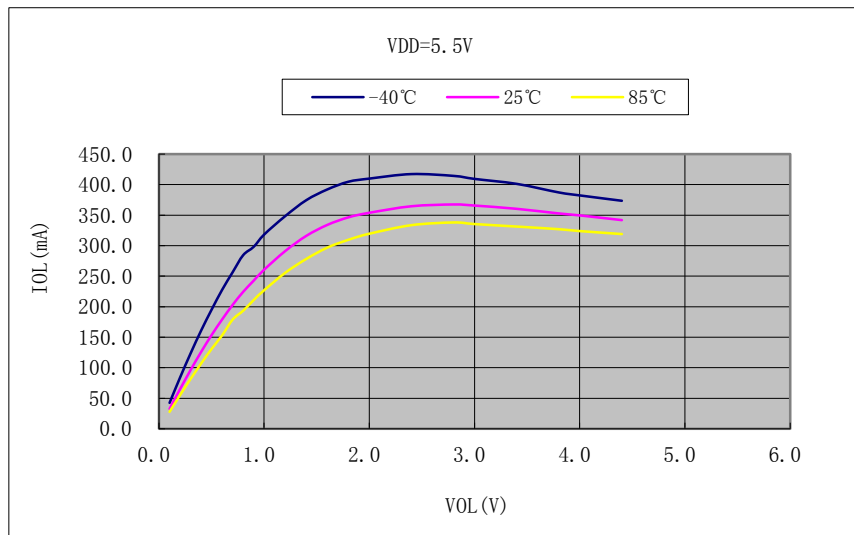
F:  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=5.0V$



G:  $V_{OH}$  vs  $I_{OH}$ @VDD=5.5V



H:  $V_{OL}$  vs  $I_{OL}$ @VDD=5.5V



## 附录3 编程调试接口

### 附录3.1 概述

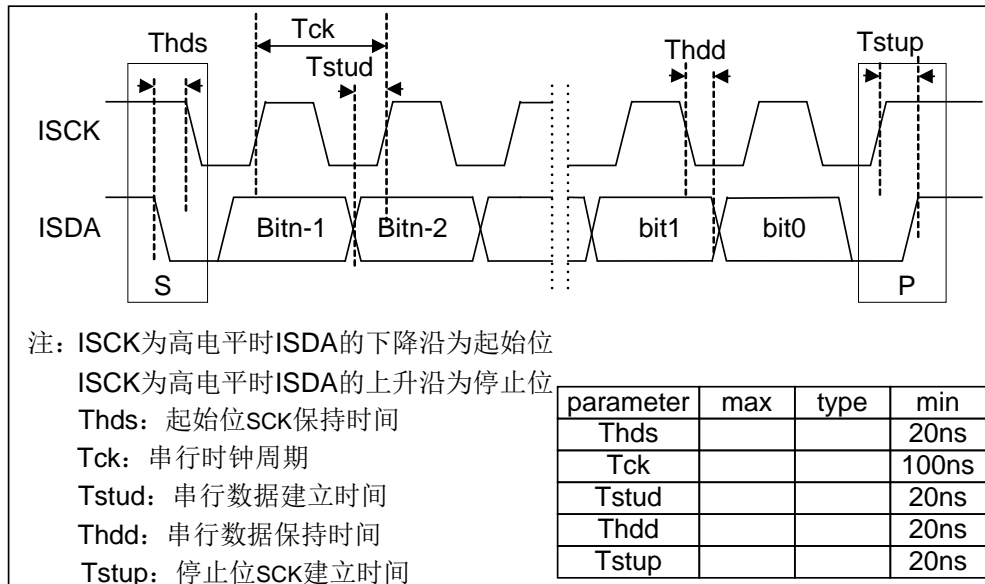
为方便应用程序及实际系统调试，芯片内部集成 ISP 在线编程接口和 SWD 串行调试接口，通过上海东软载波微电子有限公司授权的 ISP 编程器、SWD 调试器可实现芯片在线编程、仿真调试功能。

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

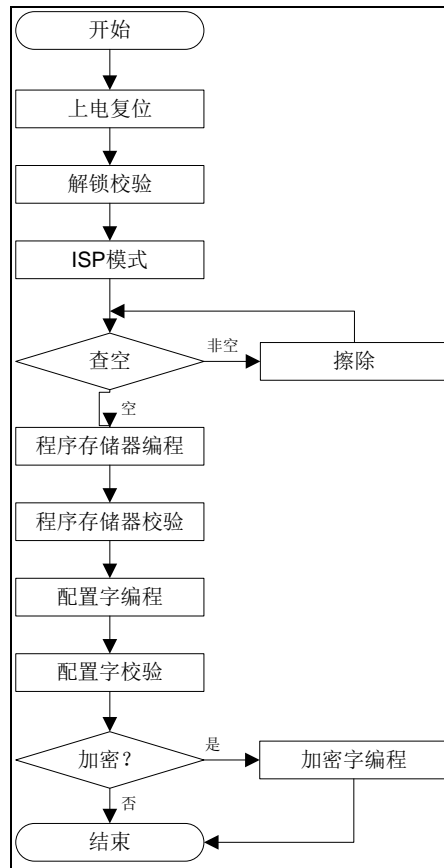
#### 附录3.2.1 通信协议

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 在线编程接口请参考“管脚对照表”章节的描述。

### 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\_SWD，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 调试接口与 GPIO 复用，具体请参考“表 1-3 ISP0/SWD 管脚对照表”。

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\_SWD 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

## 附录4 内置EEPROM芯片

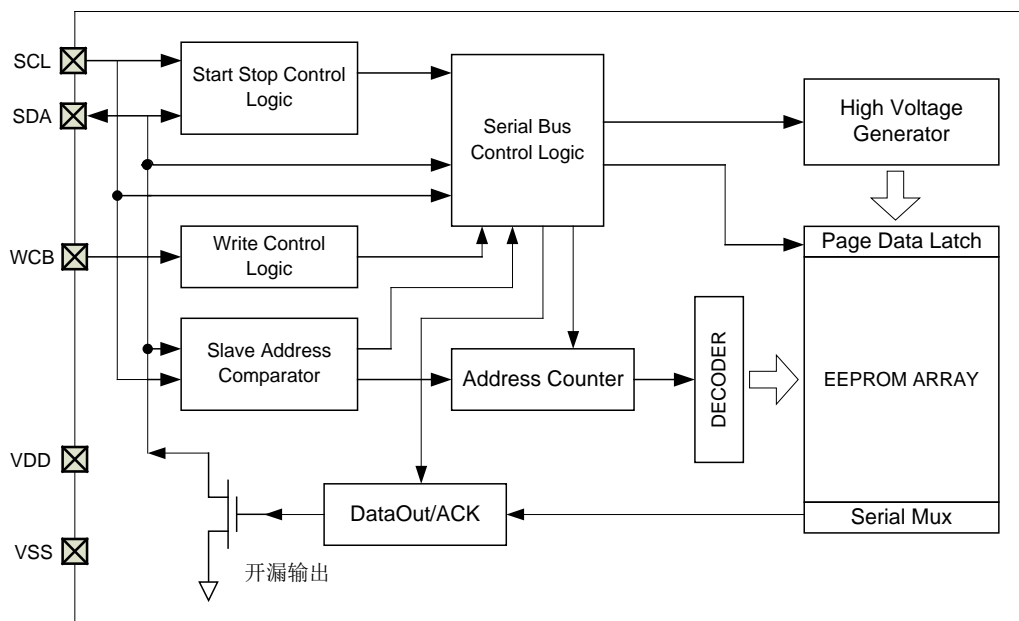
### 附录4.1 概述

- ◆ 存储器存储空间：1K 字节，每页 16 字节
- ◆ 2 线串行接口，完全兼容 I2C 总线传输协议
- ◆ I2C 时钟频率：400KHz
- ◆ 可按字节写
- ◆ 可按页写（16 字节/页）
- ◆ 内部写周期最大 5ms
- ◆ 可按字节读和顺序读
- ◆ 写操作次数 100 万次以上
- ◆ 数据保持时间 100 年以上
- ◆ 支持低功耗待机模式（Standby）
- ◆ 支持软件复位

### 附录4.2 管脚说明

管脚名称	类型	管脚说明
SCL	输入	时钟输入，在内部与 MCU 的 PB28 端口连接
SDA	输入/输出	数据输入/输出（开漏），在内部与 MCU 的 PB29 端口连接
WCB	输入	写使能，低有效，内部固定连接到 VSS
VDD	电源	电源，与 MCU 共用电源端口
VSS	地	地，与 MCU 共用地端口

### 附录4.3 结构框图

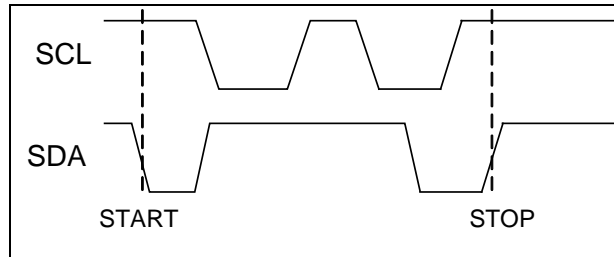


## 附录4.4 数据传输协议说明

内部 EEPROM 存储器与 MCU 的通信遵循 I2C 总线传输协议，总线上发送数据的器件为 MCU（主器件），接收数据的器件为内部 EEPROM（从器件），只有当总线处于空闲状态时才可以启动数据传输，每次数据传输均开始于 START 条件，结束于 STOP 条件，数据以字节为单位传输。

### ◆ 起始条件与停止条件

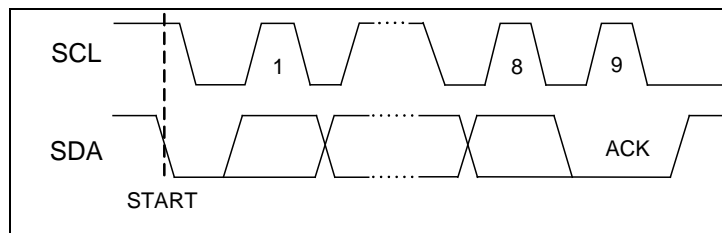
数据与时钟都为高则为总线处于空闲状态，当 SCL 为高电平时，SDA 的下降沿为起始条件，SDA 的上升沿为停止条件。如下图所示：



I2C 起始与停止条件时序图

### ◆ 应答

从器件每接收到一个字节就产生一个应答信号（低电平），因此主器件必须产生一个额外的时钟脉冲，即第 9 个脉冲。如下图所示：



I2C 应答时序图

## 附录4.5 低功耗待机模式（Standby）

内部 EEPROM 存储器芯片支持低功耗待机模式（Standby），EEPROM 在如下情况下均会进入待机模式：

- 上电后对 EEPROM 无任何读写访问操作
- 读 EEPROM 完成后收到停止位（STOP）
- 写 EEPROM 操作完成

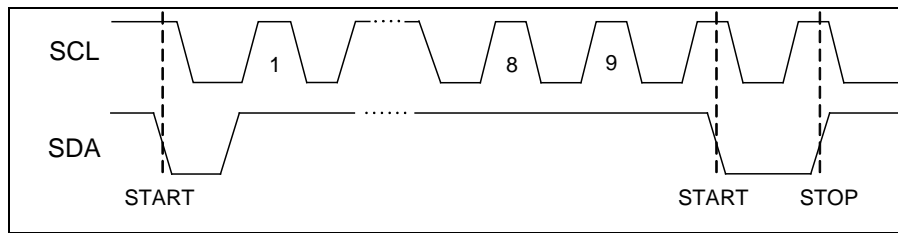
## 附录4.6 存储器软复位

在发生通信异常终止或者系统复位时，可以通过 I2C 总线对 EEPROM 芯片进行软复位，使后续通信恢复正常，具体复位步骤如下：

- (1) 产生一个起始位 START；
- (2) SDA 数据线保持为高，SCL 输入 9 个时钟；
- (3) 产生一个重启位 START 和停止位 STOP。

软复位时序图如下所示：

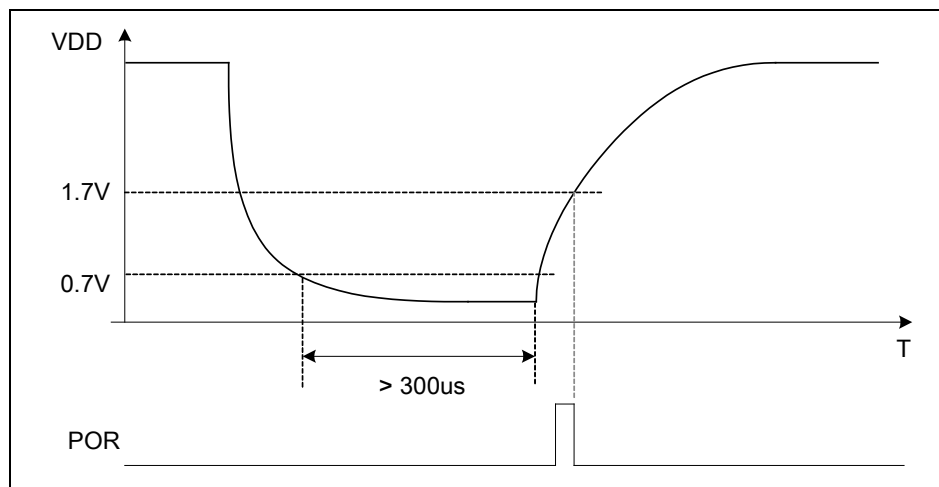




EEPROM 软复位时序图

## 附录4.7 存储器上电复位

EEPROM 芯片内置上电复位 POR，用于初始化内部电路状态，该复位电路生效的条件是 VDD 上电初始电压低于 0.7V，并且需维持至少 300us，然后再上电才能产生有效复位，当 VDD 上升到约 1.7V 时释放 POR 复位，如下图所示：



EEPROM 上电复位时序图

## 附录4.8 器件寻址

EEPROM 器件寻址字节共 8 位，其中最低位 (bit0) 为读写选择位 (1 为读, 0 为写)，高 5 位 (bit7~3) 为 EEPROM 的从器件地址，固定为 10100，后两位 (bit2~1) 为目标字节单元地址的高 2 位 (A9, A8)，与后续的 8 位字节地址 (A7~A0) 一起作为目标字节单元的 10 位地址 (A9~A0)，来访问 EEPROM 器件 1K 字节的存储空间。

具体器件地址 (DEVICE ADDRESS) 和字节地址 (BYTE ADDRESS) 格式如下表所示：

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
器件地址	1	0	1	0	0	A9	A8	R/W
字节地址	A7	A6	A5	A4	A3	A2	A1	A0

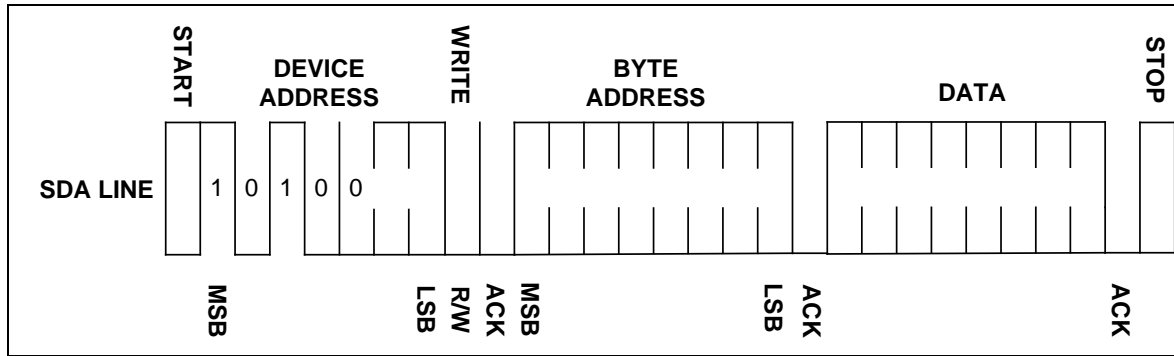
EEPROM 寻址表

## 附录4.9 EEPROM写操作

### ◆ 字节写操作

字节写操作时，主器件依次发送从器件地址，EEPROM 写操作单元的字节地址，写入的数据字节，从器件每次接收完一个通信字节后，均回复 ACK 应答，主器件发送完毕所写数据并收到 ACK 应答后，必须发送停止位 STOP，然后从器件才启动内部写周期，将数据写入到对应地址单元，并且直到写周期完成后，从器件才会响应新的访问操作。

字节写操作的通讯数据传输格式如下图所示：



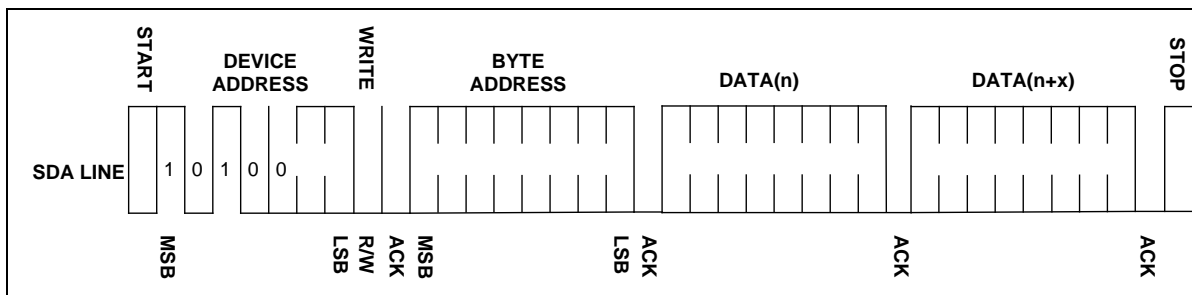
字节写时序图

◆ 页写操作

从器件内部 EEPROM 按每页 16 字节执行页写，页写初始化与字节写相同，只是主器件不会在第一个数据后发送停止条件，而是在内部 EEPROM 的应答后接着再发送数据，从器件 EEPROM 接收到每个数据后都应答“0”，最后仍然必须由主器件发送停止条件，终止写序列，然后从器件启动内部写周期，将数据写入到对应页。

接收到每个数据后，字节地址的低 4 位由内部硬件自动加 1，高位地址不变，维持在当前页内。当内部产生的字节地址达到该页边界地址时，字节地址将回到该页的首字节，随后的数据从该页的页首开始写，先前的数据被覆盖。

页写操作的通讯数据传输格式如下图所示：



页写时序图

◆ 应答查询

一旦内部写周期启动，从器件输入无效，不会响应新的访问操作，此时可启动应答查询。只有内部写周期完成，从器件才会应答“0”，之后可继续新的读写操作。

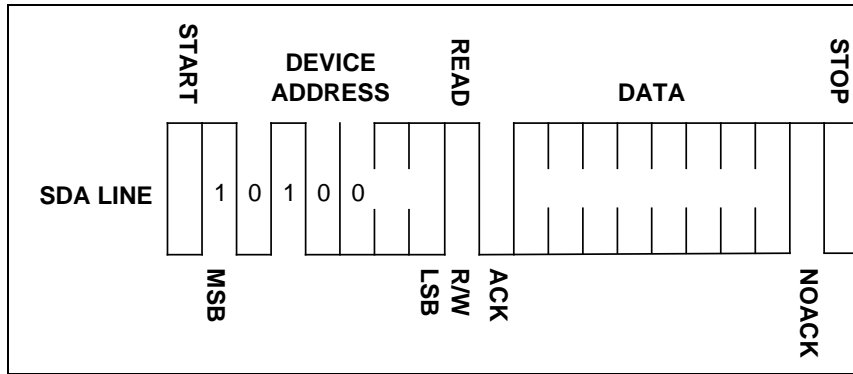
## 附录4.10 EEPROM读操作

读操作与写操作初始化相同，只是从器件地址中的读写选择位为“1”，读操作有三种方式：当前字节地址读、目标字节地址读、顺序读。

◆ 当前字节地址读

内部地址计数器保存着上次访问时的字节地址加 1 的值。只要芯片不掉电，该地址一直保存，当读到最后一页的最后一个字节地址时，地址会回转到 0，当写到某页尾的最后一个字节，地址会回到该页的首字节。EEPROM 接收到从器件地址（读写选择位为“1”），并应答 ACK 后，当前字节地址的数据就随后续时钟送出。主器件无需应答“0”，但需要发送停止条件。

当前字节地址读操作的通讯数据传输格式如下图所示：

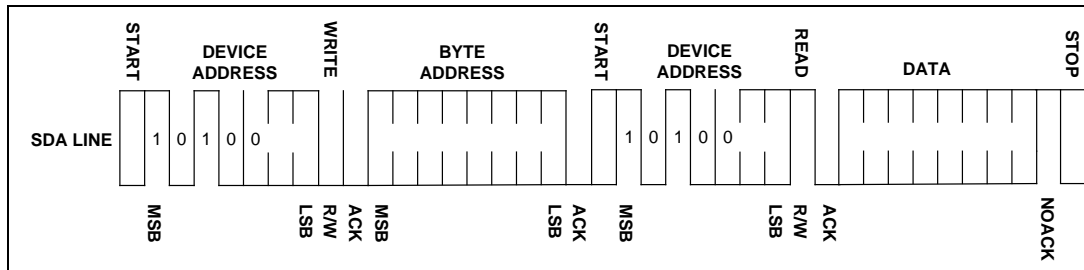


当前字节地址读时序图

◆ 目标字节地址读

目标字节地址读需先写一个目标字节地址，一旦从器件接收了器件地址和字节地址并应答了 ACK 后，主器件就产生一个重复的起始条件。主器件发送器件地址(读/写选择位为“1”)，从器件应答 ACK 后，目标字节地址的数据就随后续时钟送出数据。主器件无需应答“0”，但需发送停止条件。

目标字节地址读操作的通讯数据传输格式如下图所示：

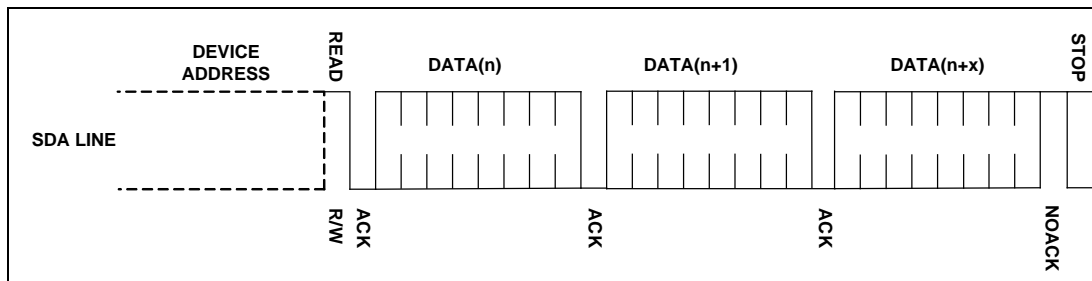


目标字节地址读时序图

◆ 顺序读

顺序读可通过“当前字节地址读”或“目标字节地址读”启动。主器件接收到一个数据后，应答 ACK，只要 EEPROM 接收到 ACK，将自动递增字节地址并继续随时钟发出后面的数据，若达到 EEPROM 存储器地址末尾时，地址自动回转到 0，仍可继续读取数据。主器件不应答“0”，而发送停止条件，即可结束顺序读操作。

顺序读操作的通讯数据传输格式如下图所示（以当前字节地址读方式启动）：



顺序读时序图（以当前字节地址读方式启动）

## 附录4.11 内置EEPROM电气特性

### 附录4.11.1 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 ~ 6.25	V
芯片存储温度	T <sub>STG</sub>	—	-55 ~ 150	°C

◆ 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	°C
芯片工作电压	VDD	—	2.2	5.5	V

◆ 芯片上电和下电工作条件表 (-40 ~ 85°C)

参数	符号	最小值	最大值	单位
VDD 上电初始电压	V <sub>start</sub>	0	0.7	V
VDD 上升速率	T <sub>VDD</sub>	20	500000	us/V

### 附录4.11.2 直流电气特性表

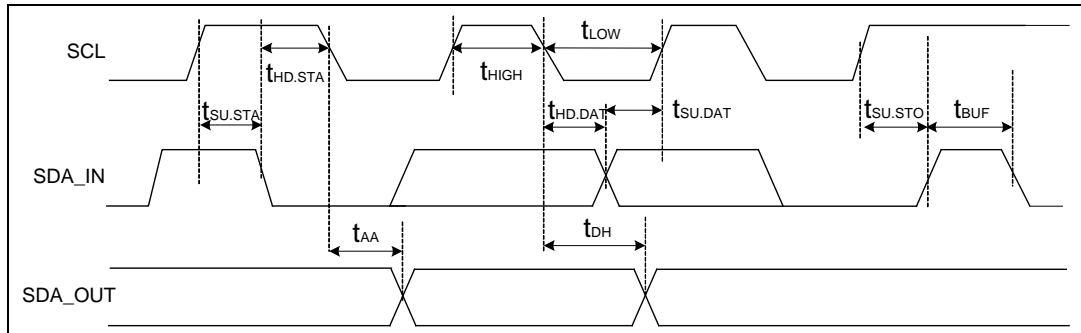
(条件: T<sub>A</sub>= -40°C ~ 85°C, VDD= 2.2V ~ 5.5V)

参数	符号	最小值	典型值	最大值	单位	测试条件
运行电流	I <sub>DD</sub>	—	0.2	0.6	mA	VDD =5V 400kHz 读
		—	1	2	mA	
待机电流	I <sub>SB</sub>	—	1	6	uA	VDD =5V

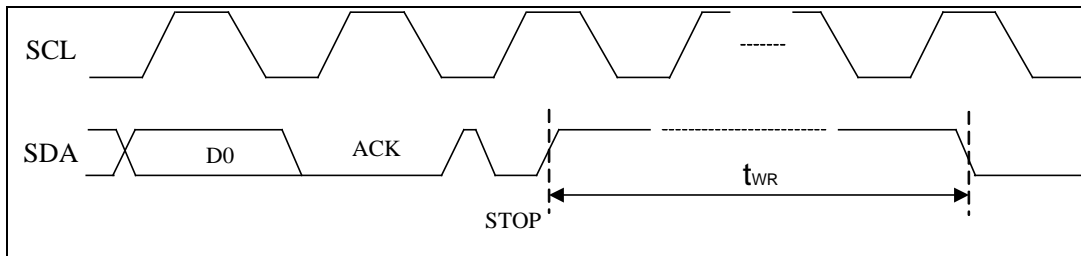
### 附录4.11.3 交流电气特性表

(条件: T<sub>A</sub>= -40°C ~ 85°C, VDD= 2.2V ~ 5.5V, C<sub>L</sub>=100pF)

参数	符号	2.2V ≤ VDD ≤ 5.5V		2.5V ≤ VDD ≤ 5.5V		单位
		最小值	最大值	最小值	最大值	
SCL 时钟频率	f <sub>SCL</sub>	—	400	—	400	KHz
时钟低电平宽度	f <sub>LOW</sub>	1.3	—	0.4	—	us
时钟高电平宽度	f <sub>HIGH</sub>	0.6	—	0.4	—	us
噪声消除时间	t <sub>i</sub>	—	0.1	—	0.05	us
数据输出间隔时间	t <sub>AA</sub>	0.05	0.9	0.05	0.55	us
总线释放时间	t <sub>BUF</sub>	1.3	—	0.5	—	us
起始条件保持时间	t <sub>HD.STA</sub>	0.6	—	0.25	—	us
起始条件建立时间	t <sub>SU.STA</sub>	0.6	—	0.25	—	us
数据输入保持时间	t <sub>HD.DAT</sub>	0	—	0	—	us
数据输入建立时间	t <sub>SU.DAT</sub>	0.1	—	0.1	—	us
停止条件建立时间	t <sub>SU.STO</sub>	0.6	—	0.25	—	us
数据输出保持时间	t <sub>DH</sub>	0.05	—	0.05	—	us
内部写周期	t <sub>WR</sub>	—	5	—	5	ms



I2C 通讯总线时序图



EEPROM 内部写周期时序图

#### 附录4. 11. 4 可靠性参数表

参数	符号	最小值	最大值	单位	测试条件
写次数 (Endurance)	EDR	1000000	—	次	VDD = 3.3V, 页写模式
数据保持时间 (Data retention)	DRET	100	—	年	T <sub>A</sub> = 55°C

注：上述可靠性参数为理论设计值，不是批量产品测试值，仅供芯片应用时参考。