

Sub1G SoC
ES32W0030

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 5 月 4 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

Part NO.	工作电压	FLASH	SRAM	I/O	Timer	UART	SPI	I2C	ADC	封装类型
ES32W0030FLNQ	2.2V ~3.6V	128KB	16KB	30	16-bit×6, 32-bit×1	6	1	1	12-bit×18ch	QFN48

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
V1.0	2023-05-04	初版

目 录

内容目录

第 1 章	芯片简介.....	19
1.1	概述.....	19
1.1.1	芯片特性.....	19
1.1.2	MCU 特性.....	19
1.1.3	RF 收发器特性.....	22
1.2	系统框图.....	24
1.3	管脚分配图.....	25
1.3.1	QFN48 封装.....	25
1.4	管脚说明.....	26
1.4.1	管脚说明.....	26
1.4.2	MCU 复用管脚对照.....	27
1.4.3	内部连线.....	29
1.4.4	MCU 内部悬空脚注意事项.....	30
第 2 章	系统控制及操作特性.....	31
2.1	系统控制保护.....	31
2.1.1	概述.....	31
2.1.2	特殊功能寄存器.....	31
2.2	系统电源.....	32
2.2.1	结构框图.....	32
2.2.2	芯片供电电源.....	32
2.3	系统复位.....	32
2.3.1	概述.....	32
2.3.2	结构框图.....	33
2.3.3	复位时序图.....	33
2.3.4	外部复位 MRSTN 参考.....	34
2.3.5	外设模块复位控制.....	35
2.3.6	特殊功能寄存器.....	36
2.4	低电压监测 (LVD).....	40
2.4.1	概述.....	40
2.4.2	特殊功能寄存器.....	40
2.5	系统低功耗操作模式.....	42
2.5.1	概述.....	42
2.5.2	浅睡眠模式.....	42
2.5.3	深度睡眠模式.....	42
2.5.4	睡眠模式的唤醒.....	43
2.5.5	睡眠模式的唤醒时间.....	43
2.5.6	FLASH 存储器等待功能.....	44
2.5.7	特殊功能寄存器.....	44
2.6	系统时钟.....	45
2.6.1	概述.....	45
2.6.2	结构框图.....	46

2.6.3	功能说明	46
2.6.4	特殊功能寄存器.....	51
2.6.5	系统时钟应用说明.....	59
2.7	中断和异常处理.....	60
2.7.1	中断和异常.....	60
2.7.2	中断和异常向量的分配.....	62
2.7.3	中断向量表的重映射	63
2.7.4	特殊功能寄存器.....	63
2.8	系统控制块 (SCB)	76
2.8.1	概述.....	76
2.8.2	特殊功能寄存器.....	76
2.9	系统定时器 (SYSTICK)	81
2.9.1	概述.....	81
2.9.2	特殊功能寄存器.....	82
2.9.3	配置字软件控制.....	84
2.10	定时器 (T16N/T32N) 同步启动关停控制	85
2.10.1	概述.....	85
2.10.2	特殊功能寄存器.....	85
2.11	模块调试相关配置	87
2.11.1	概述.....	87
2.11.2	特殊功能寄存器.....	87
第 3 章	存储器资源	89
3.1	内部存储器地址映射.....	89
3.2	FLASH 存储器.....	90
3.2.1	信息区 FLASH	90
3.2.2	程序区 FLASH	93
3.3	功能描述.....	94
3.3.1	IAP 概述.....	94
3.3.2	Flash 保护.....	95
3.3.3	Flash 程序区全擦除	96
3.3.4	Flash 页擦除	96
3.3.5	Flash 字编程.....	97
3.3.6	Flash 编程数据 FIFO	98
3.3.7	IAP 自编程硬件固化模块	98
3.3.8	特殊功能寄存器.....	100
3.4	数据存储器 (SRAM)	105
3.4.1	SRAM 地址映射	105
3.4.2	SRAM 位带扩展	105
3.5	外设寄存器	106
3.5.1	外设寄存器映射.....	106
3.5.2	外设寄存器位带扩展.....	106
3.5.3	系统控制单元 (SCU) 寄存器列表.....	107
3.5.4	CRC 寄存器列表.....	108
3.5.5	GPIO 寄存器列表.....	108

3.5.6	IAP 寄存器列表	109
3.5.7	ADC 寄存器列表	110
3.5.8	IWDT 寄存器列表	110
3.5.9	WWDT 寄存器列表	111
3.5.10	T16N0/T16N1/T16N2/T16N3/T16N4/T16N5 寄存器列表	111
3.5.11	T32N0 寄存器列表	112
3.5.12	UART0/UART1/UART2/UART3/UART4/UART5 寄存器列表	112
3.5.13	SPI0 寄存器列表	113
3.5.14	I2C0 寄存器列表	113
3.6	内核寄存器	114
3.6.1	系统定时器 (SYSTICK) 寄存器列表	114
3.6.2	中断控制器 (NVIC) 寄存器列表	114
3.6.3	系统控制块 (SCB) 寄存器列表	114
第 4 章	输入输出端口 (GPIO)	115
4.1	概述	115
4.2	结构框图	116
4.3	外部端口中断	117
4.4	外部按键中断	118
4.5	Buzz 输出	119
4.6	特殊功能寄存器	120
4.6.1	PA 端口状态寄存器 (GPIO_PAPORT)	120
4.6.2	PA 端口数据寄存器 (GPIO_PADATA)	120
4.6.3	PA 端口输出置位寄存器 (GPIO_PADATABSR)	121
4.6.4	PA 端口输出清零寄存器 (GPIO_PADATABCR)	121
4.6.5	PA 端口输出翻转寄存器 (GPIO_PADATABRR)	122
4.6.6	PA 端口方向控制寄存器 (GPIO_PADIR)	122
4.6.7	PA 端口方向置位寄存器 (GPIO_PADIRBSR)	123
4.6.8	PA 端口方向清零寄存器 (GPIO_PADIRBCR)	123
4.6.9	PA 端口方向翻转寄存器 (GPIO_PADIRBRR)	124
4.6.10	PA<7:0>端口复用选择寄存器 (GPIO_PAFUNC0)	125
4.6.11	PA<15:8>端口复用选择寄存器 (GPIO_PAFUNC1)	126
4.6.12	PA<23:16>端口复用选择寄存器 (GPIO_PAFUNC2)	127
4.6.13	PA<31:24>端口复用选择寄存器 (GPIO_PAFUNC3)	128
4.6.14	PA 端口输入控制寄存器 (GPIO_PAINEB)	129
4.6.15	PA 端口开漏控制寄存器 (GPIO_PAODE)	129
4.6.16	PA 端口弱上拉使能寄存器 (GPIO_PAPUE)	130
4.6.17	PA 端口弱下拉使能寄存器 (GPIO_PAPDE)	130
4.6.18	PA 端口驱动电流控制寄存器 (GPIO_PADS)	131
4.6.19	PA 端口类型选择寄存器 (GPIO_PATYP)	131
4.6.20	PA 端口滤波控制寄存器 (GPIO_PAFLT)	131
4.6.21	PB 端口状态寄存器 (GPIO_PBPORT)	132
4.6.22	PB 端口数据寄存器 (GPIO_PBDATA)	132
4.6.23	PB 端口输出置位寄存器 (GPIO_PBDATABSR)	133
4.6.24	PB 端口输出清零寄存器 (GPIO_PBDATABCR)	133

4.6.25	PB 端口输出翻转寄存器 (GPIO_PBDATABRR)	134
4.6.26	PB 端口方向控制寄存器 (GPIO_PBDIR)	134
4.6.27	PB 端口方向置位寄存器 (GPIO_PBDIRBSR)	135
4.6.28	PB 端口方向清零寄存器 (GPIO_PBDIRBCR)	135
4.6.29	PB 端口方向翻转寄存器 (GPIO_PBDIRBRR)	136
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)	143
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)	144
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)	145
4.6.44	PINT8~11 中断源选择寄存器 (GPIO_PINTSEL1)	146
4.6.45	PINT0~7 中断配置寄存器 (GPIO_PINTCFG0)	147
4.6.46	PINT8~11 中断配置寄存器 (GPIO_PINTCFG1)	148
4.6.47	KINT 中断使能寄存器 (GPIO_KINTIE)	149
4.6.48	KINT 中断标志寄存器 (GPIO_KINTIF)	149
4.6.49	KINT0~7 中断源选择寄存器 (GPIO_KINTSEL0)	150
4.6.50	KINT8~11 中断源选择寄存器 (GPIO_KINTSEL1)	151
4.6.51	KINT0~7 中断配置寄存器 (GPIO_KINTCFG0)	152
4.6.52	KINT8~11 中断配置寄存器 (GPIO_KINTCFG1)	153
4.6.53	脉宽调制寄存器 (GPIO_TXPWM)	154
4.6.54	BUZ 控制寄存器 (GPIO_BUZC)	156
第 5 章	外设	157
5.1	定时器/计数器	157
5.1.1	16 位定时器/计数器 T16N (T16N0~5)	157
5.1.2	32 位定时器/计数器 T32N (T32N0)	175
5.2	通用异步接收/发送器 (UART0~5)	190
5.2.1	概述	190
5.2.2	结构框图	191
5.2.3	UART 数据格式	191
5.2.4	UART 异步发送器	192
5.2.5	UART 异步接收器	194
5.2.6	UART 发送调制功能	197
5.2.7	UART 红外唤醒功能	198
5.2.8	UART 端口极性	198
5.2.9	UART 单线半双工通信	198

5.2.10	特殊功能寄存器.....	199
5.2.11	UART 应用说明.....	208
5.3	SPI 同步串口通讯控制器 (SPI0)	209
5.3.1	概述.....	209
5.3.2	结构框图	209
5.3.3	SPI 通讯模式.....	209
5.3.4	SPI 数据格式.....	209
5.3.5	SPI 帧位宽	211
5.3.6	SPI 同步发送器.....	211
5.3.7	SPI 同步接收器.....	213
5.3.8	SPI 通讯控制.....	214
5.3.9	SPI 延迟接收功能	215
5.3.10	SPI 数据帧发送间隔功能	215
5.3.11	特殊功能寄存器.....	216
5.3.12	SPI 应用说明.....	223
5.4	I2C 总线串口通讯控制器 (I2C0)	224
5.4.1	概述.....	224
5.4.2	结构框图	224
5.4.3	I2C 总线基本原理.....	224
5.4.4	I2C 通讯端口配置.....	226
5.4.5	I2C 时基定时器与 16 倍速采样器	227
5.4.6	I2C 通讯发送器	228
5.4.7	I2C 通讯接收器.....	230
5.4.8	I2C 通讯控制.....	231
5.4.9	特殊功能寄存器.....	236
5.4.10	I2C 应用说明.....	245
5.5	模数转换器 (ADC)	246
5.5.1	概述.....	246
5.5.2	结构框图	246
5.5.3	ADC 基本配置.....	246
5.5.4	ADC 高精度参考电压.....	246
5.5.5	ADC 数据转换.....	247
5.5.6	自动转换比较功能.....	249
5.5.7	特殊功能寄存器.....	251
5.6	硬件独立看门狗 (IWDT)	260
5.6.1	概述.....	260
5.6.2	特殊功能寄存器.....	262
5.7	窗口看门狗 (WWDT)	266
5.7.1	概述.....	266
5.7.2	特殊功能寄存器.....	268
5.8	循环冗余校验 (CRC)	272
5.8.1	概述.....	272
5.8.2	运用说明	272
5.8.3	特殊功能寄存器.....	274

第 6 章	RF 收发器—芯片状态控制	279
6.1	POWER DOWN	279
6.2	DEEP SLEEP	279
6.3	SLEEP (RC)	280
6.4	IDLE	280
6.5	TX	280
6.6	RX	280
6.7	芯片状态切换说明	280
第 7 章	RF 收发器—帧结构	282
7.1	增强型帧结构	282
7.2	直接 FIFO 帧结构	283
第 8 章	RF 收发器—多 PIPE 逻辑通道	284
第 9 章	RF 收发器—自动 ACK 与 ART	285
第 10 章	RF 收发器—数据与控制接口	286
10.1	FIFO	286
10.1.1	FIFO 指针	286
10.1.2	FIFO 操作说明	287
10.2	中断	289
10.2.1	增强型帧结构中中断	291
10.2.2	直接 FIFO 帧结构中中断	294
10.3	SPI 通信接口	295
10.3.1	SPI 帧格式	295
10.3.2	寄存器访问时序	295
10.3.3	FIFO 访问时序	295
10.3.4	SPI 时序参数	296
10.4	GPIO 配置	297
第 11 章	RF 收发器—频点与数据传输参数设置	298
11.1	频点设置	298
11.1.1	频点默认设置模式	298
11.1.2	频点直接设置模式	299
11.1.3	跳频接收模式	299
11.2	数据传输参数设置	300
11.2.1	数据速率设置	300
11.2.2	发送调制特性设置	300
11.2.3	发送功率设置	301
11.2.4	接收滤波器带宽设置	301
第 12 章	RF 收发器—收发操作流程	302
12.1	增强型帧结构收发流程	302
12.2	直接 FIFO 帧结构收发流程	304
12.3	直接收发 DIRECT 模式	306
12.4	连续发送模式	307
第 13 章	RF 收发器—唤醒	308
13.1	Wake-up Timer 唤醒	308
13.2	GPIO 外部中断唤醒	308

第 14 章	RF 收发器—RC 校准.....	309
14.1	软件校准	309
14.2	硬件自动校准.....	309
第 15 章	RF 收发器—异常状态的检测及复位.....	310
15.1	芯片异常状态检测	310
15.2	寄存器与 FIFO 校验功能	310
15.3	芯片复位	310
第 16 章	RF 收发器—其它功能与设置	311
16.1	白化与 CRC.....	311
16.2	FEC	311
16.3	数据编码	312
16.4	RSSI	312
16.5	载波检测 (Carrier Detect)	312
16.6	AFC 及晶振的校准功能	312
16.7	有效/无效前导检测	313
16.8	有效/无效 SFD 检测.....	313
16.9	低功耗接收模式	313
第 17 章	RF 收发器—寄存器	314
17.1	寄存器列表	314
17.2	寄存器说明	316
17.2.1	RFID 寄存器.....	316
17.2.2	TRCTRL 寄存器	316
17.2.3	PKTCTRL 寄存器.....	317
17.2.4	PKTCFG0 寄存器	318
17.2.5	CRCSEED 寄存器	318
17.2.6	PKTCFG1 寄存器	318
17.2.7	FIFOTHRES 寄存器.....	319
17.2.8	TXPHR0 寄存器	319
17.2.9	TXPHR1 寄存器.....	319
17.2.10	PSDULEN 寄存器	320
17.2.11	RXPHR0 寄存器	320
17.2.12	RXPHR1 寄存器	320
17.2.13	FIFOCTRL 寄存器	321
17.2.14	LEN0PKLEN 寄存器	322
17.2.15	FIFOSTA 寄存器	322
17.2.16	INT 寄存器	323
17.2.17	INTIE 寄存器.....	324
17.2.18	INTIC 寄存器.....	325
17.2.19	CRCVAL 寄存器.....	325
17.2.20	PIPECTRL 寄存器	326
17.2.21	P0ADDR0 寄存器	326
17.2.22	P0ADDR1 寄存器	326
17.2.23	P0ADDR2 寄存器	326
17.2.24	P1ADDR0 寄存器	327

17. 2. 25	P1ADDR1 寄存器	327
17. 2. 26	P1ADDR2 寄存器	327
17. 2. 27	P23ADDR 寄存器	327
17. 2. 28	CHIPSTA0 寄存器.....	328
17. 2. 29	MODECTRL 寄存器.....	328
17. 2. 30	GPIOCFG0 寄存器	329
17. 2. 31	GPIOCFG1 寄存器	331
17. 2. 32	WDTCFG0 寄存器	332
17. 2. 33	WDTCFG1 寄存器	332
17. 2. 34	WDTWDV 寄存器	333
17. 2. 35	AGCGAIN 寄存器	333
17. 2. 36	RSSI 寄存器.....	333
17. 2. 37	PREACFG 寄存器.....	333
17. 2. 38	MODEMCTRL 寄存器.....	334
17. 2. 39	IFSET 寄存器	334
17. 2. 40	CDTH 寄存器	334
17. 2. 41	CHCFG0 寄存器	335
17. 2. 42	CHCFG1 寄存器	335
17. 2. 43	CHCFG2 寄存器	335
17. 2. 44	CHCFG3 寄存器	335
17. 2. 45	DEVIATION 寄存器.....	336
17. 2. 46	HOPCFG 寄存器.....	336
17. 2. 47	filterband 寄存器.....	336
17. 2. 48	FREQCFG0 寄存器	337
17. 2. 49	FREQCFG1 寄存器	337
17. 2. 50	FREQCFG2 寄存器	337
17. 2. 51	SYMRATE0 寄存器.....	338
17. 2. 52	SYMRATE1 寄存器.....	338
17. 2. 53	RFCFG 寄存器.....	338
17. 2. 54	CHIPSTA1 寄存器.....	339
17. 2. 55	XOSC 寄存器.....	339
17. 2. 56	RCCFG0 寄存器	340
17. 2. 57	RCSTA 寄存器	340
17. 2. 58	RCCFG1 寄存器	341
17. 2. 59	HOPSPACE0 寄存器	341
17. 2. 60	HOPSPACE1 寄存器	342
17. 2. 61	RSSICFG 寄存器	342
17. 2. 62	HOPCH 寄存器.....	342
17. 2. 63	PACFG 寄存器	343
17. 2. 64	ACKCFG 寄存器.....	343
17. 2. 65	FIFOPTR 寄存器.....	343
17. 2. 66	LEN0RXADD 寄存器	344
17. 2. 67	INVSFDTIME 寄存器	344
17. 2. 68	SFTRST0 寄存器	344

17. 2. 69	SFTRST1 寄存器	345
17. 2. 70	REGCHECK 寄存器	345
17. 2. 71	FIFODATA 寄存器	345
第 18 章	封装尺寸图	346
18. 1	封装尺寸图	346
第 19 章	典型应用原理图	347
19. 1	典型应用原理图	347
19. 2	设计应用注意事项	348
第 20 章	Cortex-M0 内核描述	349
20. 1	Cortex-M0 指令集	349
20. 2	Cortex-M0 内核寄存器	351
20. 2. 1	通用寄存器 R0~R12	351
20. 2. 2	堆栈指针寄存器 SP (R13)	351
20. 2. 3	链接寄存器 LR (R14)	352
20. 2. 4	程序计数器 PC (R15)	352
20. 2. 5	程序状态寄存器 xPSR	352
20. 2. 6	异常/中断屏蔽寄存器 PRIMASK	353
20. 2. 7	控制寄存器 CONTROL	353
第 21 章	电气特性	355
21. 1	MCU 电气特性	355
21. 1. 1	最大标称值	355
21. 1. 2	MCU 工作条件	355
21. 1. 3	MCU 功能模块工作电压范围	355
21. 1. 4	芯片上电和下电工作条件表 (-40 ~ 85°C)	356
21. 1. 5	芯片功耗参数测量方法	356
21. 1. 6	芯片 IO 端口参数测量方法	356
21. 1. 7	MCU 功耗特性参数	356
21. 1. 8	MCU 功能模块功耗特性参数表	357
21. 1. 9	MCU 芯片输入端口特性	359
21. 1. 10	MCU 芯片输出端口特性	360
21. 1. 11	MCU 系统时钟规格	360
21. 1. 12	MCU 芯片 ESD 特性	360
21. 1. 13	ADC 模块特性	361
21. 1. 14	ADC 转换时钟源选择	361
21. 1. 15	ADC 内部参考和 1/4VDD 电源分压电压特性表	362
21. 1. 16	内部时钟源特性	362
21. 1. 17	BOR 模块特性	363
21. 1. 18	LVD 模块特性	363
21. 2	MCU 参数特性图	364
21. 2. 1	MCU 功耗特性	364
21. 2. 2	MCU IO 端口输入特性	366
21. 2. 3	MCU IO 端口输出特性 (普通驱动)	367
21. 2. 4	MCU IO 端口输出特性 (增强驱动)	370
21. 3	RF 收发器电气特性	373

21.3.1	功耗参数	373
21.3.2	接收机 (RX) 参数	374
21.3.3	发射机 (TX) 参数	375
21.3.4	频率合成器参数	376
21.3.5	振荡器参数	377
第 22 章	编程调试接口	378
22.1	概述	378
22.2	ISP 编程接口	378
22.2.1	通信协议	378
22.2.2	操作流程	379
22.3	SWD 调试接口	379
22.3.1	概述	379
22.3.2	SWD 特性	380

图目录

图 1-1	系统内部结构框图	24
图 1-2	QFN48 封装顶视图 (TopView)	25
图 2-1	系统电源结构框图	32
图 2-2	系统复位电路结构框图	33
图 2-3	上电复位时序示意图	33
图 2-4	掉电复位时序示意图	34
图 2-5	外部复位时序示意图	34
图 2-6	MRSTN 复位参考电路图 1	34
图 2-7	MRSTN 复位参考电路图 2	34
图 2-8	MRSTN 复位参考电路图 3	35
图 2-9	MRSTN 复位参考电路图 4	35
图 2-10	LVD 低电压监测示意图	40
图 2-11	系统时钟电路结构框图	46
图 2-12	XTAL 振荡器电路结构示意图	47
图 2-13	晶体振荡器等效电路示意图	47
图 2-14	系统定时器框图	81
图 3-1	内部存储系统分配示意图	89
图 3-2	Boot Flash 启动地址分配示意图	94
图 3-3	SRAM 映射图	105
图 3-4	外设存储器分配示意图	106
图 4-1	IO 端口电流结构图	116
图 4-2	外部端口中断 PINT0 电路结构示意图	117
图 4-3	外部按键中断 KINT 电路结构示意图	118
图 4-4	Buzz 高电平调制输出波形图	119
图 4-5	Buzz 低电平调制输出波形图	119
图 5-1	T16N0 电路结构框图	158
图 5-2	T16N 计数匹配功能示意图	159
图 5-3	T16N0 捕捉功能示意图	161
图 5-4	T16N 输出调制功能示意图	163
图 5-5	T32N 电路结构框图	175
图 5-6	T32N 计数匹配功能示意图	177
图 5-7	T32N 捕捉功能示意图	178
图 5-8	T32N 输出调制功能示意图	180
图 5-9	UART 电路结构图	191
图 5-10	UART 7 位数据格式	191
图 5-11	UART 8 位数据格式	191
图 5-12	UART 9 位数据格式	191
图 5-13	UART 发送数据流示意图	192
图 5-14	UART 发送中断标志示意图	193
图 5-15	UART 发送数据操作流程图示例	194
图 5-16	UART 接收数据流示意图	195
图 5-17	UART 接收中断标志示意图	196
图 5-18	UART 接收数据操作流程图示例	197

图 5-19	高电平调制输出波形图.....	198
图 5-20	低电平调制输出波形图.....	198
图 5-21	SPI 电路结构框图	209
图 5-22	SPI 时钟上升沿发送，下降沿接收波形示意图.....	210
图 5-23	SPI 时钟下降沿发送，上升沿接收波形示意图.....	210
图 5-24	SPI 时钟上升沿接收，下降沿发送波形示意图.....	210
图 5-25	SPI 时钟下降沿接收，上升沿发送波形示意图.....	211
图 5-26	SPI 发送数据流示意图	212
图 5-27	SPI 发送中断标志示意图.....	212
图 5-28	SPI 接收数据流示意图	213
图 5-29	SPI 接收中断标志示意图.....	214
图 5-30	SPI 延迟接收功能波形示意图	215
图 5-31	I2C 电路结构框图.....	224
图 5-32	I2C 总线通讯协议示意图.....	225
图 5-33	I2C 主控器写入从动器数据示意图	226
图 5-34	I2C 主控器读取从动器数据示意图	226
图 5-35	开漏输出端口示意图	227
图 5-36	I2C 总线端口信号的波形示意图.....	228
图 5-37	I2C 发送数据流示意图	228
图 5-38	I2C 发送中断标志示意图.....	229
图 5-39	I2C 接收数据流示意图	230
图 5-40	I2C 接收中断标志示意图.....	231
图 5-41	I2C 起始位波形图.....	232
图 5-42	I2C 自动寻呼波形图	232
图 5-43	I2C 停止位波形图.....	233
图 5-44	I2C 应答延迟功能波形示意图	234
图 5-45	I2C 数据帧传输间隔功能波形示意图	234
图 5-46	I2C 时钟线下拉等待波形示意图.....	235
图 5-47	ADC 内部结构图	246
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)	247
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)	248
图 5-50	看门狗时序图	261
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%)	267
图 5-52	错误的喂狗时序图 (WWDTWIN 设定为 25%)	267
图 6-1	RF 收发器状态控制示意图.....	279
图 6-2	芯片上电进入发送模式示意图.....	281
图 6-3	芯片上电进入接收模式示意图.....	281
图 7-1	增强型帧结构示意图	282
图 7-2	直接 FIFO 帧结构示意图.....	283
图 8-1	4 路 PIPE 通讯示意图	284
图 10-1	FIFO 控制示意图.....	286
图 10-2	两种帧结构模式下 FIFO 结构.....	287
图 10-3	直接 FIFO 帧结构发送半空操作示意图.....	288
图 10-4	直接 FIFO 帧结构接收半满操作示意图.....	288

图 10-5	IRQ 中断输出示意图	290
图 10-6	ACK 不使能情况中断示意图	291
图 10-7	ACK 使能正常通讯中断示意图	291
图 10-8	PTX 自动重传情况 1 中断示意图	292
图 10-9	PTX 自动重传情况 2 中断示意图	292
图 10-10	PTX 自动重传情况 3 中断示意图(重传超时, 重传次数为 2)	292
图 10-11	PRX 自动重收中断示意图	293
图 10-12	ACK 使能情况下 NOACK 功能中断示意图	293
图 10-13	直接 FIFO 帧结构中断示意图	294
图 10-14	SPI 帧格式	295
图 10-15	SPI 写寄存器时序	295
图 10-16	SPI 读寄存器时序	295
图 10-17	SPI 写 FIFO 时序	296
图 10-18	SPI 读 FIFO 时序	296
图 10-19	SPI 时序参数示意图	296
图 12-1	PTX 发送流程图	302
图 12-2	PRX 接收流程图	303
图 12-3	直接 FIFO 帧结构 LENO_TXMODE='1'发送停止示意图	304
图 12-4	直接 FIFO 帧结构 LENO_TXMODE='0'发送停止示意图	304
图 12-5	帧长信息配置示意图	305
图 12-6	直接发送模式时序图	306
图 12-7	直接接收模式时序图	306
图 12-8	直接收发模式数据时序示意图	307
图 16-1	PN9 数据白化生成器	311
图 16-2	CRC 硬件实现示意图	311
图 18-1	QFN48 封装尺寸图	346
图 19-1	典型应用原理图	347
图 20-1	内核寄存器	351
图 20-2	堆栈操作示意图	352

表目录

表 1-1	管脚说明表.....	27
表 1-2	管脚对照图.....	29
表 1-3	内部连线.....	29
表 1-4	芯片悬空管脚设置说明表.....	30
表 2-1	低功耗模式时钟状态表.....	42
表 2-2	异常/中断优先级操作类型说明表.....	60
表 2-3	异常/中断优先级列表.....	61
表 4-1	PINT 选择分组对应列表.....	117
表 4-2	KINT 选择分组对应列表.....	118
表 5-1	I2C 总线端口信号的时序参数列表.....	227
表 5-2	ADC 精度与转换速率的对应关系列表.....	247
表 7-1	标准识别号 (ACK 不使能).....	282
表 7-2	标准识别号 (ACK 使能).....	283
表 10-1	推荐 SPI 时序 (8MHz) 参数.....	296
表 10-2	GPIO 默认状态.....	297
表 11-1	20MHz 晶振下各频段范围与配置.....	298
表 11-2	26MHz 晶振下各频段范围与配置.....	298
表 11-3	系数 k_{freq} 配置表.....	299
表 11-4	系数 k_{dev} 配置表.....	300
表 15-1	CHIPSTA1 各状态说明.....	310
表 17-1	寄存器列表.....	315
表 19-1	不同频率下的射频元件 BOM.....	347
表 21-1	芯片功耗.....	373
表 21-2	接收机 (RX) 特性.....	374
表 21-3	发射机 (TX) 特性.....	375
表 21-4	频率合成器特性.....	376
表 21-5	低功耗 RC 振荡器特性.....	377
表 21-6	石英晶体振荡器特性.....	377

第 1 章 芯片简介

1.1 概述

ES32W0030 是一款高集成度的低功耗 Sub-1GHz 无线 SoC 芯片，可广泛应用于无线抄表、无线门禁、智能家居、工业控制、无线传感器网络、无线遥控、遥测设备等领域。片上集成高性能、低功耗的 RF 收发器和 32 位 MCU。

片上 RF 收发器的外围电路较简单，只需少量的外围被动器件即可构成完整的 Sub-1GHz 无线收发系统。RF 收发器发射功率最大可以到 20dBm，接收灵敏度可达到 -114dBm@10Kbps (433MHz)。

片上 MCU 内部集成 32 位 ARM Cortex-M0 CPU 内核，具备 128KB 的 Flash 程序存储空间。

1.1.1 芯片特性

◆ 工作条件

- ◇ 工作电压范围：2.2V~3.6V
- ◇ 工作温度范围：-40°C~85°C

◆ 封装形式

- ◇ QFN48

1.1.2 MCU特性

◆ 工作特性

- ◇ 工作主时钟频率：32KHz，400KHz~48MHz
- ◇ 工作电流：I_{vdd} = 4mA (@内部 HRC 16MHz，典型值)
- ◇ 待机电流：I_{vdd} = 2.8μA (常温，典型值)

◆ 电源

- ◇ 系统电源输入 VDD，支持工作电压为 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%)
- ◇ 内部 32KHz 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
 - ◇ 128K 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
 - ◇ 16K 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 30 个双向 I/O 端口
 - PA 端口 (PA0~PA15, PA17~PA24)
 - PB 端口 (PB16~PB19, PB24~PB25)
 - ◇ 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 18 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

1.1.3 RF收发器特性

- ◆ 频带
 - ◇ 315/433/779/868/915MHz
- ◆ 接收灵敏度
 - ◇ -114 dBm @10kbps(315/433MHz)
 - ◇ -112dBm @10kbps(779/868/915MHz)
- ◆ 调制方式
 - ◇ (G)FSK
- ◆ 最大输出功率
 - ◇ +20 dBm
- ◆ 低电流消耗
 - ◇ 8 mA RX(低功耗接收模式)
 - ◇ 18 mA RX(非低功耗接收模式)
- ◆ 各模式下功耗
 - ◇ 100 nA POWER DOWN
 - ◇ 3 μ A DEEP SLEEP
 - ◇ 3.5 μ A SLEEP(RC)
 - ◇ 2 mA IDLE
- ◆ 数据速率：1.2 Kbps to 100 Kbps
- ◆ 支持自动跳频功能
- ◆ 8 个可配通道
- ◆ 选择性性能
 - ◇ 邻道抑制：41 dB @ 200KHz
 - ◇ 阻断性能：55 dB @ 1MHz
- ◆ 独立 256 字节发送与接收 FIFO
- ◆ 支持自动应答功能(ACK)
- ◆ 支持自动重传功能(ART)

- ◆ 支持自动频率控制(AFC)
- ◆ 支持自动增益控制(AGC)
- ◆ 支持白化、FEC、CRC 校验
- ◆ 支持曼切斯特编码、8bit/10bit 线性编码
- ◆ 支持数字 RSSI 测量
- ◆ 支持 20/26MHz(± 20 ppm)晶振
- ◆ 支持 IEEE 802.15.4g 帧结构

1.2 系统框图

ES32W0030 的整体系统框图如图 1-1 所示：

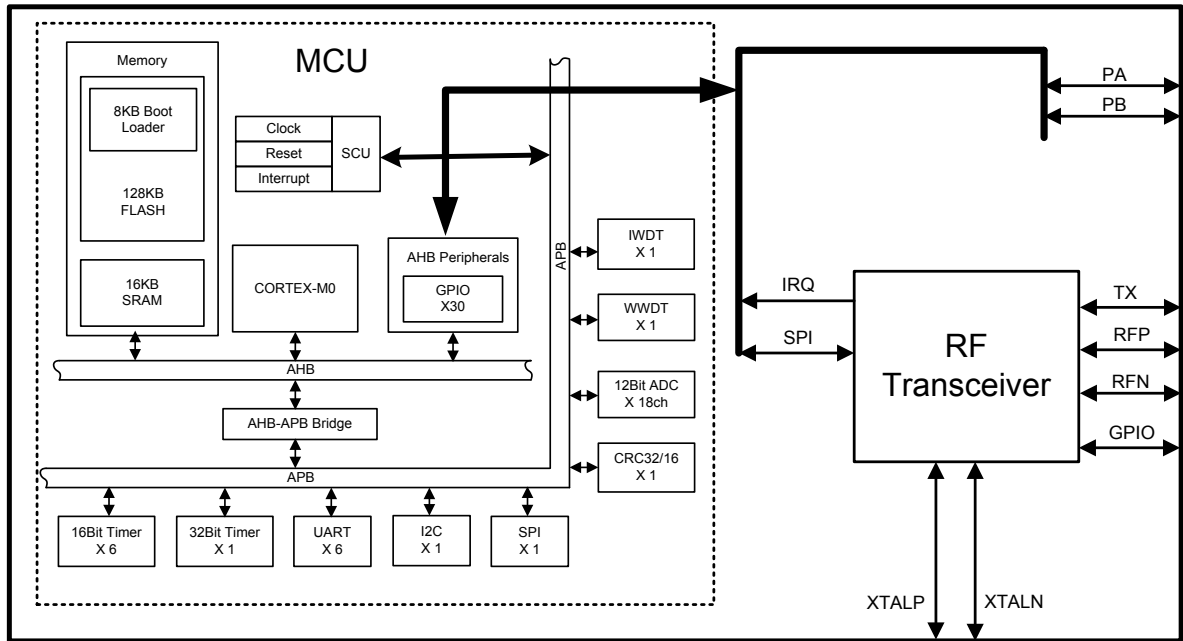


图 1-1 系统内部结构框图

1.3 管脚分配图

1.3.1 QFN48 封装

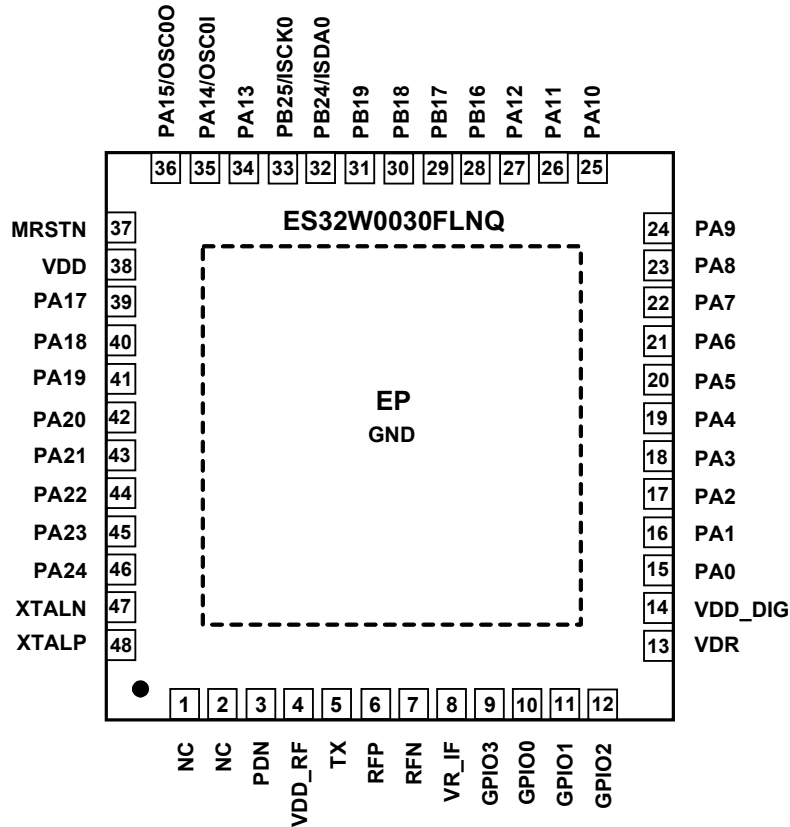


图 1-2 QFN48 封装顶视图 (TopView)

- 注 1: 编程/调试接口共用 5 线接口配置, 即电源线 VDD、地线 GND、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 ISP0 (ISCK0, ISDA0) 接口可复用为 SWD 调试接口, 推荐使用 ISP0 接口, 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 GND、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。ES32W0030FLNQ 未引出的 I/O 管脚包括 PA25~PA31, PB2~PB4, PB8~PB15, PB20~PB23, PB26~PB27。
- 注 4: PA19 端口在芯片 VDD 上电过程中, 内部弱上拉自动使能, 端口电平跟随 VDD 上升, 直到 VDD 上升到芯片开始工作的电压后, 该 IO 端口的内部弱上拉恢复为默认的禁止状态。
- 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!

1.4 管脚说明

1.4.1 管脚说明

管脚编号	管脚名称	管脚功能	管脚说明
1	NC	未连	内部未连管脚，可以悬空
2	NC	未连	内部未连管脚，可以悬空
3	PDN	数字输入	RF 收发器 POWER DOWN 模式使能输入引脚(高电平有效)
4	VDDRF	电源输入	RF 收发器模拟电源输入
5	TX	射频输出	RF 收发器发射机射频输出端口
6	RFP	射频输入	RF 收发器接收机射频输入端口
7	RFN	射频输入	
8	VR_IF	模拟电源输出	RF 收发器内置的模拟稳压电源输出
9	GPIO3	数字输入输出	RF 收发器通用数字 I/O
10	GPIO0	数字输入输出	
11	GPIO1	数字输入输出	
12	GPIO2	数字输入输出	
13	VDR	数字电源输出	RF 收发器数字稳压电源输出
14	VDD_DIG	电源输入	RF 收发器数字电源输入
15	PA0	数字输入输出	MCU 通用 IO 口
16	PA1	数字输入输出	MCU 通用 IO 口
17	PA2	数字输入输出	MCU 通用 IO 口
18	PA3	数字输入输出	MCU 通用 IO 口
19	PA4	数字输入输出	MCU 通用 IO 口
20	PA5	数字输入输出	MCU 通用 IO 口
21	PA6	数字输入输出	MCU 通用 IO 口
22	PA7	数字输入输出	MCU 通用 IO 口
23	PA8	数字输入输出	MCU 通用 IO 口
24	PA9	数字输入输出	MCU 通用 IO 口
25	PA10	数字输入输出	MCU 通用 IO 口
26	PA11	数字输入输出	MCU 通用 IO 口
27	PA12	数字输入输出	MCU 通用 IO 口
28	PB16	数字输入输出	MCU 通用 IO 口
29	PB17	数字输入输出	MCU 通用 IO 口
30	PB18	数字输入输出	MCU 通用 IO 口
31	PB19	数字输入输出	MCU 通用 IO 口
32	PB24/ISDA0	数字输入输出	MCU 通用 IO 口，编程/调试串行数据端口
33	PB25/ISCK0	数字输入输出	MCU 通用 IO 口，编程/调试串行时钟端口
34	PA13	数字输入输出	MCU 通用 IO 口
35	PA14/OSC0I	数字输入输出	MCU 通用 IO 口，MCU 外部晶体振荡器端口
36	PA15/OSC0O	数字输入输出	MCU 通用 IO 口，MCU 外部晶体振荡器端口
37	MRSTN (PA16)	数字输入输出	芯片主复位，低电平有效

管脚编号	管脚名称	管脚功能	管脚说明
38	VDD	电源输入	MCU 数字电源输入
39	PA17	数字输入输出	MCU 通用 IO 口
40	PA18	数字输入输出	MCU 通用 IO 口
41	PA19	数字输入输出	MCU 通用 IO 口
42	PA20	数字输入输出	MCU 通用 IO 口
43	PA21	数字输入输出	MCU 通用 IO 口
44	PA22	数字输入输出	MCU 通用 IO 口
45	PA23	数字输入输出	MCU 通用 IO 口
46	PA24	数字输入输出	MCU 通用 IO 口
47	XTALN	模拟输入	RF 收发器外接晶振的输出端
48	XTALP	模拟输出	RF 收发器外接晶振的输入端
EP	GND	地线	芯片底部中央的 PAD 必须接地, 包括 RF 和数字地, 建议 PCB 设计时在接地 PAD 上均匀放置多个过孔, 保证良好的接地。

表 1-1 管脚说明表

1.4.2 MCU 复用管脚对照

PIN NAME (FUNC0(D))	FUNC1 (D)	FUNC2 (D)	FUNC3 (D)	FUNC4 (D)	FUNC5 (D)	FUNC6 (D)	FUNC7 (D)	FUNC (A)
PB20	RXD0	T16N0_1	RXD1	RXD4	T16N5_0	CLKO0	SDA0	AIN16
PB21	RXD2	T16N0_0	TXD1	RXD3	T16N4_1	CLKO1	T16N1_1	AIN17
PB22	RXD1	MISO0	NSS0	RXD5	T16N4_0	T16N0_0	TXD5	—
PB23	T16N1_1	RXD5	SCK0	TXD5	T16N5_0	T16N4_0	TXD2	—
PB24	BUZ	T16N5_0	MISO0	RXD1	T16N3_0	T32N0_0	TXD5	—
PB25	T16N1_0	T16N5_1	MOSI0	TXD5	SCK0	T16N0_1	RXD1	AIN19
PB26	T16N1_1	MISO0	RXD5	T32N0_1	RXD1	T16N4_1	SDA0	—
PB27	T16N2_1	MOSI0	TXD5	T32N0_0	RXD3	T16N1_1	SCL0	—
PA13	T32N0_1	TXD3	MOSI0	TXD5	BUZ	T16N4_0	SCL0	—
PA14	T16N1_1	RXD3	T16N2_0	RXD5	TXD2	T16N0_0	SDA0	OSC0I
PA15	TXD5	TXD2	T16N2_1	SDA0	T16N1_0	T16N0_1	RXD2	OSC00/AIN14
PA16*	RXD5	RXD2	TXD1	CLKO0	T16N4_1	T32N0_0	SCL0	AIN15
PA17	TXD1	T16N0_0	T16N1_0	SCL0	T32N0_0	RXD1	RXD0	*OSC1I
PA18	RXD1	SDA0	T16N1_1	SCL0	T16N5_1	TXD0	T32N0_1	*OSC1O
PA19	T16N0_0	SCL0	TXD0	RXD3	T16N4_0	NSS0	BUZ	AIN22
PA20	T16N0_1	CLKO1	RXD0	TXD3	T16N4_1	SCK0	BUZ	AIN20
PA21	T32N0_0	TXD4	T16N3_0	T16N0_0	BUZ	RXD5	RXD0	—
PA22	T32N0_1	TXD0	T16N3_1	T16N1_1	TXD3	TXD5	MOSI0	AIN23
PA23	BUZ	RXD0	T16N1_0	TXD0	RXD3	T16N1_1	SCL0	AIN24
PA24	T16N1_0	TXD1	MOSI0	TXD0	T16N0_0	RXD2	MISO0	AIN21

PIN NAME (FUNC0(D))	FUNC1 (D)	FUNC2 (D)	FUNC3 (D)	FUNC4 (D)	FUNC5 (D)	FUNC6 (D)	FUNC7 (D)	FUNC (A)
PA25	T16N1_1	RXD1	MISO0	TXD3	T16N3_1	T16N2_0	SCK0	—
PA26	T16N2_1	T16N3_0	SCK0	TXD1	TXD3	BUZ	RXD0	—
PA27	TXD2	T16N3_1	NSS0	RXD3	SDA0	T16N2_0	MOSI0	—
PA28	RXD2	TXD2	T16N2_0	T32N0_0	T16N1_1	RXD3	SDA0	—
PA29	NSS0	TXD3	T16N2_1	RXD2	MOSI0	T32N0_1	SCL0	—
PA30	T16N0_0	RXD3	SDA0	T16N2_0	TXD5	T32N0_0	TXD4	AIN25
PA31	T16N0_1	BUZ	SCL0	T16N1_0	TXD3	T16N5_0	MOSI0	—
PB0*	TXD0	RXD0	T32N0_0	SDA0	T16N4_1	T16N5_1	RXD1	—
PB1*	RXD4	TXD0	T32N0_1	SCK0	T16N2_1	TXD5	SCL0	—
PB2	T32N0_0	RXD1	T16N3_0	T16N4_1	T16N5_0	RXD0	TXD1	—
PB3	T32N0_1	TXD1	T16N3_1	TXD2	T16N0_1	BUZ	RXD1	—
PB4	T16N0_0	NSS0	SCL0	RXD2	T16N5_1	T32N0_0	RXD1	—
PB5*	T16N0_1	SCK0	SDA0	T16N5_0	TXD2	T32N0_1	TXD1	AIN0
PB6*	T16N1_0	MISO0	RXD2	T32N0_1	TXD4	BUZ	SCL0	AIN1
PB7*	T16N1_1	MOSI0	TXD2	T32N0_0	TXD3	T16N3_1	SDA0	AIN2
PB8	BUZ	RXD3	T16N2_0	TXD4	RXD2	T16N3_0	T16N0_0	—
PB9	BUZ	TXD3	T16N2_1	MOSI0	T32N0_1	T16N4_0	RXD4	—
PB10	RXD4	NSS0	T16N5_0	TXD1	T16N0_1	SCL0	TXD4	—
PB11	TXD4	SCK0	T16N5_1	RXD1	T16N4_1	SDA0	RXD5	—
PB12	RXD5	MISO0	T32N0_1	T16N0_0	T16N1_1	TXD4	TXD5	—
PB13	TXD5	MOSI0	SCL0	T16N2_1	T32N0_0	RXD5	TXD3	—
PA0	RXD4	T16N3_0	BUZ	T16N5_1	T32N0_0	T16N4_1	TXD4	—
PA1	TXD4	T16N3_1	BUZ	TXD5	T32N0_1	T16N4_0	MOSI0	—
PA2	RXD5	T32N0_0	RXD3	TXD0	T16N0_1	CLK00	RXD1	AIN3
PA3	TXD5	T32N0_1	TXD3	RXD2	T16N2_0	TXD0	SCK0	AIN4
PA4	SCL0	RXD2	TXD5	RXD3	T16N2_0	RXD0	T32N0_0	AIN5
PA5	SDA0	TXD2	MISO0	TXD0	T16N5_1	T32N0_1	RXD5	AIN6
PA6	SDA0	T16N0_0	RXD1	T16N5_0	TXD1	TXD3	T16N1_1	AIN7
PA7	T16N2_0	T16N0_1	TXD1	T16N1_1	RXD4	RXD3	NSS0	AIN8
PA8	RXD0	T16N1_0	BUZ	MISO0	T32N0_1	TXD3	T16N2_0	AIN9
PA9	TXD0	T16N1_1	CLK00	TXD2	T16N5_0	MISO0	T16N2_1	AIN10/ AVREFN
PA10	T16N2_0	RXD4	NSS0	RXD2	T16N4_0	SCK0	TXD4	AIN11/AVREFP
PB14	T16N2_1	RXD3	TXD4	T16N3_0	RXD1	NSS0	RXD4	AIN26
PB15	MOSI0	TXD3	CLK00	T16N3_1	TXD1	T16N1_0	SDA0	AIN27
PA11	T16N2_1	TXD4	SCK0	T16N1_0	RXD3	SCL0	MISO0	AIN12
PA12	T32N0_0	BUZ	MISO0	T16N1_1	TXD3	RXD0	T16N0_0	AIN13
PB16	T32N0_1	TXD5	NSS0	RXD0	T16N0_0	T16N3_0	SDA0	—
PB17	T32N0_0	RXD5	SCK0	TXD0	T16N4_1	T16N3_1	SCL0	—

PIN NAME (FUNC0(D))	FUNC1 (D)	FUNC2 (D)	FUNC3 (D)	FUNC4 (D)	FUNC5 (D)	FUNC6 (D)	FUNC7 (D)	FUNC (A)
PB18	SDA0	T16N0_1	MISO0	BUZ	RXD4	T32N0_0	TXD4	—
PB19	SCL0	T16N0_0	MOSI0	BUZ	TXD0	T16N5_0	RXD4	—

表 1-2 管脚对照图

- 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: 支持 2 组编程接口和 1 组调试接口, ISCK0/ISDA0 位于 PB25/PB24, 支持编程、调试功能; ISCK1/ISDA1 固定位于 PA7/PA8 端口, 支持编程功能。
- 注 5: 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 PA4 端口复用为 RXD2 时, 可以设置 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 6: 支持 1 组外部晶振端口 PA14、PA15。
- 注 7: 黄底加*号的 PB0~PB1, PB5~PB7 是内部连线, 其中, PB5~PB7 必须设置为 SPI0 功能(FUN2), 而 PB0~PB1 必须设置为 GPIO。蓝底加*号的 PA16 是复位 MRSTN 管脚。
- 注 8: 加灰的是内部悬空管脚, 必须设置为 GPIO 输出低电平, 并禁止内部弱上下拉。

1.4.3 内部连线

MCU 和 RF 收发器的数字接口是直接内部连接的, 二者连接关系如下表所示:

序号	RF 收发器(I/O)	MCU(I/O)	说明
1	CSN(I)	PB0(O)	RF 收发器 SPI 的片选输入口(低有效), 内部与 MCU 的 PB0 相连, PB0 必须设置为 GPIO 输出口。
2	SCK(I)	PB5 (O)	RF 收发器 SPI 的时钟输入口, 内部与 MCU 的 PB5 相连, PB5 必须设置为 SPI0 时钟功能, 即 SCK0 (FUN2)。
3	SDI(I)	PB7(O)	RF 收发器 SPI 的输入口, 内部与 MCU 的 PB7 相连, PB7 必须设置为 SPI0 的数据输出功能, 即 MOSI0 (FUN2)。
4	SDO(O)	PB6(I)	RF 收发器 SPI 的输出口, 内部与 MCU 的 PB6 相连, PB6 必须设置为 SPI0 的数据输入功能, 即 MISO0 (FUN2)。
5	IRQ(O)	PB1(I)	RF 收发器中断输出口(极性可配, 默认低电平有效), 内部与 MCU 的 PB1 相连, PB1 必须设置为 GPIO (即 FUN0) 输入口, 并建议使能中断。

表 1-3 内部连线

1.4.4 MCU内部悬空脚注意事项

需要注意 MCU 内部悬空脚在初始化编程时,应该设置为输出低电平,以保证芯片的正常工作。

悬空脚的状态请按照下表所示来设置:

序号	悬空管脚名	设置状态
1	PA25	必须设置为 GPIO, 即 FUN0, 并输出低电平
2	PA26	必须设置为 GPIO, 即 FUN0, 并输出低电平
3	PA27	必须设置为 GPIO, 即 FUN0, 并输出低电平
4	PA28	必须设置为 GPIO, 即 FUN0, 并输出低电平
5	PA29	必须设置为 GPIO, 即 FUN0, 并输出低电平
6	PA30	必须设置为 GPIO, 即 FUN0, 并输出低电平
7	PA31	必须设置为 GPIO, 即 FUN0, 并输出低电平
8	PB2	必须设置为 GPIO, 即 FUN0, 并输出低电平
9	PB3	必须设置为 GPIO, 即 FUN0, 并输出低电平
10	PB4	必须设置为 GPIO, 即 FUN0, 并输出低电平
11	PB8	必须设置为 GPIO, 即 FUN0, 并输出低电平
12	PB9	必须设置为 GPIO, 即 FUN0, 并输出低电平
13	PB10	必须设置为 GPIO, 即 FUN0, 并输出低电平
14	PB11	必须设置为 GPIO, 即 FUN0, 并输出低电平
15	PB12	必须设置为 GPIO, 即 FUN0, 并输出低电平
16	PB13	必须设置为 GPIO, 即 FUN0, 并输出低电平
17	PB14	必须设置为 GPIO, 即 FUN0, 并输出低电平
18	PB15	必须设置为 GPIO, 即 FUN0, 并输出低电平
19	PB20	必须设置为 GPIO, 即 FUN0, 并输出低电平
20	PB21	必须设置为 GPIO, 即 FUN0, 并输出低电平
21	PB22	必须设置为 GPIO, 即 FUN0, 并输出低电平
22	PB23	必须设置为 GPIO, 即 FUN0, 并输出低电平
23	PB26	必须设置为 GPIO, 即 FUN0, 并输出低电平
24	PB27	必须设置为 GPIO, 即 FUN0, 并输出低电平

表 1-4 芯片悬空管脚设置说明表

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 ₆															
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	SCU 写保护位 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_PLLKCON, SCU_SCLKEN0, SCU_SCLKEN1, SCU_PCLKEN0, SCU_PCLKEN1, 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

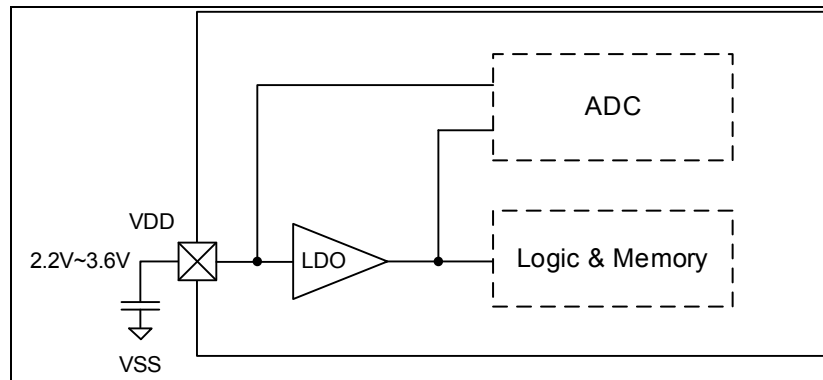


图 2-1 系统电源结构框图

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。

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 调试接口软件复位
- ◇ 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

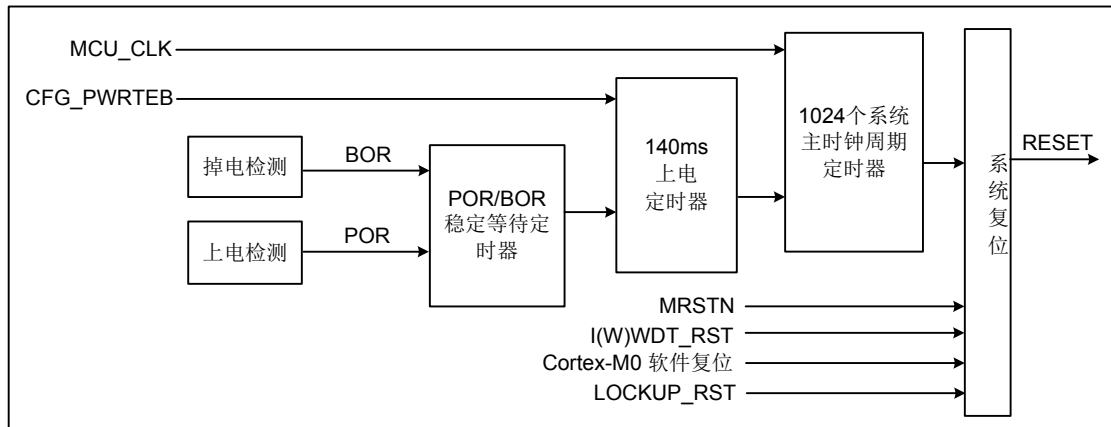


图 2-2 系统复位电路结构框图

- 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。
- 注 2: 芯片上电稳定后, 在工作过程中, 如果发生外部复位, IWDT/WWDT 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。

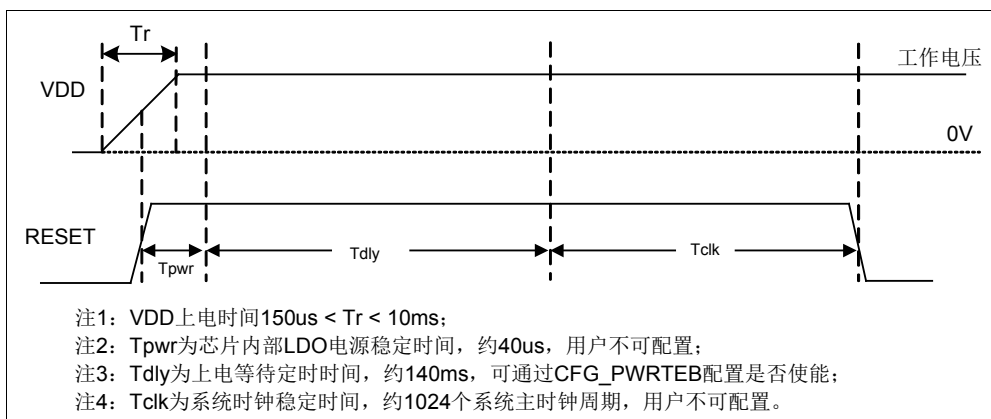


图 2-3 上电复位时序示意图

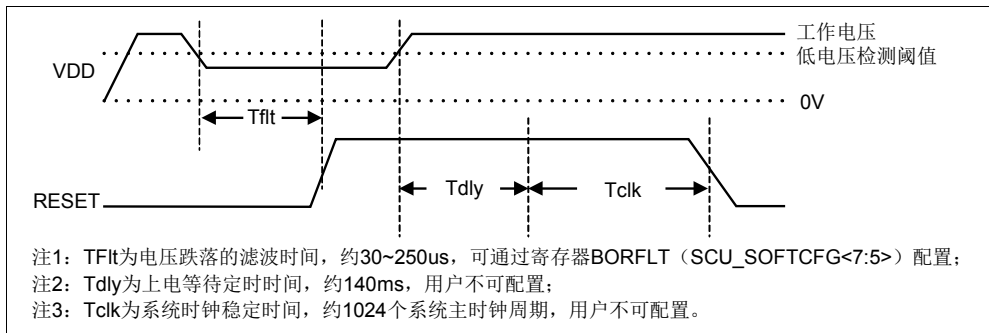


图 2-4 掉电复位时序示意图

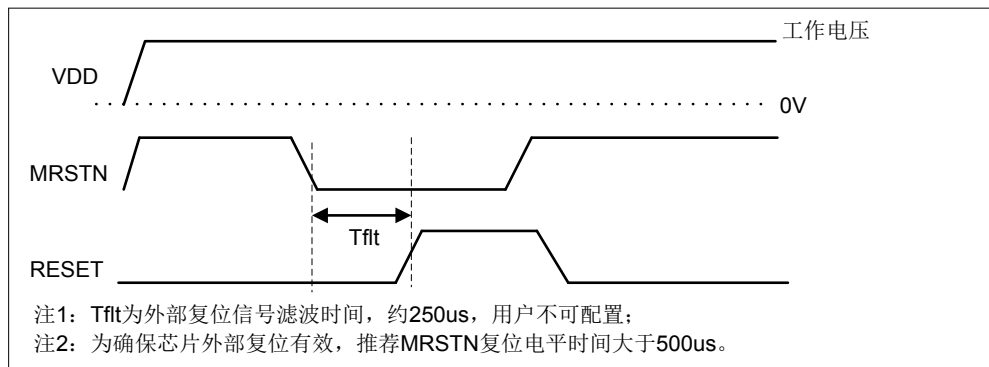


图 2-5 外部复位时序示意图

2.3.4 外部复位MRSTN参考

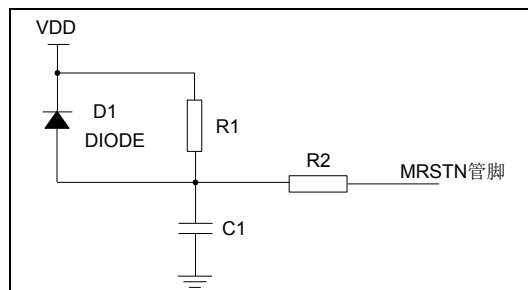


图 2-6 MRSTN 复位参考电路图 1

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。

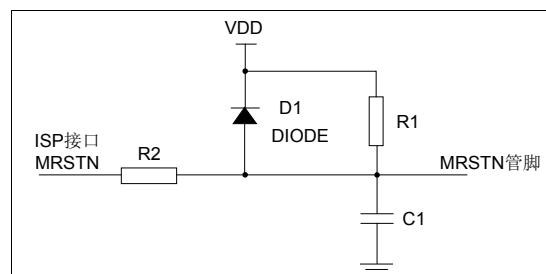


图 2-7 MRSTN 复位参考电路图 2

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 $R2$ 。

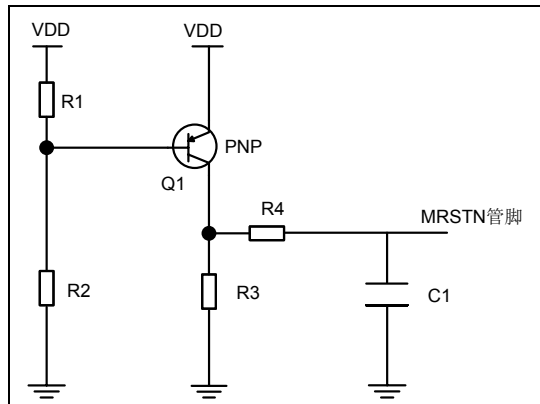


图 2-8 MRSTN 复位参考电路图 3

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

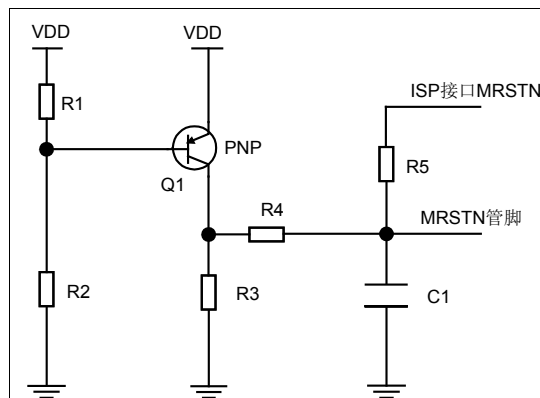


图 2-9 MRSTN 复位参考电路图 4

注 1: 采用 PNP 三极管复位, 通过 $R1$ ($2K\Omega$) 和 $R2$ ($10K\Omega$) 分压作为基极输入, 发射极接 VDD, 集电极一路通过 $R3$ ($20K\Omega$) 接地, 另一路通过 $R4$ 和 $C1$ ($0.1\mu F$) 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 _H															
复位值: 00000000_00000000_0000xxx_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				LKUPRS	保留			SOFT_	MRST	IWDT	WWDT	BOR	POR	POR	POR
				TF				RSTF	F	RSTF	RSTF	F	RSTF	RCF	F

—	bit31-8	—	—
LKUPRSTF	bit10	R/W	LOCKUP 复位标志位 0: 无软件复位 1: 有软件复位
—	bit9-8	—	—
SOFT_RSTF	bit7	R/W	软件复位标志位 0: 无软件复位 1: 有软件复位
MRSTF	bit6	R/W	MRSTN 复位标志位 0: 无 MRSTN 复位 1: 有 MRSTN 复位
IWDTRSTF	bit5	R/W	IWDT 复位标志位 0: 无 IWDT 复位 1: 有 IWDT 复位
WWDTRSTF	bit4	R/W	WWDT 复位标志位 0: 无 WWDT 复位 1: 有 WWDT 复位
BORF	bit3	R/W	BOR 复位标志位 0: 无 BOR 复位 1: 有 BOR 复位
PORRSTF	bit2	R/W	PORRST 复位标志位 (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	PORRC 复位标志位 0: 无 PORRC 复位 1: 有 PORRC 复位
PORF	bit0	R/W	POR 复位标志位 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: 50H															
复位值: 00000000_00000000_00000011_10111110 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							WWDT_RST	IWDT_RST	保留		ADC_RST	CRC_RST	IAP_RST	GPIO_RST	保留

—	bit 31-9	—	—
WWDT_RST	bit 8	R/W	WWDT 复位使能位 0: 无复位 1: 使能复位
IWDT_RST	bit 7	R/W	IWDT 复位使能位 0: 无复位 1: 使能复位
—	bit 6-5	—	—
ADC_RST	bit 4	R/W	ADC 复位使能位 0: 无复位 1: 使能复位
CRC_RST	bit 3	R/W	CRC 复位使能位 0: 无复位 1: 使能复位
IAP_RST	bit 2	R/W	FLASH_IAP 复位使能位 0: 无复位 1: 使能复位
GPIO_RST	bit 1	R/W	GPIO 复位使能位 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 _H															
复位值: 00010001_00111111_00000001_00001111 _B															
3	3	2	28	2	26	2	24	23	22	21	20	19	18	17	16
1	0	9		7		5									
保留	I2C0_ RST	保留	SPI0_R ST	保留	UART5 _RST	UART4_RS T	UART3_RST	UART2_R ST	UART1 _RST	UART0 _RST					
1	1	1	12	11	10	9	8	7	6	5	4	3	2	1	0
5	4	3													
保留							T32N0_ RST	保留	T16 N5_ RST	T16N4_ RST	T16N3_ RST	T16N2 _RST	T16N1 _RST	T16N0 _RST	

—	bit 31-29	—	—
I2C0_RST	bit28	R/W	I2C0 复位使能位 0: 无复位 1: 使能复位
—	bit 27-25	—	—
SPI0_RST	bit 24	R/W	SPI0 复位使能位 0: 无复位 1: 使能复位
—	bit 23-22	—	—
UART5_RST	bit 21	R/W	UART5 复位使能位 0: 无复位 1: 使能复位
UART4_RST	bit 20	R/W	UART4 复位使能位 0: 无复位 1: 使能复位
UART3_RST	bit 19	R/W	UART3 复位使能位 0: 无复位 1: 使能复位
UART2_RST	bit 18	R/W	UART2 复位使能位 0: 无复位 1: 使能复位
UART1_RST	bit 17	R/W	UART1 复位使能位 0: 无复位 1: 使能复位
UART0_RST	bit 16	R/W	UART0 复位使能位 0: 无复位 1: 使能复位
—	bit 15-9	—	—
T32N0_RST	bit 8	R/W	T32N0 复位使能位 0: 无复位

			1: 使能复位
—	bit 7-6	—	—
T16N5_RST	bit 5	R/W	T16N5 复位使能位 0: 无复位 1: 使能复位
T16N4_RST	bit 4	R/W	T16N4 复位使能位 0: 无复位 1: 使能复位
T16N3_RST	bit 3	R/W	T16N3 复位使能位 0: 无复位 1: 使能复位
T16N2_RST	bit 2	R/W	T16N2 复位使能位 0: 无复位 1: 使能复位
T16N1_RST	bit 1	R/W	T16N1 复位使能位 0: 无复位 1: 使能复位
T16N0_RST	bit 0	R/W	T16N0 复位使能位 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。

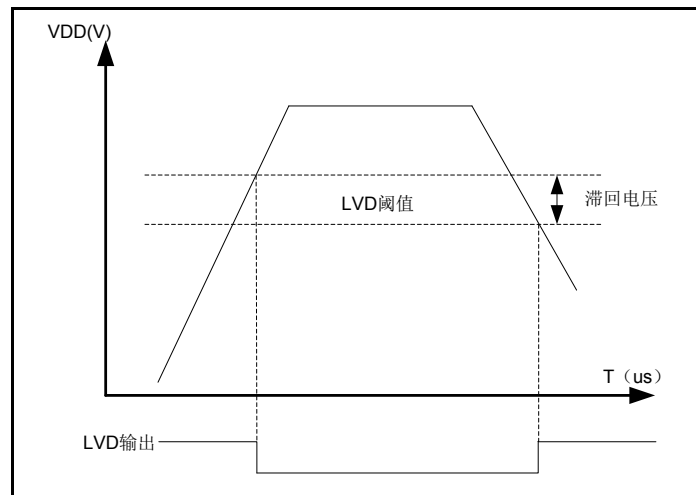


图 2-10 LVD 低电压监测示意图

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	LVD 中断标志产生模式选择位 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	LVD 中断使能位 0: 禁止 1: 使能
IF	bit8	R/W	LVD 中断标志位 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	LVD 触发电压 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	LVD 滤波使能位 0: 禁止 1: 使能
EN	bit0	R/W	LVD 使能位 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
 - IWDG、WWDG 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：20 _H															
复位值：00000000_00000000_00000000_00000010 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											HS	ACCT<3:0>			

—	bit31-5	—	—
HS	bit4	R/W	FLASH 访问模式 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	FLASH 读取访问等待时间设置位 0x0: 1T _{CLK} 完成 FLASH 读取 0x1: 2T _{CLK} 0x2: 3T _{CLK} ... 0xF: 16T _{CLK}

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_{CLK} 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 32KHz 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 32KHz：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

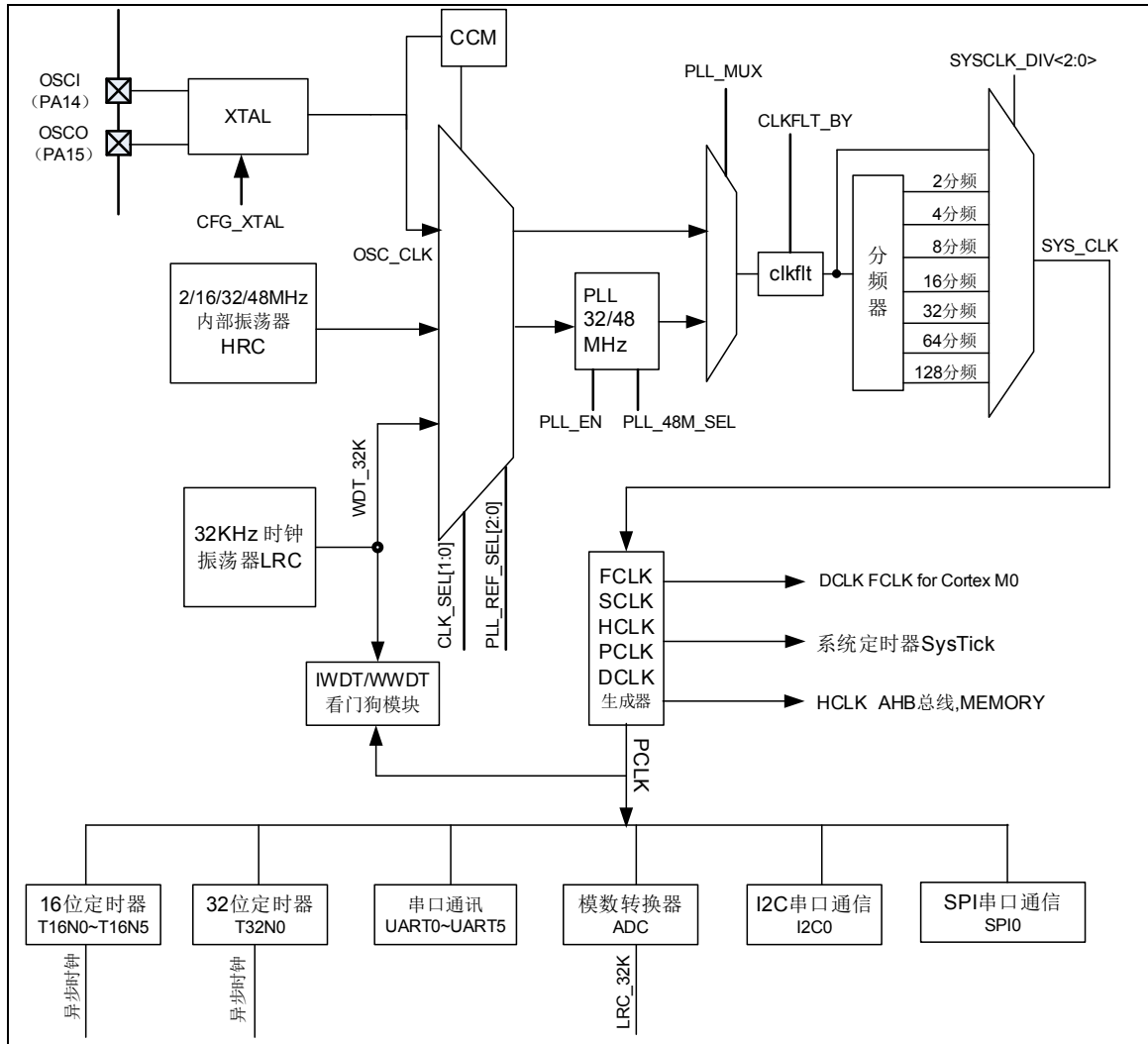


图 2-11 系统时钟电路结构框图

注：每款芯片支持 1 组外部晶振端口，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。

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

当使用外部振荡器时，需外接匹配电容。XTAL 振荡器电路示意图如下：

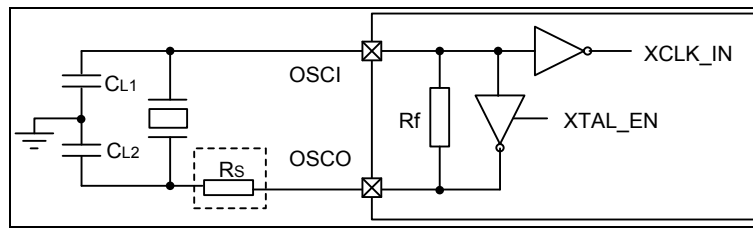


图 2-12 XTAL 振荡器电路结构示意图

注 1：电阻 Rs 为可选配置。

注 2：Cl1 和 Cl2 为晶振匹配电容，根据所使用的晶振，电容参考取值范围为 10~20pF，建议 1~20MHz 晶振匹配 15pf 电容，32.768KHz 晶振匹配 12pf 电容，具体电容值需根据外接晶振的参数需求确定。

晶体振荡器的等效电路如下图所示：

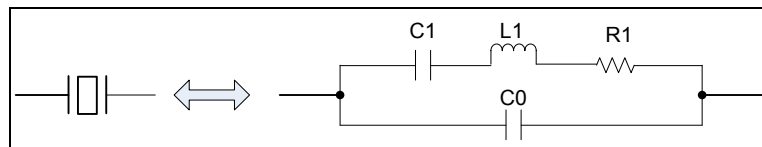


图 2-13 晶体振荡器等效电路示意图

注 1：C1：动态电容；L1：动态电感；R1 动态电阻；C0：静电容；等效串联电阻 $ESR=R1x(1+C0/CL)^2$ ，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 $\pm 1\%$ 。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。

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

在 HRC_EN=1 的情况下，当 MOSC_EN=0，芯片进入深度睡眠模式时，HRC 时钟会自动关断，被唤醒后，HRC 时钟会自动使能；当 MOSC_EN=1，芯片进入深度睡眠模式时，HRC 时钟不会关闭。

2.6.3.3 内部低速时钟LRC

芯片支持内部低速时钟 LRC（频率约为 32KHz），且无法关闭，始终保持工作。常温条件下 LRC 时钟频率精度约为±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（32KHz）。

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 32.768MHz。当 PLL_48M_SEL=1 时：PLL 倍频系数为 1536 倍，PLL 输出时钟约为 49.152MHz。

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。

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 LRC 时钟，同时会置起 SCU_CCM 寄存器的外部时钟停振标志 FLAG 和 CCM 中断标志位 IF，通过 SCU_CCM 寄存器的 IFS 位，可配置 CCM 中断标志 IF 的产生方式，通过停振中断使能位 IE，可设置中断标志位 IF 是否触发 CCM 中断请求 IRQ。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。

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 LRC 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 CLKFKT，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 _H															
复位值: 00000000_00000100_01110000_01111111 _B															
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 (推荐设置为使能, 以降低功耗)
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 电压选择位 01: 低功耗电压 1 (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	深度睡眠模式时钟控制位

			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
WAKEUPTIME<11:0>	bit11-0	R/W	唤醒时间控制位 $T_{PCLK} * WAKEUPTIME$

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。
 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, 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。

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: 40 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	CLKO1 选择位 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	CLKO0 选择位 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	CLKFLT 旁路控制位 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	系统时钟后分频选择位 000: 1:1

			1: 稳定
XTAL_RDY	bit16	R	外部时钟振荡模式稳定标志位 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
—	bit15-14	—	—
PLL_BYLOCK	bit13	R/W	PLL 锁频信号旁路控制位 0: 不旁路 1: 旁路
PLL_EN	bit12	R/W	PLL 倍频电路工作使能位 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	PLL 输出时钟选择位 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	PLL 输入时钟源选择位 (仅在 PLL_EN=0 时可写入) 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	HRC 频率选择位 00: 2MHz 01: 16MHz 10: 32MHz 11: 48MHz
—	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: 48 _H																
复位值: 00000000_00000000_00000011_10111111 _B																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留							WWDT_EN	IWDT_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
—	bit 6-5	—	—
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	FLASH_IAP 时钟使能位 0: 禁止 1: 使能
GPIO_EN	bit 1	R/W	GPIO 时钟使能位 0: 禁止 1: 使能
SCU_EN	bit 0	R/W	SCU 时钟使能位 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 _H															
复位值: 00010001_00111111_00000001_00001111 _B															
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	bit 28	R/W	I2C0 时钟使能位 0: 禁止 1: 使能
—	bit 27-25	—	—
SPI0_EN	bit 24	R/W	SPI0 时钟使能位 0: 禁止 1: 使能
—	bit 23-22	—	—
UART5_EN	bit 21	R/W	UART5 时钟使能位 0: 禁止 1: 使能
UART4_EN	bit 20	R/W	UART4 时钟使能位 0: 禁止 1: 使能
UART3_EN	bit 19	R/W	UART3 时钟使能位 0: 禁止 1: 使能
UART2_EN	bit 18	R/W	UART2 时钟使能位 0: 禁止 1: 使能
UART1_EN	bit 17	R/W	UART1 时钟使能位 0: 禁止 1: 使能
UART0_EN	bit 16	R/W	UART0 时钟使能位 0: 禁止

—	bit 15-9	—	1: 使能
T32N0_EN	bit 8	R/W	T32N0 时钟使能位 0: 禁止 1: 使能
—	bit 7-6	—	—
T16N5_EN	bit 5	R/W	T16N5 时钟使能位 0: 禁止 1: 使能
T16N4_EN	bit 4	R/W	T16N4 时钟使能位 0: 禁止 1: 使能
T16N3_EN	bit 3	R/W	T16N3 时钟使能位 0: 禁止 1: 使能
T16N2_EN	bit 2	R/W	T16N2 时钟使能位 0: 禁止 1: 使能
T16N1_EN	bit 1	R/W	T16N1 时钟使能位 0: 禁止 1: 使能
T16N0_EN	bit 0	R/W	T16N0 时钟使能位 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 _H															
复位值: 00000000_00000000_00000000_0000001 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															FLAG
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							IF	IFS<2:0>			IE	保留		EN	
—	bit31-17		—	—											
FLAG	bit16		R	CCM 外部时钟停振检测标志位 0: 晶振未停振 1: 晶振停振											

—	bit15-9	—	—
IF	bit8	R/W	CCM 中断标志位 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	CCM 中断标志产生模式选择位 000: CCM_FLAG 上升沿产生中断, 晶振停振 001: CCM_FLAG 下降沿产生中断, 晶振恢复振荡 010: CCM_FLAG 高电平产生中断, 晶振停振 011: CCM_FLAG 低电平产生中断, 晶振恢复振荡 1xx: CCM_FLAG 变化 (上升或下降沿) 产生中断
IE	bit 4	R/W	外部时钟停振中断使能位 0: 中断禁止 1: 中断使能
—	bit3-1	—	—
EN	bit0	R/W	外部时钟停振检测使能位 0: 禁止 1: 使能

注 1: 对 SCU_CCM 寄存器进行写操作前, 需要设置 SCU_PROT 寄存器, 关闭写保护。

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

2.6.4.7 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	PLL 锁定检测标志位 0: PLL 未锁定 1: PLL 锁定, 锁频成功
—	bit15-9	—	—
IF	bit8	R/W	PLL 中断标志位 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	PLL 锁定标志产生模式选择位 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	PLL 锁定中断使能位 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。

2.6.5 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
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
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 _H															
复位值：00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										NMICS<4:0>			NMIEN		
—				bit31-6				—				—			
NMICS<4:0>				bit5-1				R/W				NMI 不可屏蔽中断选择位 00000: IRQ0 00001: IRQ1 ... 11111: IRQ31			
NMIEN				bit0				R/W				NMI 不可屏蔽中断使能位 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 _H															
复位值：00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															EN
—				bit31-1				—				—			
EN				bit0				R/W				中断向量表重映射使能			

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

注：对 SCU_TBLREMAPEN 寄存器进行写操作前，需要设置 SCU_PROT 寄存器，关闭写保护。

2.7.4.3 中断向量表偏移寄存器 (SCU_TBLOFF)

中断向量表偏移寄存器 (SCU_TBLOFF)															
偏移地址: 64 _H															
复位值: 00100000_00000000_00000000_00000000 _B															
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>中断向量表偏移地址</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<31:8>可读可写，但低 8 位 TBLOFF<7:0>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: 0CH															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													FLAG2	FLAG1	FLAG0

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

- 注 1: 读指令代码为空, 表示 Cortex-M0 内核读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：00 _H															
复位值：00000000_00000000_00000000_00000000 _B															
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	IRQ 使能位 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：80 _H															
复位值：00000000_00000000_00000000_00000000 _B															
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	IRQ 禁止位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IRQ3 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_2<1:0>	bit23-22	R/W	IRQ2 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_1<1:0>	bit15-14	R/W	IRQ1 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_0<1:0>	bit7-6	R/W	IRQ0 优先级设置位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IRQ7 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_6<1:0>	bit23-22	R/W	IRQ6 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_5<1:0>	bit15-14	R/W	IRQ5 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_4<1:0>	bit7-6	R/W	IRQ4 优先级设置位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IRQ11 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_10<1:0>	bit23-22	R/W	IRQ10 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_9<1:0>	bit15-14	R/W	IRQ9 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_8<1:0>	bit7-6	R/W	IRQ8 优先级设置位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IRQ15 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_14<1:0>	bit23-22	R/W	IRQ14 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_13<1:0>	bit15-14	R/W	IRQ13 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_12<1:0>	bit7-6	R/W	IRQ12 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

2.7.4.13 IRQ16~19 优先级控制寄存器 (NVIC_PR4)

IRQ16~19 优先级控制寄存器 (NVIC_PR4)															
偏移地址: 310 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IRQ19 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_18<1:0>	bit23-22	R/W	IRQ18 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_17<1:0>	bit15-14	R/W	IRQ17 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_16<1:0>	bit7-6	R/W	IRQ16 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

2.7.4.14 IRQ20~23 优先级控制寄存器 (NVIC_PR5)

IRQ20~23 优先级控制寄存器 (NVIC_PR5)															
偏移地址: 314 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IRQ23 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_22<1:0>	bit23-22	R/W	IRQ22 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_21<1:0>	bit15-14	R/W	IRQ21 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_20<1:0>	bit7-6	R/W	IRQ20 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit5-0	—	—

2.7.4.15 IRQ24~27 优先级控制寄存器 (NVIC_PR6)

IRQ24~27 优先级控制寄存器 (NVIC_PR6)															
偏移地址: 318 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IRQ27 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_26<1:0>	bit23-22	R/W	IRQ26 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_25<1:0>	bit15-14	R/W	IRQ25 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_24<1:0>	bit7-6	R/W	IRQ24 优先级设置位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IRQ31 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit29-24	—	—
PRI_30<1:0>	bit23-22	R/W	IRQ30 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit21-16	—	—
PRI_29<1:0>	bit15-14	R/W	IRQ29 优先级设置位 00: 最高优先级 11: 最低优先级
—	bit13-8	—	—
PRI_28<1:0>	bit7-6	R/W	IRQ28 优先级设置位 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 _H															
复位值: 01000001_00001100_11000010_00000000 _B															
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 mpn 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 mpn 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	NMI 中断挂起控制位 0: 不置 NMI 中断挂起 1: 置 NMI 中断挂起
—	bit30-27	—	—
PENDSTSET	bit26	R/W	置 SysTick 异常挂起位 0: 无效 1: 置 SysTick 异常挂起
PENDSTCLR	bit25	W	清 SysTick 异常挂起位 0: 无效 1: 清除 SysTick 异常挂起
—	bit24-23	—	—
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: 0CH																
复位值: 11111010_00000101_00000000_00000000 _B																
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											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 _H																
复位值: 00000000_00000000_00000010_00001000 _B																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留							STKALIGN	保留					UNALIGN_TRP	保留		

—	bit31-10	—	—
STKALIGN	bit9	R	非堆栈对齐标志位 读取始终为 1，指示异常入口 8 字节堆栈对齐
—	bit8-4	—	—
UNALIGN_TRP	bit3	R	字或半字访问操作的非对齐故障标志位 读取始终为 1，指示非对齐访问产生硬故障
—	bit2-0	—	—

2.8.2.6 系统处理程序优先级寄存器 2 (SCB_SHPR2)

系统处理程序优先级寄存器 2 (SCB_SHPR2)															
偏移地址: 1C _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_15<1:0>		保留						PRI_14<1:0>		保留					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
PRI_15<1:0>	bit31-30	R/W	SysTick (异常编号 15) 的优先级设置位												
—	bit29-24	—	—												
PRI_14<1:0>	bit23-22	R/W	PendSV (异常编号 14) 的优先级设置位												
—	bit21-0	—	—												

2.9 系统定时器 (SYSTICK)

2.9.1 概述

- ◇ 24 位系统递减计数器，递减至零可自动重载计数初值
- ◇ 可产生周期性 SysTick 异常，用作嵌入式操作系统的多任务调度计数器；或对于无嵌入式操作系统的运用，可用于调用需周期性执行的任务
- ◇ SysTick 亦可用作普通定时器，如用于延时计数
- ◇ SysTick 异常优先级可由系统处理优先级寄存器 SHPR3 的 PRI_15<1:0> 设定
- ◇ SysTick 异常处理的挂起可由中断控制和状态寄存器 SCB_ICSR 的 PENDSTSET 位设置
- ◇ 工作时钟可为系统时钟 HCLK 或其三分频

SysTick 是一个系统递减计数器，配置 SYST_RVR 寄存器，可设定计数初值。当 SysTick 计数为 0 时，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。

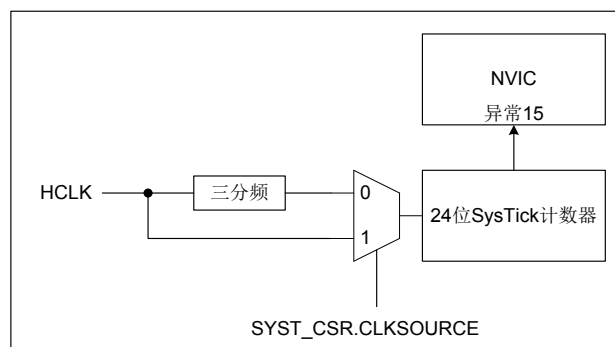


图 2-14 系统定时器框图

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	SYSTICK 递减计数到零的标志位 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	SYSTICK 时钟源选择位 0: 基准时钟 (HCLK 的 3 分频) 1: 处理器时钟 HCLK
TICKINT	bit1	R/W	SYSTICK 异常挂起使能位 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	SYSTICK 计数器使能位 0: 禁止 1: 使能

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

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

2.9.2.2 SYSTICK重装值寄存器 (SYST_RVR)

SYSTICK 重装值寄存器 (SYST_RVR)															
偏移地址: 14 _H															
复位值: 00000000_11111111_11111111_11111111 _B															
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	SYSTICK 计数器重载值												

			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 _H															
复位值: 00000000_11111111_11111111_11111111 _B															
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	SYSTICK 计数器当前值 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 _H															
复位值: 01000000_00000010_10001011_00001010 _B															
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	基准时钟标志位 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	TENMS 校准值是否准确的标志位 0: TENMS 校准值准确 1: TENMS 校准值不准确
—	bit29-24	—	—
TENMS<23:0>	bit23-0	R/W	SYSTICK 校准值 读取为 0 时, 表示校准值未知

注: 本产品只提供内部基准时钟, 其频率为 $F_{HCLK}/3$;

注 2: 在应用中一般不需要设置和使用 SYSTICK 校准值寄存器。

2.9.3 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。

2.9.3.1 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 _H															
复位值: 00000000_00000000_00000000_11100001 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BORFLT<2:0>				保留			

—	bit31-8	—	—
BORFLT	bit7-5	R/W	BOR 复位信号滤波长度 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。

2.10 定时器 (T16N/T32N) 同步启动关停控制

2.10.1 概述

通过 SCU_TIMEREN 和 SCU_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。

2.10.2 特殊功能寄存器

2.10.2.1 SCU_TIMEREN使能控制寄存器 (SCU_TIMEREN)

SCU_TIMEREN 使能控制寄存器 (SCU_TIMEREN)															
偏移地址: 58 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							T32N0EN	保留	T16N5EN	T16N4EN	T16N3EN	T16N2EN	T16N1EN	T16N0EN	

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

T16N0EN	bit0	W	T16N0使能位 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. 10. 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							T32N0DIS	保留		T16N5DIS	T16N4DIS	T16N3DIS	T16N2DIS	T16N1DIS	T16N0DIS

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

2.11 模块调试相关配置

2.11.1 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。

2.11.2 特殊功能寄存器

2.11.2.1 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 _H															
复位值: xxxxxxxx_xxxxxx00_xxxxxx0_xxxx0000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													WWDT HT	IWDTH	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						T32N0HT	保留	T16N5H T	T16N4H T	T16N3H T	T16N2H T	T16N1 HT	T16N0H T		

—	bit31-18	—	—
WWDTHT	bit17	R/W	WWDT调试暂停使能位 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	IWDT调试暂停使能位 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	T32N0调试暂停使能位 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	T16N5调试暂停使能位 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	T16N4调试暂停使能位 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	T16N3调试暂停使能位 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	T16N2调试暂停使能位 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	T16N1调试暂停使能位 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	T16N0调试暂停使能位 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。

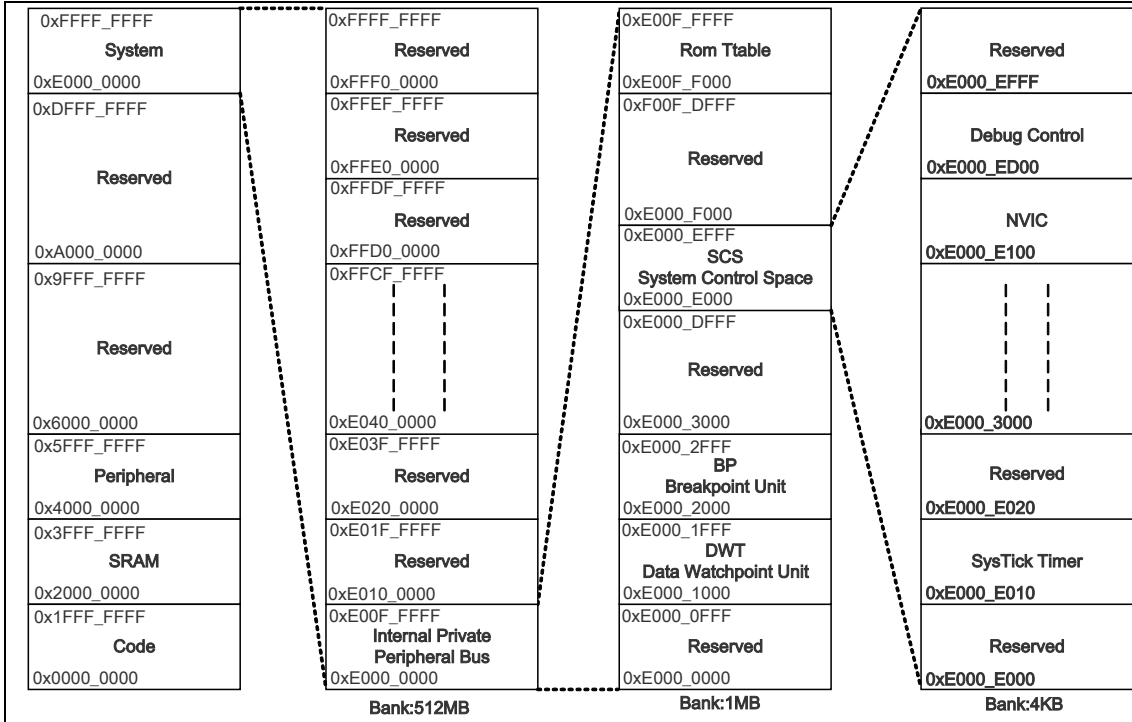


图 3-1 内部存储系统分配示意图

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。

- ◇ 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
- ◇ 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 0x20000, 芯片配置字 CFG_WORD0, CFG_WORD1, CFG_WRP0, CFG_WRP1, CFG_DAFLS 和 CFG_GBRDP 的地址分别为 0x20000, 0x20008, 0x20020, 0x20028, 0x20030 和 0x20040, 芯片唯一识别码 UID0~UID2 的地址分别为 0x209E0, 0x209E8, 0x209F0

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 _H	
低 16 位复位值	1000_0100_1010_0000 _B (84A0 _H)	
—	bit 63-32	保留未用
—	bit 31-16	bit 15-0 取反值 (不满足取反时 bit15-0 强制为默认值)
CFG_SWD	bit 15	SWD 调试模式使能位 0: 禁止 (调试完毕后需禁止, 避免管脚输入悬空) 1: DEBUG 自动识别 (默认)
CFG_PB25R	bit 14	PB25 用于复位使能位 0: 禁止 (默认) 1: PB25 可用作 MRSTN
—	bit 13-11	固定为 000
CFG_START	bit 10	程序起始区域选择位 0: BootFlash 1: AppFlash (默认)
—	bit 9-7	固定为 001
CFG_PWRTEB	bit 6	上电 140ms 延时使能位 0: 使能 (默认) 1: 禁止
CFG_BORV	bit5-4	BOR 电压点选择位 00: 保留

		01: 2.5V 10: 2.1V (默认) 11: 3.1V
CFG_WWDTEN	bit 3	WWDT 硬件看门狗使能 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	IWDT 硬件看门狗使能 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。
—	bit1	保留
CFG_XTAL	bit 0	外部振荡器模式选择位 0: 1~20MHz (高速、默认, 为 HS 或 XT 模式) 1: 32KHz (低速, 为 LP 模式) 该位应与 CFG_WORD1 中的 OSCMD 位对外部振荡器 HS, XT 和 LP 模式的选择保持一致。

- 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。
- 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>), 等外部时钟振荡器稳定后, 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: SWD 调试模式使能后, 需避免在程序软件中设置对应的 IO 端口控制寄存器, 否则会导致调试异常。
- 注 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 _H	
低 16 位复位值	0000_0000_1100_0010 _B (C2 _H)	
—	bit 31-16	bit 15-0 取反值
—	bit 15-8	保留未用
OSCMD	bit7-0	XTAL 振荡器工作模式选择位 0xC2: 高速 HS 模式 (5~20MHz) (默认) 0x84: 高速 XT 模式 (1~4MHz) 0x1F: 低速 LP 模式 (32KHz) 该位应与 CFG_WORD0 中的 CFG_XTAL 位配置一致。

寄存器名称	写保护区域 x 配置字 (CFG_WRPx) (x=0,1)	
地址偏移	0020 _H ~0028 _H	
低 16 位复位值	0000_0000_0000_0001 _B (0001 _H)	
—	bit 31-16	bit 15-0 取反值
END	bit 15-10	保护结束页配置位 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 0x3F: Flash Page 255 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 9-8	保留未用
START	bit 7-2	保护起始页配置位 0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 0x3F: Flash Page 252
—	bit 1	保留未用
ENB	bit 0	保护使能位 0: 使能 1: 禁止 (默认)

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

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

寄存器名称	数据 Flash 配置字 (CFG_DAFLS)	
地址偏移	0030 _H	
低 16 位复位值	0000_0000_0000_0001 _B (0001 _H)	
—	bit 31-16	bit 15-0 取反值
END	bit 15-10	数据 Flash 结束页配置位 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 0x3F: Flash Page 255 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 9-8	保留未用
START	bit 7-2	数据 Flash 起始页配置位

		0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 0x3F: Flash Page 252
—	bit 1	保留未用
ENB	bit 0	数据 Flash 使能位 0: 使能 1: 禁止 (默认)

寄存器名称	全局读保护配置字 (CFG_GBRDP)	
地址偏移	0040 _H	
复位值	0101_0101_1010_1010_0101_0101_1010_1010 _B (55AA_55AA _H)	
GBRDP	bit 31-0	全部读保护配置位 0xFFFF_FFFF: 读保护等级 Level 0 0xFFFF_XXXX: 读保护等级 Level 1 (XXXX 不为 FFFF) 0xYYYY_XXXX: 读保护等级为 Level 2 (YYYY 不为 FFFF) (默认)

3.2.1.2 芯片唯一识别码UID

96 位芯片唯一识别码 UID 位于 FLASH 存储器的 INFO4 信息区，以 word 为单位存放，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 _H (UID2)、000209E8 _H (UID1)、000209E0 _H (UID0)	
UID2	bit95~64	芯片唯一识别码 UID2
UID1	bit63~32	芯片唯一识别码 UID1
UID0	bit31~0	芯片唯一识别码 UID0

3.2.2 程序区FLASH

ES32W0030 系列芯片内部的程序存储器 FLASH 容量最大为 128K 字节，地址范围为 0000_0000_H~0001_FFFF_H，共分为 256 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
ES32W0030	128K Bytes	0x0000_0000 _H ~0x0001_FFFF _H	256

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 DEBUG 使能时，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。

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：0x0001_E000。

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。

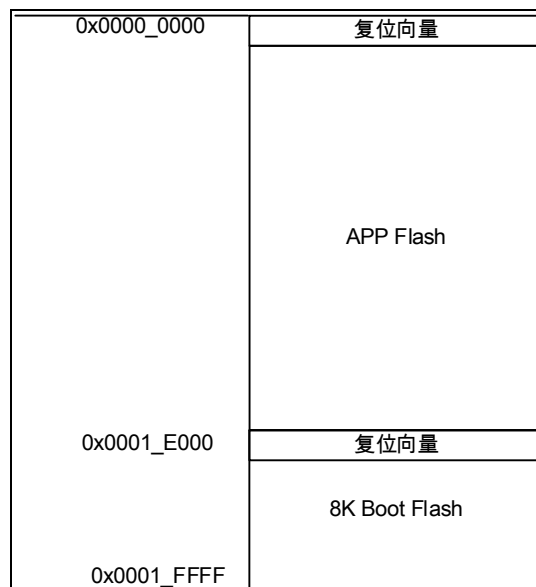


图 3-2 Boot Flash 启动地址分配示意图

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

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 (512Bytes), 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_FLASHADDINV.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_FLASHADDINV, 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;
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，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，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_H

复位值: 00000000_00000000_00000000_00000011_B

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

—	bit 31-2	—	保留
STATUS	bit 1-0	R	FLASH 程序区状态位 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_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit 31-17	—	保留
ADDR	bit 16-0	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_H

复位值: 00000000_00000000_00000000_00000000_B

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_H

复位值: 00000000_00000000_00000000_00000000_B

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_H

复位值: 00000000_00000000_00000000_00000000_B

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	FLASH编程命令 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：1CH

复位值：00000000_00000000_00000000_00000000_B

3 1	3 0	2 9	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

1 5	1 4	1 3	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			PGSZ	FIFODF	保留	FIFOEN	FLASHREQ	保留			IAPRS T	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	FIFO 编程数据 Flash 使能 0: 禁止 1: 使能
—	bit 6	—	保留
FIFOEN	bit 5	R/W	FIFO 使能 0: 禁止 1: 使能
FLASHREQ	bit 4	R/W	FLASH 操作请求使能 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_H

复位值: 00000000_00000000_00000000_00000000_B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							MASEUL	ADDERR	保留						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							TIMEOUT	PROG	SERA	MASE	WAE	WPE	BUSY	FLASHACK	

—	bit 31-25	—	保留
MASEUL	bit 24	R	程序区全擦保护解除位 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	地址反码错误标识位 0: 反码正确 1: 反码错误
—	bit 22-8	—	保留
TIMEOUT	bit 7	R	超时错误标志 0: 无错误 1: 发生错误 未在规定时间内完成相应擦除或编程动作时产生错误标志, 可能硬件发生了故障, 需软件触发一次IAP复位
PROG	bit 6	R	字编程完成标志 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
SERA	bit 5	R	页擦除完成标志 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
MASE	bit 4	R	程序区全擦除完成标志 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
WAE	bit 3	R	擦写地址错误标志

			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	擦写保护错误标志 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	自编程状态复位 0: 空闲 1: 正在进行
FLASHACK	bit 0	R	FLASH 操作许可状态 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_H

复位值: 00000000_00000000_00000000_00000000_B

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

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

—	bit 31-17	—	保留
ADDRINV	bit 16-0	R/W	FLASH 地址反码

3.4 数据存储器 (SRAM)

ES32W0030 系列芯片内部集成最大 16K 字节数据存储器 SRAM，地址范围为 2000_0000_H~2000_3FFF_H。

型号	SRAM 数据存储器	地址范围
ES32W0030	16K Bytes	2000_0000 _H ~2000_3FFF _H

3.4.1 SRAM地址映射

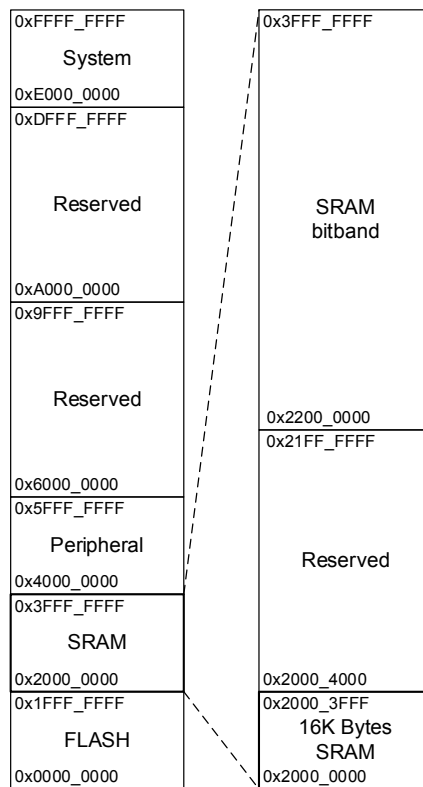


图 3-3 SRAM 映射图

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：
 $AliasAddress_A_N = 0x2200_0000 + (A - 0x2000_0000) \times 32 + N \times 4$

例如，字节地址 A 为 0x2000_0001，访问该地址的 bit1，地址为：
 $AliasAddress_A_N = 0x2200_0000 + 1 \times 32 + 1 \times 4 = 0x2200_0024$

3.5 外设寄存器

3.5.1 外设寄存器映射

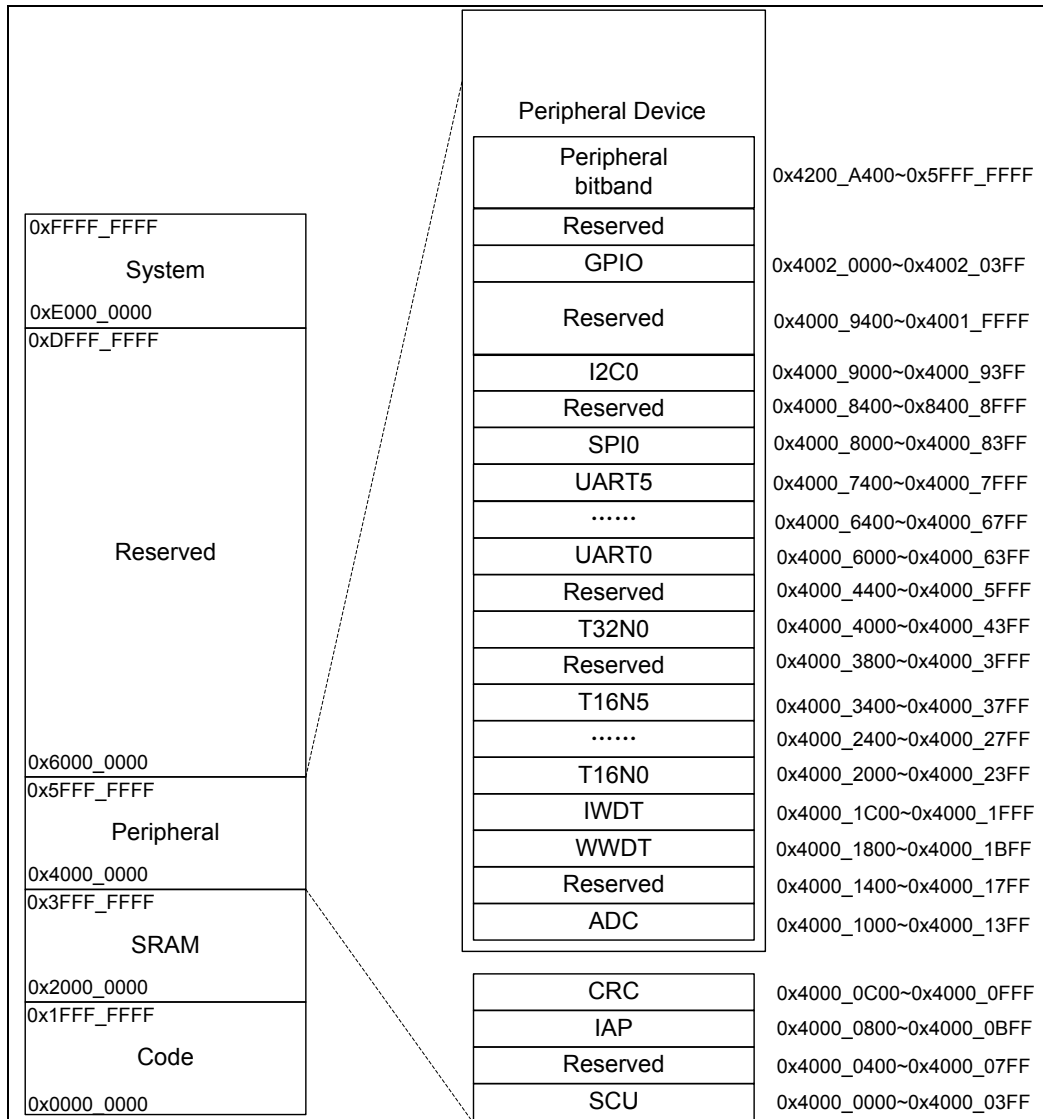


图 3-4 外设存储器分配示意图

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 _H		
SCU_PROT	0000 _H	系统设置保护寄存器
SCU_NMICON	0004 _H	不可屏蔽中断控制寄存器
SCU_PWRC	0008 _H	复位寄存器
SCU_FAULTFLAG	000C _H	硬件错误标志寄存器
SCU_WAKEUPTIME	0010 _H	系统唤醒时间控制寄存器
SCU_DBGHALT	0018 _H	调试模式模块暂停控制寄存器
SCU_FLASHWAIT	0020 _H	FLASH 访问等待时间寄存器
SCU_SOFTCFG	0024 _H	系统配置软件控制寄存器
SCU_LVDCON	0028 _H	LVD 控制寄存器
SCU_CCM	002C _H	外部时钟检测控制寄存器
SCU_PLLLKCON	0030 _H	PLL 锁定中断控制寄存器
SCU_SCLKEN0	0040 _H	系统时钟控制寄存器 0
SCU_SCLKEN1	0044 _H	系统时钟控制寄存器 1
SCU_PCLKEN0	0048 _H	外设时钟控制寄存器 0
SCU_PCLKEN1	004C _H	外设时钟控制寄存器 1
SCU_PRSTEN0	0050 _H	外设复位控制寄存器 0
SCU_PRSTEN1	0054 _H	外设复位控制寄存器 1
SCU_TIMEREN	0058 _H	TIMER 使能控制寄存器
SCU_TIMERDIS	005C _H	TIMER 关停控制寄存器
SCU_TBLREMAPEN	0060 _H	中断向量表重映射使能寄存器
SCU_TBLOFF	0064 _H	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 _H		
CRC_UL	0000 _H	CRC 解锁寄存器
CRC_CON	0004 _H	CRC 控制寄存器
CRC_TRIG	0008 _H	CRC 触发寄存器
CRC_ADDR	000C _H	CRC FLASH 校验起始地址寄存器
CRC_SIZE	0010 _H	CRC FLASH 校验数据块大小寄存器
CRC_DI	0014 _H	CRC 用户模式输入数据寄存器
CRC_DO	0018 _H	CRC 校验结果寄存器
CRC_STA	001C _H	CRC 状态寄存器
CRC_FA	0020 _H	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 _H		
GPIO_PAPORT	0000 _H	PA 端口状态寄存器
GPIO_PADATA	0010 _H	PA 端口数据寄存器
GPIO_PADATABSR	0014 _H	PA 输出置位寄存器
GPIO_PADATABCR	0018 _H	PA 输出清零寄存器
GPIO_PADATABRR	001C _H	PA 输出翻转寄存器
GPIO_PADIR	0020 _H	PA 端口方向控制寄存器
GPIO_PADIRBSR	0024 _H	PA 端口方向置位寄存器
GPIO_PADIRBCR	0028 _H	PA 端口方向清零寄存器
GPIO_PADIRBRR	002C _H	PA 端口方向翻转寄存器
GPIO_PAFUNC0	0030 _H	PA[7:0]端口复用选择寄存器
GPIO_PAFUNC1	0034 _H	PA[15:8]端口复用选择寄存器
GPIO_PAFUNC2	0038 _H	PA[23:16]端口复用选择寄存器
GPIO_PAFUNC3	003C _H	PA[31:24]端口复用选择寄存器
GPIO_PAINEB	0040 _H	PA 端口输入控制寄存器
GPIO_PAODE	0044 _H	PA 端口开漏控制寄存器
GPIO_PAPUE	0048 _H	PA 端口弱上拉使能寄存器
GPIO_PAPDE	004C _H	PA 端口弱下拉使能寄存器
GPIO_PADS	0050 _H	PA 端口驱动电流控制寄存器
GPIO_PATYP	0054 _H	PA 端口类型选择寄存器
GPIO_PAFLT	0058 _H	PA 端口滤波控制寄存器
GPIO_PBPORT	0080 _H	PB 端口状态寄存器
GPIO_PBDATA	0090 _H	PB 端口数据寄存器
GPIO_PBDATABSR	0094 _H	PB 端口输出置位寄存器

GPIO 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
GPIO_PBDATABCR	0098 _H	PB 端口输出清零寄存器
GPIO_PBDATABRR	009C _H	PB 端口输出翻转寄存器
GPIO_PBDIR	00A0 _H	PB 端口方向控制寄存器
GPIO_PBDIRBSR	00A4 _H	PB 端口方向置位寄存器
GPIO_PBDIRBCR	00A8 _H	PB 端口方向清零寄存器
GPIO_PBDIRBRR	00AC _H	PB 端口方向翻转寄存器
GPIO_PBFUNC0	00B0 _H	PB[7:0]端口复用选择寄存器
GPIO_PBFUNC1	00B4 _H	PB[15:8]端口复用选择寄存器
GPIO_PBFUNC2	00B8 _H	PB[23:16]端口复用选择寄存器
GPIO_PBFUNC3	00BC _H	PB[27:24]端口复用选择寄存器
GPIO_PBINEB	00C0 _H	PB 端口输入控制寄存器
GPIO_PBODE	00C4 _H	PB 端口开漏控制寄存器
GPIO_PBPUE	00C8 _H	PB 端口弱上拉使能寄存器
GPIO_PBPDE	00CC _H	PB 端口弱下拉使能寄存器
GPIO_PBDS	00D0 _H	PB 端口驱动电流控制寄存器
GPIO_PBTYP	00D4 _H	PB 端口类型选择寄存器
GPIO_PBFLT	00D8 _H	PB 端口滤波控制寄存器
GPIO_PINTIE	0300 _H	PINT 中断使能寄存器
GPIO_PINTIF	0304 _H	PINT 中断标志寄存器
GPIO_PINTSEL0	0308 _H	PINT0~7 中断源选择寄存器
GPIO_PINTSEL1	030C _H	PINT8~11 中断源选择寄存器
GPIO_PINTCFG0	0310 _H	PINT0~7 中断配置寄存器
GPIO_PINTCFG1	0314 _H	PINT8~11 中断配置寄存器
GPIO_KINTIE	0318 _H	KINT 中断使能寄存器
GPIO_KINTIF	031C _H	KINT 中断标志寄存器
GPIO_KINTSEL0	0320 _H	KINT0~7 中断源选择寄存器
GPIO_KINTSEL1	0324 _H	KINT8~11 中断源选择寄存器
GPIO_KINTCFG0	0328 _H	KINT0~7 中断配置寄存器
GPIO_KINTCFG1	032C _H	KINT8~11 中断配置寄存器
GPIO_TXPWM	0380 _H	脉宽调制寄存器
GPIO_BUZC	0390 _H	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 _H		
IAP_FLASHKEY	000 _H	FLASH 程序区操作关键码寄存器
—	004 _H	保留
IAP_FLASHADDR	008 _H	FLASH 擦除编程地址寄存器

IAP 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
IAP_FLASHFIFO	00C _H	FLASH 编程数据 FIFO 寄存器
IAP_FLASHDR	010 _H	FLASH 编程数据寄存器
—	014 _H	保留
IAP_FLASHCMD	018 _H	FLASH 操作命令寄存器
IAP_FLASHCR	01C _H	FLASH 控制寄存器
IAP_FLASHSR	020 _H	FLASH 状态寄存器
—	024 _H	保留
—	028 _H	保留
IAP_FLASHADDINV	02C _H	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 _H		
ADC_DR	0000 _H	ADC 转换值寄存器
ADC_CON0	0004 _H	ADC 控制寄存器 0
ADC_CON1	0008 _H	ADC 控制寄存器 1
ADC_CHS	000C _H	ADC 通道选择寄存器
ADC_IE	0010 _H	ADC 中断使能寄存器
ADC_IF	0014 _H	ADC 中断标志寄存器
ADC_ACPC	0028 _H	ADC 自动转换比较控制寄存器
ADC_ACPCMP	0030 _H	ADC 自动转换比较阈值寄存器
ADC_ACPMEAN	0034 _H	ADC 自动转换均值数据寄存器
ADC_VREFCON	0040 _H	ADC 参考控制寄存器

3.5.8 IWDTC寄存器列表

IWDTC 寄存器列表		
寄存器名称	寄存器地址	寄存器描述
IWDTC 基地址: 4000_1C00 _H		
IWDTC_LOAD	0000 _H	IWDTC 计数器装载值寄存器
IWDTC_VALUE	0004 _H	IWDTC 计数器当前值寄存器
IWDTC_CON	0008 _H	IWDTC 控制寄存器
IWDTC_INTCLR	000C _H	IWDTC 中断标志清除寄存器
IWDTC_RIS	0010 _H	IWDTC 中断标志寄存器
IWDTC_LOCK	0100 _H	IWDTC 锁定寄存器

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 _H		
WWDT_LOAD	0000 _H	WWDT 计数器装载值寄存器
WWDT_VALUE	0004 _H	WWDT 计数器当前值寄存器
WWDT_CON	0008 _H	WWDT 控制寄存器
WWDT_INTCLR	000C _H	WWDT 中断标志清除寄存器
WWDT_RIS	0010 _H	WWDT 中断标志寄存器
WWDT_LOCK	0100 _H	WWDT 锁定寄存器

3.5.10 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 _H		
T16N1 基地址: 4000_2400 _H		
T16N2 基地址: 4000_2800 _H		
T16N3 基地址: 4000_2C00 _H		
T16N4 基地址: 4000_3000 _H		
T16N5 基地址: 4000_3400 _H		
T16N_CNT	0000 _H	T16N 计数值寄存器
T16N_PRECNT	0008 _H	T16N 预分频器计数值寄存器
T16N_PREMAT	000C _H	T16N 预分频器计数匹配寄存器
T16N_CON0	0010 _H	T16N 控制寄存器 0
T16N_CON1	0014 _H	T16N 控制寄存器 1
T16N_CON2	0018 _H	T16N 控制寄存器 2
T16N_IE	0020 _H	T16N 中断使能寄存器
T16N_IF	0024 _H	T16N 中断标志寄存器
T16N_TRG	0028 _H	T16N 触发寄存器
T16N_MAT0	0030 _H	T16N 计数匹配寄存器 0
T16N_MAT1	0034 _H	T16N 计数匹配寄存器 1
T16N_MAT2	0038 _H	T16N 计数匹配寄存器 2
T16N_MAT3	003C _H	T16N 计数匹配寄存器 3
T16N_TOP	0040 _H	T16N 计数峰值寄存器

3.5.11 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 _H		
T32N_CNT	0000 _H	T32N 计数值寄存器
T32N_PRECNT	0008 _H	T32N 预分频器计数值寄存器
T32N_PREMAT	000C _H	T32N 预分频器计数匹配寄存器
T32N_CON0	0010 _H	T32N 控制寄存器 0
T32N_CON1	0014 _H	T32N 控制寄存器 1
T32N_CON2	0018 _H	T32N 控制寄存器 2
T32N_IE	0020 _H	T32N 中断使能寄存器
T32N_IF	0024 _H	T32N 中断标志寄存器
T32N_TRG	0028 _H	T32N 触发寄存器
T32N_MAT0	0030 _H	T32N 计数匹配寄存器 0
T32N_MAT1	0034 _H	T32N 计数匹配寄存器 1
T32N_MAT2	0038 _H	T32N 计数匹配寄存器 2
T32N_MAT3	003C _H	T32N 计数匹配寄存器 3

3.5.12 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 _H		
UART1 基地址: 4000_6400 _H		
UART2 基地址: 4000_6800 _H		
UART3 基地址: 4000_6C00 _H		
UART4 基地址: 4000_7000 _H		
UART5 基地址: 4000_7400 _H		
UART_CON	0000 _H	UART 控制寄存器
UART_BRR	0004 _H	UART 波特率寄存器
UART_TBW	0008 _H	UART 发送数据写入寄存器
UART_RBR	000C _H	UART 接收数据读取寄存器
UART_TB01	0010 _H	UART 发送缓冲 0/1 寄存器
UART_TB23	0014 _H	UART 发送缓冲 2/3 寄存器
UART_RB01	0018 _H	UART 接收缓冲 0/1 寄存器
UART_RB23	001C _H	UART 接收缓冲 2/3 寄存器
UART_IE	0020 _H	UART 中断使能寄存器
UART_IF	0024 _H	UART 中断标志寄存器

3.5.13 SPI 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 _H		
SPI_CON	0000 _H	SPI 控制寄存器
SPI_TBW	0008 _H	SPI 发送数据写入寄存器
SPI_RBR	000C _H	SPI 接收数据读取寄存器
SPI_IE	0010 _H	SPI 中断使能寄存器
SPI_IF	0014 _H	SPI 中断标志寄存器
SPI_TB	0018 _H	SPI 发送缓冲寄存器
SPI_RB	001C _H	SPI 接收缓冲寄存器
SPI_STA	0020 _H	SPI 状态寄存器
SPI_CKS	0024 _H	SPI 波特率设置寄存器

3.5.14 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 _H		
I2C_CON	0000 _H	I2C 控制寄存器
I2C_MOD	0004 _H	I2C 工作模式寄存器
I2C_IE	0008 _H	I2C 中断使能寄存器
I2C_IF	000C _H	I2C 中断标志寄存器
I2C_TBW	0010 _H	I2C 发送数据写入寄存器
I2C_RBR	0014 _H	I2C 接收数据读取寄存器
I2C_TB	0018 _H	I2C 发送缓冲寄存器
I2C_RB	001C _H	I2C 接收缓冲寄存器
I2C_STA	0020 _H	I2C 状态寄存器

3.6 内核寄存器

3.6.1 系统定时器 (SYSTICK) 寄存器列表

系统定时器 (SYSTICK)		
寄存器名称	寄存器地址	寄存器描述
SYSTICK 基地址: E000_E000 _H		
SYST_CSR	0010 _H	SYSTICK 控制和状态寄存器
SYST_RVR	0014 _H	SYSTICK 重装值寄存器
SYST_CVR	0018 _H	SYSTICK 当前值寄存器
SYST_CALIB	001C _H	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 _H		
NVIC_ISER	0000 _H	IRQ0~31 置中断请求使能寄存器
NVIC_ICER	0080 _H	IRQ0~31 清中断请求使能寄存器
NVIC_ISPR	0100 _H	IRQ0~31 置中断挂起寄存器
NVIC_ICPR	0180 _H	IRQ0~31 清中断挂起寄存器
NVIC_PR0	0300 _H	IRQ0~3 优先级控制寄存器
NVIC_PR1	0304 _H	IRQ4~7 优先级控制寄存器
NVIC_PR2	0308 _H	IRQ8~11 优先级控制寄存器
NVIC_PR3	030C _H	IRQ12~15 优先级控制寄存器
NVIC_PR4	0310 _H	IRQ16~19 优先级控制寄存器
NVIC_PR5	0314 _H	IRQ20~23 优先级控制寄存器
NVIC_PR6	0318 _H	IRQ24~27 优先级控制寄存器
NVIC_PR7	031C _H	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 _H		
SCB_CPUID	0000 _H	SCB_CPUID 寄存器
SCB_ICSR	0004 _H	中断控制和状态寄存器
SCB_AIRCR	000C _H	应用中断和复位控制寄存器
SCB_SCR	0010 _H	系统控制寄存器
SCB_CCR	0014 _H	配置和控制寄存器
SCB_SHPR2	001C _H	系统处理程序优先级寄存器 2
SCB_SHPR3	0020 _H	系统处理程序优先级寄存器 3

第 4 章 输入输出端口 (GPIO)

4.1 概述

本芯片支持两组 GPIO 端口，最多支持共 30 个 I/O 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_PAFLT/GPIO_PBFLT 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均必须大于 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: 本芯片总共有 31 个 GPIO,但有 1 个 GPIO 的位置会封装为 MRSTN 管脚,所以芯片实际能支持的 GPIO 最多 30 个。

4.2 结构框图

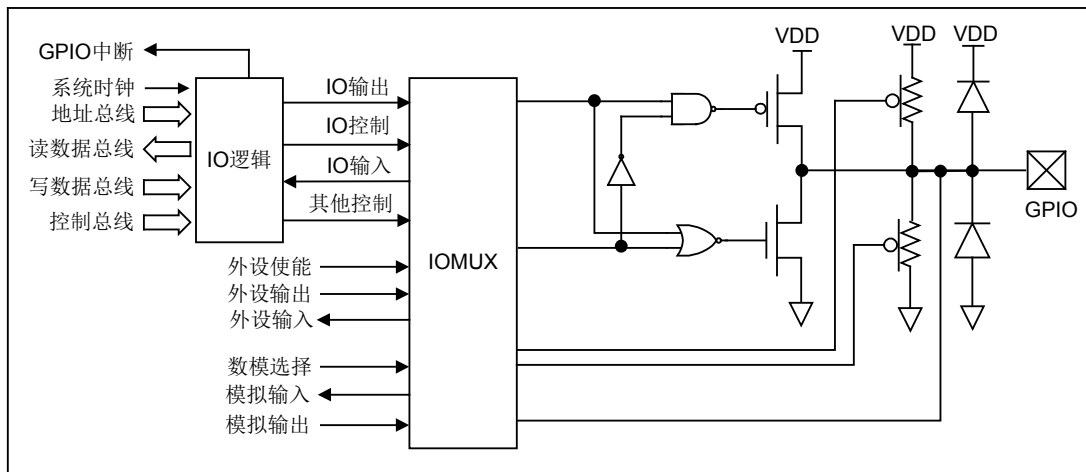


图 4-1 IO 端口电流结构图

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。

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。

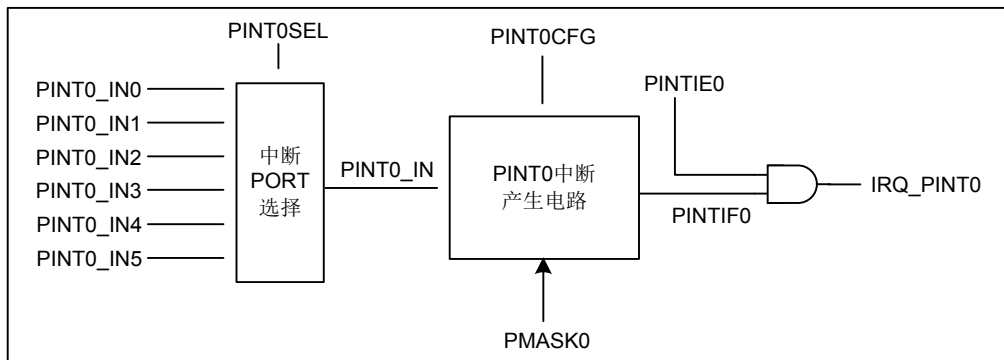


图 4-2 外部端口中断 PINT0 电路结构示意图

上图以外外部端口中断 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，PB16 和 PB10。PINT0 ~PINT11 的外部中断源参见下表：

PINT _x	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	—	—	—
PINT9	PB1	PB7	PB25	—	—	—
PINT10	PB18	PB21	PB26	—	—	—
PINT11	PB19	PB23	PB27	—	—	—

表 4-1 PINT 选择分组对应列表

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。

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。

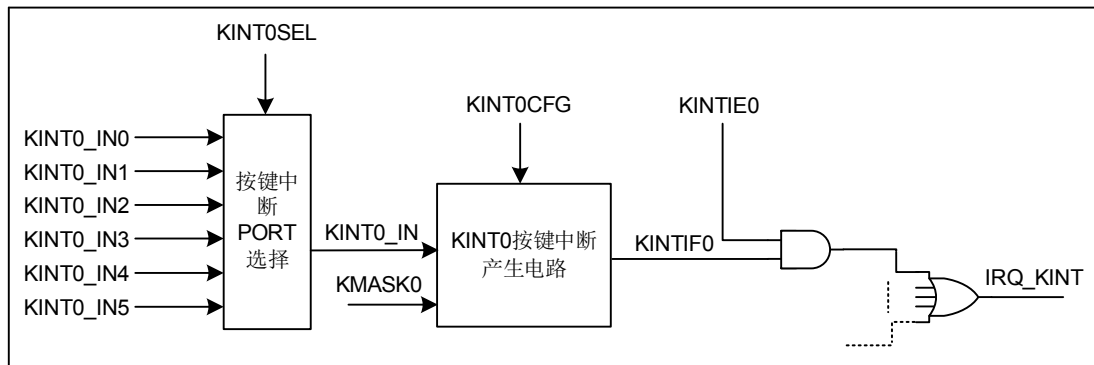


图 4-3 外部按键中断 KINT 电路结构示意图

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、PB16 和 PB10。KINT0 ~KINT11 的按键中断源如下表所示：

KINT _x	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	—	—	—
KINT9	PB1	PB7	PB25	—	—	—
KINT10	PB18	PB21	PB26	—	—	—
KINT11	PB19	PB23	PB27	—	—	—

表 4-2 KINT 选择分组对应列表

4.5 Buzz输出

Buzz 输出可用于驱动蜂鸣器等音频发声器件。

GPIO_BUZC 控制寄存器用于使能 Buzz 和设定其输出信号的频率。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。

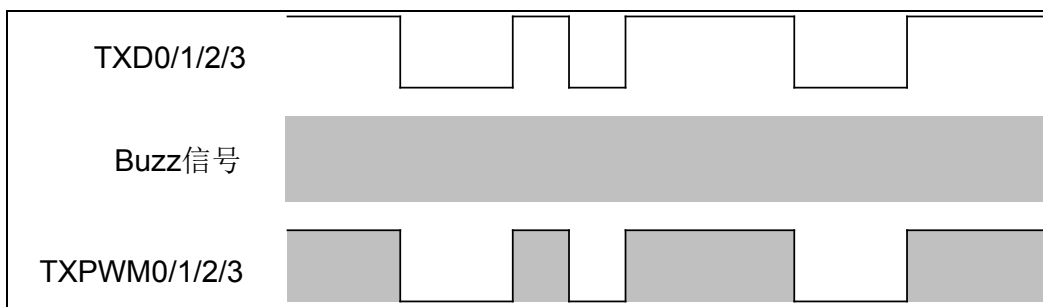


图 4-4 Buzz 高电平调制输出波形图

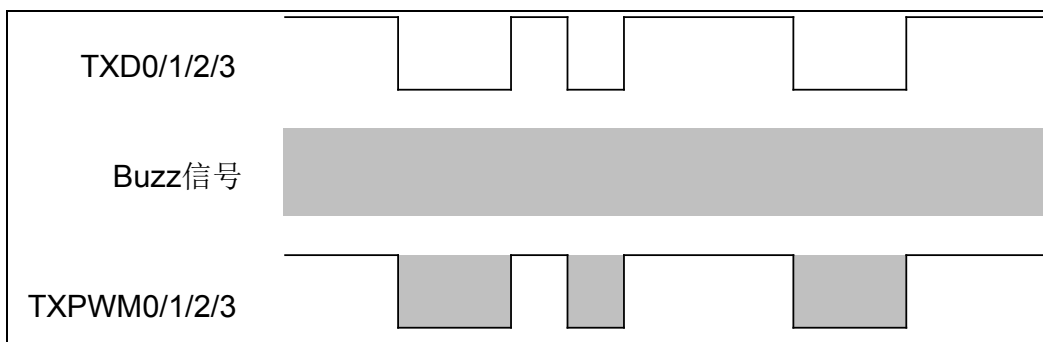


图 4-5 Buzz 低电平调制输出波形图

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 _H															
复位值: xxxxxxxx_xxxxxxxx_xxxxxxxx_xxxxxxxx _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
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 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
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	PA 输出清零选择 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 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
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>															

DIR<31:0>	bit 31-0	W	PA 输出翻转选择 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 _H															
复位值: 11111111_11111111_11111111_11111111 _B															
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	PA 端口方向控制位 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)

PA 端口方向置位寄存器 (GPIO_PADIRBSR)															
偏移地址: 24 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
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	PA 端口方向置位选择 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, 读数为 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)

PA 端口方向清零寄存器 (GPIO_PADIRBCR)															
偏移地址: 28 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
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	PA 端口方向清零选择 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, 读数为 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 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
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	PA 端口方向翻转选择 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, 读数为 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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>		
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>		

—	bit31	—	—
PA7<2:0>	bit30-28	R/W	PA7 复用选择位 000~111: FUN0~FUN7
—	bit27	—	—
PA6<2:0>	bit26-24	R/W	PA6 复用选择位 000~111: FUN0~FUN7
—	bit23	—	—
PA5<2:0>	bit22-20	R/W	PA5 复用选择位 000~111: FUN0~FUN7
—	bit19	—	—
PA4<2:0>	bit18-16	R/W	PA4 复用选择位 000~111: FUN0~FUN7
—	bit15	—	—
PA3<2:0>	bit14-12	R/W	PA3 复用选择位 000~111: FUN0~FUN7
—	bit11	—	—
PA2<2:0>	bit10-8	R/W	PA2 复用选择位 000~111: FUN0~FUN7
—	bit7	—	—
PA1<2:0>	bit6-4	R/W	PA1 复用选择位 000~111: FUN0~FUN7
—	bit3	—	—
PA0<2:0>	bit2-0	R/W	PA0 复用选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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>					
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>					

—	bit31	—	—
PA15<2:0>	bit30-28	R/W	PA15 复用选择位 000~111: FUN0~FUN7
—	bit27	—	—
PA14<2:0>	bit26-24	R/W	PA14 复用选择位 000~111: FUN0~FUN7
—	bit23	—	—
PA13<2:0>	bit22-20	R/W	PA13 复用选择位 000~111: FUN0~FUN7
—	bit19	—	—
PA12<2:0>	bit18-16	R/W	PA12 复用选择位 000~111: FUN0~FUN7
—	bit15	—	—
PA11<2:0>	bit14-12	R/W	PA11 复用选择位 000~111: FUN0~FUN7
—	bit11	—	—
PA10<2:0>	bit10-8	R/W	PA10 复用选择位 000~111: FUN0~FUN7
—	bit7	—	—
PA9<2:0>	bit6-4	R/W	PA9 复用选择位 000~111: FUN0~FUN7
—	bit3	—	—
PA8<2:0>	bit2-0	R/W	PA8 复用选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	PA23 复用选择位 000~111: FUN0~FUN7
—	bit27	—	—
PA22<2:0>	bit26-24	R/W	PA22 复用选择位 000~111: FUN0~FUN7
—	bit23	—	—
PA21<2:0>	bit22-20	R/W	PA21 复用选择位 000~111: FUN0~FUN7
—	bit19	—	—
PA20<2:0>	bit18-16	R/W	PA20 复用选择位 000~111: FUN0~FUN7
—	bit15	—	—
PA19<2:0>	bit14-12	R/W	PA19 复用选择位 000~111: FUN0~FUN7
—	bit11	—	—
PA18<2:0>	bit10-8	R/W	PA18 复用选择位 000~111: FUN0~FUN7
—	bit7	—	—
PA17<2:0>	bit6-4	R/W	PA17 复用选择位 000~111: FUN0~FUN7
—	bit3	—	—
PA16<2:0>	bit2-0	R/W	PA16 复用选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	PA27<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>		保留	PA24<2:0>		保留

—	bit31	—	—
PA31<2:0>	bit30-28	R/W	PA31 复用选择位 000~111: FUN0~FUN7
—	bit27	—	—
PA30<2:0>	bit26-24	R/W	PA30 复用选择位 000~111: FUN0~FUN7
—	bit23	—	—
PA29<2:0>	bit22-20	R/W	PA29 复用选择位 000~111: FUN0~FUN7
—	bit19	—	—
PA28<2:0>	bit18-16	R/W	PA28 复用选择位 000~111: FUN0~FUN7
—	bit15	—	—
PA27<2:0>	bit14-12	R/W	PA27 复用选择位 000~111: FUN0~FUN7
—	bit11	—	—
PA26<2:0>	bit10-8	R/W	PA26 复用选择位 000~111: FUN0~FUN7
—	bit7	—	—
PA25<2:0>	bit6-4	R/W	PA25 复用选择位 000~111: FUN0~FUN7
—	bit3	—	—
PA24<2:0>	bit2-0	R/W	PA24 复用选择位 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: 40 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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

注: PA19 端口在芯片 VDD 上电过程中, 内部弱上拉自动使能, 端口电平跟随 VDD 上升, 直到 VDD 上升到芯片开始工作的电压后, 该 IO 端口的内部弱上拉恢复为默认的禁止状态。

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DS <31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS <15:0>															

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

4.6.19 PA端口类型选择寄存器 (GPIO_PATYP)

PA 端口类型寄存器 (GPIO_PATYP)															
偏移地址: 54 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYP <31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYP <15:0>															

TYP<31:0>	bit31-0	R/W	PA 端口类型选择位 0: CMOS 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT <31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT <15:0>															

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

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 _H															
复位值: 00000000_00000000_00xxxxxx_xxxxxxxx _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				PORT<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORT<15:0>															
—	bit31-28			—	—										
PORT<27:0>				bit27-0	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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DATA<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA<15:0>															
—	bit31-28			—	—										
DATA<27:0>				bit27-0	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 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DATABSR<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATABSR<15:0>															

—	bit31-28	—	—
DATABSR<27:0>	bit27-0	W	PB 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_PBDATABSR 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。

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

—	bit31-28	—	—
DATABCR<27:0>	bit27-0	W	PB 输出清零选择 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, 读数为 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 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DATABRR<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATABRR<15:0>															
—				bit31-28		—				—					
DATABRR<27:0>				bit27-0		W		PB 输出翻转选择 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, 读数为 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 _H															
复位值: 11111111_11111111_11111111_11111111 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DIR<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR<15:0>															
—				bit31-28		—				—					
DIR<27:0>				bit27-0		RW		PB 端口方向控制位 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 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DIRBSR<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRBSR<15:0>															
—				bit31-28		—				—					
DIRBSR<27:0>				bit27-0		W		PB 端口方向置位选择 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, 读数为 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 _H															
复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DIRBCR<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRBCR<15:0>															
—				bit31-28		—				—					
DIRBCR<27:0>				bit27-0		W		PB 端口方向清零选择 0: 不改变 GPIO_PBDIR 的值 1: 对应的 GPIO_PBDIR 位设置为 0							

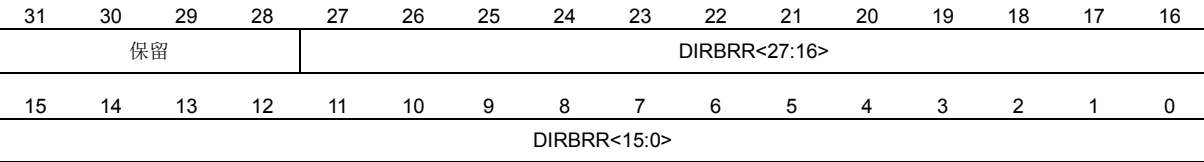
注: GPIO_PBDIRBCR 寄存器仅支持 Word 写入的方式, 读数为 0, 对其进行 Bitband 操作无效。

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_H

复位值: XXXXXXXX_XXXXXXX_XXXXXXX_XXXXXXX_B



—	bit31-28	—	—
DIRBRR<27:0>	bit27-0	W	PB 端口方向翻转选择 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, 读数为 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	PB7 复用选择位 000~111: FUN0~FUN7
—	bit27	—	—
PB6<2:0>	bit26-24	R/W	PB6 复用选择位 000~111: FUN0~FUN7
—	bit23	—	—
PB5<2:0>	bit22-20	R/W	PB5 复用选择位 000~111: FUN0~FUN7
—	bit19	—	—
PB4<2:0>	bit18-16	R/W	PB4 复用选择位 000~111: FUN0~FUN7
—	bit15	—	—
PB3<2:0>	bit14-12	R/W	PB3 复用选择位 000~111: FUN0~FUN7
—	bit11	—	—
PB2<2:0>	bit10-8	R/W	PB2 复用选择位 000~111: FUN0~FUN7
—	bit7	—	—
PB1<2:0>	bit6-4	R/W	PB1 复用选择位 000~111: FUN0~FUN7
—	bit3	—	—
PB0<2:0>	bit2-0	R/W	PB0 复用选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	PB15 复用选择位 000~111: FUN0~FUN7
—	bit27	—	—
PB14<2:0>	bit26-24	R/W	PB14 复用选择位 000~111: FUN0~FUN7
—	bit23	—	—
PB13<2:0>	bit22-20	R/W	PB13 复用选择位 000~111: FUN0~FUN7
—	bit19	—	—
PB12<2:0>	bit18-16	R/W	PB12 复用选择位 000~111: FUN0~FUN7
—	bit15	—	—
PB11<2:0>	bit14-12	R/W	PB11 复用选择位 000~111: FUN0~FUN7
—	bit11	—	—
PB10<2:0>	bit10-8	R/W	PB10 复用选择位 000~111: FUN0~FUN7
—	bit7	—	—
PB9<2:0>	bit6-4	R/W	PB9 复用选择位 000~111: FUN0~FUN7
—	bit3	—	—
PB8<2:0>	bit2-0	R/W	PB8 复用选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	PB23 复用选择位 000~111: FUN0~FUN7
—	bit27	—	—
PB22<2:0>	bit26-24	R/W	PB22 复用选择位 000~111: FUN0~FUN7
—	bit23	—	—
PB21<2:0>	bit22-20	R/W	PB21 复用选择位 000~111: FUN0~FUN7
—	bit19	—	—
PB20<2:0>	bit18-16	R/W	PB20 复用选择位 000~111: FUN0~FUN7
—	bit15	—	—
PB19<2:0>	bit14-12	R/W	PB19 复用选择位 000~111: FUN0~FUN7
—	bit11	—	—
PB18<2:0>	bit10-8	R/W	PB18 复用选择位 000~111: FUN0~FUN7
—	bit7	—	—
PB17<2:0>	bit6-4	R/W	PB17 复用选择位 000~111: FUN0~FUN7
—	bit3	—	—
PB16<2:0>	bit2-0	R/W	PB16 复用选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PB27<2:0>		保留	PB26<2:0>		保留	PB25<2:0>		保留	PB24<2:0>					

—	bit31-15	—	—
PB27<2:0>	bit14-12	R/W	PB27 复用选择位 000~111: FUN0~FUN7
—	bit11	—	—
PB26<2:0>	bit10-8	R/W	PB26 复用选择位 000~111: FUN0~FUN7
—	bit7	—	—
PB25<2:0>	bit6-4	R/W	PB25 复用选择位 000~111: FUN0~FUN7
—	bit3	—	—
PB24<2:0>	bit2-0	R/W	PB24 复用选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				INEB<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEB<15:0>															

—	bit31-28	—	—
INEB<27:0>	bit27-0	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 _H																
复位值: 00000000_00000000_00000000_00000000 _B																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留				ODE<27:16>												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ODE<15:0>																
—	bit31-28			—	—											
ODE<27:0>				bit27-0			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 _H																
复位值: 00000000_00000000_00000000_00000000 _B																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留				PUEN<27:16>												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PUEN<15:0>																
—	bit31-28			—	—											
PUEN<27:0>				bit27-0			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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				PDEN<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDEN<15:0>															
—	bit31-28			—	—										
PDEN<27:0>				bit27-0		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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DS<27:16>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS<15:0>															
—	bit31-28			—	—										
DS<27:0>				bit27-0		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_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-28	—	—
TYP<27:0>	bit27-0	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_PBFILT)

PB 端口 20ns 滤波寄存器 (GPIO_PBFILT)

偏移地址: D8_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-28	—	—
FLT<27:0>	bit27-0	R/W	PB 端口 20ns 滤波使能位 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 _H															
复位值: 00000000_11110000_11111111_00000000 _B															
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	PINT11~8 中断源屏蔽控制位 0: 不屏蔽 1: 屏蔽
PINTIE<11:8>	bit19-16	R/W	PINT11~8 使能位 0: 禁止 1: 使能
PMASK<7:0>	bit15-8	R/W	PINT7~0 中断源屏蔽控制位 0: 不屏蔽 1: 屏蔽
PINTIE<7:0>	bit7-0	R/W	PINT7~0 使能位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					PINTIF<11:0>										

—	bit 31-12	—	—
PINTIF<11:0>	bit11-0	R/W	GPIO 外部中断标志位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	PINT7 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit27	—	—
PINT6SEL<2:0>	bit26-24	R/W	PINT6 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit23	—	—
PINT5SEL<2:0>	bit22-20	R/W	PINT5 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit19	—	—
PINT4SEL<2:0>	bit18-16	R/W	PINT4 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit15	—	—
PINT3SEL<2:0>	bit14-12	R/W	PINT3 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit11	—	—
PINT2SEL<2:0>	bit10-8	R/W	PINT2 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit7	—	—
PINT1SEL<2:0>	bit6-4	R/W	PINT1 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit3	—	—
PINT0SEL<2:0>	bit2-0	R/W	PINT0 输入选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	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	PINT11 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit11	—	—
PINT10SEL<2:0>	bit10-8	R/W	PINT10 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit7	—	—
PINT9SEL<2:0>	bit6-4	R/W	PINT9 输入选择位 000~101: SEL0~SEL5 其余: SEL0
—	bit3	—	—
PINT8SEL<2:0>	bit2-0	R/W	PINT8 输入选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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>		保留	PINT3<2:0>		保留
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>		保留	PINT0<2:0>		保留

—	bit31	—	—
PINT7CFG<2:0>	bit30-28	R/W	PINT7 配置位 参见 GPIO_PINTCFG 详情
—	bit27	—	—
PINT6CFG<2:0>	bit26-24	R/W	PINT6 配置位 参见 GPIO_PINTCFG 详情
—	bit23	—	—
PINT5CFG<2:0>	bit22-20	R/W	PINT5 配置位 参见 GPIO_PINTCFG 详情
—	bit19	—	—
PINT4CFG<2:0>	bit18-16	R/W	PINT4 配置位 参见 GPIO_PINTCFG 详情
—	bit15	—	—
PINT3CFG<2:0>	bit14-12	R/W	PINT3 配置位 参见 GPIO_PINTCFG 详情
—	bit11	—	—
PINT2CFG<2:0>	bit10-8	R/W	PINT2 配置位 参见 GPIO_PINTCFG 详情
—	bit7	—	—
PINT1CFG<2:0>	bit6-4	R/W	PINT1 配置位 参见 GPIO_PINTCFG 详情
—	bit3	—	—
PINT0CFG<2:0>	bit2-0	R/W	PINT0 配置位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	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	PINT11 配置位 参见 GPIO_PINTCFG 详情
—	bit11	—	—
PINT10CFG<2:0>	bit10-8	R/W	PINT10 配置位 参见 GPIO_PINTCFG 详情
—	bit7	—	—
PINT9CFG<2:0>	bit6-4	R/W	PINT9 配置位 参见 GPIO_PINTCFG 详情
—	bit3	—	—
PINT8CFG<2:0>	bit2-0	R/W	PINT8 配置位 参见 GPIO_PINTCFG 详情

寄存器名称	GPIO_PINTCFG 详情		
PINTCFG<2:0>	bit 2-0	R/W	GPIO_PINTCFG 配置位 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 _H															
复位值: 00000000_11110000_11111111_00000000 _B															
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	KINT11~8 中断源屏蔽控制位 0: 不屏蔽 1: 屏蔽
KINTIE<11:8>	bit19-16	R/W	KINT11~8 使能位 0: 禁止 1: 使能
KMASK<7:0>	bit15-8	R/W	KINT7~0 中断源屏蔽控制位 0: 不屏蔽 1: 屏蔽
KINTIE<7:0>	bit7-0	R/W	KINT7~0 使能位 0: 禁止 1: 使能

4.6.48 KINT中断标志寄存器 (GPIO_KINTIF)

KINT 中断标志寄存器 (GPIO_KINTIF)															
偏移地址: 31C _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				KINTIF<11:0>											

—	bit 31-12	—	—
KINTIF<11:0>	bit11-0	R/W	GPIO 按键中断标志位 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)

KINT0~7 中断源选择寄存器 (GPIO_KINTSEL0)															
偏移地址: 320 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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>					
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>					

—	bit31	—	—
KINT7SEL<2:0>	bit30-28	R/W	KINT7 输入选择位 000~101: SEL0~SEL5
—	bit27	—	—
KINT6SEL<2:0>	bit26-24	R/W	KINT6 输入选择位 000~101: SEL0~SEL5
—	bit23	—	—
KINT5SEL<2:0>	bit22-20	R/W	KINT5 输入选择位 000~101: SEL0~SEL5
—	bit19	—	—
KINT4SEL<2:0>	bit18-16	R/W	KINT4 输入选择位 000~101: SEL0~SEL5
—	bit15	—	—
KINT3SEL<2:0>	bit14-12	R/W	KINT3 输入选择位 000~101: SEL0~SEL5
—	bit11	—	—
KINT2SEL<2:0>	bit10-8	R/W	KINT2 输入选择位 000~101: SEL0~SEL5
—	bit7	—	—
KINT1SEL<2:0>	bit6-4	R/W	KINT1 输入选择位 000~101: SEL0~SEL5
—	bit3	—	—
KINT0SEL<2:0>	bit2-0	R/W	KINT0 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	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	KINT11 输入选择位 000~101: SEL0~SEL5
—	bit11	—	—
KINT10SEL<2:0>	bit10-8	R/W	KINT10 输入选择位 000~101: SEL0~SEL5
—	bit7	—	—
KINT9SEL<2:0>	bit6-4	R/W	KINT9 输入选择位 000~101: SEL0~SEL5
—	bit3	—	—
KINT8SEL<2:0>	bit2-0	R/W	KINT8 输入选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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>		
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>		

—	bit31	—	—
KINT7CFG<2:0>	bit30-28	R/W	KINT7 配置位 参见 GPIO_KINTCFG 详情
—	bit27	—	—
KINT6CFG<2:0>	bit26-24	R/W	KINT6 配置位 参见 GPIO_KINTCFG 详情
—	bit23	—	—
KINT5CFG<2:0>	bit22-20	R/W	KINT5 配置位 参见 GPIO_KINTCFG 详情
—	bit19	—	—
KINT4CFG<2:0>	bit18-16	R/W	KINT4 配置位 参见 GPIO_KINTCFG 详情
—	bit15	—	—
KINT3CFG<2:0>	bit14-12	R/W	KINT3 配置位 参见 GPIO_KINTCFG 详情
—	bit11	—	—
KINT2CFG<2:0>	bit10-8	R/W	KINT2 配置位 参见 GPIO_KINTCFG 详情
—	bit7	—	—
KINT1CFG<2:0>	bit6-4	R/W	KINT1 配置位 参见 GPIO_KINTCFG 详情
—	bit3	—	—
KINT0CFG<2:0>	bit2-0	R/W	KINT0 配置位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	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	KINT11 配置位 参见 GPIO_KINTCFG 详情
—	bit11	—	—
KINT10CFG<2:0>	bit10-8	R/W	KINT10 配置位 参见 GPIO_KINTCFG 详情
—	bit7	—	—
KINT9CFG<2:0>	bit6-4	R/W	KINT9 配置位 参见 GPIO_KINTCFG 详情
—	bit3	—	—
KINT8CFG<2:0>	bit2-0	R/W	KINT8 配置位 参见 GPIO_KINTCFG 详情

寄存器名称	GPIO_KINTCFG 详情		
GPIO_KINTCFG	bit 2-0	R/W	GPIO_KINTCFG 配置位 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 脉宽调制寄存器 (GPIO_TXPWM)

脉宽调制寄存器 (GPIO_TXPWM)

偏移地址: 380_H

复位值: 00000000_00000000_00000000_00000000_b

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>	保留	TX1_S<1:0>	TX1PLV	保留	TX1PS<1:0>	保留
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>	保留	TX0PS<1:0>	保留	TX0PS<1:0>	保留	TX0PS<1:0>

—	bit31	—	—
TX3_S<1:0>	bit30-29	R/W	TXPWM3 调制输出选择 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	TXPWM3 调制方式选择位 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	TXPWM3 调制 PWM 脉冲选择位 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	TXPWM2 调制输出选择 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	TXPWM2 调制方式选择位 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	TXPWM2 调制 PWM 脉冲选择位 00: 调制禁止 01: 与 BUZ 进行调制 10: 与 T16N2OUT0 进行调制

—	bit15	—	—
TX1_S<1:0>	bit14-13	R/W	TXPWM1 调制输出选择 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	TXPWM1 调制方式选择位 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	TXPWM1 调制 PWM 脉冲选择位 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	TXPWM0 调制输出选择 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	TXPWM0 调制方式选择位 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	TXPWM0 调制 PWM 脉冲选择位 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、T16NxOUT0、T16NxOUT1 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。

4.6.54 BUZ控制寄存器 (GPIO_BUZC)

BUZ 控制寄存器 (GPIO_BUZC)															
偏移地址: 390 _H															
复位值: 00001111_11111111_11111111_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				BUZ_LOAD<19:8>											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUZ_LOAD<7:0>								保留						BUZEN	

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

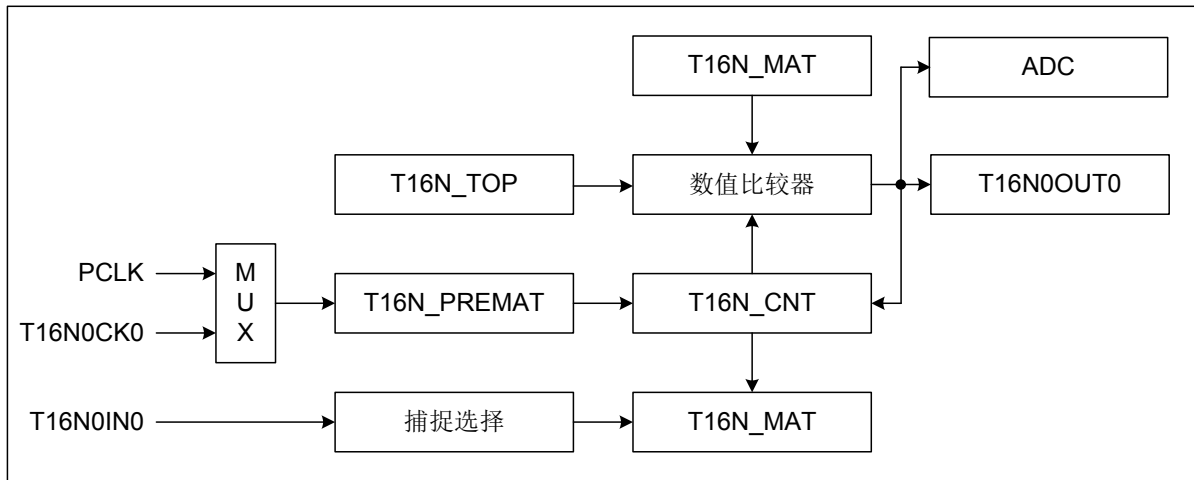


图 5-1 T16N0 电路结构框图

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：

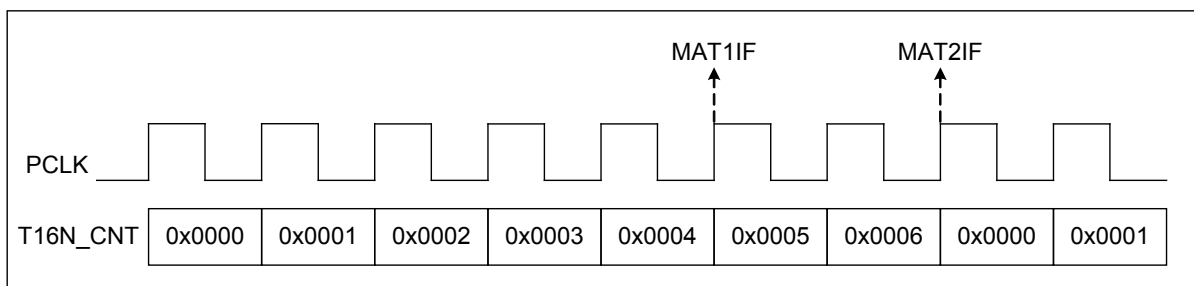


图 5-2 T16N 计数匹配功能示意图

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。

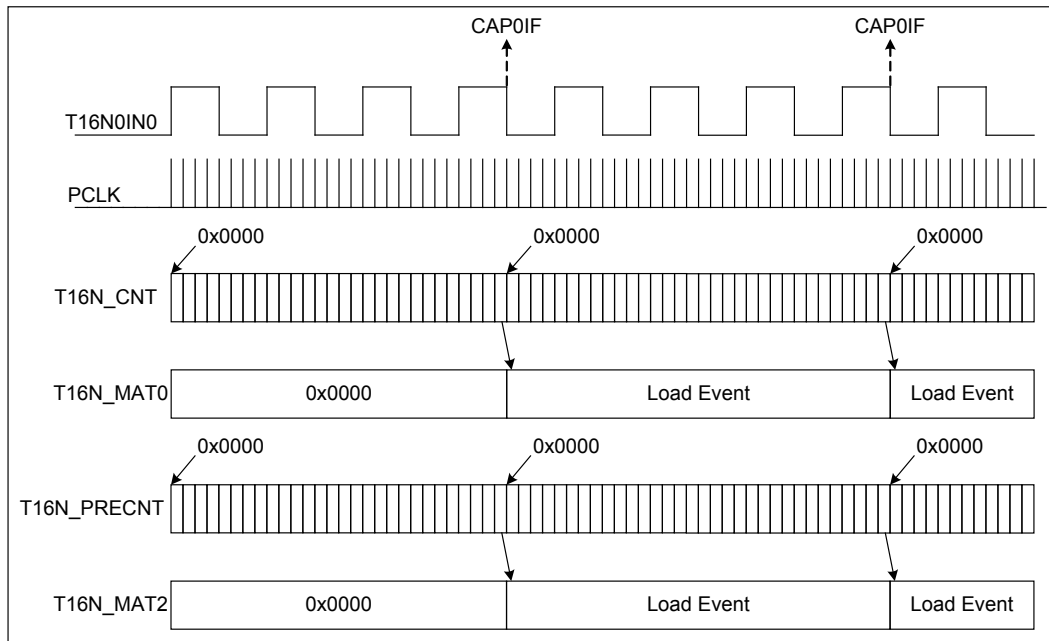


图 5-3 T16N0 捕捉功能示意图

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

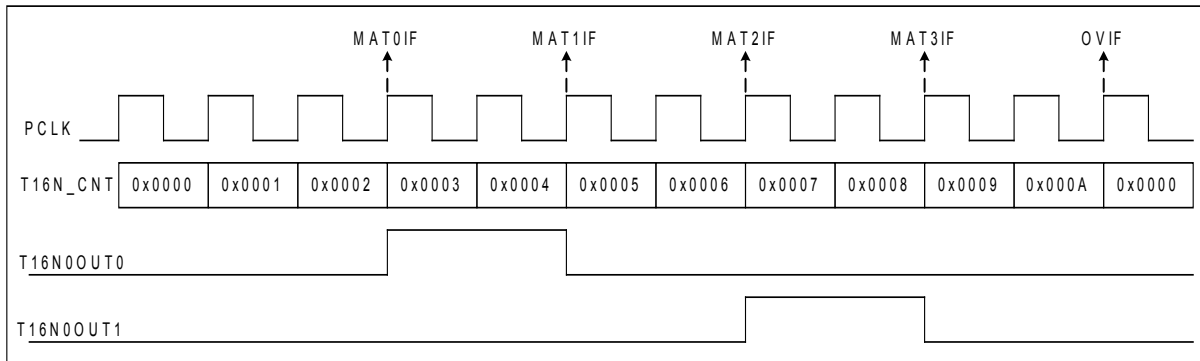


图 5-4 T16N 输出调制功能示意图

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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>	

—	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	T16N_CNT 匹配 T16N_MAT3 后的工作模式选择位 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	T16N_CNT 匹配 T16N_MAT2 后的工作模式选择位 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	T16N_CNT 匹配 T16N_MAT1 后的工作模式选择位 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	T16N_CNT 匹配 T16N_MAT0 后的工作模式选择位 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	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>	bit 5~4	R/W	外部时钟计数边沿选择位 00: 上升沿计数 01: 下降沿计数 10: 上升沿/下降沿均计数 (仅同步计数模式) 11: 上升沿/下降沿均计数 (仅同步计数模式)
SYNC	bit 3	R/W	外部时钟同步使能位 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	T16N 计数时钟源选择位 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	T16N 使能位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
停留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
停留						CAPL1	CAPL0	CAPT<3:0>			CAPIS1	CAPIS0	CAPNE	CAPPE	

—	bit31-10	—	—
CAPL1	bit9	R/W	捕捉 1 重载计数器使能位 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: 18H															
复位值: 00000000_00000000_00000000_00000000 _B															
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
PBKL0	bit20	R/W	PWM 通道 0 刹车输出电平选择位 0: 输出低电平 1: 输出高电平
PBKS<3:0>	bit19-16	R/W	PWM 刹车信号源选择位 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	PWM 刹车事件标志位 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	PWM 刹车信号极性选择位 0: 高电平刹车 1: 低电平刹车
PBKEN	bit12	R/W	PWM 刹车使能位 0: 禁止 1: 使能
MOM3<1:0>	bit11-10	R/W	T16N_MAT3 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	T16N_MAT2 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	T16N_MAT1 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	T16N_MAT0 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	T16N0OUT1 输出极性选择位 0: 正极性 1: 反极性
POL0	bit2	R/W	T16N0OUT0 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

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: 20H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PBKIE	CAP1IE	CAP0IE	OVIE	MAT3IE	MAT2IE	MAT1IE	MAT0IE

—	bit31-8	—	—
PBKIE	bit7	R/W	PWM 刹车中断使能位 0: 禁止 1: 使能
CAP1IE	bit6	R/W	输入端口 T16N0IN1 捕捉中断使能位 0: 禁止 1: 使能
CAP0IE	bit5	R/W	输入端口 T16N0IN0 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PBKIF	CAP1IF	CAP0IF	OVIF	MAT3IF	MAT2IF	MAT1IF	MAT0IF

—	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	T16N_CNT 计数溢出中断标志位 0: 未溢出 1: 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 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											OVT AR	MAT3TAR	MAT2TA R	MAT1TAR	MAT0T AR

—	bit31-5	—	—
OVTAR	bit4	R/W	T16N_CNT 计数溢出触发 ADC 使能位 0: 禁止 1: 使能
MAT3TAR	bit3	R/W	匹配 3 触发 ADC 使能位 0: 禁止 1: 使能
MAT2TAR	bit2	R/W	匹配 2 触发 ADC 使能位 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 _H															
复位值: 00000000_00000000_11111111_11111111 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 _H															
复位值: 00000000_00000000_11111111_11111111 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 _H															
复位值: 00000000_00000000_11111111_11111111 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 _H															
复位值: 00000000_00000000_11111111_11111111 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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_H

复位值: 00000000_00000000_11111111_11111111_B

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

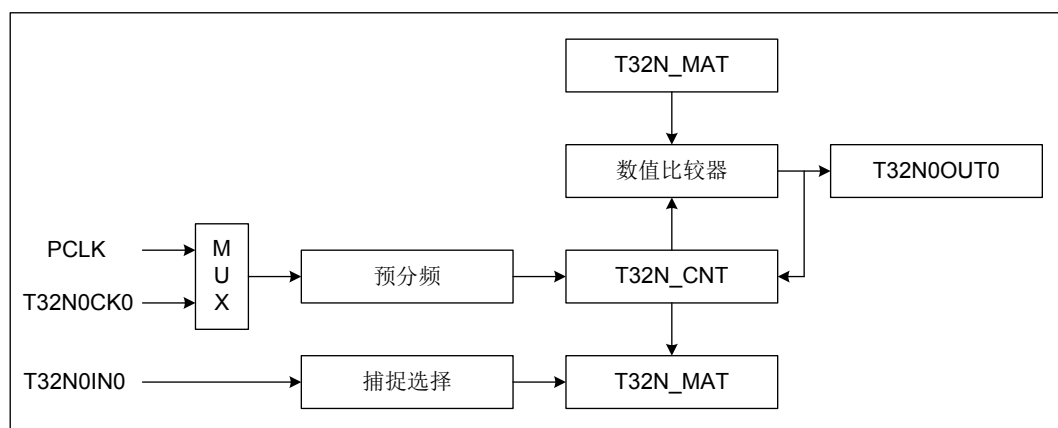


图 5-5 T32N 电路结构框图

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：

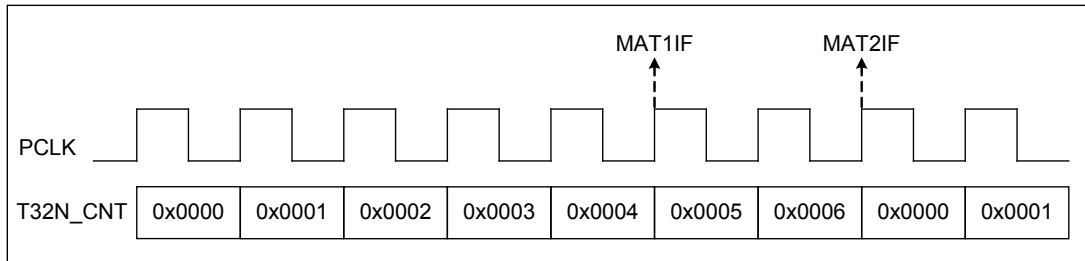


图 5-6 T32N 计数匹配功能示意图

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。

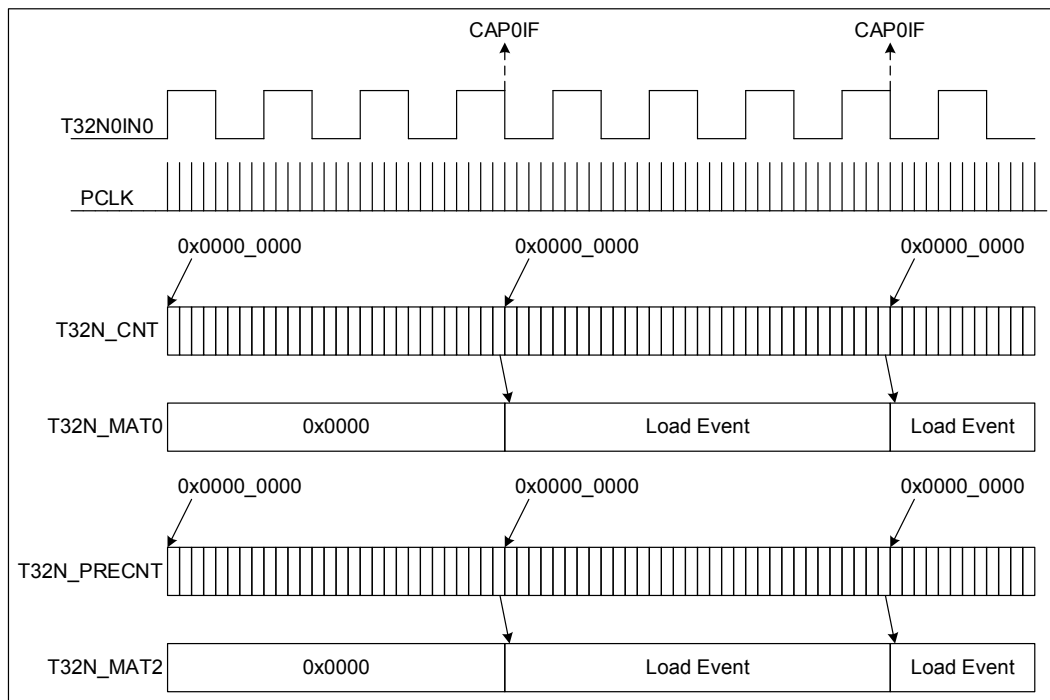


图 5-7 T32N 捕捉功能示意图

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 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 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 (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 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 POLx 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

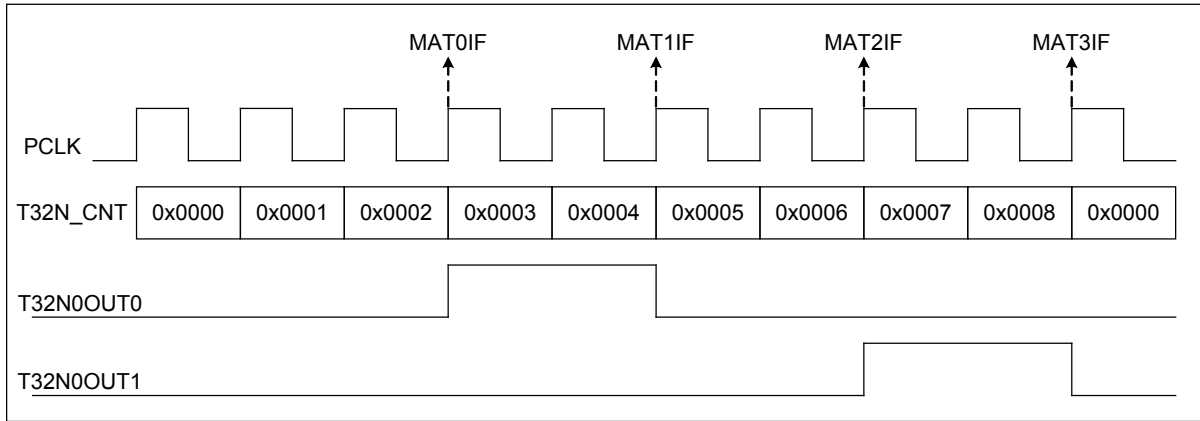


图 5-8 T32N 输出调制功能示意图

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								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 _H																
复位值: 00000000_00000000_00000000_00000000 _B																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留															ASYNCWREN	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MAT3S<1:0>	MAT2S<1:0>	MAT1S<1:0>	MAT0S<1:0>	MOD<1:0>	EDGE<1:0>	SYNC	CS<1:0>									EN

—	bit31-17	—	—
ASYNC_WREN	bit16	R/W	外部时钟异步计数模式下，对计数器的写使能位 0: 禁止写 T32N_CNT 和 T32N_PRECNT，如果强制写，有可能写操作不成功（为避免对计数器的写操作错误，不建议用户对该位写 0） 1: 使能写 T32N_CNT 和 T32N_PRECNT
MAT3S<1:0>	bit15-14	R/W	T32N_CNT 匹配 T32N_MAT3 后的工作模式选择位 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	T32N_CNT 匹配 T32N_MAT2 后的工作模式选择位 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	T32N_CNT 匹配 T32N_MAT1 后的工作模式选择位 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	T32N 匹配 T32N_MAT0 后的工作模式选择位 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	外部时钟计数边沿选择位 00: 上升沿计数 01: 下降沿计数 10: 上升沿/下降沿均计数 (仅同步计数模式) 11: 上升沿/下降沿均计数 (仅同步计数模式)
SYNC	bit3	R/W	外部时钟同步使能位 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	T32N 计数时钟源选择位 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	T32N使能位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CAPL1	CAPL0	CAPT<3:0>				CAPIS1	CAPIS0	CAPNE	CAPPE

—	bit31-10	—	—
CAPL1	bit9	R/W	捕捉 1 重载计数器使能位 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 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				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	T32N_MAT3 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	T32N_MAT2 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	T32N_MAT1 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	T32N_MAT0 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	T32N0OUT1 输出极性选择位 0: 正极性 1: 反极性
POL0	bit2	R/W	T32N0OUT0 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									CAP1IE	CAP0IE	OVIE	MAT3IE	MAT2IE	MAT1IE	MAT0IE

—	bit31-7	—	—
CAP1IE	bit6	R/W	输入端口 1 捕捉中断使能位 0: 禁止 1: 使能
CAP0IE	bit5	R/W	输入端口 0 捕捉中断使能位 0: 禁止 1: 使能
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									CAP1IF	CAP0IF	OVIF	MAT3IF	MAT2IF	MAT1IF	MAT0IF

—	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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											OVT AR	MAT3TA R	MAT2TAR	MAT1TAR	MAT0T AR

—	bit31-5	—	—
OVTAR	bit4	R/W	T32N_CNT 计数溢出触发 ADC 使能位 0: 禁止 1: 使能
MAT3TAR	bit3	R/W	匹配 3 触发 ADC 使能位 0: 禁止 1: 使能
MAT2TAR	bit2	R/W	匹配 2 触发 ADC 使能位 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 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 _H															
复位值: 11111111_11111111_11111111_11111111 _B															
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 _H															
复位值: 11111111_11111111_11111111_11111111 _B															
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 _H															
复位值: 11111111_11111111_11111111_11111111 _B															
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 _H															
复位值: 11111111_11111111_11111111_11111111 _B															
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

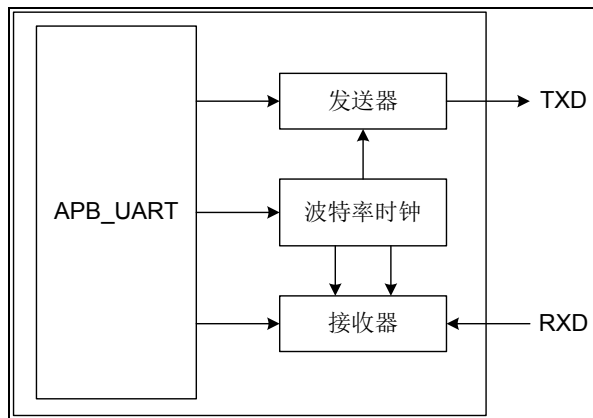


图 5-9 UART 电路结构图

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：

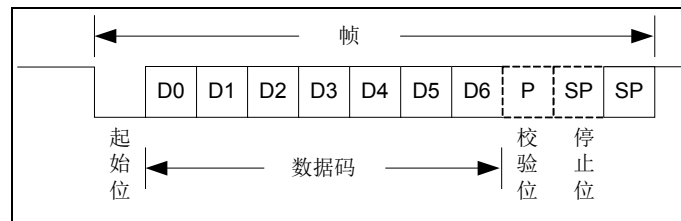


图 5-10 UART 7 位数据格式

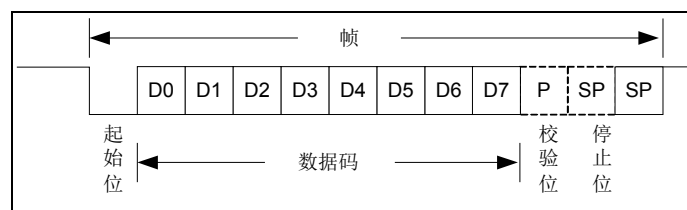


图 5-11 UART 8 位数据格式

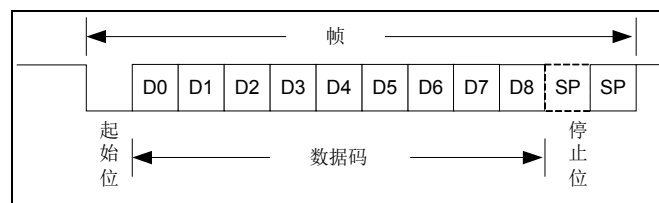


图 5-12 UART 9 位数据格式

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：

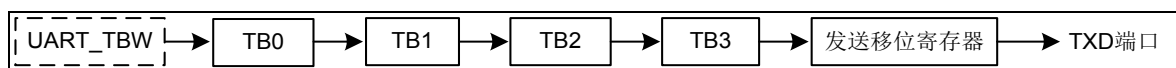


图 5-13 UART 发送数据流示意图

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：

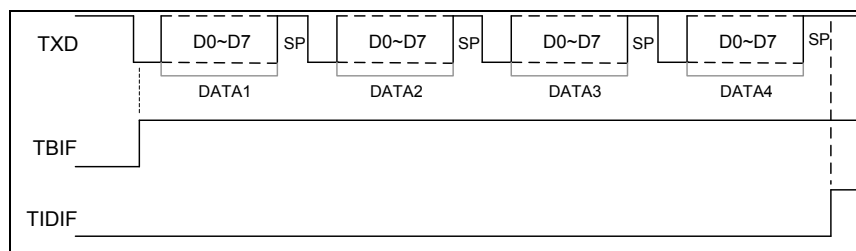


图 5-14 UART 发送中断标志示意图

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：

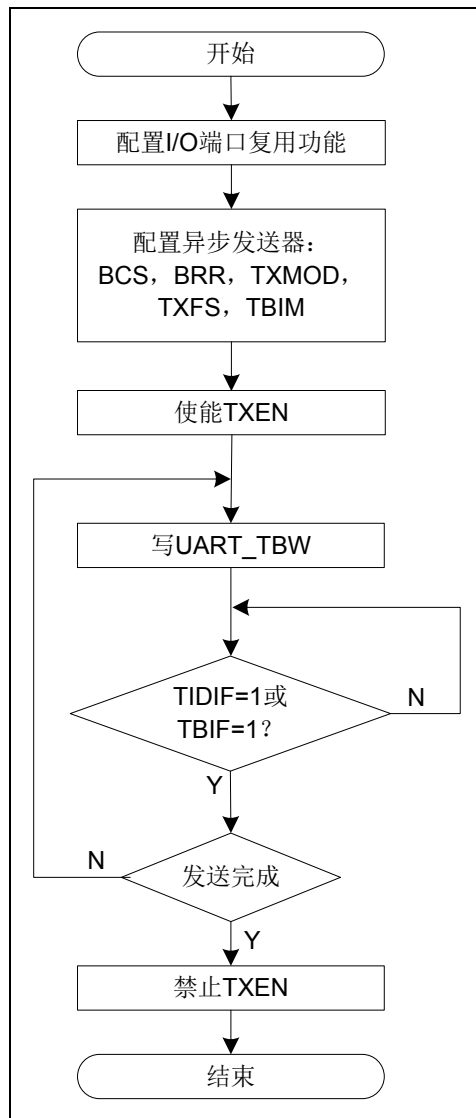


图 5-15 UART 发送数据操作流程图示例

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

FEEx, 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:

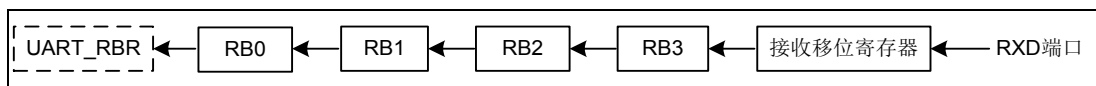


图 5-16 UART 接收数据流示意图

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：

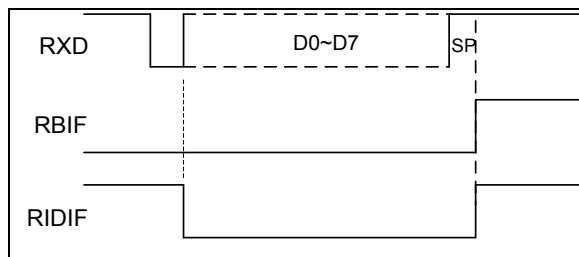


图 5-17 UART 接收中断标志示意图

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：

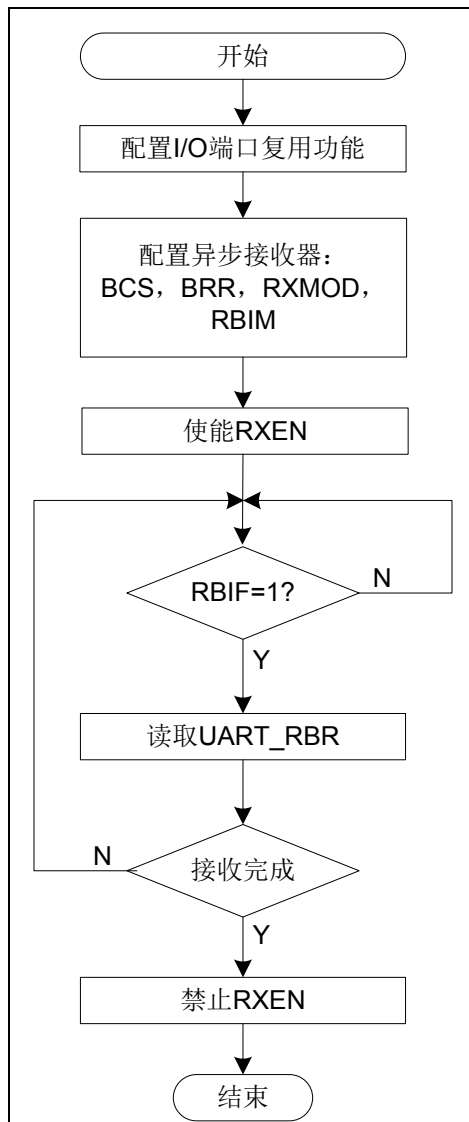


图 5-18 UART 接收数据操作流程图示例

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：

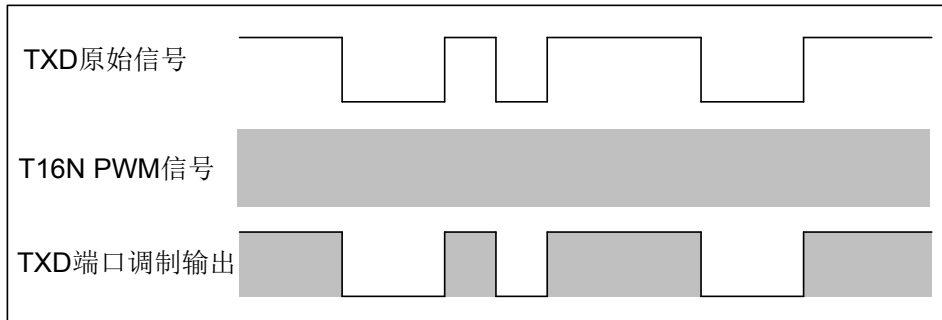


图 5-19 高电平调制输出波形图

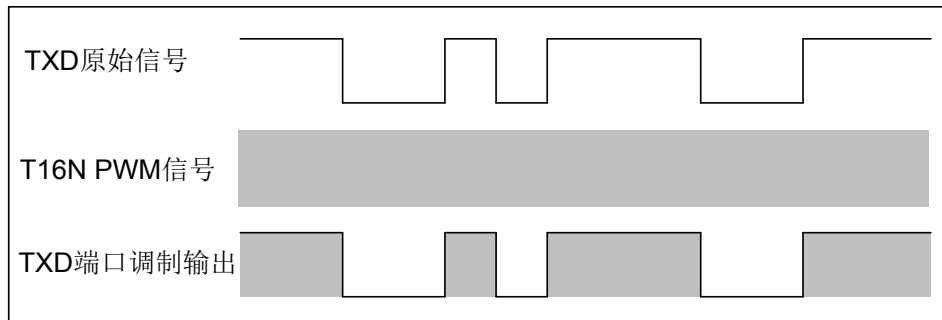


图 5-20 低电平调制输出波形图

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	BCS<2:0>		保留											FERST DIS	SWH F
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXP	RXMOD<2:0>		保留	RRS T	RXE N	TXP	TXMOD<2: 0>	保留	TXFS	TRST	TXEN				

—	bit 31	—	—
BCS<2:0>	bit30-28	R/W	传输波特率发生器时钟选择位 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	接收帧错误后自动复位禁止位 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	单线半双工选择位 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	接收端口极性选择位 0: 正极性 (标准 UART 极性) 1: 负极性 (反向 UART 极性)
RXMOD<2:0>	bit14-12	R/W	接收数据格式选择位 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
—	bit11-10	—	—
RRST	bit9	W	接收器软件复位 0: 读取时始终为 0 1: 软件复位
RXEN	bit8	R/W	接收使能位

			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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		BRFRA<3:0>				BRR<10:0>									

—	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:

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

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

TBW<31:0>	bit31-0	W	<p>写入的发送数据</p> <p>字节写入时: 仅允许对 UART_TBW<7:0>写入</p> <p>半字写入时: 仅允许对 UART_TBW<15:0>写入</p> <p>字写入时: 对 UART_TBW<31:0>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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBR<15:0>															

RBR<31:0>	bit31-0	R	<p>读取的接收数据</p> <p>字节读取时: 仅允许对 UART_RBR<7:0>读取</p> <p>半字读取时: 仅允许对 UART_RBR<15:0>读取</p> <p>字读取时: 对 UART_RBR<31:0>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 _H															
复位值: 00100000_00000000_00100000_00000000 _B															
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	<p>发送缓冲器 1 空标志位</p> <p>0: 满</p> <p>1: 空</p>
TP1	bit28	R	<p>发送的奇偶校验位</p> <p>发送缓冲器 1 对应的奇偶校验位</p>
—	bit27-25	—	—
TB1<8:0>	bit24-16	R	发送缓冲器 1 数据
—	bit15-14	—	—
TBEF0	bit13	R	<p>发送缓冲器 0 空标志位</p> <p>0: 满</p> <p>1: 空</p>
TP0	bit12	R	<p>发送的奇偶校验位</p> <p>发送缓冲器 0 对应的奇偶校验位</p>
—	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 _H															
复位值: 00100000_00000000_00100000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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: 20H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	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 _H															
复位值: 00000000_00000000_XX110000_00000001 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		RIDIF	TIDIF	保留						TBEIF	PEIF	FEIF	ROIF	RBIF	TBIF

—	bit31-14	—	—
RIDIF	bit13	R/W	接收空闲标志中断标志位 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	发送空闲标志中断标志位 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	发送缓冲错误中断标志位 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	接收校验错误中断标志位 0: 接收校验正常 1: 接收校验错误 软件写 1 清除标志位, 写 0 无效
FEIF	bit3	R/W	接收帧错误中断标志位 0: 接收帧正常 1: 接收帧错误 软件写 1 清除标志位, 写 0 无效
ROIF	bit2	R/W	接收数据溢出中断标志位 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
RBIF	bit1	R	接收缓冲器满中断标志位 0: 非满 1: 满 (满足 RBIM 所选择的条件)
TBIF	bit0	R	发送缓冲器空中断标志位

			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

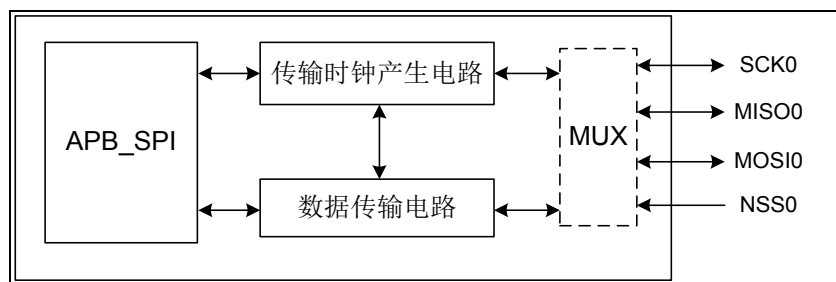


图 5-21 SPI 电路结构框图

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）：

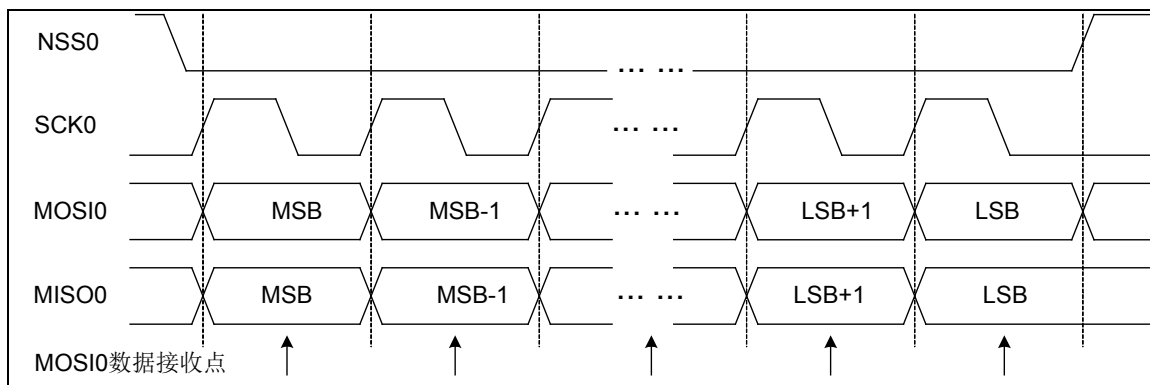


图 5-22 SPI 时钟上升沿发送，下降沿接收波形示意图

DFS<1:0> = 01，下降沿发送（先），上升沿接收（后）：

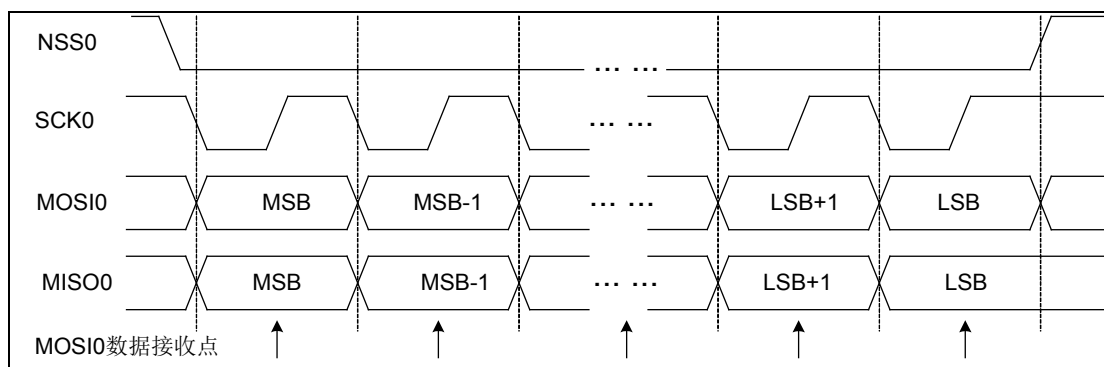


图 5-23 SPI 时钟下降沿发送，上升沿接收波形示意图

DFS<1:0> = 10，上升沿接收（先），下降沿发送（后）：

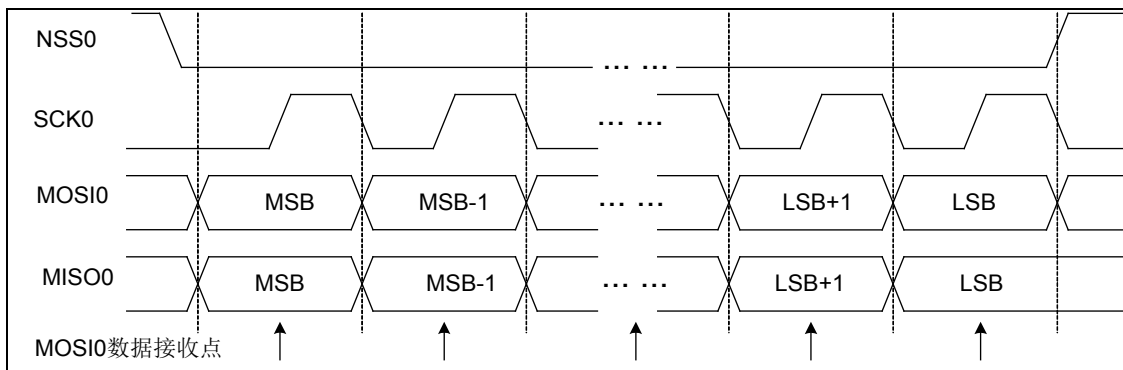


图 5-24 SPI 时钟上升沿接收，下降沿发送波形示意图

DFS<1:0> = 11, 下降沿接收 (先), 上升沿发送 (后):

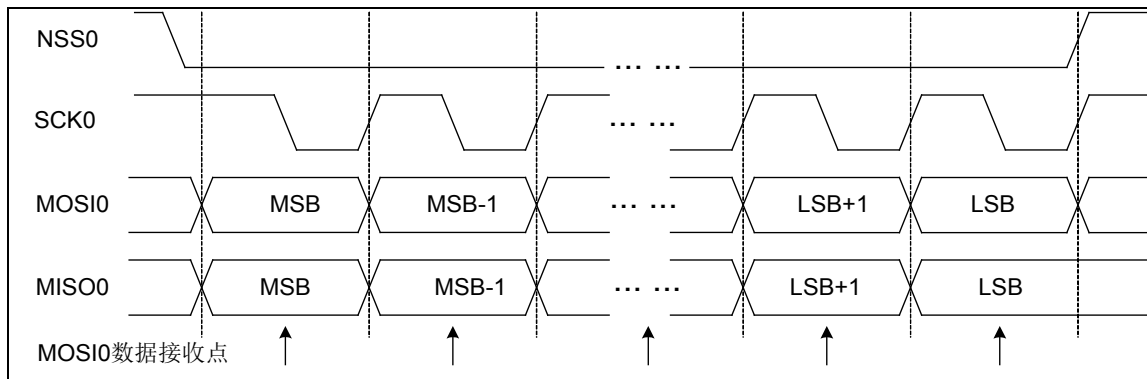


图 5-25 SPI 时钟下降沿接收, 上升沿发送波形示意图

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):

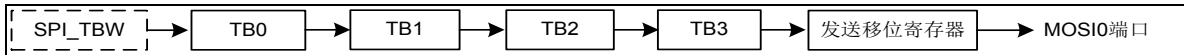


图 5-26 SPI 发送数据流示意图

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：

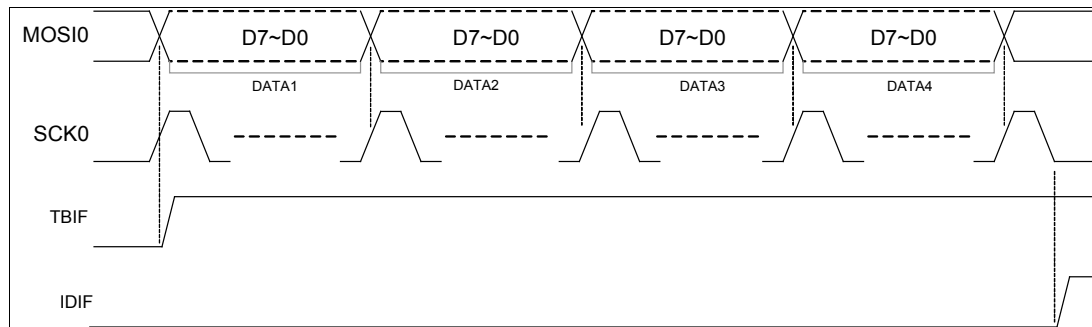


图 5-27 SPI 发送中断标志示意图

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:

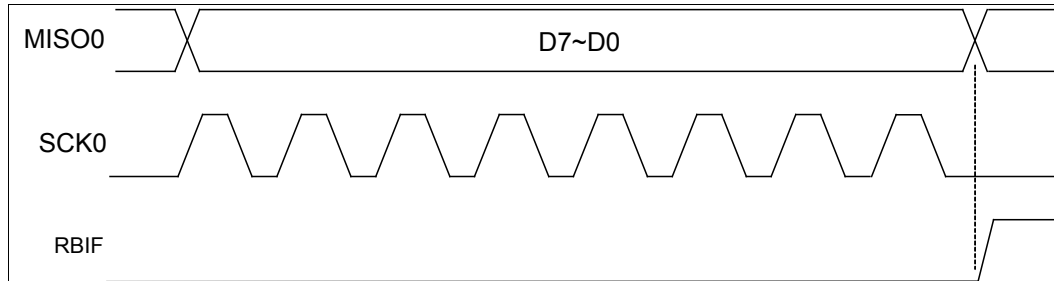


图 5-29 SPI 接收中断标志示意图

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）。

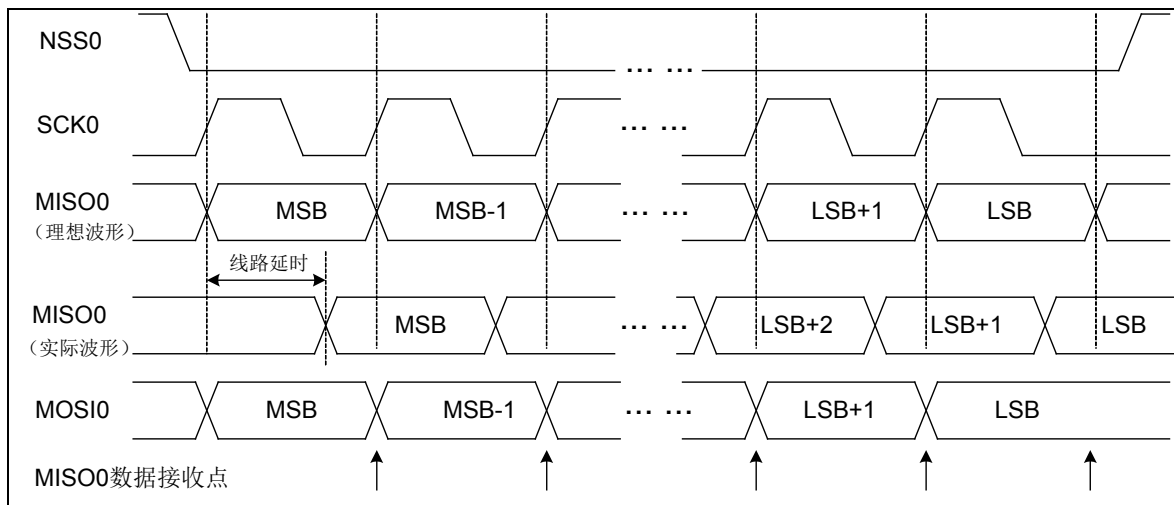


图 5-30 SPI 延迟接收功能波形示意图

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 _H															
复位值: 00000111_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXCLR	TXCLR	保留			DW<2:0>			TMP<5:0>					TMS	TME	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								DFS<1:0>	DRE	保留	REN	MS	RST	EN	

RXCLR	bit31	W	SPI 接收缓冲器清空控制位 0: 无效; 1: 清空接收缓冲器
TXCLR	bit30	W	SPI 发送缓冲器清空控制位 0: 无效; 1: 清空发送缓冲器
—	bit29-27	—	—
DW<2:0>	bit26-24	R/W	SPI 发送帧位宽 (1~8 位) 一帧数据位宽为 SPIDW+1 位
TMP<5:0>	bit23-18	R/W	SPI 帧发送间隔周期设置位 (仅主控模式支持) 详细说明见备注
TMS	bit17	R	SPI 帧发送间隔状态标志位 (仅主控模式支持) 0: 非发送间隔状态 1: 发送间隔状态
TME	bit16	R/W	SPI 帧发送间隔使能位 (仅主控模式支持) 0: 禁止 1: 使能
—	bit15-8	—	—
DFS<1:0>	bit7-6	R/W	SPI 通讯数据格式 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	SPI 延迟接收使能位 (仅主控模式支持) 0: 禁止 1: 使能
—	bit4	—	—
REN	bit3	R/W	SPI 接收使能位 0: 禁止 1: 使能 (需 EN 同时使能)
MS	bit2	R/W	SPI 通讯模式选择位 0: 主控模式

			1: 从动模式
RST	bit1	W	SPI 软件复位 0: 读取时始终为 0 1: 软件复位, 自动清零
EN	bit0	R/W	SPI 通讯使能位 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: 08 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBW<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBW<15:0>															

TBW<31:0>	bit31-0	W	写入的发送数据 字节写入时: 仅允许对 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: 0C _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBR<15:0>															

RBR<31:0>	bit31-0	R	读取的接收数据 字节读取时: 仅允许对 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: 10H															
复位值: 00000000_00000000_00000000_00000000b															
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	SPI 接收缓冲器满中断模式选择位 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	SPI 发送缓冲器空中断模式选择位 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	SPI 发送数据写错误中断使能位 0: 禁止 1: 使能
NSSIE	bit5	R/W	SPI 片选变化中断使能位 (仅从动模式支持) 0: 禁止 1: 使能
IDIE	bit4	R/W	SPI 空闲状态中断使能位 (仅主控模式支持) 0: 禁止 1: 使能
ROIE	bit3	R/W	SPI 接收数据溢出中断使能位 0: 禁止 1: 使能
TEIE	bit2	R/W	SPI 发送数据错误中断使能位 (仅从动模式支持) 0: 禁止 1: 使能
RBIE	bit1	R/W	SPI 接收缓冲器满中断使能位 0: 禁止 1: 使能
TBIE	bit0	R/W	SPI 发送缓冲器空中断使能位 0: 禁止 1: 使能

5.3.11.5 SPI中断标志寄存器 (SPI_IF)

SPI 中断标志寄存器 (SPI_IF)															
偏移地址: 14 _H															
复位值: 00000000_00000000_00000000_00000001 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TBWEIF	NSSIF	IDIF	ROIF	TEIF	RBIF	TBIF

—	bit31-7	—	—
TBWEIF	bit6	R/W	SPI 发送数据写错误中断标志位 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	SPI 片选变化中断标志位 (仅从动模式支持) 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	SPI 空闲中断标志位 (仅主控模式支持) 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	SPI 接收数据溢出中断标志位 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
TEIF	bit2	R/W	SPI 发送错误中断标志位 (仅从动模式支持) 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	SPI 接收缓冲器满中断标志位 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	SPI 发送缓冲器空中断标志位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															

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: 20H															
复位值: 00000000_00000001_00001111_10000000 _B															
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	SPI 空闲标志位 (仅主控模式支持) 0: 非空闲状态 1: 空闲状态
RBFF3	bit15	R	RB3 满标志位 0: 空 1: 满
RBFF2	bit14	R	RB2 满标志位 0: 空 1: 满
RBFF1	bit13	R	RB1 满标志位 0: 空 1: 满
RBFF0	bit12	R	RB0 满标志位 0: 空 1: 满
TBEF3	bit11	R	TB3 空标志位 0: 满 1: 空
TBEF2	bit10	R	TB2 空标志位 0: 满 1: 空
TBEF1	bit9	R	TB1 空标志位 0: 满 1: 空
TBEF0	bit8	R	TB0 空标志位 0: 满 1: 空
NSS	bit7	R	SPI 片选标志位 (仅从动模式支持) 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

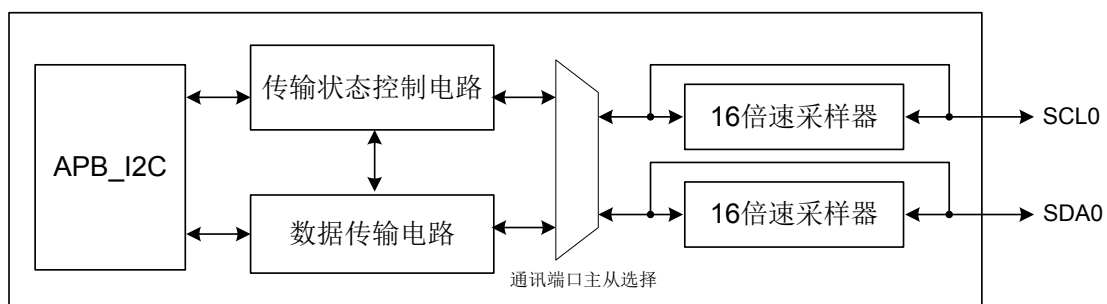


图 5-31 I2C 电路结构框图

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：

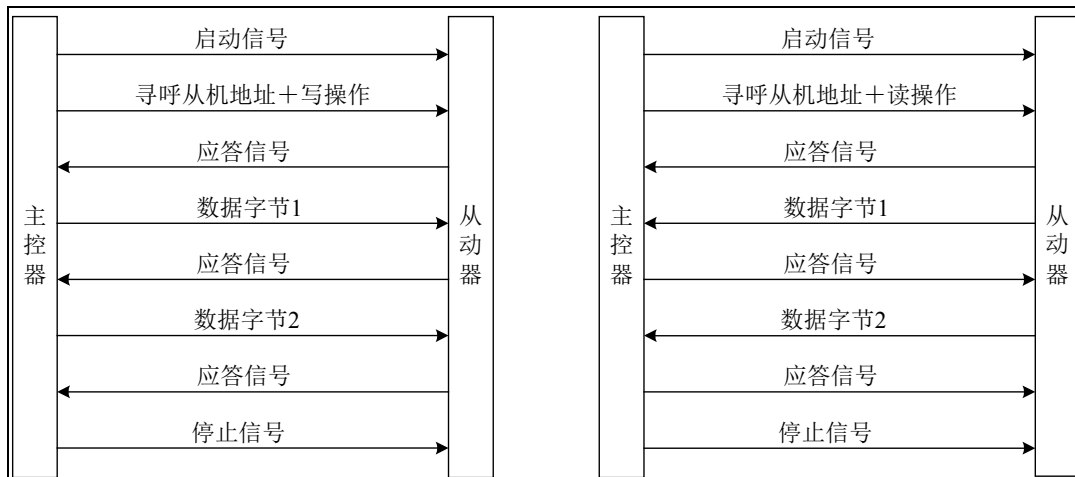


图 5-32 I2C 总线通讯协议示意图

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 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：

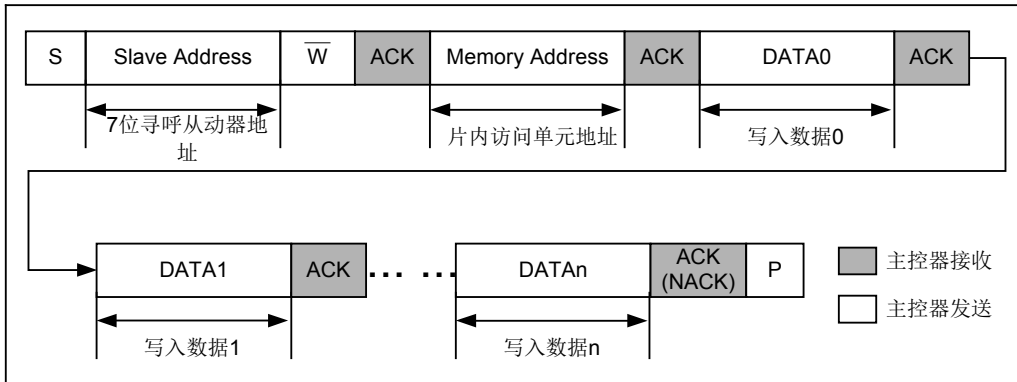


图 5-33 I2C 主控器写入从动器数据示意图

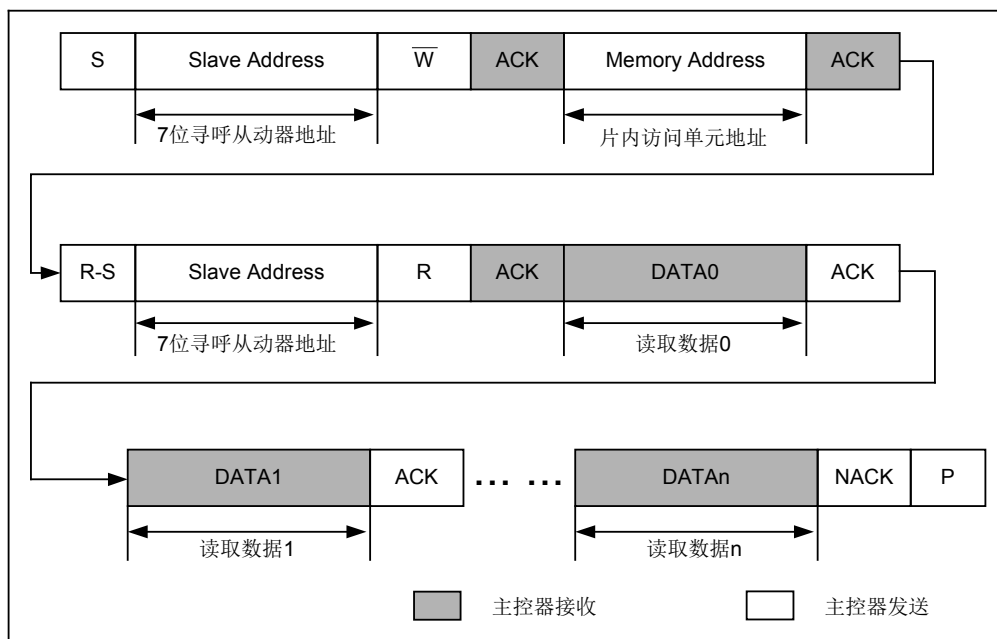


图 5-34 I2C 主控器读取从动器数据示意图

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：

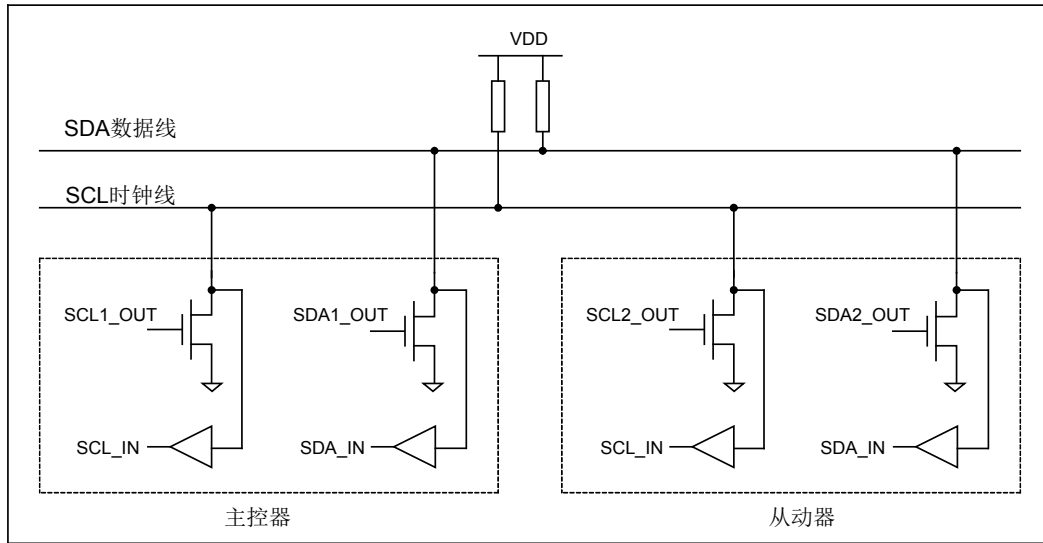


图 5-35 开漏输出端口示意图

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/重启动位建立时间	T _{SU} :S	$> T_{osc} \times (T_{JP} + 1) \times 12$	$T_{osc} \times (T_{JP} + 1) \times 8$
启动/重启动位保持时间	T _{HD} :S	$> T_{osc} \times (T_{JP} + 1) \times 12$	$T_{osc} \times (T_{JP} + 1) \times 8$
停止位建立时间	T _{SU} :P	$> T_{osc} \times (T_{JP} + 1) \times 12$	$T_{osc} \times (T_{JP} + 1) \times 8$
停止位保持时间	T _{HD} :P	$> T_{osc} \times (T_{JP} + 1) \times 12$	$T_{osc} \times (T_{JP} + 1) \times 8$
数据/应答位建立时间	T _{SU} :DA	$> T_{osc} \times (T_{JP} + 1) \times 4$	$T_{osc} \times (T_{JP} + 1) \times 4$
数据/应答位保持时间	T _{HD} :DA	$> T_{osc} \times (T_{JP} + 1) \times 8$	$T_{osc} \times (T_{JP} + 1) \times 4$
通讯时钟高电平脉宽	T _{HIGH}	$T_{osc} \times (T_{JP} + 1) \times 12$	$T_{osc} \times (T_{JP} + 1) \times 8$
通讯时钟低电平脉宽	T _{LOW}	$T_{osc} \times (T_{JP} + 1) \times 12$	$T_{osc} \times (T_{JP} + 1) \times 8$

表 5-1 I2C 总线端口信号的时序参数列表

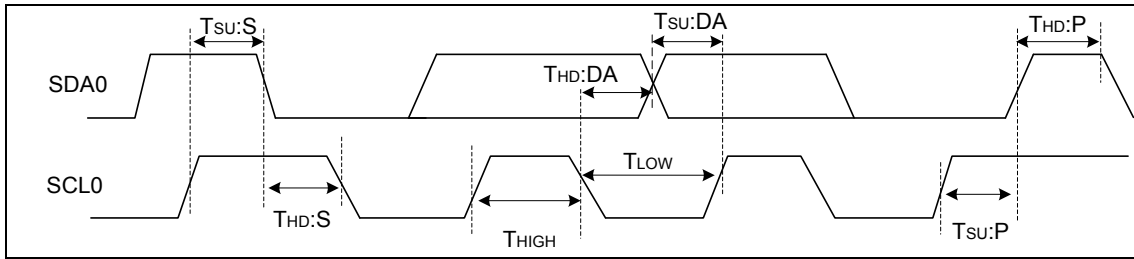


图 5-36 I2C 总线端口信号的波形示意图

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（F_{osc} 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：

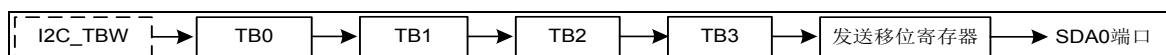


图 5-37 I2C 发送数据流示意图

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：

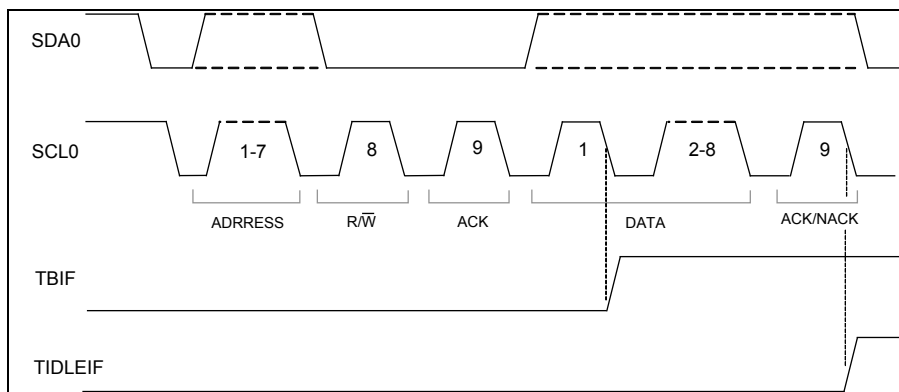


图 5-38 I2C 发送中断标志示意图

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：

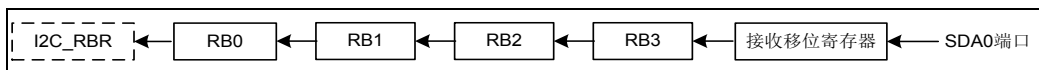


图 5-39 I2C 接收数据流示意图

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，此后仍可以继续接收数据，但接收到的新数据会被丢弃，4 级接收缓冲器中的数据保持不变，读取数据时，只能读到 4 级接收缓冲器中的数据，而接收移位寄存器中的数据会被丢弃。

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 寄存器的中断标志 RBIF；

RBIM<1:0>=10, 为字满产生中断, 即接收缓冲器 RB0, RB1, RB2 和 RB3 均为满时, 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:

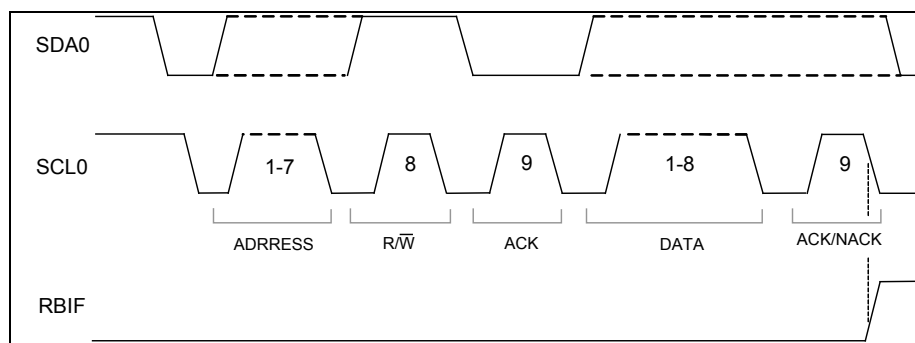


图 5-40 I2C 接收中断标志示意图

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：

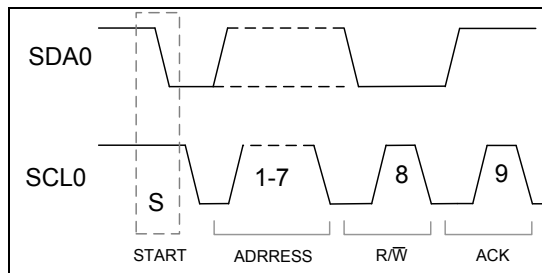


图 5-41 I2C 起始位波形图

I2C 主控模式，支持自动寻呼功能。配置 I2C_MOD 寄存器的 SRAE，可使能自动寻呼功能，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。

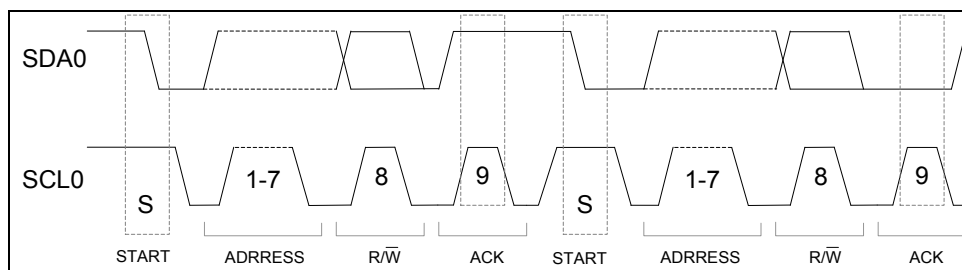


图 5-42 I2C 自动寻呼波形图

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：

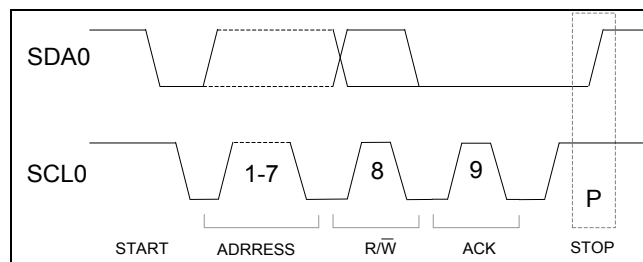


图 5-43 I2C 停止位波形图

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_{SCL0}，通讯波形示意图如下：

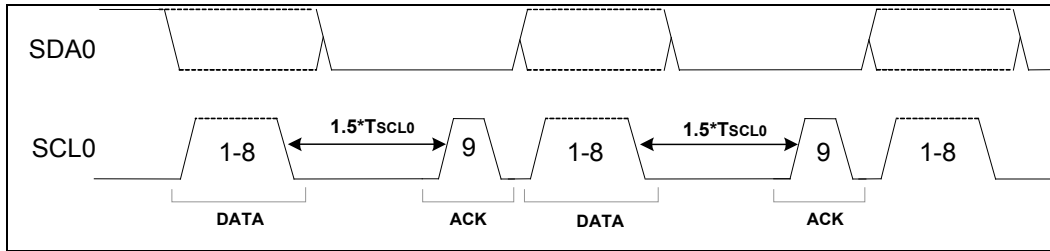


图 5-44 I2C 应答延迟功能波形示意图

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_{SCL0}，通讯波形示意图如下：

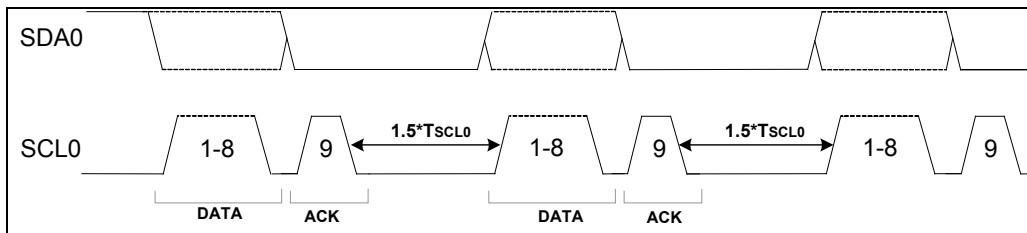


图 5-45 I2C 数据帧传输间隔功能波形示意图

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：

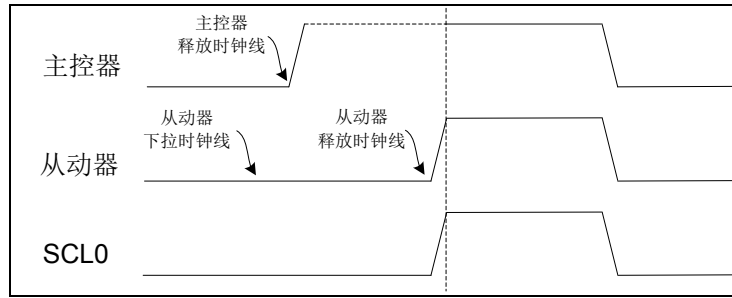


图 5-46 I2C 时钟线下拉等待波形示意图

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 $SDA0$ 选择为开漏输出模式的情况， $SCL0$ 可选择为推挽输出或开漏输出模式。

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 _H																
复位值: 00000000_00000000_11111111_00000000 _B																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留								SA<6:0>								RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TJP<7:0>								TJE	保留	SDASE	SCKSE	SDAOD	SCKOD	RST	EN	

—	bit31-24	—	—
SA<6:0>	bit23-17	R/W	从机地址位 主控模式: 触发“启动/重新启动”操作时, 自动发送 从动模式: 接收到“启动/重新启动”后用于匹配比较
RW	bit16	R/W	I2C 读写控制位 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	I2C 时基定时周期设置位 00~FF: 分别为 1~256 个 TPCLK
TJE	bit7	R/W	I2C 时基定时器使能位 0: 禁止 1: 使能
—	bit6	—	—
SDASE	bit5	R/W	SDA0 端口 16 倍速采样使能位 0: 禁止 1: 使能
SCKSE	bit4	R/W	SCL0 端口 16 倍速采样使能位 0: 禁止 1: 使能
SDAOD	bit3	R/W	SDA0 端口输出模式选择位 0: 推挽输出 1: 开漏输出
SCKOD	bit2	R/W	SCL0 端口输出模式选择位 0: 推挽输出 1: 开漏输出
RST	bit1	W	I2C 软件复位 0: 读取时始终为 0 1: 软件复位, 自动清零

EN	bit0	R/W	I2C 通讯使能位 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 _H			
复位值: 00000000_00000000_00000000_00000000 _B			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留							TAS	保留					BLD	RDT	SPT	SRT
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TIS<3:0>			ADE	ADLY<2:0>			SPAE	SRAE	ANAE	CSE	RDM<2:0>			MS		

—	bit31-25	—	—
TAS	bit24	R/W	I2C 发送应答设置位 (仅从动模式支持) 0: 发送 ACK 1: 发送 NACK
—	bit23-20	—	—
BLD	bit19	R/W	I2C 总线释放功能控制位 (仅主控模式支持) 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	I2C 接收数据触发位 (仅主控模式支持) 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	I2C 停止位触发位 (仅主控模式支持) 0: 无效 1: 触发停止位
SRT	bit16	R/W	I2C 起始位触发位 (仅主控模式支持) 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	I2C 数据帧传输间隔设置位 (仅主控模式支持) 0000: 禁止 0001~1111: 分别为 1~15 个 I2C 通讯时钟周期
ADE	bit11	R/W	I2C 应答延迟使能位 (仅主控模式支持) 0: 禁止 1: 使能
ADLY<2:0>	bit10-8	R/W	I2C 应答延迟时间设置位 (仅主控模式支持) 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	I2C 自动结束使能位 (仅主控模式支持) 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	I2C 自动寻呼使能位 (仅主控模式支持) 0: 禁止 1: 使能 (若寻呼地址的应答位为 NACK, 则自动重启本次寻呼操作)
ANAE	bit5	R/W	I2C 自动发送未应答使能位 (仅从动模式支持) 0: 禁止 1: 使能
CSE	bit4	R/W	I2C 时钟线自动下拉等待请求使能位 (仅从动模式支持) 0: 禁止 1: 使能
RDM<2:0>	bit3-1	R/W	I2C 接收模式选择位 (仅主控模式支持) 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	I2C 通讯模式选择位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		TIDLEIE	RBIM<1:0>	TBIM<1:0>	TBWEIE	NAIE	ROIE	TEIE	RBIE	TBIE	SPIE	SRIE			

—	bit31-13	—	—
TIDLEIE	bit12	R/W	I2C 发送空闲中断使能位 0: 禁止 1: 使能
RBIM<1:0>	bit11-10	R/W	I2C 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 10: 字满产生中断 11: 保留
TBIM<1:0>	bit9-8	R/W	I2C 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 10: 字空产生中断 11: 保留
TBWEIE	bit7	R/W	I2C 发送数据写错误中断使能位 0: 禁止 1: 使能
NAIE	bit6	R/W	I2C 未应答 NACK 中断使能位 0: 禁止 1: 使能
ROIE	bit5	R/W	I2C 接收数据溢出中断使能位 0: 禁止 1: 使能
TEIE	bit4	R/W	I2C 发送数据错误中断使能位 0: 禁止 1: 使能
RBIE	bit3	R/W	I2C 接收缓冲器满中断使能位 0: 禁止 1: 使能
TBIE	bit2	R/W	I2C 发送缓冲器空中断使能位 0: 禁止 1: 使能
SPIE	bit1	R/W	I2C 停止位中断使能位 0: 禁止

			1: 使能
SRIE	bit0	R/W	I2C 起始位中断使能位 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 _H															
复位值: 00000000_00000000_00000000_00000100 _b															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIDLEIF	保留				TBWEIF	NAIF	ROIF	TEIF	RBIF	TBIF	SPIF	SRIF		

—	bit31-13	—	—
TIDLEIF	bit12	R/W	I2C 发送空闲中断标志位 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	I2C 发送数据写错误中断标志位 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	I2C 未应答中断标志位 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	I2C 接收数据溢出中断标志位 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
TEIF	bit4	R/W	I2C 发送错误中断标志位 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	I2C 接收缓冲器满中断标志位 0: 非满 1: 满（满足 RBIM 所选择的条件）
TBIF	bit2	R	I2C 发送缓冲器空中断标志位 0: 非空 1: 空（满足 TBIM 所选择的条件）
SPIF	bit1	R/W	I2C 停止位中断标志位 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	I2C 起始位中断标志位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBW<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBW<15:0>															

TBW<31:0>	bit31-0	W	写入的发送数据 字节写入时: 仅允许对 I2C_TBW<7:0>写入 半字写入时: 仅允许对 I2C_TBW<15:0>写入 字写入时: 对 I2C_TBW<31:0>写入
-----------	---------	---	--

5.4.9.6 I2C接收数据读取寄存器 (I2C_RBR)

I2C 接收数据读取寄存器 (I2C_RBR)															
偏移地址: 14 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBR<15:0>															

RBR<31:0>	bit31-0	R	读取的接收数据 字节读取时: 仅允许对 I2C_RBR<7:0>读取 半字读取时: 仅允许对 I2C_RBR<15:0>读取 字读取时: 对 I2C_RBR<31:0>读取
-----------	---------	---	--

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB3<7:0>								RB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1<7:0>								RB0<7:0>							
RB3<7:0>	bit31-24			R	接收数据缓冲器 3										
RB2<7:0>	bit23-16			R	接收数据缓冲器 2										
RB1<7:0>	bit15-8			R	接收数据缓冲器 1										
RB0<7:0>	bit7-0			R	接收数据缓冲器 0										

5.4.9.9 I2C状态寄存器 (I2C_STA)

I2C 状态寄存器 (I2C_STA)															
偏移地址: 20 _H															
复位值: 00000000_00000010_00001111_00000000 _B															
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	I2C 空闲标志位 0: 非空闲状态 1: 空闲状态
ACK	bit16	R	I2C 应答位 0: 应答 ACK 1: 未应答 NACK
RBFF3	bit15	R	RB3 满标志位 0: 空 1: 满
RBFF2	bit14	R	RB2 满标志位 0: 空 1: 满
RBFF1	bit13	R	RB1 满标志位 0: 空 1: 满
RBFF0	bit12	R	RB0 满标志位 0: 空 1: 满
TBEF3	bit11	R	TB3 空标志位 0: 满 1: 空
TBEF2	bit10	R	TB2 空标志位 0: 满 1: 空
TBEF1	bit9	R	TB1 空标志位 0: 满 1: 空
TBEF0	bit8	R	TB0 空标志位 0: 满 1: 空
—	bit7-0	—	—

5.4.10 I2C应用说明

芯片支持 1 个 I2C 总线串口通讯控制器 I2C0。

I2C 总线在连续发送数据的应用中，利用发送空闲标志（TIDLEIF）进入中断的方式进行数据发送时，要注意以下 2 点：

1. 主控模式下，在主程序中设置 Memory Address 之后，发送空闲标志（TIDLEIF）置 1 并触发中断，在中断服务程序中，将数据写入 I2C_TBW 中，并开始发送数据；
2. 从动模式下，必须在主程序中等待 I2C_STA 寄存器的 IDLE 置 1 之后，将以 Memory Address 为起始地址的第一个数据写入 I2C_TBW 中。当主机发送读命令后，发送空闲标志（TIDLEIF）置 1 并触发中断，在中断服务程序中发送数据。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 18 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

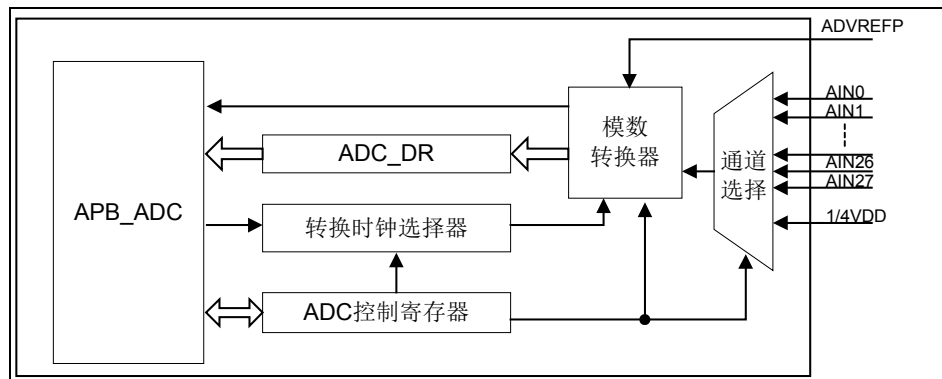


图 5-47 ADC 内部结构图

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，并等待至少 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

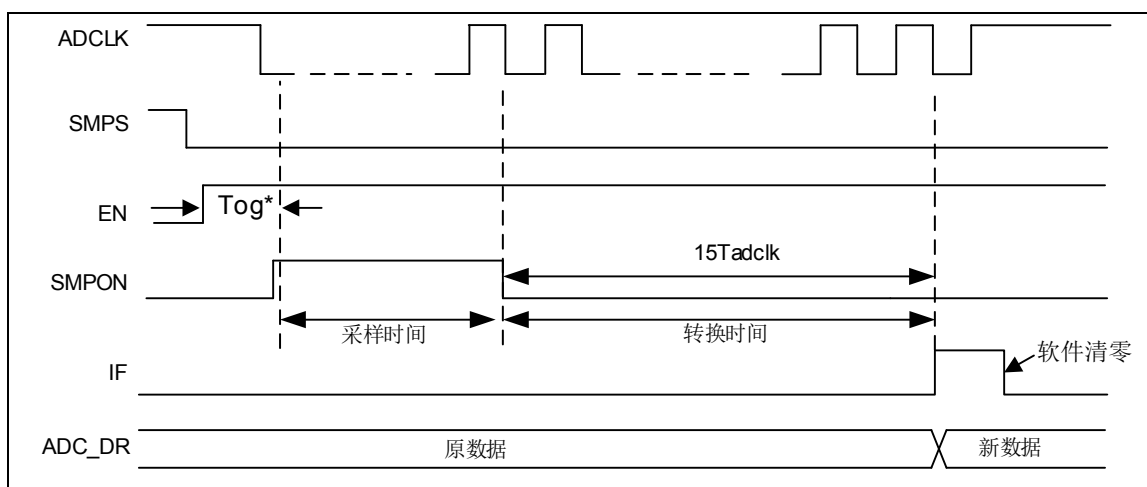


图 5-48 ADC 数据转换时序示意图 (ADC_CON1 寄存器的 SMPS=0, 软件控制采样)

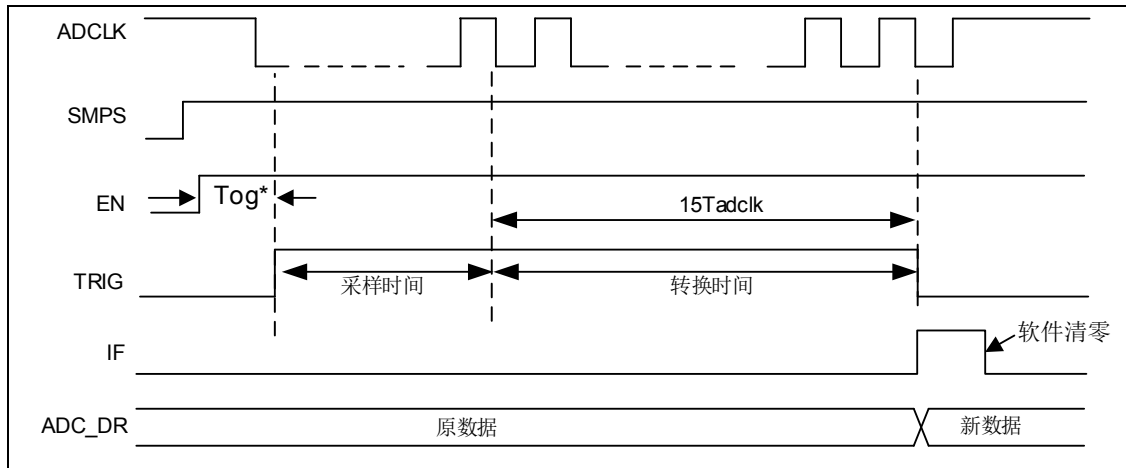


图 5-49 ADC 数据转换时序示意图 (ADC_CON1 寄存器的 SMPS=1, 硬件控制采样)

- 注 1: Tog > 100us;
- 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    ;延时约 300us
LDR    R1, =0X0D          ;使能 CHOP_EN
STR    R1, [R0]
.....
LDR    R0, =ADC_CON0      ;延时约 1ms
LDR    R1, =0X33          ;启动 A/D 转换
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满足，则溢出中断标志 ACPOVIF 无法被软件清除，当前 ADC 自动转换仍继续进行，直到完成所有 ADC 转换次数后，溢出中断标志才可被软件清除，才自动启动下一个 ADC 自动转换过程和溢出时间计算。

配置 ADC_ACPC 寄存器的 CLKS 可选择溢出计数的时钟源，为 PCLK 或者 LRC(32KHz) 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: 40_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-4	—	—
CHOP_EN	bit3	R/W	VREF CHOP 使能控制位 0: 禁止 1: 使能
IREF_EN	bit2	R/W	IREF 使能控制位 0: 禁止 1: 使能 (ADC 工作时需使能 IREF)
VREF_SEL	bit1	R/W	内部 VREFP 电压选择控制位 0: 2.048V 1: 保留, 仅作测试使用 (禁止设置为 1)
VREF_EN	bit0	R/W	内部 VREFP 使能控制位 0: 禁止 1: 使能

注 1: ADC 正常工作时, 无论正向参考电压选择为 VDD, 内部 VREFP 或外部 AVREFP, 均必须开启 IREF_EN, 否则会导致 ADC 工作异常。

注 2: 选择内部参考电压 VREF 2.048V 作为 ADC 正向参考电压时, 需先设置 VREF_EN 位使能内部参考, 并设置 IREF_EN 和 ADC_CON0 寄存器的 EN 位使能 ADC, 然后等待至少 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: 00_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-12	—	—
DR<11:0>	bit11-0	R	A/D 转换结果 (对不同分辨率均高位对齐)

5.5.7.3 ADC控制寄存器 0 (ADC_CON0)

ADC 控制寄存器 0 (ADC_CON0)

偏移地址: 04_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-6	—	—
BIT_SEL<1:0>	bit5-4	R/W	A/D 分辨率选择位 00: 6 位分辨率, 结果存放在 ADC_DR<11:6> 01: 8 位分辨率, 结果存放在 ADC_DR<11:4> 10: 10 位分辨率, 结果存放在 ADC_DR<11:2> 11: 12 位分辨率, 结果存放在 ADC_DR<11:0>
—	bit3	—	—
ACP_EN	bit2	R/W	A/D 自动转换比较功能使能位 0: 关闭 1: 启动
TRIG	bit1	R/W	A/D 转换状态位 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	A/D 转换使能位 (ACP_EN 为 1 时, 该位无效) 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_H

复位值: 00000000_00000100_00010000_00000000_B

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	ADC 调零使能位 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	A/D 采样时间选择位 (硬件控制有效) 采样时间: ST*2+1 个 Tadc
—	bit15	—	—
HSEN	bit14	R/W	AD 转换速度控制位 0: 高速 1: 保留, 仅作测试使用 (禁止设置为 1)
SMPON	bit13	R/W	A/D 采样软件控制位 (ACP_EN 为 1 时, 该位无效) 0: 关闭 AD 采样 1: 启动 AD 采样
SMPS	bit12	R/W	A/D 采样模式选择位 (ACP_EN 为 1 时, 固定为 1) 0: 软件控制 1: 硬件控制
—	bit11	—	—
VREFN	bit10	R/W	A/D 负向参考电压选择位 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	A/D 正向参考电压选择位 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	A/D 时钟源选择位 0: PCLK 1: LRC (32KHz)
CLKDIV<2:0>	bit2-0	R/W	A/D 时钟源预分频选择位

			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。

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_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-5	—	—
CHS<4:0>	bit4-0	R/W	A/D 模拟通道选择位 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_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-4	—	—
ACPOVIE	bit3	R/W	ADC 自动转换溢出中断使能位 0: 禁止 1: 使能
ACPMAXIE	bit2	R/W	ADC 自动转换高阈值超出中断使能位 0: 禁止 1: 使能
ACPMINIE	bit1	R/W	ADC 自动转换低阈值超出中断使能位 0: 禁止 1: 使能
IE	bit0	R/W	ADC 中断使能位 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_H

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-4	—	—
ACPOVIF	bit3	R/W	ADC 自动转换溢出中断标志位 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	ADC 自动转换高阈值超出中断标志位 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	ADC 自动转换低阈值超出中断标志位 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	ADC 中断标志位 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_H

复位值: 00000000_00000000_00000000_00000000_B

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	ACP 溢出计数的时钟源选择位 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 OVFL_TIME 设定的时间段内) 00: 1 次 01: 2 次 10: 4 次 11: 8 次
—	bit15-12	—	—
OVFL_TIME<11:0>	bit11-0	R/W	每次自动转换比较的溢出时间, 可配置范围为 0~9C3_H , 计数时钟周期为 T_{acp} , 分别对应如下溢出时间: 0: 1 x T _{acp} 1: 2 x T _{acp} 2: 3 x T _{acp} ... 9C3 _H : 2500 x T _{acp} T _{acp} 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_H

复位值: 00001111_11111111_00000000_00000000_B

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_H

复位值: 00000000_00000000_00000000_00000000_B

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

复位值: 00000000_00000000_00000000_00000000_B

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

—	bit31-12	—	—
OFFDATA<11:0>	bit 11-0	W/R	转换结果偏置值

注: 本芯片不支持使用 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 32KHz 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 32KHz 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 32KHz LRC 时钟），IWDT 载入 IWDT_LOAD 值（默认为 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。

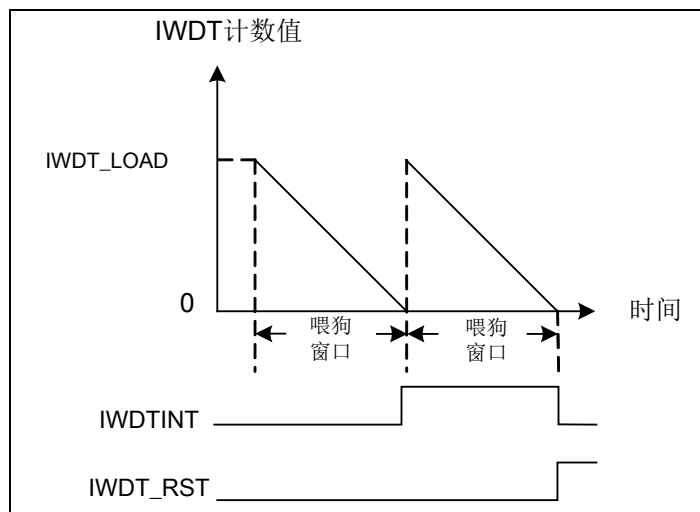


图 5-50 看门狗时序图

写 IWDW_INTCLR，则 IWDW 重新载入计数初值，并重新进行递减计数。

IWDW 模块支持寄存器写保护，配置寄存器 IWDW_LOCK=0x1ACCE551，可去除 IWDW 寄存器的写保护状态，此后可进行被保护的 IWDW 寄存器的写操作，否则无法写 IWDW 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 _H															
复位值: 00000000_00000000_01000000_00000000 _B															
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 _H															
复位值: 11111111_11111111_11111111_11111111 _B															
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	IWDT 计数器当前值 读取时返回 IWDT 计数器的当前计数值
-------------	---------	---	--

5.6.2.3 IWDTC控制寄存器 (IWDTC_CON)

IWDTC 控制寄存器 (IWDTC_CON)															
偏移地址: 08H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CLKS	RSTEN	IE	EN

—	bit31-4	—	—
CLKS	bit3	R/W	IWDTC 计数时钟选择位 0: PCLK 1: LRC 时钟 (约 32KHz)
RSTEN	bit2	R/W	IWDTC 复位使能位 0: 禁止 1: 使能, IWDTC 计数到 0 时, 产生复位信号, 将芯片复位
IE	bit1	R/W	IWDTC 中断使能位 0: 禁止 1: 使能, IWDTC 计数到 0 时, 产生中断标志
EN	bit0	R/W	IWDTC 模块使能位 0: 禁止 1: 使能

注 1: IWDTC_CON 寄存器中的各个控制位, 仅在配置字 CFG_WORD 的配置位 CFG_IWDTCEN=0 时才有效。

注 2: 如果 IWDTC 使用 LRC 时钟计数, 则程序不能在 3 个 LRC 时钟周期内连续两次或多次喂狗, 否则可能会导致后续无法正常喂狗, 误产生 IWDTC 计数溢出复位, 所以用户程序中的喂狗时间间隔应大于 3 个 LRC 时钟周期, 或者确保两次喂狗的时间间隔小于 IWDTC 计数溢出时间, 也不会误产生 IWDTC 计数溢出复位; 如果 IWDTC 使用 PCLK 时钟计数, 则无需考虑两次喂狗的最小时间间隔。

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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	IWDT 中断标志清 0 位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															IWDTIF

—	bit31-1	—	—
IWDTIF	bit0	R	IWDT 中断标志位 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: 100H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															LOCK

—	bit31-1	W	对 IWDT_LOCK<31:0>写 0x1ACCE551 时, 位 LOCK 为 0; 写其它值时位 LOCK 为 1
LOCK	bit0	R	IWDT 寄存器保护状态位 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。

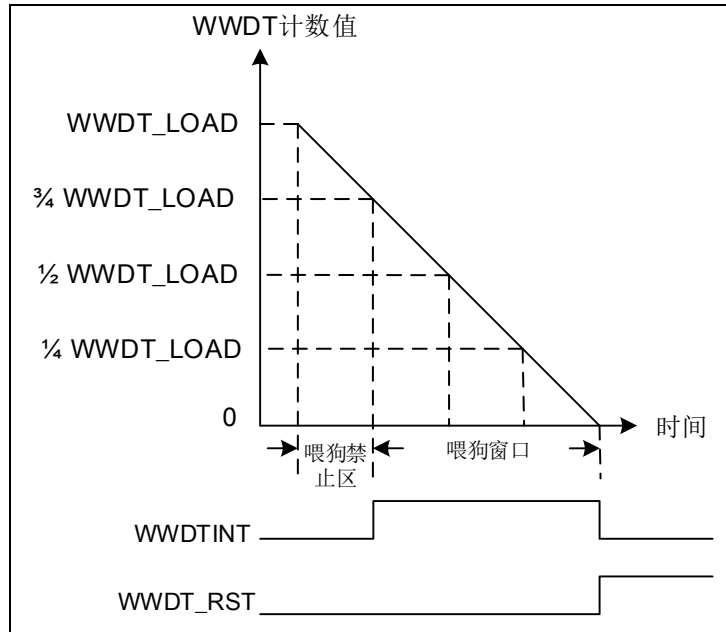


图 5-51 窗口看门狗中断和溢出复位产生时序图 (WWDTWIN 设定为 25%)

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。

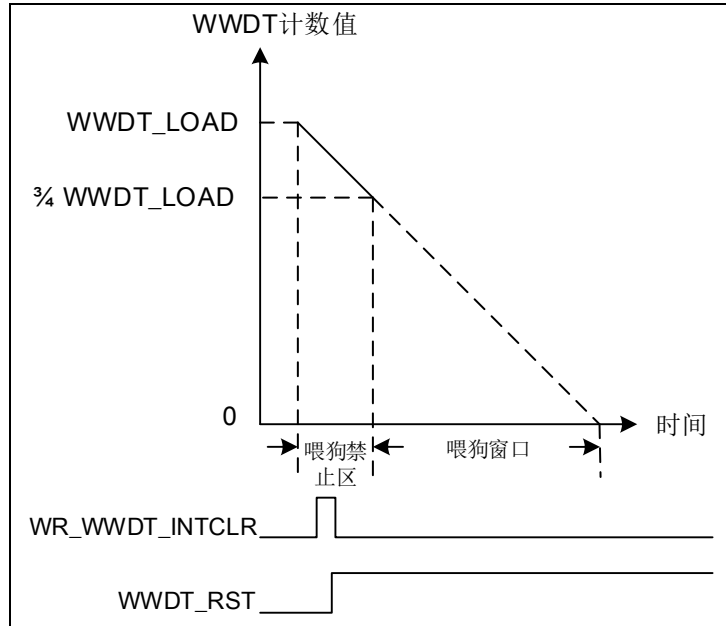


图 5-52 错误的喂狗时序图 (WWDTWIN 设定为 25%)

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 _H															
复位值: 00000000_00000010_00000000_00000000 _B															
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	WWDT 计数器重载值 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 _H															
复位值: 00111111_11111111_11111111_11111111 _B															
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	WWDT 计数器当前值 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										WWDTWIN<1:0>	CLKS	RSTEN	IE	EN	

—	bit31-6	—	—
WWDTWIN<1:0>	bit5-4	R/W	WWDT 禁止喂狗窗口选择位 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	WWDT 计数时钟选择位 0: PCLK 1: LRC 时钟 (约 32KHz)
RSTEN	bit2	R/W	WWDT 复位使能位 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	WWDT 中断使能位 0: 禁止 1: 使能, WWDT 计数到 0 时, 产生中断标志
EN	bit0	R/W	WWDT 模块使能位 0: 禁止 1: 使能

注: 如果 WWDT 使用 LRC 时钟计数, 则程序不能在 3 个 LRC 时钟周期内连续两次或多次喂狗, 否则可能会导致后续无法正常喂狗, 推荐在 WWDT 中断服务程序中喂狗。

5.7.2.4 WWDT中断标志清除寄存器 (WWDT_INTCLR)

WWDT 中断标志清除寄存器 (WWDT_INTCLR)															
偏移地址: 0C _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	WWDT 中断标志清 0 位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															WWDTIF

—	bit31-1	—	—
WWDTIF	bit0	R	WWDT 中断标志位 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: 100H															
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															LOCK

—	bit31-1	W	对 WWDT_LOCK<31:0>写 0x1ACCE551 时, 位 LOCK 为 0; 写其它值时位 LOCK 为 1
LOCK	bit0	R	WWDT 寄存器保护状态位 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
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	CRC 解锁: 写入 0x4352_4355, CRC 解锁; CRC 上锁: 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 _H															
复位值: 00000000_00000000_00000000_00000000 _B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	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	数据输出取反选择位 0: 不取反 1: 取反 (对数据按位取反)
REFOUT	bit13	R/W	数据输出顺序选择位 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	数据输入顺序选择位 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	CRC 位宽选择位 00: 字节

			01: 半字 1x: 字
MOD<1:0>	bit9-8	R/W	CRC 模式选择位 0x: CRC32 10: CRC16 11: CRC16-CCITT
—	bit7-6	—	—
ACK	bit5	R	CRC 访问 FLASH 应答位 0: 不允许操作 Flash 1: 允许操作 Flash
REQ	bit4	R/W	CRC 访问 FLASH 请求位 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	CRC 初始数据格式选择 0: 初始为全 0 1: 初始为全 1
HS	bit2	R/W	CRC 高速模式使能位 0: 禁止 1: 使能
RST	bit1	W	CRC 复位 0: 无任何操作 1: 复位
EN	bit0	R/W	CRC 使能位 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。

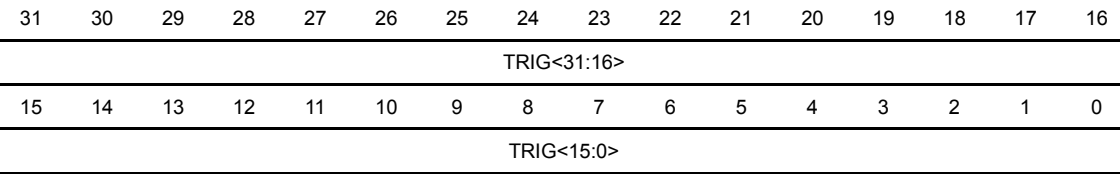
注 2: 在对 FLASH 进行 CRC 访问操作前, 必须先清除 IAP_FLASHCR 寄存器的 FLASH 操作请求使能位 FLASH_REQ=0, 然后置位 CRC 访问 FLASH 请求位 REQ=1, 查询 CRC 访问 FLASH 应答位 ACK=1 后, 再开始 CRC 操作。

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_H

复位值: 00000000_00000000_00000000_00000000_B



TRIG<31:0>	bit31-0	R/W	CRC 工作模式选择 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: 0C_H

复位值: 00000000_00000000_00000000_00000000_B



—	bit31-21	—	—
IFREN	bit20	R/W	Flash 区域选择位 0: 选择 Code 区 1: 选择 Info 区
—	bit19-17	—	—
ADDR<16:0>	bit16-0	R/W	CRC 起始地址 (字节地址) 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: 10_H

复位值: 00000000_00000000_00000000_00000111_B

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

—	bit31-15	—	—
CRC_SIZE<14:0>	bit14-0	R/W	CRC 数据块长度 (以字为单位) 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_H

复位值: 00000000_00000000_00000000_00000000_B

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	CRC 用户数据 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_H

复位值: 00000000_00000000_00000000_00000000_B

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>															

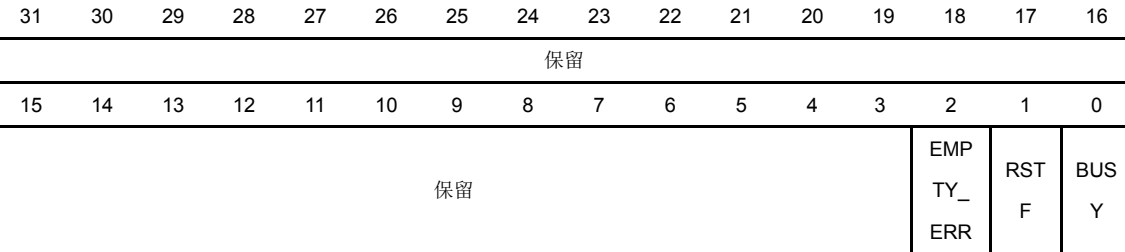
DO<31:0>	bit31-0	R	CRC 校验结果
----------	---------	---	-----------------

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_H

复位值: 00000000_00000000_00000000_00000010_B



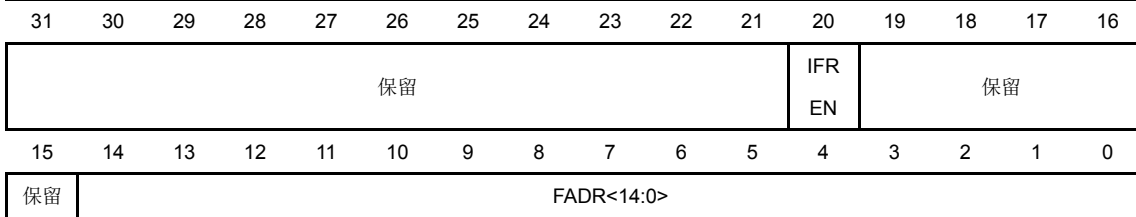
—	bit31-3	—	—
EMPTY_ERR	bit2	R/W	查空错误标志位 0: 无错误 1: 错误 (写 1 或再次触发 Flash 校验/查空清除)
RSTF	bit1	R/W	CRC 复位标志位 0: 未发生复位 1: 发生复位 (写 1 清除)
BUSY	bit0	R	CRC 工作状态位 0: 已完成 CRC 计算 1: 正在进行 CRC 计算

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_H

复位值: 00000000_00000000_00000000_00000000_B



—	bit31-21	—	—
IFREN	bit20	R	CRC 对应的当前 Flash 区域 0: Code 区 1: Info 区
—	bit19-15	—	—
FADR<16:0>	bit14-0	R	CRC 对应的当前 Flash 地址 (字节对齐)

第 6 章 RF收发器—芯片状态控制

RF 收发器各工作模式之间的切换控制如下图所示：

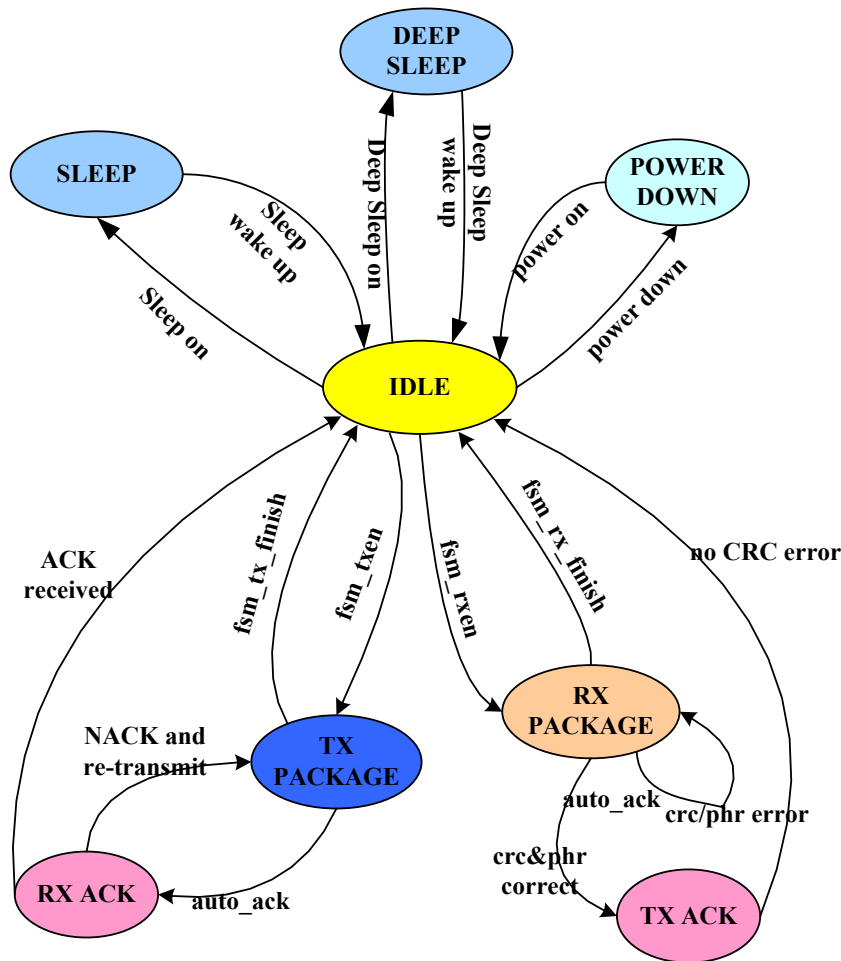


图 6-1 RF 收发器状态控制示意图

6.1 POWER DOWN

在 POWER DOWN 状态下，整个芯片全部掉电，芯片功耗约为 100nA。

通过设置 PDN 输入引脚为高电平进入 POWER DOWN 状态，设置 PDN 输入引脚为低电平，芯片从 POWER DOWN 状态恢复至 IDLE 状态，恢复时间约 2ms。

芯片从 POWER DOWN 恢复至 IDLE 状态，全芯片被复位，寄存器恢复默认值。通过读取 CHIPRDY_INT 中断标志确定芯片是否恢复至 IDLE 状态并且晶振稳定可进行收发。

6.2 DEEP SLEEP

在 DEEP SLEEP 状态下，数字的低功耗 LDO 工作，仅提供数字电路的状态寄存器供电，其

余数字电路处于掉电状态，芯片功耗约为 3 μ A。

在 WDT_EN (0x20) 不使能的情况下，通过设置 SLEEP_ON (0x1C) 寄存器为'1'，芯片进入 DEEP SLEEP 状态。设置 SLEEP_ON 寄存器为'0'，芯片从 DEEP SLEEP 状态唤醒，进入 IDLE 状态。

6.3 SLEEP (RC)

在 SLEEP 状态下，低功耗的数字 LDO 给整个数字电路供电，芯片功耗 3.5 μ A。

在 WDT_EN 使能的情况下，通过设置 SLEEP_ON 寄存器为'1'，芯片进入 SLEEP (RC) 状态。可通过设置 SLEEP_ON 寄存器为'0'、RC 定时或外部 GPIO 输入中断三种方式将芯片唤醒，唤醒后芯片进入 IDLE 状态。

芯片从 DEEP SLEEP/SLEEP 状态恢复至 IDLE 状态，芯片除寄存器外全部被复位。通过读取 CHIPRDY_INT 中断标志确定芯片是否恢复至 IDLE 状态并且晶振稳定可进行收发。

注 1: 在 SLEEP(RC)或 DEEP SLEEP 状态下 SPI 可访问寄存器，但不可访问 FIFO 或进行中断标志位清'0'操作。

注 2: 配置芯片进入 SLEEP(RC)或 DEEP SLEEP 状态之前，需关闭收发使能 DBUS_TXEN/DBUS_RXEN。

6.4 IDLE

在 IDLE 状态下，数字主 LDO 使能，晶振电路和晶振电路 BUFFER 电路都使能，芯片功耗小于 2mA。

6.5 TX

当使能芯片 DBUS_TXEN (0x01) 并且发送 FIFO 处于有效状态后，芯片进入 TX 发送状态。发送完成后，芯片进入 IDLE 状态。

6.6 RX

当使能芯片 DBUS_RXEN (0x01) 并且接收 FIFO 处于有效状态后，芯片进入 RX 接收状态。接收完成后，芯片进入 IDLE 状态。

6.7 芯片状态切换说明

芯片从上电、POWER DOWN 模式或从 SLEEP、DEEP SLEEP 模式恢复进入发送或接收模式示意图如图 6-2 与图 6-3 所示。

若芯片直接从 IDLE 状态进入发送或接收状态，可跳过芯片时钟稳定时间阶段。

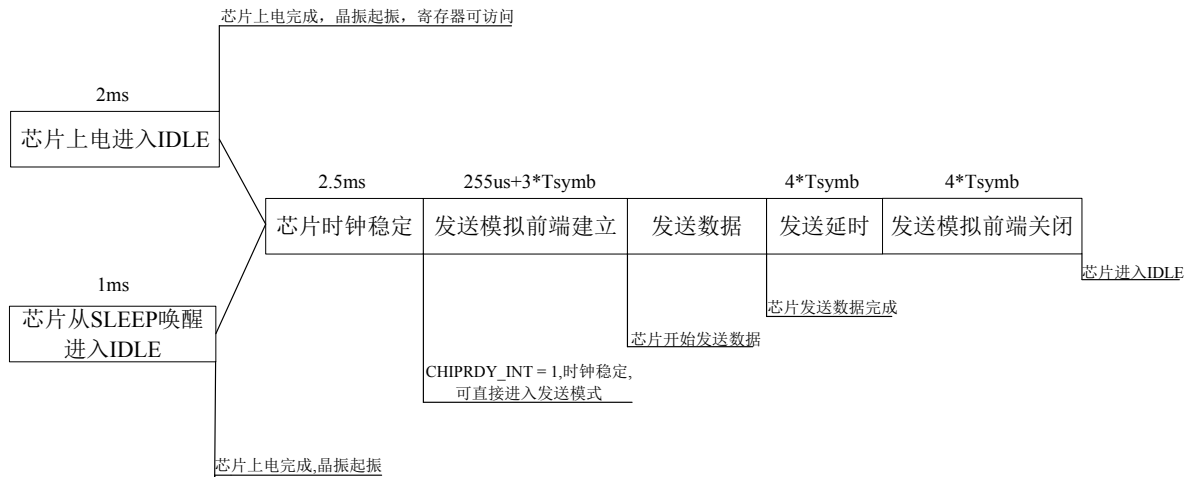


图 6-2 芯片上电进入发送模式示意图

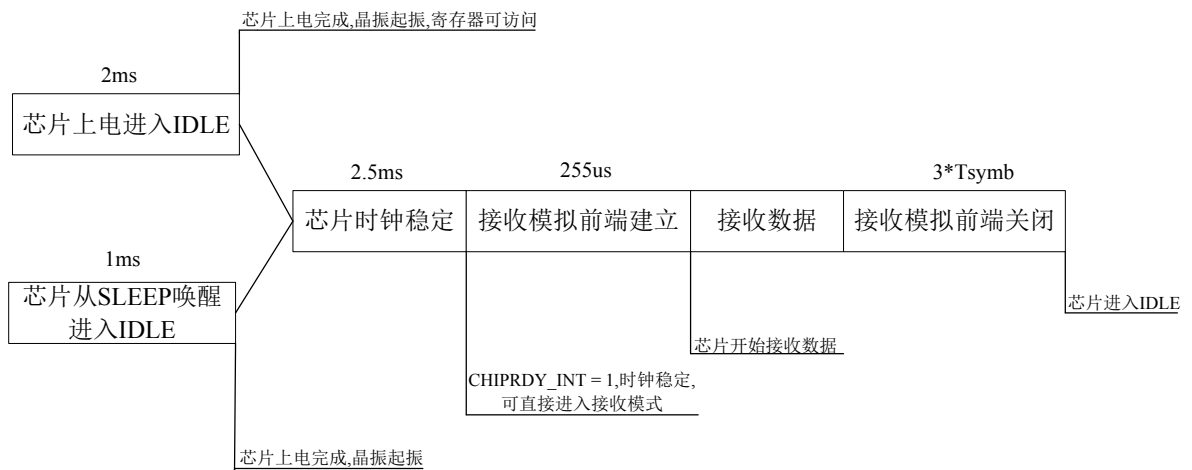


图 6-3 芯片上电进入接收模式示意图

注: T_{symb} 为数据速率周期,如数据速率为 10kbps,则 T_{symb} = 100 μs,以下相同。

第 7 章 RF收发器—帧结构

ES32W0030 芯片支持增强型帧结构与直接 FIFO 帧结构，可通过 PACK_LENGTH_EN 寄存器配置。增强型帧结构可兼容国家电网电力用户用电信息采集系统通信协议帧结构，并支持自动 ACK 功能；直接 FIFO 帧结构配置灵活，可兼容增强型帧结构、802.15.4g 协议等帧结构。

7.1 增强型帧结构

设置 PACK_LENGTH_EN = '1'，芯片选择增强型帧结构，如图 7-1 所示。

字节数：4~1023	2~6	1	1	1	1	0~252	2
前导码	帧分隔符	帧长	信道索引	标准识别号	帧头校验码	物理层载荷	帧校验序列
SHR		PHR				PSDU	FCS

图 7-1 增强型帧结构示意图

对所有具有多个字节的域，如前导码、帧分隔符、帧校验序列，芯片最先发射或接收最低比特位的字节；对每个字节，芯片最先发射或接收最低的比特位。

◆ 前导码 (Preamble)

前导码长度可配置 PREAMBLE_LEN 寄存器 (0x03)，支持 4~1023 个字节长度。前导格式可通过寄存器 PREAMBLE_MODE 配置 (0x03)。默认由 80 个字节组成，其顺序为“010101...01”。

◆ 帧分隔符 (SFD)

帧分隔符 SFD 长度可配置 SYNCWORD_LEN 寄存器 (0x03)，SFD 地址可配置 0x14~0x1A 寄存器。默认情况下，SFD 长度为 2bytes，SFD 地址指定为：低字节 0x98，高字节 0xF3。

◆ 物理层头 (PHR)

物理层头 PHR 包括帧长、信道索引、标准识别号、帧头校验码 4 个部分，每部分长度为 1 字节。发送 PHR 可通过寄存器 0x07、0x08 与 0x09 配置，其中帧头校验码、帧长支持硬件自动生成与软件配置两种方式（详见 0x07、0x09 寄存器说明）。

◆ 帧长

帧长域占 1 字节长度，表示长度为 PSDU 中包含的字节数加上 3（即加上信道索引、标准识别号、帧头校验码所占的 3 个字节）。

◆ 信道索引

信道索引为发送设备指定发送数据的无线信道号，信道索引表示公式为：

信道索引 = 信道组号×2 + 信道号

◆ 标准识别号

帧标准识别号为 1 字节长度，范围为 1~99 的 BCD 码。

标准识别号	描述
01	当前规范标准识别号
02~99	保留

表 7-1 标准识别号 (ACK 不使能)

当使能自动 ACK 功能时，发送帧标准识别号的 bit7~bit5 将自动插入 NOACK 与 PID 指示位，如表 7-2 所示。

◇ PID

PID 长度为 2bits，PTX 每成功接收一次 ACK 信号后将自动加'1'，PRX 依据 PID 与 CRC 值确定当前包为新接收包或重传包（丢弃）。PTX 若重传超时，在下次启动发送时 PID 不累加。

◇ NOACK

此指示位用于当 ACK 功能使能时，PTX 告知 PRX 当前数据包无需回复 ACK 的特例情况，可以通过 PTX_FIFO0_NOACK 寄存器（0x0C）进行设置。

Bit string index	0~4	5	6~7
Bit name	标准识别号	NOACK	PID

表 7-2 标准识别号（ACK 使能）

➢ 帧头校验码

帧头校验码占 1 字节长度，为物理层数据包 PHR 前面三个字节（帧长、信道索引、标准识别号）异或运算结果。

◆ 物理层载荷（PSDU）

物理层载荷域具有可变的长度，负责传送物理层数据包里的数据。最大支持 PSDU 长度为 252 bytes，即 FIFO 最大可填写 252bytes。

◆ 帧校验序列（FCS）

FCS 域为 16bits 的 CRC 序列，校验范围为 PHR-PSDU。

生成多项式 $G(X) = 1+X^5+X^{12}+X^{16}$ 。

注：PTX 表示发送方，PRX 表示接收方，以下相同。

7.2 直接FIFO帧结构

设置 PACK_LENGTH_EN = '0'，芯片选择直接 FIFO 帧结构。

字节数：4~1023	2~6	1~65535
前导码	帧分隔符	物理层载荷(FIFO)
SHR		PSDU

图 7-2 直接 FIFO 帧结构示意图

前导码与帧分隔符的配置方式与增强型帧结构方式一致，发送除前导与 SFD 部分都需软件填写至 FIFO 内，填写内容与顺序可依据收发双方的约定灵活设置。

直接 FIFO 帧结构不支持 ACK、硬件 CRC 校验、FEC 功能。

直接 FIFO 下发送可依据 LEN0_TXMODE 控制位配置发送完成模式，接收可依据 LEN0_RXMODE 控制位配置接收完成模式，详见 0x02 寄存器相关说明。

第 8 章 RF收发器—多PIPE逻辑通道

在增强型帧结构控制方式下，芯片支持 4 个数据 PIPE，即 PIPE0~PIPE3。

每个 PIPE 具有独立的物理地址 (SFD)，各 PIPE 地址配置详见寄存器 0x14~0x1A，各 PIPE 使能、各 PIPE 自动 ACK 功能使能详见寄存器 0x13。

各 PIPE 的 SFD 地址长度可通过 SYNCWORD_LEN 配置，默认 PIPE0 使能，长度为 2bytes，发送 SFD 为 0xF398。

PTX 在发送前使能需要的 PIPE，配置 PTX_FIFO0_PIPE 控制位 (0x0C 寄存器)，设置对应的 PIPE SFD 地址寄存器 Pn_ADDR 与 SFD 长度。

PRX 在接收到可同步的 PIPE SFD 之后(SFDDet_INT='1')，将数据包所属的 PIPE 信息存于寄存器 PRX_FIFO0_PIPE，若当前 PIPE 自动 ACK 功能使能，PRX 在接收完成后自动回复带有相同 PIPE 地址信息的 ACK 帧。

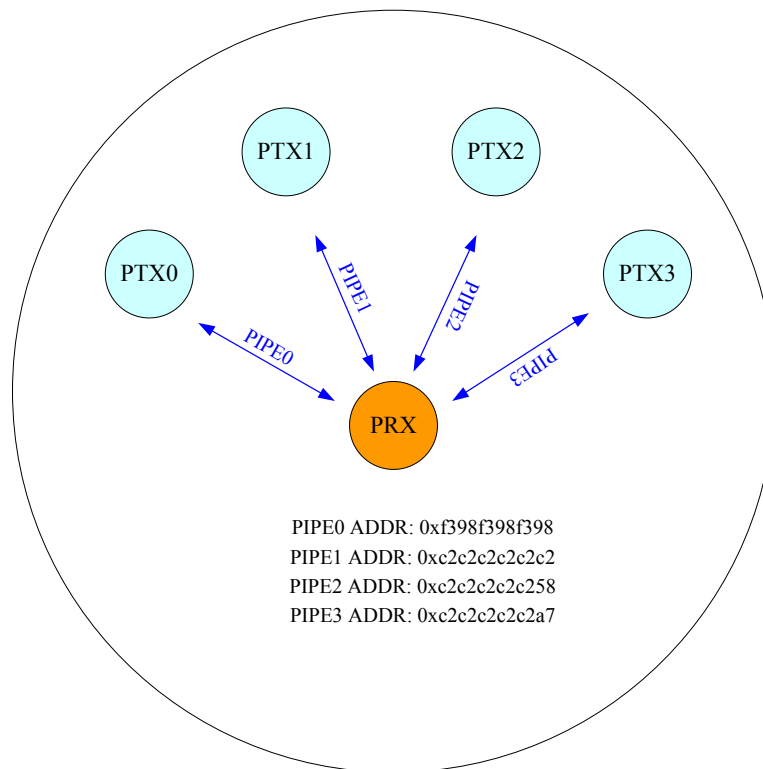


图 8-1 4 路 PIPE 通讯示意图

图 8-1 所示为 4 路 PIPE 通讯示意图，4 路 PIPE 都具有独立 PIPE 地址，PRX 最大可支持与 4 个不同 PIPE 地址的 PTX 通讯。

PRX 可分时与不同 PIPE 相关通讯，但在一路 PIPE 完整收发流程完成之前，PRX 不可与另一路 PIPE 地址的 PTX 通讯。当多路 PTX 与一路 PRX 通讯时，设置各 PTX 寄存器的 AUTO_RXACK_TIME 值 (0x45 寄存器) 可以有效避免各个 PIPE 之间的干扰。

注：SFD 同步电路触发工作条件由寄存器 SFD_OPT (0x02) 控制，包括以下三种：检测到有效前导触发工作、检测到有效载波能量触发工作或无条件始终工作，默认设置为检测到有效前导后触发工作，具体可参考寄存器说明。

第 9 章 RF收发器—自动ACK与ART

在增强型帧结构模式下，芯片支持自动 ACK 及自动重传功能（ART）。在自动 ACK 使能模式下，芯片将在增强型帧结构的基础上，在标准识别号字节中插入 PID 与 NOACK 指示位，详见表 7-2 说明。

发送 PID 由硬件自动生成，芯片在发送数据包完成后成功接收 ACK 信息则 PID 自动加‘1’；NOACK 指示位由 PTX_FIFO0_NOACK(0x0C) 寄存器设置。

- ◆ 当 PTX 与 PRX 同时使能当前通信 PIPE 的 ACK 功能后（Pn_EN = ‘1’并且 Pn_ACKEN = ‘1’）。PRX 在正确接收一帧数据后由 RX 模式自动切换为 TX 模式发送 ACK 帧，并在完成 ACK 帧发送后置起 FIFO0_INT（0x11）中断标志位（详见图 10-7）。PTX 在一帧数据发送完成后由 TX 模式自动切换为 RX 模式等待接收 ACK 帧，并在成功接收到 ACK 帧后置起 FIFO0_INT 中断标志位（详见图 10-7）。
- ◆ PTX 如果在 AUTO_RXACK_TIME 寄存器设置时间内没有成功接收到 PRX 发送的 ACK 帧，PTX 将自动重传上一帧数据包。若重传次数超过 RE_TX_TIMES 寄存器（0x45 寄存器）设定次数，FIFO0_INT 中断标志位与重传超时指示位 FIFO0_MAX_RETX（0x0F 寄存器）将同时置‘1’指示重传超时（详见图 10-8 至图 10-10）。
- ◆ PRX 在接收完成后若出现 PHR 或 CRC 错误将自动重收不回复 ACK 帧（详见图 10-11）。
 - ◆ 在 ACK 功能使能时，通过配置 PTX_FIFO0_NOACK 控制位可告知 PRX 当前发送帧无需回复 ACK 帧（详见图 10-12）。

第 10 章 RF收发器—数据与控制接口

10.1 FIFO

ES32W0030 支持 1 级 256bytes 独立收发 FIFO, 用于存放收发的数据, 其访问地址为 0x70。

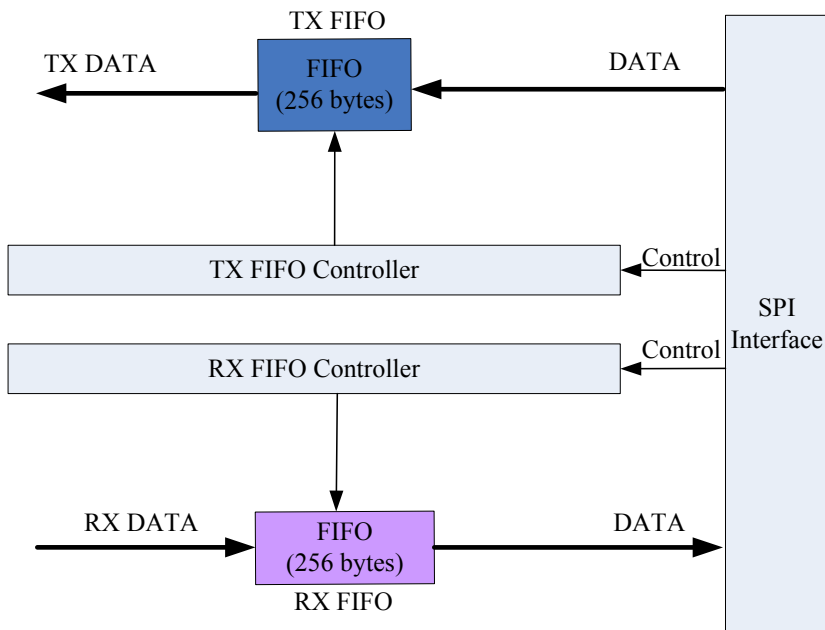


图 10-1 FIFO 控制示意图

10.1.1 FIFO指针

对于 PTX, 在开始发送之前可以通过 SPI 接口向 FIFO 内填写数据, 芯片在发送过程中 FIFO 的读取权限交于内部状态机, 在发送过程中禁止 SPI 接口读 FIFO 操作。

对于 PRX, 在接收完成后可以通过 SPI 接口读取 FIFO 内的接收数据, 芯片在接收过程中 FIFO 的写入权限交于内部状态机, 在此过程中禁止通过 SPI 接口写 FIFO 操作。

FIFO 的读写指针可通过访问 0x4D 寄存器获取, 通过操作 CLR_WR_PTR 或 CLR_RD_PTR 寄存器 (0x0E) 可对读写指针清'0'。

在收发过程中, 硬件自带对读写指针的清'0'机制, 在收发角色不切换的情况下无需软件参与读写指针的清'0'动作, 在收发角色切换的情况下 (接收切换为发送) 需要软件在写 FIFO 之前将写指针清'0'。

- ◆ 写指针在以下情况将自动清'0'
 - PTX 关闭发送 (DBUS_TXEN = '0')
 - PTX 清发送 FIFO0_INT 中断标志
 - PRX SFD 正确同步 (SFDDET_INT = '1')
- ◆ 读指针在以下情况将自动清'0'

- PTX 发送前导过程中
- PRX SFD 正确同步 (SFDDDET_INT = '1')

10.1.2 FIFO操作说明

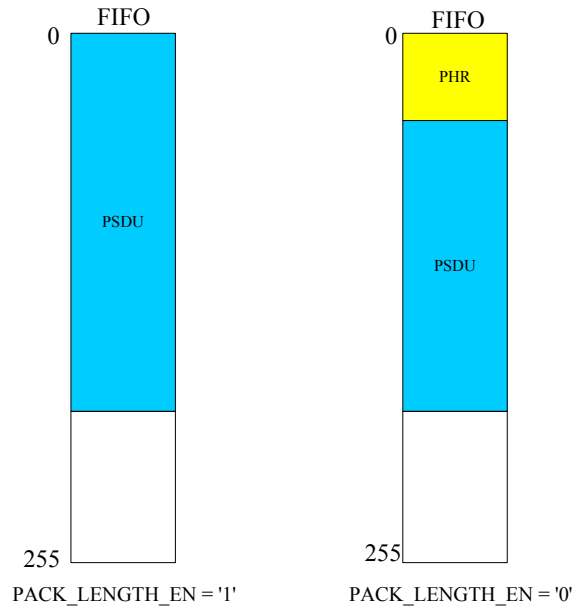


图 10-2 两种帧结构模式下 FIFO 结构

增强型帧结构模式下只需将 PSDU 填入 FIFO，直接 FIFO 帧结构若需兼容增强型帧结构模式需将 PHR 与 PSDU 依次填入 FIFO（如图 10-2 所示）。

增强型帧结构模式下芯片依据 PHR 设置的帧长信息自动完成 FIFO 内 PSDU 的发送与接收。

直接 FIFO 帧结构模式下芯片依据 `LEN0_TXMODE` 与 `LEN0_RXMODE` 的不同配置，完成 FIFO 内数据的发送与接收。

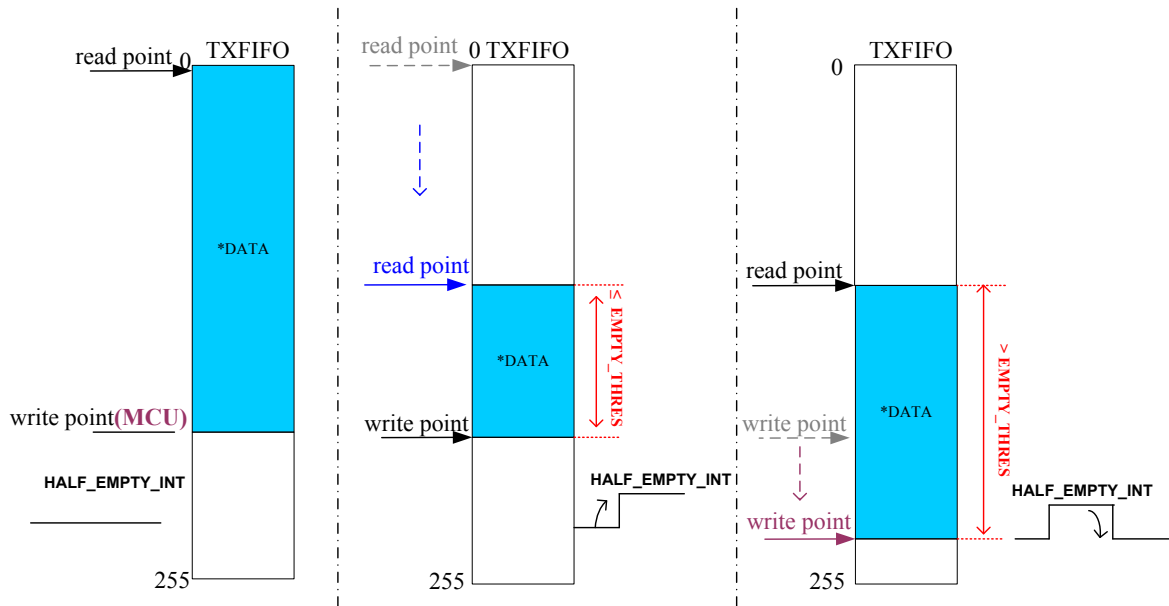


图 10-3 直接 FIFO 帧结构发送半空操作示意图

图 10-3 为直接 FIFO 帧结构模式下发送 FIFO 半空操作示意图。当写指针（MCU 通过 SPI 控制）与读指针（芯片状态机控制）之间的差值小于等于半空阈值 $EMPTY_THRES$ ($0x06$) 时，发送 FIFO 半空中断标志位 $HALF_EMPTY_INT$ ($0x0F$) 将会置'1'(软件可轮询中断标志位或通过 IRQ、GPIO 送出中断标志)，主控 MCU 响应中断向 FIFO 中补充数据，当写指针与读指针之间的差值大于半空阈值 $EMPTY_THRES$ 时，半空中断标志位 $HALF_EMPTY_INT$ 将会自动被清'0'。发送方主控 MCU 可依据半空标志完成发送数据长度大于 256bytes 情况的操作流程。

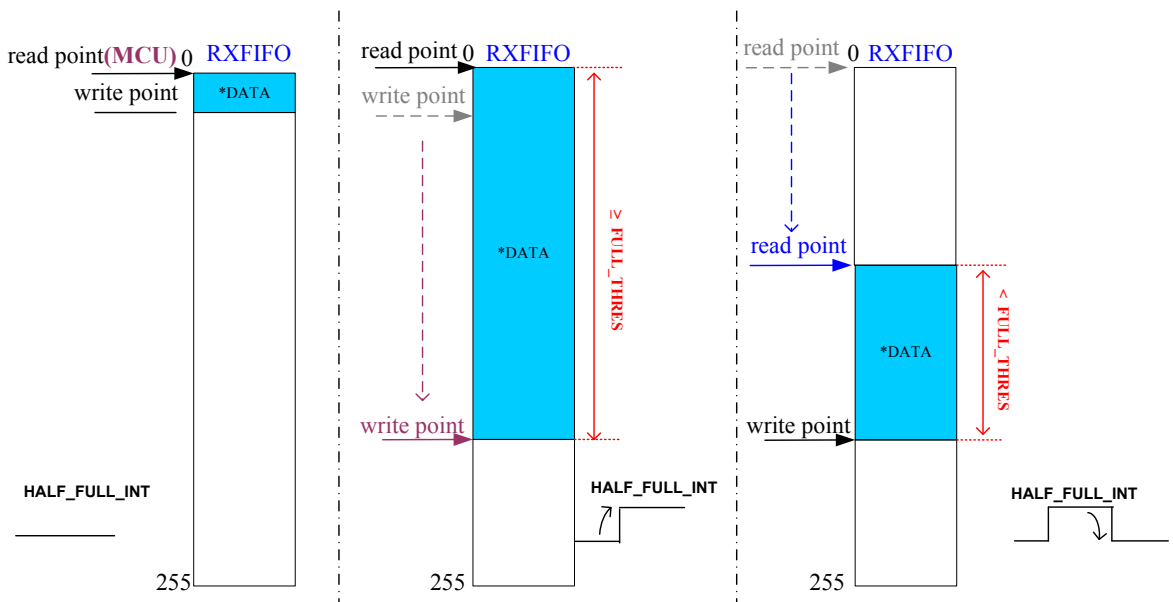


图 10-4 直接 FIFO 帧结构接收半满操作示意图

图 10-4 为直接 FIFO 帧结构模式下接收 FIFO 半满操作示意图。当写指针（芯片状态机控制）与读指针（MCU 通过 SPI 控制）之间的差值大于等于半满阈值 FULL_THRES（0x06）时，接收 FIFO 半满中断标志位 HALF_FULL_INT（0x0F）将会置'1'（软件可轮询中断标志位或通过 IRQ、GPIO 送出中断标志），主控 MCU 响应中断读取 FIFO 中数据，当写指针与读指针之间的差值小于半满阈值 FULL_THRES 时，半满中断标志位 HALF_FULL_INT 将会自动被清除'0'。接收方主控 MCU 可依据半满标志完成接收数据长度大于 256bytes 情况的操作流程。

注 1：硬件对读写指针有溢出保护处理，读写指针大于 255 时返回地址 0。

注 2：半空半满阈值可通过寄存器 0x06 配置，阈值设置的大小需结合当前数据速率与 SPI 的通讯速率。

10.2 中断

通过配置中断使能 INTIE（0x10）寄存器，可将芯片内部各中断标志通过 IRQ 引脚输出，方便外部 MCU 获知芯片的状态进行相应的中断处理。GPIO 也可输出中断标志，具体见 0x1D 及 0x1E 中 GPIO 功能选择位的描述。

中断源包括：

- ◆ POR 中断
- ◆ chip ready 中断
- ◆ 有效 SFD 中断
- ◆ 无效 SFD 中断
- ◆ 有效前导中断
- ◆ 无效前导中断
- ◆ GPIO 外部中断
- ◆ WDT 中断
- ◆ 发送 FIFO 半空、接收 FIFO 半满中断(只在直接 FIFO 帧结构下有效，详见 10.1.2 章节说明)
- ◆ RSSI 中断
- ◆ 数据同步时钟中断（直接收发模式时有效）
- ◆ FIFO 中断

每个中断源都有相应的中断使能位，若中断不使能，中断源只作为内部标志位使用。各中断源详见 0x0F 寄存器说明。

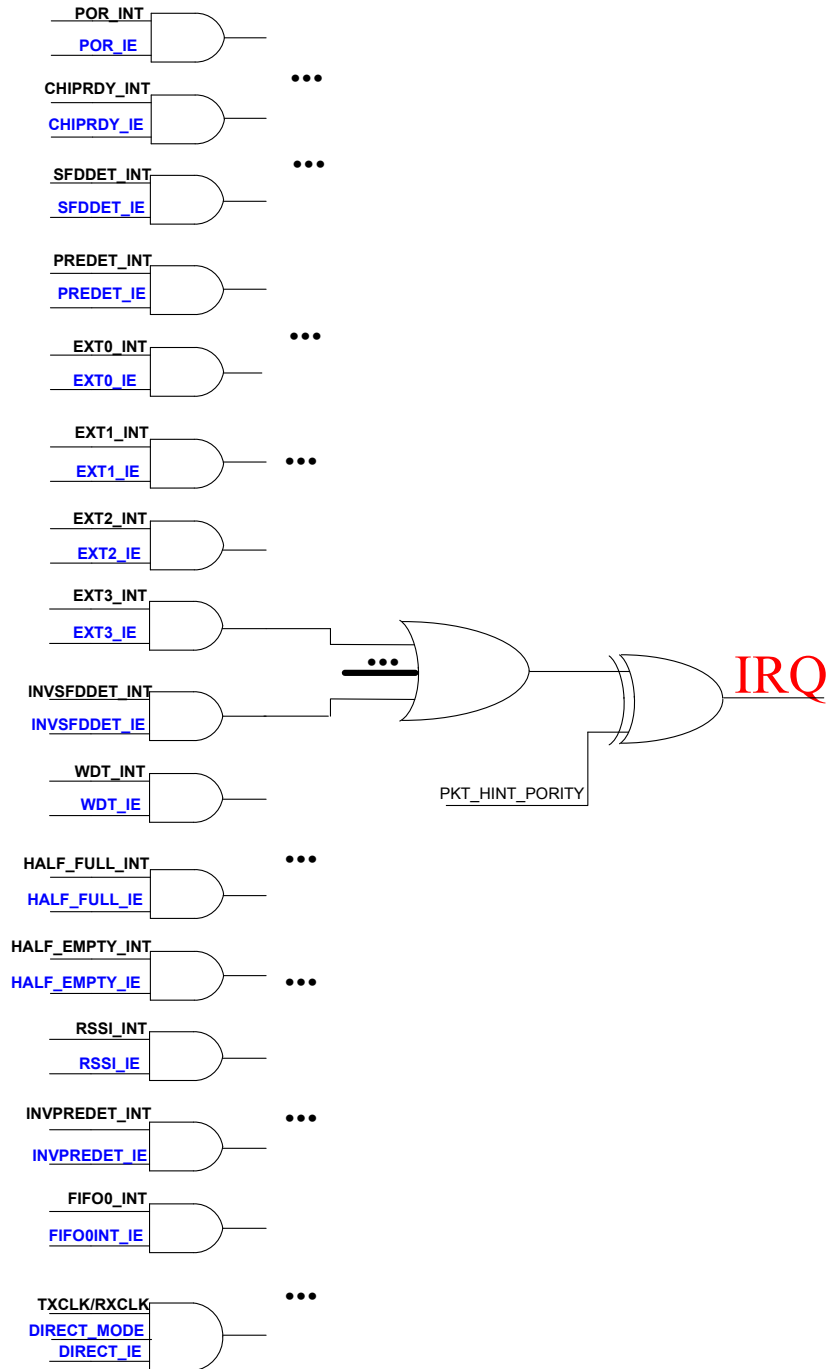


图 10-5 IRQ 中断输出示意图

其中，发送方 FIFO 中断 FIFO0_INT 指示 FIFO 发送完成、ACK 接收成功、或重传超时（FIFO0_MAX_RETX 为‘1’）；接收方指示接收完成、ACK 发送完成，各情况具体可参考图 10-6 至图 10-13。

中断输出的有效电平可以由中断极性控制位（PKT_HINT_PORITY）配置，默认为低电平中断有效。

中断及其标志位可以通过寄存器 INTIC(0x11)清‘0’或特定条件清‘0’，详见 INT 寄存器(0x0F)说明。

10.2.1 增强型帧结构中断

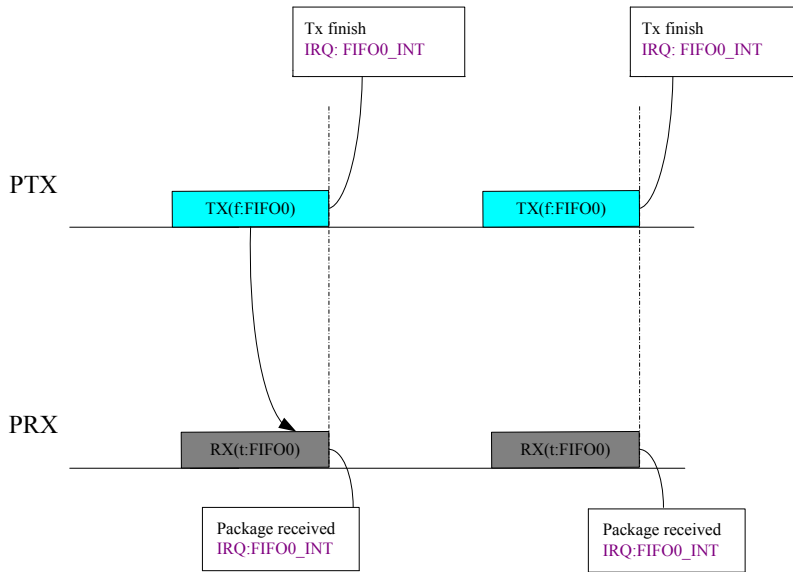


图 10-6 ACK 不使能情况中断示意图

ACK 不使能情况下($Pn_ACKEN=0$), PTX 在发送完成后中断标志位 FIFO0_INT 将置'1', PRX 在接收数据包完成后中断标志位 FIFO0_INT 将置'1' (如图 10-6 所示)。接收端在响应中断时需检查 PRX_PHR_ERR 与 PRX_CRC_ERR 指示位 (0x0C 寄存器) 判断所收取包 PHR 与 PSDU 是否正确。

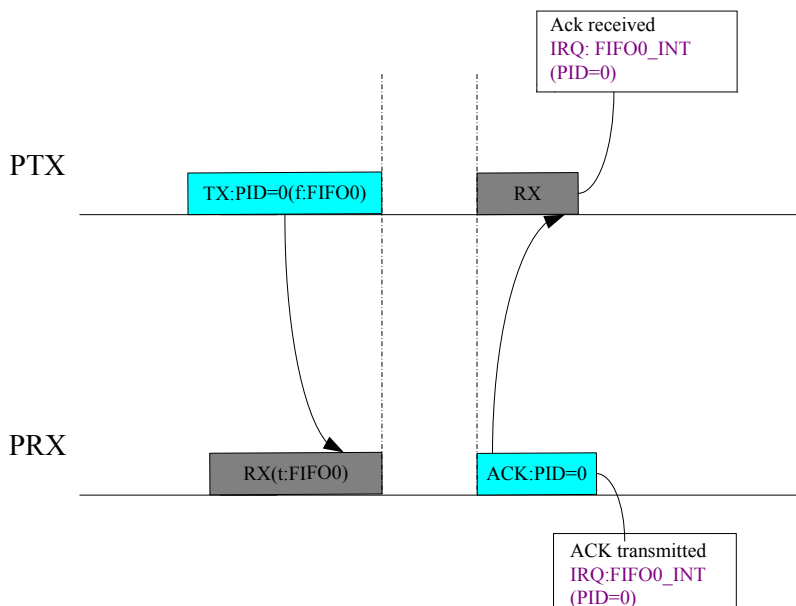


图 10-7 ACK 使能正常通讯中断示意图

ACK 使能情况下($Pn_ACKEN=1$), PTX 在成功接收 ACK 后, 中断标志位 FIFO0_INT 将置'1'。若 PRX 成功接收数据包(PHR 与 PSDU 正确), 在回复 ACK 后中断标志位 FIFO0_INT 将置'1', 如图 10-7 所示。

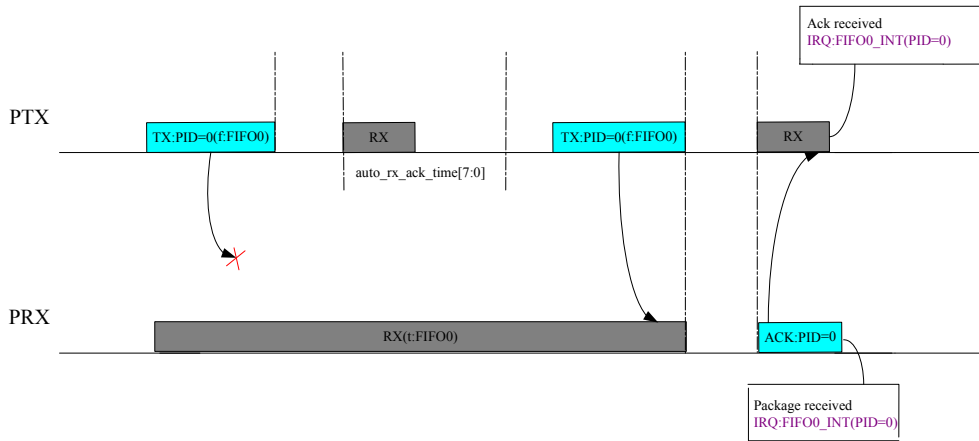


图 10-8 PTX 自动重传情况 1 中断示意图

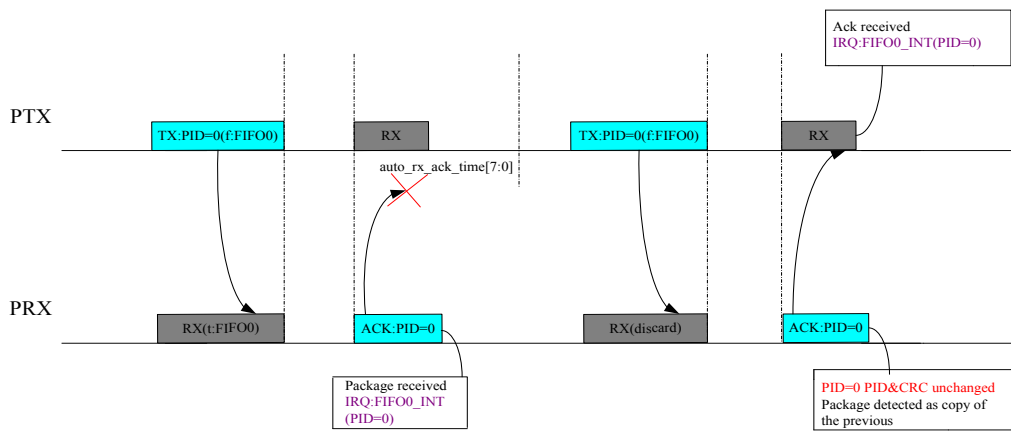


图 10-9 PTX 自动重传情况 2 中断示意图

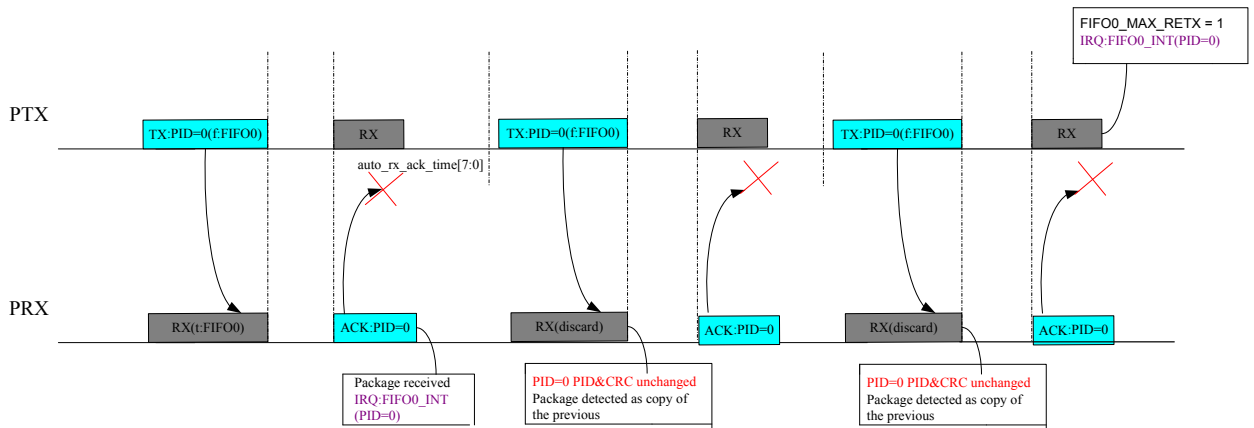


图 10-10 PTX 自动重传情况 3 中断示意图(重传超时, 重传次数为 2)

图 10-8 至图 10-10 所示为可能出现的收发不成功情况,PTX 在发送数据包完成后切换为接收 ACK 状态,若在等待 ACK 时间内(AUTO_RXACK_TIME 寄存器设置)没有收到有效 ACK 帧,将自动切换为自动重传,重传过程中发送帧 PID 保持不变。

PRX 在成功接收数据包后自动判断收取的数据帧 PID,若当前收取的 PID 与 CRC 信息与前一次相同,将视为重收包,不再出中断(只返回 ACK)。

注 1: PTX 自动重传功能只在 ACK 使能的情况下有效。
 注 2: 如果 PTX 在重传 RE_TX_TIMES 次数内没有正常接收到 ACK, PTX 在置起中断 FIFO0_INT 的同时, FIFO0_MAX_RETX 将置'1'。
 注 3: 重收弃包功能只在 ACK 使能的情况下有效。

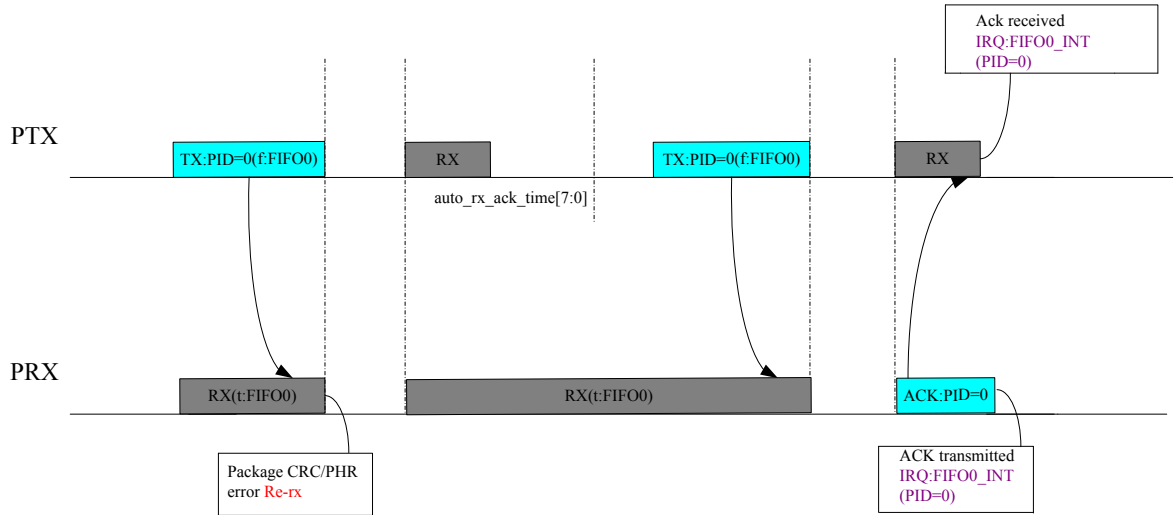


图 10-11 PRX 自动重收中断示意图

图 10-11 所示为 PRX 自动重收的情况, PRX 在接收到数据包之后若检测 CRC 或 PHR 有误将不返回 ACK 而切换至自动重收, 直至收到正确的数据包返回 ACK 之后中断标志位 FIFO0_INT 将置'1'。

注 1: PRX 自动重收功能只在 ACK 使能的情况下有效。

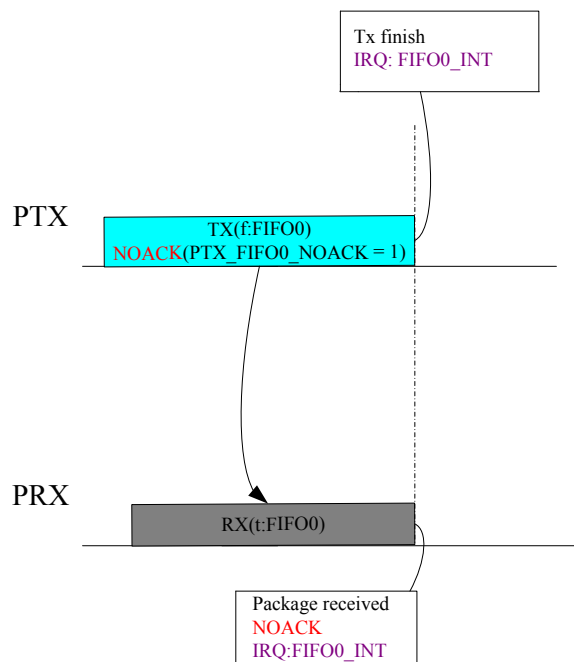


图 10-12 ACK 使能情况下 NOACK 功能中断示意图

图 10-12 所示为收发 ACK 使能时, 若设置 PTX_FIFO0_NOACK 为'1', PTX 在发送数据包完

成后中断标志位 FIFO0_INT 将置'1'，PRX 在接收数据包完成后中断标志位 FIFO0_INT 将置'1'。与 ACK 不使能情况类似，接收端在响应中断时需检查 PRX_PHR_ERR 与 PRX_CRC_ERR 指示位判断所收取包 PHR 与 PSDU 是否正确。

10.2.2 直接FIFO帧结构中断

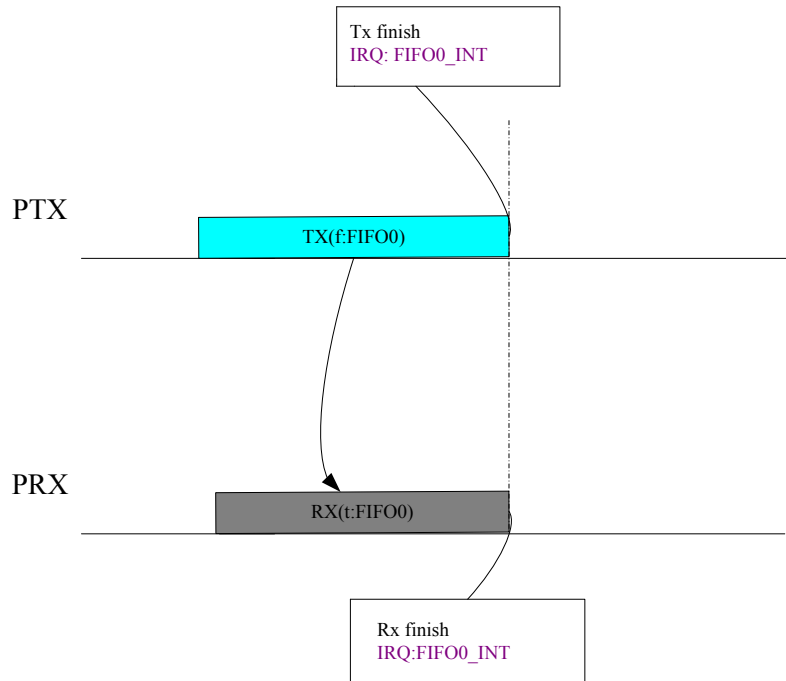


图 10-13 直接 FIFO 帧结构中断示意图

直接 FIFO 帧结构情况下 PTX 发送数据包完成后中断标志位 FIFO0_INT 将置'1'，PRX 在接收数据包完成后中断标志位 FIFO0_INT 将置'1'，收发完成条件配置详见 12.2 章节说明。

10.3 SPI通信接口

ES32W0030 支持 4 线 SPI 通信接口，可访问芯片内部寄存器与收发 FIFO，支持对 FIFO 连读连写操作。

10.3.1 SPI帧格式

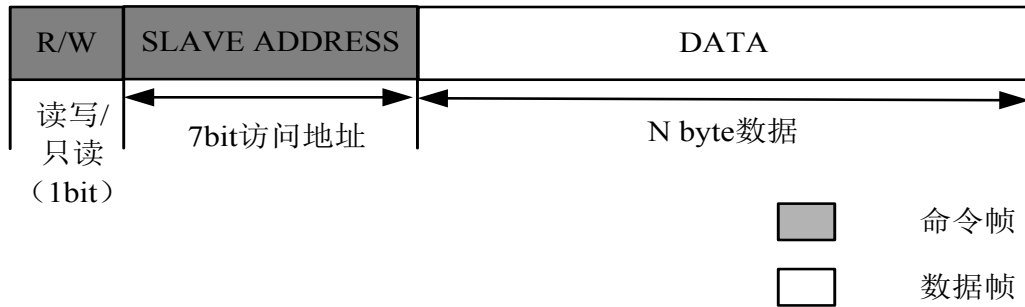


图 10-14 SPI 帧格式

SPI 帧格式由命令帧与数据帧两部分构成。

其中 8bits 命令帧首位为读写控制位，'0'为只读操作，'1'为读写操作（读取某寄存器值的同时可以写入值，对 FIFO 为只写操作），后 7 位为读写地址位。

数据帧按 SPI 接口访问对象不同有所区别，若 SPI 访问内部寄存器则数据帧固定为 2bytes，若 SPI 访问 FIFO，在非连读连写情况下为 1byte（内部 FIFO 数据宽度），在连读连写情况下数据帧长度由主设备访问长度决定。

数据格式为高位（MSB）在前低位（LSB）在后。

10.3.2 寄存器访问时序

SPI 接口读写寄存器操作时序如图 10-15 与图 10-16 所示。

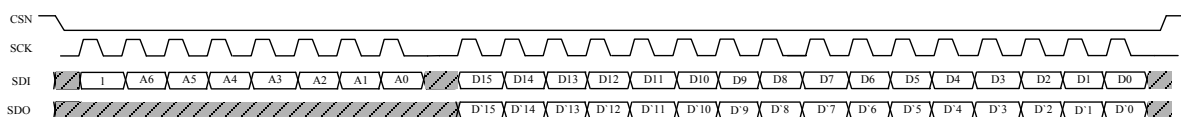


图 10-15 SPI 写寄存器时序

注：写寄存器时 SDO 送出的 D'x 为所访问寄存器的原值。

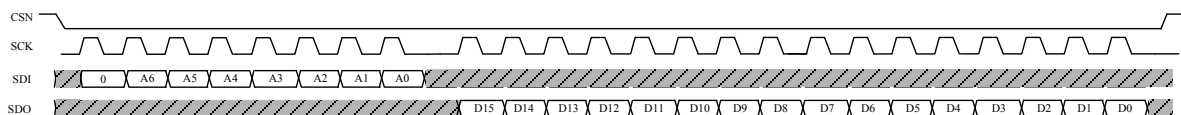


图 10-16 SPI 读寄存器时序

10.3.3 FIFO访问时序

FIFO 支持连续读写操作，最小单位为 1byte，FIFO 访问地址为 0x70。SPI 接口读写 FIFO 操作时序如图 10-17 与图 10-18 所示。

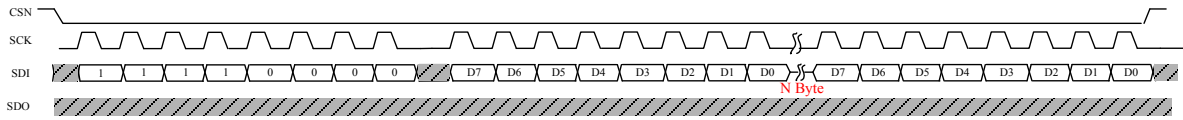


图 10-17 SPI 写 FIFO 时序

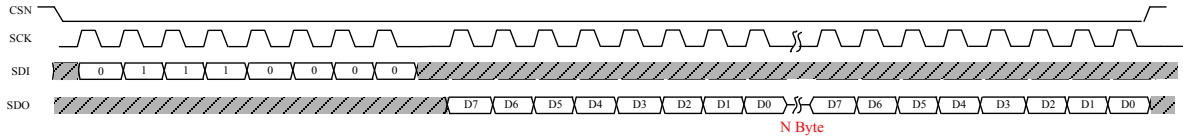


图 10-18 SPI 读 FIFO 时序

10.3.4 SPI时序参数

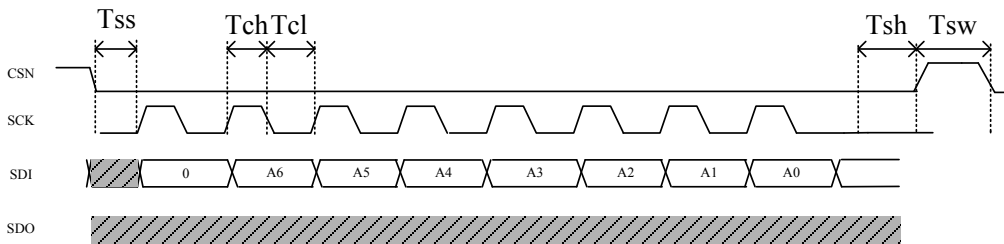


图 10-19 SPI 时序参数示意图

Symbol	Parameter	T(ns)
Tss	Select setup time	20
Tcl	Clock low time	62.5
Tch	Clock high time	62.5
Tsh	Select hold time	40
Tsw	Select high period	200

表 10-1 推荐 SPI 时序 (8MHz) 参数

10.4 GPIO配置

ES32W0030 支持四组 GPIO，分别为 GPIO0、GPIO1、GPIO2、GPIO3，可通过寄存器 IO_CFG0 (0x1D) 及 IO_CFG1 (0x1E) 对 GPIO 进行配置。

GPIO 默认状态如表 10-2 所示：

GPIO	默认状态
GPIO0	POR 输出
GPIO1	POR 反向输出
GPIO2	芯片时钟输出 (晶振频率/64)
GPIO3	保留状态 (输入)

表 10-2 GPIO 默认状态

当 GPIO 设置为输入功能时 (GPIO_n_IE = '1')，通过寄存器 GPIO_n_PE 可以配置 GPIO 管脚的上拉功能，在默认状态下，GPIO 管脚无上拉电阻。当设置 GPIO_n_IE 为'0'时，屏蔽 GPIO 的外部输入，芯片内部接收为低电平，默认状态下，GPIO 不屏蔽外部输入。

当 GPIO_n_FUN 配置为输出功能时，GPIO 的输出使能自动打开，GPIO_n 的输出有效电平可以通过 GPIO_n_PORITY (0x1C) 进行配置；通过寄存器 GPIO_n_DS 可以配置 GPIO 的输出驱动电流，芯片的默认状态为大电流模式。

GPIO 支持内部 clock 输出功能 (设置 GPIO_n_FUN=5'b01111)，以方便某些低 BOM 应用场合，向主控 MCU 提供时钟。

芯片上电默认从 GPIO2 输出主晶振 64 分频时钟，用户可配置 GPIO_n_FUN 选择其它 GPIO，输出时钟频率可通过寄存器 CLKCFG (0x1C) 配置。

当设置 LPFEN (0x1C) 为'1'且 WDT_EN 为'1'时，当芯片进入 SLEEP(RC)模式后将自动切换为 RC 时钟输出。

第 11 章 RF收发器—频点与数据传输参数设置

11.1 频点设置

ES32W0030 共支持 315/433/779/868/915MHz 5 个典型频段，其中 433MHz 频段频点支持默认设置与直接设置两种模式，其余频段只支持直接设置模式。ES32W0030 支持 20/26MHz 晶振，对于不同的晶振，支持的频点范围为：

- 270-348MHz/403-522MHz/806-1045MHz (26MHz 晶振)
- 235-522MHz/700-1045MHz (20MHz 晶振)

芯片默认配置下支持 433 频段，779 频段仅在 20MHz 晶振模式下支持。详细频段设置参考表 11-1 和表 11-2。

20M 晶振模式				
VCO_HB_SEL	PLL_TRXLB_SEL			
	2'b00	2'b01	2'b10	2'b11
1'b0	700-874MHz	350-437MHz	235-291MHz	-
1'b1	856-1045MHz	428-522MHz	285-350MHz	-

表 11-1 20MHz 晶振下各频段范围与配置

26M 晶振模式				
VCO_HB_SEL	PLL_TRXLB_SEL			
	2'b00	2'b01	2'b10	2'b11
1'b0	806-874MHz	403-437MHz	270-291MHz	-
1'b1	856-1045MHz	428-522MHz	285-348MHz	-

表 11-2 26MHz 晶振下各频段范围与配置

11.1.1 频点默认设置模式

在数据收发时，收发两端的频点设置值必须一致。

频点起始值可以通过寄存器 RF_FREQ_BASE (0x2F) 配置，信道间隔由寄存器 CH_SPACE (0x2F) 配置，信道号可通过寄存器 CHANNEL1~CHANNEL8 (0x28~0x2B) 配置。

其中 CHANNEL1 为主信道号，CHANNEL2~CHANNEL8 为从信道号，从信道号只在信道跳频模式开启后有效。信道号的取值可以从 0 到 255，也就是在单个应用中最多支持 256 个信道。

RF_FREQ_BASE 寄存器设置值以 100kHz 为单位，即起始频点为：

RF_FREQ_BASE*100(kHz)。

(例：需要的起始频点为 433MHz，RF_FREQ_BASE 设置值为 0x10EA，即十进制数 4330)

CH_SPACE 确定频点间隔：

00: 间隔为 100kHz

01: 间隔为 200kHz

10: 间隔为 400kHz

11: 间隔为 800kHz

最终发送或接收的主信道频点值为:

$$(RF_FREQ_BASE + CHANNEL1 * 2^{CH_SPACE}) * 100(kHz)。$$

11.1.2 频点直接设置模式

在直接频点设置模式下 (RF_PLL_DIRECT = '1', 0x30 寄存器), 可通过软件直接设置 Sigma-Delta 调制器的整数部分 (Integer) 和小数部分 (Fraction) 分频比。

整数部分可通过 RF_FREQ_BASE 寄存器设置, 小数部分可通过 RF_FREQ_FRACTION 寄存器 (0x30、0x31) 设置, 具体的计算公式如下:

$$RF_FREQ_BASE = \text{floor}(f_c * k_{\text{freq}} / f_{\text{xtal}})$$

$$RF_FREQ_FRACTION = \text{round}((f_c * k_{\text{freq}} / f_{\text{xtal}} - \text{Integer}) * 2^{21})$$

其中, f_c 为需要设置的信号载波频率, f_{xtal} 为使用晶振的频率, 系数 k_{freq} 配置详见表 11-3。

k _{freq} 值配置	
PLLTRXLB_SEL	k _{freq}
2'b00	2
2'b01	4
2'b10	6

表 11-3 系数 k_{freq} 配置表

直接频点设置顺序:

1. 使能 RF_PLL_DIRECT 并设置 RF_FREQ_FRACTION 寄存器
2. 设置 RF_FREQ_BASE 寄存器

注: 直接设频点模式下, 芯片从 DEEP SLEEP/SLEEP 唤醒后需重新配置频点。

11.1.3 跳频接收模式

ES32W0030 支持硬件跳频接收功能 (HOP_ENABLE = '1', 0x2D 寄存器)。在硬件跳频接收模式下, 接收端将以设定的时间间隔扫描需要的各个频点, 当发现某一频点接收到有效信号后, 接收会停留在该频点完成数据包的接收。

系统最大可以扫描 8 个频点, 具体需要扫描的频点数可以通过寄存器 HOP_NUM (0x2D) 设置, 各扫描频点可以通过寄存器 CHANNEL1~CHANNEL8 设置。

在频点默认设置模式下各信道频点: 主信道频点 + Δf_1 (CH_SPACE 设置的频点间隔) * CHANNELN

在频点直接设置模式下各信道频点: 主信道频点 + Δf_2 (HOP_SPACE (0x3C、0x3D) 设置的频点间隔) * CHANNELN, HOP_SPACE 具体的计算公式如下:

$$HOP_SPACE = \text{round}((f_h * k_{\text{freq}} / f_{\text{osc}}) * 2^{20})$$

其中 f_n 为需要的扫描频率间隔， k_{freq} 取值详见表 11-3， f_{osc} 为晶振频率。

每个频点的扫描时间间隔可以通过寄存器 HOP_TIMER (0x2D) 设置，实际的扫描时间间隔为：

$$T_s = (HOP_TIMER + LP_TIMER) * 1 / (10^3 * DR_GFSK)$$

其中 DR_GFSK 为数据速率 (kbps)，LP_TIMER (0x25) 为低功耗接收模式时间配置。

跳频锁定的频点与当前跳频所在频点可通过寄存器 HOPCH (0x3F) 读取。

注：跳频中的有效信号可选择接收信号能量满足条件或接收到有效前导，可通过 HOP_SEL 寄存器 (0x25) 配置。

11.2 数据传输参数设置

11.2.1 数据速率设置

ES32W0030 支持数据速率为 1.2~100kbps，可通过寄存器 SYMBOL_RATE (0x32、0x33) 设置，具体设置方法如下：

$$SYMBOL_RATE = \text{round}((DR_GFSK * 2^{22}) / (2 * 10^6))$$

其中 DR_GFSK 为数据速率 (kbps)。

例如：选择速率速率为 10kbps， $SYMBOL_RATE = \text{round}((10 * 10^3 * 2^{22}) / (2 * 10^6)) = 20972$ ，转换为 16 进制为 18'h051EC。

11.2.2 发送调制特性设置

◆ GFSK BT

通过寄存器 BT_MODE (0x25) 将 BT 设置为 0.5 或者 1，默认为 0.5。

◆ 调制频偏

$$DEVIATION = \text{round}(f_{dev} * k_{dev} / f_{xtal})$$

其中 f_{dev} 为期望调制频偏，支持范围为 0~50kHz， f_{xtal} 为芯片使用的晶振频率，系数 k_{dev} 配置详见表 11-4。

k _{dev} 值配置	
PLLTRXLB_SEL	k _{dev}
2'b00	2 ¹⁵
2'b01	2 ¹⁶
2'b10	3*2 ¹⁵

表 11-4 系数 k_{dev} 配置表

例如：期望调制频偏：25kHz，使用晶振频率 26MHz。

$$DEVIATION = \text{ROUND}((25 * 10^3 * 2^{16}) / (26 * 10^6)) = 63, \text{ 转换为 16 进制为 } 10'h03F.$$

11.2.3 发送功率设置

发送功率由 PA_VB1 和 PA_VB2 (0x40) 设置，设置值与输出功率的对应关系详见《AN1045_应用笔记_HW3000 应用注意事项》3.5 章节相关说明。

11.2.4 接收滤波器带宽设置

通过设置寄存器 FILTER2_BAND (0x2E) 可以改变接收端的滤波器带宽，使用的滤波器带宽值为

$$BW = (FILTER2_BAND+1)*1kHz$$

可配置的范围 1kHz ~ 128kHz。

第 12 章 RF收发器—收发操作流程

12.1 增强型帧结构收发流程

对于 PTX, 通过 SPI 接口设置 PHR (TXPHR 和 PSDULEN) 并向 FIFO 内填写发送的 PSDU 数据。填写 PSDU 完成之后需设置 PTX_FIFO0_PIPE 并将 PTX_FIFO0_OCPY(0x0C)置'1', 以指示 FIFO 被占用并等待发送。

待 PTX 发送完成中断置起后, 需软件清中断标志位 FIFO0_INT, 关闭 DBUS_TXEN 完成一次发送流程。

增强型帧结构收发流程详细参考《AN1045_应用笔记_HW3000 应用注意事项》2.2.1 章节。

图 12-1 为简单的 PTX 发送流程图。

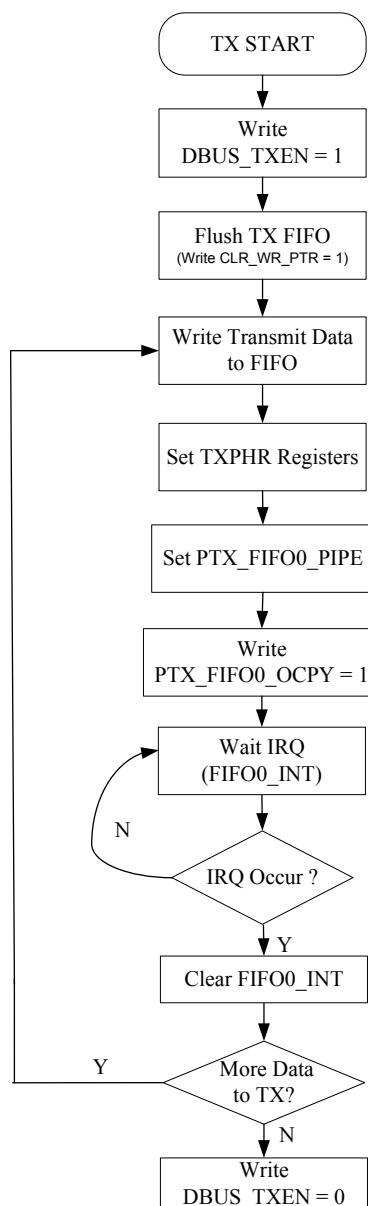


图 12-1 PTX 发送流程图

在发送过程中，芯片根据发送的 PHR 以及 PSDU 自动计算 CRC，并将计算出的 CRC 跟随在 PSDU 之后进行发送。可以设置 LEN1_CRCSEL (0x02) 为 '1' 屏蔽硬件 CRC 的功能，将寄存器 CRCVAL (0x12) 中的数值作为数据包的 CRC，跟随 PSDU 进行发送。

对于 PRX，硬件在正确同步 SFD 之后解析 PHR，得到包括数据长度在内的相关信息，然后向 FIFO 内填写接收的 PSDU 数据，接收完成后中断标志位 FIFO0_INT 将置 '1'。

软件可读取 PHR 数据 (RX_PHR0 和 RX_PHR1)、FIFO 的数据以及 CRC 的校验值 (CRCVAL)，之后需软件清中断标志位 FIFO0_INT。图 12-2 为简单的 PRX 接收流程图。

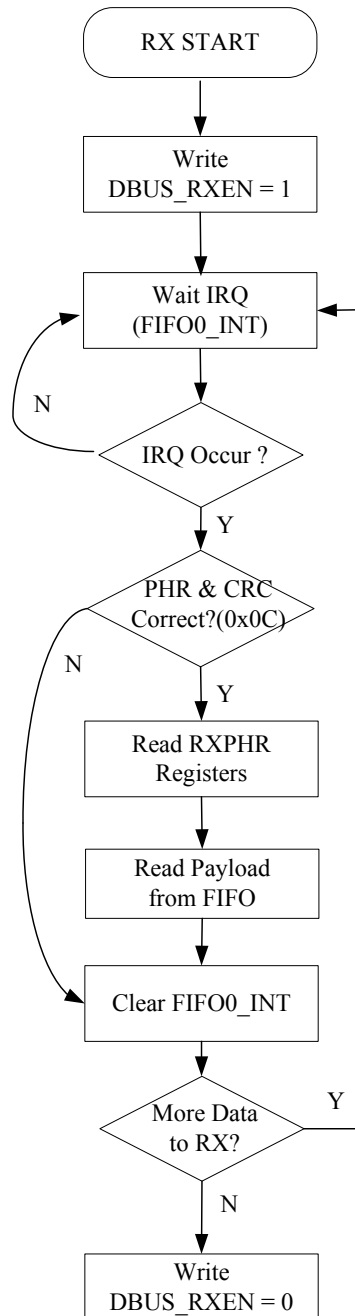


图 12-2 PRX 接收流程图

注 1: 关闭发送或接收使能后需等待 3*T_{symp} 以上的时间间隔后才能再次使能发送或接收。
 注 2: 在自动 ACK 使能的情况下 (Pn_ACKEN='1'), 必须使用硬件进行 CRC 校验, 即 LEN1_CRCSEL 设为 '0'。

12.2 直接FIFO帧结构收发流程

与增强型帧结构类似，PTX 在发送前需将 PTX_FIFO0_OCPY 位置'1'，以指示所填的 FIFO 被占用。

直接 FIFO 帧详细收发流程参考《AN1045_应用笔记_HW3000 应用注意事项》2.2.2 章节。

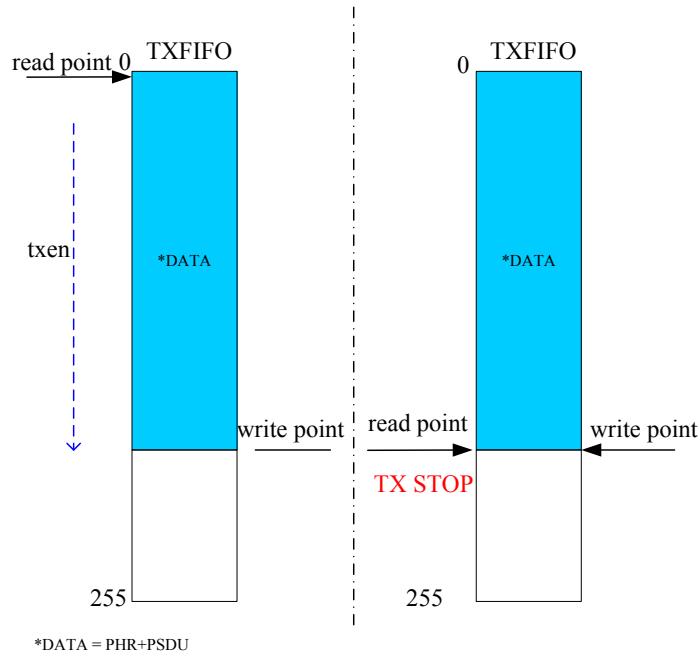


图 12-3 直接 FIFO 帧结构 LENO_TXMODE='1'发送停止示意图

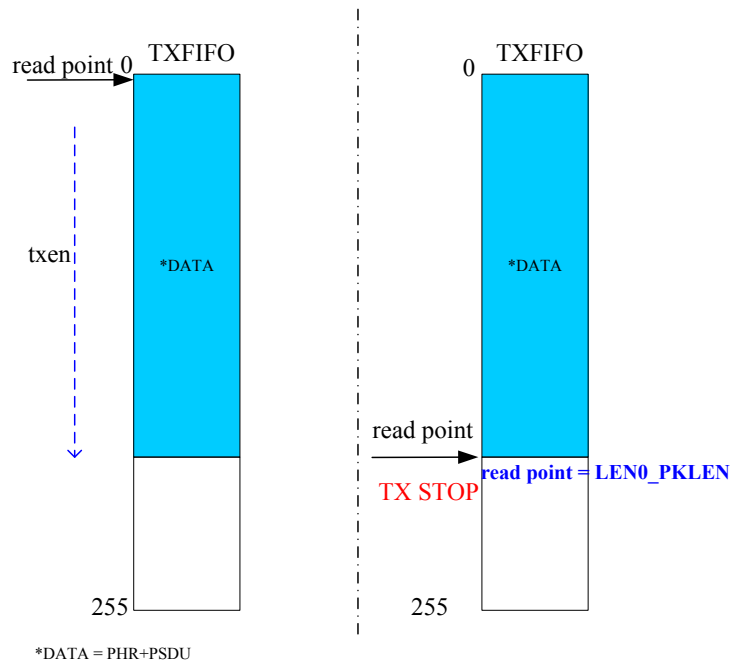


图 12-4 直接 FIFO 帧结构 LENO_TXMODE='0'发送停止示意图

若配置 LEN0_TXMODE 为'1'， PTX 在读写指针相等时自动停止发送（如图 8-3 所示），发送完成后将中断标志 FIFO0_INT 置'1'；

若配置 LEN0_TXMODE 为'0'， PTX 按照 LEN0_PKLEN 寄存器（0x0D）设置值发送（如图 12-4 所示），发送完成后将中断标志 FIFO0_INT 置'1'。

若配置 LEN0_RXMODE 为'1'， PRX 按照 LEN0_PKLEN 设置值进行接收。PRX 可通过响应半满中断标志 HALF_FULL_INT 获取 FIFO 内收取数据包携带的长度信息，并根据该长度信息设置接收数据长度寄存器 LEN0_PKLEN, PRX 接收完成后将中断标志 FIFO0_INT 置'1'；

若配置 LEN0_RXMODE 为'0'， PRX 按照用户设定的帧长信息配置（如图 12-5）自动完成接收。帧长信息配置详见寄存器 0x0E， PTX 需按照规定的帧长信息位置、长度等配置信息填充发送 FIFO， PRX 自动识别帧长信息接收完成后将中断标志 FIFO0_INT 置'1'。

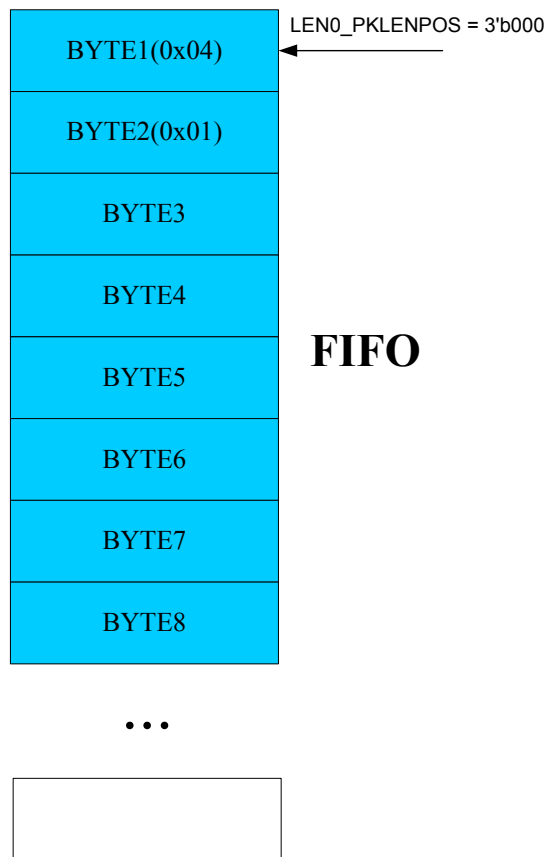


图 12-5 帧长信息配置示意图

以图 12-5 为例，若设定 LEN0_PKLENPOS = 3'b000，则 FIFO 内以第一个 byte 作为自动接收帧长信息起始 byte。

- ◆ 若 LEN0_PKLENBYTES = '0'，则以 0x04 作为自动接收帧长度信息，PRX 将共接收 4 +LEN0_PKLENADD bytes 数据后停止接收，其中 LEN0_PKLENADD (0x4E) 为额外接收 byte 数寄存器，由软件根据需要调整。
- ◆ 若 LEN0_PKLENBYTES = '1'， LEN0_PKLENPOLAR = '0'，则以 0x0401 作为自动接收帧长度信息，PRX 将共接收 1025 (0x0401) +LEN0_PKLENADD bytes 数据后停止接收。
- ◆ 若 LEN0_PKLENBYTES = '1'， LEN0_PKLENPOLAR = '1'，则以 0x0104 作为自动接收帧长度信息，PRX 将共接收 260 (0x0104) +LEN0_PKLENADD bytes 数据后停止接收。

在直接 FIFO 帧结构下，若发送或接收长度大于 256bytes，需要借助发送半空与接收半满中断配合完成收发流程。

注：在使能接收前，建议软件可以先将 LEN0_PKLEN 设置为较大的初始值，以便留出足够的时间给软件获取 FIFO 内收取数据包携带的长度信息并填入 LEN0_PKLEN 寄存器，防止 FIFO0_INT 的误触发。

12.3 直接收发DIRECT模式

设置 DIRECT_MODE 寄存器，芯片进入直接收发模式。

发送端可通过 IRQ(DIRECT_IE = '1')或 GPIO(GPIOn_FUN = 5'b00010)输出发送同步时钟，通过 GPIO(GPIOn_FUN = 5'b00011)直接灌入发送数据；

接收端可以通过 IRQ (DIRECT_IE = '1') 或 GPIO(GPIOn_FUN = 5'b00100)端口输出接收同步时钟，通过 GPIO(GPIOn_FUN = 5'b00101)端口输出接收数据。

具体 GPIO 的设置可参考第 17 章寄存器设置。

直接发送数据的时序如图 12-6 所示：

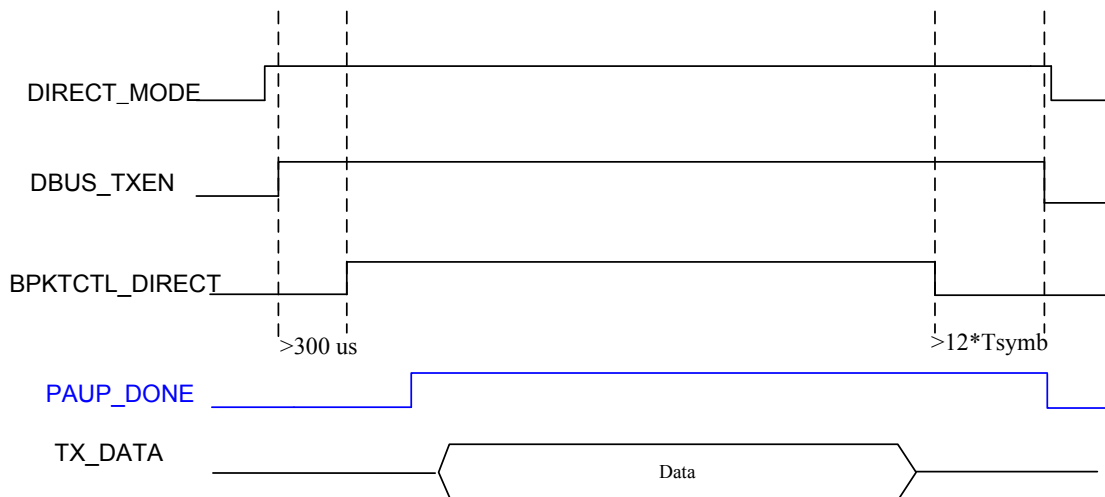


图 12-6 直接发送模式时序图

直接接收数据的时序如图 12-7 所示：

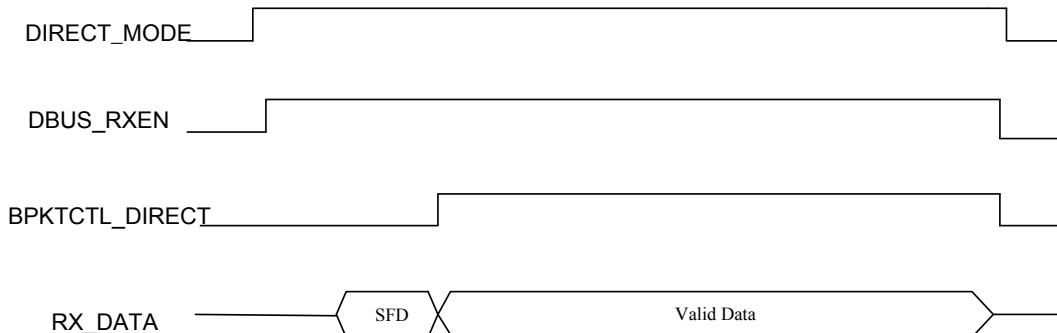


图 12-7 直接接收模式时序图

在直接发送模式下，需要在 DBUS_TXEN 置高 300μs 后置高寄存器 BPKTCTL_DIRECT (0x1C)，以保证前端 PLL 的稳定，等待 PA Ramp 完成后，PAUP_DONE(0x1B)指示位将置为'1'，之后才能发送有效数据。

当 DBUS_RXEN 置高 300μs 左右后，接收机能够正常的接收数据，当发现有效的 SFD 以后，需将 BPKTCTL_DIRECT 置高，调整接收机的环路带宽，提高接收机性能。

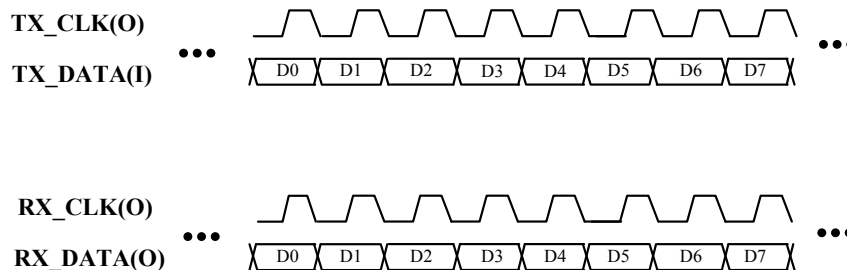


图 12-8 直接收发模式数据时序示意图

直接收发模式下数据与时钟的时序如图 12-8 所示。发送端 MCU 依据芯片输出的 TX_CLK 向芯片灌入需要发送的数据，芯片内部以 TX_CLK 上升沿采样数据；接收端 MCU 可依据芯片输出的 RX_CLK，上升沿采样输出的接收数据 RX_DATA。

12.4 连续发送模式

在直接收发模式的情况下，芯片支持内部数据的连续发送的方式，以方便芯片的测试。

使能 REP_MODE (0x1C)，依据 12.3 节使能直接发送模式，芯片将连续发送“0101”的数据序列。

使能 PN9_MODE (0x1C)，依据 12.3 节使能直接发送模式，芯片将连续发送 PN9 数据序列。

第 13 章 RF收发器—唤醒

13.1 Wake-up Timer唤醒

ES32W0030 支持 32.768kHz 的内部 RC 时钟，用于芯片在 SLEEP(RC)模式下的自动唤醒。

Wake-up Timer 唤醒功能需设置 WDT_EN = '1'，WDT_IE = '1'，自动唤醒间隔可通过寄存器 WDT_WTM (0x1F) 与 WDT_WTR (0x20) 配置。唤醒间隔 (s)：

$$T_{WUT} = \frac{32 \times (WDT_WTR + 1) \times (WDT_WTM + 1)}{32768}$$

Wake-up Timer 计数器溢出时，芯片将中断标志 WDT_INT 置'1'，并从 SLEEP(RC)状态唤醒，可进行正常收发操作，软件清中断标志位 WDT_INT 后芯片再次进入 SLEEP(RC)状态。

若 WDT_IE = '0'，Wake-up Timer 计数器溢出时将不会唤醒芯片，可通过配置 GPIO_n_FUN，将 Wake-up Timer 的计时器溢出标志脉冲通过 GPIO 送出。

13.2 GPIO外部中断唤醒

GPIO 外部中断唤醒功能需设置 EXTn_IE = '1'，支持上升沿、下降沿、双边沿电平唤醒，详细见 GPIO_n_FUN 寄存器配置。

芯片响应外部中断后标志位 EXTn_INT 将置'1'，芯片从 SLEEP(RC)状态被唤醒后可进行正常收发操作，软件清中断标志位 EXTn_INT 后芯片再次进入 SLEEP(RC)状态。

若使能 EXTn_IE，外部中断标志位 EXTn_INT 通过 IRQ 引脚输出。

第 14 章 RF收发器—RC校准

芯片内部 RC 支持软件校准与硬件自动校准两种模式，可由 RCCODE_SEL 寄存器（0x38）配置。

14.1 软件校准

软件校准模式下，调整 WDT_CAL 寄存器（0x38）值可调整内部 RC 时钟频率，具体可参考 WDT_CAL 寄存器说明。

14.2 硬件自动校准

硬件自动校准需在芯片处于 IDLE 模式下完成，配置 RCCAL_STEN 寄存器（0x3A）可支持单次校准与连续校准两种模式。校准完成后 CAL_DONE 标志位（0x39）将置'1'，并可通过 RCCODE 寄存器（0x39）获取 RC 校准配置字。

第 15 章 RF收发器—异常状态的检测及复位

15.1 芯片异常状态检测

寄存器 CHIPSTA1(0x36)提供芯片内部电路的工作状态，如表 15-1 所示：

芯片状态	CHIPSTA1(HEX)
SLEEP/DEEP SLEEP	0x0777
IDLE	0x0077
TX (进入发送状态 100 μ s 之后)	0x008A
RX (进入接收状态 100 μ s 之后)	0x000D

表 15-1 CHIPSTA1 各状态说明

如果在相关状态下，0x36 寄存器的读取值与参考值不符，则说明芯片工作进入异常状态，可根据需要通过 PDN 端口或软件复位对芯片进行复位操作。

15.2 寄存器与FIFO校验功能

芯片提供寄存器与 FIFO 写入数据的校验功能，以防止 SPI 通信过程中受到外界干扰可能造成的误写，校验寄存器为 REGCHECK(0x6F)。

若需要对 SPI 写入值进行校验，软件在进行寄存器写入或 FIFO 写入之前，读取 REGCHECK 寄存器初始值，将之后的写入数据以 byte 为单位（寄存器写入先校验高 byte，再校验低 byte）与 REGCHECK 初始读取值进行异或计算。在数据写入完成之后，通过比较软件计算的校验值与 REGCHECK 当前读取值是否一致，判断 SPI 写入数据是否有误。

15.3 芯片复位

芯片从 POWER DOWN 状态恢复至 IDLE 状态过程中内部 POR 将复位全芯片。

芯片从 DEEP SLEEP/SLEEP 恢复至 IDLE 状态，芯片除寄存器将全部被复位。

芯片共提供两种软复位方式，分别为 SFT_RST0 (0x60) 和 SFT_RST1 (0x61)，其中：

SFT_RST0 复位硬件电路与 FIFO 而保留原有的寄存器设置值；

SFT_RST1 进行全芯片复位，寄存器同时也会被复位成默认值。

第 16 章 RF收发器—其它功能与设置

16.1 白化与CRC

芯片支持数据的白化功能，以增强长‘0’或长‘1’类型数据的传输能力。可以通过寄存器 SCRAMBLE_ON (0x02) 开启白化功能，白化种子可通过寄存器 SCRAMBLE_DATA (0x05) 设置，若 SCRAMBLE_DATA 设置为全 0 则不具备白化功能。

增强型帧结构白化区域为 PHR-PSDU-FCS；直接 FIFO 帧结构白化区域为 FIFO 内的填充数据。

其基本数据流为：

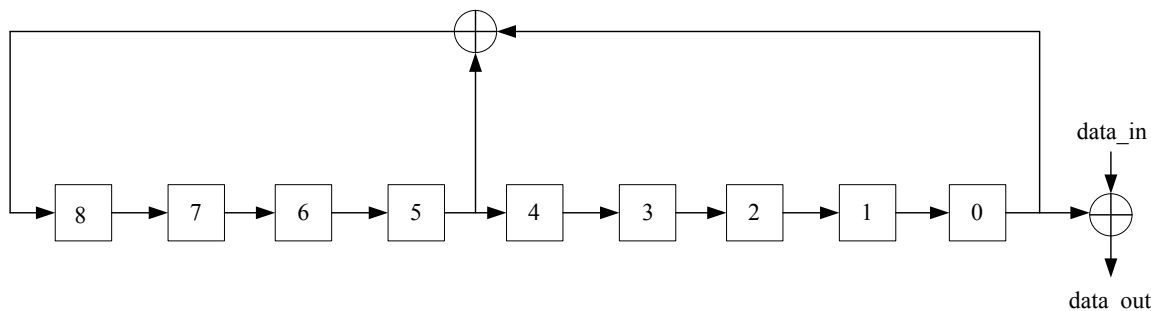


图 16-1 PN9 数据白化生成器

增强型帧结构 CRC 校验范围为 PHR-PSDU。

增强型帧结构支持 CRC 硬件实现与软件实现两种模式，可通过 LEN1_CRCSEL 寄存器配置。若选用软件实现 CRC 功能，发送端需在发送 CRC 之前将 CRC 值填入 CRCVAL 寄存器。

接收端可通过 PRX_CRC_ERR 标志位判断接收数据包 CRC 校验是否正确，也可通过 CRCVAL 寄存器读取接收到的 CRC 由软件自行校验。

硬件 CRC 选用的生成多项式为 $G(X) = 1 + X^5 + X^{12} + X^{16}$ ，其实现的移位反馈寄存器的逻辑如图 16-2 所示，生成后的 CRC 会进行取反操作，然后跟随在数据之后进行发送。移位反馈寄存器初始值可通过寄存器 CRC_INIT_DATA (0x04) 设置。

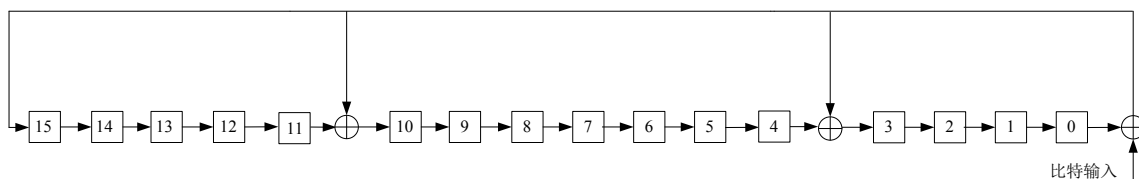


图 16-2 CRC 硬件实现示意图

直接 FIFO 帧结构不支持硬件 CRC 功能。

16.2 FEC

增强型帧结构模式下支持数据自动的 FEC 纠错码功能，码率 R 为 2/3。使用 FEC 可以在一定程度上纠正传输中的传输错误，可以提高系统在低信噪比环境下传输成功率，但 FEC 会增

加冗余数据，增加数据包的传送时间。可以通过配置寄存器 FEC_ON (0x02) 使能 FEC 功能。

注：白化或 FEC 功能需在 PTX 和 PRX 两方同时开启或关闭。

16.3 数据编码

芯片支持 Manchester 与 8bit/10bit line code 两种编码方式，可通过 PACK_TYPE (0x02) 寄存器配置，默认不对数据进行编码。

增强型帧结构编码区域为 PHR-PSDU-FCS；直接 FIFO 帧结构编码区域为 FIFO 内的填充数据。

16.4 RSSI

在接收模式时，芯片会评估天线端接收信号能量的大小，该数值会保存在寄存器 RSSI (0x23) 中。RSSI 的读数单位为 dBm，数据的格式为二进制补码形式的符号数。在 RSSI 寄存器里提供两个 RSSI 读数值，其中 RSSI1 保存的是上一个有效数据包 (SFD 正确同步) 的 RSSI 计算值，而 RSSI2 中保存的是实时的 RSSI 计算值，可用于 CSMA/CA 工作。

16.5 载波检测 (Carrier Detect)

芯片支持载波检测功能，即在接收模式时，芯片会监控带内的接收信号能量，当接收信号能量大于设置的域值后，芯片将给出载波检测指示信号。

载波指示的能量域值可通过寄存器 CDTH (0x27) 设定，当输入信号能量大于 CD_TH1 设置值后，载波检测指示位 CD 会置高，当信号能量降低并小于 CD_TH2 的设置，载波检测指示位 CD 将立刻复位。

CD_TH1 和 CD_TH2 设置值单位为 dBm，数据格式为二进制补码形式的符号数。

建议的 CD_TH1 与 CD_TH2 的设置方法：

1. 在接收灵敏度点读取 RSSI 寄存器值；
2. CD_TH1 的设置值为 RSSI - 2dBm；
3. CD_TH2 的设置值为 RSSI - 5dBm。

CD 信号可通过配置从 GPIO 输出，当 CD 信号置高时，RSSI_INT 中断标志位将置'1'。

16.6 AFC及晶振的校准功能

在接收与发送之间，由于使用晶振的自身因素、温度、电压工作环境等变化的原因，可能造成收发晶振的频率不一致，进而引起收发两端的载波频偏。

芯片在接收端提供载波频偏自动补偿功能 (AFC)，可通过 AFC_EN (0x25) 寄存器使能。

AFC 补偿范围为 $\pm 20\text{kHz}$ ，在 433MHz 的工作频段，补偿范围约为 $\pm 40\text{ppm}$ 。由于该补偿范围包括了收发两端的晶振偏差，则对于单端晶振，其频率变化需控制在 $\pm 20\text{ppm}$ 以内。

晶振校准寄存器为 XOSC_CAL(0x37)，设置值支持 0x00 至 0xFF，步长约 15Hz。

16.7 有效/无效前导检测

芯片提供有效前导码的检测功能。在有效前导检测长度时间内若发现有效前导序列，有效前导检测中断标志 PREDET_INT 将置'1'，有效前导检测长度可以通过寄存器 PRENUM(0x24) 配置。

接收端前导码检测的长度必须小于发送数据的前导序列，考虑到 AGC 及 AFC 功能的时间开销，推荐接收前导检测的长度应比发送前导序列少 8 ~ 16bits。默认的前导检测长度为 32bits。

为方便跳频应用，芯片还提供无效前导的检测功能。在无效前导检测长度时间（详见 0x24 寄存器说明）内若未发现有效前导序列，则无效前导检测中断标志 INVPREDET_INT 将置'1'，无效前导检测长度可配置 INV_PRENUM 寄存器（0x24），默认的 INV_PRENUM 设置长度为 16bits。

通过配置中断使能 PREDET_IE/INVPREDET_IE 或 GPIO 功能寄存器，有效前导检测中断标志 PREDET_INT 及无效前导检测中断标志 INVPREDET_INT 可以通过 IRQ 或任意 GPIO 管脚输出。

16.8 有效/无效SFD检测

芯片支持 1~6bytes 可配置长度的 SFD 地址，长度可通过 SYNCWORD_LEN 寄存器设置，地址可通过寄存器 0x14~0x1A 设置。PRX 在接收到可同步的 SFD 地址后，中断标志 SFDDET_INT 将置'1'。

为方便软件跳频等应用，PRX 在检测到有效前导之后，若在 INVSFD_TIME (0x4F) 设定时间长度内没有检测到有效 SFD，无效 SFD 中断标志 INVSFDDET_INT 将置'1'。

通过配置中断使能 SFDDET_IE/INVSFDDET_IE 或 GPIO 功能寄存器，有效 SFD 中断标志 SFDDET_INT 及无效 SFD 中断标志 INVSFDDET_INT 可以通过 IRQ 或任意 GPIO 管脚输出。

16.9 低功耗接收模式

芯片提供低功耗的接收模式，可通过设置寄存器 LP_ENABLE (0x25) 使能该工作模式。

在低功耗接收模式下，接收功耗约为正常接收功耗的： $\text{HOP_TIMER}/(\text{HOP_TIMER}+\text{LP_TIMER})$

第 17 章 RF收发器—寄存器

17.1 寄存器列表

地址	寄存器名称	功能说明	复位值
0x00	RFID	RF ID 版本号	-
0x01	TRCTRL	收发使能寄存器	0x0000
0x02	PKTCTRL	帧结构选择寄存器	0xC000
0x03	PKTCFG0	帧配置寄存器 0	0x4008
0x04	CRCSEED	CRC 移位寄存器初始值寄存器	0xFFFF
0x05	PKTCFG1	帧配置寄存器 0	0x11FF
0x06	FIFOTHRES	FIFO 阈值寄存器	0x1010
0x07	TXPHR0	增强型帧结构发送 PHR0 寄存器	0x0100
0x08	TXPHR1	增强型帧结构发送 PHR1 寄存器	0x0001
0x09	PSDULEN	增强型帧结构发送 PSDU 长度配置寄存器	0x0100
0x0A	RXPHR0	增强型帧结构接收 PHR 寄存器 0	-
0x0B	RXPHR1	增强型帧结构接收 PHR 寄存器 1	-
0x0C	FIFOCTRL	FIFO 控制寄存器	0x000E
0x0D	LEN0PKLEN	直接 FIFO 帧结构长度配置寄存器	0x0080
0x0E	FIFOSTA	FIFO 状态寄存器	0x0000
0x0F	INT	中断标志位寄存器	-
0x10	INTIE	中断使能寄存器	0x8001
0x11	INTIC	中断标志清'0'寄存器	-
0x12	CRCVAL	CRC 寄存器	0x0000
0x13	PIPECTRL	PIPE 控制寄存器	0x0001
0x14	P0ADDR0	PIPE0 SFD 地址低 16bit	0xF398
0x15	P0ADDR1	PIPE0 SFD 地址中间 16bit	0xF398
0x16	P0ADDR2	PIPE0 SFD 地址高 16bit	0xF398
0x17	P1ADDR0	PIPE1 SFD 地址低 16bit	0xC2C2
0x18	P1ADDR1	PIPE1 SFD 地址中间 16bit	0xC2C2
0x19	P1ADDR2	PIPE1 SFD 地址高 16bit	0xC2C2
0x1A	P23ADDR	PIPE2/PIPE3 SFD 地址低 8bit	0x58A7
0x1B	CHIPSTA0	芯片状态指示寄存器 0	-
0x1C	MODECTRL	芯片模式控制寄存器	0x1046
0x1D	GPIOCFG0	GPIO 配置寄存器 0	0xC0C0
0x1E	GPIOCFG1	GPIO 配置寄存器 1	0xCFDF
0x1F	WDTCFG0	WDT 配置寄存器 0	0x0010
0x20	WDTCFG1	WDT 配置寄存器 1	0x0081
0x21	WDTWDV	WDT 计数值指示寄存器	-
0x22	AGCGAIN	AGC 指示寄存器	-
0x23	RSSI	RSSI 指示寄存器	-

0x24	PREACFG	Preamble 配置寄存器	0x0208
0x25	MODEMCTRL	MODEN 控制寄存器	0x1201
0x26	IFSET	中频设置寄存器	0x2CCD
0x27	CDTH	CD 阈值寄存器	0x8281
0x28	CHCFG0	CHANNEL 配置寄存器 0	0x0000
0x29	CHCFG1	CHANNEL 配置寄存器 1	0x0000
0x2A	CHCFG2	CHANNEL 配置寄存器 2	0x0000
0x2B	CHCFG3	CHANNEL 配置寄存器 3	0x0000
0x2C	DEVIATION	调制频偏设置寄存器	0x003F
0x2D	HOPCFG	跳频配置寄存器	0x1566
0x2E	FILTERBAND	接收滤波器带宽设置寄存器	0x004A
0x2F	FREQCFG0	频点设置寄存器 0	0x325C
0x30	FREQCFG1	频点设置寄存器 1	0x0000
0x31	FREQCFG2	频点设置寄存器 2	0x0000
0x32	SYMRATE0	数据速率配置寄存器 0	0x0051
0x33	SYMRATE1	数据速率配置寄存器 1	0x00EC
0x35	RFCFG	模拟前端配置寄存器	0x3332
0x36	CHIPSTA1	芯片状态指示寄存器 1	-
0x37	XOSCCAL	晶振校准寄存器	0x0080
0x38	RCCFG0	RC 校准配置寄存器 0	0x18C3
0x39	RCSTA	RC 校准状态寄存器	-
0x3A	RCCFG1	RC 校准配置寄存器 1	0x002A
0x3C	HOPSPACE0	跳频频点间隔设置寄存器 0	0x0000
0x3D	HOPSPACE1	跳频频点间隔设置寄存器 0	0x7E08
0x3E	RSSICFG	RSSI 配置寄存器	0x021D
0x3F	HOPCH	跳频频点指示寄存器	-
0x40	PACFG	PA 配置寄存器	0x003F
0x45	ACKCFG	ACK 配置寄存器	0x30FF
0x4D	FIFOPTR	FIFO 指针寄存器	0x0000
0x4E	LEN0RXADD	直接 FIFO 帧结构接收补偿寄存器	0x0003
0x4F	INVSFDTIME	无效 SFD 延时计数寄存器	0x0020
0x60	SFTRST0	软件复位寄存器 0	-
0x61	SFTRST1	软件复位寄存器 1	-
0x6F	REGCHECK	寄存器与 FIFO 写入数据校验寄存器	0x0000
0x70	FIFODATA	FIFO 访问寄存器	-

表 17-1 寄存器列表

17.2 寄存器说明

17.2.1 RFID寄存器

寄存器	RFID				
地址	0x00				
复位值	-				
位名称	位	读写	位说明	1	0
RF_VER_ID	15:0	R	RF ID 版本号	-	

17.2.2 TRCTRL寄存器

寄存器	TRCTRL				
地址	0x01				
复位值	0x0000				
位名称	位	读写	位说明	1	0
-	15:9	-	-	-	
DBUS_TXEN	8	R/W	发送使能 (DBUS_TXEN 与 DBUS_RXEN 不能同时设为'1')	发送使能	发送关闭
DBUS_RXEN	7	R/W	接收使能 (DBUS_TXEN 与 DBUS_RXEN 不能同时设为'1')	接收使能	接收关闭
-	6:0	-	-	-	

注：关闭发送或接收使能后需等待 $3 \cdot T_{\text{symb}}$ 以上的时间间隔后(等待 FSM_IDLE='1'之后)才能再次使能发送或接收。
T_{symb} 为数据速率周期，如数据速率为 10Kbps，则 T_{symb} = 100 μs。

17.2.3 PKTCTRL寄存器

寄存器	PKTCTRL				
地址	0x02				
复位值	0xC000				
位名称	位	读写	位说明	1	0
PACK_LENGTH_EN	15	R/W	芯片帧模式选择位	增强型帧结构	直接 FIFO 帧结构
LEN0_TXMODE	14	R/W	直接 FIFO 帧结构发送完成模式配置	读写指针相等时发送结束	发送长度按 LEN0_PKLEN 设置值发送
LEN0_TXCWMODE	13	R/W	单载波发送模式使能位	使能	不使能
LEN1_CRCSEL	12	R/W	增强型帧结构模式下 CRC 生成配置	CRC 由软件生成 (自动 ACK 使能情况下不支持)	CRC 由硬件生成
-	11:9	R/W	-	-	
LEN0_RXMODE	8	R/W	直接 FIFO 帧结构接收完成模式配置	接收长度按 LEN0_PKLEN 设置值接收	芯片自动识别包长度进行接收 (详见 12.2 章节描述)
-	7:6	-	-	-	
SFD_OPT	5:4	R/W	SFD 同步前提条件选择配置	00:valid preamble 01:valid CD 10/11:NONE	
PACK_TYPE	3:2	R/W	芯片编码方式选择	00/11: NRZ raw data 01: Manchester data type 10: 8bit/10bit line code	
FEC_ON	1	R/W	FEC2/3 使能信号 (只针对增强型帧结构)	使能	不使能
SCRAMBLE_ON	0	R/W	白化使能信号	使能	不使能

17.2.4 PKTCFG0 寄存器

寄存器名称		PKTCFG0				
地址	0x03					
复位值	0x4008					
位名称	位	读写	位说明	1	0	
SYNCWORD_LEN	15:13	R/W	SFD 长度 (SFD 地址设置寄存器为 0x14~0x1A)	000/001: 禁用 010: 2 bytes 011: 3 bytes 100: 4 bytes 101: 5 bytes 110/111: 6 bytes		
PREAMBLE_MODE	12	R/W	发送前导序列 模式选择位	发送序列 10101010	发送序列 01010101	
-	11:10	-	-	-		
PREAMBLE_LEN	9:0	R/W	前导长度 (支持 4~1023bytes)	10'h0~10'h4 : 4 bytes Others: given value used as length		

17.2.5 CRCSEED寄存器

寄存器		CRCSEED				
地址	0x04					
复位值	0xFFFF					
位名称	位	读写	位说明	1	0	
CRC_INIT_DATA	15:0	R/W	CRC 移位寄存器初始值	-		

17.2.6 PKTCFG1 寄存器

寄存器名称		PKTCFG1				
地址	0x05					
复位值	0x11FF					
位名称	位	读写	位说明	1	0	
SYNC_THRES	15:12	R/W	SFD 接收 允许错误个数阈值	建议设为'0'或'1'		
-	11:9	-	-	-		
SCRAMBLE_DATA	8:0	R/W	白化初始值	-		

17.2.7 FIFOTHRES寄存器

寄存器名称		FIFOTHRES				
地址	0x06					
复位值	0x1010					
位名称	位	读写	位说明	1	0	
EMPTY_THRES	15:8	R/W	FIFO0 半空阈值	-	-	
FULL_THRES	7:0	R/W	FIFO0 半满阈值	-	-	

17.2.8 TXPHR0 寄存器

寄存器名称		TXPHR0				
地址	0x07					
复位值	0x0100					
位名称	位	读写	位说明	1	0	
-	15:9	R/W	-	-	-	
AUTO_PHR_VERIFY	8	R/W	增强型帧结构 PHR 校验位模式选择	硬件依据 PHR 前 3 个 bytes 自动计算校验值进行发送	软件填写 PHR 校验值，填写寄存器为 TX_PHR_VERIFY。 (ACK 功能使能时不支持)	
TX_PHR_VERIFY	7:0	R/W	增强型帧结构 软件填写 PHR 校验值寄存器	-	-	

17.2.9 TXPHR1 寄存器

寄存器名称		TXPHR1				
地址	0x08					
复位值	0x0001					
位名称	位	读写	位说明	1	0	
TX_PHR_CHANNEL_INDEX	15:8	R/W	增强型帧结构发送帧信道索引，发送前由软件填写。	-	-	
TX_PHR_STD_IDEF	7:0	R/W	增强型帧结构发送帧标准识别号，发送前由软件填写。	-	ACK 使能模式下，bit7~bit5 为 PID 与 NOACK 指示位，该 3bits 寄存器设置无效，详见表 7-2 说明。	

17. 2. 10 PSDULEN 寄存器

寄存器名称		PSDULEN				
地址	0x09					
复位值	0x0100					
位名称	位	读写	位说明	1	0	
-	15:9	-	-	-		
AUTO_LEN_CALC	8	R/W	增强型帧结构 硬件自动计算所填 FIFO 内 PSDU 长度使能位	硬件自动计 算 PSDU 长 度	软件填写 PSDU 长度, 填 写寄存器为 TXPSDU_LEN	
TXPSDU_LEN	7:0	R/W	增强型帧结构 软件填写 PSDU 长度	最大设置值为 252 bytes, 最终 发送的 PHR 帧长硬件自动加 3, 设置值>252 按 252 处理。		

17. 2. 11 RXPHR0 寄存器

寄存器名称		RXPHR0				
地址	0x0A					
复位值	-					
位名称	位	读写	位说明	1	0	
RX_PHR_PSDU_LEN	15:8	R	增强型帧结构接收帧长 (长度为实际 PSDU 长度加 3)	-		
RX_PHR_CHANNEL_INDEX	7:0	R	增强型帧结构接收信道索引	-		

17. 2. 12 RXPHR1 寄存器

寄存器名称		RXPHR1				
地址	0x0B					
复位值	-					
位名称	位	读写	位说明	1	0	
RX_PHR_STD_IDEF	15:8	R	增强型帧结构接收标准识别号	-		
RX_PHR_VERIFY	7:0	R	增强型帧结构接收 PHR 校验	-		

17. 2. 13 FIFOCTRL寄存器

寄存器名称	FIFOCTRL				
地址	0x0C				
复位值	0x000E				
位名称	位	读写	位说明	1	0
PRX_PHR_ERR	15	R	接收 PHR 校验出错标志位 (只针对增强型帧结构,并且 AUTO_PHR_VERIFY = '1'时有效)	错误	正确
PRX_CRC_ERR	14	R	接收 CRC 校验出错标志位 (只针对增强型帧结构)	错误	正确
PRX_FEC23_ERR	13	R	接收 FEC 校验出错标志位 (只针对增强型帧结构)	错误	正确
PRX_LEN0RXLEN_ERR	12	R	直接 FIFO 帧结构硬件自动识别 帧长信息接收模式下接收 帧长信息错误标志位 (只针对直接 FIFO 帧结构)	错误	正确
PRX_FIFO0_PIPE	11:9	R	PRX FIFO0 中 数据所属的 PIPE	000: PIPE0 001: PIPE1 010: PIPE2 011: PIPE3 Others:无效	
PRX_FIFO0_OCPY	8	R	PRX FIFO0 数据填写 完成信号	PRX FIFO0 被占用	PRX FIFO0 未被占用
-	7:5	-	-	-	
PTX_FIFO0_NOACK	4	R/W	告知 PRX 当前帧无需 ACK (只针对增强型帧结构)	NACK	ACK
PTX_FIFO0_PIPE	3:1	R/W	PTX FIFO0 中 数据所属的 PIPE (发送前由软件填写)	000: PIPE0 001: PIPE1 010: PIPE2 011: PIPE3 Others:无效	
PTX_FIFO0_OCPY	0	W	FIFO 填写完成后, 软件置'1'指示发送 FIFO 有 效, 等待发送。	指示发送 FIFO 有效	无效操作

17.2. 14LEN0PKLEN寄存器

寄存器名称	LEN0PKLEN				
地址	0x0D				
复位值	0x0080				
位名称	位	读写	位说明	1	0
LEN0_PKLEN	15:0	R/W	PTX:直接 FIFO 帧结构发送长度配置 PRX:直接 FIFO 帧结构接收长度配置	最大可支持 65535 bytes	

17.2. 15FIFOSTA寄存器

寄存器名称	FIFOSTA				
地址	0x0E				
复位值	0x0000				
位名称	位	读写	位说明	1	0
RXZERO_LEN	15	R	PRX 接收到长度为 0 的帧指示位	接收到帧长度为 0	接收到帧长度不为 0
FIFO0_EMPTY	14	R	FIFO0 为空指示位	FIFO0 为空	FIFO0 不为空
WR_PTR_OV	13	R	直接 FIFO 帧结构 FIFO 填写溢出 (>256bytes) 标志位, 可由 CLR_WR_PTR 清'0'或 RD_PTR_OV 置'1'时自动清'0'。	溢出	未溢出
RD_PTR_OV	12	R	直接 FIFO 帧结构 FIFO 读取溢出 (>256bytes) 标志位, 可由 CLR_RD_PTR 清'0'或 WR_PTR_OV 置'1'时自动清'0'。	溢出	未溢出
FIFO0_OCPY	11	R	FIFO 被占用指示位	被占用	未被占用
MAXTXLEN	10	R	发送 FIFO 达到最大发送度 (只针对增强型帧结构)	发送 FIFO 达到最大发送度	发送 FIFO 未达到最大发送度
CLR_RD_PTR	9	W	FIFO 读指针清'0'	读指针清'0'	无效操作
CLR_WR_PTR	8	W	FIFO 写指针清'0'	写指针清'0'	无效操作
-	7:5	-	-	-	
LEN0_PKLENPOLAR	4	R/W	直接 FIFO 帧结构若包长信息设置为 2 个 bytes, 高低 byte 配置位。 (LEN0_RXMODE='0'芯片自动识别帧长接收模式下有效)	0: FIFO 的第一个 byte 为包长高 byte, 第二个 byte 为包长低 byte 1: FIFO 的第一个 byte 为包长低 byte, 第二个 byte 为包长高 byte (详细参考 12.2 章节说明)	
	3	R/W	直接 FIFO 帧结构包长信	2 个 bytes	1 个 byte

LEN0_PKLENBYTES			息占用的 byte 数 (LEN0_RXMODE='0'芯片自动识别帧长接收模式下有效)		
LEN0_PKLENPOS	2:0	R/W	直接 FIFO 帧结构包长信息所在 FIFO 的起始位置 (LEN0_RXMODE='0'芯片自动识别帧长接收模式下有效)	000: byte1 001: byte2 ... 111.: byte8	

17. 2. 16INT寄存器

寄存器名称	INT				
地址	0x0F				
复位值	-				
位名称	位	读写	位说明	1	0
POR_INT	15	R	POR 中断标志位 (芯片上电后需将此标志位清'0')	中断	无中断
CHIPRDY_INT	14	R	芯片内部时钟 ready, 可进行收发操作中断标志位 (使用前需软件清'0')	中断 (芯片上电、软件复位或从 SLEEP 退出时钟恢复后置'1')	无中断
SFDDET_INT	13	R	检测到有效的 SFD 中断标志位 (芯片再次切换为接收状态约 250μs 后自动清'0'或由 SFDDET_IC 软件清'0')	中断	无中断
PREDET_INT	12	R	检测到有效的前导中断标志位 (芯片再次切换为接收状态自动清'0'或由 PREDET_IC 软件清'0')	中断	无中断
EXT0_INT	11	R	GPIO0 外部中断标志位	中断	无中断
EXT1_INT	10	R	GPIO1 外部中断标志位	中断	无中断
EXT2_INT	9	R	GPIO2 外部中断标志位	中断	无中断
EXT3_INT	8	R	GPIO3 外部中断标志位	中断	无中断
INVSFDDET_INT	7	R	检测到无效的 SFD 中断标志位 (芯片再次切换为接收状态约 250μs 后自动清'0'或由 INVSFDDET_IC 软件清'0')	中断	无中断
WDT_INT	6	R	WDT 中断标志位	中断	无中断
HALF_FULL_INT	5	R	FIFO 半满中断标志位 (接收端 MCU 读 RX FIFO 至不满足半满条件自动清'0', 详见 6.1.2 章节描述)	中断	无中断
HALF_EMPTY_INT	4	R	FIFO 半空中断标志位 (发送端 MCU 写 TX FIFO 至不满足半空条件自动清'0', 详见 6.1.2 章节描述)	中断	无中断
RSSI_INT	3	R	RSSI 中断标志位 (芯片再次切换为接收状态约 250μs 后)	中断	无中断

			自动清'0'或由 RSSI_IC 软件清'0')		
INVPREDET_INT	2	R	无效的前导中断标志位 (芯片再次切换为接收状态约 250μs 后 自动清'0'或由 INVPREDET_IC 软件 清'0')	中断	无中断
FIFO0_MAX_RETX	1	R	PTX 重传超时标志位 (只作为标志位使用, 非中断, 只针对增强型帧结构)	重传超时	未超时
FIFO0_INT	0	R	FIFO 中断标志位(具体详见 10.2 章节) PTX :发送完成或发送超时 PRX :接收完成或发送 ACK 完成	中断	无中断

17.2. 17INTIE寄存器

寄存器名称		INTIE				
地址		0x10				
复位值		0x8001				
位名称	位	读写	位说明	1	0	
POR_IE	15	R/W	POR 中断使能	从 IRQ 输出	不输出	
CHIPRDY_IE	14	R/W	CHIPRDY_INT 中断使能	从 IRQ 输出	不输出	
SFDDDET_IE	13	R/W	SFDDDET_INT 中断使能	从 IRQ 输出	不输出	
PREDET_IE	12	R/W	PREDET_INT 中断使能	从 IRQ 输出	不输出	
EXT0_IE	11	R/W	GPIO0 外部中断使能	从 IRQ 输出	不输出	
EXT1_IE	10	R/W	GPIO1 外部中断使能	从 IRQ 输出	不输出	
EXT2_IE	9	R/W	GPIO2 外部中断使能	从 IRQ 输出	不输出	
EXT3_IE	8	R/W	GPIO3 外部中断使能	从 IRQ 输出	不输出	
INVSFDDDET_IE	7	R/W	INVSFDDDET_INT 中断使能	从 IRQ 输出	不输出	
WDT_IE	6	R/W	WDT_INT 中断使能	从 IRQ 输出	不输出	
HALF_FULL_IE	5	R/W	半满中断使能	从 IRQ 输出	不输出	
HALF_EMPTY_IE	4	R/W	半空中断使能	从 IRQ 输出	不输出	
RSSI_IE	3	R/W	RSSI_INT 中断使能	从 IRQ 输出	不输出	
INVPREDET_IE	2	R/W	INVPREDET_INT 中断使能	从 IRQ 输出	不输出	
DIRECT_IE	1	R/W	DIRECT 模式 时钟输出中断使能 (发送端为数据发送时钟, 接 收端为数据接收时钟)	从 IRQ 输出	不输出	
FIFO0INT_IE	0	R/W	FIFO0_INT 中断使能	从 IRQ 输出	不输出	

17. 2. 18INTIC寄存器

寄存器名称	INTIC				
地址	0x11				
复位值	-				
位名称	位	读写	位说明	1	0
POR_IC	15	W	POR 中断清'0'信号	清'0'	无效操作
CHIPRDY_IC	14	W	CHIPRDY_INT 中断清'0'信号	清'0'	无效操作
SFDDDET_IC	13	W	SFDDDET_INT 中断清'0'信号	清'0'	无效操作
PREDET_IC	12	W	PREDET_INT 中断清'0'信号	清'0'	无效操作
EXT_IC	11	W	外部中断清'0'信号	清'0'	无效操作
-	10:8	-	-	-	
INVSFDDDET_IC	7	W	INVSFDDDET 中断清'0'信号	清'0'	无效操作
WDT_IC	6	W	WDT_INT 中断清'0'信号	清'0'	无效操作
-	5:4	-	-	-	
RSSI_IC	3	W	RSSI_INT 中断清'0'信号	清'0'	无效操作
INVPREDET_IC	2	W	INVPREDET_INT 中断清'0'信号	清'0'	无效操作
-	1	-	-	-	-
FIFO0INT_IC	0	W	FIFO0_INT 中断清'0'信号	清'0'	无效操作

17. 2. 19CRCVAL寄存器

寄存器名称	CRCVAL				
地址	0x12				
复位值	0x0000				
位名称	位	读写	位说明	1	0
CRCVAL	15:0	W	CRC 数值寄存器 (只针对增强型帧结构)	软件填写的发送 CRC，高 bit 在前，低 bit 在后。	

17. 2. 20 PIPECTRL 寄存器

寄存器名称		PIPECTRL				
地址		0x13				
复位值		0x0001				
位名称	位	读写	位说明	1	0	
-	15:8	-	-	-		
P3_ACKEN	7	R/W	PIPE3 自动 ACK 使能信号	使能	不使能	
P2_ACKEN	6	R/W	PIPE2 自动 ACK 使能信号	使能	不使能	
P1_ACKEN	5	R/W	PIPE1 自动 ACK 使能信号	使能	不使能	
P0_ACKEN	4	R/W	PIPE0 自动 ACK 使能信号	使能	不使能	
P3_EN	3	R/W	PIPE3 使能信号	使能	不使能	
P2_EN	2	R/W	PIPE2 使能信号	使能	不使能	
P1_EN	1	R/W	PIPE1 使能信号	使能	不使能	
P0_EN	0	R/W	PIPE0 使能信号	使能	不使能	

17. 2. 21 P0ADDR0 寄存器

寄存器名称		P0ADDR0				
地址		0x14				
复位值		0xF398				
位名称	位	读写	位说明	1	0	
P0_ADDR[15:0]	15:0	R/W	PIPE0 SFD 地址 低 16 位	-		

17. 2. 22 P0ADDR1 寄存器

寄存器名称		P0ADDR1				
地址		0x15				
复位值		0xF398				
位名称	位	读写	位说明	1	0	
P0_ADDR[31:16]	15:0	R/W	PIPE0 SFD 地址 中间 16 位	-		

17. 2. 23 P0ADDR2 寄存器

寄存器名称		P0ADDR2				
地址		0x16				
复位值		0xF398				
位名称	位	读写	位说明	1	0	
P0_ADDR[47:32]	15:0	R/W	PIPE0 SFD 地址 高 16 位	-		

17. 2. 24P1ADDR0 寄存器

寄存器名称	P1ADDR0				
地址	0x17				
复位值	0xC2C2				
位名称	位	读写	位说明	1	0
P1_ADDR[15:0]	15:0	R/W	PIPE1 SFD 地址 低 16 位	-	

17. 2. 25P1ADDR1 寄存器

寄存器名称	P1ADDR1				
地址	0x18				
复位值	0xC2C2				
位名称	位	读写	位说明	1	0
P1_ADDR[31:16]	15:0	R/W	PIPE1 SFD 地址 中间 16 位	-	

17. 2. 26P1ADDR2 寄存器

寄存器名称	P1ADDR2				
地址	0x19				
复位值	0xC2C2				
位名称	位	读写	位说明	1	0
P1_ADDR[47:32]	15:0	R/W	PIPE1 SFD 地址 高 16 位	-	

17. 2. 27P23ADDR 寄存器

寄存器名称	P23ADDR				
地址	0x1A				
复位值	0x58A7				
位名称	位	读写	位说明	1	0
P2_ADDR[7:0]	15:8	R/W	PIPE2 SFD 地址低 8 位 PIPE2 SFD 完整地址 { P1_ADDR[47:8],P2_ADDR[7:0]}	-	
P3_ADDR[7:0]	7:0	R/W	PIPE3 SFD 地址低 8 位 PIPE3 SFD 完整地址 { P1_ADDR[47:8],P3_ADDR[7:0]}	-	

17. 2. 28CHIPSTA0 寄存器

寄存器名称	CHIPSTA0				
地址	0x1B				
复位值	-				
位名称	位	读写	位说明	1	0
FSM_IDLE	15	R	芯片 IDLE 状态指示位	IDLE 状态	非 IDLE 状态
-	14:13	-	-	-	-
FSM_SLEEP	12	R	芯片 SLEEP 状态指示位	SLEEP 状态	非 SLEEP 状态
-	11:7	-	-	-	-
PADN_DONE	6	R	PA ramp down 完成指示位	完成	未完成
PAUP_DONE	5	R	PA ramp up 完成指示位	完成	未完成
LOCK_DET	4	R	PLL 锁定指示位	完成	未完成
-	3:0	R	-	-	-

17. 2. 29MODECTRL 寄存器

寄存器名称	MODECTRL				
地址	0x1C				
复位值	0x1046				
位名称	位	读写	位说明	1	0
SLEEP_ON	15	R/W	芯片进入 SLEEP 模式控制信号	进入 SLEEP 模式	退出 SLEEP 模式
-	14	-	-	-	-
LPFEN	13	R/W	低功耗输出模式 控制信号	芯片进入 SLEEP 模式后 GPIO 自动切换为 RC 时钟输出 (GPIO _n _FUN 需配置为 5'b01111)	芯片进入 SLEEP 模式后 GPIO 不输出时钟 (除 CLKCFG 配置为 4'b0111 情况)
PKT_HINT_PRIORITY	12	R/W	IRQ 中断极性 选择配置	由"1"变为"0"	由"0"变为"1"
REP_MODE	11	R/W	直接发送模式 发送 "0101" 序列	使能	不使能
PN9_MODE	10	R/W	直接发送模式 发送 PN9 序列	使能	不使能
BPKTCTL_DIRECT	9	R/W	直接收发模式 控制信号 (详见 8.3 章节)	使能	不使能
DIRECT_MODE	8	R/W	Direct 模式	使能	不使能

			使能信号		
GPIO0_PORITY	7	R/W	GPIO0 有效电平配置	输出反向	输出不反向
GPIO1_PORITY	6	R/W	GPIO1 有效电平配置	输出反向	输出不反向
GPIO2_PORITY	5	R/W	GPIO2 有效电平配置	输出反向	输出不反向
GPIO3_PORITY	4	R/W	GPIO3 有效电平配置	输出反向	输出不反向
CLKCFG	3:0	R/W	GPIO 时钟输出频率配置	x000: fxtal 0001: fxtal /2 0010: fxtal /4 0011: fxtal /8 0100: fxtal /16 0101: fxtal /32 0110: fxtal /64 0111: frc (WDT_EN= '1'有效) others:不输出 fxtal 为主晶振时钟频率, frc 为 RC 时钟频率	

17. 2. 30 GPIOCFG0 寄存器

寄存器名称	GPIOCFG0				
地址	0x1D				
复位值	0xC0C0				
位名称	位	读写	位说明	1	0
GPIO0_DS	15	R/W	gpio0 驱动能力选择位	大电流驱动模式	正常模式
GPIO0_IE	14	R/W	gpio0 输入使能	使能	不使能
GPIO0_PE	13	R/W	gpio0 上拉使能	上拉	不上拉
GPIO0_FUN	12:8	R/W	gpio0 功能选择位	00000: POR(O) 00001: Wake-Up Timer :1 when WUT has expired (O) 00010: TX clock output (O) 00011: Direct Mode TX data input (I) 00100: RX clock output (O) 00101: RX data(O) 00110: TX state(O) 00111: RX state(O) 01000: TX FIFO almost empty(O) 01001: RX FIFO almost full(O) 01010: SFD detected(O) 01011: CD(O)	

				01100: Valid preamble detected(O) 01101: Invalid SFD detected(O) 01110: Invalid preamble detected(O) 01111: chip clock output, configed by clkcfg (O) 10000: External Interrupt, falling edge(I) 10001: External Interrupt, rising edge(I) 10010: External Interrupt, state change (I) 10011: Analog Output (O) 10100; TX data(O) <10101~10111>: VSS <11000~11011>: VDD 11100~11110: reserved(O) 11111: reserved(I)	
GPIO1_DS	7	R/W	gpio1 驱动能力选择位	大电流驱动模式	正常模式
GPIO1_IE	6	R/W	gpio1 输入使能	使能	不使能
GPIO1_PE	5	R/W	gpio1 上拉使能	上拉	不上拉
GPIO1_FUN	4:0	R/W	gpio1 功能选择位	00000: POR(O) 00001: Wake-Up Timer :1 when WUT has expired (O) 00010: TX clock output (O) 00011: Direct Mode TX data input (I) 00100: RX clock output (O) 00101: RX data(O) 00110: TX state(O) 00111: RX state(O) 01000: TX FIFO almost empty(O) 01001: RX FIFO almost full(O) 01010: SFD detected(O) 01011: CD(O) 01100: Valid preamble detected(O) 01101: Invalid SFD detected(O) 01110: Invalid preamble detected(O) 01111: chip clock output, configed by clkcfg (O) 10000: External Interrupt, falling edge(I) 10001: External Interrupt, rising edge(I) 10010: External Interrupt, state change (I) 10011: Analog Output (O) 10100; TX data(O) <10101~10111>: VSS <11000~11011>: VDD 11100~11110: reserved(O) 11111: reserved(I)	

17. 2. 31 GPIOCFG1 寄存器

寄存器名称	GPIOCFG1				
地址	0x1E				
复位值	0xCFDF				
位名称	位	读写	位说明	1	0
GPIO2_DS	15	R/W	gpio2 驱动能力选择位	大电流驱动模式	正常模式
GPIO2_IE	14	R/W	gpio2 输入使能	使能	不使能
GPIO2_PE	13	R/W	gpio2 上拉使能	上拉	不上拉
GPIO2_FUN	12:8	R/W	gpio2 功能选择位	00000: POR(O) 00001: Wake-Up Timer :1 when WUT has expired (O) 00010: TX clock output (O) 00011: Direct Mode TX data input (I) 00100: RX clock output (O) 00101: RX data(O) 00110: TX state(O) 00111: RX state(O) 01000: TX FIFO almost empty(O) 01001: RX FIFO almost full(O) 01010: SFD detected(O) 01011: CD(O) 01100: Valid preamble detected(O) 01101: Invalid SFD detected(O) 01110: Invalid preamble detected(O) 01111: chip clock output, configed by clkcfg (O) 10000: External Interrupt, falling edge(I) 10001: External Interrupt, rising edge(I) 10010: External Interrupt, state change (I) 10011: Analog Output (O) 10100: TX data(O) <10101~10111>: VSS <11000~11011>: VDD 11100~11110: reserved(O) 11111: reserved(I)	
GPIO3_DS	7	R/W	gpio3 驱动能力选择位	大电流驱动模式	正常模式
GPIO3_IE	6	R/W	gpio3 输入使能	使能	不使能
GPIO3_PE	5	R/W	gpio3 上拉使能	上拉	不上拉
GPIO3_FUN	4:0	R/W	gpio3 功能选择位	00000: POR(O) 00001: Wake-Up Timer :1 when WUT has expired (O) 00010: TX clock output (O) 00011: Direct Mode TX data input (I) 00100: RX clock output (O)	

				00101: RX data(O) 00110: TX state(O) 00111: RX state(O) 01000: TX FIFO almost empty(O) 01001: RX FIFO almost full(O) 01010: SFD detected(O) 01011: CD(O) 01100: Valid preamble detected(O) 01101: Invalid SFD detected(O) 01110: Invalid preamble detected(O) 01111: chip clock output, configed by clkcfg (O) 10000: External Interrupt, falling edge(I) 10001: External Interrupt, rising edge(I) 10010: External Interrupt, state change (I) 10011: Analog Output (O) 10100: TX data(O) <10101~10111>: VSS <11000~11011>: VDD 11100~11110: reserved(O) 11111: reserved(I)
--	--	--	--	---

17. 2. 32WDTCFG0 寄存器

寄存器名称		WDTCFG0			
地址	0x1F				
复位值	0x0010				
位名称	位	读写	位说明	1	0
WDT_WTM	15:0	R/W	Wake-up Timer wtm value	-	-

17. 2. 33WDTCFG1 寄存器

寄存器名称		WDTCFG1			
地址	0x20				
复位值	0x0081				
位名称	位	读写	位说明	1	0
-	15:8	-	-	-	-
WDT_EN	7	R/W	WDT 使能信号	使能	不使能
-	6:4	-	-	-	-
WDT_WTR	3:0	R/W	Wake-up Timer wtr value	-	-

17. 2. 34WDTWDV寄存器

寄存器名称		WDTWDV			
地址		0x21			
复位值		-			
位名称	位	读写	位说明	1	0
WDT_WDV	15:0	R	Wake-up Timer 当前计数值	-	

17. 2. 35AGCGAIN寄存器

寄存器名称		AGCGAIN			
地址		0x22			
复位值		-			
位名称	位	读写	位说明	1	0
LAGC_GAIN	15:8	R	AGC 增益档	-	
-	7:0	-	-	-	

17. 2. 36RSSI寄存器

寄存器名称		RSSI			
地址		0x23			
复位值		-			
位名称	位	读写	位说明	1	0
RSSI2	15:8	R	实时 RSSI 值	二进制补码表示	
RSSI1	7:0	R	上一个有效数据包的 RSSI 值	二进制补码表示	

17. 2. 37PREACFG寄存器

寄存器名称		PREACFG			
地址		0x24			
复位值		0x0208			
位名称	位	读写	位说明	1	0
-	15:11	-	-	-	
INV_PRENUM	10:8	R/W	检测无效前导长度 设置寄存器	检测长度为 INV_PRENUM * 8 bits (最终检测无效前导长度为 INV_PRENUM * 8 + PRENUM * 4 bits)	
PRENUM	7:0	R/W	检测有效前导长度 设置寄存器	检测长度为 PRENUM * 4 bits 其中 1bits 长度代表 1 个数据速率符 号长度。	

17. 2. 38MODEMCTRL寄存器

寄存器名称		MODEMCTRL			
地址	0x25				
复位值	0x1201				
位名称	位	读写	位说明	1	0
-	15	-	-	-	
CFG_XTCS	14	R/W	外部晶振选择	20MHz 晶振	26MHz 晶振
BT_MODE	13	R/W	BT 参数设置	1	0.5
GFSK_MODE	12	R/W	GFSK 模式使能信号	使能	不使能
LP_TIMER	11:8	R/W	低功耗接收模式 时间配置	T= LP_TIMER * 16Tsymb	
-	7:3	R/W	-	-	
LP_ENABLE	2	R/W	低功耗接收模式 使能信号	使能	不使能
HOP_SEL	1	R/W	跳频有效信号 选择配置位	有效能量	有效前导
AFC_EN	0	R/W	AFC 功能使能信号	使能	不使能

17. 2. 39IFSET寄存器

寄存器名称		IFSET			
地址	0x26				
复位值	0x2CCD				
位名称	位	读写	位说明	1	0
MIXFW	15:0	R/W	中频设置寄存器	按《AN1045_应用笔记_HW3000 应用注意事项》寄存器配置章节推荐值进行配置	

17. 2. 40CDTH寄存器

寄存器名称		CDTH			
地址	0x27				
复位值	0x8281				
位名称	位	读写	位说明	1	0
CD_TH1	15:8	R/W	载波检测阈值	具体详见 10.2 和 16.5 章节相关说明	
CD_TH2	7:0	R/W	载波检测阈值		

17. 2. 41 CHCFG0 寄存器

寄存器名称		CHCFG0			
地址		0x28			
复位值		0x0000			
位名称	位	读写	位说明	1	0
CHANNEL1	15:8	R/W	主信道号	-	-
CHANNEL2	7:0	R/W	从信道号 1	-	-

17. 2. 42 CHCFG1 寄存器

寄存器名称		CHCFG1			
地址		0x29			
复位值		0x0000			
位名称	位	读写	位说明	1	0
CHANNEL3	15:8	R/W	从信道号 2	-	-
CHANNEL4	7:0	R/W	从信道号 3	-	-

17. 2. 43 CHCFG2 寄存器

寄存器名称		CHCFG2			
地址		0x2A			
复位值		0x0000			
位名称	位	读写	位说明	1	0
CHANNEL5	15:8	R/W	从信道号 4	-	-
CHANNEL6	7:0	R/W	从信道号 5	-	-

17. 2. 44 CHCFG3 寄存器

寄存器名称		CHCFG3			
地址		0x2B			
复位值		0x0000			
位名称	位	读写	位说明	1	0
CHANNEL7	15:8	R/W	从信道号 6	-	-
CHANNEL8	7:0	R/W	从信道号 7	-	-

17. 2. 45 DEVIATION寄存器

寄存器名称	DEVIATION				
地址	0x2C				
复位值	0x003F				
位名称	位	读写	位说明	1	0
-	15:10	R/W	-	-	
DEVIATION	9:0	R/W	调制频偏设置	具体详见 7.2.2 章节	

17. 2. 46 HOPCFG寄存器

寄存器名称	HOPCFG				
地址	0x2D				
复位值	0x1566				
位名称	位	读写	位说明	1	0
SFD_TIMER	15:8	R/W	跳频检测 SFD 时间控制 (发现有效 preamble 后, 若在设定时间内无有效 SFD, 将自动跳至下一个频点。)	T= SFD_TIMER * 32Tsymb	
HOP_TIMER	7:4	R/W	跳频时间控制	T= HOP_TIMER * 16Tsymb	
HOP_NUM	3:1	R/W	扫描的信道数目	信道数=HOP_NUM+1	
HOP_ENABLE	0	R/W	跳频接收功能使能信号	使能	不使能

17. 2. 47 filterband寄存器

寄存器名称	FILTERBAND				
地址	0x2E				
复位值	0x004A				
位名称	位	读写	位说明	1	0
-	15:7	-	-	-	
FILTER2_BAND	6:0	R/W	接收滤波器带宽设置	具体详见 11.2.4 章节	

17. 2. 48FREQCFG0 寄存器

寄存器名称		FREQCFG0				
地址		0x2F				
复位值		0x325C				
位名称	位	读写	位说明	1	0	
-	15	-	-	-	-	
CH_SPACE	14:13	R/W	信道间隔设置	00:100kHz 01: 200kHz 10: 400kHz 11: 800kHz		
RF_FREQ_BASE	12:0	R/W	起始频点设置/ 直接频点设置模式下, 整数部分值	-		

17. 2. 49FREQCFG1 寄存器

寄存器名称		FREQCFG1				
地址		0x30				
复位值		0x0000				
位名称	位	读写	位说明	1	0	
RF_PLL_DIRECT	15	R/W	直接设频点模式使能	使能需先设置RF_FREQ_FRACTION再设置RF_FREQ_BASE。	不使能	
-	14:5	-	-	-	-	
RF_FREQ_FRACTION[20:16]	4:0	R/W	直接设频点模式频点小数部分 bit20~bit16	具体详见 11.1.2 章节		

17. 2. 50FREQCFG2 寄存器

寄存器名称		FREQCFG2				
地址		0x31				
复位值		0x0000				
位名称	位	读写	位说明	1	0	
RF_FREQ_FRACTION[15:0]	15:0	R/W	直接设频点模式频点小数部分 bit15~bit0	具体详见 11.1.2 章节		

17. 2. 51 SYMRATE0 寄存器

寄存器名称		SYMRATE0				
地址		0x32				
复位值		0x0051				
位名称	位	读写	位说明	1	0	
-	15:10	-	-	-		
SYMBOL_RATE[17:8]	9:0	R/W	数据速率设置寄存器	具体详见 11.2.1 章节		

17. 2. 52 SYMRATE1 寄存器

寄存器名称		SYMRATE1				
地址		0x33				
复位值		0x00EC				
位名称	位	读写	位说明	1	0	
-	15:8	-	-	-		
SYMBOL_RATE[7:0]	7:0	R/W	数据速率设置寄存器	具体详见 11.2.1 章节		

17. 2. 53 RFCFG 寄存器

寄存器名称		RFCFG				
地址		0x35				
复位值		0x3332				
位名称	位	读写	位说明	1	0	
-	15	-	-	-		
PLL_TRXLB_SEL	14:13	R/W	VCO 输出频率分频数控制	详见《AN1045_应用笔记_HW3000 应用注意事项》3.3 章节相关说明		
VCO_HB_SEL	12	R/W	高低频 VCO 选择控制位	High freq VCO	Low freq VCO	
-	11	-	-	-		
PLL_LPFR	10:8	R/W	PLL 环路带宽控制位	000: BW50K 001: BW75K 011: BW100K 111: BW150K		
PLL_CPI	7:4	R/W	电荷泵电流控制位	0001: 12.5μA 0010: 25μA 0100: 50μA 1000: 100μA 0011: 37.5μA		
TRXBUF_CUR	3	R/W	TX/RX BUFFER 电流	大电流模式	正常模式	

			控制位		
TRXBUF_LOAD	2	R/W	TX/RX BUFFER 负载 驱动能力控制 位	low load	high load
PLL_TRXBUFI	1:0	R/W	TX/RX 分频器 中 BUFFER 的 电流设置	00: 6μA 01: 8μA 10: 10μA (默认) 11: 12μA	

17. 2. 54CHIPSTA1 寄存器

寄存器名称	CHIPSTA1				
地址	0x36				
复位值	-				
位名称	位	读写	位说明	1	0
CHIPSTA1	15:0	R/W	芯片状态指示寄存器	SLEEP: 0x0777 IDLE: 0x0077 TX: 0x008A (使能发送 150μs 后) RX: 0x000D (使能接收 150μs 后) Others: 异常状态	

17. 2. 55XOSC寄存器

寄存器名称	XOSCCAL				
地址	0x37				
复位值	0x0080				
位名称	位	读写	位说明	1	0
-	15:8	-	-	-	
XOSC_CAL	7:0	R/W	晶振校准寄存器	0xFF: min freq 0x00: max freq	

17. 2. 56RCCFG0 寄存器

寄存器名称		RCCFG0			
地址		0x38			
复位值		0x18C3			
位名称	位	读写	位说明	1	0
RCCODE_SEL	15	R/W	RC 模块校准配置值选择位	软件设置	自动校准
-	14	-	-	-	
WDT_IBIAS	13:12	R/W	32KHz 振荡器电流控制	0x11: max 0x00: min	
WDT_CAL	11:0	R/W	RC 软件校准配置值	0xFFF: max freq 0x000: min freq	

17. 2. 57RCSTA寄存器

寄存器名称		RCSTA			
地址		0x39			
复位值		-			
位名称	位	读写	位说明	1	0
CAL_DONE	15	R	校准完成标志位	校准完成后置'1'。 连续校准模式下每间隔约 30s 开始下一次校准前自动清'0'；非连续校准模式下,使能 RCCAL_EN 自动清'0'	
-	14:12	-	-	-	
RCCODE	11:0	R	自动校准 RC 频率控制字	-	

17. 2. 58RCCFG1 寄存器

寄存器名称		RCCFG1				
地址		0x3A				
复位值		0x002A				
位名称	位	读写	位说明	1	0	
-	15:8	-	-	-		
RCCAL_STEN	7	R/W	连续校准模式使能位 (每约 30s 进行一次校准)	使能	不使能	
RCST_DLY	6:4	R/W	RC 在新的配置值下 再次校准需要等待的时间	3'b000: 0 个 rcclk 周期; 3'b111: 7 个 rcclk 周期;		
RCCAL_ACC	3:1	R/W	频率控制字有效位选择	3'b000: 忽略控制字的低 5 位; 3'b001: 忽略控制字的低 4 位; 3'b010: 忽略控制字的低 3 位; 3'b011: 忽略控制字的低 2 位; 3'b100: 忽略控制字的低 1 位; 3'b101: 使用完整的控制字; 其他: 保留;		
RCCAL_EN	0	W	RC 校准使能位	使能	无效操作	

17. 2. 59HOPSPACE0 寄存器

寄存器名称		HOPSPACE0				
地址		0x3C				
复位值		0x0000				
位名称	位	读写	位说明	1	0	
-	-	-	-	-		
HOP_SPACE[19:16]	3:0	R/W	跳频频点间隔设置寄存器 0 (RF_PLL_DIRECT = '1'时有效)	HOP_SPACE[19:0]=round((fh* k _{freq} /fosc)*2 ²⁰), 其中 fh 为需要的扫描频率间隔, k _{freq} 取值详见数据手册 11.1.3 介绍, fosc 为晶振频率, 默认设置 fh 为 200kHz, fosc 为 26MHz		

17. 2. 60 HOPSPACE1 寄存器

寄存器名称	HOPSPACE1				
地址	0x3D				
复位值	0x7E08				
位名称	位	读写	位说明	1	0
HOP_SPACE[15:0]	15:0	R/W	跳频扫频间隔设置 寄存器 1 (RF_PLL_DIRECT = '1' 时有效)	HOP_SPACE[19:0]=round((fh* k _{freq} /fosc)*2^20), 其中 fh 为需 要的扫描频率间隔, k _{freq} 取值详 见数据手册 11.1.3 介绍, fosc 为晶振频率, 默认设置 fh 为 200kHz, fosc 为 26MHz	

17. 2. 61 RSSICFG 寄存器

寄存器名称	RSSICFG				
地址	0x3E				
复位值	0x021D				
位名称	位	读写	位说明	1	0
-	15:11	-	-	-	
RSSI2_LEN	10:8	R/W	RSSI 的计算长度	000: 128μs 001: 256μs 010: 512μs 011/100: reserved Others: 4096μs	
RSSI2_OFFSET	7:0	R/W	RSSI 的偏差补偿	-	

17. 2. 62 HOPCH 寄存器

寄存器名称	HOPCH				
地址	0x3F				
复位值	-				
位名称	位	读写	位说明	1	0
HOP_CH	15:8	R	跳频所在当前 channel (寄存器指示跳频所在当前 channel 寄存器的设定值, 接 收完成后返回主信道)	-	
LOCK_CH	7:0	R	跳频锁定的 channel (寄存器指示跳频锁定 channel 寄存器的设定值)	可在检测到 有效前导之后读取	

17. 2. 63PACFG寄存器

寄存器名称		PACFG				
地址		0x40				
复位值		0x003F				
位名称	位	读写	位说明	1	0	
PA_VB2	15:12	R/W	输出功率设置寄存器 2	-	-	
PA_VB1	11:8	R/W	输出功率设置寄存器 1	-	-	
-	7:6	-	-	-	-	
PA_SET	5:0	R/W	PA ramp 档位设置	设置值支持 0x01~0x3F		

17. 2. 64ACKCFG寄存器

寄存器名称		ACKCFG				
地址		0x45				
复位值		0x30FF				
位名称	位	读写	位说明	1	0	
RE_TX_TIMES	15:12	R/W	最大重传次数 (只针对增强型帧结构)	设置值只支持 0x1~0xE		
AUTO_RXACK_TIME	11:0	R/W	接收 ACK 等待时间 (只针对增强型帧结构)	$T = \text{AUTO_RXACK_TIME} * T_{\text{symb}}$ 建议设置值为： (PREAMBLE (bytes) + SFD (bytes)) * 8 + 25		

17. 2. 65FIFOPTR寄存器

寄存器名称		FIFOPTR				
地址		0x4D				
复位值		0x0000				
位名称	位	读写	位说明	1	0	
FIFO0_WR_PTR	15:8	R	FIFO 写指针	-	-	
FIFO0_RD_PTR	7:0	R	FIFO 读指针	-	-	

17. 2. 66LEN0RXADD寄存器

寄存器名称	LEN0RXADD				
地址	0x4E				
复位值	0x0003				
位名称	位	读写	位说明	1	0
-	15:8	-	-	-	-
LEN0_PKLENADD	7:0	R/W	直接 FIFO 帧结构 芯片额外接收的 byte 数 (LEN0_RXMODE='0'芯片自动 识别帧长接收模式下有效)	8'h0:不额外接收 8'h1:1 个 byte 8'h2:2 个 bytes ...	

17. 2. 67INVSFDTIME寄存器

寄存器名称	INVSFDTIME				
地址	0x4F				
复位值	0x0020				
位名称	位	读写	位说明	1	0
-	15:8	-	-	-	-
INVSFD_TIME	7:0	R/W	发现有效前导后若在 INVSFD_TIME 设定长 度内未检测到有效 SFD, 无效 SFD 中断 标志 INVSFDDET_INT 将置'1'。	设定长度为 INVSFD_TIME * 8 bits, 其中 1bits 长度代表 1 个数据速率符 号长度。 可依据通讯使用的前导、SFD 长度与 接收设置的前导检测时间综合考虑。 (该寄存器适用于软件跳频等应用)	

17. 2. 68SFTRST0 寄存器

寄存器名称	SFTRST0				
地址	0x60				
复位值	-				
位名称	位	读写	位说明	1	0
SFTRST0	15:0	R/W	软件复位方式 0 (不可复位寄存器)	16'h55AA: 软件复位 Others: 无效操作	

17. 2. 69SFTRST1 寄存器

寄存器名称	SFTRST1				
地址	0x61				
复位值	-				
位名称	位	读写	位说明	1	0
SFTRST1	15:0	RW	软件复位方式 1 (全芯片复位)	16'h55AA: 软件复位 Others: 无效操作	

17. 2. 70REGCHECK寄存器

寄存器名称	REGCHECK				
地址	0x6F				
复位值	0x0000				
位名称	位	读写	位说明	1	0
-	15:8	-	-	-	-
REGCHECK	7:0	R	校验寄存器	校验范围为寄存器写入值与 FIFO 写入值，芯片以 byte 为单位对写入值进行异或处理，寄存器先校验高 byte，后校验低 byte。 软件复位后 REGCHECK 寄存器清'0'。	

17. 2. 71 FIFODATA寄存器

寄存器名称	FIFODATA				
地址	0x70				
复位值	-				
位名称	位	读写	位说明	1	0
-	15:8			-	
FIFODATA	7:0	RW	FIFO 访问寄存器	FIFO 读写操作详见 10.3.3 章节	

第 19 章 典型应用原理图

19.1 典型应用原理图

下面给出 ES32W0030 的典型应用原理图。

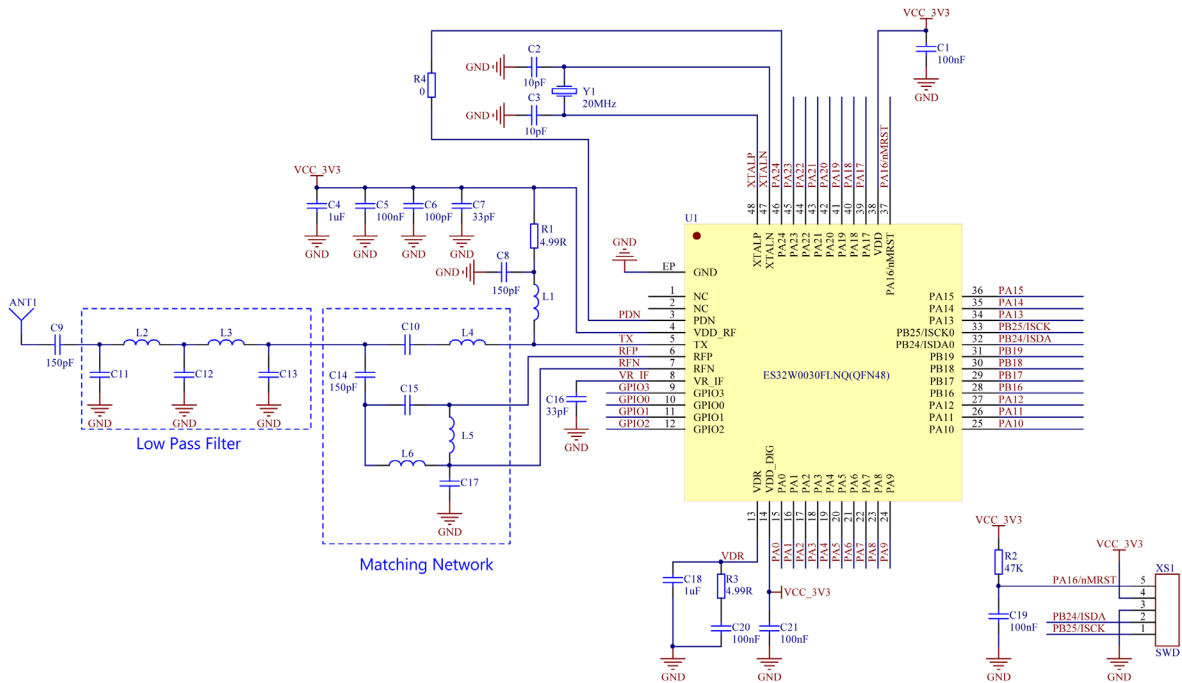


图 19-1 典型应用原理图

各频段的 RF 匹配元器件参数详见表 19-1 所示。

标号 频率	L2,L3 (nH)	C11,C13 (pF)	C12 (pF)	L4 (nH)	C10 (pF)	L5 (nH)	L6 (nH)	C15 (pF)	C17 (pF)	L1 (nH)
315MHz	22	12	22	27	6.8	47	27	3.3	6.8	120
433MHz	18	6.8	12	22	6.8	33	27	5.6	6.8	100
779MHz	6.8	6.8	12	6.8	6.8	22	10	3.3	2.7	6.8
868MHz	5.6	6.8	12	5.6	5.6	10	8.2	2.7	5.6	5.6
915MHz	4.7	6.8	12	6.8	5.6	8.2	8.2	2.7	5.6	5.1

表 19-1 不同频率下的射频元件 BOM

19.2 设计应用注意事项

(1) 芯片中央的 EP 脚必须接地。

(2) 芯片内部 RF 的电源是 VDD_RF (4 脚) 和 VDD_DIG (14 脚), MCU 的电源是 VDD (38 脚), 它们在内部是不连的, 所以都需要连接到 VCC_3V3 外部电源上。

(3) RF 收发器的 POWER DOWN 模式使能输入引脚 PDN, 可以连接 MCU 的一个通用 IO 口来控制, 如果 IO 口不够用, 也可以直接接地 (注意, 接地后无法进入 POWER DOWN 模式)。由图 19-1 可知, 焊接 0 欧姆电阻 R4 时, 可以用 PA24 脚来控制 PDN。

第 20 章 Cortex-M0 内核描述

20.1 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_S，区分该指令是否影响条件标志位。

{Rd,}，表示指令中的目标操作数 Rd 可有可没有，没有时根据不同指令直接确定。

助记符	操作数	描述	影响标志位
ADR	Rd,Label	取 Label 地址到寄存器	-
LDR	Rt,Label	按字读 memory 到 Rt，立即数寻址	-
LDR	Rt,[Rn,<Rm #imm>]	按字读 memory 到 Rt，带地址偏移寻址	-
LDRB	Rt,[Rn,<Rm #imm>]	按字节读 memory 到 Rt，0 扩展为 32 位	-
LDRH	Rt,[Rn,<Rm #imm>]	按半字读 memory 到 Rt，0 扩展为 32 位	-
LDRSB	Rt,[Rn,Rm]	按字节读 memory 到 Rt，符号位扩展	-
LDRSH	Rt,[Rn,Rm]	按半字读 memory 到 Rt，符号位扩展	-
LDM	Rn{!},reglist	批量读 memory 到 reglist，Rn 递增	-
STR	Rt,[Rn,<Rm #imm>]	按字写 memory，带地址偏移寻址	-
STRB	Rt,[Rn,<Rm #imm>]	按字节写 memory，0 扩展为 32 位	-
STRH	Rt,[Rn,<Rm #imm>]	按半字写 memory，0 扩展为 32 位	-
STM	Rn!,reglist	批量写 memory，Rn 递增	-
PUSH	Reglist	寄存器压栈	-
POP	Reglist	寄存器出栈	-
MOV{S}	Rd,<Rm #imm>	数据传送 Rd= <Rm #imm>	N,Z 或-
MVNS	Rd,Rm	Rm 按位求反之后传送到 Rd	N,Z
MRS	Rd,spec_reg	读特殊功能寄存器，Rd=spec_reg	-
MSR	Spec_reg,Rm	写特殊功能寄存器，spec_reg=Rm	N,Z,C,V 或-
ADCS	{Rd,}Rn,Rm	带进位加法	N,Z,C,V
ADD{S}	{Rd,}Rn,<Rm #imm>	加法	N,Z,C,V 或-
RSBS	{Rd,}Rn,#0	算术取反，Rd = 0-Rn	N,Z,C,V
SBCS	{Rd,}Rn,Rm	带借位减法，Rd = Rn-Rm-C	N,Z,C,V
SUB{S}	{Rt,}Rn,<Rm #imm>	不带借位减法	N,Z,C,V 或-
ANDS	{Rd,}Rn,Rm	按位逻辑与，Rd = Rn&Rm	N,Z
ORRS	{Rd,}Rn,Rm	按位逻辑或，Rd = Rn Rm	N,Z

助记符	操作数	描述	影响标志位
EORS	{Rd,}Rn,Rm	按位逻辑异或, $Rd = Rn \wedge Rm$	N,Z
BICS	{Rd,}Rn,Rm	位清除, Rm 为掩码	N,Z
ASRS	{Rd,}Rn,<Rm #imm>	算术右移	N,Z,C
LSLS	{Rd,}Rn,<Rm #imm>	逻辑左移	N,Z,C
LSRS	{Rd,}Rn,<Rm #imm>	逻辑右移	N,Z,C
RORS	{Rd,}Rn,Rm	循环右移	N,Z,C
CMP	{Rn,}<Rm #imm>	比较	N,Z,C,V
CMN	Rn,Rm	负比较, 先将 Rm 取反, 再比较	N,Z,C,V
MULS	Rd,Rn,Rm	乘法, 结果为 32 位	N,Z
REV	Rd,Rm	按字节反转 (32 位大小端数据转换)	-
REV16	Rd,Rm	按半字反转 (2 个 16 位大小端数据转换)	-
REVSH	Rd,Rm	低半字反转, 按有符号数扩展为 32 位	-
SXTB	Rd,Rm	低字节, 按有符号数扩展到 32 位	-
SXTH	Rd,Rm	低半字, 按有符号数扩展到 32 位	-
UXTB	Rd,Rm	低字节, 零扩展到 32 位	-
UXTH	Rd,Rm	低半字, 零扩展到 32 位	-
TST	Rd,Rm	位测试	N,Z
B{cond}	Label	(条件) 分支短跳转到 Label 所指处	-
BL	Label	带链接的分支跳转, 跳转到 Label 所指处	-
BX	Rm	分支长跳转	-
BLX	Rm	带链接分支长跳转, 跳转到 Rm 所指处	-
CPSID	i	屏蔽中断响应, PRIMASK.PM=1	-
CPSIE	i	允许中断响应, PRIMASK.PM=0	-
SVC	#imm	管理调用, 产生 SVC 异常	-
DMB	-	数据存储器访问隔离	-
DSB	-	数据同步隔离	-
ISB	-	指令同步隔离	-
SEV	-	触发事件	-
WFE	-	等待事件	-
WFI	-	等待中断	-
BKPT	#imm	断点	-
NOP	-	空操作	-

注: 指令 CPSID 和 CPSIE, 分别用于禁止和允许中断请求, 指令操作码相同, 只是操作数不同, 实际为一条指令。

20.2 Cortex-M0 内核寄存器

Cortex-M0 内核寄存器如下图所示：

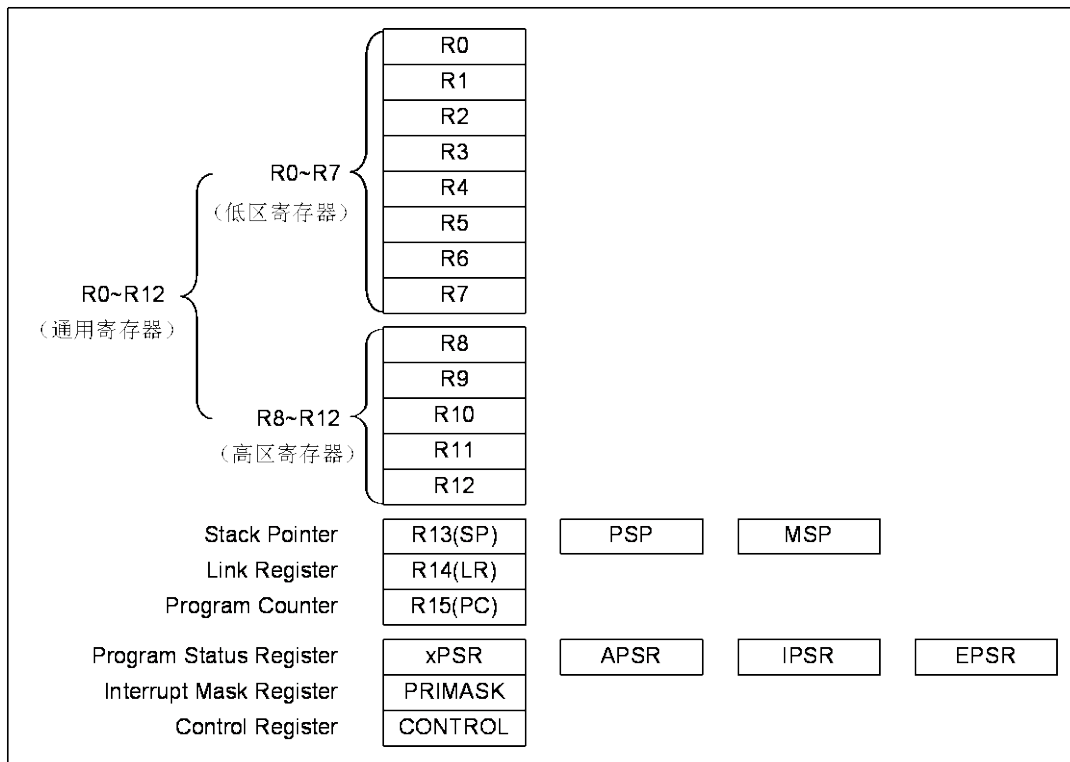


图 20-1 内核寄存器

20.2.1 通用寄存器R0~R12

R0~R12 为 32 位通用寄存器，用于数据操作。

20.2.2 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：

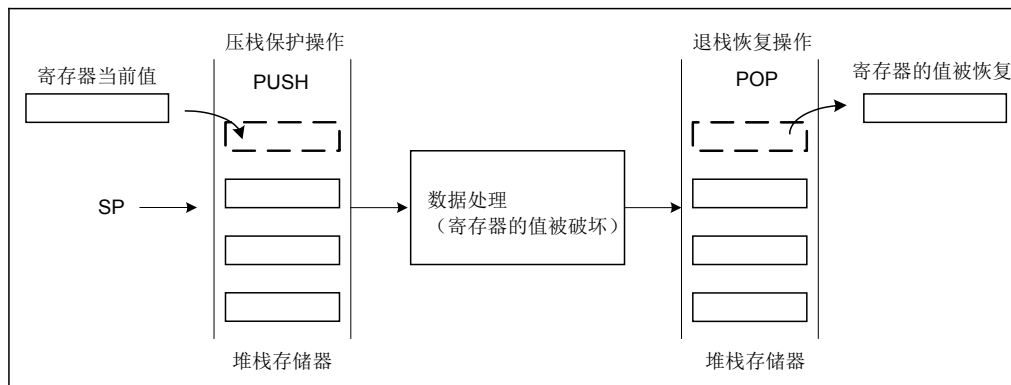


图 20-2 堆栈操作示意图

20.2.3 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。

20.2.4 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。

20.2.5 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 ≥ 232), 则 **C=1**, 否则 **C=0**;

对减法指令, 如果执行结果无借位 (结果 ≥ 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。

20.2.6 异常/中断屏蔽寄存器 PRIMASK

异常/中断屏蔽寄存器 PRIMASK, 可用于处理器屏蔽所有的异常/中断处理。

	31:1	0
PRIMASK	Reserved	PM

当屏蔽位 **PM=1** 时, 禁止处理器响应所有可屏蔽异常/中断, 不可屏蔽中断 NMI 除外。当 **PM=0** 时, 不影响处理器对异常/中断的正常响应。

可以使用 MRS 和 MSR 指令访问 PRIMASK 寄存器, 还可以使用专用的 CPSID 和 CPSIE 指令来设置寄存器中的 **PM** 位。

20.2.7 控制寄存器 CONTROL

控制寄存器 CONTROL, 可用于在线程模式下, 选择所使用的堆栈指针。

	31:2	1	0
CONTROL	Reserved	SPSEL	Reserved

当堆栈指针选择位 **SPSEL=0** 时，选择 **MSP (SP_main)** 作为当前堆栈指针；当 **SPSEL=1** 时，选择 **PSP (SP_process)** 作为当前堆栈指针。

在异常/中断处理模式下，总是使用 **MSP** 作为堆栈指针，**SPSEL=0**，且只读，不可写；处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。

第 21 章 电气特性

21.1 MCU电气特性

21.1.1 最大标称值

参数	符号	条件	标称值	单位
电源电压	VDD	VSS=0V	-0.3 ~ 7.5	V
管脚输入电压	V _{IN}	VSS=0V	-0.3 ~ VDD + 0.3	V
管脚输出电压	V _{OUT}	VSS=0V	-0.3 ~ VDD + 0.3	V
VDD 管脚最大输入电流	I _{MAXVDD}	VDD=5.0V, 25°C	100	mA
VSS 管脚最大输出电流	I _{MAXVSS}	VDD=5.0V, 25°C	120	mA
芯片存储温度	T _{STG}	—	-55 ~ 125	°C
芯片最高结温	T _{JUNC}	—	125	°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。

21.1.2 MCU工作条件

参数	符号	工作条件	最小值	最大值	单位
芯片工作温度	T _{OPR}	—	-40	85	°C
芯片结温	T _{JUNC}	—	-40	105	°C
芯片工作电压	VDD	—	2.2	5.5	V
AHB 总线频率	F _{HCLK}	—	0	48	MHz
APB 总线频率	F _{PCLK}	—	0	48	MHz

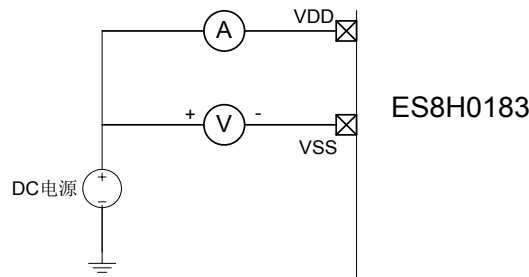
21.1.3 MCU功能模块工作电压范围

参数	符号	工作温度	VDD 电压	备注
ADC 工作电压	V _{ADC1}	-40 ~ 85°C	2.5~5.5V	参考电压为 VDD 或外部 AVREFP
	V _{ADC2}	-40 ~ 85°C	2.8~5.5V	参考电压为内部 VREFP

21.1.4 芯片上电和下电工作条件表 (-40 ~ 85°C)

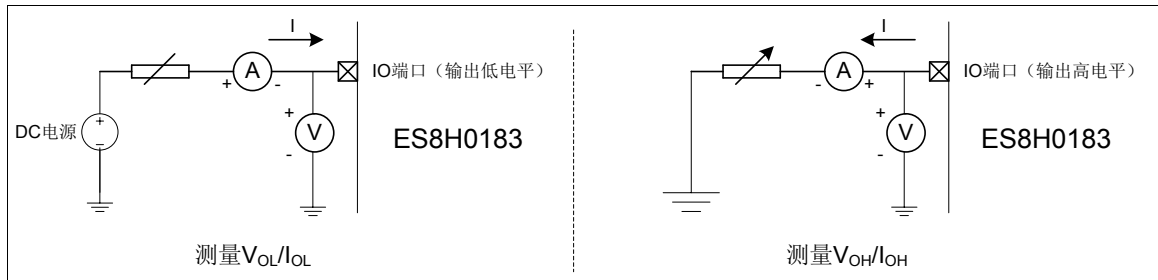
参数	符号	最小值	最大值	单位
VDD 上电初始电压	V_{start}	0	0.2	V
VDD 上升速率	T_{VDD}	20	—	us/V
VDD 下降速率		50	—	us/V

21.1.5 芯片功耗参数测量方法



芯片功耗测量连接示意图

21.1.6 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

21.1.7 MCU功耗特性参数

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°C ~ 85°C
芯片静态电流	I_{DD}	—	2.5	—	mA	25°C, 上电复位, VDD = 5V, 所有的 I/O 端口输入低电平, MRSTN=0。
深度睡眠模式下芯片电流	I_{PD1}	—	2.8	—	μA	25°C, 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)

参数	符号	最小值	典型值	最大值	单位	工作条件
浅睡眠模式下 芯片电流	I_{PD2}	—	1.2	—	mA	25°C, 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_{PD3}	—	2.8	—	mA	25°C, 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_{OP1}	—	1.2	—	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 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}	—	9.0	—	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 48MHz HRC 时钟。

21.1.8 MCU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	I_{HRC1}	—	1	—	mA	25°C, VDD = 5V

参数	符号	最小值	典型值	最大值	单位	工作条件
48MHz 电流						
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	—	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}	—	0.6	—	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
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	

21.1.9 MCU芯片输入端口特性

芯片工作温度范围：-40℃ ~ 85℃						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输入高电平 (TTL 输入)	V_{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℃, 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}	—	0.7	—	V	25℃, 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=VSS 或 VDD
I/O 端口弱上拉电阻	R_{WPU}	—	55	—	kΩ	25℃, VDD=5.0V Vpin = VSS
I/O 端口弱下拉电阻	R_{WPD}	—	55	—	kΩ	25℃, VDD=5.0V Vpin = VDD
I/O 输入端口 VDD/2 输出精度	$V_{VDD/2}$	—	±3%	—		25℃, VDD=5V, 弱上拉和弱下拉同时使能

注：I/O 端口弱上拉和弱下拉电阻，在全温度范围内（-40~ 85℃），相对于常温的变化量在±10%以内。

21. 1. 10 MCU芯片输出端口特性

芯片工作温度范围: -40°C ~ 85°C						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输出高电平	V_{OH}	0.8VDD	—	VDD	V	2.2V ≤ VDD ≤ 5.5V, 端口无负载
I/O 端口输出低电平	V_{OL}	VSS	—	0.2VDD	V	
I/O 端口灌电流	I_{OL1}	—	10	—	mA	25°C, VDD = 5V $V_{OL} = 0.6V$, 普通驱动
	I_{OL2}	—	22	—		25°C, VDD = 5V $V_{OL} = 0.6V$, 增强驱动
	I_{OL3}	—	40	—		25°C, VDD = 5V $V_{OL} = 1.4V$, 增强驱动
I/O 端口拉电流	I_{OH1}	—	7	—	mA	25°C, VDD = 5V $V_{OH} = 4.4V$, 普通驱动
	I_{OH2}	—	14	—		25°C, VDD = 5V $V_{OH} = 4.4V$, 增强驱动

注: I/O 端口拉电流 I_{OH} 和灌电流 I_{OL} 的驱动能力随温度降低而增大。

21. 1. 11 MCU系统时钟规格

参数	符号	最小值	典型值	最大值	单位	测试条件
系统时钟频率	F_{OSC}	—	—	48M	Hz	2.2V ≤ VDD ≤ 5.5V
系统时钟周期	T_{OSC}	20.8	—	—	ns	
机器周期	T_{inst}	—	$T_{OSC} * 1$	—	ns	
外部时钟高电平和低电平时间	T_{OSL}, T_{OSH}	20	—	—	ns	
外部时钟边沿上升和下降时间	T_{OSR}, T_{OSF}	—	—	8	ns	

21. 1. 12 MCU芯片ESD特性

◆ 静电放电特性参数

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 (机器模型)	V_{ESDMM}	2	300	V	25°C, 遵循标准 JESD22-A115
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

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。

21. 1. 13 ADC模块特性

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	
输入电容	C_{IN}	—	40	—	pF	
模拟通道推荐输入电阻	R_{IN}	—	—	2K	Ω	
AD 转换时钟周期	T_{AD1}	1	—	—	μs	
	T_{AD2}	0.5	—	—	μs	
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	—	± 1	± 2	LSB	ADC 时钟频率为 1MHz
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。

21. 1. 14 ADC转换时钟源选择

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	$T_{ADCLK} = 1\mu s$
FPClk /4	不推荐使用	不推荐使用	不推荐使用	$T_{ADCLK} = 2\mu s$
FPClk /8	不推荐使用	不推荐使用	不推荐使用	$T_{ADCLK} = 4\mu s$
FPClk /16	不推荐使用	不推荐使用	$T_{ADCLK} = 1\mu s$	$T_{ADCLK} = 8\mu s$

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 /32	不推荐使用	T _{ADCLK} = 1us	T _{ADCLK} = 2us	T _{ADCLK} = 16us
FPCLK /64	T _{ADCLK} = 1.3us	T _{ADCLK} = 2us	T _{ADCLK} = 4us	T _{ADCLK} = 32us
FPCLK /256	T _{ADCLK} = 5.3us	T _{ADCLK} = 8us	T _{ADCLK} = 16us	T _{ADCLK} = 128us
FLRC	T _{ADCLK} = 31us	T _{ADCLK} = 31us	T _{ADCLK} = 31us	T _{ADCLK} = 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 _{ADCLK} = 0.5us
FPCLK /2	不推荐使用	不推荐使用	不推荐使用	T _{ADCLK} = 1us
FPCLK /4	不推荐使用	不推荐使用	不推荐使用	T _{ADCLK} = 2us
FPCLK /8	不推荐使用	不推荐使用	T _{ADCLK} = 0.5us	T _{ADCLK} = 4us
FPCLK /16	不推荐使用	T _{ADCLK} = 0.5us	T _{ADCLK} = 1us	T _{ADCLK} = 8us
FPCLK /32	T _{ADCLK} = 0.67us	T _{ADCLK} = 1us	T _{ADCLK} = 2us	T _{ADCLK} = 16us
FPCLK /64	T _{ADCLK} = 1.3us	T _{ADCLK} = 2us	T _{ADCLK} = 4us	T _{ADCLK} = 32us
FPCLK /256	T _{ADCLK} = 5.3us	T _{ADCLK} = 8us	T _{ADCLK} = 16us	T _{ADCLK} = 128us
FLRC	T _{ADCLK} = 31us	T _{ADCLK} = 31us	T _{ADCLK} = 31us	T _{ADCLK} = 31us

21. 1. 15 ADC内部参考和 1/4VDD电源分压电压特性表

参数	符号	最小值	典型值	最大值	单位	测试条件
ADC 内部参考电压	V _{REF}	2.028	2.048	2.068	V	25°C, VDD=5V, CHOP_EN=1
1/4VDD 电源分压	V _{1/4VDD}	1.23	1.25	1.27	V	25°C, VDD=5.0V
		1.225	1.25	1.275	V	-40°C~85°C, VDD=5.0V

注：ADC 内部参考电压在全温度范围内 (-40°C~85°C) 相对于常温条件下的偏差范围约±1%。

21. 1. 16 内部时钟源特性

参数	符号	最小值	典型值	最大值	单位	测试条件
HRC 时钟频率	F _{HRC1}	47.52	48	48.48	MHz	25°C, VDD=5.0V
	F _{HRC2}	31.68	32	32.32	MHz	
	F _{HRC3}	15.84	16	16.16	MHz	
	F _{HRC4}	1.98	2	2.02	MHz	
	F _{HRC1}	47.18	48	48.96	MHz	-20°C~85°C, VDD=2.2V~5.5V

	F _{HRC2}	31.36	32	32.64	MHz	-40℃~85℃, VDD=2.2V~5.5V
	F _{HRC3}	15.68	16	16.32	MHz	
	F _{HRC4}	1.954	2	2.046	MHz	
	F _{HRC1}	47.04	48	48.96	MHz	
	F _{HRC2}	31.2	32	32.64	MHz	
	F _{HRC3}	15.6	16	16.32	MHz	
	F _{HRC4}	1.944	2	2.046	MHz	
HRC 起振时间	T _{HRC}	—	10	—	us	-40℃~85℃, VDD=2.2V~5.5V
LRC 时钟频率	F _{LRC}	30.7	32	33.3	KHz	25℃, VDD=5.0V
		30	32	34	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。

21. 1. 17 BOR 模块特性

BOR 档位由配置字位 CFG_BORV (CFG_WORD0<5:4>) 设置。

CFG_BORV<1:0>	最小值	典型值	最大值	单位	测试条件
00	—	—	—	—	-40~85℃
01	2.35	2.5	2.65	V	
10	1.9	2.1	2.3	V	
11	2.95	3.1	3.25	V	

21. 1. 18 LVD 模块特性

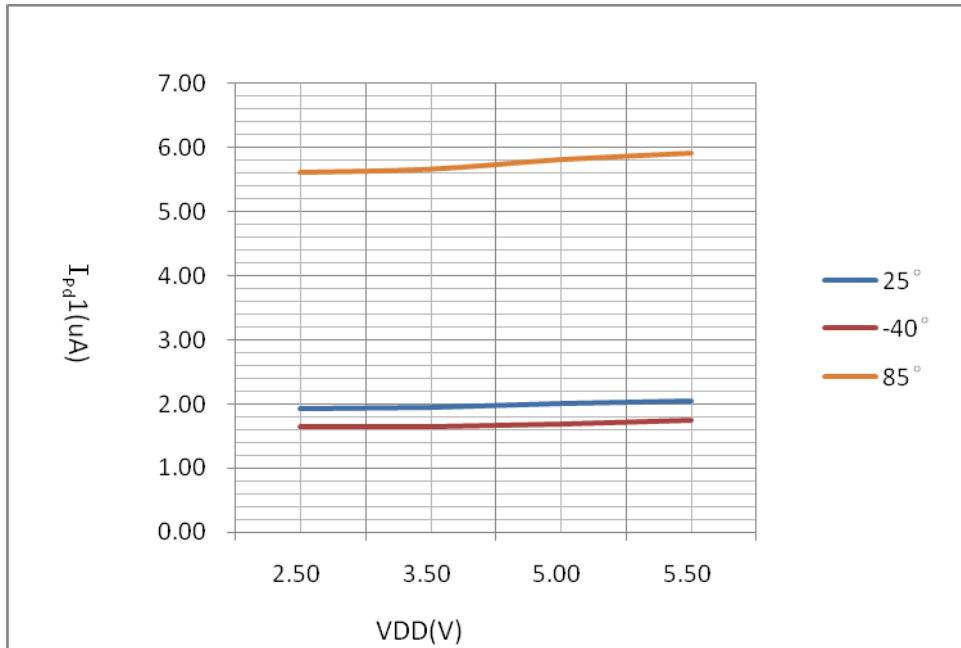
LVD_VS<3:0>	最小值	典型值	最大值	单位	测试条件	
VDD 下降, LVDO 低电压 状态标志置 1	0000	2.15	2.3	2.4	V	-40~85℃
	0001	2.25	2.4	2.6	V	
	0010	2.55	2.7	2.8	V	
	0011	2.65	2.8	3.0	V	
	0100	2.85	3.0	3.2	V	
	0101	3.45	3.6	3.8	V	
	0110	3.95	4.1	4.3	V	
	0111	4.55	4.7	4.9	V	
LVD 电压检测迟滞窗口	—	50	130	mV	-40~85℃	

21.2 MCU 参数特性图

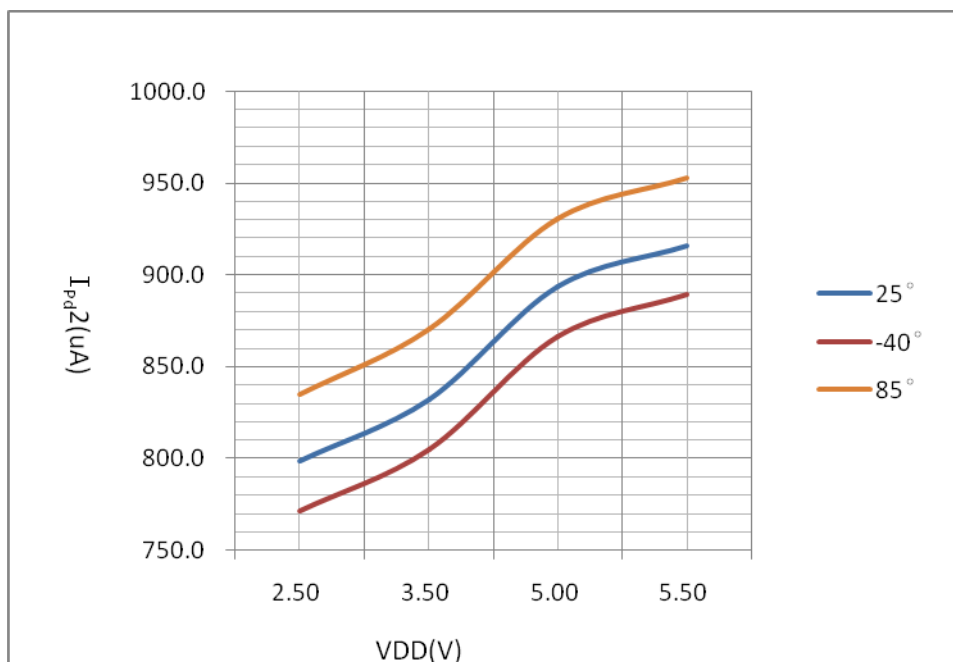
本节中所列图示均为抽样测试，仅作为设计参考之用。其中部分图示中所列的数据已超出指定的操作范围，此类信息也仅供参考，芯片只保证在指定的范围内正常工作。

21.2.1 MCU功耗特性

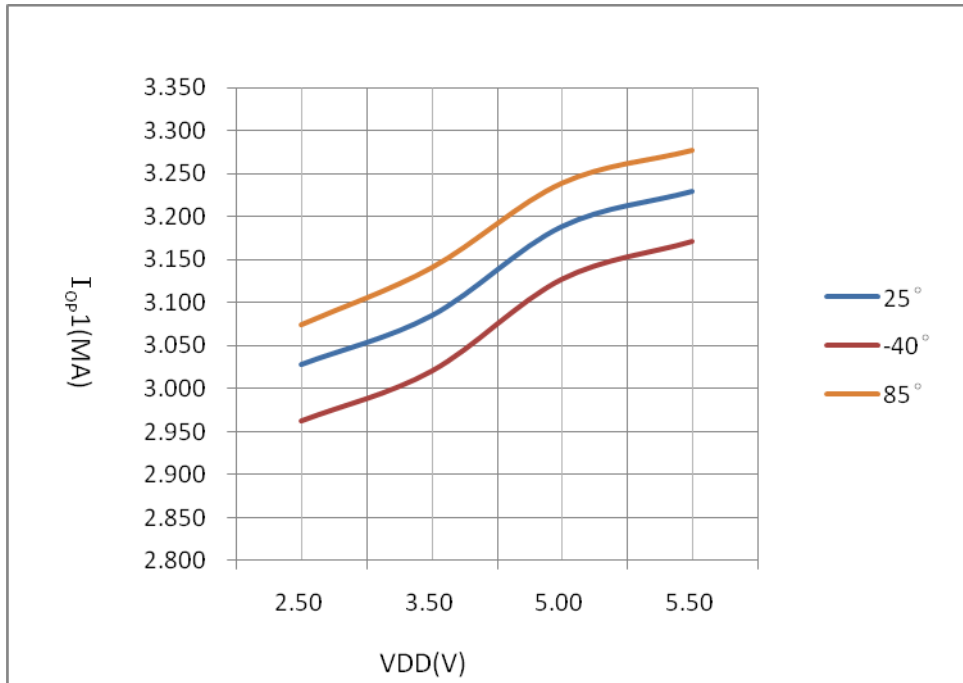
- ◆ MCU 深度睡眠模式电流随电压-温度变化特性图（IWDT 不使能，所有 I/O 端口输出固定电平，无负载）



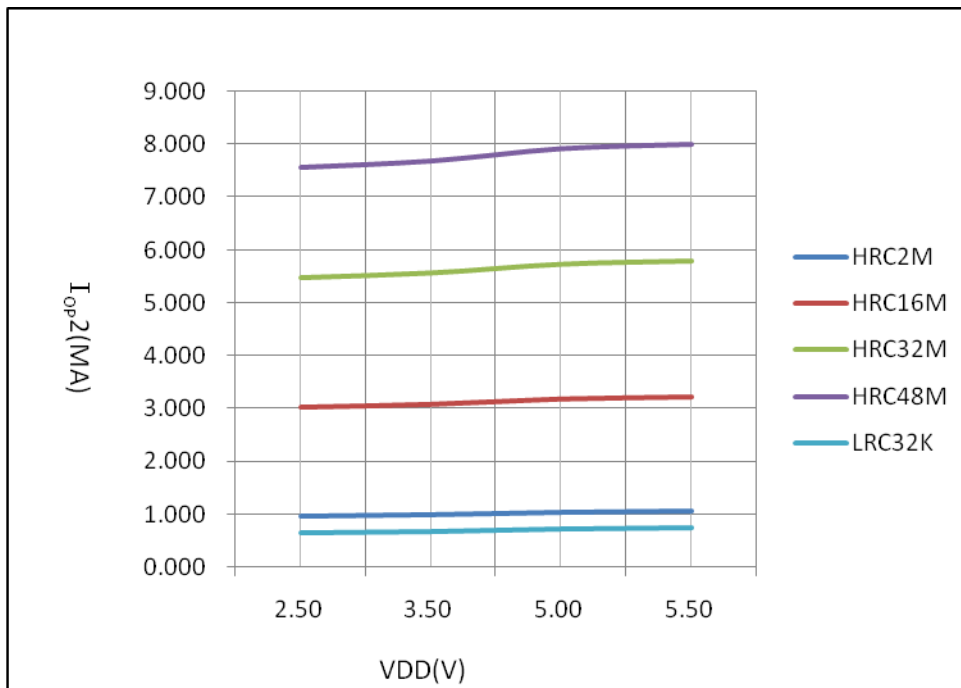
- ◆ MCU 浅睡眠模式电流随电压-温度变化特性图（IWDT 不使能，所有 I/O 端口输出固定电平，无负载；系统主时钟为内部 HRC 16MHz 时钟）



- ◆ MCU 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)

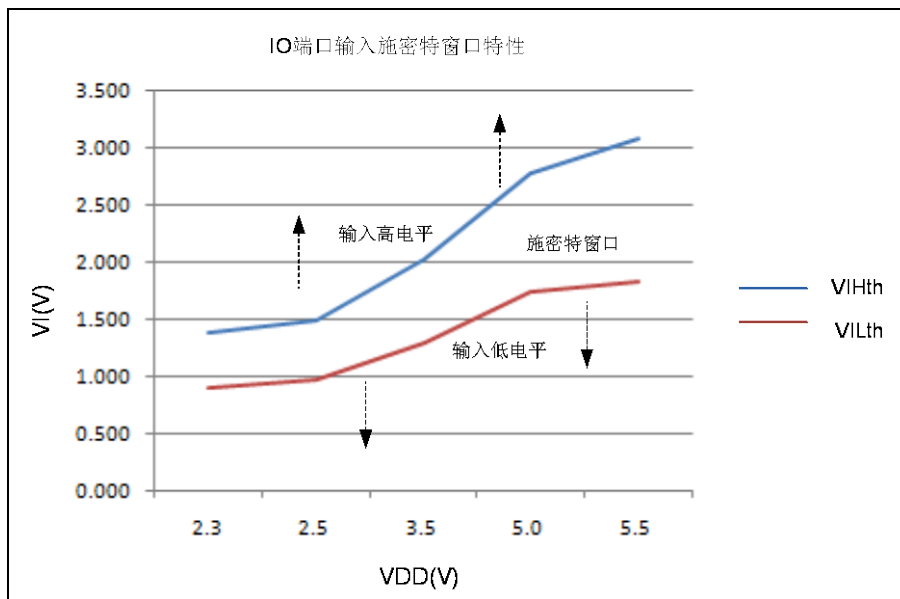


- ◆ MCU 运行模式电流随电压-系统时钟频率变化特性图 (IWDT 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, ADC 使用 VDD 作为正向参考电压; 室温 25°C)

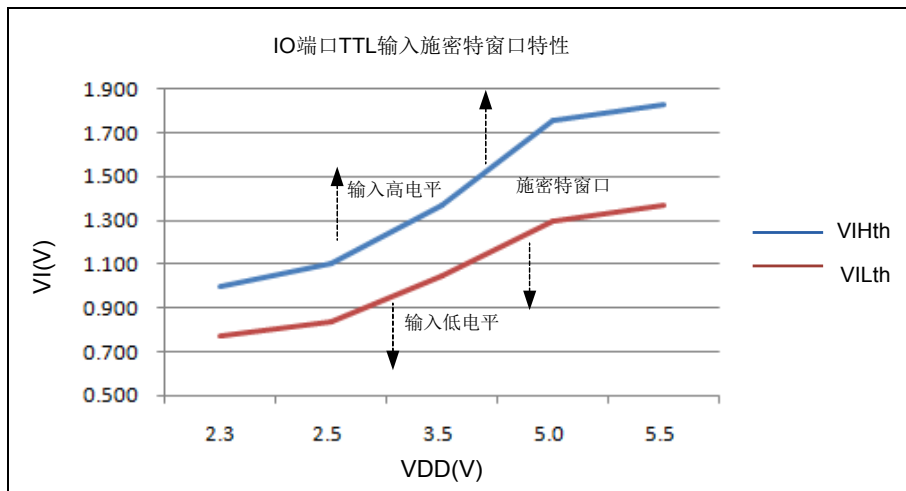


21.2.2 MCU IO端口输入特性

◆ 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℃）



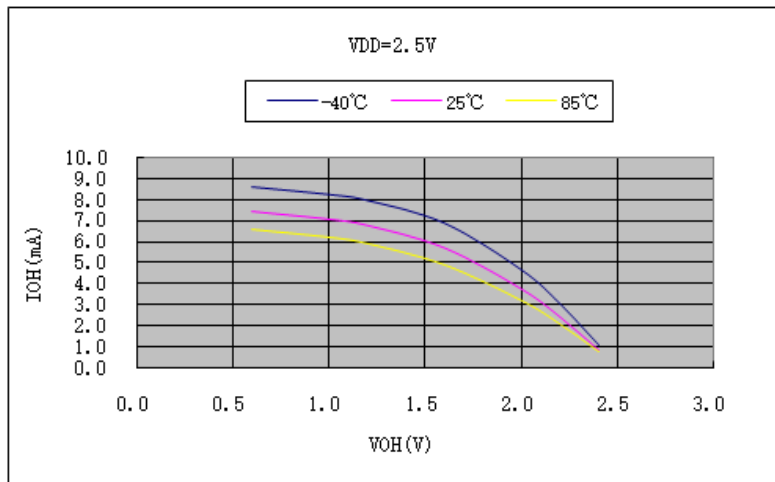
注 1: VIHth 为施密特窗口的上阈值电平，大于该阈值的输入电平为高；

注 2: VILth 为施密特窗口的下阈值电平，小于该阈值的输入电平为低；

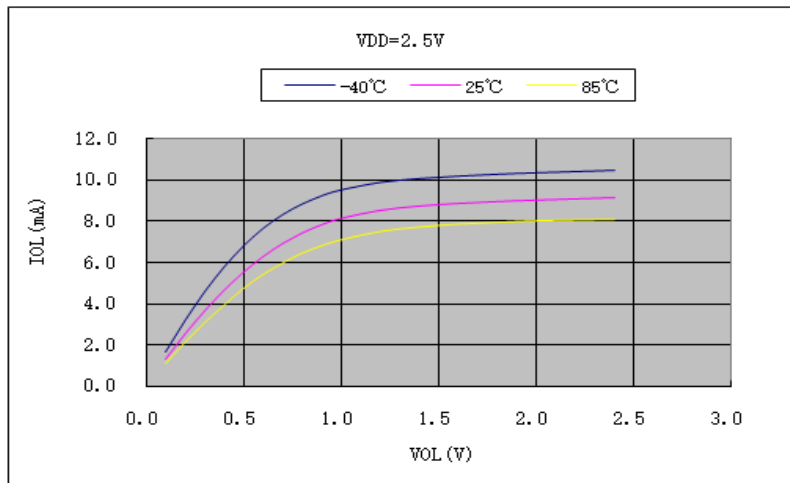
注 3: VIHth 和 VILth 之间为施密特窗口，在窗口内的输入电平不确定，可能为高或低。

21.2.3 MCU IO端口输出特性（普通驱动）

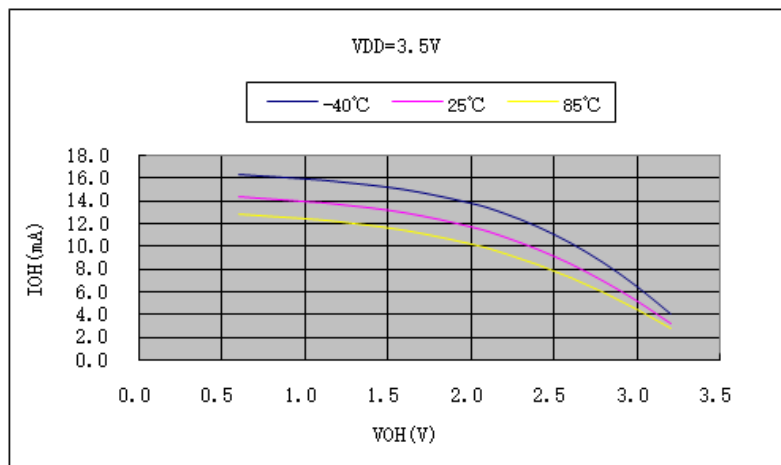
◆ A: V_{OH} vs I_{OH} @VDD=2.5V



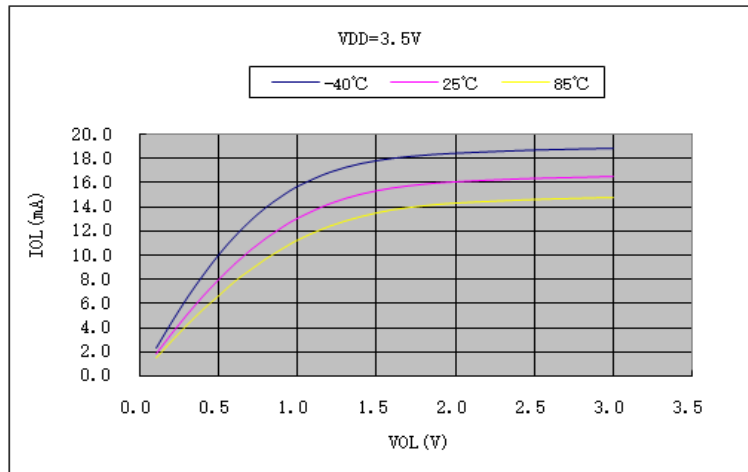
◆ B: V_{OL} vs I_{OL} @VDD=2.5V



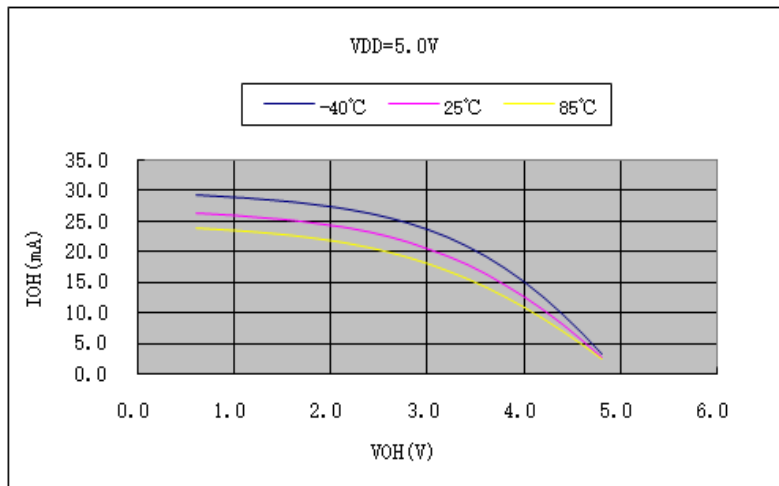
◆ C: V_{OH} vs I_{OH} @VDD=3.5V



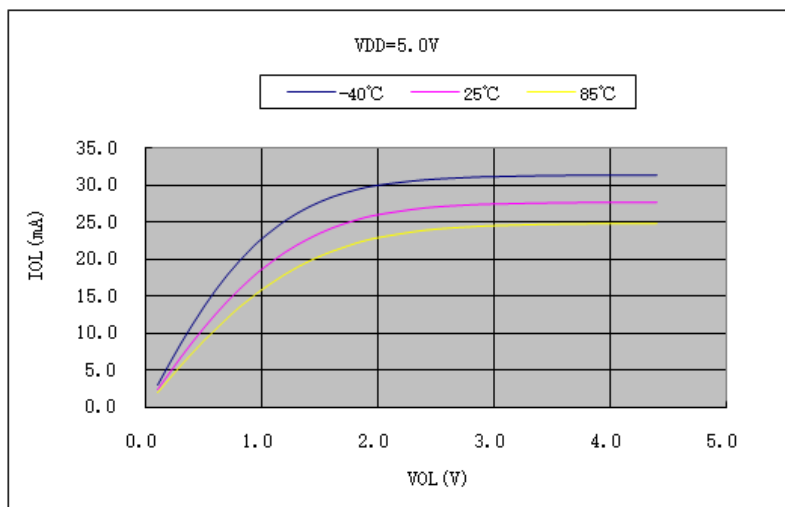
◆ D: V_{OL} vs I_{OL} @VDD=3.5V



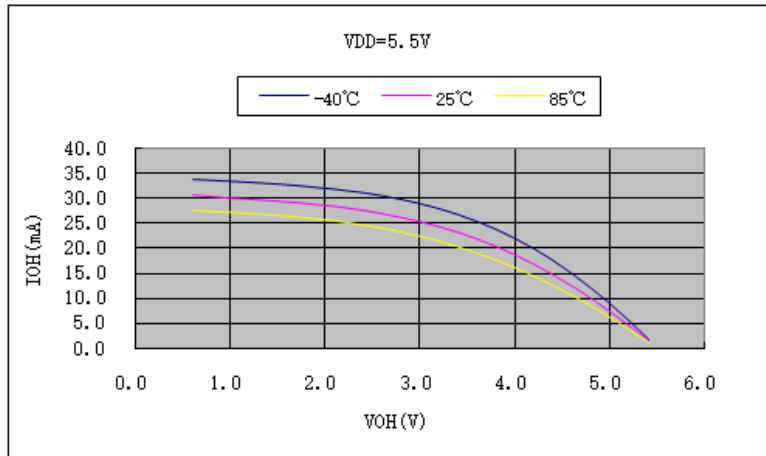
◆ E: V_{OH} vs I_{OH} @VDD=5.0V



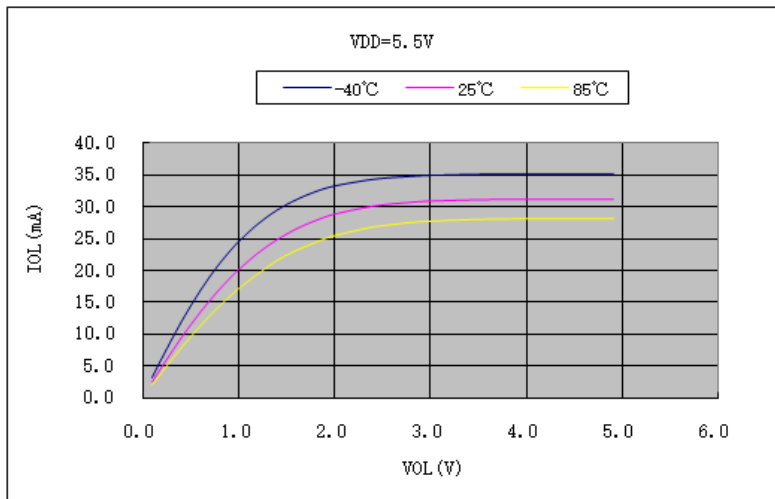
◆ F: V_{OL} vs I_{OL} @VDD=5.0V



◆ G: V_{OH} vs I_{OH} @VDD=5.5V

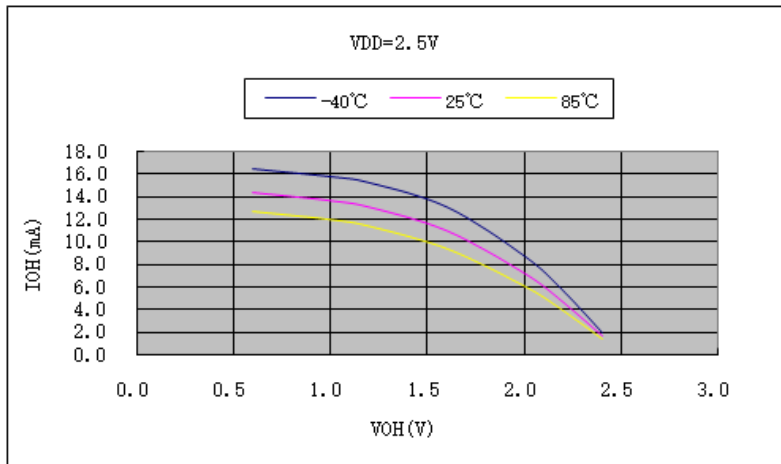


◆ H: V_{OL} vs I_{OL} @VDD=5.5V

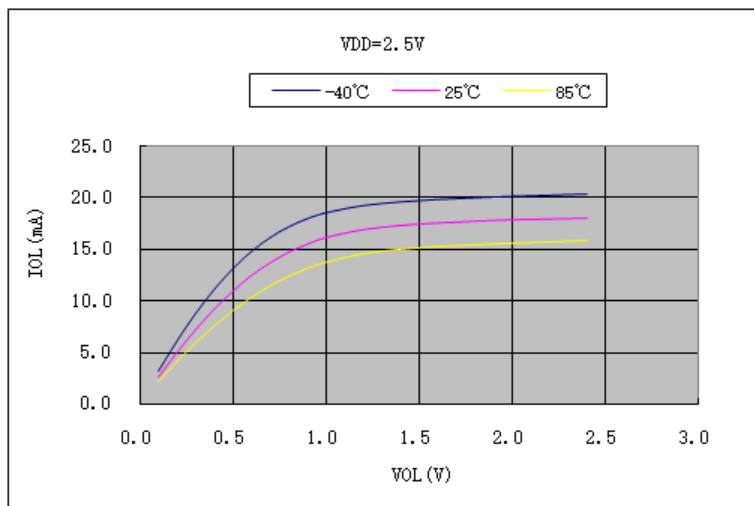


21.2.4 MCU IO端口输出特性（增强驱动）

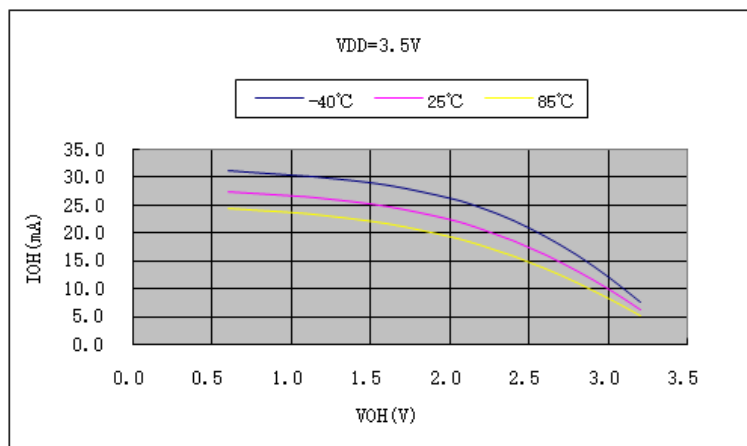
◆ A: V_{OH} vs I_{OH} @ $V_{DD}=2.5V$



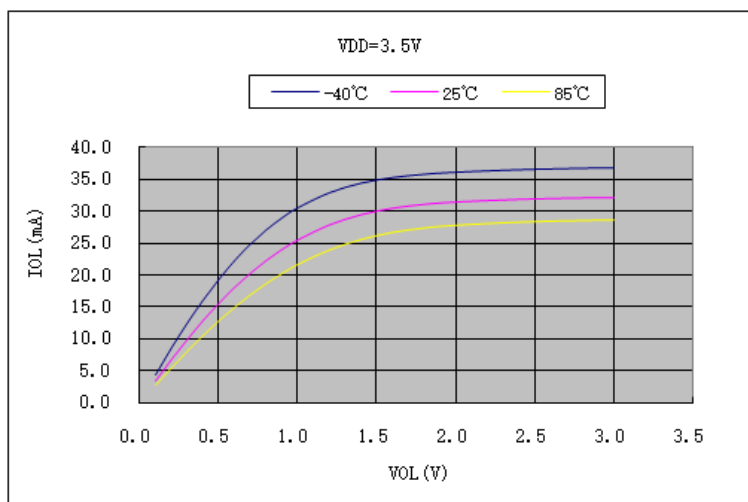
◆ B: V_{OL} vs I_{OL} @ $V_{DD}=2.5V$



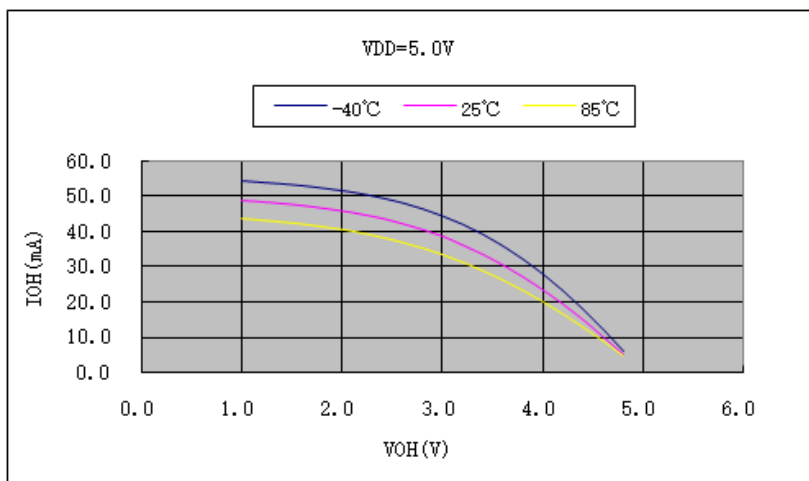
◆ C: V_{OH} vs I_{OH} @ $V_{DD}=3.5V$



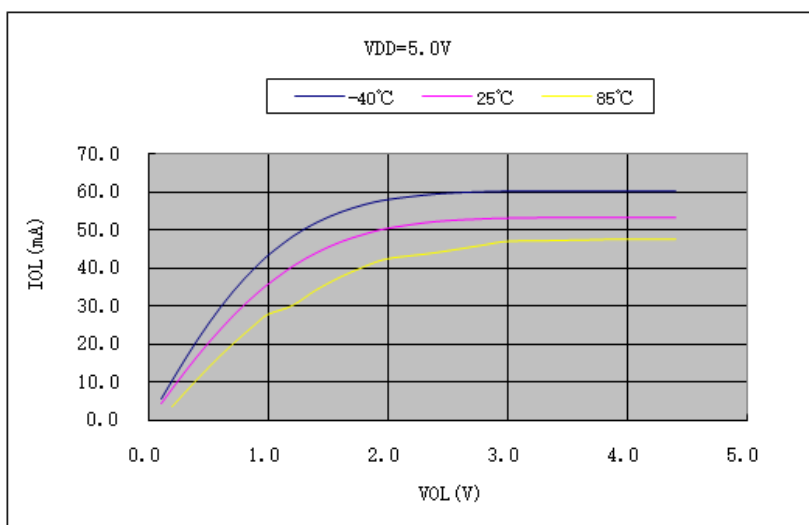
◆ D: V_{OL} vs I_{OL} @ $V_{DD}=3.5V$



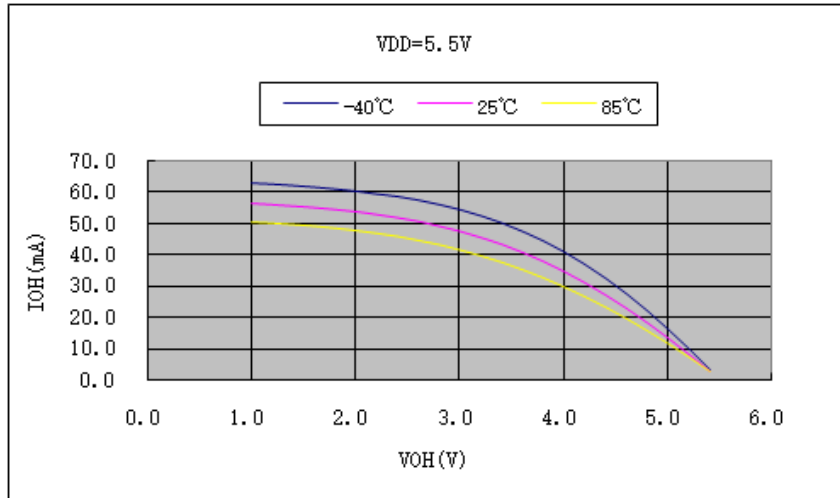
◆ E: V_{OH} vs I_{OH} @ $V_{DD}=5.0V$



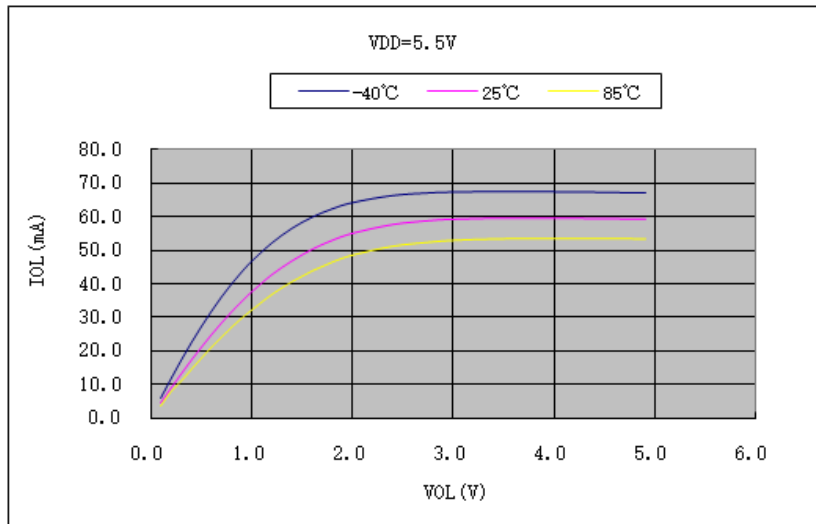
◆ F: V_{OL} vs I_{OL} @ $V_{DD}=5.0V$



◆ G: V_{OH} vs I_{OH} @ $V_{DD}=5.5V$



◆ H: V_{OL} vs I_{OL} @ $V_{DD}=5.5V$



21.3 RF收发器电气特性

21.3.1 功耗参数

参数	符号	条件	最小值	典型值	最大值	单位
省电模式功耗	POWER DOWN	所有模块关闭	-	100	-	nA
	DEEP SLEEP	低功耗数字接口打开(保持寄存器的值), 其它模块关闭	-	3	-	μA
	IDLE	数字 LDO 工作在正常模式, 晶振使能, 数字电路有时钟	-	2	-	mA
接收模式功耗	RX	码率 10kbps, 载波 433MHz	10	18	-	mA
发送模式功耗	TX1	+20 dBm 发射功率	-	90	-	mA
	TX2	+10 dBm 发射功率	-	30	-	mA

表 21-1 芯片功耗

21.3.2 接收机 (RX) 参数

参数	符号	条件	最小值	典型值	最大值	单位
接收频率范围	FSYNTH1	20MHz 晶振	235	-	522	MHz
	FSYNTH2	20MHz 晶振	700	-	1045	MHz
灵敏度	RX_1	(BER<0.1%,10kbps, dev=25kHz,G2FSK, BT=0.5,Frq=433MHz)	-	-114	-	dBm
	RX_2	(BER<0.1%,50kbps, dev=25kHz,G2FSK, BT=0.5,Frq=433MHz)	-	-109	-	dBm
接收最大信号	PMAX_IN	-	-	>17	-	dBm
接收带宽	BW	-	10	-	200	kHz
输入三阶交调 (IIP3)	IIP3RX	-	-	-20	-	dBm
RX 输入阻抗 (未做匹配时, RX 的双端输入)	RIN-RX	433MHz	-	124.85-85.55j	-	Ω
RSSI 分辨率	RES-RSSI	-	-	±3	-	dB
±1-Ch 邻道抑制 (BER <0.1%)	C/I1-CH	要求参考信号灵敏度 3db 以上, 码速 10kbps, 高斯频率偏移键控(G2FSK), 频偏 25kHz, 基带滤波 (BT=0.5), 带宽 70k, 通道间隔 200K, 干扰源为连续载波信号	-	-41	-	dB
±2-Ch 邻道抑制 (BER <0.1%)	C/I2-CH		-	-44	-	dB
≥ ±3-Ch 邻道抑制 (BER <0.1%)	C/I3-CH		-	-47	-	dB
≥ ±4-Ch 邻道抑制 (BER <0.1%)	C/I4-CH		-	-50	-	dB
阻塞 (1MHz)	1M BLOCK	要求参考信号灵敏度 3db 以上, 码速 10kbps, 高斯频率偏移键控(G2FSK), 频偏 25kHz, 基带滤波 (BT=0.5), 带宽 70k, 通道间隔 200K, 干扰源为连续载波信号	-	-55	-	dB
阻塞 (10MHz)	10M BLOCK	要求参考信号灵敏度 3db 以上, 码速 10kbps, 高斯频率偏移键控(G2FSK), 频偏 25kHz, 基带滤波 (BT=0.5), 带宽 70k, 通道间隔 200K, 干扰源为连续载波信号	-	-65	-	dB
镜相抑制	ImREJ	中频 IF=350kHz	-	-45	-	dB
同频率干扰	CoREJ	-	-	9	-	dB

表 21-2 接收机 (RX) 特性

21.3.3 发射机 (TX) 参数

参数	符号	条件	最小值	典型值	最大值	单位
发射机频率范围	FSYNTH1	20MHz 晶振	235	-	522	MHz
	FSYNTH2	20MHz 晶振	700	-	1045	MHz
GFSK 模式数据比特率	DR GFSK	-	1	-	100	kbps
调制频率偏差	Δf	-	-50	-	50	kHz
调制频偏步长	Δf_{RES}	-	-	305@20MHz 396@26MHz	-	Hz
输出功率范围	PTX	-	-20	-	+20	dBm
TX RFPA 输出步长	RFPA_OUT	-	-	3	-	dBm
TX RF 输出功率随温度变化	PRF_TEMP	-25°C ~ +85°C	-	2	-	dB
TX RF 输出功率随频率变化	PRF_FREQ	任何的频段测量	-	1	-	dB
发射调制滤波	BT	高斯滤波带宽时间积	0.5	-	1	
杂散辐射	POB-TX1	Pout=10dBm, Frequencies <1 GHz	-	-	-37	dBm
	POB-TX2	1~12.75 GHz, 除谐波以外	-	-	-37	dBm
谐波	P2 HARM	利用参考设计最大输出功率 (+17dBm)发射匹配网络和滤波, 输出功率线性递减	-	-	-30	dBm
	P3 HARM		-	-	-30	dBm
说明: 1. 所有规格保证生产测试, 特殊情况另有说明。 2. 资格认证。						

表 21-3 发射机 (TX) 特性

21.3.4 频率合成器参数

参数	符号	条件	最小值	典型值	最大值	单位
频率合成器的频率范围	FSYNTH1	20MHz 晶振	235	-	522	MHz
	FSYNTH2	20MHz 晶振	700	-	1045	MHz
频率合成器分辨率	FRES	-	2.2@20MHz 2.86@26MHz	-	4.4@20MHz 5.7@26MHz	Hz
参考频率	fREF	20MHz 或者 26MHz	-	20/26	-	MHz
频率合成器频率容差	ERR_FREQ	其容差值主要由晶体决定	-	±20	-	ppm
参考频率输入电平	fREF_LV	当使用参考频率替代晶体时，输入信号峰峰值(Vpp)	0.4	-	-	V
频率合成器建立时间	tLOCK	XOSC 运行稳定后，PLL 的建立时间（包括 VCO 频率校正时间）	30	60	80	μs
相位噪声	L (fM)	F = 10 kHz@20MHz F = 10 kHz@26MHz	-	-95	-	dBc/Hz
		F = 100 kHz@20MHz F = 100 kHz@26MHz	-	-98	-	dBc/Hz
		F = 1 MHz@20MHz F = 1 MHz@26MHz	-	-118	-	dBc/Hz
		F = 10 MHz@20MHz F = 10 MHz@26MHz	-	-130	-	dBc/Hz
说明： 1. 所有规格保证生产测试，特殊情况另有说明。 2. 资格认证。						

表 21-4 频率合成器特性

21.3.5 振荡器参数

参数	符号	条件	最小值	典型值	最大值	单位
校准频率	FREQ	-	31.94	32	32.05	kHz
校准之后的频率精度	RES_FREQ	理论值	-	-	±1	%
温度系数	FAC_TEMP	在校准之后当供给电压改变时的频率漂移	-	+0.6	-	% / °C
供给电压系数	FAC_V	在校准之后当供给电压改变时的频率漂移	-	+5	-	% / V
初始校准时间	CAL_TIME	当 RC 振荡器开启时，尽管晶体振荡器在工作，校准还是在后台持续的进行。	-	3	-	ms
唤起周期	TIME_START	可根据寄存器配置做相应的调整	-	-	-	s
说明： 1. 所有规格保证生产测试，特殊情况另有说明。 2. 资格认证。						

表 21-5 低功耗 RC 振荡器特性

参数	符号	条件	最小值	典型值	最大值	单位
晶体频率	CRY_FRE		-	20/26	-	MHz
容差	ERR_FREQ	这是总容差，包括初始容差、老化和温度依赖，可接受的晶体容差决定于 RF 频率和信道空间/带宽	-	±20	-	ppm
ESR	ESR	-	-	-	100	Ω
开始时间	T_START	-	-	300	-	μs
说明： 1. 所有规格保证生产测试，特殊情况另有说明。 2. 资格认证。						

表 21-6 石英晶体振荡器特性

第 22 章 编程调试接口

22.1 概述

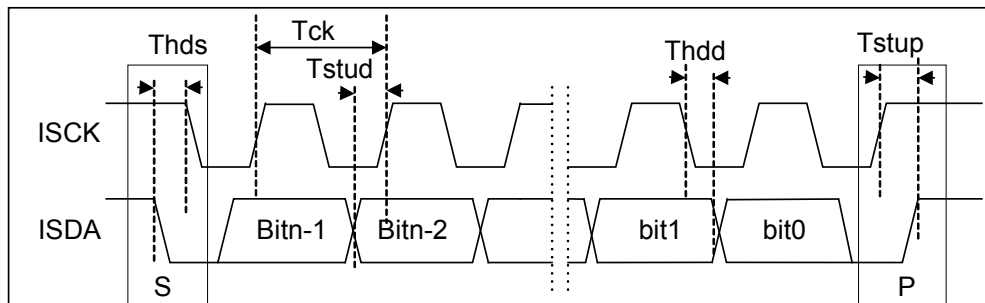
为方便应用程序及实际系统调试，芯片内部集成 ISP 在线编程接口和 SWD 串行调试接口，通过上海东软载波微电子有限公司授权的 ISP 编程器、SWD 调试器可实现芯片在线编程、仿真调试功能。

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。

22.2 ISP编程接口

22.2.1 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：



注：ISCK 为高电平时 ISDA 的下降沿为起始位

ISCK 为高电平时 ISDA 的上升沿为停止位

Thds: 起始位 SCK 保持时间

Tck: 串行时钟周期

Tstud: 串行数据建立时间

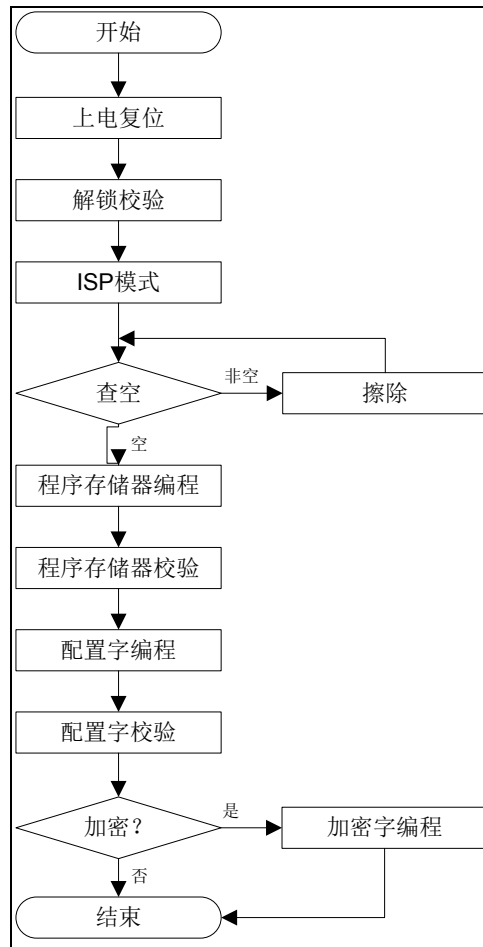
Thdd: 串行数据保持时间

Tstup: 停止位 SCK 建立时间

parameter	max	type	min
Thds			20ns
Tck			100ns
Tstud			20ns
Thdd			20ns
Tstup			20ns

◇ ISP 在线编程接口请参考“管脚对照表”章节的描述。

22.2.2 操作流程



22.3 SWD调试接口

22.3.1 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_SWID，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，与 GPIO 复用，位于 PB25。

SWDIO: 串行数据输入/输出端口，与 GPIO 复用，位于 PB24。

在 SWD 使用过程中，软件不能将 SWDIO、SWCLK 对应的 GPIO 设置为输出，否则会导致 SWD 通讯失效，即仅当 SWDIO、SWCLK 对应的 GPIO 为输入口时 SWD 才可正常使用。用户在程序发布时应注意 SWDIO、SWCLK 对应的 GPIO 端口的处理方式，避免输入悬空。

建议用户在产品量产时将配置字控制位 CFG_SWID 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。

22.3.2 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