

1 简介

本参考手册是对 SC32L14T/14G 系列数据手册的补充，提供了应用（特别是软件开发）所需的信息，有关特定 SC32L14T/14G 器件的功能集、订购信息以及机械和电气特征的信息，请参见其相应的数据手册。

Preliminary

目录

1	简介	1
	目录	2
2	文档约定	11
2.1	名词解释	11
2.2	外设可用性	11
3	资源框图	12
4	存储	13
4.1	概述	13
4.2	存储框图	13
4.3	特性	14
4.4	APROM (主存储区)	14
4.5	2 Kbytes 用户存储区域 (类 EEPROM)	15
4.6	4 Kbytes LDROM (系统存储区)	15
4.6.1	BootLoader	15
4.7	SRAM	16
4.8	启动区域选择 (自举)	16
4.8.1	从主存储区自举	16
4.8.2	从系统存储器自举	16
4.8.3	从嵌入式 SRAM 自举	16
4.8.4	自举模式设置	16
4.9	96 bits Unique ID	17
4.10	User ID 区域	17
4.11	编程	17
4.11.1	JTAG 专用模式	17
4.11.2	常规模式 (JTAG 专用口无效)	18
4.12	安全加密	18
4.12.1	安全加密操作权限	18
4.13	In Application Programming (IAP)	18
4.13.1	IAP 操作相关寄存器表	19
4.13.2	IAP 寄存器映射	21
4.14	选项字节区域 (Customer Option)	21
4.14.1	Customer Option 的映射寄存器	22
5	上电、复位和时钟控制 (RCC)	24
5.1	上电过程	24

5.1.1	复位阶段	24
5.1.2	调入信息阶段	24
5.1.3	正常操作阶段	24
5.2	复位	24
5.2.1	硬件复位	25
5.2.2	软件复位	26
5.2.3	复位后的启动区域	26
5.2.4	复位初始状态	26
5.3	时钟	26
5.3.1	系统时钟源	26
5.3.2	内建高频 48MHz 振荡器 (HIRC)	27
5.3.3	内置高频晶体振荡器电路, 可外接 2~16MHz 高频振荡器 (HXT)	27
5.3.4	内建低频 32kHz 振荡器 (LIRC)	27
5.3.5	内置低频振荡电路, 可外接 32.768kHz 低频振荡器 (LXT)	27
5.4	寄存器访问时钟和外设时钟源	28
5.4.1	寄存器访问时钟	28
5.4.2	外设时钟源	28
5.4.3	总线时钟	28
5.4.4	外设时钟	30
5.4.5	外设时钟框图	31
5.4.6	外设时钟使能寄存器	31
5.4.7	低功耗模式时钟	32
5.5	RCC 寄存器	32
5.5.1	RCC 相关寄存器表	32
5.5.2	RCC 寄存器映射	44
6	中断	45
6.1	外部中断 INT0~15	45
6.2	中断与事件	45
6.3	中断向量表	45
6.4	外部中断寄存器	47
6.4.1	外部中断相关寄存器表	47
6.4.2	外部中断寄存器映射	51
7	省电模式	52
8	低电压监测模块 (LPD)	53
8.1	概述	53
8.2	迟滞功能	53

8.3	LPD 中断	54
8.4	LPD 寄存器	54
8.4.1	LPD 相关寄存器表	54
8.4.2	LPD 寄存器映射	55
9	GPIO	56
9.1	时钟源	56
9.2	特性	56
9.3	GPIO 结构图	56
9.3.1	强推挽输出模式	56
9.3.2	带上拉的输入模式	57
9.3.3	高阻输入模式(Input only)	57
9.4	GPIO 寄存器	57
9.4.1	GPIO 相关寄存器表	57
9.4.2	GPIO 寄存器映射	60
10	模数转换器 ADC	62
10.1	时钟源	62
10.2	特性	62
10.3	ADC 转换步骤	62
10.4	ADC 中断	63
10.5	ADC 连接电路图	63
10.6	ADC 寄存器	63
10.6.1	ADC 相关寄存器表	63
10.6.2	ADC 寄存器映射	66
11	温度传感器	67
11.1	概述	67
11.2	温度传感器操作步骤	67
11.3	温度传感器寄存器	67
11.3.1	温度传感器相关寄存器表	67
11.3.2	温度传感器寄存器映射	68
12	运放及可编程增益放大器 (OP)	69
12.1	概述	69
12.2	特性	69
12.3	OP 端口选择	69
12.3.1	OP 精度调整	69
12.3.2	OP 同相端输入选择	69
12.3.3	OP 反相端输入选择	69
12.3.4	OP 输出选择	70

12.4	OP 电路结构框图	70
12.5	OP 寄存器	70
12.5.1	OP 相关寄存器表	70
12.5.2	OP 寄存器映射	71
13	模拟比较器 CMP	72
13.1	特性	72
13.2	模拟比较器结构框图	72
13.3	CMP 寄存器	73
13.3.1	CMP 相关寄存器表	73
13.3.2	CMP 寄存器映射	74
14	高级加密运算 (AES)	75
14.1	概述	75
14.2	时钟源	75
14.3	特性	75
14.4	操作步骤	75
14.5	AES 中断	75
14.6	AES 寄存器	76
14.6.1	AES 相关寄存器表	76
14.6.2	AES 寄存器映射	77
15	真随机数发生器 (TRNG)	78
15.1	概述	78
15.2	时钟源	78
15.3	特性	78
15.4	TRNG 操作步骤	78
15.5	TRNG 中断	78
15.6	TRNG 熵源验证	78
15.6.1	简介	78
15.6.2	验证条件	79
15.7	TRNG 寄存器	79
15.7.1	TRNG 相关寄存器表	79
15.7.2	TRNG 寄存器映射	80
16	UART0~5	81
16.1	时钟源	81
16.2	特性	81
16.3	UART2-LIN	82
16.3.1	LIN 帧结构	82
16.3.2	LIN 主机模式	82

16.3.3	LIN 从机模式	82
16.4	UART 数据匹配功能	83
16.4.1	配置说明	83
16.5	UART 中断	84
16.6	UART0/1/3/4/5 寄存器	84
16.6.1	UART0/1/3/4/5 相关寄存器表	84
16.6.2	UART0/1/3/4/5 寄存器映射	89
16.7	UART2 寄存器	90
16.7.1	UART2 相关寄存器表	90
16.7.2	UART2 寄存器映射	96
17	智能卡接口 (SC)	97
17.1	概述	97
17.2	时钟源	97
17.3	特性	97
17.4	工作模式	97
17.4.1	智能卡描述	97
17.4.2	智能卡数据传输	100
17.4.3	初始化字符 TS	100
17.4.4	错误信号和字符重复	101
17.4.5	扩展保护时间	101
17.5	智能卡接口基本操作流程	102
17.6	智能卡接口操作步骤	102
17.6.1	SC 接口底层驱动配置初始化	102
17.6.2	插卡检测	103
17.6.3	激活和冷复位	103
17.6.4	数据传输	103
17.6.5	释放	104
17.7	智能卡接口中断	104
17.8	SC 寄存器	104
17.8.1	SC 相关寄存器表	104
17.8.2	SC 寄存器映射	108
18	SPI0/1	109
18.1	时钟源	109
18.2	SPI0/1 特性	109
18.3	信号描述	109
18.4	工作模式	109
18.4.1	主模式	110

18.4.2	从模式.....	110
18.5	传送形式.....	111
18.6	出错检测.....	111
18.7	SPI0/1 寄存器.....	112
18.7.1	SPI0/1 相关寄存器表.....	112
18.7.2	SPI0/1 寄存器映射.....	115
19	TWI0/1.....	116
19.1	时钟源.....	116
19.2	TWI0/1 特性.....	116
19.3	TWI 信号描述.....	116
19.4	从机工作模式.....	116
19.5	从机模式操作步骤.....	119
19.6	主机工作模式.....	119
19.7	主机模式操作步骤.....	121
19.8	TWI0/1 中断.....	121
19.9	TWI0/1 寄存器.....	122
19.9.1	TWI0/1 相关寄存器表.....	122
19.9.2	TWI0/1 寄存器映射.....	126
20	硬件看门狗 WDT.....	128
20.1	概述.....	128
20.2	时钟源.....	128
20.3	WDT 寄存器.....	128
20.3.1	WDT 相关寄存器表.....	128
20.3.2	WDT 寄存器映射.....	129
21	BTM.....	130
21.1	概述.....	130
21.2	时钟源.....	130
21.3	特性.....	130
21.4	BTM 中断.....	130
21.5	BTM 寄存器.....	131
21.5.1	BTM 相关寄存器表.....	131
21.5.2	BTM 寄存器映射.....	132
22	内建 CRC 校验模块.....	133
22.1	概述.....	133
22.2	时钟源.....	133
22.3	特性.....	133
22.4	CRC 寄存器.....	134

22.4.1	CRC 相关寄存器表	134
22.4.2	CRC 寄存器映射	136
23	PWM0: 8 路 16 位多功能 PWM	137
23.1	概述	137
23.2	时钟源	137
23.3	特性	137
23.4	PWM0 结构框图	138
23.5	PWM0 通用配置	138
23.5.1	输出模式	138
23.5.2	对齐类型	139
23.5.3	占空比变化特性	140
23.5.4	周期变化特性	140
23.5.5	周期和占空比的关系	141
23.6	PWM0 独立模式	141
23.7	PWM0 互补模式	142
23.7.1	PWM0 互补模式死区时间设置	143
23.7.2	PWM0 死区输出波形	143
23.8	PWM0 故障检测功能设置	143
23.9	PWM0 中断	144
23.10	PWM0 寄存器	144
23.10.1	PWM0 相关寄存器表	144
23.10.2	PWM0 寄存器映射	149
24	32 路高灵敏度触控电路 (TK)	150
25	实时时钟 (RTC)	151
25.1	概述	151
25.2	时钟源	151
25.3	特性	151
25.4	结构框图	152
25.5	功能说明	152
25.5.1	RTC 时钟	152
25.5.2	实时时钟和日历	152
25.5.3	可编程闹钟	153
25.5.4	固定周期中断	153
25.5.5	RTC 初始化和配置	153
25.5.6	低功耗应用	155
25.6	RTC 中断	155

25.7	RTC 寄存器	156
25.7.1	RTC 相关寄存器表	156
25.7.2	RTC 寄存器映射	161
26	低功耗计数器 (LPC)	162
26.1	概述	162
26.2	特性	162
26.3	LPC 框图	163
26.4	方向判定	163
26.5	计数方式	163
26.6	LPC 中断	164
26.7	LPC 寄存器	165
26.7.1	LPC 相关寄存器表	165
26.7.2	LPC 寄存器映射	169
27	16 位定时/计数器 (TIM) TIMER0~TIMER7	171
27.1	时钟源	171
27.2	特性	171
27.3	计数方式	171
27.3.1	定时器模式下 TIM 计数方式	171
27.3.2	PWM 输出模式下 TIM 计数方式	171
27.4	定时器相关的信号口	171
27.5	TIM 的中断及对应标志位	172
27.6	TIM 的工作模式	172
27.6.1	工作模式 0: 16 位捕获	172
27.6.2	工作模式 1: 16 位自动重载定时器/计数器模式	173
27.6.3	工作模式 3: 可编程时钟输出	173
27.6.4	工作模式 4: PWM 输出模式	173
27.7	TIM 中断	174
27.8	TIM 寄存器	174
27.8.1	TIM 相关寄存器表	174
27.8.2	TIM 寄存器映射	181
28	LCD 驱动器	185
28.1	时钟源	185
28.2	内置 8 COM x 51 SEG LCD 驱动	185
28.2.1	电阻型 LCD 驱动器	185
28.2.2	电容偏压型 LCD 驱动器	186
28.3	LCD 驱动波形	187
28.3.1	1/3 BIAS 1/4 duty	187

28.3.2	1/4 BIAS 1/4 duty	188
28.4	LCD 寄存器	189
28.4.1	LCD 相关寄存器表	189
28.4.2	LCD 寄存器映射	193
29	DMA 控制器	196
29.1	概述	196
29.2	时钟源	196
29.3	特性	196
29.4	功能说明	196
29.4.1	传输方向	196
29.4.2	DMA 访问区域限制	196
29.4.3	通道优先级	196
29.4.4	单次传输和批量传输	197
29.4.5	循环模式	197
29.5	DMA 中断	197
29.6	DMA 寄存器	198
29.6.1	DMA 相关寄存器表	198
29.6.2	DMA 寄存器映射	203
30	SYSTICK	204
30.1	时钟源	204
30.2	SysTick 校准寄存器默认值	204
31	版本记录	205
32	声明	206

2 文档约定

2.1 名词解释

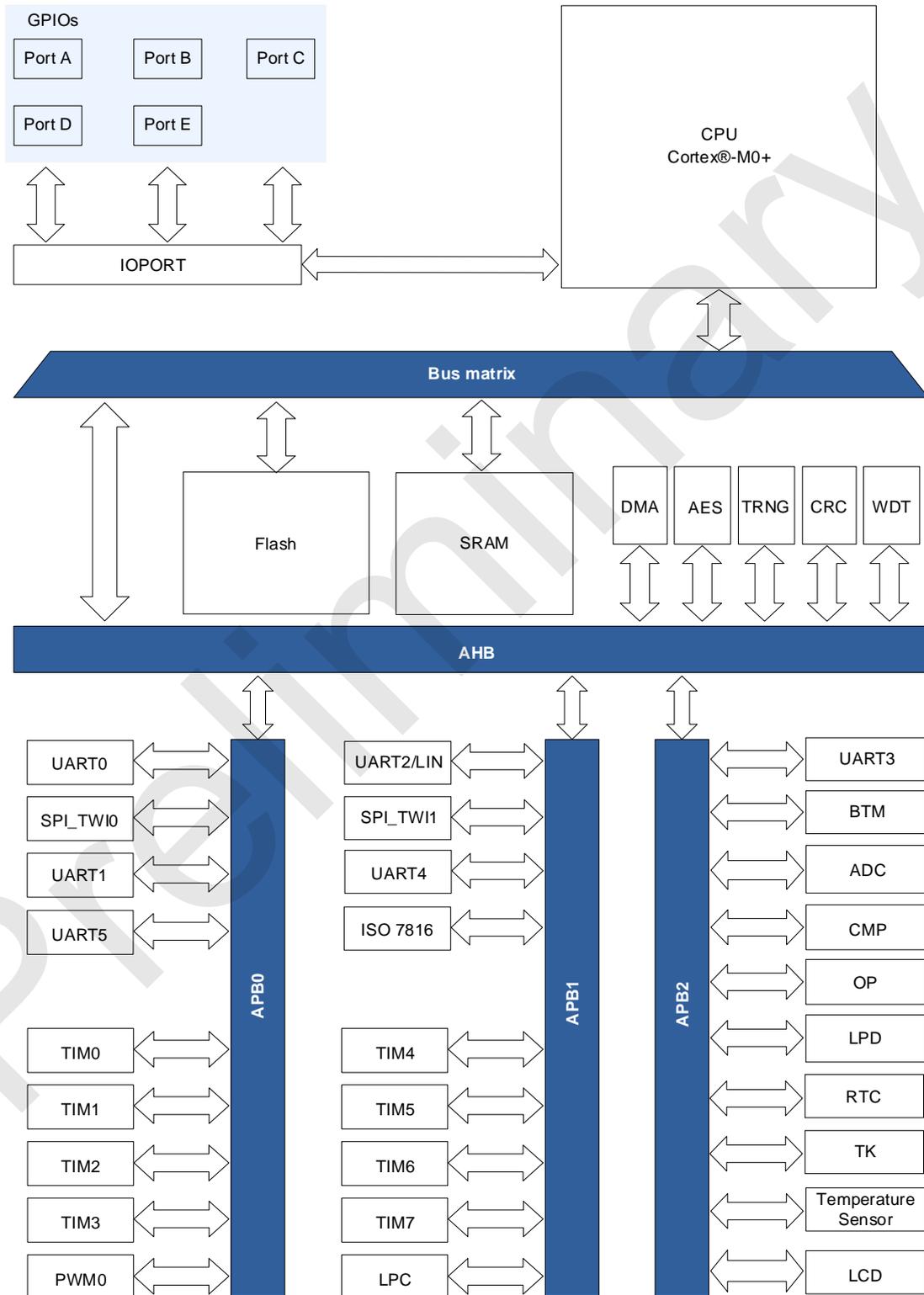
本节主要解释本文档中各首字母缩略词和缩写词的定义：

- 字：32 位数据
- 半字：16 位数据
- 字节：8 位数据
- 双字：64 位数据
- IAP(在应用中编程)：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程
- ICP(在线编程)：ICP 是指可以在器件安装于用户电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程
- JTAG 协议：JTAG 协议是一种国际标准测试协议，主要用于芯片内部测试
- SWD 协议：SWD 协议是 ARM 设计的协议，代表串行线测试，用于对其微控制器进行编程和调试
- 选项字节：存储于 Flash 中的产品配置位
- AHB：高级高性能总线
- APB：高级外设总线

2.2 外设可用性

有关各型号产品的外设可用性及数量信息，请参考各产品最新的数据手册的产品外设资源表章节。

3 资源框图

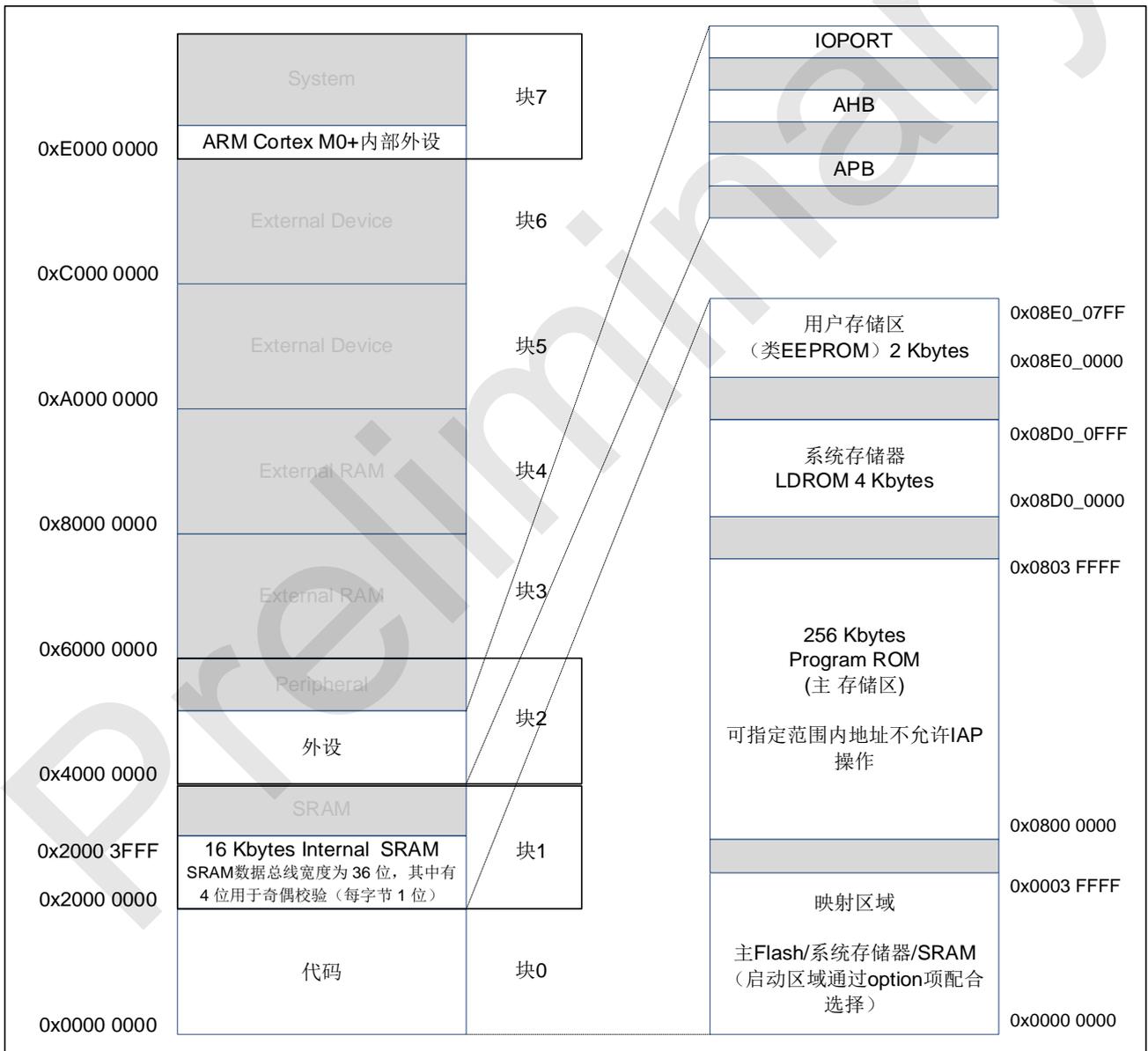


4 存储

4.1 概述

程序存储器、数据存储器、寄存器排列在同一个线性（即地址连续）的 4 GB 地址空间内。各字节按小端格式在存储器中编码。字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。可寻址的存储空间分为 8 个主要块，每个块为 512 MB。

4.2 存储框图



SC32L14T/14G 存储器映射图

4.3 特性

- 存储 Flash 位宽为 32 bits，可反复写入 10 万次
- 常温 25°C 下数据可保存 100 年
- Flash 的组成结构如下：
 - 最大 256 Kbytes APROM（主存储区）
 - 4 Kbytes LDROM（系统存储区）
 - 2 Kbytes 用户存储区（类 EEPROM）
 - 16 Kbytes Internal SRAM，支持奇偶校验功能
 - 96 bits Unique ID

4.4 APROM（主存储区）

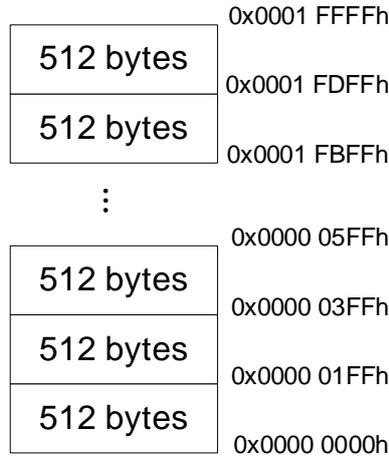
- SC32L14xx8 系列的主存储区（APROM）有 256 Kbytes
- SC32L14xx7 系列的主存储区（APROM）有 128 Kbytes
- 扇区（sector）大小：512 bytes
- 支持：读/写/扇区擦除/全擦除/查空
- CPU (Cortex[®]-M0+) 通过 AHB 总线访问 Flash
- 程序默认从主存储区启动，用户可通过 customer option OP_BL[1:0]选择程序从 SRAM/LDROM 等其它区域启动
- 读保护：即读取加密，读保护开启后，仅可从主存储区启动运行的程序可以读取主存储区信息，其它区域启动或第三方工具无法获取主存储区信息
- 写保护：提供两段硬件写保护区域，写保护区域中禁止 IAP 操作，用户可以根据实际需求以扇区为单位设定两段写保护区域的范围

256 Kbytes 的 Flash ROM 分为 512 个扇区（sector），每个 sector 为 512 bytes，烧录时目标地址所属的 Sector 都会被烧写器强制擦除，再写入数据；用户写操作时，必须先擦除，再写入数据。



SC32L14T 256 Kbytes Flash ROM Sector 分区示意

128 Kbytes 的 APROM 分为 256 个扇区（sector），每个 Sector 为 512 bytes，烧录时目标地址所属的 Sector 都会被烧写器强制擦除，再写入数据；用户写操作时，必须先擦除，再写入数据。

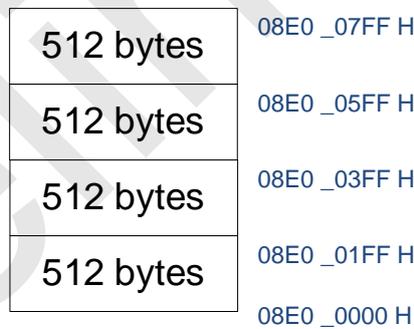


SC32L14xx7 系列 128 Kbytes APROM Sector 分区示意

4.5 2 Kbytes 用户存储区域（类 EEPROM）

2 Kbytes 独立 EEPROM 区域地址为 0x08E0_0000 H~ 0x08E0_07FF H，由 IAPADE 寄存器设定。独立 EEPROM 可反复写入 10 万次，常温下数据保存时间为 100 年以上。独立 EEPROM 支持查空、编程、校验、擦除和读取功能。

EEPROM 共有 4 个扇区（sector），每个 sector 为 512 bytes:



SC32L14T EEPROM Sector 分区示意

注意：EEPROM 擦写次数为 10 万次，用户擦写不要超过 EEPROM 的额定烧写次数，否则会出现异常！

4.6 4 Kbytes LDROM（系统存储区）

- 系统存储区为 4 Kbytes LDROM，出厂固化 ISP 程序，该区域用户无法改写也无法访问
- 嵌入式自举程序：固化的 ISP 程序，该指令公开，可以利用该程序通过 UART 更新程序 Flash。程序执行会等待升级命令，在 500ms 内未来接收到更新指令，则跳转到主存储区执行（0X0800_0000）

4.6.1 BootLoader

支持硬件和软件两种 BootLoad 方式：

- 软件做法：直接软件划分 BootLoad 和 APP 区域，通过修改 VTOR 可轻易实现 BootLoad、APP 中

- 断共用，各区大小可随意裁切；
- 硬件做法：4 Kbytes 固定“系统存储器”为专用 BootLoader 区域，用户不可读写：
 - 系统存储区作为一个固化的 BootLoader 空间，其中的程序是出厂前烧录好，用户不可读写。
 - 嵌入式自举程序：嵌入式自举程序位于系统存储器中，在生产阶段编程。固化的 ISP 程序，该指令公开，可以利用该程序通过 UART 重新编程 Flash。

4.7 SRAM

- Internal SRAM: 16 Kbytes, 地址 0x2000_0000 ~ 0x2000 3FFF
- 支持奇偶校验：
 - 额外的 2 Kbytes RAM 用于奇偶校验：即 SRAM 数据总线宽度为 36 位，其中有 4 位用于奇偶校验（每字节 1 位）
 - 奇偶校验位在写入 SRAM 时进行计算和保存，在读取时自动进行校验。如果某一位失败，则将生成不可屏蔽中断（Cortex®-M0+ NMI）
 - 提供独立的 SRAM 奇偶校验错误标志 SRAMPEIF。

注意：当启用 SRAM 奇偶校验时，建议在代码开始处使用软件初始化整个 SRAM，以免在读取非初始化位置时出现奇偶校验错误。

- 用户可通过 customer option OP_BL[1:0]选择程序从 SRAM 启动
- 能够以最大系统时钟频率按字节、半字（16 位）或全字（32 位）访问，无等待状态，因此可由 CPU 和 DMA 访问。

4.8 启动区域选择（自举）

复位后，用户可自行设置所需的自举模式配置。

退出待机模式后，还可以对启动模式配置进行重新采样。该启动延迟结束后，CPU 将从地址 0x0000_0000 获取栈顶值，然后从始于 0x0000_0004 的自举存储器开始执行代码。

自举区域选择有三种：主存储区、系统存储区和 SRAM，详细描述如下：

4.8.1 从主存储区自举

主存储区在自举存储器空间 (0x0000_0000) 中有别名，但也可从它原来的存储器空间 (0x0800_0000) 访问。换句话说：程序可从地址 0x0000_0000 或 0x0800_0000 开始访问。

4.8.2 从系统存储器自举

- 系统存储区（4 Kbytes 独立 LDR0M）作为一个固化的 BootLoader 空间，其中的程序是出厂前烧录好，用户不可读写。
- 嵌入式自举程序：嵌入式自举程序位于系统存储器中，在生产阶段编程。固化的 ISP 程序，该指令公开，可以利用该程序通过 UART 重新编程 Flash。

4.8.3 从嵌入式 SRAM 自举

SRAM 在自举存储器空间 (0x0000_0000) 中有别名，但也可从它原来的存储器空间 (0x2000_0000) 访问。

4.8.4 自举模式设置

通过寄存器控制位 BTLD[1:0]配合软件复位 RST 控制位可实现三种自举模式，BTLD 和 RST 受 IAP_KEY 保护：

- ① 设置 BTLD[1:0]= 00，芯片软件复位后从主存储区（APROM）启动

- ② 设置 BTLD[1:0]= 01，芯片软件复位后从系统存储区（LDROM）启动
- ③ 设置 BTLD[1:0]= 10，芯片软件复位后从嵌入式 SRAM 启动

通过 customer option 项 OP_BL[1:0]实现芯片上电初始启动区域选择：

- ① 在 customer option 中设置 OP_BL[1:0]= 00，芯片复位后从主存储区（APROM）启动
- ② 在 customer option 中设置 OP_BL[1:0]= 01，芯片复位后从系统存储区（LDROM）启动
- ③ 在 customer option 中设置 OP_BL[1:0]= 10，芯片复位后从嵌入式 SRAM 启动

4.9 96 bits Unique ID

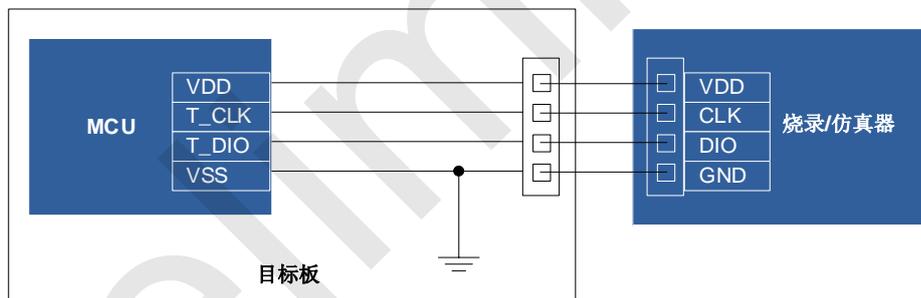
SC32L14T 提供了一个独立的 Unique ID 区域，出厂前会预烧一个 96 bits 的唯一码，用以确保该芯片的唯一性。用户获得序列号的唯一方式是通过 IAP 指令读取。

4.10 User ID 区域

User ID 区域，出厂时写入用户定制 ID，用户可对其进行读操作，但禁止对 User ID 区域进行写操作。

4.11 编程

SC32L14T/14G 的 Flash 通过 T_DIO、T_CLK、VDD、VSS 来进行编程，具体连接关系如下：



ICP 模式 Flash Writer 编程连接示意图

T_DIO、T_CLK 是 2 线 JTAG 烧写和仿真的信号线，用户在烧录时可通过 Customer Option 项配置这两端口的模式：JTAG 专用模式和常规模式（JTAG 专用口无效）。

注意：UART3 的信号引脚支持两种映射方案

- 映射 1：RXD3 / TXD3
- 映射 2：RXD3A / TXD3A

当选择映射 1（RXD3/TXD3）时，该组引脚与系统烧录/调试接口（T_CLK / T_DIO）复用。在此映射下，若启用全双工通信，T_CLK/T_DIO 可能会与 UART3 的 RXD3 接收时序产生冲突，导致通信异常。因此，选择映射 1 时必须将 UART3 配置为半双工通信模式，以避免此硬件冲突并保证通信稳定性。

如需使用全双工 UART 通信，请将管脚映射至映射 2（RXD3A/TXD3A）。

4.11.1 JTAG 专用模式

JTAG 专用模式下，T_DIO、T_CLK 为烧写仿真专用口，与之复用的其它功能不可用。此模式一般用于在线调试阶段，方便用户仿真调试；JTAG 专用模式生效后，芯片无需重新上下电即可直接进入烧录或仿真模式。

4.11.2 常规模式（JTAG 专用口无效）

常规模式下，JTAG 功能不可用，端口上与之复用的其它功能可正常使用。此模式可防止烧录口占用 MCU 管脚，方便用户最大化利用 MCU 资源。

注意：当 JTAG 专用口无效的配置设定成功后，芯片必须彻底下电再重新上电后才能进入烧录或仿真模式，这样就会影响到带电模式下的烧录和仿真。赛元建议用户在量产烧录时选择 JTAG 专用口无效的配置，在研发调试阶段选择 JTAG 模式。

相关 Customer Option 如下：

寄存器	读/写	说明	复位值
COPT1_CFG@0xC2	读/写	Customer Option 映射寄存器 1	0x0000_0000

7	6	5	4	3	2	1	0
ENWDT	DISJTG	DISRST	-	-	-	OP_BL[1:0]	

位编号	位符号	说明
6	DISJTG	JTAG 口切换控制位 0: JTAG 模式使能，对应管脚只能作为 T_CLK/ T_DIO 使用 1: 常规模式（Normal），JTAG 功能无效

4.12 安全加密

SC32L14T/14G 系列的安全加密功能主要是对 APROM 进行读保护加密：用户可在烧录阶段通过赛元专用烧录套件的烧录上位机的 customer option 项配置读保护加密功能，选择是否开启 flash 读保护，进入加密模式：

- 芯片默认出厂时 flash 为非加密状态
- 读保护加密功能无映射寄存器，用户只能在烧录阶段通过赛元专用烧录套件的烧录上位机的 customer option 项配置，必须经过烧录才能完成修改。
- 加密失能：可对主存储区执行读取、编程和擦除操作。也可对选项字节和备份寄存器进行所有操作。
- 加密使能：
 - 主存储区启动：在用户模式下执行的代码（从用户 APROM 自举）可对主存储区执行所有操作。
 - 调试、从 SRAM 启动以及从系统存储区启动：在调试模式下或当代码从 SRAM 或系统存储区启动时，主存储区完全不可访问。
- 取消加密使能必须先对主存储区进行全擦操作。

4.12.1 安全加密操作权限

启动区域/工具	解锁状态					读保护加密状态				
	读	写	块擦	全擦	操作写保护区域	读	写	块擦	全擦	操作写保护区域
从 APROM 自举	✓	✓	✓	\	禁止	✓	✓	✓	\	禁止
调试/从 SRAM 自举	✓	✓	✓	✓	禁止	禁止	禁止	禁止	禁止	禁止
从系统存储区自举	✓	✓	✓	✓	✓	禁止	禁止	禁止	✓	禁止

4.13 In Application Programming (IAP)

SC32L14T/14G 的 APROM 中的 IAP 区域可进行 In Application Programming (IAP)操作，用户可以通过 IAP 操作实现远程程序更新，也可以通过 IAP 读操作获取 Unique ID 区域或 User ID 区域信息。进行 IAP 写数据操作前，用户必须对目标地址所属的 Sector 进行扇区擦除操作。

芯片默认出厂时 APROM 允许全局 IAP 操作。芯片内部提供两组 flash 写保护区域，按照扇区单位设置起始，被保护的区域禁止 IAP 操作，设置规则如下：

IAPPORx 寄存器值 (x=A 或 B)	IAPPOR 保护区域
IAPPORx_ST = IAPPORx_ED	扇区 IAPPORx
IAPPORx_ST > IAPPORx_ED	无 (不受保护)
IAPPORx_ST < IAPPORx_ED	从 IAPPORx_ST 到 IAPPORx_ED 的扇区

用户在烧录时可通过 Customer Option 项里的“Flash sectors protection”配置这两段 APROM 写保护区域。

注意：IAP 不支持字节/半字编程，即：IAP 写入时必须按字对齐（4 字节对齐）；若按字节/半字写入，都会自动重复写入字节补齐至字写入，例：写入 0x12，将自动补齐为 0x12121212 写入；写入 0x1234，将自动补齐为 0x12341234 写入。

4.13.1 IAP 操作相关寄存器表

对写保护区域之外的 APROM 进行 IAP 操作，可通过以下寄存器实现：

4.13.1.1 数据保护寄存器 IAP_KEY

寄存器	读/写	说明	复位值
IAP_KEY	读/写	数据保护寄存器	0x0000_0000

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

位编号	位符号	说明
31~0	IAPKEY[31:0]	数据保护锁 为防止因电气干扰等原因出现对 Flash 的意外操作，IAP_CON 寄存器执行写操作前需要通过 IAPKEY 解锁。解锁顺序如下： 1. 写入 KEY1 = 0x1234_5678 2. 写入 KEY2 = 0xA05F_05FA 如果操作顺序不正确，会锁定 IAP_CON 寄存器，直到下次系统复位才会解锁。

4.13.1.2 IAP 扇区编号设置寄存器 IAP_SNB

寄存器	读/写	说明	复位值
IAP_SNB	读/写	IAP 扇区编号设置寄存器	0x0000_0000

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

位编号	位符号	说明
31~24	IAPADE[7:0]	IAP 操作区域扩展地址 通过向 IAPADE 写入不同的值，使得 IAP 操作指向不同的操作区域： 0x00: 无效 0x4C: APROM 0x69: EEPROM 0xF1: customer option
9~0	IAPSNB[9:0]	扇擦/页擦时的 IAP 操作扇区编号设置： 实际操作扇区的起始地址= Flash 基址 + [IAPSNB[9:0] x 0x200]
23~10	-	保留

4.13.1.3 IAP 控制寄存器 IAP_CON (写保护)

*该寄存器受写保护，必须操作数据保护寄存器 IAP_KEY 才能修改。

寄存器	读/写	说明	复位值
IAP_CON	读/写	IAP 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
LOCK	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	CONT[5:0]					
15	14	13	12	11	10	9	8
-	-	-	-	DMAEN	BTLD[1:0]		RST
7	6	5	4	3	2	1	0
ERASE	-	SERASE	PRG	-	-	CMD[1:0]	

位编号	位符号	说明
31	LOCK	对该位写 1 后，IAP_CON 寄存器被锁定。当检测到解锁序列时，由硬件将该位清 0。 如果解锁操作失败，该位仍保持置 1，直到下一次系统复位。
21~16	CONT[5:0]	IAP 写数据加速设置位 以 4bytes 连续地址为单位执行连续的 IAP 写操作，通过 CONT[5:0] 设置连续烧录长度，有效范围 0x01 ~ 0x20，即一次最多可连续烧录 32bits*32=128bytes 操作步骤： 1.待烧录的数据暂存在 RAM 区域，RAM buffer 最大 128bytes; 2.解锁 IAP_KEY，使能 IAP_CON.PRGM。 3.使能 IAP_CON.DMAEN，设置 CONT[5:0]; 4.配置 DMA，选择其中一个空闲通道，通道源地址指向的 RAM buffer，目标地址指向待烧录区域的起始地址，注意烧录目标区域起始地址要 4bytes 对齐； 5.配置 DMA 通道，TPTYPE=1，选择批量模式，设置 TXWIDTH[1:0]=32bit，DMACNT[31:0]=CONT[5:0]; 6.触发 DMA 通道的软件搬运，SWREQ=1，开始执行连续烧录。
11	DMAEN	DMA 协助连续编程控制位 0: 关闭 DMA 协助连续编程功能 1: 开启 DMA 协助连续编程功能
10~9	BTLD[1:0]	软件复位后程序启动区域选择位： 00: 芯片软件复位后从 APROM (主存储区) 启动 01: 芯片软件复位后从 LDRM (系统存储区) 启动

位编号	位符号	说明
		10: 芯片软件复位后从嵌入式 SRAM 启动 11: 保留
8	RST	软件复位控制位 0: 程序正常运行 1: 对该位写 1 后系统立刻复位
7	ERASE	全擦 (All Erase) 控制位 0: 无操作 1: 对该位写 1 后再配置 CMD[1:0]=10, 则进入 APROM 全擦除操作, APROM 将全部擦除
5	SERASE	扇区擦除 (Sector Erase) 控制位 0: 无操作 1: 对该位写 1 后再配置 CMD[1:0]=10, 则进入 APROM 扇区擦除操作, APROM 的指定 Sector 将被擦除
4	PRG	编程 (Program) 控制位 0: 禁止 Flash 编程 1: 使能 Flash 编程
1~0	CMD[1:0]	IAP 命令使能控制位 10: 执行擦除操作命令 其它: 保留 注意: 1. 擦除操作命令位写 1 后, 必须配置 CMD[1:0]=10, 相应的操作才会开始执行 2. 一次只能执行 1 种 IAP 操作, 所以 ERASE/SERASE 位同一时间只能有一位置 1
30~11 6 3~2	-	保留

4.13.2 IAP 寄存器映射

寄存器	偏移地址	读/写	说明	复位值
IAP 基地址: 0x4000_03C0				
IAP_KEY	0x00	读/写	数据保护寄存器	0x0000_0000
IAP_SNB	0x04	读/写	IAP 扇区编号设置寄存器	0x0000_0000
IAP_CON	0x0C	读/写	IAP 控制寄存器	0x0000_0000

4.14 选项字节区域 (Customer Option)

SC32L14T/14G 有单独的一块 Flash 区域用于保存客户的上电初始值设置, 此区域称为选项字节区域 (Customer Option) 区域。用户在烧录时可通过上位机对 Customer Option 项进行配置, 在烧写过程中将配置值写入 Customer Option 区域, IC 在复位初始化阶段将调用 Customer Option 数据作为初始设置。

也可通过操作 Customer Option 的映射寄存器临时修改 Customer Option 项, 但是需要注意: 修改映射寄存器仅可实现临时调整, 不会对 Customer Option 区域的设置值造成任何影响, 芯片复位后, 仍会根据烧录时用户选择的 Customer Option 参数进行初始化。

Customer Option 相关映射寄存器的操作方式如下:

Customer Option 相关 SFR 的读写操作由 OPINX 和 OPREG 两个寄存器进行控制，各 Customer Option SFR 的具体位置由 OPINX 确定，如下表所示：

寄存器	地址	说明	复位值	上电初始值
OPINX	0x4000_03F8	Customer Option 指针	0x0000_0000	0x0000_0000
OPREG	0x4000_03FC	Customer Option 寄存器	0x0000_0000	0x0000_0000
COPT0_CFG	0XC1 @ OPINX	Customer Option 映射寄存器 0	0x0000_0000	0x0000_0000
COPT1_CFG	0XC2 @ OPINX	Customer Option 映射寄存器 1	0x0000_0000	0x0000_0000

4.14.1 Customer Option 的映射寄存器

使用 OPINX 配合 OPREG 改写 IFB 映射寄存器之前，应先要打开 Customer Option 寄存器的时钟使能开关 AHB_CFG.IFBEN：

4.14.1.1 AHB 总线外设时钟使能寄存器 AHB_CFG

寄存器	地址	说明	复位值	上电初始值
AHB_CFG	读/写	AHB 总线外设时钟使能寄存器	0x0010_0000	0x0010_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	CLKDIV[2:0]			-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	IFBEN	CRCEN	DMAEN

位编号	位符号	说明
2	IFBEN	Customer Option 映射寄存器时钟使能位 使用 OPINX 配合 OPREG 改写 IFB 映射寄存器之前，应先要打开时钟使能。 0：禁止 1：使能

4.14.1.2 Customer Option 映射寄存器 0 COPT0_CFG

寄存器	地址	说明	复位值	上电初始值
COPT0_CFG	读/写	Customer Option 映射寄存器 0	0x0000_0000	0x0000_0000

7	6	5	4	3	2	1	0
HIRC_SDIV[1:0]		-	-	-	DISLVR	LVRS [1:0]	

位编号	位符号	说明
7~6	HIRC_SDIV[1:0]	HIRC 分频选择位 00：当 SYSCLKSW=0 时，系统时钟频率为 $f_{HIRC}/2$ 01：当 SYSCLKSW=0 时，系统时钟频率为 $f_{HIRC}/2$ 10：当 SYSCLKSW=0 时，系统时钟频率为 $f_{HIRC}/4$ 11：当 SYSCLKSW=0 时，系统时钟频率为 $f_{HIRC}/8$
2	DISLVR	LVR 开关 0：LVR 有效 1：LVR 无效

位编号	位符号	说明
1~0	LVRS [1:0]	LVR 电压选择控制 11: 4.3V 复位 10: 3.7V 复位 01: 2.9V 复位 00: 1.7V 复位
5~3	-	保留

4.14.1.3 Customer Option 映射寄存器 1 COPT1_CFG

寄存器	读/写	说明	复位值	上电初始值
COPT1_CFG	读/写	Customer Option 映射寄存器 1	0x0000_0000	0x0000_0000

7	6	5	4	3	2	1	0
ENWDT	DISJTG	DISRST	-	-	-	OP_BL[1:0]	

位编号	位符号	说明
7	ENWDT	WDT 开关 1: WDT 开始工作 0: WDT 关闭
6	DISJTG	JTAG 口切换控制位 0: JTAG 模式使能, 对应管脚只能作为 T_CLK/ T_DIO 使用 1: 常规模式 (Normal), JTAG 功能无效
5	DISRST	复位信号口切换控制位 该位只读, 用户不可改写。 0: RST 对应管脚当复位脚使用 1: RST 所在管脚做正常的 GPIO 管脚使用
1~0	OP_BL[1:0]	芯片复位后启动区域选择 该位只读, 用户不可改写。 00: 芯片复位后从 APROM (主存储区) 启动 01: 芯片复位后从 LDROM (系统存储区) 启动 10: 芯片复位后从嵌入式 SRAM 启动 11: 保留
4~2	-	保留

5 上电、复位和时钟控制（RCC）

5.1 上电过程

SC32L14T/14G 上电后，在客户端软件执行前，会经过以下三个阶段：

- ① 复位阶段；
- ② 调入信息阶段；
- ③ 正常操作阶段。

5.1.1 复位阶段

复位阶段是指 SC32L14T/14G 持续保持在复位状态，直到供应电压超过特定的阈值，其内部时钟机制才开始有效运作。

该阶段的具体时长受外部电源电压上升速率的影响，当外部电源提供的电压达到内建 POR 电压时，复位阶段即告结束。

5.1.2 调入信息阶段

在 SC32L14T/14G 内部设有一个预热计数器。复位阶段，此预热计数器持续清零，供电电压上升至 POR 后，内部 HIRC 振荡器预热计数器开始计数。该预热计数器计数到一定数值后，Flash ROM 中的 IFB（包含 Customer Option）数据将被周期性得读取至系统寄存器。所有 IFB 数据读取完毕后，预热及信息调入阶段结束，系统进入正常操作模式。

5.1.3 正常操作阶段

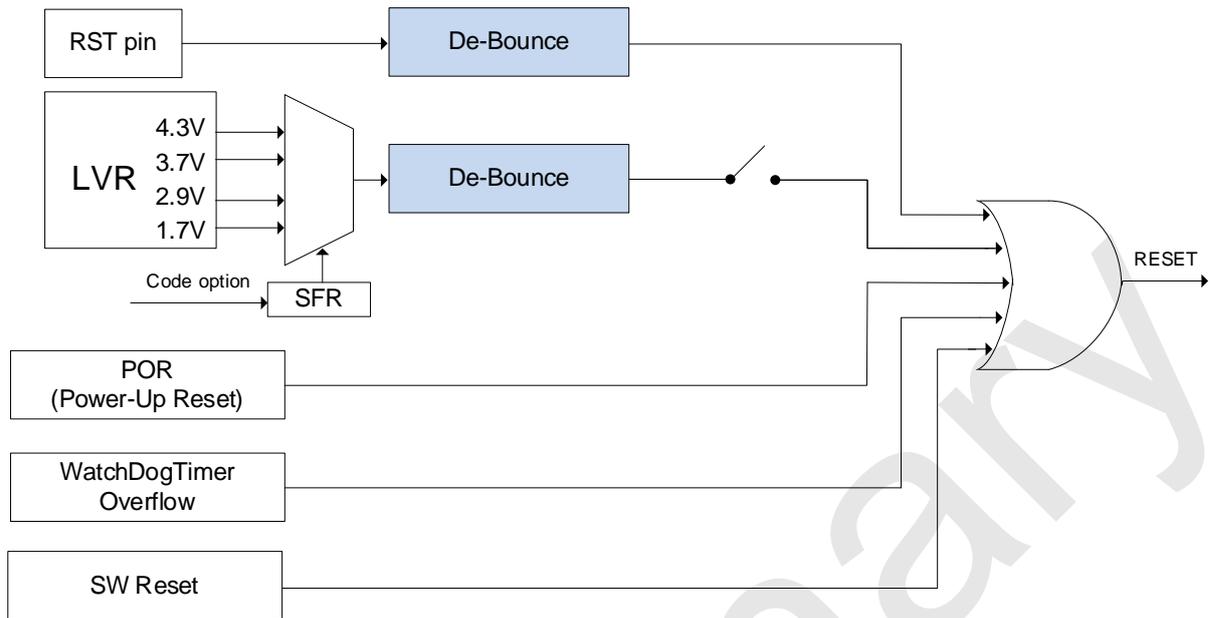
完成信息调入后，SC32L14T/14G 从 Flash 读取指令进入正常操作阶段。此时 LVR 电压值为用户写入 Customer Option 的设置值。

5.2 复位

SC32L14T/14G 有两类复位方式：硬件复位和软件复位；当复位发生后，全部寄存器都将恢复成其复位值。SC32L14T/14G 有 5 种复位方式，前四种为硬件复位：

- ① 外部 RST 复位
- ② 低电压复位 LVR
- ③ 上电复位 POR
- ④ 看门狗 WDT 复位
- ⑤ 软件复位

SC32L14T/14G 的复位部分电路结构图如下：



SC32L14T/14G 复位电路图

5.2.1 硬件复位

硬件复位可通过以下四种方式完成：

- ① 外部 RST 复位
- ② 低电压复位 LVR
- ③ 上电复位 POR
- ④ 看门狗 WDT 复位

5.2.1.1 外部 RST 复位

SC32L14T/14G 的外部 RST 复位可通过在外部 RST 引脚上输入一个脉宽大于 18us 的低电平复位脉冲信号实现。

用户在烧录程序前可通过烧录上位机软件配置 Customer Option 项将 PE5 / NRST 管脚配置为 RST（复位脚）使用。

- 当该管脚配置为复位功能时，该管脚不能作为 GPIO 功能使用。
- 当该管脚配置为 GPIO 功能时，该管脚的复位功能不可用。

5.2.1.2 低电压复位 LVR

SC32L14T/14G 内建一个低电压复位电路，支持 4 种门限电压选择：4.3V、3.7V、2.9V、1.7V。出厂时，默认的门限电压值为 1.7V，用户可通过烧录时设置 Customer Option 的值来设定缺省值。当 V_{DD} 电压低于设定的门限值，且持续时间超过约 30 μ s 的消抖时间 T_{LVR} 时，芯片将被复位。

5.2.1.3 上电复位 POR

SC32L14T/14G 内部有上电复位电路，当 V_{DD} 电压达到 POR 复位电压时，系统自动复位。

5.2.1.4 看门狗复位 WDT

SC32L14T/14G 有一个 WDT，其时钟源为内部的 32kHz 振荡器。用户可以通过编程器的 Customer Option 选择是否开启看门狗复位功能。当看门狗事件发生后，将发生看门狗复位。

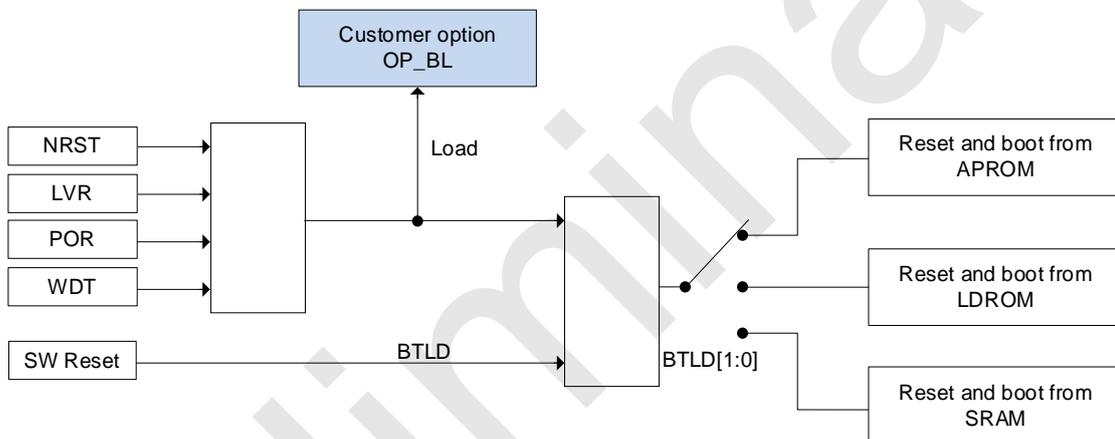
5.2.2 软件复位

SC32L14T/14G 提供软件复位功能，用户可将 RST (IAP_CON.8) 位置 1 对系统立刻进行软件复位。

5.2.3 复位后的启动区域

外部 RST 复位、低电压复位 LVR、上电复位 POR、看门狗 WDT 这四种硬件复位后，芯片从用户 OP_BL 设定的启动区域(APROM / LDROM / SRAM)启动。

软件复位后，芯片根据 BTLD[1:0]设定的启动区域(APROM / LDROM / SRAM)启动。



SC32L14T/14G 复位后启动区域切换示意图

5.2.4 复位初始状态

当 SC32L14T/14G 处于复位状态时，多数寄存器会回到其初始状态。看门狗 WDT 功能将在此阶段被禁用。“热启动”下的 Reset (如 WDT、LVR、软件复位等) 不会对 SRAM 中的数据产生影响，因此 SRAM 将保留复位前的值。

注意：当电源电压降低到无法维持 RAM 数据保存的程度时，SRAM 中的内容将会丢失。

5.3 时钟

5.3.1 系统时钟源

系统时钟 SYSCLK 可通过四种不同的时钟源驱动：

- 内建高频 48MHz 振荡器 (HIRC)
- 内建低频 32kHz 振荡器 (LIRC)
- 外接高频晶振 (HXT)
- 外接低频晶振 (LXT)

对于每个振荡器，在未使用时可单独进行打开或关闭以降低功耗。

注意：

1. 上电默认的系统时钟源为 HIRC，上电默认的系统时钟频率为 $f_{HIRC}/2$ ，用户可在上电后的正常操作阶段，通过软件方式切换时钟源。切换前务必确保所选时钟源已处于稳定的工作状态；
2. 系统时钟源切换时，必须先将系统时钟源切换至 HIRC，再切换至目标时钟源。

5.3.2 内建高频 48MHz 振荡器 (HIRC)

内建高频时钟 HIRC 可通过对寄存器 RCC_CFG0 的 HIRCEN 位置 1 开启，或清 0 关闭；且用户需要先对 RCC 保护寄存器 RCC_KEY 写入一个大于 0x40 的值后才能对 RCC_CFG0 和 RCC_CFG1 进行改写。

HIRC 有以下功能及特性：

- 作为系统运行时钟
- 系统上电默认时钟频率 f_{SYS} 为 $f_{HIRC}/2$
- 频率误差：跨越 (1.8V~5.5V) 及 (-40 ~ 105°C) 应用环境下的频率误差 $\leq \pm 1\%$
- 可通过 32.768kHz 外接晶振进行自动校准，校准后 HIRC 精度可无限接近外接 32.768kHz 晶振的精度

5.3.3 内置高频晶体振荡器电路，可外接 2~16MHz 高频振荡器 (HXT)

外接高频晶振时钟 HXT 可通过对寄存器 RCC_CFG0 的 HXTEN 位置 1 开启，或清 0 关闭；且用户需要先对 RCC 保护寄存器 RCC_KEY 写入一个大于 0x40 的值后才能对 RCC_CFG0 和 RCC_CFG1 进行改写。

另外，用户还需根据外接高频晶振频率设置合理的 CRY_HF 值，当外接高频晶振频率小于 12MHz 时，应对 CRY_HF 写 0；当外接高频晶振频率大于 12MHz 时，应对 CRY_HF 写 1。

HXT 有以下功能及特性

- 可作为系统运行时钟
- 可外接 2~16MHz 高频振荡器

5.3.4 内建低频 32kHz 振荡器 (LIRC)

内建低频时钟 LIRC 可通过对寄存器 RCC_CFG0 的 LIRCEN 位置 1 开启，或清 0 关闭；且用户需要先对 RCC 保护寄存器 RCC_KEY 写入一个大于 0x40 的值后才能对 RCC_CFG0 和 RCC_CFG1 进行改写。

LIRC 有以下功能及特性

- 作为系统运行时钟
- 作为 Base Timer 的时钟源
- 固定为 WDT 时钟源，WDT 使能后此时钟源必开启
- 频率误差：跨越 (1.8 ~ 5.5V) 及 25°C 应用环境，经寄存器修正后频率误差不超过 $\pm 4\%$

5.3.5 内置低频振荡电路，可外接 32.768kHz 低频振荡器 (LXT)

外接低频晶振时钟 LXT 可通过对寄存器 RCC_CFG0 的 LXTEN 位置 1 开启，或清 0 关闭；且用户需要先对 RCC 保护寄存器 RCC_KEY 写入一个大于 0x40 的值后才能对 RCC_CFG0 和 RCC_CFG1 进行改写。

LXT 有以下功能及特性

- 作为系统运行时钟
- 作为 Base Timer 时钟源
- 外接 32.768kHz 低频振荡器

- 可通过 LXT 对 HIRC 进行自动校准

5.4 寄存器访问时钟和外设时钟源

5.4.1 寄存器访问时钟

寄存器访问时钟是驱动 Cortex®-M0+内核及内部总线操作的时钟信号，该信号直接驱动指令执行流水线、寄存器组访问以及内核与总线之间的数据传输。

寄存器访问时钟与 AHB（Advanced High-performance Bus）和 APB（Advanced Peripheral Bus）总线时钟保持严格同步，确保 CPU 核心操作与外设接口在统一时序框架下协同工作。用户需通过 AHB 总线外设时钟使能寄存器（AHB_CFG）或 APB 总线外设时钟使能寄存器（APBn_CFG, n=0~2）进行总线时钟使能和外设时钟门控操作。仅当对应外设的总线和寄存器访问时钟位均被使能（Enable）时，方可进行该外设寄存器的读写操作，否则将触发硬件保护机制导致访问无效。

5.4.2 外设时钟源

外设时钟源是为特定外设模块（如 PWM、UART、ADC 等）提供独立工作节拍的时钟信号源。它可以是系统时钟的分频/倍频产物，也可以是独立的内置或外部时钟源（如内部 RC 振荡器、外部晶振等）。外设时钟源的存在使得不同外设可根据自身需求选择最优的时钟频率和稳定性。用户也可通过单独关闭未使用外设的时钟以降低功耗。

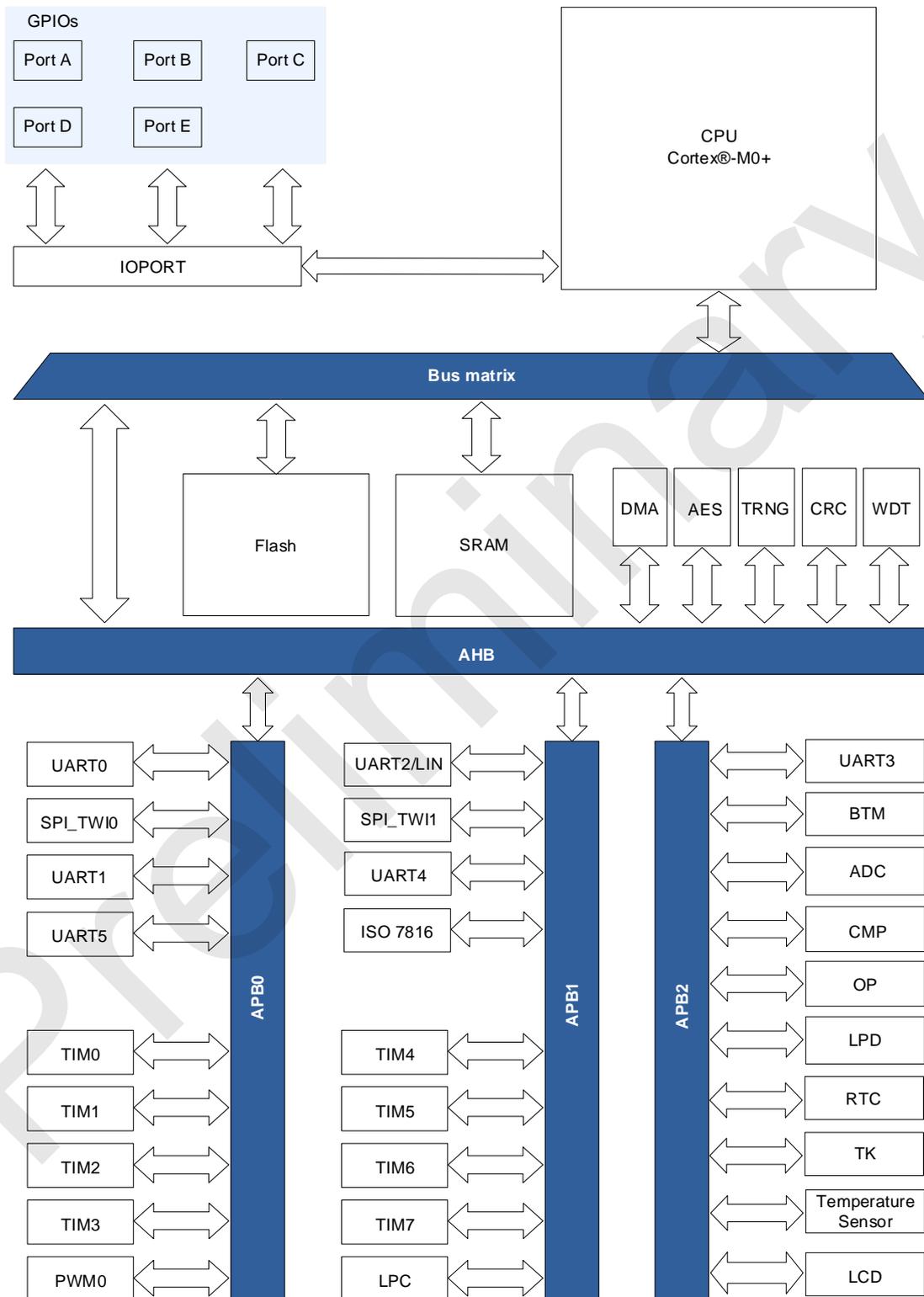
SC32L14T/14G 的部分外设提供了多种可选的时钟源，用户可通过外设时钟源选择寄存器（RCC_CFGn, n=0~1）为外设选择适合的时钟源。

5.4.3 总线时钟

用户可通过多个预分频器配置 IOPORT、AHB、APB0、APB1、APB2 域的频率。

- IOPORT: GPIO 端口域主时钟，由系统时钟 SYSCLK 分频得到，最大频率是 48MHz，所有 GPIO 都由 IOPORT 驱动。
- HCLK: AHB 域主时钟，由系统时钟 SYSCLK 分频得到，最大频率是 48MHz，包括 Cortex®-M0+内核、内存、DMA 等都由 HCLK 驱动。
- PCLK0: APB0 域主时钟，由 HCLK 分频得到，最大频率是 HCLK 的频率，APB0 总线上的外设寄存器访问时钟都由 PCLK0 驱动，总线上的部分外设时钟源由 PCLK0 提供；
- PCLK1: APB1 域主时钟，由 HCLK 分频得到，最大频率是 HCLK 的频率，APB1 总线上的外设寄存器访问时钟都由 PCLK1 驱动，总线上的部分外设时钟源由 PCLK1 提供；
- PCLK2: APB2 域主时钟，由 HCLK 分频得到，最大频率是 HCLK 的频率，APB2 总线上的外设寄存器访问时钟都由 PCLK2 驱动，总线上的部分外设时钟源由 PCLK2 提供；

总线架构图如下图所示，外设寄存器读取频率取决于挂载该外设的总线频率。



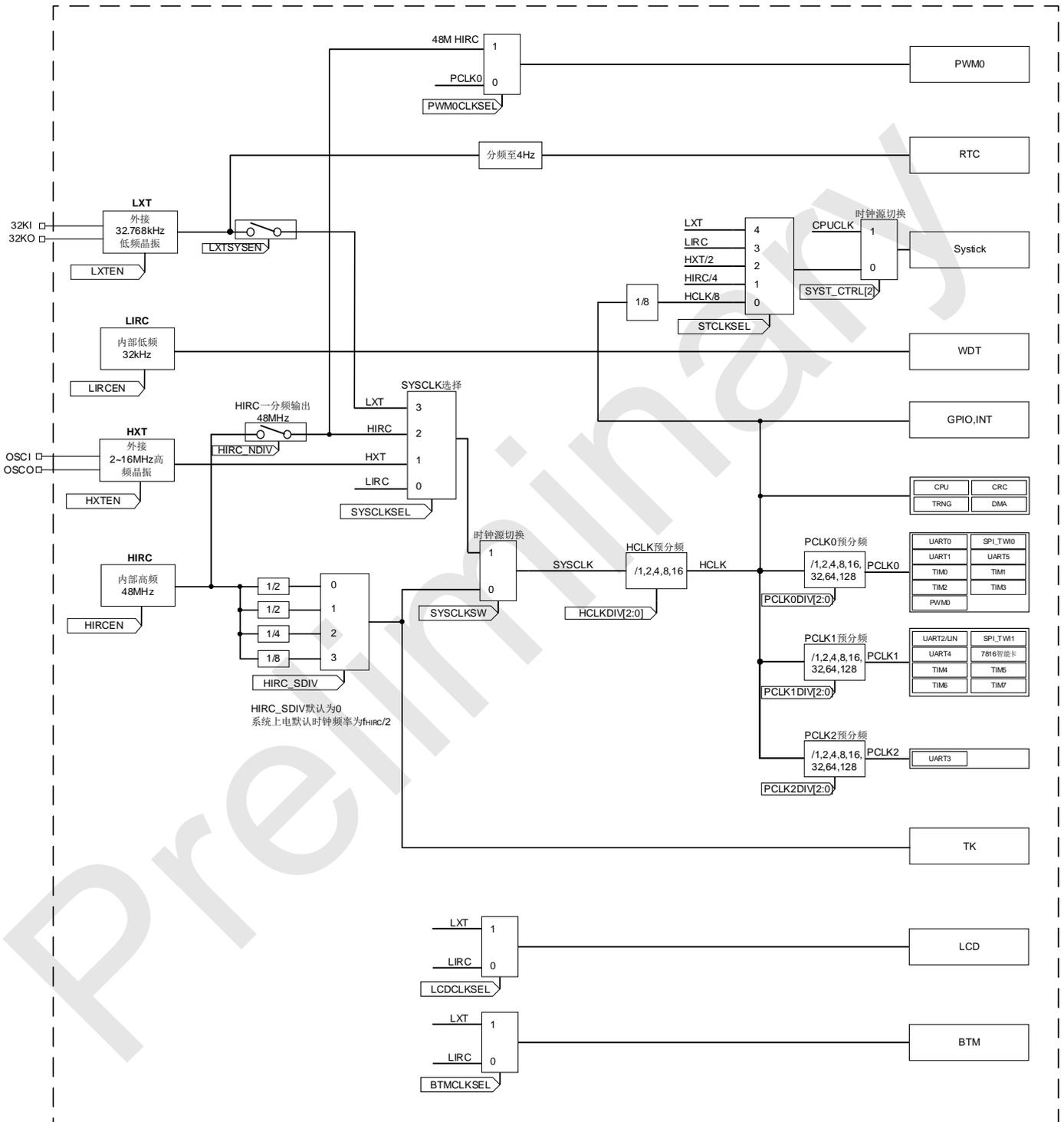
总线结构框图

5.4.4 外设时钟

总线搭载的外设时钟一般由其对应的总线时钟提供，但以下外设可由其他时钟源提供时钟：

- RTC 以 LXT 分频至 4Hz 的频率作为 RTC 时钟频率
- TK 以 HIRC 经过 HIRC_SDIV 选择的分频后频率作为 TK 时钟频率
- PWM0 有以下时钟源可供选择：
 - PCLK0，为 PWM0 所在总线上的时钟
 - 48M HIRC，为 HIRC 1 分频时钟
- LCD 有以下时钟源可供选择：
 - LIRC，为内部低频时钟
 - LXT，为外接低频晶振时钟
- BTM 有以下时钟源可选：
 - LIRC，为内部低频时钟
 - LXT，为外接低频晶振时钟
- WDT 时钟源固定为 LIRC，当 WDT 使能后，LIRC 会自动开启，WDT 工作的过程中 LIRC 始终保持振荡，用户无法关闭。
- SysTick 有以下时钟源可选：
 - LXT，为外接低频晶振时钟
 - LIRC，为内部低频时钟
 - HXT/2，为外接高频晶振时钟 2 分频后的时钟
 - HIRC/4，为内部高频时钟 4 分频后的时钟
 - HCLK/8，为 AHB 总线上的时钟 8 分频后的时钟
 - CPUCLK，为内核时钟

5.4.5 外设时钟框图



5.4.6 外设时钟使能寄存器

每个外设时钟都可通过 `AHB_CFG` 和 `APBn_CFG` 寄存器的相应使能位使能，当外设时钟未激活时，不支持

外设寄存器进行读写访问。

5.4.7 低功耗模式时钟

在低功耗 STOP 模式下，CPU 时钟及大部分时钟源和外设时钟都会停止，且程序停止运行。但以下时钟源在低功耗模式下仍能正常工作：

- 内建低频时钟 LIRC
- 外接低频振荡器时钟 LXT

5.5 RCC 寄存器

5.5.1 RCC 相关寄存器表

5.5.1.1 RCC 保护寄存器 RCC_KEY

寄存器	读/写	说明	复位值	上电初始值
RCC_KEY	读/写	RCC 保护寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7~0	RCCKEY[7:0]	RCC_CFG0、RCC_CFG1 这两个寄存器的操作使能开关及时限设置。 写入一个大于等于 0x40 的值 n，代表： 1. 打开 RCC_CFG0、RCC_CFG1 这两个寄存器的写操作功能； 2. n 个系统时钟后如果接收不到寄存器写入命令，则 RCC 改写功能被重新关闭。
31~8	-	保留

5.5.1.2 系统时钟源选择寄存器 RCC_CFG0（写保护）

*该寄存器受写保护，必须操作 RCC 保护寄存器 RCC_KEY 才能修改。

寄存器	读/写	说明	复位值	上电初始值
RCC_CFG0	读/写	系统时钟源选择寄存器	0x0000_1040	0x0000_1040

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	LXTSYSEN
15	14	13	12	11	10	9	8
INTEN	HIRC_NDIV	WAIT[1:0]	HPLDO_DP	-	SYSCLKSEL[1:0]		
7	6	5	4	3	2	1	0
SYSCLKSW	HIRCEN	HXTEN	CRY_HF	-	-	LIRCEN	LXTEN

位编号	位符号	说明
16	LXTSYSEN	LXT 作系统时钟源使能位 0: 切断 LXT 与系统时钟间的联系, 降低功耗。此时 LXT 仅作 RTC 的时钟源 1: LXT 可作系统时钟源。当 SYSCLKSEL[1:0]=11 时, LXT 作系统时钟源
15	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
14	HIRC_NDIV	内建高频晶振一分频输出 48MHz 使能位 0: 禁止 1: 使能
13~12	WAIT[1:0] 复位值=01	00: 保留, 不建议用户设置为“00”, 防止出现异常 01: 1 个 wait, 24MHz 主频推荐, 上电默认值 10: 2 个 wait, 48MHz 及以上主频推荐 11: 3 个 wait, 48MHz 及以上主频推荐 注意: 用户设置主频为 24MHz 时, 至少需要 1 个 wait; 设置主频高于 48MHz 时, 至少需要 2 个 wait
11	HPLDO_DP	低频系统时钟功耗调整位 0: 系统时钟源非 LIRC 或 LXT 推荐设置 1: 系统时钟源选择 LIRC 或 LXT 推荐设置。当系统时钟选择 LIRC 或 LXT 时, 此位写 1 可降低整体功耗
9~8	SYSCLKSEL[1:0]	系统时钟源选择位 00: 系统时钟源来自 LIRC 01: 系统时钟源来自 HXT 10: 系统时钟源来自 HIRC 一分频, 48MHz 11: 系统时钟源来自 LXT 注意: 1. 上电默认的系统时钟源为 HIRC, 上电默认的系统时钟频率为 $f_{HIRC}/2$, 用户可在上电后的正常操作阶段, 通过软件方式切换时钟源。切换前务必确保所选时钟源已处于稳定的工作状态; 2. 系统时钟源无论选择切换至哪一种, 都必须先将系统时钟源切换至 HIRC, 再切换至目标时钟源。
7	SYSCLKSW	系统时钟源切换位, 使能后系统时钟源从 HIRC 切换至 SYSCLKSEL 所选的时钟: 0: 系统时钟源为 HIRC, 上电默认系统时钟频率为 $f_{HIRC}/2$ 1: 系统时钟源为 SYSCLKSEL 设置项 对此位改写后, 必须内部电路切换成功才会更新改写的值, 否则读到的一直是改写前的状态。用户可以通过读取此位的方式判断时钟源是否已切换成功。复位/唤醒后此位将自动清零, 即复位/唤醒后由 HIRC 提供系统时钟。 注意: 1. 上电默认的系统时钟源为 HIRC, 上电默认的系统时钟频率为 $f_{HIRC}/2$, 用户可在上电后的正常操作阶段, 通过软件方式切换时钟源。切换前务必确保所选时钟源已处于稳定的工作状态; 2. 系统时钟源无论选择切换至哪一种, 都必须先将系统时钟源切换至 HIRC, 再切换至目标时钟源。

位编号	位符号	说明
6	HIRCEN	内建高频 48MHz 振荡器 HIRC 使能位 0: 禁止 1: 使能 当 SYSCLSW = 0, 选择 HIRC 作为系统时钟时, 此位不可写入。 复位/唤醒后此位将由硬件置 1, 即复位/唤醒后由 HIRC 提供系统时钟。
5	HXTEN	外接高频晶振 HXT 使能位 0: 禁止 1: 使能 复位/唤醒后此位将自动清零。
4	CRY_HF	外接高频晶振 HXT 振荡频率范围选择位 0: 外接晶振振荡频率 < 12M 1: 外接晶振振荡频率 ≥ 12M 注意: 对 CRY_HF 位配置时需与实际外接晶振频率相对应, 否则芯片会发生错误!
1	LIRCEN	内建低频 32kHz 振荡器 LIRC 使能位 0: 禁止 1: 使能
0	LXTEN	外接低频晶振 LXT 使能位 0: 禁止 1: 使能
31~16 10 3~2	-	保留

5.5.1.3 外设时钟源选择寄存器 RCC_CFG1 (写保护)

*该寄存器受写保护, 必须操作 RCC 保护寄存器 RCC_KEY 才能修改。

寄存器	读/写	说明	复位值	上电初始值
RCC_CFG1	读/写	外设时钟源选择寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
STCLKSEL[2:0]		-	-	-	PWM0CLKSEL	LCDCLKSEL	BTMCLKSEL

位编号	位符号	说明
7~5	STCLKSEL[2:0]	SysTick 时钟源选择位 000: 时钟源来自 HCLK/8 001: 时钟源来自 HIRC/4 010: 时钟源来自 HXT/2 011: 时钟源来自 LIRC 100: 时钟源来自 LXT 注意: 用户配置时注意, 如果 SysTick 时钟源不是来自 HCLK, SysTick 时钟源频率必须小于或等于 $f_{HCLK}/2$ 。

位编号	位符号	说明
2	PWM0CLKSEL	8 路 16 位多功能 PWM0 时钟源选择位 0: 时钟源来自 PCLK 1: 时钟源来自 48MHz HIRC 对此位改写后, 必须等待内部电路切换成功才会更新改写的值, 否则读到的一直是改写前的状态。用户可以通过读取此位的方式判断时钟源是否已切换成功。
1	LCDCLKSEL	LCD 时钟源选择位 0: 时钟源来自 LIRC 1: 时钟源来自 LXT 对此位改写后, 必须内部电路切换成功才会更新改写的值, 否则读到的一直是改写前的状态。 用户可以通过读取此位的方式判断时钟源是否已切换成功。
0	BTMCLKSEL	BTM 时钟源选择位 0: 时钟源来自 LIRC 1: 时钟源来自 LXT 对此位改写后, 必须等待内部电路切换成功才会更新改写的值, 否则读到的一直是改写前的状态。 用户可以通过读取此位的方式判断时钟源是否已切换成功。
31~8 4~3	-	保留

5.5.1.4 时钟状态寄存器 RCC_STS

寄存器	读/写	说明	复位值	上电初始值
RCC_STS	读/写	时钟状态寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
3	SRAMPEIF	SRAM 奇偶校验错误标志位 检测到 SRAM 奇偶校验错误时, 此位由硬件置 1。通过软件写入 1 可将此位清 0 0: 未检测到 SRAM 奇偶校验错误 1: 检测到 SRAM 奇偶校验错误
0	CLKFIF	时钟源异常标志位, 针对系统时钟源为外振时的情况 0: 当前时钟源无异常 1: 当前时钟源异常, 系统时钟源已自动切换为 HIRC, 如果 RCC 中断使能已开启 (RCC_CFG0.INTEN=1), 可产生中断。 复位后可清除 CLKFIF 标志。
31~4 2~1	-	保留

5.5.1.5 SysTick 校准参数寄存器 SYST_CALIB

寄存器	读/写	说明	复位值	上电初始值
SYST_CALIB	只读	SysTick 校准参数寄存器	0x0000_2327	0x0000_2327

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

位编号	位符号	说明
23~0	CALIB[23:0]	校准寄存器默认值： 若，上电默认时钟为 f_{HCLK}/n (MHz)， n 是上电默认分频系数，上电默认时钟源为 HIRC 则，SysTick 校准值初始值为 $1000 * (f_{HCLK}/n)$ ，即保证默认可产生 1ms 时间基准
31~24	-	保留

5.5.1.6 AHB 总线外设时钟使能寄存器 AHB_CFG

寄存器	读/写	说明	复位值	上电初始值
AHB_CFG	读/写	AHB 总线外设时钟使能寄存器	0x0010_0000	0x0010_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	CLKDIV[2:0]		-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	AESEN	-	-	IFBEN	CRGEN	DMAEN

位编号	位符号	说明
22~20	CLKDIV[2:0]	AHB 时钟分频设置位 AHB 总线时钟 HCLK 来自系统时钟 SYSCLK 的分频： 000: $f_{HCLK} = f_{SYS}$ 001: $f_{HCLK} = f_{SYS} / 2$ 010: $f_{HCLK} = f_{SYS} / 4$ 011: $f_{HCLK} = f_{SYS} / 8$ 100: $f_{HCLK} = f_{SYS} / 16$ 其它：保留
5	AESEN	AES 模块时钟使能位 0: 禁止 1: 使能

位编号	位符号	说明
2	IFBEN	Customer Option 映射寄存器时钟使能位 使用 OPINX 配合 OPREG 改写 IFB 映射寄存器之前，应先要打开时钟使能。 0: 禁止 1: 使能
1	CRCEN	CRC 模块时钟使能位 0: 禁止 1: 使能
0	DMAEN	DMA 时钟使能位 0: 禁止 1: 使能
31~23 19~6 4~3	-	保留

5.5.1.7 APB0 总线外设时钟使能寄存器 APB0_CFG

寄存器	读/写	说明	复位值	上电初始值
APB0_CFG	读/写	APB0 总线外设时钟使能寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
ENAPB	CLKDIV[2:0]		-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	UART5EN	PWM0EN
7	6	5	4	3	2	1	0
UART1EN	UART0EN	-	SPI_TWI0EN	TIM3EN	TIM2EN	TIM1EN	TIM0EN

位编号	位符号	说明
23	ENAPB	APB0 总线时钟开关控制位 0: 禁止 1: 使能
22~20	CLKDIV[2:0]	APB0 时钟分频设置位 APB0 总线时钟 PCLK0 来自 HCLK 的分频: 000: $f_{PCLK0} = f_{HCLK}$ 001: $f_{PCLK0} = f_{HCLK} / 2$ 010: $f_{PCLK0} = f_{HCLK} / 4$ 011: $f_{PCLK0} = f_{HCLK} / 8$ 100: $f_{PCLK0} = f_{HCLK} / 16$ 101: $f_{PCLK0} = f_{HCLK} / 32$ 110: $f_{PCLK0} = f_{HCLK} / 64$ 111: $f_{PCLK0} = f_{HCLK} / 128$
9	UART5EN	UART5 时钟使能位 0: 禁止 1: 使能
8	PWM0EN	PWM0 时钟使能位 0: 禁止 1: 使能
7	UART1EN	UART1 时钟使能位 0: 禁止 1: 使能

位编号	位符号	说明
6	UART0EN	UART0 时钟使能位 0: 禁止 1: 使能
4	SPI_TWI0EN	SPI_TWI0 时钟使能位 0: 禁止 1: 使能
3	TIM3EN	Timer3 时钟使能位 0: 禁止 1: 使能
2	TIM2EN	Timer2 时钟使能位 0: 禁止 1: 使能
1	TIM1EN	Timer1 时钟使能位 0: 禁止 1: 使能
0	TIM0EN	Timer0 时钟使能位 0: 禁止 1: 使能
31~24 19~10 5	-	保留

5.5.1.8 APB1 总线外设时钟使能寄存器 APB1_CFG

寄存器	读/写	说明	复位值	上电初始值
APB1_CFG	读/写	APB1 总线外设时钟使能寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
ENAPB	CLKDIV[2:0]		-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	SCEN	-
7	6	5	4	3	2	1	0
UART2EN	UART4EN	-	SPI_TWI1EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN

位编号	位符号	说明
23	ENAPB	APB1 总线时钟开关控制位 0: 禁止 1: 使能
22~20	CLKDIV[2:0]	APB1 时钟分频设置位 APB1 总线时钟 PCLK1 来自 HCLK 的分频: 000: $f_{PCLK1} = f_{HCLK}$ 001: $f_{PCLK1} = f_{HCLK} / 2$ 010: $f_{PCLK1} = f_{HCLK} / 4$ 011: $f_{PCLK1} = f_{HCLK} / 8$ 100: $f_{PCLK1} = f_{HCLK} / 16$ 101: $f_{PCLK1} = f_{HCLK} / 32$ 110: $f_{PCLK1} = f_{HCLK} / 64$ 111: $f_{PCLK1} = f_{HCLK} / 128$

位编号	位符号	说明
9	SCEN	SC 时钟使能位 0: 禁止 1: 使能
7	UART2EN	UART2 时钟使能位 0: 禁止 1: 使能
6	UART4EN	UART4 时钟使能位 0: 禁止 1: 使能
4	SPI_TWI1EN	SPI_TWI1 时钟使能位 0: 禁止 1: 使能
3	TIM7EN	Timer7 时钟使能位 0: 禁止 1: 使能
2	TIM6EN	Timer6 时钟使能位 0: 禁止 1: 使能
1	TIM5EN	Timer5 时钟使能位 0: 禁止 1: 使能
0	TIM4EN	Timer4 时钟使能位 0: 禁止 1: 使能
31~24 19~10 8, 5	-	保留

5.5.1.9 APB2 总线外设时钟使能寄存器 APB2_CFG

寄存器	读/写	说明	复位值	上电初始值
APB2_CFG	读/写	APB2 总线外设时钟使能寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
ENAPB	CLKDIV[2:0]		-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	UART3EN	LCDEN	-

位编号	位符号	说明
23	ENAPB	APB2 总线时钟开关控制位 0: 禁止 1: 使能
22~20	CLKDIV[2:0]	APB2 时钟分频设置位 APB2 总线时钟 PCLK2 来自 HCLK 的分频: 000: $f_{PCLK2} = f_{HCLK}$ 001: $f_{PCLK2} = f_{HCLK} / 2$ 010: $f_{PCLK2} = f_{HCLK} / 4$ 011: $f_{PCLK2} = f_{HCLK} / 8$

位编号	位符号	说明
		100: $f_{PCLK2} = f_{HCLK} / 16$ 101: $f_{PCLK2} = f_{HCLK} / 32$ 110: $f_{PCLK2} = f_{HCLK} / 64$ 111: $f_{PCLK2} = f_{HCLK} / 128$
2	UART3EN	UART3 时钟使能位 0: 禁止 1: 使能
1	LCDEN	LCD 时钟使能位 0: 禁止 1: 使能
31~24 19~3 0	-	保留

5.5.1.10 AHB 总线外设复位控制寄存器 AHB_RST

寄存器	读/写	说明	复位值	上电初始值
AHB_RST	读/写	AHB 总线外设复位控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
5	AESRST	AES 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 AES
1	CRCRST	CRC 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 RCC
0	DMARST	DMA 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 DMA
31~6 4~2	-	保留

5.5.1.11 APB0 总线外设复位控制寄存器 APB0_RST

寄存器	读/写	说明	复位值	上电初始值
APB0_RST	读/写	APB0 总线外设复位控制寄存器	0x0000_0000	0x0000_0000

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

-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	UART5RST	PWM0RST
7	6	5	4	3	2	1	0
UART1RST	UART0RST	-	SPI_TWI0RST	TIM3RST	TIM2RST	TIM1RST	TIM0RST

位编号	位符号	说明
9	UART5RST	UART5 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 UART5
8	PWM0RST	PWM0 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 PWM0
7	UART1RST	UART1 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 UART1
6	UART0RST	UART0 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 UART0
4	SPI_TWI0RST	SPI_TWI0 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 SPI_TWI0
3	TIM3RST	Timer3 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 Timer3
2	TIM2RST	Timer2 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 Timer2
1	TIM1RST	Timer1 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 Timer1
0	TIM0RST	Timer0 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 Timer0
31~10 5	-	保留

5.5.1.12 APB1 总线外设复位控制寄存器 APB1_RST

寄存器	读/写	说明	复位值	上电初始值
APB1_RST	读/写	APB1 总线外设复位控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	SCRST	-
7	6	5	4	3	2	1	0
UART2RST	UART4RST	-	SPI_TWI1RST	TIM7RST	TIM6RST	TIM5RST	TIM4RST

位编号	位符号	说明
9	SCRST	SC 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 SC
7	UART2RST	UART2 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 UART2
6	UART4RST	UART4 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 UART4
4	SPI_TWI1RST	SPI_TWI1 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 SPI_TWI1
3	TIM7RST	Timer7 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 Timer7
2	TIM6RST	Timer6 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 Timer6
1	TIM5RST	Timer5 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 Timer5
0	TIM4RST	Timer4 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 Timer4
31~10 8, 5	-	保留

5.5.1.13 APB2 总线外设复位控制寄存器 APB2_RST

寄存器	读/写	说明	复位值	上电初始值
APB2_RST	读/写	APB2 总线外设复位控制寄存器	0x0000_0000	0x0000_0000

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

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	UART3RST	LCDRST	-

位编号	位符号	说明
2	UART3RST	UART3 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 UART3
1	LCDRST	LCD 复位控制位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: 复位 LCD
31~3	-	保留

5.5.1.14 NMI 中断配置寄存器 NMI_CFG

寄存器	读/写	说明	复位值	上电初始值
NMI_CFG	读/写	非可屏蔽中断 (NMI) 中断配置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
KEY[15:8]							
23	22	21	20	19	18	17	16
KEY[7:0]							
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	SRAMPEEN	INT0EN	CMPEN	CSSSEN

位编号	位符号	说明
31~16	KEY[15:0]	NMI_CFG 寄存器写保护开关 向 KEY[15:0]写入 0xA05F 解锁后才能对当前寄存器的低位进行改写操作。
3	SRAMPEEN	SRAM 奇偶校验错误中断使能位 0: SRAM 奇偶校验错误禁止触发 NMI 1: SRAM 奇偶校验错误非屏蔽中断源使能 使能后，读取 SRAM 时检测到 SRAM 奇偶校验错误会触发 NMI，手动清除相应标志位后方可退出 NMI 中断
2	INT0EN	外部中断 INT0 非屏蔽中断源使能位 0: INT0 禁止触发 NMI 1: INT0 非屏蔽中断源使能 使能后，INT0 口上的上升/下降沿中断均会触发 NMI，手动清除相应标志位后方可退出 NMI 中断。 注意：若 INT0 中断已使能，仍会优先处理 NMI。
1	CMPEN	比较器非屏蔽中断源使能位 0: CMP 禁止触发 NMI 1: CMP 非屏蔽中断源使能 使能后，CMPIF 置起时会触发 NMI，手动清除 CMPIF 标志后方可退出 NMI 中断 注意：如果 CMP 中断使能已开启 (CMPCFG->CMPIM[1:0]=1)，仍会优先处理 NMI

位编号	位符号	说明
0	CSSEN	时钟安全系统非屏蔽中断源使能位 0: CSS 禁止触发 NMI 1: CSS 系统非屏蔽中断源使能 使能后, CLKFIF 置起时会触发 NMI, 复位后可清除 CLKFIF 标志, CLKFIF 清除后方可退出 NMI 中断 注意: 如果 RCC 中断使能已开启 (RCC_CFG->INTEN=1), 仍会优先处理 NMI
15~4	-	保留

5.5.2 RCC 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
AHB 基地址: 0x4000_3000					
AHB_CFG	0x00	读/写	AHB 总线外设时钟使能寄存器	0x0010_0000	0x0010_0000
AHB_RST	0x04	读/写	AHB 总线外设复位控制寄存器	0x0000_0000	0x0000_0000
RCC_KEY	0x0C	读/写	RCC 保护寄存器	0x0000_0000	0x0000_0000
RCC_CFG0	0x14	读/写	系统时钟源选择寄存器	0x0000_1040	0x0000_1040
RCC_CFG1	0x18	读/写	外设时钟源选择寄存器	0x0000_0000	0x0000_0000
RCC_STS	0x20	读/写	时钟状态寄存器	0x0000_0000	0x0000_0000
SYST_CALIB	0x28	读/写	SysTick 校准参数寄存器	0x0000_2327	0x0000_2327
NMI_CFG	0x2C	读/写	NMI 中断配置寄存器	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
APB0 基地址: 0x4002_0000					
APB0_CFG	0x00	读/写	APB0 总线外设时钟使能寄存器	0x0000_0000	0x0000_0000
APB0_RST	0x04	读/写	APB0 总线外设复位控制寄存器	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
APB1 基地址: 0x4002_1000					
APB1_CFG	0x00	读/写	APB1 总线外设时钟使能寄存器	0x0000_0000	0x0000_0000
APB1_RST	0x04	读/写	APB1 总线外设复位控制寄存器	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
APB2 基地址: 0x4002_2000					
APB2_CFG	0x00	读/写	APB2 总线外设时钟使能寄存器	0x0000_0000	0x0000_0000
APB2_RST	0x04	读/写	APB2 总线外设复位控制寄存器	0x0000_0000	0x0000_0000

6 中断

- M0+内核最多提供 32 个中断源，中断号为 0~31，SC32L14T/14G 共 27 个中断源
- 四级中断优先级可设（内核特性），中断优先级通过内核寄存器组 Interrupt priority registers 设置

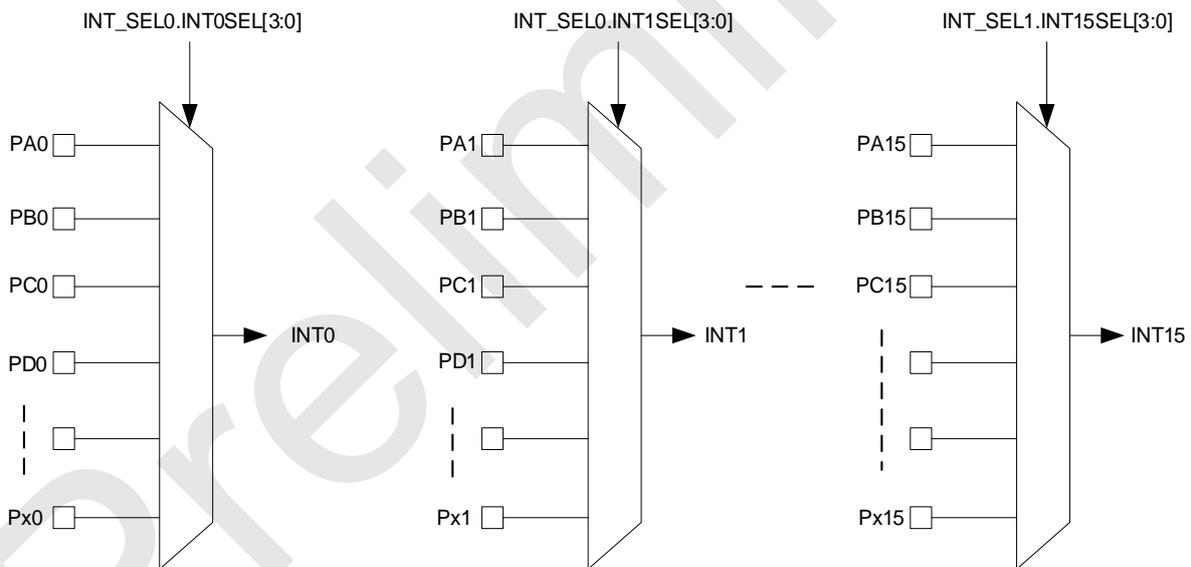
6.1 外部中断 INT0~15

外部中断有 16 个中断源，共占用 4 个中断向量，这 16 个外部中断源，全部可设上升沿、下降沿、双沿中断，经设置后可覆盖到所有的 GPIO 管脚，软件置起相应中断标志位（RIF/FIF 置 1），可触发进入相应的中断。

SC32L14T/14G 系列外部中断特性如下：

- 16 个 INT 中断源，共占用 4 个中断向量
- INT 经切换设置后可覆盖到所有的 GPIO 管脚
- 全部可设上升沿、下降沿、双沿中断，且均有独立对应中断标志位
- 软件置起相应中断标志位，可触发进入相应的中断

注意：切换 INT 功能时，用户需手动将 INTn（n=0~15）所在的 GPIO 端口设置为输入带上拉状态，端口输出状态下检测不到外部中断。



6.2 中断与事件

- NVIC 关闭，中断请求屏蔽位开启，可产生事件，不产生中断
- NVIC 开启后，中断请求屏蔽位做模块内的总中断开关

6.3 中断向量表

中断向量号	中断号	优先级	中断向量地址	中断源	内核/NVIC 使能位	中断请求屏蔽位	中断子开关	中断标志位	唤醒 stop
0	-	-	0x0000_0000	-		-	\	\	能

中断向量号	中断号	优先级	中断向量地址	中断源	内核/NVIC 使能位	中断请求屏蔽位	中断子开关	中断标志位	唤醒 stop
1	-	固定	0x0000_0004	RESET	PRIMASK	SCB	\	\	能
2	-	固定	0x0000_0008	NMI_Handler		SCB	\	\	能
3	-	固定	0x0000_000C	HardFault_Handler	PRIMASK	SCB	\	\	能
4~10	-	-	0x0000_0010 - 0x0000_0028	-		-	\	\	能
11	-	可设		SVC_Handler	PRIMASK	SCB	\	\	能
12~13	-	-	0x0000_0030 0x0000_0034	-		-	\	\	能
14	-	可设	0x0000_0038	PendSV_Handler	PRIMASK	SCB	\	\	能
15	-	可设	0x0000_003C	SysTick_Handler	PRIMASK	SysTick_CTRL	\	\	不能
16	0	可设	0x0000_0040	INT0	NVIC->ISER[0].0	INTF_IE->ENF _x , x=0 INTR_IE->ENR _x	\	INTF_STS->FIF _x INTR_STS->RIF _x	能
17	1	可设	0x0000_0044	INT1-7	NVIC->ISER[0].1	INTF_IE->ENF _x , x=1~7 INTR_IE->ENR _x	\	INTF_STS->FIF _x INTR_STS->RIF _x	能
18	2	可设	0x0000_0048	INT8-11	NVIC->ISER[0].2	INTF_IE->ENF _x , x=8~11 INTR_IE->ENR _x	\	INTF_STS->FIF _x INTR_STS->RIF _x	能
19	3	可设	0x0000_004C	INT12-15	NVIC->ISER[0].3	INTF_IE->ENF _x , x=12~15 INTR_IE->ENR _x	\	INTF_STS->FIF _x INTR_STS->RIF _x	能
20	4	可设	0x0000_0050	RCC 停振检测	NVIC->ISER[0].4	RCC_CFG->INTEN	\	RCC_STS->CLKFIF	不能
21	5	可设	0x0000_0054	LPD	NVIC->ISER[0].5	LPD_IDE->INTEN	\	LPD_CON->LPDIF	不能
22	6	可设	0x0000_0058	BTM	NVIC->ISER[0].6	BTM_CON->INTEN	\	BTM_STS->BTMIF	能
23	7	可设	0x0000_005C	UART0	NVIC->ISER[0].7	UART0_IDE->INTEN	UART0_IDE->TXIE UART0_IDE->RXIE	UART0_STS->TXIF UART0_STS->RXIF	能
				UART2/LIN	\	UART2_IDE->INTEN	UART2_IDE->TXIE UART2_IDE->RXIE UART2_IDE->BKIE UART2_IDE->SLVHEIE	UART2_STS->TXIF UART2_STS->RXIF UART2_STS->BKIF UART2_STS->SLVHEIF	能
				UART4	\	UART4_IDE->INTEN	UART4_IDE->TXIE UART4_IDE->RXIE	UART4_STS->TXIF UART4_STS->RXIF	能
24	8	可设	0x0000_0060	UART1	NVIC->ISER[0].8	UART1_IDE->INTEN	UART1_IDE->TXIE UART1_IDE->RXIE	UART1_STS->TXIF UART1_STS->RXIF	能
				UART3	\	UART3_IDE->INTEN	UART3_IDE->TXIE UART3_IDE->RXIE	UART3_STS->TXIF UART3_STS->RXIF	能
				UART5	\	UART5_IDE->INTEN	UART5_IDE->TXIE UART5_IDE->RXIE	UART5_STS->TXIF UART5_STS->RXIF	能
				7816 智能卡	\	SC0_IDE->INTEN	SC0_IDE->TXIE SC0_IDE->RXIE SC0_IDE->ERRIE	SC0_STS->TC SC0_STS->RC SC0_STS->ROVF SC0_STS->FER SC0_STS->WTER SC0_STS->RPER SC0_STS->TPER	不能
25	9	可设	0x0000_0064	SPI0/TWI0	NVIC->ISER[0].9	TWI_SPI0_IDE->INTEN	TWI_SPI0_IDE->TBIE	TWI_SPI0_STS->QTWIF TWI_SPI0_STS->TXEIF	不能
26	10	可设	0x0000_0068	SPH/TWI1	NVIC->ISER[0].10	TWI_SPI1_IDE->INTEN	TWI_SPI1_IDE->TBIE	TWI_SPI1_STS->QTWIF TWI_SPI1_STS->TXEIF	不能
27	11	可设	0x0000_006C	DMA0	NVIC->ISER[0].11	DMA0_CFG->INTEN	DMA0_CFG->TCIE DMA0_CFG->HTIE DMA0_CFG->TEIE	DMA0_STS->GIF DMA0_STS->TCIF DMA0_STS->HTIF DMA0_STS->TEIF	不能
28	12	可设	0x0000_0070	DMA1	NVIC->ISER[0].12	DMA1_CFG->INTEN	DMA1_CFG->TCIE DMA1_CFG->HTIE DMA1_CFG->TEIE	DMA1_STS->GIF DMA1_STS->TCIF DMA1_STS->HTIF DMA1_STS->TEIF	不能
29	13	预留	0x0000_0074	\	NVIC->ISER[0].13	\	\	\	
30	14	预留	0x0000_0078	\	NVIC->ISER[0].14	\	\	\	
31	15	可设	0x0000_007C	TIM0	NVIC->ISER[0].15	TIM0_IDE->INTEN	TIM0_IDE->TIE TIM0_IDE->EXFIE TIM0_IDE->EXRIE	TIM0_STS->TIF TIM0_STS->EXFIF TIM0_STS->EXIR	不能
32	16	可设	0x0000_0080	TIM1	NVIC->ISER[0].16	TIM1_IDE->INTEN	TIM1_IDE->TIE TIM1_IDE->EXFIE TIM1_IDE->EXRIE	TIM1_STS->TIF TIM1_STS->EXFIF TIM1_STS->EXIR	不能
33	17	可设	0x0000_0084	TIM2	NVIC->ISER[0].17	TIM2_IDE->INTEN	TIM2_IDE->TIE TIM2_IDE->EXFIE TIM2_IDE->EXRIE	TIM2_STS->TIF TIM2_STS->EXFIF TIM2_STS->EXIR	不能

中断向量号	中断号	优先级	中断向量地址	中断源	内核/NVIC 使能位	中断请求屏蔽位	中断子开关	中断标志位	唤醒 stop
34	18	可设	0x0000_0088	TIM3	NVIC->ISER[0].18	TIM3_IDE->INTEN	TIM3_IDE->TIE TIM3_IDE->EXFIE TIM3_IDE->EXRIE	TIM3_STS->TIF TIM3_STS->EXIF TIM3_STS->EXIR	不能
35	19	可设	0x0000_008C	TIM4	NVIC->ISER[0].19	TIM4_IDE->INTEN	TIM4_IDE->TIE TIM4_IDE->EXFIE TIM4_IDE->EXRIE	TIM4_STS->TIF TIM4_STS->EXIF TIM4_STS->EXIR	不能
				TIM5	\	TIM5_IDE->INTEN	TIM5_IDE->TIE TIM5_IDE->EXFIE TIM5_IDE->EXRIE	TIM5_STS->TIF TIM5_STS->EXIF TIM5_STS->EXIR	不能
36	20	可设	0x0000_0090	TIM6	NVIC->ISER[0].20	TIM6_IDE->INTEN	TIM6_IDE->TIE TIM6_IDE->EXFIE TIM6_IDE->EXRIE	TIM6_STS->TIF TIM6_STS->EXIF TIM6_STS->EXIR	不能
				TIM7	\	TIM7_IDE->INTEN	TIM7_IDE->TIE TIM7_IDE->EXFIE TIM7_IDE->EXRIE	TIM7_STS->TIF TIM7_STS->EXIF TIM7_STS->EXIR	不能
37	21	可设	0x0000_0094	PWM0	NVIC->ISER[0].21	PWM0_CON->INTEN	\	PWM0_STS->PWMIF	不能
38	22	预留	0x0000_0098	\	NVIC->ISER[0].22	\	\	\	
39	23	可设	0x0000_009C	TRNG	NVIC->ISER[0].23	SUB_CFG->TINTEN	SUB_CFG->DRDYIE	TRNG_STS->SEIS TRNG_STS->DRDYIF	不能
40	24	可设	0x0000_00A0	AES	NVIC->ISER[0].24	\	SUB_CFG->CCFIE	AES_STS->CCFIF	不能
41	25	可设	0x0000_00A4	RTC	NVIC->ISER[0].25	RTC_CON->INTEN	RTC_CON->WALIE RTC_CON->CT[2:0]	RTC_STS->WALIF RTC_STS->RTCCTIF	能
42	26	可设	0x0000_00A8	LPC	NVIC->ISER[0].26	LPC_IDE->INTEN	LPC_IDE->DIRIE LPC_IDE->CAIE LPC_IDE->CBIE LPC_IDE->RFAIE LPC_IDE->RFBIE	LPC_STS->DIRIF LPC_STS->CAIF LPC_STS->CBIF LPC_STS->RFAIF LPC_STS->RFBIF	能
43	27	预留	0x0000_00AC	\	\	\	\	\	
44	28	预留	0x0000_00B0	\	\	\	\	\	
45	29	可设	0x0000_00B4	ADC	NVIC->ISER[0].29	ADC_CON->INTEN	\	ADC_STS->ADCIF	不能
46	30	可设	0x0000_00B8	CMP	NVIC->ISER[0].30	CMP_CFG->CMPIM[1:0]	\	CMP_STS->CMPIF	能
47	31	可设	0x0000_00BC	TK	NVIC->ISER[0].31	TKCON->INTEN	\	TKCON->TKIF	能

6.4 外部中断寄存器

6.4.1 外部中断相关寄存器表

6.4.1.1 INT 中断下降沿使能寄存器 INTF_IE

寄存器	读/写	说明	复位值	上电初始值
INTF_IE	读/写	INT 中断下降沿使能寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
ENF15	ENF14	ENF13	ENF12	ENF11	ENF10	ENF9	ENF8
7	6	5	4	3	2	1	0
ENF7	ENF6	ENF5	ENF4	ENF3	ENF2	ENF1	ENF0

位编号	位符号	说明
15~0	ENFx (x=0~15)	INTx 下降沿中断使能控制位 (x=0~15) 0: 无效 1: 使能
31~16	-	保留

6.4.1.2 INT 中断上升沿使能寄存器 INTR_IE

寄存器	读/写	说明	复位值	上电初始值
INTR_IE	读/写	INT 中断上升沿使能寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
ENR15	ENR14	ENR13	ENR12	ENR11	ENR10	ENR9	ENR8
7	6	5	4	3	2	1	0
ENR7	ENR6	ENR5	ENR4	ENR3	ENR2	ENR1	ENR0

位编号	位符号	说明
15~0	ENRx (x=0~15)	INTx 上升沿中断使能控制位 (x=0~15) 0: 无效 1: 使能
31~16	-	保留

6.4.1.3 外部中断端口选择寄存器 0 INT_SELO

寄存器	读/写	说明	复位值	上电初始值
INT_SELO	读/写	外部中断端口选择寄存器 0	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
INT7SEL[3:0]				INT6SEL[3:0]			
23	22	21	20	19	18	17	16
INT5SEL[3:0]				INT4SEL[3:0]			
15	14	13	12	11	10	9	8
INT3SEL[3:0]				INT2SEL[3:0]			
7	6	5	4	3	2	1	0
INT1SEL[3:0]				INT0SEL[3:0]			

位编号	位符号	说明
31~0	INTxSEL[3:0] (x=0~7)	外部中断 INTx 端口选择位 (x=0~7) 0000: 选择 PAX 端口 0001: 选择 PBx 端口 0010: 选择 PCx 端口 0011: 选择 PDx 端口 0100: 选择 PEx 端口 其它: 保留 注意: 同一个外部中断口在同一时间内只能选择到一个 GPIO 口

6.4.1.4 外部中断端口选择寄存器 1 INT_SEL1

寄存器	读/写	说明	复位值	上电初始值
INT_SEL1	读/写	外部中断端口选择寄存器 1	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
INT15SEL[3:0]				INT14SEL[3:0]			
23	22	21	20	19	18	17	16
INT13SEL[3:0]				INT12SEL[3:0]			
15	14	13	12	11	10	9	8
INT11SEL[3:0]				INT10SEL[3:0]			
7	6	5	4	3	2	1	0
INT9SEL[3:0]				INT8SEL[3:0]			

位编号	位符号	说明
31~0	INTxSEL[3:0] (x=8~15)	外部中断 INTx 端口选择位 (x=8~15) 0000: 选择 PAX 端口 0001: 选择 PBx 端口 0010: 选择 PCx 端口 0011: 选择 PDx 端口 0100: 选择 PEx 端口 其它: 保留

6.4.1.5 外部中断下降沿控制寄存器 INTF_CON

寄存器	读/写	说明	复位值	上电初始值
INTF_CON	读/写	外部中断下降沿控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8
7	6	5	4	3	2	1	0
FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0

位编号	位符号	说明
15~0	FTx (x=0~15)	INTx 下降沿检测使能 (x=0~15) 0: 无效 1: 使能
31~16	-	保留

6.4.1.6 外部中断上升沿控制寄存器 INTR_CON

寄存器	读/写	说明	复位值	上电初始值
INTR_CON	读/写	外部中断上升沿控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8
7	6	5	4	3	2	1	0
RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0

位编号	位符号	说明
15~0	RTx (x=0~15)	INTx 上升沿检测使能 (x=0~15) 0: 无效 1: 使能
31~16	-	保留

6.4.1.7 外部中断下降沿标志寄存器 INTF_STS

寄存器	读/写	说明	复位值	上电初始值
INTF_STS	读/写	外部中断下降沿标志寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
FIF15	FIF14	FIF13	FIF12	FIF11	FIF10	FIF9	FIF8
7	6	5	4	3	2	1	0
FIF7	FIF6	FIF5	FIF4	FIF3	FIF2	FIF1	FIF0

位编号	位符号	说明
15~0	FIFx (x=0~15)	INTx 端口下降沿捕获标志位 (x=0~15) 检测到下降沿, 该位由硬件置 1, 通过软件写 0 清 0。 也可通过软件对该位写 1 触发下降沿捕获中断。
31~16	-	保留

6.4.1.8 外部中断上升沿标志寄存器 INTR_STS

寄存器	读/写	说明	复位值	上电初始值
INTR_STS	读/写	外部中断上升沿标志寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
RIF15	RIF14	RIF13	RIF12	RIF11	RIF10	RIF9	RIF8
7	6	5	4	3	2	1	0
RIF7	RIF6	RIF5	RIF4	RIF3	RIF2	RIF1	RIF0

位编号	位符号	说明
15~0	RIFx (x=0~15)	INTx 端口上升沿捕获标志位 (x=0~15) 检测到上升沿, 该位由硬件置 1, 通过软件写 0 清 0。 也可通过软件对该位写 1 触发上升沿捕获中断。
31~16	-	保留

6.4.2 外部中断寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
INT 基地址: 0x4001_1800					
INTF_IE	0x00	读/写	INT 中断下降沿使能寄存器	0x0000_0000	0x0000_0000
INTR_IE	0x20	读/写	INT 中断上升沿使能寄存器	0x0000_0000	0x0000_0000
INT_SEL0	0x40	读/写	外部中断端口选择寄存器 0	0x0000_0000	0x0000_0000
INT_SEL1	0x60	读/写	外部中断端口选择寄存器 1	0x0000_0000	0x0000_0000
INTF_CON	0x80	读/写	外部中断下降沿控制寄存器	0x0000_0000	0x0000_0000
INTR_CON	0xA0	读/写	外部中断上升沿控制寄存器	0x0000_0000	0x0000_0000
INTF_STS	0xC0	读/写	外部中断下降沿标志寄存器	0x0000_0000	0x0000_0000
INTR_STS	0xE0	读/写	外部中断上升沿标志寄存器	0x0000_0000	0x0000_0000

7 省电模式

初始上电默认运行在常规模式，即 Operation Mode，额外提供三种省电模式：

- 低速模式：系统时钟源可选择 LIRC，CPU 可工作在 32kHz
- IDLE Mode，可由任何中断唤醒
- STOP Mode，可由 INT0~15、Base Timer、TK、UART0~5、RTC、LPC 和 CMP 唤醒

Preliminary

8 低电压监测模块 (LPD)

8.1 概述

SC32L14T/14G 内建有低电压监测电路 (LPD)，监测电源电压 V_{DD} ，并将其与 LPD 门限电压 V_{LPD} 阈值进行比较。

当 V_{DD} 电压值下降至低于 V_{LPD} 或当 V_{DD} 电压值上升至高于 V_{LPD} ，LPD 状态标志位 LPDOF 会有相应的改变，LPD 中断标志位 LPDIF 会置起；如果此时 LPD 中断使能，将会产生 LPD 中断。状态标志位 LPDOF 由硬件自动置“1”和清“0”，中断标志位 LPDIF 必须由软件清“0”。

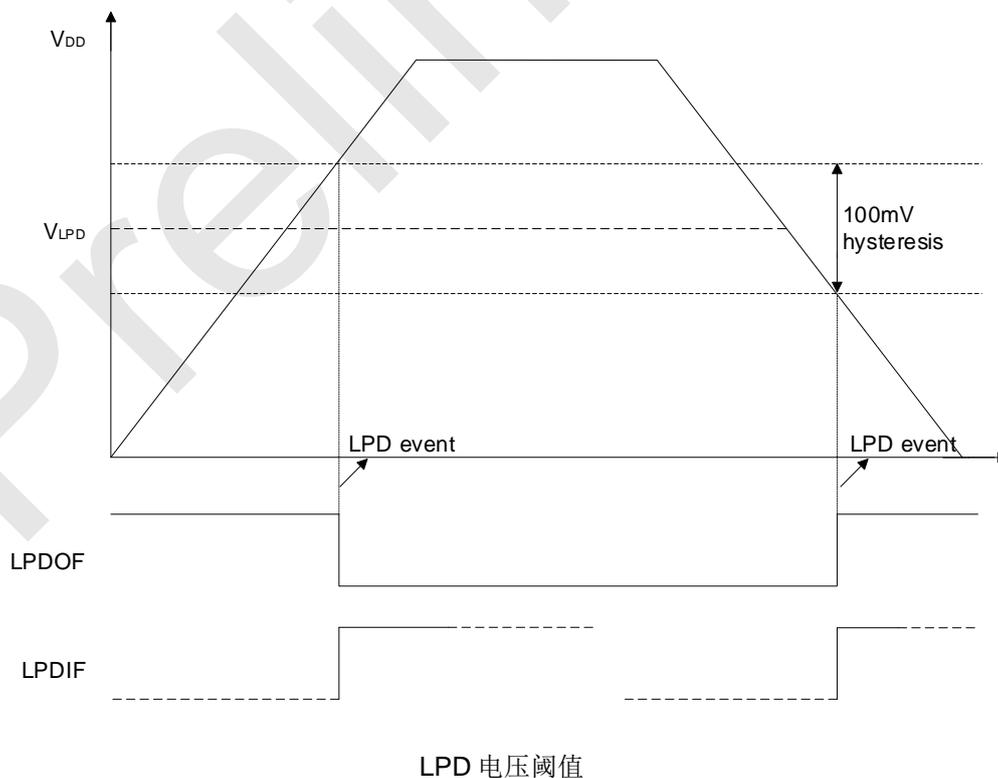
SC32L14T/14G 的 LPD 门限电压从 1.85V~3.25V，以 200mV 步进分为一档，共 8 档。用户可以通过 LPD_CON 寄存器读取 LPD 状态、获取 LPD 中断标志、配置 LPD 门限电压档位。

注意：在 STOP 模式下，LPD 电源将被关闭。

8.2 迟滞功能

LPD 具有迟滞功能，可增强芯片的抗干扰能力。门限电压 V_{LPD} 具有一个约 100mV 的阈值范围，当电源电压 V_{DD} 高于阈值范围最大值或低于阈值范围的最小值时中断标志位 LPDIF 才会置起。

LPD 的门限阈值示意图如下：



8.3 LPD 中断

SC32L14T/14G 系列的 LPD 在 V_{DD} 电压值下降至低于 V_{LPD} 或上升至高于 V_{LPD} 时会将 LPDIF 标志位置起，如果 LPD_IDE.INTEN=1，将产生中断。

中断事件	事件标志位	中断使能控制位
V_{DD} 电压值下降至低于 V_{LPD} 或上升至高于 V_{LPD}	LPDIF	LPD_IDE->INTEN

8.4 LPD 寄存器

8.4.1 LPD 相关寄存器表

8.4.1.1 LPD 控制寄存器 LPD_CON

寄存器	读/写	说明	复位值	上电初始值
LPD_CON	读/写	LPD 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
LPDEN	-	-	LPDIS[2:0]			LPDIF	LPDOF

位编号	位符号	说明
7	LPDEN	LPD 启动控制位 0: LPD 不启动 1: LPD 启动
4~2	LPDIS[2:0]	LPD 门限电压档位选择, V_{LPD} 为 LPD 门限电压阈值: 000: $V_{LPD} = 1.85V$ 001: $V_{LPD} = 2.05V$ 010: $V_{LPD} = 2.25V$ 011: $V_{LPD} = 2.45V$ 100: $V_{LPD} = 2.65V$ 101: $V_{LPD} = 2.85V$ 110: $V_{LPD} = 3.05V$ 111: $V_{LPD} = 3.25V$
1	LPDIF	LPD 中断请求标志。 当 V_{DD} 电压值下降至低于 V_{LPD} 或上升至高于 V_{LPD} 时, 该位由硬件置“1”, 如果此时 LPD_IDE.INTEN=1, 将产生中断。 该位由软件写 1 清 0。
0	LPDOF	LPD 状态标志位 0: V_{DD} 高于 LPD 门限电压 (由硬件清零) 1: V_{DD} 低于 LPD 门限电压 (由硬件置位)

位编号	位符号	说明
31~8 6~5	-	保留

8.4.1.2 LPD 中断使能寄存器 LPD_IDE

寄存器	读/写	说明	复位值	上电初始值
LPD_IDE	读/写	LPD 中断使能寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
0	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
31~1	-	保留

8.4.2 LPD 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
LPD 基地址: 0x4002_2190					
LPD_CON	0x00	读/写	LPD 控制寄存器	0x0000_0000	0x0000_0000
LPD_IDE	0x04	读/写	LPD 中断使能寄存器	0x0000_0000	0x0000_0000

9 GPIO

9.1 时钟源

M0+内核可通过 IOPORT 总线实现单周期访问 GPIO，数据传输效率极高。IOPORT 总线的时钟来自 HCLK。

9.2 特性

SC32L14T/14G 系列 GPIO 端口特性如下：

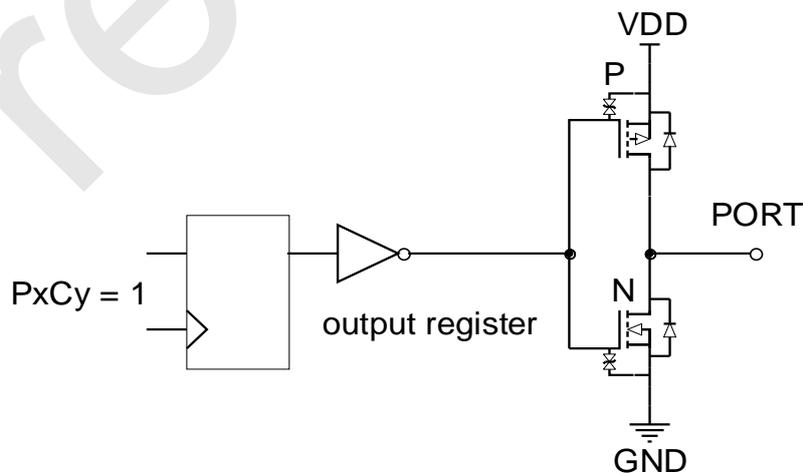
- 最大 77 个双向可独立控制的 GPIO
- CPU 可在单周期内通过 IOPORT 总线访问 GPIO 端口
- 可独立设定上拉电阻
- 所有口源驱动能力分四级控制
- 全部 GPIO 具有大灌电流驱动能力 (50mA@ $V_{PIN}=0.8V$)
- 16 个 GPIO 一组
- I/O 端口在输入或输出状态下，从端口数据寄存器里读到的都是端口的实际状态值

注意：未使用及封装未引出的端口均要设置为强推挽输出模式。

9.3 GPIO 结构图

9.3.1 强推挽输出模式

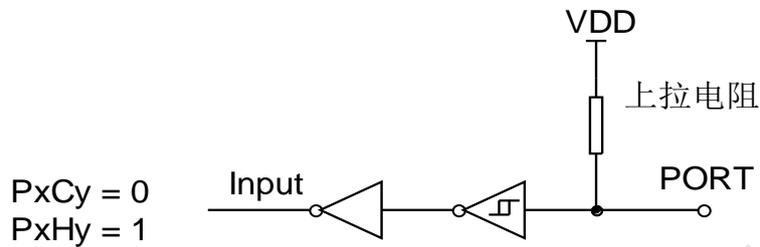
强推挽输出模式下，能够提供持续的大电流驱动：相关电气参数详见 [《GPIO 参数》](#) 章节。



强推挽输出模式的端口结构示意图

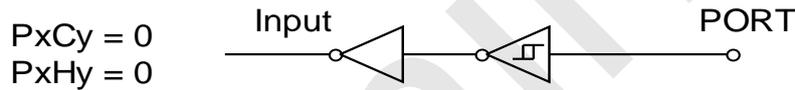
9.3.2 带上拉的输入模式

带上拉的输入模式下，输入口上恒定接一个上拉电阻，仅当输入口上电平被拉低时，才会检测到低电平信号。



带上拉输入模式的端口结构示意图

9.3.3 高阻输入模式(Input only)



高阻输入模式的端口结构示意图

9.4 GPIO 寄存器

9.4.1 GPIO 相关寄存器表

9.4.1.1 PX 端口数据寄存器 PX

寄存器	读/写	说明	复位值	上电初始值
PX X=A,B,C,D,E	读/写	PX 端口数据寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8
7	6	5	4	3	2	1	0
PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

位编号	位符号	说明
15~0	PD _n (n=0~15)	PX _n 端口数据寄存器, X=A,B,C,D,E, n=0~15 端口锁存寄存器数据, 从端口数据寄存器里读到的都是端口的实际状态值。
31~16	-	保留

9.4.1.2 PX 端口数据寄存器 PX_n_BIT

寄存器	读/写	说明	复位值	上电初始值
PX _n _BIT X=A,B,C,D,E n=0~15	读/写	PX 端口数据寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
0	BSR _n	PX _n 端口位赋值控制, n=0~15 用于对 PX _n 端口单独赋值
31~1	-	保留

9.4.1.3 PX 端口数据寄存器 PX_n_XR

寄存器	读/写	说明	复位值	上电初始值
PX _n _XR X=A,B,C,D,E n=0~15	读/写	对 PX _n 翻转	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
0	XR _n	PX _n 端口位翻转控制, n=0~15 0: 无效 1: 对 PX _n 输出进行翻转

位编号	位符号	说明
31~1	-	保留

9.4.1.4 PX 端口输入/输出控制寄存器 PXCON

寄存器	读/写	说明	复位值	上电初始值
PXCON X=A,B,C,D,E	读/写	PX 端口输入/输出控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
MODE15	MODE14	MODE13	MODE12	MODE11	MODE10	MODE9	MODE8
7	6	5	4	3	2	1	0
MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0

位编号	位符号	说明
15~0	MODEn (n=0~15)	PXn 端口强推挽模式使能位, n=0~15 0: PXn 为输入模式 (上电初始值) 1: PXn 为强推挽输出模式
31~16	-	保留

9.4.1.5 PX 端口上拉电阻控制寄存器 PXPB

寄存器	读/写	说明	复位值	上电初始值
PXPB X=A,B,C,D,E	读/写	PX 端口上拉电阻控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
PUPD15	PUPD14	PUPD13	PUPD12	PUPD11	PUPD10	PUPD9	PUPD8
7	6	5	4	3	2	1	0
PUPD7	PUPD6	PUPD5	PUPD4	PUPD3	PUPD2	PUPD1	PUPD0

位编号	位符号	说明
15~0	PUPDn (n=0~15)	PXn 端口上拉电阻使能位, n=0~15 0: PXn 为高阻输入模式 (上电初始值), 上拉电阻关闭; 1: PXn 上拉电阻打开
31~16	-	保留

9.4.1.6 GPIO 驱动等级寄存器 PXLEV

寄存器	读/写	说明	复位值	上电初始值
PXLEV X=A,B,C,D,E	读/写	GPIO 驱动等级寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
LEV15[1:0]		LEV14[1:0]		LEV13[1:0]		LEV12[1:0]	
23	22	21	20	19	18	17	16
LEV11[1:0]		LEV10[1:0]		LEV9[1:0]		LEV8[1:0]	
15	14	13	12	11	10	9	8
LEV7[1:0]		LEV6[1:0]		LEV5[1:0]		LEV4[1:0]	
7	6	5	4	3	2	1	0
LEV3[1:0]		LEV2[1:0]		LEV1[1:0]		LEV0[1:0]	

位编号	位符号	说明
31~0	LEVn[1:0] (n=0~15)	PXn 端口等级控制位, n=0~15 用于设置 PXn 端口的 I _{OH} 等级: 00: 等级 0 (最大); 01: 等级 1; 10: 等级 2; 11: 等级 3 (最小);

9.4.2 GPIO 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
PA 基地址: 0x4001_1000					
PA	0x00	读/写	PA 端口数据寄存器	0x0000_0000	0x0000_0000
PACON	0x20	读/写	PA 口输入/输出控制寄存器	0x0000_0000	0x0000_0000
PAPH	0x40	读/写	PA 口上拉电阻控制寄存器	0x0000_0000	0x0000_0000
PALEV	0x60	读/写	IOH 设置寄存器 IO 驱动等级	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
PB 基地址: 0x4001_1100					
PB	0x00	读/写	PB 端口数据寄存器	0x0000_0000	0x0000_0000
PBCON	0x20	读/写	PB 口输入/输出控制寄存器	0x0000_0000	0x0000_0000
PBPH	0x40	读/写	PB 口上拉电阻控制寄存器	0x0000_0000	0x0000_0000
PBLEV	0x60	读/写	IOH 设置寄存器 IO 驱动等级	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
PC 基地址: 0x4001_1200					
PC	0x00	读/写	PC 端口数据寄存器	0x0000_0000	0x0000_0000
PCCON	0x20	读/写	PC 口输入/输出控制寄存器	0x0000_0000	0x0000_0000
PCPH	0x40	读/写	PC 口上拉电阻控制寄存器	0x0000_0000	0x0000_0000
PCLEV	0x60	读/写	IOH 设置寄存器 IO 驱动等级	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
PD 基地址: 0x4001_1300					
PD	0x00	读/写	PD 端口数据寄存器	0x0000_0000	0x0000_0000
PDCON	0x20	读/写	PD 口输入/输出控制寄存器	0x0000_0000	0x0000_0000
PDPH	0x40	读/写	PD 口上拉电阻控制寄存器	0x0000_0000	0x0000_0000
PDLEV	0x60	读/写	IOH 设置寄存器 IO 驱动等级	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
PE 基地址: 0x4001_1400					
PE	0x00	读/写	PE 端口数据寄存器	0x0000_0000	0x0000_0000
PECON	0x20	读/写	PE 口输入/输出控制寄存器	0x0000_0000	0x0000_0000
PEPH	0x40	读/写	PE 口上拉电阻控制寄存器	0x0000_0000	0x0000_0000
PELEV	0x60	读/写	IOH 设置寄存器 IO 驱动等级	0x0000_0000	0x0000_0000

10 模数转换器 ADC

10.1 时钟源

- SC32L14T/14G 系列的 ADC 的时钟源仅一种，来自 PCLK
- ADC 的转换时间约为 950ns

10.2 特性

- 精度：14 位
- 23 路 ADC 采样通道：
 - 外部 20 路 ADC 采样通道和 IO 口的其它功能复用
 - 内部一路 ADC 可直接测量 V_{DD} 电压
 - 内部一路 ADC 可直接测量 OP 输出
 - 内部一路温度采样通道
- 内建 2.4V、2.048V 和 1.024V 三种基准电压
- ADC 的参考电压有 4 种选择： V_{DD} 、2.4V、2.048V、1.024V
- ADC 输入通道选择
 - 可通过设置 ADCIS[4:0]位，选择 ADC 输入通道
- 可通过软件启动转换过程
- 可设 ADC 转换完成中断
- 采样到完成转换的总时间低至 2 μ s
- 支持 DMA 传输：ADC 转换完成可产生 DMA 请求
- 支持单通道连续转换模式
- ADC 转换结果支持溢出标志提醒，且 OVERRUN 标志位与 ADC 转换结果在同一寄存器 ADCV，用户可一次性读取

10.3 ADC 转换步骤

用户实际进行 ADC 转换所需要的操作步骤如下：

- ① 设定 ADC 输入管脚；（设定 AINx 对应的位为 ADC 输入，通常 ADC 管脚会预先固定）；
- ② 设定 ADC 参考电压 Vref，设定 ADC 转换所用的频率；
- ③ ADCEN 写 1，开启 ADC 模块电源；
- ④ 选择 ADC 输入通道；（设置 ADCIS 位，选择 ADC 输入通道）；
- ⑤ 启动 ADCS，转换开始；
- ⑥ 等待 ADCIF=1，如果 ADC 中断使能，则 ADC 中断会产生，用户需要软件清 0 ADCIF 标志；
- ⑦ 从 ADCV 获得 14 位数据，一次转换完成；
- ⑧ 如不更换输入通道，可通过软件将 CONT 置 1，设定单通道连续转换模式。转换将持续进行，直到该位清零。
- ⑨ ADC 转换结果溢出时，OVERRUN 标志位置 1。
- ⑩ 可通过 DMA 传输转换数据。

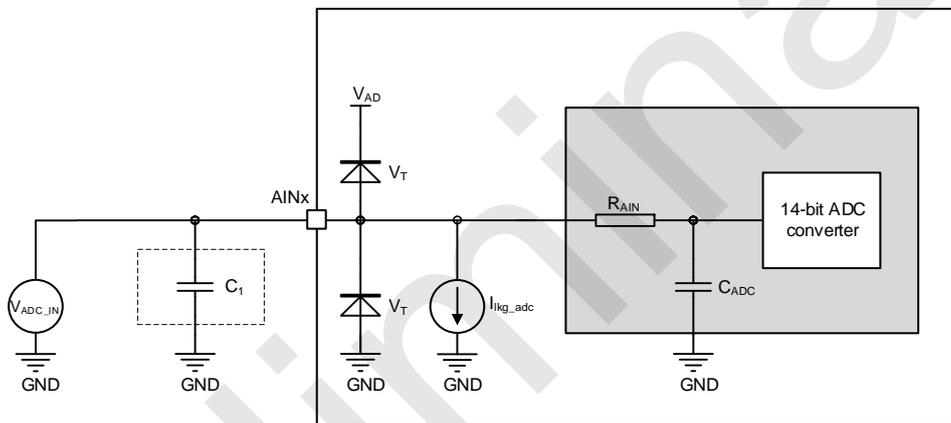
注意：在设定 ADC_CON[8](INTEN)前, 使用者最好用软件先清除 ADCIF, 并且在 ADC 中断服务程序执行完时, 也清除该 ADCIF, 以避免不断的产生 ADC 中断。

10.4 ADC 中断

SC32L14T/14G 系列的 ADC 在转换完成后, ADCIF 将置起, 如果 ADC_CON.INTEN=1, 将产生中断。

中断事件	事件标志位	中断使能控制位
ADC 转换完成中断请求	ADCIF	ADC_CON->INTEN

10.5 ADC 连接电路图



说明:

- C1 为外接 0.01 μ F 电容, 建议用户增加此电容以提升 ADC 性能;
- ADC 相关电气参数详见规格书《SC32L14T/14G_Datasheet》。

10.6 ADC 寄存器

10.6.1 ADC 相关寄存器表

10.6.1.1 ADC 控制寄存器 ADC_CON

寄存器	读/写	说明	复位值	上电初始值
ADC_CON	读/写	ADC 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	VREFS[1:0]		-	LOWSP[2:0]		
15	14	13	12	11	10	9	8
ADCEN	-	-	DMAEN	CONT	-	-	INTEN
7	6	5	4	3	2	1	0

ADCS	-	-	ADCIS[4:0]
------	---	---	------------

位编号	位符号	说明
21~20	VREFS[1:0]	参考电压选择控制位 00: 选择 ADC 的 V_{REF} 为 V_{DD} ; 01: 选择 ADC 的 V_{REF} 为内部准确的 2.048V; 10: 选择 ADC 的 V_{REF} 为内部准确的 1.024V; 11: 选择 ADC 的 V_{REF} 为内部准确的 2.4V;
18~16	LOWSP[2:0]	ADC 采样周期数选择控制位 100: 采样时间为 3 个系统时钟, (约 125ns @ $f_{PCLK2} = 24MHz$) 101: 采样时间约 6 个系统时钟, (约 250ns @ $f_{PCLK2} = 24MHz$) 110: 采样时间约 16 个系统时钟, (约 667ns @ $f_{PCLK2} = 24MHz$) 111: 采样时间约 32 个系统时钟, (约 1333ns @ $f_{PCLK2} = 24MHz$) 其它: 保留 说明: ADC 从采样到完成转换的总时间计算方式如下: $T_{ADC} = \text{采样时间} + \text{转换时间}$ 其中, ADC 转换时间约为 950ns
15	ADCEN	ADC 模块电源启动控制位 0: 关闭 ADC 模块电源 1: 开启 ADC 模块电源
12	DMAEN	DMA 请求使能控制位 该位用于使能 DMA 请求的生成。写 1 后可使用 DMA 控制器自动管理 ADC 转换的数据。 0: 禁止 DMA 请求 1: 使能 DMA 请求 注意: 通过软件对此位执行写操作时, 应确保当前未进行任何转换。
11	CONT	单次/连续转换模式选择位 该位由软件置 1 和清 0。该位置 1 时, 转换将持续进行, 直到该位清 0。 0: 单次转换模式 1: 连续转换模式
8	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
7	ADCS	ADC 转换触发控制位 该位是 ADC 转换的触发信号, 由软件置 1, 由硬件清 0。对此位写 1 会触发一次 ADC 的转换。 注意: 对 ADCS 置 1 之后, 到中断标志 ADCIF 置起前不要对 ADC_CON 寄存器进行写操作。
4~0	ADCIS[4:0]	ADC 输入通道选择位 00000: 选择 AIN0 为 ADC 的输入 00001: 选择 AIN1 为 ADC 的输入 00010: 选用 AIN2 为 ADC 的输入 00011: 选用 AIN3 为 ADC 的输入 00100: 选用 AIN4 为 ADC 的输入 00101: 选用 AIN5 为 ADC 的输入 00110: 选用 AIN6 为 ADC 的输入 00111: 选择 AIN7 为 ADC 的输入 01000: 选择 AIN8 为 ADC 的输入 01001: 选择 AIN9 为 ADC 的输入 01010: 选择 AIN10 为 ADC 的输入 01011: 选择 AIN11 为 ADC 的输入 01100: 选择 AIN12 为 ADC 的输入 01101: 选择 AIN13 为 ADC 的输入

位编号	位符号	说明
		01110: 选择 AIN14 为 ADC 的输入 01111: 选择 AIN15 为 ADC 的输入 10000: 选择 AIN16 为 ADC 的输入 10001: 选择 AIN17 为 ADC 的输入 10010: 选择 AIN18 为 ADC 的输入 10011: 选择 AIN19 为 ADC 的输入 10100: 选择温度采样 为 ADC 的输入 10101~11101: 保留 11110: 选择 OP 输出为 ADC 的输入 11111: ADC 输入为 1/4 V _{DD} , 可用于测量电源电压
31~22 19 14~13 10~9 6~5	-	保留

10.6.1.2 ADC 标志状态位寄存器 ADC_STS

寄存器	读/写	说明	复位值	上电初始值
ADC_STS	读/写	ADC 转换数值寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
0	ADCIF	ADC 中断请求标志位 该位由硬件置 1，通过软件写 1 清 0。 ADC 转换完成后，该位由硬件置 1，如果 ADC_CON.INTEN=1，将产生中断。
31~1	-	保留

10.6.1.3 ADC 转换数值寄存器 ADCV

寄存器	读/写	说明	复位值	上电初始值
ADCV	只读	ADC 转换数值寄存器	0x0000_3FFF	0x0000_0000

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

ADCV[7:0]		
位编号	位符号	说明
31	OVERRUN	溢出标志位（只读） 当 CPU 或 DMA 无法及时处理 ADC 转换请求而发生溢出时，该位由硬件置 1，读取 ADCV 后，该位自动清 0。 注意：产生溢出时，上一次的 ADC 转换结果将被新的 ADC 转换结果覆盖。
13~0	ADCV[13:0]	14 位的 ADC 转换结果
30~14	-	保留

10.6.1.4 ADC 端口设置寄存器 ADC_CFG

寄存器	读/写	说明	复位值	上电初始值
ADC_CFG	读/写	ADC 端口设置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	AINTEMP	AIN19	AIN18	AIN17	AIN16
15	14	13	12	11	10	9	8
AIN15	AIN14	AIN13	AIN12	AIN11	AIN10	AIN9	AIN8
7	6	5	4	3	2	1	0
AIN7	AIN6	AIN5	AIN4	AIN3	AIN2	AIN1	AIN0

位编号	位符号	说明
20	AINTEMP	温度采样输入通道 0: 关闭温度采样输入通道 1: 开启温度采样输入通道 说明：此位仅在 ADCIS[4:0]=10100 选择温度采样输入通道时需要开启，开启后温度采样输入有效。
19~0	AINx (x=0~19)	ADC 端口设置寄存器 0: AINx 对应端口不可作为 ADC 输入通道 1: AINx 对应端口可作为 ADC 输入通道，当 ADCIS[4:0]选择 AINx 作为 ADC 输入通道时，AINx 对应端口的上拉电阻将自动移除。
31~21	-	保留

10.6.2 ADC 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
ADC 基地址：0x4002_2110					
ADC_CON	0x00	读/写	ADC 控制寄存器	0x0000_0000	0x0000_0000
ADC_STS	0x04	读/写	ADC 标志状态位寄存器	0x0000_0000	0x0000_0000
ADCV	0x08	读/写	ADC 转换数值寄存器	0x0000_3FFF	0x0000_0000
ADC_CFG	0x0C	读/写	ADC 端口设置寄存器	0x0000_0000	0x0000_0000

11 温度传感器

11.1 概述

SC32L14T/14G 内建一个温度传感器，可通过 ADC 电路测量温度传感器电压。

11.2 温度传感器操作步骤

使用温度传感器时，ADC 参考电压选择内部 2.4V 作为参考，温度传感器每增加 1°C，ADC 转换值会增加固定值。赛元出厂时已将每颗芯片 25°C 对应的 ADC 转换结果写入到对应地址中。

用户使用温度传感器的操作步骤如下：

- ① 设定 ADC 参考电压 Vref 为内部 2.4V 基准源，设定 ADC 采样周期，建议选择 60 个以上采样时钟，之后开启 ADC 模块电源；
- ② 选择 ADC 输入通道为温度传感器通道；
- ③ 使能温度传感器，TS_EN 写 1；
- ④ 延时 20μs
- ⑤ TS_CHOP 写 0，启动 ADC 转换，一次转换完成，记录转换值 ADC_{Value1}；
- ⑥ TS_CHOP 写 1，启动 ADC 转换，一次转换完成，记录转换值 ADC_{Value2}；
- ⑦ 将两次转换值求平均：

$$ADC_{Value} = \frac{(ADC_{Value1} + ADC_{Value2})}{2}$$

- ⑧ 从对应地址读取出厂时写入的 25 摄氏度 ADC 转换值 ADC_{ValueTest}；
- ⑨ 代入公式计算得到当前温度：

$$Temperature = 25^{\circ}C + \frac{(ADC_{Value} - ADC_{ValueTest})}{33}$$

用户若需获取更多温度传感器相关信息，请参考《赛元 SC32L14T/14G 系列 MCU 应用指南》

11.3 温度传感器寄存器

11.3.1 温度传感器相关寄存器表

11.3.1.1 温度传感器设置寄存器 TS_CFG

寄存器	读/写	说明	复位值	上电初始值
TS_CFG	读/写	温度传感器设置寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7	TS_EN	温度传感器使能控制位 0: 关闭温度传感器 1: 使能温度传感器
0	TS_CHOP	温度传感器的抵消 offset 的应用控制位 TS_CHOP 写 0 后启动一次 ADC 转换得到一个数值，TS_CHOP 再写 1 后启动一次 ADC 转换得到第二个数值，两个数做平均得到最终数

11.3.2 温度传感器寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
温度传感器基地址: 0x4002_21E0					
TS_CFG	0x00	读/写	温度传感器设置寄存器	0x0000_0000	0x0000_0000

12 运放及可编程增益放大器（OP）

12.1 概述

内建一个运放及可编程增益放大器 OP，提供轨到轨运放，且 OP 可配置为 PGA 模式，有 5 个同相输入端，2 个反相输入端以及 3 个输出端，并可提供 8/16/32/64 同相增益，7/15/31/63 倍反相增益。

12.2 特性

内建一个可变增益放大器，此 OP 特性如下：

- 一个 Rail-to-Rail 的运放
- 可配置为 PGA 模式
 - 同相输入增益：8/16/32/64
 - 反相输入增益：7/15/31/63
- 同相输入端引出两个外部引脚：OP_P0 或 OP_P1
- 反相输入端引出一个外部引脚：OP_N
- 输出端引出一个外部引脚：OP_O
- 输出端可直接接入 ADC 输入
- 输出端可直接接入 CMP 正端
- 可通过设置 PGA 输入端 offset 调整控制位 PGOFC=1，将 OP 模块的同相端与反相端输入短接来实现精度调整

12.3 OP 端口选择

12.3.1 OP 精度调整

可通过设置 PGA 输入端 offset 调整控制位 PGOFC=1，将 OP 模块的同相端与反相端输入短接来实现精度调整。其他情况下，PGOFC 设置为 0。

12.3.2 OP 同相端输入选择

OP 模块的同相端输入有五种：OP_P0 外部引脚、OP_P1 外部引脚、内部 VSS、内部 1.2V 基准和 VDD，可通过 OPPSEL[2:0] 切换选择。

12.3.3 OP 反相端输入选择

OP 模块的反相端输入有两种：OP_N 外部引脚和内部反馈电阻。

- 选择 OP_N 外部引脚为 OP 的反相端输入时，需设置 OP 输入控制位 OPNSEL=0，反馈电阻端连接选择位 FDBRSEL[1:0]=01。
- 选择内部反馈电阻为 OP 的反相端输入时，需设置 OPNSEL=1，FDBRSEL[1:0]=00 或 11 或 10，并通过内部增益档位选择位 PGAGAN[1:0] 进行内部增益档位选择。

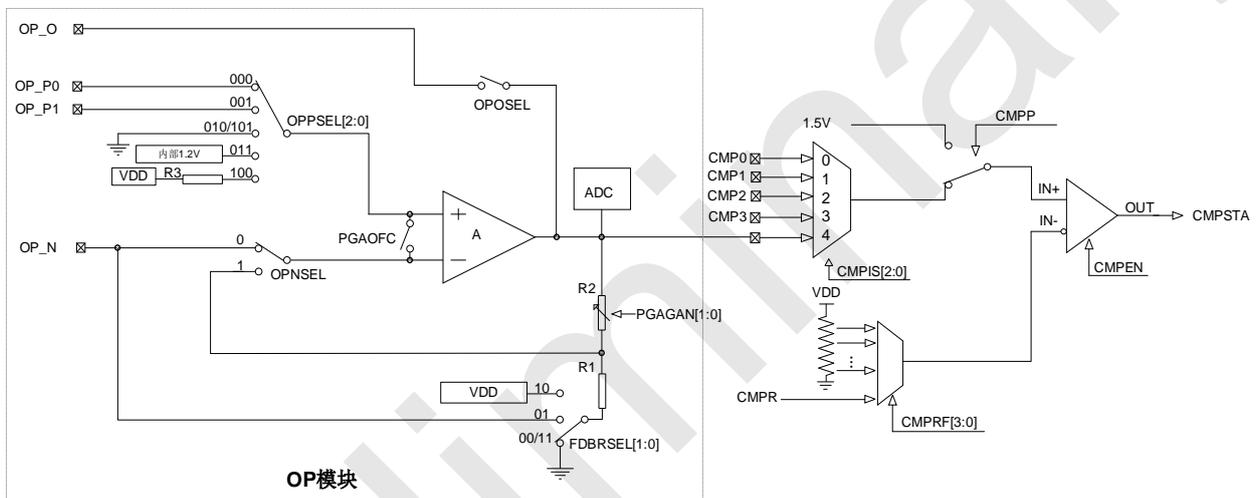
12.3.4 OP 输出选择

OP 模块的输出有三种：有 3 个输出端：AD 转换器的采样通道、CMP 正端输入和 OP_O 引脚。

OP 输出用于 AD 转换器的模拟输入或者 CMP 正端的模拟输入时，具体设置方式如下：

- OP 作为 ADC 输入时，需要先设置 ENOP=1，使能 OP 模块，再设置 ADCEN=1，开启 ADC 电源，通过 ADCIS[4:0]选择 OP 输出端为 ADC 输入端，则 OP 的转换结果可直接在 ADCV 寄存器获取。
- OP 作为 CMP 正端输入时，当 OP 使能时，如果 CMP 的输入通道控制位 CMPIS[2:0]设置为 OP，则选用 OP 输出为 CMP 的正端输入。

12.4 OP 电路结构框图



12.5 OP 寄存器

12.5.1 OP 相关寄存器表

12.5.1.1 OP 控制寄存器 OP_CON

寄存器	读/写	说明	上电初始值	复位值
OP_CON	读/写	OP 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	TRIMOFFSETN[4:0]				-
23	22	21	20	19	18	17	16
PGAOFc	-	-	TRIMOFFSETP[4:0]				-
15	14	13	12	11	10	9	8
-	-	FDBRSEL[1:0]		-	-	PGAGAN[1:0]	
7	6	5	4	3	2	1	0
OPPSEL[2:0]		-	OPNSEL	-	OPOSEL	ENOP	

位编号	位符号	说明
28~24	TRIMOFFSETN[4:0]	Trim for NMOS differential pairs 运放的 NMOS 差分 offset 校准值
23	PGAOFC	PGA 输入端 offset 调整控制位 0: OP 同相和反相输入端不短接 1: OP 同相和反相输入端在内部短接 (注: OP 同相和反相输入端在内部短接与断开均不会影响 OPPSEL 和 OPNSEL 选择)
20~16	TRIMOFFSETP[4:0]	Trim for PMOS differential pairs 运放的 PMOS 差分 offset 校准值
13~12	FDBRSEL[1:0]	反馈电阻端连接选择位 00/11: 接内部接 VSS 01: OP_N 端口 10: VDD
9~8	PGAGAN[1:0]	内部增益档位选择: 00: 同相 8, 反相 7 01: 同相 16, 反相 15 10: 同相 32, 反相 31 11: 同相 64, 反相 63
7~5	OPPSEL[2:0]	OP 同相端信号连接选择 000: OP_P0 (外部引脚) 001: OP_P1 (外部引脚) 010: 内部接 VSS, 0V 011: 接内部 1.2V 基准 100: VDD 101: 内部接 VSS, 0V
3	OPNSEL	OP 反相端信号连接选择 0: 选择 OP_N (外部引脚) 1: 选择内部反馈电阻。
1	OPOSEL	OP 输出端连接选择位 0: 与 OP_O 的连接断开 1: OP 输出连接到 OP_O (外部引脚)
0	ENOP	OP 模块电源使能位 0: 关闭 OP 模块电源 1: 使能 OP 模块电源
31~29 22~21 15~14 11~10 4, 2	-	保留

12.5.2 OP 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
OP 基地址: 0x4002_2140					
OP_CON	0x00	读/写	OP 控制寄存器	0x0000_0000	0x0000_0000

13 模拟比较器 CMP

SC32L14T/14G 系列内建一个模拟比较器（CMP），CMP 中断可唤醒 STOP 模式。可用于报警器电路、电源电压监测电路、过零检测电路等。

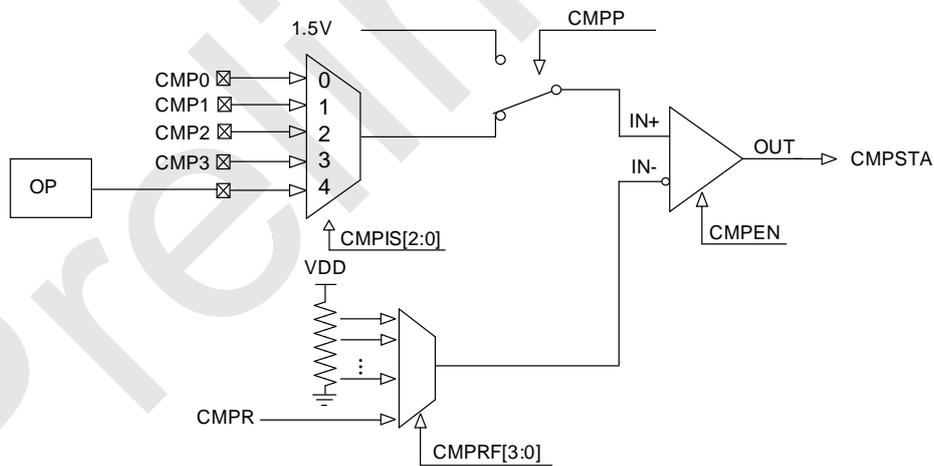
此比较器具有五个模拟信号正输入端：CMP0~3 以及 OP 输出端，可通过 CMPIS [2:0] 切换选择。负输入端电压可通过 CMPRF[3:0] 切换为 CMPR 脚上的外部电压或内部的 15 档比较电压中的一种。

通过 CMPIM[1:0] 可以方便的设定比较器的中断模式，当 CMPIM[1:0] 所设定的中断条件发生时比较器中断标志 CMPIF 会被置 1，该中断标志需要软件清除。

13.1 特性

- 正端输入信号五种可选：
 - 四个模拟信号正输入端 CMP0~CMP3
 - OP 输出端
- 负端输入电压可选择 CMPR 引脚或内部 V_{DD} 分压的 15 档比较电压中的一种
- CMP 中断可唤醒 STOP Mode

13.2 模拟比较器结构框图



模拟比较器结构框图

13.3 CMP 寄存器

13.3.1 CMP 相关寄存器表

13.3.1.1 模拟比较器状态寄存器 CMP_STS

寄存器	读/写	说明	复位值	上电初始值
CMP_STS	读/写	模拟比较器状态寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
1	CMPSTA	模拟比较器输出状态位 0: 比较器正端电压小于负端电压 1: 比较器正端电压大于负端电压
0	CMPIF	模拟比较器中断标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 比较器中断未被触发； 1: 当比较器满足中断触发条件时，此位会被硬件置 1。如果此时 CMPIM[1:0]不为 00，比较器中断产生。
31~2	-	保留

13.3.1.2 模拟比较器配置寄存器 CMP_CFG

寄存器	读/写	说明	复位值	上电初始值
CMP_CFG	读/写	模拟比较器配置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	CMPRF[3:0]			
7	6	5	4	3	2	1	0
CMPEN	CMPIM[1:0]		CMPP	-	CMPIS[2:0]		

位编号	位符号	说明
11~8	CMPRF[3:0]	模拟比较器负端比较电压选择位 模拟比较器负端比较电压设置项如下： 0000: CMPR; 0001: 1/16V _{DD} 0010: 2/16V _{DD} 0011: 3/16V _{DD} 0100: 4/16V _{DD} 0101: 5/16V _{DD} 0110: 6/16V _{DD} 0111: 7/16V _{DD} 1000: 8/16V _{DD} 1001: 9/16V _{DD} 1010: 10/16V _{DD} 1011: 11/16V _{DD} 1100: 12/16V _{DD} 1101: 13/16V _{DD} 1110: 14/16V _{DD} 1111: 15/16V _{DD}
7	CMPEN	模拟比较器使能位 0: 关闭模拟比较器 1: 使能模拟比较器
6~5	CMPIM[1:0]	模拟比较器中断模式选择位 00: 不产生中断 01: 上升沿中断: IN+从小于 IN- 到大于 IN- 后会产生中断; 10: 下降沿中断: IN+从大于 IN- 到小于 IN- 后会产生中断; 11: 双沿中断: IN+从小于 IN- 到大于 IN- , 或 IN+从大于 IN- 到小于 IN-后均会产生中断;
4	CMPP	模拟比较器正端输入选择: 0: 模拟比较器正端的输入为 CMP0~3 之一为, 由 CMPIS[1:0]设定 1: 模拟比较器正端的输入为内部 1.5V 基准电压
2~0	CMPIS[2:0]	模拟比较器正端输入通道选择位 CMPP 为 1 时, 该位无效: 000: 选用 CMP0 为模拟比较器正端的输入; 001: 选用 CMP1 为模拟比较器正端的输入; 010: 选用 CMP2 为模拟比较器正端的输入; 011: 选用 CMP3 为模拟比较器正端的输入; 100~110: 不选 111: 选用 PGA 输出端为模拟比较器的输入
31~12 3	-	保留

13.3.2 CMP 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
CMP 基地址: 0x4002_2130					
CMP_STS	0x00	读/写	模拟比较器状态寄存器	0x0000_0000	0x0000_0000
CMP_CFG	0x04	读/写	模拟比较器配置寄存器	0x0000_0000	0x0000_0000

14 高级加密运算（AES）

14.1 概述

SC32L14T/14G 系列内置的高级加密运算模块（AES）可对数据进行加密或解密，支持多种链接模式（ECB、CBC、CTR），支持 128/192/256 位密钥大小。

14.2 时钟源

SC32L14T/14G 系列 AES 的时钟源来自 HCLK

14.3 特性

- 以 128 位数据为一个数据块进行处理
- 支持全密钥长度：128/192/256 位
- 支持多种链接模式的加密和解密：
 - 电子密码本模式（ECB）
 - 密码分组链接模式（CBC）
 - 计数器模式（CTR）

14.4 操作步骤

AES 的加解密具体操作步骤请参考《赛元 SC32 AES 应用指南》。

14.5 AES 中断

SC32L14T/14G 系列的 AES 在加解密完成时，AES 完成中断标志位 CCFIF 会置起，若此时 CCFIE=1，将会产生中断。

中断事件	事件标志位	中断使能控制位
AES 加解密完成	CCFIF	AES_CONFIG->CCFIE

14.6 AES 寄存器

14.6.1 AES 相关寄存器表

14.6.1.1 AES 配置寄存器

寄存器	读/写	说明	复位值	上电初始值
AES_CONFIG	读/写	AES 配置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
MODE	-	-	-	DRDYIE	TINTEN	CCFIE	-
7	6	5	4	3	2	1	0
TRNGEN	TLSEL[2:0]			TLDIV[2:0]			-

位编号	位符号	说明
15	MODE	AES 工作模式选择控制位 0: AES 加密模式 1: AES 解密模式
9	CCFIE	AES 中断使能控制位 0: CCFIF 置起时禁止产生中断 1: CCFIF 置起时允许产生中断
31~16 14~12 8, 0	-	保留

14.6.1.2 AES 标志寄存器

寄存器	读/写	说明	复位值	上电初始值
AES_STS	读/写	AES 标志寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
1	CCFIF	AES 计算完成标志位 该位由硬件置 1，通过软件写 1 清 0。 0: AES 未计算完成 1: AES 已计算完成

位编号	位符号	说明
0	BUSY	AES 计算中状态位 0: 空闲 1: 正在计算
31~4	-	保留

14.6.2 AES 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
AES 基地址: 0x4001_0400					
AES_CONFIG	0x10	读/写	AES 配置寄存器	0x0000_0000	0x0000_0000
AES_STS	0x18	读/写	AES 标志寄存器	0x0000_0000	0x0000_0000

15 真随机数发生器 (TRNG)

15.1 概述

SC32L14T/14G 系列内部集成一个真随机数发生器 (简称 TRNG)，可以产生 32 位的真随机数，随机数由输入时钟经过模拟熵源生成。

TRNG 真随机数发生器已按照德国 BSI AIS-31 统计测试 (T0 到 T8) 进行了测试。

15.2 时钟源

SC32L14T/14G 系列 TRNG 的时钟源仅一种：来自 HCLK

15.3 特性

- 可产生 32 位的真随机数
- 若随机性较低可触发种子错误中断
- 提供一个随机数产生完毕中断

15.4 TRNG 操作步骤

使能 TRNGEN 后，随机数将开始产生。用户可通过读取 TRNG_DATA 寄存器获取 32 位随机数，当有新的随机数产生时，TRNG_DATA 中旧的随机数会被新的随机数覆盖。

15.5 TRNG 中断

SC32L14T/14G 系列的 TRNG 在随机数生成完毕时，生成完毕标志位 DRDYIF 将会置起，若此时 TRNG_CONFIG->TINTEN=1 且 DRDYIE=1，将会产生中断。

当 TRNG 产生随机数被硬件判断为随机性较低时，会置起种子错误标志位 SEIF，若此时 TRNG_CONFIG->TINTEN=1，将会产生中断。

中断事件	中断请求控制位	事件标志位	中断使能子开关
TRNG 随机数生成完毕	TRNG_CONFIG->TINTEN	DRDYIF	DRDYIE
TRNG 发生种子错误		SEIF	\

15.6 TRNG 熵源验证

15.6.1 简介

为了评估 TRNG 可提供的熵大小，赛元按照德国 BSI AIS-31 统计测试 (T0 到 T8) 对外设进行了测试。测试结果可根据需要提供，客户也可重现测试结果。

15.6.2 验证条件

赛元在以下条件对 TRNG 真随机数发生器进行了测试，推荐用户使用时按照以下方法设置：

TRNG 时钟 $f_{HCLK}=24\text{MHz}$ ， $TLSEL[2:0]=000\sim 011$ ， $TLDIV[2:0]=101$ 。

15.7 TRNG 寄存器

15.7.1 TRNG 相关寄存器表

15.7.1.1 TRNG 配置寄存器 TRNG_CONFIG

寄存器	读/写	说明	复位值	上电初始值
TRNG_CONFIG	读/写	TRNG 配置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
MODE	-	-	-	DRDYIE	TINTEN	CCFIE	-
7	6	5	4	3	2	1	0
TRNGEN	TLSEL[2:0]			TLDIV[2:0]			-

位编号	位符号	说明
11	DRDYIE	TRNG 数据生成完毕中断使能位 0: DRDYIF 置起时禁止产生中断 1: DRDYIF 置起时允许产生中断
10	TINTEN	TRNG 总中断使能位 0: 中断请求屏蔽 1: 中断请求开启
7	TRNGEN	TRNG 模块使能位 0: TRNG 模块关闭 1: TRNG 模块开启
6~4	TLSEL[2:0]	TRNG 随机数产生方式选择 建议参考 15.6 TRNG 熵源验证 设置
3~1	TLDIV[2:0]	TRNG 随机数产生速率选择 写值越大，随机数产生速率越低，建议参考 15.6 TRNG 熵源验证 设置
31~16 14~12 0	-	保留

15.7.1.2 TRNG 数据寄存器 TRNG_DATA

寄存器	读/写	说明	复位值	上电初始值
TRNG_DATA	读/写	TRNG 数据寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
TRNG_DATA[31:24]							
23	22	21	20	19	18	17	16
TRNG_DATA[23:16]							
15	14	13	12	11	10	9	8

TRNG_DATA[15:8]							
7	6	5	4	3	2	1	0
TRNG_DATA[7:0]							

位编号	位符号	说明
31~0	TRNG_DATA	真随机数数据寄存器

15.7.1.3 TRNG 标志寄存器 TRNG_STS

寄存器	读/写	说明	复位值	上电初始值
TRNG_STS	读/写	TRNG 标志寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
3	DRDYIF	TRNG 数据生成完毕标志位 该位通过读 TRNG_DATA 清零 0: 真随机数未生成完毕 1: 真随机数已生成完毕
2	SEIF	TRNG 种子错误中断标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 未发生种子错误 1: 已发送种子错误
31~4	-	保留

15.7.2 TRNG 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
TRNG 基地址: 0x4001_0400					
TRNG_CONFIG	0x10	读/写	TRNG 配置寄存器	0x0000_0000	0x0000_0000
TRNG_DATA	0x14	读/写	TRNG 数据寄存器	0x0000_0000	0x0000_0000
TRNG_STS	0x18	读/写	TRNG 标志寄存器	0x0000_0000	0x0000_0000

16 UART0~5

16.1 时钟源

- SC32L14T/14G 系列所有 UART 的时钟源仅一种，来自 PCLK

16.2 特性

- 六个 UART：UART0~5
- UART2 具有完整的 LIN 接口：
 - 主从模式可切换
 - 支持主机模式下硬件 break 发送（10/13bits）
 - 支持从机模式下硬件 break 检测（10/11bits）
 - 支持从机模式下波特率同步
 - 提供相关中断/状态位/标志位/容错范围
- UART0~5 的信号口均有两组映射可选
- 独立波特率发生器
- SM0~1 串行通信模式控制位可选四种通讯模式：
 - 模式 0，8 位半双工同步通信模式，在 RX 引脚上收发串行数据。TX 引脚用作发送移位时钟。每帧收发 8 位，低位先接收或发送；
 - 模式 1，10 位全双工异步通信，由 1 个起始位，8 个数据位和 1 个停止位组成，通信波特率可变；
 - 模式 2，保留
 - 模式 3，11 位全双工异步通信，由 1 个起始位，8 个数据位，一个可编程的第 9 位和 1 个停止位组成，通信波特率可变。
- 发送和接收完成可产生中断并置起对应的标志位 TXIF 和 RXIF，中断标志需要软件清除
- UART0~5 均支持数据匹配功能
- UART0 和 UART1 可产生 DMA 请求
- UART2~5 不能产生 DMA 请求
- UART0~5 均支持从 STOP Mode 唤醒：
 - START 位下降沿可唤醒 STOP Mode
 - 提供对应的唤醒中断使能位 WKIE 及唤醒中断标志位 WKIF

T_DIO、T_CLK 是 2 线 JTAG 烧写和仿真的信号线，用户在烧录时可通过 Customer Option 项配置这两端口的模式：JTAG 专用模式和常规模式（JTAG 专用口无效）。

注意：UART3 的信号引脚支持两种映射方案

- 映射 1：RXD3 / TXD3
- 映射 2：RXD3A / TXD3A

当选择映射 1（RXD3/TXD3）时，该组引脚与系统烧录/调试接口（T_CLK / T_DIO）复用。在此映射下，若启用全双工通信，T_CLK / T_DIO 可能会与 UART3 的 RXD3 接收时序产生冲突，导致通信异常。因此，选择映射 1 时必须将 UART3 配置为半双工通信模式，以避免此硬件冲突并保证通信稳定性。

如需使用全双工 UART 通信，请将管脚映射至映射 2（RXD3A/TXD3A）。

16.3 UART2-LIN

UART2 支持标准的 LIN 通信协议

16.3.1 LIN 帧结构

根据 LIN 协议，所有的传输信息被打包为帧。一个帧由一个报头（主机任务提供）和一个紧跟其后的应答（从机任务提供）组成。报头（主机任务提供）由一个 break 域和一个 sync（同步）域再跟一个帧识别码 (frame ID) 组成。帧 ID 仅作为定义帧的用途。从机任务负责回应相关的帧 ID。响应由一个数据域和一个校验域组成。下图是 LIN 帧的结构



16.3.2 LIN 主机模式

通过设置 FUNCSEL=1, SLVEN=0, UART 控制器支持 LIN 主机模式。在 LIN 模式，根据 LIN 的标准，每个字节由值为 0（显性）的 START 位开始，接着是 8 位数据位，没有校验位，LSB 优先，由一个值为 1（隐性）的 STOP 位结束。使能并初始化 LIN 主机模式需要如下步骤：

- ① 设置 UART_BAUD 寄存器设定波特率。
- ② 设置 FUNCSEL=1, 选择 LIN 功能模式
- ③ 设置 SM[1:0]=01 配置 UART 为模式 1

一个完整的报头由一个 break 域和同步域再跟一个帧标识符(帧 ID)组成。UART 控制器可以选择“break 域”作为发送的报头。“同步域”和“帧 ID 域”需要用户通过软件写入，即：发送一个完整的报头到总线，软件必须依次填同步数据(0x55)和帧 ID 数据到 UART_DAT 寄存器。

16.3.3 LIN 从机模式

通过设置 FUNCSEL=1, SLVEN=1, UART 控制器支持 LIN 从机模式。在 LIN 模式，根据 LIN 的标准，每一个字节域都是由一个值为 0 的显性位开始的，跟着 8 个数据位，没有校验位，最低位在前，由一个值为 1 的隐性 stop 位结束。

LIN 从机模式初始化的流程如下：

- ① 设置 UART_BAUD 寄存器设定波特率。
- ② 设置 FUNCSEL=1, 选择 LIN 功能模式。
- ③ 设定 SM[1:0]=1 配置 UART 为模式 1
- ④ 设定 SLVEN=1, 使能 LIN 从机模式

LIN 从模式下，通过设定 LBDL 使能从机 break 域检测功能来侦测接收“break 域”。接收到一个 break 后，

BKIF 标志将被置位。如果 BKIE =1 中断将发生。为了避免比特率偏差，控制器支持自动重同步功能，避免时钟误差错误，通过设定 SLVAREN 使能该特性。

16.3.3.1 同步域误差错误

自动重新同步模式下，控制器将检测同步域的误差错误。误差错误检测比较当前波特率和接收到的同步域的波特率。两个检测被同步执行。

检查 1:根据同步域的第一个下降沿和最后一个下降沿的测量值

- 如果误差大于 15%，报头错误标志 SLVHEIF 将被置位
- 如果误差在 14% 和 15%之间，报头错误标志 SLVHEIF 可能被置位也可能没有被置位 (取决于数据失相)

检查 2: 根据同步域的每一个下降沿的测量值。

- 如果误差大于 19%，报头错误标志 SLVHEIF 将被置位
- 如果误差在 15% 和 19%之间，报头错误标志 SLVHEIF 可能被置位也可能没有被置位(取决于数据失相)

注: 误差检测基于当前波特率时钟。因而，为了保证误差检测的正确性，建议用户在新的 break 域收到之前，通过软件将波特率重新加载为初始值。

16.4 UART 数据匹配功能

UART0 ~ UART5 均支持数据匹配功能。此功能在一主多从的应用场景下尤为有用：从机使能数据匹配后，仅当接收到的数据与预置匹配值一致时，才会置位接收标志位 RXIF 并在接收中断使能时触发接收中断。

16.4.1 配置说明

- 模式 1
 - 在 UART 工作于模式 1 时，需设置从机 UART 数据匹配寄存器 UARTn_DMR(n= 0 ~ 5) 中的 MATD[7:0]。
 - 当从机接收到的数据（8 位）与 MATD[7:0]的值完全匹配时，RXIF 将被置位；若此时接收中断使能位 RXIE = 1，则将触发接收中断。
- 模式 3
 - 在 UART 工作于模式 3 时，除需设置 MATD[7:0]外，还需设置数据匹配寄存器 UARTn_DMR(n= 0 ~ 5) 中的 MATD8。
 - 匹配条件扩展至接收数据的第九位。即，仅当接收到的 9 位数据（低 8 位 + 第九位）与 MATD[8:0] 的值完全匹配时，RXIF 才会被置位，并在 RXIE = 1 时触发中断。
- 功能使能与关闭
 - 完成数据匹配寄存器 (UARTn_DMR) 的配置后，用户需将 UART 控制寄存器 UARTn_CON(n= 0 ~ 5) 中的数据匹配使能位 ENDMR 置 1，以激活数据匹配功能。
 - 如需关闭数据匹配功能，需将 ENDMR 位清 0

16.5 UART 中断

对于 UARTn, n=0~5, 在发生“唤醒”或“数据收发完成”时都会生成中断。可以使用单独的中断使能位以提高灵活性。

中断事件	中断请求控制位	事件标志位	中断使能子开关
UART 从 STOP 模式唤醒	UARTn_IDE ->INTEN	WKIF	WKIE
数据发送完成		TXIF	TXIE
数据接收完成		RXIF	RXIE
断开中断	UART2_IDE ->INTEN	BKIF	BKIE
LIN 从机侦测到报头错误		SLVHEIF	SLVHEIE
波特率同步完成		SYNCIF	SYNCIE

16.6 UART0/1/3/4/5 寄存器

16.6.1 UART0/1/3/4/5 相关寄存器表

16.6.1.1 UART 控制寄存器 UARTn_CON

寄存器	读/写	说明	复位值	上电初始值
UARTn_CON (n=0/1/3/4/5)	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	SPOS	-	ENDMR	-	-	-
7	6	5	4	3	2	1	0
TXEN	RXEN	-	PRESCALER	-	SM2	SM1	SM0

位编号	位符号	说明												
13	SPOS	<ul style="list-style-type: none"> ● UART0 信号口映射控制位@UART0_CON <table border="1"> <thead> <tr> <th>信号</th> <th>RX0</th> <th>TX0</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS[1:0]=0</td> <td>PC10</td> <td>PC9</td> </tr> <tr> <td>SPOS[1:0]=1</td> <td>PA14</td> <td>PA15</td> </tr> </tbody> </table>	信号	RX0	TX0	SPOS 值			SPOS[1:0]=0	PC10	PC9	SPOS[1:0]=1	PA14	PA15
		信号	RX0	TX0										
		SPOS 值												
		SPOS[1:0]=0	PC10	PC9										
		SPOS[1:0]=1	PA14	PA15										
		<ul style="list-style-type: none"> ● UART1 信号口映射控制位@UART1_CON <table border="1"> <thead> <tr> <th>信号</th> <th>RX1</th> <th>TX1</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS[1:0]=0</td> <td>PA9</td> <td>PA8</td> </tr> <tr> <td>SPOS[1:0]=1</td> <td>PE3</td> <td>PE4</td> </tr> </tbody> </table>	信号	RX1	TX1	SPOS 值			SPOS[1:0]=0	PA9	PA8	SPOS[1:0]=1	PE3	PE4
		信号	RX1	TX1										
		SPOS 值												
		SPOS[1:0]=0	PA9	PA8										
		SPOS[1:0]=1	PE3	PE4										
		<ul style="list-style-type: none"> ● UART3 信号口映射控制位@UART3_CON <table border="1"> <thead> <tr> <th>信号</th> <th>RX3</th> <th>TX3</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS[1:0]=0</td> <td>PE6</td> <td>PE7</td> </tr> <tr> <td>SPOS[1:0]=1</td> <td>PC7</td> <td>PC6</td> </tr> </tbody> </table>	信号	RX3	TX3	SPOS 值			SPOS[1:0]=0	PE6	PE7	SPOS[1:0]=1	PC7	PC6
		信号	RX3	TX3										
SPOS 值														
SPOS[1:0]=0	PE6	PE7												
SPOS[1:0]=1	PC7	PC6												
<ul style="list-style-type: none"> ● UART4 信号口映射控制位@UART4_CON 														

位编号	位符号	说明																		
		<table border="1"> <thead> <tr> <th>SPOS 值 \ 信号</th> <th>RX4</th> <th>TX4</th> </tr> </thead> <tbody> <tr> <td>SPOS[1:0]=0</td> <td>PB3</td> <td>PB4</td> </tr> <tr> <td>SPOS[1:0]=1</td> <td>PD8</td> <td>PD7</td> </tr> </tbody> </table> <p>● UART5 信号口映射控制位@UART5_CON</p> <table border="1"> <thead> <tr> <th>SPOS 值 \ 信号</th> <th>RX5</th> <th>TX5</th> </tr> </thead> <tbody> <tr> <td>SPOS[1:0]=0</td> <td>PB9</td> <td>PB10</td> </tr> <tr> <td>SPOS[1:0]=1</td> <td>PB7</td> <td>PB8</td> </tr> </tbody> </table>	SPOS 值 \ 信号	RX4	TX4	SPOS[1:0]=0	PB3	PB4	SPOS[1:0]=1	PD8	PD7	SPOS 值 \ 信号	RX5	TX5	SPOS[1:0]=0	PB9	PB10	SPOS[1:0]=1	PB7	PB8
SPOS 值 \ 信号	RX4	TX4																		
SPOS[1:0]=0	PB3	PB4																		
SPOS[1:0]=1	PD8	PD7																		
SPOS 值 \ 信号	RX5	TX5																		
SPOS[1:0]=0	PB9	PB10																		
SPOS[1:0]=1	PB7	PB8																		
11	ENDMR	数据匹配功能使能位 0: 不使能数据匹配功能 1: 使能数据匹配功能																		
7	TXEN	UART 发送允许控制位 0: 不允许发送数据, TXD 信号不再影响所在管脚的状态。此时若用户程序限制发送功能, 仅用到了接收, 那么与 TX 脚复用的其它功能将不受影响 1: 允许发送数据, TXD 所在管脚切换为 TXD 信号口模式																		
6	RXEN	UART 接收允许控制位 0: 不允许接收数据 1: 允许接收数据																		
4	PRESCALER	波特率倍率设置位 该位在 UART 的不同模式下有不同定义: ● 当 SM0~1 = 01 (UART 模式 1) 或 SM0~1 = 11 (UART 模式 3): ■ 0: 串行端口在系统时钟的 1 分频下运行 ■ 1: 串行端口在系统时钟的 16 分频下运行 ● 当 SM0~1 = 00 (UART 模式 0) 波特率倍率设置位: ■ 0: 串行端口在系统时钟的 12 分频下运行 ■ 1: 串行端口在系统时钟的 4 分频下运行																		
2	SM2	RB8 置位中断允许位 该位仅在模式 3 有效 0: 每收到一个完整的数据帧就置位 RI 产生中断请求; 1: 收到一个完整的数据帧时, 只有当 RB8=1 时才会置位 RI 产生中断请求																		
1~0	SM[1:0]	UART 通信模式控制位 00: 模式 0, 8 位半双工同步通信模式, 在 RX 引脚上收发串行数据。TX 引脚用作发送移位时钟。每帧收发 8 位, 低位先接收或发送。在该模式下使能 RXEN 位, UART 将发出一个完整一帧的时钟, 并且将 RXIF 置 1; 01: 模式 1, 10 位全双工异步通信, 由 1 个起始位, 8 个数据位和 1 个停止位组成, 通信波特率可变; 10: 保留 11: 模式 3, 11 位全双工异步通信, 由 1 个起始位, 8 个数据位, 一个可编程的第 9 位和 1 个停止位组成, 通信波特率可变																		
31~14 12, 10~8 5 3	-	保留																		

16.6.1.2 UART 标志状态位寄存器 UARTn_STS

寄存器	读/写	说明	复位值	上电初始值
UARTn_CON (n=0/1/3/4/5)	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
4	WKIF	UART 唤醒标志位 UART 从 STOP 模式唤醒后, 该位由硬件置 1, 如果此时 WKIE = 1, 将产生中断。 该位由软件写 1 清 0。
1	TXIF	发送中断标志位 数据发送完成该位由硬件置 1, 如果此时 TXIE = 1, 将产生中断。 该位由软件写 1 清 0。 注意: 在 DMA 模式下, DMA 写入发送缓存后, 该位由 DMA 模块清 0, 此时用户无需通过软件清 0。
0	RXIF	接收中断标志位 数据接收完成该位由硬件置 1, 如果此时 RXIE = 1, 将产生中断。 该位由软件写 1 清 0。 注意: 在 DMA 模式下, DMA 读取接收缓存后, 该位由 DMA 模块清 0, 此时用户无需通过软件清 0。
31~5 3~2	-	保留

16.6.1.3 UART 波特率配置寄存器 UARTn_BAUD

寄存器	读/写	说明	复位值	上电初始值
UARTn_CON (n=0/1/3/4/5)	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	BAUD[15:0]	UART 波特率配置位 写入 BAUD[15:0]后, UART 的波特率将按照以下公式配置:

位编号	位符号	说明
		BaudRate=f _{UART} / BAUD[15:0] 其中 f _{UART} 为 UART 时钟源经预分频之后的最终频率，见 PRESCALER 位说明。 注意：BAUD[15:0] 必须大于 0x0010。
31~16	-	保留

16.6.1.4 UART 数据寄存器 UARTn_DATA

寄存器	读/写	说明	复位值	上电初始值
UARTn_CON (n=0/1/3/4/5)	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	SBUF8
7	6	5	4	3	2	1	0

SBUF[7:0]

位编号	位符号	说明
8	SBUF8	UART 的发送/接收数据的第 9 位 该位仅在模式 3 有效。
7~0	SBUF[7:0]	UART 数据缓存 读操作：将返回接收锁存器中的内容 写操作：SBUF 的数据将送至发送移位寄存器，并启动发送流程
31~9	-	保留

16.6.1.5 UART 的中断使能及 DMA 控制寄存器 UARTn_IDE

寄存器	读/写	说明	复位值	上电初始值
UARTn_CON (n=0/1/3/4/5)	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7	TXDMAEN	DMA 发送通道使能位 0：禁用 DMA 发送功能 1：使能 DMA 发送功能 该位使能后，TXIF 置起可触发 DMA 通道发送请求 注意： 1. UART0 和 UART1 可产生 DMA 请求 2. UART2/3/4/5 不能产生 DMA 请求
6	RXDMAEN	DMA 接收通道使能位

位编号	位符号	说明
		0: 禁用 DMA 接收功能 1: 使能 DMA 接收功能 该位使能后, RXIF 置起可触发 DMA 通道接收请求 注意: 1. UART0 和 UART1 可产生 DMA 请求 2. UART2/3/4/5 不能产生 DMA 请求
4	WKIE	UART 唤醒中断使能位 0: WKIF 置起时, 不允许产生中断 1: WKIF 置起时, 产生中断
2	TXIE	UART 发送中断使能位 0: TXIF 置起时, 不允许产生中断 1: TXIF 置起时, 产生中断
1	RXIE	UART 接收中断使能位 0: RXIF 置起时, 不允许产生中断 1: RXIF 置起时, 产生中断
0	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
31~8 5 3	-	保留

16.6.1.6 UART 数据匹配寄存器 UARTn_DMR

寄存器	读/写	说明	复位值	上电初始值
UARTn_DATA (n=0/1/3/4/5)	读/写	UART 数据匹配寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	MATD8
7	6	5	4	3	2	1	0
MATD[7:0]							

位编号	位符号	说明
8	MATD8	UART 预设数据匹配接收数据的第 9 位 该位仅在模式 3 有效, 在模式 3 中: 如果 MATD8 为 0, 则接收数据的第九位为 0 且接收数据与 MATD[7:0] 中数据匹配时会触发 RXIF 置起并触发接收中断 如果 MATD8 为 1, 则接收数据的第九位为 1 且接收数据与 MATD[7:0] 中数据匹配时会触发 RXIF 置起并触发接收中断
7~0	MATD[7:0]	UART 预设数据匹配接收数据 当接收到的数据与 MATD[7:0]中数据匹配时, RXIF 才会置起并触发接收中断。
31~9	-	保留

16.6.2 UART0/1/3/4/5 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
UART0 基地址: 0x4002_0020						
UART0_CON	0x00	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000	-
UART0_STS	0x04	读/写	UART 标志位寄存器	0x0000_0000	0x0000_0000	-
UART0_BAUD	0x08	读/写	UART 波特率寄存器	0x0000_0000	0x0000_0000	-
UART0_DATA	0x0C	读/写	UART 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
UART0_IDE	0x10	读/写	通信口 DMA 控制寄存器	0x0000_0000	0x0000_0000	-
UART0_DMR	0x14	读/写	UART 数据匹配寄存器	0x0000_0000	0x0000_0000	-

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
UART1 基地址: 0x4002_0080						
UART1_CON	0x00	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000	-
UART1_STS	0x04	读/写	UART 标志位寄存器	0x0000_0000	0x0000_0000	-
UART1_BAUD	0x08	读/写	UART 波特率寄存器	0x0000_0000	0x0000_0000	-
UART1_DATA	0x0C	读/写	UART 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
UART1_IDE	0x10	读/写	通信口 DMA 控制寄存器	0x0000_0000	0x0000_0000	-
UART1_DMR	0x14	读/写	UART 数据匹配寄存器	0x0000_0000	0x0000_0000	-

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
UART3 基地址: 0x4002_2020						
UART3_CON	0x00	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000	-
UART3_STS	0x04	读/写	UART 标志位寄存器	0x0000_0000	0x0000_0000	-
UART3_BAUD	0x08	读/写	UART 波特率寄存器	0x0000_0000	0x0000_0000	-
UART3_DATA	0x0C	读/写	UART 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
UART3_IDE	0x10	读/写	通信口 DMA 控制寄存器	0x0000_0000	0x0000_0000	-
UART3_DMR	0x14	读/写	UART 数据匹配寄存器	0x0000_0000	0x0000_0000	-

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
UART4 基地址: 0x4002_1080						

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
UART4_CON	0x00	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000	-
UART4_STS	0x04	读/写	UART 标志位寄存器	0x0000_0000	0x0000_0000	-
UART4_BAUD	0x08	读/写	UART 波特率寄存器	0x0000_0000	0x0000_0000	-
UART4_DATA	0x0C	读/写	UART 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
UART4_IDE	0x10	读/写	通信口 DMA 控制寄存器	0x0000_0000	0x0000_0000	-
UART4_DMR	0x14	读/写	UART 数据匹配寄存器	0x0000_0000	0x0000_0000	-

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
UART5 基地址: 0x4002_00A0						
UART5_CON	0x00	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000	-
UART5_STS	0x04	读/写	UART 标志位寄存器	0x0000_0000	0x0000_0000	-
UART5_BAUD	0x08	读/写	UART 波特率寄存器	0x0000_0000	0x0000_0000	-
UART5_DATA	0x0C	读/写	UART 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
UART5_IDE	0x10	读/写	通信口 DMA 控制寄存器	0x0000_0000	0x0000_0000	-
UART5_DMR	0x14	读/写	UART 数据匹配寄存器	0x0000_0000	0x0000_0000	-

16.7 UART2 寄存器

16.7.1 UART2 相关寄存器表

16.7.1.1 UART 控制寄存器 UARTn_CON

寄存器	读/写	说明	复位值	上电初始值
UARTn_CON (n=2)	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	BKSIZE	-	BKTR
23	22	21	20	19	18	17	16
-	-	-	-	-	LBDL	SLVAREN	SLVEN
15	14	13	12	11	10	9	8
-	-	SPOS	-	ENDMR	-	-	FUNCSEL
7	6	5	4	3	2	1	0
TXEN	RXEN	-	PRESCALER	-	SM2	SM1	SM0

位编号	位符号	说明
26	BKSIZE	间隔段生成位长度选择位 0: 选择用 10 位时间长度发送

位编号	位符号	说明												
		1: 选择用 13 位时间长度发送												
24	BKTR	LIN 模式 断开字符发送触发位 0: 不发送断开字符 1: 发送断开字符 注意: 1. 当 LIN 是从机模式 (SLVEN=1) 时, 此位无效 2. 当 break 域发送完之后, 此位自动清除												
18	LBDL	LIN 断路检测长度选择位 0: 选择 10 位中断检测 1: 选择 11 位中断检测 注意: 1. 检测长度需要配合主机发送间隔长度进行选择; 2. 当 LIN 是主机模式 (SLVEN=0) 时, 此位无效。												
17	SLVAREN	波特率自动重同步使能位 0: LIN 从机自动重同步模式禁止 1: LIN 从机自动重同步模式使能 说明: 当自动重新同步功能使能后, 每个 LIN break 域后面, 用 LIN 的工作时钟持续采样 5 个下降沿的时间, 测量的结果储存在内部波特率缓存寄存器中, UARTn_BAUD 寄存器的值将被自动更新 注意: 当 LIN 是主机模式 (SLVEN=0) 时, 此位无效。												
16	SLVEN	LIN 从模式使能位 0: LIN 从模式禁止 (LIN 主模式使能) 1: LIN 从模式使能 (LIN 主模式禁止) 说明: 1. LIN 主模式下, SLVAREN、LBDL 无效 2. LIN 从模式下, 可检测断路												
13	SPOS[1:0]	<p>● UART2 信号口映射控制位@UART2_CON</p> <table border="1"> <thead> <tr> <th>信号</th> <th>RX2</th> <th>TX2</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS[1:0]=0</td> <td>PC14</td> <td>PC13</td> </tr> <tr> <td>SPOS[1:0]=1</td> <td>PD15</td> <td>PD14</td> </tr> </tbody> </table>	信号	RX2	TX2	SPOS 值			SPOS[1:0]=0	PC14	PC13	SPOS[1:0]=1	PD15	PD14
信号	RX2	TX2												
SPOS 值														
SPOS[1:0]=0	PC14	PC13												
SPOS[1:0]=1	PD15	PD14												
11	ENDMR	数据匹配功能使能位 0: 不使能数据匹配功能 1: 使能数据匹配功能												
8	FUNCSEL	通讯模式选择位 0: UART 功能 1: LIN 功能使能, 此时 LIN 硬件模块和 UART 模块是同时开启, LIN 模块负责 break 检测/发送、波特率同步/更新 注意: FUNCSEL = 1 时, UART2_CON 的 bit 31~16 才有效												

位编号	位符号	说明
7	TXEN	UART 发送允许控制位 0: 不允许发送数据, TXD 信号不再影响所在管脚的状态。此时若用户程序限制发送功能, 仅用到了接收, 那么与 TX 脚复用的其它功能将不受影响 1: 允许发送数据, TXD 所在管脚切换为 TXD 信号口模式
6	RXEN	UART 接收允许控制位 0: 不允许接收数据 1: 允许接收数据
4	PRESCALER	波特率倍率设置位 该位在 UART 的不同模式下有不同定义: ● 当 SM0~1 = 01 (UART 模式 1) 或 SM0~1 = 11 (UART 模式 3): ■ 0: 串行端口在系统时钟的 1 分频下运行 ■ 1: 串行端口在系统时钟的 16 分频下运行 ● 当 SM0~1 = 00 (UART 模式 0) 波特率倍率设置位: ■ 0: 串行端口在系统时钟的 12 分频下运行 ■ 1: 串行端口在系统时钟的 4 分频下运行
2	SM2	RB8 置位中断允许位 该位仅在模式 3 有效 0: 每收到一个完整的数据帧就置位 RI 产生中断请求; 1: 收到一个完整的数据帧时, 只有当 RB8=1 时才会置位 RI 产生中断请求
1~0	SM[1:0]	UART 通信模式控制位 00: 模式 0, 8 位半双工同步通信模式, 在 RX 引脚上收发串行数据。TX 引脚用作发送移位时钟。每帧收发 8 位, 低位先接收或发送。在该模式下使能 RXEN 位, UART 将发出一个完整一帧的时钟, 并且将 RXIF 置 1; 01: 模式 1, 10 位全双工异步通信, 由 1 个起始位, 8 个数据位和 1 个停止位组成, 通信波特率可变; 10: 保留 11: 模式 3, 11 位全双工异步通信, 由 1 个起始位, 8 个数据位, 一个可编程的第 9 位和 1 个停止位组成, 通信波特率可变
31~27 25 23~19 15~14 12, 10~9 5 3	-	保留

16.7.1.2 UART 标志状态位寄存器 UARTn_STS

寄存器	读/写	说明	复位值	上电初始值
UARTn_STS (n=2)	读/写	UART 标志状态位寄存器	0x0000_0000	0x0000_0000

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

-	-	-	WKIF	-	-	TXIF	RXIF
---	---	---	------	---	---	------	------

位编号	位符号	说明
11	SYNCIF	LIN 模式下，波特率同步完成标志位 该位会在同步域 0x55 后置起
10	SLVSYNIF	LIN 从机同步域状态位 该位表示在自动重同步模式，LIN 同步域正在被分析。 正常通讯情况下，该位只读，作状态位使用 当接收器报头侦测到一些错误，用户可以对该位写 1 以复位内部电路来重新搜索新的帧报头 0: 当前字符不在 LIN 同步状态 1: 当前字符在 LIN 同步状态 注意： 1. 该位仅在 LIN 从机模式有效 2. 当向该位写 1，硬件会重载初始波特率并重新搜索新的帧报头
9	SLVHEIF	LIN 从机报头错误标志 在 LIN 从机模式，当侦测到一个 LIN 报头错误时，该位由硬件置 1，向该位写 1 清 0 0: LIN 未检测到报头错误 1: LIN 检测到报头错误 错误情况包括： 1. break 间隔符太短（小于 0.5 位的时间） 2. 在非自动重同步模式同步域数据不是 0x55 3. 自动重同步模式同步域偏离错误 注意：该位仅在 LIN 从机模式时有效。
8	BKIF	LIN 模式断开中断标志位 break 域发送或接收完毕后该位由硬件置 1，如果此时 BKIE = 1，将产生中断。 该位由软件写 1 清 0。
4	WKIF	UART 唤醒标志位 UART 从 STOP 模式唤醒后，该位由硬件置 1，如果此时 WKIE = 1，将产生中断。 该位由软件写 1 清 0。
1	TXIF	发送中断标志位 数据发送完成该位由硬件置 1，如果此时 TXIE = 1，将产生中断。 该位由软件写 1 清 0。 注意：在 DMA 模式下，DMA 写入发送缓存后，该位由 DMA 模块清 0，此时用户无需通过软件清 0。
0	RXIF	接收中断标志位 数据接收完成该位由硬件置 1，如果此时 RXIE = 1，将产生中断。 该位由软件写 1 清 0。 注意：在 DMA 模式下，DMA 读取接收缓存后，该位由 DMA 模块清 0，此时用户无需通过软件清 0。
31~5 3~2	-	保留

16.7.1.3 UART 波特率配置寄存器 UARTn_BAUD

寄存器	读/写	说明	复位值	上电初始值
UARTn_BAUD (n=2)	读/写	UART 波特率配置寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	BAUD[15:0]	UART 波特率配置位 写入 BAUD[15:0]后, UART 的波特率将按照以下公式配置: $BaudRate=f_{UART} / BAUD[15:0]$ 其中 f_{UART} 为 UART 时钟源经预分频之后的最终频率, 见 PRESCALER 位说明。 注意: BAUD[15:0]必须大于 0x0010。
31~16	-	保留

16.7.1.4 UART 数据寄存器 UARTn_DATA

寄存器	读/写	说明	复位值	上电初始值
UARTn_DATA (n=2)	读/写	UART 数据寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	SBUF8
7	6	5	4	3	2	1	0
SBUF[7:0]							

位编号	位符号	说明
8	SBUF8	UART 的发送/接收数据的第 9 位 该位仅在模式 3 有效。
7~0	SBUF[7:0]	UART 数据缓存 读操作: 将返回接收锁存器中的内容 写操作: SBUF 的数据将送至发送移位寄存器, 并启动发送流程
31~9	-	保留

16.7.1.5 UART 的中断使能及 DMA 控制寄存器 UARTn_IDE

寄存器	读/写	说明	复位值	上电初始值
UARTn_IDE (n=2)	读/写	UART 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	SYNCIE	-	SLVHEIE	BKIE
7	6	5	4	3	2	1	0
-	-	-	WKIE	-	TXIE	RXIE	INTEN

位编号	位符号	说明
11	SYNCIE	LIN 模式下，波特率同步完成中断使能位
9	SLVHEIE	LIN 从机报头错误中断使能，仅在 LIN 从机模式时有效 LIN 从机侦测到一个 LIN 报头错误时，SLVHEIE 位由硬件置 1 如果此时 SLVHEIE=1，将产生中断。 错误情况包括： 1. break 间隔符太短（小于 0.5 位的时间） 2. 在非自动重同步模式同步域数据不是 0x55 3. 自动重同步模式同步域偏离错误
8	BKIE	LIN 模式断开中断控制位 0: BKIF 置起时，不允许产生中断 1: BKIF 置起时，产生中断
4	WKIE	UART 唤醒中断使能位 0: WKIF 置起时，不允许产生中断 1: WKIF 置起时，产生中断
2	TXIE	UART 发送中断使能位 0: TXIF 置起时，不允许产生中断 1: TXIF 置起时，产生中断
1	RXIE	UART 接收中断使能位 0: RXIF 置起时，不允许产生中断 1: RXIF 置起时，产生中断
0	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
31~12 10 7~5 3	-	保留

16.7.1.6 UART 数据匹配寄存器 UARTn_DMR

寄存器	读/写	说明	复位值	上电初始值
UARTn_DATA (n=2)	读/写	UART 数据匹配寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	MATD8
7	6	5	4	3	2	1	0
MATD[7:0]							

位编号	位符号	说明
8	MATD8	UART 预设数据匹配接收数据的第 9 位 该位仅在模式 3 有效，在模式 3 中： 如果 MATD8 为 0，则接收数据的第九位为 0 且接收数据与 MATD[7:0] 中数据匹配时会触发 RXIF 置起并触发接收中断 如果 MATD8 为 1，则接收数据的第九位为 1 且接收数据与 MATD[7:0] 中数据匹配时会触发 RXIF 置起并触发接收中断
7~0	MATD[7:0]	UART 预设数据匹配接收数据 当接收到的数据与 MATD[7:0] 中数据匹配时，RXIF 才会置起并触发接收中断。
31~9	-	保留

16.7.2 UART2 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
UART2 基地址: 0x4002_1020						
UART2_CON	0x00	读/写	UART 控制寄存器	0x0000_0000	0x0000_0000	-
UART2_STS	0x04	读/写	UART 标志位寄存器	0x0000_0000	0x0000_0000	-
UART2_BAUD	0x08	读/写	UART 波特率寄存器	0x0000_0000	0x0000_0000	-
UART2_DATA	0x0C	读/写	UART 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
UART2_IDE	0x10	读/写	通信口 DMA 控制寄存器	0x0000_0000	0x0000_0000	-
UART2_DMR	0x14	读/写	UART 数据匹配寄存器	0x0000_0000	0x0000_0000	-

17 智能卡接口 (SC)

17.1 概述

SC32L14T/14G 系列内置的智能卡接口控制器(Smart Card controller)基于 ISO/IEC 7816-3 标准, 可通过两线进行八位数据的串行通信, 可软件控制 GPIO 引脚作为智能卡复位功能和智能卡插入检测功能。

17.2 时钟源

SC32L14T/14G 系列 SC 的时钟源仅一种: 来自 HCLK

17.3 特性

SC32L14T/14G 智能卡接口特点如下:

- 支持 ISO 7816-3 T = 0 异步半双工传输协议标准
- 可选择正/反向电平约定
- 可编程时钟源频率 f_{sc}
- 基本时间单元 (ETU) 灵活可调
- 可配置扩展保护时间
- 数据帧控制:
 - 支持可编程校验模式: 偶校验 (Even Parity) 或无校验 (No Parity)
 - 自动生成并检测校验位
 - 停止位长度 (1~2 位) 可配置
 - 错误指示信号 (Error Signal) 脉冲宽度可设
- 信号口有两组映射可选

17.4 工作模式

17.4.1 智能卡描述

智能卡接口控制器支持激活, 冷复位, 暖复位和释放序列。激活, 冷复位, 暖复位和释放序列如下描述。

17.4.1.1 SC 接口信号脚配置

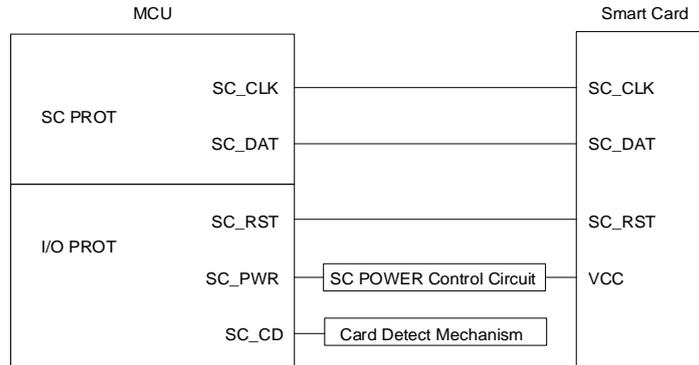
配置 MCU 的智能卡硬件接口:

- SCCLK0 (SC_CLK) 信号: SC 接口时钟引脚 (MCU 输出)
- SCIO0 (SC_DAT) 信号: SC 接口数据引脚(双向)

SC 接口的以下信号口可以通过 GPIO 自定义:

- SC_RST: 智能卡的复位脚, 由 MCU 的 GPIO 控制, 复位信号低有效
- SC_PWR: 智能卡的电源开关控制引脚, 接 MCU 的 GPIO, 此处以 SC_PWR 输出高代表智能卡 VCC 上电

- SC_CD: 智能卡检测引脚，接 MCU 的 GPIO，检测卡是否插入

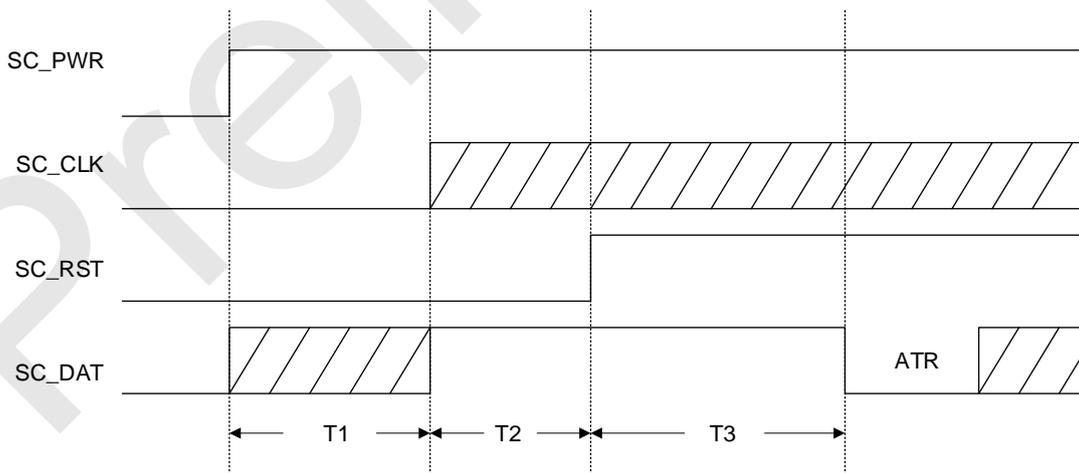


SC 接口连接示意图

17.4.1.2 激活和冷复位

激活和冷复位时序要求如下：

1. 设置 SC_RST 脚输出低电平
2. 智能卡电源开启（此处为设置 SC_PWR 输出高），进入 T1 周期，在 T1 周期结束前必须将 SC 接口设置为接收模式、SC_DAT 口拉至稳定的高电平
3. 设置 SC0_CON.CKEN=1 使能 SC_CLK 输出时钟，进入 T2 周期
4. 等待一段时间后，设置 SC_RST 脚输出高，结束 T2 周期，进入 T3 周期
5. 一段时间后，智能卡应答 ATR 信号，T3 周期结束，用户可以通过 SC0_DATA 寄存器获取 SC 接口读到的 ATR 数据



SC 激活和冷复位时序

周期	时序
T1	智能卡VCC上电至SC_CLK输出使能阶段
T2	SC_CLK信号输出至SC_RST复位信号上升沿输出前的阶段
T3	SC_RST上升沿至智能卡应答ATR信号阶段

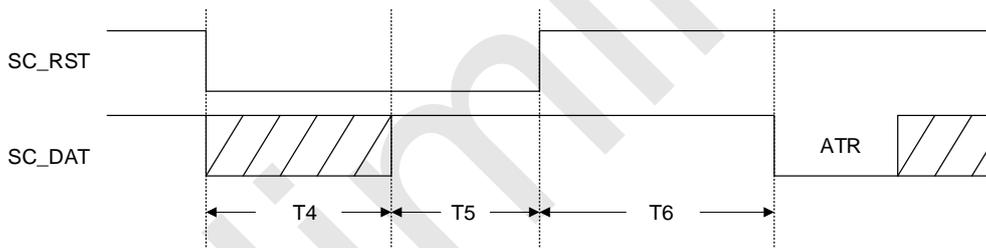
推荐时序关系（单位：SC CLOCK）

T1	T2	T3
83.5	491	400 ≤ T3 ≤ 40000
133	537	
165	569	
165	42060	

17.4.1.3 暖复位

暖复位时序要求如下：

1. SC_RST 拉低进入 T4 周期，在 T4 周期结束前必须将 SC_DAT 口拉至稳定的高电平
2. 进入 T5 周期，等待一段时间后设置 SC_RST 输出高电平，结束 T5 周期
3. 进入 T6 周期，等待智能卡应答
4. 一段时间后，智能卡应答 ATR 信号，T6 周期结束，用户可以通过 SC0_DATA 寄存器获取 SC 接口读到的 ATR 数据



SC 暖复位时序

周期	时序
T4	SC_RST复位信号使能至SC_DAT设置阶段
T5	SC_DAT稳定拉高至SC_RST上升沿输出阶段
T6	SC_RST复位信号结束至智能卡应答ATR信号阶段

推荐时序关系（单位：SC CLOCK）

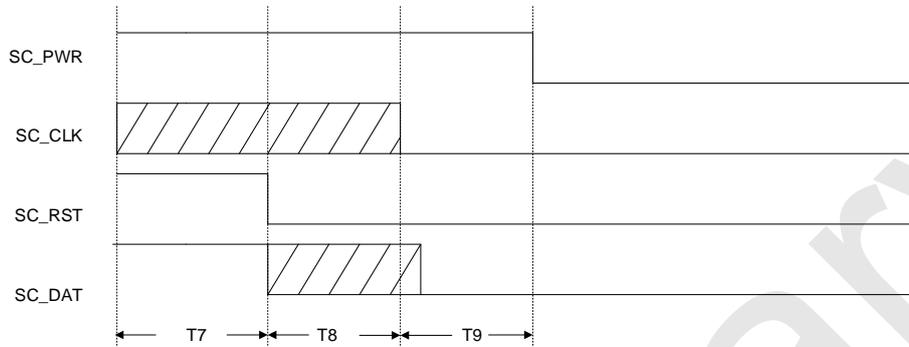
T4	T5	T6
81	483	400 ≤ T6 ≤ 40000
129	531	
161	563	
161	42106	

17.4.1.4 释放

通信结束后，无论结果是否正常，SC 接口必须按以下步骤释放智能卡触点：

1. T7 周期 SC_DAT 为空闲状态，当 SC_RST 拉低，复位信号输出，开始触点释放序列
2. 在 T8 周期结束前设置 SC0_CON.CKEN=0，停止 SC_CLK 信号输出

3. 进入 T9 周期，SC_CLK 信号输出停止后，将 SC_DAT 拉低
4. 等待一段时间后，关闭智能卡电源（此处为设置 SC_PWR 输出低）
5. 完成释放序列。



SC 释放序列

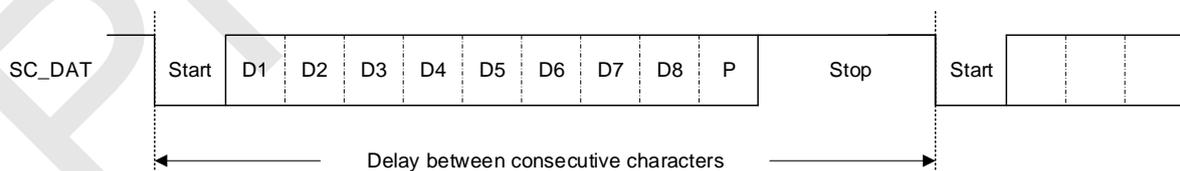
周期	时序
T7	SC_DAT 为空闲状态，智能卡检测复位信号
T8	SC_RST 复位信号使能至 SC_CLK 输出停止阶段
T9	SC_CLK 及 SC_DAT 输出停止至智能卡 VCC 关断阶段

推荐时序关系（单位：SC CLOCK）

T7	T8	T9
97	83	87
145	131	135
177	163	167

17.4.2 智能卡数据传输

智能卡接口的数据格式组成由十个连续的位如下：



SC 数据字符

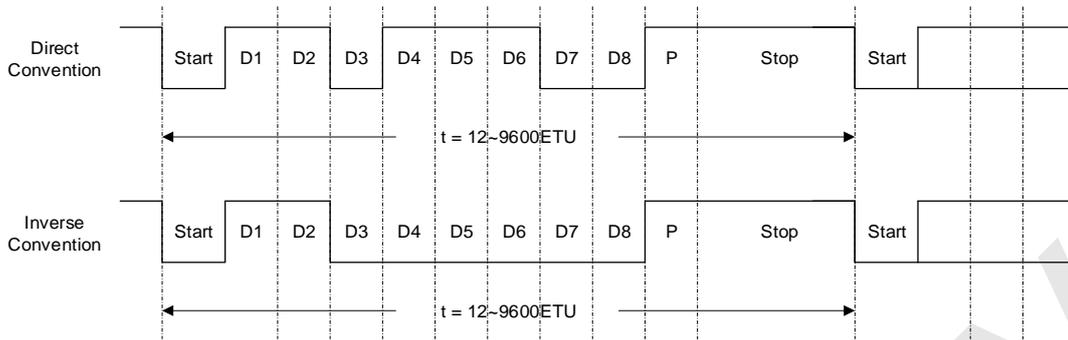
17.4.3 初始化字符 TS

依据 7816-3，智能卡 ATR 信息中的初始化字符 TS 有两种可能的模式：

如果 TS 模式是 1101_1100，则是正向约定。按正向约定解码，传送的字节等于 0x3B。

如果 TS 模式是 1100_0000，则是反向约定。按反向约定解码，传送的字节等于 0x3F。

用户可以设置 SC0_CON.CONF 为 0 或 1，在收到 ATR 的 TS 后去改变操作的约定。

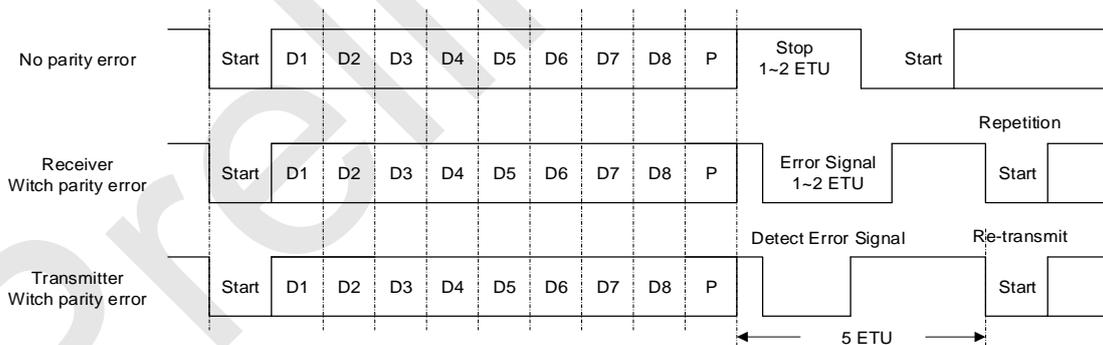


初始化字符

17.4.4 错误信号和字符重复

SC32L14T/14G 系列的 SC 接口符合 ISO7816-3 标准 T=0 模式下的错误信号和字符重复规则：

- 正常通信情况下，停止位 Stop 释放为高电平，SC 提供的停止位持续时间由 ERS[1:0] 设定；
- 接收模式下，用户可以通过设置 SC0_CON.TRER 来选择 SC 接收状态下出现校验位异常时的处理方式：
 - SC0_CON.TRER=0，SC 将产生一个传输错误中断给 CPU；
 - SC0_CON.TRER=1，接收到的数据校验错误时在停止位时间发送与 ERS[1:0] 所设定的 ETU 周期设置相符的低电平，即 Error signal，当前的发送方应当重传该字符。
- 发送模式下，在停止位时间检测到接收方将 SC_DAT 拉低，且持续时间与 ERS[1:0] 设定的 Error signal 一致，SC 将重传该字符，字符重传时不会加上扩展保护时间 T_{EGT}。从 Stop 位开始，到重传字符的 Start 开始，这段时间固定为 5 个 ETU，最多重传 3 次。

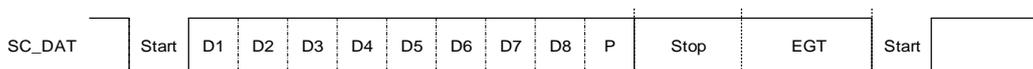


SC 错误信号

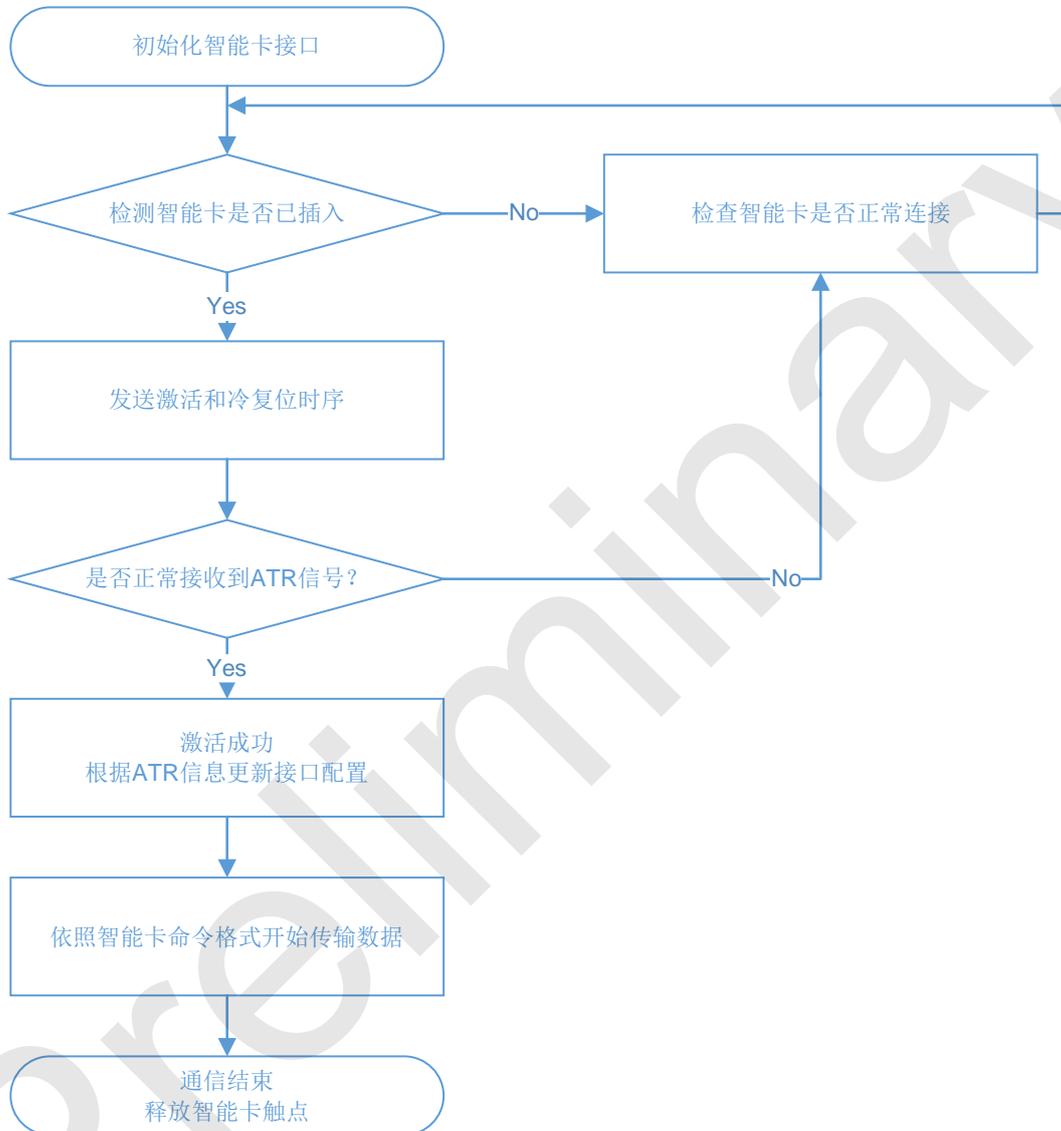
17.4.5 扩展保护时间

扩展保护时间 EGT 仅在 SC 接口发送状态下有效，用户可通过 EGT [7:0] 设置扩展保护时间。

SC 接口发送状态下，字符传输的 Start 之前，会有一段扩展保护时间 T_{EGT}，T_{EGT} 在字符重传时无效。



17.5 智能卡接口基本操作流程



17.6 智能卡接口操作步骤

17.6.1 SC 接口底层驱动配置初始化

这一阶段配置 MCU 的智能卡接口的初始化参数，软件配置的步骤如下：

1. 清除 SC0_STS 标志寄存器里的标志位，注意部分标志位为写“1”清零；
2. 设置 SC0_CON 控制寄存器，初始化 SC 接口、开启 SC 时钟，配置通信模式；
3. 配置中断优先级，使能 7816 智能卡中断；
4. 设置 ETU 率分频寄存器（默认初始值即可）

17.6.2 插卡检测

当智能卡被插入到一个阅读器，没有电源提供给任何触点。如果把电源提供给了错误的触点，卡上的芯片可能被严重损坏，这种情形很可能发生在触点已经上电的情况下去插智能卡。在可接受的触点机械容许条件下（针对于阅读器），在边沿探测器尚未决定智能卡是否正确分配前，触点保持无电状态。当阅读器发现智能卡已经正确地插入，电源就会提供给智能卡。

由于智能卡的检测与外部所接电路逻辑强相关，初学者前期可以先略过插卡检测阶段：在断电的情况下先将智能卡各信号脚稳定连接到 SC 接口，然后再接进入激活和冷复位阶段，这样可以快速的熟悉智能卡，但仍存在烧卡的风险，在实际生产及应用时插卡依然需要检测功能。

17.6.3 激活和冷复位

为了触发一个已经通过机械方式连接的卡片，接口设备需要按照 7816 智能卡标准时序激活电路，到激活前位置，卡已经准备好冷复位。在冷复位之前，卡的内部状态没有定义。冷复位成功后，智能卡会返回 ATR 信息，MCU 根据获取的 ATR 信息更新接口配置（市面上大部分的卡默认 ETU 为 372，无需更新接口配置），接下来就可以和智能卡通信。

17.6.4 数据传输

只有完成激活和冷复位之后才能进行数据收发，各种方式的收发详细描述如下：

17.6.4.1 发送（中断方式）

1. 将待发送数据写入 SC 数据寄存器 SC0_DATA;
2. 当数据发送成功或失败（超过重试设置），则会产生中断，则软件应首先清除中断标志，并读取中断标志寄存器确认中断清除成功；
3. 如继续发送，则重复步骤 2-3；
4. 数据发送过程结束后，硬件自动把 7816 智能卡模块设成接收模式，不需要软件设置

17.6.4.2 发送（轮询方式）

1. 将待发送数据写入 SC 数据寄存器 SC0_DATA;
2. 软件查询发送完成标志位 SC0_STS.TC，若数据传输成功则硬件会置起
3. 如继续发送，重复步骤 2-3；
4. 数据发送过程结束后，硬件自动把 7816 智能卡模块设成硬件接收模式，不需要软件设置；

17.6.4.3 接收（中断方式）

1. 当接收了一个字节数据成功或失败之后，会产生中断请求，若数据接收成功则硬件会置 SC0_STS.RC 标志；
2. 若接收成功，则软件应从 SC 数据寄存器 SC0_DATA 中读取收到的数据；若接收失败，则进行出错处理；
3. 继续接收重复步骤 2-3；

17.6.4.4 接收（轮询方式）

1. 软件查询 SC0_STS 标志寄存器，若数据接收成功则硬件会置 SC0_STS.RC 标志

2. 若接收成功，则软件应从 SC0_DATA 中读取收到的数据；若接收失败，则进行出错处理；
3. 继续接收重复步骤 2-3；

注意：

1. 以上数据传输过程中，若程序查询到状态标志置起后，需把相应标志清 0，并读取对应的状态标志寄存器确认清除成功；
2. 部分标志位为写“1”清零。

17.6.5 释放

当信息交换被完成或者被终止（例如卡没有应答，或检测过程中卡被移走），SC 接口设备将使电路按照 7816 标准释放规则停止工作，释放智能卡触点。

17.7 智能卡接口中断

智能卡接口在发送完成，接收完成以及发生各类通讯异常中断时均会置起对应标志位。可以使用单独的中断使能位以提高灵活性。

中断事件	中断请求控制位	事件标志位	中断使能子开关
数据发送完成	SC0_IDE->INTEN	TC	TXIE
数据接收完成		RC	RXIE
数据接收溢出		ROVF	ERRIE
接收帧错误		FER	
等待超时		WTER	
发送数据奇偶校验错误		RPER	
接收数据奇偶校验错误		TPER	

17.8 SC 寄存器

17.8.1 SC 相关寄存器表

17.8.1.1 SC 控制寄存器

寄存器	读/写	说明	复位值	上电初始值
SC0_CON	读/写	SC 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	SPOS	-	-	-	-	-
7	6	5	4	3	2	1	0
SCEN	CKEN	TREN	ERS[1:0]		CONS	TRER	PCS

位编号	位符号	说明
13	SPOS	● SC 信号口映射控制位@UART0_CON
		信号
		SCCLK0 SCIO0
		SPOS=0 PB9 PB10
		SPOS=1 PB7 PB8
7	SCEN	SC 接口设置 0: 通道作为普通 IO 1: 通道作为 7816 智能卡接口
6	CKEN	SC 时钟输出使能位 0: 禁止时钟输出 1: 打开时钟输出
5	TREN	SC 发送接收使能位 0: 接收使能, 发送禁止 1: 发送使能, 接收禁止, 发送完一帧数据之后接口会释放 SC_DAT, 开始检测停止位上的 Error Signal
4~3	ERS[1:0]	Stop 及 Error Signal 长度选择位 00: Stop 及 Error Signal 长度均为 2 个 ETU 01: Stop 及 Error Signal 长度均为 2 个 ETU 10: Stop 及 Error Signal 长度均为 1.5 个 ETU 11: Stop 及 Error Signal 长度均为 1 个 ETU
2	CONS	编码方式控制位 0: 正向约定, LSB 传输, 正逻辑电平 1: 反向约定, MSB 传输, 反逻辑电平
1	TRER	数据发送接收校验出错重发控制位 0: 校验出错, 直接置位中断标志位 1: 接收数据校验出错会发送一个低电平应答, 发送数据后接收到低电平应答会重发该数据
0	PCS	奇偶校验选择位 0: 不校验 1: 偶校验
31~15 12~8	-	保留

17.8.1.2 SC 标志寄存器

寄存器	读/写	说明	复位值	上电初始值
SC0_STS	读/写	SC 标志寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	WTRT	TBUSY	RBUSY
7	6	5	4	3	2	1	0
-	TPER	RPER	WTER	FER	ROVF	TC	RC

位编号	位符号	说明
10	WTRT	等待数据重发状态位 该位只读, 由硬件置起或清零 0: 未发生数据重发等待事件 1: 正在等待数据重发

位编号	位符号	说明
9	TBUSY	数据发送忙状态位 该位只读，由硬件置起或清零 0: 数据发送处于空闲状态 1: 数据正在发送，发送起始位时由硬件置 1，停止位结束、发送完成标志位置起时由硬件清零
8	RBUSY	数据接收忙状态位 该位只读，由硬件置起或清零 0: 数据接收处于空闲状态 1: 数据正在接收，收到起始位时由硬件置 1，收到停止位时由硬件清零
6	TPER	发送数据奇偶校验错误标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 发送数据校验正常 1: 发送数据校验错误
5	RPER	接收数据奇偶校验错误标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 接收数据校验正常 1: 接收数据校验错误
4	WTER	等待超时标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 未发生等待超时 1: 接收模式下连续两个字符之间超过等待时间或者复位应答时超过 40000 个时钟周期
3	FER	接收帧错误标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 帧格式无错误 1: 帧格式错误，未收到有效停止位
2	ROVF	接收溢出标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 无溢出错误 1: 接收数据缓冲器发生溢出
1	TC	发送完成标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 发送未完成 1: 发送已完成
0	RC	接收完成标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 接收未完成 1: 接收已完成
31~11,7	-	保留

17.8.1.3 SC 波特率配置寄存器

寄存器	读/写	说明	复位值	上电初始值
SC0_BAUD	读/写	SC 波特率配置寄存器	0x0000_0174	0x0000_0000

31	30	29	28	27	26	25	24
EGT[7:0]							
23	22	21	20	19	18	17	16
SCCK[4:0]							
15	14	13	12	11	10	9	8

-	-	-	-	ETUCK[11:8]			
7	6	5	4	3	2	1	0
ETUCK[7:0]							

位编号	位符号	说明
31~24	EGT[7:0]	SC 扩展保护时间设置 拓展保护时间为 EGT [7:0]个 ETU 时钟周期，即： SC 通信过程中的实际扩展保护时间 $T_{EGT} = T_{ETU} * EGT [7:0]$
20~16	SCCK[4:0]	SC 时钟周期设置： SC 时钟周期 $T_{SC} = (SCCK[4:0]+1)*2 / f_{sys}$
11~0	ETUCK[11:0]	ETU 时钟周期设置 ETU 时钟周期为(ETUCK [11:0] + 1)个 SC 时钟周期，即： ETU 时钟周期 $T_{ETU} = T_{SC} * (ETUCK [11:0] + 1)$ 注意：智能卡通信的频率范围在 1MHz~5MHz 之间，因此 ETUCK[11:0] 必须大于 0x004
23~21 15~12	-	保留

17.8.1.4 SC 数据寄存器

寄存器	读/写	说明	复位值	上电初始值
SC0_DATA	读/写	SC 数据寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7~0	SCD[7:0]	SC 数据缓存 读操作：将返回接收锁存器中的内容 写操作：SCD[7:0]的数据将送至发送移位寄存器，并启动发送流程
31~8	-	保留

17.8.1.5 SC 中断使能寄存器

寄存器	读/写	说明	复位值	上电初始值
SC0_IDE	读/写	SC 中断使能寄存器	0x0000_0000	0x0000_0000

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

-	-	-	-	ERRIE	TXIE	RXIE	INTEN
---	---	---	---	-------	------	------	-------

位编号	位符号	说明
3	ERRIE	通讯异常中断使能位 0: ERRIF 置起时, 不允许产生中断 1: ERRIF 置起时, 产生中断
2	TXIE	发送中断使能控制位 0: TC 置起时, 不允许产生中断 1: TC 置起时, 产生中断
1	RXIE	接收中断使能控制位 0: RC 置起时, 不允许产生中断 1: RC 置起时, 产生中断
0	INTEN	中断请求 CPU 的屏蔽位 0: 中断请求屏蔽 1: 中断请求开启
31~4	-	保留

17.8.2 SC 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
SC 基地址: 0x4002_10A0					
SC0_CON	0x00	读/写	SC 控制寄存器	0x0000_0000	0x0000_0000
SC0_STS	0x04	读/写	SC 标志寄存器	0x0000_0000	0x0000_0000
SC0_BAUD	0x08	读/写	SC 波特率配置寄存器	0x0000_0174	0x0000_0000
SC0_DATA	0x0C	读/写	SC 数据寄存器	0x0000_0000	0x0000_0000
SC0_IDE	0x10	读/写	SC 中断使能寄存器	0x0000_0000	0x0000_0000

18 SPI0/1

18.1 时钟源

- SC32L14T/14G 系列的 SPI 的时钟源仅一种，来自 PCLK

18.2 SPI0/1 特性

- 与 SPI0/1 共用寄存器地址和信号口，但功能完全独立
- 支持 13 档 SPI 时钟预分频，用户可设置到较低频率。
- 信号口共 4 组映射可选
- 支持接收缓存溢出中断及对应标志位，及时通知异常
- 支持主机/从机模式
- SPI0 和 SPI1 都支持 DMA

18.3 信号描述

主输出从输入(MOSI):

该路信号连接主设备和一个从设备。数据通过 MOSI 从主设备串行传送到从设备，主设备输出，从设备输入。

主输入从输出(MISO):

该路信号连接从设备和主设备。数据通过 MISO 从从设备串行传送到主设备，从设备输出，主设备输入。当 SPI 配置为从设备并未被选中，从设备的 MISO 引脚处于高阻状态。

SPI 串行时钟(SCK):

SCK 信号用作控制 MOSI 和 MISO 线上输入输出数据的同步移动。每 8 时钟周期线上传送一个字节。如果从设备未被选中，SCK 信号被此从设备忽略。

18.4 工作模式

SPI 可配置为主模式或从模式中的一种。SPI 模块的配置和初始化通过设置 SPI 控制寄存器 TWI_SPI0_CON/TWI_SPI1_CON 和 SPI 中断使能及 DMA 控制寄存器 TWI_SPI0_IDE/TWI_SPI1_IDE 来完成。配置完成后，通过设置 SPI 数据寄存器 TWI_SPI0_DATA/TWI_SPI1_DATA (以下简称 SPD)来完成数据传送。

在 SPI 通讯期间，数据同步地被串行的移进移出。串行时钟线(SCK)使两条串行数据线(MOSI 和 MISO)上数据的移动和采样保持同步。如果从设备没有被选中，则不能参与 SPI 总线上的活动。

当 SPI 主设备通过 MOSI 线传送数据到从设备时，从设备通过 MISO 线发送数据到主设备作为响应，这就实现了在同一时钟下数据发送和接收的同步全双工传输。发送移位寄存器和接收移位寄存器使用相同的特殊功能器地址，对 SPI 数据寄存器 SPD 进行写操作将写入发送移位寄存器，对 SPD 进行读操作将获得接收移位寄存器的数据。

有些设备的 SPI 接口会引出 SS 脚（从设备选择引脚，低有效），与 SC32L14T/14G 的 SPI 通信时，SPI 总

线上其它设备的 SS 脚的连接方式需根据不同的通信模式进行连接。下表列出了 SC32L14T/14G 的 SPI 不同通信模式下，SPI 总线上其它设备 SS 脚的连接方式：

SC32L14T/14G SPI	SPI 总线上其它设备	模式	从机的 SS（从设备选择引脚）
主模式	从模式	一主一从	拉低
		一主多从	SC32L14T/14G 引出多根 I/O，分别接至从机的 SS 脚。在数据传送之前，从设备的 SS 引脚必须被置低
从模式	主模式	一主一从	拉高

18.4.1 主模式

- 模式启动：

SPI 主设备控制 SPI 总线上所有数据传送的启动。当 SPI 控制寄存器 TWI_SPI0_CON/TWI_SPI1_CON 中的 SMSTR 位置 1 时，SPI 在主模式下运行，只有一个主设备可以启动传送。

- 发送：

在 SPI 主模式下，对 SPD 进行以下操作：在 8/16/位模式下将数据写到 SPD，数据将会写入发送移位寄存器。如果发送移位寄存器已经存在一个数据，那么主 SPI 产生一个 WCOL 信号以表明写入太快。但是在发送移位寄存器中的数据不会受到影响，发送也不会中断。另外如果发送移位寄存器不为空，那么主设备立即按照 SCK 上的 SPI 时钟频率串行地移出发送移位寄存器中的数据到 MOSI 线上。当传送完毕，SPI 标志状态位寄存器 TWI_SPI0_STS/TWI_SPI1_STS 中的 QTWIF 位被置 1。如果 SPI 中断被允许，当 QTWIF 位置 1 时，也会产生一个中断。

- 接收：

当主设备通过 MOSI 线传送数据给从设备时，相对应的从设备同时也通过 MISO 线将其发送移位寄存器的内容传送给主设备的接收移位寄存器，实现全双工操作。因此，QTWIF 标志位置 1 即表示传送完成也表示接收数据完毕。从设备接收的数据按照 MSB 或 LSB 优先的传送方向存入主设备的接收移位寄存器。当一个字节的数据完全被移入接收寄存器时，处理器可以通过读 SPD 获得该数据。

18.4.2 从模式

- 模式启动：

当 SPI 控制寄存器 TWI_SPI0_CON/TWI_SPI1_CON 中的 SMSTR 位清 0，SPI 在从模式下运行。

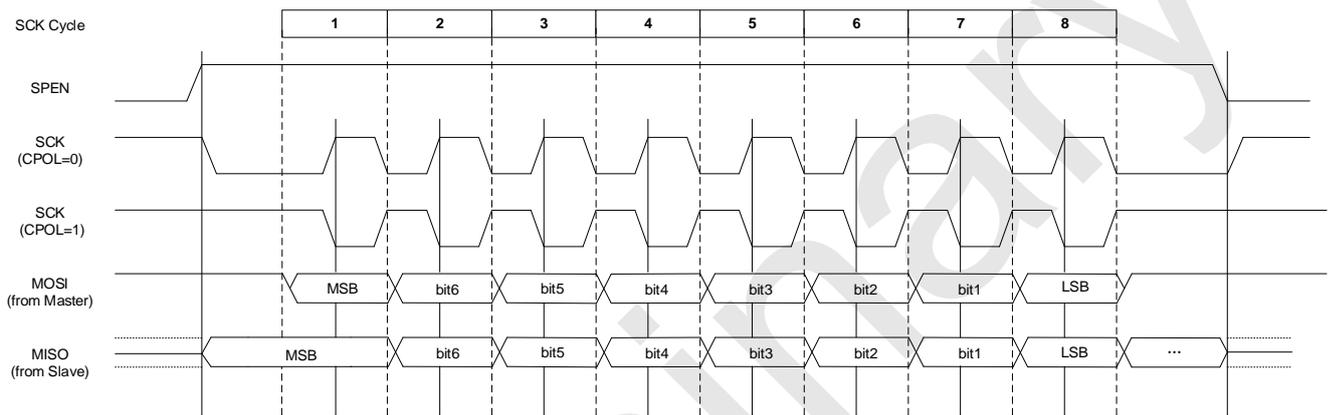
- 发送与接收：

从属模式下，按照主设备控制的 SCK 信号，数据通过 MOSI 引脚移入，MISO 引脚移出。一个位计数器记录 SCK 的边沿数，当接收移位寄存器移入 8 位数据(一个字节)同时发送移位寄存器移出 8 位数据(一个字节)，QTWIF 标志位被置 1。数据可以通过读取 SPD 寄存器获得。如果 SPI 中断被允许，当 QTWIF 置 1 时，也会产生一个中断。此时接收移位寄存器保持原有数据并且 QTWIF 位置 1，这样 SPI 从设备将不会接收任何数据直到 QTWIF 清 0。SPI 从设备必须在主设备开始一次新的数据传送之前将要传送的数据写入发送移位寄存器。如果在开始发送之前未写入数据，从设备将传送“0x00”字节给主设备。如果写 SPD 操作发生在传送过程中，那么 SPI 从设备的 WCOL 标志位置 1，即如果传送移位寄存器已经含有数据，SPI 从设备的 WCOL 位置 1，表示写 SPD 冲突。但是移位寄存器的数据不受影响，传送也不会被中断。

18.5 传送形式

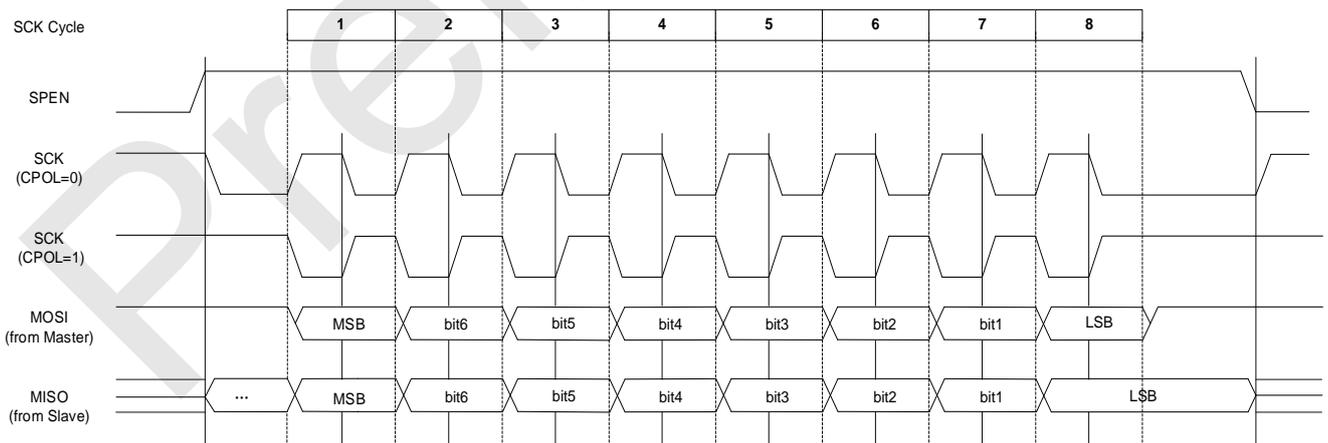
通过软件设置 SPI 控制寄存器 TWI_SPI0_CON/TWI_SPI1_CON 的 CPOL 位和 CPHA 位，用户可以选择 SPI 时钟极性和相位的四种组合方式。CPOL 位定义时钟的极性，即空闲时的电平状态，它对 SPI 传输格式影响不大。CPHA 位定义时钟的相位，即定义允许数据采样移位的时钟边沿。在主从通讯的两个设备中，时钟极性相位的设置应一致。

当 CPHA = 0，SCK 的第一个沿捕获数据，从设备必须在 SCK 的第一个沿之前将数据准备好。



CPHA = 0 数据传输图

当 CPHA = 1，主设备在 SCK 的第一个沿将数据输出到 MOSI 线上，从设备把 SCK 的第一个沿作为开始发送信号，SCK 的第二沿开始捕获数据，因此用户必须在第一个 SCK 的两个沿内完成写 SPD 的操作。这种数据传输形式是一个主设备一个从设备之间通信的首选形式。



CPHA = 1 数据传输图

18.6 出错检测

在发送数据序列期间写入 SPD 会引起写冲突，SPI 标志状态位寄存器 TWI_SPI0_STS/TWI_SPI1_STS 中的 WCOL 位置 1。WCOL 位置 1 不会引起中断，发送也不会中止。WCOL 位需由软件清 0。

18.7 SPI0/1 寄存器

18.7.1 SPI0/1 相关寄存器表

18.7.1.1 TWI_SPI 控制寄存器 TWI_SPIn_CON

寄存器	读/写	说明	复位值	上电初始值
TWI_SPIn_CON (n=0/1)	读/写	TWI_SPIn 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	DWIDTH	SSHIFT	-	-	-
23	22	21	20	19	18	17	16
-	MODE	-	CPOL	CPHA	DORD	-	SMSTR
15	14	13	12	11	10	9	8
-	SPOS[1:0]		-	QTWCK[3:0]			
7	6	5	4	3	2	1	0
QTWEN	-	START	STOP	-	-	AA	STRETCH

位编号	位符号	说明																																								
28	DWIDTH	全双工 SPI 模式接口传输宽度选择位 0: 8 位 1: 16 位																																								
27	SSHIFT	采样移位选择位 默认情况下, SPI 在 FLASH 驱动数据后过半个时钟周期开始采集数据。将该位置 1, 可考虑外部信号延迟, 推迟数据采集。 0: 不进行移位 1: 移位半个时钟周期																																								
22	MODE	TWI_SPI 模式选择位 0: TWI 模式 1: 全双工 SPI 模式																																								
20	CPOL	SPI 时钟极性控制位 0: SCK 在空闲状态下为低电平 1: SCK 在空闲状态下为高电平																																								
19	CPHA	SPI 时钟相位控制位 0: SCK 周期的第一沿采集数据 1: SCK 周期的第二沿采集数据																																								
18	DORD	SPI 传送方向选择位 0: MSB 优先发送 1: LSB 优先发送																																								
16	SMSTR	SPI 主从机模式选择位 0: SPI 为从设备 1: SPI 为主设备																																								
14~13	SPOS[1:0]	<ul style="list-style-type: none"> ● SPI0 信号口映射控制位@TWI_SPI0_CON <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>信号</th> <th>SCK0</th> <th>MOSI0</th> <th>MISO0</th> </tr> </thead> <tbody> <tr> <td>SPOS[1:0]=00</td> <td>PC8</td> <td>PC7</td> <td>PC6</td> </tr> <tr> <td>SPOS[1:0]=01</td> <td>PE0</td> <td>PD15</td> <td>PD14</td> </tr> <tr> <td>SPOS[1:0]=10</td> <td>PA1</td> <td>PA0</td> <td>PE13</td> </tr> <tr> <td>SPOS[1:0]=11</td> <td>PD9</td> <td>PD8</td> <td>PD7</td> </tr> </tbody> </table> ● SPI1 信号口映射控制位@TWI_SPI1_CON <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>信号</th> <th>SCK1</th> <th>MOSI1</th> <th>MISO1</th> </tr> </thead> <tbody> <tr> <td>SPOS[1:0]=00</td> <td>PC8</td> <td>PC7</td> <td>PC6</td> </tr> <tr> <td>SPOS[1:0]=01</td> <td>PE0</td> <td>PD15</td> <td>PD14</td> </tr> <tr> <td>SPOS[1:0]=10</td> <td>PA1</td> <td>PA0</td> <td>PE13</td> </tr> <tr> <td>SPOS[1:0]=11</td> <td>PD9</td> <td>PD8</td> <td>PD7</td> </tr> </tbody> </table> 	信号	SCK0	MOSI0	MISO0	SPOS[1:0]=00	PC8	PC7	PC6	SPOS[1:0]=01	PE0	PD15	PD14	SPOS[1:0]=10	PA1	PA0	PE13	SPOS[1:0]=11	PD9	PD8	PD7	信号	SCK1	MOSI1	MISO1	SPOS[1:0]=00	PC8	PC7	PC6	SPOS[1:0]=01	PE0	PD15	PD14	SPOS[1:0]=10	PA1	PA0	PE13	SPOS[1:0]=11	PD9	PD8	PD7
信号	SCK0	MOSI0	MISO0																																							
SPOS[1:0]=00	PC8	PC7	PC6																																							
SPOS[1:0]=01	PE0	PD15	PD14																																							
SPOS[1:0]=10	PA1	PA0	PE13																																							
SPOS[1:0]=11	PD9	PD8	PD7																																							
信号	SCK1	MOSI1	MISO1																																							
SPOS[1:0]=00	PC8	PC7	PC6																																							
SPOS[1:0]=01	PE0	PD15	PD14																																							
SPOS[1:0]=10	PA1	PA0	PE13																																							
SPOS[1:0]=11	PD9	PD8	PD7																																							

位编号	位符号	说明																				
		<table border="1"> <tr> <td>SPOS 值</td> <td></td> <td></td> <td></td> </tr> <tr> <td>SPOS[1:0]=00</td> <td>PC15</td> <td>PC14</td> <td>PC13</td> </tr> <tr> <td>SPOS[1:0]=01</td> <td>PE2</td> <td>PE3</td> <td>PE4</td> </tr> <tr> <td>SPOS[1:0]=10</td> <td>PA10</td> <td>PA11</td> <td>PA12</td> </tr> <tr> <td>SPOS[1:0]=11</td> <td>PB13</td> <td>PB12</td> <td>PB11</td> </tr> </table>	SPOS 值				SPOS[1:0]=00	PC15	PC14	PC13	SPOS[1:0]=01	PE2	PE3	PE4	SPOS[1:0]=10	PA10	PA11	PA12	SPOS[1:0]=11	PB13	PB12	PB11
SPOS 值																						
SPOS[1:0]=00	PC15	PC14	PC13																			
SPOS[1:0]=01	PE2	PE3	PE4																			
SPOS[1:0]=10	PA10	PA11	PA12																			
SPOS[1:0]=11	PB13	PB12	PB11																			
11~8	QTWCK[3:0]	<p>SPI 时钟预分频控制位</p> <p>0000: f_{PCLK}</p> <p>0001: f_{PCLK} /2</p> <p>0010: f_{PCLK} /4</p> <p>0011: f_{PCLK} /8</p> <p>0100: f_{PCLK} /16</p> <p>0101: f_{PCLK} /32</p> <p>0110: f_{PCLK} /64</p> <p>0111: f_{PCLK} /128</p> <p>1000: f_{PCLK} /256</p> <p>1001: f_{PCLK} /512</p> <p>1010: f_{PCLK} /1024</p> <p>1011: f_{PCLK} /2048</p> <p>1100: f_{PCLK} /4096</p> <p>其它: f_{PCLK} /4096</p> <p>注意: 为确保 SC32L14T/14G 系列 SPI0/1 正确通信, 通信频率请选择 6MHz 以下</p>																				
7	QWEN	<p>TWI_QSPI 模块使能控制位</p> <p>0: 关闭模块</p> <p>1: 打开模块</p>																				
31~29 27~24 21, 17, 15, 12, 6 3~2	-	保留																				

18.7.1.2 TWI_SPI 标志状态位寄存器 TWI_SPIn_STS

寄存器	读/写	说明	复位值	上电初始值
TWI_SPIn_STS (n=0/1)	读/写	TWI_SPIn 标志状态位寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
NBYTES[7:0]							
15	14	13	12	11	10	9	8
-	TMSTR	GCA	TXnE/RXnE	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	-	-	-	-	TXEIF	-	QTWIF

位编号	位符号	说明
7	WCOL	<p>写入冲突标志位</p> <p>该位由硬件置 1, 通过软件写 1 清 0, 用于指示是否发生写入冲突:</p> <p>0: 未检测到写入冲突</p> <p>1: 检测到一个写入冲突</p>
2	TXEIF	发送 FIFO 为空标志位

位编号	位符号	说明
		用于指示当前发送 FIFO 是否为空： 0：发送 FIFO 非空 1：发送 FIFO 为空 该位由硬件置 1，通过软件写 1 清 0。 注意：在 DMA 模式下，DMA 写入发送缓存后，该位由 DMA 模块清 0，此时用户无需通过软件清 0。
0	QTWIF	SPI 数据传送标志位 该位由硬件置 1，通过软件写 1 清 0，用于指示当前一帧 SPI 数据传输是否完成： 0：数据传输未完成/进行中 1：数据传输已完成
31~24 15, 11 6~3, 1	-	保留

18.7.1.3 TWI_SPI 数据寄存器 TWI_SPI_n_DATA

寄存器	读/写	说明	复位值	上电初始值
TWI_SPI _n _DATA (n=0/1)	读/写	TWI_SPI _n 数据寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	QTWIDAT[15:0]	SPI 数据缓存 读操作：从 SPI 接收 FIFO 读取接收到的数据 写操作：向 SPI 发送 FIFO 写入待发送的数据
31~16	-	保留

18.7.1.4 TWI_SPI 的中断使能及 DMA 控制寄存器 TWI_SPI_n_IDE

寄存器	读/写	说明	复位值	上电初始值
TWI_SPI _n _IDE (n=0/1)	读/写	TWI_SPI _n 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7	TXDMAEN	DMA 发送通道使能位 0: 禁用 DMA 发送功能 1: 使能 DMA 发送功能 该位使能后, TXnE 置起可触发 DMA 通道发送请求
6	RXDMAEN	DMA 接收通道使能位 0: 禁用 DMA 接收功能 1: 使能 DMA 接收功能 该位使能后, RXnE 置起可触发 DMA 通道接收请求
2	TBIE	发送 FIFO 为空时的中断使能位 0: TXEIF 置起时, 不允许产生中断 1: TXEIF 置起时, 产生中断
0	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
31~8 5~3, 1	-	保留

18.7.2 SPI0/1 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
TWI_SPI0 基地址: 0x4002_0060						
TWI_SPI0_CON	0x00	读/写	TWI_SPI0 控制寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI0_STS	0x04	读/写	TWI_SPI0 标志状态位寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI0_DATA	0x0C	读/写	TWI_SPI0 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
TWI_SPI0_IDE	0x10	读/写	TWI_SPI0 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000	-

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
TWI_SPI1 基地址: 0x4002_1060						
TWI_SPI1_CON	0x00	读/写	TWI_SPI1 控制寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI1_STS	0x04	读/写	TWI_SPI1 标志状态位寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI1_DATA	0x0C	读/写	TWI_SPI1 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
TWI_SPI1_IDE	0x10	读/写	TWI_SPI1 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000	-

19 TWI0/1

19.1 时钟源

- SC32L14T/14G 系列的 TWI 的时钟源仅一种，来自 PCLK

19.2 TWI0/1 特性

- 与 SPI0/1 共用寄存器地址和信号口，但功能完全独立
- 支持 11 档 TWI 时钟预分频，主机模式下 TWI 通信速率，默认为最小分频档位 ($f_{PCLK}/4$)
- 信号口共 4 组映射可选
- 可配置为主机模式或从机模式
- 主从机之间双向数据传输
- 可选的时钟延展
- TWI0 和 TWI1 都支持 DMA

19.3 TWI 信号描述

在 TWI 总线上，数据通过时钟线 SCL 和数据线 SDA 在主从机间逐一字节同步传送。每个字节数据长度是 8 位，一个 SCL 时钟脉冲传输一个数据位，数据由最高位 MSB 开始传输，每个字节传输后跟随一个应答位，每个位在 SCL 为高时采样。因此，SDA 线可能在 SCL 为低时改变，但在 SCL 为高时必须保持稳定。当 SCL 为高时，SDA 线上的跳变视为一个命令 (START 或 STOP)。

- **TWI 时钟信号线 (SCL)**

该时钟信号由主机发出，连接到所有的从机。每 9 个时钟周期传送一个字节数据。前 8 个周期作数据的传送，最后一个时钟作为接收方应答时钟。空闲时应为高电平，由 SCL 线上的上拉电阻拉高。

- **TWI 数据信号线 (SDA)**

SDA 是双向信号线，空闲时应为高电平，由 SDA 线上的上拉电阻拉高。

19.4 从机工作模式

- 模式启动：

当 TWI 使能标志位打开 (QTWEN=1)，同时接收到主机发送的启动信号时，模式启动。

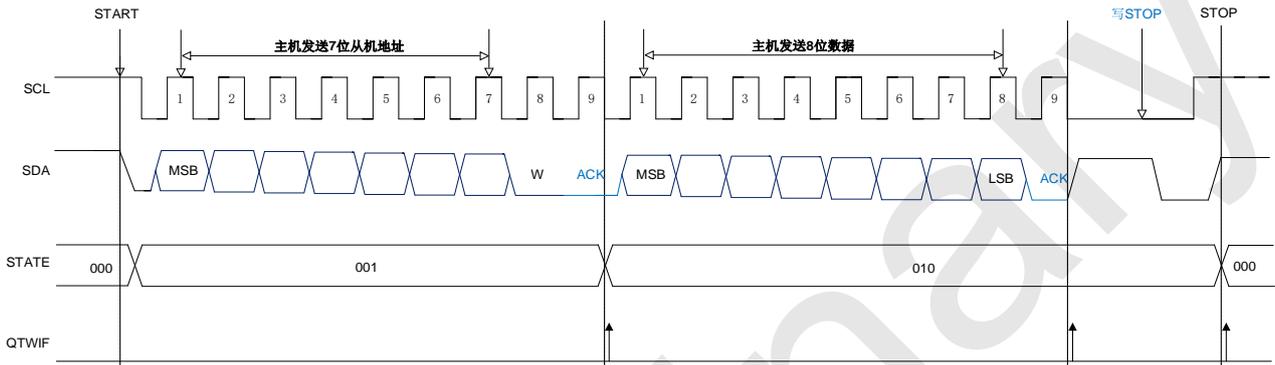
从机从空闲模式 (STATE[2:0] = 000) 进入接收第一帧地址 (STATE[2:0] = 001) 状态，等待主机的第一帧数据。第一帧数据由主机发送，包括了 7 位地址位和 1 位读写位，TWI 总线上所有从机都会收到主机的第一帧数据。主机发送完第一帧数据后释放 SDA 信号线。若主机所发地址与某一从机自身地址寄存器中的值相同，说明该从机被选中，被选中的从机会判断接总线上的第 8 位，即数据读写位 (=1，读命令；=0，写命令)，然后占用 SDA 信号线，在 SCL 的第 9 个时钟周期给主机一个低电平的应答信号，之后会释放总线。从机被选中后，会根据读写位的不同而进入不同的状态：

- 非通用地址响应，从机接收模式：

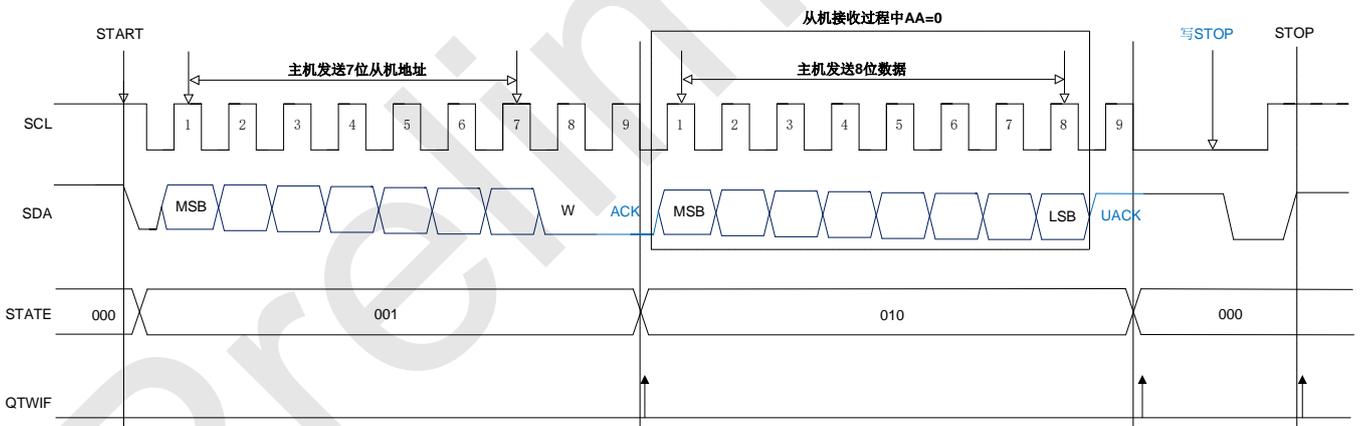
如果第一帧接收到的读写位是写（0），则从机进入到从机接收状态（STATE[2:0] = 010）等待接收主机发送的数据。主机每发送 8 位，都要释放总线，等待第 9 个周期从机的应答信号。

如果从机的应答信号是低电平，主机的通信可以有以下三种方式：

- 继续发送数据；
- 重新发送启动信号（start），此时从机重新进入接收第一帧地址（STATE[2:0] = 001）状态；
- 发送停止信号，表示本次传输结束，从机回到空闲状态，等待主机下一次的启动信号。



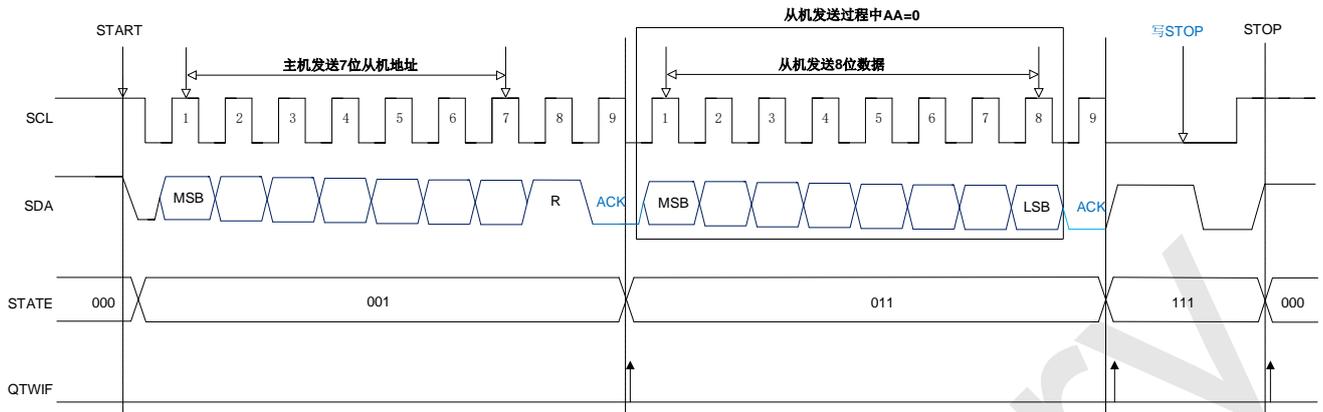
如果从机应答的是高电平（在接收过程中，从机寄存器中的 AA 值改写为 0），表示当前字节传输完以后，从机会主动结束本次传输，回到空闲状态（STATE[2:0] = 000），不再接收主机发送的数据。



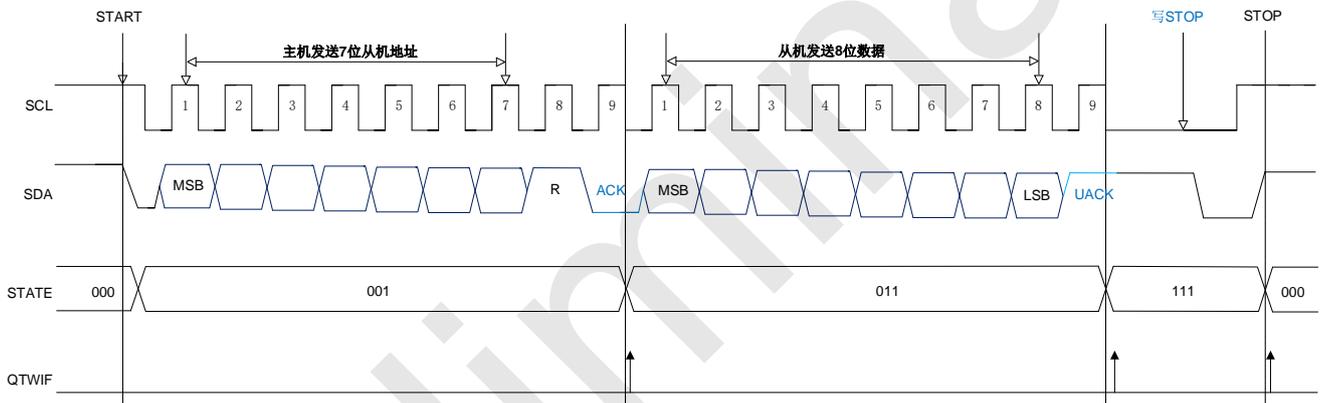
● 非通用地址响应，从机发送模式：

如果第一帧接收到的读写位是读（1），则从机会占用总线，向主机发送数据。每发送 8 位数据，从机释放总线，等待主机的应答：

如果主机应答的是低电平，则从机继续发送数据。在发送过程中，如果从机寄存器中的 AA 值被改写为 0，则传输完当前字节从机会主动结束传输并释放总线，等待主机的停止信号或重新启动信号（STATE[2:0] = 101）。



如果主机应答的是高电平，则从机 STATE[2:0] = 100，等待主机的停止信号或重新启动信号。

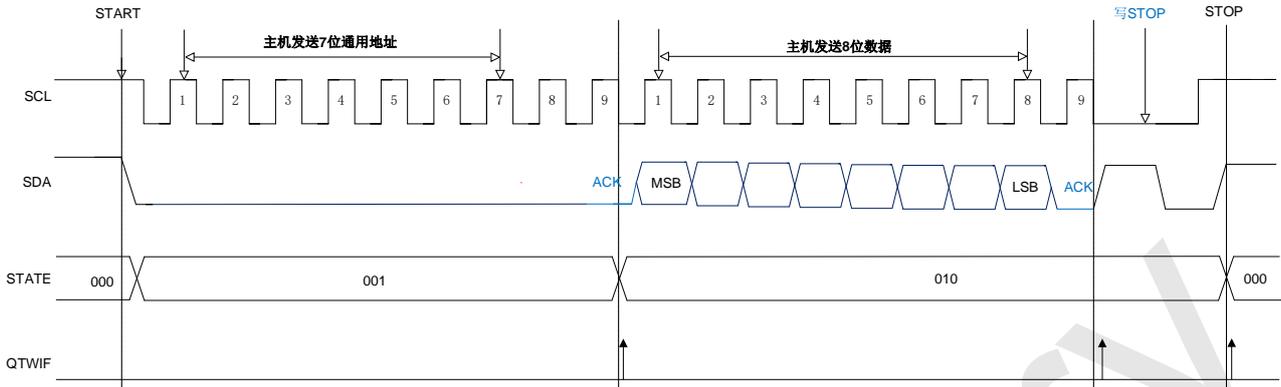


● 通用地址的响应:

GC=1 时，此时通用地址允许使用。从机进入到接收第一帧地址 (STATE[2:0] = 001) 状态，接收的第一帧数据中的地址位数据为 0x00，此时所有从机响应主机。主机发送的读写位是必须是写 (0)，所有从机接收后进入接收数据 (STATE[2:0] = 010) 状态。主机每发送 8 个数据释放一次 SDA 线，并读取 SDA 线上的状态:

如果有从机应答，则主机的通信可以有以下三种方式:

- 继续发送数据;
- 重新启动;
- 发送停止信号，结束本次通讯。



如果无从机应答，则 SDA 为空闲状态。

注意：在一主多从模式下使用通用地址时，主机发送的读写位不能为读（1）状态，否则除发送数据的设备，总线上其它设备均会响应。

19.5 从机模式操作步骤

- ① 配置 TWI0/1 控制寄存器 TWI_SPI0_CON/TWI_SPI1_CON.QTWEN = 1
- ② 配置 TWI0/1 控制寄存器 TWI_SPI0_CON/TWI_SPI1_CON
- ③ 配置 TWI0/1 地址寄存器 TWI_SPI0_ADD/TWI_SPI1_ADD
- ④ 如果从机接收数据，则等待 TWI_SPI0_STS/TWI_SPI1_STS 中的中断标志位 QTWIF 置 1。从机每接收到 8 位数据，QTWIF 会被置 1。中断标志位 QTWIF 需手动清零；
- ⑤ 如果从机发送数据，则要将待发送的数据写进 TWI_SPI0_DATA/TWI_SPI1_DATA 的低 8 位中，TWI 会自动将数据发送出去。每发送 8 位，中断标志位 QTWIF 就会被置 1。

19.6 主机工作模式

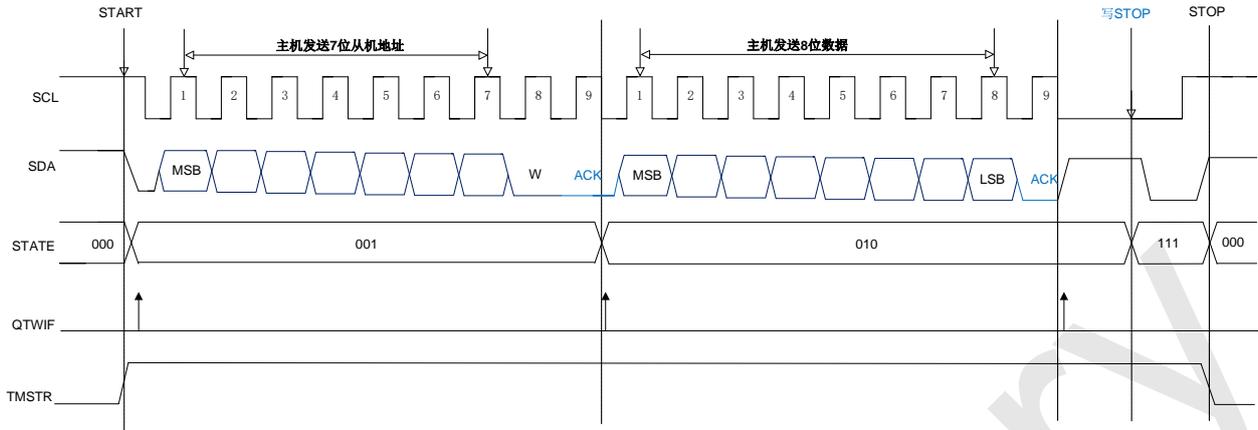
● 模式启动

当 TWI 接口向总线发出起始条件后，会自动切换为主模式，同时硬件将 TWI_SPI0_STS/TWI_SPI1_STS 的 TMSTR 位置 1。主机状态位 STATE[2:0] 从 000 切换到 001，同时中断条件 QTWIF 被置 1。

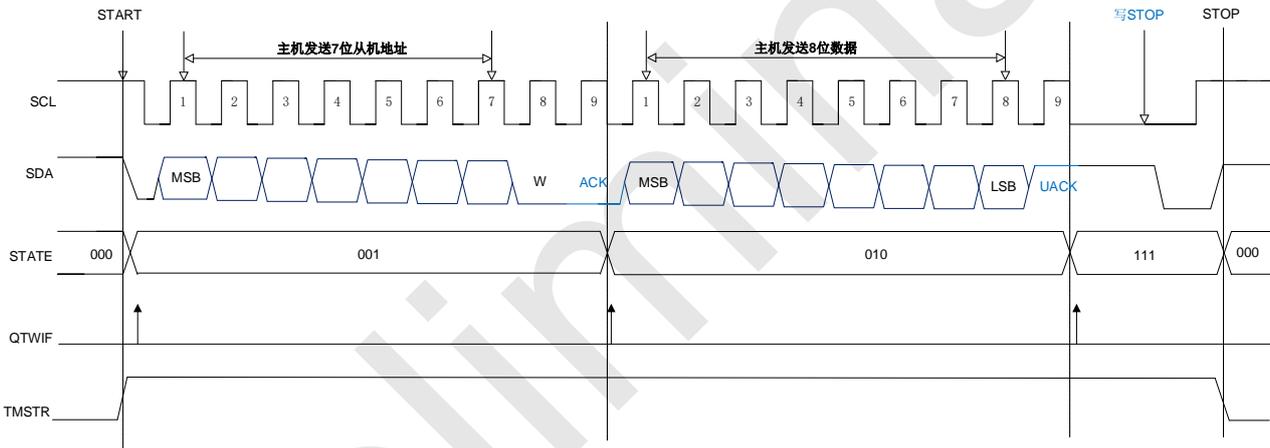
● TWI 主机发送模式

主机发送模式下，主机发送的第一帧数据包括了 7 位地址位（被选中的从机地址）和 1 位读写位（=0，写命令），TWI 总线上所有从机都会收到主机的第一帧数据。主机发送完第一帧数据后释放 SDA 信号线。被选中的从机在 SCL 的第 9 个时钟周期给主机一个应答信号，之后会释放总线并进入到从机接收状态等待接收主机发送的数据。主机每发送 8 位，都要释放总线，等待第 9 个周期从机的应答信号。

如果从机应答低电平，主机可以继续发送数据。也可以重新发送启动信号：



如果从机应答高电平，表示当前字节传输完以后，从机会主动结束本次传输，不再接收主机发送的数据，主机 STATE[2:0]从发送数据状态 010 切换为 100:

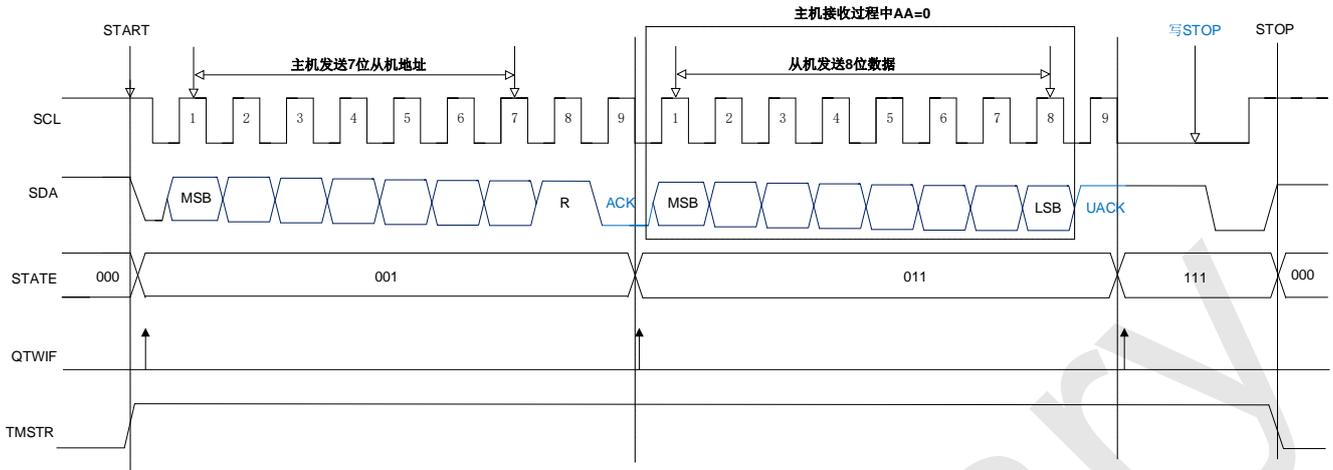


● TWI 主机接收模式

主机发送模式下，主机发送的第一帧数据包括了包括了 7 位地址位（被选中的从机地址）和 1 位读写位（=1，读命令），TWI 总线上所有从机都会收到主机的第一帧数据。主机发送完第一帧数据后释放 SDA 信号线。被选中的从机在 SCL 的第 9 个时钟周期给主机一个应答信号，之后会占用总线，向主机发送数据。每发送 8 位数据，从机释放总线，等待主机的应答。主机接收到从机地址匹配成功后的应答信号 ACK，并开始接收从机数据（STATE=011）：

1. 若主机应答位使能（AA=1），则每接收到一 byte 数据，主机回复应答信号 ACK，QTWIF 被置位；
2. 在接收最后一 byte 数据前，若应答使能位关闭（AA=0），则主机接收完最后一 byte 数据后回复 UACK，然后主机可发送停止信号。

主机接收模式下，主动释放总线方式如下：



19.7 主机模式操作步骤

- ① 配置 TWI0/TWI1 控制寄存器 TWI_SPI0_CON/TWI_SPI1_CON.QTWEN=1，使能 TWI
 - ② 配置 TWI0/1 控制寄存器 TWI_SP0_CON/TWI_SPI1_CON：配置 TWI0/1 通信速率（QTWCK[3:0]），将起始位 START 置“1”
 - ③ 配置 TWI0/1 数据寄存器 TWI_SPI0_DATA/TWI_SPI1_DATA：将“从机地址+读写位”写入 QTWIDAT，总线上发出地址帧
 - ④ 如果主机接收数据，则等待 TWI_SPI0_STS/TWI_SPI1_STS 中的中断标志位 QTWIF 置 1。主机每接收到 8 位数据，中断标志位 QTWIF 会被置 1。中断标志位 QTWIF 需手动清零；
 - ⑤ 如果主机发送数据，则要将待发送的数据写进 QTWIDAT 中，TWI 会自动将数据发送出去。每发送 8 位，中断标志位 QTWIF 就会被置 1。
 - ⑥ 数据接收发送完成，主机可发送停止条件（STOP=1），主机状态切换为 000。或发送重复起始信号，开始新一轮的数据传输。
- 注意：主机产生 stop 之后主机的 QTWIF 不会置位！**

19.8 TWI0/1 中断

对于 TWI0/1，在发生以下事件都可以触发中断，所有的 TWI 事件共用一个中断标志位。

中断事件	事件标志位	中断请求控制位
主机模式，发送完启动信号	QTWIF	TWI_SPI0_IDE ->INTEN TWI_SPI1_IDE ->INTEN
主机模式，发送完地址帧		
主机模式，接收或发送完数据帧		
从机模式，第一帧地址匹配成功		
从机模式，成功接收或发送 8 位数据		
从机模式，接收到重复起始条件		
从机模式，收到停止信号		

19.9 TWI0/1 寄存器

19.9.1 TWI0/1 相关寄存器表

19.9.1.1 TWI_SPI 控制寄存器 TWI_SPIn_CON

寄存器	读/写	说明	复位值	上电初始值
TWI_SPIn_CON (n=0/1)	读/写	TWI_SPIn 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	DWIDTH	SSHIFT	-	-	-
23	22	21	20	19	18	17	16
-	MODE	-	CPOL	CPHA	DORD	-	SMSTR
15	14	13	12	11	10	9	8
-	SPOS[1:0]		-	QWCK[3:0]			
7	6	5	4	3	2	1	0
QWEN	-	START	STOP	-	-	AA	STRETCH

位编号	位符号	说明																														
22	MODE	TWI_SPI 模式选择位 0: TWI 模式 1: 全双工 SPI 模式																														
14~13	SPOS[1:0]	<ul style="list-style-type: none"> ● TWI0 信号口映射控制位@TWI_SPI0_CON <table border="1"> <thead> <tr> <th>SPOS 值 \ 信号</th> <th>SCL0</th> <th>SDA0</th> </tr> </thead> <tbody> <tr> <td>SPOS[1:0]=00</td> <td>PC8</td> <td>PC7</td> </tr> <tr> <td>SPOS[1:0]=01</td> <td>PE0</td> <td>PD15</td> </tr> <tr> <td>SPOS[1:0]=10</td> <td>PA1</td> <td>PA0</td> </tr> <tr> <td>SPOS[1:0]=11</td> <td>PD9</td> <td>PD8</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ● TWI1 信号口映射控制位@TWI_SPI1_CON <table border="1"> <thead> <tr> <th>SPOS 值 \ 信号</th> <th>SCL1</th> <th>SDA1</th> </tr> </thead> <tbody> <tr> <td>SPOS[1:0]=00</td> <td>PC15</td> <td>PC14</td> </tr> <tr> <td>SPOS[1:0]=01</td> <td>PE2</td> <td>PE3</td> </tr> <tr> <td>SPOS[1:0]=10</td> <td>PA10</td> <td>PA11</td> </tr> <tr> <td>SPOS[1:0]=11</td> <td>PB13</td> <td>PB12</td> </tr> </tbody> </table> 	SPOS 值 \ 信号	SCL0	SDA0	SPOS[1:0]=00	PC8	PC7	SPOS[1:0]=01	PE0	PD15	SPOS[1:0]=10	PA1	PA0	SPOS[1:0]=11	PD9	PD8	SPOS 值 \ 信号	SCL1	SDA1	SPOS[1:0]=00	PC15	PC14	SPOS[1:0]=01	PE2	PE3	SPOS[1:0]=10	PA10	PA11	SPOS[1:0]=11	PB13	PB12
SPOS 值 \ 信号	SCL0	SDA0																														
SPOS[1:0]=00	PC8	PC7																														
SPOS[1:0]=01	PE0	PD15																														
SPOS[1:0]=10	PA1	PA0																														
SPOS[1:0]=11	PD9	PD8																														
SPOS 值 \ 信号	SCL1	SDA1																														
SPOS[1:0]=00	PC15	PC14																														
SPOS[1:0]=01	PE2	PE3																														
SPOS[1:0]=10	PA10	PA11																														
SPOS[1:0]=11	PB13	PB12																														
11~8	QWCK[3:0]	TWI 时钟预分频控制位 0000: f _{PCLK} 0001: f _{PCLK} /2 0010: f _{PCLK} /4 0011: f _{PCLK} /8 0100: f _{PCLK} /16 0101: f _{PCLK} /32 0110: f _{PCLK} /64 0111: f _{PCLK} /128 1000: f _{PCLK} /256 1001: f _{PCLK} /512 1010: f _{PCLK} /1024 1011: f _{PCLK} /2048																														

位编号	位符号	说明
		1100: f _{PCLK} /4096 其它: f _{PCLK} /4096 注意: 为确保 SC32F14T/14G 系列 TWI0/1 正确通信, TWI 分频请选择 4 分频及以上, 且通信频率请选择 1.5MHz 以下
7	QTWEN	TWI_QSPI 模块使能控制位 0: 关闭模块 1: 打开模块
5	START	TWI 起始位触发开关 该位写 1 将产生起始条件, TWI 将切换为主机模式。 软件可以设置或清除该位, 或当起始条件发出后, 由硬件清 0。
4	STOP	TWI 停止位触发开关 主机模式下, 对该位写 1, 在当前字节传输或起始条件发出后产生停止条件。 软件可以设置或清除该位, 或当检测到停止条件时, 由硬件清除。
1	AA	TWI 应答使能位 0: 无应答, 返回 UACK (应答位为高电平) 1: 在接收到一个匹配的地址或数据后返回一个应答 ACK
0	STRETCH	TWI 时钟延长使能位 该位仅在从机模式下有效 0: 禁止时钟延长 1: 允许时钟延长, 主机需要支持时钟延长功能 说明: 在数据传输完成之后, 且 ACK 为 0, 此时时钟延长发生。
31~29 27~24 21, 17, 15, 12, 6 3~2	-	保留

19.9.1.2 TWI_SPI 标志状态位寄存器 TWI_SPIn_STS

寄存器	读/写	说明	复位值	上电初始值
TWI_SPIn_STS (n=0/1)	读/写	TWI_SPIn 标志状态位寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
NBYTES[7:0]							
15	14	13	12	11	10	9	8
-	TMSTR	GCA	TXnE/RXnE	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	-	-	-	-	TXEIF	-	QTWIF

位编号	位符号	说明
23~16	NBYTES[7:0]	收发缓存字节数设置位 用于设置待发送/接收的字节数。 每发送/接收成功一个自己 NBYTES 自动减 1, 当 NBYTES 为 0 时 TC 标志位将置起。 注意: STA 置 1 时不允许修改。
14	TMSTR	TWI 主/从机模式标志位 0: 从机模式 1: 主机模式

位编号	位符号	说明
		<p>说明:</p> <p>1. 当 TWI 接口向总线发出起始条件后, 会自动切换为主机模式, 同时硬件将该位置位; 当总线上检测到一个停止条件时, 硬件清除该位。</p>
13	GCA	<p>TWI 通用地址响应标志位</p> <p>0: 非响应通用地址 1: 当 GC 置 1, 同时通用地址匹配时该位由硬件置 1, 并自动清 0</p>
12	TXnE/RXnE	<p>TWI 传输完成标志位</p> <p>以下情况, TXnE/RXnE 由硬件置 1</p> <ul style="list-style-type: none"> ● 主机模式: <ul style="list-style-type: none"> ■ 主机发送地址帧 (写), 且收到从机的 ACK ■ 主机发送完数据, 且接收到从机 ACK ■ 主机接收到数据, 且主机回从机 ACK ● 从机模式: <ul style="list-style-type: none"> ■ 从机接收地址帧 (读), 且和从机地址 (TWA) 匹配 ■ 从机接收到数据, 且从机回主机 ACK ■ 从机发送完数据, 且接收到主机 ACK (AA=1) <p>对 TWIDAT 进行读写操作后, 该位将被硬件清除。</p>
10~8	STATE[2:0]	<p>TWI 状态位</p> <p>用于指示 TWI 状态, 主/从机模式下状态位含义不同。</p> <ul style="list-style-type: none"> ● 从机模式: <ul style="list-style-type: none"> 000: 从机处于空闲状态, 等待 QTWEN 置 1, 检测 TWI 启动信号。当从机接收到停止条件后跳转到此状态 001: 从机正在接收第一帧地址和读写位 (第 8 位为读写位, 1 为读, 0 为写)。从机接收到起始条件后会跳转到此状态 010: 从机接收数据状态 011: 从机发送数据状态 111: 从机处于意外状态 <p>从机进入意外状态的条件如下:</p> <ul style="list-style-type: none"> ■ 在从机发送数据状态中, 当主机回 UACK 时跳转到此状态, 等待重新启动信号或停止信号 ■ 从机处于发送状态时, 将 AA 写 0 会进入此状态, 等待重新启动信号或停止信号 ■ 从机处于接收状态时, 回复 UACK 时跳转到此状态, 等待重新启动信号或停止信号 ■ 从机的地址与主机发送的地址不匹配会跳转到此状态, 等待新的起始条件或停止条件 ● 主机模式: <ul style="list-style-type: none"> 000: 状态机为空闲状态 001: 主机发送起始条件或主机正在发送从设备地址 010: 主机发送数据 011: 主机接收数据 111: 模块处于意外状态 <p>主机进入意外状态的条件如下:</p> <ul style="list-style-type: none"> ■ 主机发送停止条件或接收到从机的 UACK 信号会跳转到此状态 ■ 从机的地址与主机发送的地址不匹配会跳转到此状态
0	QTWIF	<p>TWI 中断标志位</p> <p>该位由硬件置 1, 通过软件写 1 清 0。</p> <ul style="list-style-type: none"> ● 主机模式: <ul style="list-style-type: none"> ■ 发送启动信号 ■ 发送完地址帧 ■ 接收或发送完数据帧 ● 从机模式:

位编号	位符号	说明
		<ul style="list-style-type: none"> ■ 第一帧地址匹配成功 ■ 成功接收或发送 8 位数据 ■ 接收到重复起始条件 ■ 从机收到停止信号
31~24 15, 11 6~3, 1	-	保留

19.9.1.3 TWI_SPI 地址寄存器 TWI_SPIn_ADD

寄存器	读/写	说明	复位值	上电初始值
TWI_SPIn_ADD (n=0/1)	读/写	TWI_SPIn 地址寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7~1	QTWADD[6:0]	TWI 地址寄存器 QTWADD[6:0]不能写为全 0，00H 为通用地址寻址专用。 该位在主机模式下无效。
0	GC	TWI 通用地址响应使能位 0: 禁止响应通用地址 00H 1: 允许响应通用地址 00H
31~8	-	保留

19.9.1.4 TWI_SPI 数据寄存器 TWI_SPIn_DATA

寄存器	读/写	说明	复位值	上电初始值
TWI_SPIn_DATA (n=0/1)	读/写	TWI_SPIn 数据寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7~0	QTWIDAT[7:0]	TWI 数据缓存 读操作: 从 TWI 接收 FIFO 读取接收到的数据 写操作: 向 TWI 发送 FIFO 写入待发送的数据

位编号	位符号	说明
31~16	-	保留

19.9.1.5 TWI_SPI 的中断使能及 DMA 控制寄存器 TWI_SPIn_IDE

寄存器	读/写	说明	复位值	上电初始值
TWI_SPI _n _IDE (n=0/1)	读/写	TWI_SPI _n 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7	TXDMAEN	DMA 发送通道使能位 0: 禁用 DMA 发送功能 1: 使能 DMA 发送功能 该位使能后, TXnE 置起可触发 DMA 通道发送请求
6	RXDMAEN	DMA 接收通道使能位 0: 禁用 DMA 接收功能 1: 使能 DMA 接收功能 该位使能后, RXnE 置起可触发 DMA 通道接收请求
0	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
31~8 5~3, 1	-	保留

19.9.2 TWI0/1 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
TWI_SPI0 基地址: 0x4002_0060						
TWI_SPI0_CON	0x00	读/写	TWI_SPI0 控制寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI0_STS	0x04	读/写	TWI_SPI0 标志状态位寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI0_ADD	0x08	读/写	TWI_SPI0 地址寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI0_DATA	0x0C	读/写	TWI_SPI0 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
TWI_SPI0_IDE	0x10	读/写	TWI_SPI0 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000	-

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
TWI_SPI1 基地址: 0x4002_1060						
TWI_SPI1_CON	0x00	读/写	TWI_SPI1 控制寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI1_STS	0x04	读/写	TWI_SPI1 标志状态位寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI1_ADD	0x08	读/写	TWI_SPI1 地址寄存器	0x0000_0000	0x0000_0000	-
TWI_SPI1_DATA	0x0C	读/写	TWI_SPI1 数据寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
TWI_SPI1_IDE	0x10	读/写	TWI_SPI1 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000	-

20 硬件看门狗 WDT

20.1 概述

SC32L14T/14G 系列内建一个独立的硬件看门狗 WDT，其时钟源为内部的 32kHz 振荡器 LIRC。用户可以通过编程器的 Customer Option 中的 ENWDT 控制位选择是否开启看门狗复位功能。

硬件看门狗 WDT，具有安全性高、定时准确及使用灵活的优点。此看门狗外设可检测并解决由软件错误导致的故障，并在计数器达到给定的溢出时间时触发系统复位。

WDT 由其内部低频振荡器驱动，因此即便在主时钟发生故障时仍然保持工作状态。

20.2 时钟源

SC32L14T/14G 系列的 WDT 的时钟源固定为 LIRC。WDT 使能后，LIRC 会自动开启，WDT 工作的过程中 LIRC 始终保持振荡，用户无法关闭。

20.3 WDT 寄存器

20.3.1 WDT 相关寄存器表

20.3.1.1 WDT 控制寄存器 WDT_CON

寄存器	读/写	说明	复位值	上电初始值
WDT_CON	读/写	WDT 控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
0	CLRWDT	WDT 计数清零位 该位通过软件写 1，由硬件自动清 0。 0: 无影响 1: WDT 计数器从 0 开始计数
31~1	-	保留

20.3.1.2 WDT 设置寄存器 WDT_CFG

寄存器	读/写	说明	复位值	上电初始值
WDT_CFG	读/写	WDT 设置寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明																		
2~0	WDTCKS[2:0]	看门狗时钟选择:																		
		<table border="1"> <thead> <tr> <th>WDTCKS[2:0]</th> <th>WDT 溢出时间</th> </tr> </thead> <tbody> <tr><td>000</td><td>500ms</td></tr> <tr><td>001</td><td>250ms</td></tr> <tr><td>010</td><td>125ms</td></tr> <tr><td>011</td><td>62.5ms</td></tr> <tr><td>100</td><td>31.5ms</td></tr> <tr><td>101</td><td>15.75ms</td></tr> <tr><td>110</td><td>7.88ms</td></tr> <tr><td>111</td><td>3.94ms</td></tr> </tbody> </table>	WDTCKS[2:0]	WDT 溢出时间	000	500ms	001	250ms	010	125ms	011	62.5ms	100	31.5ms	101	15.75ms	110	7.88ms	111	3.94ms
		WDTCKS[2:0]	WDT 溢出时间																	
		000	500ms																	
		001	250ms																	
		010	125ms																	
		011	62.5ms																	
		100	31.5ms																	
		101	15.75ms																	
110	7.88ms																			
111	3.94ms																			
31~3	-	保留																		

20.3.2 WDT 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
WDT 基地址: 0x4000_0330						
WDT_CON	0x0C	读/写	WDT 控制寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问
WDT_CFG	0x10	读/写	WDT 设置寄存器	0x0000_0000	0x0000_0000	不支持字节/半字访问

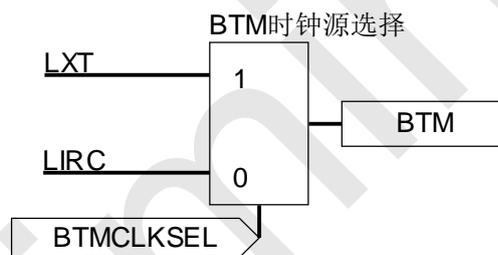
21 BTM

21.1 概述

SC32L14T/14G 系列内建一个 Base Timer (BTM)，可以按照 15.625ms ~ 32s 的间隔产生中断。32kHz LIRC 及外接 32.768kHz 晶体振荡器 LXT 都可作为 BTM 的时钟源。BTM 产生的中断可以将 CPU 从 STOP mode 唤醒。

21.2 时钟源

SC32L14T/14G 系列的 BTM 时钟源有两种：LXT 和 LIRC 可选



21.3 特性

- 时钟源 LXT 和 LIRC 可选
- 中断频率间隔 15.625ms ~ 32s 可选
- 可唤醒 STOP Mode 并且产生中断

21.4 BTM 中断

SC32L14T/14G 系列的 BTM 计数满足 BTMFS 设置条件时，BTMIF 置起，如果 BTM_CON.INTEN=1，将产生中断。

中断事件	事件标志位	中断使能控制位
BTM 中断请求	BTMIF	BTM_CON->INTEN

21.5 BTM 寄存器

21.5.1 BTM 相关寄存器表

21.5.1.1 低频定时器控制寄存器 BTM_CON

寄存器	读/写	说明	复位值	上电初始值
BTM_CON	读/写	低频定时器控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7	ENBTM	Base Timer 启动控制位 0: Base Timer 不启动 1: Base Timer 启动
6	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
3~0	BTMFS[3:0]	低频时钟中断频率选择 0000: 每 15.625ms 产生一个中断 0001: 每 31.25ms 产生一个中断 0010: 每 62.5ms 产生一个中断 0011: 每 125ms 产生一个中断 0100: 每 0.25 s 产生一个中断 0101: 每 0.5 s 产生一个中断 0110: 每 1.0 s 产生一个中断 0111: 每 2.0 s 产生一个中断 1000: 每 4.0 s 产生一个中断 1001: 每 8.0 s 产生一个中断 1010: 每 16.0 s 产生一个中断 1011: 每 32.0 s 产生一个中断 1100~1111: 保留
31~8 5~4	-	保留

21.5.1.2 BTM 标志位寄存器 BTM_STS

寄存器	读/写	说明	复位值	上电初始值
BTM_STS	读/写	BTM 标志位寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	BTMIF

位编号	位符号	说明
0	BTMIF	Base Timer 中断标志位 该位由硬件置 1，通过软件写 1 清 0。 BTM 计数满足 BTMFS 设置条件时该位自动置起。
31~1	-	保留

21.5.2 BTM 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
BTM 基地址: 0x4002_2100					
BTM_CON	0x00	读/写	低频定时器控制寄存器	0x0000_0000	0x0000_0000
BTM_STS	0x04	读/写	BTM 标志位寄存器	0x0000_0000	0x0000_0000

22 内建 CRC 校验模块

22.1 概述

SC32L14T/14G 系列内建一个 CRC 校验模块，使用多项式发生器从一个 8 位/16 位/32 位的数据字中产生 CRC 码。在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

22.2 时钟源

SC32L14T/14G 系列的 CRC 时钟源来自 HCLK

22.3 特性

- 内建了 1 个硬件 CRC 模块
- 初始值可设，默认为 0xFFFFFFFF
- 支持 8bit/16bit/32bit 数据单元
- 多项式可编程，默认为 0x04C1_1DB7
- 仅支持软件送数计算模式
- 支持 DMA: CRC_DR 可作为 DMA 的目标地址，也可直接寄存器访问
- 单独一个 byte 计算 CRC 需要 1 个系统时钟。

CRC 算法名称	CRC-32/MPEG-2
多项式公式	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
数据宽度	32bit
初始值	0xFFFF_FFFF
结果异或值	0x00000000
输入值反转	false
输出值反转	false
LSB/MSB	MSB

注意：CRCDR 写入数据和读出不是同一数据。

22.4 CRC 寄存器

22.4.1 CRC 相关寄存器表

22.4.1.1 CRC 数据寄存器 CRC_DR

寄存器	读/写	说明	复位值	上电初始值
CRC_DR	读/写	CRC 数据寄存器（被计算的数/结果）	0xFFFF_FFFF	0x0000_0000

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

位编号	位符号	说明
31~0	CRCDR[31:0]	<p>CRC 数据寄存器位 该寄存器用于向 CRC 计算器写入新数据。 读取寄存器时可读出之前的 CRC 计算结果。 如果数据大小小于 32 位，则最低有效位可用于写入 / 读取正确值。 该寄存器的操作要求如下：</p> <ol style="list-style-type: none"> 1. 首先必须对 CRC_CON.CRCRST 写 1，将 CRCDR 复位 2. 当 CRCREG 被写入时，硬件自动计算 CRC 结果，并继续存放于 CRCDR 内 <p>当需要时，即时读出 CRC 计算结果。</p>

22.4.1.2 CRC 控制寄存器 CRC_CON

寄存器	读/写	说明	复位值	上电初始值
CRC_CON	读/写	CRC 控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7~6	POLYSIZE[1:0]	CRC 多项式大小设置位 00: 32 位多项式 01: 16 位多项式 10: 8 位多项式 11: 7 位多项式
0	CRCRST	CRCDR 寄存器复位(Q31~Q0) 该位通过软件写 1, 由硬件自动清 0。 0: 无影响 1: 复位 CRCDR, 复位的值为用户写入 CRC_INIT 寄存器的值
31~8 5~1	-	保留

22.4.1.3 CRC 初始值寄存器 CRC_INIT

寄存器	读/写	说明	复位值	上电初始值
CRC_INIT	读/写	CRC 初始值寄存器	0xFFFF_FFFF	0x0000_0000

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

位编号	位符号	说明
31~0	CRC_INIT[31:0]	可编程 CRC 初始值, 复位值: 0xFFFF_FFFF 此寄存器用于写入 CRC 初始值

22.4.1.4 CRC 多项式设置寄存器 CRC_POL

寄存器	读/写	说明	复位值	上电初始值
CRC_POL	读/写	CRC 多项式设置寄存器	0x04C1_1DB7	0x0000_0000

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

位编号	位符号	说明
31~0	POL[31:0]	可编程多项式, 复位值: 0x04C1_1DB7

位编号	位符号	说明
		此寄存器用于写入要用于 CRC 计算的多项式系数，如果多项式数值小于 32 位，则必须使用最低有效位编程正确值。

22.4.2 CRC 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值	访问限制
CRC 基地址: 0x4000_2000						
CRC_DR	0x00	读/写	CRC 数据寄存器	0xFFFF_FFFF	0x0000_0000	-
CRC_CON	0x04	读/写	CRC 控制寄存器	0x0000_0000	0x0000_0000	-
CRC_INT	0x08	读/写	CRC 初始值寄存器	0xFFFF_FFFF	0x0000_0000	不支持字节/半字访问
CRC_POL	0x0C	读/写	CRC 多项式设置寄存器	0x04C1_1DB7	0x0000_0000	不支持字节/半字访问

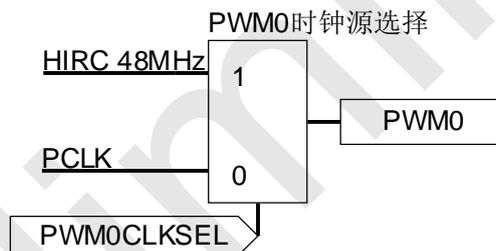
23 PWM0: 8 路 16 位多功能 PWM

23.1 概述

SC32L14T/14G 系列的 PWM0 是 8 路 16 位共周期多功能 PWM。PWM0 的功能非常丰富：支持周期及占空比的调整、输出波形类型可选择中心对齐型或边沿对齐型，输出模式可选择独立模式或互补模式，支持死区功能、支持故障检测机制。寄存器 PWM0_CON、PWM0_STS 控制 PWM 的状态及周期，各路 PWM 的打开及输出波形、波形反相及占空比均可单独调整。

23.2 时钟源

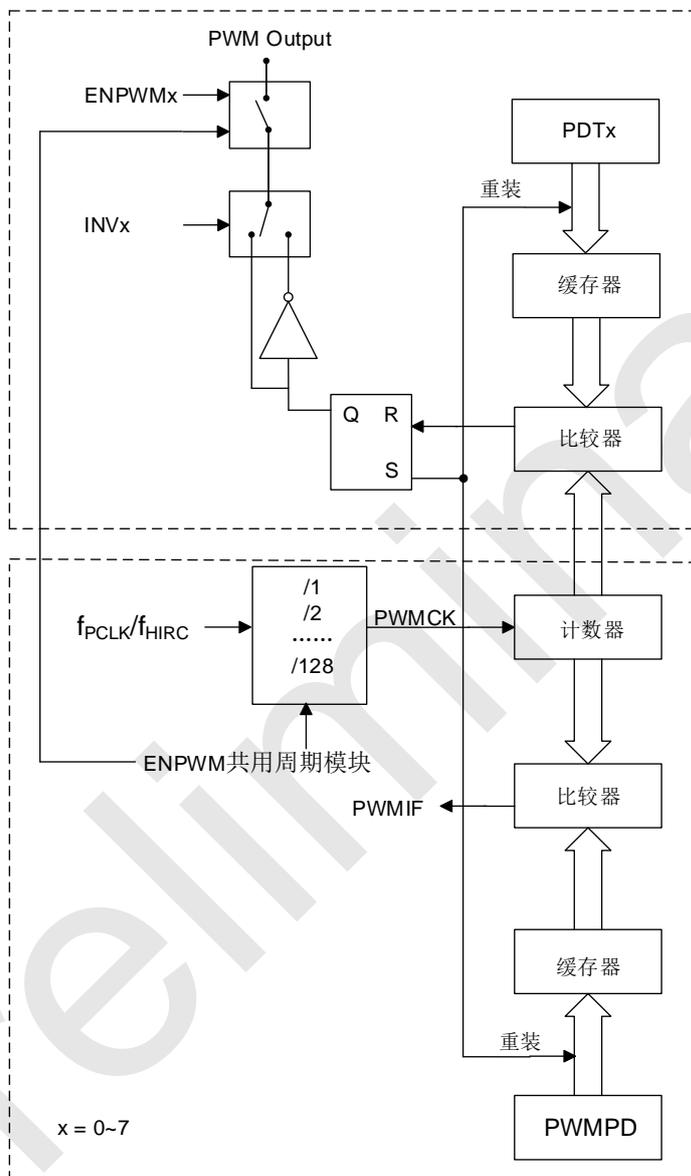
- SC32L14T/14G 系列 PWM0 的时钟源有两种：HIRC 48MHz 和 PCLK 可选
- PCLK 和 HIRC 可选
- PWM 输出频率最高可达到 HIRC



23.3 特性

- 8 路 16 位共周期多功能 PWM
- 输出波形可反向；
- 波形类型：可设为中心对齐型或边沿对齐型；
- PWM 模式：可设为独立模式或互补模式：
 - 独立模式下，8 路 PWM 周期相同，但每一路 PWM 的打开及输出波形占空比可单独调整
 - 互补模式下可同时输出四组互补、带死区的 PWM 波形；
- 提供 1 个 PWM 溢出的中断；
- 支持故障检测机制
- 有独立的中断请求标志位

23.4 PWM0 结构框图



PWM0 结构框图

23.5 PWM0 通用配置

23.5.1 输出模式

- 独立模式下 8 路 PWM 周期相同，但每一路 PWM 输出波形的占空比单独可设置。
- 互补模式下可同时输出四组共周期，互补、带死区的 PWM 波形。

23.5.2 对齐类型

23.5.2.1 边沿对齐型

PWM 计数器从 0 开始向上计数，当计数值与占空比设置项 PDT0x [15:0] 的值匹配时 PWM 输出波形切换高低电平，接着 PWM 计数器继续向上计数直至与周期设置项 PWMPD[15:0] + 1 的值匹配（一个 PWM 周期结束），PWM 计数器清零，如果 PWM 中断已使能，此时会产生 PWM 中断。PWM 输出波形为左边沿对齐方式。

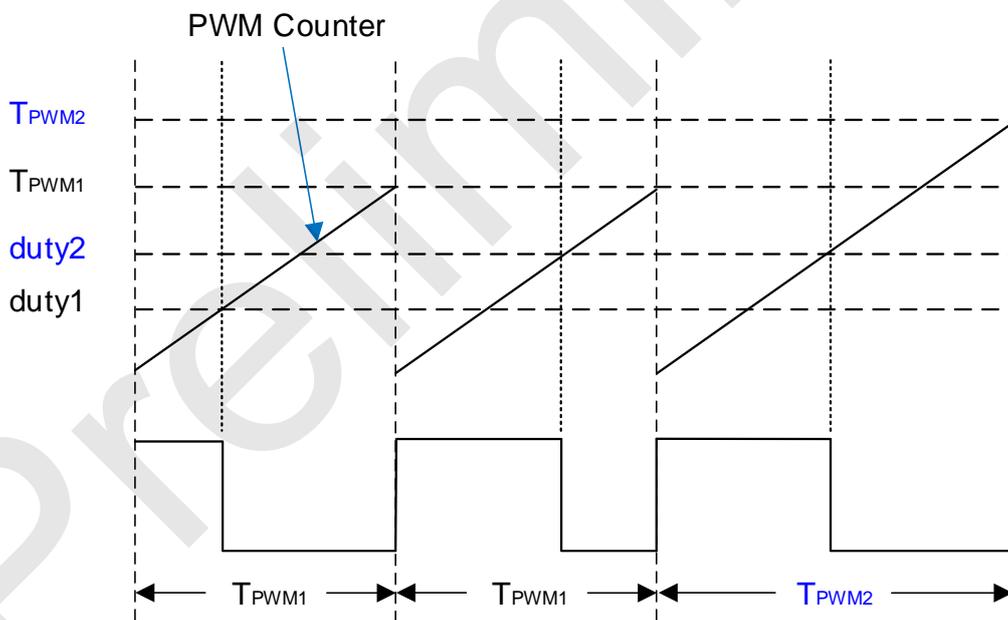
边沿对齐型周期 T_{PWM} 计算公式：

$$T_{PWM} = \frac{PWMPD[15:0] + 1}{\text{PWM 时钟频率}}$$

边沿对齐型占空比 $duty$ 计算公式：

$$duty = \frac{PDT0x [15:0]}{PWMPD[15:0] + 1}$$

边沿对齐波形图如下：



边沿对齐的 PWM

23.5.2.2 中心对齐型

PWM 计数器从 0 开始向上计数，当计数值与占空比设置项 PDT0x [15:0] 的值匹配时 PWM 输出波形切换高低电平，接着 PWM 计数器继续向上计数，当计数值与周期设置项 PWMPD[15:0] + 1 的值匹配时（即 PWM 周期的中点）自动开始向下计数，当计数值与 PDT0x [15:0] 的值再次匹配时 PWM 输出波形再次切换高低电平，接着 PWM 计数器继续向下计数直至溢出（一个 PWM 周期结束），如果 PWM 中断已使能，此时会产生 PWM 中断。

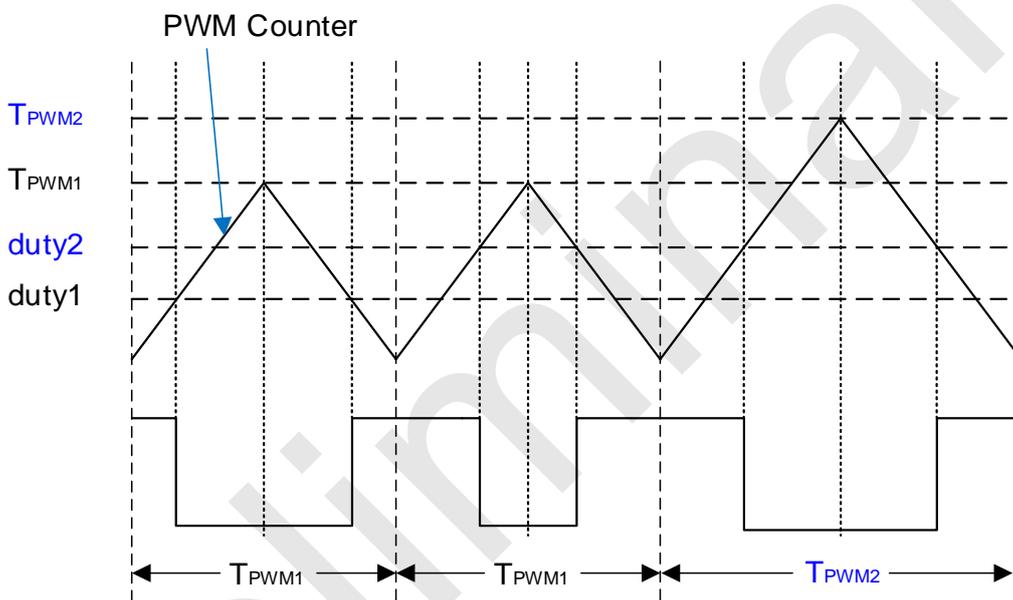
中心对齐型周期 T_{PWM} 计算公式:

$$T_{PWM} = 2 * \frac{PWMPD[15:0] + 1}{PWM \text{ 时钟频率}}$$

中心对齐型占空比 $duty$ 计算公式:

$$duty = \frac{PDT0x [15:0]}{PWMPD[15:0] + 1}$$

中心对齐波形图如下:



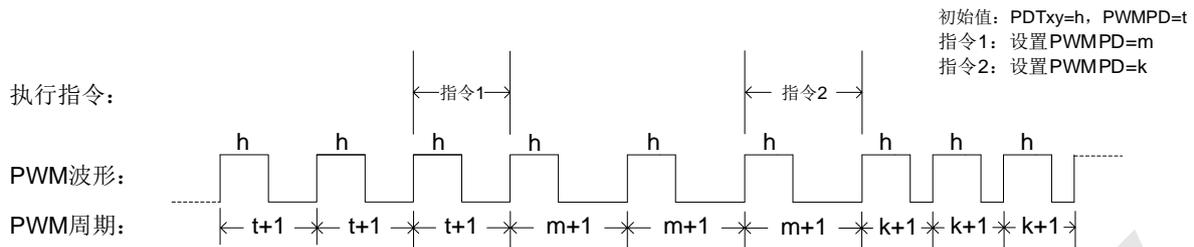
中心对齐的 PWM

23.5.3 占空比变化特性

当 PWM0n 输出波形时, 若需改变占空比, 可通过改变高电平设置寄存器(PDT0x)的值实现。但需要注意: 更改 PDT0x 的值, 占空比不会立即改变, 而是等待 PWM 计数器计数到 0 或向上计数至与周期设置项 PWMPD[15:0] + 1 的值匹配时改变。

23.5.4 周期变化特性

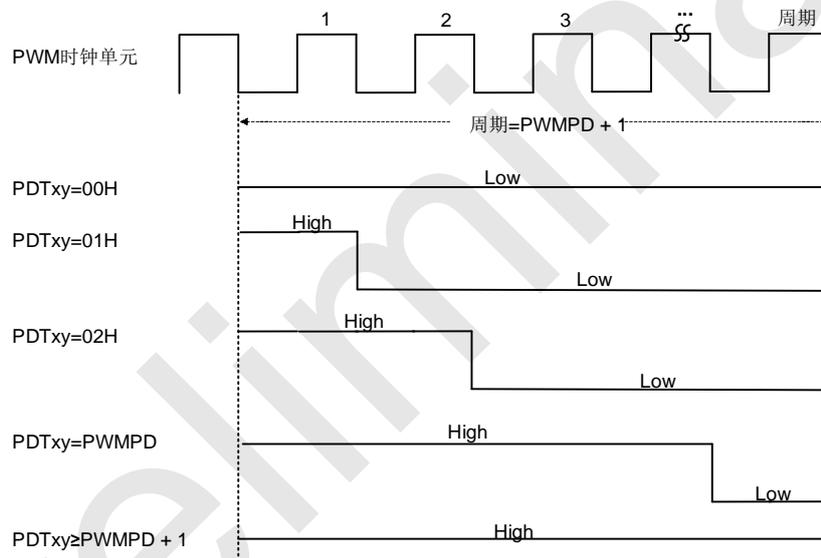
当 PWM 输出波形时, 若需改变周期, 可通过改变周期设置寄存器 PWMPD 的值实现。更改 PWMPD 的值, 周期不会立即改变, 而是等待 PWM 计数器计数到 0 或向上计数至与周期设置项 PWMPD[15:0] + 1 的值匹配时改变, 参考上图所示。



周期变化特性图

23.5.5 周期和占空比的关系

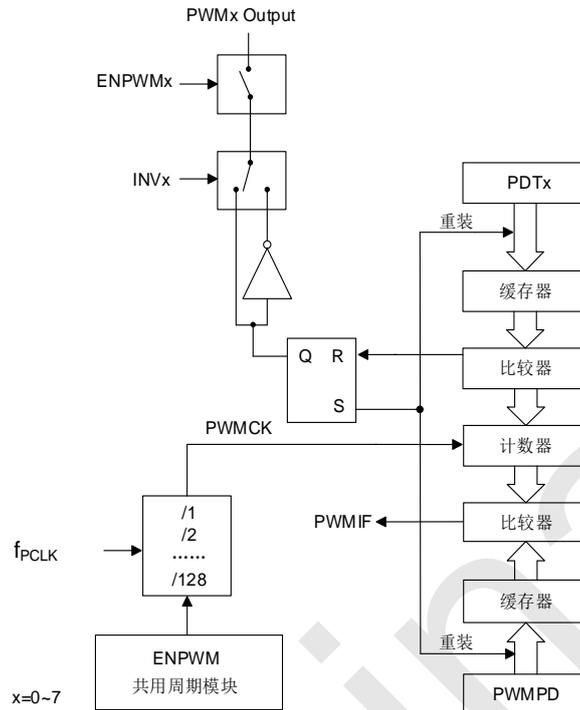
周期和占空比的关系如上图所示。该结果的前提是 PWM 输出反向控制(INVx, x=0~7)初始为 0, 若需得到相反结果, 可置 INVx 为 1。



周期与占空比关系图

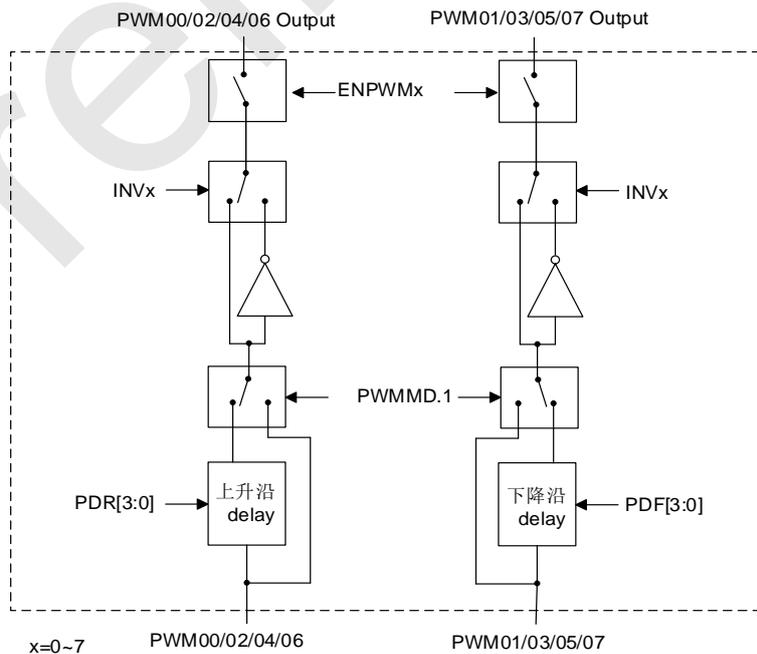
23.6 PWM0 独立模式

独立模式下(PWM0_CON.PWMMMD1=0), 8 路 PWM 通道的占空比均可独立设置。用户配置好 PWM 的输出状态及周期, 再通过配置相应 PWM 通道的占空比寄存器即可按固定占空比输出 PWM 波形。



SC32L14T PWM0 独立模式框图

23.7 PWM0 互补模式



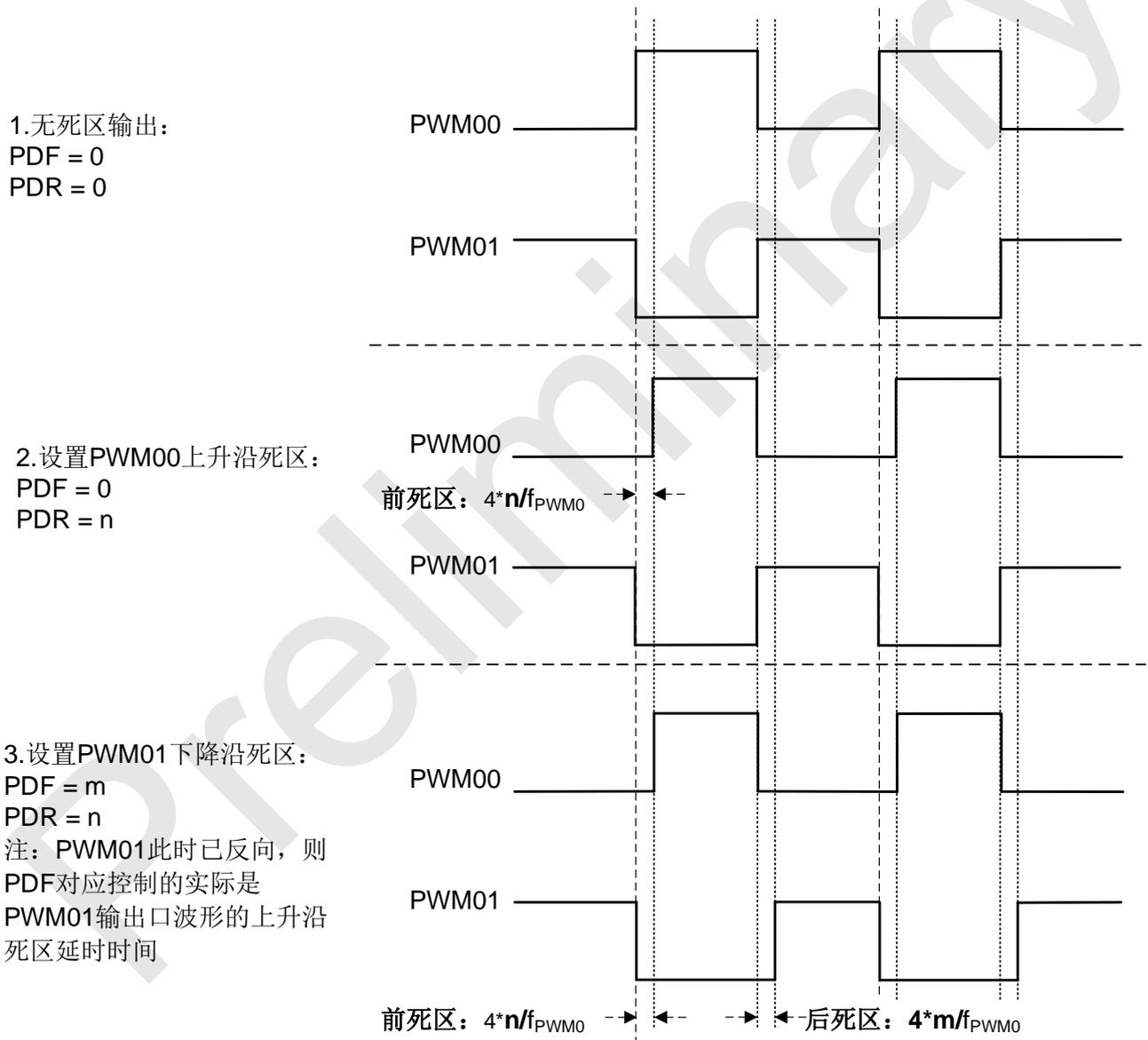
SC32L14T PWM0 互补模式框图

23.7.1 PWM0 互补模式死区时间设置

当 SC32L14T 的 PWM0 工作在互补模式时，死区控制模块能够防止互补输出的两路 PWM 信号有效时区的互相交叠，以保证实际应用中 PWM 信号驱动的一对互补功率开关管不会同时导通。

23.7.2 PWM0 死区输出波形

下图是以 PWM00 和 PWM01 在互补模式下的死区时间调整波形图，为了便于区分，PWM01 已反向（INV1=1）。



PWM0 死区输出波形

23.8 PWM0 故障检测功能设置

故障检测功能常应用于电机系统的防护。当故障检测功能开启，PWM0_FLT.FLTEN 置 1，故障检测信号输

入脚(FLT)生效。当 FLT 管脚的信号满足故障条件，标志位 FLTSTA 通过硬件置 1，PWM 输出停止，PWM 计数器仍保持计数，PWM 中断不受影响。故障检测模式分为锁存模式和立即模式；立即模式下，当 FLT 管脚上的故障信号满足失能条件，标志位 FLTSTA 通过硬件清零，直到 PWM 计数器归零后 PWM 恢复输出；锁存模式下，当 FLT 管脚上的故障信号满足失能条件，标志位 FLTSTA 状态保持不变，用户可通过软件清零，FLTSTA 状态一旦清零，PWM 计数器恢复计数，直到 PWM 计数器归零后 PWM 恢复输出。

23.9 PWM0 中断

SC32L14T/14G 系列的 PWM 完成一个周期的输出后，PWMIF 会置起，如果 PWM0_CON.INTEN=1，将产生中断。

中断事件	事件标志位	中断使能控制位
PWM0 中断请求	PWMIF	PWM0_CON->INTEN

23.10 PWM0 寄存器

23.10.1 PWM0 相关寄存器表

23.10.1.1 PWM0 控制寄存器 PWM0_CON

寄存器	读/写	说明	复位值	上电初始值
PWM0_CON	读/写	PWM0 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	INTEN
7	6	5	4	3	2	1	0
ENPWM	PWMMD0	PWMMD1	-	-	PWMCK[2:0]		

位编号	位符号	说明
8	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
7	ENPWM	PWM 模块开关控制位 1: 允许 Clock 进到 PWM 单元，PWM 处于工作状态，PWM 输出口的状态由寄存器 ENPWMx 控制 (x=0~7) 0: PWM 单元停止工作，PWM 计数器清 0，全部 PWM 输出口设置为 GPIO 状态
6	PWMMD0	PWM 波形对齐模式选择位 0: 边沿对齐模式 1: 中心对齐模式

位编号	位符号	说明
5	PWMMD1	PWM 波形互补模式设置位 0: 独立模式 1: 互补模式
2~0	PWMCK[2:0]	PWM 时钟频率档位控制位 用于设定 PWM 时钟频率 f_{PWM0} 000: $f_{SOURCE}/1$ 001: $f_{SOURCE}/2$ 010: $f_{SOURCE}/4$ 011: $f_{SOURCE}/8$ 100: $f_{SOURCE}/16$ 101: $f_{SOURCE}/32$ 110: $f_{SOURCE}/64$ 111: $f_{SOURCE}/128$ 说明: f_{SOURCE} 受 PWM0CLKSEL 影响, 时钟源可选 PCLK 或 48MHz HIRC
31~9 4~3	-	保留

23.10.1.2 PWM0 通道设置寄存器 PWM0_CHN

寄存器	读/写	说明	复位值	上电初始值
PWM0_CHN	读/写	PWM0 通道设置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
ENPWM7	ENPWM6	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0

位编号	位符号	说明
7~0	ENPWMx (x=0~7)	PWM0x 波形输出选择 0: PWM0x 输出被关闭并作为 GPIO 1: 当 ENPWM=1 时, PWM0x 所在的管脚作为波形输出口 说明: 如果 ENPWM 置 1, PWM 模块被打开, 但 ENPWMx=0, PWM 输出被关闭并作为 GPIO 口。此时 PWM 模块可以作为一个 16 位 Timer 使用, 若此时 PWM0_CON.INTEN = 1, PWM 仍然会产生中断。
31~9	-	保留

23.10.1.3 PWM0 状态标志寄存器 PWM0_STS

寄存器	读/写	说明	复位值	上电初始值
PWM0_STS	读/写	PWM0 状态标志寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
1	FLTSTA	PWM 故障检测状态标志位 该位由硬件置 1，通过软件写 1 清 0。0：PWM 处于正常输出状态； 1：故障检测有效，PWM 输出处于高阻状态，如果处于锁存模式，此 位可软件清 0
0	PWMIF	PWM 中断请求标志位 该位由硬件置 1，通过软件写 1 清 0。 当 PWM 计数器溢出时(即：计数值超过 PWMPD)，该位由硬件置 1。 如果此时 PWM0_CON.INTEN = 1，将产生 PWM0 中断。
31~2	-	保留

23.10.1.4 PWM0 波形输出反向控制寄存器 PWM0_INV

寄存器	读/写	说明	复位值	上电初始值
PWM0_INV	读/写	PWM0 波形输出反向控制 寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
INV7	INV6	INV5	INV4	INV3	INV2	INV1	INV0

位编号	位符号	说明
7~0	INVx (x=0~7)	PWM0x 波形输出反向控制 1：PWM0x 波形输出反向 0：PWM0x 波形输出不反向
31~8	-	保留

23.10.1.5 PWM0 死区设置寄存器 PWM0_DFR

寄存器	读/写	说明	复位值	上电初始值
PWM0_DFR	读/写	PWM0 死区设置寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
11~8	PDF[3:0]	下降沿死区时间设置位 此位仅在互补模式设置有效： PWM 下降沿死区时间 = $4 * PDF[3:0] / f_{PWM0}$
3~0	PDR[3:0]	上升沿死区时间设置位 此位仅在互补模式设置有效： PWM 上升沿死区时间 = $4 * PDR[3:0] / f_{PWM0}$
31~12 7~4	-	保留

23.10.1.6 PWM0 故障检测设置寄存器 PWM0_FLT

寄存器	读/写	说明	复位值	上电初始值
PWM0_FLT	读/写	PWM0 故障检测设置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
FLTEN	-	FLTMD	FLTLV	-	-	FLTDT[1:0]	

位编号	位符号	说明
7	FLTEN	PWM 故障检测功能控制位 0: 故障检测功能关闭 1: 故障检测功能开启
5	FLTMD	PWM 故障检测模式设置位 0: 锁存模式，当故障输入有效时，故障检测状态标志位 FLTSTA 由硬件置 1，PWM 停止输出，当故障输入无效时 FLTSTA 状态不变 1: 立即模式：当故障输入有效时，故障检测状态标志位 FLTSTA 由硬件置 1，PWM 停止输出，当故障输入无效时 FLTSTA 状态立刻被硬件清 0，PWM 波形将在 PWM 计数器计数到 0 时恢复输出
4	FLTLV	PWM 故障检测电平选择位 0: 故障检测低电平有效 1: 故障检测高电平有效
1~0	FLTDT[1:0]	PWM 故障检测输入信号滤波时间设置 00: 滤波时间为 0

位编号	位符号	说明
		01: 滤波时间为 1us 10: 滤波时间为 4us 11: 滤波时间为 16us
31~8 6, 3~2	-	保留

23.10.1.7 PWM0 周期寄存器 PWM0_CYCLE

寄存器	读/写	说明	复位值	上电初始值
PWM0_CYCLE	读/写	PWM0 周期寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	PWMPD[15:0]	PWM0 周期设置位 此数值代表 PWM 输出波形的 (周期 - 1); 即 PWM 输出的周期值为 (PWMPD[15:0] + 1) * f _{PWM0} ;
31~16	-	保留

23.10.1.8 PWM0 通道占空比调节寄存器 PWM0_DT_x (x = 0~7)

寄存器	读/写	说明	复位值	上电初始值
PWM0_DT _x (x = 0~7)	读/写	PWM0 通道 n duty 寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	PDT[15:0]	PWM0 _x 波形占空比长度设置, x = 0~7 <ul style="list-style-type: none"> 独立模式: PWM0_x 的波形的高电平宽度是 PDT_x [15:0]个 PWM 时钟; 互补模式: 互补通道 PWM0_x 和 PWM0_y, y=x+1, PWM0_x 和 PWM0_y 的波形的高电平宽度是 PDT_x [15:0]个 PWM 时钟。

位编号	位符号	说明
31~16	-	保留

23.10.2 PWM0 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
PWM0 基地址: 0x4002_0200					
PWM0_CON	0x00	读/写	PWM0 控制寄存器	0x0000_0000	0x0000_0000
PWM0_CHN	0x04	读/写	PWM0 通道设置寄存器	0x0000_0000	0x0000_0000
PWM0_STS	0x08	读/写	PWM0 状态标志寄存器	0x0000_0000	0x0000_0000
PWM0_INV	0x0C	读/写	PWM0 波形输出反向控制寄存器	0x0000_0000	0x0000_0000
PWM0_DFR	0x10	读/写	PWM0 死区设置寄存器	0x0000_0000	0x0000_0000
PWM0_FLT	0x14	读/写	PWM0 故障检测设置寄存器	0x0000_0000	0x0000_0000
PWM0_CYCL E	0x18	读/写	PWM0 周期寄存器	0x0000_0000	0x0000_0000
PWM0_DT _x (x = 0~7) 基地址: 0x4002_0230					
PWM0_DT0	0x00	读/写	PWM0 通道 0 duty 寄存器	0x0000_0000	0x0000_0000
PWM0_DT1	0x04	读/写	PWM0 通道 1 duty 寄存器	0x0000_0000	0x0000_0000
PWM0_DT2	0x08	读/写	PWM0 通道 2 duty 寄存器	0x0000_0000	0x0000_0000
PWM0_DT3	0x0C	读/写	PWM0 通道 3 duty 寄存器	0x0000_0000	0x0000_0000
PWM0_DT4	0x10	读/写	PWM0 通道 4 duty 寄存器	0x0000_0000	0x0000_0000
PWM0_DT5	0x14	读/写	PWM0 通道 5 duty 寄存器	0x0000_0000	0x0000_0000
PWM0_DT6	0x18	读/写	PWM0 通道 6 duty 寄存器	0x0000_0000	0x0000_0000
PWM0_DT7	0x1C	读/写	PWM0 通道 7 duty 寄存器	0x0000_0000	0x0000_0000

24 32 路高灵敏度触控电路 (TK)

- 高灵敏度模式
- 可适应隔空按键触控、接近感应等对灵敏度要求较高的触控应用
- 可通过 10V 动态 CS 测试
- 通道可以并联扫描
- CMOD 管脚需对地接入 103 电容
- 支持自电容方案和互电容模式
- 支持低功耗模式
- 支持快速唤醒 STOP Mode
- 全套开发支持：高灵活触控软件库，智能化调试软件

25 实时时钟（RTC）

25.1 概述

SC32L14T/14G 系列内部集成一个实时时钟模块（简称 RTC），RTC 是一个独立的 BCD 定时器/计数器，可提供具有可编程闹钟功能的日历时钟/日历。且 RTC 可以工作在正常模式、低功耗模式或复位状态中，在低功耗模式下，RTC 还可作为低功耗模式的唤醒单元。

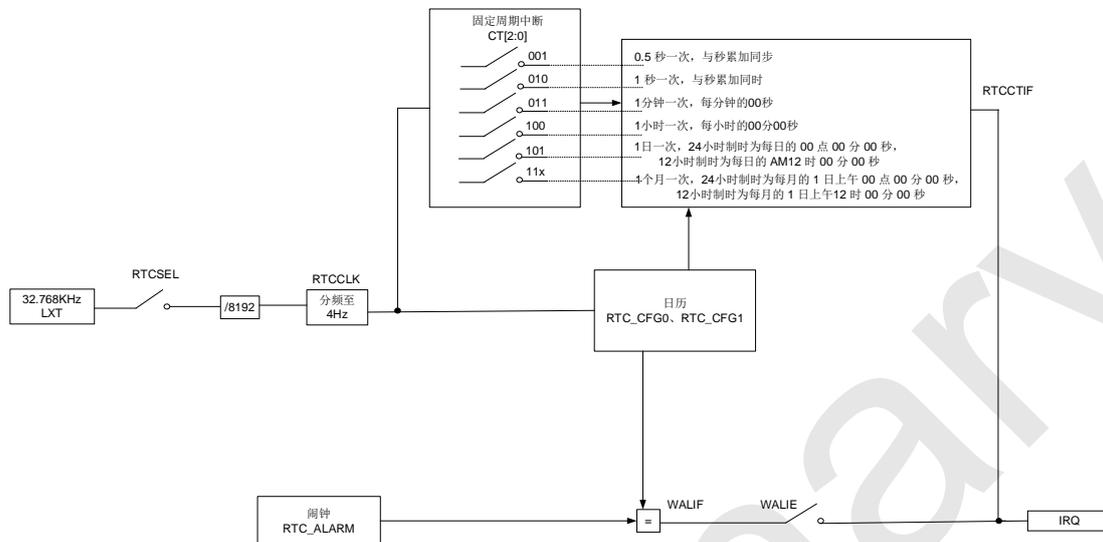
25.2 时钟源

SC32L14T/14G 系列 RTC 的时钟源仅一种：来自 LXT

25.3 特性

- 时钟源为 LXT
- 可在低功耗模式下运行，支持从 STOP Mode 唤醒
- 万年历功能
 - 支持 BCD 格式时间/日期寄存器
 - 硬件自动处理闰年修正
 - 精度跟随外部 32.768K 晶振（LXT）
- 中断系统
 - 可设闹钟中断
 - 可配置固定中断周期：年、月、日、小时、分钟、秒或半秒（0.5 秒）
 - 支持通过中断事件唤醒 STOP 模式
- 时间模式配置
 - 支持 24 小时制或 12 小时制时间显示模式
- 安全读写机制：
 - 独立半秒标志位（Half-Second Flag）：用户可在半秒标志置位时执行寄存器读写操作，规避进位过程导致的时序冲突
- 寄存器同步设计：写入时间数据后，读操作立即返回写入值（缓冲机制），但实际生效需 250ms 同步周期（此期间寄存器未更新）

25.4 结构框图



25.5 功能说明

25.5.1 RTC 时钟

RTC 时钟源可通过 RTC 控制器 RTC_CON[4:3]选择关闭时钟或者 LXT 作为时钟源。选择关闭时钟时, RTC 不工作。

注意: 在配置 RTC 寄存器前, 需要先使能 RTC 寄存器的时钟。由于 RTC 外设挂在 APB2 总线上, 因此需要先使能 APB2 时钟。

25.5.2 实时时钟和日历

用户可以直接读取 RTC_CFG0/1 寄存器获取实时时钟和日历, 读取的时钟和日历均为 BCD 码。

- RTC_CFG0[31:24]对应于年
- RTC_CFG0[20:16]对应于月
- RTC_CFG0[10:8]对应于星期
- RTC_CFG0[5:0]对应于日
- RTC_CFG1[21:16]对应于小时
- RTC_CFG1[14:8]对应于分钟
- RTC_CFG1[6:0]对应于秒

注意: 用户在初始化设置时钟和日历时, 为避免读写时遇到进位, 需要等待 RTC_CFG1[7]半秒位置起后, 再进行操作。半秒位置起后会自动清零, 用户无需对其进行清零操作。

25.5.3 可编程闹钟

RTC 单元提供一个可编程闹钟，用户可通过 RTC_ALARM 寄存器配置闹钟功能。如果日历分钟、小时和星期与闹钟寄存器 RTC_ALARM 中编程的值相匹配，则 RTC_STS[0] WALIF 标志会被置为 1；若此时闹钟中断使能位 RTC_CON.WALIE=1 且总中断使能位 RTC_CON.INTEN=1，则会触发 RTC 中断。

注意：通过 RTC_ALARM 寄存器配置闹钟时，小时和分钟采用 BCD 码配置，星期与寄存器位对应关系如下表。

星期	星期六	星期五	星期四	星期三	星期二	星期一	星期日
Bit 位	WW6	WW5	WW4	WW3	WW2	WW1	WW0

25.5.4 固定周期中断

RTC 单元提供一个固定周期中断，用户可通过 RTC_CON 寄存器的 CT[2:0]选择进入固定周期中断的周期或者不使用此功能。固定周期可设置为以下模式：

- 0.5 秒：与秒累加同步置起标志位
- 1 秒：与秒累加同时置起标志位
- 1 分钟：每分钟的 00 秒置起标志位
- 1 小时：每小时的 00 分 00 秒置起标志位
- 1 日：24 小时制时为每日的 00 点 00 分 00 秒，12 小时制时为每日的 AM12 时 00 分 00 秒
- 1 月：24 小时制时为每月的 1 日上午 00 点 00 分 00 秒，12 小时制时为每月的 1 日上午 12 时 00 分 00 秒

当日历运行到达以上设定周期时，RTCCTIF 标志位会置起，若此时总中断 RTC_CON.INTEN=1，则会触发 RTC 中断。

注意：固定周期中断置起时间与设置的时间无关，只在固定时间置起。例如用户设置固定时间周期为 1 分钟，无论用户是在上一分钟的哪一秒设置并开启了固定周期中断，固定周期中断的置起时间一定是在每分钟的 00 秒。

25.5.5 RTC 初始化和配置

25.5.5.1 日历初始化和配置

要编程初始时间的日期日历值，需按照以下顺序操作：

1. 在 RCC_CFG0 寄存器中使能 LXT 时钟源 LXTEN
2. 在 RTC_CON 寄存器中配置 RTC 时钟源 RTCSEL[1:0]为 LXT，并选择小时系统 AMPM 为 12 小时制或 24 小时制
3. 等待 RTC_CFG1 寄存器 HSEC 半秒位置起后，将初始日期值的 BCD 码写入 RTC_CFG0 寄存器，星期配置如下表：

星期	WEEK
星期日	00H
星期一	01H
星期二	02H

星期	WEEK
星期三	03H
星期四	04H
星期五	05H
星期六	06H

4. 等待 RTC_CFG1 寄存器 HSEC 半秒位置起后，将初始时间值的 BCD 码写入 RTC_CFG1 寄存器，不同小时系统时间配置如下表：

24 小时表示 (AMPM=1)		12 小时表示 (AMPM=0)	
时间	HOUR 寄存器	时间	HOUR 寄存器
0 时	00H	AM 12 时	12H
1 时	01H	AM 1 时	01H
2 时	02H	AM 2 时	02H
3 时	03H	AM 3 时	03H
4 时	04H	AM 4 时	04H
5 时	05H	AM 5 时	05H
6 时	06H	AM 6 时	06H
7 时	07H	AM 7 时	07H
8 时	08H	AM 8 时	08H
9 时	09H	AM 9 时	09H
10 时	10H	AM 10 时	10H
11 时	11H	AM 11 时	11H
12 时	12H	PM 12 时	32H
13 时	13H	PM 1 时	21H
14 时	14H	PM 2 时	22H
15 时	15H	PM 3 时	23H
16 时	16H	PM 4 时	24H
17 时	17H	PM 5 时	25H
18 时	18H	PM 6 时	26H
19 时	19H	PM 7 时	27H
20 时	20H	PM 8 时	28H
21 时	21H	PM 9 时	29H
22 时	22H	PM 10 时	30H
23 时	23H	PM 11 时	31H

当初始化序列完成之后，日历开始计数。

25.5.5.2 写入日历

初始化完成后，若需改写日历寄存器，需进行以下操作：

等待 RTC_CFG1[7]半秒置起后，对 RTC_CFG0 和 RTC_CFG1 写入目标时间的 BCD 值。

注意：用户在读写日历时，为避免遇到进位，建议等待 RTC_CFG1[7]半秒位置起后，再进行操作。半秒位置起后会自动清零，用户无需对其进行清零操作。

25.5.5.3 读取日历

初始化完成后，读取日历寄存器时，需进行以下操作：

等待 RTC_CFG1[7]半秒置起后，对 RTC_CFG0 读取当前日期的 BCD 值，对 RTC_CFG1 读取当前时间的 BCD 值。

注意：用户在读写日历时，为避免遇到进位，建议等待 RTC_CFG1[7]半秒位置起后，再进行操作。半秒位置起后会自动清零，用户无需对其进行清零操作。

25.5.5.4 编程闹钟

初始化完成后，对 RTC_ALARM 寄存器进行星期、小时和分钟配置，若星期配置为 00H，或小时或分钟设置为范围以外的值，则不检测闹钟。

例如：

星期配置如下表：

闹钟设定	星期日 WW0	星期一 WW1	星期二 WW2	星期三 WW3	星期四 WW4	星期五 WW5	星期六 WW6
每天	1	1	1	1	1	1	1
星期一	0	1	0	0	0	0	0
星期一~星期五	0	1	1	1	1	1	0

小时以 BCD 码设定十进制的 00 ~ 23 或者 01 ~ 12、21 ~ 32，配置如下表：

闹钟设置		WH20	WH10	WH8	WH4	WH2	WH1
上午 1 时 01: 00	12 小时制	0	0	0	0	0	1
	24 小时制	0	0	0	0	0	1
下午 1 时 13: 00	12 小时制	1	0	0	0	0	1
	24 小时制	0	1	0	0	1	1

注意：若小时系统设置位 AMPM 发生变化，用户也需同步调整闹钟寄存器中小时寄存器的时钟格式，否则可能会导致日历与闹钟中的小时时钟格式不匹配从而无法正确触发闹钟功能。

分钟以 BCD 码设定十进制的 00 ~ 59，配置如下：

闹钟设置	WM40	WM20	WM10	WM8	WM4	WM2	WM1
30 分	0	1	1	0	0	0	0
59 分	1	0	1	1	0	0	1

在检测到闹钟一致，经过 1 个 f_{RTC} 时钟后，WALIF 闹钟检测标志位置起；若此时闹钟中断使能位 RTC_CON.WALIE=1 且总中断使能位 RTC_CON.INTEN=1，则会触发 RTC 中断；WALIF 闹钟检测标志位需用户软件写 1 清 0。

25.5.6 低功耗应用

RTC 可在低功耗模式下运行，并可正常触发闹钟和固定周期中断，且 RTC 中断事件可唤醒 STOP 模式。

25.6 RTC 中断

SC32L14T/14G 系列的 RTC 在日历时间与闹钟设定的分、时以及星期相匹配时，闹钟检测标志位 WALIF 将会置起；若 RTC 开启后时间经过了 CT[2:0]设置的固定周期，固定周期中断标志位 RTCCTIF 将会置起。若此时 RTC_CON.INTEN=1，将会产生中断。

中断事件	中断请求控制位	事件标志位	中断使能子开关
日历时间与闹钟相匹配	RTC_CON ->INTEN	WALIE	WALIF

中断事件	中断请求控制位	事件标志位	中断使能子开关
经过固定周期设置时间		RTCCTIF	\

25.7 RTC 寄存器

25.7.1 RTC 相关寄存器表

25.7.1.1 RTC 控制寄存器 RTC_CON

寄存器	读/写	说明	复位值	上电初始值
RTC_CON	读/写	RTC 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	WALIE	INTEN
7	6	5	4	3	2	1	0
-	-	AMPM	-	RTCSEL	CT[2:0]		

位编号	位符号	说明
9	WALIE	闹钟中断使能控制位 0: WALIF 置起时禁止产生中断 1: WALIF 置起时允许产生中断
8	INTEN	中断请求 CPU 的使能位 0: 中断请求屏蔽 1: 中断请求开启
5	AMPM	小时系统设置位 0: 12 小时系统 1: 24 小时系统 当 AMPM=0 时, HOUR 寄存器的 bit5 (HOUR20) 表示上午或下午: HOUR20=0 时表示上午 AM, HOUR20=1 时表示下午 PM。 注意: 建议用户在 HSEC=1 时更改 AMPM 的值, 更改 AMPM 的值后, RCC_CFG1 中的小时位会改变为所设小时系统的对应值。
3	RTCSEL	RTC 时钟源选择位 0: 无时钟, 即关闭 RTC 时钟源 1: 选择 LXT 为 RTC 时钟源
2~0	CT[2:0]	固定周期中断时间设置位 000: 不使用固定周期中断功能。 001: 0.5 秒一次 (与秒累加同步) 010: 1 秒一次 (与秒累加同时) 011: 1 分钟一次 (每分钟的 00 秒) 100: 1 小时一次 (每小时的 00 分 00 秒) 101: 1 日一次 (24 小时制时为每日的 00 点 00 分 00 秒, 12 小时制时为每日的 AM12 时 00 分 00 秒) 11x: 1 个月一次 (24 小时制时为每月的 1 日上午 00 点 00 分 00 秒, 12 小时制时为每月的 1 日上午 12 时 00 分 00 秒)
31~10 7~6, 4	-	保留

25.7.1.2 RTC 配置寄存器 0 RTC_CFG0

寄存器	读/写	说明	复位值	上电初始值
RTC_CFG0	读/写	RTC 配置寄存器 0	0x0001_0001	0x0000_0000

31	30	29	28	27	26	25	24
YEART[3:0]			YEARU[3:0]				
23	22	21	20	19	18	17	16
-	-	-	MONTHT	MONTHU[3:0]			
15	14	13	12	11	10	9	8
-	-	-	-	-	WEEKU[2:0]		
7	6	5	4	3	2	1	0
-	-	DAYT[1:0]		DAYU[3:0]			

位编号	位符号	说明						
31~28	YEART[3:0]	<p>年计数值的十位 以 BCD 格式设置的值表示年计数值的十位，且通过月计数器（MONTH）的上溢进行递增计数。 00、04、08、……、92、96 是闰年。 在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中若发生月寄存器上溢，将忽视月寄存器的上溢并且设定为写入值。</p>						
27~24	YEARU[3:0]	<p>年计数值的个位 以 BCD 格式设置的值表示年计数值的个位，且通过月计数器（MONTH）的上溢进行递增计数。 00、04、08、……、92、96 是闰年。 在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中若发生月寄存器上溢，将忽视月寄存器的上溢并且设定为写入值。</p>						
20	MONTHT	<p>月计数值的十位 以 BCD 格式设置的值表示月计数值的十位，且通过日计数器（DAY）的上溢进行递增计数。 在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中若发生日寄存器上溢，将忽视日寄存器的上溢并且设定为写入值。</p>						
19~16	MONTHU[3:0]	<p>月计数值的个位 以 BCD 格式设置的值表示月计数值的个位，且通过日计数器（DAY）的上溢进行递增计数。 在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中若发生日寄存器上溢，将忽视日寄存器的上溢并且设定为写入值。</p>						
10~8	WEEKU[2:0]	<p>星期计数值的个位 以 BCD 格式设置的值表示星期计数值的个位，且通过日计数器（DAY）的上溢进行递增计数。 在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。 注意：星期计数位（WEEK）不会根据月计数位（MONTH）和日计数位（DAY）的设置值自动设置对应星期。用户必须在复位完成后根据下表软件进行星期设置：</p> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th>星期</th> <th>WEEK</th> </tr> </thead> <tbody> <tr> <td>星期日</td> <td>00H</td> </tr> <tr> <td>星期一</td> <td>01H</td> </tr> </tbody> </table>	星期	WEEK	星期日	00H	星期一	01H
星期	WEEK							
星期日	00H							
星期一	01H							

位编号	位符号	说明										
		<table border="1"> <tr><td>星期二</td><td>02H</td></tr> <tr><td>星期三</td><td>03H</td></tr> <tr><td>星期四</td><td>04H</td></tr> <tr><td>星期五</td><td>05H</td></tr> <tr><td>星期六</td><td>06H</td></tr> </table>	星期二	02H	星期三	03H	星期四	04H	星期五	05H	星期六	06H
星期二	02H											
星期三	03H											
星期四	04H											
星期五	05H											
星期六	06H											
5~4	DAYT[1:0]	<p>日计数值的十位 以 BCD 格式设置的值表示日计数值的十位，且通过小时计数器（HOUR）的上溢进行递增计数。 计数器进行以下的计数。</p> <ul style="list-style-type: none"> • 01~ 31（1、3、5、7、8、10、12月） • 01~ 30（4、6、9、11月） • 01~ 29（2月 闰年） • 01~ 28（2月 非闰年） <p>在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中若发生小时寄存器上溢，将忽视小时计数寄存器的上溢并且设定为写入值。</p>										
3~0	DAYU[3:0]	<p>日计数值的个位 以 BCD 格式设置的值表示日计数值的个位，且通过小时计数器（HOUR）的上溢进行递增计数。 计数器进行以下的计数。</p> <ul style="list-style-type: none"> • 01~ 31（1、3、5、7、8、10、12月） • 01~ 30（4、6、9、11月） • 01~ 29（2月 闰年） • 01~ 28（2月 非闰年） <p>在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中若发生小时寄存器上溢，将忽视小时计数寄存器的上溢并且设定为写入值。</p>										
23~21 15~11 7~6	-	保留										

25.7.1.3 RTC 配置寄存器 1 RTC_CFG1

寄存器	读/写	说明	复位值	上电初始值
RTC_CFG1	读/写	RTC 配置寄存器 1	0x0012_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	HOURT[1:0]		HOURU[3:0]			
15	14	13	12	11	10	9	8
-	MINT[2:0]			MINU[3:0]			
7	6	5	4	3	2	1	0
HSEC	SECT[2:0]			SECU[3:0]			

位编号	位符号	说明
21~20	HOURT[1:0]	<p>小时计数值的十位 根据 AMPM 设定的小时系统，以 BCD 格式设置的值表示小时计数值的十位，且通过分钟计数器的上溢进行递增计数。</p>

位编号	位符号	说明
		<p>在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。写操作过程中若发生分钟寄存器上溢，将忽视分钟计数寄存器的上溢并且设定为写入值。</p> <p>更改 AMPM 位的值，HOUR 寄存器的值就变为设定的小时系统所对应的值。当 AMPM=0 时，HOURT[1]表示上午或下午：为 0 时表示上午 AM，为 1 时表示下午 PM。</p> <p>若在复位后将 AMPM 位置“1”，HOURT[1:0]的复位值为 00b。</p>
19~16	HOURU[3:0]	<p>小时计数值的个位</p> <p>根据 AMPM 设定的小时系统，以 BCD 格式设置的值表示小时计数值的个位，且通过分钟计数器的上溢进行递增计数。</p> <p>在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。写操作过程中若发生分钟寄存器上溢，将忽视分钟计数寄存器的上溢并且设定为写入值。</p> <p>若在复位后将 AMPM 位置“1”，HOURU[3:0]的复位值为 0000b。</p>
14~12	MINT[2:0]	<p>分钟计数值的十位</p> <p>以 BCD 格式设置的值表示分钟计数值的十位，且通过秒计数器的上溢进行递增计数。</p> <p>在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中若发生秒寄存器上溢，将忽视秒计数寄存器的上溢并且设定为写入值。</p>
11~8	MINU[3:0]	<p>分钟计数值的个位</p> <p>以 BCD 格式设置的值表示分钟计数值的个位，且通过秒计数器的上溢进行递增计数。</p> <p>在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。在写操作过程中若发生秒寄存器上溢，将忽视秒计数寄存器的上溢并且设定为写入值。</p>
7	HSEC	<p>半秒标志位</p> <p>此标志位只读，由硬件置 1 和清 0，用户无法对其进行写操作。</p> <p>HSEC 在一秒的前半秒为 0，后半秒为 1。</p> <p>注意：用户在对日历时间进行读写操作时，推荐在 HSEC 置 1 后再进行读写操作，以避免正好在进位时进行读写。</p>
6~4	SECT[2:0]	<p>秒计数值的十位</p> <p>以 BCD 格式设置的值表示秒计数值的十位，且通过内部计数器（16 位）的上溢进行递增计数。</p> <p>在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。</p> <p>注意：写秒计数值会清零内部计数器（16 位），使该秒重新开始计数</p>
3~0	SECU[3:0]	<p>秒计数值的个位</p> <p>以 BCD 格式设置的值表示秒计数值的个位，且通过内部计数器（16 位）的上溢进行递增计数。</p> <p>在写入时，数据先被写到缓冲器，在经过最多 2 个 f_{RTC} 时钟后被写到计数器。</p> <p>注意：写秒计数值会清零内部计数器（16 位），使该秒重新开始计数</p>
31~22, 15	-	保留

25.7.1.4 RTC 闹钟设置寄存器_RTC_ALARM

寄存器	读/写	说明	复位值	上电初始值
RTC_ALARM	读/写	RTC 闹钟设置寄存器	0x0000_1200	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	WW[6:0]						
15	14	13	12	11	10	9	8
-	WHT[1:0]			WHU[3:0]			
7	6	5	4	3	2	1	0
-	WMT[2:0]			WMU[3:0]			

位编号	位符号	说明																
22~16	WW	<p>闹钟星期设置位 对 WW[6:0]中的任意一位写 1，其代表的星期将会设置闹钟。 WW[6:0]与星期的对应关系如下表：</p> <table border="1"> <thead> <tr> <th>星期</th> <th>WW[6:0]</th> </tr> </thead> <tbody> <tr> <td>星期日</td> <td>WW[0]</td> </tr> <tr> <td>星期一</td> <td>WW[1]</td> </tr> <tr> <td>星期二</td> <td>WW[2]</td> </tr> <tr> <td>星期三</td> <td>WW[3]</td> </tr> <tr> <td>星期四</td> <td>WW[4]</td> </tr> <tr> <td>星期五</td> <td>WW[5]</td> </tr> <tr> <td>星期六</td> <td>WW[6]</td> </tr> </tbody> </table> <p>例如：周一至周五都需要开启闹钟，则将 WW[5:1]都写 1</p>	星期	WW[6:0]	星期日	WW[0]	星期一	WW[1]	星期二	WW[2]	星期三	WW[3]	星期四	WW[4]	星期五	WW[5]	星期六	WW[6]
星期	WW[6:0]																	
星期日	WW[0]																	
星期一	WW[1]																	
星期二	WW[2]																	
星期三	WW[3]																	
星期四	WW[4]																	
星期五	WW[5]																	
星期六	WW[6]																	
13~12	WHT[1:0]	<p>闹钟小时设置位的十位 以 BCD 码设置十进制的值表示闹钟小时的十位。如果设定范围以外的值，就不检测闹钟。 当 AMPM=0 时，WH[1]表示上午或下午：为 0 时表示上午 AM，为 1 时表示下午 PM。 若在复位后将 AMPM 位置“1”，WHT[1:0]的复位值为 00b。</p>																
11~8	WHU[3:0]	<p>闹钟小时设置位的个位 以 BCD 码设置十进制的值表示闹钟小时的个位。如果设定范围以外的值，就不检测闹钟。 若在复位后将 AMPM 位置“1”，WHU[3:0]的复位值为 0000b。</p>																
6~4	WMT[2:0]	<p>闹钟分钟设置位的十位 以 BCD 码设置十进制的值表示分钟的十位。如果设定范围以外的值，就不检测闹钟。</p>																
3~0	WMU[3:0]	<p>闹钟分钟设置位的个位 以 BCD 码设置十进制的值表示分钟的个位。如果设定范围以外的值，就不检测闹钟。</p>																
31~23 15~14, 7	-	保留																

25.7.1.5 RTC 标志寄存器 RTC_STS

寄存器	读/写	说明	复位值	上电初始值
RTC_STS	读/写	RTC 标志寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
1	RTCCTIF	固定周期中断标志位 该位由硬件置 1，通过软件写 1 清 0。 0：没有产生固定周期中断 1：产生固定周期中断
0	WALIF	闹钟检测标志位 该位由硬件置 1，通过软件写 1 清 0。 0：闹钟不一致 1：检测到闹钟一致 在检测到日历时间与闹钟一致并且经过 1 个 RTC 时钟（f _{RTC} ）后由硬件置“1”
31~2	-	保留

25.7.2 RTC 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
RTC 基地址：0x4002_21B0					
RTC_CON	0x00	读/写	RTC 控制寄存器	0x0000_0000	0x0000_0000
RTC_CFG0	0x04	读/写	RTC 配置寄存器 0	0x0001_0001	0x0000_0000
RTC_CFG1	0x08	读/写	RTC 配置寄存器 1	0x0012_0000	0x0000_0000
RTC_ALARM	0x0C	读/写	RTC 闹钟设置寄存器	0x0000_1200	0x0000_0000
RTC_STS	0x14	读/写	RTC 标志寄存器	0x0000_0000	0x0000_0000

26 低功耗计数器（LPC）

26.1 概述

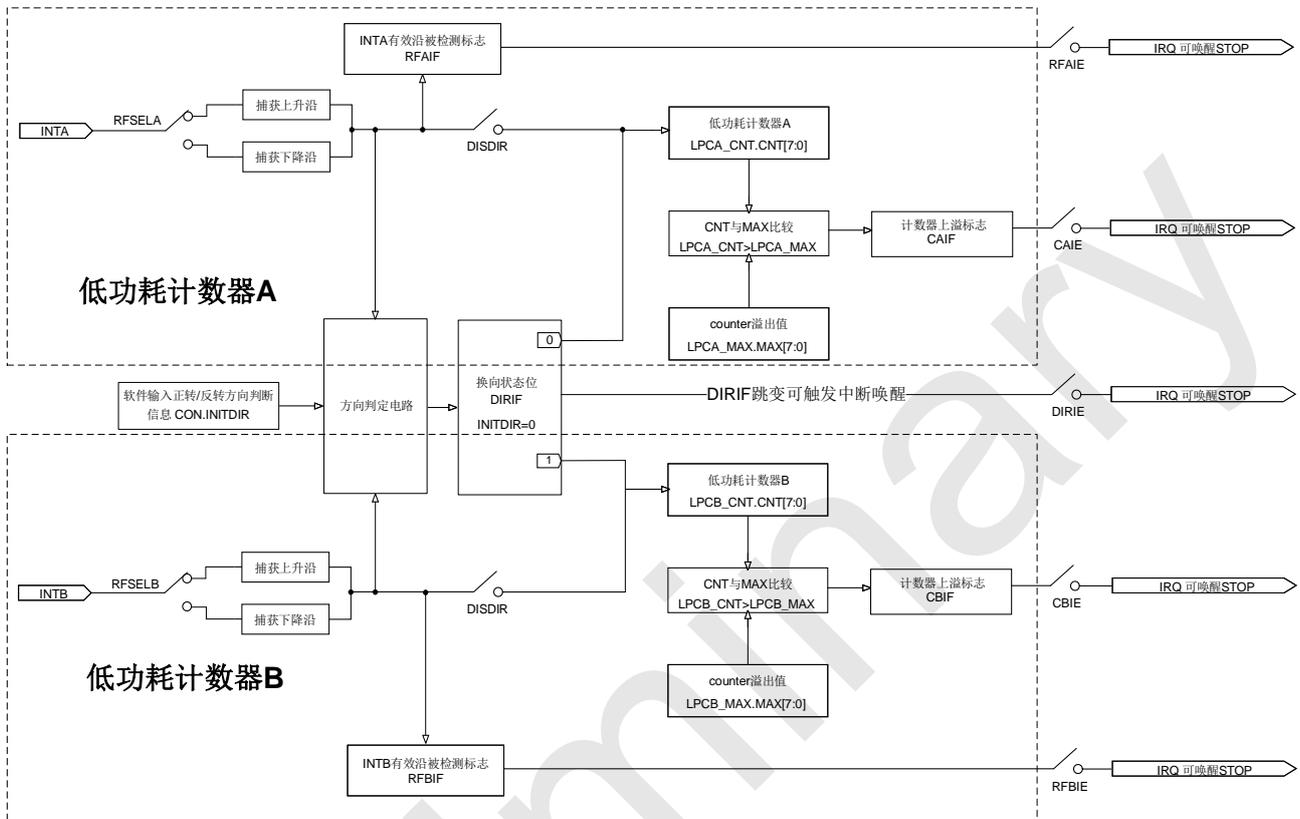
SC32L14T/14G 内部集成一组低功耗计数器（简称 LPC），可与线性/增量编码器等设备连接，用于获取计数、方向等信息。LPC 支持在 STOP 模式下计数，减少 MCU 唤醒频率，降低整体功耗。

LCP 提供两路独立边沿检测通道 INTA 和 INTB，各通道配有独立计数器，可分别设置上升/下降沿有效及计数溢出值；硬件可判断两路输入信号的方向。所有中断事件（包括两路边沿检测中断、两个计数器的溢出中断以及一个方向跳变中断）均可唤醒 STOP 模式。

26.2 特性

- 可以与线性或增量编码器等设备连接，用于获取计数，方向等信息
- 可实现 STOP 模式下计数，减少 MCU 唤醒频率，降低整体功耗
- 提供两个外部信号输入 INTA 和 INTB，可独立检测输入信号的上升/下降沿并独立计数
 - 边沿检测可触发中断
 - 计数值溢出可触发中断
- 可硬件判断输入信号的方向信息
 - 方向跳变可触发中断
- LPC 中断可唤醒 STOP 模式

26.3 LPC 框图



26.4 方向判定

配合外接编码器等外部设备使用 LPC 时，LPC 可以通过两个外接端口 INTA 和 INTB 上检测到有效沿的顺序判断方向信息。

INTA 引脚先发生边沿捕获，INTB 引脚后发生边沿捕获的顺序为“正转”；INTB 引脚先发生边沿捕获，INTA 引脚后发生边沿捕获的顺序为“反转”。当用户通过 INITDIR 设置初始方向后，可通过状态位 DIRIF 观察目前方向是否与初始方向相同，若相同则为 0，若不同则为 1；且若此时总中断控制位 INTEN 和方向改变中断使能位 DIRIE 已使能，DIRIF 改变一次，就会触发一次中断。

另外提供一个方向翻转标志位，无论初始方向如何，只要方向发生翻转，此位都会由硬件置 1，需要用户手动清零。

26.5 计数方式

当 DISDIR=0 时，INTA 和 INTB 端口捕获的有效边沿需通过方向判定电路才能进行有效计数：

- 若 INITDIR=0，初始方向为“正转”：
 - 当前方向为“正转”时，INTA 端口每有一个有效沿被检测到，LPCA_CNT 的计数值加一，从方向从“反转”变为“正转”的第一个有效沿不会使 LPCA_CNT 的计数值加一。

- 当前方向为“反转”时，INTB 端口每有一个有效沿被检测到，LPCB_CNT 的计数值加一，从方向从“正转”变为“反转”的第一个有效沿不会使 LPCB_CNT 的计数值加一。
- 若 INITDIR=0，初始方向为“反转”：
 - 当前方向为“反转”时，INTB 端口每有一个有效沿被检测到，LPCB_CNT 的计数值加一，从方向从“正转”变为“反转”的第一个有效沿不会使 LPCB_CNT 的计数值加一。
 - 当前方向为“正转”时，INTA 端口每有一个有效沿被检测到，LPCA_CNT 的计数值加一，从方向从“反转”变为“正转”的第一个有效沿不会使 LPCA_CNT 的计数值加一。

注意：若在运行过程中因外界因素导致 INTA 或 INVB 端口上无法捕获到有效沿，那么会导致无法判断方向信息，计数值不会增加。

当 DISDIR=1 时，INTA 和 INTB 只要捕获到有效沿，计数值就会增加：

- INTA 端口每有一个有效沿被检测到，LPCA_CNT 的计数值加一。
- INTB 端口每有一个有效沿被检测到，LPCB_CNT 的计数值加一。

26.6 LPC 中断

当 INTA 检测到有效沿输入时，RFAIF 将置起，如果此时 LPC_IDE.INTEN=1 且 LPC_IDE.RFAIE =1，将产生中断。

当 INTB 检测到有效沿输入时，RFBIF 将置起，如果此时 LPC_IDE.INTEN=1 且 LPC_IDE.RFBIE=1，将产生中断。

当 LPCA_CNT 计数值大于 LPCA_MAX 产生溢出时，CAIF 将置起，如果此时 LPC_IDE.INTEN=1 且 LPC_IDE.CAIE=1，将产生中断

当 LPCB_CNT 计数值大于 LPCB_MAX 产生溢出时，CBIF 将置起，如果此时 LPC_IDE.INTEN=1 且 LPC_IDE.CBIE=1，将产生中断

当检测到方向发生改变时，DIRIF 会根据初始方向进行置位，如果此时 LPC_IDE.INTEN=1 且 LPC_IDE.DIRIE=1，DIRIF 每改变一次，就会触发一次中断。

中断事件	中断请求控制位	事件标志位	中断使能子开关
INTA 检测到有效沿输入	LPC_IDE->INTEN	RFAIF	RFAIE
INTB 检测到有效沿输入		RFBIF	RFBIE
LPCA_CNT 计数溢出		CAIF	CAIE
LPCB_CNT 计数溢出		CBIF	CBIE
方向发生改变		DIRIF	DIRIE

26.7 LPC 寄存器

26.7.1 LPC 相关寄存器表

26.7.1.1 LPC 低功耗计数器控制寄存器 LPC_CON

寄存器	读/写	说明	复位值	上电初始值
LPC_CON	读/写	LPC 低功耗计数器控制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7	LPCTEN	LPC 低功耗计数模块使能位 0: 不使能 LPC 低功耗计数模块 1: 使能 LPC 低功耗计数模块
5	RFSELB	INTB 上升沿/下降沿计数选择位 0: 检测到 INTB 引脚上一个上升沿, 产生一个计数 1: 检测到 INTB 引脚上一个下降沿, 产生一个计数
4	RFSELA	INTA 上升沿/下降沿计数选择位 0: 检测到 INTA 引脚上一个上升沿, 产生一个计数 1: 检测到 INTA 引脚上一个下降沿, 产生一个计数
1	DISDIR	方向判断屏蔽位 0: INTA 和 INTB 端口捕获的有效边沿需通过方向判定电路才能进行有效计数: 若方向判定为“正转”, 则计数值按照 LPC_CON.INITDIR 中规则增加, DIRIF 根据自身定义转变为对应状态; 若方向判定为“反转”, 则计数值按照 LPC_CON.INITDIR 中规则增加, DIRIF 根据自身定义转变为对应状态; 1: LPCA 和 LPCB 两组低功耗计数器计数独立: 当 INTA 捕获到有效沿, LPCA_CNT.CNT[7:0]加 1 当 INTB 捕获到有效沿, LPCB_CNT.CNT[7:0]加 1 但此时 DIRIF 方向状态位仍会配合 INITDIR 设定的初始状态改变状态, 并触发相应中断
0	INITDIR	初始方向设置位: 根据 26.4 方向判定 中的判定方法: 0: 设定初始方向为“正转”。 INTA 引脚先发生边沿捕获, INTB 引脚后发生边沿捕获的顺序为“正转”; INTB 引脚先发生边沿捕获, INTA 引脚后发生边沿捕获的顺序为“反转”。 在此设定下正转时, INTA 端口有效边沿计数值计入 LPCA_CNT; 若此时 DISDIR=0, INTB 端口发生边沿捕获时不改变 LPCB_CNT 计数值。

位编号	位符号	说明
		<p>在此设定下反转时，INTB 端口有效边沿计数值计入 LPCB_CNT；若此时 DISDIR=0，INTA 端口发生边沿捕获时不改变 LPCA_CNT 计数值。（方向改变时，检测到的第一个方向改变后的有效边沿不纳入计数）</p> <p>1: 设定初始方向为“反转”。 INTB 引脚先发生边沿捕获，INTA 引脚后发生边沿捕获的顺序为“反转”；INTA 引脚先发生边沿捕获，INTB 引脚后发生边沿捕获的顺序为“正转”。</p> <p>在此设定下反转时，INTB 端口有效边沿计数值计入 LPCB_CNT；若此时 DISDIR=0，INTA 端口发生边沿捕获时不改变 LPCA_CNT 计数值。</p> <p>在此设定下正转时，INTA 端口有效边沿计数值计入 LPCA_CNT；若此时 DISDIR=0，INTB 端口发生边沿捕获时不改变 LPCB_CNT 计数值。（方向改变时，检测到的第一个方向改变后的有效边沿不纳入计数）</p> <p>注意： 当 LPCTEN=0，该位允许写入，提供初始方向； 当 LPCTEN=1，该位用于方向指示，此时用户改写需谨慎。</p>
31~8 6 3~2	-	保留

26.7.1.2 LPC 低功耗计数器标志状态位寄存器 LPC_STS

寄存器	读/写	说明	复位值	上电初始值
LPC_STS	读/写	LPC 低功耗计数器标志状态位寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
5	RFBIF	<p>INTB 输入有效沿被检测到的标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 无外部事件输入 1: 检测到 INTB 有效沿输入，有效沿为 LPC_CON.RFSELB 所选沿</p>
4	RFAIF	<p>INTA 输入有效沿被检测到的标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 无外部事件输入 1: 检测到 INTA 有效沿输入，有效沿为 LPC_CON.RFSELA 所选沿</p>
3	CBIF	<p>LPCB 计数溢出标志位 该位由硬件置 1，通过软件写 1 清 0。 0: 无溢出，LPCB_CNT ≤ LPCB_MAX</p>

位编号	位符号	说明
		1: 溢出, $LPCB_CNT > LPCB_MAX$, 该位由硬件置 1
2	CAIF	LPCA 计数溢出标志位 该位由硬件置 1, 通过软件写 1 清 0。 0: 无溢出, $LPCA_CNT \leq LPCA_MAX$ 1: 溢出, $LPCA_CNT > LPCA_MAX$, 该位由硬件置 1
1	DIRIF	该位为状态位, 只读, 由硬件置 1 及清 0: 0: 当前波形判定方向符合 INITDIR 设置值 1: 当前波形判定方向与 INITDIR 设置值相反 DIRIE 使能后, DIRIF 每跳变一次, 可触发一次中断
0	FLIPIF	方向翻转标志位 每次方向切换的时候, 该位都会置起 该位由硬件置 1, 通过软件写 1 清 0。
31~6	-	保留

26.7.1.3 INTA 有效边沿计数值寄存器 LPCA_CNT

寄存器	读/写	说明	复位值	上电初始值
LPCA_CNT	读/写	INTA 有效边沿计数值寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7~0	CNT[7:0]	INTA 有效边沿计数值 进入 LPCA_CNT 计数溢出中断后自动清零 CNT[7:0] 注意: 当转向发生变化时, 转向后第一轮边沿触发不会改变 CNT[7:0] 的值
31~8	-	保留

26.7.1.4 INTB 有效边沿计数值寄存器 LPCB_CNT

寄存器	读/写	说明	复位值	上电初始值
LPCB_CNT	读/写	INTB 有效边沿计数值寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
CNT[7:0]							

位编号	位符号	说明
7~0	CNT[7:0]	INTB 有效边沿计数值 进入 LPCB_CNT 计数溢出中断后自动清零 CNT[7:0] 注意：当转向发生变化时，转向后第一轮边沿触发不会改变 CNT[7:0] 的值
31~8	-	保留

26.7.1.5 INTA 有效边沿计数溢出值寄存器 LPCA_MAX

寄存器	读/写	说明	复位值	上电初始值
LPCA_MAX	读/写	INTA 有效边沿计数溢出值寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7~0	MAX[7:0]	INTA 有效边沿计数溢出值，当 CNT[7:0] = MAX[7:0]+1，CAIF 置起。 注意：溢出中断开启时，如果 MAX 为 0，有效信号沿无论产生多少次，仅在第一次有效信号沿到来时进入一次溢出中断。
31~8	-	保留

26.7.1.6 INTB 有效边沿计数溢出值寄存器 LPCB_MAX

寄存器	读/写	说明	复位值	上电初始值
LPCB_MAX	读/写	INTB 有效边沿计数溢出值寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
7~0	MAX[7:0]	INTB 有效边沿计数溢出值，当 CNT[7:0] = MAX[7:0]+1，CAIF 置起。 注意：溢出中断开启时，如果 MAX 为 0，有效信号沿无论产生多少次，仅在第一次有效信号沿到来时进入一次溢出中断。
31~8	-	保留

26.7.1.7 LPC 低功耗计数器中断使能寄存器 LPC_IDE

寄存器	读/写	说明	复位值	上电初始值
LPC_IDE	读/写	LPC 低功耗计数器中断使能寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
5	RFBIE	INTB 输入有效沿中断使能位 0: RBIF 置起时不允许产生中断 1: RBIF 置起时产生中断
4	RFAIE	INTA 输入有效沿中断使能位 0: RAIF 置起时不允许产生中断 1: RAIF 置起时产生中断
3	CBIE	LPCB_CNT 计数溢出中断使能位 0: 允许 CBIF 置起且不允许产生 LPCB_CNT 计数溢出中断 1: 允许 CBIF 置起且允许产生 LPCB_CNT 计数溢出中断
2	CAIE	LPCA_CNT 计数溢出中断使能位 0: 允许 CAIF 置起且不允许产生 LPCA_CNT 计数溢出中断 1: 允许 CAIF 置起且允许产生 LPCA_CNT 计数溢出中断
1	DIRIE	方向跳变中断使能位： 0: DIRIF 跳变不允许触发中断 1: DIRIE 使能后，DIRIF 每跳变一次，可触发一次中断
0	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
31~6	-	保留

26.7.2 LPC 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
LPC 基地址: 0x4002_1200					
LPC_CON	0x00	读/写	LPC 低功耗计数器控制寄存器	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
LPC_STS	0x04	读/写	LPC 低功耗计数器标志状态位寄存器	0x0000_0000	0x0000_0000
LPCA_CNT	0x08	读/写	INTA 有效边沿计数值寄存器	0x0000_0000	0x0000_0000
LPCB_CNT	0x0C	读/写	INTB 有效边沿计数值寄存器	0x0000_0000	0x0000_0000
LPCA_MAX	0x10	读/写	INTA 有效边沿计数溢出值寄存器	0x0000_0000	0x0000_0000
LPCB_MAX	0x14	读/写	INTB 有效边沿计数溢出值寄存器	0x0000_0000	0x0000_0000
LPC_IDE	0x18	读/写	LPC 低功耗计数器中断使能寄存器	0x0000_0000	0x0000_0000

Preliminary

27 16 位定时/计数器 (TIM) Timer0~Timer7

27.1 时钟源

- 定时模式/PWM 输出模式下，TIM 时钟源来自 PCLK
- 计数模式下，Tn 引脚为计数源输入

27.2 特性

- 支持 8 档 TIM 时钟预分频
- 8 个独立 16 bit 自动重载计数器 Timer0~Timer7
- 16 位递增、递减、递增/递减自动重载计数器
- 支持上升沿/下降沿捕获，可实现 PWM duty 和周期捕获
- 每个 TIM 提供两路共周期、占空比可调的 PWM (TPWMA / TPWMB) 输出
- TIM1、TIM2、TIM6 的定时器溢出及捕获事件可触发 DMA 请求
- TIM0~TIM7 的 Tn 管脚可以重映射

27.3 计数方式

27.3.1 定时器模式下 TIM 计数方式

- 向上计数：从设定值开始向上计数，至 0xFFFF 溢出
- 向下计数：从 0xFFFF 开始向下计数至设定值

27.3.2 PWM 输出模式下 TIM 计数方式

PWM 输出模式下只能选择向上计数：从 0 开始向上计数，至占空比设置项 PDT 时 PWM 输出波形切换高低电平，之后继续向上计数到设定的重载值 RLD，产生溢出并从 0 重新开始计数。

TIM 输出的 PWM 周期 T_{PWM} 计算公式如下：

$$T_{PWM} = \frac{RLD[15:0] + 1}{PCLK}$$

占空比 duty 计算公式：

$$duty = \frac{PDT[15:0]}{RLD[15:0] + 1}$$

27.4 定时器相关的信号口

- Tn/TnCAP, n=0~7
 - Tn 时钟输入/输出
 - TnCAP 上升沿/下降沿均可捕获

- 注意：Tn 和 TnCAP 为复用功能，不可同时使用
- TnEX, n=0~7
 - 重载模式下，TnEX 引脚上的外部事件输入(下降沿)用作重载允许/禁止控制
 - 捕获模式，当 FSEL = 1 时为下降沿捕获信号输入脚，检测到 TnEX 引脚上一个下降沿，产生一个捕获，EXIF 被置起，TnCNT 寄存器的值捕获到寄存器 FCAP 里
- TnPWM, n=0~7
 - Timer0~7 可通过各自的 Tn 端口提供 duty 可单独调的 PWM: TnPWMA
 - Timer0~7 可通过各自的 TnEX 端口提供 duty 可单独调的 PWM: TnPWMB
 - TnPWMA 和 TnPWMB 共周期，时钟源随 TIM

注意：TIM 的 PWM 捕获功能与 PWM 输出功能不可同时开启

27.5 TIM 的中断及对应标志位

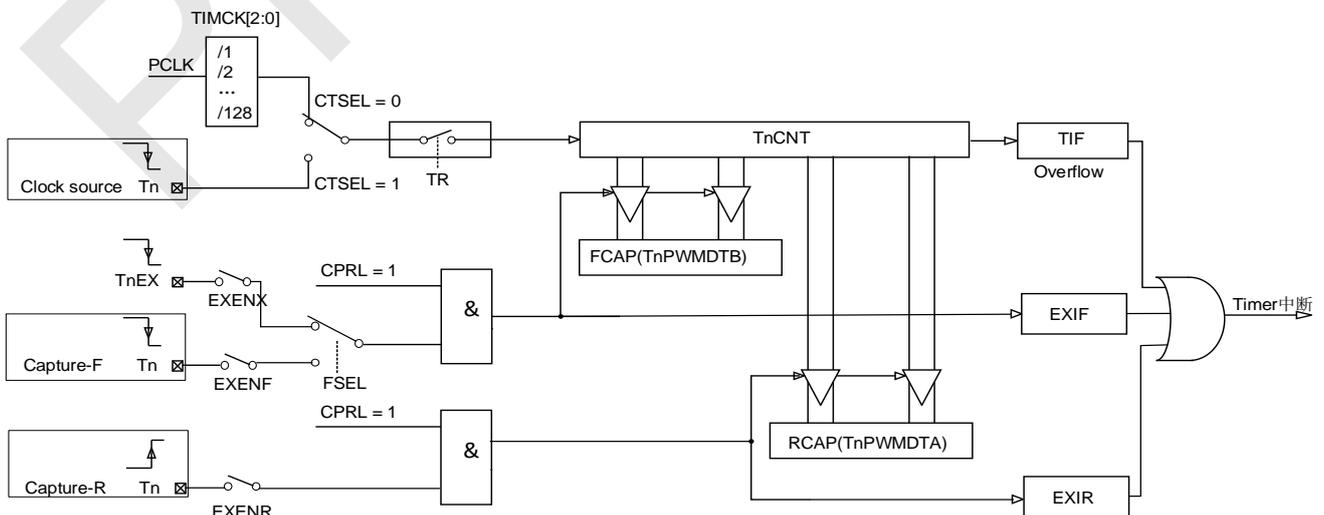
- 计数器上溢/下溢——共用中断标志位 TIF
- 捕获状态标志：
 - EXIF 外部事件输入下降沿被检测到的标志位
 - EXIR 外部事件输入上升沿被检测到的标志位
- 中断及优先级配置控制位合并至 NVIC 模块

27.6 TIM 的工作模式

- 模式 0：16 位捕获模式，可实现 PWM 双沿捕获
- 模式 1：16 位自动重载定时/计数器模式
- 模式 3：可编程时钟输出模式
- 模式 4：PWM 输出模式

27.6.1 工作模式 0：16 位捕获

27.6.1.1 双沿捕获结构图



模式 0：16 位捕获

1.1.1.1 PWM 双沿捕获模式

Tn 和 TnEX 都可作为 PWM 捕获口，可通过 FSEL 寄存器位进行选择，分别采样上升沿和下降沿：

- 专门一个上升沿捕获计数寄存器 16bit——与 PWMA 占空比设定寄存器 TIMn_PDTA 复用
- 专门一个下降沿捕获计数寄存器 16bit——与 PWMB 占空比设定寄存器 TIMn_PDTB 复用
- 捕获标志位：
 - 单独的上/下沿捕获使能位
 - 单独的上/下捕获标志

27.6.2 工作模式 1：16 位自动重载定时器/计数器模式

在 16 位自动重载方式下，定时器可以被选为递增计数或递减计数。这个功能通过将 TIMn_CON(n=0~7)中的 DEC 置 1 后，通过 TnEX 选择计数方向。系统复位后，DEC 位复位值为 0，定时器 n 默认递增计数；当 DEC 置 1 时，定时器 n 递增计数或递减计数取决于 TnEX 引脚上的电平。

当 DEC=0，通过在 TIMn_CON 中的 EXENX 位选择两个选项：

若 EXENX = 0，当 TIMn_CNT 递增到 0xFFFF 时，定时器溢出位 TIF 置起，同时定时器自动将用户软件写好的重载寄存器 TIMn_RLD 中的 16 位值装入 TIMn_CNT 寄存器。

若 EXENX = 1，溢出或在外部输入 TnEX 上的下降沿都能触发一个 16 位重载。TnEX 上有下降沿产生时，EXIF 位置起。如果 TIE 被使能，TIF 和 EXIF 位都能产生一个中断。

当 DCEN = 1 时，TnEX 引脚控制计数的方向，而 EXENX 控制无效。

若 TnEX = 1，则 TIMn 递增计数。当 TIMn_CNT 递增到 0xFFFF 时，定时器溢出位 TIF 置起，同时定时器自动将用户软件写好的重载寄存器 TIMn_RLD 中的 16 位值装入 TIMn_CNT 寄存器。

若 TnEX = 0，则 TIMn 递减计数。当 TIMn_CNT 的值从 0xFFFF 递减至等于 TIMn_RLD 的值时，定时器溢出，且定时器溢出位 TIF 置起，同时 0xFFFF 重载入 TIMn_CNT。

在此工作方式下，无论 TIMn 溢出与否，EXFIF 不作为中断标志。

27.6.3 工作模式 3：可编程时钟输出

在这种方式中，TIMn(n=0~7)可以编程为输出 50%占空比时钟周期：当 CTSEL=0，TXOE=1 时，使能 TIMn 作为时钟发生器。

在这种方式下 Tn 输出的时钟频率为：

$$f_{OUT} = \frac{f_{TIM}}{(65536 - TIMn_{RLD}) * 4}$$

27.6.4 工作模式 4：PWM 输出模式

- PWM 占空比变化特性

更改 PDTx[15:0]的值，占空比不会立即改变，而是等待本周期结束，在下个周期改变。

- PWM 周期变化特性

通过改变周期设置寄存器[RCAPXL / RCAPXH]的值实现。定义当前周期计数值为 Tn，写入周期寄存器时，定时器计到的值为 Tm，待更新的周期计数值为 Tx，则：

- Tm ≤ Tx：周期按照 Tx 实时改变；

- $T_m > T_x$: 此时周期变化会分为两个阶段。第一个阶段，写入周期寄存器之后，周期计数器会从当前计数值累加至溢出清零。第二个阶段，周期按照 T_x 改变。

27.7 TIM 中断

定时或计数模式下，CNT 计数达到 TIMn 计数值，TIF 将置起，如果 TIMn_IDE.INTEN=1，将产生中断。

外部事件输入模式下，检测到有效跳变沿，EXIR/EXIF 将置起，如果 TIMn_IDE.INTEN=1，将产生中断。

中断事件	事件标志位	中断使能总控制位	模块中断使能子控制位
定时器溢出	TIF	TIMn_IDE->INTEN (n=0~7)	TIMn_IDE->TIE
外部事件输入上升沿中断	EXIR		TIMn_IDE->EXRIE
外部事件输入下降沿中断	EXIF		TIMn_IDE->EXFIE

27.8 TIM 寄存器

27.8.1 TIM 相关寄存器表

27.8.1.1 定时器控制寄存器 TIMn_CON

寄存器	读/写	说明	复位值	上电初始值
TIMn_CON (n=0~7)	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	SPOS	-	-	-	-	-
15	14	13	12	11	10	9	8
TXOE	EPWMNA	EPWMNB	INVNA	INVNB	TIMCK[2:0]		
7	6	5	4	3	2	1	0
TR	DEC	EXENX	FSEL	EXENF	EXENR	CTSEL	CPRL

位编号	位符号	说明												
21	SPOS	● TIM0 信号口映射控制位 @TIM0_CON												
		<table border="1"> <thead> <tr> <th>信号</th> <th>T0CAP</th> <th>T0EX</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS=0</td> <td>PB11</td> <td>PB10</td> </tr> <tr> <td>SPOS=1</td> <td>PE6</td> <td>\</td> </tr> </tbody> </table>	信号	T0CAP	T0EX	SPOS 值			SPOS=0	PB11	PB10	SPOS=1	PE6	\
		信号	T0CAP	T0EX										
		SPOS 值												
		SPOS=0	PB11	PB10										
		SPOS=1	PE6	\										
		● TIM1 信号口映射控制位 @TIM1_CON												
		<table border="1"> <thead> <tr> <th>信号</th> <th>T1CAP</th> <th>T1EX</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS=0</td> <td>PA11</td> <td>PC6</td> </tr> <tr> <td>SPOS=1</td> <td>PA1</td> <td>\</td> </tr> </tbody> </table>	信号	T1CAP	T1EX	SPOS 值			SPOS=0	PA11	PC6	SPOS=1	PA1	\
		信号	T1CAP	T1EX										
		SPOS 值												
SPOS=0	PA11	PC6												
SPOS=1	PA1	\												
● TIM2 信号口映射控制位 @TIM2_CON														

位编号	位符号	说明																																																																								
		<table border="1"> <thead> <tr> <th>信号</th> <th>T2CAP</th> <th>T2EX</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS=0</td> <td>PB9</td> <td>PB8</td> </tr> <tr> <td>SPOS=1</td> <td>PA0</td> <td>\</td> </tr> </tbody> </table> <p>● TIM3 信号口映射控制位 @TIM3_CON</p> <table border="1"> <thead> <tr> <th>信号</th> <th>T3CAP</th> <th>T3EX</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS=0</td> <td>PA13</td> <td>PD11</td> </tr> <tr> <td>SPOS=1</td> <td>PA2</td> <td>\</td> </tr> </tbody> </table> <p>● TIM4 信号口映射控制位 @TIM4_CON</p> <table border="1"> <thead> <tr> <th>信号</th> <th>T4CAP</th> <th>T4EX</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS=0</td> <td>PD1</td> <td>PD10</td> </tr> <tr> <td>SPOS=1</td> <td>PA3</td> <td>\</td> </tr> </tbody> </table> <p>● TIM5 信号口映射控制位 @TIM5_CON</p> <table border="1"> <thead> <tr> <th>信号</th> <th>T5CAP</th> <th>T5EX</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS=0</td> <td>PE4</td> <td>PC11</td> </tr> <tr> <td>SPOS=1</td> <td>PC12</td> <td>\</td> </tr> </tbody> </table> <p>● TIM6 信号口映射控制位 @TIM6_CON</p> <table border="1"> <thead> <tr> <th>信号</th> <th>T6CAP</th> <th>T6EX</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS=0</td> <td>PB7</td> <td>PB6</td> </tr> <tr> <td>SPOS=1</td> <td>PA7</td> <td>\</td> </tr> </tbody> </table> <p>● TIM7 信号口映射控制位 @TIM7_CON</p> <table border="1"> <thead> <tr> <th>信号</th> <th>T7CAP</th> <th>T7EX</th> </tr> </thead> <tbody> <tr> <td>SPOS 值</td> <td></td> <td></td> </tr> <tr> <td>SPOS=0</td> <td>PD0</td> <td>PD14</td> </tr> <tr> <td>SPOS=1</td> <td>PE5</td> <td>\</td> </tr> </tbody> </table>	信号	T2CAP	T2EX	SPOS 值			SPOS=0	PB9	PB8	SPOS=1	PA0	\	信号	T3CAP	T3EX	SPOS 值			SPOS=0	PA13	PD11	SPOS=1	PA2	\	信号	T4CAP	T4EX	SPOS 值			SPOS=0	PD1	PD10	SPOS=1	PA3	\	信号	T5CAP	T5EX	SPOS 值			SPOS=0	PE4	PC11	SPOS=1	PC12	\	信号	T6CAP	T6EX	SPOS 值			SPOS=0	PB7	PB6	SPOS=1	PA7	\	信号	T7CAP	T7EX	SPOS 值			SPOS=0	PD0	PD14	SPOS=1	PE5	\
信号	T2CAP	T2EX																																																																								
SPOS 值																																																																										
SPOS=0	PB9	PB8																																																																								
SPOS=1	PA0	\																																																																								
信号	T3CAP	T3EX																																																																								
SPOS 值																																																																										
SPOS=0	PA13	PD11																																																																								
SPOS=1	PA2	\																																																																								
信号	T4CAP	T4EX																																																																								
SPOS 值																																																																										
SPOS=0	PD1	PD10																																																																								
SPOS=1	PA3	\																																																																								
信号	T5CAP	T5EX																																																																								
SPOS 值																																																																										
SPOS=0	PE4	PC11																																																																								
SPOS=1	PC12	\																																																																								
信号	T6CAP	T6EX																																																																								
SPOS 值																																																																										
SPOS=0	PB7	PB6																																																																								
SPOS=1	PA7	\																																																																								
信号	T7CAP	T7EX																																																																								
SPOS 值																																																																										
SPOS=0	PD0	PD14																																																																								
SPOS=1	PE5	\																																																																								
15	TXOE	Tn 引脚信号方向控制位 0: Tn 作为时钟输入或 I/O 端口 1: Tn 用于可编程时钟输出																																																																								
14	EPWMNA	Tn_PWMA 所在管脚 PWM 波形输出使能位 0: 关闭 1: 使能																																																																								
13	EPWMNB	TnEX_PWMB 所在管脚 PWM 波形输出使能位 0: 关闭 1: 使能																																																																								
12	INVNA	TPWMnA 波形输出反向控制位 0: 正常 1: 波形输出反向																																																																								
11	INVNB	TPWMnB 波形输出反向控制位 0: 正常 1: 波形输出反向																																																																								
10~8	TIMCK[2:0]	TIM 时钟频率档位控制位 用于设定 TIM 时钟频率 f_{TIM} 为: 000: $f_{SOURCE}/1$ 001: $f_{SOURCE}/2$ 010: $f_{SOURCE}/4$																																																																								

位编号	位符号	说明
		011: $f_{SOURCE}/8$ 100: $f_{SOURCE}/16$ 101: $f_{SOURCE}/32$ 110: $f_{SOURCE}/64$ 111: $f_{SOURCE}/128$ f_{SOURCE} 对应的时钟可为 PCLK 或 Tn 输入
7	TR	TIMn 开始/停止控制位, n=0~7 0: 停止 TIMn / TPWMn 计数器 1: 开始 TIMn / TPWMn 计数器
6	DEC	递增/递减方向控制位 0: TIMn 为递增的定时/计数器 1: TIMn 作为递增/递减的定时/计数器, TnEX 用来选择计数方向
5	EXENX	TnEX 设置位, n=0~7 该位在不同模式下作用不同: <ul style="list-style-type: none"> ● 重载模式: (CPRL = 0) 该位用于控制 TnEX 引脚上的外部事件输入(下降沿), 用作重载允许/禁止控制: <ul style="list-style-type: none"> 0: 忽略 TnEX 引脚上的事件 1: 检测到 TnEX 引脚上一个下降沿, 产生一个重载 ● 捕获模式: (CPRL = 1) 该位用作 TnEX 下降沿信号捕获选通位: <ul style="list-style-type: none"> 0: 忽略 TnEX 引脚上的事件 1: 当 FSEL = 1, 检测到 TnEX 引脚上一个下降沿, 产生一个捕获, EXIF 被置起, TnCNT 寄存器的值捕获到寄存器 FCAP 里
4	FSEL	下降沿信号选择位 该位仅在捕获模式 (CPRL=1)下有效 0: 检测到 Tn 引脚上一个下降沿, 产生一个捕获。忽略 TnEX 引脚上的事件 1: 检测到 TnEX 引脚上一个下降沿, 产生一个捕获。忽略 Tn 引脚上的事件
3	EXENF	下降沿信号捕获使能位: <ul style="list-style-type: none"> 0: 忽略 Tn 引脚上的事件 1: 检测到 Tn 引脚上一个下降沿, 产生一个捕获, EXIF 被置起, TnCNT 寄存器的值捕获到寄存器 FCAP 里
2	EXENR	Tn 引脚上的上升沿信号捕获使能位 <ul style="list-style-type: none"> 0: 忽略 Tn 引脚上的事件 1: 检测到 Tn 引脚上一个上升沿, 产生一个捕获, EXIR 被置起, TnCNT 寄存器的值捕获到寄存器 RCAP 里
1	CTSEL	定时器/计数器选定位 0: 定时器方式 1: 计数器方式
0	CPRL	捕获/重载方式设置位 0: 重载功能 1: 捕获功能
31~22 20~16	-	保留

27.8.1.2 定时器计数值寄存器 TIMn_CNT

寄存器	读/写	说明	复位值	上电初始值
TIMn_CNT (n=0~7)	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	CNT[15:0]	TIMn 计数值
31~16	-	保留

27.8.1.3 定时器重载寄存器 TIMn_RLD

寄存器	读/写	说明	复位值	上电初始值
TIMn_RLD (n=0~7)	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	RLD[15:0]	定时器溢出或在外部输入 TnEX 上的下降沿都能触发一个 16 位重载。产生重载时，定时器自动将用户软件写好的 RLD[15:0]值装入 TnCNT 寄存器。
31~16	-	保留

27.8.1.4 定时器标志位寄存器 TIMn_STS

寄存器	读/写	说明	复位值	上电初始值
TIMn_STS (n=0~7)	读/写	定时器标志制寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
2	EXIF	外部事件输入下降沿被检测到的标志位 该位由硬件置 1，通过软件写 1 清 0。 0：无外部事件输入 1：检测到外部输入(如果 EXENF = 1，由硬件设 1) 注意：捕获模式下，该位软件清 0 之前不允许更新 TnFCAP 值
1	EXIR	Tn 引脚外部事件输入上升沿被检测到的标志位 该位由硬件置 1，通过软件写 1 清 0。 0：无外部事件输入 1：检测到外部输入(如果 EXENR = 1，由硬件设 1) 注意：捕获模式下，该位软件清 0 之前不允许更新 TnRCAP 值
0	TIF	定时器溢出标志位 该位由硬件置 1，通过软件写 1 清 0。 0：无溢出(必须由软件清 0) 1：溢出(如果 RCLK = 0 和 TCLK = 0，由硬件设 1)
31~3	-	保留

27.8.1.5 TnPWMA 占空比设定寄存器 TIMn_PDTA (@CPRL = 0)

寄存器	读/写	说明	复位值	上电初始值
TIMn_PDTA (n=0~7)	读/写	TnPWMA 占空比设定寄存器 TIMn_PDTA (@CPRL = 0)	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	PDT[15:0]	TPWMnA 占空比寄存器，n=0~3 TPWMnA 的波形的高电平宽度是 PDT[15:0]个 TIM 时钟
31~16	-	保留

27.8.1.6 TnPWMB 占空比设定寄存器 TIMn_PDTB (@CPRL = 0)

寄存器	读/写	说明	复位值	上电初始值
TIMn_PDTB (n=0~7)	读/写	TnPWMB 占空比设定寄存器 TIMn_PDTB (@CPRL = 0)	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	PDT[15:0]	TPWMnB 占空比寄存器, n=0~3 TPWMnB 的波形的高电平宽度是 PDT[15:0]个 TIM 时钟
31~16	-	保留

27.8.1.7 上升沿数据捕获寄存器 TIMn_RCAP (@CPRL = 1)

寄存器	读/写	说明	复位值	上电初始值
TIMn_RCAP (n=0~7)	读/写	上升沿数据捕获寄存器 TIMn_RCAP (@CPRL = 1)	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	RCAP [15:0]	TIMn 的 PWM 捕获模式下, 当上升沿捕获条件发生, CNT 计数器值将被保存到该寄存器。
31~16	-	保留

27.8.1.8 下降沿数据捕获寄存器 TIMn_FCAP (@CPRL = 1)

寄存器	读/写	说明	复位值	上电初始值
TIMn_FCAP (n=0~7)	读/写	下降沿数据捕获寄存器 TIMn_FCAP (@CPRL = 1)	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
15~0	FCAP[15:0]	TIMn 的 PWM 捕获模式下，当下降沿捕获条件发生，CNT 计数器值将被保存到该寄存器。
31~16	-	保留

27.8.1.9 TIMn 的中断使能及 DMA 控制寄存器 TIMn_IDE

寄存器	读/写	说明	复位值	上电初始值
TIMn_IDE (n=0~7)	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	CAPFDE	CAPRDE	TIDE	EXFIE	EXRIE	TIE	INTEN

位编号	位符号	说明
6	CAPFDE	下降沿捕获事件触发 DMA 请求使能位 0: 下降沿捕获事件禁止产生 DMA 请求 1: 产生新的下降沿捕获时，触发 DMA 请求，DMA 搬运 FCAP 寄存器的值
5	CAPRDE	上升沿捕获事件触发 DMA 请求使能位 0: 上升沿捕获事件禁止产生 DMA 请求 1: 产生新的上升沿捕获时，触发 DMA 请求，DMA 搬运 RCAP 寄存器的值
4	TIDE	定时器溢出事件触发 DMA 请求使能位 0: 定时器溢出禁止产生 DMA 请求 1: 定时器溢出允许产生 DMA 请求
3	EXFIE	外部事件输入下降沿中断开关 0: 禁止下降沿中断 1: 使能下降沿中断
2	EXRIE	外部事件输入上升沿中断开关 0: 禁止上升沿中断 1: 使能上升沿中断

位编号	位符号	说明
1	TIE	定时器溢出中断开关 0: 禁止溢出中断 1: 使能溢出中断
0	INTEN	中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求
31~7	-	保留

27.8.2 TIM 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
TIM0 基地址: 0x4002_0100					
TIM0_CON	0x00	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000
TIM0_CNT	0x04	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000
TIM0_RLD	0x08	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000
TIM0_STS	0x0C	读/写	定时器标志位寄存器	0x0000_0000	0x0000_0000
TIM0_PDTA	0x10	读/写	TnPWMA 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM0_RCAP	0x10	读/写	上升沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM0_PDTB	0x14	读/写	TnPWMB 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM0_FCAP	0x14	读/写	下降沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM0_IDE	0x18	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000
TIM1 基地址: 0x4002_0140					
TIM1_CON	0x00	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000
TIM1_CNT	0x04	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000
TIM1_RLD	0x08	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000
TIM1_STS	0x0C	读/写	定时标志位寄存器	0x0000_0000	0x0000_0000
TIM1_PDTA	0x10	读/写	TnPWMA 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM1_RCAP	0x10	读/写	上升沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM1_PDTB	0x14	读/写	TnPWMB 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM1_FCAP	0x14	读/写	下降沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM1_IDE	0x18	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000
TIM2 基地址: 0x4002_0180					

寄存器	偏移地址	读/写	说明	复位值	上电初始值
TIM2_CON	0x00	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000
TIM2_CNT	0x04	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000
TIM2_RLD	0x08	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000
TIM2_STS	0x0C	读/写	定时标志位寄存器	0x0000_0000	0x0000_0000
TIM2_PDTA	0x10	读/写	TnPWMA 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM2_RCAP	0x10	读/写	上升沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM2_PDTB	0x14	读/写	TnPWMB 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM2_FCAP	0x14	读/写	下降沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM2_IDE	0x18	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000
TIM3 基地址: 0x4002_01C0					
TIM3_CON	0x00	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000
TIM3_CNT	0x04	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000
TIM3_RLD	0x08	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000
TIM3_STS	0x0C	读/写	定时标志位寄存器	0x0000_0000	0x0000_0000
TIM3_PDTA	0x10	读/写	TnPWMA 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM3_RCAP	0x10	读/写	上升沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM3_PDTB	0x14	读/写	TnPWMB 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM3_FCAP	0x14	读/写	下降沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM3_IDE	0x18	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000
TIM4 基地址: 0x4002_1100					
TIM4_CON	0x00	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000
TIM4_CNT	0x04	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000
TIM4_RLD	0x08	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000
TIM4_STS	0x0C	读/写	定时标志位寄存器	0x0000_0000	0x0000_0000
TIM4_PDTA	0x10	读/写	TnPWMA 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM4_RCAP	0x10	读/写	上升沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000

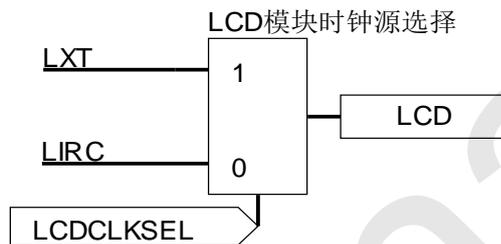
寄存器	偏移地址	读/写	说明	复位值	上电初始值
TIM4_PDTB	0x14	读/写	TnPWMB 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM4_FCAP	0x14	读/写	下降沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM4_IDE	0x18	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000
TIM5 基地址: 0x4002_1140					
TIM5_CON	0x00	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000
TIM5_CNT	0x04	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000
TIM5_RLD	0x08	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000
TIM5_STS	0x0C	读/写	定时标志位寄存器	0x0000_0000	0x0000_0000
TIM5_PDTA	0x10	读/写	TnPWMA 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM5_RCAP	0x10	读/写	上升沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM5_PDTB	0x14	读/写	TnPWMB 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM5_FCAP	0x14	读/写	下降沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM5_IDE	0x18	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000
TIM6 基地址: 0x4002_1180					
TIM6_CON	0x00	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000
TIM6_CNT	0x04	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000
TIM6_RLD	0x08	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000
TIM6_STS	0x0C	读/写	定时标志位寄存器	0x0000_0000	0x0000_0000
TIM6_PDTA	0x10	读/写	TnPWMA 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM6_RCAP	0x10	读/写	上升沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM6_PDTB	0x14	读/写	TnPWMB 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM6_FCAP	0x14	读/写	下降沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM6_IDE	0x18	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000
TIM7 基地址: 0x4002_11C0					
TIM7_CON	0x00	读/写	定时器控制寄存器	0x0000_0000	0x0000_0000
TIM7_CNT	0x04	读/写	定时器计数值寄存器	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
TIM7_RLD	0x08	读/写	定时器重载寄存器	0x0000_0000	0x0000_0000
TIM7_STS	0x0C	读/写	定时标志位寄存器	0x0000_0000	0x0000_0000
TIM7_PDTA	0x10	读/写	TnPWMA 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM7_RCAP	0x10	读/写	上升沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM7_PDTB	0x14	读/写	TnPWMB 占空比设定寄存器 (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM7_FCAP	0x14	读/写	下降沿数据捕获寄存器 (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM7_IDE	0x18	读/写	TIMn 的中断使能及 DMA 控制寄存器	0x0000_0000	0x0000_0000

28 LCD 驱动器

28.1 时钟源

- LXT 和 LIRC 可选
- 用户可以通过 LCDCLKSEL 选择位，选择 LCD 时钟源



28.2 内置 8 COM x 51 SEG LCD 驱动

- 支持在 STOP Mode 下显示
- 两种 LCD 驱动模式可选
 - 电阻型 LCD 驱动：电阻型 LCD 驱动支持快速充电模式，LCD 电压输出口分压电阻可选：11KΩ、100KΩ、300KΩ、800KΩ
 - 电容型 LCD 驱动：电容型 LCD 驱动为电容偏压模式，该模式下 LCD 电路总功耗可低至：2~3μA @STOP 模式
- Type A / Type B 波形可选
- 8 X 51、6 X 53、5 X 54、或 4 X 55 段 LCD 驱动
- LCD 显示驱动偏置电压
 - 1/4 偏置电压
 - 1/3 偏置电压
- 帧频三档可选：
 - Type A 模式下 32/64/128Hz
 - Type B 模式下 64/128/256Hz

28.2.1 电阻型 LCD 驱动器

当 DDR_CON.LCDSEL=0 时，LCD 为电阻型驱动。

电阻型 LCD 驱动的电压可通过 DDR_CFG.VLCD[3:0] 进行调节：

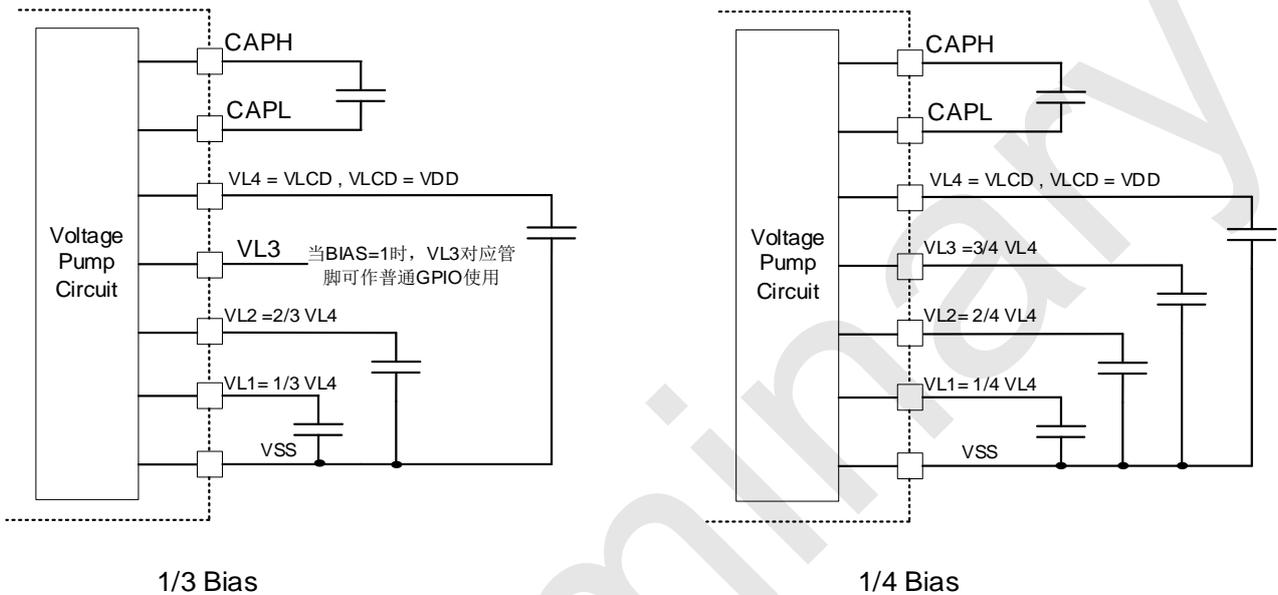
$$V_{LCD} = V_{DD} * \frac{17 + VLCD[3:0]}{32}$$

28.2.2 电容偏压型 LCD 驱动器

当 DDR_CON.LCDSEL=1 时，LCD 为电容偏压型驱动。

选择电容偏压型驱动时，需同时将 DDR_CON.PUMPON 写 1，开启 PUMP 电路，电容偏压型 LCD 才有效。

电容偏压型 LCD 驱动电压 V_{LCD} 等于 V_{DD} 。

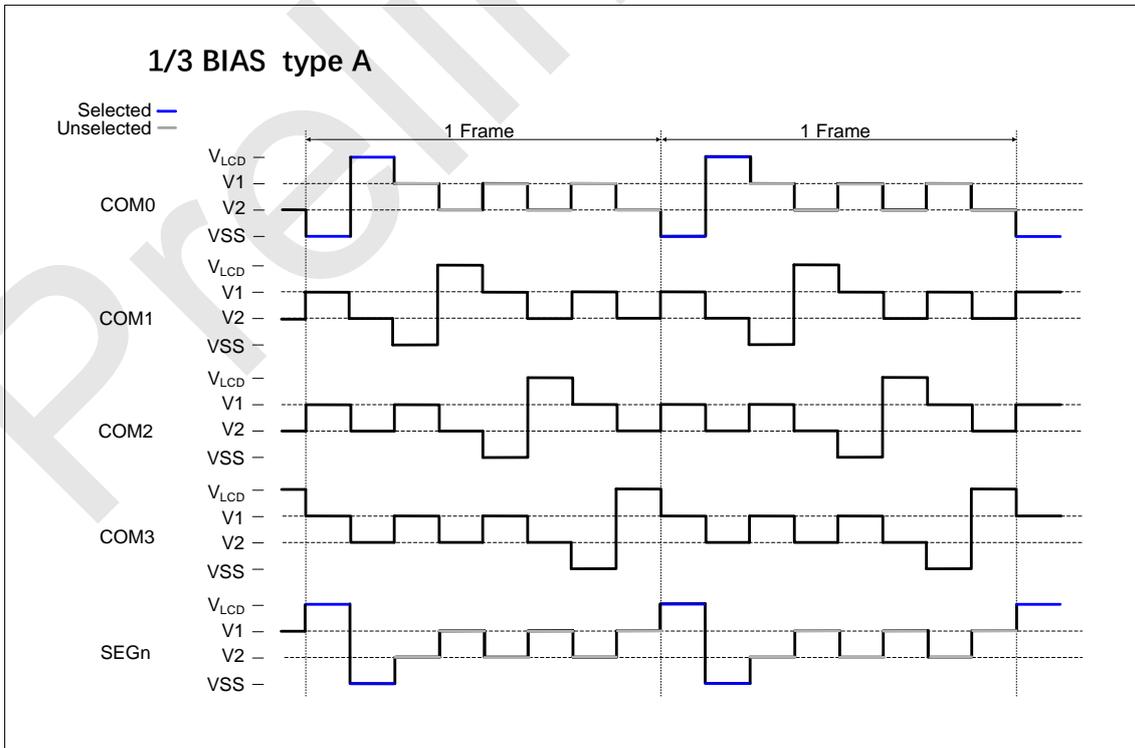
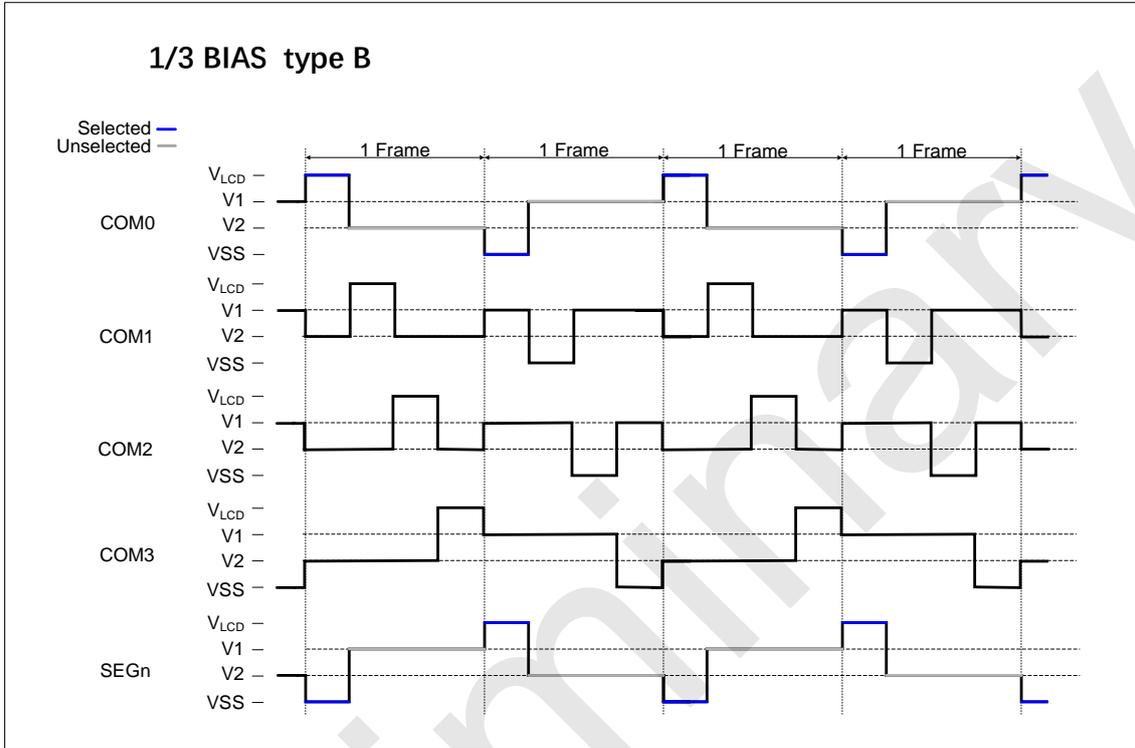


注意：

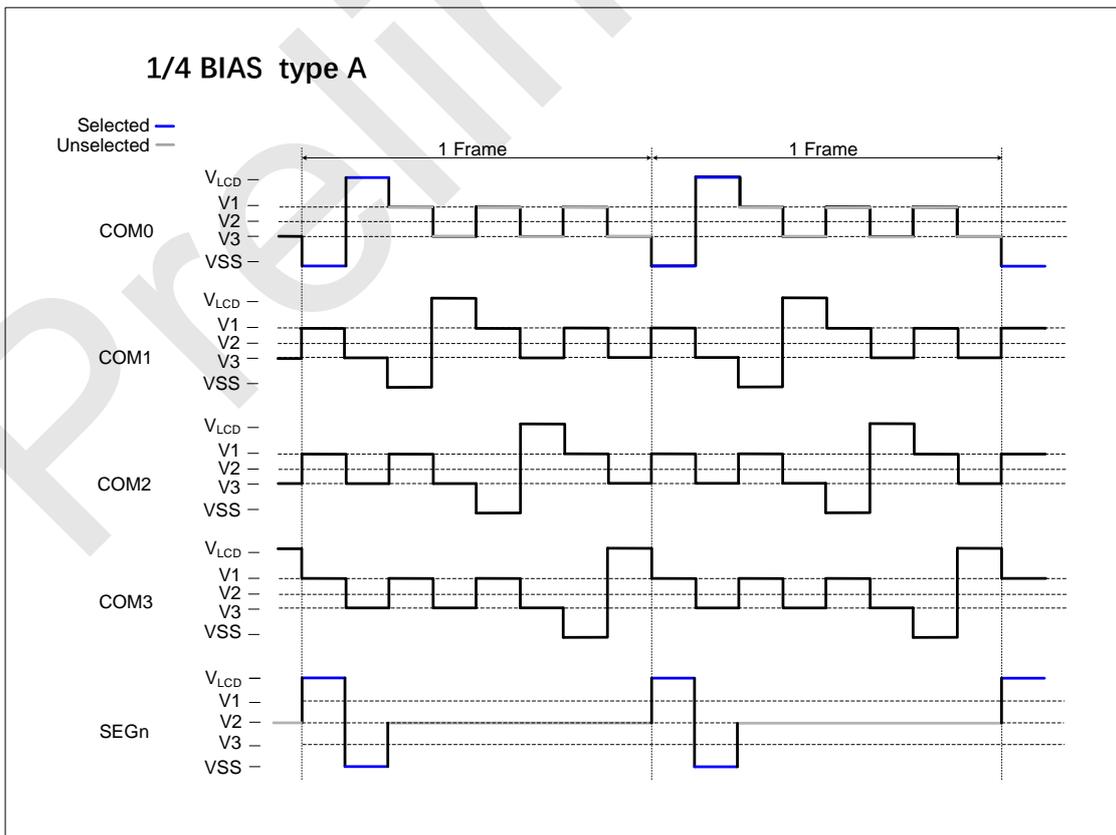
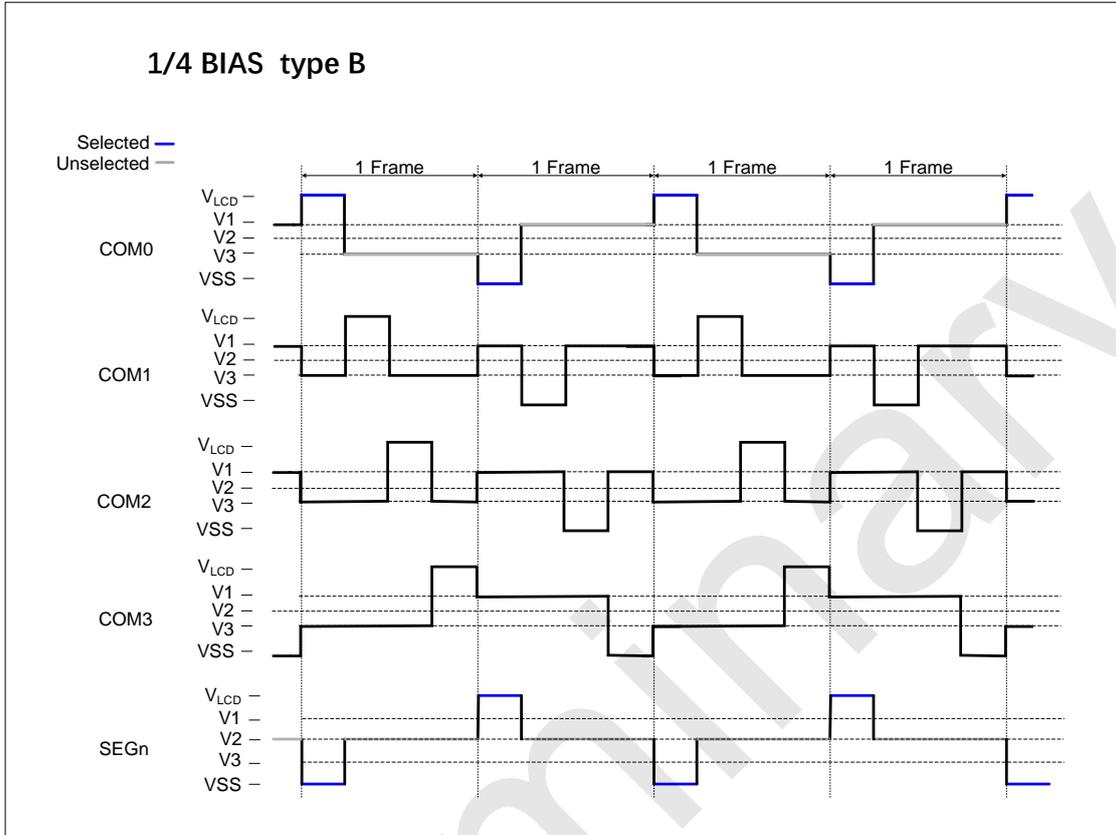
1. 选择电容偏压型 LCD 驱动时，需在 CAPH 与 CAPL 之间和作为 LCD 驱动电源引脚的 VL_n ($n=1\sim4$) 对地接容值为 $0.47\mu F \pm 30\%$ ，漏电流尽量小的电容
2. 若 DDR_CON.BIAS=1，设置 1/3 偏置电压，那么 VL3 对应管脚可作普通 GPIO 使用

28.3 LCD 驱动波形

28.3.1 1/3 BIAS 1/4 duty



28.3.2 1/4 BIAS 1/4 duty



28.4 LCD 寄存器

28.4.1 LCD 相关寄存器表

28.4.1.1 显示驱动控制寄存器 DDR_CON

寄存器	读/写	说明	复位值	上电初始值
DDR_CON	读/写	显示驱动控制寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
TRIMODE	TRICOM	PUMPON	-	LCDSEL	PUMPCK	DDRCK[1:0]	
7	6	5	4	3	2	1	0
DDRON	-	-	TPYE	VOIRSIF	-	BIAS	-

位编号	位符号	说明
15	TRIMODE	<p>自定义帧频模式控制位</p> <p>0: 关闭自定义帧频模式</p> <p>1: 打开自定义帧频模式</p> <p>注意: 自定义帧频模式需配合 TIM 等外设中断来控制帧频; 当选择 LXT 作为时钟源并启用自定义帧频模式后, 在对 TRICOM 写 1 切换 COM 扫描口前需要确保晶振完全起振。</p>
14	TRICOM	<p>扫描 COM 口切换控制位</p> <p>当 TRIMODE=1, 开启自定义帧频模式后, 对此位每写一次“1”, 会切换一次起始扫描 COM 口。</p> <p>以 1/8 占空比为例:</p> <ul style="list-style-type: none"> 第 1 次对 TRICOM 写 1, 将从 COM0 开始扫描, 并持续扫描 COM0, 第 2 次写 1, 将切换至持续扫描 COM1.....第 8 次写 1, 将切换至持续扫描 COM7, 到此一个扫描周期结束, 当第 9 次写 1 时会再从 COM0 开始并持续扫描 COM0。 <p>以下为不同占空比配置下的一个扫描周期:</p> <ul style="list-style-type: none"> 1/8 占空比: 从 COM0 开始扫描, 到 COM7 为一个扫描周期; 1/6 占空比: 从 COM2 开始扫描, 到 COM7 为一个扫描周期; 1/5 占空比: 从 COM3 开始扫描, 到 COM7 为一个扫描周期; 1/4 占空比@SCS=0: 从 COM4 开始扫描, 到 COM7 为一个扫描周期; 1/4 占空比@SCS=1: 从 COM0 开始扫描, 到 COM3 为一个扫描周期。
13	PUMPON	<p>PUMP 开关控制位</p> <p>0: 禁止 LCD PUMP</p> <p>1: 允许 LCD PUMP</p>

位编号	位符号	说明
		注意：PUMPON 为 1 时电容偏压型 LCD 才有效
11	LCDSEL	LCD 驱动模式选择 0: 选择电阻型 LCD 驱动器 1: 选择电容型 LCD 驱动器 注意：选择电容型 LCD 驱动时需要外接电容
10	PUMPCK	PUMP 时钟频率 0: 2kHz 1: 8kHz 注意：PUMP 时钟频率越高，功耗越高，此位建议用户写 0
9~8	DDRCK[1:0]	LCD 频帧分频设置位 00: B 波形帧频 64Hz, A 波形帧频 32Hz 01: B 波形帧频 128Hz, A 波形帧频 64Hz 10: B 波形帧频 256Hz, A 波形帧频 128Hz 11: 保留
7	DDRON	LCD 显示驱动使能控制位 0: 显示驱动扫描关闭 1: 显示驱动扫描打开
4	TPYE	LCD 驱动波形选择位 0: B 波形 1: A 波形
3	VOIRSIF	LCD 快速充电使能位 0: 关闭快速充电 1: 打开快速充电，选择 11k 电阻快速充电 5 个周期，之后切换到 VOIRS 选择的电阻值
1	BIAS	LCD 显示驱动偏置电压设置 0: 1/4 偏置电压 1: 1/3 偏置电压
31~16 12 6~5 2, 0	-	保留

28.4.1.2 显示驱动配置寄存器 DDR_CFG

寄存器	读/写	说明	复位值	上电初始值
DDR_CFG	读/写	显示驱动配置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	VLCD[3:0]			
7	6	5	4	3	2	1	0
SCS	-	DUTY[1:0]		-	-	VOIRS[1:0]	

位编号	位符号	说明
11~8	VLCD[3:0]	LCD 电压调节设置位 LCD 的输出电压： $V_{LCD}=V_{DD}*(17+VLCD[3:0])/32$ 注意：当选择电容型 LCD 驱动时，此位无效
7	SCS	LCD Segment/Common 复用管脚选择位 0: 设定 1/4 占空比时，S0~S54 为 segment，C0~C3 为 common 1: 设定 1/4 占空比时，S4~S54 为 segment，C4~C7 为 common
5~4	DUTY[1:0]	LCD 显示占空比设置位 00: 1/8 占空比，S4~S54 为 segment，C0~C7 为 common 01: 1/6 占空比，S2~S54 为 segment，C0~C5 为 common 10: 1/5 占空比，S1~S54 为 segment，C0~C4 为 common 11: 1/4 占空比，S0~S54 为 segment，C0~C3 为 common 或 S4~S54 为 segment，C4~C7 为 common
1~0	VOIRS[1:0]	LCD 电压输出口分压电阻选择位 00: 设定内部分压电阻总电阻值为 11k 01: 设定内部分压电阻总电阻值为 100k 10: 设定内部分压电阻总电阻值为 300k 11: 设定内部分压电阻总电阻值为 800k 注意：当选择电容型 LCD 驱动时，此位无效
31~12 6 3~2	-	保留

28.4.1.3 SEG 口使能寄存器 SEG_EN0

寄存器	读/写	说明	复位值	上电初始值
SEG_EN0	读/写	SEG 口使能寄存器 0	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
23	22	21	20	19	18	17	16
SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
15	14	13	12	11	10	9	8
SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
7	6	5	4	3	2	1	0
SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0

位编号	位符号	说明
31~0	SEGx (x=0~31)	SEGx 口显示驱动输出控制位，x= 0~31 0: 关闭 SEGx 口的显示驱动输出功能 1: 打开 SEGx 口的显示驱动输出功能

28.4.1.4 SEG 口使能寄存器 SEG_EN1

寄存器	读/写	说明	复位值	上电初始值
SEG_EN1	读/写	SEG 口使能寄存器 1	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	SEG54	SEG53	SEG52	SEG51	SEG50	SEG49	SEG48
15	14	13	12	11	10	9	8
SEG47	SEG46	SEG45	SEG44	SEG43	SEG42	SEG41	SEG40
7	6	5	4	3	2	1	0
SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32

位编号	位符号	说明
22~0	SEGx (x=54~32)	SEGx 口显示驱动输出控制位, x= 54~32 0: 关闭 SEGx 口的显示驱动输出功能 1: 打开 SEGx 口的显示驱动输出功能

28.4.1.5 COM 口使能寄存器 COM_EN

寄存器	读/写	说明	复位值	上电初始值
COM_EN	读/写	COM 口使能寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0

位编号	位符号	说明
7~0	COMx (x=0~7)	COMx 口显示驱动输出控制位, x= 0~7 0: 关闭 COMx 口的显示驱动输出功能 1: 打开 COMx 口的显示驱动输出功能
31~8	-	保留

28.4.1.6 SEGn 显示寄存器 SEGRn

寄存器	读/写	说明	复位值	上电初始值
SEGRn (n=0~54)	读/写	SEGRn 显示寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8

-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0

位编号	位符号	说明
7~0	COMx (x=0~7)	COMm 对应的 SEGn 显示驱动输出控制位, n=0~54, m=0~7 用于设置 COMm 口对应的 SEGn 显示驱动输出 0: 关闭 1: 打开
31~8	-	保留

28.4.2 LCD 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
LCD 基地址: 0x4002_2280					
DDR_CON	0x00	读/写	显示驱动控制寄存器	0x0000_0000	0x0000_0000
DDR_CFG	0x04	读/写	显示驱动配置寄存器	0x0000_0000	0x0000_0000
SEG_EN0	0x08	读/写	SEG 口使能寄存器 0	0x0000_0000	0x0000_0000
SEG_EN1	0x0C	读/写	SEG 口使能寄存器 1	0x0000_0000	0x0000_0000
COM_EN	0x10	读/写	COM 口使能寄存器	0x0000_0000	0x0000_0000
SEGR 基地址: 0x4002_2300					
SEGR0	0x00	读/写	SEG0 显示寄存器	0x0000_0000	0x0000_0000
SEGR1	0x04	读/写	SEG1 显示寄存器	0x0000_0000	0x0000_0000
SEGR2	0x08	读/写	SEG2 显示寄存器	0x0000_0000	0x0000_0000
SEGR3	0x0C	读/写	SEG3 显示寄存器	0x0000_0000	0x0000_0000
SEGR4	0x10	读/写	SEG4 显示寄存器	0x0000_0000	0x0000_0000
SEGR5	0x14	读/写	SEG5 显示寄存器	0x0000_0000	0x0000_0000
SEGR6	0x18	读/写	SEG6 显示寄存器	0x0000_0000	0x0000_0000
SEGR7	0x1C	读/写	SEG7 显示寄存器	0x0000_0000	0x0000_0000
SEGR8	0x20	读/写	SEG8 显示寄存器	0x0000_0000	0x0000_0000
SEGR9	0x24	读/写	SEG9 显示寄存器	0x0000_0000	0x0000_0000
SEGR10	0x28	读/写	SEG10 显示寄存器	0x0000_0000	0x0000_0000
SEGR11	0x2C	读/写	SEG11 显示寄存器	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
SEGR12	0x30	读/写	SEG12 显示寄存器	0x0000_0000	0x0000_0000
SEGR13	0x34	读/写	SEG13 显示寄存器	0x0000_0000	0x0000_0000
SEGR14	0x38	读/写	SEG14 显示寄存器	0x0000_0000	0x0000_0000
SEGR15	0x3C	读/写	SEG15 显示寄存器	0x0000_0000	0x0000_0000
SEGR16	0x40	读/写	SEG16 显示寄存器	0x0000_0000	0x0000_0000
SEGR17	0x44	读/写	SEG17 显示寄存器	0x0000_0000	0x0000_0000
SEGR18	0x48	读/写	SEG18 显示寄存器	0x0000_0000	0x0000_0000
SEGR19	0x4C	读/写	SEG19 显示寄存器	0x0000_0000	0x0000_0000
SEGR20	0x50	读/写	SEG20 显示寄存器	0x0000_0000	0x0000_0000
SEGR21	0x54	读/写	SEG21 显示寄存器	0x0000_0000	0x0000_0000
SEGR22	0x58	读/写	SEG22 显示寄存器	0x0000_0000	0x0000_0000
SEGR23	0x5C	读/写	SEG23 显示寄存器	0x0000_0000	0x0000_0000
SEGR24	0x60	读/写	SEG24 显示寄存器	0x0000_0000	0x0000_0000
SEGR25	0x64	读/写	SEG25 显示寄存器	0x0000_0000	0x0000_0000
SEGR26	0x68	读/写	SEG26 显示寄存器	0x0000_0000	0x0000_0000
SEGR27	0x6C	读/写	SEG27 显示寄存器	0x0000_0000	0x0000_0000
SEGR28	0x70	读/写	SEG28 显示寄存器	0x0000_0000	0x0000_0000
SEGR29	0x74	读/写	SEG29 显示寄存器	0x0000_0000	0x0000_0000
SEGR30	0x78	读/写	SEG30 显示寄存器	0x0000_0000	0x0000_0000
SEGR31	0x7C	读/写	SEG31 显示寄存器	0x0000_0000	0x0000_0000
SEGR32	0x80	读/写	SEG32 显示寄存器	0x0000_0000	0x0000_0000
SEGR33	0x84	读/写	SEG33 显示寄存器	0x0000_0000	0x0000_0000
SEGR34	0x88	读/写	SEG34 显示寄存器	0x0000_0000	0x0000_0000
SEGR35	0x8C	读/写	SEG35 显示寄存器	0x0000_0000	0x0000_0000
SEGR36	0x90	读/写	SEG36 显示寄存器	0x0000_0000	0x0000_0000
SEGR37	0x94	读/写	SEG37 显示寄存器	0x0000_0000	0x0000_0000
SEGR38	0x98	读/写	SEG38 显示寄存器	0x0000_0000	0x0000_0000

寄存器	偏移地址	读/写	说明	复位值	上电初始值
SEGR39	0x9C	读/写	SEG39 显示寄存器	0x0000_0000	0x0000_0000
SEGR40	0xA0	读/写	SEG40 显示寄存器	0x0000_0000	0x0000_0000
SEGR41	0xA4	读/写	SEG41 显示寄存器	0x0000_0000	0x0000_0000
SEGR42	0xA8	读/写	SEG42 显示寄存器	0x0000_0000	0x0000_0000
SEGR43	0xAC	读/写	SEG43 显示寄存器	0x0000_0000	0x0000_0000
SEGR44	0xB0	读/写	SEG44 显示寄存器	0x0000_0000	0x0000_0000
SEGR45	0xB4	读/写	SEG45 显示寄存器	0x0000_0000	0x0000_0000
SEGR46	0xB8	读/写	SEG46 显示寄存器	0x0000_0000	0x0000_0000
SEGR47	0xBC	读/写	SEG47 显示寄存器	0x0000_0000	0x0000_0000
SEGR48	0xC0	读/写	SEG48 显示寄存器	0x0000_0000	0x0000_0000
SEGR49	0xC4	读/写	SEG49 显示寄存器	0x0000_0000	0x0000_0000
SEGR50	0xC8	读/写	SEG50 显示寄存器	0x0000_0000	0x0000_0000
SEGR51	0xCC	读/写	SEG51 显示寄存器	0x0000_0000	0x0000_0000
SEGR52	0xD0	读/写	SEG52 显示寄存器	0x0000_0000	0x0000_0000
SEGR53	0xD4	读/写	SEG53 显示寄存器	0x0000_0000	0x0000_0000
SEGR54	0xD8	读/写	SEG54 显示寄存器	0x0000_0000	0x0000_0000

29 DMA 控制器

29.1 概述

直接存储器访问(DMA)控制器用于高速数据传输。DMA 控制器可以从一个地址到另一个地址传输数据，无需 CPU 介入。通过 DMA 进行数据传输可减少 CPU 的工作量，将节省下的 CPU 资源做其他应用。DMA 控制器包含 2 个通道，每个通道都直接连接专用的硬件 DMA 请求，每个通道都同样支持软件触发。DMA 控制器支持 2 级通道优先级，用于处理 DMA 请求间的优先级，确保同一时刻只有一个 DMA 通道工作。DMA 控制器也支持单次传输和批量传输，请求源可以是软件请求或接口请求，内存之间的数据传输是使用软件请求。

注：对于一个双向数据传输应用，需要 2 个 DMA 通道分别完成发送和接收。

29.2 时钟源

DMA 的时钟源为 HCLK，通过 AHB_CFG.DMAEN 使能 DMA 的外设时钟

29.3 特性

- 支持 2 个可独立配置的通道
- 支持 2 个请求优先级
- 支持 8 位，16 位，32 位数据传输
- 支持源和目标地址自动增加或者固定，数据宽度支持字节，半字，字
- 支持单次和批量传输方式
- 传输方向支持：内存到内存、内存到外设、外设到内存、外设到外设

29.4 功能说明

29.4.1 传输方向

内存到内存	内存到外设	外设到内存	外设到外设
无限制	无限制	无限制	无限制

29.4.2 DMA 访问区域限制

用户操作 DMA 时，不允许对 Flash 进行写操作，也不允许通过 DMA 操作内核，否则将产生无法预估的异常。

29.4.3 通道优先级

通过寄存器 PL 可设置两个级别的优先级：

- 0：低
- 1：高

29.4.4 单次传输和批量传输

DMA 控制器支持单一和成组数据的传输类型，请求源可以是软件请求，接口请求，内存之间的数据传输是使用软件请求。单次传输的意思是软件或接口准备好传输一个数据（每个数据需要一次请求），批量传输的意思是软件或接口将传输多个数据（多个数据仅需一次请求）。

单次传输和批量传输模式可通过寄存器 TPTYPE (DMA_n_CFG[15]) 设定。

当 DMA 控制器运行在单次传输模式，每搬运一个数据需要一次请求，当搬运一次数据，寄存器 DMA_n_CNT[31:0]，n=0~1 会减 1，直到 DMA_n_CNT[31:0] 中的数目递减为 0，搬运才会完成。在该模式，BURSIZE(DMA_n_CFG[14:12]) 不用于控制搬运数据量大小，它的值固定为 1。

在批量搬运模式，DMA 控制器搬运 DMA_n_CNT[31:0] 个数据，仅需一次请求。当搬运 BURSIZE(DMA_n_CFG[14:12]) 数据后，DMA_n_CNT[31:0] 中的数目会减去 BURSIZE。直到 DMA_n_CNT[31:0] 中的数目递减为 0，搬运数据才完成。

29.4.5 循环模式

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。在循环模式传输过程中，待传输数据的数目将自动重新装载为在通道配置阶段设置的初始值，并继续响应 DMA 请求。为停止循环传输，软件需要在禁止 DMA 通道前使外设停止生成 DMA 请求（例如退出 ADC 扫描模式）。软件必须在启动/使能传输前，以及在停止循环传输后，明确设定 DMACNT 值。

SC32L14T 系列的 DMA 控制器支持常规模式和循环模式，用户可根据实际需求灵活选择：

- 当 CIRC=0（DMA 通道处于非循环模式）时，在达到设定的待传输数据数目时，将不再接受任何 DMA 请求；
- 当 CIRC=1（DMA 通道处于循环模式）时，在传输完成后该通道的 DMACNT 会自动重新装载之前设定的值，等待下一次循环。

用户可以根据实际需求灵活选择。

29.5 DMA 中断

对于每个 DMA 通道 n，n=0~1，在发生“传输完成”、“半传输”或“传输错误”时都会生成中断。可以使用单独的中断使能位以提高灵活性。

中断事件	事件标志位	中断请求控制位	子事件标志位	中断使能子开关
DMA 通道 n 传输完成	GIF	DMA _n _CFG ->INTEN	TCIF	TCIE
DMA 通道 n 传输一半			HTIF	HTIE
DMA 通道 n 传输错误			TEIF	TEIE

29.6 DMA 寄存器

29.6.1 DMA 相关寄存器表

29.6.1.1 DMA 通道 n 传输源地址缓存寄存器 DMA_n_SADR

寄存器	读/写	说明	复位值	上电初始值
DMA _n _SADR n = 0~1	读/写	DMA 通道 n 传输源地址缓存寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
31~0	SADR[31:0]	<p>DMA 传输源地址缓存</p> <ul style="list-style-type: none"> ● 读取： <ul style="list-style-type: none"> ■ 当通道打开时，读到的是内部的源地址工作寄存器； ■ 当通道禁止时，读到的是表面上的源地址缓存寄存器。 ● 更新： <ul style="list-style-type: none"> ■ 每搬运完一次，源地址工作寄存器会根据 SAINC[1:0]设置自动变化，其变化的宽度根据 TXWIDTH[1:0]决定。 ■ 循环模式下(SAINC == 11)，源地址缓存寄存器会重载至源地址工作寄存器。 ● 写入： <ul style="list-style-type: none"> ■ 写源地址缓存寄存器的条件：CHEN=0，或 CHEN=1，但 DMA 通道已传输完成，且处于 IDLE 状态。

29.6.1.2 DMA 通道 n 传输目标地址缓存寄存器 DMA_n_DADR

寄存器	读/写	说明	复位值	上电初始值
DMA _n _DADR n = 0~1	读/写	DMA 通道 n 目标地址缓存寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
31~0	DADR[31:0]	<p>DMA 传输目标地址缓存</p> <ul style="list-style-type: none"> ● 读取： <ul style="list-style-type: none"> ■ 当通道打开时，读到的是内部的目标地址工作寄存器； ■ 当通道禁止时，读到的是表面上的目标地址缓存寄存器。

位编号	位符号	说明
		<ul style="list-style-type: none"> ● 更新： <ul style="list-style-type: none"> ■ 每搬运完一次，目标地址工作寄存器会根据 DAINC[1:0]设置自动变化，其变化的宽度根据 TXWIDTH[1:0]决定。 ■ 循环模式下(SAINC == 11)，目标地址缓存寄存器会重载至目标地址工作寄存器。 ● 写入： <ul style="list-style-type: none"> ■ 写目标地址缓存寄存器的条件：CHEN=0，或 CHEN=1，但 DMA 通道已传输完成，且处于 IDLE 状态。

29.6.1.3 DMA 通道 n 控制/配置寄存器 DMA_n_CFG

寄存器	读/写	说明	复位值	上电初始值
DMA _n _CFG n = 0~1	读/写	DMA 通道 n 控制/配置寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	REQSRC[5:0]					
23	22	21	20	19	18	17	16
CHRQ	-	-	-	TEIE	HTIE	TCIE	INTEN
15	14	13	12	11	10	9	8
TPTYPE	BURSIZE[2:0]			SAINC[1:0]		DAINC[1:0]	
7	6	5	4	3	2	1	0
CHEN	CHRST	PAUSE	CIRC	TXWIDTH[1:0]		-	PL

位编号	位符号	说明
29~24	REQSRC[5:0]	<p>DMA 通道请求源选择位</p> <p>0: 禁用当前 DMA 通道的外设请求</p> <p>选择以下配置值，当前 DMA 通道的外设请求源将配合设置项中的外设 DMA 请求使能开关产生：</p> <p>2: UART0_IDE->TXDMAEN</p> <p>3: UART0_IDE->RXDMAEN</p> <p>4: UART1_IDE->TXDMAEN</p> <p>5: UART1_IDE->RXDMAEN</p> <p>12: TWI_SPI0_IDE->TXDMAEN</p> <p>13: TWI_SPI0_IDE->RXDMAEN</p> <p>14: TWI_SPI1_IDE->TXDMAEN</p> <p>15: TWI_SPI1_IDE->RXDMAEN</p> <p>33: TIM1_IDE->TIDE</p> <p>34: TIM1_IDE->CAPFDE</p> <p>35: TIM1_IDE->CAPRDE</p> <p>36: TIM2_IDE->TIDE</p> <p>37: TIM2_IDE->CAPFDE</p> <p>38: TIM2_IDE->CAPRDE</p> <p>48: TIM6_IDE->TIDE</p> <p>49: TIM6_IDE->CAPFDE</p> <p>50: TIM6_IDE->CAPRDE</p> <p>59: ADCCON->DMAEN</p> <p>60: DMA0_CFG->CHRQ</p> <p>61: DMA1_CFG->CHRQ</p> <p>其它：禁用 DMA 外设请求</p>
23	CHRQ	<p>DMA 通道的 DMA 请求使能位</p> <p>0: 禁止，当前 DMA 通道禁止作为其它 DMA 通道的请求源</p>

位编号	位符号	说明
		<p>1: 使能, 当前 DMA 通道可作为其它 DMA 通道的请求源, 即当前 DMA 通道和其它外设一样, 可产生 DMA 请求。 该位使能后, 可以实现 DMA 请求 DMA, 例如: CHRQ =1, DMA 通道 n 完成数据搬运后, 向 DMA 通道 m 产生一个 DMA 请求, 通道 m 响应请求, 将预先配置好的参数表更新至通道 n 的寄存器, 从而实现通道 n 的参数自动更新。 注意: CHRQ 置起后, 作为请求源的 DMA 能够进行数据搬运, 但不会置起标志位和进入相应中断, 需要 CHRQ 位置 0 后, 才会置起标志位及进入中断。</p>
19	TEIE	<p>DMA 传输错误中断使能位 0: DMA 传输错误中断除能 1: DMA 传输错误中断使能</p>
18	HTIE	<p>DMA 传输一半中断使能位 0: DMA 传输一半中断除能 1: DMA 传输一半中断使能</p>
17	TCIE	<p>DMA 传输完成中断使能位 0: DMA 传输完成中断除能 1: DMA 传输完成中断使能</p>
16	INTEN	<p>中断请求 CPU 的使能控制位 0: 禁止中断请求 1: 使能中断请求</p>
15	TPTYPE	<p>DMA 通道传输类型选择位 0: 单次传输 1: 批量传输。批量传输模式下, DMA 控制器搬运 DMACNT 个数据仅需一次请求, 通道响应该请求后, 数据将以 Burst 方式进行传输, 即以 BURSIZE 为单位进行数据搬运直到 DMACNT 递减为 0, 一次批量传输模式下的数据处理才算完成。</p>
14~12	BURSIZE[2:0]	<p>批量传输时, 基于 Burst 传输方式下的定义, Burst 大小可选择: 000: 128 001: 64 010: 32 011: 16 100: 8 101: 4 110: 2 111: 1</p>
11~10	SAINC[1:0]	<p>DMA 通道传输源地址增减模式设置位 00: 无增量 (固定地址模式) 01: 增量模式 10: 减量模式 11: 递增循环模式 (见 DMA 传输源地址缓存寄存器) SAINC[1:0]的值可以任意修改, 在通道禁止时立即生效; 在通道使能时, 修改值在循环模式重装载时生效</p>
9~8	DAINC[1:0]	<p>DMA 传输目标地址增减模式设置位 00: 无增量 (固定地址模式) 01: 增量模式 10: 减量模式 11: 递增循环模式 (见 DMA 传输目标地址缓存寄存器) DAINC[1:0]的值可以任意修改, 在通道禁止时立即生效; 在通道使能时, 修改值在循环模式重装载时生效</p>
7	CHEN	<p>DMA 通道使能位 0: DMA 通道禁止 1: DMA 通道使能</p>

位编号	位符号	说明
6	CHRST	DMA 通道复位控制位，该位用于控制 DMA 通道复位。 0: 无效 1: 当前 DMA 通道复位。此时，当前 DMA 通道的 CHEN 被除能，中断标志为被清除，其他寄存器的值保持不变
5	PAUSE	DMA 通道传输暂停控制位 0: 无效 1: 当前 DMA 通道暂停。此时，当前 DMA 通道的 CHEN 被除能，状态机在完成当前读写周期后回到 state=1，内部寄存器的值处于保持状态(源/目的地址，计数器)，当再次使能当前 DMA 通道的 CHEN 时，当前 DMA 通道的会继续上一次的传输；PAUSE 会被写 CHEN 的动作清 0
4	CIRC	DMA 通道循环模式使能位 0: 通道未处于循环模式，在达到设定的待传输数据数目时，该通道的 DMACNT 会保持为零； 1: 通道处于循环模式，在传输完成后该通道的 DMACNT 会自动重新装载之前设定的值。 循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。在循环模式传输过程中，待传输数据的数目将自动重新装载为在通道配置阶段设置的初始值，并继续响应 DMA 请求。为停止循环传输，软件需要在禁止 DMA 通道前使外设停止生成 DMA 请求（例如退出 ADC 扫描模式）。软件必须在启动/使能传输前，以及在停止循环传输后，明确设定 DMACNT 值。
3~2	TXWIDTH[1:0]	DMA 通道传输宽度选择位 选择当前 DMA 通道的源地址及目标地址每次传输的数据宽度： 00: 8bit 01: 16bit 10: 32bit 11: 32bit TXWIDTH[1:0] 的值可以任意修改，在通道禁止时，立即生效；在通道打开时，修改值在循环模式重新装载时生效。
0	PL	DMA 通道优先级设置位 在 DMA 已经有通道在工作，且其它通道也接收到请求正挂起，当正在工作的通道结束后将启动优先级仲裁。 0: 低 1: 高 注意：同等优先级配置，通道号越小优先级越高。
31~30 22~20 1	-	保留

29.6.1.4 DMA 通道 n 计数器缓存寄存器 DMA_n_CNT

寄存器	读/写	说明	复位值	上电初始值
DMA _n _CNT n = 0~1	读/写	DMA 通道 n 计数器缓存寄存器	0x0000_0000	0x0000_0000

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

位编号	位符号	说明
31~0	DMACNT[31:0]	<p>DMA 通道计数器缓存寄存器</p> <ul style="list-style-type: none"> ● 写入： <ul style="list-style-type: none"> ■ DMACNT 的值等于当前 DMA 通道的剩余传输次数； ■ 每个 DMA 通道内部都有一个“工作计数器”，该计数器会在每次搬运后以为 TXWIDTH 为单位递减： <ul style="list-style-type: none"> ◆ 当 CIRC=0（DMA 通道处于非循环模式）时，“工作计数器”递减到 0 后，将不再接受任何 DMA 请求。 ◆ 当 CIRC=1（DMA 通道处于循环模式）时，“工作计数器”递减到 0 后，会将 DMACNT 的值重载到“工作计数器”内，等待下一次循环。 ● 读取： <ul style="list-style-type: none"> ■ 当通道禁止时，读出的是 DMACNT 的值； ■ 当通道打开时，读出的是内部的“工作计数器”的实时数据。

29.6.1.5 DMA 通道 n 状态寄存器 DMA_n_STS

寄存器	读/写	说明	复位值	上电初始值
DMA _n _STS n = 0~1	读/写	DMA 通道 n 状态寄存器	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	SWREQ
7	6	5	4	3	2	1	0
STATUS[3:0]				TEIF	HTIF	TCIF	GIF

位编号	位符号	说明
8	SWREQ	<p>DMA 通道的软件请求触发位</p> <p>该位写 1 后，当前 DMA 通道会一直挂起软件请求，直到当前 DMA 通道响应，并将该位自动硬件清 0。</p>
7~4	STATUS[3:0]	<p>DMA 通道状态位</p> <p>0000：空闲。</p> <p>0001：写入源地址</p> <p>0010：读取源地址数据，并写入目的地址</p> <p>0011：写入目的地址数据</p> <p>0100：保留</p> <p>0101：挂起等待中(有通道在忙，其他通道请求挂起)</p> <p>0110：暂停等待中(批量传输模式时 PAUSE 写 1 后)</p> <p>0111：burst 传输中</p> <p>1000：burst 传输停止：PAUSE 使能、DMACNT 计数到 0，或 bursize 计数到 0 均会进入此状态</p>
3	TEIF	<p>DMA 传输错误中断标志位</p> <p>该位由硬件置 1，通过软件写 1 清 0。</p> <p>当 DMA 读写到未定义的地址时，TEIF 会被硬件置 1。</p>
2	HTIF	<p>DMA 传输一半中断标志位</p> <p>该位由硬件置 1，通过软件写 1 清 0。</p> <p>当 DMACNT 的计数值计数到 DMACNT/2 时，HTIF 会被硬件置 1。</p>
1	TCIF	<p>DMA 传输完成中断标志位</p> <p>该位由硬件置 1，通过软件写 1 清 0。</p>

		当 DMACNT 的计数值计到 0 时，TCIF 会被硬件置 1。
0	GIF	DMA 通道全局中断标志位 0: 当前 DMA 通道无中断产生 1: 当前 DMA 通道产生中断: 传输错误、传输到一半或传输完成
31~9	-	保留

29.6.2 DMA 寄存器映射

寄存器	偏移地址	读/写	说明	复位值	上电初始值
DMA0 基地址: 0x4001_0800					
DMA0_SADR	0x00	读/写	DMA 源地址缓存寄存器	0x0000_0000	0x0000_0000
DMA0_DADR	0x04	读/写	DMA 目标地址缓存寄存器	0x0000_0000	0x0000_0000
DMA0_CFG	0x08	读/写	DMA 控制/配置寄存器	0x0000_0000	0x0000_0000
DMA0_CNT	0x0C	读/写	DMA 计数器缓存寄存器	0x0000_0000	0x0000_0000
DMA0_STS	0x10	读/写	DMA 状态寄存器	0x0000_0000	0x0000_0000
DMA1 基地址: 0x4001_0840					
DMA1_SADR	0x00	读/写	DMA 源地址缓存寄存器	0x0000_0000	0x0000_0000
DMA1_DADR	0x04	读/写	DMA 目标地址缓存寄存器	0x0000_0000	0x0000_0000
DMA1_CFG	0x08	读/写	DMA 控制/配置寄存器	0x0000_0000	0x0000_0000
DMA1_CNT	0x0C	读/写	DMA 计数器缓存寄存器	0x0000_0000	0x0000_0000
DMA1_STS	0x10	读/写	DMA 状态寄存器	0x0000_0000	0x0000_0000

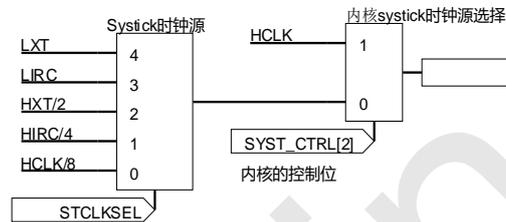
30 SysTick

SysTick 是一个简单的、24 位写入清 0、递减、带灵活控制机制的自动装载计数器。该计数器可以用作实时操作系统（RTOS）的滴答定时器或作为一个简单的计数。

30.1 时钟源

SysTick（Cortex®-M0+内核系统定时器）的时钟源分为内部时钟源和外部时钟源：

- 内部时钟源即 CPU 时钟
- 外部时钟源有 5 个



SysTick 时钟源框图

30.2 SysTick 校准寄存器默认值

SysTick 校准寄存器校准值设置方法如下：

- 若上电默认时钟为 f_{HCLK}/n （MHz）， n 是上电默认分频系数，上电默认时钟源为 HIRC；
- 则 SysTick 校准值初始值为 $1000 * (f_{HCLK}/n)$ ，即保证默认可产生 1ms 时间基准。

31 版本记录

版本	记录	日期
V0.3	<ol style="list-style-type: none"> 更正描述：更正 ADC 章节描述“ADC 的转换时间约为 950ns” 更正描述：更正 ADC 采样周期控制位 LOWSP[2:0]中系统时钟为 24M，采样时间根据对应系统时钟调整 更正描述：更正 ADCV 寄存器复位值为 0x0000_3FFF 更正描述：更正外部中断寄存器映射中 INT 基地址为 0x4001_1080 更正描述：更正 SPI 寄存器位 QTWCK[3:0]描述：为确保 SC32L14T/14G 系列 SPI0/1 正确通信，通信频率请选择 6MHz 以下 	2025 年 11 月 27 日
V0.2	<ol style="list-style-type: none"> 更正描述：删除 GPIO PE7 及其对应的外部中断 INT 功能 补充描述：新增 UART3 映射到烧录管脚时禁止使用全双工模式的相关说明 更正描述：LIRC 的±4%的误差温度限制在 25℃ 更正描述：IAP_CON 寄存器中添加连续烧录相关寄存器位 CONT[5:0]和 DMAEN 更正描述：更正 APB1_CFG 寄存器，补充寄存器位 TIM4EN 更正描述：更正 RTC 基地址为 0x4002_21B0 更正描述：更正 DDR_CON 寄存器，删除寄存器位 DMOD 	2025 年 9 月 11 日
V0.1	初版	2025 年 6 月 4 日

32 声明

深圳市赛元微电子股份有限公司（以下简称赛元）保留随时对赛元产品、文档或服务进行变更、更正、增强、修改和改进的权利，恕不另行通知。赛元认为提供的信息是准确可信的。本文档信息于 2025 年 6 月开始使用。在实际进行生产设计时，请参阅各产品最新的数据手册等相关资料。

Preliminary