

**32 位 MCU  
HR8P296**

# 数 据 手 册

ARM Cortex™-M0 内核

- 产品简介
- 数据手册
- 产品规格

上海东软载波微电子有限公司

2020 年 03 月 26 日

## 使用注意事项

### 关于芯片的上/下电

本公司系列芯片具有独立电源管脚。当芯片应用在多电源供电系统时，应保证芯片与所连系统其它部件同步上电，或先对芯片上电，再对所连系统其它部件上电；反之，下电时，应避免 MCU 先于所连系统其它部件下电的情况。否则可能导致芯片内部元件过压或过流，从而导致芯片故障或元件退化，甚至出现系统异常。具体可参照芯片的数据手册说明。

### 关于芯片的复位

本公司系列芯片具有内部上电复位。对于不同的快速上/下电或慢速上/下电系统，内部上电复位电路可能失效，建议用户使用外部复位、下电复位、看门狗复位等，确保复位电路正常工作。在系统设计时，若使用外部复位电路，建议采用三极管复位电路、RC 复位电路。若不使用外部复位电路，建议将复位管脚通过电阻接到固定电平，或采取必要的电源抖动处理电路或其它保护电路。具体可参照芯片的数据手册说明。

### 关于芯片的时钟

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

### 关于芯片的初始化

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

### 关于芯片的管脚

本公司系列芯片具有宽范围的输入管脚电平，建议用户输入高电平应在  $V_{IH}$  的最小阈值之上，低电平应在  $V_{IL}$  的最大阈值之下。避免输入电压介于上述两个电平阈值之间，以免波动噪声进入芯片。对于未使用的管脚，建议用户设为输入状态，并通过电阻接至电源或地。对未使用的管脚处理因应用系统而异，具体遵循应用系统的相关规定和说明。

### 关于芯片的 ESD 防护措施

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

### 关于芯片的 EFT 防护措施

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

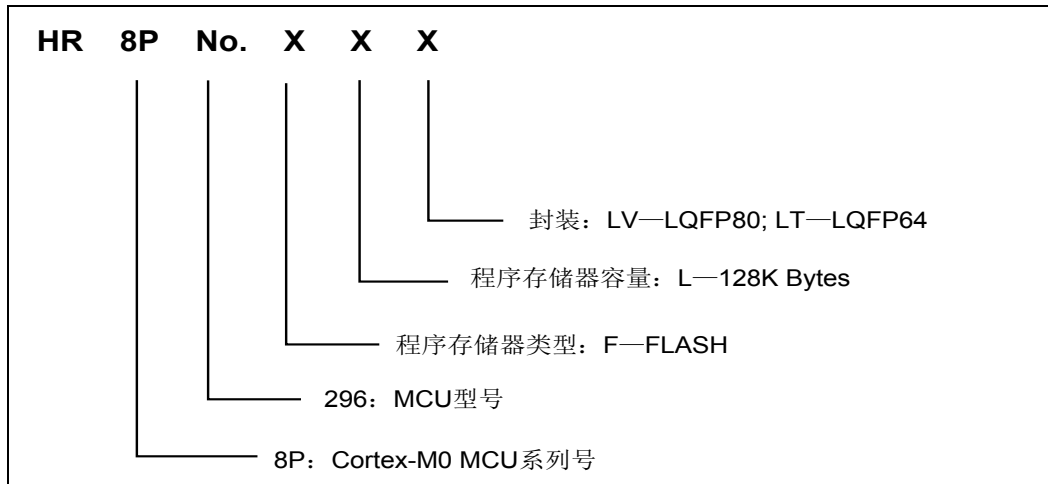
### 关于芯片的开发环境

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

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

## 产品订购信息

型号	内核	程序存储器	数据存储器	封装
HR8P296FLLV	ARM Cortex™-M0	128K Bytes	12K Bytes	LQFP80
HR8P296FLLT				LQFP64



地址: 中国上海市龙漕路 299 号天华信息科技园 2A 楼 5 层

邮编: 200235

E-mail: support@essemi.com

电话: +86-21-60910333

传真: +86-21-60914991

网址: <http://www.essemi.com/>

版权所有©

**上海东软载波微电子有限公司**

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

## 修订历史

版本	日期	修订摘要
V1.0	2016-12-20	初版发布
V1.1	2017-6-16	<ol style="list-style-type: none"> <li>1. 更新 PC 端口相关寄存器的偏移地址值;</li> <li>2. 更新 RTC 相关寄存器;</li> <li>3. 新增芯片唯一识别码小节;</li> <li>4. 更新 LCD 偏置选择说明;</li> <li>5. 增加 I2C 发送空闲中断标志 TIDLEIF 清除标志位; TIDLECF 定义;</li> <li>6. 更新 VR30 和 VR36 对 LCD 和 ADC 供电的描述;</li> <li>7. 更新 RTC_WA、RTC_DA、RTC_HOUR 寄存器 bit 位的定义。</li> </ol>
V1.2	2018-1-17	<ol style="list-style-type: none"> <li>1. 新增 PA18 管脚禁用说明;</li> <li>2. BOR 最低档位电压更新为 1.8V。</li> </ol>
V1.3	2019-3-12	<ol style="list-style-type: none"> <li>1. 补充外部复位 MRSTN 和 PLL 锁定的部分描述;</li> <li>2. 补充电气特性说明细节;</li> <li>3. 更新低电压检测寄存器 LVDVS bit 值定义;</li> <li>4. 添加睡眠模式相关注意事项;</li> <li>5. 变更 Logo。</li> </ol>
V1.4	2019-9-19	<ol style="list-style-type: none"> <li>1. 更新 LCD BIAS 偏置描述;</li> <li>2. 更新低速 ADC 转换时的 ADC core 功耗配置推荐描述;</li> <li>3. 芯片功耗特性表中增加 RTC 不使能时深度睡眠的芯片电流;</li> <li>4. 更新 LCD 段使能寄存器 LCD_SEL1 的描述。</li> </ol>
V1.5	2019-12-30	<ol style="list-style-type: none"> <li>1. 更新 ADC 寄存器中关于 ADC 转换速度控制及 ADC 功耗选择的描述;</li> <li>2. 更新 MRSTN 复位参考电路图和相關注意事项。</li> </ol>
V1.6	2020-03-26	<ol style="list-style-type: none"> <li>1. 更新 HOSC Rf,Rs 电阻描述;</li> <li>2. 修改 T32N 输出调制功能示意图。</li> </ol>

## 目录

### 内容目录

<b>第 1 章</b>	<b>芯片简介</b> .....	<b>13</b>
1.1	概述 .....	13
1.2	应用领域 .....	15
1.3	结构框图 .....	16
1.4	管脚图 .....	17
1.4.1	LQFP80 封装图 .....	17
1.4.2	LQFP64 封装图 .....	18
1.5	管脚说明 .....	19
1.6	管脚复用 .....	20
1.6.1	PA 管脚复用 .....	20
1.6.2	PB 管脚复用 .....	21
1.6.3	PC 管脚复用 .....	22
<b>第 2 章</b>	<b>系统控制及操作特性</b> .....	<b>23</b>
2.1	系统控制保护 .....	23
2.1.1	概述 .....	23
2.1.2	特殊功能寄存器 .....	23
2.2	系统电源 .....	23
2.2.1	结构框图 .....	23
2.2.2	芯片供电电源 .....	24
2.2.3	芯片内部 3.0V 稳压输出 VR30 .....	24
2.2.4	芯片内部 3.6V 稳压输出 VR36 .....	24
2.2.5	特殊功能寄存器 .....	24
2.3	系统复位 .....	25
2.3.1	概述 .....	25
2.3.2	结构框图 .....	25
2.3.3	复位时序图 .....	25
2.3.4	外部复位 MRSTN 参考 .....	26
2.3.5	特殊功能寄存器 .....	26
2.4	低电压监测 (LVD) .....	28
2.4.1	概述 .....	28
2.4.2	特殊功能寄存器 .....	28
2.5	系统低功耗操作模式 .....	30
2.5.1	概述 .....	30
2.5.2	睡眠模式 .....	30
2.5.3	深度睡眠模式 .....	30
2.5.4	唤醒方式 .....	31
2.5.5	唤醒时间 .....	31
2.5.6	FLASH 存储器等待功能 .....	31
2.5.7	特殊功能寄存器 .....	31
2.6	系统时钟 .....	32
2.6.1	时钟源概述 .....	32
2.6.2	结构框图 .....	32
2.6.3	时钟源 .....	32
2.6.4	系统时钟源切换 .....	34
2.6.5	系统时钟分频 .....	34
2.6.6	外设模块时钟配置 .....	34
2.6.7	特殊功能寄存器 .....	34

2.7	中断和异常处理 .....	40
2.7.1	中断和异常 .....	40
2.7.2	中断和异常向量的分配 .....	41
2.7.3	中断向量表的重映射 .....	42
2.7.4	特殊功能寄存器 .....	42
2.8	系统控制块 (SCB) .....	50
2.8.1	概述 .....	50
2.8.2	特殊功能寄存器 .....	50
2.9	系统定时器 (SYSTICK) .....	53
2.9.1	概述 .....	53
2.9.2	特殊功能寄存器 .....	53
<b>第3章</b>	<b>存储资源 .....</b>	<b>56</b>
3.1	内部存储器地址映射 .....	56
3.2	FLASH 存储器 .....	57
3.2.1	芯片配置字 .....	57
3.2.1.1	概述 .....	57
3.2.1.2	特殊功能寄存器 .....	59
3.2.2	芯片唯一识别码 .....	59
3.2.3	程序区 FLASH .....	60
3.2.4	自编程操作 (IAP) .....	61
3.2.4.1	概述 .....	61
3.2.4.2	IAP 操作流程图 .....	61
3.2.4.3	特殊功能寄存器 .....	65
3.3	SRAM 存储器 .....	69
3.3.1	SRAM 地址映射 .....	69
3.3.2	SRAM 位带扩展 .....	69
3.4	外设寄存器 .....	70
3.4.1	外设寄存器地址映射 .....	70
3.4.2	系统控制单元 (SCU) 寄存器列表 .....	71
3.4.3	GPIO 寄存器列表 .....	71
3.4.4	FLASH 存储器 IAP 寄存器列表 .....	72
3.4.5	ADC 寄存器列表 .....	73
3.4.6	RTC 寄存器列表 .....	73
3.4.7	LCDC 寄存器列表 .....	73
3.4.8	WDT 寄存器列表 .....	74
3.4.9	T16N0/T16N1/T16N2/T16N3 寄存器列表 .....	74
3.4.10	T32N0/T32N1/T32N2 寄存器列表 .....	75
3.4.11	UART0/UART1/UART2/UART3 寄存器列表 .....	75
3.4.12	EUART0/1 寄存器列表 .....	76
3.4.13	SPI 寄存器列表 .....	76
3.4.14	I2C 寄存器列表 .....	77
3.4.15	外设寄存器位带扩展 .....	77
3.5	内核寄存器 .....	77
3.5.1	系统定时器 (SYSTICK) 寄存器列表 .....	77
3.5.2	中断控制器 (NVIC) 寄存器列表 .....	78
3.5.3	系统控制块 (SCB) 寄存器列表 .....	78
<b>第4章</b>	<b>输入输出端口 .....</b>	<b>79</b>
4.1	概述 .....	79
4.2	结构框图 .....	80
4.3	外部端口中断 .....	80
4.4	特殊功能寄存器 .....	81

<b>第 5 章</b>	<b>外设 .....</b>	<b>104</b>
5.1	定时器/计数器 .....	104
5.1.1	16 位定时器/计数器 T16N (T16N0/T16N1/T16N2/T16N3) .....	104
5.1.1.1	概述 .....	104
5.1.1.2	结构框图 .....	104
5.1.1.3	T16N 定时/计数功能 .....	104
5.1.1.4	T16N 输入捕捉功能 .....	106
5.1.1.5	T16N 输出调制功能 .....	107
5.1.1.6	特殊功能寄存器 .....	108
5.1.1.7	T16N 应用说明 .....	115
5.1.2	32 位定时器/计数器 T32N (T32N0/T32N1/T32N2) .....	116
5.1.2.1	概述 .....	116
5.1.2.2	结构框图 .....	116
5.1.2.3	T32N 定时/计数功能 .....	116
5.1.2.4	T32N 输入捕捉功能 .....	118
5.1.2.5	T32N 输出调制功能 .....	119
5.1.2.6	特殊功能寄存器 .....	120
5.1.2.7	T32N 应用说明 .....	126
5.2	通用异步接收/发送器 (UART0/UART1/UART2/UART3) .....	127
5.2.1	概述 .....	127
5.2.2	结构框图 .....	127
5.2.3	UART 数据格式 .....	127
5.2.4	UART 异步发送器 .....	128
5.2.5	UART 异步接收器 .....	130
5.2.6	UART 红外发送调制功能 .....	132
5.2.7	UART 红外唤醒功能 .....	133
5.2.8	UART 端口极性 .....	133
5.2.9	特殊功能寄存器 .....	133
5.2.10	UART 应用说明 .....	142
5.3	增强型通用异步接收/发送器 (EUART0/EUART1) .....	143
5.3.1	概述 .....	143
5.3.2	结构框图 .....	143
5.3.3	EUART 端口复用 .....	143
5.3.4	普通 UART 通讯模式 .....	144
5.3.5	7816 通讯模式的异步接收器和发送器 .....	144
5.3.6	7816 通讯模式的数据格式 .....	145
5.3.7	7816 通讯模式的自动重发功能 .....	145
5.3.8	7816 通讯模式的自动重收功能 .....	146
5.3.9	特殊功能寄存器 .....	146
5.3.10	EUART 应用说明 .....	155
5.4	SPI 同步串口通讯控制器 (SPI) .....	156
5.4.1	概述 .....	156
5.4.2	结构框图 .....	156
5.4.3	SPI 通讯模式 .....	156
5.4.4	SPI 数据格式 .....	156
5.4.5	SPI 同步发送器 .....	158
5.4.6	SPI 同步接收器 .....	159
5.4.7	SPI 通讯控制 .....	159
5.4.8	SPI 延迟接收功能 .....	160
5.4.9	SPI 数据帧发送间隔功能 .....	160
5.4.10	特殊功能寄存器 .....	161

5.4.11	SPI 应用说明.....	166
5.5	I2C 总线串口通讯控制器 (I2C) .....	167
5.5.1	概述.....	167
5.5.2	结构框图 .....	167
5.5.3	I2C 总线基本原理.....	168
5.5.3.1	I2C 通讯协议.....	168
5.5.3.2	I2C 数据传输格式 .....	168
5.5.4	I2C 通讯端口配置.....	169
5.5.5	I2C 时基定时器与 16 倍速采样器 .....	170
5.5.6	I2C 通讯发送器 .....	171
5.5.7	I2C 通讯接收器 .....	172
5.5.8	I2C 通讯控制.....	172
5.5.8.1	I2C 起始位 .....	173
5.5.8.2	I2C 停止位 .....	173
5.5.8.3	I2C 应答延迟功能 .....	174
5.5.8.4	I2C 数据帧传输间隔功能 .....	174
5.5.8.5	I2C 时钟线自动下拉等待请求功能.....	175
5.5.8.6	I2C 自动发送未应答功能 .....	175
5.5.9	特殊功能寄存器.....	176
5.5.10	I2C 应用说明.....	183
5.6	模数转换器 (ADC) .....	184
5.6.1	概述.....	184
5.6.2	结构框图 .....	184
5.6.3	ADC 数据转换.....	184
5.6.4	特殊功能寄存器.....	185
5.7	实时时钟 (RTC) .....	189
5.7.1	概述.....	189
5.7.2	RTC 写保护.....	189
5.7.3	时间日期设置 .....	189
5.7.4	RTC 中断源.....	190
5.7.5	特殊功能寄存器.....	191
5.8	液晶显示控制器 (LCDC) .....	200
5.8.1	概述.....	200
5.8.2	结构框图 .....	200
5.8.3	LCDC 基本设置.....	200
5.8.4	LCDC 偏置选择.....	200
5.8.5	LCDC 像素对照表 .....	200
5.8.6	LCDC 工作时钟源.....	202
5.8.7	LCD 显示帧频率.....	202
5.8.8	LCDC 像素寄存器的刷新操作.....	202
5.8.9	LCD 闪烁功能 .....	202
5.8.10	LCDC 中断.....	203
5.8.11	特殊功能寄存器.....	203
5.9	看门狗定时器 (WDT) .....	212
5.9.1	概述.....	212
5.9.2	特殊功能寄存器.....	212
<b>第 6 章</b>	<b>芯片封装图 .....</b>	<b>215</b>
6.1	80-PIN 封装图.....	215
6.2	64-PIN 封装图.....	216
<b>附录 1</b>	<b>Cortex-M0 内核描述.....</b>	<b>217</b>
附录 1.1	Cortex-M0 指令集 .....	217



附录 1.2	Cortex-M0 内核寄存器 .....	219
附录 1.2.1	通用寄存器 R0~R12 .....	219
附录 1.2.2	堆栈指针寄存器 SP (R13) .....	219
附录 1.2.3	链接寄存器 LR (R14) .....	220
附录 1.2.4	程序计数器 PC (R15) .....	220
附录 1.2.5	程序状态寄存器 xPSR .....	220
附录 1.2.6	异常/中断屏蔽寄存器 PRIMASK .....	221
附录 1.2.7	控制寄存器 CONTROL .....	221
<b>附录 2</b>	<b>电气特性 .....</b>	<b>223</b>
附录 2.1	参数特性表 .....	223
附录 2.1.1	芯片工作条件 .....	223
附录 2.1.2	芯片特性参数测量方法 .....	224
附录 2.1.3	芯片功耗特性 .....	224
附录 2.1.4	芯片 IO 端口特性 .....	226
附录 2.1.5	芯片系统时钟特性 .....	227
附录 2.1.6	上电和下电工作条件 .....	227
附录 2.1.7	芯片 ESD 特性 .....	227
附录 2.1.8	芯片功能模块特性 .....	228
附录 2.2	参数特性图 .....	231
附录 2.2.1	芯片功耗特性 .....	231
附录 2.2.2	芯片 IO 端口输入特性 .....	232
附录 2.2.3	芯片 IO 端口输出特性 (普通驱动) .....	235
附录 2.2.4	芯片 IO 端口输出特性 (大电流驱动) .....	237
<b>附录 3</b>	<b>编程调试接口 .....</b>	<b>239</b>
附录 3.1	概述 .....	239
附录 3.2	ISP 编程接口 .....	239
附录 3.2.1	通信协议 .....	239
附录 3.2.2	操作流程 .....	240
附录 3.3	SWD 调试接口 .....	241
附录 3.3.1	概述 .....	241
附录 3.3.2	SWD 特性 .....	241
<b>附录 4</b>	<b>LCD 驱动波形 .....</b>	<b>242</b>
附录 4.1	概述 .....	242
附录 4.2	驱动波形图 .....	242

图目录

图 1-1	芯片结构框图	16
图 1-2	芯片 LQFP80 封装顶视图	17
图 1-3	芯片 LQFP64 封装顶视图	18
图 2-1	系统电源结构框图	23
图 2-2	系统复位电路结构框图	25
图 2-3	上电复位时序示意图	25
图 2-4	掉电复位时序示意图	25
图 2-5	MRSTN 复位参考电路图 1	26
图 2-6	MRSTN 复位参考电路图 2	26
图 2-7	系统时钟电路结构框图	32
图 2-8	LOSC 振荡器电路示意图	33
图 2-9	HOSC 振荡器电路示意图	33
图 2-10	系统定时器框图	53
图 3-1	内部存储器地址映射	56
图 3-2	FLASH 映射图	60
图 3-3	IAP 操作请求流程图	61
图 3-4	IAP 全擦除操作流程图	62
图 3-5	IAP 页擦除操作流程图	62
图 3-6	IAP 编程操作流程图	63
图 3-7	中断向量表重映射操作示意图	64
图 3-8	SRAM 映射图	69
图 3-9	外设寄存器映射图	70
图 4-1	I/O 端口电路结构框图	80
图 4-2	PINT0 外部端口中断电路结构示意图	80
图 5-1	T16N 电路结构框图	104
图 5-2	T16N 计数匹配功能示意图	106
图 5-3	T16N 捕捉功能示意图	107
图 5-4	T16N 输出调制功能示意图	108
图 5-5	T32N 电路结构框图	116
图 5-6	T32N 计数匹配功能示意图	117
图 5-7	T32N 捕捉功能示意图	119
图 5-8	T32N 输出调制功能示意图	120
图 5-9	UART 电路结构图	127
图 5-10	UART 7 位数据格式	127
图 5-11	UART 8 位数据格式	128
图 5-12	UART 9 位数据格式	128
图 5-13	UART 发送数据流示意图	129
图 5-14	UART 发送数据操作流程图示例	130
图 5-15	UART 接收数据流示意图	131
图 5-16	UART 接收数据操作流程图示例	132
图 5-17	TX 高电平调制输出波形图	132
图 5-18	TX 低电平调制输出波形图	133
图 5-19	EUART0 电路结构框图	143
图 5-20	7816 通讯模式发送数据流示意图	144
图 5-21	7816 通讯模式接收数据流示意图	144
图 5-22	SPI 电路结构框图	156
图 5-23	SPI 时钟上升沿发送, 下降沿接收波形示意图	157
图 5-24	SPI 时钟下降沿发送, 上升沿接收波形示意图	157
图 5-25	SPI 时钟上升沿接收, 下降沿发送波形示意图	157

图 5-26	SPI 时钟下降沿接收，上升沿发送波形示意图.....	158
图 5-27	SPI 发送数据流示意图 .....	158
图 5-28	SPI 接收数据流示意图 .....	159
图 5-29	SPI 延迟接收功能波形示意图 .....	160
图 5-30	I2C 电路结构框图.....	167
图 5-31	I2C 总线通讯协议示意图.....	168
图 5-32	I2C 主控器写入从动器数据示意图 .....	169
图 5-33	I2C 主控器读取从动器数据示意图 .....	169
图 5-34	开漏输出端口示意图 .....	170
图 5-35	I2C 总线端口信号的波形示意图.....	171
图 5-36	I2C 发送数据流示意图 .....	171
图 5-37	I2C 接收数据流示意图 .....	172
图 5-38	I2C 起始位波形图.....	173
图 5-39	I2C 自动寻呼波形图 .....	173
图 5-40	I2C 停止位波形图.....	174
图 5-41	I2C 应答延迟功能波形示意图 .....	174
图 5-42	I2C 数据帧传输间隔功能波形示意图 .....	175
图 5-43	I2C 时钟线下拉等待波形示意图.....	175
图 5-44	ADC 内部结构图 .....	184
图 5-45	ADC 数据转换时序示意图 (ADC_CON1 寄存器的 SMPS=0) .....	184
图 5-46	ADC 数据转换时序示意图 (ADC_CON1 寄存器的 SMPS=1) .....	185
图 5-47	LCDC 驱动模块框图 .....	200

## 表目录

表 1-1	芯片管脚说明表.....	20
表 1-2	PA 复用表.....	21
表 1-3	PB 复用表 .....	21
表 1-4	PC 复用表 .....	22
表 2-1	低功耗模式时钟状态列表 .....	30
表 2-2	异常/中断优先级操作类型说明列表 .....	40
表 2-3	异常/中断优先级列表 .....	40
表 2-4	IRQ 分配列表 .....	42
表 4-1	PINT 选择对应列表 .....	102
表 5-1	I2C 总线端口信号的时序参数列表 .....	170
表 5-2	12/24 小时模式对照表.....	190
表 5-3	LCDC 像素对照列表 .....	202
表 5-4	LCD 复用类型列表 .....	204

## 第1章 芯片简介

### 1.1 概述

该产品是一款高集成度的通用 MCU 芯片，内部集成 32 位 ARM Cortex-M0 CPU 内核。内部集成多个 16 位和 32 位定时器/计数器，带红外发送调制功能的 UART 模块，兼容 7816 协议的通信接口，SPI 和 I2C 通信模块，带实时时钟模块 RTC，支持停显及闪烁功能的 LCD 驱动模块，以及用于系统电源和电池电源监测的 12 位 ADC 和 LVD 模块等外设。

- ◆ 工作条件
  - ◇ 工作电压范围：2.5V ~ 5.5V
  - ◇ 工作温度范围：-40 ~ 85℃（工业级）
- ◆ 封装
  - ◇ LQFP64 封装（支持 56 个 I/O 端口）
  - ◇ LQFP80 封装（支持 72 个 I/O 端口）
- ◆ 电源
  - ◇ 系统电源输入 VDD，支持工作电压为 5V 或 3.3V 的应用系统
  - ◇ 低功耗 LVD 用于监测系统电源掉电和上电，可选择产生掉电或上电中断
  - ◇ 内部 3.0V LDO 电压输出 VR30，用于内部 ADC，需外接电容
- ◆ 时钟
  - ◇ 32.768KHz 晶体振荡器（支持外部 32.768KHz 时钟输入）作为 RTC 时钟源（内置振荡器匹配电容），可配置为系统时钟源
  - ◇ 2M~16MHz 晶体振荡器（支持外部时钟输入）可配置为系统时钟源，支持内部分频器
  - ◇ 内部 16MHz RC 振荡器可配置为系统时钟源，出厂校准精度±1%（32.768KHz 晶体振荡器工作时，支持实时调校，确保全工作范围时钟精度）
  - ◇ 内部 32KHz RC 振荡器作为 WDT 时钟源，可配置为系统时钟源
  - ◇ 支持 PLL 倍频，最大可倍频至 33MHz（用于系统时钟）。当 PLL 时钟源选择为 LOSC 或 LRC 时，倍频至约 33.55MHz（ $32.768\text{KHz} \times 1024 = 33.554432\text{MHz}$ ）；当选择时钟源为外部或内部 4MHz 时，倍频至 32MHz（ $4\text{MHz} \times 8 = 32\text{MHz}$ ）时钟输出。
- ◆ 内核
  - ◇ ARM Cortex-M0 32 位嵌入式处理器内核
  - ◇ 支持 SWD 串行调试接口，支持 2 个监视点（watchpoint）和 4 个断点（breakpoint）
  - ◇ 内嵌向量中断控制器 NVIC
  - ◇ 支持唤醒中断控制器 WIC
  - ◇ NVIC 包含一个不可屏蔽中断 NMI
  - ◇ 内置 1 个 SysTick 系统定时器
- ◆ 硬件看门狗
  - ◇ 时钟源可选择
  - ◇ 支持低功耗模式下唤醒
  - ◇ 超时计数溢出可选择触发中断或复位
- ◆ 存储器
  - ◇ 128K 字节 FLASH 存储器

- 支持 ISP 在线串行编程
- 支持 IAP 在应用中编程
- 支持 FLASH 编程代码加密保护
- ◇ 12K 字节 SRAM 存储器
- SRAM 存储空间及外设寄存器地址空间支持位带 (Bit band) 扩展
- ◆ I/O 端口
  - ◇ 最大支持 72 个 I/O 端口
  - PA 端口 (PA0~PA31)
  - PB 端口 (PB0~PB23)
  - PC 端口 (PC0~PC15) (仅 80PIN 支持)
  - ◇ 支持 2 个大电流驱动口 PA12、PA24, 最大灌电流能力 20mA
  - ◇ 支持 8 路外部中断输入, 触发方式可配置
- ◆ 定时器/计数器
  - ◇ 支持 4 路 16 位定时器/计数器 T16N0, T16N1, T16N2, T16N3
  - ◇ 支持 3 路 32 位定时器/计数器 T32N0, T32N1, T32N2
  - ◇ 定时器/计数器带预分频器, 扩展输入捕捉/输出调制 PWM 功能
- ◆ UART 通信接口
  - ◇ 支持 4 路通信接口 UART0, UART1, UART2, UART3
  - ◇ 支持全/半双工异步通信模式
  - ◇ 支持传输波特率可配置
  - ◇ 支持 4 字节接收和 4 字节发送缓冲器
  - ◇ 支持 7 位/8 位/9 位数据格式
  - ◇ 支持硬件产生奇偶校验位作为第 9 位数据传输, 支持用户的第 9 位数据传输
  - ◇ 支持接收帧错误标志、溢出标志、奇偶校验错误标志
  - ◇ 支持数据接收和发送中断
  - ◇ 支持红外发送调制输出
  - ◇ 接收端口支持红外唤醒功能
  - ◇ 支持接收和发送端口极性可配置
- ◆ EUART 通信接口
  - ◇ 支持 2 路通信接口 EUART0, EUART1
  - ◇ 兼容 UART 通信接口, 可配置为普通 UART 模式
  - ◇ 扩展支持异步半双工接收/发送 (7816 模式)
  - ◇ 扩展支持 8 位数据位和 1 位奇偶校验位 (7816 模式)
  - ◇ 扩展支持自动重发重收模式 (7816 模式)
  - ◇ 扩展支持可配置内部时钟输出 (7816 模式)
  - ◇ 扩展支持双通道通讯可配置 (7816 模式)
- ◆ I2C 通信接口
  - ◇ 支持 1 路通信接口 I2C
  - ◇ 支持单主控模式/从动模式
  - ◇ 支持标准的 I2C 总线协议, 最大传输速率 400Kbit/s
  - ◇ 支持 7 位从机地址

- ◇ 支持 4 字节接收/发送缓冲器
- ◇ 支持数据接收和发送中断
- ◇ SCL/SDA 端口支持推挽/开漏模式，开漏时必须使能内部弱上拉或使用外部上拉电阻
- ◇ SCL 端口支持时钟线自动下拉等待请求功能
- ◆ SPI 通信接口
  - ◇ 支持 1 路通信接口 SPI
  - ◇ 支持主控模式/从动模式
  - ◇ 支持 4 种通信数据格式
  - ◇ 支持 4 字节接收/发送缓冲器
  - ◇ 支持数据接收和发送中断
- ◆ ADC 模数转换器
  - ◇ 支持 1 路模数转换器 ADC
  - ◇ 支持 12 位采样精度
  - ◇ 支持 11 通道模拟输入
  - ◇ 支持外部和内部参考电压选择
  - ◇ 内置可调校参考电压，可从 AVREFP 复用端口输出
  - ◇ 支持中断产生
- ◆ LCDC 液晶显示控制器
  - ◇ 支持最大 8 COM x 56SEG
  - ◇ 支持灰度调节功能
  - ◇ 支持显示闪烁功能，闪烁频率可调
- ◆ RTC 实时时钟
  - ◇ 仅 POR 上电复位有效，支持软件写保护，有效避免系统不稳定对时钟造成的影响
  - ◇ 采用外部 32.768KHz 晶体振荡器作为时钟源
  - ◇ 时钟调校可实现最大时间精度为±0.0254ppm
  - ◇ 时间计数（实现小时、分钟和秒）和日历计数（实现年、月、日和星期），BCD 格式
  - ◇ 提供 4 个可编程定时中断和 2 个可编程日历闹钟
  - ◇ 提供一路可配置时钟输出
  - ◇ 自动闰年识别，有效期到 2099 年
  - ◇ 12 小时和 24 小时模式可配置
  - ◇ 低功耗设计：工作电压为 3.6V 时，RTC 模块工作电流典型值为 1.5μA (最大值为 3μA)

## 1.2 应用领域

本芯片可用于家电，汽车电子，智能家居，航模玩具以及工业控制仪表等领域。

### 1.3 结构框图

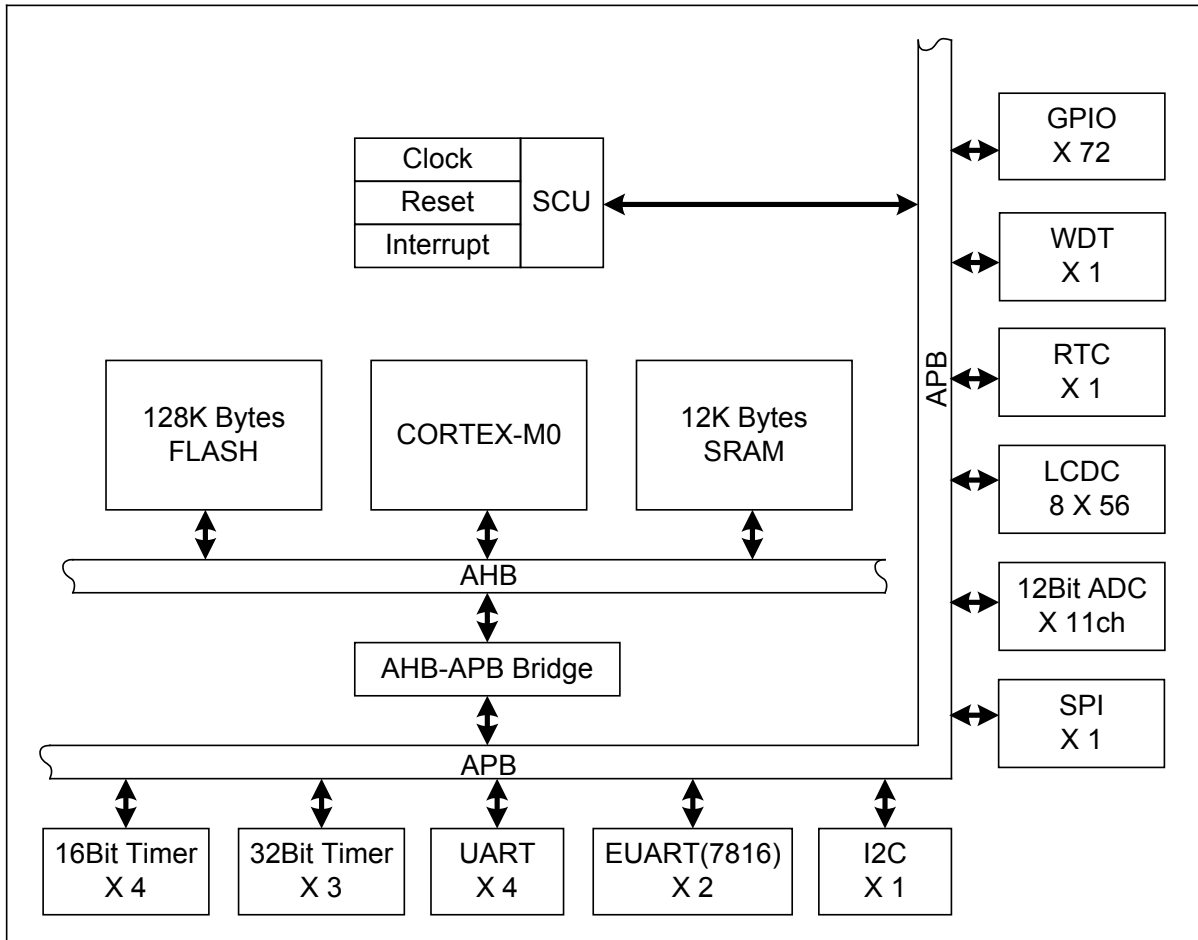


图 1-1 芯片结构框图



## 1.4 管脚图

### 1.4.1 LQFP80 封装图

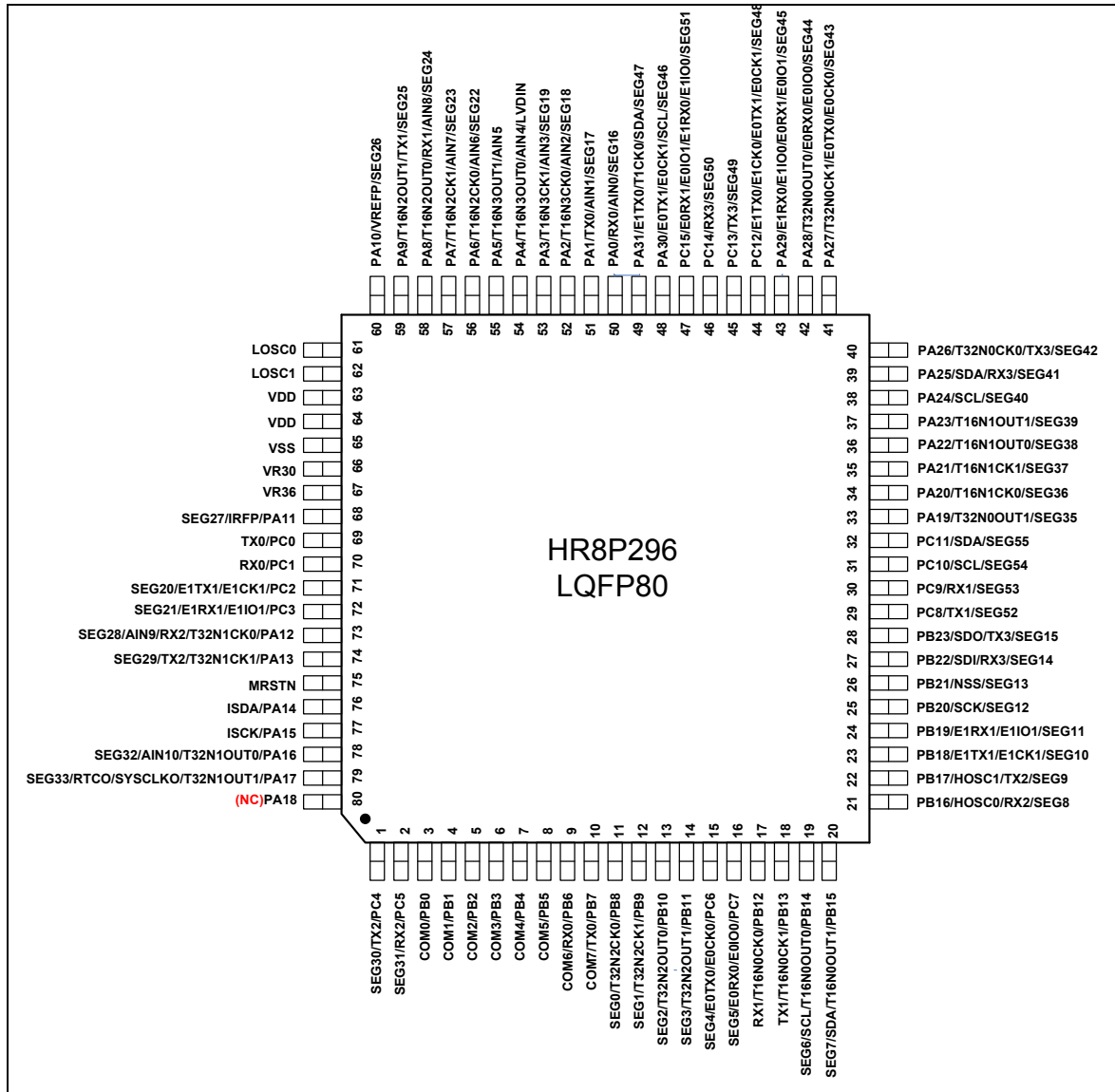


图 1-2 芯片 LQFP80 封装顶视图

### 1.4.2 LQFP64 封装图

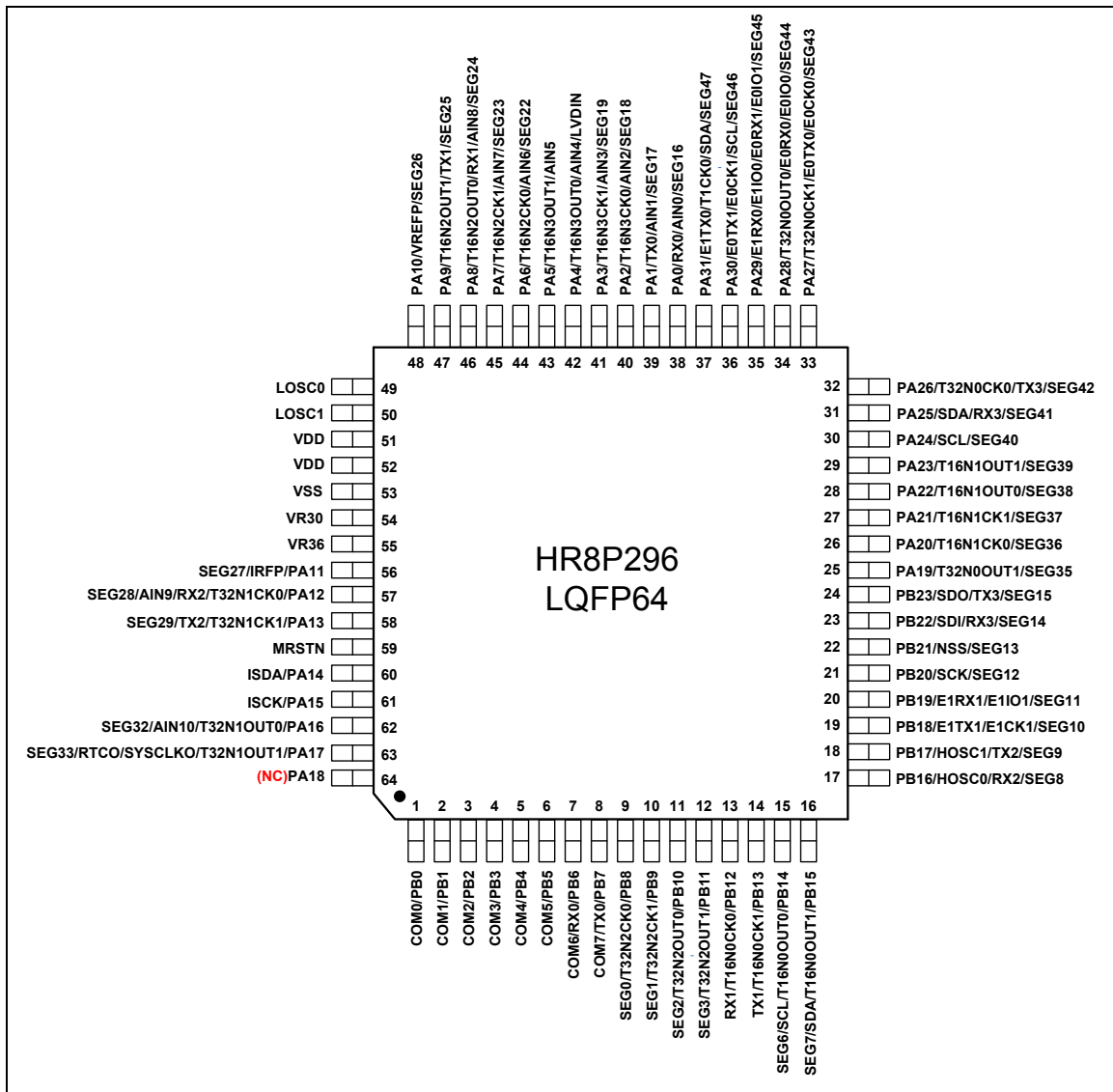


图 1-3 芯片 LQFP64 封装顶视图

- 注 1: 所有 PA/PB 端口都可以软件配置为外部端口中断功能, 具体参考 4.3 节《外部端口中断》。
- 注 2: 支持两组编程调试端口, 可配置芯片配置字选择其中一组作为调试端口。
- 注 3: 编程/调试接口共用 5 线接口配置, 即电源线 VDD、地线 VSS、复位线 MRSTN、时钟线 ISCK 和数据线 ISDA。
- 注 4: 其中 PA18 由于与芯片内部基准电压测试输出复用, 用户禁止使用 PA18, 必须将 PA18 悬空, 程序将 PA18 配置为 GPIO 输出口, 且禁止内部上下拉。
- 注 5: 在应用系统中, 若不使用外部 LOSC 晶振, 则需将 LOSC 配置为禁止状态, 并将 LOSC0 和 LOSC1 管脚均下拉到地 VSS (推荐下拉电阻约 1K), 否则可能会导致芯片功耗异常或降低芯片抗干扰性能。

## 1.5 管脚说明

管脚名称	输入类型	输出类型	A/D	管脚说明
PA0~PA31	TTL	CMOS	D	通用 I/O 端口
PB0~PB23	TTL	CMOS	D	通用 I/O 端口
PC0~PC15	TTL	CMOS	D	通用 I/O 端口
ISCK	TTL	—	D	编程/调试串行时钟端口
ISDA	TTL	CMOS	D	编程/调试串行数据端口
AIN0~AIN10	—	—	A	ADC 模拟通道 0~10
TX0~TX3	—	CMOS	D	UART0~UART3 发送输出
RX0~RX3	TTL	—	D	UART0~UART3 接收输入
E0TX0/E0TX1	—	CMOS	D	EUART0 发送输出
E0RX0/E0RX1	TTL	—	D	EUART0 接收输入
E0CK0/E0CK1	—	CMOS	D	EUART0 模块内部时钟输出端口
E0IO0/E0IO1	TTL	CMOS	D	EUART0 模块数据端口
E1TX0/E1TX1	—	CMOS	D	EUART1 发送输出
E1RX0/E1RX1	TTL	—	D	EUART1 接收输入
E1CK0/E1CK1	—	CMOS	D	EUART1 模块内部时钟输出端口
E1IO0/E1IO1	TTL	CMOS	D	EUART1 模块数据端口
SCK	TTL	—	D	SPI 时钟输入
NSS	TTL	CMOS	D	SPI 片选信号
SDI	TTL	—	D	SPI 数据输入
SDO	—	CMOS	D	SPI 数据输出
SCL	TTL	—	D	I2C 时钟
SDA	TTL	CMOS	D	I2C 数据
SEG0~SEG55	—	—	D	LCD Segment 端口
COM0~COM7	—	—	D	LCD Common 端口
T16N0CK0/T16N0CK1 T16N1CK0/T16N1CK1 T16N2CK0/T16N2CK1 T16N3CK0/T16N3CK1	TTL	—	D	T16N0/T16N1/T16N2/T16N3 外部时钟输入
T16N0IN0/T16N0IN1 T16N1IN0/T16N1IN1 T16N2IN0/T16N2IN1 T16N3IN0/T16N3IN1	TTL	—	D	T16N0/T16N1/T16N2/T16N3 输入捕捉端口
T16N0OUT0/T16N0OUT1 T16N1OUT0/T16N1OUT1 T16N2OUT0/T16N2OUT1 T16N3OUT0/T16N3OUT1	—	CMOS	D	T16N0/T16N1/T16N2/T16N3 调制输出端口
T32N0CK0/T32N0CK1 T32N1CK0/T32N1CK1 T32N2CK0/T32N2CK1	TTL	—	D	T32N0/T32N1/T32N2 外部时钟输入
T32N0IN0/T32N0IN1 T32N1IN0/T32N1IN1 T32N2IN0/T32N2IN1	TTL	—	D	T32N0/T32N1/T32N2 输入捕捉端口
T32N0OUT0/T32N0OUT1 T32N1OUT0/T32N1OUT1 T32N2OUT0/T32N2OUT1	—	CMOS	D	T32N0/T32N1/T32N2 调制输出端口
PINT0~PINT7	TTL	—	D	外部中断输入端口
LVDIN	—	—	A	LVD 检测通道

管脚名称	输入类型	输出类型	A/D	管脚说明
VREFP	—	—	A	外部参考电压
MRSTN	TTL	—	D	芯片主复位，低电平有效
VR30	—	—	A	芯片内部 LDO 输出 3.0V，外接 10nf 电容
VR36	—	—	A	芯片内部 LDO 输出 3.6V
HOSC0	—	—	A	2~20MHz 晶体振荡器端口
HOSC1	—	—	A	
LOSC0	—	—	A	32.768KHz 晶体振荡器端口
LOSC1	—	—	A	
VDD	—	—	P	系统主电源
VSS	—	—	P	系统地

表 1-1 芯片管脚说明表

注 1: A = 模拟端口, D = 数字端口, P = 电源/地;

注 2: MRSTN 表示低电平复位有效;

注 3: ISCK 和 ISDA 为 FLASH 存储器串行烧录和 SWD 调试协议复用端口, 具体参考附录 3。

## 1.6 管脚复用

### 1.6.1 PA管脚复用

PA 端口	FUN0(D)	FUN1(D)	FUN2(D)	FUN3(A)
PA0	PA0	—	RX0	SEG16/AIN0
PA1	PA1	—	TX0	SEG17/AIN1
PA2	PA2	T16N3CK0/T16N3IN0	—	SEG18/AIN2
PA3	PA3	T16N3CK1/T16N3IN1	—	SEG19/AIN3
PA4	PA4/ISDA	T16N3OUT0	—	AIN4/LVDIN
PA5	PA5/ISCK	T16N3OUT1	—	AIN5
PA6	PA6	T16N2CK0/T16N2IN0	—	SEG22/AIN6
PA7	PA7	T16N2CK1/T16N2IN1	—	SEG23/AIN7
PA8	PA8	T16N2OUT0	RX1	SEG24/AIN8
PA9	PA9	T16N2OUT1	TX1	SEG25
PA10	PA10	—	—	SEG26/VREFP
PA11	PA11	—	—	SEG27/IRFP
PA12	PA12	T32N1CK0/T32N1IN0	RX2	SEG28/AIN9
PA13	PA13	T32N1CK1/T32N1IN1	TX2	SEG29
PA14	PA14/ISDA	—	—	—
PA15	PA15/ISCK	—	—	—
PA16	PA16	T32N1OUT0	—	SEG32/AIN10
PA17	PA17	T32N1OUT1	RTCO/SYSCLKO	SEG33
PA18	PA18	—	—	—
PA19	PA19	T32N0OUT1	—	SEG35
PA20	PA20	T16N1CK0/T16N1IN0	—	SEG36
PA21	PA21	T16N1CK1/T16N1IN1	—	SEG37
PA22	PA22	T16N1OUT0	—	SEG38
PA23	PA23	T16N1OUT1	—	SEG39
PA24	PA24	SCL	—	SEG40
PA25	PA25	SDA	RX3	SEG41
PA26	PA26	T32N0CK0/T32N0IN0	TX3	SEG42

PA 端口	FUN0(D)	FUN1(D)	FUN2(D)	FUN3(A)
PA27	PA27	T32N0CK1/T32N0IN1	E0TX0/E0CK0	SEG43
PA28	PA28	T32N0OUT0	E0RX0/E0IO0	SEG44
PA29	PA29	—	E0RX1/E0IO1	SEG45
PA30	PA30	E0TX1/E0CK1	SCL	SEG46
PA31	PA31	—	SDA	SEG47

表 1-2 PA 复用表

- 注 1: 端口类型代表不同的驱动电流能力, 具体参考附录 2 电气特性章节。  
 注 2: 复用功能 FUN 的类型: (D)表示复用为数字端口, (A)表示复用为模拟端口。  
 注 3: PA12、PA24 是大电流端口。  
 注 4: 所有 PA 端口都可软件配置为外部中断端口, 具体参考 4.3 节《外部端口中断》。  
 注 5: 支持两组编程调试端口, 可配置芯片配置字选择其中一组作为调试端口。

### 1.6.2 PB管脚复用

PB 端口	FUN0(D)	FUN1(D)	FUN2(D)	FUN3(A)
PB0	PB0	—	—	COM0
PB1	PB1	—	—	COM1
PB2	PB2	—	—	COM2
PB3	PB3	—	—	COM3
PB4	PB4	—	—	COM4
PB5	PB5	—	—	COM5
PB6	PB6	—	RX0	COM6
PB7	PB7	—	TX0	COM7
PB8	PB8	T32N2CK0/T32N2IN0	—	SEG0
PB9	PB9	T32N2CK1/T32N2IN1	—	SEG1
PB10	PB10	T32N2OUT0	—	SEG2
PB11	PB11	T32N2OUT1	—	SEG3
PB12	PB12	T16N0CK0/T16N0IN0	RX1	—
PB13	PB13	T16N0CK1/T16N0IN1	TX1	—
PB14	PB14	T16N0OUT0	SCL	SEG6
PB15	PB15	T16N0OUT1	SDA	SEG7
PB16	PB16/HOSC0	—	RX2	SEG8
PB17	PB17/HOSC1	—	TX2	SEG9
PB18	PB18	E1TX/E1CK1	—	SEG10
PB19	PB19	E1RX/E1IO1	—	SEG11
PB20	PB20	SCK	—	SEG12
PB21	PB21	NSS	—	SEG13
PB22	PB22	SDI	RX3	SEG14
PB23	PB23	SDO	TX3	SEG15

表 1-3 PB 复用表

- 注 1: 端口类型代表不同的驱动电流能力, 具体参考附录 2 电气特性章节。  
 注 2: 复用功能 FUN 的类型: (D)表示复用为数字端口, (A)表示复用为模拟端口。  
 注 3: PB16~PB23 支持 SMIT 窗口选择。  
 注 4: 所有 PB 端口都可软件配置为外部中断端口, 具体参考 4.3 节《外部端口中断》。

### 1.6.3 PC管脚复用

PC 端口	FUN0(D)	FUN1(D)	FUN2(D)	FUN3(A)
PC0	PC0	—	TX0	—
PC1	PC1	—	RX0	—
PC2	PC2	E1TX1/E1CK1	—	SEG20
PC3	PC3	E1RX1/E1IO1	—	SEG21
PC4	PC4	—	TX2	SEG30
PC5	PC5	—	RX2	SEG31
PC6	PC6	E0TX0/E0CK0	—	SEG4
PC7	PC7	E0RX0/E0IO0	—	SEG5
PC8	PC8	—	TX1	SEG52
PC9	PC9	—	RX1	SEG53
PC10	PC10	SCL	—	SEG54
PC11	PC11	SDA	—	SEG55
PC12	PC12	E1TX0/E1CK0	E0TX1/E0CK1	SEG48
PC13	PC13	—	TX3	SEG49
PC14	PC14	—	RX3	SEG50
PC15	PC15	E0RX1/E0IO1	E1RX0/E1IO0	SEG51

表 1-4 PC 复用表

注 1: 端口类型代表不同的驱动电流能力, 具体参考附录 2 电气特性章节。

注 2: 复用功能 FUNx 的类型: (D)表示复用为数字端口, (A)表示复用为模拟端口。

## 第2章 系统控制及操作特性

### 2.1 系统控制保护

#### 2.1.1 概述

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

寄存器列表和基址参见：3.4.2 系统控制单元（SCU）寄存器列表。

#### 2.1.2 特殊功能寄存器

系统设置保护寄存器 (SCU_PROT)															
偏移地址: 00H															
复位值: 00000000_00000000_00000000_00000001B															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															PORT

—	bit31-1	W	写 0x55AA6996 时，位 PROT 为 0； 写其它值时位 PROT 为 1
PROT	bit0	R	<b>SCU 写保护位</b> 0: 写保护关闭 1: 写保护使能

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

注 2: SCU\_PROT 保护的寄存器为 SCU\_NMIC, SCU\_PWRC, SCU\_LVDC, SCU\_PCLKEN0, SCU\_PCLKEN1, SCU\_SCLKEN0, SCU\_SCLKEN1, SCU\_TBLRMEN, SCU\_TBLOFFS。

### 2.2 系统电源

#### 2.2.1 结构框图

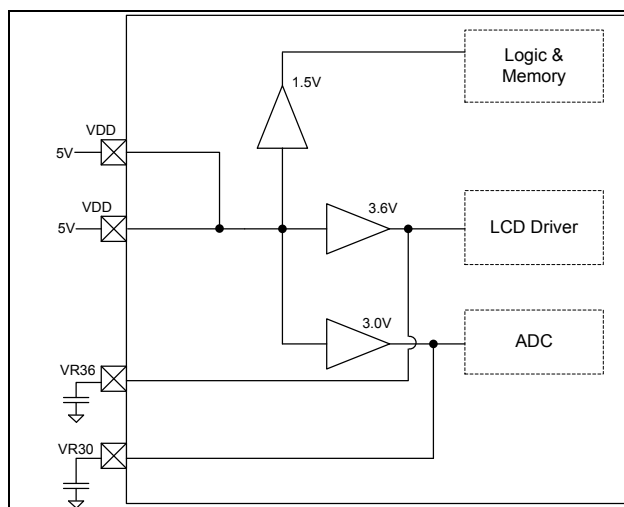


图 2-1 系统电源结构框图

### 2.2.2 芯片供电电源

芯片由 VDD 输入供电。VDD 输入电气特性请参考附录。

### 2.2.3 芯片内部 3.0V 稳压输出 VR30

当 VR30\_EN 使能, 芯片内部 ADC 采用内部 3.0V LDO VR30 供电, 当 VDD 电压低于 3.0V, LDO 跟随 VDD 变化。此时为保证内部 LDO 输出的稳定, 在系统应用时应通过 VR30 端口外接 LDO 稳压电容, 外部稳压电容值推荐选用 0.01uF。

当 VR30\_EN 禁止, 芯片内部 ADC 采用 VDD 供电, 此时 VR30 端口不对外输出电压。

VR30 仅用于芯片内部电路模块供电, 请勿用于芯片外部电路供电。

### 2.2.4 芯片内部 3.6V 稳压输出 VR36

当 VR36\_EN 使能, 芯片内部 LCD 电源 VLCD 采用内部 3.6V LDO VR36 供电, 当 VDD 电压低于 3.6V, LDO 跟随 VDD 变化。在系统应用时 VR36 端口无需外接 LDO 稳压电容。

当 VR36\_EN 禁止, 芯片内部 LCD 电源 VLCD 采用 VDD 供电, 此时 VR36 端口不对外输出电压。

VR36 仅用于芯片内部电路供电, 请勿用于芯片外部电路供电。

### 2.2.5 特殊功能寄存器

VR 控制寄存器 (SCU_VRCON)																
偏移地址: 50 <sub>H</sub>																
复位值: 00000000_00000000_00000001_00000000 <sub>B</sub>																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留							VR15_HEN	保留				VR36_EN	保留			VR30_EN
-	bit31-9							-	-							
VR15_HEN	bit8							R/W	VR15 高功耗模式电流增加 100uA 控制位 0: 禁止 1: 使能							
-	bit7-5							-	-							
VR36_EN	Bit4								VR36 输出使能位 0: 禁止(VR36 输出配置禁止后, 3.6V LDO 关闭, 不产生电流消耗, 3.6V LDO 输出短接至 VDD 5V 电源) 1: 使能							
-	bit3-1							-	-							
VR30_EN	bit0							R/W	VR30 输出使能位 0: 禁止(VR30 输出配置禁止后, 3.0V LDO 关闭, 不产生电流消耗, 3.0V LDO 输出短接至 VDD 5V 电源) 1: 使能							

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



## 2.3 系统复位

### 2.3.1 概述

- ◆ 支持 POR 上电复位
- ◆ 支持 BOR 低电压监测复位
- ◆ 支持 MRSTN 外部端口复位
- ◆ 支持 WDT 看门狗溢出复位
- ◆ 支持 Cortex-M0 调试接口软件复位

### 2.3.2 结构框图

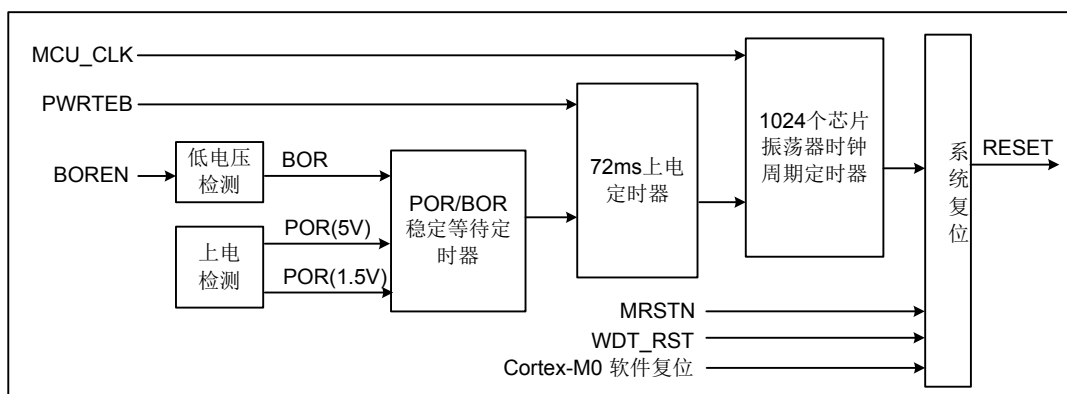


图 2-2 系统复位电路结构框图

### 2.3.3 复位时序图

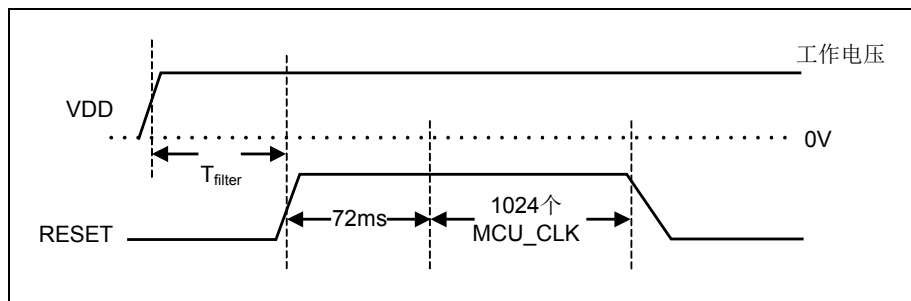


图 2-3 上电复位时序示意图

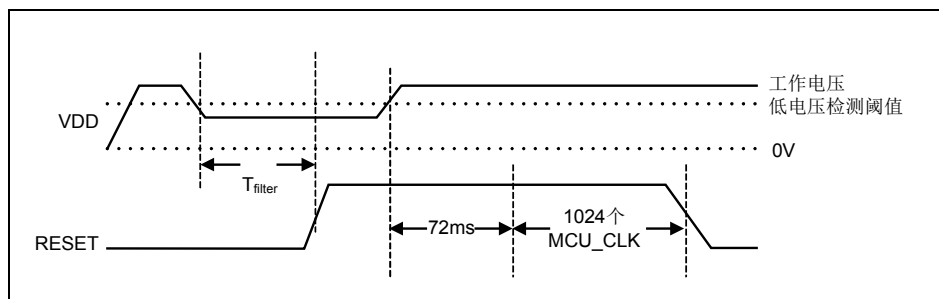


图 2-4 掉电复位时序示意图

通过配置字控制位 PWRTEB，可选择是否屏蔽 72ms 上电等待时间。

通过配置字控制位 BORV<1:0>，可选择掉电复位电压。

### 2.3.4 外部复位MRSTN参考

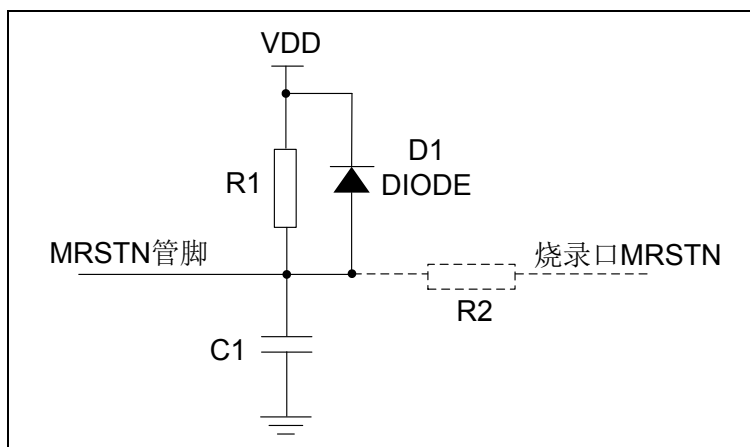


图 2-5 MRSTN 复位参考电路图 1

注：采用 RC 复位，其中  $47K\Omega \leq R1 \leq 100K\Omega$ ，电容  $C1 = (0.1\mu F)$ ，在有外部烧录口的应用系统中，需要串接 R2 作为限流电阻， $0.1K\Omega \leq R2 \leq 1K\Omega$ 。

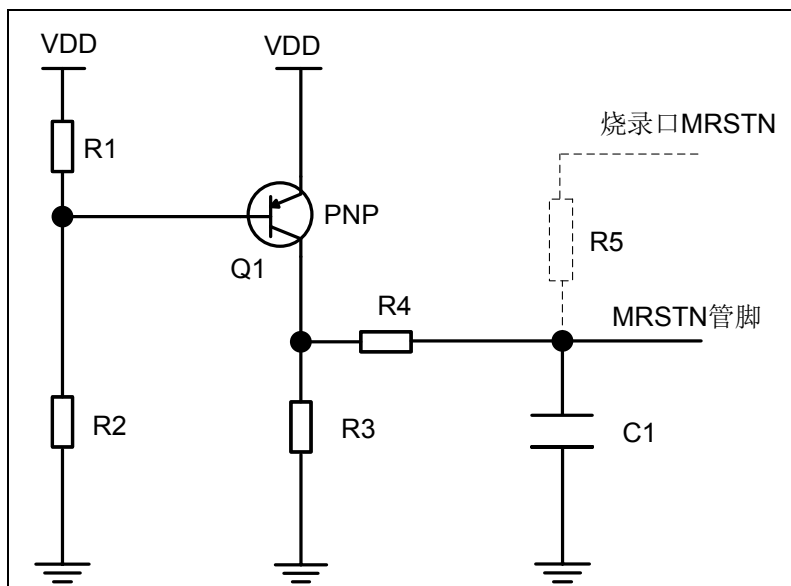


图 2-6 MRSTN 复位参考电路图 2

注：采用 PNP 三极管复位，通过 R1 和 R2 分压作为基极输入，发射极接 VDD，集电极一路通过 R3 (20KΩ) 接地，另一路通过 R4 ( $47K\Omega \leq R4 \leq 100K\Omega$ ) 和 C1 (0.1μF) 接地，C1 另一端作为 MRSTN 输入，当 VDD 为 3.3v 时，建议 R1 为 4.7KΩ，R2 为 8KΩ；VDD 为 5v 时，建议 R1 为 1.5KΩ，R2 为 5.1KΩ；在有外部烧录口的应用系统中，需要串接 R5 作为限流电阻， $0.1K\Omega \leq R5 \leq 1K\Omega$ 。

### 2.3.5 特殊功能寄存器

#### 复位状态寄存器 (SCU\_PWRC)

偏移地址: 08<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_0000X11<sub>B</sub>

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

—	bit31-8	—	—
POR_LOST	bit7	R/W	<b>POR 丢失标志位</b> 0: 无 POR 丢失 1: 有 POR 丢失
SOFT_RSTF	bit6	R/W	<b>软件复位标志位</b> 0: 无软件复位 1: 有软件复位
MRSTF	bit5	R/W	<b>MRSTN 复位标志位</b> 0: 无 MRSTN 复位 1: 有 MRSTN 复位
WDTRSTF	bit4	R/W	<b>WDT 复位标志位</b> 0: 无 WDT 复位 1: 有 WDT 复位
BORF	bit3	R/W	<b>BOR 复位标志位</b> 0: 无 BOR 复位 1: 有 BOR 复位
PORF	bit2	R/W	<b>POR 总复位标志位</b> 0: 无 POR 复位 (5V 或 1.5V) 1: 有 POR 复位 (5V 或 1.5V)
PORRCF	bit1	R/W	<b>1.5V POR 复位标志位</b> 0: 无 1.5V POR 复位 1: 有 1.5V POR 复位
PORVF	bit0	R/W	<b>5V POR 复位标志位</b> 0: 无 5V POR 复位 1: 有 5V POR 复位

注：对 SCU\_PWRC 寄存器进行写操作前，需要设置 SCU\_PROT 寄存器，关闭写保护。为避免误操作关键控制位信息，建议采用“字节”写方式操作 SCU\_PWRC 寄存器，且 bit7~bit0 只支持写“0”清除。

## 2.4 低电压监测 (LVD)

### 2.4.1 概述

芯片内置 LVD 低电压监测模块。LVD 模块监测 VDD 或 LVDIN 模拟通道，监测 VDD 电压时阈值可选，监测 LVDIN 电压时阈值固定为 1.25V。触发条件可选择掉电触发或上电触发，触发后产生 LVD 中断标志，当 LVD 中断使能开启时产生 LVD 中断请求。在睡眠模式下此中断可唤醒芯片。

### 2.4.2 特殊功能寄存器

低电压监测控制寄存器 (SCU_LVDC)															
偏移地址: 10 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDO	保留	LVDIFS<2:0>			LVDIE	LVDIF	LVDVS<3:0>			保留	LVD_FLTEN	LV DEN			

—	bit31-16	—	—
LVDO	bit15	R	<b>LVD 输出状态位</b> 0: 被监测电压高于电压阈值 1: 被监测电压低于电压阈值
—	bit14-13	—	—
LVDIFS<2:0>	bit12-10	R/W	<b>LVD 中断标志产生模式选择位</b> 000: LVD 输出状态位 (LVDO) 上升沿产生中断 001: LVD 输出状态位 (LVDO) 下降沿产生中断 010: LVD 输出状态位 (LVDO) 高电平产生中断 011: LVD 输出状态位 (LVDO) 低电平产生中断 1xx: LVD 输出状态位 (LVDO) 变化 (上升或下降沿) 产生中断
LVDIE	bit9	R/W	<b>LVD 中断使能位</b> 0: 禁止 1: 使能
LVDIF	bit8	R/W	<b>LVD 中断标志位</b> 0: 未发生 LVD 触发事件 1: 发生 LVD 触发事件 边沿模式产生中断标志时，可以写 1 清除标志； 电平模式产生中断标志时，该标志只读，触发电平消失后，中断标志自动清零。
LVDVS<3:0>	bit7-4	R/W	<b>LVD 触发电压 (电压为设计值)</b> 0000: 2.4V 0001: 2.6V 0010: 2.8V 0011: 3.0V 0100: 3.2V 0101: 3.4V 0110: 3.7V 0111: 3.9V 1000: 4.1V 1001: 4.3V

			1010: 4.6V 1011: 4.8V 1100: 5.0V 1101: 5.2V 111x: 监测 LVDIN, 阈值电压 1.25V
—	bit3-2	—	—
LVD_FLTEN	bit1	R/W	<b>LVD 滤波使能位</b> 0: 禁止 1: 使能
LV DEN	bit0	R/W	<b>LVD 使能位</b> 0: 禁止 1: 使能

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

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

## 2.5 系统低功耗操作模式

### 2.5.1 概述

配置外设时钟控制寄存器 SCU\_PCLKEN0 和 SCU\_PCLKEN1，可分别关闭芯片各个外设功能模块电路的时钟，使该部分电路功耗降到最低。

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

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

芯片进入休眠状态后，时钟工作状态参考下表：

时钟	睡眠模式	深度睡眠模式
LOSC	工作（当配置字 CFG_WORD0 的 LOSCEN 位使能时）	工作（当配置字 CFG_WORD0 的 LOSCEN 位使能时）
HOSC	工作（若使能）	关闭
HRC	工作（若使能）	关闭
LRC	工作（若使能）	工作

表 2-1 低功耗模式时钟状态列表

注：LOSC 运行状态由配置字 CFG\_WORD0 的 LOSCEN（LOSC 使能/禁止位）来决定，当该位使能时 LOSC 工作，当该位禁止时 LOSC 不工作。

### 2.5.2 睡眠模式

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

芯片进入睡眠模式的步骤如下：

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

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

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

### 2.5.3 深度睡眠模式

在深度睡眠模式下，芯片内核时钟停止，指令停止运行，系统主时钟振荡器停止工作（LOSC 和 LRC 除外）。可通过复位或中断唤醒深度睡眠模式。

芯片进入深度睡眠模式的步骤如下：

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

在深度睡眠模式下，外设功能模块（除支持深度睡眠唤醒模块外）都停止工作。深度睡眠模式下不访问存储器系统，相关控制器和内部总线。

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

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

### 2.5.4 唤醒方式

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

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

### 2.5.5 唤醒时间

芯片从睡眠模式唤醒，无等待时间，直接进入正常运行模式。

芯片从深度睡眠模式唤醒，在系统时钟振荡器起振后，还需要等待一段时间，时钟振荡器稳定工作后，才会进入正常运行模式，该等待时间与芯片主系统时钟源相关：

- 1) 主系统时钟源为 HRC 时，等待时间为 2048 个 HRC 时钟周期；
- 2) 主系统时钟源为 PLLCLK 时，在 PLL 锁频稳定后，等待时间为 256 个 PLL 时钟周期；
- 3) 主系统时钟源为 HOSC 时，等待时间由 HOSC 振荡器稳定时间选择位 HOSC\_TS 进行设置；
- 4) 主系统时钟源为 LRC 或 LOSC 时，无等待时间。

### 2.5.6 FLASH存储器等待功能

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

芯片支持增加 FLASH 存储器等待时间的设置，在不降低系统时钟频率的前提下，降低 FLASH 存储器取指令或数据的频率，从而降低整体芯片功耗。配置 SCU\_FLASHW<3:0>，可设定 FLASH 访问的等待时间。

### 2.5.7 特殊功能寄存器

FLASH 访问等待时间寄存器 (SCU_FLASHW)															
偏移地址: 20H															
复位值: 00000000_00000000_00000000_00000000 <sub>8</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												ACCT<3:0>			
—				bit31-4				—				—			
ACCT<3:0>				bit3-0				R/W				<b>FLASH 读取访问等待时间设置位</b> 0: 2T <sub>CLK</sub> 完成 FLASH 读取 1: 3T <sub>CLK</sub> 2: 4T <sub>CLK</sub> ...			

			F: 17T <sub>CLK</sub>
--	--	--	-----------------------

注 1: 对 SCU\_FLASHW 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。  
注 2: T<sub>CLK</sub> 时钟周期与芯片系统时钟周期相同。

## 2.6 系统时钟

### 2.6.1 时钟源概述

- ◇ LOSC: 32.768KHz 晶体振荡器 (支持外部 32.768KHz 时钟输入) 作为 RTC 时钟源 (匹配电容内置), 可配置为系统时钟源
- ◇ HOSC: 1~4MHz 晶体振荡器 (XT 模式) 或 8~20MHz 晶体振荡器 (HS 模式), 支持外部时钟输入, 可配置为系统时钟源
- ◇ HRC: 内部 16MHz RC 时钟振荡器可配置为系统时钟源, 出厂校准精度 ±1% (32.768KHz 晶体振荡器工作时, 支持实时调校, 确保全工作范围时钟精度)
- ◇ LRC: 内部 32KHz RC 时钟振荡器用于 WDT 时钟源, 可配置为系统时钟源
- ◇ PLL: 内部集成锁相环电路 PLL, 倍频时钟源可选择外部 4MHz、LOSC、内部 4MHz (HRC/4) 或 LRC 时钟源。当选择时钟源为 LOSC 或 LRC 时, 倍频至约 33.55MHz (32.768KHz\*1024 = 33.554432MHz); 当选择时钟源为外部或内部 4MHz 时, 倍频至 32MHz (4MHz\*8 = 32MHz) 时钟输出。

### 2.6.2 结构框图

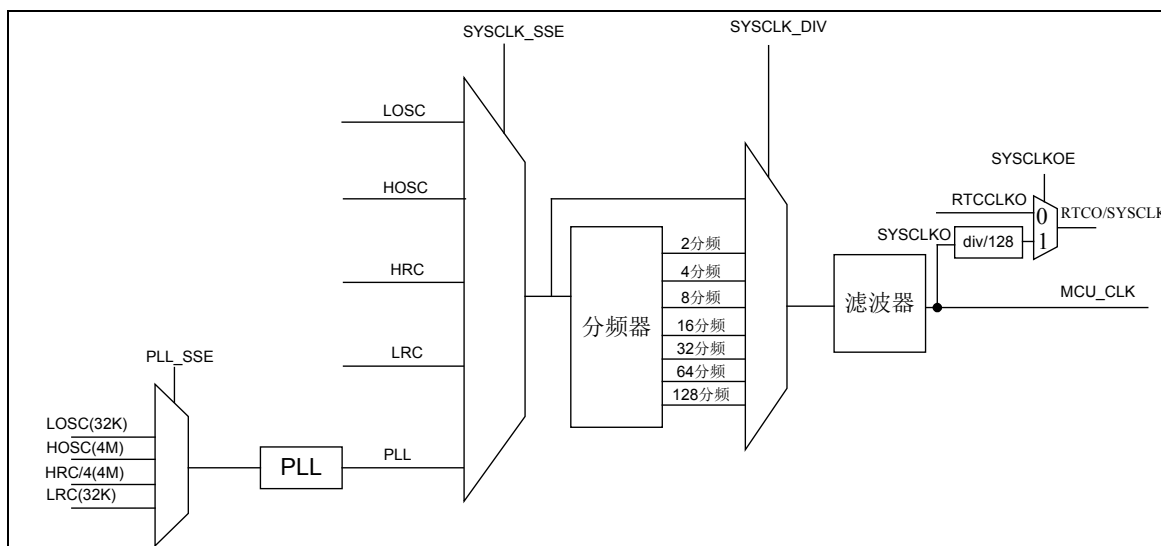


图 2-7 系统时钟电路结构框图

### 2.6.3 时钟源

#### LOSC

通过 LOSC0 和 LOSC1 端口外接 32.768 KHz 晶振, 作为 LOSC 振荡器。芯片内置晶振匹配电容, 用户无需外接匹配电容, 但需要外接反馈电阻。

LOSC 可用于实时时钟 RTC 模块、ADC 模块、LCD 模块和 WDT 模块的时钟源, 也可配置作为芯片的主系统时钟源。

LOSC 振荡器电路示意图如下:



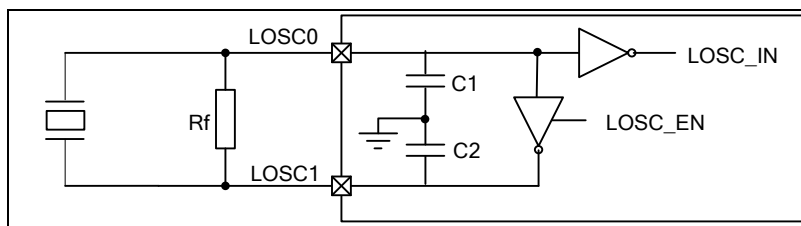


图 2-8 LOSC 振荡器电路示意图

注：电阻 Rf 推荐取值为 10MΩ。

### HOSC

芯片配置字控制位 HOSCM=00 时，选择外部时钟振荡器 HS 模式，此时 PB16 和 PB17 端口被强制用作 HOSC0 和 HOSC1 晶振端口。通过 HOSC0 和 HOSC1 端口外接 8M~20MHz 晶振，作为 HOSC 振荡器，需外接匹配电容。

芯片配置字控制位 HOSCM=01 时，选择外部时钟振荡器 XT 模式，此时 PB16 和 PB17 端口被强制用作 HOSC0 和 HOSC1 晶振端口。通过 HOSC0 和 HOSC1 端口外接 1M~4MHz 晶振，作为 HOSC 振荡器，需外接匹配电容。

芯片配置字控制位 HOSCM=10 时，可从 HOSC0 端口输入外部时钟，HOSC1 端口用作普通 I/O 端口。

芯片配置字控制位 HOSCM=11 时，HOSC0 和 HOSC1 都用作普通 I/O 端口。

HOSC 可配置作为芯片的主系统时钟源。

HOSC 振荡器电路示意图如下：

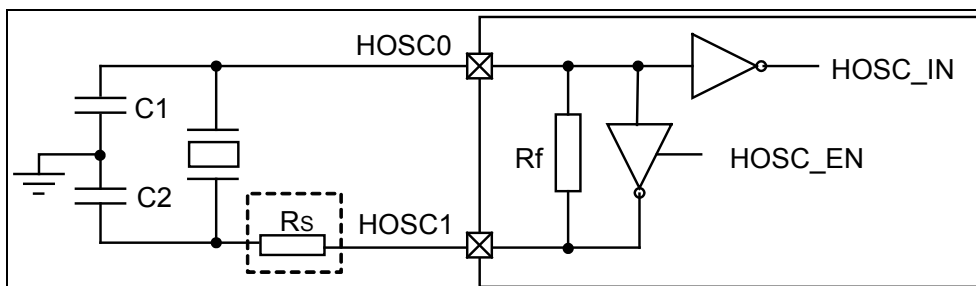


图 2-9 HOSC 振荡器电路示意图

注：其中 Rf 内部电路已集成，阻值约 1MΩ；电阻 Rs 有减小时钟谐波，提升 EMI 性能的作用，也可减小晶振过驱，延长晶振使用寿命，其为可选配置，具体阻值可按照晶振规格书的建议进行配置。

Osc Type	晶振频率	C1*	C2*
XT	1MHz	15 ~ 33pF	15 ~ 33pF
	2MHz		
	4MHz		
HS	8MHz	15pF	15pF
	16MHz		
	20MHz		

表 2-1 晶体振荡器电容参数参考表

### HRC

芯片内置 16MHz RC 时钟振荡器，出厂校准精度 ±1% (VDD=5V, 温度 25℃)。

当外部 RTC 时钟振荡器 LOSC 工作时，可通过配置字控制位 HRCCEB，使能 HRC 实时调校功能，调校精度±1%（精度与 LOSC 振荡器自身的精度有关）。

### LRC

芯片内置 32KHz RC 时钟振荡器，用作 WDT 计数时钟源，也可配置作为主系统时钟源。

### PLL

当 PLL 时钟源选择为 LOSC 或 LRC 时，倍频至约 33.55MHz（32.768KHz\*1024 = 33.554432MHz）；当选择时钟源为外部或内部 4MHz 时，倍频至 32MHz（4MHz\*8 = 32MHz）时钟输出。通过寄存器位 PLL\_SSE<2:0>可配置 PLL 时钟源，通过寄存器位 SYSCLK\_SSE<2:0>可配置 PLL 输出时钟作为主系统时钟。

## 2.6.4 系统时钟源切换

芯片上电复位后，HRC 默认作为系统时钟源。配置 SYSCLK\_SSE<2:0>，可切换系统时钟源，芯片内置时钟无缝切换电路，保证时钟切换稳定。但无缝切换电路会对时钟切换产生一定周期的切换延迟，SYSCLK\_CHG 标志位为 1 表示系统时钟正在切换中，建议用户此时不要更改系统时钟切换选择位 SYSCLK\_SSE<2:0>；SYSCLK\_CHG 标志位为 0 表示上一次系统时钟切换完成。

芯片初始化时，必须进行系统时钟源切换。

为保证系统工作稳定，芯片对系统时钟的切换有保护机制。即当选择切换的目标时钟源处于工作异常或不稳定状态时，即使时钟切换完成，当前采用的时钟源有可能仍与切换目标不一致，即本次时钟切换不成功。在切换完成后，可通过状态位 SYSCLK\_SST 查询当前工作状态下所采用的系统时钟源，以判断时钟切换是否成功。

用户每次写 SYSCLK\_SSE<2:0>位（包括不改变原有值）都会触发一次时钟切换动作。

## 2.6.5 系统时钟分频

系统时钟支持一个最大分频比为 1:128 的后分频器。

后分频器本身不可读写，配置系统时钟切换选择位 SYSCLK\_SSE<2:0>后，后分频器计数自动清零，但不影响分频比设置。

## 2.6.6 外设模块时钟配置

芯片各外设功能模块的工作时钟源，可以通过外设时钟控制寄存器 SCU\_PCLKEN0 和 SCU\_PCLKEN1 的各控制位使能或关闭，从而最大限度降低芯片功耗。

在写入和读取外设模块的特殊功能寄存器时，需软件使能对应外设模块的时钟，否则无法访问外设模块的特殊功能寄存器。

## 2.6.7 特殊功能寄存器

系统时钟控制寄存器 (SCU_SCLKEN0)															
偏移地址: 40 <sub>H</sub>															
复位值: 00000001_00000011_00100000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				LRC_ON	LRC_EN	保留				HRC_WAIT_S EL	HRC_ON	HRC_EN			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		SYSCLK _CHG	SYS CLKOE	保留		SYSCLK_ DIV<2:0>		保留		SYSCLK_ SST<2:0>		保留		SYSCLK_SSE<2:0>	
—				bit31-26				—				—			
LRC_ON				bit25				R				LRC 振荡器工作标志位 0: 关闭状态			

			1: 工作状态
LRC_EN	bit24	R/W	<b>LRC 振荡器使能位</b> (配置字 <b>WDTEN</b> 为 0 时有效) 0: 禁止 1: 使能
—	bit23-20	—	—
HRC_WAIT_SEL<1:0>	bit19-18	R/W	<b>HRC 调校间隔时间设置位</b> 00: 8 个 RTC 时钟周期 01: 32 个 RTC 时钟周期 10: 128 个 RTC 时钟周期 11: 512 个 RTC 时钟周期
HRC_ON	bit17	R	<b>HRC 振荡器工作标志位</b> 0: 关闭状态 1: 开启状态
HRC_EN	bit16	R/W	<b>HRC 振荡器使能位</b> 0: 禁止 1: 使能
—	bit15-14	—	—
SYSClk_CHG	bit13	R	<b>系统时钟切换标志位</b> 0: 切换完成 1: 切换进行中
SYSClkOE	bit12	R/W	<b>系统时钟 128 分频输出使能位</b> 0: 禁止 1: 使能 (通过 SYSClKO 端口输出)
—	bit11	R/W	—
SYSClk_DIV<2:0>	bit10-8	R/W	<b>系统时钟后分频选择位</b> 000: 1:1 001: 1:2 010: 1:4 011: 1:8 100: 1:16 101: 1:32 110: 1:64 111: 1:128
—	bit7	—	—
SYSClk_SST<2:0>	bit6-4	R	<b>系统时钟源标志位</b> 000: HRC 001: LRC 010: HOSC 011: PLLCLK 100: LOSC 101~111: 保留
—	bit3	—	—
SYSClk_SSE<2:0>	bit2-0	R/W	<b>系统时钟切换选择位</b> 000: HRC (内部 16MHz RC 时钟) 001: LRC (内部 32KHz RC 时钟) 010: HOSC (外部 2M~20MHz 主时钟) 011: PLLCLK 100: LOSC (外部 32.768KHz RTC)

		时钟) 101~111: 保留
--	--	--------------------

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

**系统时钟设置寄存器 (SCU\_SCLKEN1)**

偏移地址: 44<sub>H</sub>

复位值: 00000000\_00000000\_10000000\_00000100<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				WDT32K_SEL	LCD32K_SEL	ADC32K_SEL	PLL_LOCK	PLL_SSE<2:0>	PLL_ULOCK_SET<1:0>	PLL_RDY	PLL_EN				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOSC_LPEN	LOSC_RDY	保留										HOSC_TS<1:0>	HOSC_RDY	HOSC_EN	

—	bit31-27	—	—
WDT32K_SEL	bit26	R/W	<b>WDT 32K 时钟选择位</b> 0: LRC 1: LOSC
LCD32K_SEL	bit25	R/W	<b>LCD 32K 时钟选择位</b> 0: LRC 1: LOSC
ADC32K_SEL	bit24	R/W	<b>ADC 32K 时钟选择位</b> 0: LRC 1: LOSC
PLL_LOCK	bit23	R	<b>PLL 锁频标志</b> 0: PLL 失锁 1: 锁频信号稳定
PLL_SSE<2:0>	bit22-20	R/W	<b>PLL 时钟源选择 (PLL_EN 为 0 时可写)</b> 000: LRC(内部 32KHz) 001: LOSC(外部 32KHz) 010: 保留 011: 保留 100: HRC/4 (内部 4MHz) 101: HOSC/4 (外接 16MHz 晶振) 110: HOSC (外接 4MHz 晶振) 111: 保留
PLL_ULOCK_SET<1:0>	bit19-18	R/W	<b>PLL 作为系统时钟时失锁处理</b> 00: 禁止, 无任何处理 01: PLL 时钟门控禁止, 无系统时钟, 等重新锁频后, 提供时钟 10: 系统时钟切换成 HRC 11: 系统时钟切换成 HRC, PLL 锁频后自动切回
PLL_RDY	bit17	R	<b>PLL 稳定标志</b> 0: 不稳定 1: 稳定
PLL_EN	bit16	R/W	<b>PLL 使能</b> 0: 禁止 1: 使能
LOSC_LPEN	bit15	R/W	<b>LOSC 低功耗模式</b>

			0: 禁止 1: 使能
LOSC_RDY	bit14	R	<b>LOSC 振荡器稳定标志位</b> 0: 不稳定 1: 稳定
—	bit13-4	—	—
HOSC_TS<1:0>	bit3-2	R/W	<b>HOSC 振荡器稳定时间选择位</b> (HOSC_EN=0 时可写) 00: 512 个外部时钟周期 01: 1024 个外部时钟周期 10: 2048 个外部时钟周期 11: 保留
HOSC_RDY	bit1	R	<b>HOSC 振荡器稳定标志位</b> 0: 不稳定 1: 稳定
HOSC_EN	bit0	R/W	<b>HOSC 振荡器使能位</b> 0: 禁止 1: 使能 (HOSC 振荡器配置为 HS 或 XT 模式时有效)

注 1: 对 SCU\_SCLKEN1 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。  
 注 2: 在时钟管脚无外接晶振或存在外部干扰的情况下, 外部时钟振荡器稳定标志位 HOSC\_RDY 可能会被误置 1。  
 注 3: PLL\_RDY/PLL\_LOCK 作为 PLL 的锁频标志信号, 仅供应用参考, 在高可靠应用中, 为防止 PLL 锁频标志信号失效而无法退出等待状态, 需要在等待 PLL 锁频标志信号的同时设定等待溢出时间, 建议设置等待溢出时间为 10ms, 在等待溢出时间到达时, 如仍未检测到 PLL 锁频标志信号置起, 则退出等待状态, 此时 PLL 已实际锁频稳定。

外设时钟控制寄存器 0 (SCU_PCLKEN0)															
偏移地址: 48 <sub>H</sub>															
复位值: 00000000_00000000_00000000_10000011 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								WDT	LCDC	RTC	ADC	保留	IAP	GPIO	SCU

—	bit31-8	—	—
WDT	bit7	R/W	<b>WDT 时钟使能位</b> 0: 禁止 1: 使能
LCDC	bit6	R/W	<b>LCDC 时钟使能位</b> 0: 禁止 1: 使能
RTC	bit5	R/W	<b>RTC 时钟使能位</b> 0: 禁止 1: 使能
ADC	bit4	R/W	<b>ADC 时钟使能位</b> 0: 禁止 1: 使能
—	bit3	—	—
IAP	bit2	R/W	<b>FLASH_IAP 时钟使能位</b> 0: 禁止 1: 使能

GPIO	bit1	R/W	<b>GPIO 时钟使能位</b> 0: 禁止 1: 使能
SCU	bit0	R	<b>SCU 时钟使能位</b> 读取该位始终为 1

注 1: 对 SCU\_PCLKEN0 寄存器进行写操作前, 需要设置 SCU\_PROT 寄存器, 关闭写保护。  
注 2: WDT 时钟使能位 WDT=0 时, 对 WDT 模块的寄存器读写操作被禁止, 但 WDT 计数器仍保持工作状态, 看门狗定时器功能仍有效。

**外设时钟控制寄存器 1 (SCU\_PCLKEN1)**

偏移地址: 4C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	I2C	保留	保留	保留	SPI	保留	保留	保留	保留	EUART1	EUART0	UART3	UART2	UART1	UART0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	保留	T32N2	T32N1	T32N0	保留	保留	T16N3	T16N2	T16N1	T16N0

—	bit31-29	—	—
I2C	bit28	R/W	<b>I2C 时钟使能位</b> 0: 禁止 1: 使能
—	bit27-25	—	—
SPI	bit24	R/W	<b>SPI 时钟使能位</b> 0: 禁止 1: 使能
—	bit23-22	—	—
EUART1	bit21	R/W	<b>EUART1 时钟使能位</b> 0: 禁止 1: 使能
EUART0	bit20	R/W	<b>EUART0 时钟使能位</b> 0: 禁止 1: 使能
UART3	bit19	R/W	<b>UART3 时钟使能位</b> 0: 禁止 1: 使能
UART2	bit18	R/W	<b>UART2 时钟使能位</b> 0: 禁止 1: 使能
UART1	bit17	R/W	<b>UART1 时钟使能位</b> 0: 禁止 1: 使能
UART0	bit16	R/W	<b>UART0 时钟使能位</b> 0: 禁止 1: 使能
—	bit15-11	—	—
T32N2	bit10	R/W	<b>T32N2 时钟使能位</b> 0: 禁止 1: 使能
T32N1	bit9	R/W	<b>T32N1 时钟使能位</b> 0: 禁止

			1: 使能
T32N0	bit8	R/W	<b>T32N0 时钟使能位</b> 0: 禁止 1: 使能
—	bit7-4	—	—
T16N3	bit3	R/W	<b>T16N3 时钟使能位</b> 0: 禁止 1: 使能
T16N2	bit2	R/W	<b>T16N2 时钟使能位</b> 0: 禁止 1: 使能
T16N1	bit1	R/W	<b>T16N1 时钟使能位</b> 0: 禁止 1: 使能
T16N0	bit0	R/W	<b>T16N0 时钟使能位</b> 0: 禁止 1: 使能

注：对 SCU\_PCLKEN1 寄存器进行写操作前，需要设置 SCU\_PROT 寄存器，关闭写保护。

## 2.7 中断和异常处理

### 2.7.1 中断和异常

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

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

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

异常/中断优先级操作说明：

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

表 2-2 异常/中断优先级操作类型说明列表

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

异常/中断优先级：

编号	类型	优先级	简介
0	N/A	N/A	没有异常在运行
1	复位	-3 (最高)	复位
2	NMI	-2	不可屏蔽中断 (来自外设 NMI 中断输入)
3	Hard Fault	-1	所有被禁用的 Fault，都将升级为 Hard Fault
4~10	保留	NA	-
11	SVC	可编程控制	系统服务调用
12~13	保留	NA	-
14	PendSV	可编程控制	为系统设备而设的“可悬挂请求”
15	SysTick	可编程控制	系统定时计数器
16	IRQ0	可编程控制	外设中断 0
17	IRQ1	可编程控制	外设中断 1
...	...	...	...
47	IRQ31	可编程控制	外设中断 31

表 2-3 异常/中断优先级列表

Cortex-M0 支持如下异常/中断：



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

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

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

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

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

### 2.7.2 中断和异常向量的分配

编号	类型	功能	说明
0~15	异常	—	Cortex-M0 内核异常，包括 NMI 不可屏蔽中断
16	IRQ0	PINT0 中断	外部端口中断 0
17	IRQ1	PINT1 中断	外部端口中断 1
18	IRQ2	PINT2 中断	外部端口中断 2
19	IRQ3	PINT3 中断	外部端口中断 3
20	IRQ4	PINT4 中断	外部端口中断 4
21	IRQ5	PINT5 中断	外部端口中断 5
22	IRQ6	PINT6 中断	外部端口中断 6
23	IRQ7	PINT7 中断	外部端口中断 7
24	IRQ8	T16N0 中断	16 位定时器/计数器 0 中断
25	IRQ9	T16N1 中断	16 位定时器/计数器 1 中断
26	IRQ10	T16N2 中断	16 位定时器/计数器 2 中断
27	IRQ11	T16N3 中断	16 位定时器/计数器 3 中断
28	IRQ12	T32N0 中断	32 位定时器/计数器 0 中断
29	IRQ13	T32N1 中断	32 位定时器/计数器 1 中断
30	IRQ14	T32N2 中断	32 位定时器/计数器 2 中断
31	IRQ15	Reserved	预留
32	IRQ16	WDT 中断	看门狗中断
33	IRQ17	RTC 中断	实时时钟中断
34	IRQ18	Reserved	预留
35	IRQ19	ADC 中断	模数转换中断
36	IRQ20	LCD 中断	液晶显示中断
37	IRQ21	LVD 中断	低电压监测 0 中断
38	IRQ22	Reserved	预留
39	IRQ23	UART0 中断	UART0 中断
40	IRQ24	UART1 中断	UART1 中断
41	IRQ25	UART2 中断	UART2 中断
42	IRQ26	UART3 中断	UART3 中断
43	IRQ27	EUART0 中断	EUART0 中断
44	IRQ28	EUART1 中断	EUART1 中断
45	IRQ29	SPI 中断	SPI 中断

编号	类型	功能	说明
46	IRQ30	I2C 中断	I2C 中断
47	IRQ31	Reserved	预留

表 2-4 IRQ 分配列表

### 2.7.3 中断向量的重映射

Cortex-M0 核心本身并不支持中断向量的重映射，在 HR8P296 芯片中有两个特殊功能寄存器“中断向量表重映射使能寄存器”和“中断向量表偏移寄存器”，可以支持中断向量的重映射。具体的使用方式可参考 FLASH 自编程（IAP）相关章节的描述。

### 2.7.4 特殊功能寄存器

NMI 控制寄存器 (SCU_NMIC)															
偏移地址: 04 <sub>H</sub>															
复位值: 00000000_00000000_00000001_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											NMICS<4:0>			NMIEN	

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

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

中断向量表重映射使能寄存器 (SCU_TBLRMEN)															
偏移地址: 60 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															TBLREMAPEN

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

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

**中断向量表偏移寄存器 (SCU\_TBLOFFS)**

偏移地址: 64<sub>H</sub>

复位值: 00100000\_00000000\_00000001\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBLOFF<31:8>															

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

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

注：该地址为起始地址，需满足如下要求：必须先求出系统中共有多少个向量，再把这个数字向上增大到是 2 的整次幂，而起始地址必须对齐到后者的边界上。如果一共有 32 个中断，则共有 32+16（系统异常）=48 个向量，向上增大到 2 的整次幂后值为 64，因此地址必须能被 64×4=256 整除，从而合法的起始地址可以是：0x000，0x100，0x200 等。

以下 NVIC 寄存器列表及基址参见：3.5.2 中断控制器（NVIC）寄存器列表。

**IRQ0~31 置中断请求使能寄存器 (NVIC\_ISER)**

偏移地址: 00<sub>H</sub>

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

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

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

SETENA<31:0>	bit31-0	R/W	<p><b>IRQ 使能位</b></p> <p>0: 中断使能无效</p> <p>1: 中断使能有效</p> <p>软件写 1 使能中断请求，写 0 无效</p>
--------------	---------	-----	--

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

**IRQ0~31 清中断请求使能寄存器 (NVIC\_ICER)**

偏移地址: 80<sub>H</sub>

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

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

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

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

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

**IRQ0~31 置中断挂起寄存器 (NVIC\_ISPR)**

偏移地址: 00<sub>H</sub>

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

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

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

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

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

**IRQ0~31 清中断挂起寄存器 (NVIC\_ICPR)**

偏移地址: 80<sub>H</sub>

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

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

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

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

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

**IRQ0~3 优先级控制寄存器 (NVIC\_PR0)**

偏移地址: 00<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_3<1:0>				保留				PRI_2<1:0>				保留			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_1<1:0>				保留				PRI_0<1:0>				保留			

PRI_3<1:0>	bit31-30	R/W	<b>IRQ3 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit29-24	—	—
PRI_2<1:0>	bit23-22	R/W	<b>IRQ2 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级

—	bit21-16	—	—
PRI_1<1:0>	bit15-14	R/W	<b>IRQ1 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit13-8	—	—
PRI_0<1:0>	bit7-6	R/W	<b>IRQ0 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit5-0	—	—

**IRQ4~7 优先级控制寄存器 (NVIC\_PR1)**

偏移地址: 04<sub>H</sub>

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

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

PRI_7<1:0>	bit31-30	R/W	<b>IRQ7 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit29-24	—	—
PRI_6<1:0>	bit23-22	R/W	<b>IRQ6 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit21-16	—	—
PRI_5<1:0>	bit15-14	R/W	<b>IRQ5 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit13-8	—	—
PRI_4<1:0>	bit7-6	R/W	<b>IRQ4 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit5-0	—	—

**IRQ8~11 优先级控制寄存器 (NVIC\_PR2)**

偏移地址: 08<sub>H</sub>

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

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

PRI_11<1:0>	bit31-30	R/W	<b>IRQ11 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit29-24	—	—
PRI_10<1:0>	bit23-22	R/W	<b>IRQ10 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit21-16	—	—
PRI_9<1:0>	bit15-14	R/W	<b>IRQ9 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit13-8	—	—
PRI_8<1:0>	bit7-6	R/W	<b>IRQ8 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit5-0	—	—

**IRQ12~15 优先级控制寄存器 (NVIC\_PR3)**

偏移地址: 0C<sub>H</sub>

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

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

PRI_15<1:0>	bit31-30	R/W	<b>IRQ15 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit29-24	—	—
PRI_14<1:0>	bit23-22	R/W	<b>IRQ14 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级

—	bit21-16	—	11: 最低优先级
PRI_13<1:0>	bit15-14	R/W	<b>IRQ13 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit13-8	—	—
PRI_12<1:0>	bit7-6	R/W	<b>IRQ12 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit5-0	—	—

**IRQ16~19 优先级控制寄存器 (NVIC\_PR4)**

偏移地址: 10<sub>H</sub>

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

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

PRI_19<1:0>	bit31-30	R/W	<b>IRQ19 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit29-24	—	—
PRI_18<1:0>	bit23-22	R/W	<b>IRQ18 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit21-16	—	—
PRI_17<1:0>	bit15-14	R/W	<b>IRQ17 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit13-8	—	—
PRI_16<1:0>	bit7-6	R/W	<b>IRQ16 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit5-0	—	—

**IRQ20~23 优先级控制寄存器 (NVIC\_PR5)**

偏移地址: 14<sub>H</sub>

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

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

PRI_23<1:0>	bit31-30	R/W	<b>IRQ23 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit29-24	—	—
PRI_22<1:0>	bit23-22	R/W	<b>IRQ22 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit21-16	—	—
PRI_21<1:0>	bit15-14	R/W	<b>IRQ21 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit13-8	—	—
PRI_20<1:0>	bit7-6	R/W	<b>IRQ20 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit5-0	—	—

**IRQ24~27 优先级控制寄存器 (NVIC\_PR6)**

偏移地址: 18<sub>H</sub>

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

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

PRI_27<1:0>	bit31-30	R/W	<b>IRQ27 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit29-24	—	—
PRI_26<1:0>	bit23-22	R/W	<b>IRQ26 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级



—	bit21-16	—	11: 最低优先级
PRI_25<1:0>	bit15-14	R/W	<b>IRQ25 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit13-8	—	—
PRI_24<1:0>	bit7-6	R/W	<b>IRQ24 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit5-0	—	—

**IRQ28~31 优先级控制寄存器 (NVIC\_PR7)**

偏移地址: 1C<sub>H</sub>

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

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

PRI_31<1:0>	bit31-30	R/W	<b>IRQ31 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit29-24	—	—
PRI_30<1:0>	bit23-22	R/W	<b>IRQ30 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit21-16	—	—
PRI_29<1:0>	bit15-14	R/W	<b>IRQ29 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit13-8	—	—
PRI_28<1:0>	bit7-6	R/W	<b>IRQ28 优先级设置位</b> 00: 最高优先级 01: 次高优先级 10: 次低优先级 11: 最低优先级
—	bit5-0	—	—

## 2.8 系统控制块 (SCB)

### 2.8.1 概述

系统控制块提供芯片内核系统实现的状态信息，并对内核系统工作进行控制。

SCB 寄存器列表和基址参见：3.5.3 系统控制块 (SCB) 寄存器列表。

### 2.8.2 特殊功能寄存器

CPUID 寄存器 (CPUID)															
偏移地址: 00 <sub>H</sub>															
复位值: 01000001_00001100_11000010_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMPLEMENTER<7:0>							VARIANT<3:0>					CONSTANT<3:0>			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARTNO<11:0>											REVISION<3:0>				

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

中断控制和状态寄存器 (ICSR)															
偏移地址: 04 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NMIPEN DSET	保留	PENDS VSET	PENDS VCLR	PENDS TSET	PENDS TCLR	保留	保留	ISR PENDING	保留	保留	保留	保留	保留	保留	VECTPENDING G<5:4>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VECTPENDING<3:0>			保留							VECTACTIVE<5:0>					

NMIPENDSET	bit31	R/W	<b>NMI 中断挂起控制位</b> 0: 不置 NMI 中断挂起 1: 置 NMI 中断挂起
—	bit30-29	R	—
PENDSVSET	bit28	R/W	<b>置 PendSV 异常挂起位</b> 0: 无效 1: 置 PendSV 异常挂起
PENDSVCLR	bit27	R/W	<b>清 PendSV 异常挂起位</b> 0: 无效 1: 清除 PendSV 异常挂起
PENDSTSET	bit26	R/W	<b>置 SysTick 异常挂起位</b> 0: 无效 1: 置 SysTick 异常挂起
PENDSTCLR	bit25	R/W	<b>清 SysTick 异常挂起位</b>

—	bit24-23	R	0: 无效 1: 清除 SysTick 异常挂起
ISRPENDDING	bit22	R	中断挂起标志位 0: 无中断挂起 1: 有中断挂起
—	bit21-18	R	—
VECTPENDING<5:0>	bit17-12	R	当前的挂起中, 优先级最高的异常/中断号 0x0: 无挂起异常/中断 非 0: 当前被挂起的异常/中断中, 优先级最高的异常/中断号
—	bit11-6	R	—
VECTACTIVE<5:0>	bit5-0	R	当前被处理的异常/中断号 0x0: 线程 (Thread) 模式 非 0: 当前被处理的异常/中断号

**应用中断和复位控制寄存器 (AIRCR)**

偏移地址: 0C<sub>H</sub>

复位值: 11111010\_00000101\_00000000\_00000000<sub>B</sub>

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

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

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

**系统控制寄存器 (SCR)**

偏移地址: 10<sub>H</sub>

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

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

—	bit31-5	—	—
SEVONPEND	bit4	R/W	中断被挂起时, 是否作为唤醒事件的选择

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

**配置和控制寄存器 (CCR)**

偏移地址: 14<sub>H</sub>

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

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

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

**系统处理程序优先级寄存器 2 (SHPR2)**

偏移地址: 1C<sub>H</sub>

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

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

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

**系统处理程序优先级寄存器 3 (SHPR3)**

偏移地址: 20<sub>H</sub>

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

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

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

## 2.9 系统定时器 (SYSTICK)

### 2.9.1 概述

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

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

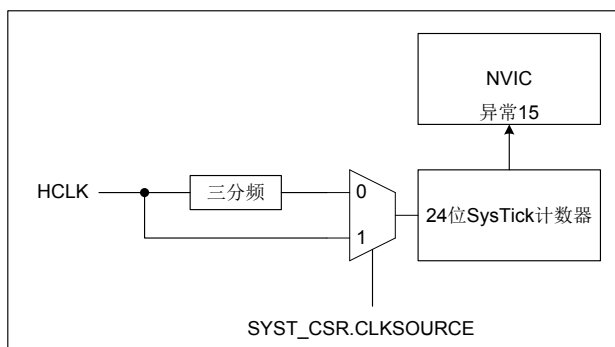


图 2-10 系统定时器框图

SysTick 的当前计数值可以通过读 SYST\_CVR 寄存器获得。如果写 SYST\_CVR 寄存器，则将该寄存器清零，并且将 COUNTFLAG 位清 0，写操作不会触发 SysTick 异常事件。

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

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

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

### 2.9.2 特殊功能寄存器

SYSTICK 控制和状态寄存器 (SYST_CSR)															
偏移地址: 10 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														COUNTFLAG	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CLKSOURCE	TICKINT	ENABLE	
—				bit31-17				—				—			
COUNTFLAG				bit16				R				SYSTICK 递减计数到零的标志位 0: 未计数到 0 1: 计数到 0			

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

注 1: 处理器时钟为芯片内核工作时钟 HCLK，时钟频率与系统时钟频率相同。  
注 2: SYSTICK 基准时钟，实际是处理器时钟 3 分频后的时钟，频率为 F<sub>HCLK</sub>/3。

**SYSTICK 重装值寄存器 (SYST\_RVR)**

偏移地址: 14<sub>H</sub>

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

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

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

**SYSTICK 当前值寄存器 (SYST\_CVR)**

偏移地址: 18<sub>H</sub>

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

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

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

**SYSTICK 校准值寄存器 (SYST\_CALIB)**

偏移地址: 1C<sub>H</sub>

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

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

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

注: 本产品只提供内部基准时钟, 其频率为  $F_{HCLK}/3$ 。

### 第3章 存储资源

#### 3.1 内部存储器地址映射

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

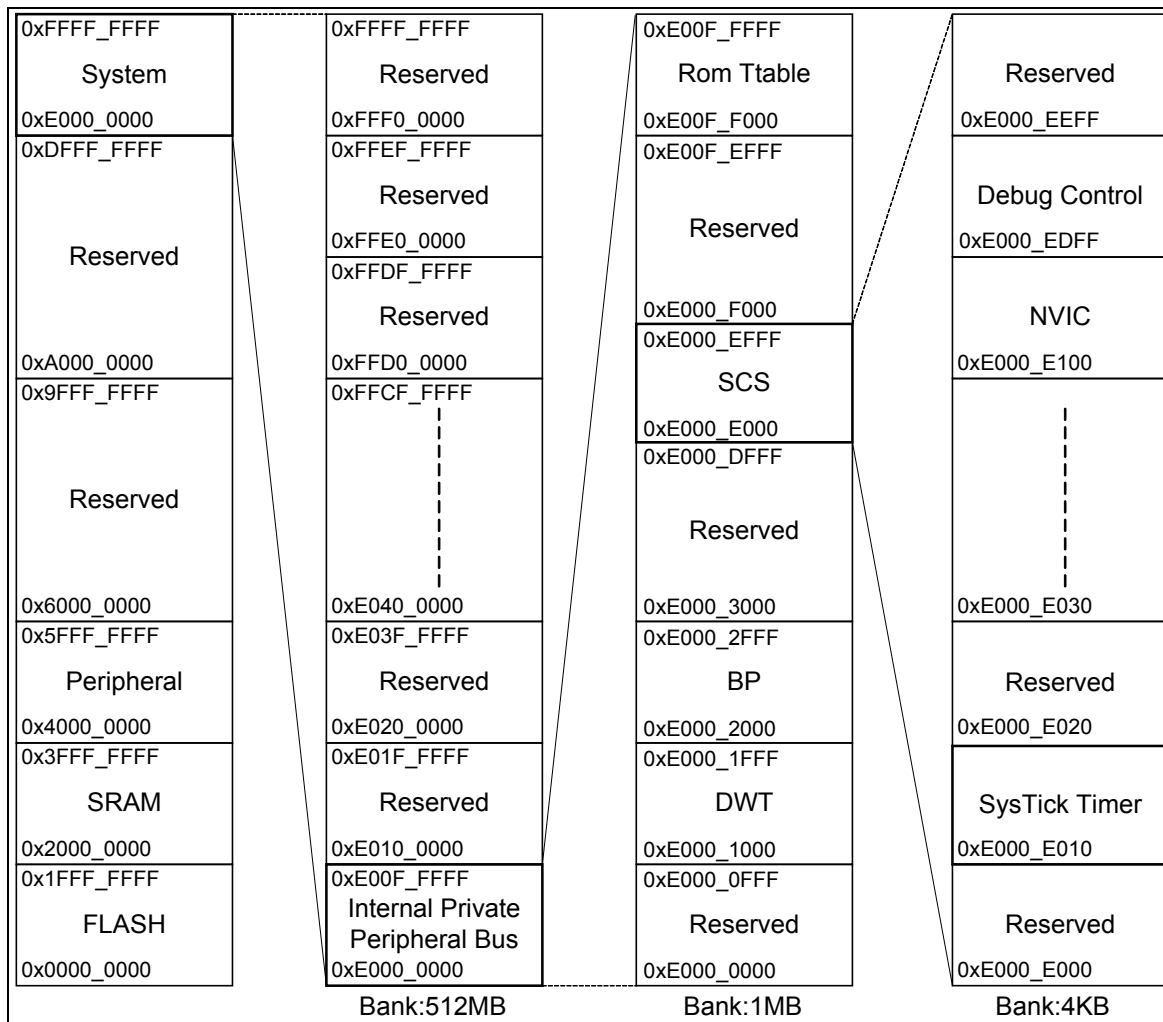


图 3-1 内部存储器地址映射



## 3.2 FLASH存储器

### 3.2.1 芯片配置字

#### 3.2.1.1 概述

芯片配置字位于 FLASH 存储器的信息区，用户可在 ISP 编程时进行设置。芯片的各种功能配置由芯片配置字和各功能相关寄存器共同设置完成。芯片配置字包括 HOSC 振荡器工作模式的选择、WDT 使能控制、低电压复位使能控制、BOR 电压选择、IAP 操作使能控制等。

CFG_WORD0								
Bit	31	30	29	28	27	26	25	24
Define	—	—	—	—	—	—	—	—
Bit	23	22	21	20	19	18	17	16
Define	—	—	—	—	—	—	—	—
Bit	15	14	13	12	11	10	9	8
Define	WDTRL<2:0>			WDTWIN<1:0>		WDTINTEN	LOSCEN	DBGEB
Bit	7	6	5	4	3	2	1	0
Define	PWRTEB	BOREN	BORV<1:0>		WDTEN	—	HOSCM<1:0>	

Bit 31-16 保留

Bit 15-13 WDTRL<2:0>: 上电复位 WDT 重载值

- 000: 0x0000\_0200 (选择 LRC 时钟源, 对应时间约 16ms)
- 001: 0x0000\_0400 (选择 LRC 时钟源, 对应时间约 32ms)
- 010: 0x0000\_1000 (选择 LRC 时钟源, 对应时间约 128ms)
- 011: 0x0000\_4000 (选择 LRC 时钟源, 对应时间约 512ms)
- 100: 0x0000\_8000 (选择 LRC 时钟源, 对应时间约 1s)
- 101: 0x0001\_0000 (选择 LRC 时钟源, 对应时间约 2s)
- 110: 0x0002\_0000 (选择 LRC 时钟源, 对应时间约 4s)
- 111: 0x0004\_0000 (选择 LRC 时钟源, 对应时间约 8s)

Bit 12-11 WDTWIN<1:0>: WDT 喂狗窗口选择位

- 00: 0-25% WDT 复位周期 (仅供测试使用, 禁止设置为 00)
- 01: 0-50% WDT 复位周期 (仅供测试使用, 禁止设置为 01)
- 10: 0-75% WDT 复位周期 (仅供测试使用, 禁止设置为 10)
- 11: 100% WDT 复位周期 (默认)

Bit 10 WDTINTEN: WDT 看门狗间隔中断使能位

- 0: 禁止
- 1: 使能

Bit 9 LOSCEN: LOSC 振荡器使能位

- 0: 禁止
- 1: 使能

Bit 8 DBG0EB: DEBUG0 端口使能位 (PA15 用于 ISCK, PA14 用于 ISDA)

- 0: 使能
- 1: 禁止

Bit 7 PWRTEB: 上电延时使能位

- 0: 使能
- 1: 禁止
- Bit 6      **BOREN: BOR 掉电复位使能位**
  - 0: 禁止
  - 1: 使能
- Bit 5-4   **BORV<1:0>: BOR 监测电压选择位**
  - 00: 1.8V
  - 01: 2.2V
  - 10: 3.0V
  - 11: 3.6V
- Bit 3      **WDTEN: WDT 看门狗使能位**
  - 0: 禁止
  - 1: 使能
- Bit 2      保留 (固定为 1)
- Bit 1-0   **HOSCM<1:0>: HOSC 振荡器工作模式选择位**
  - 00: HS 模式
  - 01: XT 模式
  - 10: HOSC0 输入系统时钟, HOSC1 用作普通 I/O
  - 11: 禁止振荡器, HOSC0 和 HOSC1 用作普通 I/O

CFG_WORD1									
Bit	31	30	29	28	27	26	25	24	
Define	—	—	—	—	—	—	—	—	
Bit	23	22	21	20	19	18	17	16	
Define	—	—	—	—	—	—	—	—	
Bit	15	14	13	12	11	10	9	8	
Define	—	—	—	—	—	—	—	—	
Bit	7	6	5	4	3	2	1	0	
Define	FWPEB	FWPS<4:0>					HRCCEB	DBG1EB	

- Bit 31-8   保留
- Bit 7      **FWPEB: FLASH 自编程操作 (IAP) 写保护区使能位**
  - 0: 使能
  - 1: 禁止
- Bit 6-2   **FWPS<4:0>: FLASH 自编程操作 (IAP) 写保护区选择**
  - 00000: FLASH 地址 (0000\_0000H~0000\_0FFFH) 为写保护区
  - 00001: FLASH 地址 (0000\_0000H~0000\_1FFFH) 为写保护区
  - 00010: FLASH 地址 (0000\_0000H~0000\_2FFFH) 为写保护区
  - 00011: FLASH 地址 (0000\_0000H~0000\_3FFFH) 为写保护区
  - :
  - :
  - 01101: FLASH 地址 (0000\_0000<sub>H</sub>~0000\_DFFF<sub>H</sub>) 为写保护区
  - 01110: FLASH 地址 (0000\_0000<sub>H</sub>~0000\_EFFF<sub>H</sub>) 为写保护区
  - 01111: FLASH 地址 (0000\_0000<sub>H</sub>~0000\_FFFF<sub>H</sub>) 为写保护区

- Bit 1 HRCCEB: HRC 自动调校使能位 (基于 LOSC 进行频偏修正)  
0: 使能  
1: 禁止
- Bit 0 DBG1EB: DEBUG 端口 1 使能位 (PA5 用于 ISCK, PA4 用于 ISDA)  
0: 使能  
1: 禁止

### 3.2.1.2 特殊功能寄存器

#### 配置字 CFG\_WORD0 映射寄存器 (SCU\_CFGWORD0)

偏移地址: 1C<sub>H</sub>

复位值: 00000000\_00000000\_11111100\_11001111<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTRL<2:0>		WDTWIN<1:0>		WDTINTEN	LOSCEN	DBGEB	PWRTEB	BOREN	BORV<1:0>		WDTEN	—	HOSCM<1:0>		

—	bit31-16	—	—
CFGWORD<15:0>	bit15-0	R	<b>CFG_WORD0 配置字映射位</b> 读操作时, 可得到配置字 CFG_WORD0<15:0>的内容

### 3.2.2 芯片唯一识别码

96 位芯片唯一识别码 UID 位于 FLASH 存储器的信息区, 以字为单位存放, 分为 3 个 word, 用户程序可读。

#### 芯片唯一识别码 UID0

起始地址: 0020\_0100<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

UID<31:0>	Bit31-0	R	芯片唯一识别码 bits<31:0>
-----------	---------	---	--------------------

#### 芯片唯一识别码 UID1

起始地址: 0020\_0104<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

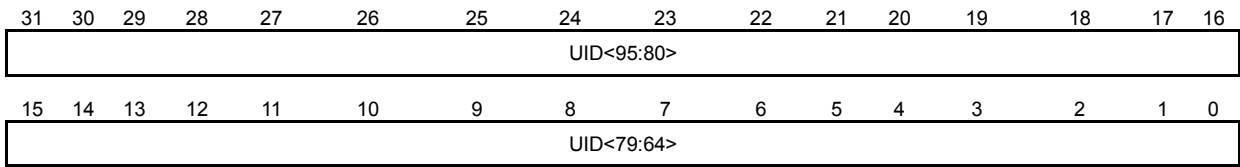
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID<63:48>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID<47:32>															

UID<63:32>	Bit31-0	R	芯片唯一识别码 bits<63:32>
------------	---------	---	---------------------

**芯片唯一识别码 UID2**

起始地址: 0020\_0108<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>



UID<95:64>	Bit31-0	R	芯片唯一识别码 bits<95:64>
------------	---------	---	---------------------

### 3.2.3 程序区FLASH

芯片内部的 Code FLASH 存储器总容量为 128K 字节，地址范围为 0000\_0000<sub>H</sub>~0001\_FFFF<sub>H</sub>，共分为 128 页，每页 1K 字节。Code FLASH 存储器支持至少 10 万次擦写次数，10 年以上的数据保持时间。

Code FLASH 存储器支持如下操作模式：

- 1) 全擦除模式，擦除整个程序存储区，擦除时间约为 10ms；
- 2) 页擦除模式，擦除指定页面，擦除时间约为 2ms；
- 3) 字编程模式，对字地址单元进行编程，时间约为 20us。

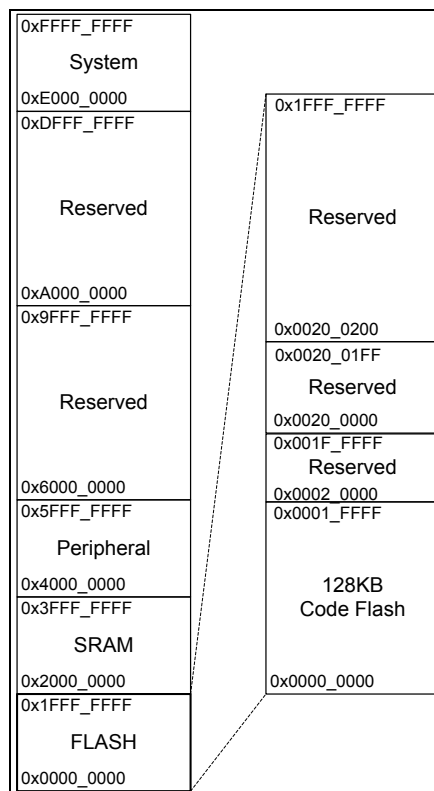


图 3-2 FLASH 映射图

### 3.2.4 自编程操作 (IAP)

芯片内部 FLASH 存储器，支持应用中自编程操作 IAP (In-Application Programming)。

写保护区使能后，位于写保护区的 FLASH 存储器地址单元，不支持 IAP 擦除和编程操作。可通过配置字 FWPEB 使能 FLASH 写保护区，通过配置字 FWPS 选择写保护区地址范围。

#### 3.2.4.1 概述

- ◇ 支持 FLASH 数据保护，进行 IAP 操作前需先进行解锁，去除相关寄存器的写保护
- ◇ 支持页擦除模式
- ◇ 支持字编程模式，每个字包括 4 个字节
- ◇ IAP 自编程操作程序需放在芯片的 SRAM 中执行，并在程序中对 FLASH 擦除或编程结果进行校验
- ◇ IAP 操作过程中可以响应中断

#### 3.2.4.2 IAP 操作流程

首先通过 IAP 控制寄存器，置位访问 FLASH 请求信号，查询得到允许应答；再进行对应的 IAP 操作；操作完成后清除 FLASH 请求信号，查询应答信号也被清零后，结束本次 FLASH 的访问操作。具体流程图如下所示：

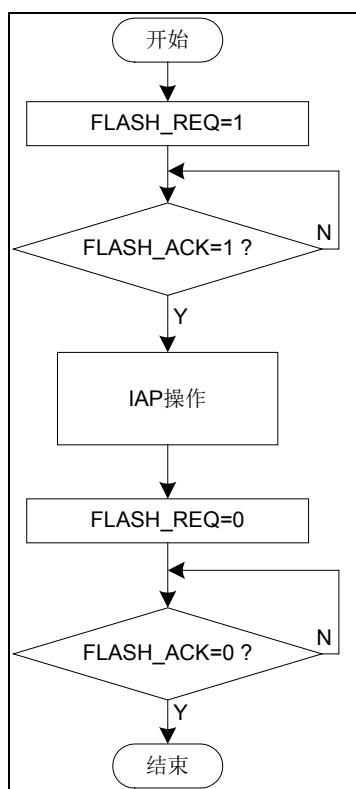


图 3-3 IAP 操作请求流程图

IAP 操作包括 FLASH 全擦除，页擦除和编程三种操作，其中 FLASH 全擦除操作，只在芯片调试模式使能时有效。具体各操作流程如下所示：

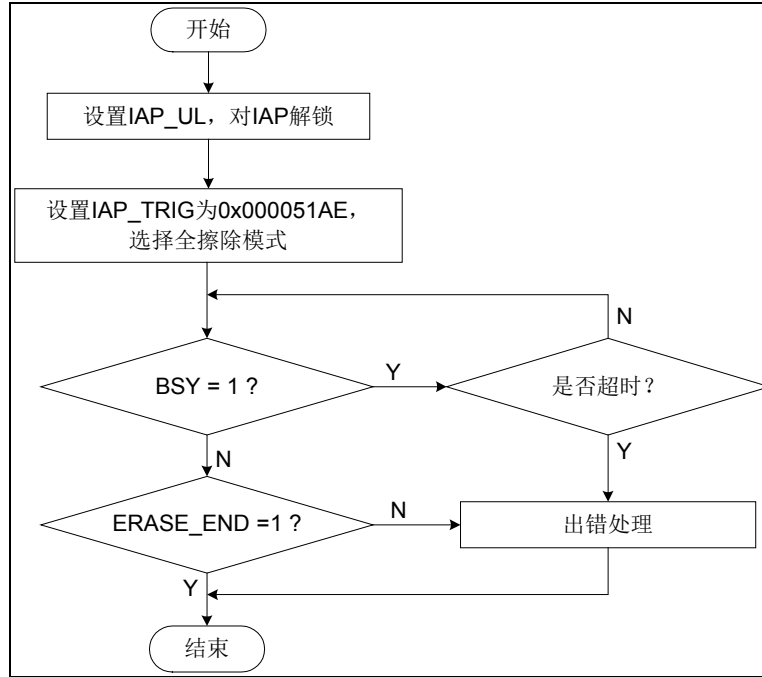


图 3-4 IAP 全擦除操作流程

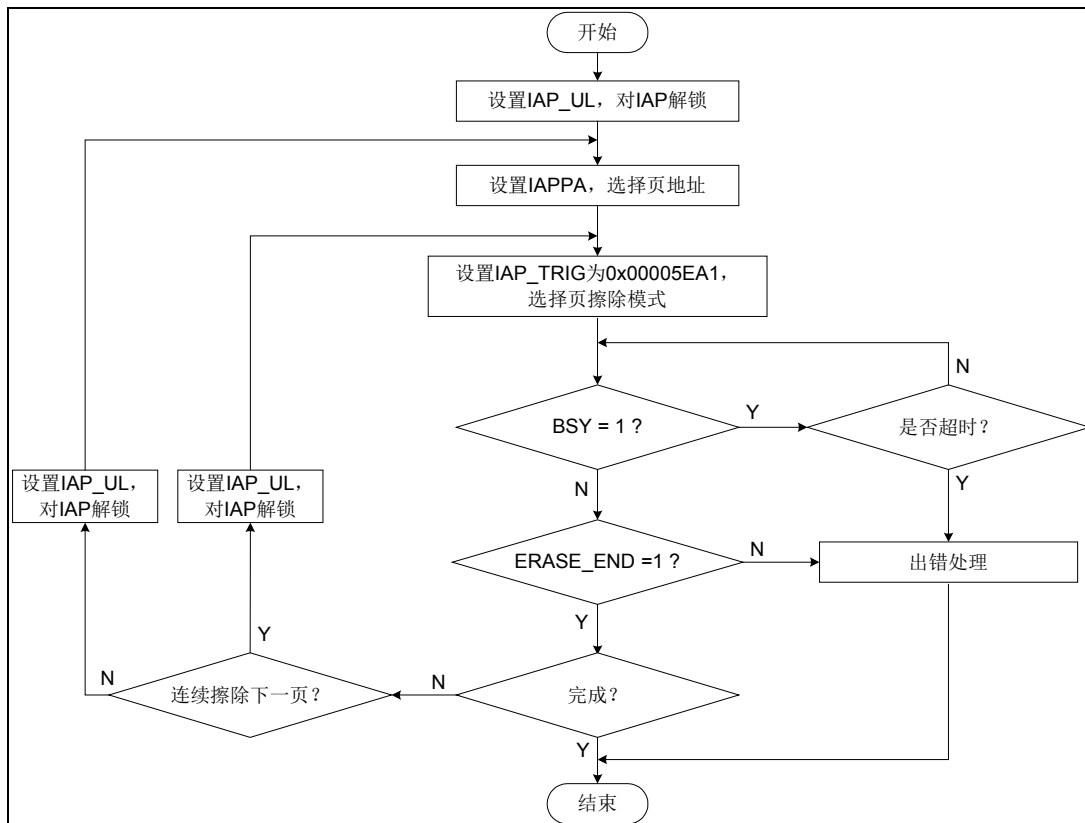


图 3-5 IAP 页擦除操作流程

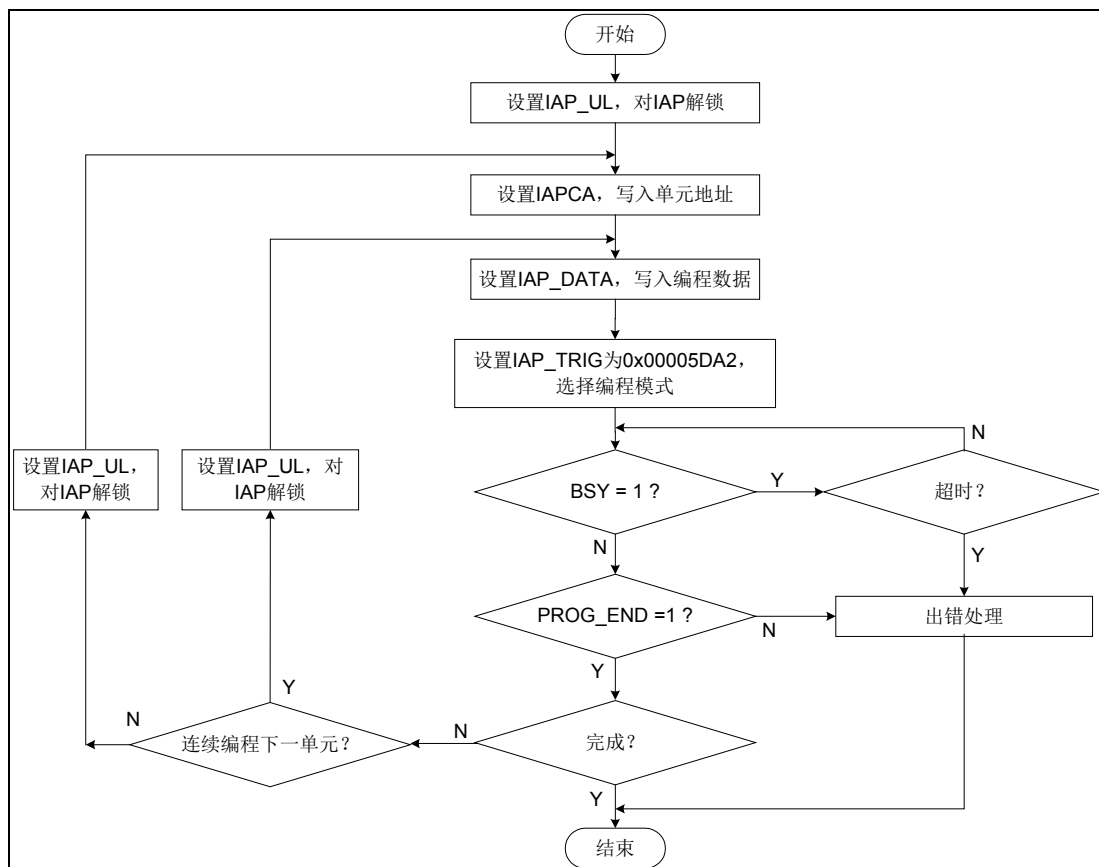


图 3-6 IAP 编程操作流程

由于在 IAP 自编程过程中，FLASH 的操作权限交给了 IAP 自编程操作程序，因此 CPU 无法从 FLASH 中读取指令，因而也无法响应中断。然而，只要保证在 IAP 过程中，CPU 运行的所有指令都从 SRAM 中取得，即可完成对中断的响应。这需要软件准备两个中断向量表，一个是在 FLASH 中使用，一个在 SRAM 中使用，当需要 IAP 时，用户需要将要在 FLASH 中使用的中断向量表写入到 SRAM 中即可。Cortex-M0 核心本身并不支持中断向量表的重映射，在 HR8P296 芯片中有两个特殊功能寄存器“中断向量表重映射使能寄存器”和“中断向量表偏移寄存器”，支持用户对中断向量表的重映射。当需要进行 IAP 时，软件将 IAP 过程中需要的中断向量表、需要响应的中断服务程序，以及 IAP 自编程操作程序写入到 SRAM 中。完成写入后，填写“中断向量表偏移寄存器”，并使能“中断向量表重映射使能寄存器”后，跳转到 SRAM 中去执行 IAP 自编程操作程序；完成 IAP 操作后，可关闭“中断向量表重映射使能寄存器”，跳回 FLASH 中继续正常工作。

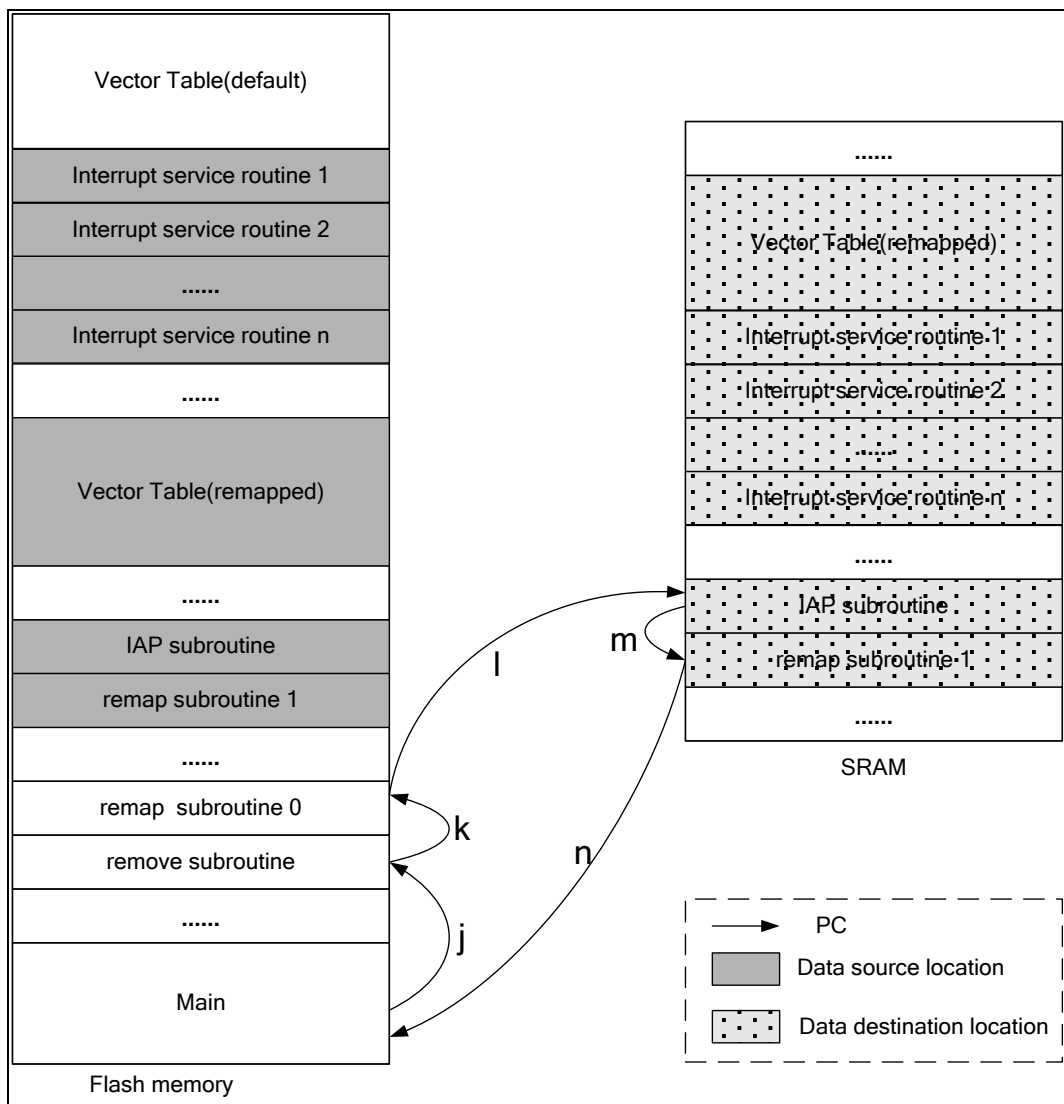


图 3-7 中断向量表重映射操作示意图

在中断向量表重映射之前，除了默认的中断向量表和中断服务程序外，还需要在 FLASH 中准备好重映射后的新的中断向量表、IAP 操作子程序和重映射子程序 1（将向量表重映射为默认状态，即向量表在 FLASH 中）。这些向量表和子程序均需烧写在 FLASH 中。需要注意，新中断向量表里的中断服务程序的入口地址变更为中断服务程序在 SRAM 中的地址。

软件需要经过以下步骤来完成 IAP 并且保证对中断的响应：

- 1) 程序跳转到搬运子程序中，完成对新中断向量表、中断服务程序、IAP 操作子程序、重映射子程序 1 从 FLASH（源方）到 SRAM（目标方）的搬运；为了保证所有的搬运数据均已“落实”到 SRAM 中，在搬运完成后需要追加一条“数据同步隔离（DSB）”指令；
- 2) 程序跳到重映射子程序 0 中，完成对中断向量表从 FLASH 到 SRAM 的映射；
- 3) 程序跳到 SRAM 中 IAP 子程序处，完成对 FLASH 的 IAP 操作；
- 4) 程序跳到 SRAM 中重映射子程序 1 中，完成对中断向量表从 SRAM 到 FLASH 的映射；



5) 程序跳回 FLASH 中，继续执行其它正常功能。

如果 IAP 过程中不需要响应中断，则需经过以下步骤：

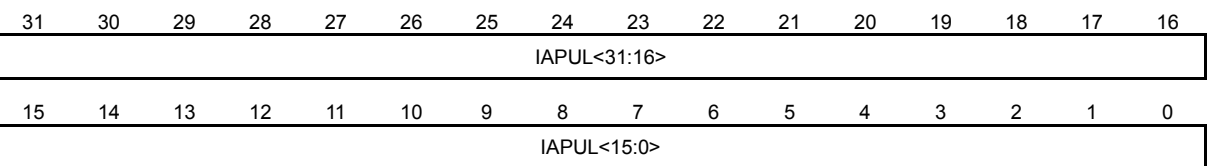
- 1) 程序跳到搬运子程序中，完成对 IAP 操作子程序的搬运；为了保证所有的搬运数据均已“落实”到 SRAM 中，在搬运完成后需要追加一条“数据同步隔离 (DSB)”指令；
- 2) 程序跳到 SRAM 中 IAP 子程序处，完成对 FLASH 的 IAP 操作；IAP 时需要关闭全局中断；
- 3) 程序跳回 FLASH 中，继续执行其它正常功能。

### 3.2.4.3 特殊功能寄存器

#### IAP 解锁寄存器 (IAP\_UL)

偏移地址：10<sub>H</sub>

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



IAPUL<31:0>	bit31-0	R/W	<p><b>IAP 解锁</b></p> <p>写入 0x0000_00A5，对 IAP 解锁；</p> <p>写入其它值，IAP 上锁；</p> <p>写入 IAP 触发寄存器 IAP_TRIG，IAP 自动上锁；</p> <p>写保留地址，IAP 上锁；</p> <p>IAP 软件复位后，IAP 上锁。</p>
-------------	---------	-----	--

注 1: IAP 上锁后，处于写保护状态的寄存器为 IAP\_CON, IAP\_ADDR, IAP\_DATA, IAP\_TRIG。

注 2: 写保留地址，IAP 上锁，是指对 40000800<sub>H</sub>~40000BFF<sub>H</sub> 空间中未定义的地址单元，进行写操作时，IAP 上锁。

#### IAP 控制寄存器 (IAP\_CON)

偏移地址：00<sub>H</sub>

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



—	bit31-8	—	—
FLASH_FAIL	bit7	R	<p><b>IAP 访问 FLASH 地址失败标识位</b></p> <p>0: IAP 未访问 FLASH 地址保护区</p>

—	bit6	—	1: IAP 访问 FLASH 地址保护区
FLASH_ACK	bit5	R	<b>FLASH 应答信号</b> 0: 不允许访问 1: 允许 IAP 访问 FLASH 存储器
FLASH_REQ	bit4	R/W	<b>IAP 访问 FLASH 请求信号</b> 0: 无请求 1: IAP 请求访问 FLASH 存储器
—	bit3-2	—	—
RST	bit1	W	<b>IAP 软件复位</b> 0: 读取时始终为 0 1: 复位
EN	bit0	R/W	<b>IAP 使能位</b> 0: 禁止 1: 使能

注：对 IAP\_CON 寄存器进行写操作前，需要先设置 IAP\_UL 寄存器，对 IAP 解锁，去除写保护。

**IAP 地址寄存器 (IAP\_ADDR)**

偏移地址: 04<sub>H</sub>

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

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

—	bit31-17	—	—
IAPPA<6:0>	bit16-10	R/W	<b>IAP 页地址</b> 0x00~0x7F: 共 128 页, 对应的地址范围 0x0000_0000~0x0001_FFFF
IAPCA<7:0>	bit9-2	R/W	<b>IAP 单元地址</b> 擦除模式时, 单元地址无效 编程模式时, 作为当前页中被编程单元的相对地址, 每页 256 个单元, 每单元 4 字节。编程前需保证该单元已经被擦除
—	bit1-0	—	—

注 1: 对 IAP\_ADDR 寄存器进行写操作前, 需要先设置 IAP\_UL 寄存器, 对 IAP 解锁, 去除写保护。

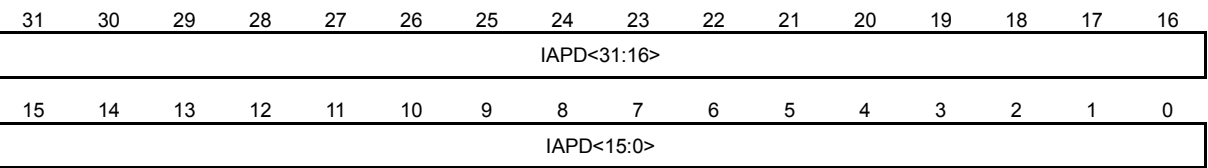
注 2: 完成页擦除后, IAPPA 自动+1;

注 3: 完成单元编程后, IAPCA 自动+1; 由于 IAPCA 只在当前页中进行单元寻址, 所以跨页编程时, 必须重新填写 IAPPA, 指定下一页地址。

**IAP 数据寄存器 (IAP\_DATA)**

偏移地址: 08<sub>H</sub>

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



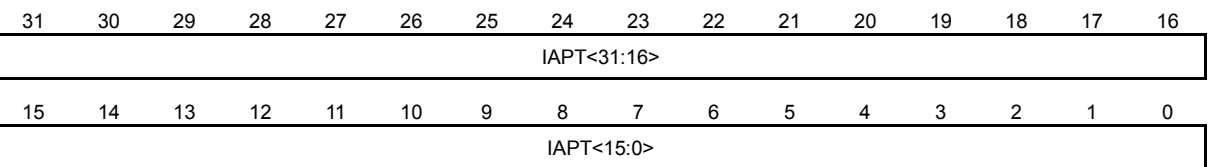
IAPD<31:0>	bit31-0	R/W	IAP 单元数据
------------	---------	-----	----------

注: 对 IAP\_DATA 寄存器进行写操作前, 需要先设置 IAP\_UL 寄存器, 对 IAP 解锁, 去除写保护。

**IAP 触发寄存器 (IAP\_TRIG)**

偏移地址: 0C<sub>H</sub>

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



IAPT	bit31-0	R/W	<b>IAP 操作命令 (写入该寄存器后, IAP 重新上锁)</b> 0x0000_51AE: 全擦除 (只在 SWD 调试时能用) 0x0000_5EA1: 页擦除 0x0000_5DA2: 编程模式 其它: 无操作 (IAP 完成后, 硬件自动更改为无操作)
------	---------	-----	--

注: 对 IAP\_TRIG 寄存器进行写操作前, 需要先设置 IAP\_UL 寄存器, 对 IAP 解锁, 去除写保护。

**IAP 状态寄存器 (IAP\_STA)**

偏移地址: 14<sub>H</sub>

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



—	bit31-4	—	—
TIMEOUT_ERR	bit3	R/W	<b>IAP 超时错误标志</b> 0: 写 0 清除, 触发 IAP_TRIG 自动清除, 或 IAP 软件复位清除

			1: IAP 操作超时
PROG_END	bit2	R/W	<b>IAP 编程结束标志</b> 0: 写 0 清除, 触发 IAP_TRIG 自动清除, 或 IAP 软件复位清除 1: 当前单元编程完成
ERASE_END	bit1	R/W	<b>IAP 页擦除结束标志</b> 0: 写 0 清除, 触发 IAP_TRIG 自动清除, 或 IAP 软件复位清除 1: 当前页擦除完成
BSY	bit0	R	<b>IAP 工作状态</b> 0: 空闲, IAP 软件复位可将该位清零 1: IAP 操作正在进行中

### 3.3 SRAM存储器

芯片内部集成 12K 字节数据存储器 SRAM，地址范围为 2000\_0000<sub>H</sub>~2000\_2FFF<sub>H</sub>。

#### 3.3.1 SRAM地址映射

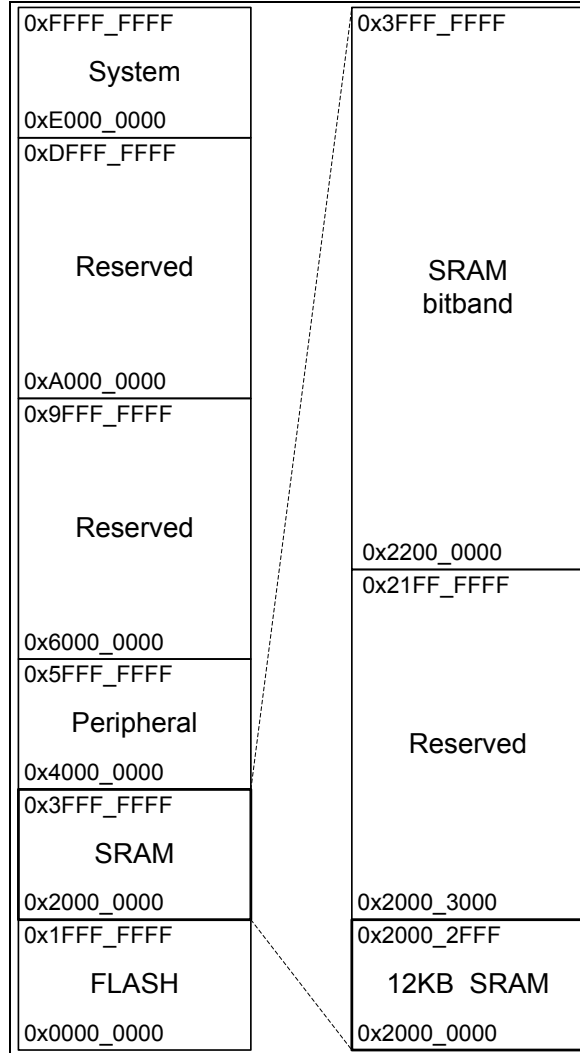


图 3-8 SRAM 映射图

#### 3.3.2 SRAM位带扩展

SRAM 位带扩展功能，对 SRAM 的每个 bit，都赋予了一个扩展地址，通过该扩展地址，可直接访问其对应的 SRAM 数据位，从而极大的方便了对 SRAM 单元的位读写操作。对于 SRAM 的某个 bit，记它所在字节地址为 A，位序号为 N (0≤N≤7)，SRAM 位带扩展映射区的基地址为 0x2200\_0000，则该 bit 的位带扩展地址为：

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

### 3.4 外设寄存器

#### 3.4.1 外设寄存器地址映射

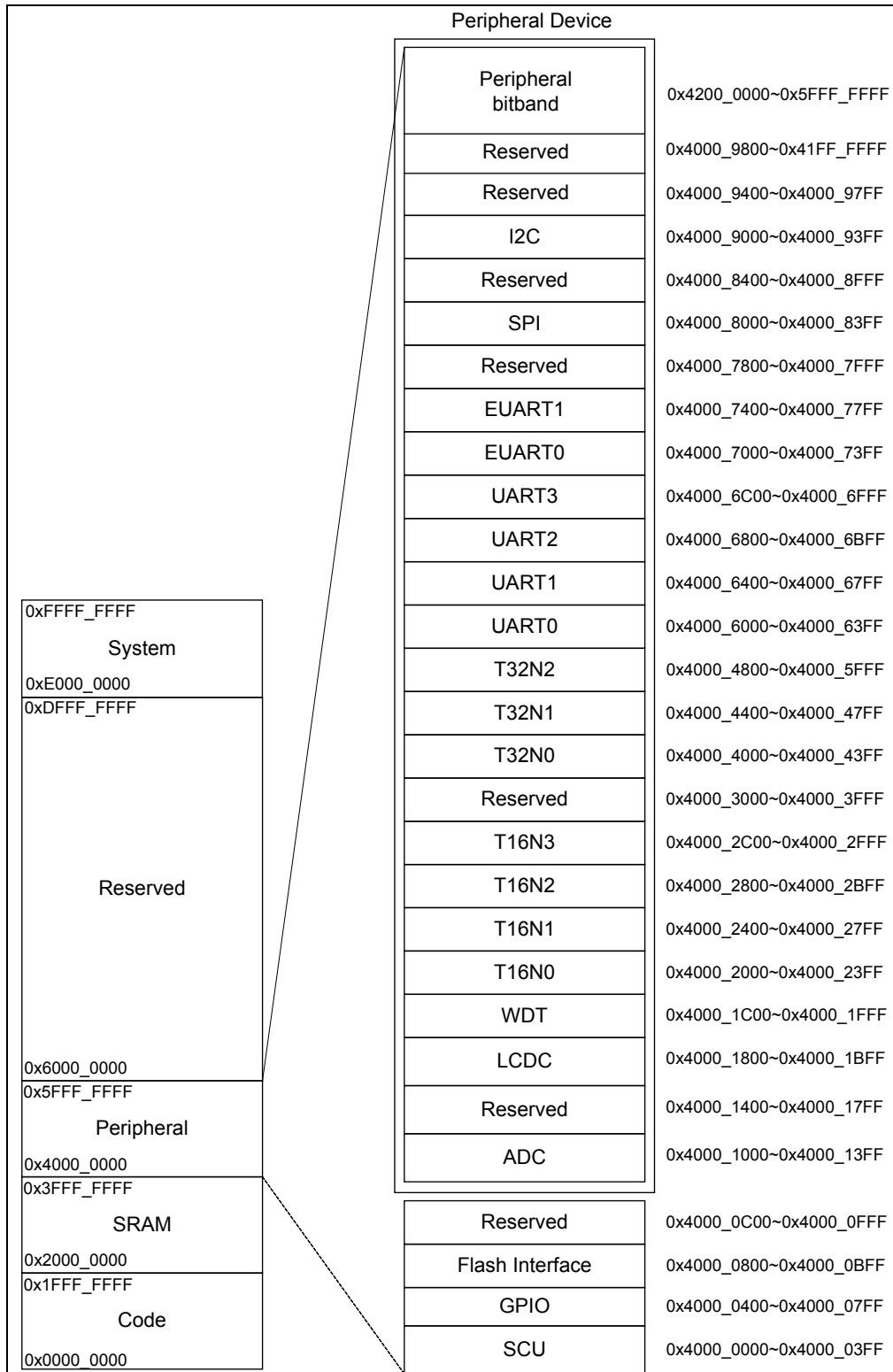


图 3-9 外设寄存器映射图

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

### 3.4.2 系统控制单元（SCU）寄存器列表

SCU 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
SCU 基地址: 4000_0000 <sub>H</sub>		
SCU_PROT	0000 <sub>H</sub>	系统设置保护寄存器
SCU_NMIC	0004 <sub>H</sub>	NMI 控制寄存器
SCU_PWRC	0008 <sub>H</sub>	复位寄存器
SCU_LVDC	0010 <sub>H</sub>	LVD 控制寄存器
SCU_CFGWORD0	001C <sub>H</sub>	配置字 CFG_WORD0 映射寄存器
SCU_FLASHW	0020 <sub>H</sub>	FLASH 访问等待时间寄存器
SCU_SCLKEN0	0040 <sub>H</sub>	系统时钟控制寄存器 0
SCU_SCLKEN1	0044 <sub>H</sub>	系统时钟设置寄存器 1
SCU_PCLKEN0	0048 <sub>H</sub>	外设时钟控制寄存器 0
SCU_PCLKEN1	004C <sub>H</sub>	外设时钟控制寄存器 1
SCU_VRCON	0050 <sub>H</sub>	VR 控制寄存器
SCU_TBLRMEN	0060 <sub>H</sub>	中断向量表重映射使能寄存器
SCU_TBLOFFS	0064 <sub>H</sub>	中断向量表偏移寄存器

### 3.4.3 GPIO寄存器列表

GPIO 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
GPIO 基地址: 4000_0400 <sub>H</sub>		
GPIO_PADIR	0000 <sub>H</sub>	PA 端口方向控制寄存器
GPIO_PADIRS	0004 <sub>H</sub>	PA 端口方向控制置位寄存器
GPIO_PADIRC	0008 <sub>H</sub>	PA 端口方向控制清零寄存器
GPIO_PADIRI	000C <sub>H</sub>	PA 端口方向控制取反寄存器
GPIO_PA	0010 <sub>H</sub>	PA 端口数据寄存器
GPIO_PAS	0014 <sub>H</sub>	PA 端口数据置位寄存器
GPIO_PAC	0018 <sub>H</sub>	PA 端口数据清零寄存器
GPIO_PAI	001C <sub>H</sub>	PA 端口数据取反寄存器
GPIO_PAFUN0	0020 <sub>H</sub>	PA[7:0]端口复用选择寄存器
GPIO_PAFUN1	0024 <sub>H</sub>	PA[15:8]端口复用选择寄存器
GPIO_PAFUN2	0028 <sub>H</sub>	PA[23:16]端口复用选择寄存器
GPIO_PAFUN3	002C <sub>H</sub>	PA[31:24]端口复用选择寄存器
GPIO_PAPUEN	0030 <sub>H</sub>	PA 端口弱上拉使能寄存器
GPIO_PAPDEN	0034 <sub>H</sub>	PA 端口弱下拉使能寄存器
GPIO_PAOD	0038 <sub>H</sub>	PA 端口输出开漏使能寄存器
GPIO_PBDIR	0080 <sub>H</sub>	PB 端口方向控制寄存器
GPIO_PBDIRS	0084 <sub>H</sub>	PB 端口方向控制置位寄存器
GPIO_PBDIRC	0088 <sub>H</sub>	PB 端口方向控制清零寄存器
GPIO_PBDIRI	008C <sub>H</sub>	PB 端口方向控制取反寄存器

GPIO 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
GPIO_PB	0090 <sub>H</sub>	PB 端口数据寄存器
GPIO_PBS	0094 <sub>H</sub>	PB 端口数据置位寄存器
GPIO_PBC	0098 <sub>H</sub>	PB 端口数据清零寄存器
GPIO_PBI	009C <sub>H</sub>	PB 端口数据取反寄存器
GPIO_PBFUN0	00A0 <sub>H</sub>	PB[7:0]端口复用选择寄存器
GPIO_PBFUN1	00A4 <sub>H</sub>	PB[15:8]端口复用选择寄存器
GPIO_PBFUN2	00A8 <sub>H</sub>	PB[23:16]端口复用选择寄存器
GPIO_PBPUEEN	00B0 <sub>H</sub>	PB 端口弱上拉使能寄存器
GPIO_PBPDEN	00B4 <sub>H</sub>	PB 端口弱下拉使能寄存器
GPIO_PBOD	00B8 <sub>H</sub>	PB 端口输出开漏使能寄存器
GPIO_PBSMITSEL	00C0 <sub>H</sub>	PB 端口 SMIT 窗口选择寄存器
GPIO_PCDIR	0100 <sub>H</sub>	PC 端口方向控制寄存器
GPIO_PCDIRS	0104 <sub>H</sub>	PC 端口方向控制置位寄存器
GPIO_PCDIRC	0108 <sub>H</sub>	PC 端口方向控制清零寄存器
GPIO_PCDIRI	010C <sub>H</sub>	PC 端口方向控制取反寄存器
GPIO_PC	0110 <sub>H</sub>	PC 端口数据寄存器
GPIO_PCS	0114 <sub>H</sub>	PC 端口数据置位寄存器
GPIO_PCC	0118 <sub>H</sub>	PC 端口数据清零寄存器
GPIO_PCI	011C <sub>H</sub>	PC 端口数据取反寄存器
GPIO_PCFUN0	0120 <sub>H</sub>	PC[7:0]端口复用选择寄存器
GPIO_PCFUN1	0124 <sub>H</sub>	PC[15:8]端口复用选择寄存器
GPIO_PCPUEN	0130 <sub>H</sub>	PC 端口弱上拉使能寄存器
GPIO_PCPDEN	0134 <sub>H</sub>	PC 端口弱下拉使能寄存器
GPIO_PCOD	0138 <sub>H</sub>	PC 端口输出开漏使能寄存器
GPIO_PIE	0200 <sub>H</sub>	PINT 中断使能寄存器
GPIO_PIF	0204 <sub>H</sub>	PINT 中断标志寄存器
GPIO_PSEL	0208 <sub>H</sub>	PINT 中断源选择寄存器
GPIO_PCFG	020C <sub>H</sub>	PINT 中断配置寄存器
GPIO_TXPC	0220 <sub>H</sub>	UART 脉宽调制寄存器
GPIO_PROT	02FC <sub>H</sub>	GPIO 保护寄存器

### 3.4.4 FLASH存储器IAP寄存器列表

IAP 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
IAP 基地址: 4000_0800 <sub>H</sub>		
IAP_CON	0000 <sub>H</sub>	IAP 控制寄存器
IAP_ADDR	0004 <sub>H</sub>	IAP 地址寄存器
IAP_DATA	0008 <sub>H</sub>	IAP 数据寄存器
IAP_TRIG	000C <sub>H</sub>	IAP 触发寄存器
IAP_UL	0010 <sub>H</sub>	IAP 解锁寄存器
IAP_STA	0014 <sub>H</sub>	IAP 状态寄存器



### 3.4.5 ADC寄存器列表

ADC 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
ADC 基地址: 4000_1000 <sub>H</sub>		
ADC_DR	0000 <sub>H</sub>	ADC 转换值寄存器
ADC_CON0	0004 <sub>H</sub>	ADC 控制寄存器 0
ADC_CON1	0008 <sub>H</sub>	ADC 控制寄存器 1
ADC_CHS	000C <sub>H</sub>	ADC 通道选择寄存器
ADC_IE	0010 <sub>H</sub>	ADC 中断使能寄存器
ADC_IF	0014 <sub>H</sub>	ADC 中断标志寄存器
ADC_BUF	0020 <sub>H</sub>	ADC 缓冲控制寄存器

### 3.4.6 RTC寄存器列表

RTC 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
RTC 基地址: 4000_1400 <sub>H</sub>		
RTC_CON	0000 <sub>H</sub>	RTC 控制寄存器
RTC_CALC	0004 <sub>H</sub>	RTC 调校控制寄存器
RTC_CALF	0008 <sub>H</sub>	RTC 调校值寄存器
RTC_WA	0010 <sub>H</sub>	RTC 周闹钟寄存器
RTC_DA	0014 <sub>H</sub>	RTC 日闹钟寄存器
RTC_SEC	0018 <sub>H</sub>	RTC 秒寄存器
RTC_MIN	001C <sub>H</sub>	RTC 分钟寄存器
RTC_HOUR	0020 <sub>H</sub>	RTC 小时寄存器
RTC_WEEK	0024 <sub>H</sub>	RTC 周寄存器
RTC_DAY	0028 <sub>H</sub>	RTC 日寄存器
RTC_MON	002C <sub>H</sub>	RTC 月寄存器
RTC_YEAR	0030 <sub>H</sub>	RTC 年寄存器
RTC_WP	0040 <sub>H</sub>	RTC 写保护寄存器
RTC_IE	0050 <sub>H</sub>	RTC 中断使能寄存器
RTC_IF	0054 <sub>H</sub>	RTC 中断标志寄存器

### 3.4.7 LCDC寄存器列表

LCDC 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
LCDC 基地址: 4000_1800 <sub>H</sub>		
LCD_CON0	0000 <sub>H</sub>	LCD 控制寄存器 0
LCD_FLKT	0004 <sub>H</sub>	LCD 闪烁时间寄存器
LCD_SEL0	0008 <sub>H</sub>	LCD 段使能寄存器<31:0>
LCD_SEL1	000C <sub>H</sub>	LCD 段使能寄存器<55:32>
LCD_CON1	0010 <sub>H</sub>	LCD 控制寄存器 1

LCDC 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
LCD_IE	0014 <sub>H</sub>	LCD 中断使能寄存器
LCD_IF	0018 <sub>H</sub>	LCD 中断标志寄存器
LCD_D0	0020 <sub>H</sub>	LCD 像素寄存器 0
LCD_D1	0024 <sub>H</sub>	LCD 像素寄存器 1
LCD_D2	0028 <sub>H</sub>	LCD 像素寄存器 2
LCD_D3	002C <sub>H</sub>	LCD 像素寄存器 3
LCD_D4	0030 <sub>H</sub>	LCD 像素寄存器 4
LCD_D5	0034 <sub>H</sub>	LCD 像素寄存器 5
LCD_D6	0038 <sub>H</sub>	LCD 像素寄存器 6
LCD_D7	003C <sub>H</sub>	LCD 像素寄存器 7
LCD_D8	0040 <sub>H</sub>	LCD 像素寄存器 8
LCD_D9	0044 <sub>H</sub>	LCD 像素寄存器 9
LCD_D10	0048 <sub>H</sub>	LCD 像素寄存器 10
LCD_D11	004C <sub>H</sub>	LCD 像素寄存器 11
LCD_D12	0050 <sub>H</sub>	LCD 像素寄存器 12
LCD_D13	0054 <sub>H</sub>	LCD 像素寄存器 13

### 3.4.8 WDT寄存器列表

WDT 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
WDT 基地址: 4000_1C00 <sub>H</sub>		
WDT_LOAD	0000 <sub>H</sub>	WDT 计数器装载值寄存器
WDT_VALUE	0004 <sub>H</sub>	WDT 计数器当前值寄存器
WDT_CON	0008 <sub>H</sub>	WDT 控制寄存器
WDT_INTCLR	000C <sub>H</sub>	WDT 中断标志清除寄存器
WDT_RIS	0010 <sub>H</sub>	WDT 中断标志寄存器
WDT_LOCK	0100 <sub>H</sub>	WDT 访问使能寄存器

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

T16N 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
T16N0 基地址: 4000_2000 <sub>H</sub>		
T16N1 基地址: 4000_2400 <sub>H</sub>		
T16N2 基地址: 4000_2800 <sub>H</sub>		
T16N3 基地址: 4000_2C00 <sub>H</sub>		
T16N_CNT	0000 <sub>H</sub>	T16N 计数值寄存器
T16N_CON0	0004 <sub>H</sub>	T16N 控制寄存器 0
T16N_CON1	0008 <sub>H</sub>	T16N 控制寄存器 1
T16N_PRECNT	0010 <sub>H</sub>	T16N 预分频器计数值寄存器

T16N 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
T16N_PREMAT	0014 <sub>H</sub>	T16N 预分频器计数匹配寄存器
T16N_IE	0018 <sub>H</sub>	T16N 中断使能寄存器
T16N_IF	001C <sub>H</sub>	T16N 中断标志寄存器
T16N_MAT0	0020 <sub>H</sub>	T16N 计数匹配寄存器 0
T16N_MAT1	0024 <sub>H</sub>	T16N 计数匹配寄存器 1
T16N_MAT2	0028 <sub>H</sub>	T16N 计数匹配寄存器 2
T16N_MAT3	002C <sub>H</sub>	T16N 计数匹配寄存器 3

### 3.4.10 T32N0/T32N1/T32N2 寄存器列表

T32N 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
T32N0 基地址: 4000_4000 <sub>H</sub>		
T32N1 基地址: 4000_4400 <sub>H</sub>		
T32N2 基地址: 4000_4800 <sub>H</sub>		
T32N_CNT	0000 <sub>H</sub>	T32N 计数值寄存器
T32N_CON0	0004 <sub>H</sub>	T32N 控制寄存器 0
T32N_CON1	0008 <sub>H</sub>	T32N 控制寄存器 1
T32N_PRECNT	0010 <sub>H</sub>	T32N 预分频器计数值寄存器
T32N_PREMAT	0014 <sub>H</sub>	T32N 预分频器计数匹配寄存器
T32N_IE	0018 <sub>H</sub>	T32N 中断使能寄存器
T32N_IF	001C <sub>H</sub>	T32N 中断标志寄存器
T32N_MAT0	0020 <sub>H</sub>	T32N 计数匹配寄存器 0
T32N_MAT1	0024 <sub>H</sub>	T32N 计数匹配寄存器 1
T32N_MAT2	0028 <sub>H</sub>	T32N 计数匹配寄存器 2
T32N_MAT3	002C <sub>H</sub>	T32N 计数匹配寄存器 3

### 3.4.11 UART0/UART1/UART2/UART3 寄存器列表

UART 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
UART0 基地址: 4000_6000 <sub>H</sub>		
UART1 基地址: 4000_6400 <sub>H</sub>		
UART2 基地址: 4000_6800 <sub>H</sub>		
UART3 基地址: 4000_6C00 <sub>H</sub>		
UART_CON	0000 <sub>H</sub>	UART 控制寄存器
UART_BRR	0004 <sub>H</sub>	UART 波特率寄存器
UART_TBW	0008 <sub>H</sub>	UART 发送数据写入寄存器
UART_RBR	000C <sub>H</sub>	UART 接收数据读取寄存器
UART_TB01	0010 <sub>H</sub>	UART 发送缓冲寄存器 0/1
UART_TB23	0014 <sub>H</sub>	UART 发送缓冲寄存器 2/3

UART 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
UART_RB01	0018 <sub>H</sub>	UART 接收缓冲寄存器 0/1
UART_RB23	001C <sub>H</sub>	UART 接收缓冲寄存器 2/3
UART_IE	0020 <sub>H</sub>	UART 中断使能寄存器
UART_IF	0024 <sub>H</sub>	UART 中断标志寄存器

### 3.4.12 EUART0/1 寄存器列表

EUART 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
EUART0 基地址: 4000_7000 <sub>H</sub>		
EUART1 基地址: 4000_7400 <sub>H</sub>		
EUART_CON	0000 <sub>H</sub>	EUART 控制寄存器
EUART_BRR	0004 <sub>H</sub>	EUART 波特率寄存器
EUART_TBW	0008 <sub>H</sub>	EUART 发送数据写入寄存器
EUART_RBR	000C <sub>H</sub>	EUART 接收数据读取寄存器
EUART_TB01	0010 <sub>H</sub>	EUART 发送缓冲寄存器 0/1
EUART_TB23	0014 <sub>H</sub>	EUART 发送缓冲寄存器 2/3
EUART_RB01	0018 <sub>H</sub>	EUART 接收缓冲寄存器 0/1
EUART_RB23	001C <sub>H</sub>	EUART 接收缓冲寄存器 2/3
EUART_IE	0020 <sub>H</sub>	EUART 中断使能寄存器
EUART_IF	0024 <sub>H</sub>	EUART 中断标志寄存器
EUART_MOD	0030 <sub>H</sub>	EUART 7816 模式控制寄存器

### 3.4.13 SPI寄存器列表

SPI 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
SPI 基地址: 4000_8000 <sub>H</sub>		
SPI_CON	0000 <sub>H</sub>	SPI 控制寄存器
SPI_TBW	0008 <sub>H</sub>	SPI 发送数据写入寄存器
SPI_RBR	000C <sub>H</sub>	SPI 接收数据读取寄存器
SPI_IE	0010 <sub>H</sub>	SPI 中断使能寄存器
SPI_IF	0014 <sub>H</sub>	SPI 中断标志寄存器
SPI_TB	0018 <sub>H</sub>	SPI 发送缓冲寄存器
SPI_RB	001C <sub>H</sub>	SPI 接收缓冲寄存器
SPI_STA	0020 <sub>H</sub>	SPI 状态寄存器

### 3.4.14 I2C寄存器列表

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

### 3.4.15 外设寄存器位带扩展

外设寄存器位带扩展功能，对外设寄存器的每个 bit，都赋予了一个扩展地址，通过该扩展地址，可直接访问其对应的寄存器位，从而极大的方便了对外设寄存器的位读写操作。对于外设寄存器的某个 bit，记它所在字节地址为 A，位序号为 N (0≤N≤7)，外设寄存器位带扩展映射区的基地址为 0x4200\_0000，则该 bit 的位带扩展地址为：

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

GPIO 端口寄存器 GPIO\_PADIRS, GPIO\_PADIRC, GPIO\_PADIRI, GPIO\_PAS, GPIO\_PAC, GPIO\_PAI, GPIO\_PBDIRS, GPIO\_PBDIRC, GPIO\_PBDIRI, GPIO\_PBS, GPIO\_PBC, GPIO\_PBI, GPIO\_PCDIRS, GPIO\_PCDIRC, GPIO\_PCDIRI, GPIO\_PCS, GPIO\_PCC, GPIO\_PCI, 用于实现对 GPIO 端口方向控制寄存器和数据寄存器的位操作，这些端口位操作寄存器不再支持位带扩展访问。除上述寄存器外，其它外设寄存器均支持位带扩展访问操作。

RTC 模块里的 RTC\_WA, RTC\_DA, RTC\_HOUR 也不支持位带扩展访问。

## 3.5 内核寄存器

### 3.5.1 系统定时器 (SYSTICK) 寄存器列表

SYSTICK 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
SYSTICK 基地址: E000_E000 <sub>H</sub>		
SYST_CSR	0010 <sub>H</sub>	SYSTICK 控制和状态寄存器
SYST_RVR	0014 <sub>H</sub>	SYSTICK 重装值寄存器
SYST_CVR	0018 <sub>H</sub>	SYSTICK 当前值寄存器
SYST_CALIB	001C <sub>H</sub>	SYSTICK 校准值寄存器

### 3.5.2 中断控制器（NVIC）寄存器列表

NVIC 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
NVIC 基地址: E000_E100 <sub>H</sub>		
NVIC_ISER	0000 <sub>H</sub>	IRQ0~31 置中断请求使能寄存器
NVIC_ICER	0080 <sub>H</sub>	IRQ0~31 清中断请求使能寄存器
NVIC_ISPR	0100 <sub>H</sub>	IRQ0~31 置中断挂起寄存器
NVIC_ICPR	0180 <sub>H</sub>	IRQ0~31 清中断挂起寄存器
NVIC_PR0	0300 <sub>H</sub>	IRQ0~3 优先级控制寄存器
NVIC_PR1	0304 <sub>H</sub>	IRQ4~7 优先级控制寄存器
NVIC_PR2	0308 <sub>H</sub>	IRQ8~11 优先级控制寄存器
NVIC_PR3	030C <sub>H</sub>	IRQ12~15 优先级控制寄存器
NVIC_PR4	0310 <sub>H</sub>	IRQ16~19 优先级控制寄存器
NVIC_PR5	0314 <sub>H</sub>	IRQ20~23 优先级控制寄存器
NVIC_PR6	0318 <sub>H</sub>	IRQ24~27 优先级控制寄存器
NVIC_PR7	031C <sub>H</sub>	IRQ28~31 优先级控制寄存器

### 3.5.3 系统控制块（SCB）寄存器列表

SCB 寄存器列表		
寄存器名称	寄存器偏移地址	寄存器描述
SCB 基地址: E000_ED00 <sub>H</sub>		
CPUID	0000 <sub>H</sub>	CPUID 寄存器
ICSR	0004 <sub>H</sub>	中断控制和状态寄存器
AIRCR	000C <sub>H</sub>	应用中断和复位控制寄存器
SCR	0010 <sub>H</sub>	系统控制寄存器
CCR	0014 <sub>H</sub>	配置和控制寄存器
SHPR2	001C <sub>H</sub>	系统处理程序优先级寄存器 2
SHPR3	0020 <sub>H</sub>	系统处理程序优先级寄存器 3

## 第4章 输入输出端口

### 4.1 概述

本芯片支持 56 个 (HR8P296FLLT) / 72 个 (HR8P296FLLV) I/O 端口。

所有 I/O 端口都是 TTL 施密特输入和 CMOS 输出驱动。每个 I/O 端口的功能由端口复用选择寄存器 GPIO\_PAFUNC/GPIO\_PBFUNC/GPIO\_PCFUNC 配置。

每个 I/O 端口最多可有四种复用功能 (FUN0, FUN1, FUN2, FUN3)，通过端口复用选择寄存器 GPIO\_PAFUNC/GPIO\_PBFUNC/GPIO\_PCFUNC 进行配置。

当 I/O 端口配置为复用功能 FUN0 时，作为通用数字 I/O 端口，其输入输出状态由端口方向控制寄存器 GPIO\_PADIR/GPIO\_PBDIR/GPIO\_PCDIR 配置。当 I/O 端口处于输出状态时，其电平由端口数据寄存器 GPIO\_PA/ GPIO\_PB/ GPIO\_PC 决定，1 为高电平，0 为低电平；当 I/O 端口处于输入状态时，其电平状态可通过读取端口数据寄存器 GPIO\_PA/GPIO\_PB/ GPIO\_PC 获得；当 I/O 端口处于输入状态时，端口输入不允许处于悬空状态，必须输入高电平或低电平。

当 I/O 端口配置为复用功能 FUN1 或 FUN2 时，作为芯片外设功能模块的复用端口，其输入输出状态由外设功能模块决定，此时端口方向控制寄存器 GPIO\_PADIR/GPIO\_PBDIR/GPIO\_PCDIR 的配置无效；当 I/O 端口处于输入状态时，端口输入不允许处于悬空状态，必须输入高电平或低电平。

当 I/O 端口配置为复用功能 FUN3 时，作为模拟端口，禁止其数字端口功能。当 IO 端口在应用中悬空时，建议配置为复用功能 FUN3。

每个 I/O 端口均支持弱上拉和弱下拉，由端口弱上/下拉使能寄存器 GPIO\_PAPUEN/GPIO\_PBPEN/GPIO\_PCPEN 和 GPIO\_PAPDEN/GPIO\_PBPDEN/GPIO\_PCPDEN 控制使能。

支持 2 个大电流驱动口 PA12、PA24，最大灌电流能力 20mA。

支持 8 个外部端口中断 (PINT0~PINT7)，每个 PINT 可以从 8 个 I/O 端口输入中选择一个作为其中断源。

## 4.2 结构框图

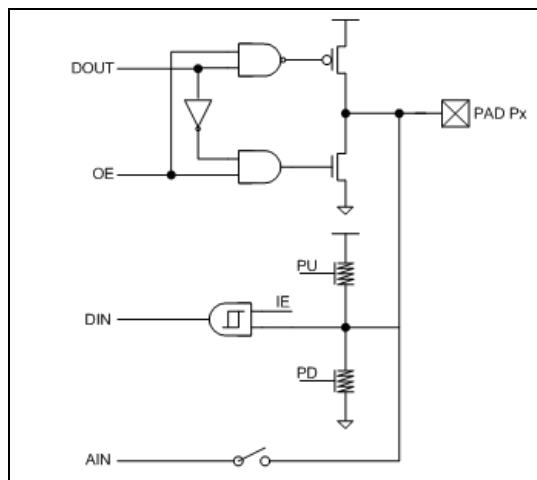


图 4-1 I/O 端口电路结构框图

## 4.3 外部端口中断

支持 8 个外部端口中断 (PINT0~PINT7)，每个中断均可以从 8 个 I/O 端口输入中选择一个作为其中断源 PINTIN。

输入屏蔽使能位  $PMASK<7:0>$ ，可对每个中断源输入 PINTIN 进行屏蔽。屏蔽使能时，不论 PINTIN 输入如何变化，GPIO\_PIF 寄存器中的中断标志位  $PINTIF<7:0>$  均保持不变。

GPIO\_PIE 寄存器中的中断使能位  $PINTIE<7:0>$ ，可对每个外部端口中断标志  $PINTIF<7:0>$  是否触发外部端口中断请求，进行配置。

外部端口中断配置寄存器 GPIO\_PCFG，可对每个中断源的触发方式进行配置，根据端口电平变化的不同状态，触发中断。切换中断源触发方式时，需先屏蔽中断源输入，避免误产生中断；或先禁止外部端口中断，并在切换完成后，对中断标志进行清零。

被选择作为中断源的 I/O 端口，其复用设置不能设为复用功能 FUN3 模拟端口使用。

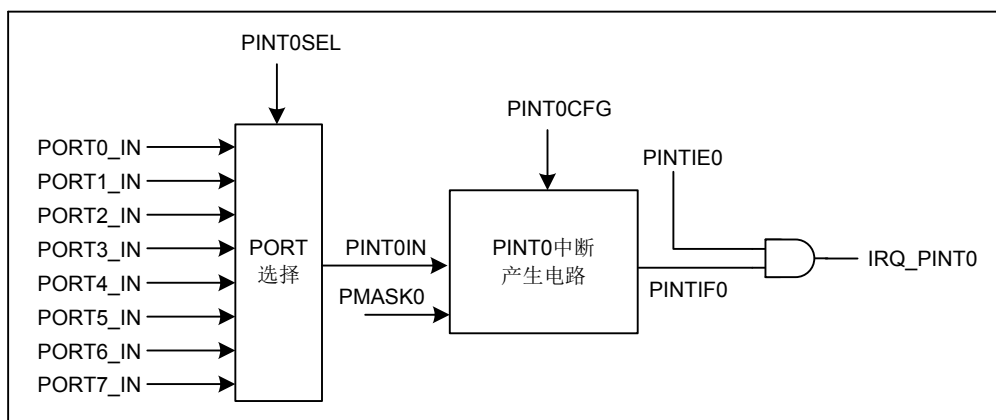


图 4-2 PINT0 外部端口中断电路结构示意图



## 4.4 特殊功能寄存器

### GPIO 保护寄存器 (GPIO\_PROT)

偏移地址: FC<sub>H</sub>

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

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

—	bit31-0	W	写 0x78879669 时位 PROT 位为 0; 写其它值时位 PROT 位为 1
PROT	bit0	R	<b>GPIO 写保护位</b> 0: 写保护关闭 1: 写保护使能

注 1: 只有以字方式对 GPIO\_PROT 寄存器写入 0x78879669 才能关闭写保护, 其他任何对 GPIO\_PROT 寄存器的写操作都将使能写保护功能。

注 2: GPIO\_PROT 保护的 GPIO 寄存器为 GPIO\_PAFUN0, GPIO\_PAFUN1, GPIO\_PAFUN2, GPIO\_PAFUN3, GPIO\_PBFUN0, GPIO\_PBFUN1, GPIO\_PBFUN2, GPIO\_PCFUN0, GPIO\_PCFUN1, GPIO\_PAPUEN, GPIO\_PBPUEEN, GPIO\_PCPUEEN, GPIO\_PAPDEN, GPIO\_PBPDEN, GPIO\_PCPDEN 和 GPIO\_TXPC。

注 3: GPIO 寄存器写保护使能后, 被保护的寄存器无法被写入, 但仍可被读取。

### PA 端口方向控制寄存器 (GPIO\_PADIR)

偏移地址: 00<sub>H</sub>

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

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

DIR<31:0>	bit31-0	R/W	<b>PA 端口方向控制位</b> 0: 输出 1: 输入
-----------	---------	-----	-------------------------------------

**PA 端口方向控制置位寄存器 (GPIO\_PADIRS)**

偏移地址: 04<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

DIRS<31:0>	bit31-0	W	<b>PA 端口方向控制置位选择</b> 0: 不改变端口方向 1: 端口方向设置为输入
------------	---------	---	--

**PA 端口方向控制清零寄存器 (GPIO\_PADIRC)**

偏移地址: 08<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

DIRC<31:0>	bit31-0	W	<b>PA 端口方向控制清零选择</b> 0: 不改变端口方向 1: 端口方向设置为输出
------------	---------	---	--

**PA 端口方向控制取反寄存器 (GPIO\_PADIRI)**

偏移地址: 0C<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

DIRI<31:0>	bit31-0	W	<b>PA 端口方向控制取反选择</b> 0: 不改变端口方向 1: 端口方向取反
------------	---------	---	---

**PA 端口数据寄存器 (GPIO\_PA)**

偏移地址: 10<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

DATA<31:0>	bit31-0	R/W	<b>PA 端口输入/输出电平</b> 0: 低电平 1: 高电平
------------	---------	-----	---

注: 对 GPIO\_PA 寄存器, 写操作时, 实际是写 PA 端口输出的数据; 读操作时, 实际是读取 PA 端口的电平状态。

**PA 端口数据置位寄存器 (GPIO\_PAS)**

偏移地址: 14<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

DATAS<31:0>	bit31-0	W	<b>PA 端口输出置位选择</b> 0: 不改变输出电平 1: 相应端口输出高电平
-------------	---------	---	--

**PA 端口数据清零寄存器 (GPIO\_PAC)**

偏移地址: 18<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

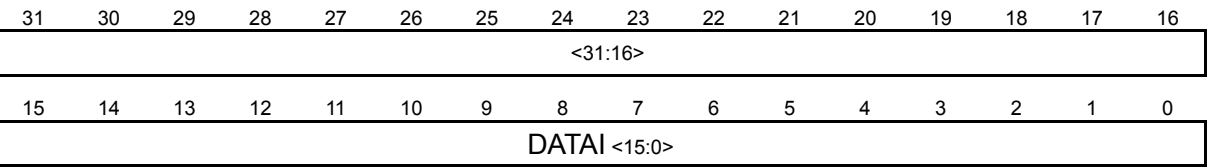
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAAC <31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAAC <15:0>															

DATAAC<31:0>	bit31-0	W	<b>PA 端口输出清零选择</b> 0: 不改变输出电平 1: 相应端口输出低电平
--------------	---------	---	--

**PA 端口数据取反寄存器 (GPIO\_PA1)**

偏移地址: 1C<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

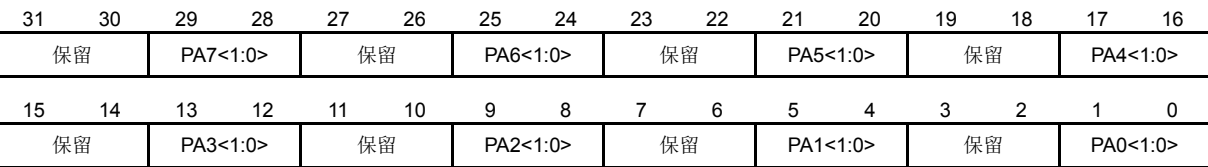


DATAI<31:0>	bit31-0	W	<b>PA 端口输出取反选择</b> 0: 不改变输出电平 1: 相应端口输出取反
-------------	---------	---	---

**PA[7:0]端口复用选择寄存器 (GPIO\_PAFUN0)**

偏移地址: 20<sub>H</sub>

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



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

—	bit3-2	—	—
PA0<1:0>	bit1-0	R/W	<b>PA0 复用选择位</b> 00~11: FUN0~FUN3

**PA[15:8]端口复用选择寄存器 (GPIO\_PAFUN1)**

偏移地址: 24<sub>H</sub>

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

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

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

**PA[23:16]端口复用选择寄存器 (GPIO\_PAFUN2)**

偏移地址: 28<sub>H</sub>

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

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

—	bit31-30	—	—
PA23<1:0>	bit29-28	R/W	<b>PA23 复用选择位</b> 00~11: FUN0~FUN3
—	bit27-26	—	—
PA22<1:0>	bit25-24	R/W	<b>PA22 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PA21<1:0>	bit21-20	R/W	<b>PA21 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PA20<1:0>	bit17-16	R/W	<b>PA20 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PA19<1:0>	bit13-12	R/W	<b>PA19 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PA18<1:0>	bit9-8	R/W	<b>PA18 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PA17<1:0>	bit5-4	R/W	<b>PA17 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PA16<1:0>	bit1-0	R/W	<b>PA16 复用选择位</b> 00~11: FUN0~FUN3

**PA[31:24]端口复用选择寄存器 (GPIO\_PAFUN3)**

偏移地址: 2C<sub>H</sub>

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

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

—	bit31-30	—	—
PA31<1:0>	bit29-28	R/W	<b>PA31 复用选择位</b> 00~11: FUN0~FUN3
—	bit27-26	—	—
PA30<1:0>	bit25-24	R/W	<b>PA30 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PA29<1:0>	bit21-20	R/W	<b>PA29 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PA28<1:0>	bit17-16	R/W	<b>PA28 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PA27<1:0>	bit13-12	R/W	<b>PA27 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PA26<1:0>	bit9-8	R/W	<b>PA26 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PA25<1:0>	bit5-4	R/W	<b>PA25 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PA24<1:0>	bit1-0	R/W	<b>PA24 复用选择位</b> 00~11: FUN0~FUN3

**PA 端口弱上拉使能寄存器 (GPIO\_PAPUEN)**

偏移地址: 30<sub>H</sub>

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

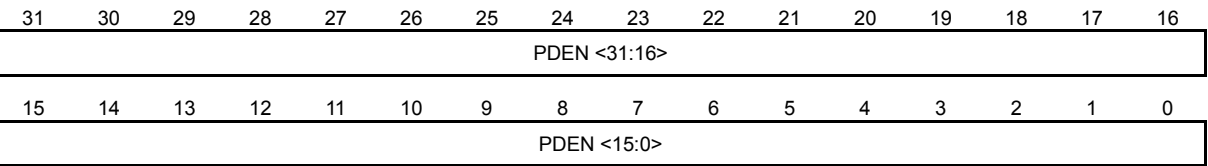
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUEN <31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUEN <15:0>															

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

**PA 端口弱下拉使能寄存器 (GPIO\_PAPDEN)**

偏移地址: 34<sub>H</sub>

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

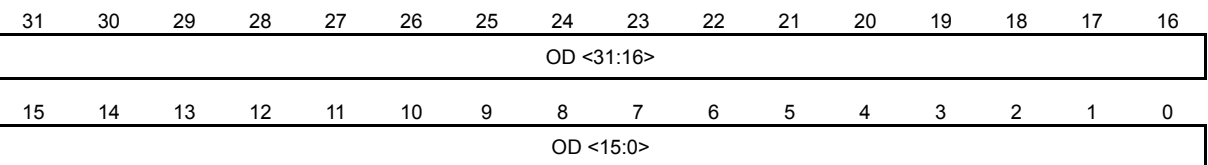


PDEN<31:0>	bit31-0	R/W	<b>PA 端口弱下拉使能位</b> 0: 禁止 1: 使能
------------	---------	-----	--------------------------------------

**PA 端口输出开漏使能寄存器 (GPIO\_PAOD)**

偏移地址: 38<sub>H</sub>

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

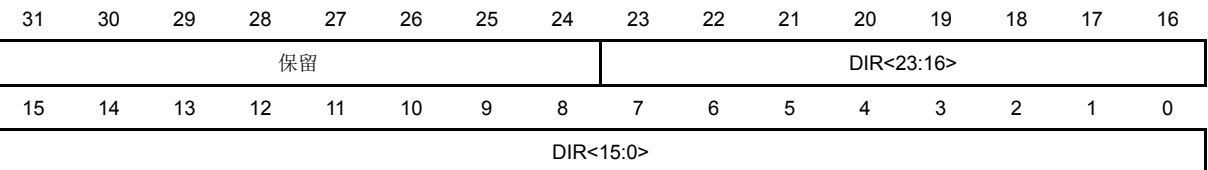


OD<31:0>	bit31-0	R/W	<b>PA 输出开漏使能位</b> 0: 禁止 1: 使能
----------	---------	-----	-------------------------------------

**PB 端口方向控制寄存器 (GPIO\_PBDIR)**

偏移地址: 80<sub>H</sub>

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



—	bit31-24	—	—
DIR<23:0>	bit23-0	R/W	<b>PB 端口方向控制位</b> 0: 输出 1: 输入



**PB 端口方向控制置位寄存器 (GPIO\_PBDIRS)**

偏移地址: 84<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-24	—	—
DIRS<23:0>	bit23-0	W	<b>PB 端口方向控制置位选择</b> 0: 不改变端口方向 1: 端口方向设置为输入

**PB 端口方向控制寄清零寄存器 (GPIO\_PBDIRC)**

偏移地址: 88<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

DIRC<23:0>	bit23-0	W	<b>PB 端口方向控制清零选择</b> 0: 不改变端口方向 1: 端口方向设置为输出
—	bit31-24	—	—

**PB 端口方向控制取反寄存器 (GPIO\_PBDIRI)**

偏移地址: 8C<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-24	—	—
DIRI<23:0>	bit23-0	W	<b>PB 端口方向控制取反选择</b> 0: 不改变端口方向 1: 端口方向取反

**PB 端口数据寄存器 (GPIO\_PB)**

偏移地址: 90<sub>H</sub>

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

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

—	bit31-24	—	—
DATA<23:0>	bit23-0	R/W	<b>PB 端口输入/输出电平</b> 0: 低电平 1: 高电平

注: 对 GPIO\_PB 寄存器, 写操作时, 实际是写 PB 端口输出的数据; 读操作时, 实际是读取 PB 端口的电平状态。

**PB 端口数据置位寄存器 (GPIO\_PBS)**

偏移地址: 94<sub>H</sub>

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

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

—	bit31-24	—	—
DATAS<23:0>	bit23-0	W	<b>PB 端口输出置位选择</b> 0: 不改变输出电平 1: 相应端口输出高电平

**PB 端口数据清零寄存器 (GPIO\_PBC)**

偏移地址: 98<sub>H</sub>

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

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

—	bit31-24	—	—
DATAC<23:0>	bit23-0	W	<b>PB 端口输出清零选择</b> 0: 不改变输出电平 1: 相应端口输出低电平

**PB 端口输出取反寄存器 (GPIO\_PBI)**

偏移地址: 9CH

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

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

—	bit31-24	—	—
DATAI<23:0>	bit23-0	W	<b>PB 端口输出取反选择</b> 0: 不改变输出电平 1: 相应端口输出取反

**PB[7:0]端口复用选择寄存器 (GPIO\_PBFUN0)**

偏移地址: A0H

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

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

—	bit31-30	—	—
PB7<1:0>	bit29-28	R/W	<b>PB7 复用选择位</b> 00~11: FUN0~FUN3
—	bit27-26	—	—
PB6<1:0>	bit25-24	R/W	<b>PB6 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PB5<1:0>	bit21-20	R/W	<b>PB5 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PB4<1:0>	bit17-16	R/W	<b>PB4 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PB3<1:0>	bit13-12	R/W	<b>PB3 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PB2<1:0>	bit9-8	R/W	<b>PB2 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—

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

**PB[15:8]端口复用选择寄存器 (GPIO\_PBFUN1)**

偏移地址: A4<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	PB15<1:0>	保留	保留	PB14<1:0>	保留	保留	保留	PB13<1:0>	保留	保留	保留	保留	保留	PB12<1:0>	保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PB11<1:0>	保留	保留	PB10<1:0>	保留	保留	保留	PB9<1:0>	保留	保留	保留	保留	保留	PB8<1:0>	保留

—	bit31-30	—	—
PB15<1:0>	bit29-28	R/W	<b>PB15 复用选择位</b> 00~11: FUN0~FUN3
—	bit27-26	—	—
PB14<1:0>	bit25-24	R/W	<b>PB14 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PB13<1:0>	bit21-20	R/W	<b>PB13 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PB12<1:0>	bit17-16	R/W	<b>PB12 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PB11<1:0>	bit13-12	R/W	<b>PB11 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PB10<1:0>	bit9-8	R/W	<b>PB10 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PB9<1:0>	bit5-4	R/W	<b>PB9 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PB8<1:0>	bit1-0	R/W	<b>PB8 复用选择位</b> 00~11: FUN0~FUN3

**PB[23:16]端口复用选择寄存器 (GPIO\_PBFUN2)**

偏移地址: A8<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	PB23<1:0>		保留	PB22<1:0>		保留	PB21<1:0>		保留	PB20<1:0>		保留	PB19<1:0>		保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PB19<1:0>		保留	PB18<1:0>		保留	PB17<1:0>		保留	PB16<1:0>		保留	PB16<1:0>		保留

—	bit31-30	—	—
PB23<1:0>	bit29-28	R/W	<b>PB23 复用选择位</b> 00~11: FUN0~FUN3
—	bit27-26	—	—
PB22<1:0>	bit25-24	R/W	<b>PB22 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PB21<1:0>	bit21-20	R/W	<b>PB21 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PB20<1:0>	bit17-16	R/W	<b>PB20 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PB19<1:0>	bit13-12	R/W	<b>PB19 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PB18<1:0>	bit9-8	R/W	<b>PB18 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PB17<1:0>	bit5-4	R/W	<b>PB17 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PB16<1:0>	bit1-0	R/W	<b>PB16 复用选择位</b> 00~11: FUN0~FUN3

**PB 端口输入弱上拉使能寄存器 (GPIO\_PBUEN)**

偏移地址: B0<sub>H</sub>

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

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

—	bit31-24	—	—
PUEN<23:0>	bit23-0	R/W	<b>PB 端口弱上拉使能位</b> 0: 禁止 1: 使能

**PB 端口输入弱下拉使能寄存器 (GPIO\_PBPDEN)**

偏移地址: B4<sub>H</sub>

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

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

—	bit31-24	—	—
PDEN<23:0>	bit23-0	R/W	<b>PB 端口弱下拉使能位</b> 0: 禁止 1: 使能

**PB 端口输出开漏使能寄存器 (GPIO\_PBOD)**

偏移地址: B8<sub>H</sub>

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

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

—	bit31-24	—	—
OD<23:0>	bit23-0	R/W	<b>PB 输出开漏使能位</b> 0: 禁止 1: 使能

**PB 端口 SMIT 窗口选择寄存器 (GPIO\_PBSMITSEL)**

偏移地址: C0<sub>H</sub>

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

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

—	bit31-24	—	—
SMITSEL<7:0>	bit23-16	R/W	<b>PB 端口 SIMT 窗口选择位 (仅支持 PB23~PB16)</b>

—	bit15-0	—	1: VIL=0.8V VIH=2V 0: VIL=1.5V VIH=3V
—	—	—	—

**PC 端口方向控制寄存器 (GPIO\_PCDIR)**

偏移地址: 0100<sub>H</sub>

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

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

—	bit31-16	—	—
DIR<15:0>	Bit15-0	R/W	<b>PC 端口方向控制位</b> 0: 输出 1: 输入

**PC 端口方向控制置位寄存器 (GPIO\_PCDIRS)**

偏移地址: 0104<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-16	—	—
DIRS<15:0>	Bit15-0	W	<b>PC 端口方向控制置位选择</b> 0: 不改变端口方向 1: 端口方向设置为输入

**PC 端口方向控制清零寄存器 (GPIO\_PCDIRC)**

偏移地址: 0108<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-16	—	—
DIRC<15:0>	Bit15-0	W	<b>PC 端口方向控制清零选择</b> 0: 不改变端口方向

			1: 端口方向设置为输出
--	--	--	--------------

**PC 端口方向控制取反寄存器 (GPIO\_PCDIRI)**

偏移地址: 010C<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-16	—	—
DIRI<15:0>	Bit15-0	W	<b>PC 端口方向控制取反选择</b> 0: 不改变端口方向 1: 端口方向取反

**PC 端口数据寄存器 (GPIO\_PC)**

偏移地址: 0110<sub>H</sub>

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

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

—	bit31-16	—	—
DATA<15:0>	Bit15-0	R/W	<b>PC 端口输入/输出电平</b> 0: 低电平 1: 高电平

注: 对 GPIO\_PC 寄存器, 写操作时, 实际是写 PC 端口输出的数据; 读操作时, 实际是读取 PB 端口的电平状态。

**PC 端口数据置位寄存器 (GPIO\_PCS)**

偏移地址: 0114<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-16	—	—
DATAS<15:0>	Bit15-0	W	<b>PC 端口输出置位选择</b> 0: 不改变输出电平



			1: 相应端口输出高电平
--	--	--	--------------

**PC 端口数据清零寄存器 (GPIO\_PCC)**

偏移地址: 0118<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-16	—	—
DATA<15:0>	Bit15-0	W	<b>PC 端口输出清零选择</b> 0: 不改变输出电平 1: 相应端口输出低电平

**PC 端口输出取反寄存器 (GPIO\_PCI)**

偏移地址: 011C<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-16	—	—
DATA<15:0>	Bit15-0	W	<b>PC 端口输出取反选择</b> 0: 不改变输出电平 1: 相应端口输出取反

**PC[7:0]端口复用选择寄存器 (GPIO\_PCFUN0)**

偏移地址: 0120<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	PC7<1:0>	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留

—	bit31-30	—	—
PC7<1:0>	bit29-28	R/W	<b>PC7 复用选择位</b> 00~11: FUN0~FUN3

—	bit27-26	—	—
PC6<1:0>	bit25-24	R/W	<b>PC6 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PC5<1:0>	bit21-20	R/W	<b>PC5 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PC4<1:0>	bit17-16	R/W	<b>PC4 复用选择位</b> 00~11: FUN0~FUN3
—	bit15-14	—	—
PC3<1:0>	bit13-12	R/W	<b>PC3 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PC2<1:0>	bit9-8	R/W	<b>PC2 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PC1<1:0>	bit5-4	R/W	<b>PC1 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PC0<1:0>	bit1-0	R/W	<b>PC0 复用选择位</b> 00~11: FUN0~FUN3

**PC[15:8]端口复用选择寄存器 (GPIO\_PCFUN1)**

偏移地址: 0124<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	PC15<1:0>	保留	保留	保留	保留	PC14<1:0>	保留	保留	保留	PC13<1:0>	保留	保留	保留	保留	PC12<1:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	PC11<1:0>	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	PC8<1:0>

—	bit31-30	—	—
PC15<1:0>	bit29-28	R/W	<b>PC15 复用选择位</b> 00~11: FUN0~FUN3
—	bit27-26	—	—
PC14<1:0>	bit25-24	R/W	<b>PC14 复用选择位</b> 00~11: FUN0~FUN3
—	bit23-22	—	—
PC13<1:0>	bit21-20	R/W	<b>PC13 复用选择位</b> 00~11: FUN0~FUN3
—	bit19-18	—	—
PC12<1:0>	bit17-16	R/W	<b>PC12 复用选择位</b>

—	bit15-14	—	00~11: FUN0~FUN3
PC11<1:0>	bit13-12	R/W	<b>PC11 复用选择位</b> 00~11: FUN0~FUN3
—	bit11-10	—	—
PC10<1:0>	bit9-8	R/W	<b>PC10 复用选择位</b> 00~11: FUN0~FUN3
—	bit7-6	—	—
PC9<1:0>	bit5-4	R/W	<b>PC9 复用选择位</b> 00~11: FUN0~FUN3
—	bit3-2	—	—
PC8<1:0>	bit1-0	R/W	<b>PC8 复用选择位</b> 00~11: FUN0~FUN3

**PC 端口输入弱上拉使能寄存器 (GPIO\_PCPUEN)**

偏移地址: 0130<sub>H</sub>

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

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

—	bit31-16	—	—
PUEN<15:0>	Bit15-0	R/W	<b>PC 端口弱上拉使能位</b> 0: 禁止 1: 使能

**PC 端口输入弱下拉使能寄存器 (GPIO\_PCPDEN)**

偏移地址: 0134<sub>H</sub>

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

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

—	bit31-16	—	—
PDEN<15:0>	Bit15-0	R/W	<b>PC 端口弱下拉使能位</b> 0: 禁止 1: 使能

**PC 端口输出开漏使能寄存器 (GPIO\_PCOD)**

偏移地址: 0138<sub>H</sub>

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

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

—	bit31-16	—	—
OD<15:0>	Bit15-0	R/W	<b>PC 输出开漏使能位</b> 0: 禁止 1: 使能

**PINT 中断使能寄存器 (GPIO\_PIE)**

偏移地址: 00<sub>H</sub>

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

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

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

**PINT 中断标志寄存器 (GPIO\_PIF)**

偏移地址: 04<sub>H</sub>

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

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

—	bit31-8	—	—
PINTIF<7:0>	bit7-0	R/W	<b>PINT 中断标志位</b> 0: 无中断

			1: 有中断 软件写 1 清除中断标志位, 写 0 无效
--	--	--	---------------------------------

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

**PINT0~7 中断源选择寄存器 (GPIO\_PSEL)**

偏移地址: 08<sub>H</sub>

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

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

—	bit31	—	—
PINT7SEL<2:0>	bit30-28	R/W	<b>PINT7 输入选择位</b> 000~111: SEL0~SEL7
—	bit27	—	—
PINT6SEL<2:0>	bit26-24	R/W	<b>PINT6 输入选择位</b> 000~111: SEL0~SEL7
—	bit23	—	—
PINT5SEL<2:0>	bit22-20	R/W	<b>PINT5 输入选择位</b> 000~111: SEL0~SEL7
—	bit19	—	—
PINT4SEL<2:0>	bit18-16	R/W	<b>PINT4 输入选择位</b> 000~111: SEL0~SEL7
—	bit15	—	—
PINT3SEL<2:0>	bit14-12	R/W	<b>PINT3 输入选择位</b> 000~111: SEL0~SEL7
—	bit11	—	—
PINT2SEL<2:0>	bit10-8	R/W	<b>PINT2 输入选择位</b> 000~111: SEL0~SEL7
—	bit7	—	—
PINT1SEL<2:0>	bit6-4	R/W	<b>PINT1 输入选择位</b> 000~111: SEL0~SEL7
—	bit3	—	—
PINT0SEL<2:0>	bit2-0	R/W	<b>PINT0 输入选择位</b> 000~111: SEL0~SEL7

PINT	SEL0	SEL1	SEL2	SEL3	SEL4	SEL5	SEL6	SEL7
PINT0	PA0	PA8	PA16	PA24	PB0	PB8	PB16	—
PINT1	PA1	PA9	PA17	PA25	PB1	PB9	PB17	—
PINT2	PA2	PA10	PA18	PA26	PB2	PB10	PB18	—
PINT3	PA3	PA11	PA19	PA27	PB3	PB11	PB19	—
PINT4	PA4	PA12	PA20	PA28	PB4	PB12	PB20	—
PINT5	PA5	PA13	PA21	PA29	PB5	PB13	PB21	—
PINT6	PA6	PA14	PA22	PA30	PB6	PB14	PB22	—
PINT7	PA7	PA15	PA23	PA31	PB7	PB15	PB23	—

表 4-1 PINT 选择对应列表

**PINT 中断配置寄存器 (GPIO\_PCFG)**

偏移地址: 0C<sub>H</sub>

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

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

—	bit31	—	—
PINT7CFG<2:0>	bit30-28	R/W	<b>PINT7 配置位</b> 000: 上升沿 001: 下降沿 010: 高电平 011: 低电平 1xx: 上升沿和下降沿
—	bit27	—	—
PINT6CFG<2:0>	bit26-24	R/W	<b>PINT6 配置位</b> 000: 上升沿 001: 下降沿 010: 高电平 011: 低电平 1xx: 上升沿和下降沿
—	bit23	—	—
PINT5CFG<2:0>	bit22-20	R/W	<b>PINT5 配置位</b> 000: 上升沿 001: 下降沿 010: 高电平 011: 低电平 1xx: 上升沿和下降沿
—	bit19	—	—
PINT4CFG<2:0>	bit18-16	R/W	<b>PINT4 配置位</b> 000: 上升沿

			001: 下降沿 010: 高电平 011: 低电平 1xx: 上升沿和下降沿
—	bit15	—	—
PINT3CFG<2:0>	bit14-12	R/W	<b>PINT3 配置位</b> 000: 上升沿 001: 下降沿 010: 高电平 011: 低电平 1xx: 上升沿和下降沿
—	bit11	—	—
PINT2CFG<2:0>	bit10-8	R/W	<b>PINT2 配置位</b> 000: 上升沿 001: 下降沿 010: 高电平 011: 低电平 1xx: 上升沿和下降沿
—	bit7	—	—
PINT1CFG<2:0>	bit6-4	R/W	<b>PINT1 配置位</b> 000: 上升沿 001: 下降沿 010: 高电平 011: 低电平 1xx: 上升沿和下降沿
—	bit3	—	—
PINT0CFG<2:0>	bit2-0	R/W	<b>PINT0 配置位</b> 000: 上升沿 001: 下降沿 010: 高电平 011: 低电平 1xx: 上升沿和下降沿

## 第5章 外设

### 5.1 定时器/计数器

#### 5.1.1 16 位定时器/计数器T16N (T16N0/T16N1/T16N2/T16N3)

以 T16N0 为例，T16N1/T16N2/T16N3 同 T16N0。

##### 5.1.1.1 概述

- ◇ 支持定时/计数工作模式可配置
- ◇ 支持 1 组 16 位可配置定时/计数寄存器 T16N\_CNT
- ◇ 支持 1 组 8 位可配置预分频计数匹配寄存器 T16N\_PREMAT
- ◇ 支持 4 组 16 位计数匹配寄存器 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3，计数匹配后支持下列操作：
  - 产生中断
  - 支持 T16N\_CNT 计数寄存器三种操作：保持，清 0，或继续计数
  - 支持 T16N0OUT0/T16N0OUT1 端口四种操作：保持，清 0，置 1，或取反
- ◇ 支持输入捕捉功能
  - 支持捕捉边沿可配置
  - 支持捕捉次数可配置
- ◇ 支持输出调制功能 PWM

##### 5.1.1.2 结构框图

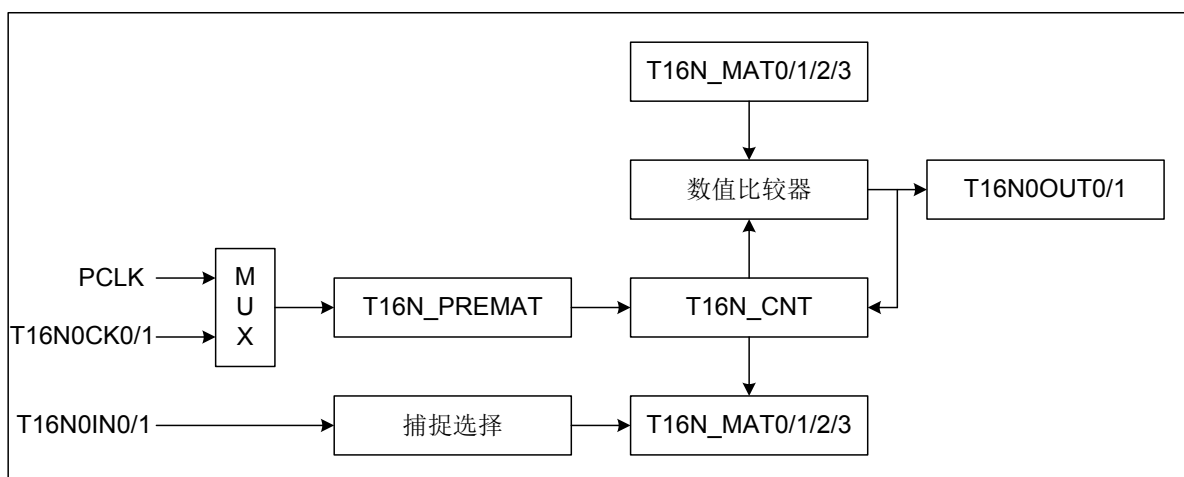


图 5-1 T16N 电路结构框图

##### 5.1.1.3 T16N定时/计数功能

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

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

设置 T16N\_CON0 寄存器的 CS<1:0>位，选择计数时钟源。时钟源为内部时钟 PCLK 时，



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

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

设置 T16N\_CON0 寄存器的 EDGE<1:0>位，选择外部时钟计数方式：上升沿计数，下降沿计数，或上升/下降沿均计数，其中上升/下降沿均计数只适用于同步计数模式。

设置 T16N\_CON0 寄存器的 MAT0S/MAT1S/MAT2S/MAT3S，选择计数匹配后 T16N\_CNT 计数值寄存器的工作状态。

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

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

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

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

当 T16N\_CON1 寄存器的 MOE0=1 时，如果 T16N\_CNT 的计数值匹配 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3，T16N0OUT0 端口输出翻转。

举例说明：T16N\_CNT 计数值匹配 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 后的工作方式

- 1) T16N\_MAT0<15:0>=0x0002, T16N\_CON0 寄存器的 MAT0S<1:0>=00, 继续计数, 不产生中断;
- 2) T16N\_MAT1<15:0>=0x0004, T16N\_CON0 寄存器的 MAT1S<1:0>=11, 继续计数, 产生中断;
- 3) T16N\_MAT2<15:0>=0x0006, T16N\_CON0 寄存器的 MAT2S<1:0>=10, 清 0, 产生中断, 重新计数。

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

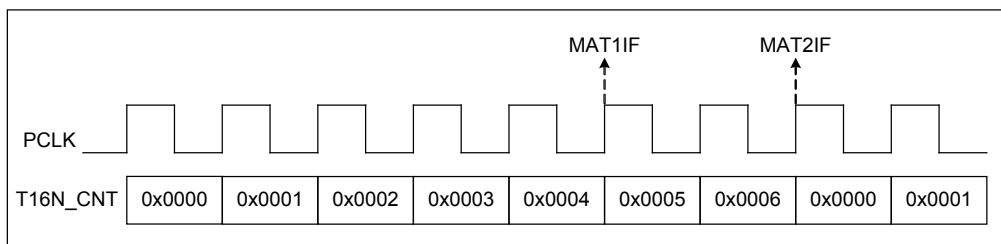


图 5-2 T16N 计数匹配功能示意图

#### 5.1.1.4 T16N输入捕捉功能

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

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

对端口 T16N0IN0 和 T16N0IN1 的状态进行检测。

当 T16N0IN0 端口的状态变化符合所设定的捕捉事件时，将 T16N\_CNT 和 T16N\_PRECNT 的当前值分别装载到 T16N\_MAT0 和 T16N\_MAT2 寄存器中，产生 T16N 的 CAP0IF 中断，并将 T16N\_CNT 和 T16N\_PRECNT 清零。

当 T16N0IN1 端口的状态变化符合所设定的捕捉事件时，将 T16N\_CNT 和 T16N\_PRECNT 的当前值分别装载到 T16N\_MAT1 和 T16N\_MAT3 寄存器中，产生 T16N 的 CAP1IF 中断，并将 T16N\_CNT 和 T16N\_PRECNT 清零。

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

设置 T16N\_CON1 寄存器的 CAPPE 位和 CAPNE 位，选择 T16N0IN0 和 T16N0IN1 端口信号的捕捉事件：捕捉上升沿，捕捉下降沿，捕捉上升沿/下降沿。

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

设置 T16N\_CON1 寄存器的 CAPT<3:0>位，可选择捕捉事件发生的次数。

当 T16N\_CON1 寄存器的 MOE0=1 时，如果 T16N0IN0 端口捕捉事件发生，T16N0OUT0 端口输出翻转；

当 T16N\_CON1 寄存器的 MOE1=1 时，如果 T16N0IN1 端口捕捉事件发生，T16N0OUT1 端口输出翻转。

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

- 1) T16N\_CON0 寄存器的 MOD<1:0>=10，CS<1:0>=00，MAT0S<1:0>=00；
- 2) T16N\_CON1 寄存器的 CAPPE=1，CAPNE=1，CAPIS=1，CAPT<3:0>=0111。

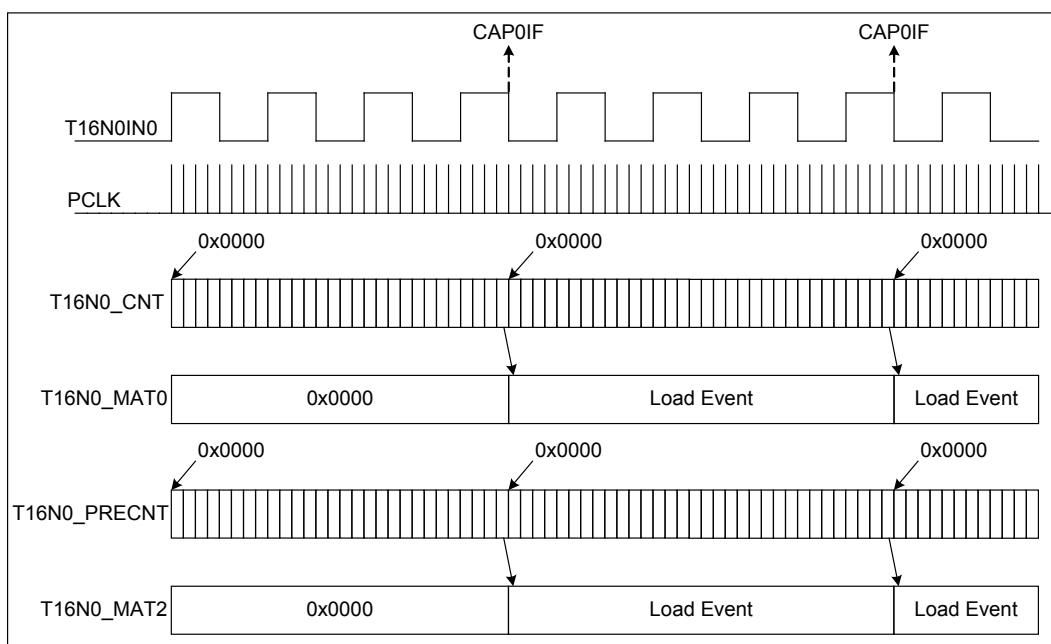


图 5-3 T16N 捕捉功能示意图

在捕捉工作模式下，修改 T16N0 预分频器计数匹配寄存器 T16N0\_PREMAT 时，预分频器计数不会被清零。因此，首次捕捉可以从一个非零预分频器计数开始。当捕捉事件匹配发生时，产生的中断标志位必须通过软件清除，并及时读取捕捉到 T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 寄存器的值，在下一次捕捉事件发生时，T16N\_MAT0/T16N\_MAT1/T16N\_MAT2/T16N\_MAT3 寄存器会装载为新的计数值。

### 5.1.1.5 T16N 输出调制功能

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

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

设置 T16N\_CON1 寄存器的 MOE0 位，选择 T16N0OUT0 是否使能为匹配输出端口，使能时启用 T16N\_MAT0/T16N\_MAT1 匹配寄存器和 T16N\_CNT 进行匹配；设置 T16N\_CON1 寄存器的 MOE1 位，选择 T16N0OUT1 是否使能为匹配输出端口，使能时启用 T16N\_MAT2/T16N\_MAT3 匹配寄存器和 T16N\_CNT 进行匹配。

设置 T16N\_CON1 寄存器 MOM0/MOM1，选择计数匹配发生时，对 T16N0OUT 端口的影响：保持，清 0，置 1，取反。

举例说明：在 T16N0OUT0 和 T16N0OUT1 端口，产生双边 PWM 波形。T16N\_CON1 寄存器中：

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

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

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

MOM2<1:0>=10; T16N\_MAT2 匹配，T16N0OUT1 输出高电平

MOM3<1:0>=01; T16N\_MAT3 匹配, T16N0OUT1 输出低电平

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

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

T16N\_CON0 寄存器的 MOD<1:0>=11; T16N 设置为调制输出, T16N\_CON0 寄存器中:

MAT0S<1:0>=11; T16N\_CNT 继续计数, 并产生中断

MAT1S<1:0>=11; T16N\_CNT 继续计数, 并产生中断

MAT2S<1:0>=11; T16N\_CNT 继续计数, 并产生中断

MAT3S<1:0>=10; T16N\_CNT 清 0, 并产生中断

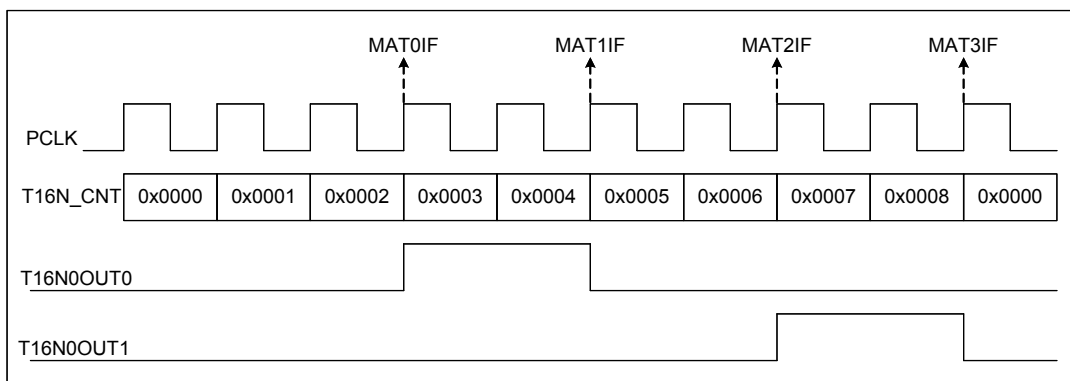


图 5-4 T16N 输出调制功能示意图

### 5.1.1.6 特殊功能寄存器

#### T16N 计数值寄存器 (T16N\_CNT)

偏移地址: 00<sub>H</sub>

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

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

—	bit31-16	—	—
CNT<15:0>	bit 15-0	R/W	T16N计数值

**T16N 控制寄存器 0 (T16N\_CON0)**

偏移地址: 04<sub>H</sub>

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

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

—	bit 31~17	—	未使用
ASYWEN	bit 16	R/W	外部时钟异步计数模式下, 对计数器的写使能位 0: 禁止写 T16N_CNT 和 T16N_PRECNT, 如果强制写, 有可能写操作不成功 1: 使能写 T16N_CNT 和 T16N_PRECNT
MAT3S<1:0>	bit 15~14	R/W	<b>T16N_CNT 匹配 T16N_MAT3 后的工作模式选择位</b> 00: T16N_CNT 继续计数, 不产生中断 01: T16N_CNT 保持, 产生中断 10: T16N_CNT 清 0 并重新计数, 产生中断 11: T16N_CNT 继续计数, 产生中断
MAT2S<1:0>	bit 13~12	R/W	<b>T16N_CNT 匹配 T16N_MAT2 后的工作模式选择位</b> 00: T16N_CNT 继续计数, 不产生中断 01: T16N_CNT 保持, 产生中断 10: T16N_CNT 清 0 并重新计数, 产生中断 11: T16N_CNT 继续计数, 产生中断
MAT1S<1:0>	bit 11~10	R/W	<b>T16N_CNT 匹配 T16N_MAT1 后的工作模式选择位</b> 00: T16N_CNT 继续计数, 不产生中断 01: T16N_CNT 保持, 产生中断 10: T16N_CNT 清 0 并重新计数, 产生中断 11: T16N_CNT 继续计数, 产生中断
MAT0S<1:0>	bit 9~8	R/W	<b>T16N_CNT 匹配 T16N_MAT0 后的工作模式选择位</b> 00: T16N_CNT 继续计数, 不产生中断 01: T16N_CNT 保持, 产生中断 10: T16N_CNT 清 0 并重新计数, 产生中断 11: T16N_CNT 继续计数, 产生中断
MOD<1:0>	bit 7~6	R/W	<b>工作模式选择位</b> 00: 定时/计数模式 01: 定时/计数模式 10: 捕捉模式 11: 调制模式
EDGE<1:0>	bit 5~4	R/W	<b>外部时钟计数边沿选择位</b> 00: 上升沿计数 01: 下降沿计数 10: 上升沿/下降沿均计数 (仅同步计数模式) 11: 上升沿/下降沿均计数 (仅同步计数模式)

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

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

**T16N 控制寄存器 1 (T16N\_CON1)**

偏移地址: 08<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MOM3<1:0>	MOM2<1:0>	MOM1<1:0>	MOM0<1:0>	保留										MOE1	MOE0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CAPT<3:0>			CAPIS1	CAPIS0	CAPNE	CAPPE	

MOM3<1:0>	bit31-30	R/W	<b>T16N_MAT3 匹配后的端口 1 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清 0 10: 匹配端口置 1 11: 匹配端口取反
MOM2<1:0>	bit29-28	R/W	<b>T16N_MAT2 匹配后的端口 1 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清 0 10: 匹配端口置 1 11: 匹配端口取反
MOM1<1:0>	bit27-26	R/W	<b>T16N_MAT1 匹配后的端口 0 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清 0 10: 匹配端口置 1 11: 匹配端口取反

MOM0<1:0>	bit25-24	R/W	<b>T16N_MAT0 匹配后的端口 0 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清 0 10: 匹配端口置 1 11: 匹配端口取反
—	bit23-18	—	—
MOE1	bit17	R/W	<b>输出端口 1 使能位</b> 0: 禁止 1: 使能
MOE0	bit16	R/W	<b>输出端口 0 使能位</b> 0: 禁止 1: 使能
—	Bit15-8	—	—
CAPT<3:0>	bit7-4	R/W	<b>捕捉次数控制位</b> 0: 捕捉 1 次后, 产生装载动作 1: 捕捉 2 次后, 产生装载动作 2: 捕捉 3 次后, 产生装载动作 ..... F: 捕捉 16 次后, 产生装载动作
CAPIS1	bit3	R/W	<b>捕捉输入端口 1 使能位</b> 0: 禁止 1: 使能
CAPIS0	bit2	R/W	<b>捕捉输入端口 0 使能位</b> 0: 禁止 1: 使能
CAPNE	bit1	R/W	<b>下降沿捕捉使能位</b> 0: 禁止 1: 使能
CAPPE	bit0	R/W	<b>上升沿捕捉使能位</b> 0: 禁止 1: 使能

**T16N 预分频器计数值寄存器 (T16N\_PRECNT)**

偏移地址: 10<sub>H</sub>

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

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

—	bit31-8	—	—
PRECNT<7:0>	bit7-0	R/W	T16N预分频器计数值

**T16N 预分频器计数比例寄存器 (T16N\_PREMAT)**

偏移地址: 14<sub>H</sub>

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

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

—	bit31-8	—	—
PREMAT<7:0>	bit7-0	R/W	<b>预分频比例设置位</b> 00: 预分频1: 1 01: 预分频1: 2 02: 预分频1: 3 ..... FE: 预分频1: 255 FF: 预分频1: 256

**T16N 中断使能寄存器 (T16N\_IE)**

偏移地址: 18<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CAP1IE	CAP0IE	IE	MAT3IE	MAT2IE	MAT1IE	MAT0IE	

—	bit31-7	—	—
CAP1IE	bit6	R/W	<b>输入端口 1 捕捉中断使能位</b> 0: 禁止 1: 使能
CAP0IE	bit5	R/W	<b>输入端口 0 捕捉中断使能位</b> 0: 禁止 1: 使能
IE	bit4	R/W	<b>匹配 0xFFFF 中断使能位</b> 0: 禁止 1: 使能
MAT3IE	bit3	R/W	<b>匹配 3 中断使能位</b> 0: 禁止 1: 使能
MAT2IE	bit2	R/W	<b>匹配 2 中断使能位</b> 0: 禁止



			1: 使能
MAT1IE	bit1	R/W	匹配 1 中断使能位 0: 禁止 1: 使能
MAT0IE	bit0	R/W	匹配 0 中断使能位 0: 禁止 1: 使能

**T16N 中断标志寄存器 (T16N\_IF)**

偏移地址: 1C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									CAP1IF	CAP0IF	IF	MAT3IF	MAT2IF	MAT1IF	MAT0IF

—	bit31-7	—	—
CAP1IF	bit6	R/W	输入端口 1 捕捉成功中断标志位 0: 输入端口 1 捕捉未成功 1: 输入端口 1 捕捉成功 软件写 1 清标志位, 写 0 无效
CAP0IF	bit5	R/W	输入端口 0 捕捉成功中断标志位 0: 输入端口 0 捕捉未成功 1: 输入端口 0 捕捉成功 软件写 1 清标志位, 写 0 无效
IF	bit4	R/W	匹配 0xFFFF 中断标志位 0: 计数器值不等于 0xFFFF 1: 计数器值等于 0xFFFF 软件写 1 清标志位, 写 0 无效
MAT3IF	bit3	R/W	匹配 3 中断标志位 0: 计数器值与匹配寄存器 3 不相等 1: 计数器值与匹配寄存器 3 相等 软件写 1 清标志位, 写 0 无效
MAT2IF	bit2	R/W	匹配 2 中断标志位 0: 计数器值与匹配寄存器 2 不相等 1: 计数器值与匹配寄存器 2 相等 软件写 1 清标志位, 写 0 无效
MAT1IF	bit1	R/W	匹配 1 中断标志位 0: 计数器值与匹配寄存器 1 不相等 1: 计数器值与匹配寄存器 1 相等 软件写 1 清标志位, 写 0 无效
MAT0IF	bit0	R/W	匹配 0 中断标志位

			0: 计数器值与匹配寄存器 0 不相等 1: 计数器值与匹配寄存器 0 相等 软件写 1 清标志位, 写 0 无效
--	--	--	---

注: 对 T16N\_IF 寄存器中的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

**T16N 计数匹配寄存器 0 (T16N\_MAT0)**

偏移地址: 20<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT0<15:0>															

—	bit31-16	—	—
MAT0<15:0>	bit15-0	R/W	T16N计数匹配值0

**T16N 计数匹配寄存器 1 (T16N\_MAT1)**

偏移地址: 24<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT1<15:0>															

—	bit31-16	—	—
MAT1<15:0>	bit15-0	R/W	T16N计数匹配值1

**T16N 计数匹配寄存器 2 (T16N\_MAT2)**

偏移地址: 28<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT2<15:0>															

—	bit31-16	—	—
MAT2<15:0>	bit15-0	R/W	T16N计数匹配值2

**T16N 计数匹配寄存器 3 (T16N\_MAT3)**

偏移地址: 2C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT3<15:0>															
—				bit31-16				—				—			
MAT3<15:0>				bit15-0				R/W				T16N计数匹配值3			

**5.1.1.7 T16N应用说明**

芯片支持 4 个 16 位定时器/计数器，分别为 T16N0，T16N1，T16N2，T16N3。

### 5.1.2 32 位定时器/计数器T32N (T32N0/T32N1/T32N2)

以 T32N0 为例，T32N1/T32N2 同 T32N0。

#### 5.1.2.1 概述

- ◇ 支持定时/计数工作模式可配置
- ◇ 支持 1 组 32 位可配置定时/计数寄存器 T32N\_CNT
- ◇ 支持 1 组 8 位可配置预分频计数匹配寄存器 T32N\_PREMAT
- ◇ 支持 4 组 32 位计数匹配寄存器 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3，计数匹配后支持下列操作：
  - 产生中断
  - 支持 T32N\_CNT 计数寄存器三种操作：保持，清 0，或继续计数
  - 支持 T32N0OUT0/T32N0OUT1 端口四种操作：保持，清 0，置 1，或取反
- ◇ 支持输入捕捉功能
  - 支持捕捉边沿可配置
  - 支持捕捉次数可配置
- ◇ 支持输出调制功能 PWM

#### 5.1.2.2 结构框图

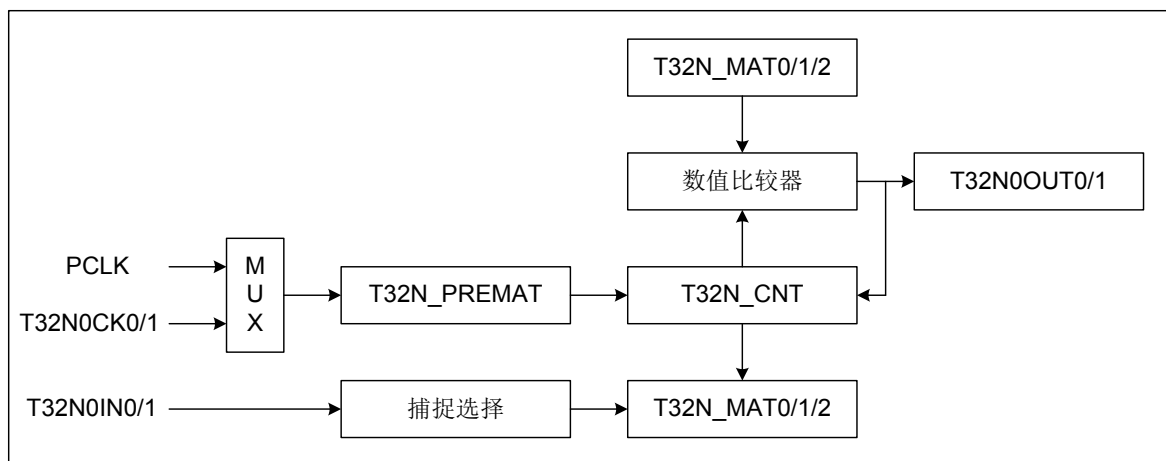


图 5-5 T32N 电路结构框图

#### 5.1.2.3 T32N定时/计数功能

设置 T32N\_CON0 寄存器的 MOD<1:0> =00 或 01，时 T32N 工作在定时/计数模式。

设置 T32N\_CON0 寄存器的 EN=1，使能 T32N，计数值寄存器 T32N\_CNT 从预设值开始累加计数。

设置 T32N\_CON0 寄存器的 CS<1:0>位，选择择计数时钟源。时钟源为内部时钟 PCLK 时，为定时模式；时钟源为外部时钟 T32N0CK0/T32N0CK1 端口输入时，为计数模式。

设置 T32N\_CON0 寄存器的 SYNC 位，选择外部时钟 T32N0CK0/T32N0CK1 是否被内部时钟 PCLK 同步。当选择外部时钟被同步时，为同步计数模式，否则为异步计数模式。同步计数模式时，T32N0CK0/T32N0CK1 端口输入的高/低电平脉宽必须分别大于 2 个

PCLK 时钟周期。

设置 T32N\_CON0 寄存器的 EDGE<1:0>位，选择外部时钟计数方式：上升沿计数，下降沿计数，或上升/下降沿均计数，其中上升/下降沿均计数只适用于同步计数模式。

设置 T32N\_CON0 寄存器的 MAT0S/MAT1S/MAT2S/MAT3S，选择计数匹配后 T32N\_CNT 计数值寄存器的工作状态。T32N0\_CON0 寄存器中：

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=00：当 T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 时，继续累加计数，不产生中断，当计数到 0xFFFFFFFF 后，下一次累加计数溢出，T32N0\_CNT 的值为 0x00000000，并产生中断，重新开始累加计数。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=01：当 T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 时，计数值将保持，即在下一个计数时钟（经过预分频之后的时钟）到来时，T32N\_CNT 不再累加计数，只产生中断。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=10：当 T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 时，计数值在下一个计数时钟（经过预分频之后的时钟）到来时被清 0，并产生中断，重新开始累加计数。

MAT0S/MAT1S/MAT2S/MAT3S<1:0>=11：当 T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 时，继续累加计数，并在下一个计数时钟（经过预分频之后的时钟）到来时，产生中断，当计数到 0xFFFFFFFF 后，下一次累加计数溢出，T32N\_CNT 的值为 0x00000000，并产生中断，重新开始累加计数。

当 MOE0=1 时，如果 T32N\_CNT 的计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3，T32N0OUT0 端口输出翻转。

举例说明：T32N\_CNT 计数值匹配 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 后的工作方式，T32N0\_CON0 寄存器中：

MAT0<31:0>=0x00000002， MAT0S<1:0>=00，继续计数，不产生中断；

MAT1<31:0>=0x00000004， MAT1S<1:0>=11，继续计数，产生中断；

MAT2<31:0>=0x00000006， MAT2S<1:0>=10，清 0，产生中断，重新计数。

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

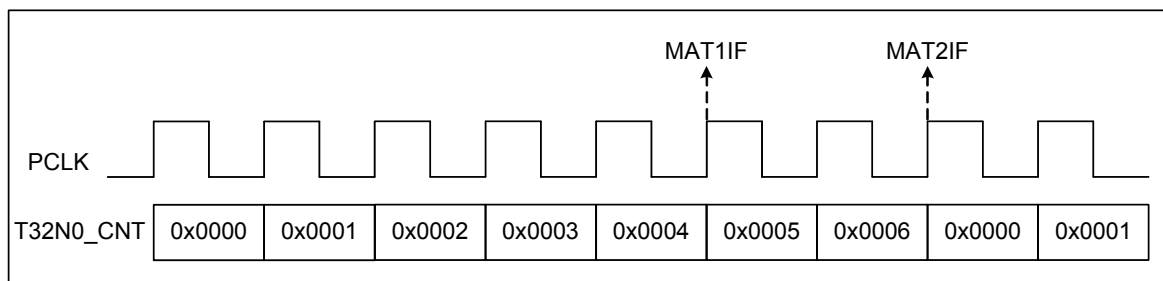


图 5-6 T32N 计数匹配功能示意图

#### 5.1.2.4 T32N输入捕捉功能

设置 T32N\_CON0 寄存器的 MOD<1:0>=10, 使 T32N 工作在捕捉模式。

在捕捉工作模式下, 需设置 T32N\_CON0 寄存器的 CS<1:0>=00, 使 T32N\_CNT 采用内部 PCLK 时钟源计数; 并且设置 T32N\_CON0 寄存器的 MAT0S/MAT1S/MAT2S/MAT3S <1:0>=00, 计数匹配不影响 T32N\_CNT 的工作。

对端口 T32N0IN0 和 T32N0IN1 的状态进行检测。

当 T32N0IN0 端口的状态变化符合所设定的捕捉事件时, 将 T32N\_CNT 和 T32N\_PRECNT 的当前值分别装载到 T32N\_MAT0 和 T32N\_MAT2 寄存器中, 产生 CAP0IF 中断, 并将 T32N\_CNT 和 T32N\_PRECNT 清零。

当 T32N0IN1 端口的状态变化符合所设定的捕捉事件时, 将 T32N\_CNT 和 T32N\_PRECNT 的当前值分别装载到 T32N\_MAT1 和 T32N\_MAT3 寄存器中, 产生 CAP1IF 中断, 并将 T32N\_CNT 和 T32N\_PRECNT 清零。

当 T32N\_CNT 计数直到溢出时, 仍未检测到设定的捕捉事件, T32N\_CNT 的值被清零, 并重新开始累加计数。

设置 T32N\_CON1 寄存器的 CAPPE 位和 CAPNE 位, 选择 T32N0IN0 和 T32N0IN1 端口信号的捕捉事件: 捕捉上升沿, 捕捉下降沿, 捕捉上升沿/下降沿。

设置 T32N\_CON1 寄存器的 CAPIS0 位, 选择 T32N0IN0 是否作为捕捉输入端口; 设置 T32N\_CON1 寄存器的 CAPIS1 位, 选择 T32N0IN1 是否作为捕捉输入端口; 可同时选择两个端口作为捕捉输入端口。

设置 T32N\_CON1 寄存器的 CAPT<3:0>位, 可选择捕捉事件发生的次数。

当 T32N\_CON1 寄存器的 MOE0=1 时, 如果 T32N0IN0 端口捕捉事件发生, T32N0OUT0 端口输出翻转;

当 T32N\_CON1 寄存器的 MOE1=1 时, 如果 T32N0IN1 端口捕捉事件发生, T32N0OUT1 端口输出翻转。

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

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

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

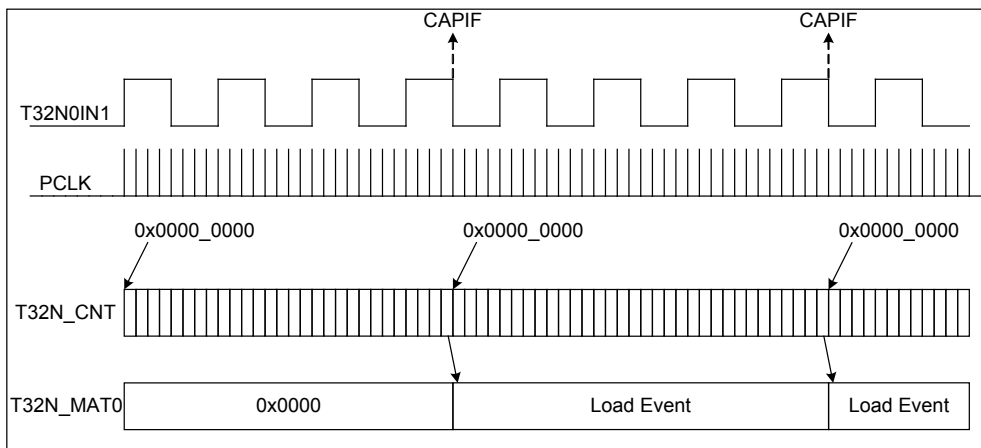


图 5-7 T32N 捕捉功能示意图

在捕捉工作模式时，修改 T32N 预分频器计数匹配寄存器 T32N\_PREMAT 时，预分频器计数不会被清零。因此，首次捕捉可以从一个非零预分频器计数开始。当捕捉事件匹配发生时，产生的中断标志位必须通过软件清除，并及时读取捕捉到的 T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 寄存器的值，在下次捕捉事件发生时，T32N\_MAT0/T32N\_MAT1/T32N\_MAT2/T32N\_MAT3 寄存器会装载为新的计数值。

### 5.1.2.5 T32N 输出调制功能

设置 T32N\_CON0 寄存器的 MOD<1:0>=11，使 T32N 工作在调制模式。

在调制工作模式下，需设置 T32N\_CON0 寄存器的 CS<1:0>=00，使 T32N\_CNT 采用内部 PCLK 时钟源计数。

设置 T32N\_CON1 寄存器的 MOE0 位，选择 T32N0OUT0 是否使能为匹配输出端口，使能时启用 T32N\_MAT0/T32N\_MAT1 匹配寄存器和 T32N\_CNT 进行匹配；设置 T32N\_CON1 寄存器的 MOE1，选择 T32N0OUT1 是否使能为匹配输出端口，使能时启用 T32N\_MAT2/T32N\_MAT3 匹配寄存器和 T32N\_CNT 进行匹配。

设置 T32N\_CON1 寄存器的 MOM0/1，选择计数匹配发生时，对 T32N0OUT 端口的影响：保持，清 0，置 1，取反。

举例说明：在 T32N0OUT0 和 T32N0OUT1 端口，产生双边 PWM 波形。

T32N\_CON1 寄存器中

MOE0=1, MOE1=1; T32N0OUT0 和 T32N0OUT1 匹配输出端口使能

MOM0<1:0>=10; T32N\_MAT0 匹配, T32N0OUT0 输出高电平

MOM1<1:0>=01; T32N\_MAT1 匹配, T32N0OUT0 输出低电平

MOM2<1:0>=10; T32N\_MAT2 匹配, T32N0OUT1 输出高电平

MOM3<1:0>=01; T32N\_MAT3 匹配, T32N0OUT1 输出低电平

T32N\_MAT0 = 0x0002; T32N\_MAT1 = 0x0004;

T32N\_MAT2 = 0x0006; T32N\_MAT3 = 0x0008;

T32N\_CON0 寄存器中 MOD<1:0>=11; T32N 设置为调制输出

MAT0S<1:0>=11; T32N\_CNT 继续计数, 并产生中断

MAT1S<1:0>=11; T32N\_CNT 继续计数, 并产生中断

MAT2S<1:0>=11; T32N\_CNT 继续计数, 并产生中断

MAT3S<1:0>=10; T32N\_CNT 清 0, 并产生中断

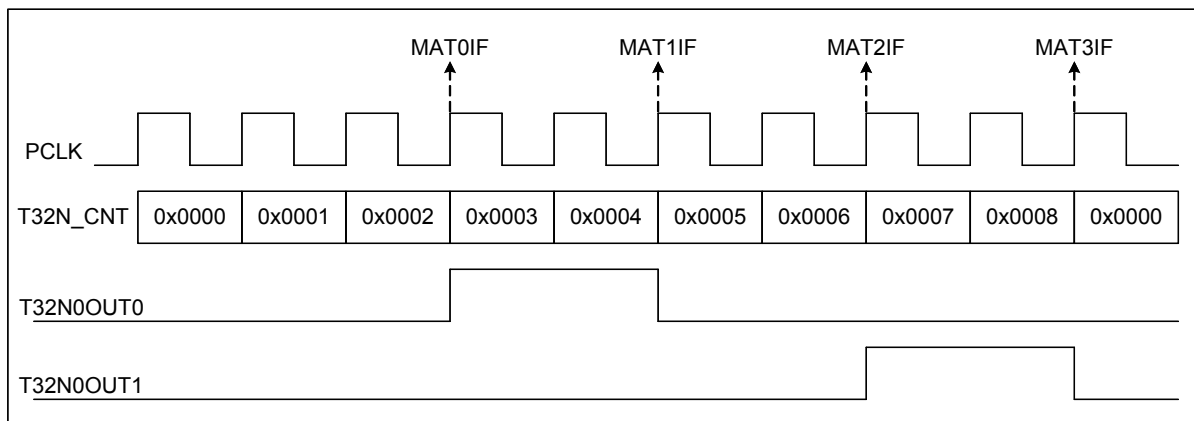


图 5-8 T32N 输出调制功能示意图

### 5.1.2.6 特殊功能寄存器

#### T32N 计数值寄存器 (T32N\_CNT)

偏移地址: 00<sub>H</sub>

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

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

CNT<31:0>	bit 31-0	R/W	T32N计数值
-----------	----------	-----	---------

#### T32N 控制寄存器 0 (T32N\_CON0)

偏移地址: 04<sub>H</sub>

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

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

—	bit31-17	—	—
ASYNCWREN	bit16	R/W	外部时钟异步计数模式下, 对计数器的写使能位 0: 禁止写 T32N_CNT 和 T32N_PRECNT, 如果强



			制写，有可能写操作不成功 1: 使能写T32N_CNT和T32N_PRECNT
MAT3S<1:0>	bit15-14	R/W	<b>T32N_CNT 匹配 T32N_MAT3 后的工作模式选择位</b> 00: T32N_CNT 继续计数，不产生中断 01: T32N_CNT 保持，产生中断 10: T32N_CNT 清 0 并重新计数，产生中断 11: T32N_CNT 继续计数，产生中断
MAT2S<1:0>	bit13-12	R/W	<b>T32N_CNT 匹配 T32N_MAT2 后的工作模式选择位</b> 00: T32N_CNT 继续计数，不产生中断 01: T32N_CNT 保持，产生中断 10: T32N_CNT 清 0 并重新计数，产生中断 11: T32N_CNT 继续计数，产生中断
MAT1S<1:0>	bit11-10	R/W	<b>T32N_CNT 匹配 T32N_MAT1 后的工作模式选择位</b> 00: T32N_CNT 继续计数，不产生中断 01: T32N_CNT 保持，产生中断 10: T32N_CNT 清 0 并重新计数，产生中断 11: T32N_CNT 继续计数，产生中断
MAT0S<1:0>	bit9-8	R/W	<b>T32N_CNT 匹配 T32N_MAT0 后的工作模式选择位</b> 00: T32N_CNT 继续计数，不产生中断 01: T32N_CNT 保持，产生中断 10: T32N_CNT 清 0 并重新计数，产生中断 11: T32N_CNT 继续计数，产生中断
MOD<1:0>	bit7-6	R/W	<b>工作模式选择位</b> 00: 定时/计数模式 01: 定时/计数模式 10: 捕捉模式 11: 调制模式
EDGE<1:0>	bit5-4	R/W	<b>外部时钟计数边沿选择位</b> 00: 上升沿计数 01: 下降沿计数 10: 上升沿/下降沿均计数（仅同步计数模式） 11: 上升沿/下降沿均计数（仅同步计数模式）
SYNC	bit3	R/W	<b>外部时钟同步使能位</b> 0: 不同步外部时钟 T32N0CK0/T32N0CK1，为异步计数模式 1: 通过 PCLK 对外部时钟 T32N0CK0/T32N0CK1 同步，为同步计数模式，外部时钟的高/低电平至少保持 2 个 PCLK 时钟周期
CS<1:0>	bit2-1	R/W	<b>T32N 计数时钟源选择位</b> 00: 内部时钟 PCLK 01: 外部时钟 T32N0CK0 时钟输入 10: 外部时钟 T32N0CK1 时钟输入 11: 内部时钟 PCLK
EN	bit0	R/W	<b>T32N使能位</b>

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

**T32N 控制寄存器 1 (T32N\_CON1)**

偏移地址: 08<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
MOM3<1:0>				MOM2<1:0>		MOM1<1:0>		MOM0<1:0>		保留						MOE1	MOE0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
保留								CAPT<3:0>			CAPIS1	CAPIS0	CAPNE	CAPPE			

MOM3<1:0>	bit31-30	R/W	<b>T32N_MAT3 匹配后的端口 1 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清 0 10: 匹配端口置 1 11: 匹配端口取反
MOM2<1:0>	bit29-28	R/W	<b>T32N_MAT2 匹配后的端口 1 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清 0 10: 匹配端口置 1 11: 匹配端口取反
MOM1<1:0>	bit27-26	R/W	<b>T32N_MAT1 匹配后的端口 0 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清 0 10: 匹配端口置 1 11: 匹配端口取反
MOM0<1:0>	bit25-24	R/W	<b>T32N_MAT0 匹配后的端口 0 工作模式选择位</b> 00: 匹配端口保持 01: 匹配端口清 0 10: 匹配端口置 1 11: 匹配端口取反
—	bit23-18	—	—
MOE1	bit17	R/W	<b>输出端口 1 使能位</b> 0: 禁止 1: 使能
MOE0	bit16	R/W	<b>输出端口 0 使能位</b> 0: 禁止 1: 使能

—	bit15-8	—	—
CAPT<3:0>	bit7-4	R/W	<b>捕捉次数控制位</b> 0: 捕捉 1 次后, 产生装载动作 1: 捕捉 2 次后, 产生装载动作 2: 捕捉 3 次后, 产生装载动作 ..... F: 捕捉 16 次后, 产生装载动作
CAPIS1	bit3	R/W	<b>捕捉输入端口 1 使能位</b> 0: 禁止 1: 使能
CAPIS0	bit2	R/W	<b>捕捉输入端口 0 使能位</b> 0: 禁止 1: 使能
CAPNE	bit1	R/W	<b>下降沿捕捉使能位</b> 0: 禁止 1: 使能
CAPPE	bit0	R/W	<b>上升沿捕捉使能位</b> 0: 禁止 1: 使能

**T32N 预分频器计数值寄存器 (T32N\_PRECNT)**

偏移地址: 10<sub>H</sub>

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

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

—	bit31-8	—	—
PRECNT<7:0>	bit7-0	R/W	T32N预分频器计数值

**T32N 预分频器计数比例寄存器 (T32N\_PREMAT)**

偏移地址: 14<sub>H</sub>

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

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

—	bit31-8	—	—
---	---------	---	---

PREMAT<7:0>	bit7-0	R/W	<p>预分频比例设置</p> <p>00: 预分频1: 1</p> <p>01: 预分频1: 2</p> <p>02: 预分频1: 3</p> <p>.....</p> <p>FE: 预分频1: 255</p> <p>FF: 预分频1: 256</p>
-------------	--------	-----	--

**T32N 中断使能寄存器 (T32N\_IE)**

偏移地址: 18<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									CAP1IE	CAP0IE	IE	MAT3IE	MAT2IE	MAT1IE	MAT0IE

—	bit31-7	—	—
CAP1IE	bit6	R/W	<p>输入端口 1 捕捉中断使能位</p> <p>0: 禁止</p> <p>1: 使能</p>
CAP0IE	bit5	R/W	<p>输入端口 0 捕捉中断使能位</p> <p>0: 禁止</p> <p>1: 使能</p>
IE	bit4	R/W	<p>匹配 0xFFFFFFFF 中断使能位</p> <p>0: 禁止</p> <p>1: 使能</p>
MAT3IE	bit3	R/W	<p>匹配 3 中断使能位</p> <p>0: 禁止</p> <p>1: 使能</p>
MAT2IE	bit2	R/W	<p>匹配 2 中断使能位</p> <p>0: 禁止</p> <p>1: 使能</p>
MAT1IE	bit1	R/W	<p>匹配 1 中断使能位</p> <p>0: 禁止</p> <p>1: 使能</p>
MAT0IE	bit0	R/W	<p>匹配 0 中断使能位</p> <p>0: 禁止</p> <p>1: 使能</p>

**T32N 中断标志寄存器 (T32N\_IF)**

偏移地址: 1C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									CAP1IF	CAP0IF	IF	MAT3IF	MAT2IF	MAT1IF	MAT0IF

—	bit31-7	—	—
CAP1IF	bit6	R/W	<b>输入端口 1 捕捉成功中断标志位</b> 0: 输入端口 1 捕捉未成功 1: 输入端口 1 捕捉成功 软件写 1 清标志位, 写 0 无效
CAP0IF	bit5	R/W	<b>输入端口 0 捕捉成功中断标志位</b> 0: 输入端口 0 捕捉未成功 1: 输入端口 0 捕捉成功 软件写 1 清标志位, 写 0 无效
IF	bit4	R/W	<b>匹配 0xFFFFFFFF 中断标志位</b> 0: 计数器值不等于 0xFFFFFFFF 1: 计数器值等于 0xFFFFFFFF 软件写 1 清标志位, 写 0 无效
MAT3IF	bit3	R/W	<b>匹配 3 中断标志位</b> 0: 计数器值与匹配寄存器 3 不相等 1: 计数器值与匹配寄存器 3 相等 软件写 1 清标志位, 写 0 无效
MAT2IF	bit2	R/W	<b>匹配 2 中断标志位</b> 0: 计数器值与匹配寄存器 2 不相等 1: 计数器值与匹配寄存器 2 相等 软件写 1 清标志位, 写 0 无效
MAT1IF	bit1	R/W	<b>匹配 1 中断标志位</b> 0: 计数器值与匹配寄存器 1 不相等 1: 计数器值与匹配寄存器 1 相等 软件写 1 清标志位, 写 0 无效
MAT0IF	bit0	R/W	<b>匹配 0 中断标志位</b> 0: 计数器值与匹配寄存器 0 不相等 1: 计数器值与匹配寄存器 0 相等 软件写 1 清标志位, 写 0 无效

注: 对 T32N\_IF 寄存器中的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

**T32N 计数匹配寄存器 0 (T32N\_MAT0)**

偏移地址: 20<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAT0<31:16>															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT0<15:0>															

MAT0<31:0>	bit31-0	R/W	T32N计数匹配值0
------------	---------	-----	------------

**T32N 计数匹配寄存器 1 (T32N\_MAT1)**

偏移地址: 24<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAT1<31:16>															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT1<15:0>															

MAT1<31:0>	bit31-0	R/W	T32N计数匹配值1
------------	---------	-----	------------

**T32N 计数匹配寄存器 2 (T32N\_MAT2)**

偏移地址: 28<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAT2<31:16>															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT2<15:0>															

MAT2<31:0>	bit31-0	R/W	T32N计数匹配值2
------------	---------	-----	------------

**T32N 计数匹配寄存器 3 (T32N\_MAT3)**

偏移地址: 2C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAT3<31:16>															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAT3<15:0>															

MAT3<31:0>	bit31-0	R/W	T32N计数匹配值3
------------	---------	-----	------------

**5. 1. 2. 7 T32N应用说明**

芯片支持 3 个 32 位定时器/计数器，分别为 T32N0，T32N1，T32N2。

## 5.2 通用异步接收/发送器 (UART0/UART1/UART2/UART3)

以 UART0 为例，UART1/UART2/UART3 同 UART0。

### 5.2.1 概述

- ◇ 支持异步接收和异步发送
- ◇ 支持全/半双工通讯模式
- ◇ 支持 7/8/9 位数据传输格式
- ◇ 支持 4 级发送缓冲寄存器和 4 级接收缓冲寄存器
- ◇ 支持通讯传输波特率可配置
- ◇ 兼容 RS-232/RS-442/RS-485 的通讯接口
- ◇ 支持 PWM 调制输出，且 PWM 占空比线性可调
- ◇ 支持 UART 输入输出通讯端口极性可配置
- ◇ UART 接收端口支持红外唤醒功能

### 5.2.2 结构框图

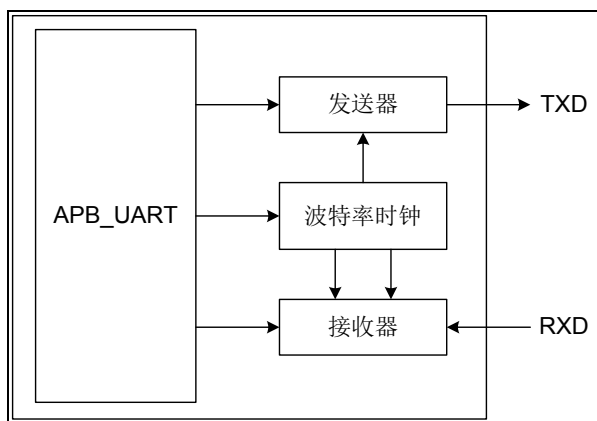


图 5-9 UART 电路结构图

### 5.2.3 UART 数据格式

UART 通讯每帧数据由 1 位起始位，7/8/9 位数据位、可配置奇偶校验位和停止位组成。UART\_CON 寄存器中配置 TXMOD 和 RXMOD 选择发送和接收的数据格式，配置 TXFS 选择发送 1 位或 2 位停止位。接收数据时，只判断第 1 位停止位，若不为高电平则产生“帧错误”中断标志。在没有数据传输时，通讯端口处于高电平状态。

帧数据格式如下图所示：

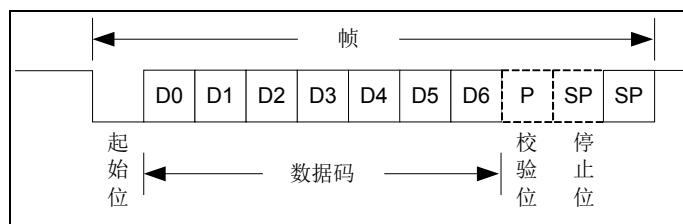


图 5-10 UART 7 位数据格式

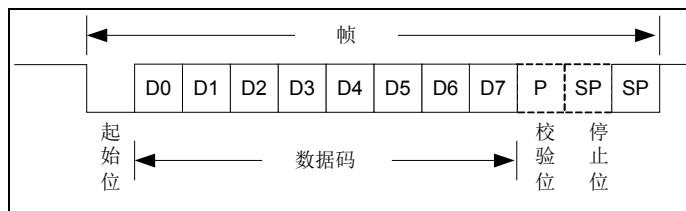


图 5-11 UART 8 位数据格式

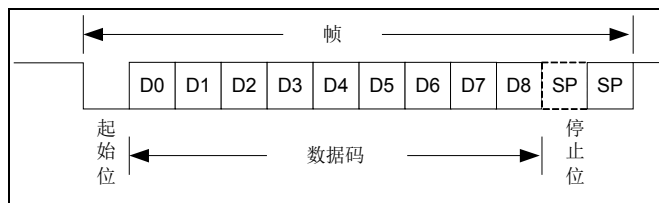


图 5-12 UART 9 位数据格式

数据发送和接收时，均是低位在前，高位在后，即先发送或接收数据的 LSB。通过发送数据寄存器 UART\_TBW 写入发送的数据，通过接收数据寄存器 UART\_RBR 读取接收的数据。

### 5.2.4 UART 异步发送器

发送数据时，起始位 START 和停止位 STOP 由芯片硬件电路自动产生，用户只需要配置相应的 I/O 端口复用功能；配置 UART\_BRR 和 UART\_CON 寄存器的 BCS，设定传输波特率；UART\_CON 寄存器中配置 TXMOD，选择发送的数据格式；配置 TXFS，选择发送的停止位数；配置 TXEN，使能数据发送；将要发送的数据写入发送数据寄存器 UART\_TBW，就可以开始数据的异步发送。如果数据格式支持奇偶校验位，硬件电路会根据相应的数据位产生校验位，在数据位后自动发送校验位。

配置 UART\_CON 寄存器的 TXP，可选择发送端口极性。选择为正极性时，发送端口的数据与被发送数据一致；选择为负极性时，发送端口的数据与被发送数据相反，即被发送数据位为 1 时，发送端口的数据位为 0。

支持 4 级发送缓冲器 TB0, TB1, TB2, TB3 和 1 级发送移位寄存器，可进行数据的连续发送，直到发送缓冲器和移位寄存器全空，最多可连续写入和发送 5 帧数据，配置 UART\_CON 寄存器的 TXFS 可选择相邻两帧数据发送的时间间隔。发送缓冲器 TB0/1/2/3 为只读寄存器（位于寄存器 UART\_TB01 和 UART\_TB23 中），只能通过发送数据寄存器 UART\_TBW 写入。

发送数据寄存器 UART\_TBW 为一个虚拟地址单元，物理上不存在实际的寄存器电路，写该寄存器地址单元时，实际上是将发送数据写入到发送缓冲器 TB0/TB1/TB2/TB3 中，再传输到发送移位寄存器，通过发送端口 TX0 进行数据发送。

发送数据寄存器 UART\_TBW 支持 3 种写入方式：字节写入，半字写入和字写入。

当发送数据为 7 位和 8 位数据格式时：字节方式写入 UART\_TBW 时，发送数据被写入到 UART 的发送缓冲器 TB0；半字方式写入 UART\_TBW 时，发送数据被同时写入到 UART 的发送缓冲器 TB0 和 TB1，其中低字节存放在 TB1 中；字方式写入 UART\_TBW 时，发送数据被同时写入到 UART 的发送缓冲器 TB0, TB1, TB2 和 TB3，其中低字节存放在 TB3 中。



当发送数据为 9 位数据格式时：只能以半字或字方式写入 UART\_TBW，且发送数据均被写入到 UART 的发送缓冲器 TB0。

发送数据从写入到发送到端口的数据流示意图如下所示：

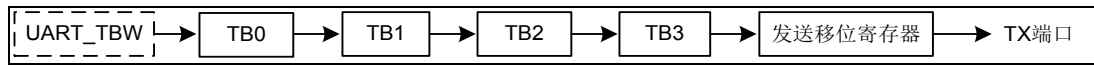


图 5-13 UART 发送数据流示意图

UART 的发送缓冲器 TB0/TB1/TB2/TB3 的数据移到下一级缓冲器或发送移位寄存器后，会置起发送空标志 TBEF0/TBEF1/TBEF2/TBEF3。当缓冲器 TB0 的发送空标志 TBEF0 为 0 时，表示 4 级发送缓冲器和发送移位寄存器均满，此时如果仍继续写入 UART\_TBW，则会置起发送数据错误中断标志 TBWEIF，同时新写入的数据无效，缓冲器数据仍保持。

当 4 级发送缓冲器和发送移位寄存器均空时，会置起 UART\_IF 寄存器中发送空闲标志 TXIDLEIF，表示当前的数据发送进程结束。

支持发送缓冲器空中断，配置 UART\_IE 寄存器的 TBIM，可选择中断模式：

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

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

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

UART\_CON 寄存器中配置 TRST，可将异步发送器软件复位，复位后：禁止数据发送 TXEN=0；UART\_IE 寄存器中禁止发送相关中断 TBIE=0，TBWEIE=0；UART\_IF 寄存器中复位相关中断标志为默认值 TBIF=1，TBWEIF=0；置起发送空闲标志 TXIDLEIF=1；UART\_TB01/UART\_TB23 寄存器中置起各发送缓冲器空标志 TBEF0/TBEF1/TBEF2/TBEF3=1。

发送数据的操作流程图示例如下：

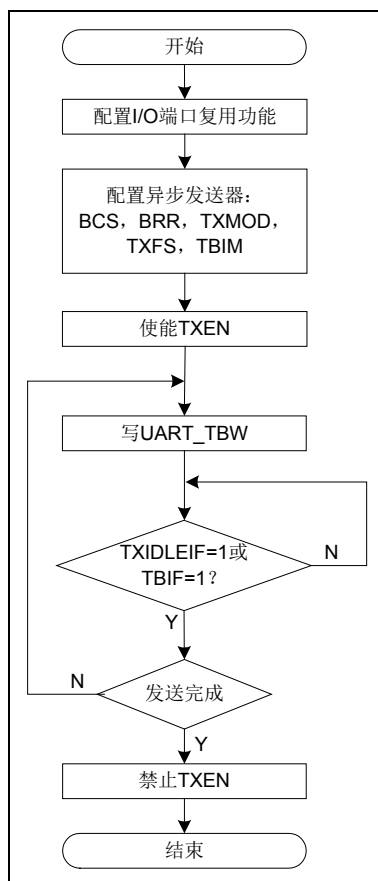


图 5-14 UART 发送数据操作流程示意图

### 5.2.5 UART异步接收器

接收数据时，配置相应的 I/O 端口复用功能；配置 UART\_BRR 和 UART\_CON 寄存器的 BCS 位，设定传输波特率；配置 UART\_CON 寄存器的 RXMOD，选择接收的数据格式；配置 UART\_CON 寄存器的 RXEN，使能数据接收，就可以开始数据的异步接收。如果数据格式支持奇偶校验位，硬件电路会自动判断奇偶校验位是否正确，若不正确则会置起 UART\_IF 寄存器中奇偶校验错误中断标志 PEIF。如果接收到的第 1 位停止位不为高电平，则会置起帧错误中断标志 FEIF。

配置 UART\_CON 寄存器的 RXP，可选择接收端口极性。选择为正极性时，接收端口的数据即作为接收数据；选择为负极性时，接收端口的数据取反后，作为接收数据，即接收端口的数据位为 1 时，接收的数据位为 0。

支持 4 级接收缓冲器 RB0, RB1, RB2, RB3（位于寄存器 UART\_RB01 和 UART\_RB23 中）和 1 级接收移位寄存器，可进行数据的连续接收，直到接收缓冲器和移位寄存器全满，最多可连续接收 5 帧数据，再执行数据读取操作。读取接收数据寄存器 UART\_RBR，可得到接收的数据，对应的接收缓冲器清除满标志 RBFF0/RBFF1/RBFF2/RBFF3（位于寄存器 UART\_RB01 和 UART\_RB23 中）；也可以读取接收缓冲器 RB0/RB1/RB2/RB3 得到接收的数据，但不会清除满标志位 RBFF0/RBFF1/RBFF2/RBFF3。

接收数据寄存器 UART\_RBR 为一个虚拟地址单元，物理上不存在实际的寄存器电路，读该寄存器地址单元时，实际上是读取接收缓冲器 RB0/RB1/RB2/RB3 中的数据。

接收数据寄存器 UART\_RBR 支持 3 种读取方式：字节读取，半字读取和字读取。

当接收数据为 7 位和 8 位数据格式时：字节方式读取 UART\_RBR 时，实际是读取接收缓冲器 RB0 的数据；半字方式读取 UART\_RBR 时，实际是同时读取接收缓冲器 RB0 和 RB1 的数据，其中 RB0 中的数据为低字节；字方式读取 UART\_RBR 时，实际是同时读取接收缓冲器 RB0, RB1, RB2 和 RB3，其中 RB0 中的数据为低字节。

当接收数据为 9 位数据格式时：只能以半字或字方式读取 UART\_RBR，且均是读取接收缓冲器 RB0 的数据。

接收数据从接收端口到各级缓冲器的数据流示意图如下所示：

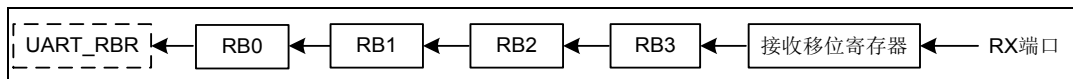


图 5-15 UART 接收数据流示意图

接收缓冲器 RB0/RB1/RB2/RB3 的数据移到下一级缓冲器后，会清除 UART\_RB01/UART\_RB23 寄存器中的满标志位 RBFF0/RBFF1/RBFF2/RBFF3。

当 4 级接收缓冲器和 1 级接收移位寄存器均满时，如果再次接收到数据起始位，会置起 UART\_IF 寄存器中接收数据溢出中断标志 ROIF，同时不会接收新数据，缓冲器数据仍保持。

当 4 级接收缓冲器和接收移位寄存器均空时，会置起 UART\_IF 寄存器中的接收空闲标志 RXIDLEIF，表示当前无数据接收。

支持接收缓冲器满中断，配置 UART\_IE 寄存器中 RBIM，可选择中断模式。

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

RBIM<1:0>=01，为半字满产生中断，即接收缓冲器 RB0 和 RB1 均为满时，会置起中断标志 RBIF；

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

配置 UART\_CON 寄存器中 RRST 位，可将异步接收器软件复位，复位后：禁止数据接收 RXEN=0；禁止接收相关中断 RBIE=0, ROIE=0, FEIE=0, PEIE=0；复位相关中断标志为默认值 RBIF=0, ROIF=0, FEIF=0, PEIF=0；置起接收空闲标志 RXIDLEIF=1；清除各接收缓冲器满标志 RBFF0/RBFF1/RBFF2/RBFF3=0；清除各接收缓冲器错误标志 FE0/FE1/FE2/FE3=0, PE0/PE1/PE2/PE3=0。

接收数据的操作流程图示例如下：

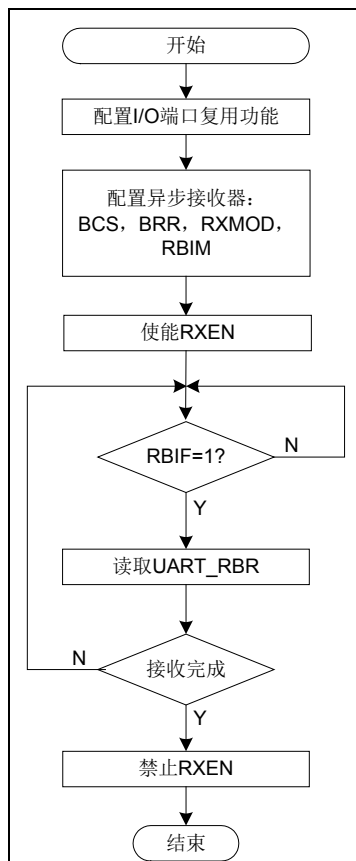


图 5-16 UART 接收数据操作流程图示例

### 5.2.6 UART红外发送调制功能

红外发送调制模式是将 UART 传输到发送端口的信号电平，由内部独立 38K PWM 信号源进行调制后，再从发送端口 TX0 输出。在寄存器 GPIO\_TXPC 中，配置 TX0PS，可启用 UART 红外发送模式。配置 TX0PLV，选择发送端口 TX0 被调制的信号电平。

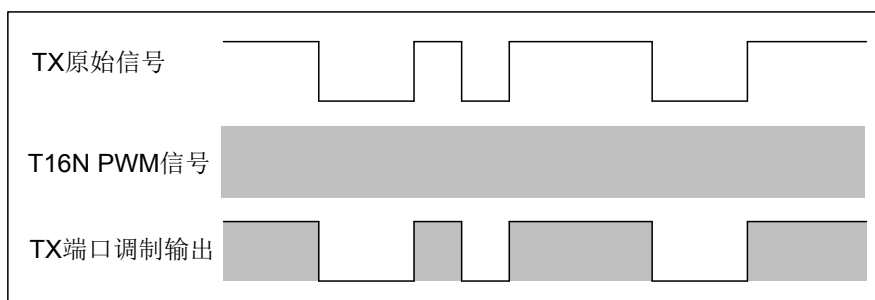


图 5-17 TX 高电平调制输出波形图

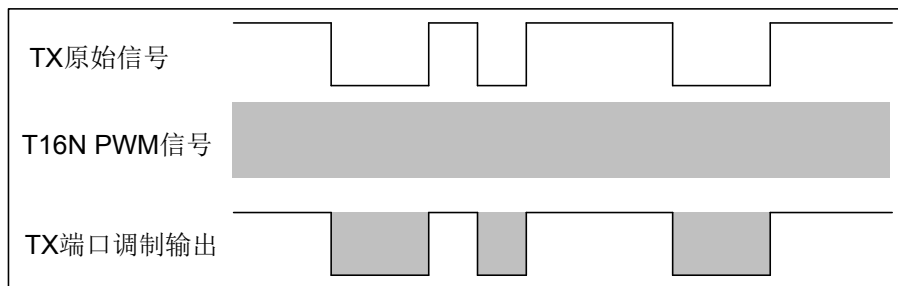


图 5-18 TX 低电平调制输出波形图

### 5.2.7 UART红外唤醒功能

芯片支持 UART 接收端口的红外唤醒功能，需要通过软件控制实现。使能 UART 接收复用端口 RX0 的外部中断 PINT 功能，通过端口的 PINT 中断实现芯片的红外唤醒功能。对中断和唤醒的具体控制操作，可参考外部中断和睡眠模式，唤醒模式的相关章节描述。

### 5.2.8 UART端口极性

配置 UART\_CON 寄存器的 TXP 和 RXP，可分别选择发送端口 TX0 和接收端口 RX0 的正负极性。选择为正极性时，UART 端口数据与传输的数据一致；选择为负极性时，UART 端口数据与传输的数据相反。

### 5.2.9 特殊功能寄存器

#### UART 控制寄存器 (UART\_CON)

偏移地址: 00H

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		BCS<1:0>		保留											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXP	RXMOD<2:0>		保留		RRST	RXEN	TXP	TXMOD<2:0>		保留		TXFS	TRST	TXEN	

—	bit31-30	—	—
BCS<1:0>	bit29-28	R/W	传输波特率发生器时钟选择位 00: 禁止，波特率发生器停止工作 01: PCLK 10: PCLK/2 11: PCLK/4
—	bit27-16	—	—
RXP	bit15	R/W	接收端口极性选择位 0: 正极性 (标准 UART 极性) 1: 负极性 (反向 UART 极性)
RXMOD<2:0>	bit14-12	R/W	接收数据格式选择位 000: 7 位数据 001: 8 位数据

			01x: 9 位数据 100: 7 位数据+奇校验位 101: 7 位数据+偶校验位 110: 8 位数据+奇校验位 111: 8 位数据+偶校验位
—	bit11-10	—	—
RRST	bit9	W	接收器软件复位 0: 读取时始终为 0 1: 软件复位
RXEN	bit8	R/W	接收使能位 0: 禁止 1: 使能
TXP	bit7	R/W	发送端口极性选择位 0: 正极性 (标准 UART 极性) 1: 负极性 (反向 UART 极性)
TXMOD<2:0>	bit6-4	R/W	发送数据格式选择位 000: 7 位数据 001: 8 位数据 01x: 9 位数据 100: 7 位数据+奇校验位 101: 7 位数据+偶校验位 110: 8 位数据+奇校验位 111: 8 位数据+偶校验位
—	bit3	—	—
TXFS	bit2	R/W	发送帧停止位选择 0: 1 位停止位 1: 2 位停止位
TRST	bit1	W	发送器软件复位 0: 读取时始终为 0 1: 软件复位
TXEN	bit0	R/W	发送使能位 0: 禁止 1: 使能

**UART 波特率寄存器 (UART\_BRR)**

偏移地址: 04<sub>H</sub>

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

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

—	bit31-11	—	—
BRR<10:0>	bit10-0	R/W	传输波特率设置位

注：UART 传输波特率计算公式如下：

BCS<1:0>=01 时：FPCLK/((BRR+1)\*16)；

BCS<1:0>=10 时：FPCLK/((BRR+1)\*32)；

BCS<1:0>=11 时：FPCLK/((BRR+1)\*64)。

**UART 发送数据写入寄存器 (UART\_TBW)**

偏移地址：08<sub>H</sub>

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

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

TBW<31:0>	bit31-0	W	<p><b>写入的发送数据</b></p> <p>字节写入时：仅允许对 TBW&lt;7:0&gt;写入</p> <p>半字写入时：仅允许对 TBW&lt;15:0&gt;写入</p> <p>字写入时：对 TBW&lt;31:0&gt;写入</p>
-----------	---------	---	--

**UART 接收数据读取寄存器 (UART\_RBR)**

偏移地址：0C<sub>H</sub>

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

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

RBR<31:0>	bit31-0	R	<p><b>读取的接收数据</b></p> <p>字节读取时：仅允许对 RBR&lt;7:0&gt;读取</p> <p>半字读取时：仅允许对 RBR&lt;15:0&gt;读取</p> <p>字读取时：对 RBR&lt;31:0&gt;读取</p>
-----------	---------	---	--

**UART 发送缓冲寄存器 0/1 (UART\_TB01)**

偏移地址：10<sub>H</sub>

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

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

—	bit31-30	—	—
TBEF1	bit29	R	发送缓冲器 1 空标志位 0: 满 1: 空
TP1	bit28	R	发送的奇偶校验位 发送缓冲器 1 对应的奇偶校验位
—	bit27-25	—	—
TB1<8:0>	bit24-16	R	发送缓冲器 1
—	bit15-14	—	—
TBEF0	bit13	R	发送缓冲器 0 空标志位 0: 满 1: 空
TP0	bit12	R	发送的奇偶校验位 发送缓冲器 0 对应的奇偶校验位
—	bit11-9	—	—
TB0<8:0>	bit8-0	R	发送缓冲器 0

UART 发送缓冲寄存器 2/3 (UART\_TB23)

偏移地址: 14<sub>H</sub>

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

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

—	bit31-30	—	—
TBEF3	bit29	R	发送缓冲器 3 空标志位 0: 满 1: 空
TP3	bit28	R	发送的奇偶校验位 发送缓冲器 3 对应的奇偶校验位
—	bit27-25	—	—
TB3	bit24-16	R	发送缓冲器 3
—	bit15-14	—	—
TBEF2	bit13	R	发送缓冲器 2 空标志位 0: 满 1: 空
TP2	bit12	R	发送的奇偶校验位 发送缓冲器 2 对应的奇偶校验位
—	bit11-9	—	—
TB2	bit8-0	R	发送缓冲器 2



**UART 接收缓冲寄存器 0/1 (UART\_RB0/1)**

偏移地址: 18<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PE1	FE1	RBFF1	RP1	保留			RB1<8:0>								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PE0	FE0	RBFF0	RP0	保留			RB0<8:0>								

PE1	bit31	R	接收缓冲器 1 数据校验错误标志位 0: 正确 1: 错误
FE1	bit30	R	接收缓冲器 1 数据帧错误标志位 0: 正确 1: 错误
RBFF1	bit29	R	接收缓冲器 1 满标志位 0: 空 1: 满
RP1	bit28	R	接收的奇偶校验位 接收缓冲器 1 对应的奇偶校验位
—	bit27-25	—	—
RB1<8:0>	bit24-16	R	接收缓冲器 1
PE0	bit15	R	接收缓冲器 0 数据校验错误标志位 0: 正确 1: 错误
FE0	bit14	R	接收缓冲器 0 数据帧错误标志位 0: 正确 1: 错误
RBFF0	bit13	R	接收缓冲器 0 满标志位 0: 空 1: 满
RP0	bit12	R	接收的奇偶校验位 接收缓冲器 0 对应的奇偶校验位
—	bit11-9	—	—
RB0<8:0>	bit8-0	R	接收缓冲器 0

**UART 接收缓冲寄存器 2/3 (UART\_RB23)**

偏移地址: 1C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PE3	FE3	RBFF3	RP3	保留			RB3<8:0>								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PE2	FE2	RBFF2	RP2	保留			RB2<8:0>								

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

**UART 中断使能寄存器 (UART\_IE)**

偏移地址: 20<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RXIDLEIE	TXIDLEIE	RBIM<1:0>	TBIM<1:0>	保留	TBWEIE	PEIE	FEIE	ROIE	RBIE	TBIE				

—	bit31-14	—	—
RXIDLEIE	bit13	R/W	接收空闲标志中断使能位 0: 禁止 1: 使能
TXIDLEIE	bit12	R/W	发送空闲标志中断使能位 0: 禁止 1: 使能
RBIM<1:0>	bit11-10	R/W	接收缓冲器满中断模式选择位 00: 字节满产生中断 01: 半字满产生中断 1x: 字满产生中断
TBIM<1:0>	bit9-8	R/W	发送缓冲器空中断模式选择位 00: 字节空产生中断 01: 半字空产生中断 1x: 字空产生中断
—	bit7-6	—	—
TBWEIE	bit5	R/W	发送数据错误中断使能位 0: 禁止 1: 使能
PEIE	bit4	R/W	接收校验错误中断使能位 0: 禁止 1: 使能
FEIE	bit3	R/W	接收帧错误中断使能位 0: 禁止 1: 使能
ROIE	bit2	R/W	接收数据溢出中断使能位 0: 禁止 1: 使能
RBIE	bit1	R/W	接收缓冲器满中断使能位 0: 禁止 1: 使能
TBIE	bit0	R/W	发送缓冲器空中断使能位 0: 禁止 1: 使能

**UART 中断标志寄存器 (UART\_IF)**

偏移地址: 24<sub>H</sub>

复位值: 00000000\_00000000\_11110000\_00000001<sub>B</sub>

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

—	bit31-14	—	—
RXIDLEIF	bit13	R/W	接收空闲中断标志位 0: 忙碌 1: 接收空闲中断标志, 初始复位为“1” 软件写 1 清除标志位, 写 0 无效
TXIDLEIF	bit12	R/W	发送空闲中断标志位 0: 忙碌 1: 发送空闲中断标志, 初始复位为“1” 软件写 1 清除标志位, 写 0 无效
—	bit11-6	—	—
TBWEIF	bit5	R/W	发送数据错误中断标志位 0: 发送数据正常 1: 发送数据错误 软件写 1 清除标志位, 写 0 无效
PEIF	bit4	R/W	接收校验错误中断标志位 0: 接收校验正常 1: 接收校验错误 软件写 1 清除标志位, 写 0 无效
FEIF	bit3	R/W	接收帧错误中断标志位 0: 接收帧正常 1: 接收帧错误 软件写 1 清除标志位, 写 0 无效
ROIF	bit2	R/W	接收数据溢出中断标志位 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
RBIF	bit1	R	接收缓冲器满中断标志位 0: 非满 1: 满
TBIF	bit0	R	发送缓冲器空中断标志位 0: 非空 1: 空

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

**UART 脉宽调制寄存器 (GPIO\_TXPC)**

偏移地址：20<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TX3PLV	TX3PS<1:0>	保留	TX2PLV<1:0>	TX2PS	保留	TX1PLV	TX1PS	保留	TX0PLV	TX0PS<1:0>				

—	bit31-15	—	—
TX3PLV	bit14	R/W	<b>TX3 调制电平选择位</b> 0: 低电平 1: 高电平
TX3PS<1:0>	bit13-12	R/W	<b>TX3 调制输出选择位</b> 00: 非调制输出 01: 固定 38KHz 调制输出 10: T16N3OUT0 PWM 调制输出 11: T16N3OUT1 PWM 调制输出
—	bit11	R/W	—
TX2PLV	bit10	R/W	<b>TX2 调制电平选择位</b> 0: 低电平 1: 高电平
TX2PS<1:0>	bit9-8	R/W	<b>TX2 调制输出选择位</b> 00: 非调制输出 01: 固定 38K 调制输出 10: T16N2OUT0 PWM 调制输出 11: T16N2OUT1 PWM 调制输出
—	bit7	R/W	—
TX1PLV	bit6	R/W	<b>TX1 调制电平选择位</b> 0: 低电平 1: 高电平
TX1PS<1:0>	bit5-4	R/W	<b>TX1 调制输出选择位</b> 00: 非调制输出 01: 固定 38K 调制输出 10: T16N1OUT0 PWM 调制输出 11: T16N1OUT1 PWM 调制输出
—	bit3	R/W	—
TX0PLV	bit2	R/W	<b>TX0 调制电平选择位</b> 0: 低电平

			1: 高电平
TX0PS<1:0>	bit1-0	R/W	<b>TX0 调制输出选择位</b> 00: 非调制输出 01: 固定 38K 调制输出 10: T16N0OUT0 PWM 调制输出 11: T16N0OUT1 PWM 调制输出

注：GPIO\_TXPC 寄存器的地址分配，在外设寄存器地址映射表中，位于 GPIO 区域，但由于该寄存器主要用于配置 UART 的发送脉宽调制功能，所以将其放在本章节中描述。

### 5.2.10 UART应用说明

芯片支持 4 个通用异步接收/发送器，分别为 UART0，UART1，UART2，UART3。

UART 的发送脉宽调制模式，使用 T16N 产生的 PWM 脉冲进行调制，然后通过 TX0/TX1/TX2/TX3 端口输出调制后的信号，与 PWM 脉冲本身是否通过 T16N 的输出端口 0 和输出端口 1 的端口输出无关，所以对 T16N0，T16N1，T16N2，T16N3 的输出调制模式，即使芯片不支持其 I/O 端口复用功能，仍可以在正确配置后，产生 PWM 脉冲分别对 TX0，TX1，TX2，TX3 输出端口的信号进行调制。

### 5.3 增强型通用异步接收/发送器 (EUART0/EUART1)

以 EUART0 为例, EUART1 同 UART0。

#### 5.3.1 概述

- ◇ 配置为普通 UART 模式, 与 UART 功能完全兼容
- ◇ 配置为 7816 模式, 支持 7816 通讯协议
- ◇ 支持异步接收器/发送器
- ◇ 支持半双工通讯模式
- ◇ 支持 8 位数据位和 1 位奇偶校验位数据传输格式
- ◇ 支持自动重发重收功能
- ◇ 支持可配置内部时钟输出
- ◇ 支持双通道通讯可配置

#### 5.3.2 结构框图

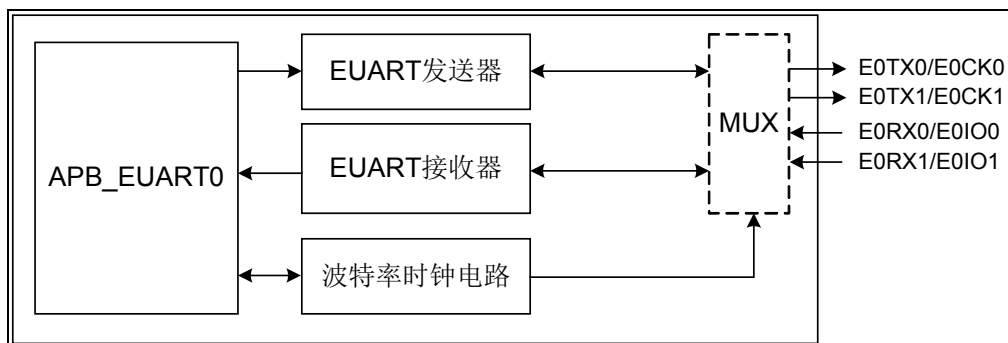


图 5-19 EUART0 电路结构框图

#### 5.3.3 EUART 端口复用

EUART 复用端口	普通 UART 模式	7816 模式
E0TX0/E0CK0	E0TX0	E0CK0
E0TX1/E0CK1	E0TX1	E0CK1
E0RX0/E0IO0	E0RX0	E0IO0
E0RX1/E0IO1	E0RX1	E0IO1

配置相应 I/O 端口的复用功能为 EUART 通讯端口复用功能。

EUART\_MOD 寄存器中,

- 1) 配置 MOD=0, 选择普通 UART 模式, 通讯端口的复用功能为 E0TX0/E0TX1 和 E0RX0/E0RX1。
- 2) 配置 MOD=1, 选择 7816 模式, 通讯端口的复用功能为 E0CK0/E0CK1 和 E0IO0/E0IO1。

EUART 模块支持两个独立的内部时钟输出端口 (E0CK0 和 E0CK1)。配置 EUART\_MOD

寄存器的 CK0E/CK1E 位，可启用 E0CK0/E0CK1 时钟端口，输出芯片内部时钟，配置该寄存器的 CKS<1:0>位，可选择输出的时钟源。

EUART 模块支持两个数据端口 (E0IO0 和 E0IO1)，这两个数据端口可分时复用，构成通讯双通道。配置 EUART\_MOD 寄存器的 CHS 位，可选择通讯时使用的 E0IO0/E0IO1 数据端口。

### 5.3.4 普通UART通讯模式

EUART 模块配置为普通 UART 通讯模式后，除不支持发送脉宽调制外，其它功能与 UART 模块一致，可参考与前面章节的相关内容，此处不再描述。

### 5.3.5 7816 通讯模式的异步接收器和发送器

7816 模式为半双工通讯方式，该模式下 EUART\_CON 寄存器的 RXEN 与 TXEN 使能位无效，由 EUART\_MOD 寄存器的 IOC 方向控制位交替使能接收器和发送器，并发送或接收 ACK 应答信号，同时控制 E0IO0/E0IO1 端口的输入输出状态。

配置 EUART\_MOD 寄存器的 IOC=1，EUART 发送数据，接收 ACK 应答信号。发送数据时，依次发送起始位 (1bit)，数据位 (8bit，先发低位，后发高位)，校验位 (1bit)；然后接收 ACK 应答信号。当对连续两帧数据进行背靠背发送时，两帧数据间的间隔必须大于一定的时间，该时间称为保护时间，由通讯双方事先约定。配置该寄存器的 ETUS<7:0>，可选择保护时间。每帧数据发送完成后，EUART 均会接收应答信号。读取 RNACK，可判断是否接收到 ACK 应答信号，其中 1 为 ACK，0 为 NACK。如果在保护时间内未接收到应答信号，还会置起 NACK 中断标志 RNAIF。

配置 EUART\_MOD 寄存器的 IOC=0，EUART 接收数据，发送 ACK 应答信号。接收数据时，先等待起始位，再依次接收数据位 (8bit，先收低位，后收高位)，校验位 (1bit)；然后判断校验位是否正确，正确则发送 ACK 应答，否则发送 NACK，其中 ACK=1，NACK=0。配置该寄存器的 TNAS<1:0>，可选择 NACK 电平的宽度。

配置 EUART\_MOD 寄存器的 PS，可选择数据的奇偶校验方式。发送时，硬件电路自动计算发送数据的奇偶校验位并发送。接收时，硬件电路自动对接收的数据进行奇偶校验，如果校验错误，会置起 EUART\_IF 寄存器的奇偶校验错误中断标志 PEIF 位。

EUART 支持 4 级发送缓冲器 TB0，TB1，TB2，TB3 和 1 级发送移位寄存器。发送缓冲器 TB0/TB1/TB2/TB3 为只读寄存器，只能通过发送数据寄存器 EUART\_TBW 写入。

对异步发送器，数据从写入到发送到端口的数据流示意图如下所示：

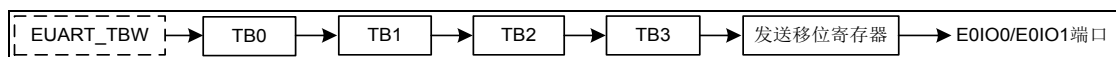


图 5-20 7816 通讯模式发送数据流示意图

EUART 支持 4 级接收缓冲器 RB0，RB1，RB2，RB3 和 1 级接收移位寄存器。可通过读取接收数据寄存器 EUART\_RBR，得到接收的数据。

对异步接收器，数据从接收端口到各级缓冲器的数据流示意图如下所示：

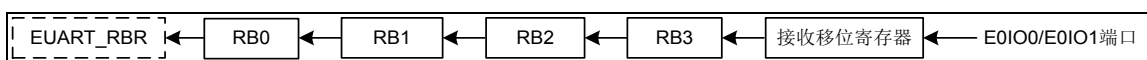


图 5-21 7816 通讯模式接收数据流示意图



配置 ERST, 可将 7816 通讯模块软件复位, 复位后: 禁止数据传输, EUART\_CON 寄存器中 TXEN=0, RXEN=0; 禁止 EUART\_IE 寄存器中各中断 TBIE=0, TBWEIE=0, RBIE=0, ROIE=0, FEIE=0, PEIE=0, ERNAIE=0; 复位相关中断标志为默认值, EUART\_IF

寄存器中 TBIF=1, TBWEIF=0, RBIF=0, ROIF=0, FEIF=0, PEIF=0, RNAIF=0, 置起空闲标志 TXIDLEIF=1, RXIDLEIF=1; 置起 EUART\_TB01 和 EUART\_TB23 寄存器中的各发送缓冲器空标志 TBEF0/TBEF1/TBEF2/TBEF3=1; 清除 EUART\_RB01 和 EUART\_RB23 寄存器中的各接收缓冲器满标志 RBFF0/RBFF1/RBFF2/RBFF3=0, 各接收缓冲器错误标志 FE0/FE1/FE2/FE3=0, PE0/PE1/PE2/PE3=0。

### 5.3.6 7816 通讯模式的数据格式

7816 模式通讯数据帧为 8 位数据位和 1 位奇偶校验位, 支持两种数据传输格式, 正向约定数据格式和反向约定数据格式。

配置 EUART\_MOD 寄存器的 DAS=0, 选择正向约定数据格式。实际发送的数据与写入到寄存器 EUART\_TBW 的数据一致。

举例说明正向约定数据格式: 以字节方式向 EUART\_TBW 写入数据 0x50, 传输到发送移位寄存器的值为 0x50, 并对 0x50 计算奇偶校验位, E0IO0/E0IO1 端口发送的数据序列依次为“0(起始位)+00001010+校验位”。同理, 如果 E0IO0/E0IO1 端口收到的数据是 0x50, 则读取寄存器 EUART\_RBR, 得到的数据为 0x50。

配置 DAS=1, 选择反向约定数据格式。实际发送的数据, 是将写入到寄存器 EUART\_TBW 的数据, 先高低位互换再取反后的数据。

举例说明反向约定数据格式: 以字节方式向 EUART\_TBW 写入数据 0x50, 二进制格式为 01010000, 高低位互换后为 00001010, 再取反后为 11110101, 所以传输到发送移位寄存器的值为 0xF5, 并对 0xF5 计算奇偶校验位, E0IO0/E0IO1 端口发送的数据序列依次为“0(起始位)+10101111+校验位”。同理, 如果 E0IO0/E0IO1 端口收到的数据是 0xF5 时, 则读取寄存器 EUART\_RBR, 得到的数据为 0x50。

### 5.3.7 7816 通讯模式的自动重发功能

7816 模式支持数据自动重发功能。

配置 EUART\_MOD 寄存器的 ARTE=0, 禁止自动重发, 当收到 NACK (即应答信号电平为 0) 时, 置起 EUART\_IF 寄存器的 NACK 中断标志 RNAIF, 且 RNACK=1, 然后继续发送下一个数据帧, 并在发送完成后, 根据 ACK 的状态更新 RNACK 位, 但中断标志 RNAIF 始终保持为 1, 直到 EUART 软件复位或通过软件将该标志清 0。

举例说明禁止自动重发时的数据传输过程: 发送移位寄存器中的数据为 0x55, 发送缓冲器 TB3 中的数据为 0xAA。如果数据 0x55 发送成功, 收到 ACK 应答, 则 NACK 中断标志 RNAIF=0, 且 RNACK=0, 然后继续发送数据 0xAA, 并根据发送结果更新 RNAIF 和 RNACK; 如果数据 0x55 发送失败, 未收到 ACK 应答, 则 NACK 中断标志 RNAIF=1, 且 RNACK=1, 然后继续发送数据 0xAA, 并根据发送结果更新 RNACK, 但无论数据 0xAA 发送成功与否, 中断标志 RNAIF 均保持为 1。

配置 EUART\_MOD 寄存器的 ARTE=1, 使能自动重发, 当收到 NACK (即应答信号电平为 0) 时, 自动重发前一帧数据, 每次重发均会影响 RNACK, 但不会影响 NACK 中断标

志 RNAIF，配置该寄存器的 ARTS<1:0>，可选择重发的次数。在数据重发过程中，若某次重发后，收到 ACK（即应答信号电平为 1），则停止重发。若重发次数达到设置的上限后，数据发送依然失败，则停止重发，同时后续未发送的数据也被终止，置起 EUART\_IF 寄存器的自动重发失败中断标志 ARTEIF 和发送空闲标志 TXIDLEIF。

举例说明使能自动重发时的数据传输过程：发送移位寄存器中的数据为 0x55，发送缓冲器 TB3 中的数据为 0xAA，重发次数为 2 次。如果数据 0x55 发送失败，则发送移位寄存器中的数据依然是 0x55，发送缓冲器 TB3 中的数据依然是 0xAA，NACK 中断标志 RNAIF=1，且 RNACK=1，ARTEIF=0，TXIDLEIF=0。然后重发第 1 次，如果收到 ACK 应答，则 RNACK=0，RNAIF 仍保持为 1，ARTEIF=0，TXIDLEIF=0，并继续发送数据 0xAA；如果收到 NACK，则 RNACK=1，RNAIF 保持为 1，ARTEIF=0，TXIDLEIF=0，并重发第 2 次。如果第 2 次重发依然失败，则停止重发，RNACK=1，RNAIF 保持为 1，ARTEIF=1，TXIDLEIF=1，并不再继续发送数据 0xAA。

### 5.3.8 7816 通讯模式的自动重收功能

7816 模式支持数据自动重收功能。

EUART\_MOD 寄存器中配置 ARRE=0，禁止自动重收，无论接收数据的奇偶校验值是否与校验位相符，均会在保护时间内发送 ACK（即应答信号电平为 1）。

配置 ARRE=1，使能自动重收，当接收数据的奇偶校验值与校验位不符时，会在保护时间内发送 NACK（即应答信号电平为 0）。如果发送方支持重发功能，在收到 NACK 后会重发前一帧数据。

### 5.3.9 特殊功能寄存器

#### EUART 控制寄存器 (EUART\_CON)

偏移地址: 00<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		BCS<1:0>		保留											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXP	RXMOD<2:0>		保留		RRST	RXEN	TXP	TXMOD<2:0>		保留		TXFS	TRST	TXEN	

—	bit31-30	—	—
BCS<1:0>	bit29-28	R/W	传输波特率发生器时钟选择位 00: 禁止，波特率发生器停止工作 01: PCLK 10: PCLK/2 11: PCLK/4
—	bit27-16	—	—
RXP	bit15	R/W	接收端口极性选择位 0: 正极性（标准 UART 极性） 1: 负极性（反向 UART 极性）
RXMOD<2:0>	bit14-12	R/W	接收数据格式选择位 000: 7 位数据

			001: 8 位数据 01x: 9 位数据 100: 7 位数据+奇校验位 101: 7 位数据+偶校验位 110: 8 位数据+奇校验位 111: 8 位数据+偶校验位
—	bit11-10	—	—
RRST	bit9	W	接收器软件复位 0: 读取时始终为 0 1: 软件复位
RXEN	bit8	R/W	接收使能位 0: 禁止 1: 使能
TXP	bit7	R/W	发送端口极性选择位 0: 正极性 (标准 UART 极性) 1: 负极性 (反向 UART 极性)
TXMOD<2:0>	bit6-4	R/W	发送数据格式选择位 000: 7 位数据 001: 8 位数据 01x: 9 位数据 100: 7 位数据+奇校验位 101: 7 位数据+偶校验位 110: 8 位数据+奇校验位 111: 8 位数据+偶校验位
—	bit3	—	—
TXFS	bit2	R/W	发送帧停止位选择 0: 1 位停止位 1: 2 位停止位
TRST	bit1	W	发送器软件复位 0: 读取时始终为 0 1: 软件复位
TXEN	bit0	R/W	发送使能位 0: 禁止 1: 使能

**EUART 波特率寄存器 (EUART\_BRR)**

偏移地址: 04<sub>H</sub>

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

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

—	bit31-11	—	—
BRR<10:0>	bit10-0	R/W	传输波特率设置位，7816 通讯模式同样需要设置波特率

注：EUART 传输波特率计算公式如下：

BCS<1:0>=01 时：FPCLK/((BRR+1)\*16)；

BCS<1:0>=10 时：FPCLK/((BRR+1)\*32)；

BCS<1:0>=11 时：FPCLK/((BRR+1)\*64)。

**EUART 发送数据写入寄存器 (EUART\_TBW)**

偏移地址：08<sub>H</sub>

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

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

TBW<31:0>	bit31-0	W	<p><b>写入的发送数据</b></p> <p>字节写入时：仅允许对 TBW&lt;7:0&gt;写入</p> <p>半字写入时：仅允许对 TBW&lt;15:0&gt;写入</p> <p>字写入时：对 TBW&lt;31:0&gt;写入</p>
-----------	---------	---	--

**EUART 接收数据读取寄存器 (EUART\_RBR)**

偏移地址：0C<sub>H</sub>

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

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

RBR<31:0>	bit31-0	R	<p><b>读取的接收数据</b></p> <p>字节读取时：仅允许对 RBR&lt;7:0&gt;读取</p> <p>半字读取时：仅允许对 RBR&lt;15:0&gt;读取</p> <p>字读取时：对 RBR&lt;31:0&gt;读取</p>
-----------	---------	---	--

**EUART 发送缓冲寄存器 0/1 (EUART\_TB01)**

偏移地址：10<sub>H</sub>

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

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

—	bit31-30	—	—
TBEF1	bit29	R	发送缓冲器 1 空标志位 0: 满 1: 空
TP1	bit28	R	发送的奇偶校验位 发送缓冲器 1 对应的奇偶校验位
—	bit27-25	—	—
TB1<8:0>	bit24-16	R	发送缓冲器 1
—	bit15-14	—	—
TBEF0	bit13	R	发送缓冲器 0 空标志位 0: 满 1: 空
TP0	bit12	R	发送的奇偶校验位 发送缓冲器 0 对应的奇偶校验位
—	bit11-9	—	—
TB0<8:0>	bit8-0	R	发送缓冲器 0

**EUART 发送缓冲寄存器 2/3 (EUART\_TB23)**

偏移地址: 14<sub>H</sub>

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

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

—	bit31-30	—	—
TBEF3	bit29	R	发送缓冲器 3 空标志位 0: 满 1: 空
TP3	bit28	R	发送的奇偶校验位 发送缓冲器 3 对应的奇偶校验位
—	bit27-25	—	—
TB3<8:0>	bit24-16	R	发送缓冲器 3
—	bit15-14	—	—
TBEF2	bit13	R	发送缓冲器 2 空标志位 0: 满 1: 空
TP2	bit12	R	发送的奇偶校验位 发送缓冲器 2 对应的奇偶校验位
—	bit11-9	—	—
TB2<8:0>	bit8-0	R	发送缓冲器 2

**EUART 接收缓冲寄存器 0/1 (EUART\_RB01)**

偏移地址: 18<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PE1	FE1	RBFF1	RP1	保留				RB1<8:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PE0	FE0	RBFF0	RP0	保留				RB0<8:0>							

PE1	bit31	R	接收缓冲器 1 数据校验错误标志位 0: 正确 1: 错误
FE1	bit30	R	接收缓冲器 1 数据帧错误标志位 0: 正确 1: 错误
RBFF1	bit29	R	接收缓冲器 1 满标志位 0: 空 1: 满
RP1	bit28	R	接收的奇偶校验位 接收缓冲器 1 对应的奇偶校验位
—	bit27-25	—	—
RB1<8:0>	bit24-16	R	接收缓冲器 1
PE0	bit15	R	接收缓冲器 0 数据校验错误标志位 0: 正确 1: 错误
FE0	bit14	R	接收缓冲器 0 数据帧错误标志位 0: 正确 1: 错误
RBFF0	bit13	R	接收缓冲器 0 满标志位 0: 空 1: 满
RP0	bit12	R	接收的奇偶校验位 接收缓冲器 0 对应的奇偶校验位
—	bit11-9	—	—
RB0<8:0>	bit8-0	R	接收缓冲器 0

**EUART 接收缓冲寄存器 2/3 (EUART\_RB23)**

偏移地址: 1C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PE3	FE3	RBFF3	RP3	保留			RB3<8:0>								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PE2	FE2	RBFF2	RP2	保留			RB2<8:0>								

PE3	bit31	R	接收缓冲器 3 数据校验错误标志位 0: 正确 1: 错误
FE3	bit30	R	接收缓冲器 3 数据帧错误标志位 0: 正确 1: 错误
RBEF3	bit29	R	接收缓冲器 3 满标志位 0: 空 1: 满
RP3	bit28	R	接收的奇偶校验位 接收缓冲器 3 对应的奇偶校验位
—	bit27-25	—	—
RB3<8:0>	bit24-16	R	接收缓冲器 3
PE2	bit15	R	接收缓冲器 2 数据校验错误标志位 0: 正确 1: 错误
FE2	bit14	R	接收缓冲器 2 数据帧错误标志位 0: 正确 1: 错误
RBEF2	bit13	R	接收缓冲器 2 满标志位 0: 空 1: 满
RP2	bit12	R	接收的奇偶校验位 接收缓冲器 2 对应的奇偶校验位
—	bit11-9	—	—
RB2<8:0>	bit8-0	R	接收缓冲器 2

**EUART 中断使能寄存器 (EUART\_IE)**

偏移地址: 20<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RBIM<1:0>	TBIM<1:0>	RXIDLEIE	TXIDLEIE	RNAIE	ARTEIE	TBWEIE	PEIE	FEIE	ROIE	RBIE	TBIE			

—	bit31-16	—	—
RBIM<1:0>	bit15-14	R/W	接收缓冲器满中断模式选择位 00: 字节满产生中断 01: 半字满产生中断 1x: 字满产生中断
TBIM<1:0>	bit13-12	R/W	发送缓冲器空中断模式选择位 00: 字节空产生中断 01: 半字空产生中断 1x: 字空产生中断
—	bit11-10	—	—
RXIDLEIE	bit9	R/W	接收空闲标志中断使能位 0: 禁止 1: 使能
TXIDLEIE	bit8	R/W	发送空闲标志中断使能位 0: 禁止 1: 使能
RNAIE	bit7	R/W	接收 <b>NACK</b> 中断使能位 0: 禁止 1: 使能
ARTEIE	bit6	R/W	自动重发失败中断使能位 0: 禁止 1: 使能
TBWEIE	bit5	R/W	发送数据错误中断使能位 0: 禁止 1: 使能
PEIE	bit4	R/W	接收校验错误中断使能位 0: 禁止 1: 使能
FEIE	bit3	R/W	接收帧错误中断使能位 0: 禁止 1: 使能
ROIE	bit2	R/W	接收数据溢出中断使能位 0: 禁止 1: 使能
RBIE	bit1	R/W	接收缓冲器满中断使能位 0: 禁止 1: 使能
TBIE	bit0	R/W	发送缓冲器空中断使能位 0: 禁止



			1: 使能
--	--	--	-------

注: 普通 UART 模式时, EUART\_CON 寄存器中的 TRST 清除 TBIE 和 TBWEIE, RRST 清除 RBIE, ROIE, FEIE 和 PEIE。

**EUART 中断标志寄存器 (EUART\_IF)**

偏移地址: 24H

复位值: 00000000\_00000000\_00110011\_00000001<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RNACK	RXIDLEIF	TXIDLEIF	RNAIF	ARTEIF	TBWEIF	PEIF	FEIF	ROIF	RBIF	TBIF	

—	bit31-11	—	—
RNACK	bit10	R	<p><b>接收到 NACK 的标志位</b></p> <p>0: 接收到 ACK 1: 接收到 NACK</p>
RXIDLEIF	bit9	R/W	<p><b>接收空闲标志位</b></p> <p>0: 忙碌 1: 接收空闲中断标志, 初始复位为“1” 软件写 1 清除标志位, 写 0 无效</p>
TXIDLEIF	bit8	R/W	<p><b>发送空闲标志位</b></p> <p>0: 忙碌 1: 发送空闲中断标志, 初始复位为“1” 软件写 1 清除标志位, 写 0 无效</p>
RNAIF	bit7	R/W	<p><b>接收 NACK 中断标志位</b></p> <p>0: 未接收到 NACK 1: 接收到 NACK (自动重发时不会更新该标志) 软件写 1 清除标志位, 写 0 无效</p>
ARTEIF	bit6	R/W	<p><b>自动重发失败中断标志位</b></p> <p>0: 自动重发状态正常 1: 自动重发失败 软件写 1 清除标志位, 写 0 无效</p>
TBWEIF	bit5	R/W	<p><b>发送数据错误中断标志位</b></p> <p>0: 发送数据正常 1: 发送数据错误 软件写 1 清除标志位, 写 0 无效</p>
PEIF	bit4	R/W	<p><b>接收校验错误中断标志位</b></p> <p>0: 接收校验正常 1: 接收校验错误 软件写 1 清除标志位, 写 0 无效</p>

FEIF	bit3	R/W	接收帧错误中断标志位 0: 接收帧正常 1: 接收帧错误 软件写 1 清除标志位, 写 0 无效
ROIF	bit2	R/W	接收数据溢出中断标志位 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
RBIF	bit1	R	接收缓冲器满中断标志位 0: 未滿 1: 滿
TBIF	bit0	R	发送缓冲器空中断标志位 0: 非空 1: 空

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

**EUART 7816 模式控制寄存器 (EUART\_MOD)**

偏移地址: 30<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								ETUS<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS<1:0>	ARTS<1:0>	TNAS<1:0>	ARRE	ARTE	PS	DAS	IOC	CHS	CK1E	CK0E	ERST	MOD			

—	bit31-24	—	—
ETUS<7:0>	bit23-16	R/W	<b>ETU 保护时间选择位</b> 0: 2 个 ETU 时间 1: 3 个 ETU 时间 ... 255: 257 个 ETU 时间
CKS<1:0>	bit15-14	R/W	<b>ECK 时钟源选择位</b> 00: PCLK 01: PCLK/2 10: PCLK/4 11: PCLK/8
ARTS<1:0>	bit13-12	R/W	<b>自动重发次数选择位</b> 00: 重发 1 次 01: 最多重发 2 次 10: 最多重发 3 次 11: 连续重发, 直到数据发送正确为止

TNAS<1:0>	bit11-10	R/W	<b>发送 NACK 信号宽度的选择位</b> 00: 1 个 ETU 01: 1.5 个 ETU 1x: 2 个 ETU
ARRE	bit9	R/W	<b>自动重收使能位</b> 0: 禁止 1: 使能
ARTE	bit8	R/W	<b>自动重发使能位</b> 0: 禁止 1: 使能
PS	bit7	R/W	<b>奇偶校验选择位</b> 0: 奇校验 1: 偶校验
DAS	bit6	R/W	<b>数据格式选择位</b> 0: 正向约定 1: 反向约定
IOC	bit5	R/W	<b>EIO 端口方向控制位</b> 0: 接收数据, 发送应答 1: 发送数据, 接收应答
CHS	bit4	R/W	<b>EIO 通讯通道选择位</b> 0: E0IO0/E1IO0 端口 1: E0IO1/E1IO1 端口
CK1E	bit3	R/W	<b>ECK1 端口使能位</b> 0: 禁止 1: 使能
CK0E	bit2	R/W	<b>ECK0 端口使能位</b> 0: 禁止 1: 使能
ERST	bit1	W	<b>7816 通讯模块软件复位</b> 0: 读取时始终为 0 1: 软件复位
MOD	bit0	R/W	<b>7816 通讯模式选择</b> 0: 普通 UART 模式 1: 7816 模式

### 5.3.10 EUART应用说明

芯片支持 2 个增强型通用异步接收/发送器 EUART0/EUART1。

## 5.4 SPI同步串口通讯控制器 (SPI)

### 5.4.1 概述

- ◇ 支持主控模式、从动模式
- ◇ 支持 4 种数据传输格式
- ◇ 支持主控模式通讯时钟速率可配置
- ◇ 支持 4 级发送缓冲器和 4 级接收缓冲器
- ◇ 支持发送和接收缓冲器空/满中断
- ◇ 支持接收数据溢出中断、发送数据写错误中断、从动模式的发送数据错误中断
- ◇ 支持从动模式的片选变化中断、主控模式的空闲状态中断
- ◇ 支持主控模式延迟接收
- ◇ 支持主控模式发送间隔

### 5.4.2 结构框图

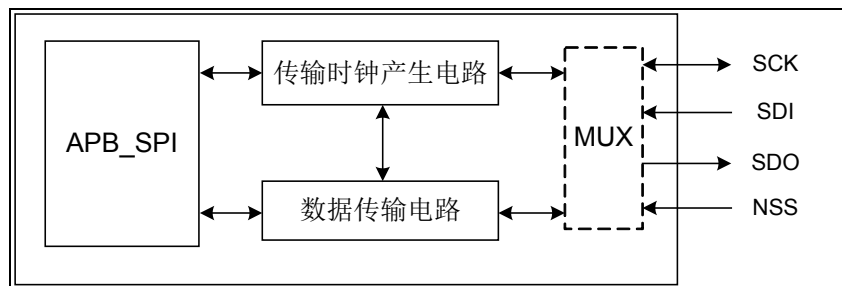


图 5-22 SPI 电路结构框图

### 5.4.3 SPI通讯模式

SPI 支持主控和从动两种通讯模式，配置 SPI\_CON 寄存器的 MS 位，可选择通讯模式。

通讯时钟端口为 SCK，数据输出端口 SDO，数据输入端口 SDI；从动模式下的片选信号端口为 NSS，主控模式下可使用普通 I/O 端口作为片外从设备的片选信号端口。具体见下表所示：

SPI 通讯端口	SPI 主控模式	SPI 从动模式
SCK	支持	支持
SDO	支持	支持
SDI	支持	支持
NSS	—	支持

### 5.4.4 SPI数据格式

配置 SPI\_CON 寄存器中 DFS，可选择 SPI 通讯数据格式，数据发送和接收时，均是高位在前，低位在后。如果是发送数据在先，接收数据在后，输出端口 SDO 会在第一个 SCK 时钟边沿，输出数据的 MSB 位；反之 SDO 在第一个 SCK 时钟边沿之前，输出数据的 MSB 位。

以下以 SPI 从动通讯模式为例，对数据通讯时序进行说明。

DFS<1:0> = 00: 上升沿发送 (先), 下降沿接收 (后)

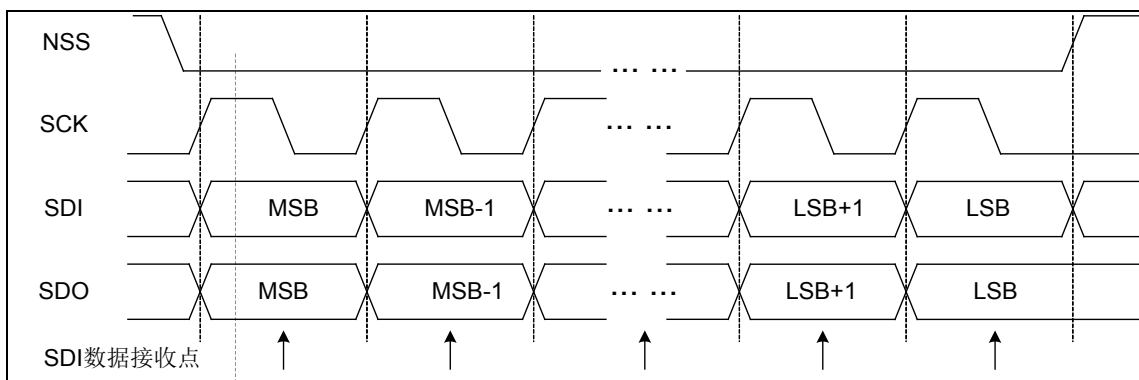


图 5-23 SPI 时钟上升沿发送, 下降沿接收波形示意图

DFS<1:0> = 01: 下降沿发送 (先), 上升沿接收 (后)

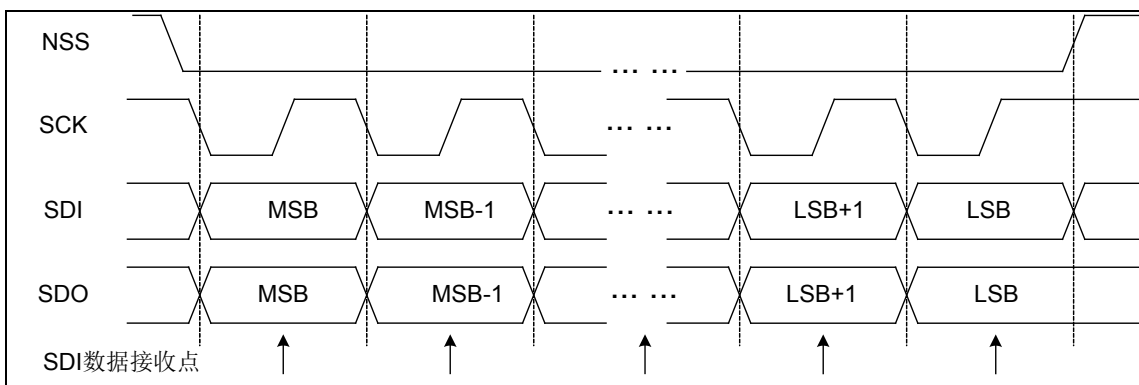


图 5-24 SPI 时钟下降沿发送, 上升沿接收波形示意图

DFS<1:0> = 10: 上升沿接收 (先), 下降沿发送 (后)

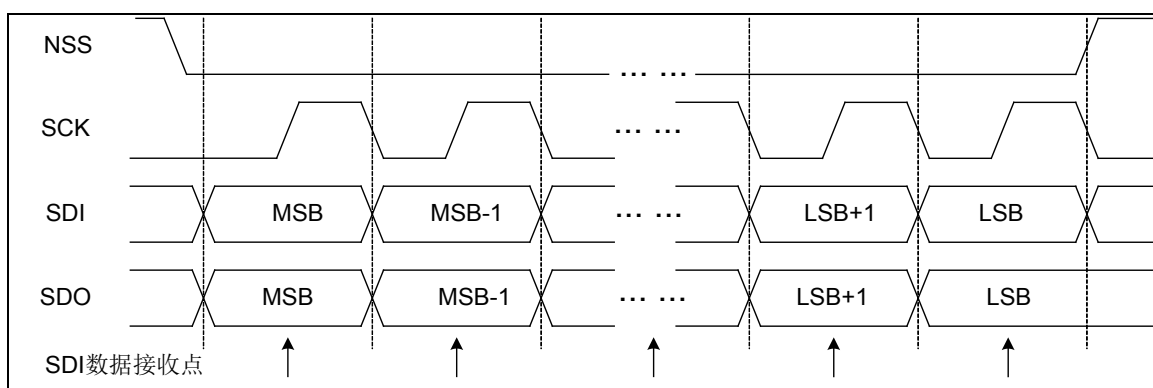


图 5-25 SPI 时钟上升沿接收, 下降沿发送波形示意图

DFS<1:0> = 11: 下降沿接收 (先), 上升沿发送 (后)

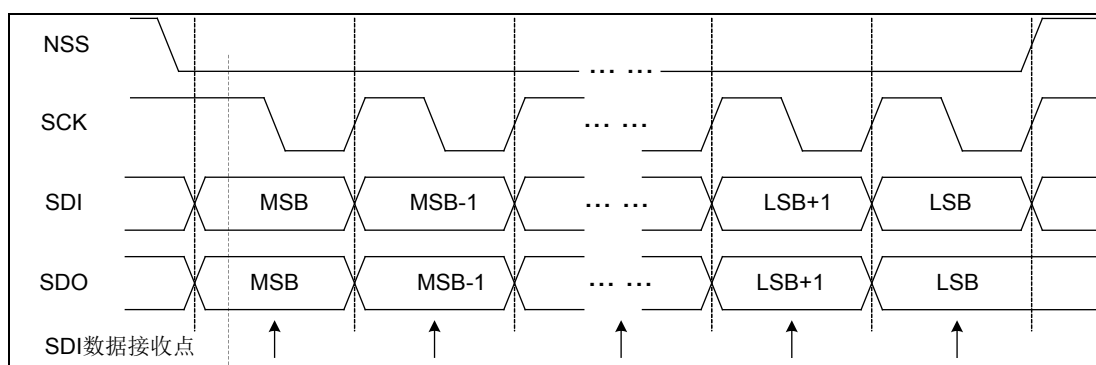


图 5-26 SPI 时钟下降沿接收, 上升沿发送波形示意图

### 5.4.5 SPI同步发送器

支持 4 级发送缓冲器 TB0, TB1, TB2, TB3 和 1 级发送移位寄存器, 可进行数据的连续发送, 直到发送缓冲器和移位寄存器全空, 最多可连续写入和发送 5 帧数据。发送缓冲器 TB0/1/2/3 为只读寄存器, 只能通过发送数据寄存器 SPI\_TBW 写入。

发送数据寄存器 SPI\_TBW 为一个虚拟地址单元, 物理上不存在实际的寄存器电路, 写该寄存器地址单元时, 实际上是将发送数据写入到发送缓冲器 TB0~TB3 中, 再传输到发送移位寄存器, 通过发送数据端口 SDO 进行数据发送。

发送数据寄存器 SPI\_TBW 支持 3 种写入方式: 字节写入, 半字写入和字写入。

字节方式写入 SPI\_TBW 时, 发送数据被写入到发送缓冲器 TB0; 半字方式写入 SPI\_TBW 时, 发送数据被同时写入到发送缓冲器 TB0 和 TB1, 其中低字节存放在 TB1 中; 字方式写入 SPI\_TBW 时, 发送数据被同时写入到发送缓冲器 TB0, TB1, TB2 和 TB3, 其中低字节存放在 TB3 中。

发送数据从写入到发送到端口的数据流示意图如下所示:

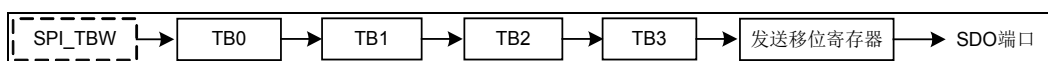


图 5-27 SPI 发送数据流示意图

支持发送缓冲器空中断, 配置 SPI\_IE 寄存器中的 TBIM, 可选择中断模式。

TBIM<1:0>=00, 为字节空产生中断, 即发送缓冲器 TB0 为空时, 会置起 SPI\_IF 寄存器中的中断标志 TBIF;

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

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

支持发送数据寄存器 SPI\_TBW 误写中断, 当对 SPI\_TBW 的写入方式, 与发送缓冲器 TB0~TB3 的空闲状态冲突时, 或写入操作访问错误时, 会置起 SPI\_IF 寄存器中的误写中断标志 TBWEIF。

### 5.4.6 SPI同步接收器

支持 4 级接收缓冲器 RB0, RB1, RB2, RB3 和 1 级接收移位寄存器, 可进行数据的连续接收, 直到接收缓冲器和移位寄存器全满, 最多可连续接收 5 帧数据, 再执行数据读取操作。读取接收数据寄存器 SPI\_RBR, 可得到接收的数据, 对应的接收缓冲器清除满标志 RBFF0/RBFF1/RBFF2/RBFF3; 也可以读取接收缓冲器 RB0~RB3 得到接收的数据, 但不会清除满标志 RBFF0/RBFF1/RBFF2/RBFF3。

接收数据寄存器 SPI\_RBR 为一个虚拟地址单元, 物理上不存在实际的寄存器电路, 读该寄存器地址单元时, 实际上是读取接收缓冲器 RB0~RB3 中的数据。

接收数据寄存器 SPI\_RBR 支持 3 种读取方式: 字节读取, 半字读取和字读取。

字节方式读取 SPI\_RBR 时, 实际是读取接收缓冲器 RB0 的数据; 半字方式读取 SPI\_RBR 时, 实际是同时读取接收缓冲器 RB0 和 RB1 的数据, 其中 RB0 中的数据为低字节; 字方式读取 SPI\_RBR 时, 实际是同时读取接收缓冲器 RB0, RB1, RB2 和 RB3, 其中 RB0 中的数据为低字节。

接收数据从接收端口到各级缓冲器的数据流示意图如下所示:

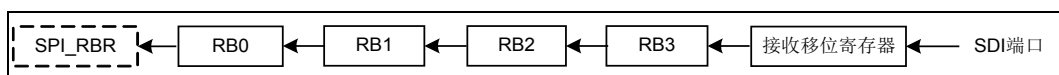


图 5-28 SPI 接收数据流示意图

接收缓冲器 RB0~RB3 的数据移到下一级缓冲器后, 会清除其接收满标志 RBFF0~RBFF3。

当 4 级接收缓冲器和 1 级接收移位寄存器均满时, 如果再次接收到数据位, 会置起 SPI\_IF 寄存器中的接收数据溢出中断标志 ROIF, 同时不会接收新数据, 缓冲器数据仍保持。

支持接收缓冲器满中断, 配置 SPI\_IE 寄存器中的 RBIM 位, 可选择中断模式。

RBIM<1:0>=00, 为字节满产生中断, 即接收缓冲器 RB0 为满时, 会置起 SPI\_IF 寄存器中的中断标志 RBIF;

RBIM<1:0>=01, 为半字满产生中断, 即接收缓冲器 RB0 和 RB1 均为满时, 会置起中断标志 RBIF;

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

### 5.4.7 SPI通讯控制

配置 SPI 通讯模式, 数据格式; 对主控制模式, 还需配置 SPI\_CON 寄存器的 CKS 位, 设定传输时钟速率, 并选择是否使能接收延迟和发送间隔, 对从动模式, 传输时钟由主机方提供; 配置 SPI\_CON 寄存器中的 EN 位和 REN 位, 使能数据发送和接收; 将要发送的数据写入发送数据寄存器 SPI\_TBW, 就可以开始数据的发送, 读取接收数据寄存器 SPI\_RBR, 可以获得接收到的数据。

SPI 主控模式下, 发送缓冲器和发送移位寄存器中的数据发送完毕后, 进入空闲状态, 会置起空闲标志 SPI\_STA 寄存器的 IDLE 位, 并且产生 SPI\_IF 寄存器的空闲中断标志 IDIF。

SPI 从动模式下, 如果发送缓冲器 TB0~TB3 和发送移位寄存器均为空时, 又收到主机提

供的通讯时钟，则会置起发送错误中断标志 SPI\_IF 寄存器的 TEIF。

SPI 从动模式，支持片选信号变化中断，配置 SPI\_IE 寄存器的 NSSIE，可启用该中断。

配置 SPI\_CON 寄存器的 RST，可将 SPI 通讯模块软件复位，复位后：禁止数据通讯，SPI\_CON 寄存器的 EN=0；禁止相关中断，SPI\_IE 寄存器的 TBIE=0, TBWEIE=0, RBIE=0, TEIE=0, ROIE=0, IDIE=0, NSSIE=0；复位相关中断标志为默认值，SPI\_IF 寄存器中的 TBIF=1, TBWEIF=0, RBIF=0, TEIF=0, ROIF=0, IDIF=0, NSSIF=0；置起空闲标志，SPI\_STA 寄存器的 IDLE=1；置起各发送缓冲器空标志 TBEF0/TBEF1/TBEF2/TBEF3=1；清除各接收缓冲器满标志 RBFF0/RBFF1/RBFF2/RBFF3=0。

### 5.4.8 SPI延迟接收功能

SPI 通讯时，是利用时钟的上升/下降沿分别对数据的发送和接收进行同步。正常通讯时，对主机接收数据来说，从机送出的数据应在半个时钟周期内，到达主机接收端口，否则会造成主机接收数据的丢失。

SPI 主控模式支持延迟接收功能，配置 SPI\_CON 寄存器中的 DRE 位，可启用该功能，主机可再延迟半个时钟周期，在下一个发送时钟边沿处，进行数据的接收采集。所以延迟接收功能启用后，从机发送端口和主机接收端口之间的线路延时，最大可接近 1 个通讯时钟周期。

举例说明 SPI 延迟接收功能：SPI\_CON 寄存器中的 DFS<1:0>=00，上升沿发送（先），下降沿接收（后）。

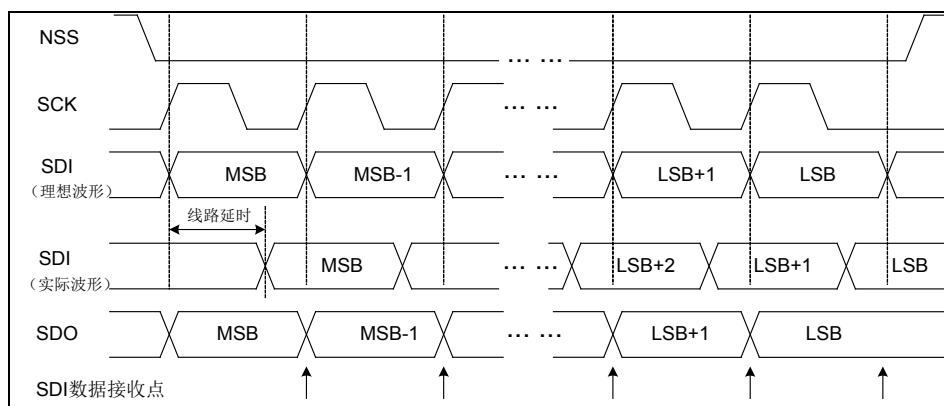


图 5-29 SPI 延迟接收功能波形示意图

### 5.4.9 SPI数据帧发送间隔功能

SPI 主控模式支持数据帧发送间隔功能，配置 SPI\_CON 寄存器的 TME 位，可启用该功能，配置 SPI\_CON 寄存器的 TMP<5:0>位，可设定发送间隔周期。当启用 SPI 数据帧发送间隔功能时，每帧数据发送完成后，会等待预先设定的发送间隔时间，再发送下一帧数据。



### 5.4.10 特殊功能寄存器

#### SPI 控制寄存器 (SPI\_CON)

偏移地址: 00<sub>H</sub>

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

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

—	bit31-24	—	—
TMP<5:0>	bit23-18	R/W	<b>SPI 帧发送间隔周期设置位 (仅主控模式支持)</b>
TMS	bit17	R	<b>SPI 帧发送间隔状态标志位 (仅主控模式支持)</b> 0: 非发送间隔状态 1: 发送间隔状态
TME	bit16	R/W	<b>SPI 帧发送间隔使能位 (仅主控模式支持)</b> 0: 禁止 1: 使能
CKS<7:0>	bit15-8	R/W	<b>SPI 通讯时钟设置位 (仅主控模式支持)</b>
DFS<1:0>	bit7-6	R/W	<b>SPI 通讯数据格式</b> 00: 上升沿发送 (先), 下降沿接收 (后) 01: 下降沿发送 (先), 上升沿接收 (后) 10: 上升沿接收 (先), 下降沿发送 (后) 11: 下降沿接收 (先), 上升沿发送 (后)
DRE	bit5	R/W	<b>SPI 延迟接收使能位 (仅主控模式支持)</b> 0: 禁止 1: 使能
—	bit4	—	—
REN	bit3	R/W	<b>SPI 接收使能位</b> 0: 禁止 1: 使能 (需 EN 同时使能)
MS	bit2	R/W	<b>SPI 通讯模式选择位</b> 0: 主控模式 1: 从动模式
RST	bit1	W	<b>SPI 软件复位</b> 0: 读取时始终为 0 1: 软件复位, 自动清零
EN	bit0	R/W	<b>SPI 通讯使能位</b> 0: 禁止 1: 使能 (SPI 通讯使能, 但仅使能数据发送)

注 1: SPI 通讯时钟特率计算公式如下:

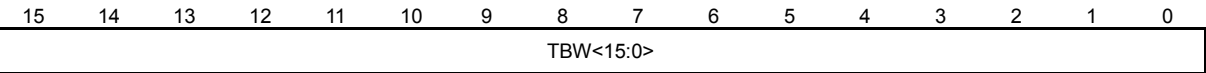
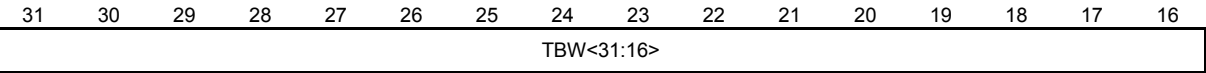
CKS<7:0>=0x00 时: FPCLK;

CKS<7:0>=0x01~0xFF 时: FPCLK/(CKS \* 2)。  
注 2: SPI 帧发送间隔周期计算公式如下:  
TSCK \* (1 + TMP), 即时间间隔为 1~64 个通讯时钟周期 TSCK。

**SPI 发送数据写入寄存器 (SPI\_TBW)**

偏移地址: 08<sub>H</sub>

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

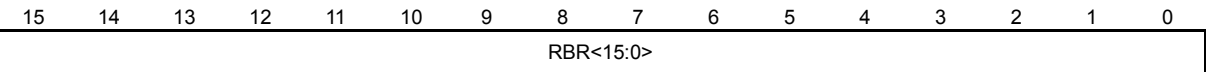
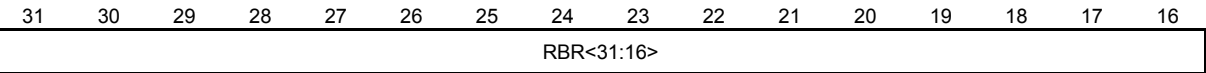


TBW<31:0>	bit31-0	W	<b>写入的发送数据</b> 字节写入时: 仅允许对 TBW<7:0>写入 半字写入时: 仅允许对 TBW<15:0>写入 字写入时: 对 TBW<31:0>写入
-----------	---------	---	--

**SPI 接收数据读取寄存器 (SPI\_RBR)**

偏移地址: 0C<sub>H</sub>

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

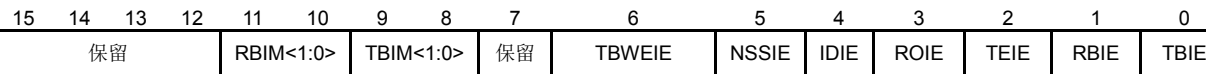
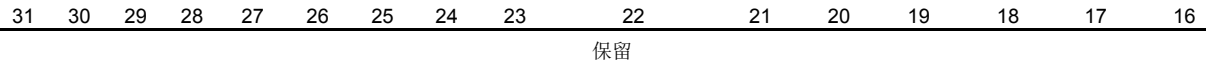


RBR<31:0>	bit31-0	R	<b>读取的接收数据</b> 字节读取时: 仅允许对 RBR<7:0>读取 半字读取时: 仅允许对 RBR<15:0>读取 字读取时: 对 RBR<31:0>读取
-----------	---------	---	--

**SPI 中断使能寄存器 (SPI\_IE)**

偏移地址: 10<sub>H</sub>

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



—	bit31-12	—	—
---	----------	---	---

RBIM<1:0>	bit11-10	R/W	<b>SPI 接收缓冲器满中断模式选择位</b> 00: 字节满产生中断 01: 半字满产生中断 10: 字满产生中断 11: 保留
TBIM<1:0>	bit9-8	R/W	<b>SPI 发送缓冲器空中断模式选择位</b> 00: 字节空产生中断 01: 半字空产生中断 10: 字空产生中断 11: 保留
—	bit7	—	—
TBWEIE	bit6	R/W	<b>SPI 发送数据写错误中断使能位</b> 0: 禁止 1: 使能
NSSIE	bit5	R/W	<b>SPI 片选变化中断使能位 (仅从动模式支持)</b> 0: 禁止 1: 使能
IDIE	bit4	R/W	<b>SPI 空闲状态中断使能位 (仅主控模式支持)</b> 0: 禁止 1: 使能
ROIE	bit3	R/W	<b>SPI 接收数据溢出中断使能位</b> 0: 禁止 1: 使能
TEIE	bit2	R/W	<b>SPI 发送数据错误中断使能位 (仅从动模式支持)</b> 0: 禁止 1: 使能
RBIE	bit1	R/W	<b>SPI 接收缓冲器满中断使能位</b> 0: 禁止 1: 使能
TBIE	bit0	R/W	<b>SPI 发送缓冲器空中断使能位</b> 0: 禁止 1: 使能

**SPI 中断标志寄存器 (SPI\_IF)**

偏移地址: 14<sub>H</sub>

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

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

—	bit31-7	—	—
TBWEIF	bit6	R/W	<b>SPI 发送数据写错误中断标志位</b>

			<p>0: 未发生写错误</p> <p>1: 发生写错误, 可能会出现下列错误: 对 TBW 字写入时, 发送缓冲器 TB0/1/2/3 未全空; 对 TBW 半字写入时, 发送缓冲器 TB0/1/2/3 未半空; 对 TBW 字节写入时, 发送缓冲器 TB0/1/2/3 全满; 对 TBW&lt;31:16&gt;进行半字写入; 对 TBW&lt;31:8&gt;进行字节写入。</p> <p>软件写 1 清除标志位, 写 0 无效</p>
NSSIF	bit5	R/W	<p><b>SPI 片选变化中断标志位 (仅从动模式支持)</b></p> <p>0: 片选信号未发生变化</p> <p>1: 片选信号发生变化</p> <p>软件写 1 清除标志位, 写 0 无效</p>
IDIF	bit4	R/W	<p><b>SPI 空闲中断标志位 (仅主控模式支持)</b></p> <p>0: 未进入空闲状态</p> <p>1: 进入空闲状态</p> <p>软件写 1 清除标志位, 写 0 无效; 或软件写寄存器 SPI_TBW 清除标志位</p>
ROIF	bit3	R/W	<p><b>SPI 接收数据溢出中断标志位</b></p> <p>0: 未溢出</p> <p>1: 溢出</p> <p>软件写 1 清除标志位, 写 0 无效</p>
TEIF	bit2	R/W	<p><b>SPI 发送错误中断标志位 (仅从动模式支持)</b></p> <p>0: 未发生发送错误</p> <p>1: 发生发送错误: 发送缓冲器和发送移位寄存器全空时, 又收到主控方提供的通讯时钟</p> <p>软件写 1 清除标志位, 写 0 无效</p>
RBIF	bit1	R	<p><b>SPI 接收缓冲器满中断标志位</b></p> <p>读 RBR 可清除中断标志</p>
TBIF	bit0	R	<p><b>SPI 发送缓冲器空中断标志位</b></p> <p>写 TBW 可清除中断标志</p>

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

**SPI 发送缓冲寄存器 (SPI\_TB)**

偏移地址: 18<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB3<7:0>								TB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB1<7:0>								TB0<7:0>							

TB3<7:0>	bit31-24	R	发送数据缓冲器 3
TB2<7:0>	bit23-16	R	发送数据缓冲器 2
TB1<7:0>	bit15-8	R	发送数据缓冲器 1
TB0<7:0>	bit7-0	R	发送数据缓冲器 0

**SPI 接收缓冲寄存器 (SPI\_RB)**

偏移地址: 1C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB3<7:0>								RB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1<7:0>								RB0<7:0>							

RB3<7:0>	bit31-24	R	接收数据缓冲器 3
RB2<7:0>	bit23-16	R	接收数据缓冲器 2
RB1<7:0>	bit15-8	R	接收数据缓冲器 1
RB0<7:0>	bit7-0	R	接收数据缓冲器 0

**SPI 状态寄存器 (SPI\_STA)**

偏移地址: 20<sub>H</sub>

复位值: 00000000\_00000001\_00001111\_10000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															IDLE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBFF3	RBFF2	RBFF1	RBFF0	TBEF3	TBEF2	TBEF1	TBEF0	NSS	保留						

—	bit31-17	—	—
IDLE	bit16	R	<b>SPI 空闲标志位 (仅主控模式支持)</b> 0: 非空闲状态 1: 空闲状态
RBFF3	bit15	R	接收缓冲器 <b>RB3</b> 满标志位 0: 空 1: 满
RBFF2	bit14	R	接收缓冲器 <b>RB2</b> 满标志位 0: 空 1: 满
RBFF1	bit13	R	接收缓冲器 <b>RB1</b> 满标志位 0: 空 1: 满

RBF0	bit12	R	接收缓冲器 <b>RB0</b> 满标志位 0: 空 1: 满
TBEF3	bit11	R	发送缓冲器 <b>TB3</b> 空标志位 0: 满 1: 空
TBEF2	bit10	R	发送缓冲器 <b>TB2</b> 空标志位 0: 满 1: 空
TBEF1	bit9	R	发送缓冲器 <b>TB1</b> 空标志位 0: 满 1: 空
TBEF0	bit8	R	发送缓冲器 <b>TB0</b> 空标志位 0: 满 1: 空
NSS	bit7	R	<b>SPI</b> 片选标志位 (仅从动模式支持) 0: 选中 1: 未选中
—	bit6-0	—	—

#### 5.4.11 SPI应用说明

芯片支持 1 个 SPI 同步串口通讯控制器，为 SPI。

## 5.5 I2C总线串口通讯控制器（I2C）

### 5.5.1 概述

- ◇ 支持单主控模式
  - 支持自动重复寻呼功能
  - 支持自动发送“停止位”功能
  - 支持数据应答延迟功能
  - 支持数据帧传输间隔功能
  - 支持软件触发“起始位”
  - 支持软件触发“停止位”
  - 支持软件触发数据接收，接收模式可配
- ◇ 支持从动模式
  - 支持7位从机地址可配
  - 支持从机地址匹配中断标志
  - 支持接收“停止位”中断标志
  - 支持时钟线自动下拉等待请求功能
  - 支持自动发送“未应答”功能
- ◇ 支持4级发送缓冲器和4级接收缓冲器
- ◇ 通讯端口 SCL 和 SDA，均支持输出模式可配置：推挽输出或开漏输出
- ◇ 通讯端口 SCL 和 SDA 支持16倍速采样器可配置
- ◇ 支持发送和接收缓冲器空/满中断
- ◇ 支持起始位中断、停止位中断
- ◇ 支持接收数据溢出中断、发送数据写错误中断

### 5.5.2 结构框图

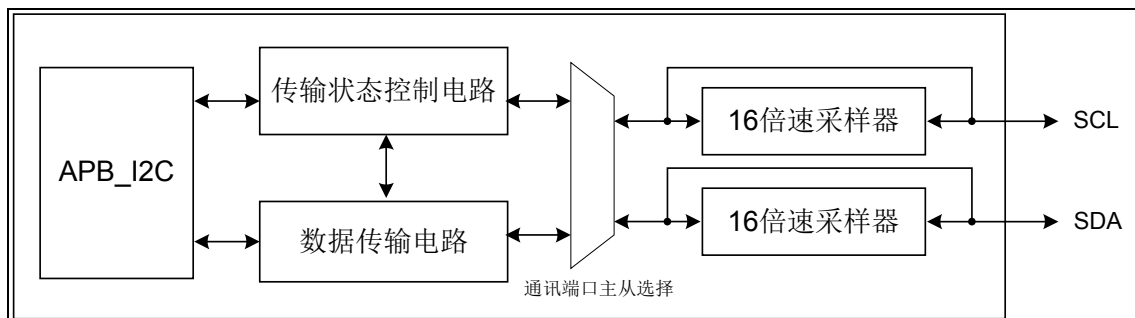


图 5-30 I2C 电路结构框图

### 5.5.3 I2C总线基本原理

#### 5.5.3.1 I2C通讯协议

I2C 总线通讯协议，读写操作示意图如下所示：

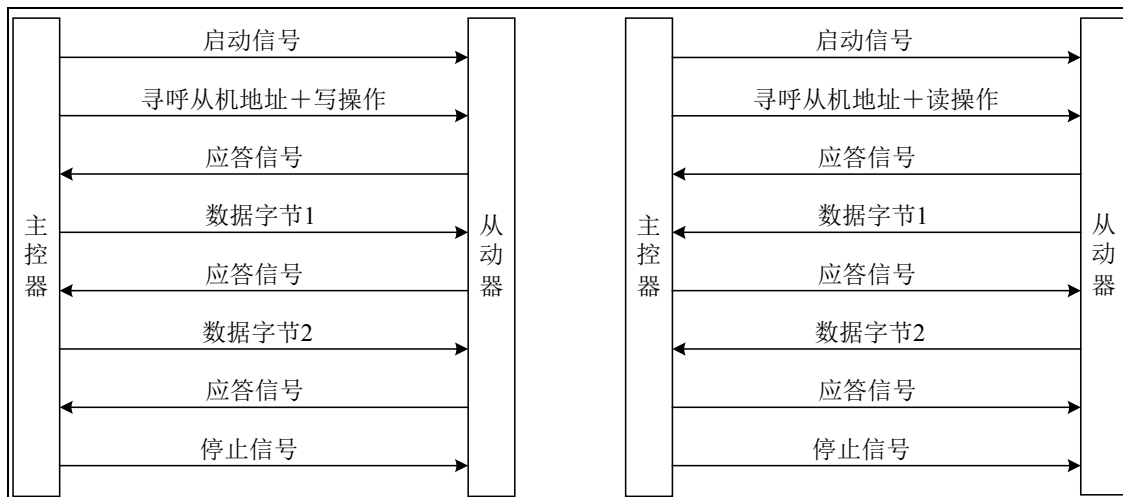


图 5-31 I2C 总线通讯协议示意图

I2C 通讯由主控器发起，发送启动信号 S 控制总线，发送停止信号 P 释放总线。

I2C 总线上可以同时有多个主控器（前提是每个主控器都支持多主机仲裁机制），并至少需要一个从动器，且每一个从动器都必须有一个独立且唯一的寻呼地址。

主控器在发送启动信号后，紧接着发送寻呼从机地址和读写控制位。

读写控制位  $\overline{R/\overline{W}}$ ，用于通知从动器数据传送的方向，“0”表示由主控器向从动器“写”数据，“1”表示由主控器向从动器“读”数据。

I2C 通讯协议支持应答机制，即发送方每传送一个字节的的数据（包括寻呼地址），接收方必须反馈一个应答信号（ACK 或 NACK），发送方再根据应答信号进行下一步的操作。

如果主控器和从动器的时钟端口（SCL）都使用输出开漏设计，且主控器支持时钟线等待请求操作，那么从动器可以在时钟线为低电平时下拉时钟线，使主控器等待从动器，直到从动器释放时钟线。

I2C 通讯时，每个数据字节在传输时都是高位在前，低位在后。

I2C 通讯时，数据线 SDA 的数据信号电平，只在时钟线 SCL 的低电平期间变化，在 SCL 高电平期间应保持稳定。如果 SDA 电平在 SCL 高电平期间变化，则会触发起始位或停止位，由高到低变化触发起始位，由低到高变化触发停止位。

#### 5.5.3.2 I2C数据传输格式

I2C 通讯时，根据从动器的具体设计规格，确定实际的数据传输格式。以下只介绍一种常用的 I2C 通讯数据传输格式：



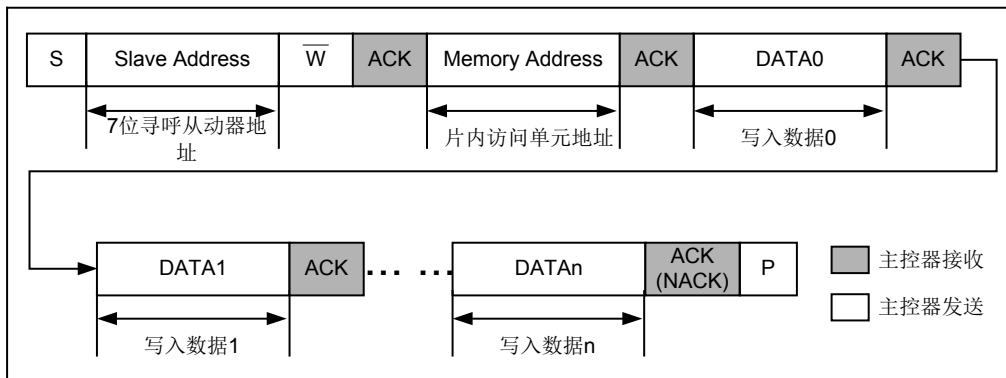


图 5-32 I2C 主控器写入从动器数据示意图

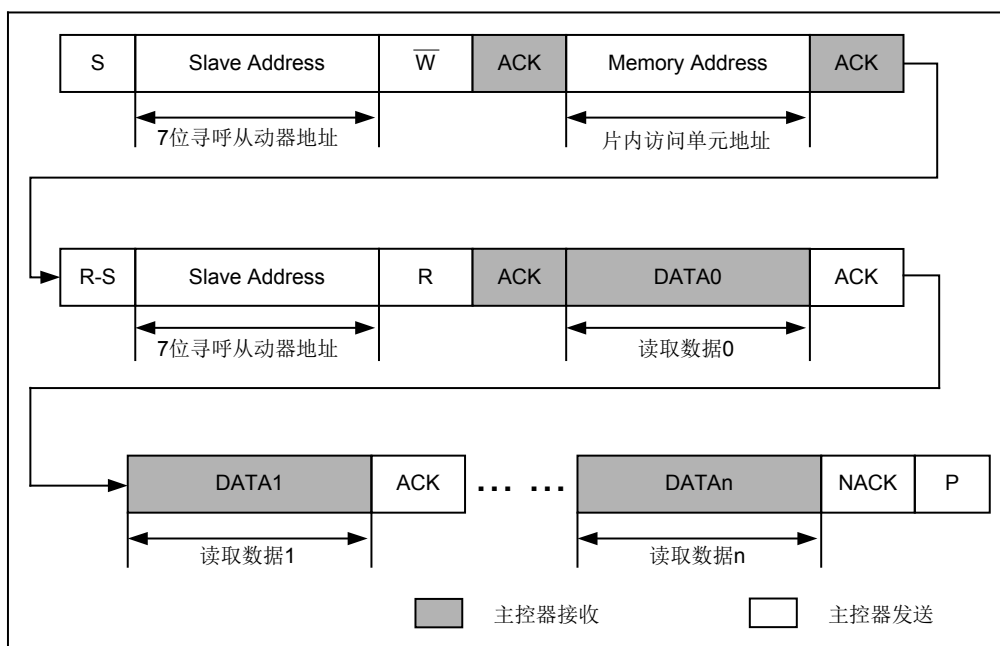


图 5-33 I2C 主控器读取从动器数据示意图

#### 5.5.4 I2C通讯端口配置

I2C 通讯端口 SCL 和 SDA，均支持推挽输出和开漏输出两种模式，配置 I2C\_CON 寄存器的 SCLOD 位和 SDAOD 位，可分别进行选择。

推挽输出是 I/O 端口的标准输出，输出数据 0 和 1 时，I/O 端口电平也分别为 0 和 1。

对推挽输出模式，存在端口电平冲突的风险。例如：当主控器输出 0，而从动器输出 1 时，会发生端口信号电平冲突，导致端口状态不确定。

开漏输出是 I2C 总线协议中的标准模式，可以避免端口电平冲突问题。开漏输出端口的示意图如下所示：

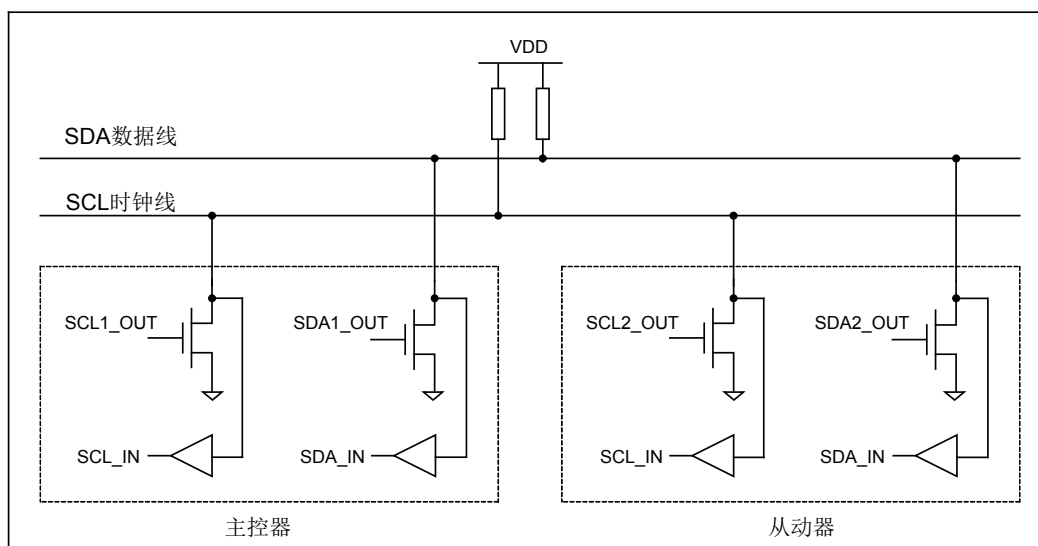


图 5-34 开漏输出端口示意图

开漏输出端口的高电平由 I2C 总线的上拉电阻提供，而低电平由主控器与从动器共同决定。任何一方都可以将总线电平下拉到 0，但只有当双方都释放总线后，总线电平才能被上拉到 1。

### 5.5.5 I2C 时基定时器与 16 倍速采样器

通讯端口 SCL 和 SDA 支持 16 倍采样器，配置 I2C\_CON 寄存器的 SCLSE 和 SDASE 可使能两个端口的采样器。I2C 时基定时器的计数周期，作为 16 倍速采样器的采样周期。

在 I2C 主控模式下，I2C 时基定时器还用于提供通讯传输波特率。

如果需要使用 16 倍速采样器，或使用 I2C 主控模式，均必须使能 I2C 时基定时器。I2C\_CON 寄存器中配置 TJE，可使能该时基定时器，配置 TJP，可设定时基定时周期。

I2C 主控模式下，总线端口信号的时序参数如下：

参数名称	参数符号	使能 16 倍速采样器	禁止 16 速倍采样器
启动/重新启动位建立时间	T <sub>SU:S</sub>	$> T_{osc} \times (T_{JP}+1) \times 12$	$T_{osc} \times (T_{JP}+1) \times 8$
启动/重新启动位保持时间	T <sub>HD:S</sub>	$> T_{osc} \times (T_{JP}+1) \times 12$	$T_{osc} \times (T_{JP}+1) \times 8$
停止位建立时间	T <sub>SU:P</sub>	$> T_{osc} \times (T_{JP}+1) \times 12$	$T_{osc} \times (T_{JP}+1) \times 8$
停止位保持时间	T <sub>HD:P</sub>	$> T_{osc} \times (T_{JP}+1) \times 12$	$T_{osc} \times (T_{JP}+1) \times 8$
数据/应答位建立时间	T <sub>SU:DA</sub>	$> T_{osc} \times (T_{JP}+1) \times 4$	$T_{osc} \times (T_{JP}+1) \times 4$
数据/应答位保持时间	T <sub>HD:DA</sub>	$> T_{osc} \times (T_{JP}+1) \times 8$	$T_{osc} \times (T_{JP}+1) \times 4$
通讯时钟高电平脉宽	T <sub>HIGH</sub>	$T_{osc} \times (T_{JP}+1) \times 12$	$T_{osc} \times (T_{JP}+1) \times 8$
通讯时钟低电平脉宽	T <sub>LOW</sub>	$T_{osc} \times (T_{JP}+1) \times 12$	$T_{osc} \times (T_{JP}+1) \times 8$

表 5-1 I2C 总线端口信号的时序参数列表

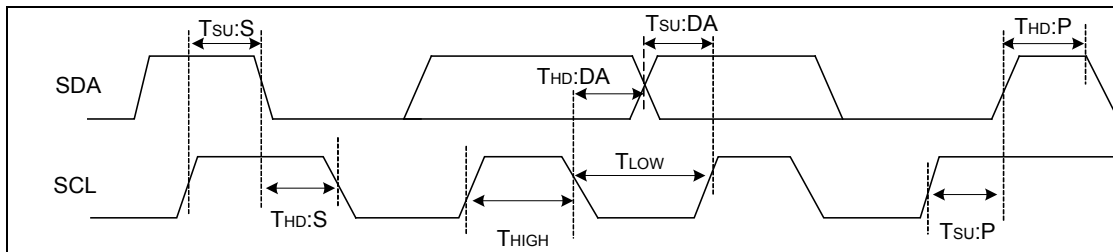


图 5-35 I2C 总线端口信号的波形示意图

使能 16 倍速采样器后，采样器对 I2C 总线进行采样，而 I2C 总线由于上拉电阻的作用，开始时的电平是不稳定的，当 I2C 总线电平稳定后，采样器的输出才能稳定。因此，这段不稳定的时间将造成 I2C 传输波特率的下降，下降程度取决于总线电平上升边沿的时间。

I2C 主控模式下，传输波特率计算公式如下（Fosc 为系统时钟频率）：

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

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

### 5.5.6 I2C 通讯发送器

支持 4 级发送缓冲器 TB0, TB1, TB2, TB3 和 1 级发送移位寄存器，可进行数据的连续发送，直到发送缓冲器和移位寄存器全空，最多可连续写入和发送 5 帧数据。发送缓冲器 TB0~TB3 为只读寄存器，只能通过发送数据寄存器 I2C\_TBW 写入。

发送数据寄存器 I2C\_TBW 为一个虚拟地址单元，物理上不存在实际的寄存器电路，写该寄存器地址单元时，实际上是将发送数据写入到发送缓冲器 TB0~TB3 中，再传输到发送移位寄存器，通过数据端口 SDA 进行数据发送。

发送数据寄存器 I2C\_TBW 支持 3 种写入方式：字节写入，半字写入和字写入。

字节方式写入 I2C\_TBW 时，发送数据被写入到发送缓冲器 TB0；半字方式写入 I2C\_TBW 时，发送数据被同时写入到发送缓冲器 TB0 和 TB1，其中低字节存放在 TB1 中；字方式写入 I2C\_TBW 时，发送数据被同时写入到发送缓冲器 TB0, TB1, TB2 和 TB3，其中低字节存放在 TB3 中。

发送数据从写入到发送到端口的数据流示意图如下所示：

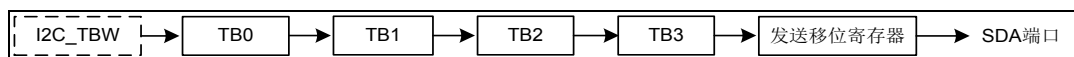


图 5-36 I2C 发送数据流示意图

支持发送缓冲器空中断，配置 I2C\_IE 寄存器中的 TBIM，可选择中断模式。

TBIM<1:0>=00，为字节空产生中断，即发送缓冲器 TB0 为空时，会置起 I2C\_IF 寄存器的中断标志 TBIF；

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

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

支持发送数据寄存器 I2C\_TBW 误写中断，当对 I2C\_TBW 的写入方式，与发送缓冲器 TB0~TB3 的空闲状态冲突时，或写入操作访问错误时，会置起 I2C\_IF 寄存器中的误写中断标志位 TBWEIF。

### 5.5.7 I2C通讯接收器

支持 4 级接收缓冲器 RB0, RB1, RB2, RB3 和 1 级接收移位寄存器，可进行数据的连续接收，直到接收缓冲器和移位寄存器全满，最多可连续接收 5 帧数据，再执行数据读取操作。读取接收数据寄存器 I2C\_RBR，可得到接收的数据，对应的接收缓冲器清除 I2C\_STA 寄存器中的满标志 RBFF0~RBFF3；也可以读取接收缓冲器 RB0~RB3 得到接收的数据，但不会清除 I2C\_STA 寄存器中的满标志 RBFF0~RBFF3。

接收数据寄存器 I2C\_RBR 为一个虚拟地址单元，物理上不存在实际的寄存器电路，读该寄存器地址单元时，实际上是读取接收缓冲器 RB0~RB3 中的数据。

接收数据寄存器 I2C\_RBR 支持 3 种读取方式：字节读取，半字读取和字读取。

字节方式读取 I2C\_RBR 时，实际是读取接收缓冲器 RB0 的数据；半字方式读取 I2C\_RBR 时，实际是同时读取接收缓冲器 RB0 和 RB1 的数据，其中 RB0 中的数据为低字节；字方式读取 I2C\_RBR 时，实际是同时读取接收缓冲器 RB0, RB1, RB2 和 RB3，其中 RB0 中的数据为低字节。

接收数据从数据端口到各级缓冲器的数据流示意图如下所示：

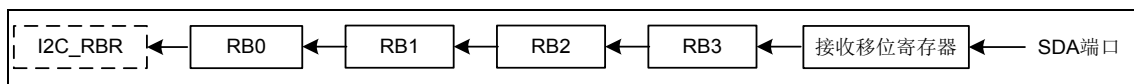


图 5-37 I2C 接收数据流示意图

接收缓冲器 RB0~RB3 的数据移到下一级缓冲器后，会清除其接收 I2C\_STA 寄存器中的满标志 RBFF0~RBFF3。

当 4 级接收缓冲器和 1 级接收移位寄存器均满时，会立即置起 I2C\_IF 寄存器中的接收数据溢出中断标志 ROIF，同时不会接收新数据。

支持接收缓冲器满中断，配置 I2C\_IE 寄存器中的 RBIM，可选择中断模式。

RBIM<1:0>=00，为字节满产生中断，即接收缓冲器 RB0 为满时，会置起 I2C\_IF 寄存器的中断标志 RBIF；

RBIM<1:0>=01，为半字满产生中断，即接收缓冲器 RB0 和 RB1 均为满时，会置起中断标志 RBIF；

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

### 5.5.8 I2C通讯控制

I2C\_CON 寄存器中，配置 RST，可将 I2C 通讯模块软件复位，复位后：禁止数据通讯 EN=0；I2C\_IE 寄存器中，禁止相关中断，SRIE=0，SPIE=0，TBIE=0，RBIE=0，TEIE=0，ROIE=0，NAIE=0；I2C\_IF 寄存器中复位相关中断标志为默认值，SRIF=0，SPIF=0，TBIF=1，RBIF=0，TEIF=0，ROIF=0，NAIF=0；I2C\_STA 寄存器中，置起空闲标志 IDLE=1；

置起各发送缓冲器空标志 TBEF0~TBEF3=1；清除各接收缓冲器满标志 RBFF0~RBFF3=0。

### 5.5.8.1 I2C起始位

I2C\_MOD 寄存器中配置 SRT，可触发 I2C 发送起始位，启动或重新启动一次传输操作，发送寻呼从动器地址。起始位波形图如下：

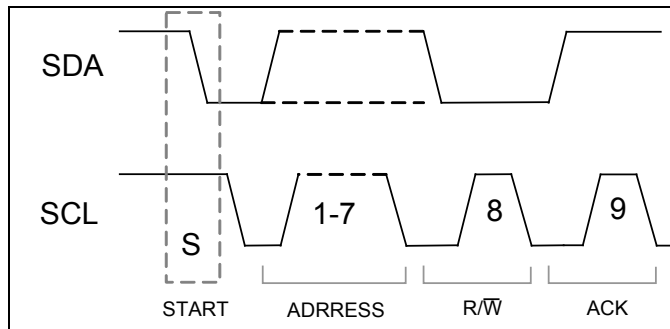


图 5-38 I2C 起始位波形图

I2C 主控模式，支持自动寻呼功能。I2C\_MOD 寄存器中配置 SRAE，可使能自动寻呼功能，I2C 主控器会自动判断“地址应答”位，若该“地址应答”位为未应答 NACK，则自动发送起始位，重新启动本次地址寻呼操作，并且直到接收到应答 ACK 才停止继续重新启动。在芯片应用时，使能 I2C 自动寻呼功能前，需确保被寻呼的地址是真实存在的，否则会造成主控器芯片持续重新启动地址寻呼。

举例说明自动寻呼功能：芯片通过 I2C 向 EEPROM 器件写入数据时，存在写等待时间。在 EEPROM 存储器本身写数据期间，主控器芯片寻呼该器件时，将收到未应答 NACK。可有两种方法解决 EEPROM 写数据期间的地址寻呼问题：一种是主控器芯片设定寻呼间隔，在 EEPROM 数据写操作完成后，再寻呼该器件；另一种是主控器芯片启动自动寻呼功能，持续寻呼该器件，直到收到应答 ACK 为止。

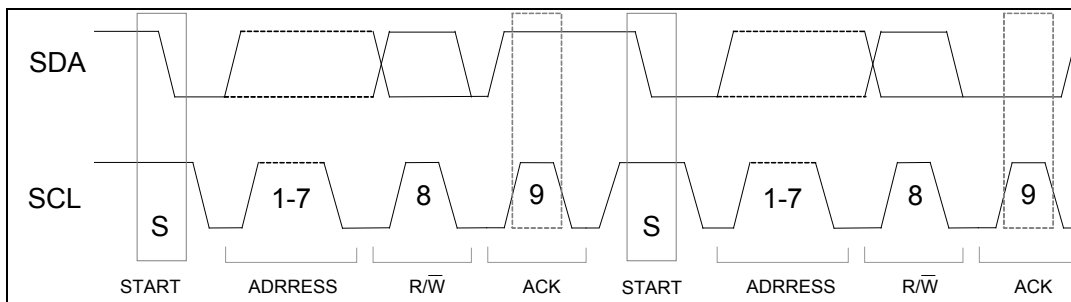


图 5-39 I2C 自动寻呼波形图

### 5.5.8.2 I2C停止位

I2C\_MOD 寄存器中配置 SPT，可触发 I2C 发送停止位，结束本次传输操作。停止位波形图如下：

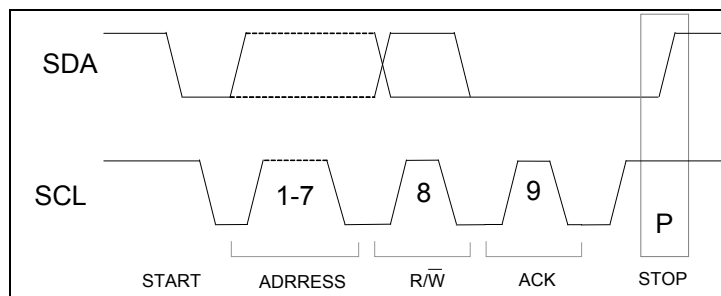


图 5-40 I2C 停止位波形图

I2C 主控模式，支持自动结束功能。配置 I2C\_MOD 寄存器中的 SPAE，可启用自动结束功能，在发送 NACK 或接收 NACK 后，自动发送停止位，结束本次不成功的传输操作。自动结束功能的优先级小于自动寻呼功能。

### 5.5.8.3 I2C 应答延迟功能

I2C 的主控模式，支持应答延迟功能，I2C\_MOD 寄存器中配置 ADE，可启用该功能，I2C\_MOD 寄存器中配置 ADLY，可设定应答延迟的时间。应答延迟功能启用后，I2C 主控器将延迟发送通讯时钟 SCL 的应答位脉冲。

当从动器不能按照正常数据的通讯速率，对应答位进行接收和发送时，主控器可启用应答延迟功能，并根据从动器的具体设计规格，设定应答延迟时间。

举例说明 I2C 应答延迟功能：ADLY<2:0>=001，延迟时间为 1 个 T<sub>SCL</sub>，通讯波形示意图如下：

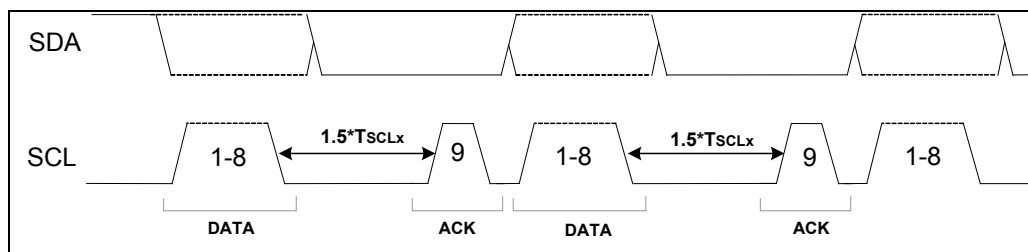


图 5-41 I2C 应答延迟功能波形示意图

### 5.5.8.4 I2C 数据帧传输间隔功能

I2C 主控模式，支持数据帧传输间隔功能，I2C\_MOD 寄存器中配置 TIS，可启用该功能，并设定间隔的时间。数据帧传输间隔时间设定后，在当前数据帧的应答位脉冲之后，I2C 主控器将延迟一段设定的时间，再发送下一个数据帧的通讯脉冲。

当从动器不能及时读取接收到的数据，或准备好发送的数据时，主控器可根据从动器的具体设计规格，设定数据帧的传输间隔。

举例说明 I2C 数据帧传输间隔功能：TIS<3:0>=0001，间隔时间为 1 个 T<sub>SCL</sub>，通讯波形示意图如下：

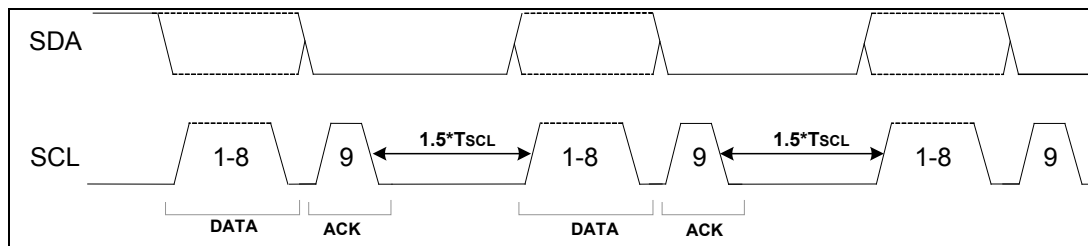


图 5-42 I2C 数据帧传输间隔功能波形示意图

### 5.5.8.5 I2C 时钟线自动下拉等待请求功能

I2C 从动模式，支持时钟线自动下拉等待请求功能，I2C\_MOD 寄存器中配置 CSE，可启用该功能。

为实现 I2C 时钟线的下拉等待请求功能，还需 I2C\_CON 寄存器中配置 SCLOD，将通讯端口 SCL 选择为开漏输出模式，通过上拉电阻提供高电平，使从动器可对时钟线下拉控制，使主控器等待。

在通常情况下，从动器处于释放时钟线的状态，时钟线 SCL 完全由主控器控制。但当从动器出现异常情况，短时间内无法继续进行数据传输时，从动器可以在时钟线 SCL 为低电平时输出 0（不可以在高电平时输出 0，否则会破坏数据传输过程），强行使 SCL 保持低电平，使主控器进入通讯等待状态，直到从动器释放时钟线。时钟线下拉等待请求波形示意图如下：

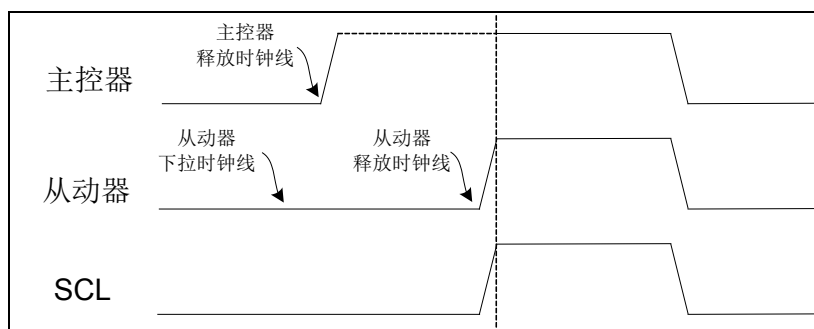


图 5-43 I2C 时钟线下拉等待波形示意图

I2C 从动模式下，时钟自动下拉等待请求功能使能后，当接收到本芯片寻呼地址和读操作位时，如果 I2C 的发送缓冲器和发送移位寄存器全空，则会自动将时钟线下拉；当接收到本芯片寻呼地址和写操作位时，如果 I2C 的接收缓冲器和接收移位寄存器全满，则会自动将时钟线下拉。

### 5.5.8.6 I2C 自动发送未应答功能

I2C 从动模式，支持自动发送未应答 NACK 功能，I2C\_MOD 寄存器中配置 ANAE，可启用该功能。从动器使能自动发送未应答功能时，不会强制控制时钟线，适用于通讯端口 SCL 选择为推挽输出或开漏输出模式两种情况。

I2C 从动模式下，自动发送未应答 NACK 功能使能后，当接收到本芯片寻呼地址和读操作位时，如果 I2C 的发送缓冲器和发送移位寄存器全空，则会自动发送未应答 NACK；当接收到本芯片寻呼地址和写操作位时，如果 I2C 的接收缓冲器和接收移位寄存器全满，则会自动发送未应答 NACK，通知主控器重新通讯。

### 5.5.9 特殊功能寄存器

#### I2C 控制寄存器 (I2C\_CON)

偏移地址: 00H

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

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

—	bit31-24	—	—
SA<7:0>	bit23-17	R/W	<b>从机地址位</b> 主控模式: 触发“启动/重新启动”操作时, 自动发送 从动模式: 接收到“启动/重新启动”后用于匹配比较
RW	bit16	R/W	<b>I2C 读写控制位</b> 0: 写操作 1: 读操作 主控模式: 该位可读可写, 触发“启动/重新启动”操作时, 自动发送该位 从动模式: 该位只可读, 从机地址匹配后, 硬件自动根据接收到的控制位值, 更新该位
TJP<7:0>	bit15-8	R/W	<b>I2C 时基定时周期设置位</b> 00~FF: 分别为 1~256 个 TPCLK
TJE	bit7	R/W	<b>I2C 时基定时器使能位</b> 0: 禁止 1: 使能
—	bit6	—	—
SDASE	bit5	R/W	<b>SDA 端口 16 倍速采样使能位</b> 0: 禁止 1: 使能
SCLSE	bit4	R/W	<b>SCL 端口 16 倍速采样使能位</b> 0: 禁止 1: 使能
SDAOD	bit3	R/W	<b>SDA 端口输出模式选择位</b> 0: 推挽输出 1: 开漏输出
SCLOD	bit2	R/W	<b>SCL 端口输出模式选择位</b> 0: 推挽输出 1: 开漏输出
RST	bit1	W	<b>I2C 软件复位</b> 0: 读取时始终为 0



			1: 软件复位, 自动清零
EN	bit0	R/W	<b>I2C 通讯使能位</b> 0: 禁止 1: 使能

**I2C 工作模式寄存器 (I2C\_MOD)**

偏移地址: 04H

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

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

—	bit31-25	—	—
TAS	bit24	R/W	<b>I2C 发送应答设置位 (仅从动模式支持)</b> 0: 发送 ACK 1: 发送 NACK
—	bit23-20	—	—
BLD	bit19	R/W	<b>I2C 总线释放功能控制位 (仅主控模式支持)</b> 0: 无效 1: SDA 端口为高电平, SCL 端口发送 8 个时钟。该功能需配合 I2C 起始位和停止位使用, 达到释放时钟线和数据线的目的。
RDT	bit18	R/W	<b>I2C 接收数据触发位 (仅主控模式支持)</b> 0: 无效 1: 开始发送通讯时钟, 接收数据, 由 RDM 位配置接收数据模式
SPT	bit17	R/W	<b>I2C 停止位触发位 (仅主控模式支持)</b> 0: 无效 1: 触发停止位
SRT	bit16	R/W	<b>I2C 起始位触发位 (仅主控模式支持)</b> 0: 无效 1: 触发起始位, 产生起始位发送完成中断标志
TIS<3:0>	bit15-12	R/W	<b>I2C 数据帧传输间隔设置位 (仅主控模式支持)</b> 0000: 禁止 0001~1111: 分别为 1~15 个 I2C 通讯时钟周期
ADE	bit11	R/W	<b>I2C 应答延迟使能位 (仅主控模式支持)</b> 0: 禁止 1: 使能
ADLY<2:0>	bit10-8	R/W	<b>I2C 应答延迟时间设置位 (仅主控模式支持)</b> 000: 0.5 个 I2C 通讯时钟周期 001: 1 个 I2C 通讯时钟周期

			010: 1.5 个 I2C 通讯时钟周期 011: 2 个 I2C 通讯时钟周期 100: 2.5 个 I2C 通讯时钟周期 101: 3 个 I2C 通讯时钟周期 110: 3.5 个 I2C 通讯时钟周期 111: 4 个 I2C 通讯时钟周期
SPAE	bit7	R/W	<b>I2C 自动结束使能位 (仅主控模式支持)</b> 0: 禁止 1: 使能 (当发送或接收 NACK 后, 自动发送停止位, 优先级小于 SRAE)
SRAE	bit6	R/W	<b>I2C 自动寻呼使能位 (仅主控模式支持)</b> 0: 禁止 1: 使能 (若寻呼地址的应答位为 NACK, 则自动重启动本次寻呼操作)
ANAE	bit5	R/W	<b>I2C 自动发送未应答使能位 (仅从动模式支持)</b> 0: 禁止 1: 使能
CSE	bit4	R/W	<b>I2C 时钟线自动下拉等待请求使能位 (仅从动模式支持)</b> 0: 禁止 1: 使能
RDM<2:0>	bit3-1	R/W	<b>I2C 接收模式选择位 (仅主控模式支持)</b> 000: 接收 1 字节, 发送 ACK 001: 接收 1 字节, 发送 NACK 010: 连续接收 2 字节, 每个字节发送 ACK 011: 连续接收 2 字节, 前 1 字节发送 ACK, 后 1 字节, 发送 NACK 100: 连续接收 4 字节, 每个字节发送 ACK 101: 连续接收 4 字节, 前 3 字节发送 ACK, 后 1 字节, 发送 NACK 110: 连续接收, 每个字节发送 ACK 111: 完成该字节接收, 发送 NACK
MS	bit0	R/W	<b>I2C 通讯模式选择位</b> 0: 主控模式 1: 从动模式

**I2C 中断使能寄存器 (I2C\_IE)**

偏移地址: 08<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIDLEIE	RBIM<1:0>	TBIM<1:0>	保留	NAIE	ROIE	TEIE	RBIE	TBIE	SPIE	SRIE				

—	bit31-13	—	—
TIDLEIE	bit12	R/W	<b>I2C 发送空闲中断使能位</b> 0: 禁止 1: 使能
RBIM<1:0>	bit11-10	R/W	<b>I2C 接收缓冲器满中断模式选择位</b> 00: 字节满产生中断 01: 半字满产生中断 10: 字满产生中断 11: 保留
TBIM<1:0>	bit9-8	R/W	<b>I2C 发送缓冲器空中断模式选择位</b> 00: 字节空产生中断 01: 半字空产生中断 10: 字空产生中断 11: 保留
—	bit7	—	—
NAIE	bit6	R/W	<b>I2C 未应答 NACK 中断使能位</b> 0: 禁止 1: 使能
ROIE	bit5	R/W	<b>I2C 接收数据溢出中断使能位</b> 0: 禁止 1: 使能
TEIE	bit4	R/W	<b>I2C 发送数据错误中断使能位</b> 0: 禁止 1: 使能
RBIE	bit3	R/W	<b>I2C 接收缓冲器满中断使能位</b> 0: 禁止 1: 使能
TBIE	bit2	R/W	<b>I2C 发送缓冲器空中断使能位</b> 0: 禁止 1: 使能
SPIE	bit1	R/W	<b>I2C 停止位中断使能位</b> 0: 禁止 1: 使能
SRIE	bit0	R/W	<b>I2C 起始位中断使能位</b> 0: 禁止 1: 使能

I2C 中断标志寄存器 (I2C\_IF)

偏移地址: 0C<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00000100<sub>b</sub>

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIDLECF	TIDLEIF	保留			NAIF	ROIF	TEIF	RBIF	TBIF	SPIF	SRIF			

—	bit31-13	—	—
TIDLECF	bit12	W	<b>I2C 发送空闲中断标志 TIDLEIF 清除标志位</b> 0: 无效 1: 清除中断标志 TIDLEIF
TIDLEIF	bit11	R	<b>I2C 发送空闲中断标志位</b> 0: 未产生空闲中断 1: 空闲中断标志
—	bit10-7	—	—
NAIF	bit6	R/W	<b>I2C 未应答中断标志位</b> 0: 未产生未应答 NACK 1: 产生未应答 NACK I2C 接收或发送 NACK 位后, 产生中断标志 软件写 1 清除标志位, 写 0 无效
ROIF	bit5	R/W	<b>I2C 接收数据溢出中断标志位</b> 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
TEIF	bit4	R/W	<b>I2C 发送错误中断标志位</b> 0: 未发生发送错误 1: 发生发送错误: 发送缓冲器和发送移位寄存器全空时, 又收到主控方提供的通讯时钟 软件写 1 清除标志位, 写 0 无效
RBIF	bit3	R	<b>I2C 接收缓冲器满中断标志位</b> 0: 非满 1: 满
TBIF	bit2	R	<b>I2C 发送缓冲器空中断标志位</b> 0: 非空 1: 空
SPIF	bit1	R/W	<b>I2C 停止位中断标志位</b> 0: 未产生停止位 1: 产生停止位 主控模式: 发送停止位后产生中断标志。 从动模式: 接收停止位后产生中断标志。 软件写 1 清除标志位, 写 0 无效
SRIF	bit0	R/W	<b>I2C 起始位中断标志位</b>

			<p>0: 未产生起始位 1: 产生起始位</p> <p>主控模式: 如果禁止自动寻呼, 发送完“起始位+地址+接收应答位后”产生中断标志。如果使能自动寻呼, 发送完“起始位+地址+接受应答位”, 并且应答位为 ACK 时, 产生中断标志。</p> <p>从动模式: 接收到“起始位+地址位”, 且地址匹配, 发送应答位后, 产生中断标志。</p> <p>软件写 1 清除标志位, 写 0 无效</p>
--	--	--	---

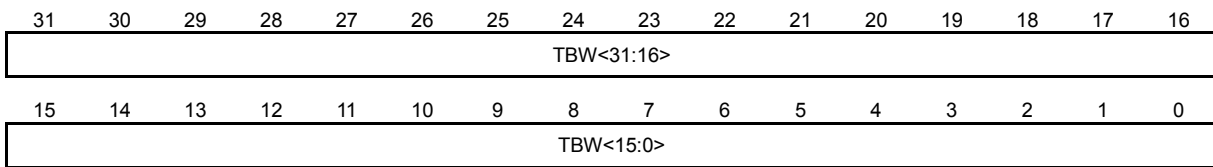
注 1: 对 I2C\_IF 寄存器中的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

注 2: 对发送空闲中断标志位 TIDLEIF, 写操作时对应于寄存器位 I2C\_IF<12>, 读操作时对应于 I2C\_IF<11>。

**I2C 发送数据写入寄存器 (I2C\_TBW)**

偏移地址: 10<sub>H</sub>

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

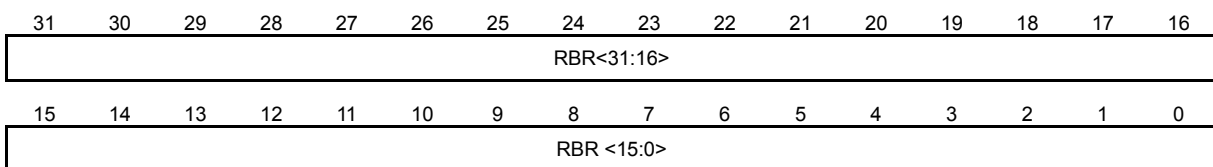


TBW<31:0>	bit31-0	W	<p><b>写入的发送数据</b></p> <p>字节写入时: 仅允许对 TBW&lt;7:0&gt;写入 半字写入时: 仅允许对 TBW&lt;15:0&gt;写入 字写入时: 对 TBW&lt;31:0&gt;写入</p>
-----------	---------	---	---

**I2C 接收数据读取寄存器 (I2C\_RBR)**

偏移地址: 14<sub>H</sub>

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



RBR<31:0>	bit31-0	R	<p><b>读取的接收数据</b></p> <p>字节读取时: 仅允许对 RBR&lt;7:0&gt;读取 半字读取时: 仅允许对 RBR&lt;15:0&gt;读取 字读取时: 对 RBR&lt;31:0&gt;读取</p>
-----------	---------	---	---

**I2C 发送缓冲寄存器 (I2C\_TB)**

偏移地址: 18<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB3<7:0>								TB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB1<7:0>								TB0<7:0>							

TB3<7:0>	bit31-24	R	发送数据缓冲器 3
TB2<7:0>	bit23-16	R	发送数据缓冲器 2
TB1<7:0>	bit15-8	R	发送数据缓冲器 1
TB0<7:0>	bit7-0	R	发送数据缓冲器 0

**I2C 接收缓冲寄存器 (I2C\_RB)**

偏移地址: 1C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB3<7:0>								RB2<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1<7:0>								RB0<7:0>							

RB3<7:0>	bit31-24	R	接收数据缓冲器 3
RB2<7:0>	bit23-16	R	接收数据缓冲器 2
RB1<7:0>	bit15-8	R	接收数据缓冲器 1
RB0<7:0>	bit7-0	R	接收数据缓冲器 0

**I2C 状态寄存器 (I2C\_STA)**

偏移地址: 20<sub>H</sub>

复位值: 00000000\_00000010\_00001111\_00000000<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														IDLE	ACK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBFF3	RBFF2	RBFF1	RBFF0	TBEF3	TBEF2	TBEF1	TBEF0	保留							

—	bit31-18	—	—
IDLE	bit17	R	<b>I2C 空闲标志位</b> 0: 非空闲状态 1: 空闲状态
ACK	bit16	R	<b>I2C 应答位</b> 0: 应答 ACK 1: 未应答 NACK

RBFF3	bit15	R	<b>RB3 满标志位</b> 0: 空 1: 满
RBFF2	bit14	R	<b>RB2 满标志位</b> 0: 空 1: 满
RBFF1	bit13	R	<b>RB1 满标志位</b> 0: 空 1: 满
RBFF0	bit12	R	<b>RB0 满标志位</b> 0: 空 1: 满
TBEF3	bit11	R	<b>TB3 空标志位</b> 0: 满 1: 空
TBEF2	bit10	R	<b>TB2 空标志位</b> 0: 满 1: 空
TBEF1	bit9	R	<b>TB1 空标志位</b> 0: 满 1: 空
TBEF0	bit8	R	<b>TB0 空标志位</b> 0: 满 1: 空
—	bit7-0	—	—

### 5.5.10 I2C应用说明

芯片支持 1 个 I2C 总线串口通讯控制器，为 I2C。

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

- 1) 主控模式下，在主程序中设置 Memory Address 之后，发送空闲标志（TIDLEIF）置 1 并触发中断，在中断服务程序中，将数据写入 I2C\_TBW 中，并开始发送数据；
- 2) 从动模式下，必须在主程序中等待 I2C\_STA 寄存器的 IDLE 置 1 之后，将以 Memory Address 为起始地址的第一个数据写入 I2C\_TBW 中。当主机发送读命令后，发送空闲标志（TIDLEIF）置 1 并触发中断，在中断服务程序中发送数据。在主机读完数据后，必须发送 STOP 位，否则，若要直接再次 START 读操作，可能导致数据发送错误。

I2C\_IE 寄存器的 TBIM（发送缓冲器中断模式选择位）会影响 TIDLEIF，因此，对于字节发送或半字发送，在 I2C 开始发送数据时最好使用字空产生中断模式（TBIM=2'b10），否则容易出错。如果使用字节空（TBIM=2'b00）或者半字空（TBIM=2'b01）中断模式，则第一次发送的数据必须将 I2C\_TBW 的四个字节填满。

使用发送空闲标志（TIDLEIF）的好处在于，在 I2C 发送空闲中断使能位（TIDLEIE）有效的情况下，只需清除发送空闲标志位（TIDLEIF），即可达到连续发送数据的目的。

## 5.6 模数转换器 (ADC)

### 5.6.1 概述

- ◇ 支持 12 位采样分辨率
- ◇ 支持 11 个模拟输入端
- ◇ 支持 12 位转换结果，可选择高位对齐或低位对齐格式
- ◇ 支持 ADC 中断标志 IF，可唤醒睡眠模式
- ◇ 支持正负向参考电压可配置
- ◇ 支持转换时钟可配置

### 5.6.2 结构框图

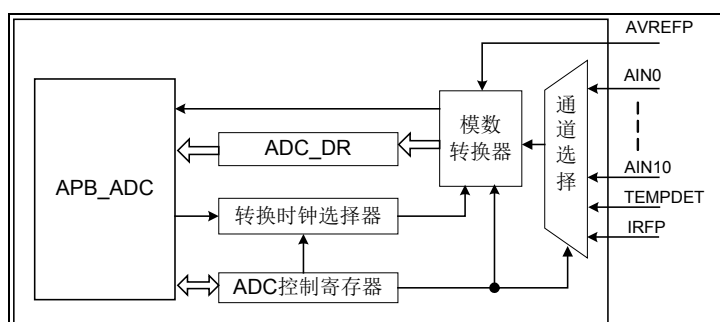


图 5-44 ADC 内部结构图

### 5.6.3 ADC数据转换

配置 ADC\_CHS 寄存器中的 CHS<3:0>位，可选择 ADC 模拟通道；ADC\_CON1 寄存器中，配置 CLKS 位，可选择工作时钟源；配置 CLKDIV<2:0>位，可选择时钟源预分频；配置 FM 位，可选择转换结果的数据格式；配置 VREFP<1:0>位，可选择正向参考电压；ADC\_CON0 寄存器中配置 EN 位，使能 ADC；最后配置 TRG 位，启动 ADC 转换，转换完成后，硬件电路自动将 TRG 位清零。

ADC 在每次转换完成后，会产生 ADC\_IF 寄存器的中断标志 IF，需软件清零；启动下一次 ADC 转换时，需重新配置 ADC\_CON0 寄存器的 TRG 位。

ADC 采样时间可选择硬件或软件控制，默认为硬件自动控制。

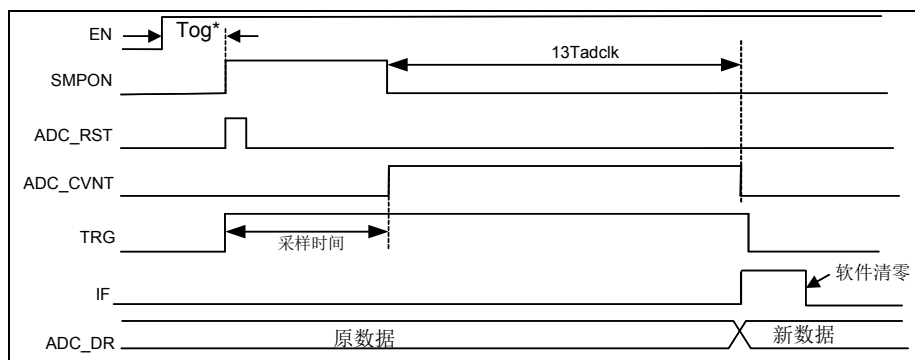


图 5-45 ADC 数据转换时序示意图 (ADC\_CON1 寄存器的 SMPS=0)



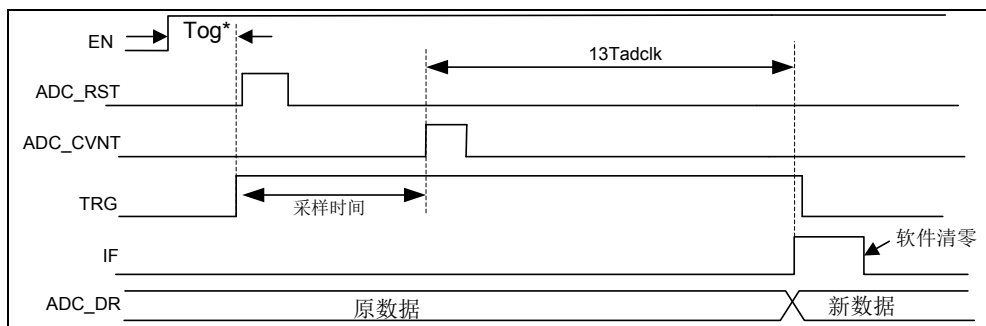


图 5-46 ADC 数据转换时序示意图 (ADC\_CON1 寄存器的 SMPS=1)

### 5.6.4 特殊功能寄存器

#### ADC 转换值寄存器 (ADC\_DR)

偏移地址: 00H

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

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

—	bit31-16	—	—
DR<15:0>	bit15-0	R	ADC 转换结果

#### ADC 控制寄存器 0 (ADC\_CON0)

偏移地址: 04H

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

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

—	bit31-2	—	—
TRG	bit1	R/W	<b>ADC 转换状态位</b> 0: ADC 未进行转换, 或 ADC 转换已完成 (硬件清 0, 且硬件清 0 优先) 1: ADC 转换正在进行, 该位置 1 启动 ADC 转换
EN	bit0	R/W	<b>ADC 转换使能位</b> 0: 禁止 1: 使能

ADC 控制寄存器 1 (ADC\_CON1)

偏移地址: 08<sub>H</sub>

复位值: 00000000\_00000000\_10010000\_00000000<sub>B</sub>

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

—	bit31-17	—	—
HSEN	bit16	R/W	<b>ADC 转换速度控制位，必须软件设置为 1</b>
ST<1:0>	bit15-14	R/W	<b>ADC 采样时间选择位（硬件控制有效）</b> 00: 大约 2 个 Tadclk 01: 大约 4 个 Tadclk 10: 大约 8 个 Tadclk 11: 大约 16 个 Tadclk
SMPON	bit13	R/W	<b>ADC 采样软件控制位</b> 0: 关闭 ADC 采样 1: 启动 ADC 采样
SMPS	bit12	R/W	<b>ADC 采样模式选择位</b> 0: 软件控制 1: 硬件控制
—	bit11-10	—	—
VREFP<1:0>	bit9-8	R/W	<b>ADC 正向参考电压选择位</b> 00: 选择 ADC 的工作电压，即 3.0V LDO 电压 01: 选择内部参考电压 VREF 2.048V， AVREFP 端口，复用作普通 I/O 端口 10: 选择内部参考电压 VREF 2.048V， AVREFP 端口，输出内部参考电压 VREF 11: 外部参考电压 AVREFP，该电压不能高于 ADC 的工作电压
—	bit7	—	—
FM	bit6	R/W	<b>ADC 转换数据格式选择位</b> 0: 高位对齐（转换结果存放在 DR<15:4>） 1: 低位对齐（转换结果存放在 DR<11:0>）
—	bit5-4	—	—
CLKS	bit3	R/W	<b>ADC 时钟源选择位</b> 0: PCLK 1: ADCCLK（32KHz）
CLKDIV<2:0>	bit2-0	R/W	<b>ADC 时钟源预分频选择位</b> 000: 1:1 001: 1:2 010: 1:4 011: 1:8

			100 : 1:16 101 : 1:32 110 : 1:64 111 : 1:256
--	--	--	---

ADC 通道选择寄存器 (ADC_CHS)															
偏移地址: 0C <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CHS<3:0>			

—	bit31-4	—	保留位 禁止用户写操作, 否则可能会导致 ADC 工作异常
CHS<3:0>	bit3-0	R/W	<b>ADC 模拟通道选择位</b> 0000: 通道 0 (AIN0) 0001: 通道 1 (AIN1) 0010: 通道 2 (AIN2) 0011: 通道 3 (AIN3) 0100: 通道 4 (AIN4) 0101: 通道 5 (AIN5) 0110: 通道 6 (AIN6) 0111: 通道 7 (AIN7) 1000: 通道 8 (AIN8) 1001: 通道 9 (AIN9) 1010: 通道 10 (AIN10) 其它: 保留

ADC 中断使能寄存器 (ADC_IE)															
偏移地址: 10 <sub>H</sub>															
复位值: 00000000_00000000_00000000_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															IE

—	bit31-1	—	—
IE	bit0	R/W	<b>ADC 中断使能位</b> 0: 禁止 1: 使能

**ADC 中断标志寄存器 (ADC\_IF)**

偏移地址: 14<sub>H</sub>

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

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

—	bit31-1	—	—
IF	bit0	RW	<b>ADC 中断标志位</b> 0: 正在进行转换 1: ADC 转换完成 (由硬件置 1, 软件清除) 软件写 1 清除标志位, 写 0 无效

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

**ADC 缓冲控制寄存器 (ADC\_BUF)**

偏移地址: 20<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_00001001<sub>B</sub>

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

—	bit31-4	—	—
LP<1:0>	bit3-2	RW	<b>ADC core 功耗选择位, 必须软件设置为 00</b> 00: 215 uA 其他: 保留
BUF_EN<1:0>	bit1-0	RW	<b>ADC 输入 buffer 选择位</b> 00: bypass, buffer 功耗 0 uA 01: 选通低功耗 buffer, 功耗 34 uA 10: 选通高功耗 buffer, 功耗 95 uA 11: 保留 推荐在芯片使用时配置为 00 (bypass buffer)

## 5.7 实时时钟 (RTC)

### 5.7.1 概述

- ◇ 仅 POR 上电复位有效，支持程序写保护，有效避免系统干扰对时钟造成的影响
- ◇ 采用外部 32.768KHz 晶体振荡器作为 RTC 精确计时的时钟源；如果应用系统对 RTC 计时精度要求不高，还可选用内部 LRC 作为时钟源
- ◇ 时钟调校最高精度  $\pm 0.0254$  ppm
- ◇ 时间计数（实现小时、分钟和秒）和日历计数（实现年、月、日和星期），BCD 格式
- ◇ 支持 5 个可编程定时中断
- ◇ 支持 2 个可编程日历闹钟中断
- ◇ 支持振荡器停振检测中断
- ◇ 提供一路可配置时钟输出
- ◇ 自动闰年识别，有效期至 2099 年
- ◇ 12 小时和 24 小时模式设置可选
- ◇ 低功耗设计：工作电压为 3.6V 时模块工作电流典型值为 1.5 $\mu$ A (最大值为 3 $\mu$ A)

### 5.7.2 RTC写保护

为避免程序的异常运行对 RTC 模块的误操作，RTC 写保护寄存器 RTC\_WP 用于阻止程序对 RTC 模块其它寄存器的误写（不包括 RTC\_WP 寄存器本身）。

RTC\_WP 寄存器为虚拟寄存器。要对 RTC 模块其它寄存器进行写操作时，需先对 RTC\_WP 寄存器写 0x55AAAA55，之后可对 RTC 模块其它寄存器进行连续写操作。对 RTC\_WP 寄存器写入全零时重新进入写保护状态，写保护状态下对其他寄存器进行的写操作将被忽略，对 RTC\_WP 寄存器写入其他值时无效。

可通过读 RTC\_WP 寄存器确认 RTC 模块是否处于写保护状态，读出值为 0x55AAAA55，表示当前可对 RTC 模块其它寄存器进行写操作；读出值为 0x00000000 表示 RTC 模块处于写保护状态。RTC\_WP 寄存器无其它读出值。

### 5.7.3 时间日期设置

由于 APB 总线时钟与 RTC 时间计数器时钟异步，因此 RTC 时间计数器不能直接读写，只能通过缓冲器进行读写。RTC 时、分、秒寄存器和 RTC 年、月、日、周寄存器用于 RTC 时间计数器的写入和读取。这些寄存器只是读写缓冲器，而不是时间计数器本身。读时间计数器的操作步骤如下：

- 1) 配置 RTC\_CON 寄存器中的读写选择位 TMWR=0，选择时间计数器读操作；
- 2) 配置 RTC\_CON 寄存器中的读写触发位 TMUP=1，触发读操作；
- 3) 读操作完成后，时间计数器的值被分别读出到时、分、秒寄存器 RTC\_HOUR/RTC\_MIN/RTC\_SEC 和年、月、日、周寄存器 RTC\_YEAR/RTC\_MON/RTC\_DAY/RTC\_WEEK，并且 RTC\_CON 寄存器中的 TMUP 位自动清零。

写时间计数器的操作步骤如下（复位或上一次时间计数器读/写操作完成后）：

- 1) 配置 12/24 小时模式选择位 HSWI;
- 2) 写入设置值到时、分、秒寄存器 RTC\_HOUR/RTC\_MIN/RTC\_SEC 和/或年、月、日、周寄存器 RTC\_YEAR/RTC\_MON/RTC\_DAY/RTC\_WEEK;
- 3) 配置 RTC\_CON 寄存器中的读写选择位 TMWR=1, 选择时间计数器写操作;
- 4) 配置 RTC\_CON 寄存器中的读写触发位 TMUP=1, 触发写操作;
- 5) 写操作完成后, 时、分、秒寄存器 RTC\_HOUR/RTC\_MIN/RTC\_SEC 和年、月、日、周寄存器 RTC\_YEAR/RTC\_MON/RTC\_DAY/RTC\_WEEK 更新过的设置值被写入时间计数器, 并且 TMUP 位自动清零。

注 1: 写时间计数器操作是将从复位或上一次读操作完成后, 更新过的时、分、秒寄存器 RTC\_HOUR/RTC\_MIN/RTC\_SEC 和/或年、月、日、周寄存器 RTC\_YEAR/RTC\_MON/RTC\_DAY/RTC\_WEEK 的内容写入相应时间计数器, 而对应于未更新字段的时间计数器不受写入影响。

注 2: 当 TMUP 位为 1 时, 可通过将 TMWR 清零提前中止当前的写操作。被中止的写操作结果不确定。

时间和日期寄存器数据格式采用 BCD 编码。秒计数范围从 00 到 59, 进位到分钟后从 59 变为 00。分钟计数范围从 00 到 59, 进位到小时后从 59 变为 00。小时计数范围根据控制位 HSWI 的设置选择 12/24 小时模式, 进位后从 PM11 到 AM12 或 AM11 到 PM12, 或 23 到 00。

星期计数器为 3 位计数器, 数值为 0-6, 初始值可配置。

日计数按照每月最后一天加 1 进位到下月, 日计数范围按月分为:

- 1) 一、三、五、七、八、十、十二月从 1 到 31;
- 2) 四、六、九、十一月从 1 到 30;
- 3) 二月 (普通年份) 从 1 到 28; 二月 (闰年) 从 1 到 29;
- 4) 月计数范围从 1 到 12, 进位到年后从 12 变为 1;
- 5) 年计数范围从 00 到 99 (00, 04, 08, ..., 92, 96 为闰年), 99 后不再进位到 00。

12/24 小时模式对照表如下:

24 小时模式	12 小时模式	24 小时模式	12 小时模式
00	12 (AM12)	12	32 (PM12)
01	01 (AM1)	13	21 (PM1)
02	02 (AM2)	14	22 (PM2)
03	03 (AM3)	15	23 (PM3)
04	04 (AM4)	16	24 (PM4)
05	05 (AM5)	17	25 (PM5)
06	06 (AM6)	18	26 (PM6)
07	07 (AM7)	19	27 (PM7)
08	08 (AM8)	20	28 (PM8)
09	09 (AM9)	21	29 (PM9)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

表 5-2 12/24 小时模式对照表

#### 5.7.4 RTC中断源

RTC 模块共有 8 个中断源, 即:

- ◇ 周闹钟中断 WAFG
- ◇ 日闹钟中断 DAFG
- ◇ 5个周期中断月、日、时、分、秒中断
- ◇ 振荡器停振检测中断

每个中断源都有独立的使能位，使能位影响该中断是否产生 IRQ 中断请求，而不影响中断功能。即关闭相应中断使能，标志位仍可用于相应功能查询。当有多个中断使能时，各中断经过“或”逻辑产生 IRQ 中断请求。即任何一个被使能的中断产生中断事件时，均产生 IRQ 中断请求，且只有将所有的产生中断事件的中断标志清零后，IRQ 中断请求才解除。

### 5.7.5 特殊功能寄存器

#### RTC 写保护寄存器 (RTC\_WP)

偏移地址: 40<sub>H</sub>

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

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

WP<31:0>	bit31-0	RW	0x0000_0000: RTC 模块写保护使能 0x55AA_AA55: RTC 模块写保护关闭
----------	---------	----	--

注: RTC\_WP 寄存器保护的寄存器为: RTC\_CON, RTC\_CALC, RTC\_CALF, RTC\_WA, RTC\_DA, RTC\_SEC, RTC\_MIN, RTC\_HOUR, RTC\_WEEK, RTC\_DAY, RTC\_MON, RTC\_YEAR, RTC\_IE。

#### RTC 控制寄存器 (RTC\_CON)

偏移地址: 00<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_10000000<sub>B</sub>

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

—	bit31-8	—	—
PON	bit7	RW	<b>RTC 上电复位标志位</b> (软件只能将该位清 0) 0: 实时时钟正常工作 1: 监测到 RTC 上电复位 (必须软件清 0 后, RTC 才能开始工作)
XST	bit6	R	<b>振荡器停振监测标志位</b> 0: 振荡器持续工作正常

—	bit5-4	—	1: 振荡器有停振现象发生
CLKS	bit3	R/W	<b>RTC 时钟源选择位</b> （必须在写时间数据前设置） 0: 32.768 KHz 振荡器时钟源 1: LRC 时钟源
HSWI	bit2	R/W	<b>12/24 小时模式选择位</b> （必须在写时间数据前设置） 0: 12 小时模式 1: 24 小时模式
TMWR	bit1	R/W	<b>时间计数器读写选择位</b> 0: 时间计数器读出操作 1: 时间计数器写入操作
TMUP	bit0	R/W	<b>时间计数器读写触发位</b> （程序只能写 1，读写完成后自动清零） 0: 时间计数器读写操作已完成 1: 时间计数器正在进行读写操作

注 1: RTC 模块上电复位后一直处于复位状态，只有将 RTC\_CON 寄存器的 PON 位清零后，RTC 才进入工作状态。  
注 2: 为保证精度，建议 CLKS 值设置为 00，即选用外部 32.768KHz 晶振；对于低精度要求的运用，可选用 LRC 时钟源。

**RTC 调校控制寄存器 (RTC\_CALC)**

偏移地址: 04<sub>H</sub>

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

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

—	bit31-9	—	—
CLKC	Bit8	R/W	<b>RTC 输出脉冲端口使能位</b> 0: 禁止 1: 使能（端口需复用为 RTCO，输出脉冲）
COCR<3:0>	Bit7-4	R/W	<b>RTC 输出脉冲频率选择位</b> 0000: 32KHz 0001: 1024Hz 0010: 32Hz 0011: 1Hz 0100: 校正后 1Hz 时钟输出 0101: 精确 1Hz 时钟输出 其他: 保留
—	Bit3	—	—
DEV<2:0>	Bit2-0	R/W	<b>调校模式选择位</b> 000: 每 10 秒调校一次



			001: 每 20 秒调校一次 010: 每 1 分钟调校一次 011: 每 2 分钟调校一次 100: 每 5 分钟调校一次 101: 每 10 分钟调校一次 110: 每 20 分钟调校一次 111: 每秒调校 (模拟调校时用)
--	--	--	--

**RTC 调校值寄存器 (RTC\_CALF)**

偏移地址: 08<sub>H</sub>

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

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

—	bit31-16	—	—
CALF<15:0>	Bit15-0	R/W	RTC 调校值

**RTC 周闹钟寄存器 (RTC\_WA)**

偏移地址: 10<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-23	—	—
WW<6:0>	bit22-16	R/W	周闹钟位 WW<6>: 周六闹钟位 WW<5>: 周五闹钟位 WW<4>: 周四闹钟位 WW<3>: 周三闹钟位 WW<2>: 周二闹钟位 WW<1>: 周一闹钟位 WW<0>: 周日闹钟位
—	bit15-14	—	—
WH<5:0>	bit13-8	R/W	小时(BCD 码)位 WH<5>: 24 小时模式: 20 小时位 12 小时模式: 1 代表 pm; 0 代表 am WH<4>: 10 小时位 WH<3>: 8 小时位

			WH<2>: 4 小时位 WH<1>: 2 小时位 WH<0>: 1 小时位
—	bit7	—	—
WM<6:0>	bit6-0	R/W	<b>分钟(BCD 码)位</b> WM<6>: 40 分钟位 WM<5>: 20 分钟位 WM<4>: 10 分钟位 WM<3>: 8 分钟位 WM<2>: 4 分钟位 WM<1>: 2 分钟位 WM<0>: 1 分钟位

**RTC 日闹钟寄存器 (RTC\_DA)**

偏移地址: 14<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-14	—	—
DH<5:0>	bit13-8	R/W	<b>小时(BCD 码)位</b> DH<5>: 24 小时模式: 20 小时位 12 小时模式: 1 代表 pm; 0 代表 am DH<4>: 10 小时位 DH<3>: 8 小时位 DH<2>: 4 小时位 DH<1>: 2 小时位 DH<0>: 1 小时位
—	bit7	—	—
DM<6:0>	bit6-0	R/W	<b>分钟(BCD 码)位</b> DM<6>: 40 分钟位 DM<5>: 20 分钟位 DM<4>: 10 分钟位 DM<3>: 8 分钟位 DM<2>: 4 分钟位 DM<1>: 2 分钟位 DM<0>: 1 分钟位

**RTC 秒寄存器 (RTC\_SEC)**

偏移地址: 18<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									S40	S20	S10	S08	S04	S02	S01

—	bit31-7	—	—
S40	bit6	R/W	40 秒位
S20	bit5	R/W	20 秒位
S10	bit4	R/W	10 秒位
S08	bit3	R/W	8 秒位
S04	bit2	R/W	4 秒位
S02	bit1	R/W	2 秒位
S01	bit0	R/W	1 秒位

**RTC 分钟寄存器 (RTC\_MIN)**

偏移地址: 1C<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									M40	M20	M10	M08	M04	M02	M01

—	bit31-7	—	—
M40	bit6	R/W	40 分钟位
M20	bit5	R/W	20 分钟位
M10	bit4	R/W	10 分钟位
M08	bit3	R/W	8 分钟位
M04	bit2	R/W	4 分钟位
M02	bit1	R/W	2 分钟位
M01	bit0	R/W	1 分钟位

**RTC 小时寄存器 (RTC\_HOUR)**

偏移地址: 20<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

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

—	bit31-6	—	—
HOUR<5:0>	bit5-0	R/W	<p>小时(BCD 码)位</p> <p>HOUR&lt;5&gt;: 24 小时模式: 20 小时位 12 小时模式: 1 代表 pm; 0 代表 am</p> <p>HOUR&lt;4&gt;: 10 小时位</p> <p>HOUR&lt;3&gt;: 8 小时位</p> <p>HOUR&lt;2&gt;: 4 小时位</p> <p>HOUR&lt;1&gt;: 2 小时位</p> <p>HOUR&lt;0&gt;: 1 小时位</p>

**RTC 周寄存器 (RTC\_WEEK)**

偏移地址: 24<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxxxx\_xxxxxxxxxx\_xxxxxxxxxx<sub>B</sub>

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

—	bit31-3	—	—
WW<2:0>	bit2-0	R/W	<p>000: 周日</p> <p>001: 周一</p> <p>... ..</p> <p>110: 周六</p> <p>111: 保留</p>

**RTC 日寄存器 (RTC\_DAY)**

偏移地址: 28<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxxxx\_xxxxxxxxxx\_xxxxxxxxxx<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留											D20	D10	D08	D04	D02	D01

—	bit31-6	—	—
D20	bit5	R/W	20 日位
D10	bit4	R/W	10 日位
D08	bit3	R/W	8 日位
D04	bit2	R/W	4 日位
D02	bit1	R/W	2 日位
D01	bit0	R/W	1 日位

**RTC 月寄存器 (RTC\_MON)**

偏移地址: 2C<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											MN10	MN08	MN04	MN02	MN01

—	bit31-5	—	—
MN10	bit4	R/W	10 月位
MN08	bit3	R/W	8 月位
MN04	bit2	R/W	4 月位
MN02	bit1	R/W	2 月位
MN01	bit0	R/W	1 月位

**RTC 年寄存器 (RTC\_YEAR)**

偏移地址: 30<sub>H</sub>

复位值: xxxxxxxx\_xxxxxxxx\_xxxxxxxx\_xxxxxxxx<sub>B</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								Y80	Y40	Y20	Y10	Y08	Y04	Y02	Y01

—	bit31-8	—	—
Y80	bit7	R/W	80 年位
Y40	bit6	R/W	40 年位
Y20	bit5	R/W	20 年位
Y10	bit4	R/W	10 年位
Y08	bit3	R/W	8 年位
Y04	bit2	R/W	4 年位
Y02	bit1	R/W	2 年位
Y01	bit0	R/W	1 年位

**RTC 中断使能寄存器 (RTC\_IE)**

偏移地址: 50<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_10000000<sub>B</sub>

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

—	bit31-13	—	—
OSCDIE	bit12	R/W	振荡器停振中断使能位 0: 禁止 1: 使能
—	bit11-10	—	—
WALE	bit9	R/W	周闹钟使能位 0: 禁止 1: 使能
DALE	bit8	R/W	日闹钟使能位 0: 禁止 1: 使能
—	bit7-5	—	—
MONIE	bit4	R/W	月中断使能位 0: 禁止 1: 使能
DAYIE	bit3	R/W	日中断使能位 0: 禁止 1: 使能
HORIE	bit2	R/W	时中断使能位 0: 禁止 1: 使能
MINIE	bit1	R/W	分中断使能位 0: 禁止 1: 使能
SCDIE	bit0	R/W	秒中断使能位 0: 禁止 1: 使能

RTC 中断标志寄存器 (RTC\_IF)

偏移地址: 54<sub>H</sub>

复位值: 00000000\_00000000\_00000000\_10000000<sub>B</sub>

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

—	bit31-13	—	—
OSCDIF	bit12	R/W	振荡器停振中断标志位 0: 无振荡器停振中断 (或标志已清除) 1: 产生振荡器停振中断 软件写 1 清除标志位, 写 0 无效
—	bit11-10	—	—
WAFG	bit9	R/W	周闹钟标志位

			0: 闹钟事件不匹配 1: 闹钟事件匹配 软件写 1 清除标志位, 写 0 无效
DAFG	bit8	R/W	<b>日闹钟标志位</b> 0: 闹钟事件不匹配 1: 闹钟事件匹配 软件写 1 清除标志位, 写 0 无效
—	bit7-5	—	—
MONIF	bit4	R/W	<b>月中断标志位</b> (中断周期为每月第一天 00 小时 00 分钟 00 秒) 0: 未产生月中断 1: 产生月中断 软件写 1 清除标志位, 写 0 无效
DAYIF	bit3	R/W	<b>日中断标志位</b> (中断周期为每天) 0: 未产生日中断 1: 产生日中断 软件写 1 清除标志位, 写 0 无效
HORIF	bit2	R/W	<b>时中断标志位</b> (中断周期为每小时) 0: 未产生小时中断 1: 产生小时中断 软件写 1 清除标志位, 写 0 无效
MINIF	bit1	R/W	<b>分中断标志位</b> (中断周期为每分钟) 0: 未产生分钟中断 1: 产生分钟中断 软件写 1 清除标志位, 写 0 无效
SCDIF	bit0	R/W	<b>秒中断标志位</b> (中断周期为每秒) 0: 未产生秒中断 1: 产生秒中断 软件写 1 清除标志位, 写 0 无效

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

## 5.8 液晶显示控制器 (LCDC)

### 5.8.1 概述

- ◇ 支持 8COM x 56SEG
- ◇ 支持灰度调节功能
- ◇ 支持显示闪烁功能，闪烁频率可调

### 5.8.2 结构框图

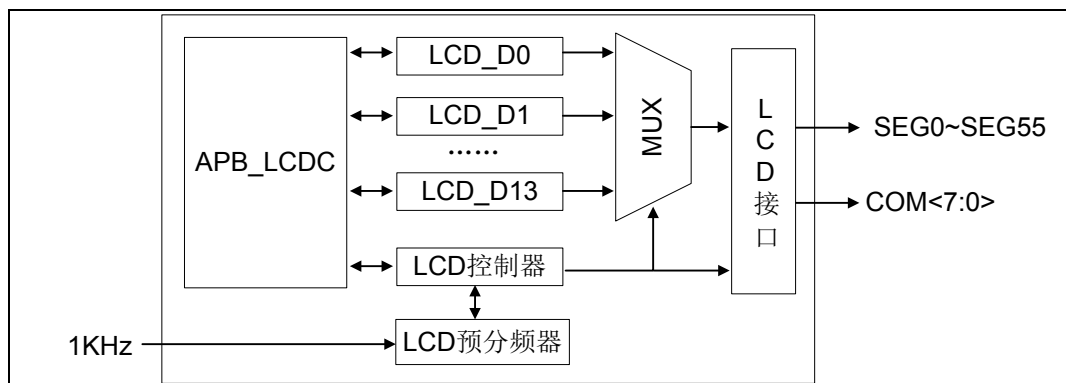


图 5-47 LCDC 驱动模块框图

### 5.8.3 LCDC基本设置

配置 LCDC 驱动模块的操作步骤如下：

- 1) 设置 LCDC 工作时钟预分频比。
- 2) 设置 LCDC 驱动端口：配置 I/O 端口复用选择寄存器 GPIO\_PAFUN/GPIO\_PBFUN/GPIO\_PCFUN，将端口复用为 LCDC 驱动端口；配置 COMS<2:0>和 SEG<55:0>使能端口的公共端和段功能，及选择偏置模式。
- 3) 初始化 LCDC 像素数据寄存器 LCD\_D0~LCD\_D13。
- 4) 使能 LCDC 驱动模块。

### 5.8.4 LCDC偏置选择

LCDC 支持内部偏置电压选择。

内部偏置类型如下，VBIAS 为 VLCD 分压后提供，分压由 BVS 寄存器配置：

- ◇ 1/3 偏置（4 种电压等级：VSS、1/3 VBIAS、2/3 VBIAS 和 VBIAS）
- ◇ 1/4 偏置（5 种电压等级：VSS、1/4 VBIAS、2/4 VBIAS、3/4 VBIAS 和 VBIAS）

用户只需设置所需的偏置类型，无需进行各个偏置电压的设置，偏置电压由内部电路自动产生。

### 5.8.5 LCDC像素对照表

LCDC COM 像素寄存器的各个数值位，与公共端-段的像素点对应关系如下表所示：



COM7	COM8	COM5	COM4	COM3	COM2	COM1	COM0	
LCD_D13[31]	LCD_D13[30]	LCD_D13[29]	LCD_D13[28]	LCD_D13[27]	LCD_D13[26]	LCD_D13[25]	LCD_D13[24]	SEG55
LCD_D13[23]	LCD_D13[22]	LCD_D13[21]	LCD_D13[20]	LCD_D13[19]	LCD_D13[18]	LCD_D13[17]	LCD_D13[16]	SEG54
LCD_D13[15]	LCD_D13[14]	LCD_D13[13]	LCD_D13[12]	LCD_D13[11]	LCD_D13[10]	LCD_D13[9]	LCD_D13[8]	SEG53
LCD_D13[7]	LCD_D13[6]	LCD_D13[5]	LCD_D13[4]	LCD_D13[3]	LCD_D13[2]	LCD_D13[1]	LCD_D13[0]	SEG52
LCD_D12[31]	LCD_D12[30]	LCD_D12[29]	LCD_D12[28]	LCD_D12[27]	LCD_D12[26]	LCD_D12[25]	LCD_D12[24]	SEG51
LCD_D12[23]	LCD_D12[22]	LCD_D12[21]	LCD_D12[20]	LCD_D12[19]	LCD_D12[18]	LCD_D12[17]	LCD_D12[16]	SEG50
LCD_D12[15]	LCD_D12[14]	LCD_D12[13]	LCD_D12[12]	LCD_D12[11]	LCD_D12[10]	LCD_D12[9]	LCD_D12[8]	SEG49
LCD_D12[7]	LCD_D12[6]	LCD_D12[5]	LCD_D12[4]	LCD_D12[3]	LCD_D12[2]	LCD_D12[1]	LCD_D12[0]	SEG48
LCD_D11[31]	LCD_D11[30]	LCD_D11[29]	LCD_D11[28]	LCD_D11[27]	LCD_D11[26]	LCD_D11[25]	LCD_D11[24]	SEG47
LCD_D11[23]	LCD_D11[22]	LCD_D11[21]	LCD_D11[20]	LCD_D11[19]	LCD_D11[18]	LCD_D11[17]	LCD_D11[16]	SEG46
LCD_D11[15]	LCD_D11[14]	LCD_D11[13]	LCD_D11[12]	LCD_D11[11]	LCD_D11[10]	LCD_D11[9]	LCD_D11[8]	SEG45
LCD_D11[7]	LCD_D11[6]	LCD_D11[5]	LCD_D11[4]	LCD_D11[3]	LCD_D11[2]	LCD_D11[1]	LCD_D11[0]	SEG44
LCD_D10[31]	LCD_D10[30]	LCD_D10[29]	LCD_D10[28]	LCD_D10[27]	LCD_D10[26]	LCD_D10[25]	LCD_D10[24]	SEG43
LCD_D10[23]	LCD_D10[22]	LCD_D10[21]	LCD_D10[20]	LCD_D10[19]	LCD_D10[18]	LCD_D10[17]	LCD_D10[16]	SEG42
LCD_D10[15]	LCD_D10[14]	LCD_D10[13]	LCD_D10[12]	LCD_D10[11]	LCD_D10[10]	LCD_D10[9]	LCD_D10[8]	SEG41
LCD_D10[7]	LCD_D10[6]	LCD_D10[5]	LCD_D10[4]	LCD_D10[3]	LCD_D10[2]	LCD_D10[1]	LCD_D10[0]	SEG40
LCD_D9[31]	LCD_D9[30]	LCD_D9[29]	LCD_D9[28]	LCD_D9[27]	LCD_D9[26]	LCD_D9[25]	LCD_D9[24]	SEG39
LCD_D9[23]	LCD_D9[22]	LCD_D9[21]	LCD_D9[20]	LCD_D9[19]	LCD_D9[18]	LCD_D9[17]	LCD_D9[16]	SEG38
LCD_D9[15]	LCD_D9[14]	LCD_D9[13]	LCD_D9[12]	LCD_D9[11]	LCD_D9[10]	LCD_D9[9]	LCD_D9[8]	SEG37
LCD_D9[7]	LCD_D9[6]	LCD_D9[5]	LCD_D9[4]	LCD_D9[3]	LCD_D9[2]	LCD_D9[1]	LCD_D9[0]	SEG36
LCD_D8[31]	LCD_D8[30]	LCD_D8[29]	LCD_D8[28]	LCD_D8[27]	LCD_D8[26]	LCD_D8[25]	LCD_D8[24]	SEG35
LCD_D8[23]	LCD_D8[22]	LCD_D8[21]	LCD_D8[20]	LCD_D8[19]	LCD_D8[18]	LCD_D8[17]	LCD_D8[16]	SEG34
LCD_D8[15]	LCD_D8[14]	LCD_D8[13]	LCD_D8[12]	LCD_D8[11]	LCD_D8[10]	LCD_D8[9]	LCD_D8[8]	SEG33
LCD_D8[7]	LCD_D8[6]	LCD_D8[5]	LCD_D8[4]	LCD_D8[3]	LCD_D8[2]	LCD_D8[1]	LCD_D8[0]	SEG32
LCD_D7[31]	LCD_D7[30]	LCD_D7[29]	LCD_D7[28]	LCD_D7[27]	LCD_D7[26]	LCD_D7[25]	LCD_D7[24]	SEG31
LCD_D7[23]	LCD_D7[22]	LCD_D7[21]	LCD_D7[20]	LCD_D7[19]	LCD_D7[18]	LCD_D7[17]	LCD_D7[16]	SEG30
LCD_D7[15]	LCD_D7[14]	LCD_D7[13]	LCD_D7[12]	LCD_D7[11]	LCD_D7[10]	LCD_D7[9]	LCD_D7[8]	SEG29
LCD_D7[7]	LCD_D7[6]	LCD_D7[5]	LCD_D7[4]	LCD_D7[3]	LCD_D7[2]	LCD_D7[1]	LCD_D7[0]	SEG28
LCD_D6[31]	LCD_D6[30]	LCD_D6[29]	LCD_D6[28]	LCD_D6[27]	LCD_D6[26]	LCD_D6[25]	LCD_D6[24]	SEG27
LCD_D6[23]	LCD_D6[22]	LCD_D6[21]	LCD_D6[20]	LCD_D6[19]	LCD_D6[18]	LCD_D6[17]	LCD_D6[16]	SEG26
LCD_D6[15]	LCD_D6[14]	LCD_D6[13]	LCD_D6[12]	LCD_D6[11]	LCD_D6[10]	LCD_D6[9]	LCD_D6[8]	SEG25
LCD_D6[7]	LCD_D6[6]	LCD_D6[5]	LCD_D6[4]	LCD_D6[3]	LCD_D6[2]	LCD_D6[1]	LCD_D6[0]	SEG24
LCD_D5[31]	LCD_D5[30]	LCD_D5[29]	LCD_D5[28]	LCD_D5[27]	LCD_D5[26]	LCD_D5[25]	LCD_D5[24]	SEG23
LCD_D5[23]	LCD_D5[22]	LCD_D5[21]	LCD_D5[20]	LCD_D5[19]	LCD_D5[18]	LCD_D5[17]	LCD_D5[16]	SEG22
LCD_D5[15]	LCD_D5[14]	LCD_D5[13]	LCD_D5[12]	LCD_D5[11]	LCD_D5[10]	LCD_D5[9]	LCD_D5[8]	SEG21
LCD_D5[7]	LCD_D5[6]	LCD_D5[5]	LCD_D5[4]	LCD_D5[3]	LCD_D5[2]	LCD_D5[1]	LCD_D5[0]	SEG20
LCD_D4[31]	LCD_D4[30]	LCD_D4[29]	LCD_D4[28]	LCD_D4[27]	LCD_D4[26]	LCD_D4[25]	LCD_D4[24]	SEG19
LCD_D4[23]	LCD_D4[22]	LCD_D4[21]	LCD_D4[20]	LCD_D4[19]	LCD_D4[18]	LCD_D4[17]	LCD_D4[16]	SEG18
LCD_D4[15]	LCD_D4[14]	LCD_D4[13]	LCD_D4[12]	LCD_D4[11]	LCD_D4[10]	LCD_D4[9]	LCD_D4[8]	SEG17
LCD_D4[7]	LCD_D4[6]	LCD_D4[5]	LCD_D4[4]	LCD_D4[3]	LCD_D4[2]	LCD_D4[1]	LCD_D4[0]	SEG16
LCD_D3[31]	LCD_D3[30]	LCD_D3[29]	LCD_D3[28]	LCD_D3[27]	LCD_D3[26]	LCD_D3[25]	LCD_D3[24]	SEG15
LCD_D3[23]	LCD_D3[22]	LCD_D3[21]	LCD_D3[20]	LCD_D3[19]	LCD_D3[18]	LCD_D3[17]	LCD_D3[16]	SEG14

COM7	COM8	COM5	COM4	COM3	COM2	COM1	COM0	
LCD_D3[15]	LCD_D3[14]	LCD_D3[13]	LCD_D3[12]	LCD_D3[11]	LCD_D3[10]	LCD_D3[9]	LCD_D3[8]	SEG13
LCD_D3[7]	LCD_D3[6]	LCD_D3[5]	LCD_D3[4]	LCD_D3[3]	LCD_D3[2]	LCD_D3[1]	LCD_D3[0]	SEG12
LCD_D2[31]	LCD_D2[30]	LCD_D2[29]	LCD_D2[28]	LCD_D2[27]	LCD_D2[26]	LCD_D2[25]	LCD_D2[24]	SEG11
LCD_D2[23]	LCD_D2[22]	LCD_D2[21]	LCD_D2[20]	LCD_D2[19]	LCD_D2[18]	LCD_D2[17]	LCD_D2[16]	SEG10
LCD_D2[15]	LCD_D2[14]	LCD_D2[13]	LCD_D2[12]	LCD_D2[11]	LCD_D2[10]	LCD_D2[9]	LCD_D2[8]	SEG9
LCD_D2[7]	LCD_D2[6]	LCD_D2[5]	LCD_D2[4]	LCD_D2[3]	LCD_D2[2]	LCD_D2[1]	LCD_D2[0]	SEG8
LCD_D1[31]	LCD_D1[30]	LCD_D1[29]	LCD_D1[28]	LCD_D1[27]	LCD_D1[26]	LCD_D1[25]	LCD_D1[24]	SEG7
LCD_D1[23]	LCD_D1[22]	LCD_D1[21]	LCD_D1[20]	LCD_D1[19]	LCD_D1[18]	LCD_D1[17]	LCD_D1[16]	SEG6
LCD_D1[15]	LCD_D1[14]	LCD_D1[13]	LCD_D1[12]	LCD_D1[11]	LCD_D1[10]	LCD_D1[9]	LCD_D1[8]	SEG5
LCD_D1[7]	LCD_D1[6]	LCD_D1[5]	LCD_D1[4]	LCD_D1[3]	LCD_D1[2]	LCD_D1[1]	LCD_D1[0]	SEG4
LCD_D0[31]	LCD_D0[30]	LCD_D0[29]	LCD_D0[28]	LCD_D0[27]	LCD_D0[26]	LCD_D0[25]	LCD_D0[24]	SEG3
LCD_D0[23]	LCD_D0[22]	LCD_D0[21]	LCD_D0[20]	LCD_D0[19]	LCD_D0[18]	LCD_D0[17]	LCD_D0[16]	SEG2
LCD_D0[15]	LCD_D0[14]	LCD_D0[13]	LCD_D0[12]	LCD_D0[11]	LCD_D0[10]	LCD_D0[9]	LCD_D0[8]	SEG1
LCD_D0[7]	LCD_D0[6]	LCD_D0[5]	LCD_D0[4]	LCD_D0[3]	LCD_D0[2]	LCD_D0[1]	LCD_D0[0]	SEG0

表 5-3 LCDC 像素对照列表

### 5.8.6 LCDC工作时钟源

LCDC 驱动模块工作时钟源为芯片内部 LRC 时钟或外部 LOSC 时钟（通过寄存器 SCU\_SCLKEN1 中 LCD32K\_SEL 进行选择）4 分频后的时钟，约为 8KHz。

支持工作时钟源预分频，配置 LCD\_CON0 寄存器的位 PRS<3:0>，可设置预分频比。

### 5.8.7 LCD显示帧频率

LCD 的显示帧频率，与 COM 端口复用模式，工作时钟源频率及其预分频比有关，其中 COM 端口复用模式，由 LCD\_CON0 寄存器中的 COMS<2:0>进行选择，描述如下：

COMS<2:0> = 001: 帧频率 = 时钟源频率 / (1 × 4 × (PRS<5:0>+1))

COMS<2:0> = 010: 帧频率 = 时钟源频率 / (1 × 6 × (PRS<5:0>+1))

COMS<2:0> = 011: 帧频率 = 时钟源频率 / (1 × 8 × (PRS<5:0>+1))

COMS<2:0> = 10x: 帧频率 = 时钟源频率 / (2 × 2 × (PRS<5:0>+1))

COMS<2:0> = 11x: 帧频率 = 时钟源频率 / (1 × 3 × (PRS<5:0>+1))

### 5.8.8 LCDC像素寄存器的刷新操作

可通过两种方式刷新 LCDC 像素寄存器 LCD\_D0~LCD\_D13。一种是在任意时刻，对像素寄存器进行写操作，更新像素，但可能会出现显示不完整帧现象。另一种是配置像素刷新请求位 LCD\_CON1 寄存器的 RFREQ，请求 LCDC 像素寄存器更新，LCDC 驱动模块在完成当前帧显示之后，置起像素刷新 LCD\_IF 寄存器的中断标志 RFIF，可对像素寄存器 LCD\_D0~LCD\_D13 进行写操作，更新像素数据，写操作完成后需软件将 RFREQ 和 RFIF 清 0。

### 5.8.9 LCD闪烁功能

配置 LCD\_CON0 寄存器的 FLIK 位，可使能 LCD 显示闪烁功能。配置闪烁时间寄存器 LCD\_FLKT，可设置 LCD 闪烁时的点亮时间和熄灭时间。在使能闪烁功能之前，应先设

置 LCD\_FLKT 寄存器。

### 5.8.10 LCDC中断

LCDC 模块支持 2 个中断源：像素寄存器刷新中断 RFIF 和 LCDC 模块关闭中断 OFFIF。每个中断源都有独立的使能位，使能位控制该中断是否产生 IRQ 中断请求，而不影响中断标志位的产生，即禁止中断使能时，标志位仍会在中断产生时置起。

### 5.8.11 特殊功能寄存器

LCD 控制寄存器 0 (LCD_CON0)															
偏移地址: 00H															
复位值: 00000000_00000000_00001100_00000000 <sub>B</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				保留		BIAS_SEL<1:0>		保留		PRS<5:0>					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	BVS<2:0>		RS<1:0>		BIAS<1:0>		保留	WFS	FLIK	保留		COMS<2:0>			

—	bit31-26	—	—
BIAS_SEL<1:0>	bit25-24	R/W	<b>LCDC 内部偏置选择位</b> 00: 偏置使能位设置为“0” 01: 偏置使能位设置为“1” 1x: 保留
—	bit23-22	—	—
PRS<5:0>	bit21-16	R/W	<b>LCDC 时钟源预分频比选择位</b> 000000 = 1:1 000001 = 1:2 000010 = 1:3 ... .. 111111 = 1:64
—	bit15	—	—
BVS<2:0>	bit14-12	R/W	<b>LCD 显示灰度控制</b> 000: VBIAS = VLCD 001: VBIAS = (35/36)VLCD 010: VBIAS = (34/36)VLCD 011: VBIAS = (32/36)VLCD 100: VBIAS = (31/36)VLCD 101: VBIAS = (30/36)VLCD 110: VBIAS = (29/36)VLCD 111: VBIAS = (28/36)VLCD
RS<1:0>	bit11-10	R/W	<b>LCDC 内部偏压电阻选择</b> 00: 15K 欧姆 01: 50K 欧姆 10: 100K 欧姆 11: 200K 欧姆

BIAS<1:0>	bit9-8	R/W	<b>LCDC 偏置选择</b> 00: 1/3 BIAS 01: 保留 10: 保留 11: 1/4 BIAS
—	bit7	—	—
WFS	bit6	R/W	<b>LCDC 驱动波形类型选择位</b> 0: A 型波形（在每一公共端类型内改变相位） 1: B 型波形（在每一帧边界改变相位）
FLIK	bit5	R/W	<b>LCD 显示闪烁使能位</b> 0: 禁止 1: 使能
—	bit4-3	—	—
COMS<2:0>	bit2-0	R/W	公共端选择位（参考下表）

COMS<2:0>	COM 端口复用	最大像素数
11X	1/3 (COM2~COM0)	168
10X	1/2 (COM1~COM0)	112
011	1/8 (COM7~COM0)	448
010	1/6 (COM5~COM0)	336
001	1/4 (COM3~COM0)	224
000	—	—

表 5-4 LCD 复用类型列表

**LCD 闪烁时间寄存器 (LCD\_FLKT)**

偏移地址: 04<sub>H</sub>

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

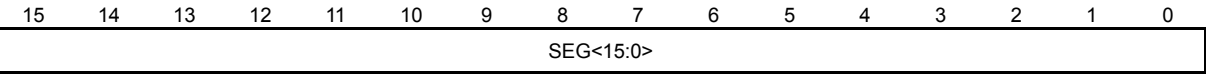
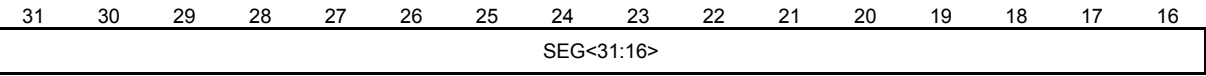
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOFF<7:0>								TON<7:0>							

—	bit31-16	—	—
TOFF<7:0>	bit15-8	R/W	<b>LCD 熄灭时间</b> LCD 闪烁熄灭时间= (TOFF+1) x 0.25 秒
TON<7:0>	bit7-0	R/W	<b>LCD 点亮时间</b> LCD 闪烁点亮时间= (TON+1) x 0.25 秒;

**LCD 段使能寄存器<31:0> (LCD\_SEL0)**

偏移地址: 08<sub>H</sub>

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

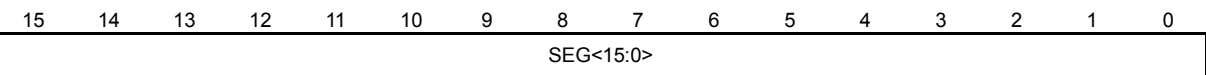
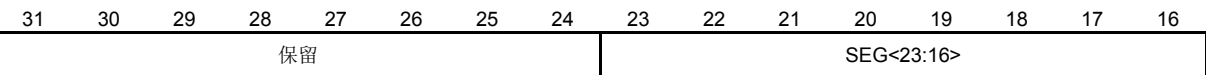


SEG<31:0>	bit31-0	R/W	<b>LCD 段驱动端口使能位</b> 0: 禁止 1: 使能
-----------	---------	-----	---------------------------------------

**LCD 段使能寄存器<55:32> (LCD\_SEL1)**

偏移地址: 0C<sub>H</sub>

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

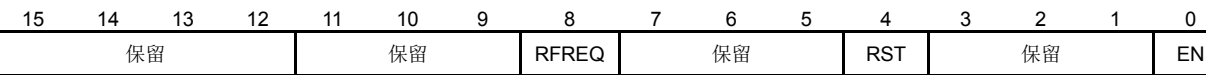
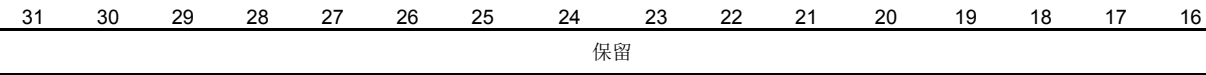


—	bit31-24	—	—
SEG<23:0>	bit23-0	R/W	<b>LCD 段驱动端口使能位</b> 0: 禁止 1: 使能

**LCD 控制寄存器 1 (LCD\_CON1)**

偏移地址: 10<sub>H</sub>

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



—	bit31-16	—	—
—	bit15~12	—	—
—	bit11-9	—	—
RFREQ	bit8	R/W	<b>像素寄存器刷新请求位 (需软件清 0)</b> 0: 无请求 1: 有请求
—	bit7-5	—	—

RST	bit4	W	<b>LCDC 驱动模块软件复位</b> 0: 读取时始终为 0 1: 软件复位
—	bit3-1	—	—
EN	bit0	R/W	<b>LCDC 驱动模块使能位</b> 0: 禁止 1: 使能

注: LCDC 驱动模块软件复位后, 禁止 LCDC 模块, LCD\_CON1 寄存器的位 EN=0; 禁止 LCDC 相关中断, LCD\_IE 寄存器的 OFFIE=0, RFIE=0; 复位相关中断标志为默认值, LCD\_IF 寄存器的 OFFIF=0, RFIF=0。

**LCD 中断使能寄存器 (LCD\_IE)**

偏移地址: 14<sub>H</sub>

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

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

—	bit31-2	—	—
RFIE	bit1	R/W	<b>LCDC 像素寄存器刷新中断使能位</b> 0: 禁止 1: 使能
OFFIE	bit0	R/W	<b>LCDC 驱动模块关闭中断使能位</b> 0: 禁止 1: 使能

**LCD 中断标志寄存器 (LCD\_IF)**

偏移地址: 18<sub>H</sub>

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

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

—	bit31-2	—	—
RFIF	bit1	R/W	<b>LCDC 像素寄存器刷新中断标志位</b> 0: 不允许刷新像素寄存器 1: 当前帧显示完毕, 允许刷新像素寄存器 软件写 1 清除标志位, 写 0 无效
OFFIF	bit0	R/W	<b>LCDC 驱动模块关闭中断标志位</b> 0: 未关闭

			1: 当前帧显示完毕后关闭 软件写 1 清除标志位, 写 0 无效
--	--	--	--------------------------------------

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

**LCD 像素寄存器 0 (LCD\_D0)**

偏移地址: 20<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCDD0<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDD0<15:0>															

LCDD0<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 1 (LCD\_D1)**

偏移地址: 24<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCDD1<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDD1<15:0>															

LCDD1<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 2 (LCD\_D2)**

偏移地址: 28<sub>H</sub>

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

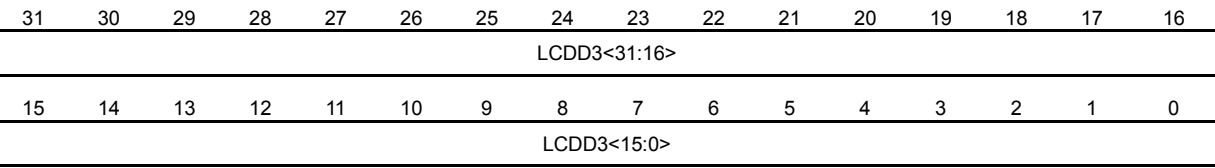
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCDD2<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDD2<15:0>															

LCDD2<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 3 (LCD\_D3)**

偏移地址: 2C<sub>H</sub>

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

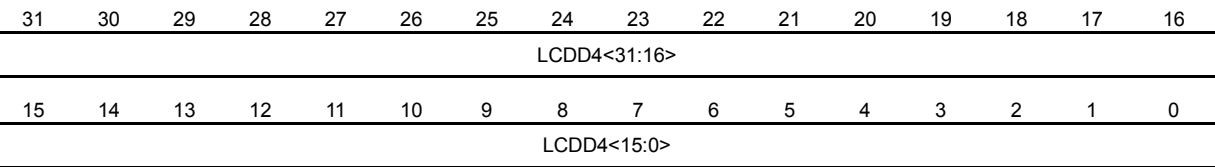


LCDD3<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 4 (LCD\_D4)**

偏移地址: 30<sub>H</sub>

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

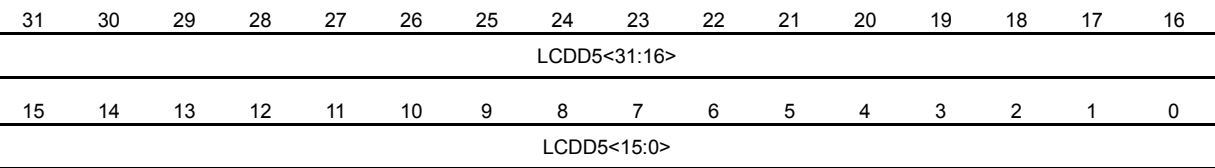


LCDD4<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 5 (LCD\_D5)**

偏移地址: 34<sub>H</sub>

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



LCDD5<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 6 (LCD\_D6)**



偏移地址: 38<sub>H</sub>

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

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16  
LCDD6<31:16>

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0  
LCDD6<15:0>

LCDD6<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 7 (LCD\_D7)**

偏移地址: 3C<sub>H</sub>

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

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16  
LCDD7<31:16>

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0  
LCDD7<15:0>

LCDD7<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 8 (LCD\_D8)**

偏移地址: 40<sub>H</sub>

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

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16  
LCDD8<31:16>

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0  
LCDD8<15:0>

LCDD8<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 9 (LCD\_D9)**

偏移地址: 44<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCDD9<31:16>															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDD9<15:0>															

LCDD9<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
-------------	---------	-----	--

**LCD 像素寄存器 10 (LCD\_D10)**

偏移地址: 48<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCDD10<31:16>															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDD10<15:0>															

LCDD10<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
--------------	---------	-----	--

**LCD 像素寄存器 11 (LCD\_D11)**

偏移地址: 4C<sub>H</sub>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCDD11<31:16>															

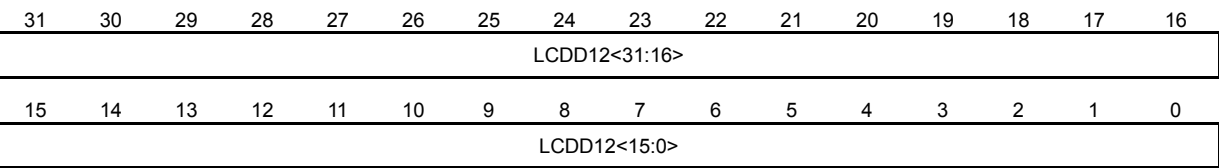
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDD11<15:0>															

LCDD11<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
--------------	---------	-----	--

**LCD 像素寄存器 12 (LCD\_D12)**

偏移地址: 50<sub>H</sub>

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

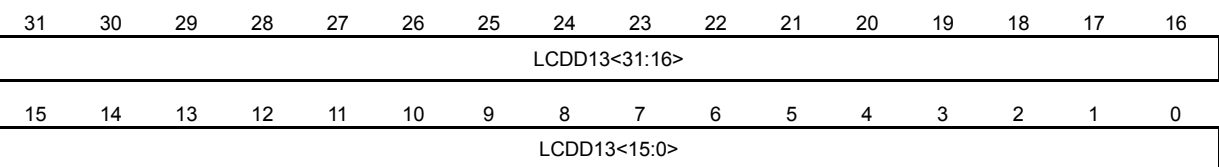


LCDD12<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
--------------	---------	-----	--

**LCD 像素寄存器 13 (LCD\_D13)**

偏移地址: 54<sub>H</sub>

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



LCDD13<31:0>	bit31-0	R/W	<b>像素点亮位</b> 0: 不点亮像素 (透明) 1: 点亮像素 (不透明)
--------------	---------	-----	--

## 5.9 看门狗定时器 (WDT)

### 5.9.1 概述

当配置字配置为“看门狗使能”时，上电复位后看门狗立即启动 (WDT 使用 32KHz WDT 时钟计数)，WDT 的载入配置字中 WDTRL 参数对应的计数初值的 1/4，并进行递减计数，计数到 0 时，窗口计数器加“1”，并在下一计数时钟到来时，计数器再次载入 WDTRL 参数对应的计数初值的 1/4，并继续递减计数。当窗口计数器计数到 2 时 (WDT 累计计数为 WDTRL 的一半)，WDT 产生中断标志。当窗口计数器计数到 4 (WDT 累计计数等于 WDTRL) 之前，没有在相应的喂狗窗口期进行喂狗动作，WDT 模块将产生复位信号。软件只能读写 WDT\_LOCK、写 WDT\_INTCLR，其他 WDT 相关特殊功能寄存器不再可写，只能读取，读 WDT\_LOAD 得到的值为“WDTRL 参数对应的计数初值”，读 WDT\_CON 将返回“0x0000\_000F”。

当配置字配置为“看门狗禁止”时，上电看门狗不启动，但软件可配置看门狗使能。软件配置 WDT\_CON 寄存器的 EN=1，WDT 计数器载入 WDT\_LOAD 寄存器值的 1/4，开始递减计数，当计数到 0 时，窗口计数器加“1”，并在下一个计数时钟到来时，计数器再次载入 WDT\_LOAD 寄存器值的 1/4，并继续递减计数。当窗口计数器计数到 2 时 (WDT 累计计数为 WDT\_LOAD 的一半)，WDT 产生中断标志。当窗口计数器计数到 4 (WDT 累计计数等于 WDT\_LOAD) 之前，没有在相应的喂狗窗口期进行喂狗动作，则 WDT 模块将产生复位信号。配置 WDT\_CON 寄存器的 CLKS，可以选择计数时钟源；配置 WDT\_LOAD 寄存器，可设置计数初值；读取 WDT\_VALUE，可得到 WDT 当前计数值。写入装载值寄存器 WDT\_LOAD 时，计数器当前值寄存器被清 0。

当配置字配置为“看门狗间隔中断使能”时，看门狗计数器计数到装载值的 50% 时产生 IRQ 中断请求，该中断请求可对睡眠模式进行唤醒。

在喂狗窗口期写 WDT\_INTCLR 会导致 WDT 重新载入计数初值，并重新进行递减计数。在喂狗窗口期外写 WDT\_INTCLR 会导致产生 WDT 复位。喂狗窗口期的定义参考“芯片配置字”。

WDT 模块支持寄存器写保护，配置寄存器 WDT\_LOCK=0x1ACCE551，可去除 WDT 寄存器的写保护状态，进行被保护寄存器的写操作，否则无法对被保护寄存器写入。

### 5.9.2 特殊功能寄存器

#### WDT 计数器装载值寄存器 (WDT\_LOAD)

偏移地址: 00<sub>H</sub>

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

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

LOAD<31:0>	bit31-0	W	<b>WDT 计数器重载值</b> 计数范围 0x0000_0001~0xFFFF_FFFF。如果为 0，WDT 不计数。
------------	---------	---	--

**WDT 计数器当前值寄存器 (WDT\_VALUE)**

偏移地址: 04<sub>H</sub>

复位值: 00111111\_11111111\_11111111\_11111111<sub>B</sub>

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

ST<1:0>	bit31-30	R	<b>WDT 计数器当前窗口计数器值</b> 00: WDT 计数值处于 0~25%之间 01: WDT 计数值处于 25~50%之间 10: WDT 计数值处于 50~75%之间 11: WDT 计数值处于 75~100%之间
VALUE<29:0>	bit29-0	R	<b>WDT 计数器当前值</b> 当前窗口下 WDT 计数器的计数值

**WDT 控制寄存器 (WDT\_CON)**

偏移地址: 08<sub>H</sub>

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

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

—	bit31-4	—	—
CLKS	bit3	R/W	<b>WDT 计数时钟选择位</b> 0: PCLK 1: LRC (寄存器位 WDT32K_SEL=0), 或 LOSC (寄存器位 WDT32K_SEL=1)
RSTEN	bit2	R/W	<b>WDT 复位使能位</b> 0: 禁止 1: 使能, WDT 计数到 0 时, 产生复位信号, 将芯片复位
IE	bit1	R/W	<b>WDT 中断使能位</b> 0: 禁止 1: 使能, WDT 计数到 0 时, 产生中断标志
EN	bit0	R/W	<b>WDT 模块使能位</b> 0: 禁止 1: 使能

**WDT 中断标志清除寄存器 (WDT\_INTCLR)**

偏移地址: 0C<sub>H</sub>

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

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

INTCLR<31:0>	bit31-0	W	<b>WDT 中断标志清 0 位</b> 对 WDT_INTCLR 寄存器进行任意写操作, WDT 中断标志位均被清零, 计数器重载 WDT_LOAD 寄存器值, 继续递减计数。
--------------	---------	---	--

**WDT 中断标志寄存器 (WDT\_RIS)**

偏移地址: 10<sub>H</sub>

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

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

—	bit31-1	—	—
WDTIF	bit0	R	<b>WDT 中断标志位</b> 0: 未产生中断 1: WDT 计数器计数到 0, 产生中断

**WDT 访问使能寄存器 (WDT\_LOCK)**

偏移地址: 00<sub>H</sub>

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

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

—	bit31-1	W	—
WDTLOCK	bit0	R/W	<b>WDT 寄存器保护状态位</b> 0: WDT 寄存器处于未保护状态 1: WDT 寄存器处于保护状态 对 WDT_LOCK 寄存器写入 0x1ACCE551, 被保护的寄存器处于未保护状态; 写入其它值, 处于保护状态

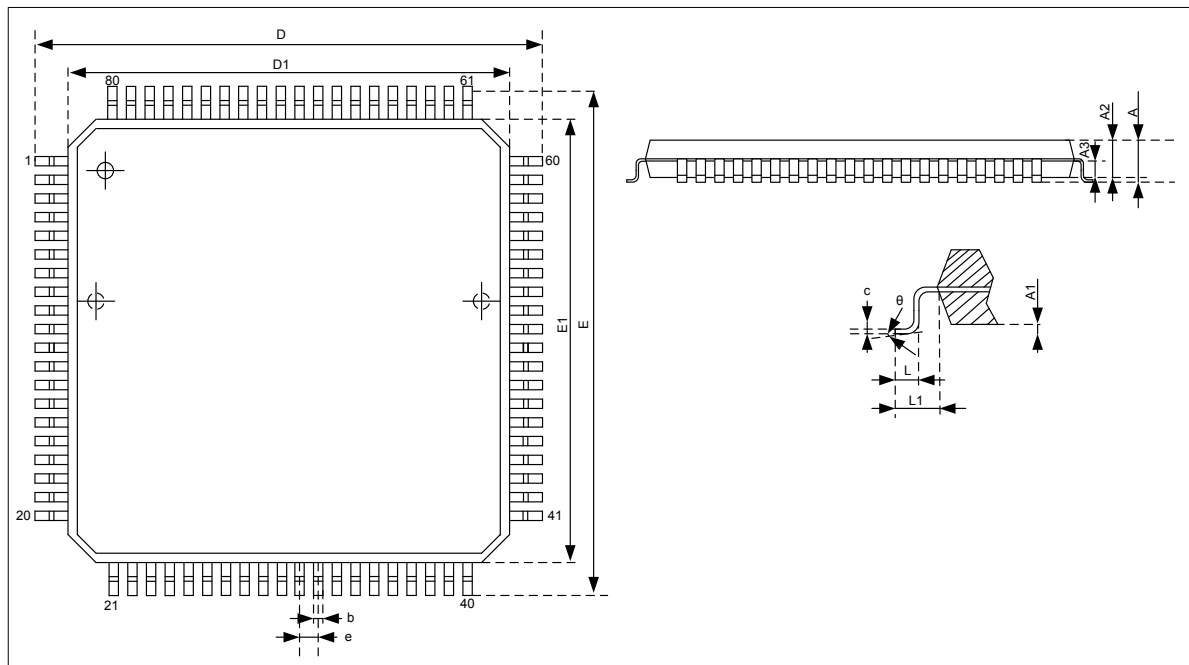
注 1: WDT\_LOCK 寄存器为只写 32 位寄存器, 其中仅 LOCK 位可读, 该寄存器必须使用字操作访问方式。

注 2: WDT\_LOCK 保护的寄存器为 WDT\_LOAD, WDT\_CON, WDT\_INTCLR。

## 第6章 芯片封装图

### 6.1 80-PIN 封装图

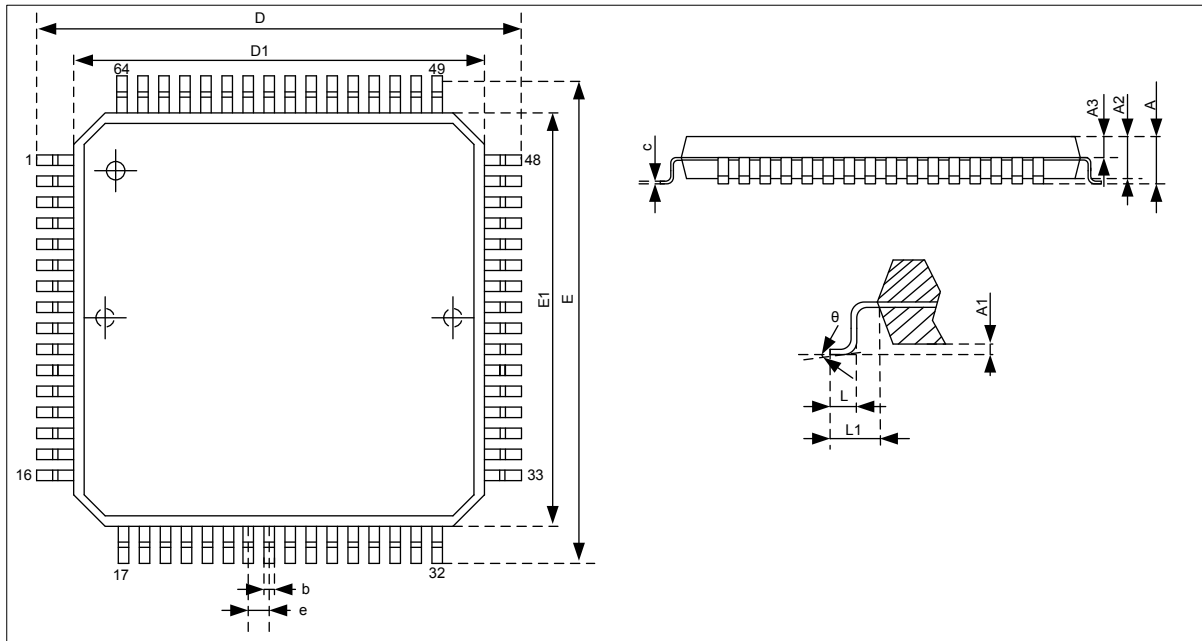
LQFP80



标号	公制 (mm)			英制 (inch)		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
A3	0.59	0.64	0.69	0.023	0.025	0.027
b	0.17	—	0.27	0.007	—	0.011
c	0.09	—	0.20	0.004	—	0.008
D	13.80	14.00	14.20	0.543	0.551	0.559
D1	11.90	12.00	12.10	0.469	0.472	0.476
E	13.80	14.00	14.20	0.543	0.551	0.559
E1	11.90	12.00	12.10	0.469	0.472	0.476
e	0.50BSC			0.50BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00BSC			1.00BSC		
θ	0	—	7°	0	—	7°

## 6.2 64-PIN 封装图

LQFP64



标号	公制 (mm)			英制 (inch)		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.20	0.002	—	0.008
A2	1.35	1.40	1.45	0.053	0.055	0.057
A3	0.59	0.64	0.69	0.023	0.025	0.027
b	0.19	—	0.27	0.007	—	0.011
c	0.13	—	0.18	0.005	—	0.007
D	11.80	12.00	12.20	0.465	0.472	0.480
D1	9.90	10.00	10.10	0.390	0.394	0.398
E	11.80	12.00	12.20	0.465	0.472	0.480
E1	9.90	10.00	10.10	0.390	0.394	0.398
e	0.50BSC			0.50BSC		
L	0.45	—	0.75	0.018	—	0.030
L1	1.00BSC			1.00BSC		
θ	0	—	7°	0	—	7°



## 附录1 Cortex-M0 内核描述

### 附录1.1 Cortex-M0 指令集

Cortex-M0 指令集，基本指令共 56 条，其中 50 条指令为 16 位，6 条指令为 32 位，并由多条指令可以进行助记符扩展，完成不同功能，如指令运行结果是否影响条件标志位等。

32 位指令是：BL，DSB，DMB，ISB，MRS 和 MSR。

指令中的符号说明：

- 1) 方括号 < >，表示括号内的任一种格式的操作数，均可作为指令操作数。  
如 <Rm | #imm> 表示操作数可以是寄存器 Rm，或者立即数 #imm。
- 2) 大括号 { }，表示括号内的操作数或符号可选。  
如：MOV{S}，表示指令助记符可以是 MOV 或 MOV<sub>S</sub>，区分该指令是否影响条件标志位。  
{Rd,}，表示指令中的目标操作数 Rd 可有可没有，没有时根据不同指令直接确定。

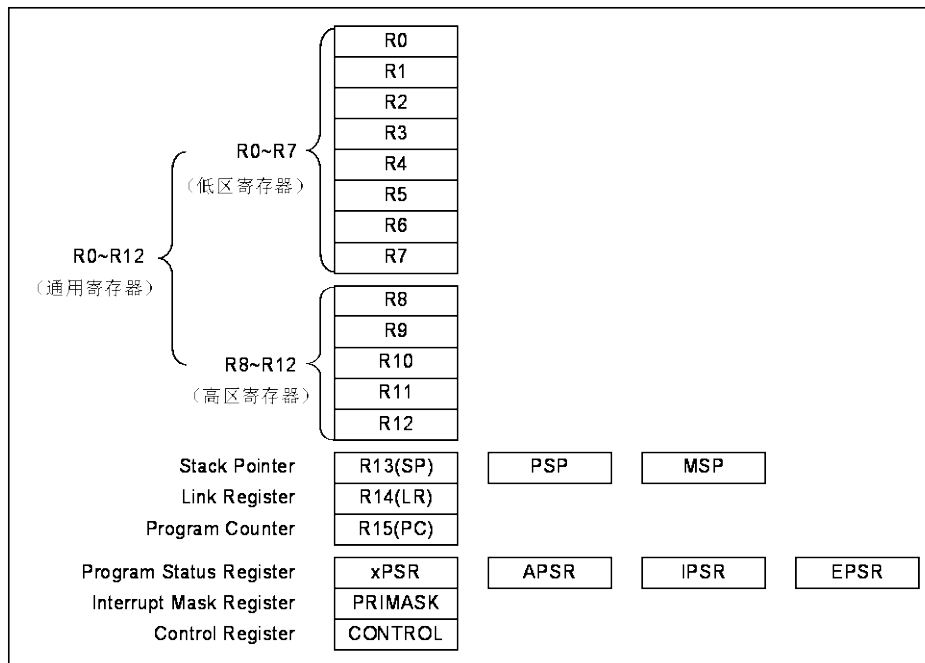
助记符	操作数	描述	影响标志位
ADR	Rd,Label	取 Label 地址到寄存器	—
LDR	Rt,Label	按字读 memory 到 Rt，立即数寻址	—
LDR	Rt,[Rn,<Rm   #imm>]	按字读 memory 到 Rt，带地址偏移寻址	—
LDRB	Rt,[Rn,<Rm   #imm>]	按字节读 memory 到 Rt，0 扩展为 32 位	—
LDRH	Rt,[Rn,<Rm   #imm>]	按半字读 memory 到 Rt，0 扩展为 32 位	—
LDRSB	Rt,[Rn,Rm]	按字节读 memory 到 Rt，符号位扩展	—
LDRSH	Rt,[Rn,Rm]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

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

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

## 附录1.2 Cortex-M0 内核寄存器

Cortex-M0 内核寄存器如下图所示：



### 附录1.2.1 通用寄存器R0~R12

R0~R12 为 32 位通用寄存器，用于数据操作。

### 附录1.2.2 堆栈指针寄存器SP (R13)

Cortex-M0 内核有两个堆栈指针 MSP 和 PSP，但两者不能同时使用，具体使用的堆栈指针与进程模式有关。在线程模式下，配置 CONTROL 寄存器的 SPSEL 位，可选择当前使用的堆栈指针。编写指令时，两个堆栈指针均可通过 R13 或 SP 调用，访问当前正在使用的堆栈指针，也可通过 MRS/MSR 指令访问指定的堆栈指针。

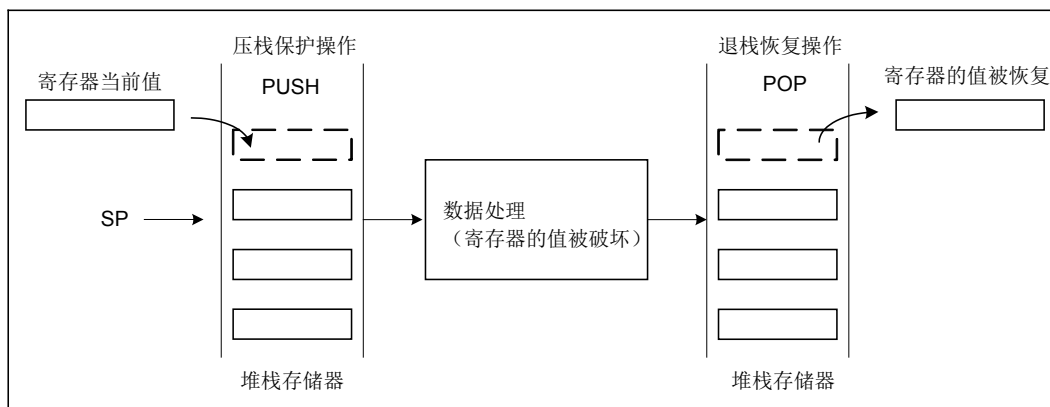
主堆栈指针 (MSP)：或写作 SP\_main，主要由操作系统内核，异常/中断服务程序，以及其它被授权访问的应用程序来使用，芯片复位后缺省使用主堆栈指针。

进程堆栈指针 (PSP)：或写作 SP\_process，在线程模式下，用户可选用进程堆栈指针；但在异常/中断服务程序中不能选用进程堆栈指针。

堆栈指针的最低两位始终是 0，即堆栈总是字（4 个字节）对齐的。

对应用程序，通常只需使用主堆栈指针 MSP，并且 PUSH 和 POP 指令也默认使用 MSP。

堆栈由一块地址连续的存储器空间，和一个栈顶指针组成，实现“先进后出”操作的缓冲区，常用于在异常/中断处理前后，保存和恢复一些关键寄存器的值。堆栈操作示意图如下：



### 附录1.2.3 链接寄存器LR (R14)

链接寄存器 LR，也称为寄存器 R14，用于在调用子程序时存储返回地址。例如，当执行 BL 指令时，硬件电路会自动将下一条指令的地址保存到寄存器 LR。

### 附录1.2.4 程序计数器PC (R15)

程序计数器 PC，也称为寄存器 R15。Cortex-M0 内核使用了指令流水线，所以读 PC 时，得到的值是当前指令的地址加 4。

如果对 PC 进行写操作，会产生程序跳转（但不更新 LR 寄存器），新写入的值即为程序跳转目的地址。Cortex-M0 中的指令至少是半字对齐的，所以 PC 的 LSB 位始终读取为 0。但无论是直接写 PC 还是使用跳转指令，都必须保证加载到 PC 的值 LSB 位为 1，用于表示这是在 Thumb 模式下执行指令，否则会被视为企图转入 ARM 模式，Cortex-M0 内核将产生一个 Fault 异常。

### 附录1.2.5 程序状态寄存器xPSR

程序状态寄存器 xPSR，根据其各个状态位的功能，又划分为三个子状态寄存器：应用程序状态寄存器 APSR，中断服务程序状态寄存器 IPSR，执行程序状态寄存器 EPSR。

通过 MRS/MSR 指令，可对 3 个子状态寄存器进行单独访问，也可以同时访问其中 2 个或 3 个子状态寄存器。寄存器名称 IAPSR 表示同时访问 IPSR 和 APSR；寄存器名称 EAPSR 表示同时访问 EPSR 和 APSR；寄存器名称 IEPSPR 表示同时访问 IPSR 和 EPSR；寄存器名称 XPSR 表示同时访问 3 个子状态寄存器。

程序状态寄存器 xPSR 及其 3 个子状态寄存器的状态位划分如下表所示：

	31	30	29	28	27:25	24	23:6	5:0	
xPSR	N	Z	C	V	Reserved	T	Reserved	Exception Number	
APSR	N	Z	C	V	Reserved				
IPSR	Reserved							Exception Number	
EPSR	Reserved					T	Reserved		

应用程序状态寄存器 APSR 中的各个状态位，用于说明指令执行结果，各状态位描述如下：

N：负数标志。指令执行结果为负数时，标志位 N=1，否则 N=0。

**Z:** 零标志。指令执行结果为零时，标志位  $Z=1$ ，否则  $Z=0$ 。对比较指令，如果被比较的两个数相等，则  $Z=1$ 。

**C:** 进位或借位标志。

对加法指令，如果执行结果有进位（结果 $\geq 2^{32}$ ），则  $C=1$ ，否则  $C=0$ ；

对减法指令，如果执行结果无借位（结果 $\geq 0$ ），则  $C=1$ ，否则  $C=0$ ；

对移位循环指令，取决于移位到 **C** 标志的数据位。

**V:** 溢出标志。

两个负数相加，结果为正数（ $\text{bit}\langle 31 \rangle = 0$ ）时溢出，则  $V=1$ ，否则  $V=0$ ；

两个正数相加，结果为负数（ $\text{bit}\langle 31 \rangle = 1$ ）时溢出，则  $V=1$ ，否则  $V=0$ ；

负数减去正数，结果为正数（ $\text{bit}\langle 31 \rangle = 0$ ）时溢出，则  $V=1$ ，否则  $V=0$ ；

正数减去负数，结果为负数（ $\text{bit}\langle 31 \rangle = 1$ ）时溢出，则  $V=1$ ，否则  $V=0$ 。

中断服务程序状态寄存器 **IPSR** 中的状态位，用于表示正在处理的异常/中断号，目前正在执行异常/中断服务程序；如果  $\text{IPSR}\langle 5:0 \rangle = 0$ ，则表示目前是线程模式，未进行异常/中断处理。

执行程序状态寄存器 **ESPR** 中的 **T** 状态位，用于表示处理器是否处于 **Thumb** 模式。由于 **Cortex-M0** 处理器只支持 **Thumb** 模式，**T** 状态位应该始终为 **1**，如果将 **T** 位写为 **0**，会产生 **HardFault** 异常。使用 **MRS** 指令读取寄存器 **EPSR** 时，返回值为 **0**；如果使用 **MSR** 指令向 **EPSR** 写数据，写操作会被忽略。

### 附录1.2.6 异常/中断屏蔽寄存器**PRIMASK**

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

	31:1	0
PRIMASK	Reserved	PM

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

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

### 附录1.2.7 控制寄存器**CONTROL**

控制寄存器 **CONTROL**，可用于在线程模式下，选择所使用的堆栈指针。

	31:2	1	0
CONTROL	Reserved	SPSEL	Reserved

当堆栈指针选择位  $\text{SPSEL}=0$  时，选择 **MSP** (**SP\_main**) 作为当前堆栈指针；当  $\text{SPSEL}=1$  时，选择 **PSP** (**SP\_process**) 作为当前堆栈指针。

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

处理器硬件电路会在异常/中断处理程序入口和返回时，对 **SPSEL** 位进行更新，确保进入异常/中断处理程序后，使用 **MSP** 作为堆栈指针，并在返回时恢复线程模式下的选择。在线程模式下，可配置 **SPSEL**，选择当前使用的堆栈指针。

通过 **MRS/MSR** 指令可访问两个的堆栈指针。在修改 **SPSEL** 位的指令后，需立即执行 **ISB**（指令同步隔离）指令，确保在 **SPSEL** 位修改完成，新的堆栈指针生效后，才会执行后续其它指令。

## 附录2 电气特性

### 附录2.1 参数特性表

#### 附录2.1.1 芯片工作条件

◆ 最大标称值

参数	符号	条件	标称值	单位
电源电压	VDD	VSS=0V	-0.3 ~ 7.0	V
管脚输入电压	V <sub>IN</sub>	VSS=0V	-0.3 ~ VDD + 0.3	V
管脚输出电压	V <sub>OUT</sub>	VSS=0V	-0.3 ~ VDD + 0.3	V
VDD 管脚最大输入电流	I <sub>MAXVDD</sub>	VDD=5.0V, 25°C	100	mA
VSS 管脚最大输出电流	I <sub>MAXVSS</sub>	VDD=5.0V, 25°C	120	mA
芯片存储温度	T <sub>STG</sub>	—	-55 ~ 150	°C
芯片最高结温	T <sub>JUNC</sub>	—	125	°C
芯片湿度敏感等级	MSL	IPC/JEDEC J-STD-020	3	等级

◆ 芯片工作条件表

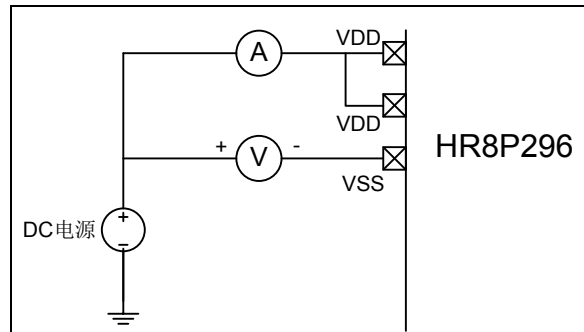
参数	符号	工作条件	最小值	最大值	单位
芯片工作温度	T <sub>OPR</sub>	—	-40	85	°C
芯片结温	T <sub>JUNC</sub>	—	-40	105	°C
芯片工作电压	VDD	—	2.5	5.5	V
AHB 总线频率	F <sub>HCLK</sub>	—	—	33	MHz
APB 总线频率	F <sub>PCLK</sub>	—	—	33	MHz

◆ 芯片功能模块工作电压范围表

参数	符号	工作条件	VDD 电压	备注
VR30 工作电压	V <sub>VR30</sub>	-40 ~ 85°C VR30 负载电 流等于 1mA	3.2~5.5V	当 VDD 电压小于 3.2V 时, VR30 输出跟随 VDD
VR36 工作电压	V <sub>VR36</sub>	-40 ~ 85°C VR36 负载电 流等于 1mA	3.8~5.5V	当 VDD 电压小于 3.8V 时, VR36 输出跟随 VDD
ADC 工作电压	V <sub>ADC</sub>	-40 ~ 85°C	2.5~5.5V	当 ADC 采用内部 3.0V LDO VR30 供电时, VDD 操作电压范 围为 2.7~5.5V
LCD 工作电压	V <sub>LCD</sub>	-40 ~ 85°C	2.8~5.5V	当 LCD 采用内部 3.6V LDO VR36 供电时, VDD 操作电压范 围范围为 3.0~5.5V

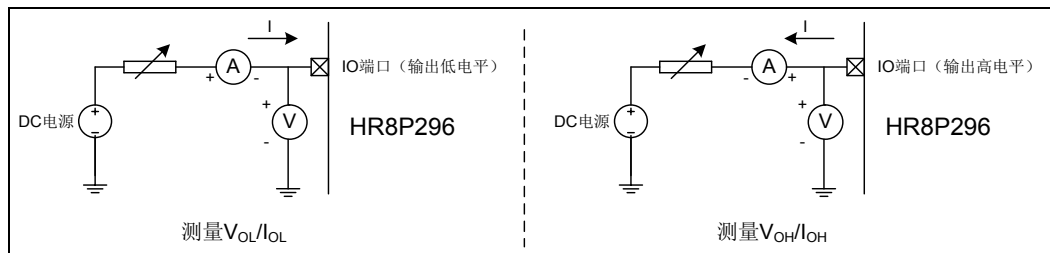
### 附录2.1.2 芯片特性参数测量方法

#### ◆ 芯片功耗参数测量方法



芯片功耗测量连接示意图

#### ◆ 芯片 IO 端口参数测量方法



芯片 IO 端口输出特性参数测量连接示意图

### 附录2.1.3 芯片功耗特性

#### ◆ 芯片功耗特性参数表

参数	符号	最小值	典型值	最大值	单位	工作条件
芯片供电电压	VDD	2.5	—	5.5	V	-40℃ ~ 85℃
芯片静态电流	I <sub>DD</sub>	—	350	500	μA	25℃, 上电复位, VDD = 5V, 所有的 I/O 端口输入低电平, MRSTN=0, LOSC0=0, LOSC1 悬空。
深度睡眠模式下芯片电流	I <sub>PD1</sub>	—	12	16	μA	25℃, VDD = 5V, BOR 不使能, WDT 不使能, RTC 使能, ADC buffer 被 bypass, 所有 I/O 端口输出固定电平, 无负载。
			10	14	μA	25℃, VDD = 5V, BOR 不使能, WDT 不使能, RTC 不使能, ADC buffer 被 bypass, 所有 I/O 端口输出固定电平, 无负载。



参数	符号	最小值	典型值	最大值	单位	工作条件
睡眠模式下芯片电流	$I_{PD2}$	—	0.8	1	mA	25°C, VDD = 5V, BOR 不使能, WDT 不使能, RTC 使能, ADC buffer 被 bypass, 所有 I/O 端口输出固定电平, 无负载; 系统主时钟为内部 16MHz RC 时钟。
	$I_{PD3}$	—	1.2	1.5	mA	25°C, VDD = 5V, BOR 不使能, WDT 不使能, RTC 使能, ADC buffer 被 bypass, 所有 I/O 端口输出固定电平, 无负载; 系统主时钟为外部 20MHz HS 时钟。
正常运行模式芯片电流	$I_{OP1}$	—	3.1	4	mA	25°C, VDD = 5V, BOR 使能, WDT 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, 7816 通讯时钟管脚, 作为普通 I/O, 无时钟输出; 系统主时钟为内部 16MHz RC 时钟。
	$I_{OP2}$	—	3.5	4.5	mA	25°C, VDD = 5V, BOR 使能, WDT 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, 7816 通讯时钟管脚, 作为普通 I/O, 无时钟输出; 系统主时钟为外部 20MHz HS 时钟。

注: 在芯片应用时, 需软件设置寄存器位 ADC\_BUF\_EN[1:0]=00, 否则可能会导致芯片 ADC 模块工作异常, 芯片功耗偏大。

◆ 芯片功能模块功耗特性参数表

参数	符号	最小值	典型值	最大值	单位	工作条件
BOR 模块电流	$I_{BOR}$	—	1	—	μA	25°C, VDD = 5V
LVD 模块电流	$I_{LVD}$	—	1	—	μA	25°C, VDD = 5V
VR30 模块电流	$I_{VR30}$	—	3	—	μA	25°C, VDD = 5V

参数	符号	最小值	典型值	最大值	单位	工作条件
VR36 模块电流	$I_{VR36}$	—	3	—	$\mu\text{A}$	$25^{\circ}\text{C}$ , $V_{DD} = 5\text{V}$
WDT 模块电流	$I_{WDT}$	—	1	—	$\mu\text{A}$	$25^{\circ}\text{C}$ , $V_{DD} = 5\text{V}$
ADC 模块电流	$I_{ADC}$	—	500	—	$\mu\text{A}$	$25^{\circ}\text{C}$ , $V_{DD} = 5\text{V}$
LCDC 模块电流	$I_{LCDC}$	—	10	—	$\mu\text{A}$	$25^{\circ}\text{C}$ , $V_{DD} = 5\text{V}$
RTC 模块电流	$I_{RTC}$	—	1.5	—	$\mu\text{A}$	$25^{\circ}\text{C}$ , $V_{DD} = 5\text{V}$
FLASH 编程电流	$I_{PROG}$	—	—	4	$\text{mA}$	$-40\sim 85^{\circ}\text{C}$ , $V_{DD} = 5\text{V}$
FLASH 擦除电流	$I_{ERAS}$	—	—	4	$\text{mA}$	

## 附录2.1.4 芯片IO端口特性

### ◆ 芯片输入端口特性表

芯片工作温度范围: $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输入高电平 (见备注)	$V_{IH1}$	0.8VDD	—	VDD	V	$2.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
	$V_{IH2}$	0.5VDD	—	VDD	V	
I/O 端口输入低电平 (见备注)	$V_{IL1}$	VSS	—	0.2VDD	V	
	$V_{IL2}$	VSS	—	0.1VDD	V	
主复位信号, 输入高电平	$V_{IH}$	0.8VDD	—	VDD	V	
主复位信号, 输入低电平	$V_{IL}$	VSS	—	0.2VDD	V	
IO 端口施密特迟滞窗口 (见备注)	$V_{HYST1}$	—	1.4	—	V	VDD = 5.0V
	$V_{HYST2}$	—	1.0	—	V	
I/O 端口输入漏电流	$I_{IL}$	—	—	1	$\mu\text{A}$	VDD = 5.0V (端口高阻状态)
IO 端口弱上拉电阻	$R_{WPU}$	35	50	65	$\text{k}\Omega$	VDD = 5.0V $V_{pin} = V_{SS}$
IO 端口弱下拉电阻	$R_{WPD}$	25	35	45	$\text{k}\Omega$	VDD = 5.0V $V_{pin} = V_{DD}$

注 1:  $V_{IH2}$ ,  $V_{IL2}$ ,  $V_{HYST2}$  适用于 PB23~PB16, 且  $\text{GPIO\_PBSMITSEL}[7:0]=8'b1111\_1111$  时的情况;

注 2:  $V_{IH1}$ ,  $V_{IL1}$ ,  $V_{HYST1}$  适用于 PB23~PB16, 且  $\text{GPIO\_PBSMITSEL}[7:0]=8'b0000\_0000$  时的情况, 以及其它不支持施密特迟滞窗口可选的 IO 端口;

◆ 芯片输出端口特性表

芯片工作温度范围: -40°C ~ 85°C						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输出高电平	I <sub>OH</sub>	-5.0	—	—	mA	VDD = 5.0V V <sub>OH</sub> = 4.6V
I/O 端口 (普通驱动) 输出低电平	I <sub>OL1</sub>	10	—	—	mA	VDD = 5.0V V <sub>OL</sub> = 0.4V
I/O 端口 (大电流驱动) 输出低电平	I <sub>OL2</sub>	30	—	—		

附录2.1.5 芯片系统时钟特性

◆ 系统时钟规格表

参数	符号	最小值	典型值	最大值	单位	测试条件
系统时钟频率	F <sub>OSC</sub>	—	—	33M	Hz	2.5V ≤ VDD ≤ 5.5V
系统时钟周期	T <sub>OSC</sub>	30.3	—	—	ns	
机器周期	T <sub>inst</sub>	60.6	—	—	ns	
外部时钟高电平和低电平时间	T <sub>OSL</sub> , T <sub>OSSH</sub>	25	—	—	ns	
外部时钟边沿上升和下降时间	T <sub>OSR</sub> , T <sub>OSF</sub>	—	—	10	ns	

附录2.1.6 上电和下电工作条件

参数	符号	最小值	典型值	最大值	单位	测试条件
上电复位	V <sub>POR</sub>	1.8	—	1.9	V	-40°C ~ 85°C 2.5V ≤ VDD ≤ 5.5V
VDD 上电初始电压	V <sub>start</sub>	0	—	0.2	V	
VDD 上升速率	t <sub>VDD</sub>	800	—	—	us/V	
VDD 下降速率		20	—	—	us/V	

附录2.1.7 芯片ESD特性

◆ ESD 特性参数表

参数	符号	等级	最大值	单位	测试条件
ESD 电压 (人体模型)	V <sub>ESDHBM</sub>	3A	4000	V	25°C, MIL-STD-883H
ESD 电压 (机器模型)	V <sub>ESDMM</sub>	3	400	V	25°C, JESD22-A115
Latchup 电流	I <sub>LAT</sub>	I	±350	mA	25°C, JESD78

注: 上述 ESD 特性参数值是基于理论设计值和被测样品的测试值, 不是批量产品测试值, 仅供芯片应用时参考。

### 附录2.1.8 芯片功能模块特性

#### ◆ ADC 模块特性表

参数名称	符号	最小值	典型值	最大值	单位	测试条件
分辨率	RES	—	—	12	bit	—
参考电压范围	V <sub>ADVREF</sub>	2.0	—	VDD	V	
模拟电压输入范围	V <sub>IN</sub>	VSS	—	V <sub>ADVREF</sub>	V	
输入电容	C <sub>IN</sub>	30	40	50	pF	
模拟通道推荐输入电阻	R <sub>IN</sub>	—	—	10k	Ω	
AD 转换时钟周期	T <sub>AD</sub>	1.5	8	256	μs	
AD 转换时间（不包括采样时间）	T <sub>CONV</sub>	—	13	—	T <sub>AD</sub>	
差分线性度	DNL	—	±1	±2	LSB	-40°C~85°C
失调误差	V <sub>OFFSET</sub>	—	2	4	LSB	

#### ◆ ADC 转换时钟源选择表

A/D 时钟源选择	工作频率			
	16M	8M	4M	1M
Fosc	不推荐使用	不推荐使用	不推荐使用	不推荐使用
Fosc/2	不推荐使用	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 2us
Fosc/4	不推荐使用	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 4us
Fosc/8	不推荐使用	不推荐使用	T <sub>ADCLK</sub> = 2us	T <sub>ADCLK</sub> = 8us
Fosc/16	不推荐使用	T <sub>ADCLK</sub> = 2us	T <sub>ADCLK</sub> = 4us	T <sub>ADCLK</sub> = 16us
Fosc/32	T <sub>ADCLK</sub> = 2us	T <sub>ADCLK</sub> = 4us	T <sub>ADCLK</sub> = 8us	T <sub>ADCLK</sub> = 32us
Fosc/64	T <sub>ADCLK</sub> = 4us	T <sub>ADCLK</sub> = 8us	T <sub>ADCLK</sub> = 16us	T <sub>ADCLK</sub> = 64us
Fosc/256	T <sub>ADCLK</sub> = 16us	T <sub>ADCLK</sub> = 32us	T <sub>ADCLK</sub> = 64us	T <sub>ADCLK</sub> = 256us

#### ◆ ADC 内部参考电压特性表

参数	符号	最小值	典型值	最大值	单位	测试条件
ADC 内部参考电压 VREF	V <sub>REF</sub>	2.027	2.048	2.069	V	25°C, VDD=5V
		1.987	—	2.110	V	-40°C~85°C, VDD=2.5~5.5V

#### ◆ 内部时钟源特性表

参数	符号	最小值	典型值	最大值	单位	测试条件
HRC 时钟频率	F <sub>HRC</sub>	15.84	16	16.16	MHz	25°C, VDD=2.5V~5.5V
		15.52	16	16.48	MHz	-40°C~85°C, VDD=2.5V~5.5V

参数	符号	最小值	典型值	最大值	单位	测试条件
HRC 起振时间	T <sub>HRC</sub>	—	30	—	us	—
LRC 时钟频率	F <sub>LRC</sub>	28.8	32	35.2	KHz	25°C, VDD=2.5V~5.5V
		12.8	32	51.2	KHz	-40°C~85°C, VDD=2.5V~5.5V
LRC 起振时间	T <sub>LRC</sub>	—	800	—	us	—

◆ PLL 锁相环特性表

参数		符号	最小值	典型值	最大值	单位	测试条件
时钟源 选择为 32KHz	PLL 输入时钟频率	F <sub>PLL1</sub>	25	32	38	KHz	-40~85°C
	PLL 倍频输出时钟频率	F <sub>PLLO1</sub>	25.6	32.768	38.912	MHz	
	PLL 锁频时间	T <sub>LOCK1</sub>	—	1	10	ms	
时钟源 选择为 4MHz	PLL 输入时钟频率	F <sub>PLL2</sub>	3.2	4	4.8	MHz	-40~85°C
	PLL 倍频输出时钟频率	F <sub>PLLO2</sub>	25.6	32	38.4	MHz	
	PLL 锁频时间	T <sub>LOCK2</sub>	—	1	10	ms	

注 1: PLL 的锁频时间 T<sub>LOCK1</sub> 和 T<sub>LOCK2</sub> 均为设计规格, 未经过实际芯片测试; 设计条件为-40~85°C。

注 2: PLL\_RDY/PLL\_LOCK 作为 PLL 的锁频标志信号, 仅供应用参考, 在高可靠应用中, 为防止 PLL 锁频标志信号失效而无法退出等待状态, 需要在等待 PLL 锁频标志信号的同时设定等待溢出时间, 建议设置等待溢出时间为 10ms, 在等待溢出时间到达时, 如仍未检测到 PLL 锁频标志信号置起, 则退出等待状态, 此时 PLL 已实际锁频稳定。

◆ BOR 模块特性表

BORV[1:0]	最小值	典型值	最大值	单位	测试条件
00	1.6	1.8	2.0	V	-40~85°C
01	2.0	2.2	2.4	V	
10	2.7	3.0	3.3	V	
11	3.3	3.6	3.9	V	

◆ LVD 模块特性表

LVDVS[3:0]		最小值	典型值	最大值	单位	测试条件
VDD 下降, LVDO 低电压 状态标志置 1	0000	2.2	2.4	2.6	V	-40~85°C
	0001	2.4	2.6	2.8	V	
	0010	2.6	2.8	3.0	V	
	0011	2.8	3.0	3.2	V	
	0100	3.0	3.2	3.4	V	

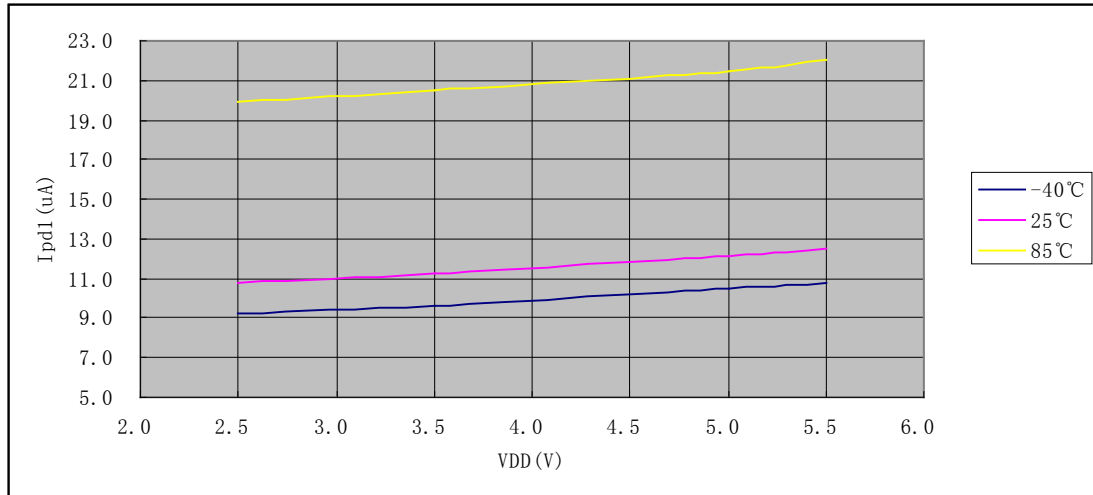
LVDVS[3:0]		最小值	典型值	最大值	单位	测试条件
	0101	3.2	3.4	3.6	V	
	0110	3.4	3.7	4.0	V	
	0111	3.6	3.9	4.2	V	
	1000	3.8	4.1	4.4	V	
	1001	4.0	4.3	4.6	V	
	1010	4.3	4.6	4.9	V	
	1011	4.5	4.8	5.1	V	
	1100	4.7	5.0	5.3	V	
	1101	4.9	5.2	5.5	V	
	111x	—	—	—	V	
LVD 电压检测迟滞窗口		—	100	—	mV	-40~85°C
LVDIN 阈值电压		1.1	1.2	1.3	V	-40~85°C

## 附录2.2 参数特性图

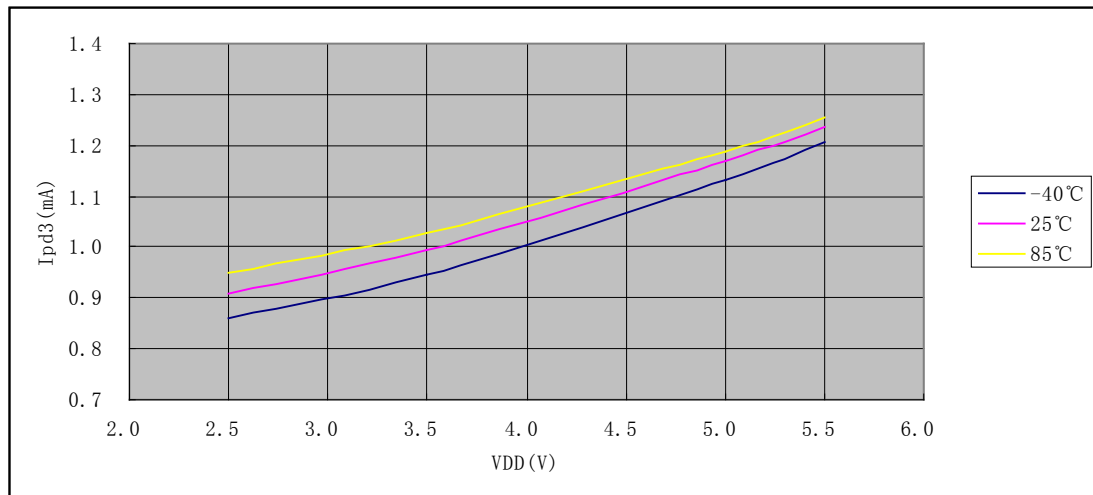
本节中所列图示均为抽样测试，仅作为设计参考之用。其中部分图示中所列的数据已超出指定的操作范围，此类信息也仅供参考，芯片只保证在指定的范围内正常工作。

### 附录2.2.1 芯片功耗特性

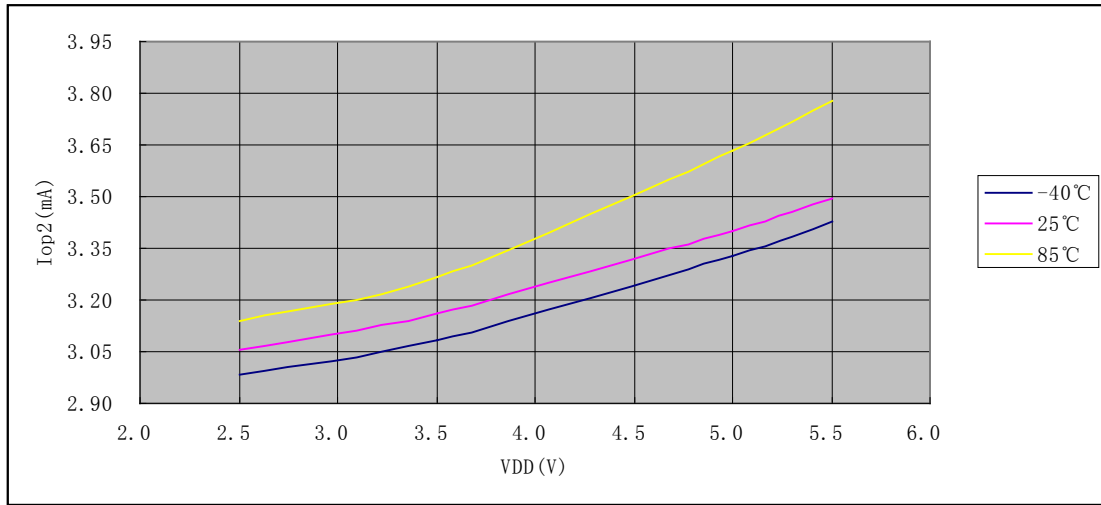
- ◆ 芯片深度睡眠模式电流随电压-温度变化特性图（WDT，BOR 不使能，RTC 使能，ADC buffer 被 bypass，所有 I/O 端口输出固定电平，无负载）



- ◆ 芯片睡眠模式电流随电压-温度变化特性图（WDT，BOR 不使能，RTC 使能，ADC buffer 被 bypass，所有 I/O 端口输出固定电平，无负载；系统主时钟为外部 20MHz HS 时钟）



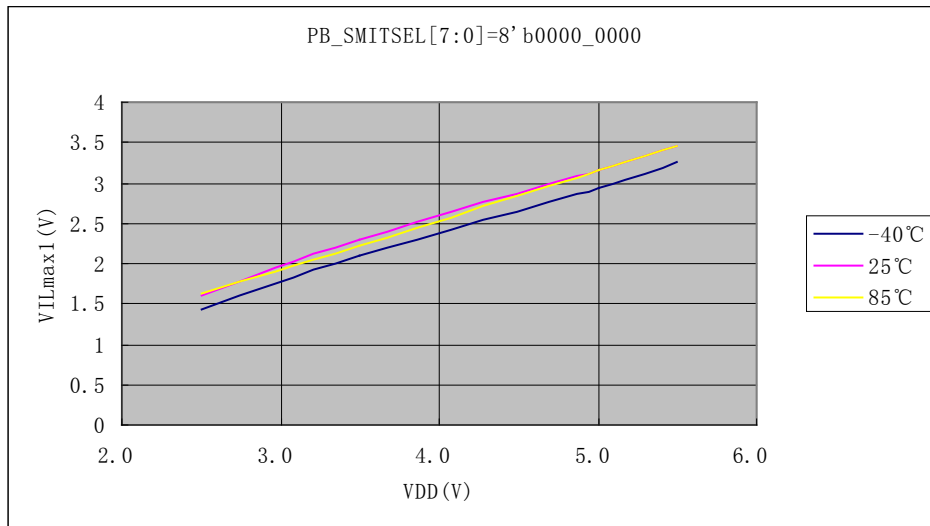
- ◆ 芯片运行模式电流随电压-温度变化特性图 (WDT, BOR 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, 7816 通讯时钟管脚作为普通 I/O, 无时钟输出; 系统主时钟为外部 20MHz HS 时钟)



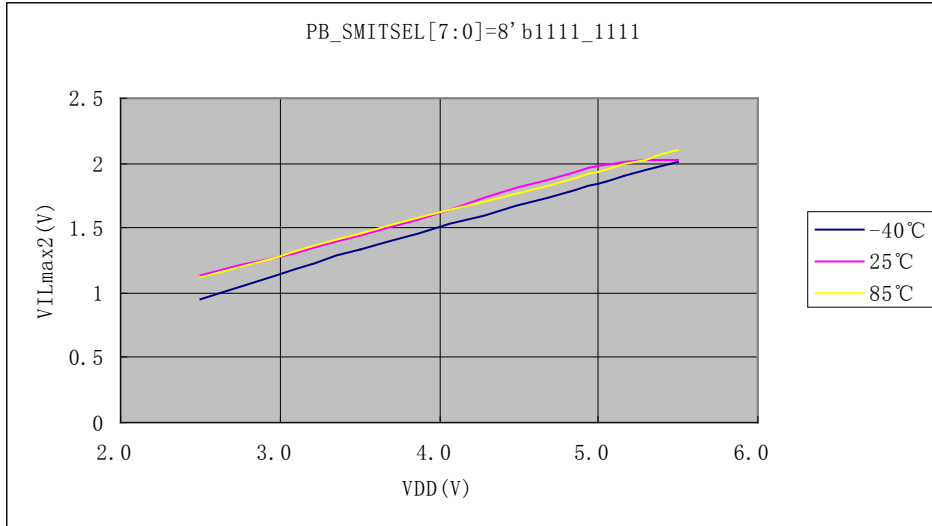
- ◆ 芯片运行模式电流随系统时钟频率-温度变化特性图 (WDT, BOR 使能, 外设模块均工作, 所有 I/O 端口输出固定电平, 无负载, 7816 通讯时钟管脚作为普通 I/O, 无时钟输出; 芯片工作电压 VDD=5.0V)

### 附录2.2.2 芯片IO端口输入特性

- ◆ I/O 端口最大输入低电平随电压-温度变化特性图

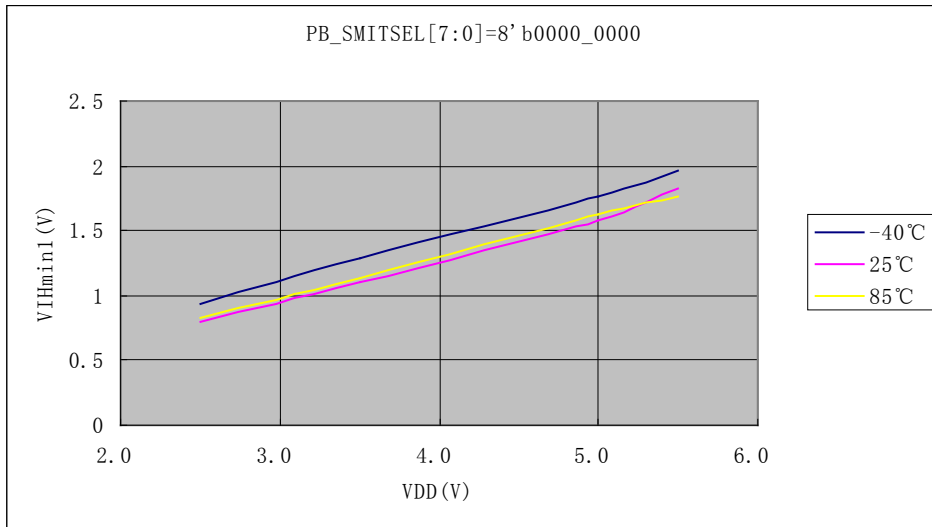


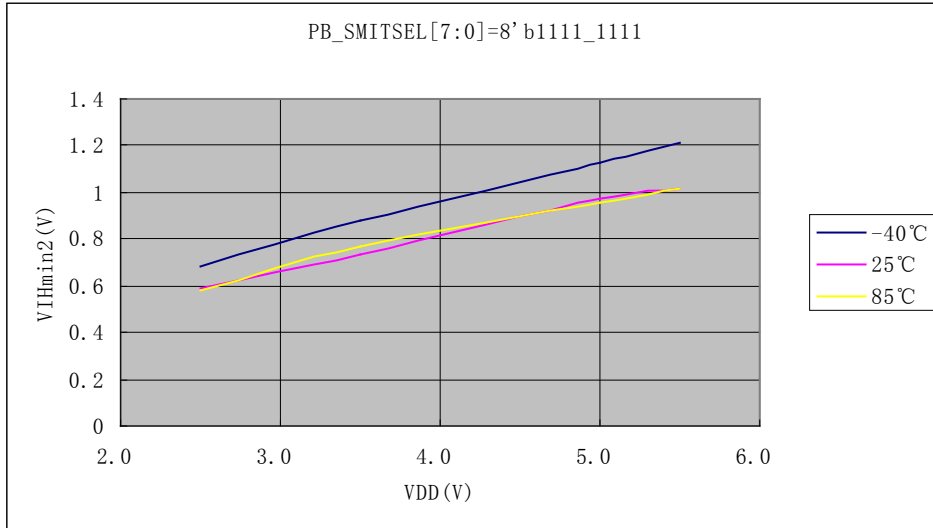




注 1: 测量  $V_{ILmax}$  时, IO 端口输入电平由 0V 开始上升, 直到端口输入采样到逻辑值 1;  
 注 2:  $V_{ILmax2}$  适用于 PB23~PB16, 且  $GPIO\_PBSMITSEL[7:0]=8'b1111\_1111$  时的情况;  
 注 3:  $V_{ILmax1}$  适用于 PB23~PB16, 且  $GPIO\_PBSMITSEL[7:0]=8'b0000\_0000$  时的情况, 以及其它不支持施密特迟滞窗口可选的 IO 端口。

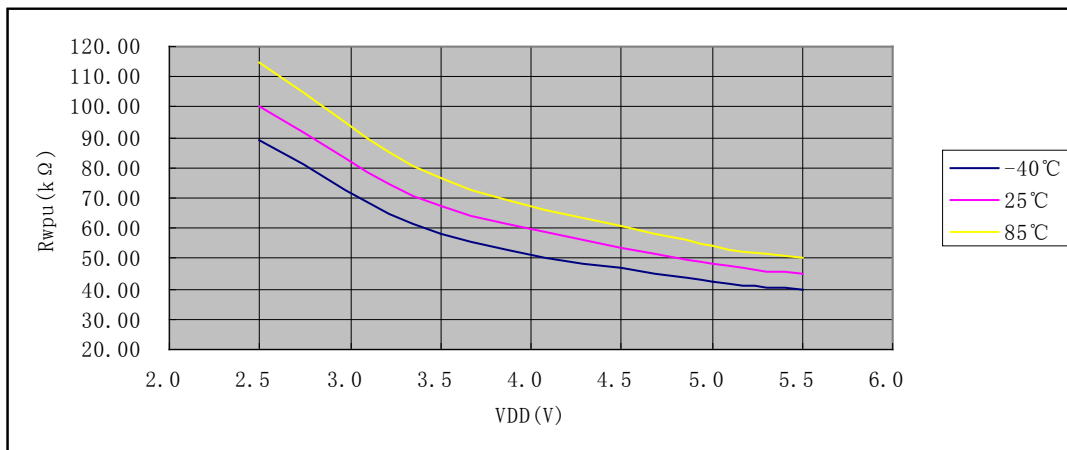
◆ I/O 端口最小输入高电平随电压-温度变化特性图



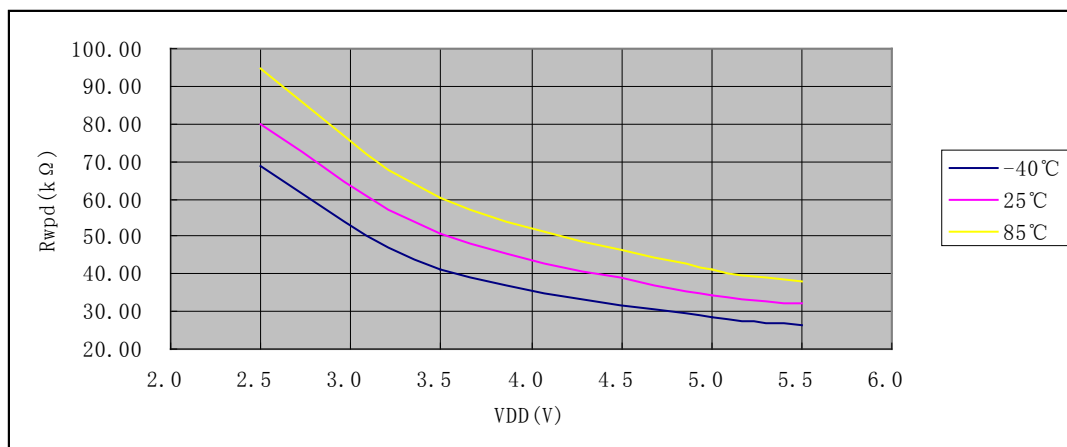


注 1: 测量 VIHmin 时, IO 端口输入电平由 VDD 开始下降, 直到端口输入采样到逻辑值 0;  
 注 2: VIHmin2 适用于 PB23~PB16, 且 GPIO\_PBSMITSEL[7:0]=8'b1111\_1111 时的情况;  
 注 3: VIHmin1 适用于 PB23~PB16, 且 GPIO\_PBSMITSEL[7:0]=8'b0000\_0000 时的情况, 以及其它不支持施密特迟滞窗口可选的 IO 端口。

◆ I/O 端口弱上拉电阻随电压-温度变化特性图

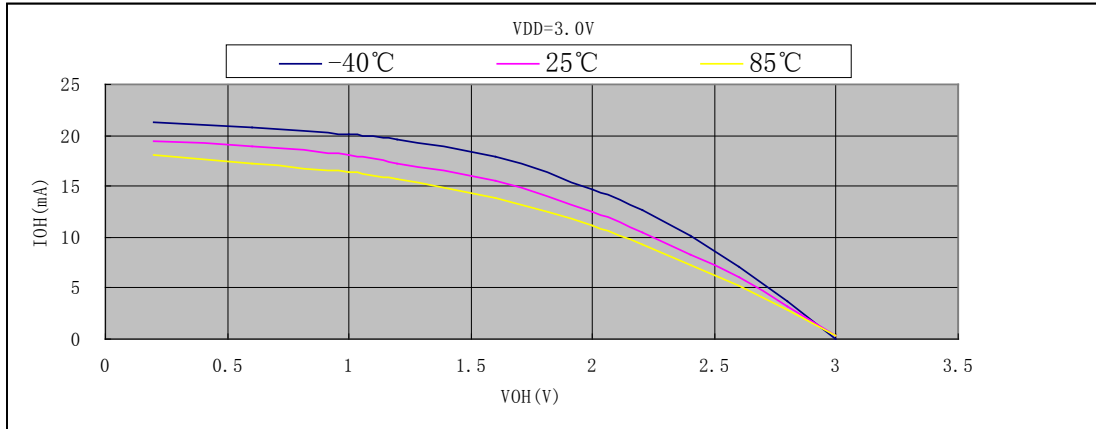


◆ I/O 端口弱下拉电阻随电压-温度变化特性图

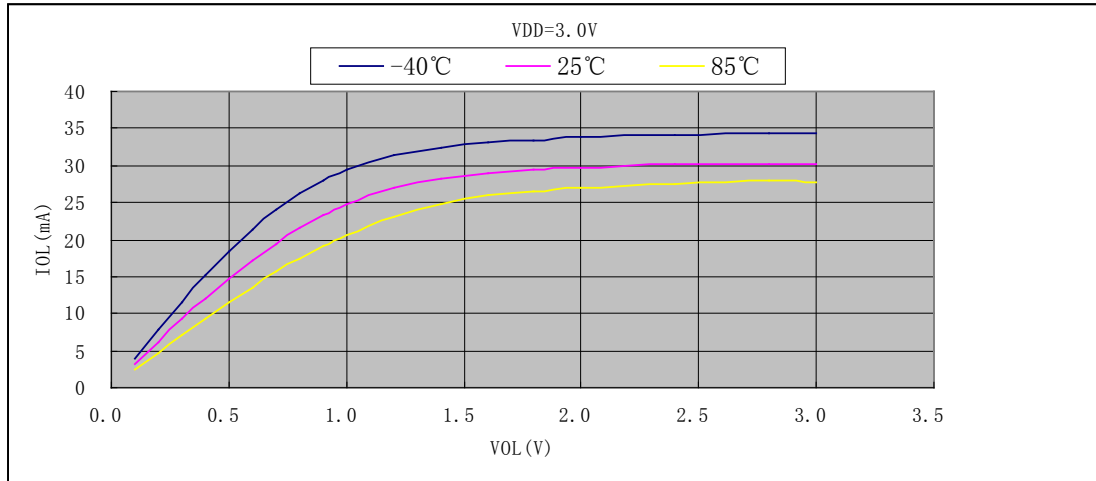


附录2. 2. 3 芯片IO端口输出特性（普通驱动）

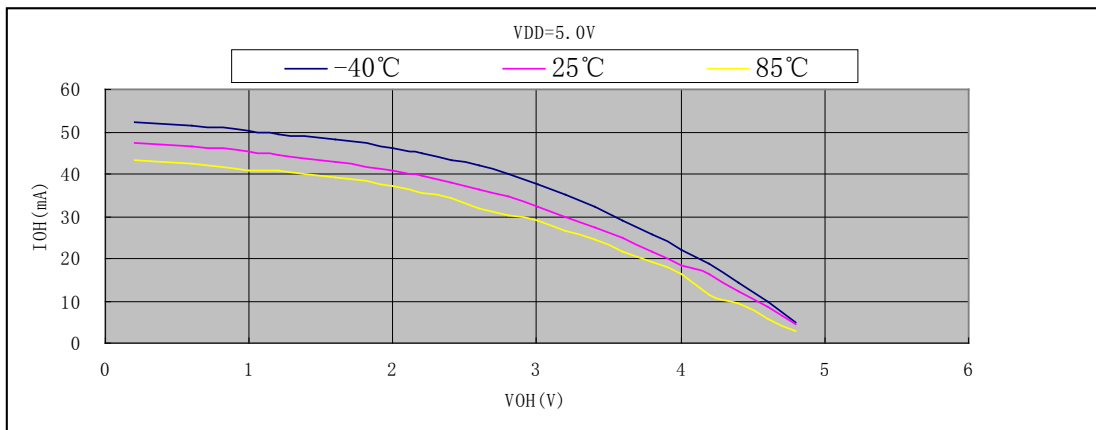
◆  $V_{OH}$  vs  $I_{OH}$ @VDD=3.0V（普通驱动端口）



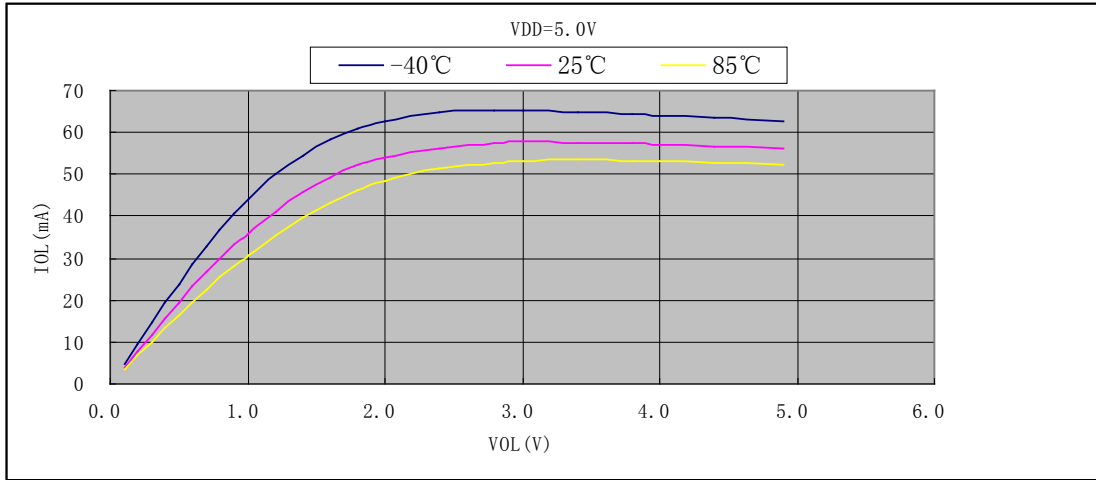
◆  $V_{OL}$  vs  $I_{OL}$ @VDD=3.0V（普通驱动端口）



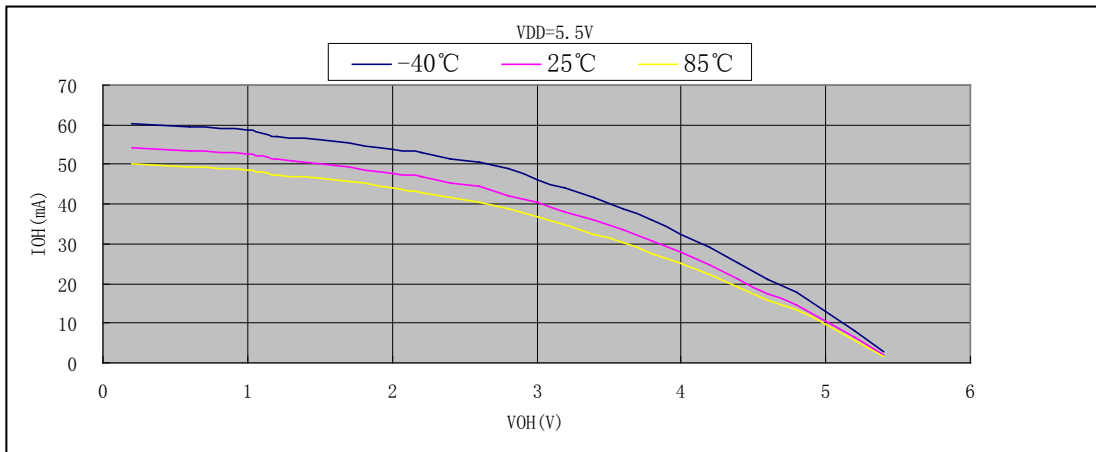
◆  $V_{OH}$  vs  $I_{OH}$ @VDD=5.0V（普通驱动端口）



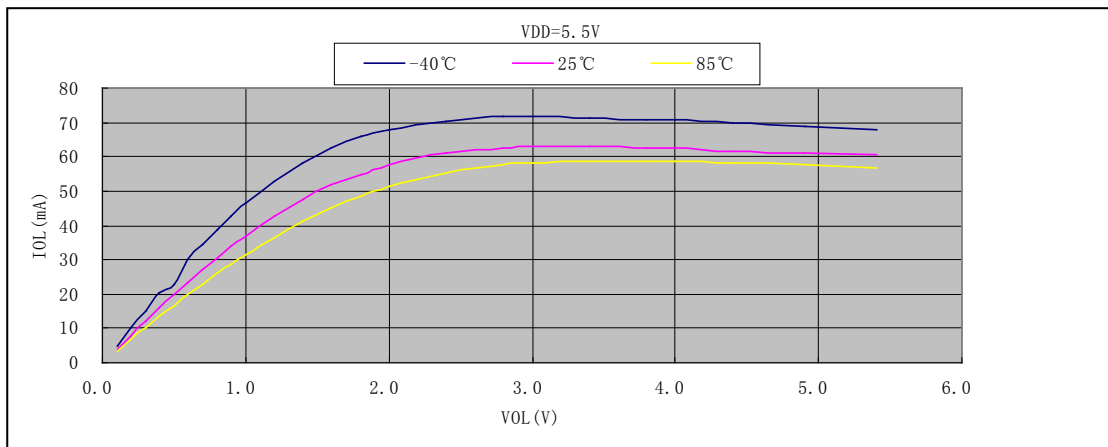
◆  $V_{OL}$  vs  $I_{OL}$ @VDD=5.0V (普通驱动端口)



◆  $V_{OH}$  vs  $I_{OH}$ @VDD=5.5V (普通驱动端口)

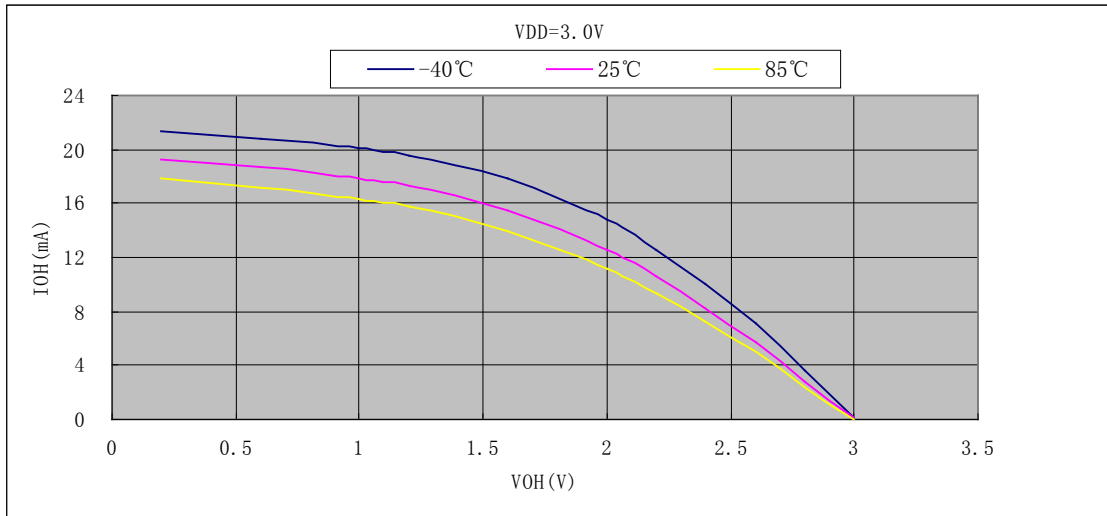


◆  $V_{OL}$  vs  $I_{OL}$ @VDD=5.5V (普通驱动端口)

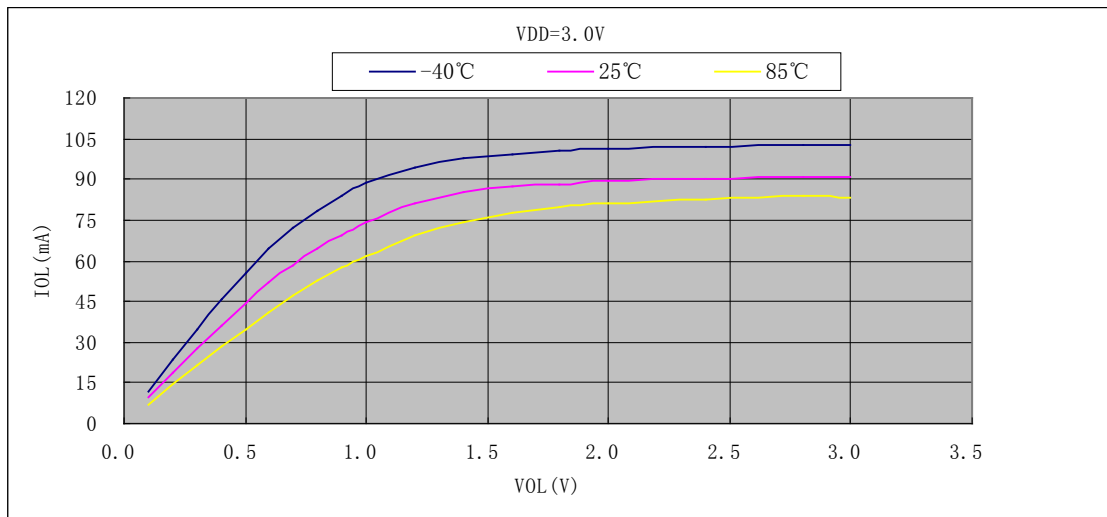


**附录2.2.4 芯片IO端口输出特性（大电流驱动）**

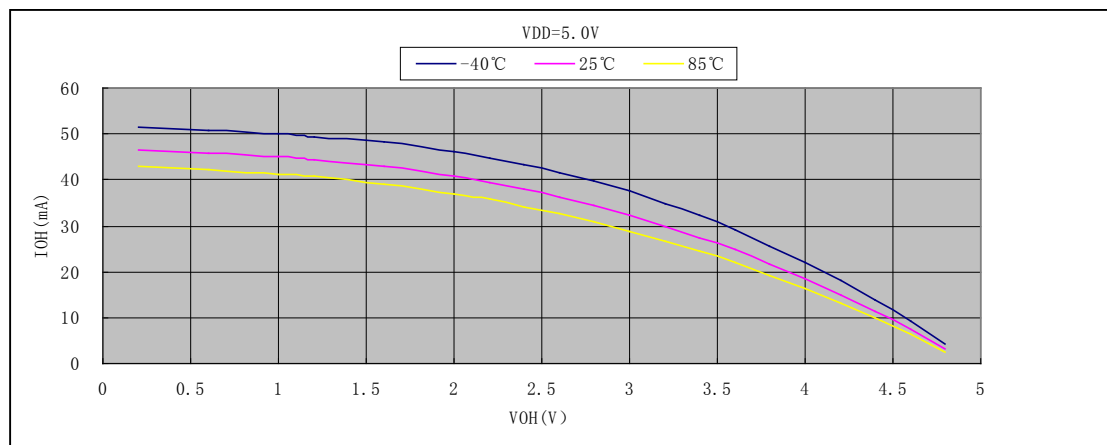
◆  $V_{OH}$  vs  $I_{OH}$ @ $V_{DD}=3.0V$ （大电流驱动端口）



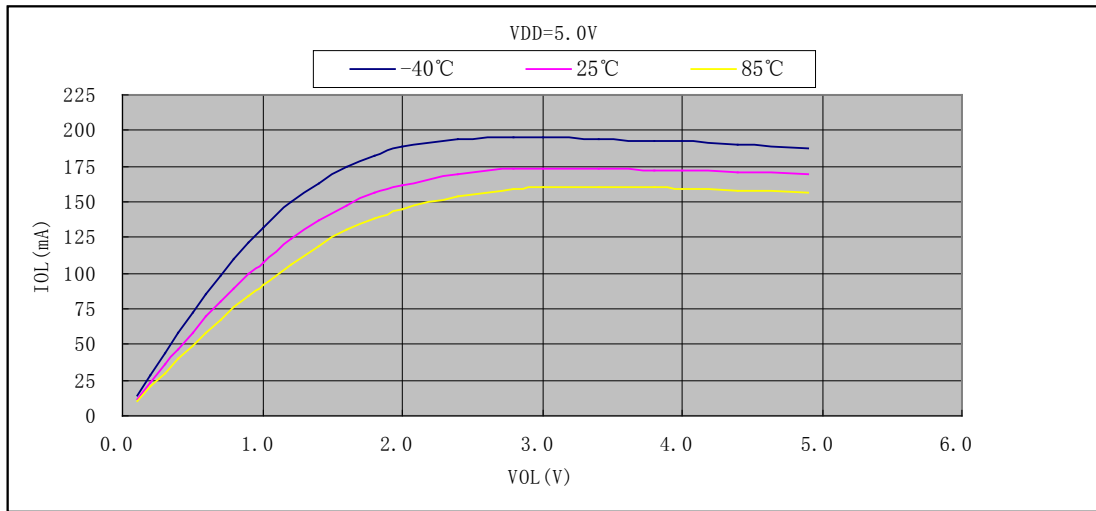
◆  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=3.0V$ （大电流驱动端口）



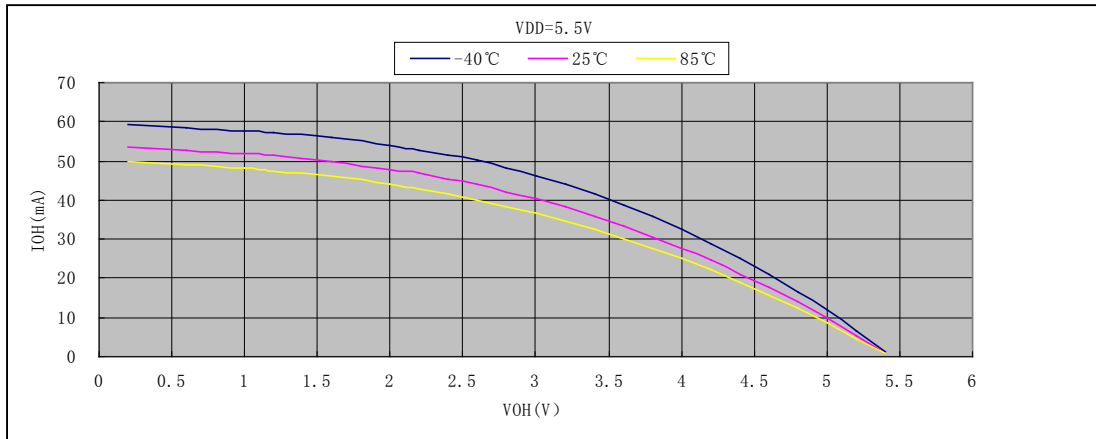
◆  $V_{OH}$  vs  $I_{OH}$ @ $V_{DD}=5.0V$ （大电流驱动端口）



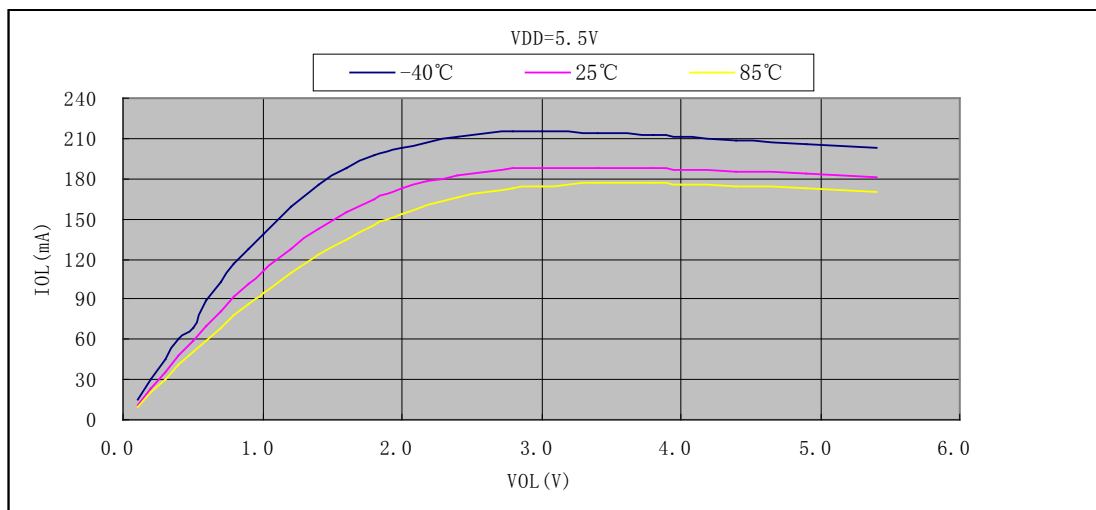
◆  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=5.0V$  (大电流驱动端口)



◆  $V_{OH}$  vs  $I_{OH}$ @ $V_{DD}=5.5V$  (大电流驱动端口)



◆  $V_{OL}$  vs  $I_{OL}$ @ $V_{DD}=5.5V$  (大电流驱动端口)



## 附录3 编程调试接口

### 附录3.1 概述

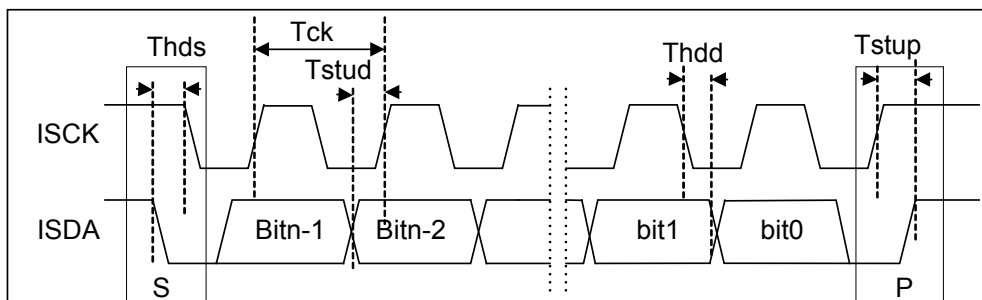
为方便应用程序及实际系统调试，芯片内部集成 ISP 在线编程接口和 SWD 串行调试接口，通过上海东软载波微电子有限公司授权的 ISP 编程器、SWD 调试器可实现芯片在线编程、仿真调试功能。

芯片 ISP 和 SWD 功能模块共用 5 线接口配置，即电源线 VDD、地线 VSS、复位线 MRSTN、时钟线 ISCK 和数据线 ISDA。

### 附录3.2 ISP编程接口

#### 附录3.2.1 通信协议

ISP 接口协议采用两线制半双工通信协议，每个信息块包含 n 个数据信息位，以及起始位和停止位。烧录器为主控设备，ISCK 时钟由烧录器发送；芯片为受控设备。如下图示：



注：ISCK为高电平时ISDA的下降沿为起始位

ISCK为高电平时ISDA的上升沿为停止位

Thds: 起始位SCK保持时间

Tck: 串行时钟周期

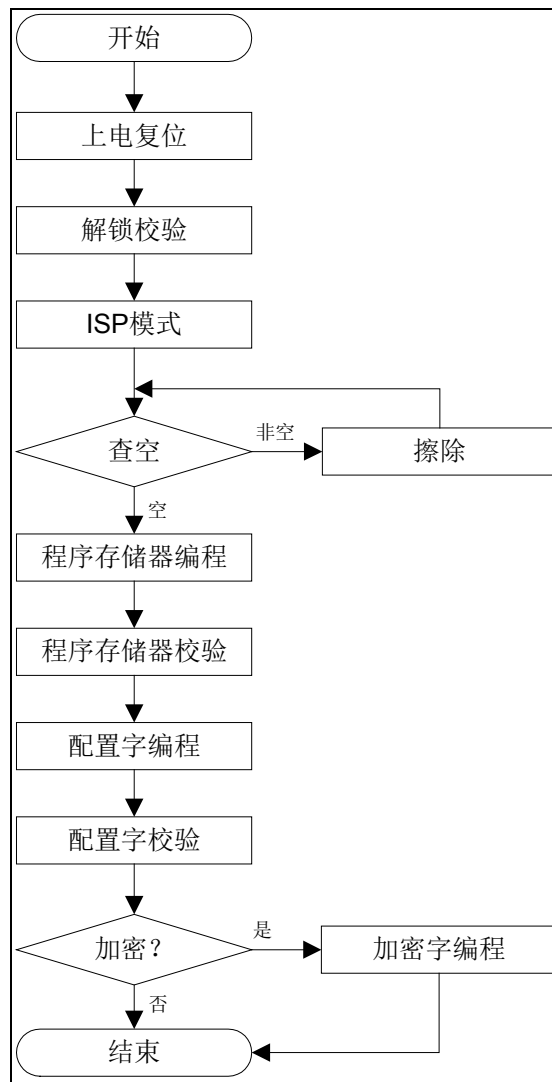
Tstud: 串行数据建立时间

Thdd: 串行数据保持时间

Tstup: 停止位SCK建立时间

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

### 附录3.2.2 操作流程





## 附录3.3 SWD调试接口

### 附录3.3.1 概述

SWD 是 Cortex-M0 内核自带的串行调试接口，与 ARM 的 CoreSight 调试技术兼容。

芯片通过 SWD 调试器完成调试程序（需通过配置字控制位 DEBUG，使能调试模式）下载，然后重新上电，芯片的 SWDIO（复用为 ISDA）和 SWCLK（复用为 ISCK）端口功能可用。

SWCLK：串行时钟输入端口，提供 SWD 串行通讯时钟。

SWDIO：串行数据输入/输出端口。

### 附录3.3.2 SWD特性

SWD 调试功能可分为侵入式调试和非侵入式调试两部分。

#### 侵入式调试

- ◇ 停机
- ◇ 单步执行
- ◇ 硬件断点（支持 4 个硬件断点）
- ◇ 软件断点（支持 BKPT 指令）
- ◇ 修改程序指针 PC 值
- ◇ 数据观察点 DWT  
（Data Watchpoint and Trace，只支持 Watchpoint 功能，不支持 Trace 功能）
- ◇ 内部寄存器和 RAM 存储器的读写访问操作
- ◇ 矢量捕捉（包括 Reset 和 HardFault 异常的捕捉）

#### 非侵入式调试

- ◇ 程序指针 PC 值采样

## 附录4 LCD驱动波形

### 附录4.1 概述

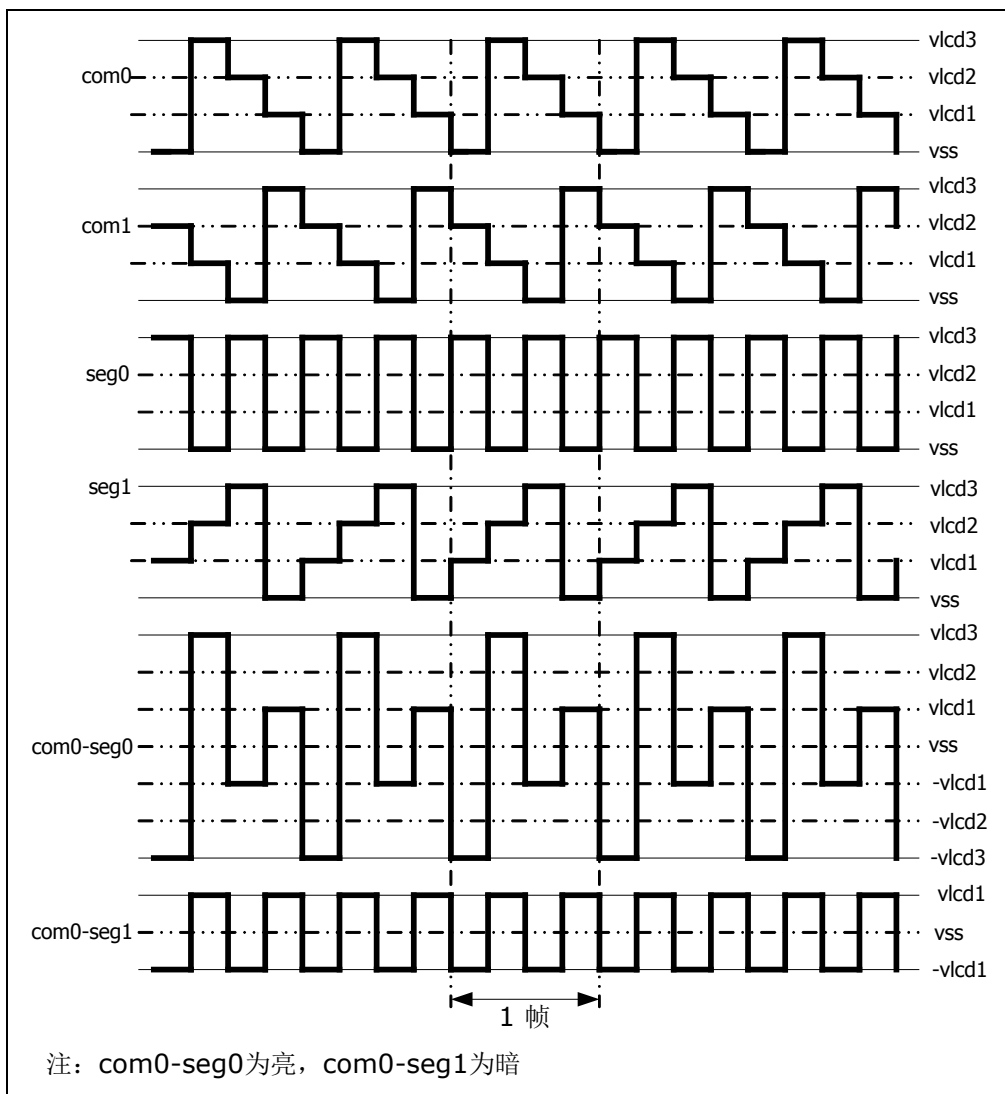
因为要产生 LCD 波形，所以不透明像素上的净 AC 电压应该是最大值，而透明像素上的净 AC 电压应该是最小值。任何像素上的净 DC 电压应该为零。COM 信号表示每个公共端的时间片，而 SEG 中包含像素数据。

像素信号 (COM-SEG) 中不包含 DC 分量，并且只可取两个 rms 值中的一个。高 rms 值会产生不透明像素，而低 rms 值会产生透明像素。随着公共端数量的增加，两个 rms 值间的差值逐渐减小。这个差值表示 LCD 显示器可具有的最大对比度。

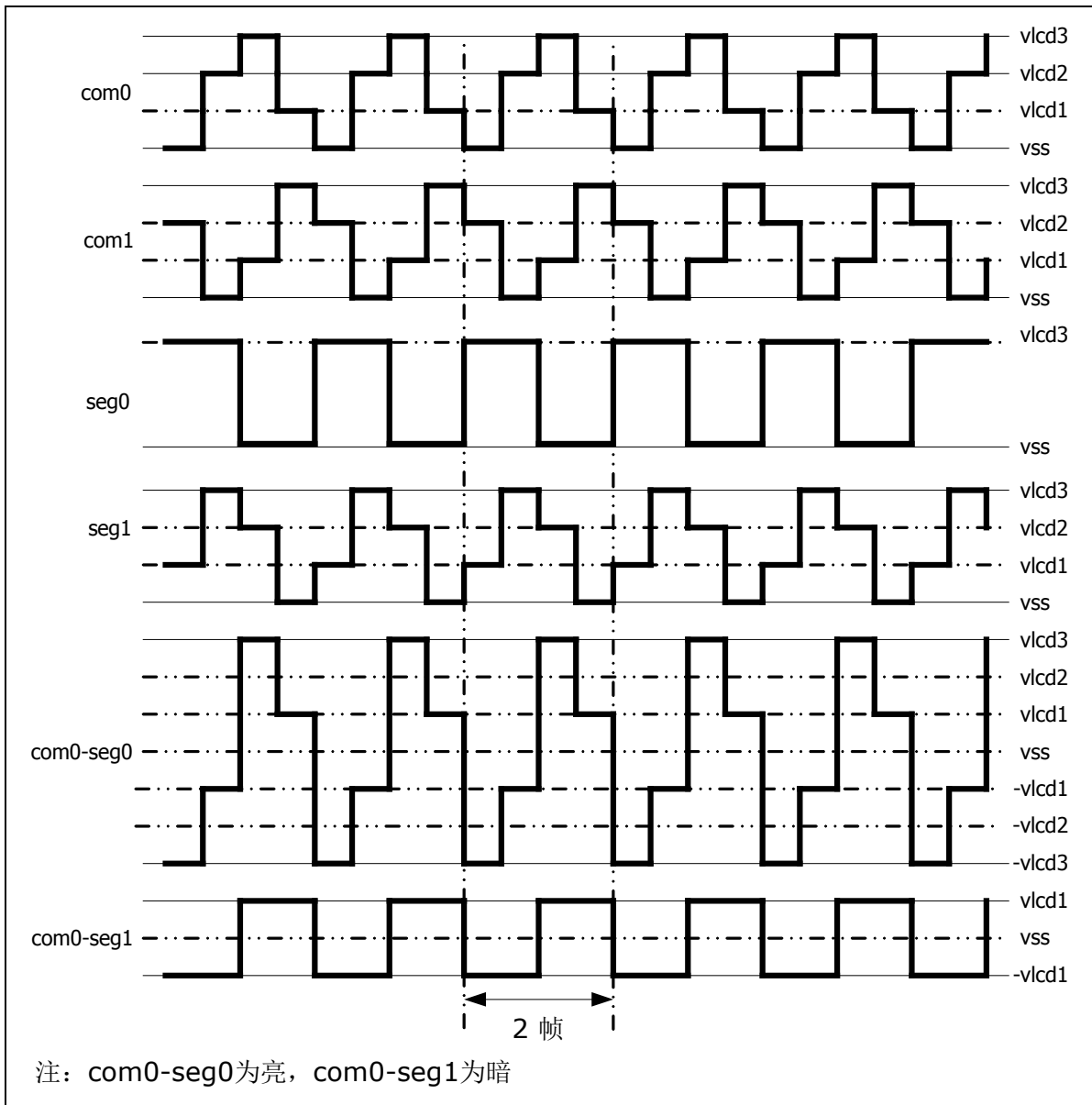
可以用两种波形驱动 LCD: A 型和 B 型。在 A 型波形中，相位在每个公共端类型中改变，在 B 型波形中，相位在每个帧边界上改变。这样，A 型波形在单帧中即可维持 0V DC 分量，而 B 型波形则需要两个帧维持 0V DC 分量。

### 附录4.2 驱动波形图

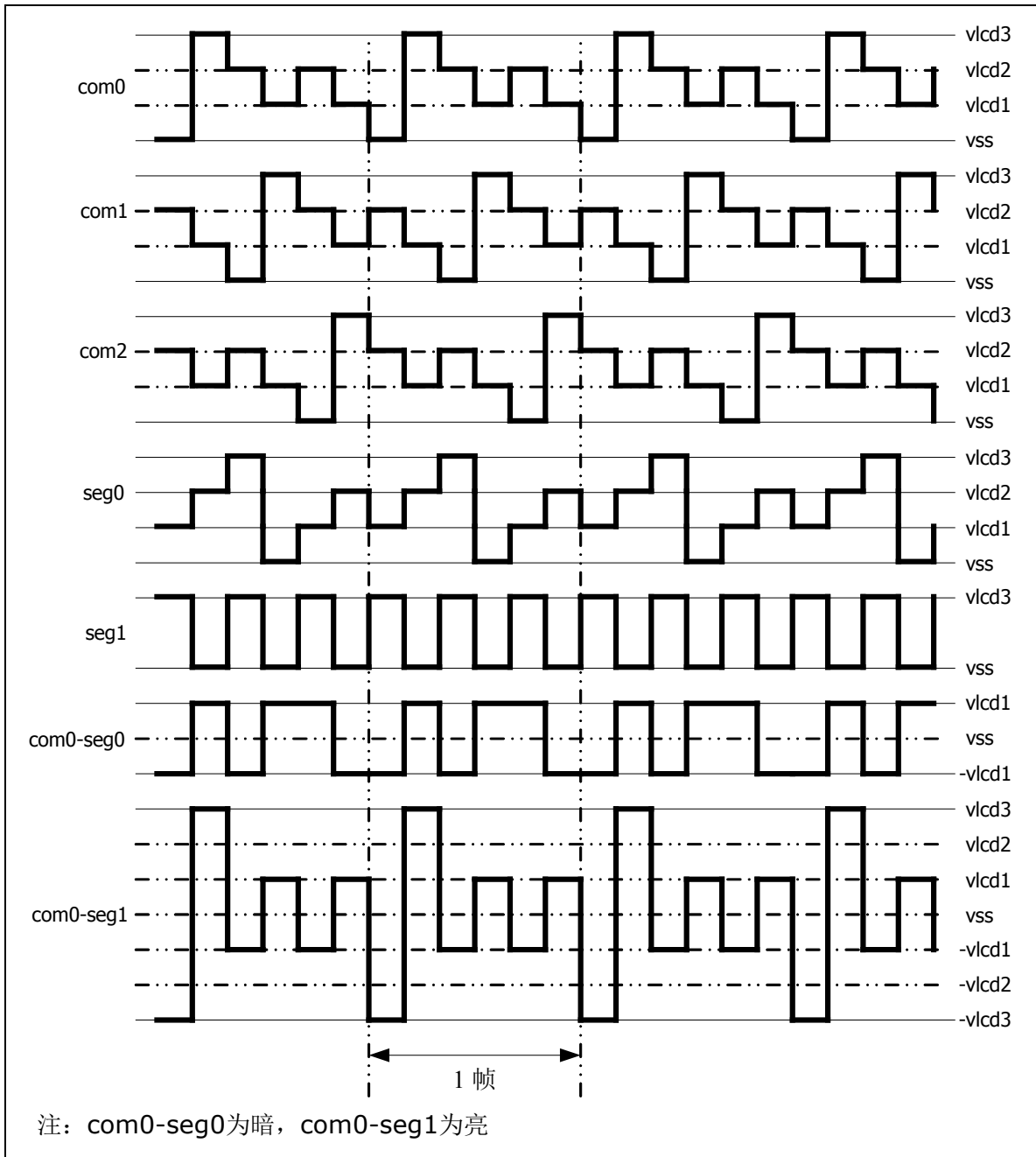
◆ 1/2 复用、1/3 偏置驱动时的 A 型波形图



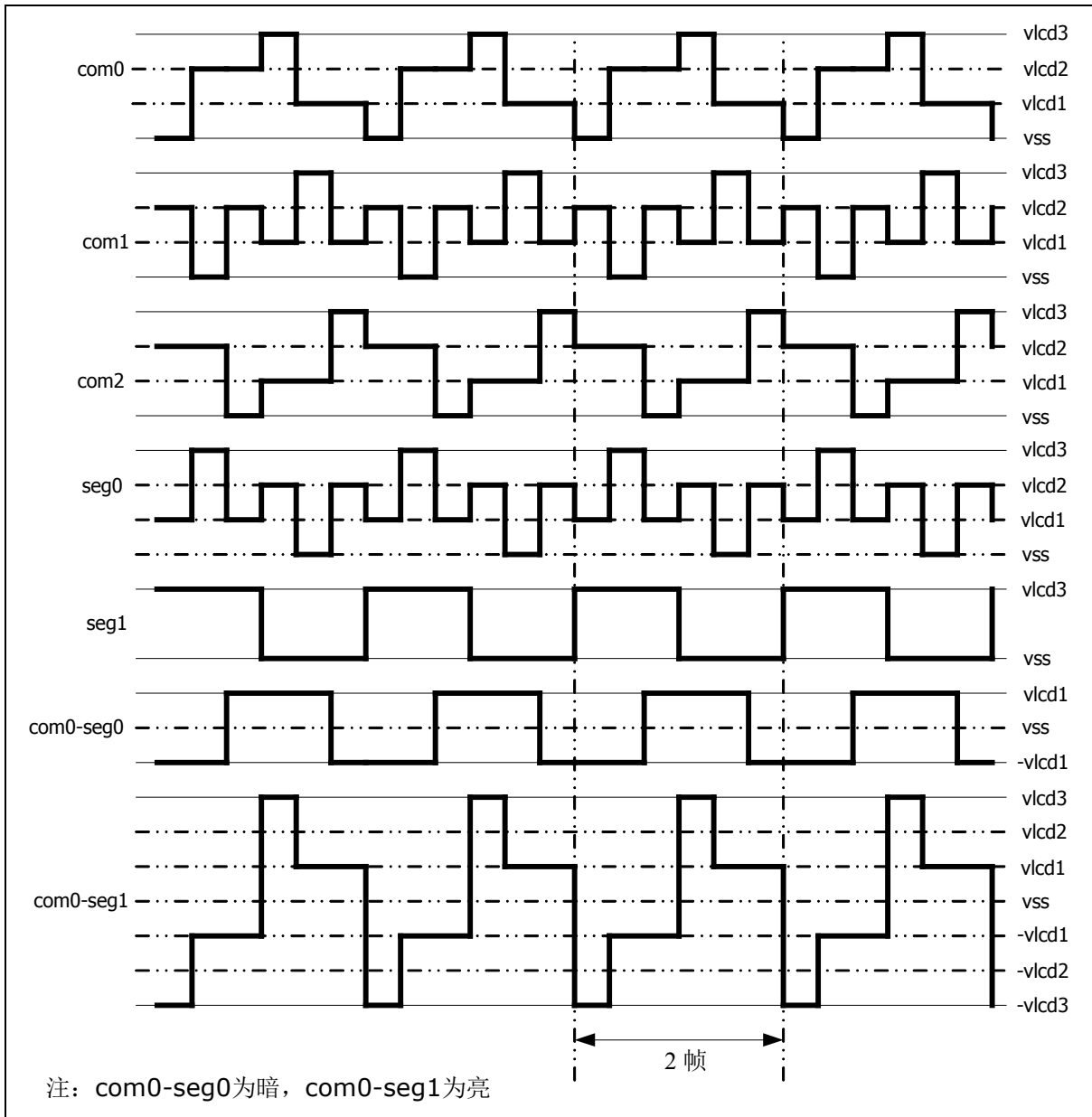
◆ 1/2 复用、1/3 偏置驱动时的 B 型波形图



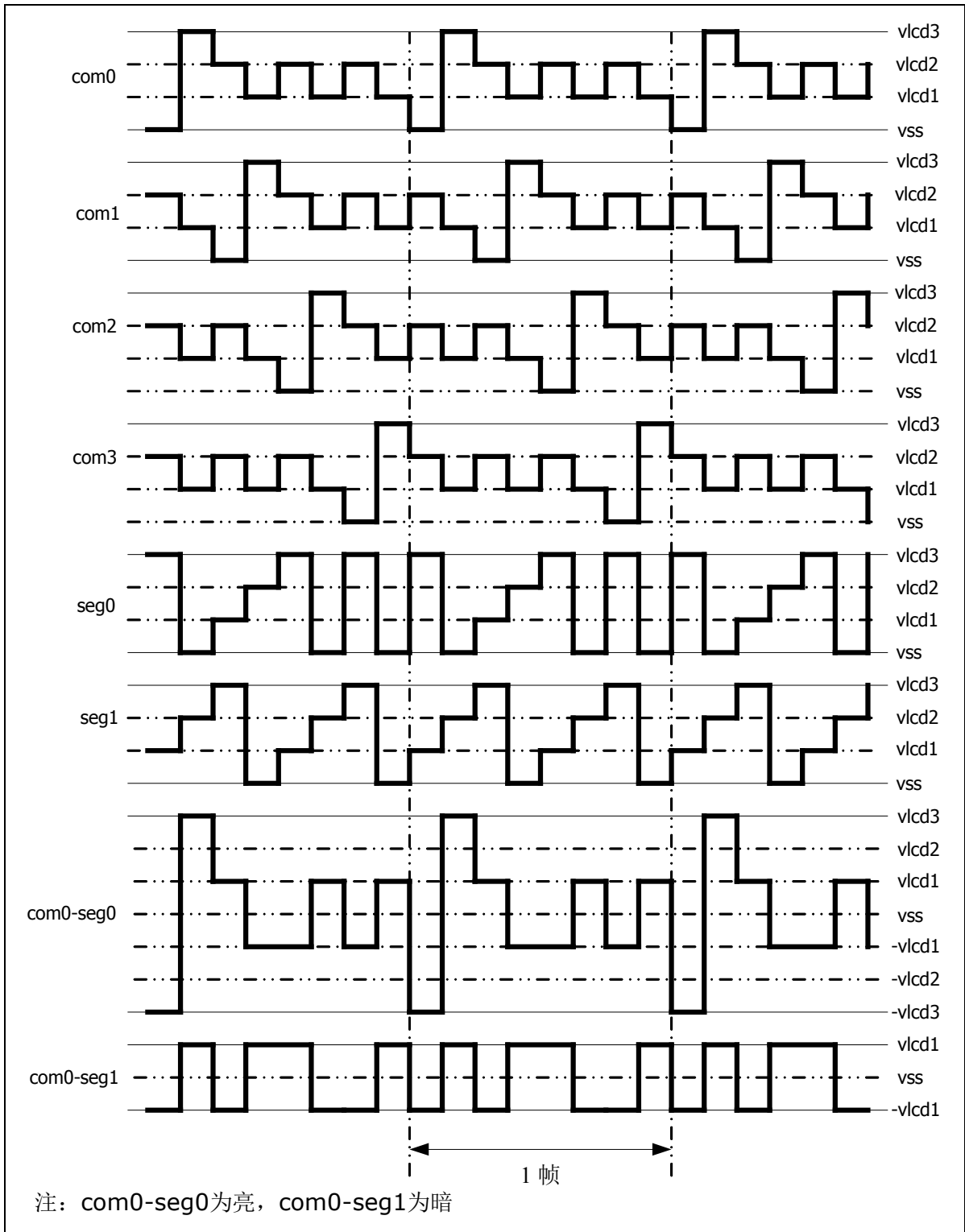
◆ 1/3 复用、1/3 偏置驱动时的 A 型波形图



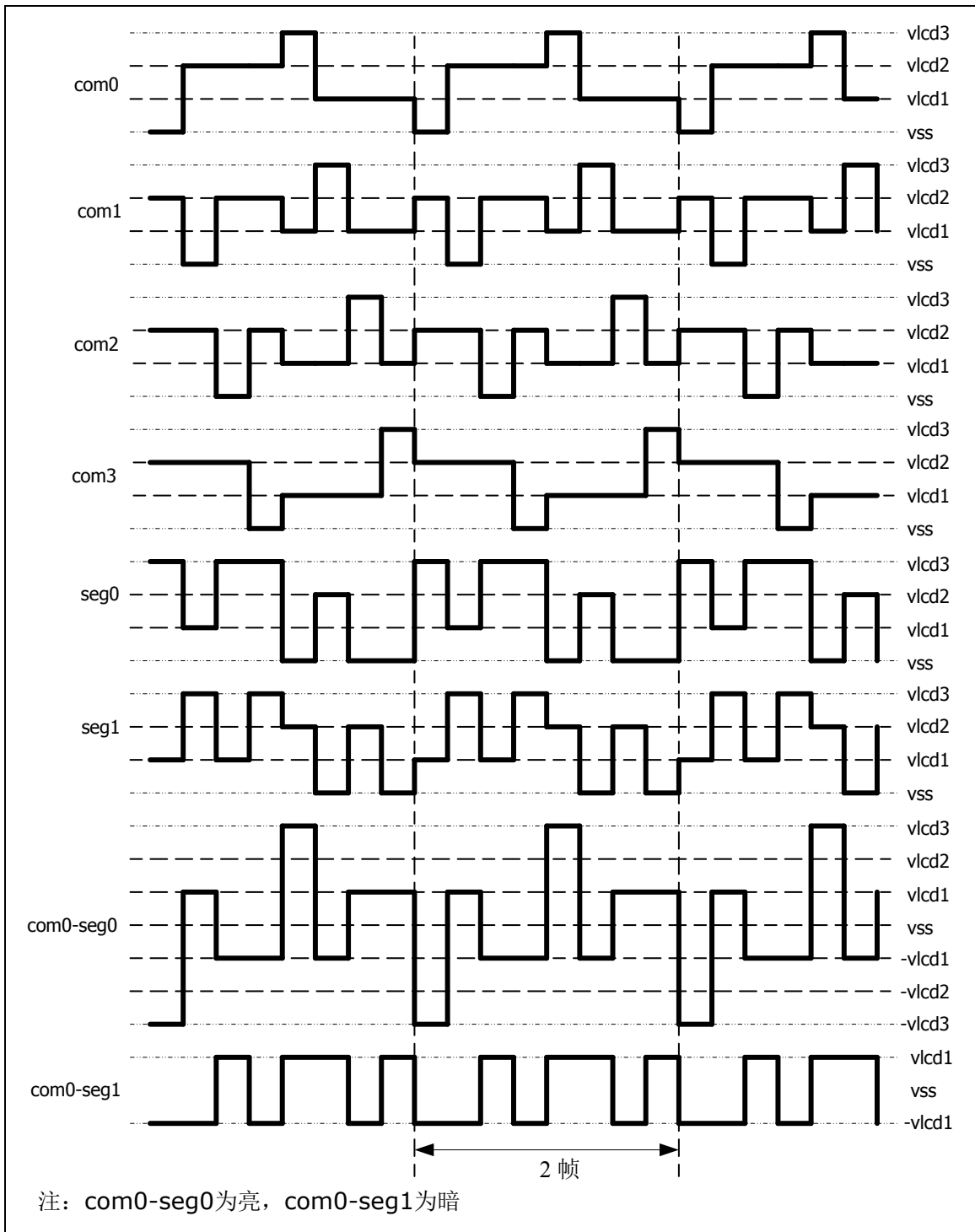
◆ 1/3 复用、1/3 偏置驱动时的 B 型波形图



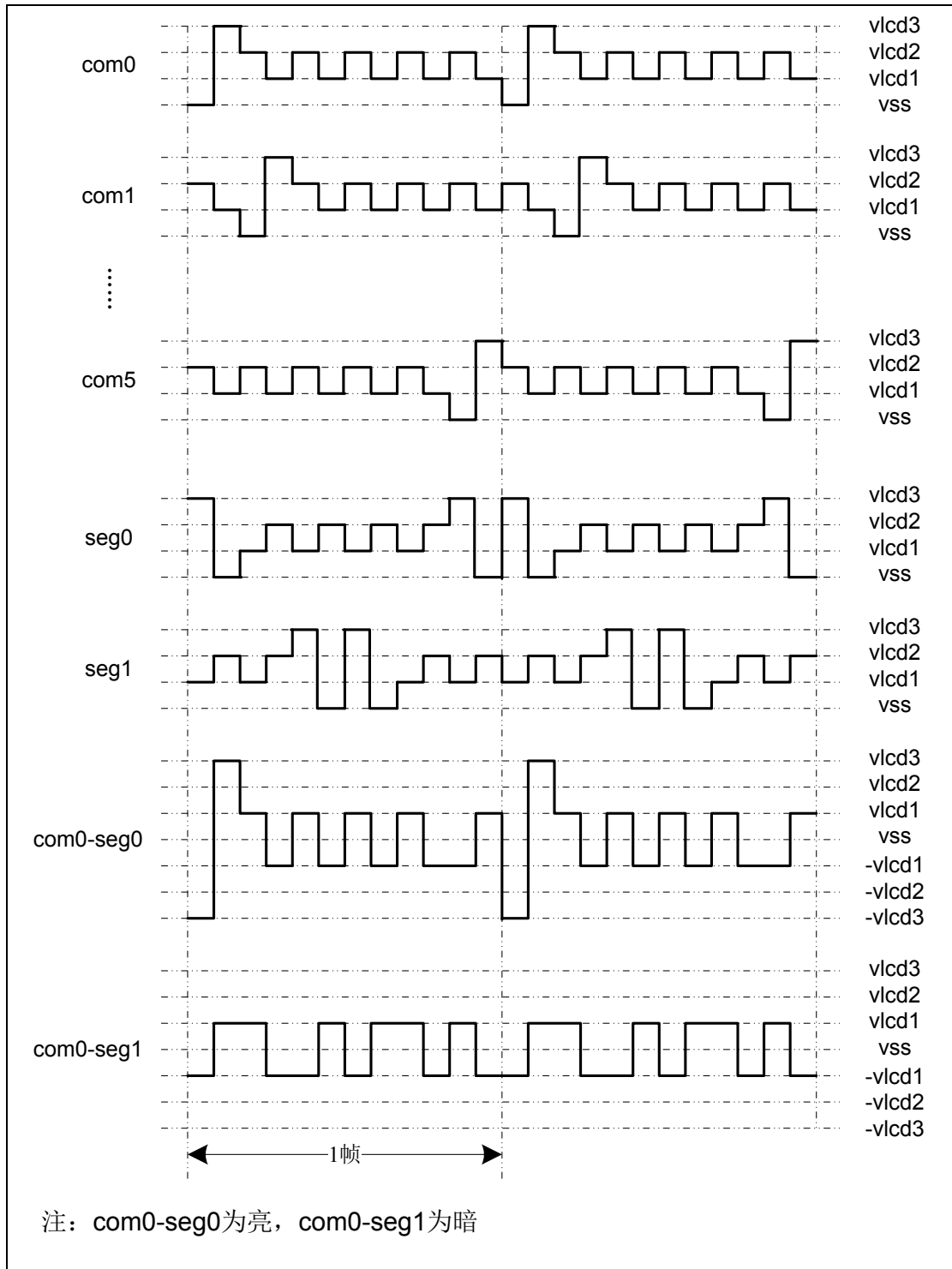
◆ 1/4 复用、1/3 偏置驱动时的 A 型波形图



◆ 1/4 复用、1/3 偏置驱动时的 B 型波形图

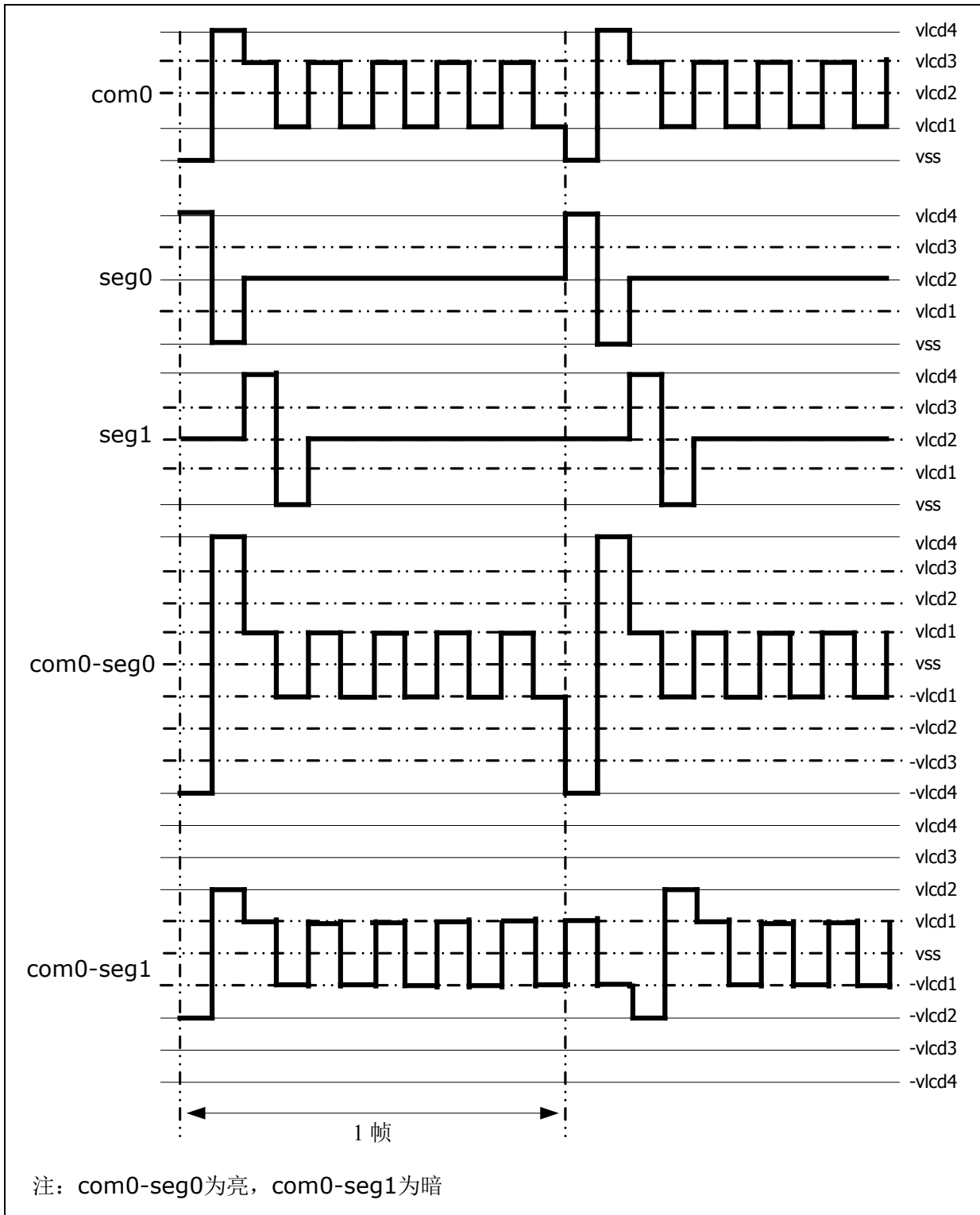


◆ 1/6 复用、1/3 偏置驱动时的 A 型波形图





◆ 1/6 复用、1/4 偏置驱动时的 A 型波形图



◆ 1/8 复用、1/4 偏置驱动时的 B 型波形图

