

SN32F240/230 系列

用户参考手册

SN32F249/239FG

SN32F248/238FG

SN32F247/237FG

SN32F246/236JG

SN32F245/235JG

SONiX 32 位 Cortex-M0 单片机

SONiX 公司保留对以下所有产品在可靠性，功能和设计方面的改进作进一步说明的权利。SONiX 不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，SONiX 的产品不是专门设计来应用于外科植入、生命维持和任何 SONiX 产品的故障会对个体造成伤害甚至死亡的领域。如果将 SONiX 的产品应用于上述领域，即使这些是由 SONiX 在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接产生的律师费用，并且用户保证 SONiX 及其雇员、子公司、分支机构和销售商与上述事宜无关。

修改记录

版本	时间	修改说明
V1.0	2014.02.27	初版。
V1.1	2014.05.23	1、修改书写错误。 2、更新 23.3 标记示例部分。 3、更新 SN-LINK-V2 图片。
V1.2	2014.07.17	1、修改书写错误。
V1.3	2014.10.08	1、修改运行环境温度。 2、修改书写错误 USB_CFG.27(ESD_EN)。 3、修改书写错误 USB_EP0CTL[5: 0]。
V1.4	2015.08.24	1、更新电气特性，USB 模式运行电压 Vdd2 为 3.1V。 2、更新 QFN46 封装信息。 3、使能 ADC 功能时增加一条 P2.n GPIO 设置的注释。 4、更新 UART 波特率的示例说明。 5、更新 USB 功能使能的 IHRC 频率。 6、更新 PLL 推荐输入/输出频率设置表。
V1.5	2016.07.19	1、修改编程/擦除操作的系统时钟。
V1.6	2017.12.25	1、增加未引出引脚的设置注意事项。 2、更新 SN-LINK-V3。 3、移除 SN32F22X 系列。
V1.7	2018.09.19	1、修改 DP/DN 命名。 2、修改 I2Cn_TOCTR's 超时时间的书写错误。 3、修改 SN-LINK-V3 开发工具。
V1.8	2019.03.19	1、增加唯一码描述。 2、移除深度掉电模式。 3、修改 GPIO2_CFG 寄存器，增加输入 Gating 功能，替换 repeat 模式。 4、修改电气特性 FADCLK 为 16MHz。
V1.9	2019.04.18	1、修改 22 章 FLASH ROM 烧录引脚。
V2.0	2019.06.13	1、更新 LVD 电气特性。
V2.1	2020.01.03	1、增加 7.3.3 ADC 引脚配置。 2、更新 23 章封装信息。 3、补充 13 章 I2C。 4、增加 1.4 章引脚配置 BOOT 引脚注释。 5、增加 2.6 章唯一码介绍。
V2.2	2023.05.17	1、修改 1.6 章引脚电路结构图； 2、更新 3.1.4.1 章 RC 复位电路； 3、修改 3.4.1 章 AHB 时钟使能寄存器描述； 4、修改 10.3.1 章 WDT_CFG 书写错误； 5、更新 24 章日期码规格； 6、修改书写错误。

目录

1	产品简介	11
1.1	功能特性	11
1.2	系统框图	13
1.3	时钟产生框图	14
1.4	引脚配置	15
1.5	引脚说明	20
1.6	引脚电路结构图	23
2	中央处理器 (CPU)	25
2.1	存储器	25
2.2	系统 TICK 定时器	26
2.2.1	操作	26
2.2.2	SYSTICK 用法说明及技巧	27
2.2.3	SYSTICK 寄存器	27
2.2.3.1	系统 Tick 定时器控制和状态寄存器 (SYSTICK_CTRL)	27
2.2.3.2	系统 Tick 定时器重装值寄存器 (SYSTICK_LOAD)	27
2.2.3.3	系统 Tick 定时器当前值寄存器 (SYSTICK_VAL)	28
2.2.3.4	系统 Tick 定时器校准值寄存器 (SYSTICK_CALIB)	28
2.3	嵌套向量中断控制器 (NVIC)	29
2.3.1	中断和异常向量	29
2.3.2	NVIC 寄存器	30
2.3.2.1	IRQ0~31 中断设置使能寄存器 (NVIC_ISER)	30
2.3.2.2	IRQ0~31 中断清零使能寄存器 (NVIC_ICER)	30
2.3.2.3	IRQ0~31 中断挂起设置寄存器 (NVIC_ISPR)	30
2.3.2.4	IRQ0~31 中断挂起清零寄存器 (NVIC_ICPR)	30
2.3.2.5	IRQ0~31 中断优先级寄存器 (NVIC_IPRn) (n=0~7)	31
2.4	应用中断和复位控制 (AIRC)	31
2.5	编译选项列表 (CODE OPTION)	32
2.6	唯一码	32
2.7	核心寄存器简介	33
3	系统控制	34
3.1	复位	34
3.1.1	上电复位 (POR)	34
3.1.2	看门狗复位 (WDT RESET)	35
3.1.3	掉电复位	35
3.1.3.1	掉电复位简述	35
3.1.3.2	系统工作电压	36
3.1.3.3	掉电复位性能改进	36
3.1.4	外部复位	37
3.1.4.1	RC 复位电路	38
3.1.4.2	二极管&RC 复位电路	38
3.1.4.3	稳压二极管复位电路	39
3.1.4.4	电压偏置复位电路	39
3.1.4.5	外部 IC 复位电路	40
3.1.5	软件复位	40
3.2	系统时钟	41
3.2.1	内部 RC 时钟源	41
3.2.1.1	内部高速 RC 振荡器 (IHRC)	41
3.2.1.2	内部低速 RC 振荡器 (ILRC)	41
3.2.2	PLL	42
3.2.2.1	PLL 频率选择	42
3.2.3	外部时钟源	43
3.2.3.1	外部高速时钟 (EHS)	43
3.2.3.2	石英/陶瓷振荡器	43
3.2.3.3	外部低速时钟 (ELS)	44

3.2.3.4	CRYSTAL	44
3.2.3.5	旁路模式	45
3.2.4	系统时钟 (SYSCLK) 选择	45
3.2.5	CLOCK-OUT 功能	45
3.3	系统控制寄存器 0	46
3.3.1	模拟模块控制寄存器 (SYS0_ANBCTRL)	46
3.3.2	PLL 控制寄存器 (SYS0_PLLCTRL)	47
3.3.2.1	频率的推荐设置	48
3.3.3	时钟源状态寄存器 (SYS0_CSST)	48
3.3.4	系统时钟配置寄存器 (SYS0_CLKCFG)	49
3.3.5	AHB 时钟预分频寄存器 (SYS0_AHBCP)	49
3.3.6	系统复位状态寄存器 (SYS0_RSTST)	50
3.3.7	LVD 控制寄存器 (SYS0_LVDCTRL)	50
3.3.8	外部复位引脚控制寄存器 (SYS0_EXRSTCTRL)	51
3.3.9	SWD 引脚控制寄存器 (SYS0_SWDCTRL)	51
3.3.10	噪音检测控制寄存器 (SYS0_NDTCTRL)	51
3.3.11	噪音检测状态寄存器 (SYS0_NDTSTS)	51
3.3.12	Anti-EFT Ability 控制寄存器 (SYS0_ANTIEFT)	52
3.4	系统控制寄存器 1	52
3.4.1	AHB 时钟使能寄存器 (SYS1_AHBCLKEN)	52
3.4.2	APB 时钟预分频寄存器 0 (SYS1_APB0)	53
3.4.3	APB 时钟预分频寄存器 1 (SYS1_APB1)	55
3.4.4	APB 时钟预分频寄存器 2 (SYS1_APB2)	56
3.4.5	外设复位寄存器 (SYS1_PRST)	56
3.4.6	除法被除数寄存器 (SYS1_DIVIDEND)	57
3.4.7	除法除数寄存器 (SYS1_DIVISOR)	58
3.4.8	除法商寄存器 (SYS1_QUOTIENT)	58
3.4.9	除法余数寄存器 (SYS1_REMAINDER)	58
3.4.10	除法控制寄存器 (SYS1_DIVCTRL)	58
4	系统操作模式	59
4.1	概述	59
4.2	Normal 模式	59
4.3	低功耗模式	59
4.3.1	睡眠模式	60
4.3.2	深度睡眠模式	60
4.4	系统唤醒	61
4.4.1	概述	61
4.4.2	唤醒时间	61
4.5	PMU 状态结构图	62
4.6	操作模式 Comparison 表	62
4.7	PMU 寄存器	63
4.7.1	电源控制寄存器 (PMU_CTRL)	63
4.7.2	I/O 锁存控制寄存器 1 (PMU_LATCHCTRL1)	63
4.7.3	I/O 锁存控制寄存器 2 (PMU_LATCHCTRL2)	63
4.7.4	I/O 锁存状态寄存器 (PMU_LATCHST)	64
5	GPIO 端口	65
5.1	概述	65
5.2	GPIO 模式	65
5.3	GPIO 寄存器	66
5.3.1	GPIO Pn 数据寄存器 (GPIO _n _DATA) (n=0,1,2,3)	66
5.3.2	GPIO Pn 模式寄存器 (GPIO _n _MODE) (n=0,1,2,3)	66
5.3.3	GPIO Pn 配置寄存器 (GPIO _n _CFG) (n=0,1,3)	66
5.3.4	GPIO Pn 配置寄存器 (GPIO _n _CFG) (n=2)	68
5.3.5	GPIO Pn 中断 SENSE 寄存器 (GPIO _n _IS) (n=0,1,2,3)	69
5.3.6	GPIO Pn 中断双边沿 SENSE 寄存器 (GPIO _n _IBS) (n=0,1,2,3)	69
5.3.7	GPIO Pn 中断事件寄存器 (GPIO _n _IEV) (n=0,1,2,3)	69
5.3.8	GPIO Pn 中断使能寄存器 (GPIO _n _IE) (n=0,1,2,3)	70
5.3.9	GPIO Pn 中断源状态寄存器 (GPIO _n _RIS) (n=0,1,2,3)	70

5.3.10	GPIO Pn 中断清零寄存器 (GPIO _n _IC) (n=0,1,2,3)	70
5.3.11	GPIO Pn 位设置操作寄存器 (GPIO _n _BSET) (n=0,1,2,3)	70
5.3.12	GPIO Pn 位清零操作寄存器 (GPIO _n _BCLR) (n=0,1,2,3)	70
5.3.13	GPIO Pn 开漏控制寄存器 (GPIO _n _ODCTRL) (n=0,1,2,3)	71
6	外设功能引脚配置 (PFPA)	72
6.1	概述	72
6.2	特性	72
6.3	引脚配置表	72
6.4	PFPA 寄存器	73
6.4.1	PFPA UART 寄存器 (PFPA_UART)	73
6.4.2	PFPA I2C 寄存器 (PFPA_I2C)	74
6.4.3	PFPA SSP 寄存器 (PFPA_SSP)	75
6.4.4	PFPA I2S 寄存器 (PFPA_I2S)	77
6.4.5	PFPA CT16B0 寄存器 (PFPA_CT16B0)	78
6.4.6	PFPA CT16B1 寄存器 (PFPA_CT16B1)	79
6.4.7	PFPA CT16B2 寄存器 (PFPA_CT16B2)	80
6.4.8	PFPA CT32B0 寄存器 (PFPA_CT32B0)	81
6.4.9	PFPA CT32B1 寄存器 (PFPA_CT32B1)	82
6.4.10	PFPA CT32B2 寄存器 (PFPA_CT32B2)	83
7	14+1 通道 ADC	84
7.1	概述	84
7.2	AD 转换时间	85
7.3	ADC 控制注意事项	86
7.3.1	ADC 信号	86
7.3.2	ADC 编程	86
7.3.3	ADC 引脚配置	86
7.4	ADC 电路	87
7.5	温度传感器 (TS)	88
7.6	ADC 寄存器	89
7.6.1	ADC 管理寄存器 (ADC_ADM)	89
7.6.2	ADC 数据寄存器 (ADC_ADB)	90
7.6.3	P2 控制寄存器 (ADC_P2CON)	90
7.6.4	ADC 中断使能寄存器 (ADC_IE)	91
7.6.5	ADC 中断源状态寄存器 (ADC_RIS)	91
8	16 位带捕获功能定时器	92
8.1	概述	92
8.2	特性	92
8.3	引脚说明	92
8.4	框图	93
8.5	定时器操作	94
8.5.1	边沿对齐向上计数模式	94
8.5.2	边沿对齐向下计数模式	95
8.5.3	中间对齐计数模式	95
8.6	PWM	96
8.6.1	PWM 模式 1	96
8.6.2	PWM 模式 2	97
8.7	CT16Bn 寄存器	98
8.7.1	CT16Bn 定时器控制寄存器 (CT16Bn_TMRCTRL) (n=0,1,2)	98
8.7.2	CT16Bn 定时器计数器寄存器 (CT16Bn_TC) (n=0,1,2)	98
8.7.3	CT16Bn 前置分频寄存器 (CT16Bn_PRE) (n=0,1,2)	98
8.7.4	CT16Bn 前置分频计数器寄存器 (CT16Bn_PC) (n=0,1,2)	99
8.7.5	CT16Bn 计数控制寄存器 (CT16Bn_CNTCTRL) (n=0,1,2)	99
8.7.6	CT16Bn 匹配控制寄存器 (CT16Bn_MCTRL) (n=0,1,2)	100
8.7.7	CT16Bn 匹配寄存器 0~3 (CT16Bn_MR0~3) (n=0,1,2)	100
8.7.8	CT16Bn 捕获控制寄存器 (CT16Bn_CAPCTRL) (n=0,1,2)	101
8.7.9	CT16Bn 捕获 0 寄存器 (CT16Bn_CAP0) (n=0,1,2)	101
8.7.10	CT16Bn 外部匹配寄存器 (CT16Bn_EM) (n=0,1,2)	102
8.7.11	CT16Bn PWM 控制寄存器 (CT16Bn_PWMCTRL) (n=0,1,2)	103

8.7.12	CT16Bn 定时器中断源状态寄存器 (CT16Bn_RIS) (n=0,1,2)	104
8.7.13	CT16Bn 定时器中断清零寄存器 (CT16Bn_IC) (n=0,1,2)	104
9	32 位带捕获功能定时器	105
9.1	概述	105
9.2	特性	105
9.3	引脚说明	105
9.4	框图	106
9.5	定时器操作	107
9.5.1	边沿对齐向上计数模式	107
9.5.2	边沿对齐向下计数模式	108
9.5.3	中间对齐计数模式	108
9.6	PWM	109
9.6.1	PWM 模式 1	109
9.6.2	PWM 模式 2	110
9.7	CT32Bn 寄存器	111
9.7.1	CT32Bn 定时器控制寄存器 (CT32Bn_TMRCTRL) (n=0,1,2)	111
9.7.2	CT32Bn 定时器计数器寄存器 (CT32Bn_TC) (n=0,1,2)	111
9.7.3	CT32Bn 前置分频寄存器 (CT32Bn_PRE) (n=0,1,2)	111
9.7.4	CT32Bn 前置分频计数器寄存器 (CT32Bn_PC) (n=0,1,2)	112
9.7.5	CT32Bn 计数控制寄存器 (CT32Bn_CNTCTRL) (n=0,1,2)	112
9.7.6	CT32Bn 匹配控制寄存器 (CT32Bn_MCTRL) (n=0,1,2)	113
9.7.7	CT32Bn 匹配寄存器 0~3 (CT32Bn_MR0~3) (n=0,1,2)	113
9.7.8	CT32Bn 捕获控制寄存器 (CT32Bn_CAPCTRL) (n=0,1,2)	114
9.7.9	CT32Bn 捕获 0 寄存器 (CT32Bn_CAP0) (n=0,1,2)	114
9.7.10	CT32Bn 外部匹配寄存器 (CT32Bn_EM) (n=0,1,2)	115
9.7.11	CT32Bn PWM 控制寄存器 (CT32Bn_PWMCTRL) (n=0,1,2)	116
9.7.12	CT32Bn 定时器中断源状态寄存器 (CT32Bn_RIS) (n=0,1,2)	117
9.7.13	CT32Bn 定时器中断清零寄存器 (CT32Bn_IC) (n=0,1,2)	117
10	看门狗定时器 (WDT)	118
10.1	概述	118
10.2	框图	119
10.3	WDT 寄存器	120
10.3.1	看门狗配置寄存器 (WDT_CFG)	120
10.3.2	看门狗时钟源寄存器 (WDT_CLKSOURCE)	120
10.3.3	看门狗定时器常量寄存器 (WDT_TC)	121
10.3.4	看门狗 Feed 寄存器 (WDT_FEED)	121
11	实时时钟 (RTC)	122
11.1	概述	122
11.2	特性	122
11.3	功能描述	122
11.3.1	简介	122
11.3.2	RTC 复位寄存器	122
11.3.3	RTC 标志说明	122
11.3.4	RTC 操作	123
11.4	框图	124
11.5	RTC 寄存器	125
11.5.1	RTC 控制寄存器 (RTC_CTRL)	125
11.5.2	RTC 时钟源选择寄存器 (RTC_CLKS)	125
11.5.3	RTC 中断使能寄存器 (RTC_IE)	125
11.5.4	RTC 中断源状态寄存器 (RTC_RIS)	126
11.5.5	RTC 中断清零寄存器 (RTC_IC)	126
11.5.6	RTC 秒值计数器重装值寄存器 (RTC_SECCNTV)	126
11.5.7	RTC 秒值计数寄存器 (RTC_SECCNT)	126
11.5.8	RTC 报警计数器重装值寄存器 (RTC_ALMCNTV)	127
11.5.9	RTC 报警计数寄存器 (RTC_ALMCNT)	127
12	SPI/SSP	128
12.1	概述	128
12.2	特性	128

12.3	引脚说明	128
12.4	接口说明	129
12.4.1	SPI	129
12.4.2	SSI	130
12.4.3	通讯流程	130
12.4.3.1	单帧	130
12.4.3.2	多帧	131
12.5	AUTO-SEL	131
12.6	SSP 寄存器	132
12.6.1	SSP n 控制寄存器 0 (SSPn_CTRL0) (n=0, 1)	132
12.6.2	SSP n 控制寄存器 1 (SSPn_CTRL1) (n=0, 1)	133
12.6.3	SSP n 时钟分频寄存器 (SSPn_CLKDIV) (n=0, 1)	133
12.6.4	SSP n 状态寄存器 (SSPn_STAT) (n=0, 1)	133
12.6.5	SSP n 中断使能寄存器 (SSPn_IE) (n=0, 1)	134
12.6.6	SSP n 中断源状态寄存器 (SSPn_RIS) (n=0, 1)	134
12.6.7	SSP n 中断清零寄存器 (SSPn_IC) (n=0, 1)	134
12.6.8	SSP n 数据寄存器 (SSPn_DATA) (n=0, 1)	135
12.6.9	SSP n 数据 Fetch 寄存器 (SSPn_DF) (n=0,1)	135
13	I2C	136
13.1	概述	136
13.2	特性	136
13.3	引脚说明	137
13.4	I2C 协议	137
13.4.1	7 位寻址模式	138
13.4.1.1	主机发送模式	138
13.4.1.2	主机接收模式	138
13.4.1.3	从机发送模式	139
13.4.1.4	从机接收模式	139
13.4.2	10 位寻址模式	139
13.4.2.1	主机发送模式	139
13.4.2.2	主机接收模式	139
13.5	仲裁	140
13.6	时钟伸展	140
13.7	广播呼叫地址	140
13.8	时序特性	141
13.8.1	主机发送模式	141
13.8.2	从机发送模式	141
13.9	监控模式	142
13.9.1	中断	142
13.9.2	仲裁丢失	142
13.10	I2C 寄存器	143
13.10.1	I2C n 控制寄存器 (I2Cn_CTRL) (n=0,1)	143
13.10.2	I2C n 状态寄存器 (I2Cn_STAT) (n=0,1)	144
13.10.3	I2C n TX 数据寄存器 (I2Cn_TXDATA) (n=0,1)	145
13.10.4	I2C n RX 数据寄存器 (I2Cn_RXDATA) (n=0,1)	145
13.10.5	I2C n 从机地址 0 寄存器 (I2Cn_SLVADDR0) (n=0,1)	145
13.10.6	I2C n 从机地址 1~3 寄存器 (I2Cn_SLVADDR1~3) (n=0,1)	145
13.10.7	I2C n SCL 高电平时间寄存器 (I2Cn_SCLHT) (n=0,1)	146
13.10.8	I2C n SCL 低电平时间寄存器 (I2Cn_SCLLT) (n=0,1)	146
13.10.9	I2C n 超时控制寄存器 (I2Cn_TOCTRL) (n=0,1)	146
13.10.10	I2C n 监控模式控制寄存器 (I2Cn_MMCTRL) (n=0,1)	147
14	通用同步异步串行收发器 (USART)	148
14.1	概述	148
14.2	特性	148
14.3	引脚说明	148
14.4	框图	149
14.5	EIA-485/RS-485 模式	150
14.5.1	RS-485/EIA-485 普通多支路模式 (NMM)	150

14.5.2	RS-485/EIA-485 自动地址检测模式 (AAD)	150
14.5.3	RS-485/EIA-485 自动方向控制 (ADC)	150
14.5.4	RS485/EIA-485 驱动延迟时间	151
14.5.5	RS485/EIA-485 输出反向	151
14.5.6	RS485/EIA-485 帧结构图	151
14.6	波特率计算	152
14.7	调制解调器控制 (MC)	153
14.7.1	AUTO-RTS	153
14.7.2	AUTO-CTS	154
14.8	自动波特流 (AUTO-BAUD FLOW)	155
14.8.1	AUTO-BAUD	155
14.8.2	AUTO-BAUD 模式	156
14.9	智能卡模式	157
14.9.1	智能卡设置流程	157
14.10	同步模式	158
14.11	USART 寄存器	159
14.11.1	USART n 接收缓存寄存器 (USARTn_RB) (n=0, 1)	159
14.11.2	USART n 发送保持寄存器 (USARTn_TH) (n=0, 1)	159
14.11.3	USART n 除数锁存 LSB 寄存器 (USARTn_DLL) (n=0, 1)	159
14.11.4	USART n 除数锁存 MSB 寄存器 (USARTn_DLM) (n=0, 1)	159
14.11.5	USART n 中断使能寄存器 (USARTn_IE) (n=0, 1)	160
14.11.6	USART n 中断标识寄存器 (USARTn_II) (n=0, 1)	161
14.11.7	USART n 线状态寄存器 (USARTn_LS) (n=0, 1)	162
14.11.8	USART n FIFO 控制寄存器 (USARTn_FIFOCTRL) (n=0, 1)	163
14.11.9	USART n 线控制寄存器 (USARTn_LC) (n=0, 1)	163
14.11.10	USART n 调制解调器控制寄存器 (USARTn_MC) (n=0, 1)	164
14.11.11	USART n 调制解调器状态寄存器 (USARTn_MS) (n=0, 1)	164
14.11.12	USART n 高速缓存寄存器 (USARTn_SP) (n=0, 1)	164
14.11.13	USART n 自动波特控制寄存器 (USARTn_ABCTRL) (n=0, 1)	165
14.11.14	USART n 小数分频寄存器 (USARTn_FD) (n=0, 1)	165
14.11.15	USART n 控制寄存器 (USARTn_CTRL) (n=0, 1)	166
14.11.16	USART n 半双工传输使能寄存器 (USARTn_HDEN) (n=0, 1)	167
14.11.17	USART n 智能卡接口控制寄存器 (USARTn_SCICTRL) (n=0, 1)	167
14.11.18	USART n RS485 控制寄存器 (USARTn_RS485CTRL) (n=0, 1)	168
14.11.19	USART n RS485 地址匹配寄存器 (USARTn_RS485ADRMATCH) (n=0, 1)	168
14.11.20	USART n RS485 延时寄存器 (USARTn_RS485DLYV) (n=0, 1)	168
14.11.21	USART n 同步模式控制寄存器 (USARTn_SYNCCTRL) (n=0, 1)	169
15	I2S	170
15.1	概述	170
15.2	特性	170
15.3	引脚说明	170
15.4	框图	171
15.4.1	I2S 时钟控制	171
15.4.2	I2S 框图	171
15.5	功能说明	172
15.5.1	I2S 操作	172
15.5.2	I2S FIFO 操作	173
15.5.2.1	单声道	173
15.5.2.2	立体声	173
15.6	I2S 寄存器	174
15.6.1	I2S 控制寄存器 (I2S_CTRL)	174
15.6.2	I2S 时钟寄存器 (I2S_CLK)	175
15.6.3	I2S 状态寄存器 (I2S_STATUS)	175
15.6.4	I2S 中断使能寄存器 (I2S_IE)	176
15.6.5	I2S 中断源状态寄存器 (I2S_RIS)	177
15.6.6	I2S 中断清零寄存器 (I2S_IC)	177
15.6.7	I2S RXFIFO 寄存器 (I2S_RXFIFO)	177
15.6.8	I2S TXFIFO 寄存器 (I2S_TXFIFO)	177

16	4x32 LCD 驱动.....	178
16.1	概述.....	178
16.2	特性.....	178
16.3	引脚说明.....	178
16.4	框图.....	179
16.4.1	LCD 时钟控制.....	179
16.4.2	LCD 框图.....	179
16.5	LCD 时序.....	180
16.5.1	LCD 时序列表.....	180
16.5.2	LCD 驱动波形图.....	180
16.6	R 型 LCD 应用电路.....	182
16.7	C 型 LCD 应用电路.....	184
16.7.1	4C-TYPE.....	185
16.7.2	1C-Type.....	187
16.8	LCD 显示分配图 LAY MEMORY MAP.....	188
16.9	LCD 寄存器.....	189
16.9.1	LCD 控制寄存器 (LCD_CTRL).....	189
16.9.2	LCD 控制寄存器 1 (LCD_CTRL1).....	190
16.9.3	LCD C 型控制寄存器 1 (LCD_CCTRL1).....	190
16.9.4	LCD C 型控制寄存器 2 (LCD_CCTRL2).....	191
16.9.5	LCD 帧计数器控制寄存器 (LCD_FCC).....	191
16.9.6	LCD Raw 中断状态寄存器 (LCD_RIS).....	191
16.9.7	LCD SEG 存储器寄存器 0 (LCD_SEGM0).....	191
16.9.8	LCD SEG 存储器寄存器 1 (LCD_SEGM1).....	192
16.9.9	LCD SEG 存储器寄存器 2 (LCD_SEGM2).....	192
16.9.10	LCD SEG 存储器寄存器 3 (LCD_SEGM3).....	192
17	USB FS 设备接口.....	193
17.1	概述.....	193
17.2	特性.....	193
17.3	引脚说明.....	193
17.4	框图.....	194
17.5	USB SRAM 访问.....	194
17.6	USB 机构.....	195
17.7	USB 中断.....	195
17.8	USB 枚举.....	195
17.9	USB 寄存器.....	196
17.9.1	USB 中断使能寄存器 (USB_INTEN).....	197
17.9.2	USB 中断事件状态寄存器 (USB_INSTS).....	198
17.9.3	USB 中断事件状态清零寄存器 (USB_INSTSC).....	200
17.9.4	USB 设备地址寄存器 (USB_ADDR).....	201
17.9.5	USB 配置寄存器 (USB_CFG).....	201
17.9.6	USB 信号控制寄存器 (USB_SGCTL).....	202
17.9.7	USB EP0 控制寄存器 (USB_EP0CTL).....	202
17.9.8	USB Epn 控制寄存器 (USB_EPnCTL, n = 1 ~ 6).....	203
17.9.9	USB Epn 数据 Toggle 寄存器 (USB_EPTOGGLE).....	203
17.9.10	USB Epn 缓存器偏移寄存器 (USB_EPnBUFOS, n = 1 ~ 6).....	204
17.9.11	USB 帧数字寄存器 (USB_FRMNO).....	204
17.9.12	USB PHY 参数寄存器 (USB_PHYPRM).....	204
18	FLASH.....	205
18.1	概述.....	205
18.2	嵌入式 FLASH 存储器.....	205
18.3	特性.....	205
18.4	机构.....	206
18.5	读操作.....	206
18.6	编程/擦除.....	206
18.7	嵌入式引导加载程序.....	206
18.8	FLASH 存储控制器 (FMC).....	207
18.8.1	代码加密 (CS).....	207

18.8.2	编程 FLASH 存储器	208
18.8.3	擦除操作	208
18.8.3.1	按页擦除	208
18.8.3.2	批量擦除	208
18.9	读保护	208
18.10	HW CHECKSUM	208
18.11	FMC 寄存器	209
18.11.1	Flash 低电压控制寄存器 (FLASH_LPCTRL)	209
18.11.2	Flash 状态寄存器 (FLASH_STATUS)	209
18.11.3	Flash 控制寄存器 (FLASH_CTRL)	209
18.11.4	Flash 数据寄存器 (FLASH_DATA)	210
18.11.5	Flash 地址寄存器 (FLASH_ADDR)	210
18.11.6	Flash Checksum 寄存器 (FLASH_CHKSUM)	210
19	SERIAL-WIRE 调试 (SWD)	211
19.1	概述	211
19.2	特性	211
19.3	引脚说明	211
19.4	调试注意事项	211
19.4.1	局限性	211
19.4.2	恢复调试功能	211
19.4.3	SWD 引脚上的内部上拉/下拉电阻	212
20	开发工具	213
20.1	SN-LINK	213
20.2	SN32F249 STARTER-KIT	214
21	电气特性	215
21.1	极限参数	215
21.2	电气特性	215
21.3	特性曲线图	217
22	FLASH ROM 引脚烧录	219
23	封装信息	220
23.1	LQFP 80 PIN	220
23.2	LQFP 64 PIN	221
23.3	LQFP 48 PIN	222
23.4	QFN 46 PIN	223
23.5	QFN 33 PIN 5x5	224
24	芯片正印命名规则	225
24.1	概述	225
24.2	芯片型号说明	225
24.3	命名举例	226
24.4	日期码规则	226

1 产品简介

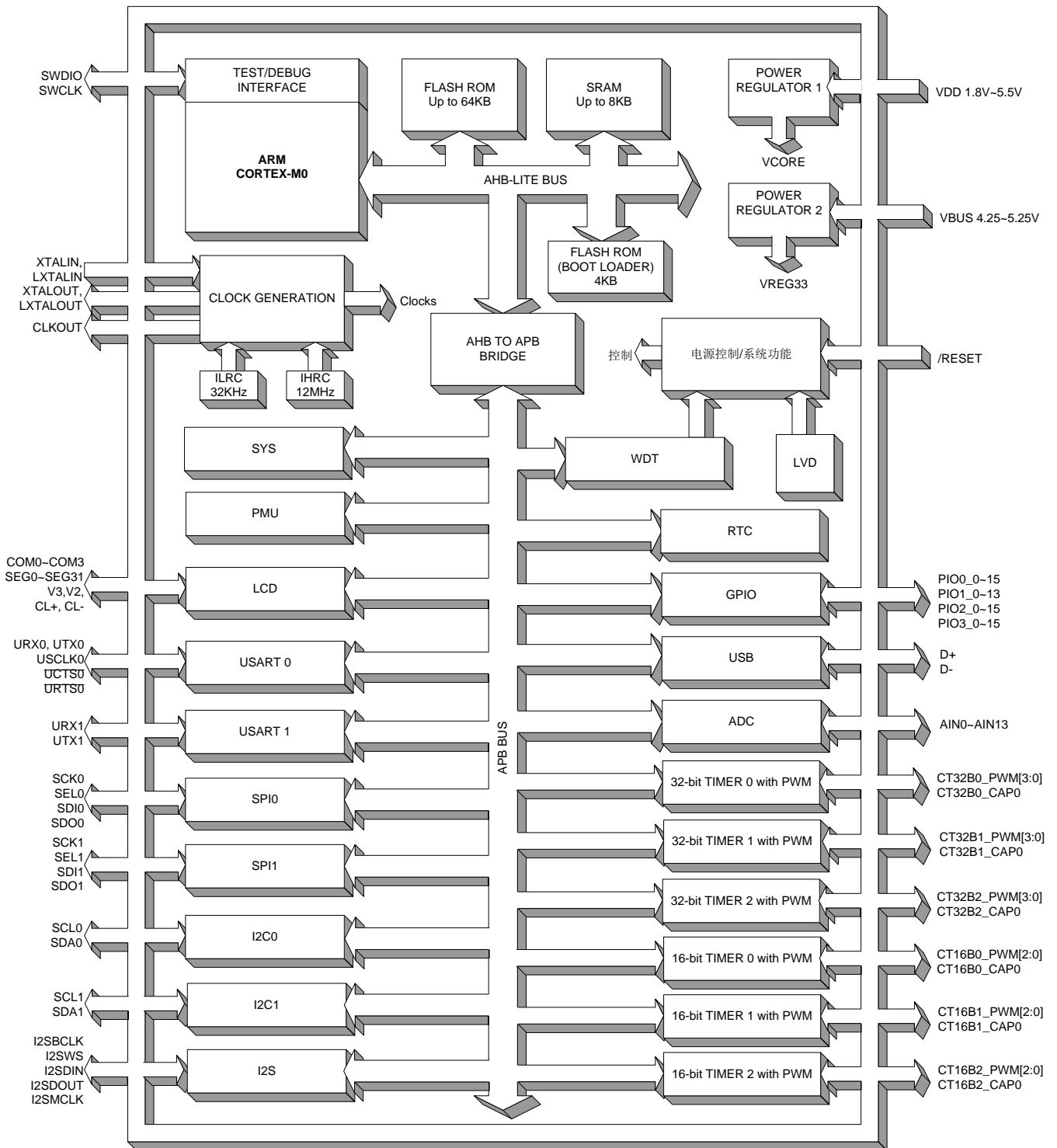
1.1 功能特性

- ◆ **存储器配置**
64KB 内置 Flash 可编程存储器
8KB SRAM
4KB Boot ROM
- ◆ **工作频率高达 50MHz**
- ◆ **中断源**
ARM Cortex-M0 内建嵌套向量中断控制器 (NVIC)
- ◆ **IO 引脚配置**
64 个可配置为上拉/下拉模式的 GPIO 引脚
GPIO 引脚可响应边沿或电平中断源触发
高电流源驱动 (20mA)
- ◆ **可编程的看门狗定时器 (WDT)**
可编程看门狗频率和时钟源以及分频器
- ◆ **SysTick 定时器**
24 位定时器
系统时钟频率决定 SysTick 定时器时钟
SysTick 定时器用于产生一个 10ms 的中断
- ◆ **硬件除法器**
- ◆ **实时时钟 (RTC)**
- ◆ **LVD 支持不同的门限值**
复位: 1.65V for V_{CORE}; 1.8/ 2.0/2.4/2.7/3.0/3.6V for V_{DD}
中断: 1.8/2.0/2.4/2.7/3.0/3.6V for V_{DD}
- ◆ **FCPU (指令周期)**
 $FCPU = FHCLK = FSYCLK/1, FSYCLK/2, FSYCLK/4, \dots, FSYCLK/512$ 。
- ◆ **工作模式**
Normal 模式, 睡眠模式, 深度睡眠模式
- ◆ **串行调试 (SWD)**
- ◆ **定时器**
3 个 16 位和 3 个 32 位通用定时器,
总共带有 6 路捕获输入和 21 路 PWM 输出
- ◆ **工作电压: 1.8V ~ 5.5V**
- ◆ **ADC**
14 通道 12 位 SAR ADC
内置温度传感器
- ◆ **接口**
- 2 个 I2C 控制器, 支持 I2C-bus 规格, 带有
多地址识别功能和监控模式
- 带有波特率发生器的 USART 控制器, 支持 EIA-485
- 带有波特率发生器的 UART 控制器
- 2 个 SPI 控制器, 具有 SSP 特性和多协议性能
- I2S 功能, 支持单声道和立体声数据, 支持 MSB 正常
数据格式, 并且可配置为主机模式或从机模式
- ◆ **系统时钟**
- 外部高速时钟: 晶体模式, 10MHz~25MHz
- 外部低速时钟: 晶体模式, 32.768 KHz
- 内部高速时钟: RC 模式, 12 MHz
- 内部低速时钟: RC 模式, 32 KHz
- PLL 无需高频率的晶振就可使系统达到最大运行速率
- 时钟输出功能可以反映内部高/低速 RC 振荡时钟,
HCLK, PLL 输出和外部高/低速时钟
- ◆ **全速 USB 2.0**
遵循 USB 规范 2.0
输出 3.3V 电压到 USB D+ 引脚 (内置 1.5K 上拉电阻)
支持 1 个全速 USB 设备地址
1 个控制端点和 6 个可配置的实时/中断/bulk 端点,
共 7 个端点共享 512B USB SRAM
- ◆ **LCD 驱动**
支持 R 型和 C 型 LCD
4 common x 32 segment
1/3 或 1/2 偏压
C 型 LCD 的电压范围: 2.7~5.0V
R 型 LCD 可选偏压的电阻值: 400K, 200K, 100K, 35K
支持 1/2duty, 1/3duty, 1/4duty
- ◆ **支持在线烧录 (ISP)**
- ◆ **封装形式**
LQFP 80/64/48 pin
QFN 46/33 pin

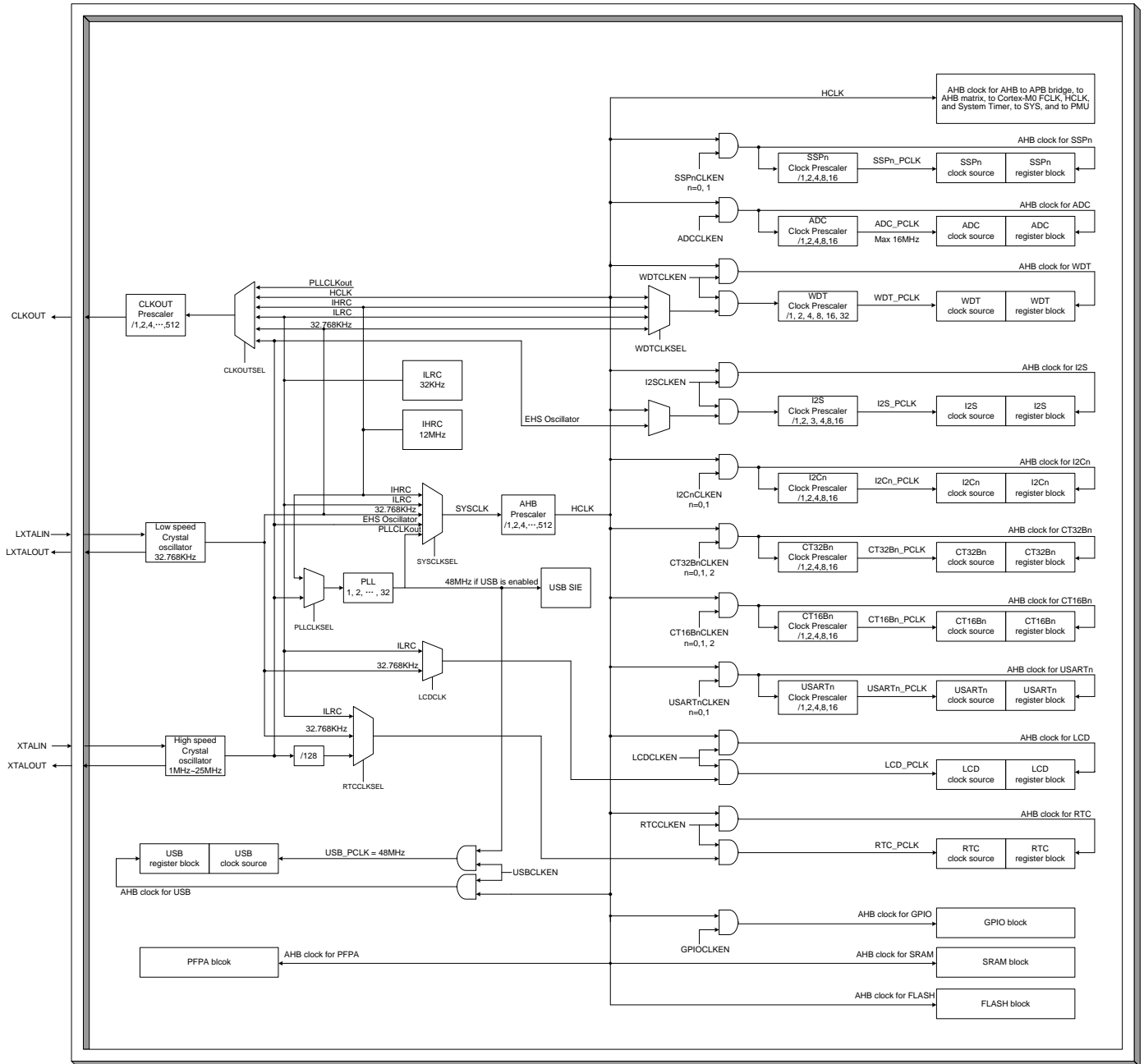
☞ 产品列表

单片机名称	ROM	RAM	引导 下载区	FCPU (Max)	定时器	USART	SPI	I2C	I2S	PWM	12 位 ADC	LCD	唤醒引脚 数目	封装
SN32F249FG	64KB	8KB	4KB	50 MHz	16-bitx3 32-bitx3	USARTx1 UARTx1	2	2	1	21	14CH	4x32	64	LQFP80
SN32F248FG	64KB	8KB	4KB	50 MHz	16-bitx3 32-bitx3	UARTx2	2	2	1	21	11CH	4x28	55	LQFP64
SN32F247FG	64KB	8KB	4KB	50 MHz	16-bitx3 32-bitx3	UARTx2	2	2	1	21	8CH	4x15	39	LQFP48
SN32F246JG	64KB	8KB	4KB	50 MHz	16-bitx3 32-bitx3	UARTx2	2	2	1	21	6CH	4x15	37	QFN46
SN32F245JG	64KB	8KB	4KB	50 MHz	16-bitx3 32-bitx3	UARTx2	2	2	-	21	4CH	-	23	QFN33
SN32F239FG	32KB	4KB	4KB	50 MHz	16-bitx3 32-bitx3	USARTx1 UARTx1	2	2	1	21	14CH	4x32	64	LQFP80
SN32F238FG	32KB	4KB	4KB	50 MHz	16-bitx3 32-bitx3	UARTx2	2	2	1	21	11CH	4x28	55	LQFP64
SN32F237FG	32KB	4KB	4KB	50 MHz	16-bitx3 32-bitx3	UARTx2	2	2	1	21	8CH	4x15	39	LQFP48
SN32F236JG	32KB	4KB	4KB	50 MHz	16-bitx3 32-bitx3	UARTx2	2	2	1	21	6CH	4x15	37	QFN46
SN32F235JG	32KB	4KB	4KB	50 MHz	16-bitx3 32-bitx3	UARTx2	2	2	-	21	4CH	-	23	QFN33

1.2 系统框图

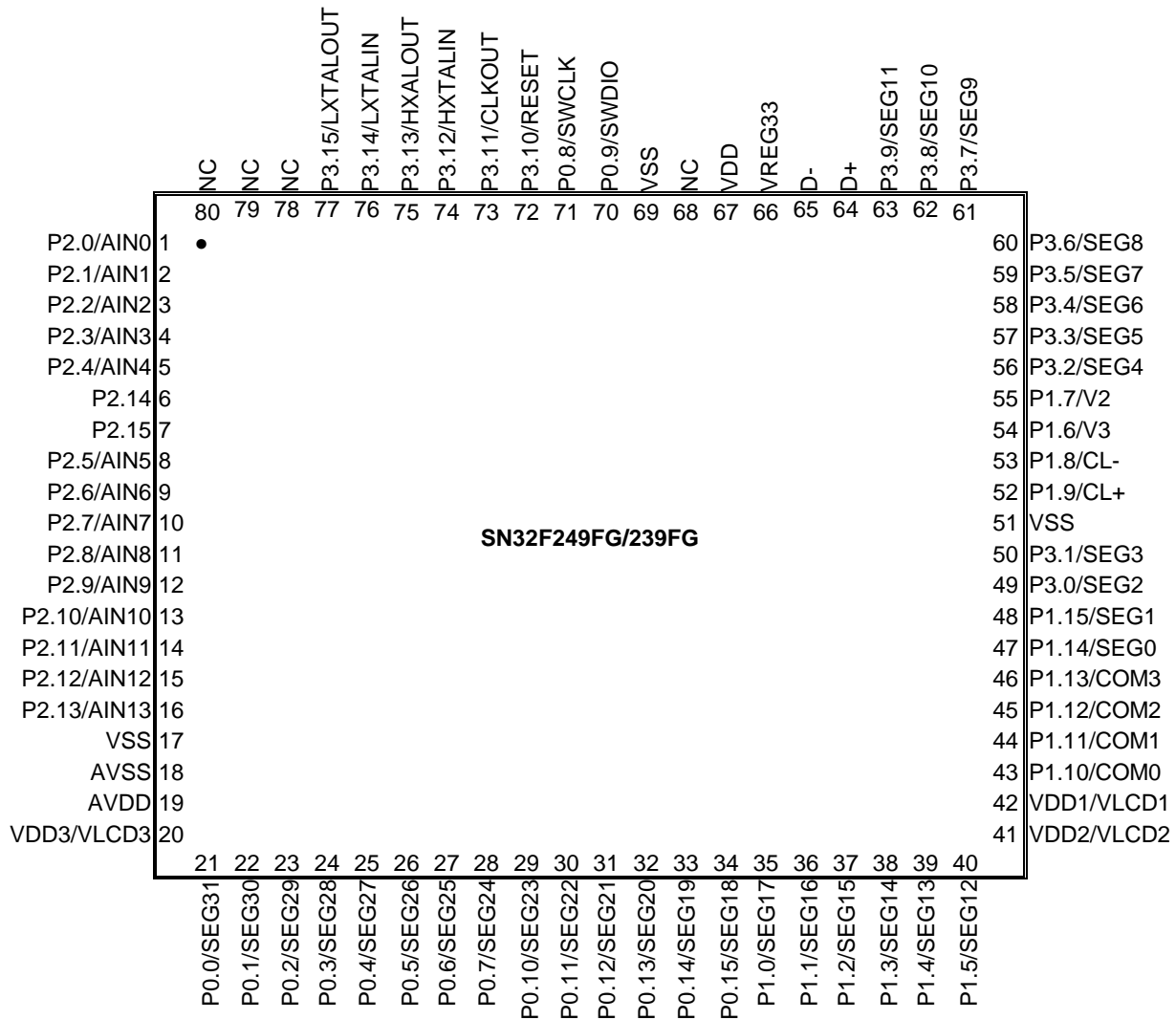


1.3 时钟产生框图



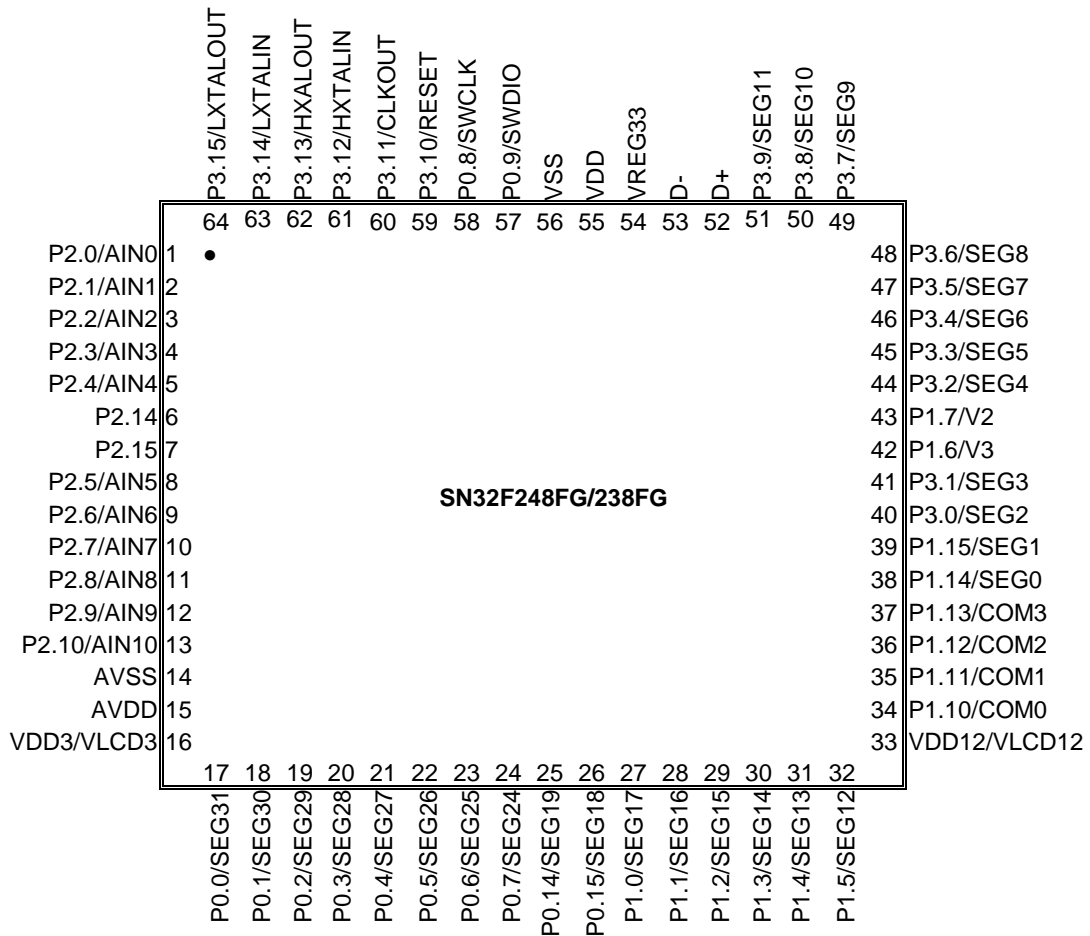
1.4 引脚配置

- SN32F249FG/239FG (LQFP 80 pins)



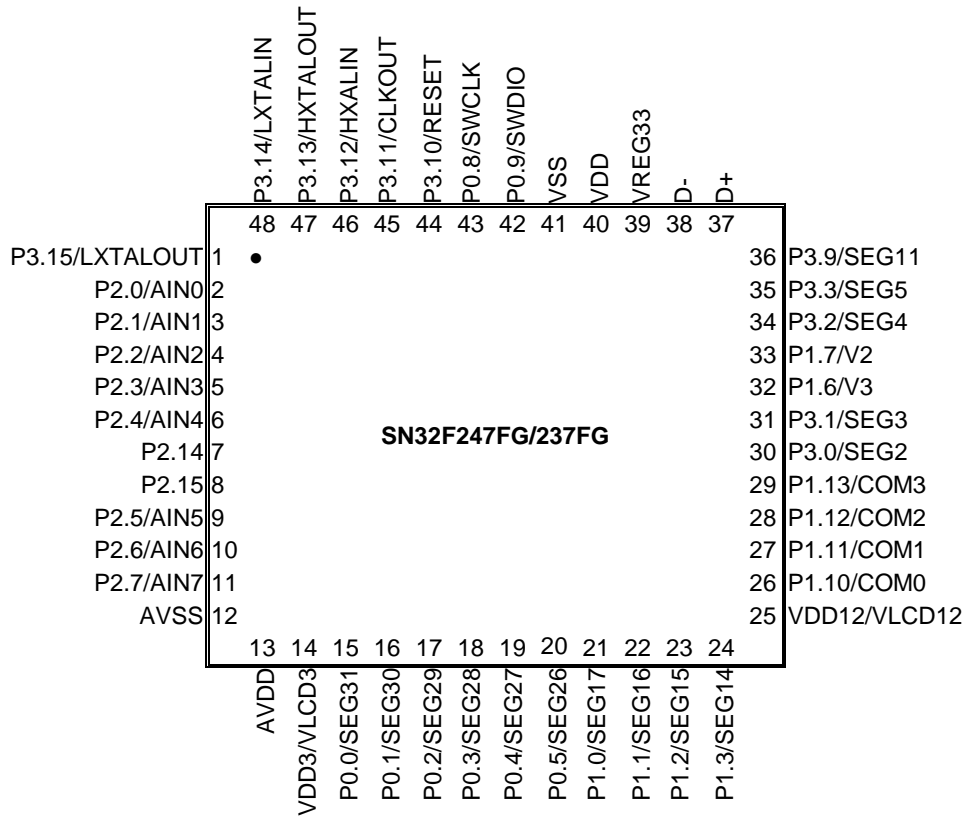
* 注: SONiX提供Boot loader在Boot进程中来检查P2.2 (BOOT引脚)的状态。如果在Boot进程中BOOT引脚为低, MCU将在Boot loader中执行代码,而不是在用户代码。我们强烈建议不要使用BOOT引脚作为输出引脚来驱动LED,否则在Boot进程中BOOT引脚可能为低。

● SN32F248FG/238FG (LQFP 64 pins)



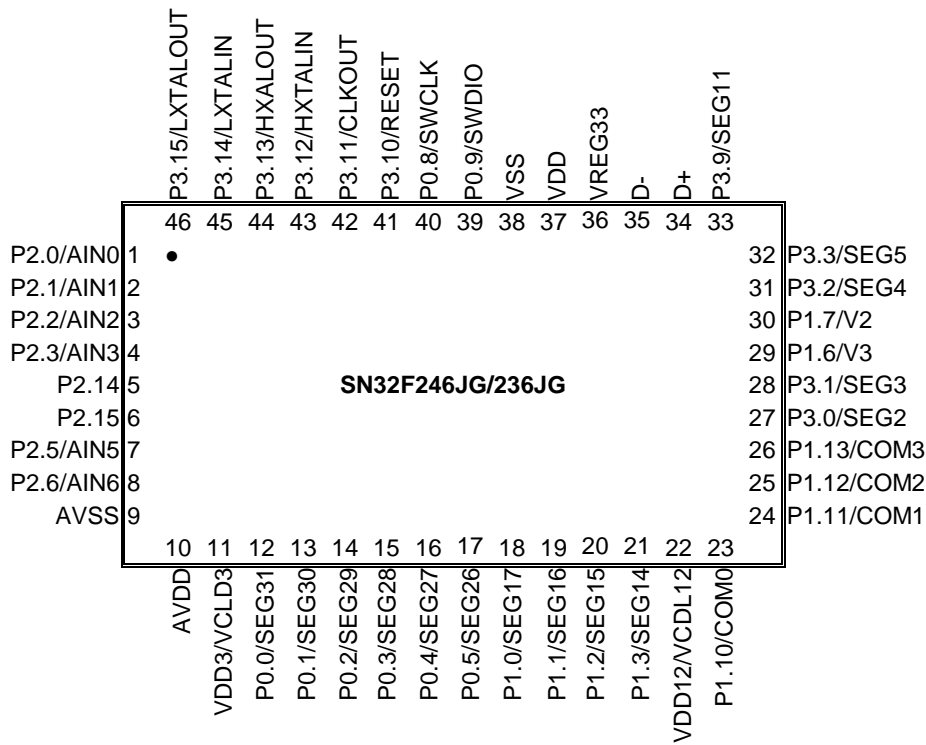
* 注: 1. 在低功耗模式下, 没有配置的引脚需要设置正确来减低功耗。强烈建议将这些引脚设置为输入上拉。
 2. SONiX 提供 Boot loader 在 Boot 进程中来检查 P2.2 (BOOT 引脚) 的状态。如果在 Boot 进程中 BOOT 引脚为低, MCU 将在 Boot loader 中执行代码, 而不是在用户代码。我们强烈建议不要使用 BOOT 引脚作为输出引脚来驱动 LED, 否则在 Boot 进程中 BOOT 引脚可能为低。

● SN32F247FG/237FG (LQFP 48 pins)



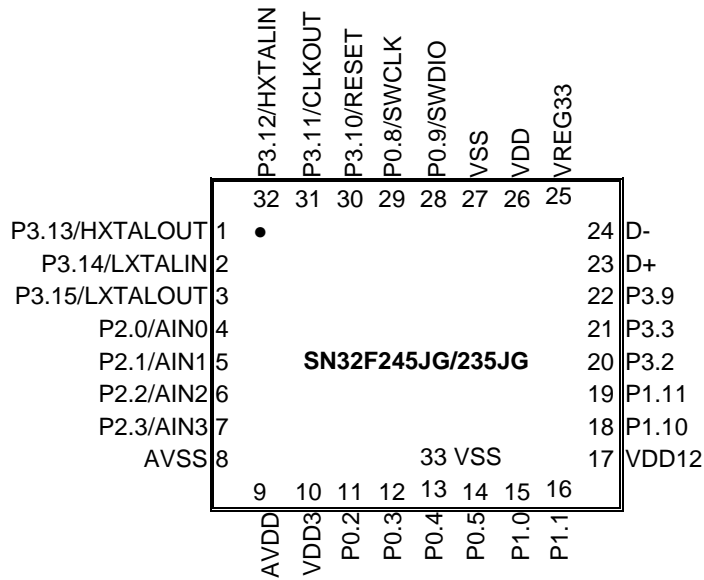
* 注: 1. 在低功耗模式下, 没有配置的引脚需要设置正确来减低功耗。强烈建议将这些引脚设置为输入上拉。
 2. SONiX 提供 Boot loader 在 Boot 进程中来检查 P2.2 (BOOT 引脚)的状态。如果在 Boot 进程中 BOOT 引脚为低, MCU 将在 Boot loader 中执行代码, 而不是在用户代码。我们强烈建议不要使用 BOOT 引脚作为输出引脚来驱动 LED, 否则在 Boot 进程中 BOOT 引脚可能为低。

● SN32F246JG/236JG (QFN 46 pins)



* 注: 1. 在低功耗模式下, 没有配置的引脚需要设置正确来减低功耗。强烈建议将这些引脚设置为输入上拉。
 2. SONiX 提供 Boot loader 在 Boot 进程中来检查 P2.2 (BOOT 引脚) 的状态。如果在 Boot 进程中 BOOT 引脚为低, MCU 将在 Boot loader 中执行代码, 而不是在用户代码。我们强烈建议不要使用 BOOT 引脚作为输出引脚来驱动 LED, 否则在 Boot 进程中 BOOT 引脚可能为低。

- SN32F245JG/235JG (QFN 33 pins)



- * 注: 1. 在低功耗模式下, 没有配置的引脚需要设置正确来减低功耗。强烈建议将这些引脚设置为输入上拉。
2. SONiX 提供 Boot loader 在 Boot 进程中来检查 P2.2 (BOOT 引脚) 的状态。如果在 Boot 进程中 BOOT 引脚为低, MCU 将在 Boot loader 中执行代码, 而不是在用户代码。我们强烈建议不要使用 BOOT 引脚作为输出引脚来驱动 LED, 否则在 Boot 进程中 BOOT 引脚可能为低。

1.5 引脚说明

引脚名称	引脚类型	功能说明
VDD, VSS	P	数字电路电源输入端。
AVDD, AVSS	P	模拟电路电源输入端。
VREG33	O	3.3V 电源引脚。
D+, D-	I/O	USB 差分数据引脚。
VDD1/VLCD1	P	I/O 引脚 P1.6~P1.15、P3.0~P3.9 和 LCD 驱动电源输入引脚。作为 I/O 口的电源电压时须等于 VDD。
VDD2/VLCD2	P	I/O 引脚 P0.10~P0.15、P1.0~P1.5 和 LCD 驱动电源输入引脚
VDD12/VLCD12	P	VDD1 和 VDD2 的 bonding 引脚。作为 I/O 口的电源电压时须等于 VDD。
VDD3/VLCD3	P	I/O 引脚 P0.0~P0.7 和 LCD 驱动电源输入引脚。若 VDD3 的电压值小于 VDD，为防止漏电，用户必须手动设置 P1.6、P1.7 为输入下拉状态
P1.9/CL+, P1.8/CL-	I/O, P	P1.8, P1.9—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。CL+, CL—C 型 LCD charge pump 电容。
P1.6/V3, P1.7/V2	I/O, P	P1.6, P1.7—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。V3—2/3 VLCD 偏压。V2—1/3 VLCD 偏压。
P0.0~P0.7	I/O	P0.0~P0.7—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻。
P0.8/SWCLK	I/O	P0.8—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。SWCLK—串行时钟引脚。
P0.9/SWDIO	I/O	P0.9—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。SWDIO—串行数据输入/输出引脚。
P0.10~P0.15	I/O	P0.10~P0.15—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。
P1.0~P1.5, P1.10~P1.15	I/O	P1.0~P1.5, P1.10~P1.15—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。
P2.0~P2.13/ AIN0~13	I/O	P2.0~P2.13—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。AIN0~AIN13—ADC 输入通道。
P2.14	I/O	P2.14—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。
P2.15	I/O	P2.15—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。
P3.0~P3.9	I/O	P3.0~P3.9—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。
P3.10/RESET	I/O	P3.10—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。RESET—外部复位输入引脚，施密特触发，低电平有效，通常保持高电平。
P3.11/CLKOUT	I/O	P3.11—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。CLKOUT—时钟输出引脚。
P3.12/HXTALIN	I/O	P3.12—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。HXTALIN—外部高速 X'tal 输入引脚。
P3.13/ HXTALOUT	I/O	P3.13—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。HXTALOUT—外部高速 X'tal 输出引脚。
P3.14/LXTALIN	I/O	P3.14—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。LXTALIN—外部低速 X'tal 输入引脚。
P3.15/ LXTALOUT	I/O	P3.15—双向输入输出引脚，施密特触发，输入模式时内置上拉/下拉电阻，具有唤醒功能。LXTALOUT—外部低速 X'tal 输出引脚。

有关 GPIO 引脚的设置请参考外围设备功能引脚配置（PFPA）章节。

引脚名称	共用引脚
P0.0	SEG31 / URXD0 / CT32B0_CAP0 / SCL1 / MISO0 CT16B0_PWM0 / CT16B2_PWM1 / CT32B2_PWM2
P0.1	SEG30 / UTXD0 / SDA1 / MOSI0 CT16B0_PWM1 / CT16B2_PWM2 / CT32B1_PWM0
P0.2	SEG29 / SCL0 / MISO0 / SEL1 CT16B0_CAP0 / CT32B1_CAP0 / CT16B2_PWM0/CT32B2_PWM1
P0.3	SEG28 / SDA0 / MOSI0 / CT16B2_CAP0 / CT32B2_PWM2 / CT32B0_PWM3 / CT32B2_PWM0
P0.4	SEG27 / URXD0 / SCK0 / MISO1

	CT16B0_PWM1 / CT32B0_PWM0 / CT32B1_PWM1
P0.5	SEG26 / UTXD0 / SEL0 / MOSI1 CT16B1_PWM0 / CT32B0_PWM2 / CT32B1_PWM2
P0.6	SEG25 / UTXD1 / SCL1 / MISO0 CT16B1_PWM2 / CT32B1_PWM3 / C32B2_PWM3
P0.7	SEG24 / URXD1 / SDA1 / MOSI0 / SCK1 CT16B1_CAP0 / CT32B0_PWM3 / CT32B1_PWM3
P0.8	SWCLK / CT16B0_CAP0 / CT32B1_CAP0 CT16B1_PWM1 / CT32B0_PWM1 / CT32B1_PWM2 / CT32B2_PWM0
P0.9	SWDIO / CT32B0_CAP0 / CT32B2_CAP0 CT16B0_PWM2 / CT16B2_PWM1 / CT32B0_PWM2 / CT32B1_PWM0 / CT32B2_PWM2
P0.10	SEG23 / I2SDIN / SDA0 / SEL0 / MISO1 CT16B0_PWM1 / CT16B1_PWM0 / CT32B2_CAP0
P0.11	SEG22 / I2SDOUT / SCK0 CT16B0_PWM2 / CT32B0_PWM0 / CT16B1_PWM1 / CT32B1_PWM2
P0.12	SEG21 / URXD1 / SEL0 / MOSI1 / I2SMCLK CT16B1_PWM1 / CT16B1_CAP0
P0.13	SEG20 / UTXD1 / SCK0 / SEL1 / I2SBCLK CT16B2_CAP0 / CT32B2_PWM0
P0.14	SEG19 / MOSI0 / SCK1 / I2SWS CT32B2_PWM1 / CT16B2_PWM0 / CT32B0_PWM1
P0.15	SEG18 / SCL0 / MISO0 CT32B1_PWM1 / CT16B1_PWM2 / CT32B0_PWM3 / CT32B2_PWM1 / CT32B1_CAP0
P1.0	SEG17 / URXD1 / SDA1 / MISO0 / SEL1 CT16B0_CAP0 / CT16B2_PWM2 / CT32B2_PWM3
P1.1	SEG16 / UTXD1 / SCL1 / MOSI0 / SCK1 CT16B0_PWM0 / CT32B2_PWM0 / CT32B0_CAP0
P1.2	SEG15 / UTXD0 / SDA0 / MOSI1 CT16B1_PWM2 / CT32B1_PWM1 / CT32B0_PWM3 / CT32B1_PWM3
P1.3	SEG14 / URXD0 / SCL0 / MISO1 CT32B1_CAP0 / CT16B1_PWM1 / CT32B1_PWM0
P1.4	SEG13 / UTXD1 / SDA0 / SCK0 / SEL1 CT16B2_PWM1 / CT16B2_PWM0 / CT32B2_CAP0
P1.5	SEG12 / URXD1 / SCL0 / SEL0 CT16B2_CAP0 / CT32B0_PWM1 / CT32B2_PWM1
P1.6	V3 / UCTS0 / I2SDIN / MOSI1 CT16B0_PWM2 / CT32B1_PWM2 / CT32B2_PWM2 / CT32B1_CAP0
P1.7	V2 / USCLK0 / I2SDOUT / SEL1 CT16B1_CAP0 / CT32B1_PWM3 / CT32B2_PWM0 / CT32B2_PWM3
P1.8	CL- / URTS0 / SDA1 / I2SMCLK CT16B0_PWM0 / CT16B2_CAP0 / CT32B0_PWM0
P1.9	CL+ / SCL1 / I2SBCLK CT16B1_PWM2 / CT16B1_PWM0 / CT32B0_PWM2
P1.10	COM0 / MISO1 / I2SWS CT16B1_PWM3 / CT16B1_PWM1 / CT32B1_PWM1
P1.11	COM1 / SEL0 / SCK1 CT16B1_CAP0 / CT16B2_PWM1 / CT32B0_PWM1
P1.12	COM2 / SCK0 CT16B0_PWM0 / CT16B0_PWM2 / CT16B2_PWM2 / CT32B0_PWM3
P1.13	COM3 / URXD1 / SDA0 / MOSI1 CT16B0_PWM1 / CT32B1_PWM0 / CT32B2_CAP0
P1.14	SEG0 / UTXD1 / SCL0 / MISO0 / SEL1 CT16B1_PWM2 / CT32B0_PWM0 / CT32B0_PWM2
P1.15	SEG1 / URXD1 / MOSI0 / SCK1 CT16B1_PWM0 / CT32B0_PWM0 / CT32B2_PWM0
P2.0	AIN0 / I2SDIN / SEL0 / MOSI1 CT16B0_CAP0 / CT16B1_PWM0 / CT32B0_PWM2
P2.1	AIN1 / MISO0 / MISO1 / SEL1 / I2SWS CT16B1_CAP0 / CT16B2_PWM1 / CT32B1_PWM1
P2.2	AIN2 / MOSI0 / SCK1 / I2SMCLK CT16B2_CAP0 / CT16B0_PWM1 / CT32B0_PWM3
P2.3	AIN3 / SCK0 CT16B0_PWM0 / CT16B1_PWM2 / CT32B0_PWM0 / CT32B2_PWM3 / CT32B0_CAP0
P2.4	AIN4 / CT32B1_CAP0 CT16B0_PWM2 / CT16B1_PWM1 / CT16B2_PWM2 / CT32B0_PWM1 / CT32B1_PWM3

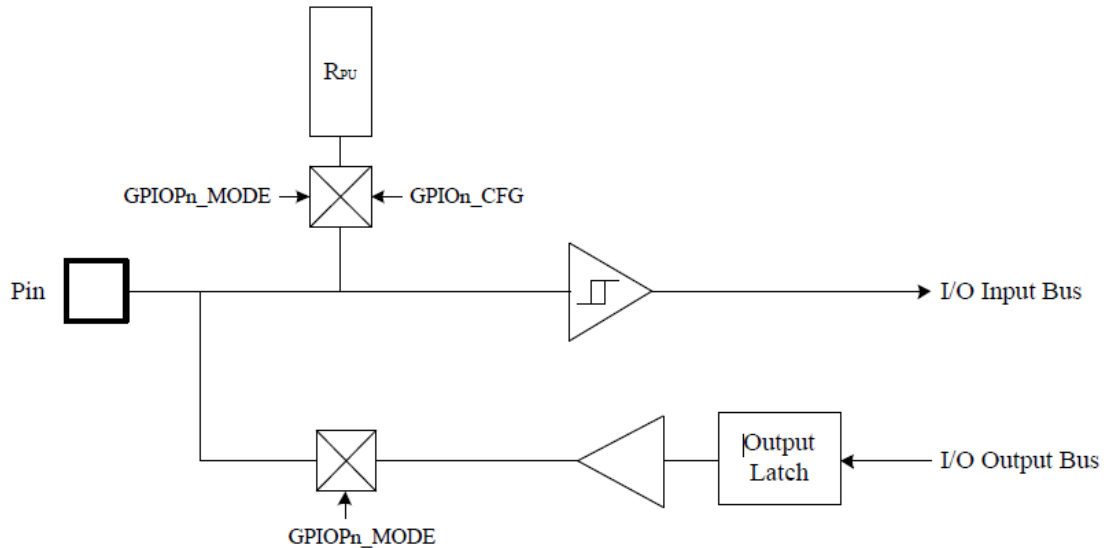
P2.5	AIN5 / I2SDIN CT16B2_PWM0 / CT32B1_PWM2 / CT32B2_CAP0
P2.6	AIN6 / I2SMCLK CT16B2_PWM1 / CT32B1_PWM0
P2.7	AIN7 / I2SWS CT16B1_PWM2 / CT32B2_PWM1
P2.8	AIN8 / I2SDOUT CT16B1_PWM1 / CT32B1_PWM1 / CT32B2_PWM3
P2.9	AIN9 / I2SWS CT16B1_CAP0 / CT16B2_PWM0 / CT32B2_PWM3
P2.10	AIN10 / I2SBCLK CT16B0_PWM2 / CT16B2_PWM2 / CT32B1_CAP0
P2.11	AIN11 / I2SBCLK CT16B0_PWM1 / CT32B0_PWM2 / CT32B2_PWM0
P2.12	AIN12 / MISO0 / MISO1 / I2SDOUT CT16B1_PWM0 / CT32B0_PWM3 / CT32B1_PWM2 / CT32B2_CAP0
P2.13	AIN13 / MOSI0 / SCK1 CT16B0_CAP0 / CT32B0_PWM1 / CT32B1_PWM3 / CT32B2_PWM1
P2.14	SCK0 / SEL1 CT16B2_CAP0 / CT32B0_PWM0 / CT32B1_PWM2 / CT32B2_PWM2
P2.15	SEL0 / MOSI1 CT16B0_PWM0 / CT32B1_PWM0 / CT32B2_PWM0 / CT32B2_PWM3 / CT32B0_CAP0
P3.0	SEG2 / URXD0 / I2SDIN / SCL1 / MISO1 CT16B0_CAP0 / CT16B2_PWM2 / CT32B2_PWM1
P3.1	SEG3 / I2SDOUT / UTXD0 / SEL0 CT16B2_PWM2 / CT16B2_PWM0 / CT32B2_PWM2 / CT32B0_CAP0
P3.2	SEG4 / UTXD0 / SDA1 / I2SMCLK / SCK0 / MOSI1 CT16B0_CAP0 / CT32B1_PWM0
P3.3	SEG5 / URXD0 / SCL1 / MISO0 / SCK1 / I2SBCLK CT16B0_PWM0 / CT32B0_CAP0
P3.4	SEG6 / UTXD0 / SDA1 / MOSI0 / MISO1 / I2SWS CT16B0_PWM1 / CT32B0_PWM3
P3.5	SEG7 / URXD0 / SDA1 / SEL0 CT16B1_CAP0 / CT16B2_PWM0 / CT16B2_PWM1 / CT32B1_PWM1
P3.6	SEG8 / URXD1 / SCL1 / SEL1 / SCK0 / I2SBCLK CT16B2_CAP0 / CT16B0_PWM2
P3.7	SEG9 / SDA0 / SCK1 / I2SMCLK CT16B1_PWM0 / CT32B0_PWM2 / CT32B2_PWM1 / CT32B2_CAP0
P3.8	SEG10 / UTXD1 / I2SDOUT / MOSI1 CT16B1_PWM2 / CT32B0_PWM1 / CT32B1_CAP0
P3.9	SEG11 / I2SDIN / MISO1 / SCL0 CT16B1_PWM1 / CT32B0_PWM0 / CT32B1_PWM3 / CT32B0_CAP0
P3.10	RESET / UTXD0 / UTXD1 / SEL0 CT16B0_CAP0 / CT16B2_PWM0 / CT32B1_PWM1 / CT32B2_PWM3
P3.11	CLKOUT / URXD0 / SCL0 / SCK0 / SEL1 CT16B0_PWM0 / CT16B2_PWM2 / CT32B1_PWM3
P3.12	HXTALIN / URXD1 / SDA1 / MISO1 CT16B0_PWM1 / CT16B2_CAP0 / CT32B1_PWM2
P3.13	HXALOUT / UTXD1 / SCL1 / SDA0 / MOSI1 CT16B1_CAP0 / CT32B0_PWM2 / CT32B2_PWM2
P3.14	LXTALIN / URXD0 / SCL0 / MOSI0 / SCK1 CT16B1_PWM0 / CT32B0_PWM1 / CT32B2_CAP0
P3.15	LXTALOUT / UTXD0 / SDA0 / MISO0 CT16B0_PWM2 / CT16B2_PWM1 / CT32B1_PWM0 / CT32B2_PWM2

* 注:

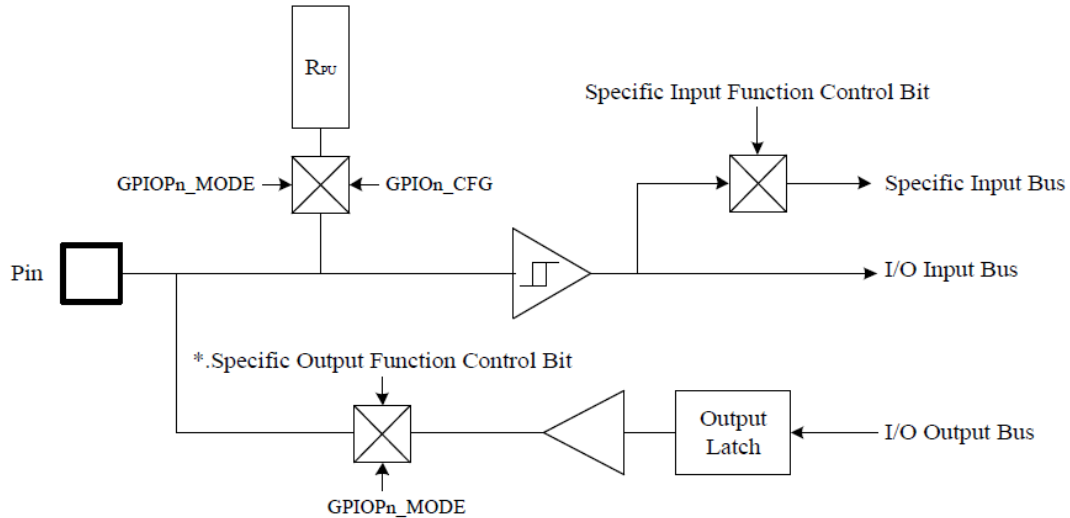
1. VDD1/VLCD1 是 I/O 引脚 P1.6~P1.15、P3.0~P3.9 和 LCD 驱动电源输入引脚。作为 I/O 口的电源电压时必须等于 VDD。
2. VDD12/VLCD12 是 VDD1 和 VDD2 的 bonding 引脚。作为 I/O 口的电源电压时必须等于 VDD。
3. VDD3/VLCD3 是 I/O 引脚 P0.0~P0.7 和 LCD 驱动电源输入引脚。若 VDD3 的电压值小于 VDD，为防止漏电，用户必须手动设置 P1.6、P1.7 为输入下拉状态

1.6 引脚电路结构图

- GPIO 引脚:

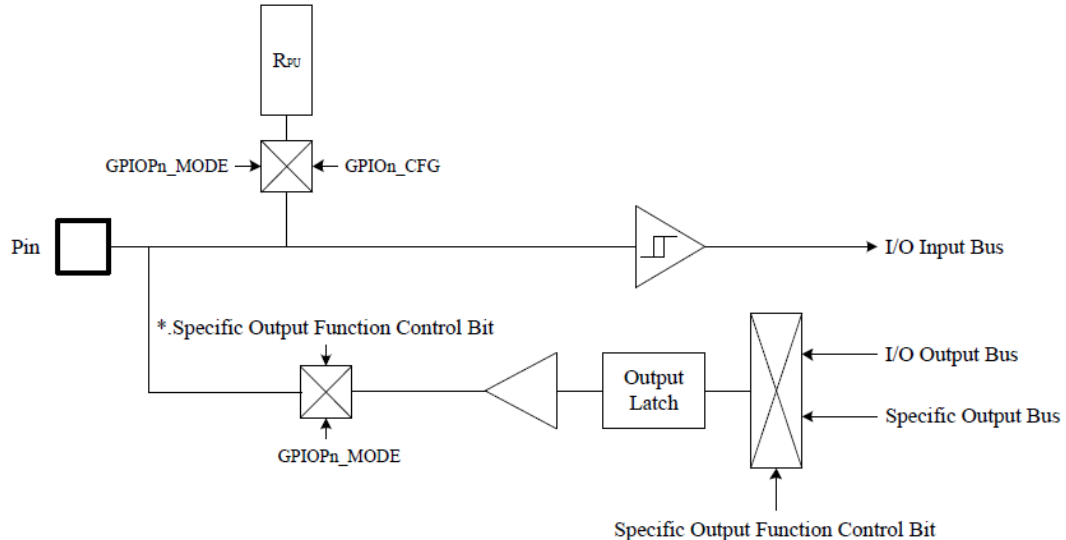


- GPIO 引脚, 与特殊数字输入功能引脚共用, 如 SPI, I2C.....:



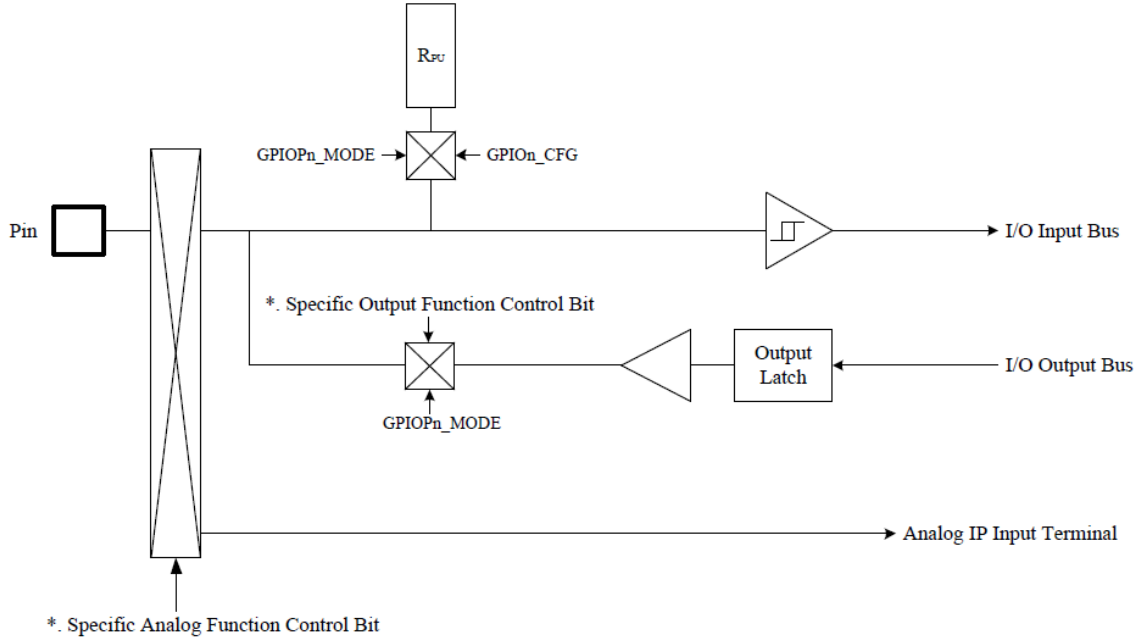
*. Some specific functions switch I/O direction directly, not through $GPIOOn_MODE$ register.

- GPIO 引脚, 与特殊数字输出功能引脚共用, 如 SPI, I2C.....:



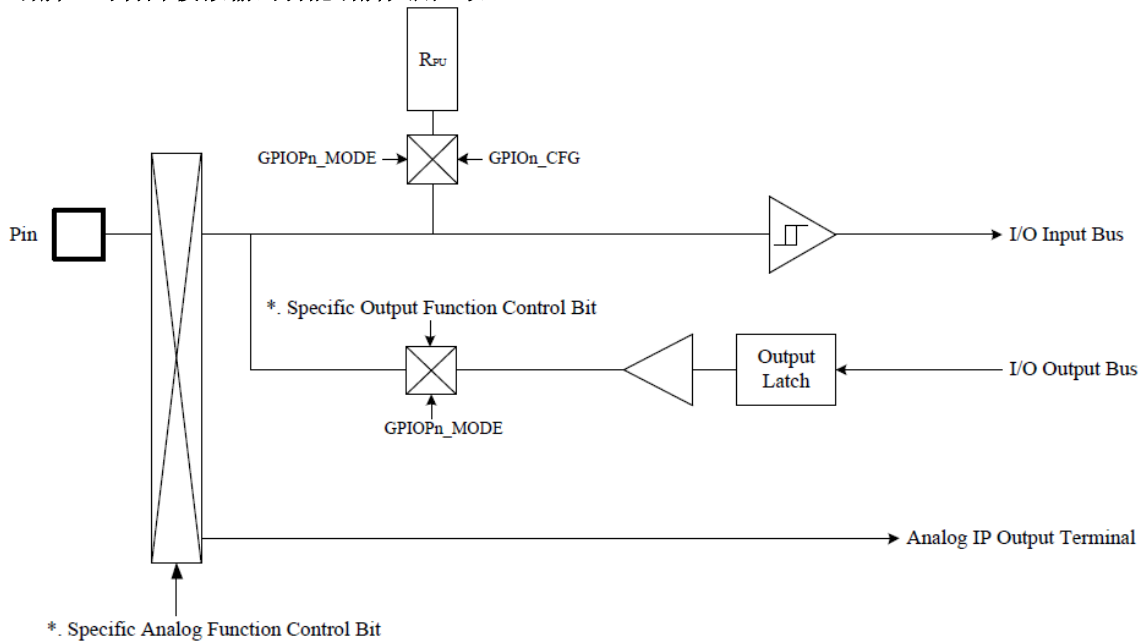
*. Some specific functions switch I/O direction directly, not through $GPIOOn_MODE$ register.

- **GPIO 引脚，与特殊模拟输入功能引脚共用，如 XIN, ADC.....:**



*. Some specific functions switch I/O direction directly, not through GPIO_n_MODE register.

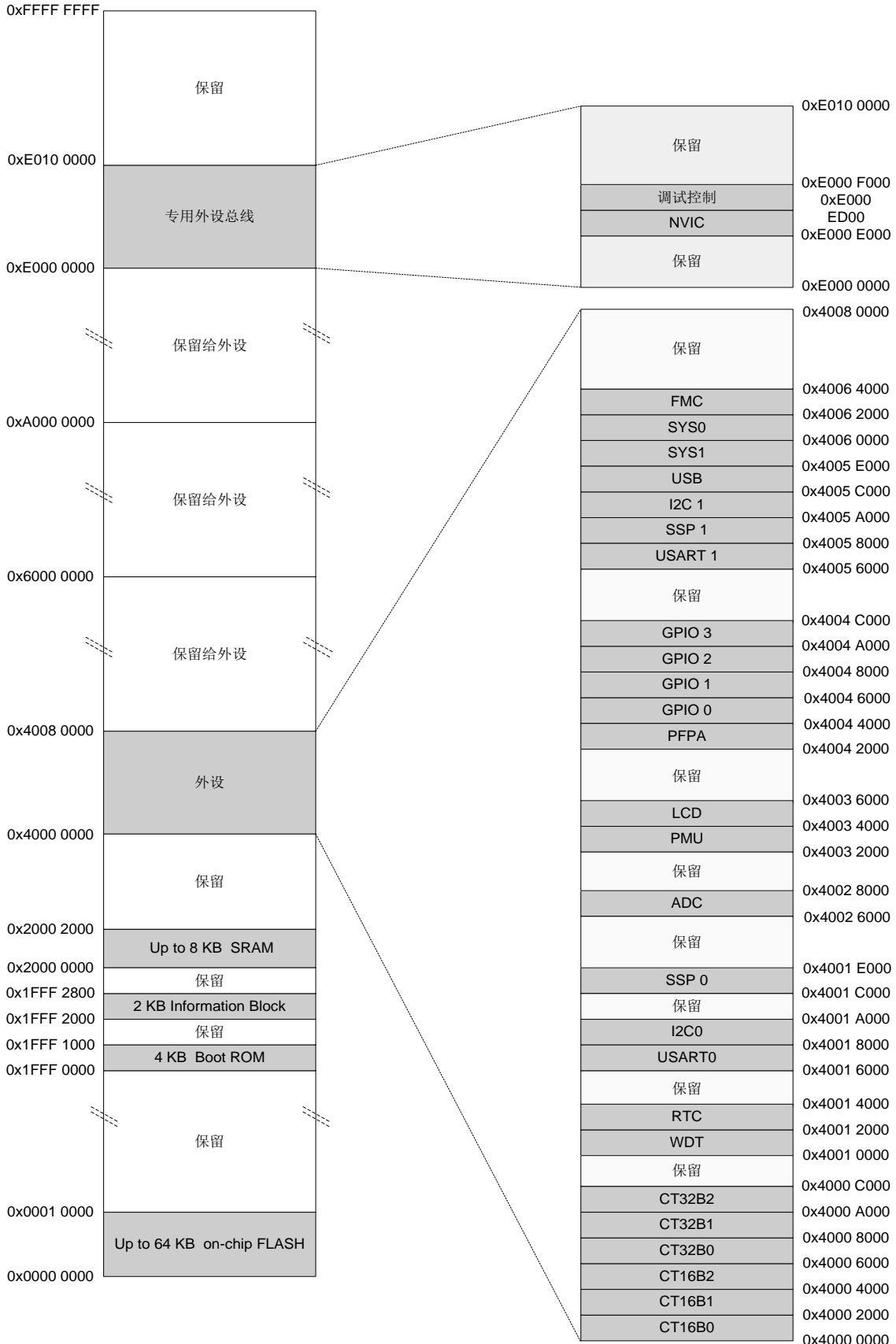
- **GPIO 引脚，与特殊模拟输出功能引脚共用，如 XOUT.....:**



*. Some specific functions switch I/O direction directly, not through GPIO_n_MODE register.

2 中央处理器 (CPU)

2.1 存储器



2.2 系统 TICK 定时器

SysTick 定时器是 Cortex-M0 的一个集成部分，可以产生一个 10ms 的中断，用于操作系统或者其他系统软件管理。

由于 SysTick 定时器是 Cortex-M0 的一部分，通过提供一个标准定时器（基于 Cortex-M0 设备上有效）使软件端口更容易。

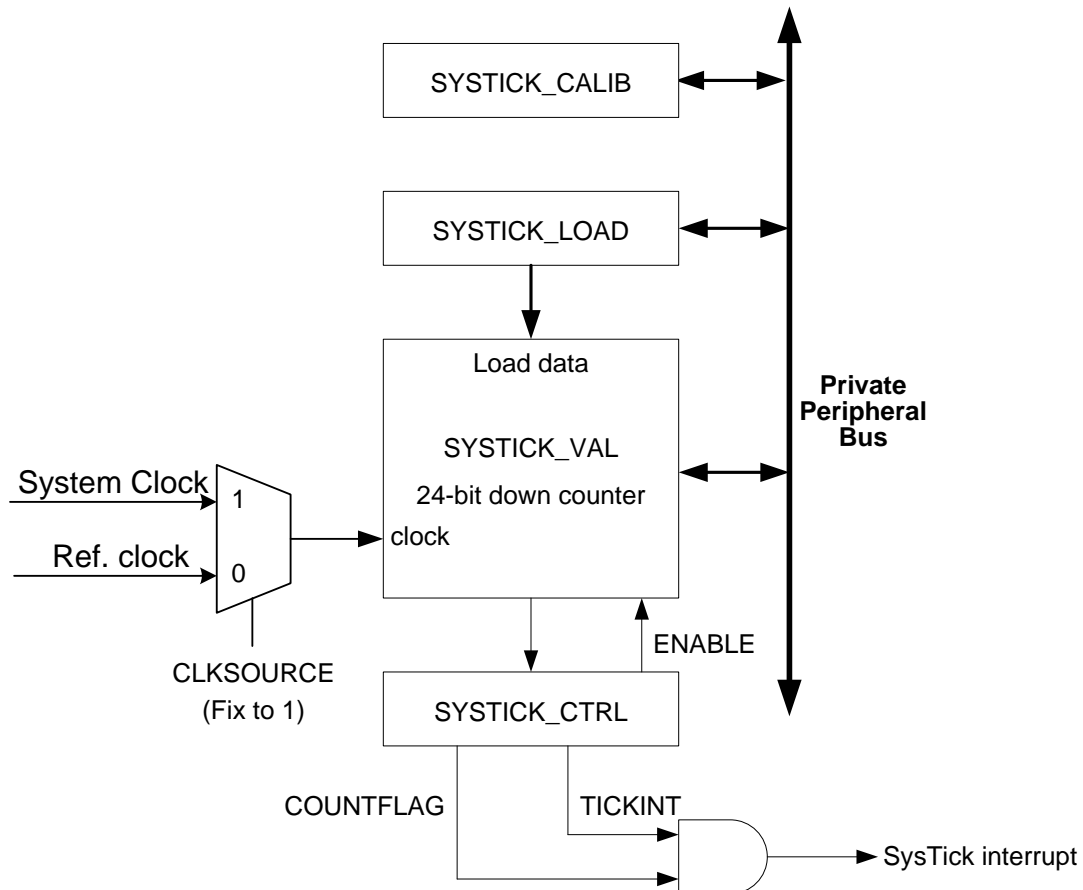
详情请参考 Cortex-M0 用户指南。

2.2.1 操作

SysTick 定时器是一个 24 位递减计数定时器，递减计数至 0 时产生中断。

Systick 定时器由 Systick 控制寄存器控制使能，时钟频率固定为系统时钟，它可以提供一个固定的 10ms 中断。

☞ SysTick 定时器的结构框图



当使能 SysTick 定时器，定时器从当前数值（SYSTICK_VAL）开始递减计数至 0，在下一个时钟沿将重新载入 SYST_RVR 的值到 SysTick 重装值寄存器（SYSTICK_LOAD）中，然后在随后的时钟里递减计数。当计数器计数至 0 时，COUNTFLAG 状态位置 1，读取后该位清零。

* 注：当处理器因调试而停止时，计数器不会进行递减计数。

2.2.2 SYSTICK 用法说明及技巧

中断控制器时钟更新 SysTick 计数器。一些操作可以停止 SysTick 时钟信号以进入低功耗模式，此时 SysTick 计数器停止计数。

软件务必使用整字（word）方式操作 SysTick 寄存器。

SysTick 计数器重装数据和当前数据不会由 HW 来初始化。这意味着 SysTick 计数器正确的初始化流程如下：

1. 编程重装值到SYSTICK_LOAD寄存器中。
2. 通过写入任意值到SYSTICK_VAL寄存器，将当前值清除。
3. 设置控制和状态（SYSTICK_CTRL）寄存器。

2.2.3 SYSTICK 寄存器

2.2.3.1 系统 Tick 定时器控制和状态寄存器（SYSTICK_CTRL）

地址：0xE000 E010（参考 Cortex-M0 规格）

Bit	Name	Description	Attribute	Reset
31:17	Reserved		R	0
16	COUNTFLAG	SysTick 计数器递减计数到 0 时，该位置 1，被读取后清零。	R/W	0
15:3	Reserved		R	0
2	CLKSOURCE	选择 SysTick 定时器时钟源。 0: 参考时钟； 1: 系统时钟（固定）。	R	1
1	TICKINT	SysTick 中断使能位。 0: 禁止 SysTick 中断。 1: 使能 SysTick 中断（当 SysTick 计数器递减计数到 0 时产生中断）。	R/W	0
0	ENABLE	SysTick 计数器使能位。 0: 禁止； 1: 使能。	R/W	0

2.2.3.2 系统 Tick 定时器重装值寄存器（SYSTICK_LOAD）

地址：0xE000 E014（参考 Cortex-M0 规格）

重装寄存器需设置当 SysTick 递减计数到 0 时，重新装载的数据。此寄存器由软件来初始化。如果系统或外部时钟运行在预期的频率，SYSTICK_CALIB 寄存器的值可以使用作为重装寄存器的值。

下面的示例说明选择 SysTick 定时器重装值，以获得 10ms 的间隔时间，系统时钟设置为 50MHz。

SysTick 时钟 = 系统时钟 = 50 MHz

RELOAD = (SysTick 时钟频率 × 10 ms) - 1 = (50 MHz × 10 ms) - 1 = 0x0007A11F

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	RELOAD	使能计数器且计数到 0 时，此值重新装入 SYSTICK_VAL 寄存器。	R/W	0x5F7F9B

2.2.3.3 系统 Tick 定时器当前值寄存器 (SYSTICK_VAL)

地址: 0xE000 E018 (Refer to Cortex-M0 Spec)

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	CURRENT	读取该寄存器时返回 SysTick 计数器的当前值; 写入任何数据清除 SysTick 计数器和 SYSTICK_CTRL 寄存器中的 SOUNFLAG 位。	R/W	0x7E7F35

2.2.3.4 系统 Tick 定时器校准值寄存器 (SYSTICK_CALIB)

地址: 0xE000 E01C (参考 Cortex-M0 规格)

Bit	Name	Description	Attribute	Reset
31	NOREF	显示是否提供参考时钟给 M0。 1: 没有提供参考时钟	R	1
30	SKEW	显示 TENMS 的值是否准确, 一个不准确的 TENMS 值会影响 SysTick 作为软件实时时钟的匹配度。 0: TENMS 的值是准确的; 1: TENMS 的值不准确, 或者不存在。	R	0
29:24	Reserved		R	0
23:0	TENMS	为 10ms 定时作用的重装数值, 同时受系统时钟偏差影响。如果这个值读数为 0, 则此校准值是不确定的	R/W	0xA71FF

2.3 嵌套向量中断控制器（NVIC）

包括内核异常在内的所有中断，都由 NVIC 来管理。NVIC 的特性如下：

1. NVIC 支持 32 个向量中断；
2. 带有硬件掩膜优先级的 4 级可编程中断优先级；
3. 快速响应异常情形和中断处理；
4. 对延迟中断的有效处理；
5. 对系统控制寄存器的操作；
6. 产生软件中断。

2.3.1 中断和异常向量

Execution No.	Priority	Function	Description	Address Offset
0	-	-	Reserved	0x0000 0000
1	-3	Reset	Reset	0x0000 0004
2	-2	NMI_Handler	Non maskable interrupt.	0x0000 0008
3	-1	HardFault_Handler	All class of fault	0x0000 000C
4~10	Reserved	Reserved	Reserved	-
11	Settable	SVCCall		0x0000 002C
12~13	Reserved	Reserved	Reserved	-
14	Settable	PendSV		0x0000 0038
15	Settable	SysTick		0x0000 003C
16	Settable	IRQ0/NDTIRQ	NDT	0x0000 0040
17	Settable	IRQ1/USBIRQ	USB	0x0000 0044
18	Settable	IRQ2/LCDIRQ	LCD	0x0000 0048
19	Settable	IRQ3/I2SIRQ	I2S	0x0000 004C
20	Settable	IRQ4/		0x0000 0050
21	Settable	IRQ5/		0x0000 0054
22	Settable	IRQ6/SSP0IRQ	SSP0	0x0000 0058
23	Settable	IRQ7/SSP1IRQ	SSP1	0x0000 005C
24	Settable	IRQ8/		0x0000 0060
25	Settable	IRQ9/		0x0000 0064
26	Settable	IRQ10/I2C0IRQ	I2C0	0x0000 0068
27	Settable	IRQ11/I2C1IRQ	I2C1	0x0000 006C
28	Settable	IRQ12/		0x0000 0070
29	Settable	IRQ13/USART0IRQ	USART0	0x0000 0074
30	Settable	IRQ14/USART1IRQ	USART1	0x0000 0078
31	Settable	IRQ15/CT16B0IRQ	CT16B0	0x0000 007C
32	Settable	IRQ16/CT16B1IRQ	CT16B1	0x0000 0080
33	Settable	IRQ17/CT16B2IRQ	CT16B2	0x0000 0084
34	Settable	IRQ18/		0x0000 0088
35	Settable	IRQ19/CT32B0IRQ	CT32B0	0x0000 008C
36	Settable	IRQ20/CT32B1IRQ	CT32B1	0x0000 0090
37	Settable	IRQ21/CT32B2IRQ	CT32B2	0x0000 0094
38	Settable	IRQ22/		0x0000 0098
39	Settable	IRQ23/RTCIRQ	RTC	0x0000 009C
40	Settable	IRQ24/ADCIRQ	ADC	0x0000 00A0
41	Settable	IRQ25/WDTIRQ	WDT	0x0000 00A4
42	Settable	IRQ26/LVDIRQ	LVD	0x0000 00A8
43	Settable	IRQ27/		0x0000 00AC
44	Settable	IRQ28/P3IRQ	GPIO interrupt status of port 3	0x0000 00B0
45	Settable	IRQ29/P2IRQ	GPIO interrupt status of port 2	0x0000 00B4
46	Settable	IRQ30/P1IRQ	GPIO interrupt status of port 1	0x0000 00B8
47	Settable	IRQ31/P0IRQ	GPIO interrupt status of port 0	0x0000 00BC

2.3.2 NVIC 寄存器

2.3.2.1 IRQ0~31 中断设置使能寄存器 (NVIC_ISER)

地址: 0xE000 E100 (参考 Cortex-M0 规格)

ISER 使能中断, 并显示使能的中断。

Bit	Name	Description	Attribute	Reset
31:0	SETENA[31:0]	中断设置使能位。 写→0: 无影响; 1: 使能中断。 读→0: 禁止中断; 1: 使能中断。	R/W	0

2.3.2.2 IRQ0~31 中断清零使能寄存器 (NVIC_ICER)

地址: 0xE000 E180 (参考 Cortex-M0 规格)

ICER 禁止中断, 并显示使能的中断。

Bit	Name	Description	Attribute	Reset
31:0	CLRENA[31:0]	中断清零使能位。 写→0: 无影响; 1: 禁止中断。 读→0: 禁止中断; 1: 使能中断。	R/W	0

2.3.2.3 IRQ0~31 中断挂起设置寄存器 (NVIC_ISPR)

地址: 0xE000 E200 (参考 Cortex-M0 规格)

ISPR 迫使中断进入挂起状态, 并显示正在挂起的中断。

- * 注: 写 1 到 ISPR 位需符合:
- 1、对正处于挂起状态的中断请求无影响;
 - 2、禁止的中断设置其状态为挂起状态。

Bit	Name	Description	Attribute	Reset
31:0	SETPEND[31:0]	中断挂起设置位。 写→0: 无影响; 1: 改变中断状态为挂起状态 读→0: 中断未处于挂起状态; 1: 中断处于挂起状态	R/W	0

2.3.2.4 IRQ0~31 中断挂起清零寄存器 (NVIC_ICPR)

地址: 0xE000 E280 (参考 Cortex-M0 规格)

ICPR 可以取消中断的挂起状态, 并显示正在挂起的中断。

- * 注: 写入 1 到 ICPR, 并不影响相应中断的有效状态。

Bit	Name	Description	Attribute	Reset
31:0	CLRPEND[31:0]	中断挂起清除位。 写→0: 无影响; 1: 禁止中断的挂起状态。 读→0: 中断没有处于挂起状态; 1: 中断处于挂起状态。	R/W	0

2.3.2.5 IRQ0~31 中断优先级寄存器 (NVIC_IPRn) (n=0~7)

地址: 0xE000 E400 + 0x4 * n (参考 Cortex-M0 规格)

中断优先级寄存器为每个中断提供 8-bit 优先权选择级别, 每个寄存器管理 4 个中断源的优先权。这意味着寄存器的数量是明确定义的, 且与执行的中断的数量相对应。

Bit	Name	Description	Attribute	Reset
31:24	PRI_(4*n+3)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 bits[31:30], 而 bits[29:24]读出为 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192	R/W	0
23:16	PRI_(4*n+2)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 bits[23:22], 而 bits[21:16]读出为 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192	R/W	0
15:8	PRI_(4*n+1)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 bits[15:14], 而 bits[13:8]读出为 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192	R/W	0
7:0	PRI_4*n	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 bits[7:6], 而 bits[5:0]读出为 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192	R/W	0

2.4 应用中断和复位控制 (AIRC)

地址: 0xE000 ED0C (参考 Cortex-M0 规格)

整个 MCU, 包括内核, 都可以通过软件设置 AIRC 寄存器的 SYSRESREQ 位来复位。

* 注: 在写入数据到该寄存器时, 用户必须同时写入 0x05FA 到 VECTKEY, 否则处理器会忽略此次写动作。

Bit	Name	Description	Attribute	Reset
31:16	VECTKEY	寄存器关键字。 读取为未知数; 写入 0x05FA 到 VECTKEY, 否则写动作无效。	R/W	0
15	ENDIANESS	数据字节顺序应用位。 0: 小端排列; 1: 大端排列。	R	0
14:3	Reserved		R	0
2	SYSRESETREQ	系统复位请求位, 该位读取为 0。 0: 没有影响; 1: 请求系统复位。	W	0
1	VECTCLRACTIVE	保留以用来调试, 该位读取为 0。写入数据到寄存器时, 用户必须写入 0 到该位, 否则会发生未知情形。	W	0
0	Reserved		R	0

2.5 编译选项列表 (CODE OPTION)

地址: 0x1FFF 2000

Bit	Name	Description	Attribute	Reset
31:16	Code Security[15:0]	代码加密。 0xFFFF: CS0; 0x5A5A: CS1; 0xA5A5: CS2; 0x55AA: CS3。	R/W	0xFFFF
15:0	Reserved		R	All 1

2.6 唯一码

唯一码是每个 IC 的一个 8 字节唯一设备序列号。换句话说，每个 IC 的唯一码都是不同的，不连续的。用户可以在 RF 应用中使用唯一码作为配对码，或作为 USB 序列号使用。

地址: 0x1FFF 2C08

Bit	Name	Description	Attribute	Reset
31:0	L4BYTE[31:0]	唯一码低 4Bytes	R/W	0x0000

地址: 0x1FFF 2C0C

Bit	Name	Description	Attribute	Reset
31:0	H4BYTE[31:0]	唯一码高 4Bytes	R/W	0x0000

3 系统控制

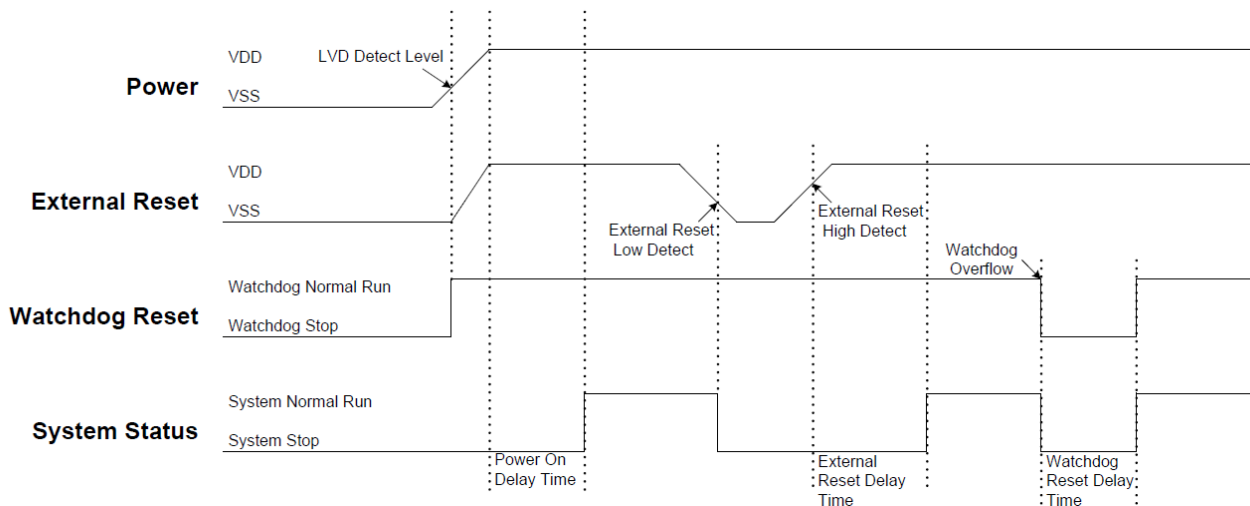
3.1 复位

SN32F240 系列单片机的复位方式有以下几种：

- 低电平到 RST 引脚触发复位（外部复位）；
- 上电复位（POR 复位）
- LVD 复位；
- 看门狗定时器复位（WDT 复位）；
- 软件复位（SW 复位）；

系统复位源可通过系统复位状态寄存器（SYS0_RSTST）的不同标志位来识别。复位源作用于 RST 引脚，且 RST 引脚在延迟阶段保持低电平。复位程序的复位向量地址位于寄存器的 0x00000004 处。详情请参考中断和 Exception 向量。

系统复位需要一定的时间，并提供完整的上电复位规程。对于不同类型的振荡器，完成复位所需要的时间也不同。因此，VDD 的上升速度和不同晶振的起振时间都不固定。RC 振荡器的启动时间非常短，晶体振荡器的启动时间则比较长。在用户终端使用的过程中，应注意考虑主机对上电复位时间的要求。下面给出了复位时序图。



3.1.1 上电复位（POR）

上电复位与 LVD 操作密切相关。系统上电过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。上电复位的时序如下：

- **上电：**系统检测到电源电压上升并等待其稳定；
- **外部复位（使能外部复位引脚时才有效）：**系统检测外部复位引脚状态。如果不为高电平，系统保持复位状态直到外部复位引脚的复位结束。
- **系统初始化：**所有的系统寄存器被置为默认状态；
- **振荡器开始工作：**振荡器开始提供系统时钟；
- **执行程序：**上电结束后，程序从引导装载程序处开始运行。

3.1.2 看门狗复位 (WDT RESET)

看门狗复位是系统的一种保护设置。在正常状态下，由程序将看门狗定时器清零。若出错，系统处于未知状态，程序无法清除看门狗，导致看门狗定时器溢出，此时系统复位。看门狗复位后，系统重启进入正常状态。看门狗复位的时序如下：

- **看门狗定时器状态：**系统检测看门狗定时器是否溢出，若溢出，则系统复位；
- **系统初始化：**初始化所有的系统寄存器；
- **振荡器起振：**振荡器正常工作并提供系统时钟；
- **执行程序：**上电复位结束，程序从地址 0x00 开始执行。

看门狗定时器应用时需注意：

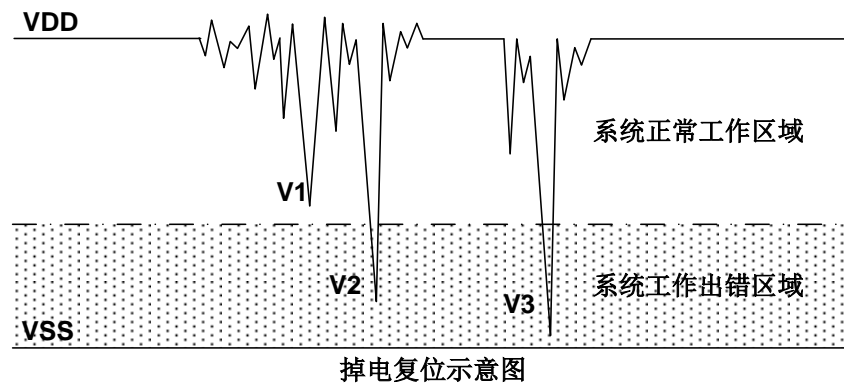
- 清看门狗定时器之前，请先检查 I/O 口的状态和 RAM 数据，可降低系统错误率；
- 不能在中断里清看门狗定时器，否则无法起到侦测程序跑飞的目的；
- 程序中应该只有一个清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。

* 注：关于看门狗定时器的详细内容，请参阅看门狗定时器有关章节。

3.1.3 掉电复位

3.1.3.1 掉电复位简述

掉电复位针对外部因素引起的系统电压跌落情形（例如：干扰或外部负载的变化），掉电复位可能会引起系统工作状态不正常或程序执行错误。



电压跌落可能会进入系统死区。系统死区意味着电源不能满足系统的最小工作电压要求。上图是一个典型的掉电复位示意图。图中，VDD 受到严重的干扰，电压值降的非常低。虚线以上区域系统正常工作，在虚线以下的区域内，系统进入未知的工作状态，这个区域称作死区。当 VDD 跌至 V1 时，系统仍处于正常状态；当 VDD 跌至 V2 和 V3 时，系统进入死区，则容易导致出错。以下情况系统可能进入死区：

DC 运用中：

DC 运用中一般都采用电池供电，当电池电压过低或单片机驱动负载时，系统电压可能跌落并进入死区。这时，电源不会进一步下降到 LVD 检测电压，因此系统维持在死区。

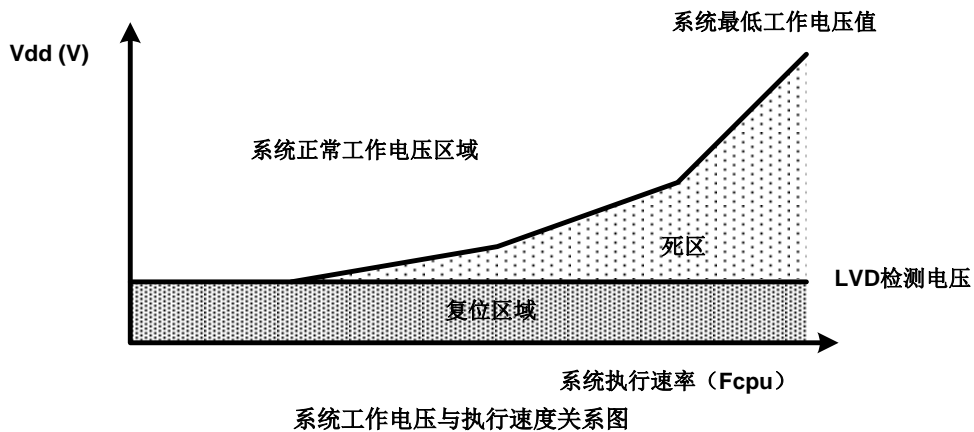
AC 运用中：

系统采用 AC 供电时，DC 电压值受 AC 电源中的噪声影响。当外部负载过高，如驱动马达时，负载动作产生的干扰也影响到 DC 电源。VDD 若由于受到干扰而跌落至最低工作电压以下时，则系统将有可能进入不稳定工作状态。

在 AC 运用中，系统上、下电时间都较长。其中，上电时序保护使得系统正常上电，但下电过程却和 DC 运用中情形类似，AC 电源关断后，VDD 电压在缓慢下降的过程中易进入死区。

3.1.3.2 系统工作电压

为了改善系统掉电复位的性能，首先必须明确系统具有的最低工作电压值。系统最低工作电压与系统执行速度有关，不同的执行速度下最低工作电压值也不同。



如上图所示，系统正常工作电压区域一般高于系统复位电压，同时复位电压由低电压检测（LVD）电平决定。当系统执行速度提高时，系统最低工作电压也相应提高，但由于系统复位电压是固定的，因此在系统最低工作电压与系统复位电压之间就会出现一个电压区域，系统不能正常工作，也不会复位，这个区域即为死区。

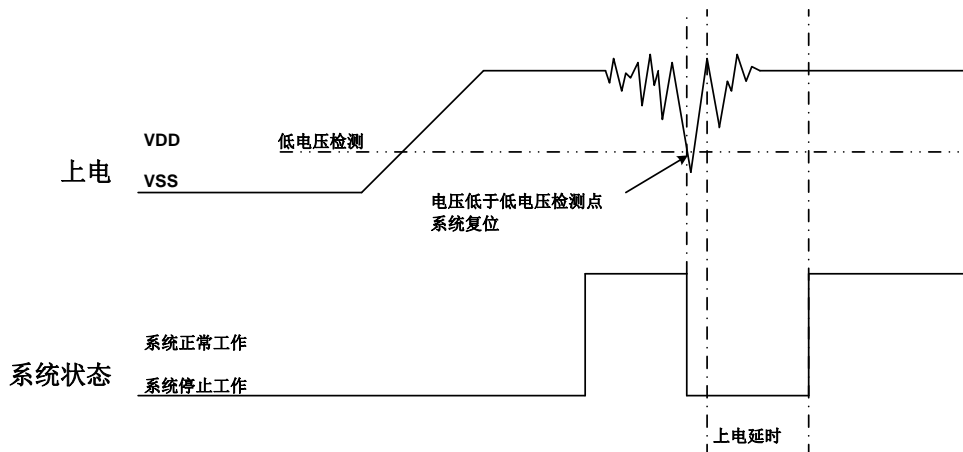
3.1.3.3 掉电复位性能改进

如何改善系统掉电复位性能，有以下几点建议：

- LVD 复位；
- 看门狗复位；
- 降低系统工作速度；
- 采用外部复位电路（稳压二极管复位电路，电压偏移复位电路，外部 IC 复位）。

* 注：“稳压二极管复位电路”、“电压偏移复位电路”和“外部 IC 复位”能够完全避免掉电复位出错。

LVD 复位：



低电压检测（LVD）是 SONiX 32 位单片机内置的掉电复位保护装置，当 VDD 跌落并低于 LVD 检测电压值时，LVD 给出一个中断信号到 NVIC，以便触发 CPU 中断；若不能触发中断，则 SW 通过读取寄存器的显示状态来监控该信号。可以选择一个附加的极限值强制系统复位。不同的单片机有不同的 LVD 检测电平，LVD 检测电平值仅为一个电压点，并不能覆盖所有死区范围。因此采用 LVD 依赖于系统要求和环境状况。电源变化较大时，LVD 能够起到保护作用，如果电源变化触发 LVD，系统工作仍出错，那么 LVD 就不能起到保护作用，就需要采用其它复位方法。更多 LVD 信息请参考电气特性。

看门狗复位:

看门狗定时器用于保证系统正常工作。通常,会在主程序中将看门狗定时器清零,但不要在多个分支程序中清除看门狗。若程序正常运行,看门狗不会复位。当系统进入死区或程序运行出错的时候,看门狗定时器继续计数直至溢出,系统复位。

如果看门狗复位后电源仍处于死区,则系统复位失败,保持复位状态,直到系统工作电压恢复到正常值。

降低系统工作速度:

系统工作速度越快最低工作电压值越高,从而加大工作死区的范围,因此降低系统工作速度不失为降低系统进入死区几率的有效措施。所以,可选择合适的工作速度以避免系统进入死区,这个方法需要调整整个程序使其满足系统要求。

附加外部复位电路:

外部复位也能够完全改善掉电复位性能。有三种外部复位方式可改善掉电复位性能:稳压二极管复位电路,电压偏移复位电路和外部 IC 复位。它们都采用外部复位信号控制单片机可靠复位。

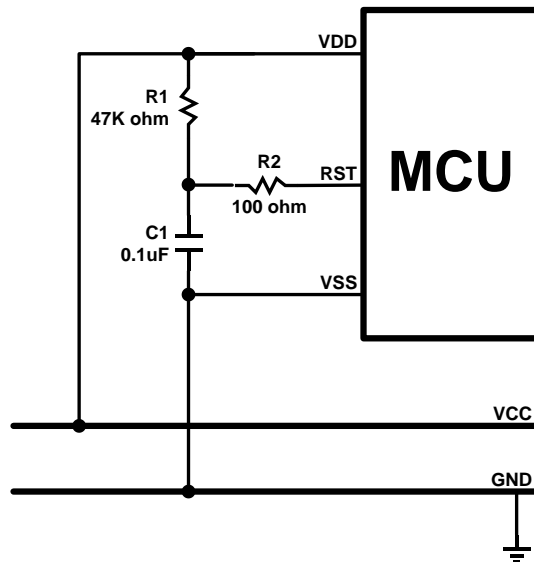
3.1.4 外部复位

外部复位功能由外部复位引脚控制寄存器(SYS0_EXRSTCTRL)控制,初始值为1,即使能外部复位功能。外部复位引脚为施密特触发结构,低电平有效。复位引脚处于高电平时,系统正常运行。当复位引脚输入低电平信号时,系统复位。外部复位操作在上电和正常工作模式时有效。需要注意的是,在系统上电完成后,外部复位引脚必须输入高电平,否则系统将一直保持在复位状态。外部复位的时序如下:

- **外部复位(当且仅当外部复位引脚为使能状态):** 系统检测复位引脚的状态,如果复位引脚不为高电平,则系统会一直保持在复位状态,直到外部复位结束;
- **系统初始化:** 所有的系统寄存器被置为初始状态;
- **振荡器开始工作:** 振荡器开始提供系统时钟;
- **执行程序:** 上电结束后,程序从引导装载程序处开始运行。

外部复位可以在上电过程中使系统复位。良好的外部复位电路可以保护系统以免进入未知的工作状态,如 AC 应用中的掉电复位等。

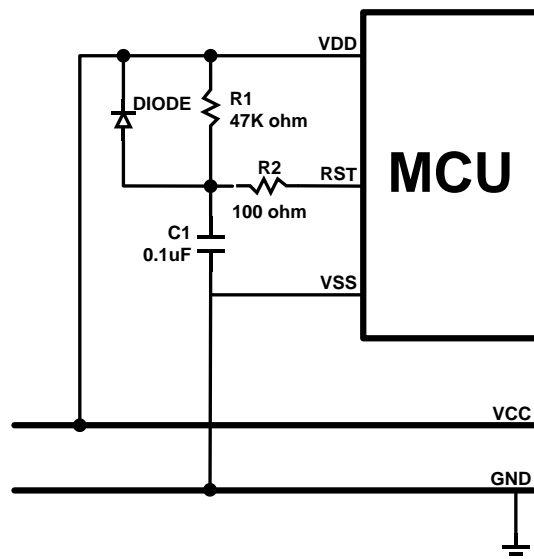
3.1.4.1 RC 复位电路



上图为一个由电阻 R1 和电容 C1 组成的基本 RC 复位电路，它在系统上电的过程中能够为复位引脚提供一个缓慢上升的复位信号。这个复位信号的上升速度低于 VDD 的上电速度，为系统提供合理的复位时序，当复位引脚检测到高电平时，系统复位结束，进入正常工作状态。

* 注：此 RC 复位电路不能解决非正常上电和掉电复位问题。

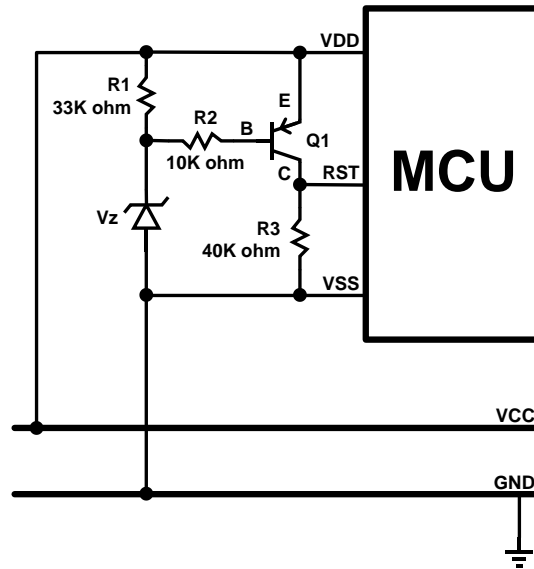
3.1.4.2 二极管&RC 复位电路



上图中，R1 和 C1 同样是为复位引脚提供输入信号。对于电源异常情况，二极管正向导通使 C1 快速放电并与 VDD 保持一致，避免复位引脚持续高电平、系统无法正常复位。

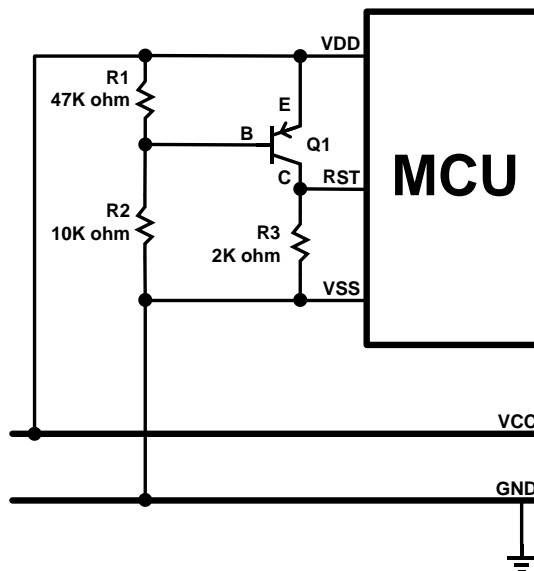
* 注：“基本 RC 复位电路”和“二极管及 RC 复位电路”中的电阻 R2 都是必不可少的限流电阻，以避免复位引脚 ESD (Electrostatic Discharge) 或 EOS (Electrical Over-stress) 击穿。

3.1.4.3 稳压二极管复位电路



稳压二极管复位电路是一种简单的 LVD 电路，基本上可以完全解决掉电复位问题。如上图电路中，利用稳压管的击穿电压作为电路复位检测值，当 VDD 高于“ $V_z + 0.7V$ ”时，三极管集电极输出高电平，单片机正常工作；当 VDD 低于“ $V_z + 0.7V$ ”时，三极管集电极输出低电平，单片机复位。稳压管规格不同则电路复位检测值不同，根据电路的要求选择合适的二极管。

3.1.4.4 电压偏置复位电路

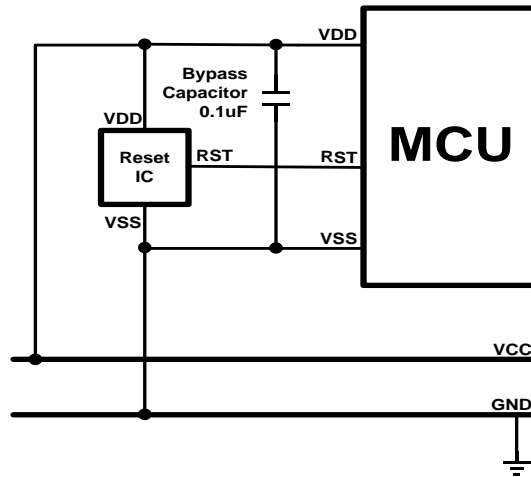


电压偏置复位电路是一种简单的 LVD 电路，基本上可以完全解决掉电复位问题。与稳压二极管复位电路相比，这种复位电路的检测电压值的精确度有所降低。电路中，R1 和 R2 构成分压电路，当 VDD 高于和等于分压值“ $0.7V \times (R1 + R2) / R1$ ”时，三极管集电极 C 输出高电平，单片机正常工作；VDD 低于“ $0.7V \times (R1 + R2) / R1$ ”时，集电极 C 输出低电平，单片机复位。

对于不同应用需求，选择适当的分压电阻。单片机复位引脚上电压的变化与 VDD 电压变化之间的差值为 0.7V。如果 VDD 跌落并低于复位引脚复位检测值，那么系统将被复位。如果希望提升电路复位电平，可将分压电阻设置为 $R2 > R1$ ，并选择 VDD 与集电极之间的结电压高于 0.7V。分压电阻 R1 和 R2 在电路中要耗电，此处的功耗必须计入整个系统的功耗中。

* 注：在电源不稳定或掉电复位的情况下。“稳压二极管复位电路”和“偏压复位电路”能够保护电路在电压跌落时避免系统出错。当电压跌落至低于复位检测值时，系统将被复位。从而保证系统正常工作。

3.1.4.5 外部 IC 复位电路



外部 IC 复位可以增强 MCU 复位的可靠性。

3.1.5 软件复位

整个单片机，包括内核部分，都可以由软件通过设置 SHYSRESREQ 位（在 Cortex-M0 规格中位于 AIRC—应用中断和复位控制寄存器中）来复位。

软件初始化系统复位的流程如下：

- 1、通过设置 SYSRESREQ 位初始化；
- 2、宣告内部复位；
- 3、内部复位完成后，单片机从存储器加载初始的堆栈指针，初始的程序计数器，并由程序计数器指定第一条指令，然后开始执行程序。

3.2 系统时钟

不同的时钟源都可用来驱动系统时钟（SYSCLK）：

- 12MHz 内部高速 RC 时钟（IHRC）；
- 32KHz 内部低速 RC 时钟（ILRC）；
- PLL 时钟；
- 外部高速晶体时钟（EHS）；
- 外部低速 32.768KHz 晶体时钟（ELS）。

每个时钟源在不使用时都可以独立的打开或关闭，以使功耗达到最优状态。

该单片机为双时钟系统，包括高速时钟和低速时钟。高速时钟由外部振荡电路和内部 PLL 电路提供；低速时钟则由内部低速 RC 振荡电路（ILRC 32KHz）提供。

3.2.1 内部 RC 时钟源

3.2.1.1 内部高速 RC 振荡器（IHRC）

内部高速 12MHz RC 振荡器，在普通环境下，其精确度为 $\pm 2\%$ 。通过模拟模块控制寄存器（SYS0_ANBCTRL）中的 IHRCEN 位，可以使 IHRC 自由切换是否使能。

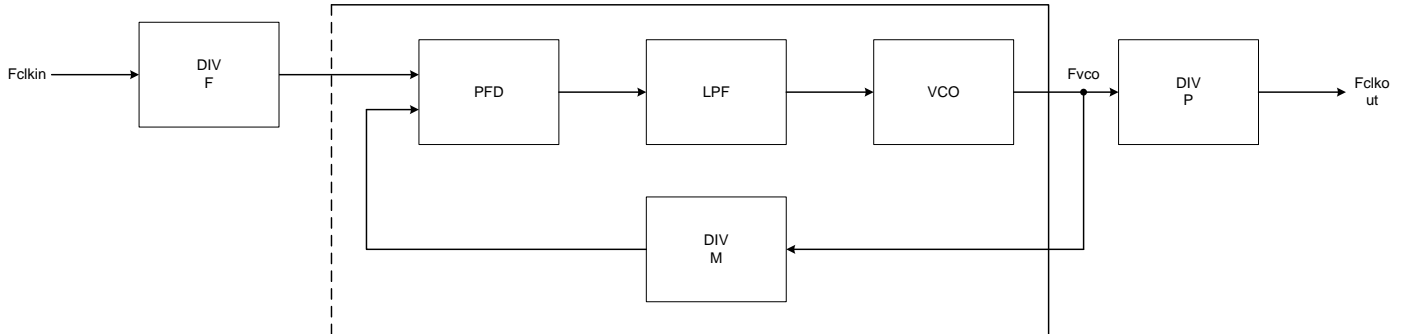
3.2.1.2 内部低速 RC 振荡器（ILRC）

系统低速时钟源即内置的低速振荡器，采用 RC 振荡电路。低速时钟的输出频率受系统电压和环境温度的影响，通常为 32KHz。

* 注：只能通过 HW 来切换 ILRC 的状态。

3.2.2 PLL

SONiX 32 位单片机利用 PLL 电路产生时钟，以驱动内核部分和外设部分。输入频率为 10MHz~25MHz，输入时钟通过分频提供给相位频率检测器（PFD）。PFD 比较输入的相位和频率，若它们不匹配，则产生一个控制信号。环路滤波器滤除控制信号，并驱动电压控制振荡器（VCO）。产生一个主时钟和两个任意相位。VCO 频率为 156MHz~320MHz。主时钟由 P 通过可编程的 post 分频器进行分频，产生输出时钟（S）。VCO 输出时钟由 M 通过可编程的 feedback 分频器分频，产生 feedback 时钟。相位频率检测器的输出信号由 Lock 检测器进行监控，当 PLL 锁住时，发出信号到输入时钟。PLL 的稳定时间为 100us。



3.2.2.1 PLL 频率选择

PLL 频率方程式如下：

$$F_{VCO} = F_{CLKIN} / F * M$$

$$F_{CLKOUT} = F_{VCO} / P$$

PLL 的频率由下列参数决定：

- F_{CLKIN} : PLLCLKSEL 多路器频率；
- F_{VCO} : 电压控制振荡器（VCO）频率，156~320MHz；
- F_{CLKOUT} : PLL 输出频率；
- P: 系统 PLL post 分频器比例，由 PLL 控制寄存器（SYS0_PLLCTRL）的 PSEL 位控制；
- F: 系统 PLL front 分频器比例，由 PLL 控制寄存器（SYS0_PLLCTRL）的 FSEL 位控制；
- M: 系统 PLL feedback 分频器比例，由 PLL 控制寄存器（SYS0_PLLCTRL）的 MSEL 位控制。

为给 M、P 和 F 选择合适的值，建议按照下列条件选择：

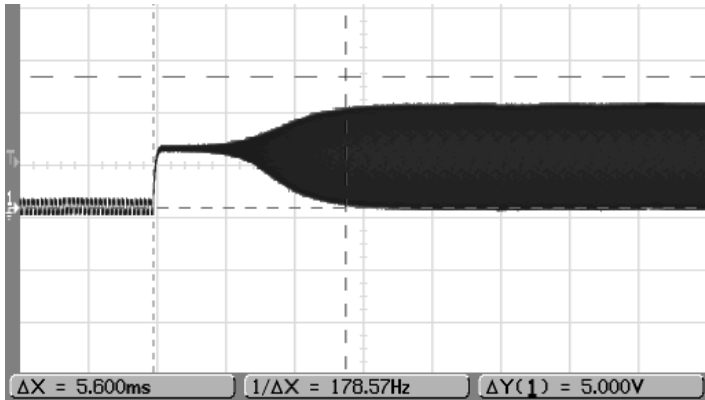
- 1、 $10\text{MHz} \leq F_{CLKIN} \leq 25\text{MHz}$
- 2、 $150\text{MHz} \leq F_{VCO} \leq 330\text{MHz}$
- 3、 $2 < M \leq 31$
- 4、 $F = 1, \text{ or } 2$
- 5、 $P = 6, 8, 10, 12, \text{ or } 14$ (duty 50% +/- 2.5%)
- 6、 $F_{CLKOUT} = 20\text{MHz}, 30\text{MHz}, 40\text{MHz}, 50\text{MHz}, 24\text{MHz}, 36\text{MHz}, 48\text{MHz}, 32\text{MHz}, 22\text{MHz}, 24\text{MHz}, 50\text{MHz}$ with jitter < ±500 ps

3.2.3 外部时钟源

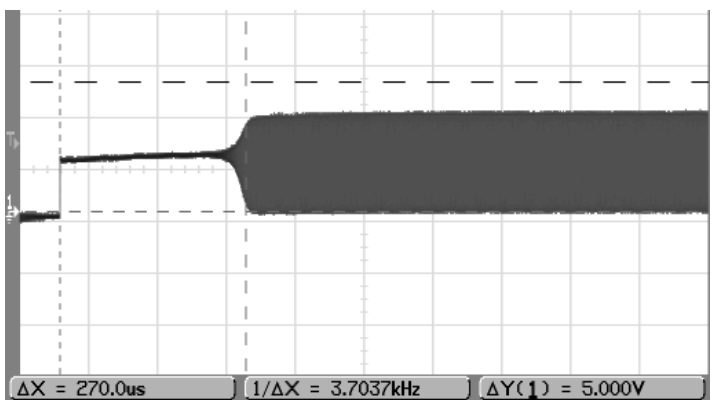
3.2.3.1 外部高速时钟（EHS）

外部高速时钟包括石英/陶瓷模块，启动时间比较长。系统的复位时间与振荡器的启动时间息息相关。

4MHz Crystal

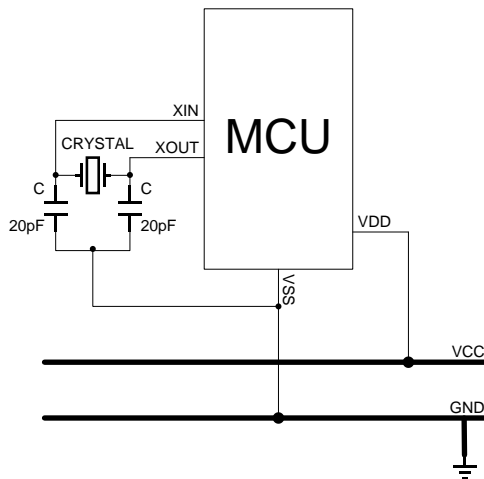


4MHz Ceramic



3.2.3.2 石英/陶瓷振荡器

石英/陶瓷振荡器通过单片机的 XIN、XOUT 引脚驱动，针对不同的模式（高/普通/低频模式），其驱动电流也是不同的。



* 注：石英/陶瓷振荡器和电容 C 要尽可能的靠近单片机的 XIN/XOUT/VSS 引脚。

- 结构：1MHz~25MHz EHS 外部石英/陶瓷振荡器。
- 主要用途：系统高速时钟源，RTC 时钟源，PLL 时钟源。
- Warm-up 时间：2048*FEHS。
- XIN/XOUT 共用引脚列表：

振荡器模式	XTALIN 引脚	XTALOUT 引脚
IHRC	GPIO	GPIO
EHS X'TAL	石英/陶瓷振荡器	石英/陶瓷振荡器

振荡器和负载电容必须尽可能的靠近单片机的振荡器驱动引脚，以便输出最稳定的输出信号。负载电容的电容值必须与所选择的振荡器相匹配。

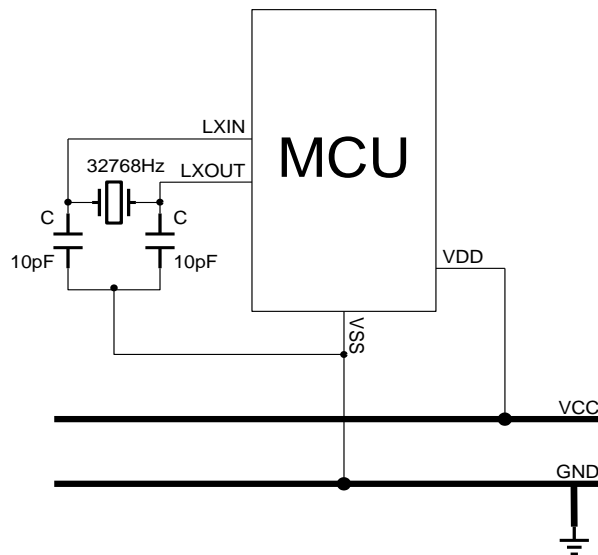
通过模拟模块控制寄存器（SYS0_ANBCTRL）的 EHSEN 位可以控制 EHS 晶体的状态。

3.2.3.3 外部低速时钟（ELS）

低速振荡器采用 32768Hz 晶体振荡电路。

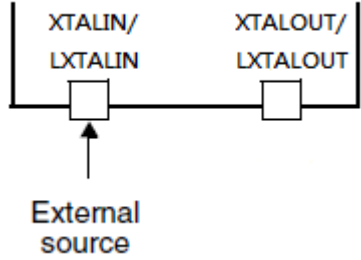
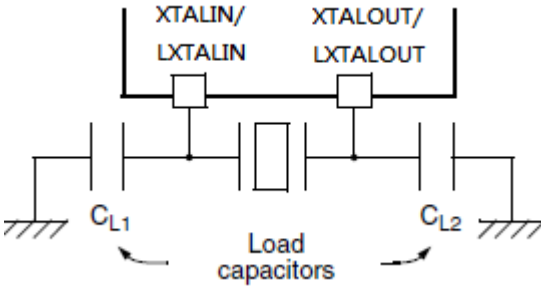
3.2.3.4 CRYSTAL

晶体设备由单片机的 LXIN、LXOUT 引脚驱动。32768 晶体和 10pF 的电容必须尽可能的靠近单片机。通过模拟模块控制寄存器（SYS0_ANBCTRL）的 ELSEN 位控制 ELS 晶体的状态。



* 注：晶体/陶瓷振荡器和电容 C 必须尽可能的靠近单片机的 LXIN/LXOUT/VSS 引脚，LXIN/XLOUT 与 VSS 之间的电容 C 的值必须为 10pF。

3.2.3.5 旁路模式

时钟源	H/W 配置	说明
外部时钟源（旁路）		<p>旁路模式下，外部时钟信号以~50%占空比的周期提供给 XTALIN/LXTALIN，而 XTALOUT/LXTALOUT 脚应该是输入信号的反向信号。外部高速晶体能到 25M，通过模拟模块控制寄存器 SYS0_ANBCTRL 的 EHSEN 位选择此模式</p> <p>外部低速晶体使用 32.678K，通过寄存器 SYS0_ANBCTRL 的 ELSEN 位选择此模式</p>
外部 X'TAL (EHS/ELS X'TAL)		<p>以主时钟为基础，10-25MHz 的 EHS X'TAL 可以产生一个非常精确的比例。</p> <p>ELS X'TAL 的频率必须为 32.768KHz。</p>

3.2.4 系统时钟（SYSCLK）选择

系统复位后，IHRC 作为系统时钟。当直接使用时钟源或者凭借 PLL 作为系统时钟时，IHRC 仍然正常工作。

若一个时钟源转换到另一个时钟源时，需等待目标时钟源准备好（启动延迟后时钟稳定或者锁定 PLL）。若选择一个未准备好的时钟源，则需等待其准备好之后才会开始转换。

SYS0_CSST 寄存器的 Ready 位显示已经准备好的时钟源，SYS0_CLKCFG 寄存器的 SYSCLKST 位则显示的是当前作为系统时钟的时钟源。

3.2.5 CLOCK-OUT 功能

单片机时钟输出（CLKOUT）性能允许时钟输出到外部 CLKOUT 引脚。此对应 GPIO 的配置寄存器需设定成时钟输出功能。

6 种时钟信号可以作为时钟输出：

- HCLK
- IHRC
- ILRC
- PLL 时钟输出
- ELS X'TAL
- EHS X'TAL

由 SYS1_AHBCLKEN 寄存器的 CLKOUTSEL 位选择控制。

3.3 系统控制寄存器 0

基地址：0x4006 0000

3.3.1 模拟模块控制寄存器 (SYS0_ANBCTRL)

地址偏移量：0x00

复位值：0x0000 0001

* 注：若选择 EHS X'tal/ELS X'tal/IHRC 作为系统时钟或者准备作为系统时钟，则 EHSEN/ELSEN/IHRCEN 位不能被清零。

Bit	Name	说明	Attribute	Reset
31:6	Reserved		R	0
5	EHSFREQ	EHS X'TAL 的频率范围（驱动能力）。 0: <=12MHz; 1: >12MHz。	R/W	0
4	EHSEN	外部高速时钟使能位。 0: 禁止 EHS X'TAL; 1: 使能 EHS X'TAL。	R/W	0
3	Reserved		R	0
2	ELSEN	外部低速振荡器使能位。 0: 禁止外部 32.768KHz 振荡器; 1: 使能外部 32.768KHz 振荡器。	R/W	0
1	Reserved		R	0
0	IHRCEN	内部高速时钟使能位。 0: 禁止内部 12MHz RC 振荡器; 1: 使能内部 12MHz RC 振荡器。	R/W	1

3.3.2 PLL 控制寄存器 (SYS0_PLLCTRL)

地址偏移量: 0x04

* 注: 若选择 PLL 作为系统时钟或者准备作为系统时钟, 则 PLEN 位不能被清零。

Bit	Name	说明	Attribute	Reset
31:16	Reserved		R	0
15	PLEN	PLL 使能位。 0: 禁止; 1: 使能。	R/W	0
14	Reserved		R	0
13:12	PLLCLKSEL[1:0]	系统 PLL 时钟源。 00: IHRC 12 MHz 振荡器; 01: EHS X'TAL 10 MHz ~ 25 MHz; 其它: 保留。	R/W	0
11:9	Reserved		R	0
8	FSEL	Front 分频值, 分频值 F 是可编程控制的, 为 2^{FSEL} 。 0: F = 1; 1: F = 2。	R/W	0
7:5	PSEL[2:0]	Post 分频值, P = PSEL[2:0]*2。 000~010: 保留; 011: P = 6; 100: P = 8; 101: P = 10; 110: P = 12; 111: P = 14。	R/W	011b
4:0	MSEL[4:0]	Feedback 分频值。 M: 3~31。	R/W	0x3

为给 M、P 和 F 选择合适的值, 建议按照下列条件选择:

- 10MHz $\leq F_{CLKIN} \leq 25$ MHz
- 150MHz $\leq F_{VCO} \leq 330$ MHz
- $2 < M \leq 31$
- F = 1, or 2
- P = 6, 8, 10, 12, or 14 (duty 50% +/- 2.5%)
- $F_{CLKOUT} = 20$ MHz, 30MHz, 40MHz, 50MHz, 24MHz, 36MHz, 48MHz, 32MHz, 22MHz, 24MHz, 50MHz
with jitter $< \pm 500$ ps

	Fckout	10M Hz	12M Hz	16M Hz	20M Hz	22M Hz	24M Hz	25M Hz	30M Hz	32M Hz	36M Hz	40M Hz	44M Hz	48M Hz	50M Hz
Fckin															
10M Hz					V				V			V			V
12M Hz							V				V			V	
16M Hz										V				V	
22M Hz													V		
24M Hz														V	
25M Hz															V

3.3.2.1 频率的推荐设置

$$F_{VCO} = F_{CLKIN} / F * M$$

$$F_{CLKOUT} = F_{VCO} / P$$

FCLKIN (MHz)	FSEL	F=2FEL	MSEL[4:0]=M	FVCO (MHz) =FCLKIN / F*M	PSEL[2:0]	P= PSEL[2:0]*2	FCLKOUT (MHz)
10	0	1	20	200	5	10	20
10	0	1	22	220	5	10	22
10	0	1	18	180	3	6	30
10	0	1	24	240	3	6	40
10	0	1	30	300	3	6	50
12	0	1	16	192	4	8	24
12	0	1	18	216	3	6	36
12	0	1	24	288	3	6	48
12	0	1	25	300	3	6	50
16	0	1	16	256	4	8	32
16	0	1	18	288	3	6	48
20	1	2	30	300	3	6	50
22	0	1	12	264	3	6	44
24	0	1	12	288	3	6	48
25	0	1	12	300	3	6	50

3.3.3 时钟源状态寄存器 (SYS0_CSST)

地址偏移量: 0x08

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6	PLLRDY	PLL 时钟准备标志位。 0: PLL 未锁定; 1: PLL 被锁定。	R	0
5	Reserved		R	0
4	EHSRDY	外部高速时钟准备标志位。 0: EHS 振荡器没有准备好; 1: EHS 振荡器已经准备好。	R	0
3	Reserved		R	0
2	ELSRDY	外部低速时钟准备标志位。 0: ELS 振荡器没有准备好; 1: ELS 振荡器已经准备好。	R	0
1	Reserved		R	0
0	IHRCRDY	IHRC 时钟准备标志位。 0: IHRC 没有准备好; 1: IHRC 已经准备好。	R	1

3.3.4 系统时钟配置寄存器 (SYS0_CLKCFG)

地址偏移量: 0x0C

Bit	Name	说明	Attribute	Reset
31:7	Reserved		R	0
6:4	SYSCLKST[2:0]	系统时钟切换状态。 通过 HW 设置和清零，以显示当前作为系统时钟的时钟源。 000: IHRC 作为系统时钟; 001: ILRC 作为系统时钟; 010: EHS X'TAL 作为系统时钟; 011: ELS X'TAL 作为系统时钟; 100: PLL 作为系统时钟; 其它: 保留。	R	0
3	Reserved		R	0
2:0	SYSCLKSEL[2:0]	系统时钟切换。通过 SW 设置和清零。 000: IHRC; 001: ILRC; 010: EHS X'TAL; 011: ELS X'TAL; 100: PLL 输出; 其它: 保留。	R/W	0

3.3.5 AHB 时钟预分频寄存器 (SYS0_AHBCP)

地址偏移量: 0x10

Bit	Name	说明	Attribute	Reset
31:4	Reserved		R	0
3:0	AHBPRES[3:0]	AHB 时钟源预分频值。 0000: SYSCLK / 1; 0001: SYSCLK / 2; 0010: SYSCLK / 4; 0011: SYSCLK / 8; 0100: SYSCLK / 16; 0101: SYSCLK / 32; 0110: SYSCLK / 64; 0111: SYSCLK / 128; 1000: SYSCLK / 256; 1001: SYSCLK / 512; 其它: 保留。	R/W	0

3.3.6 系统复位状态寄存器 (SYS0_RSTST)

地址偏移量: 0x14

该寄存器包含了除 DPDWAKEUP 复位以外的所有的复位源, 这是由于 PMU_CTRL 寄存器的 MODE 位已经显示了这种复位情况。

Bit	Name	说明	Attribute	Reset
31:5	Reserved		R	0
4	PORRSTF	POR 复位标志位。发生 POR 复位时, 由 HW 设置。 0: Read→ 没有发生 POR 复位; Write→ 将该位清零; 1: 发生 POR 复位。	R/W	1
3	EXTRSTF	外部复位标志位。发生外部复位时, 由 HW 设置。 0: Read→ 没有发生外部复位; Write→ 将该位清零; 1: 发生外部复位。	R/W	0
2	LVDRSTF	LVD 复位标志位。发生 LVD 复位时, 由 HW 设置。 0: Read→ 没有发生 LVD 复位; Write→ 将该位清零; 1: 发生 LVD 复位。	R/W	0
1	WDTRSTF	WDT 复位标志位。发生 WDT 复位时, 由 HW 设置。 0: Read→ 没有发生 WDT 复位; Write→ 将该位清零; 1: 发生 WDT 复位。	R/W	0
0	SWRSTF	软件复位标志位。发生软件复位时, 由 HW 设置。 0: Read→ 没有发生软件复位; Write→ 将该位清零; 1: 发生软件复位。	R/W	1

3.3.7 LVD 控制寄存器 (SYS0_LVDCTRL)

地址偏移量: 0x18

LVD 寄存器选择 4 个独立的临界值, 用于产生一个 LVD 中断或 LVD 复位。

Bit	Name	说明	Attribute	Reset
31:16	Reserved		R	0
15	LV DEN	LVD 使能位。 0: 禁止; 1: 使能。	R/W	0
14	LVDRSTEN	LVD 复位使能位。 0: 禁止; 1: 使能。	R/W	0
13:7	Reserved		R	0
6:4	LV DINTLVL[2:0]	LVD 中断电平。 000: 1.80V; 001: 2.00V; 010: 2.40V; 011: 2.70V; 100: 3.00V; 101: 3.60V; 其它: 保留。	R/W	0
3	Reserved		R	0
2:0	LVDRSTLVL[2:0]	LVD 复位电平。 000: 1.80V; 001: 2.00V; 010: 2.40V; 011: 2.70V; 100: 3.00V; 101: 3.60V; 其它: 保留。	R/W	0

3.3.8 外部复位引脚控制寄存器 (SYS0_EXRSTCTRL)

地址偏移量: 0x1C

Bit	Name	说明	Attribute	Reset
31:1	Reserved		R	0
0	RESETDIS	外部复位引脚禁止位。 0: 使能外部复位引脚 (P3.10 作为复位引脚); 1: 禁止 (P3.10 作为 GPIO 引脚)。	R/W	1

3.3.9 SWD 引脚控制寄存器 (SYS0_SWDCtrl)

地址偏移量: 0x20

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	SWDDIS	SWD 引脚禁止位。 0: 使能 SWD 引脚 (P0.9 作为 SWDIO 引脚, P0.8 作为 SWCLK 引脚); 1: 禁止 (P0.9 和 P0.8 作为 GPIO 引脚)	R/W	0

3.3.10 噪音检测控制寄存器 (SYS0_NDTCTRL)

地址偏移量: 0x28

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	NDT2_IE	NDT2 for VDD 中断使能位。 0: 禁止; 1: 使能 (在 VDD 检测到噪音会触发 NDT 中断)。	R/W	0
0	NDT1_IE	NDT1 for VCore 中断使能位。 0: 禁止; 1: 使能 (在 VCore 检测到噪音会触发 NDT 中断)	R/W	0

3.3.11 噪音检测状态寄存器 (SYS0_NDTSTS)

地址偏移量: 0x2C

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	NDT2_DET	NDT2 的电源噪音状态。 0: 没有检测到电源噪音; 1: 读→检测到电源噪音; 写→将该位清零。	R/W	0
0	NDT1_DET	NDT1 的电源噪音状态。 0: 没有检测到电源噪音; 1: 读→检测到电源噪音; 写→将该位清零。	R/W	0

3.3.12 Anti-EFT Ability 控制寄存器 (SYS0_ANTIEFT)

地址偏移量: 0x30

该寄存器决定 HW anti-EFT 能力。

Bit	Name	说明	Attribute	Reset
31:3	Reserved		R	0
2:0	AEFT[2:0]	HW anti-EFT 能力。 000: 无; 010: 低; 011: 中; 100: 强。	R/W	000

3.4 系统控制寄存器 1

基地址: 0x4005 E000

3.4.1 AHB 时钟使能寄存器 (SYS1_AHBCLKEN)

地址偏移量: 0x00

SYS_AHBCLKEN 寄存器使能 AHB 时钟提供给独立系统和外设模块。

- * 注:
- 1.禁止时钟时, 不能通过 SW 读取外设模块的值, 该值返回为 0。
 - 2.若 CLKOUTSEL 不为 0, 则 HW 直接将 GPIO 设置为 CLKOUT 功能。

Bit	Name	说明	Attribute	Reset
31	Reserved		R	0
30:28	CLKOUTSEL[2:0]	时钟输出源。 000: 禁止; 001: ILRC 时钟; 010: ELS 时钟; 100: HCLK; 101: IHRC 时钟; 110: EHS 时钟; 111: PLL 时钟输出。	R/W	0
27:25	Reserved		R	0
24	WDTCLKEN	使能 WDT 时钟。 0: 禁止; 1: 使能。	R/W	1
23	RTCCLKEN	使能 RTC 时钟。 0: 禁止; 1: 使能。	R/W	0
22	I2SCLKEN	使能 I2S 时钟。 0: 禁止; 1: 使能。	R/W	0
21	I2C0CLKEN	使能 I2C0 时钟。 0: 禁止; 1: 使能。	R/W	0
20	I2C1CLKEN	使能 I2C1 时钟。 0: 禁止; 1: 使能。	R/W	0
19:18	Reserved		R	0
17	USART1CLKEN	使能 USART1 时钟。 0: 禁止; 1: 使能。	R/W	0
16	USART0CLKEN	使能 USART0 时钟。 0: 禁止;	R/W	0

		1: 使能。		
15:14	Reserved		R	0
13	SSP1CLKEN	使能 SSP1 时钟。 0: 禁止; 1: 使能。	R/W	0
12	SSP0CLKEN	使能 SSP0 时钟。 0: 禁止; 1: 使能。	R/W	0
11	ADCCLKEN	使能 ADC 时钟。 0: 禁止; 1: 使能。	R/W	0
10	CT32B2CLKEN	使能 CT32B2 时钟。 0: 禁止; 1: 使能。	R/W	0
9	CT32B1CLKEN	使能 CT32B1 时钟。 0: 禁止; 1: 使能。	R/W	0
8	CT32B0CLKEN	使能 CT32B0 时钟。 0: 禁止; 1: 使能。	R/W	0
7	CT16B2CLKEN	使能 CT16B2 时钟。 0: 禁止; 1: 使能。	R/W	0
6	CT16B1CLKEN	使能 CT16B1 时钟。 0: 禁止; 1: 使能。	R/W	0
5	CT16B0CLKEN	使能 CT16B0 时钟。 0: 禁止; 1: 使能。	R/W	0
4:3	Reserved		R	0
2	LCDCLKEN	使能 LCD 时钟。 0: 禁止; 1: 使能。	R/W	0
1	USBCLKEN	使能 USB 时钟。 0: 禁止; 1: 使能。	R/W	0
0	GPIOCLKEN	使能 GPIO 时钟。 0: 禁止; 1: 使能。	R/W	1

3.4.2 APB 时钟预分频寄存器 0 (SYS1_APB0)

地址偏移量: 0x04

* 注: 改变预分频值后, 必须通过 SYS1_PRST 寄存器复位相应的外设。

Bit	Name	Description	Attribute	Reset
31	Reserved		R	0
30:28	CT32B2PRE[2:0]	CT32B2 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
27	Reserved		R	0
26:24	SSP1PRE[2:0]	SSP1 时钟源预分频值。 000: HCLK / 1;	R/W	0

		001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。		
23	Reserved		R	0
22:20	SSP0PRE[2:0]	SSP0 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
19	Reserved		R	0
18:16	ADCPRE[2:0]	ADC 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
15	Reserved		R	0
14:12	CT32B1PRE[2:0]	CT32B1 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
11	Reserved		R	0
10:8	CT32B0PRE[2:0]	CT32B0 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
7	Reserved		R	0
6:4	CT16B1PRE[2:0]	CT16B1 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
3	Reserved		R	0
2:0	CT16B0PRE[2:0]	CT16B0 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0

3.4.3 APB 时钟预分频寄存器 1 (SYS1_APB1)

地址偏移量: 0x08

* 注: 改变预分频值后, 必须通过 SYS1_PRST 寄存器复位相应的外设。

Bit	Name	说明	Attribute	Reset
31	Reserved		R	0
30:28	CT16B2PRE[2:0]	CT16B2 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
27	Reserved		R	0
26:24	I2C1PRE[2:0]	I2C1 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
23	Reserved		R	0
22:20	WDTPRE[2:0]	WDT 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 101: HCLK / 32; 其它: 保留。	R/W	0
19:18	Reserved		R	0
17:16	SYSTICKPRE[1:0]	SysTick 时钟源预分频值。 00: HCLK / 1; 01: HCLK / 2; 10: HCLK / 4; 11: HCLK / 8;	R/W	0
15	Reserved		R	0
14:12	I2SPRE[2:0]	I2S 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
11	Reserved		R	0
10:8	I2C0PRE[2:0]	I2C0 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0
7	Reserved		R	0
6:4	USART1PRE[2:0]	USART1 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8;	R/W	0

		100: HCLK / 16; 其它: 保留。		
3	Reserved		R	0
2:0	USART0PRE[2:0]	USART0 时钟源预分频值。 000: HCLK / 1; 001: HCLK / 2; 010: HCLK / 4; 011: HCLK / 8; 100: HCLK / 16; 其它: 保留。	R/W	0

3.4.4 APB 时钟预分频寄存器 2 (SYS1_APBPCP2)

地址偏移量: 0x0C

Bit	Name	说明	Attribute	Reset
31:4	Reserved		R	0
3:0	CLKOUTPRE[3:0]	时钟输出源预分频值。 0000: 时钟输出源/ 1; 0001: 时钟输出源/ 2; 0010: 时钟输出源/ 4; 0011: 时钟输出源/ 8; 0100: 时钟输出源/ 16; 0101: 时钟输出源/ 32; 0110: 时钟输出源/ 64; 0111: 时钟输出源/ 128; 1000: 时钟输出源/ 256; 1001: 时钟输出源/ 512; 其它: 保留。	R/W	0

3.4.5 外设复位寄存器 (SYS1_PRST)

地址偏移量: 0x10

所有位设置为 1 后由 HW 自动清零。

Bit	Name	说明	Attribute	Reset
31:26	Reserved		R	0
25	USBRST	USB 复位。 0: 没有影响; 1: 复位 USB。	R/W	0
24	WDTRST	WDT 复位。 0: 没有影响; 1: 复位 WDT。	R/W	0
23	RTCST	RTC 复位。 0: 没有影响; 1: 复位 RTC。	R/W	0
22	I2SRST	I2S 复位。 0: 没有影响; 1: 复位 I2S。	R/W	0
21	I2C0RST	I2C0 复位。 0: 没有影响; 1: 复位 I2C0。	R/W	0
20	I2C1RST	I2C1 复位。 0: 没有影响; 1: 复位 I2C1。	R/W	0
19:18	Reserved		R	0
17	USART1RST	USART1 复位。 0: 没有影响; 1: 复位 USART1。	R/W	0
16	USART0RST	USART0 复位。	R/W	0

		0: 没有影响; 1: 复位 USART0。		
15	LCDRST	LCD 复位。 0: 没有影响; 1: 复位 LCD。	R/W	0
14	Reserved		R	0
13	SSP1RST	SSP1 复位。 0: 没有影响; 1: 复位 SSP1。	R/W	0
12	SSP0RST	SSP0 复位。 0: 没有影响; 1: 复位 SSP0。	R/W	0
11	ADCRST	ADC 复位。 0: 没有影响; 1: 复位 ADC。	R/W	0
10	CT32B2RST	CT32B2 复位。 0: 没有影响; 1: 复位 CT32B2。	R/W	0
9	CT32B1RST	CT32B1 复位。 0: 没有影响; 1: 复位 CT32B1。	R/W	0
8	CT32B0RST	CT32B0 复位。 0: 没有影响; 1: 复位 CT32B0。	R/W	0
7	CT16B2RST	CT16B2 复位。 0: 没有影响; 1: 复位 CT16B2。	R/W	0
6	CT16B1RST	CT16B1 复位。 0: 没有影响; 1: 复位 CT16B1。	R/W	0
5	CT16B0RST	CT16B0 复位。 0: 没有影响; 1: 复位 CT16B0。	R/W	0
4	Reserved		R	0
3	GPIOP3RST	GPIO P3 复位。 0: 没有影响; 1: 复位 GPIO P3。	R/W	0
2	GPIOP2RST	GPIO P2 复位。 0: 没有影响; 1: 复位 GPIO P2。	R/W	0
1	GPIOP1RST	GPIO P1 复位。 0: 没有影响; 1: 复位 GPIO P1。	R/W	0
0	GPIOP0RST	GPIO P0 复位。 0: 没有影响; 1: 复位 GPIO P0。	R/W	0

3.4.6 除法被除数寄存器 (SYS1_DIVIDEND)

地址偏移量: 0x20

Bit	Name	Description	Attribute	Reset
31:0	Dividend[31:0]	无符号的整数被除数。	R/W	0

3.4.7 除法除数寄存器 (SYS1_DIVISOR)

地址偏移量: 0x24

* 注: 除数为 0x0 时, 商是 0xFFFFFFFF, 由于 FW 可以处理这种情况, 因此不会产生 Hard Fault。

Bit	Name	Description	Attribute	Reset
31:0	Divisor[31:0]	无符号的整数除数。	R/W	0

3.4.8 除法商寄存器 (SYS1_QUOTIENT)

地址偏移量: 0x28

* 注: 除数为 0x0 时, 商是 0xFFFFFFFF, 由于 FW 可以处理这种情况, 因此不会产生 Hard Fault。

Bit	Name	Description	Attribute	Reset
31:0	Quotient[31:0]	无符号的整数商。	R/W	0

3.4.9 除法余数寄存器 (SYS1_REMAINDER)

地址偏移量: 0x2C

Bit	Name	Description	Attribute	Reset
31:0	Remainder[31:0]	无符号的整数余数。	R/W	0

3.4.10 除法控制寄存器 (SYS1_DIVCTRL)

地址偏移量: 0x30

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	DIVS	除法开始控制位。 0: 停止/完成除法操作; 1: 开始执行除法操作, 完成除法操作后由 HW 自动将 DIVS 位清零。	R/W	0

4 系统操作模式

4.1 概述

针对不同的时钟速率以及省电等原因，SN32F200 系列单片机内置 4 种不同的操作模式。这些模式控制振荡器，OP 代码的操作和模拟外设的操作。

- Normal 模式；
- 睡眠模式；
- 深度睡眠模式。

4.2 Normal 模式

Normal 模式下，ARM Cortex-M0 核，存储器，外设都由系统时钟来计时。SYS1_AHBCLKEN 寄存器控制外设的运行状况。

不同的外设都有独立的外设时钟，同时它们不光有系统时钟，还有自己的时钟分频器，而且这些外设时钟还可以单独禁止或使能。

通过模块的使能位，可以随时单独的控制各种不同的模拟模块（IHRC，EHS X'TAL，ELS X'TAL，PLL，Flash，LVD 和 ADC）。

4.3 低功耗模式

有 2 种模式来降低功耗：睡眠模式，深度睡眠模式。PMU_CTRL 寄存器控制是否进入不同的低功耗模式。

根据需要，通过改变时钟源，重新配置 PLL 值，或者改变系统时钟的分频值来改变系统时钟速率，从而可以在应用需求的 CPU 速度和系统功耗之间找到一个平衡点。

程序运行期间，功耗管理可选择关闭芯片上各独立外设的时钟，通过关闭应用中未使用外设的时钟来达到降低功耗的目的。

- * 注 1：深度睡眠模式下，不支持调试模式。
- * 注 2：低功耗模式下，将没有输出的引脚设置为合适的模式以减小功耗，强烈建议将这些引脚设置为输入上拉模式。

4.3.1 睡眠模式

睡眠模式下，ARM Cortex-M0 核的系统时钟停止工作，指令也暂停执行。

外设功能，若在 SYS1_AHBCLKEN 寄存器中使能时钟，则睡眠模式下继续工作，并且任何中断可使处理器恢复运行。睡眠模式消除了处理器，内存系统和相关的控制器以及内部总线使用的动态功耗。

模拟模块（IHRC，EHS X'TAL，ELS X'TAL，PLL，Flash，LVD 和 ADC）的电源状态由模块的使能位决定。

处理器状态和寄存器，外设寄存器和内部 SRAM 值都保持原样，引脚的逻辑电平保持静态。

中断发生时可以将系统从睡眠模式下唤醒。

睡眠模式下，RESET 引脚保持有效状态。

通过下面的步骤，可以进入睡眠模式：

1. 写入 4 到 PMU_CTRL 寄存器。
2. 执行 ARM Cortex-M0 WFI 指令。

4.3.2 深度睡眠模式

在深度睡眠模式下，ARM Cortex-M0 核的系统时钟停止工作，指令也暂停执行。

振荡器的电源状态处于掉电模式时，外设功能的时钟停止工作，时钟源也停止工作，此时只有 RTC 或 LCD 低速时钟源（ELS X'TAL 和 ILRC）还在工作。

*** 注：若使能 RTC 或 LCD，用户应该决定是否关闭低速时钟源（ELS X'TAL 和 ILRC）。**

处理器状态和寄存器，外设寄存器和内部 SRAM 值都保持原样，引脚的逻辑电平保持静态。

所有的 GPIO 引脚都有唤醒功能，用户必须通过 GPIO 寄存器设置 GPIO 引脚的边沿极性，使其可以唤醒 MCU（只有边沿触发才有唤醒功能）。当 GPIO 产生 GPIO 中断事件到 ARM core 时，系统退出深度睡眠模式，此外，必须在 NVIC 中使能中断对应的输入引脚。

深度睡眠模式下，RESET 引脚保持有效状态。

通过下面的步骤，可以进入深度睡眠模式：

1. 写入 2 到 PMU_CTRL 寄存器中。
2. 执行 ARM WFI 指令。

深度睡眠模式的优点在于，它可以使振荡器和 PLL 等时钟发生模块处于掉电状态，从而更好地降低系统的动态功耗。此外，在深度睡眠模式下可使 Flash 处于掉电状态以降低静态的漏电功耗，但这也会导致唤醒 Flash 存储器需要更长的时间。

4.4 系统唤醒

4.4.1 概述

系统在睡眠模式时并不执行程序。唤醒触发信号可以将系统唤醒进入普通模式。唤醒功能内置中断操作，系统被唤醒时触发系统执行中断服务程序。

- 睡眠模式的唤醒触发源为所有中断和 RESET 引脚。
- 深度睡眠的唤醒触发源为 GPIO 中断，RTC 中断，LCD 中断和 RESET 引脚。

4.4.2 唤醒时间

系统进入睡眠模式后，由 F/W 使能或禁止高速时钟。若高速时钟停止运行，单片机需要从睡眠模式下唤醒，则单片机需要等待 2048 个外部高速振荡器时钟周期和 32 个内部高速振荡器时钟周期，以使振荡电路进入稳定工作状态，等待的这一段就称为唤醒时间。唤醒时间结束后，系统才进入到普通模式。

* 注：若时钟没有停止运行，则系统从睡眠模式唤醒就不需要唤醒时间。

系统在深度睡眠模式下时，高速时钟停止运行。单片机从深度睡眠模式唤醒时，单片机需要等待 2048 个外部高速振荡器时钟周期和 32 个内部高速振荡器时钟周期，以使振荡电路进入稳定工作状态，等待的这一段就称为唤醒时间。唤醒时间结束后，系统才进入到普通模式。

高速时钟振荡器的唤醒时间的计算如下：

$$\boxed{\text{EHS X'tal 总的唤醒时间} = 1/F_{\text{EHS}} * 2048 \text{ (sec)} + \text{高速时钟启动时间}}$$

➤ 例：F_{EHS}=20MHz，唤醒时间计算如下：

$$\text{总的唤醒时间} = 1/F_{\text{EHS}} * 2048 + \text{振荡器启动时间} = 102.4\mu\text{s} + \text{振荡器启动时间} \quad (F_{\text{EHS}} = 20\text{MHz})$$

IHRC 的唤醒时间的计算如下：

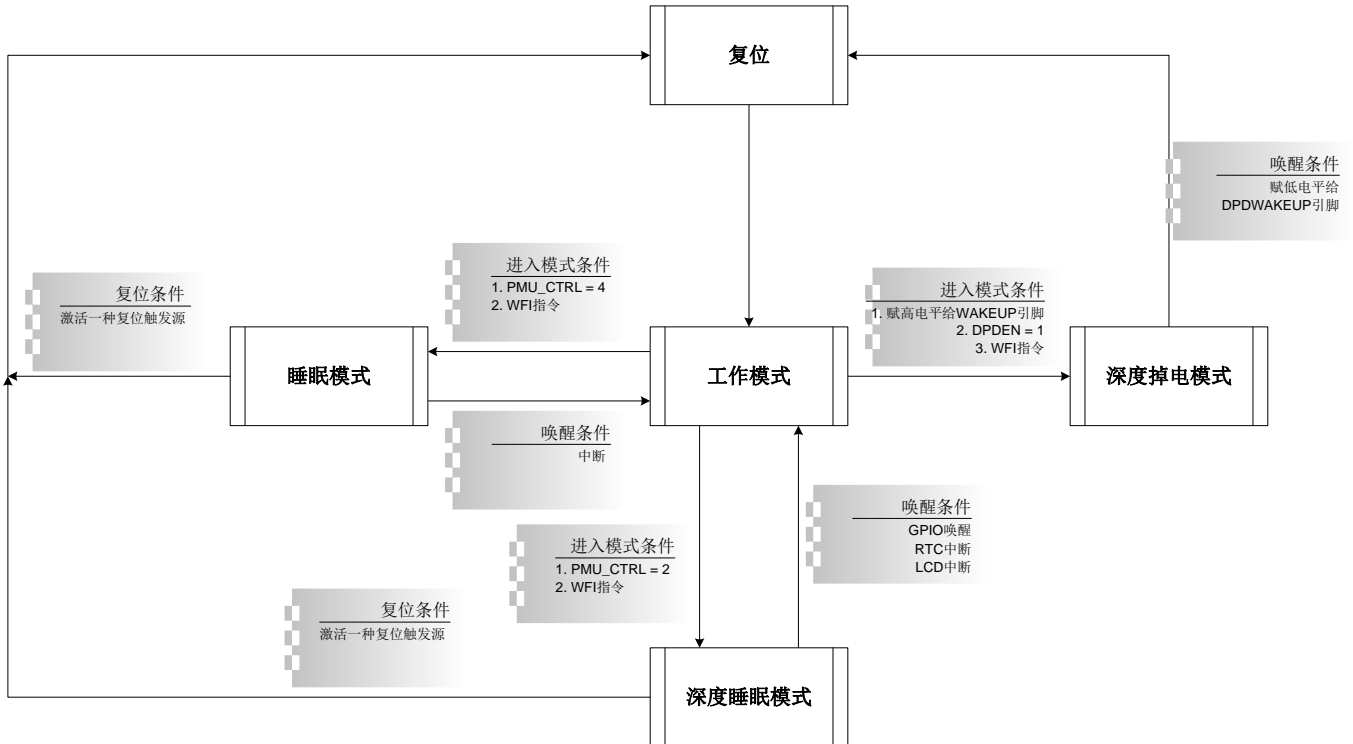
$$\boxed{\text{IHRC 总的唤醒时间} = 1/F_{\text{IHRC}} * 32 \text{ (sec)}}$$

➤ 例：F_{IHRC}=12MHz，唤醒时间计算如下：

$$\text{总的唤醒时间} = 1/F_{\text{IHRC}} * 32 = 2.67\mu\text{s} \quad (F_{\text{IHRC}} = 12\text{MHz})$$

* 注：高速时钟的启动时间与 VDD 和振荡器类型有关。

4.5 PMU 状态结构图



4.6 操作模式 Comparison 表

工作模式	普通模式	睡眠模式	深度睡眠模式
IHRC	By IHRGEN		禁止
ILRC	ON		***
EHS X'TAL	By EHSEN		禁止
ELS X'TAL	By ELSEN		***
PLL	By PLEN		禁止
Cortex-M0	运行	停止	停止
Flash ROM	使能	禁止	禁止
RAM	使能	保持	保持
ADC	By ADENB		禁止
LVD	By LVDEN		禁止
USB	By SIE_EN		禁止
LCD	By LCDENB		***
RTC	By RTCEN		By RTCEN
外设	外设的使能位		禁止 HCLK
IO 状态	-	保持	保持
唤醒源	N/A	所有中断, 复位引脚	GPIO 中断, RTC 中断, LCD 中断, 复位引脚

LCDENB	LCDCLK	RTCENB	RTC_CLKS	ILRC*	ELS*
0	---	0	---	X	X
		1	0 (ILRC) 1 (ELS)	O X	X O
1	0 (ILRC) 1 (ELS)	0	---	O X	X O
		1	0 (ILRC) 1 (ELS)	O X	X O
1	0 (ILRC) 1 (ELS)	0	0 (ILRC) 1 (ELS)	O X	X O
		1	1 (ELS) 0 (ILRC)	O O	O O

4.7 PMU 寄存器

基地址：0x4003 2000

4.7.1 电源控制寄存器 (PMU_CTRL)

地址偏移量：0x40

电源控制寄存器可选择控制 ARM Cortex-M0 进入不同的掉电模式（睡眠模式或深度睡眠模式），并且提供独立的标志位来指示睡眠模式、深度睡眠模式。

* 注：在低功耗模式下，没有配置的引脚需要设置正确来减低功耗。强烈建议将这些引脚设置为输入上拉。

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2:0	MODE[2:0]	低功耗模式选择控制位。 000：禁止； 010：WFI 指令使 MCU 进入深度睡眠模式； 100：WFI 指令使 MCU 进入睡眠模式； 其它：禁止。	R/W	0

4.7.2 I/O 锁存控制寄存器 1 (PMU_LATCHCTRL1)

地址偏移量：0x44

Bit	Name	Description	Attribute	Reset
31:16	LATCHKEY	寄存器锁存键。 读取为 0，需要写入数据到该寄存器时，必须写入 0x5A5A 到 LATCHKEY，否则忽略写入该寄存器的动作。	W	0
15:1	Reserved		R	0
0	LATCHEN	锁存使能位。 0：没有影响； 1：使能 GPIO 锁存功能。	R/W	0

4.7.3 I/O 锁存控制寄存器 2 (PMU_LATCHCTRL2)

地址偏移量：0x48

Bit	Name	Description	Attribute	Reset
31:16	LATCHKEY	寄存器锁存键。 读取为 0，需要写入数据到该寄存器时，必须写入 0x5A5A 到 LATCHKEY，否则忽略写入该寄存器的动作。	W	0
15:1	Reserved		R	0
0	LATCHDIS	锁存禁止位。 0：没有影响； 1：禁止 GPIO 锁存功能。	R/W	0

4.7.4 I/O 锁存状态寄存器 (PMU_LATCHST)

地址偏移量: 0x4C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	LATCH	锁存状态位。 0: 没有锁存; 1: 锁存 GPIO 状态。	R	0

5 GPIO 端口

5.1 概述

GPIO 端口可以通过 SW 来设置为输入或输出模式。

- 每个独立的引脚可以作为外部中断输入引脚使用。
- 可以通过单一的上升沿或者下降沿，或者双边沿触发来触发中断。
- I/O 配置寄存器控制对应 I/O pad 的电气特性。
- 内部上拉/下拉电阻寄存器。
- 大多数的 I/O 引脚都与模拟引脚或特殊功能引脚复用。

5.2 GPIO 模式

所有的 GPIO 引脚都为输入模式并默认为悬浮状态。GPIO_n_CFG (n = 0、1、2、3) 寄存器的 MODE 位配置每个引脚的上拉或下拉电阻，或者选择 repeater 模式。

若引脚为逻辑高电平时，repeater 模式使能上拉电阻；若引脚为逻辑低电平时，则使能下拉电阻。这就导致了若该引脚为输入模式且不由外部驱动时，保留最后已知状态。

5.3 GPIO 寄存器

基地址: 0x4004 4000 (GPIO 0)
0x4004 6000 (GPIO 1)
0x4004 8000 (GPIO 2)
0x4004 A000 (GPIO 3)

5.3.1 GPIO Pn 数据寄存器 (GPIO_n_DATA) (n=0,1,2,3)

地址偏移量: 0x00

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	DATA[15:0]	输入数据 (读) 或输出数据 (写)。	R/W	0

5.3.2 GPIO Pn 模式寄存器 (GPIO_n_MODE) (n=0,1,2,3)

地址偏移量: 0x04

* 注: 使能特殊功能 (外设, ADC) 时, HW 会直接切换 I/O 模式, 而不通过 GPIO_n_MODE 寄存器。

Bit	Name	Description	Attribute	Reset
31:16	CURRENT[15:0]	驱动/灌电流当前值选择位 (x=0~15)。 0: 10mA; 1: 20mA。	R/W	0
15:0	MODE[15:0]	选择引脚 x 作为输入或输出引脚 (x = 0~15) 0: 输入引脚; 1: 输出引脚。	R/W	0

5.3.3 GPIO Pn 配置寄存器 (GPIO_n_CFG) (n=0,1,3)

地址偏移量: 0x08
复位值: 0xAAAAAAAA

* 注: 使能特殊功能 (外设, ADC) 时, HW 会直接切换 I/O 模式, 而不通过 GPIO_n_MODE 寄存器。

Bit	Name	Description	Attribute	Reset
31:30	CFG15[1:0]	Pn.15 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
29:28	CFG14[1:0]	Pn.14 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
27:26	CFG13[1:0]	Pn.13 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
25:24	CFG12[1:0]	Pn.12 的配置位。 00: 使能上拉电阻;	R/W	10b

		01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。		
23:22	CFG11[1:0]	Pn.11 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
21:20	CFG10[1:0]	Pn.10 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
19:18	CFG9[1:0]	Pn.9 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
17:16	CFG8[1:0]	Pn.8 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
15:14	CFG7[1:0]	Pn.7 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
13:12	CFG6[1:0]	Pn.6 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
11:10	CFG5[1:0]	Pn.5 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
9:8	CFG4[1:0]	Pn.4 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
7:6	CFG3[1:0]	Pn.3 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
5:4	CFG2[1:0]	Pn.2 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
3:2	CFG1[1:0]	Pn.1 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b
1:0	CFG0[1:0]	Pn.0 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻); 11: Repeater 模式。	R/W	10b

5.3.4 GPIO Pn 配置寄存器 (GPIO_n_CFG) (n=2)

地址偏移量: 0x08

复位值: 0xAAAAAAAA

* 注: 使能特殊功能(外设, ADC)时, HW 会直接切换 I/O 模式, 而不通过 GPIO_n_MODE 寄存器。

Bit	Name	Description	Attribute	Reset
31:30	CFG15[1:0]	Pn.15 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
29:28	CFG14[1:0]	Pn.14 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
27:26	CFG13[1:0]	Pn.13 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
25:24	CFG13[1:0]	Pn.13 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
23:22	CFG11[1:0]	Pn.11 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
21:20	CFG10[1:0]	Pn.10 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
19:18	CFG9[1:0]	Pn.9 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
17:16	CFG8[1:0]	Pn.8 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
15:14	CFG7[1:0]	Pn.7 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
13:12	CFG6[1:0]	Pn.6 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式(禁止上拉/下拉电阻, 使能施密特); 11: 无效模式(禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
11:10	CFG5[1:0]	Pn.5 的配置位。	R/W	10b

		00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻, 使能施密特); 11: 无效模式 (禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);		
9:8	CFG4[1:0]	Pn.4 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻, 使能施密特); 11: 无效模式 (禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
7:6	CFG3[1:0]	Pn.3 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻, 使能施密特); 11: 无效模式 (禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
5:4	CFG2[1:0]	Pn.2 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻, 使能施密特); 11: 无效模式 (禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
3:2	CFG1[1:0]	Pn.1 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻, 使能施密特); 11: 无效模式 (禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b
1:0	CFG0[1:0]	Pn.0 的配置位。 00: 使能上拉电阻; 01: 使能下拉电阻; 10: 无效模式 (禁止上拉/下拉电阻, 使能施密特); 11: 无效模式 (禁止上拉/下拉电阻, 禁止施密特, 数据寄存器保持低);	R/W	10b

5.3.5 GPIO Pn 中断 SENSE 寄存器 (GPIO_n_IS) (n=0,1,2,3)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IS[15:0]	选择 Pn.x 是由电平触发还是边沿触发 (x = 0~15)。 0: 边沿触发; 1: 电平触发。	R/W	0

5.3.6 GPIO Pn 中断双边沿 SENSE 寄存器 (GPIO_n_IBS) (n=0,1,2,3)

地址偏移量: 0x10

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IBS[15:0]	选择是否由双边沿触发 Pn.x 的中断 (x = 0~15)。 0: 由寄存器 GPIO _n _IEV 控制 Pn.x 的中断; 1: 由双边沿触发 Pn.x 的中断。	R/W	0

5.3.7 GPIO Pn 中断事件寄存器 (GPIO_n_IEV) (n=0,1,2,3)

地址偏移量: 0x14

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IEV[15:0]	选择 Pn.x 是在上升沿还是下降沿触发中断 (x = 0~15)。 0: 取决于寄存器 GPIO _n _IS 的设置, 上升沿或者 Pn.x 为高电平触发中断; 1: 取决于寄存器 GPIO _n _IS 的设置, 下降沿或者 Pn.x 为低电平触发中断。	R/W	0

5.3.8 GPIO Pn 中断使能寄存器 (GPIO_n_IE) (n=0,1,2,3)

地址偏移量: 0x18

将 GPIO_n_IE 寄存器的位置 1 时, 允许相对应的引脚触发各自的中断; 反之将寄存器的位清零则禁止各自的中断。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IE[15:0]	Pn.x 中断使能位 (x = 0~15)。 0: 禁止; 1: 使能。	R/W	0

5.3.9 GPIO Pn 中断源状态寄存器 (GPIO_n_RIS) (n=0,1,2,3)

地址偏移量: 0x1C

该寄存器显示 GPIO 控制中断源的状态, 若 GPIO_n_IE 寄存器的位置 1, 则发送 GPIO 中断到中断控制器中。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IF[15:0]	GPIO 中断源标志位 (x = 0~15)。 0: 无中断; 1: 有中断发生。	R	0

5.3.10 GPIO Pn 中断清零寄存器 (GPIO_n_IC) (n=0,1,2,3)

地址偏移量: 0x20

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IC[15:0]	Pn.x 中断标志清零使能位 (x = 0~5)。 0: 没有影响; 1: 清除 Pn.x 中断标志位。	W	0

5.3.11 GPIO Pn 位设置操作寄存器 (GPIO_n_BSET) (n=0,1,2,3)

地址偏移量: 0x24

为了让 SW 设置 GPIO 位而不影响到其它引脚的写操作, 若该寄存器的相应位置 1, 则 GPIO 位也置 1。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	BSET[15:0]	位设置使能位 (x = 0~15)。 0: 不影响 Pn.x; 1: Pn.x 位置 1。	W	0

5.3.12 GPIO Pn 位清零操作寄存器 (GPIO_n_BCLR) (n=0,1,2,3)

地址偏移量: 0x28

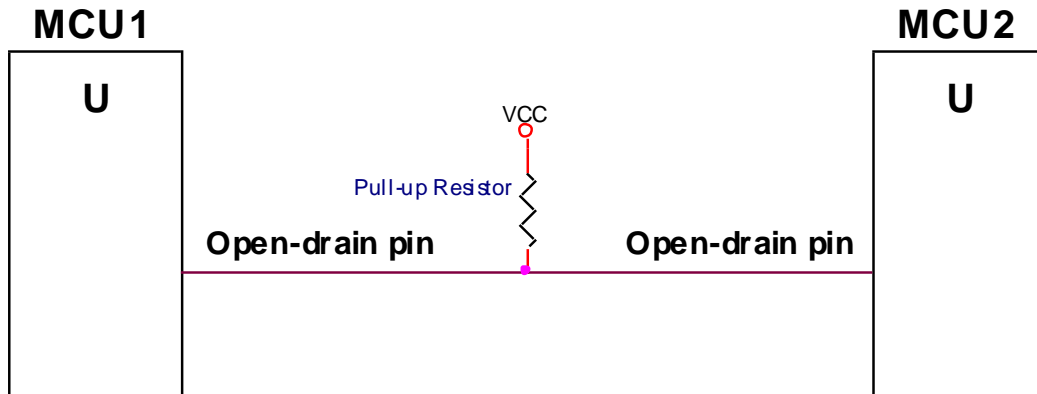
为了让 SW 将 GPIO 位清零而不影响到其它引脚的写操作, 若该寄存器的相应位置 1, 则 GPIO 位被清零。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	BCLR[15:0]	位清零使能位 (x = 0~15)。 0: 不影响 Pn.x; 1: Pn.x 位清零。	W	0

5.3.13 GPIO Pn 开漏控制寄存器 (GPIO_n_ODCTRL) (n=0,1,2,3)

地址偏移量: 0x2C

部分 I/O 引脚都有内置开漏功能，使能开漏功能时必须将引脚设置为输出模式。开漏外部电路如下：



外部上拉电阻是必不可少的。I/O 引脚的数字输出功能只支持灌电流，故通过上拉电阻驱动开漏输出高电平，通过灌电流输出低电平。

- * 注：VCC 必须要小于或者等于 MCU1 和 MCU2 的 VDD。
- * 注：只有 P0, P1 和 P3 支持开漏功能。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	OC[15:0]	开漏功能控制位 (x = 0~15)。 0: 禁止; 1: 使能 Pn.x 的开漏功能, HW 自动设置 Pn.x 为输出模式。	R/W	0

6 外设功能引脚配置 (PFPA)

6.1 概述

PFPA 寄存器用于在不同的封装中配置 GPIO 作为不同的外设功能

6.2 特性

配置 GPIO 作为希望的外设功能
支持 USART, I2C, SSP, I2S, Capture 和 PWM 功能。

6.3 引脚配置表

Peripheral	Pin Name	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA8	PA9
UART0	URXD0	P0.0	P0.4	P1.3	P3.0	P3.3	P3.5	P3.11	P3.14		
	UTXD0	P0.1	P0.5	P1.2	P3.1	P3.2	P3.4	P3.10	P3.15		
UART1	URXD1	P1.0	P0.7	P0.12	P1.5	P1.13	P1.15	P3.6	P3.12		
	UTXD1	P1.1	P0.6	P0.13	P1.4	P1.14	P3.8	P3.10	P3.13		
SSP0	SCK0	P0.4	P0.11	P0.13	P1.4	P1.12	P3.2	P3.6	P3.11	P2.3	P2.14
	SEL0	P0.5	P0.10	P0.12	P1.5	P1.11	P3.1	P3.5	P3.10	P2.0	P2.15
	MISO0	P0.2	P0.0	P0.6	P0.15	P1.0	P1.14	P3.3	P3.15	P2.1	P2.12
	MOSI0	P0.3	P0.1	P0.7	P0.14	P1.1	P1.15	P3.4	P3.14	P2.2	P2.13
SSP1	SCK1	P3.7	P0.7	P0.14	P1.1	P1.11	P1.15	P3.3	P3.14	P2.2	P2.13
	SEL1	P3.6	P0.2	P0.13	P1.0	P1.4	P1.7	P1.14	P3.11	P2.1	P2.14
	MISO1	P3.9	P0.4	P0.10	P1.3	P1.10	P3.0	P3.4	P3.12	P2.1	P2.12
	MOSI1	P3.8	P0.5	P0.12	P1.2	P1.6	P1.13	P3.2	P3.13	P2.0	P2.15
I2C0	SCL0	P1.5	P0.2	P0.15	P1.3	P1.14	P3.9	P3.11	P3.14		
	SDA0	P1.4	P0.3	P0.10	P1.2	P1.13	P3.7	P3.13	P3.15		
I2C1	SCL1	P0.6	P0.0	P1.1	P1.9	P3.0	P3.3	P3.6	P3.13		
	SDA1	P0.7	P0.1	P1.0	P1.8	P3.2	P3.4	P3.5	P3.12		
I2S	MCLK	P3.2	P0.12	P1.8	P3.7	P2.2	P2.6				
	BCLK	P3.3	P0.13	P1.9	P3.6	P2.10	P2.11				
	WS	P3.4	P0.14	P1.10	P2.1	P2.7	P2.9				
	DOUT	P3.1	P0.11	P1.7	P3.8	P2.8	P2.12				
	DIN	P3.0	P0.10	P1.6	P3.9	P2.0	P2.5				
CT16B0	CAPO	P0.2	P0.8	P1.0	P3.0	P3.2	P3.10	P2.0	P2.13		
	PWM0	P0.0	P1.1	P1.8	P1.12	P3.3	P3.11	P2.3	P2.15		
	PWM1	P0.1	P0.4	P0.10	P1.13	P3.4	P3.12	P2.2	P2.11		
	PWM2	P1.12	P0.9	P0.11	P1.6	P3.6	P3.15	P2.4	P2.10		
CT16B1	CAPO	P0.12	P0.7	P1.7	P1.11	P3.5	P3.13	P2.1	P2.9		
	PWM0	P0.10	P0.5	P1.9	P1.15	P3.7	P3.14	P2.0	P2.12		
	PWM1	P0.11	P0.8	P0.12	P1.3	P1.10	P3.9	P2.4	P2.8		
CT16B2	PWM2	P1.9	P0.6	P0.15	P1.2	P1.14	P3.8	P2.3	P2.7		
	CAPO	P1.8	P0.3	P0.13	P1.5	P3.6	P3.12	P2.2	P2.14		
	PWM0	P3.5	P0.2	P0.14	P1.4	P3.1	P3.10	P2.5	P2.9		
CT32B0	PWM1	P1.4	P0.0	P0.9	P1.11	P3.5	P3.15	P2.1	P2.6		
	PWM2	P3.1	P0.1	P1.0	P1.12	P3.0	P3.11	P2.4	P2.10		
	CAPO	P3.9	P0.0	P0.9	P1.1	P3.1	P3.3	P2.3	P2.15		
	PWM0	P1.15	P0.4	P0.11	P1.8	P1.14	P3.9	P2.3	P2.14		
CT32B1	PWM1	P3.8	P0.8	P0.14	P1.5	P1.11	P3.14	P2.4	P2.13		
	PWM2	P1.14	P0.5	P0.9	P1.9	P3.7	P3.13	P2.0	P2.11		
	PWM3	P1.2	P0.3	P0.7	P0.15	P1.12	P3.4	P2.2	P2.12		
	CAPO	P1.3	P0.2	P0.8	P0.15	P1.6	P3.8	P2.4	P2.10		
CT32B2	PWM0	P1.13	P0.1	P0.9	P1.3	P3.2	P3.15	P2.6	P2.15		
	PWM1	P0.15	P0.4	P1.2	P1.10	P3.5	P3.10	P2.1	P2.8		
	PWM2	P1.6	P0.5	P0.8	P0.11	P3.12	P2.5	P2.12	P2.14		
	PWM3	P1.7	P0.6	P0.7	P1.2	P3.9	P3.11	P2.4	P2.13		
	CAPO	P3.7	P0.9	P0.10	P1.4	P1.13	P3.14	P2.5	P2.12		

	PWM0	P0.13	P0.3	P0.8	P1.1	P1.7	P1.15	P2.11	P2.15		
	PWM1	P0.14	P0.2	P0.15	P1.5	P3.0	P3.7	P2.7	P2.13		
	PWM2	P0.3	P0.0	P0.9	P1.6	P3.1	P3.13	P3.15	P2.14		
	PWM3	P0.6	P1.0	P1.7	P3.10	P2.3	P2.8	P2.9	P2.15		

6.4 PFPA 寄存器

基地址: 0x4004 2000

6.4.1 PFPA UART 寄存器 (PFPA_UART)

地址偏移量: 0x00

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:12	URXD1[3:0]	下列引脚配置为 URXD1。 0000: P1.0; 0001: P0.7; 0010: P0.12; 0011: P1.5; 0100: P1.13; 0101: P1.15; 0110: P3.6; 0111: P3.12; 其它: 保留。	R/W	0000b
11:8	UTXD1[3:0]	下列引脚配置为 UTXD1。 0000: P1.1; 0001: P0.6; 0010: P0.13; 0011: P1.4; 0100: P1.14; 0101: P3.8; 0110: P3.10; 0111: P3.13; 其它: 保留。	R/W	0000b
7:4	URXD0[3:0]	下列引脚配置为 URXD0。 0000: P0.0; 0001: P0.4; 0010: P1.3; 0011: P3.0; 0100: P3.3; 0101: P3.5; 0110: P3.11; 0111: P3.14; 其它: 保留。	R/W	0000b
3:0	UTXD0[3:0]	下列引脚配置为 UTXD0。 0000: P0.1; 0001: P0.5; 0010: P1.2; 0011: P3.1; 0100: P3.2; 0101: P3.4; 0110: P3.10; 0111: P3.15; 其它: 保留。	R/W	0000b

6.4.2 PFPA I2C 寄存器 (PFPA_I2C)

地址偏移量: 0x04

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:12	SCL1[3:0]	下列引脚配置为 SCL1。 0000: P0.6; 0001: P0.0; 0010: P1.1; 0011: P1.9; 0100: P3.0; 0101: P3.3; 0110: P3.6; 0111: P3.13; 其它: 保留。	R/W	0000b
11:8	SDA1[3:0]	下列引脚配置为 SDA1。 0000: P0.7; 0001: P0.1; 0010: P1.0; 0011: P1.8; 0100: P3.2; 0101: P3.4; 0110: P3.5; 0111: P3.12; 其它: 保留。	R/W	0000b
7:4	SCL0[3:0]	下列引脚配置为 SCL0。 0000: P1.5; 0001: P0.2; 0010: P0.15; 0011: P1.3; 0100: P1.14; 0101: P3.9; 0110: P3.11; 0111: P3.14; 其它: 保留。	R/W	0000b
3:0	SDA0[3:0]	下列引脚配置为 SDA0。 0000: P1.4; 0001: P0.3; 0010: P0.10; 0011: P1.2; 0100: P1.13; 0101: P3.7; 0110: P3.13; 0111: P3.15; 其它: 保留。	R/W	0000b

6.4.3 PFPA SSP 寄存器 (PFPA_SSP)

地址偏移量: 0x08

Bit	Name	Description	Attribute	Reset
31:28	SEL1[3:0]	下列引脚配置为 SEL1。 0000: P3.6; 0001: P0.2; 0010: P0.13; 0011: P1.0; 0100: P1.4; 0101: P1.7; 0110: P1.14; 0111: P3.11; 1000: P2.1; 1001: P2.14; 其它: 保留。	R/W	0000b
27:24	SCK1[3:0]	下列引脚配置为 SCK1。 0000: P3.7; 0001: P0.7; 0010: P0.14; 0011: P1.1; 0100: P1.11; 0101: P1.15; 0110: P3.3; 0111: P3.14; 1000: P2.2; 1001: P2.13; 其它: 保留。	R/W	0000b
23:20	MOSI1[3:0]	下列引脚配置为 MOSI1。 0000: P3.8; 0001: P0.5; 0010: P0.12; 0011: P1.2; 0100: P1.6; 0101: P1.13; 0110: P3.2; 0111: P3.13; 1000: P2.0; 1001: P2.15; 其它: 保留。	R/W	0000b
19:16	MISO1[3:0]	下列引脚配置为 MISO1。 0000: P3.9; 0001: P0.4; 0010: P0.10; 0011: P1.3; 0100: P1.10; 0101: P3.0; 0110: P3.4; 0111: P3.12; 1000: P2.1; 1001: P2.12; 其它: 保留。	R/W	0000b
15:12	SEL0[3:0]	下列引脚配置为 SEL0。 0000: P0.5; 0001: P0.10; 0010: P0.12; 0011: P1.5; 0100: P1.11; 0101: P3.1; 0110: P3.5; 0111: P3.10; 1000: P2.0;	R/W	0000b

		1001: P2.15; 其它: 保留。		
11:8	SCK0[3:0]	下列引脚配置为 SCK0。 0000: P0.4; 0001: P0.11; 0010: P0.13; 0011: P1.4; 0100: P1.12; 0101: P3.2; 0110: P3.6; 0111: P3.11; 1000: P2.3; 1001: P2.14; 其它: 保留。	R/W	0000b
7:4	MOSI0[3:0]	下列引脚配置为 MOSI0。 0000: P0.3; 0001: P0.1; 0010: P0.7; 0011: P0.14; 0100: P1.1; 0101: P1.15; 0110: P3.4; 0111: P3.14; 1000: P2.2; 1001: P2.13; 其它: 保留。	R/W	0000b
3:0	MISO0[3:0]	下列引脚配置为 MISO0。 0000: P0.2; 0001: P0.0; 0010: P0.6; 0011: P0.15; 0100: P1.0; 0101: P1.14; 0110: P3.3; 0111: P3.15; 1000: P2.1; 1001: P2.12; 其它: 保留。	R/W	0000b

6.4.4 PFPA I2S 寄存器 (PFPA_I2S)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:20	Reserved		R	0
19:16	DIN[3:0]	下列引脚配置为 I2SDIN。 0000: P3.0; 0001: P0.10; 0010: P1.6; 0011: P3.9; 0100: P2.0; 0101: P2.5; 其它: 保留。	R/W	0000b
15:12	DOUT[3:0]	下列引脚配置为 I2SDOUT。 0000: P3.1; 0001: P0.11; 0010: P1.7; 0011: P3.8; 0100: P2.8; 0101: P2.12; 其它: 保留。	R/W	0000b
11:8	WS[3:0]	下列引脚配置为 I2SWS。 0000: P3.4; 0001: P0.14; 0010: P1.10; 0011: P2.1; 0100: P2.7; 0101: P2.9; 其它: 保留。	R/W	0000b
7:4	BCLK[3:0]	下列引脚配置为 I2SBCLK。 0000: P3.3; 0001: P0.13; 0010: P1.9; 0011: P3.6; 0100: P2.10; 0101: P2.11; 其它: 保留。	R/W	0000b
3:0	MCLK[3:0]	下列引脚配置为 I2SMCLK。 0000: P3.2; 0001: P0.12; 0010: P1.8; 0011: P3.7; 0100: P2.2; 0101: P2.6; 其它: 保留。	R/W	0000b

6.4.5 PFPA CT16B0 寄存器 (PFPA_CT16B0)

地址偏移量: 0x10

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:12	PWM2[3:0]	下列引脚配置为 CT16B0_PWM2。 0000: P1.12; 0001: P0.9; 0010: P0.11; 0011: P1.6; 0100: P3.6; 0101: P3.15; 0110: P2.4; 0111: P2.10; 其它: 保留。	R/W	0000b
11:8	PWM1[3:0]	下列引脚配置为 CT16B0_PWM1。 0000: P0.1; 0001: P0.4; 0010: P0.10; 0011: P1.13; 0100: P3.4; 0101: P3.12; 0110: P2.2; 0111: P2.11; 其它: 保留。	R/W	0000b
7:4	PWM0[3:0]	下列引脚配置为 CT16B0_PWM0。 0000: P0.0; 0001: P1.1; 0010: P1.8; 0011: P1.12; 0100: P3.3; 0101: P3.11; 0110: P2.3; 0111: P2.15; 其它: 保留。	R/W	0000b
3:0	CAP0[3:0]	下列引脚配置为 CT16B0_CAP0。 0000: P0.2; 0001: P0.8; 0010: P1.0; 0011: P3.0; 0100: P3.2; 0101: P3.10; 0110: P2.0; 0111: P2.13; 其它: 保留。	R/W	0000b

6.4.6 PFPA CT16B1 寄存器 (PFPA_CT16B1)

地址偏移量: 0x14

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:12	PWM2[3:0]	下列引脚配置为 CT16B1_PWM2。 0000: P1.9; 0001: P0.6; 0010: P0.15; 0011: P1.2; 0100: P1.14; 0101: P3.8; 0110: P2.3; 0111: P2.7; 其它: 保留。	R/W	0000b
11:8	PWM1[3:0]	下列引脚配置为 CT16B1_PWM1。 0000: P0.11; 0001: P0.8; 0010: P0.12; 0011: P1.3; 0100: P1.10; 0101: P3.9; 0110: P2.4; 0111: P2.8; 其它: 保留。	R/W	0000b
7:4	PWM0[3:0]	下列引脚配置为 CT16B1_PWM0。 0000: P0.10; 0001: P0.5; 0010: P1.9; 0011: P1.15; 0100: P3.7; 0101: P3.14; 0110: P2.0; 0111: P2.12; 其它: 保留。	R/W	0000b
3:0	CAP0[3:0]	下列引脚配置为 CT16B1_CAP0。 0000: P0.12; 0001: P0.7; 0010: P1.7; 0011: P1.11; 0100: P3.5; 0101: P3.13; 0110: P2.1; 0111: P2.9; 其它: 保留。	R/W	0000b

6.4.7 PFPA CT16B2 寄存器 (PFPA_CT16B2)

地址偏移量: 0x18

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:12	PWM2[3:0]	下列引脚配置为 CT16B2_PWM2。 0000: P3.1; 0001: P0.1; 0010: P1.0; 0011: P1.12; 0100: P3.0; 0101: P3.11; 0110: P2.4; 0111: P2.10; 其它: 保留。	R/W	0000b
11:8	PWM1[3:0]	下列引脚配置为 CT16B2_PWM1。 0000: P1.4; 0001: P0.0; 0010: P0.9; 0011: P1.11; 0100: P3.5; 0101: P3.15; 0110: P2.1; 0111: P2.6; 其它: 保留。	R/W	0000b
7:4	PWM0[3:0]	下列引脚配置为 CT16B2_PWM0。 0000: P3.5; 0001: P0.2; 0010: P0.14; 0011: P1.4; 0100: P3.1; 0101: P3.10; 0110: P2.5; 0111: P2.9; 其它: 保留。	R/W	0000b
3:0	CAP0[3:0]	下列引脚配置为 CT16B2_CAP0。 0000: P1.8; 0001: P0.3; 0010: P0.13; 0011: P1.5; 0100: P3.6; 0101: P3.12; 0110: P2.2; 0111: P2.14; 其它: 保留。	R/W	0000b

6.4.8 PFPA CT32B0 寄存器 (PFPA_CT32B0)

地址偏移量: 0x20

Bit	Name	Description	Attribute	Reset
31:20	Reserved		R	0
19:16	PWM3[3:0]	下列引脚配置为 CT32B0_PWM3。 0000: P1.2; 0001: P0.3; 0010: P0.7; 0011: P0.15; 0100: P1.12; 0101: P3.4; 0110: P2.2; 0111: P2.12; 其它: 保留。	R/W	0000b
15:12	PWM2[3:0]	下列引脚配置为 CT32B0_PWM2。 0000: P1.14; 0001: P0.5; 0010: P0.9; 0011: P1.9; 0100: P3.7; 0101: P3.13; 0110: P2.0; 0111: P2.11; 其它: 保留。	R/W	0000b
11:8	PWM1[3:0]	下列引脚配置为 CT32B0_PWM1。 0000: P3.8; 0001: P0.8; 0010: P0.14; 0011: P1.5; 0100: P1.11; 0101: P3.14; 0110: P2.4; 0111: P2.13; 其它: 保留。	R/W	0000b
7:4	PWM0[3:0]	下列引脚配置为 CT32B0_PWM0。 0000: P1.15; 0001: P0.4; 0010: P0.11; 0011: P1.8; 0100: P1.14; 0101: P3.9; 0110: P2.3; 0111: P2.14; 其它: 保留。	R/W	0000b
3:0	CAP0[3:0]	下列引脚配置为 CT32B0_CAP0。 0000: P3.9; 0001: P0.0; 0010: P0.9; 0011: P1.1; 0100: P3.1; 0101: P3.3; 0110: P2.3; 0111: P2.15; 其它: 保留。	R/W	0000b

6.4.9 PFPA CT32B1 寄存器 (PFPA_CT32B1)

地址偏移量: 0x24

Bit	Name	Description	Attribute	Reset
31:20	Reserved		R	0
19:16	PWM3[3:0]	下列引脚配置为 CT32B1_PWM3。 0000: P1.7; 0001: P0.6; 0010: P0.7; 0011: P1.2; 0100: P3.9; 0101: P3.11; 0110: P2.4; 0111: P2.13; 其它: 保留。	R/W	0000b
15:12	PWM2[3:0]	下列引脚配置为 CT32B1_PWM2。 0000: P1.6; 0001: P0.5; 0010: P0.8; 0011: P0.11; 0100: P3.12; 0101: P2.5; 0110: P2.12; 0111: P2.14; 其它: 保留。	R/W	0000b
11:8	PWM1[3:0]	下列引脚配置为 CT32B1_PWM1。 0000: P0.15; 0001: P0.4; 0010: P1.2; 0011: P1.10; 0100: P3.5; 0101: P3.10; 0110: P2.1; 0111: P2.8; 其它: 保留。	R/W	0000b
7:4	PWM0[3:0]	下列引脚配置为 CT32B1_PWM0。 0000: P1.13; 0001: P0.1; 0010: P0.9; 0011: P1.3; 0100: P3.2; 0101: P3.15; 0110: P2.6; 0111: P2.15; 其它: 保留。	R/W	0000b
3:0	CAP0[3:0]	下列引脚配置为 CT32B1_CAP0。 0000: P1.3; 0001: P0.2; 0010: P0.8; 0011: P0.15; 0100: P1.6; 0101: P3.8; 0110: P2.4; 0111: P2.10; 其它: 保留。	R/W	0000b

6.4.10 PFPA CT32B2 寄存器 (PFPA_CT32B2)

地址偏移量: 0x28

Bit	Name	Description	Attribute	Reset
31:20	Reserved		R	0
19:16	PWM3[3:0]	下列引脚配置为 CT32B2_PWM3。 0000: P0.6; 0001: P1.0; 0010: P1.7; 0011: P3.10; 0100: P2.3; 0101: P2.8; 0110: P2.9; 0111: P2.15; 其它: 保留。	R/W	0000b
15:12	PWM2[3:0]	下列引脚配置为 CT32B2_PWM2。 0000: P0.3; 0001: P0.0; 0010: P0.9; 0011: P1.6; 0100: P3.1; 0101: P3.13; 0110: P3.15; 0111: P2.14; 其它: 保留。	R/W	0000b
11:8	PWM1[3:0]	下列引脚配置为 CT32B2_PWM1。 0000: P0.14; 0001: P0.2; 0010: P0.15; 0011: P1.5; 0100: P3.0; 0101: P3.7; 0110: P2.7; 0111: P2.13; 其它: 保留。	R/W	0000b
7:4	PWM0[3:0]	下列引脚配置为 CT32B2_PWM0。 0000: P0.13; 0001: P0.3; 0010: P0.8; 0011: P1.1; 0100: P1.7; 0101: P1.15; 0110: P2.11; 0111: P2.15; 其它: 保留。	R/W	0000b
3:0	CAP0[3:0]	下列引脚配置为 CT32B2_CAP0。 0000: P3.7; 0001: P0.9; 0010: P0.10; 0011: P1.4; 0100: P1.13; 0101: P3.14; 0110: P2.5; 0111: P2.12; 其它: 保留。	R/W	0000b

7 14+1 通道 ADC

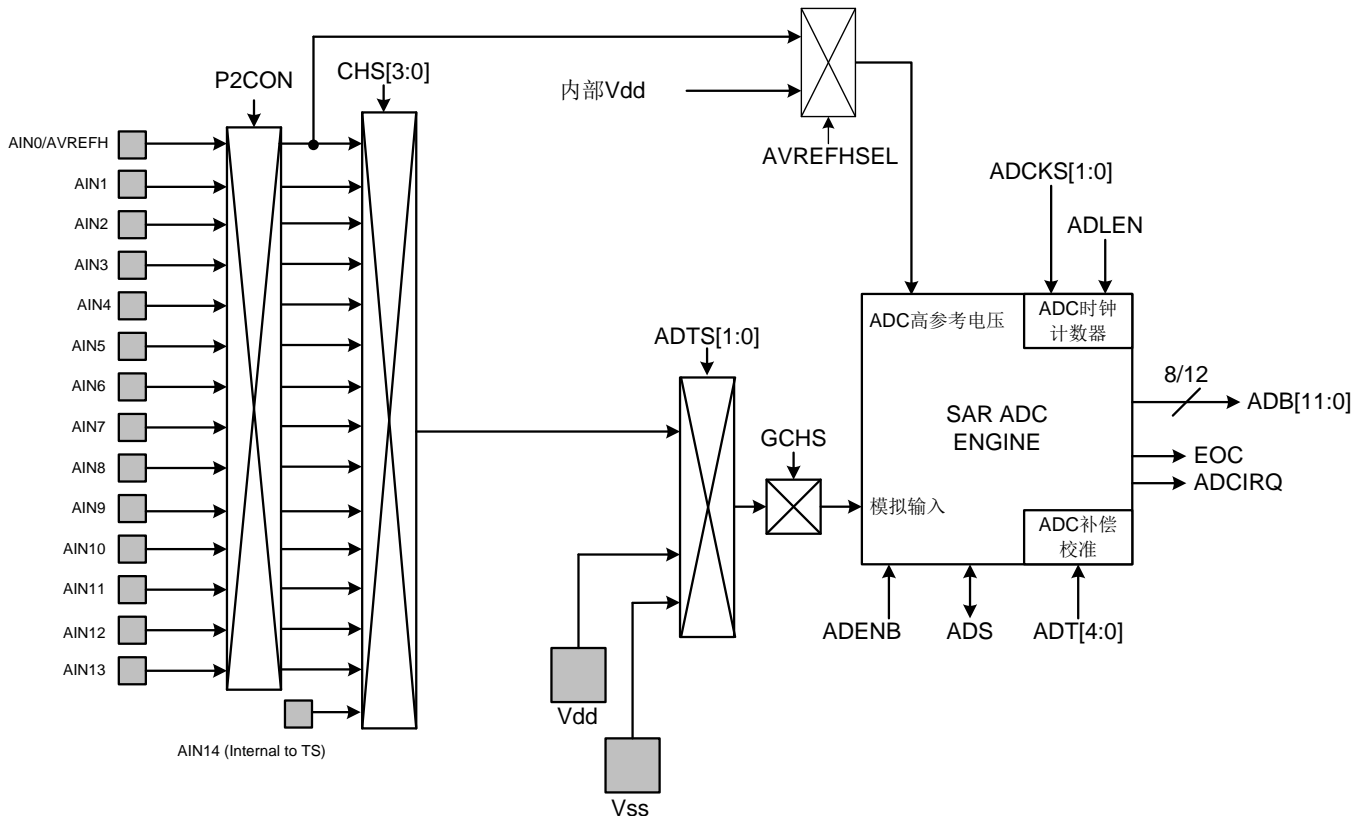
7.1 概述

SN32F200 系列单片机的 AD 转换共有 14 个外部输入通道和 1 个内部通道用于温度传感器，高达 4096 阶的分辨率，将模拟信号转换成 12 位数字信号。ADC 操作流程为：首先选择输入源（AIN0~AIN13），然后设置 GCHS 和 ADS 位为 1，开始 AD 转换。转换完成后，ADC 电路将 EOC 位置 1，最终值输出到 ADB 寄存器。

利用 CHS[3:0]选择 AIN 引脚，GCHS 使能全局 ADC 通道，模拟信号输入到 ADC Engine。

ADC 参考高压源有两种：内部 VDD（AVREFHSEL = 0）和由 P2.0 输入的外部参考电压（AVREFHSEL = 1）。

ADC 的分辨率可通过 ADR 寄存器的 ADLEN 位选择 8 位或者 12 位，ADC 的转换率可通过 ADCKS[1:0]位来选择，这两个参数决定了 ADC 的转换时间。



- * 注：分辨率为 8 位时，转换时间为 12 steps；分辨率为 12 位时，转换时间为 16 steps。
- * 注：ADC_PCLK 应小于 16MHz。
- * 注：模拟输入引脚的输入电压值必须在 AVREFH 和 AVREFL 之间。
- * 注：AVREFH 的值必须在 AVDD 和 AVREFL+2.0V 之间。
- * 注：ADC 应用时须注意以下几点：
 1. 进入省电模式（睡眠/深度睡眠模式）前必须禁止 ADC（ADENB = 0）以省电。
 2. 使能 ADC（ADENB =1）后延时 100us 等待 ADC 电路稳定。

7.2 AD 转换时间

AD 转换时间从 ADS=1（开始 AD 转换）到 EOC=1（AD 转换结束），转换时间取决于 ADC 的分辨率和 ADC 的时钟频率。

ADCKS[2:0]控制 ADC 的时钟源，AD 转换时间会影响 ADC 的性能。若输入高速率的模拟信号，则必须选择高速率的 ADC。如果 AD 转换时间慢于模拟信号的转换速率，ADC 的结果则会出错。故选择一个正确的 ADC 时钟频率和 ADC 分辨率决定一个正确的 ADC 转换速度是非常重要的。

$$\boxed{12 \text{ 位 ADC 转换时间} = 1/(\text{ADC clock} / 4) * 16 \text{ sec}}$$

ADLEN	ADCKS [2:0]	ADC 时钟	ADC_PCLK = 4 MHz		ADC_PCLK = 16 MHz	
			AD 转换时间(us)	AD 转换速率(KHz)	AD 转换时间(us)	AD 转换速率(KHz)
1	000	ADC_PCLK	16	62.5	4	250
	001	ADC_PCLK/2	32	31.25	8	125
	010	ADC_PCLK/4	64	15.625	16	62.5
	011	ADC_PCLK/8	128	7.813	32	31.25
	100	ADC_PCLK/16	256	3.906	64	15.625
	101	ADC_PCLK/32	512	1.953	128	7.813

$$\boxed{8 \text{ 位 ADC 转换时间} = 1/(\text{ADC clock} / 4) * 12 \text{ sec}}$$

ADLEN	ADCKS [2:0]	ADC 时钟	ADC_PCLK = 4 MHz		ADC_PCLK = 16 MHz	
			AD 转换时间(us)	AD 转换速率(KHz)	AD 转换时间(us)	AD 转换速率(KHz)
0	000	ADC_PCLK	12	83.333	3	333.333
	001	ADC_PCLK/2	24	41.667	6	166.667
	010	ADC_PCLK/4	48	20.83	12	83.333
	011	ADC_PCLK/8	96	10.416	24	41.667
	100	ADC_PCLK/16	192	5.208	48	20.83
	101	ADC_PCLK/32	384	2.604	96	10.416

7.3 ADC 控制注意事项

7.3.1 ADC 信号

ADC 高参考电压是内部 Vdd 或外部电压源。ADC 低参考电压为地。ADC 输入信号电压范围必须是在高参考电压和低参考电压之间。

外部高参考电压从 P2.0 输入，必须高于“低参考电压+ 2V”。低参考电压为地。所以外部参考电压范围必须在 2V ~ Vdd 之间。

7.3.2 ADC 编程

执行 ADC 之前首先要设置 ADC 配置，ADC 程序设置流程和注意事项如下所示：

- **步骤 1:** 使能 ADC。ADENB 位控制 ADC，ADENB=1 时使能 ADC，ADENB=0 时禁止 ADC。使能 ADENB 后，系统必须由程序延时 100us 作为 ADC 启动时间，然后设置 ADS 开始 AD 转换。100us 的延迟时间必须在 ADENB 位置 1(而不是 ADS 位置 1)后，否则 AD 转换结果会出错。通常来说，系统在普通工作条件下设置 ADENB 位一次，延迟时间也只执行一次。
- **步骤 2:** 若外部电压源作为 ADC 高参考电压，设置 AVREFHESL=1。ADC 外部高参考电压从 P2.0 输入，必须将 P2.0 设置为无上拉电阻的输入模式。
- **步骤 3:** 通过 CHS[3:0]选择 ADC 输入引脚，并使能 ADC 全局输入。当某个 AIN 引脚作为模拟信号输入引脚时，必须将该引脚设置为输入引脚，并由程序禁止上拉电阻。
- **步骤 4:** 设置 ADS=1，开始 AD 转换。
- **步骤 5:** 通过检测 EOC=1 或者 ADCIF=1，等待 AD 转换结束。若使能 ADC 中断，在 ADC 中断发生时，程序先执行 ADC 中断。ADC 结束后，ADS 自动清零。EOC 实时显示 ADC 的处理状态，ADS=1 时清零。用户无需通过程序清零。

7.3.3 ADC 引脚配置

ADC 输入引脚与 P2 口共用，ADC 输入通道的选择由 ADC_ADM 寄存器 CHS[3:0]控制，CHS[3:0]=000 时选择 AIN0，CHS[4:0]=001 时选择 AIN1.....

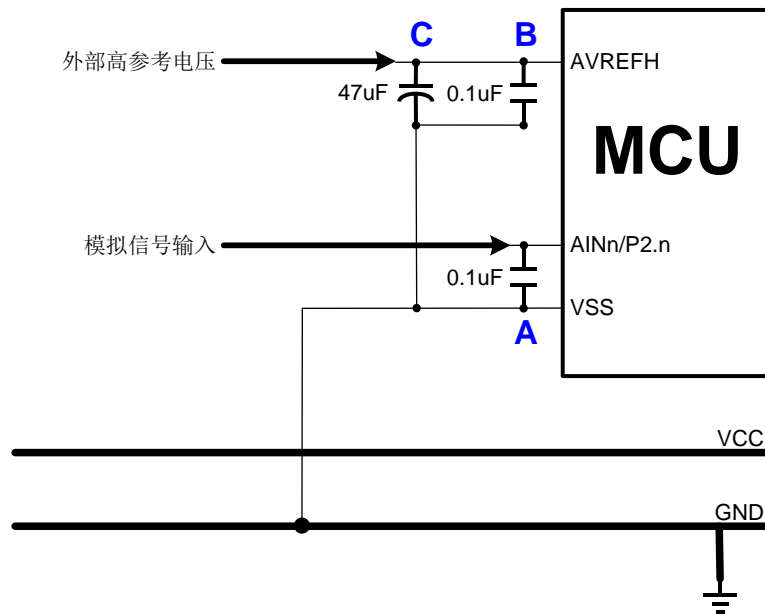
将模拟信号连接到 COMS 的数字输入引脚，特别是模拟信号电平约为 1/2 VDD 时，会造成额外的漏电流。在低功耗模式下，漏电流将是一个大问题。不幸的是，如果用户连接多个模拟输入信号到 P2，也会出现漏电流情况。

当 AVREFHSEL =1 时，P2.0/AIN0 可以作为 ADC 外部高参考电压输入引脚。在此情况下，必须通过程序将 P2.0 GPIO 模式设置为输入模式，禁用 GPIO2_MODE 和 GPIO2_CFG 寄存器(禁止下拉/上拉电阻，禁用 Schmitt 触发器)。

同一时段，只能配置 P2 的一个引脚作为 ADC 输入。设置为 ADC 输入通道的 P2 引脚必须通过程序设置为输入模式，禁用 GPIO2_MODE 和 GPIO2_CFG 寄存器(不开启下拉/上拉电阻，禁用 Schmitt 触发器，数据寄存器保持低)以避免漏电流。

* 注：所有 ADC 输入通道的 GPIO 模式都必须通过程序设置输入模式，禁止禁用 GPIO2_MODE 和 GPIO2_CFG 寄存器(不开启下拉/上拉电阻，禁用 Schmitt 触发器，数据寄存器保持低)。

7.4 ADC 电路



模拟信号从 ADC 输入引脚 AINn/P2.n 输入。在 ADC 输入引脚和 VSS 之间 (A) 必须连接一个 0.1uF 的电容，且要尽可能的靠近 ADC 输入引脚。不能将电容的 GND 直接连接到电源干线上的 GND，必须通过 VSS 引脚。该电容可以减少电源干扰对模拟信号的影响。

ADC 高参考电压由外部电压源提供，外部高参考电压连接到 AVREFH (P2.0) 引脚。在 AVREFH 引脚和 VSS 之间连接电容，首先在图中 C 处连接一个 47uF 的电解电容，再在 B 处连接一个 0.1uF 的电容，且要尽可能的靠近 AVREFH 引脚。不能将电容的 GND 直接连接到电源干线上的 GND，必须通过 VSS 引脚。

7.5 温度传感器（TS）

应用中，不同的温度会改变传感器的特性，为获得温度信息，内置一个温度传感器（TS）用于温度测量，在内部连接到 AIN14 输入通道，用于转换传感器输出电压为数字数据值。

若 TSENB=1，使能温度传感器；TSENB=0 且无需使用温度传感器时，温度传感器关闭

* 注：温度传感器只是一个参考数据而不是实际的空气温度。在精确应用的环境下，请使用外部热敏电阻传感器。

在 25°C 时，V（TS）的典型值约为 1V；若温度上升 10°C，V（TS）就上市约 35mV（VTS=1.035V）；若温度下降 10°C，V（TS）就下降约 35mV（VTS=0.965V）。

示例：

温度	V(TS)	AVrefH	ADC 输出（12 位）
15°C	0.965	3.0V	1250
25°C	1.000V	3.0V	1306
35°C	1.035V	3.0V	1352

通过 ADC 输出 V（TS），可以获取温度信息和系统补偿。

* 注：
1. 每颗芯片的 V（TS）电压和温度曲线可能是不同的，故在使用温度传感器时必须在室温环境下进行校准。
2. 3.53mV/°C 指示典型的温度参数，每颗芯片是不同的。

7.6 ADC 寄存器

基地址：0x4002 6000

7.6.1 ADC 管理寄存器 (ADC_ADM)

地址偏移量：0x00

* 注：

1. 使能 ADC (ADENB=1) 和全局通道 (GCHS=1) 时，ADC 共用引脚切换为 ADC 功能引脚并有 HW 自动禁止该引脚的 GPIO 功能和上拉/下拉电阻，P2.n/AINn 的数字 I/O 功能包括上拉电阻都被屏蔽。
2. 禁止 ADC (ADENB=0) 或全局通道 (GCHS=0) 时，ADC 共用引脚返回到上一个 GPIO 状态。
3. 若设置 P2.0 为外部参考电压输入引脚，则用户需将 P2.0 设置为无上拉电阻的输入模式。

Bit	Name	Description	Attribute	Reset
31:18	Reserved		R	0
17	TSENB	温度传感器使能位。 0: 禁止; 1: 使能。	R/W	0
16:13	Reserved		R	0
12	AVREFHSEL	ADC 参考高电压源选择位。 0: 内部 VDD (P2.0 为 GPIO 引脚或者 AIN0 引脚); 1: 外部参考电压, 由 P2.0 输入。	R/W	0
11	ADENB	ADC 使能位。 0: 禁止; 1: 使能。	R/W	0
10:8	ADCKS[2:0]	ADC 时钟源分频器。 000: ADC_PCLK / 1; 001: ADC_PCLK / 2; 010: ADC_PCLK / 4; 011: ADC_PCLK / 8; 101: ADC_PCLK / 16; 110: ADC_PCLK / 32; 其它: 保留。	R/W	0
7	ADLEN	ADC 分辨率选择控制位。 0: 8 位; 1: 12 位。	R/W	0
6	ADS	ADC 启动控制位。 0: 停止 AD 转换; 1: 开始执行 AD 转换, ADC 结束后, ADS 自动清零。	R/W	0
5	EOC	ADC 状态位, 实时显示 ADC 的处理状态, ADS=1 时自动清零。 0: ADC 处理中; 1: ADC 结束, ADS 位复位。	R/W	0
4	GCHS	ADC 全局通道选择位。 0: 禁止; 1: 使能。	R/W	0
3:0	CHS[3:0]	ADC 输入通道选择位。 0000: AIN0; 0001: AIN1; 0010: AIN2; 0011: AIN3; 0100: AIN4; 0101: AIN5; 0110: AIN6; 0111: AIN7; 1000: AIN8; 1001: AIN10; 1011: AIN11; 1100: AIN12;	R/W	0

1101: AIN13;
1110: AIN14 (温度传感器);
其它: 保留。

7.6.2 ADC 数据寄存器 (ADC_ADB)

地址偏移量: 0x04

ADB 是 ADC 的数据缓存器, 用于存储 AD 转换结果。

* 注: ADC 缓存器 (ADB) 的初始值是未知的。

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11:0	ADB[11:0]	8 位 ADC: ADB11~ADB4。 12 位 ADC: ADB11~ADB0。	R	0

AIN 的输入电压与 ADB 输出数据的对应表:

AIN n	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
0/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	0
1/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	1
.
.
.
4094/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	1

针对不同的应用, 用户可能需要精度介于 8 位到 12 位之间的 AD 转换器。首先, 用户必须选择 12 位分辨率的模式, 进行 AD 转换, 然后在转换结果中去掉最低的几位得到需要的结果。如下表所示:

	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
8-bit	O	O	O	O	O	O	O	O	X	X	X	X
9-bit	O	O	O	O	O	O	O	O	O	X	X	X
10-bit	O	O	O	O	O	O	O	O	O	O	X	X
11-bit	O	O	O	O	O	O	O	O	O	O	O	X
12-bit.	O	O	O	O	O	O	O	O	O	O	O	O

O = Selected, X = Delete

7.6.3 P2 控制寄存器 (ADC_P2CON)

地址偏移量: 0x08

ADC 输入引脚与 P2 共用, 同一时间内, 只能通过 ADM 寄存器设置 P2 口的一个引脚作为 ADC 输入引脚, P2 口的其它引脚作为数字 I/O 引脚。

当输入一个模拟信号到 CMOS 结构端口时, 尤其当模拟信号为 1/2 VDD 时, 可能产生额外的漏电流。省电模式下, 上述漏电流会严重影响到系统的整体功耗。用户连接多个模拟输入信号到 P2 口时, 极容易遇到上述漏电流的情况。

P2CON 为 P2 口的配置寄存器, 使能 ADC 时, 设置 P2CON[15:0], 其对应的 P2 脚将被设为纯模拟信号输入引脚, 从而避免上述漏电流的产生。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	P2CON[15:0]	P2.x 配置控制位 (x=0~15)。 0: P2.x 为模拟输入 (ADC 输入) 引脚或者数字 I/O 引脚; 1: P2.x 为纯模拟输入引脚, 不能作为数字 I/O 引脚。	R	0

7.6.4 ADC 中断使能寄存器 (ADC_IE)

地址偏移量: 0x0C

ADC 中断使能寄存器可以控制 A/D 通道在完成 A/D 转换完成后产生中断。例如：在实际应用中，可以通过 A/D 通道连续地执行 AD 转换来监控传感器。更多数的更新的数据结果是当用户需要时才来读取。在这种情况下，不需要这些 AD 通道在转换结束后产生中断。

Bit	Name	Description	Attribute	Reset
31:15	Reserved		R	0
14:0	IE[14:0]	转换完成后，允许控制 AD 通道产生中断。若 bit x 是 1 个，AIN x 转换完成后产生一个中断。	R/W	0

7.6.5 ADC 中断源状态寄存器 (ADC_RIS)

地址偏移量: 0x10

Bit	Name	Description	Attribute	Reset
31:15	Reserved		R	0
14:0	IF[14:0]	ADC 中断源标志位 (x = 0~ 14)。 0: Read→AINx 没有产生中断; Write→ 相应位清零, 若相应 IE 位置 1 则复位中断; 1: AINx AD 转换完成后请求中断。	R/W	0

8 16 位带捕获功能定时器

8.1 概述

16 位计数/定时器用于对外设时钟（PCLK）信号或外部提供的时钟信号进行计数。根据 4 个匹配寄存器中指定的定时时间，可以使定时器有选择地产生中断请求或完成其它操作。每个计数/定时器都有一路输入信号捕获通道，当输入信号发生变化时，便可将此时间值记录下来，可选择捕获功能触发中断。

PWM 模式下，高达 3 个匹配寄存器可以用于提供单边沿信号的 PWM 输出到匹配输出引脚上。

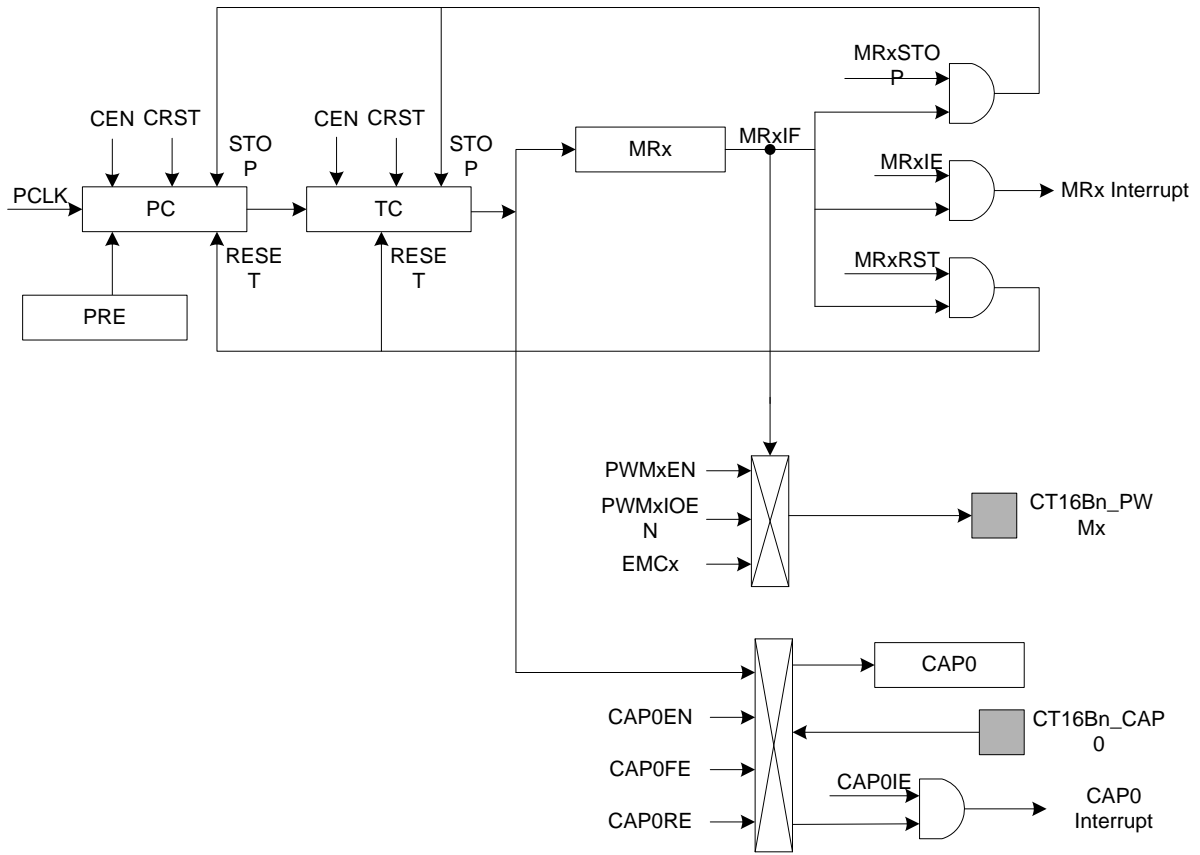
8.2 特性

- 3 个 16 位计数/定时器。
- 计数器和定时器操作。
- 3 路 16 位捕获通道，当有输入信号发生改变时，可以迅速记录下时间。每一捕获事件都可以选择是否产生中断。
- 可以设定定时器时间值在指定的捕获事件发生时被清零。这一特性可以应用于简单的脉宽测量，通过在输入脉冲的起始边沿清除定时器，在输入脉冲的尾端边沿便可取得此脉冲宽度的时间值。
- 4 个 16 位匹配寄存器有如下功能：
 - 连续计数，可选择在计数到匹配数值时是否产生中断。
 - 在计数到匹配数据时停止计数，可选择是否产生中断。
 - 在计数到匹配数据时复位定时器，可选择是否产生中断。
 - PWM 配置，最多允许三个匹配输出作为单边控制 PWM 输出。
- 多达 3 路 PWM 输出，每路 PWM 输出都和某一匹配寄存器一一对应，具体功能如下：
 - 在计数到匹配数据时，输出低。
 - 在计数到匹配数据时，输出高。
 - 在计数到匹配数据时，反转电平。
 - 在计数到匹配数据时，保持状态。

8.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
CT16Bn_CAP0	I	输入 0 的捕获通道	取决于 GPIO _n _CFG
CT16Bn_PWMx	O	Match/PWM 的输出通道 x。	

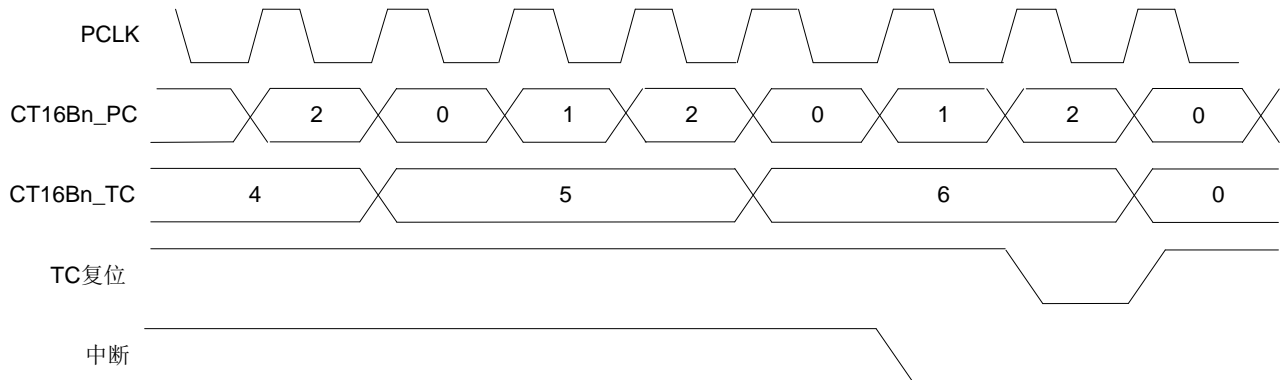
8.4 框图



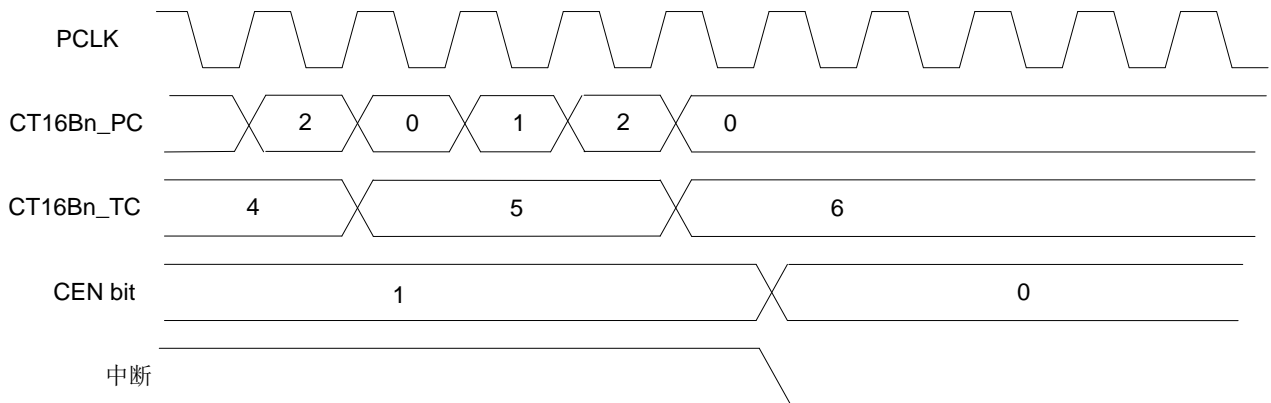
8.5 定时器操作

8.5.1 边沿对齐向上计数模式

下图显示一被配置过的定时器在边沿对齐向上计数模式时复位计数并产生中断请求。CT16Bn_PRE 寄存器设置为 2，CT16Bn_MRx 设置为 6。匹配事件发生后，在定时器周期结束时定时器复位。当定时器计数值达到匹配值后，在它的下一个时钟到达时，匹配事件发生。



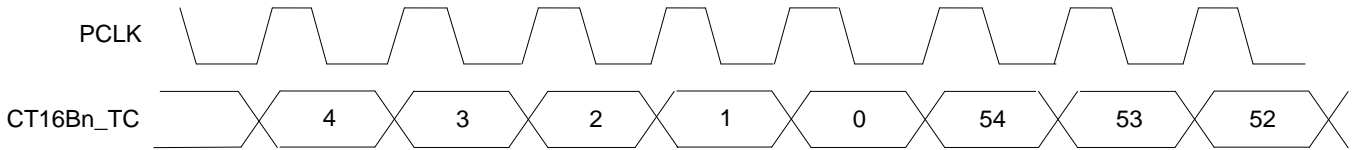
下图显示定时器被配置成在边沿对齐向上计数模式时停止计数，并产生中断。CT16Bn_PRE 寄存器设置为 2，CT16Bn_MRx 设置为 6。定时器达到匹配值后的下一个时钟内，CT16Bn_TMRCTRL 寄存器的 CEN 位清零，中断显示匹配事件发生。



8.5.2 边沿对齐向下计数模式

计数器复位后或者 TC 的值为 0 时，TC 将复位 CT16Bn_MR3 的值，此外，CT16Bn_MR3 的值为 0 时，TC 被阻止。

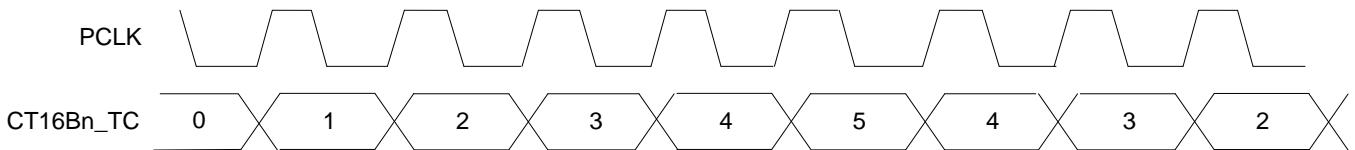
下图显示定时器被配置成在边沿对齐向下计数模式时复位计数状态，CT16Bn_PRE 寄存器设置为 0，CT16Bn_MR3 设置为 54。TC 为 0 时，定时器计数复位并装载 CT16Bn_MR3 的值。



8.5.3 中间对齐计数模式

在中间对齐计数模式下，TC 可以从 0 开始向上记录 CT16Bn_MR3 的值，也可以向下计数直到 0 为止。此外，CT16Bn_MR3 的值为 0 时，TC 被阻止。

下图显示在中间对齐计数模式下的定时器状态，CT16Bn_PRE 寄存器设置为 0，CT16Bn_MR3 设置为 5。



8.6 PWM

8.6.1 PWM 模式 1

在向上计数时，若 $TC < MRn$ ，则 $PWMn$ 为 0。

在向下计数时，若 $TC \leq MRn$ ，则 $PWMn$ 为 0。

边沿对齐向上计数模式操作流程如下：

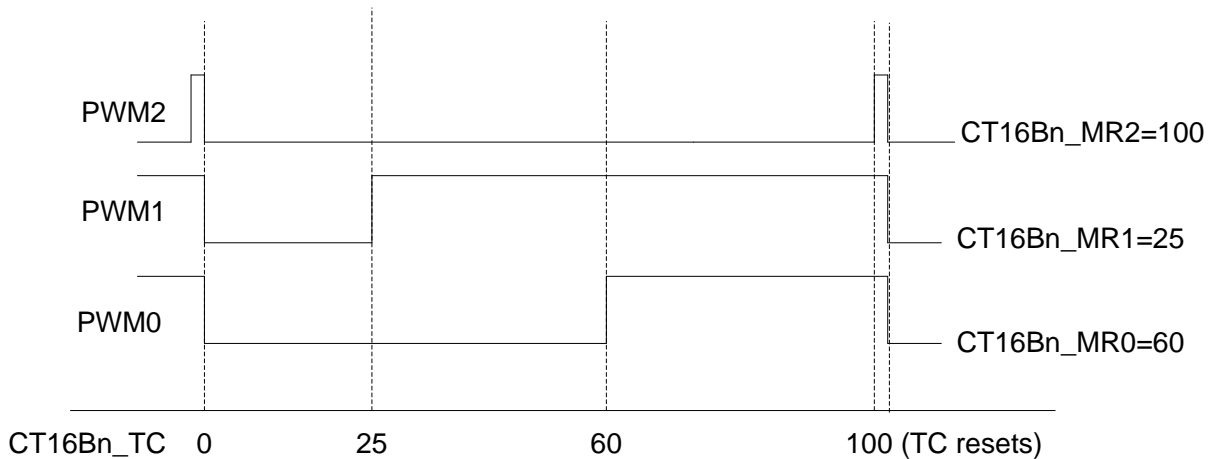
1、在每个 PWM 周期的起始端（定时器设置为 0），所有信号边沿被 PWM 控制输出低电平，除非 $CT16Bn_MR0\sim 3$ 寄存器的匹配值为 0。

2、到达匹配值后，每个 PWM 输出高电平；若没有达到匹配值，则继续输出低电平。

3、若写入 $CT16Bn_MR0\sim 3$ 寄存器的匹配值大于 PWM 周期长度，且 PWM 信号已经是高电平，则在下一个 PWM 周期开始时清除 PWM 信号。

4、若匹配寄存器的值和定时器的复位值（PWM 周期长度）相同，则 PWM 在下一个时钟 tick 输出低电平。因此，PWM 将一直输出由时钟 tick 宽度决定的正极脉冲，其周期取决于 PWM 周期的长度。

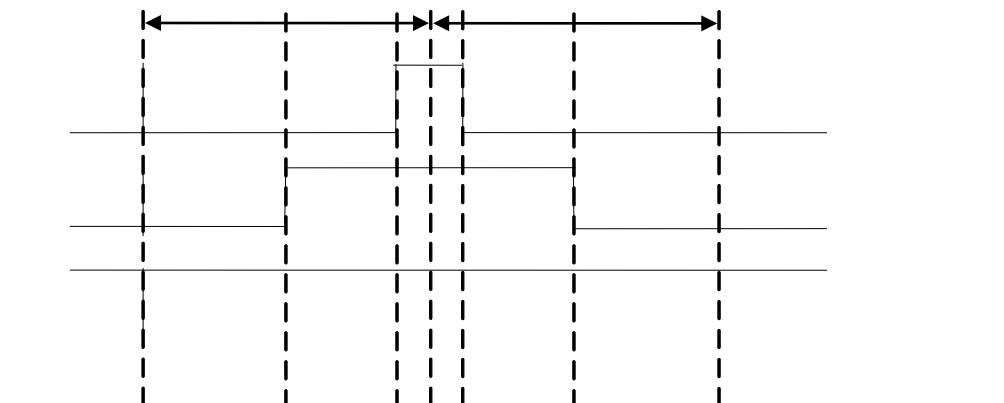
5、若匹配寄存器设置为 0，PWM 首先输出高电平，定时器返回到 0，PWM 会一直保持高电平。



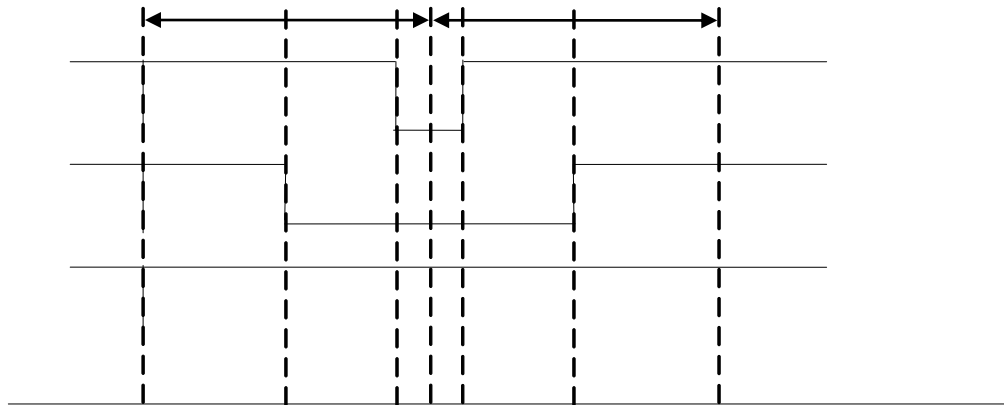
* 注：当选择的匹配输出作为 PWM 输出执行时，除了匹配寄存器需要设置 PWM 周期长度外， $CT16Bn_MCTRL$ 寄存器的定时器复位位（ $MRnRST$ ）和定时器停止位（ $MRnSTOP$ ）都必须为 0。当相应匹配寄存器的值和定时器的值相匹配时，设置 $MRnR$ 位为 1 以使能定时器复位。

下图显示在中间对齐计数模式下的 PWM 模式 1 的波形：

● **Case1:** 寄存器 $CT16Bn_PRE$ 设为 0， $CT16Bn_MR3$ 设为 8， $CT16Bn_MR2$ 设为 7， $CT16Bn_MR1$ 设为 4， $CT16Bn_MR0$ 设为 0。



- **Case 2:** 寄存器CT16Bn_PRE设为 0，CT16Bn_MR3设为 8，CT16Bn_MR2设为 1，CT16Bn_MR1设为 4，CT16Bn_MR0设为 0。



8.6.2 PWM 模式 2

在向下计数时，若 $TC < MRn$ ，则 PWMn 为 1。

在向学计数时，若 $TC \leq MRn$ ，则 PWMn 为 1。

不支持中间对齐计数模式

边沿对齐向上计数模式操作流程如下：

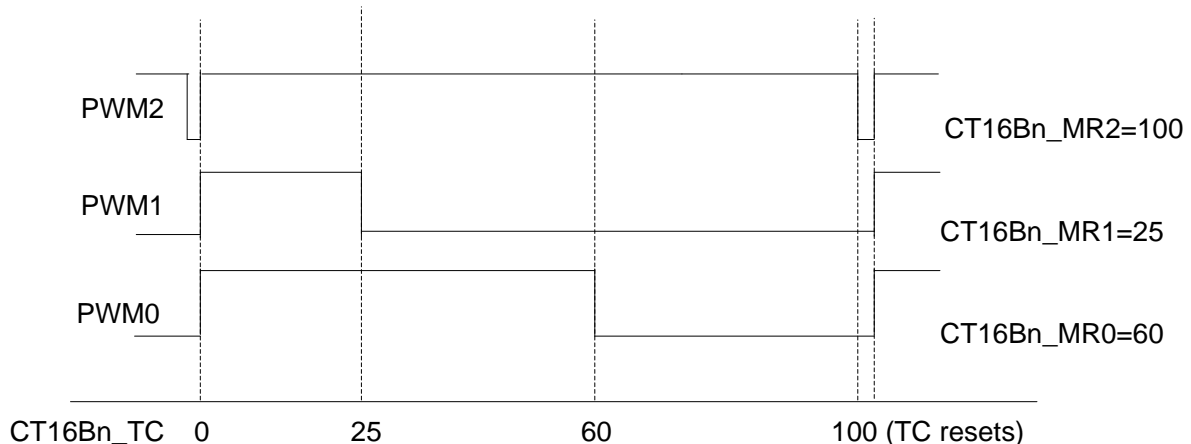
1、在每个 PWM 周期的起始端（定时器设置为 0），所有信号边沿被 PWM 控制输出高电平，除非 CT16Bn_MR0~3 寄存器的匹配值为 0。

2、到达匹配值后，每个 PWM 输出低电平；若没有达到匹配值，则继续输出高电平。

3、若写入 CT16Bn_MR0~3 寄存器的匹配值大于 PWM 周期长度，且 PWM 信号已经是低电平，则在下一个 PWM 周期开始时输出高电平。

4、若匹配寄存器的值和定时器的复位值（PWM 周期长度）相同，则 PWM 在下一个时钟 tick 输出高电平。因此，PWM 将一直输出由时钟 tick 宽度组成的低电平脉冲，其周期取决于 PWM 周期的长度。

5、若匹配寄存器设置为 0，PWM 首先输出低电平，定时器返回到 0，PWM 会一直保持低电平。



* 注：当选择的匹配输出作为 PWM 输出执行时，除了匹配寄存器需要设置 PWM 周期长度外，CT16Bn_MCTRL 寄存器的定时器复位位（MRnRST）和定时器停止位（MRnSTOP）都必须为 0。当相应匹配寄存器的值和定时器的值相匹配时，设置 MRnR 位为 1 以启用定时器复位。

8.7 CT16Bn 寄存器

基地址：0x4000 0000 (CT16B0)
0x4000 2000 (CT16B1)
0x4000 4000 (CT16B2)

8.7.1 CT16Bn 定时器控制寄存器 (CT16Bn_TMRCTRL) (n=0,1,2)

地址偏移量：0x00

* 注：应该最后设置 CEN 位。

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:4	CM[2:0]	计数模式选择控制位。 000: 边沿对齐向上计数模式; 001: 边沿对齐向下计数模式; 010: 中间对齐模式 1, 在向下计数周期时所对应的中断标志位置 1; 100: 中间对齐模式 2, 在向上计数周期时所对应的中断标志位置 1; 110: 中间对齐模式 3, 在向上/向下计数周期时所对应的中断标志位都置 1; 其它: 保留。	R/W	0
3:2	Reserved		R	0
1	CRST	计数器复位位。 0: 禁止计数器复位; 1: 在 PCLK 的下一个正极边沿, 定时/计数器同步复位, 复位完成后由 HW 将该位清零。	R/W	0
0	CEN	计数器使能位。 0: 禁止; 1: 使能定时器/计数器计数。	R/W	0

8.7.2 CT16Bn 定时器计数器寄存器 (CT16Bn_TC) (n=0,1,2)

地址偏移量：0x04

在边沿对齐向上计数模式(CM[2:0]=000b)下, 除非计数器在到达上限值之前复位, 否则 TC 一直计数到 0x0000FFFF, 然后再返回 0x00000000。这并不会引起中断, 但有需要的时候, 可以由匹配寄存器来检测是否溢出。

在边沿向下计数器模式 (CM[2:0]=001b) 下, 计数器复位后 (SW 设置 CRST 为 1), TC[15:0]会复位 CT16Bn_MR3 的值。

Bit	Name	Description	Attribute	Reset
31:0	Reserved		R	0
15:0	TC[15:0]	定时器计数器。	R/W	0

8.7.3 CT16Bn 前置分频寄存器 (CT16Bn_PRE) (n=0,1,2)

地址偏移量：0x08

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	PR[15:0]	前置分频最大值。	R/W	0

8.7.4 CT16Bn 前置分频计数器寄存器 (CT16Bn_PC) (n=0,1,2)

地址偏移量: 0x0C

在应用于定时器计数器之前, 16 位预分频计数器通过部分恒定值控制 PCLK 的分频。它可以控制定时器的分辨率和最大值之间的关系。预分频计数器在每个 PCLK 递增计数, 在达到存储在预分频寄存器的值后, 定时器计数器继续计数, 而预分频计数器在下一个 PCLK 复位。它促使下列情况出现: PR=0 时, TC 在每个 PCLK 递增计数; PR=1 是, 则在每 2 个 PCLK 递增计数。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	PC[15:0]	预分频计数器	R/W	0

8.7.5 CT16Bn 计数控制寄存器 (CT16Bn_CNTCTRL) (n=0,1,2)

地址偏移量: 0x10

该寄存器用于选择定时器模式或者计数器模式, 计数器模式下选择用于计数的引脚和边沿。

选择计数器模式作为操作模式时, CAP 输入 (由 CIS 位选择) 在 PCLK 时钟的每个上升沿采样。比较两个连续的 CAP 输入采样信号后, 在选择的 CAP 输入电平上, 下列 4 个事件中的其中一件是经过验证的: 上升沿, 下降沿, 边沿或者 CAP 输入电平无变换。只有确认过的事件发生, 而且该事件符合该寄存器的 CTM 位选择的事件, 定时器计数器寄存器才会递增计数。

为了使外部提供给计数器的时钟信号可以得到有效处理, 计数器对提供的外部时钟信号会有所限制。因为两个连续的 PCLK 时钟的上升沿才能确定 CAP 输入引脚上输入信号的一个跳变边沿, 所以 CAP 输入信号的频率不能超过 PCLK 时钟频率的 1/2, 即通过 CAP 输入的高/低电平的持续时间不能短于 $1/(2 \times \text{PCLK})$ 。

* 注: 在 CNTCTRL 寄存器下选择计数器模式, 捕获控制寄存器 (CAPCTRL) 必须由程序设置为 0x00。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3:2	CIS[1:0]	计数输入选择。在计数器模式 (CTM[1:0]不是 00) 下, 该位选择 CAP 引脚对时钟进行采样。 00: CT16Bn_CAP0; 其它: 保留。	R/W	0
1:0	CTM[1:0]	计数器/定时器模式。此字段选择 PCLK 的上升沿将 PC 清零并且定时器计数器递增计数。 00: 定时器模式, PCLK 的每个上升沿; 01: 计数器模式, TC 在 CIS 位选择的 CAP0 输入的上升沿递增计数; 10: 计数器模式, TC 在 CIS 位选择的 CAP0 输入的下降沿递增计数; 11: 计数器模式, TC 在 CIS 位选择的 CAP0 输入的上升/下降沿递增计数。	R/W	0

8.7.6 CT16Bn 匹配控制寄存器 (CT16Bn_MCTRL) (n=0,1,2)

地址偏移量: 0x14

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11	MR3STOP	停止 MR3: 若 MR3 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
10	MR3RST	MR3 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
9	MR3IE	MR3 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
8	MR2STOP	停止 MR2: 若 MR2 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
7	MR2RST	MR2 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
6	MR2IE	MR2 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
5	MR1STOP	停止 MR1: 若 MR1 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
4	MR1RST	MR1 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
3	MR1IE	MR1 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
2	MR0STOP	停止 MR0: 若 MR0 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
1	MR0RST	MR0 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
0	MR0IE	MR0 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0

8.7.7 CT16Bn 匹配寄存器 0~3 (CT16Bn_MR0~3) (n=0,1,2)

地址偏移量: 0x18, 0x1C, 0x20, 0x24

匹配寄存器的值不断地与定时器计数器 (TC) 的值进行比较, 当 2 个值相等时, 自动触发动作。该动作可能是产生中断, 复位 TC 或者停止定时器, 通过设置 CT16Bn_MCTRL 寄存器控制。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	MR[15:0]	TC 匹配值。	R/W	0

8.7.8 CT16Bn 捕获控制寄存器 (CT16Bn_CAPCTRL) (n=0,1,2)

地址偏移量: 0x28

发生捕获事件时, 捕获寄存器是否将值装入计数器/定时器, 或者是否由捕获事件产生中断, 都通过捕获控制寄存器来控制。同时设置上升位和下降位为有效配置, 这样导致上升/下降沿发生捕获事件。

* 注: CAP0EN=1 时, HW 直接切换 I/O 配置。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CAP0EN	捕获 0 功能使能位。 0: 禁止; 1: 使能。	R/W	0
2	CAP0IE	CT16Bn_CAP0 事件中断: CT16Bn_CAP0 事件产生中断。 0: 禁止; 1: 使能。	R/W	0
1	CAP0FE	CT16Bn_CAP0 下降沿捕获: 当 CT16Bn_CAP0 检测到输入信号按照先 1 后 0 的顺序变化时, CAP0 保存此时 TC 数值。 0: 禁止; 1: 使能。	R/W	0
0	CAP0RE	CT16Bn_CAP0 上升沿捕获: 当 CT16Bn_CAP0 检测到输入信号按照先 0 后 1 的顺序变化时, CAP0 保存此时 TC 数值。 0: 禁止; 1: 使能。	R/W	0

8.7.9 CT16Bn 捕获 0 寄存器 (CT16Bn_CAP0) (n=0,1,2)

地址偏移量: 0x2C

每个捕获寄存器都与一个设备引脚相对应, 当此设备引脚上有指定的事件发生时, 计数器/定时器数值便可以保存在对应的捕获寄存器中。捕获控制寄存器的设置决定是否使能捕获功能, 以及捕获事件是否在相关联的引脚的上升沿、下降沿或双边沿发生。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	CAP0[15:0]	定时器计数器捕获值。	R	0

8.7.10 CT16Bn 外部匹配寄存器 (CT16Bn_EM) (n=0,1,2)

地址偏移量: 0x30

外部匹配寄存器提供 CT16Bn_PWM[1:0]的控制和状态。若匹配输出配置为 PWM 输出，外部匹配寄存器的功能由 PWM 规则决定。

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:8	EMC2[1:0]	CT16Bn_PWM2 功能决定位。 00: 无功能; 01: CT16Bn_PWM2 引脚为低电平; 10: CT16Bn_PWM2 引脚为高电平; 11: 反转 CT16Bn_PWM2 引脚状态。	R/W	0
7:6	EMC1[1:0]	CT16Bn_PWM1 功能决定位。 00: 无功能; 01: CT16Bn_PWM1 引脚为低电平; 10: CT16Bn_PWM1 引脚为高电平; 11: 反转 CT16Bn_PWM1 引脚状态。	R/W	0
5:4	EMC0[1:0]	CT16Bn_PWM0 功能决定位。 00: 无功能; 01: CT16Bn_PWM0 引脚为低电平; 10: CT16Bn_PWM0 引脚为高电平; 11: 反转 CT16Bn_PWM0 引脚状态。	R/W	0
3	Reserved		R	0
2	EM2	TC 和 MR2 相等时, 该位根据 EMC2 位进行动作, 并驱动 CT16Bn_PWM2 输出状态。	R/W	0
1	EM1	TC 和 MR1 相等时, 该位根据 EMC1 位进行动作, 并驱动 CT16Bn_PWM1 输出状态。	R/W	0
0	EM0	TC 和 MR0 相等时, 该位根据 EMC0 位进行动作, 并驱动 CT16Bn_PWM0 输出状态。	R/W	0

8.7.11 CT16Bn PWM 控制寄存器 (CT16Bn_PWMCTRL) (n=0,1,2)

地址偏移量: 0x34

PWM 控制寄存器用于配置匹配输出作为 PWM 输出。每个匹配输出都可独立的作为 PWM 输出或者匹配输出执行，其功能由 CT16Bn_EM 寄存器控制。

对于每个定时器来说,最多控制 3 路 PWM 输出,由 CT16Bn_PWMCTRL[2:0]选择输出。额外的匹配寄存器决定 PWM 周期长度。在其它的匹配寄存器发生匹配时, PWM 输出高电平。定时器由设置 PWM 周期长度的匹配寄存器复位。定时器复位为 0 时,将当前配置为 PWM 输出的高电平输出都清零。

Bit	Name	Description	Attribute	Reset
31:23	Reserved		R	0
22	PWM2IOEN	CT16Bn_PWM2/GPIO 选择位。 0: CT16Bn_PWM2 引脚作为 GPIO 引脚; 1: CT16Bn_PWM2 引脚作为匹配输出, 其输出信号取决于 PWM2EN 位。	R/W	0
21	PWM1IOEN	CT16Bn_PWM1/GPIO 选择位。 0: CT16Bn_PWM1 引脚作为 GPIO 引脚; 1: CT16Bn_PWM1 引脚作为匹配输出, 其输出信号取决于 PWM1EN 位。	R/W	0
20	PWM0IOEN	CT16Bn_PWM0/GPIO 选择位。 0: CT16Bn_PWM0 引脚作为 GPIO 引脚; 1: CT16Bn_PWM0 引脚作为匹配输出, 其输出信号取决于 PWM0EN 位。	R/W	0
19:10	Reserved		R	0
9:8	PWM2MODE[1:0]	PWM2 输出模式控制位。 00: PWM 模式 1; 01: PWM 模式 2; 10: 强制为 0; 11: 强制为 1。	R/W	0
7:6	PWM1MODE[1:0]	PWM1 输出模式控制位。 00: PWM 模式 1; 01: PWM 模式 2; 10: 强制为 0; 11: 强制为 1。	R/W	0
5:4	PWM0MODE[1:0]	PWM0 输出模式控制位。 00: PWM 模式 1; 01: PWM 模式 2; 10: 强制为 0; 11: 强制为 1。	R/W	0
3	Reserved		R	0
2	PWM2EN	PWM2 使能位。 0: CT16Bn_PWM2 由 EM2 控制; 1: 使能 CT16Bn_PWM2 的 PWM 模式。	R/W	0
1	PWM1EN	PWM1 使能位。 0: CT16Bn_PWM1 由 EM1 控制; 1: 使能 CT16Bn_PWM1 的 PWM 模式。	R/W	0
0	PWM0EN	PWM0 使能位。 0: CT16Bn_PWM0 由 EM0 控制; 1: 使能 CT16Bn_PWM0 的 PWM 模式。	R/W	0

8.7.12 CT16Bn 定时器中断源状态寄存器 (CT16Bn_RIS) (n=0,1,2)

地址偏移量: 0x38

该寄存器显示定时器/PWM 中断的原始状态, 若 CT16Bn_IE 寄存器的相应位置 1, 定时器/PWM 中断发送到中断控制器。

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	CAP0IF	捕获通道 0 的中断标志位。 0: CAP0 无中断; 1: CAP0 发生中断请求。	R	0
3	MR3IF	匹配通道 3 的中断标志位。 0: 匹配通道 3 无中断; 1: 匹配中断 3 发生中断请求。	R	0
2	MR2IF	匹配通道 2 的中断标志位。 0: 匹配通道 2 无中断; 1: 匹配中断 2 发生中断请求。	R	0
1	MR1IF	匹配通道 1 的中断标志位。 0: 匹配通道 1 无中断; 1: 匹配中断 1 发生中断请求。	R	0
0	MR0IF	匹配通道 0 的中断标志位。 0: 匹配通道 0 无中断; 1: 匹配中断 0 发生中断请求。	R	0

8.7.13 CT16Bn 定时器中断清零寄存器 (CT16Bn_IC) (n=0,1,2)

地址偏移量: 0x3C

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	CAP0IC	0: 无影响; 1: CAP0IF 位清零。	W	0
3	MR3IC	0: 无影响; 1: MR3IF 位清零。	W	0
2	MR2IC	0: 无影响; 1: MR2IF 位清零。	W	0
1	MR1IC	0: 无影响; 1: MR1IF 位清零。	W	0
0	MR0IC	0: 无影响; 1: MR0IF 位清零。	W	0

9 32 位带捕获功能定时器

9.1 概述

32 位计数/定时器用于对外设时钟（PCLK）信号或外部提供的时钟信号进行计数。根据 4 个匹配寄存器中指定的定时时间，可以使定时器有选择地产生中断请求或完成其它操作。每个计数/定时器都有一路输入信号捕获通道，当输入信号发生变化时，便可将此时间值记录下来，可选择捕获功能触发中断。

PWM 模式下，高达 4 个匹配寄存器可以用于提供单边沿信号的 PWM 输出到匹配输出引脚上。

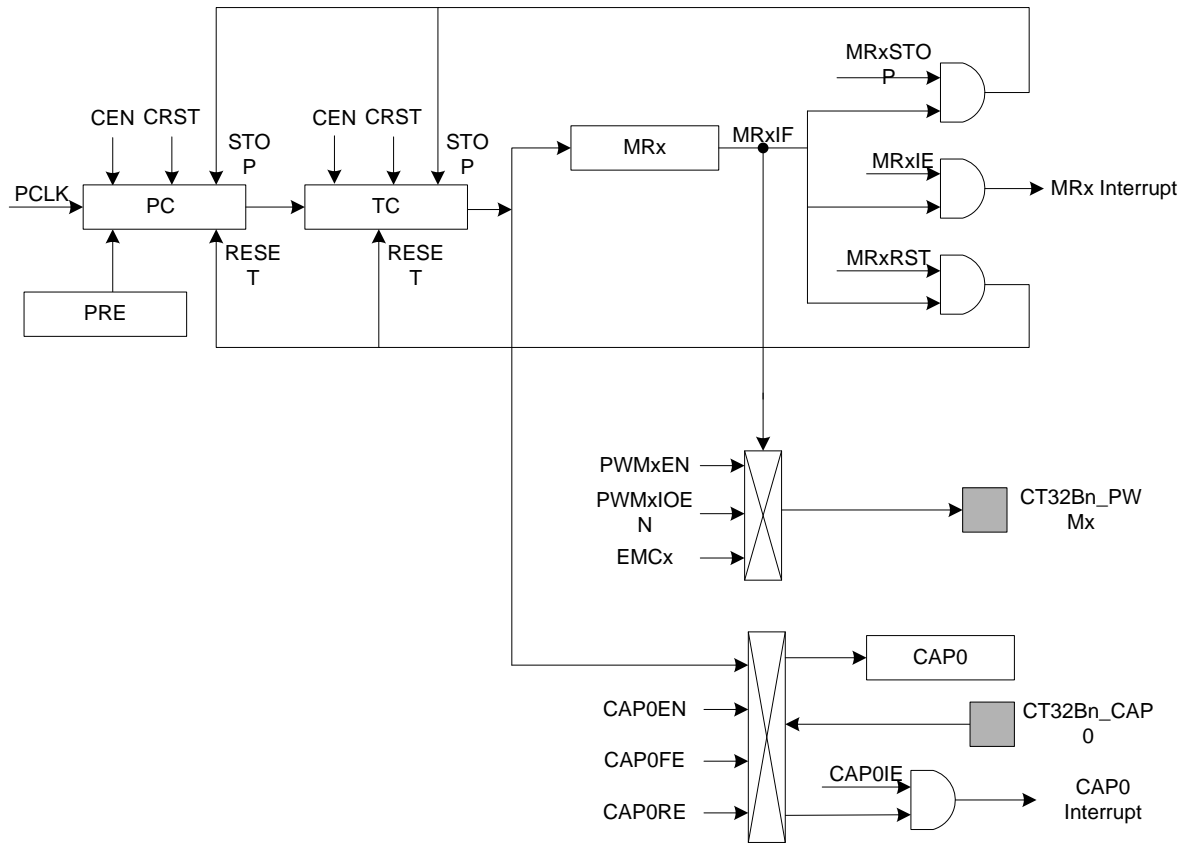
9.2 特性

- 3 个 32 位计数/定时器。
- 计数器和定时器操作。
- 3 路 32 位捕获通道，当有输入信号发生改变时，可以迅速记录下时间。每一捕获事件都可以选择是否产生中断。
- 可以设定定时器时间值在指定的捕获事件发生时被清零。这一特性可以应用于简单的脉宽测量，通过在输入脉冲的起始边沿清除定时器，在输入脉冲的尾端边沿便可取得此脉冲宽度的时间值。
- 4 个 32 位匹配寄存器有如下功能：
 - 连续计数，可选择在计数到匹配数值时是否产生中断。
 - 在计数到匹配数据时停止计数，可选择是否产生中断。
 - 在计数到匹配数据时复位定时器，可选择是否产生中断。
- PWM 配置，最多允许四个匹配输出作为单边控制 PWM 输出。
- 多达 4 路 PWM 输出，每路 PWM 输出都和某一匹配寄存器一一对应，具体功能如下：
 - 在计数到匹配数据时，输出低。
 - 在计数到匹配数据时，输出高。
 - 在计数到匹配数据时，反转电平。
 - 在计数到匹配数据时，保持状态。

9.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
CT32Bn_CAP0	I	输入 0 的捕获通道	取决于 GPIO _n _CFG
CT32Bn_PWMx	O	Match/PWM 的输出通道 x。	

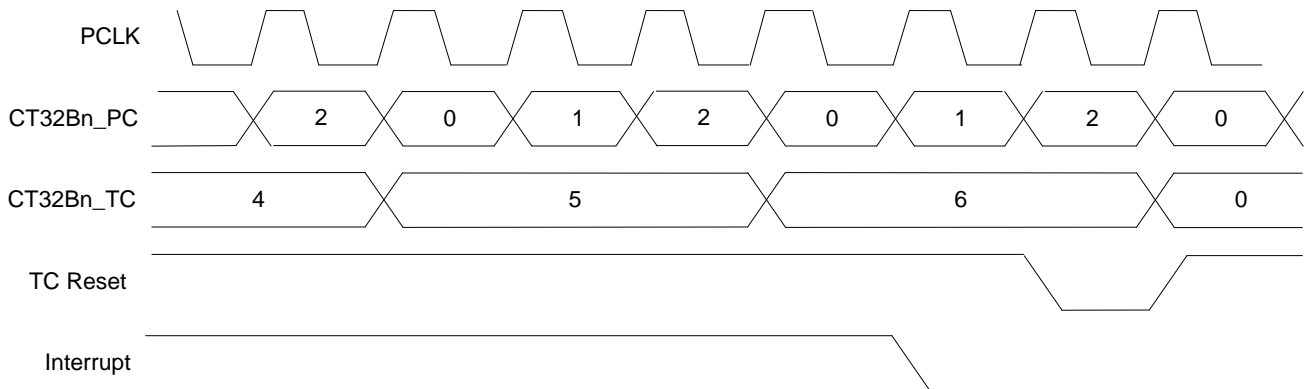
9.4 框图



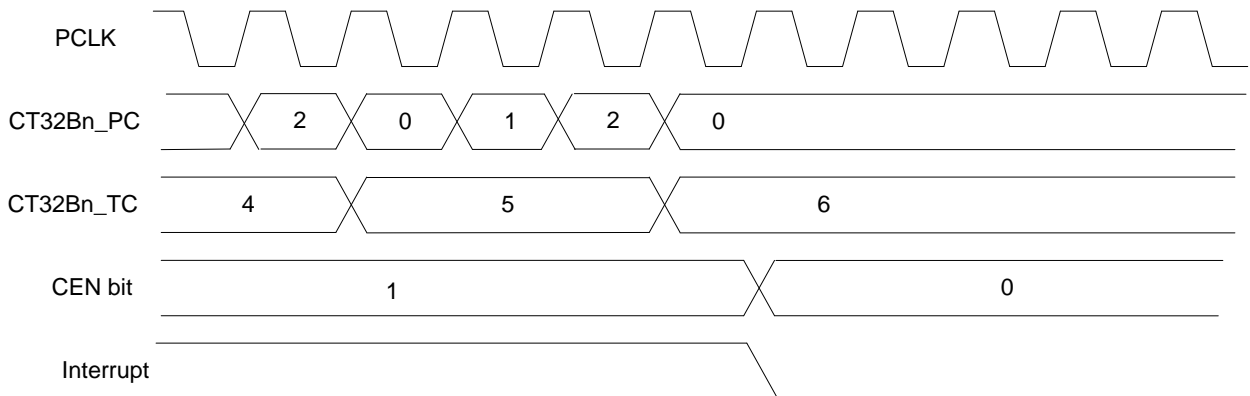
9.5 定时器操作

9.5.1 边沿对齐向上计数模式

下图显示一被配置过的定时器在边沿对齐向上计数模式时复位计数并产生中断请求。CT32Bn_PRE 寄存器设置为 2，CT32Bn_MRx 设置为 6。匹配事件发生后，在定时器周期结束时定时器复位。当定时器计数值达到匹配值后，在它的下一个时钟到达时，匹配事件发生。



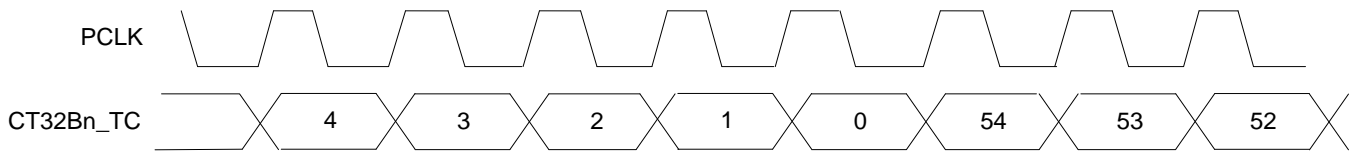
下图显示定时器被配置成在边沿对齐向上计数模式时停止计数，并产生中断。CT32Bn_PRE 寄存器设置为 2，CT32Bn_MRx 设置为 6。定时器达到匹配值后的下一个时钟内，CT32Bn_TMRCTRL 寄存器的 CEN 位清零，中断显示匹配事件发生。



9.5.2 边沿对齐向下计数模式

计数器复位后或者 TC 的值为 0 时，TC 将复位 CT32Bn_MR3 的值，此外，CT32Bn_MR3 的值为 0 时，TC 被阻止。

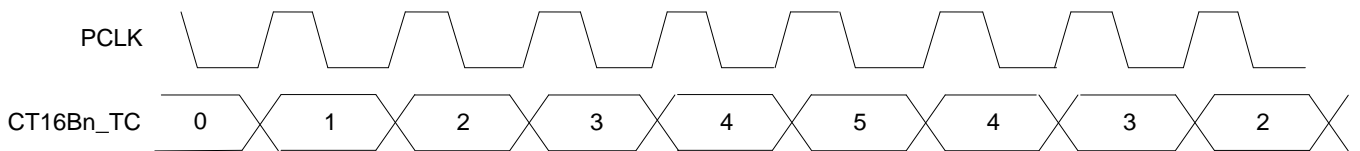
下图显示定时器被配置成在边沿对齐向下计数模式时复位计数状态，CT32Bn_PRE 寄存器设置为 0，CT32Bn_MR3 设置为 54。TC 为 0 时，定时器计数复位并装载 CT32Bn_MR3 的值。



9.5.3 中间对齐计数模式

在中间对齐计数模式下，TC 可以从 0 开始向上记录 CT32Bn_MR3 的值，也可以向下计数直到 0 为止。此外，CT32Bn_MR3 的值为 0 时，TC 被阻止。

下图显示在中间对齐计数模式下的定时器状态，CT32Bn_PRE 寄存器设置为 0，CT32Bn_MR3 设置为 5。



9.6 PWM

9.6.1 PWM 模式 1

在向上计数时，若 $TC < MRn$ ，则 $PWMn$ 为 0。

在向下计数时，若 $TC \leq MRn$ ，则 $PWMn$ 为 0。

边沿对齐向上计数模式操作流程如下：

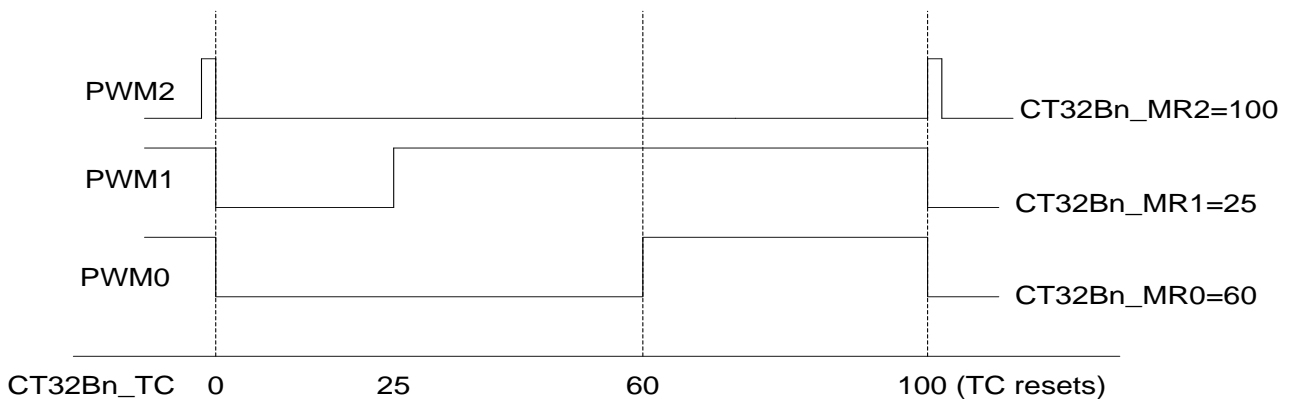
1、在每个 PWM 周期的起始端（定时器设置为 0），所有信号边沿被 PWM 控制输出低电平，除非 $CT32Bn_MR0\sim 3$ 寄存器的匹配值为 0。

2、到达匹配值后，每个 PWM 输出高电平；若没有达到匹配值，则继续输出低电平。

3、若写入 $CT32Bn_MR0\sim 3$ 寄存器的匹配值大于 PWM 周期长度，且 PWM 信号已经是高电平，则在下一个 PWM 周期开始时清除 PWM 信号。

4、若匹配寄存器的值和定时器的复位值（PWM 周期长度）相同，则 PWM 在下一个时钟 tick 输出低电平。因此，PWM 将一直输出由时钟 tick 宽度决定的正极脉冲，其周期取决于 PWM 周期的长度。

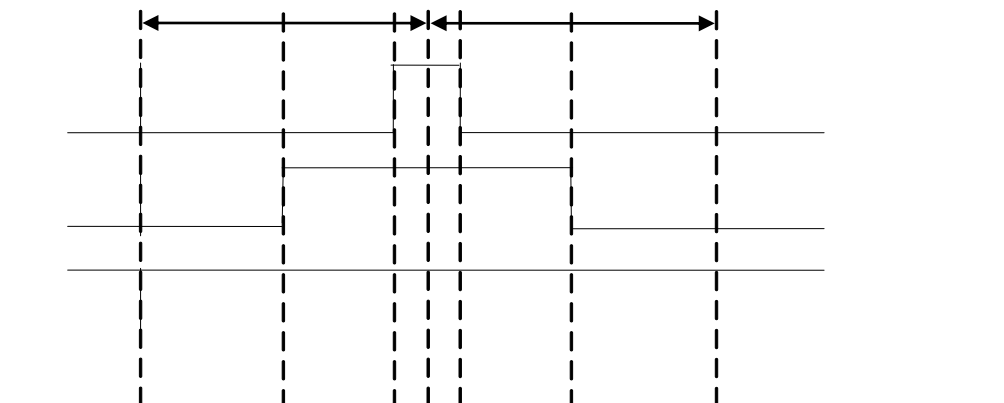
5、若匹配寄存器设置为 0，PWM 首先输出高电平，定时器返回到 0，PWM 会一直保持高电平。



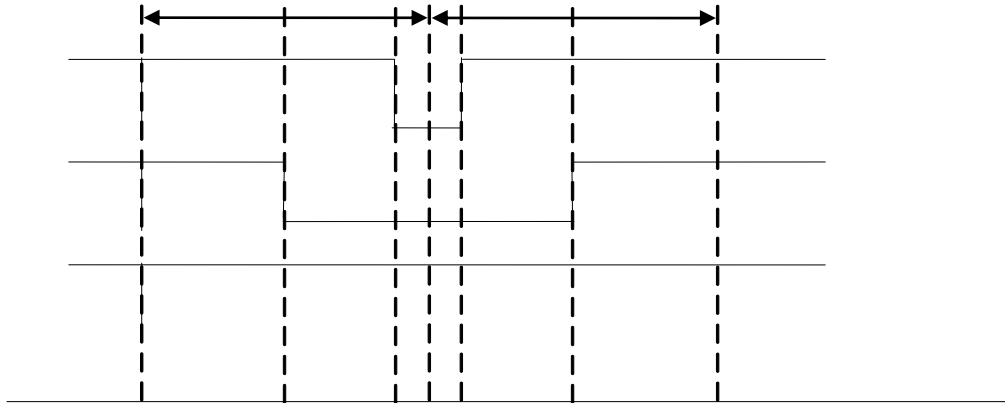
* 注：当选择的匹配输出作为 PWM 输出执行时，除了匹配寄存器需要设置 PWM 周期长度外， $CT32Bn_MCTRL$ 寄存器的定时器复位位（ $MRnRST$ ）和定时器停止位（ $MRnSTOP$ ）都必须为 0。当相应匹配寄存器的值和定时器的值相匹配时，设置 $MRnR$ 位为 1 以使能定时器复位。

下图显示在中间对齐计数模式下的 PWM 模式 1 的波形：

● **Case1:** 寄存器 $CT26Bn_PRE$ 设为 0， $CT32Bn_MR3$ 设为 8， $CT32Bn_MR2$ 设为 7， $CT32Bn_MR1$ 设为 4， $CT32Bn_MR0$ 设为 0。



- **Case2:** 寄存器CT32Bn_PRE设为 0, CT32Bn_MR3设为 8, CT32Bn_MR2设为 1, CT32Bn_MR1设为 4, CT32Bn_MR0设为 0。

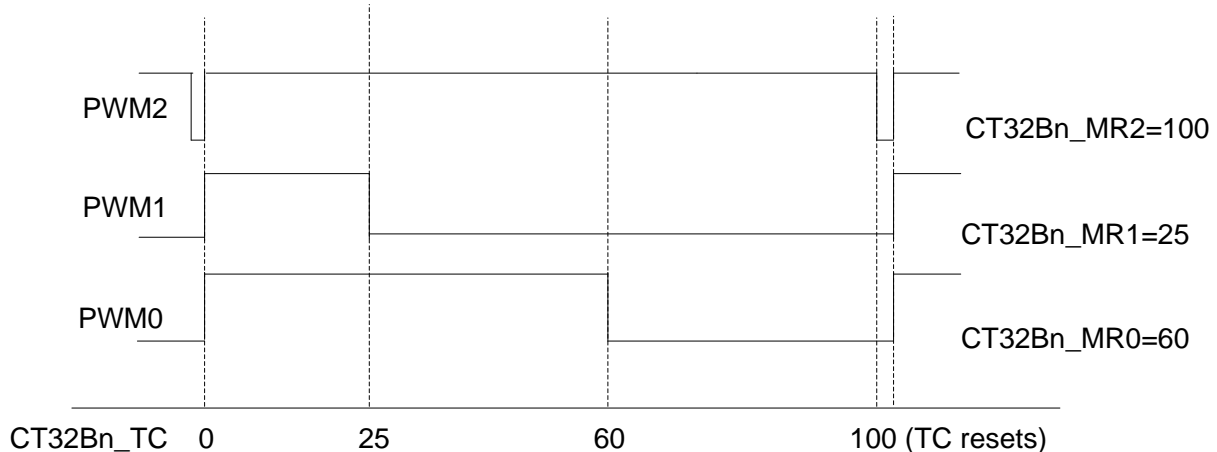


9.6.2 PWM 模式 2

在向下计数时, 若 $TC < MR_n$, 则 PWM_n 为 1。
在向上计数时, 若 $TC \leq MR_n$, 则 PWM_n 为 1。

边沿对齐向上计数模式操作流程如下:

- 1、在每个 PWM 周期的起始端 (定时器设置为 0), 所有信号边沿被 PWM 控制输出高电平, 除非 CT32Bn_MR0~3 寄存器的匹配值为 0。
- 2、到达匹配值后, 每个 PWM 输出低电平; 若没有达到匹配值, 则继续输出高电平。
- 3、若写入 CT32Bn_MR0~3 寄存器的匹配值大于 PWM 周期长度, 且 PWM 信号已经是低电平, 则在下一个 PWM 周期开始时输出高电平。
- 4、若匹配寄存器的值和定时器的复位值 (PWM 周期长度) 相同, 则 PWM 在下一个时钟 tick 输出高电平。因此, PWM 将一直输出由时钟 tick 宽度组成的低电平脉冲, 其周期取决于 PWM 周期的长度。
- 5、若匹配寄存器设置为 0, PWM 首先输出低电平, 定时器返回到 0, PWM 会一直保持低电平。



* 注: 当选择的匹配输出作为 PWM 输出执行时, 除了匹配寄存器需要设置 PWM 周期长度外, CT32Bn_MCTRL 寄存器的定时器复位位 (MRnRST) 和定时器停止位 (MRnSTOP) 都必须为 0。当相应匹配寄存器的值和定时器的值相匹配时, 设置 MRnR 位为 1 以使能定时器复位。

9.7 CT32Bn 寄存器

基地址: 0x4000 6000 (CT32B0)
0x4000 8000 (CT32B1)
0x4000 A000 (CT32B2)

9.7.1 CT32Bn 定时器控制寄存器 (CT32Bn_TMRCTRL) (n=0,1,2)

地址偏移量: 0x00

* 注: 应该最后设置 CEN 位。

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:4	CM[2:0]	计数模式选择控制位。 000: 边沿对齐向上计数模式; 001: 边沿对齐向下计数模式; 010: 中间对齐模式 1, 在向下计数周期时所对应的中断标志位置 1; 100: 中间对齐模式 2, 在向上计数周期时所对应的中断标志位置 1; 110: 中间对齐模式 3, 在向上/向下计数周期时所对应的中断标志位都置 1; 其它: 保留。	R/W	0
3:2	Reserved		R	0
1	CRST	计数器复位位。 0: 禁止计数器复位; 1: 在 PCLK 的下一个正极边沿, 定时/计数器同步复位, 复位完成后由 HW 将该位清零。	R/W	0
0	CEN	计数器使能位。 0: 禁止; 1: 使能定时器/计数器计数。	R/W	0

9.7.2 CT32Bn 定时器计数器寄存器 (CT32Bn_TC) (n=0,1,2)

地址偏移量: 0x04

在边沿对齐向上计数模式(CM[2:0]=000b)下, 除非计数器在到达上限值之前复位, 否则 TC 一直计数到 0xFFFFFFFF, 然后再返回 0x00000000。这并不会引起中断, 但有需要的时候, 可以由匹配寄存器来检测是否溢出。

在边沿向下计数器模式 (CM[2:0]=001b) 下, 计数器复位后 (SW 设置 CRST 为 1), TC[31:0]会复位 CT32Bn_MR3 的值。

Bit	Name	Description	Attribute	Reset
31:0	TC[31:0]	定时器计数器。	R/W	0

9.7.3 CT32Bn 前置分频寄存器 (CT32Bn_PRE) (n=0,1,2)

地址偏移量: 0x08

Bit	Name	Description	Attribute	Reset
31:0	PRE[31:0]	Prescale max value	R/W	0

9.7.4 CT32Bn 前置分频计数器寄存器 (CT32Bn_PC) (n=0,1,2)

地址偏移量: 0x0C

在应用于定时器计数器之前, 32 位预分频计数器通过部分恒定值控制 PCLK 的分频。它可以控制定时器的分辨率和最大值之间的关系。预分频计数器在每个 PCLK 递增计数, 在达到存储在预分频寄存器的值后, 定时器计数器继续计数, 而预分频计数器在下一个 PCLK 复位。它促使下列情况出现: PR=0 时, TC 在每个 PCLK 递增计数; PR=1 是, 则在每 2 个 PCLK 递增计数。

Bit	Name	Description	Attribute	Reset
31:0	PC[31:0]	预分频计数器	R/W	0

9.7.5 CT32Bn 计数控制寄存器 (CT32Bn_CNTCTRL) (n=0,1,2)

地址偏移量: 0x10

该寄存器用于在选择定时器模式或者计数器模式, 计数器模式下选择用于计数的引脚和边沿。

选择计数器模式作为操作模式时, CAP 输入 (由 CIS 位选择) 在 PCLK 时钟的每个上升沿采样。比较两个连续的 CAP 输入采样信号后, 在选择的 CAP 输入电平上, 下列的某一个事件是经过验证的: 上升沿, 下降沿, 边沿或者 CAP 输入电平无变换。只有确认过的事件发生, 而且该事件符合该寄存器的 CTM 位选择的事件, 定时器计数器寄存器才会递增计数。

为了使外部提供给计数器的时钟信号可以得到有效处理, 计数器对提供的外部时钟信号会有所限制。因为两个连续的 PCLK 时钟的上升沿才能确定 CAP 输入引脚上输入信号的一个跳变边沿, 所以 CAP 输入信号的频率不能超过 PCLK 时钟频率的 1/2, 即通过 CAP 输入的高/低电平的持续时间不能短于 $1/(2 \times \text{PCLK})$ 。

* 注: 在 CNTCTRL 寄存器下选择计数器模式, 捕获控制寄存器 (CAPCTRL) 必须由程序设置为 0x00。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3:2	CIS[1:0]	计数输入选择。在计数器模式 (CTM[1:0]不是 00) 下, 该位选择 CAP 引脚对时钟进行采样。 00: CT32Bn_CAP0; 其它: 保留。	R/W	0
1:0	CTM[1:0]	计数器/定时器模式。此字段选择 PCLK 的上升沿将 PC 清零并且定时器计数器递增计数。 00: 定时器模式, PCLK 的每个上升沿; 01: 计数器模式, TC 在 CIS 位选择的 CAP0 输入的上升沿递增计数; 10: 计数器模式, TC 在 CIS 位选择的 CAP0 输入的下降沿递增计数; 11: 计数器模式, TC 在 CIS 位选择的 CAP0 输入的上升/下降沿递增计数。	R/W	0

9.7.6 CT32Bn 匹配控制寄存器 (CT32Bn_MCTRL) (n=0,1,2)

地址偏移量: 0x14

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11	MR3STOP	停止 MR3: 若 MR3 和 TC 匹配, TC 和 PC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
10	MR3RST	MR3 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
9	MR3IE	MR3 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
8	MR2STOP	停止 MR2: 若 MR2 和 TC 匹配, TC 和 PC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
7	MR2RST	MR2 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
6	MR2IE	MR2 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
5	MR1STOP	停止 MR1: 若 MR1 和 TC 匹配, TC 和 PC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
4	MR1RST	MR1 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
3	MR1IE	MR1 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
2	MR0STOP	停止 MR0: 若 MR0 和 TC 匹配, TC 和 PC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
1	MR0RST	MR0 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
0	MR0IE	MR0 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0

9.7.7 CT32Bn 匹配寄存器 0~3 (CT32Bn_MR0~3) (n=0,1,2)

地址偏移量: 0x18, 0x1C, 0x20, 0x24

匹配寄存器的值不断地与定时器计数器 (TC) 的值进行比较, 当 2 个值相等时, 自动触发动作。该动作可能是产生中断, 复位 TC 或者停止定时器, 通过设置 CT32Bn_MCTRL 寄存器控制。

Bit	Name	Description	Attribute	Reset
31:0	MR[31:0]	TC 匹配值。	R/W	0

9.7.8 CT32Bn 捕获控制寄存器 (CT32Bn_CAPCTRL) (n=0,1,2)

地址偏移量: 0x28

发生捕获事件时，捕获寄存器是否将值装入计时器/定时器，或者是否由捕获事件产生中断，都通过捕获控制寄存器来控制。同时设置上升位和下降位为有效配置，这样导致上升/下降沿发生捕获事件。

* 注: CAP0EN=1 时, HW 直接切换 I/O 配置。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CAP0EN	捕获 0 功能使能位。 0: 禁止; 1: 使能。	R/W	0
2	CAP0IE	CT32Bn_CAP0 事件中断: CT32Bn_CAP0 事件产生中断。 0: 禁止; 1: 使能。	R/W	0
1	CAP0FE	CT32Bn_CAP0 下降沿捕获: 当 CT32Bn_CAP0 检测到输入信号按照先 1 后 0 的顺序变化时, CAP0 保存此时的 TC 数值 0: 禁止; 1: 使能。	R/W	0
0	CAP0RE	CT32Bn_CAP0 上升沿捕获: 当 CT32Bn_CAP0 检测到输入信号按照先 0 后 1 的顺序变化时, CAP0 保存此时的 TC 数值 0: 禁止; 1: 使能。	R/W	0

9.7.9 CT32Bn 捕获 0 寄存器 (CT32Bn_CAP0) (n=0,1,2)

地址偏移量: 0x2C

每个捕获寄存器都与一个设备引脚相对应，当此设备引脚上有指定的事件发生时，计数器/定时器数值便可以保存在对应的捕获寄存器中。捕获控制寄存器的设置决定是否使能捕获功能，以及捕获事件是否在相关联的引脚的上升沿、下降沿/或双边沿发生。

Bit	Name	Description	Attribute	Reset
31:0	CAP0[31:0]	定时器计数器捕获值。	R	0

9.7.10 CT32Bn 外部匹配寄存器 (CT32Bn_EM) (n=0,1,2)

地址偏移量: 0x30

外部匹配寄存器提供外部匹配引脚 CT32Bn_PWMCTRL[3:0]的控制和状态。若匹配输出配置为 PWM 输出，外部匹配寄存器的功能由 PWM 规则决定。

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11:10	EMC3[1:0]	CT32Bn_PWM3 功能决定位。 00: 无功能; 01: CT32Bn_PWM3 引脚为低电平; 10: CT32Bn_PWM3 引脚为高电平; 11: 反转 CT32Bn_PWM3 引脚状态。	R/W	0
9:8	EMC2[1:0]	CT32Bn_PWM2 功能决定位。 00: 无功能; 01: CT32Bn_PWM2 引脚为低电平; 10: CT32Bn_PWM2 引脚为高电平; 11: 反转 CT32Bn_PWM2 引脚状态。	R/W	0
7:6	EMC1[1:0]	CT32Bn_PWM1 功能决定位。 00: 无功能; 01: CT32Bn_PWM1 引脚为低电平; 10: CT32Bn_PWM1 引脚为高电平; 11: 反转 CT32Bn_PWM1 引脚状态。	R/W	0
5:4	EMC0[1:0]	CT32Bn_PWM0 功能决定位。 00: 无功能; 01: CT32Bn_PWM0 引脚为低电平; 10: CT32Bn_PWM0 引脚为高电平; 11: 反转 CT32Bn_PWM0 引脚状态。	R/W	0
3	EM3	TC 和 MR3 相等时, 该位根据 EMC3 位进行动作, 并驱动 CT32B1_PWM3 输出状态。	R/W	0
2	EM2	TC 和 MR2 相等时, 该位根据 EMC2 位进行动作, 并驱动 CT32B1_PWM2 输出状态。	R/W	0
1	EM1	TC 和 MR1 相等时, 该位根据 EMC1 位进行动作, 并驱动 CT32B1_PWM1 输出状态。	R/W	0
0	EM0	TC 和 MR0 相等时, 该位根据 EMC0 位进行动作, 并驱动 CT32B1_PWM0 输出状态。	R/W	0

9.7.11 CT32Bn PWM 控制寄存器 (CT32Bn_PWMCTRL) (n=0,1,2)

地址偏移量: 0x34

PWM 控制寄存器用于配置匹配输出作为 PWM 输出。每个匹配输出都可独立的作为 PWM 输出或者匹配输出执行，其功能由 CT32Bn_EM 寄存器控制。

对于每个定时器来说，最多控制 3 路 PWM 输出，由 CT32Bn_PWMCTRL[3:0]选择输出。额外的匹配寄存器决定 PWM 周期长度。在其它的匹配寄存器发生匹配时，PWM 输出高电平。定时器由设置 PWM 周期长度的匹配寄存器复位。定时器复位为 0 时，将当前配置为 PWM 输出的高电平输出都清零。

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23	PWM3IOEN	CT32Bn_PWM3/GPIO 选择位。 0: CT32Bn_PWM3 引脚作为 GPIO 引脚; 1: CT32Bn_PWM3 引脚作为匹配输出，其输出信号取决于 PWM3EN 位。	R/W	0
22	PWM2IOEN	CT32Bn_PWM2/GPIO 选择位。 0: CT32Bn_PWM2 引脚作为 GPIO 引脚; 1: CT32Bn_PWM2 引脚作为匹配输出，其输出信号取决于 PWM2EN 位。	R/W	0
21	PWM1IOEN	CT32Bn_PWM1/GPIO 选择位。 0: CT32Bn_PWM1 引脚作为 GPIO 引脚; 1: CT32Bn_PWM1 引脚作为匹配输出，其输出信号取决于 PWM1EN 位。	R/W	0
20	PWM0IOEN	CT32Bn_PWM0/GPIO 选择位。 0: CT32Bn_PWM0 引脚作为 GPIO 引脚; 1: CT32Bn_PWM0 引脚作为匹配输出，其输出信号取决于 PWM0EN 位。	R/W	0
19:12	Reserved		R	0
11:10	PWM3MODE[1:0]	PWM3 输出模式。 00: PWM 模式 1; 01: PWM 模式 2; 10: 强制为 0; 11: 强制为 1。	R/W	0
9:8	PWM2MODE[1:0]	PWM3 输出模式。 00: PWM 模式 1; 01: PWM 模式 2; 10: 强制为 0; 11: 强制为 1。	R/W	0
7:6	PWM1MODE[1:0]	PWM3 输出模式。 00: PWM 模式 1; 01: PWM 模式 2; 10: 强制为 0; 11: 强制为 1。	R/W	0
5:4	PWM0MODE[1:0]	PWM3 输出模式。 00: PWM 模式 1; 01: PWM 模式 2; 10: 强制为 0; 11: 强制为 1。	R/W	0
3	PWM3EN	PWM3 使能位。 0: CT32Bn_PWM3 由 EM3 控制; 1: 使能 CT32Bn_PWM3 的 PWM 模式。	R/W	0
2	PWM2EN	PWM2 使能位。 0: CT32Bn_PWM2 由 EM2 控制; 1: 使能 CT32Bn_PWM2 的 PWM 模式。	R/W	0
1	PWM1EN	PWM1 使能位。 0: CT32Bn_PWM1 由 EM1 控制; 1: 使能 CT32Bn_PWM1 的 PWM 模式。	R/W	0
0	PWM0EN	PWM0 使能位。 0: CT32Bn_PWM0 由 EM0 控制; 1: 使能 CT32Bn_PWM0 的 PWM 模式。	R/W	0

9.7.12 CT32Bn 定时器中断源状态寄存器 (CT32Bn_RIS) (n=0,1,2)

地址偏移量: 0x38

该寄存器显示定时器/PWM 中断的原始状态，若 CT32Bn_IE 寄存器的相应位置 1，定时器/PWM 中断发送到中断控制器。

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	CAP0IF	捕获通道 0 的中断标志位。 0: CAP0 无中断; 1: CAP0 发生中断请求。	R	0
3	MR3IF	匹配通道 3 的中断标志位。 0: 匹配通道 3 无中断; 1: 匹配中断 3 发生中断请求。	R	0
2	MR2IF	匹配通道 2 的中断标志位。 0: 匹配通道 2 无中断; 1: 匹配中断 2 发生中断请求。	R	0
1	MR1IF	匹配通道 1 的中断标志位。 0: 匹配通道 1 无中断; 1: 匹配中断 1 发生中断请求。	R	0
0	MR0IF	匹配通道 0 的中断标志位。 0: 匹配通道 0 无中断; 1: 匹配中断 0 发生中断请求。	R	0

9.7.13 CT32Bn 定时器中断清零寄存器 (CT32Bn_IC) (n=0,1,2)

地址偏移量: 0x3C

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	CAP0IC	0: 无影响; 1: CAP0IF 位清零。	W	0
3	MR3IC	0: 无影响; 1: MR3IF 位清零。	W	0
2	MR2IC	0: 无影响; 1: MR2IF 位清零。	W	0
1	MR1IC	0: 无影响; 1: MR1IF 位清零。	W	0
0	MR0IC	0: 无影响; 1: MR0IF 位清零。	W	0

10 看门狗定时器 (WDT)

10.1 概述

若系统进入了错误的工作状态，看门狗可以在合理的时间范围内使单片机复位。使能看门狗时，若用户程序清除看门狗定时器失败，则在预订的时间范围内，看门狗会使系统复位或者产生一个中断。

看门狗由 128 个预分频的分频器和 8 位计数器组成，时钟经由预分频提供给定时器。定时器递减计数，计数器递减到 0x01 时为最小值。因此，看门狗的最小间隔为 $T_{WDT_PCLK} \times 128 \times 1$ ，最大间隔为 $T_{WDT_PCLK} \times 128 \times 256$

看门狗按照下列方法进行操作：

- 1、通过 WDTCLKSEL 寄存器为看门狗选择时钟源；
- 2、通过 APB 时钟预分频寄存器 0 (SYS1_APBPCP0) 的 WDTPRE 位为看门狗设置预分频值；
- 3、在 WDT_TC 寄存器中设置看门狗定时器的恒定重装值；
- 4、使能看门狗，通过 WDT_CFG 寄存器设置看门狗定时器的操作模式；
- 5、通过写入 0x55AA 到 WDT_FEED 寄存器清除看门狗，以防止看门狗计数器溢出引起复位或产生中断

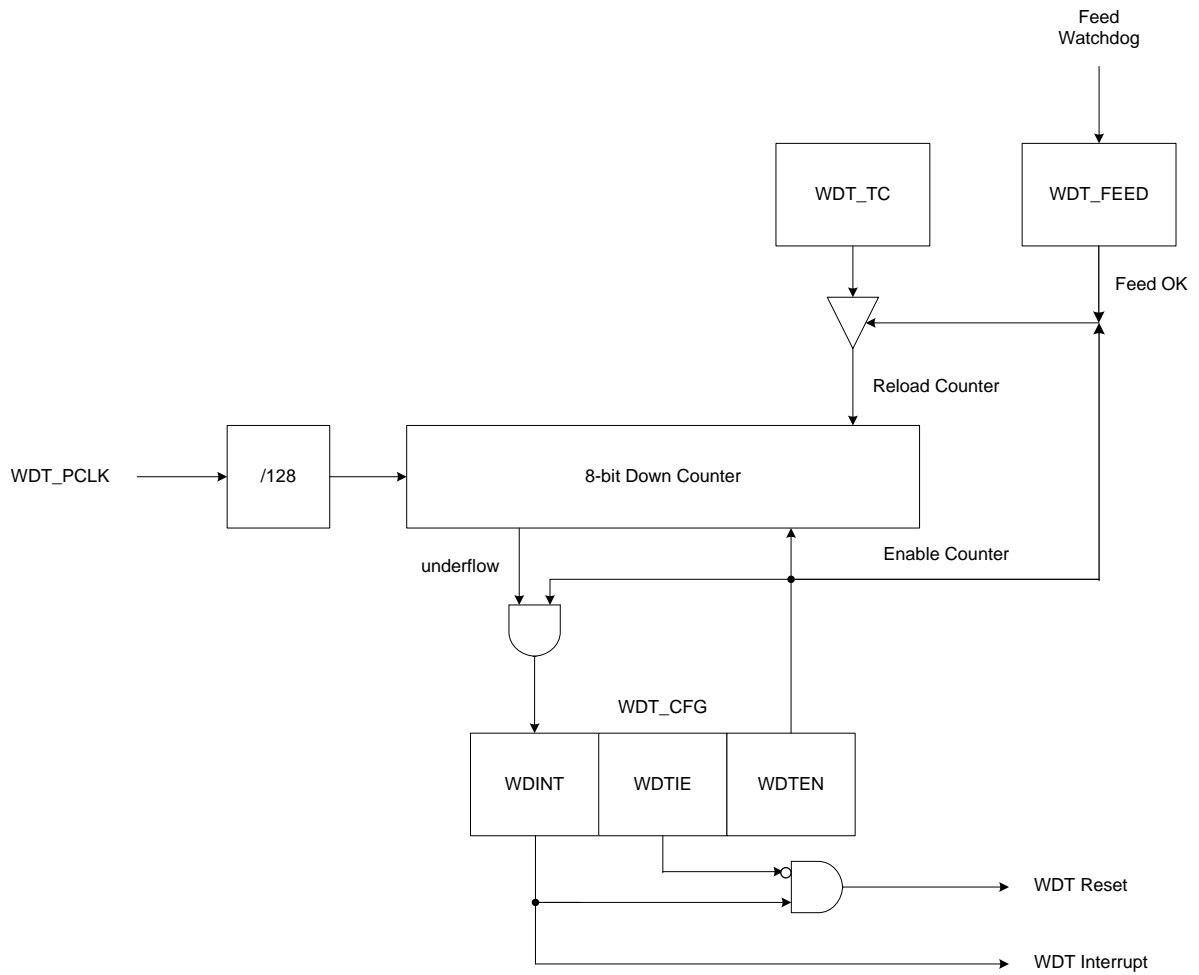
设置 WDT_CFG 寄存器的 WDTCEN 位时，看门狗开始运行，时间恒定值装入看门狗计数器，计数器开始递减计数。看门狗在复位模式下工作时，计数器溢出后，CPU 复位，随着外部复位，系统从向量表中将数据导入堆栈指针和程序计数器中。不管何时写入 0x55AA 到 WDT_FEED 寄存器中，WDT_TC 的值装入看门狗计数器，并阻止看门狗复位或者中断。

看门狗定时器的时钟有两种：HCLK 和 WDT_PCLK。HCLK 用于 AHB 访问看门狗寄存器，并由系统时钟衍生；WDT_PCLK 用于看门狗定时器计数。下列几种时钟都可用作 WDT_PCLK 的时钟源：IHRC，ILRC，ELS X'tal 和 HCLK。

在 AHB 时钟使能寄存器 (SYS1_AHBCLKEN) 中禁止看门狗寄存器模块以省电。

当看门狗正在运行时，任何时候都可能发生看门狗中断或复位

10.2 框图



10.3 WDT 寄存器

基地址：0x4001 0000

10.3.1 看门狗配置寄存器 (WDT_CFG)

地址偏移量：0x00

WDT_CFG 寄存器通过 WDTEN 和 WDTIE 位控制看门狗的操作。该寄存器显示看门狗定时器中断的 Raw 状态。若 WDTINT 和 WDTIE 位都置 1，则 WDT 中断发送到中断控制器中。

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	看门狗寄存器按键。 读取为 0。执行写动作时必须写入 0x5AFA 到寄存器的 WDKEY 位，否则执行的写动作会被忽略。	W	0
15:3	Reserved		R	0
2	WDTINT	看门狗中断标志位。 0: Read→看门狗没有引起中断； Writer→将该位清零，清零之前 SW 会重新设置看门狗。 1: 看门狗超时，引起中断（当且仅当 WDTIE=1）。	R/W	0
1	WDTIE	看门狗中断使能位。 0: 看门狗超时，引起芯片复位（看门狗复位模式）；看门狗计数器下溢引起单片机复位，并将 WDTINT 位清零； 1: 看门狗超时引起中断（看门狗中断模式）。	R/W	0
0	WDTEN	看门狗使能位。 0: 禁止； 1: 使能，使能看门狗时，WDT_TC 的值装入看门狗计数器。	R/W	0

10.3.2 看门狗时钟源寄存器 (WDT_CLKSOURCE)

地址偏移量：0x04

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	看门狗寄存器按键。 读取为 0。执行写动作时必须写入 0x5AFA 到寄存器的 WDKEY 位，否则执行的写动作会被忽略。	W	0
15:2	Reserved		R	0
1:0	CLKSEL[1:0]	选择看门狗时钟源。 00: IHRC； 01: HCLK； 10: ILRC； 11: ELS X'TAL。	R/W	0

10.3.3 看门狗定时器常量寄存器 (WDT_TC)

地址偏移量: 0x08

WDT_TC 寄存器决定超时的时间, 每次清看门狗, 都会重装 WDT_TC 的数值到看门狗定时器中。它是一个 8 位计数器, 定时器的超时时间为 $T_{WDT_PCLK} \times 128 \times 1 \sim T_{WDT_PCLK} \times 128 \times 256$ 。

看门狗的溢出时间 = $(0.02\mu s \times 1) \times 128 \times 1 \sim (0.0625ms \times 32) \times 128 \times 256 = 2.56\mu s \sim 65536ms$

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	看门狗寄存器按键。 读取为 0。执行写动作时必须写入 0x5AFA 到寄存器的 WDKEY 位, 否则执行的写动作会被忽略。	W	0
15:8	Reserved		R	0
7:0	TC[7:0]	看门狗定时器恒定重装值 = TC[7:0]+1 0000 0000: 定时器恒定值 = 1; 0000 0001: 定时器恒定值 = 2; 1111 1110: 定时器恒定值 = 255; 1111 1111: 定时器恒定值 = 256。	R/W	0xFF

10.3.4 看门狗 Feed 寄存器 (WDT_FEED)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	看门狗寄存器按键。 读取为 0。执行写动作时必须写入 0x5AFA 到寄存器的 WDKEY 位, 否则执行的写动作会被忽略。	W	0
15:0	FV[15:0]	Feed 值 (读取为 0) 0x55AA: 清看门狗, WDT_TC 的值重装到看门狗计数器。	W	0

11 实时时钟 (RTC)

11.1 概述

RTC 是一个独立的定时器，它提供的连续计数功能配合软件可实现万年历功能。

通过更改计数器的值可设置系统的时间/日期。

11.2 特性

- 可编程的预分频值：分频因素高达 2^{20} 。
- 32 位可编程计数器，可以进行长时间的测量。
- RTC 时钟源如下所示：
 - EHS X'TAL 时钟的 128 分频；
 - ELS X'TAL；
 - ILRC。
- RTC（预分频值，报警计数器，计数器和分频器）的复位源如下：
 - 冷启动；
 - DPDWAKEUP。
- 3 个专用的使能中断线：
 - 报警中断：产生一个软件可编程的报警中断；
 - 秒值中断：产生一个周期性的中断信号，可编程周期长度（高达 1 秒）；
 - 溢出中断：检测内部可编程计数器的归零状态。

11.3 功能描述

11.3.1 简介

RTC 包含一个 20 位预装值 (RTC SECCNTV)，若在 RTC_IE 寄存器中使能秒值中断，则在每个 TR_CLK 周期，RTC 都会产生一个中断（秒值中断）。秒值模块是一个 32 位可编程计数器，可以设置为当前的系统时间。若在 RTC_IE 寄存器中使能该中断，系统时间在 TR_CLK 比率上递增计数，并可和可编程的数据（存储在 RTC_ALR 寄存器中）进行比较，相同时产生报警中断。

11.3.2 RTC 复位寄存器

RTC_SECCNTV, RTC_ALMCNTV, RTC_SECCNT 和 RTC_ALMCNT 寄存器可通过冷启动或者 DPDWAKEUP 复位进行复位。

11.3.3 RTC 标志说明

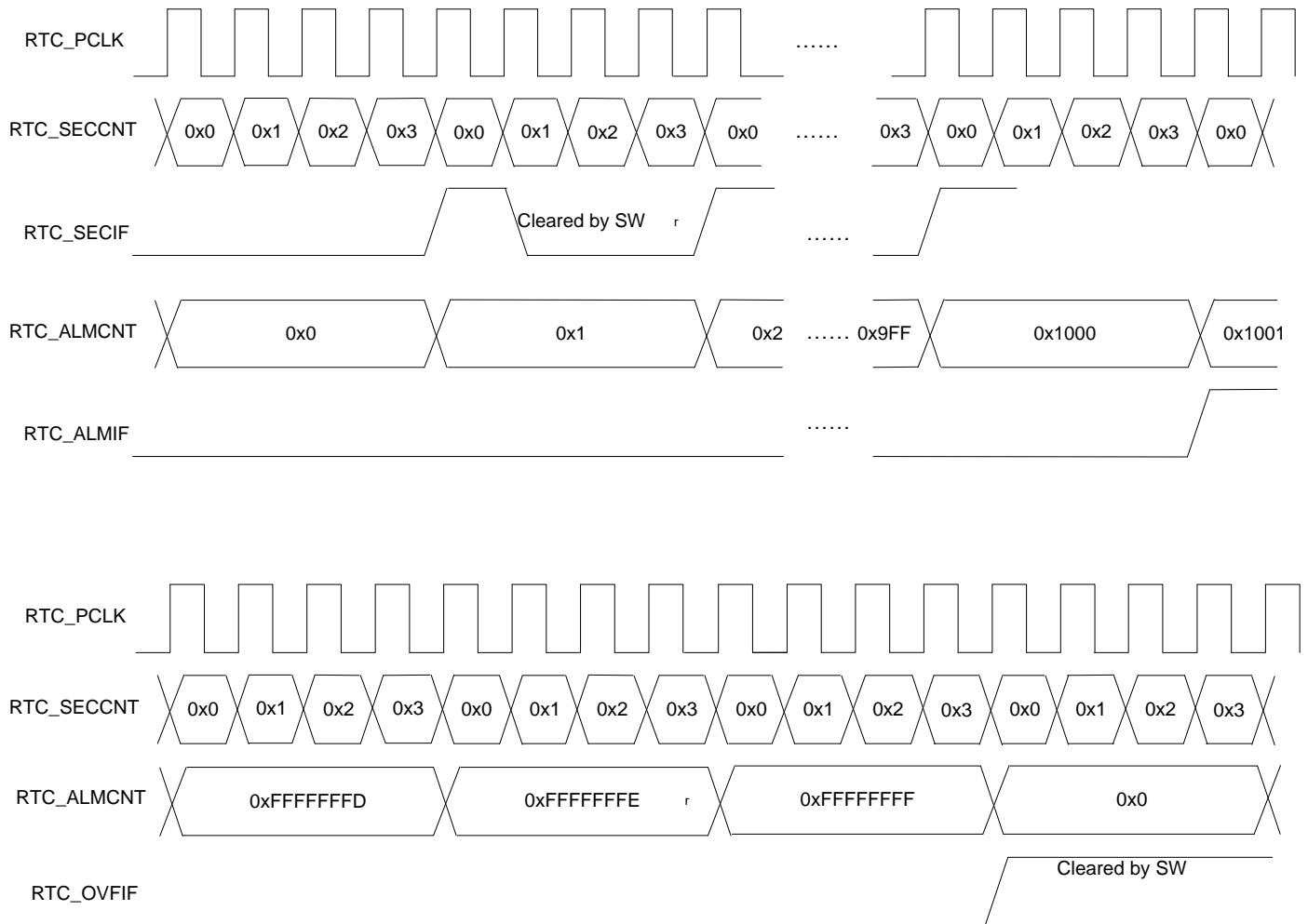
在更新 RTC 计数器之前，每个 RTC 时钟周期会产生 RTC 秒值中断标志 (SECIF)。

在计数器计数到 0 之前，最后的 RTC 时钟周期会产生 RTC 溢出中断标志 (OVFIF)。

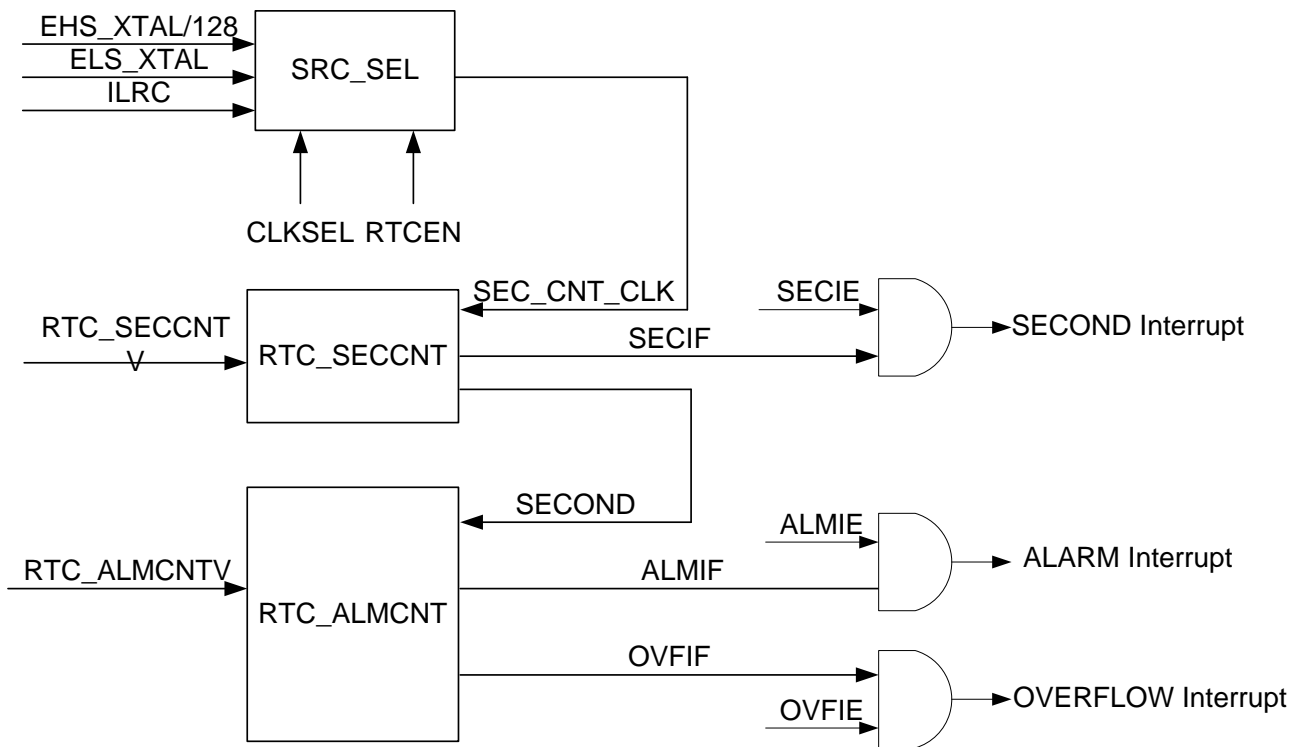
在计数器计数到 RTC 报警计数器重装值（存储于报警寄存器）之前，最后的 RTC 时钟周期会产生 RTC 报警中断标志 (ALMIF)。

11.3.4 RTC 操作

下图显示了 RTC 的波形，RTC_SECCNTV=3，RTC_ALMCNTV=0x1000。



11.4 框图



11.5 RTC 寄存器

基地址：0x4001 2000

11.5.1 RTC 控制寄存器 (RTC_CTRL)

地址偏移量：0x00

* 注：应该在最后设置 RTCEN 位。

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	RTCEN	RTC 使能位。 0: 禁止; 1: 使能, 复位 SEC_CNT 和 ALM_CNT。	R/W	0

11.5.2 RTC 时钟源选择寄存器 (RTC_CLKS)

地址偏移量：0x04

* 注：改变该寄存器的值时，应该由 SW 禁止 RTC (RTCEN=0)。

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1:0	CLKSEL[1:0]	RTC 时钟源选择位。改变值时，HW 复位 SEC_CNT 和 ALM_CNT。 00: ILRC; 01: ELS X'TAL; 10: 保留; 11: EHS X'TAL clock / 128。	R/W	0

11.5.3 RTC 中断使能寄存器 (RTC_IE)

地址偏移量：0x08

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	OVFIE	溢出中断使能位。 0: 禁止; 1: 使能。	R/W	0
1	ALMIE	报警中断使能位。 0: 禁止; 1: 使能。	R/W	0
0	SECIE	秒值中断使能位。 0: 禁止; 1: 使能。	R/W	0

11.5.4 RTC 中断源状态寄存器 (RTC_RIS)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	OVFIF	溢出中断标志位。ALM_CNT 溢出时 (ALM_CNT 计数到 0x00)，由 HW 将该位置 1。若 OVFIF=1，则产生中断。 0: 没有检测到溢出; 1: 32 位可编程计数器溢出。	R	0
1	ALMIF	报警中断标志位。ALM_CNT=ALM_CNTV 时，由 HW 将该位置 1。若 ALRIF=1，则产生中断。 0: 没有检测到报警; 1: 检测到报警。	R	0
0	SECIF	秒值中断标志位。SEC_CNT=SEC_CNTV 时，由 HW 将该位置 1。若 SECIE=1，则产生中断。 0: 没有满足秒值中断的条件; 1: 满足秒值中断的条件。	R	0

11.5.5 RTC 中断清零寄存器 (RTC_IC)

地址偏移量: 0x10

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	OVFIC	0: 没有影响; 1: 清 OVFIF 位。	W	0
1	ALMIC	0: 没有影响; 1: 清 ALMIF 位。	W	0
0	SECIC	0: 没有影响; 1: 清 SECIF 位。	W	0

11.5.6 RTC 秒值计数器重装值寄存器 (RTC_SECNTV)

地址偏移量: 0x14

复位值: 0x8000

Bit	Name	Description	Attribute	Reset
31:20	Reserved		R	0
19:0	SECNTV[19:0]	RTC 秒值计数器重装值。更新该寄存器后，可以复位 RTC_SECNT 和 RTC_ALMCNT 寄存器。建议不要将值设置为 0，由 HW 设置为默认值 0x8000。	R/W	0x8000

11.5.7 RTC 秒值计数寄存器 (RTC_SECNT)

地址偏移量: 0x18

RTC 带有一个 32 位可编程的计数器，该寄存器保持计数器的当前计数值。

Bit	Name	Description	Attribute	Reset
31:0	SECNT[31:0]	RTC 秒值计数器。RTC 计数器的当前值。	R	0

11.5.8 RTC 报警计数器重装值寄存器 (RTC_ALMCNTV)

地址偏移量: 0x1C

复位值: 0xFFFFFFFF

Bit	Name	Description	Attribute	Reset
31:0	ALMCNTV[31:0]	RTC 报警计数器重装值。更新该寄存器后可以复位 ALMCNT 寄存器。建议不要将值设置为 0，由 HW 设置为默认值 0xFFFFFFFF。	R/W	0xFFFFFFFF

11.5.9 RTC 报警计数寄存器 (RTC_ALMCNT)

地址偏移量: 0x20

Bit	Name	Description	Attribute	Reset
31:0	ALMCNT[31:0]	RTC 报警计数器。RTC 报警计数器的当前值。	R	0x0

12 SPI/SSP

12.1 概述

SSP 是同步串行接口控制器，兼容 SPI 和 4 线 SSI 总线，可完成总线上多主机和多从机间的相互通讯。在数据传送过程中，只有单个主机和单个从机可以在总线上相互通讯。数据传送为全双工的 4~16 位的数据帧从主机流向从机，或者从从机流向主机。实际上，在这些数据流中通常只有一个携带着有意义的的数据。

12.2 特性

- 兼容摩托罗拉 SPI，和 4 线 TI SSI 总线。
- 同步串行通讯。
- 支持主机或从机操作。
- 8 帧 FIFO，用于发送或接收。
- 4~16 位帧。
- SSP 模式下，SPI 的最大速度为 25Mbps（主机）或 6Mbps（从机）。
- 数据的发送格式为 MSB 或者 LSB，由寄存器控制。
- 数据采样的起始相位位置可以选择第一个相位或者第二个相位，由寄存器控制。

12.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
SCKn	O	SSP 串行时钟（主机）	
	I	SSP 串行时钟（从机）	取决于 GPIO _n _CFG
SELn	O	SPI 从机选择/SSI 帧 Sync（主机）	
	I	SSP 从机选择（从机）	取决于 GPIO _n _CFG
MISO _n	I	主机输入，从机输出（主机）	取决于 GPIO _n _CFG
	O	主机输入，从机输出（从机）	
MOSI _n	O	主机输出，从机输入（主机）	
	I	主机输出，从机输入（从机）	取决于 GPIO _n _CFG

12.4 接口说明

12.4.1 SPI

SPI 接口是 1 个 4 线接口，SSEL 信号作为一个从机信号进行选择。SPI 格式的主要特性是：空闲状态和 SCK 信号的相位都可通过 SSPn_CTRL1 寄存器的 CPOL 和 CPHA 位编程控制。

CPOL 时钟极性控制位为低时，在 SCK 引脚生成稳定的低电平状态。CPOL 时钟极性控制位为高时，则在 CLK 引脚产生稳定的高电平状态，此时并没有发送数据。

CPHA 时钟相位位控制数据在哪个时钟相位上被采样。若 CPHA=1，SCK 的第一个边沿进行数据转换，SCK 的第二个边沿进行数据的接收和发送；若 CPHA=0，第一个位已经确定，在 SCK 的第一个边沿进行数据的接收和发送。

SIO 数据发送时序图如下：

ML SB	CP OL	CP HA	SCK 空闲状态	Diagrams
0	0	1	Low	
0	1	1	High	
0	0	0	Low	
0	1	0	High	
1	0	1	Low	
1	1	1	High	
1	0	0	Low	
1	1	0	High	

12.4.2 SSI

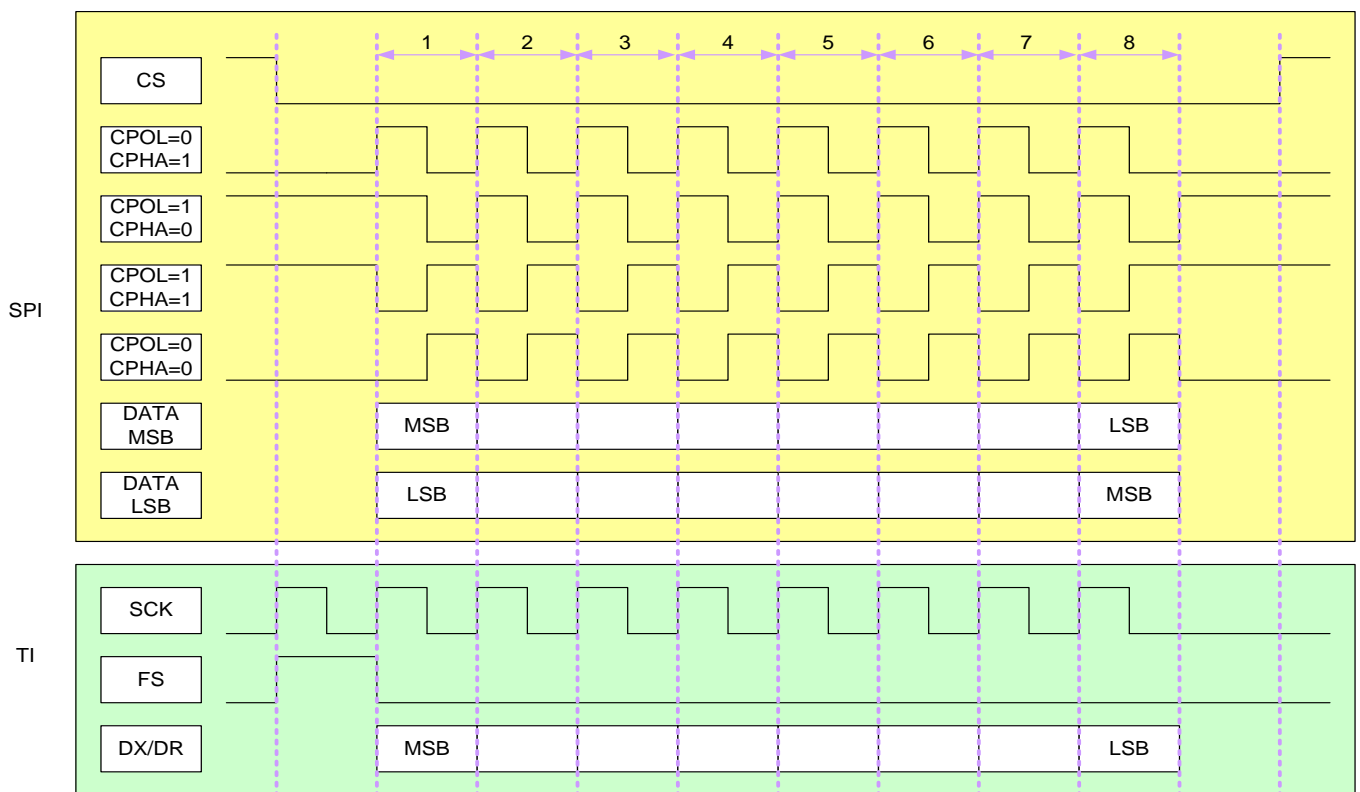
在此模式下，设备可配置为主机，CLK 和 CS 强制为低电平，只要 SSP 为空闲状态，发送数据线 DX 位于 3-state 模式。

一旦发送 FIFO 的底层入口包含数据，FS 拉高一个 CLK 周期。需要发送的值从发送 FIFO 发送到串行移位寄存器中，并在 DX 引脚移出。同样，从机设备输出 MSB 到 DR 引脚上。

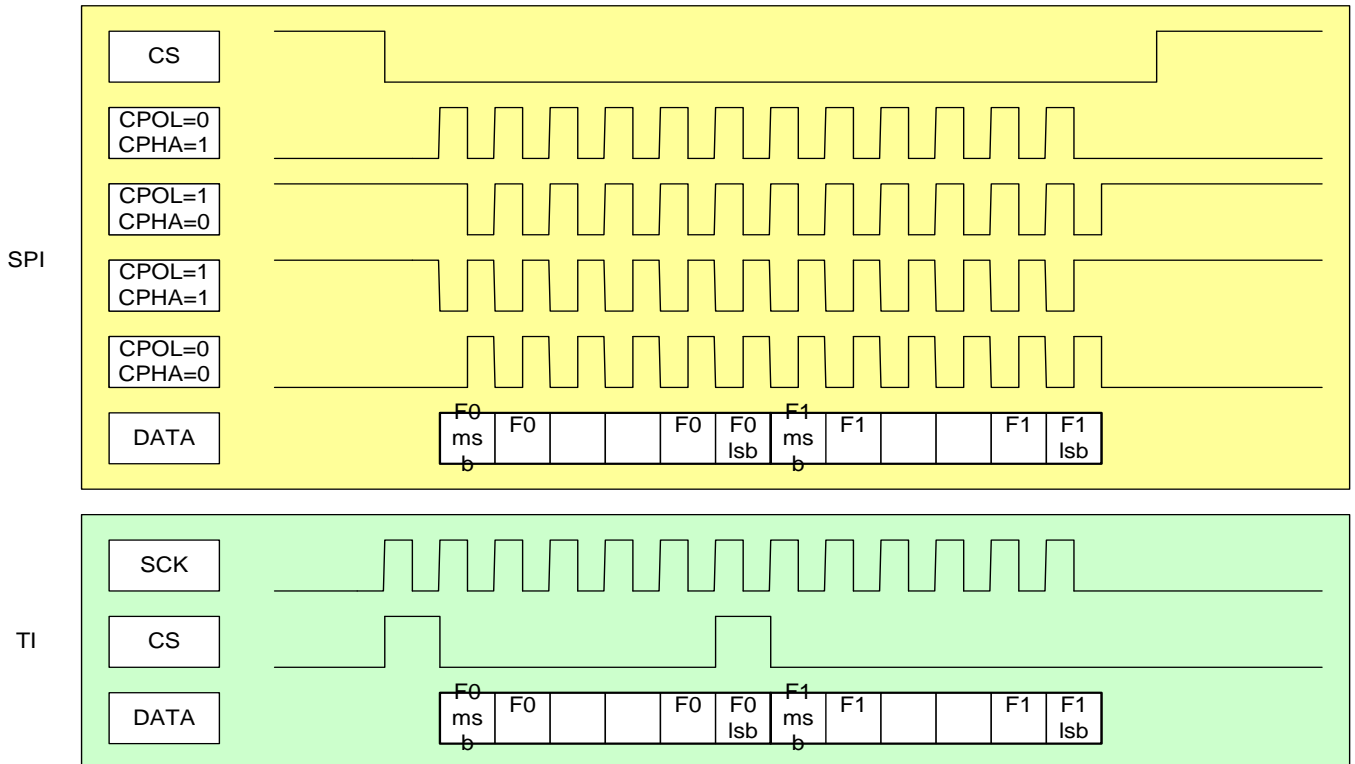
在每个 CLK 的下降沿，SSP 和从机设备都会将每个数据位移入到串行移位器中。当 LSB 被锁存后，在每个 CLK 的上升沿，串行移位器中的接收数据被移入到接收 FIFO 中。

12.4.3 通讯流程

12.4.3.1 单帧



12.4.3.2 多帧



12.5 AUTO-SEL

Auto-SEL 功能的默认状态为禁止状态 (SELDIS = 1)，HW 不能控制 SELn 引脚，SELn 引脚为 GPIO 引脚。若使能 Auto-SEL 功能 (SELDIS=0)，SPI HW 控制 SELn 的状态，由 PFFPA_SSP 寄存器指定 SELn 引脚的功能。

12.6 SSP 寄存器

基地址：0x4001 C000 (SSP0)
0x4005 8000 (SSP1)

12.6.1 SSP n 控制寄存器 0 (SSPn_CTRL0) (n=0, 1)

地址偏移量：0x00

* 注：

1. 当 SSPEN=1 时，如果改变 SSP 的任何配置，都必须使用 FRESET[1:0]复位 SSP FSM。
2. 当 SSPEN=1 时，HW 直接切换 FORMAT 位决定的 I/O 配置。

Bit	Name	Description	Attribute	Reset
31:19	Reserved		R	0
18	SELDIS	Auto-SEL 功能禁止位，仅针对 SPI 模式。 0：使能 Auto-SEL 流程控制； 1：禁止 Auto-SEL 流程控制。	R/W	1
17:15	RXFIFOTH[2:0]	RX FIFO 阈电平。 000：RX FIFO 阈电平为 0； 001：RX FIFO 阈电平为 1； ... 111：RX FIFO 阈电平为 7。	R/W	000
14:12	TX FIFOTH[2:0]	TX FIFO 阈电平。 000：TX FIFO 阈电平为 0； 001：TX FIFO 阈电平为 1； ... 111：TX FIFO 阈电平为 7。	R/W	000
11:8	DL[3:0]	数据长度= DL[3:0] + 1。 0000~0001：保留； 0010：数据长度为 3； 1110：数据长度为 15； 1111：数据长度为 16。	R/W	1111b
7:6	FRESET[1:0]	SSP FSM 和 FIFO 复位位。 00：没有影响； 01：保留； 10：保留； 11：复位状态机和 FIFO (BUF_BUSY=0, 移入 BUF 的数据清零, TX_EMPTY=1, TX_FULL=0, RX_EMPTY=1, RX_FULL=0, 和 FIFO 的数据清零)，该位由 HW 自动清零。	W	0
5	Reserved		R	0
4	FORMAT	接口格式。 0：SPI； 1：SSI。	R/W	0
3	MS	主机/从机选择位。 0：主机； 1：从机。	R/W	0
2	SDODIS	从机数据输出禁止位（仅用于从机模式）。 0：使能从机数据输出； 1：禁止从机数据输出 (MISO=0)。	R/W	0
1	LOOPBACK	回路模式使能位。 0：禁止； 1：使能回路模式。	R/W	0
0	SSPEN	SSP 使能位。 0：禁止； 1：使能。	R/W	0

12.6.2 SSP n 控制寄存器 1 (SSPn_CTRL1) (n=0, 1)

地址偏移量: 0x04

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	CPHA	边沿采样的时钟相位。 0: CPOL=0 时在时钟下降沿改变数据, 上升沿锁存, CPOL=1 时, 在时钟上升沿改变数据, 下降沿锁存; 1: CPOL=0 时在时钟上升沿改变数据, 下降沿锁存, CPOL=1 时, 在时钟下降沿改变数据, 上升沿锁存。	R/W	0
1	CPOL	时钟极性选择位。 0: SCK 空闲为低电平; 1: SCK 空闲为高电平。	R/W	0
0	MLSB	MSB/LSB 选择位。 0: 先发送高位; 1: 先发送低位。	R/W	0

12.6.3 SSP n 时钟分频寄存器 (SSPn_CLKDIV) (n=0, 1)

地址偏移量: 0x08

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DIV[7:0]	SSPn 时钟分频。 0: SCK = SSPn_PCLK / 2; 1: SCK = SSPn_PCLK / 4; 2: SCK = SSPn_PCLK / 6; X: SCK = SSPn_PCLK / (2X+2)。	R/W	0

12.6.4 SSP n 状态寄存器 (SSPn_STAT) (n=0, 1)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6	RXFIFOTHF	RX FIFO 临界值标志位。 0: Data in RX FIFO \leq RXFIFOTH; 1: Data in RX FIFO $>$ RXFIFOTH。	R	0
5	TXFIFOTHF	TX FIFO 临界值标志位。 0: Data in TX FIFO $>$ TXFIFOTH; 1: Data in TX FIFO \leq TXFIFOTH。	R	1
4	BUSY	繁忙标志位。 0: SSP 控制器空闲; 1: SSP 控制器正在工作中。	R	0
3	RX_FULL	RX FIFO 全满标志位。 0: RX FIFO 非全满; 1: RX FIFO 全满。	R	0
2	RX_EMPTY	RX FIFO 全空标志位。 0: RX FIFO 非全空; 1: RX FIFO 全空。	R	1
1	TX_FULL	TX FIFO 全满标志位。 0: TX FIFO 非全满; 1: TX FIFO 全满。	R	0
0	TX_EMPTY	TX FIFO 全空标志位。 0: TX FIFO 非全空, 主机模式下, 发送机自动开始发送; 1: TX FIFO 全空。	R	1

12.6.5 SSP n 中断使能寄存器 (SSPn_IE) (n=0, 1)

地址偏移量: 0x10

该寄存器用于控制 SSP 控制器中 4 个中断的使能状态。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	TXFIFOTHIE	TX FIFO 临界值中断使能位。 0: 禁止; 1: 使能。	R/W	0
2	RXFIFOTHIE	RX FIFO 临界值中断使能位。 0: 禁止; 1: 使能。	R/W	0
1	RXTOIE	RX 超时中断使能位。 0: 禁止; 1: 使能。	R/W	0
0	RXOVFIE	RX 溢出中断使能位。 0: 禁止; 1: 使能。	R/W	0

12.6.6 SSP n 中断源状态寄存器 (SSPn_RIS) (n=0, 1)

地址偏移量: 0x14

该寄存器包含每个中断的状态, 不管是否在 SSPn_IE 寄存器中使能中断。

该寄存器显示 SSP 中断源状态。若 SSPn_IE 寄存器的相应位置 1 时, 发送一个 SSP 中断到中断控制器中。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	TXFIFOTHIF	TX FIFO 临界值中断标志位。 0: 没有 TX FIFO 临界值中断; 1: 触发 TX FIFO 临界值中断。	R	0
2	RXFIFOTHIF	RX FIFO 临界值中断标志位。 0: 没有 RX FIFO 临界值中断; 1: 触发 RX FIFO 临界值中断。	R	0
1	RXTOIF	RX 超时中断标志位。当 RX FIFO 不空时, 并且一定时间 (32*SSPn_PCLK) 没有被读出, 则发生 RXTO。这个溢出时间对主机模式和从机模式是一样。 0: 没有发生 RXTO; 1: 发生 RXTO。	R	0
0	RXOVFIF	RX 溢出中断标志位。当 RX FIFO 全满时, 并且另外一帧被完全接收, 则发生 RXOVF。ARM 规格描述当此种情况发生时, 新帧数据覆盖前帧数据。 0: 没有发生 RXOVF; 1: 发生 RXOVF。	R	0

12.6.7 SSP n 中断清零寄存器 (SSPn_IC) (n=0, 1)

地址偏移量: 0x18

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	TXFIFOTHIC	0: 没有影响; 1: 清 TXFIFOTHIC 位。	W	0
2	RXFIFOTHIC	0: 没有影响; 1: 清 RXFIFOTHIC 位。	W	0
1	RXTOIC	0: 没有影响; 1: 清 RXTOIF 位。	W	0
0	RXOVFIC	0: 没有影响; 1: 清 RXOVFIF 位。	W	0

12.6.8 SSP n 数据寄存器 (SSPn_DATA) (n=0, 1)

地址偏移量: 0x1C

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	DATA[15:0]	<p>写</p> <p>当 SSPn_STAT 寄存器 TX_FULL=0 (TX FIFO 非全满), SW 写入需要发送的数据帧到此寄存器。若 TX FIFO 为空, 且 SSP 控制器在总线上也并不繁忙, 传送数据将立刻发送。否则需等待前面的数据已经发送 (接收), 写入的数据才会被发送</p> <p>读</p> <p>当 SSPn_STAT 寄存器的 RX_EMPTY=0 (RX FIFO 非空) 时, SW 从此寄存器中读取数据。SW 读取此寄存器的数据时, SSP 控制器从 RX FIFO 的最近的帧中返回数据。若数据长度小于 16 位, 这些数据向右靠齐, 剩下的高位则填满 0。</p>	R/W	0

12.6.9 SSP n 数据 Fetch 寄存器 (SSPn_DF) (n=0,1)

地址偏移量: 0x20

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	DF	<p>SSP 数据 fetch 控制位。</p> <p>0: 禁止;</p> <p>1: SCKn 频率大于 6MHz 时使能。</p>	R/W	0

13 I2C

13.1 概述

I2C 总线是两线的双向通讯总线：串行时钟线（SCL）和串行数据线（SDA）。每个设备都通过唯一的地址进行识别，可以作为单一接收设备（如 LCD 驱动）或者兼具收发功能的发送设备（如存储器）。根据是否已经初始化数据传送或者分配设备地址，发送设备或接收设备可工作在主机或者从机模式下。I2C 允许多个主机同时存在于总线上，并且兼容 SMBus 2.0。

根据方向位（R/W）的状态，可在 I2C 总线上处理 2 种数据传送。

- 数据由主机发送器发送到从机接收器：
主机发送的第一个字节是从机地址，接下来是数据字节；从机在接收到每个字节后，都返回一个应答信号。
- 数据由从机发送器发送到主机接收器：

主机发送第一个字节（从机地址），从机返回一个应答信号。接下来由从机发送数据字节到主机。主机在所有接收到的数据之后会返回一个应答信号，除了最后一个字节。在接收到最后一个字节之后，主机返回一个非应答信号。主机设备产生所有串行时钟脉冲和 START 以及 STOP 信号。一个发送终止以 STOP 信号为依据，或者以重复 START 信号为依据。因为重复 START 信号是下一个串行发送的前提，I2C 总线是不会释放。

I2C 接口是以字节为单位，有 4 种操作模式：

- 主机发送模式；
- 主机接收模式；
- 从机发送模式；
- 从机接收模式。

13.2 特性

I2C 接口符合整个 I2C 规范，支持独自关闭 I2C 功能，而不影响同一 I2C 总线上的其它设备。

- 标准 I2C-compliant 总线接口可配置为主机或从机。
- I2C 主机特性：
 - 产生时钟；
 - 产生 START 和 STOP 信号
- I2C 从机特性：
 - 可编程控制 I2C 的地址检测；
 - 可选择识别多达 4 个不同的从机地址；
 - STOP 数据位检测。
- 支持不同的通讯速度：
 - 标准速度（高达 100KHz）；
 - 高速速度（高达 400KHz）。
- 仲裁负责管理多个同时发送的主机以避免总线上串行数据损坏。
- 可编程时钟允许调整的 I2C 发送速率。
- 主机和从机之间的数据传输是双向传送。
- 串行时钟同步允许设备使用不同的速率进行通信
- 串行时钟同步是被用于设备挂起和恢复串行传输的一种握手机制
- 监控模式用于监测所有 I2C 总线通讯，忽略从机地址。
- I2C 总线可用于进行测试和诊断。
- 7 位/10 位寻址的生成和检测以及通用调用。

13.3 引脚说明

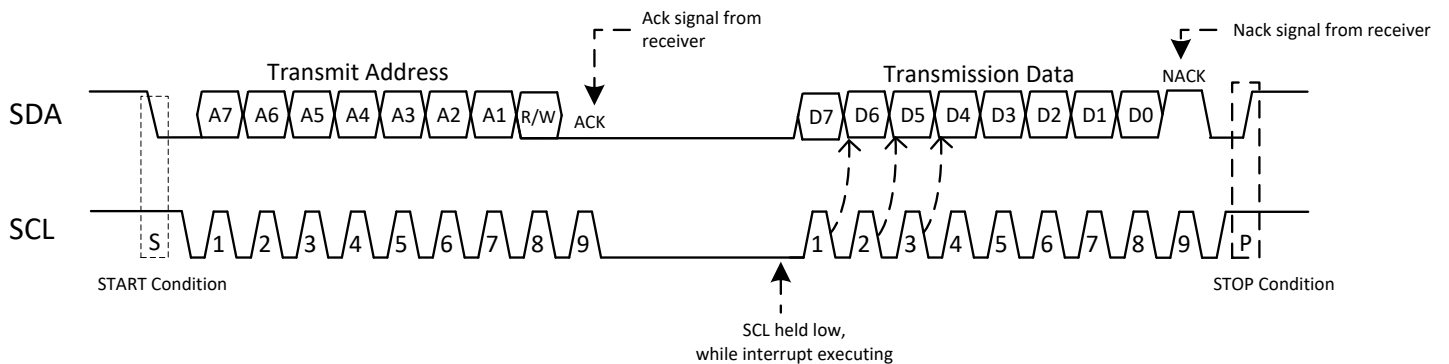
Pin Name	Type	Description	GPIO Configuration
SCLn	I/O	I2C 串行时钟	开漏输出 输入时取决于 GPIO _n _CFG
SDAn	I/O	I2C 串行数据	开漏输出 输入时取决于 GPIO _n _CFG

13.4 I2C 协议

I2C 发送结构包括 START(S)开始信号，8 位地址字节，一个或多个数据字节，以及 STOP (P)结束信号。开始信号由主机产生，以便对传输进行初始化。

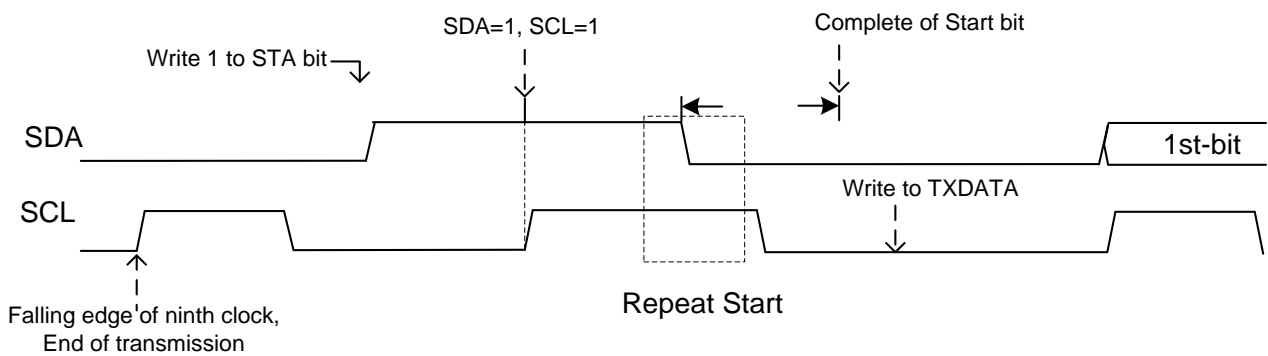
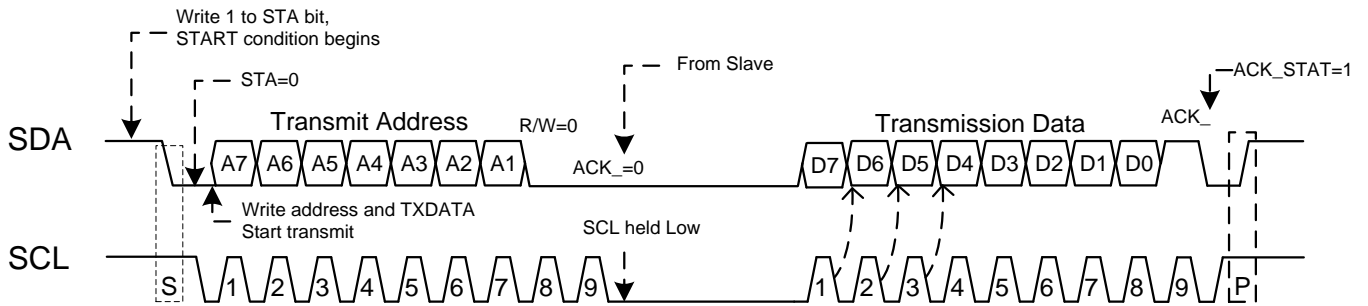
发送的数据为最高有效位 (MSB) 优先。在地址字节中，高 7 位为地址位，低位为数据方向位 (R/W)。R/W=0 时，表示该传输为“WRITE”操作；R/W=1 时，表示该传输为“READ”操作。

接收到每个字节后，接收器（主机或从机）必须发送一个 ACK 信号。若发送器没有接收到 ACK 信号，则会识别为 NACK 信号。在 WRITE 操作中，主机发送数据给从机，然后等待从机返回 ACK 信号。在 READ 操作中，从机发送数据给主机，然后等待主机返回 ACK 信号。最后，主机产生一个 STOP 信号来结束数据传输。

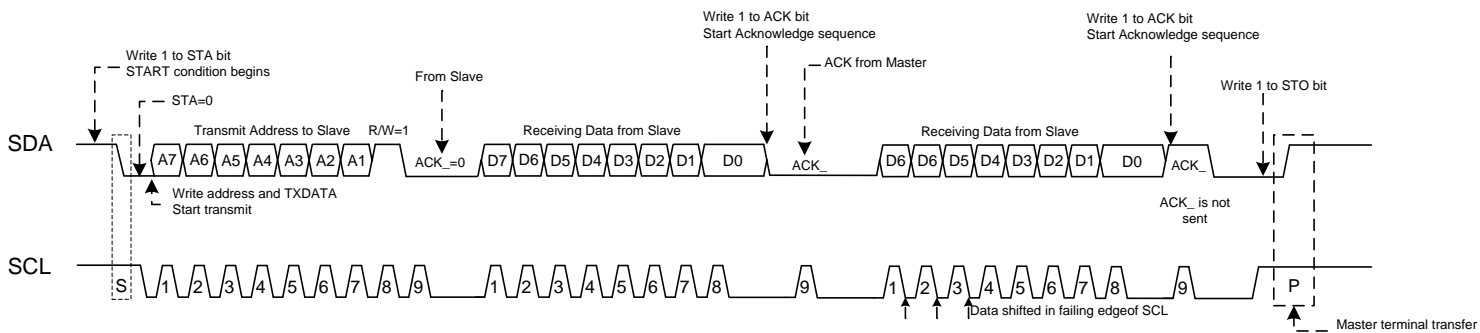


13.4.1 7 位寻址模式

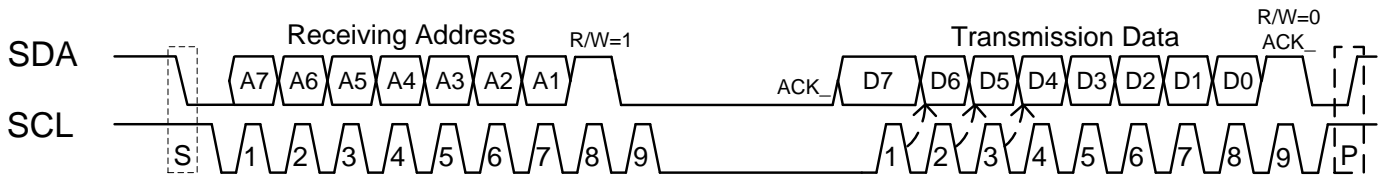
13.4.1.1 主机发送模式



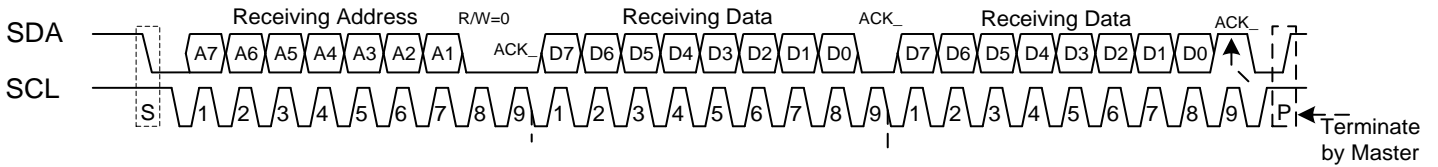
13.4.1.2 主机接收模式



13.4.1.3 从机发送模式

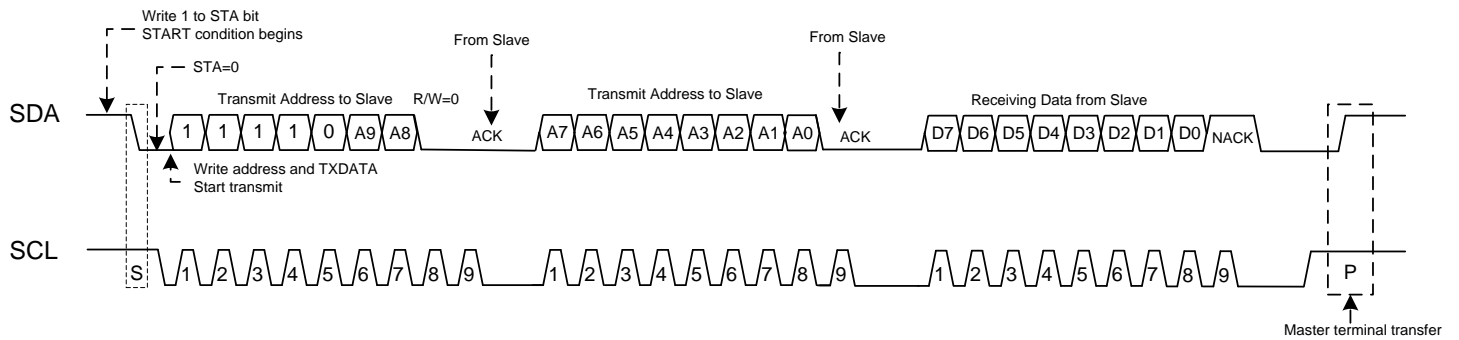


13.4.1.4 从机接收模式

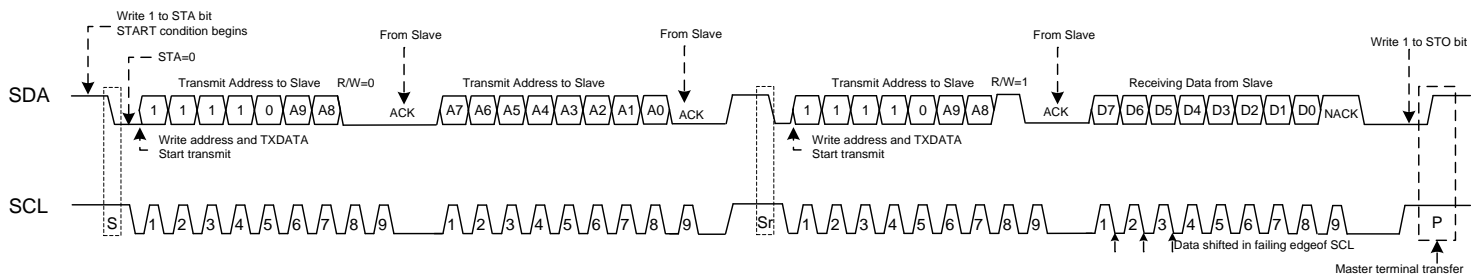


13.4.2 10 位寻址模式

13.4.2.1 主机发送模式



13.4.2.2 主机接收模式



13.5 仲裁

在多主机条件下，同一时间不止一个主机可以在总线上传输数据。需要决定由哪个主机可控制总线以及实现传输。时钟同步与仲裁应用于配置多主机传输。

时钟同步会在和其他设备同步 SCL 信号时运行。当同一时间有两个主机要传输时，时钟同步将会在 SCL 由高电平转变成低电平的时候开始。如果主机 1 先将 SCL 切换为低电平，那么主机 1 会将 SCL 锁定在低电平状态直到将其切换成高电平状态。不过，如果有其他主机的 SCL 时序依然保持在低电平状态的话，那么主机 1 将 SCL 时序由低电平切换成高电平状态的过程将不会改变 SCL 的状态，SCL 时序将仍保持在低电平状态。当所有的主机释放 SCL 时 SCL 线将由低转成到高电平状态。在这其间，主机 1 将保持在由低电平到高电平的等待状态，之后再继续它的传输。

经过时钟同步之后，所有设备的时钟和 SCL 时钟是一样的了。仲裁是用来决定哪个主机能够通过 SDA 信号来完成它的传输。两个主机可能会在同一时间发出开始信号以及传输数据，导致两者会互相影响。仲裁会强制使得其中一个主机失去总线的控制权。数据传输依然会继续，直到两个主机输出了不同的数据信号。如果其中一个主机传输了高电平状态，而另外一个主机传输了低电平状态，SDA 线会被拉低。输出高电平状态的主机将会检测到 SDA 线上的异常，并失去总线的控制权。输出低电平状态的主机成功获得总线的控制权，并继续它的传输，仲裁的过程中不会丢失数据。

仲裁也可能在主机接收模式下丢失。在此模式下，仲裁丢失仅仅发生在 I2C 模块返回一个 NAK 信号时。当总线上的另一个设备将信号拉为低电平时，仲裁丢失。因为只有当在串行字节的末端才会发生这种情况，I2C 模块不再产生时钟脉冲。

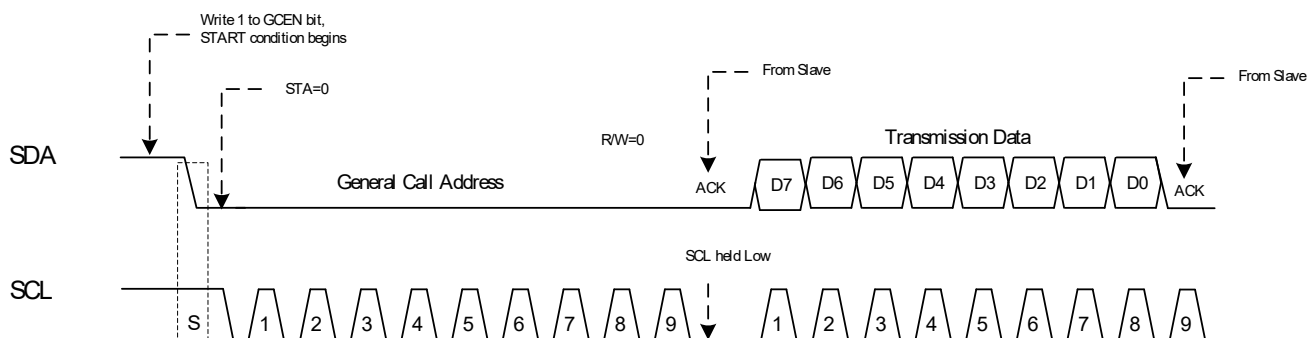
13.6 时钟伸展

时钟伸展通过将 SCL 线拉低来暂停事件。直到 SCL 线再次释放高，事件才能继续进行。时钟伸展是可选的。

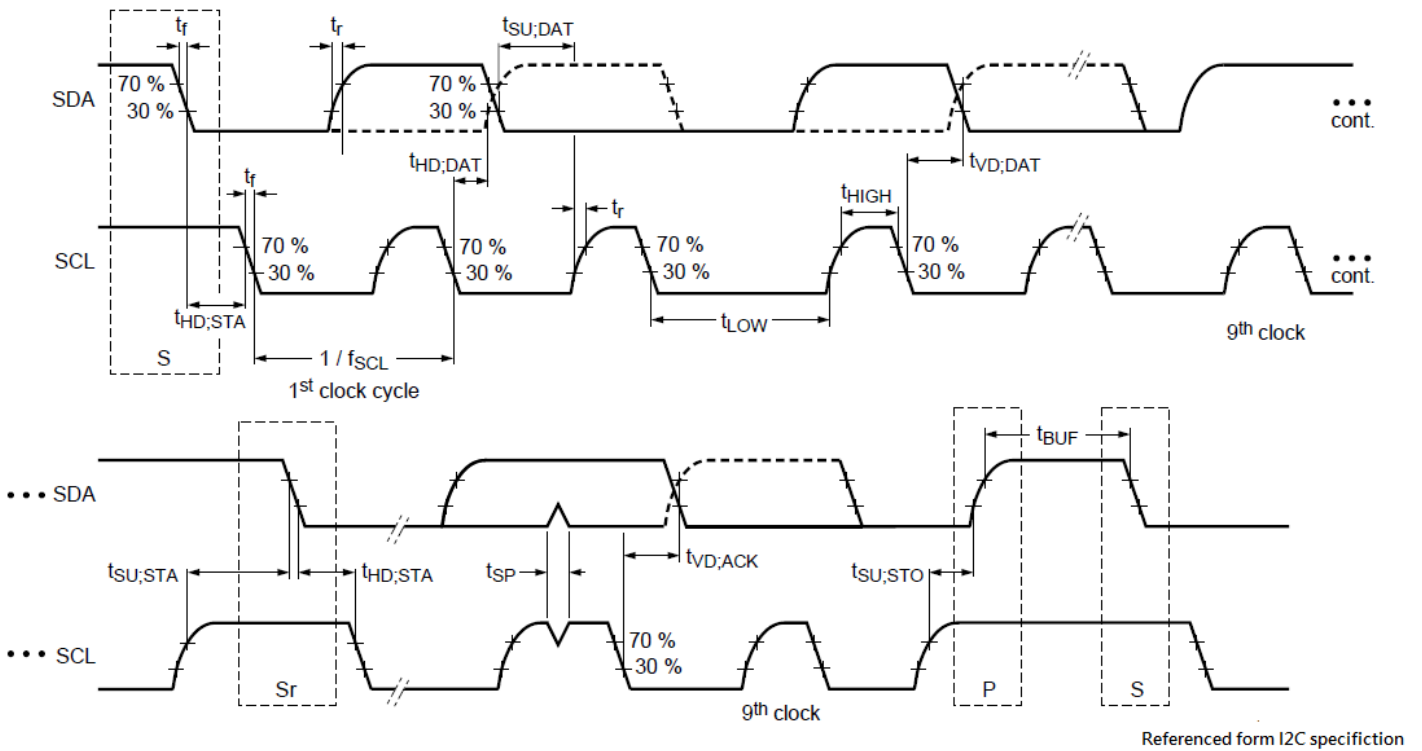
在字节级别上，一个设备可能能够以较快的速度接收数据字节，但需要更多的时间来存储接收的字节或准备另一个要发送的字节。在接收和确认一个字节后，从机可以拉低 SCL 线，迫使主机进入等待状态，直到从机在握手过程中准备好下一个字节传输。

13.7 广播呼叫地址

广播呼叫地址是一个特殊地址，它是由全“0”组成的 7 位地址，用于寻址连接到 I2C 总线的每个设备。但是，如果一个设备不需要在广播呼叫结构中提供任何数据，它可以通过不发出确认信号(ACK)来忽略这个地址。如果一个设备需要来自广播呼叫地址的数据，它需要确认这个地址并作为一个从机接收者。如果一个或多个设备响应，主机实际上并不知道确认了多少设备。第二个和后面的字节被每个有能力处理此数据的从接收器确认。从机如果不能处理其中一个字节数据，必须通过发送 NAK 来忽略它。如果一个或多个从机发出确认 ACK，主机将忽略 NAK 信号。广播呼叫地址的含义在第二个字节中详细指定。



13.8 时序特性



13.8.1 主机发送模式

- t_{HIGH} : $(SCLHT + 1) * I2C_PCLK$ cycle
- t_{LOW} : $(SCLLT + 1) * I2C_PCLK$ cycle
- $t_{HD;STA}$: $(SCLLT + 4) * I2C_PCLK$ cycle
- $t_{HD;DAT}$: $2 * I2C_PCLK$ cycle ~ $3 * I2C_PCLK$ cycle
- $t_{SU;STA}$: $(SCLL + 3) * I2C_PCLK$ cycle
- $t_{SU;STO}$: $(SCLLT + 2) * I2C_PCLK$ cycle ~ $(SCLLT + 3) * I2C_PCLK$ cycle

13.8.2 从机发送模式

- t_{HIGH} : controlled by Master
- t_{LOW} : controlled by Master
- $t_{HD;DAT}$: $2 * I2C_PCLK$ cycle ~ $3 * I2C_PCLK$ cycle
- $t_{SU;DAT}$: $t_{LOW} - 2 * I2C_PCLK$ cycle ~ $t_{LOW} - 3 * I2C_PCLK$ cycle

13.9 监控模式

13.9.1 中断

当模块处于监控模式时，所有的中断都将发生。这意味着当匹配地址被检测到时（若 MATCH_ALL 位置 1，为接收到任意地址，否则为 4 个地址寄存器中的一个匹配地址），发生第一个中断。

检测到的一个匹配地址后，当接收到了每个从机写的的数据字节或者模块认为已经发送出每个从机要读取的字节后，就会有中断请求产生。在第二种情况下，数据寄存器中实际上含有在总线上曾被主机定位过的其它从机所发送的数据。

随着这些中断的发生，处理器可以通过读取数据寄存器来确定总线上实际发送的内容。

13.9.2 仲裁丢失

监控模式下，I2C 模块不能响应主机的信息请求或者发送 ACK 信号，而总线上的其它从机将会应答。这将导致仲裁丢失直到此从机处于匹配状态。通过软件可以知道模块正处于监控模式并且不应该响应仲裁丢失。除此之外，如果仲裁丢失导致中断发生状态与预期不符，可通过硬件设计来阻止部分或所有仲裁丢失发生。硬件是否增加有待确定。

13.10 I2C 寄存器

基地址: 0x4001 8000 (I2C0)

0x4005 A000 (I2C1)

13.10.1 I2C n 控制寄存器 (I2Cn_CTRL) (n=0,1)

地址偏移量: 0x00

I2Cn_CTRL 寄存器用于控制 I2C 接口的操作。

当 STA=1 时, 且 I2C 接口还未在主机模式时, 它进入主机模式, 检测总线的状态, 若总线空闲则产生一个 START 信号。若总线正在繁忙时, 则等待一个 STOP 信号 (释放总线), 在延迟半个内部时钟周期后产生一个 START 信号。如果 I2C 接口已经进入主机模式时, 且已经开始传送和接收数据, 它传送一个重复 START 信号。重复 START 信号可以在任何时候设置, 包括 I2C 接口处于编址的从机模式。

当在主机模式下 STO=1 时, 一个 STOP 信号发送到 I2C 总线上。当总线检测到 STOP 信号时, 自动将 STO 清零。从机模式下, 设置 STO 位可以从错误环境下恢复成正常状态。在这种情况下, 不会有 STOP 信号发送至总线。硬件就好像是已经接收到一个 STOP 信号, 然后切换到从机接收模式。

如果 STA 和 STO 都设置为 1 时, 若接口位于主机模式下, 一个 STOP 信号发送到 I2C 总线上, 然后再发送一个 START 信号。若 I2C 接口位于从机模式下, 产生一个内部 STOP 信号, 但不发送到总线上。

*** 注:**

- 1、应该最后设置 I2CEN 位。
- 2、HW 将分配 SCL0/SCL1 和 SDA0/SDA1 引脚为开漏功能的输出引脚替代 GPIO 引脚。
- 3、接收数据时, ACK 和 NACK 位不能同时设置为 1。
- 4、主机模式下, 用户必须设置 ACK 或 NACK 位为 1, 以进行下一个 RX 处理。

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	I2CEN	I2C 接口使能位。 0: 禁止, STO 位被强制设为 0。 1: 使能, I2CEN 复位时, 总线状态处于丢失状态, 因此 I2CEN 不能用于暂时释放 I2C 总线, ACK 标志位可代替使用。	R/W	0
7: 6	Reserved		R	0
5	STA	START 位。 0: 无 START 信号或者产生重复的 START 信号; 1: 促使 I2C 接口进入主机模式, 发送一个 START 或者重复的 START 信号, 由 HW 自动清零。	R/W	0
4	STO	STOP 标志。 0: STOP 信号空闲; 1: 促使 I2C 接口在主机模式下发送一个 STOP 信号, 或者在从机模式下从错误的环境下恢复到正常的状态。由 HW 自动清零。	R/W	0
3	Reserved		R	0
2	ACK	ACK (SDA 为低电平) 信号标志。 0: 主机模式 → 无功能; 从机模式 → 接收地址或数据后返回一个 NACK; 1: 以下情况时, 在 SCLn 的应答时钟脉冲条件下产生一个应答信号: <ul style="list-style-type: none"> ■ 接收到从地址寄存器的地址; ■ 当 ADR 寄存器的通用调用位置 1 时, 接收到通用调用地址; ■ 当 I2C 处于主机接收模式时, 接收到数据字节; ■ 当 I2C 处于可寻址的从机接收模式时, 接收到数据字节。 硬件在发送完 ACK 后将自动清除此位。	R/W	0
1	NACK	NACK (SDA 为高电平) 信号标志。 0: 无功能。 1: 以下情况时, 在 SCL 的应答时钟脉冲条件下产生一个非应答信号: <ul style="list-style-type: none"> ■ 当 I2C 处于主机接收模式时, 数据字节被接收。 硬件在发送完 NACK 信号后将自动清除此位。	R/W	0
0	Reserved		R	0

13.10.2 I2C n 状态寄存器 (I2Cn_STAT) (n=0,1)

地址偏移量: 0x04

发生 I2C 中断时, 检查该寄存器, 通过写 I2Cn_CTRL 或 I2Cn_TXDATA 寄存器, 所有的状态都自动清零。

I2CIF=1 时, SCL 串行时钟的低电平周期被延长, 串行发送则处于挂起状态。当 SCL 为高电平时, 则不受 I2CIF 状态的影响。

在 NVIC 中断控制器中使能 I2C 中断时, 下拉时间可以触发 I2C 中断:

- START/Repeat START condition
- STOP condition
- 定时器溢出
- 发送或接收数据字节
- 发送或接收 ACK
- 发送或接收 NACK

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15	I2CIF	I2C 中断标志位。 0: I2C 的状态未改变; 1: 读 → I2C 的状态改变, 写 → 将该标志位清零。	R/W	0
14:10	Reserved		R	0
9	TIMEOUT	超时状态位。 0: 未超时; 1: 超时。	R	0
8	LOST_ARB	仲裁丢失位。 0: 未仲裁丢失; 1: 仲裁丢失。	R	0
7	SLV_TX_HIT	0: 从机地址不匹配; 1: 从机地址匹配, 从机模式下为 TX 模式。	R	0
6	SLV_RX_HIT	0: 从机地址不匹配; 1: 从机地址匹配, 从机模式下为 RX 模式。	R	0
5	MST	主机/从机状态位。 0: I2C 处于从机状态; 1: I2C 处于主机状态。	R	0
4	START_DN	Start 处理状态位。 0: 无 START 信号; 1: 主机模式 → 发送一个 START 信号, 从机模式 → 接收到一个 START 信号。	R	0
3	STOP_DN	Stop 处理状态位。 0: 无 STOP 位; 1: 主机模式 → 发送一个 STOP 信号, 从机模式 → 接收到一个 STOP 信号。	R	0
2	NACK_STAT	NACK 处理状态位。 0: 没有接收到 NACK; 1: 接收到 NACK。	R	0
1	ACK_STAT	ACK 处理状态位。 0: 没有接收到 ACK; 1: 接收到 ACK。	R	0
0	RX_DN	RX 处理状态位。 0: 没有发送 RX ACK/NACK; 1: 已经发送 8 位 RX ACK/NACK。	R	0

13.10.3 I2C n TX 数据寄存器 (I2Cn_TXDATA) (n=0,1)

地址偏移量: 0x08

该寄存器包含发送的数据。

主机 TX 模式下, CPU 写入该寄存器可触发 TX 功能, 从机 TX 模式下, CPU 必须在下一次 TX 前写入该寄存器。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DATA[7:0]	发送的数据。	R/W	0x00

13.10.4 I2C n RX 数据寄存器 (I2Cn_RXDATA) (n=0,1)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DATA[7:0]	包含接收的数据, RX_DN=1 时读该寄存器。	R	0x00

13.10.5 I2C n 从机地址 0 寄存器 (I2Cn_SLVADDR0) (n=0,1)

地址偏移量: 0x10

仅用于从机模式下。主机模式下, 该寄存器没有影响。

若该寄存器包含 0x00, I2C 不会在总线上应答任何地址。复位时, 寄存器 ADR0~ADR3 被清零, 此时处于禁止状态。

Bit	Name	Description	Attribute	Reset
31	ADD_MODE	从机地址模式。 0: 7 位地址模式; 1: 10 位地址模式。	RW	0
30	GCEN	通用调用地址使能位。 0: 禁止; 1: 使能通用调用地址 (0x00)。	RW	0
29:10	Reserved		R	0
9:0	ADDR[9:0]	I2C 从机地址。 ADD_MODE=1 时, ADD[9:0]有效; ADD_MODE=0 时, ADD[7:1]有效。	R/W	0

13.10.6 I2C n 从机地址 1~3 寄存器 (I2Cn_SLVADDR1~3) (n=0,1)

地址偏移量: 0x14, 0x18, 0x1C

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	ADDR[9:0]	I2C 从机地址。 ADD_MODE=1 时, ADD[9:0]有效; ADD_MODE=0 时, ADD[7:1]有效。	R/W	0

13.10.7 I2C n SCL 高电平时间寄存器 (I2Cn_SCLHT) (n=0,1)

地址偏移量: 0x20

* 注: I2C 位频率 = $I2Cn_PCLK / (I2Cn_SCLHT + I2Cn_SCLLT)$ 。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	SCLH[7:0]	计算 SCL 高电平周期时间。 SCL 高电平周期时间 = $(SCLH+1) * I2C0_PCLK$ 周期	R/W	0x04

13.10.8 I2C n SCL 低电平时间寄存器 (I2Cn_SCLLT) (n=0,1)

地址偏移量: 0x24

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	SCLL[7:0]	计算 SCL 低电平周期时间。 SCL 低电平周期时间 = $(SCLL+1) * I2C0_PCLK$ 周期	R/W	0x04

13.10.9 I2C n 超时控制寄存器 (I2Cn_TOCTRL) (n=0,1)

地址偏移量: 0x2C

当主机/从机 SCL 保持低电平超过以下时间时, 会发生超时现象:

$$TO * 32 * I2C0_PCLK \text{ 周期}$$

发生 I2C 超时中断时, I2C 发送会返回到“IDLE”状态, 并发出一个 TO 中断以通知用户。这就意味着超时后由 HW 释放 SCL/SDA。主机模式下, 用户可以在超时中断后发出一个 STOP 信号。

通过写 I2Cn_CTRL 或 I2Cn_TXDATA 寄存器可以自动清除超时状态。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	TO[15:0]	计数检测超时。 0: 禁止超时检测; 1: 超时周期时间 = $N * I2Cn_PCLK$ 周期。	R/W	0x0

13.10.10 I2C n 监控模式控制寄存器 (I2Cn_MMCTRL) (n=0,1)

地址偏移量: 0x30

该寄存器控制监控模式，允许 I2C 模块去监控 I2C 总线的通讯状态，没有真正参与通讯或者干扰 I2C 总线的通讯。

监控模式下, SDA 强制输出高电平以阻止 I2C 模块在 I2C 数据总线上以任何形式(包括 ACK)输出数据。根据 SLKOEN 位的状态, SCL 也强制输出高电平以阻止该模块控制 I2C 时钟线。

* 注: MMEN 位为 0 时, SCLOEN 和 MATCH_ALL 位无效(例如该模块处于非监控模式时)。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
2	MATCH_ALL	匹配地址选择位。 0: 当且仅当地址与 I2Cn_SLVADDR0~3 寄存器的某个值相匹配时产生中断; 1: 若 I2C 在监控模式下, 接收到任何地址时产生中断, 可用于监控总线的通讯状况。	R/W	0
1	SCLOEN	SCL 输出使能位。 0: 强制 SCL 输出高电平; 1: I2C 模块可用作一个从机的外设正常工作, I2C 保持时钟一直为低电平, 直到响应 I2C 中断为止。	R/W	0
0	MMEN	监控模式使能位。 0: 禁止; 1: 使能。	R/W	0

14 通用同步异步串行收发器 (USART)

14.1 概述

USART 提供了一个灵活的方式用于和符合 NRZ 异步串行数据格式行业标准的外部设备进行全双工数据交换。串行接口用于低速数据发送或者与低速外部设备进行通讯。

利用小数波特率发生器，USART 可以提供范围非常宽的波特率。USART 支持同步单向通信和单线通信，同时还支持 LIN（局部互联网），智能卡协议和调制解调器操作（CTS/RTS）。

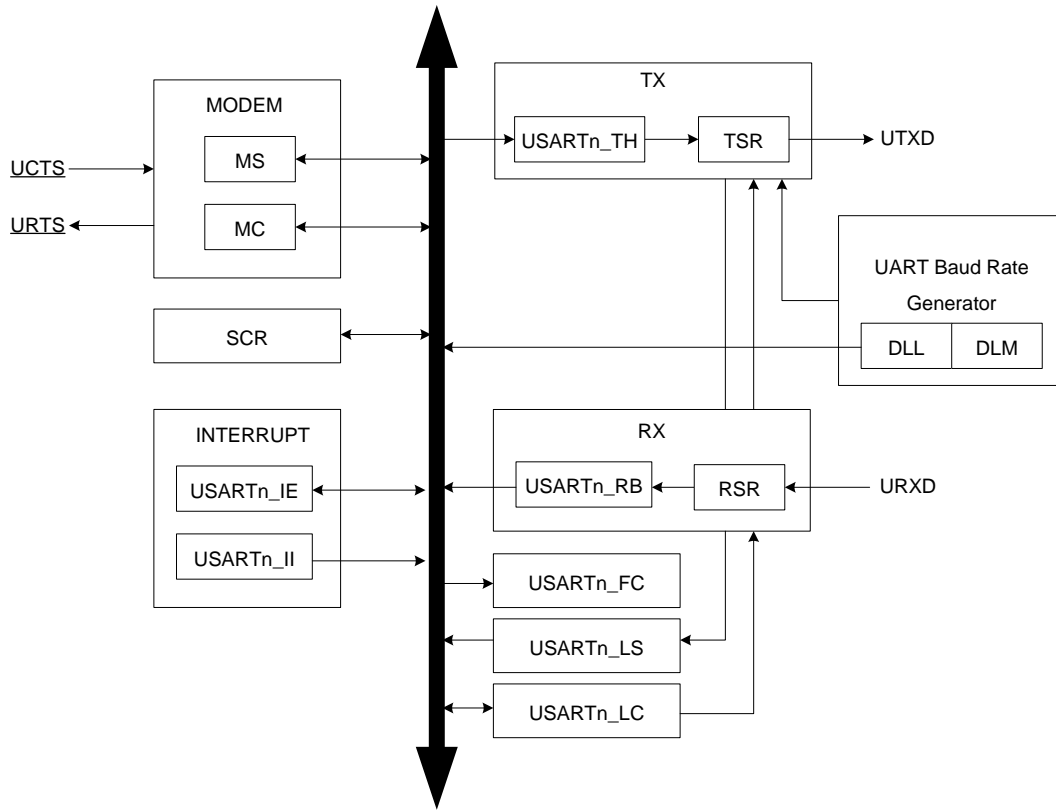
14.2 特性

- 全双工，两线异步数据传输。
- 单线半双工通信
- 同步传输发送器时钟输出。
- 16 字节长度的接收和发送 FIFO。
- 寄存器位置符合 16550 工业标准。
- 接收器 FIFO 触发点可以为 1、4、8、和 14 个字节。
- 内置波特率产生器。
- 包含软件或硬件流程控制机制。
- 支持 EIA-485 9 位模式输出使能。
- 调制解调器控制信号（CTS/RTS）。
- ISO 7816 - 3 兼容的智能卡接口。

14.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
UTXDn	O	串行发送数据引脚	
URXDn	I	串行接收数据引脚	取决于 GPIO _n _CFG
USCKn	O	同步模式（主机）的串行时钟引脚	
UCTSn	I	允许发送引脚。 为低电平时，表示调制解调器或者数据通信设备已准备好交换数据。 CTS 信号是 MODEM 的一种输入信号，可通过读取 USART _n _MS 寄存器的位 4（CTS）来检测其状态。	取决于 GPIO _n _CFG
URTSn	O	请求发送。RS-485 管理控制引脚。 为低电平时，通知调制解调器或数据通信设备 UART 已准备好交换数据。通过设置 USART _n _MC 寄存器的位 1（RTS），将 RTS 的输出信号设置为低电平有效。循环模式则保持该信号为无效状态。	

14.4 框图



14.5 EIA-485/RS-485 模式

RS-485/EIA-485 特性使 USART 可配置为一个可寻址的从机接收器，此可寻址的从机接收器是单主机控制的众多从机接收器中的一个。

USART 主机发送器可以通过设置校验位（9th）为 1 来识别地址符或设置为 0 来识别数据符。

每个 USART 从机接收器可分配一个唯一的地址。从机设备可设为自动或手动丢弃不是它地址的数据。

RS-485 模式下，通过 SW，USARTn_LC 寄存器的 PS 位可被选择强置 1 校验（地址），或强置 0 校验（数据）。此外，通过 SW 设置 USARTn_LC 寄存器的 WLS 位为 11b 将 word 长度设为 8 位。

14.5.1 RS-485/EIA-485 普通多支路模式（NMM）

在 USARTn_RS485CTRL 寄存器中设置 NMMEN 位为 1 时使能该模式。在该模式下，接收到的字节导致 USART 设置奇偶校验位错误时检测到一个导致，并产生一个中断。

若禁止接收器（USARTn_RS485CTRL 寄存器的 RXEN=0）时，将忽略接收到的数据字节，且不会存入 RXFIFO 中。当检测到的一个地址字节（奇偶校验位为 1）将会存入 RXFIFO 中，产生奇偶校验位错误（FE）中断。处理器接下来会读取地址字节并决定是否使能接收器来接收剩下的数据。

使能接收器（USARTn_RS485CTRL 寄存器的 RXEN=1）时，接受接收到的所有字节（不管是地址还是数据）都存入 RXFIFO 中，当接收到的地址字符产生奇偶校验位中断时，处理器会决定是否禁止接收器。

14.5.2 RS-485/EIA-485 自动地址检测模式（AAD）

USARTn_RS485CTRL 寄存器的 NMMEN 位（9 位模式使能位）和 AADEN 位（AAD 模式使能位）都置 1 时，USART 进入自动地址检测模式。

在此模式下，接收器把接收的地址字节与 USARTn_RS485ADRMATCH 寄存器中保存的 8 位数据进行比较。

若禁止接收器（USARTn_RS485CTRL 寄存器的 RXEN=0）时，当接收到的任何数据字节或者地址字节，如果不同于 USARTn_RS485ADRMATCH 寄存器中的值都将被丢弃。

当侦测到匹配地址符时，它将被存入 RXFIFO，包括校验位。接收器将被自动使能（HW 设置 RXEN 为 1），同时接收器产生一个 RX 数据有效中断（RDA）。

使能接收器（USARTn_RS485CTRL 寄存器的 RXEN=1）时，接受接收到的所有字节并存入 RXFIFO 中，直到接收到的地址字节不同于 MATCH 值。此时，HW 自动禁止接收器（RXEN 位由 HW 清零），接收到的不匹配的地址符不会存入 RXFIFO。

14.5.3 RS-485/EIA-485 自动方向控制（ADC）

RS485/EIA-485 模式可以配置发送器自动控制 DIR 引脚的状态，作为方向控制输出信号。设置 USARTn_RS485CTRL 寄存器的 ADCEN 位使能该特性。

除了回路模式，ADCEN 位在管理方向控制引脚上享有最高优先级。

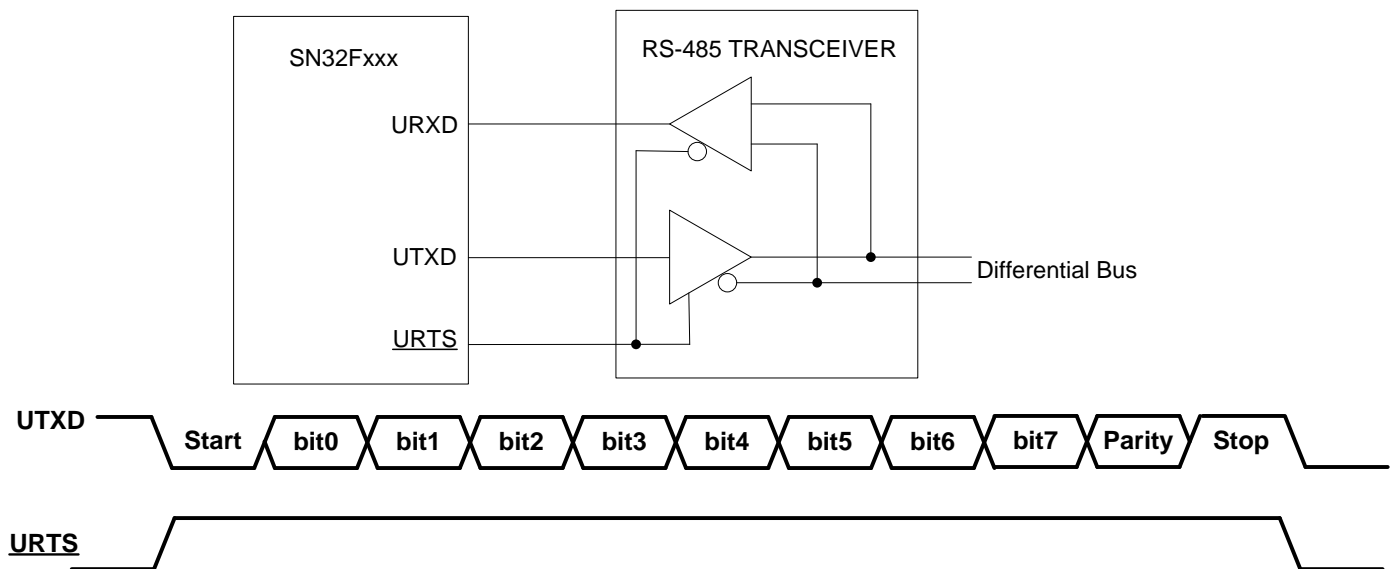
14.5.4 RS485/EIA-485 驱动延迟时间

驱动延迟时间指从 TXFIFO 移出最后一个 STOP 位到 URTS 失效之间的时间。该延迟时间可由 8 位寄存器 USARTn_RS485DLYV 编程设置，其以波特率周期为基准，可设置为 0~255 倍基准时间。

14.5.5 RS485/EIA-485 输出反向

可以通过设置 USARTn_RS485CTRL 寄存器的 OINV 位，使 URTS 引脚控制信号的方向反向。若 OINV 位置 1，发送器中有数据等待发送时，引脚信号强制设为逻辑 1（驱动低）；一旦发送完最后一位数据，该引脚信号强制设为逻辑 0（驱动高）。

14.5.6 RS485/EIA-485 帧结构图



14.6 波特率计算

USART 波特率的计算公式如下：

$$UART_{BAUDRATE} = \frac{USARTn_PCLK}{\text{Oversampling} \times (256 \times DLM + DLL) \times (1 + DIVADDVAL / MULVAL)}$$

USARTn_PCLK 是外设时钟，USARTn_DLM 和 USARTn_DLL 是标准的 UART 波特率除数寄存器，USARTn_FD 寄存器中的 DIVADDVAL 和 MULVAL 是 USART 小数波特率发生器参数。

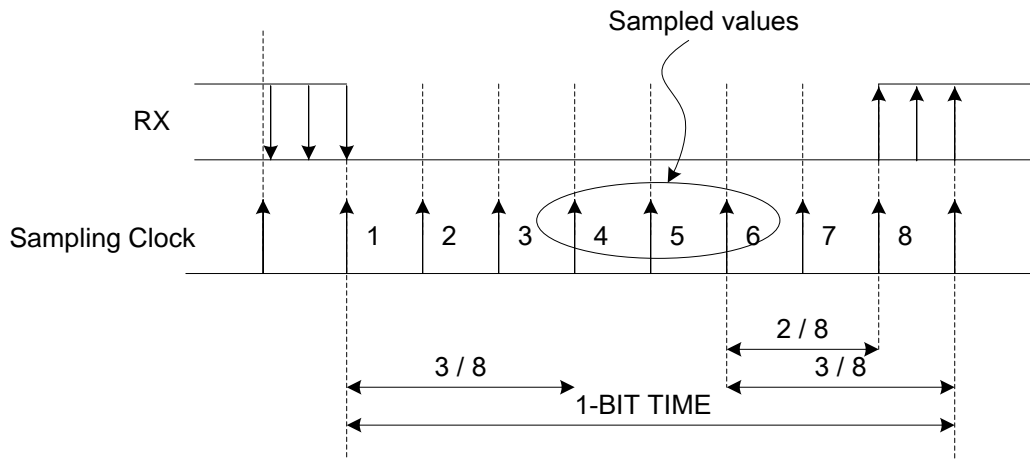
MULVAL 和 DIVADDVAL 的值应当符合下列条件：

- 1、 $1 \leq MULVAL \leq 15$;
- 2、 $0 \leq DIVADDVAL \leq 14$;
- 3、 $DIVADDVAL < MULVAL$;
- 4、重复采样为 8 次或者 16 次。

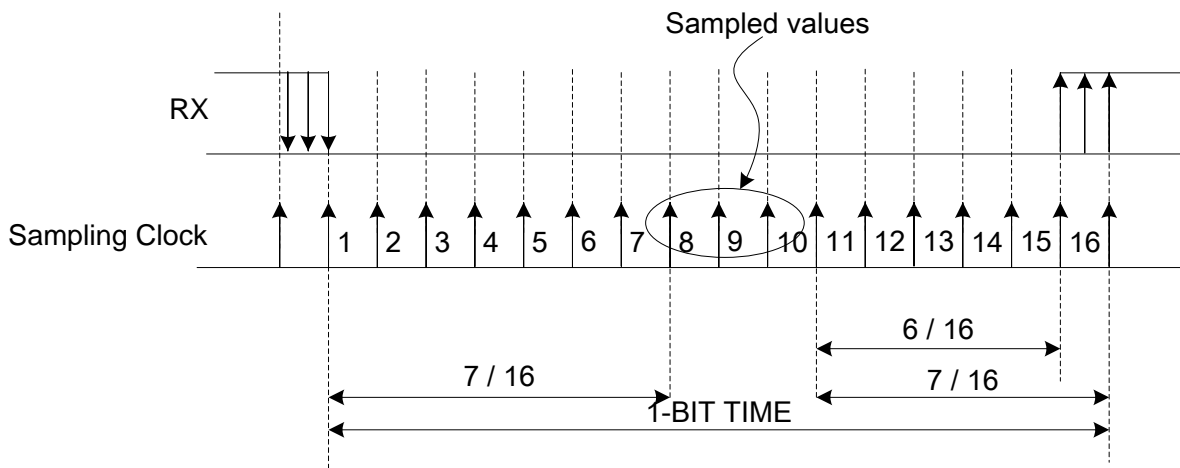
发送/接收数据时，不要修改 USARTn_FD 寄存器的值，否则数据丢失或损坏。

可以通过设置 USARTn_FD 寄存器的 OVER8 位选择重复采样的方式，且采样时钟为波特率时钟的 16 倍或 8 倍。

- OVER8=1: 只重复采样 8 次可以获得更高的速度（高达 USARTn_PCLK/8）。但在这种情况下，会降低接收器对时钟偏差的最大容错性。



- OVER8=0: 重复采样 16 次，可以增加接收器对时钟偏差的容错性。在这种情况下，最高速度限制为 USARTn_PCLK / 16。



如果 USARTn_FD 寄存器值不符合这两个要求，那么小数分频器的输出是不确定的。如果 DIVADDVAL 为零，则禁止小数分频器，时钟不会分频。

无论有无小数分频器，都可操作 USART。通过设置不同的小数分频器，可以获得所需的波特率。下面的算法说明了如何寻找一组合适的 DLM、DLL、MULVAL 和 DIVADDVAL 值，这样可以生成一个波特率，与所需波特率的误差小于 1.1%。

举例说明：选择 DIVADDVAL、MULVAL、DLM 和 DLL 生成 BR=115200，此时 USARTn_PCLK=12MHz，重复采样为 16 次。

$$\text{UART}_{\text{BAUDRATE}} = \frac{\text{USARTn_PCLK}}{\text{Oversampling} \times (256 \times \text{DLM} + \text{DLL}) \times (1 + \text{DIVADDVAL} / \text{MULVAL})}$$

$$115200 = \frac{12000000}{16 \times (256 \times \text{DLM} + \text{DLL}) \times (1 + \text{DIVADDVAL} / \text{MULVAL})}$$

$$(256 \times \text{DLM} + \text{DLL}) \times (1 + \text{DIVADDVAL} / \text{MULVAL}) = 6.51$$

由于 MULVAL 和 DIVADDVAL 的值须符合下列条件：

1. $1 \leq \text{MULVAL} \leq 15$
2. $0 \leq \text{DIVADDVAL} \leq 14$
3. $\text{DIVADDVAL} < \text{MULVAL}$

因此，建议 UART 的设置为：DLM=0，DLL=4，DIVADDVAL=5，MULVAL=8（填 7）。则生成的波特率为 115384，与原来 115200 的相对误差为 0.16%。

14.7 调制解调器控制（MC）

若使能自动 RTS 模式，USART 接收器 FIFO 硬件控制 USART 的 URTS 输出；若使能自动 CTS 模式，USART 的发送器只在 UCTS 引脚为低电平时开始发送。

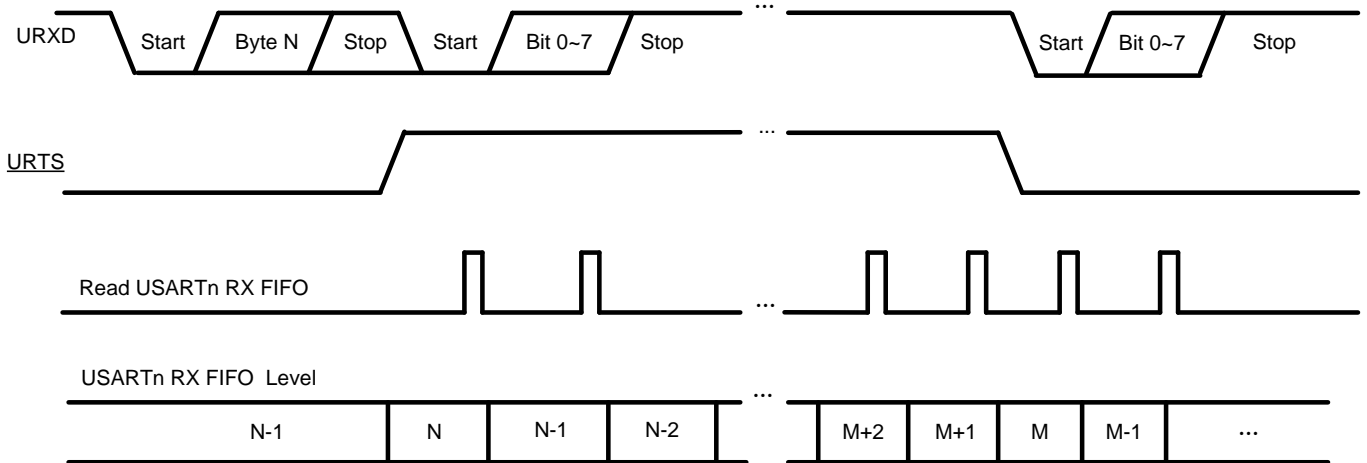
14.7.1 AUTO-RTS

通过设置 RTSEN 位使能自动 RTS 功能，自动 RTS 数据流由 USARTn_RB 模块控制，链接到程控的接收器 FIFO 触发等级。若使能自动 RTS，数据流控制如下：

当接收器 FIFO 等级到达设定的触发等级时，URTS 失效（到一个高电平）。当到达触发等级后，正在发送的 USART 可能会发送一个额外的字节（假定在发送的 USART 有其它的字节需要发送），因为此时它可能尚未识别到 URTS 已失效，直到 USART 已经开始发送额外字节。一旦接收器 FIFO 低于触发等级，URTS 自动恢复（到一个低电平）。自动恢复的 URTS 信号通知 USART 继续发送数据。

若禁止自动 RTS 模式，RTSEN 位控制 USART 的 URTS 输出。若使能自动 RTS 模式，硬件控制 RTS 输出，URTS 的实际值复制到 USART 的 URTS 控制位。只要使能自动 RTS，RTS 控制位的值是通过软件只读的。

示例：假设 16550 模式下的 USART 操作将 USARTn_FIFCTRL 寄存器的触发等级设为 0x2，这样，若使能自动 RTS，一旦接收 FIFO 包含 8 字节，URTS 输出失效。一旦接收 FIFO 低于触发等级：4 字节，URTS 输出自动恢复。

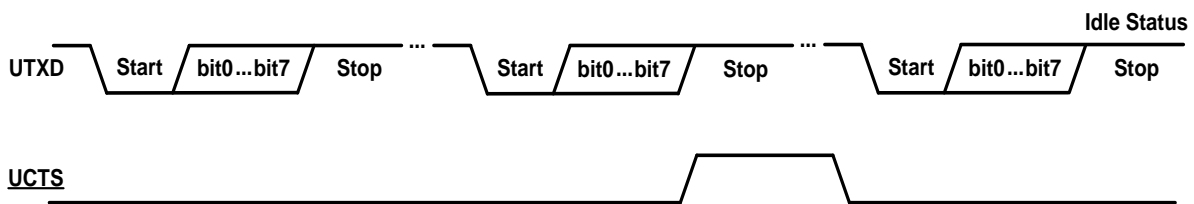


14.7.2 AUTO-CTS

CTSEN=1 时使能自动 CTS 功能。若使能自动 CTS，发送器电路会在发送下一个数据字节前检查 UCTS 输入，UCTS 有效（低电平）时，发送器发送下一个字节。若要发送器停止发送后面的字节，必须在发送最后一个停止位的中间部分之前释放 UCTS。自动 CTS 模式下，UCTS 信号的变化不会触发调制解调器状态中断，除非 CTS 中断使能位置 1，但 USARTn_MS 寄存器的 DCTS 位要置 1。

MSIE	CTSEN	Delta CTS (DCTS)	MODEM status interrupt
0	X	X	X
1	0	1	0
1	1	0	X
1	1	1	0

自动 CTS 功能可以消除 CTS 中断需求。使能数据流控制时，UCTS 状态改变不会触发主机中断，这是因为该设备自动控制发送器。不使用自动 CTS，发送器将发送 FIFO 中的任何数据，可能导致接收器溢出错误。



在传送第二个符号的过程中取消 UCTS 信号，其后不会再发送第三个符号。只要 UCTS（高）失效，USART 在 UTXD 上保持为 1。一旦 UCTS 恢复，数据传送也恢复，并且发送一个开始位，然后跟着下一个符号的数据位。

14.8 自动波特流 (AUTO-BAUD FLOW)

14.8.1 AUTO-BAUD

基于“AT”协议（海耶斯命令），USART 自动波特功能可用于测量输入的波特率。若使能自动波特性能，将测量接收数据流的位时间，相应地设置除数锁存寄存器 USARTn_DLM 和 USARTn_DLL。

通过设置 USARTn_ABCTRL 寄存器的 START 位启动自动波特率功能，通过清 START 位停止该功能。一旦完成自动波特率功能，或读取该位返回到自动波特率状态（等待/完成），START 位就清零。开始自动波特率功能后，清 FIFO，写 TX FIFO 无效，发送器停止发送，直到完成自动波特率功能或者停止自动波特率功能。

可通过 USARTn_ABCTRL 寄存器的 MODE 位选择两种自动波特率测量模式。MODE0 模式下，波特率在 USART RX 引脚的两个连续的下降沿上（START 位的下降沿和 LSB 的下降沿）测量；MODE1 模式下，波特率在 USART RX 引脚的下降沿和随后的上升沿之间测量（START 位的长度）。

有超时现象发生时（速率测量计数器溢出），USARTn_ABCTRL 寄存器的 AUTORESTART 位可以自动重新测量波特率。若该位置 1，则在 URXD 引脚的下一个下降沿重新开始测量波特率。

自动波特率功能可以产生两个中断。

- 若使能中断（USARTn_IE 寄存器的 ABTOIE 位置 1，自动波特率测量计数器溢出），会设置 USARTn_I1 寄存器的 ABTOINT 中断；
- 若使能中断（USARTn_IE 寄存器的 ABTOIE 位置 1，自动波特率已经测量完成），会设置 USARTn_I1 寄存器的 ABEOINT 中断。

必须通过设置 USARTn_IE 寄存器的相应 ABTOINTCLR 和 ABEOIE 位来将自动波特中断清零。

在自动波特率期间，必须禁止小数波特率发生器，当自动波特率使用时，任何写 USARTn_DLM 和 USARTn_DLL 寄存器的操作必须在写 USARTn_ABCTRL 寄存器之前执行。USART 支持的最大和最小波特率是受 USARTn_PCLK 和数据个位，停止位，校验位的影响。

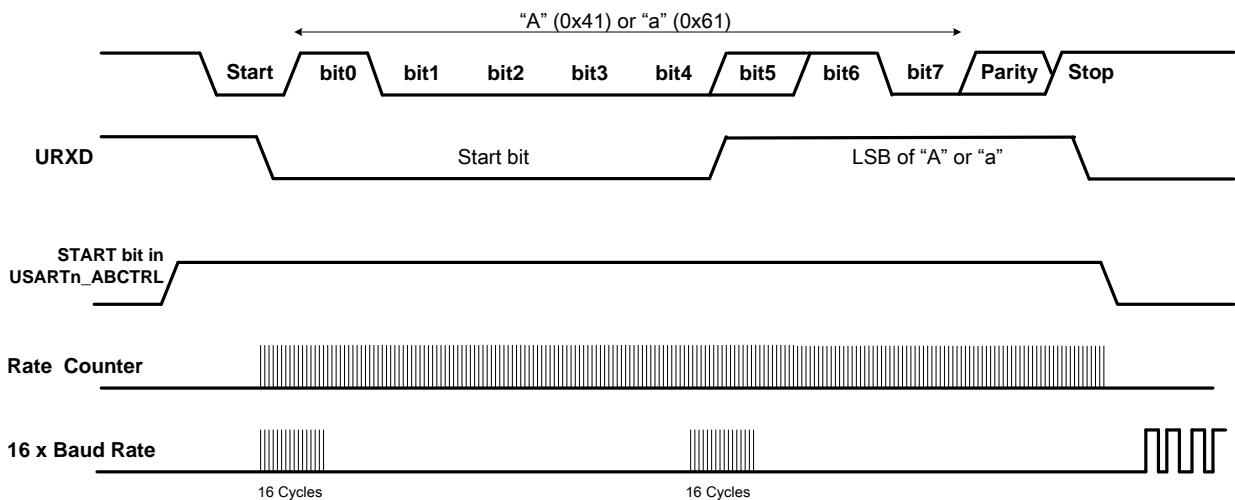
$$ratemin = \frac{2 \times PCLK}{16 \times 2^{15}} \leq U_{ART} \text{ baudrate} \leq \frac{PCLK}{16 \times (2 + \text{databits} + \text{paritybits} + \text{stopbits})} = ratemax$$

14.8.2 AUTO-BAUD 模式

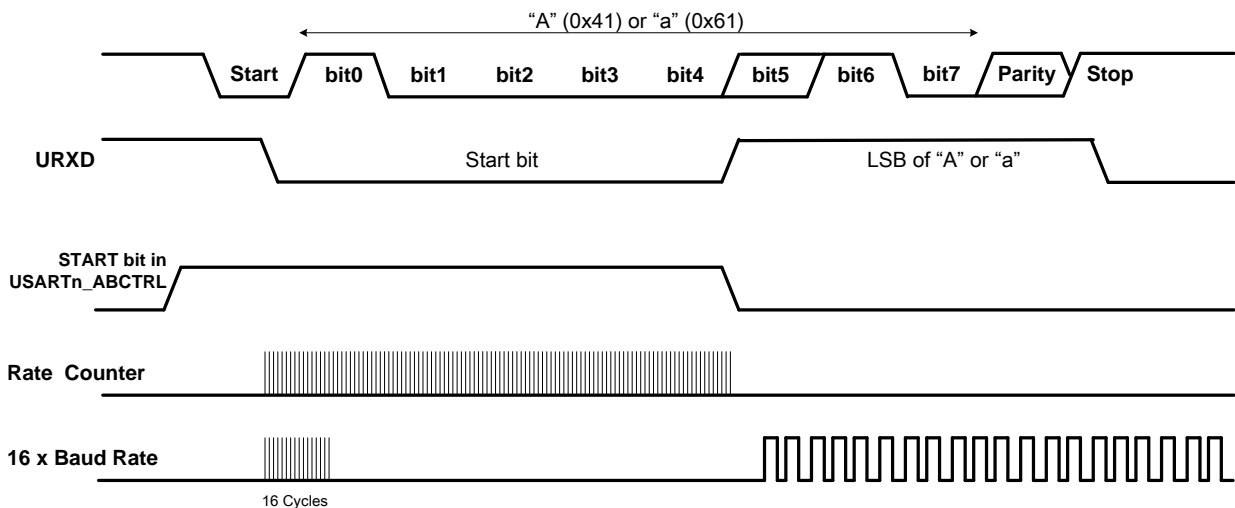
SW 等待一个“AT”命令时，需配置 USART 为预期的字符格式，并且设置 ACR START 位。DLL 和 DLM 锁存的初始值忽略不计。由于 ASCII 码的“A”和“a”（A=0x41，a=0x61），因此 USART RX 引脚侦测 START 位且预期字符的 LSB 由 2 个下降沿限定。当 ACR START 位置 1 时，自动波特率协议将按以下步骤执行：

- 1、设置 START 位时，波特率测量计数器复位，RSR 复位，RSR 波特率切换到最高速率。
- 2、URXD 引脚的下降沿触发 START 位，速率测量计数器开始对 USARTn_PCLK 的周期进行计数。
- 3、在 START 位的接收过程中，RSR 的波特率输入端产生 16 个具有 USART 输入时钟频率的脉冲，保证 START 位存储在 RSR 中。
- 4、在收到 START（以及 USARTn_ABCTRL 寄存器的 MODE=0 时字符 LSB）过程中，速率计数器以 USART 输入时钟频率（USARTn_PCLK）持续递增计数。
- 5、若 MODE=0，速率计数器在 USART RX 引脚的下一个下降沿停止计数；若 MODE=1，速率计数器在 URXD 下一个上升沿停止计数。
- 6、速率计数器加载到 USARTn_DLM/USARTn_DLL 中，波特率转换为正常运行。设置 DLM/DLL 后，若使能，USARTn_I1 寄存器的自动波特率中断结束位 ABEINT 置 1。RSR 继续接收剩下的字符。

● AUTO-BAUD RATE MODE 0 波形



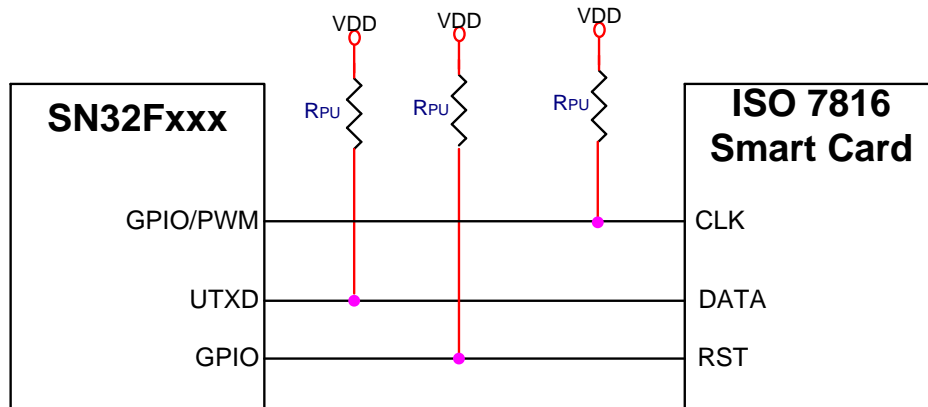
● AUTO-BAUD RATE MODE 1 波形



14.9 智能卡模式

通过设置 USARTn_CTRL 寄存器的 MODE[2:0]=011B 和 USARTEN=1，可以使能智能卡模式。在开漏 UTXD 引脚上，USART 提供双向串行数据，该模式下 URXD 引脚不能使用。若智能卡需要一个较高频率的时钟源，且频率和数据不同步，可以通过匹配的定时器输出或者 PWM 输出来提供。

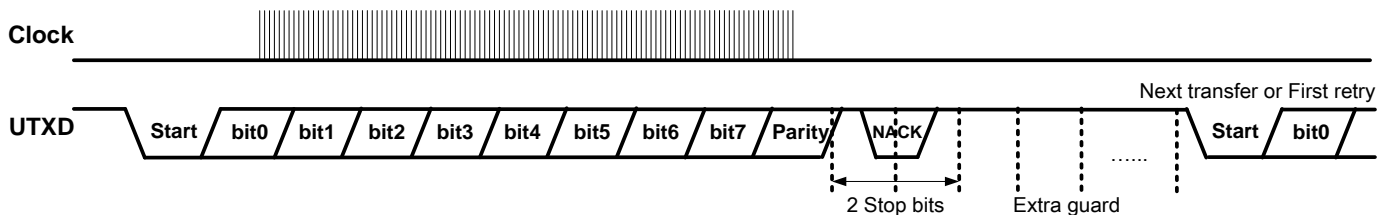
USCLK 引脚不适合大多数的异步卡，是因为它将同步输出数据和数据位率。SW 必须使用定时器实现字符和模块的等待时间。



14.9.1 智能卡设置流程

T=0 协议发送由 8 位数据，1 个偶数校验位和 2 个停止位组成，在 2 个停止位持续期间允许接收器通过 NACK 响应来标记奇偶校验位的错误，根据卡的需求可发送额外的保护位。

若没有发送 NACK，发送最后一个保护位后立即发送下一个字节；若发送了 NACK，发送器将尝试重复发送该字节，直到接收器成功接收该字节，或者达到 SCLCTRL 的重发限制次数。



智能卡必须按以下步骤设置：

- 设置 SYS1_PRST 寄存器以便 USART 不会一直复位。
- 设置 SYS1_APBPCP1 寄存器的 USARTnPRE 位，得到一个 USART 初始频率 3.58MHz。
- 若有必要，分别设置 USARTn_DLM 和 USARTn_DLL 为 00 和 01，以使 USART 时钟不分频。
- 设置 USARTn_LC 寄存器的 8 位字符，使能校验位，偶数校验位。
- 设置 USARTn_SCICTRL 寄存器以使能所需的智能卡特性，HW 自动使能 USART TXD 功能。
- 设置一个或多个定时器以提供给 ISO-7816 启动所需的时序。
- 设置 SYS_AHBCLKEN 寄存器的 USARTnCLKEN 位以使能 USART 时钟。

最后，SW 应该按照 ISO7861-3 标准实现监控卡的插入，操作激活，等待复位响应的操作

14.10 同步模式

通过在 USARTn_CTRL 寄存器的 MODE 位写入 100B 来选择同步模式。

主机模式下，USART 允许用户控制一个双向的同步串行通信。SCLK 是 USART 发送时钟的输出引脚，在 START 位和 STOP 位期间，没有时钟脉冲从 SCLK 引脚输出。

USARTn_CTRL 寄存器的 CPOL 位允许用户选择时钟的极性，CPHA 位允许用户选择时钟的相位。

空闲状态下，发送帧头和发送中断时，外部 SCLK 时钟无效。同步模式下，发送器工作方式和异步模式下工作一样，但由于 SCLK 和 TX 同步（根据 CPOL 和 CPHA），所以 TX 上的数据也是同步的。

同步模式下，USART 接收器的工作方式与异步模式下的方式是不同的。若使能接收器（RXEN=1），数据在 SCLK（取决于 CPOL 和 CPHA）被采样，且不会重复采样。必须要注意起始时间和保持时间（取决于波特率：1/16 位时间）。

*** 注：**

- 1、SCLK 引脚和 UTXD 引脚配合工作，因此只有当 USARTn_CTRL 寄存器的 TXEN=1，且有数据正在传输时（该数据已写入 USART_DR 寄存器），才会有时钟输出，这就意味在没有发送的情况下，不可能接收到一个同步数据
- 2、禁止发送器和接收器（TXEN=0 和 RXEN=0）以保证时钟脉冲功能正确时，必须选择 USARTn_SYNCCTRL 寄存器的 CPOL 和 CPHA 位，使能发送器或者接收器（TXEN=1 和 RXEN=1）时，也不能更改这两个位。
- 3、同步模式下只支持主机模式的工作，不能根据输入的时钟来接收或发送数据（SCLK 只输出）。

CPOL	CPHA	SCLK Idle Status	Diagrams
0	1	Low	
1	1	High	
0	0	Low	
1	0	High	

14.11 USART 寄存器

基地址：0x4001 6000 (USART0)
0x4005 6000 (USART1)

14.11.1 USART n 接收缓存寄存器 (USARTn_RB) (n=0, 1)

地址偏移量：0x00

该寄存器位于 USART RX FIFO 的顶端，包含最早接收到的字符，可以通过总线接口来读取。LSB（位 0）为第一个接收到的数据位，若接收到的数据不足 8 位，剩下的高位则用 0 来填充。

如果要访问 USARTn_RB 寄存器，USARTn_LC 寄存器的除数锁存访问位（DLAB）必须为 0。由于 PE、FE 和 BI 位和 USART RX FIFO 顶端的字节相对应（如从下一次从该寄存器读取字节），因此将接收字节和它的状态位成对读出的正确方式是：首先要读取 USARTn_LS 寄存器的内容，然后读取该寄存器的字节。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	RB[7:0]	USART RX FIFO 保存最早接收到的字节。	R	0

14.11.2 USART n 发送保持寄存器 (USARTn_TH) (n=0, 1)

地址偏移量：0x00

该寄存器位于 USART TX FIFO 的顶端，它包含 TX FIFO 最新的字节，可以通过总线接口写入。LSB 为第一个发送出去的数据位。如果要访问 USARTn_TH 寄存器，USARTn_LC 寄存器的除数锁存访问位（DLAB）必须为 0。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	TH[7:0]	发送器有效时，将发送 TX FIFO 中最早的字节。	W	0

14.11.3 USART n 除数锁存 LSB 寄存器 (USARTn_DLL) (n=0, 1)

地址偏移量：0x00

USART 除数锁存是 USART 波特率发生器的一部分，它保存了用于产生波特率时钟的 USARTn_PCLK 时钟的分频值（可选择是否使用小数分频器），波特率时钟应该是波特率的倍数，此倍数由重复采样寄存器决定（典型为 16 倍）

USARTn_DLL 和 USARTn_DLM 寄存器一起构成一个 16 位除数，USARTn_LC 寄存器的 DLAB 位必须为 1 以能访问这些寄存器。

DLL 包含了除数的低 8 位，DLM 则包含了高 8 位。0 值表示为 0x0001。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DLL[7:0]	USART 除数锁存 LSB 寄存器和 DLM 寄存器，共同决定 USART 的波特率。	R/W	0

14.11.4 USART n 除数锁存 MSB 寄存器 (USARTn_DLM) (n=0, 1)

地址偏移量：0x04

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DLM[7:0]	USART 除数锁存 MSB 寄存器和 DLL 寄存器，共同决定 USART 的波特率。	R/W	0

14.11.5 USART n 中断使能寄存器 (USARTn_IE) (n=0, 1)

地址偏移量: 0x04

如果要访问 USARTn_IE 寄存器, USARTn_LC 寄存器的 DLAB 位必须为 0。

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	TXERRIE	TXERR 中断使能位。通过 USARTn_LS 的 TXERR 位读取该中断的状态。 0: 禁止; 1: 使能。	R/W	0
9	ABTOIE	自动波特率超时中断使能位。 0: 禁止; 1: 使能。	R/W	0
8	ABEOIE	自动波特率结束中断使能位。 0: 禁止; 1: 使能。	R/W	0
7:5	Reserved		R	0
4	TEMTIE	TEMT 中断使能位。通过 USARTn_LS 寄存器的 TEMT 位读取该中断的状态。 0: 禁止; 1: 使能。	R/W	0
3	MSIE	调制解调器中断使能位。通过读取 USARTn_MS 寄存器识别该中断的不同中断源类型。 0: 禁止; 1: 使能。	R/W	0
2	RLSIE	接收线状态 (RLS) 中断使能位。通过 USARTn_LS[4:1] 读取该中断的状态。 0: 禁止; 1: 使能。	R/W	0
1	THREIE	THRE 中断使能位。通过 USARTn_LS 寄存器的 THRE 位读取该中断的状态。 0: 禁止; 1: 使能。	R/W	0
0	RDAIE	RDA 中断使能位。使能接收数据有效中断, 还控制字符接收超时中断。 0: 禁止; 1: 使能。	R/W	0

14.11.6 USART n 中断标识寄存器 (USARTn_II) (n=0,1)

地址偏移量: 0x08

该寄存器提供一个状态码, 来指示挂起中断的中断源和优先级。在访问 USARTn_II 寄存器期间禁止中断, 若在这期间有中断发生, 记录该中断, 并在下一次访问 USARTn_II 寄存器时读出。

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	TXERRIF	TXERR 中断标志位。 0: 无 TXERR 中断; 1: 发生 TXERR 中断并使能中断。	R	0
9	ABTOIF	自动波特率超时中断标志位。 0: 自动波特率没有超时; 1: 自动波特率超时并使能中断。	R	0
8	ABEOIF	自动波特率结束中断标志位。 0: 自动波特率没有结束; 1: 自动波特率已经结束并使能中断。	R	0
7:6	FIFOEN	等同于 USARTn_FIFOCTRL 寄存器的 FIFOEN 位。	R	1
5:4	Reserved		R	0
3:1	INTID[2:0]	中断标识, 表示识别 USARTn RX FIFO 相对应的中断。 0x3: 1 – 接收线状态 (RLS); 0x2: 2a – 接收数据有效 (RDA); 0x6: 2b – 字符超时指示 (CTI); 0x1: 3a – THRE 中断; 0x0: 4 – 调制解调器状态; 0x7: 3b – TEMT 中断; 其它: 保留。	R	0
0	INTSTATUS	中断状态。通过 USARTn_II[3:1]来确定挂起的中断。 0: 至少有一个中断挂起; 1: 没有中断挂起。	R	1

根据自动波特率功能, 超时信号或者自动波特率结束条件来设置 USARTn_II[9:8]位。在自动波特率控制寄存器中设置相应的清零位将自动波特率中断条件清零。

通过 USARTn_II[3:0]的状态, 中断处理程序可以确定中断源以及如何清除激活的中断。在退出中断服务程序前, 必须读取 USARTn_II 寄存器来清除中断。

Interrupt	USARTn_II [3:0]	Priority	Interrupt Source	Interrupt Reset
RLS	0110	Highest	溢出错误 (OE), 优先权错误 (PE), 帧错误 (FE), 间隔中断 (BI)	读取 USARTn_LS 寄存器
RDA	0100	2nd	FIFO 中的 RX 数据到达触发等级 (FCR0=1)	读取 USARTn_RB 寄存器, 或 USART FIFO 低于触发等级
CTI	1100	2nd	RX FIFO 包含至少一个字符并且在一段时间内没有字符输入或移出, 该时间取决于 FIFO 中的字符数以及在 3.5-4.5 个字符的时间内设置的触发等级	读取 USARTn_RB 寄存器
THRE	0010	3rd	THRE	读取 SARTn_II 寄存器 (中断源) 或写入 THR 寄存器
MS	0000	Lowest	CTS, DSR, RI, or DCD.	MSR Read
TEMT	1110	3rd	TEMT	读取 SARTn_II 寄存器 (中断源) 或写入 THR 寄存器

14.11.7 USART n 线状态寄存器 (USARTn_LS) (n=0,1)

地址偏移量: 0x14

* 注:

- 1、间隔中断 (BI) 与 USARTn_RB FIFO 顶端的字节有关。
- 2、帧错误 (FE) 与 USARTn_RB FIFO 顶端的字节有关。
- 3、优先级错误 (PE) 与 USARTn_RB FIFO 顶端的字节有关。

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	TXERR	TX 错误标志。(仅在智能卡 T=0 操作模式下有效) 0: 无 TX 错误; 1: 智能卡 NACK 发送字符的次数达到 TXRETRY 限定的次数	R	0
7	RXFE	RX FIFO 错误标志。 一个 RX 字符错误, 如帧错误, 奇偶校验错误, 或 Break 中断加载到 USARTn_RB 寄存器中时, RXFE = 1。读取 USARTn_LS 寄存器, 且 USART FIFO 中不再有错误时, 该位被清零 0: USARTn_RB 寄存器中没有包含 USART RX 错误, 或 FIFOEN=0。 1: USARTn_RB 寄存器至少包含了一个 USART RX 错误。	R	0
6	TEMT	发送器空标志。 THR 和 TSR 都为空时, TEMT = 1; TSR 或 THR 包含有效数据时 TEMT=0。 0: THR 和 TSR 中包含有效数据; 1: THR 和 TSR 都为空。	R	1
5	THRE	发送器保持寄存器空标志。 THRE 表示 USART 准备好接收发送过来的新字符。另外, 若 THREIE=1, 该位会使 USART 请求 THRE 中断, 从 THR 发送字符到 TSR 时, THRE=1。CPU 加载发送器保持寄存器的同时, 该位复位为 0。 0: THR 包含有效数据; 1: THR (TX FIFO) 为空。	R	1
4	BI	间隔中断标志。 RXD1 保持一个全字符 (START, DATA, PARITY, STOP) 传输时间的空白状态 (全 0) 时, 发生间隔中断。一旦检测到间隔条件, 接收器便进入空闲状态直至 RXD1 进入到标记状态 (全 1)。通过读 USARTn_LS 寄存器清 BI 位。 Break 检测的时间取决于 USARTn_FIFOCTRL 寄存器的 FIFOEN 位。 0: 间隔中断状态无效; 1: 间隔中断状态有效。	R	0
3	FE	帧错误标志。 当接收到的字符 STOP 位为逻辑 0 时, 发生帧错误。通过读 USARTn_LS 寄存器将 FE 位清零。帧错误检测的时间取决于 USARTn_FIFOCTRL 寄存器的 FIFOEN 位。 检测到帧错误时, RX 将试图重新与数据同步, 并假设错误的 STOP 位实际上就是超前的 START 位。但是, 即使没有帧错误, 也不能假定下一个接收的数据是正确的。 0: 帧错误状态无效; 1: 帧错误状态有效。	R	0
2	PE	奇偶校验错误标志。 当接收字符的校验位是错误的, 则发生奇偶校验错误。通过读 USARTn_LS 寄存器将 PE 位清零。奇偶校验错误的检测时间取决于 USARTn_FIFOCTRL 寄存器的 FIFOEN 位。 0: 奇偶校验错误状态无效; 1: 奇偶校验错误状态有效。	R	0
1	OE	溢出错误标志。 溢出错误条件在错误发生后置 1。通过读 USARTn_LS 寄存器 OE 位清零, 当 USART RSR 有新的字符且 USARTn_RB FIFO 处于全满状态时, OE=1。此时, USARTn_RB FIFO 不会被覆盖, USARTn_RS 寄存器中的字符将丢失。 0: Overrun 错误状态无效; 1: Overrun 错误状态有效。	R	0
0	RDR	接收器数据准备标志位。 USARTn_RB FIFO 中有未读取的字符时 RDR=1, USARTn_RB FIFO 为空时该位清零。 0: USARTn_RB FIFO 为空; 1: USARTn_RB FIFO 包含有效数据。	R	0

14.11.8 USART n FIFO 控制寄存器 (USARTn_FIFOCTRL) (n=0,1)

地址偏移量: 0x08

该寄存器控制 USART RX 和 TX FIFO 的操作。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:6	RXTL[1:0]	RX 触发等级。在激活一个中断之前，这 2 个位决定可以写入多少个接收器 USART FIFO 字符。 00: 触发等级 0 (1 个字符)； 01: 触发等级 1 (4 个字符)； 10: 触发等级 2 (8 个字符)； 11: 触发等级 3 (14 个字符)。	W	0
5:3	Reserved		R	0
2	TXFIFORST	TX FIFO 复位位。 0: 对 USART FIFO 没有影响； 1: 写入逻辑 1 复位 USART TX FIFO 的指针逻辑，HW 会自动将该位清零。	W	0
1	RXFIFORST	RX FIFO 复位位。 0: 对 USART FIFO 没有影响； 1: 写入逻辑 1 复位 USART RX FIFO 的指针逻辑，HW 会自动将该位清零。	W	0
0	FIFOEN	使能 FIFO。 0: 没有影响。 1: 使能 USART RX 和 TX FIFOs，访问 USARTn_FIFOCTRL[7:1]，为合适的 USART 操作，该位必须置 1。	W	1

14.11.9 USART n 线控制寄存器 (USARTn_LC) (n=0,1)

地址偏移量: 0x0C

该寄存器决定准备发送/接收的数据字符的格式，

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	DLAB	除数锁存访问位。 0: 禁止访问； 1: 使能访问。	R/W	0
6	BC	间隔控制位。 0: 禁止间隔发送； 1: 使能间隔发送，输出引脚 USART TXD 强制为逻辑 0。	R/W	0
5:4	PS[1:0]	校验选择位。 00: 奇数校验，发送字符中 1 的个数加上附加校验位 (1 或 0) 的 1 的总个数为奇数； 01: 偶数校验，发送字符中 1 的个数加上附加校验位 (1 或 0) 的 1 的总个数为偶数； 10: 强制为 1； 11: 强制为 0。	R/W	0
3	PE	校验使能位。 0: 禁止生成和检测校验位； 1: 使能生成和检测校验位。	R/W	0
2	SBS	Stop 位选择位。 0: 1 个 STOP 位； 1: 2 个 STOP 位 (若 WLS 位=00 为 1.5)，智能卡模式下必须为 1。	R/W	0
1:0	WLS[1:0]	字长度选择位。 00: 5 位字符长度； 01: 6 位字符长度； 10: 7 位字符长度； 11: 8 位字符长度。	R/W	0

14.11.10 USART n 调制解调器控制寄存器 (USARTn_MC) (n=0,1)

地址偏移量: 0x10

该寄存器使能调制解调器回路模式, 控制调制解调器的输出信号。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	CTSEN	CTS 使能位。 0: 禁止自动 CTS 流控制; 1: 使能自动 CTS 流控制。	R/W	0
6	RTSEN	RTS 使能位。 0: 禁止自动 RTS 流控制; 1: 使能自动 RTS 流控制。	R/W	0
5:2	Reserved		R	0
1	RTSCTRL	调制解调器输出引脚 RTS, .	R/W	0
0	Reserved		R	0

14.11.11 USART n 调制解调器状态寄存器 (USARTn_MS) (n=0,1)

地址偏移量: 0x18

该寄存器是一个只读寄存器, 提供 USART 输入信号的状态信息。

* 注: 每当 CTS 位的状态有所改变, 使能 MODEM 状态中断时会生成一个中断。
--

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	CTS	允许发送状态位。 CTS 输入信号的补码, Modem 回路模式下, 该位连接到 USARTn_MC[1]。	R	0
3:1	Reserved		R	0
0	DCTS	Delta CTS。 CTS 输入信号状态改变时置位。读取该寄存器后清零。 0: 未检测到 Modem 的 CTS 输入信号发生变化。 1: 检测到 Modem 的 CTS 输入信号状态改变。	R	0

14.11.12 USART n 高速缓存寄存器 (USARTn_SP) (n=0, 1)

地址偏移量: 0x1C

该寄存器对 USART 操作无影响。用户自行决定对该寄存器进行写或者读操作。不提供中断接口向主机指示此寄存器所发生的读写操作。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAD[7:0]	可读、可写的字节	R/W	0

14.11.13 USART n 自动波特控制寄存器 (USARTn_ABCTRL) (n=0, 1)

地址偏移量: 0x20

该寄存器控制测量波特率生成的输入时钟/数据率的过程，用户自行决定对该寄存器进行读或者写操作。此外，为产生波特率，该寄存器还控制时钟的预分频。寄存器的复位值保持禁止的 USART 的小数功能，以确保该 USART 的 SW 和 HW 对未配备该项功能的 UARTs 的兼容性。

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9	ABTOIFC	自动波特率超时中断标志清零位。 0: 没有影响; 1: 清 ABTOIF 位, 由 HW 自动将该位清零。	W	0
8	ABEOIFC	自动波特率中断标志清零结束位。 0: 没有影响; 1: 清 ABEOIF 位, 由 HW 自动将该位清零。	W	0
7:3	Reserved		R	0
2	AUTORESTART	重启模式。 0: 不重启; 1: 在超时情况下重启 (计数器在下一个 USART RX 下降沿重启)。	R/W	0
1	MODE	自动波特率模式选择位。 0: 模式 0; 1: 模式 1。	R/W	0
0	START	在自动波特率完成后自动将该位清零。 0: 自动波特率停止 (自动波特率不工作); 1: 自动波特率启动 (自动波特率工作), 自动波特率运行位, 自动波特率完成后, 由 HW 自动将该位清零。	R/W	0

14.11.14 USART n 小数分频寄存器 (USARTn_FD) (n=0, 1)

地址偏移量: 0x28

该寄存器控制生成波特率的时钟预分频器，用户可自行决定对该寄存器进行读或者写操作。预分频器根据指定的小数要求使用 APB 时钟并产生一个输出时钟。

多数应用情况下，USART 会在每个数据 bit 持续的时间内对该数据 bit 进行 16 次采样，且 USART 发送的数据 bit 也为 16 个输入时钟宽。OVER8 位允许软件控制输入时钟和数据 bit 时钟的比率。此项功能可以在智能卡模式下使用，还可以为其它模式提供一个可供选择的小数分频功能。

* 注: 若小数分频器有效 (DIVADDVAL > 0) 且 USARTn_DLM = 0, USARTn_DLL 寄存器的值必须 ≥ 3。

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	OVER8	重复采样值。 0: 重复采样 16 次; 1: 重复采样 8 次 (不支持 IrDA 模式)。	R/W	0
7:4	MULVAL[3:0]	波特率预分频器乘数值为 MULVAL[3:0] + 1。 0000: 波特率预分频器乘数值为 1; 0001: 波特率预分频器乘数值为 2; 1111: 波特率预分频器乘数值为 16。	R/W	0
3:0	DIVADDVAL[3:0]	波特率产生预分频除数值。若该字段为 0, 小数波特率产生器不影响 USART 波特率。	R/W	0

14.11.15 USART n 控制寄存器 (USARTn_CTRL) (n=0, 1)

地址偏移量: 0x30

除了 HW 流控制 (自动 CTS 和自动 RTS 机制) 外, 该寄存器还可使能 SW 流控制。

TXEN=1 时, 只要数据有效, USART 就一直保持发送状态, 一旦 TXEN 变为 0, 就停止发送数据。

强烈建议使用 USART HW 自动流控制特性 来抵消 SW 流控制 TXEN 的限制。

* 注: 若需要将接收器的设置和保持时间降到最小, 建议在同一条指令中设置 TXEN 和 RXEN。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	TXEN	该位置 1 时, 一旦前面的数据已经发送, 写入 USARTn_TH 寄存器的数据从 TXD 引脚输出。 正在发送一个字符时, 该位清零, 则结束该字符的发送, 同时也不发送更多的字符直到该位重新置 1。 换言之, TXEN =0 将阻止 USARTn_TH 寄存器或 TX FIFO 中的字符进入发送移位寄存器中。当检测到硬件握手 TX-permit 信号 CTS 出错或利用软件握手接收到一个 XOFF 字符 (DC3) 时, 软件将该位清零。当检测到正确的 Tx-permit 信号或接收到 XON 字符 (DC1) 时, 软件又能将该位重新置 1。	R/W	1
6	RXEN	0: 禁止 RX 相关的功能; 1: 使能 RX。	R/W	1
5:4	Reserved		R	0
3:1	MODE[2:0]	USARTn 模式。 000: UART 模式。HW 将 GPIO 切换为 UTXDn 和 URXDn。 001: 调制解调器控制模式。HW 将 GPIO 切换为 UTXDn、URXDn、UDSRn、UCTSn、UDCDn、URIn、UDTRn 和 URTSn。 010: 保留。 011: 智能卡模式。HW 将 GPIO 切换为 UTXDn, 并使能 UTXDn 引脚的开漏功能。 100: 同步模式。HW 将 GPIO 切换为 UTXDn、URXDn、和 USCLK。 101: RS-485 模式。HW 将 GPIO 切换为 UTXDn、URXDn。	R/W	0
0	USARTEN	USART 使能位。 0: 禁止, 所有的 USART 引脚都作为 GPIO 引脚; 1: 使能, 根据 MODE 位, HW 自动将 GPIO 设为 USART 引脚。	R/W	0

14.11.16 USART n 半双工传输使能寄存器 (USARTn_HDEN) (n=0, 1)

地址偏移量: 0x34

USART 复位后进入全双工传输模式, 意味着 TX 和 RX 都独立工作。HDEN 位置 1 后, USART 进入半双工传输模式, 在该模式下, USART 确保接收器空闲时为锁存状态, 否则在接收到完整的连续的字符后进入到锁存状态, 必须由 SW 处理线路冲突。

在接收正在发送的数据的过程中, USART 的动作是无法预知的, 因此, 接收或发送数据时, 不能更改 HDEN 寄存器的值, 否则会丢失或毁坏数据。

* 注: 智能卡模式或 IRDA 模式下应该禁止该寄存器 (智能卡和 IRD 默认在半双工模式下工作)。

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	HDEN	半双工传输模式使能位。 0: 禁止; 1: 使能。	R/W	0

14.11.17 USART n 智能卡接口控制寄存器 (USARTn_SCICTRL) (n=0, 1)

地址偏移量: 0x38

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:16	TC[7:0]	SCLKEN=1 时对 SCLK 时钟周期进行计数, 每 (TC[7:0]+ 1) * USARTn_PCLK 周期翻转一次 SCLK。	R/W	0x0
15:8	XTRAGUARD	当协议选择 T = 0, 这表示: USART 发送字符后的保护位的时间应该超过了 2 个 bit 的时间。0xFF 表示: 1 个单独的 bit 在 11 位的时间/字符后面。	R/W	N/A
7:5	TXRETRY[2:0]	当协议选择 T = 0 时, 如果远程设备返回 NACK 信号, TXRETRY[2:0] 值域控制 USART 进行数据重发的最大次数。当有 NACK 发生时, 重发计数加 1, USARTn_LS 寄存器的 TX 错误 (TXERR) 位置 1, 若使能则请求中断, 且 USART 被锁存直至 FIFO 被清除。	R/W	N/A
4	Reserved		R	0
3	SCLKEN	SCLK 使能位。若进行通信的智能卡需要时钟信号时使能该位。 0: 禁止; 1: 使能, HW 切换 GPIO 为 UnSCLK 引脚。	R/W	0
2	PROTSEL	协议选择 ISO7816-3 中定义的标准。 0: T = 0; 1: T = 1。	R/W	0
1	NACKDIS	NACK 响应禁止位, 仅当 T=0 时适用。 0: 使能 NACK 响应; 1: 禁止 NACK 响应。	R/W	0
0	Reserved		R	0

14.11.18 USART n RS485 控制寄存器 (USARTn_RS485CTRL) (n=0, 1)

地址偏移量: 0x3C

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5	OINV	极性控制位。该位将 RTS 引脚上控制信号的极性进行反向。 0: 发送器中还有数据要发送时, 方向控制引脚被设为逻辑 0, 发送完所有数据后设为逻辑 1; 1: 发送器中还有数据要发送时, 方向控制引脚被设为逻辑 1, 发送完所有数据后设为逻辑 0。	R/W	0
4	ADCEN	自动方向控制使能位。 0: 禁止; 1: 使能, RTS 引脚用于控制方向, HW 自动切换 GPIO 引脚为 UTRSn 引脚。	R/W	0
3	Reserved		R	0
2	AADEN	自动地址检测 (AAD) 使能位。 0: 禁止; 1: 使能。	R/W	0
1	RXEN	RS-485/EIA-485 接收器使能位 (仅在 NMMEN=1 时工作)。 0: 禁止; 1: 使能。	R/W	0
0	NMMEN	RS-485/EIA-485 标准多路模式 (NMM) 使能位。 0: 禁止; 1: 使能, 该模式下, 接收到的字节导致 USART 奇偶校验设置错误和产生中断时检测到一个地址。	R/W	0

14.11.19 USART n RS485 地址匹配寄存器 (USARTn_RS485ADRMATCH) (n=0, 1)

地址偏移量: 0x40

该寄存器包含 RS-485/EIA-485 模式的地址匹配值。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	MATCH[7:0]	地址匹配值。	R/W	0

14.11.20 USART n RS485 延时寄存器 (USARTn_RS485DLYV) (n=0, 1)

地址偏移量: 0x44

用户可以通过编程该寄存器设置在最后一个离开 TXFIFO 的 STOP 位到 RTS 失效之间的延时时间。延时时间以波特率时钟周期为基数进行计量, 可以设置为 0~255 倍基数值的延时时间。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DLY[7:0]	方向控制 (RTS) 延时值。该寄存器和 8 位计数器协同工作。	R/W	0

14.11.21 USART n 同步模式控制寄存器 (USARTn_SYNCCTRL) (n=0,1)

地址偏移量: 0x48

该寄存器控制同步模式，同步模式下，USART 在 SCLK 引脚上产生或接收一个位时钟，并将该时钟用于发送和接收移位寄存器。

同步模式不能用于智能卡模式。

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	CPHA	边沿采样的时钟相位。 0: 在 SCLK 的上升沿采样; 1: 在 SCLK 的下降沿采样。	R/W	0
1	CPOL	时钟极性选择位。 0: SCLK 空闲状态为低电平; 1: SCLK 空闲状态为高电平。	R/W	0
0	Reserved		R	0

15 I2S

15.1 概述

I2S 总线规范定义了一个 5 线串行总线，包括一个数据输入线，一个数据输出线，BCLK，MCLK 和一个字选择信号线。基本的 I2S 连接包括一个主机和一个从机。

15.2 特性

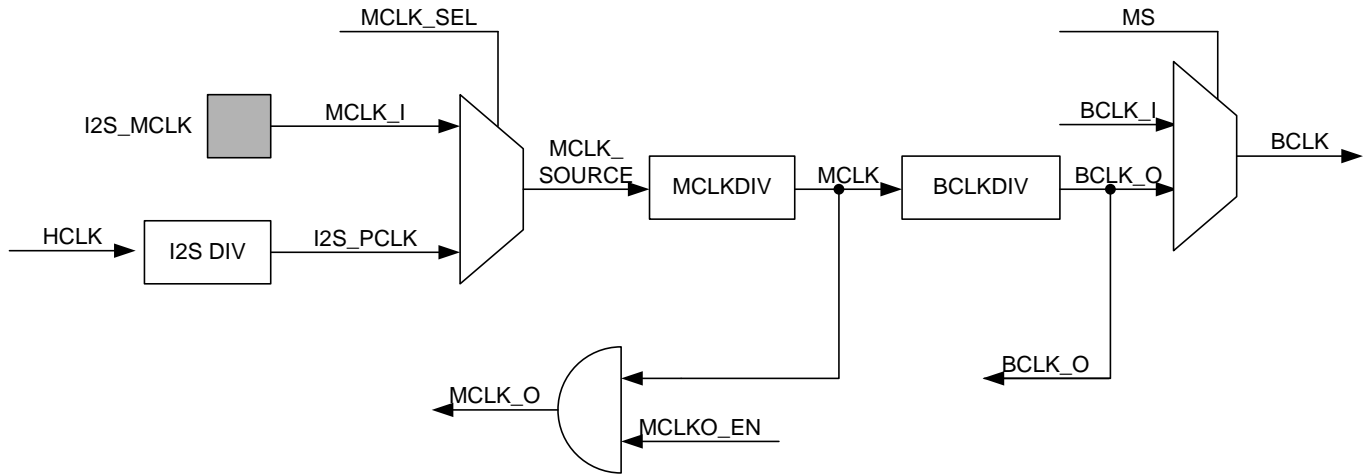
- I2S 可以作为主机或从机运行。
- 可处理 8/16/24/32 位长度的数据。
- 支持单声道和立体声音频数据。
- 支持 I2S 和 MSB 数据格式。
- 提供 8 层 32 位数据长度 FIFO 数据缓存器。
- 当缓冲等级超过设定的范围时请求中断。
- 可为 I2S 输入和输出单独选择复位，停止或静音控制方式。

15.3 引脚说明

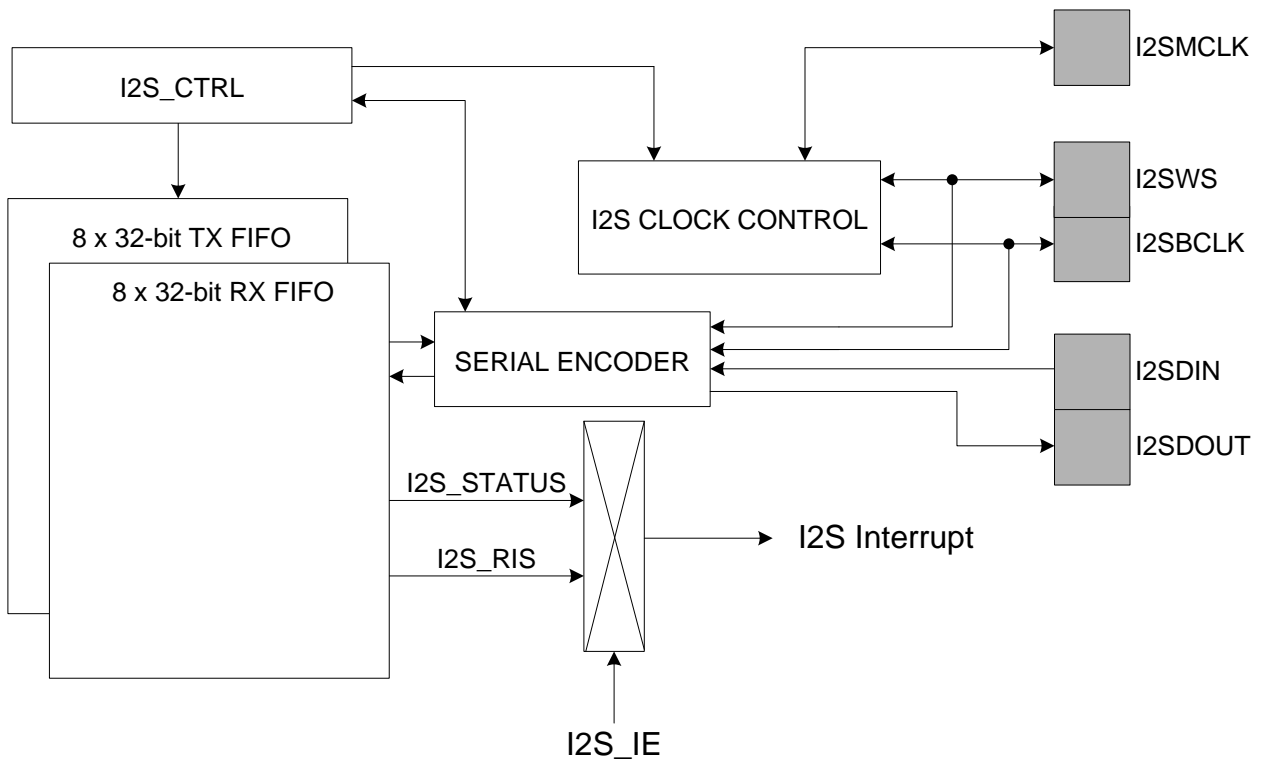
Pin Name	Type	Description	GPIO Configuration
I2SBCLK	O	I2S 位时钟 (主机)	
	I	I2S 位时钟 (从机)	取决于 GPIOOn_CFG
I2SWS	O	I2S 字选择 (主机)	
	I	I2S 字选择 (从机)	取决于 GPIOOn_CFG
I2SDIN	I	I2S 接收串行数据	取决于 GPIOOn_CFG
I2SDOUT	O	I2S 发送串行数据	
I2SMCLK	O	I2S 主机时钟输出	
	I	I2S 主机时钟从 GPIO 输入	取决于 GPIOOn_CFG

15.4 框图

15.4.1 I2S 时钟控制



15.4.2 I2S 框图

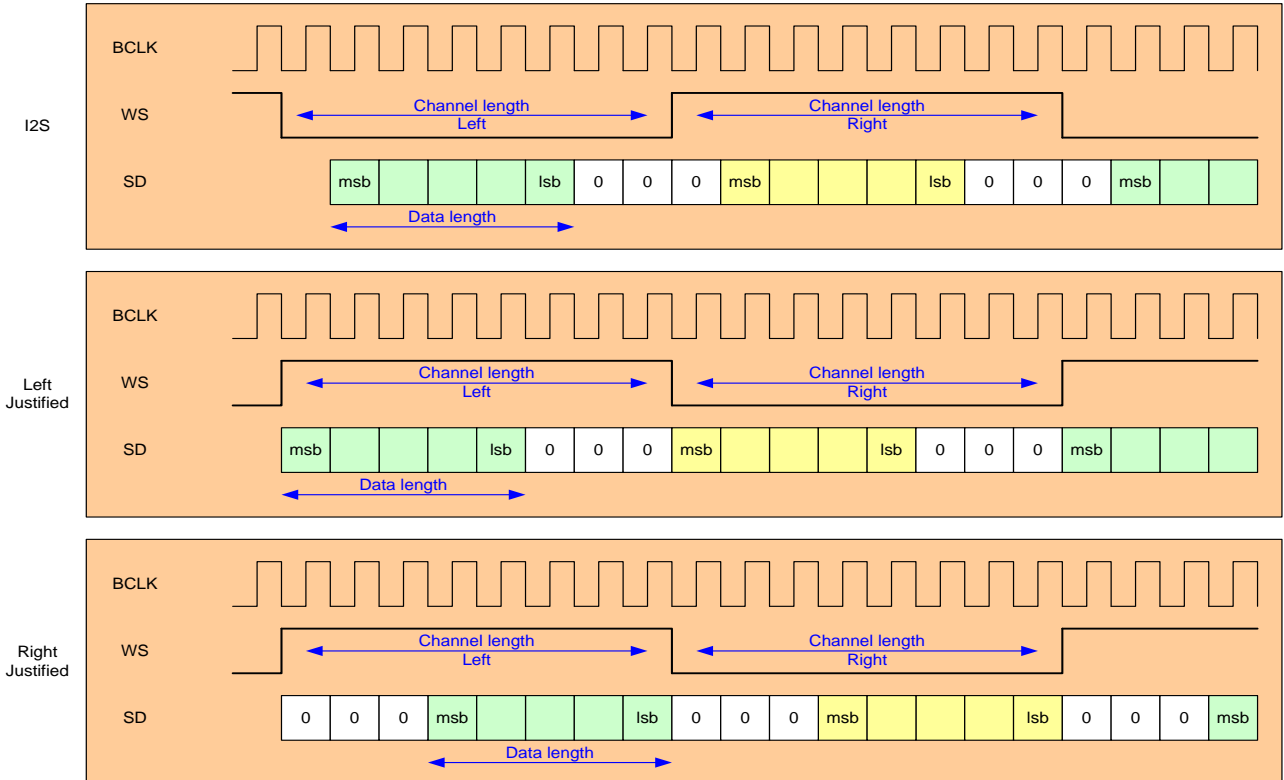


15.5 功能说明

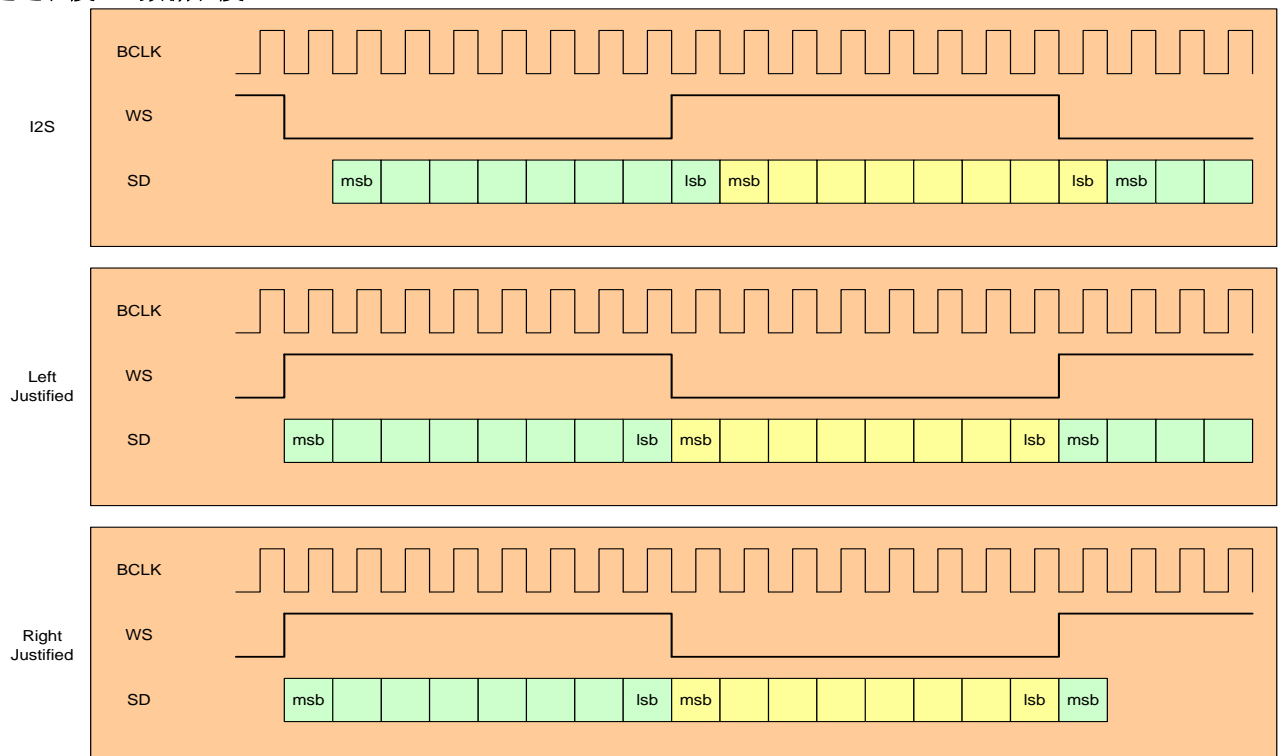
15.5.1 I2S 操作

- 标准 I2S
- 向右对齐的数据格式。
- MSB（左）-对齐的数据格式。

- 通道长度 > 数据长度:



- 通道长度 = 数据长度:



15.5.2 I2S FIFO 操作

15.5.2.1 单声道

- 8bit

N+3	N+2	N+1	N
N+7	N+6	N+5	N+4

- 16bit

N+1	N
N+3	N+2

- 24 bit

N
N+1

- 32 bit

N
N+1

15.5.2.2 立体声

- 8bit

RIGHT +1	LEFT +1	RIGHT	LEFT
RIGHT +3	LEFT +3	RIGHT +2	LEFT +2

- 16bit

RIGHT	LEFT
RIGHT +1	LEFT+1

- 24 bit

LEFT
RIGHT

- 32 bit

LEFT
RIGHT

15.6 I2S 寄存器

基地址：0x4001 A000

15.6.1 I2S 控制寄存器 (I2S_CTRL)

地址偏移量：0x00

* 注：应该最后设置 START 位。

Bit	Name	Description	Attribute	Reset
31	I2SEN	I2S 使能控制位。 0: 禁止; 1: 使能。	R/W	0
30:25	Reserved		R	0
24:20	CHLENGTH[4:0]	单通道的位数量 = CHLENGTH[4:0]+1。 0~6: 保留; 7: 8 位; 8: 9 位; 31: 32 位 (最大值)。	R/W	0x1F
19	Reserved		R	0
18:16	RXFIFOTH[2:0]	RX FIFO 阈电平。 000: RX FIFO 阈电平为 0; 001: RX FIFO 阈电平为 1; ... 111: RX FIFO 阈电平为 n。	R/W	0x3
15	Reserved		R	0
14:12	TXFIFOTH[2:0]	TX FIFO 阈电平。 000: TX FIFO 阈电平为 0; 001: TX FIFO 阈电平为 1; ... 111: TX FIFO 阈电平为 n。	R/W	0x3
11:10	DL[1:0]	数据长度。 00: 8 位; 01: 16 位; 10: 24 位; 11: 32 位。	R/W	0x1
9	CLRRXFIFO	I2S RX FIFO 清零位。 0: 没有影响; 1: RX FIFO 复位 (RXFIFOLV 位为 0, RXFIFOEMPTY 位为 1, 数据输入 RX FIFO 清零), 该位自动返回到 0。	W	0
8	CLRTXFIFO	I2S TX FIFO 清零位。 0: 没有影响; 1: TX FIFO 复位 (TXFIFOLV 位为 0, TXFIFOEMPTY 位为 1, 数据输入 TX FIFO 清零), 该位自动返回到 0。	W	0
7	RXEN	接收器使能控制位。 0: 禁止; 1: 使能。	R/W	0
6	TXEN	发送器使能控制位。 0: 禁止; 1: 使能。	R/W	0
5:4	FORMAT[1:0]	I2S 操作格式选择位。 00: 标准 I2S 格式; 01: 左对齐格式; 10: 右 (MSB) 对齐格式; 11: 保留。	R/W	0
3	MS	主机/从机选择位。 0: 主机, 内部产生 BCLK 和 WS 信号; 1: 从机, 外部产生 BCLK 和 WS 信号。	R/W	0

2	MONO	单声道/立体声选择位。 0: 立体声; 1: 单声道。	R/W	0
1	MUTE	静音使能位。 0: 禁止静音; 1: 使能静音, I2SSDA 输出为 0。	R/W	0
0	START	发送/接收起始位。 0: 停止发送/接收数据; 1: 开始发送/接收数据。	R/W	0

15.6.2 I2S 时钟寄存器 (I2S_CLK)

地址偏移量: 0x04

Bit	Name	Description	Attribute	Reset
31:17	Reserved		R	0
16	CLKSEL	I2S 时钟源选择位。 0: HCLK; 1: EHX XTAL。	R/W	0
15:8	BCLKDIV[7:0]	BCLK 分频选择位。 0: BCLK = MCLK / 2; 1: BCLK = MCLK / 4; 2: BCLK = MCLK / 6; 3: BCLK = MCLK / 8; N: BCLK = MCLK / (2*n + 2)。	R/W	1
7:5	Reserved		R	0
4	MCLKSEL	MCLK 源选择位。 0: 主机的 MCLK 源来自 I2S_PCLK; 1: 主机的 MCLK 源来自 GPIO。	R/W	0
3	MCLKOEN	MCLK 输出使能位。 0: 禁止; 1: 使能。	R/W	0
2:0	MCLKDIV[2:0]	MCLK 分频选择位。 0: MCLK = MCLK 源; 1: MCLK = MCLK 源 / 2; 2: MCLK = MCLK 源 / 4; N: MCLK = MCLK 源 / (2*n), n>0。	R/W	0

15.6.3 I2S 状态寄存器 (I2S_STATUS)

地址偏移量: 0x08

Bit	Name	Description	Attribute	Reset
31:21	Reserved		R	0
20:17	RXFIFOLV[3:0]	RX FIFO used level 0000: 0/8 FIFO is used (空); 0001: 1/8 FIFO is used 0010: 2/8 FIFO is used 1000: 8/8 FIFO is used (满) 其它: 保留。	R	0
16	Reserved		R	0
15:12	TXFIFOLV[3:0]	TX FIFO used level 0000: 0/8 FIFO is used (空); 0001: 1/8 FIFO is used 0010: 2/8 FIFO is used ...	R	0

		... 1000: 8/8 FIFO is used (满) 其它: 保留。		
11	RXFIFOEMPTY	RX FIFO 空标志位。 0: RX FIFO 非空; 1: RX FIFO 空, 从 RX FIFO 读取的数据为 0。	R	1
10	TXFIFOEMPTY	TX FIFO 空标志位。 0: TX FIFO 非空; 1: TX FIFO 空。	R	1
9	RXFIFOFULL	RX FIFO 满标志位。 0: RX FIFO 非满; 1: RX FIFO 满。	R	0
8	TXFIFOFULL	TX FIFO 满标志位。 0: TX FIFO 非满; 1: TX FIFO 满, 对 TX FIFO 的写动作无效。	R	0
7	RXFIFOTHF	RX FIFO 门限标志位。 0: RX FIFOLV \cong RX FIFOTH; 1: RX FIFOLV > RX FIFOTH。	R	0
6	TXFIFOTHF	TX FIFO 门限标志位。 0: TX FIFOLV \cong TX FIFOTH; 1: TX FIFOLV < TX FIFOTH。	R	1
5:2	Reserved		R	0
1	RIGHTCH	当前通道状态位。 0: 当前通道为左通道; 1: 当前通道为右通道。	R	1
0	I2SINT	I2S 中断标志位。 0: 无 I2S 中断; 1: 发生 I2S 中断。	R	0

15.6.4 I2S 中断使能寄存器 (I2S_IE)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	RXFIFOTHIEN	RX FIFO 门限中断使能位。 0: 禁止; 1: 使能。	R/W	0
6	TX FIFOTHIEN	TX FIFO 门限中断使能位。 0: 禁止; 1: 使能。	R/W	0
5	RXFIFOUDFIEN	RX FIFO underflow 中断使能位。 0: 禁止; 1: 使能。	R/W	0
4	TXFIFOOVFIEN	TX FIFO 溢出中断使能位。 0: 禁止; 1: 使能。	R/W	0
3:0	Reserved		R	0

15.6.5 I2S 中断源状态寄存器 (I2S_RIS)

地址偏移量: 0x10

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	RXFIFOTHIF	RX FIFO 门限中断标志位。 0: 无 RX FIFO 门限中断; 1: 触发 RX FIFO 门限中断。	R	0
6	TXFIFOTHIF	TX FIFO 门限中断标志位。 0: 无 TX FIFO 门限中断; 1: 触发 TX FIFO 门限中断。	R	0
5	RXFIFOUDIF	RX FIFO underflow 中断标志位。 0: 无 RX FIFO underflow 中断; 1: RX FIFO underflow (FIFO 为空, 但仍读取 FIFO 的数据)。		
4	TXFIFOOVIF	TX FIFO 溢出中断标志位。 0: 无 TX FIFO 溢出中断; 1: TX FIFO 溢出 (FIFO 满, 但仍向 FIFO 写入数据)。	R	0
3:0	Reserved		R	0

15.6.6 I2S 中断清零寄存器 (I2S_IC)

地址偏移量: 0x14

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	RXFIFOTHIC	0: 无影响; 1: 清 FIFOTHIF 位。	W	0
6	TXFIFOTHIC	0: 无影响; 1: 清 FIFOTHIF 位。	W	0
5	RXFIFOUDIC	0: 无影响; 1: 清 FIFOUUDIF 位。	W	0
4	TXFIFOOVIC	0: 无影响; 1: 清 FIFOOVIF 位。	W	0
3:0	Reserved		R	0

15.6.7 I2S RXFIFO 寄存器 (I2S_RXFIFO)

地址偏移量: 0x18

Bit	Name	Description	Attribute	Reset
31:0	RXFIFO[31:0]	8 x 32 位 RX FIFO。	R/W	0

15.6.8 I2S TXFIFO 寄存器 (I2S_TXFIFO)

地址偏移量: 0x1C

Bit	Name	Description	Attribute	Reset
31:0	TXFIFO[31:0]	8 x 32 位 TX FIFO。	R/W	0

16 4x32 LCD 驱动

16.1 概述

SN32F240 系列单片机内置 4*32 (4 commons 和高达 32 segments, 128 dots) 的 LCD 驱动, 包括 R 型和 C 型 (1C 和 4C) 结构, 带有 4 common 引脚和高达 32 segment 引脚。LCD 扫描时序可支持 1/2, 1/3 和 1/4duty, 除此之外, 还支持 1/2 和 1/3 偏压结构。所有的 common 引脚和 segment 引脚都与 GPIO 引脚共用, 通过可编程的 LCD_CTRL 寄存器进行选择。R 型是利用内部和外部偏压电路来调节 LCD 电压和偏压; C 型是利用内部 charge pump 来调节 LCD 电压和偏压。

16.2 特性

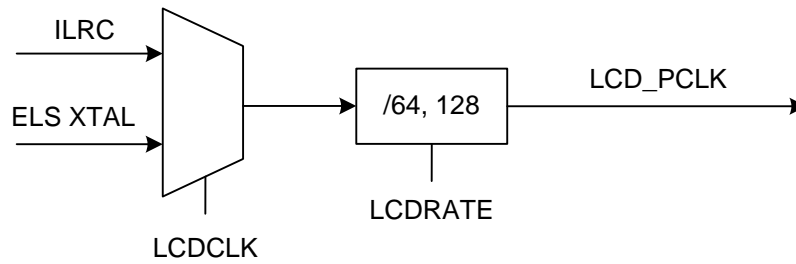
- 支持 R/C 型
- 支持高达 128 (4 x 32) dots
- Common 0~3 和 segment 0~31, V3, V2, CL+, CL-都与 GPIO 引脚共用
- 支持 1/2, 1/3 和 1/4duty
- 支持 1/2 和 1/3 偏压
- 可配置的帧频率
- 内部 Charge pump, 可调节的对比调节器和可配置的 charge pump 频率
- 嵌入式 LCD 偏压参考阶梯 (R 型)
- LCD 帧中断

16.3 引脚说明

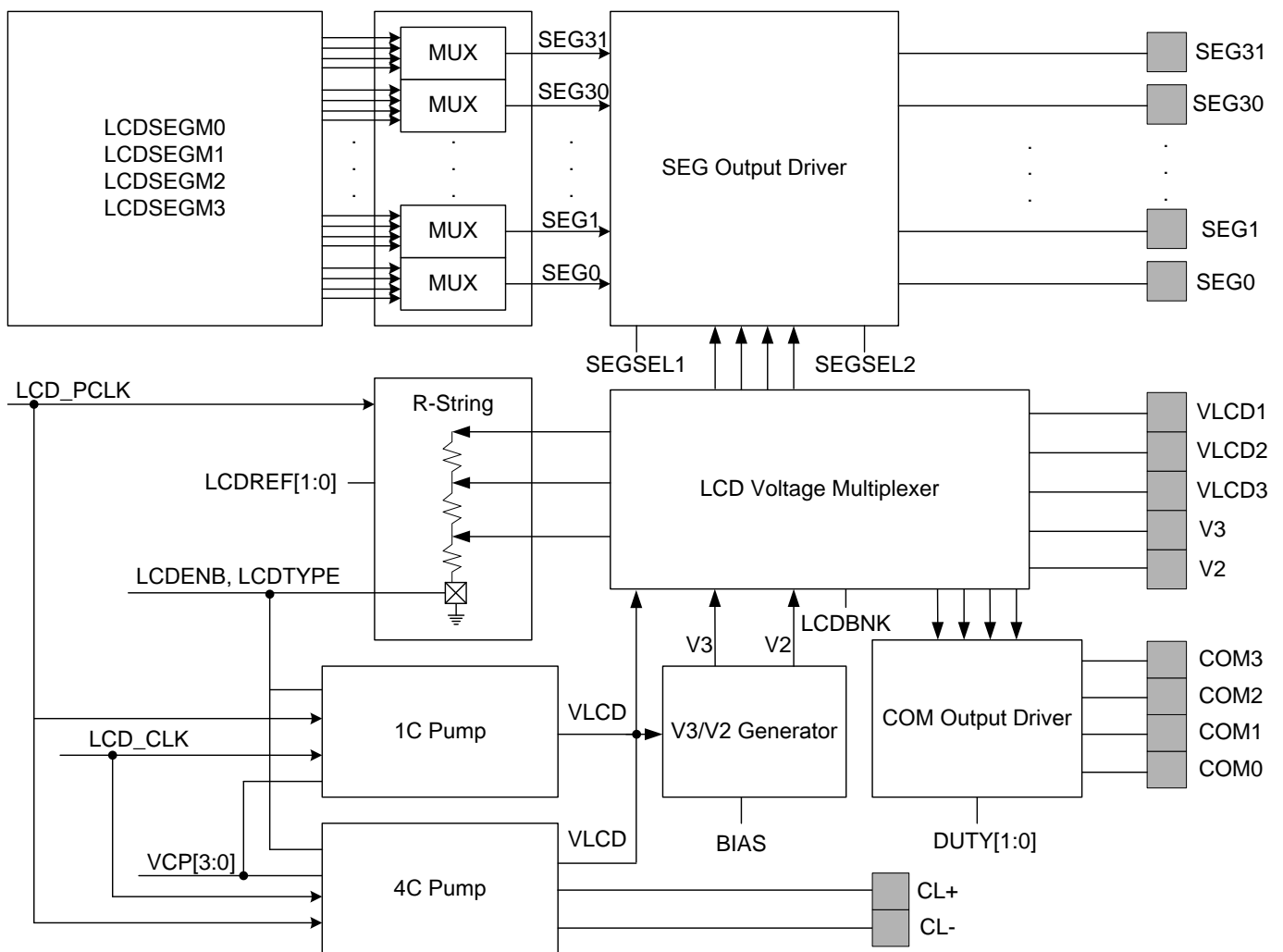
Pin Name	Type	Description	GPIO Configuration
SEGx	O	LCD 驱动 segment 引脚	
COMx	O	LCD 驱动 common 引脚	
CL+	P	C-Type LCD charge pump 电容	
CL-	P	C-Type LCD charge pump 电容	
VLCD1	P	LCD COM0~3, SEG0~11 驱动电源输入引脚	
VLCD2	P	LCD SEG12~23 驱动电源输入引脚	
VLCD3	P	LCD SEG24~31 驱动电源输入引脚	
V3	P	2/3 VLCD 偏压电源	
V2	P	1/3 VLCD 偏压电源	

16.4 框图

16.4.1 LCD 时钟控制



16.4.2 LCD 框图

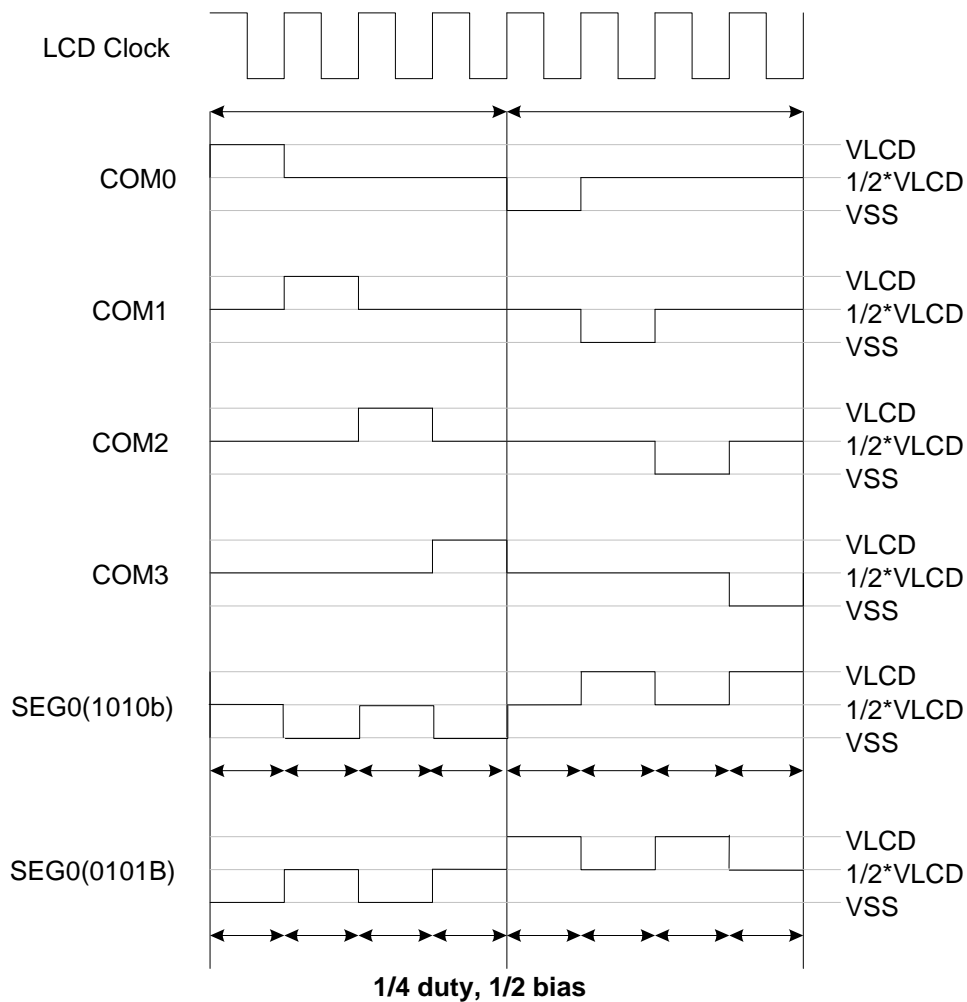


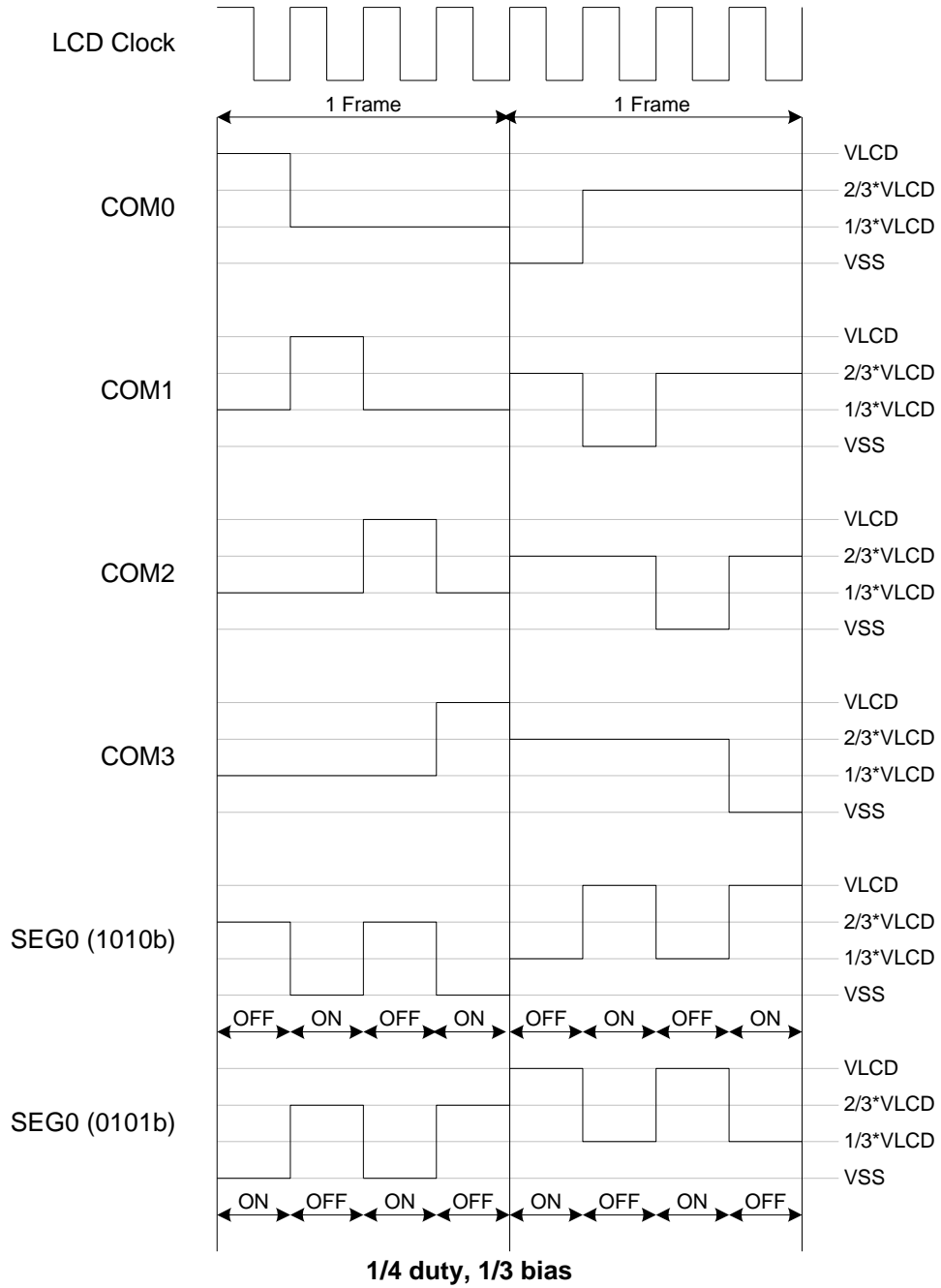
16.5 LCD 时序

16.5.1 LCD 时序列表

LCDCLK	LCD Clock Source	LCDRATE	LCD Clock Rate (LCD_PCLK)	Duty	Frame Rate = LCD clock rate * Duty
0	ILRC 32KHz	0	$32\text{KHz} / 64 = 500\text{Hz}$	1/4	$500\text{Hz} / 4 = 125\text{Hz}$
		1	$32\text{KHz} / 128 = 250\text{Hz}$	1/2	$250\text{Hz} / 2 = 125\text{Hz}$
		1	$32\text{KHz} / 128 = 250\text{Hz}$	1/4	$250\text{Hz} / 4 = 62.5\text{Hz}$
1	ELS XTAL 32768Hz	0	$32768\text{Hz} / 64 = 512\text{Hz}$	1/4	$512\text{Hz} / 4 = 128\text{Hz}$
		1	$32768\text{Hz} / 128 = 256\text{Hz}$	1/4	$256\text{Hz} / 4 = 64\text{Hz}$

16.5.2 LCD 驱动波形图



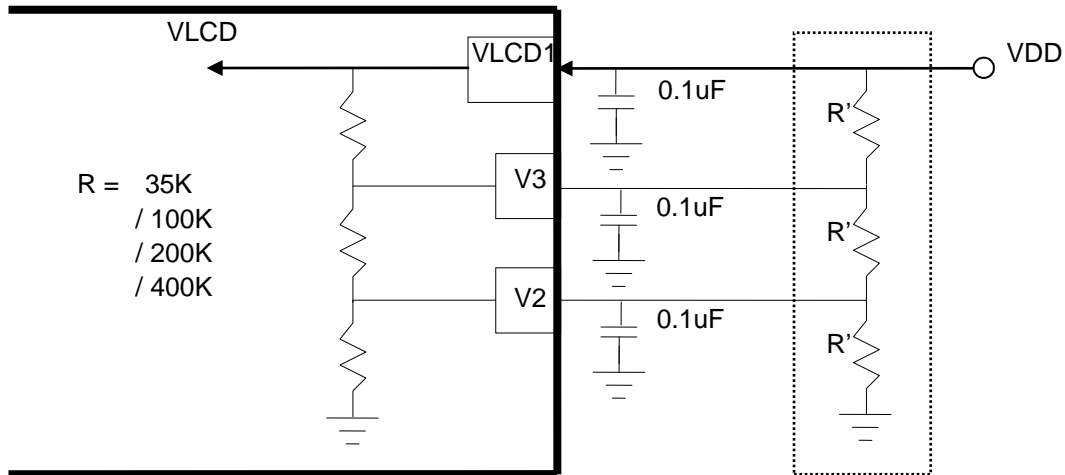


16.6 R 型 LCD 应用电路

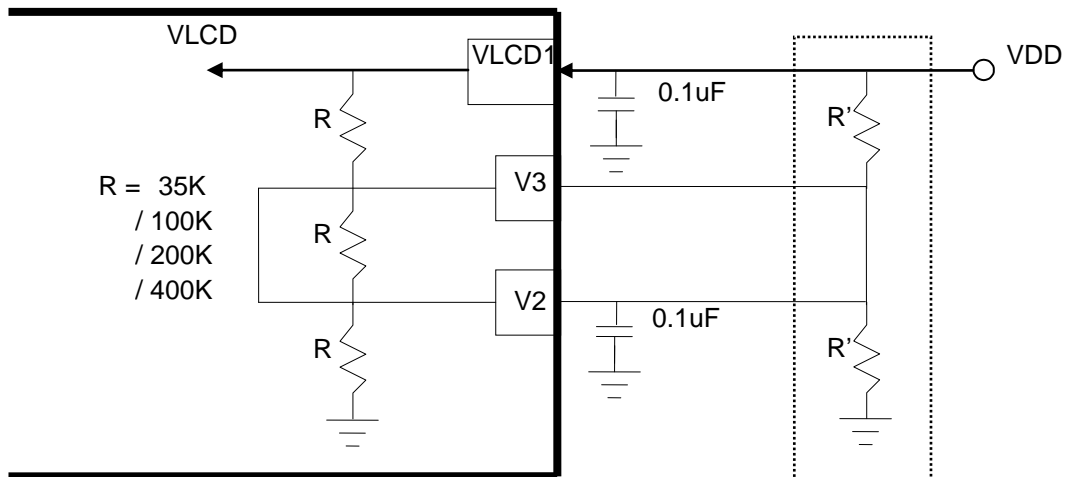
使能 R 型 LCD（在 LCD_CTRL 寄存器中设置 LCDENB=1, LCDTYPE[1:0]=00b）时，LCD 电压源（VLCD）经由 VLCD1 的外部电压提供，且在 PCB Layout 时，需要连接到 VDD，内部没有连接到 VDD。HW 会自动分配 COM0~3（参考 DUTY[1:0]，SEG0~11，V3，V2 引脚为 LCD 引脚，而不是 GPIO 引脚。

V3 和 V2 的偏压由内部电阻电压分压得到，可提供 LCD_CTRL1 寄存器的 REF[1:0] 选择内部分压电阻：35K，100K，200K 和 400K。用户可在 VLCD1/V3/V2 之间连接外部电阻以得到更多的驱动电流。

- **1/3 Bias ($V3 = 2/3 \cdot VLCD$, $V2 = 1/3 \cdot VLCD$)**



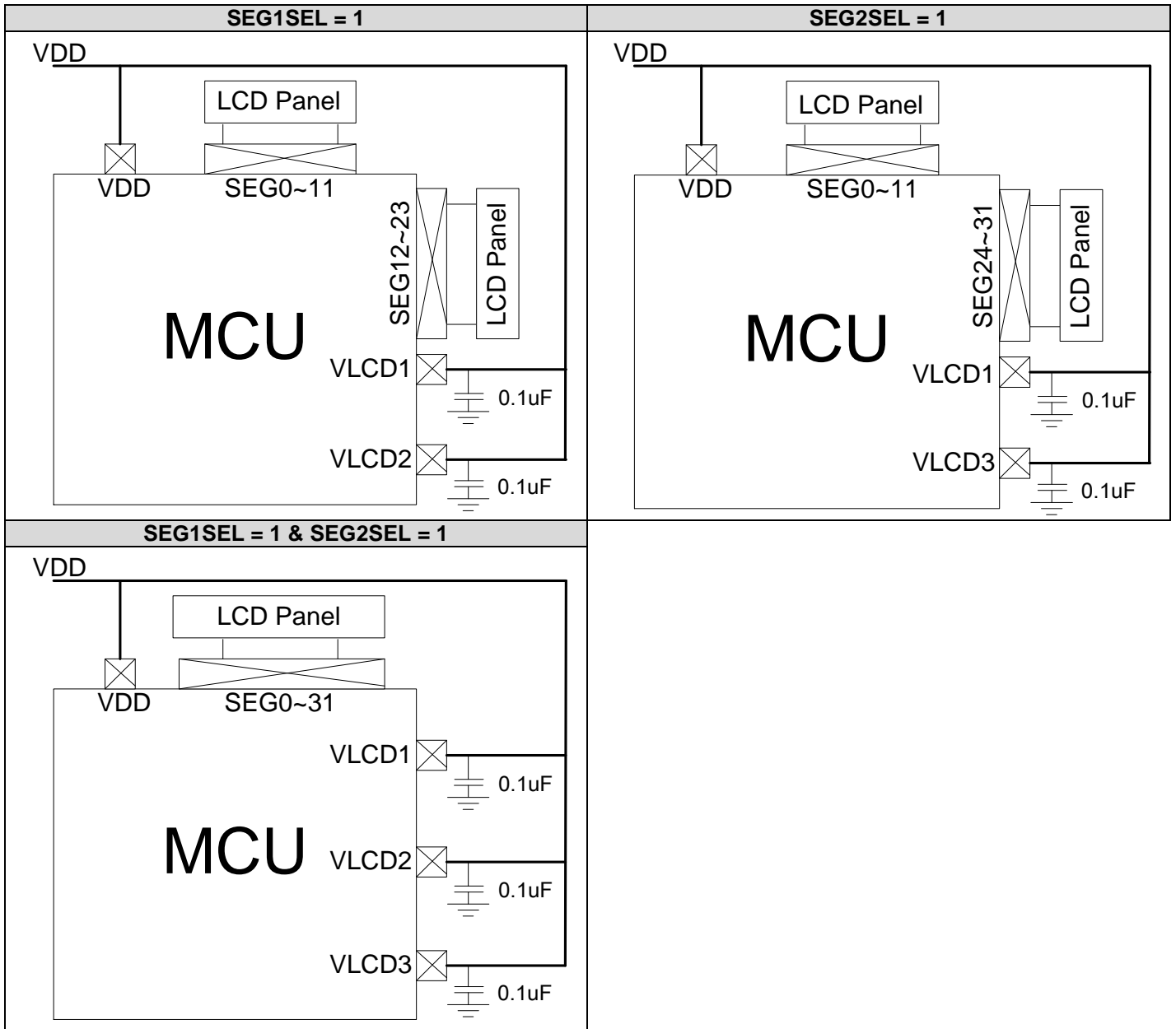
- **1/2 Bias ($V3 = V2 = 1/2 \cdot VLCD$)**



* 注：必须在 VLCD1/V3/V2 连接一个 0.1uF，且要靠近 VLCD1/V3/V2 引脚。

Segment 12~Segment 23 引脚与 P0.10~P0.15 和 P1.0~P1.5 引脚共用，这些引脚作为 LCD 引脚使用时，SEG1SEL 位须置 1，PCB Layout 时，VLCD2 也应该连接到 VDD 引脚。

Segment 24~Segment 31 引脚与 P0.0~P0.7 引脚共用，这些引脚作为 LCD 引脚使用时，SEG2SEL 位须置 1，PCB Layout 时，VLCD3 也应该连接到 VDD 引脚。



16.7 C 型 LCD 应用电路

使能 C 型 LCD（在 LCD_CTRL 寄存器中设置 LCDENB=1，LCDTYPE[1:0]=01b 或 10b），LCD 电压源（VLCD）由内部 LCD Charge pump 提供，且内部短接到 VLCD1。HW 会自动分配 COM0~3，SEG0~11，V3，V2，CL+和 CL- 引脚为 LCD 引脚，而不是 GPIO 引脚。

charge-pump 的电压 VLCD 由 LCD_CTRL1 寄存器的 VCP[3:0]控制。1/3 偏压下，V2 为 charge pump 的输出电压，其值为 $1/3 \cdot VLCD$ ，V3 为 2 倍 V2，即 V3 的值为 $2/3 \cdot VLCD$ ；1/2 偏压下，V2 为 charge-pump 输出电压，其值为 $1/2 \cdot VLCD$ ，V3 的值和 V2 的值相等。

Segment 12~Segment 23 引脚与 P0.10~P0.15 和 P1.0~P1.5 引脚共用，这些引脚作为 LCD 引脚使用时，SEG1SEL 位须置 1，PCB Layout 时，VLCD2 也应该连接到 VLCD1 引脚。

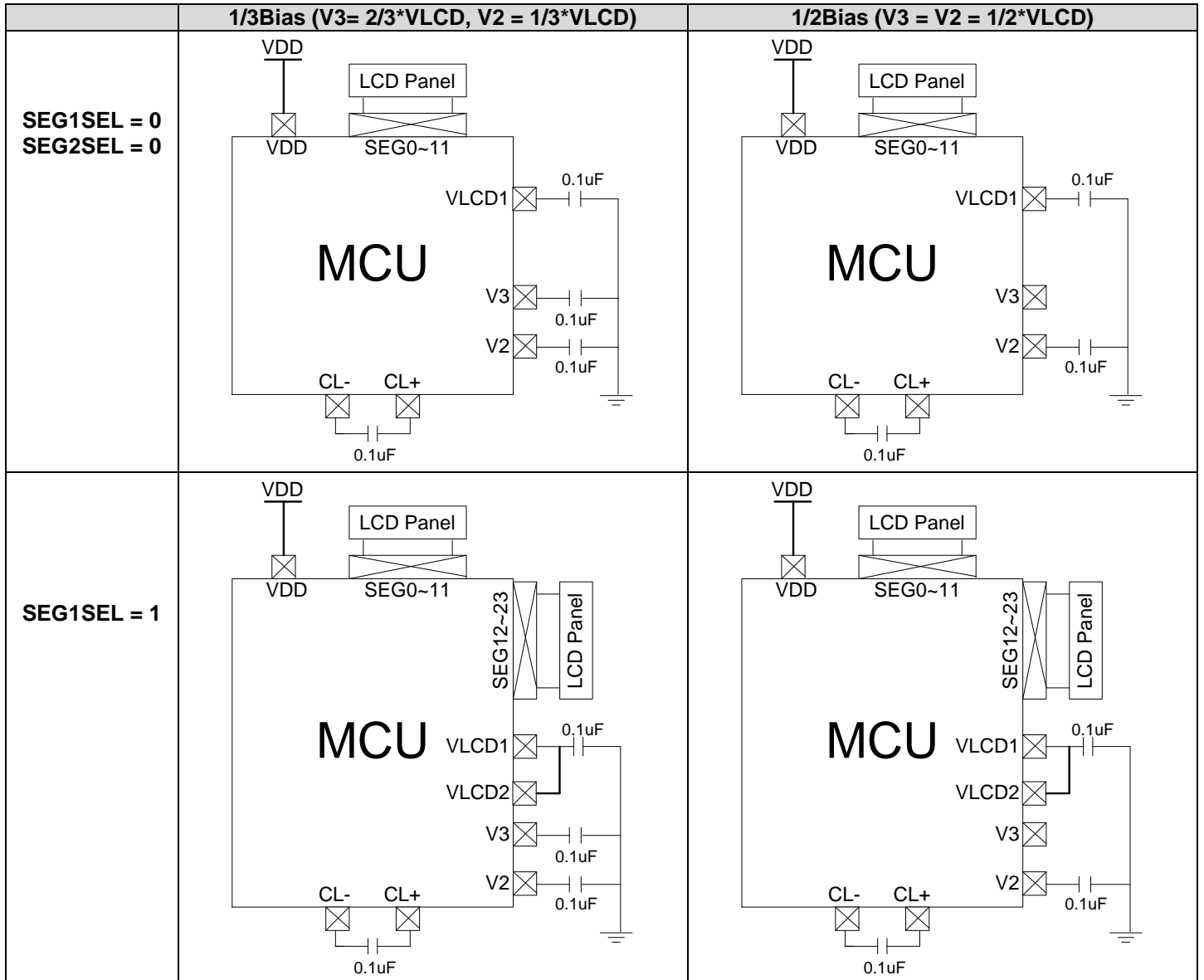
Segment 24~Segment 31 引脚与 P0.0~P0.7 引脚共用，这些引脚作为 LCD 引脚使用时，SEG2SEL 位须置 1，PCB Layout 时，VLCD3 也应该连接到 VLCD1 引脚。

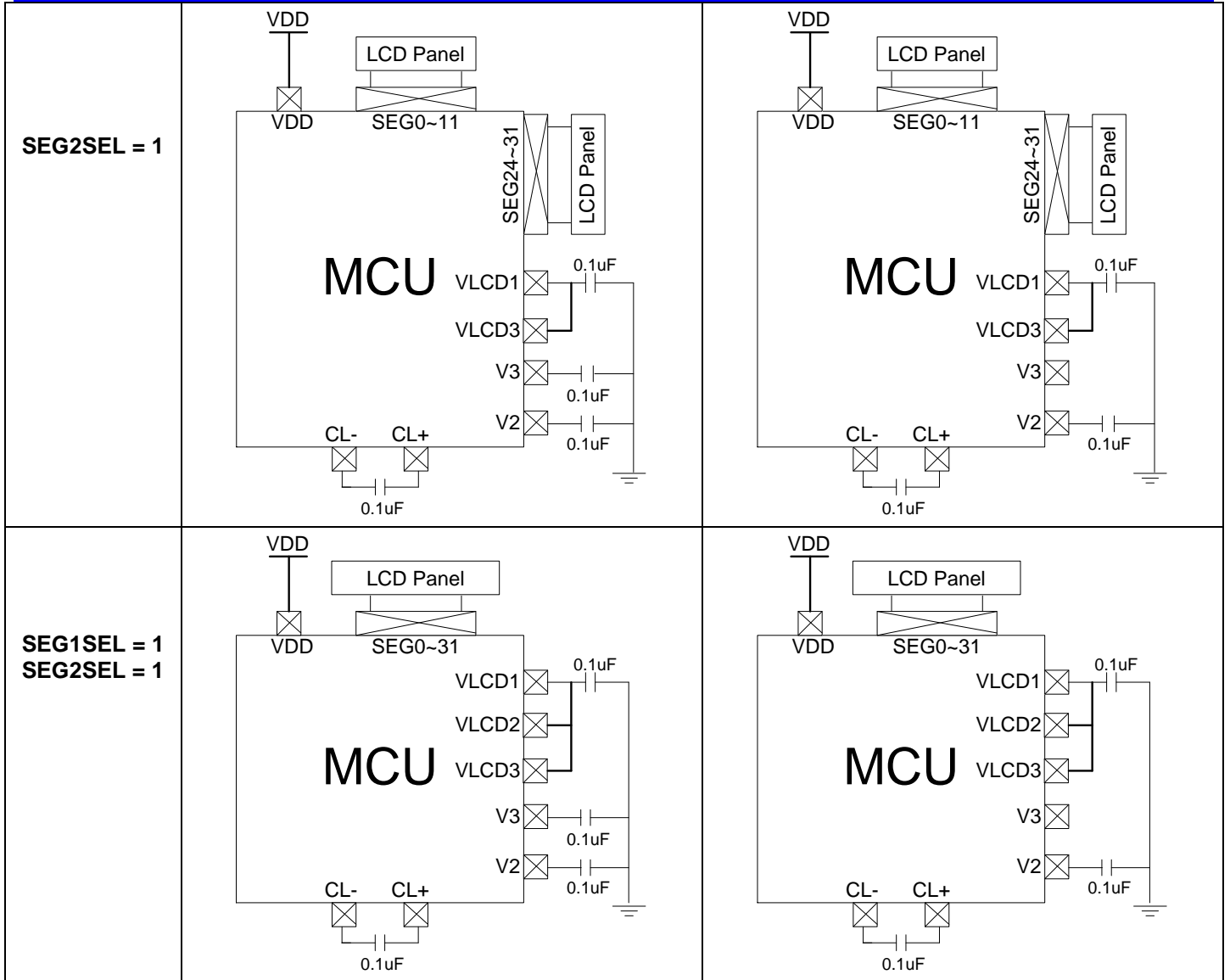
* 注：V2 的电压为 charge-pump 的输出电压。

16.7.1 4C-TYPE

* 注:

- 1、VLCD 的 Charge-pump 电压为 2.7V~5.0V
- 2、1/2 偏压时，VDD 应大于 V3。
- 3、1/3 偏压时，若 VLCD 大于 3.6V，VDD 应大于 V3-0.3V。
- 4、在 CL+ 和 CL- 之间连接一个 0.1uF 或者 0.47uF 的电容器，用户可根据 LCD 的尺寸来调节电容器的值。
- 5、为稳定电压，必须连接一个 0.1uF 的电容器到 VLCD1/V3/V2 引脚，用户可根据 LCD 的尺寸来调节电容器的值。除此之外，还要尽可能的靠近 VLCD1/V3/V2 引脚。

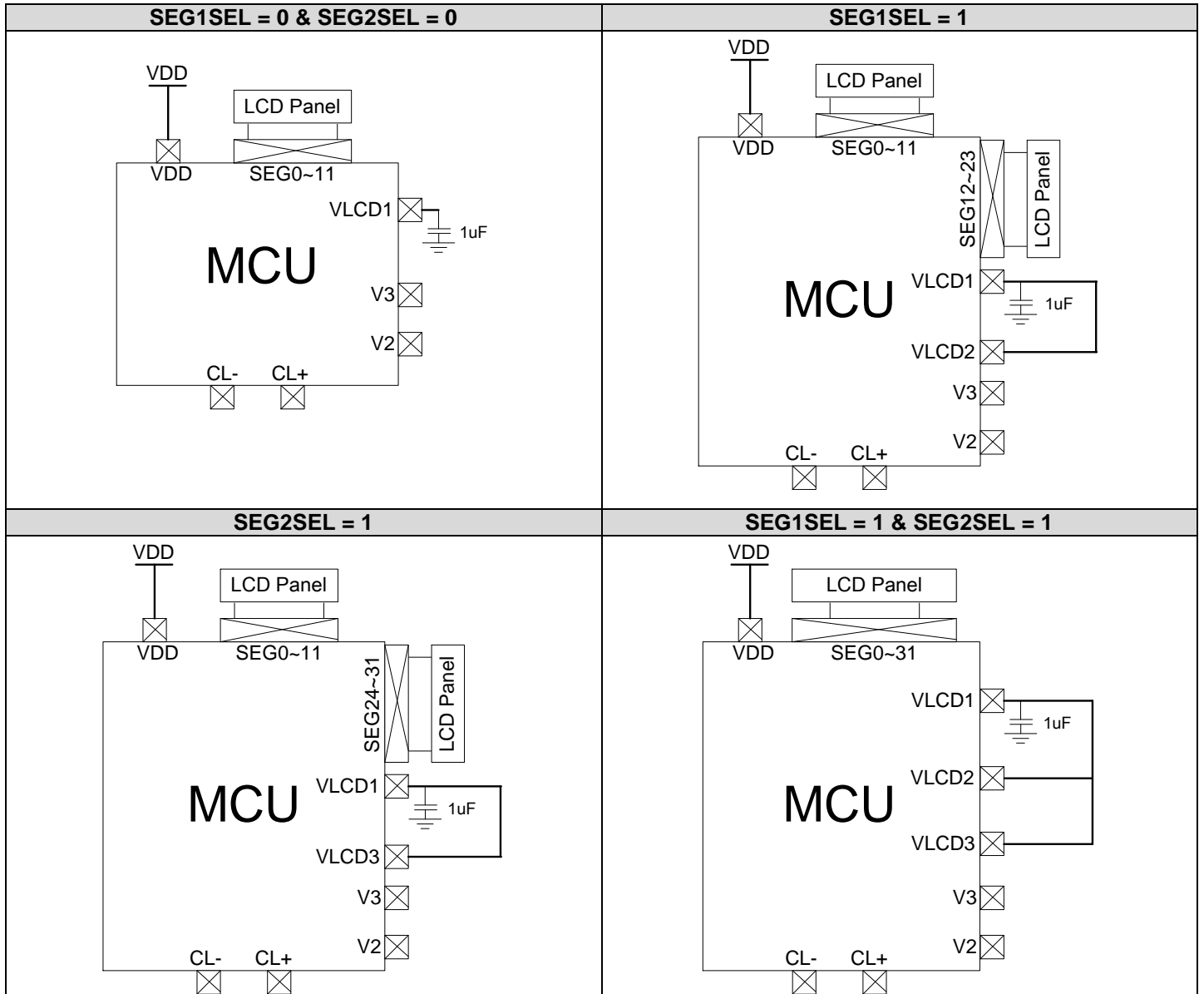




16.7.2 1C-Type

* 注:

- 1、charge-pump 的最大电压为 3.4V。
- 2、不需在 CL+和 CL-引脚间连接电容。
- 3、不需在 V3/V2 引脚间连接电容。
- 4、为稳定电压，必须连接一个 1uF 的电容到 VLCD1 引脚，用户可根据 LCD 的尺寸来调节电容的值。除此之外，还要尽可能的靠近 VLCD1 引脚。



16.8 LCD 显示分配图 LAY MEMORY MAP

Bit	LCD_SEGM3	LCD_SEGM2	LCD_SEGM1	LCD_SEGM0	
31	SEG31	SEG23	SEG15	SEG7	COM3
30					COM2
29					COM1
28					COM0
27	SEG30	SEG22	SEG14	SEG6	COM3
26					COM2
25					COM1
24					COM0
23	SEG29	SEG21	SEG13	SEG5	COM3
22					COM2
21					COM1
20					COM0
19	SEG28	SEG20	SEG12	SEG4	COM3
18					COM2
17					COM1
16					COM0
15	SEG27	SEG19	SEG11	SEG3	COM3
14					COM2
13					COM1
12					COM0
11	SEG26	SEG18	SEG10	SEG2	COM3
10					COM2
9					COM1
8					COM0
7	SEG25	SEG17	SEG9	SEG1	COM3
6					COM2
5					COM1
4					COM0
3	SEG24	SEG16	SEG8	SEG0	COM3
2					COM2
1					COM1
0					COM0

16.9 LCD 寄存器

基地址：0x4003 4000

16.9.1 LCD 控制寄存器 (LCD_CTRL)

地址偏移量：0x00

复位值：0x1000 0302

*** 注：**

- 1、Segment 24~Segment 31 引脚与 P0.0~P0.7 共用。
 - 这些引脚作为 GPIO 引脚使用时，SEG2SEL 位须置 0，PCB Layout 时，VDD3 必须要连接到 VDD。
 - 这些引脚作为 LCD 引脚使用时，SEG2SEL 位须置 1，须参考 R 型 LCD 应用电路和 C 型 LCD 应用电路。
- 2、Segment 12~Segment 23 引脚与 P0.10~P0.15 和 P1.0~P1.5 共用。
 - 这些引脚作为 GPIO 引脚使用时，SEG1SEL 位须置 0，PCB Layout 时，VDD2 必须要连接到 VDD。
 - 这些引脚作为 LCD 引脚使用时，SEG1SEL 位须置 1，须参考 R 型 LCD 应用电路和 C 型 LCD 应用电路。

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R/W	0
29:28	DRIVEP[1:0]	LCD 面板驱动能力。 00: 强 (大型面板)； 01: 中 (中型面板)； 10: 保留； 11: 低 (小型面板)。	R/W	01b
27:12	Reserved		R/W	0
11	LCDRATE	LCD 时钟 rate (LCD_PCLK)。 0: LCD 时钟源/ 64； 1: LCD 时钟源/ 128。	R/W	0
10	LCDCLK	LCD 时钟源选择控制位。 0: ILRC； 1: ELS XTAL。	R/W	0
9:8	DUTY[1:0]	占空比选择位。 00: 保留； 01: 1/2 占空比, HW 分配 CM0~1 作为 LCD 引脚, R 型 LCD 模式下, P1.13, P1.12 仍作为 GPIO 引脚使用。 10: 1/3 占空比, HW 分配 CM0~2 作为 LCD 引脚, R 型 LCD 模式下, P1.13 仍作为 GPIO 引脚使用。 11: 1/4 占空比, HW 分配 CM0~3 作为 LCD 引脚。	R/W	11b
7	Reserved		R/W	0
6	SEGSEL2	SEG24~31 使能位。 0: 禁止, SEG24~31 为 GPIO 引脚； 1: 使能, HW 分配 SEG24~31 为 LCD 引脚。	R/W	0
5	SEGSEL1	SEG12~23 使能位。 0: 禁止, SEG12~23 为 GPIO 引脚； 1: 使能, HW 分配 SEG12~23 为 LCD 引脚。	R/W	0
4	BIAS	LCD 偏压选择位。 0: 1/3 偏压； 1: 1/2 偏压。	R/W	0
3:2	LCDDTYPE[1:0]	LCD 类型控制位。 00: R-Type； 01: 4C Type, HW 分配 CL+, CL-为 LCD 引脚； 10: 1C Type, HW 分配 CL+, CL-为 LCD 引脚； 11: 保留。	R/W	00b
1	ITB	用于内部测试, 仅允许设置为 1。	R/W	1
0	LCDENB	LCD 驱动使能位。 0: 禁止； 1: 使能, HW 分配 SEG0~11, V3, V2 为 LCD 引脚。	R/W	0

16.9.2 LCD 控制寄存器 1 (LCD_CTRL1)

地址偏移量: 0x04

复位值: 0x1000 0000

Bit	Name	Description	Attribute	Reset
31:29	Reserved		R/W	0
28	ITB	用于内部测试, 仅允许设置为 1。	R/W	1
27:3	Reserved		R/W	0
2:1	REF[1:0]	LCD 偏压分压电阻选择位。 00: 400K; 01: 200K; 10: 100K; 11: 35K。	R/W	00b
0	LCDBNK	LCD 空白控制位。 0: 普通显示; 1: 关闭所有 LCD 的点。	R/W	0

16.9.3 LCD C 型控制寄存器 1 (LCD_CCTRL1)

地址偏移量: 0x08

复位值: 0x6002 0003

LCD_CCTRL1 对 C 型 LCD 有效, charge pump 的时钟源由 LCD_CTRL 寄存器的 LCDCLK 位控制。

LCDCLK	Charge-pump 时钟源
0	ILRC 32KHz
1	ELS XTAL 32.768KHz

Bit	Name	Description	Attribute	Reset
31:30	Reserved	仅允许设置为 01。	R/W	01b
29:28	IT1[1:0]	用于内部测试, 仅允许设置为 00。	R/W	10b
27:24	IT2[3:0]	用于内部测试, 仅允许设置为 0100b。	R/W	00b
25:4	Reserved		R/W	0
3:0	VCP[3:0]	C 型 LCD 输出电压。	R/W	0011b

1C Type				
VCP[3:0]	VLCD	1/3 Bias		1/2 Bias
		V2	V3	V2 = V3
0000	2.70V	0.90V	1.80V	1.35V
0001	2.80V	0.94V	1.87V	1.40V
0010	2.90V	0.98V	1.96V	1.45V
0011	3.00V	1.00V	2.00V	1.50V
0100	3.10V	1.04V	2.08V	1.53V
0101	3.20V	1.08V	2.14V	1.57V
0110	3.30V	1.10V	2.20V	1.61V
0111	3.40V	1.14V	2.28V	1.66V
Reserved	N/A	N/A	N/A	N/A

4C Type				
VCP[3:0]	VLCD	1/3 Bias		1/2 Bias
		V2	V3	V2 = V3
0000	2.70V	0.90V	1.80V	1.35V
0001	2.80V	0.93V	1.86V	1.40V
0010	2.90V	0.96V	1.93V	1.45V
0011	3.00V	1.00V	2.00V	1.50V
0100	3.06V	1.02V	2.04V	1.53V
0101	3.14V	1.05V	2.10V	1.57V
0110	3.20V	1.07V	2.14V	1.61V
0111	3.30V	1.10V	2.20V	1.66V
1000	3.40V	1.13V	2.26V	1.70V
1001	3.60V	1.20V	2.40V	1.80V
1010	3.80V	1.27V	2.54V	1.90V
1011	4.00V	1.33V	2.67V	2.00V
1100	4.20V	1.40V	2.80V	2.12V
1101	4.40V	1.49V	2.98V	2.23V
1110	4.70V	1.57V	3.14V	2.35V
1111	5.00V	1.66V	3.32V	2.50V

16.9.4 LCD C 型控制寄存器 2 (LCD_CTRL2)

地址偏移量: 0x0C

LCD_CTRL2 寄存器用于内部测试, 仅允许设置为 0x00000004。

16.9.5 LCD 帧计数器控制寄存器 (LCD_FCC)

地址偏移量: Address offset: 0x10

复位值: 0x0000 0002

FCENB=1 时, 帧计数器 (FC) 从 0x00 开始接收, 帧升级时加 1。当计数器的值到达 FCT[5:0]时, FC 由 HW 复位为 0, LCD 帧中断标志位也变为 0。若使能 LCD 帧中断 (FCIE=1), 则产生一个 LCD 帧中断并发送到中断控制器中。

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
7	FCIE	LCD 帧中断使能位。 0: 禁止; 1: 使能。	R/W	0
6:1	FCT[5:0]	LCD 帧计数器临界值。	R/W	00001b
0	FCENB	LCD 帧计数器使能位。 0: 禁止; 1: 使能。	R/W	0

16.9.6 LCD Raw 中断状态寄存器 (LCD_RIS)

地址偏移量: 0x14

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	FCIF	LCD 帧中断标志位。 0: 读 → 无中断 写 → 写入 0 将该位清零, 若 FCIE=1, 中断复位; 1: 请求 FC 中断。	R/W	0

16.9.7 LCD SEG 存储器寄存器 0 (LCD_SEGM0)

地址偏移量: 0x20

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:28	SEG7[3:0]	SEG7 data for COM0~COM3	R/W	0
27:24	SEG6[3:0]	SEG6 data for COM0~COM3	R/W	0
23:20	SEG5[3:0]	SEG5 data for COM0~COM3	R/W	0
19:16	SEG4[3:0]	SEG4 data for COM0~COM3	R/W	0
15:12	SEG3[3:0]	SEG3 data for COM0~COM3	R/W	0
11:8	SEG2[3:0]	SEG2 data for COM0~COM3	R/W	0
7:4	SEG1[3:0]	SEG1 data for COM0~COM3	R/W	0
3:0	SEG0[3:0]	SEG0 data for COM0~COM3	R/W	0

16.9.8 LCD SEG 存储器寄存器 1 (LCD_SEG1)

地址偏移量: 0x24

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:28	SEG15[3:0]	SEG15 data for COM0~COM3	R/W	0
27:24	SEG14[3:0]	SEG14 data for COM0~COM3	R/W	0
23:20	SEG13[3:0]	SEG13 data for COM0~COM3	R/W	0
19:16	SEG12[3:0]	SEG12 data for COM0~COM3	R/W	0
15:12	SEG11[3:0]	SEG11 data for COM0~COM3	R/W	0
11:8	SEG10[3:0]	SEG10 data for COM0~COM3	R/W	0
7:4	SEG9[3:0]	SEG9 data for COM0~COM3	R/W	0
3:0	SEG8[3:0]	SEG8 data for COM0~COM3	R/W	0

16.9.9 LCD SEG 存储器寄存器 2 (LCD_SEG2)

地址偏移量: 0x28

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:28	SEG23[3:0]	SEG23 data for COM0~COM3	R/W	0
27:24	SEG22[3:0]	SEG22 data for COM0~COM3	R/W	0
23:20	SEG21[3:0]	SEG21 data for COM0~COM3	R/W	0
19:16	SEG20[3:0]	SEG20 data for COM0~COM3	R/W	0
15:12	SEG19[3:0]	SEG19 data for COM0~COM3	R/W	0
11:8	SEG18[3:0]	SEG18 data for COM0~COM3	R/W	0
7:4	SEG17[3:0]	SEG17 data for COM0~COM3	R/W	0
3:0	SEG16[3:0]	SEG16 data for COM0~COM3	R/W	0

16.9.10 LCD SEG 存储器寄存器 3 (LCD_SEG3)

地址偏移量: 0x2C

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:28	SEG31[3:0]	SEG31 data for COM0~COM3	R/W	0
27:24	SEG30[3:0]	SEG30 data for COM0~COM3	R/W	0
23:20	SEG29[3:0]	SEG29 data for COM0~COM3	R/W	0
19:16	SEG28[3:0]	SEG28 data for COM0~COM3	R/W	0
15:12	SEG27[3:0]	SEG27 data for COM0~COM3	R/W	0
11:8	SEG26[3:0]	SEG26 data for COM0~COM3	R/W	0
7:4	SEG25[3:0]	SEG25 data for COM0~COM3	R/W	0
3:0	SEG24[3:0]	SEG24 data for COM0~COM3	R/W	0

17 USB FS 设备接口

17.1 概述

USB 做为 PC 与外围通信的一种接口标准，以其快速、双向、同步、低成本、热插拔等特点大大满足了 PC 平台未来发展的需要。SONiX USB 控制芯片可将诸如鼠标、键盘、摇杆和游戏垫等计算机外设的人机交互推向最优化。

USB 遵循的规范：

- 遵循USB规范V2.0
- 支持1个全速USB设备地址
- 支持1个控制端点和6个可配置端点（用于实时/中断/Bulk发送）
- 集成USB收发器
- 5V转成3.3V输出regulator，连接1.5K的内部上拉电阻到D+

* 注：USB 有效模式下（USB 挂起除外），HCLK 至少必须 $\geq 3\text{MHz}$ 。

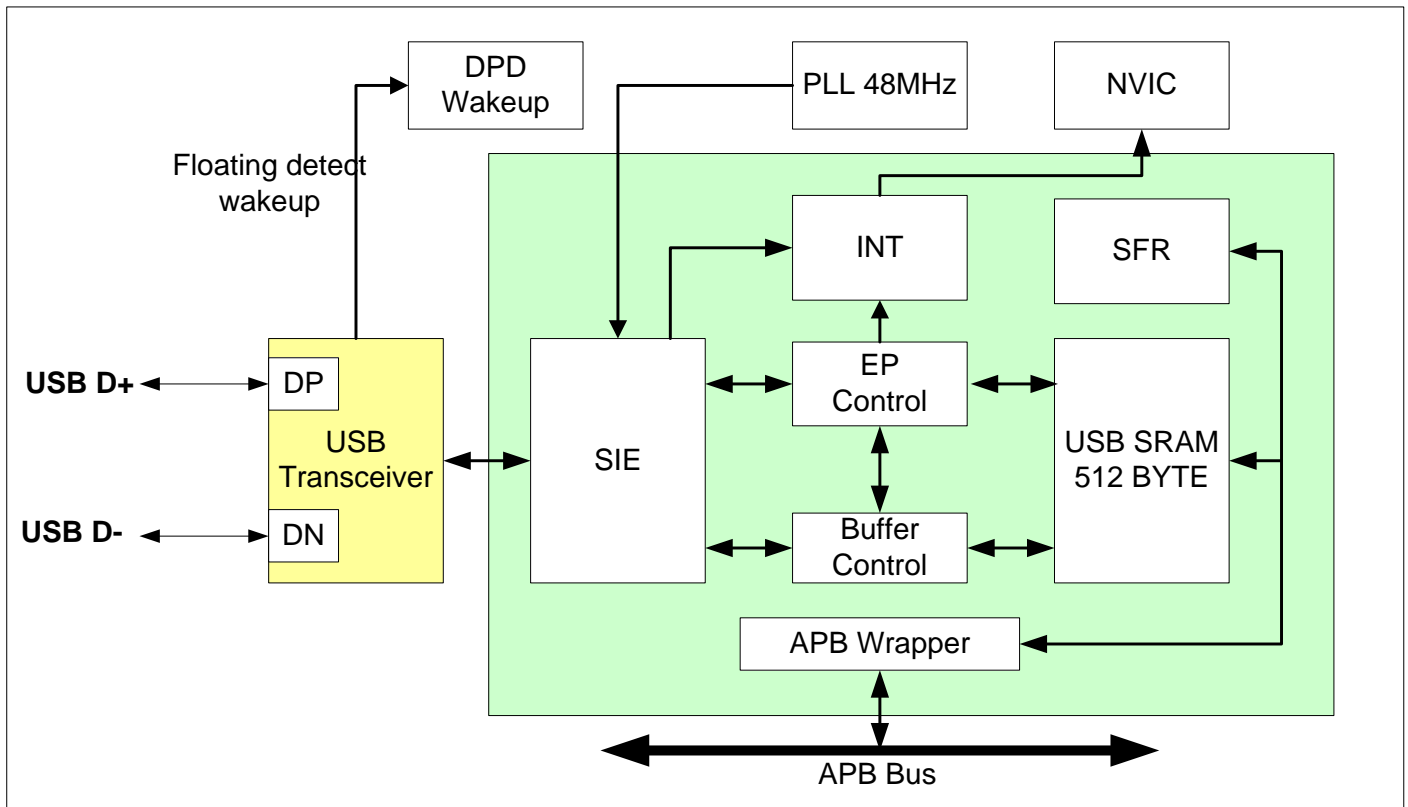
17.2 特性

- 符合 USB 规范 V2.0
- 支持 1 个全速 USB 设备地址
- 支持 1 个控制端点，最大数据包尺寸为 8 字节，16 字节，32 字节或者 64 字节
- 支持 6 个可配置端点，用于实时/中断/Bulk 发送
- 支持 USB SRAM 尺寸为 512 字节，与所有的 7 个端点共用
- 可灵活配置不同端点 FIFO 的地址偏移量（端点 0 除外）
- 5V 转 3.3V 的 regulator 上拉到内部 1.5K 电阻
- 集成 USB 收发器

17.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
DP	I/O	USB 差分信号 D+	N/A
DN	I/O	USB 差分信号 D-	N/A

17.4 框图

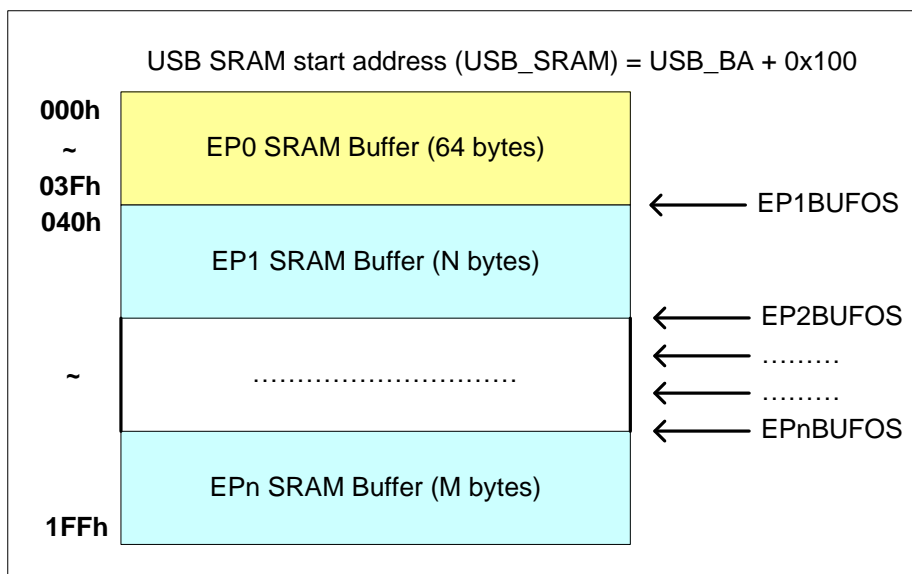


17.5 USB SRAM 访问

USB SRAM 512 字节与 EP0~EP6, 共 7 个端点共用。USB SRAM 的起始地址 (USB_SRAM) 为 (USB_BA + 0x100)。EP0 SRAM 缓存器的起始地址为 000H~03FH 之间, EP1~EP6 SRAM 缓存器的起始地址由 EPnBUFOS 寄存器决定。

访问 USB SRAM 的原则如下所示:

- 每个 EPnBUFOS 都设置为字对齐的, 2 LSB 位等于 0。
- EPn SRAM 缓存器的最大长度由用户定义, 但每个端点都应该有自己的 EPn SRAM 缓存器, 且相互间不重叠。



17.6 USB 机构

USB 机构实现单片机与 USB 主机的通信，硬件可独立地完成如下 USB 动作：

USB 机构将完成：

- 对接收到的数据译码，对将要发送的数据编码；
- CRC 的校验和产生由硬件完成。若 CRC 错误，硬件将不给主机任何回应；
- 硬件自动更新发送数据同步校准位；
- USB 控制寄存器发送相应的 ACK/NAK/STALL 握手信号；
- 不同类型令牌包（SETUP、IN、OUT）的识别。接收到有效令牌包后，对相应寄存器的状态位置“1”；
- 填充校验位。
- 地址检查。丢掉无效地址的传输。
- 端点检查。检查来自主机的请求后，对寄存器的相关位进行设置。

固件完成以下功能：

- 通过接收 USB 设备请求上传相对应的列举数据；
- 填写/清空 FIFO。
- 挂起/唤醒功能。
- 设备端唤醒功能；
- USB 传输中，决定采用何种中断请求。

17.7 USB 中断

USB 功能接收 USB 主机的命令并产生相关的中断，程序计数器跳转到中断向量地址，此时要求固件检查 USB 的状态位以了解产生的是哪一种中断。

在以下情况下会产生 USB 中断：

- 端点 0 接收到设置令牌包（SETUP token）；
- 成功完成输入事务后，设备会接收到一个 ACK 应答信号。
- 端点在 ACK OUT 模式下，接收到数据后产生中断；
- USB 主机送出 USB 挂起请求；
- USB 复位；
- USB 处理完成后端点中断；
- 使能 NAK 中断时 NAK 握手。

17.8 USB 枚举

典型的 USB 枚举流程如下：

- USB 主机发送 Setup 包后，再发送 DATA 包到地址 0，请求设备发送设备描述符。
- 固件收到请求，从 ROM 表中找到设备描述符。
- USB 主机发送输入控制时序，程序找到 USB 中的设备描述符通过内置 USB SRAM 回传给主机。
- USB 主机接收到描述符后，发送 SETUP 包和 DATA 包给地址 0 以给设备分配新 USB 地址。
- 无数据的状态控制阶段结束后，程序会将新地址存储在 USB 设备地址寄存器中。
- 主机利用新的 USB 地址请求发送设备描述符；
- 固件解码得到请求命令，并从程序存储列表中找到设备描述符；
- 主机电脑发送读取命令，固件将设备描述符发送到 USB 总线上；
- 主机发出控制读命令，请求配置和说明描述符；
- 一旦设备接收到一个 Set Configuration 请求后，USB 功能开始使用；
- 程序要完成端点 0~N 的各交易操作。

17.9 USB 寄存器

基地址: 0x4005 C000

R: 只读, W: 只写, R/W: 读/写

Register	Offset	R/W	Description	Reset Value
USB_INTEN	0x00	R/W	USB 中断使能寄存器	0x0000_0000
USB_INSTS	0x04	R	USB 中断事件状态寄存器	0x0000_0000
USB_INSTSC	0x08	W	USB 中断事件状态清零寄存器	0x0000_0000
USB_ADDR	0x0C	R/W	USB 设备地址寄存器	0x0000_0000
USB_CFG	0x10	R/W	USB 配置寄存器	0x0000_0000
USB_SGCTL	0x14	R/W	USB 信号控制寄存器	0x0000_0000
USB_EP0CTL	0x18	R/W	USB Endpoint 0 控制寄存器	0x0000_0000
USB_EP1CTL	0x1C	R/W	USB Endpoint 1 控制寄存器	0x0000_0000
USB_EP2CTL	0x20	R/W	USB Endpoint 2 控制寄存器	0x0000_0000
USB_EP3CTL	0x24	R/W	USB Endpoint 3 控制寄存器	0x0000_0000
USB_EP4CTL	0x28	R/W	USB Endpoint 4 控制寄存器	0x0000_0000
USB_EP5CTL	0x2C	R/W	USB Endpoint 5 控制寄存器	0x0000_0000
USB_EP6CTL	0x30	R/W	USB Endpoint 6 控制寄存器	0x0000_0000
USB_EPTOGGLE	0x3C	R/W	USB Endpoint 数据 Toggle 寄存器	0x0000_003F
USB_EP1BUFOS	0x48	R/W	USB Endpoint 1 缓存器偏移寄存器	0x0000_0000
USB_EP2BUFOS	0x4C	R/W	USB Endpoint 2 缓存器偏移寄存器	0x0000_0000
USB_EP3BUFOS	0x50	R/W	USB Endpoint 3 缓存器偏移寄存器	0x0000_0000
USB_EP4BUFOS	0x54	R/W	USB Endpoint 4 缓存器偏移寄存器	0x0000_0000
USB_EP5BUFOS	0x58	R/W	USB Endpoint 5 缓存器偏移寄存器	0x0000_0000
USB_EP6BUFOS	0x5C	R/W	USB Endpoint 6 缓存器偏移寄存器	0x0000_0000
USB_FRMNO	0x60	R	USB 帧数字寄存器	0x0000_0000
USB_PHYPRM	0x64	R/W	USB PHY 参数寄存器	0x0000_0000
USB_SRAM	0x100	R/W	USB 512 byte SRAM	Undefined

17.9.1 USB 中断使能寄存器 (USB_INTEN)

地址偏移量: 0x00

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	BUS_IE	总线事件中断使能位。 0: 禁止; 1: 使能, 包括 BUS_RESET、BUS_SUSPEND 和 BUS_RESUME 触发 USB 中断。	R/W	0
30	USB_SOF_IE	USB SOF 中断使能位。 0: 禁止; 1: 使能。	R/W	0
29	USB_IE	USB 事件中断使能位。 0: 禁止; 1: 使能, EP1~EP6 的 NAK 触发的 USB 中断除外。	R/W	0
28:6	Reserved		R	0
5	EP6_NAK_EN	EP6 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
4	EP5_NAK_EN	EP5 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
3	EP4_NAK_EN	EP4 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
2	EP3_NAK_EN	EP3 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
1	EP2_NAK_EN	EP2 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
0	EP1_NAK_EN	EP1 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0

17.9.2 USB 中断事件状态寄存器 (USB_INSTS)

地址偏移量: 0x04

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	BUS_RESET	USB 总线复位信号标志位。 0: 没有检测到总线复位信号; 1: 检测到总线复位信号, 写入 1 到 USB_INSTSC[31]清零。	R	0
30	BUS_SUSPEND	USB 总线挂起信号标志位。一旦 USB 总线离开挂起状态, 该位由 H/2 自动清零, 且不能由固件清零。 0: 没有检测到总线挂起; 1: 检测到总线挂起。	R	0
29	BUS_RESUME	USB 总线恢复信号标志位。 0: 没有检测到总线恢复信号; 1: 检测到挂起模式下的总线恢复信号, 写入 1 到 USB_INSTSC[29]清零。	R	0
28:27	Reserved		-	0
26	USB_SOF	USB SOF 包接收标志位。 0: 没有 USB SOF 包; 1: 接收到 USB SOF 包, 写入 1 到 USB_INSTSC[26]清零。	R	0
25	BUS_WAKEUP	USB 总线唤醒标志位。 0: 没有从挂起模式唤醒; 1: 从挂起模式唤醒, 写入 1 到 USB_INSTSC[25]清零。	R	0
24	EP0_PRESETUP	EP0 Setup token 包标志位。该标志位不触发 USB 中断。 0: 没有 EP0 Setup token 包; 1: 接收到 EP0 Setup token 包, 写入 1 到 USB_INSTSC[24]清零。	R	0
23	EP0_SETUP	EP0 Setup transaction 标志位。 0: 没有 EP0 Setup transaction; 1: 完成 EP0 Setup transaction, 写入 1 到 USB_INSTSC[23]清零。	R	0
22	EP0_IN	EP0 IN ACK transaction 标志位。 0: 没有 EP0 IN ACK Transaction; 1: 完成 EP0 IN ACK transaction, 写入 1 到 USB_INSTSC[22]清零。	R	0
21	EP0_OUT	EP0 OUT ACK transaction 标志位。 0: 没有 EP0 OUT ACK transaction; 1: 完成 EP0 OUT ACK transaction, 写入 1 到 USB_INSTSC[21]清零。	R	0
20	EP0_IN_STALL	EP0 IN STALL transaction 标志位。 0: 没有 EP0 IN STALL transaction; 1: 完成 EP0 IN STALL transaction, 写入 1 到 USB_INSTSC[20]清零。	R	0
19	EP0_OUT_STALL	EP0 OUT STALL transaction 标志位。 0: 没有 EP0 OUT STALL transaction; 1: 完成 EP0 OUT STALL transaction, 写入 1 到 USB_INSTSC[19]清零。	R	0
18	ERR_SETUP	接收到错误的 setup 数据标志位, 该位不触发 USB 中断。 0: 接收到普通的 8 字节 Setup DATA0; 1: Setup 数据不是 8 字节或者不是 DATA0, 写入 1 到 USB_INSTSC[18]清零。	R	0
17	ERR_TIMEOUT	超时状态标志位, 该位不触发 USB 中断。 0: 没有超时; 1: 发送 IN 数据包后, 主机 ACK 响应超时, 写入 1 到 USB_INSTSC[17]清零。	R	0
16:14	Reserved		-	0
13	EP6_ACK	EP6 ACK transaction 标志位, 若 USB_CFG[13] = 1, 一旦完成 EP6 ISO transaction, H/W 会设置 EP6_ACK 为 1。 0: 没有 EP6 ACK transaction; 1: 完成 EP6 ACK transaction, 写入 1 到 USB_INSTSC[13]清零。	R	0
12	EP5_ACK	EP5 ACK transaction 标志位, 若 USB_CFG[12] = 1, 一旦完成 EP5 ISO transaction, H/W 会设置 EP5_ACK 为 1。 0: 没有 EP5 ACK transaction; 1: 完成 EP5 ACK transaction, 写入 1 到 USB_INSTSC[12]清零。	R	0
11	EP4_ACK	EP4 ACK transaction 标志位, 若 USB_CFG[11] = 1, 一旦完成 EP4 ISO transaction, H/W 会设置 EP4_ACK 为 1。 0: 没有 EP4 ACK transaction; 1: 完成 EP4 ACK transaction, 写入 1 到 USB_INSTSC[11]清零。	R	0

10	EP3_ACK	EP3 ACK transaction 标志位, 若 USB_CFG[10] = 1, 一旦完成 EP3 ISO transaction, H/W 会设置 EP3_ACK 为 1。 0: 没有 EP3 ACK transaction; 1: 完成 EP3 ACK transaction, 写入 1 到 USB_INSTSC[10]清零。	R	0
9	EP2_ACK	EP2 ACK transaction 标志位, 若 USB_CFG[9] = 1, 一旦完成 EP2 ISO transaction, H/W 会设置 EP2_ACK 为 1。 0: 没有 EP2 ACK transaction; 1: 完成 EP2 ACK transaction, 写入 1 到 USB_INSTSC[9]清零。	R	0
8	EP1_ACK	EP1 ACK transaction 标志位, 若 USB_CFG[8] = 1, 一旦完成 EP1 ISO transaction, H/W 会设置 EP1_ACK 为 1。 0: 没有 EP1 ACK transaction; 1: 完成 EP1 ACK transaction, 写入 1 到 USB_INSTSC[8]清零。	R	0
7:6	Reserved		-	0
5	EP6_NAK	EP6 NAK transaction 标志位。 0: 没有 EP6 NAK transaction; 1: 完成 EP6 NAK transaction, 写入 1 到 USB_INSTSC[5]清零。	R	0
4	EP5_NAK	EP5 NAK transaction 标志位。 0: 没有 EP5 NAK transaction; 1: 完成 EP5 NAK transaction, 写入 1 到 USB_INSTSC[4]清零。	R	0
3	EP4_NAK	EP4 NAK transaction 标志位。 0: 没有 EP4 NAK transaction; 1: 完成 EP4 NAK transaction, 写入 1 到 USB_INSTSC[3]清零。	R	0
2	EP3_NAK	EP3 NAK transaction 标志位。 0: 没有 EP3 NAK transaction; 1: 完成 EP3 NAK transaction, 写入 1 到 USB_INSTSC[2]清零。	R	0
1	EP2_NAK	EP2 NAK transaction 标志位。 0: 没有 EP2 NAK transaction; 1: 完成 EP2 NAK transaction, 写入 1 到 USB_INSTSC[1]清零。	R	0
0	EP1_NAK	EP1 NAK transaction 标志位。 0: 没有 EP1 NAK transaction; 1: 完成 EP1 NAK transaction, 写入 1 到 USB_INSTSC[0]清零。	R	0

17.9.3 USB 中断事件状态清零寄存器 (USB_INSTSC)

地址偏移量: 0x08

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	BUS_RESETC	0: 没有影响; 1: BUS_RESET 位清零。	W	0
30	Reserved		R	-
29	BUS_RESUMEC	0: 没有影响; 1: BUS_RESUME 位清零。	W	0
28:27	Reserved		R	-
26	USB_SOFC	0: 没有影响; 1: USB_SOF 位清零。	W	0
25	BUS_WAKEUPC	0: 没有影响; 1: BUS_WAKEUP 位清零。	W	0
24	EP0_PRESETUPC	0: 没有影响; 1: EP0_PRESETUP 位清零。	W	0
23	EP0_SETUPC	0: 没有影响; 1: EP0_SETUP 位清零。	W	0
22	EP0_IN	0: 没有影响; 1: EP0_IN 位清零。	W	0
21	EP0_OUTC	0: 没有影响; 1: EP0_OUT 位清零。	W	0
20	EP0_IN_STALLC	0: 没有影响; 1: EP0_IN_STALL 位清零。	W	0
19	EP0_OUT_STALLC	0: 没有影响; 1: EP0_OUT_STALL 位清零。	W	0
18	ERR_SETUPC	0: 没有影响; 1: ERR_SETUP 位清零。	W	0
17	ERR_TIMEOUTC	0: 没有影响; 1: ERR_TIMEOUT 位清零。	W	0
16:14	Reserved		R	-
13	EP6_ACKC	0: 没有影响; 1: EP6_ACK 位清零。	W	0
12	EP5_ACKC	0: 没有影响; 1: EP5_ACK 位清零。	W	0
11	EP4_ACKC	0: 没有影响; 1: EP4_ACK 位清零。	W	0
10	EP3_ACKC	0: 没有影响; 1: EP3_ACK 位清零。	W	0
9	EP2_ACKC	0: 没有影响; 1: EP2_ACK 位清零。	W	0
8	EP1_ACKC	0: 没有影响; 1: EP1_ACK 位清零。	W	0
7:6	Reserved		R	-
5	EP6_NAKC	0: 没有影响; 1: EP6_NAK 位清零。	W	0
4	EP5_NAKC	0: 没有影响; 1: EP5_NAK 位清零。	W	0
3	EP4_NAKC	0: 没有影响; 1: EP4_NAK 位清零。	W	0
2	EP3_NAKC	0: 没有影响; 1: EP3_NAK 位清零。	W	0
1	EP2_NAKC	0: 没有影响; 1: EP2_NAK 位清零。	W	0
0	EP1_NAKC	0: 没有影响; 1: EP1_NAK 位清零。	W	0

17.9.4 USB 设备地址寄存器 (USB_ADDR)

地址偏移量: 0x0C

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	UADDR	USB 设备地址。	R/W	0

17.9.5 USB 配置寄存器 (USB_CFG)

地址偏移量: 0x10

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	VREG33_EN	内部 VREG33 输出功能控制位, 禁止 VREG33_EN 时, VREG33 切换为 IC_VDD。 0: 禁止; 1: 使能。	R/W	0
30	PHY_EN	PHY 收发器功能控制位。进入睡眠/深度睡眠后, 自动禁止 PHY。 0: 禁止; 1: 使能。	R/W	0
29	DPPU_EN	内部 D+ 1.5k 上拉电阻功能控制位 0: 禁止; 1: 使能。	R/W	0
28	SIE_EN	USB 串行接口 engine 使能控制位。 0: 禁止; 1: 使能。	R/W	0
27	ESD_EN	USB ESD 保护使能控制位。 0: 禁止; 1: 使能。	R/W	0
26	FLTDET_PUEN	D+和 D-之间的内部高上拉电阻使能控制位。 0: 禁止; 1: 使能。	R/W	0
25	USB_RAM_EN	USB 512-byte RAM 功能控制位。 0: 禁止; 1: 使能。	R/W	0
24	VREG33DIS_EN	使能 VREG33 放电 0: 禁止; 1: 使能。	R/W	0
23:14	Reserved		R	0
13	EP6_ISO	EP6 ISO 模式设置位。 0: 使能中断/bulk 模式; 1: 使能 ISO 模式。	R/W	0
12	EP5_ISO	EP5 ISO 模式设置位。 0: 使能中断/bulk 模式; 1: 使能 ISO 模式。	R/W	0
11	EP4_ISO	EP4 ISO 模式设置位。 0: 使能中断/bulk 模式; 1: 使能 ISO 模式。	R/W	0
10	EP3_ISO	EP3 ISO 模式设置位。 0: 使能中断/bulk 模式; 1: 使能 ISO 模式。	R/W	0
9	EP2_ISO	EP2 ISO 模式设置位。 0: 使能中断/bulk 模式; 1: 使能 ISO 模式。	R/W	0
8	EP1_ISO	EP1 ISO 模式设置位。 0: 使能中断/bulk 模式; 1: 使能 ISO 模式。	R/W	0
7:6	Reserved		R	0

5	EP6_DIR	EP6 IN/OUT 方向设置位。 0: 只与 IN token 包握手; 1: 只与 OUT token 包握手。	R/W	0
4	EP5_DIR	EP5 IN/OUT 方向设置位。 0: 只与 IN token 包握手; 1: 只与 OUT token 包握手。	R/W	0
3	EP4_DIR	EP4 IN/OUT 方向设置位。 0: 只与 IN token 包握手; 1: 只与 OUT token 包握手。	R/W	0
2	EP3_DIR	EP3 IN/OUT 方向设置位。 0: 只与 IN token 包握手; 1: 只与 OUT token 包握手。	R/W	0
1	EP2_DIR	EP2 IN/OUT 方向设置位。 0: 只与 IN token 包握手; 1: 只与 OUT token 包握手。	R/W	0
0	EP1_DIR	EP1 IN/OUT 方向设置位。 0: 只与 IN token 包握手; 1: 只与 OUT token 包握手。	R/W	0

17.9.6 USB 信号控制寄存器 (USB_SGCTL)

地址偏移量: 0x14

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	BUS_DRVEN	驱动 USB 总线使能控制位。 0: 没有驱动 USB 总线, 写操作对 BUS_DP 和 BUS_DN 没有影响; 1: 驱动 USB 总线, 通过设置 BUS_DP 和 BUS_DN 位来设置 D+/D- 的状态。	R/W	0
1	BUS_DP	USB D+ 状态位。 0: D+ 为低电平; 1: D+ 为高电平。	R/W	0
0	BUS_DN	USB D- 状态位。 0: D- 为低电平; 1: D- 为高电平。	R/W	0

17.9.7 USB EP0 控制寄存器 (USB_EP0CTL)

地址偏移量: 0x18

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	ENDP_EN	EP0 功能使能控制位。 0: 禁止, 与 EP0 SETUP/IN/OUT Token 没有握手; 1: 使能。	R/W	0
30:29	ENDP_STATE[1:0]	端点握手状态位。 00: NAK。 01: ACK。对于 IN transaction, 设备与 IN Transaction 握手 Data0/1; 对于 OUT transaction, 设备与 OUT token 和后面的 Data0/1 握手 ACK。 IN/OUT ACK Transaction 完成后, ENDP_STATE 自动返回到 NAK 状态。 10/11: INOUT_STALL。设备与 IN 或 OUT token 握手 STALL。USB Setup transaction 完成后, ENDP_STATE 自动返回到 NAK。	R/W	00
28	IN_STALL_EN	EP0 与 EP0 IN transaction 握手 STALL 使能控制位。 0: 禁止。 1: 使能。IN_STALL_EN 只在 EP0 IN token 使能有效。EP0 与 EP0 OUT transaction 的握手状态取决于 OUT_STALL_EN 和 ENDP_STATE 的设置。 USB Setup transaction 完成后, 该位自动清零。	R/W	0
27	OUT_STALL_EN	EP0 与 EP0 OUT transaction 握手 STALL 使能控制位。 0: 禁止; 1: 使能。OUT_STALL_EN 只在 EP0 OUT token 使能有效。EP0 与 EP0 IN	R/W	0

		transaction 的握手状态取决于 IN_STALL_EN 和 ENDP_STATE 的设置。 USB Setup transaction 完成后, 该位自动清零。		
26:9	Reserved	-	R	0
6:0	ENDP_CNT[6:0]	端点字节计数位。 对于 IN transaction, ENDP_CNT 显示上传给主机的字节数。IN transaction 的最大字节数取决于 USB Device Descriptor 的 MaximumPacketSize0 宣告, 不能超过 64 字节。 对于 OUT transaction, ENDP_CNT 显示从主机接收到的字节数。	R/W	0

17.9.8 USB Epn 控制寄存器 (USB_EPnCTL, n = 1 ~ 6)

地址偏移量: 0x1C, 0x20, 0x24, 0x28, 0x2C, 0x30

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	ENDP_EN	EPn 功能使能控制位。 0: 禁止, 没有与 EPn IN/OUT token 握手; 1: 使能。	R/W	0
30:29	ENDP_STATE[1:0]	端点握手状态位。 00: NAK。对于 IN direction usage, 设备与 IN token 握手 NAK; 对于 OUT direction usage, 设备与 OUT token 握手 NAK。 01: ACK。对于 IN direction usage, 设备与 IN token 握手 Data0/1; 对于 OUT direction usage, 设备与 OUT token 和后面的 Data0/1 握手 ACK。 IN/OUT ACK Transaction 完成后, ENDP_STATE 自动返回到 NAK 状态。 10/11: STALL。对于 IN direction usage, 设备与 IN token 握手 STALL。 对于 OUT direction usage, 设备与 OUT token 和后面的 Data0/1 握手 STALL。	R/W	0
28:9	Reserved	-	-	0
8:0	ENDP_CNT[8:0]	端点字节计数位。 对于 IN direction usage, ENDP_CNT 显示上传给主机的字节数。 对于 IN direction usage, ENDP_CNT 显示从主机接收到的字节数。	R/W	0

17.9.9 USB Epn 数据 Toggle 寄存器 (USB_EPTOGGLE)

地址偏移量: 0x3C

复位值: 0x0000 003F

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5	EP6_DATA01	0: 设置 EP6 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1
4	EP5_DATA01	0: 设置 EP5 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1
3	EP4_DATA01	0: 设置 EP4 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1
2	EP3_DATA01	0: 设置 EP3 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1
1	EP2_DATA01	0: 设置 EP2 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1
0	EP1_DATA01	0: 设置 EP1 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1

17.9.10 USB Epn 缓存器偏移寄存器 (USB_EPnBUFOS, n = 1 ~ 6)

地址偏移量: 0x48, 0x4C, 0x50, 0x54, 0x58, 0x5C

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8:2	OFFSET[6:0]	每个端点数据缓存器的偏移地址位。有效的偏移地址为: USB_SRAM 地址+ {EPnBUFOS[8:2], 2'b00} USB_SRAM 地址 = USB_BA + 0x100 EP0 的偏移地址固定为 USB_SRAM 地址。	R/W	0
1:0	Reserved		R	0

17.9.11 USB 帧数字寄存器 (USB_FRMNO)

地址偏移量: 0x60

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10:0	FRAME_NO[10:0]	Start-Of-Frame (SOF) 包的 11 位帧数字。接收到 SOF 包时, HW 自动更新该数字。	R	0

17.9.12 USB PHY 参数寄存器 (USB_PHYPRM)

地址偏移量: 0x64

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:26	PHY_PARAM[5:0]	USB PHY 参数值。建议设置为 0x20。	R/W	0
25:0	Reserved		R	0

18 FLASH

18.1 概述

SONiX 32 位单片机集成了在线编程 FLASH 存储器的特性，以方便存储升级的代码。可以经由 SONiX 32 位单片机编程接口或者应用代码灵活地对 Flash 存储器进行编程控制。SONiX 32 位单片机为用户提供了安全选项以防止未经授权的不安全信息存储到 Flash 存储器中。

- 在编程或者擦除 Flash 过程中，单片机停止工作，即使外设（定时器、WDT、I/O、PWM 等）仍在正常工作。
- 应该在编写 Flash 或者擦除 Flash 之前，若看门狗处于使能状态，则需要将其清零。
- 擦除动作会将 Flash 页中的所有位都置为逻辑 1。
- HW 会控制时钟，并自动将 RAM 中的数据删除开始编程，编程完成后，HW 释放时钟，单片机会执行下一条指令。

* 注：在编程和擦除操作时，系统时钟 Fcpu 必须等于小于 24MHz。

18.2 嵌入式 FLASH 存储器

Flash 存储器为 32 位宽度的存储器，可用于存储代码和数据常量，位于芯片内存的特定地址。

内置的高性能 Flash 存储器模块有以下主要特性：存储器构架，Flash 存储器包括用户 ROM 和引导 ROM。

用户 ROM	16K × 32 位，分为 64 页，每页 1024 字节
引导 ROM	1K × 32 位，分为 4 页，每页 1024 字节

基于 AHB 协议，Flash 接口实现指令和数据的访问，实现了对 Flash 存储器的操作（编程/擦除），编程/擦除的操作可在产品的整个工作电压范围内执行。

18.3 特性

- 读取接口（32 位）
- Flash 编程/擦除动作
- 编译选项（Code Option）包括代码加密选项（CS）

嵌入式 Flash 存储控制器（FMC）管理主存储块和编译选项的执行。编程/擦除操作需要的高压由内部产生。主要的 Flash 存储器可以通过设定不同的代码加密级别（CS）进行读/写保护。

在对 Flash 存储器进行写选择时，任何试图对 Flash 存储器进行读取的操作都会中断总线，写操作完成后，才会开始正确的读操作。这就意味着在进行写/擦除操作时，不能安排读取代码或者数据。

在 Flash 存储器进行写和擦除操作时，IHRC 应该由 FMC 设为 ON 状态。可以通过 ICP 和 ISP 对 Flash 存储器进行编程和擦除操作。

18.4 机构

Block	Name	Base Address	Size (Byte)
User ROM	Page 0	0x00000000 ~ 0x000003FF	1024
	Page 1	0x00000400 ~ 0x000007FF	1024
	.	.	
	.	.	
Boot Loader	Page 63	0x0000FC00 ~ 0x0000FFFF	1024
	Page 0	0x1FFF0000 ~ 0x1FFF03FF	1024
	Page 1	0x1FFF0400 ~ 0x1FFF07FF	1024
	Page 2	0x1FFF0800 ~ 0x1FFF0BFF	
	Page 3	0x1FFF0C00 ~ 0x1FFF0FFF	1024

18.5 读操作

作为一个通用的存储空间，嵌入式 Flash 模块可以直接进行访问。读操作是访问 Flash 模块内容并提供需要的数据。

读取界面由一个读取控制器（一面访问 Flash 存储器），和一个 AHB 接口（另一面与 CPU 联系）组成。读取界面的主要任务是产生控制信号，从 Flash 存储器中读取 CPU 需要的内容。

18.6 编程/擦除

Flash 存储器的擦除操作按页执行。

为了保证不产生过度的编程，IHRC 提供 Flash 编程和擦除时钟。

18.7 嵌入式引导加载程序

嵌入式引导加载程序利用 USART0 串行接口来重新编程 Flash 存储器，该程序位于引导 ROM 区域，由 SONiX 出厂时进行烧录。

18.8 FLASH 存储控制器（FMC）

FMC 控制 Flash 存储器的编程和擦除。

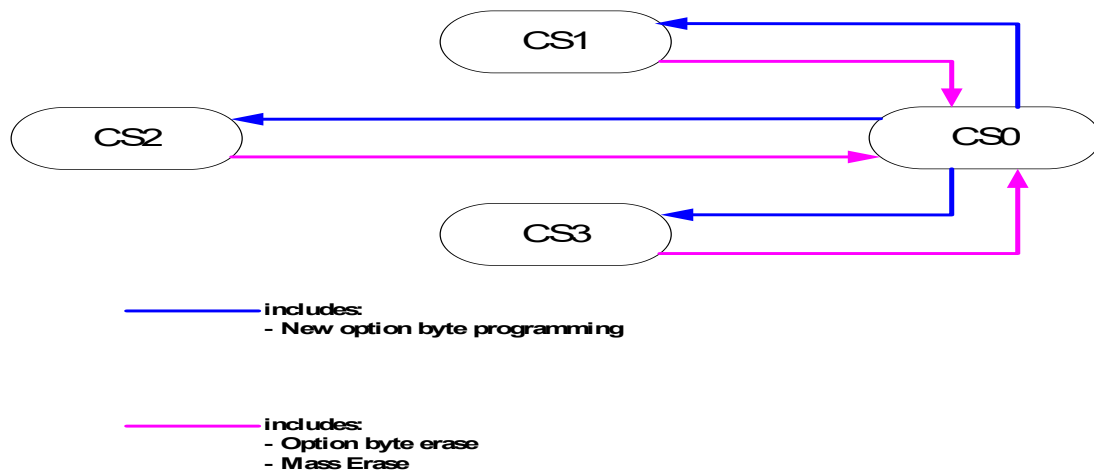
18.8.1 代码加密（CS）

代码加密是一种允许用户使能不同加密等级的机制，以便访问内置 Flash 和限制 ISP 操作。

* 注：只有 MCU 重新启动后，改变的代码加密才会生效。

User ROM		CS0	CS1	CS2	CS3	Description
WRITER	Read	O	X	X	X	
	Erase	O	O	O	O	若 CS 级别不是 CS0，则在擦除后会变成 CS0。
	Program	O	O	O	O	
FW (EEPROM emulation)	Read	O	O	O	O	
	Erase	O	O	O	X	
	Program	O	O	O	X	
SWD	Read	O	X	X	X	
	Erase	O	X	X	X	
	Program	O	X	X	X	

* 注：用户可能试图改变安全级别，从 CS3 改为 CS0，或者从 CS2 改为 CS0，或者从 CS1 改为 CS0，HW 会有以下操作：
1、首先将用户 ROM 区域擦除，用户不能在调试模式下执行该操作，这是由于 SWD 通信在擦除操作时可能会失败。
2、更新安全级别。



18.8.2 编程 FLASH 存储器

Flash 存储器可以一次编程 32 位，通过执行标准的字写入操作，CPU 可以编程主要的 Flash 存储器。FLASH_CTRL 寄存器的 PG 位必须设置为 1，FMC 读取指定的 Flash 存储器地址的数据，检查是否已经擦除，若没有擦除，跳过编程操作，通过 FLASH_STATUS 寄存器的 PGERR 位发出一个警告。通过 FLASH_STATUS 寄存器的 EOP 位显示编程操作完成。

标准模式下的主要 Flash 存储器编程流程如下：

- 1、设置 FLASH_CTRL 寄存器的 PG 位为 1；
- 2、编程数据到指定的位置；
- 3、等待 BUSY 位复位；
- 4、读取编程的值并进行校验。

18.8.3 擦除操作

Flash 存储器可以按页擦除，也可全部擦除（批量擦除）。

18.8.3.1 按页擦除

可以利用 FMC 的页擦除特性将 Flash 存储器的整页进行擦除。要擦除整页，其流程如下：

- 1、设置 FLASH_CTRL 寄存器的 PER 位为 1；
- 2、对 FLASH_ADDR 寄存器进行编程来选择一页进行擦除；
- 3、设置 FLASH_CTRL 寄存器的 STRT 位为 1；
- 4、等待 BUSY 位复位；
- 5、（可选）读取擦除页的内容并进行校验。

18.8.3.2 批量擦除

当 Flash 存储器读保护选项由受保护变为不受保护时，可以由 HW 在重新烧录读保护选项前将用户 ROM 进行批量擦除。

18.9 读保护

在编译选项中设置代码安全字节可以激活读保护选项。

当 Flash 存储器读保护选项由受保护变为不受保护时，可以由 HW 在重新烧录读保护选项前将用户 ROM 进行批量擦除。

18.10 HW CHECKSUM

HW checksum 是用户 ROM 的 checksum，若使能读保护，用户可以通过烧录器或者 ISP AP 读出 HW 的 checksum 值。

18.11 FMC 寄存器

基地址：0x4006 2000

18.11.1 Flash 低电压控制寄存器（FLASH_LPCTRL）

地址偏移量：0x00

Bit	Name	Description	Attribute	Reset
31:16	FMCKEY	FMC verify 位。 读取为 0，需要写入数据到该寄存器时，必须写入 0x5AFA 到 FMCKEY，否则会忽略对该寄存器的写动作。	W	0
15:2	Reserved		R	0
1:0	LPMODE[1:0]	Flash 低电压模式使能位。 00b: 禁止； 01b: 保留； 10b: 低速模式以省电（HCLK=ILRC=32KHz）； 11b: 保留。	R/W	0

18.11.2 Flash 状态寄存器（FLASH_STATUS）

地址偏移量：0x04

复位值：0x0000 0000

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	PGERR	编程错误标志位。 0: 读 → 无错误，写 → 将该位清零； 1: 编程前如果需要编程地址的数据不为 0xFFFFFFFF，则不能对该地址进行编程。	R/W	0
1	Reserved		R	0
0	BUSY	繁忙指示标志位。 0: Flash 操作不繁忙； 1: 正在处理 Flash 操作，在 Flash 操作开始时设置该位（同时 EOP 位清零），操作完成后或者 HW 出错时复位。	R	0

18.11.3 Flash 控制寄存器（FLASH_CTRL）

地址偏移量：0x08

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	CHK	Checksum 计算选择。 该位只能由 SW 设置为 1，BUSY 位复位时，该位也复位。	R/W	0
6	STARTE	开始擦除操作。 1: 该位由 SW 置 1 后，开始擦除操作；BUSY 位复位时，该位也复位。该位置 1 时，PER 位也应该置 1。	R/W	0
5:2	Reserved		R	0
1	PER	页擦除操作选择位。 该位只能由 SW 设置为 1，BUSY 位复位时，该位也复位。	R/W	0
0	PG	Flash 编程操作选择位。 该位只能由 SW 设置为 1，BUSY 位复位时，该位也复位。	R/W	0

18.11.4 Flash 数据寄存器 (FLASH_DATA)

地址偏移量: 0x0C

进行页编程操作时, 应该由 SW 操作该寄存器, 显示需要编程的数据。

Bit	Name	Description	Attribute	Reset
31:0	DATA[31:0]	需要编程的数据。	R/W	0

18.11.5 Flash 地址寄存器 (FLASH_ADDR)

地址偏移量: 0x10

必须由 SW 升级需要擦除或者编程的 Flash 地址, 填满 Flash 地址之前, PG 位或者 PER 位需设置为 1。

* 注: 当 FLASH_STATUS 寄存器的 BUSY 位置 1 时, 会阻止对该寄存器的写访问。

Bit	Name	Description	Attribute	Reset
31:0	FAR[31:0]	Flash 地址。 选择按页擦除时, 则选择一个页面进行擦除; 或者选择按页编程时, 则选择一个页面进行编程。	R/W	0

18.11.6 Flash Checksum 寄存器 (FLASH_CHKSUM)

地址偏移量: 0x14

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	CHKSUM[15:0]	用户 ROM 的 Checksum 值。	R	0

19 SERIAL-WIRE 调试 (SWD)

19.1 概述

SWD 功能集成到 ARM Cortex-M0 中，配置好的 ARM Cortex-M0 可支持 4 个断点和 2 个监控点。

19.2 特性

- 支持 ARM Serial Wire 调试 (SWD) 模式；
- 直接调试访问所有存储器、寄存器和外设；
- 调试期不需要目标源；
- 4 个断点；
- 2 个可作为触发器的数据观察点。

19.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
SWCLK	I	SWD 模式下的 Serial Wire 时钟引脚。	
SWDIO	I/O	SWD 模式下的 Serial Wire 数据输入/输出引脚。	

19.4 调试注意事项

19.4.1 局限性

调试模式改变了 ARM Cortex-M0 CPU 用于降低功耗的工作模式，而这一影响还会波及到整个系统。这意味着不应该在调试阶段进行功耗测量，否则测量结果会比正常应用操作时偏高。

在调试阶段，每当 CPU 停止工作时，SysTick 定时器会自动停止工作，其它的外设则不受影响。

19.4.2 恢复调试功能

用户代码可以禁止 SWD 功能以便将 P0.8 和 P0.9 作为 GPIO 使用，此时不能通过 SWD 功能来调试或者下载 FW。

在引导程序期间，SONiX 提供引导加载程序来检查 P2.2（引导引脚）的状态，若 P2.2 为低电平，MCU 会用引导加载程序代替用户程序，故不能禁止 SWD 功能。

退出引导加载程序后，用户程序可以将 P2.2 作为其它功能（如 GPIO）使用。

<p>* 注：我们强烈建议不要将 BOOT 引脚设置为输出引脚用于驱动 LED，否则 BOOT 引脚的状态会在 BOOT 步骤时变低。</p>

19.4.3 SWD 引脚上的内部上拉/下拉电阻

为了避免不受控制的 I/O 电平，在 SWD 输入引脚上内置了上拉电阻和下拉电阻。

- NJTRST: 内部上拉;
- SWDIO/JTMS: 内部上拉;
- SWCLK/JTCK: 内部下拉。

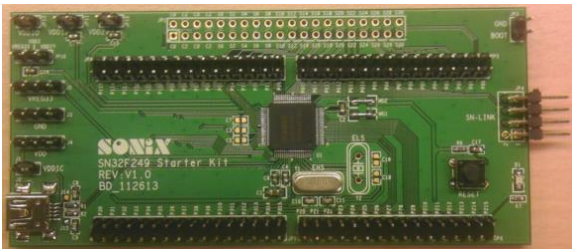
SW 一旦禁止了 SWD 功能，GPIO 控制器将再次控制相关功能。

20 开发工具

SONiX 提供一套嵌入式的 ICE 仿真系统进行 32 位系列单片机软件开发。

SONiX 32 位系列嵌入式 ICE 仿真系统包括：

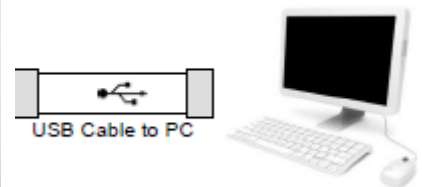
- SONiX 32 位 MCU Starter-Kit.
- SN-LINK-V3.0
- USB 线，用于 SN-LINK-V3.0 和 PC 的通讯
- IDE (KEIL RVMDK)



SONiX 32-bit MCU Starter-Kit.



SN-LINK-V3.0



IDE Tools

SONiX 32 位系列的嵌入式 ICE 仿真特性如下：

- 目标板的工作电压：1.8V~5.5V
- 高达 4 个硬件断点
- 系统时钟速率高达 50MHz
- 振荡器支持 IHRC, ILRC, EHS/ELS X'tal

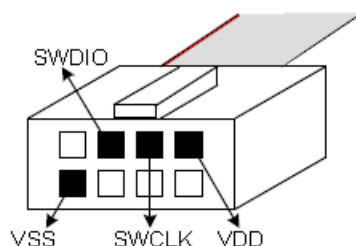
SONiX 32 位系统的嵌入式 ICE 仿真时有下拉限制：

- SWCLK 和 SWDIO 引脚与 GPIO 引脚共用，在嵌入式 ICE 模式下，该硬件的 GPIO 功能被屏蔽。

20.1 SN-LINK

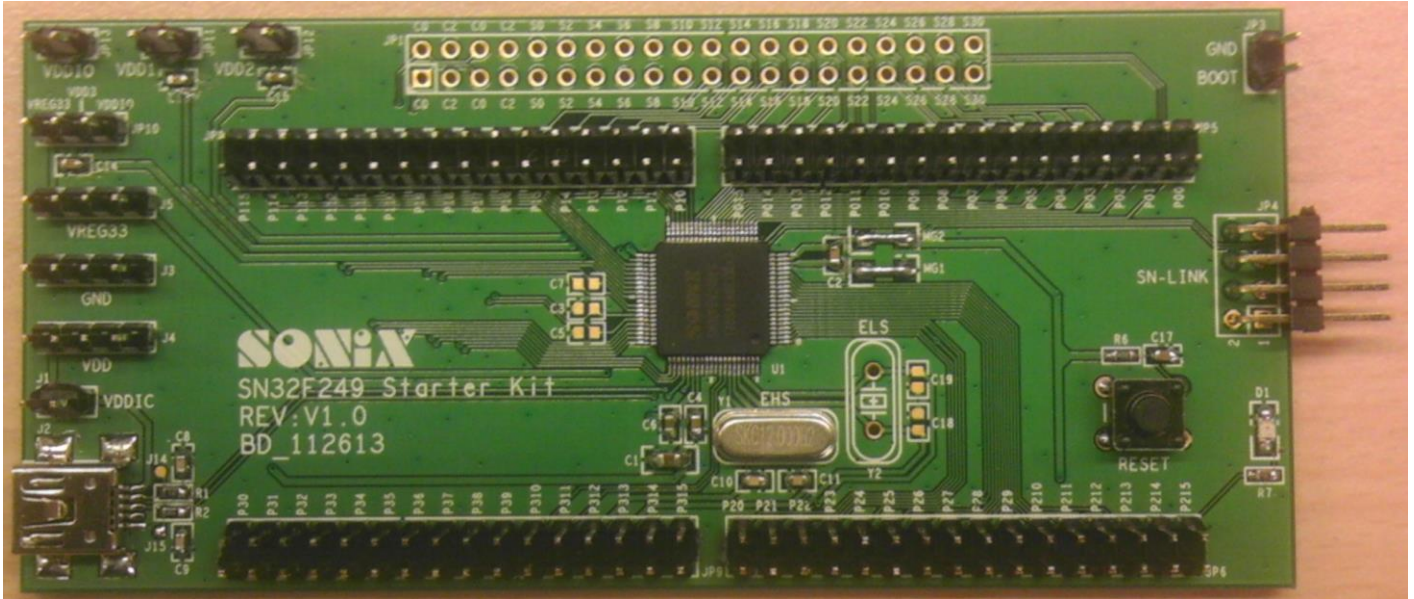
SN-LINK-V2 是一个高速仿真器，用于 SONiX 32 位系列单片机的仿真，基于 SWD 协议进行调试和编程。除了调试功能外，程序员还可以通过 SN-LINK-V2 从 PC 下载软件到单片机进行设计生产，甚至是大批量的生产。

SN-LINK-V2 通过 SWD 接口连接到 SONiX 32 位 MCU，该模块的引脚定义如下图所示：



20.2 SN32F249 STARTER-KIT

SN32F240 Starter-kit 是一个简易的开发平台，包括 SN32F240 实际芯片和 I/O 接口，用于输入信号或驱动用户应用的额外装置，当目标板未准备好时，它也是一个简单的开发平台。由于 SN32F240 系列单片机集成 SWD 调试电路，目标板也可替换 Starter-kit。



- J2: Mini USB 连接器。
- J1: USB 电源连接器。
- JP10: VDDIO1 电源连接器：选择 VDDIO1 源 (P0.0~P0.7) (板上的 5.0V/3.3V)。
- J3: GND 连接器。
- J4: VDD 连接器。
- J5: VREG33 输出连接器。
- U1: SN32F249F 实际芯片。
- RESET 按钮：外部重置触发源。
- D1: LED 测试。
- JP4: SN-LINK 连接器。
- JP3: 使单片机停在 Boot 加载程序。

21 电气特性

21.1 极限参数

Supply voltage (Vdd).....	- 0.3V ~ 5.5V
Input in voltage (Vin).....	Vss - 0.2V ~ Vdd + 0.2V
Operating ambient temperature (Topr).....	-40°C ~ + 85°C
Storage ambient temperature (Tstor).....	-40°C ~ + 125°C

21.2 电气特性

All of voltages refer to Vss, Typical Vdd = 5.0V, Fosc = 12MHz, ambient temperature is 25°C unless otherwise note.						
PARAMETER	SYM.	DESCRIPTION	MIN.	TYP.	MAX.	UNIT
Operating Voltage	Vdd1	Supply voltage for core and external rail	1.8	5	5.5	V
	Vdd2	USB mode	3.1	5	5.25	V
VDD rise rate	VPOR	VDD rise rate to ensure internal power-on reset	0.05	-	-	V/ms
Power Consumption						
Supply Current	Idd1	Normal mode	System clock = 12MHz		5	mA
			[1][2][3]			
			System clock = 50MHz		16	mA
			[1][3][4]			
	Idd2	Sleep Mode	System clock = 12MHz		1	mA
		[1][2][3][5]]				
Idd3	Sleep Mode	System clock = 32KHz		100	uA	
		[1][3][5][7]				
Idd4	Deep-sleep Mode	Vdd=5V		5	uA	
		[1][3][5]				
Port Pins, RESET pin						
High-level input voltage	VIH		0.8Vdd	-	Vdd	V
Low-level input voltage	VIL		Vss	-	0.2Vdd	V
Input voltage	Vi		0	-	Vdd	V
Output voltage	Vo		0	-	Vdd	V
I/O port pull-up resistor	RPU	Vin = Vss , Vdd = 5.0V	50	75	100	KΩ
I/O port pull-down resistor	RPD	Vin = 5.0V	50	75	100	KΩ
I/O High-level output source current	IOH	Standard port and RESET pins	VOP = Vdd - 0.5V;	5	10	mA
		High-drive output pin	VOP = Vdd - 0.5V	12	20	mA
I/O Low-level output sink current	IOL	Standard port and RESET pins	VOP = Vss + 0.5V	5	10	mA
		High-sinking output pin	VOP = Vdd - 0.5V	12	20	mA
ADC						
ADC Operating Voltage	VADC		2.5	-	5.5	V
AIN0 ~ AIN13 input voltage	Vani		0	-	Avrefh	V
ADC reference Voltage	Vref		2.5	-	-	V
*ADC enable time	Tast	Ready to start convert after set ADENB = "1"	100	-	-	us
*ADC current consumption	IADC	Vdd=5V, ADS=0	-	220	-	uA
ADC Clock Frequency	FADCLK	Vdd=5V	-	-	16	MHz
ADC Conversion Cycle Time	FADCY L	VDD=2.5V~5.5V	64	-	-	1/FADCLK
ADC Sampling Rate	FADSM P	Vdd=5V		-	250	KHz
Differential Nonlinearity	DNL	Vdd=5.5V , AVREFH=2.4V	-1	-	+1	LSB
Integral Nonlinearity	INL	Vdd=5.5V , AVREFH=2.4V	-1	-	+1	LSB
No Missing Code	NMC	Vdd=5.5V , AVREFH=2.4V	10	-	12	Bits
ADC offset Voltage	VADCO fset		-5		+5	mV
Temperature sensor Range	TR	Temperature Sensor Operation Range	-10	-	+70	°C
Temperature Sensitivity	TS	Temperature Sensor Sensitivity.		3.53		mV/°C
Temperature Sensor Accuracy	ETS	One Temperature point Calibration.	-10	-	+10	%
		Two Temperature points Calibration	-1	-	+1	%
FLASH						
Supply Voltage	Vdd1		1.8		Vdd	V
Endurance time	TEN	Erase + Program	20K	*100K	-	Cycle
Page erase time	TPE	Vdd = 2.5V, 1-Page (1024 bytes).	-	25	30	ms

1-Word Programming time	TPG	Vdd = 2.5V, 1-Word (32 bits).	-	60	70	us	
LCD							
R-Type LCD Operation Current	IRLCD	Vdd = 3.3V, 1/3 bias, 400k,bias resistor, No panel	-	3	5	uA	
		Vdd = 3.3V, 1/3 bias, 35k,bias resistor, No panel	-	30	45	uA	
1C-Type LCD Operation Current	I1CLCD'	Vdd = 3.3V, 1/3 bias, 400k,bias resistor, LCD all dots ON, LCD rate=ILRC/64, 1/4 duty, MCU in Deep sleep mode	-	10	15	uA	
		Vdd = 3.3V, 1/3 bias, No panel	-	18	25	uA	
4C-Type LCD Operation Current	I4CLCD'	Vdd = 3.3V, 1/3 bias, LCD all dots ON, LCD rate=ILRC/64, 1/4 duty, MCU in Deep sleep mode	-	50	65	uA	
		Vdd = 3.3V, 1/3 bias, No panel	-	7	15	uA	
C-Type VLCD Output Voltage	VLCD1	Vdd = 1.8~5.5V. VCP[3:0] = 0011b	2.85	3.0	3.15	V	
MISC							
		Interrupt/Reset	Level 0	1.70	1.80	1.90	V
			Level 1	1.90	2.00	2.10	V
			Level 2	2.30	2.40	2.50	V
			Level 3	2.60	2.70	2.80	V
			Level 4	2.90	3.00	3.10	V
			Level 5	3.50	3.60	3.70	V
3.3V Regulator Output voltage	Vreg33	VCC ≥ 3.60V, IVREG33 ≥ 60 mA	3.0		3.6	V	
IHRC Freq.	FIHRC	T=25°C, Vdd=1.8V~ 5.5V	11.88	12	12.12	MHz	
		T=-40°C~85°C, Vdd=1.8V~5.5V	11.70	12	12.30	MHz	
	FIHRC2	T=-40°C~85°C, Vdd=1.8V~5.5V, USB function ON	11.97	12	12.03	MHz	

* These parameters are for design reference, not tested.

[1] IDD measurements were performed with all pins configured as GPIO outputs driven LOW and pull-up resistors disabled and VDD=5V

[2] IHRC and ILRC are enabled, external X'tal are disabled, and PLL is disabled.

[3] LVD and all peripherals are disabled.

[4] IHRC is disabled, external high X'tal is enabled, and PLL is enabled.

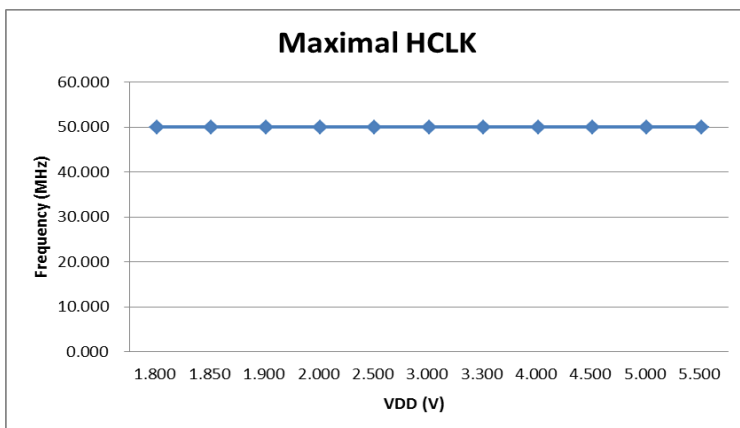
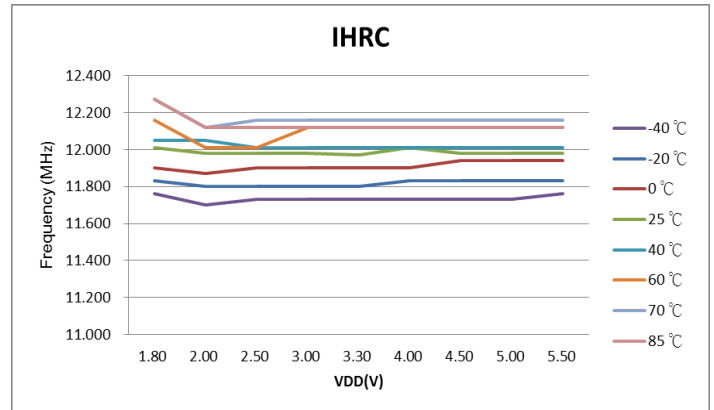
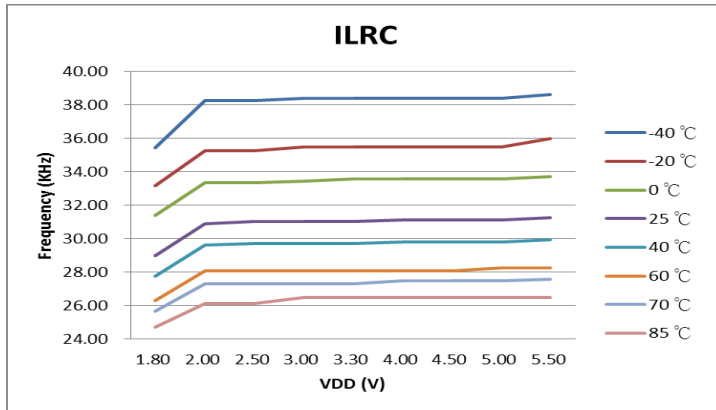
[5] All oscillators and analog blocks are turned off.

[6] DPDWAKEUP pin is pulled HIGH internally.

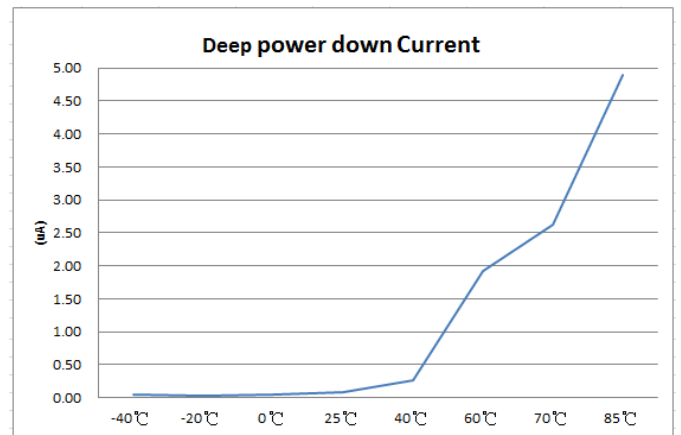
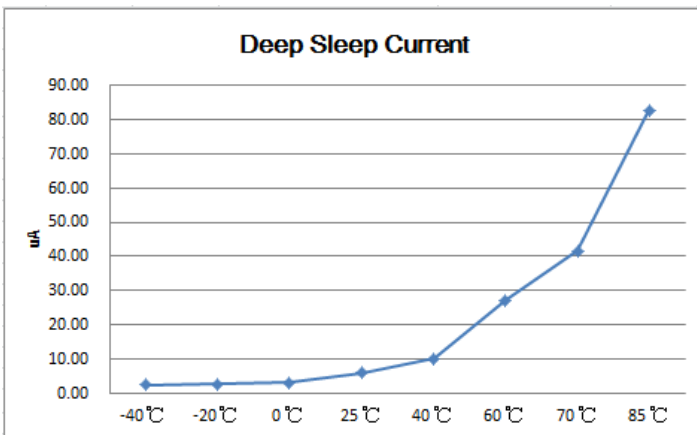
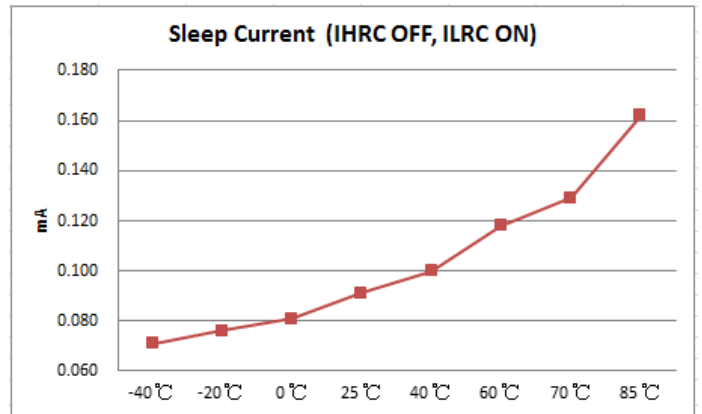
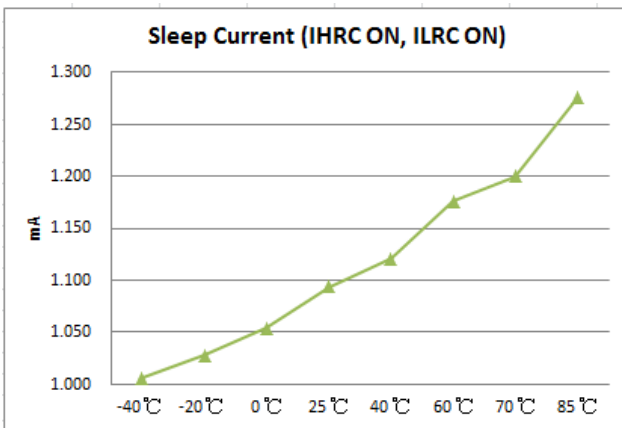
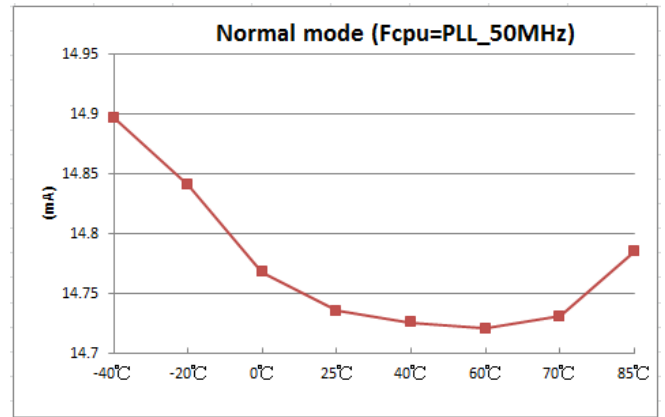
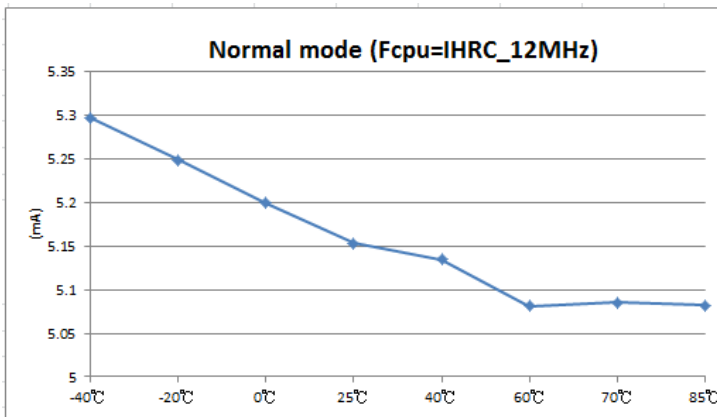
[7] ILRC is enabled, IHRC and external X'tal are disabled, and PLL is disabled.

21.3 特性曲线图

本章所列的各曲线图仅作设计参考，其中给出的部分数据可能超出了芯片指定的工作范围，为保证芯片的正常工作，请严格参照电气特性说明。



*供给电流 V.S. 工作温度 (工作条件: 所有的引脚都为 GPIO 引脚并输出低电平, VDD=3.3V 时禁止上拉电阻)

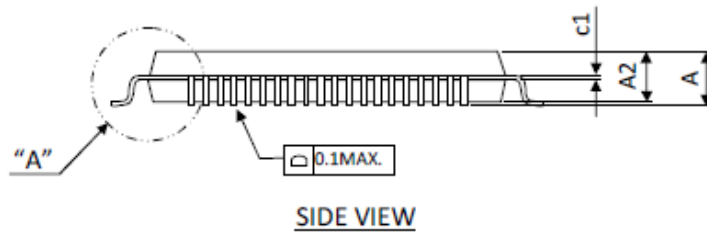
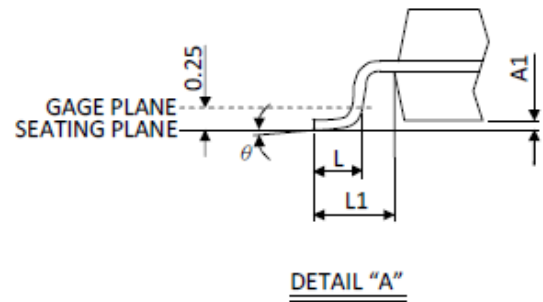
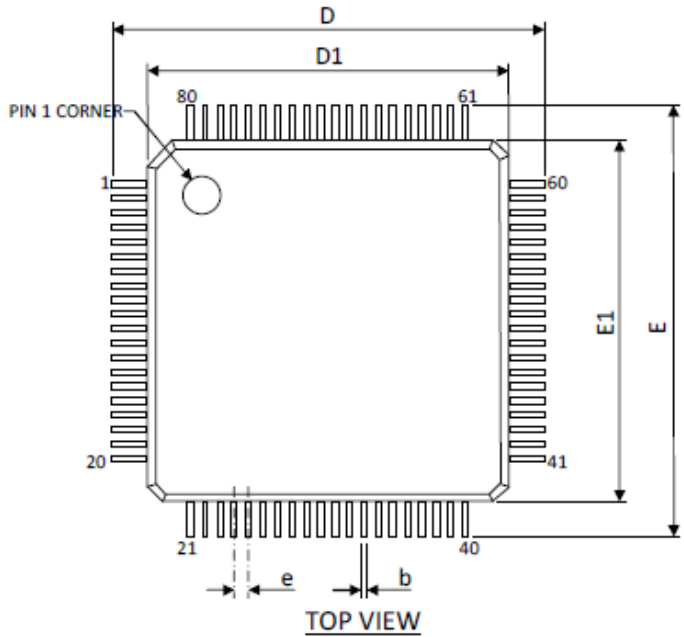


22 FLASH ROM 引脚烧录

SN32F240/230 系列的烧录信息											
单片机名称		SN32F249F SN32F239F	SN32F248F SN32F238F	SN32F247F SN32F237F	SN32F246J SN32F236J	SN32F245J SN32F235J					
MP PRO Writer		Flash IC / JP3 引脚配置									
Number	Name	Number	Pin	Number	Pin	Number	Pin	Number	Pin	Number	Pin
1	VDD	19, 20, 41, 42, 67	VDD	15, 16, 33, 55	VDD	13, 14, 25, 40	VDD	10, 11, 22, 37	VDD	9, 10, 17, 26	VDD
2	GND	17, 18, 51, 69	VSS	14, 56	VSS	12, 41	VSS	9, 38	VSS	8, 27	VSS
3	CLK	74	P3.12	61	P3.12	46	P3.12	43	P3.12	32	P3.12
4	CE										
5	PGM	71	P0.8	58	P0.8	43	P0.8	40	P0.8	29	P0.8
6	OE	70	P0.9	57	P0.9	42	P0.9	39	P0.9	28	P0.9
7	D1										
8	D0										
9	D3										
10	D2										
11	D5										
12	D4										
13	D7										
14	D6										
15	VDD										
16	-										
17	HLS										
18	RST										
19	-										
20	ALSB/PDB	73	P3.11	60	P3.11	45	P3.11	42	P3.11	31	P3.11

23 封装信息

23.1 LQFP 80 PIN

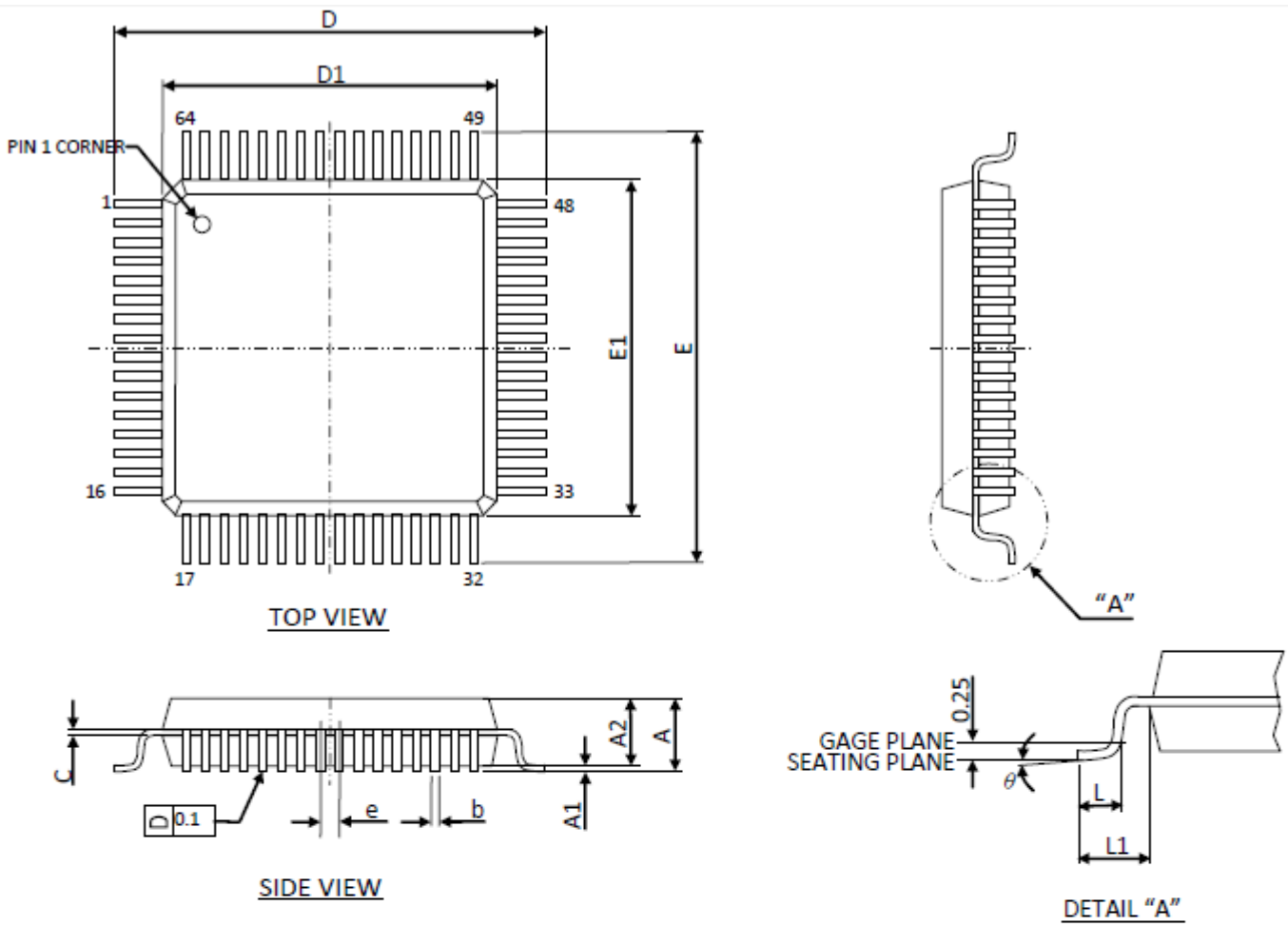


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.2	0.002	--	0.008
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.13	0.18	0.23	0.005	0.007	0.009
c1	0.09	--	0.18	0.004	--	0.007
D	12 BSC			0.472 BSC		
D1	10 BSC			0.394 BSC		
e	0.4 BSC			0.016 BSC		
E	12 BSC			0.472 BSC		
E1	10 BSC			0.394 BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.0 REF			0.039 REF		
θ	0°	3.5°	7°	0°	3.5°	7°

Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

23.2 LQFP 64 PIN

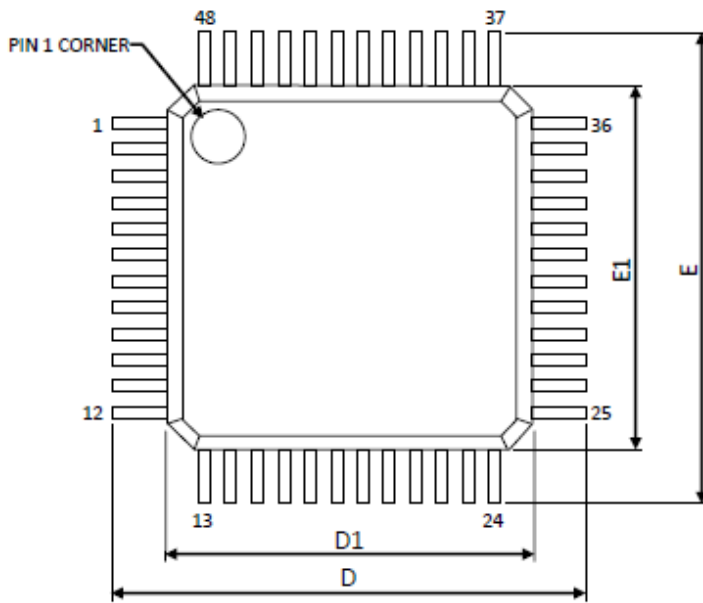


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.25	0.002	--	0.01
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.13	0.19	0.25	0.005	0.007	0.010
c	0.09	--	0.20	0.004	--	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
e	0.40 BSC			0.016 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
L	0.4	0.60	0.8	0.016	0.024	0.032
L1	1.00 REF			0.039 REF		
θ	0°	3.5°	7°	0°	3.5°	7°

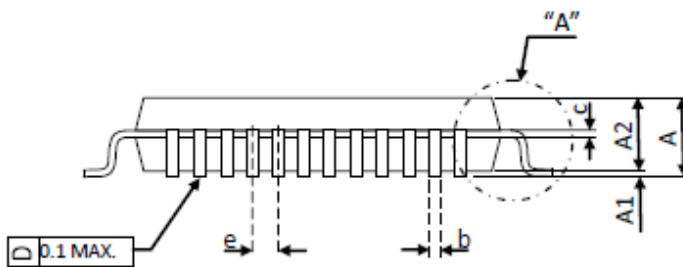
Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

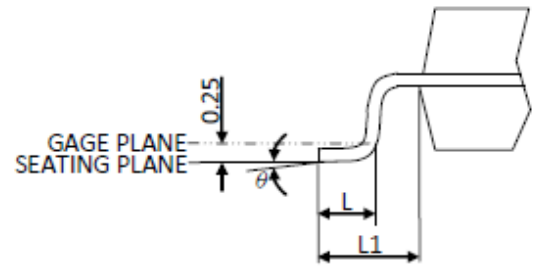
23.3 LQFP 48 PIN



TOP VIEW



SIDE VIEW



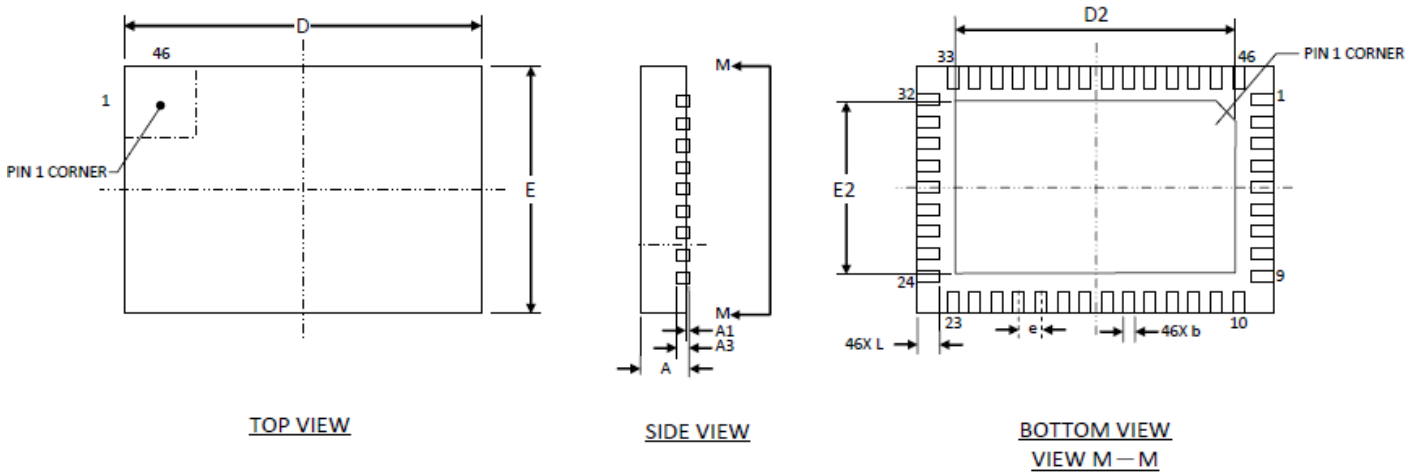
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in 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
b	0.17	0.22	0.27	0.007	0.009	0.011
c	0.09	--	0.20	0.004	--	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
e	0.50 BSC			0.020 BSC		
L	0.40	0.60	0.80	0.016	0.024	0.031
L1	1.00 REF			0.039 REF		
θ	0°	3.5°	7°	0°	3.5°	7°

Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

23.4 QFN 46 PIN

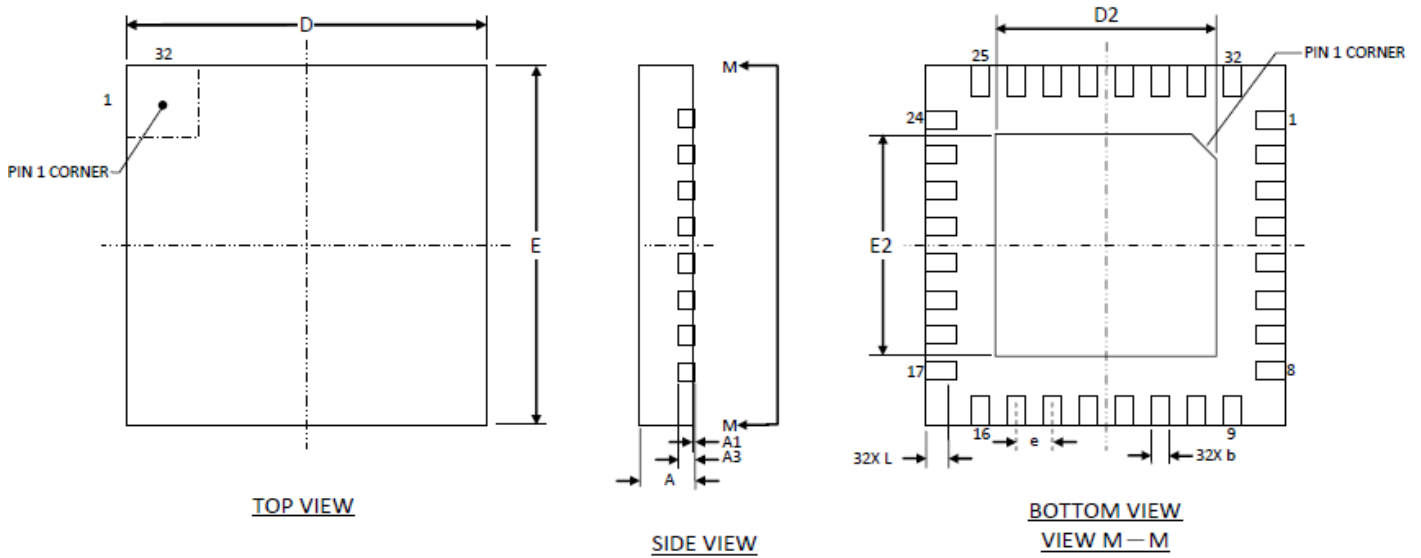


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.203 REF			0.008 REF		
b	0.20	0.25	0.30	0.008	0.010	0.012
D	6.5 BSC			0.256 BSC		
E	4.5 BSC			0.177 BSC		
e	0.4 BSC			0.016 BSC		
D2	5.00	5.10	5.20	0.197	0.201	0.205
E2	3.00	3.10	3.20	0.118	0.122	0.126
L	0.30	0.40	0.50	0.012	0.016	0.020

Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)

23.5 QFN 33 PIN 5x5



SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.203 REF			0.008 REF		
b	0.18	0.25	0.30	0.007	0.010	0.012
D	5.00 BSC			0.197 BSC		
E	5.00 BSC			0.197 BSC		
e	0.50 BSC			0.020 BSC		
D2	2.60	3.10	3.60	0.102	0.122	0.142
E2	2.60	3.10	3.60	0.102	0.122	0.142
L	0.30	0.40	0.50	0.012	0.016	0.020

Notes :

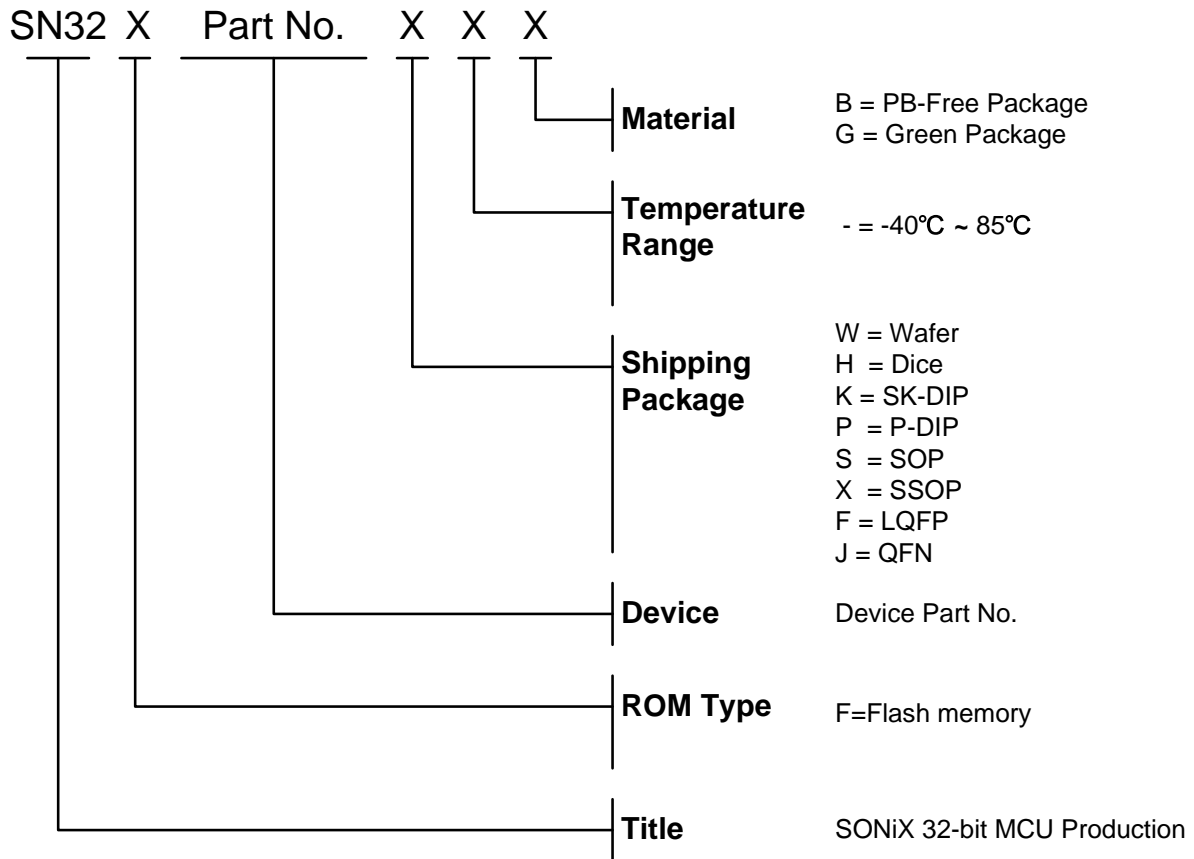
1. CONTROLLING DIMENSION : MILLIMETER (mm)

24 芯片正印命名规则

24.1 概述

SONiX 32 位单片机产品具有多种型号，本章将给出所有 32 位单片机分类命名规则。

24.2 芯片型号说明

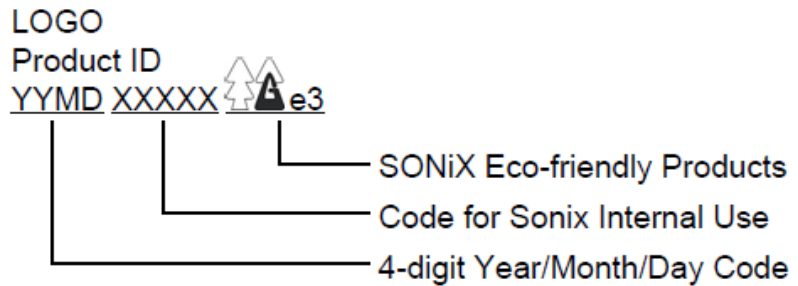


24.3 命名举例

Name	ROM 类型	器件 (Device)	封装形式	温度范围	封装材料
SN32F249FG	Flash memory	249	LQFP	0°C~70°C	绿色封装
SN32F248FG	Flash memory	249	LQFP	0°C~70°C	绿色封装
SN32F247FG	Flash memory	249	LQFP	0°C~70°C	绿色封装
SN32F246JG	Flash memory	249	QFN	0°C~70°C	绿色封装
SN32F245JG	Flash memory	249	QFN	0°C~70°C	绿色封装
SN32F249W	Flash memory	249	Wafer	0°C~70°C	-
SN32F249H	Flash memory	249	Dice	0°C~70°C	-
SN32F239FG	Flash memory	239	LQFP	0°C~70°C	绿色封装
SN32F238FG	Flash memory	239	LQFP	0°C~70°C	绿色封装
SN32F237FG	Flash memory	239	LQFP	0°C~70°C	绿色封装
SN32F236JG	Flash memory	239	QFN	0°C~70°C	绿色封装
SN32F235JG	Flash memory	239	QFN	0°C~70°C	绿色封装

24.4 日期码规则

下图为标记示例，IC 的标志 ID 可能因为不同的封装有所不同。



SONiX 公司保留对以下所有产品在可靠性，功能和设计方面的改进作进一步说明的权利。SONiX 不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，SONiX 的产品不是专门设计来应用于外科植入、生命维持和任何 SONiX 产品的故障会对个体造成伤害甚至死亡的领域。如果将 SONiX 的产品应用于上述领域，即使这些是由 SONiX 在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接产生的律师费用，并且用户保证 SONiX 及其雇员、子公司、分支机构和销售商与上述事宜无关。

总公司：

地址：台湾新竹县竹北市台元街 36 号 10 楼之一

电话：886-3-5600-888

传真：886-3-5600-889

台北办事处：

地址：台北市松德路 171 号 15 楼之 2

电话：886-2-2759 1980

传真：886-2-2759 8180

香港办事处：

地址：香港新界沙田炭禾盛街 11 号，中建电讯大厦 26 楼 03 室

电话：852-2723 8086

传真：852-2723 9179

松翰科技（深圳）有限公司

地址：深圳市南山区高新技术产业园南区 T2-B 栋 2 层

电话：86-755-2671 9666

传真：86-755-2671 9786

技术支持：

Sn8fae@SONiX.com.tw